CANADIAN APPLIED
MATHEMATICS QUARTERLY
Volume 17, Number 3, Fall 2009

SUPERCONVERGENT INTERPOLANTS FOR EFFICIENT SPATIAL ERROR ESTIMATION IN 1D PDE COLLOCATION SOLVERS

TOM ARSENAULT, TRISTAN SMITH AND PAUL MUIR

ABSTRACT. This paper considers the use of a superconvergent interpolant (SCI) for spatial error estimation when Gaussian collocation is employed as the spatial discretization scheme in a method-of-lines algorithm for the numerical solution of a system of one-dimensional parabolic partial differential equations (PDEs). Gaussian collocation is a popular approach for the spatial discretization of parabolic PDEs, and at certain points within the problem domain, the collocation solution is superconvergent. This paper describes how an interpolant based on these superconvergent values can be used to provide an efficient error estimate for the collocation solution. We implement this scheme within a modified version of the collocation PDE solver, BACOL. The original BACOL code obtains a spatial error estimate by computing a second global collocation solution of one higher order of accuracy. We show that the SCI based error estimation approach can provide spatial error estimates of comparable accuracy to those currently computed by BACOL, but at a much lower cost.

1 Introduction In this paper we focus on one of the most commonly arising subclasses of partial differential equations (PDEs) known as time dependent, one-dimensional (1D) parabolic equations. We discuss the development of a superconvergent interpolation (SCI) approach to spatial error estimation when Gaussian collocation is employed as the spatial discretization scheme in a method-of-lines (MOL) algorithm for this problem class.

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.

AMS subject classification: 65M15, 65M20, 65M70.

Keywords: 1D parabolic PDEs, collocation, spatial error estimation, interpolation, superconvergence, method-of-lines.

Over the last few decades a number of MOL software packages for the numerical solution of this problem class have been developed. In a recent study, [**21**], one such package, called BACOL [**20**], was shown to be comparable to and in some cases superior to other available packages for 1D parabolic PDEs, especially for problems exhibiting sharp spatial layer regions and for problems where a sharp tolerance is imposed. The BACOL package employs Gaussian collocation for the spatial discretization. This involves expressing the approximate solution at a given time as a linear combination of known spatial basis functions—piecewise polynomials of a given degree p—with unknown time dependent coefficients. These coefficients are determined by requiring the approximate solution to satisfy the PDEs at the images of the Gauss points on each subinterval of a mesh which partitions the problem domain. This yields a system of time dependent ordinary differential equations (ODEs) that together with the boundary conditions gives a system of differential-algebraic equations (DAEs) which is then solved using a modified version of DASSL [**5**] to obtain the time dependent coefficients. An estimate of the spatial error of this primary collocation solution is obtained by computing a secondary global collocation solution using piecewise polynomials of degree $p + 1$. The difference between these two approximate solutions gives an estimate of the spatial error in the primary collocation solution.

In the BACOL algorithm, the computation of the secondary global collocation solution represents a major computational cost. In this paper, we describe an alternative interpolation-based approach for the development of a more efficient spatial error estimation scheme. The key observation is that because BACOL employs collocation at Gauss points there are certain points within the problem domain where the collocation solution is superconvergent. We describe the development of a low cost interpolant based on these superconvergent values—the SCI—that can replace the secondary global collocation solution in the computation of the spatial error estimate.

This paper is organized as follows. In Section 2, we identify the problem class, give two examples, and provide a brief review of the relevant literature. Section 3 gives a brief description of algorithms upon which BACOL is based. In Section 4 we describe the alternative spatial error estimation scheme based on an SCI. Numerical results are provided in Section 5 to demonstrate the effectiveness of the alternative approach and in Section 6 we briefly compare the computational costs of the SCI based error estimation scheme with the one currently implemented within BACOL. We close, in Section 7, with our conclusions and

an indication of directions for future work.

2 Background We will assume a system of PDEs with *NPDE* components having the form

$$\underline{u}_t(x,t) = \underline{f}\left(t, x, \underline{u}(x,t), \underline{u}_x(x,t), \underline{u}_{xx}(x,t)\right), \quad a \le x \le b, \quad t \ge t_0, \tag{1}$$

with initial conditions

$$\underline{u}(x,t_0) = \underline{u}_0(x), \qquad a \le x \le b, \tag{2}$$

and separated boundary conditions

$$\underline{b}_L\left(t, \underline{u}(a,t), \underline{u}_x(a,t)\right) = \underline{0}, \quad \underline{b}_R\left(t, \underline{u}(b,t), \underline{u}_x(b,t)\right) = \underline{0}, \quad t \ge t_0. \tag{3}$$

This form includes the well known reaction-diffusion equations. Specific examples are:
(i) The simple test problem, [**15**]:

$$u_t = u_{xx} + \pi^2 \sin(\pi x), \quad 0 < x < 1, \quad t > 0, \tag{4}$$

with initial condition

$$u(x,0) = 1, \quad 0 \le x \le 1,$$

and boundary conditions

$$u(0,t) = u(1,t) = 1, \quad t > 0,$$

for which the exact solution is

$$u(x,t) = 1 + \sin(\pi x)\left(1 - e^{-\pi^2 t}\right).$$

(ii) Burgers' Equation, e.g., [**21**]:

$$u_t = \epsilon u_{xx} - u u_x, \qquad 0 < x < 1, \quad t > 0, \quad \epsilon > 0, \tag{5}$$

with the initial condition and boundary conditions chosen so that exact solution is given by

$$u(x,t) = \frac{1}{2} - \frac{1}{2}\tanh\left(\frac{1}{4\varepsilon}\left(x - \frac{t}{2} - \frac{1}{4}\right)\right),$$

where ε is a problem dependent parameter.

As mentioned earlier, over the last few decades, a number of high quality MOL software packages for the numerical solution of this problem class have been developed. Such packages include PDECOL [**15**] /EPDCOL [**14**], D03PPF [**7**], TOMS731 [**4**], MOVCOL [**13**], HPNEW [**16**], BACOL, and the related package, BACOLR [**23**]. In the MOL approach a spatial discretization, involving the use of a numerical scheme such as a finite difference or finite element method, based on a mesh of points which partition the spatial domain, is applied to the PDE, giving an approximation of the PDE by a system of time dependent ODEs. In the early MOL codes, the user was required to differentiate the boundary conditions and these, together with the ODEs from the discretization of the PDE, represented an initial value ODE problem whose solution, computed using high quality ODE software, approximated the solution of the PDE. More recently developed MOL codes treat the boundary conditions directly. The resultant system of coupled ODEs and algebraic equations—a system of DAEs—is then treated using high quality DAE software such as DASSL or RADAU5 [**12**].

Another important aspect of MOL software is the type of adaptivity and error control provided. All MOL codes have temporal adaptivity and temporal error control provided by the underlying initial value ODE/DAE solver. In the early MOL codes, no attempt was made to estimate and control the spatial error nor was any attempt made to adapt the spatial computation. EPDCOL is an example of a code of this type. The next development in MOL codes for PDEs involved the implementation of some form of spatial adaptivity, such as a moving mesh approach. The mesh movement is often determined by solving moving mesh PDEs which are based on some measure of solution behavior such as curvature. There is no attempt to compute and control an estimate of the spatial error. MOVCOL is an example of a code of this type. More recently developed MOL codes employ spatial adaptivity and spatial error control. Codes from this class compute an estimate of the spatial error and then adapt the spatial discretization, by changing the mesh and/or the order of accuracy of the spatial discretization scheme, in an attempt to compute a solution whose spatial error estimate is less than the user provided tolerance. HPNEW, BACOL, and BACOLR are examples of codes of this type.

There is a substantial body of literature on error estimation for the numerical solution of PDEs (see, e.g., [**1**] and references within). However, the recent work most closely related to the current investigation is by Moore [**16, 17, 18, 19**], in which interpolation error based error

estimates for 1D parabolic PDEs are discussed. We will discuss Moore's work in more detail, later in this paper. Since the interpolants we consider in this paper depend on the superconvergence properties of the collocation solution, another relevant body of literature concerns the study of superconvergence results for the problem class we consider in this paper and for the related problem class of boundary value ODEs (BVODEs). Results for 1D parabolic PDEs are discussed in [**6**] and [**9**]. Results for BVODEs are discussed in, e.g., [**3**].

3 Overview of BACOL Given a spatial mesh $a = x_0 < x_1 < \cdots < x_{NINT} = b$, the approximate solution is represented in BACOL as a linear combination of C^1-continuous B-spline basis functions [**8**] (piecewise polynomials of degree p where $3 \le p \le 11$) with time dependent coefficients. Thus the dimension of this piecewise polynomial subspace is $NC_p = NINT(p+1) - 2(NINT-1) = NINT(p-1) + 2$. Letting $\{B_{p,i}(x)\}_{i=1}^{NC_p}$ be the B-spline basis functions that support this piecewise polynomial space on the given mesh, the approximate solution, $\underline{U}(x,t)$, then has the form

$$\underline{U}(x,t) = \sum_{i=1}^{NC_p} \underline{y}_{p,i}(t) B_{p,i}(x), \tag{6}$$

where $\underline{y}_{p,i}(t)$ represents the (unknown) time dependent coefficient of the i-th B-spline basis function, $B_{p,i}(x)$.

The PDE is discretized in space by imposing collocation conditions on the approximate solution at images of the $p-1$ Gauss points (see, e.g., [**3**]) on each subinterval and by requiring the approximate solution to satisfy the boundary conditions. The collocation conditions have the form

$$\frac{d}{dt}\underline{U}(\xi_l,t) = \underline{f}\left(t,\xi_l,\underline{U}(\xi_l,t),\underline{U}_x(\xi_l,t),\underline{U}_{xx}(\xi_l,t)\right), \tag{7}$$

where $l = 2,\ldots,NC_p - 1$, and where the collocation points are defined by

$$\begin{aligned} &\xi_l = x_{i-1} + h_i\rho_j, \quad \text{where } \; l = 1 + (i-1)(p-1) + j, \\ &\quad \text{for } \; i = 1,\ldots,NINT, \quad j = 1,\ldots,p-1, \end{aligned} \tag{8}$$

where $h_i = x_i - x_{i-1}$ and $\{\rho_i\}_{i=1}^{p-1}$ are the images of the $p-1$ Gauss points on $[0,1]$. The points, $\xi_1 = a$ and $\xi_{NC_p} = b$ are associated with

requiring the approximate solution to satisfy the boundary conditions, and this gives the remaining two equations

$$\underline{b}_L\left(t, \underline{U}(a,t), \underline{U}_x(a,t)\right) = \underline{0}, \qquad \underline{b}_R\left(t, \underline{U}(b,t), \underline{U}_x(b,t)\right) = \underline{0}.$$

The collocation conditions (7) represent a system of ODEs (in time) for which the unknown solution components are the time dependent coefficients, $\underline{y}_{p,i}(t)$. These ODEs coupled with the boundary conditions give an index-1 system of DAEs, which, as mentioned earlier, is treated using DASSL. After DASSL has computed approximations for the $\underline{y}_{p,i}(t)$ values at time t, these can be employed together with the known B-spline basis functions $B_{p,i}(x)$ within (6), to obtain values of the approximate solution at desired x values, for the current time t.

The collocation solution $\underline{U}(x,t)$ for the current time is accepted by BACOL only if its spatial error estimate satisfies the user tolerance. This spatial error estimate is obtained by computing a second global collocation solution on the same spatial mesh at the same time t. This approximate solution, which we call $\bar{\underline{U}}(x,t)$, has the form

$$\bar{\underline{U}}(x,t) = \sum_{i=1}^{NC_{p+1}} \underline{y}_{p+1,i}(t) B_{p+1,i}(x). \tag{9}$$

This approximate solution is based on a set of C^1-continuous B-spline basis functions $B_{p+1,i}(x)$, polynomials of degree $p+1$ on each subinterval, with corresponding unknown time dependent coefficients $\underline{y}_{p+1,i}(t)$. Here $NC_{p+1} = NINT \cdot p + 2$. These unknowns are determined by imposing p collocation conditions per subinterval as well as the boundary conditions on $\bar{\underline{U}}(x,t)$. The collocation points in this case are the images of p Gauss points on $[0,1]$ mapped onto each subinterval. As before, this leads to a system of DAEs whose solution gives the functions $\underline{y}_{p+1,i}(t)$. In order to insure that the two approximate solutions $\underline{U}(x,t)$ and $\bar{\underline{U}}(x,t)$ are available at the same time t, the two DAE systems are provided to DASSL as one larger DAE system so that DASSL treats both systems of DAEs with the same time-stepping strategy. See [**20**] for further details.

It is shown in [**6**] and [**9**] that a collocation solution of degree p has an error that is $O(h^{p+1})$, where h is the maximum mesh spacing. (We will say that the collocation solution is of order $p+1$.) We then have

$$\begin{aligned} \|\underline{U}(x,t) - \bar{\underline{U}}(x,t)\|_\infty &= \|(\underline{U}(x,t) - \underline{u}(x,t)) - (\bar{\underline{U}}(x,t) - \underline{u}(x,t))\|_\infty \\ &= \|(\underline{U}(x,t) - \underline{u}(x,t))\|_\infty + O(h^{p+2}), \end{aligned}$$

and thus, for sufficiently small h, the difference between the two collocation solutions gives, asymptotically, an estimate of the error in the lower order collocation solution, $\underline{U}(x,t)$.

In BACOL, the following *a posteriori* spatial error estimates are computed. Denote the sth component of $\underline{U}(x,t)$ by $U_s(x,t)$ and the sth component of $\bar{\underline{U}}(x,t)$ by $\bar{U}_s(x,t)$. Let $ATOL_s$ and $RTOL_s$ be the absolute and relative tolerances for the s-th component of the approximate solution. Then BACOL computes a set of $NPDE$ normalized error estimates over the whole spatial domain of the form

$$E_s(t) = \sqrt{\int_a^b \left(\frac{U_s(x,t) - \bar{U}_s(x,t)}{ATOL_s + RTOL_s|U_s(x,t)|} \right)^2 dx}, \quad s = 1, \ldots, NPDE. \tag{10}$$

BACOL also computes a second set of $NINT$ normalized error estimates of the form,

$$\widehat{E}_i(t) = \sqrt{\sum_{s=1}^{NPDE} \int_{x_{i-1}}^{x_i} \left(\frac{U_s(x,t) - \bar{U}_s(x,t)}{ATOL_s + RTOL_s|U_s(x,t)|} \right)^2 dx}, \quad i = 1, \ldots, NINT. \tag{11}$$

Note that $E_s(t),\ s = 1, \ldots, NPDE$, and $\widehat{E}_i(t),\ i = 1, \ldots, NINT$, are estimates of the error associated with the *lower* order solution, $\underline{U}(x,t)$.

The approximate solution, $\underline{U}(x,t)$, is accepted at the current time, t, if

$$\max_{1 \le s \le NPDE} E_s(t) \le 1. \tag{12}$$

Otherwise, based on the error estimates, $\widehat{E}_i(t),\ i = 1, \ldots, NINT$, BACOL attempts to construct a new mesh that (i) has as many mesh points as necessary to yield an approximate solution whose estimated error will satisfy the user tolerances, and (ii) approximately equidistributes the estimated error over the subintervals of the new mesh. See [**20, 22**] for further details.

4 SCI based spatial error estimation

4.1 Overview Rather than compute a second more accurate and independent numerical solution to serve as the basis for an error estimate, a popular strategy for error estimation, examples of which are Gauss-Kronrod formulas in numerical quadrature and Runge-Kutta formula pairs in the numerical solution of initial value ODEs, involves an auxiliary computation that makes use of some of the information from the computation of the primary numerical solution. This auxiliary computation provides enough additional information to allow one to construct a higher order approximate solution for use in the error estimate. A similar strategy might be investigated here but it turns out that there is already higher order solution information available after the computation of $\underline{U}(x,t)$.

The key idea is as follows. In BACOL, the collocation points are chosen to be the images of the Gauss points on each subinterval and because of this it turns out that there are a number of special points on each subinterval of the spatial mesh where the collocation solution, $\underline{U}(x,t)$, generally of order $p+1$, is *superconvergent*, i.e., of order $p+2$. Thus simply evaluating $\underline{U}(x,t)$ at these known special points provides superconvergent solution information. Then a superconvergent interpolant (SCI) based on a sufficient number of these superconvergent values can be constructed and can replace $\bar{\underline{U}}(x,t)$ in the computation of the spatial error estimates (11) and (12).

4.2 Superconvergence results As mentioned previously, the papers [**6**] and [**9**] provide the standard convergence results for Gaussian collocation applied to a 1D parabolic PDE. We briefly review the relevant results here. Let $h_i = x_{i+1} - x_i$ and $h = \max_i h_i$. Collocating at k points per subinterval requires that we use piecewise polynomials of degree $p = k+1$. The collocation solution then has an error that is $O(h^{p+1}) \equiv O(h^{k+2})$ over the spatial domain, and, furthermore, at mesh points, both the solution and its derivative have errors that are $O(h^{2(p-1)}) \equiv O(h^{2k})$. Thus, the solution approximations at the mesh points are superconvergent provided $2(p-1) > p+1 \Rightarrow p > 3 \equiv k > 2$.

The above theory is consistent with that from the BVODE context. Consider the second order nonlinear BVODE (let $u^{(j)}(x)$) be the jth derivative of $u(x)$)),

$$u^{(2)}(x) - f(x, u(x), u^{(1)}(x)) = 0, \quad a < x < b, \quad g(u(a), u(b)) = 0.$$

With appropriate assumptions, Theorem 5.140/Corollary 5.142 of [**3**]

provide several results associated with applying a k-point Gaussian collocation method to this ODE. The results that are most relevant to the current study are the following. Let $U(x)$ be the collocation solution.

(i) At the mesh points, the collocation error satisfies

$$|u^{(j)}(x_i) - U^{(j)}(x_i)| = O(h^{2k}), \quad j = 0, 1, \quad i = 0, \ldots, NINT.$$

(ii) At nonmesh points, the collocation error satisfies

$$u^{(j)}(x) - U^{(j)}(x) = u^{(k+2)}(x_i) P^{(j)}\left(\frac{x - x_i}{h_i}\right) h_i^{k+2-j} + O(h_i^{k+3-j}) + O(h^{2k}),$$

where $x_i < x < x_{i+1}$, $i = 0, \ldots, NINT - 1$, $j = 0, \ldots, k + 1$, and where

$$P(\xi) = \frac{1}{k!} \int_0^{\xi} (t - \xi) \prod_{r=1}^{k} (t - \rho_r) dt.$$

The BVODE results given in (ii) provide details of the coefficient of the leading term in the error: one can expect to see higher accuracy in the collocation solution at points within each subinterval that correspond to roots of the polynomial $P(\xi)$ (as long as $k \geq 3$); similarly, one can also expect to see higher accuracy in the first derivative of the collocation solution at the roots of the first derivative of $P(\xi)$ (as long as $k \geq 2$.) To our knowledge, the result corresponding to (ii) for the PDE case has not been proved. However it appears that these results do hold for the 1D parabolic PDE case: in [**2**] we provide experimental evidence demonstrating that, for the spatial discretization of a 1D parabolic PDE by Gaussian collocation, the orders of convergence described by the above BVODE theory—point (ii) above in particular—also hold for the PDE case. Figure 1 shows the order of convergence for the collocation solution and its derivative as well as the locations of the superconvergent solution and derivative values on a single subinterval, for the case $k = 5$.

4.3 Selection of the superconvergent points We will construct C^1-continuous, piecewise polynomial interpolants that use a sufficient number of superconvergent solution and derivative values so that the interpolation error is dominated by the data error. (Here, the data error refers to the order of accuracy of the solution and derivative values to be

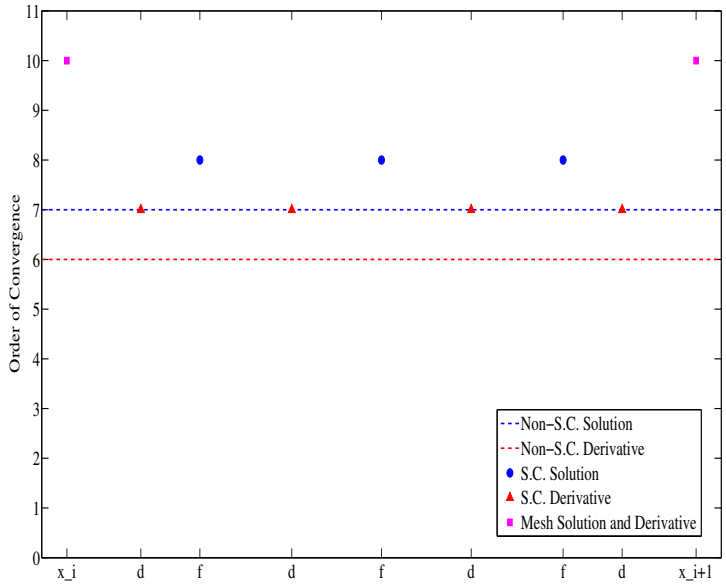

FIGURE 1: Order of convergence of the collocation solution for $k = 5$ on one subinterval. The collocation solution over the subinterval is not superconvergent (Non-S.C. Solution) and has order 7, while its derivative, also not superconvergent (Non-S.C. Derivative) has order 6. The points labeled 'f' correspond to the roots of the polynomial $P(\xi)$ at which the collocation solution is superconvergent (S.C. Solution) and has order 8. The points labeled 'd' correspond to the roots of the derivative of $P(\xi)$ where the derivative of the collocation solution is superconvergent (S.C. Derivative) and has order 7. The order of convergence for the mesh point solution and derivative values is 10.

interpolated. For example, suppose that the values to be interpolated have errors that are $O(h^p)$. Then we will use a sufficient number of interpolation points so that the interpolation error is $O(h^q)$, where $q > p$. See, e.g., [**11**] for further discussion of this point.)

Recall that when the collocation solution is a polynomial of degree p on each subinterval the number of collocation points per subinterval, $k = p-1$, and the superconvergent solution and derivatives values will be order $p+2$ and $p+1$, respectively. Thus, in order to have the interpolation error of the SCI dominated by the data error of the collocation solution values, we will need to choose the SCI to be a polynomial of degree $p+2$

on each subinterval. That polynomial will be specified by requiring it to interpolate $p+3$ superconvergent collocation solution values. Then the interpolation error will be order $p+3$, one order higher than the data error associated with the superconvergent collocation solution values. Since we will employ a combination of solution and derivative values, a Hermite-Birkhoff form for the interpolant is appropriate and we will consider results from [**10**] for the form of the interpolant and its error term.

In order to obtain C^1-continuity over the spatial domain, the polynomial that represents the SCI on a given subinterval must interpolate the superconvergent solution and derivative values at the endpoints of the subinterval. These values represent 4 of the $p+3$ required interpolation values and we must select $(p+3)-4=p-1$ additional superconvergent values. *There would appear to be more than enough additional superconvergent values available on each subinterval.* For example, for the case $k=5$ ($p=6$), we will need $p+3=9$ superconvergent values, and, as shown in Figure 1, there are 11 superconvergent values available (3 solution values internal to the subinterval, 4 derivative values internal to the subinterval, and mesh point solution and derivative values at each end of the subinterval.)

However, for all values of k, all of our attempts to construct an interpolant that uses only solution and derivative values contained within a single subinterval led to existence issues. The paper [**10**] identifies a matrix that must be non-singular in order for the Hermite-Birkhoff interpolant to exist. For several k values, we have checked a number of possible combinations of superconvergent interpolation values from those available on a given subinterval and have found that in each case this matrix is singular. *We have not been able to construct an interpolant that uses the 4 endpoint solution and derivative values together with any combination of $p-1$ internal solution and derivative values, for any value of k.*

In order to avoid this issue, the approach we have used employs, for a given subinterval, the superconvergent endpoint solution and derivative values, all of the superconvergent *solution* values that are internal to the subinterval, and *the closest available superconvergent solution values from each adjacent subinterval.* (We do not choose superconvergent derivative values from outside the current subinterval because this leads to conditional non-singularity of the matrix that defines the interpolant, depending on the ratio of the size of the current subinterval to that of the adjacent subintervals.) With this choice of interpolation values, we find that there is no issue with the existence of the interpolant. For the

leftmost and rightmost subintervals, we employ the two closest superconvergent solution values available in the lone adjacent subinterval. The choice of superconvergent values is shown for the case $k = 5$ in Figure 2. The circled values are the ones we select to define the SCI interpolating polynomial on each subinterval. *Note that we always choose two of the solution values from outside the subinterval.*

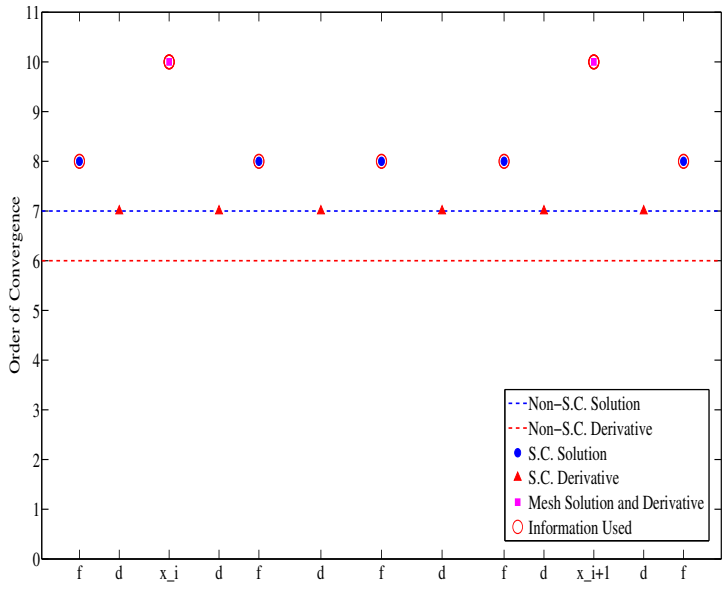

FIGURE 2: Choice of superconvergent values for $k = 5$ case. The interpolants considered in this paper use the superconvergent solution and derivative values at the endpoints of the subinterval, the superconvergent solution values within the current subinterval, and **the closest superconvergent solution value from each of the adjacent subintervals.** These values are circled (Information Used). Compare with Figure 1.

4.4 The SCI Consider the subinterval $[x_i, x_{i+1}]$. Let $s_1 = x_i$ and $s_2 = x_{i+1}$ and let w_j, $j = 1, \ldots, k$, be the nonmesh points at which we will interpolate superconvergent values. Then, associated with the

collocation solution $\underline{U}(x,t)$ on the given subinterval at time t, we have, from [**10**], the Hermite-Birkhoff SCI

$$\widetilde{\underline{U}}(x,t) = \sum_{j=1}^{2} H_j(x)\underline{U}(s_j,t) + h\sum_{j=1}^{2} \overline{H}_j(x)\underline{U}_x(s_j,t) + \sum_{j=1}^{k} G_j(x)\underline{U}(w_j,t),$$

where $x \in [x_i, x_{i+1}]$, $h = x_{i+1} - x_i$, and

$$H_j(x) = (1-(x-s_j)\gamma_j)\frac{\eta_j^2(x)\phi(x)}{\eta_j^2(s_j)\phi(s_j)},$$

$$\overline{H}_j(x) = (x-s_j)\frac{\eta_j^2(x)\phi(x)}{\eta_j^2(s_j)\phi(s_j)},$$

$$G_j(x) = \frac{\phi_j(x)\eta^2(x)}{\phi_j(w_j)\eta^2(w_j)},$$

where

$$\phi(x) = \prod_{r=1}^{k}(x-w_r), \qquad \phi_j(x) = \prod_{\substack{r=1\\ r\neq j}}^{k}(x-w_r),$$

$$\eta(x) = \prod_{r=1}^{2}(x-s_r), \qquad \eta_j(x) = \prod_{\substack{r=1\\ r\neq j}}^{2}(x-s_r),$$

and

$$\gamma_j = \sum_{i=1}^{k}\frac{1}{s_j-w_i} + 2\sum_{\substack{i=1\\ i\neq j}}^{2}\frac{1}{s_j-s_i}.$$

The paper [**10**] also provides an explicit expression for the interpolation error. For the general case, the expression is quite complicated and we therefore do not repeat it here. Because two of the superconvergent values are taken from outside the current subinterval, the locations of the corresponding interpolation points are expressed relative to the current subinterval size, and it is thus not surprising that the corresponding error term for the interpolant depends on the ratios of the size of the current subinterval to the sizes of the adjacent subintervals. The error

expression for the interpolant includes a factor that captures the dependence of the error on these ratios. For example, for $k = 5$, and for the subinterval $[x_i, x_{i+1}]$, this factor is

$$(x - x_i)^2 - (R\alpha + L\beta)(x - x_i) - R\alpha + L\beta + \frac{LR}{3} - 1,$$

where $\alpha = \frac{1}{2} - \frac{1}{6}\sqrt{3}$, $\beta = \frac{1}{2} + \frac{1}{6}\sqrt{3}$, and the adjacent subinterval ratios are

$$R = \frac{x_{i+2} - x_{i+1}}{x_{i+1} - x_i} \quad \text{and} \quad L = \frac{x_i - x_{i-1}}{x_{i+1} - x_i}.$$

For the leftmost and rightmost subintervals, the structure of the interpolant is slightly different and thus the error term is also slightly different. For the leftmost subinterval, when $k = 5$, the factor in the error term that captures the dependence on R is

$$(x - x_0)^2 - \left(R\left(\frac{1}{2} + \alpha\right) + 2\right)(x - x_0) + 1 + R\left(\frac{1}{2} + \alpha\right) - R^2\frac{\sqrt{3}}{12}.$$

In this case, we see that the error depends on the *square* of R. A similar expression holds for the rightmost subinterval, where the error depends on the square of L.

As mentioned earlier, the mesh refinement algorithm employed by BACOL is based on equidistribution and thus the size of a given subinterval is based entirely on this principle. There is no mesh smoothing, i.e., there is no imposed upper or lower bound on the subinterval ratios of the meshes determined by the BACOL mesh refinement algorithm. This allows for the mesh to adapt to the error estimate profile purely according to the equidistribution principle but obviously the absence of a bound on the adjacent subinterval ratios could impact negatively on the accuracy of the interpolants. We will explore this issue in the next section.

We can now briefly contrast the approach we describe in this paper with the related approach considered by Moore.

- In Moore's work, the computation of the primary finite element solution is based on a spatial discretization that uses a finite element Galerkin technique with a piecewise polynomial hierarchical spatial basis. A key idea is that the error estimate is based on a Lobatto interpolant to the finite element solution, for which the leading term in the interpolation error agrees asymptotically with the leading term in the error for the finite element solution. This is referred to as

the asymptotic equivalence property. One can then obtain an approximation for the error in the numerical solution of the PDE by estimating the error in the interpolant. The interpolant is based on evaluations of the finite element solution at the Lobatto points on each mesh subinterval and the explicit form of the interpolation error estimate comes from an extension of the error formula for the standard Lagrange interpolating polynomial. Another important aspect of Moore's work is that it involves the estimation of errors of order p and $p+1$ (assuming a primary computation of order p.) This is necessary because the computational algorithm he considers employs both h (mesh) and p (order) adaptivity.

- Our approach for the primary computation is based on collocation at Gauss points within each subinterval and the adaptivity is based solely on h-refinement. The interpolant we construct, the SCI, is one order higher than the collocation solution and is based on evaluation of that solution and its derivative at the mesh points and the evaluation of the solution at selected points both internal and external to the subinterval. Thus this interpolant is not a Lobatto interpolant. More fundamental is the observation that we do not expect the SCI to satisfy the asymptotic equivalence property and thus the idea of constructing an interpolant that satisfies the asymptotic equivalence property is not relevant in our approach.

5 A numerical investigation of the spatial error estimation schemes In this section we will present representative results from the testing we have performed (see [**2**]) in which the SCI error estimation scheme is compared with the original BACOL error estimation scheme. The first test problem is the simple problem (4). The second test problem is (5) where we choose $\epsilon = 10^{-3}$. For each problem, we integrate from $t = 0$ to $t = 1$. In the full set of numerical experiments [**2**], we have considered a range of *tol* values $(10^{-4}, 10^{-6}, 10^{-8})$ $(ATOL_s = RTOL_s = tol)$ and k values (3, 6, and 9.) With two error estimates available, there is also the question of which one to use in the spatial mesh refinement algorithm. We have thus considered two sets of tests, one set for which the BACOL error estimate is used to control the spatial mesh, and another set for which the SCI error estimate is used to control the spatial mesh. We provide plots to show the accuracy of the error estimation schemes. In each plot, the tick marks on the horizontal axis show the locations of the mesh points of the final mesh, when $t = 1$.

When BACOL is applied to the first test problem, with $k = 3$, *tol* =

10^{-8}, the error estimates obtained on each subinterval by each of the error estimation schemes compare well with each other and with the true error (see Figure 3). The solution to the problem is smooth, the mesh is relatively uniform, and there are no issues associated with large adjacent subinterval ratios. Figure 3 is for the case where BACOL controls the mesh refinement but since the SCI error estimates are essentially the same, SCI control of the mesh gives similar results and we do not show them here.

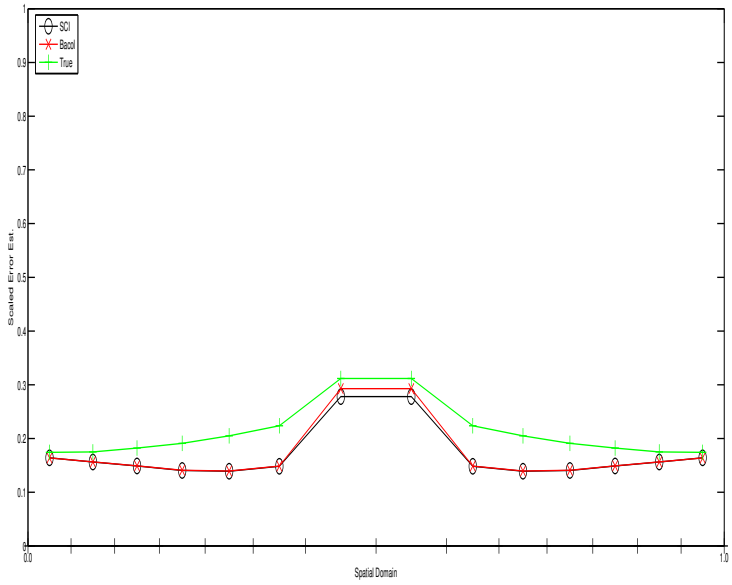

FIGURE 3: Plot of the SCI error estimate, the BACOL error estimate, and the true error associated with the collocation solution of (4) with $k = 3$, $tol = 10^{-8}$. The final mesh has $NINT = 14$. **BACOL controls the mesh refinement.**

For the next set of results, we choose $k = 6$ and $tol = 10^{-6}$. The solution to the second test problem has a sharp layer that moves from the region $(0.2, 0.3)$ to the region $(0.7, 0.8)$ as t goes from 0 to 1. BACOL employs a highly non-uniform mesh to efficiently solve this problem. A plot of the error estimates for each subinterval, as computed by the SCI and BACOL error estimation schemes, as well as the true error for each

subinterval, is given in Figure 4. *The BACOL error estimate controls the mesh adaptivity.* Most of the points of the mesh are located within the region of the spatial domain where the solution has a sharp layer. From Figure 4, we see that the SCI estimate substantially overestimates the true error on the leftmost and rightmost subintervals. (From Figure 4 it is clear that the leftmost and rightmost subintervals are substantially larger than the adjacent subintervals.) Figure 5 focuses on the part of

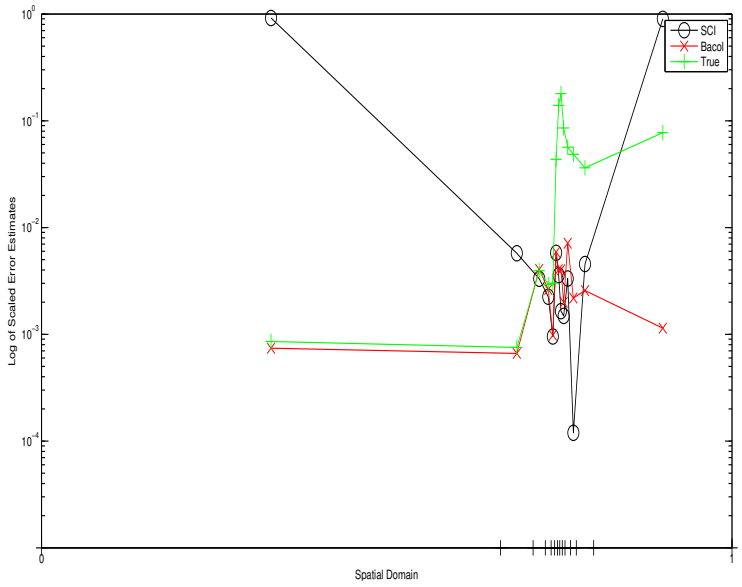

FIGURE 4: Plot of the SCI error estimate, the BACOL error estimate, and the true error associated with the collocation solution for (5) with $\epsilon = 10^{-3}$. The final mesh has $NINT = 13$. **BACOL controls the mesh refinement.**

Figure 4 corresponding to the layer region. From this plot we can see that both the BACOL error estimate and the SCI error estimate are in good agreement with each other but somewhat underestimate the true error in the layer region.

The next set of results are obtained by again considering the second test problem (and again choosing $k = 6$ and $tol = 10^{-6}$) but this time *we allow the SCI error estimate to control mesh refinement.* A plot of the

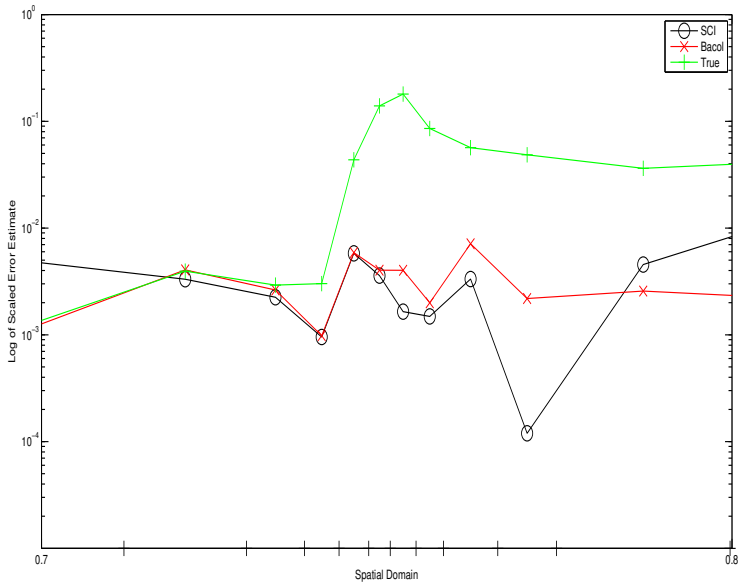

FIGURE 5: Plot of the SCI error estimate, the BACOL error estimate, and the true error associated with the collocation solution of (5) with $\epsilon = 10^{-3}$, within the layer region. **BACOL controls the mesh refinement.**

error estimates and the true error is given in Figure 6. Comparing this plot with Figure 4, we note that both estimates show better agreement with the true error on all subintervals. Comparing the meshes in Figure 4 and Figure 6 we see that when the BACOL error estimate controls mesh refinement, the first subinterval of the BACOL controlled mesh represents about 60% of the entire spatial domain, while the SCI error estimate controlled mesh has the first *two* subintervals covering approximately that same region of the spatial domain. The overestimates of the error on the subintervals where the adjacent subinterval ratios are large trigger further refinement of the mesh in these regions, and that has the effect of reducing the disparity between the sizes of adjacent subintervals. The use of the SCI error estimate to control the mesh thus "self-corrects" the issue of SCI over estimates of the error due to large subinterval ratios. Figure 7 focuses on the part of Figure 6 corresponding to the layer region. Even within the layer region we see reasonably good agreement between the error estimators and the true solution, when the

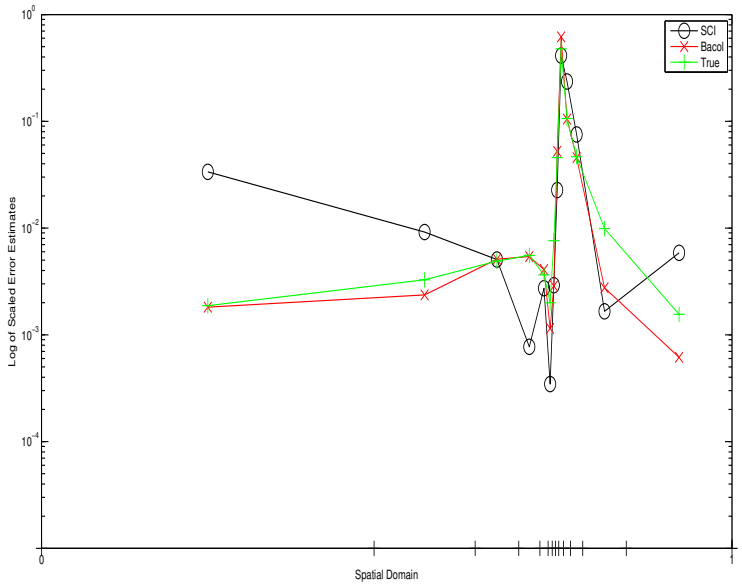

FIGURE 6: Plot of the SCI error estimate, the BACOL error estimate, and the true error associated with the collocation solution of (5) with $\epsilon = 10^{-3}$. The final mesh has $NINT = 13$. **SCI controls the mesh refinement.**

SCI estimate controls the mesh refinement. We see from Figure 6 that the error in the layer region is the largest error over the spatial domain and we see from Figure 7 that the error is well approximated by both error estimations schemes.

The above results are representative of the larger set of results reported in [**2**]. Based on the overall results we can make a few additional points regarding the limitations of the SCI approach:

(i) For the smaller k values, the SCI error estimates are sometimes not of sufficiently good quality to guide the mesh process. In such cases, we have seen BACOL fail when we let the SCI estimate control the mesh refinement. The failure happens at the beginning of the computation when $t = 0$.

(ii) Both error estimators generally lead to meshes that have the same number of subintervals, but occasionally, the meshes associated with the SCI error estimator have a few extra subintervals.

(iii) Recall from Section 4.2 that the SCI approach requires $p > 3$ whereas BACOL requires only $p \geq 3$.

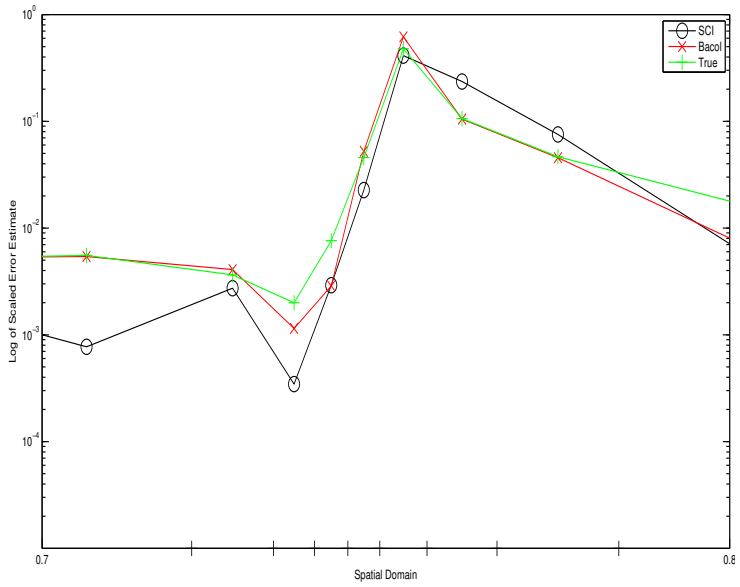

FIGURE 7: Plot of the SCI error estimate, the BACOL error estimate, and the true error associated with the collocation solution of (5) with $\epsilon = 10^{-3}$, within the layer region. **SCI controls the mesh refinement.**

6 Computational costs for the BACOL and SCI error estimates Recall that the BACOL error estimate is based on the computation of a second global collocation solution. Since this second solution involves the use of polynomials of one degree higher than the primary solution and thus one extra collocation point per subinterval is required, the cost of computing this second solution is slightly larger than the cost of computing the primary solution. For each of these solutions the most significant cost is the setup and factorization of the Newton matrices that arise and, assuming a mesh of $NINT$ subintervals, k collocation points per subinterval, and $NPDE$ equations, and the use of a linear system solver specifically designed to handle the almost block diagonal structure of the Newton matrices, these costs are $O(NINT(NPDE \times k)^3)$. Once the primary and secondary solutions are computed, the error estimate requires the evaluation of these global solutions and that involves evaluation of the corresponding B-spline basis polynomials. These costs are linear in $NINT, NPDE$, and k.

The costs for the construction of the SCI on the other hand are rel-

atively small. One must evaluate the primary solution several times on each subinterval to obtain the superconvergent solution values and then the evaluation of the SCI involves only the evaluation of the basis polynomials associated with the Hermite-Birkhoff interpolant discussed earlier. These costs are linear in $NINT, NPDE$, and k.

While the self-correction of the SCI error estimate occasionally leads to a few extra subintervals being added to the mesh, the costs per subinterval for the SCI approach are low and this does not add significantly to the overall costs. It is thus clear that the cost of the SCI error estimate is a small fraction of the cost of error estimate currently employed within BACOL.

7 Conclusions and future work We have seen from the numerical results that the SCI error estimation scheme generally yields error estimates of quality comparable to those given by the error estimation scheme currently employed by BACOL, *when the SCI error estimates are used to control mesh refinement.* Furthermore, because it employs readily available superconvergent solution information the SCI can be obtained at a relatively minor computational cost. The SCI approach therefore appears to be an interesting alternative to the error estimation scheme currently employed by BACOL and further investigation is warranted.

The numerical results reported here were obtained by using a modified form of BACOL that computes both the BACOL and SCI error estimates, and thus the primary and secondary collocation solutions are currently both computed by this modified form of BACOL. Ongoing work involves further modification of BACOL so that only the primary collocation solution and the SCI error estimate are computed. This will lead to a new version of BACOL that should have comparable performance to that of the current version but with about twice the speed.

We are also further exploring the theoretical results on collocation for one-dimensional PDEs in order to obtain supporting theory for our observed numerical results on the superconvergent solution and derivative values available within the interior of each subinterval of the spatial mesh. Another area of ongoing work is an examination of the interpolation conditions that appear to make it impossible to choose all the interpolation points from within a given subinterval. We have shown this to be true by considering specific values of k but a more general analysis may be possible, based on [**10**]. Ongoing work also includes an analysis of possibilities for the representation of the SCI. This will

include an exploration of which superconvergent values upon which to base the interpolant as well as the basis function representation for the interpolant.

It may be possible to generalize the approach discussed here to higher dimensions. The application of collocation for the numerical solution of parabolic PDEs in two or three dimensions has been studied for some time. If these collocation solutions possess appropriate superconvergence properties then it may be possible to construct superconvergent interpolants (in two or three dimensions respectively) based on a sufficient number of superconvergent solution and derivative values obtained from the collocation solution. This would then provide the basis for an error estimate similar to the approach discussed here.

Acknowledgments The authors wish to acknowledge the collaboration of Patrick Keast, Dalhousie University, in the earlier stages of this work.

The authors would also like to thank the referees for their many helpful comments.

REFERENCES

1. M. Ainsworth and J. T. Oden, *A posteriori error estimation in finite element analysis*, Pure and Applied Mathematics (New York), Wiley-Interscience, John Wiley & Sons, New York, 2000.
2. T. Arsenault, T. Smith, P. H. Muir and P. Keast, *Interpolation based spatial error estimation schemes for 1d parabolic collocation solvers*, Saint Mary's University, Dept. of Mathematics and Computing Science Technical Report Series,cs.smu.ca/tech_reports, in preparation, 2011.
3. U. M. Ascher, R. M. M. Mattheij and R. D. Russell, *Numerical solution of boundary value problems for ordinary differential equations*, Classics in Applied Mathematics **13**, SIAM, Philadelphia, PA, 1995.
4. J. G. Blom and P. A. Zegeling, *Algorithm 731: A moving-grid interface for systems of one-dimensional time-dependent partial differential equations*, ACM Trans. Math. Softw. **20**(2) (1994), 194–214.
5. K.E. Brenan, S.L. Campbell and L.R. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, SIAM, Philadelphia, PA, 1989.
6. J. H. Cerutti and S. V. Parter, *Collocation methods for parabolic partial differential equations in one space dimension*, Numer. Math. **26**(3) (1976), 227–254.
7. D03PPF, *NAG Fortran library*, The Numerical Algorithms Group, Ltd., Wilkinson House, Oxford, UK.
8. C. de Boor, *A practical guide to splines*, Applied Mathematical Sciences **27**, Springer-Verlag, New York, 1978.

9. J. Douglas, Jr. and T. Dupont, *Collocation methods for parabolic equations in a single space variable* Lecture Notes in Mathematics **385**, Springer-Verlag, Berlin, 1974.
10. W. F. Finden, *An error term and uniqueness for Hermite-Birkhoff interpolation involving only function values and/or first derivative values*, J. Comput. Appl. Math. **212**(1) (2008), 1–15.
11. I. Gladwell, L. F. Shampine, L. S. Baca and R. W. Brankin, *Practical aspects of interpolation in Runge-Kutta codes*, SIAM J. Sci. Statist. Comput. **8**(3) (1987), 322–341.
12. E. Hairer and G. Wanner, *Solving ordinary differential equations. II*, Springer Series in Computational Mathematics **14**, second edition, Springer-Verlag, Berlin, 1996.
13. W. Huang and R. D. Russell, *A moving collocation method for solving time dependent partial differential equations*, Appl. Numer. Math. **20**(1–2) (1996), 101–116.
14. P. Keast and P. H. Muir, *Algorithm 688: Epdcol: a more efficient pdecol code*, ACM Trans. Math. Softw. **17**(2) (1991), 153–166.
15. N. K. Madsen and R. F. Sincovec, *Algorithm 540: Pdecol, general collocation software for partial differential equations [d3]*, ACM Trans. Math. Softw. **5**(3) (1979), 326–351.
16. P. K. Moore, *Interpolation error-based a posteriori error estimation for two-point boundary value problems and parabolic equations in one space dimension*, Numer. Math. **90**(1) (2001), 149–177.
17. P. K. Moore, *Applications of Lobatto polynomials to an adaptive finite element method: a posteriori error estimates for hp-adaptivity and grid-to-grid interpolation*, Numer. Math. **94**(2) (2003), 367–401.
18. P. K. Moore, *An implicit interpolation error-based error estimation strategy for hp-adaptivity in one space dimension*, J. Numer. Math. **12**(2) (2004), 143–167.
19. P. K. Moore, *Interpolation error-based a posteriori error estimation for hp-refinement using first and second derivative jumps*, Appl. Numer. Math. **48**(1) (2004), 63–82.
20. R. Wang, P. Keast and P. H. Muir, *BACOL: B-spline Adaptive COLlocation software for 1-D parabolic PDEs*, ACM Trans. Math. Software **30**(4) (2004), 454–470.
21. R. Wang, P. Keast and P. H. Muir, *A comparison of adaptive software for 1D parabolic PDEs*, J. Comput. Appl. Math. **169**(1) (2004), 127–150.
22. R. Wang, P. Keast and P. H. Muir, *A high-order global spatially adaptive collocation method for 1-D parabolic PDEs*, Appl. Numer. Math. **50**(2) (2004), 239–260.
23. R. Wang, P. Keast and P. H. Muir, *Algorithm 874: BACOLR—spatial and temporal error control software for PDEs based on high-order adaptive collocation*, ACM Trans. Math. Software **34**(3) (2008), Art. 15, 28.

Corresponding author: Paul Muir
Saint Mary's University, Halifax, NS, Canada, B3H 3C3,
E-mail address: muir@smu.ca