Efficient Interpolation-based Error Estimation for 1D Time-Dependent PDE Collocation Codes *

Tom Arsenault[†], Tristan Smith[‡], Paul Muir[§], Pat Keast[¶]

Abstract

This report describes some recent work on interpolation based approaches to spatial error estimation when Gaussian collocation is employed as the spatial discretization method in a method-of-lines algorithm for the numerical solution of a system of one-dimensional parabolic partial differential equations (PDEs). At certain points within the problem domain, the collocation solution is superconvergent and this report describes how an interpolant based on these superconvergent values can be used to provide an efficient error estimate for the collocation solution. We also consider a second approach that involves an interpolant for which the interpolation error is asymptotically equivalent to the error of the collocation solution. We implement these new schemes within a modified version of a parabolic PDE collocation solver, BACOL, a recently developed software package for the numerical solution of systems of 1D time-dependent parabolic PDEs. BACOL employs a high order spatial discretization scheme based on B-spline collocation. BACOL generates the spatial error estimate by computing *two* global collocation solutions to the PDEs, one based on B-splines of degree p and the other on B-splines of degree $p+1$. The difference between the two collocation solutions gives a high order estimate of the spatial error of the lower order collocation solution. The computation of the two global collocation solutions is obviously a significant computational expense; the two interpolation based approaches mentioned above provide low cost alternatives upon which to base the error estimation scheme. Numerical results are provided to compare these new error estimates with the one currently employed within BACOL.

Subject Classification: 65M15, 65M20, 65M70
Keywords: 1D Parabolic PDEs, Collocation, Spatial Error Estimation, Interpolation, Superconvergence, Method-of-lines

1 Introduction

In this report we describe some recent work on the development of interpolation based approaches to spatial error estimation in a method-of-lines (MOL) algorithm for the numerical solution of a system of one-dimensional (1D) parabolic partial differential equations (PDEs). We will assume a system of PDEs with *NPDE* components having the form

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

with initial conditions

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

and separated boundary conditions

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

The last few decades have seen the development of a number of high quality MOL software packages for the numerical solution of this problem class. Such packages include PDECOL/EPDCOL [29], [27], D03PPF [13], TOMS731 [7], MOVCOL [26], HPNEW [30], and BACOL/BACOLR [38, 39, 40, 41]. In the MOL approach, a spatial discretization - involving the use of a numerical scheme such as a finite difference or finite element

*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.

[†]University of Western Ontario, London, ON, Canada, N6A 5B8
[‡]Scotiabank, 5201 Duke St., Halifax, NS, Canada, B3J 1N9
[§]Saint Mary's University, Halifax, NS, Canada, B3H 3C3, muir@smu.ca
[¶]Dalhousie University, Halifax, NS, Canada

method, and typically 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 ordinary differential equations (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 a 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 differential-algebraic equations (DAEs) - is then treated using high quality DAE software such as DASSL [8] or RADAU5 [23]. The former package implements backward differentiation formulas - see, e.g., [8] - while the latter implements a Radau type implicit Runge-Kutta method - see, e.g., [23].

Another aspect of MOL software for time dependent 1D parabolic problems is the type of adaptivity and error control provided. (Error control means that the computation is adapted so that a high order estimate of the error in the approximate solution is less than a user-provided tolerance, i.e., an approximate solution is not returned by the code unless the error estimate satisfies the user tolerance.) Early MOL codes had only temporal adaptivity and error control as provided by the underlying initial value ODE solver. (Such solvers typically estimate the local error in time and then adaptively choose the time step and possibly the order of the time integration scheme so that the estimated error is less than a user provided tolerance.) 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* - typically these codes employ a moving mesh approach that, for a given number of subintervals, adaptively moves the mesh points within the spatial domain; the mesh movement is often determined by solving moving mesh PDEs which are based on some measure of solution behavior such as curvature rather than on a high order estimate of the spatial error. Temporal adaptivity and error control is provided by the underlying initial value DAE solver, as before, and spatial adaptivity is provided through the moving mesh algorithm but there is no attempt to compute and control a high order estimate of the spatial error. MOVCOL is an example of a code of this type. More recently developed MOL codes employ both temporal and spatial adaptivity and *error control*. In addition to the temporal adaptivity and error control provided by the initial value solver, codes from this class also compute a high order estimate of the spatial error and then adapt the spatial discretization - by changing the mesh and/or the order of the 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.

As mentioned above, the initial value solvers employed by MOL codes usually estimate and control the *local* temporal error. It would be preferable for MOL codes to employ initial value solvers that estimate and control the *global* temporal error but the development of such codes is currently an ongoing area of investigation. See, e.g., [28], [10]. To our knowledge, there do not exist production level packages for parabolic PDEs that provide control of the temporal global error.

In a recent study, [38], the BACOL package was shown to be comparable to and in some cases superior to other available packages for this problem class, especially for problems exhibiting sharp spatial layer regions and for problems where higher accuracy is required. The BACOL package employs collocation for its 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. The unknown coefficients are determined by requiring the approximate solution to satisfy the PDE at a set of collocation points (images of the Gauss points, see, e.g., [3], on each subinterval) distributed over the problem domain - this leads to a set of time dependent ODEs that together with the boundary conditions gives a system of DAEs which is then solved using a modified version of DASSL to obtain the unknown time dependent coefficients. DASSL returns approximations for the B-spline coefficients for which the estimated error is less than the user provided tolerance. An estimate of the spatial error of the approximate solution is obtained by computing a second global solution approximation using the same general collocation approach but with piecewise polynomials of degree $p+1$. The unknown time dependent coefficients for the two collocation solutions are computed simultaneously by DASSL. After each accepted time step taken by DASSL, these coefficients are used to construct the two collocation solutions and the difference between these two approximate solutions gives an estimate of the spatial error *in the lower order collocation solution*. When the error estimate is less than the user tolerance, the solution at the current time is accepted and the computation continues on to attempt the next time step. Otherwise, the solution is rejected and the mesh is modified (points are added or taken away and/or moved) based on an estimate of the error on each subinterval. The process then repeats on this new mesh.

In the above algorithm, the computation of a spatial error estimate for the degree p collocation solution that involves the computation a second global solution of degree $p+1$ represents a major computational cost: the computation of the error estimate costs more than the computation of the primary solution. The purpose of this report, therefore, is to explore two alternative interpolation-based approaches for improving the efficiency of the BACOL code through the development of more efficient spatial error estimation schemes. Such improvements will also be applicable to the BACOLR code (BACOLR is similar to BACOL except that it uses a modified version of RADAU5 rather than a modified version of DASSL as the DAE solver) but within this report we will focus on BACOL. As mentioned above, the current spatial error estimate is based on the difference between the higher order and the lower order collocation solutions. In one of the approaches for improving the cost of the spatial error estimate that we consider in this report, we develop a low cost (superconvergent) interpolant that can replace the *higher* order global collocation solution in the computation of the spatial error estimate. The other approach we consider replaces the *lower* order global collocation solution in the spatial error estimate with a low cost interpolant whose interpolation error is asymptotically equivalent to the error of the *lower* order collocation solution. We will discuss these approaches in more detail in the next section.

There is a substantial body of literature on error estimation for the numerical solution of PDEs - see, e.g., [1] and [19] and references within. However, the recent work most closely related to the investigation described in this report is by Moore [30, 31, 34, 33], in which interpolation error based error estimates for 1D parabolic PDEs are discussed. A key idea in Moore's work is the development of an interpolant, based on the current numerical solution of the PDE, for which the leading term in the interpolation error agrees asymptotically with the leading term in the error for the numerical solution. One can then obtain an approximation for the error in the numerical solution of the PDE by estimating the error in the interpolant, a task that is more easily accomplished than the direct estimate of the numerical solution error. Other examples of recent work on error estimation for PDEs include [4], [5], [37], [32, 35]. See also the recent book [6] and references within.

Since one class of interpolants we consider in this report 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 report and for the related problem class of boundary value ordinary differential equations (BVODEs) - see, e.g., [3]. Results for 1D parabolic PDEs are discussed in [11] and [18]. Results for BVODEs are discussed in [15]. We review these results later in the report.

This report provides, in Section 2, a description of BACOL and a more detailed description of the error estimation scheme currently employed by BACOL as well as brief descriptions of the alternative approaches for spatial error estimation to be considered in this report. In Section 3 we discuss in detail the development of the alternative spatial error estimation based on a superconvergent interpolant to replace the higher order collocation solution in the spatial error estimate. Section 4 considers the details associated with the second spatial error estimation approach based on the development of an interpolant (whose interpolation error is asymptotically equivalent to the error of the lower order collocation solution) that can replace the lower order collocation solution in the spatial error estimate. Selected numerical results are provided in Section 5 to demonstrate the effectiveness of the new algorithms and in Section 6 we discuss the costs of the error estimation schemes. We close, in Section 7, with our conclusions and an indication of areas for future work. The Appendix provides further detailed numerical results for several test problems.

The new error estimation approaches are much less expensive to compute that the current BACOL spatial error estimate. The numerical results show that the two new approaches generally provide spatial error estimates that are of comparable accuracy to those currently computed by BACOL, over a range of tolerances and test problems.

2 Overview of BACOL and the Spatial Error Estimation Schemes

2.1 Overview of BACOL

Assuming a spatial mesh $a = x_0 < x_1 < \cdots < x_{NINT} = b$, the approximate solution is a piecewise polynomial, of degree p on each subinterval, represented in BACOL as a linear combination of C^1-continuous B-spline basis functions [14] 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 degree p 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

$$\underline{U}(x,t) = \sum_{i=1}^{NC_p} \underline{y}_{p,i}(t) B_{p,i}(x), \tag{4}$$

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]) mapped onto each subinterval and by requiring the approximate solution to satisfy the boundary conditions. The collocation conditions have the form

$$\frac{d}{dt}\underline{U}(\xi_l,t) = \underline{f}\left(t,\xi_l,\underline{U}(\xi_l,t),\underline{U}_x(\xi_l,t),\underline{U}_{xx}(\xi_l,t)\right), \tag{5}$$

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

$$\begin{aligned} \xi_l \quad &= \quad x_{i-1} + h_i\rho_j, \quad \text{where } \; l = 1 + (i-1)(p-1) + j, \\ &\qquad \text{for } \; i = 1,\dots,NINT, \quad j = 1,\dots,p-1, \end{aligned} \tag{6}$$

where $\{\rho_i\}_{i=1}^{p-1}$ are the set of $p-1$ Gauss points on $[0,1]$. The collocation 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,

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

The above collocation conditions represent a system of ODEs (in time) whose solution components are the unknown 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 a modified version of 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 (4), to obtain values of the approximate solution at desired x values, for the current time t.

2.2 The BACOL Spatial Error Estimation Scheme

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

$$\underline{\bar{U}}(x,t) = \sum_{i=1}^{NC_{p+1}} \underline{y}_{p+1,i}(t)B_{p+1,i}(x). \tag{7}$$

This approximate solution is based on a set of C^1 continuous B-spline basis polynomials, $B_{p+1,i}(x)$, 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 $\underline{\bar{U}}(x,t)$. The collocation points in this case are the images of the p Gauss points on $[0,1]$ mapped onto each subinterval. As before, this leads to a systems of DAEs whose solution gives the functions, $\underline{y}_{p+1,i}(t)$. In order to ensure that the two approximate solutions, $\underline{U}(x,t)$ and $\underline{\bar{U}}(x,t)$ 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. The most expensive part of the DASSL computation involves the linear algebra computations and BACOL employs a slightly modified version of DASSL that treats the linear algebra computations associated with each system of DAEs separately and that takes into account the special almost block diagonal structure of the linear systems that arise. See [40] for further details.

As mentioned earlier, it is shown in [18], [11], that the collocation solution of degree p, $\underline{U}(x,t)$, has an error that is $O(h^{p+1})$, where h is the mesh spacing. We will say that the collocation solution is of order $p+1$. Then we have

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

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, we compute *a posteriori* spatial error estimates obtained by comparing $\underline{U}(x,t)$ and $\underline{\bar{U}}(x,t)$ at time t, as follows. We will denote the sth component of $\underline{U}(x,t)$ by $U_s(x,t)$ and the sth component of $\underline{\bar{U}}(x,t)$ by $\bar{U}_s(x,t)$. And we will denote by $ATOL_s$ and $RTOL_s$ the absolute and relative tolerances for the s-th component of the error estimate. BACOL computes a set of $NPDE$ normalized error estimates over the whole spatial domain of the form

$$E_s(t) = \sqrt{\int_a^b \left(\frac{U_s(x,t) - \bar{U}_s(x,t)}{ATOL_s + RTOL_s|U_s(x,t)|} \right)^2 dx}, \quad s = 1, \ldots, NPDE. \tag{8}$$

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

$$\hat{E}_i(t) = \sqrt{\sum_{s=1}^{NPDE} \int_{x_{i-1}}^{x_i} \left(\frac{U_s(x,t) - \bar{U}_s(x,t)}{ATOL_s + RTOL_s|U_s(x,t)|} \right)^2 dx}, \quad i = 1, \ldots, NINT. \tag{9}$$

Note that $E_s(t), s = 1, \ldots, NPDE$ and $\hat{E}_i(t), i = 1, \ldots, NINT$ are estimates of the error associated with the *lower* order solution component, $U_s(x,t)$, and that $\bar{U}_s(x,t)$ is computed only to provide these error estimates.

The conditions, $E_s(t), s = 1, \ldots, NPDE$ are used to determine the acceptability of $U_s(x,t)$; this approximate solution is accepted at the current time t if

$$\max_{1 \le s \le NPDE} E_s(t) \le 1. \tag{10}$$

This condition assesses whether the approximate solution components, $U_s(x,t)$, satisfy the user tolerances over the entire spatial domain at the current time, t. If (10) is satisfied, then the code attempts to take the next time step. Otherwise, the error estimates, $\hat{E}_i(t), i = 1, \ldots, NINT$, are used to allow an assessment of the distribution of the error estimate over the spatial domain. Based on this second set of error estimates, BACOL attempts to construct a new mesh that (i) has as many mesh points as necessary to yield an approximate solution whose estimated error satisfies the user tolerances and (ii) approximately equidistributes the estimated error over the subintervals of the new mesh. See [39, 40] for further details. Once a new mesh is determined, the time integration will require current and past solution information associated with the new mesh points for both the lower order and higher order collocation solutions. In BACOL this information for both collocation solutions is obtained through high order interpolation of the original *higher order* solution information associated with the previous mesh. Once this interpolated information has been computed, *both* collocation solutions are propagated forward in time.

The additional computation of $\bar{U}(x,t)$ approximately doubles the overall cost of the computation. The goal of the current report is to therefore investigate alternative spatial error estimation strategies that provide error estimates of comparable accuracy to those currently computed by BACOL but at a substantially lower cost. In the remainder of this section we will briefly outline the alternative error estimation strategies to be explored in this report.

2.3 Superconvergent Interpolant(SCI) based Spatial Error Estimation

In this alternative error estimation scheme, we will replace the computation of the higher order global collocation solution, $\underline{\bar{U}}(x,t)$, with the computation of a local interpolant that will be shown to give an approximation to the solution of the PDE that is of the same order of accuracy as $\underline{\bar{U}}(x,t)$, namely order $p+2$. That is, only the collocation solution, $\underline{U}(x,t)$, will be propagated forward in time and at the end of each time step, when DASSL returns with the B-spline basis function coefficient values, $\underline{y}_{p,i}(t)$, we will have available only the lower order solution $\underline{U}(x,t)$. And then based on this approximate solution, we will construct an interpolant of order $p+2$ that can replace $\underline{\bar{U}}(x,t)$ in the error estimates defined previously.

A general strategy for error estimation in numerical analysis, examples of which are Gauss-Kronrod formulas in numerical quadrature, see, e.g., [9], in numerical quadrature, and Runge-Kutta formula pairs, see, e.g., [22], in the numerical solution of initial value ODEs, is to perform an extra computation in order to obtain enough additional information to allow one to construct a higher order approximate solution which can then be compared with the lower order approximate solution to obtain an estimate of the error in the lower order approximate solution. 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, as mentioned earlier, 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 $p+1$, has *superconvergent* values of order $p+2$. Thus simply evaluating $\underline{U}(x,t)$ at these known special points provides higher order solution approximations. Then a (superconvergent) interpolant based on a sufficient number of these superconvergent values can be constructed and can replace $\underline{\bar{U}}(x,t)$ in the computation of the spatial error estimates (8), (9). We will investigate this approach in detail in Section 3.

2.4 Lower Order Interpolant(LOI) based Spatial Error Estimation

While the cost of computing the higher order collocation solution is slightly more expensive than that of the lower order collocation solution, the costs are in fact comparable and thus the overall costs for BACOL will be approximately halved if we compute only one of the two global collocation solutions. In the second alternative error estimation approach we consider, we will compute and propagate only the $p+2$ order collocation solution, $\underline{\bar{U}}(x,t)$, replacing the global $p+1$ order solution, $\underline{U}(x,t)$, with an interpolant of order $p+1$ that interpolates $\underline{\bar{U}}(x,t)$ at certain points on each subinterval. *The interpolation points are chosen so that the interpolation error term is asymptotically equivalent to the collocation error term for* $\underline{U}(x,t)$. This interpolant can then replace $\underline{U}(x,t)$ in the computation of the spatial error estimates. (The difference between $\underline{\bar{U}}(x,t)$ and this interpolant (asymptotically) gives the leading term in the error for this interpolant, which has been constructed so that this error term asymptotically approaches that of $\underline{U}(x,t)$.)

In this approach we are computing and propagating only the higher order collocation solution, $\underline{\bar{U}}(x,t)$, and, using the interpolant, we obtain a spatial error estimate for the *lower* order collocation solution. This idea is similar to the well known "local extrapolation" approach sometimes employed in software for the numerical solution of initial value ODEs, see, e.g, [22]. In that approach, a Runge-Kutta formula pair is used to compute two solutions, say of orders k and $k+1$. In standard mode, the lower order solution is propagated forward in time and the higher order solution is used only to obtain an estimate of the error in the lower order solution. Thus at the end of each time step, only the lower order solution is propagated and we have an estimate of the error only for that lower order solution. In local extrapolation mode, only the *higher* order solution is propagated but we only have an error estimate for the lower order solution. We investigate the LOI approach in more detail in Section 4.

2.5 Summary

The first of the alternative error estimation strategies described above propagates only $\underline{U}(x,t)$ and uses a higher order superconvergent interpolant to replace $\underline{\bar{U}}(x,t)$ in the error estimates for $\underline{U}(x,t)$. The second strategy propagates only $\underline{\bar{U}}(x,t)$ and uses a lower order interpolant in the error estimates for $\underline{U}(x,t)$. However, one could argue that if only $\underline{\bar{U}}(x,t)$ is to be propagated, it is preferable that an error estimate for $\underline{\bar{U}}(x,t)$ be computed. This suggests a third strategy in which only $\underline{\bar{U}}(x,t)$ is computed and propagated *but we compute an error estimate appropriate for* $\underline{\bar{U}}(x,t)$. This strategy can be implemented using the same approach as in the first alternative scheme (SCI) but for a propagated collocation solution of order $p+2$ rather than $p+1$. This means that the collocation solution of order $p+2$ is computed and that an error estimate appropriate for that solution is computed (based on a superconvergent interpolant of order $p+3$.) A new implementation is not required; the point here is that it might also be appropriate to also compare the standard BACOL error estimate with the SCI approach as described in Section 2.3 except at one higher order. That is, one might compare the standard BACOL approach that uses collocation solutions of order $p+1$ and $p+2$ with the SCI approach based on a collocation solution of order $p+2$ rather than the SCI approach described above, which is based on a collocation solution of order $p+1$. We will call this strategy SCI+.

This section has discussed four spatial error estimation strategies. For a given value of p, Table 1 summarizes the solutions that are propagated (i.e., computed using DASSL) and the error estimates that are provided.

Strategy	Solutions Propagated	Error Estimate
BACOL	Orders $p+1$, $p+2$	Order $p+1$
SCI	Order $p+1$	Order $p+1$
LOI	Order $p+2$	Order $p+1$
SCI+	Order $p+2$	Order $p+2$

Table 1: *Column 1 identifies the four error estimation schemes discussed in this section. Column 2 gives the orders of the propagated collocation solutions. Column 3 indicates for what order of solution we provide an error estimate.*

This report will experimentally compare the accuracy of the BACOL, SCI, and LOI error estimates. Future work will involve the efficient implementation of the SCI and LOI strategies within modified versions of BACOL and an experimental comparison of all four strategies with respect to efficiency.

3 The SCI Approach for Spatial Error Estimation

3.1 Experimental Verification of Superconvergence

The classic results of [11] and [18] provide the standard convergence results. These authors show that the collocation solution for (1) based on piecewise polynomials of degree p has an error that is order $p+1$ over the entire spatial domain, and that at the mesh points the order of accuracy improves to order $2(p-1)$. Thus the solutions approximations at the mesh points are superconvergent provided $2(p-1) > p+1 \Rightarrow p > 3$.

The above theory is consistent with that available from the boundary value ODE context: Let $u^{(m)}(x)$ be the mth derivative of $u(x)$ with respect to x. For the second order nonlinear boundary value ODE,

$$u^{(2)}(x) - f(x, u(x), u^{(1)}(x)) = 0, \quad a < x < b, \quad g(u(a), u(b)) = 0, \tag{11}$$

and with appropriate assumptions as indicated, Theorem 5.147 of [3], provides several results associated with applying a k-point Gaussian collocation method to obtain a numerical solution of this ODE. The results that are most relevant to the current study are the following. (See Part (c) of Theorem 5.147/Corollary 5.142 of [3].) Let $h_i = x_{i+1} - x_i$ and $h = \max_{i=0}^{NINT-1} h_i$. Let these collocation points be $\{\rho_r\}_{r=1}^{k}$.

(i) At the mesh points, the collocation error (associated with the collocation solution, $U(x)$ and its first derivative) satisfies

$$|u^{(j)}(x_i) - U^{(j)}(x_i)| = O(h^{2k}), \quad j = 0, 1, \quad i = 0, \ldots, NINT. \tag{12}$$

(ii) At non-mesh points, the collocation error (associated with the collocation solution, $U(x)$, and its derivatives up to order $k+1$) satisfies

$$u^{(j)}(x) - U^{(j)}(x) = \frac{1}{k!} h_i^{k+2-j} u^{(k+2)}(x_i) P_k^{(j)}\left(\frac{x - x_i}{h_i}\right) + O(h_i^{k+3-j}) + O(h^{2k}) = O(h^{k+2-j}), \tag{13}$$

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

$$P_k(\xi) = \int_0^{\xi} (t - \xi) \prod_{r=1}^{k} (t - \rho_r) dt. \tag{14}$$

Note that the number of collocation points in the results quoted for the PDE case is $p-1$ per subinterval and the number of collocation points for the BVODE results is k. We can thus see that these two sets of results are of course in agreement: (12) implies that the collocation error at the mesh points is $O(h^{2k}) = O(h^{2(p-1)})$; (13) implies that the collocation error overall is $O(h^{k+2}) = O(h^{p+1})$.

The BVODE result given in (ii) provides details of the coefficient of the leading term in the error and from that result it follows that one can expect, in the BVODE case, to see higher accuracy in the collocation solution at points within each subinterval that are roots of the polynomial $P(\xi)$ as long as $2k \geq k+3 \Rightarrow k \geq 3$. 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 $2k \geq k+2 \Rightarrow k \geq 2$.

To our knowledge, no corresponding result has been published for the PDE case but we will show experimentally in this section that the corresponding result for the PDE case does appear to be true. (We will also show that the standard convergence results for the PDE case can be observed experimentally.)

Before we provide the experimental convergence results, we list the polynomials, $P_k(x)$, for $k = 3, \ldots, 10$, and their derivatives.

- $k = 3$:

$$P_3(x) = -1/40\, x^2 (2x - 1)(x - 1)^2,$$
$$P_3'(x) = -1/20\, x (x - 1)\left(5x^2 - 5x + 1\right).$$

- $k = 4$:

$$P_4(x) = -\frac{1}{420} x^2 \left(14x^2 - 14x + 3\right)(x - 1)^2$$

$$P_4'(x) = -\frac{1}{70}\,x\,(x-1)\,(2\,x-1)\left(7\,x^2-7\,x+1\right)$$

- $k = 5$:

$$P_5(x) = -\frac{1}{504}\,x^2\,(2\,x-1)\left(6\,x^2-6\,x+1\right)(x-1)^2$$

$$P_5'(x) = -\frac{1}{252}\,x\,(x-1)\left(42\,x^4-84\,x^3+56\,x^2-14\,x+1\right)$$

- $k = 6$:(coefficients given to 40 digits)

$$P_6(x) = -0.01785714285714285714285714285714285714290\,x^2$$

$$(x-0.1526267046965671274476218874538658199935)(x-0.3747185964571342094966817555619445257435)$$
$$(x-0.6252814035428657905033182444380554739571)(x-0.8473732953034328725523781125461341816899)$$
$$(x-0.9999999999999999996252360743081052624740)(x-1.000000000000000000374763925691894736131)$$

$$P_6'(x) = -0.1428571428571428571428571428571428571432\,x$$

$$(x-0.08488805186071653506398389301626743020613)(x-0.2655756032646428930981140590456168353038)$$
$$(x-0.4999999999999999999999999999999999999253)(x-0.7344243967353571069018859409543831650129)$$
$$(x-0.9151119481392834649360161069837325693036)(x-1.000000000000000000000000000000000000238)$$

- $k = 7$:(coefficients given to 40 digits)

$$P_7(x) = -0.01388888888888888888888888888888888888890\,x^2$$
$$(x-0.1152723378341063364884525393761867036634)$$
$$(x-0.2895425974880942776345904263602843369771)(x-0.4999999999999999999999999999999999998259)$$
$$(x-0.7104574025119057223654095736397156625978)(x-0.8847276621658936635115474606238133051638)$$
$$(x-0.9999999999999999990752003460208259747023)(x-1.000000000000000000924799653979174017077)$$

$$P_7'(x) = -0.1250000000000000000000000000000000000001\,x$$
$$(x-0.06412992574519669233127711938966828094783)$$
$$(x-0.2041499092834288489277446343010234050334)(x-0.3953503910487605656156713698273243723071)$$
$$(x-0.6046496089512394343843286301726756275209)(x-0.7958500907165711510722553656989765969931)$$
$$(x-0.9358700742548033076687228806103317133009)(x-1.000000000000000000000000000000000003903)$$

- $k = 8$:(coefficients given to 40 digits)

$$P_8(x) = -0.01111111111111111111111111111111111111110\,x^2$$
$$(x-0.09007700226825652259199898430878326642404)(x-0.2296976813063206480102970162932299271127)$$
$$(x-0.4056612887546070310868762178406954413723)(x-0.5943387112453929689131237821593045592941)$$
$$(x-0.7703023186936793519897029837067700777414)(x-0.9099229977317434774080010156912166897463)$$
$$(x^2-2.000000000000000000000000000000000038318\,x+1.000000000000000000000000000000000040922)$$

$$P_8'(x) = -0.1111111111111111111111111111111111111110\,x$$
$$(x-0.05012100229426992134382737779083102097417)(x-0.1614068602446311232770572864543287746505)$$
$$(x-0.3184412680869109206446239656456703934133)(x-0.5000000000000000000000000000000000001767)$$
$$(x-0.6815587319130890793553760343543296082427)(x-0.8385931397553688767229427135456712151311)$$

$$(x - 0.9498789977057300786561726222091690000413)(x - 0.9999999999999999999999999999999999873783)$$

- $k = 9$:(coefficients given to 40 digits)

$$P_9(x) = -0.009090909090909090909090909090909090909090\,x^2$$
$$(x - 0.07229898685756271547112349189126108292455)$$
$$(x - 0.1863109301186906409987672996290522717257)(x - 0.334185223198605098895749824607638842101)$$
$$(x - 0.5000000000000000000000000000000000001427)(x - 0.6658147768013949011004250175392361314835)$$
$$(x - 0.8136890698813093590012327003709476172477)(x - 0.9277010131424372845288765081087393203328)$$
$$(x - 0.9999999999999999963034928919316276879745)(x - 1.000000000000000003696507108068372003954)$$

$$P_9'(x) = -0.09999999999999999999999999999999999999999\,x$$
$$(x - 0.04023304591677059308553366958883093292309)$$
$$(x - 0.1306130674472474624984469125700846374981)(x - 0.2610375250947777521694124536343710009786)$$
$$(x - 0.4173605211668064876868901170209132334105)(x - 0.5826394788331935123131098829790867710517)$$
$$(x - 0.7389624749052222478305875463656289715563)(x - 0.8693869325527525375015530874299154377089)$$
$$(x - 0.9597669540832294069144663304111689561914)(x - 1.000000000000000000000000000000000058677)$$

- $k = 10$:(coefficients given to 40 digits)

$$P_{10}(x) = -0.007575757575757575757575757575757575757580\,x^2$$
$$(x - 0.05929571219129399470948561789933304132527)(x - 0.1539696908715823006543332595028406359625)$$
$$(x - 0.2792835119457420839819822032670009617045)(x - 0.4241841678533666684918885686273490270090)$$
$$(x - 0.5758158321466333315081114313726510127730)(x - 0.7207164880542579160180177967329987841902)$$
$$(x - 0.8460303091284176993456667404971602510236)(x - 0.9407042878087060052905143821006647355001)$$
$$(x^2 - 2.000000000000000000000000000000001550508\,x + 1.000000000000000000000000000000001600330)$$

$$P_{10}'(x) = -0.09090909090909090909090909090909090909096\,x$$
$$(x - 0.03299928479597043283386293195030818272980)(x - 0.1077582631684277906887910919457709482478)$$
$$(x - 0.2173823365018974967645180152611241678509)(x - 0.3521209322065303042840442422204712448688)$$
$$(x - 0.5000000000000000000000000000000000096995)(x - 0.6478790677934696957159557577795287007190)$$
$$(x - 0.7826176634981025032354819847388759869321)(x - 0.8922417368315722093112089080542287897953)$$
$$(x - 0.9670007152040295671661370680496920925162)(x - 0.9999999999999999999999999999999998766375)$$

The roots of the above polynomials and their derivatives are the points at which superconvergent solution and derivative approximations can be computed. The following table shows these superconvergent solution and derivative evaluation points, to 27 digits. We list only the points that are internal to the interval, i.e., roots equal to zero or one are not listed.

Table 3 and Table 4 provide experimental evidence demonstrating that, for the spatial discretization of a 1D parabolic PDE by Gaussian collocation, the orders of convergence indicated by the above theory - point (ii) above for the BVODE case in particular - also hold for the PDE case. (We provide results only for the smaller k values, because for larger k values, for the given test problem, the solution values are so accurate that the corresponding errors approach the round-off error level.) We used the (non-spatially adaptive) EPDCOL package with a very high time tolerance (10^{-14}) so that we could choose the meshes that were used to obtain these numerical solutions and so that the temporal error would not interfere with the spatial error. We used the simple test problem (17). Errors over the entire spatial domain were computed using a continuous L2 norm;

mesh point errors and the errors at the roots of $P_k(\xi)$ and $P_k'(\xi)$ internal to each subinterval were computed using a discrete L2 norm.

From Table 3 and Table 4, we can see that the expected convergence rates for low k values for the simple PDE (17) are observed. (For very accurate solutions ($k = 4, NINT = 10$), we see a small degradation in the observed convergence rate due to the presence of round-off error.) These results suggest that the collocation solution and its derivative are superconvergent at the mesh points. They also suggest that the collocation solution is superconvergent at the points within each subinterval that correspond to roots of the $P_k(\xi)$ polynomial appearing in the error expression (15) and that the derivative of the collocation solution is superconvergent at the points within each subinterval that correspond to the roots of $P_k'(\xi)$.

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 $k = 5$.

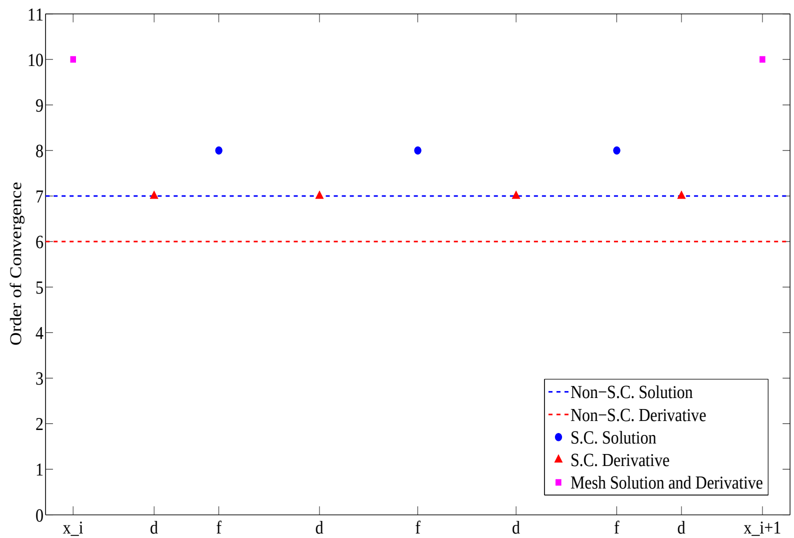


Figure 1: *Order of convergence of the collocation solution for $k = 5$ on one subinterval. The collocation solution over the entire subinterval is not superconvergent (Non-S.C. Solution) and has order 7, and its derivative, also not superconvergent over the entire subinterval (Non-S.C. Derivative) has order 6. The points labeled 'f' correspond to the roots of the polynomial $P_5(\xi)$ at which the collocation solution is superconvergent (S.C. Solution) and has order 8. The points labeled 'd' correspond to the roots of $P_5'(\xi)$ 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.*

3.2 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 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 $q > p$. For related work, see, e.g., [36], [21], [24, 25].) Recall that when the number of collocation points per subinterval is $p-1$, the collocation solution itself will have be order $p+1$ and its derivative will be order p. The interpolant can be based on the superconvergent solution and derivatives values which, as we have seen in the previous subsection, will be order $p+2$ and $p+1$, respectively. In order to have the interpolation error dominated by the data error we will therefore need to choose $p+3$ superconvergent values; this will yield an interpolation error that is 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 [20] for the form of the interpolant and its error term.

In order to obtain C^1-continuity, the SCI must interpolant the solution and derivative values at the endpoints of each 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 superconvergent values available on each subinterval in order to construct an interpolant of the desired order and continuity. 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 a mesh point solution and derivative value at each end of the subinterval.) We will interpolate at the meshpoint values but interpolation at additional non-mesh locations is required to obtain an interpolant of the desired order. However, all of our attempts to construct such interpolants led to existence issues when the remaining interpolation points are chosen within the current subinterval. (The paper [20] 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 interpolation points from those available on a given subinterval and have found that in each case this matrix is singular.) It appears to be impossible to construct an interpolant based on any subset of these 11 values; the coefficient matrix that defines the interpolant for any given combination of 9 of the available superconvergent values is singular. A similar statement can be made for other values of k.

In order to avoid this issue, the approach we have used employs, from each 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 the subintervals to the left and right of the 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. *Note that we always choose two of the solution values from outside the subinterval.*

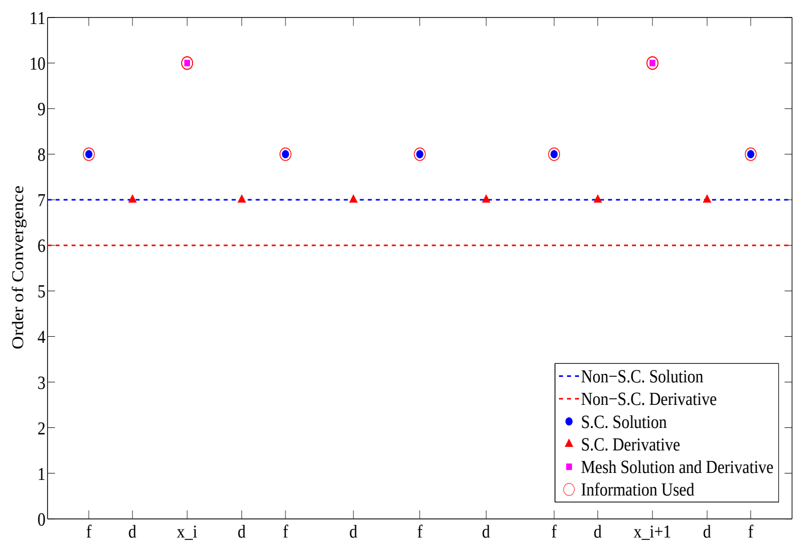


Figure 2: *Choice of superconvergent values for $k = 5$ case. The interpolants considered in this report use the superconvergent solution and derivative values at the endpoints of the subinterval, all the superconvergent solution values within the current subinterval, and the closest superconvergent solution value from each of the adjacent subintervals. This is 4 + 3 + 2 = 9 data values. These values are circled (Information Used).*

3.3 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 non-mesh points at which we will interpolate superconvergent values. (As explained earlier, $k-2$ of the w_j values are inside the subinterval and two of them are outside the subinterval.) Then, associated with the collocation solution, $\underline{U}(x,t)$, on the given subinterval, at time t, we have, from [20], the Hermite-Birkhoff SCI

$$\underline{\tilde{U}}(x,t) = \sum_{j=1}^{2} H_j(x)\underline{U}(s_j,t) + h\sum_{j=1}^{2} \overline{H}_j(x)\underline{U}'(s_j,t) + \sum_{j=1}^{k} G_j(x)\underline{U}(w_j,t),$$

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

$$H_j(x) = (1 - (x - s_j)\gamma_j)\frac{\xi_j^2(x)\phi(x)}{\xi_j^2(s_j)\phi(s_j)}, \quad \overline{H}_j(x) = (x - s_j)\frac{\xi_j^2(x)\phi(x)}{\xi_j^2(s_j)\phi(s_j)},$$

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

where

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

$$\xi(x) = \prod_{r=1}^{2}(x - s_r), \quad \xi_j(x) = \prod_{\substack{r=1\\ r\neq j}}^{2}(x - s_r),$$

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 [20] also provides an explicit expression for the interpolation error. Unfortunately, for the general case, the expression is somewhat 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

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

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

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

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 SCI estimates considered here. We will explore this issue experimentally later in this report.

In summary, once the primary collocation solution, $\underline{U}(x,t)$ is obtained, for the sth component of $\underline{U}(x,t)$, namely, $U_s(x,t)$, we will obtain superconvergent solution and derivative values and then construct a superconvergent interpolant for each subinterval based on these values. Because these interpolants interpolate both the solution and derivative approximations at the mesh points, the piecewise polynomial interpolant that is the union of the individual interpolants associated with each subinterval will have C^1 continuity over the spatial domain. As well, it will have an order of accuracy one higher than that of the primary collocation solution. We will then use the SCI to replace $\underline{\bar{U}}_s(x,t)$ in the error estimates (8) and (9).

4 The LOI Approach for Spatial Error Estimation

4.1 Requirements

As mentioned earlier, the key idea in this approach is propagate only the higher order global collocation solution, $\underline{\bar{U}}_s(x,t)$, and instead of computing the lower order global collocation solution, $\underline{U}(x,t)$, we will construct a lower order interpolant, based on evaluations of the higher order collocation solution, such that the leading order term in the error expansion for this interpolant approaches, asymptotically, the leading order term in the error expansion for the lower order collocation solution. It is this latter quantity that is currently used in BACOL for assessment of the global spatial error and for mesh refinement, and therefore it makes sense to focus on trying to approximate this quantity. While the details are different here, the general idea was considered earlier in the work by Moore on interpolation based error estimation mentioned earlier in this report.

Based on the experimental results reported earlier in the report, it appears that the lower order collocation solution has an error expansion that is a generalization of the corresponding result for the BVODE case. That is, it appears that the error expansion associated with the lower order collocation solution, $\underline{U}(x,t)$, has the form, for a given t and for the subinterval $[x_i, x_{i+1}]$,

$$u(x,t) - U(x,t) = \frac{1}{k!} h_i^{k+2} u^{(k+2)}(x_i,t) P_k\left(\frac{x-x_i}{h_i}\right) + O(h_i^{k+3}) + O(h^{2k}),$$

where

$$P_k(\xi) = \int_0^{\xi} (t-\xi) \prod_{r=1}^{k} (t-\rho_r) dt,$$

where k is the number of collocation points per subinterval, h is the maximum subinterval size, and the ρ_r values are the images of the k Gauss points on [0,1]. As explained earlier, $\bar{U}(x,t) - U(x,t)$ gives an approximation to the leading order term in the collocation error, given above. We want to construct an interpolant, $\tilde{U}(x,t)$, such that the leading order in the interpolation error for this interpolant equals the leading order term of the collocation error.

From standard interpolation theory, it is clear that the leading term in the error expansion for the interpolant will include the factors $h_i^{k+2} u^{(k+2)}(x_i,t)$ as long as we construct the interpolant so that it is based on $k+2$ data values and as long as the error in each of the data values is at least one order higher, i.e., at least $O(h_i^{k+3})$, so that the *interpolation* error dominates the *data* error. Since the data values we will employ for the interpolant are obtained from the higher order collocation solution, those values will have an error that is $O(h_i^{k+3})$ *for any choice of interpolation points.* The remaining factor in the leading term of the error expansion for the interpolant will depend on the location of the interpolation points within the interval and these points must be chosen so that this factor equals $P_k\left(\frac{x-x_i}{h_i}\right)$, so that the leading term in the interpolation error with agree with the leading term in the collocation error.

We now consider the case when $k=5$ but the approach will clearly generalize. For $k=5$, the leading term in the collocation error expansion is

$$\frac{1}{5!} h_i^7 u^{(7)}(x_i,t) P_5\left(\frac{x-x_i}{h_i}\right)$$

where

$$P_5(\xi) = \frac{-1}{540}\,\xi^2\,(2\,\xi-1)\left(6\xi^2-6\xi+1\right)(\xi-1)^2 = \frac{-1}{42}\,\xi^2\left(\xi-\frac{1}{2}\right)\left(\xi-\frac{1}{2}+\frac{1}{6}\sqrt{3}\right)\left(\xi-\frac{1}{2}-\frac{1}{6}\sqrt{3}\right)(\xi-1)^2.$$

The roots of this polynomial are $0, 0, \frac{1}{2}, \frac{1}{2} \pm \frac{1}{6}\sqrt{3}, 1, 1$. Substituting this expression for $P_5(\xi)$ into the general form for the leading term in the collocation error gives

$$u(x,t) - U(x,t) = \left(\frac{1}{5!}\right)\left(\frac{-1}{42}\right) h_i^7 u^{(7)}(x_i,t)\,\xi^2 \left(\xi - \frac{1}{2}\right)\left(\xi - \frac{1}{2} + \frac{1}{6}\sqrt{3}\right)\left(\xi - \frac{1}{2} - \frac{1}{6}\sqrt{3}\right)(\xi-1)^2 + O(h_i^8) + O(h^{10})$$

or

$$u(x,t) - U(x,t) = \left(\frac{-1}{7!}\right) h_i^7 u^{(7)}(x_i,t)\,\xi^2 \left(\xi - \frac{1}{2}\right)\left(\xi - \frac{1}{2} + \frac{1}{6}\sqrt{3}\right)\left(\xi - \frac{1}{2} - \frac{1}{6}\sqrt{3}\right)(\xi-1)^2 + O(h_i^8) + O(h^{10}) \tag{15}$$

We choose the interpolation points to be the roots of $P_5(\xi)$, with the understanding that the repeated roots (at 0 and 1) mean that the interpolant must interpolate the higher order collocation solution values at 0 and 1 and the derivative of the interpolant must interpolate the derivative of the higher order collocation solution at 0 and 1. Furthermore, the interpolant must interpolate the higher order collocation solution value at $\frac{1}{2}$ and $\frac{1}{2} \pm \frac{1}{6}\sqrt{3}$. This implies that the interpolant must be a Hermite-Birkhoff interpolant. In fact it is quite similar to those considered in the previous section except that we do not need to consider interpolation points outside the subinterval. That is, the interpolant we consider in this case is based on the same set of points identified in Figure 2 except that we do not choose the two points that are outside the current interval. Thus the total number of interpolation points is 7: interpolation of solution and derivative values at the endpoints of each subinterval and interpolation of solution values at the three interior points. (For general k, this is 4 endpoint solution and derivative values and $k-2$ internal solution values for a total of $k+2$ data values.)

4.2 The LOI

As explained above, the appropriate type of interpolant is a Hermite-Birkhoff interpolant and thus the same general form that is used for the SCI's of the previous section is relevant. Letting $s_1 = x_i$, $s_2 = x_{i+1}$ and let $w_j, j = 1, \ldots, k-2$, be the non-mesh points internal to the ith interval. (This gives a total of $4 + (k-2) = k+2$ interpolation points.) Then, associated with the collocation solution, $\underline{U}(x,t)$, on the given subinterval, at time t, we have, the Hermite-Birkhoff SCI

$$\underline{\tilde{U}}(x,t) = \sum_{j=1}^{2} H_j(x)\underline{U}(s_j,t) + h\sum_{j=1}^{2}\overline{H}_j(x)\underline{U}'(s_j,t) + \sum_{j=1}^{k-2} G_j(x)\underline{U}(w_j,t),$$

where $x \in [x_i, x_{i+1}]$, $h = x_{i+1} - x_i$, and the basis functions $H_j(x)$, $\overline{H}_j(x)$, and $G_j(x)$ are defined as in the previous section.

4.3 The Interpolation Error Term

The error expansion for this interpolant, for general k, is given in [20]. Since the expression is somewhat complicated, we do not reproduce it here. However the basic form of the leading term in the error expression for a given k (and a given t) has the general form

$$\frac{1}{(k+2)!} h_i^{k+2} u^{(k+2)}(x_i,t)\tilde{P}_k(\xi), \tag{16}$$

where $\tilde{P}_k(\xi)$ is a polynomial that depends on the points where interpolation of solution and derivative data is performed. Based on the choice of interpolation of solution and derivative data described above we can compute the specific polynomials $\tilde{P}_k(x)$ for $k = 3, \ldots, 10$:

$$\tilde{P}_3(x) = \left(x - \frac{1}{2}\right) x^2 (x-1)^2 .$$

$$\tilde{P}_4(x) = \left(x - \frac{1}{2} - \frac{1}{14}\sqrt{7}\right)\left(x - \frac{1}{2} + \frac{1}{14}\sqrt{7}\right) x^2 (x-1)^2 .$$

$$\tilde{P}_5(x) = \left(x - \frac{1}{2} - \frac{1}{6}\sqrt{3}\right)\left(x - \frac{1}{2}\right)\left(x - \frac{1}{2} + \frac{1}{6}\sqrt{3}\right) x^2 (x-1)^2$$

$$\tilde{P}_6(x) = (x - 0.15262670469656712744762188745386581999 35)(x - 0.37471859645713420949668175556194452574 35)$$

$$(x-0.62528140354286579050331824443805547395 71)(x-0.84737329530343287255237811254613418168 99)$$

$$x^2(x-1)^2.$$

$$\tilde{P}_7(x) = (x{-}0.11527233783410633648845253937618670366 34)(x{-}0.28954259748809427763459042636028433697 71)$$

$$(x{-}1/2)(x{-}0.71045740251190572236540957363971566259 78)(x{-}0.88472766216589366351154746062381330516 38)$$

$$x^2(x-1)^2.$$

$$\tilde{P}_8(x) = (x{-}0.09007700226825652259199898430878326642 404)(x{-}0.22969768130632064801029701629322992711 27)$$

$$(x-0.40566128875460703108687621784069544137 23)(x-0.59433871124539296891312378215930455929 41)$$

$$(x-0.77030231869367935198970298370677007774 14)(x-0.90992299773174347740800101569121668974 63)$$

$$x^2(x-1)^2.$$

$$\tilde{P}_9(x) = (x{-}0.07229898685756271547112349189126108292 455)(x{-}0.18631093011869064099876729962905227172 57)$$

$$(x{-}0.33418522319860509889957498246076388421 01)(x{-}1/2)(x{-}0.66581477680139490110042501753923613148 35)$$

$$(x-0.81368906988130935900123270037094761724 77)(x-0.92770101314243728452887650810873932033 28)$$

$$x^2(x-1)^2.$$

$$\tilde{P}_{10}(x) = (x{-}0.05929571219129399470948561789933304132 527)(x{-}0.15396969087158230065433325950284063596 25)$$

$$(x-0.27928351194574208398198220326700096170 45)(x-0.42418416785336666849188856862734902700 90)$$

$$(x-0.57581583214663333150811143137265101277 30)(x-0.72071648805425791601801779673299878419 02)$$

$$(x-0.84603030912841769934566674049716025102 36)(x-0.94070428780870600529051438210066473550 01)$$

$$x^2(x-1)^2.$$

It is clear from the above discussion that if the leading order term in the interpolation error is to equal the leading order term in the collocation error, then we must consider the corresponding polynomials arising in each error term, namely $P_k(\xi)$ and $\tilde{P}_k(\xi)$. We again consider the specific case of $k = 5$ but it will be clear how the approach generalizes for other values of k. With $k = 5$ the leading term in the collocation error is given in (15). The general form for the leading term in the interpolation error is given in (16); for the specific case of $k = 5$, this becomes

$$\frac{1}{7!} h_i^{k+2} u^{(k+2)}(x_i,t)\xi^2 \left(\xi - \frac{1}{2}\right)\left(\xi - \frac{1}{2} - \frac{1}{6}\sqrt{3}\right)\left(\xi - \frac{1}{2} + \frac{1}{6}\sqrt{3}\right)(\xi-1)^2,$$

and we see that the leading order term in the error expression for the collocation solution is equal to the leading order term in the error expression for the interpolant. For general k we have

$$||\tilde{U}(x,t) - \bar{U}(x,t)||_\infty = ||(\tilde{U}(x,t) - u(x,t)) - (\bar{U}(x,t) - u(x,t))||_\infty = ||\tilde{U}(x,t) - u(x,t)||_\infty + h_i^{k+3}$$

gives a computable approximation to the leading order term in the error for the collocation solution $U(x,t)$.

5 Selected Numerical Results and Discussion

5.1 Test Problems

(i) The simple test problem, [29]:

$$u_t = u_{xx} + \pi^2\sin(\pi x), \quad 0 < x < 1, \quad t > 0, \tag{17}$$

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., [39]:

$$u_t = \epsilon u_{xx} - uu_x, \qquad 0 < x < 1, \quad t > 0, \quad \epsilon > 0, \tag{18}$$

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. For $\varepsilon = 10^{-3}$, the solution to this problem is plotted in Figure 3.

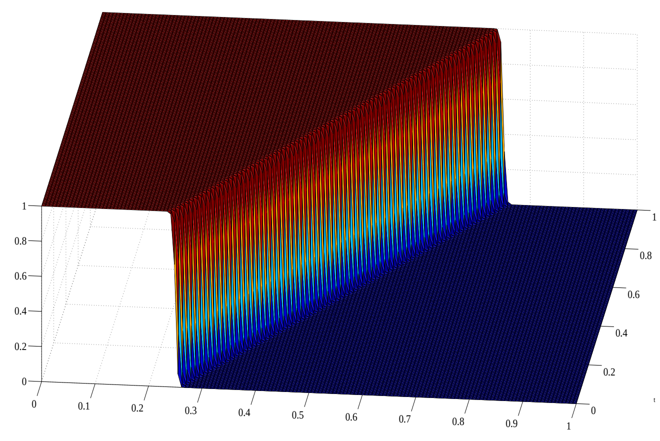


Figure 3: *Solution of Burgers' equation with $\varepsilon = 10^{-3}$. Initially, there is a sharp layer region approximately at the point 0.25 in the spatial domain. As t goes from 0 to 1, the layer moves to the right, reaching the point (approximately) 0.75 in the spatial domain at $t = 1$.*

5.2 Comparison of Accuracy of BACOL, SCI, LOI Error Estimates

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). The full set of graphs reporting the results of our testing over all test problems is given in the Appendix. Here we present selected results. For each problem, we integrate from $t = 0$ to $t = 1$. The initial spatial mesh size is $NINT_0 = 20$.

When BACOL is applied to the simple test problem (17), 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 4. The solution to the problem is smooth, the mesh is relatively uniform, and there are no issues associated with large adjacent subinterval ratios. Figure 4 is for the case where BACOL controls the mesh refinement but since the SCI and LOI error estimates are essentially the same, SCI or LOI control of the mesh would give similar results.

For the next set of results, we choose $k = 6$ and $tol = 10^{-6}$. The solution to the second test problem, (18) with $\epsilon = 10^{-3}$, has sharp layers - see Figure 3 - and BACOL employs a highly non-uniform mesh that adapts in time to efficiently solve this problem. A plot of the error estimates for each subinterval, as computed by the SCI, BACOL, and LOI error estimation schemes, as well as the true error for each subinterval, is given in Figure 5. *The BACOL error estimate controls the mesh,* which has 13 subintervals. 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 5, we see that the SCI estimate substantially overestimates the true error on the leftmost and rightmost subintervals. (From Figure 5 it is clear that the leftmost and rightmost subintervals are substantially larger than the adjacent subintervals.) Figure 6 focuses on the part of Figure 5 corresponding to the layer region. From this plot we can see that both the BACOL, SCI and LOI error estimates are in good agreement with each other but somewhat underestimate the true error in the middle of the layer region.

From the above results we see that there is good agreement between the BACOL and LOI error estimates but the large boundary subintervals are leading to issues for the SCI error estimate. In order to explore this issue

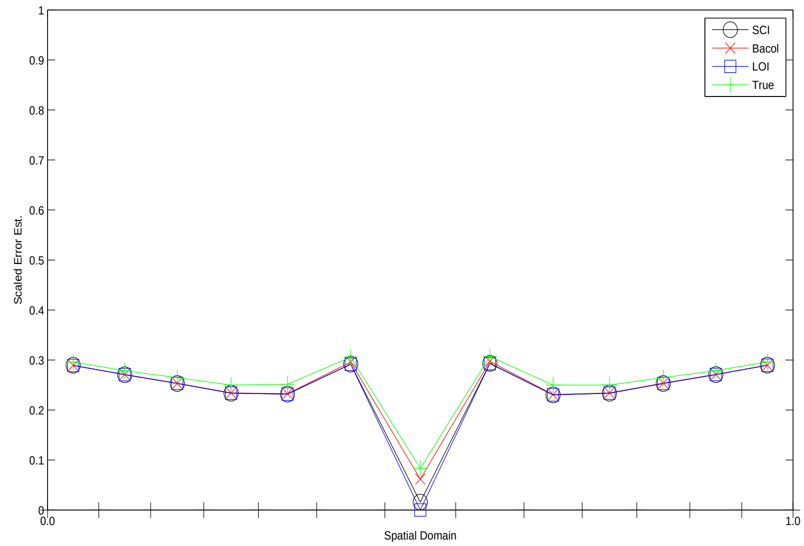


Figure 4: *Plot of the SCI error estimate, the BACOL error estimate, the LOI estimate and the true error associated with the collocation solution of* (17) *with* $k = 3$*,* $tol = 10^{-8}$*. BACOL controls the mesh refinement. The tick marks on the horizontal axis show the locations of the mesh points at the end of the computation.* $NINT = 13$*. All error estimates are in good agreement.*

further, the next set of results are obtained by again considering (18) with $\epsilon = 10^{-3}$ (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 error estimates and the true error is given in Figure 7. Comparing this plot with Figure 5, we note that all estimates show reasonably good agreement with the true error on all subintervals. The mesh again has $NINT = 13$. Comparing the meshes in Figure 5 and Figure 7 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 8 focuses on the part of Figure 7 corresponding to the layer region. Even within the layer region we see reasonably good agreement between the error estimates and the true error, when the SCI estimate controls the mesh refinement. The true error is slightly larger and it is better estimated by all estimates.

A larger set of results are reported in the Appendix; based on these results, we can make the following additional points:

(i) The above examples are generally representative of the performances of the three error estimates over all problems, $KCOL$ values, and tolerances considered. When the BACOL estimates control the mesh, the BACOL and LOI estimates are generally in good agreement and the SCI estimates are too large whenever adjacent subinterval sizes differ substantially.
(ii) For the $k = 3$ case, 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 $t = 0$. This is predicted by the theory: the super convergent points are only slightly more accurate and the initial mesh is relatively coarse. Thus the asymptotic results upon which the approach is based are not relevant.
(iii) The BACOL and SCI error estimators generally lead to meshes that have approximately the same number of subintervals.
(iv) Since the BACOL and SCI estimates lead to different meshes they also lead to a different number of remeshings over the course of a given computation. However, over all the numerical experiments we have considered we have generally not observed a significant difference in the number of remeshings associated with

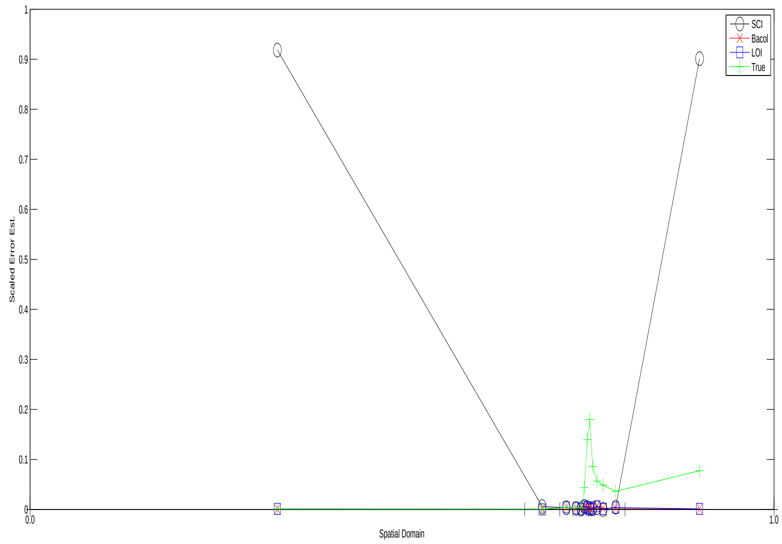


Figure 5: *Plot of the SCI error estimate, the BACOL error estimate, the LOI error estimate, and the true error associated with the collocation solution for* (18) *with* $\epsilon = 10^{-3}$. *BACOL controls the mesh refinement. The final mesh has* $NINT = 13$. *The tick marks on the horizontal axis show the locations of the mesh points at the end of the computation. Here we see that the SCI scheme substantially overestimates the error in the boundary subintervals and that all schemes underestimate the error in the layer region.*

either the BACOL or SCI estimation schemes.

6 Computational Costs for the BACOL, SCI, and LOI Error Estimates

As mentioned earlier, 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 solution 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, (BACOL uses the package COLROW [17, 16]), these costs are $O(NINT(NPDE \times k)^3)$. The cost of the secondary global solution is $O(NINT(NPDE \times (k+1))^3)$. Thus the additional computation of $\bar{U}(x,t)$ more than doubles the overall cost of the computation. 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 or the LOI on the other hand are relatively small. One must evaluate a collocation solution several times on each subinterval to obtain the collocation solution values and then the evaluation of the SCI or LOI involves only the evaluation of the basis polynomials associated with the interpolant, as 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. Because the meshes generated when the SCI error estimate is used are sometimes different from those generated from the BACOL error estimate, the number of remeshings that occur is also somewhat different. However, over all the experiments we considered, we generally did not observe a significant difference in the number of remeshings performed by either of the BACOL or SCI schemes.

It is thus clear that the cost of the SCI or LOI error estimates is a small fraction of the cost of error estimate

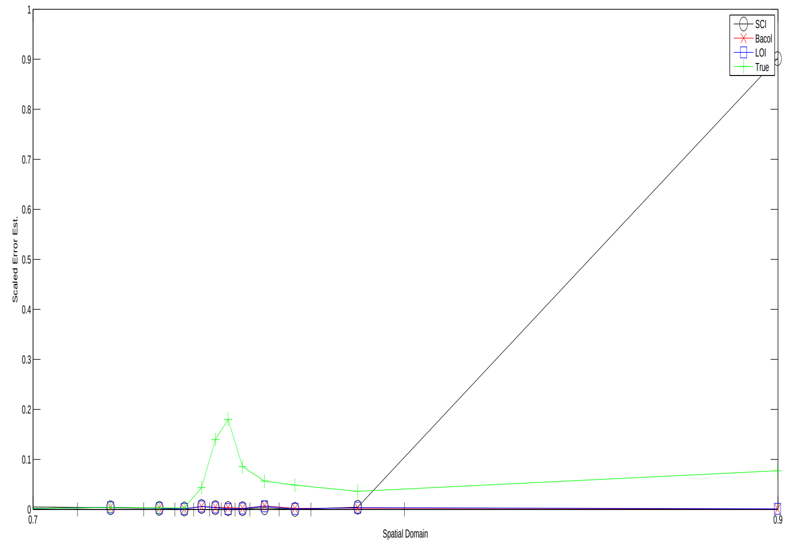


Figure 6: *Plot of the SCI error estimate, the BACOL error estimate, the LOI error estimate, and the true error associated with the collocation solution of* (18) *with* $\epsilon = 10^{-3}$*, within the layer region. BACOL controls the mesh refinement. The tick marks on the horizontal axis show the locations of the mesh points at the end of the computation. In the layer region all schemes underestimate the error.*

currently employed within BACOL. It is also clear that the memory requirements for new versions of BACOL based on either of the interpolation based schemes will be approximately 50% of what is required for the original BACOL code; this could be significant for very difficult problems where large spatial meshes are required.

7 Conclusions and Future Work

We have seen from the numerical results that the SCI and LOI error estimation schemes generally yield error estimates of comparable quality 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 they employ readily available collocation solution information the SCI and LOI estimates can be obtained at a relatively minor computational cost. This new approaches therefore appear to be interesting alternatives 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 the BACOL, SCI and LOI error estimates, and thus both the primary and secondary collocation solutions are still computed by this modified form of BACOL. Ongoing work involves the development of a new version of BACOL that will compute only the lower order collocation solution and the SCI error estimate, or only the higher order collocation solution and the LOI estimate. This will lead to new versions of BACOL that should have comparable performance to that of the current version but with about twice the speed. Furthermore the storage costs should be approximately halved. We will then be able to perform timing comparisons of the various implementations. Similar modifications to the BACOLR code can also be made (since it uses that same spatial discretization scheme as BACOL).

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 for the SCI case 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 [20]. Ongoing work also includes an analysis of possibilities for the representation of the SCI; this would include an exploration of which superconvergent values upon which to base the interpolant as well as the basis function representation for the interpolant. The

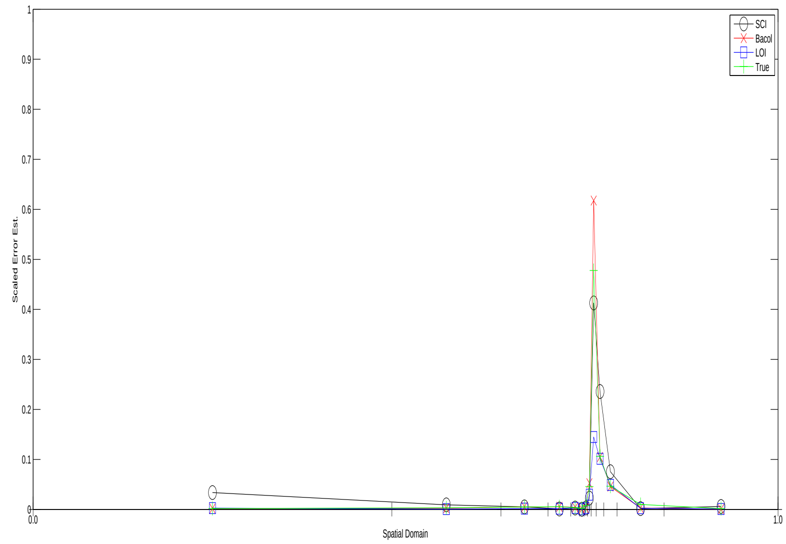


Figure 7: *Plot of the SCI error estimate, the BACOL error estimate, and the true error associated with the collocation solution of* (18) *with* $\epsilon = 10^{-3}$. *SCI controls the mesh refinement. The final mesh has* $NINT = 13$. *The tick marks on the horizontal axis show the locations of the mesh points at the end of the computation. Here we see that the SCI scheme is able to better adapt the mesh so that the SCI estimates are also in good agreement with the true error in the boundary subintervals. As well, all schemes do a better job of estimating the error in the layer region.*

work discussed in [12] on barycentric Hermite interpolants may be relevant.

From the experimental results, it is apparent that the SCI based error estimation approach self-corrects for the overestimates associated with large adjacent subinterval ratios and that sometimes this leads to the mesh having a few extra subintervals. An interesting question is whether it might be better to introduce some form of mesh smoothing to directly limit the adjacent subinterval ratios; this would also add a few extra subintervals to the mesh.

It is also worth noting that the spatial discretization scheme employed by BACOL is essentially identical to that employed by the widely used BVODE solver, COLSYS [2] and thus it would be interesting to explore the possibility of developing a modified version of COLSYS that would employ the approach described in this report, rather than the substantially more expensive approach based on Richardson extrapolation that it currently employs for the global error estimate used in its termination criterion.

It may be possible to generalize the approaches 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.

8 Acknowledgments

The authors would like to thank Jack Pew for his assistance in the preparation of this report and in particular for his work in the preparation of the final versions of many of the figures appearing in this report. The authors would also like to thank Zhi Li for his help in the verification of the numerical solution for one of the computations.

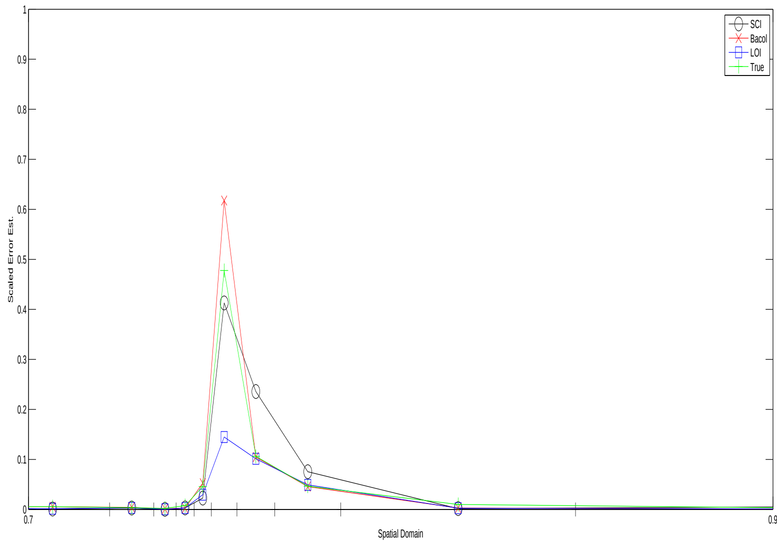


Figure 8: *Plot of the SCI error estimate, the BACOL error estimate, the LOI error estimate, and the true error associated with the collocation solution of* (18) *with* $\epsilon = 10^{-3}$*, within the layer region. SCI controls the mesh refinement. The tick marks on the horizontal axis show the locations of the mesh points at the end of the computation. All schemes do a better job of estimating the error in the layer region.*

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

[3] U. M. Ascher, R. M. M. Mattheij, and R. D. Russell. *Numerical solution of boundary value problems for ordinary differential equations*, volume 13 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1995.

[4] I. Babuška, M. Feistauer, and P. Šolín. On one approach to a posteriori error estimates for evolution problems solved by the method of lines. *Numer. Math.*, 89(2):225–256, 2001.

[5] I. Babuška and S. Ohnimus. A posteriori error estimation for the semidiscrete finite element method of parabolic differential equations. *Comput. Methods Appl. Mech. Engrg.*, 190(35-36):4691–4712, 2001.

[6] I. Babuška, J.R. Whiteman, and T. Strouboulis. *Finite Elements: An Introduction to the Method and Error Estimation*. Oxford University Press, New York, NY, 2011.

[7] 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):194–214, 1994.

[8] K. E. Brennan, S. L. Campbell, and L. R. Petzold. *Numerical solution of initial-value problems in differential-algebraic equations*, volume 14 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1996.

[9] D. Calvetti, G. H. Golub, W. B. Gragg, and L. Reichel. Computation of Gauss-Kronrod quadrature rules. *Math. Comp.*, 69(231):1035–1052, 2000.

[10] Yang Cao and Linda Petzold. A posteriori error estimation and global error control for ordinary differential equations by the adjoint method. *SIAM J. Sci. Comput.*, 26(2):359–374 (electronic), 2004.

[11] J. H. Cerutti and S. V. Parter. Collocation methods for parabolic partial differential equations in one space dimension. *Numer. Math.*, 26(3):227–254, 1976.

[12] R. M. Corless, A. Shakoori, D. A. Aruliah, and L. Gonzalez-Vega. Barycentric Hermite interpolants for event location in initial-value problems. *JNAIAM J. Numer. Anal. Ind. Appl. Math.*, 3(1-2):1–16, 2008.

[13] D03PPF. *NAG Fortran library.* The Numerical Algorithms Group, Ltd., Wilkinson House, Oxford, UK.

[14] C. de Boor. *A practical guide to splines*, volume 27 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 1978.

[15] Carl de Boor and Blâir Swartz. Collocation at Gaussian points. *SIAM J. Numer. Anal.*, 10:582–606, 1973.

[16] J. C. Díaz, G. Fairweather, and P. Keast. Algorithm 603. COLROW and ARCECO: FORTRAN packages for solving certain almost block diagonal linear systems by modified alternate row and column elimination. *ACM Trans. Math. Software*, 9(3):376–380, 1983.

[17] J. C. Díaz, G. Fairweather, and P. Keast. FORTRAN packages for solving certain almost block diagonal linear systems by modified alternate row and column elimination. *ACM Trans. Math. Software*, 9(3):358–375, 1983.

[18] J. Douglas, Jr. and T. Dupont. *Collocation methods for parabolic equations in a single space variable.* Lecture Notes in Mathematics, Vol. 385. Springer-Verlag, Berlin, 1974.

[19] D. J. Estep, M. G. Larson, and R. D. Williams. Estimating the error of numerical solutions of systems of reaction-diffusion equations. *Mem. Amer. Math. Soc.*, 146(696):viii+109, 2000.

[20] 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):1–15, 2008.

[21] 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):322–341, 1987.

[22] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving ordinary differential equations. I*, volume 8 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 1993.

[23] E. Hairer and G. Wanner. *Solving ordinary differential equations. II*, volume 14 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 1996.

[24] D. J. Higham. Robust defect control with Runge-Kutta schemes. *SIAM J. Numer. Anal.*, 26(5):1175–1183, 1989.

[25] D. J. Higham. Runge-Kutta defect control using Hermite-Birkhoff interpolation. *SIAM J. Sci. Statist. Comput.*, 12(5):991–999, 1991.

[26] W. Huang and R. D. Russell. A moving collocation method for solving time dependent partial differential equations. *Appl. Numer. Math.*, 20(1-2):101–116, 1996.

[27] P. Keast and P. H. Muir. Algorithm 688: Epdcol: a more efficient pdecol code. *ACM Trans. Math. Softw.*, 17(2):153–166, 1991.

[28] J. Lang and J. G. Verwer. On global error estimation and control for initial value problems. *SIAM J. Sci. Comput.*, 29(4):1460–1475 (electronic), 2007.

[29] N. K. Madsen and R. F. Sincovec. Algorithm 540: Pdecol, general collocation software for partial differential equations [d3]. *ACM Trans. Math. Softw.*, 5(3):326–351, 1979.

[30] 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):149–177, 2001.

[31] 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):367–401, 2003.

[32] P. K. Moore. Implicit interpolation error-based error estimation for reaction-diffusion equations in two space dimensions. *Comput. Methods Appl. Mech. Engrg.*, 192(39-40):4379–4401, 2003.

[33] P. K. Moore. An implicit interpolation error-based error estimation strategy for hp-adaptivity in one space dimension. *J. Numer. Math.*, 12(2):143–167, 2004.

[34] P. K. Moore. Interpolation error-based a posteriori error estimation for hp-refinement using first and second derivative jumps. *Appl. Numer. Math.*, 48(1):63–82, 2004.

[35] P. K. Moore. Effects of basis selection and h-refinement on error estimator reliability and solution efficiency for high-order methods in three space dimensions. *Int. J. Numer. Anal. Model.*, 3(1):21–51, 2006.

[36] L. F. Shampine. Interpolation for Runge-Kutta methods. *SIAM J. Numer. Anal.*, 22(5):1014–1027, 1985.

[37] T. Strouboulis, I. Babuška, and D. K. Datta. Guaranteed a posteriori error estimation for fully discrete solutions of parabolic problems. *Internat. J. Numer. Methods Engrg.*, 56(9):1243–1259, 2003.

[38] 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):454–470, 2004.

[39] R. Wang, P. Keast, and P. H. Muir. A comparison of adaptive software for 1D parabolic PDEs. *J. Comput. Appl. Math.*, 169(1):127–150, 2004.

[40] 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):239–260, 2004.

[41] 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):Art. 15, 28, 2008.

[42] W. Zhang. Diffusive effects on a catalytic surface reaction: an initial boundary value problem in reaction-diffusion-convection equations. *J. Bifur. Chaos*, 3:79–95, 1993.

k	Superconvergent Solution Points	Superconvergent Derivative Points
3	$\frac{1}{2}$	$\frac{1}{2} \pm \frac{1}{10}\sqrt{5}$
4	$\frac{1}{2} \pm \frac{1}{14}\sqrt{7}$	$\frac{1}{2} \pm \frac{1}{14}\sqrt{21},\ \frac{1}{2}$
5	$\frac{1}{2} \pm \frac{1}{6}\sqrt{3},\ \frac{1}{2}$	$\frac{1}{2} \pm \frac{1}{42}\sqrt{147 \pm 42\sqrt{7}}$
6	0.15262670469656712744762188 7	0.08488805186071653506398389 3
	0.37471859645713420949668175 5	0.26557560326464289309811405 9
	0.62528140354286579050331824 4	0.5
	0.84737329530343287255237811 2	0.73442439673535710690188594 0
		0.91511194813928346493601610 7
7	0.11527233783410633648845253 9	0.06412992574519669233127711 9
	0.28954259748809427763459042 6	0.20414990928342884892774463 4
	0.5	0.39535039104876056561567137 0
	0.71045740251190572236540957 4	0.60464960895123943438432863 0
	0.88472766216589366351154746 1	0.79585009071657115107225536 6
		0.93587007425480330766872288 1
8	0.09007700226825652259199898 4	0.05012100229426992134382737 8
	0.22969768130632064801029701 6	0.16140686024463112327705728 6
	0.40566128875460703108687621 8	0.31844126808691092064462396 6
	0.59433871124539296891312378 2	0.5
	0.77030231869367935198970298 4	0.68155873191308907935537603 4
	0.90992299773174347740800101 6	0.83859313975536887672294271 4
		0.94987899770573007865617262 2
9	0.07229898685756271547112349 2	0.04023304591677059308553367 0
	0.18631093011869064099876730 0	0.13061306744724746249844691 3
	0.33418522319860509889957498 2	0.26103752509477775216941245 4
	0.5	0.41736052116680648768689011 7
	0.66581477680139490110042501 8	0.58263947883319351231310988 3
	0.81368906988130935900123270 0	0.73896247490522224783058754 6
	0.92770101314243728452887650 8	0.86938693255275253750155308 7
		0.95976695408322940691446633 0
10	0.05929571219129399470948561 8	0.03299928479597043283386293 2
	0.15396969087158230065433326 0	0.10775826316842779068879109 2
	0.27928351194574208398198220 3	0.21738233650189749676451801 5
	0.42418416785336666849188856 7	0.35212093220653030428404424 2
	0.57581583214663333150811143 1	0.5
	0.72071648805425791601801779 7	0.64787906779346969571595575 8
	0.84603030912841769934566674 0	0.78261766349810250323548198 5
	0.94070428780870600529051438 2	0.89224173683157220931120890 8
		0.96700071520402956716613706 8

Table 2: *Superconvergent Solution and Derivative Points, i.e., roots of* P_k *and* P_k'

k	$NINT$	GE	Ratio	Rate	ME	Ratio	Rate
2	4	3.15×10^{-4}	-	-	3.86×10^{-4}	-	-
2	8	1.93×10^{-5}	16.29	4.03	2.36×10^{-5}	16.4	4.04
3	4	1.06×10^{-5}	-	-	8.48×10^{-7}	-	-
3	8	3.31×10^{-7}	32.13	5.01	1.30×10^{-8}	65.4	6.03
4	4	2.84×10^{-7}	-	-	8.30×10^{-10}	-	-
4	8	4.44×10^{-9}	64.06	6.00	3.15×10^{-12}	263.5	8.04

Table 3: *Experimental verification of expected convergence rates of collocation solutions for the simple test problem (17), where* k *is the number of collocation points per subinterval and* $NINT$ *is the number of subintervals of the (uniform) mesh. We report collocation solution errors across the entire problem domain (GE) and at the mesh points (ME). Expected convergence rates for GE are, respectively,* $4, 5, 6$, *and for ME are, respectively,* $4, 6, 8$.

k	$NINT$	Solution	Ratio	Rate	Derivative	Ratio	Rate
3	10	7.96×10^{-9}	-	-	2.02×10^{-7}	-	-
3	20	1.26×10^{-10}	63.2	5.98	6.38×10^{-9}	31.7	4.99
4	10	4.42×10^{-11}	-	-	1.85×10^{-9}	-	-
4	20	3.55×10^{-13}	124.5	6.96	2.94×10^{-11}	62.9	5.97
5	5	8.71×10^{-11}	-	-	1.90×10^{-9}	-	-
4	10	3.78×10^{-13}	230.4	7.85	1.47×10^{-11}	129.3	7.01

Table 4: *Experimental verification of expected convergence rates for the collocation solution and its derivative for the simple test problem (17), where the number of collocation points is* k *and* $NINT$ *is the number of subintervals of the (uniform) mesh. We report errors at the images of the roots of the polynomial* $P_k(\xi)$ *(15) on each subinterval (for the collocation solution, in the column titled "Solution") and at images of the roots of* $P_k'(\xi)$ *(for the derivative of the collocation solution, in the column titled "Derivative"). Expected convergence rates for* $k = 3, 4, 5$ *are, respectively,* $6, 7, 8$, *for the collocation solution, and* $5, 6, 7$, *for its derivative.*

9 Appendix

In this appendix, we provide detailed numerical results comparing the error estimates obtained from the SCI and LOI schemes with the original error estimation scheme used in BACOL. Where exact solutions are available, we also compare the estimates with the exact error. We consider a number of test problems: (i) Burgers' Equation (1st Burgers' Equation) with $\epsilon = 10^{-3}$, (ii) Burgers' Equation (1st Burgers' Equation) with $\epsilon = 10^{-4}$, (iii) Burgers' Equation (2nd Burgers' Equation) with $\epsilon = 10^{-3}$, (iv) Burgers' Equation (2nd Burgers' Equation) with $\epsilon = 10^{-4}$, (v) the Cahn-Allen Equation with $\epsilon = 10^{-6}$, (vi) the problem $u_t = (e^{5u}u_x)_x$, with appropriate boundary and initial conditions, and (vii) a catalytic surface reaction system. For each problem 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). In each case the initial spatial mesh size is $NINT_0 = 20$.

9.1 1st Burgers' Equation

9.1.1 Problem

$$u_t = \epsilon u_{xx} - uu_x, \qquad 0 < x < 1, \quad t > 0$$

Initial Condition:

$$u(x,0) = \frac{1}{2} - \frac{1}{2}\tanh\left(\frac{1}{4\epsilon}\left(x - \frac{1}{4}\right)\right), \quad t \geq 0$$

Boundary Conditions:

$$u(0,t) = \frac{1}{2} - \frac{1}{2}\tanh\left(\frac{1}{4\epsilon}\left(-\frac{t}{2} - \frac{1}{4}\right)\right), \quad t \geq 0$$

$$u(1,t) = \frac{1}{2} - \frac{1}{2}\tanh\left(\frac{1}{4\epsilon}\left(\frac{3}{4} - \frac{t}{2}\right)\right), \quad t \geq 0$$

Exact Solution:

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

The following six subsubsections provide results for $\epsilon = 10^{-3}$ and tolerances of 10^{-4}, 10^{-6}, and 10^{-8}, and $\epsilon = 10^{-4}$ and tolerances of 10^{-4}, 10^{-6}, and 10^{-8}. Within each subsection, we provide results for $k = KCOL = 3, 6$, and 9. The title of each graph indicates: the problem (1st Burgers), the value of ϵ (Eps $= 10^{-3}$ or Eps $= 10^{-4}$), which of the error estimation schemes controls the mesh refinement (Bac. Controlled or SCI Controlled), the value of KCOL, the tolerance TOL, the final Time (close to 1.0), the number of subintervals in the final mesh, NINT, and the total number of spatial remeshings that took place over the entire computation, Remeshes.

9.1.2 Results for $\epsilon = 10^{-3}$ and Tolerance $= 10^{-4}$

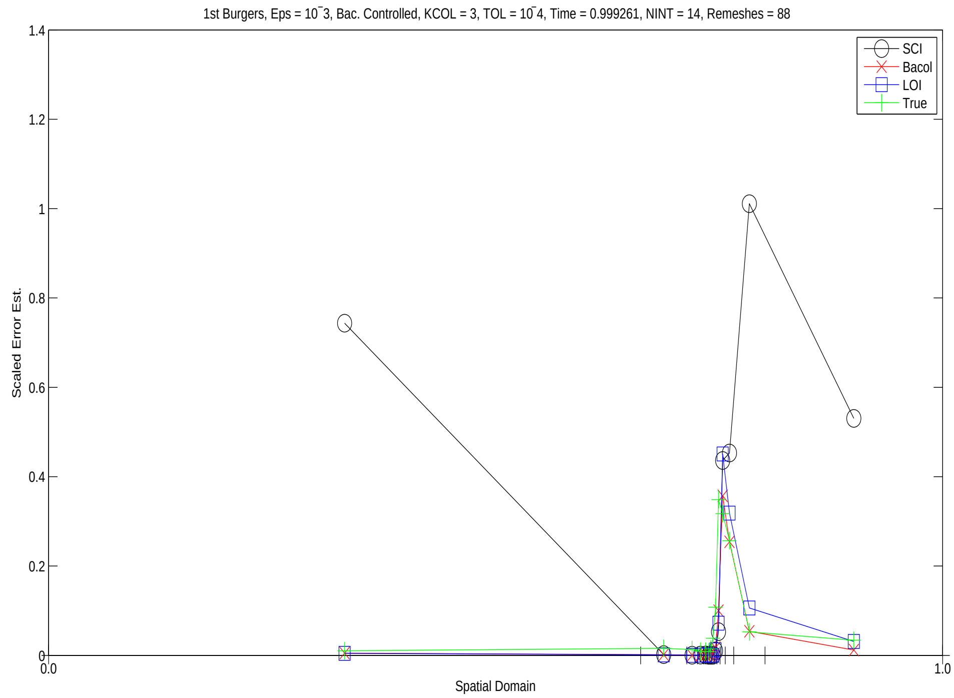
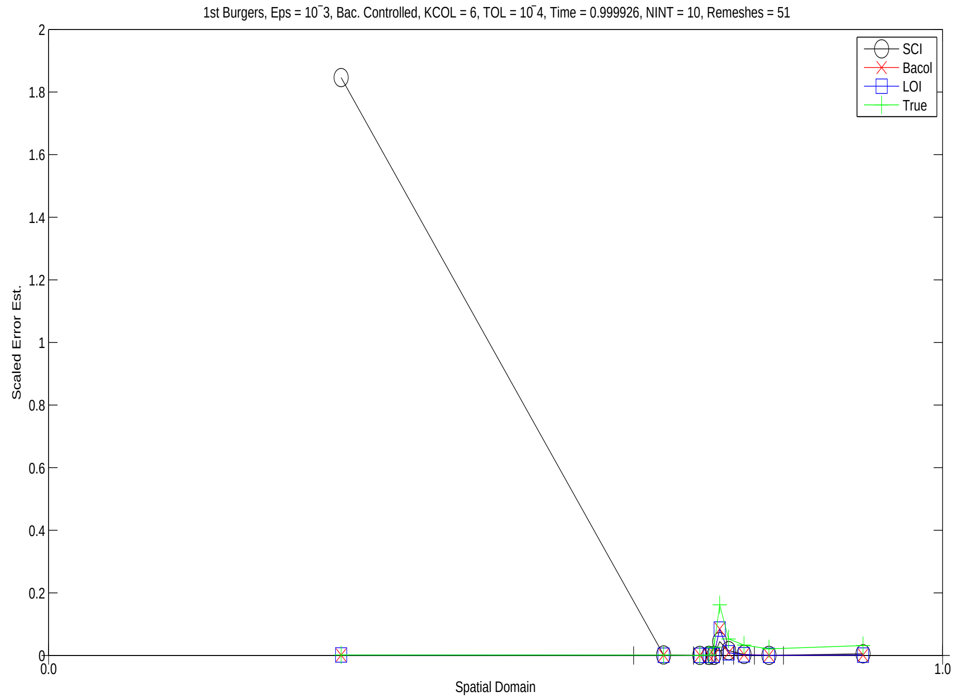


Figure 9: *All estimation schemes are in good agreement in the layer region, when BACOL controls the mesh, although, as expected, the SCI scheme greatly overestimates the error where subinterval ratios are large, particularly at the boundary subintervals. When the SCI scheme controls mesh refinement, the time integration fails at the beginning of the integration even with $NINT_0$ increased from 20 to 500.*



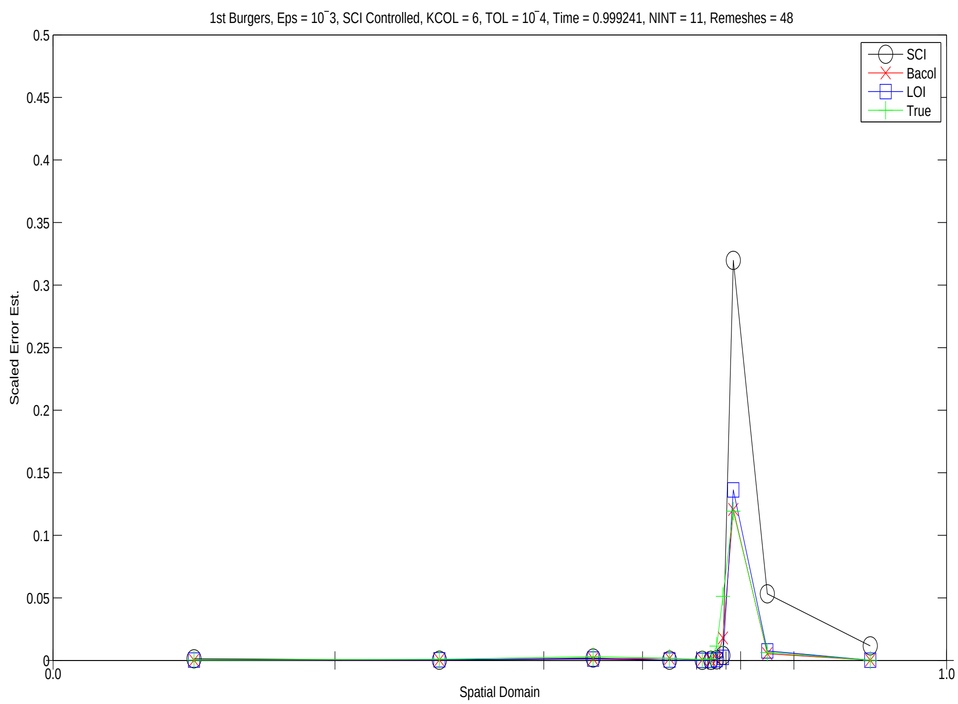
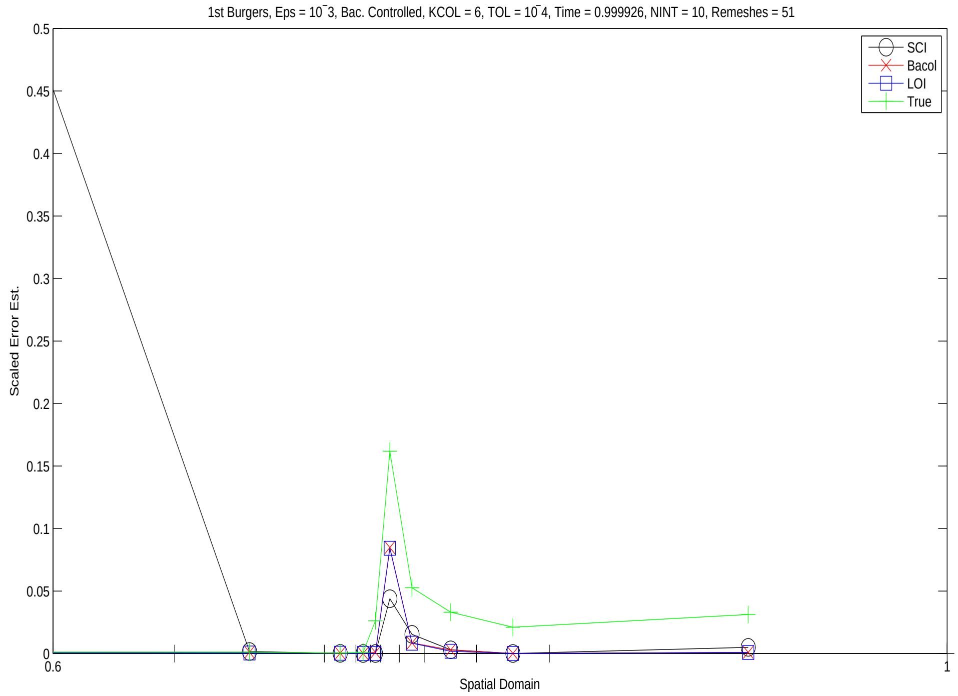


Figure 10: *There SCI estimates for the boundary subintervals overestimate the error when the BACOL estimates control the mesh. Note the self-correcting of the SCI scheme for its overestimation of boundary subinterval errors when the SCI estimates control the mesh. Issues with the error estimates in the layer region in either case.*



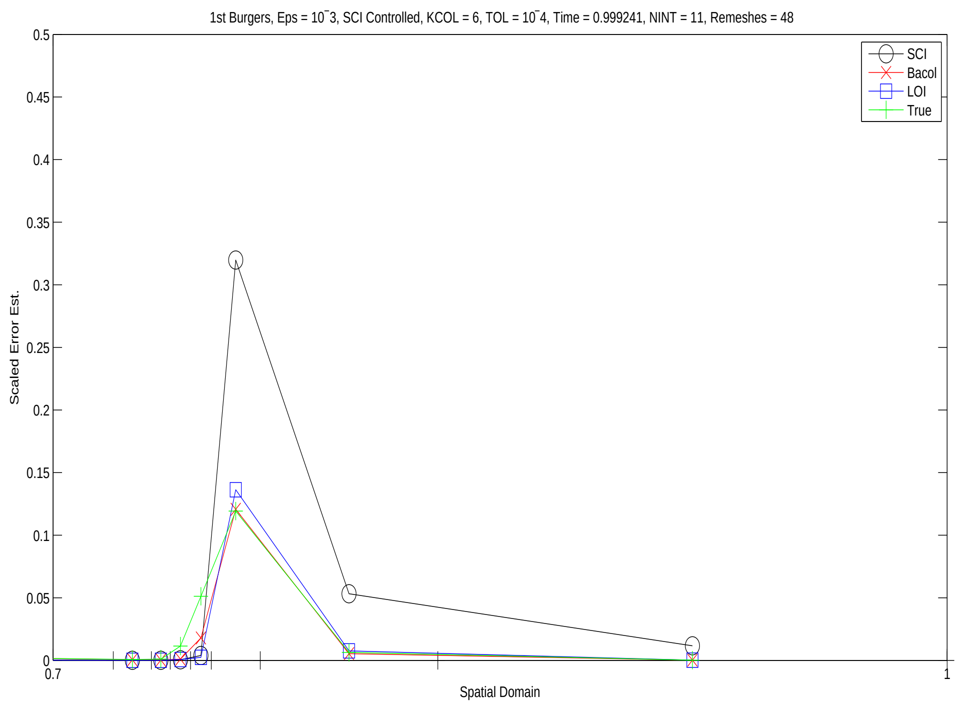


Figure 11: *Results only for the layer region. For the BACOL controlled case, all three schemes underestimate the error in the layer region, particularly the SCI scheme. When the SCI scheme is in control, it distributes the mesh such that there are slightly fewer subintervals inside the layer region and more toward the boundaries. In this case, that creates some larger subinterval ratios inside the layer region and hence the SCI's overestimation of errors there. The LOI and BACOL schemes are in good agreement with each other and the true error.*

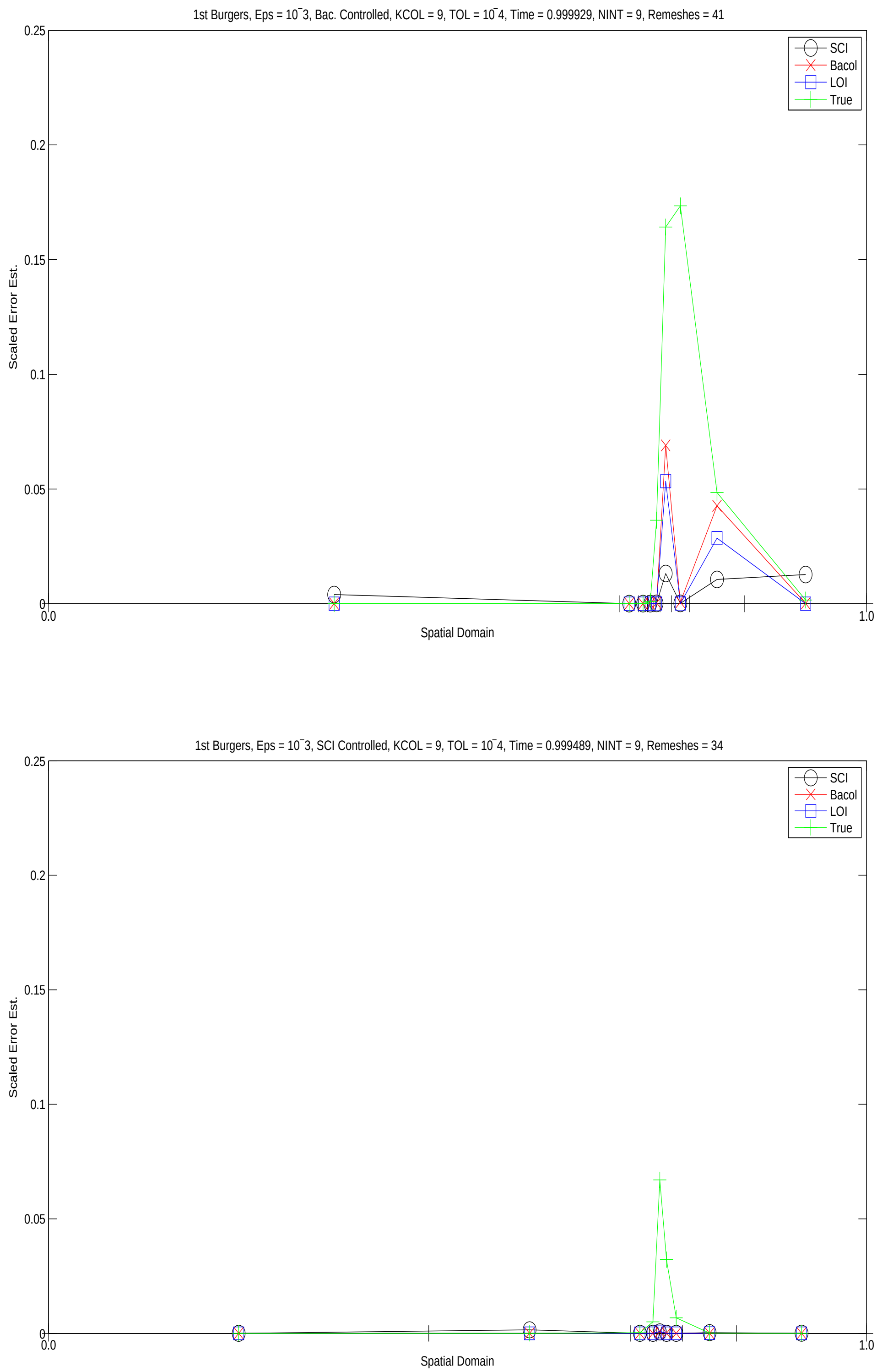
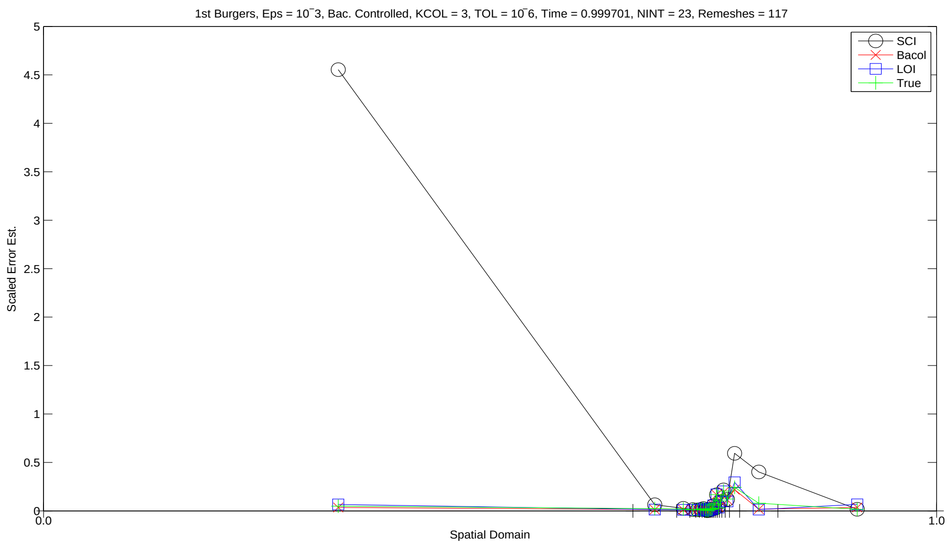


Figure 12: *All three schemes are generally in good agreement but all three underestimate the error in the layer region.*

9.1.3 Results for $\epsilon = 10^{-3}$ and Tolerance $= 10^{-6}$



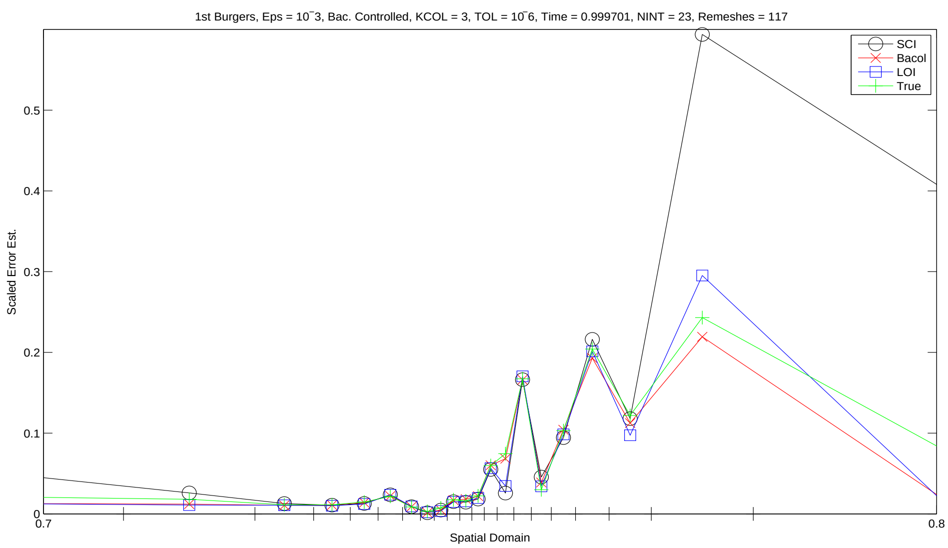
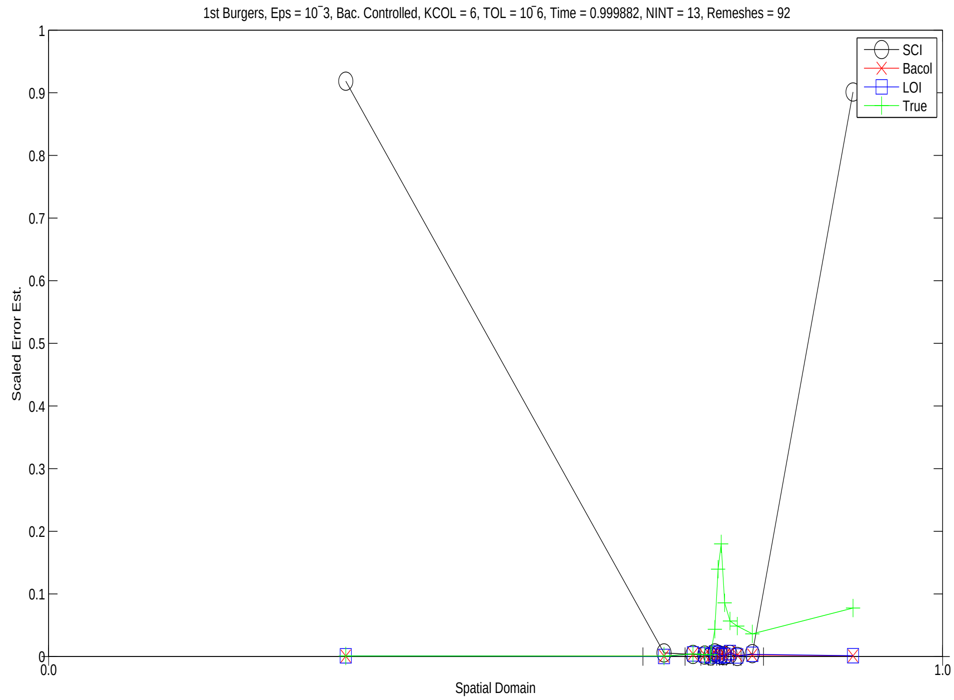


Figure 13: *In the top graph, with BACOL controlling the mesh, the SCI scheme vastly overestimates the error for the boundary subintervals. In the bottom graph, again with BACOL controlling the mesh, it can be seen that there is remarkable agreement between the error estimation schemes inside the layer region. With the SCI scheme in control, the time integration fails at the beginning of the integration.*



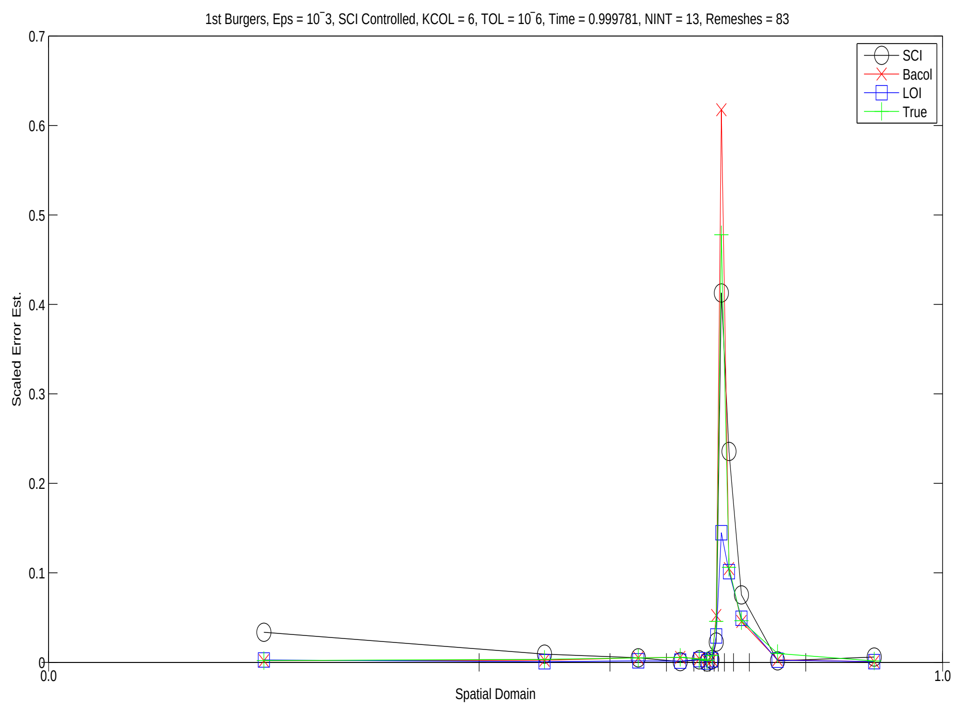
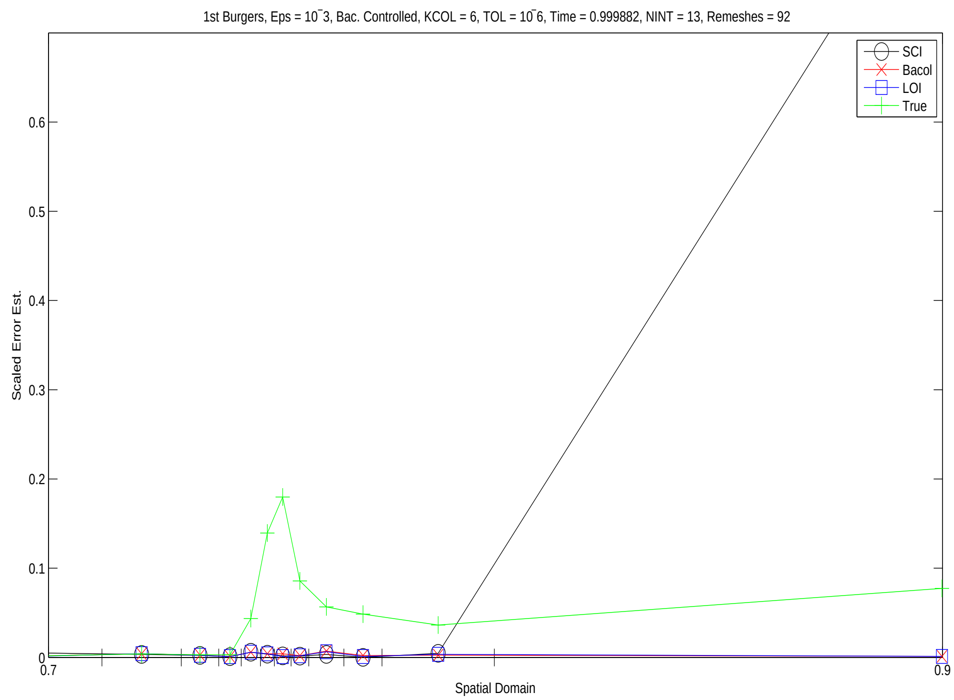


Figure 14: *When the BACOL error estimate controls the mesh, the SCI estimates for the boundary subintervals are poor and none of the schemes do a good job of estimating the error in the layer. When the SCI estimates control the mesh, there is some self-correction, and the SCI estimates are much better. All schemes do a better job of estimating the error in the layer region.*



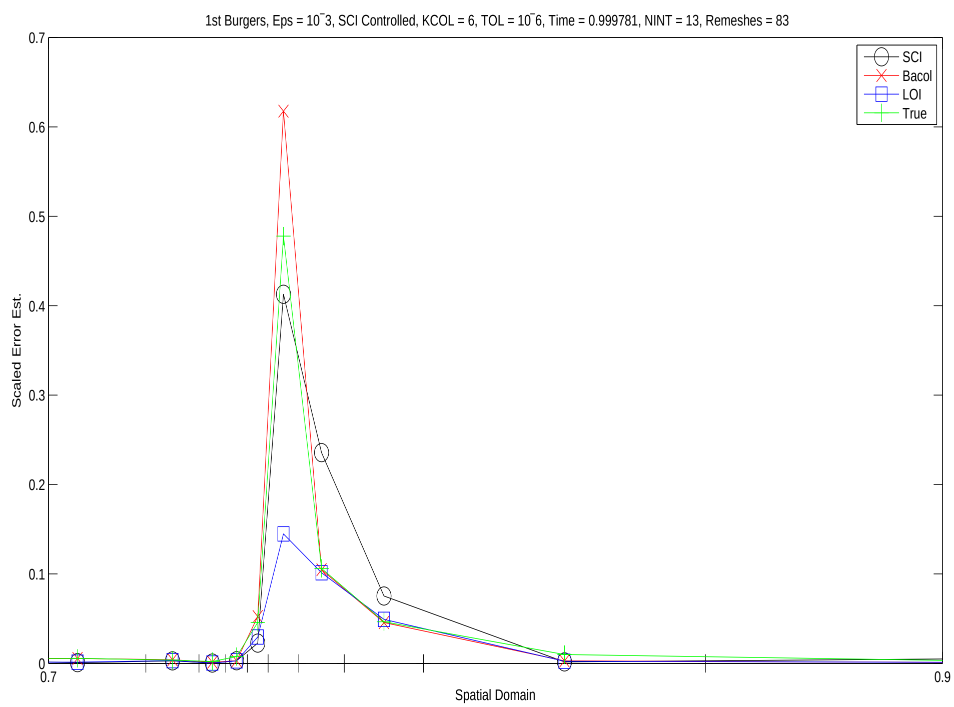


Figure 15: *Results only for the layer region. All schemes perform poorly with the BACOL scheme controlling the mesh, but fairly well in the case where the SCI scheme controls the mesh.*

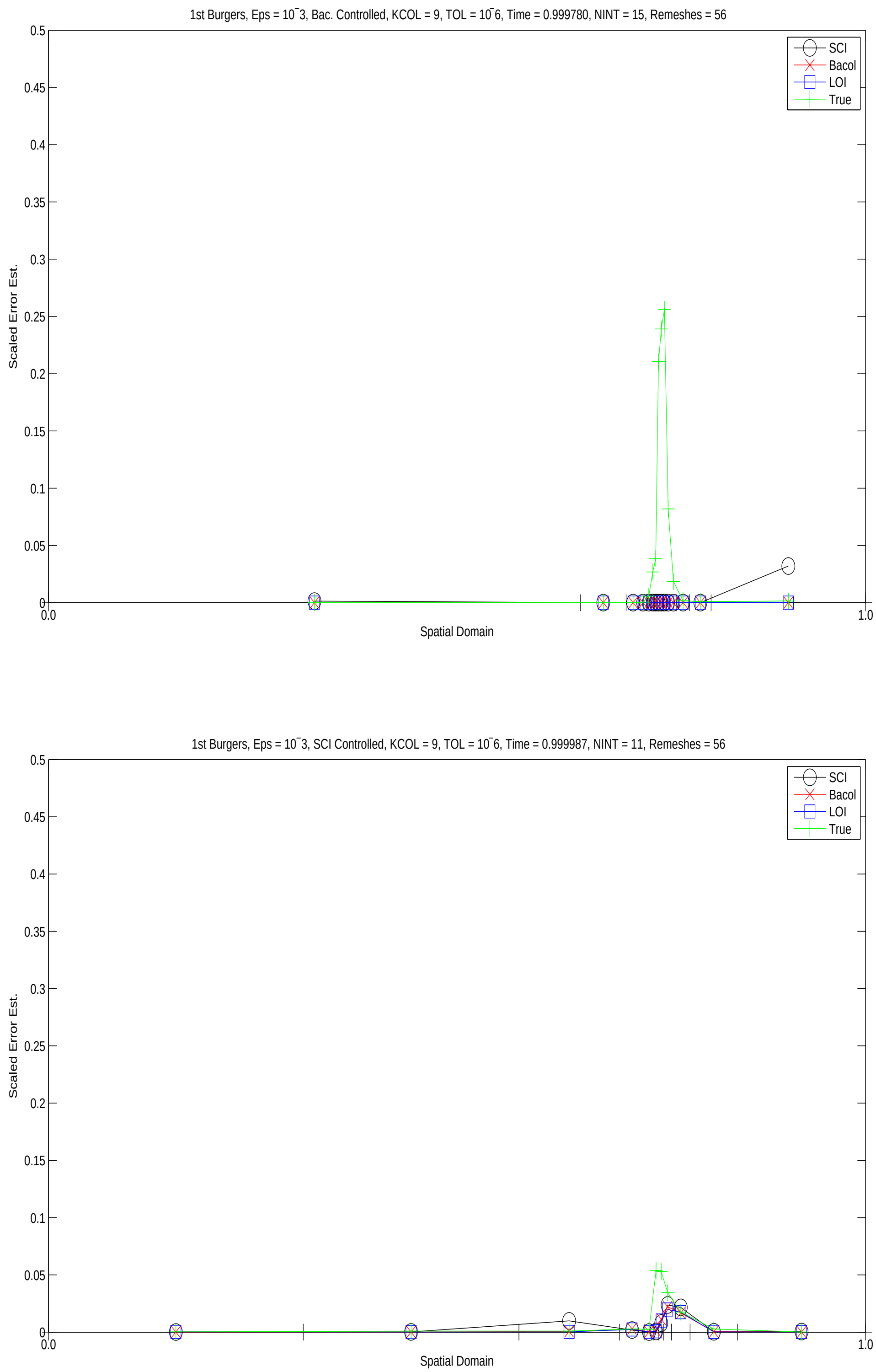


Figure 16: *Outside the layer region, all error estimates generally do a good job. There are issues with all estimation schemes in the layer regions.*

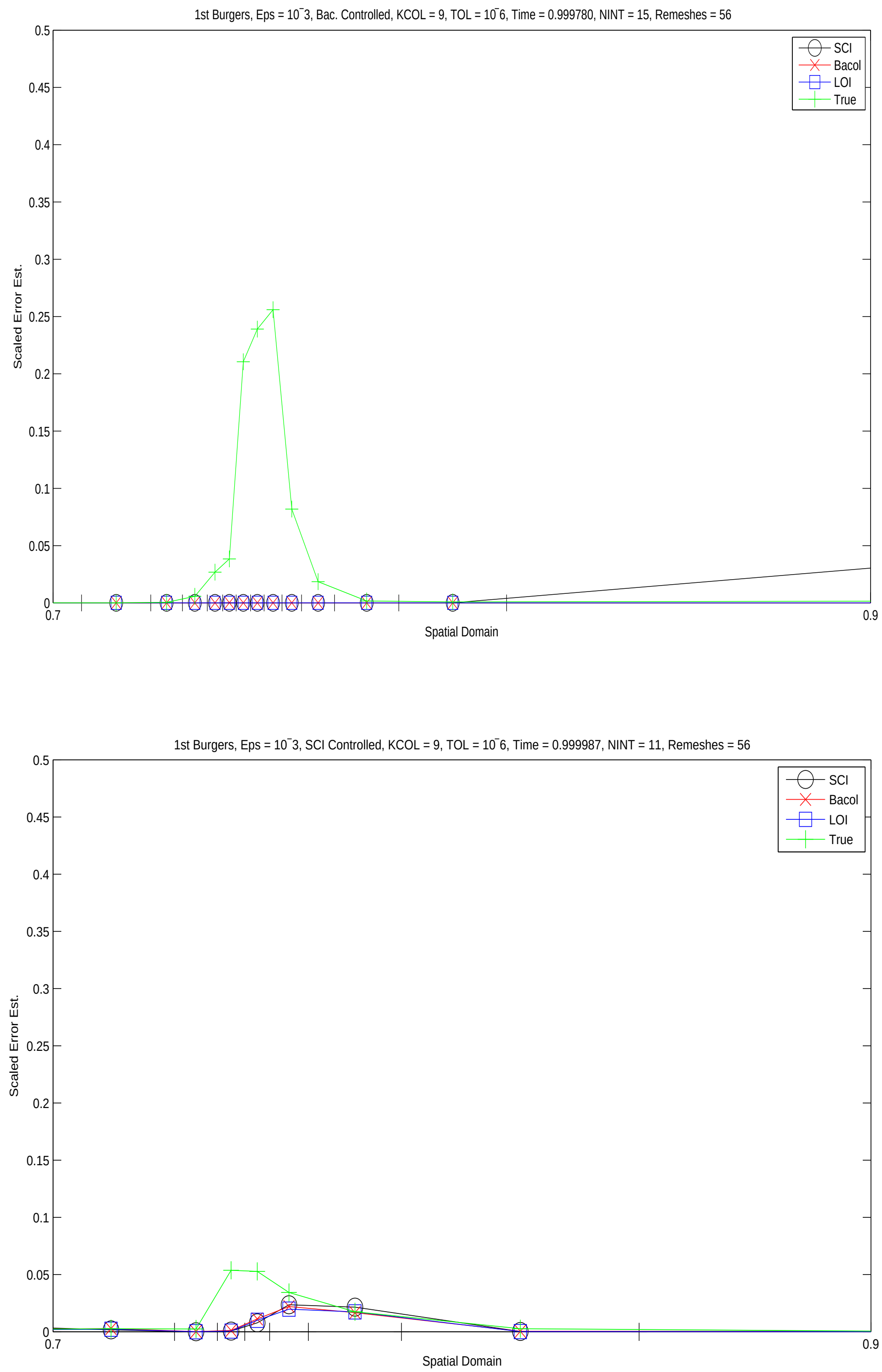
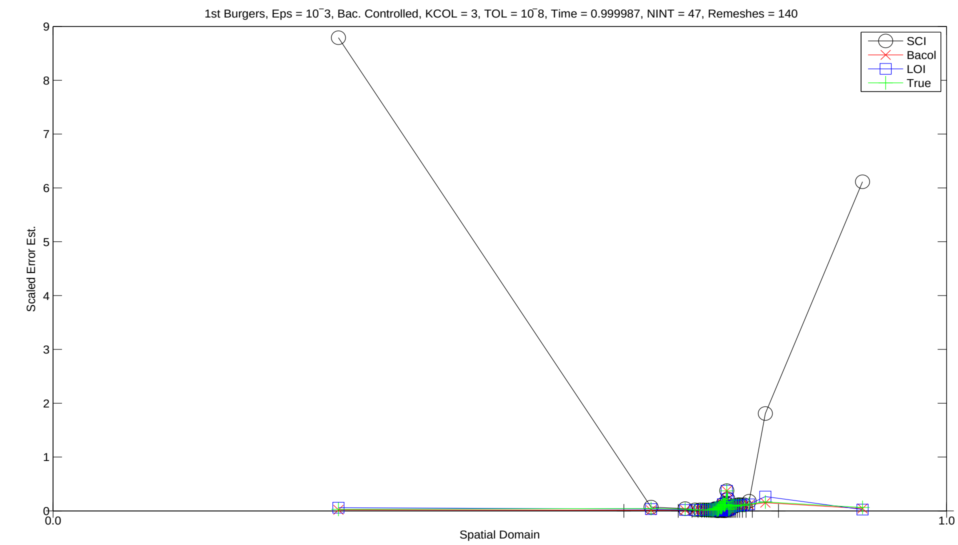


Figure 17: *Results only for the layer region. Error estimation in the layer region is poor for all schemes with the BACOL estimates in control of the mesh but the results are somewhat better with the SCI scheme in control.*

9.1.4 Results for $\epsilon = 10^{-3}$ and Tolerance $= 10^{-8}$



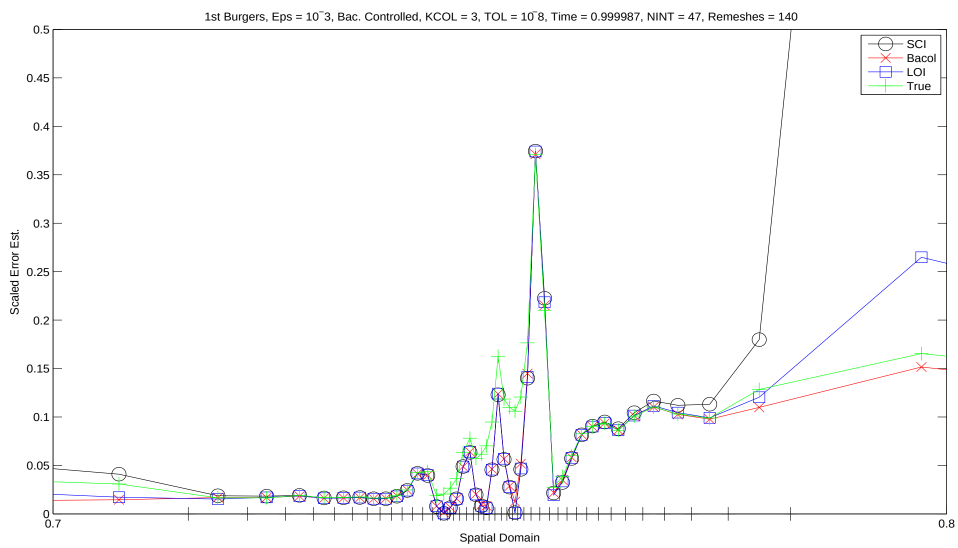
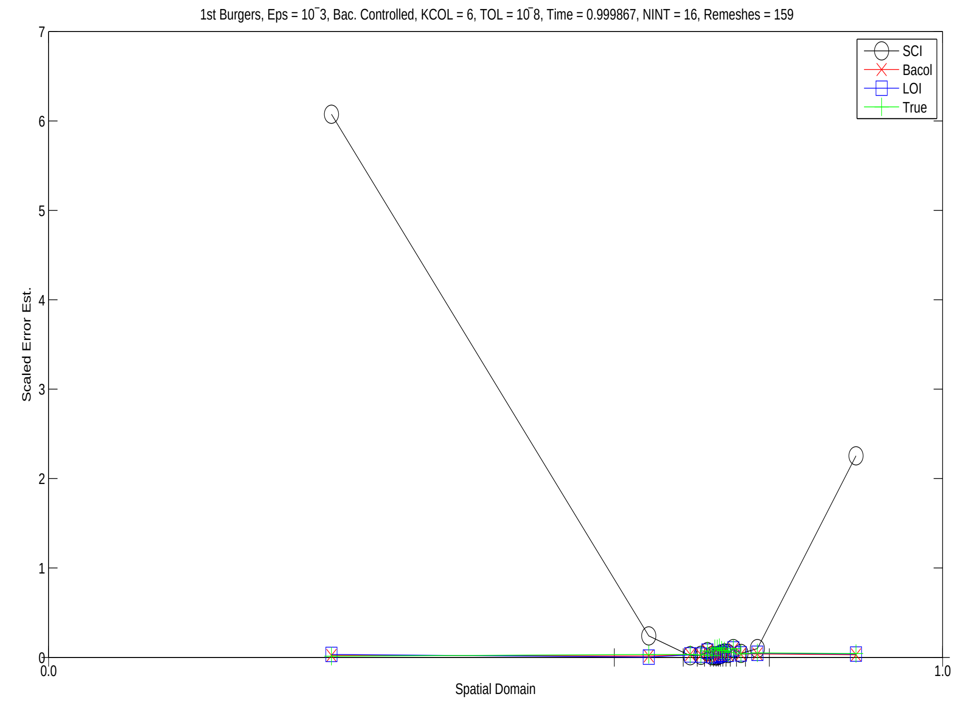


Figure 18: *In the top graph, we see that large boundary subinterval ratios cause the SCI scheme to overestimate the error with BACOL controlling the mesh. In the bottom graph, we see that all three schemes do a mediocre job of error estimation in the layer region, but they are in very close agreement. The time integration failed on the initial time step with the SCI scheme in control of mesh refinement.*



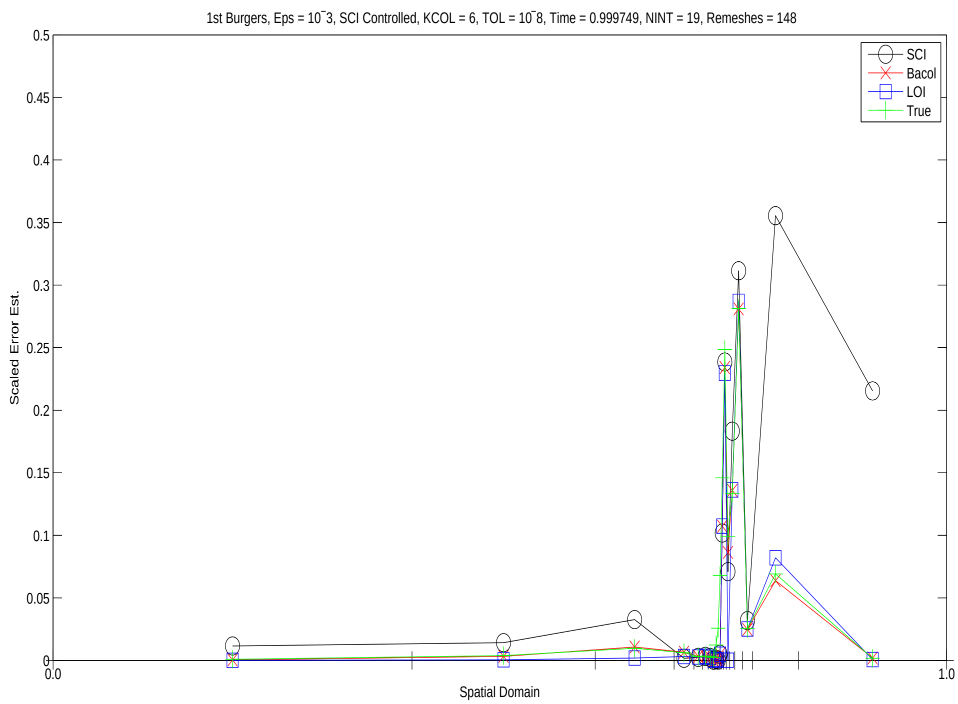
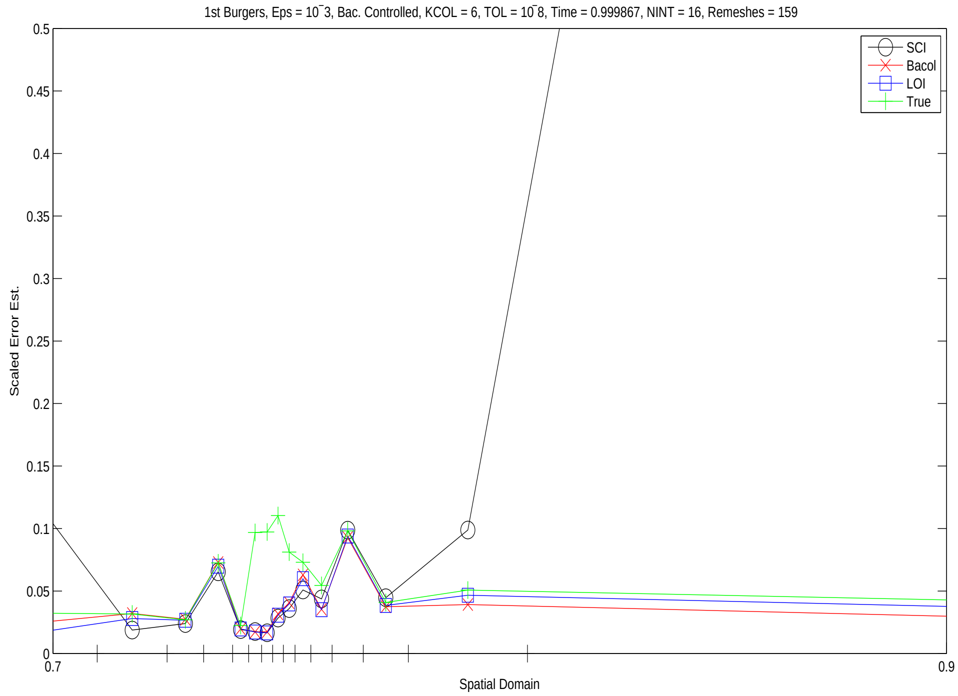


Figure 19: *With the BACOL estimate in control of the mesh, the SCI estimates at the boundaries are poor. With the SCI estimates in control of the mesh, the SCI estimates generally improve, except for the two subintervals closest to the right boundary.*



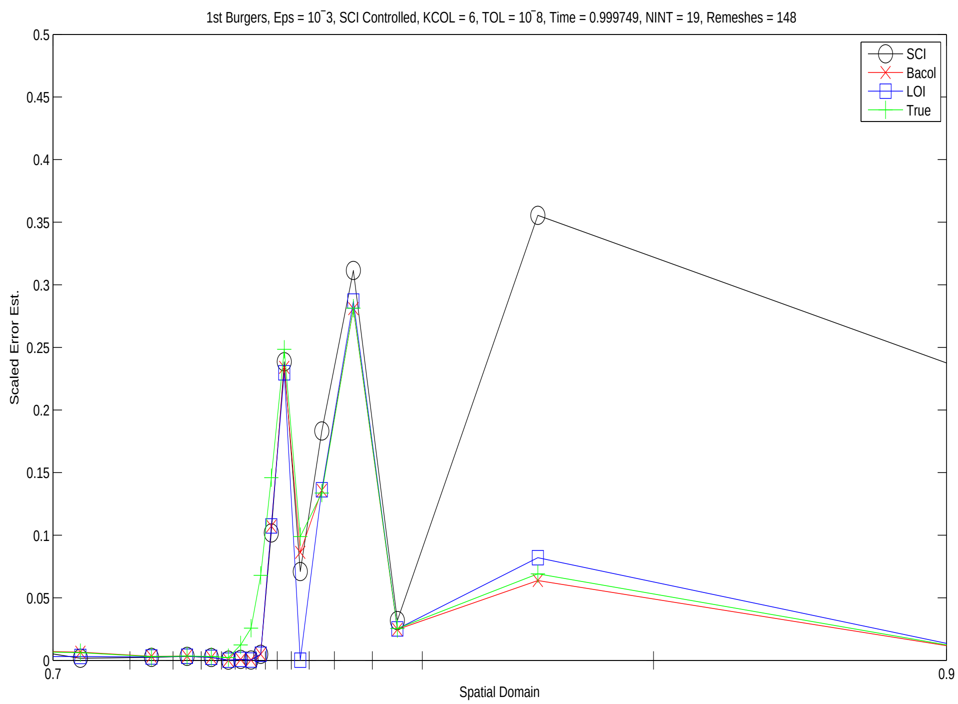


Figure 20: *Results only for the layer region. All three estimates are similar within the layer region when the BACOL scheme controls the mesh, and all three underestimate the error. Error estimates are more in line with the true error when the SCI scheme controls the mesh, except one subinterval where the LOI scheme estimates an error of zero. There is also an overestimate of the error by the SCI scheme in one of the transition subintervals to the right of the layer.*

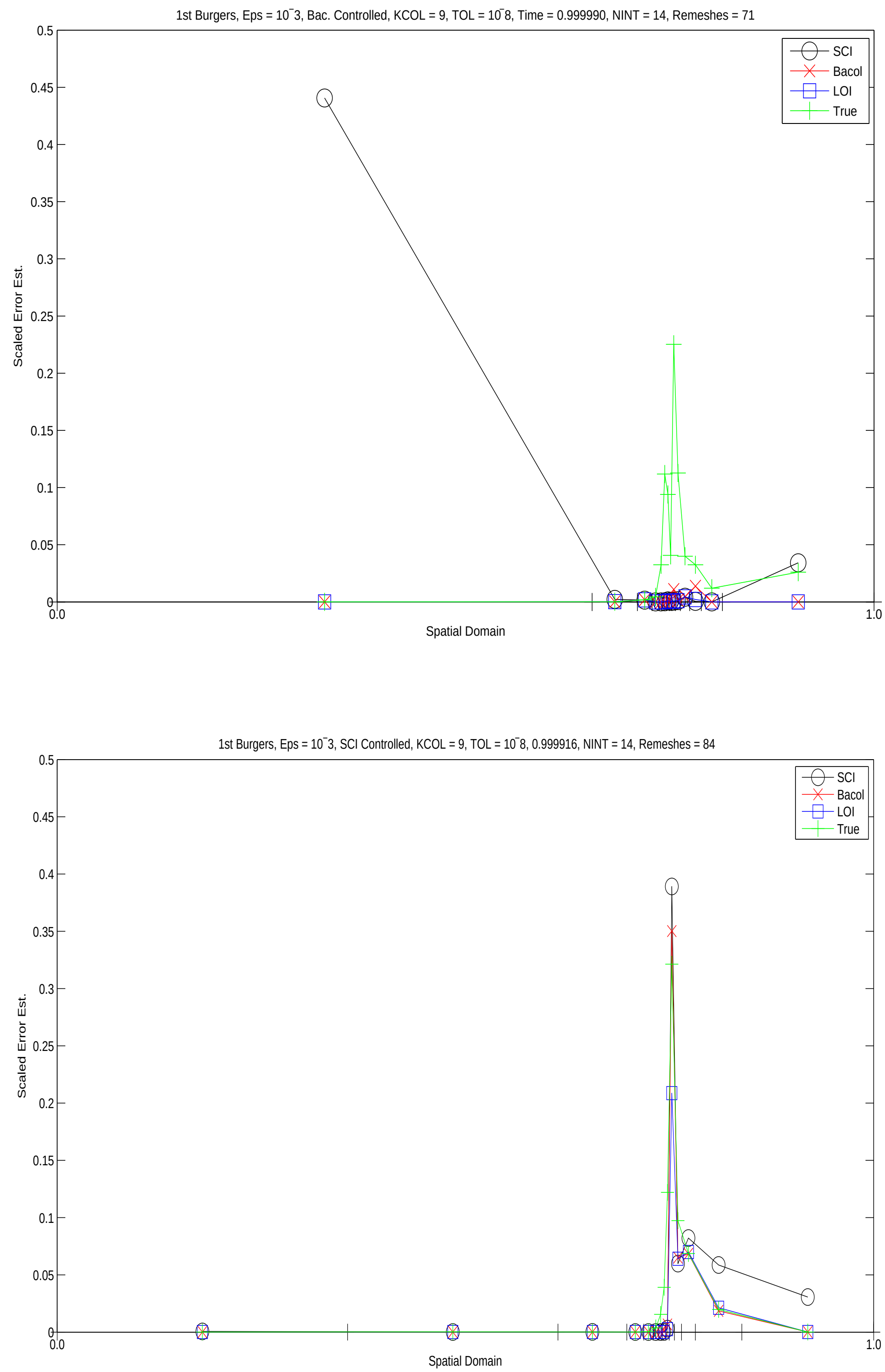
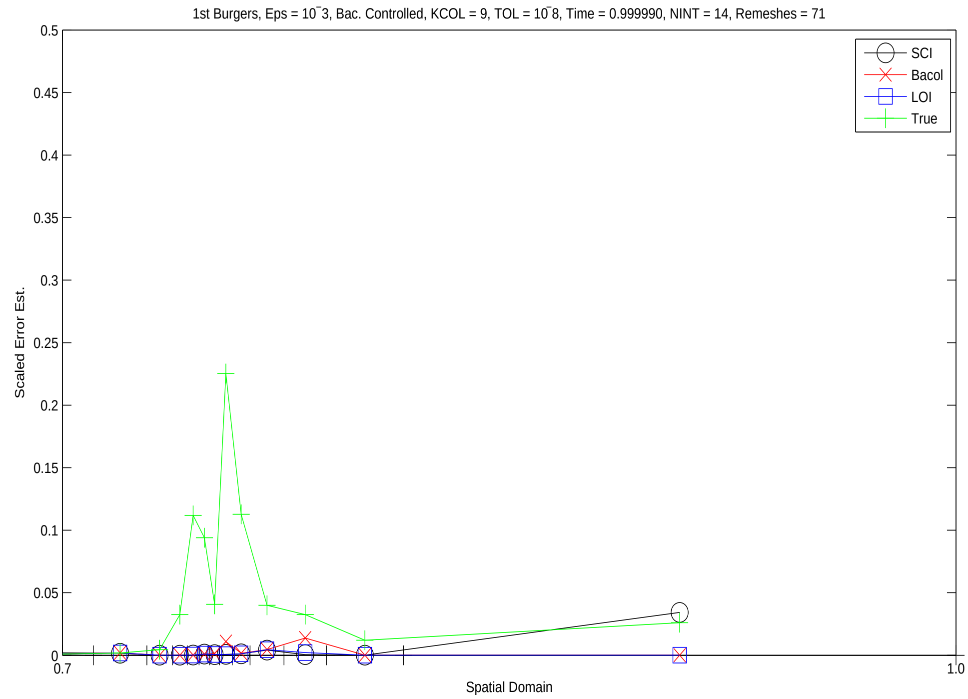


Figure 21: *In the top graph, the BACOL estimate controls the mesh and we see that the SCI scheme overestimates the errors on the boundary subintervals and that all the schemes underestimate the error in the layer region. In the bottom graph, the SCI scheme controls the mesh and we see generally good results for all schemes.*



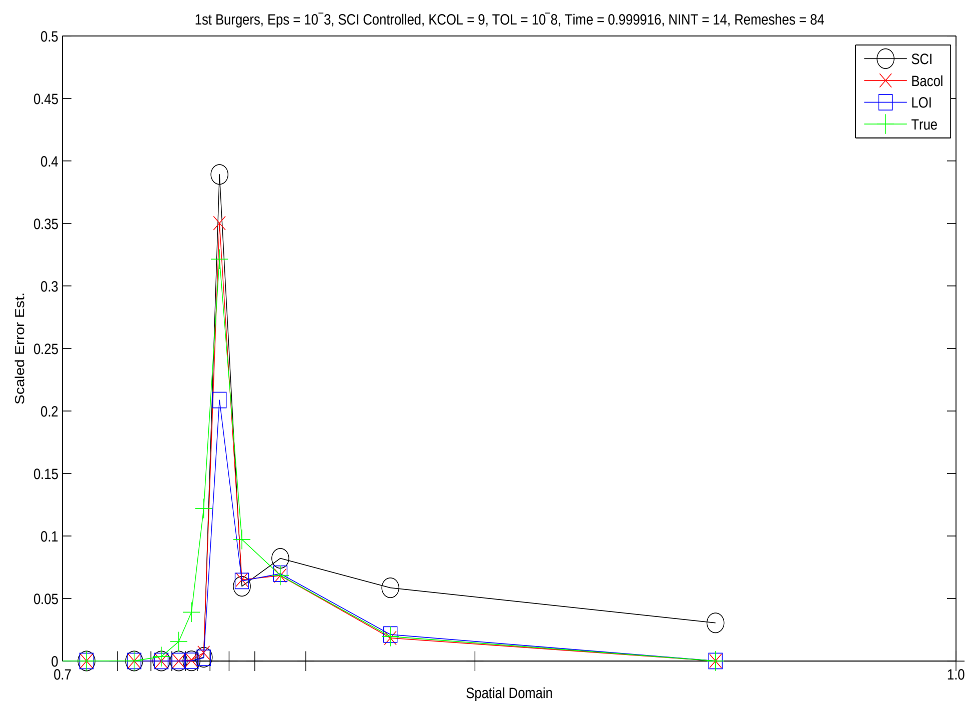


Figure 22: *Results only for the layer region. With the BACOL scheme in control, all three estimation schemes greatly underestimate errors in the layer region. The three perform better with the SCI scheme in control. Perhaps this is because the BACOL scheme overresolves the mesh in the layer, while the SCI scheme, due to subinterval ratios inflating error estimations, spreads the subintervals out toward the boundaries.*

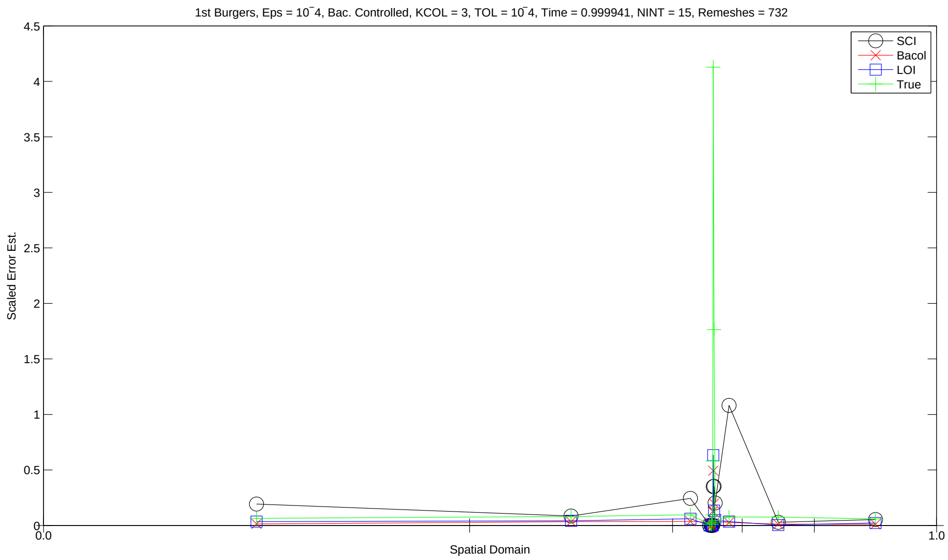
9.1.5 Summary of Results for the 1st Burgers Equation with $\epsilon = 10^{-3}$

We see that for this problem, the SCI scheme fails for $KCOL = 3$, which may be explained by the fact that the superconvergent values are not much more accurate than the other solution and derivative values. With the BACOL scheme driving mesh refinement, the SCI scheme greatly overestimates errors at the boundary subintervals, but with the SCI scheme in control it is able to self-correct for this. The BACOL scheme sometimes overresolves the mesh inside the layer region, while the SCI scheme's overestimation of error where subinterval ratios are large counteracts this by dispersing mesh points toward the boundary subintervals. When $KCOL = 9$, all schemes appear more likely to underestimate the error in the layer region with the BACOL scheme controlling the mesh than with the SCI scheme in control, and this may be related to the SCI scheme having a coarser mesh inside the layer region.

TOL \ KCOL		3	6	9
		NINT, Remeshes	NINT, Remeshes	NINT, Remeshes
10^{-4}	BAC	14, 88	10, 51	9, 41
	SCI	NA	11, 48	9, 34
10^{-6}	BAC	23, 117	13, 92	15, 56
	SCI	NA	13, 83	11, 56
10^{-8}	BAC	47, 140	16, 159	14, 71
	SCI	NA	19, 148	14, 84

Table 5: *Comparing results for $KCOL = 6$ and 9, we see that results for computations in which the BACOL estimates control the mesh are comparable to those for which the SCI scheme is in control.*

9.2 Results for $\epsilon = 10^{-4}$ and Tolerance $= 10^{-4}$



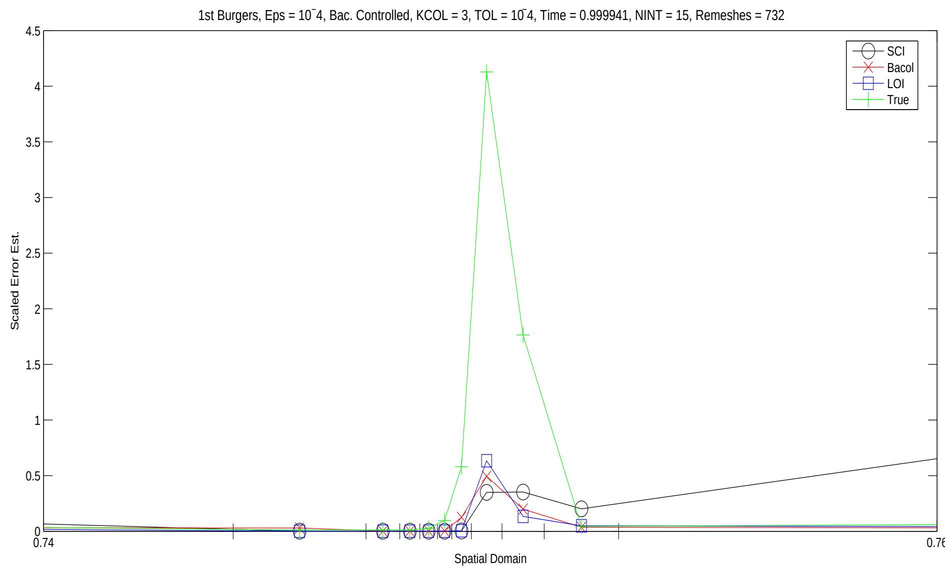
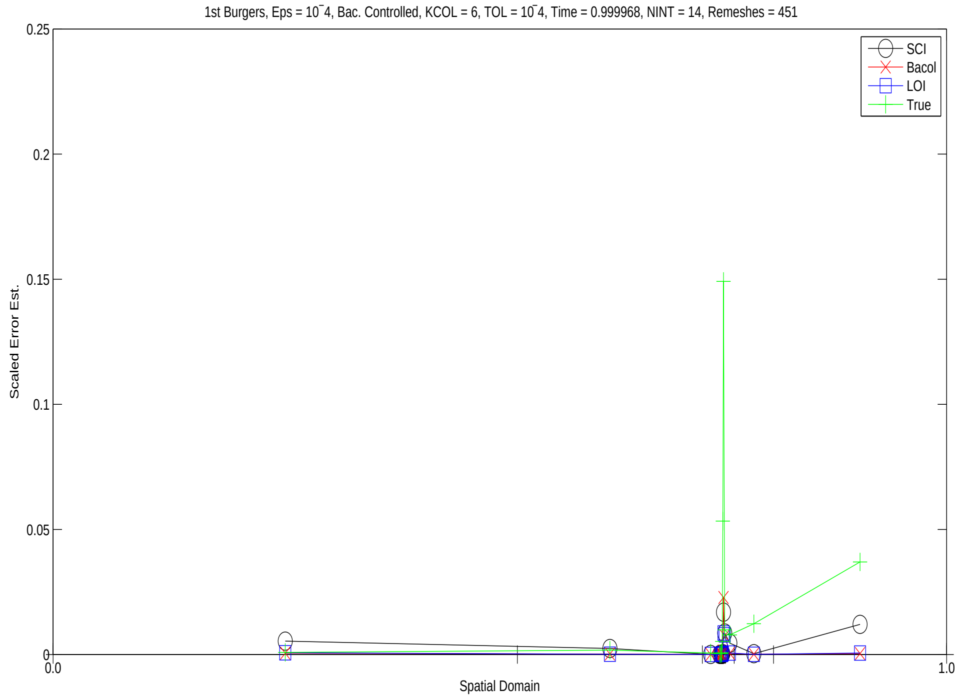


Figure 23: *With the BACOL estimation scheme controlling the mesh, all estimates are in good agreement except in the layer region where they all underestimate the error. With the SCI scheme controlling the mesh, the integration failed at the beginning.*



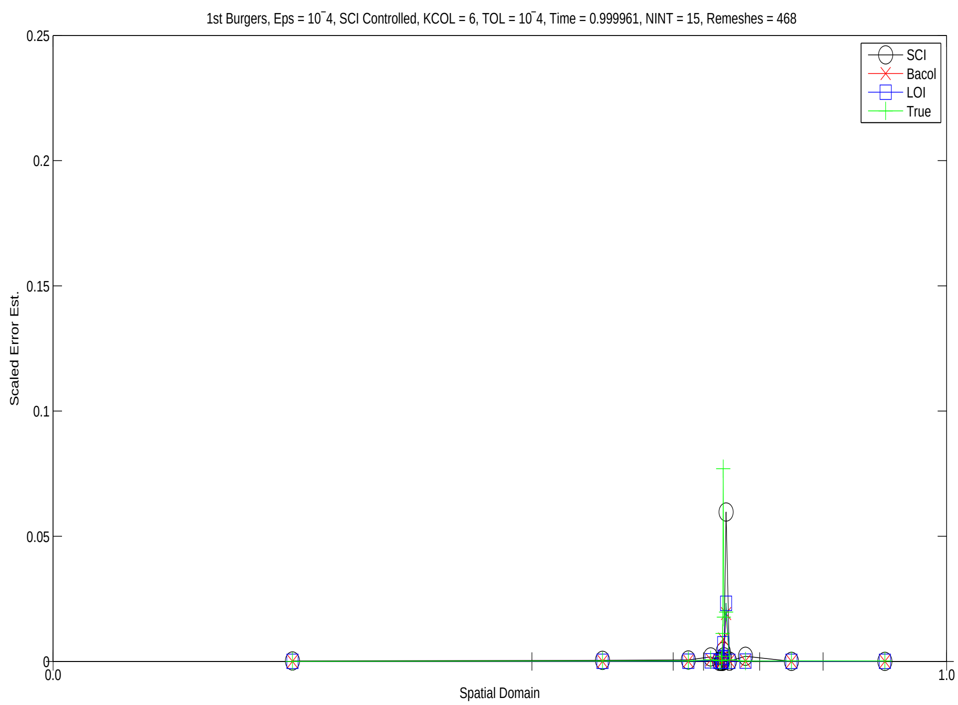
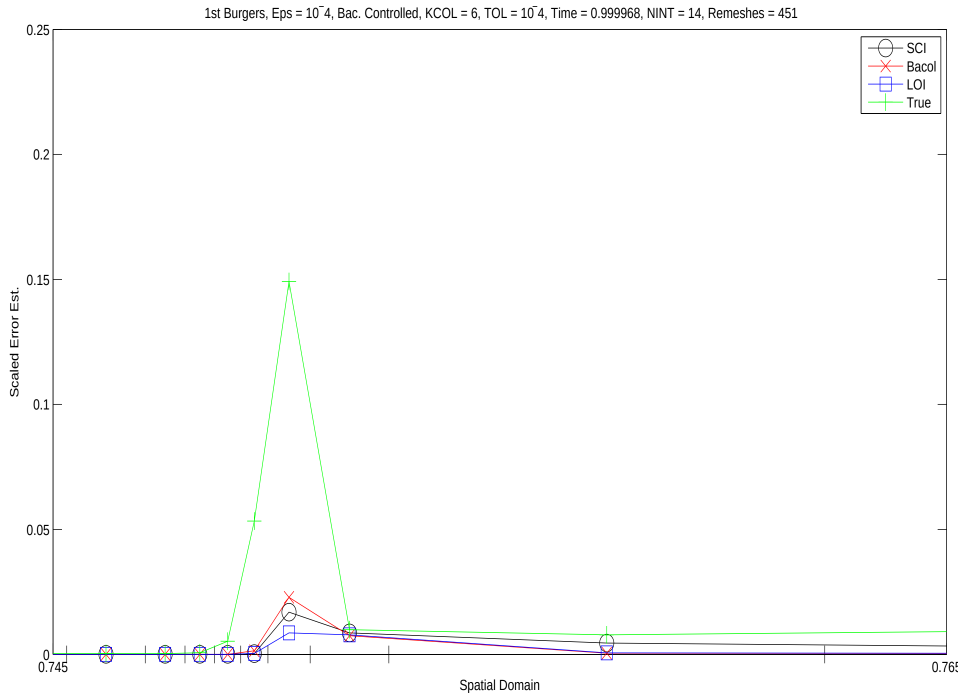


Figure 24: *With either estimation scheme controlling the mesh, all estimates are in good agreement except in the layer region where they all underestimate the error. The situation in the layer region is worse when the BACOL estimates control the mesh.*



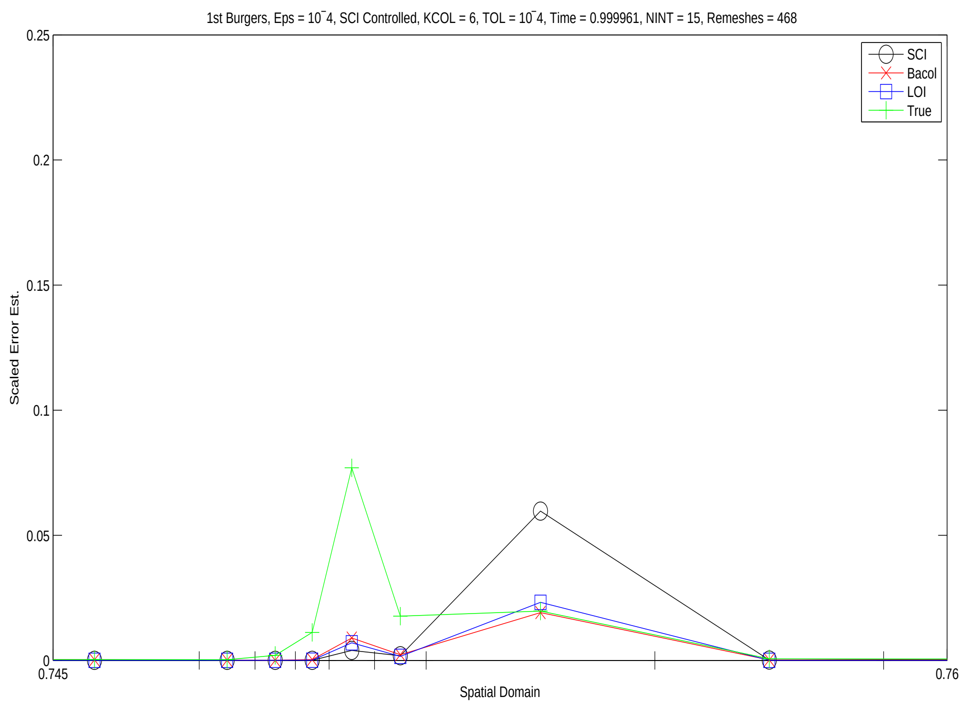
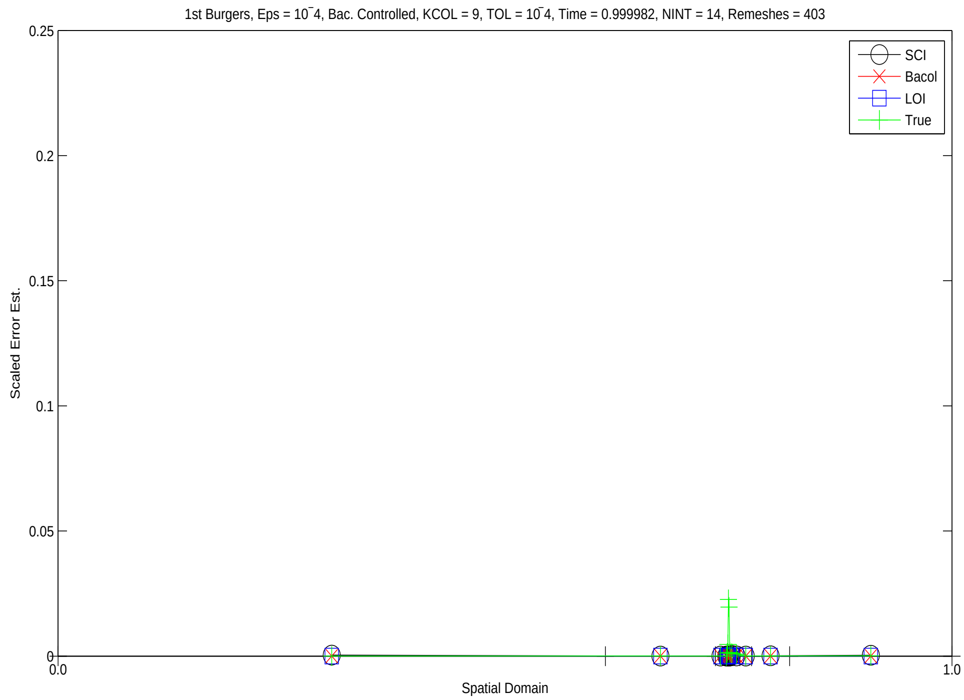


Figure 25: *Results only for the layer region. In both cases, error estimates in the layer region are too small. The situation is worse when the BACOL estimates control the mesh. When the SCI estimates control the mesh, the SCI estimate is too large on the transition subinterval to the right of the layer region.*



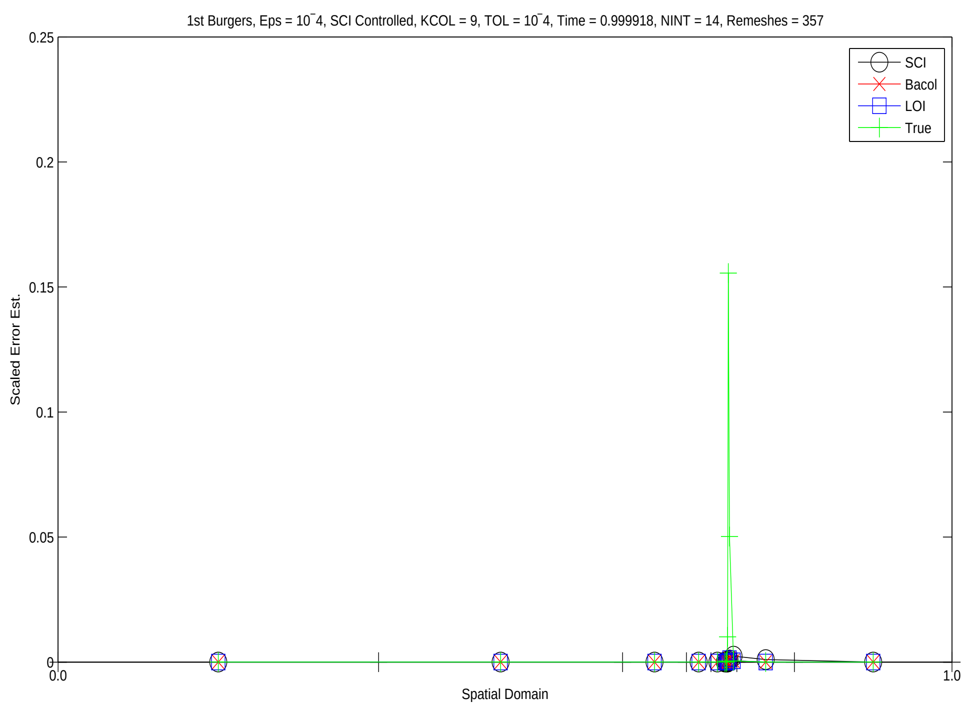
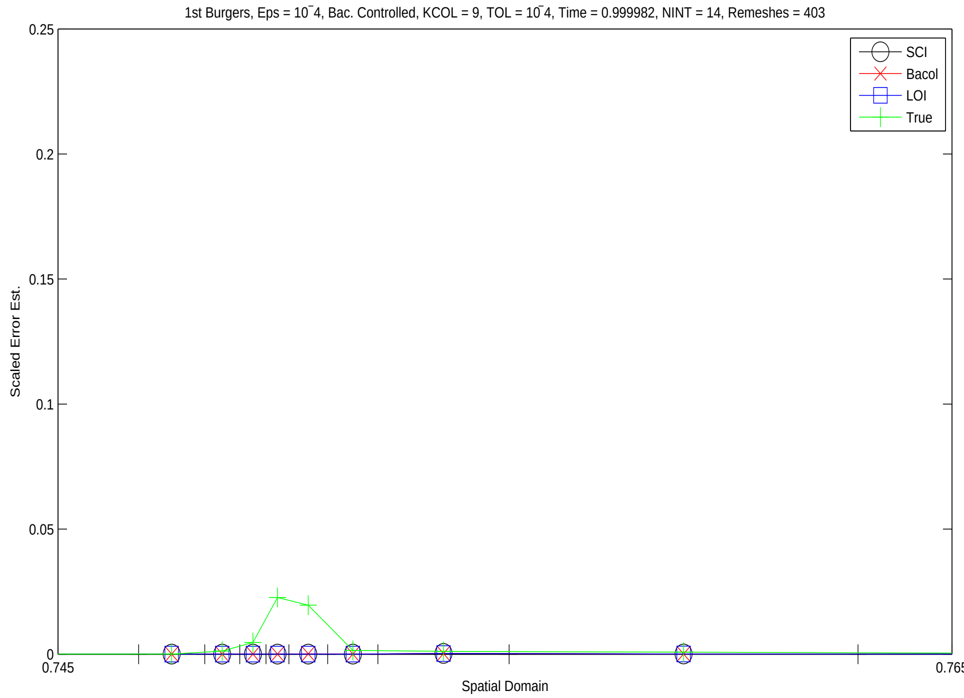


Figure 26: *In both cases, the error estimates are in good agreement except in the layer region where all estimates are too small. The situation is worse when the SCI estimates control the mesh.*



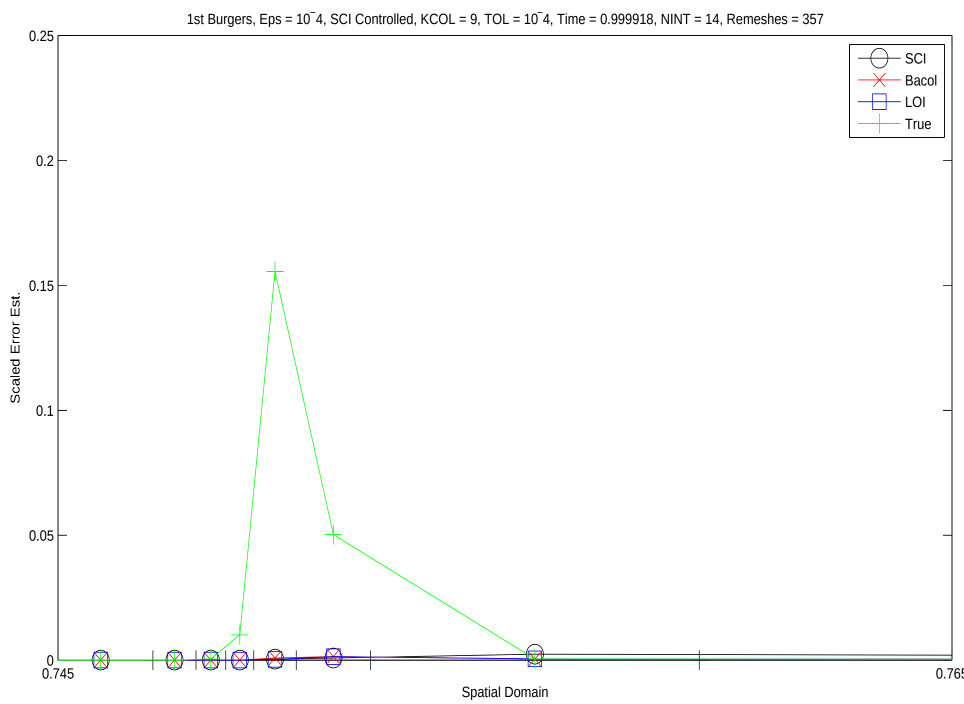
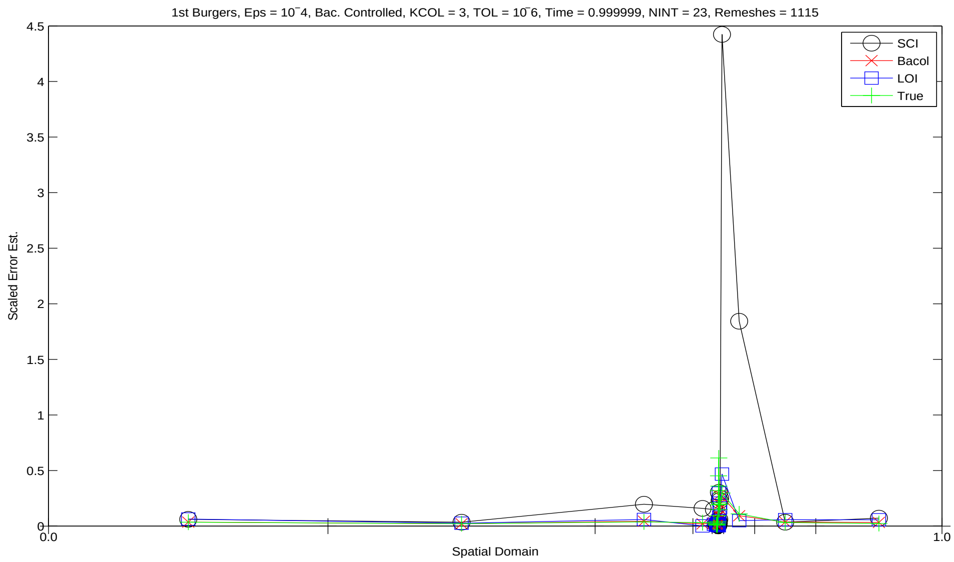


Figure 27: *Results only for the layer region. With either the BACOL or SCI schemes controlling the mesh, the error estimates in the layer region are too small. The situation is worse when the SCI estimates control the mesh.*

9.2.1 Results for $\epsilon = 10^{-4}$ and Tolerance $= 10^{-6}$



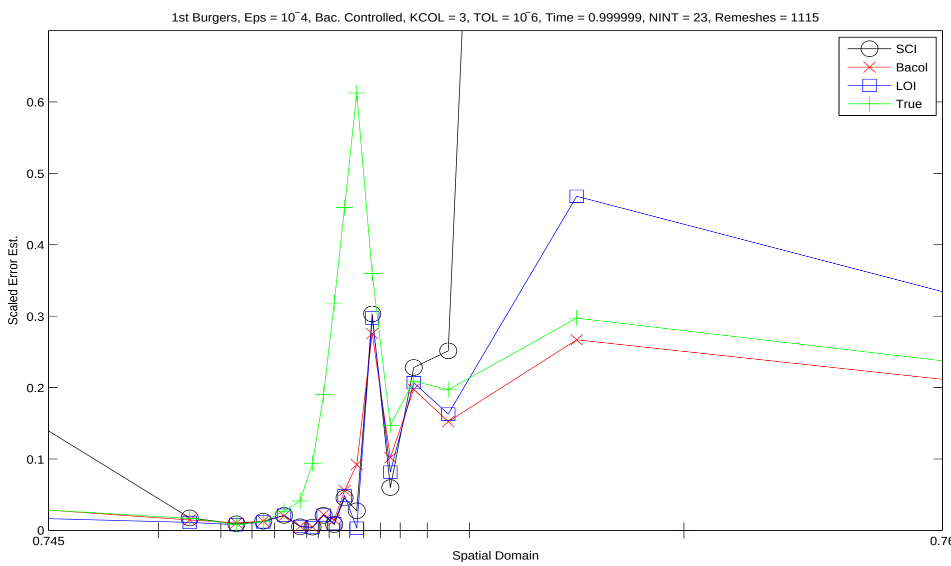
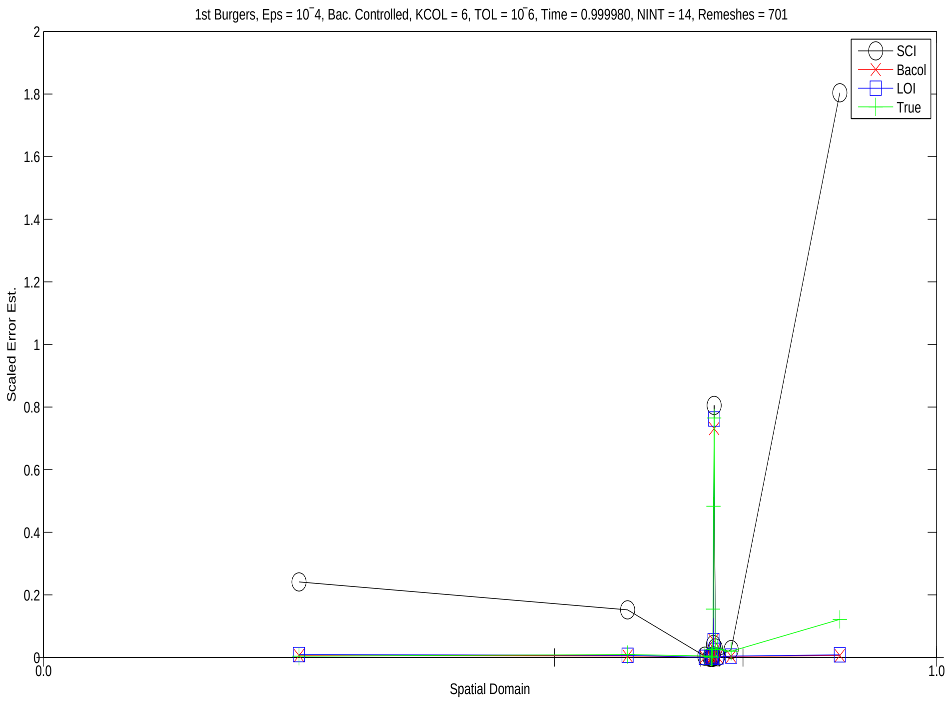


Figure 28: *All estimates are in good agreement except in the layer region. The SCI estimate is a substantial overestimate for one subinterval immediately to the right of the layer region. When the SCI estimates control the mesh, the integration failed at the starting time.*



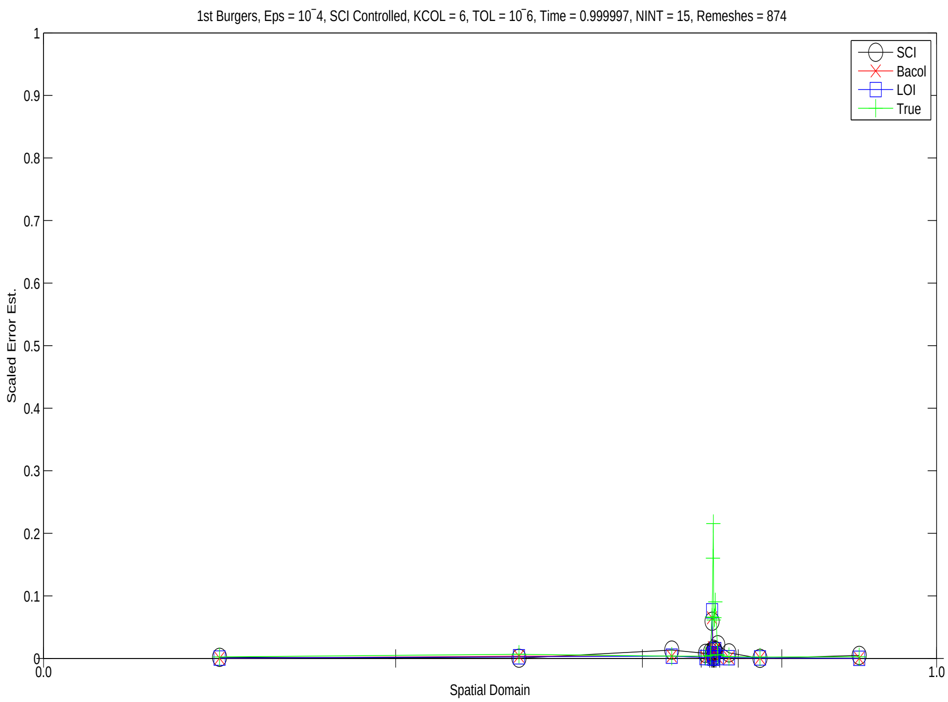
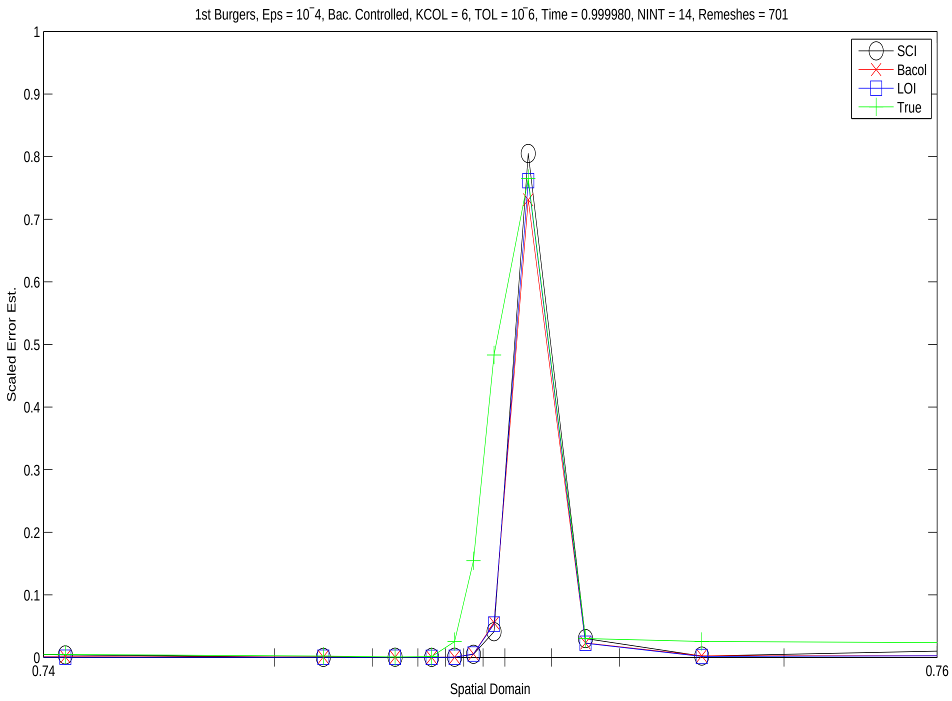


Figure 29: *With the BACOL scheme controlling the mesh, the estimates are in good agreement, except for the rightmost boundary subinterval where the SCI estimate is too big. When the SCI estimates control the mesh, all estimates are in good agreement outside the layer region.*



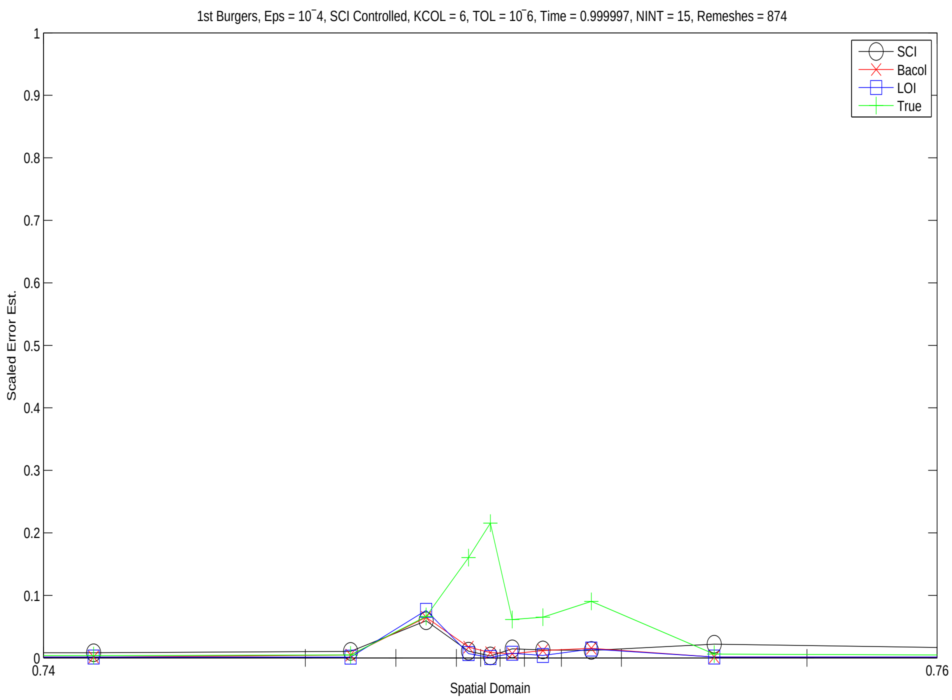
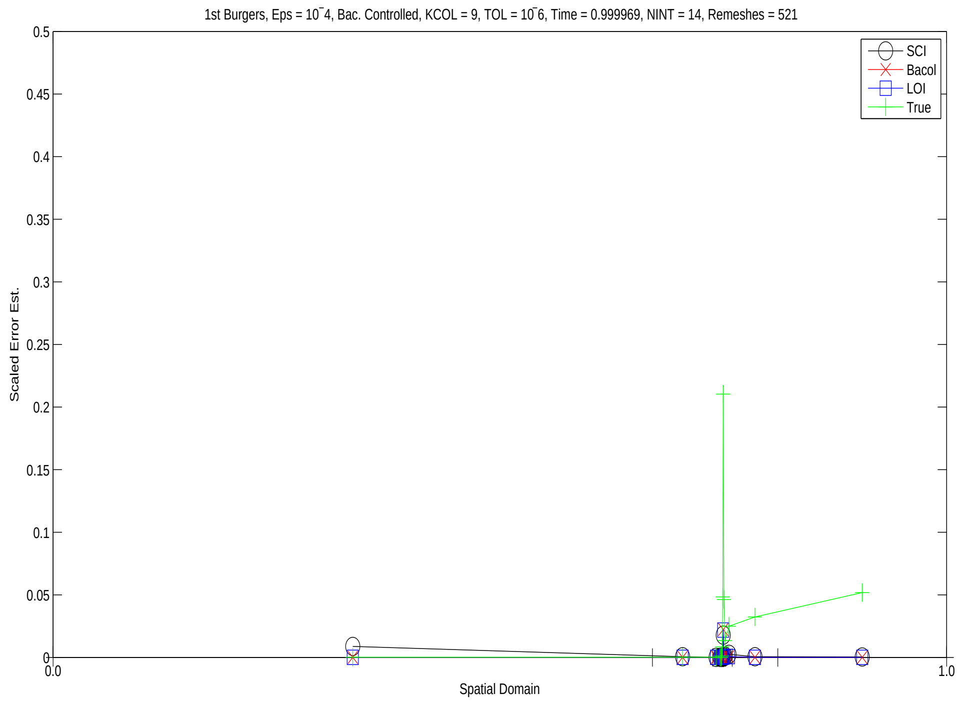


Figure 30: *Results only for the layer region. With the BACOL scheme controlling the mesh, the estimates are in good agreement inside the layer region. When the SCI estimates control the mesh, all estimates underestimate the error inside the layer region.*



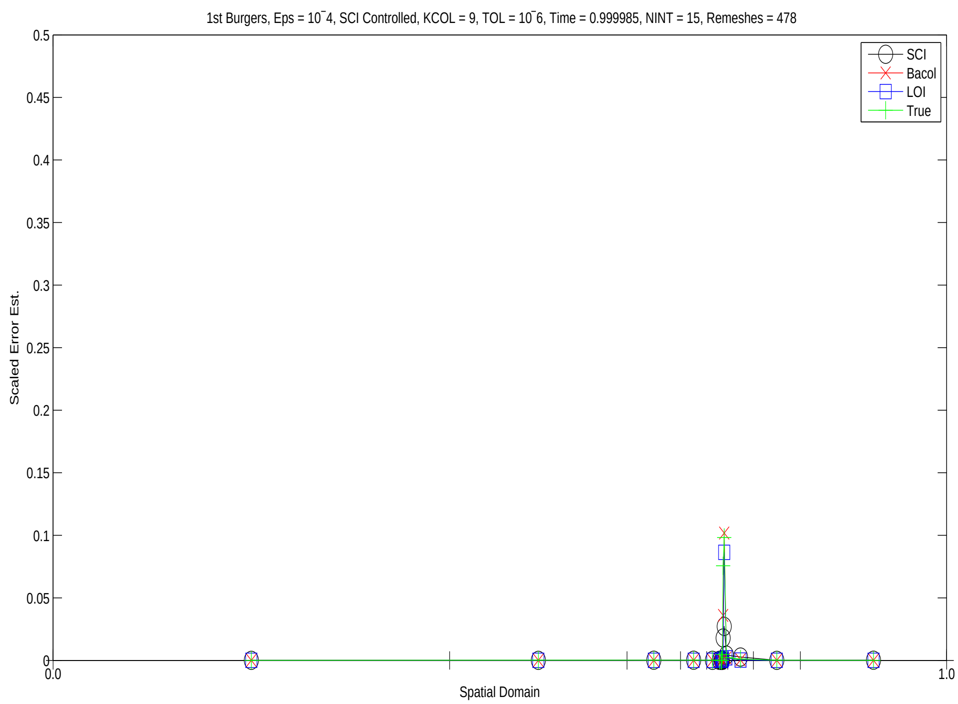
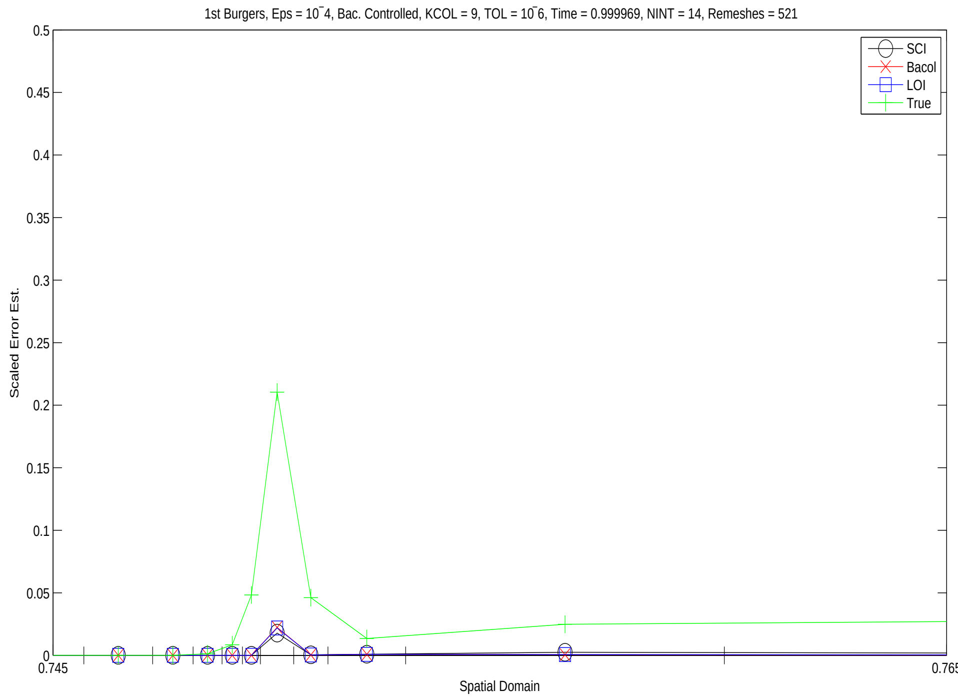


Figure 31: *With the BACOL scheme controlling the mesh, the estimates are in good agreement to the left of the layer region but within the layer region and to the right of the layer region the estimates underestimate the error. When the SCI estimates control the mesh, all estimates are in good agreement except the SCI estimate.*



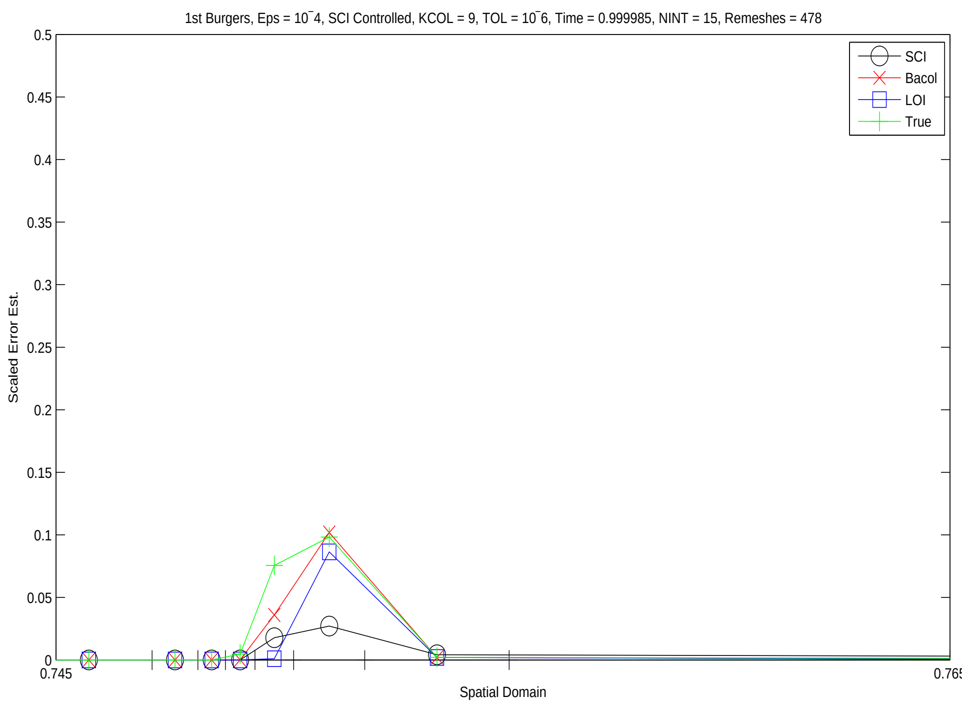
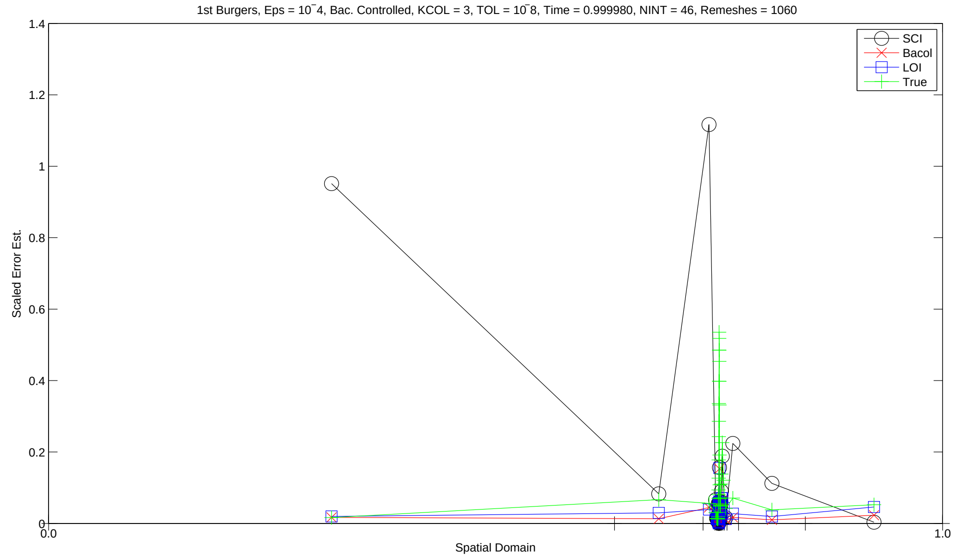


Figure 32: *Results only for the layer region. With the BACOL scheme controlling the mesh, all estimates underestimate the error. When the SCI estimates control the mesh, all estimates are in good agreement.*

9.2.2 Results for $\epsilon = 10^{-4}$ and Tolerance $= 10^{-8}$



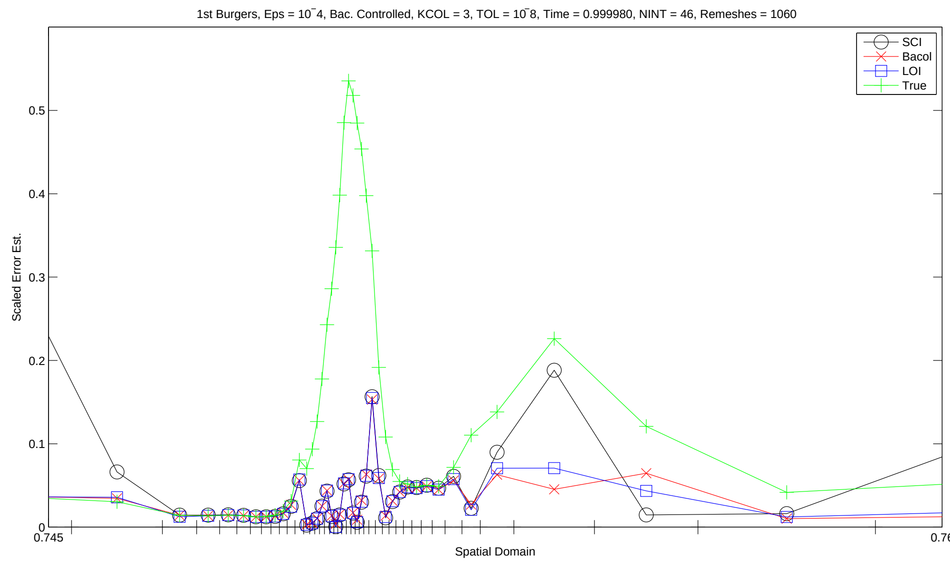
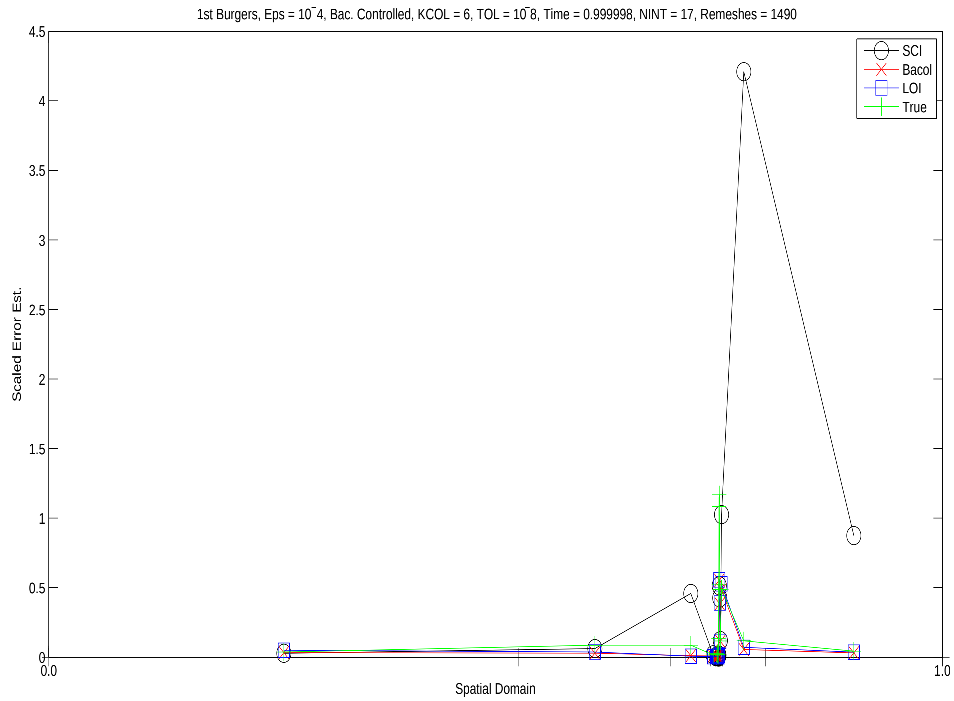


Figure 33: *The estimates are in generally good agreement except for the SCI estimates which substantially overestimate the error on some of the subintervals. Within the layer region, all estimates substantially underestimate the error. When the SCI estimates control the mesh, the integration failed at the starting time.*



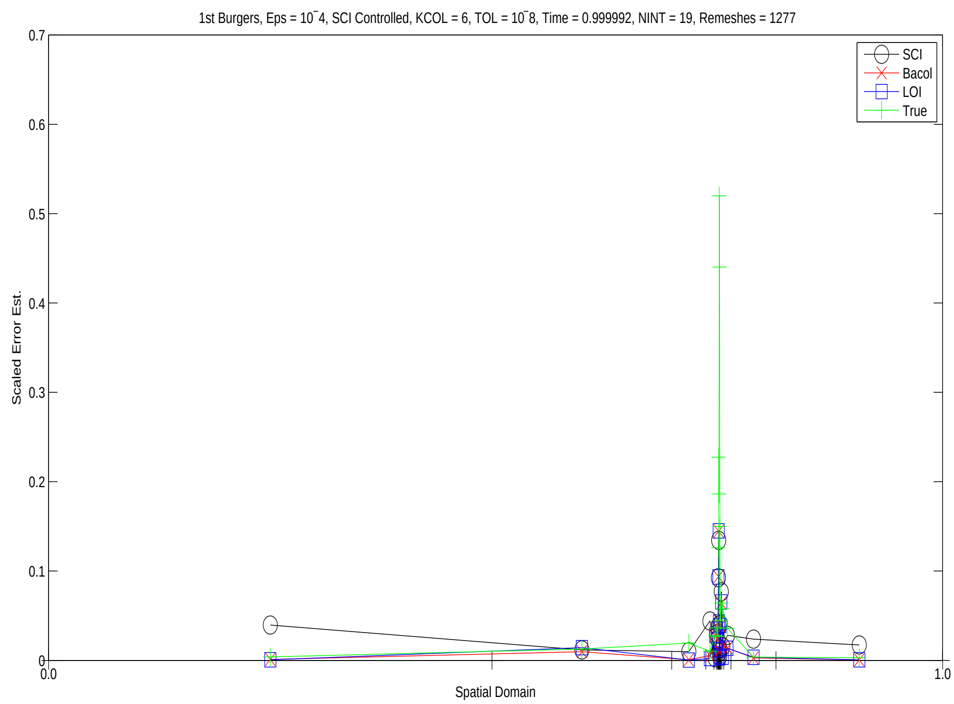


Figure 34: *With the BACOL scheme controlling the mesh, the estimates are generally in good agreement except for the SCI estimates which overestimate the error near the layer region. When the SCI estimates control the mesh, all estimates are in good agreement except in the layer region where all estimates underestimate the error.*

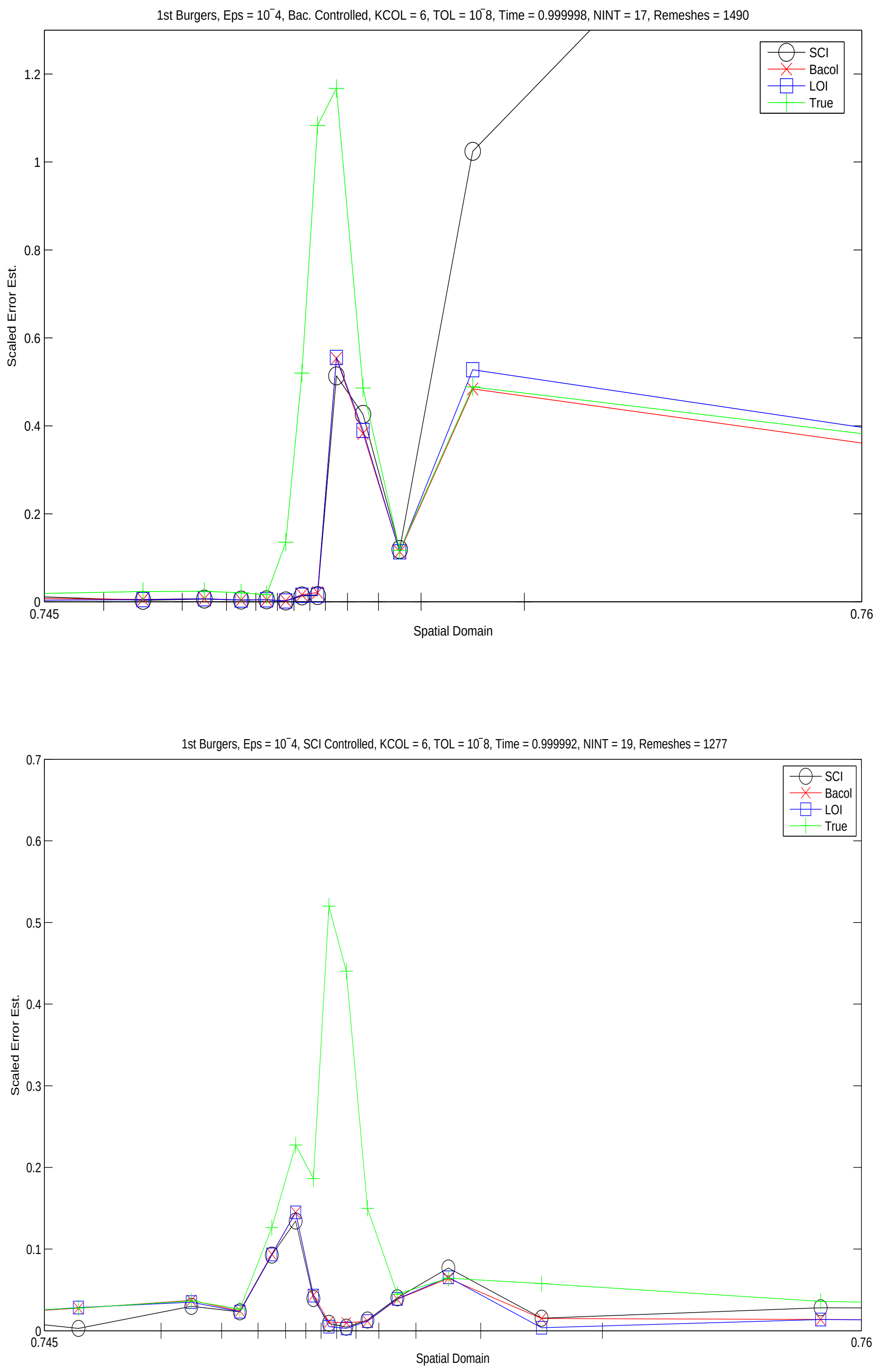
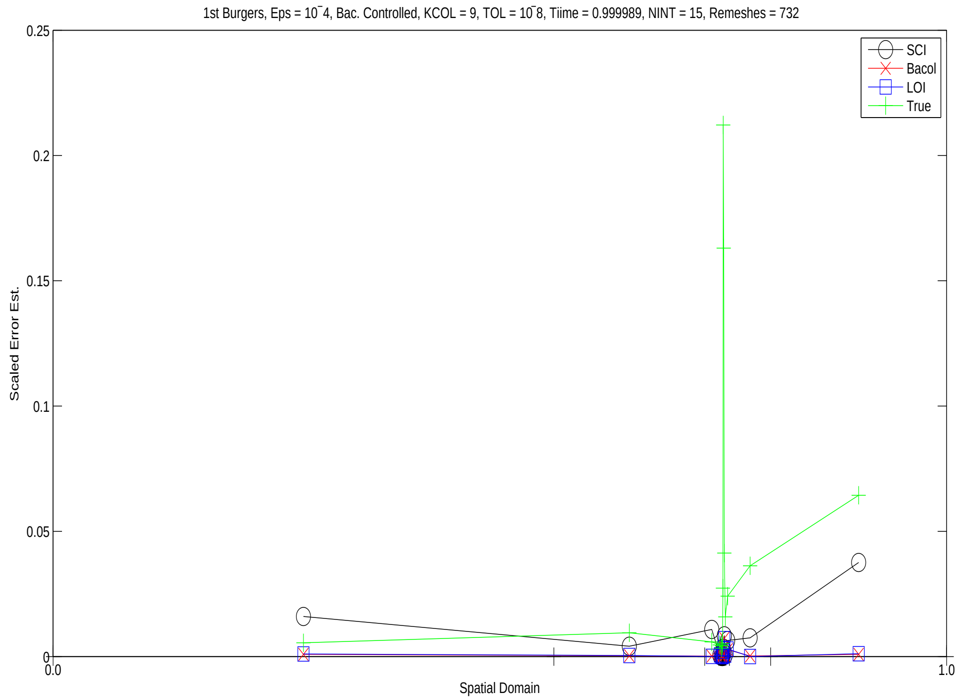


Figure 35: *Results only for the layer region. With either scheme controlling the mesh, all estimates underestimate the error in the layer region.*



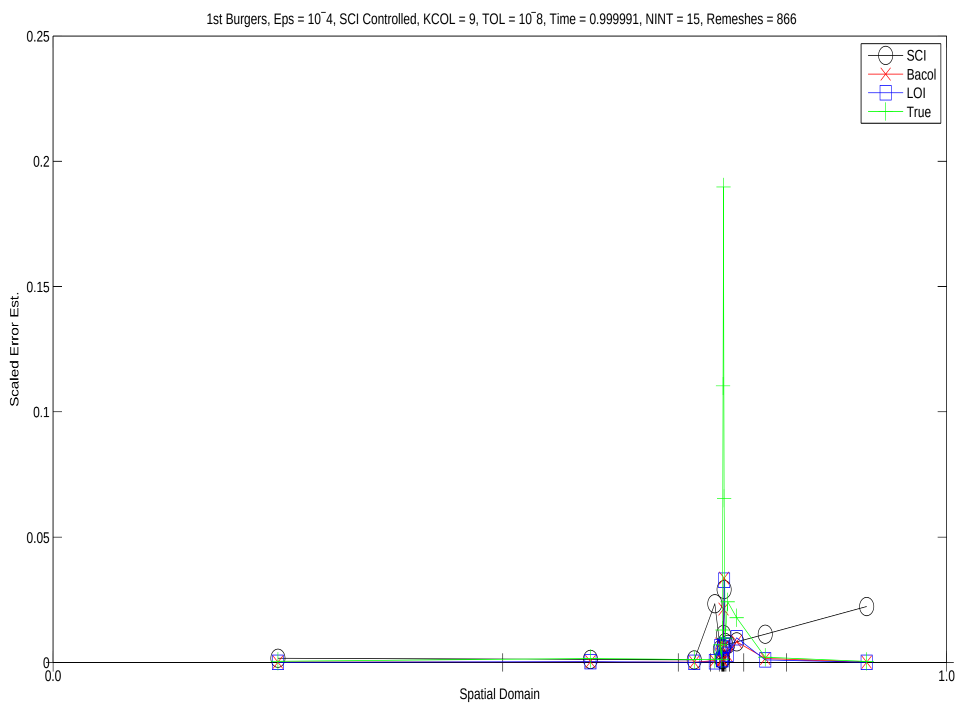
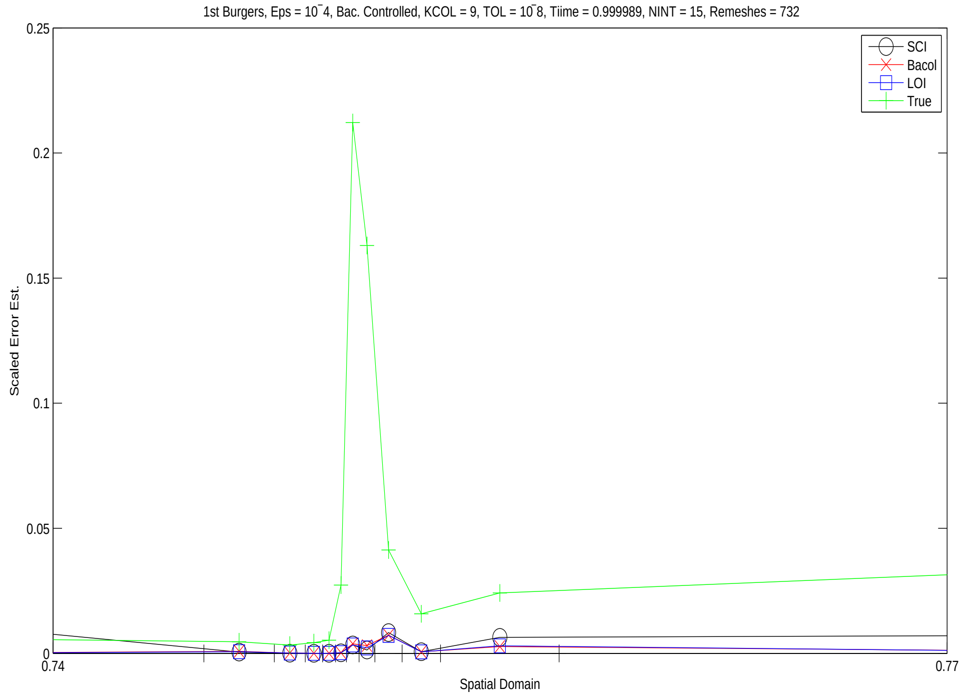


Figure 36: *With the BACOL scheme controlling the mesh, the estimates are generally in good agreement except in the layer region and to the right of the layer region where the estimates are too small. When the SCI estimates control the mesh, all estimates are in good agreement except in the layer region where all estimates underestimate the error.*



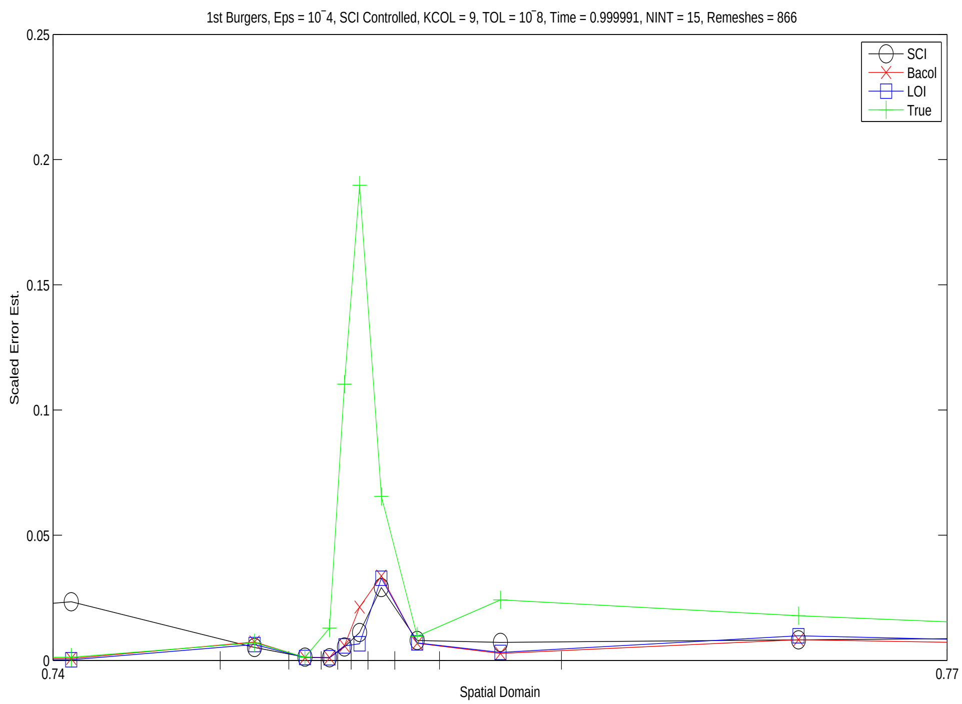


Figure 37: *Results only for the layer region. With either scheme controlling the mesh, all estimates underestimate the error in the layer region.*

9.2.3 Summary of Results for the 1st Burgers Equation with $\epsilon = 10^{-4}$

TOL \ KCOL		3	6	9
		NINT, Remeshes	NINT, Remeshes	NINT, Remeshes
10^{-4}	BAC	15, 732	14, 451	14, 403
	SCI	NA	15, 468	14, 357
10^{-6}	BAC	13, 1115	14, 701	14, 521
	SCI	NA	15, 874	15, 478
10^{-8}	BAC	46, 1060	17, 1490	15, 732
	SCI	NA	19, 1277	15, 866

Table 6: *The columns for $KCOL = 6$ and 9 show that the number of subintervals used in the meshes and the number of remeshings are comparable, when mesh refinement is based on either the BACOL estimates or the SCI estimates.*

9.3 2nd Burgers' Equation

9.3.1 Problem

$$u_t = -uu_x + \epsilon u_{xx}, \qquad 0 < x < 1, \quad t > 0$$

Initial Condition:

$$u(x,0) = \frac{0.1e^{-A_0} + 0.5e^{-B_0} + e^{-C_0}}{e^{-A_0} + e^{-B_0} + e^{-C_0}}, \qquad 0 \le x \le 1$$

Boundary Conditions:

$$u(0,t) = \frac{0.1e^{-A_L} + 0.5e^{-B_L} + e^{-C_L}}{e^{-A_L} + e^{-B_L} + e^{-C_L}}, \quad t \ge 0$$

$$u(0,t) = \frac{0.1e^{-A_R} + 0.5e^{-B_R} + e^{-C_R}}{e^{-A_R} + e^{-B_R} + e^{-C_R}}, \quad t \ge 0$$

where,

$$A_0 = \frac{0.05}{\epsilon}(x - 0.5), \quad B_0 = \frac{0.25}{\epsilon}(x - 0.5), \quad C_0 = \frac{0.5}{\epsilon}(x - 0.375)$$

$$A_L = \frac{0.05}{\epsilon}(-0.5 + 4.95t), \quad B_L = \frac{0.25}{\epsilon}(-0.5 + 0.75t), \quad C_L = \frac{0.5}{\epsilon}(-0.375)$$

$$A_R = \frac{0.05}{\epsilon}(0.5 + 4.95t), \quad B_R = \frac{0.25}{\epsilon}(0.5 + 0.75t), \quad C_R = \frac{0.5}{\epsilon}(0.625)$$

Exact Solution:

$$u(x,t) = \frac{0.1e^{-A} + 0.5e^{-B} + e^{-C}}{e^{-A} + e^{-B} + e^{-C}},$$

where,

$$A = \frac{0.05}{\epsilon}(x - 0.5 + 4.95t), \quad B = \frac{0.25}{\epsilon}(x - 0.5 + 0.75t), \quad C = \frac{0.5}{\epsilon}(x - 0.375)$$

See [38] for a figure showing the exact solution.

The following six subsubsections provide results for $\epsilon = 10^{-3}$ and tolerances of 10^{-4}, 10^{-6}, and 10^{-8}, and $\epsilon = 10^{-4}$ and tolerances of 10^{-4}, 10^{-6}, and 10^{-8}. Within each subsection, we provide results for $k = KCOL = 3, 6$, and 9. The title of each graph indicates: the problem (2nd Burgers), the value of ϵ (Eps $= 10^{-3}$ or Eps $= 10^{-4}$), which of the error estimation schemes controls the mesh refinement (Bac. Controlled or SCI Controlled), the value of KCOL, the tolerance TOL, the final Time (close to 1.0), the number of subintervals in the final mesh, NINT, and the total number of spatial remeshings that took place over the entire computation, Remeshes.

9.3.2 Results for $\epsilon = 10^{-3}$ and Tolerance $= 10^{-4}$

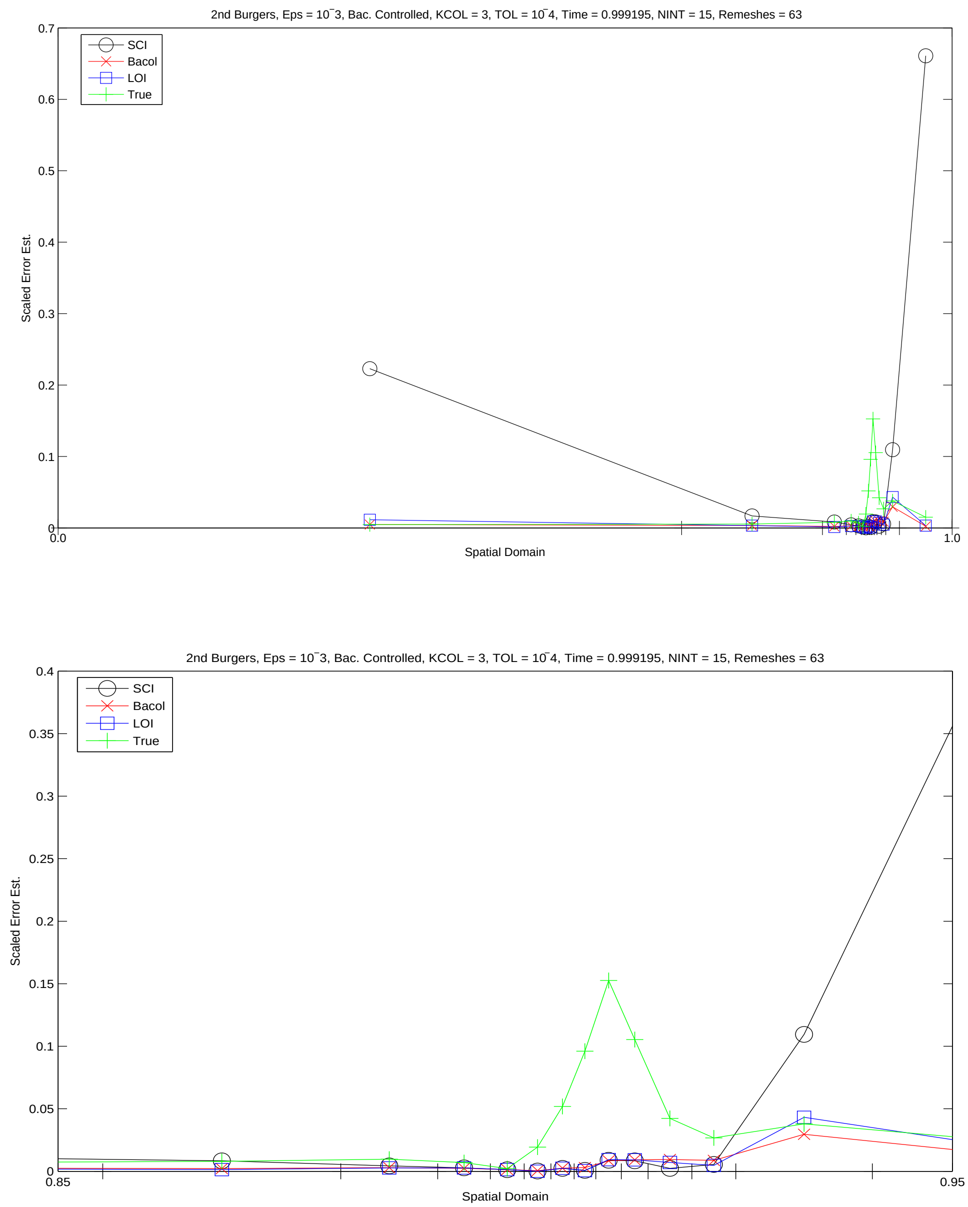
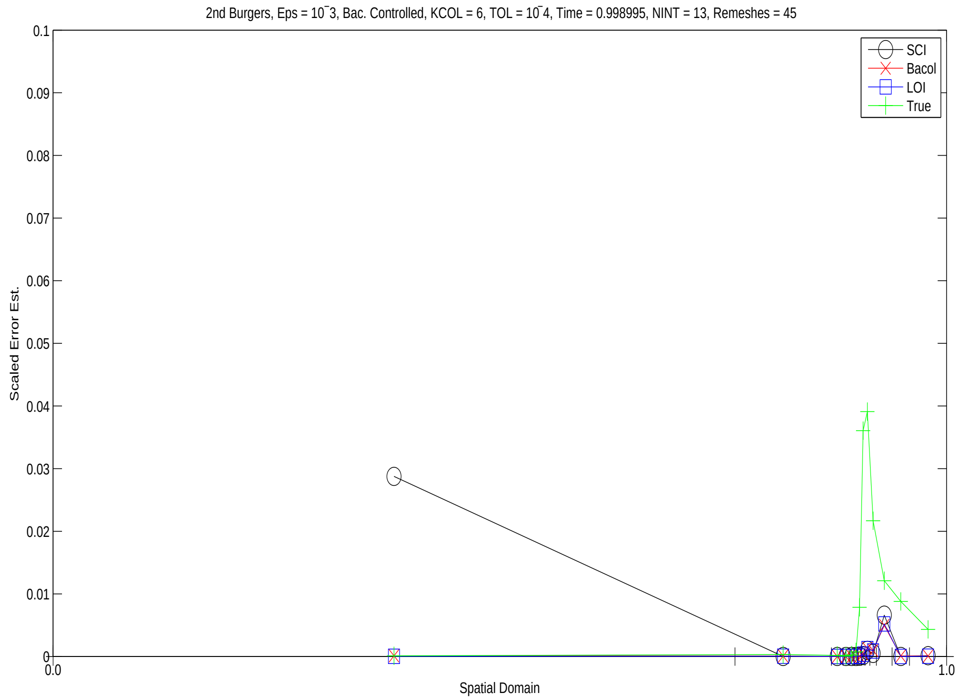


Figure 38: *These graphs are based on BACOL estimate control of the mesh. The SCI estimates on the boundary subintervals are too large. All estimates underestimate the error in the layer region. When the SCI estimates control the mesh, the time integration fails at the starting point.*



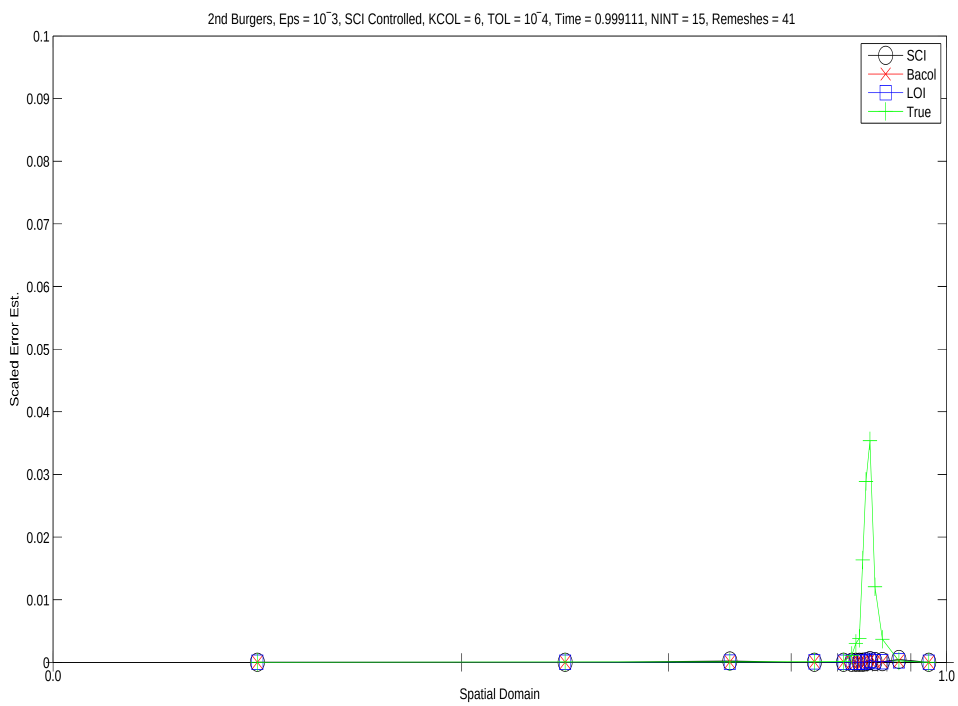
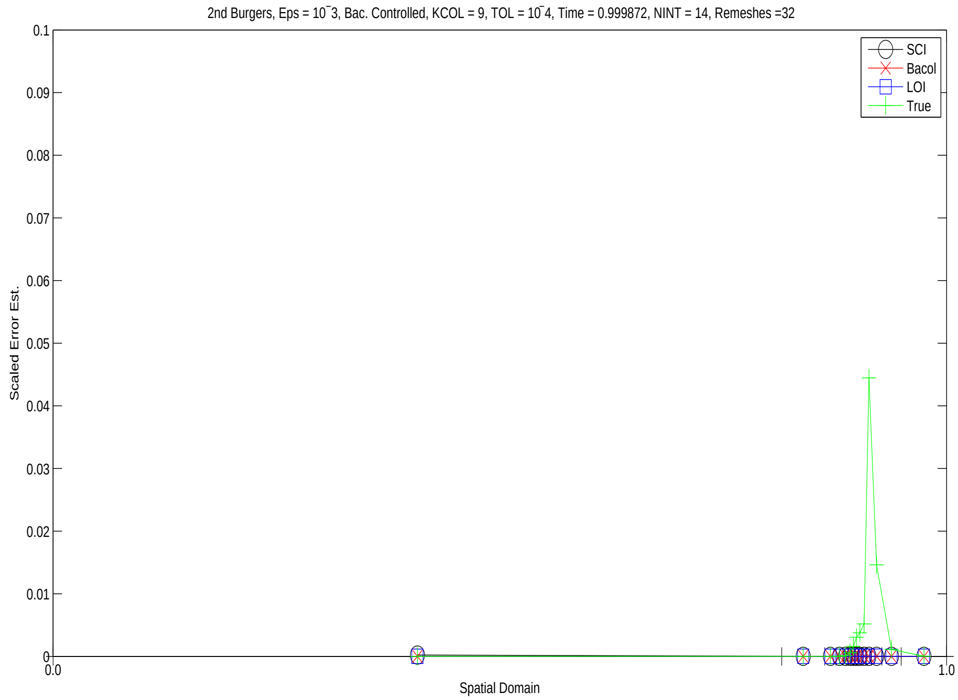


Figure 39: *When the BACOL estimates control the mesh, the SCI estimates on the boundary subintervals are too large. All estimates underestimate the error in the layer region. When the SCI estimates control the mesh, the SCI estimates on the boundary subintervals are OK but all estimates underestimate the error in the layer region.*



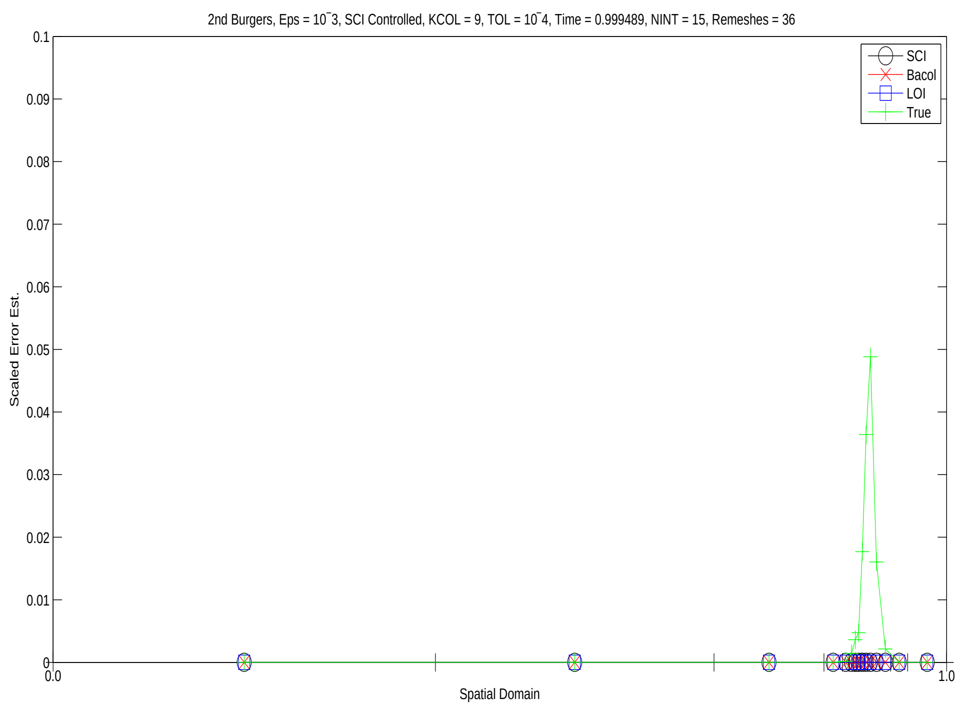
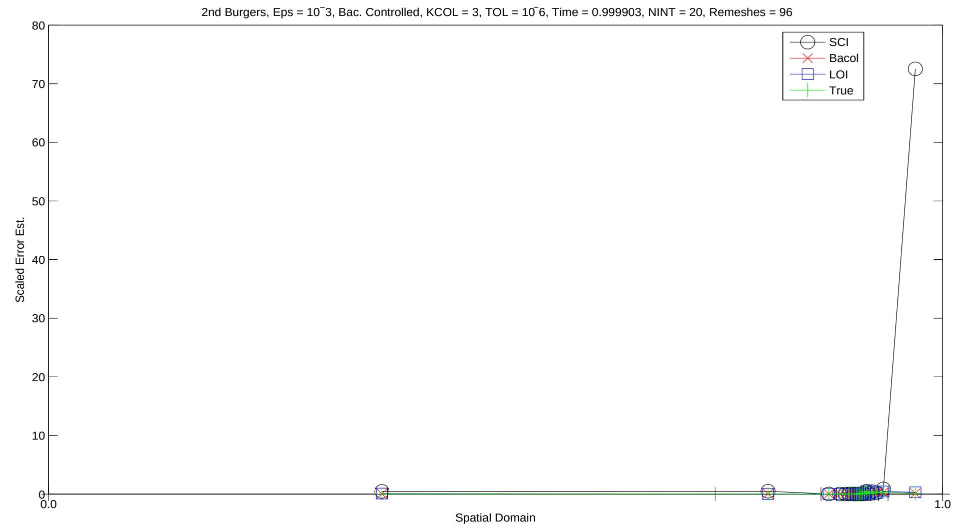


Figure 40: *When either the BACOL estimates or the SCI estimates control the mesh, all estimates are OK except in the layer region where all estimates underestimate the error.*

9.3.3 Results for $\epsilon = 10^{-3}$ and Tolerance $= 10^{-6}$



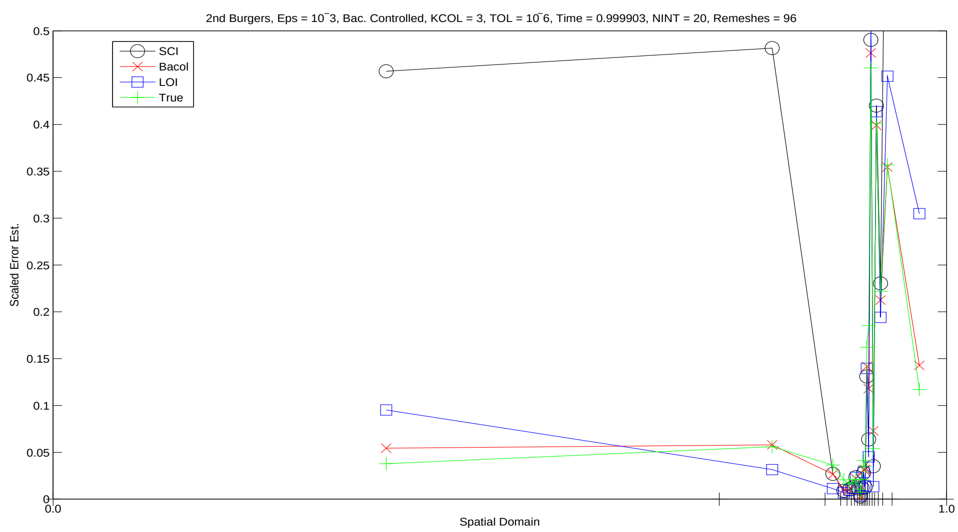
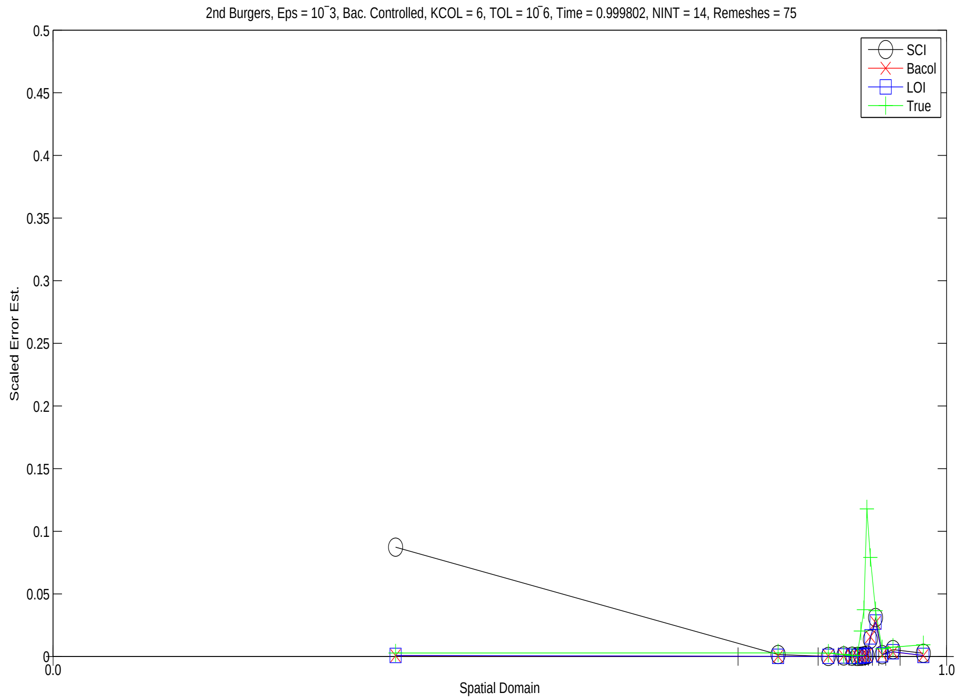


Figure 41: *These graphs show results when BACOL estimates control the mesh. In the top graph we see that the SCI estimate for the rightmost subinterval is far too large. In the bottom graph, the vertical axis is restricted to show a smaller region [0,0.5]; we see that the SCI estimate for the leftmost subinterval is also too large. Estimates of the error in the layer region may be reasonable. When the SCI estimates control the mesh, the time integration fails at the starting point.*



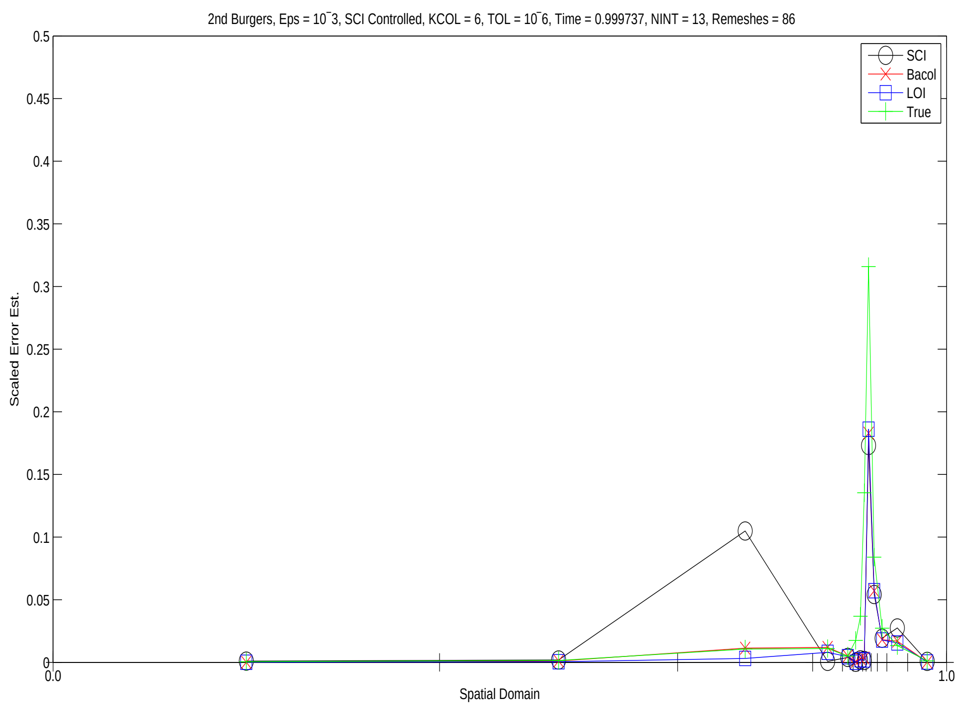
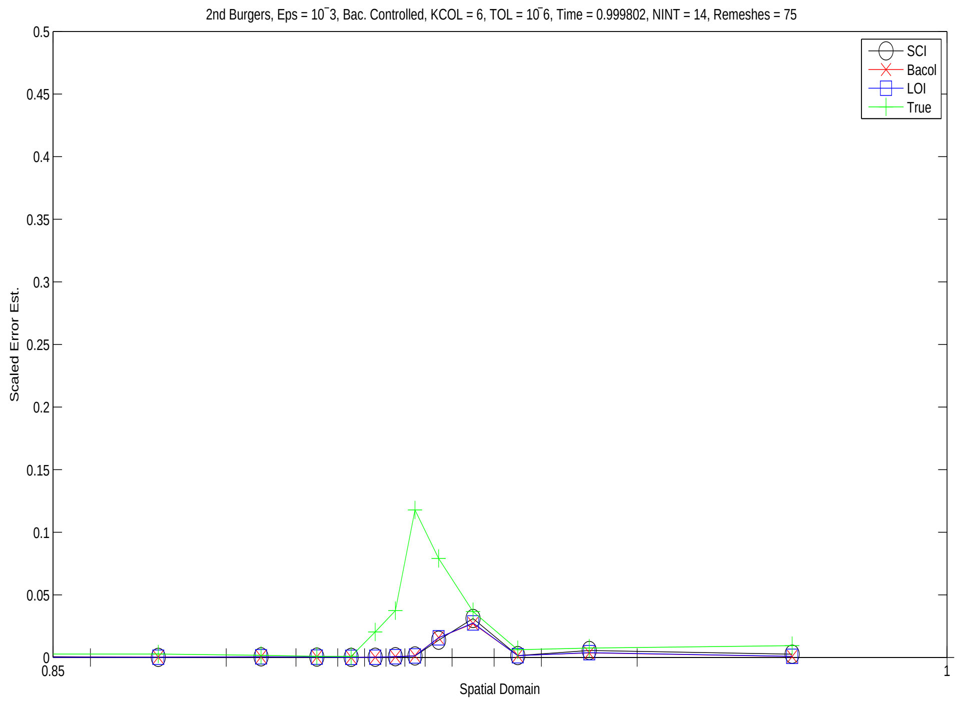


Figure 42: *When the BACOL estimates control the mesh, the SCI estimate for the leftmost subinterval is too large and the estimates within the layer region are too small. When the SCI estimates control the mesh, there is better estimation of the error throughout the interval, except for the third subinterval from the left where the SCI estimate is too large.*



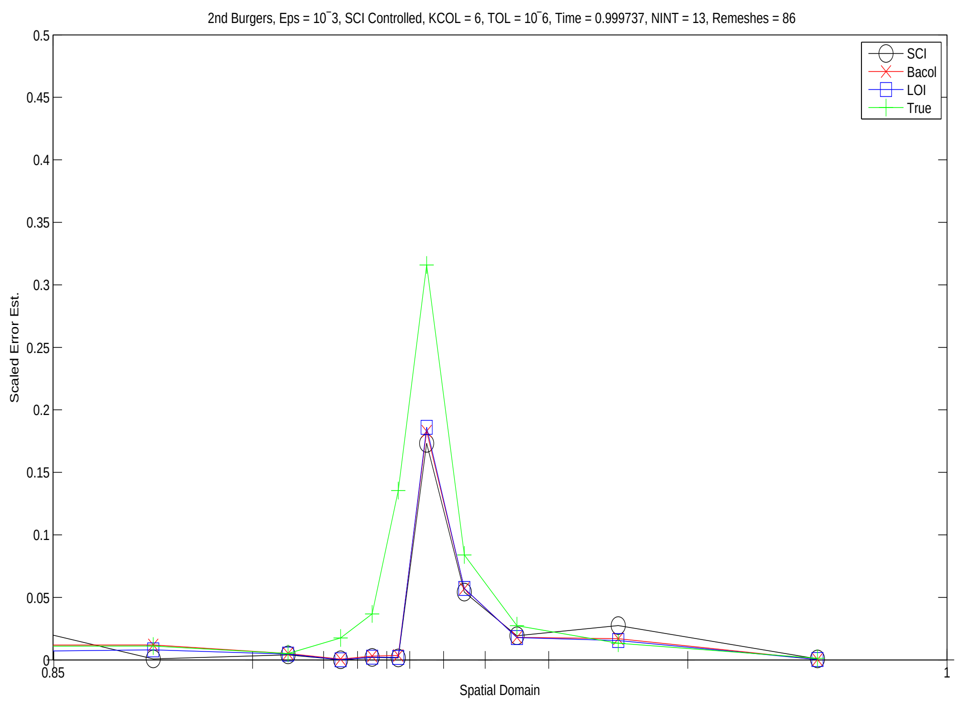


Figure 43: *Results for the layer region. When the BACOL estimates control the mesh, the estimates within the layer region are too small. When the SCI estimates control the mesh, there is better estimation of the error within the layer region.*

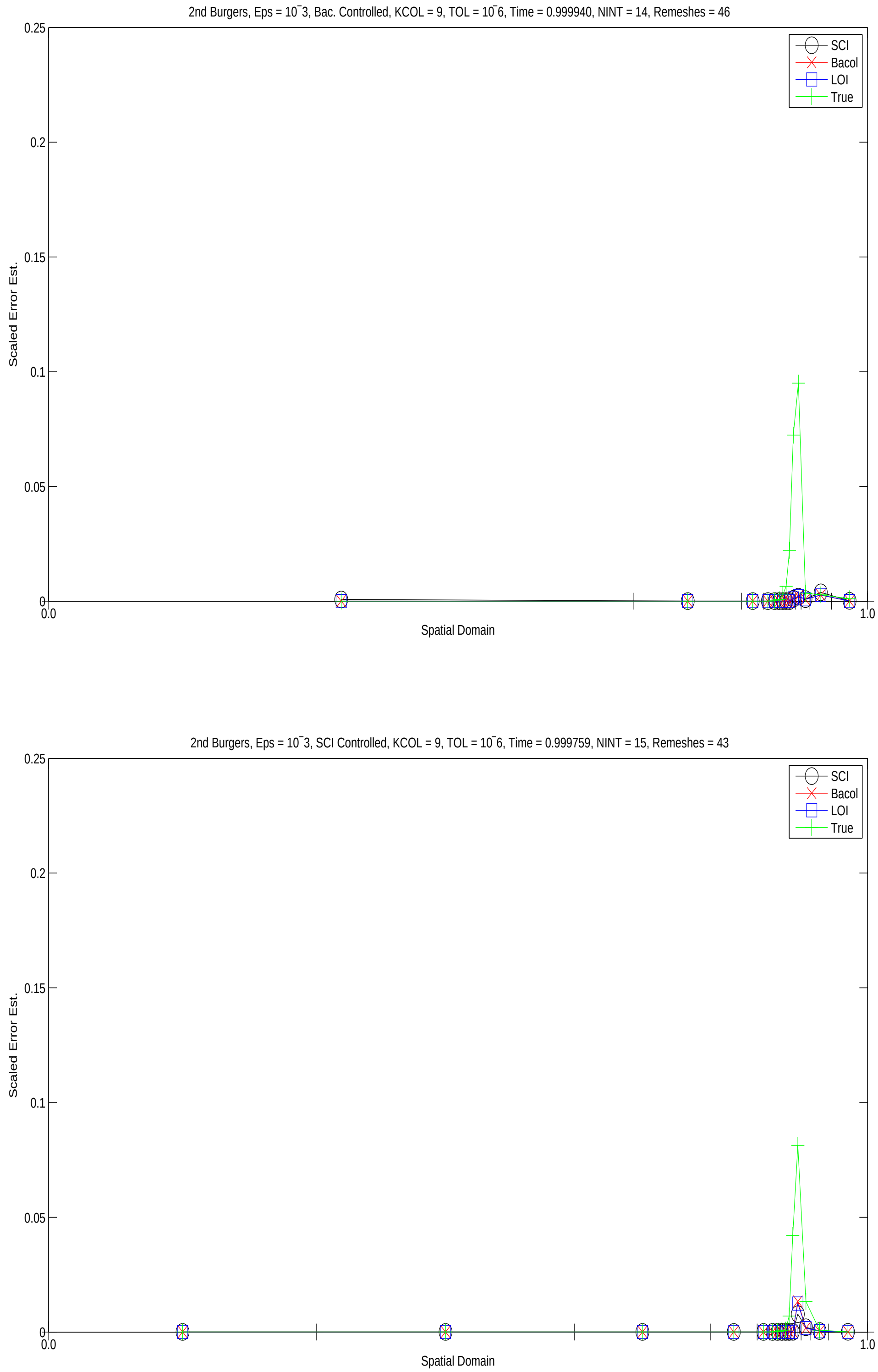


Figure 44: *When either the BACOL estimates or the SCI estimates control the mesh, all estimates are OK except in the layer region where all estimates underestimate the error.*

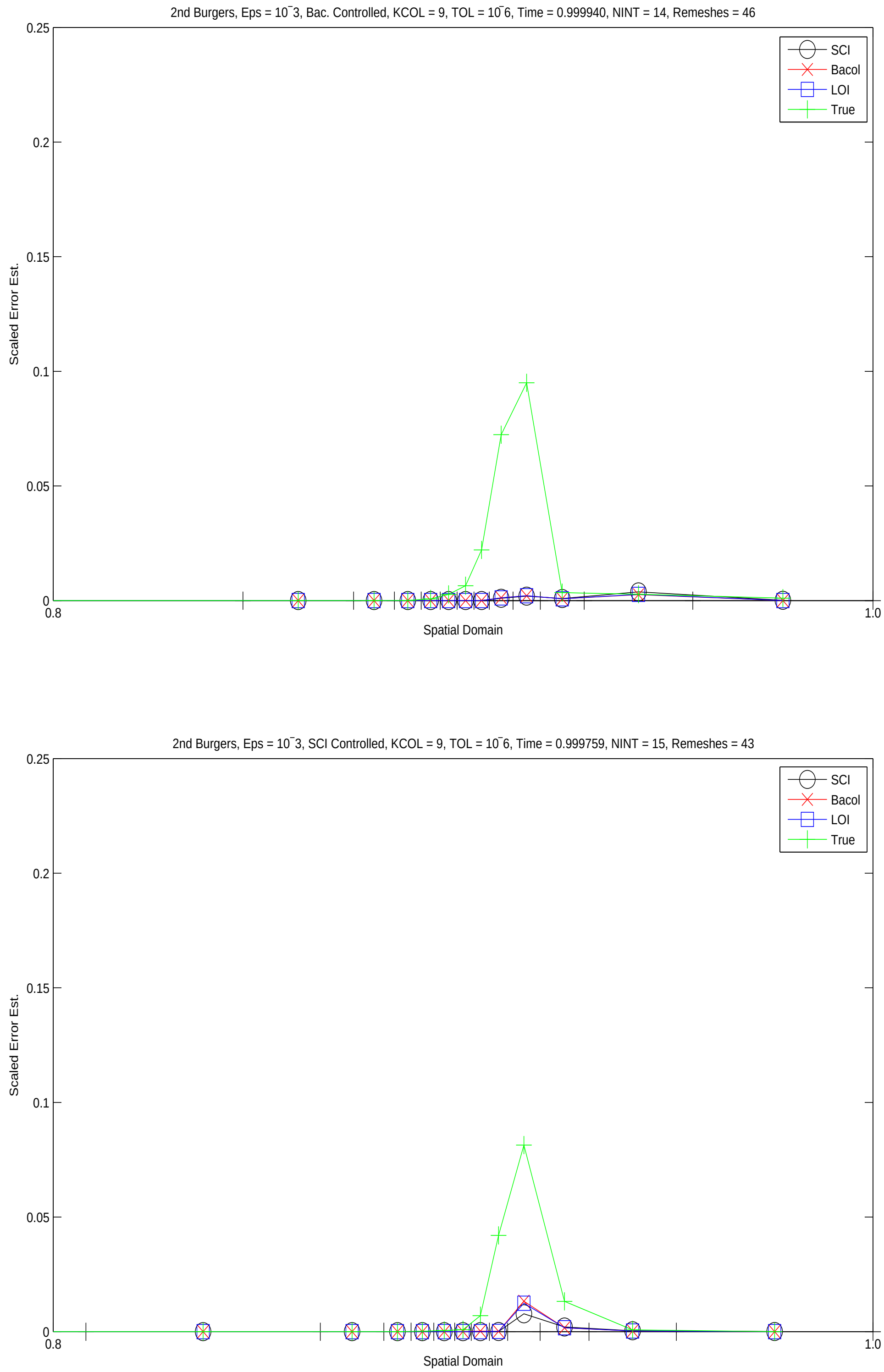
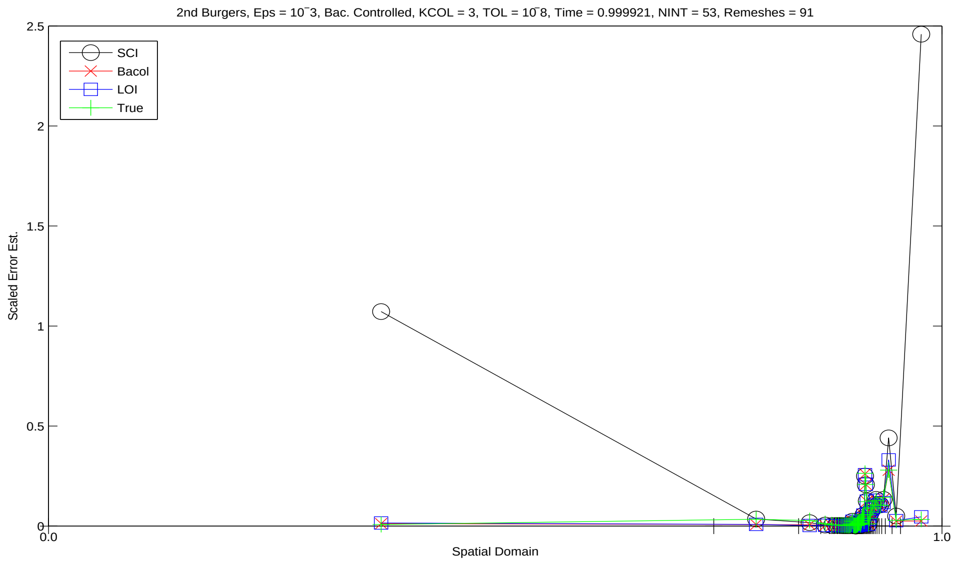


Figure 45: *Results for the layer region. When either the BACOL estimates or the SCI estimates control the mesh, in the layer region all estimates underestimate the error.*

9.3.4 Results for $\epsilon = 10^{-3}$ and Tolerance $= 10^{-8}$



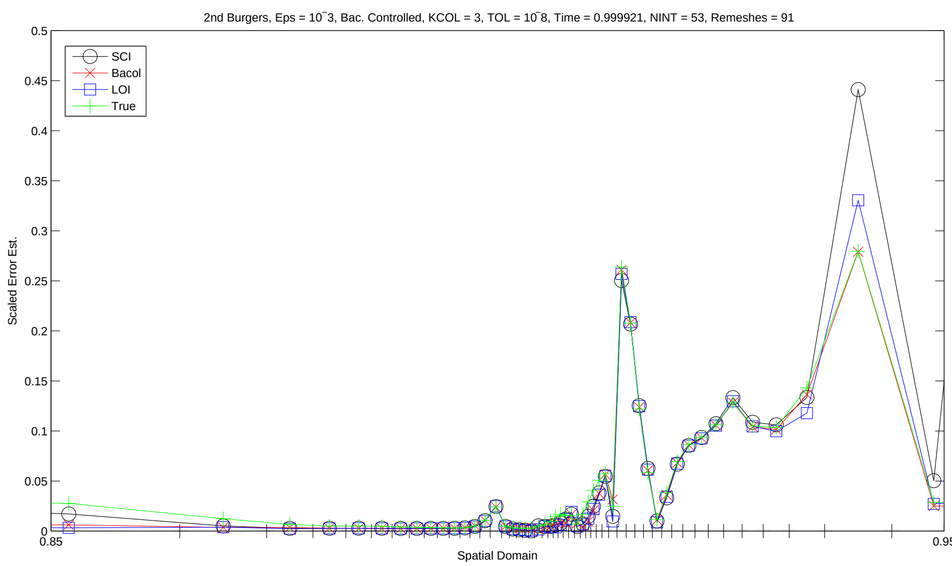
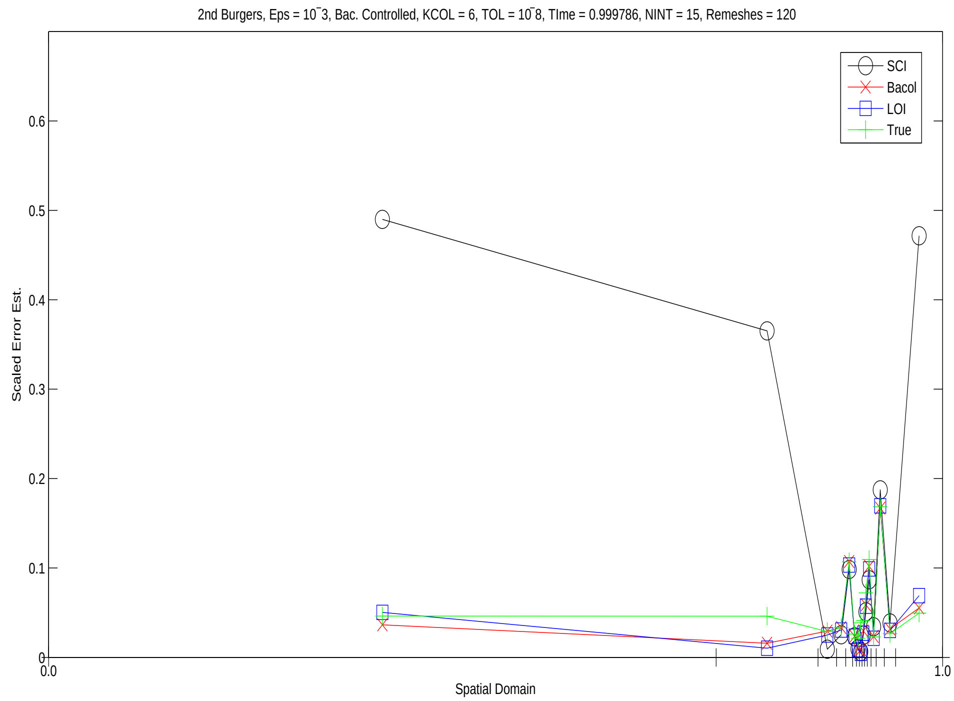


Figure 46: *When the BACOL estimates control the mesh, there estimates are OK except for the SCI estimates on the boundary subintervals. Within the layer region, there is good estimation of the error by all schemes. When the SCI estimates control the mesh, the time integration fails at the starting point.*



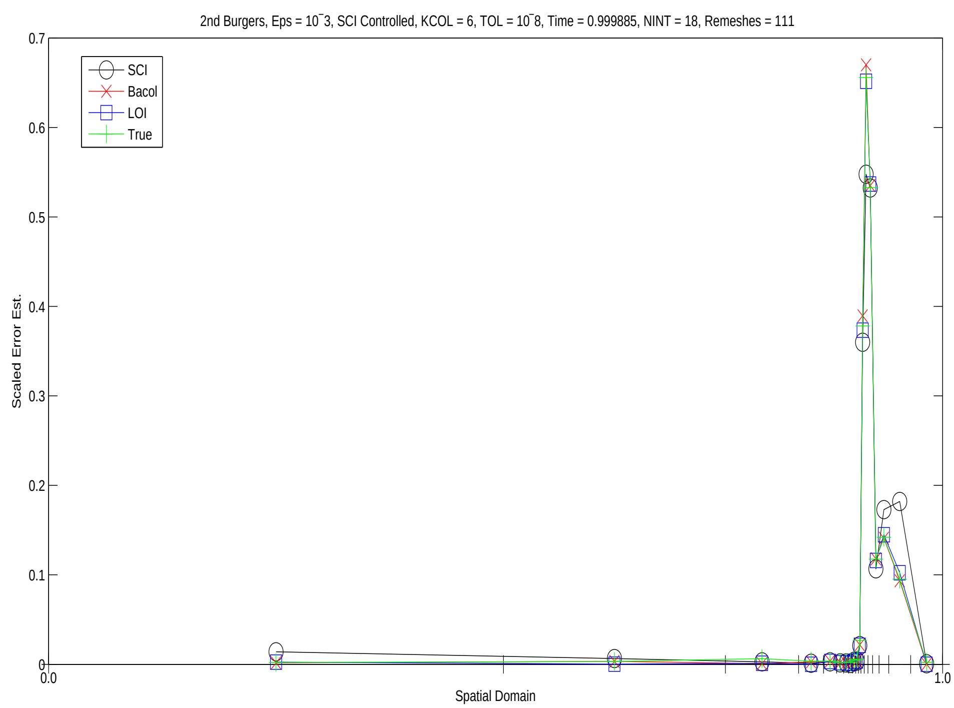
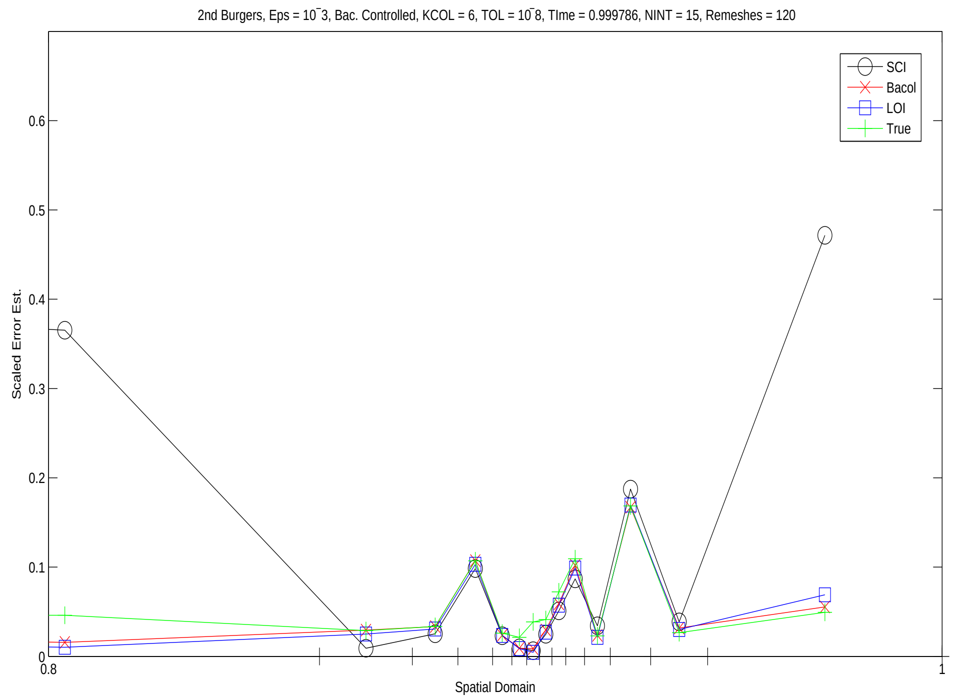


Figure 47: *When the BACOL estimates control the mesh, all estimates are OK throughout the problem interval, even in the layer region, except for the SCI estimates at the boundary subintervals. When the SCI estimates control the mesh, we see good estimation of the error by all schemes throughout the spatial domain.*



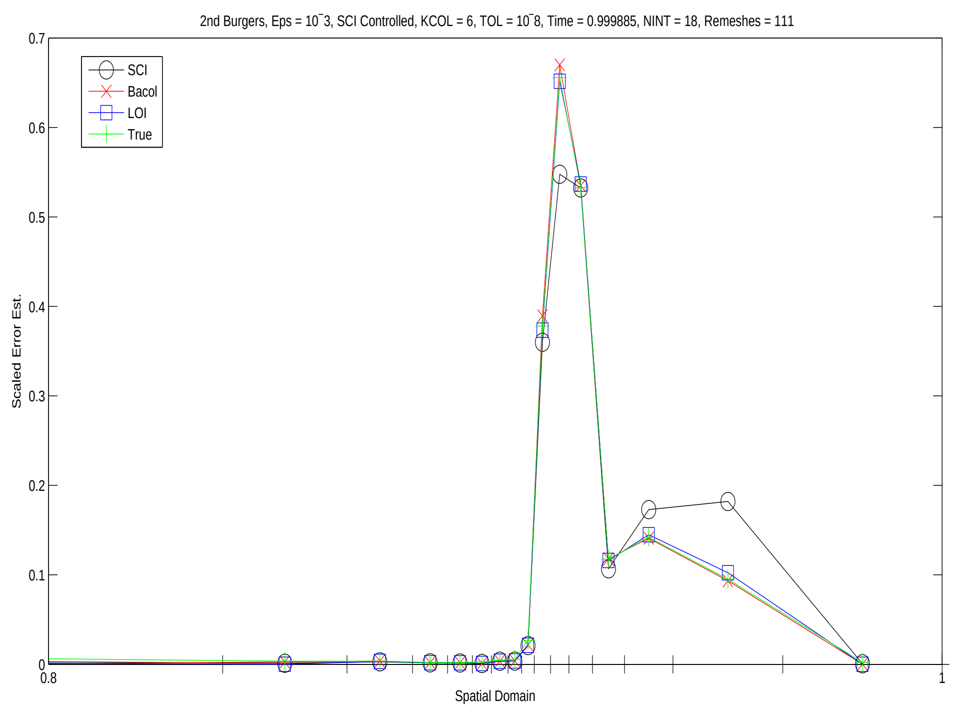
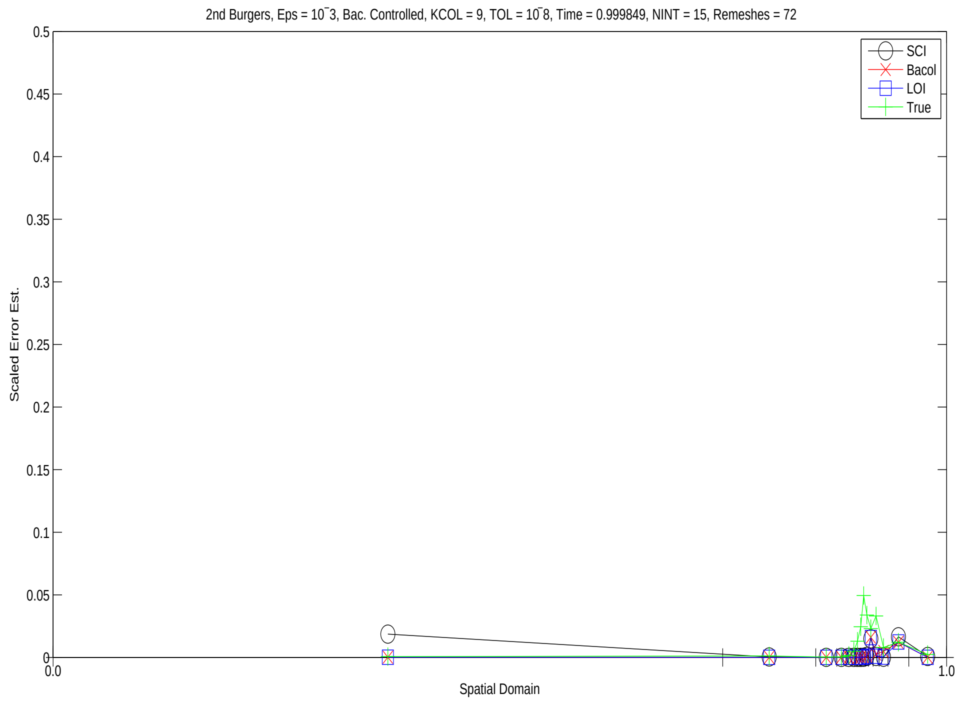


Figure 48: *Results within the layer region. When the BACOL estimates control the mesh, all estimates are OK in the layer region except for the SCI estimates on the transition subintervals. When the SCI estimates control the mesh, we see good estimation of the error by all schemes throughout the layer region.*



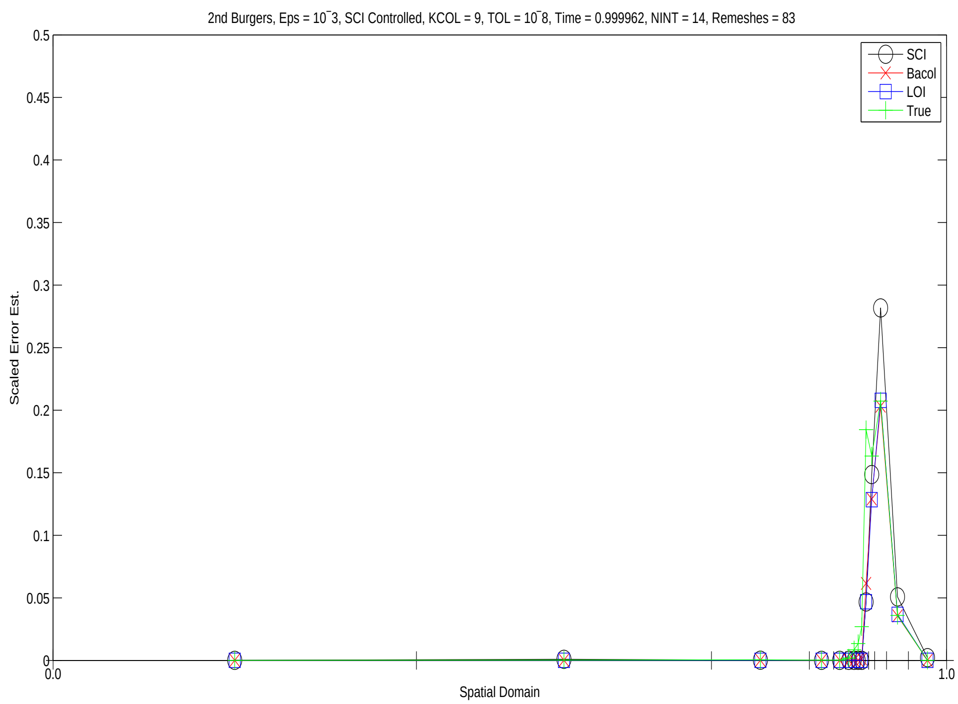
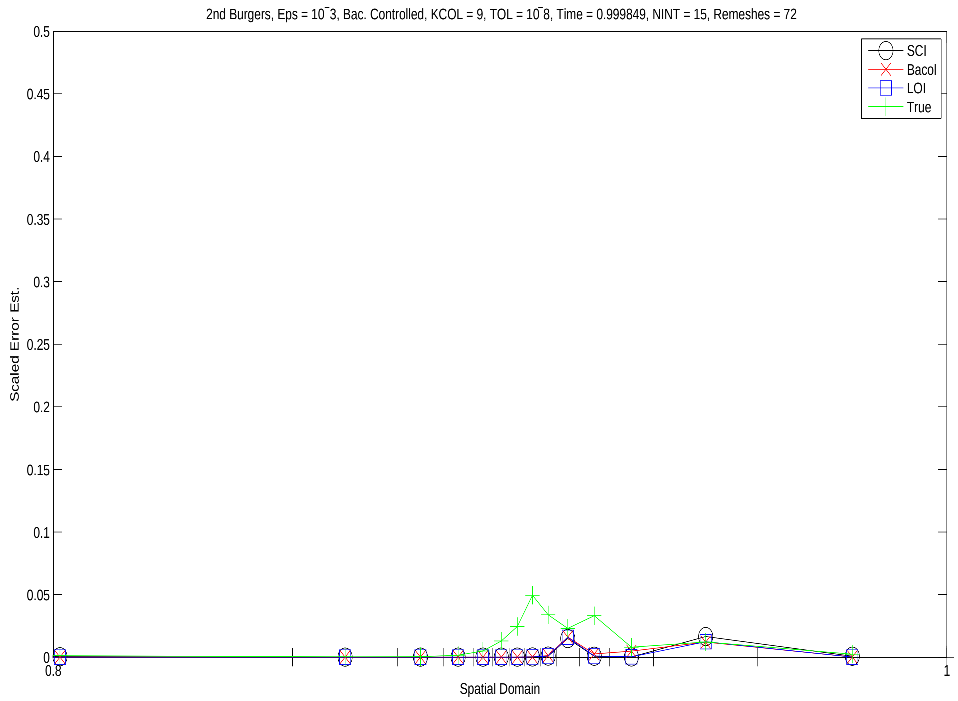


Figure 49: *When either the BACOL estimates or the SCI estimates control the mesh, all estimates are OK throughout the problem interval, except that when BACOL estimates control the mesh, we see underestimation of the error by all schemes within the layer region.*



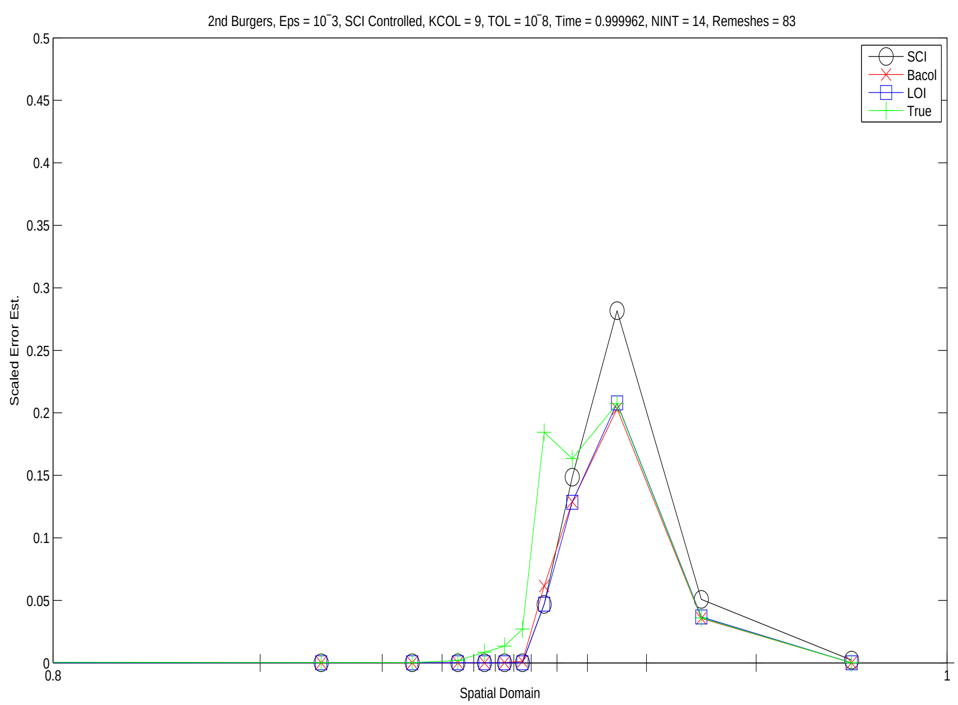


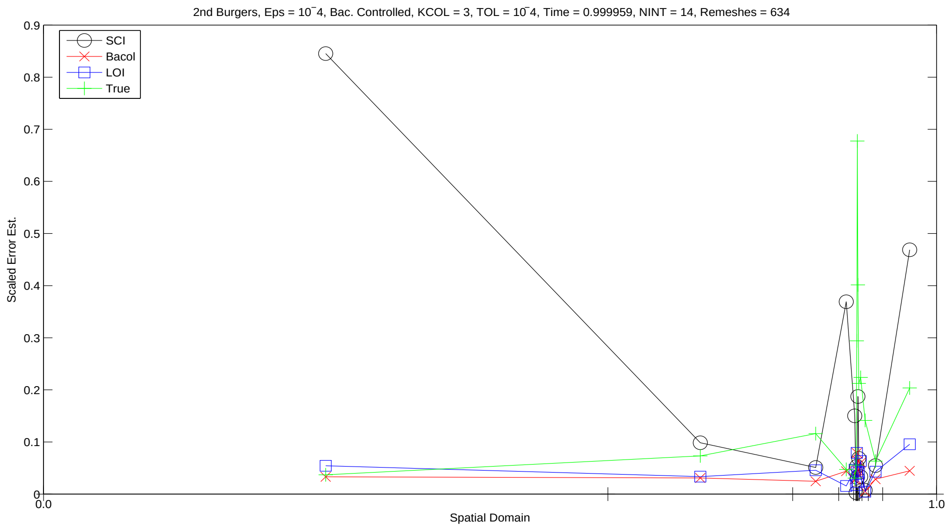
Figure 50: *Results for the layer region. When the BACOL estimates control the mesh, we see underestimation of the error by all schemes within the layer region. When the SCI estimates control the mesh, we see generally good estimation of the error by all schemes.*

9.3.5 Summary of Results for $\epsilon = 10^{-3}$

TOL \ KCOL		3	6	9
		NINT, Remeshes	NINT, Remeshes	NINT, Remeshes
10^{-4}	BAC	15, 63	13, 45	14, 32
	SCI	NA	15, 41	15, 36
10^{-6}	BAC	20, 96	14, 75	14, 46
	SCI	NA	13, 86	15, 43
10^{-8}	BAC	53, 91	15, 120	15, 72
	SCI	NA	18, 111	14, 83

Table 7: *Comparing results for $KCOL = 6, 9$ we see generally comparable results for the number of subintervals in the final mesh and the total number of remeshings when the BACOL estimates control the mesh or when the SCI estimates control the mesh.*

9.3.6 Results for $\epsilon = 10^{-4}$ and Tolerance $= 10^{-4}$



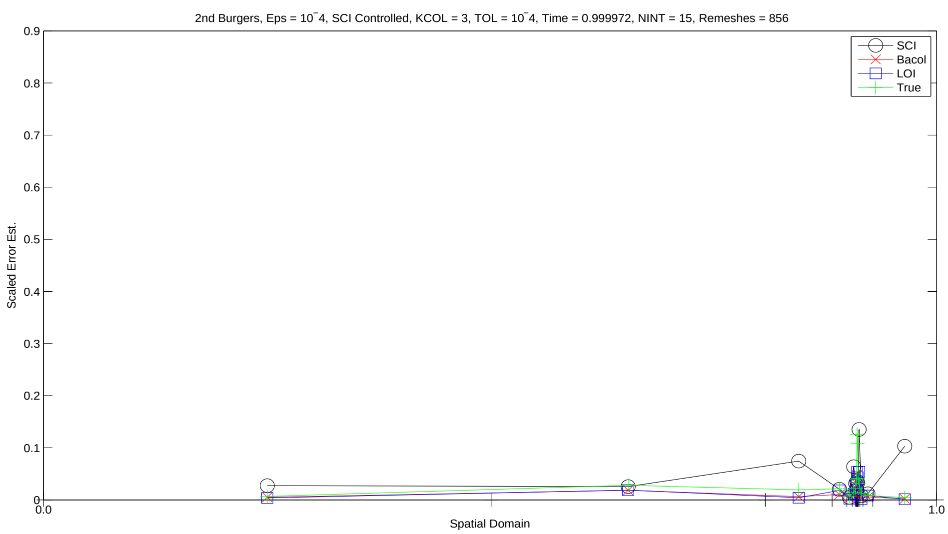
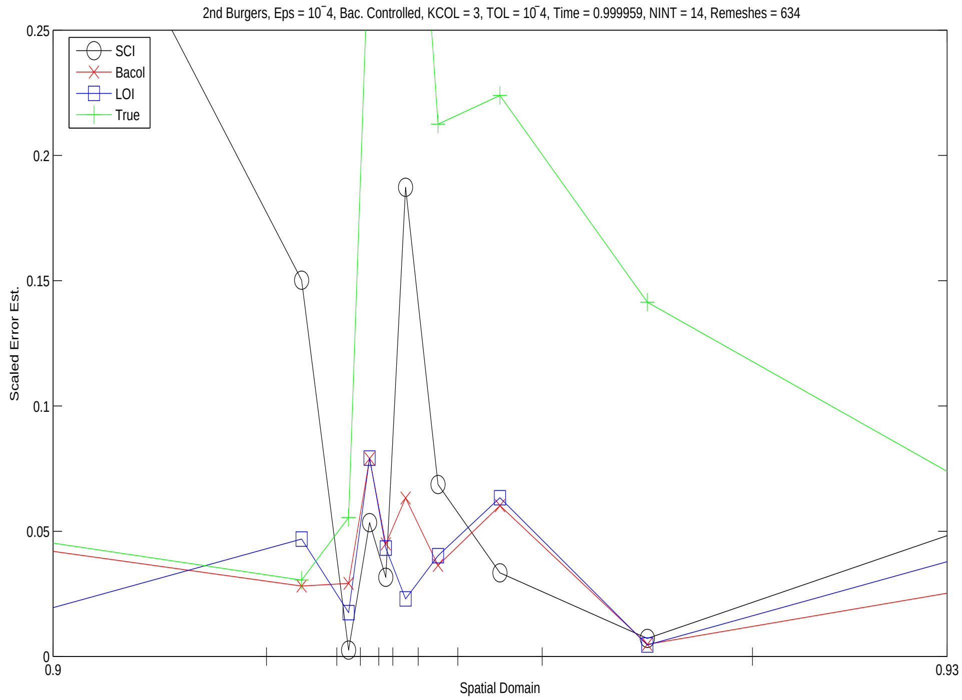


Figure 51: *When BACOL estimates control the mesh, the SCI estimates for the boundary subintervals are too large and all the schemes underestimate the error in the layer region. When the SCI estimates control the error, the estimates for all schemes are OK over the whole spatial domain. The SCI estimates slightly overestimate the error on several subintervals adjacent to the layer region. Note that the* $KCOL = 3$ *case for SCI control works in this case.*



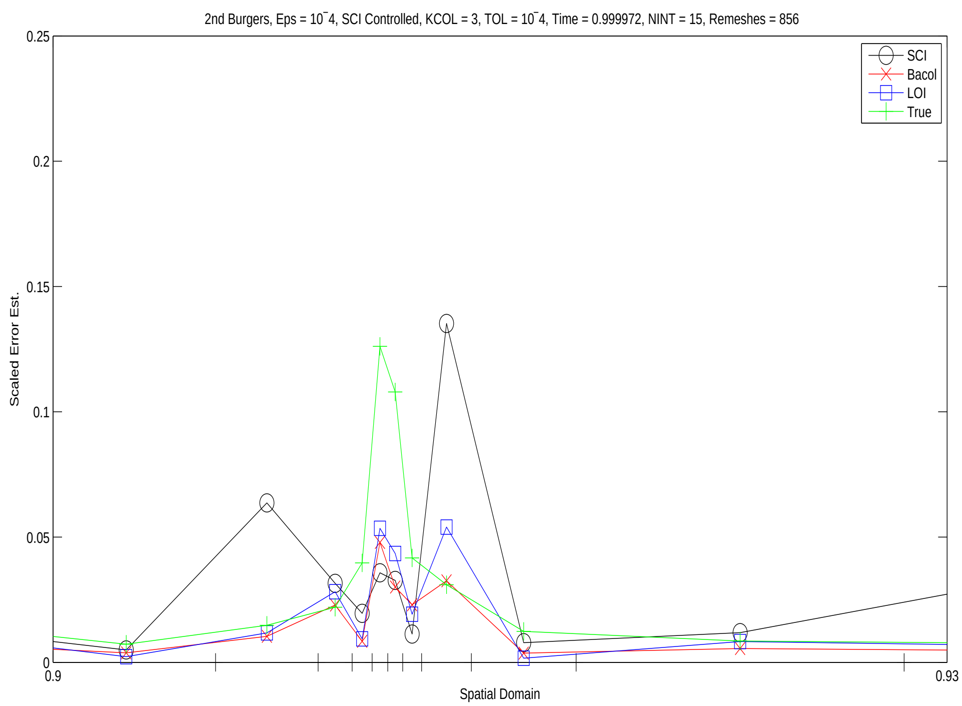


Figure 52: *Results for the layer region. When BACOL estimates control the mesh, all schemes underestimate the error in the layer region. When the SCI estimates control the error, the are generally better, except for the SCI estimates on subintervals adjacent to the layer region, however there is still underestimation of the error within the layer region.*

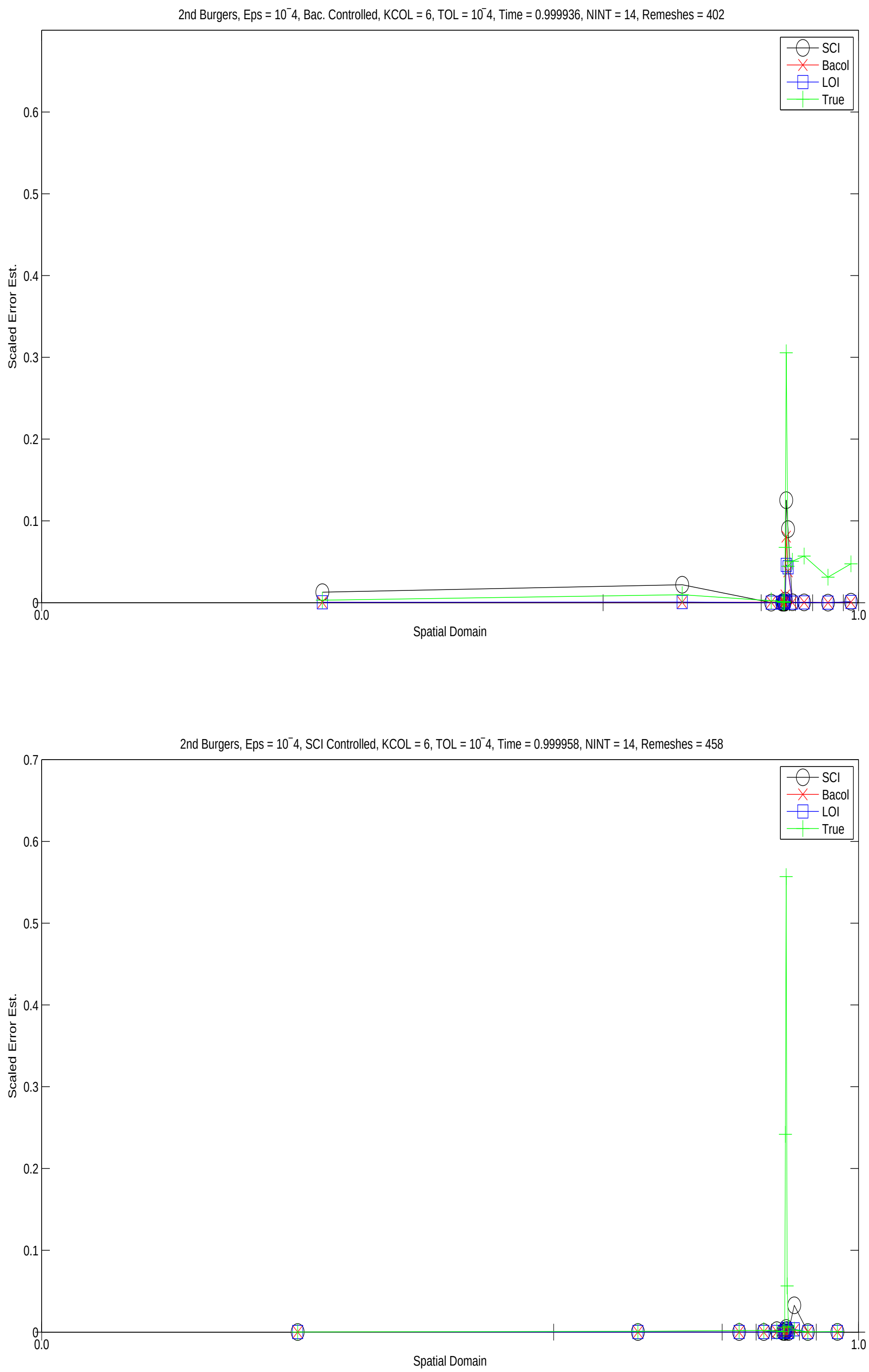
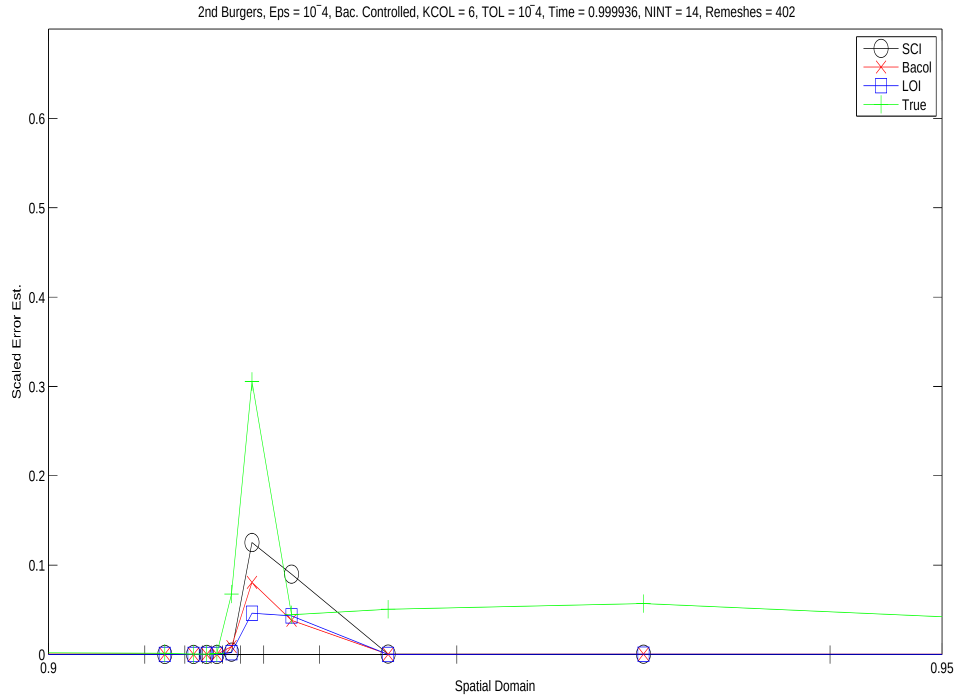


Figure 53: *When either the BACOL or SCI estimates control the mesh, there is good agreement among the estimates and the true error except for the layer region where all the schemes underestimate the error.*



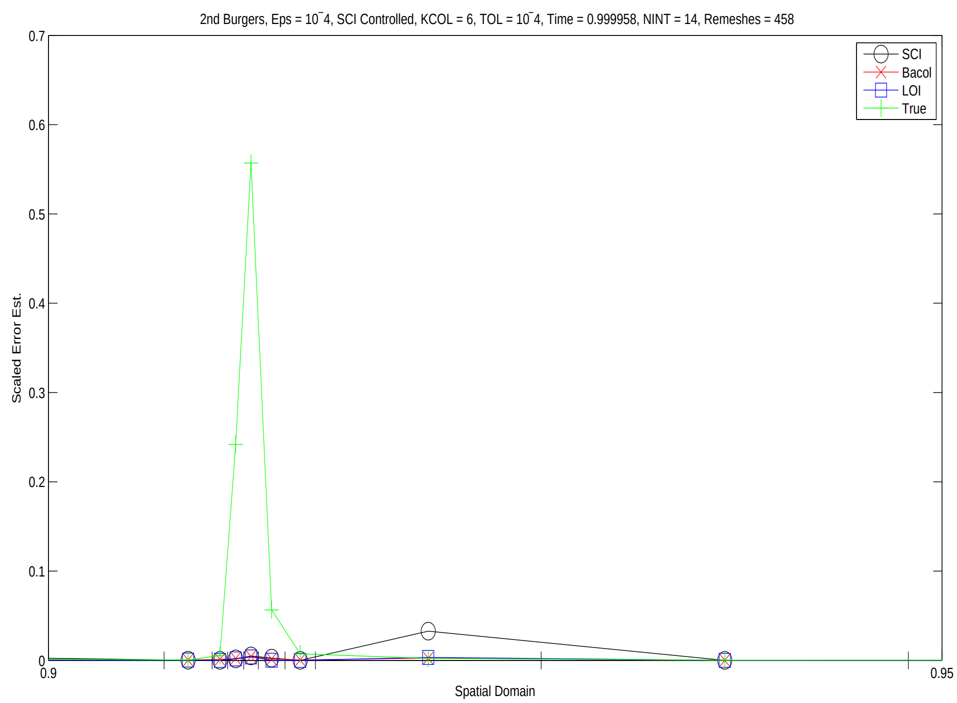
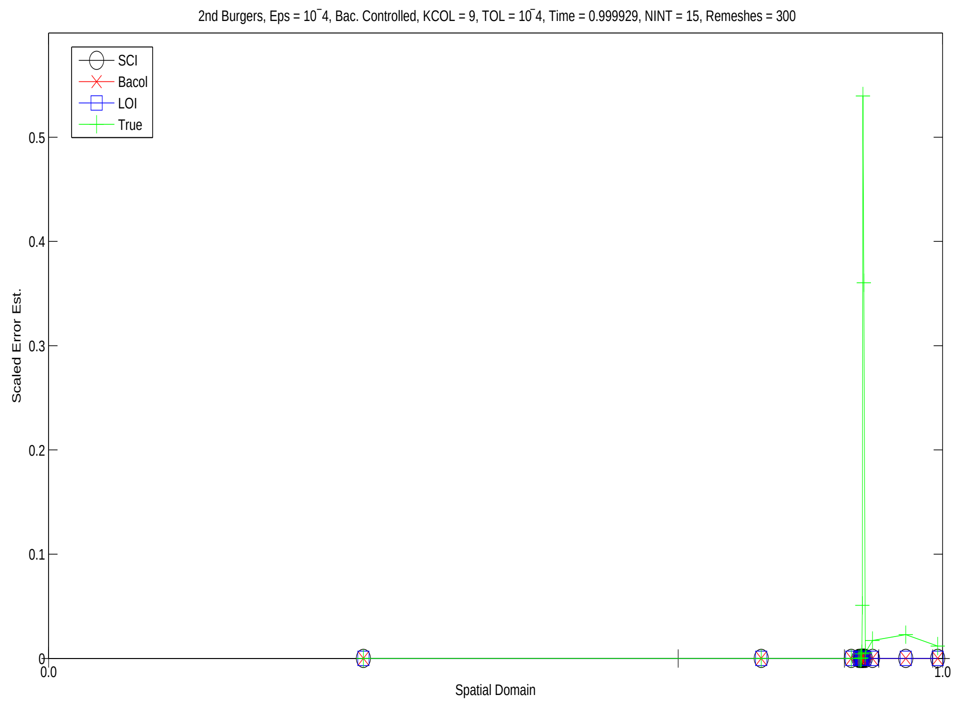


Figure 54: *Results for the layer region. When either the BACOL or SCI estimates control the mesh, in the layer region, all the schemes underestimate the error. The situation is somewhat worse when the SCI estimates control the mesh.*



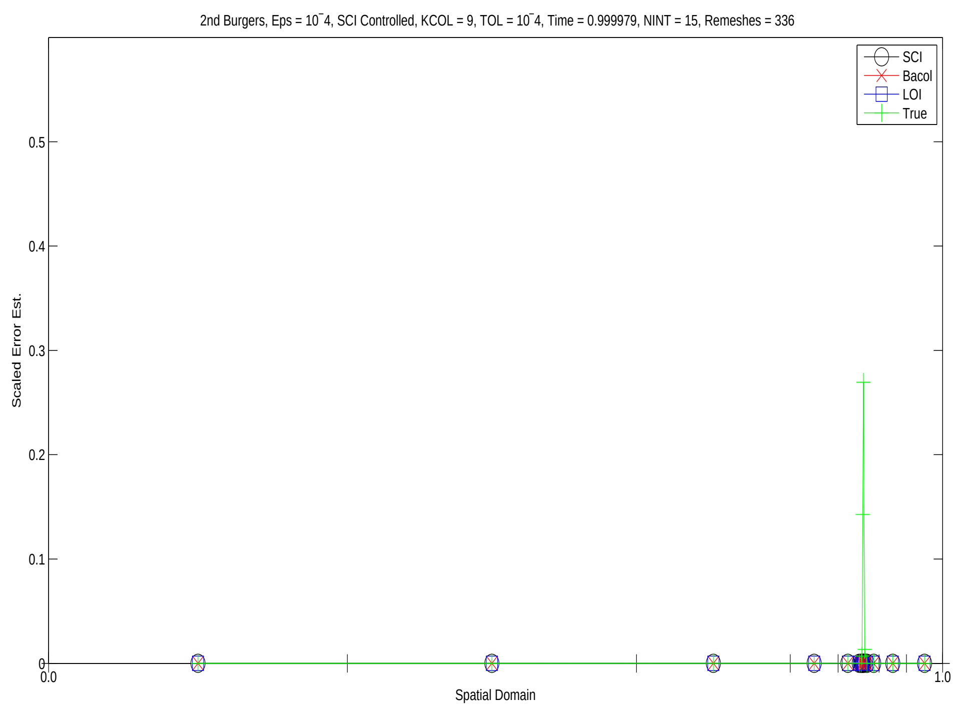
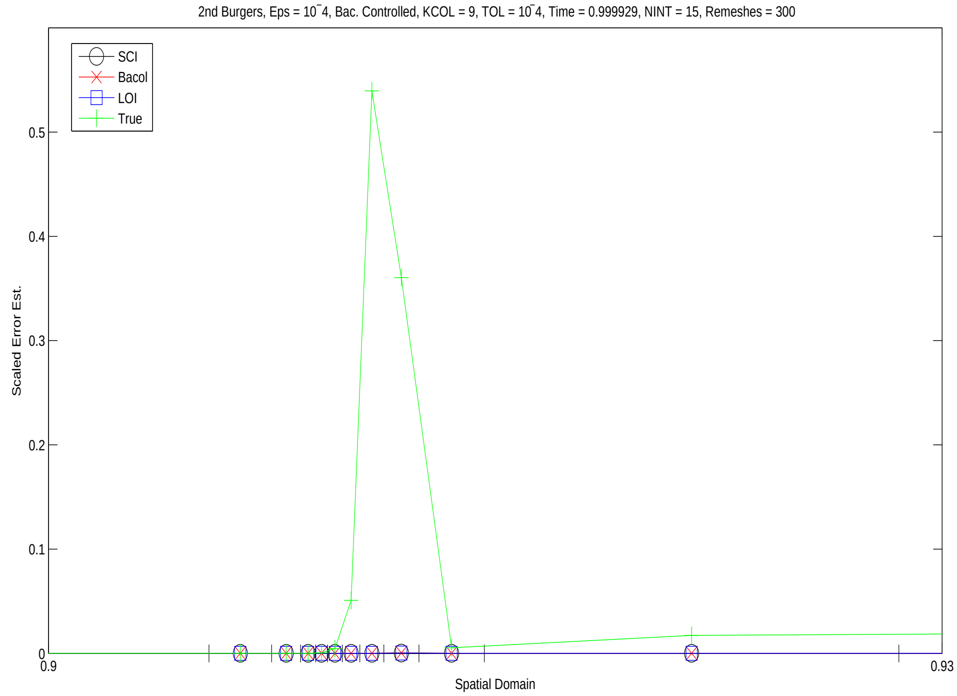


Figure 55: *When either the BACOL or SCI estimates control the mesh, there is good agreement among the estimates and the true error except for the layer region where all the schemes underestimate the error.*



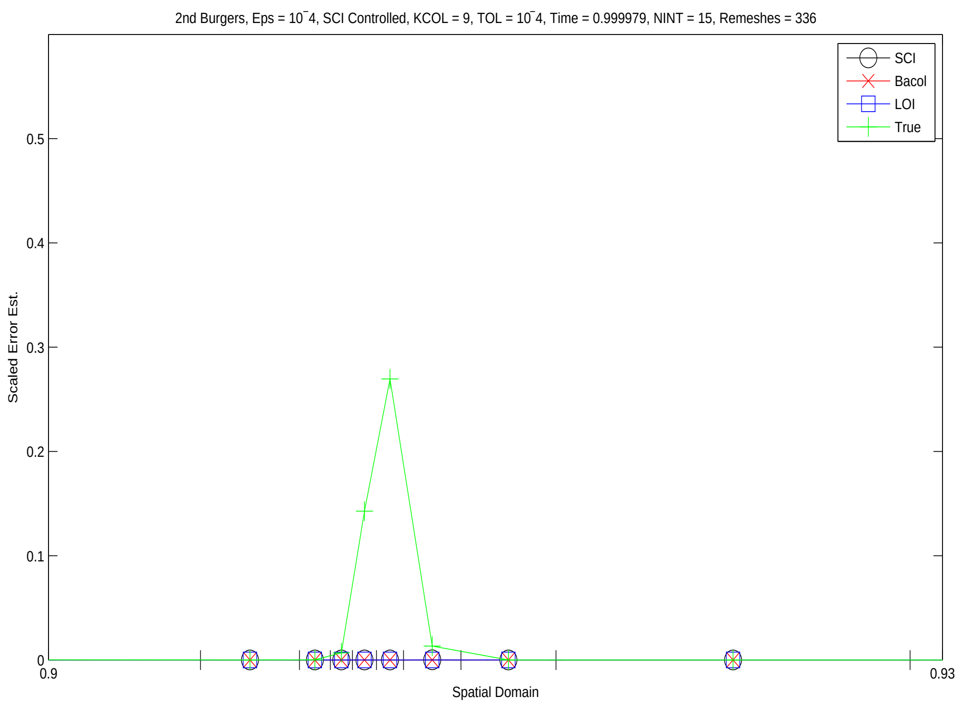
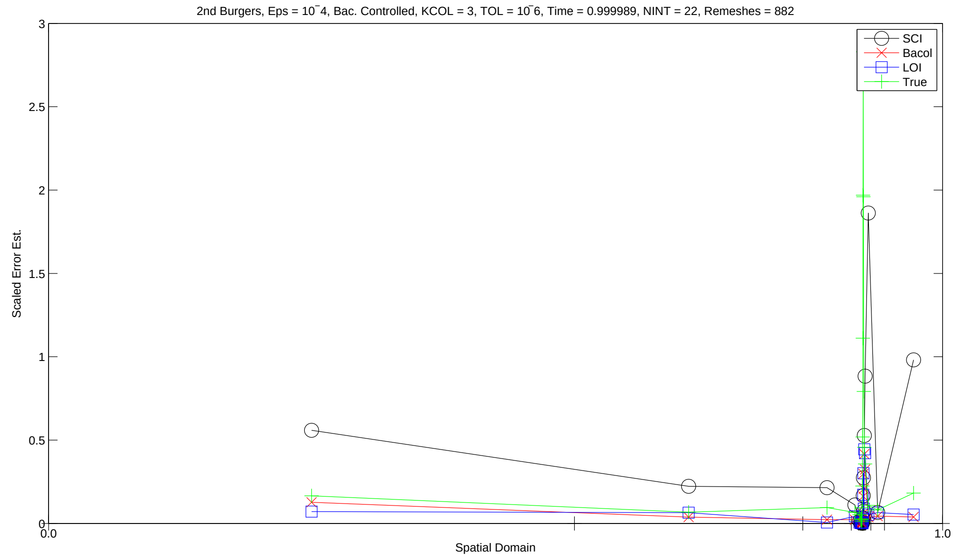


Figure 56: *Results for the layer region. When either the BACOL or SCI estimates control the mesh, in the layer region, all the schemes underestimate the error.*

9.3.7 Results for $\epsilon = 10^{-4}$ and Tolerance $= 10^{-6}$



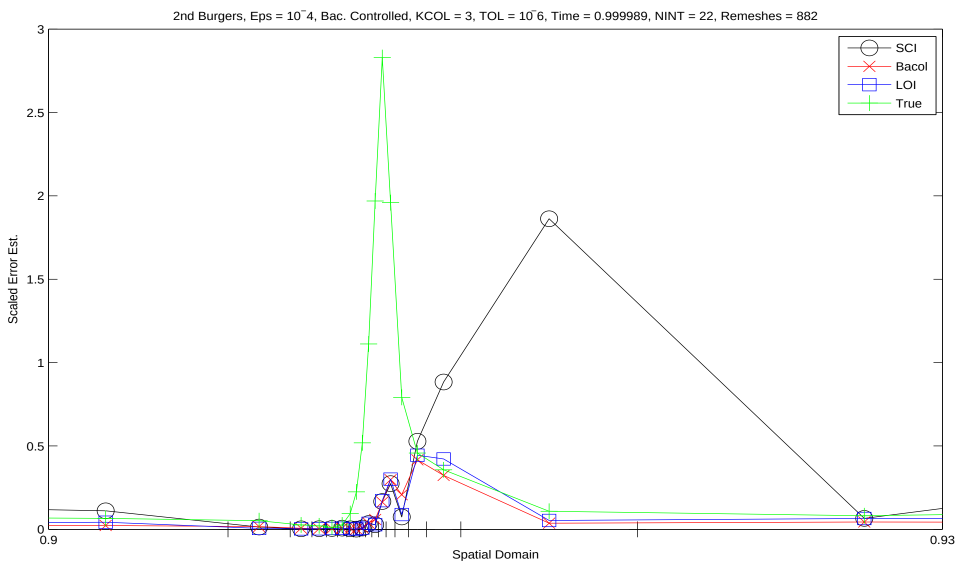
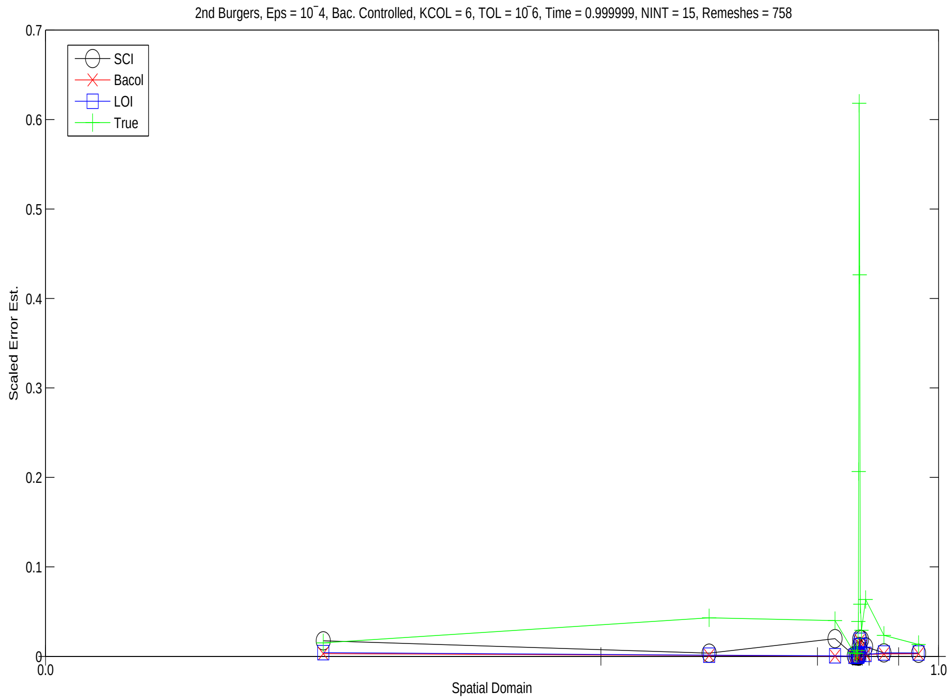


Figure 57: *When the BACOL estimates control the mesh, the SCI estimates are too large on the boundary and transition subintervals and the error in the layer region is underestimated by all schemes. When the SCI estimates control the mesh, the time integration fails at the starting point.*



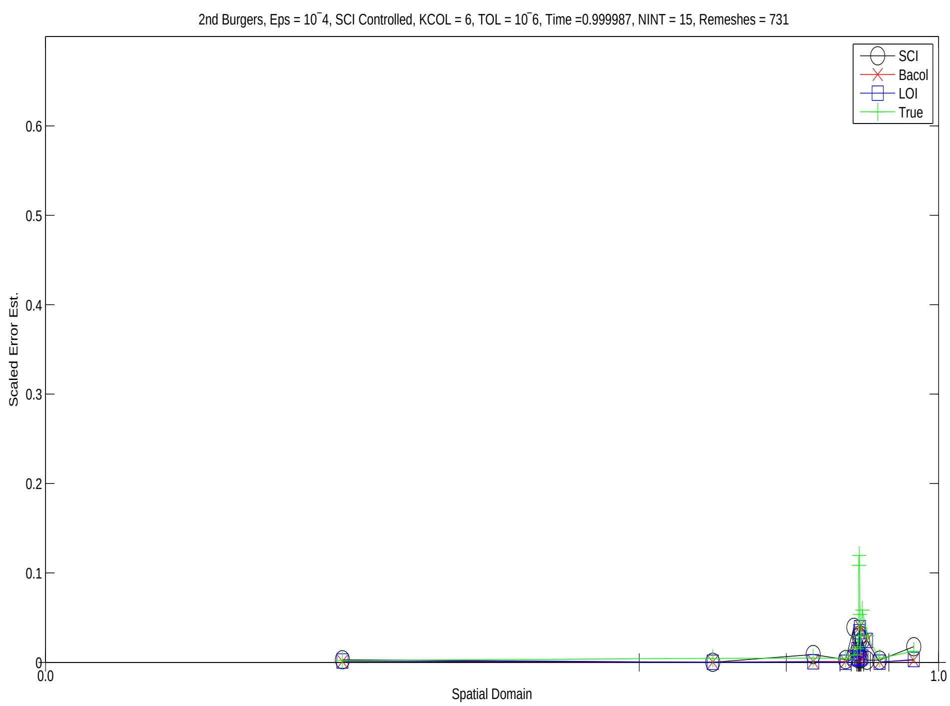


Figure 58: *With either the BACOL or SCI estimates in control of the mesh, the error estimates away from the layer region are reasonable; in the layer region all schemes underestimate the error.*

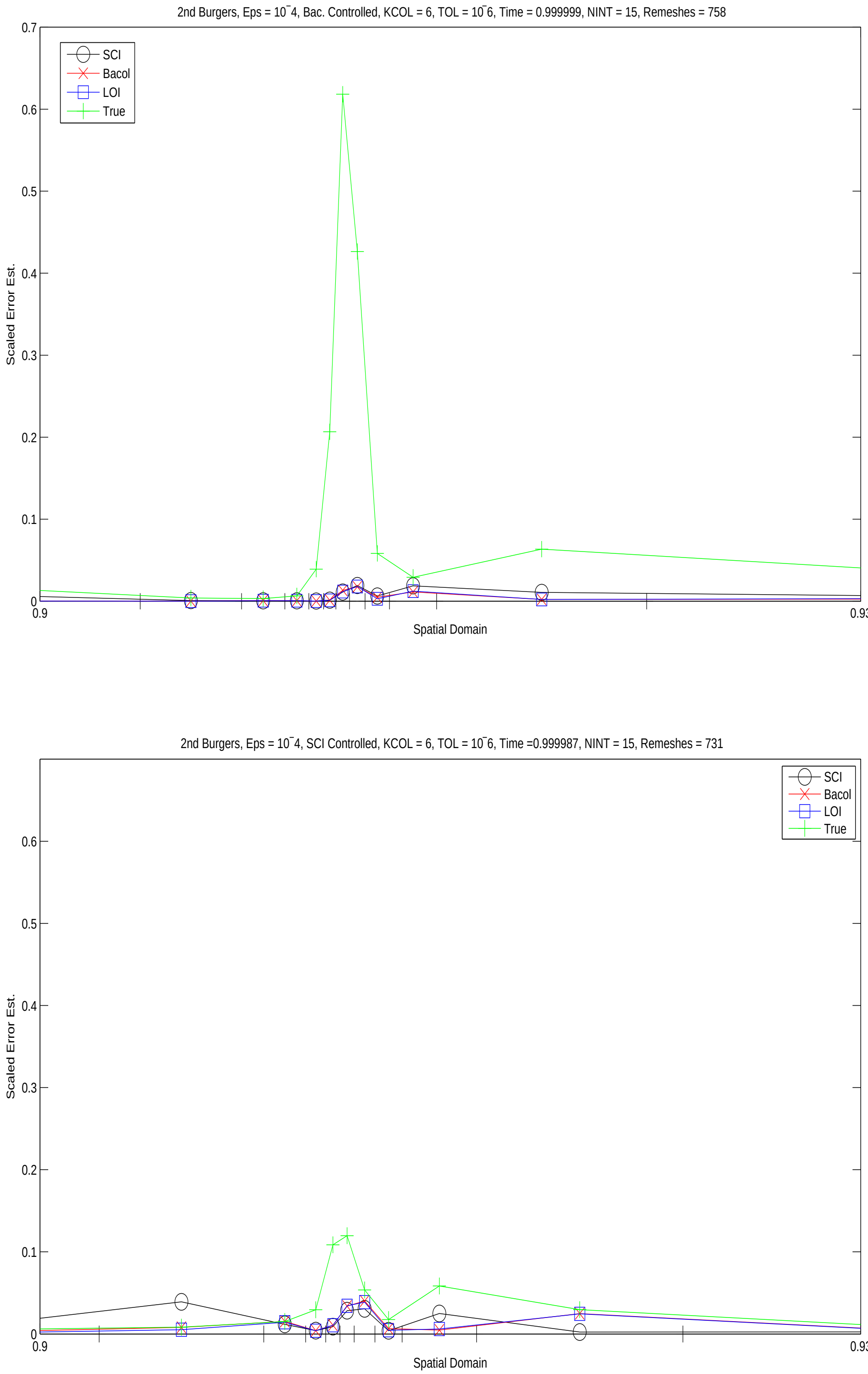
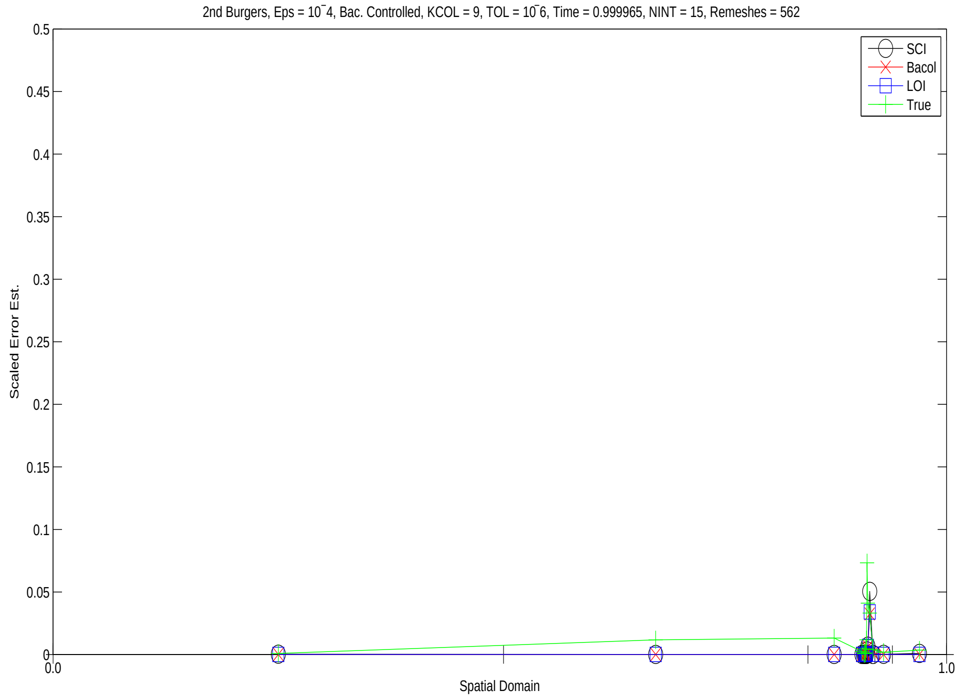


Figure 59: *Results only for the layer region. With either the BACOL or SCI estimates in control of the mesh, the error estimates for all schemes in the layer region underestimate the error. The situation is worse when the BACOL estimates control the mesh.*



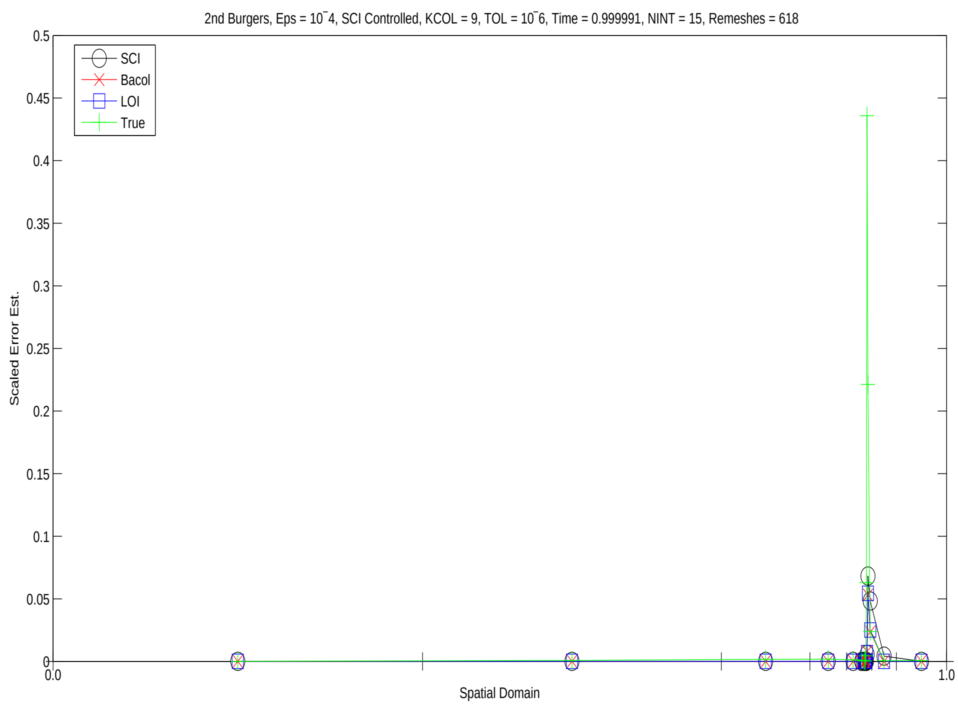
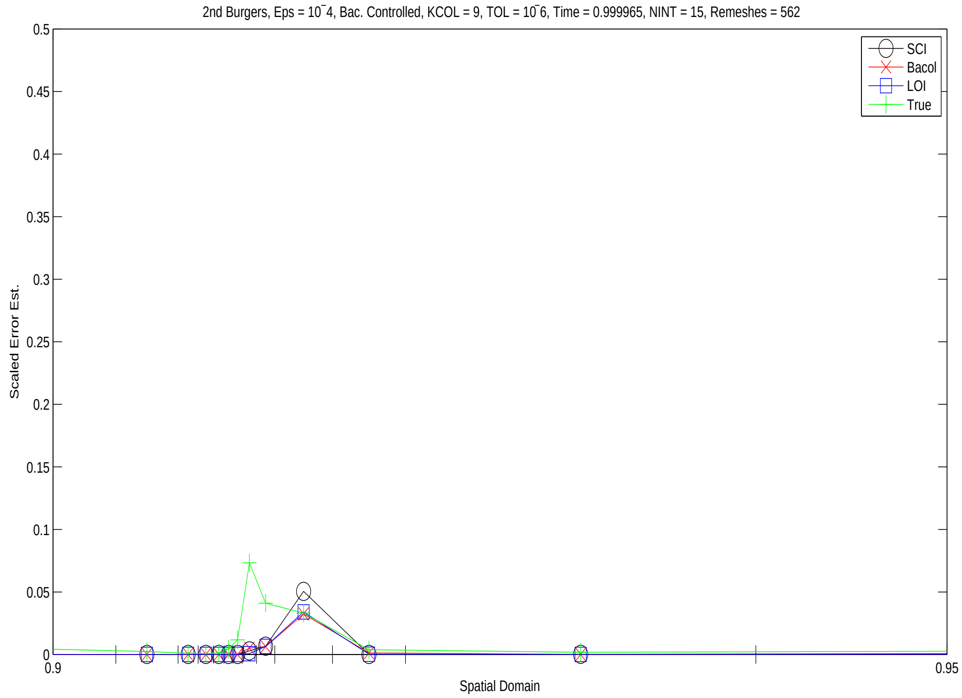


Figure 60: *With either the BACOL or SCI estimates in control of the mesh, the error estimates away from the layer region are reasonable; in the layer region all schemes underestimate the error to some extent. The situation is worse when the SCI estimates control the mesh.*



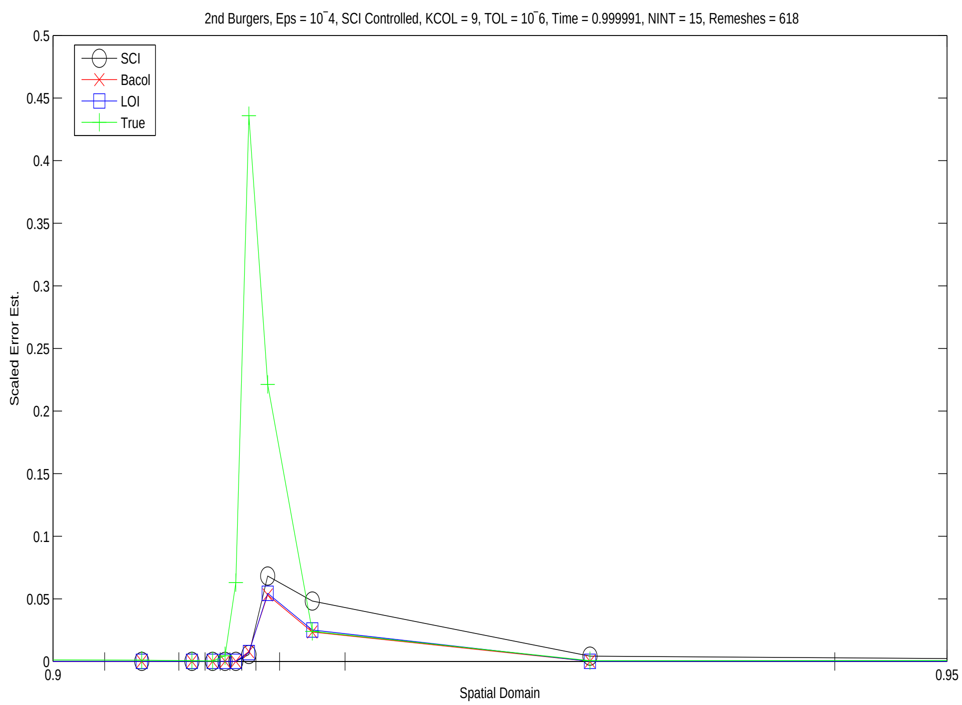
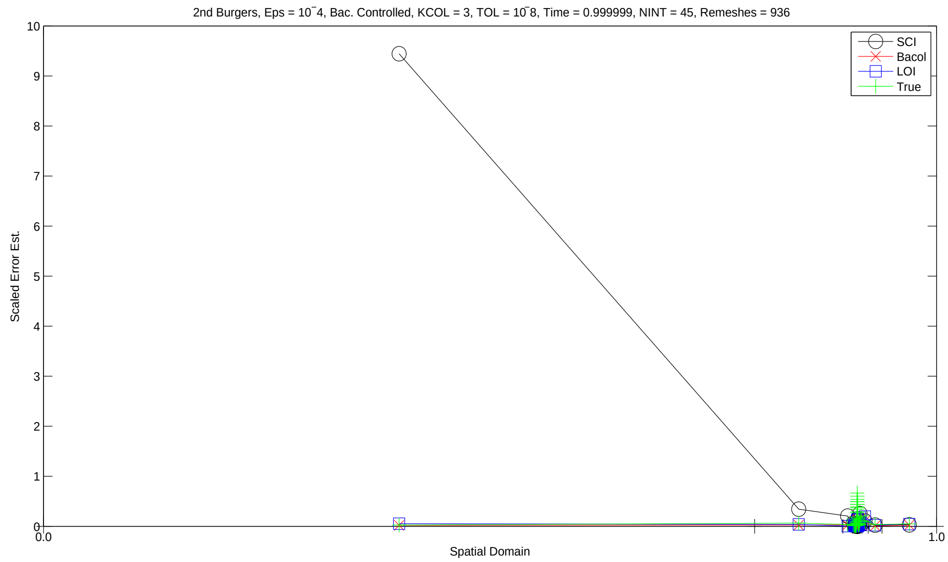


Figure 61: *Results only for the layer region. With either the BACOL or SCI estimates in control of the mesh, the error estimates for all schemes underestimate the error to some extent. The situation is worse when the SCI estimates control the mesh.*

9.3.8 Results for $\epsilon = 10^{-4}$ and Tolerance $= 10^{-8}$



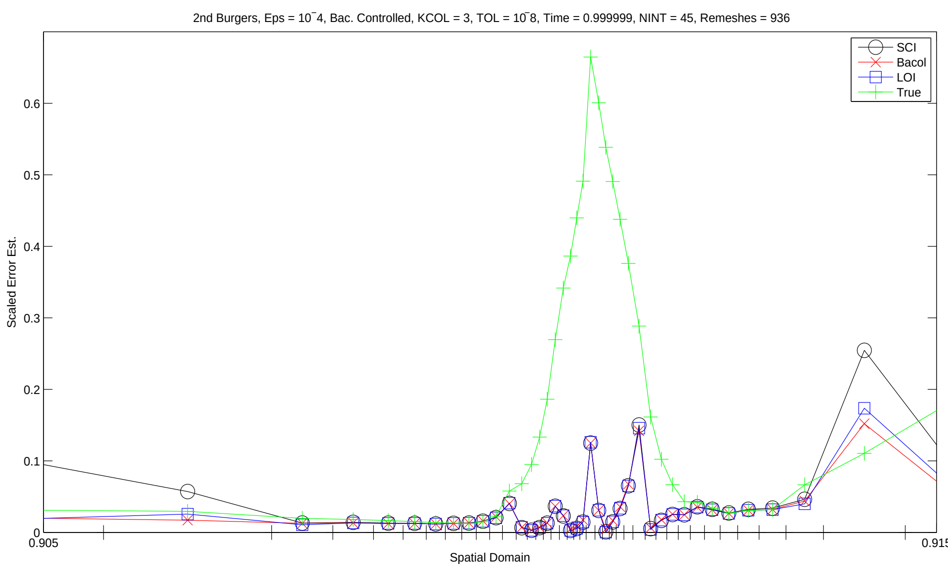
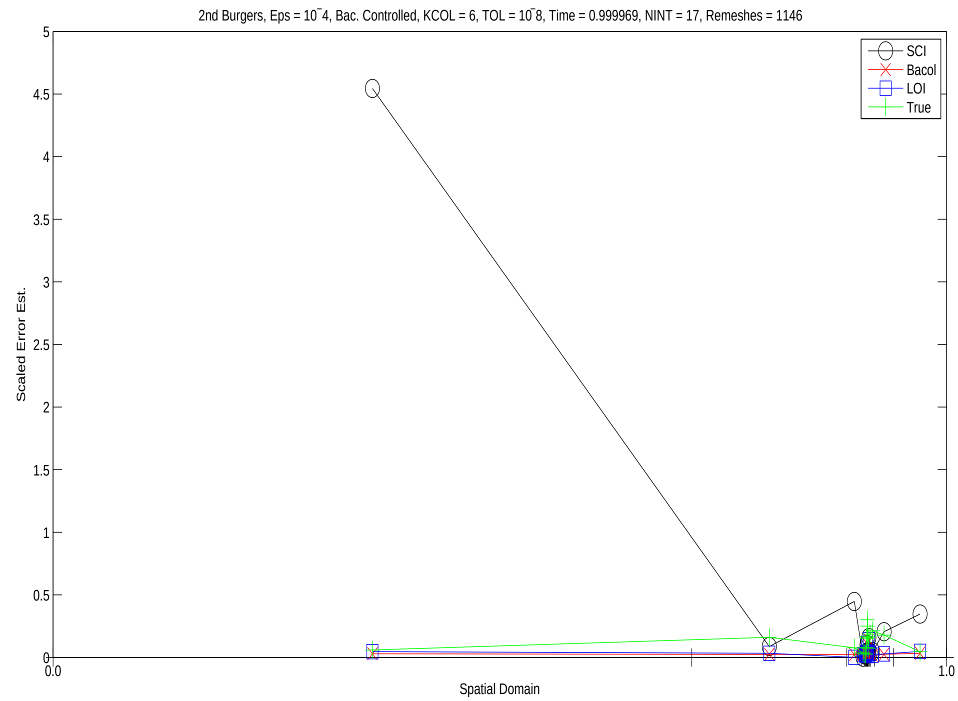


Figure 62: *When the BACOL estimates control the mesh, the SCI estimate is too large on the leftmost boundary subinterval and the error in the layer region is underestimated by all schemes. When the SCI estimates control the mesh, the time integration fails at the starting point.*



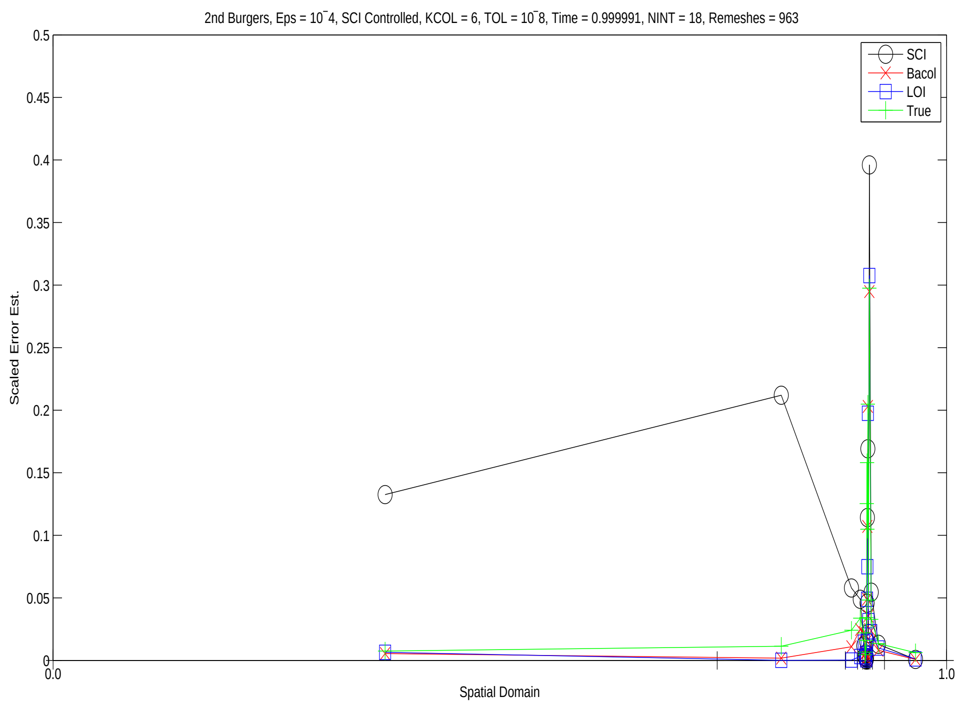
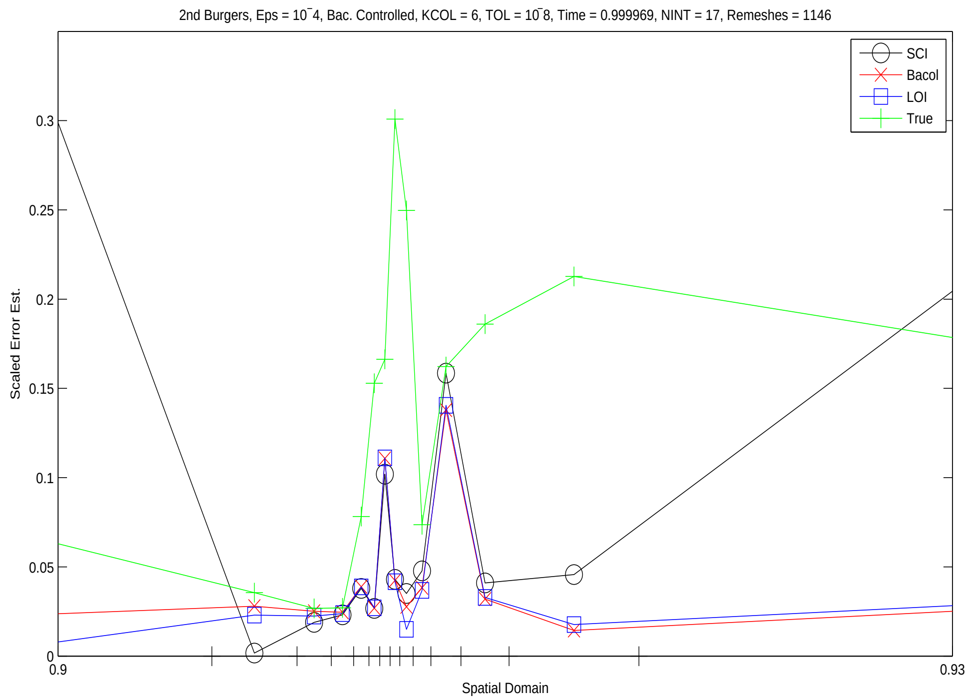


Figure 63: *When the BACOL estimates control the mesh, the SCI estimate is too large on the leftmost boundary subinterval. Even when the SCI estimates control the mesh, the two leftmost subintervals have SCI estimates that are too large.*



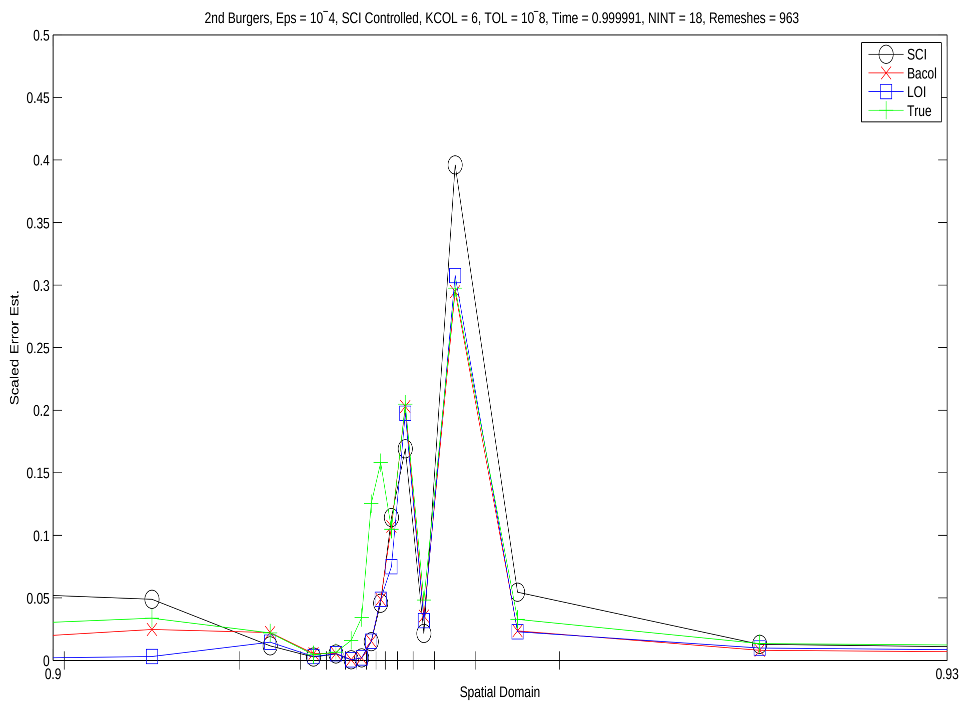
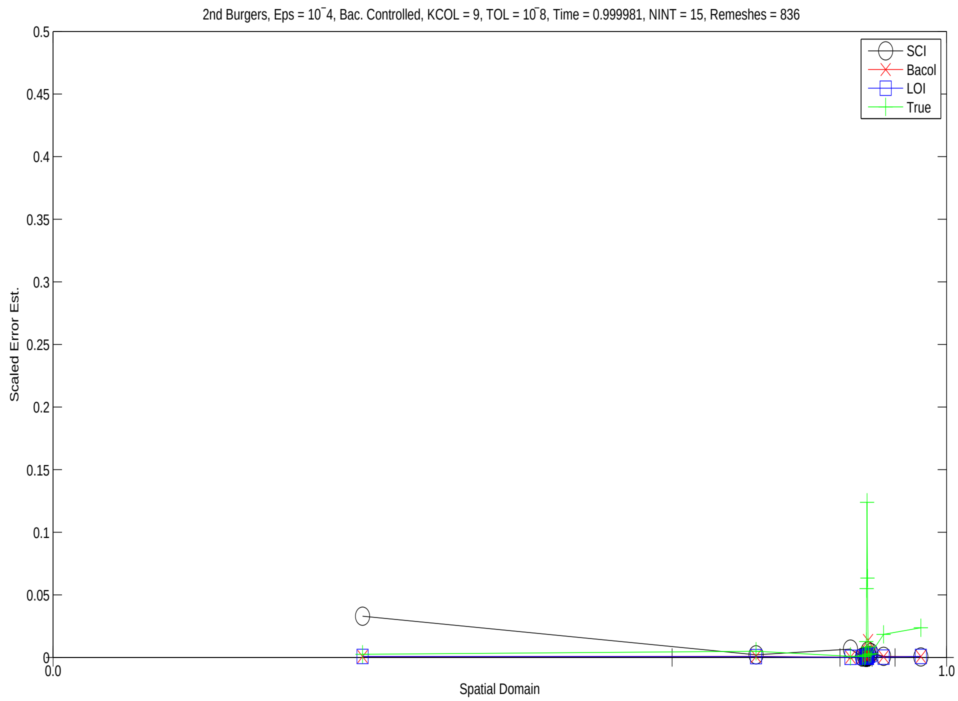


Figure 64: *Results for the layer region. When the BACOL estimates control the mesh, the error in the layer region is underestimated by all schemes. When the SCI estimates control the mesh, there is much better estimation of the error in the layer region.*



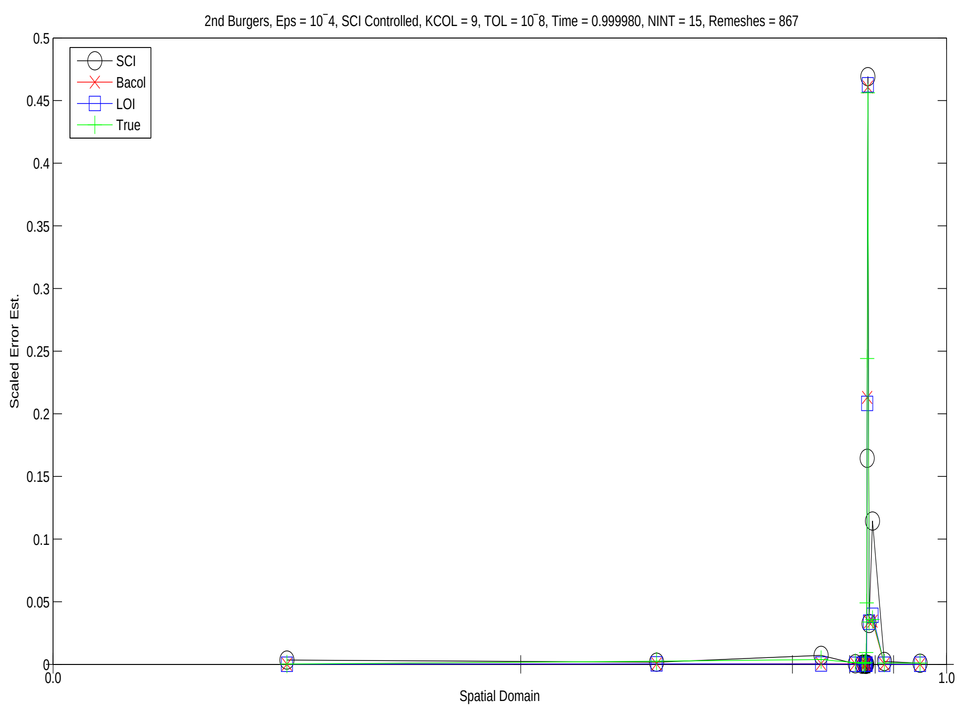
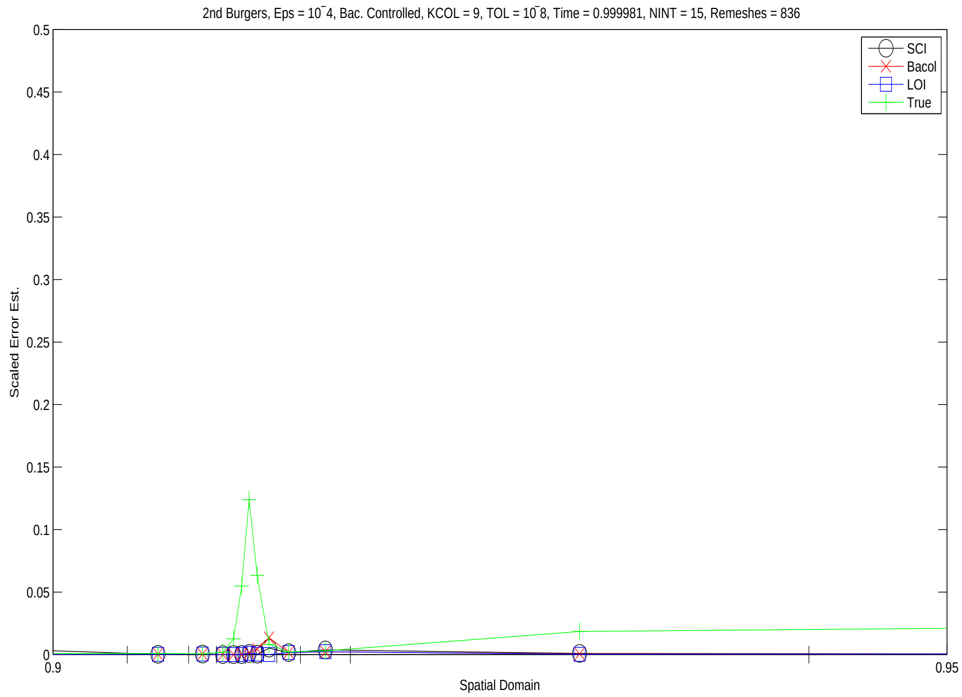


Figure 65: *When the BACOL estimates control the mesh, all schemes estimate the error away form the layer region reasonably well and the error in the layer region is underestimated by all schemes. When the SCI estimates control the mesh, there is much better estimation of the error across the spatial domain.*



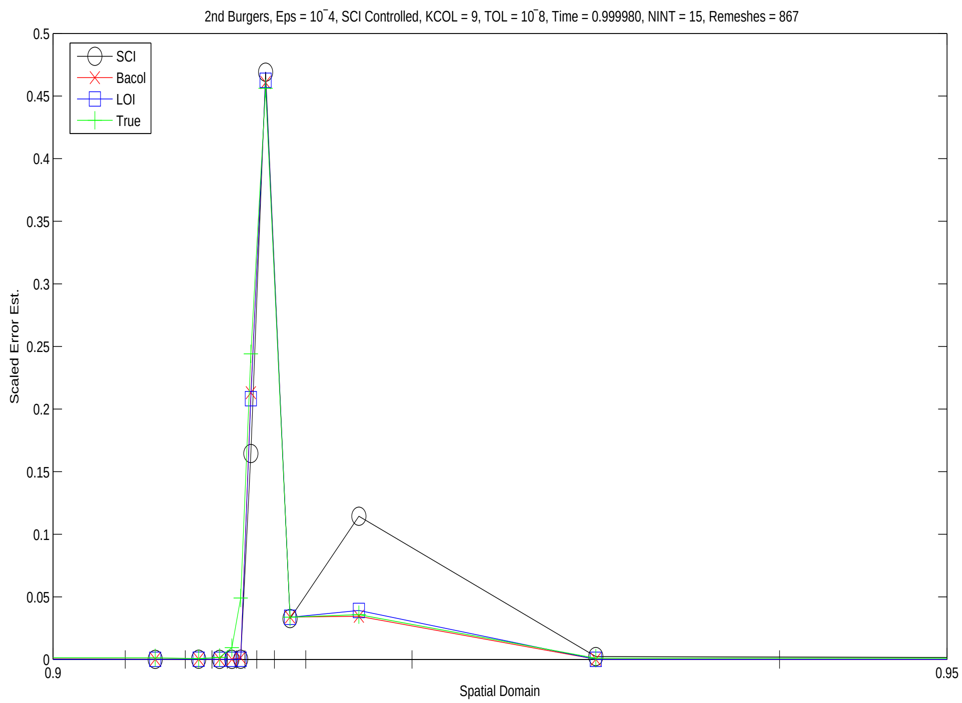


Figure 66: *Results for the layer region. When the BACOL estimates control the mesh, the error in the layer region is underestimated by all schemes. When the SCI estimates control the mesh, there is much better estimation of the error.*

9.3.9 Summary of Results for $\epsilon = 10^{-4}$

TOL \ KCOL		3	6	9
		NINT, Remeshes	NINT, Remeshes	NINT, Remeshes
10^{-4}	BAC	14, 634	14, 402	15, 300
	SCI	15, 856	14, 458	15, 336
10^{-6}	BAC	22, 882	15, 758	15, 562
	SCI	NA	15, 731	15, 618
10^{-8}	BAC	45, 936	17, 1146	15, 836
	SCI	NA	18, 963	15, 867

Table 8: *Both BACOL estimates controlling the mesh and SCI estimates controlling the mesh lead to comparable performance in terms of the number of subintervals in the final mesh and in terms of the total number of remeshings.*

9.4 Cahn-Allen Equation

9.4.1 Problem

$$u_t = \epsilon u_{xx} - u^3 + u, \qquad 0 < x < 1, \quad t > 0$$

using $\epsilon = 10^{-6}$.
Initial condition:

$$u(x,0) = 0.01\cos(10\pi x), \qquad 0 \le x \le 1$$

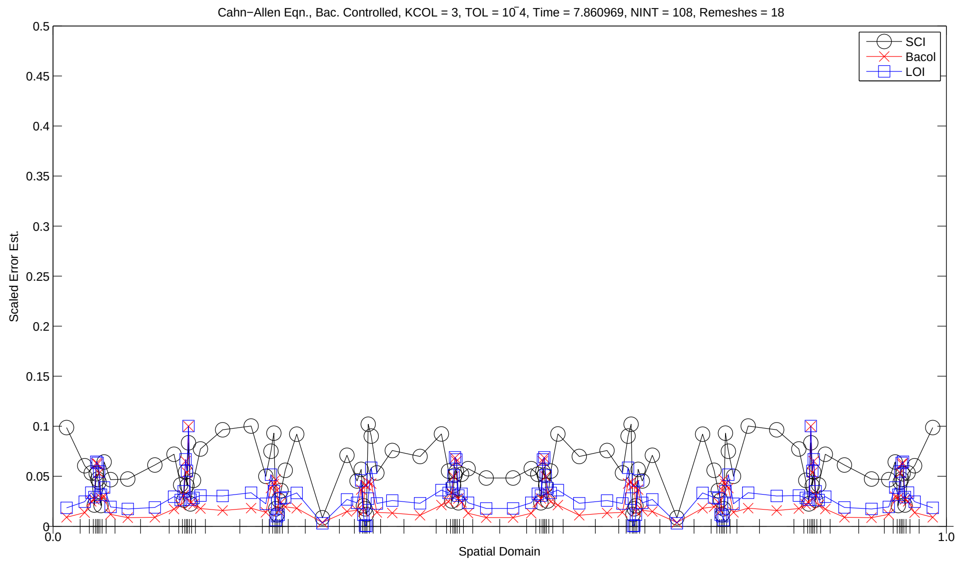
Boundary conditions:

$$u_x(0,t) = 0, \quad u_x(1,t) = 0, \quad t \ge 0.$$

See [38] for a graph of the solution.

The following three subsubsections provide results tolerances of 10^{-4}, 10^{-6}, and 10^{-8}. Within each subsubsection, we provide results for $k = KCOL = 3, 6$, and 9. The title of each graph indicates: the problem (Cahn-Allen Eqn.), which of the error estimation schemes controls the mesh refinement (Bac. Controlled or SCI Controlled), the value of KCOL, the tolerance TOL, the final Time (close to 7.9), the number of subintervals in the final mesh, NINT, and the total number of spatial remeshings that took place over the entire computation, Remeshes.

9.4.2 Results for Tolerance $= 10^{-4}$



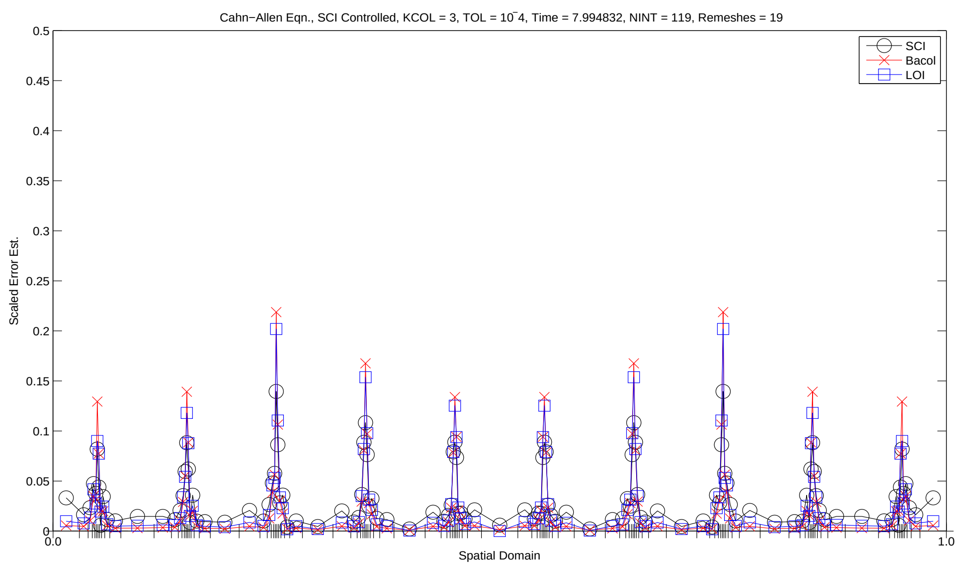
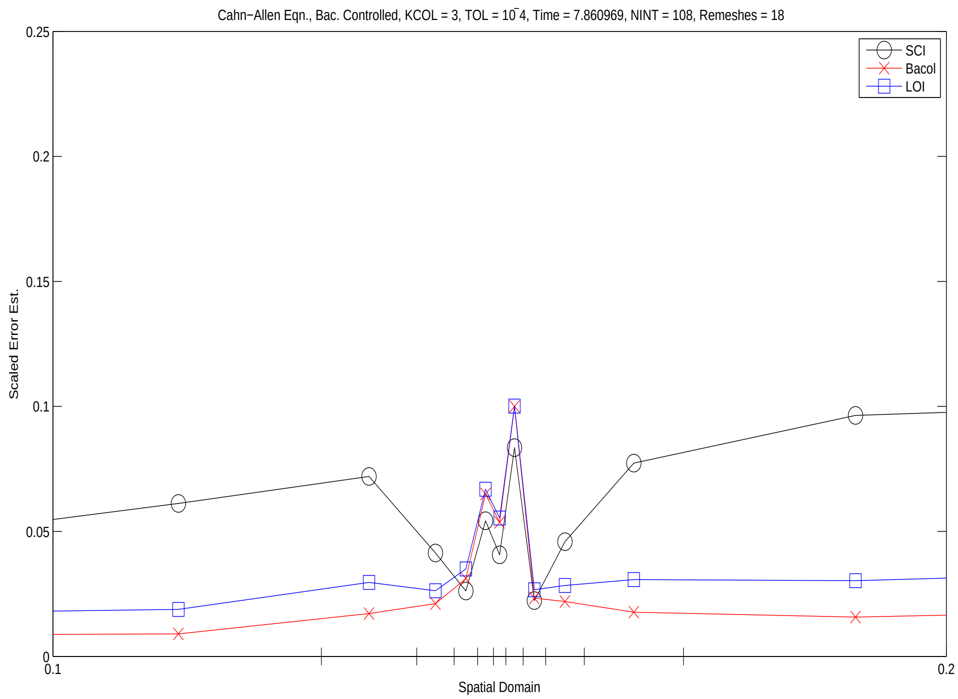


Figure 67: *When the BACOL estimates control the mesh, the SCI estimates are larger than the BACOL or LOI estimates. When the SCI estimates control the mesh, all schemes are in good agreement.*



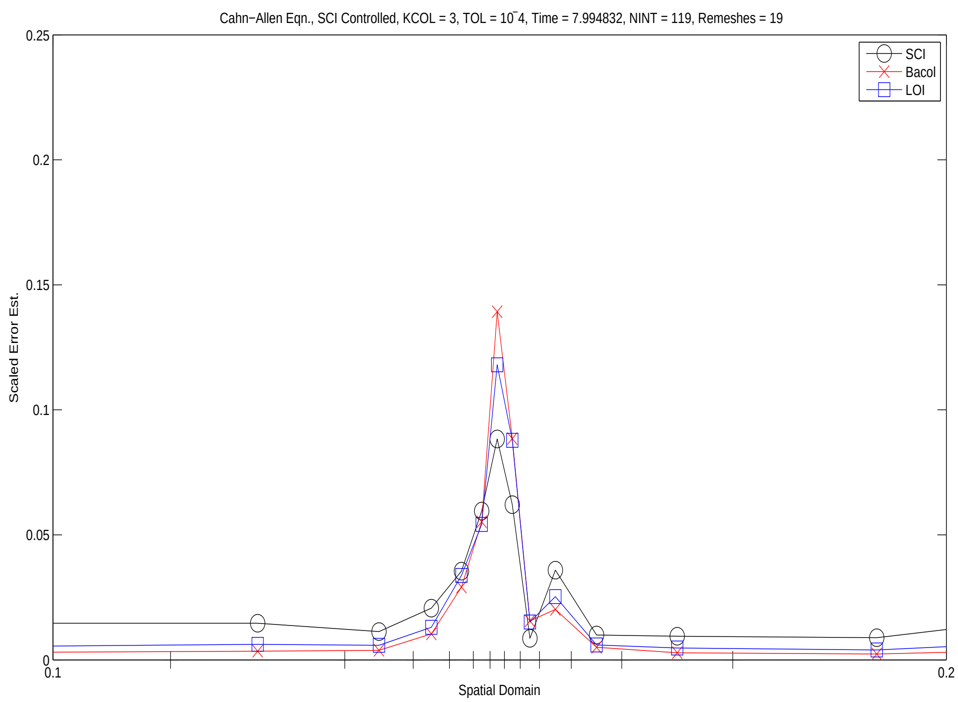
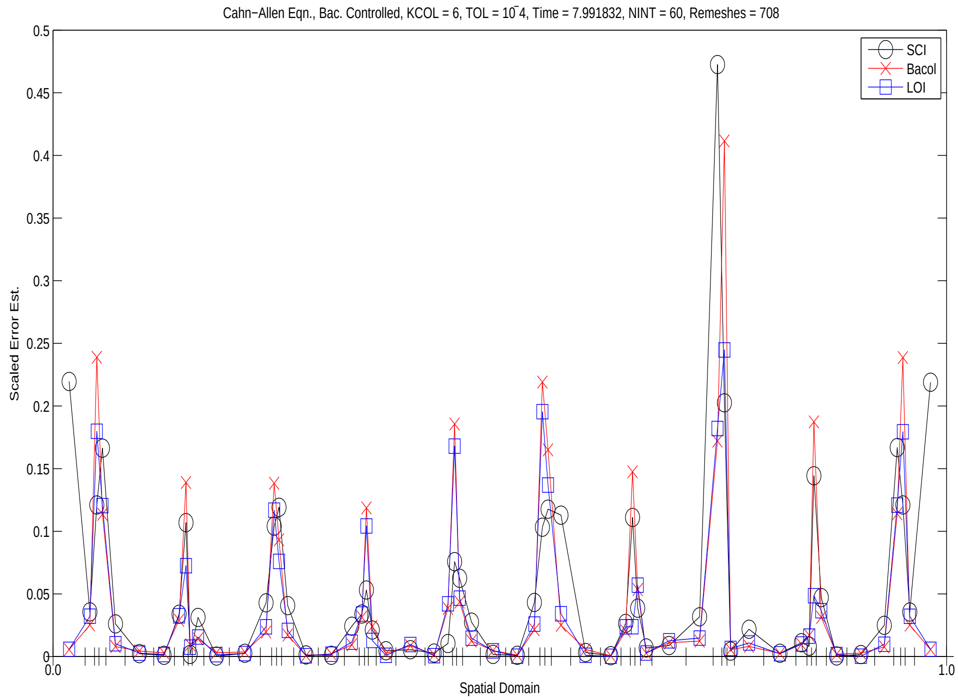


Figure 68: *Results for one of the layer regions. When the BACOL estimates control the mesh, the SCI estimates are larger than the other schemes which are in good agreement. When the SCI estimates control the mesh, all schemes are in good agreement.*



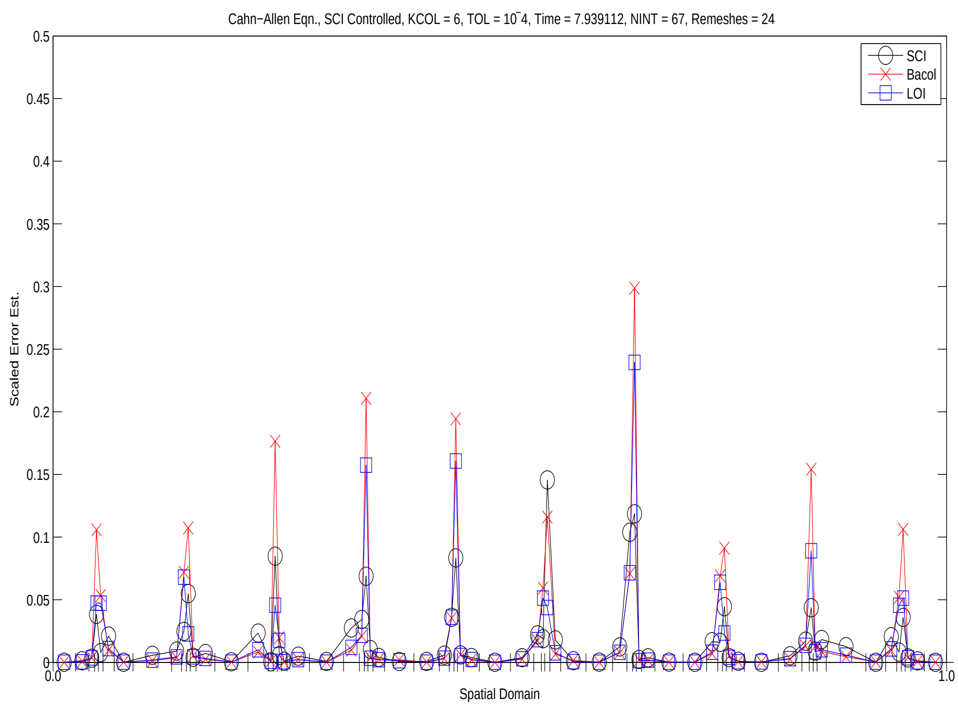
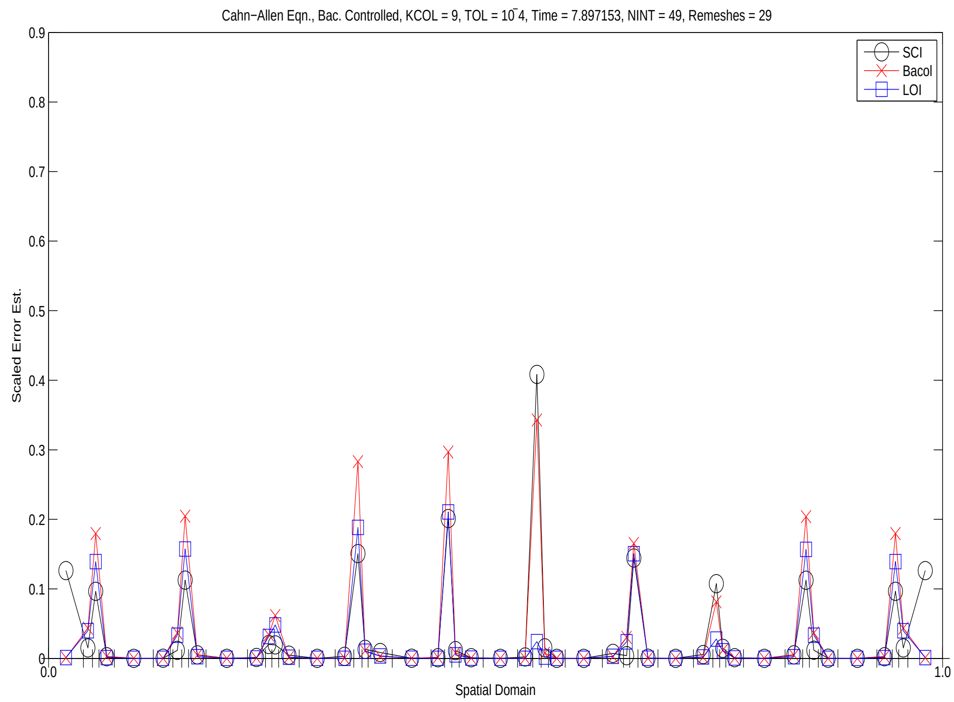


Figure 69: *When the BACOL estimates control the mesh, the SCI estimates are larger on the boundary subintervals than the BACOL or LOI estimates; otherwise there is reasonable agreement. When the SCI estimates control the mesh, the BACOL estimates are somewaht larger than the other estimates but there is fairly good agreement overall.*



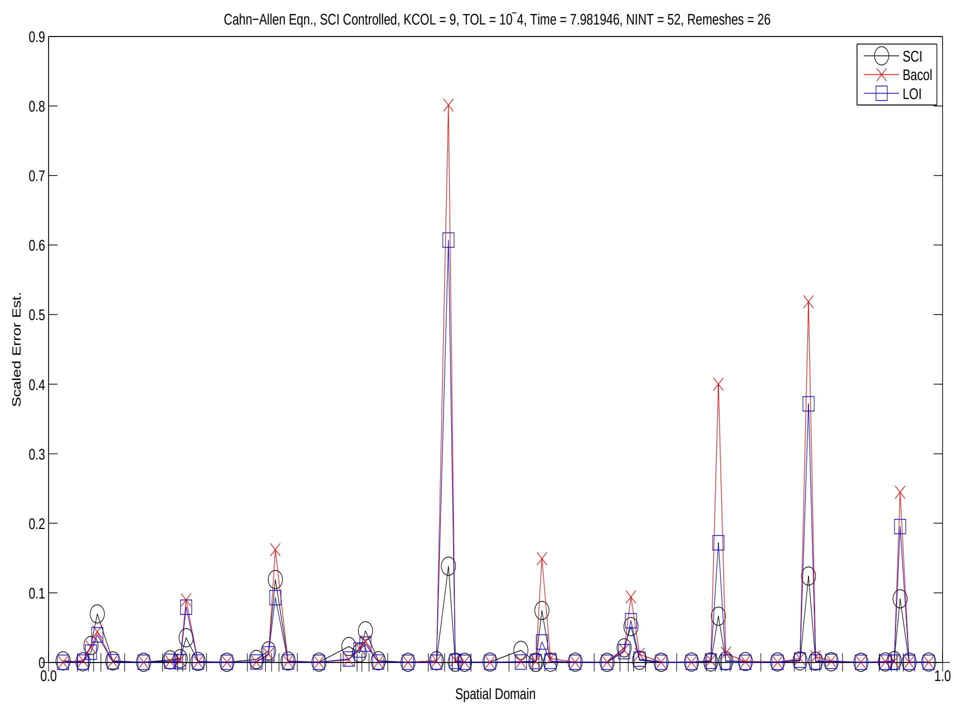
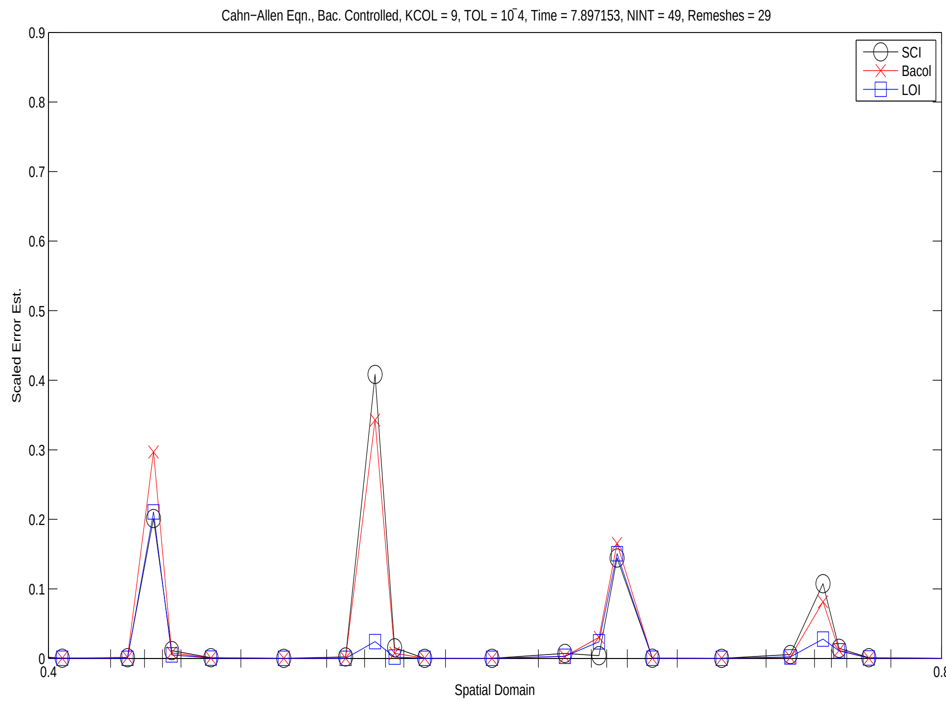


Figure 70: *Reasonably good agreement among all schemes over the entire spatial domain.*



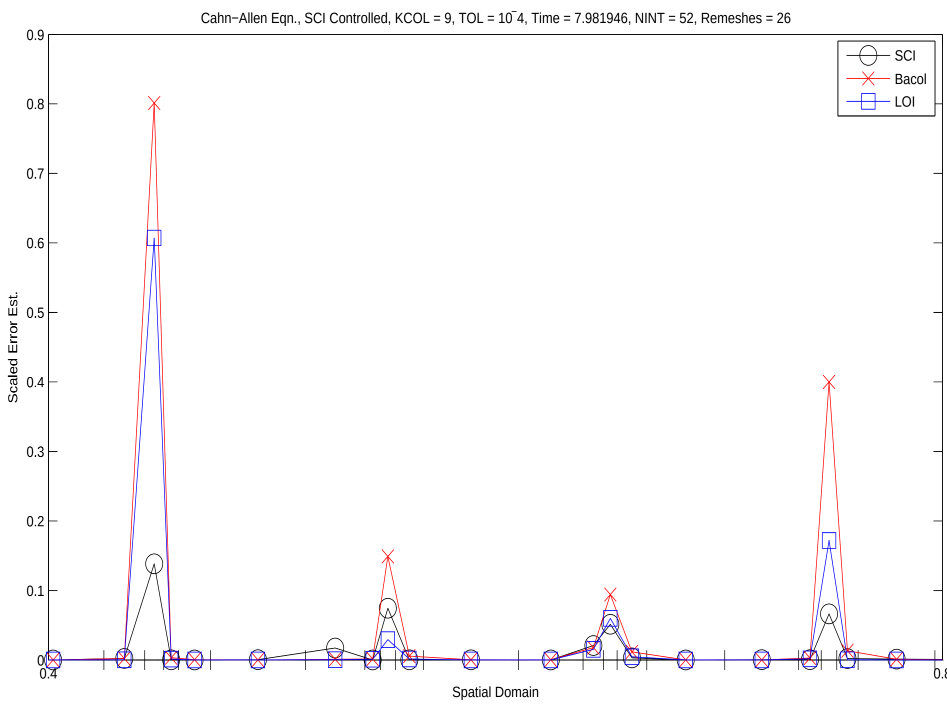
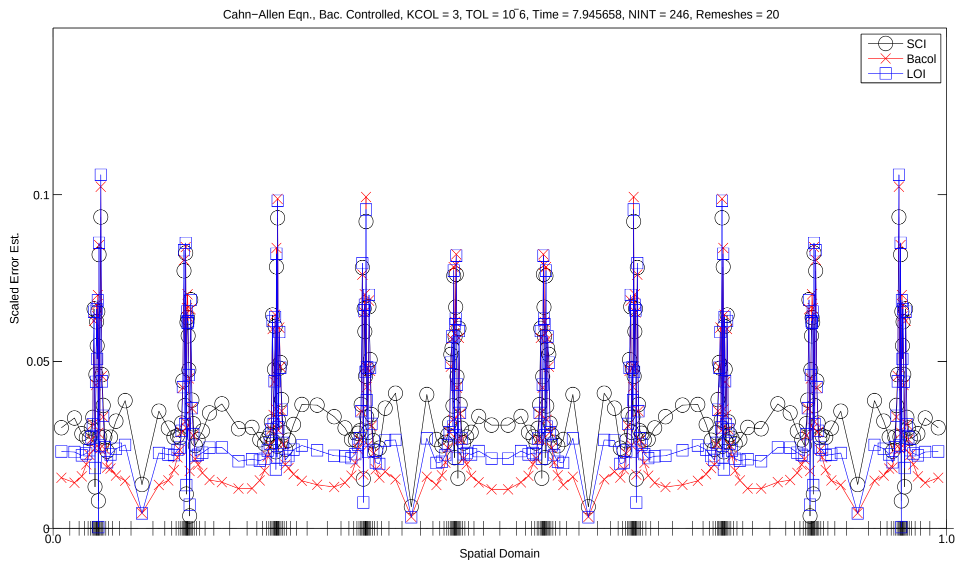


Figure 71: *Results only for a portion of the spatial domain. Reasonably good agreement among all schemes.*

9.4.3 Results for Tolerance $= 10^{-6}$



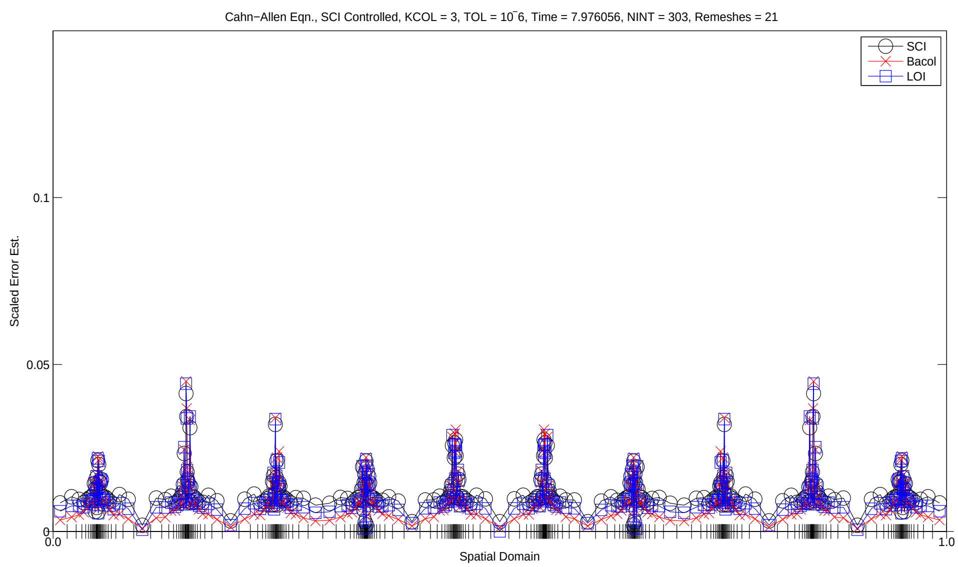
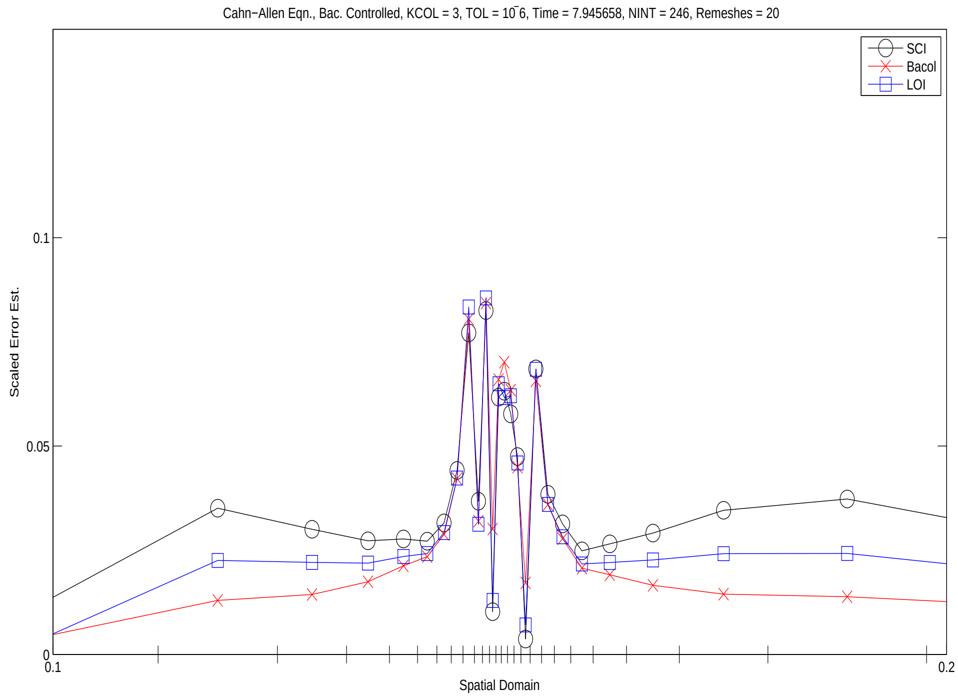


Figure 72: *Reasonably good agreement among all schemes over the entire spatial domain.*



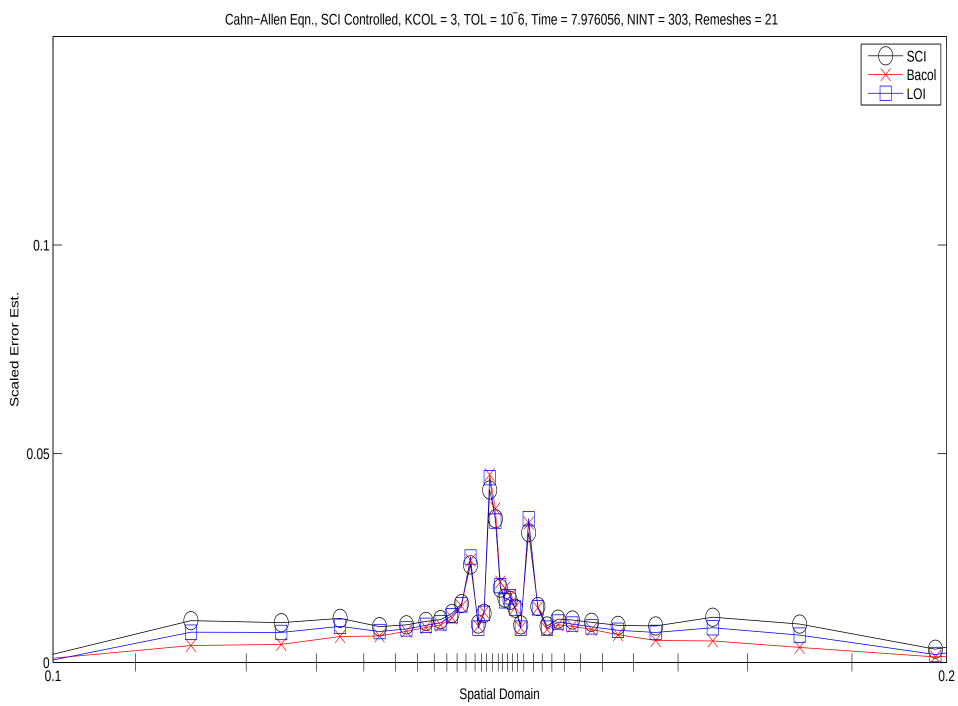


Figure 73: *Results for only one layer region. Good agreement among all schemes.*

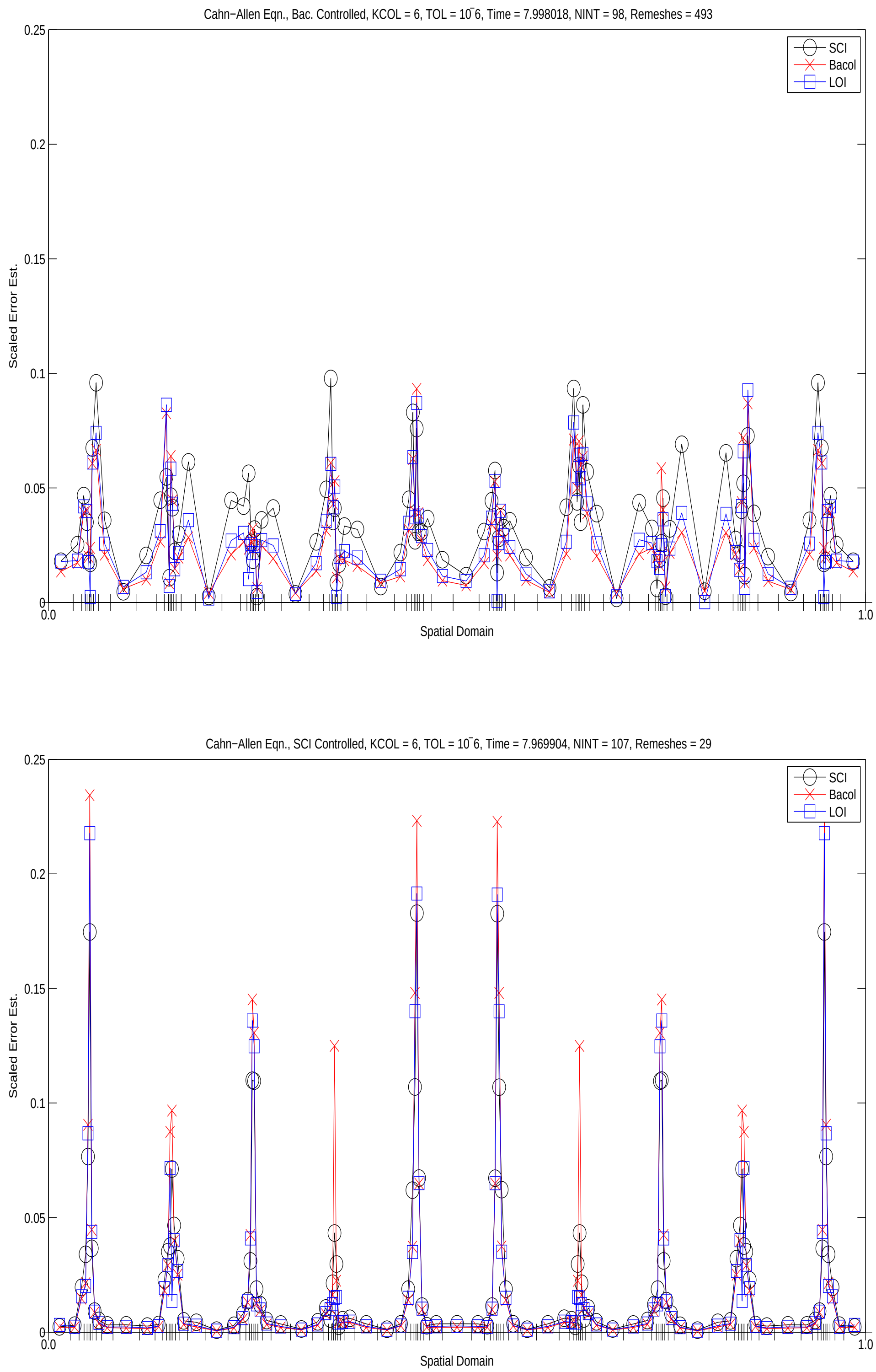


Figure 74: *Reasonably good agreement among all schemes over the entire spatial domain.*

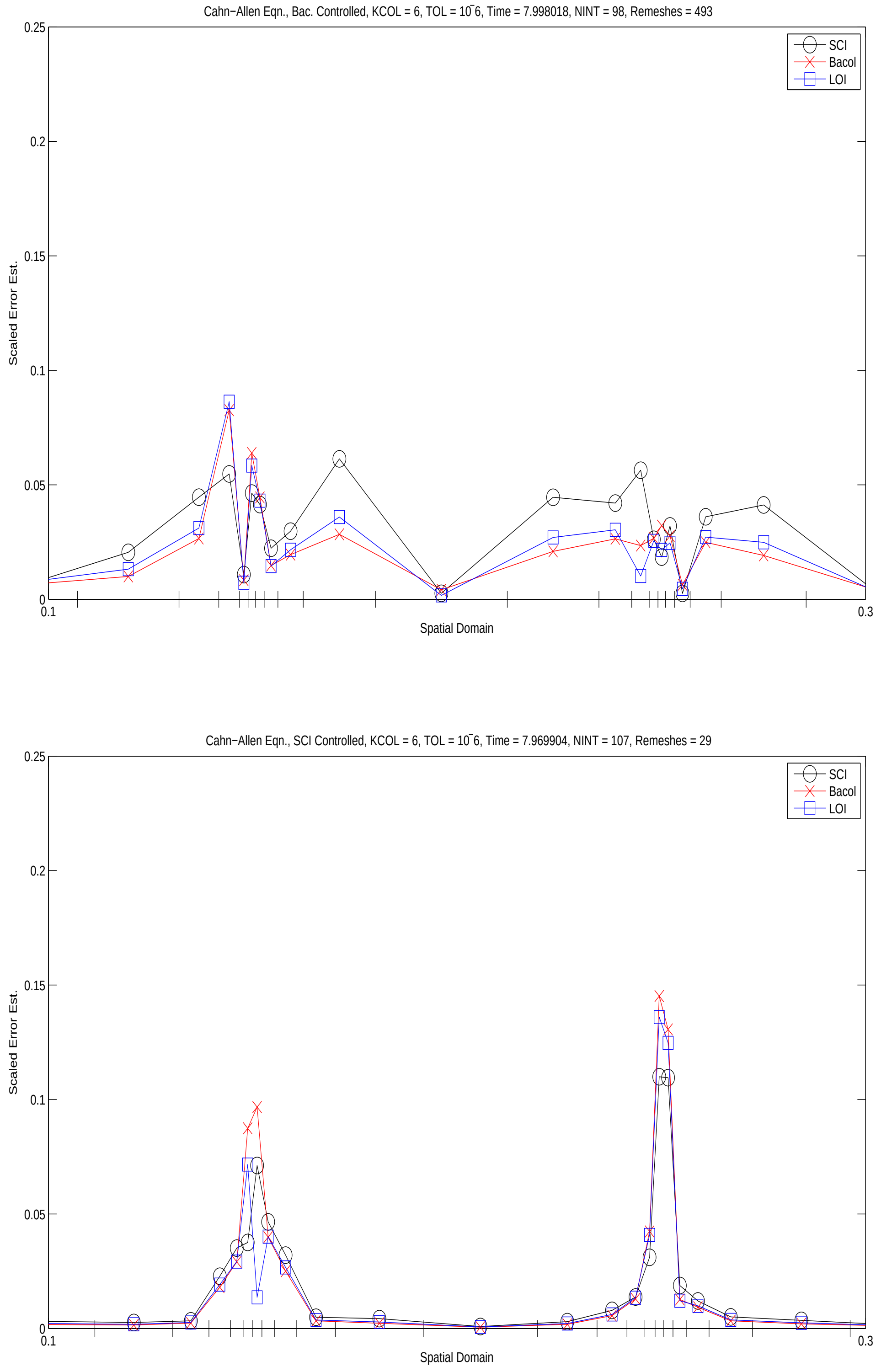
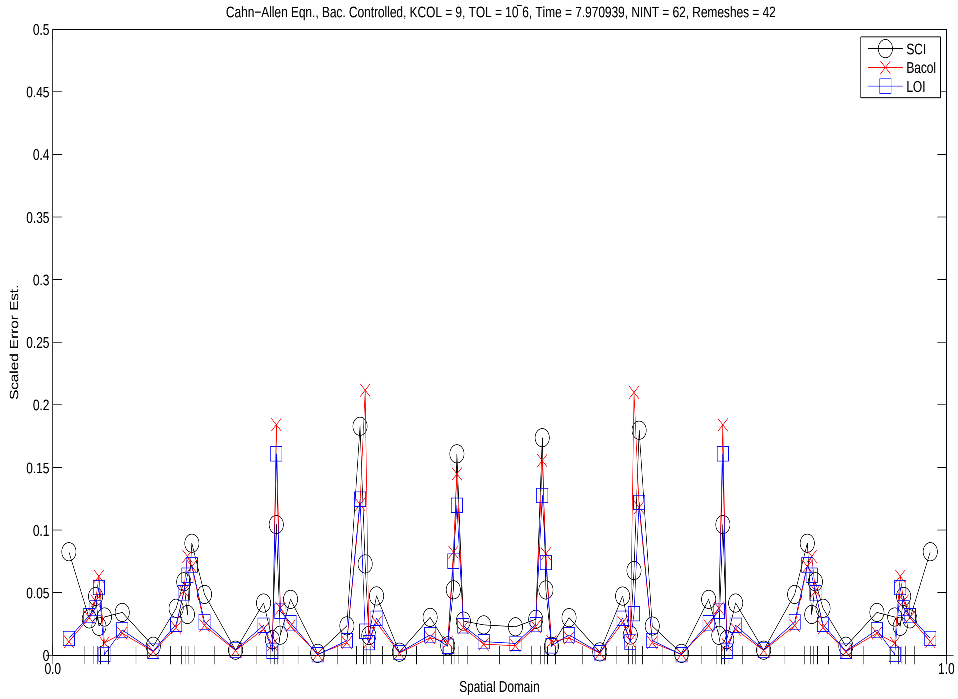


Figure 75: *Results for only two layer regions. Good agreement among all schemes.*



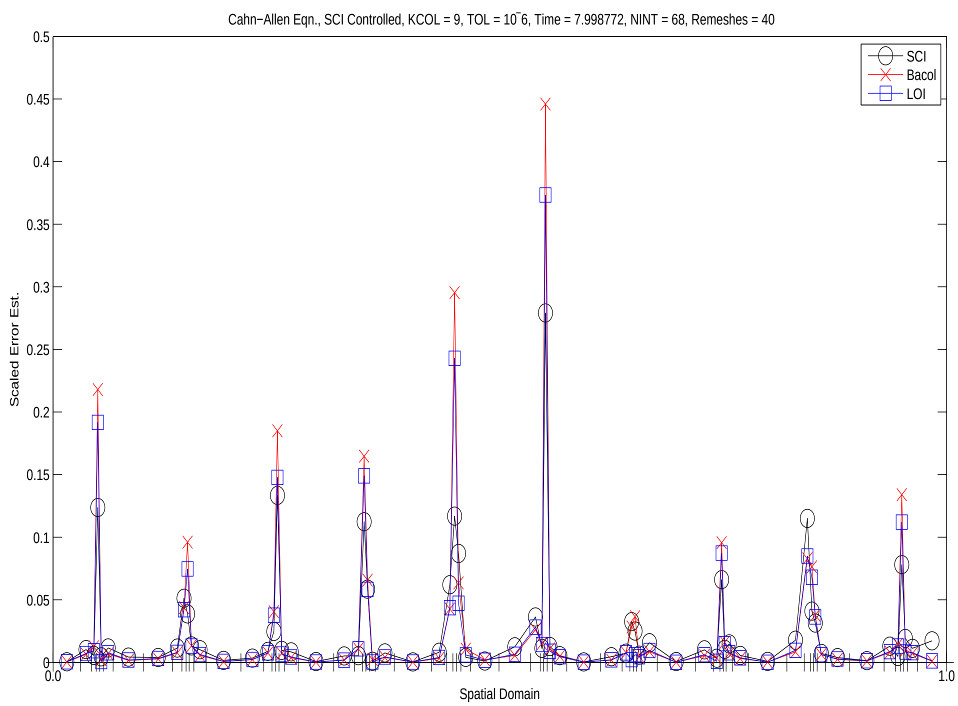


Figure 76: *Reasonably good agreement among all schemes over the entire spatial domain.*

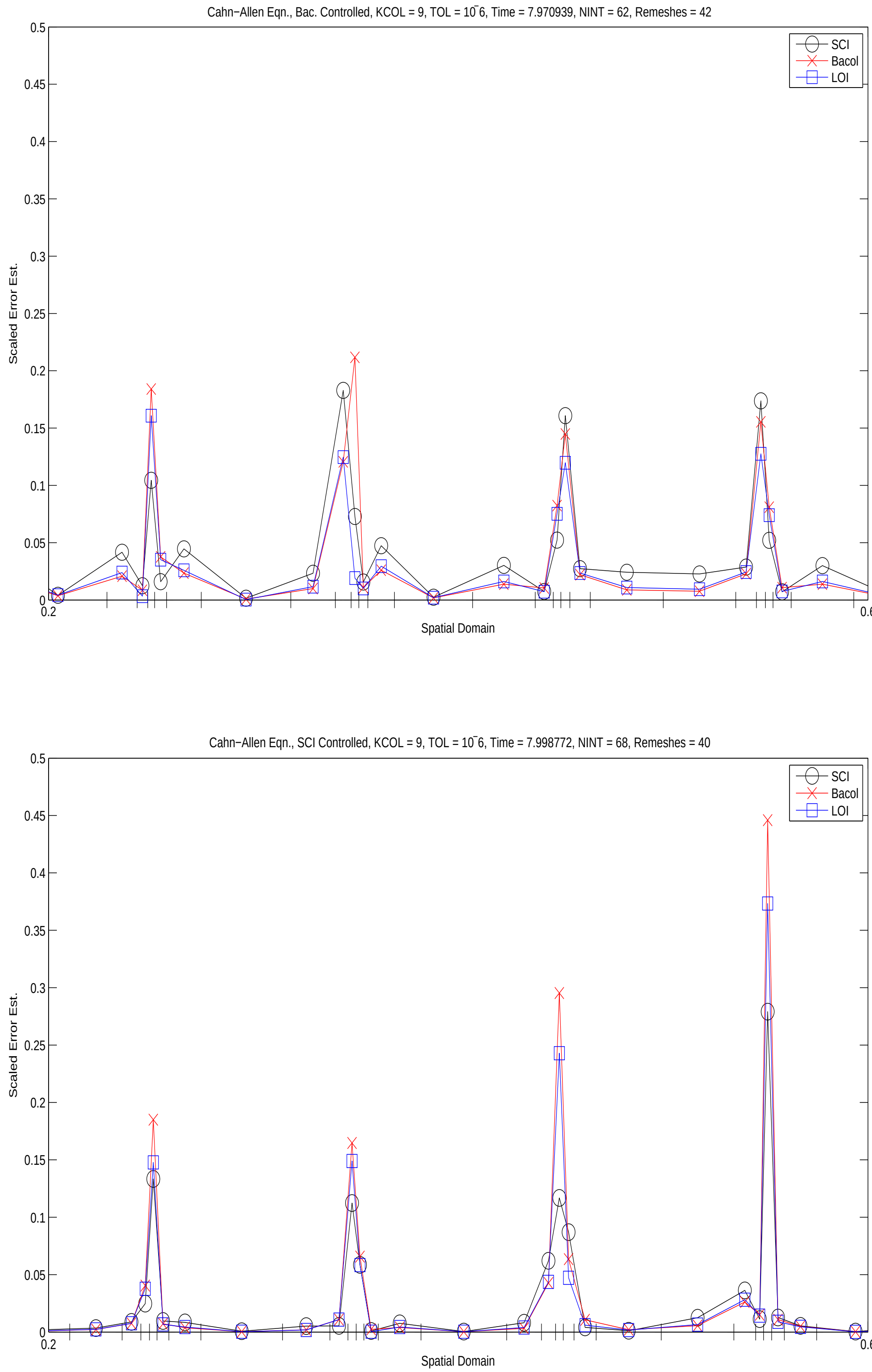


Figure 77: *Results only for a portion of the spatial domain. Reasonably good agreement among all schemes.*

9.4.4 Results for Tolerance = 10^{-8}

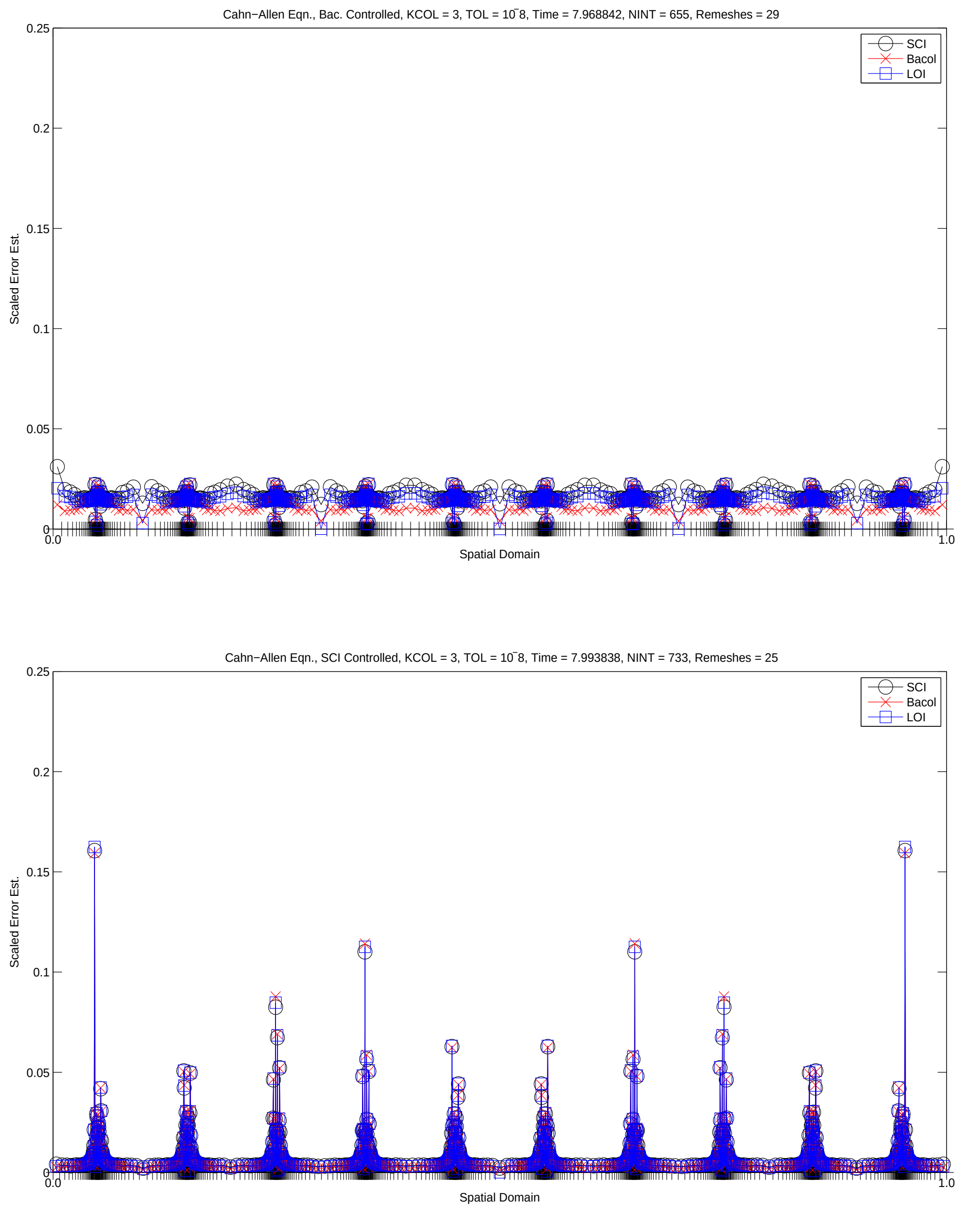
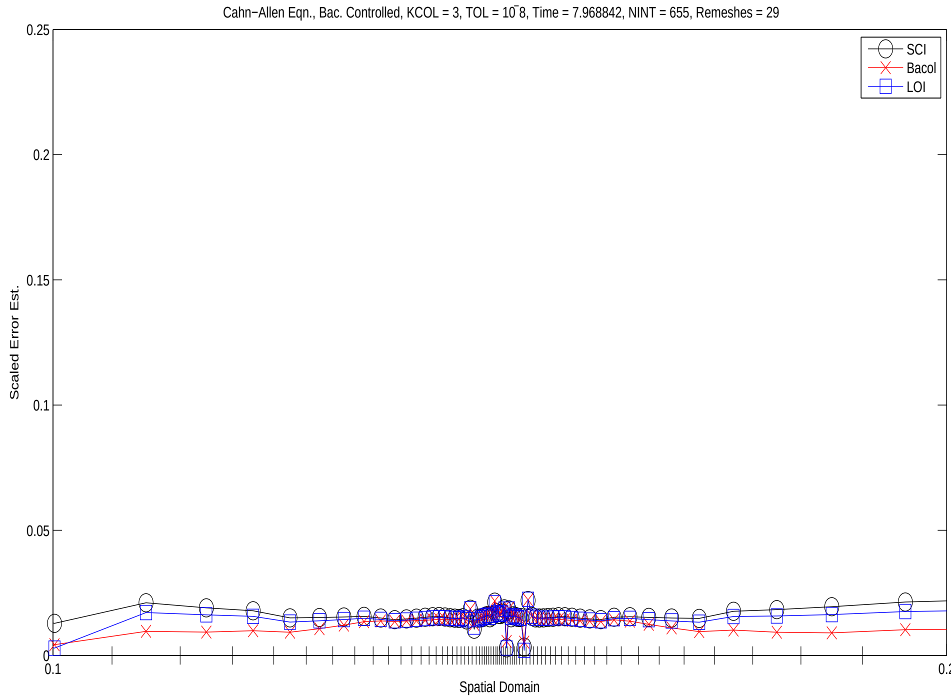


Figure 78: *Reasonably good agreement among all schemes over the entire spatial domain.*



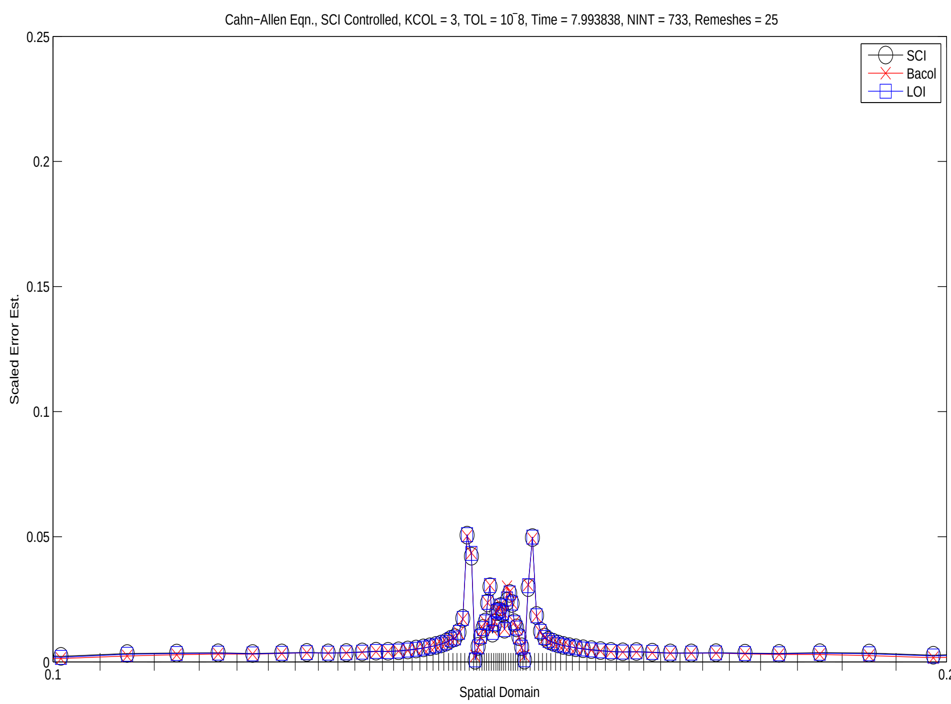
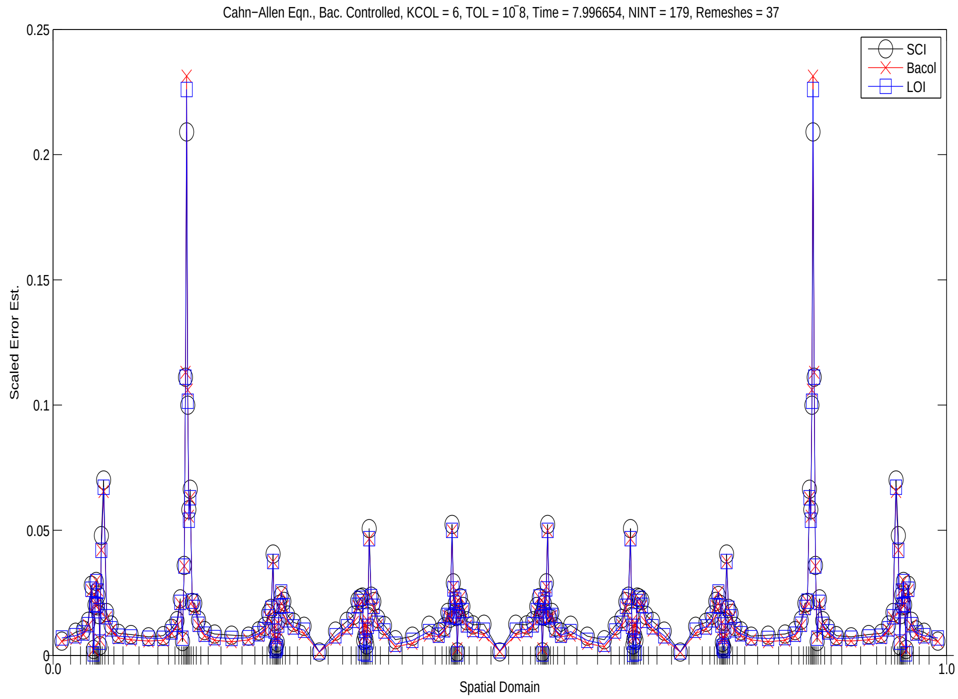


Figure 79: *Results for only one layer region. Good agreement among all schemes.*



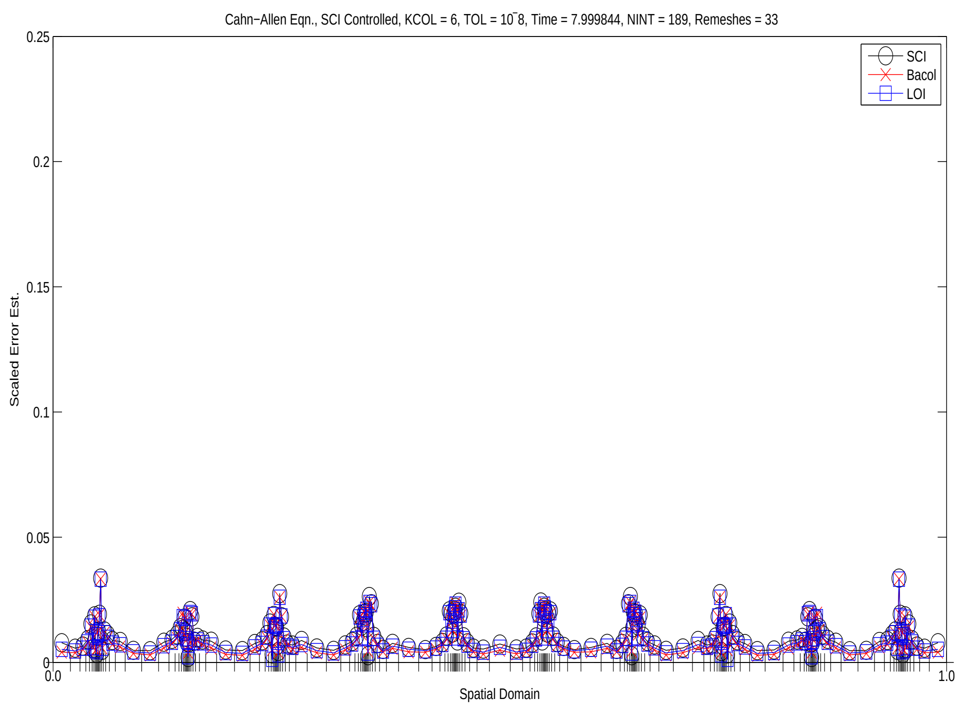
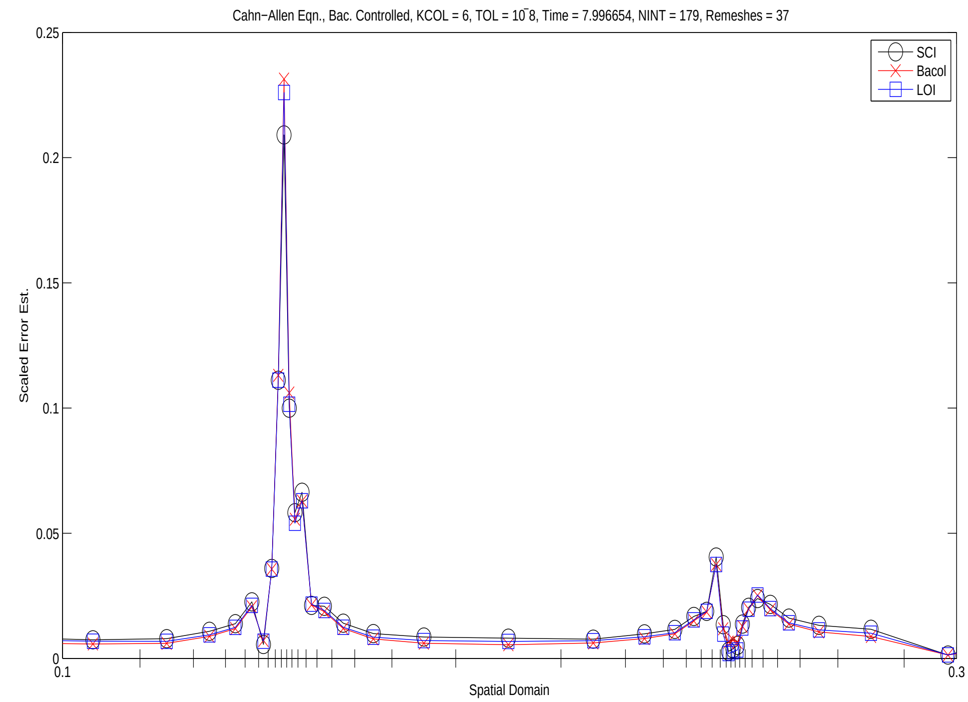


Figure 80: *Excellent agreement among all schemes over the entire spatial domain.*



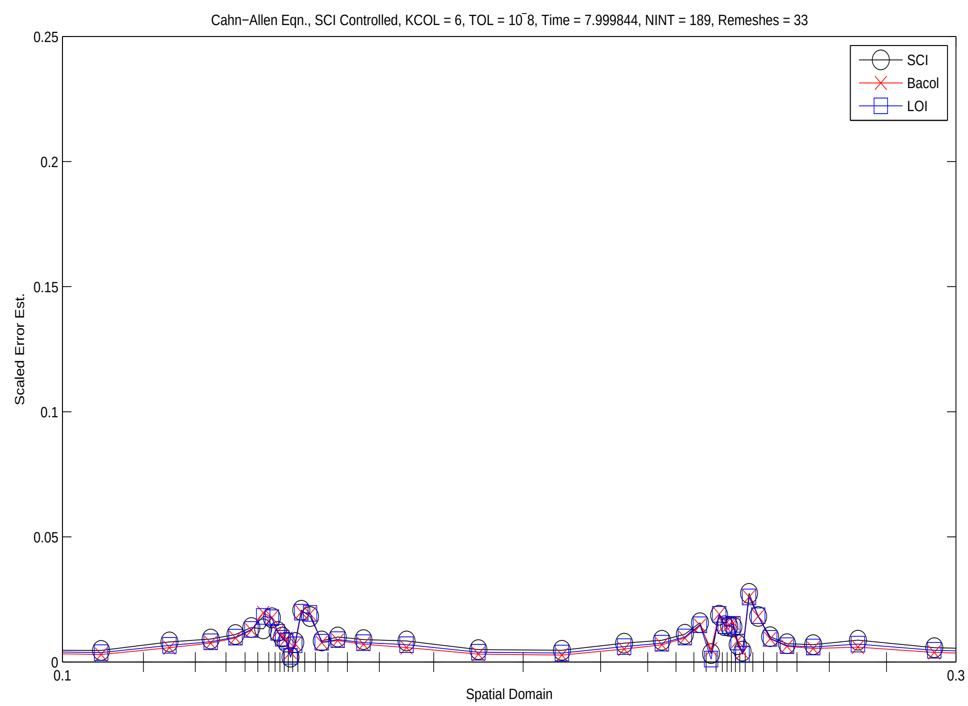


Figure 81: *Results for only two layer regions. Good agreement among all schemes.*

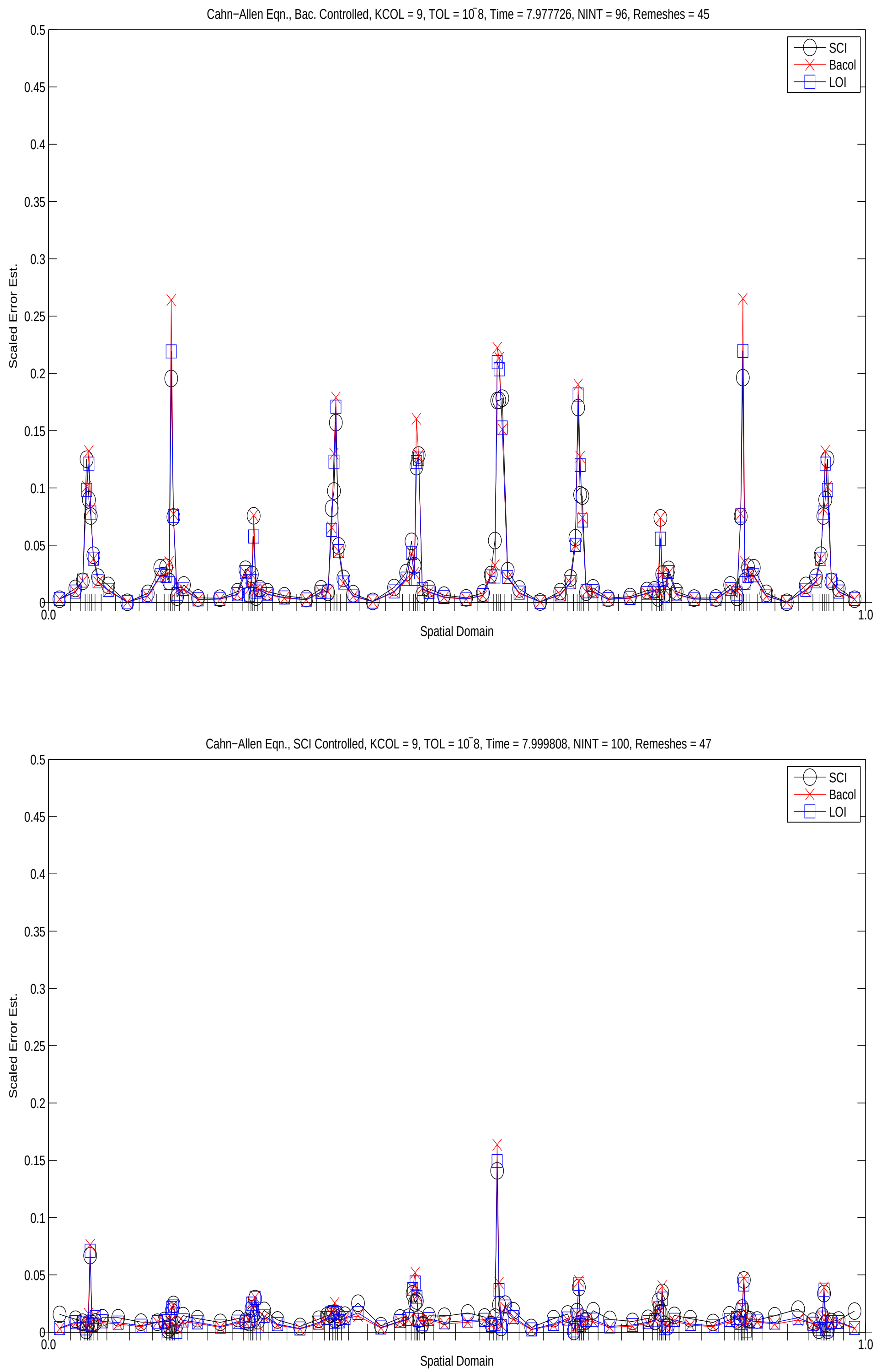


Figure 82: *Excellent agreement among all schemes over the entire spatial domain.*

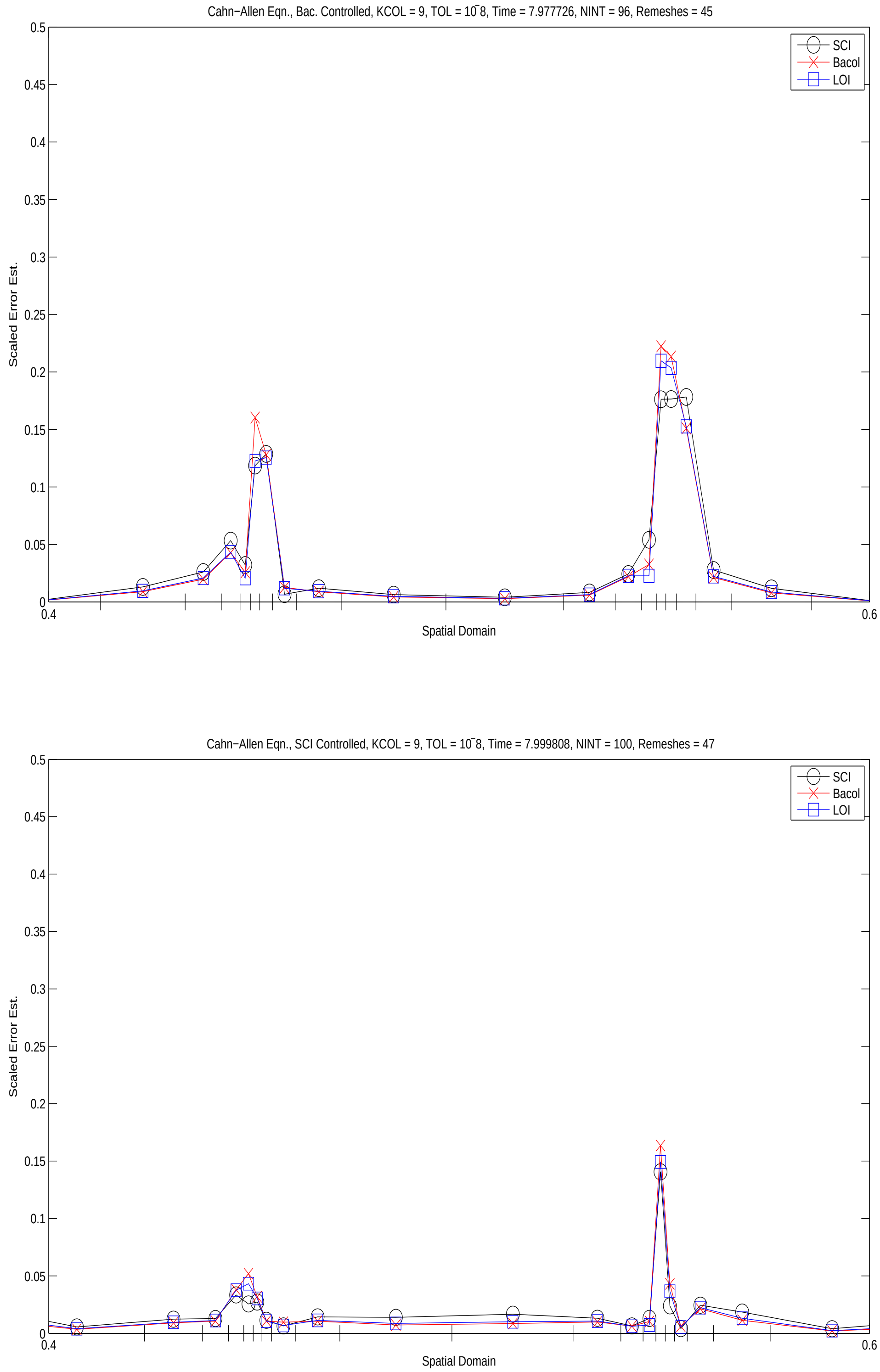


Figure 83: *Results for only two layer regions. Excellent agreement among all schemes.*

9.4.5 Summary of Results

TOL \ KCOL		3	6	9
		NINT, Remeshes	NINT, Remeshes	NINT, Remeshes
10^{-4}	BAC	108, 18	60, 708	49, 29
	SCI	119, 19	67, 24	52, 26
10^{-6}	BAC	246, 20	98, 493	62, 42
	SCI	303, 21	107, 29	68, 40
10^{-8}	BAC	655, 29	179, 37	96, 45
	SCI	733, 25	189, 33	100, 47

Table 9: *Generally comparable performance for BACOL estimates controlling the mesh vs. SCI estimates controlling the mesh except for the $KCOL = 6$, $TOL = 10^{-4}, 10^{-6}$ cases; in these cases the number of remeshings required by the BACOL estimate controlled computation is 20 to 30 times greater.*

9.5 Nonlinear Equation with Steady-State Solution

9.5.1 Problem

$$u_t = (\exp^{5u} u_x)_x, \qquad 0 < x < 1, \quad t > 0$$

Initial condition:

$$u(x, 0) = 2x, \quad 0 \leq x \leq 1$$

Boundary conditions:

$$u(0, t) = 0, \quad u(1, t) = 2, \quad t \geq 0$$

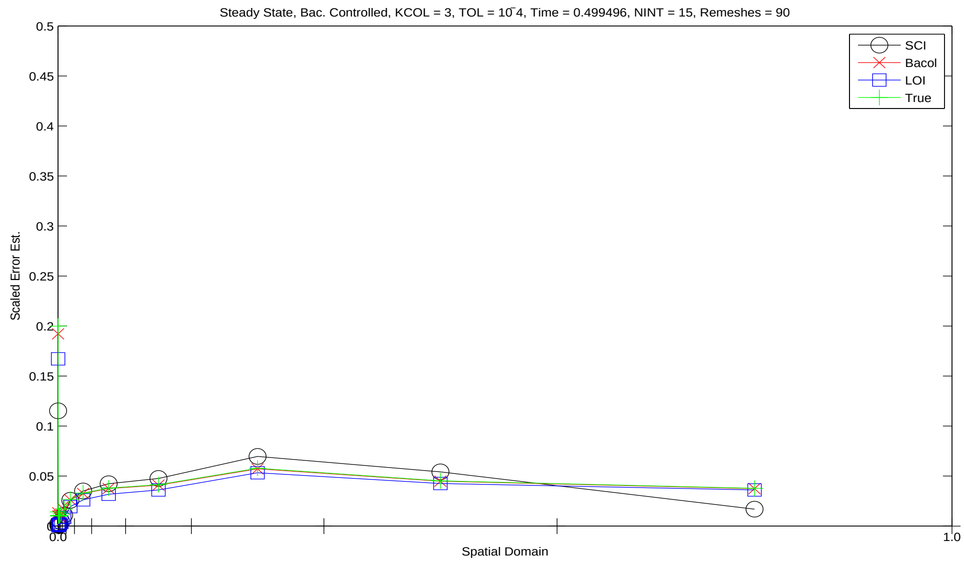
Steady-state solution:

$$u(x, t \to \infty) = \frac{1}{5} \log \left[1 + (\exp^{10} - 1)x\right]$$

See [38] for a graph of the solution.

The following three subsubsections provide results tolerances of 10^{-4}, 10^{-6}, and 10^{-8}. Within each subsubsection, we provide results for $k = KCOL = 3, 6$, and 9. The title of each graph indicates: the problem (Steady State), which of the error estimation schemes controls the mesh refinement (Bac. Controlled or SCI Controlled), the value of KCOL, the tolerance TOL, the final Time (close to 0.5), the number of subintervals in the final mesh, NINT, and the total number of spatial remeshings that took place over the entire computation, Remeshes.

9.5.2 Results for Tolerance = 10^{-4}



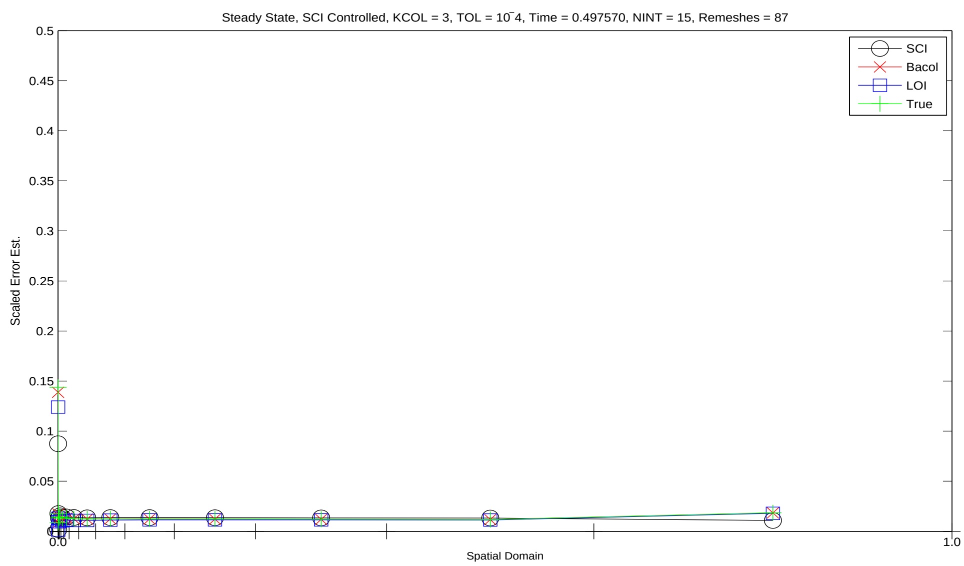
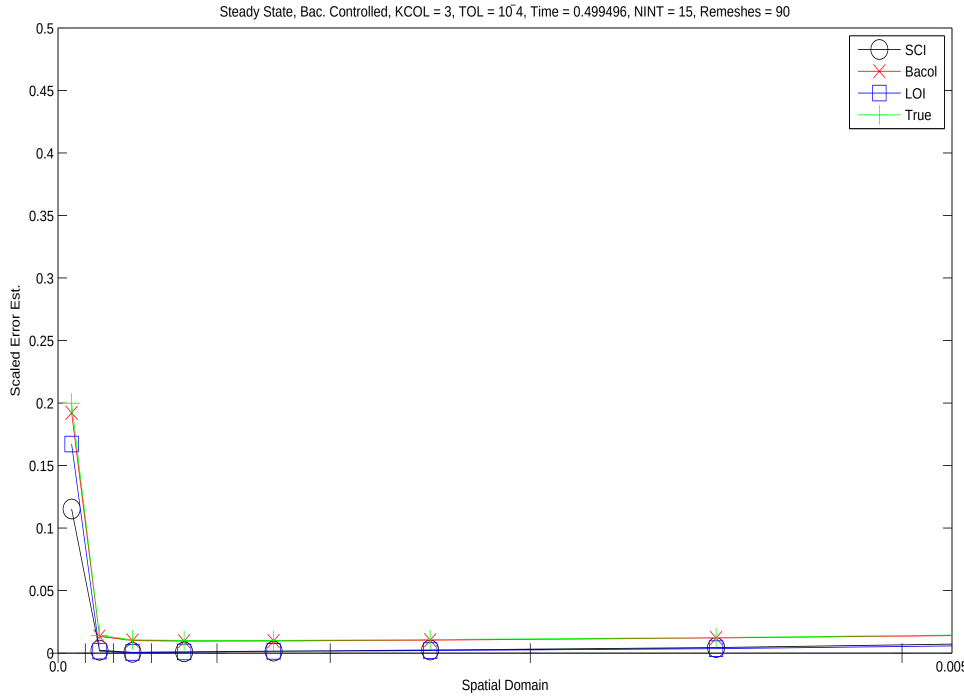


Figure 84: *Good agreement, with either estimate in control of the mesh, between the estimates and the true error throughout the spatial domain.*



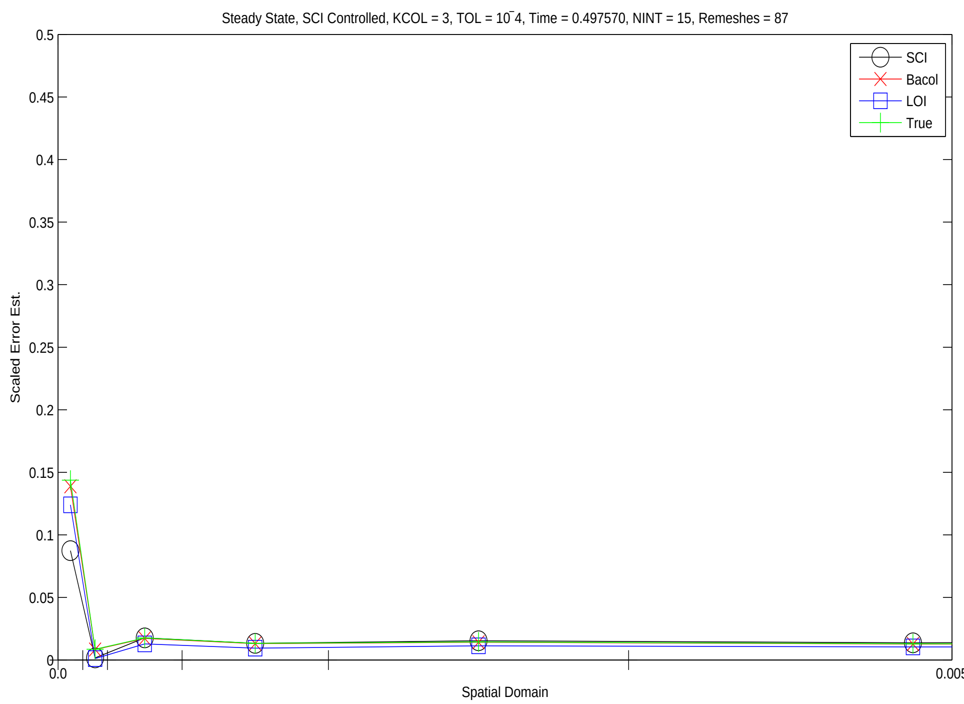
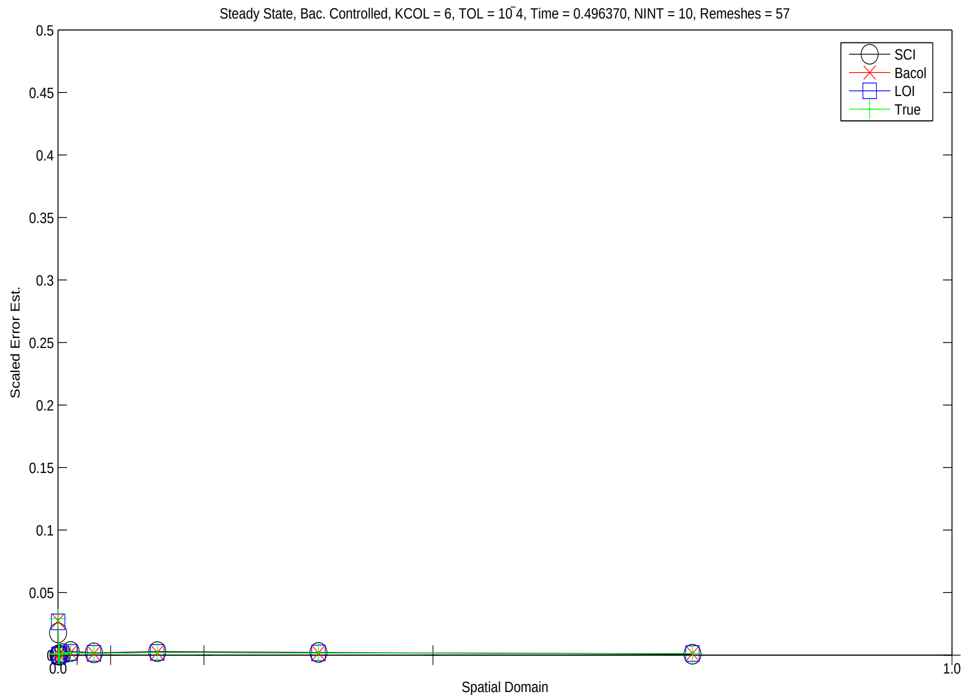


Figure 85: *Results only for the extreme left end of the spatial domain. Good agreement, with either estimate in control of the mesh, between the estimates and the true error.*



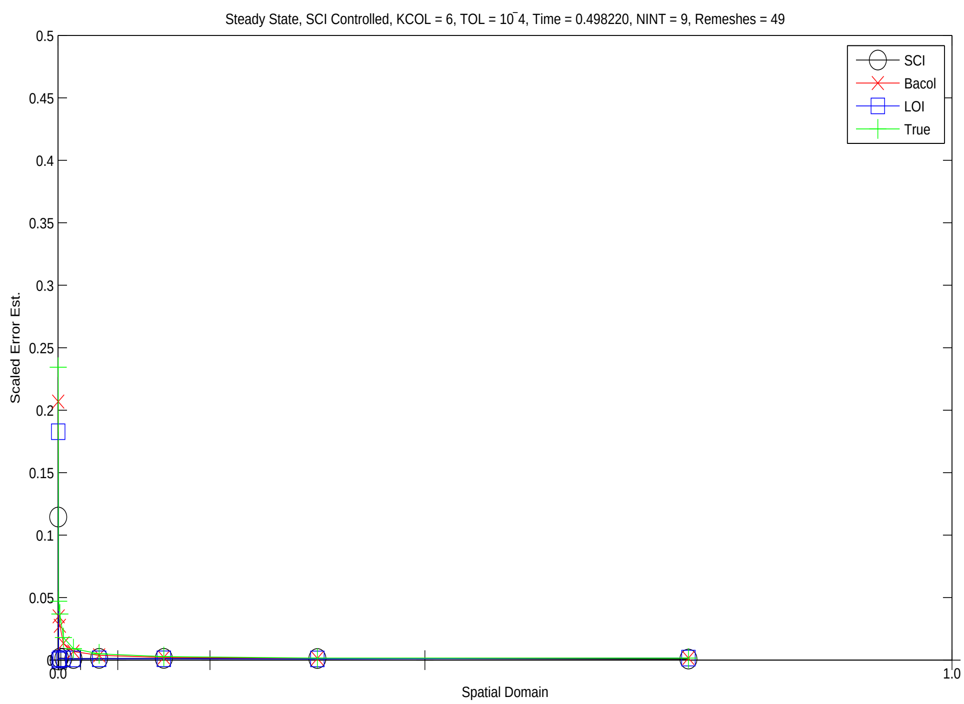
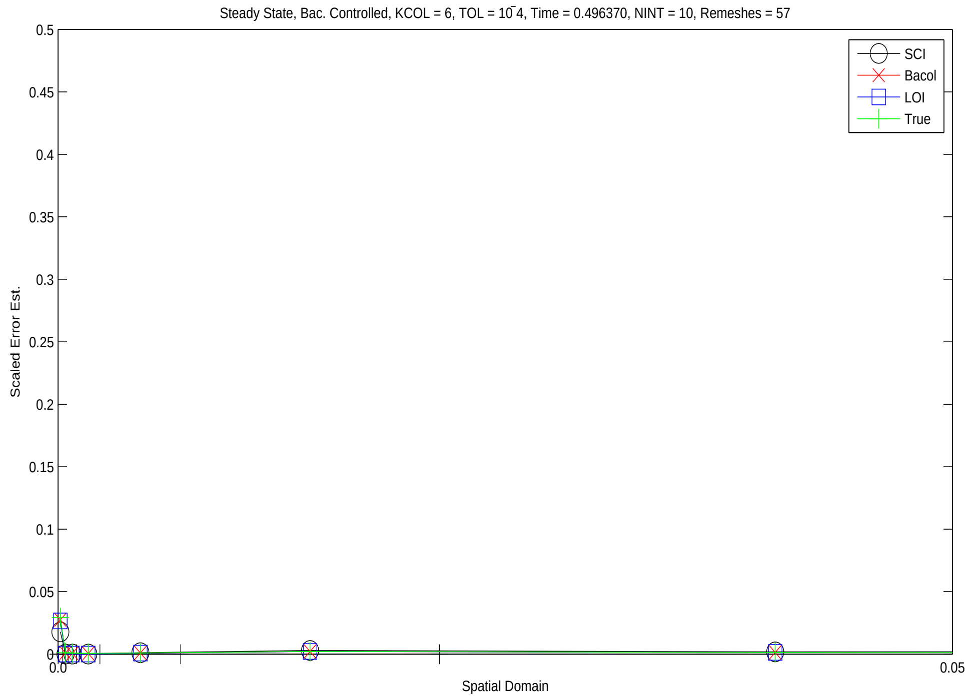


Figure 86: *Good agreement, with either estimate in control of the mesh, between the estimates and the true error throughout the spatial domain.*



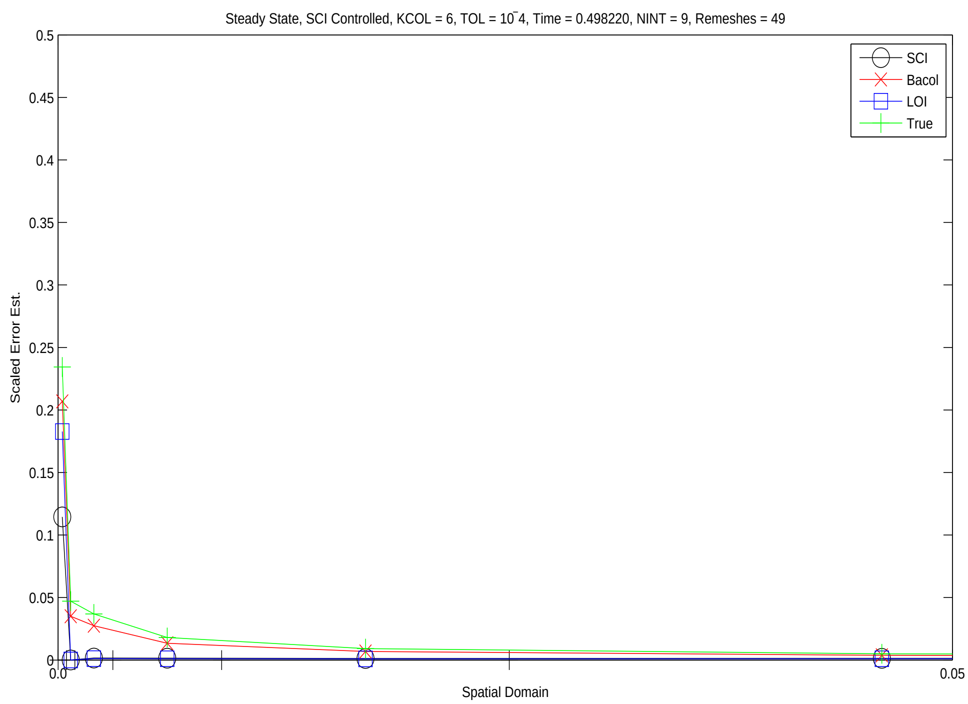
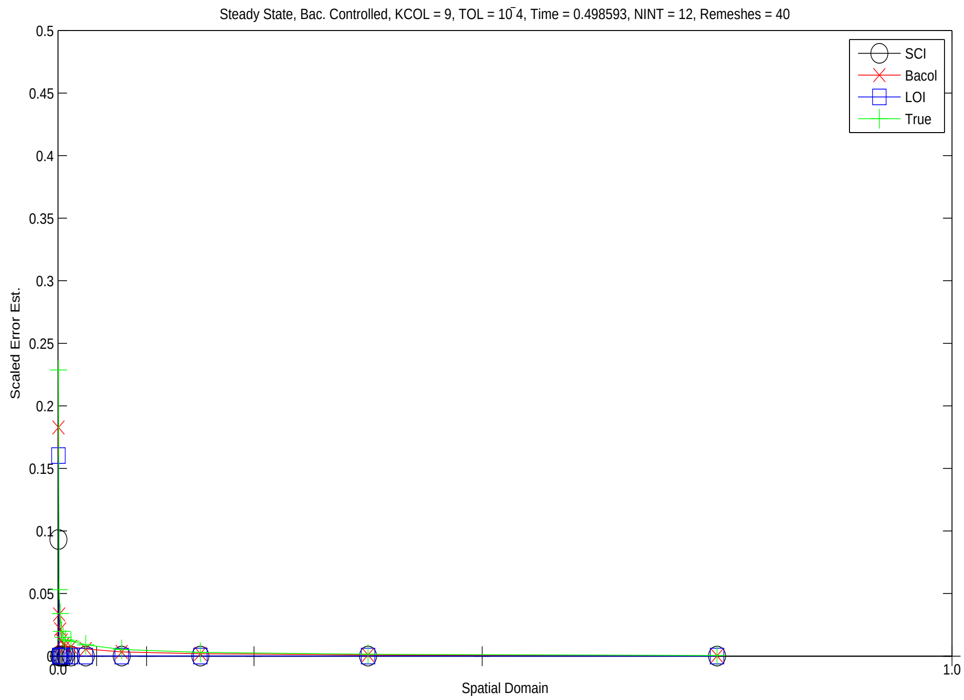


Figure 87: *Results only for the extreme left end of the spatial domain. When BACOL estimates control the mesh, there is good agreement between the estimates and the true error. When the SCI estimates control the mesh, the LOI estimates are too small on part of the layer region.*



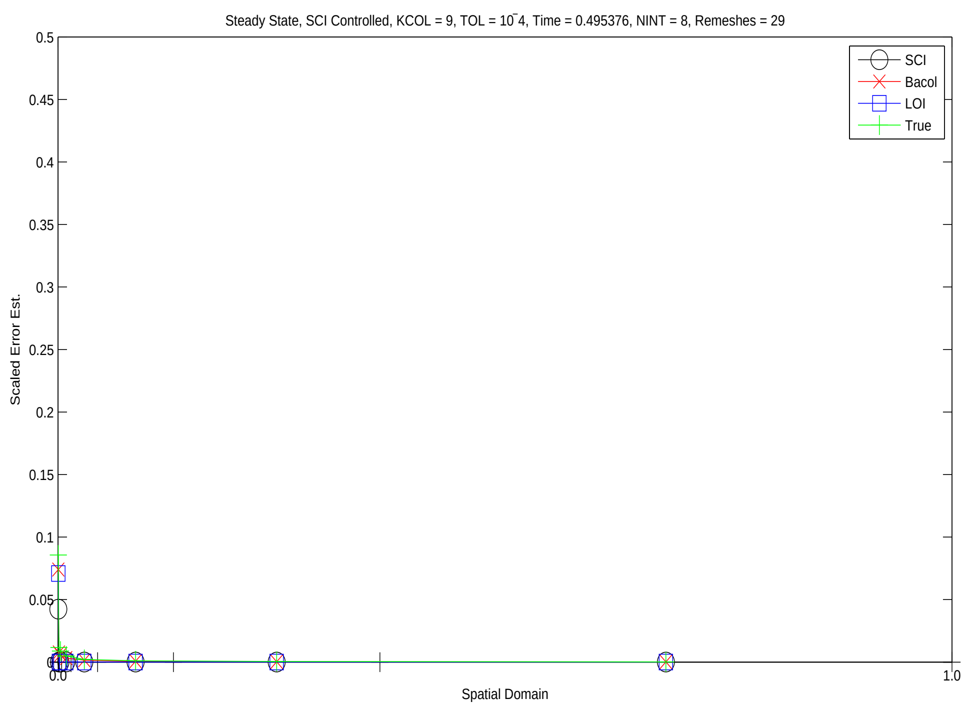
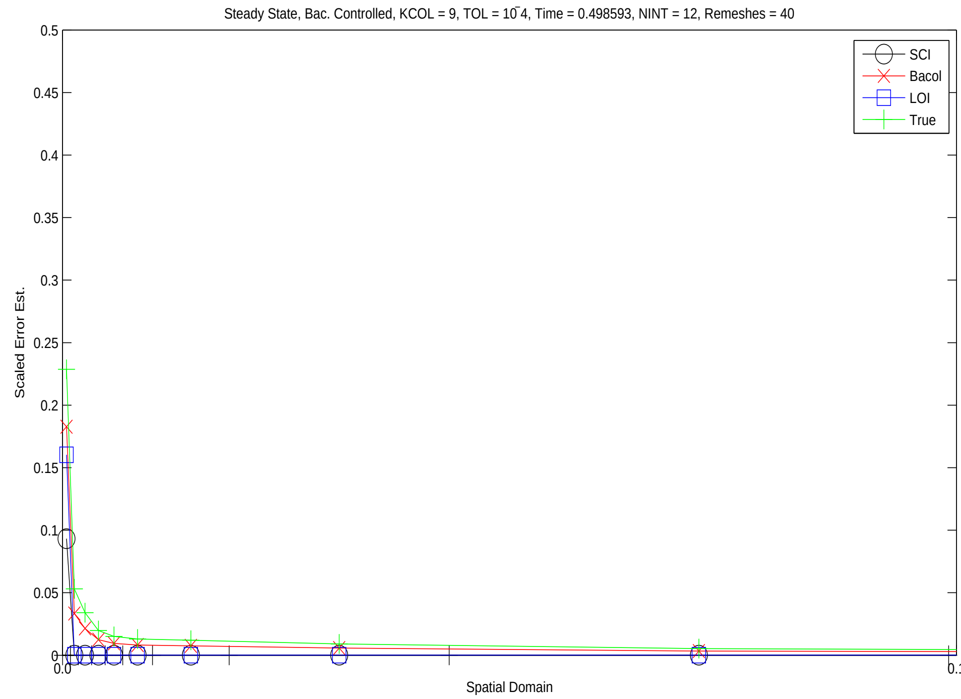


Figure 88: *When the BACOL or SCI estimates are in control of the mesh, there is generally good agreement between all the schemes and the true error.*



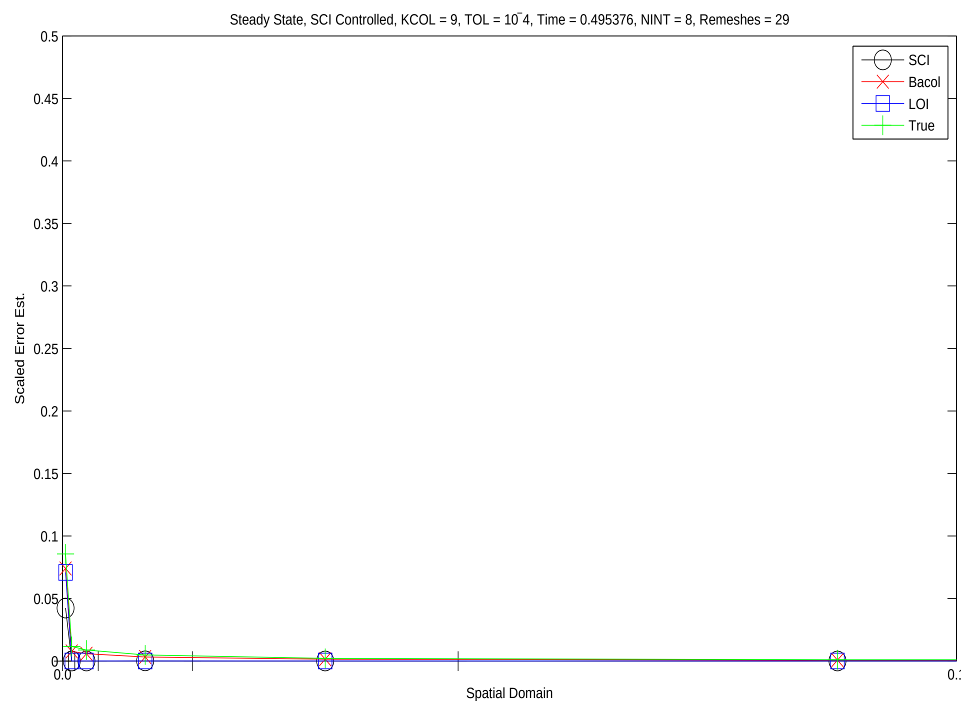
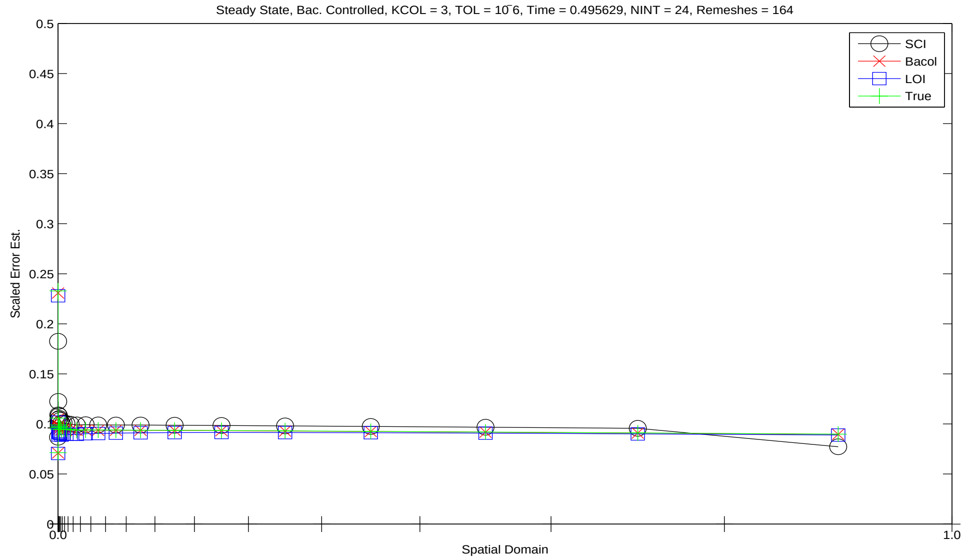


Figure 89: *Results only for the extreme left end of the spatial domain. When the BACOL estimates are in control of the mesh, there is generally good agreement between all the schemes and the true error, except for the left part of the spatial domain, where the LOI scheme underestimates the error. With the SCI estimates in control of the mesh, there is good agreement between the estimates and the true error.*

9.5.3 Results for Tolerance $= 10^{-6}$



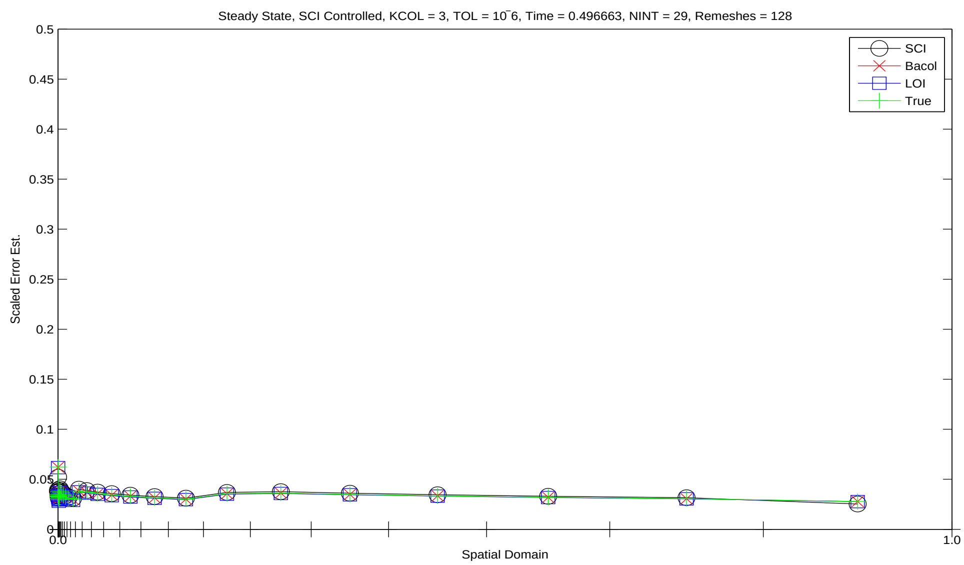
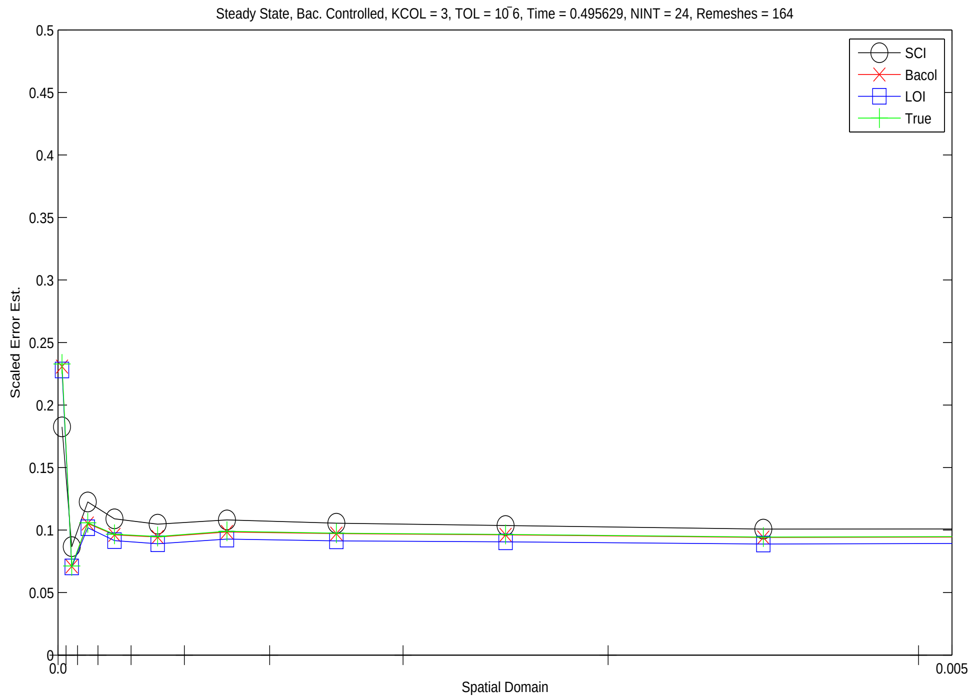


Figure 90: *Good agreement, with either estimate in control of the mesh, between the estimates and the true error throughout the spatial domain.*



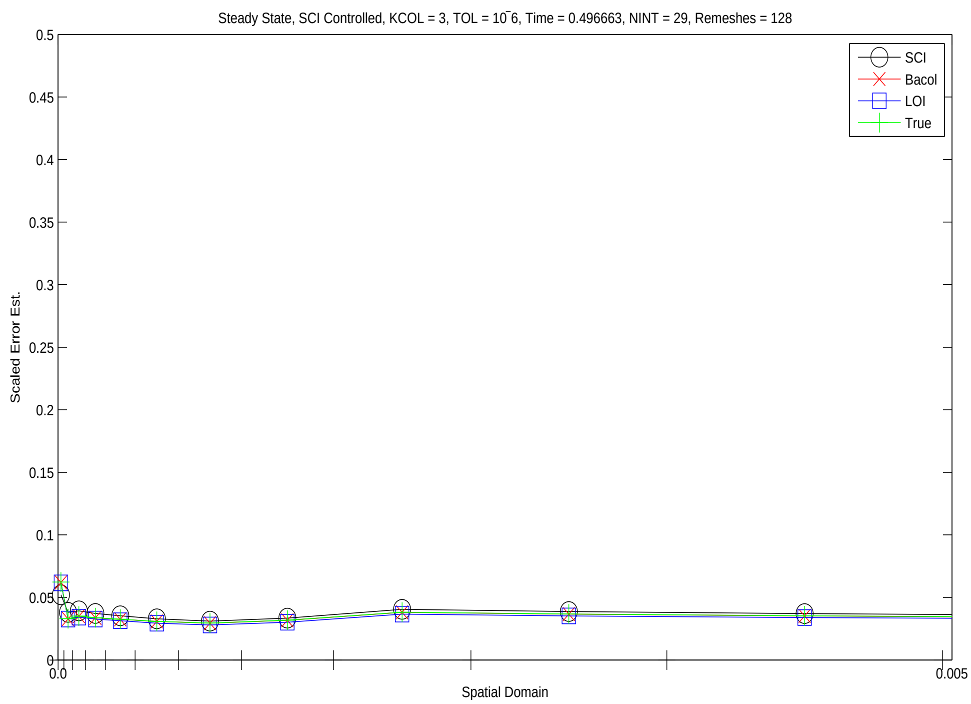


Figure 91: *Results only for the extreme left end of the spatial domain. Good agreement, with either estimate in control of the mesh, between the estimates and the true error.*

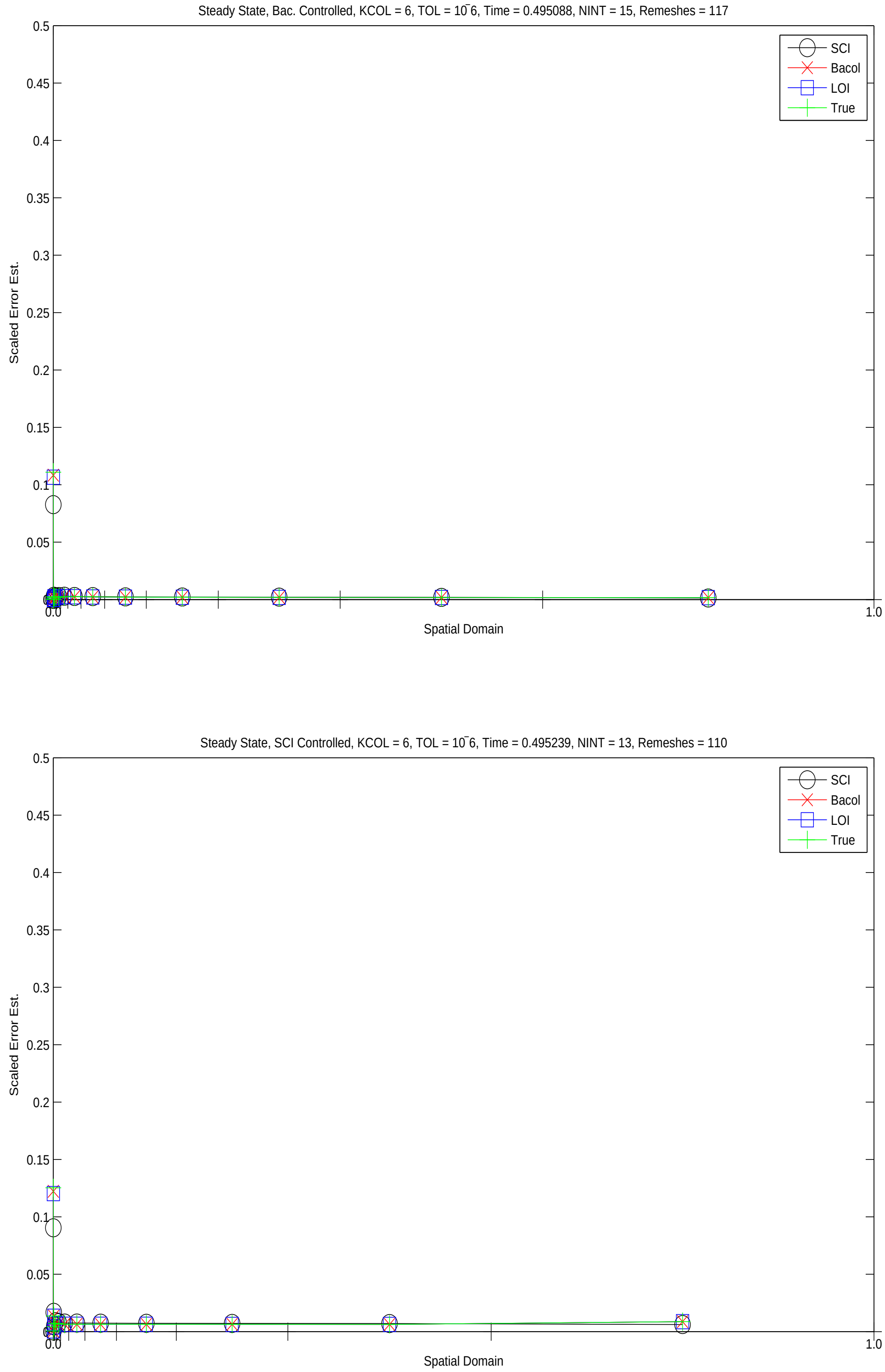
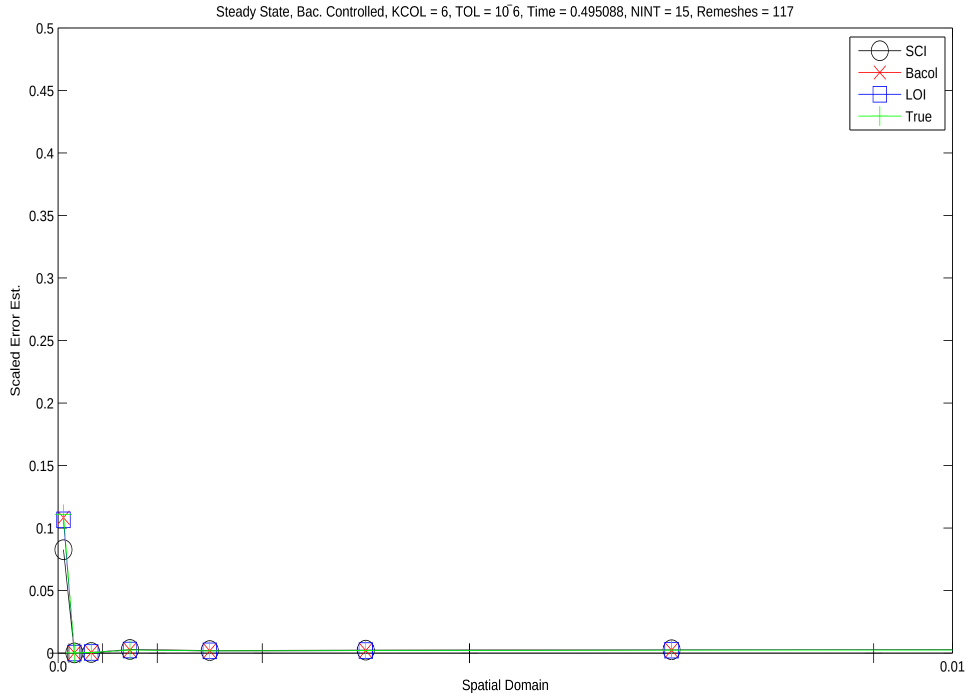


Figure 92: *Good agreement, with either estimate in control of the mesh, between the estimates and the true error throughout the spatial domain.*



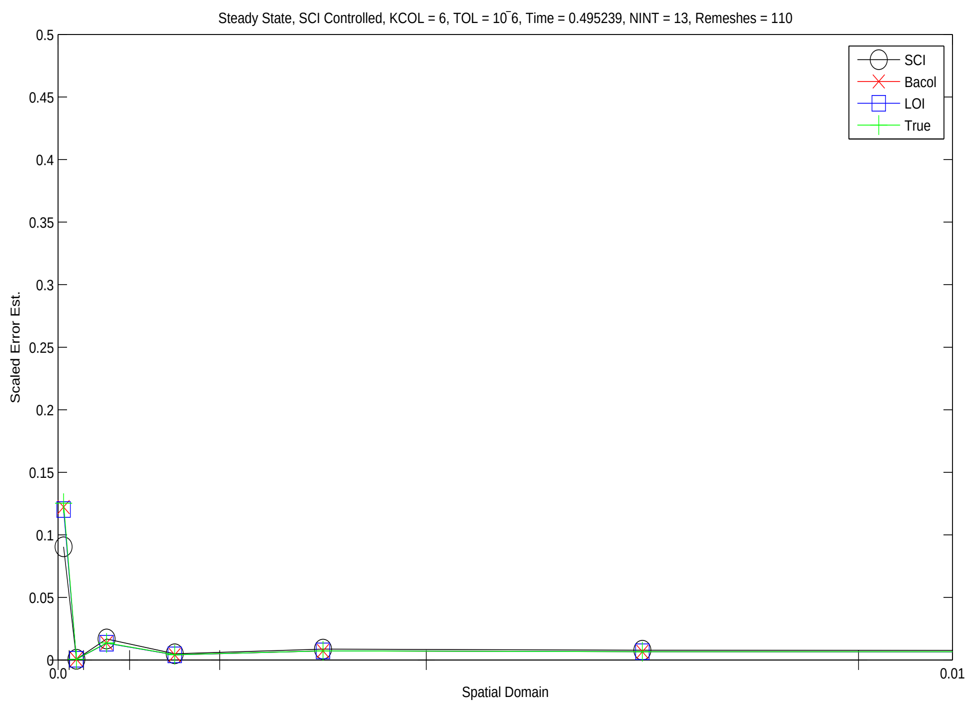
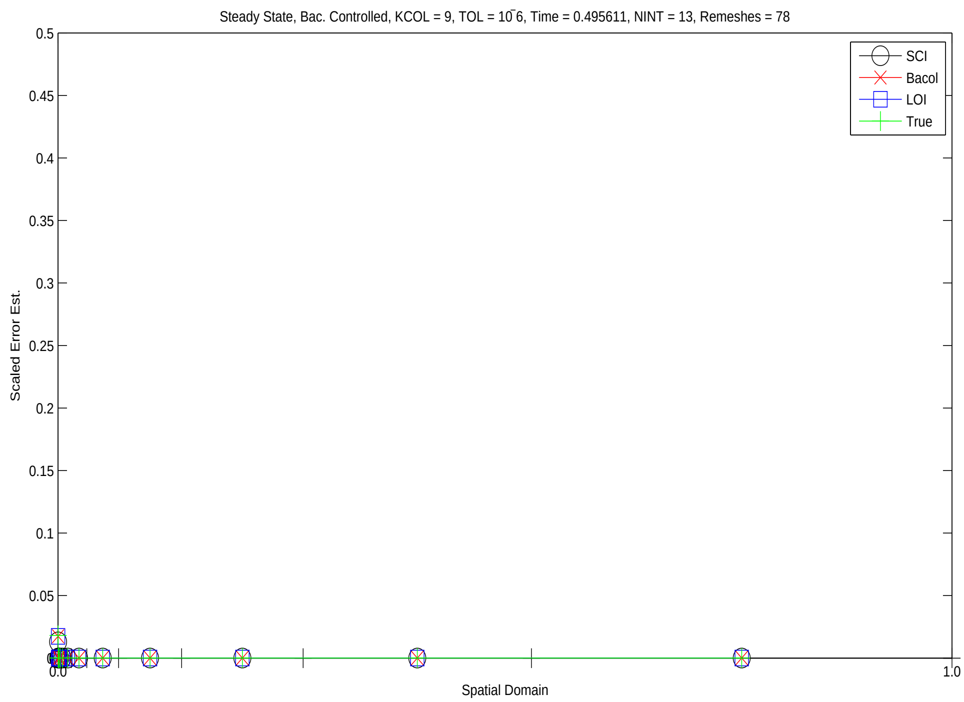


Figure 93: *Results only for the extreme left end of the spatial domain. Good agreement, with either estimate in control of the mesh, between the estimates and the true error.*



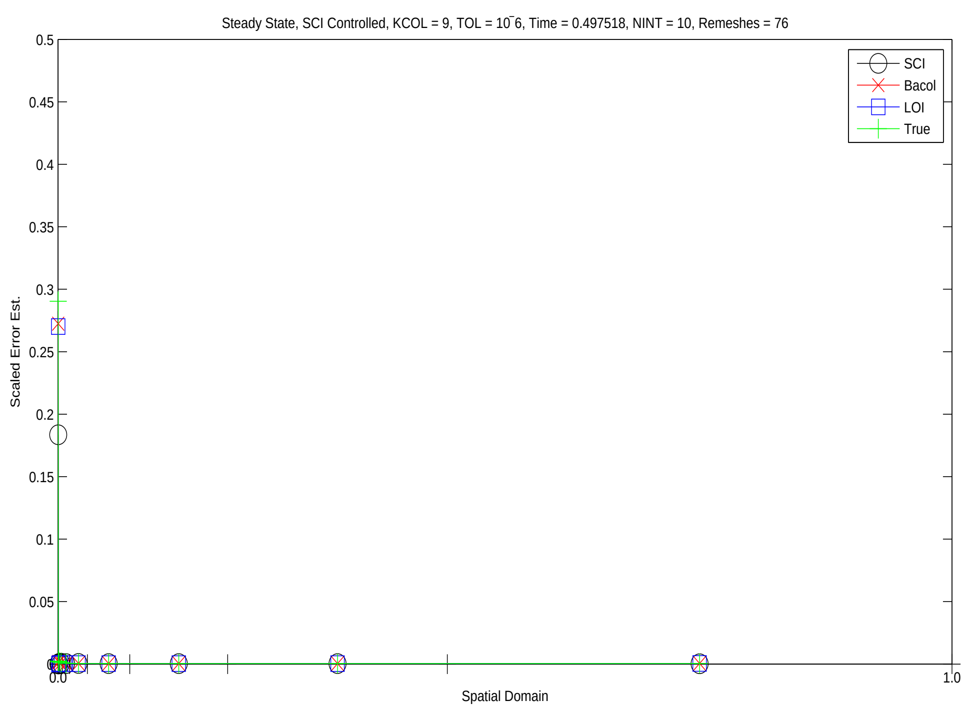


Figure 94: *Good agreement, with either estimate in control of the mesh, between the estimates and the true error throughout the spatial domain.*

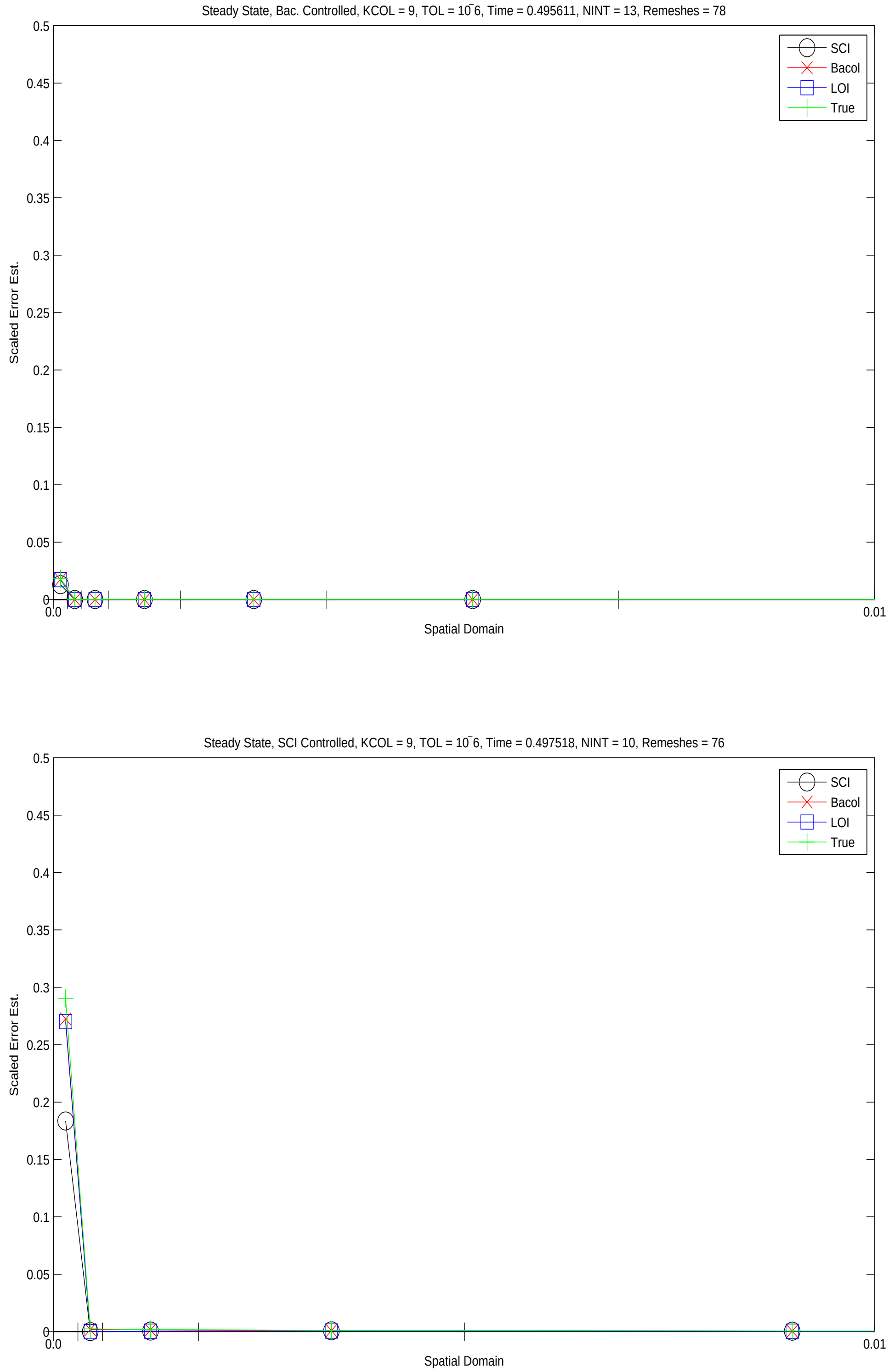
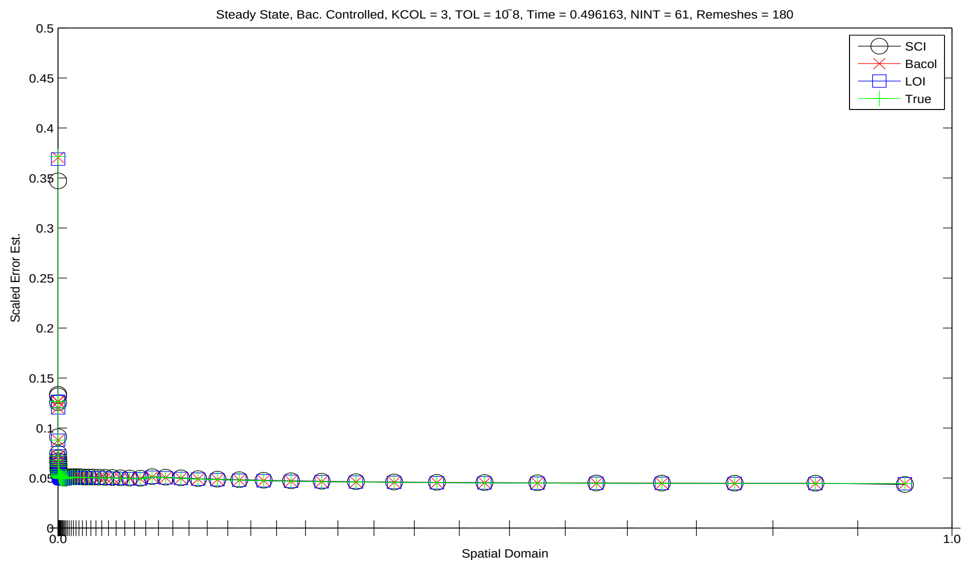


Figure 95: *Results only for the extreme left end of the spatial domain. Good agreement, with either estimate in control of the mesh, between the estimates and the true error.*

9.5.4 Results for Tolerance = 10^{-8}



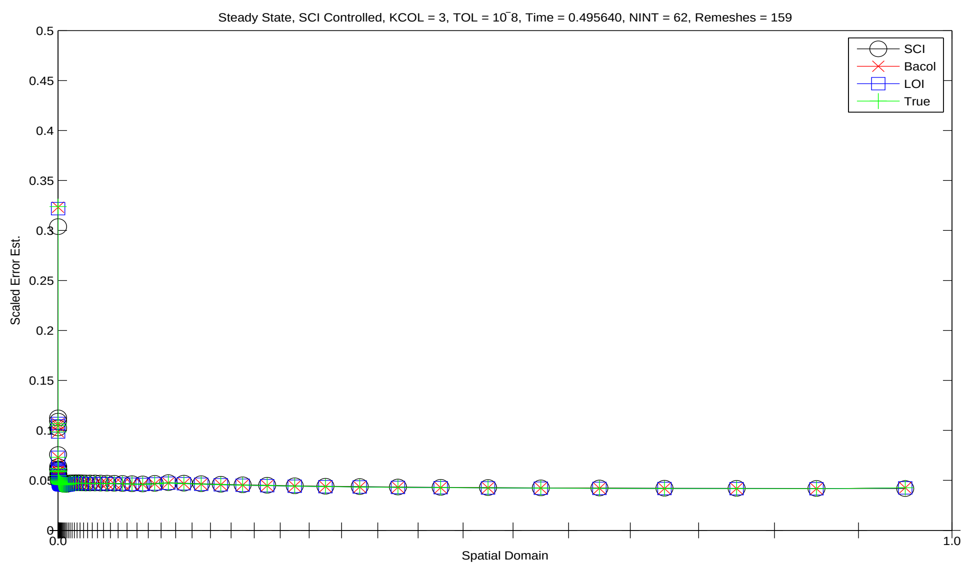


Figure 96: *Good agreement, with either estimate in control of the mesh, between the estimates and the true error throughout the spatial domain.*

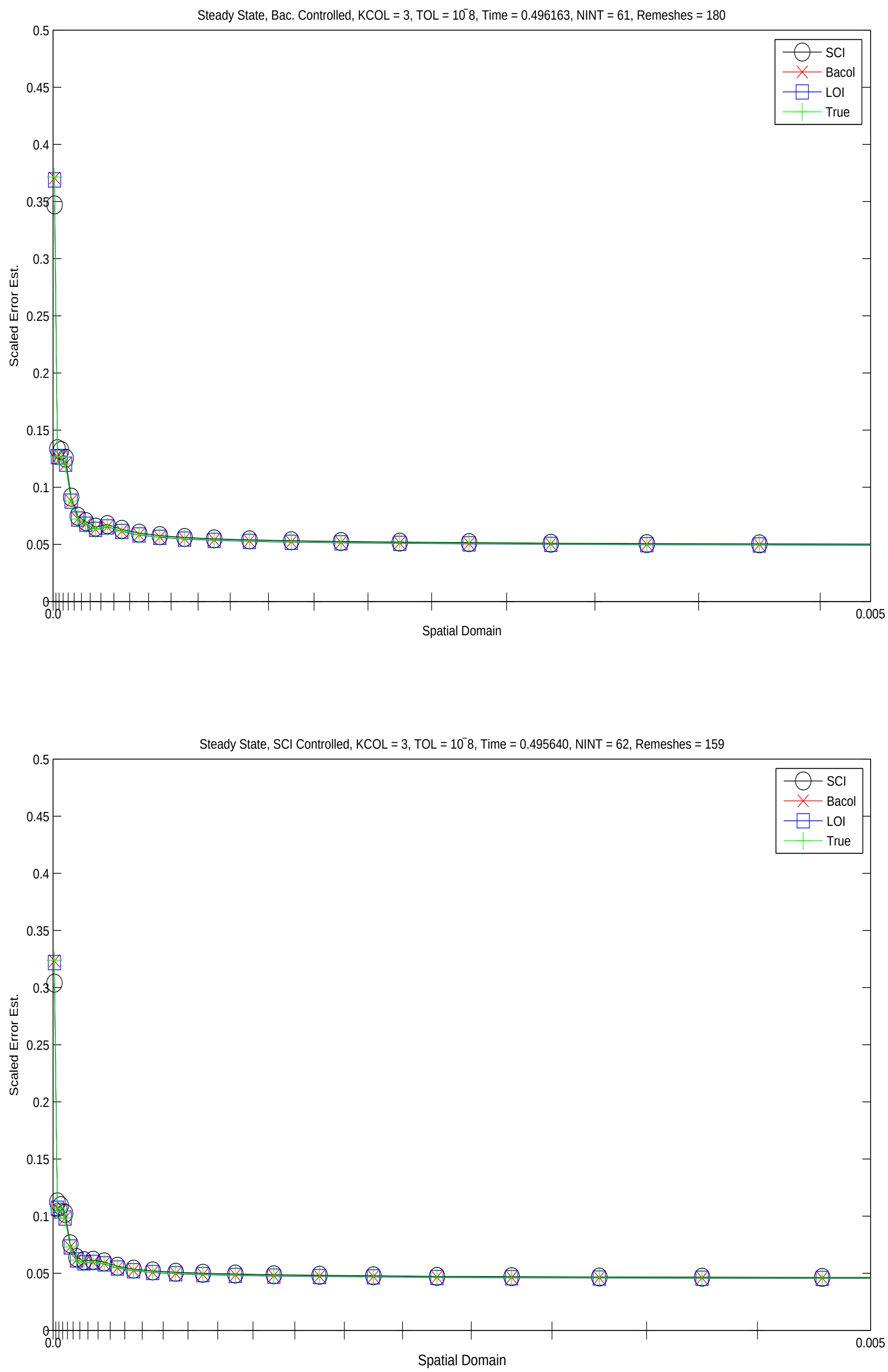
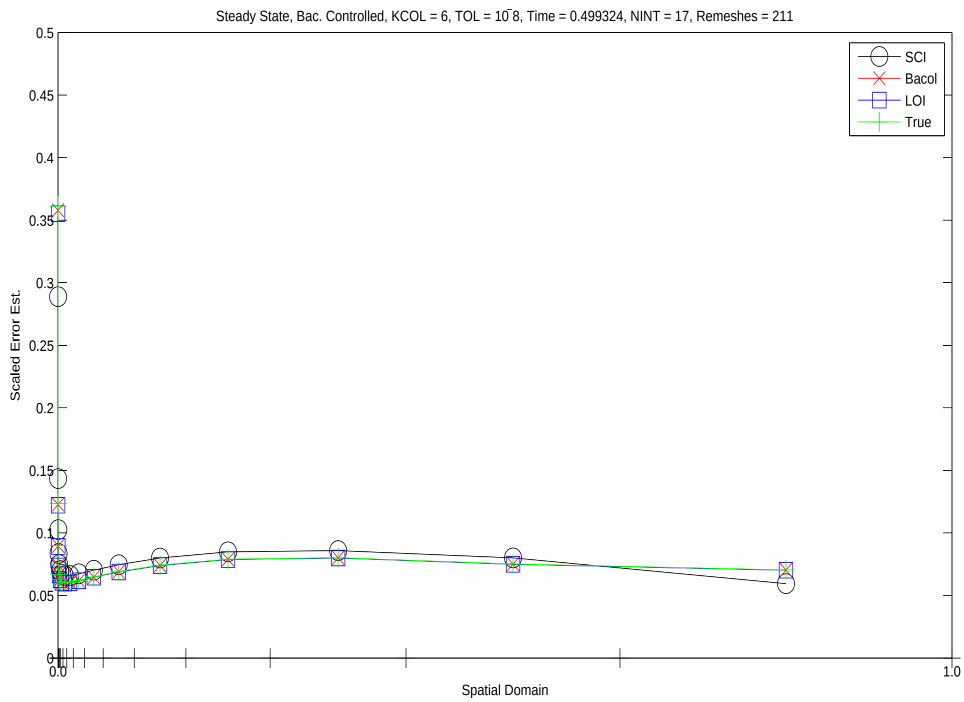


Figure 97: *Results only for the extreme left end of the spatial domain. Good agreement, with either estimate in control of the mesh, between the estimates and the true error.*



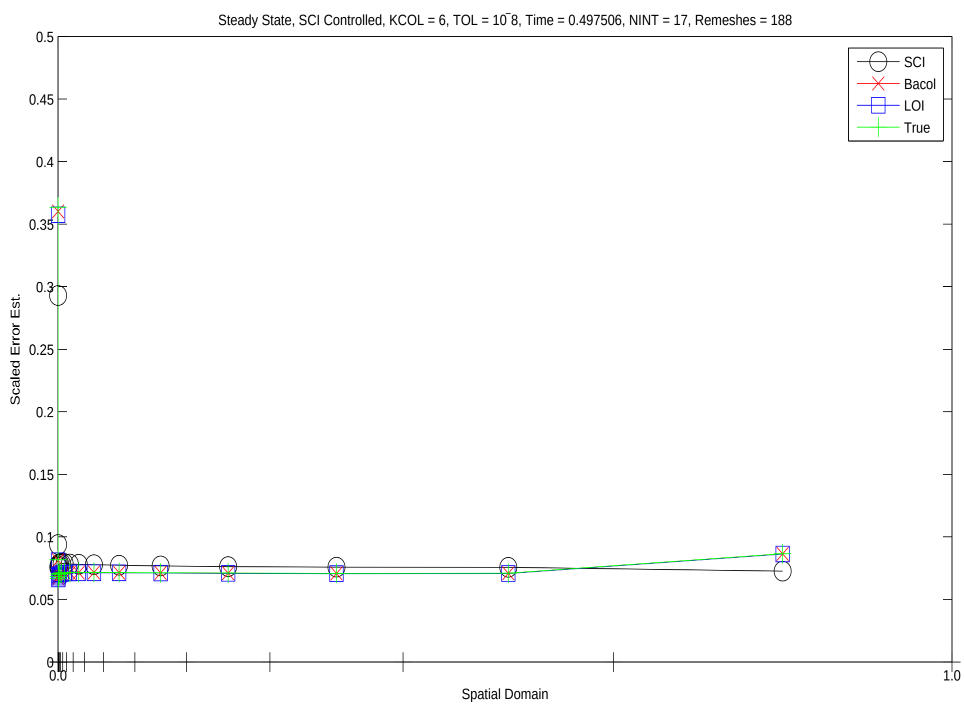
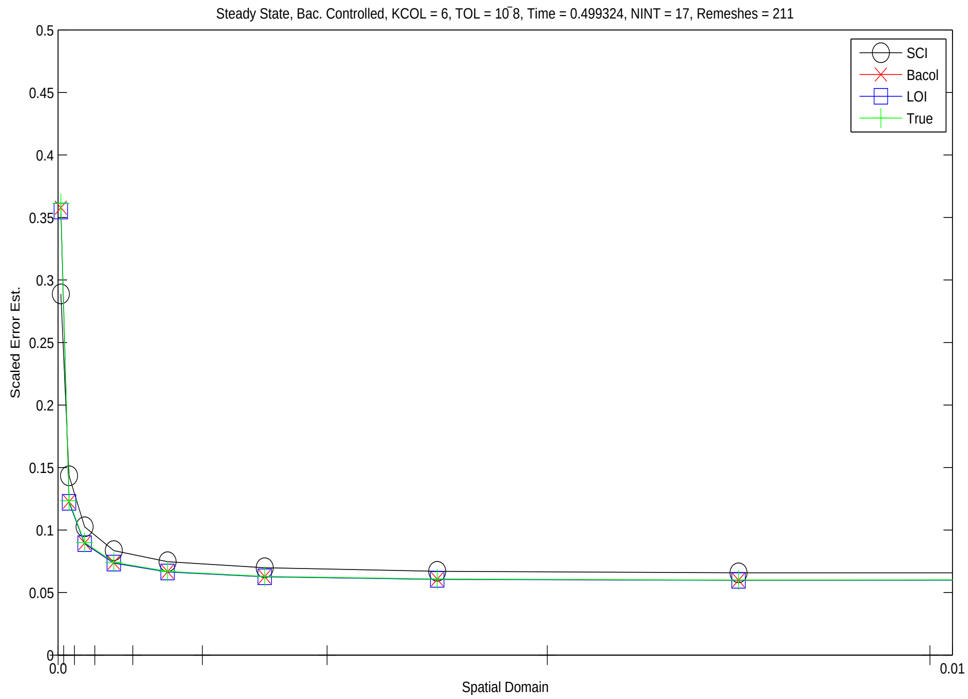


Figure 98: *Good agreement, with either estimate in control of the mesh, between the estimates and the true error.*



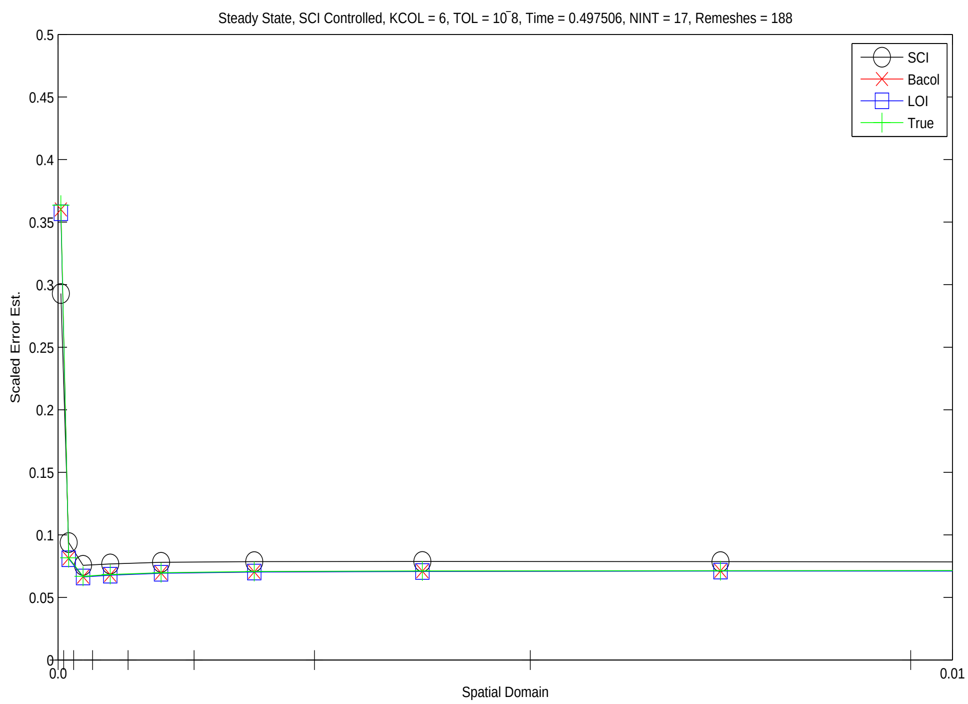
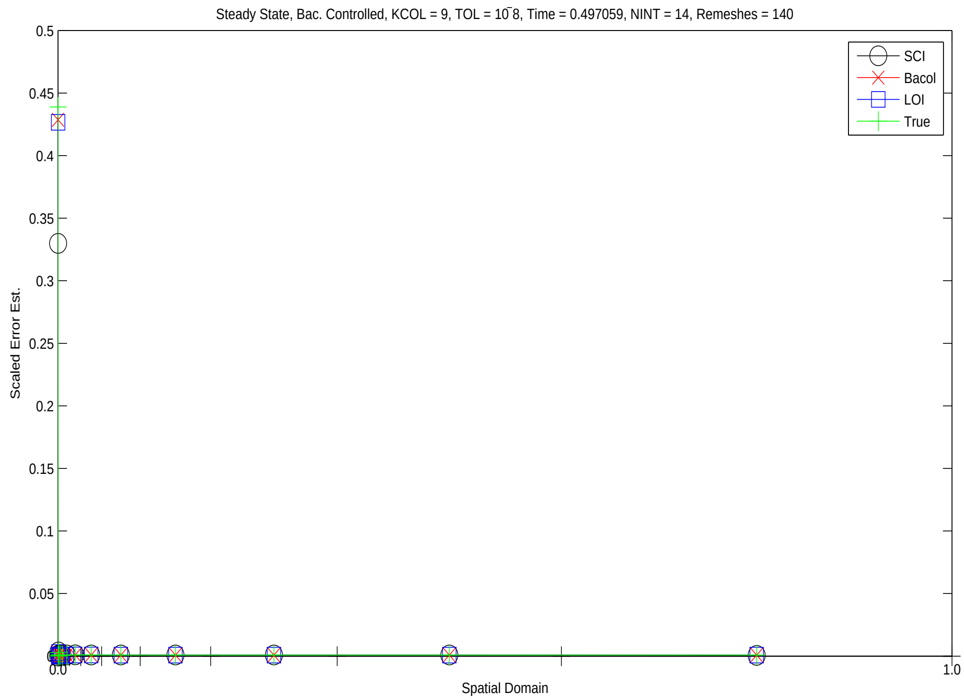


Figure 99: *Results only for the extreme left end of the spatial domain. Good agreement, with either estimate in control of the mesh, between the estimates and the true error.*



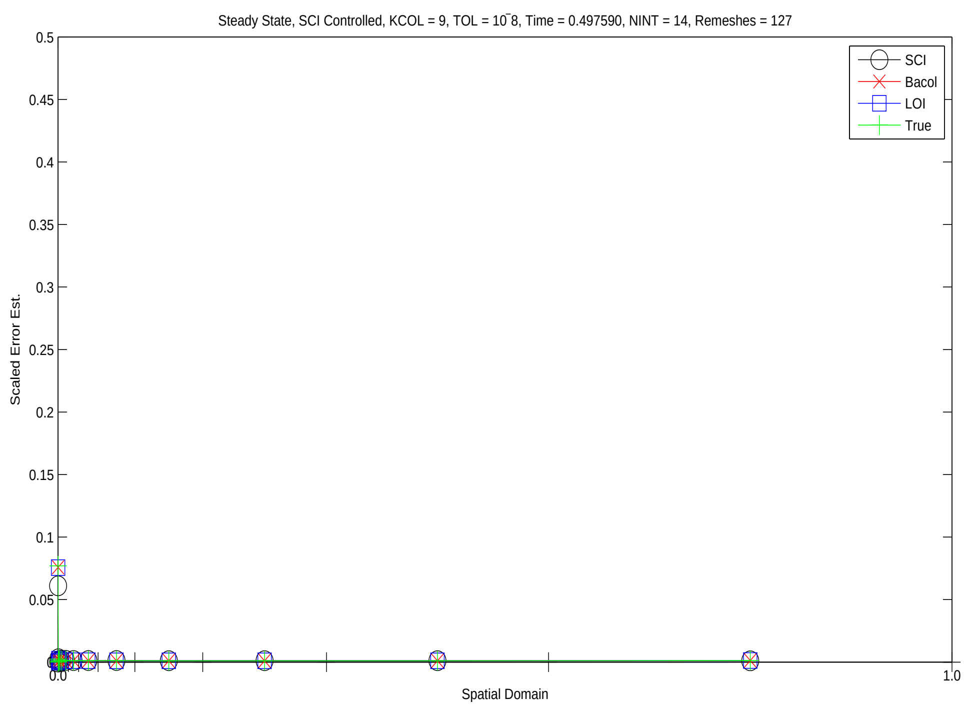
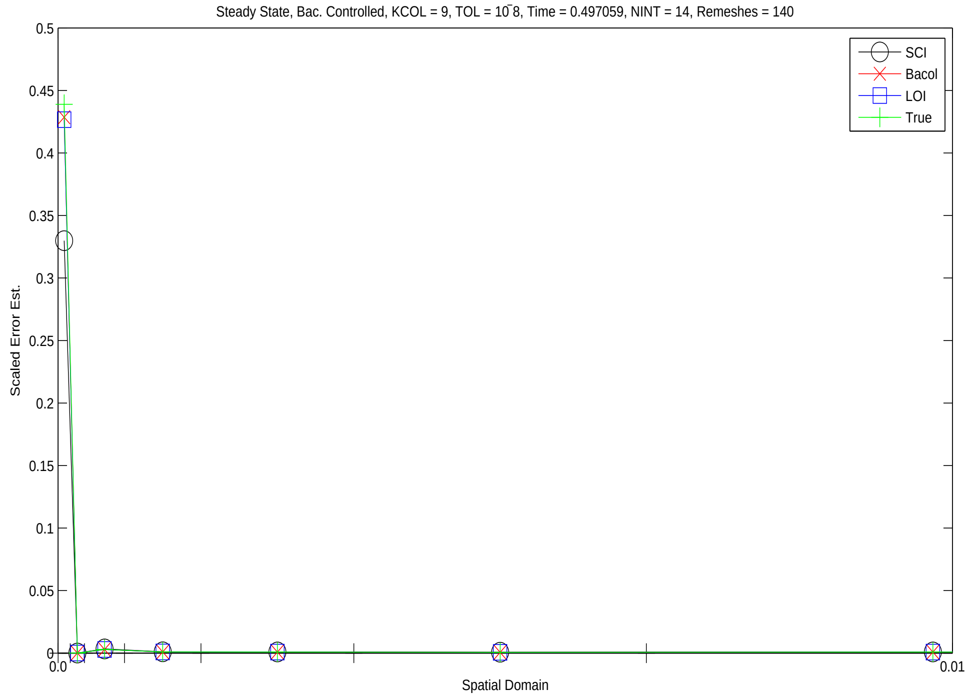


Figure 100: *Good agreement, with either estimate in control of the mesh, between the estimates and the true error.*



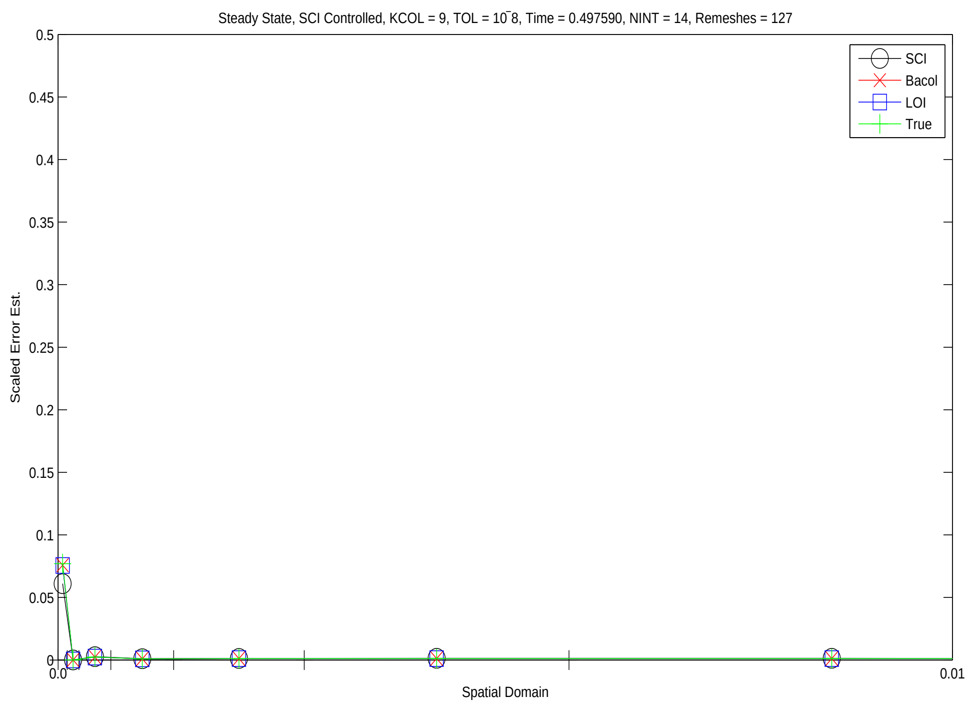


Figure 101: *Results only for the extreme left end of the spatial domain. Good agreement, with either estimate in control of the mesh, between the estimates and the true error.*

9.5.5 Summary of Results

TOL \ KCOL		3	6	9
		NINT, Remeshes	NINT, Remeshes	NINT, Remeshes
10^{-4}	BAC	15, 90	10, 57	12, 40
	SCI	15, 87	9, 49	8, 29
10^{-6}	BAC	24, 164	15, 117	13, 78
	SCI	29, 128	13, 110	10, 76
10^{-8}	BAC	61, 180	17, 211	14, 140
	SCI	62, 159	17, 188	14, 127

Table 10: *Generably comparable performance when either the BACOL or SCI schemes control the mesh except for the $KCOL = 9, TOL = 10^{-4}$ case where the BACOL scheme uses more subintervals and has more remeshings.*

9.6 Reaction Convection Diffusion System

9.6.1 Problem

The catalytic surface reaction model, [42]:

$$\begin{aligned}
(u_1)_t &= -(u_1)_x + n(D_1 u_3 - A_1 u_1 \gamma) + (u_1)_{xx}/Pe_1,\\
(u_2)_t &= -(u_2)_x + n(D_2 u_4 - A_2 u_2 \gamma) + (u_2)_{xx}/Pe_1,\\
(u_3)_t &= A_1 u_1 \gamma - D_1 u_3 - R u_3 u_4 \gamma^2 + (u_3)_{xx}/Pe_2,\\
(u_4)_t &= A_2 u_2 \gamma - D_2 u_4 - R u_3 u_4 \gamma^2 + (u_4)_{xx}/Pe_2,
\end{aligned} \tag{19}$$

where $\gamma = 1 - u_3 - u_4, 0 < x < 1, t > 0$, and $n, r, Pe_1, Pe_2, D_1, D_2, R, A_1$, and A_2 are problem dependent parameters. The initial conditions are

$$u_1(x,0) = 2 - r, \quad u_2(x,0) = r, \quad u_3(x,0) = u_4(x,0) = 0,$$

and the boundary conditions are

$$(u_1)_x(0,t) = -Pe_1(2 - r - u_1(0,t)), \quad (u_2)_x(0,t) = -Pe_1(r - u_2(0,t)),$$

$$(u_3)_x(0,t) = (u_4)_x(0,t) = 0,$$

$$(u_1)_x(1,t) = (u_2)_x(1,t) = (u_3)_x(1,t) = (u_4)_x(1,t) = 0.$$

To our knowledge, this problem does not have a closed form solution. With $Pe_1 = Pe_2 = 100, D_1 = 1.5, D_2 = 1.2, R = 1000$, and $A_1 = A_2 = 30$, a plot of the first solution component is shown in Figure 102. The following

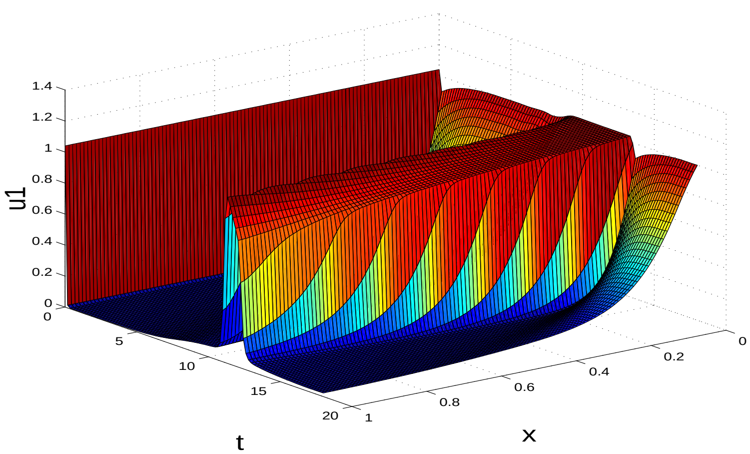
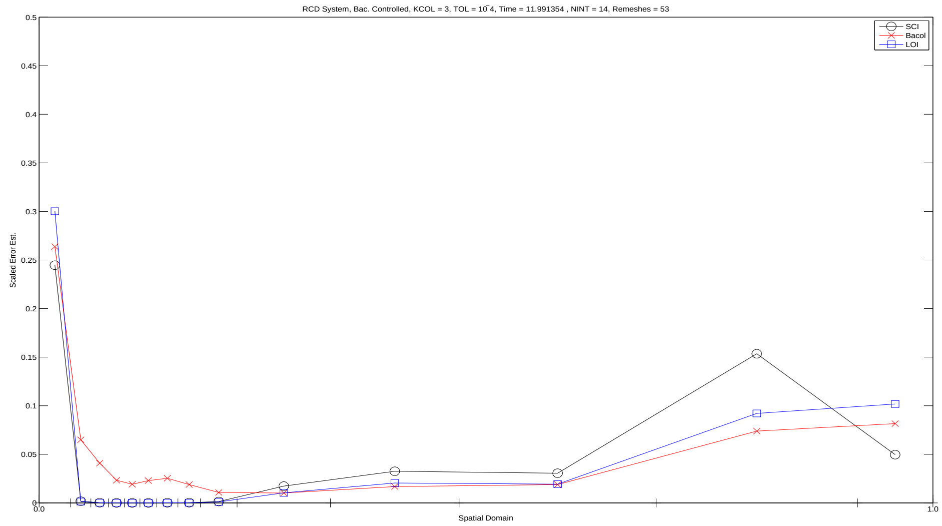


Figure 102: *Catalytic Surface Reaction Model,* $u_1(x,t)$

three subsubsections provide results tolerances of 10^{-4}, 10^{-6}, and 10^{-8}. Within each subsubsection, we provide results for $k = KCOL = 3, 6$, and 9. The title of each graph indicates: the problem (RCD System), which of the error estimation schemes controls the mesh refinement (Bac. Controlled or SCI Controlled), the value of KCOL, the tolerance TOL, the final Time (close to 12.0), the number of subintervals in the final mesh, NINT, and the total number of spatial remeshings that took place over the entire computation, Remeshes.

9.6.2 Results for Tolerance = 10^{-4}



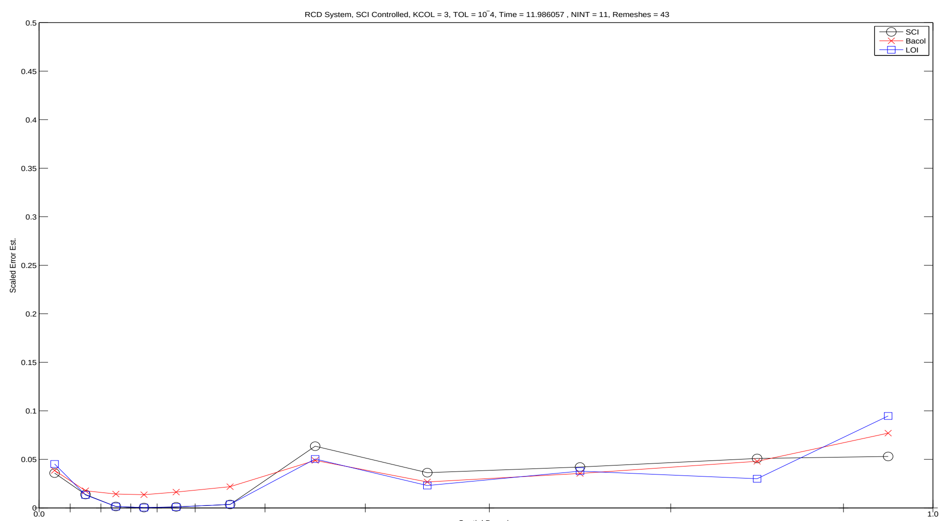


Figure 103: *When the BACOL estimates control the mesh, there is generally good agreement among the three schemes, except just to the right of the boundary layer at the left end of the spatial domain, where the SCI and LOI estimates are smaller than the BACOL estimate. When the SCI estimates control the mesh, there is generally good agreement anon the three schemes throughout the spatial domain.*

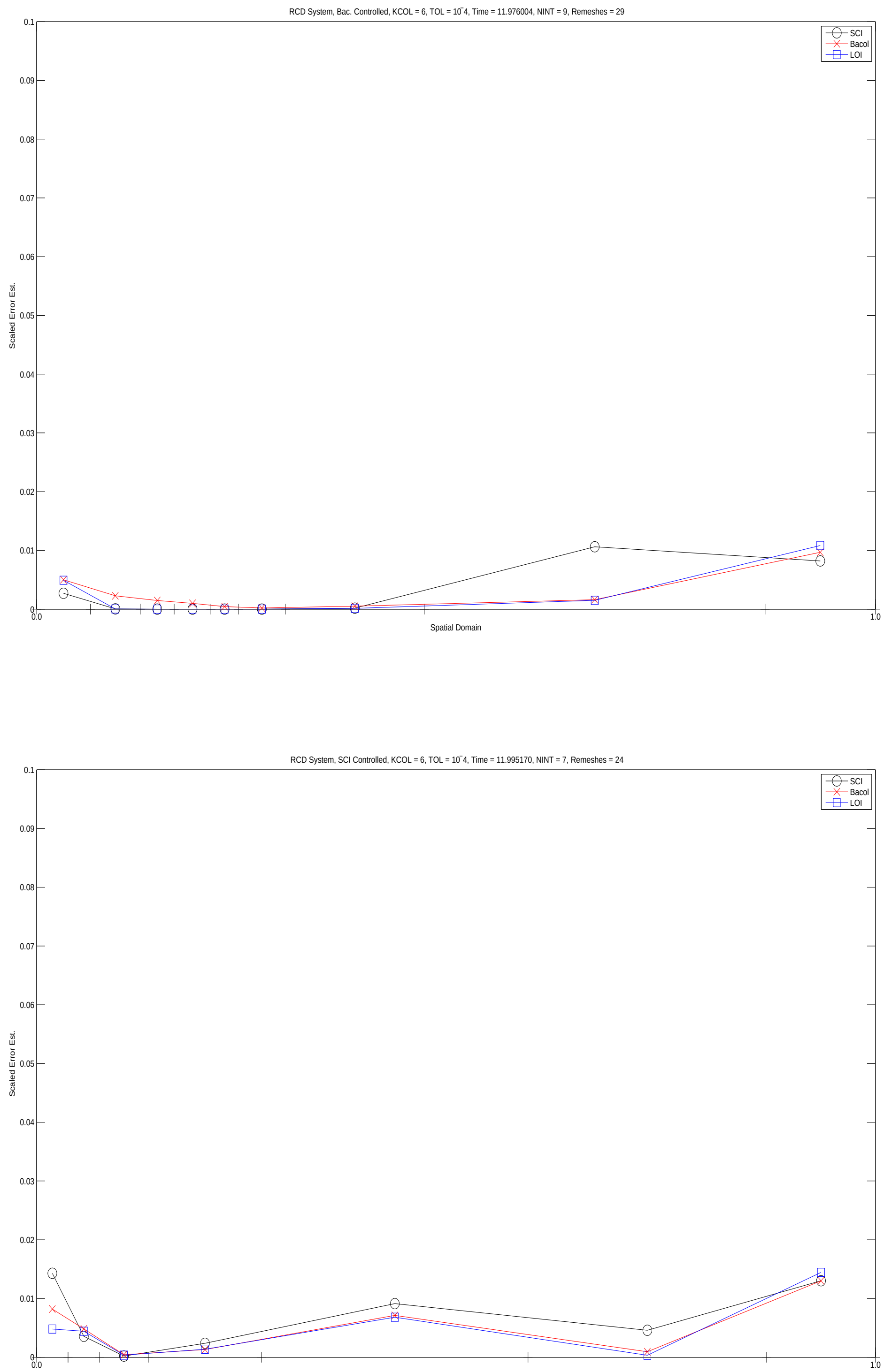
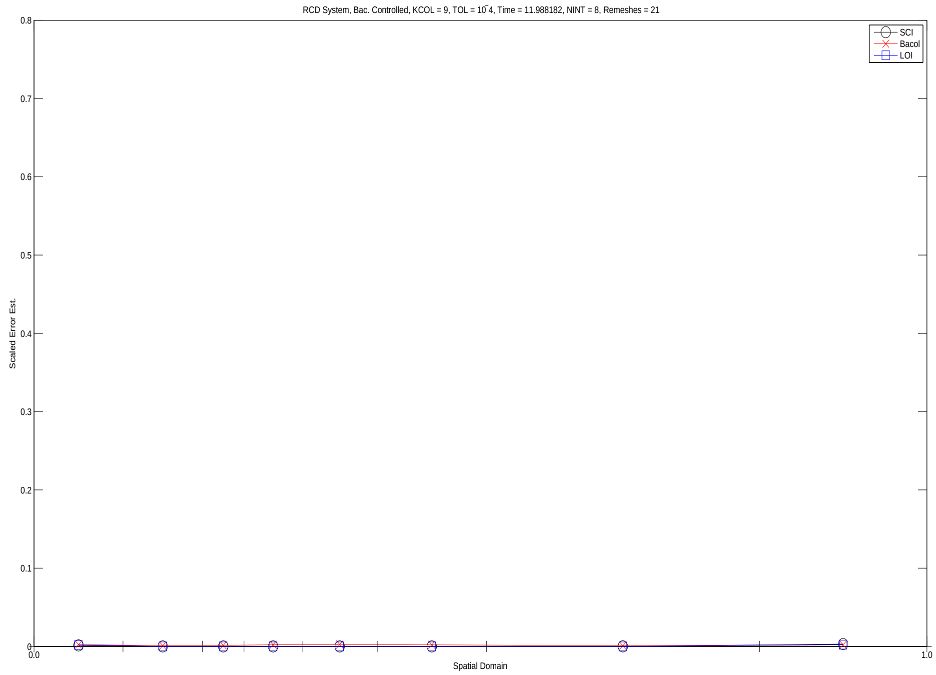


Figure 104: *Generally good agreement among all schemes when either the BACOL or SCI estimates control the mesh.*



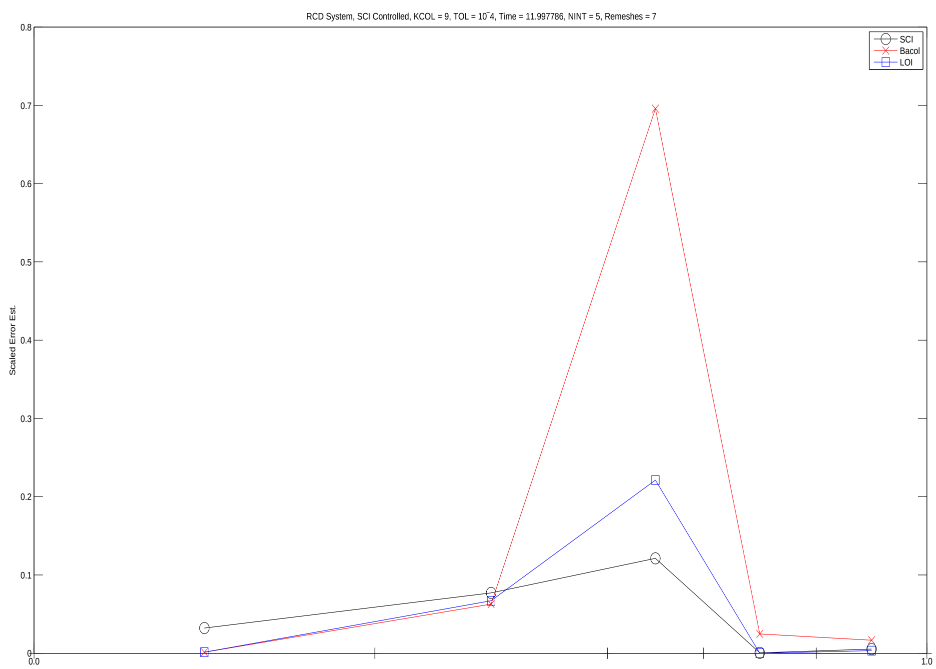
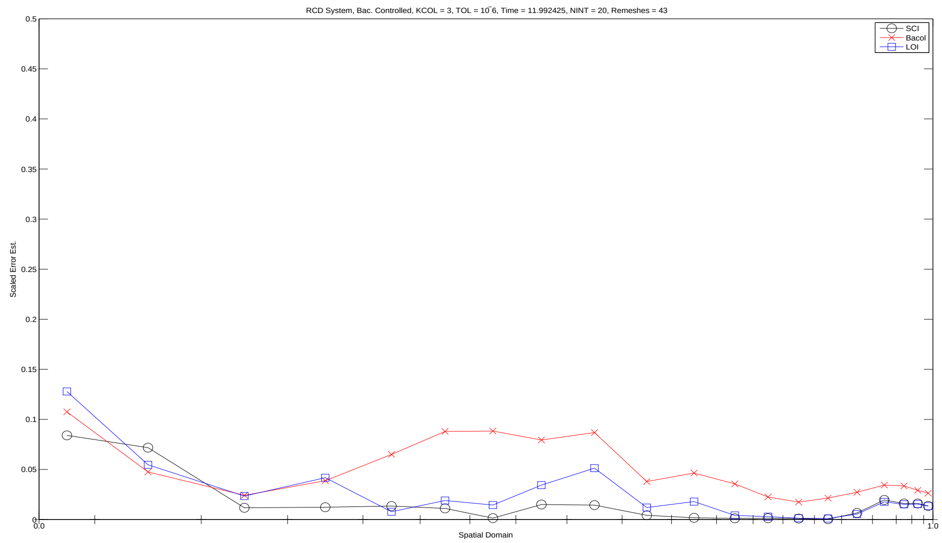


Figure 105: *Good agreement among all three schemes when the BACOL estimates control the mesh. Also generally good agreement among the schemes when the SCI estimates control the mesh, except for the third subinterval where the SCI and LOI schemes give estimates that are smaller than the estimate from the BACOL scheme.*

9.6.3 Results for Tolerance = 10^{-6}



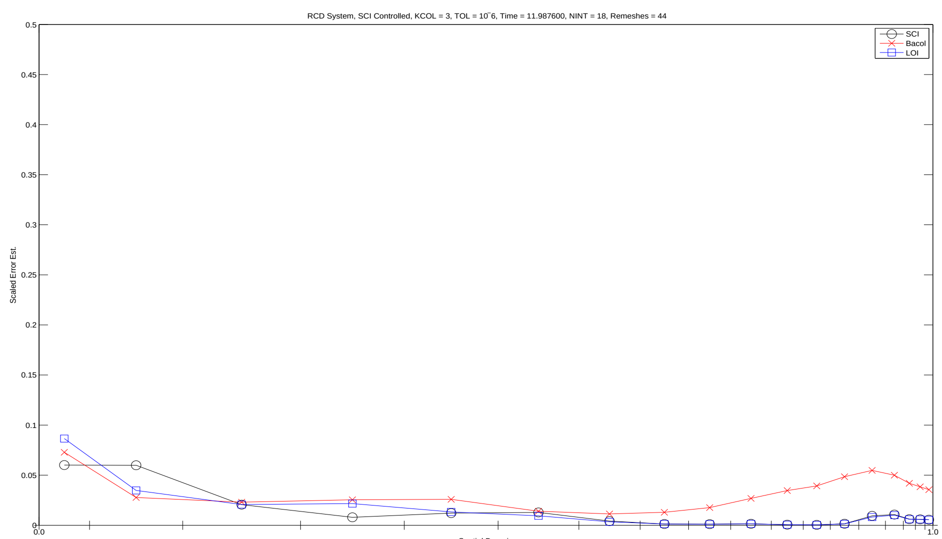
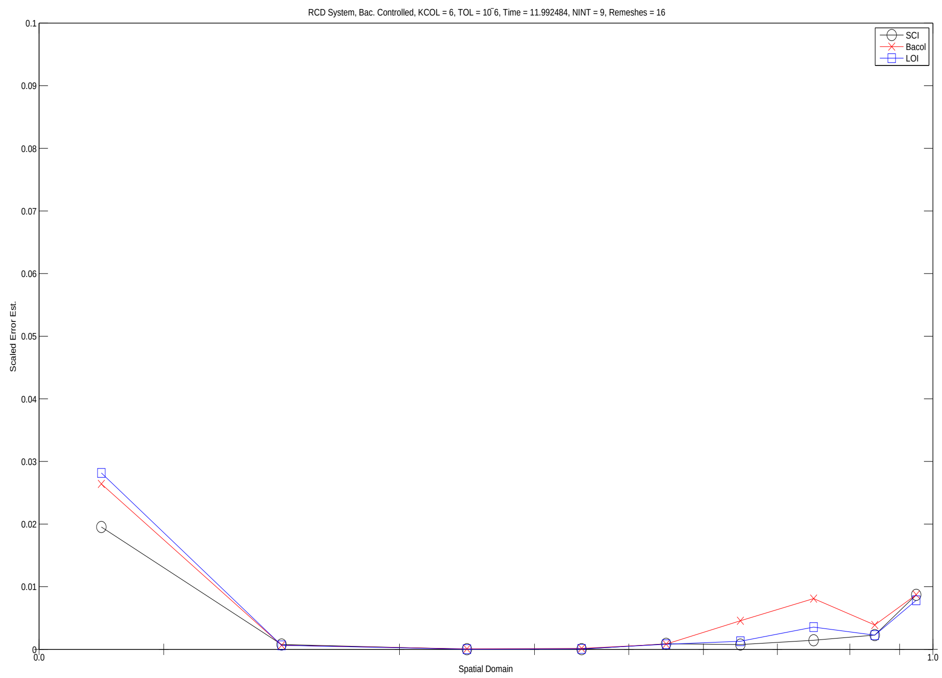


Figure 106: *When the BACOL estimates control the mesh, the SCI and LOI estimates are smaller than the BACOL estimates except for the first third of the spatial domain. When the SCI estimates control the mesh, the SCI and LOI estimates are smaller than the BACOL estimates on the last third of the spatial domain.*



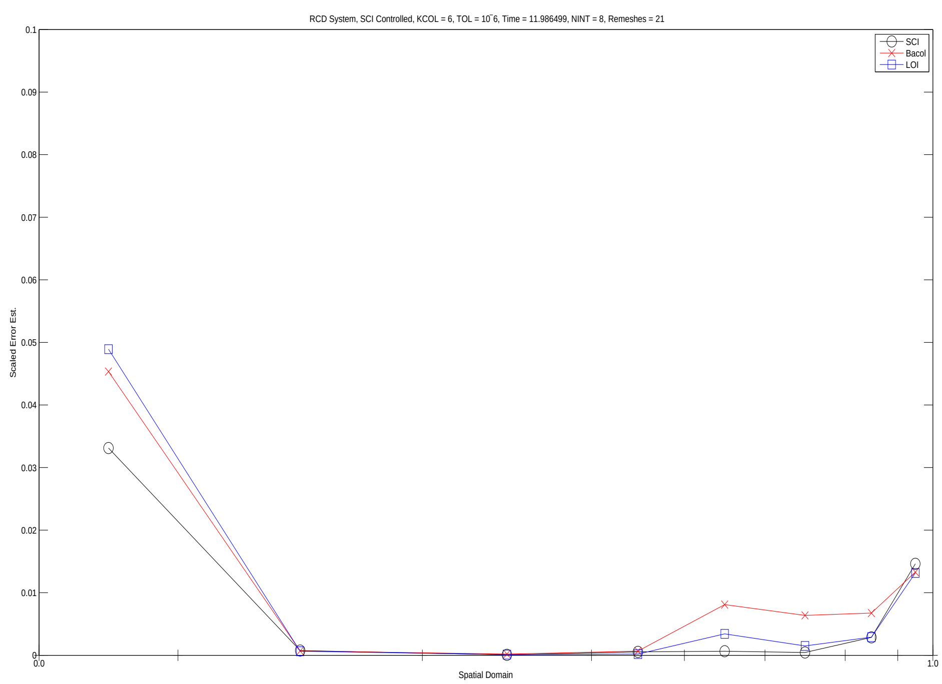
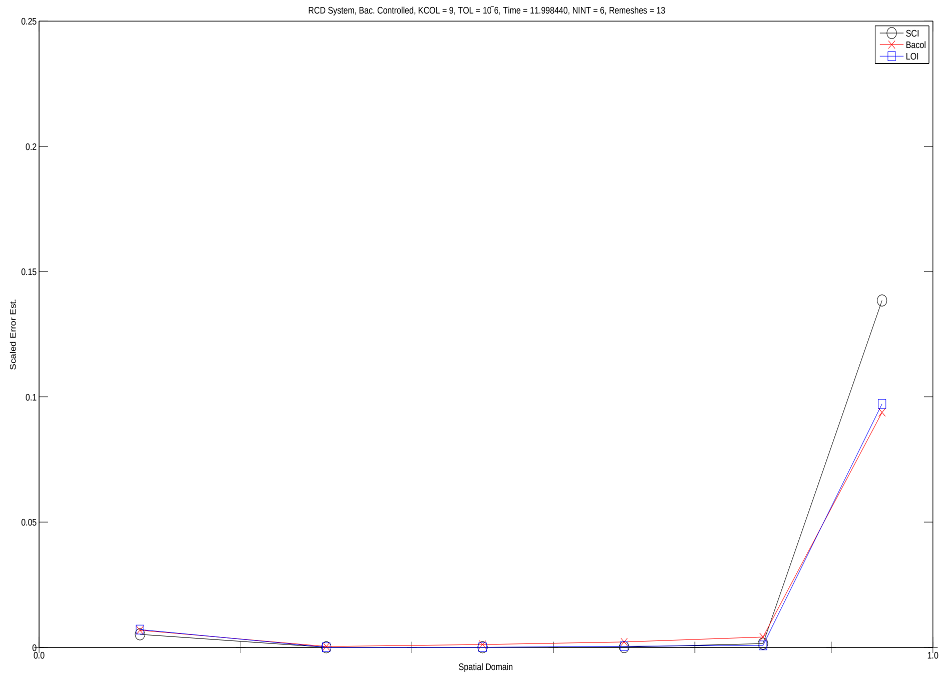


Figure 107: *Generally good agreement among all schemes when either the BACOL or SCI estimates control the mesh.*



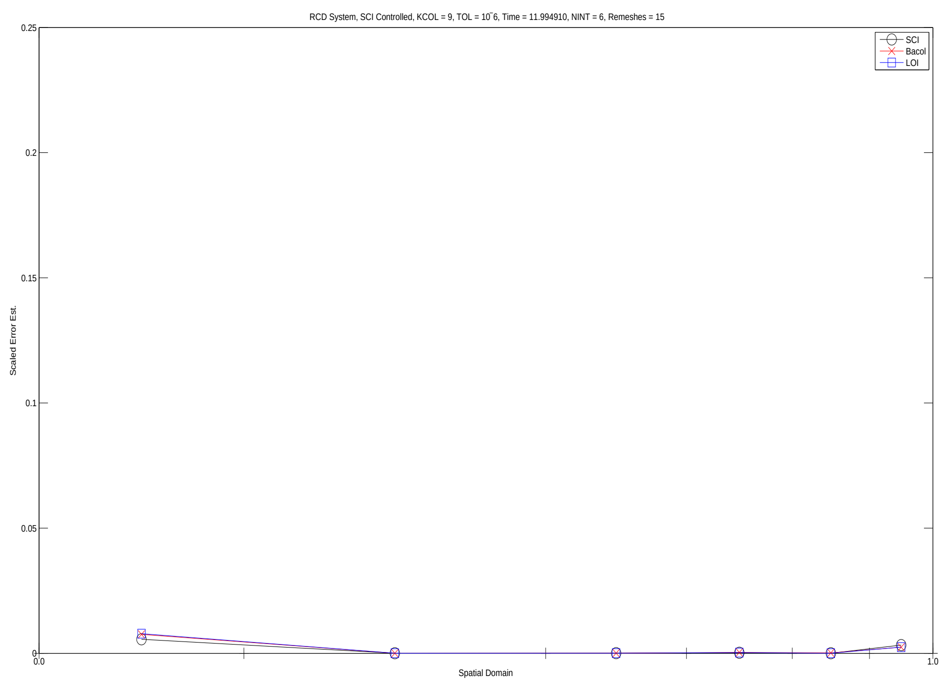
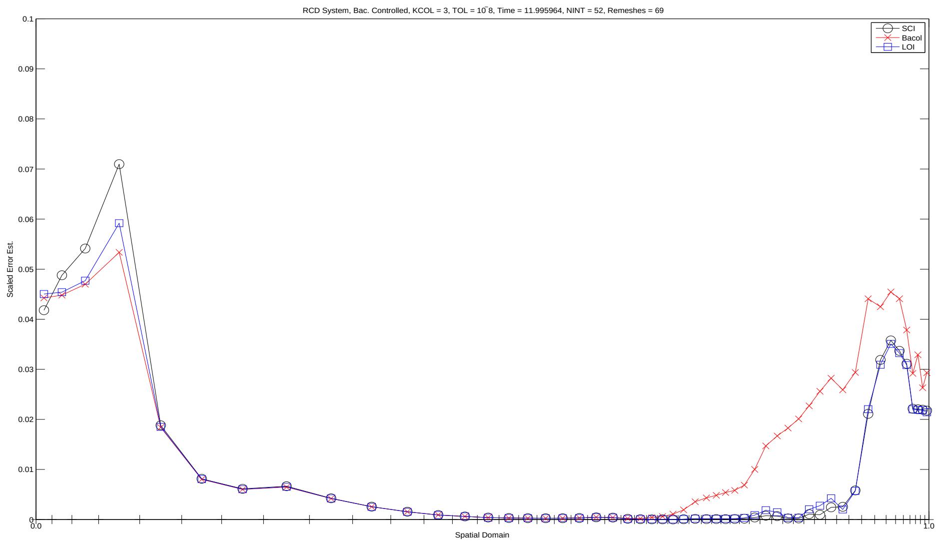


Figure 108: *Generally good agreement among all schemes when either the BACOL or SCI estimates control the mesh.*

9.6.4 Results for Tolerance $= 10^{-8}$



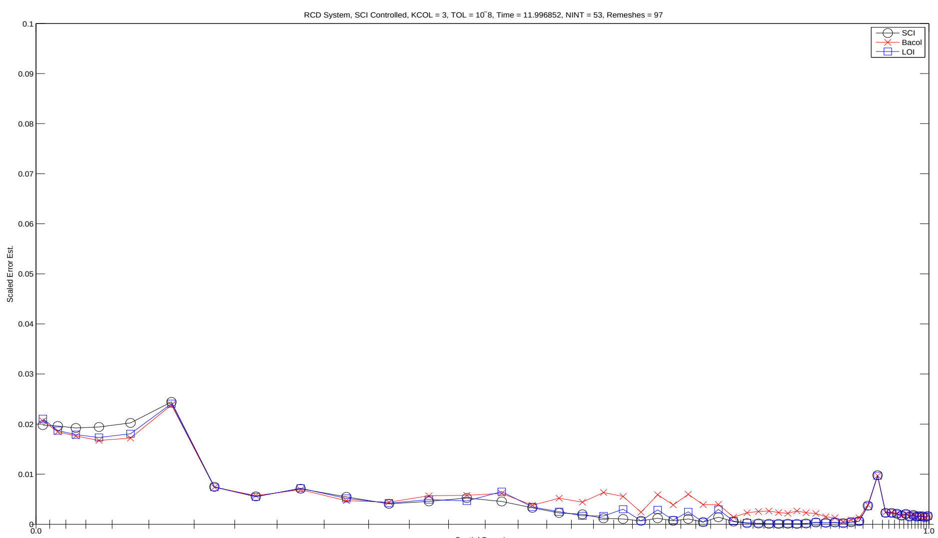
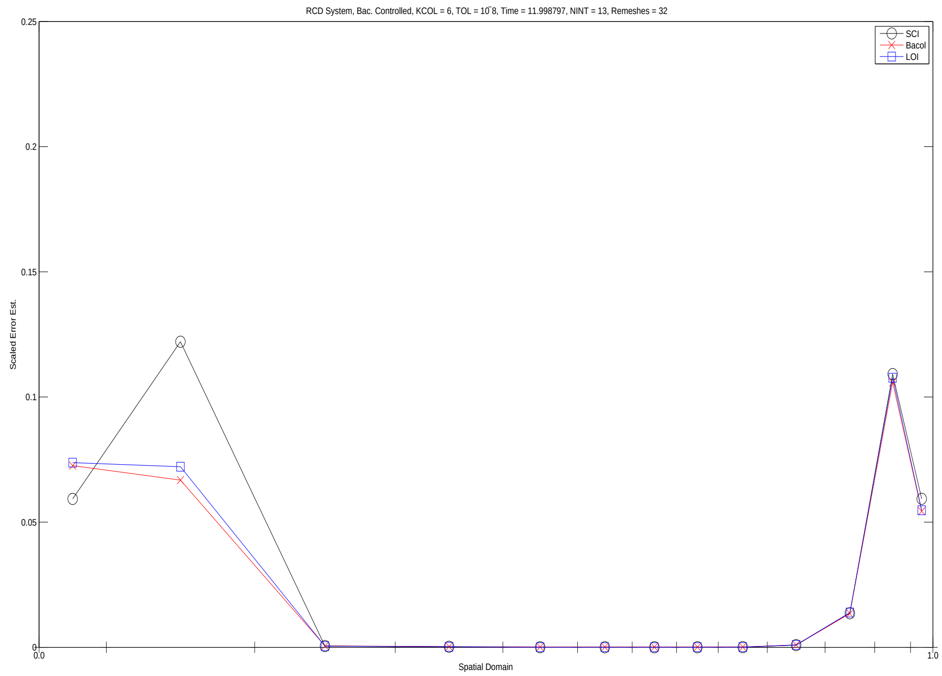


Figure 109: *When the BACOL estimates control the mesh, the SCI and LOI estimates are smaller than the BACOL estimates over the last third of the spatial domain. When the SCI estimates control the mesh, all schemes are in good agreement over the entire spatial domain.*



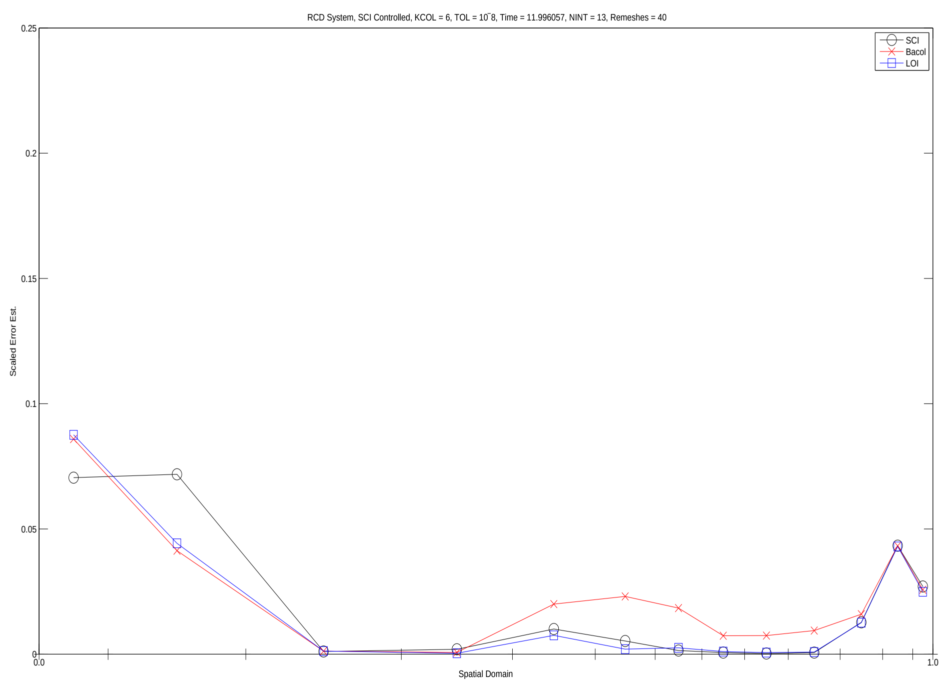
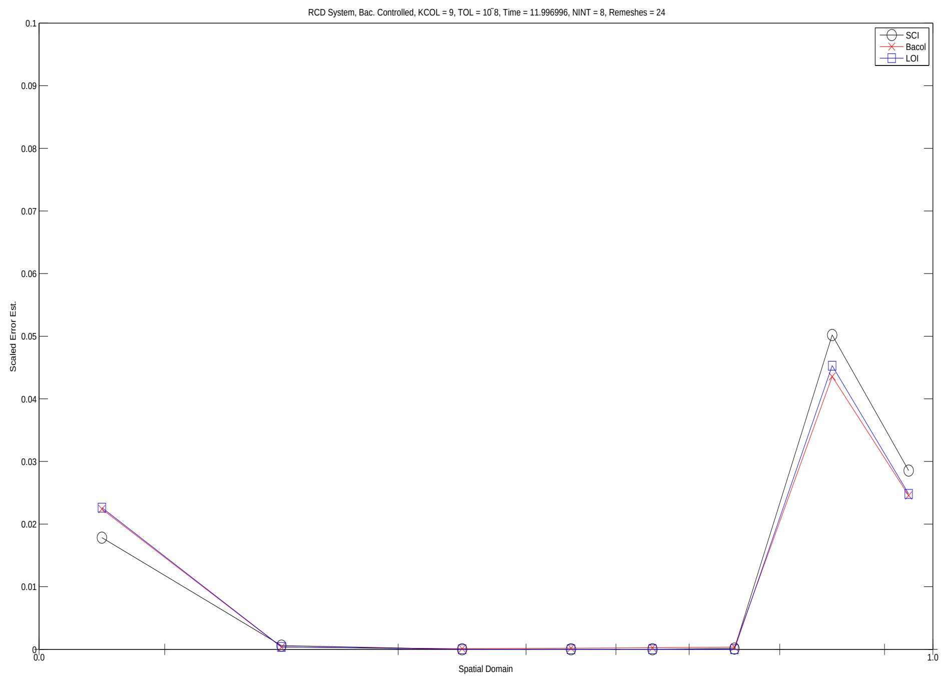


Figure 110: *When the BACOL estimates control the mesh, estimates are in good agreement except for the second subinterval where the SCI estimate is larger. When the SCI estimates control the mesh, there is reasonable agreement among all three schemes over the entire spatial domain.*



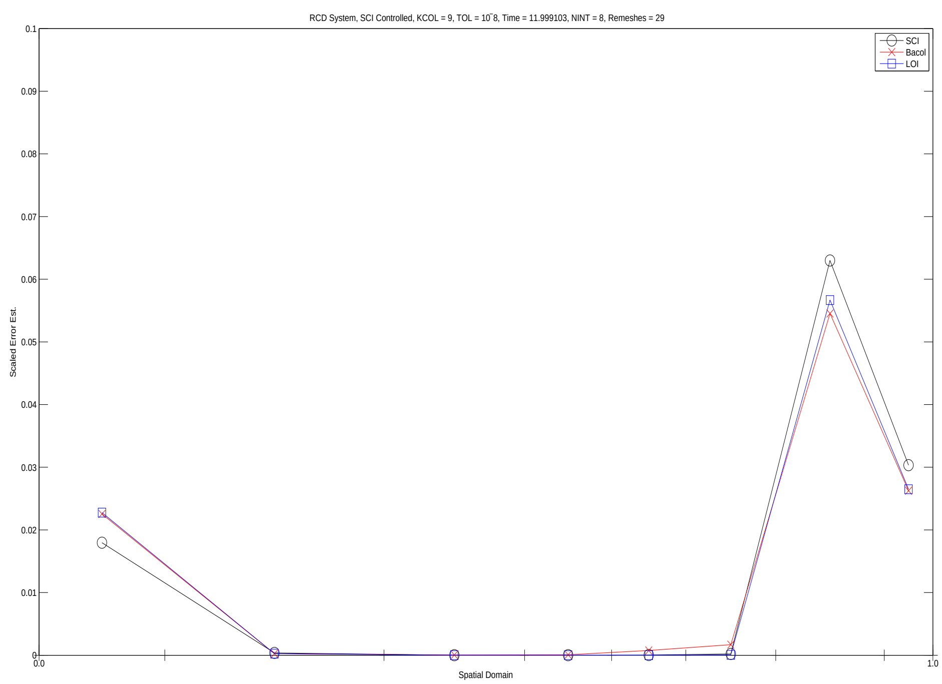


Figure 111: *Generally good agreement among all schemes when either the BACOL or SCI estimates control the mesh.*

9.6.5 Summary of Results

TOL \ KCOL		3	6	9
		NINT, Remeshes	NINT, Remeshes	NINT, Remeshes
10^{-4}	BAC	14, 53	9, 29	8, 21
	SCI	11, 43	7, 24	5, 7
10^{-6}	BAC	20, 43	9, 16	6, 13
	SCI	18, 44	8, 21	6, 15
10^{-8}	BAC	52, 69	13, 32	8, 24
	SCI	53, 97	13, 40	8, 29

Table 11: *Results are generally comparable between the two mesh control options, BACOL and SCI, with a few exceptions: for $KCOL = 3, TOL = 10^{-8}$ the SCI scheme uses more remeshings; for $KCOL = 9, TOL = 10^{-4}$ the BACOL scheme uses more remeshings.*