Error Control B-spline Gaussian Collocation/Runge-Kutta PDE Software with Interpolation-based Spatial Error Estimation *

Jack Pew[†], Tom Murtha [‡], Connor Tannahill[§], Paul Muir[¶]

Abstract

This report considers a family of software packages for the *error controlled* numerical solution of systems of one-dimensional partial differential equations. B-spline Gaussian collocation is used for the spatial discretization; the time integration is performed in the two earliest members of this family, BACOL and BACOLR, using multi-step and Runge-Kutta methods, respectively. BACOLR was shown to have better performance than BACOL for problems where the multi-step methods have stability issues. The recently released third member of this family, called BACOLI, is a modification of BACOL that improves the efficiency of the spatial error estimation computation by introducing two new interpolation-based schemes. In this report, we consider the newest member of this family, BACOLRI, a modification of BACOLR, that combines the improved Runge-Kutta stability of the time integration employed by BACOLR with the improved efficiency of the interpolation-based spatial error estimation schemes employed by BACOLI. We describe the modifications undertaken to obtain BACOLRI from BACOLR and provide extensive numerical results to compare the performances of the two packages.

Subject Classification: 65L10, 65M20, 65M70
Keywords: B-Splines, Collocation, Interpolation, Error Estimation, Error Control, Partial differential equations, Efficiency, Reliability.

1 Introduction

In this report we consider B-spline [6] Gaussian collocation software that implements adaptive control of estimates of the spatial and temporal errors for

*This work was supported by the Mathematics of Information Technology and Complex Systems Network, the Natural Sciences and Engineering Research Council of Canada, and Saint Mary's University.

[†]Saint Mary's University, Halifax, NS, Canada, B3H 3C3
[‡]Saint Mary's University, Halifax, NS, Canada, B3H 3C3
[§]Saint Mary's University, Halifax, NS, Canada, B3H 3C3
[¶]Saint Mary's University, Halifax, NS, Canada, B3H 3C3, muir@smu.ca

a system of one-dimensional (1D) partial differential equations (PDEs). The software computes a numerical solution such that corresponding high quality estimates of the spatial and temporal errors satisfy a user-prescribed tolerance. An error controlled computation provides two advantages:

- The user can have reasonable confidence that the returned numerical solution has an error that is consistent with the requested tolerance.
- The user can expect that the computational costs will be consistent with the requested tolerance.

The B-spline Gaussian collocation process is used to perform the spatial discretization, leading to an approximation of the original PDE system by a larger system of time-dependent ODEs which is coupled with the boundary conditions to give a system of time-dependent Differential-Algebraic Equations (DAEs). The DAE system is solved using a high quality DAE solver that controls an estimate of the temporal error using adaptive time-stepping and possibly also adaptive method order selection, depending on which DAE solver is employed. The spatial adaptivity, through which control of the spatial error estimate is obtained, involves the adaptive refinement of a spatial mesh which partitions the spatial domain.

The problem class we consider in this report is a PDE system of size *NPDE* of 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 boundary conditions,

$$\underline{b}_L\left(t, \underline{u}(a,t), \underline{u}_x(a,t)\right) = \underline{0}, \qquad \underline{b}_R\left(t, \underline{u}(b,t), \underline{u}_x(b,t)\right) = \underline{0}, \quad t \ge t_0, \tag{2}$$

and initial conditions,

$$\underline{u}(x,t_0) = \underline{u}_0(x), \qquad a \le x \le b. \tag{3}$$

The earliest member of the error control B-spline Gaussian collocation software family we consider in this report is BACOL [17, 19], which was developed about 15 years ago. BACOL uses the DAE solver DASSL [4], which is based on a family of multi-step methods known as Backward Differentiation Formulas (BDFs) [4]. DASSL uses both adaptive time stepping and BDF order selection to control an estimate of the temporal error. BACOL has been shown, in a comparison with comparable software for 1D PDEs, to provide superior performance, especially for problems with solutions exhibiting sharp moving layers and for sharp tolerances [18]. The second member of this family, developed about 10 years ago, is BACOLR; this package, a modification of BACOL, replaces DASSL with the DAE solver RADAU5 [10], which is based on a 5th order implicit Runge-Kutta method of Radau IIA type [10]. In [16], numerical comparisons of BACOL and BACOLR show that the two codes perform similarly on standard test problems and that BACOLR has much superior performance on problems for which the stability of the higher order BDFs is an issue. Such problems are characterized as those which lead to DAE systems which have Jacobians with eigenvalues near the imaginary axis, such as, Schrödinger type problems.

The B-spline Gaussian collocation algorithm assumes that the numerical solution is expressed in terms of a C^1-continuous B-spline basis of degree p on a mesh that partitions $[a, b]$. In both BACOL and BACOLR, the spatial error estimate is obtained by computing a second numerical solution which is based on B-splines of degree $p+1$. The computation of the second numerical solution essentially doubles the overall execution time.

This issue has recently been addressed in the modified version of BACOL known as BACOLI [14], released about two years ago. In BACOLI, the computation of the degree $p+1$ numerical solution is replaced with the efficient computation of a special type of interpolant to the degree p numerical solution.BACOLI provides two new interpolation-based spatial error estimation schemes, each of which employs a different spatial error control mode. We review details of the BACOL, BACOLR, and BACOLI packages in the next section of this report.

In this report, we introduce the newest member of this software family, BACOLRI, which is a modification of BACOLR to remove the computation of the degree $p+1$ numerical solution and instead employ the new interpolation-based spatial error estimation schemes that were introduced in BACOLI. BACOLRI therefore has both the improved performance for Schrödinger type problems that was introduced in BACOLR and the improved spatial error estimation schemes that were introduced in BACOLI.

The following diagram shows the relationship among the four packages.

BACOL	$\Rightarrow$	*Improved Temporal Stability*	$\Rightarrow$	**BACOLR**
$\Downarrow$				$\Downarrow$
Improved Spatial Error Estimate				*Improved Spatial Error Estimate*
$\Downarrow$				$\Downarrow$
BACOLI	$\Rightarrow$	*Improved Temporal Stability*	$\Rightarrow$	**BACOLRI**

This report is organized as follows. In Section 2, we provide a review of the details of the algorithms implemented in the BACOL, BACOLR, and BACOLI. In Section 3, we review the software development efforts associated with these packages and then describe the modification process that was used to develop BACOLRI from BACOLR. We also describe the differences between the interfaces for BACOLR and BACOLRI. This section also describes the new Fortran 95 wrapper we have developed for BACOLRI that significantly improves the ease-of-use of the new package. In Section 4, we provide numerical results from an extensive set of tests performed to compare BACOLR with BACOLRI. We close, in Section 5, with a summary, our conclusions, and suggestions for future work.

2 Review of BACOL, BACOLR, and BACOLI

BACOL, BACOLR, and BACOLI assume a spatial mesh, $\{x_i\}_{i=0}^{NINT}$, which partitions $[a,b]$. $NINT$ is the number of spatial subintervals defined by this mesh. Based on this mesh, and for a given degree p, the numerical solution is expressed as a linear combination of C^1-continuous, degree p, piecewise polynomials, represented in terms of a B-spline basis. The numerical solution, $\underline{U}(x,t)$, has the form,

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

where $\underline{y}_{p,i}(t)$ is the (unknown) time dependent (vector) coefficient of the i-th B-spline basis function, $B_{p,i}(x)$, and $NC_p = NINT(p-1)+2$. The Gaussian collocation spatial discretization process involves requiring that $\underline{U}(x,t)$ satisfy (1) at $p-1$ *collocation* points within each spatial mesh subinterval; these points are the images of the order $p-1$ Gauss points, $\{\rho_j\}_{j=1}^{p-1}$, on $[0,1]$. The collocation points, $\xi_l, l = 2,\ldots,NC_p-1$, are given by,

$$\begin{aligned} \xi_l &= x_{i-1} + h_i\rho_j, \quad \text{where } l = 1+(i-1)(p-1)+j, \\ &\quad \text{for } i = 1,\ldots,NINT, \quad j = 1,\ldots,p-1, \end{aligned} \tag{5}$$

where $h_i = x_{i-1} - x_i$. The corresponding collocation conditions are

$$\underline{U}_t(\xi_l,t) = \underline{f}\left(t,\xi_l,\underline{U}(\xi_l,t),\underline{U}_x(\xi_l,t),\underline{U}_{xx}(\xi_l,t)\right), \quad l = 1+(i-1)(p-1)+j, \tag{6}$$

where $i = 1,\ldots,NINT, j = 1,\ldots,p-1$. The numerical solution, $\underline{U}(x,t)$, is also required to also satisfy the boundary conditions at the points, $\xi_1 = a$ and $\xi_{NC_p} = b$; this gives the conditions,

$$\underline{b}_L\left(t,\underline{U}(a,t),\underline{U}_x(a,t)\right) = \underline{0}, \qquad \underline{b}_R\left(t,\underline{U}(b,t),\underline{U}_x(b,t)\right) = \underline{0}. \tag{7}$$

The B-spline coefficients, $\underline{y}_{p,i}(t)$, for a given time t, are computed (using temporal error control) by solving the DAE system consisting of the coupled system of collocation conditions, (6), and boundary conditions, (7). Once these are computed at time t, the numerical solution, for any $x \in [a,b]$, can be obtained from (4). Because the codes use Gaussian collocation based on Gauss points of order $p-1$, the numerical solution has a spatial error that is $O(h^{p+1})$, where $h = \max_{i=1}^{NINT} h_i$ [8, 5].

As mentioned in the previous section, BACOL and BACOLI solve (6), (7), using DASSL while BACOLR uses RADAU5. In either case, the DAE solvers require, as a central part of their computations, the solution of linear systems, that, due to the use of B-spline collocation, have an *almost block diagonal (ABD)* structure [7]. In BACOL, BACOLR, and BACOLI, these linear systems are therefore treated using the software package, COLROW [7], which is designed to efficiently handle such systems. Within COLROW, the CRDCMP routine performs factorizations of the ABD coefficient matrices, while the CRSLVE routine performs backsolves on the factored ABD systems. In BACOLR, because of the type of implicit Runge-Kutta method employed by RADAU5, it is also necessary to solve ABD systems that involve complex numbers. BACOLR therefore also employs the complex version of COLROW known as COMPLEXCOLROW [12]; the routine CCRCMP performs factorizations of the complex ABD coefficient matrices, while the routine CCRSLV performs backsolves of the factored complex ABD systems. See [16], pages 15:8-15:9, for further details.

After each accepted time step, the BACOL, BACOLR, and BACOLI packages compute an estimate of the spatial error. If the spatial error estimate does not satisfy the tolerance, the numerical solution is rejected and a remeshing (i.e., a redistribution and possible refinement of the spatial mesh) is performed. The spatial mesh adaption algorithm is based on the principle of equidistributing the spatial error estimate. Both the location and number of mesh points can be changed during a remeshing in order to adapt to the size (with respect to the user tolerance) and distribution of the spatial error estimate over the spatial domain. See [19] for further details.

As mentioned in the previous section, BACOL and BACOLR obtain the spatial error estimates by computing a second approximate solution, $\underline{\bar{U}}(x,t)$, on the same spatial mesh, using the B-spline collocation spatial discretization algorithm described earlier, followed by the solution of a time-dependent DAE system. The only differences from the computation associated with $\underline{U}(x,t)$ are the use of B-splines of degree $p+1$ and collocation points that are the images of the order p Gauss points mapped onto to each spatial subinterval. This implies that the spatial error of $\underline{\bar{U}}(x,t)$ is $O(h^{p+2})$. A scaled difference of $\underline{U}(x,t)$ and $\underline{\bar{U}}(x,t)$ is then computed to provide a spatial error estimate for $\underline{U}(x,t)$. The computation of $\underline{\bar{U}}(x,t)$, as mentioned earlier, essentially doubles the overall cost and represents an obvious inefficiency in the computation.

More recent work has involved the goal of trying to avoid the computation of $\underline{\bar{U}}(x,t)$ and obtain a spatial error estimate in a more efficient manner. One direction of investigation [1] is based on the observation that, at certain points within the spatial domain, the spatial accuracy of $\underline{U}(x,t)$ is at least one order higher, i.e., $O(h^{p+2})$, than it is at an arbitrary point in the spatial domain; these solution values are said to be superconvergent. The points at which $\underline{U}(x,t)$ is superconvergent include the mesh points as well as certain other points (see [1]) internal to each subinterval. It is also the case that the $\underline{U}_x(x,t)$ values at the mesh points are superconvergent. Using these superconvergent $\underline{U}(x,t)$ and $\underline{U}_x(x,t)$ values, a Hermite-Birkhoff polynomial interpolant associated with each spatial mesh subinterval can be constructed. A sufficient number of higher order values are interpolated in order to ensure that the interpolation error is dominated by the spatial error of the interpolated values. The spatial error of these interpolants is therefore $O(h^{p+2})$. Over $[a,b]$, these Hermite-Birkhoff interpolants give a C^1-continuous piecewise SuperConvergent Interpolant, which we call the SCI. Further details of this scheme are given in [1]. In this approach, the computation of $\underline{\bar{U}}(x,t)$ is replaced by the construction of the SCI and the latter then replaces $\underline{\bar{U}}(x,t)$ in the computation of the spatial error estimate for $\underline{U}(x,t)$.

As explained earlier, the spatial error estimation and control scheme implemented in BACOL and BACOLR computes two numerical solutions, $\underline{U}(x,t)$, of order $p+1$, and $\bar{\underline{U}}(x,t)$, of order $p+2$. The higher order solution, $\bar{\underline{U}}(x,t)$, is computed only for use in the computation of a spatial error estimate for the lower order solution, $\underline{U}(x,t)$. The numerical solution $\underline{U}(x,t)$ is returned to the user and the spatial error estimate for $\underline{U}(x,t)$ is controlled to be less than the user tolerance and is used to drive the spatial mesh adaptation process. This is an example of what is known as *standard (ST) spatial error control*.

However, it could be argued that it might be preferable to return to the user the higher order, i.e., more accurate, numerical solution, $\bar{\underline{U}}(x,t)$. This could be done with a simple modification to BACOL or BACOLR since these packages compute both $\underline{U}(x,t)$ and $\bar{\underline{U}}(x,t)$ for every time step. However, while $\bar{\underline{U}}(x,t)$ would be returned to the user, the spatial error control would of course continue to be based on the spatial error estimate for $\underline{U}(x,t)$. That is, while the packages compute both $\underline{U}(x,t)$ and $\bar{\underline{U}}(x,t)$ and either of these could be returned to the user, the difference between the two gives a spatial error estimate only for $\underline{U}(x,t)$. To reiterate, in the case where the higher order solution $\bar{\underline{U}}(x,t)$ is returned to the user, the spatial error control is still based on the spatial error estimate for the lower order solution, $\underline{U}(x,t)$. This alternative type of error control has been used for many decades in the context of Runge-Kutta formula pairs for the numerical solution of initial value ordinary differential equations (IVODEs) - see, e.g., [9] - and is known as *local extrapolation (LE) error control*.

The point raised in the previous paragraph suggests an alternative approach to addressing the inefficient computation of the two numerical solutions that is done in BACOL and BACOLR. Rather than removing the computation of $\bar{\underline{U}}(x,t)$ and replacing it with the construction of the SCI, another possibility is to remove the computation of $\underline{U}(x,t)$ and replace it with an interpolant. The idea is to construct an interpolant whose error would be the same as that of $\underline{U}(x,t)$. Then the difference between $\bar{\underline{U}}(x,t)$ and this interpolant would provide an estimate that would be the same as is currently computed in BACOL or BACOLR. In this case, $\bar{\underline{U}}(x,t)$ would be returned to the user and the spatial error estimate and control would be the same as described in the previous paragraph, namely, LE error control.

The interpolant required for this alternative approach is again a Hermite-Birkhoff polynomial interpolant on each subinterval. However it is of a different type than that upon which the SCI is based. It interpolants $\bar{\underline{U}}(x,t)$ and $\bar{\underline{U}}_x(x,t)$ at the mesh points but the remaining interpolation points (which are internal to the subinterval) are chosen so that the interpolation error of this Hermite-Birkhoff interpolant is asymptotically equivalent to the spatial error for $\underline{U}(x,t)$. The leading order term in the spatial error for $\underline{U}(x,t)$ has a known form - see, e.g., [3] - and it is possible to construct an interpolant whose interpolation error has this same form. In this case the interpolation error dominates the spatial error associated with the $\bar{\underline{U}}(x,t)$ and $\bar{\underline{U}}_x(x,t)$ values upon which the interpolant is based. Over $[a,b]$, these Hermite-Birkhoff interpolants give a C^1-continuous interpolant which is has an error that is $O(h^{p+1})$, one order lower than that of $\bar{\underline{U}}(x,t)$. This interpolant is referred to as the Lower Order Interpolant (LOI). See [2] for further details. A scaled difference of $\bar{\underline{U}}(x,t)$ and the LOI then gives the spatial error estimate. Since $\underline{\bar{U}}(x,t)$ is returned to the user but the spatial error control is based on a spatial error estimate that is for a numerical solution that is of one lower spatial order, this is an example of LE spatial error control.

Although in the above we have described the SCI as being associated with the case where $\underline{U}(x,t)$ is the returned solution and the LOI being associated with the case where $\underline{\bar{U}}(x,t)$ is the returned solution, in fact the situation is somewhat simpler. When BACOLI is called with a given input value for p, it computes and returns a numerical solution based on B-splines of degree p. If the ST spatial error control mode is chosen, then BACOLI constructs the SCI based on the degree p numerical solution and uses it to generate the spatial error estimate. If the LE spatial error control mode is chosen, then BACOLI constructs the LOI based on the degree p numerical solution and uses it to generate the spatial error estimate. Thus the availability of the two types of interpolants provides an option for two modes of spatial error control, ST mode or LE mode, similar to what is available when a Runge-Kutta formula pair is used to provide error control for an IVODE.

The report [15] provides extensive numerical results comparing BACOL, in ST and LE spatial error control modes, with BACOLI, in ST and LE spatial error control modes. The packages are tested on a standard set of test problems over a range of tolerances and p values. The results show that, generally, BACOLI is approximately twice as fast as BACOL.

3 Development of BACOLRI and BACOLRI95

In this section we first provide an overview of the previous software development projects that have led to the BACOL, BACOLR, and BACOLI packages. We then discuss the details of the software development process associated with the modification of the Fortran 77 package BACOLR to obtain the corresponding Fortran 77 package, BACOLRI. This is followed by a description of the development of the Fortran 95 wrapper, BACOLRI95, which was obtained through a modification of the Fortran 95 wrapper for BACOLI, known as BACOLI95 [14].

3.1 Background Software Development

As mentioned earlier, the original member of the software family considered in this report is the Fortran 77 code, BACOL. The paper [17] describes the development of BACOL. This includes a detailed description of where the ABD systems arise during the solution of a DAE system by DASSL as well as a detailed explanation of the specific structure of these ABD matrices. The paper also describes the modifications that were made to DASSL. These include introducing a capability for solving ABD systems and the scaling of the algebraic equations to improve the conditioning of the ABD matrices. As well, the paper discusses the details of the implementations of the BACOL algorithms associated with the efficient calling of DASSL after a remeshing (known as a warm start) and the construction of consistent initial conditions for the first call to DASSL. (A warm start requires that BACOL save solution information for up to five previous time steps and interpolate this information from the current spatial mesh to the new spatial mesh after a remeshing. This is necessary because DASSL uses multi-step methods.) The paper [17] also describes in detail the subroutines which make up the core of the BACOL package as well as identifying subroutines from the B-spline and DASSL packages that are used within BACOL. Figure 4 of [17] provides a structure diagram to explain how the subroutines interact with each other. There is also a brief description of the subroutines that must be supplied by the user. As well, there is an on-line appendix to the paper that explains all the arguments in the calling sequence for BACOL and provides an example of how to use the package to solve a specific PDE. The original BACOL software is available at http://cs.smu.ca/∼muir/BACOLI-3_Webpage.htm.

As mentioned earlier, in order to address issues associated with the stability of the BDFs employed by DASSL for certain classes of PDEs, the Fortran 77 code, BACOLR was developed through an extensive modification of BACOL to replace DASSL with RADAU5, described in [16]. The paper [16] also includes a detailed discussion of the modifications that were be made to RADAU5; these include the introduction of scalings to improve the conditioning of the linear systems that arise and code for the treatment of the real and complex ABD systems that arise. Figure 1 of [16] gives a structure diagram for BACOLR. The paper [16] also includes a detailed description of the differences between the routines employed by BACOL and BACOLR. A significant difference between BACOL and BACOLR is that, due to the one-step nature of the implicit Runge-Kutta method employed by RADAU5, warm starts are not required and it is therefore not necessary to save and interpolate solution information from previous time steps. Also included in [16] is a detailed description of the differences in the calling sequences of BACOL and BACOLR; the authors state that an effort was made to have to BACOLR employ, as much as possible, the same user supplied subroutines and the same calling arguments as BACOL. The BACOLR software is published within the Association for Computing Machinery (ACM) Collected Algorithms, as Algorithm 874, and is available at calgo.acm.org. The posted material includes all the software that comprises BACOLR as well as sample routines that show how to use BACOLR.

As explained earlier, in order to address the efficiency issue associated with the way in which the spatial error estimate is computed in BACOL, a modification of BACOL, the Fortran 77 code, BACOLI, was developed. A detailed description of the changes that were made to BACOL in order to obtain BACOLI are described in [13]; a summary of these changes is also given in [14]. The primary changes involved introducing the SCINT and LOWINT routines that implement the SCI and LOI error estimation schemes and removing the code associated with the computation of the degree $p+1$ numerical solution. The specific software development phases associated with developing BACOLI from BACOL are described. Here we provide a structure diagram for BACOLI - see Figure 1. This diagram should be compared with that of BACOL (Figure 4 of [17]) in order to see the differences between the two codes. In [13], [14], the

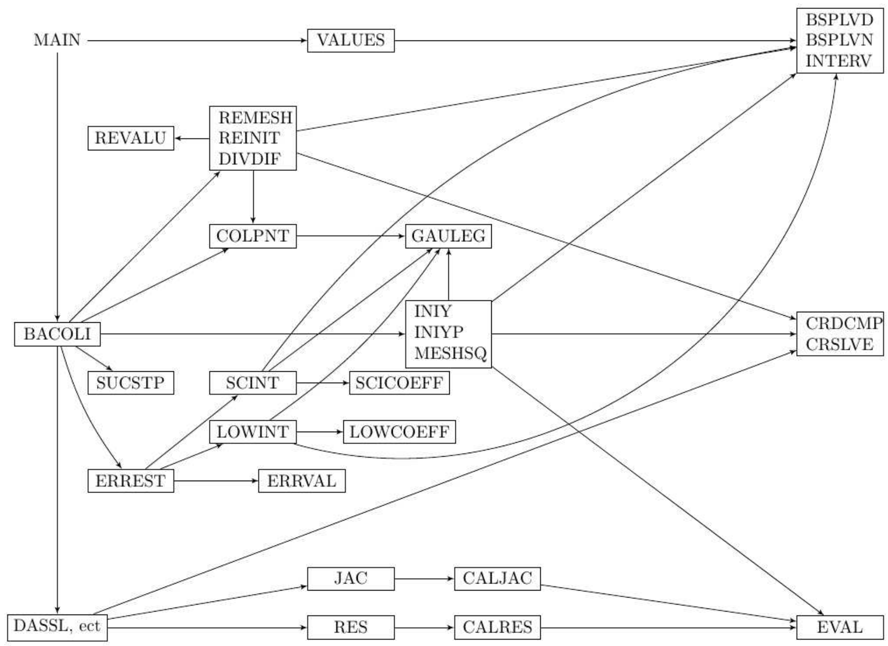

Figure 1: Structure Diagram for BACOLI.

differences between the argument lists for BACOL and BACOLI are also described. The authors indicate that the arguments list are largely the same except for several small differences. A significant addition to the software development effort for BACOLI is the introduction of a Fortran 95 wrapper for the Fortran 77 package. The paper [14] describes this wrapper, known as BACOLI95, and the simplified argument list that it provides. Both the Fortran 77 BACOLI package and the Fortran 95 BACOLI95 wrapper are available as Algorithm 962 of the ACM Collected Algorithms (calgo.acm.org). The posted material includes the BACOLI software as well as examples of the software that needs to be developed by the user in order to use BACOLI to solve a PDE. The BACOLI software is also available at http://cs.smu.ca/∼muir/BACOLI-3_Webpage.htm.

3.2 Fortran 77: BACOLR to BACOLRI

In this subsection, we detail the development process which used to construct the new Fortran 77 code, BACOLRI, from the Fortran 77 code, BACOLR. In order to obtain BACOLRI from BACOLR, a number of significant modifications were made to several fundamental components of BACOLR. Much of this work was guided by the previous work that involved modifying the Fortran 77 code, BACOL, to obtain the Fortran 77 code, BACOLI, as described briefly in the previous subsection. Several techniques and relevant subroutines developed during this process were reused in the development of BACOLRI.

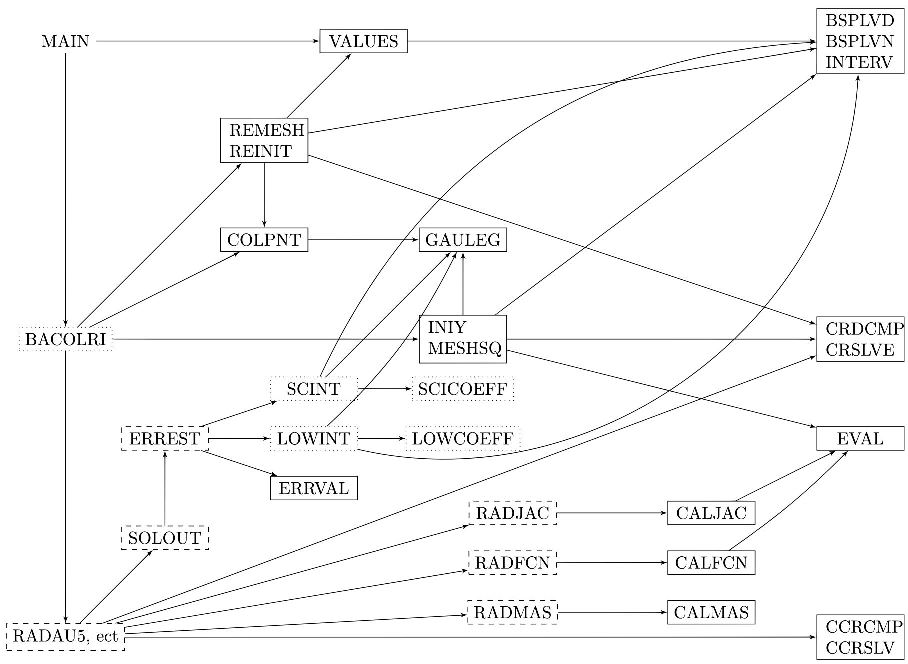

Figure 2: Structure diagram for BACOLRI. Boxes with dashed borders correspond to subroutines which have been modified in the transition from BACOLR to BACOLRI and those with dotted lines are new subroutines in BACOLRI.

In Figure 2, we provide a structure diagram for the new BACOLRI package that corresponds to Figure 1 of [16], the structure diagram for BACOLR. In Figure 2, changes to the module structure are highlighted. These include BACOLR subroutines which have been modified in the transition from BACOLR to BACOLRI as well as new subroutines that have been added to obtain BACOLRI.

The primary challenges in the modification of BACOLR to obtain BACOLRI were the introduction of the interpolation-based error estimation schemes (based on the SCI and LOI, as described in the previous section) and the removal of the code that implements the computation of the degree $p+1$ numerical solution.

The introduction of the interpolation-based error estimation schemes involved the following steps:

- New subroutines, SCINT and LOWINT, were added; these routines are used to evaluate the interpolants required by the ST and LE error control modes, respectively. These subroutines were carefully optimized for efficiency based on a rigorous performance analysis to ensure that the overall execution costs associated with these routines is substantially lower than the computation of the degree $p+1$ numerical solution. These optimization steps are described in detail in [13]. The SCINT and LOWINT routines are called within the BACOLR subroutine ERREST to obtain the spatial error estimate at each time step. This is done instead of a second call to ERRVAL, which was used to evaluate the degree $p+1$ numerical solution.

- In order to accommodate these modifications, several changes had to be made to the ERREST subroutine. The argument list was modified to remove arguments related to the evaluation of the degree $p+1$ numerical solution. The arguments x, h, *est* and *errcoef* were added. These arguments, respectively, provide the ERREST subroutine with the current spatial mesh, the current mesh subinterval size sequence, the user's choice of spatial error control mode, and the pre-computed Hermite-Birkhoff coefficients to be used to evaluate the interpolant (either the SCI or the LOI) associated with the selected spatial error control mode. The amount of working memory required by the ERREST routine was substantially increased to allow calls to the SCINT and LOWINT subroutines. In the case of the ST spatial error control mode, code has been added which scales the spatial error estimate by the mesh subinterval ratios in order to dampen over-estimation of the spatial error due to the dependence of mesh subinterval ratios in the error term of the SCI. See [13], Page 15, for further discussion on this point.

- In order for the user to be able to select which of the two new error control modes to use, an additional (fifth) entry was added to the user level control array MFLAG which enables this selection.

Once the interpolation-based error estimates had been implemented, this allowed for the removal of the computation of the degree $p+1$ numerical solution. This process involved the following steps:

- In BACOLR, following a remeshing, both the degree p and degree $p+1$ numerical solutions were reinitialized on the new mesh using the previously computed degree $p+1$ numerical solution through two calls to REINIT. For BACOLRI, we now require that only one call be made to REINIT which saves a costly call to the CRDCMP and CRSLVE routines. Additionally the routines INIY, which initializes the B-spline coefficients, and COLPNT, which computes the collocation points on the current mesh, are called only once.

- In BACOLR, each of the two DAE systems associated with the degree p and degree $p+1$ numerical solutions are passed to RADAU5 to be solved simultaneously. Without the degree $p+1$ numerical solution to consider, changes could be made to the 'NEQ', 'WORK' , 'IWORK', 'RPAR', 'IPAR' and 'CWORK' arguments of RADAU5, which leads to a substantial reduction in the total memory requirements of BACOLRI. The subroutines RADJAC and RADFCN were modified to avoid the computation of the second DAE system by removing the second call to the CALJAC and CALFCN subroutines, respectively. Of the efficiency gains obtained in the modification of BACOLR to obtain BACOLRI, the most substantial are those due to the decreased number of real and complex ABD systems that must be solved, which involves calls to CRDCMP/CRSLVE and CCRCMP/CCRSLV, respectively, from within RADAU5. In ESTRAD, the subroutine used for error estimation in RADAU5, a second call to CRSLVE was removed. In SLVRAD, duplicate calls to CRSLVE and CCRSLV were removed. In DECOMR, a duplicate call to CRDCMP was removed and in DECOMC, a duplicate call to CCRCMP was removed. In RADCOR, the code associated with scaling the boundary conditions of the combined system for both collocation solutions was modified such that the scaling is now appropriate for the single system case.

With these changes, the transition from BACOLR to BACOLRI was complete.

3.3 Fortran 95: BACOLI95 to BACOLRI95

BACOLRI95 is a user-friendly Fortran 95 module wrapping the Fortran 77 code BACOLRI. BACOLRI95 contains structured data types and subroutines which provide a vastly simplified user interface for BACOLRI compared to that of BACOLR. The benefits of this wrapper over the standard FORTRAN 77 interface include automated management of several large work arrays and the use of optional arguments (with default values) which dramatically decreases the number of arguments for a standard call to the solver. As mentioned above, BACOLRI95 is a modification of BACOLI95, the Fortran 95 wrapper developed for BACOLI. As such, BACOLRI95 is structured much the same as BACOLI95, with only minor modifications.

The primary modifications made to obtain BACOLRI95 from BACOLI95 are:

- Calls to BACOLI were replaced with calls to BACOLRI. This involved several other minor modifications such as changes to work array size calculations and to the way control flags are handled.

- Additional considerations associated with the complex work array CPAR were introduced. CPAR was added to the main structured type SOL. Additional code was added to calculate the size of CPAR based on parameter choices and to dynamically allocate this array.

- The parameters MAXORD, NSTEPS and TSTOP were removed since these are unused in BACOLRI.

4 Numerical Results

In this section, we present results from numerical experiments that show the performance of BACOLR and BACOLRI, in each of the ST and LE error control modes, applied to a collection of test problems. We will employ the following notation to identify each code/spatial error control combination:

- **(BACOLR/ST):** BACOLR in ST Spatial Error Control Mode,
- **(BACOLR/LE):** BACOLR in LE Spatial Error Control Mode,
- **(BACOLRI/ST):** BACOLRI using the SCI Spatial Error Estimation Scheme, in ST Spatial Error Control Mode,
- **(BACOLRI/LE):** BACOLRI using the LOI Spatial Error Estimation Scheme, in LE Spatial Error Control Mode.

Based on a standard set of test problems (described below), we will consider *machine independent* measures of performance and machine dependent error vs. execution time comparisons. We will also examine the performance of the spatial error estimation schemes and corresponding spatial error control modes. As well, *we will investigate the effect that the choice of p, the degree of the B-spline basis, has on the efficiency of the solvers.*

4.1 Test Problems

In this subsection we identify the ten test problems to be considered.

- **OLBE:** One Layer Burgers Equation [18]:

$$u_t = \epsilon u_{xx} - u u_x, \tag{8}$$

  with boundary conditions at $x = 0$ and $x = 1$ ($t > 0$) and an initial condition at $t_0 = 0$ ($0 \leq x \leq 1$) chosen so that the exact solution is

$$u(x,t) = \frac{1}{2} - \frac{1}{2}\tanh\left(\frac{x - \frac{t}{2} - \frac{1}{4}}{4\epsilon}\right), \tag{9}$$

  where ϵ is a problem-dependent parameter. We will consider two instances of this problem, one with $\epsilon = 10^{-3}$ and one with $\epsilon = 10^{-4}$. We solve this problem from $t_0 = 0$ to $t_{end} = 1$.

- **TLBE:** Two Layer Burgers Equation [18]: This equation employs the same PDE (8) as in the previous problem but the boundary conditions at $x = 0$ and $x = 1$ ($t > 0$) and the initial condition at $t_0 = 0$ ($0 \leq x \leq 1$) are chosen so that the exact solution is,

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

  where,

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

  where ϵ is a problem-dependent parameter. We will consider two instances of this problem involving $\epsilon = 10^{-3}$ and $\epsilon = 10^{-4}$ respectively. We solve this problem from $t_0 = 0$ to $t_{end} = 1$.

We will also consider generalizations of this problem, which we will refer to as **TLBEx6**, which is a system of PDEs consisting of 6 copies of **TLBE**, and **TLBEx12**, which is a system of PDEs consisting of 12 copies of **TLBE**.

- **CSRM:** Catalytic Surface Reaction Model [20]:

$$\begin{aligned}(u_1)_t &= -(u_1)_x + n(D_1u_3 - A_1u_1\gamma) + (u_1)_{xx}/Pe_1,\\ (u_2)_t &= -(u_2)_x + n(D_2u_4 - A_2u_2\gamma) + (u_2)_{xx}/Pe_1,\\ (u_3)_t &= A_1u_1\gamma - D_1u_3 - Ru_3u_4\gamma^2 + (u_3)_{xx}/Pe_2,\\ (u_4)_t &= A_2u_2\gamma - D_2u_4 - Ru_3u_4\gamma^2 + (u_4)_{xx}/Pe_2, \qquad (10)\end{aligned}$$

where $\gamma = 1 - u_3 - u_4$, and $n, r, Pe_1, Pe_2, D_1, D_2, R, A_1$, and A_2 are problem dependent parameters. The initial conditions at $t = 0$ $(0 \leq x \leq 1)$ 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 at $x = 0$ and $x = 1$ $(t > 0)$ 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.) Standard choices for the problem dependent parameters are $Pe_1 = Pe_2 = 10000, D_1 = 1.5, D_2 = 1.2, R = 1000, r = 0.96, n = 1$, and $A_1 = A_2 = 30$. We solve this problem from $t_0 = 0$ to $t_{end} = 18$.

- **SCHR:** Nonlinear Schrödinger System [11]:

$$\begin{aligned}(u_1)_t &= i\left(\frac{1}{2}(u_1)_{xx} + \eta\,(u_1)_x + (|u_1|^2 + \rho\,|u_2|^2)\,u_1\right),\\ (u_2)_t &= i\left(\frac{1}{2}(u_2)_{xx} - \eta\,(u_2)_x + (\rho\,|u_1|^2 + |u_2|^2)\,u_2\right),\end{aligned}$$

where $i^2 = -1$, η and ρ are positive constants. The boundary conditions are,

$$(u_1)_x(a,t) = (u_2)_x(a,t) = 0, \quad (u_1)_x(b,t) = (u_2)_x(b,t) = 0, \quad t > 0,$$

where $a \to -\infty$ and $b \to +\infty$. The initial conditions are,

$$u_1(x,0) = g_1(x), \quad u_2(x,0) = g_2(x), \quad a \leq x \leq b,$$

where $g_1(x)$ and $g_2(x)$ are chosen so that the modulus for each of the exact solutions is a soliton; in this case, the exact solutions are given by

$$u_1(x,t) = \sqrt{\frac{2\kappa}{1+\rho}} \operatorname{sech}\left(\sqrt{2\kappa}(x-\phi t)\right) e^{i\left((\phi-\eta)x - \left(\frac{\phi^2-\eta^2}{2}-\kappa\right)t\right)},$$
$$u_2(x,t) = \sqrt{\frac{2\kappa}{1+\rho}} \operatorname{sech}\left(\sqrt{2\kappa}(x-\phi t)\right) e^{i\left((\phi+\eta)x - \left(\frac{\phi^2-\eta^2}{2}-\kappa\right)t\right)},$$

where κ is a constant and ϕ represents the speed of the soliton.

In order to obtain a version of this problem that can be treated by the software we consider in this report, we set $a = -30$ and $b = 90$, where these values are chosen so that the solution values outside $[a, b]$ are negligible. In our numerical experiments we choose $\phi = 1$, $\eta = 1/2$, $\rho = 2/3$ and $\kappa = 1$. Since we have perturbed the boundary points of the original problem, the exact solutions given above will not be the exact solutions to the problem on the truncated internal $[-30, 90]$. We therefore compute a high accuracy numerical reference solution using BACOLR with a very sharp tolerance for use in some of the numerical experiments presented in this report where an error must be computed for a numerical solution.

4.2 Machine Independent Performance Measures

In this subsection, we compare BACOLR/ST, BACOLR/LE, BACOLRI/ST, and BACOLRI/LE with respect to several *machine independent* measures of the algorithms employed in the codes that can contribute significantly to overall performance. These machine independent measures provide an important complement to standard machine dependent timing results; additional insights regarding code performance can be obtained by considering such measures.

The machine independent measures we consider in this report are: the number of subintervals in the spatial mesh at the final time (**Final NINT**), the total number of accepted time steps (**Accepted Time Steps**), the total number of spatial remeshings (**Remeshings**), the total number of factorizations (**Calls to CRDCMP**) and backsolves (**Calls to CRSLVE**) of real ABD systems, and the total number of factorizations (**Calls to CCRCMP**) and backsolves (**Calls to CCRSLV**) of complex ABD systems.

We provide results for the ten test problems identified earlier: (i) **OLBE** with $\epsilon = 10^{-3}$, (ii) **OLBE** with $\epsilon = 10^{-4}$, (iii) **TLBE** with $\epsilon = 10^{-3}$, (iv) **TLBE** with $\epsilon = 10^{-4}$, (v) **TLBEx6** with $\epsilon = 10^{-3}$, (vi) **TLBEx6** with $\epsilon = 10^{-4}$, (vii) **TLBEx12** with $\epsilon = 10^{-3}$, (viii) **TLBEx12** with $\epsilon = 10^{-4}$, (ix) **CSRM**, and (x) **SCHR**.

Tables 2-11 give machine independent performance measures for the four codes, for $p = 4, 5, 7, 9$, and $tol = 10^{-4}$, 10^{-6}, 10^{-8}. Each table entry consists of two rows; the first row gives **Final NINT, Accepted Time Steps**, and **Remeshings**; the second row gives [**Calls to CRDCMP, Calls to CRSLVE**] and {**Calls to CCRCMP, Calls to CCRSLV**}. (We note that BACOLRI/ST fails on **TLBE** with $\epsilon = 10^{-3}$ when $p = 4$ and $tol = 10^{-6}$; in this case the corresponding table entries are blank.)

From these tables can make several observations. We see that for smaller p values and sharper tolerances, the LE codes have larger **Final NINT** than do the ST codes; otherwise the **Final NINT** values are similar. For a given tolerance value, all codes use about the same number of **Accepted Time Steps**, independent of p. For all codes, the number of **Remeshings** is much smaller than the total number of **Accepted Time Steps**.

In order to assist with the presentation of the results from Tables 2-11, we next present figures that provide visualizations of some of the tabular data:

- In Figures 3-32, we plot **Final NINT** vs. tol for a range of p values for three of the codes: BACOLR/ST for $p = 3, ..., 11$, BACOLRI/ST for $p = 4, ..., 11$, and BACOLRI/LE for $p = 4, ..., 11$. We consider $tol = 10^{-2}, 10^{-3}, \ldots, 10^{-10}$. There is one plot for each code and problem combination.

  From these plots we see that for smaller p values the **Final NINT** values grow approximately linearly (on a log-log scale) as the tol values decrease, while for larger p values the **Final NINT** values remain approximately at the lowest value for all tol values. The only exception is for the **SCHR** problem where, even for larger p values, we see approximately linear growth (on a log-log scale) as the tol values decrease.

- In Figures 33-62, we plot **Final NINT** vs. p for $tol = 10^{-2}, 10^{-3}, \ldots, 10^{-10}$ for the three codes: BACOLR/ST for $p = 3, ..., 11$, BACOLRI/ST for $p = 4, ..., 11$, and BACOLRI/LE for $p = 4, ..., 11$. There is one plot for each code and problem combination.

  From these plots we see that **Final NINT** has roughly the same small value for all p values when the tolerance is coarse. This is also true for sharper tolerances when the p value is large. However, for smaller p values, the value of **Final NINT** grows quite dramatically as the tolerance decreases.

- In Figures 63-92, we plot **Accepted Time Steps** vs. tol for a range of p values for each code and problem combination. The codes and corresponding p value ranges are: BACOLR/ST for $p = 3, ..., 11$, BACOLRI/ST for $p = 4, ..., 11$, and and BACOLRI/LE for $p = 4, ..., 11$. We consider $tol = 10^{-2}, 10^{-3}, \ldots, 10^{-10}$.

  From these plots we see that over all p values, the **Accepted Time Steps** values grow approximately linearly (on a log-log scale) as the tol values decrease. Furthermore, the results are generally independent of p. The only exception is that for some problems and for coarse tolerances, the larger p values correspond to a larger number of **Accepted Time Steps** than do the smaller p values.

- In Figures 93-119, we plot **Remeshings** vs. tol for a range of p values, for each code and problem combination, except **SCHR**. For **SCHR**, there are fewer times steps than for the other problems and thus very few remeshings, making it not worthwhile to plot **Remeshings** vs. *tol* for this case. The codes and corresponding p value ranges are: BACOLR/ST for $p = 3, ..., 11$, BACOLRI/ST for $p = 4, ..., 11$, and BACOLRI/LE for $p = 4, ..., 11$. We consider $tol = 10^{-2}, 10^{-3}, \ldots, 10^{-10}$.

For a given p value, the **Remeshings** values grow roughly linearly (on a log-log scale) as the *tol* values decrease. The **Remeshings** values are generally larger for the smaller p values.

- In Figures 120-159, we plot the number of real ABD matrix factorizations, i.e., **Calls to CRDCMP**, and the number of backsolves of real ABD systems, i.e., **Calls to CRSLVE**, vs. *tol* for each of the codes BACOLR/ST, BACOLRI/ST, and BACOLRI/LE. We consider $tol = 10^{-2}, 10^{-3}, \ldots, 10^{-10}$, and p values, 4, 5, 7, and 9. There is one plot for each p value and problem combination.

  From these plots we see that for all problems, the **Calls to CRSLVE** grow approximately linearly (on a log-log scale) as the tolerances get sharper. However, except for the **CSRM** and **SCHR** problems, the number of **Calls to CRDCMP** is largely independent of *tol*. For the **CSRM** and **SCHR** problems, the **Calls to CRDCMP** grow approximately linearly (on a log-log scale) as the tolerances get sharper. The **Calls to CRDCMP** value is about one order of magnitude smaller than the **Calls to CRSLVE** value. *The main observation is that the* **Calls to CRDCMP** *and* **Calls to CRSLVE** *values for BACOLR/ST are typically double those of BACOLRI/ST and BACOLRI/LE.* The only exception is for some problems when p is small; in such cases, the number of **Calls to CRDCMP** performed by BACOLR and BACOLRI is about the same. However, even in this case, the number of **Calls to CRSLVE** performed by BACOLR is still approximately double that of BACOLRI.

- In Figures 160-199, we plot the number of complex ABD matrix factorizations, i.e., **Calls to CCRCMP**, and the number of backsolves of complex ABD systems, i.e., **Calls to CCRSLV**, vs. *tol* for each of the codes BACOLR/ST, BACOLRI/ST, and BACOLRI/LE. We consider $tol = 10^{-2}, 10^{-3}, \ldots, 10^{-10}$ and p values, 4, 5, 7, and 9. There is one plot for each p value and problem combination.

  From these plots we again see that the **Calls to CCRSLV** grow approximately linearly (on a log-log scale) as the tolerances get sharper. And again, except for the **CSRM** and **SCHR** problems, the number of **Calls to CCRCMP** is largely independent of *tol*. For the **CSRM** and **SCHR** problems, the **Calls to CCRCMP** grow approximately linearly (on a log-log scale) as the tolerances get sharper. We again see that the **Calls to CCRCMP** value is about one order of magnitude smaller than the **Calls to CCRSLV** value. Once again, the main observation is that the **Calls to CCRCMP** and **Calls to CCRSLV** values for BACOLR/ST are typically double those of BACOLRI/ST and BACOLRI/LE. We again see that the only exception is for some problems when p is small; in such cases, the number of **Calls to CCRCMP** performed by BACOLR and BACOLRI is about the same.

4.3 Machine Dependent Timing Results

We next provide machine dependent timing results for each code applied to the ten test problems identified earlier in this report. For all problems, these tests were conducted on a system with two Intel(R) Xeon(R) CPU E5-4617 processors and 172 gigabytes of RAM. The operating system was Ubuntu 16.04.4 LTS and the Fortran compiler was GNU Fortran (Ubuntu 5.4.0-6ubuntu1 16.04.10) 5.4.0. The tests were run on a virtual machine installed on this system; the virtual machine was allowed access to 1 CPU and 65 gigabytes of RAM. Each code was run on each problem for $p = 4, \ldots, 11$ and for $tol = 10^{-4}$, 10^{-6}, 10^{-8}. The results are provided in Tables 12-21. Instances where a failure occurred correspond to blank table entries; these occurred for BACOLRI/ST on **TLBEx6**, **TLBEx6** and **TLBEx12** with $\epsilon = 10^{-3}$, $p = 4$, $tol = 10^{-6}$, BACOLRI/LE on **OLBEx6**, **OLBEx6** and **OLBEx12** with $\epsilon = 10^{-4}$, **SCHR**, $p = 4$, $tol = 10^{-10}$, BACOLR/LE on **SCHR**, $p = 4$, $tol = 10^{-10}$, and BACOLR/ST on **SCHR**, $p = 11$, $tol = 10^{-10}$.

From these tables we see that, for midrange to high p values BACOLRI/ST and BACOLRI/LE are faster - in many cases twice as fast - as the BACOLR/ST and BACOLR/LE codes. For the smallest p value ($p = 4$), the BACOLRI/LE code is sometimes as expensive as BACOLR/ST or BACOLR/LE. For a given problem, BACOLRI/ST always has the fastest time. For sharper tolerances, the use of BACOLRI/ST with a larger p value gives the fastest time. (We note that these are tolerance vs. CPU time comparisons; a more significant comparison is accuracy vs. CPU time; we discuss these types of comparisons later in this report.) For a given problem and tolerance, the fastest run by BACOLRI/ST depends on p; see Table 1.

As mentioned earlier, for sharper tolerances, larger p values lead to the fastest execution times. However, we also note that for larger systems of PDEs, slightly smaller p values correspond to faster execution times.

4.4 Error vs. Execution Time across Codes

Here we present error vs. execution time results for BACOLR/ST, BACOLR/LE, BACOLRI/ST and BACOLRI/LE, for **OLBE** and **TLBE**, with $\epsilon = 10^{-3}$ and 10^{-4}, and **SCHR**. We consider p values from 4 to 11. The results were obtained by running the codes over a range of 81 tolerance values, uniformly distributed on a log scale, from 10^{-2} to 10^{-10}.

In Figures 200-239, we plot error achieved vs. CPU time required. Each figure gives results, for all four codes, for a given problem and p value. The plots also show lines fitted to the data for each code to help clarify comparisons among the codes. We see generally that BACOLRI is substantially faster than BACOLR in either spatial error control mode.

	$tol = 10^{-4}$	$tol = 10^{-6}$	$tol = 10^{-8}$	$tol = 10^{-10}$
OLBE, $\epsilon = 10^{-3}$	4	4	6	8
OLBE, $\epsilon = 10^{-4}$	4	4	6	7
TLBE, $\epsilon = 10^{-3}$	4,5	5	6	8,9
TLBE, $\epsilon = 10^{-4}$	4	4	6	7
TLBEx6, $\epsilon = 10^{-3}$	5	5	6	7
TLBEx6, $\epsilon = 10^{-4}$	4	4	4	6
TLBEx12, $\epsilon = 10^{-3}$	5	5	5	7
TLBEx12, $\epsilon = 10^{-4}$	4	4	4	6
CSRM	4	4	5	6
SCHR	4	4,5,6	6	7

Table 1: *Value of p that corresponds to the fastest BACOLRI/ST run, for a given problem and tolerance. BACOLRI/ST always gives the fastest run for a given problem and tolerance. For sharper tolerances, larger p values lead to the fastest times.*

The comparisons among these codes can be seen more clearly if we plot errors vs. execution time data for BACOLR/LE, BACOLRI/ST, and BACOLRI/LE *relative* to that of BACOLR/ST. The plots were developed as follows. We describe this process for the BACOLR/LE data.

- We first perform a linear fit to the log of the error vs. log of time data associated with BACOLR/ST in order to obtain a continuous representation of the baseline BACOLR/ST data.
- Then, for each (error,time) ordered pair from the BACOLR/LE data set, we use the above mentioned linear fit to the BACOLR/ST data to obtain a corresponding time estimate for BACOLR/ST (i.e., an estimate of how much time BACOLR/ST would take to compute a solution with the same error as BACOLR/LE).
- We then compute the ratio of the actual BACOLR/LE time to this estimated BACOLR/ST time. This yields a set of ordered pairs of the form (error, time ratio) that we can associate with BACOLR/LE.
- Finally we fit a line to (log of error, time ratio) ordered pairs and plot this line on a semi-log scale.

This process is repeated for the BACOLRI/ST data and the BACOLRI/LE data.

In Figures 240-279, we plot error achieved vs. relative CPU time, for BACOLR/LE, BACOLRI/ST, and BACOLRI/LE relative to BACOLR/ST, for a given problem and p value. We see that, generally, BACOLRI/ST and BACOLRI/LE are substantially less expensive than BACOLR/ST and BACOLR/LE. The average costs for BACOLRI/ST and BACOLRI/LE are about

50% of the costs for BACOLR/ST and BACOLR/LE. (The only exception is for BACOLRI/LE when p is small and the tolerance is sharp - see below). We see that for larger errors, i.e., coarser tolerances, the LE error control codes are generally less expensive than the ST error control codes but as the error gets smaller, i.e., as the tolerance gets sharper, the LE error control codes generally become comparable to or sometimes more expensive than the ST error control codes. The results for small p values differ from the above general behaviour. For small p values, the LE codes are substantially less expensive than the corresponding ST codes for coarse tolerances but substantially more expensive for sharp tolerances; for small p and sharp tolerances, BACOLRI/LE is substantially more expensive than even BACOLR/ST.

4.5 Error vs. Execution Time across p Values

In this subsection, we consider error vs. execution time results for BACOLR/ST, BACOLR/LE, BACOLRI/ST, BACOLRI/LE, over a range of p values, $4, \ldots, 11$, for the **OLBE** and the **TLBE**, with $\epsilon = 10^{-3}$ and 10^{-4}, and for **SCHR**. *Because these graphs give error vs. execution time results over a range of p values, we can examine the impact that the choice of p has on performance.*

Figures 280-299, provide plots, for each problem and code, showing the performance of the codes with respect to error vs. execution time, over a range of p values.

From an examination of these figures, a general observation is that, for low accuracy all codes are generally more efficient when p is small, while for higher accuracy, a larger p value leads to a more efficient computation. For higher accuracy demands, small p values lead to substantially higher costs than do higher p values. Intermediate p values provide good performance over the entire range of errors.

5 Summary, Conclusions, and Future Work

B-spline collocation software for the numerical solution of 1D PDEs that features both spatial and temporal error control has been available for about 15 years. The earliest codes from this family are BACOL and BACOLR, which differ in how the time integration is performed; BACOL uses the DAE solver DASSL, while BACOLR uses RADAU5. The recently released code, BACOLI, a modification of BACOL, improves upon the efficiency of BACOL by employing interpolation-based schemes for the computation of spatial error estimates. As well, BACOLI implements, as options through its two error estimation schemes, a standard (ST) error control scheme as well as an alternative error control scheme known as local extrapolation (LE) error control. The newest code, BACOLRI, a modification of BACOLR, improves the efficiency of BACOLR, by introducing the interpolation-based schemes for the computation and control of spatial error estimates that were previously implemented in BACOLI.

This report describes the development of BACOLRI and presents a detailed examination of the performance of BACOLR and BACOLRI. It is shown that the new error estimation and control schemes generally give comparable performance measures, except for the number of real and complex ABD matrix factorizations and ABD system backsolves; for these measures BACOLRI/ST and BACOLRI/LE generally use approximately half as many factorizations and backsolves as do BACOLR/ST and BACOLR/LE. This leads to substantial savings in execution time. These results also show that for small p and sharp tol, the codes that run in LE error control mode generally have greater execution costs than the ST codes. Conversely, for coarser tolerances, the LE codes are generally observed to be relatively more efficient that the corresponding ST control codes.

This report also looks at how the choice of p effects performance. For coarser tolerances, the codes generally have smaller execution times when p is small. However, as the accuracy demands increase, larger p values lead to better efficiency.

There are several directions for future work. The results of this report suggest that it may be worthwhile to modify the BACOLRI code in order to have it choose p based on the tolerance requested. Also a modification of BACOLRI to automate the choice of error control mode may be worthwhile since it appears that LE error control is better for coarser tolerances while ST error control is better for sharper tolerances.

References

[1] T. Arsenault, T. Smith, and P.H. Muir. Superconvergent interpolants for efficient spatial error estimation in 1D PDE collocation solvers. *Can. Appl. Math. Q.*, 17:409–431, 2009.

[2] T. Arsenault, T. Smith, P.H. Muir, and J. Pew. Asymptotically correct interpolation-based spatial error estimation for 1D PDE solvers. *Can. Appl. Math. Q.*, 20:307–328, 2012.

[3] U.M. Ascher, R.M.M. Mattheij, and R.D. Russell. *Numerical Solution of Boundary Value Problems for Ordinary Dfferential Euations*, volume 13 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1995.

[4] K.E. Brenan, 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.

[5] 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.

[6] Carl de Boor. *A Practical Guide to Splines*, volume 27 of *Applied Mathematical Sciences*. Springer-Verlag, New York, revised edition, 2001.

[7] 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.

[8] 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.

[9] 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.

[10] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations. II*, volume 14 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 1996.

[11] M. S. Ismail and Thiab R. Taha. Numerical simulation of coupled nonlinear Schrödinger equation. *Math. Comput. Simulation*, 56(6):547–562, 2001.

[12] P. Keast. A complex version of the colrow package. *Unpublished software*, 1992.

[13] J. Pew, Z. Li, and P.H. Muir. A computational study of the efficiency of collocation software for 1D parabolic PDEs with interpolation-based spatial error estimation. *Saint Mary's University, Dept. of Mathematics and Computing Science Report Series, Technical Report 2013_001, http://cs.smu.ca/tech_reports*, 2013.

[14] J. Pew, Z. Li, and P.H. Muir. Algorithm 962: BACOLI: B-spline adaptive collocation software for PDEs with interpolation-based spatial error control. *ACM Trans. Math. Softw.*, 42(3):25:1–25:17, 2016.

[15] J. Pew, C. Tannahill, and P.H. Muir. Performance analysis results for error control B-spline Gaussian collocation PDE solvers. *Saint Mary's University, Dept. of Mathematics and Computing Science Technical Report Series, Technical Report 2018_001, http://cs.smu.ca/tech_reports*, 2018.

[16] 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. Softw.*, 34(3):15:1–15:28, 2008.

[17] R. Wang, P. Keast, and P.H. Muir. BACOL: B-spline Adaptive COLlocation software for 1D parabolic PDEs. *ACM Trans. Math. Software*, 30(4):454–470, 2004.

[18] 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.

[19] 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.

[20] 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.

tol	10^{-4}	10^{-6}	10^{-8}
p	4		
BACOLR/ST	15, 257, 70 [706, 3164] {546, 2342}	24, 603, 97 [648, 5906] {430, 4278}	50, 1341, 120 [580, 12568] {308, 9360}
BACOLR/LE	17, 261, 85 [690, 3324] {496, 2428}	39, 602, 103 [676, 5916] {454, 4278}	111, 1357, 124 [560, 12640] {284, 9390}
BACOLRI/ST	15, 265, 80 [382, 1650] {295, 1207}	25, 606, 91 [386, 2928] {282, 2127}	53, 1350, 139 [320, 6479] {157, 4827}
BACOLRI/LE	20, 261, 113 [356, 1783] {237, 1290}	45, 603, 144 [387, 3215] {237, 2318}	130, 1324, 413 [847, 6946] {426, 4788}
p	5		
BACOLR/ST	15, 257, 63 [716, 3152] {570, 2356}	17, 597, 117 [766, 6212] {508, 4516}	32, 1338, 108 [504, 12568] {268, 9432}
BACOLR/LE	15, 257, 70 [706, 3164] {546, 2342}	24, 603, 97 [648, 5906] {430, 4278}	50, 1341, 120 [580, 12568] {308, 9360}
BACOLRI/ST	14, 262, 58 [360, 1555] {296, 1164}	19, 598, 113 [435, 3076] {313, 2243}	35, 1350, 141 [329, 6397] {161, 4738}
BACOLRI/LE	13, 256, 84 [375, 1651] {285, 1221}	26, 599, 113 [349, 2988] {230, 2157}	50, 1341, 148 [334, 6280] {180, 4637}
p	7		
BACOLR/ST	11, 275, 48 [698, 3222] {586, 2448}	15, 582, 64 [680, 5706] {532, 4258}	16, 1329, 144 [654, 12714] {342, 9446}
BACOLR/LE	13, 256, 54 [668, 3118] {544, 2368}	15, 591, 87 [738, 5944] {544, 4386}	23, 1329, 132 [624, 12582] {324, 9344}
BACOLRI/ST	11, 303, 41 [360, 1650] {314, 1244}	15, 588, 62 [366, 2857] {297, 2134}	19, 1341, 140 [472, 6379] {324, 4750}
BACOLRI/LE	13, 253, 67 [350, 1579] {278, 1187}	14, 593, 95 [372, 2979] {271, 2190}	21, 1327, 143 [312, 6301] {163, 4682}
p	9		
BACOLR/ST	10, 328, 40 [746, 3616] {650, 2740}	15, 574, 56 [660, 5574] {532, 4180}	12, 1312, 90 [1004, 12360] {804, 9348}
BACOLR/LE	11, 276, 46 [696, 3212] {588, 2448}	15, 576, 59 [670, 5632] {532, 4216}	15, 1329, 110 [642, 12666] {402, 9540}
BACOLRI/ST	10, 386, 36 [451, 1997] {410, 1510}	11, 615, 71 [501, 3175] {423, 2361}	15, 1300, 94 [651, 6182] {548, 4676}
BACOLRI/LE	11, 259, 55 [346, 1586] {286, 1211}	15, 578, 67 [323, 2835] {250, 2117}	15, 1333, 129 [370, 6351] {235, 4754}

Table 2: *Machine independent results for the One Layer Burgers equation with* $\epsilon = 10^{-3}$. *We consider* $p = 4, 5, 7, 9$ *and* $tol = 10^{-4}, 10^{-6}, 10^{-8}$. *Table entries are of the form Final Nint, Accepted Time Steps, Remeshings [Calls to CRDCMP, CRSLVE] {Calls to CCRCMP, CCRSLVE}.*

tol	10^{-4}	10^{-6}	10^{-8}
p	4		
BACOLR/ST	14, 2654, 767 [7614, 33102] {6056, 24394}	24, 5851, 1017 [6524, 59372] {4462, 43562}	49, 13254, 1047 [4298, 124706] {2164, 93952}
BACOLR/LE	18, 2571, 857 [7482, 32778] {5744, 24168}	36, 5909, 1039 [6288, 59098] {4166, 43060}	89, 13469, 1134 [4610, 125798] {2314, 94284}
BACOLRI/ST	17, 2613, 708 [3546, 16146] {2825, 11764}	24, 5955, 864 [3964, 29440] {3058, 21609}	48, 13823, 1956 [4555, 69752] {2589, 52007}
BACOLRI/LE	20, 2505, 1129 [4077, 17837] {2917, 13043}	42, 5885, 1335 [3579, 31429] {2237, 22867}	115, 13128, 3792 [7605, 67684] {3804, 46963}
p	5		
BACOLR/ST	14, 2718, 562 [7000, 31726] {5852, 23772}	17, 5940, 1145 [8760, 62046] {6446, 45550}	26, 13193, 1187 [4868, 124106] {2446, 92902}
BACOLR/LE	14, 2654, 767 [7614, 33102] {6056, 24394}	24, 5851, 1017 [6524, 59372] {4462, 43562}	49, 13254, 1047 [4298, 124706] {2164, 93952}
BACOLRI/ST	15, 2807, 660 [3584, 16894] {2912, 12346}	19, 5859, 898 [4324, 30089] {3410, 22290}	30, 13216, 1106 [5266, 63226] {4120, 47758}
BACOLRI/LE	15, 2578, 993 [4263, 17835] {3263, 13145}	27, 5805, 1093 [3413, 30025] {2312, 22026}	46, 13264, 1278 [2590, 62487] {1301, 46656}
p	7		
BACOLR/ST	14, 2728, 471 [6792, 31748] {5826, 24236}	14, 5840, 720 [7992, 58998] {6528, 43966}	17, 13271, 1424 [8042, 127538] {5166, 95256}
BACOLR/LE	15, 2966, 468 [6890, 33182] {5930, 24956}	13, 5901, 1083 [9690, 62506] {7500, 46116}	21, 13087, 1304 [5560, 125262] {2928, 93838}
BACOLRI/ST	15, 2924, 401 [3539, 16310] {3131, 12470}	15, 5874, 791 [4258, 30316] {3458, 22299}	21, 12810, 1304 [7558, 63265] {6245, 47695}
BACOLRI/LE	15, 2616, 572 [3651, 16005] {3072, 12214}	14, 5825, 1249 [4999, 31805] {3743, 23300}	27, 13028, 1368 [3022, 63026] {1646, 47254}
p	9		
BACOLR/ST	13, 4252, 327 [7872, 41192] {7198, 30630}	15, 5962, 543 [8426, 59686] {7316, 44886}	14, 12805, 954 [11684, 123034] {9752, 92980}
BACOLR/LE	14, 2771, 449 [6896, 32258] {5974, 24752}	15, 5776, 559 [7418, 57378] {6276, 43170}	15, 12945, 1454 [14578, 127238] {11646, 95450}
BACOLRI/ST	13, 4839, 292 [5058, 22539] {4759, 16592}	15, 6229, 475 [4702, 30955] {4219, 23298}	15, 12883, 1459 [6537, 64704] {5066, 47902}
BACOLRI/LE	13, 3568, 448 [4382, 19211] {3927, 14439}	15, 5762, 615 [3922, 28993] {3300, 21722}	15, 12861, 1701 [7378, 64673] {5670, 48313}

Table 3: *Machine independent results for the One Layer Burgers equation with $\epsilon = 10^{-4}$. We consider $p = 4, 5, 7, 9$ and $tol = 10^{-4}, 10^{-6}, 10^{-8}$. Table entries are of the form Final Nint, Accepted Time Steps, Remeshings [Calls to CRDCMP, CRSLVE] {Calls to CCRCMP, CCRSLVE}.*

tol	10^{-4}	10^{-6}	10^{-8}
p		4	
BACOLR/ST	14, 200, 57 [472, 2358] {338, 1702}	21, 446, 83 [570, 4548] {384, 3296}	53, 974, 85 [556, 8752] {374, 6448}
BACOLR/LE	17, 197, 69 [468, 2458] {310, 1760}	40, 445, 83 [572, 4518] {394, 3280}	113, 990, 86 [510, 8846] {318, 6494}
BACOLRI/ST	14, 200, 88 [320, 1301] {215, 905}	— —	52, 968, 148 [454, 4641] {253, 3324}
BACOLRI/LE	18, 196, 81 [252, 1255] {166, 892}	44, 446, 107 [309, 2381] {198, 1717}	133, 978, 363 [883, 5209] {513, 3498}
p		5	
BACOLR/ST	15, 203, 48 [440, 2302] {328, 1682}	15, 447, 88 [544, 4550] {348, 3276}	29, 970, 93 [560, 8694] {358, 6360}
BACOLR/LE	14, 200, 57 [472, 2358] {338, 1702}	21, 446, 83 [570, 4548] {384, 3296}	53, 974, 85 [556, 8752] {374, 6448}
BACOLRI/ST	14, 206, 50 [257, 1148] {201, 835}	19, 443, 72 [258, 2158] {179, 1564}	33, 964, 89 [275, 4364] {179, 3215}
BACOLRI/LE	14, 197, 64 [248, 1194] {179, 864}	23, 444, 83 [277, 2246] {189, 1631}	49, 971, 105 [299, 4400] {189, 3214}
p		7	
BACOLR/ST	14, 219, 40 [500, 2356] {404, 1734}	14, 452, 66 [506, 4414] {350, 3212}	15, 966, 103 [574, 8756] {348, 6384}
BACOLR/LE	15, 208, 41 [446, 2286] {348, 1684}	15, 448, 69 [500, 4378] {342, 3178}	21, 964, 93 [584, 8720] {378, 6392}
BACOLRI/ST	14, 206, 36 [233, 1116] {192, 832}	12, 452, 68 [302, 2218] {228, 1620}	18, 977, 107 [305, 4472] {192, 3275}
BACOLRI/LE	15, 200, 49 [226, 1136] {172, 833}	15, 445, 78 [256, 2193] {172, 1586}	19, 964, 102 [283, 4388] {175, 3214}
p		9	
BACOLR/ST	13, 235, 32 [482, 2424] {406, 1804}	13, 461, 54 [604, 4448] {480, 3284}	14, 1001, 80 [490, 9086] {306, 6730}
BACOLR/LE	11, 248, 38 [532, 2582] {440, 1908}	14, 456, 57 [512, 4388] {382, 3226}	14, 978, 94 [532, 8912] {320, 6546}
BACOLRI/ST	15, 198, 34 [220, 1098] {182, 828}	15, 451, 47 [269, 2138] {217, 1588}	15, 982, 76 [339, 4487] {258, 3346}
BACOLRI/LE	12, 231, 43 [270, 1269] {222, 940}	15, 446, 60 [260, 2143] {195, 1571}	14, 987, 104 [279, 4488] {168, 3286}

Table 4: *Machine independent results for the Two Layer Burgers equation with* $\epsilon = 10^{-3}$. *We consider* $p = 4, 5, 7, 9$ *and* $tol = 10^{-4}, 10^{-6}, 10^{-8}$. *Table entries are of the form Final Nint, Accepted Time Steps, Remeshings [Calls to CRDCMP, CRSLVE] {Calls to CCRCMP, CCRSLVE}.*

tol	10^{-4}	10^{-6}	10^{-8}
p	4		
BACOLR/ST	14, 2003, 631 [5354, 23840] {4032, 17154}	22, 4361, 812 [4594, 43872] {2946, 31868}	47, 9628, 844 [3610, 86996] {1894, 64324}
BACOLR/LE	18, 1936, 758 [5148, 24536] {3608, 17598}	35, 4415, 927 [5286, 45708] {3400, 33124}	94, 9831, 964 [4036, 87786] {2080, 64228}
BACOLRI/ST	15, 1965, 649 [2620, 11946] {1942, 8552}	23, 4543, 783 [2539, 21960] {1719, 15802}	49, 9910, 831 [1824, 44109] {985, 32529}
BACOLRI/LE	23, 1894, 873 [2641, 12698] {1762, 9052}	39, 4397, 1095 [2701, 23677] {1596, 17080}	100, 9690, 2891 [5909, 48923] {3009, 33442}
p	5		
BACOLR/ST	15, 2118, 503 [5216, 23598] {4162, 17200}	16, 4418, 914 [5704, 45408] {3852, 32882}	28, 9565, 940 [3990, 86486] {2086, 63562}
BACOLR/LE	14, 2003, 631 [5354, 23840] {4032, 17154}	22, 4361, 812 [4594, 43872] {2946, 31868}	47, 9628, 844 [3610, 86996] {1894, 64324}
BACOLRI/ST	15, 2122, 493 [2484, 11747] {1981, 8520}	19, 4392, 755 [2763, 21884] {1983, 15924}	32, 9686, 858 [2601, 43661] {1687, 32203}
BACOLRI/LE	13, 1961, 721 [2822, 12346] {2095, 8911}	24, 4336, 880 [2387, 22055] {1494, 15946}	45, 9602, 988 [2113, 43809] {1115, 32221}
p	7		
BACOLR/ST	14, 2570, 424 [5386, 26448] {4514, 19384}	15, 4502, 730 [6336, 45542] {4852, 33452}	17, 9700, 1079 [5224, 88444] {3018, 64658}
BACOLR/LE	15, 2353, 436 [5114, 24828] {4218, 18176}	15, 4448, 906 [6660, 46432] {4824, 33816}	21, 9508, 990 [4174, 87094] {2170, 64084}
BACOLRI/ST	14, 2505, 376 [2789, 12854] {2404, 9472}	15, 4459, 665 [3157, 22560] {2483, 16575}	19, 9655, 965 [4090, 44347] {3115, 32727}
BACOLRI/LE	15, 2248, 527 [2905, 12693] {2366, 9301}	14, 4394, 976 [3449, 23283] {2449, 16856}	23, 9635, 1017 [2119, 44013] {1090, 32332}
p	9		
BACOLR/ST	14, 3257, 323 [5758, 30616] {5088, 22366}	15, 4805, 646 [7990, 49170] {6674, 36386}	14, 9535, 956 [9220, 89568] {7280, 66492}
BACOLR/LE	15, 2746, 387 [5468, 27516] {4670, 20196}	15, 4602, 722 [7604, 47504] {6136, 34984}	14, 9687, 1122 [7762, 89644] {5486, 65732}
BACOLRI/ST	15, 2876, 352 [3335, 14480] {2975, 10662}	15, 4843, 625 [4017, 24602] {3382, 18190}	15, 9548, 992 [4609, 45240] {3607, 33367}
BACOLRI/LE	15, 2702, 442 [3263, 14226] {2815, 10393}	14, 4563, 769 [3917, 23819] {3141, 17453}	14, 9596, 1220 [4149, 45074] {2922, 33014}

Table 5: *Machine independent results for the Two Layer Burgers equation with* $\epsilon = 10^{-4}$. *We consider* $p = 4, 5, 7, 9$ *and* $tol = 10^{-4}, 10^{-6}, 10^{-8}$. *Table entries are of the form Final Nint, Accepted Time Steps, Remeshings [Calls to CRDCMP, CRSLVE] {Calls to CCRCMP, CCRSLVE}.*

tol	10^{-4}	10^{-6}	10^{-8}
p	4		
BACOLR/ST	15, 2003, 610 [5264, 23626] {3984, 17046}	23, 4360, 770 [4468, 43368] {2904, 31534}	44, 9631, 889 [3810, 86192] {1988, 63310}
BACOLR/LE	17, 1932, 757 [5144, 24518] {3606, 17592}	33, 4407, 975 [5604, 46340] {3630, 33592}	99, 9829, 956 [4000, 87702] {2060, 64180}
BACOLRI/ST	17, 1966, 610 [2530, 11717] {1890, 8418}	26, 4504, 763 [2566, 21872] {1787, 15807}	45, 9881, 837 [1927, 43853] {1074, 32282}
BACOLRI/LE	20, 1898, 901 [2677, 12869] {1770, 9163}	38, 4377, 1094 [2715, 23596] {1611, 17021}	99, 9783, 2877 [5894, 49263] {3008, 33717}
p	5		
BACOLR/ST	15, 2144, 508 [5290, 23718] {4226, 17230}	16, 4415, 884 [5544, 45206] {3752, 32806}	29, 9570, 837 [3580, 86486] {1882, 63964}
BACOLR/LE	15, 2003, 610 [5264, 23626] {3984, 17046}	23, 4360, 770 [4468, 43368] {2904, 31534}	44, 9631, 889 [3810, 86192] {1988, 63310}
BACOLRI/ST	15, 2123, 504 [2483, 11805] {1959, 8541}	20, 4387, 720 [2704, 21624] {1962, 15756}	36, 9694, 837 [2596, 43480] {1727, 32080}
BACOLRI/LE	15, 1958, 736 [2853, 12447] {2111, 8984}	22, 4340, 884 [2375, 22108] {1484, 15993}	45, 9612, 1025 [2152, 43676] {1117, 32004}
p	7		
BACOLR/ST	15, 2479, 399 [5266, 25620] {4444, 18892}	15, 4501, 784 [6792, 46142] {5200, 33800}	16, 9679, 1100 [5248, 88458] {3000, 64630}
BACOLR/LE	15, 2322, 420 [5062, 24408] {4198, 17938}	15, 4444, 929 [6786, 46564] {4904, 33850}	21, 9551, 1011 [4266, 87196] {2220, 64016}
BACOLRI/ST	15, 2360, 423 [2836, 12648] {2404, 9328}	15, 4465, 673 [3193, 22638] {2511, 16629}	19, 9560, 951 [4238, 44026] {3277, 32530}
BACOLRI/LE	14, 2289, 534 [2954, 12903] {2407, 9453}	14, 4394, 961 [3388, 23230] {2415, 16857}	27, 9542, 1041 [2177, 43842] {1124, 32206}
p	9		
BACOLR/ST	15, 3133, 330 [5662, 29812] {4978, 21790}	13, 4887, 663 [8490, 50250] {7140, 37118}	14, 9572, 927 [8638, 89548] {6748, 66466}
BACOLR/LE	15, 2862, 364 [5538, 27940] {4786, 20432}	15, 4576, 696 [7276, 46646] {5860, 34434}	15, 9680, 1131 [7850, 89778] {5556, 65842}
BACOLRI/ST	15, 2882, 363 [3360, 14632] {2986, 10758}	15, 4848, 595 [4005, 24480] {3398, 18170}	15, 9574, 952 [4417, 45005] {3455, 33211}
BACOLRI/LE	14, 2861, 423 [3460, 14803] {3031, 10812}	15, 4509, 749 [3695, 23350] {2939, 17163}	15, 9595, 1261 [4155, 45107] {2887, 32964}

Table 6: *Machine independent results for the Two Layer Burgers equation*×6 *with* $\epsilon = 10^{-4}$. *We consider* $p = 4, 5, 7, 9$ *and* $tol = 10^{-4}, 10^{-6}, 10^{-8}$. *Table entries are of the form Final Nint, Accepted Time Steps, Remeshings [Calls to CRDCMP, CRSLVE]* {*Calls to CCRCMP, CCRSLVE*}.

tol	10^{-4}	10^{-6}	10^{-8}
p		4	
BACOLR/ST	15, 2003, 610 [5264, 23626] {3984, 17046}	23, 4360, 770 [4468, 43368] {2904, 31534}	44, 9631, 889 [3810, 86192] {1988, 63310}
BACOLR/LE	17, 1932, 757 [5144, 24518] {3606, 17592}	33, 4407, 975 [5604, 46340] {3630, 33592}	99, 9829, 956 [4000, 87702] {2060, 64180}
BACOLRI/ST	17, 1966, 610 [2530, 11717] {1890, 8418}	26, 4504, 763 [2566, 21872] {1787, 15807}	45, 9881, 837 [1927, 43853] {1074, 32282}
BACOLRI/LE	20, 1898, 901 [2677, 12869] {1770, 9163}	38, 4377, 1094 [2715, 23596] {1611, 17021}	99, 9783, 2877 [5894, 49263] {3008, 33717}
p		5	
BACOLR/ST	15, 2144, 508 [5290, 23718] {4226, 17230}	16, 4415, 884 [5544, 45206] {3752, 32806}	29, 9570, 837 [3580, 86486] {1882, 63964}
BACOLR/LE	15, 2003, 610 [5264, 23626] {3984, 17046}	23, 4360, 770 [4468, 43368] {2904, 31534}	44, 9631, 889 [3810, 86192] {1988, 63310}
BACOLRI/ST	15, 2123, 504 [2483, 11805] {1959, 8541}	20, 4387, 720 [2704, 21624] {1962, 15756}	36, 9694, 837 [2596, 43480] {1727, 32080}
BACOLRI/LE	15, 1958, 736 [2853, 12447] {2111, 8984}	22, 4340, 884 [2375, 22108] {1484, 15993}	45, 9612, 1025 [2152, 43676] {1117, 32004}
p		7	
BACOLR/ST	15, 2479, 399 [5266, 25620] {4444, 18892}	15, 4501, 784 [6792, 46142] {5200, 33800}	16, 9679, 1100 [5248, 88458] {3000, 64630}
BACOLR/LE	15, 2322, 420 [5062, 24408] {4198, 17938}	15, 4444, 929 [6786, 46564] {4904, 33850}	21, 9551, 1011 [4266, 87196] {2220, 64016}
BACOLRI/ST	15, 2360, 423 [2836, 12648] {2404, 9328}	15, 4465, 673 [3193, 22638] {2511, 16629}	19, 9560, 951 [4238, 44026] {3277, 32530}
BACOLRI/LE	14, 2289, 534 [2954, 12903] {2407, 9453}	14, 4394, 961 [3388, 23230] {2415, 16857}	27, 9542, 1041 [2177, 43842] {1124, 32206}
p		9	
BACOLR/ST	15, 3133, 330 [5662, 29812] {4978, 21790}	13, 4887, 663 [8490, 50250] {7140, 37118}	14, 9572, 927 [8638, 89548] {6748, 66466}
BACOLR/LE	15, 2862, 364 [5538, 27940] {4786, 20432}	15, 4576, 696 [7276, 46646] {5860, 34434}	15, 9680, 1131 [7850, 89778] {5556, 65842}
BACOLRI/ST	15, 2882, 363 [3360, 14632] {2986, 10758}	15, 4848, 595 [4005, 24480] {3398, 18170}	15, 9574, 952 [4417, 45005] {3455, 33211}
BACOLRI/LE	14, 2861, 423 [3460, 14803] {3031, 10812}	15, 4509, 749 [3695, 23350] {2939, 17163}	15, 9595, 1261 [4155, 45107] {2887, 32964}

Table 7: *Machine independent results for the Two Layer Burgers equation*×6 *with* $\epsilon = 10^{-4}$. *We consider* $p = 4, 5, 7, 9$ *and tol* $= 10^{-4}, 10^{-6}, 10^{-8}$. *Table entries are of the form Final Nint, Accepted Time Steps, Remeshings [Calls to CRDCMP, CRSLVE] {Calls to CCRCMP, CCRSLVE}.*

tol	10^{-4}	10^{-6}	10^{-8}
p	4		
BACOLR/ST	15, 2002, 593 [5108, 23482] {3866, 16970}	23, 4364, 795 [4538, 43640] {2924, 31698}	47, 9619, 833 [3544, 86876] {1850, 64266}
BACOLR/LE	19, 1934, 728 [5022, 24298] {3542, 17484}	35, 4409, 921 [5296, 45656] {3430, 33120}	96, 9825, 968 [4044, 87788] {2080, 64226}
BACOLRI/ST	15, 1960, 609 [2507, 11705] {1882, 8415}	25, 4466, 725 [2547, 21719] {1786, 15757}	50, 9883, 811 [1740, 44247] {921, 32734}
BACOLRI/LE	20, 1891, 882 [2634, 12726] {1746, 9065}	41, 4398, 1097 [2728, 23669] {1621, 17067}	108, 9677, 3083 [6302, 49639] {3210, 33787}
p	5		
BACOLR/ST	15, 2121, 500 [5238, 23520] {4190, 17138}	17, 4424, 870 [5458, 45010] {3694, 32648}	27, 9572, 974 [4126, 86898] {2154, 63824}
BACOLR/LE	15, 2002, 593 [5108, 23482] {3866, 16970}	23, 4364, 795 [4538, 43640] {2924, 31698}	47, 9619, 833 [3544, 86876] {1850, 64266}
BACOLRI/ST	14, 2118, 484 [2445, 11705] {1951, 8489}	20, 4387, 698 [2690, 21573] {1978, 15746}	31, 9729, 852 [2537, 43618] {1656, 32156}
BACOLRI/LE	18, 1958, 741 [2869, 12423] {2122, 8957}	25, 4330, 902 [2422, 22159] {1507, 16012}	48, 9616, 1007 [2112, 43944] {1095, 32304}
p	7		
BACOLR/ST	15, 2554, 422 [5306, 26278] {4438, 19258}	15, 4485, 743 [6402, 45566] {4892, 33452}	16, 9662, 1096 [5396, 88300] {3180, 64558}
BACOLR/LE	12, 2316, 456 [5236, 24734] {4300, 18112}	15, 4450, 856 [6372, 45784] {4636, 33370}	20, 9501, 982 [4154, 86962] {2166, 63998}
BACOLRI/ST	14, 2506, 388 [2856, 12891] {2459, 9476}	15, 4477, 696 [3256, 22809] {2550, 16721}	18, 9612, 994 [4219, 44316] {3216, 32674}
BACOLRI/LE	14, 2318, 539 [2946, 13045] {2394, 9516}	15, 4384, 1021 [3525, 23461] {2475, 16941}	23, 9486, 1051 [2184, 43851] {1121, 32251}
p	9		
BACOLR/ST	15, 2925, 341 [5416, 28356] {4710, 20856}	15, 4779, 633 [7852, 48578] {6562, 35968}	15, 9567, 980 [9060, 89906] {7052, 66604}
BACOLR/LE	14, 2901, 364 [5590, 28282] {4838, 20664}	15, 4595, 688 [7370, 46950] {5970, 34644}	15, 9702, 1077 [7582, 89306] {5396, 65546}
BACOLRI/ST	14, 3089, 342 [3531, 15218] {3178, 11177}	15, 4963, 642 [4252, 25212] {3602, 18605}	15, 9542, 1024 [4719, 45275] {3686, 33347}
BACOLRI/LE	13, 3099, 411 [3722, 15604] {3305, 11331}	15, 4524, 761 [3797, 23519] {3029, 17239}	17, 9623, 1196 [4020, 45042] {2817, 32999}

Table 8: *Machine independent results for the Two Layer Burgers equation*×12 *with* $\epsilon = 10^{-4}$. *We consider* $p = 4, 5, 7, 9$ *and tol* $= 10^{-4}, 10^{-6}, 10^{-8}$. *Table entries are of the form Final Nint, Accepted Time Steps, Remeshings [Calls to CRDCMP, CRSLVE] {Calls to CCRCMP, CCRSLVE}.*

tol	10^{-4}	10^{-6}	10^{-8}
p	4		
BACOLR/ST	15, 2002, 593 [5108, 23482] {3866, 16970}	23, 4364, 795 [4538, 43640] {2924, 31698}	47, 9619, 833 [3544, 86876] {1850, 64266}
BACOLR/LE	19, 1934, 728 [5022, 24298] {3542, 17484}	35, 4409, 921 [5296, 45656] {3430, 33120}	96, 9825, 968 [4044, 87788] {2080, 64226}
BACOLRI/ST	15, 1960, 609 [2507, 11705] {1882, 8415}	25, 4466, 725 [2547, 21719] {1786, 15757}	50, 9883, 811 [1740, 44247] {921, 32734}
BACOLRI/LE	20, 1891, 882 [2634, 12726] {1746, 9065}	41, 4398, 1097 [2728, 23669] {1621, 17067}	108, 9677, 3083 [6302, 49639] {3210, 33787}
p	5		
BACOLR/ST	15, 2121, 500 [5238, 23520] {4190, 17138}	17, 4424, 870 [5458, 45010] {3694, 32648}	27, 9572, 974 [4126, 86898] {2154, 63824}
BACOLR/LE	15, 2002, 593 [5108, 23482] {3866, 16970}	23, 4364, 795 [4538, 43640] {2924, 31698}	47, 9619, 833 [3544, 86876] {1850, 64266}
BACOLRI/ST	14, 2118, 484 [2445, 11705] {1951, 8489}	20, 4387, 698 [2690, 21573] {1978, 15746}	31, 9729, 852 [2537, 43618] {1656, 32156}
BACOLRI/LE	18, 1958, 741 [2869, 12423] {2122, 8957}	25, 4330, 902 [2422, 22159] {1507, 16012}	48, 9616, 1007 [2112, 43944] {1095, 32304}
p	7		
BACOLR/ST	15, 2554, 422 [5306, 26278] {4438, 19258}	15, 4485, 743 [6402, 45566] {4892, 33452}	16, 9662, 1096 [5396, 88300] {3180, 64558}
BACOLR/LE	12, 2316, 456 [5236, 24734] {4300, 18112}	15, 4450, 856 [6372, 45784] {4636, 33370}	20, 9501, 982 [4154, 86962] {2166, 63998}
BACOLRI/ST	14, 2506, 388 [2856, 12891] {2459, 9476}	15, 4477, 696 [3256, 22809] {2550, 16721}	18, 9612, 994 [4219, 44316] {3216, 32674}
BACOLRI/LE	14, 2318, 539 [2946, 13045] {2394, 9516}	15, 4384, 1021 [3525, 23461] {2475, 16941}	23, 9486, 1051 [2184, 43851] {1121, 32251}
p	9		
BACOLR/ST	15, 2925, 341 [5416, 28356] {4710, 20856}	15, 4779, 633 [7852, 48578] {6562, 35968}	15, 9567, 980 [9060, 89906] {7052, 66604}
BACOLR/LE	14, 2901, 364 [5590, 28282] {4838, 20664}	15, 4595, 688 [7370, 46950] {5970, 34644}	15, 9702, 1077 [7582, 89306] {5396, 65546}
BACOLRI/ST	14, 3089, 342 [3531, 15218] {3178, 11177}	15, 4963, 642 [4252, 25212] {3602, 18605}	15, 9542, 1024 [4719, 45275] {3686, 33347}
BACOLRI/LE	13, 3099, 411 [3722, 15604] {3305, 11331}	15, 4524, 761 [3797, 23519] {3029, 17239}	17, 9623, 1196 [4020, 45042] {2817, 32999}

Table 9: *Machine independent results for the Two Layer Burgers equation*$\times 12$ *with* $\epsilon = 10^{-4}$. *We consider* $p = 4, 5, 7, 9$ *and* $tol = 10^{-4}, 10^{-6}, 10^{-8}$. *Table entries are of the form Final Nint, Accepted Time Steps, Remeshings [Calls to CRDCMP, CRSLVE]* {*Calls to CCRCMP, CCRSLVE*}.

tol	10^{-4}	10^{-6}	10^{-8}
p	4		
BACOLR/ST	19, 923, 226 [4150, 16296] {3682, 13524}	41, 1694, 358 [6652, 29024] {5912, 24168}	86, 3068, 407 [8806, 45678] {7968, 37872}
BACOLR/LE	28, 957, 253 [4454, 16662] {3932, 13714}	98, 1705, 478 [7408, 30926] {6428, 25570}	220, 3109, 488 [9364, 47580] {8364, 39368}
BACOLRI/ST	18, 910, 162 [1679, 6870] {1510, 5626}	44, 1687, 300 [3139, 13625] {2829, 11328}	95, 3058, 422 [4386, 22729] {3952, 18815}
BACOLRI/LE	28, 926, 260 [2213, 8346] {1947, 6892}	78, 1761, 381 [3542, 15054] {3155, 12524}	237, 3613, 881 [6385, 30254] {5497, 24871}
p	5		
BACOLR/ST	14, 926, 220 [3906, 16084] {3446, 13320}	34, 1754, 394 [7134, 29926] {6322, 24798}	52, 3110, 475 [9300, 47648] {8326, 39488}
BACOLR/LE	19, 923, 226 [4150, 16296] {3682, 13524}	41, 1694, 358 [6652, 29024] {5912, 24168}	86, 3068, 407 [8806, 45678] {7968, 37872}
BACOLRI/ST	14, 957, 147 [1664, 6853] {1512, 5595}	30, 1670, 283 [3116, 13434] {2820, 11184}	49, 3028, 383 [4254, 22282] {3865, 18482}
BACOLRI/LE	15, 861, 180 [1824, 7378] {1639, 6151}	40, 1683, 343 [3324, 14219] {2975, 11844}	82, 3119, 476 [4668, 23672] {4184, 19590}
p	7		
BACOLR/ST	14, 1037, 216 [3764, 16566] {3312, 13550}	21, 1784, 337 [6848, 29264] {6150, 24288}	27, 3107, 453 [9194, 47248] {8264, 39184}
BACOLR/LE	15, 971, 231 [3872, 16540] {3390, 13628}	26, 1732, 337 [6766, 28566] {6068, 23706}	35, 3075, 439 [8942, 46478] {8040, 38534}
BACOLRI/ST	13, 1085, 121 [1667, 7185] {1539, 5833}	21, 1757, 331 [3302, 14354] {2961, 11919}	26, 3091, 414 [4430, 23044] {4009, 19118}
BACOLRI/LE	15, 995, 188 [1877, 7942] {1683, 6560}	24, 1715, 341 [3330, 14280] {2982, 11868}	35, 3040, 426 [4380, 22820] {3947, 18921}
p	9		
BACOLR/ST	12, 1284, 193 [3948, 17876] {3542, 14216}	16, 1918, 425 [7196, 31846] {6322, 26014}	20, 3230, 480 [9916, 48586] {8932, 40150}
BACOLR/LE	15, 1145, 208 [3832, 17094] {3396, 13748}	17, 1869, 380 [7190, 31024] {6406, 25672}	22, 3148, 461 [9304, 47664] {8358, 39478}
BACOLRI/ST	15, 1297, 98 [1807, 7948] {1703, 6399}	15, 1914, 307 [3285, 14660] {2971, 12044}	18, 3176, 444 [4728, 23583] {4277, 19511}
BACOLRI/LE	15, 1209, 149 [1888, 8337] {1733, 6752}	18, 1880, 460 [3762, 16139] {3295, 13287}	22, 3130, 491 [4645, 23894] {4146, 19766}

Table 10: *Machine independent results for the Catalytic Surface Reaction Model. We consider* $p = 4, 5, 7, 9$ *and* $tol = 10^{-4}, 10^{-6}, 10^{-8}$. *Table entries are of the form Final Nint, Accepted Time Steps, Remeshings [Calls to CRDCMP, CRSLVE] {Calls to CCRCMP, CCRSLVE}.*

tol	10^{-4}	10^{-6}	10^{-8}
p		4	
BACOLR/ST	31, 9, 4 [62, 136] {34, 74}	76, 15, 2 [60, 138] {40, 78}	166, 24, 0 [82, 186] {58, 104}
BACOLR/LE	49, 9, 1 [44, 94] {26, 50}	155, 14, 1 [70, 150] {40, 78}	497, 24, 1 [92, 204] {62, 112}
BACOLRI/ST	30, 9, 6 [29, 57] {15, 28}	71, 14, 6 [32, 71] {20, 39}	179, 26, 8 [43, 122] {26, 71}
BACOLRI/LE	59, 8, 4 [21, 47] {12, 26}	178, 14, 6 [27, 79] {15, 47}	640, 25, 6 [37, 107] {24, 63}
p		5	
BACOLR/ST	21, 9, 2 [54, 116] {30, 62}	40, 14, 0 [50, 114] {34, 64}	80, 24, 0 [64, 156] {52, 92}
BACOLR/LE	31, 9, 4 [62, 136] {34, 74}	76, 15, 2 [60, 138] {40, 78}	166, 24, 0 [82, 186] {58, 104}
BACOLRI/ST	22, 9, 7 [32, 62] {16, 30}	40, 14, 7 [36, 77] {21, 41}	93, 24, 4 [38, 88] {28, 50}
BACOLRI/LE	27, 8, 4 [22, 45] {12, 23}	80, 14, 2 [22, 52] {16, 30}	173, 24, 6 [44, 97] {30, 53}
p		7	
BACOLR/ST	15, 9, 0 [40, 84] {24, 44}	24, 15, 2 [64, 150] {40, 84}	36, 25, 3 [86, 220] {60, 130}
BACOLR/LE	17, 10, 2 [56, 124] {32, 68}	32, 15, 2 [64, 150] {40, 84}	56, 25, 2 [78, 198] {58, 118}
BACOLRI/ST	13, 10, 6 [28, 62] {16, 34}	22, 14, 5 [30, 67] {19, 37}	37, 24, 5 [40, 93] {29, 53}
BACOLRI/LE	18, 9, 6 [27, 57] {15, 30}	33, 14, 6 [32, 71] {20, 39}	53, 24, 6 [41, 98] {30, 57}
p		9	
BACOLR/ST	11, 9, 0 [40, 84] {24, 44}	13, 16, 2 [64, 164] {40, 96}	22, 26, 5 [84, 256] {54, 156}
BACOLR/LE	12, 9, 0 [40, 84] {24, 44}	17, 16, 2 [60, 158] {36, 90}	28, 25, 2 [74, 218] {50, 132}
BACOLRI/ST	12, 8, 3 [19, 40] {11, 21}	16, 14, 4 [28, 62] {18, 34}	22, 25, 6 [42, 104] {29, 60}
BACOLRI/LE	12, 8, 3 [19, 40] {11, 21}	18, 15, 6 [32, 75] {20, 42}	29, 24, 6 [39, 103] {27, 61}

Table 11: *Machine independent results for the Schrödinger Problem. We consider* $p = 4, 5, 7, 9$ *and* $tol = 10^{-4}, 10^{-6}, 10^{-8}$. *Table entries are of the form Final Nint, Accepted Time Steps, Remeshings [Calls to CRDCMP, CRSLVE] {Calls to CCRCMP, CCRSLVE}.*

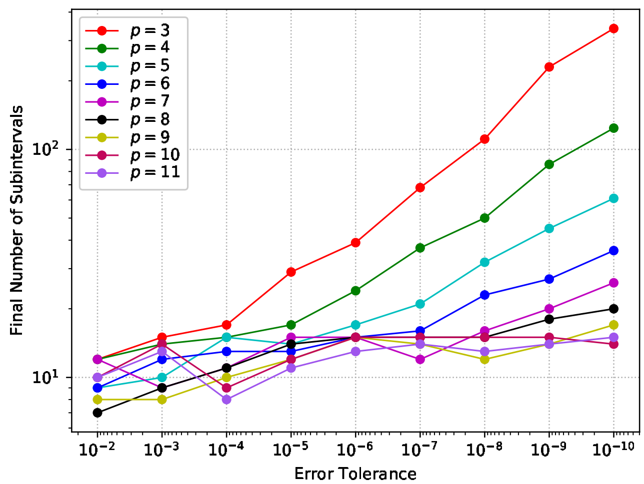

Figure 3: BACOLR/ST Number of Subintervals vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 3 \ldots 11$

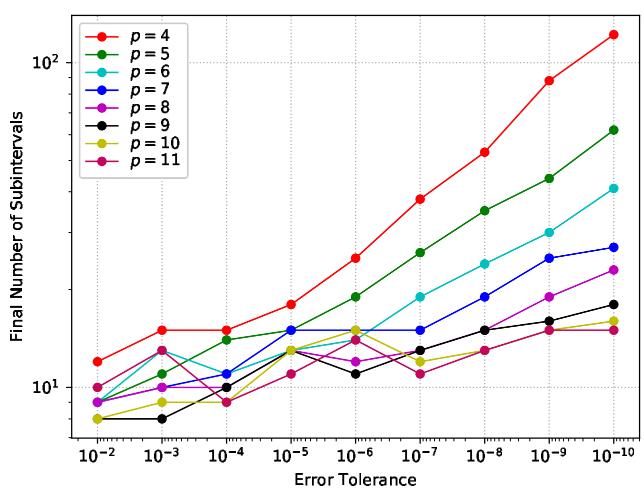

Figure 4: BACOLRI/ST Number of Subintervals vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

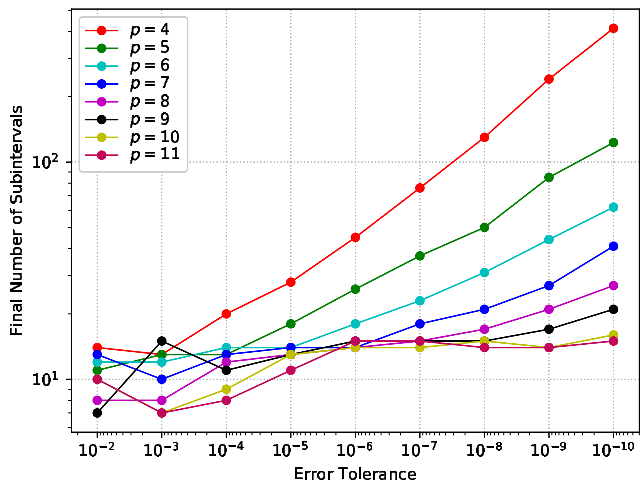

Figure 5: BACOLRI/LE Number of Subintervals vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \dots 11$

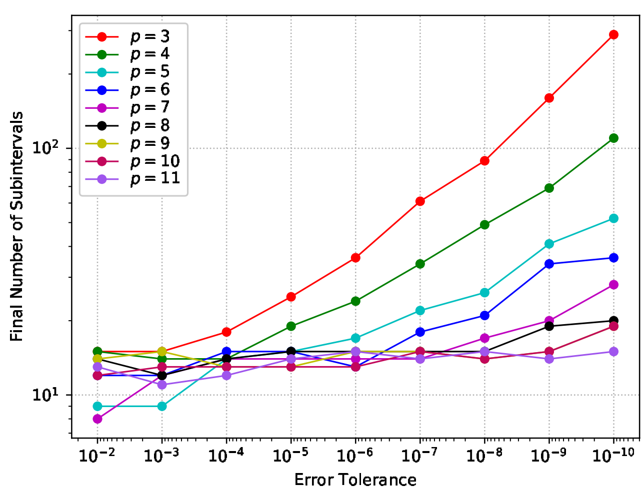

Figure 6: BACOLR/ST Number of Subintervals vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 3 \dots 11$

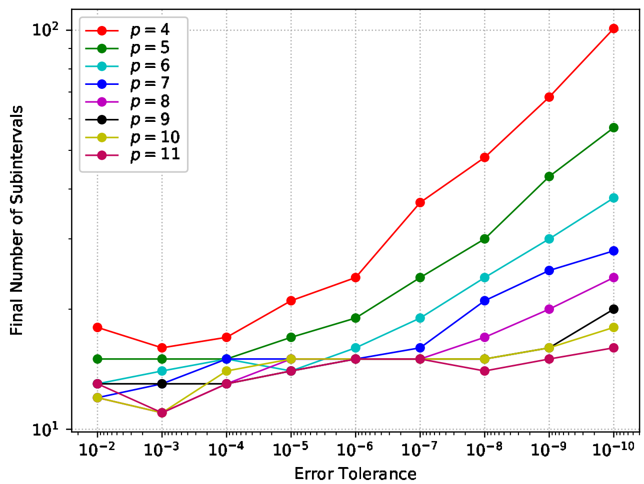

Figure 7: BACOLRI/ST Number of Subintervals vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

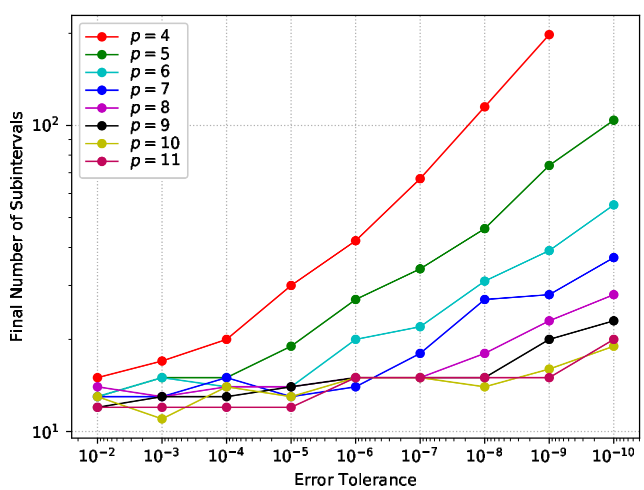

Figure 8: BACOLRI/LE Number of Subintervals vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

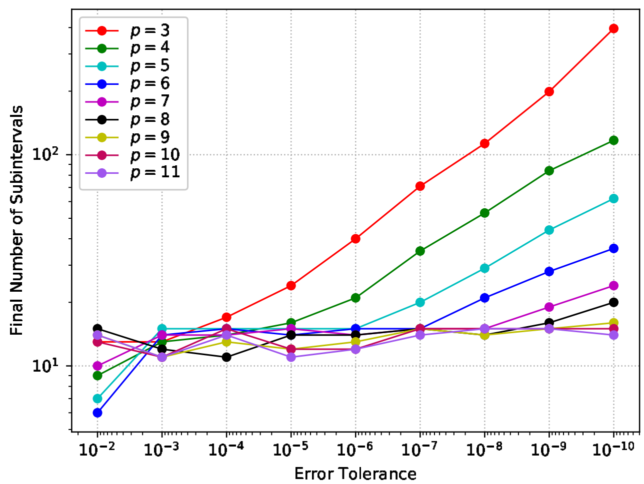

Figure 9: BACOLR/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 3 \ldots 11$

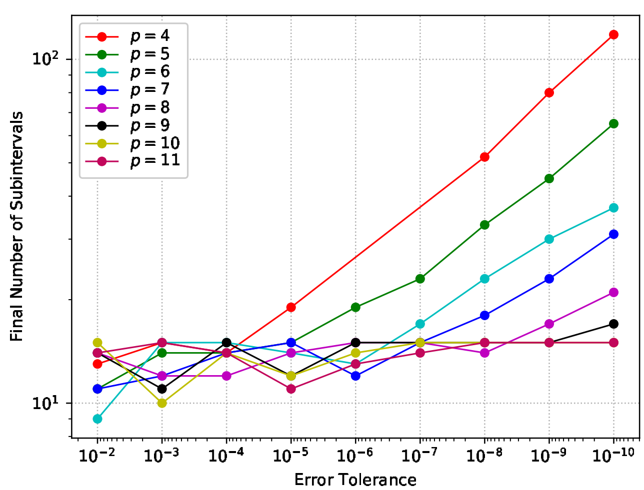

Figure 10: BACOLRI/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

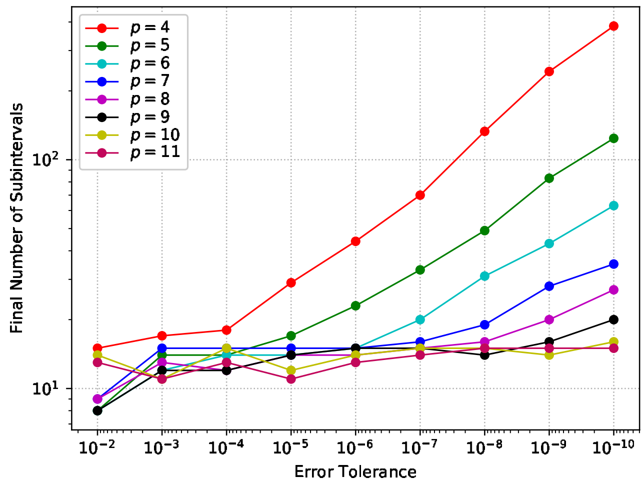

Figure 11: BACOLRI/LE Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

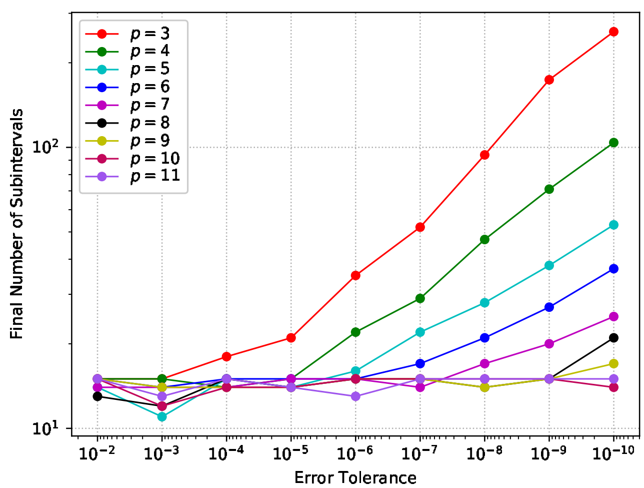

Figure 12: BACOLR/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 3 \ldots 11$

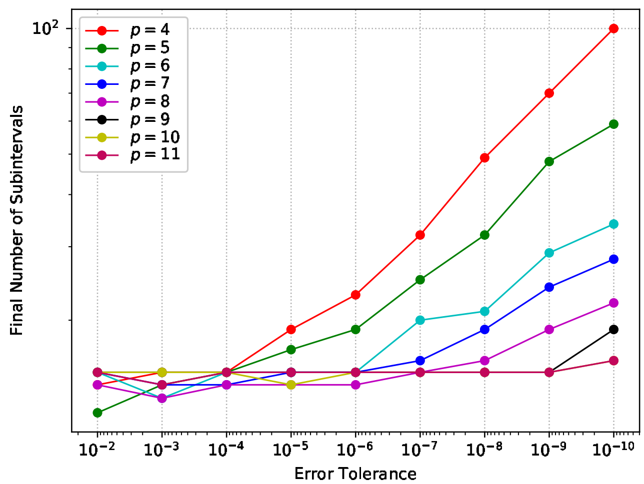

Figure 13: BACOLRI/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

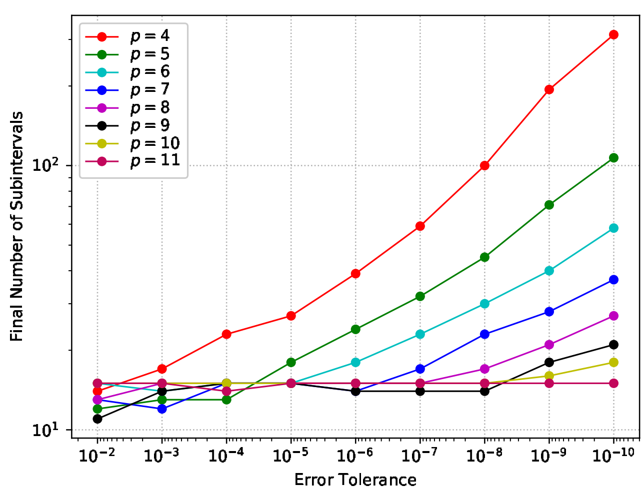

Figure 14: BACOLRI/LE Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

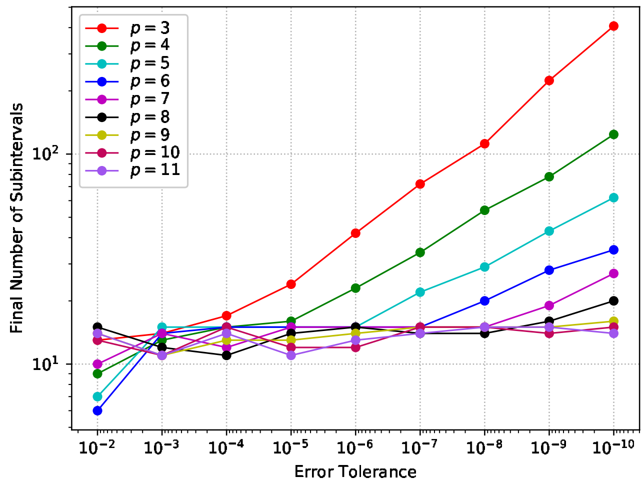

Figure 15: BACOLR/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 3 \ldots 11$

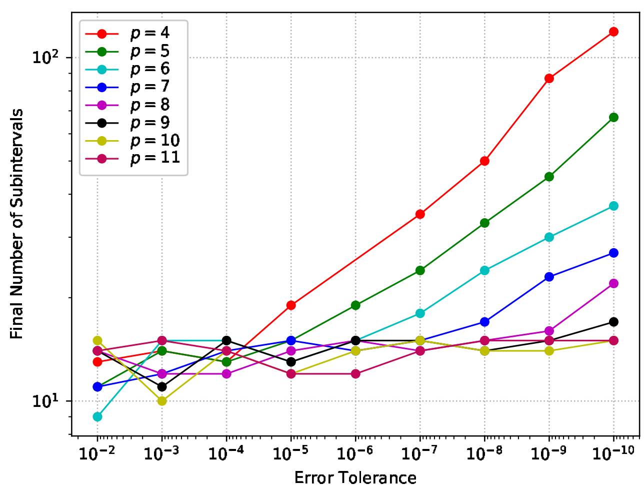

Figure 16: BACOLRI/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

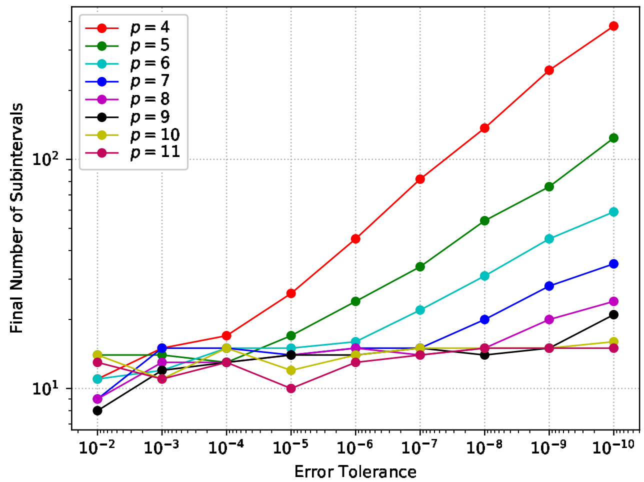

Figure 17: BACOLRI/LE Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

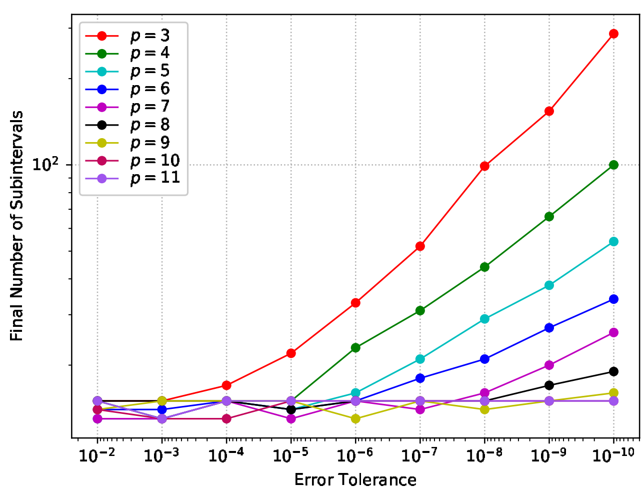

Figure 18: BACOLR/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 3 \ldots 11$

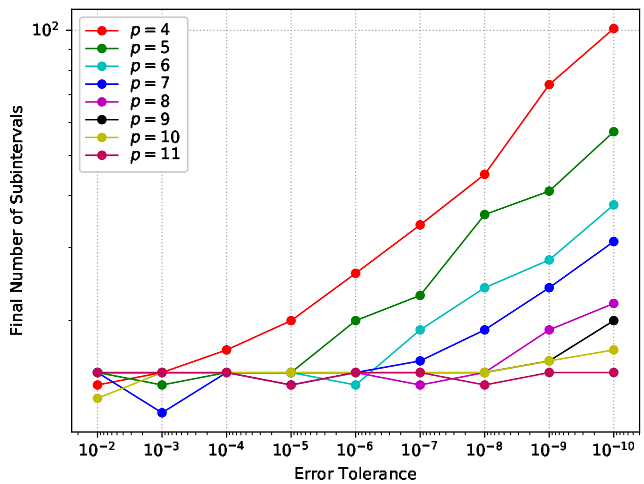

Figure 19: BACOLRI/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

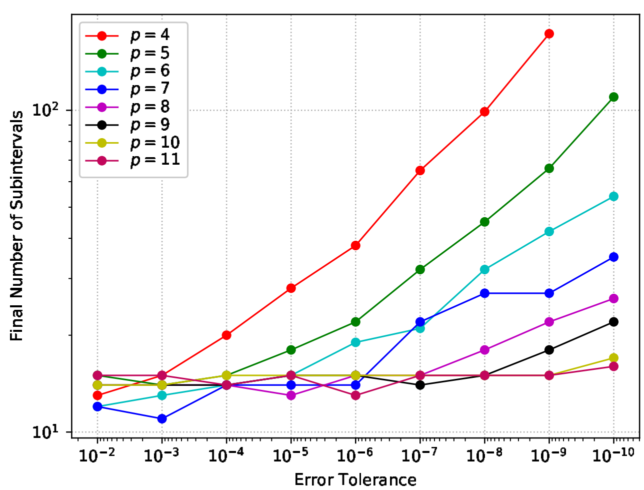

Figure 20: BACOLRI/LE Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

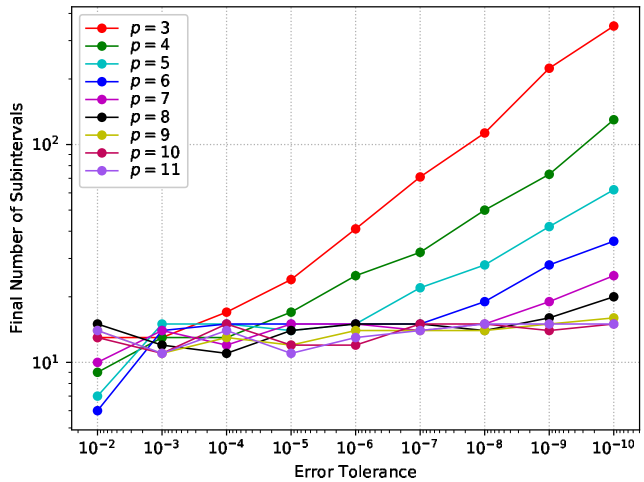

Figure 21: BACOLR/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 3 \ldots 11$

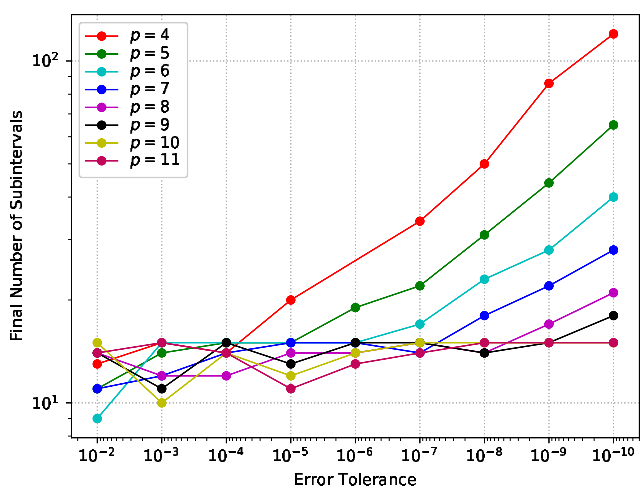

Figure 22: BACOLRI/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

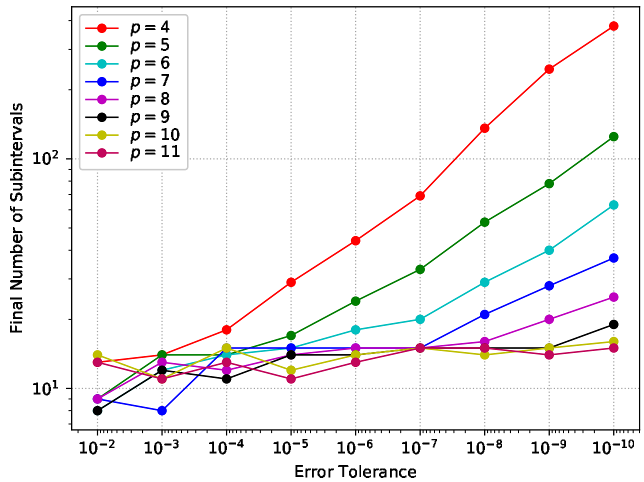

Figure 23: BACOLRI/LE Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

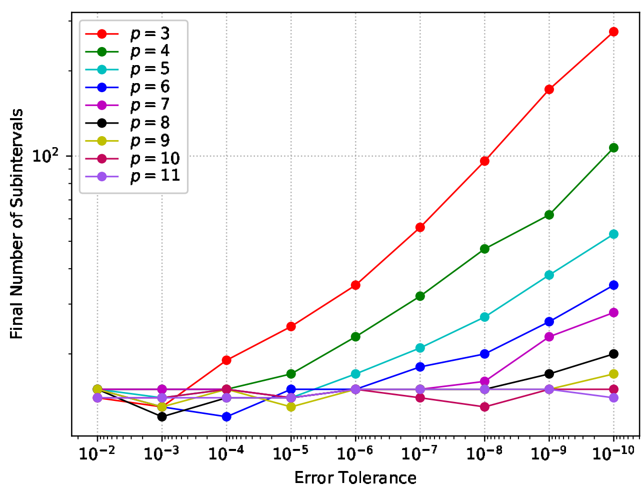

Figure 24: BACOLR/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 3 \ldots 11$

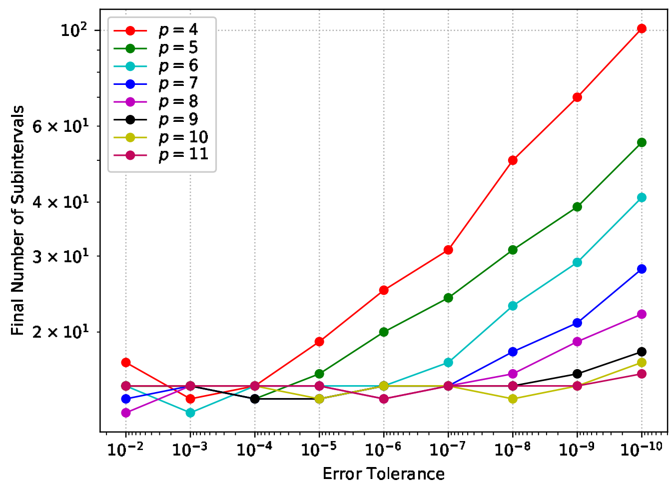

Figure 25: BACOLRI/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

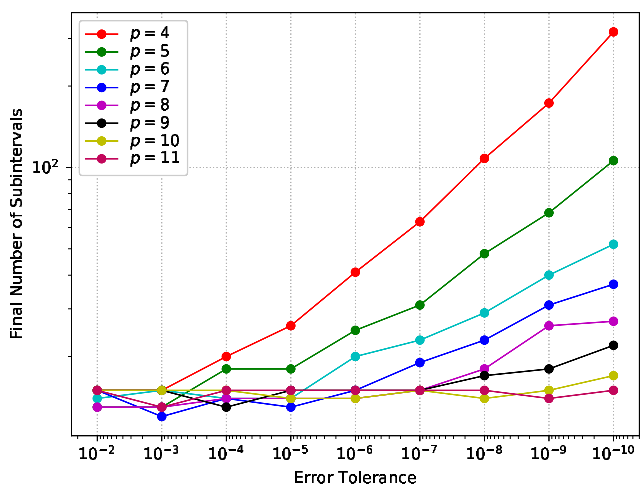

Figure 26: BACOLRI/LE Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

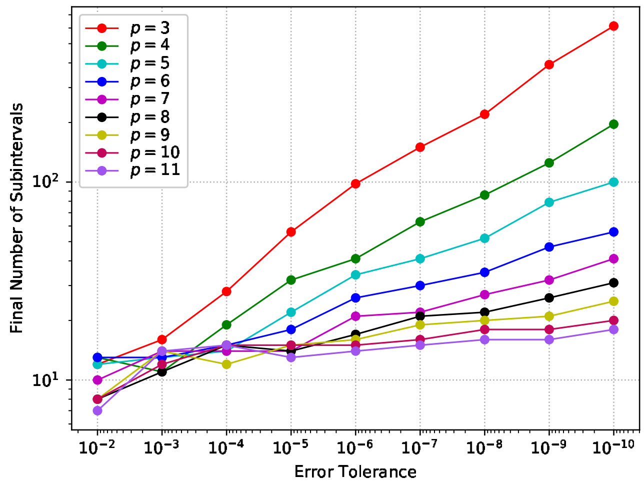

Figure 27: BACOLR/ST Number of Subintervals vs. Error Tolerance: Catalytic Surface Reaction Model with $p = 3 \ldots 11$

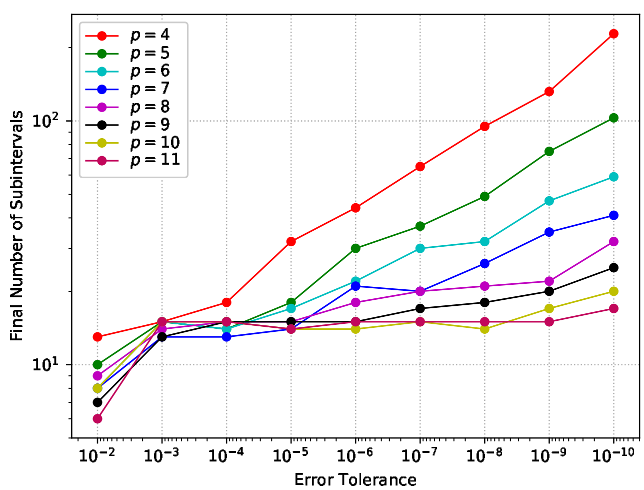

Figure 28: BACOLRI/ST Number of Subintervals vs. Error Tolerance: Catalytic Surface Reaction Model with $p = 4 \ldots 11$

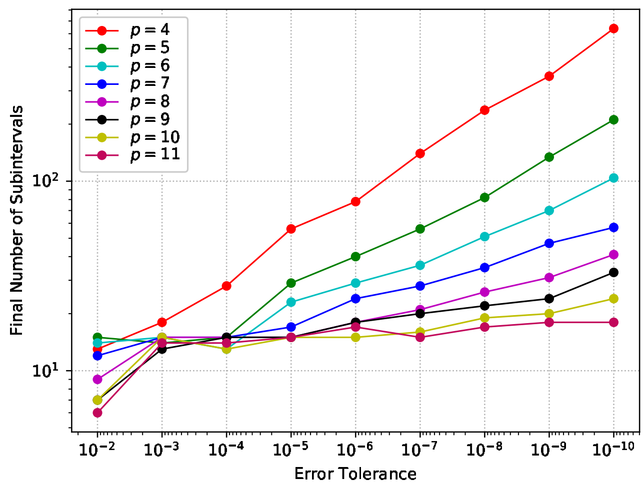

Figure 29: BACOLRI/LE Number of Subintervals vs. Error Tolerance: Catalytic Surface Reaction Model with $p = 4 \ldots 11$

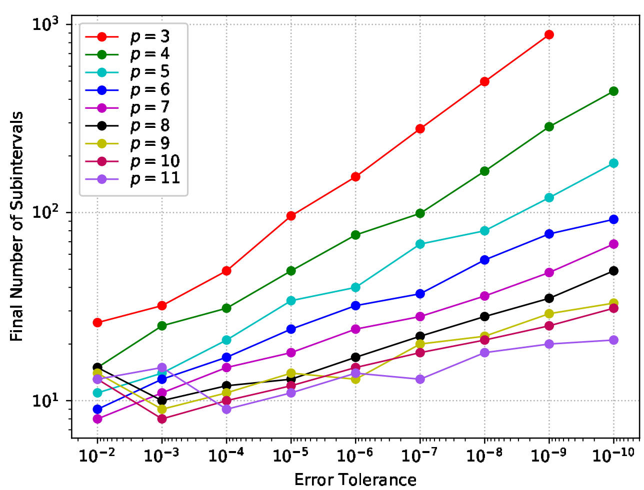

Figure 30: BACOLR/ST Number of Subintervals vs. Error Tolerance: Schrödinger Equation with $p = 3 \ldots 11$

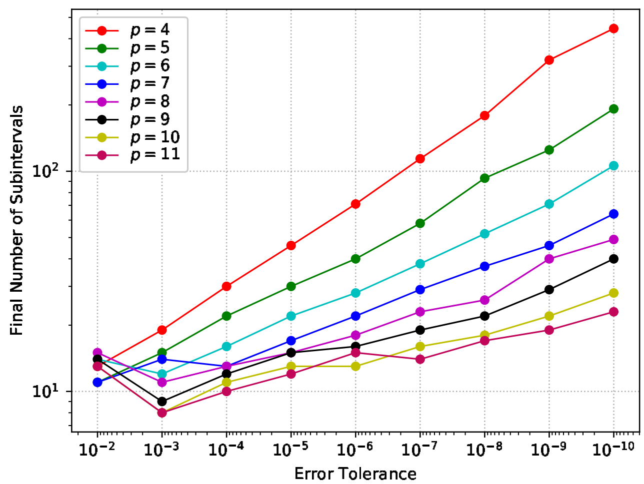

Figure 31: BACOLRI/ST Number of Subintervals vs. Error Tolerance: Schrödinger Equation with $p = 4 \ldots 11$

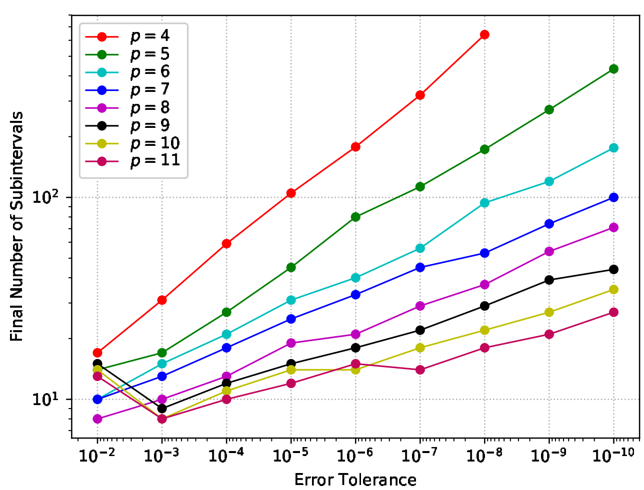

Figure 32: BACOLRI/LE Number of Subintervals vs. Error Tolerance: Schrödinger Equation with $p = 4 \ldots 11$

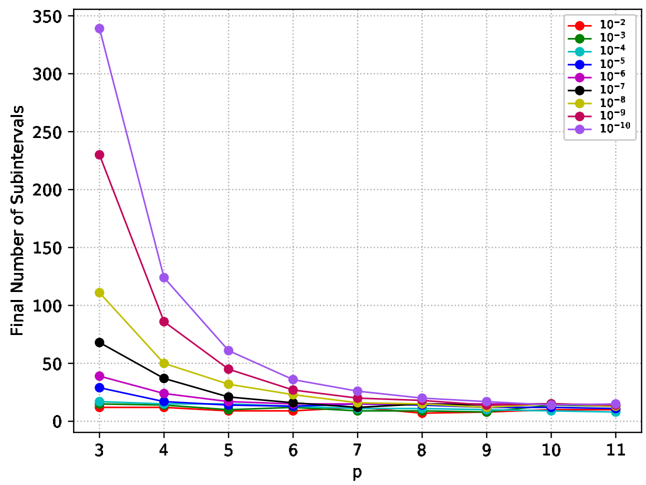

Figure 33: BACOLR/ST Number of Subintervals vs p: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $tol = 10^{-i}, i = 2 \ldots 10$

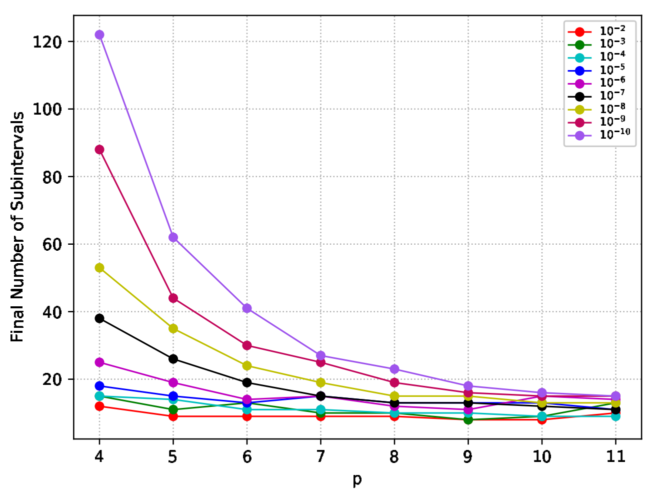

Figure 34: BACOLRI/ST Number of Subintervals vs p: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $tol = 10^{-i}, i = 2 \ldots 10$

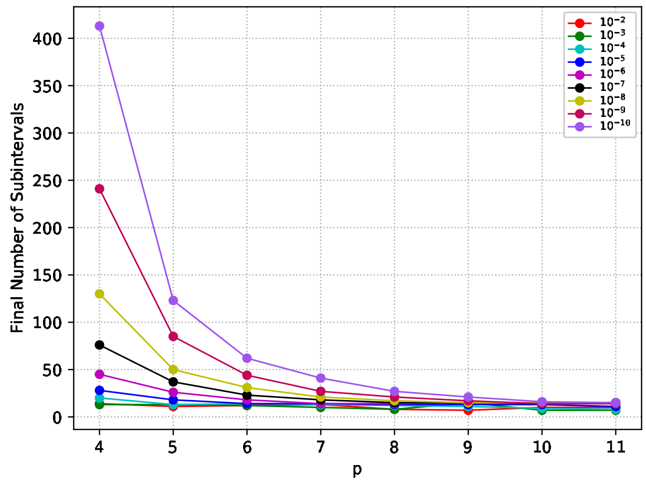

Figure 35: BACOLRI/LE Number of Subintervals vs p: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $tol = 10^{-i}, i = 2 \ldots 10$

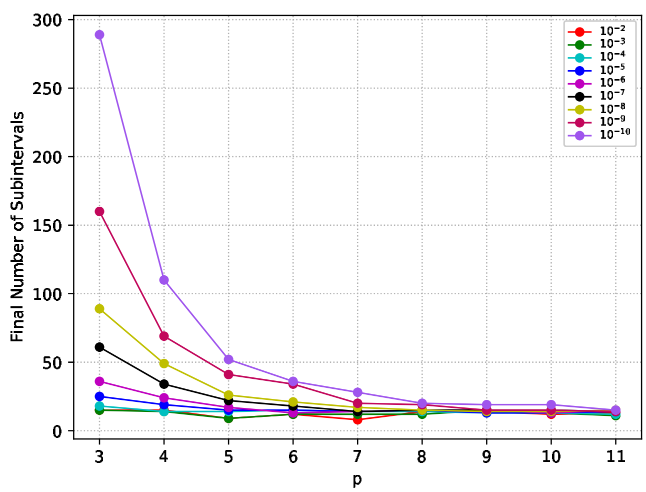

Figure 36: BACOLR/ST Number of Subintervals vs p: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $tol = 10^{-i}, i = 2 \ldots 10$

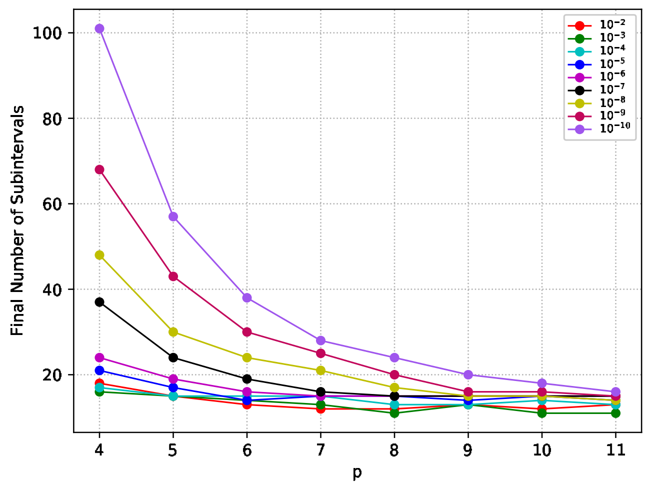

Figure 37: BACOLRI/ST Number of Subintervals vs p: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $tol = 10^{-i}, i = 2 \ldots 10$

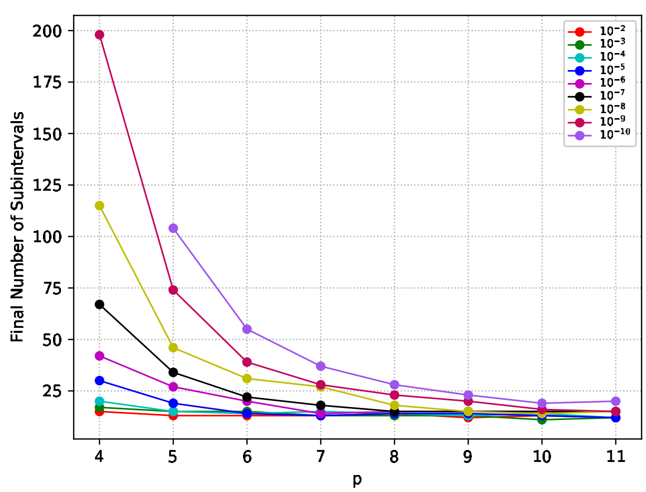

Figure 38: BACOLRI/LE Number of Subintervals vs p: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $tol = 10^{-i}, i = 2 \ldots 10$

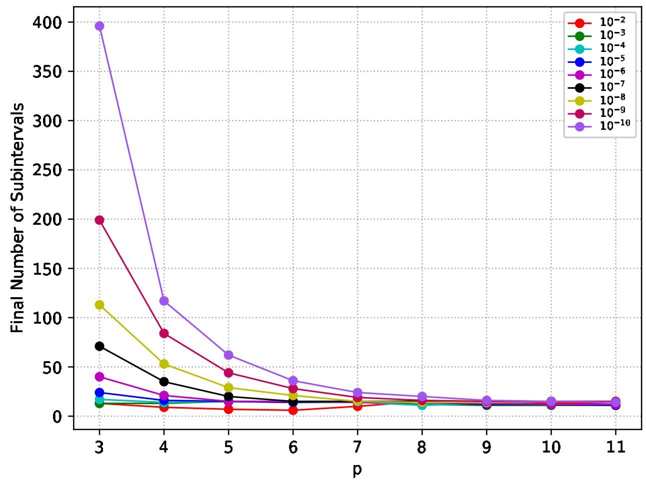

Figure 39: BACOLR/ST Number of Subintervals vs p: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $tol = 10^{-i}, i = 2 \ldots 10$

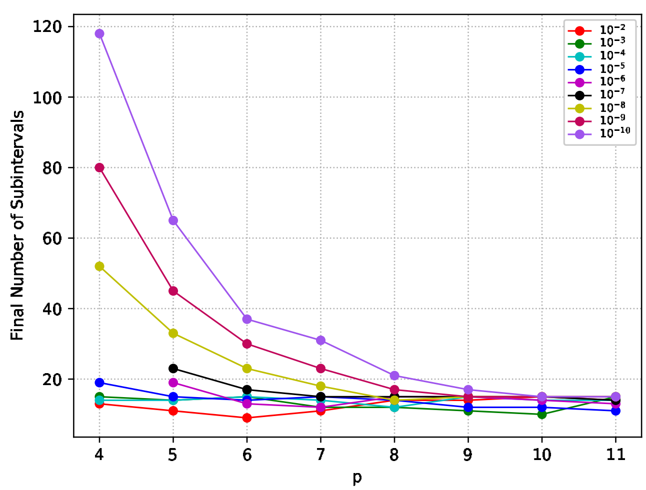

Figure 40: BACOLRI/ST Number of Subintervals vs p: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $tol = 10^{-i}, i = 2 \ldots 10$

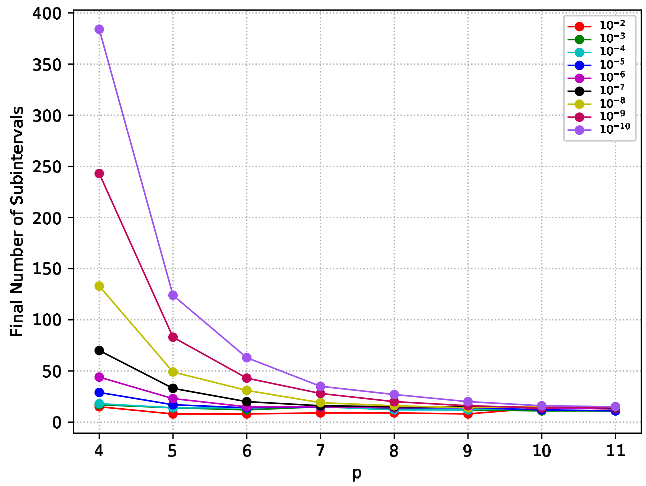

Figure 41: BACOLRI/LE Number of Subintervals vs p: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $tol = 10^{-i}, i = 2 \ldots 10$

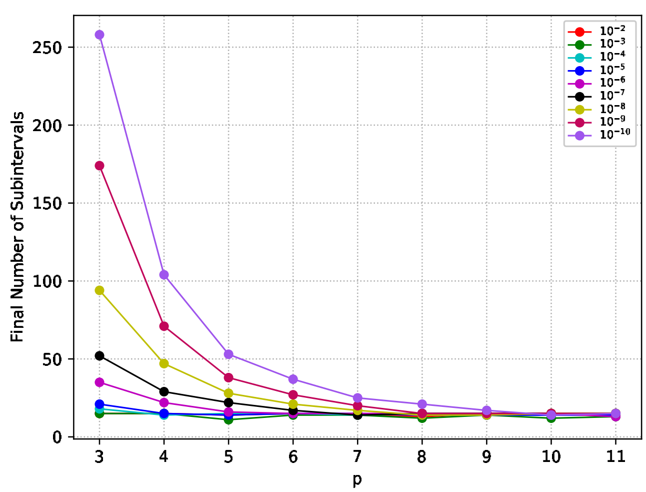

Figure 42: BACOLR/ST Number of Subintervals vs p: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $tol = 10^{-i}, i = 2 \ldots 10$

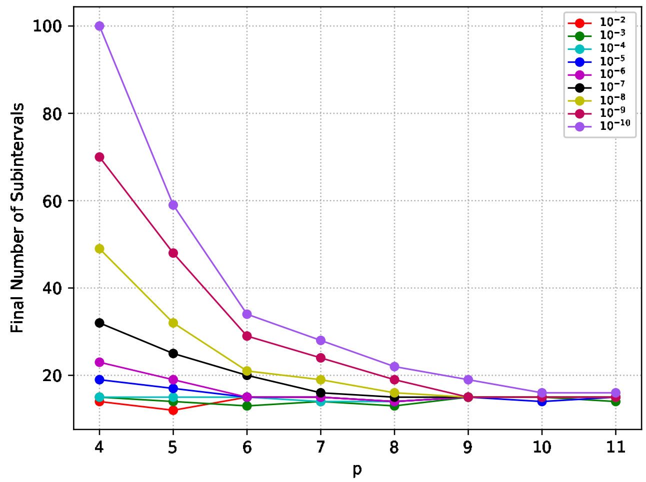

Figure 43: BACOLRI/ST Number of Subintervals vs p: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $tol = 10^{-i}, i = 2 \dots 10$

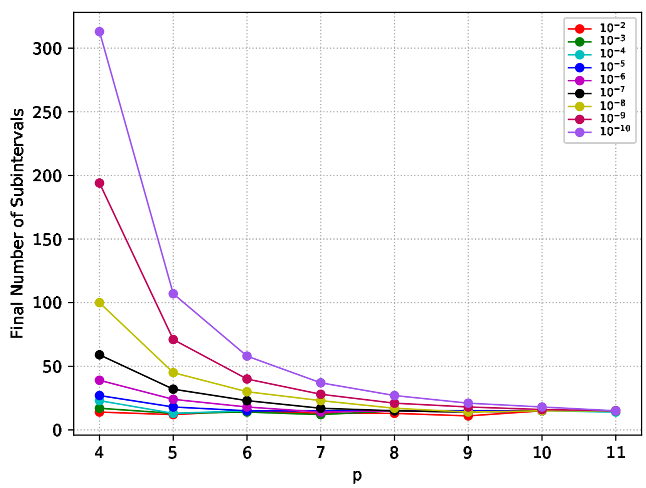

Figure 44: BACOLRI/LE Number of Subintervals vs p: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $tol = 10^{-i}, i = 2 \dots 10$

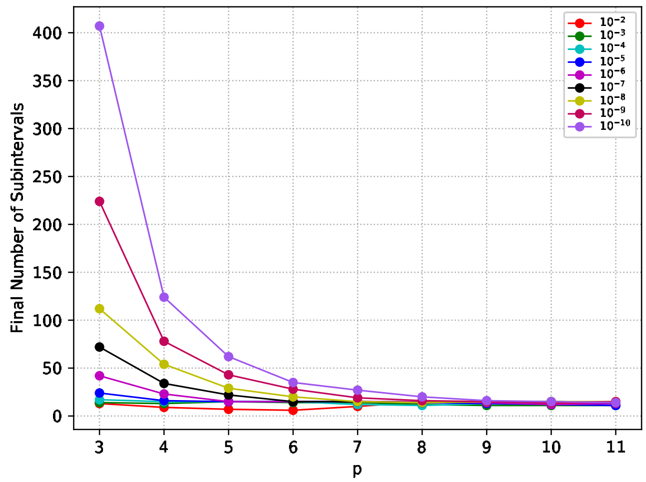

Figure 45: BACOLR/ST Number of Subintervals vs p: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $tol = 10^{-i}, i = 2 \ldots 10$

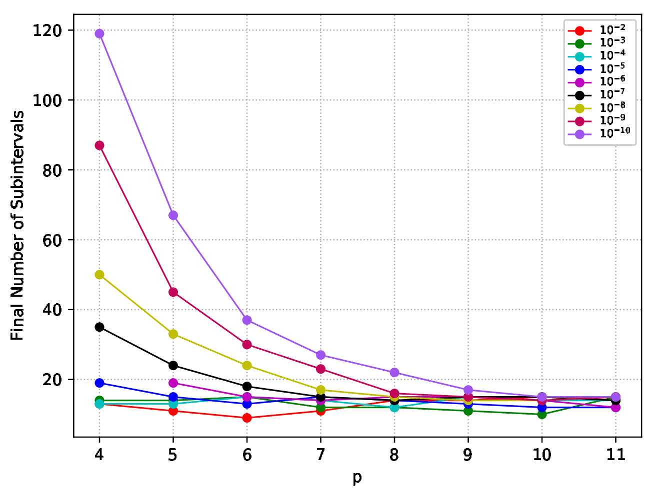

Figure 46: BACOLRI/ST Number of Subintervals vs p: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $tol = 10^{-i}, i = 2 \ldots 10$

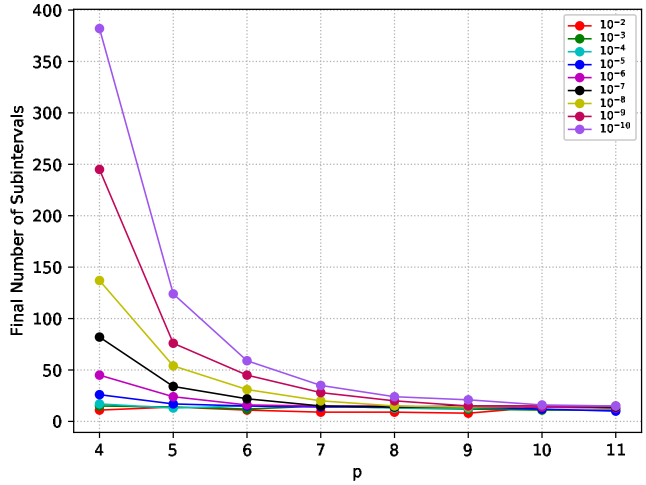

Figure 47: BACOLRI/LE Number of Subintervals vs p: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $tol = 10^{-i}, i = 2 \ldots 10$

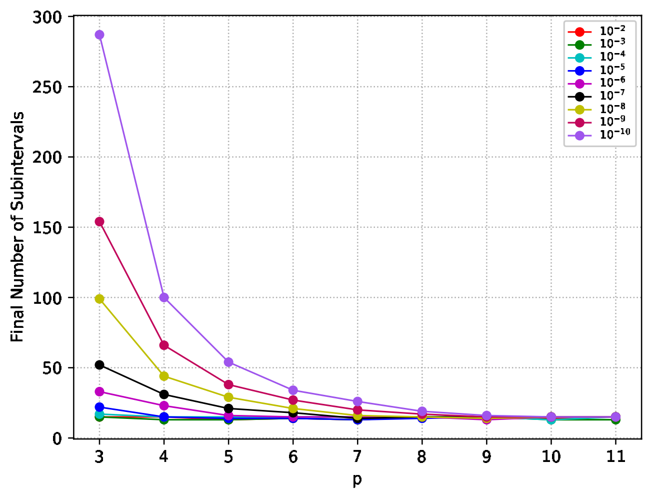

Figure 48: BACOLR/ST Number of Subintervals vs p: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $tol = 10^{-i}, i = 2 \ldots 10$

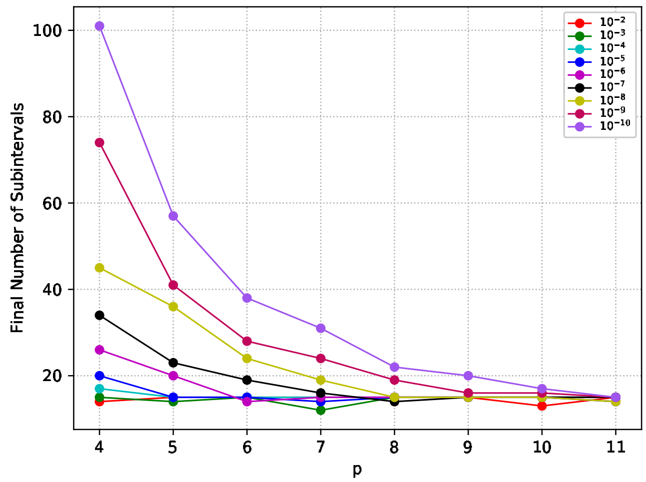

Figure 49: BACOLRI/ST Number of Subintervals vs p: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $tol = 10^{-i}, i = 2 \ldots 10$

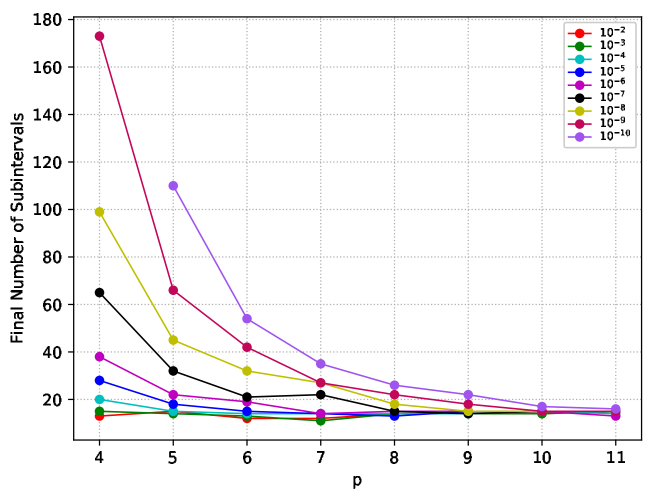

Figure 50: BACOLRI/LE Number of Subintervals vs p: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $tol = 10^{-i}, i = 2 \ldots 10$

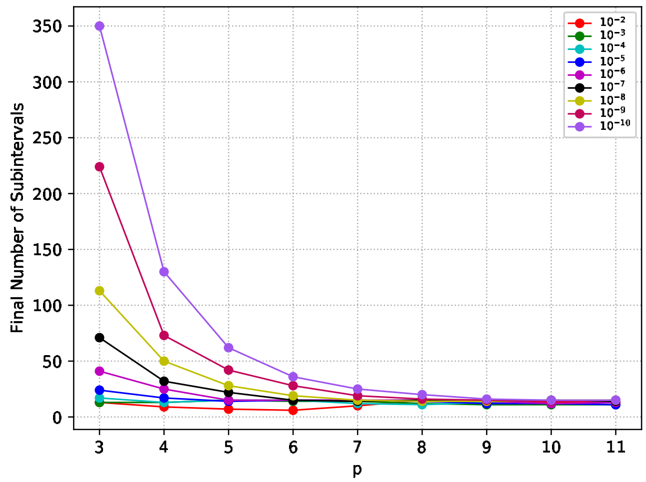

Figure 51: BACOLR/ST Number of Subintervals vs p: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $tol = 10^{-i}, i = 2 \dots 10$

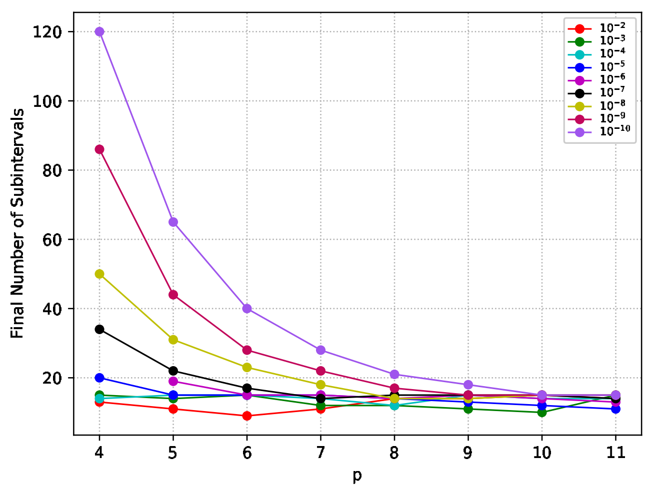

Figure 52: BACOLRI/ST Number of Subintervals vs p: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $tol = 10^{-i}, i = 2 \dots 10$

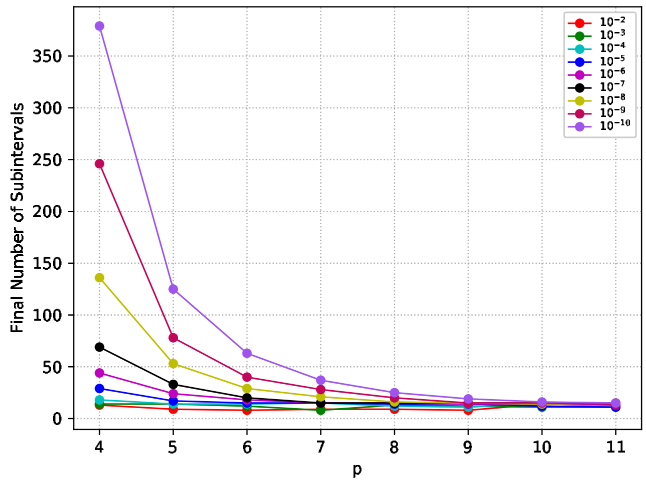

Figure 53: BACOLRI/LE Number of Subintervals vs p: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $tol = 10^{-i}, i = 2 \ldots 10$

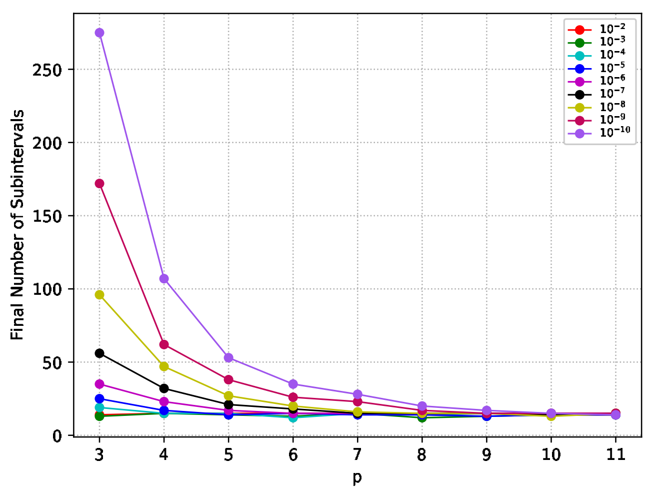

Figure 54: BACOLR/ST Number of Subintervals vs p: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $tol = 10^{-i}, i = 2 \ldots 10$

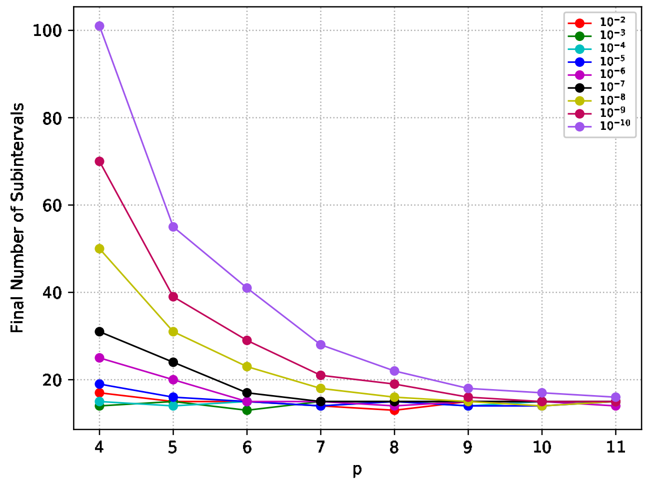

Figure 55: BACOLRI/ST Number of Subintervals vs p: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $tol = 10^{-i}, i = 2 \ldots 10$

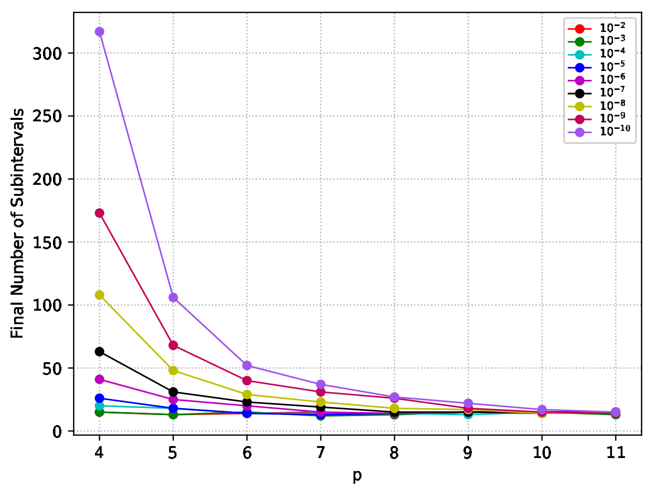

Figure 56: BACOLRI/LE Number of Subintervals vs p: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $tol = 10^{-i}, i = 2 \ldots 10$

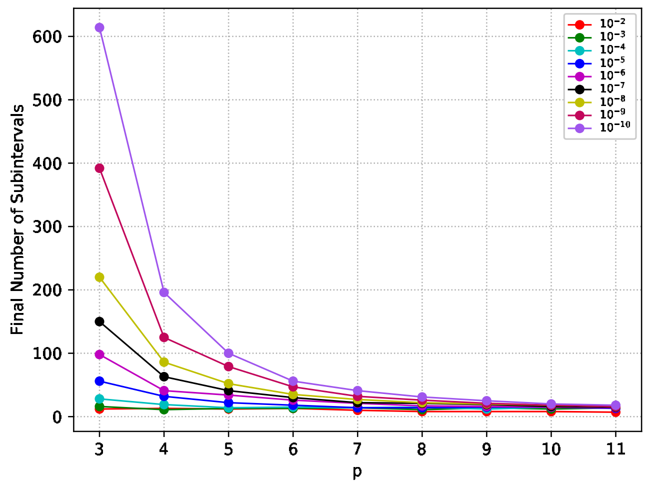

Figure 57: BACOLR/ST Number of Subintervals vs p: Catalytic Surface Reaction Model with $tol = 10^{-i}, i = 2 \ldots 10$

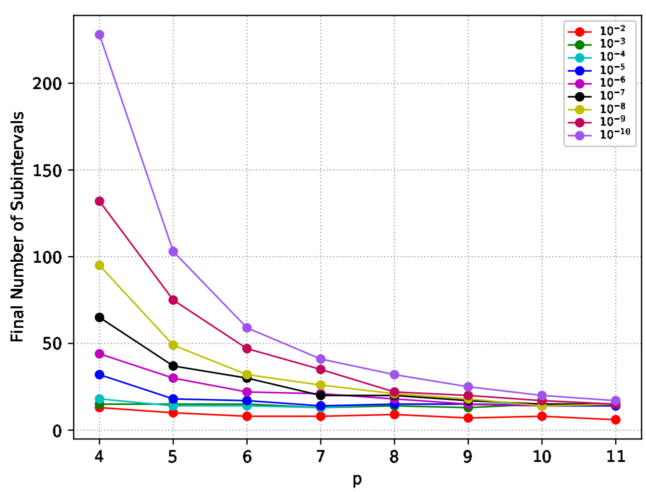

Figure 58: BACOLRI/ST Number of Subintervals vs p: Catalytic Surface Reaction Model with $tol = 10^{-i}, i = 2 \ldots 10$

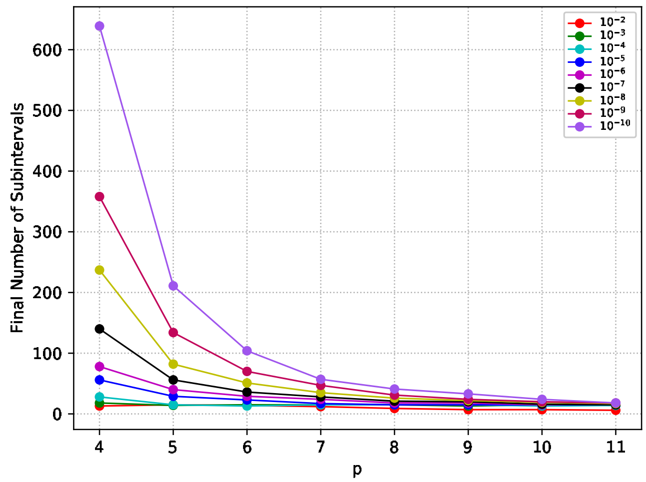

Figure 59: BACOLRI/LE Number of Subintervals vs p: Catalytic Surface Reaction Model with $tol = 10^{-i}, i = 2 \ldots 10$

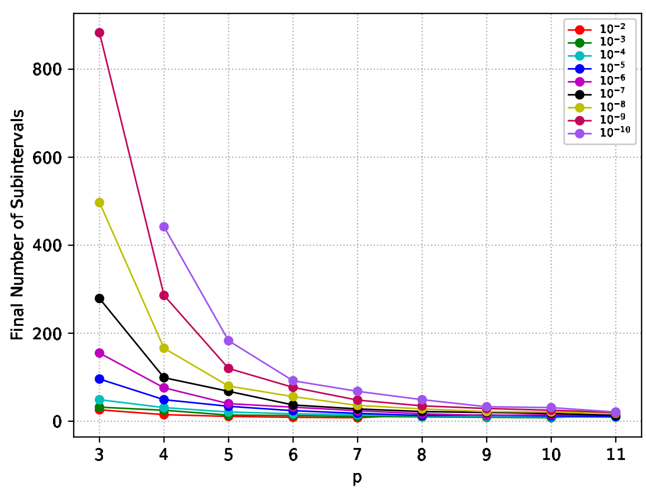

Figure 60: BACOLR/ST Number of Subintervals vs p: Schrödinger Equation with $tol = 10^{-i}, i = 2 \ldots 10$

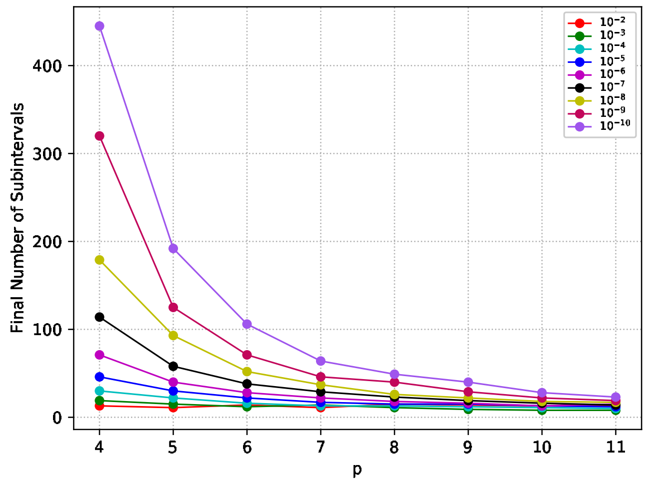

Figure 61: BACOLRI/ST Number of Subintervals vs p: Schrödinger Equation with $tol = 10^{-i}, i = 2 \ldots 10$

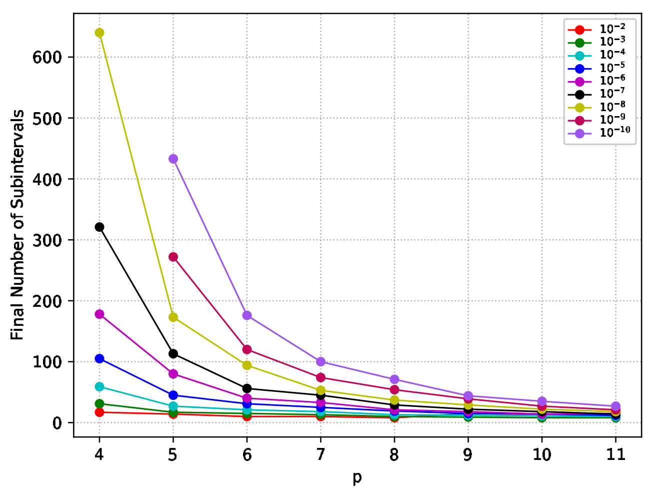

Figure 62: BACOLRI/LE Number of Subintervals vs p: Schrödinger Equation with $tol = 10^{-i}, i = 2 \ldots 10$

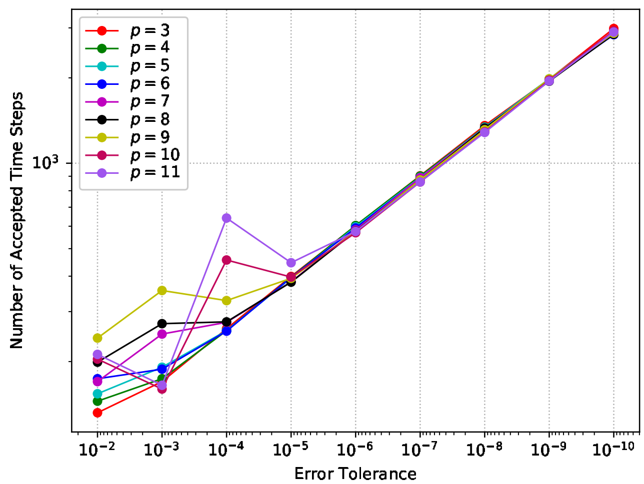

Figure 63: BACOLR/ST Number of Accepted Time Steps vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 3 \ldots 11$

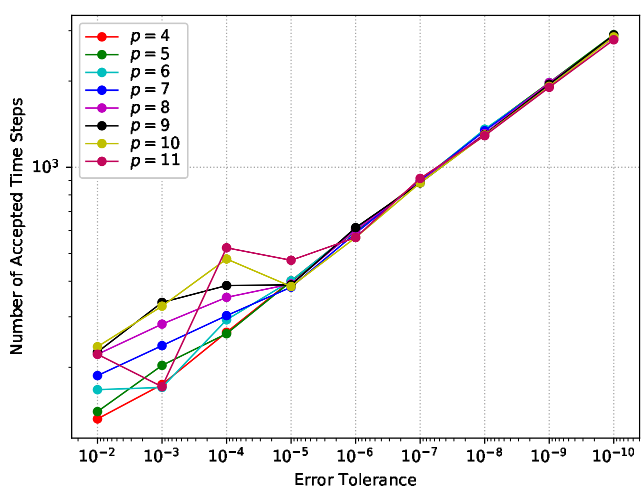

Figure 64: BACOLRI/ST Number of Accepted Time Steps vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

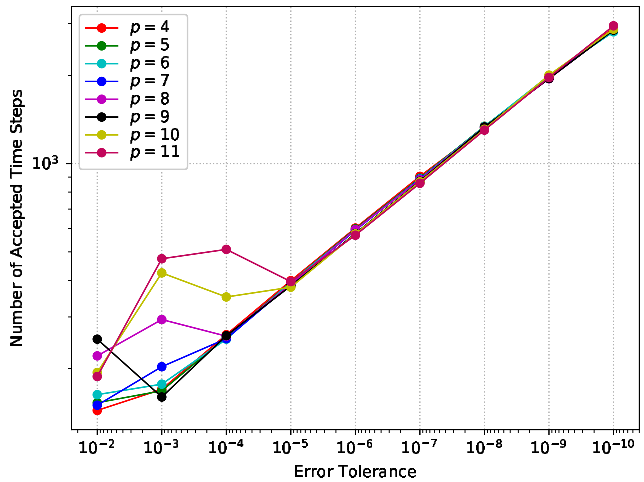

Figure 65: BACOLRI/LE Number of Accepted Time Steps vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

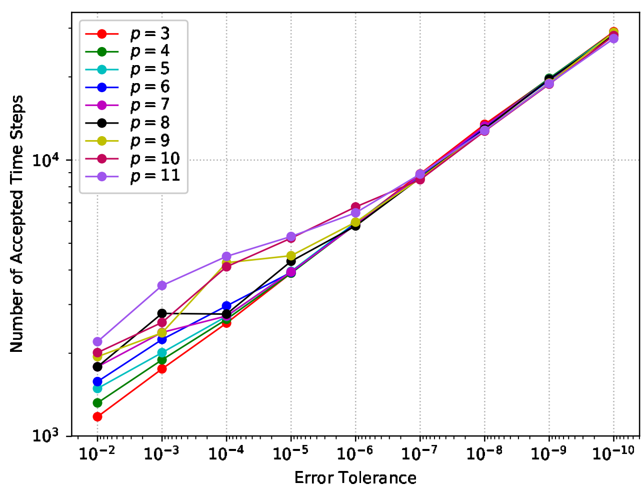

Figure 66: BACOLR/ST Number of Accepted Time Steps vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 3 \ldots 11$

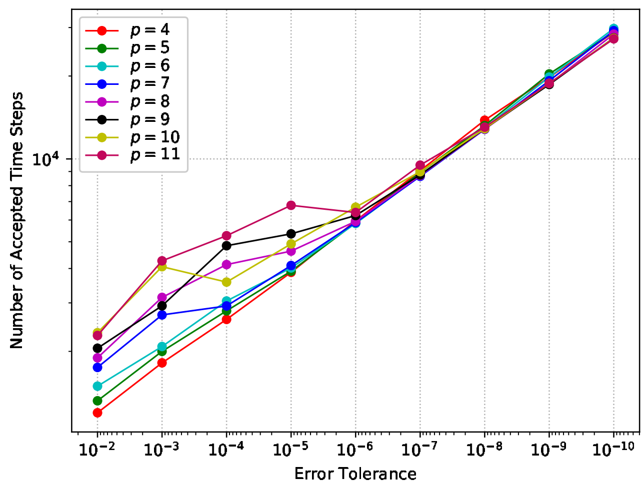

Figure 67: BACOLRI/ST Number of Accepted Time Steps vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \dots 11$

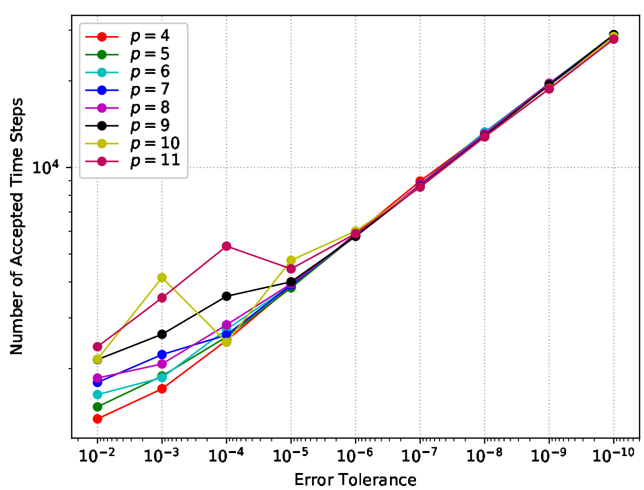

Figure 68: BACOLRI/LE Number of Accepted Time Steps vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \dots 11$

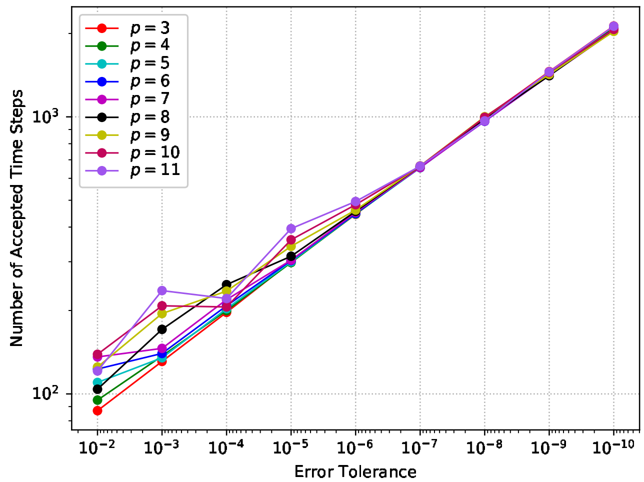

Figure 69: BACOLR/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 3 \ldots 11$

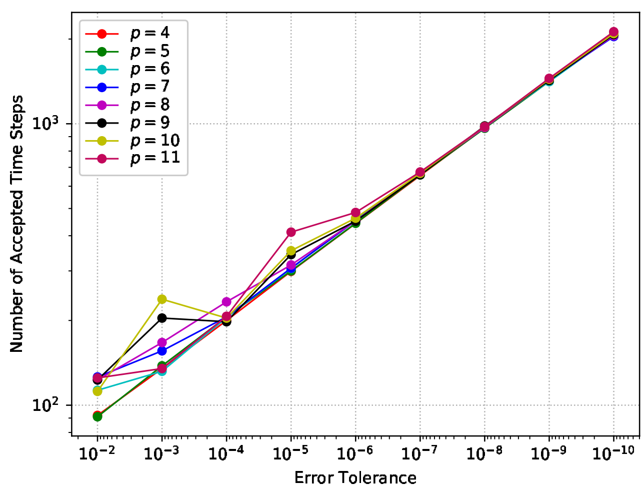

Figure 70: BACOLRI/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

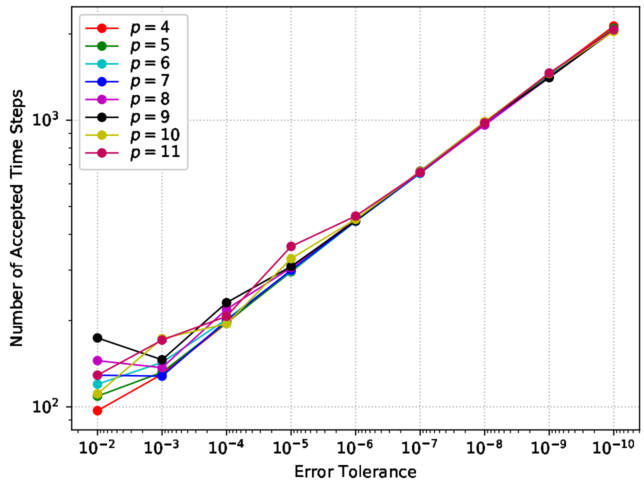

Figure 71: BACOLRI/LE Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

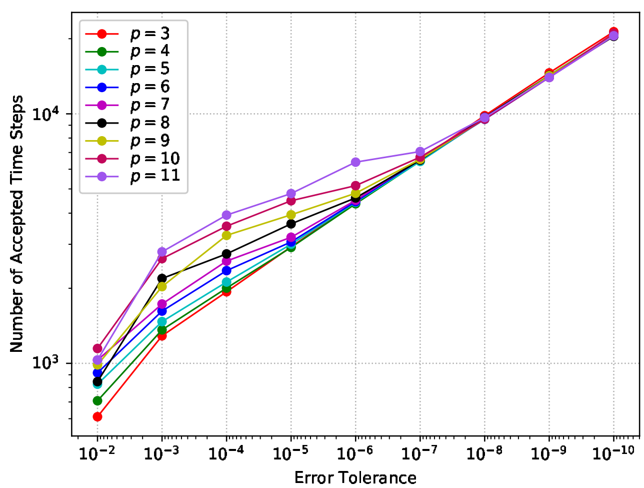

Figure 72: BACOLR/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 3 \ldots 11$

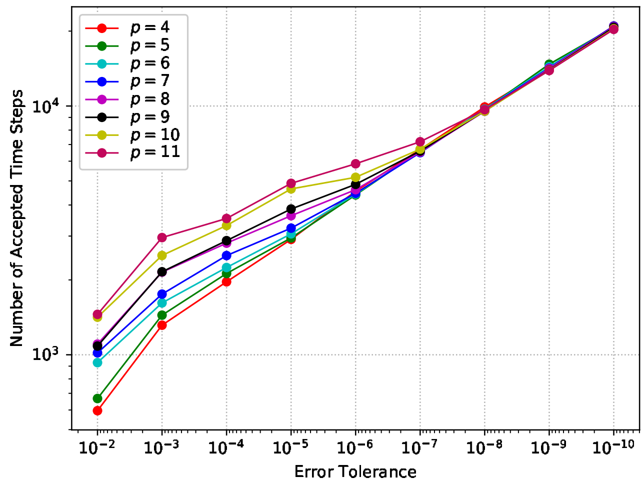

Figure 73: BACOLRI/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

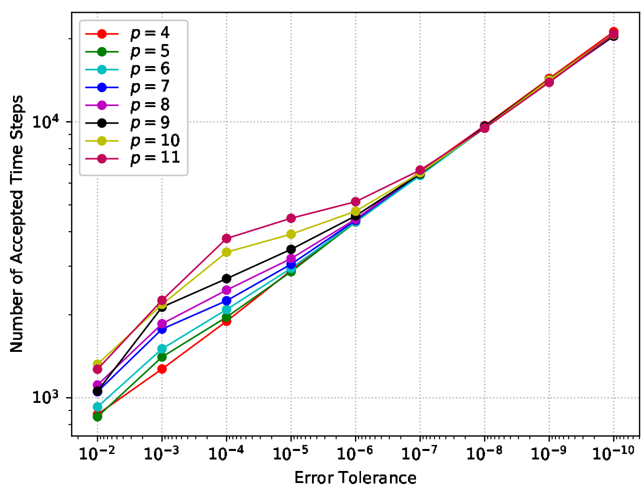

Figure 74: BACOLRI/LE Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

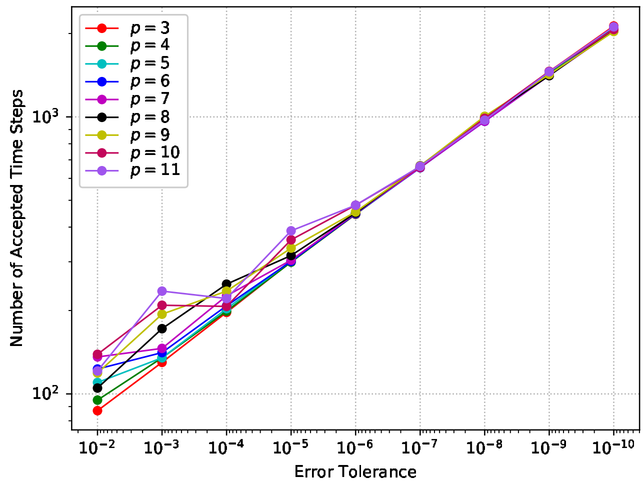

Figure 75: BACOLR/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 3 \ldots 11$

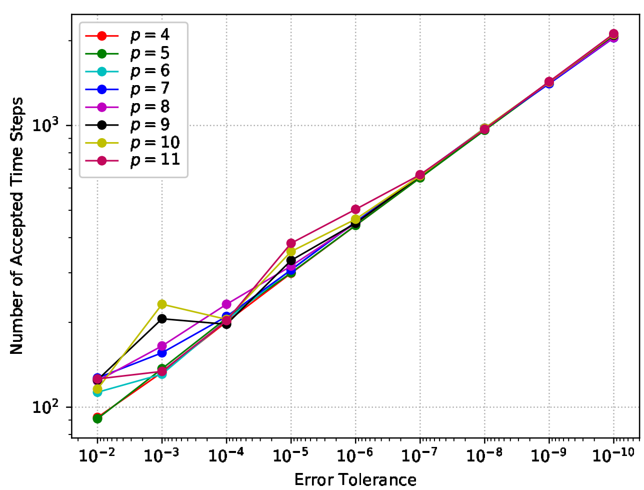

Figure 76: BACOLRI/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

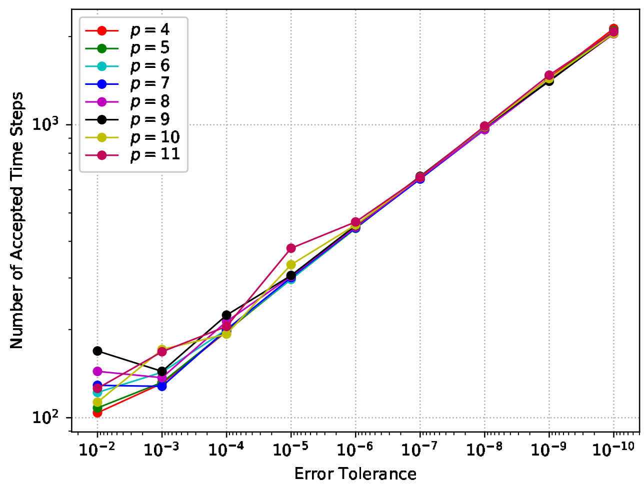

Figure 77: BACOLRI/LE Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

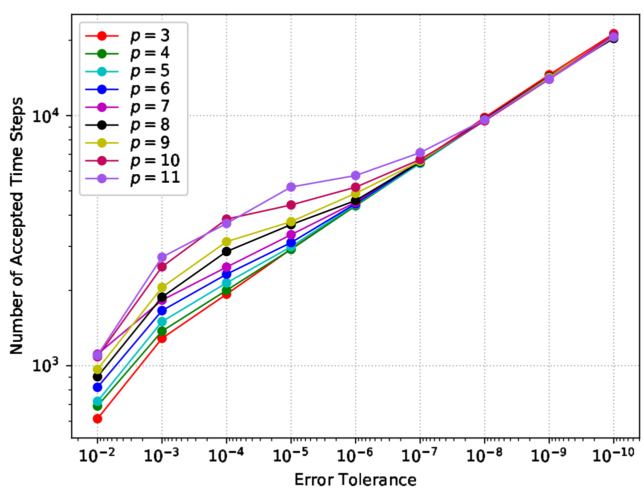

Figure 78: BACOLR/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 3 \ldots 11$

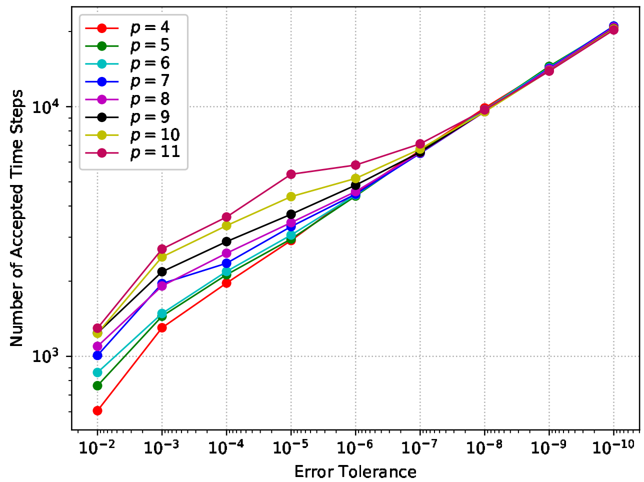

Figure 79: BACOLRI/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 4 \dots 11$

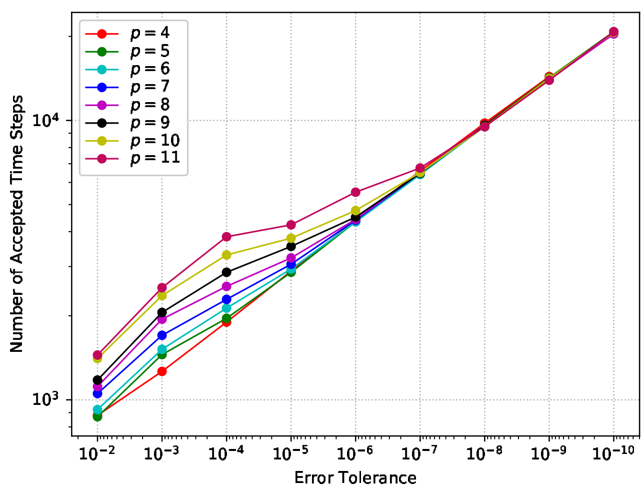

Figure 80: BACOLRI/LE Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 4 \dots 11$

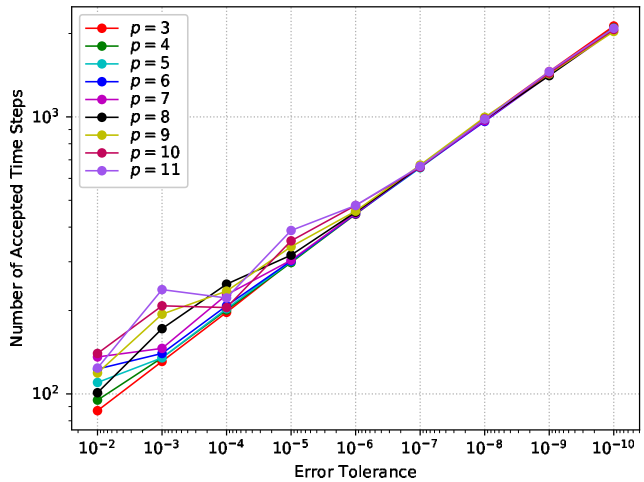

Figure 81: BACOLR/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 3 \ldots 11$

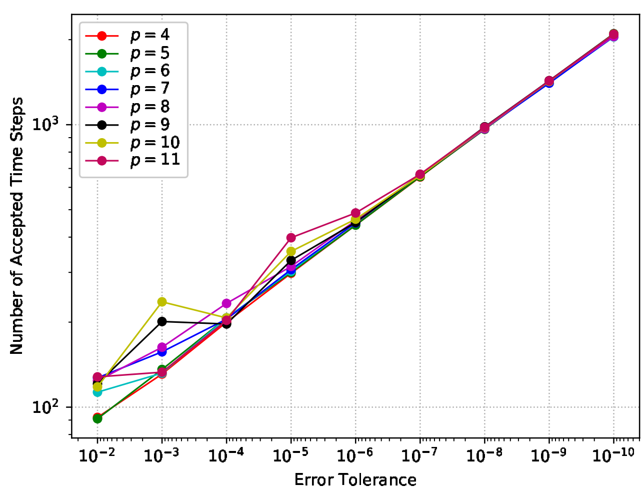

Figure 82: BACOLRI/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

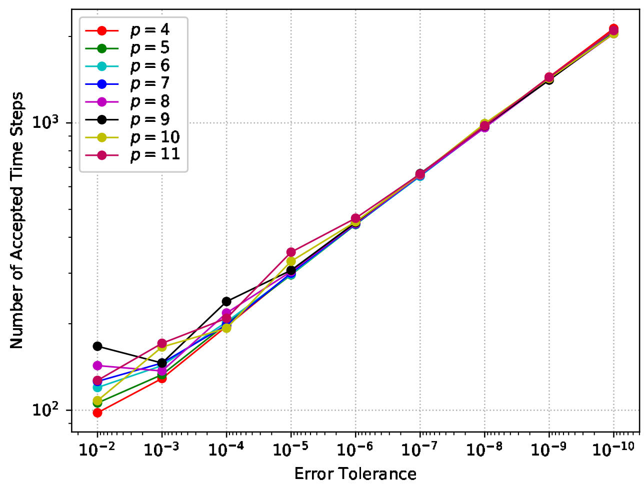

Figure 83: BACOLRI/LE Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

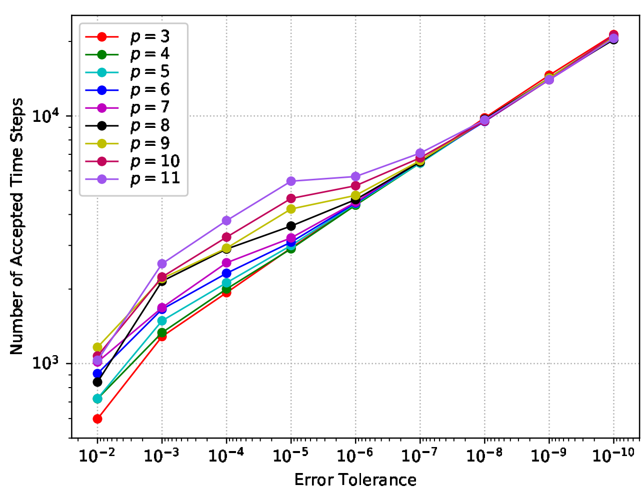

Figure 84: BACOLR/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 3 \ldots 11$

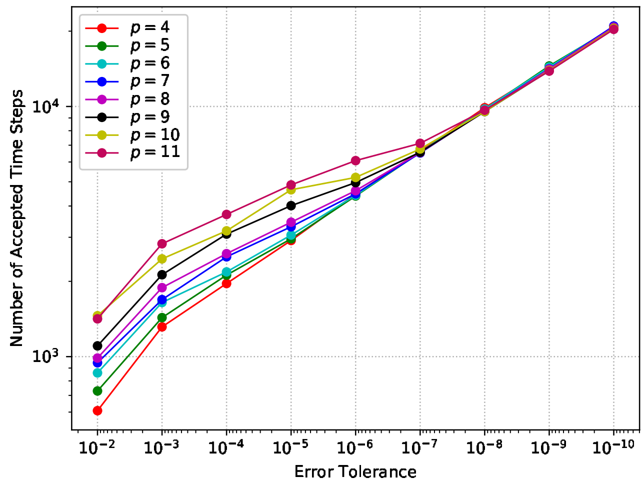

Figure 85: BACOLRI/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

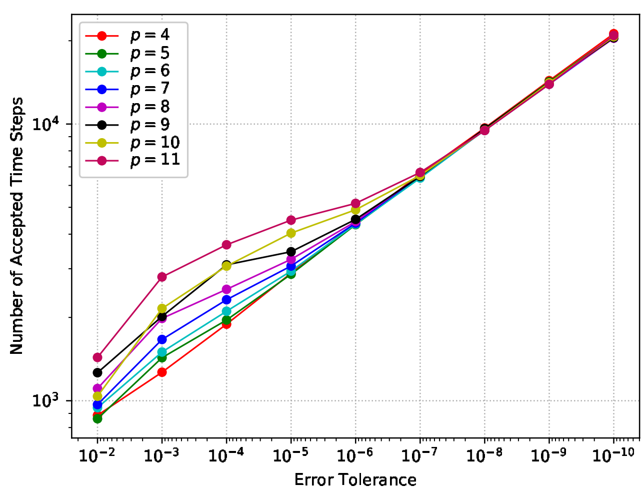

Figure 86: BACOLRI/LE Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

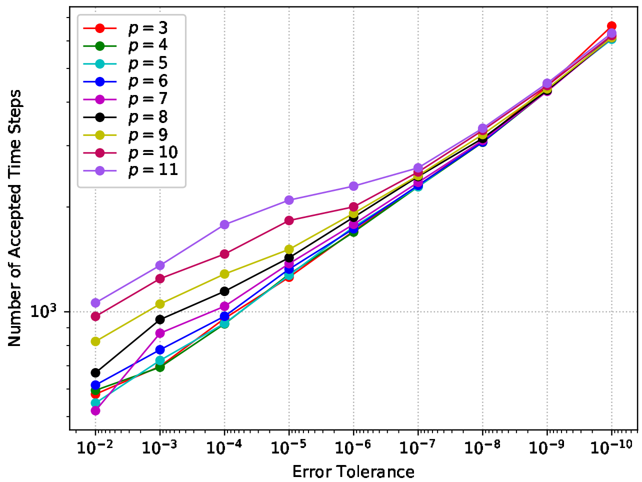

Figure 87: BACOLR/ST Number of Accepted Time Steps vs. Error Tolerance: Catalytic Surface Reaction Model with $p = 3 \ldots 11$

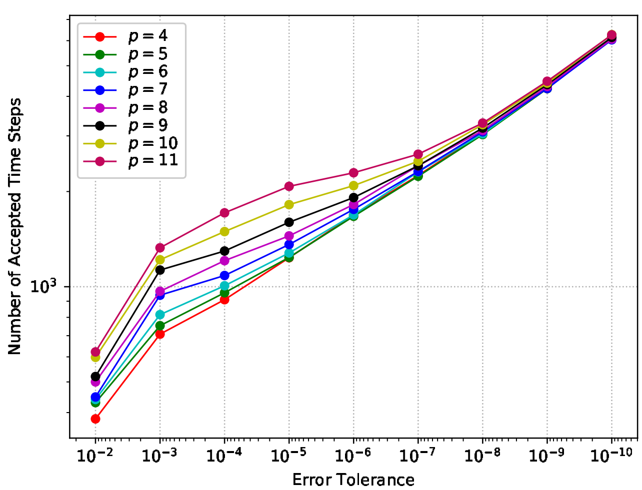

Figure 88: BACOLRI/ST Number of Accepted Time Steps vs. Error Tolerance: Catalytic Surface Reaction Model with $p = 4 \ldots 11$

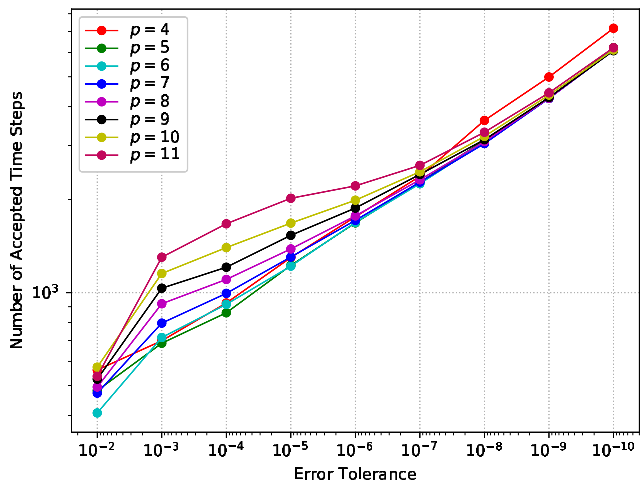

Figure 89: BACOLRI/LE Number of Accepted Time Steps vs. Error Tolerance: Catalytic Surface Reaction Model with $p = 4 \ldots 11$

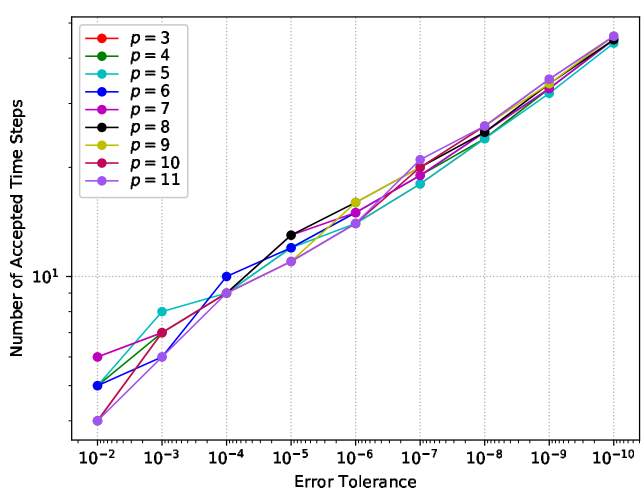

Figure 90: BACOLR/ST Number of Accepted Time Steps vs. Error Tolerance: Schrödinger Equation with $p = 3 \ldots 11$

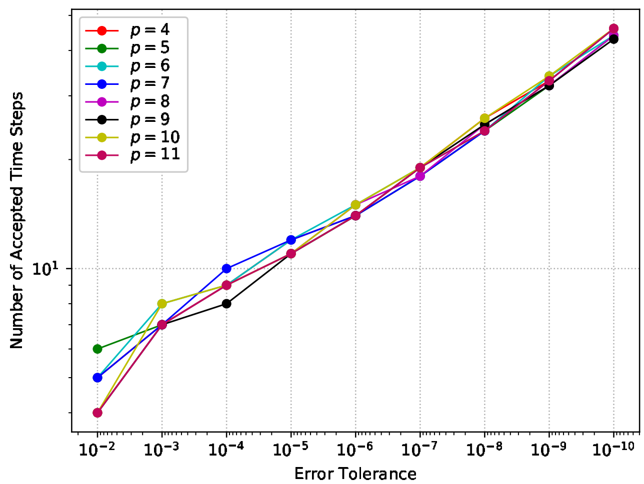

Figure 91: BACOLRI/ST Number of Accepted Time Steps vs. Error Tolerance: Schrödinger Equation with $p = 4 \ldots 11$

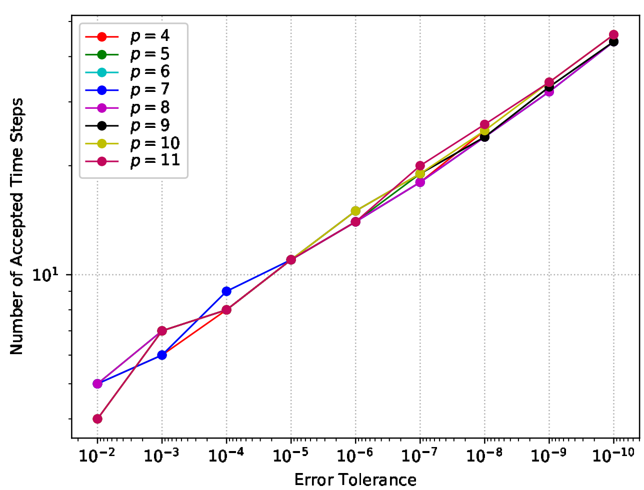

Figure 92: BACOLRI/LE Number of Accepted Time Steps vs. Error Tolerance: Schrödinger Equation with $p = 4 \ldots 11$

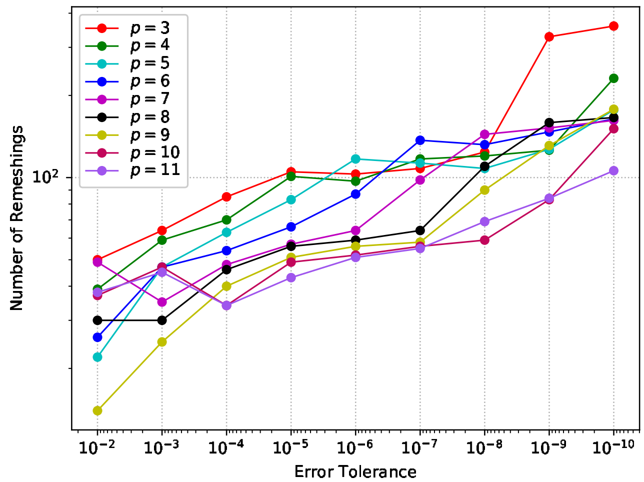

Figure 93: BACOLR/ST Number of Remeshings vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 3 \ldots 11$

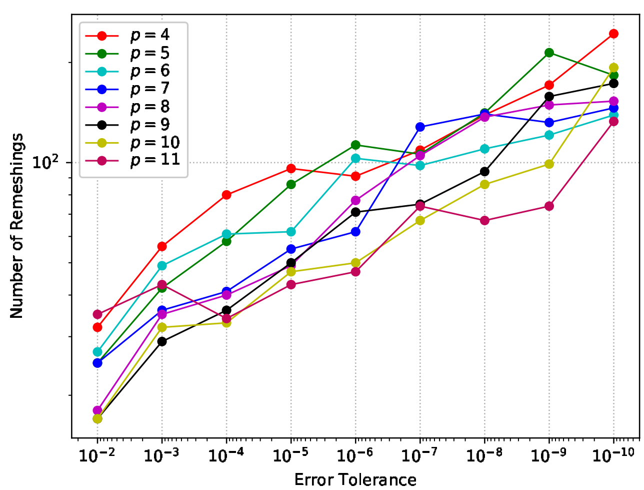

Figure 94: BACOLRI/ST Number of Remeshings vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

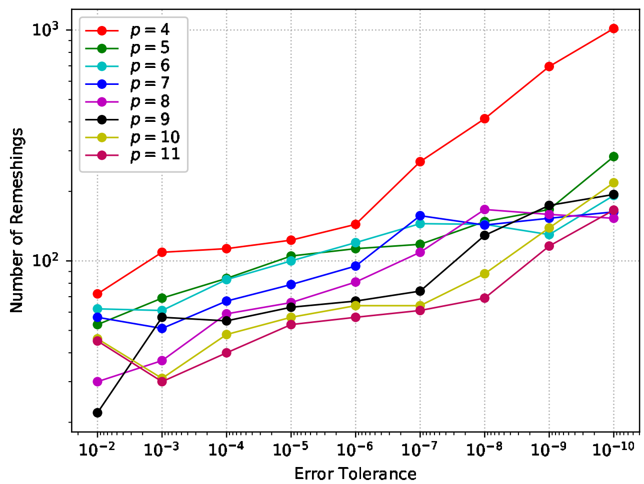

Figure 95: BACOLRI/LE Number of Remeshings vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

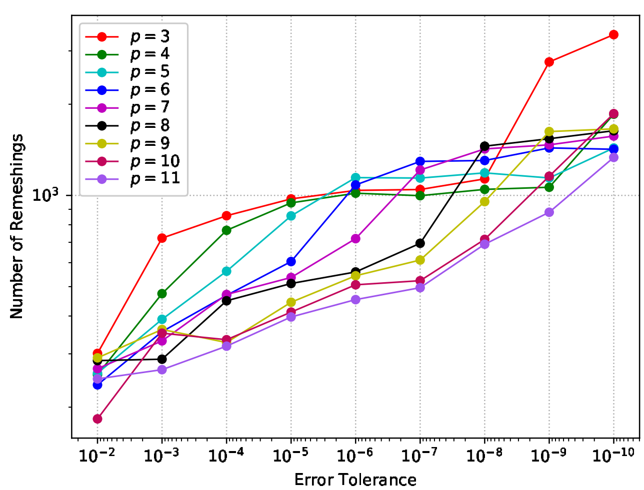

Figure 96: BACOLR/ST Number of Remeshings vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 3 \ldots 11$

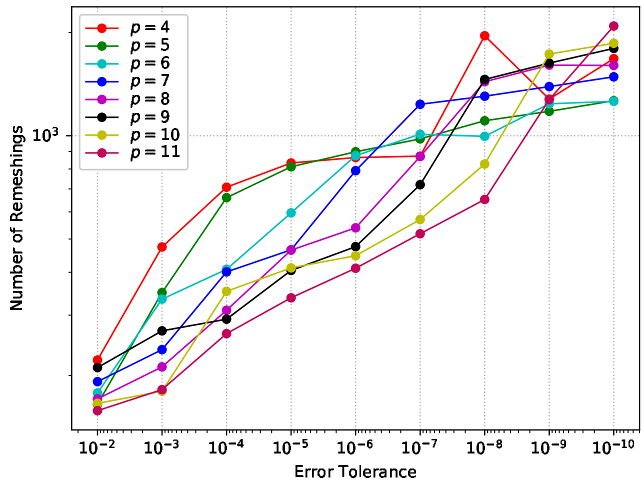

Figure 97: BACOLRI/ST Number of Remeshings vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

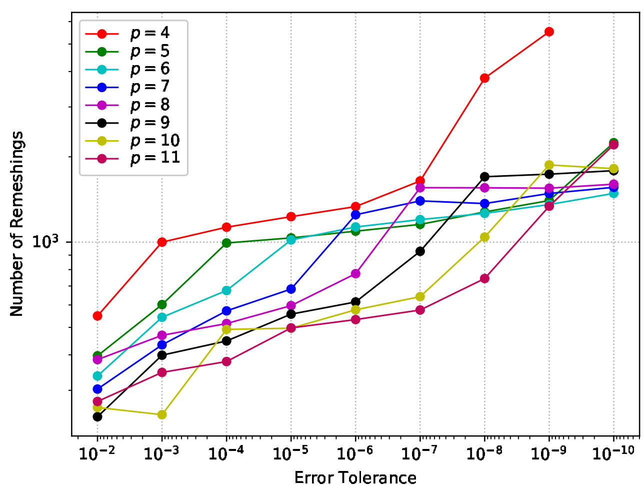

Figure 98: BACOLRI/LE Number of Remeshings vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

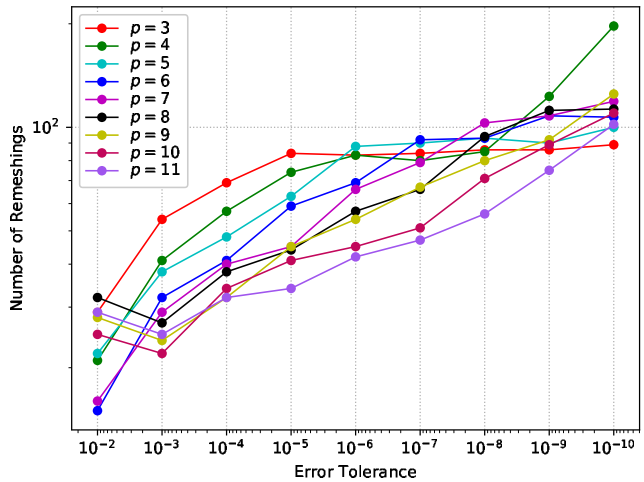

Figure 99: BACOLR/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 3 \ldots 11$

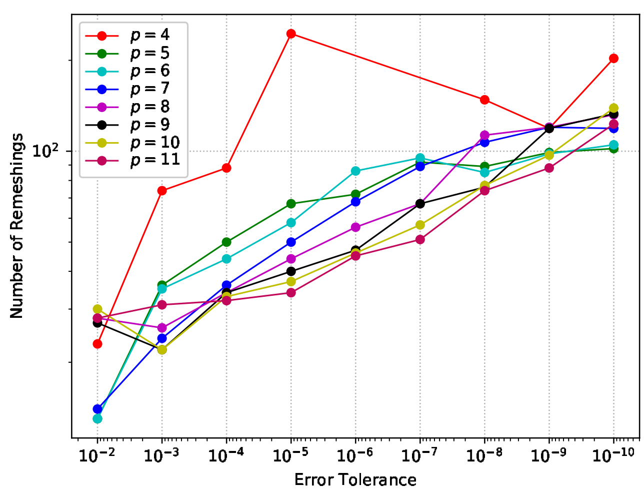

Figure 100: BACOLRI/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

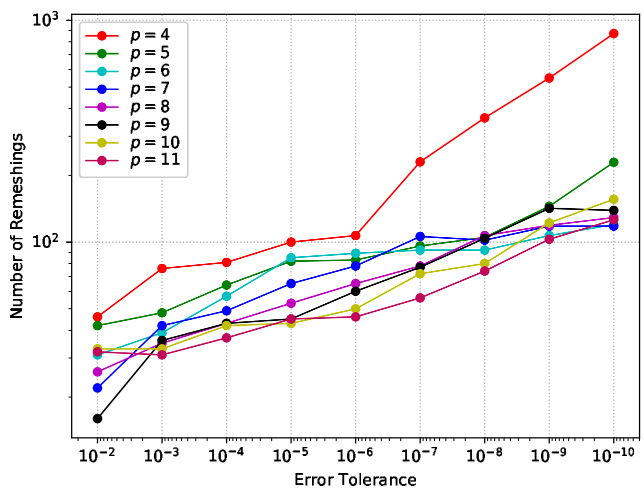

Figure 101: BACOLRI/LE Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

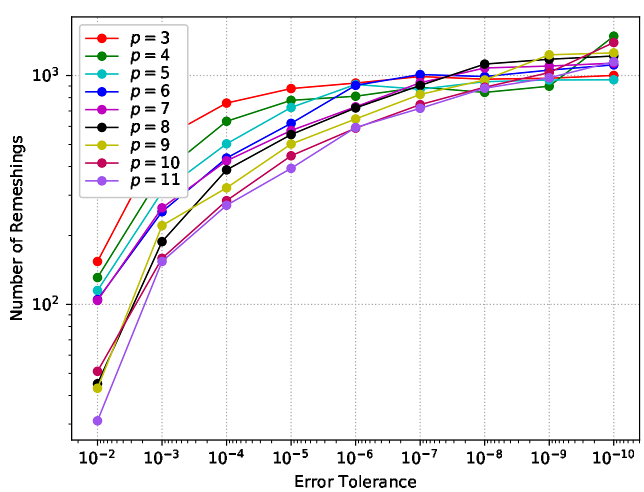

Figure 102: BACOLR/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 3 \ldots 11$

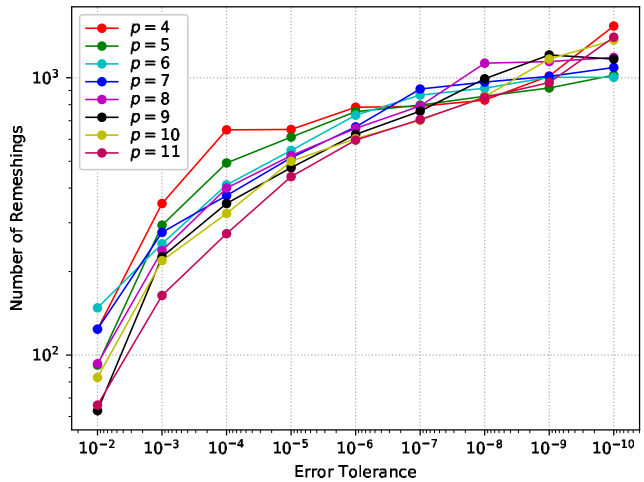

Figure 103: BACOLRI/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

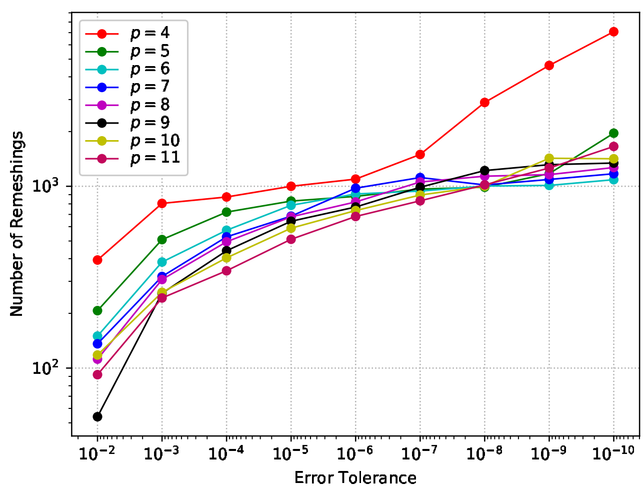

Figure 104: BACOLRI/LE Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

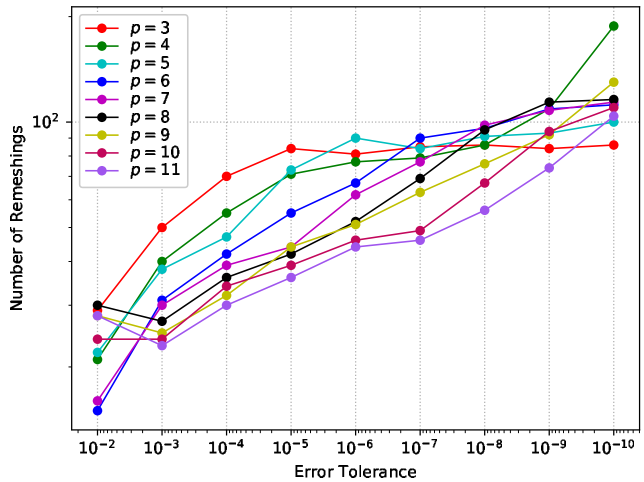

Figure 105: BACOLR/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 3 \ldots 11$

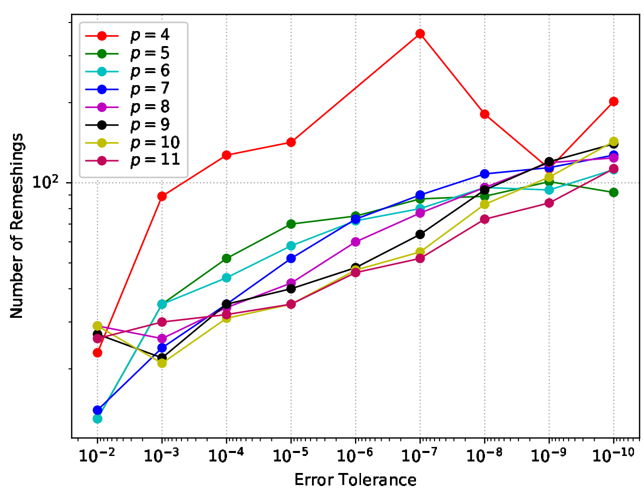

Figure 106: BACOLRI/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

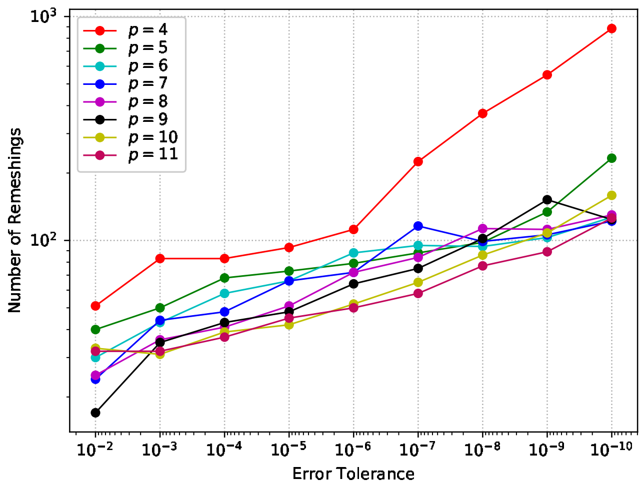

Figure 107: BACOLRI/LE Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

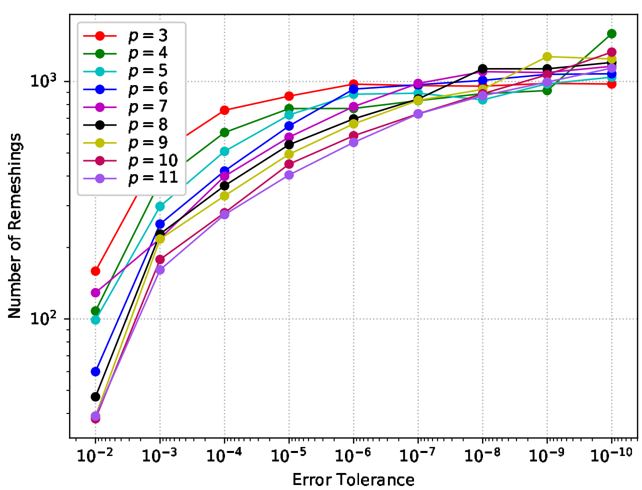

Figure 108: BACOLR/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 3 \ldots 11$

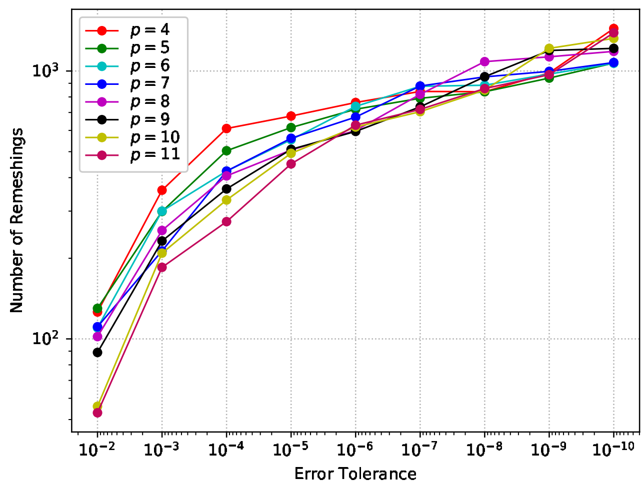

Figure 109: BACOLRI/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

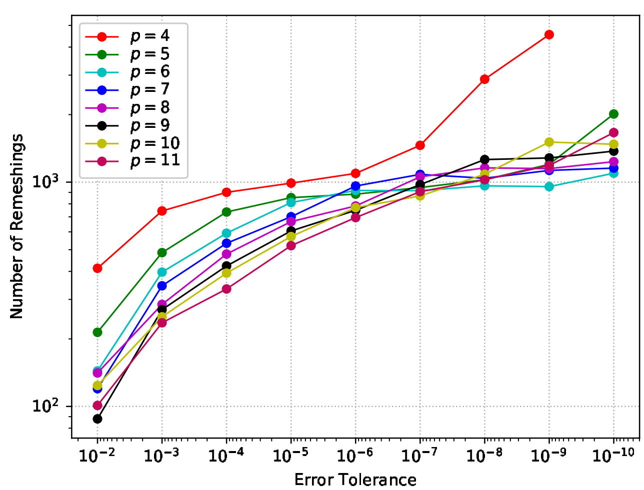

Figure 110: BACOLRI/LE Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

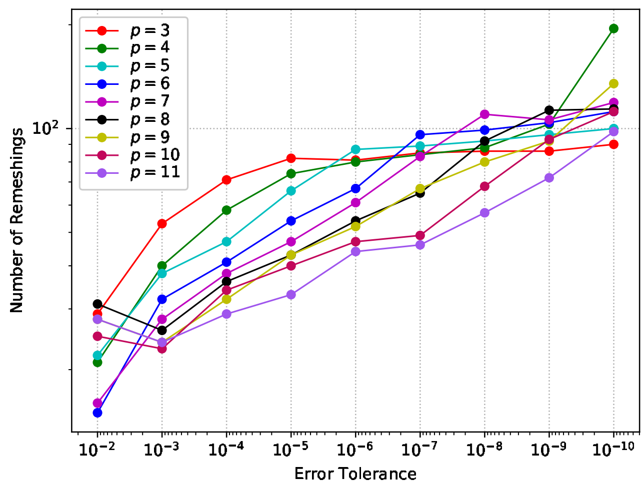

Figure 111: BACOLR/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 3 \ldots 11$

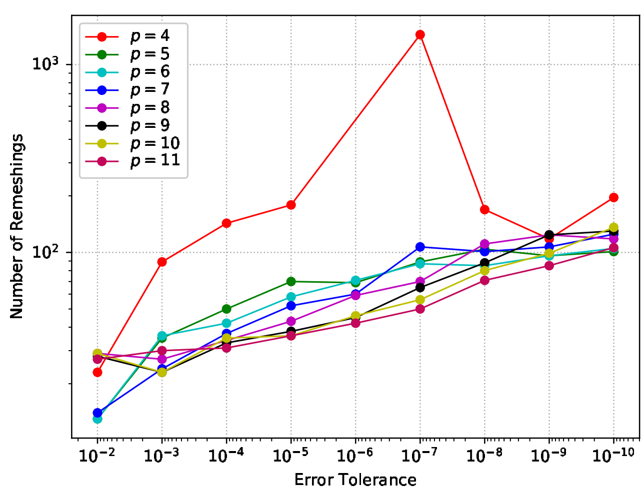

Figure 112: BACOLRI/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

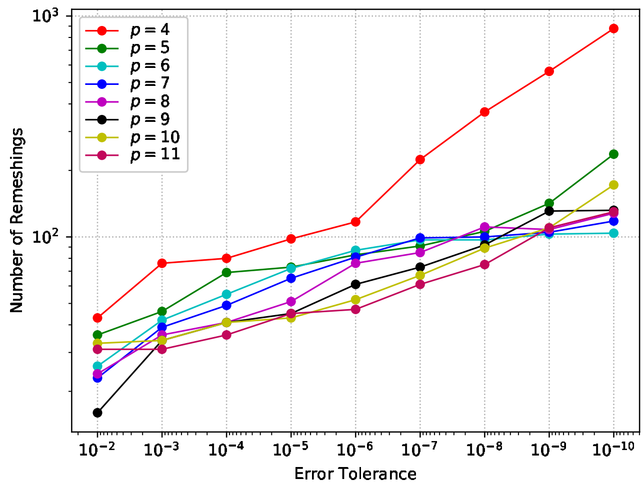

Figure 113: BACOLRI/LE Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

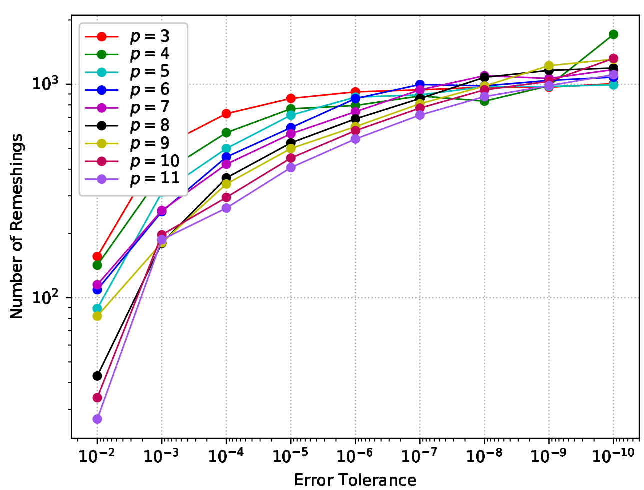

Figure 114: BACOLR/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 3 \ldots 11$

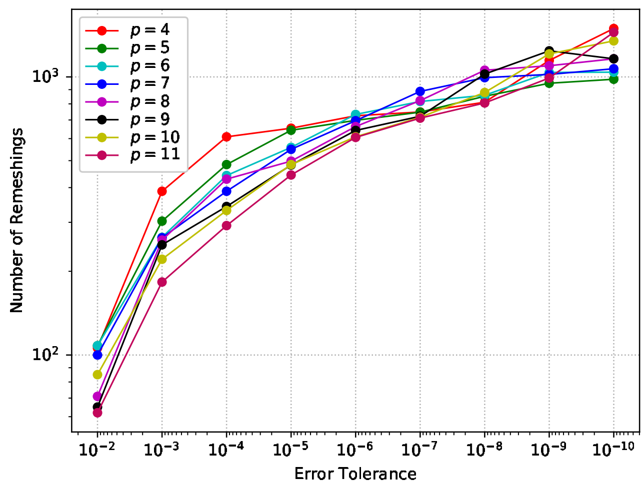

Figure 115: BACOLRI/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

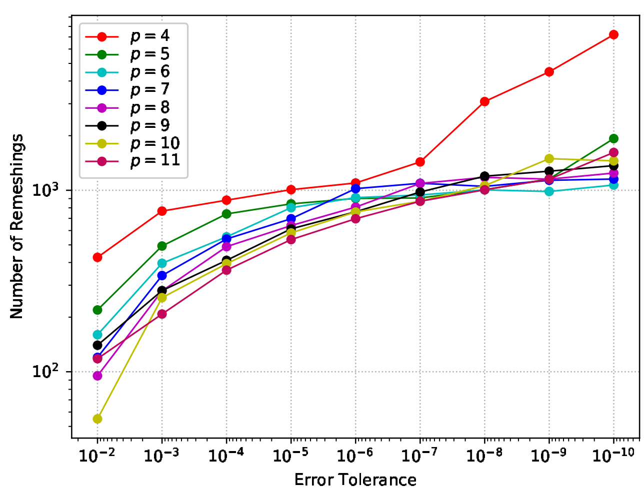

Figure 116: BACOLRI/LE Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

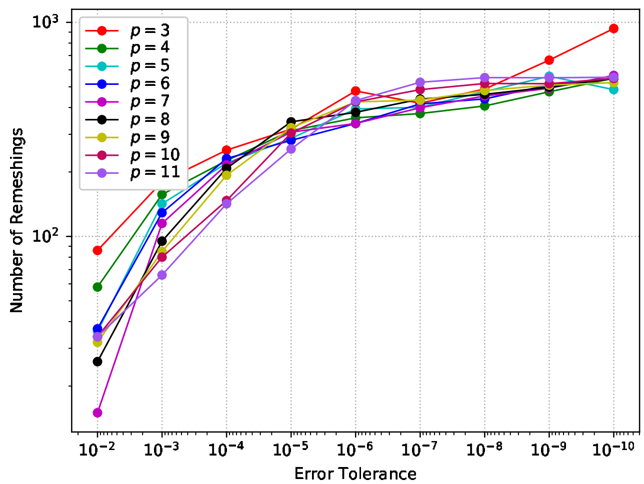

Figure 117: BACOLR/ST Number of Remeshings vs. Error Tolerance: Catalytic Surface Reaction Model with $p = 3 \dots 11$

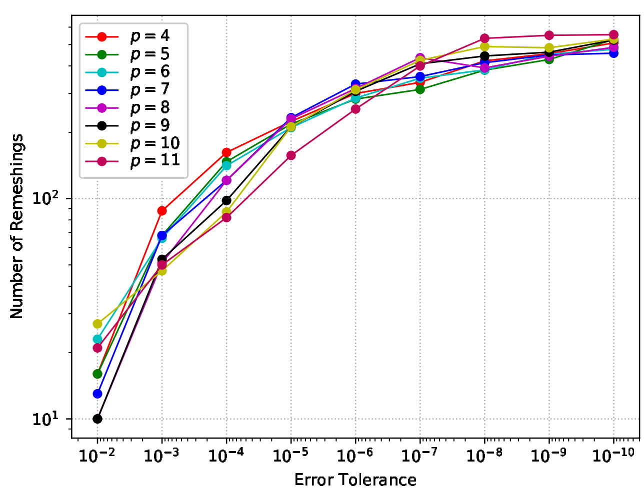

Figure 118: BACOLRI/ST Number of Remeshings vs. Error Tolerance: Catalytic Surface Reaction Model with $p = 4 \dots 11$

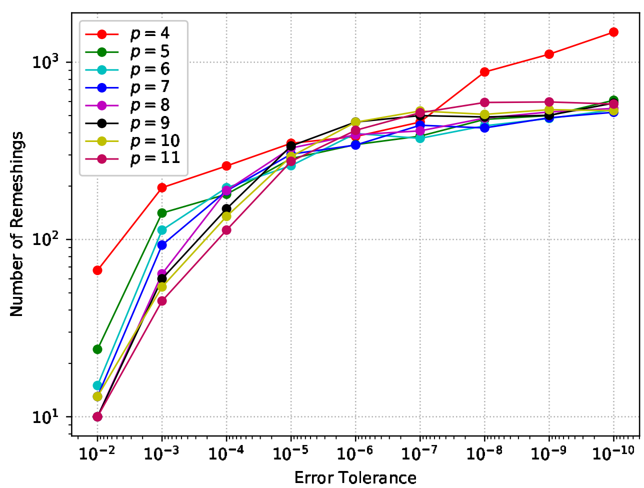

Figure 119: BACOLRI/LE Number of Remeshings vs. Error Tolerance: Catalytic Surface Reaction Model with $p = 4 \ldots 11$

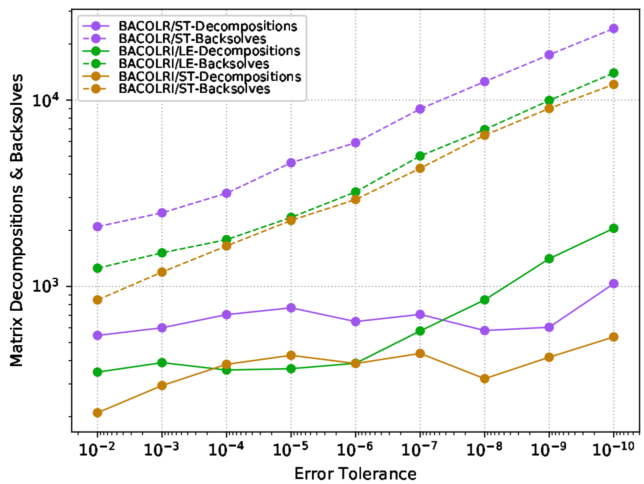

Figure 120: Number of Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4$

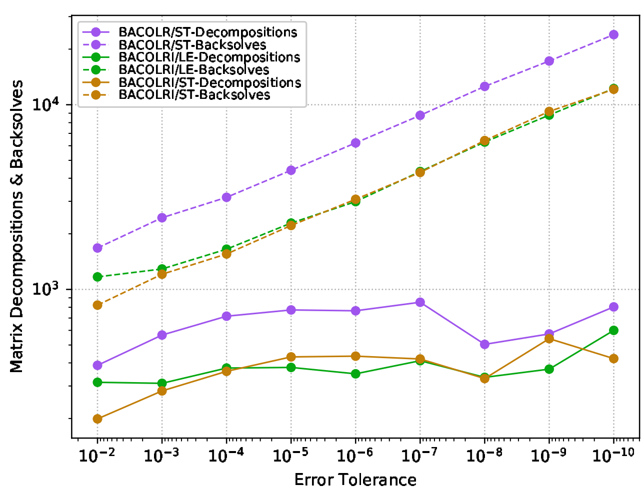

Figure 121: Number of Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 5$

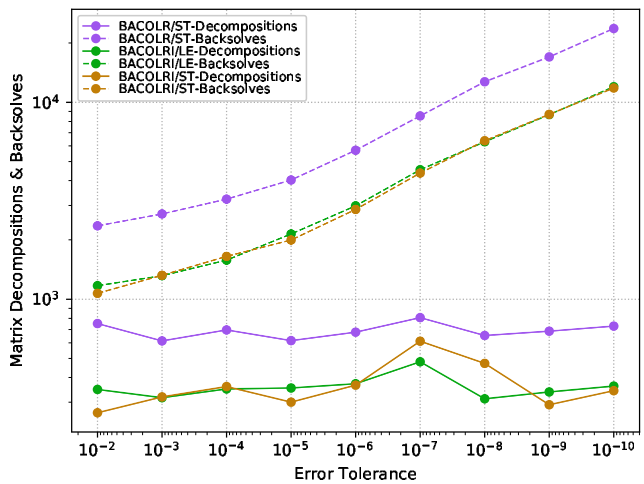

Figure 122: Number of Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 7$

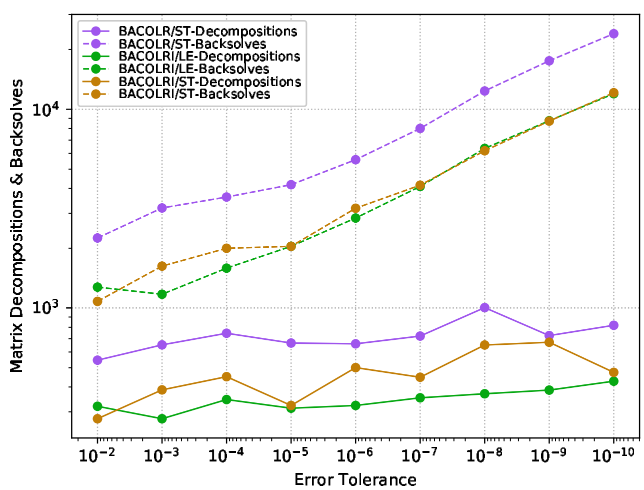

Figure 123: Number of Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 9$

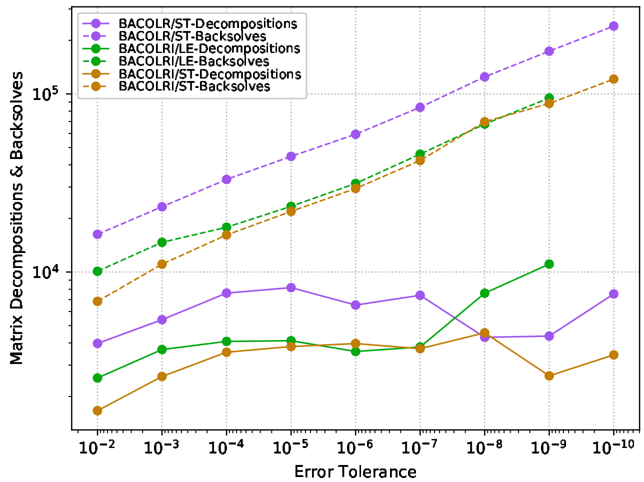

Figure 124: Number of Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4$

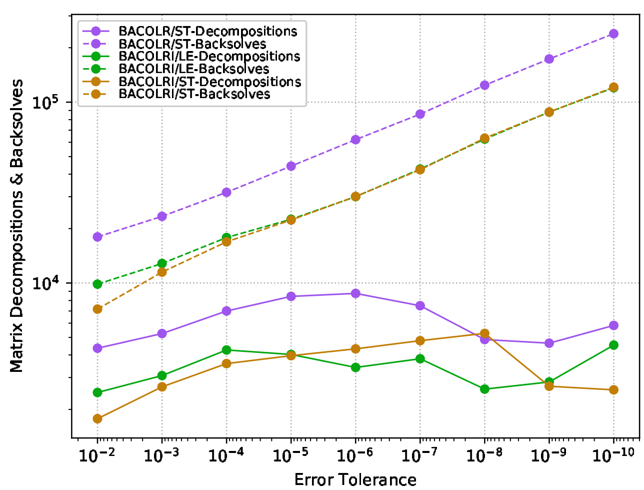

Figure 125: Number of Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 5$

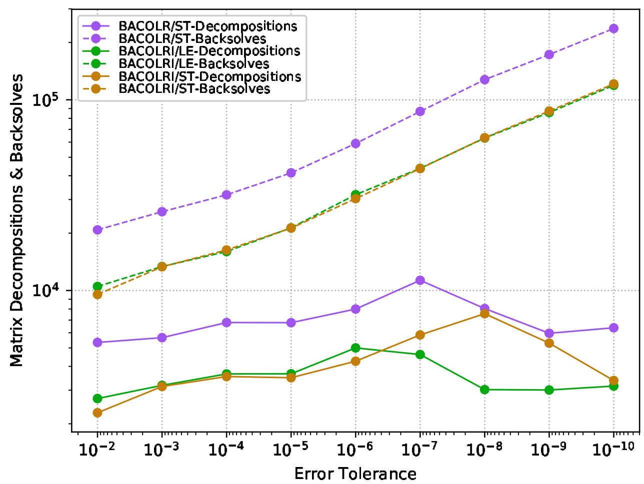

Figure 126: Number of Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 7$

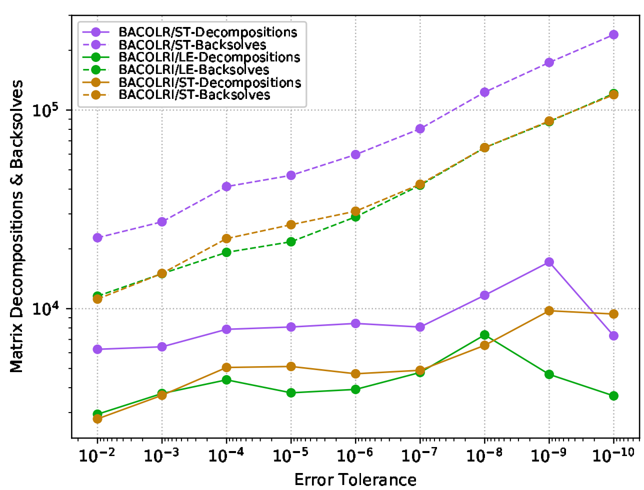

Figure 127: Number of Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 9$

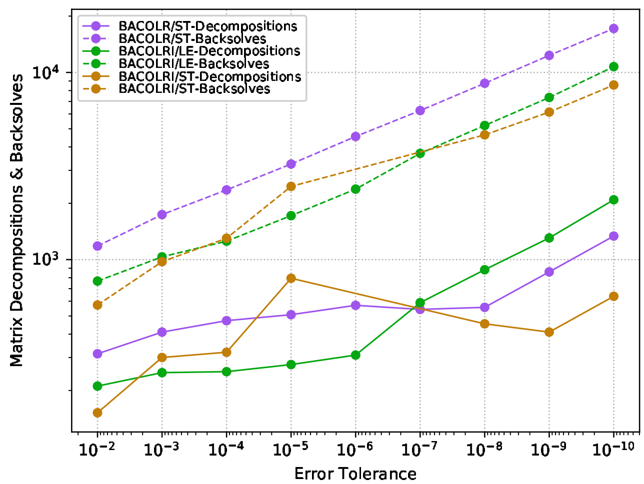

Figure 128: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4$

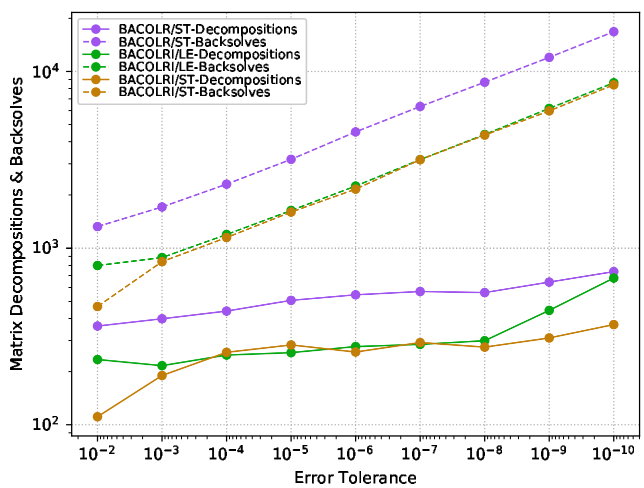

Figure 129: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 5$

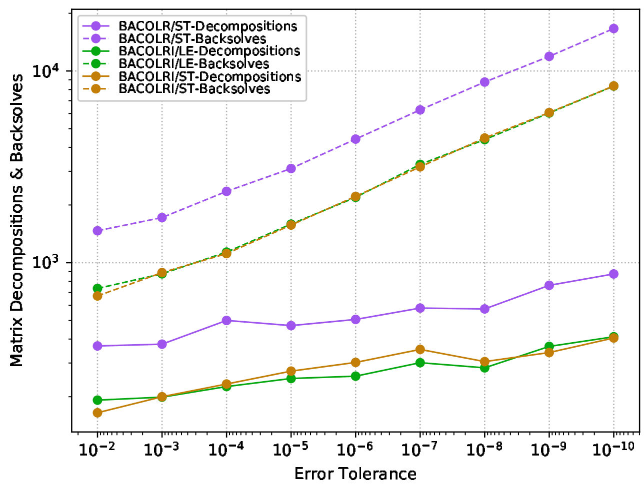

Figure 130: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 7$

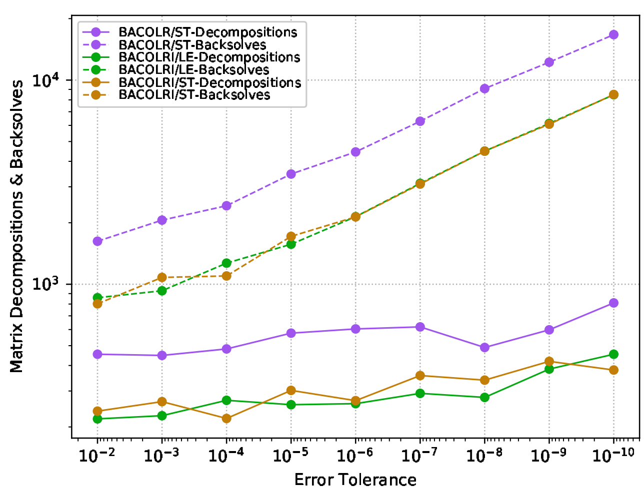

Figure 131: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 9$

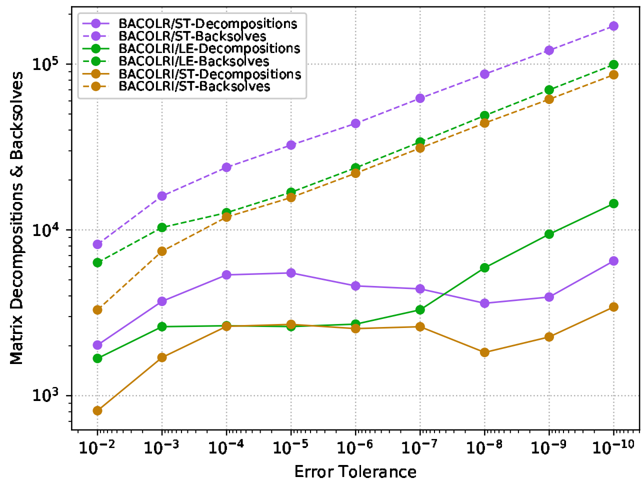

Figure 132: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4$

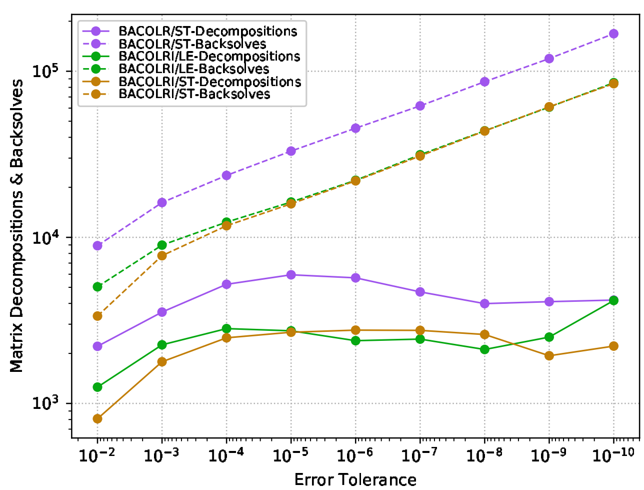

Figure 133: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 5$

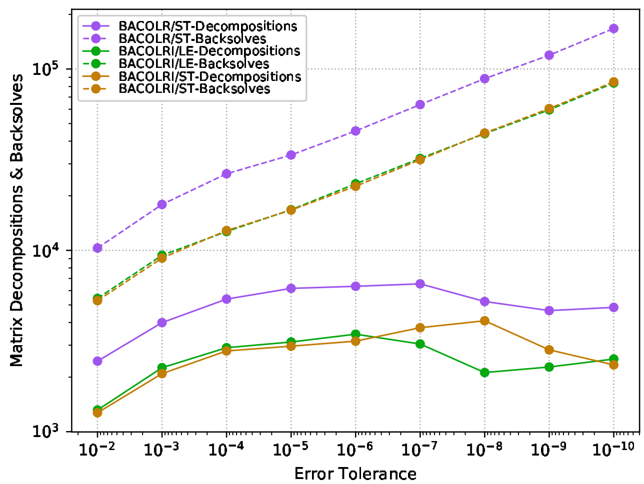

Figure 134: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 7$

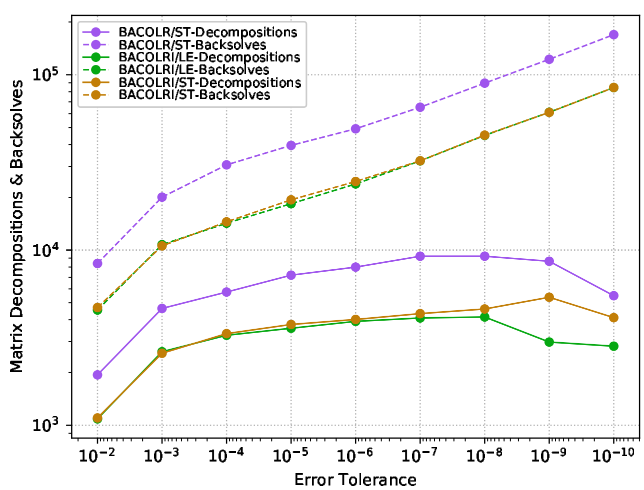

Figure 135: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 9$

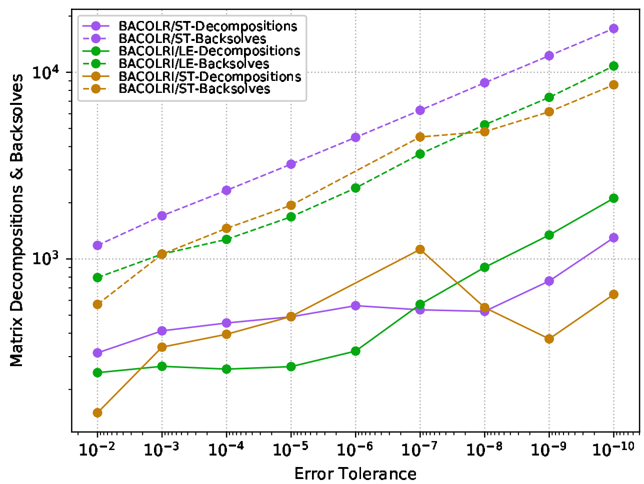

Figure 136: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 4$

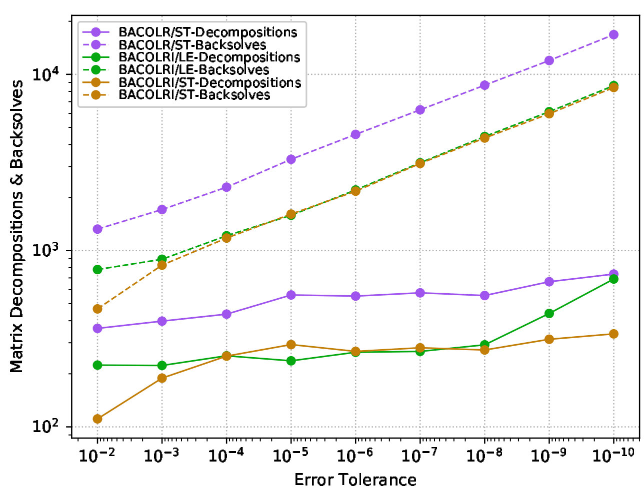

Figure 137: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 5$

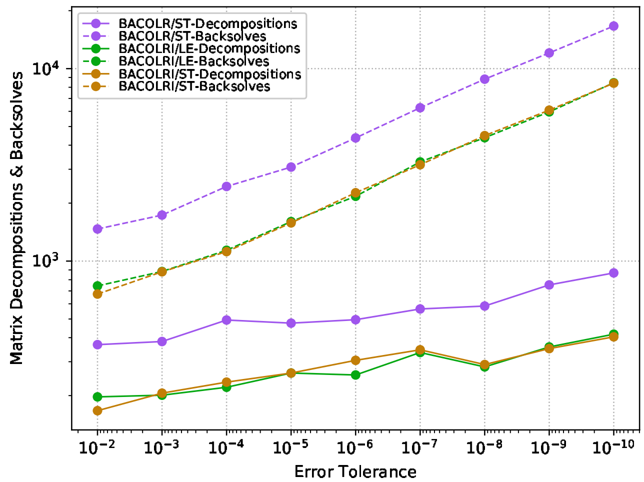

Figure 138: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 7$

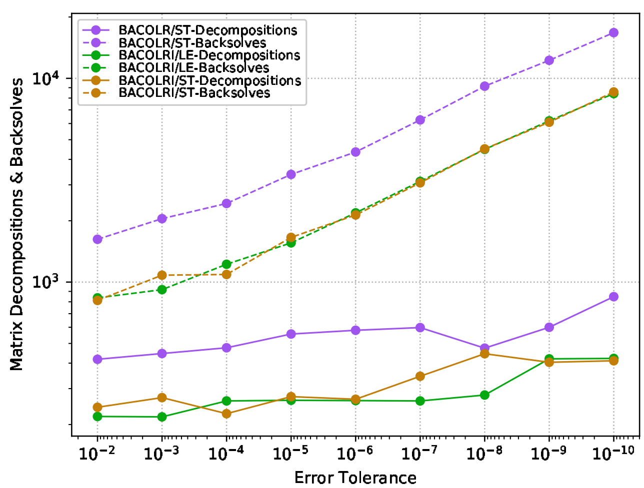

Figure 139: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 9$

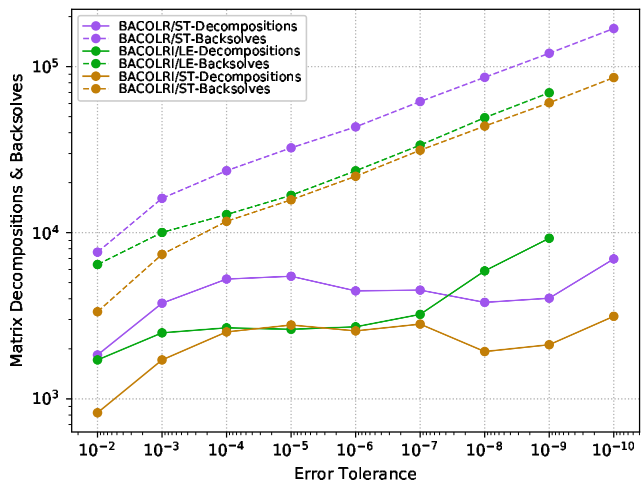

Figure 140: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 4$

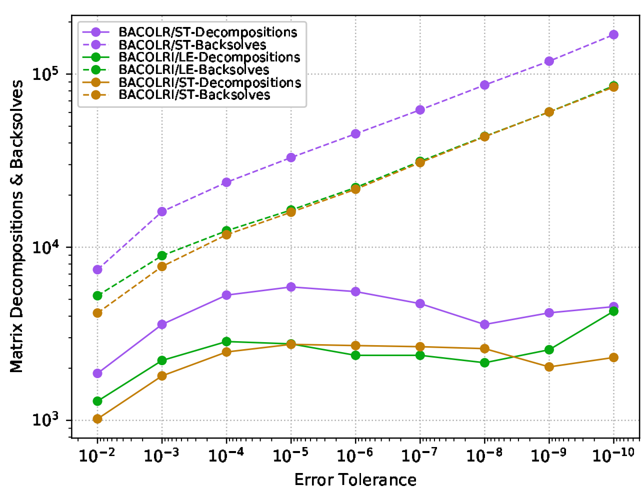

Figure 141: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 5$

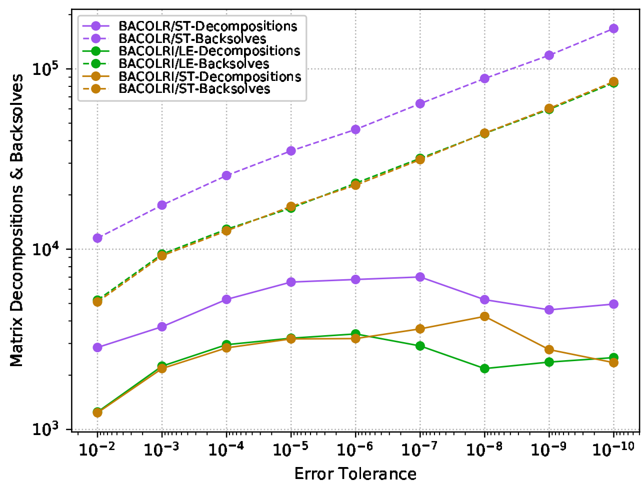

Figure 142: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 7$

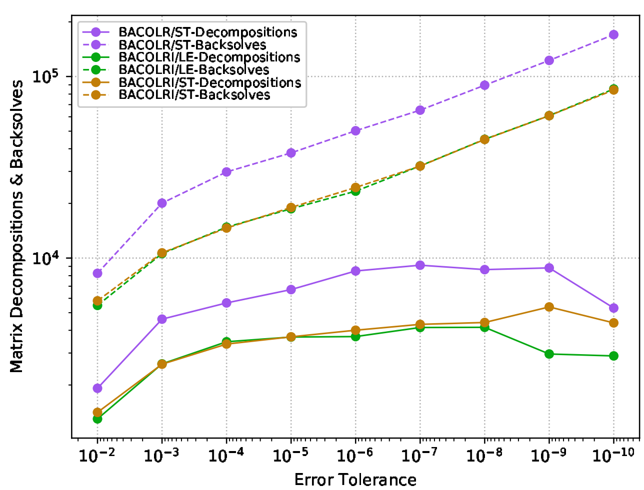

Figure 143: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 9$

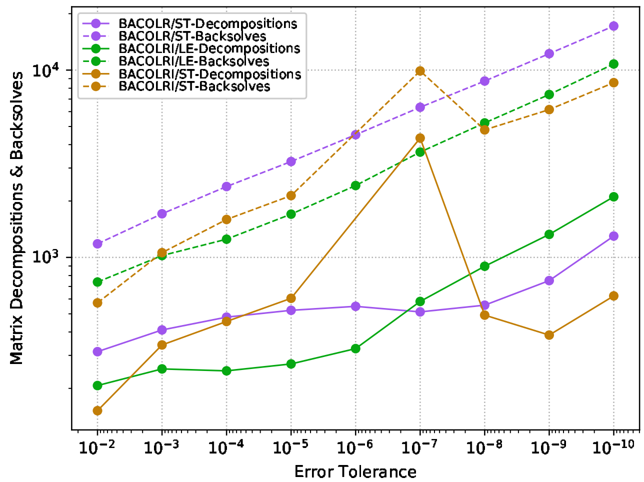

Figure 144: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 4$

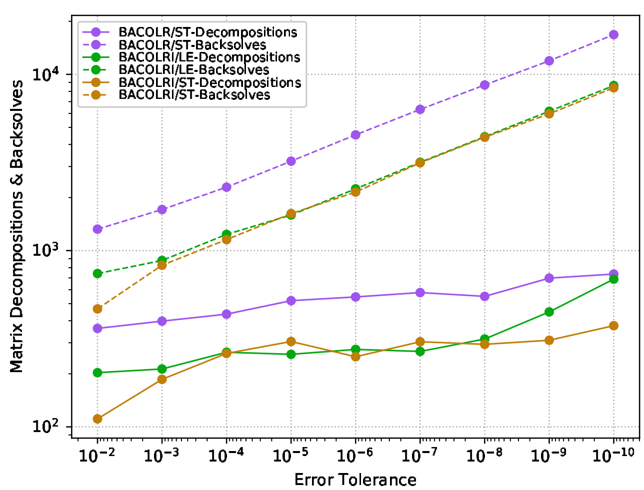

Figure 145: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 5$

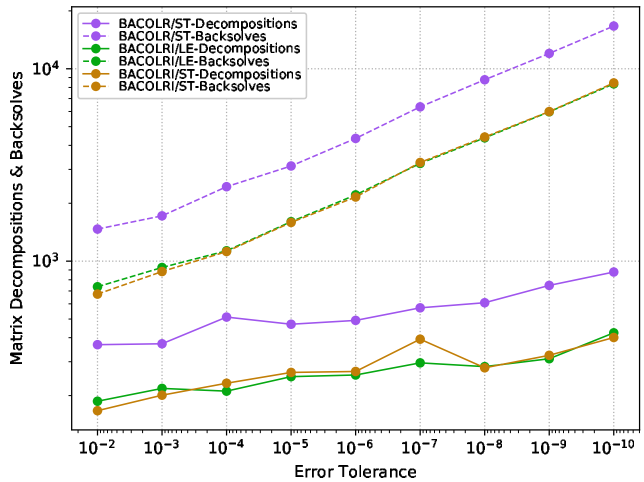

Figure 146: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 7$

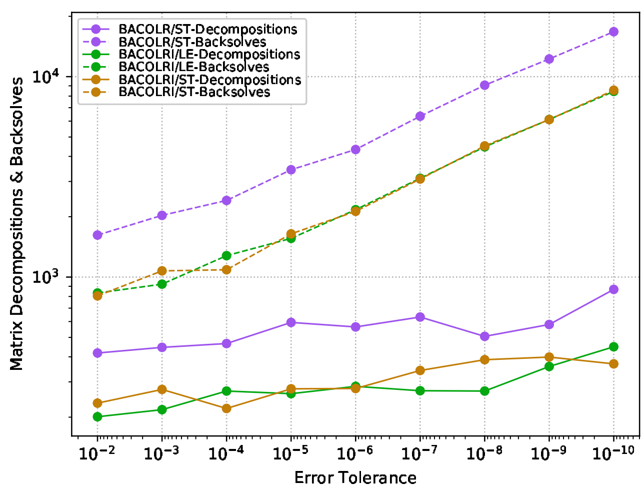

Figure 147: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 9$

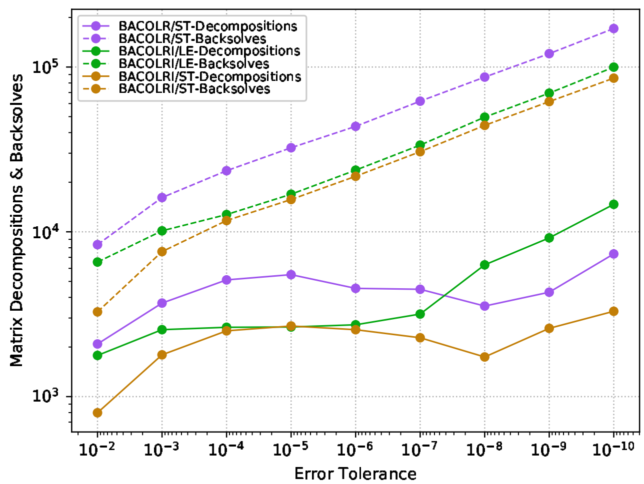

Figure 148: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 4$

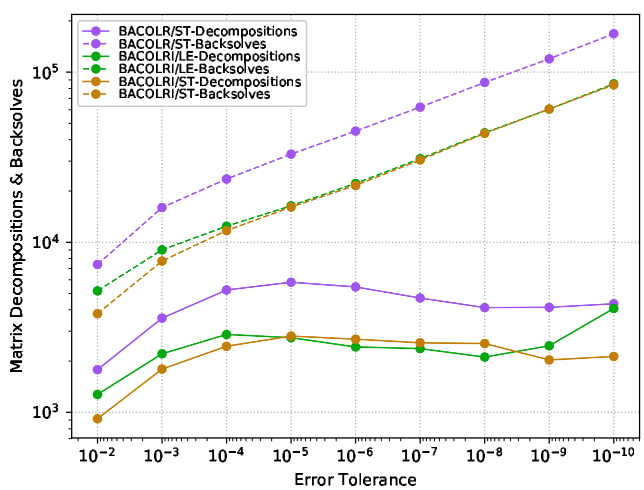

Figure 149: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 5$

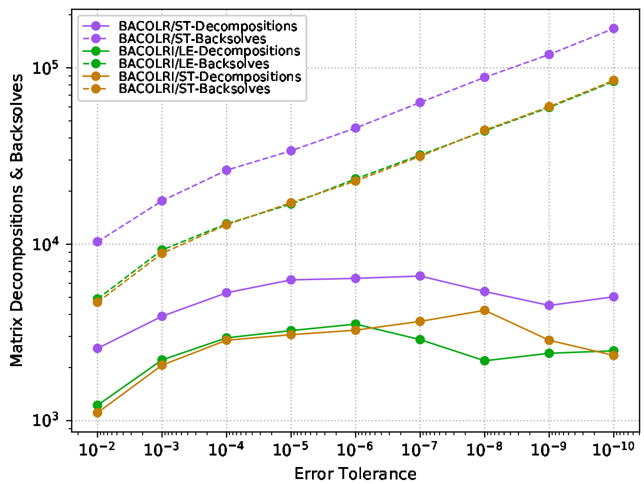

Figure 150: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 7$

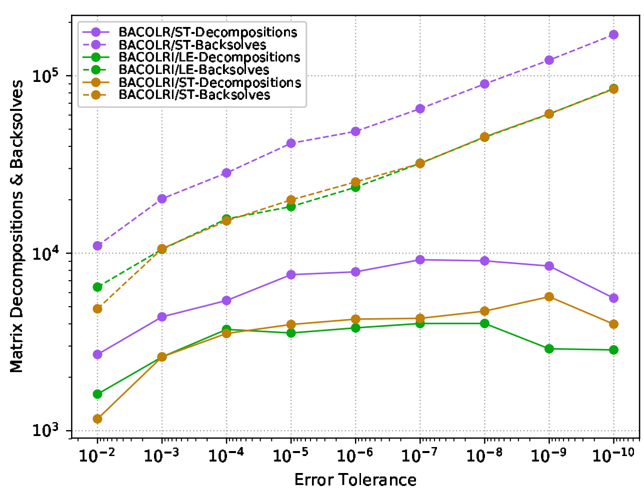

Figure 151: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 9$

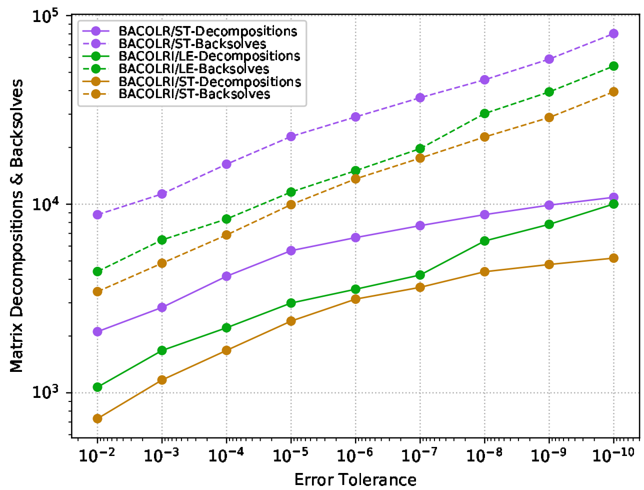

Figure 152: Number of Matrix Factorizations and Backsolves vs. Tolerance: Catalytic Surface Reaction Model with $p = 4$

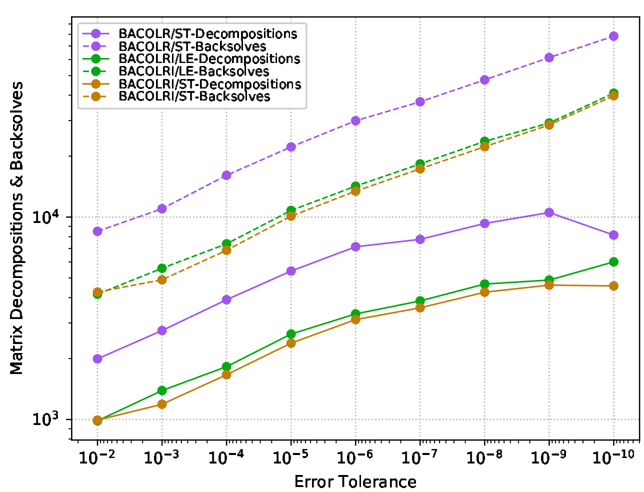

Figure 153: Number of Matrix Factorizations and Backsolves vs. Tolerance: Catalytic Surface Reaction Model with $p = 5$

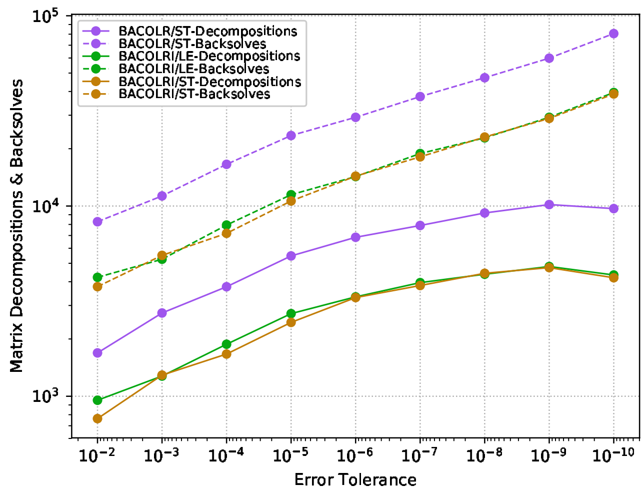

Figure 154: Number of Matrix Factorizations and Backsolves vs. Tolerance: Catalytic Surface Reaction Model with $p = 7$

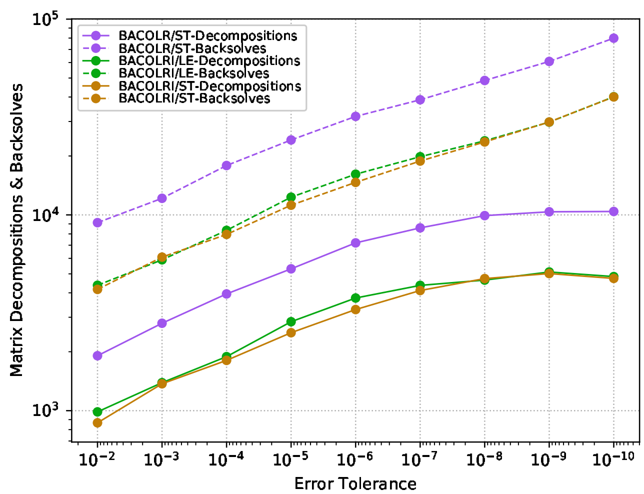

Figure 155: Number of Matrix Factorizations and Backsolves vs. Tolerance: Catalytic Surface Reaction Model with $p = 9$

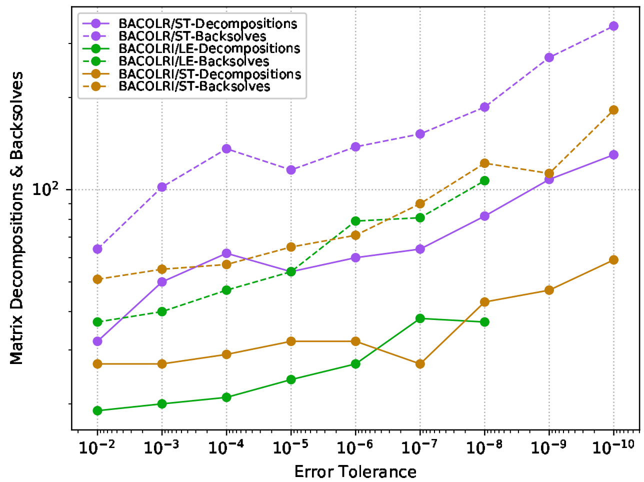

Figure 156: Number of Matrix Factorizations and Backsolves vs. Tolerance: Schrödinger Equation with $p = 4$

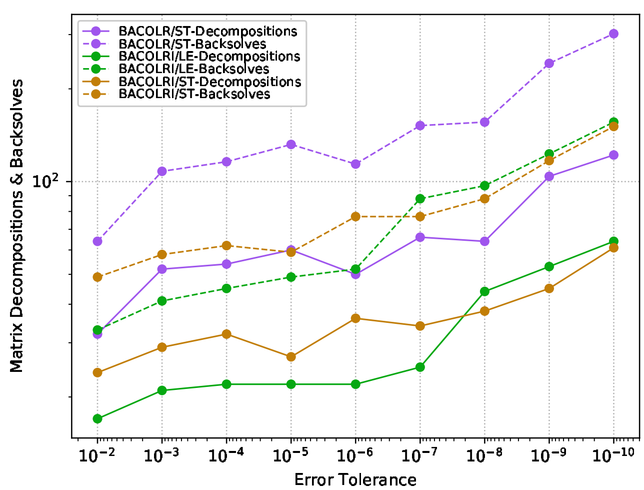

Figure 157: Number of Matrix Factorizations and Backsolves vs. Tolerance: Schrödinger Equation with $p = 5$

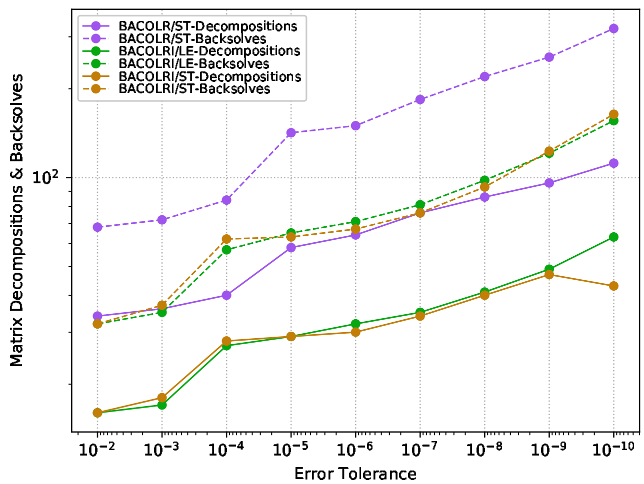

Figure 158: Number of Matrix Factorizations and Backsolves vs. Tolerance: Schrödinger Equation with $p = 7$

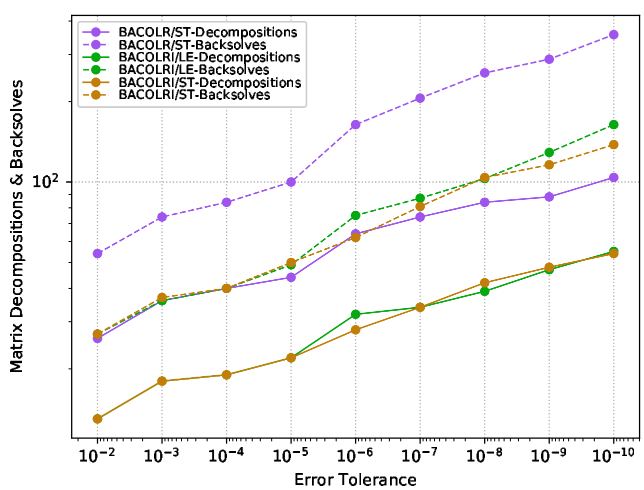

Figure 159: Number of Matrix Factorizations and Backsolves vs. Tolerance: Schrödinger Equation with $p = 9$

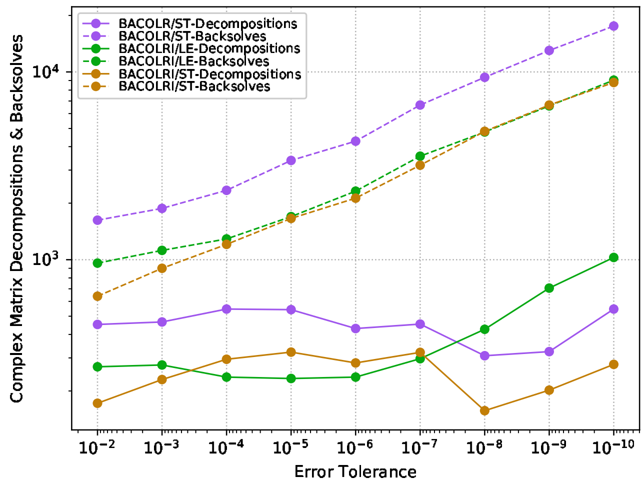

Figure 160: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4$

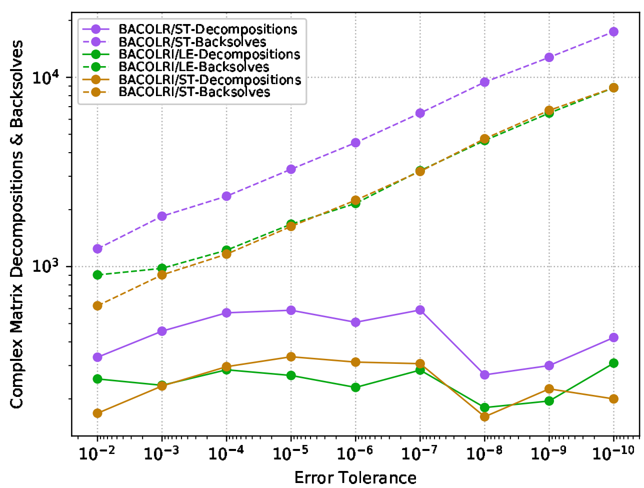

Figure 161: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 5$

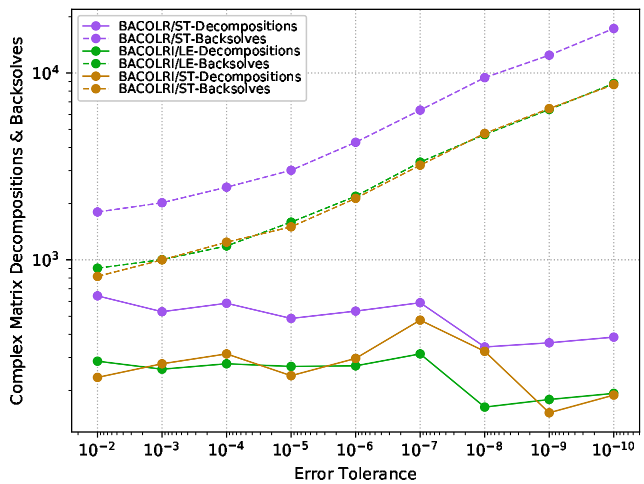

Figure 162: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 7$

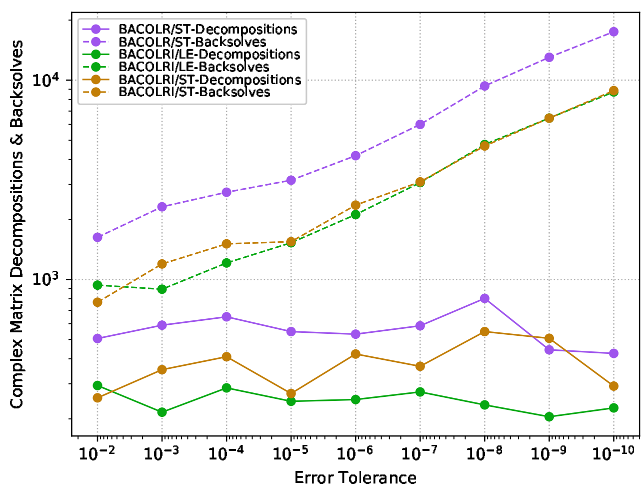

Figure 163: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 9$

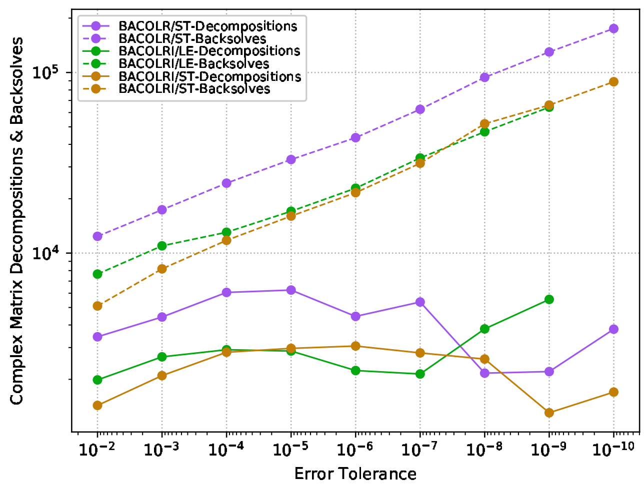

Figure 164: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4$

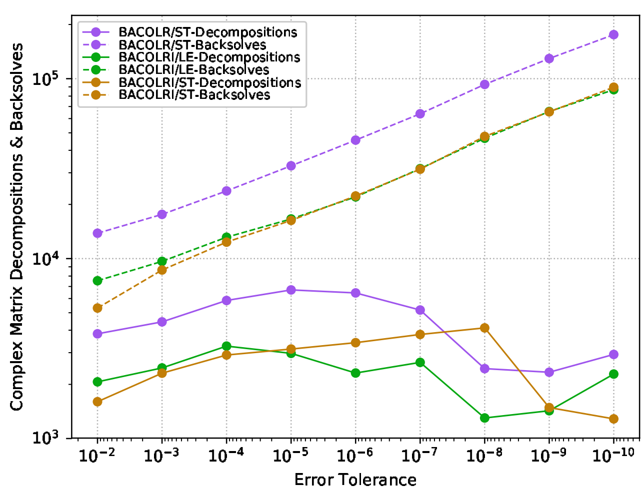

Figure 165: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 5$

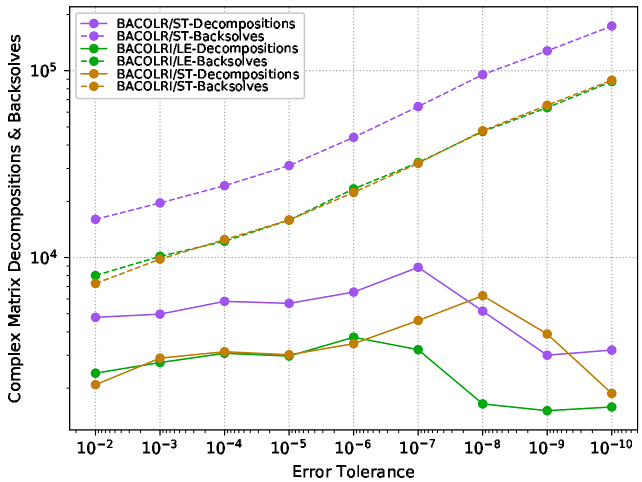

Figure 166: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 7$

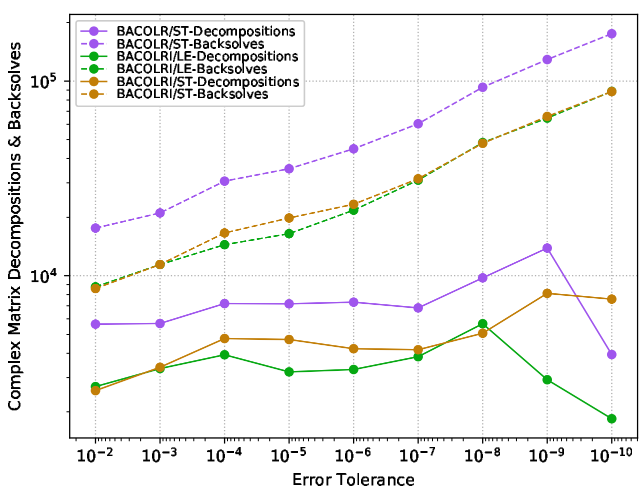

Figure 167: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 9$

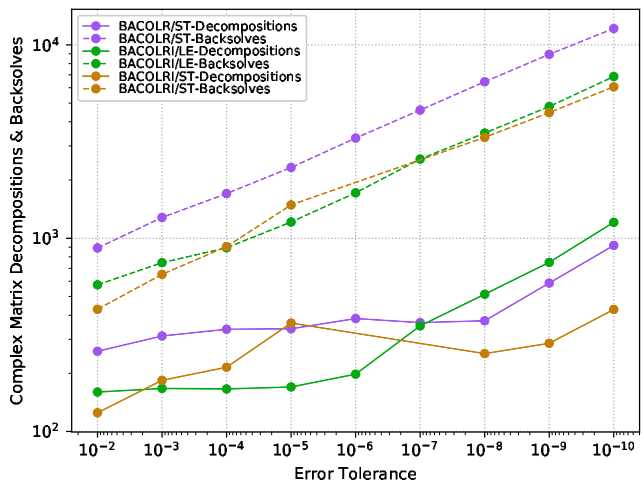

Figure 168: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4$

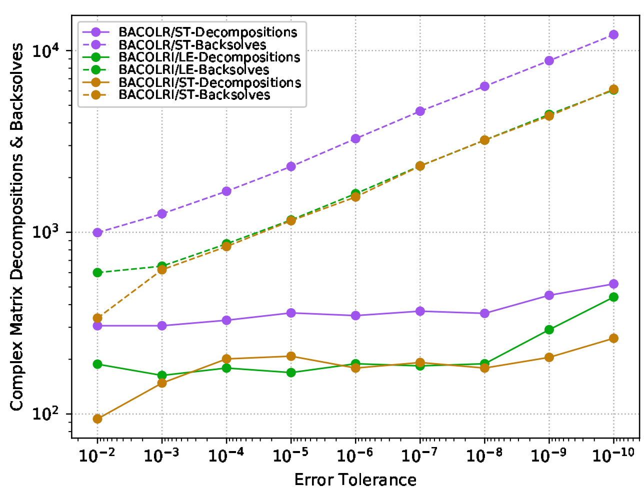

Figure 169: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 5$

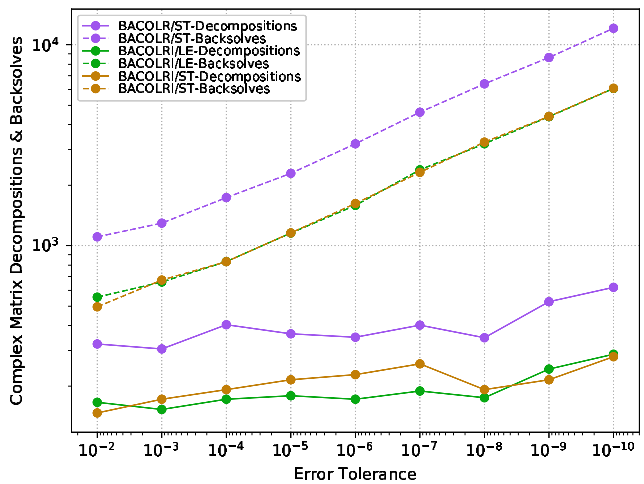

Figure 170: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 7$

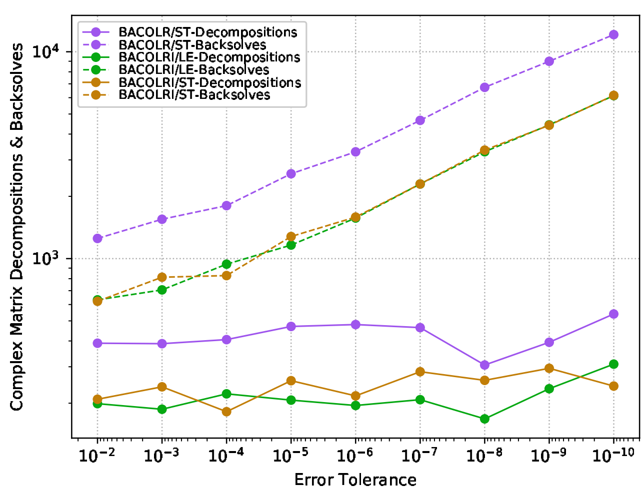

Figure 171: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 9$

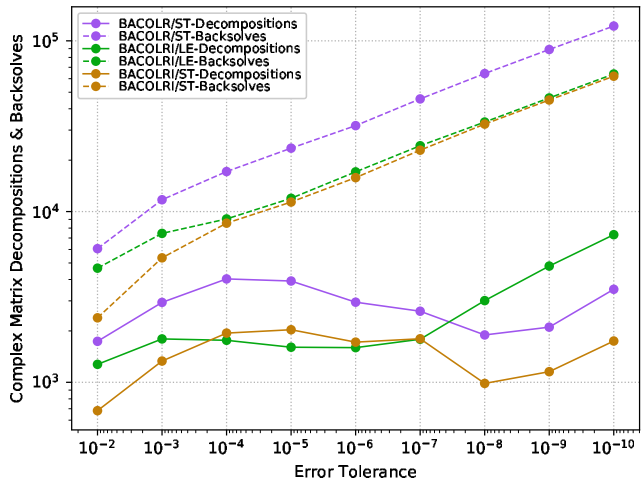

Figure 172: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4$

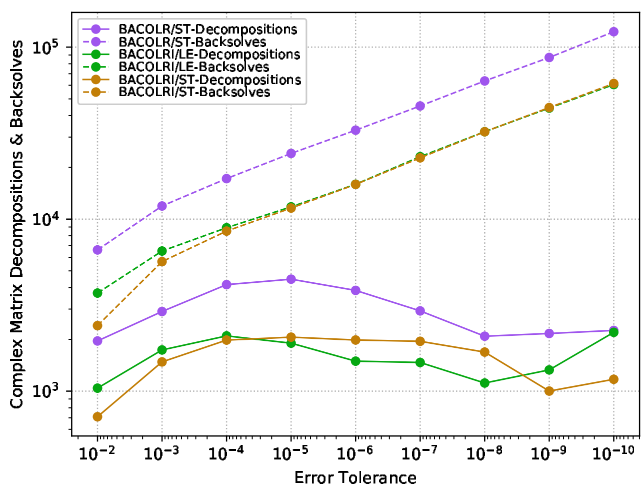

Figure 173: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 5$

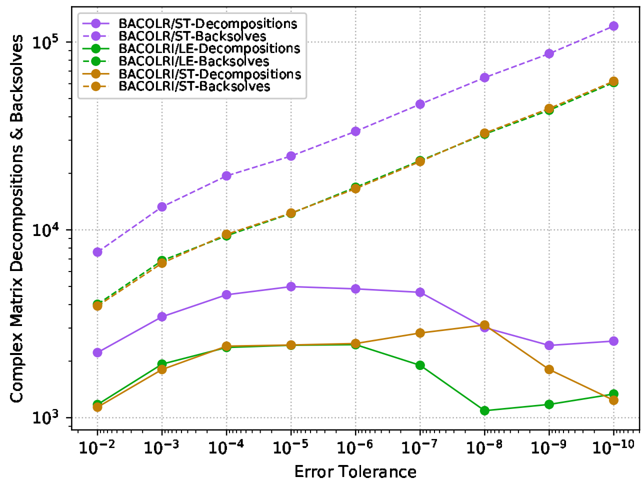

Figure 174: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 7$

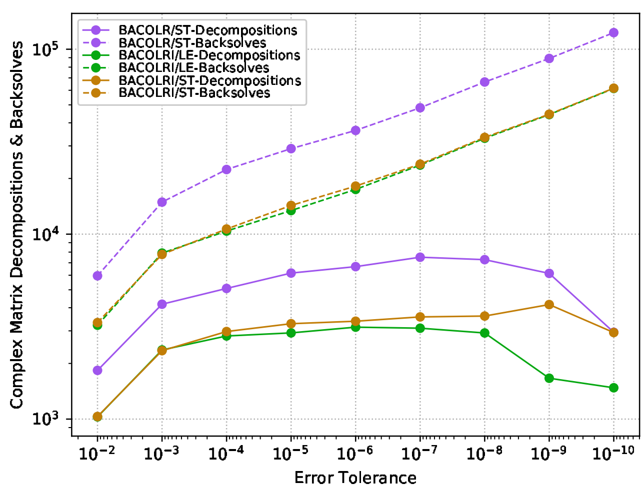

Figure 175: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 9$

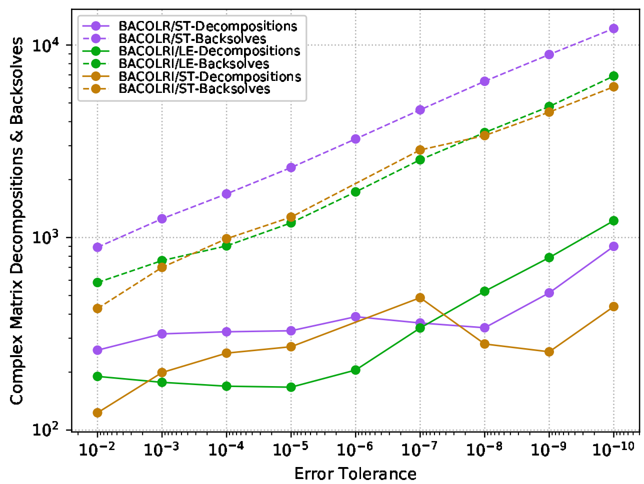

Figure 176: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 4$

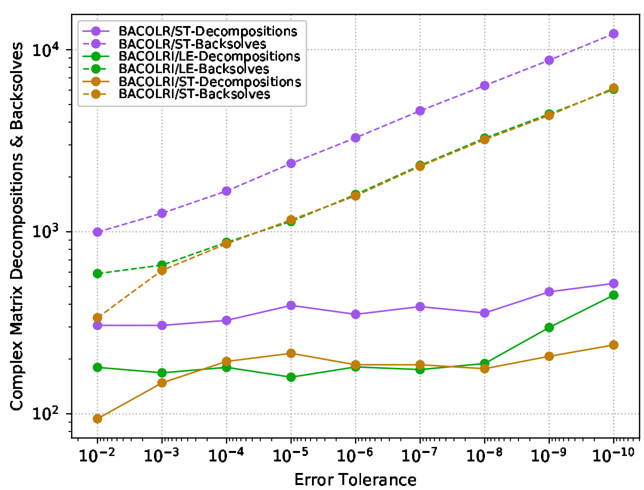

Figure 177: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 5$

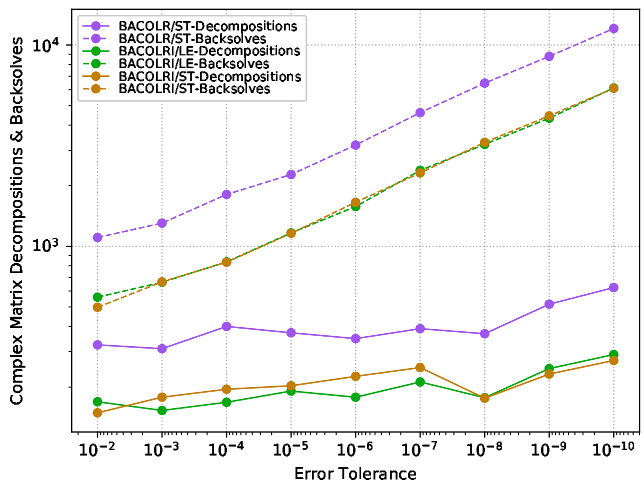

Figure 178: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 7$

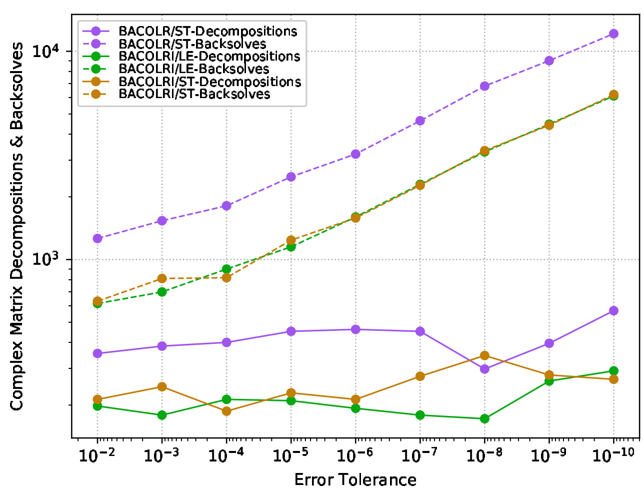

Figure 179: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 9$

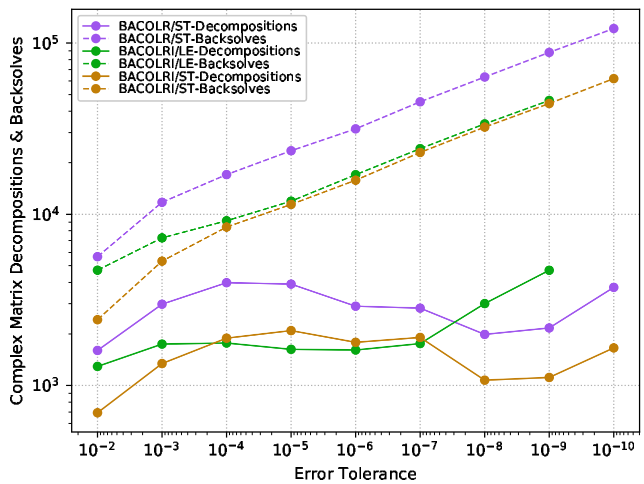

Figure 180: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 4$

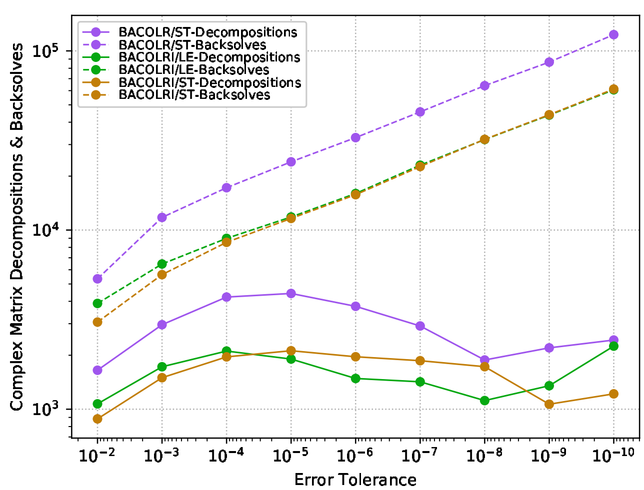

Figure 181: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 5$

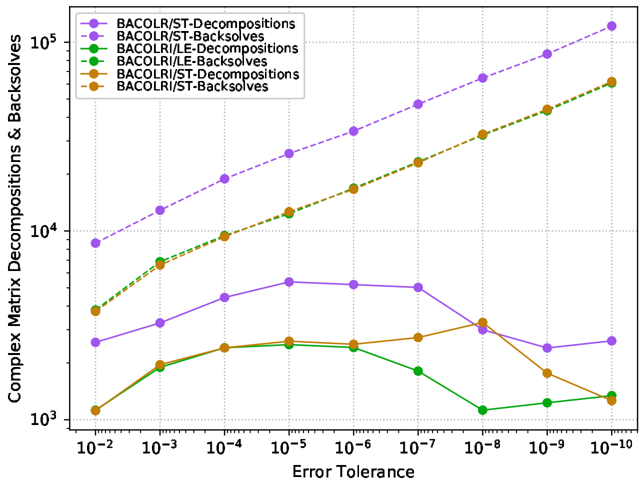

Figure 182: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 7$

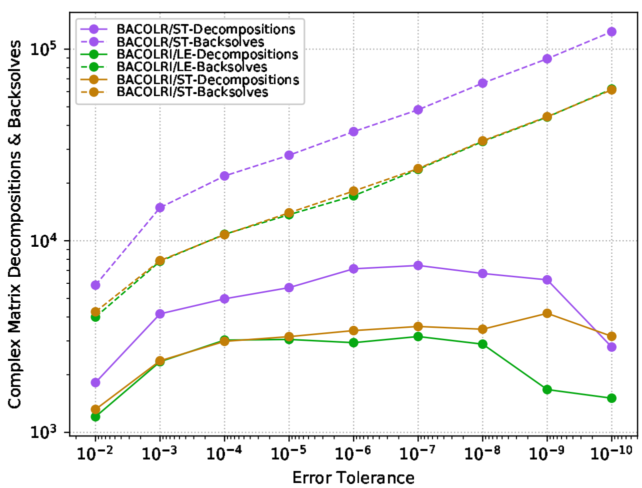

Figure 183: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 9$

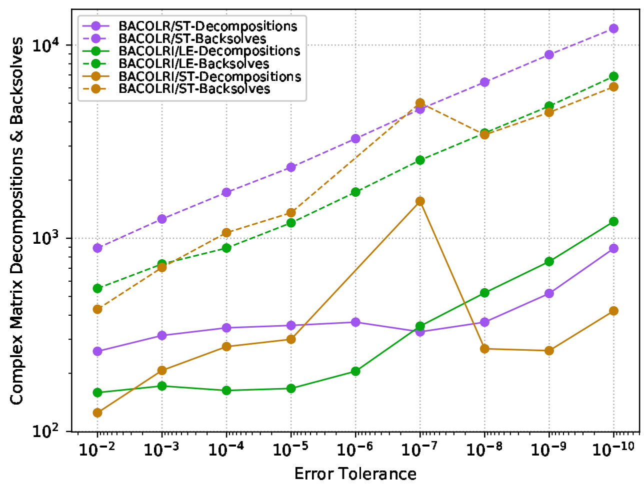

Figure 184: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 4$

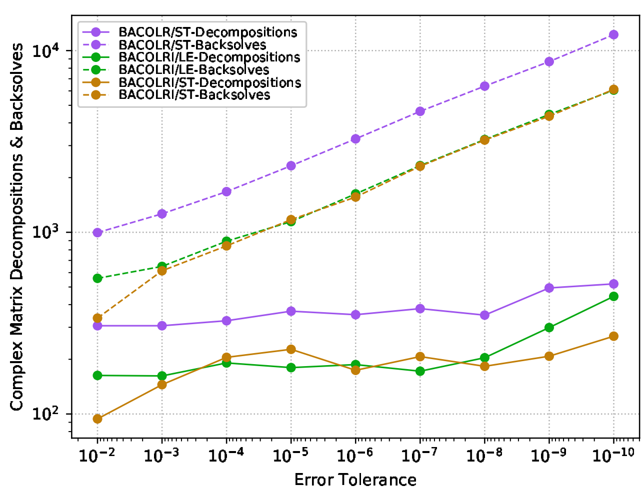

Figure 185: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 5$

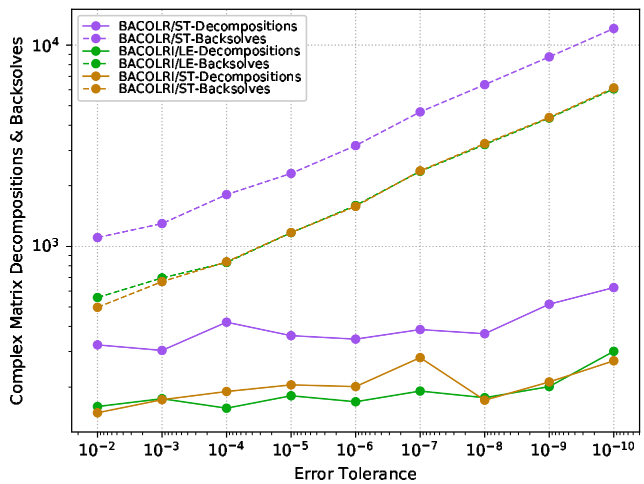

Figure 186: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 7$

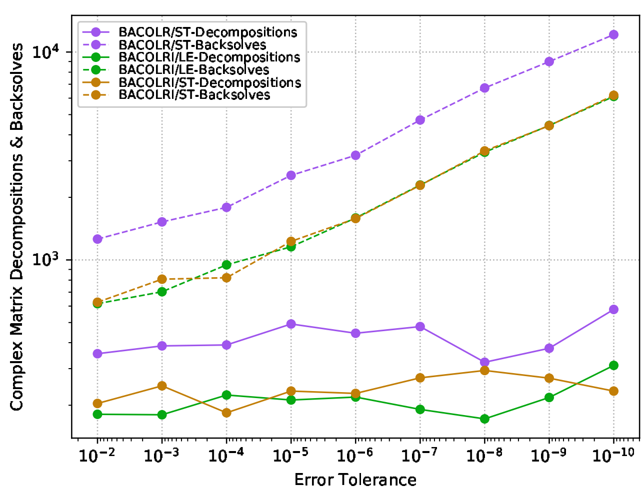

Figure 187: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 9$

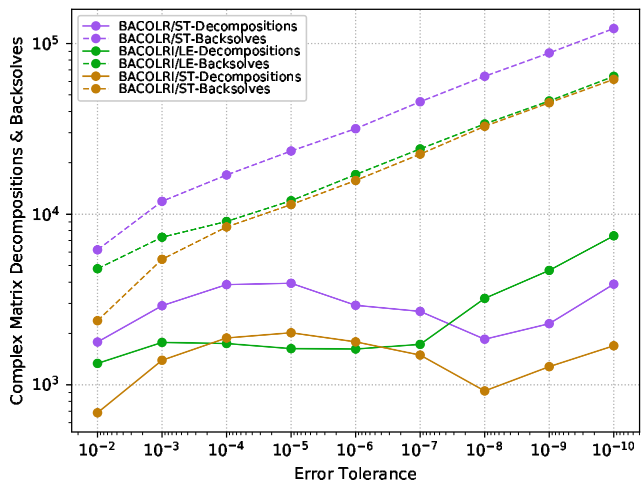

Figure 188: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 4$

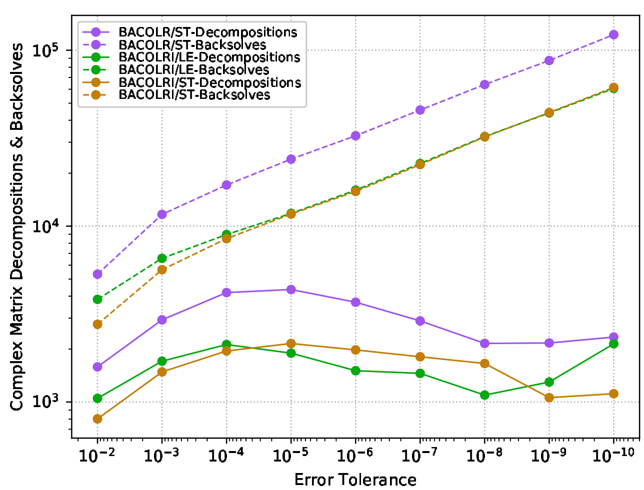

Figure 189: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 5$

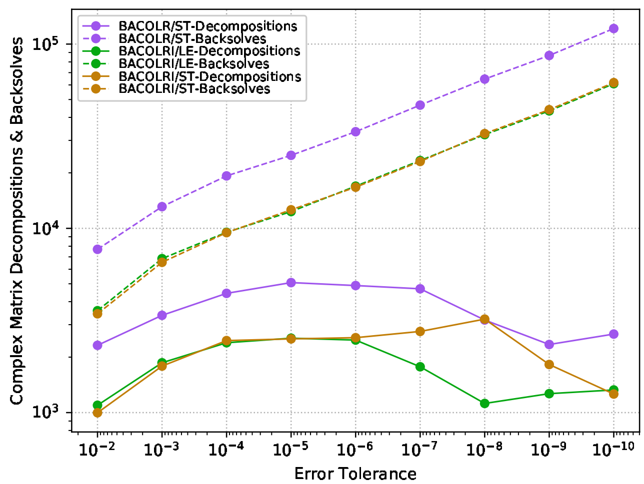

Figure 190: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 7$

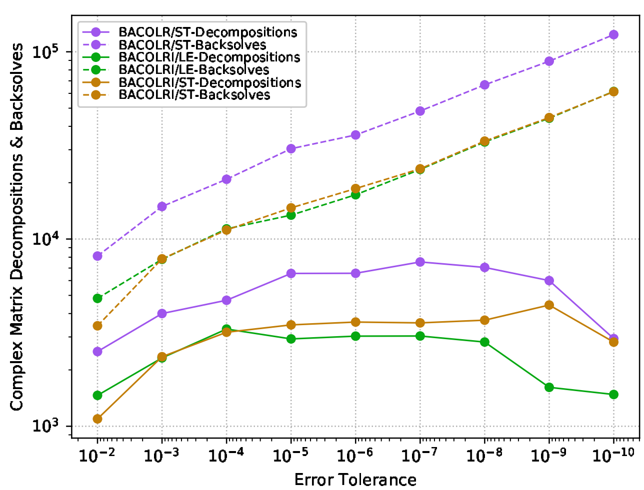

Figure 191: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 9$

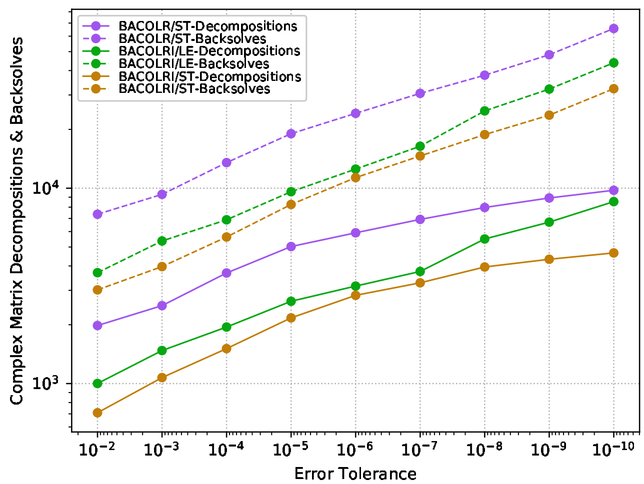

Figure 192: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Catalytic Surface Reaction Model with $p = 4$

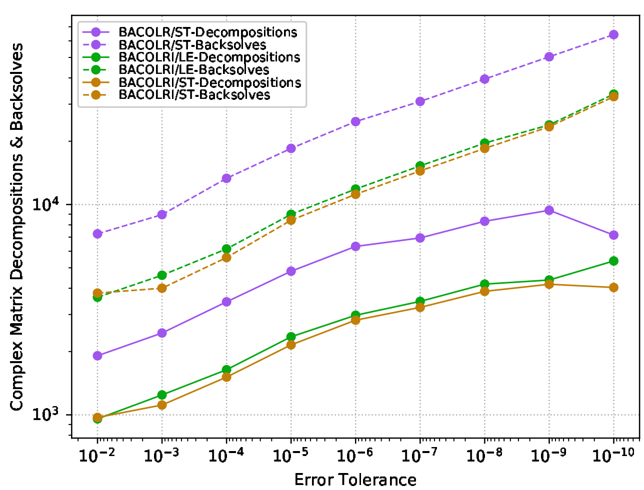

Figure 193: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Catalytic Surface Reaction Model with $p = 5$

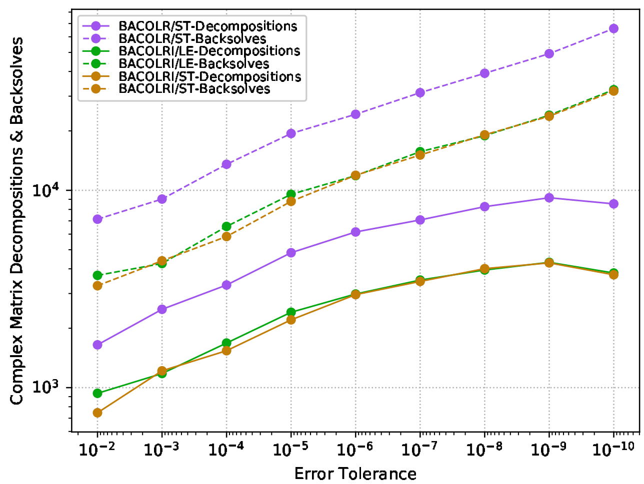

Figure 194: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Catalytic Surface Reaction Model with $p = 7$

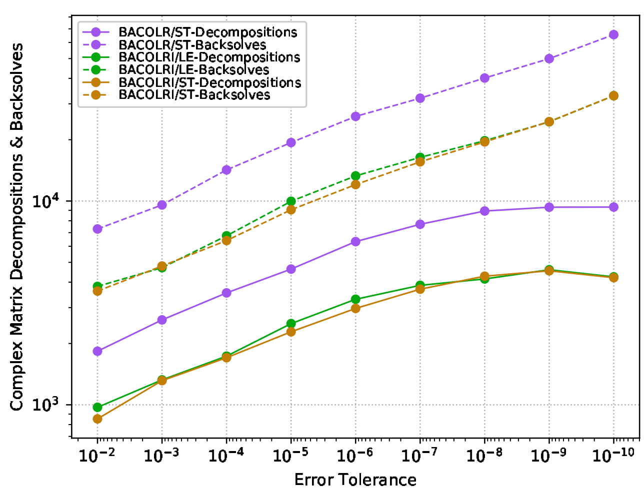

Figure 195: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Catalytic Surface Reaction Model with $p = 9$

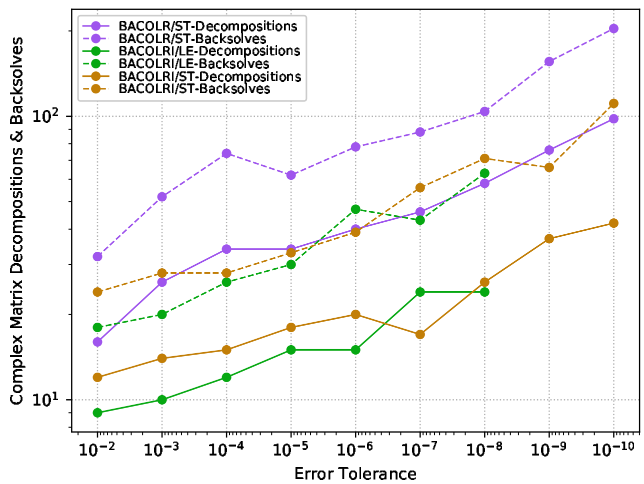

Figure 196: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Schrödinger Equation with $p = 4$

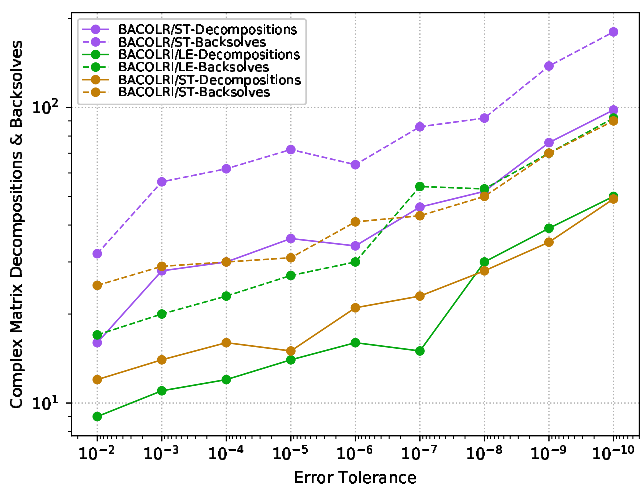

Figure 197: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Schrödinger Equation with $p = 5$

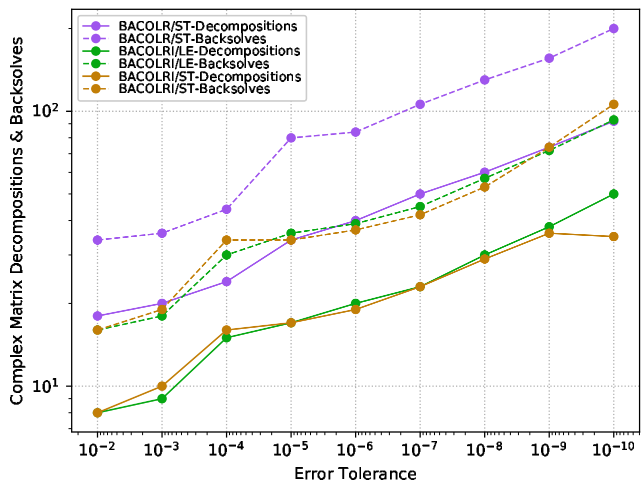

Figure 198: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Schrödinger Equation with $p = 7$

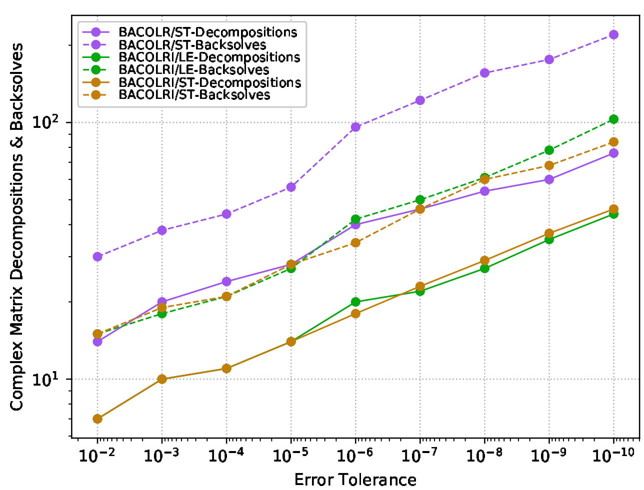

Figure 199: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Schrödinger Equation with $p = 9$

$tol = 10^{-4}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	0.06	0.07	0.08	0.09	0.11	0.13	0.15	0.19
BACOLR/LE	0.06	0.06	0.07	0.08	0.09	0.11	0.13	0.15
BACOLRI/ST	0.03	0.04	0.04	0.04	0.06	0.07	0.08	0.10
BACOLRI/LE	0.04	0.04	0.05	0.05	0.06	0.06	0.07	0.09
$tol = 10^{-6}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	0.13	0.13	0.14	0.16	0.19	0.23	0.27	0.28
BACOLR/LE	0.16	0.13	0.13	0.14	0.16	0.19	0.23	0.27
BACOLRI/ST	0.06	0.07	0.07	0.08	0.09	0.10	0.13	0.14
BACOLRI/LE	0.11	0.08	0.08	0.08	0.09	0.12	0.13	0.16
$tol = 10^{-8}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	0.45	0.38	0.35	0.34	0.37	0.40	0.47	0.50
BACOLR/LE	0.67	0.45	0.38	0.35	0.34	0.37	0.40	0.47
BACOLRI/ST	0.22	0.19	0.18	0.19	0.20	0.22	0.23	0.25
BACOLRI/LE	0.62	0.28	0.23	0.21	0.21	0.22	0.24	0.26
$tol = 10^{-10}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	2.03	1.26	1.01	0.90	0.89	0.88	0.87	0.96
BACOLR/LE	4.14	2.03	1.26	1.01	0.90	0.89	0.88	0.87
BACOLRI/ST	0.92	0.65	0.52	0.48	0.46	0.49	0.52	0.55
BACOLRI/LE	4.06	1.26	0.73	0.63	0.54	0.52	0.54	0.53

Table 12: *Machine dependent timings (in seconds), One Layer Burgers equation,* $\epsilon = 10^{-3}$, $p = 4, \ldots, 11$, $tol = 10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}$.

$tol = 10^{-4}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	0.46	0.55	0.67	0.88	1.12	1.29	1.61	2.01
BACOLR/LE	0.38	0.46	0.55	0.67	0.88	1.12	1.29	1.61
BACOLRI/ST	0.23	0.29	0.33	0.45	0.56	0.72	0.90	1.18
BACOLRI/LE	0.31	0.34	0.36	0.44	0.54	0.68	0.79	1.09
$tol = 10^{-6}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	1.08	1.19	1.36	1.49	1.78	2.17	2.59	3.35
BACOLR/LE	1.13	1.08	1.19	1.36	1.49	1.78	2.17	2.59
BACOLRI/ST	0.54	0.61	0.64	0.77	0.90	1.13	1.40	1.63
BACOLRI/LE	0.89	0.72	0.77	0.88	0.90	1.06	1.27	1.53
$tol = 10^{-8}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	3.95	3.32	3.35	3.54	4.01	4.14	4.54	5.43
BACOLR/LE	5.19	3.95	3.32	3.35	3.54	4.01	4.14	4.54
BACOLRI/ST	2.31	1.78	1.69	1.94	2.08	2.39	2.40	2.81
BACOLRI/LE	5.20	2.45	2.16	2.09	2.24	2.53	2.45	2.62
$tol = 10^{-10}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	17.17	11.23	9.63	8.94	8.96	9.10	10.21	10.74
BACOLR/LE	33.75	17.17	11.23	9.63	8.94	8.96	9.10	10.21
BACOLRI/ST	7.40	5.66	4.83	4.71	4.76	5.30	5.58	6.35
BACOLRI/LE	—	9.90	6.90	5.78	5.43	5.50	5.57	6.38

Table 13: *Machine dependent timings (in seconds), One Layer Burgers equation,* $\epsilon = 10^{-4}$, $p = 4, \ldots, 11$, $tol = 10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}$.

$tol = 10^{-4}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	0.05	0.06	0.07	0.08	0.09	0.11	0.14	0.16
BACOLR/LE	0.05	0.05	0.06	0.07	0.08	0.09	0.11	0.14
BACOLRI/ST	0.03	0.03	0.04	0.04	0.05	0.06	0.07	0.08
BACOLRI/LE	0.03	0.03	0.04	0.04	0.05	0.06	0.07	0.08
$tol = 10^{-6}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	0.12	0.12	0.12	0.13	0.15	0.17	0.19	0.23
BACOLR/LE	0.15	0.12	0.12	0.12	0.13	0.15	0.17	0.19
BACOLRI/ST	—	0.06	0.07	0.07	0.08	0.09	0.10	0.12
BACOLRI/LE	0.10	0.07	0.07	0.07	0.08	0.09	0.10	0.11
$tol = 10^{-8}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	0.42	0.32	0.30	0.29	0.30	0.33	0.39	0.44
BACOLR/LE	0.66	0.42	0.32	0.30	0.29	0.30	0.33	0.39
BACOLRI/ST	0.23	0.17	0.15	0.16	0.17	0.18	0.20	0.24
BACOLRI/LE	0.66	0.26	0.20	0.18	0.18	0.18	0.20	0.22
$tol = 10^{-10}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	1.88	1.19	0.92	0.82	0.77	0.76	0.76	0.80
BACOLR/LE	3.90	1.88	1.19	0.92	0.82	0.77	0.76	0.76
BACOLRI/ST	0.95	0.58	0.46	0.42	0.41	0.41	0.43	0.45
BACOLRI/LE	4.41	1.22	0.73	0.55	0.49	0.47	0.47	0.46

Table 14: *Machine dependent timings (in seconds), Two Layer Burgers equation,* $\epsilon = 10^{-3}$, $p = 4, \ldots, 11$, $tol = 10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}$.

$tol = 10^{-4}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	0.41	0.50	0.63	0.80	1.05	1.29	1.56	2.01
BACOLR/LE	0.35	0.41	0.50	0.63	0.80	1.05	1.29	1.56
BACOLRI/ST	0.22	0.24	0.29	0.38	0.51	0.64	0.82	0.98
BACOLRI/LE	0.27	0.28	0.33	0.41	0.52	0.65	0.86	1.03
$tol = 10^{-6}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	1.03	1.11	1.28	1.47	1.84	2.23	2.70	3.48
BACOLR/LE	1.20	1.03	1.11	1.28	1.47	1.84	2.23	2.70
BACOLRI/ST	0.53	0.57	0.61	0.73	0.89	1.13	1.38	1.77
BACOLRI/LE	0.85	0.68	0.70	0.81	0.90	1.10	1.33	1.62
$tol = 10^{-8}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	3.60	3.08	3.01	3.12	3.51	3.94	4.61	5.41
BACOLR/LE	5.54	3.60	3.08	3.01	3.12	3.51	3.94	4.61
BACOLRI/ST	1.73	1.61	1.55	1.70	1.85	2.06	2.36	2.79
BACOLRI/LE	4.98	2.32	1.95	1.86	1.99	2.16	2.37	2.79
$tol = 10^{-10}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	15.78	10.73	8.93	8.24	8.26	8.41	9.00	9.62
BACOLR/LE	30.78	15.78	10.73	8.93	8.24	8.26	8.41	9.00
BACOLRI/ST	7.51	5.20	4.50	4.28	4.31	4.58	4.89	5.39
BACOLRI/LE	30.82	10.42	6.53	5.36	5.09	5.04	5.15	5.67

Table 15: *Machine dependent timings (in seconds), Two Layer Burgers equation,* $\epsilon = 10^{-4}$, $p = 4, \ldots, 11$, $tol = 10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}$.

$tol = 10^{-4}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	0.64	0.80	1.08	1.32	1.67	2.44	3.47	4.22
BACOLR/LE	0.55	0.64	0.80	1.08	1.32	1.67	2.44	3.47
BACOLRI/ST	0.36	0.34	0.45	0.59	0.76	1.06	1.36	1.72
BACOLRI/LE	0.33	0.36	0.46	0.59	0.72	1.00	1.32	1.60
$tol = 10^{-6}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	1.29	1.34	1.51	1.94	2.55	3.41	4.13	5.72
BACOLR/LE	1.46	1.29	1.34	1.51	1.94	2.55	3.41	4.13
BACOLRI/ST	—	0.60	0.70	0.87	1.14	1.41	1.77	2.47
BACOLRI/LE	0.96	0.74	0.76	0.87	1.07	1.32	1.67	2.12
$tol = 10^{-8}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	4.70	3.70	3.63	3.76	3.96	4.77	6.48	8.77
BACOLR/LE	7.29	4.70	3.70	3.63	3.76	3.96	4.77	6.48
BACOLRI/ST	2.17	1.47	1.45	1.62	1.96	2.53	3.10	3.96
BACOLRI/LE	7.72	2.64	2.09	1.95	2.13	2.12	2.63	3.16
$tol = 10^{-10}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	27.23	16.87	13.35	12.58	12.06	12.14	12.74	13.49
BACOLR/LE	52.44	27.23	16.87	13.35	12.58	12.06	12.14	12.74
BACOLRI/ST	9.19	5.41	4.53	4.37	4.54	4.83	5.14	5.99
BACOLRI/LE	63.10	15.73	8.55	6.45	5.95	5.69	6.00	5.97

Table 16: *Machine dependent timings (in seconds), Two Layer Burgers equation*$\times 6$, $\epsilon = 10^{-3}$, $p = 4, \ldots, 11$, $tol = 10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}$.

$tol = 10^{-4}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	4.80	6.99	10.68	16.10	24.07	32.86	42.87	60.14
BACOLR/LE	3.43	4.80	6.99	10.68	16.10	24.07	32.86	42.87
BACOLRI/ST	2.02	2.78	4.09	6.42	9.56	14.10	20.75	28.21
BACOLRI/LE	2.51	3.21	4.50	6.49	9.48	14.04	21.90	29.29
$tol = 10^{-6}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	9.90	12.35	17.77	24.32	35.26	50.19	68.30	99.44
BACOLR/LE	10.23	9.90	12.35	17.77	24.32	35.26	50.19	68.30
BACOLRI/ST	4.24	5.46	6.82	9.86	13.69	20.48	28.83	42.04
BACOLRI/LE	7.20	6.09	7.51	10.19	13.34	19.17	26.28	37.77
$tol = 10^{-8}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	35.29	31.97	33.07	38.85	54.37	72.98	101.37	139.70
BACOLR/LE	57.42	35.29	31.97	33.07	38.85	54.37	72.98	101.37
BACOLRI/ST	12.65	13.40	14.54	19.44	24.21	29.82	39.80	57.01
BACOLRI/LE	49.31	20.02	17.71	18.20	22.12	28.63	37.60	50.56
$tol = 10^{-10}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	213.53	139.61	118.58	114.23	118.34	125.05	146.05	197.02
BACOLR/LE	387.40	213.53	139.61	118.58	114.23	118.34	125.05	146.05
BACOLRI/ST	64.19	43.67	39.70	40.57	44.37	55.41	67.47	86.79
BACOLRI/LE	—	118.19	68.97	57.56	55.76	57.42	63.27	76.48

Table 17: *Machine dependent timings (in seconds), Two Layer Burgers equation*$\times 6$, $\epsilon = 10^{-4}$, $p = 4, \ldots, 11$, $tol = 10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}$.

$tol = 10^{-4}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	2.94	4.07	6.03	7.80	10.22	14.20	19.79	25.13
BACOLR/LE	2.37	2.94	4.07	6.03	7.80	10.22	14.20	19.79
BACOLRI/ST	1.73	1.67	2.27	3.17	4.34	6.35	8.51	11.05
BACOLRI/LE	1.44	1.70	2.26	3.07	3.98	5.43	8.03	10.02
$tol = 10^{-6}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	6.62	6.97	8.28	11.11	14.88	20.11	26.09	34.15
BACOLR/LE	7.27	6.62	6.97	8.28	11.11	14.88	20.11	26.09
BACOLRI/ST	—	2.69	3.38	4.52	6.06	8.74	11.48	15.73
BACOLRI/LE	4.69	3.70	3.86	4.50	5.92	8.11	10.85	13.27
$tol = 10^{-8}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	23.23	19.93	20.57	20.51	23.64	28.01	37.83	48.90
BACOLR/LE	32.38	23.23	19.93	20.57	20.51	23.64	28.01	37.83
BACOLRI/ST	10.88	7.15	7.18	8.19	11.52	14.18	19.52	25.70
BACOLRI/LE	40.13	14.57	10.92	10.71	12.24	12.58	15.39	21.97
$tol = 10^{-10}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	120.49	75.17	65.21	63.22	62.52	64.92	71.02	81.04
BACOLR/LE	196.35	120.49	75.17	65.21	63.22	62.52	64.92	71.02
BACOLRI/ST	46.26	30.44	26.19	25.65	26.53	28.68	33.10	39.39
BACOLRI/LE	288.69	78.40	47.19	37.79	34.51	36.26	39.92	41.02

Table 18: *Machine dependent timings (in seconds), Two Layer Burgers equation*$\times 12$, $\epsilon = 10^{-3}$, $p = 4, \ldots, 11$, $tol = 10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}$.

$tol = 10^{-4}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	26.79	43.38	64.51	95.33	139.98	180.60	257.60	341.84
BACOLR/LE	16.43	26.79	43.38	64.51	95.33	139.98	180.60	257.60
BACOLRI/ST	9.84	14.42	23.74	37.75	60.11	88.15	127.47	185.86
BACOLRI/LE	12.51	17.40	25.55	39.08	58.32	91.80	132.70	192.89
$tol = 10^{-6}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	54.82	70.18	101.78	141.27	209.24	288.91	414.65	559.11
BACOLR/LE	50.29	54.82	70.18	101.78	141.27	209.24	288.91	414.65
BACOLRI/ST	19.37	27.24	37.12	55.94	85.33	128.31	181.19	269.66
BACOLRI/LE	35.77	32.33	40.83	61.38	79.55	119.74	173.98	225.42
$tol = 10^{-8}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	185.55	171.09	181.80	218.50	286.54	416.69	588.50	776.33
BACOLR/LE	253.16	185.55	171.09	181.80	218.50	286.54	416.69	588.50
BACOLRI/ST	57.90	66.57	77.52	111.62	137.44	189.13	246.85	359.23
BACOLRI/LE	274.80	112.43	98.81	102.78	124.60	169.11	235.49	328.76
$tol = 10^{-10}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	952.93	609.53	558.62	548.26	592.67	656.21	760.71	1046.63
BACOLR/LE	1398.00	952.93	609.53	558.62	548.26	592.67	656.21	760.71
BACOLRI/ST	329.52	241.92	226.28	232.36	272.72	335.29	434.17	550.88
BACOLRI/LE	—	570.95	379.06	329.05	327.68	348.56	399.72	478.84

Table 19: *Machine dependent timings (in seconds), Two Layer Burgers equation*$\times 12$, $\epsilon = 10^{-4}$, $p = 4, \ldots, 11$, $tol = 10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}$.

$tol = 10^{-4}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	2.36	2.80	3.49	4.45	5.39	7.31	9.92	13.05
BACOLR/LE	2.03	2.36	2.80	3.49	4.45	5.39	7.31	9.92
BACOLRI/ST	0.77	0.95	1.26	1.78	2.26	3.00	4.02	5.41
BACOLRI/LE	1.24	1.19	1.43	1.78	2.27	2.99	3.93	5.17
$tol = 10^{-6}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	6.83	8.06	9.51	11.54	14.16	17.03	20.11	24.89
BACOLR/LE	8.33	6.83	8.06	9.51	11.54	14.16	17.03	20.11
BACOLRI/ST	2.30	2.80	3.38	4.26	5.18	6.29	7.86	9.59
BACOLRI/LE	4.62	3.96	4.31	4.88	5.89	7.45	8.84	10.65
$tol = 10^{-8}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	19.48	19.96	19.56	22.47	25.14	30.24	36.25	41.71
BACOLR/LE	34.91	19.48	19.96	19.56	22.47	25.14	30.24	36.25
BACOLRI/ST	7.04	6.95	7.04	8.31	9.40	11.24	13.79	16.13
BACOLRI/LE	28.59	11.31	10.08	10.26	11.18	12.59	14.87	17.75
$tol = 10^{-10}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	83.59	57.96	46.63	46.69	46.20	51.45	54.73	59.65
BACOLR/LE	210.76	83.59	57.96	46.63	46.69	46.20	51.45	54.73
BACOLRI/ST	29.36	19.08	16.62	17.13	17.48	19.96	22.13	25.10
BACOLRI/LE	173.76	49.53	28.04	23.51	22.67	23.71	25.08	26.74

Table 20: *Machine dependent timings (in seconds), Catalytic Surface Reaction Model,* $p = 4, \ldots, 11$, $tol = 10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}$.

$tol = 10^{-4}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	0.16	0.15	0.17	0.17	0.18	0.20	0.21	0.23
BACOLR/LE	0.15	0.16	0.15	0.17	0.17	0.18	0.20	0.21
BACOLRI/ST	0.08	0.08	0.08	0.08	0.09	0.09	0.10	0.11
BACOLRI/LE	0.08	0.07	0.08	0.09	0.08	0.09	0.10	0.10
$tol = 10^{-6}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	0.19	0.18	0.21	0.22	0.22	0.24	0.26	0.28
BACOLR/LE	0.23	0.19	0.18	0.21	0.22	0.22	0.24	0.26
BACOLRI/ST	0.09	0.09	0.09	0.10	0.11	0.11	0.12	0.13
BACOLRI/LE	0.12	0.09	0.10	0.11	0.11	0.12	0.13	0.13
$tol = 10^{-8}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	0.37	0.27	0.30	0.32	0.31	0.34	0.35	0.40
BACOLR/LE	0.59	0.37	0.27	0.30	0.32	0.31	0.34	0.35
BACOLRI/ST	0.16	0.14	0.13	0.14	0.14	0.16	0.16	0.19
BACOLRI/LE	0.40	0.22	0.16	0.17	0.17	0.17	0.17	0.18
$tol = 10^{-10}/p =$	4	5	6	7	8	9	10	11
BACOLR/ST	1.22	0.88	0.67	0.63	0.57	0.56	0.59	—
BACOLR/LE	—	1.22	0.88	0.67	0.63	0.57	0.56	0.59
BACOLRI/ST	0.50	0.35	0.29	0.26	0.28	0.29	0.27	0.28
BACOLRI/LE	—	0.79	0.51	0.41	0.33	0.31	0.29	0.30

Table 21: *Machine dependent timings (in seconds), Schrödinger Equation,* $p = 4, \ldots, 11$, $tol = 10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}$.

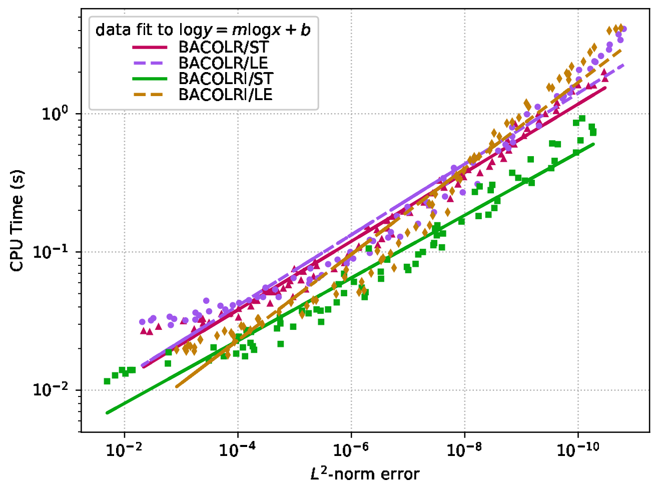

Figure 200: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 4$

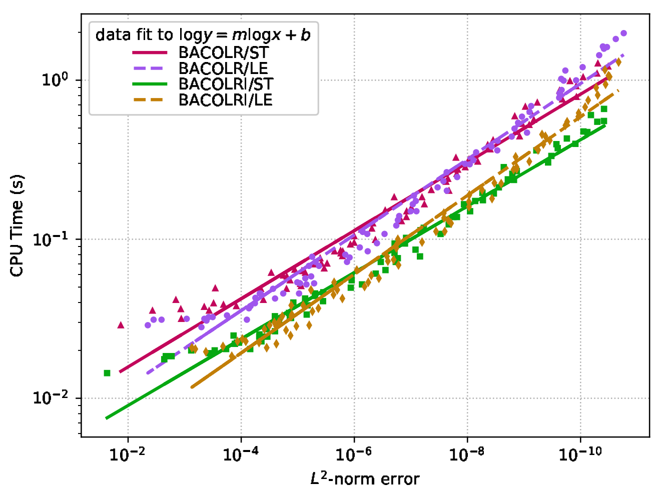

Figure 201: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 5$

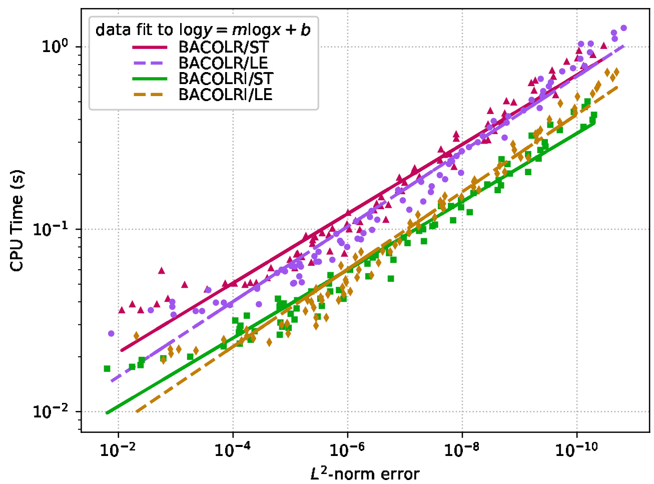

Figure 202: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 6$

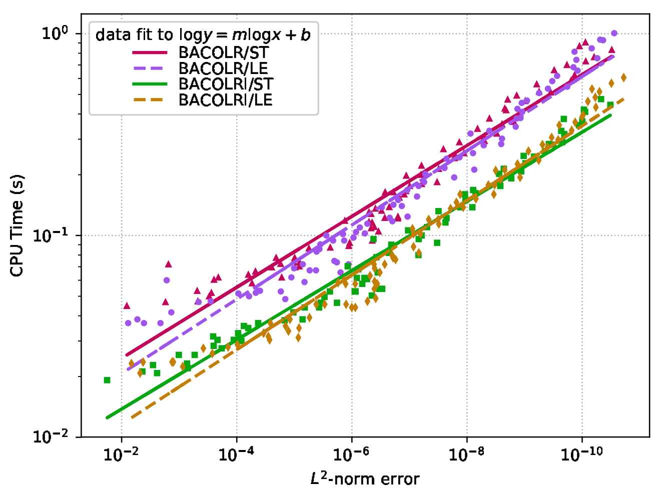

Figure 203: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 7$

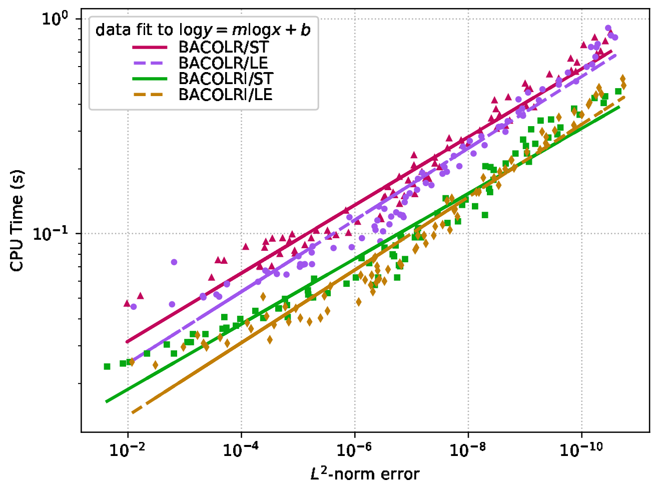

Figure 204: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 8$

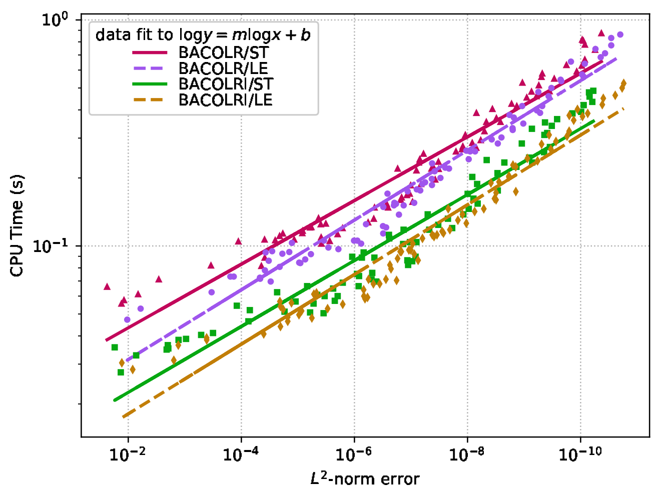

Figure 205: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 9$

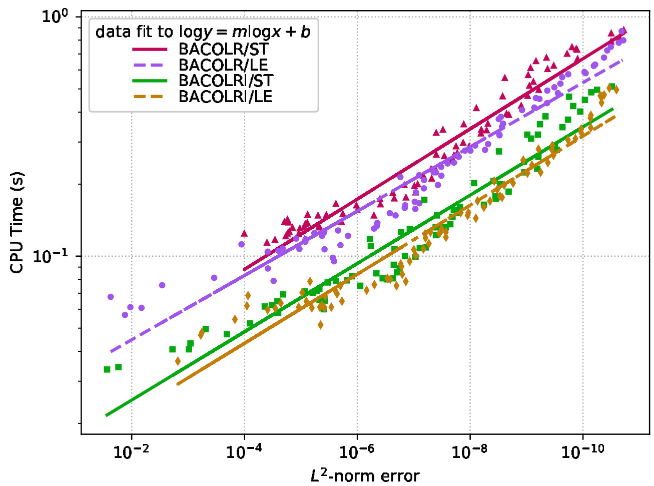

Figure 206: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 10$

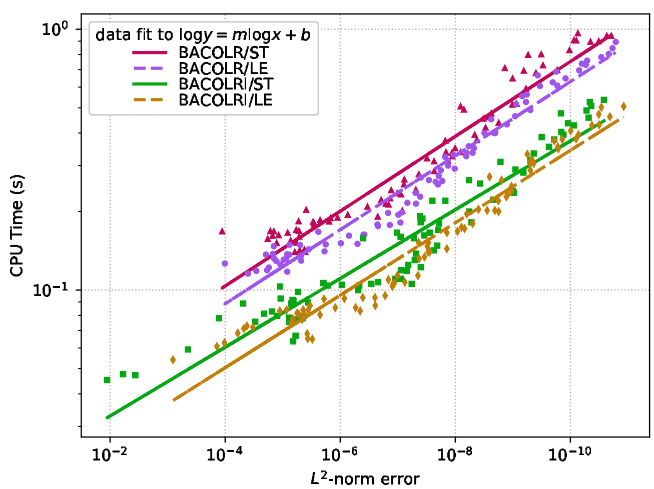

Figure 207: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 11$

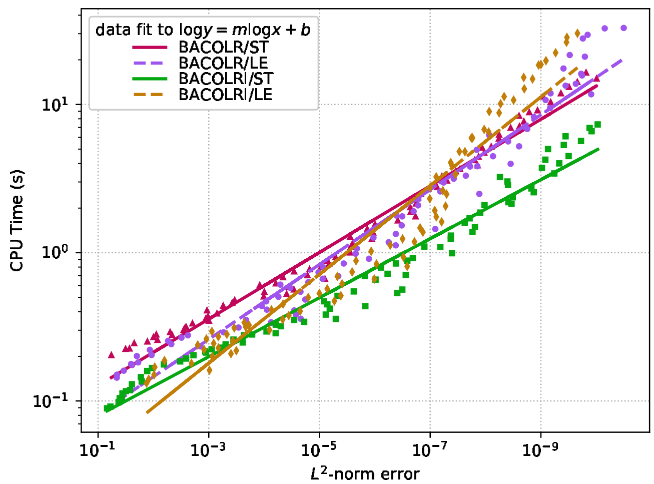

Figure 208: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 4$

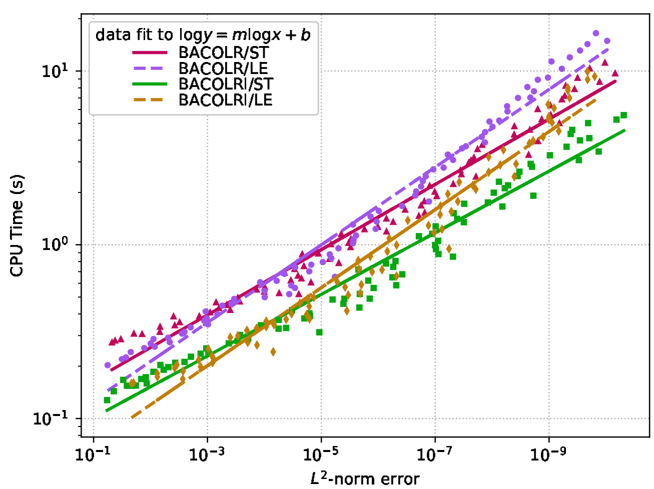

Figure 209: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 5$

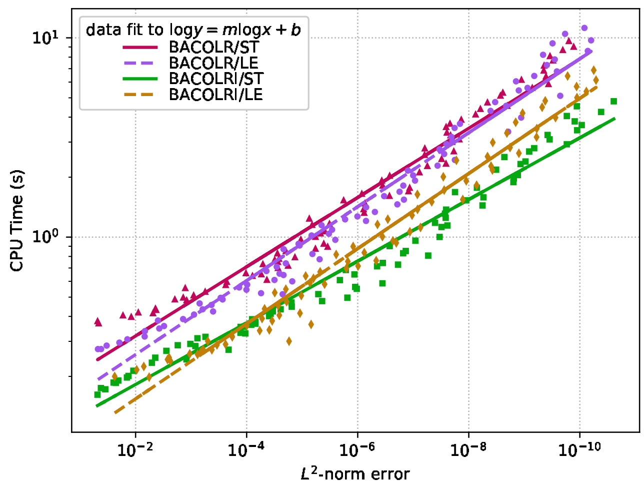

Figure 210: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 6$

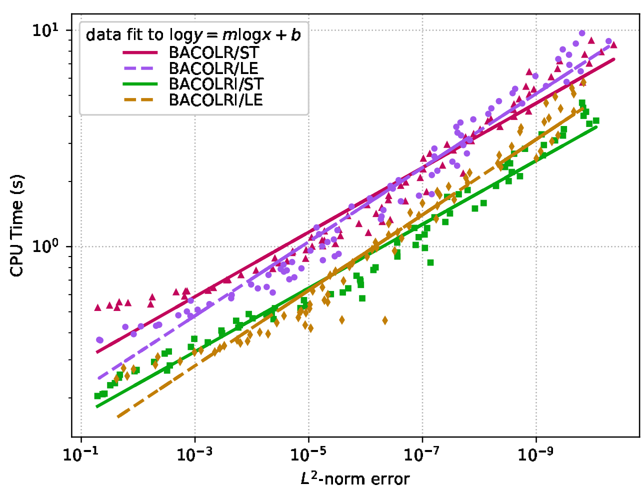

Figure 211: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 7$

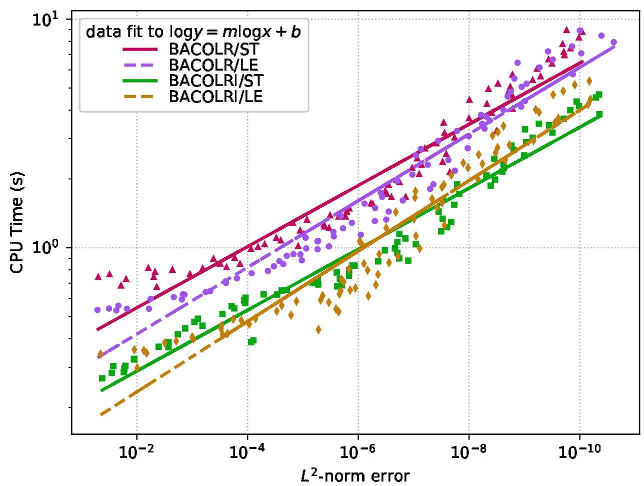

Figure 212: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 8$

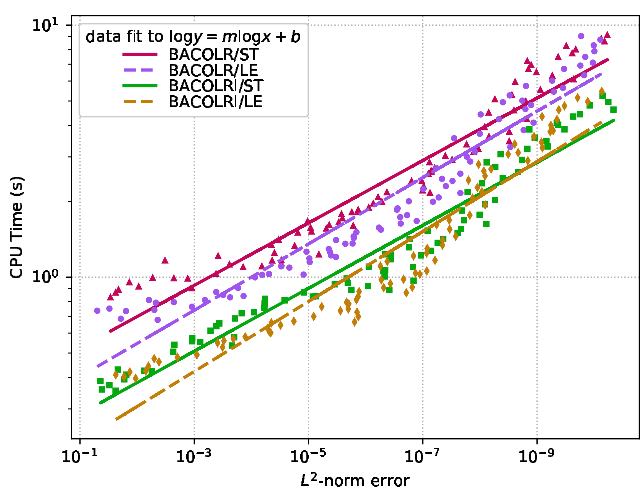

Figure 213: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 9$

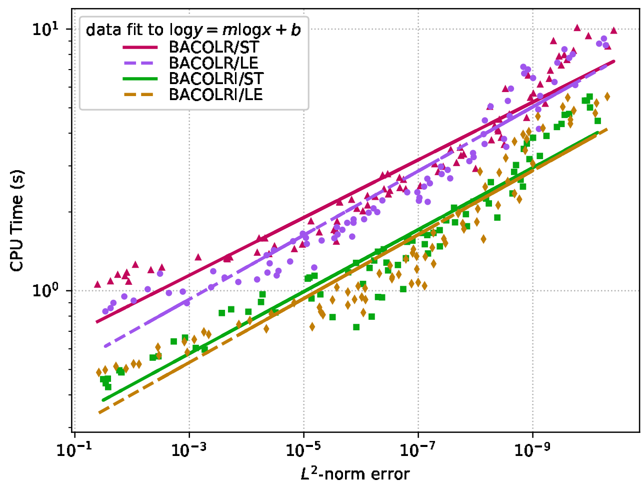

Figure 214: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 10$

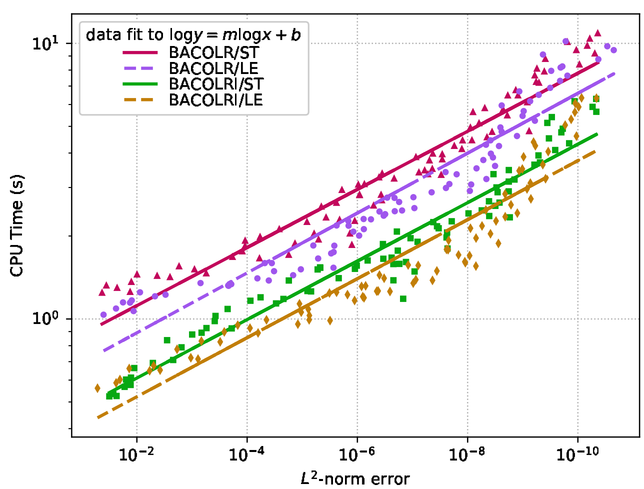

Figure 215: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 11$

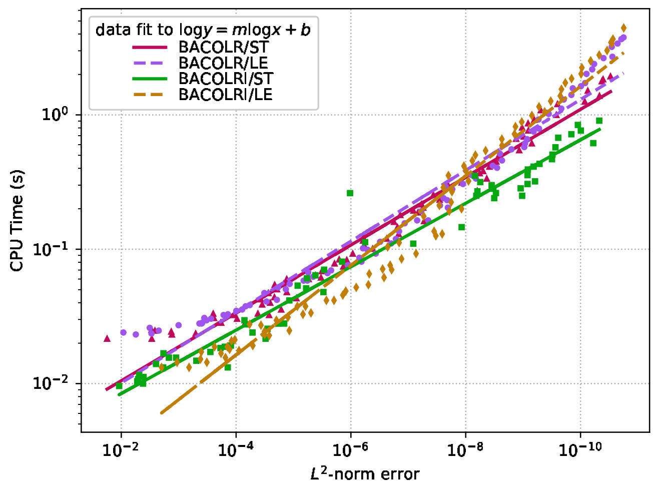

Figure 216: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 4$

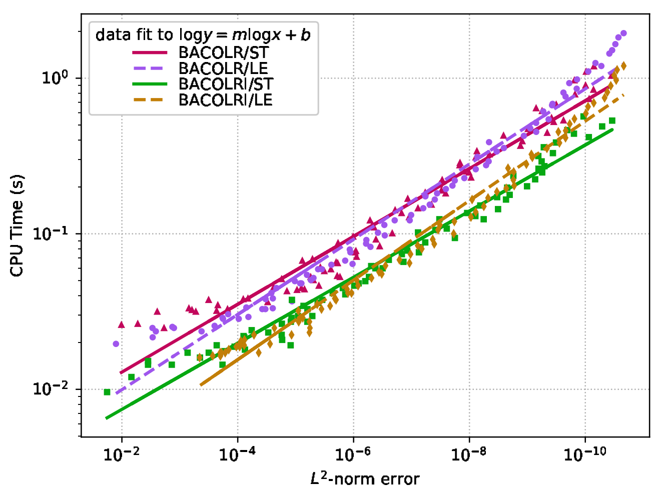

Figure 217: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 5$

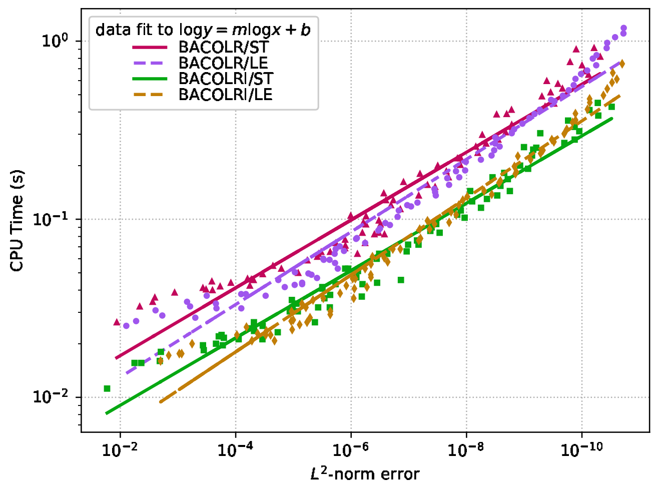

Figure 218: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 6$

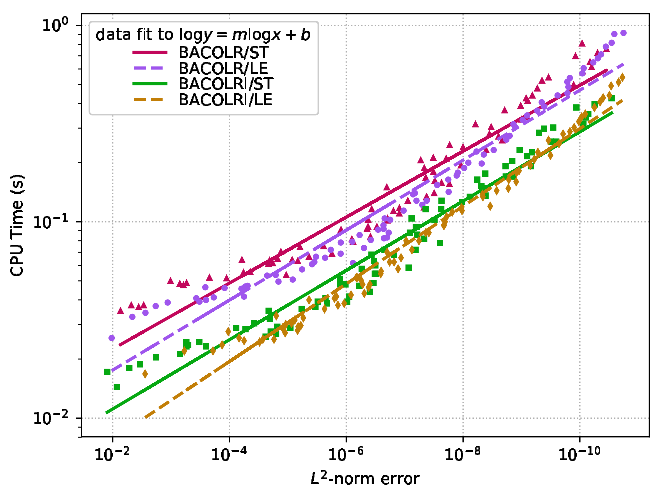

Figure 219: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 7$

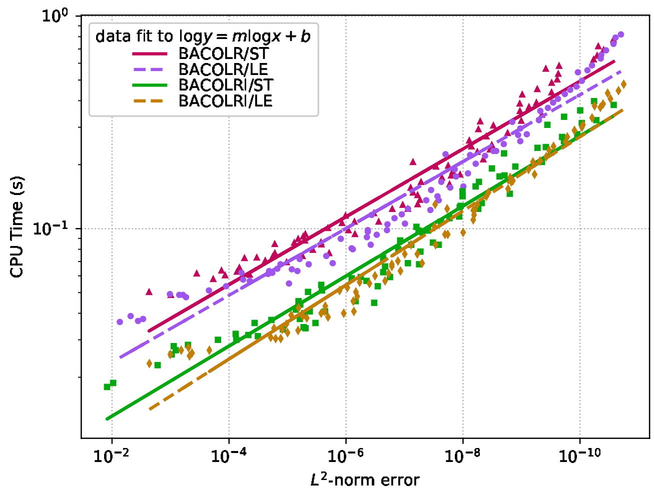

Figure 220: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 8$

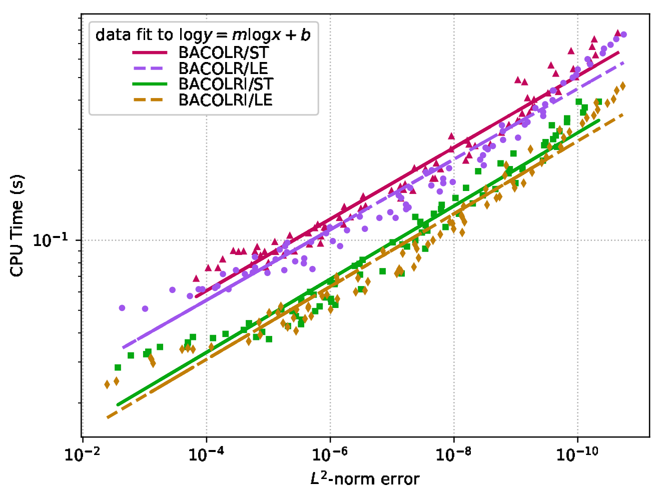

Figure 221: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 9$

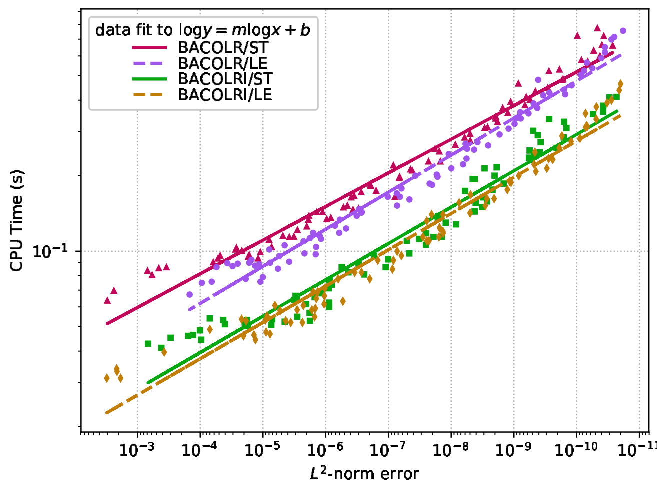

Figure 222: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 10$

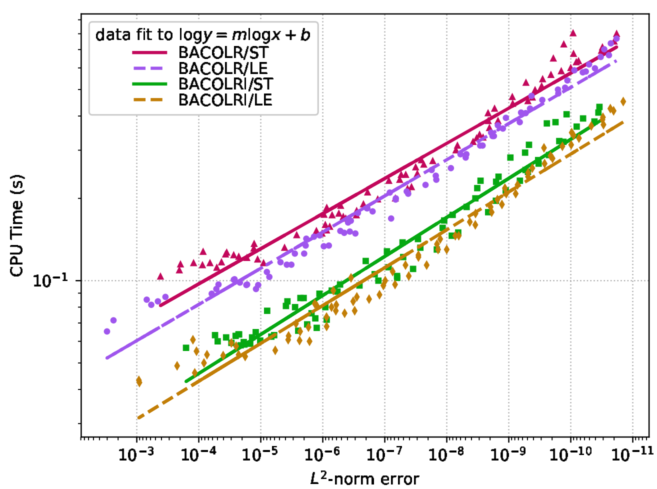

Figure 223: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 11$

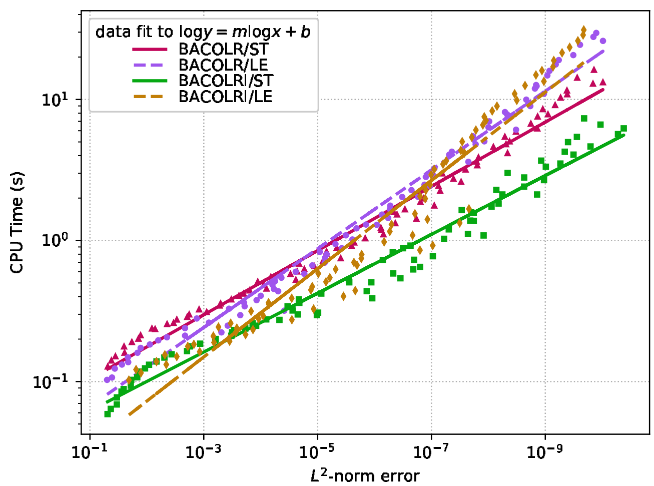

Figure 224: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 4$

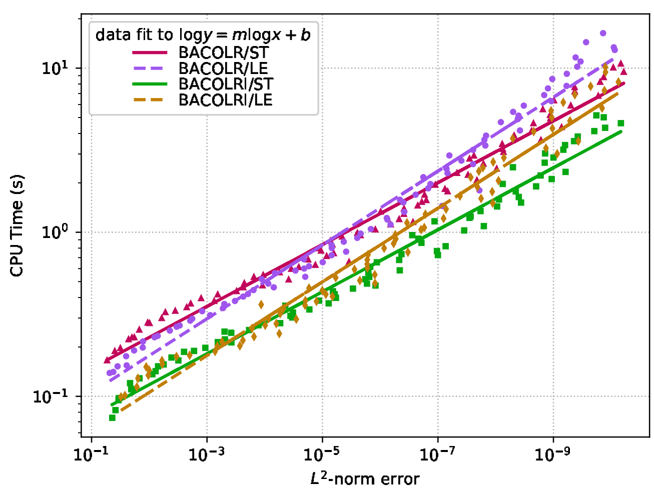

Figure 225: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 5$

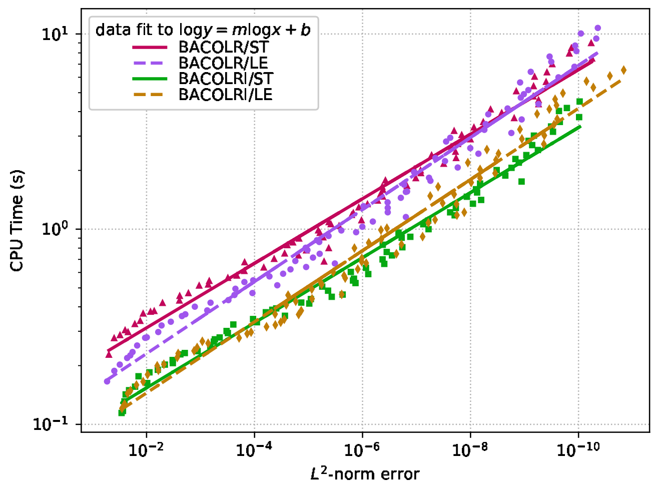

Figure 226: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 6$

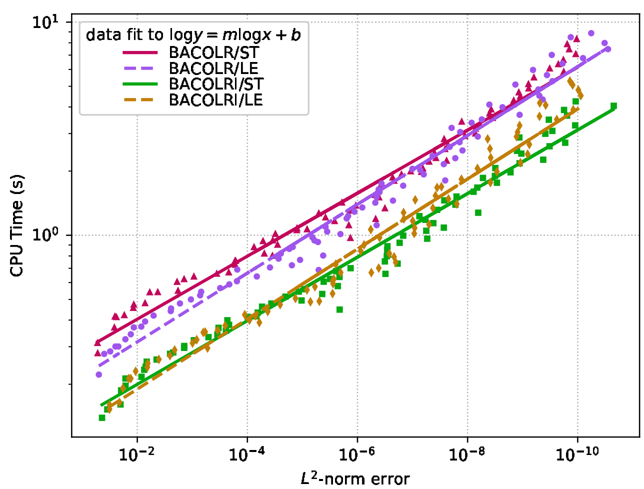

Figure 227: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 7$

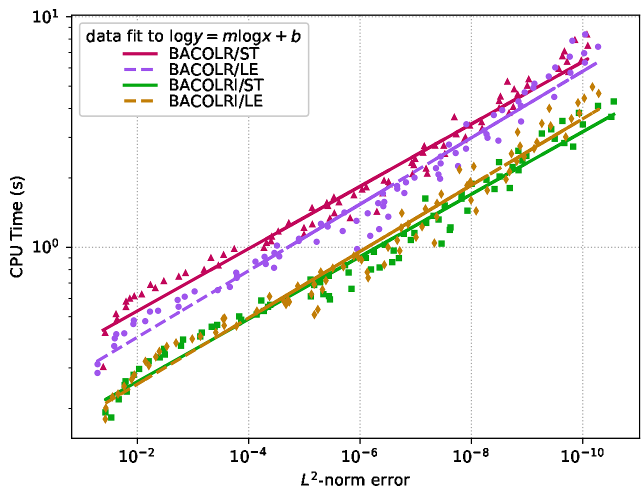

Figure 228: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 8$

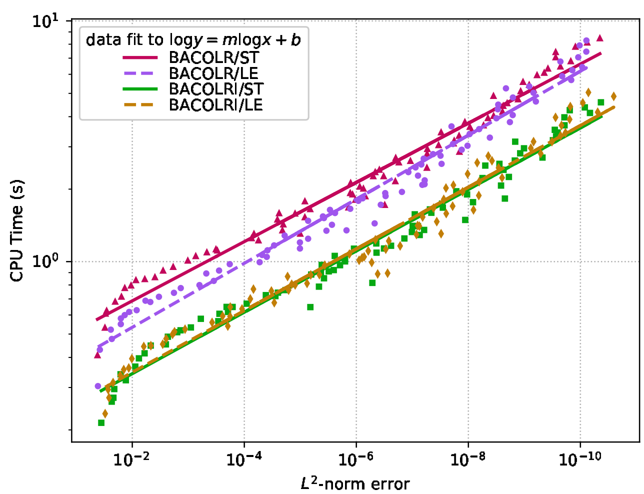

Figure 229: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 9$

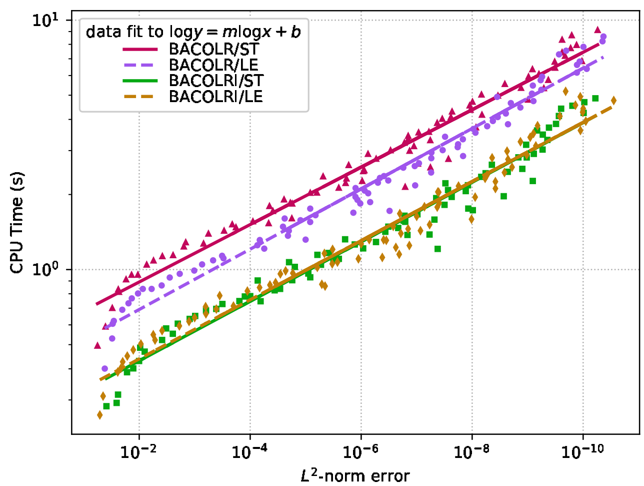

Figure 230: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 10$

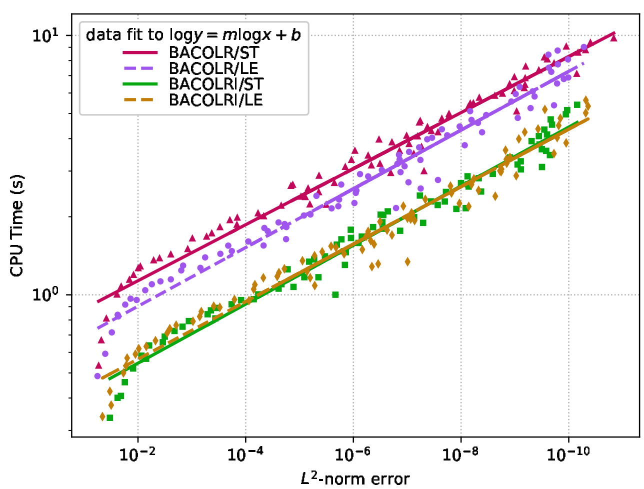

Figure 231: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 11$

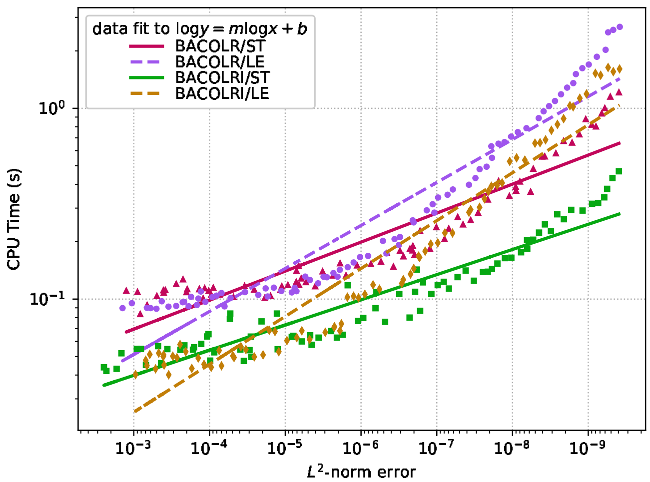

Figure 232: Work vs. Accuracy: Schrödinger System, $p = 4$

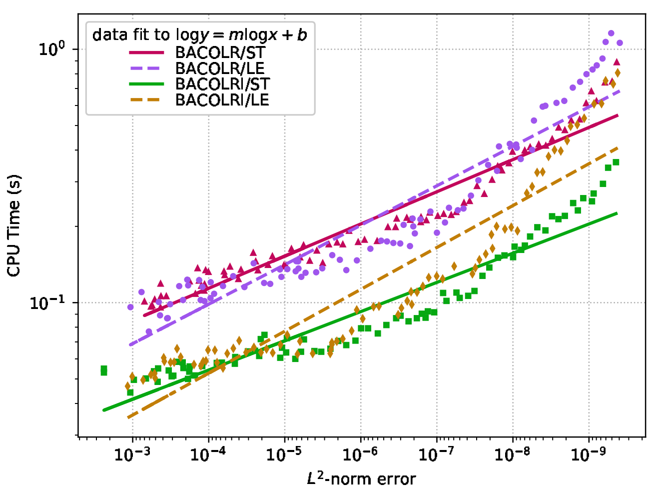

Figure 233: Work vs. Accuracy: Schrödinger System, $p = 5$

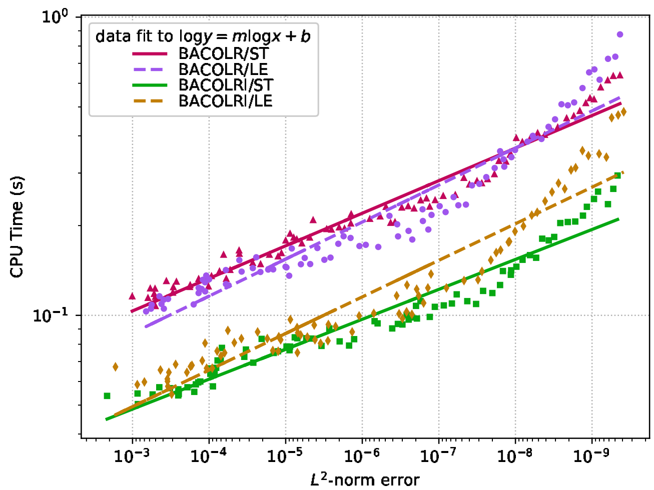

Figure 234: Work vs. Accuracy: Schrödinger System, $p = 6$

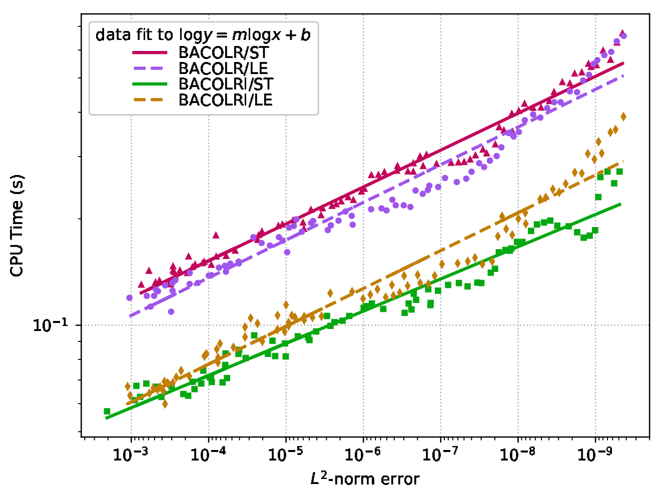

Figure 235: Work vs. Accuracy: Schrödinger System, $p = 7$

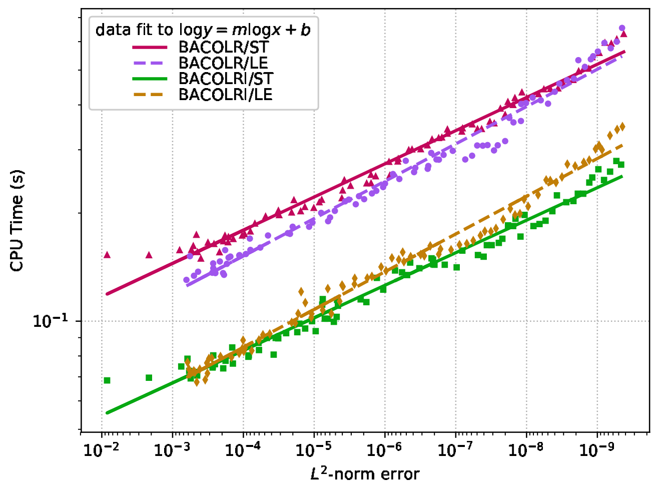

Figure 236: Work vs. Accuracy: Schrödinger System, $p = 8$

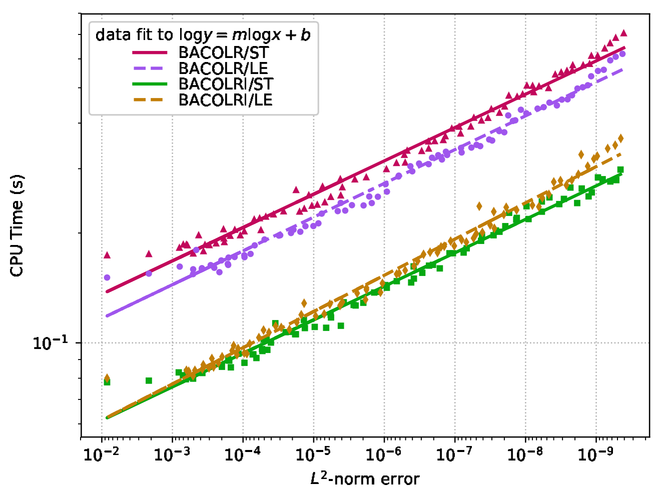

Figure 237: Work vs. Accuracy: Schrödinger System, $p = 9$

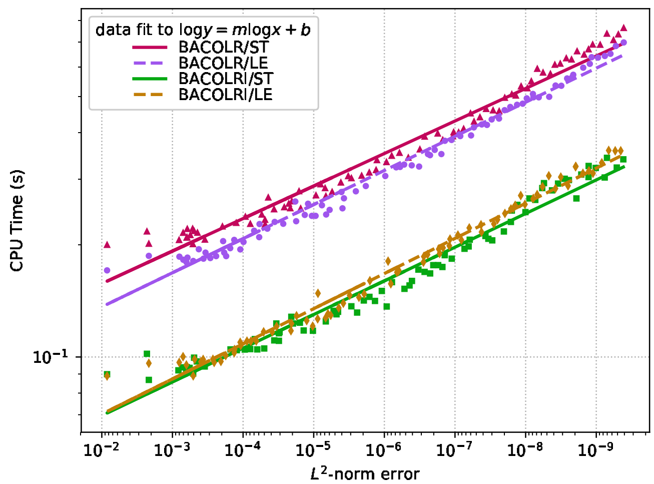

Figure 238: Work vs. Accuracy: Schrödinger System, $p = 10$

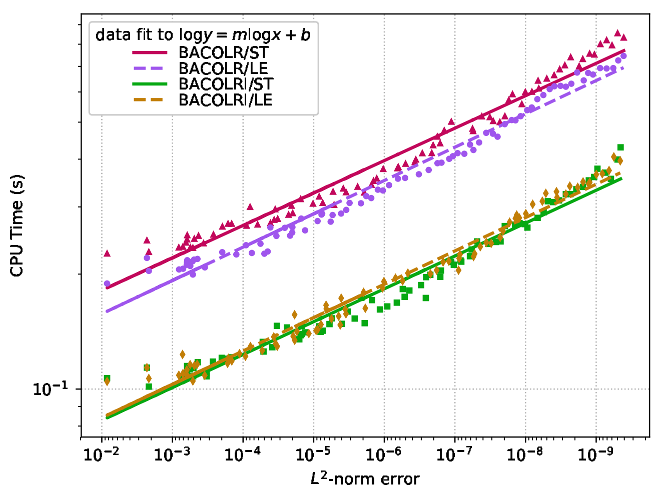

Figure 239: Work vs. Accuracy: Schrödinger System, $p = 11$

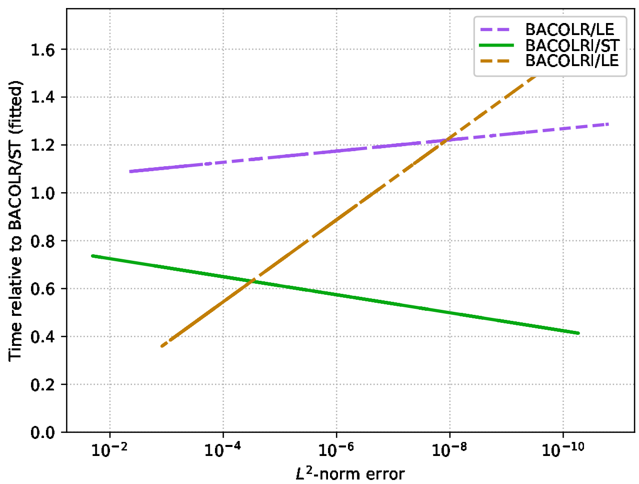

Figure 240: Rel. Work-Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 4$

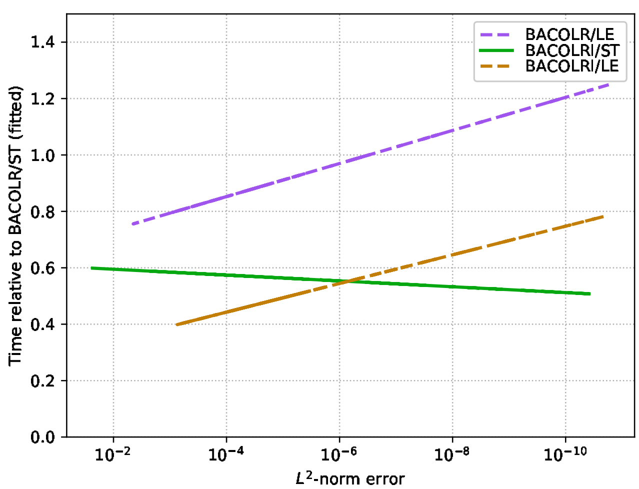

Figure 241: Rel. Work-Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 5$

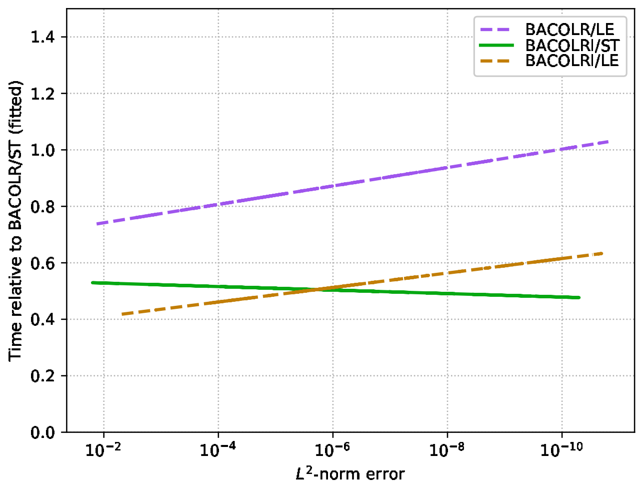

Figure 242: Rel. Work-Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 6$

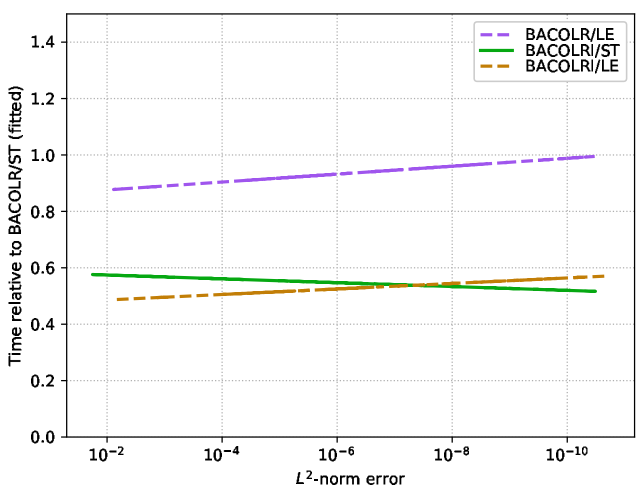

Figure 243: Rel. Work-Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 7$

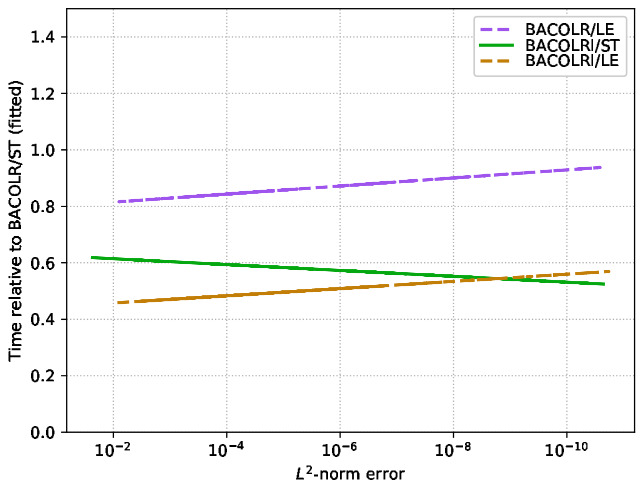

Figure 244: Rel. Work-Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 8$

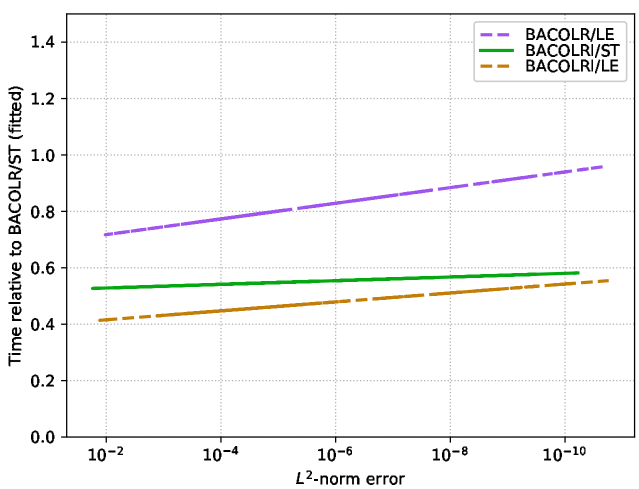

Figure 245: Rel. Work-Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 9$

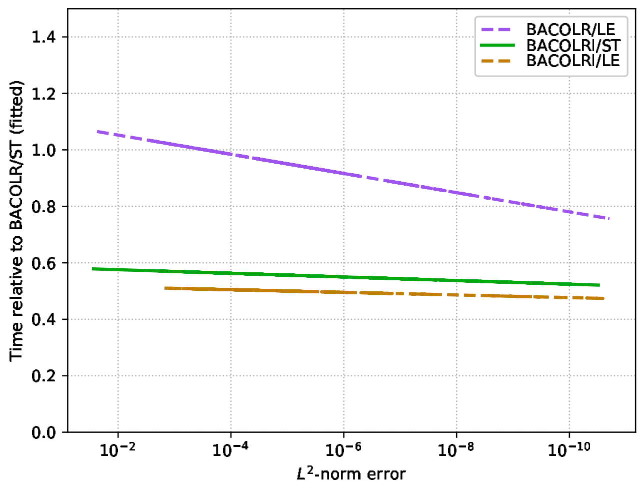

Figure 246: Rel. Work-Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 10$

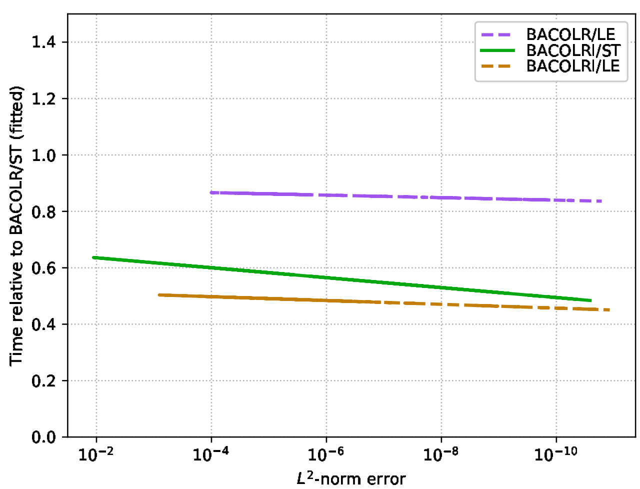

Figure 247: Rel. Work-Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 11$

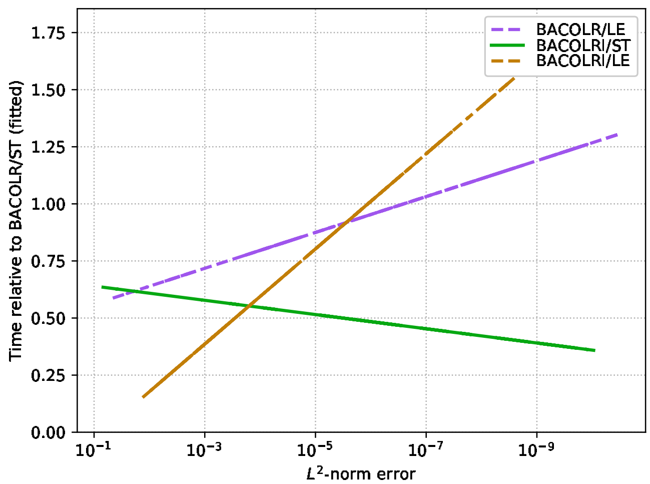

Figure 248: Rel. Work-Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 4$

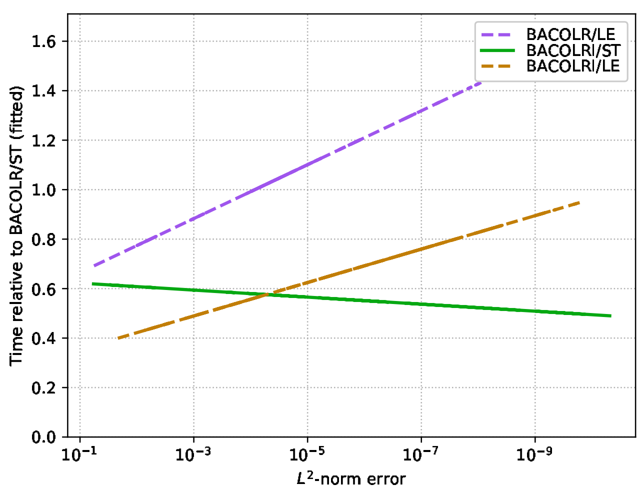

Figure 249: Rel. Work-Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 5$

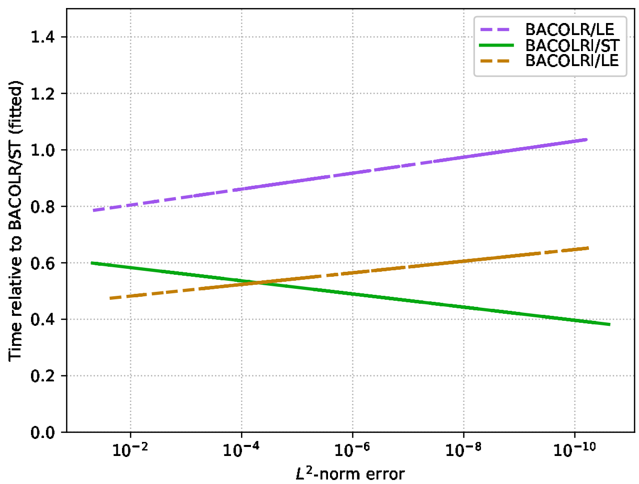

Figure 250: Rel. Work-Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 6$

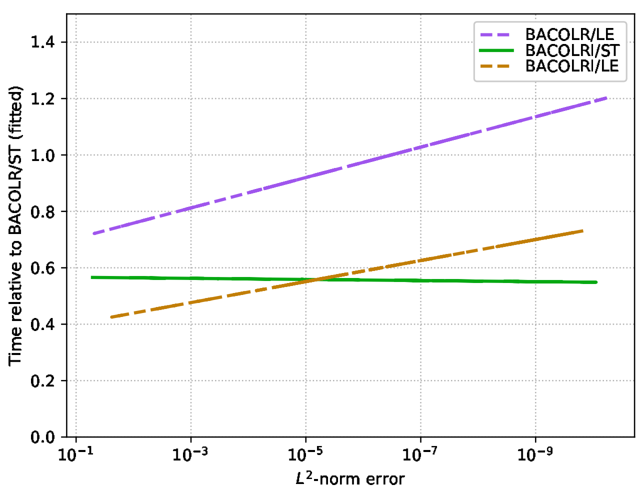

Figure 251: Rel. Work-Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 7$

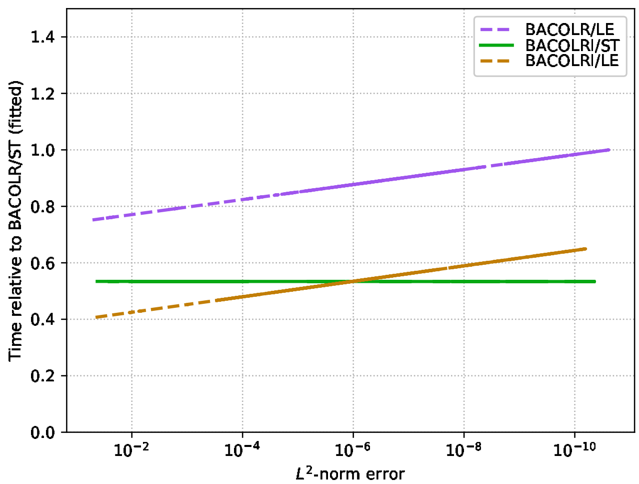

Figure 252: Rel. Work-Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 8$

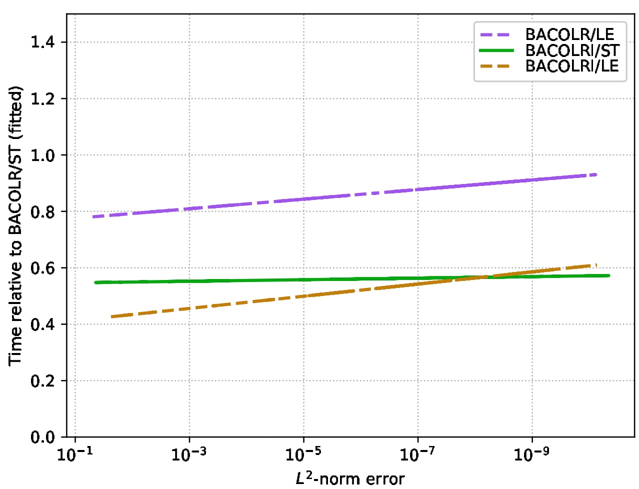

Figure 253: Rel. Work-Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 9$

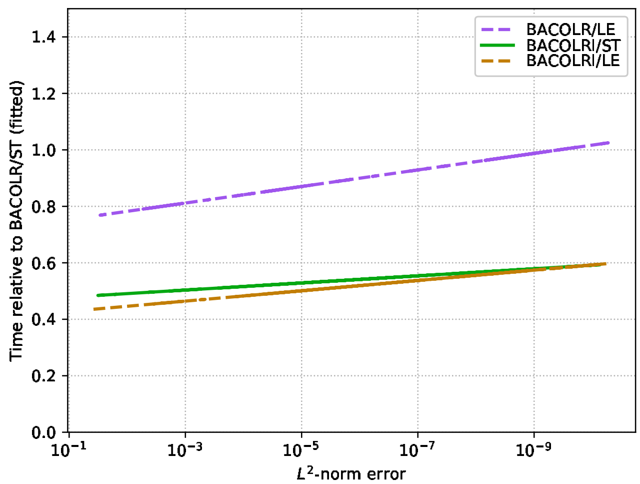

Figure 254: Rel. Work-Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 10$

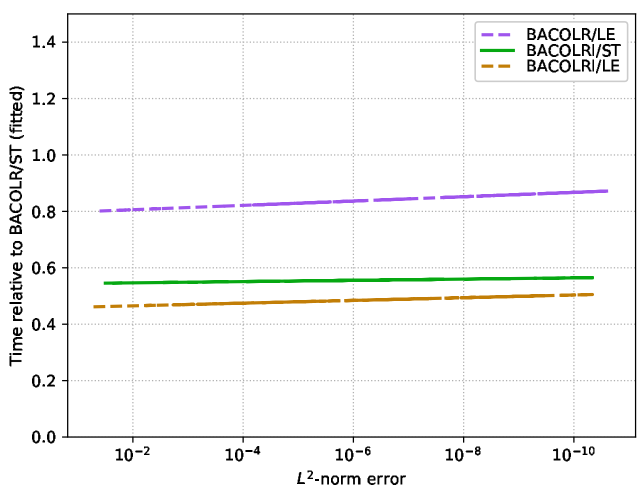

Figure 255: Rel. Work-Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 11$

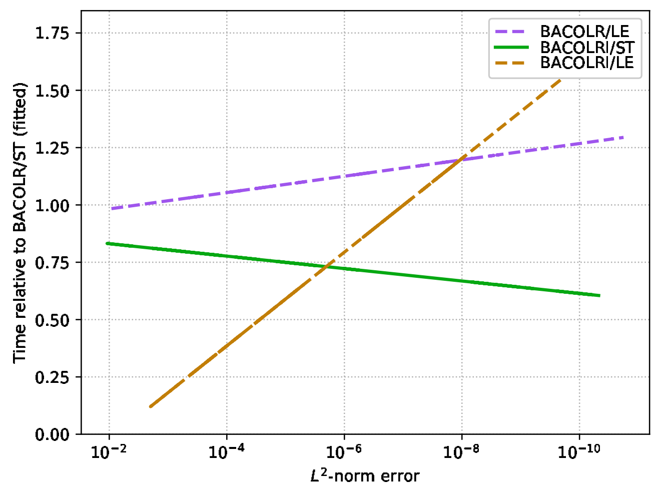

Figure 256: Rel. Work-Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 4$

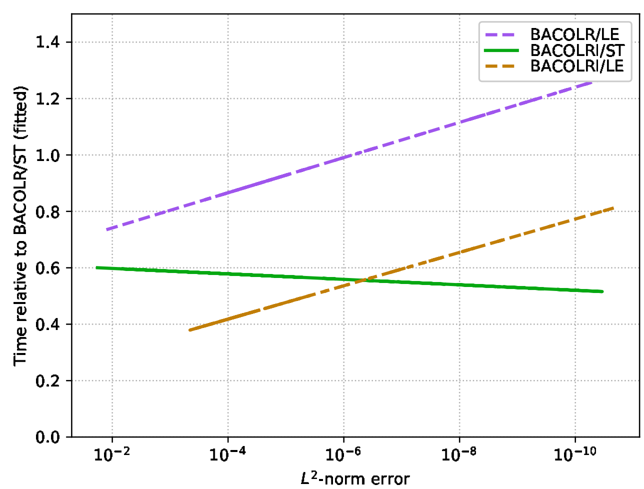

Figure 257: Rel. Work-Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 5$

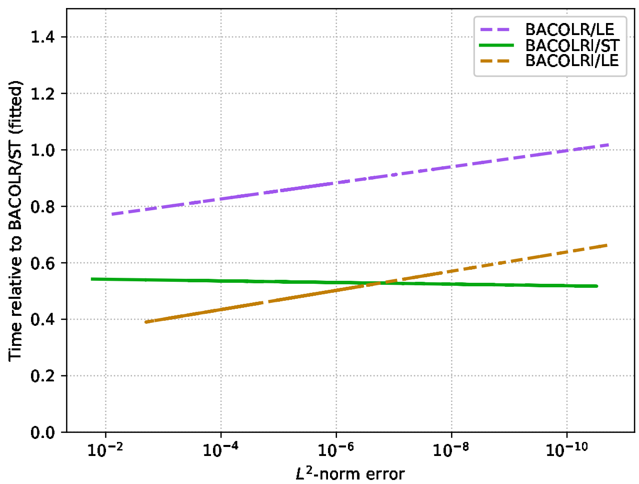

Figure 258: Rel. Work-Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 6$

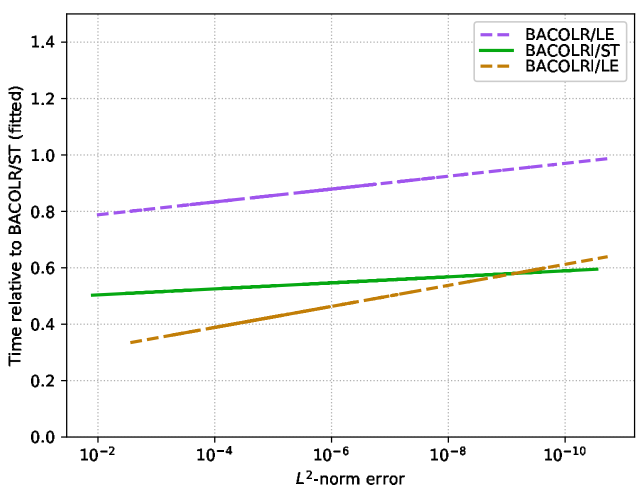

Figure 259: Rel. Work-Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 7$

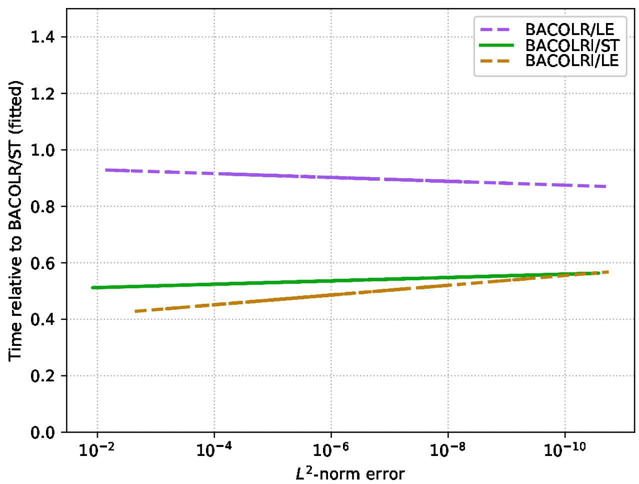

Figure 260: Rel. Work-Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 8$

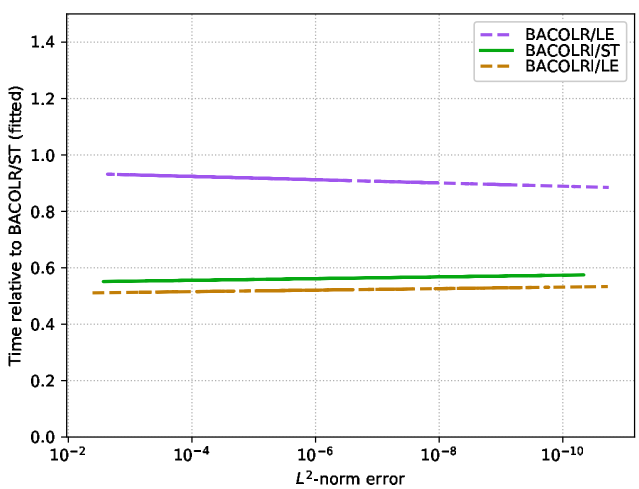

Figure 261: Rel. Work-Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 9$

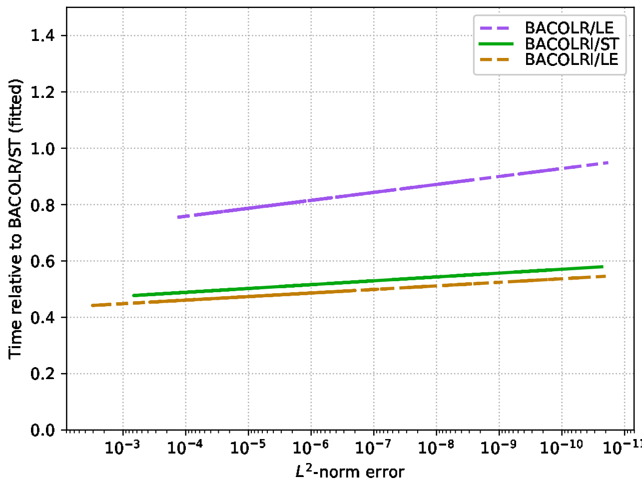

Figure 262: Rel. Work-Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 10$

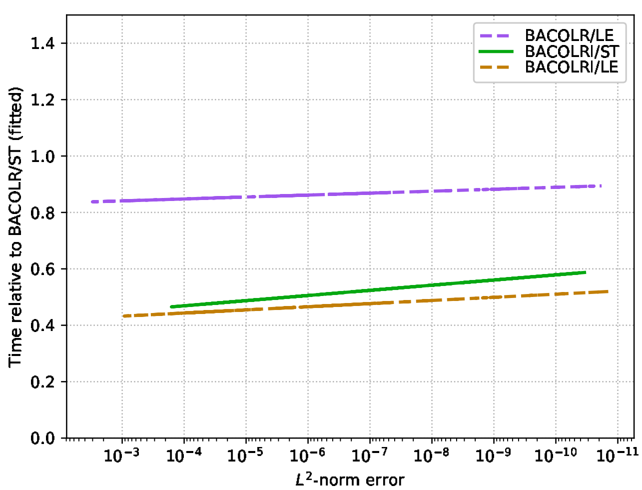

Figure 263: Rel. Work-Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 11$

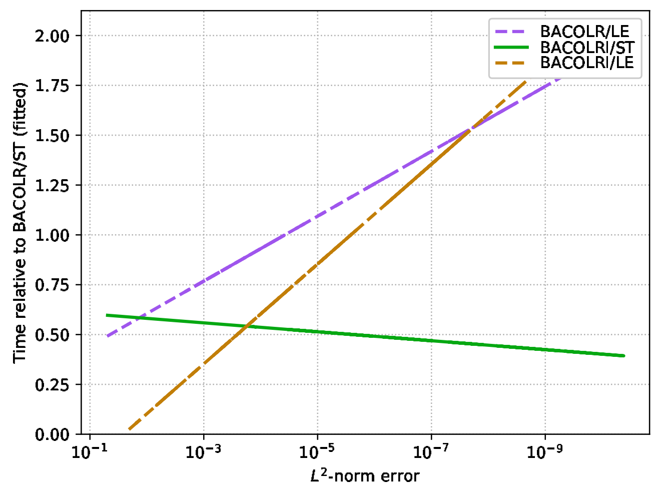

Figure 264: Rel. Work-Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 4$

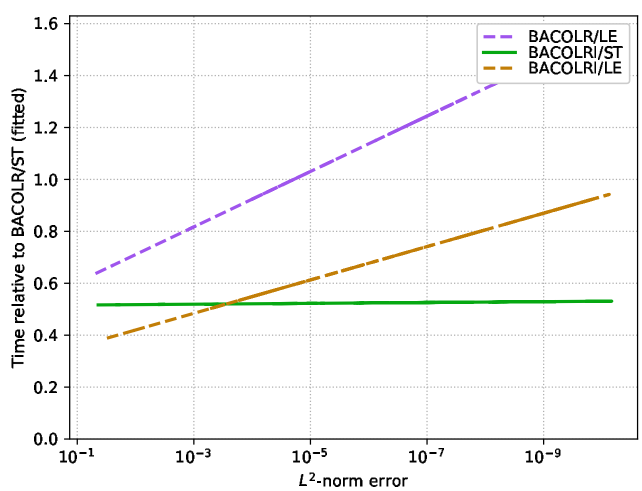

Figure 265: Rel. Work-Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 5$

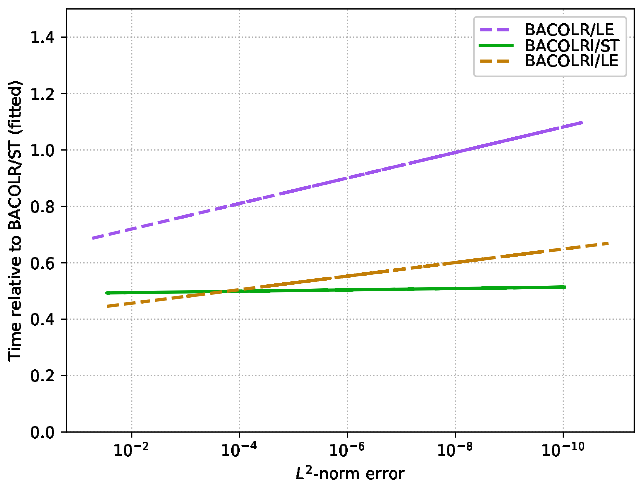

Figure 266: Rel. Work-Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 6$

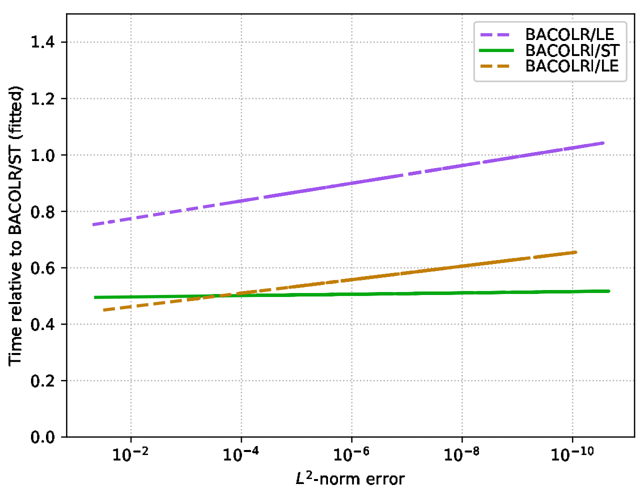

Figure 267: Rel. Work-Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 7$

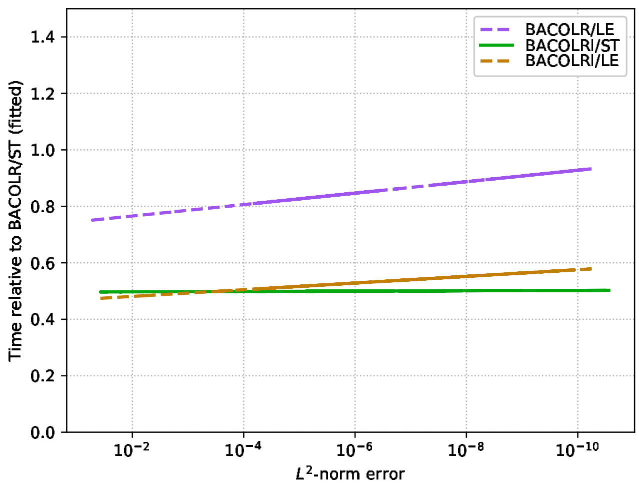

Figure 268: Rel. Work-Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 8$

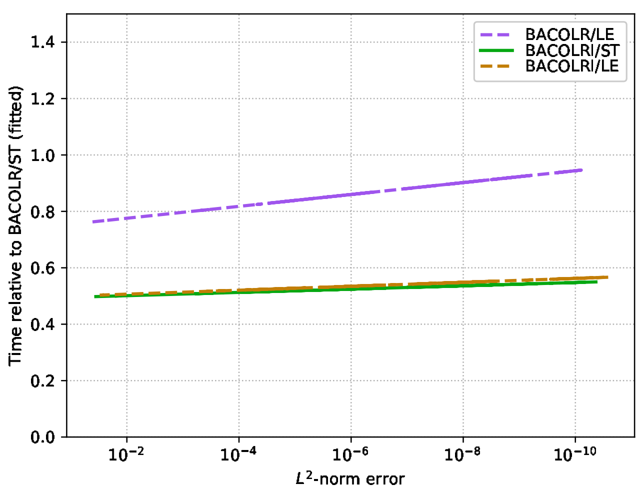

Figure 269: Rel. Work-Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 9$

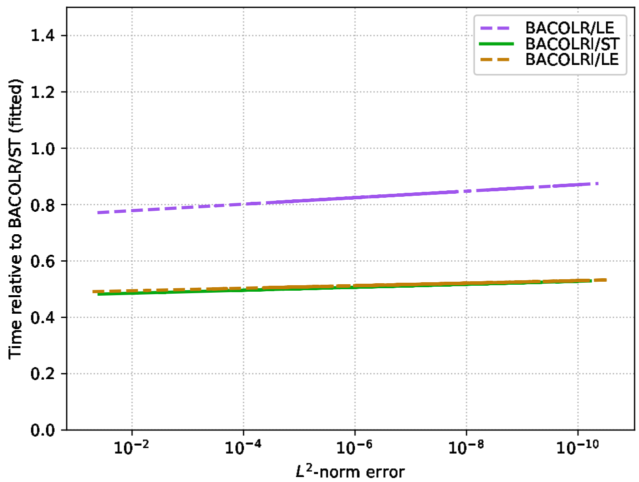

Figure 270: Rel. Work-Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 10$

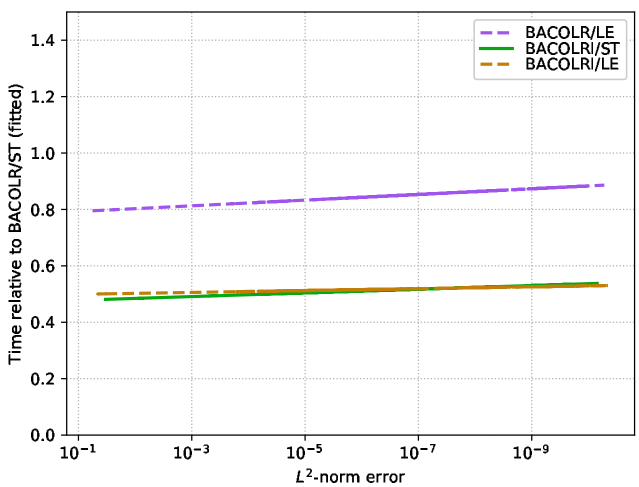

Figure 271: Rel. Work-Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 11$

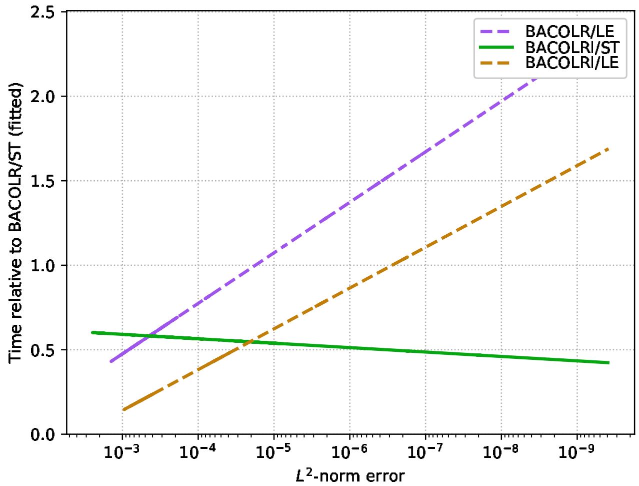

Figure 272: Rel. Work-Accuracy: Schrödinger System, $p = 4$

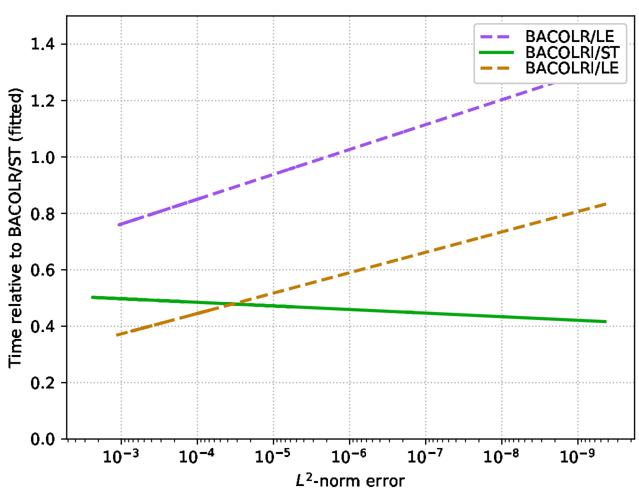

Figure 273: Rel. Work-Accuracy: Schrödinger System, $p = 5$

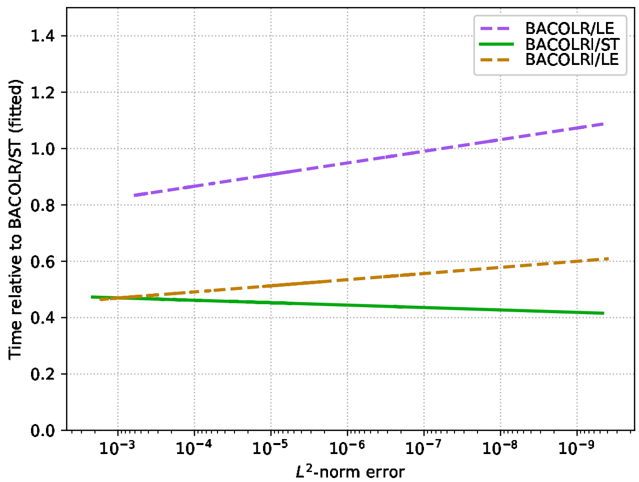

Figure 274: Rel. Work-Accuracy: Schrödinger System, $p = 6$

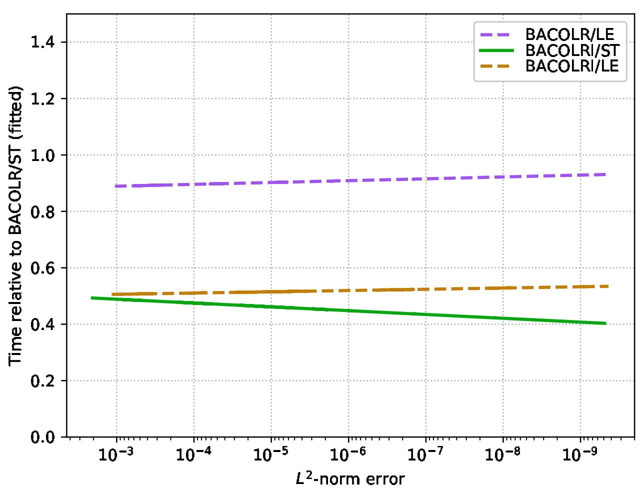

Figure 275: Rel. Work-Accuracy: Schrödinger System, $p = 7$

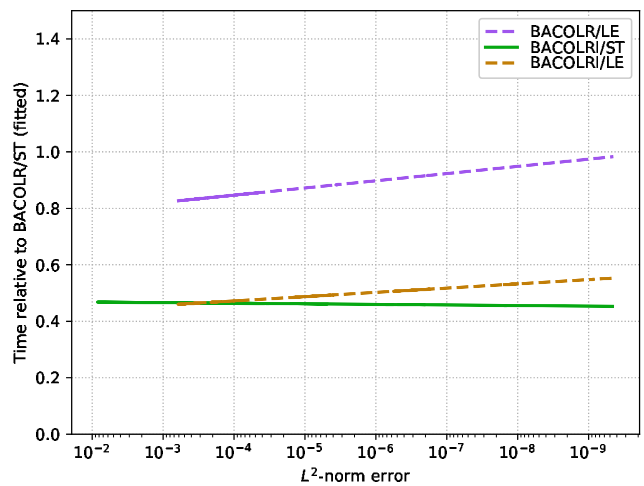

Figure 276: Rel. Work-Accuracy: Schrödinger System, $p = 8$

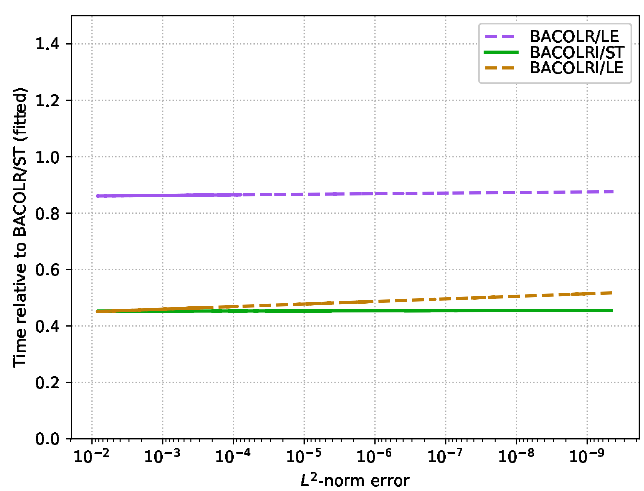

Figure 277: Rel. Work-Accuracy: Schrödinger System, $p = 9$

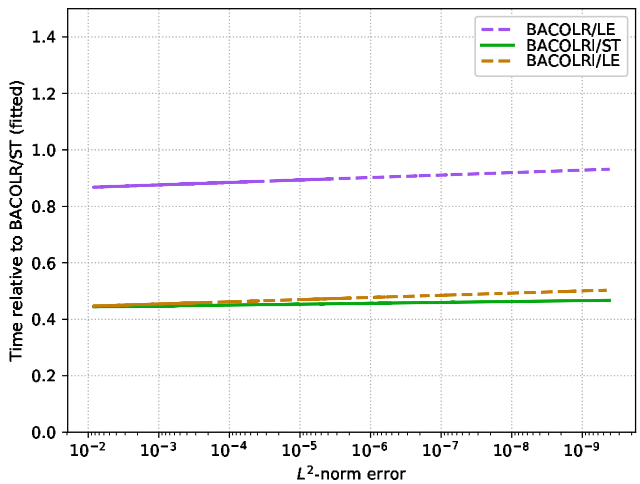

Figure 278: Rel. Work-Accuracy: Schrödinger System, $p = 10$

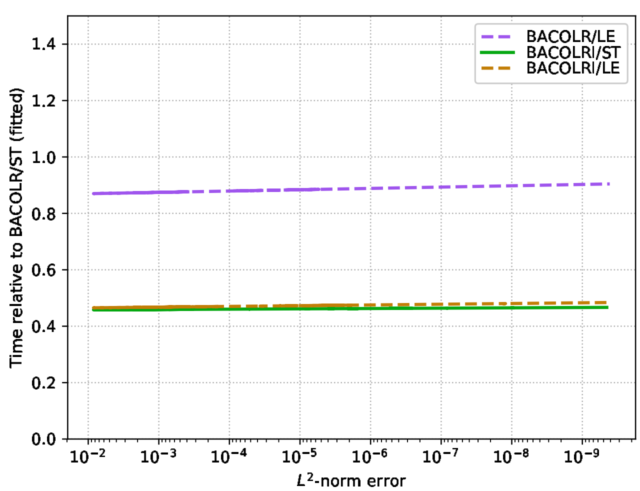

Figure 279: Rel. Work-Accuracy: Schrödinger System, $p = 11$

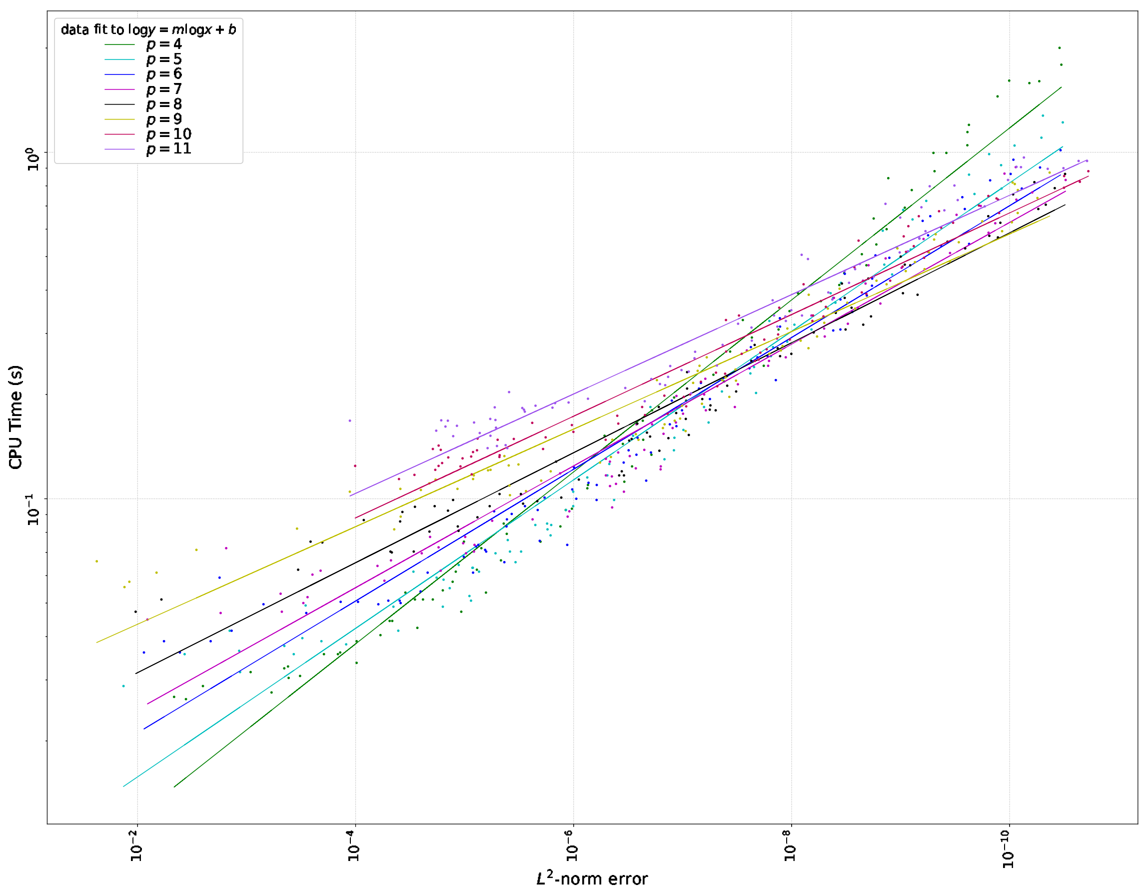

Figure 280: BACOLR/ST Work vs. Accuracy: One Layer Burgers equation $\epsilon = 10^{-3}$; $p = 4 \ldots 11$

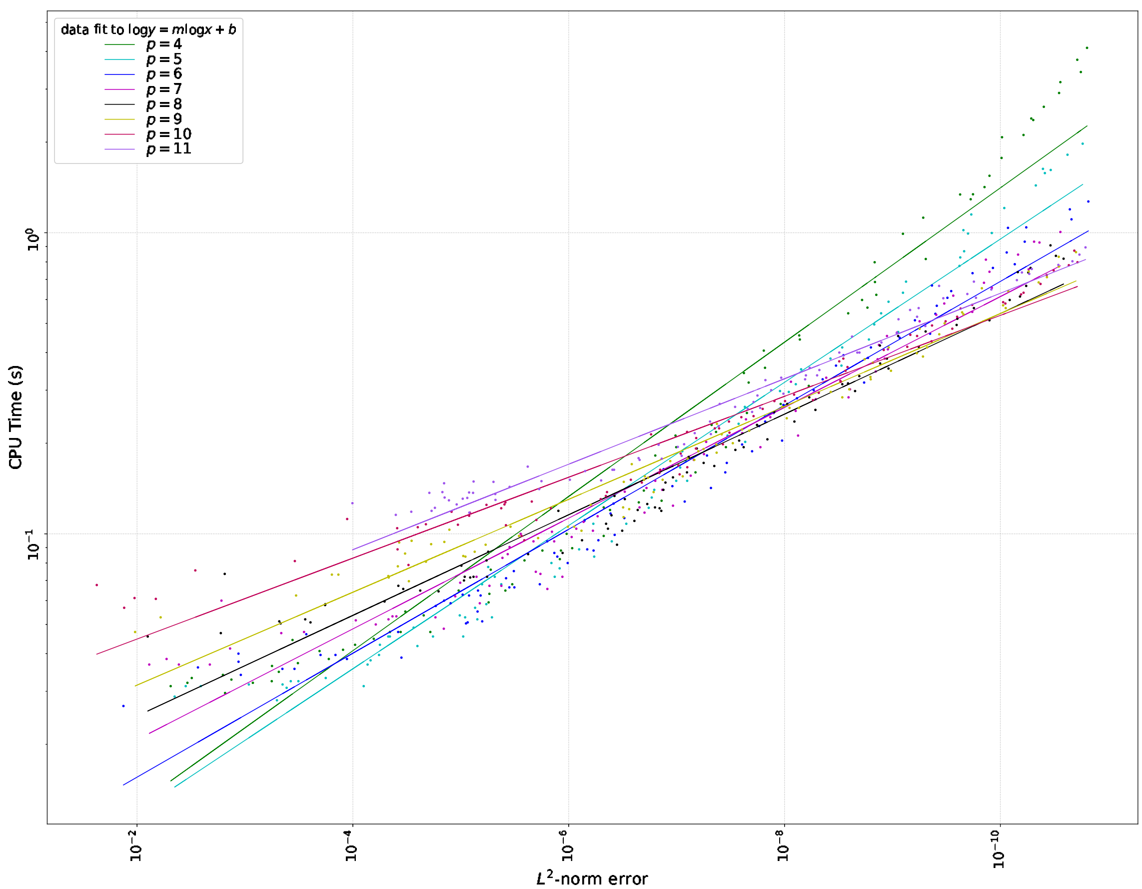

Figure 281: BACOLR/LE Work vs. Accuracy: One Layer Burgers equation $\epsilon = 10^{-3}$; $p = 4 \ldots 11$

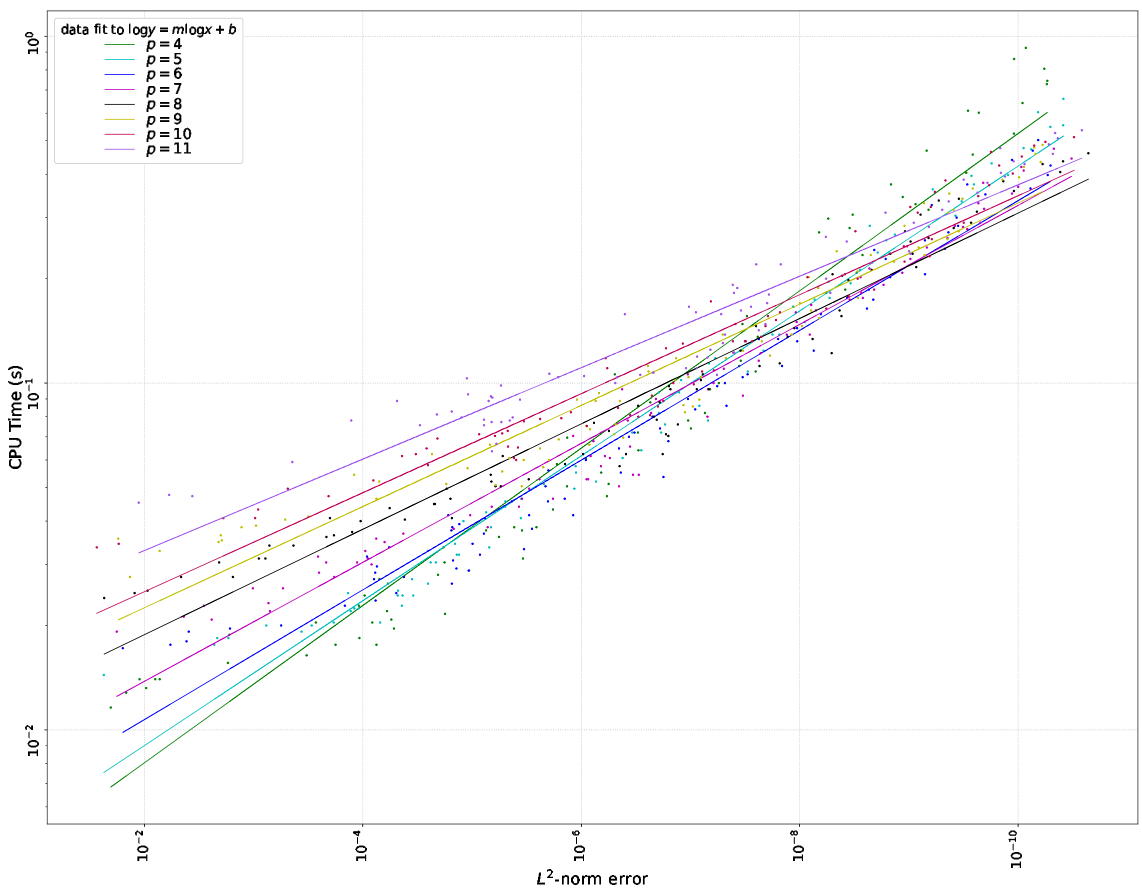

Figure 282: BACOLRI/ST Work vs. Accuracy: One Layer Burgers equation $\epsilon = 10^{-3}$; $p = 4 \ldots 11$

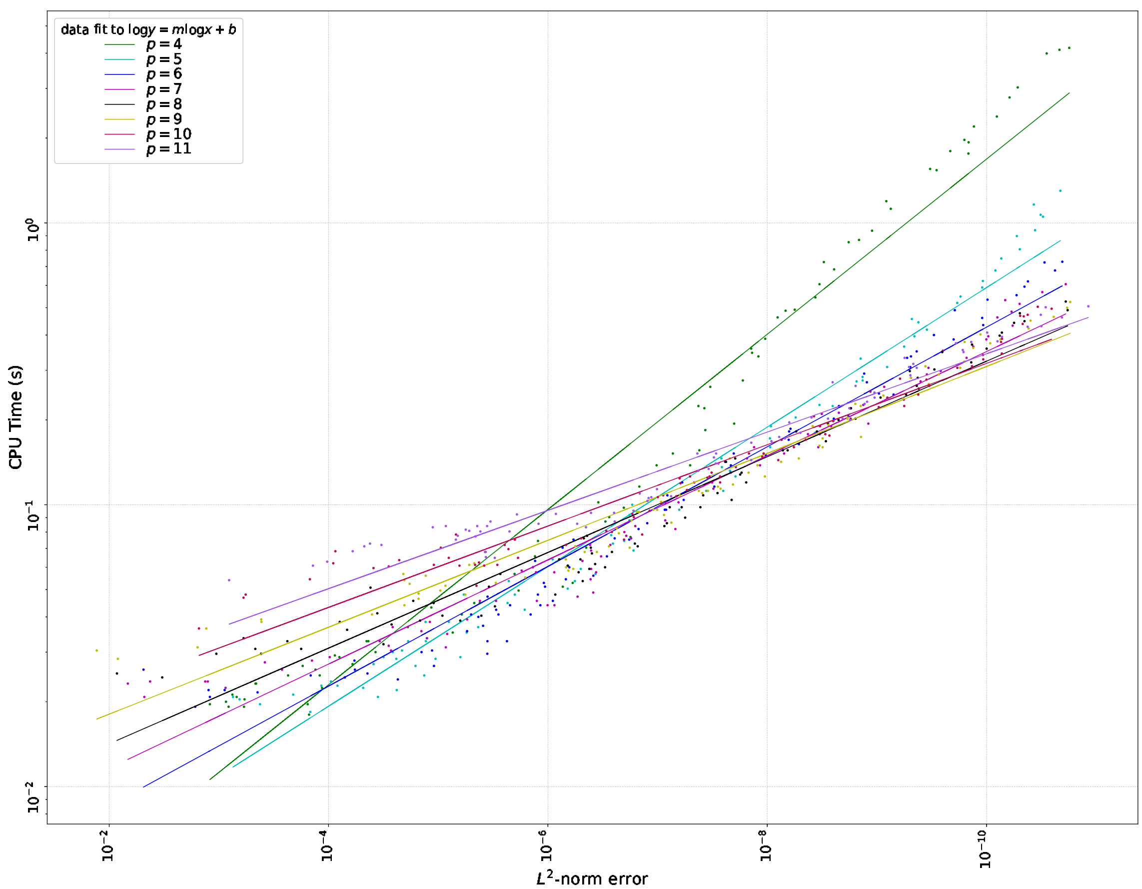

Figure 283: BACOLRI/LE Work vs. Accuracy: One Layer Burgers equation $\epsilon = 10^{-3}$; $p = 4 \ldots 11$

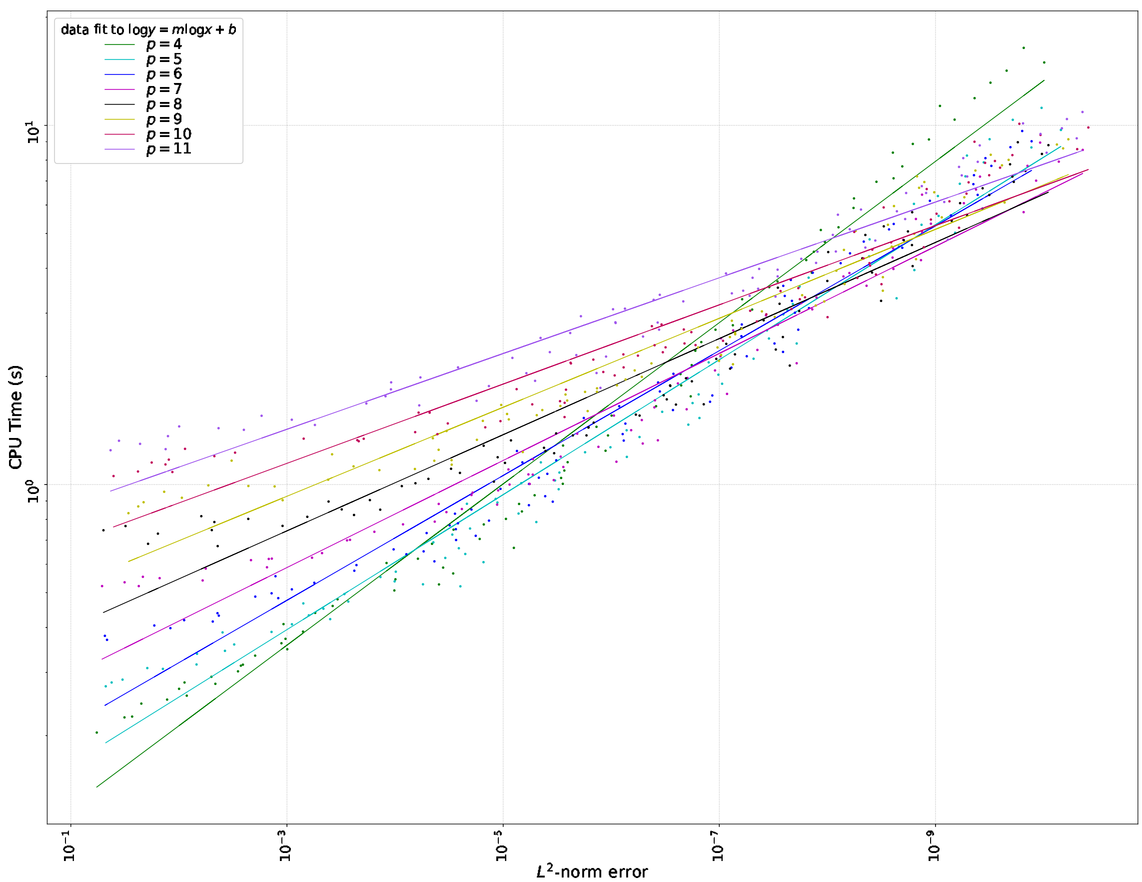

Figure 284: BACOLR/ST Work vs. Accuracy: One Layer Burgers equation $\epsilon = 10^{-4}$; $p = 4 \ldots 11$

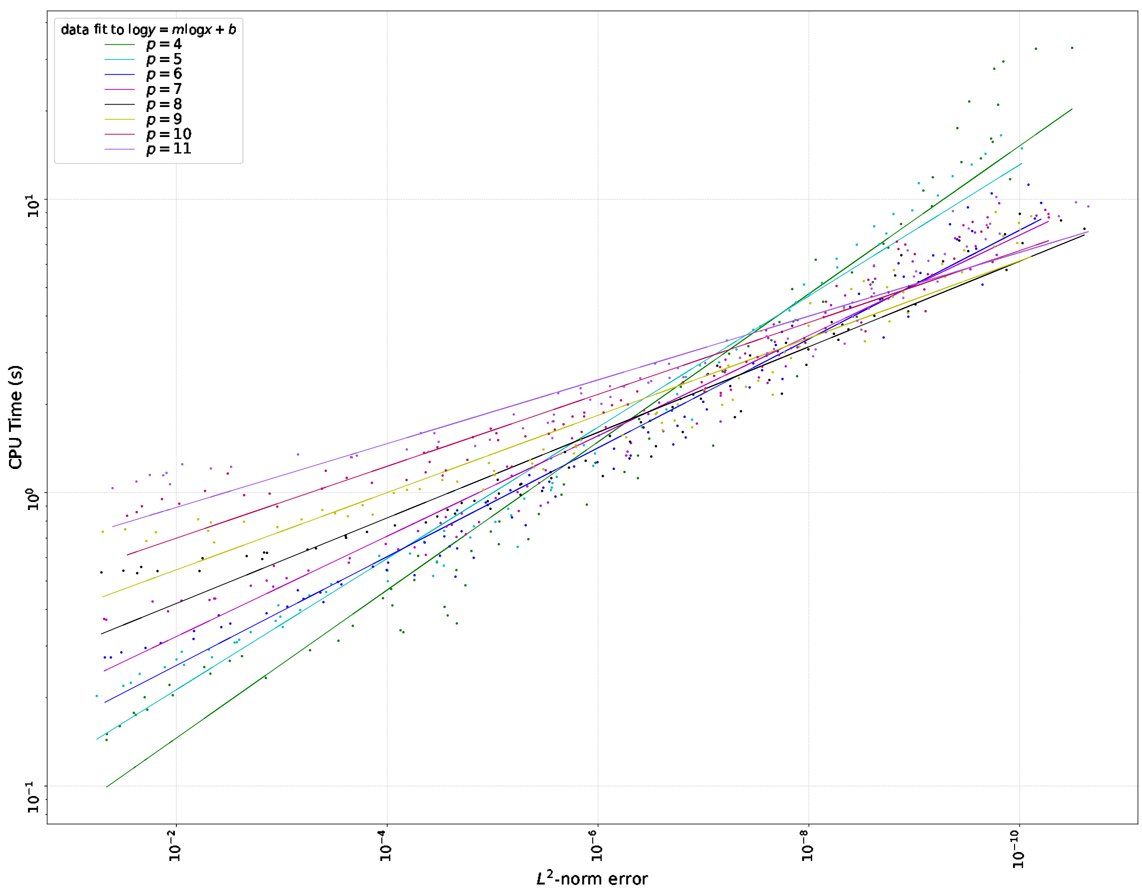

Figure 285: BACOLR/LE Work vs. Accuracy: One Layer Burgers equation $\epsilon = 10^{-4}$; $p = 4 \ldots 11$

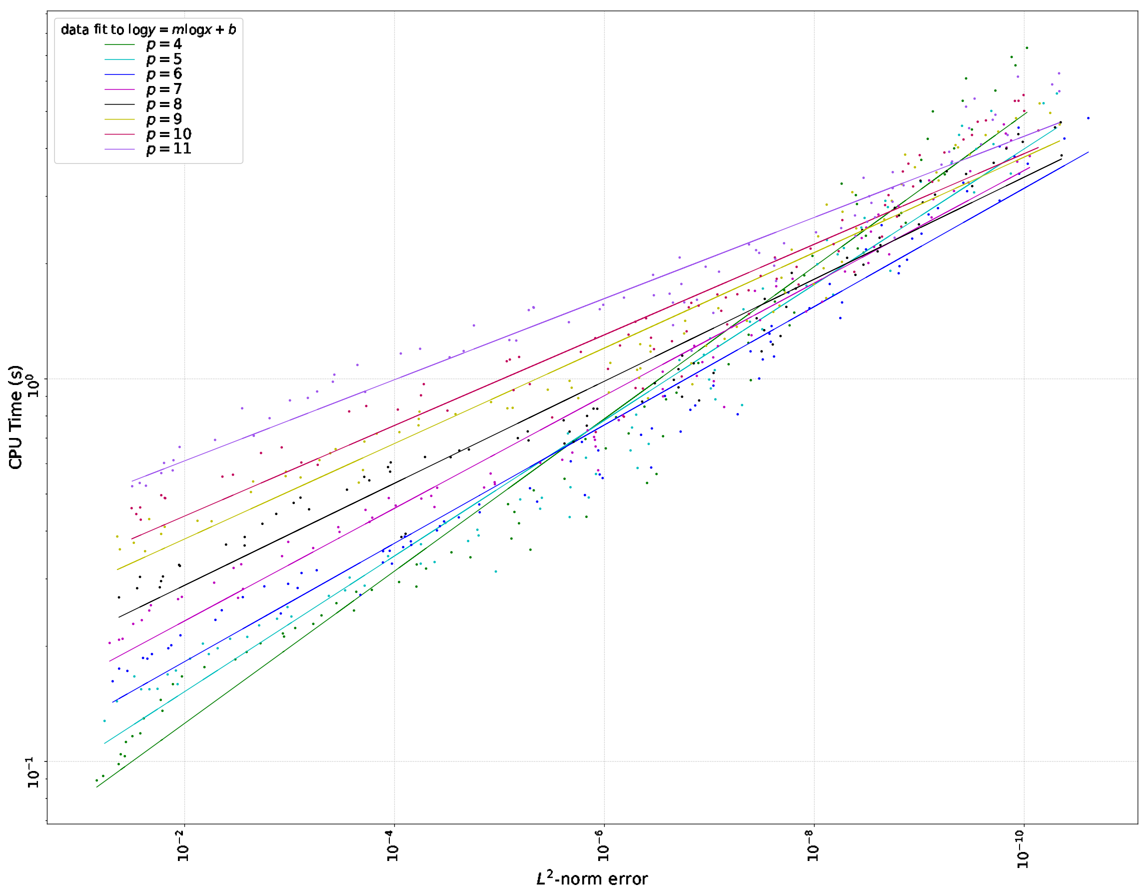

Figure 286: BACOLRI/ST Work vs. Accuracy: One Layer Burgers equation $\epsilon = 10^{-4}$; $p = 4 \ldots 11$

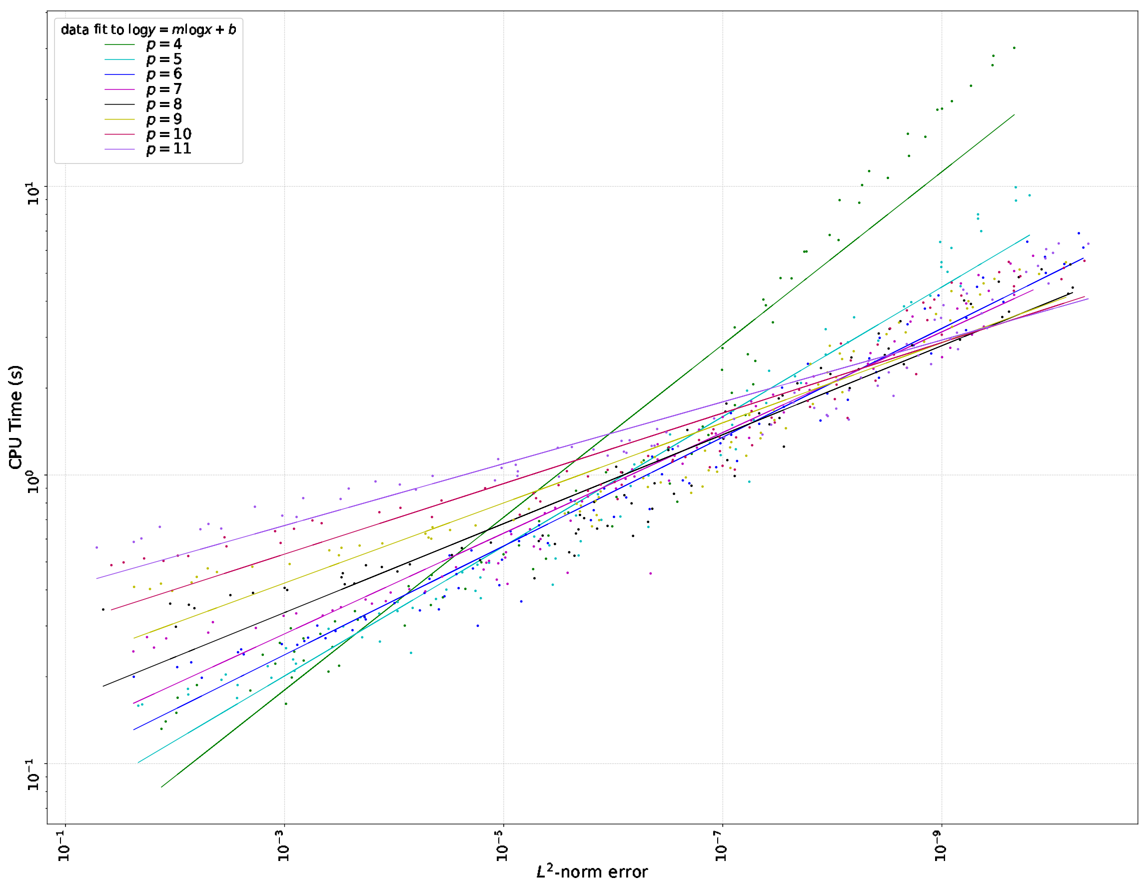

Figure 287: BACOLRI/LE Work vs. Accuracy: One Layer Burgers equation $\epsilon = 10^{-4}$; $p = 4 \ldots 11$

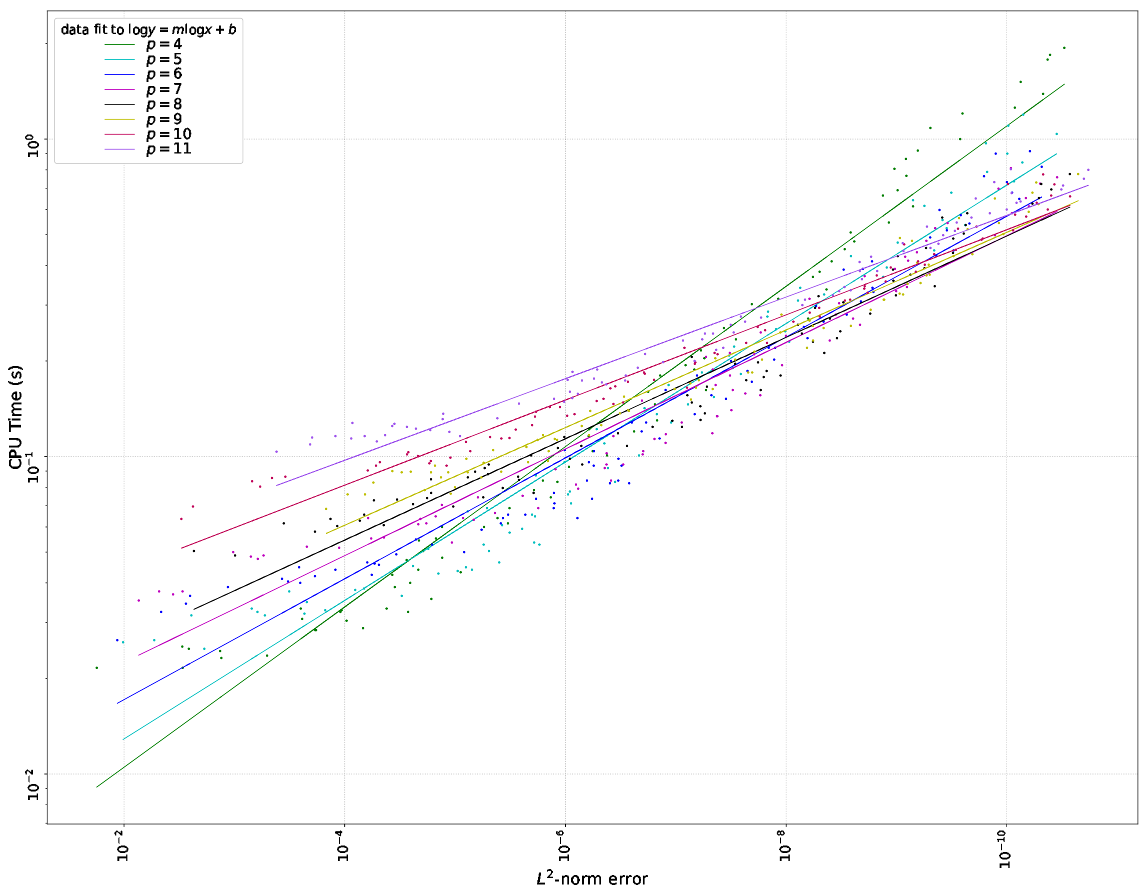

Figure 288: BACOLR/ST Work vs. Accuracy: Two Layer Burgers equation $\epsilon = 10^{-3}$; $p = 4 \ldots 11$

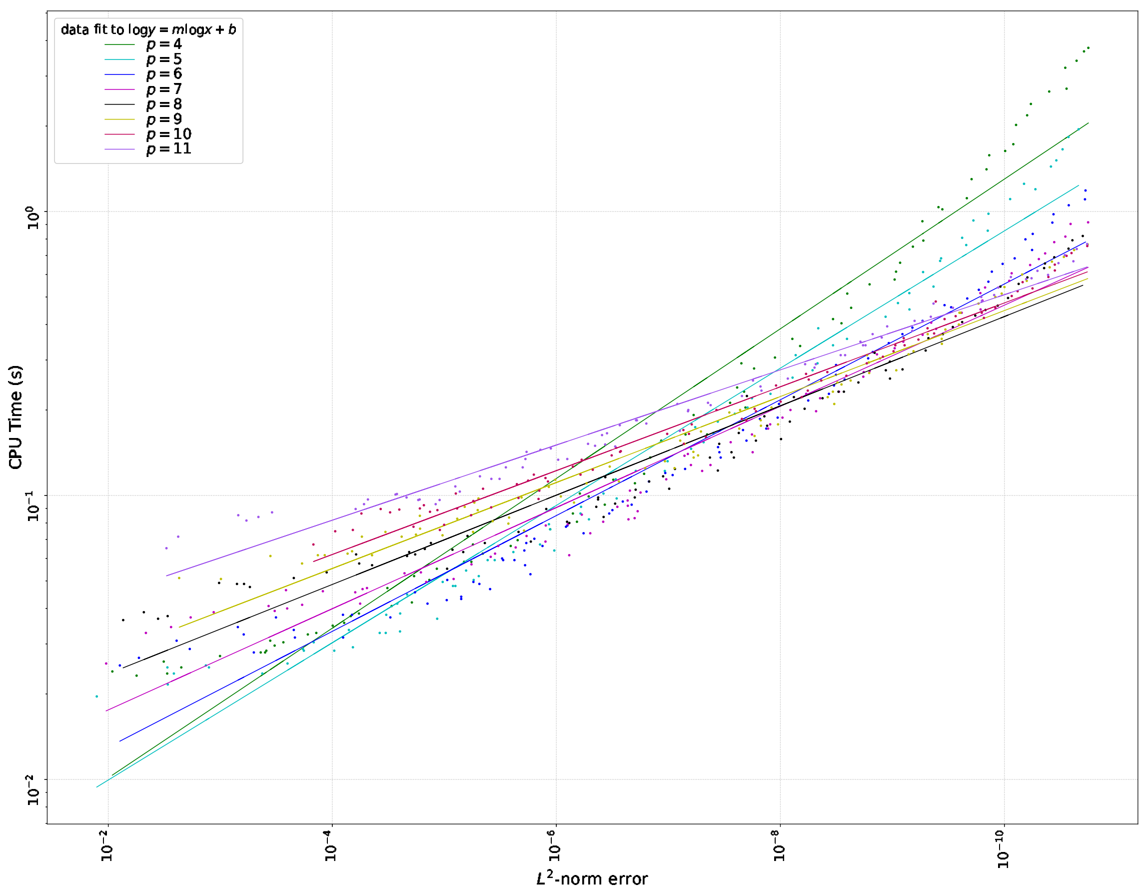

Figure 289: BACOLR/LE Work vs. Accuracy: Two Layer Burgers equation $\epsilon = 10^{-3}$; $p = 4 \ldots 11$

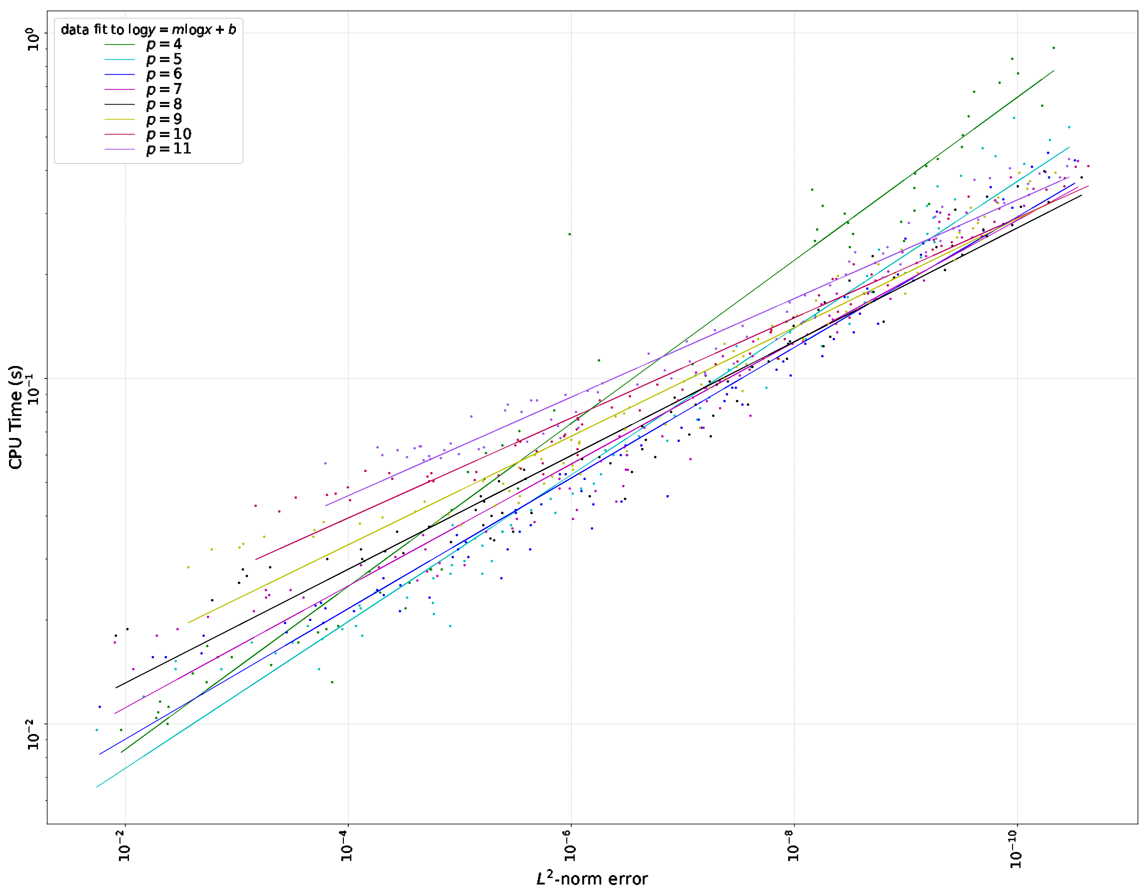

Figure 290: BACOLRI/ST Work vs. Accuracy: Two Layer Burgers equation $\epsilon = 10^{-3}$; $p = 4 \ldots 11$

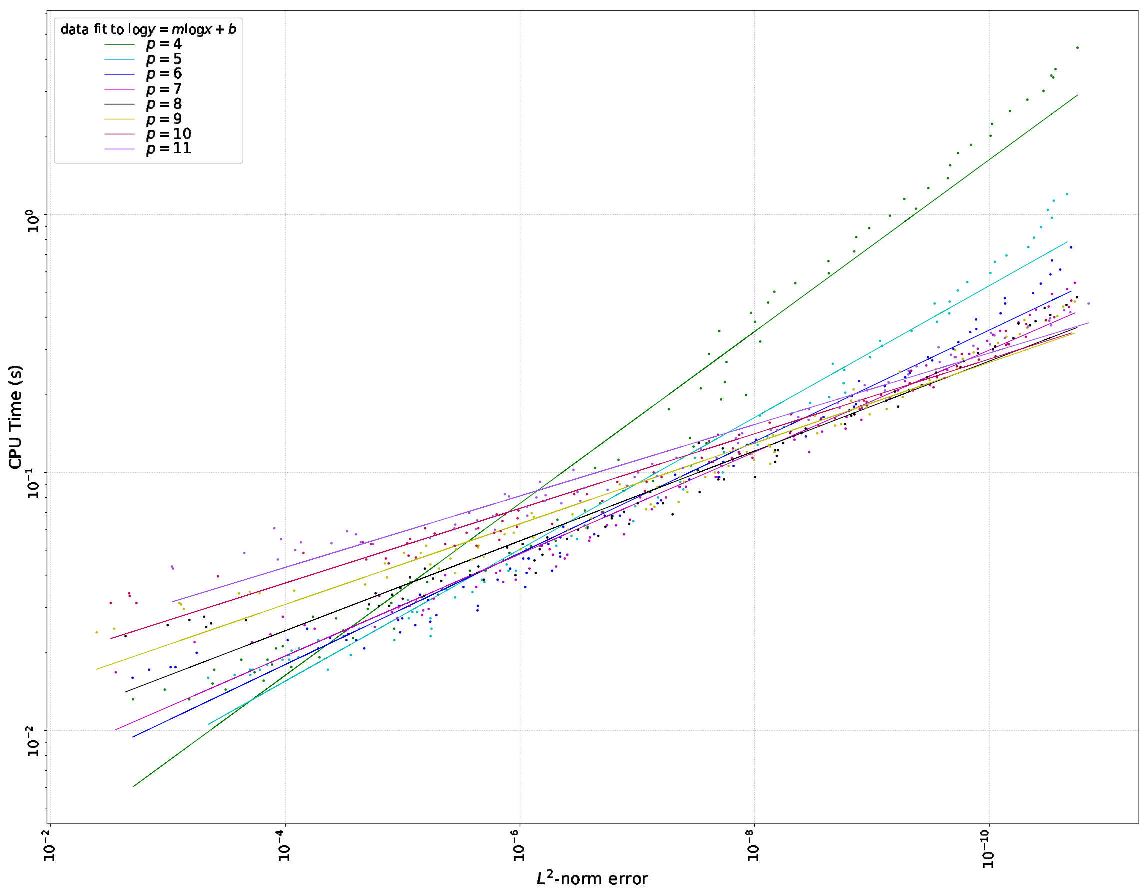

Figure 291: BACOLRI/LE Work vs. Accuracy: Two Layer Burgers equation $\epsilon = 10^{-3}$; $p = 4 \ldots 11$

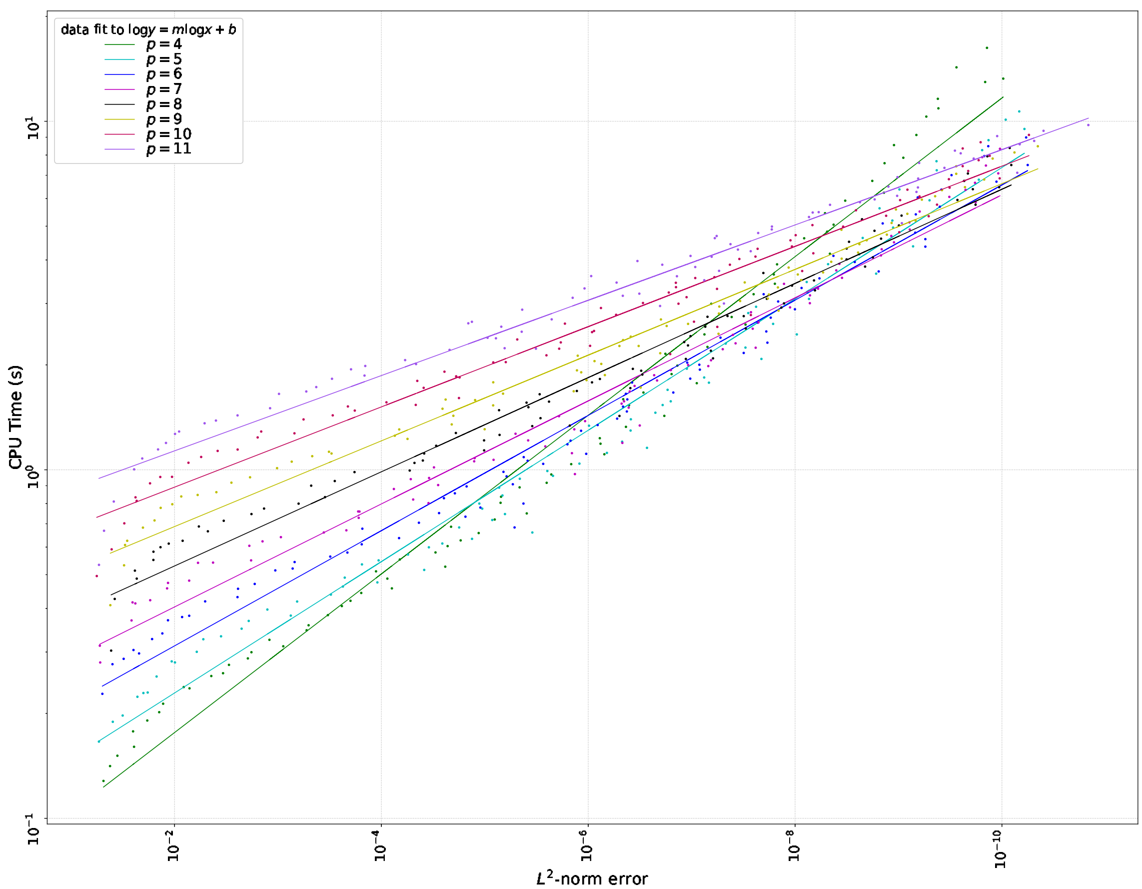

Figure 292: BACOLR/ST Work vs. Accuracy: Two Layer Burgers equation $\epsilon = 10^{-4}$; $p = 4 \ldots 11$

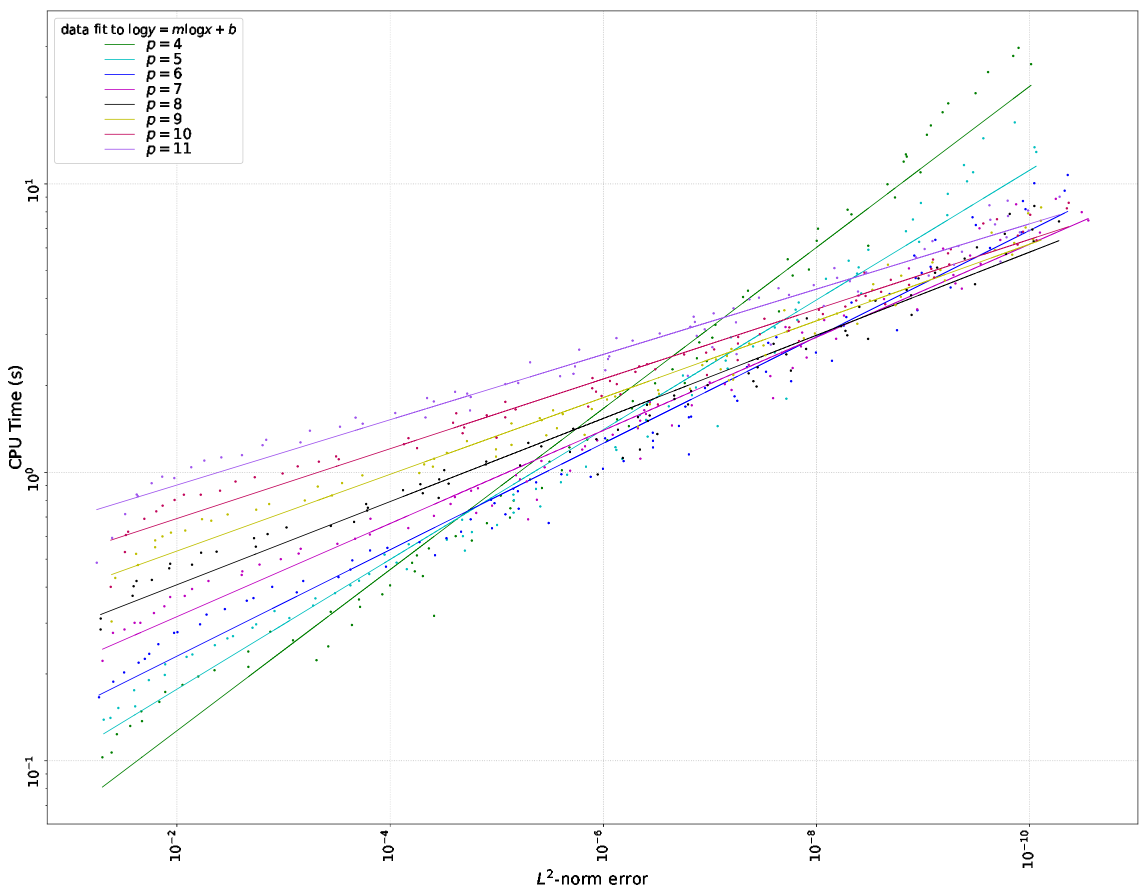

Figure 293: BACOLR/LE Work vs. Accuracy: Two Layer Burgers equation $\epsilon = 10^{-4}$; $p = 4 \ldots 11$

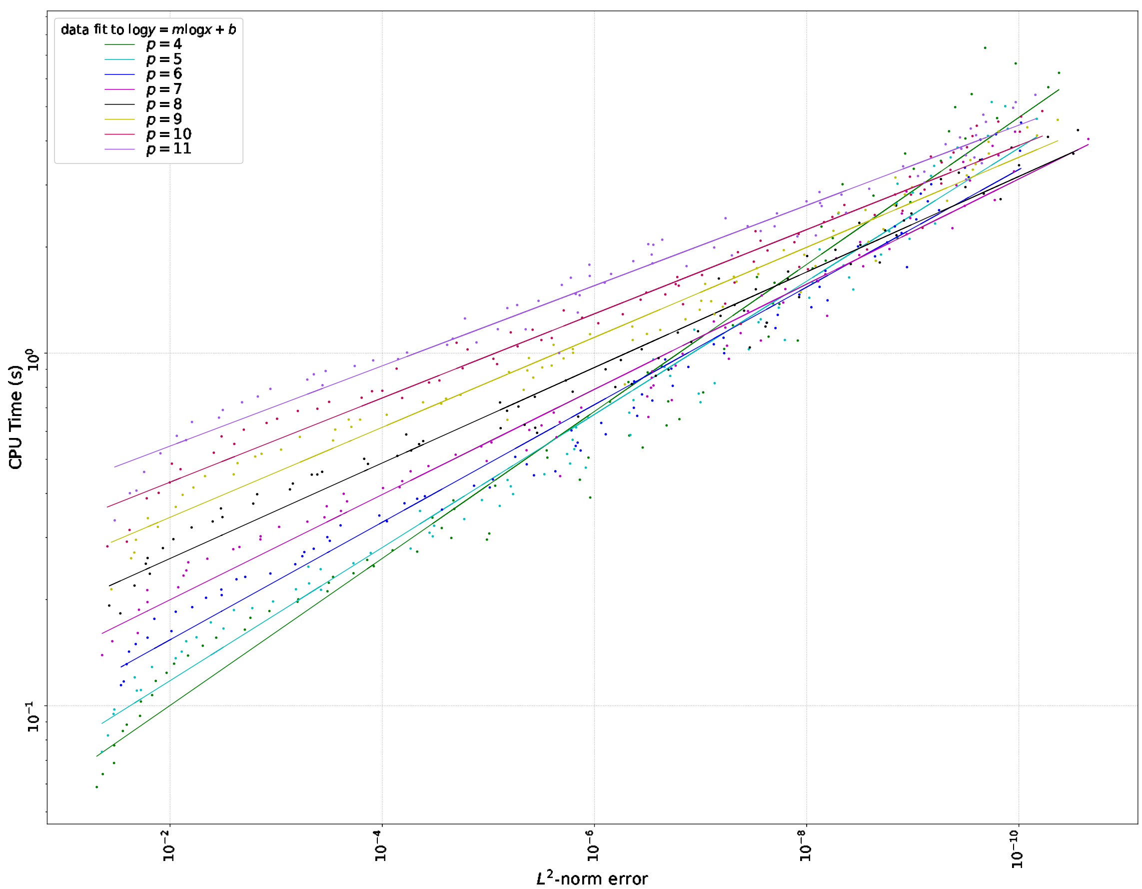

Figure 294: BACOLRI/ST Work vs. Accuracy: Two Layer Burgers equation $\epsilon = 10^{-4}$; $p = 4 \ldots 11$

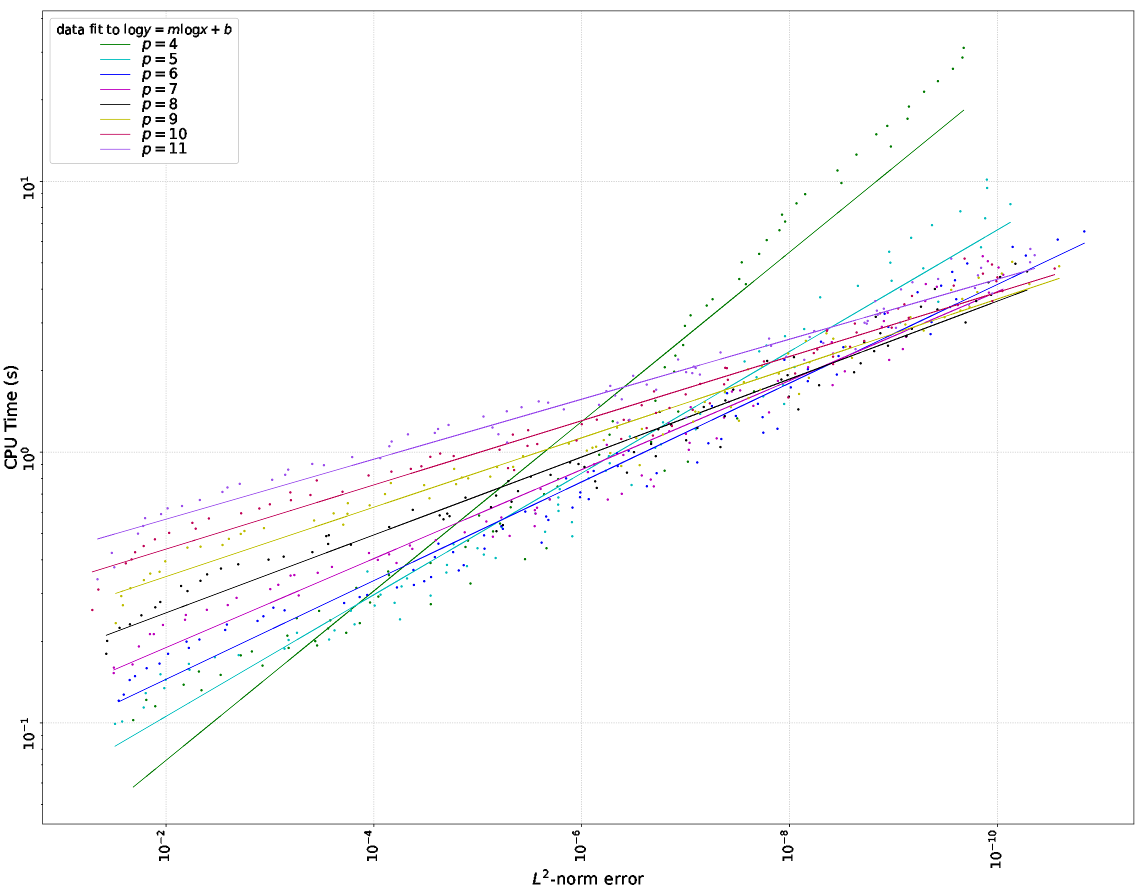

Figure 295: BACOLRI/LE Work vs. Accuracy: Two Layer Burgers equation $\epsilon = 10^{-4}$; $p = 4 \ldots 11$

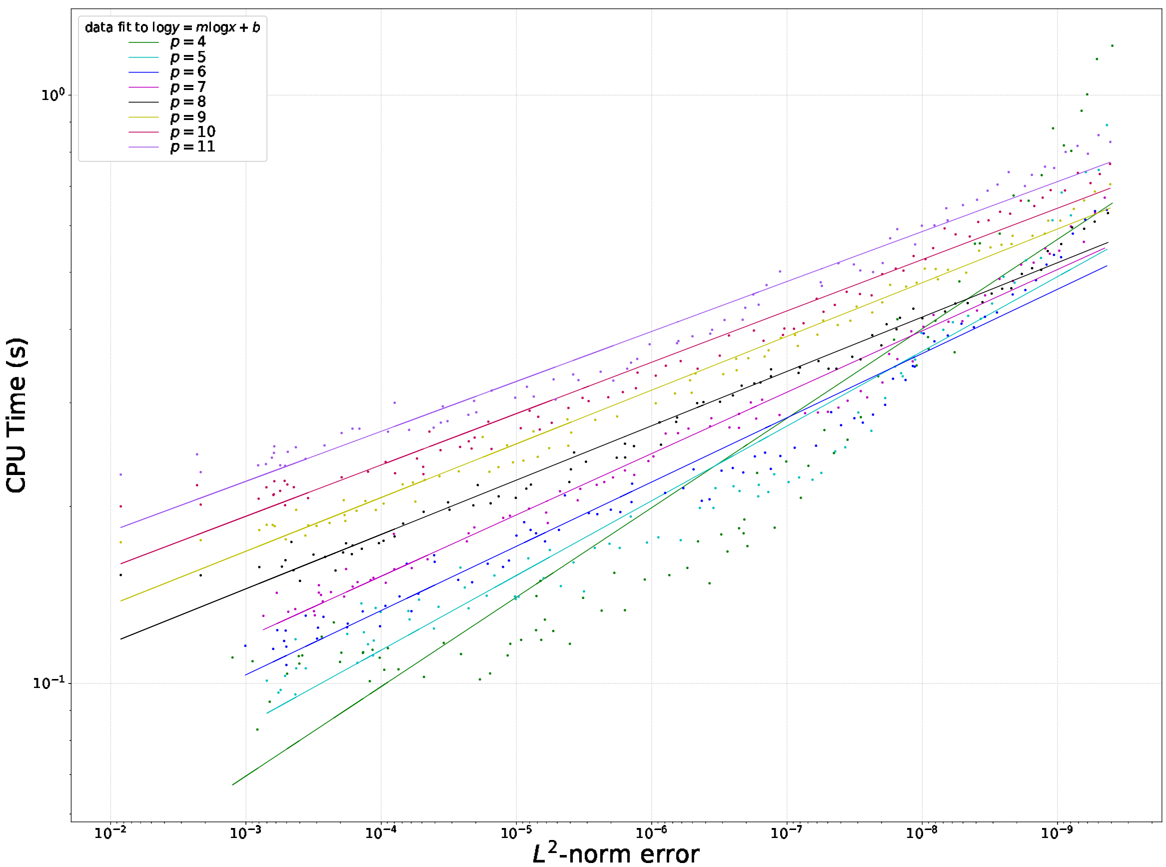

Figure 296: BACOLR/ST Work vs. Accuracy: Schrödinger System; $p = 3 \ldots 11$

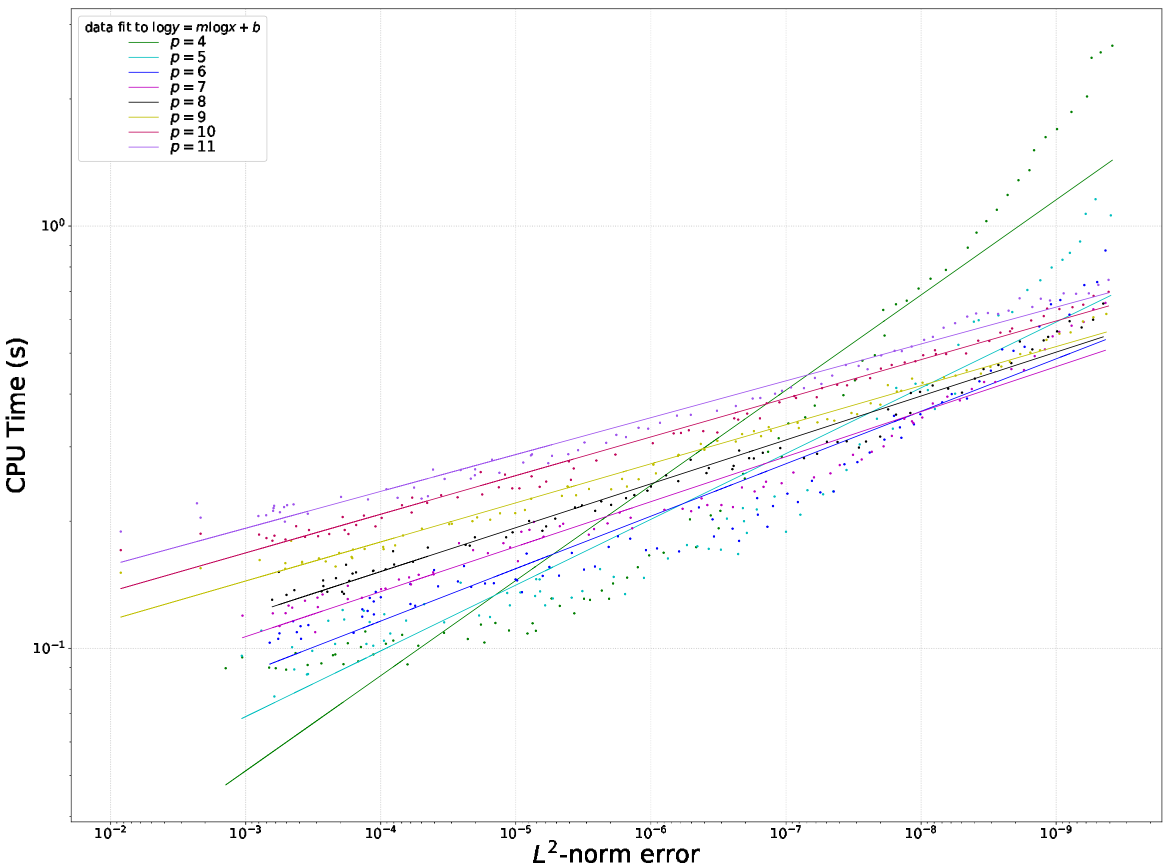

Figure 297: BACOLR/LE Work vs. Accuracy: Schrödinger System; $p = 4 \ldots 11$

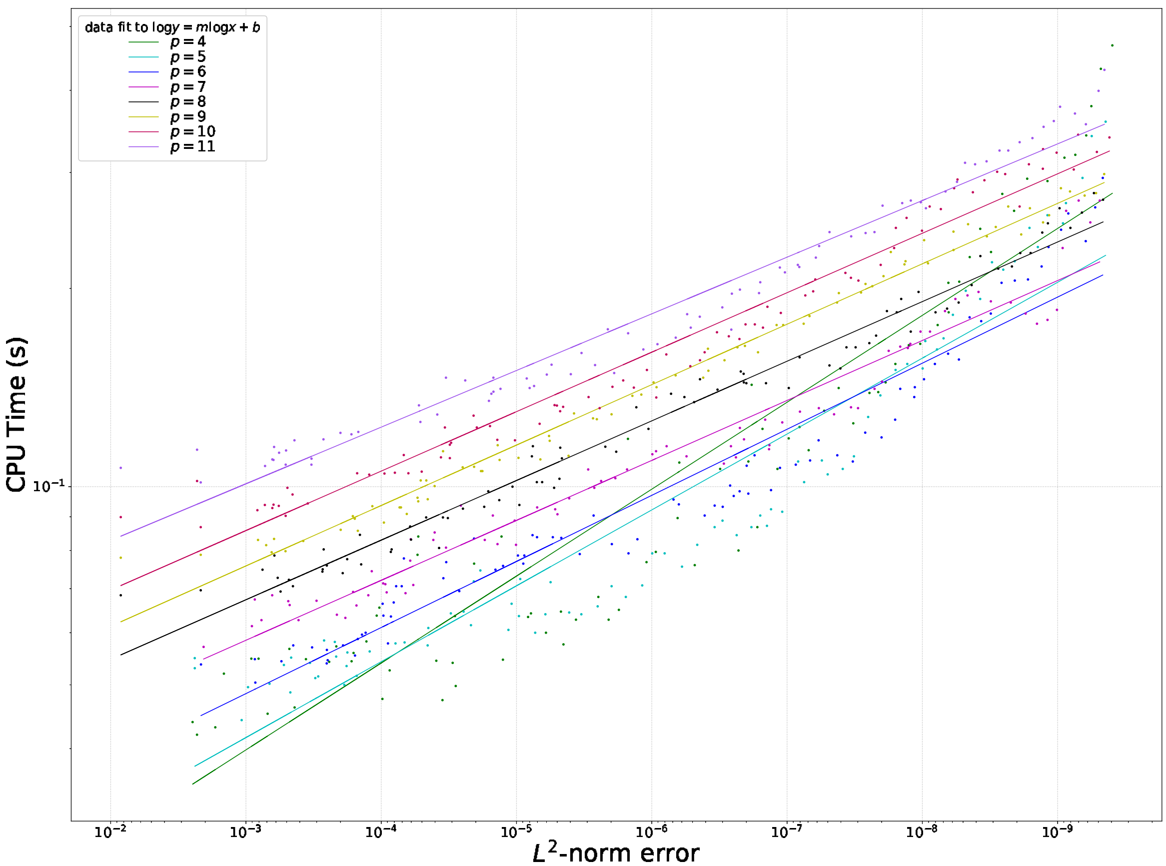

Figure 298: BACOLRI/ST Work vs. Accuracy: Schrödinger System; $p = 4 \ldots 11$

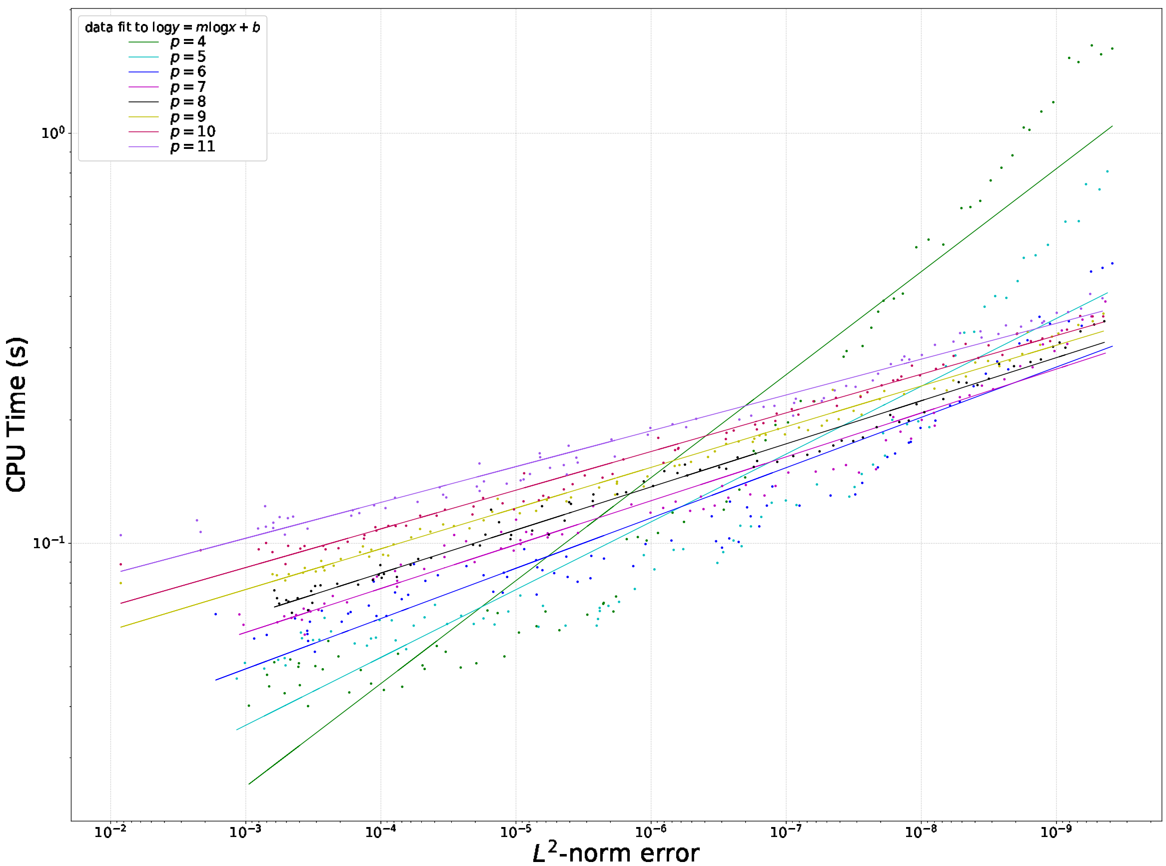

Figure 299: BACOLRI/LE Work vs. Accuracy: Schrödinger System; $p = 4 \ldots 11$