A Runge–Kutta BVODE Solver with Global Error and Defect Control

JASON J. BOISVERT
University of Saskatchewan
PAUL H. MUIR
Saint Mary's University
and
RAYMOND J. SPITERI
University of Saskatchewan

Boundary value ordinary differential equations (BVODEs) are systems of ODEs with boundary conditions imposed at two or more distinct points. The global error (GE) of a numerical solution to a BVODE is the amount by which the numerical solution differs from the exact solution. The defect is the amount by which the numerical solution fails to satisfy the ODEs and boundary conditions. Although GE control is often familiar to users, the defect controlled numerical solution can be interpreted as the exact solution to a perturbation of the original BVODE. Software packages based on GE control and on defect control are in wide use.

The defect control solver, `BVP_SOLVER`, can provide an a posteriori estimate of the GE using Richardson extrapolation. In this paper, we consider three more strategies for GE estimation based on (i) the direct use of a higher order discretization formula (HO), (ii) the use of a higher order discretization formula within a deferred correction (DC) framework, and (iii) the product of an estimate of the maximum defect and an estimate of the BVODE conditioning constant, and demonstrate that the HO and DC approaches have superior performance. We also modify `BVP_SOLVER` to introduce *GE control*.

Categories and Subject Descriptors: G.1.7 [**Numerical Analysis**]: Ordinary Differential Equations – Boundary-Value Problems; G.1.0 [**Numerical Analysis**]: General – Conditioning and Ill-Conditioning

General Terms: Experimentation, Performance, Reliability

Additional Key Words and Phrases: boundary value ordinary differential equations, conditioning, defect control, deferred correction, global error estimation, Richardson extrapolation, Runge–Kutta methods

This work was supported by the Mathematics of Information Technology and Complex Systems Network and by the Natural Sciences and Engineering Research Council of Canada.
Authors' addresses J. J. Boisvert and R. J. Spiteri, Department of Computer Science, University of Saskatchewan, Saskatoon SK, Canada, S7N 5C9, P. H. Muir, Department of Mathematics and Computing Science, Saint Mary's University, Halifax, NS, Canada, B3H 3C3

1. INTRODUCTION

In this paper, we consider software for the numerical solution of boundary value ordinary differential equations (BVODEs) having the form

$$\mathbf{y}'(x) = \mathbf{f}(x, \mathbf{y}(x)), \quad a \leq x \leq b, \qquad \mathbf{g}_a(\mathbf{y}(a)) = \mathbf{0}, \quad \mathbf{g}_b(\mathbf{y}(b)) = \mathbf{0}, \tag{1}$$

where $\mathbf{y}(x)$, $\mathbf{f}(x, \mathbf{y}(x))$, and $[\mathbf{g}_a^T(\mathbf{y}(a)), \mathbf{g}_b^T(\mathbf{y}(b))]^T$ are vector functions of length n.

There are two common approaches to controlling solution accuracy in software for BVODEs: global error (GE) control and defect control. The GE is the difference between the numerical solution and the exact solution. The defect of a continuous numerical solution is the amount by which the solution fails to satisfy the ODEs and boundary conditions. Typically the GE or defect estimate is scaled to accommodate a blend of absolute and relative tolerances, based on the numerical solution or its derivative. This estimate is used to adapt the computation to return a solution for which the estimate is less than a user provided tolerance. We refer to such solvers as providing GE control or defect control, respectively. Although control of the GE is often more familiar to users, control of the defect has an interesting backward error interpretation: the defect controlled numerical solution is the exact solution to a perturbation (on the order of the tolerance) of the original BVODE; see Section 4.3.

This paper describes the development of a BVODE solver that features hybrid defect control/GE control. Our approach is to modify the defect control solver, `BVP_SOLVER` [Shampine et al. 2006]. `BVP_SOLVER` returns a defect controlled numerical solution but provides an option for an a posteriori estimate of the GE of the numerical solution based on Richardson extrapolation (RE). We first introduce, within the `BVP_SOLVER` framework, implementations of three GE estimation schemes as alternatives for the a posteriori estimate of the GE. These schemes are based on (i) the direct use of a higher order discretization formula, (ii) the use of a higher order discretization formula within a deferred correction framework, and (iii) the product of an estimate of the maximum defect and an estimate of the BVODE conditioning constant. We compare their performance with respect to accuracy and efficiency. We then further modify `BVP_SOLVER` to introduce an option for estimation and *control* of the GE. This new version of `BVP_SOLVER` provides options for GE control, defect control, and combinations thereof; it is available at `http://cs.smu.ca/~muir/BVP_SOLVER_Webpage.shtml`. We provide numerical results demonstrating the use of these options within the new version of `BVP_SOLVER`. Although the discussion and examples are generally presented in the context of `BVP_SOLVER`, many of the conclusions are expected to be applicable to a wider class of general purpose BVODE solvers; see Section 7.

It is important to note that RE and other GE estimation schemes suffer from two well known fundamental difficulties. First, at coarse tolerances, when the mesh is not sufficiently fine, the asymptotic behavior upon which the estimates are based may not be applicable. Second, at sharp tolerances, the estimates may be affected by the presence of round-off error. On the other hand, given a continuous numerical solution (and its derivative), it is possible to compute the defect at any point in the problem domain and estimate the maximum value of the defect even if the mesh is coarse. These observations motivate our investigation of hybrid defect control/GE

control options for BVP_SOLVER, as mentioned above.

The paper is organized as follows. Computational approaches for solving BVODEs are commonly divided into initial value methods and global methods. This paper focuses on the second class of methods, and Section 2 provides a brief review of software packages of this type. Section 3 reviews the algorithms used in BVP_SOLVER. Section 4 describes the three alternative GE estimation techniques and their efficient implementation within BVP_SOLVER. This section also briefly discusses a slight modification of the RE based scheme currently implemented in BVP_SOLVER. Section 5 presents numerical experiments comparing the four GE estimators with respect to accuracy and efficiency. Section 6 introduces a new version of BVP_SOLVER that provides options for defect or GE control as well as options for combinations of defect and GE control. The results presented in Sections 5 and 6 are a selection of the set of experiments presented in [Boisvert et al. 2012]. Section 7 provides our conclusions and suggestions for future work.

2. BVODE SOLVERS

2.1 Global Error and Local Truncation Error Control Solvers

In some codes of this class, a direct estimate of the GE is computed and controlled; in others, only an estimate of the local truncation error (LTE) is computed and controlled. Section 5.2 of [Ascher et al. 1995] describes the relationship between these two errors. For a consistent numerical method, i.e., having a LTE that is $O(h^p)$ for $p \geq 1$, where h is the maximum mesh spacing, the global error is bounded by the product of the LTE and the conditioning constant for the BVODE. Assuming that the BVODE has a reasonably sized conditioning constant, control of the LTE therefore implies control of the GE.

One of the earliest BVODE solvers that uses error control is the Fortran collocation solver COLSYS [Ascher et al. 1981]. Several modifications of this solver have been developed to improve its capabilities; examples include COLNEW [Bader and Ascher 1987], COLDAE [Ascher and Spiteri 1994], and COLMOD [Cash et al. 1995]. These solvers estimate the error in two ways. First the discretization error is estimated and used for mesh refinement and a preliminary assessment of the acceptability of the numerical solution. Because this estimate may be unreliable for crude tolerances or high order, when it appears from this estimate that the solution is acceptable, an estimate of the GE is computed using RE. Only after this second estimate satisfies the user tolerance is the numerical solution accepted.

A special class of multi-step methods for BVODEs, called Top-Order Methods, is employed in the MATLAB BVODE solver, TOM [Mazzia and Trigiante 2004]. This solver employs mesh refinement based on an estimate of the LTE and an estimate of the conditioning of the BVODE. The solution is accepted when the LTE estimate satisfies a user tolerance.

Deferred correction has been the basis for a number of codes for the numerical solution of BVODEs. The PASVA3 solver [Lentini and Pereyra 1974; 1977] employs deferred correction based on the box scheme. An experimental solver, generalizing the approach employed in PASVA3 through the use of Mono-Implicit Runge–Kutta (MIRK) methods (see, e.g., [Burrage et al. 1994] and references therein) is discussed in [Gupta 1985]. MIRK methods and Lobatto collocation methods (see,

e.g., [Ascher et al. 1995]) are employed within a deferred correction framework in the Fortran BVODE solvers TWPBVP [Cash and Wright 1991] and ACDC [Cash et al. 1995], and the related solver, TWPBVPL [Capper et al. 2007]. All of these solvers control estimates of the LTE and base mesh refinement on these estimates. Extensions that consider mesh refinement based on the LTE estimates and on estimates of the conditioning constant of the BVODE have led to new versions of TWPBVP and TWPBVPL, called TWPBVPC and TWPBVPLC [Cash and Mazzia 2006]. In all of these solvers the solution is accepted when the LTE satisfies the user tolerance. We also note the MATLAB BVODE solver sbvp [Auzinger et al. 2002] controls the GE based on a modification of deferred correction.

2.2 Defect Control Solvers

Most BVODE solvers provide a continuous solution approximation through the use of some form of interpolation. In such cases, it is suggested in [Hanson and Enright 1983] that one use an estimate of the maximum defect rather than the GE to assess solution quality and guide mesh refinement.

The MATLAB codes bvp4c [Kierzenka and Shampine 2001], bvp5c [Kierzenka and Shampine 2008], and bvp6c [Hale and Moore 2008] and the Fortran codes MIRKDC [Enright and Muir 1996] and BVP_SOLVER all use defect control. The bvp4c, bvp6c, MIRKDC, and BVP_SOLVER are based on MIRK formulas and do not attempt to *directly* control the GE. The solver bvp5c is based on a four point Lobatto formula, and it is shown in [Kierzenka and Shampine 2008] that a scaled norm of the defect asymptotically approaches the norm of the GE. Thus for bvp5c, direct control of the defect is equivalent to direct control of the GE, and it can therefore be described as controlling both.

It is possible for a numerical solution with an estimated maximum defect that satisfies a user tolerance to nonetheless have a large GE. In extreme cases, a defect control solver can return a numerical solution for a problem that has no solution. The paper [Shampine and Muir 2004] refers to such solutions as *pseudosolutions* and provides examples where bvp4c and MIRKDC return pseudosolutions under certain conditions. It should be emphasized that such a solution is in fact an acceptable numerical solution in the following sense: the solver has returned a numerical solution whose estimated defect satisfies the user tolerance. In such cases, the numerical solution is the exact solution to a BVODE that is reasonably close to the original one. However, if the BVODE is ill-conditioned and the tolerance is coarse, the solution of the perturbed problem may not be close to the solution of the original problem. This suggests that it can be important for a defect control solver to also provide an assessment of the GE of the defect controlled numerical solution it computes.

3. REVIEW OF BVP_SOLVER

BVP_SOLVER is capable of solving a first order system of n ODEs of the form

$$\mathbf{y}'(x) = \left(\frac{1}{x-a}\right)\mathbf{\Lambda}\mathbf{y}(x) + \mathbf{f}(x, \mathbf{y}(x), \mathbf{p}), \qquad a \le x \le b,$$

subject to separated nonlinear two point boundary conditions (BCs)

$$\mathbf{g}_a(\mathbf{y}(a), \mathbf{p}) = \mathbf{0}, \qquad \mathbf{g}_b(\mathbf{y}(b), \mathbf{p}) = \mathbf{0}.$$

Here $\mathbf{y}(x)$ and $\mathbf{f}(x, \mathbf{y}(x))$ are vector functions of length n and $\mathbf{p}$ is an optional vector of length n_p of unknown parameters. The vector function $[\mathbf{g}_a^T(\mathbf{y}(a), \mathbf{p}), \mathbf{g}_b^T(\mathbf{y}(b), \mathbf{p})]^T$ is of length $n + n_p$. The $n \times n$ constant matrix $\mathbf{\Lambda}$ is optional. The presence of the singularity at $x = a$ is not germane to the current study and so for simplicity we assume the form (1).

In order to solve a BVODE, `BVP_SOLVER` generates a system of nonlinear equations for which the unknowns, $\mathbf{y}_i$, are approximations to the solution values, $\mathbf{y}(x_i)$, at the points of a mesh that partitions the problem domain: $a = x_0 < x_1 < \ldots < x_N = b$, where the initial mesh can be specified by the user or by default is taken to be uniform with $N = 10$. Let $h_{i+1} = x_{i+1} - x_i$, $i = 0, 1, \ldots, N-1$. On the subinterval, $[x_i, x_{i+1}]$, these nonlinear equations have the form

$$\boldsymbol{\phi}_{i+1}(\mathbf{y}_i, \mathbf{y}_{i+1}) = \mathbf{y}_{i+1} - \mathbf{y}_i - h_{i+1} \sum_{j=1}^{s} b_j \mathbf{f}(x_i + c_j h_{i+1}, \mathbf{y}_{ij}) = \mathbf{0}, \tag{2}$$

where

$$\mathbf{y}_{ij} = (1 - v_j)\mathbf{y}_i + v_j \mathbf{y}_{i+1} + h_{i+1} \sum_{k=1}^{j-1} a_{j,k} \mathbf{f}(x_i + c_k h_{i+1}, \mathbf{y}_{ik}), \tag{3}$$

for $j = 1, 2, \ldots, s$, are the stages of a MIRK method; see, e.g., [Muir 1999] and references within. The coefficients, $v_j, b_j, a_{j,k}$, $j = 1, 2, \ldots, s$, $k = 1, 2, \ldots, j - 1$, define the MIRK method, and $c_j = v_j + \sum_{k=1}^{j-1} a_{j,k}$. We note that in (3) the computation of $\mathbf{y}_{ij}$, i.e., the jth stage for the ith subinterval, depends only on $\mathbf{y}_i$, $\mathbf{y}_{i+1}$, and the *previously computed* stages, $\mathbf{y}_{ik}, k = 1, 2, \ldots, j-1$; hence (3) captures a form of parameter condensation.

Equation (2) represents n nonlinear equations involving the unknowns $\mathbf{y}_i$ and $\mathbf{y}_{i+1}$. Taking these equations for all subintervals together with the BCs gives a system of $(N+1)n$ nonlinear equations whose solution gives a discrete approximate solution at the mesh points, $\mathbf{Y} \equiv [\mathbf{y}_0^T, \mathbf{y}_1^T, \ldots, \mathbf{y}_N^T]^T$. This nonlinear system has the form

$$\mathbf{\Phi}(\mathbf{Y}) \equiv \begin{pmatrix} \mathbf{g}_a(\mathbf{y}_0) \\ \boldsymbol{\phi}_1(\mathbf{y}_0, \mathbf{y}_1) \\ \vdots \\ \boldsymbol{\phi}_N(\mathbf{y}_{N-1}, \mathbf{y}_N) \\ \mathbf{g}_b(\mathbf{y}_N) \end{pmatrix} = \mathbf{0}. \tag{4}$$

System (4) is solved using a modified Newton iteration, which requires the evaluation and factorization of the Jacobian

$$\boldsymbol{J}_{\mathbf{\Phi}}(\mathbf{Y}) \equiv \frac{\partial \mathbf{\Phi}(\mathbf{Y})}{\partial \mathbf{Y}}. \tag{5}$$

`BVP_SOLVER` implements a hybrid damped Newton/fixed Jacobian iteration. When there are convergence issues, the solver re-evaluates the Jacobian and uses a damping factor to control the contribution of the Newton correction to the next iterate. Otherwise, it holds the Jacobian constant and takes full Newton steps as long as convergence is sufficiently rapid. Once the Newton iteration converges, we obtain

the discrete solution, $\{\mathbf{y}_i\}_{i=0}^{N}$, which serves as the basis for a (vector) piecewise polynomial, $\mathbf{S}(x)$, that is based on a continuous MIRK (CMIRK) formula [Muir and Owren 1993]. On the subinterval, $[x_i, x_{i+1}]$, $\mathbf{S}(x)$ takes the form

$$\mathbf{S}(x_i + \theta h_{i+1}) = \mathbf{y}_i + h_{i+1} \sum_{j=1}^{s^*} b_j(\theta) \mathbf{f}(x_i + c_j h_{i+1}, \mathbf{y}_{ij}),$$

where $0 \leq \theta \leq 1$ and $s^* \geq s$. In the above equation, each $b_j(\theta)$ is a known polynomial in θ, defined by the CMIRK method. Because $s^* \geq s$, it follows that $\mathbf{S}(x)$ may need to use extra stages; each such stage has the same general form as in (3). The piecewise polynomial, $\mathbf{S}(x)$, is a C^1 continuous approximation to the exact solution to the BVODE, $\mathbf{y}(x)$. We note that the CMIRK scheme is constructed, i.e., the coefficients and weight polynomials, $b_r(\theta)$, of the scheme are chosen, so that on the subinterval $[x_i, x_{i+1}]$, we have

$$\mathbf{S}(x_i) = \mathbf{y}_i, \quad \mathbf{S}(x_{i+1}) = \mathbf{y}_{i+1}, \quad \mathbf{S}'(x_i) = \mathbf{f}(x_i, \mathbf{y}_i), \quad \mathbf{S}'(x_{i+1}) = \mathbf{f}(x_{i+1}, \mathbf{y}_{i+1}),$$

(to within the Newton tolerance) and these conditions imply C^1 continuity (to within the Newton tolerance).

On each subinterval, `BVP_SOLVER` computes a scaled defect, $\boldsymbol{\delta}(x)$, of the approximate solution at several points on each subinterval. The jth component of $\boldsymbol{\delta}(x)$ is

$$\delta_j(x) = \frac{|S_j'(x) - f_j(x, \mathbf{S}(x))|}{1 + |f_j(x, \mathbf{S}(x))|}, \tag{6}$$

where $S_j'(x)$ is the derivative of the jth component of the vector function $\mathbf{S}(x)$ and $f_j(x, \mathbf{S}(x))$ is the jth component of the vector function $\mathbf{f}(x, \mathbf{S}(x))$. The maximum of these scaled defect samples is taken to be an estimate of the maximum scaled defect (MSD) for the subinterval. If the estimated MSD is greater than the user prescribed tolerance on any subinterval, the current solution approximation is rejected and estimates of the MSD on each subinterval are then used to guide a process that attempts to construct a new mesh such that (i) the MSD estimates are approximately equidistributed over the subintervals of the new mesh, and (ii) the MSD estimate on each subinterval of the new mesh is less than the user tolerance. Achieving such a mesh typically involves changing the total number of mesh points and redistributing them over the problem domain. Once a new mesh is obtained, a new continuous solution approximation is computed and the defect sampling process is repeated. If the estimated MSD for each subinterval is less than the user tolerance, the solution is accepted.

The current version of `BVP_SOLVER` simply samples the defect at two points on each subinterval; a more robust estimate of the MSD on each subinterval can be obtained at a modest increase in cost using an approach called *asymptotically correct defect control* [Enright and Muir 2010]. The approach relies on the development of a new type of interpolant for the continuous solution approximation on each subinterval (building upon the CMIRK interpolants mentioned earlier) for which the maximum defect is (asymptotically) guaranteed to occur at a known, problem independent location within each subinterval.

Although `BVP_SOLVER` does not attempt to directly control the GE, it does provide, as mentioned earlier, an option for the computation of an a posteriori estimate

of the GE based on RE, which we now briefly describe (see also, e.g., Section 5.5.2 of [Ascher et al. 1995]). Let π be the final mesh upon which the accepted, defect controlled numerical solution is obtained. Let $\mathbf{Y}_\pi$ be the numerical solution evaluated at the points of the mesh π. Let $\mathbf{Y}_\pi^{(i,j)}$ be the jth component of the numerical solution evaluated at the ith mesh point of π. Let π_2 be a new mesh is obtained by halving each subinterval of π. In the RE scheme, a second discrete solution is computed on this new mesh, using the same MIRK scheme that was used to obtain $\mathbf{Y}_\pi$. Define $\mathbf{Y}_{\pi_2}$ to be this second solution *evaluated only at the points of* π. Let $\mathbf{Y}_{\pi_2}^{(i,j)}$ be the jth component of this second numerical solution at the ith mesh point of π. The GE estimate by RE is then given by

$$\frac{2^p}{2^p-1}\max_{i,j}\frac{|\mathbf{Y}_\pi^{(i,j)}-\mathbf{Y}_{\pi_2}^{(i,j)}|}{1+|\mathbf{Y}_\pi^{(i,j)}|}, \tag{7}$$

where p is the order of the MIRK method used to compute these approximate solutions.

The computation of $\mathbf{Y}_{\pi_2}$ requires the setup and solution via Newton's method of a nonlinear system similar in form to (4) but with approximately twice as many nonlinear equations and unknowns. In `BVP_SOLVER` this nonlinear system is solved using the same modified Newton iteration as described for the computation of the primary solution, using a Newton tolerance that is half the size. Because at least one Jacobian matrix must be evaluated and factored, this scheme can be quite computationally expensive. The initial estimate of the solution provided to the Newton iteration for the determination of $\mathbf{Y}_{\pi_2}$ is obtained from the evaluation of the continuous numerical solution, $\mathbf{S}(x)$, at the points of the mesh π_2.

4. ALTERNATIVE GE ESTIMATORS FOR `BVP_SOLVER`

We now discuss the practical implementation of several GE estimation techniques within `BVP_SOLVER`. All assume that the defect controlled numerical solution has been accepted by the solver. Subsection 4.1 describes a GE estimation technique based on the direct use of higher order MIRK formulas. Subsection 4.2 describes an approach based on the use of higher order MIRK formulas within a deferred correction framework. Subsection 4.3 examines defect control from a backward error analysis viewpoint and describes a GE bound based on the norm of the defect and an estimate of a conditioning constant for the BVODE. Subsection 4.4 discusses an improved implementation of the RE algorithm described in the previous section.

Except for the approach based on the conditioning constant, the other approaches mentioned above are all examples of well known techniques for the estimation of the GE. The paper [Russell and Christiansen 1978] considers mesh adaptation based on a number of error estimation schemes and looks at relationships between them; the focus is on collocation methods for the discretization but the authors indicate that the conclusions of the paper may be relevant to other approaches. The paper [Wright et al. 1994] describes an improved error estimation scheme for COLSYS. More recent work on error estimation for BVODEs includes the paper [Moore 2001], in which an interpolation based approach for obtaining a posteriori error estimates for the finite element solution of BVODEs is developed. Another recent effort is the paper [Koch 2005], in which an asymptotically correct error estimate for the collo-

cation solution of a BVODE involving a singularity is developed based on a defect correction approach. The papers [Cash et al. 2006] and [Cash and Mazzia 2006] discuss an approach for the control of mesh adaption in the numerical solution of BVODEs that uses a hybrid global error/conditioning constant estimation scheme; the papers also discuss efficient algorithms for the computation of a conditioning constant estimate. The paper [Wright 2007] describes a numerical investigation of mesh refinement based on several error estimation criteria for the collocation solution of BVODEs.

4.1 Direct Use of a Higher Order MIRK Formula for GE Estimation

Assuming that the primary solution is obtained using a MIRK method of order p on some final mesh, we obtain a second numerical solution of order $p+2$ on the same mesh by constructing and solving a nonlinear system of the form (4) using a MIRK method of order $p+2$. We choose a method 2 orders higher (rather than only 1) because it is important to employ symmetric Runge–Kutta methods when solving a BVODE, and such symmetric methods have only even orders; see, e.g., [Muir 1999]. This computation yields only a discrete solution approximation.

`BVP_SOLVER` can solve BVODEs using a second, fourth, or sixth order MIRK method; see [Muir 1999] for the tableaux that define these formulas and their associated interpolants. Thus for primary solutions obtained using a second or fourth order MIRK formula, there is a natural MIRK formula available for the computation of the higher order numerical solution. For the case when the primary solution is obtained by using the sixth order MIRK formula, we have added an eighth order MIRK method [Gupta 1985] to `BVP_SOLVER`. This MIRK formula contains embedded formulas of orders 2, 4, and 6, but if the embedded sixth order formula is not required, stage five can be ignored. We do not make use of the embedded formulas, and thus we implement this formula as a nine stage method. We also choose the free parameter β of this eighth order formula to be 0, which is a reasonable approximation to the optimal value $\beta \approx 0.006970$ that we have computed that minimizes the 2-norm of the principal error function (see, e.g., [Muir 1999]) for this eighth order method. (The corresponding values for the norms of the ninth and tenth order principal error coefficients are 6.9950×10^{-6} and 6.9751×10^{-6}, respectively.)

Let $\mathbf{Y}_p$ be the primary solution of order p, evaluated at the mesh points of the final mesh and let $\mathbf{Y}_{p+2}$ be the solution of order $p+2$, evaluated at the same set of points. Let $\mathbf{Y}_p^{(i,j)}$ be the jth component of $\mathbf{Y}_p$ at the ith mesh point and let $\mathbf{Y}_{p+2}^{(i,j)}$ be the corresponding component of $\mathbf{Y}_{p+2}$. Then the estimate of the GE for $\mathbf{Y}_p$ in this case is

$$\max_{i,j} \frac{|\mathbf{Y}_p^{(i,j)} - \mathbf{Y}_{p+2}^{(i,j)}|}{1 + |\mathbf{Y}_p^{(i,j)}|}. \tag{8}$$

When implementing this scheme, several observations were exploited in order to obtain substantial savings in computation time.

(1) From our numerical experiments, we have observed that the primary solution, $\mathbf{Y}_p$, proves to be an effective initial guess for the solution of the system of nonlinear equations based on the higher order MIRK formula. Because $\mathbf{Y}_p$ is saved in the solution structure employed by `BVP_SOLVER`, it is available for use

as the initial guess at no additional computational cost.

(2) From our numerical experiments, we have observed that the Jacobian matrix from the primary solution computation is a good approximation for the Jacobian associated with the nonlinear system based on the higher order MIRK formula. This matrix is also available within one of the arrays used by `BVP_SOLVER` during the computation of the primary solution. We are therefore able to avoid the expensive evaluation and factorization of this matrix during the Newton iteration for the determination of $\mathbf{Y}_{p+2}$.

(3) We initially employed the same form of Newton iteration for the determination of $\mathbf{Y}_{p+2}$ that is used in the computation of the primary solution; this meant that full and damped Newton steps were allowed, and the iteration terminated when an appropriately scaled norm of the Newton correction was less than the user tolerance. However, we have experimented with a simpler version of this scheme in which only one Newton correction is performed. Our experiments showed that the resultant estimates were sufficiently accurate for our purposes. Accordingly, we perform only one Newton correction for the computation of $\mathbf{Y}_{p+2}$.

By making use of the (factored) Jacobian from the primary computation and by employing only one Newton iteration, the implementation of this GE estimate involves only one backward substitution, based on one evaluation of $\mathbf{\Phi}$ in (4).

4.2 GE Estimation based on Deferred Correction

When the MIRK method upon which (4) is based is of order p, we rewrite (4) as

$$\mathbf{\Phi}_p(\mathbf{Y}_p) = \mathbf{0},$$

where the pth order discrete solution, obtained by solving this system, is $\mathbf{Y}_p$. In [Cash and Wright 1991], the authors describe a deferred correction method based on MIRK formulas. `BVP_SOLVER` uses MIRK formulas and thus we can use an approach similar to that of [Cash and Wright 1991]; the deferred correction equation that allows us to obtain a higher order solution, $\mathbf{Y}_{p+2}$, is

$$\mathbf{\Phi}_p(\mathbf{Y}_{p+2}) = -\mathbf{\Phi}_{p+2}(\mathbf{Y}_p).$$

That is, we need to solve the nonlinear system

$$\mathbf{\Phi}_p(\mathbf{z}) + \mathbf{\Phi}_{p+2}(\mathbf{Y}_p) = \mathbf{0}, \tag{9}$$

for $\mathbf{z} = \mathbf{Y}_{p+2}$. The primary expense is the construction and factorization of the Jacobian matrix of this nonlinear system. However, the system

$$\mathbf{\Phi}_p(\mathbf{z}) = \mathbf{0}, \tag{10}$$

is the one that was just solved during the primary computation to get $\mathbf{Y}_p$. The corresponding Jacobian (evaluated at $\mathbf{Y}_p$ or an approximation to it) was computed and factored for use in that iteration; a significant advantage of employing (9) to determine $\mathbf{Y}_{p+2}$ is that it has the same Jacobian matrix as (10), and thus the Jacobian and its factorization are available at no cost. Furthermore, a natural initial guess for $\mathbf{Y}_{p+2}$ to start the Newton iteration for (9) is $\mathbf{Y}_p$. As in the approach

described in Section 4.1, we also apply only one Newton step to obtain an estimate of $\mathbf{Y}_{p+2}$.

Once $\mathbf{Y}_{p+2}$ is available, the estimate of the norm of the GE for $\mathbf{Y}_p$ is computed as in (8). The computational costs incurred in this approach involve the computation of the correction term, $\mathbf{\Phi}_{p+2}(\mathbf{Y}_p)$, one evaluation of $\mathbf{\Phi}_p(\mathbf{z})$, and one backward substitution associated with applying Newton's method to (9).

4.3 A GE Bound based on estimating the BVODE Conditioning Constant

The third GE estimation approach we consider is based on a backward error analysis for the numerical solution of a BVODE. Here we briefly review the main points; see, e.g., [Shampine and Muir 2004] for further details.

We consider a linear BVODE and assume that the exact solution, $\mathbf{y}(x)$, satisfies

$$\mathbf{y}'(x) = \mathbf{A}(x)\mathbf{y}(x) + \mathbf{q}(x), \quad \mathbf{B}_a\mathbf{y}(a) + \mathbf{B}_b\mathbf{y}(b) = \boldsymbol{\beta}. \tag{11}$$

In (11), $\mathbf{A}(x),\ \mathbf{B}_a,\ \mathbf{B}_b \in R^{n\times n}$ and $\mathbf{q}(x),\ \mathbf{y}(x),\ \boldsymbol{\beta} \in R^n$. Then the approximate solution, $\mathbf{S}(x)$, exactly satisfies the perturbed BVODE and BCs

$$\mathbf{S}'(x) = \mathbf{A}(x)\mathbf{S}(x) + \mathbf{q}(x) + \boldsymbol{\delta}(x), \quad \mathbf{B}_a\mathbf{S}(a) + \mathbf{B}_b\mathbf{S}(b) = \boldsymbol{\beta} + \boldsymbol{\sigma},$$

where $\boldsymbol{\delta}(x) = \mathbf{S}'(x) - \mathbf{A}(x)\mathbf{S}(x) - \mathbf{q}(x)$ and $\boldsymbol{\sigma} = \mathbf{B}_a\mathbf{S}(a) + \mathbf{B}_b\mathbf{S}(b) - \boldsymbol{\beta}$ are the defects associated with the ODE and the BCs, respectively.

The main result is that

$$||\mathbf{y}(x) - \mathbf{S}(x)||_{\mathbf{W}_3} \leq \kappa \max\left(||\boldsymbol{\delta}(x)||_{\mathbf{W}_1}, ||\boldsymbol{\sigma}||_{\mathbf{W}_2}\right), \tag{12}$$

where κ is a conditioning constant for the BVODE and the weighted norms are defined as follows:

$$||\boldsymbol{\delta}(x)||_{\mathbf{W}_1} = \max_{a\leq x\leq b} ||\mathbf{W}_1^{-1}(x)\boldsymbol{\delta}(x)||_\infty, \quad ||\boldsymbol{\sigma}||_{\mathbf{W}_2} = ||\mathbf{W}_2^{-1}\boldsymbol{\sigma}||_\infty,$$

$$||\mathbf{S}(x) - \mathbf{y}(x)||_{\mathbf{W}_3} = \max_{a\leq x\leq b} ||\mathbf{W}_3^{-1}(x)\left(\mathbf{S}(x) - \mathbf{y}(x)\right)||_\infty.$$

Here $\mathbf{W}_1(x)$, $\mathbf{W}_2$, and $\mathbf{W}_3(x)$ are $n\times n$ diagonal matrices with positive entries. The matrix $\mathbf{W}_3(x)$ is associated with the scaling of the defect (6); the matrix $\mathbf{W}_1(x)$ is associated with the scaling for the GE (8). Because we do not scale the BCs, $\mathbf{W}_2$ is taken to be the identity matrix.

The conditioning constant, κ, depends on the fundamental solution of the corresponding homogeneous BVODE and the boundary condition matrices $\mathbf{B}_a$ and $\mathbf{B}_b$. (The conditioning constant is given by $\kappa = \max(\kappa_1, \kappa_2)$, where κ_1 is the conditioning constant for the BCs and κ_2 is the conditioning constant for the ODEs. However, in `BVP_SOLVER`, the BCs are solved much more accurately than the ODEs; so in fact only κ_2 is relevant in the present context.)

The paper [Shampine and Muir 2004] also explains how to compute an estimate of κ. Let

$$\overline{\mathbf{W}}_{12} = \mathrm{diag}\{\mathbf{W}_1(x_1), \ldots, \mathbf{W}_1(x_N), \mathbf{W}_2\}, \quad \overline{\mathbf{W}}_3 = \mathrm{diag}\{\mathbf{W}_3(x_0), \ldots, \mathbf{W}_3(x_N)\}.$$

Then in [Shampine and Muir 2004], it is shown that, for a sufficiently fine mesh,

$$\kappa \approx ||\overline{\mathbf{W}}_3^{-1}\mathbf{J}_{\mathbf{\Phi}}^{-1}\overline{\mathbf{W}}_{12}||_\infty,$$

where $\mathbf{J}_{\mathbf{\Phi}}$ is the Jacobian matrix (5). The paper [Shampine and Muir 2004] suggests the use of the Higham–Tisseur algorithm [Higham 1988] for the efficient estimation of this norm. Because the factored Newton matrix from the primary solution computation is available, once the primary solution is accepted, the Higham–Tisseur algorithm can be used to obtain the estimate for κ using only a few additional back solves involving the matrix $\overline{\mathbf{W}}_{12}^{-1}\mathbf{J}_{\mathbf{\Phi}}\overline{\mathbf{W}}_3$. (The right hand sides involved in these backward substitutions are generated by the software based on the Higham–Tisseur algorithm and represent no significant computational cost.)

We have modified `BVP_SOLVER` to provide an option to efficiently estimate κ after the primary solution is accepted. The product of κ and the estimate of the maximum norm of the defect is then used to obtain an upper bound on the GE as in (12). However, it is worth noting that, especially for a defect control solver, it may be useful to estimate and return κ itself because this quantity gives a measure of the sensitivity of the solution to small changes in the problem.

4.4 Modification of the Implementation of the RE based GE Estimate

Based on numerical experiments, we found that the treating the nonlinear system associated with the RE approach by performing exactly one full Newton step yielded error estimates that compared well with those obtained from a full tolerance controlled Newton iteration. We therefore employ a more efficient implementation of the RE based GE estimate that computes and factors a new Jacobian matrix and then performs only one full Newton step.

5. COMPARISON OF THE GE ESTIMATES

With the implementations described in the previous section, `BVP_SOLVER` now has four possible methods for estimating the GE. In this section, we discuss accuracy and computational efficiency results for these four estimators. These results are selected from a more comprehensive study available in [Boisvert et al. 2012]. All GE estimates are for the scaled norms specified earlier. We consider three test problems and examine the performance of the GE estimators for the three MIRK formula order options (2, 4, and 6) and for the range of tolerance values $10^{-4}, 10^{-5}, \ldots, 10^{-8}$. All test problems were converted to first order systems as required by `BVP_SOLVER`.

Each problem is solved a number of times in succession in order to obtain cumulative timings that are large enough to be reliable, i.e., on the order of (at least) 10 seconds. Each problem also depends on a positive parameter ϵ, where the problem difficulty increases as ϵ decreases. Values of ϵ are chosen according to those suggested by their sources in the literature unless this led to excessively large solution times, in which case the value of ϵ was increased slightly. Consequently, the number of timing runs varies according to the problem solved and the order of the discretization. The minimum time from three cumulative timing runs is reported.

The computations were performed using an Intel Xeon w3520 quad core processor running at 2.667 GHz. The RAM consisted of 16GB of DDR3 memory running at 1.333 GHz. The operating system was 64 bit Ubuntu 10.04.2 LTS with kernel 2.6.32-32-generic and the `Fortran` compiler was gfortran with gcc 4.4.3-4ubuntu5.

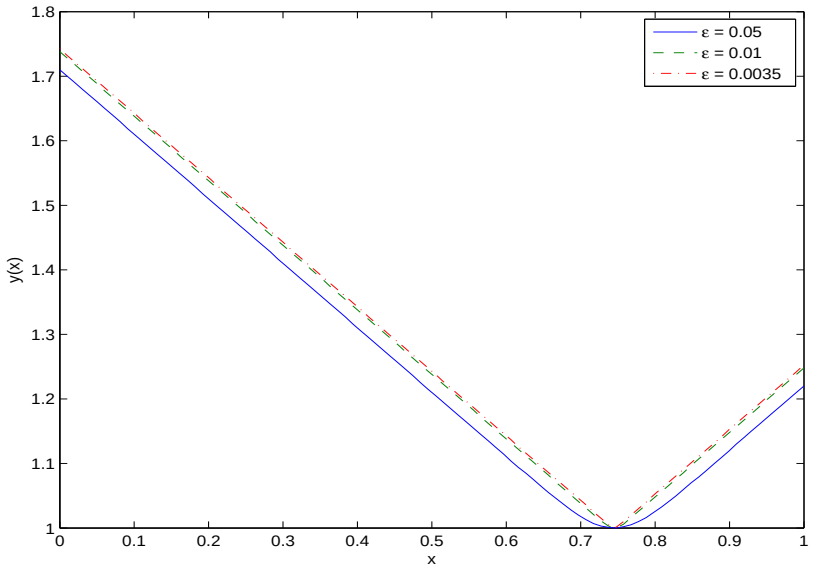

Fig. 1. Solution $y(x)$ of problem (13) for $\epsilon = 0.05$, 0.01, and 0.0035.

5.1 Test Problems

(1) The first problem is

$$\epsilon y'' + (y')^2 = 1, \tag{13a}$$

with BCs

$$y(0) = 1 + \epsilon \ln\cosh\left(\frac{-0.745}{\epsilon}\right), \quad y(1) = 1 + \epsilon \ln\cosh\left(\frac{0.255}{\epsilon}\right), \tag{13b}$$

and exact solution

$$y(x) = 1 + \epsilon \ln\cosh\left(\frac{x - 0.745}{\epsilon}\right).$$

This is Problem 20 from `www.ma.ic.ac.uk/~jcash/BVP_software`; see also [Cash and Mazzia 2006]. For MIRK order 2, we use $\epsilon = 0.05$. For MIRK orders 4 and 6, we use $\epsilon = 0.0035$; in Section 6 we also use $\epsilon = 0.01$. The solutions $y(x)$ for these values of ϵ are displayed in Figure 1. We use an initial guess of $y(x) \equiv \frac{1}{2}$, $y'(x) \equiv 0$. Timing results are the average of 500 runs.

(2) The second problem is

$$\epsilon y'' = y + y^2 - \exp\left(\frac{-2x}{\sqrt{x}}\right), \tag{14a}$$

with BCs,

$$y(0) = 1, \quad y(1) = \exp\left(\frac{-1}{\sqrt{\epsilon}}\right), \tag{14b}$$

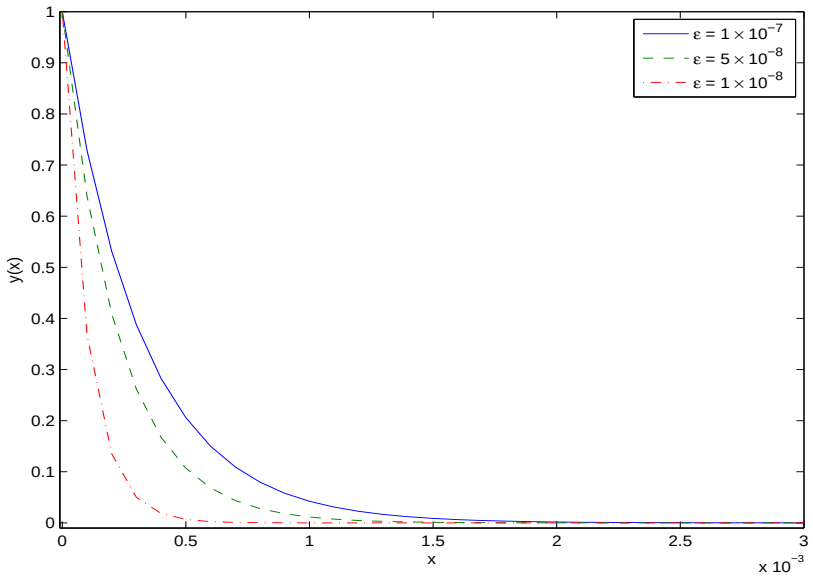

Fig. 2. Solution $y(x)$ of problem (14) for $\epsilon = 1 \times 10^{-7}$, 5×10^{-8}, and 1×10^{-8}.

and exact solution

$$y(x) = \exp\left(\frac{-x}{\sqrt{\epsilon}}\right).$$

This is Problem 21 from `www.ma.ic.ac.uk/~jcash/BVP_software`. For MIRK order 2, we use $\epsilon = 10^{-7}$. For MIRK order 4, we use $\epsilon = 5 \times 10^{-8}$. For MIRK order 6, we use $\epsilon = 10^{-8}$. The solutions $y(x)$ for these values of ϵ are displayed in Figure 2. (Note that, in order to make visible the differences between the solutions associated with different ϵ values, the horizontal axis in Figure 2 includes only the region $[0, 0.003]$.) We use an initial guess of $y(x) \equiv \frac{1}{2}$, $y'(x) \equiv 0$. Timing results are the average of 100 runs.

(3) The third problem is

$$\epsilon f'''' + f f''' + g g' = 0, \quad \epsilon g'' + f g' - f' g = 0, \tag{15a}$$

with BCs,

$$f(0) = f(1) = f'(0) = f'(1) = 0, \; g(0) = \Omega_0, \; g(1) = \Omega_1, \tag{15b}$$

where $\Omega_0 = -1$, $\Omega_1 = 1$, and $\epsilon = 9 \times 10^{-5}$; in Section 6 we also use $\epsilon = 5 \times 10^{-3}$. This is Example 1.20 of [Ascher et al. 1995]. Because no exact solution for this problem is known, a reference solution was computed by `BVP_SOLVER` using the sixth order MIRK method with a tolerance of 10^{-11}. The solutions $f(x)$ and $g(x)$ for these values of ϵ are displayed in Figure 3. We use an initial guess of $g(x) = 2x - 1$, $g'(x) = 2$, and $f(x) \equiv f'(x) \equiv f''(x) \equiv f'''(x) \equiv 0$. Timing results are the average of 3000 runs.

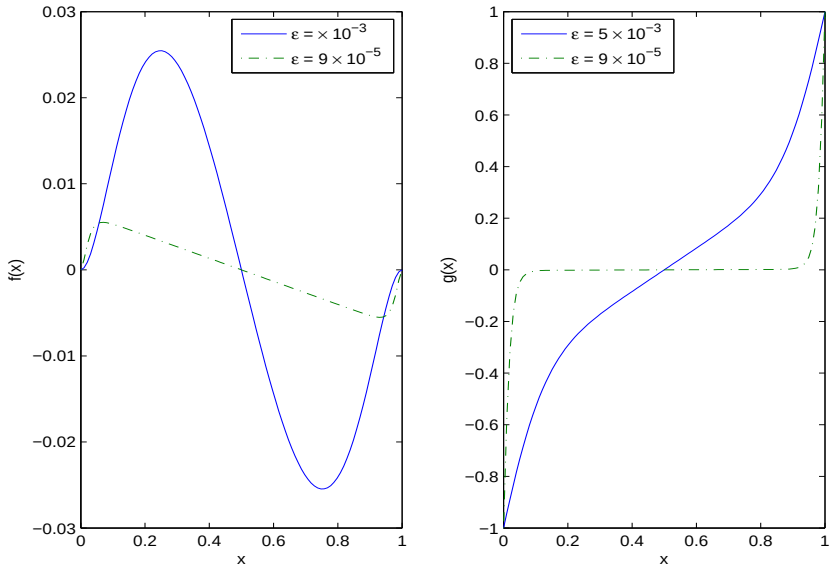

Fig. 3. Solutions $f(x)$ and $g(x)$ of problem (15) for $\epsilon = 5 \times 10^{-3}$ and 9×10^{-5}.

5.2 Results for Second Order

There is excellent agreement between the true GE and the estimated GE from RE, the approach based on the use of a higher order method (HO), and the approach based on deferred correction (DC); see Tables 1–3 in [Boisvert et al. 2012]. However, all results from the use of the conditioning constant estimate (CO) give a substantial overestimate of the GE, typically by several orders of magnitude. This behavior is in general to be expected because CO is not derived as a sharp bound on the GE. The results for CO are included mainly for completeness as an existing alternative GE estimator.

We next investigate the relative efficiency of the estimators by considering plots of execution time of each estimator (relative to the time required to compute the primary solution) vs. the tolerance. Typically, the execution time for the RE estimator is a much higher percentage of the primary solution computation time than the other estimators. Figure 4 shows results for problem (13); the relative costs of the RE estimator are approximately between 23% and 40% for all tolerances considered. The relative costs of the other estimators are less than approximately 10%. Similar results are obtained for test problem (14). The relative costs of the RE estimator are approximately between 10% and 15%; the relative costs of all other estimators are less than approximately 5%; see [Boisvert et al. 2012]. For problem (15) (see [Boisvert et al. 2012]) we see slightly different results: the relative cost of the RE estimator increases as the tolerance becomes sharper. The relative cost of the RE estimator is approximately 4% for tolerance 10^{-4} and steadily increases to approximately 27% as the tolerance approaches 10^{-8}. The relative costs of the HO

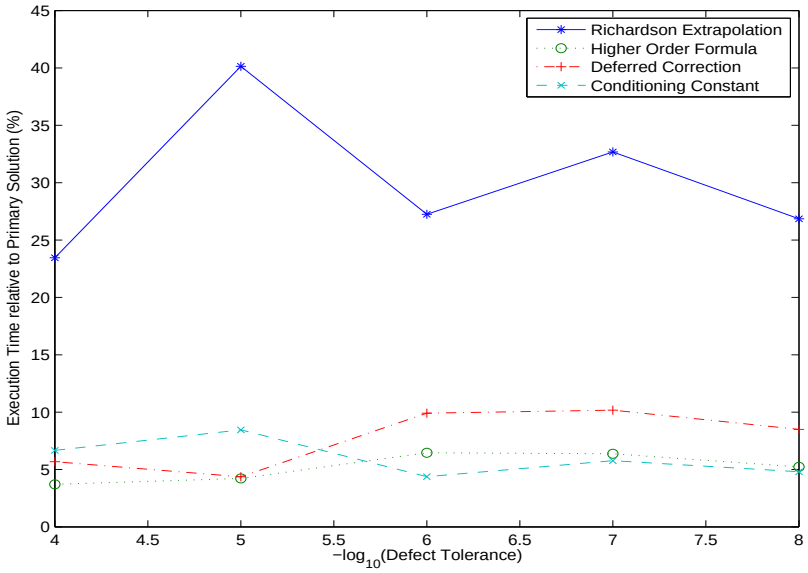

Fig. 4. Relative execution time of GE estimators vs. $-\log_{10}$ of defect tolerance with second order MIRK formula for test problem (13).

and DC estimators increase from approximately 1% to 5%. The relative cost of the CO estimator approximately increases from 2% to 15%.

5.3 Results for Fourth Order

Again there is excellent agreement between the true GE and the estimated GE from the RE, HO, and DC approaches; see Tables 4–6 in [Boisvert et al. 2012]. For all cases, the CO approach gives a significant overestimate of the GE.

Figure 5 shows timing results for test problem (14). The cost of the RE estimator is significantly larger than those of the other error estimators. The results for test problem (13) show that none of the GE estimators have significant costs; see [Boisvert et al. 2012]. For test problem (15), the relative cost of the RE estimator steadily increases (approximately between 8% and 21%) as the tolerance becomes sharper whereas the relative costs of the other error estimators are consistently less than about 5%; see [Boisvert et al. 2012].

5.4 Results for Sixth Order

Again there is excellent agreement between the true GE and the estimated GE from the RE, HO, and DC approaches; see Tables 7–9 in [Boisvert et al. 2012]. As in the previous cases, the CO estimates are several orders of magnitude too large.

For problem (13), the relative costs for all the estimators are small (approximately between 1% and 3%); see [Boisvert et al. 2012]. For problem (14), the relative costs of the RE estimator are larger (approximately between 8% and 17%). The relative

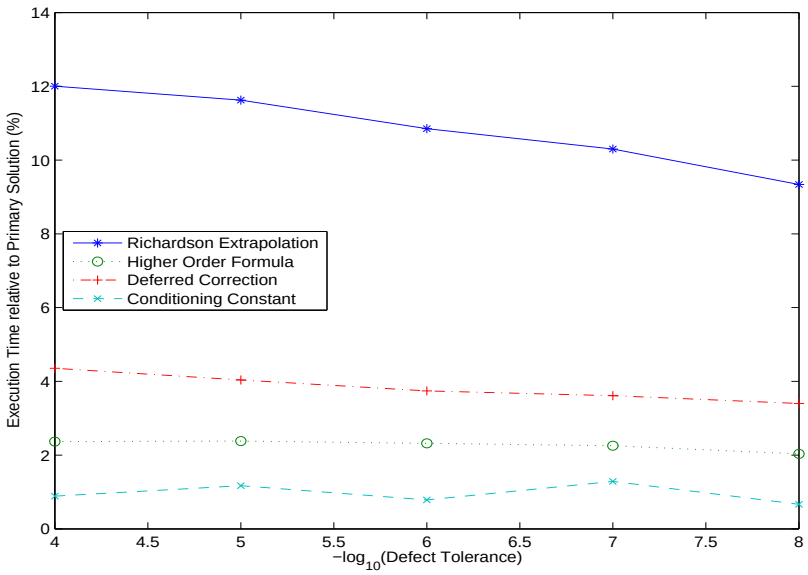

Fig. 5. Relative execution time of global error estimators vs. $-\log_{10}$ of defect tolerance with fourth order MIRK formula for test problem (14).

costs of the other error estimators are less than 6%; see [Boisvert et al. 2012]. Figure 6 gives results for problem (15); the relative cost of RE increases from approximately 8% to 26% as the tolerance becomes sharper. The relative costs of the HO, DC, and CO estimators are less than 4%.

6. BVP_SOLVER WITH GLOBAL ERROR CONTROL

We have developed a new version of BVP_SOLVER that provides an option for the computation of a *GE controlled* numerical solution. The new version of the code performs the same basic computation to obtain a discrete numerical solution at the mesh points as does the original. Once this discrete numerical solution is obtained, the new version of the solver can then compute an estimate of the (scaled) GE (as in (8)) of that solution using one of the GE estimation algorithms analyzed in the previous section. We have modified BVP_SOLVER so that it is able to compute an estimate of the GE for the discrete numerical solution obtained on each intermediate mesh rather than only at the end of the computation, as considered in the previous section. In this new version of the code, if the estimate of the GE does not satisfy the tolerance, the GE estimates for each subinterval are passed to the mesh adaptation algorithm, where they are used to construct a new mesh.

The mesh adaptation algorithm is identical to what is used in the defect control case except for one parameter setting, which we now describe. Two important quantities that are computed in the BVP_SOLVER mesh adaptation routine are related to the maximum GE or defect over all subintervals and the average GE or defect

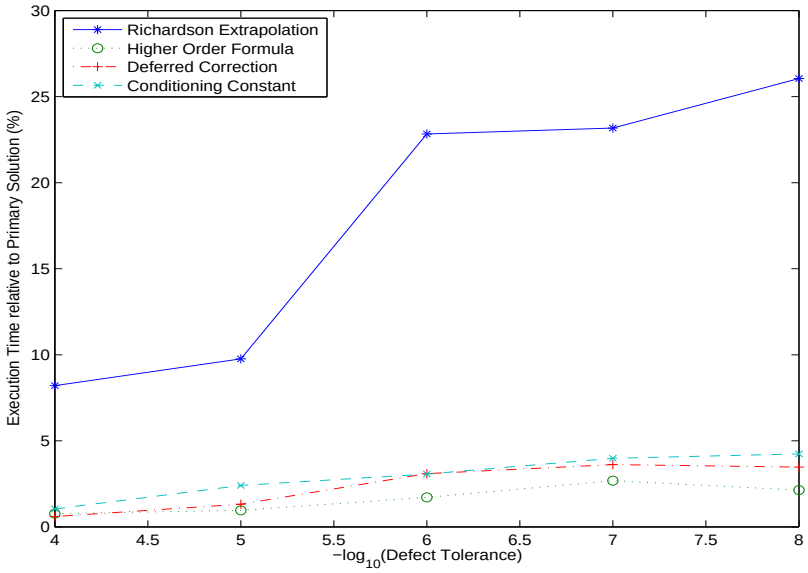

Fig. 6. Relative execution time of global error estimators vs. $-\log_{10}$ of defect tolerance with sixth order MIRK formula for test problem (15).

over all subintervals. The ratio of the former quantity to the latter is computed and compared to a parameter called ρ. If the ratio is greater than or equal to ρ, a new mesh is constructed based on equidistribution of the GE or defect. Otherwise, a new mesh is constructed by halving each subinterval of the current mesh. In the original version of `BVP_SOLVER`, $\rho = 1$, and this forces an equidistribution process for the construction of *every* new mesh. When we tried to use $\rho = 1$ for the new GE controlled version of `BVP_SOLVER`, we found that it was impossible to obtain a solution for any of the test problems, even using meshes with millions of points. It was necessary to use a larger value of ρ (we chose $\rho = 2$, a common choice in the literature) in order to force an occasional global refinement (via mesh halving) of the mesh. This was necessary to reduce the size of non-local contributions to the GE on each subinterval; see [Enright and Muir 1996] for further details regarding the mesh adaptation algorithm employed in `BVP_SOLVER` and [Ascher et al. 1995] for further discussion of mesh adaptation based on GE estimates.

6.1 `BVP_SOLVER` in GE Control Mode

In this section we present some results that represent a preliminary investigation of the use of the GE control mode in the new version of `BVP_SOLVER`. We have considered numerical experiments employing the test problems (13) with $\epsilon = 0.01$ (see Figure 1) and (15) with $\epsilon = 0.005$ (see Figure 3), for all three orders and a range of tolerances. Here we present selected results from these experiments; the full set of results is available in [Boisvert et al. 2012]. In each table presented here, we

report, for each tolerance, the required CPU time, the number of points used in the final mesh (N), the estimated and true maximum GE, and the estimated and true maximum defect. (The true maximum GE and defect were obtained by sampling them at 10 points per subinterval.) The DC algorithm of the previous section was used to estimate the GE. `BVP_SOLVER` is run in each of four control modes: defect control (DefC), GE control (GEC), and sequential and parallel combinations of defect and GE control, which are now described.

—In the sequential combination control (SCC) mode, a defect controlled solution is computed, and it and its corresponding mesh are passed as the initial data to a GE controlled computation. When the BVODE is poorly conditioned, the conditioning constant is large and (from (12)) we can expect the GE to be larger than the defect. It is thus easier to compute a defect controlled numerical solution than a GE controlled numerical solution, potentially making it more efficient in such cases to first compute a defect controlled solution rather than directly compute a GE controlled numerical solution. When the defect controlled solution is passed to BVP_SOLVER in GE control mode, the solver first estimates the GE of that solution. If this estimate does not satisfy the tolerance, then a new mesh is constructed based on the GE estimate and the GE control mode computation begins.

—In the parallel combination control (PCC) mode, both the (scaled) defect (6) and the (scaled) GE estimate are obtained for each numerical solution, and then a linear combination of the two is used as input to the mesh refinement process and the termination criterion. For the purposes of this preliminary investigation, we consider only the simple sum of the scaled defect and the scaled GE estimate. *In this case, the resultant numerical solution has a defect estimate and a GE estimate that satisfy the user tolerance, and the numerical solution can also be said to be the exact solution to a BVODE that is a perturbation (on the order of the tolerance) of the original BVODE.* As mentioned, however, the PCC mode accepts a general linear combination of the scaled defect and GE estimates. This freedom allows the user to control the relative importance of one tolerance over the other; i.e., the scaled defect estimate could be weighted more heavily than the scaled GE estimate or vice versa. The question of how to choose this weighting in practice is likely to be highly problem-dependent; hence a treatment of this question is beyond the scope of this paper.

In Table I we report results for test problem (13) with $\epsilon = 0.01$ and using MIRK formula order 4. The timing results, in seconds, are the average of 3000 runs. For each run, the minimum time of three cumulative runs is used in order to reduce the interference from other computational processes. The cost of computing a defect controlled solution is greater than the cost of computing a GE controlled solution for coarser tolerances and only slightly smaller for sharper tolerances. The estimated GE for the defect controlled solution is less than the tolerance, and thus in SCC mode the code does no extra adaptation — it simply stops after determining that the estimated GE is less than the tolerance. We observe that the final mesh is the same as the final mesh used in the defect control case. The cost of running in SCC mode is greater than the cost of running in GE control mode for coarser tolerances and only slightly smaller for sharper tolerances. For this test problem, there is thus

no significant advantage to using SCC mode. The costs for PCC mode are slightly higher than those for either defect control mode or GE control mode.

DefC						
Tol	CPU Time (s)	N	Est. Defect	True Defect	Est. GE	True GE
10^{-4}	4.657×10^{-3}	62	4.434×10^{-5}	8.908×10^{-5}	4.381×10^{-6}	4.750×10^{-6}
10^{-5}	6.566×10^{-3}	106	2.385×10^{-6}	5.813×10^{-6}	3.594×10^{-7}	3.930×10^{-7}
10^{-6}	6.927×10^{-3}	191	6.012×10^{-7}	1.208×10^{-6}	2.703×10^{-8}	2.874×10^{-8}
10^{-7}	8.178×10^{-3}	281	9.429×10^{-8}	1.894×10^{-7}	5.358×10^{-9}	5.651×10^{-9}
10^{-8}	9.947×10^{-3}	485	9.264×10^{-9}	1.861×10^{-8}	5.649×10^{-10}	5.831×10^{-10}
GEC						
Tol	CPU Time (s)	N	Est. Defect	True Defect	Est. GE	True GE
10^{-4}	3.660×10^{-3}	47	8.676×10^{-4}	1.648×10^{-3}	4.309×10^{-5}	4.931×10^{-5}
10^{-5}	4.310×10^{-3}	83	9.105×10^{-5}	1.757×10^{-4}	4.237×10^{-6}	4.729×10^{-6}
10^{-6}	5.243×10^{-3}	145	1.266×10^{-5}	2.452×10^{-5}	4.513×10^{-7}	4.857×10^{-7}
10^{-7}	9.898×10^{-3}	303	2.675×10^{-5}	5.374×10^{-5}	5.942×10^{-8}	1.207×10^{-7}
10^{-8}	1.009×10^{-2}	529	2.037×10^{-6}	4.093×10^{-6}	4.668×10^{-9}	9.050×10^{-9}
SCC						
Tol	CPU Time (s)	N	Est. Defect	True Defect	Est. GE	True GE
10^{-4}	5.480×10^{-3}	62	4.434×10^{-5}	8.908×10^{-5}	4.381×10^{-6}	4.750×10^{-6}
10^{-5}	6.699×10^{-3}	106	2.385×10^{-6}	5.813×10^{-6}	3.594×10^{-7}	3.930×10^{-7}
10^{-6}	8.236×10^{-3}	191	6.012×10^{-7}	1.208×10^{-6}	2.703×10^{-8}	2.874×10^{-8}
10^{-7}	9.631×10^{-3}	281	9.429×10^{-8}	1.894×10^{-7}	5.358×10^{-9}	5.651×10^{-9}
10^{-8}	9.997×10^{-3}	485	9.264×10^{-9}	1.861×10^{-8}	5.649×10^{-10}	5.831×10^{-10}
PCC						
Tol	CPU Time (s)	N	Est. Defect	True Defect	Est. GE	True GE
10^{-4}	4.800×10^{-3}	110	2.544×10^{-5}	4.665×10^{-5}	1.916×10^{-6}	2.122×10^{-6}
10^{-5}	5.403×10^{-3}	145	8.356×10^{-6}	1.614×10^{-5}	6.310×10^{-7}	6.685×10^{-7}
10^{-6}	9.759×10^{-3}	235	4.847×10^{-7}	9.737×10^{-7}	5.700×10^{-9}	5.780×10^{-9}
10^{-7}	9.955×10^{-3}	403	4.278×10^{-8}	8.594×10^{-8}	6.297×10^{-10}	6.347×10^{-10}
10^{-8}	1.084×10^{-2}	1047	7.168×10^{-9}	1.440×10^{-8}	1.376×10^{-11}	3.628×10^{-11}

Table I. Test problem (13) with $\epsilon = 0.01$ and MIRK formula order 4.

In Table II we report results for test problem (15) with $\epsilon = 0.005$ and MIRK formula order 2. The number of runs for each tolerance is chosen such that the accumulated time of all runs for a given tolerance is over 10 seconds where the minimum time of three cumulative runs is used for each run. The results show the average time of a run in seconds. In this case, *the cost of computing a defect controlled solution is significantly less than the cost of computing a GE controlled solution.* As well, the estimated GE for the defect controlled solutions is greater than the corresponding tolerance. In the SCC case, substantial additional computation is required to go from the defect controlled solution to the GE controlled solution, and the costs are greater than for direct GE control. On the other hand, except for the sharpest tolerance, *the solution obtained through PCC control costs less than the computation in which only the GE estimate is controlled, suggesting that the inclusion of information about the defect is of some benefit to GE control.*

In Table III we report results for test problem (15) with $\epsilon = 0.005$ and MIRK formula order 6. The timing results are the average of 3000 runs where the minimum time of three cumulative runs is used for each run. The cost of computing

DefC						
Tol	CPU Time (s)	N	Est. Defect	True Defect	Est. GE	True GE
10^{-4}	1.828×10^{-2}	935	6.430×10^{-5}	8.232×10^{-5}	4.303×10^{-4}	4.303×10^{-4}
10^{-5}	4.608×10^{-2}	2621	8.181×10^{-6}	1.069×10^{-5}	5.409×10^{-5}	5.460×10^{-5}
10^{-6}	1.664×10^{-1}	8491	6.788×10^{-7}	8.945×10^{-7}	5.177×10^{-6}	5.188×10^{-6}
10^{-7}	5.794×10^{-1}	27546	6.140×10^{-8}	8.110×10^{-8}	4.922×10^{-7}	4.924×10^{-7}
10^{-8}	1.218	71641	9.405×10^{-9}	1.243×10^{-8}	7.278×10^{-8}	7.280×10^{-8}
GEC						
Tol	CPU Time (s)	N	Est. Defect	True Defect	Est. GE	True GE
10^{-4}	2.086×10^{-1}	9217	1.551×10^{-3}	1.786×10^{-3}	5.354×10^{-5}	5.354×10^{-5}
10^{-5}	8.790×10^{-1}	36865	1.117×10^{-4}	1.417×10^{-4}	3.333×10^{-6}	3.333×10^{-6}
10^{-6}	1.772	73729	2.871×10^{-5}	3.714×10^{-5}	8.333×10^{-7}	8.333×10^{-7}
10^{-7}	7.218	294913	1.832×10^{-6}	2.410×10^{-6}	5.208×10^{-8}	5.208×10^{-8}
10^{-8}	2.912×10^{1}	1179649	1.151×10^{-7}	1.520×10^{-7}	3.255×10^{-9}	3.257×10^{-9}
SCC						
Tol	CPU Time (s)	N	Est. Defect	True Defect	Est. GE	True GE
10^{-4}	3.590×10^{-1}	14697	3.426×10^{-5}	4.423×10^{-5}	2.917×10^{-5}	2.917×10^{-5}
10^{-5}	1.137	46233	2.812×10^{-6}	3.697×10^{-6}	3.074×10^{-6}	3.074×10^{-6}
10^{-6}	3.606	145681	2.777×10^{-7}	3.671×10^{-7}	3.108×10^{-7}	3.108×10^{-7}
10^{-7}	1.145×10^{1}	460041	2.777×10^{-8}	3.671×10^{-8}	3.116×10^{-8}	3.116×10^{-8}
10^{-8}	3.591×10^{1}	1454553	2.774×10^{-9}	3.668×10^{-9}	3.119×10^{-9}	3.122×10^{-9}
PCC						
Tol	CPU Time (s)	N	Est. Defect	True Defect	Est. GE	True GE
10^{-4}	1.017×10^{-1}	4355	4.846×10^{-5}	6.230×10^{-5}	2.990×10^{-5}	2.993×10^{-5}
10^{-5}	5.790×10^{-1}	20275	2.045×10^{-6}	2.688×10^{-6}	1.940×10^{-6}	1.940×10^{-6}
10^{-6}	1.372	61907	6.660×10^{-7}	8.776×10^{-7}	3.249×10^{-7}	3.249×10^{-7}
10^{-7}	6.938	294913	5.342×10^{-8}	7.058×10^{-8}	1.943×10^{-8}	1.943×10^{-8}
10^{-8}	3.236×10^{1}	1007643	1.622×10^{-9}	2.144×10^{-9}	1.547×10^{-9}	1.549×10^{-9}

Table II. Test problem (15) with $\epsilon = 0.005$ and MIRK formula order 2.

a defect controlled solution is significantly less than the cost of computing a GE controlled solution. Except for the sharpest tolerance, the estimated GE for the defect controlled solutions is less than the corresponding tolerance. *In SCC mode, except for the coarsest tolerance, the costs for obtaining a GE controlled solution are less than for the direct GE control case.* The solution obtained through PCC control costs approximately the same as the computation in which only the GE estimate is controlled.

6.2 Use of BVP_SOLVER in GE Control Mode on Problems with Pseudosolutions

In this subsection we briefly consider the application of BVP_SOLVER in GE control mode to a problem that has a pseudosolution. As mentioned earlier, when a problem is poorly conditioned and the user tolerance is coarse, it is possible for a defect control code to compute a numerical solution with a defect whose norm satisfies the user tolerance even if an exact solution does not exist. Such a numerical solution is called a pseudosolution. Although the norm of the defect of this numerical solution is less than the user tolerance, the poor conditioning of the problem implies that the global error is generally large; cf. (12). (Defect control codes can monitor an estimate of the conditioning of the problem or an estimate of the GE to detect this situation [Shampine and Muir 2004].)

DefC						
Tol	CPU Time (s)	N	Est. Defect	True Defect	Est. GE	True GE
10^{-4}	5.000×10^{-5}	16	3.768×10^{-5}	9.267×10^{-5}	8.900×10^{-5}	1.483×10^{-4}
10^{-5}	9.000×10^{-5}	22	5.583×10^{-6}	1.813×10^{-5}	8.158×10^{-6}	2.181×10^{-5}
10^{-6}	2.967×10^{-4}	35	2.651×10^{-7}	9.181×10^{-7}	6.303×10^{-7}	9.015×10^{-7}
10^{-7}	5.933×10^{-4}	49	3.741×10^{-8}	1.283×10^{-7}	9.248×10^{-8}	1.307×10^{-7}
10^{-8}	7.933×10^{-4}	68	7.000×10^{-9}	2.160×10^{-8}	1.306×10^{-8}	1.813×10^{-8}
GEC						
Tol	CPU Time (s)	N	Est. Defect	True Defect	Est. GE	True GE
10^{-4}	8.000×10^{-5}	19	1.475×10^{-5}	5.395×10^{-5}	2.293×10^{-5}	8.404×10^{-5}
10^{-5}	3.233×10^{-4}	37	4.392×10^{-7}	1.734×10^{-6}	1.129×10^{-6}	1.289×10^{-6}
10^{-6}	1.330×10^{-3}	73	1.432×10^{-8}	4.625×10^{-8}	1.755×10^{-8}	2.005×10^{-8}
10^{-7}	1.373×10^{-3}	73	1.432×10^{-8}	4.625×10^{-8}	1.755×10^{-8}	2.005×10^{-8}
10^{-8}	5.024×10^{-3}	145	4.173×10^{-10}	1.049×10^{-9}	2.738×10^{-10}	3.169×10^{-10}
SCC						
Tol	CPU Time (s)	N	Est. Defect	True Defect	Est. GE	True GE
10^{-4}	8.667×10^{-5}	16	3.768×10^{-5}	9.267×10^{-5}	8.900×10^{-5}	1.483×10^{-4}
10^{-5}	1.067×10^{-4}	22	5.583×10^{-6}	1.813×10^{-5}	8.158×10^{-6}	2.181×10^{-5}
10^{-6}	3.933×10^{-4}	35	2.651×10^{-7}	9.181×10^{-7}	6.303×10^{-7}	9.015×10^{-7}
10^{-7}	6.300×10^{-4}	49	3.741×10^{-8}	1.283×10^{-7}	9.248×10^{-8}	1.307×10^{-7}
10^{-8}	3.942×10^{-3}	135	2.003×10^{-10}	4.755×10^{-10}	2.417×10^{-10}	2.833×10^{-10}
PCC						
Tol	CPU Time (s)	N	Est. Defect	True Defect	Est. GE	True GE
10^{-4}	9.000×10^{-5}	19	1.475×10^{-5}	5.395×10^{-5}	2.293×10^{-5}	8.404×10^{-5}
10^{-5}	3.700×10^{-4}	37	4.392×10^{-7}	1.734×10^{-6}	1.129×10^{-6}	1.289×10^{-6}
10^{-6}	1.287×10^{-3}	73	1.432×10^{-8}	4.625×10^{-8}	1.755×10^{-8}	2.005×10^{-8}
10^{-7}	1.423×10^{-3}	73	1.432×10^{-8}	4.625×10^{-8}	1.755×10^{-8}	2.005×10^{-8}
10^{-8}	5.268×10^{-3}	145	4.173×10^{-10}	1.049×10^{-9}	2.738×10^{-10}	3.169×10^{-10}

Table III. Test problem (15) with $\epsilon = 0.005$ and MIRK formula order 6.

The BVODE [Shampine and Muir 2004]

$$y''(x) + |y(x)| = 0, \quad 0 < x < \pi, \quad y(0) = 0, \quad y(\pi) = y_\pi, \tag{16}$$

has a unique solution for $y_\pi < 0$, infinitely many solutions for $y_\pi = 0$, and no solution for $y_\pi > 0$. We first run `BVP_SOLVER` in its original defect control mode and are able to obtain two pseudosolutions when $y_\pi = 0.001$. We obtain one pseudosolution with the second order MIRK method and a second pseudosolution with the fourth order MIRK method. In both cases, a tolerance of 10^{-6} is used and an initial guess of $y(x) = 1.0$ and $y'(x) = 0.0$ for $0 \le x \le \pi$ is provided. For both orders, `BVP_SOLVER` indicates that it finds a solution with a defect norm well below the tolerance. However, if we employ the option within `BVP_SOLVER` to compute an a posteriori GE estimate (using RE) we find that for both MIRK orders the estimated GE is quite large, signalling the presence of a pseudosolution; see Table IV. When we attempt to use `BVP_SOLVER` in defect control mode using the sixth order MIRK method to solve (16) with $y_\pi = 0.001$, the Newton iteration fails to converge and even a defect controlled numerical solution cannot be obtained.

We next tried using `BVP_SOLVER` *in GE control mode* to solve (16) with $y_\pi = 0.001$, using second and fourth order MIRK methods, with a tolerance of 10^{-6}, i.e., the cases that yield pseudosolutions in defect control mode. We found that `BVP_SOLVER` in GE control mode was unable to significantly reduce the GE even using a million

Order	Defect	GE
2	6.23×10^{-7}	5.17
4	7.21×10^{-7}	164.55

Table IV. Result of solving (16) with `BVP_SOLVER` using defect control with an a posteriori GE estimate. Order is the order of the MIRK formula, Defect is the estimated defect max norm, and GE is the estimated GE max norm. We see that the defect is less than the requested tolerance of 10^{-6} but the GE is quite large, signalling the presence of a pseudosolution.

mesh points and thus, appropriately, is not able to obtain a GE controlled numerical solution.

7. CONCLUSIONS AND FUTURE WORK

7.1 Conclusions: Alternative GE Estimators

In this paper we have discussed the efficient implementation of three well known approaches to estimating the GE of the numerical solution of a BVODE within a defect control solver. We have also considered an approach for obtaining a bound on the GE that is based on an estimate of a conditioning constant for the BVODE. The approaches based on the direct use of a higher order method (HO) and on the use of a higher order method within a deferred correction framework (DC) are generally less expensive than the approach based on Richardson extrapolation (RE) while achieving a GE estimate with the same overall quality. From the comments at the ends of Subsections 4.1 and 4.2, we can observe that the DC approach requires two evaluations of $\mathbf{\Phi}(\boldsymbol{Y})$ from (4) whereas the HO approach requires only one. A close inspection of Figures 4–6 shows that the execution times for the HO scheme are generally slightly smaller than those of the DC scheme. We have observed in our experiments that the GE estimate obtained by the DC scheme is only occasionally slightly more accurate than that obtained by the HO scheme and we have not observed that this extra accuracy has led to superior performance. The approach based on the conditioning constant (CO) has a negligible cost but does not have good accuracy. Nonetheless, an estimate of the conditioning constant may be useful for the detection of ill-conditioning for a given BVODE [Shampine and Muir 2004].

We can draw the following conclusions from the results presented in this paper:

(i) The a posteriori GE estimation employed by `BVP_SOLVER` should be based on the HO or DC estimate rather than the RE estimate.

(ii) The CO approach provides a less accurate estimate of the GE because the estimate of the conditioning constant does not provide a tight upper bound. When one employs `BVP_SOLVER` with the option to compute an estimate of the GE, our results suggest that one can then obtain a better estimate of the conditioning constant by using the GE estimate and the defect estimate; i.e., rewriting (12), we get

$$\frac{||\mathbf{y}(x) - \mathbf{S}(x)||_{\mathbf{W}_3}}{\max(||\boldsymbol{\delta}(x)||_{\mathbf{W}_1}, ||\boldsymbol{\sigma}||_{\mathbf{W}_2})} \leq \kappa,$$

giving a lower bound on κ.

(iii) The results presented in this paper may also be relevant for a wider class of general purpose BVODE GE control solvers. In particular, it may be possible to improve the efficiency of the GE estimation approach employed by `COLSYS` or `COLNEW` because these solvers employ RE for one type of GE estimation. It may be worthwhile to investigate the use of the HO or DC approach, with appropriate modifications, within these solvers. It may be possible to obtain a higher order approximate solution using a higher order collocation method applied on the final mesh from the computation of the primary solution.

7.2 Conclusions: GE/Defect Control Mode

Because both the GE and the defect provide valid measures of solution quality, the results presented here suggest that it may be worthwhile to have a BVODE solver that can employ either GE control, defect control, or a hybrid GE/defect control strategy. *In particular, the results indicate that in some cases a hybrid control scheme can yield a GE controlled numerical solution more efficiently than a scheme that controls only the GE.*

It should be noted that in Section 5 the GE estimates are computed only after a defect controlled converged solution has been obtained. However, in Section 6, we provide indirect evidence that the GE estimates are also practical for intermediate approximate solutions that arise before reaching a converged solution; we see that the GE estimates are of sufficiently good quality that they can be successfully used as the basis for a mesh refinement process that eventually leads to a GE controlled converged solution; see the GEC components of Tables I–III.

7.3 Future Work

Implementation of continuous extensions of the discrete higher order solutions computed by the RE, HO, and DC approaches would be useful. This would allow an assessment of the GE of the *continuous* approximate solution obtained from the primary computation. This continuous solution is in fact what is provided to the user, and thus an assessment of the GE for this continuous approximate solution would be more relevant than what is considered here. For the second and fourth order primary solutions, interpolants of appropriate order can be provided by making use of the CMIRK schemes of orders four and six already available in `BVP_SOLVER`. For the sixth order primary solution, it would be useful to derive an eighth order CMIRK scheme upon which to base the interpolant. It might also be worthwhile to develop an optimal eighth order MIRK to replace the one that is currently used.

The results suggest that further investigation into the GE control mode for `BVP_SOLVER` as well as the hybrid GE/defect control modes is warranted.

REFERENCES

Ascher, U. M., Christiansen, J., and Russell, R. D. 1981. Collocation software for boundary value odes. *ACM Trans. Math. Softw. 7*, 209–222.

Ascher, U. M., Mattheij, R. M. M., and Russell, R. D. 1995. *Numerical solution of boundary value problems for ordinary differential equations.* Classics in Applied Mathematics, vol. 13. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA. Corrected reprint of the 1988 original.

Ascher, U. M. and Spiteri, R. J. 1994. Collocation software for boundary value differential-algebraic equations. *SIAM J. Sci. Comput. 15,* 4, 938–952.

Auzinger, W., Kneisl, G., Koch, O., and Weinmüller, E. 2002. Sbvp 1.0 - a matlab solver for singular boundary value problems. Tech. Rep. 02/02, Department for Analysis and Scientific Computing, Vienna University of Technology.

Bader, G. and Ascher, U. 1987. A new basis implementation for a mixed order boundary value ODE solver. *SIAM J. Sci. Statist. Comput. 8,* 4, 483–500.

Boisvert, J. J., Muir, P. H., and Spiteri, R. J. 2012. A numerical study of global error and defect control schemes for BVODEs. Tech. rep., Saint Mary's University, Dept. of Math. and Comp. Sci. Technical Report Series, cs.smu.ca/tech_reports/.

Burrage, K., Chipman, F. H., and Muir, P. H. 1994. Order results for mono-implicit Runge–Kutta methods. *SIAM J. Numer. Anal. 31,* 3, 876–891.

Capper, S., Cash, J., and Mazzia, F. 2007. On the development of effective algorithms for the numerical solution of singularly perturbed two-point boundary value problems. *Int. J. Comput. Sci. Math. 1,* 1, 42–57.

Cash, J. R. and Mazzia, F. 2006. Hybrid mesh selection algorithms based on conditioning for two-point boundary value problems. *JNAIAM J. Numer. Anal. Ind. Appl. Math. 1,* 1, 81–90.

Cash, J. R., Mazzia, F., Sumarti, N., and Trigiante, D. 2006. The role of conditioning in mesh selection algorithms for first order systems of linear two point boundary value problems. *J. Comput. Appl. Math. 185,* 2, 212–224.

Cash, J. R., Moore, G., and Wright, R. W. 1995. An automatic continuation strategy for the solution of singularly perturbed linear two-point boundary value problems. *J. Comput. Phys. 122,* 2, 266–279.

Cash, J. R. and Wright, M. H. 1991. A deferred correction method for nonlinear two-point boundary value problems: implementation and numerical evaluation. *SIAM J. Sci. Statist. Comput. 12,* 4, 971–989.

Enright, W. H. and Muir, P. H. 1996. Runge–Kutta software with defect control for boundary value ODEs. *SIAM J. Sci. Comput. 17,* 2, 479–497.

Enright, W. H. and Muir, P. H. 2010. New interpolants for asymptotically correct defect control of BVODEs. *Numer. Algorithms 53,* 2-3, 219–238.

Gupta, S. 1985. An adaptive boundary value Runge–Kutta solver for first order boundary value problems. *SIAM J. Numer. Anal. 22,* 1, 114–126.

Hale, N. and Moore, D. R. 2008. A sixth-order extension to the MATLAB package bvp4c of J. Kierzenka and L. Shampine. Tech. Rep. 08/04, Oxford University Computing Laboratory, Numerical Analysis Group, Oxford.

Hanson, P. M. and Enright, W. H. 1983. Controlling the defect in existing variable-order Adams codes for initial value problems. *ACM Trans. Math. Software 9,* 1, 71–97.

Higham, N. J. 1988. FORTRAN codes for estimating the one-norm of a real or complex matrix, with applications to condition estimation. *ACM Trans. Math. Software 14,* 4, 381–396 (1989).

Kierzenka, J. and Shampine, L. F. 2001. A BVP solver based on residual control and the MATLAB PSE. *ACM Trans. Math. Software 27,* 3, 299–316.

Kierzenka, J. and Shampine, L. F. 2008. A BVP solver that controls residual and error. *JNAIAM J. Numer. Anal. Ind. Appl. Math. 3,* 1-2, 27–41.

Koch, O. 2005. Asymptotically correct error estimation for collocation methods applied to singular boundary value problems. *Numer. Math. 101,* 1, 143–164.

Lentini, M. and Pereyra, V. 1974. A variable order finite difference method for nonlinear multipoint boundary value problems. *Math. Comp. 28*, 981–1003.

Lentini, M. and Pereyra, V. 1977. An adaptive finite difference solver for nonlinear two-point boundary problems with mild boundary layers. *SIAM J. Numer. Anal. 14,* 1, 94–111. Papers on the numerical solution of two-point boundary-value problems (NSF-CBMS Regional Res. Conf., Texas Tech Univ., Lubbock, Tex., 1975).

Mazzia, F. and Trigiante, D. 2004. A hybrid mesh selection strategy based on conditioning for boundary value ODE problems. *Numer. Algorithms 36,* 2, 169–187.

Moore, P. K. 2001. Interpolation error-based a posteriori error estimation for two-point boundary value problems and parabolic equations in one space dimension. *Numer. Math. 90,* 1, 149–177.

Muir, P. and Owren, B. 1993. Order barriers and characterizations for continuous mono-implicit Runge–Kutta schemes. *Math. Comp. 61,* 204, 675–699.

Muir, P. H. 1999. Optimal discrete and continuous mono-implicit Runge–Kutta schemes for BVODEs. *Adv. Comput. Math. 10,* 2, 135–167.

Russell, R. D. and Christiansen, J. 1978. Adaptive mesh selection strategies for solving boundary value problems. *SIAM J. Numer. Anal. 15,* 1, 59–80.

Shampine, L. F. and Muir, P. H. 2004. Estimating conditioning of BVPs for ODEs. *Math. Comput. Modelling 40,* 11-12, 1309–1321.

Shampine, L. F., Muir, P. H., and Xu, H. 2006. A user-friendly Fortran BVP solver. *JNAIAM J. Numer. Anal. Ind. Appl. Math. 1,* 2, 201–217.

Wright, K. 2007. Adaptive methods for piecewise polynomial collocation for ordinary differential equations. *BIT 47,* 1, 197–212.

Wright, R., Cash, J., and Moore, G. 1994. Mesh selection for stiff two-point boundary value problems. *Numer. Algorithms 7,* 2-4, 205–224.