Performance Analysis of Interpolation-based Spatial Error Control B-spline Gaussian Collocation PDE Software: BDF Time Integration vs. IRK Time Integration *

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

Abstract

This report considers BACOLI and BACOLRI, two recently developed members of a family of software packages for the *error controlled* numerical solution of systems of one-dimensional partial differential equations (PDEs). The two original members of this family, BACOL and BACOLR, employ a spatial discretization scheme based on B-spline Gaussian collocation which is coupled with spatial error estimation and adaptive mesh refinement to provide spatial error control. For the error controlled time integration, BACOL employs a package called DASSL, which is based on a family of Backward Differentiation Formulas (BDFs). BACOLR, a modification of BACOL, replaces DASSL with RADAU5, which is based on a fifth order implicit Runge-Kutta method. The first of the new packages, BACOLI, was developed from BACOL; the second new package, BACOLRI, was developed from BACOLR. The fundamental difference between each new package and their corresponding earlier versions is a more efficient spatial error estimation computation based on the introduction of two new interpolation-based schemes. In this report, we provide extensive numerical results to compare the performances of the two new packages. We show that BACOLRI has comparable performance to that of BACOLI on a standard set of test problems and that, for problems where BACOLI fails due to stability issues associated with the BDF time integration methods, it is able to efficiently compute error controlled numerical solutions.

Subject Classification: 65L06, 65L10, 65L80, 65M20, 65M70
Keywords: B-Splines, Collocation, Interpolation, Error Estimation, Error Control, Partial differential equations, Efficiency.

*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

1 Introduction

In this report we consider B-spline [7] Gaussian collocation software that implements adaptive control of estimates of the spatial and temporal errors for 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. 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,

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

This reports focuses on two recently developed members of a family of error control B-spline Gaussian collocation software packages for the numerical solution of systems of PDEs having the above form. The earliest member of this family is BACOL [20, 22], which was developed about 15 years ago. BACOL uses the DAE solver DASSL [5], which is based on a family of multi-step methods known as Backward Differentiation Formulas (BDFs) [5]. 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 [21]. The second member of this family, developed about 10 years ago, is BACOLR;

this package is a modification of BACOL that replaces DASSL with the DAE solver RADAU5 [11], which is based on a 5th order implicit Runge-Kutta (IRK) method of Radau IIA type [11]. In [19], 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 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 [15], released about three years ago. In BACOLI, the computation of the degree $p+1$ numerical solution is replaced with the more efficient computation of an interpolant to the degree p numerical solution. BACOLI provides two new interpolation-based spatial error estimation schemes, each of which employs a corresponding spatial error control mode. The BACOLI project also includes a Fortran 95 wrapper that greatly simplifies the use of BACOLI compared to what is required to use BACOL.

The most recently developed member of this family of software packages is a modified version of BACOLR known as BACOLRI [1]. This package was developed from BACOLR; the computation of the degree $p+1$ numerical solution is replaced with an option to use one of two new interpolants that then serve as the basis for the spatial error estimation scheme. These new interpolants are the same ones that have been implemented in BACOLI. This project also includes a Fortran 95 wrapper that greatly simplifies the use of BACOLRI compared to what is required to use BACOLR. Because it is based on BACOLR, BACOLRI also avoids the stability issues that arise for BDFs for certain types of problems such as Schrödinger problems.

The following diagram shows the relationship among the four packages.

$$\begin{array}{ccccc}
\textbf{BACOL} & \Rightarrow & \textit{Improved Temporal Stability} & \Rightarrow & \textbf{BACOLR} \\
\Downarrow & & & & \Downarrow \\
\textit{Improved} & & & & \textit{Improved} \\
\textit{Spatial} & & & & \textit{Spatial} \\
\textit{Error} & & & & \textit{Error} \\
\textit{Estimate} & & & & \textit{Estimate} \\
\Downarrow & & & & \Downarrow \\
\textbf{BACOLI} & \Rightarrow & \textit{Improved Temporal Stability} & \Rightarrow & \textbf{BACOLRI}
\end{array}$$

As mentioned earlier, BACOL was compared with a number of comparable packages in [21], and BACOLR was compared with BACOL in [19]. The recent papers [15, 18] and report [17] compare BACOL with BACOLI and the recent report [16] compares BACOLR with BACOLRI. The purpose of this report is to complete the comparison process by providing numerical results that compare the two newest packages, BACOLI and BACOLRI.

This report is organized as follows. In Section 2, we provide a review of the details of the algorithms implemented in BACOL, BACOLR, BACOLI, and BACOLRI. In Section 3, we provide numerical results based on an extensive set of tests performed to compare BACOLI with BACOLRI. We close, in Section 4, with a summary, our conclusions, and suggestions for future work.

2 Review of BACOL Software Family

BACOL, BACOLR, BACOLI, and BACOLRI 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 positive integer p ($4 \le p \le 11$), 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 a DAE solver which solves the DAE system consisting of the collocation conditions, (6), and boundary conditions, (7). Once these coefficients 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 $h = \max_{i=1}^{NINT} h_i$ [9, 6].

As mentioned in the previous section, BACOL and BACOLI solve (6), (7), using DASSL while BACOLR and BACOLRI use 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 [8]. In BACOL, BACOLR, BACOLI, and BACOLRI these linear systems are treated using the software package, COLROW [8], 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 and BACOLRI, because of the type of IRK method employed by RADAU5, it is also necessary to solve ABD systems that involve complex numbers. BACOLR and BACOLRI therefore also employ the complex version of COLROW known as COMPLEXCOLROW [13]; the routine CCRCMP performs factorizations of the complex ABD coefficient matrices, while the routine CCRSLV performs backsolves of the factored complex ABD systems. See [19], pages 15:8-15:9, for further details.

After each accepted time step, the BACOL, BACOLR, BACOLI, and BACOLRI 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 [22] for further details.

As mentioned in the previous section, BACOL and BACOLR obtain the spatial error estimates by computing a second approximate solution, $\overline{\underline{U}}(x,t)$, on the same spatial mesh, using the B-spline collocation spatial discretization algorithm described earlier, followed by the solution of the corresponding 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 on [0,1] mapped onto to each spatial subinterval. This implies that the spatial error of $\overline{\underline{U}}(x,t)$ is $O(h^{p+2})$. A scaled difference of $\underline{U}(x,t)$ and $\overline{\underline{U}}(x,t)$ is then computed to provide a spatial error estimate for $\underline{U}(x,t)$ [22]. The computation of $\overline{\underline{U}}(x,t)$, 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 $\overline{\underline{U}}(x,t)$ and obtain a spatial error estimate in a more efficient manner. One direction of investigation [2] 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; see [2] for further details; 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 [2]) 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 (see [2]) 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. In this approach, the computation of $\overline{\underline{U}}(x,t)$ is replaced by the construction of the SCI and the latter then replaces $\overline{\underline{U}}(x,t)$ in the computation of the spatial error estimate for $\underline{U}(x,t)$.

As mentioned 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 $\overline{\underline{U}}(x,t)$, of order $p+2$. The higher order solution, $\overline{\underline{U}}(x,t)$, 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 for BACOL and BACOLR to return the higher order, i.e., more accurate, numerical solution, $\overline{\underline{U}}(x,t)$. This could be done with a simple modification to BACOL or BACOLR since these packages compute both $\underline{U}(x,t)$ and $\overline{\underline{U}}(x,t)$ for every time step. However, while either of these solutions could be returned to the user, the difference between the two gives a spatial error estimate only for $\underline{U}(x,t)$. *In the case where the higher order solution $\overline{\underline{U}}(x,t)$ 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., [10] - 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 $\overline{\underline{U}}(x,t)$ 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 (at least asymptotically) would be the same as that of $\underline{U}(x,t)$. Then the difference between $\overline{\underline{U}}(x,t)$ and this interpolant would provide an estimate that would be the same (at least asymptotically) as is currently computed in BACOL or BACOLR. In this case, $\overline{\underline{U}}(x,t)$ 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 interpolates $\overline{\underline{U}}(x,t)$ and $\overline{\underline{U}}_x(x,t)$

at the mesh points but the remaining interpolation points (which are internal to the subinterval) at which $\overline{U}(x,t)$ is interpolated, 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 $\underline{U}(x,t)$ on each subinterval has a known form - see, e.g., [4] - 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 $\overline{U}(x,t)$ and $\overline{U}_x(x,t)$ 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 $\overline{U}(x,t)$. This interpolant is referred to as the Lower Order Interpolant (LOI). See [3] for further details. A scaled difference of $\overline{U}(x,t)$ and the LOI then gives the spatial error estimate. As mentioned above, since $\overline{U}(x,t)$ 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 as being associated with the case where $\overline{U}(x,t)$ is the returned solution, in fact the situation is somewhat simpler. When BACOLI or BACOLRI 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 the codes construct the SCI based on this degree p numerical solution and use this interpolant to generate a spatial error estimate which is then used as the basis for ST spatial error control. If the LE spatial error control mode is chosen, then the codes construct the LOI based on this degree p numerical solution and use this interpolant to generate a spatial error estimate which is then used to provide LE spatial error control. 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 [14] and the paper [15] describe the modifications made to BACOL to replace the computation of the higher order approximate solution with the SCI and LOI schemes in order to obtain BACOLI. The report [16] describes the modifications made to BACOLR to replace the computation of the higher order approximate solution with the SCI and LOI schemes in order to obtain BACOLRI.

The report [17] 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. Similarly, the report [16] provides extensive numerical results comparing BACOLR and BACOLRI, in both ST and LE modes on a standard set of test problems over a range of tolerances and p values. The results from these reports show that, generally, BACOLI is approximately twice as fast as BACOL, and that BACOLRI is approximately twice as fast as BACOLR.

3 Numerical Results

In this section, we present results from numerical experiments that show the performance of BACOLI 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:

- **(BACOLI/ST):** BACOLI using the SCI Spatial Error Estimation Scheme, in ST Spatial Error Control Mode,
- **(BACOLI/LE):** BACOLI using the LOI Spatial Error Estimation Scheme, 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/tolerance vs. execution time comparisons. The performance of the spatial error estimation schemes and corresponding spatial error control modes will also be considered. 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.*

3.1 Test Problems

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

- **OLBE:** One Layer Burgers Equation [21]:

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

  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 [21]: 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 larger versions 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 [23]:

$$\begin{aligned}
(u_1)_t &= -(u_1)_x + n(D_1 u_3 - A_1 u_1 \gamma) + (u_1)_{xx}/Pe_1, \\
(u_2)_t &= -(u_2)_x + n(D_2 u_4 - A_2 u_2 \gamma) + (u_2)_{xx}/Pe_1, \\
(u_3)_t &= A_1 u_1 \gamma - D_1 u_3 - R u_3 u_4 \gamma^2 + (u_3)_{xx}/Pe_2, \\
(u_4)_t &= A_2 u_2 \gamma - D_2 u_4 - R u_3 u_4 \gamma^2 + (u_4)_{xx}/Pe_2,
\end{aligned} \qquad (10)$$

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 $t = 0$ $(0 \le x \le 1)$ 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 [12]:

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

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, for $a \le x \le b$, are, given by

$$
\begin{aligned}
u_1(x,0) &= \sqrt{\frac{2\kappa}{1+\rho}}\,\mathrm{sech}\left(\sqrt{2\kappa}x\right) e^{i((\phi-\eta)x)}, \\
u_2(x,0) &= \sqrt{\frac{2\kappa}{1+\rho}}\,\mathrm{sech}\left(\sqrt{2\kappa}x\right) e^{i((\phi+\eta)x)},
\end{aligned}
$$

where κ and ϕ are constants. For our numerical experiments, we choose $\phi = 1$, $\eta = 1/2$, $\rho = 2/3$ and $\kappa = 1$. 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 $b = 90$. We solve this problem from $t_0 = 0$ to $t_{end} = 1$. For this problem, we need to determine the errors of the numerical solutions that are computed by the software packages; we therefore compute a high accuracy numerical reference solution with BACOLR using a tolerance of 10^{-16} and extended precision.

This problem is an example of the type of problem that causes difficulties for BACOLI due to stability issues associated with the BDFs upon which DASSL is based. Unless we restrict DASSL to use only the first and second order BDFs (these BDFs are not subject to the stability issues that arise for the higher order BDFs), BACOLI will not return in a reasonable amount of time when trying to compute a solution to this problem. If we do restrict DASSL to use only first and second order BDFs, BACOLI will return but the numerical solutions provided by the code do not come close to meeting the user-specified tolerance due to inaccuracies in the computation of the B-spline coefficients. We provide evidence later in this report to support this observation.

3.2 Machine Independent Performance Measures by Tolerance

In this subsection, we compare BACOLI/ST, BACOLI/LE, BACOLRI/ST, and BACOLRI/LE with respect to several *machine independent* measures of the algorithms employed in the codes that contribute significantly to their 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 note, of course, that for BACOLI, there will be no calls to either CCRCMP or CCRSLV.

In this set of tests we consider numerical experiments in which each code is applied to one of the test problems for a given p and *tol* combination. 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**. Since BACOLI is not able to compute error controlled solution for the **SCHR** problem, we do not include results for BACOLI for this problem.

Tables 5-14 give machine independent performance measures for the four codes (except for BACOLI/ST and BACOLI/LE for the **SCHR** problem), for $p = 4, 5, 7, 9$, and $tol = 10^{-4}, 10^{-6}, 10^{-8}$. For each table entry the first row gives **Final NINT, Accepted Time Steps**, and **Remeshings** and the second row gives [**Calls to CRDCMP, Calls to CRSLVE**]. For BACOLRI/ST and BACOLRI/LE, each table entry has a third row which gives {**Calls to CCRCMP, Calls to CCRSLV**}. (We note that for **TLBE**, **TLBEx6**, and **TLBEx12**, with $\epsilon = 10^{-3}$, $p = 4$ and $tol = 10^{-6}$, BACOLI/ST and BACOLRI/ST fail; in these cases the corresponding table entries are blank.)

In order to assist with the understanding of the results from Tables 5-14, we also present figures that provide visualizations of some of the tabular data:

- In Figures 1-38, we plot **Final NINT** vs. *tol* for all for codes for $p = 4, ..., 11$, and $tol = 10^{-2}, 10^{-3}, \ldots, 10^{-10}$. There is one plot for each code and problem combination for which results appear in the tables.

  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. As well, we note that, for a given tolerance, the **Final NINT** values decrease as p gets larger. *For small p and sharper tolerances, the LE codes have* **Final NINT** *values that are 1.5 to 3 times larger than those of the ST codes.* There is no significant difference between the ST codes or between the LE codes with respect to **Final NINT** values.

- In Figures 39-76, we plot **Accepted Time Steps** vs. *tol* for for $p = 4, ..., 11$, $tol = 10^{-2}, 10^{-3}, \ldots, 10^{-10}$, and for each code and problem combination for which results appear in the tables.

  From these plots we see that 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. There is no significant difference in the **Accepted Time Steps** values for the ST and LE versions of the codes. *However, the BACOLI codes use about ten times as many* **Accepted Time Steps** *as do the BACOLRI codes.*

- In Figures 77-112, we plot **Remeshings** vs. *tol* for $p = 4, ..., 11$, $tol = 10^{-2}, 10^{-3}, \ldots, 10^{-10}$, and for each code and problem combination for which results appear in the tables. We do not include results for the **SCHR** problem since the number of remeshings is very small.

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. There is no significant difference in the number of remeshings when we compare the ST and LE versions of each code. *As well, there is no significant difference in the number of remeshings when we compare the BACOLI codes with the BACOLRI codes. This is significant because the BACOLRI codes take substantially fewer time steps than do the BACOLI codes. For the BACOLI codes, the number of remeshings is relatively small compared to the number of time steps - only around 5%. However for the BACOLRI codes, because there are fewer time steps in total, the number of remeshings is relatively large compared to the total number of time steps - around 20% to 50%.*

- In Figures 113-152, 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 BACOLI/ST, BACOLI/LE, BACOLRI/ST, and BACOLRI/LE. We consider $tol = 10^{-3}, 10^{-4}, \ldots, 10^{-10}$, and p values, 4, 5, 7, and 9. There is one plot for each p value and problem combination for which results appear in the tables.

  From these plots we see that for all problems, the **Calls to CRDCMP** and the **Calls to CRSLVE** 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 number of **Calls to CRDCMP** and **Calls to CRSLVE** is independent of p. There is no significant difference between the ST and LE versions of the codes in terms of the number of **Calls to CRDCMP** and **Calls to CRSLVE**. The BACOLI codes, for coarse tolerances, have somewhat fewer **Calls to CRDCMP** than do the BACOLRI codes but for sharper tolerances the BACOLI codes have more **Calls to CRDCMP** than do the BACOLRI codes. The BACOLI codes have significantly more **Calls to CRSLVE** than do the BACOLRI codes.

- In Figures 153-192, 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 BACOLRI/ST and BACOLRI/LE. The BACOLI/ST and BACOLI/LE codes do not make calls to these routines. We consider $tol = 10^{-3}, 10^{-4}, \ldots, 10^{-10}$ 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 the **Calls to CCRSLV** grow approximately linearly (on a log-log scale) as the tolerances get sharper. And, 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. The **Calls to CCRCMP** value is about one order of magnitude smaller than the **Calls to CCRSLV** value. The number of **Calls to CCRCMP** and **Calls to CCRSLV** is independent of p. The number of **Calls to CCRCMP** and **Calls to CCRSLV** is about 75% of the number of calls to **Calls to CRDCMP** and **Calls to CRSLVE** for given problem, p value, and tolerance. This is the case because only about 75% of the real calls are matched with complex calls; these take place inside RADAU5. About 25% of the calls to CRDCMP and CRSLVE take place outside RADAU5 and in such cases there is no need to also call the complex versions of these routines.

From the results which consider the number of real and complex calls to factorization and backsolve routines, we note firstly that the number of such calls associated with factorizations and backsolves of real matrices is about the same for BACOLI and BACOLRI, on average. However, approximately 75% of the real calls performed by BACOLRI are matched by complex calls. This means that the total number of these calls performed by BACOLRI is approximately 1.75 times the number of calls performed by BACOLI. Furthermore, we have observed from numerical experiments that the complex routine calls are approximately twice as expensive as the calls to the corresponding real routines. This means that the total costs associated with factorizations and backsolves (real and complex) for BACOLRI are approximately 2.5 times the cost of those of BACOLI.

3.3 Machine Dependent Timing Results by Tolerance and by Error

We next provide machine dependent timing results for each code applied to the ten test problems identified earlier in this report (except that BACOLI is not applied to **SCHR**). 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}, 10^{-10}.

The results are provided in Tables 15-24. Instances where a failure occurred correspond to blank table entries; these occurred on **OLBE** with $\epsilon = 10^{-4}$, $tol = 10^{-10}$, for BACOLI/LE and BACOLRI/LE with $p = 4$; **TLBE**, **TLBEx6** and **TLBEx12** with $\epsilon = 10^{-3}$, $tol = 10^{-6}$, for BACOLI/ST and BACOLRI/ST with $p = 4$; **TLBE** with $\epsilon = 10^{-4}$, $tol = 10^{-10}$, for BACOLI/LE and BACOLRI/LE with $p = 4$; **TLBEx12** with $\epsilon = 10^{-4}$, $tol = 10^{-10}$, for BACOLI/LE with $p = 4$; **CSRM**, $tol = 10^{-6}$, for BACOLI/ST with $p = 6$ and $p = 11$; **CSRM**, $tol = 10^{-8}$, for BACOLI/LE with $p = 11$; **CSRM**, $tol = 10^{-10}$, for BACOLI/ST with $p = 9$ and BACOLI/LE with $p = 11$.

From these tables we see that,

- *for coarse tolerances, smaller p values lead to the best efficiency for all codes; however, as the tolerance gets sharper, higher p values correspond to the fastest CPU times.*
- for **OLBE** and **TLBE**, the BACOLRI/ST and BACOLRI/LE are generally comparable to or faster than the BACOLI/ST and BACOLR/LE codes over all p and *tol* values. For **TLBEx6** and **TLBEx12**, BACOLI is faster for coarse and medium tolerances while BACOLRI is faster for the sharpest tolerance. For **CSRM**, BACOLI is comparable to or faster than BACOLRI over all tolerances.
- For small p values, the LE codes are slower than the ST codes but for larger p values their performance becomes roughly comparable to that of the ST codes.

It is often the case that for a given problem and for a given tolerance request, the optimal choice of p is different for different codes. Therefore we next compare tolerance vs. execution time where the choice for p is chosen optimally for each code. That is, we compare the codes for each problem and tolerance combination so that each code uses its best choice of p for that problem and tolerance combination. These results, given in Table 1, were obtained from from Tables 15-24. From this table we see that for most problem/tolerance combinations, the best performance associated with the coarsest tolerance usually corresponds to BACOLI/ST and the best performance for sharper tolerances usually corresponds to BACOLRI/ST. The exceptions to this arise for the **TLBEx12** and the **CSRM** problems, where BACOLI/ST gives the best performance for almost all tolerances.

For a given problem and tolerance, the fastest run over all codes depends on p; see Table 2, where, for each problem, we identify the fastest code and corresponding p value. These results were obtained from Table 1.

From Table 2, we see that for sharper tolerances, larger p values correspond to the fastest times. We also note that over all cases, BACOLRI/ST has the most instances where it is the fastest code. The code that has the second highest number of instances where it is the fastest is BACOLI/ST. The cases where BACOLI/ST is the fastest correspond to (i) coarser tolerances and (ii) larger problems, i.e., **TLBEx12**, which has the largest number of PDEs (12), and for **CSRM** which has four PDEs.

We note that the results considered up to this point in this subsection are tolerance vs. CPU time comparisons; a more relevant comparison is error achieved vs. CPU time. We next consider this type of comparison of the codes for a subset of the test problems, namely, **OLBE**, **TLBE**, and **TLBEx12** with $\epsilon = 10^{-3}$ and 10^{-4}, and **SCHR**. In Tables 25-31, we provide error achieved vs. CPU time for these problems. These results were obtained from the best-fit lines computed for each code and problem combination as shown in Figures 193-248, which we will consider later in this report. These best-fit lines allow us to obtain CPU time estimates for specific errors achieved. We present results for each code/problem combination for $p = 4, \ldots, 11$ and for errors of 10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}.

From these tables we can make observations that are broadly similar to those we made earlier from the tables that consider tolerance vs. CPU time. For larger errors, smaller p values lead to the best efficiency but for smaller errors, larger p values generally correspond to the fastest CPU times. Overall, BACOLRI is faster than BACOLI, except for **TLBEx12**. The LE codes are slower than the ST codes when p is small but gain in relative efficiency for larger p values.

As mentioned above, is often the case that for a given problem and for a given error attained, the optimal choice of p is different for different codes. Therefore we next compare error attained vs. execution time where the choice for p is chosen optimally for each code. That is, we compare the codes for each problem and error attained combination so that each code uses its best choice of p for that problem and error combination. These results are given in Table 3. They were obtained from from Tables 25-31. From this table we see that, for **OLBE** and **TLBE** with $\epsilon = 10^{-3}$, BACOLI is faster for larger errors while BACOLRI is faster for smaller errors. For **OLBE** and **TLBE** with $\epsilon = 10^{-4}$, BACOLRI/ST is the fastest code for all errors. For **TLBEx12**, BACOLI/LE is typically the fastest code over all errors. We also note that smaller p values are more efficient for large errors and that larger p values correspond to faster run times when the error is smaller.

For a given problem and error, the fastest run over all codes depends on p; see Table 4, where, for each problem we identify the fastest code and corresponding p value for a given error attained, based on the Table 3.

From Table 4, we see that for smaller errors, larger p values generally correspond to the fastest times while smaller p values correspond to the fastest times for larger errors. Of the 34 entries in this table, BACOLRI/ST is the fastest code 16 times, while BACOLI/LE is the fastest code 11 times. Most of the instances where BACOLI/LE is the fastest code arise for **TLBEx12**. For the smaller problems, BACOLRI/ST is the fastest code when the error is smaller or the problem is more difficult. BACOLI/LE is the fastest code for the largest problem, **TLBEx12**.

It is important to compare Table 4, in which the execution times are compared by error achieved, with Table 2, in which the execution times are compared by tolerance requested. The most striking difference between the two tables is that while BACOLRI/ST is the fastest code in the majority of entries in both tables, it is BACOLI/LE rather than BACOLI/ST that has the second highest number of appearances when we consider error achieved instead of tolerance requested. This happens because the LOI based error estimates that are employed by BACOLI/LE are known to represent over estimates of the true error. This means that for a given tolerance request, BACOLI/LE does more work that BACOLI/ST. This suggests that a slight scaling of the LOI error estimate to reduce the size of the error estimate is likely appropriate. It is not clear why this effect is less important for BACOLRI.

3.4 Error vs. Execution Time across Codes

Here we present error vs. execution time results for BACOLI/ST, BACOLI/LE, BACOLRI/ST and BACOLRI/LE, for **OLBE**, **TLBE**, and **TLBEx12**, 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 193-248, 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. For **OLBE** and **TLBE**, we see, generally that, for larger errors BACOLI is faster but that for smaller errors, BACOLRI is faster. This is however not the case for **TLBEx12** where we see that BACOLI is the faster code over the entire range of errors.

Some additional comments must be made regarding the performance of BACOLI on the **SCHR** problem. Recall that BACOLI must run with DASSL restricted to orders 1 and 2 in order to be able to even return with a solution. However, even with this restriction imposed, from an inspection of Figures 241-248, we see that, while BACOLI is faster for larger errors, the code is not able to compute a numerical solution for which the corresponding error is less than approximately 10^{-7} even when the requested tolerance is as sharp as 10^{-10}. The data points for the BACOLI codes that appear nearest to top of the figures correspond to runs where the the requested tolerance was 10^{-10} but we see that the achieved error is only about 10^{-7}. (Recall that the data plotted in these figures correspond to requested tolerance values from 10^{-2} to 10^{-10}.) This result is attributable to DASSL since the results for BACOLRI, which are obtained using RADAU5 rather than DASSL, show good correlation between the requested tolerance and the achieved accuracy. For example, an examination of the right hand side of the figures shows that for a tolerance request of 10^{-10}, the BACOLRI codes compute numerical solutions for which the achieved accuracy is a small multiple of 10^{-10}. (This same behavior is evident for all codes when applied to the **OLBE** and **TLBE** problems.)

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

- We first perform a linear fit to the log of the error vs. log of time data associated with BACOLI/ST in order to obtain a continuous representation of the baseline BACOLI/ST data.
- Then, for each (error,time) ordered pair from the BACOLI/LE data set, we use the above mentioned linear fit to the BACOLI/ST data to obtain a corresponding time estimate for BACOLI/ST (i.e., an estimate of how much time BACOLI/ST would take to compute a solution with the same error as BACOLI/LE).

- We then compute the ratio of the actual BACOLI/LE time to this estimated BACOLI/ST time. This yields a set of ordered pairs of the form (error, time ratio) that we can associate with BACOLI/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 249-280, 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 do not include the **SCHR** problem since the results for BACOLI are not relevant since, as mentioned above, the evidence shows that for this problem, BACOLI does not approximately meet the requested tolerance. The behavior of BACOLI/ST is of course represented by the horizontal line at 1.00 in each figure.

We see that, for **OLBE** and **TLBE**, and $p = 4$, compared to BACOLI/ST, BACOLI/LE is better for large errors but worse for small errors. On the other hand, both BACOLRI/ST and BACOLRI/LE are worse than BACOLI/ST for large errors but better than BACOLI/ST for small errors. For **TLBEx12**, BACOLI/LE is comparable to BACOLI/ST but BACOLRI/ST and BACOLRI/LE are much worse for large errors and at best comparable for small errors.

3.5 Error vs. Execution Time across p Values

In this subsection, we consider error vs. execution time results for BACOLI/ST, BACOLI/LE, BACOLRI/ST, BACOLRI/LE, over a range of p values, $4, \ldots, 11$, for **OLBE**, **TLBE**, **TLBEx12** with $\epsilon = 10^{-3}$ and 10^{-4}, and for **SCHR**. For the **SCHR** problem, we provide results only for the BACOLRI codes. *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 297-322, 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, **OLBE**, **TLBE**, for low accuracy requirements, all codes are generally more efficient when p is small, while for higher accuracy requirements, 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. Even for a tolerance of 10^{-10}, $p = 7$ or 8 provides performance comparable to that exhibited from the use higher p values. This suggests that a reasonable working range of p values should the range from 4 to 8. For **TLBEx12**, the above observations are correct for $p = 4$ but for higher p values there tends to be less relative improvement as the errors get smaller; that is, even for small errors, the smaller p values deliver better performance. This is also the case for the **SCHR** problem.

4 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 spatial error estimation schemes, standard (ST) spatial error control as well as an alternative spatial error control 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 presents a detailed examination of the performance of the two newest codes, BACOLI and BACOLRI. The results (see in particular Table 4 where we compare error achieved vs. CPU time) show that for the test problems that have a small number of PDEs, and where the difficult of the problem is greater (due to a smaller ϵ value or a sharper tolerance), BACOLRI with ST spatial error control gives the best performance overall; for the easier versions of these test problems, BACOLI/LE or BACOLRI/LE have the better or comparable performance. For the larger problems, **TLBEx12**, BACOLI/LE is the generally the fastest code. For the **SCHR** problem, BACOLRI/ST is faster than BACOLRI/LE. BACOLI/ST is the fastest only for the simpler version of **OLBE** with the largest error.

It should be noted that when one compares tolerance requested vs. CPU time, BACOLI/ST appears to offer much better performance. This is due to the fact that the LE codes employ an overestimate of the error and thus for a given tolerance, they have to work harder than to the ST codes, typically delivering more accuracy than requested. This suggests that a modification of the LE error control is warranted to better match it to the actual error of the numerical solution.

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. For larger PDE systems, the advantage associated with larger p values is less apparent.

A final observation from this study is that for problems, such as the **SCHR** problem, that lead to stability issues for the BDFs implemented in DASSL, BACOLI, which is based on DASSL, cannot be used. BACOLI will not terminate unless DASSL is restricted to use BDFs of orders one and two, and even when this restriction is applied, BACOLI cannot compute numerical solutions for which the corresponding error is a small multiple of the requested tolerance. This is due to DASSL being unable to adequately control the time error of the B-spline coefficients it is asked to compute.

There are several directions for future work. The results of this report suggest that it may be worthwhile to modify the BACOLI and BACOLRI codes in order to have them choose p based on the tolerance requested. Also, as mentioned above, a modification of the codes to improve the LE spatial error estimate appears to be worthwhile.

The results from this work may also be useful in the development of improved methods for Boundary Value ODEs and 2D PDEs.

References

[1] Mark Adams, Connor Tannahill, and Paul Muir. Error control Gaussian collocation software for boundary value ODEs and 1D time-dependent PDEs. *Numer. Algor.*, 2019.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

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

[8] 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.

[9] 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.

[10] 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.

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

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

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

[14] 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.

[15] 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.

[16] J. Pew, T. Murtha, C. Tannahill, and P.H. Muir. Error control B-spline Gaussian collocation/Runge-Kutta PDE software with interpolation-based spatial error estimation. *Saint Mary's University, Dept. of Mathematics and Computing Science Technical Report Series, Technical Report 2018_002, http://cs.smu.ca/tech_reports*, 2018.

[17] 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.

[18] Jack Pew, Zhi Li, Connor Tannahill, Paul Muir, and Graeme Fairweather. Performance analysis of error-control B-spline Gaussian collocation software for PDEs. *Comput. Math. Appl.*, 77(7):1888–1901, 2019.

[19] 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.

[20] 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.

[21] 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.

[22] R. Wang, P. Keast, and P.H. Muir. A high-order global spatially adaptive collocation method for 1-D parabolic PDEs. *Appl. Numer. Math.*, 50(2):239–260, 2004.

[23] 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.

Problem	Code	$tol = 10^{-4}$	$tol = 10^{-6}$	$tol = 10^{-8}$	$tol = 10^{-10}$
OLBE	LI/ST	**4, 0.02**	4, 0.09	6, 0.27	8, 0.76
$\epsilon = 10^{-3}$	LI/LE	4,5, 0.03	5, 0.09	8, 0.30	8, 0.76
	RI/ST	4, 0.03	**4-6, 0.07**	**6, 0.18**	**8, 0.47**
	RI/LE	4, 0.04	5, 0.08	7, 0.21	9, 0.51
OLBE	LI/ST	4, 0.28	4, 0.91	6, 2.76	7, 7.49
$\epsilon = 10^{-4}$	LI/LE	4, 0.32	5, 1.04	7, 3.07	7, 8.77
	RI/ST	**4, 0.23**	**4, 0.53**	**6, 1.69**	**7,8, 4.72**
	RI/LE	4, 0.31	5, 0.71	7, 2.07	8, 5.38
TLBE	LI/ST	**4, 0.02**	5,6, 0.08	6, 0.23	9, 0.69
$\epsilon = 10^{-3}$	LI/LE	4-6, 0.03	5, 0.08	7, 0.26	8, 0.77
	RI/ST	4,5, 0.03	**5, 0.06**	**6, 0.15**	**8, 0.40**
	RI/LE	4,5, 0.03	6,7, 0.07	7-9, 0.18	9-11, 0.46
TLBE	LI/ST	4, 0.27	4, 0.89	6, 2.69	8, 7.64
$\epsilon = 10^{-4}$	LI/LE	4, 0.30	5, 0.92	7, 3.01	8, 8.42
	RI/ST	**4, 0.22**	**4, 0.51**	**5,6, 1.57**	**7, 4.28**
	RI/LE	4, 0.27	5, 0.67	7, 1.88	8, 5.05
TLBEx6	LI/ST	**4, 0.11**	**5, 0.41**	6, 1.54	6, 5.19
$\epsilon = 10^{-3}$	LI/LE	4, 0.12	5, 0.42	6, 1.63	8, 5.16
	RI/ST	4,5, 0.34	5, 0.59	**5,6, 1.47**	**7, 4.50**
	RI/LE	4, 0.32	6, 0.76	7, 2.00	9, 5.79
TLBEx6	LI/ST	**4, 1.26**	4, 4.55	5, 17.38	7, 57.39
$\epsilon = 10^{-4}$	LI/LE	4, 1.47	5, 4.89	6, 18.89	7, 59.15
	RI/ST	4, 2.01	**4, 4.27**	**4, 12.72**	**6, 40.16**
	RI/LE	4, 2.49	5, 6.17	6, 17.61	8, 56.29
TLBEx12	LI/ST	**5, 0.39**	**5, 1.32**	6, 5.69	6, 20.52
$\epsilon = 10^{-3}$	LI/LE	4, 0.43	5, 1.48	**6, 5.42**	**8, 20.12**
	RI/ST	4, 1.37	5, 2.66	5, 7.26	7, 25.71
	RI/LE	4, 1.47	6, 3.73	7, 10.65	8, 34.83
TLBEx12	LI/ST	**4, 3.80**	**4, 13.42**	**4, 55.00**	6, 238.42
$\epsilon = 10^{-4}$	LI/LE	4, 4.87	5, 15.70	5, 64.20	7, 244.37
	RI/ST	4, 9.79	4, 18.54	5, 63.64	**6, 221.16**
	RI/LE	4, 12.17	5, 31.11	6, 95.75	7, 326.85
CSRM	LI/ST	**4,5, 0.01**	**4,5, 0.05**	**5, 0.19**	**7, 0.68**
	LI/LE	4-6, 0.02	6,7, 0.08	6, 0.25	7, 0.69
	RI/ST	4,5, 0.08	4,5, 0.13	5, 0.27	6, 0.79
	RI/LE	4-6, 0.10	6, 0.17	6, 0.41	8, 1.12
SCHR	RI/ST	**4, 0.13**	**5,6, 0.15**	**5,6, 0.19**	**7,10, 0.32**
	RI/LE	**4,5, 0.13**	**5,6, 0.15**	6,7, 0.22	10,11, 0.35

Table 1: *Each table entry gives p and CPU time corresponding to the fastest run, for each code, for each problem and* **tolerance requested**. *BACOLI/ST ≡ LI/ST, BACOLI/LE ≡ LI/LE, BACOLRI/ST ≡ RI/ST, BACOLRI/LE ≡ RI/LE. Best time and corresponding p value shown in bold.*

Problem	$tol = 10^{-4}$	$tol = 10^{-6}$	$tol = 10^{-8}$	$tol = 10^{-10}$
OLBE, $\epsilon = 10^{-3}$	**LI/ST**, 4	**RI/ST**, 4-6	**RI/ST**, 6	**RI/ST**, 8
OLBE, $\epsilon = 10^{-4}$	**RI/ST**, 4	**RI/ST**, 4	**RI/ST**, 6	**RI/ST**, 7,8
TLBE, $\epsilon = 10^{-3}$	**LI/ST**, 4	**RI/ST**, 5	**RI/ST**, 6	**RI/ST**, 8
TLBE, $\epsilon = 10^{-4}$	**RI/ST**, 4	**RI/ST**, 4	**RI/ST**, 5,6	**RI/ST**, 7
TLBEx6, $\epsilon = 10^{-3}$	**LI/ST**, 4	**LI/ST**, 5	**RI/ST**, 5,6	**RI/ST**, 7
TLBEx6, $\epsilon = 10^{-4}$	**LI/ST**, 4	**RI/ST**, 4	**RI/ST**, 4	**RI/ST**, 6
TLBEx12, $\epsilon = 10^{-3}$	**LI/ST**, 4	**LI/ST**, 5	**LI/LE**, 6	**LI/LE**, 8
TLBEx12, $\epsilon = 10^{-4}$	**LI/ST**, 4	**LI/ST**, 4	**LI/ST**, 4	**RI/ST**, 6
CSRM	**LI/ST**, 4	**LI/ST**, 4,5	**LI/ST**, 5	**LI/ST**, 7
SCHR	**RI/ST**, 4	**RI/ST**, 5,6	**RI/ST**, 5,6	**RI/ST**, 7,10
	RI/LE, 4,5	**RI/LE**, 5,6		

Table 2: *Code and value of p that corresponds to the fastest run, for a given problem and tolerance. BACOLI/ST* ≡ **LI/ST**, *BACOLI/LE* ≡ **LI/LE**, *BACOLRI/ST* ≡ **RI/ST**, *BACOLRI/LE* ≡ **RI/LE**.

Problem	Code	$error = 10^{-4}$	$error = 10^{-6}$	$error = 10^{-8}$	$error = 10^{-10}$
OLBE	LI/ST	**4-6, 0.02**	5-7, 0.08	6,7, 0.24	8, 0.65
$\epsilon = 10^{-3}$	LI/LE	**4-6, 0.02**	**5, 0.06**	7, 0.20	7, 0.57
	RI/ST	4-7, 0.03	4-7, 0.07	**6, 0.15**	**8, 0.32**
	RI/LE	**5, 0.02**	5-8, 0.07	7-9, 0.16	**9-10, 0.32**
OLBE	LI/ST	4,6, 0.44	6, 1.08	6, 2.65	8, 6.37
$\epsilon = 10^{-4}$	LI/LE	4, 0.35	6, 1.12	6, 3.03	6, 8.20
	RI/ST	**4, 0.31**	**6, 0.76**	**6, 1.54**	**6, 3.15**
	RI/LE	5, 0.34	6, 0.87	8, 1.96	11, 3.77
TLBE	LI/ST	4-7, 0.02	5-7, 0.07	6, 0.21	10, 0.62
$\epsilon = 10^{-3}$	LI/LE	**4-5, 0.01**	**5-6, 0.05**	6-7, 0.17	7, 0.51
	RI/ST	5, 0.02	5-8, 0.06	**6-8, 0.13**	8, 0.28
	RI/LE	4-7, 0.02	**6-7, 0.05**	**7-8, 0.13**	**9, 0.27**
TLBE	LI/ST	4-5, 0.37	6, 1.04	6, 2.63	7, 6.57
$\epsilon = 10^{-4}$	LI/LE	4, 0.29	6, 1.05	6, 2.89	8, 7.21
	RI/ST	**4, 0.26**	**5, 0.67**	**6, 1.55**	**7, 3.13**
	RI/LE	5, 0.30	6, 0.78	6, 1.80	8, 3.61
TLBEx12	LI/ST	5, 0.30	5, 1.15	5, 4.37	6, 15.28
$\epsilon = 10^{-3}$	LI/LE	**4, 0.20**	**5, 0.89**	**6, 3.22**	**7, 10.27**
	RI/ST	5, 1.20	5, 2.68	6, 5.94	6, 11.73
	RI/LE	4, 0.82	5, 2.86	7, 7.12	8, 14.50
TLBEx12	LI/ST	4, 5.79	5, 18.93	5, 57.01	6, 165.00
$\epsilon = 10^{-4}$	LI/LE	**4, 4.71**	**5, 18.52**	**6, 51.68**	6, 140.68
	RI/ST	4, 9.99	4, 23.06	4, 53.24	**6, 112.26**
	RI/LE	4, 12.26	5, 38.43	6, 79.74	6, 154.54
SCHR	RI/ST	**5, 0.05**	**5, 0.10**	**6, 0.17**	**7, 0.28**
	RI/LE	**4, 0.05**	6, 0.12	7, 0.22	8, 0.38

Table 3: *Each table entry gives p and CPU time corresponding to the fastest run, for each code, for each problem and* **error achieved**. *BACOLI/ST ≡ LI/ST, BACOLI/LE ≡ LI/LE, BACOLRI/ST ≡ RI/ST, BACOLRI/LE ≡ RI/LE*

Problem	$error = 10^{-4}$	$error = 10^{-6}$	$error = 10^{-8}$	$error = 10^{-10}$
OLBE, $\epsilon = 10^{-3}$	**LI/ST**, 4-6	**LI/LE**, 5	**RI/ST**, 6	**RI/ST**, 8
	LI/LE, 4-6			**RI/LE**, 9,10
	RI/LE, 5			
OLBE, $\epsilon = 10^{-4}$	**RI/ST**, 4	**RI/ST**, 6	**RI/ST**, 6	**RI/ST**, 6
TLBE, $\epsilon = 10^{-3}$	**LI/LE**, 4,5	**LI/LE**, 5,6	**RI/ST**, 6-8	**RI/LE**, 9
		RI/LE, 6,7	**RI/LE**, 7,8	
TLBE, $\epsilon = 10^{-4}$	**RI/ST**, 4	**RI/ST**, 5	**RI/ST**, 6	**RI/ST**, 7
TLBEx12, $\epsilon = 10^{-3}$	**LI/LE**, 4	**LI/LE**, 5	**LI/LE**, 6	**LI/LE**, 7
TLBEx12, $\epsilon = 10^{-4}$	**LI/LE**, 4	**LI/LE**, 5	**LI/LE**, 6	**RI/ST**, 6
SCHR	**RI/ST**, 5	**RI/ST**, 5	**RI/ST**, 6	**RI/ST**, 7
	RI/LE, 4			

Table 4: *Code and value of p that corresponds to the fastest run, for a given problem and error. BACOLI/ST* ≡ **LI/ST**, *BACOLI/LE* ≡ **LI/LE**, *BACOLRI/ST* ≡ **RI/ST**, *BACOLRI/LE* ≡ **RI/LE**.

tol	10^{-4}	10^{-6}	10^{-8}
p	4		
BACOLI/ST	14, 1285, 71 [187, 2151]	24, 3015, 103 [402, 4543]	54, 6970, 135 [1123, 10540]
BACOLI/LE	17, 1102, 131 [280, 2217]	45, 2178, 154 [336, 3511]	124, 5715, 468 [962, 9905]
BACOLRI/ST	15, 264, 78 [383, 1623] {304, 1194}	25, 609, 148 [483, 3174] {334, 2268}	53, 1349, 193 [406, 6614] {212, 4878}
BACOLRI/LE	20, 261, 118 [359, 1804] {240, 1306}	53, 602, 134 [369, 3158] {234, 2287}	130, 1325, 414 [845, 6945] {430, 4791}
p	5		
BACOLI/ST	13, 1196, 66 [163, 1981]	20, 3046, 121 [491, 4992]	35, 6564, 113 [1050, 9701]
BACOLI/LE	15, 1153, 91 [202, 2020]	25, 2463, 123 [282, 3799]	57, 4885, 142 [346, 6478]
BACOLRI/ST	15, 254, 62 [357, 1520] {294, 1140}	18, 600, 110 [419, 3051] {308, 2230}	34, 1333, 122 [259, 6365] {136, 4787}
BACOLRI/LE	15, 255, 80 [366, 1621] {285, 1205}	26, 600, 119 [349, 3014] {229, 2175}	50, 1339, 145 [324, 6250] {178, 4620}
p	7		
BACOLI/ST	12, 1283, 51 [128, 2031]	15, 3055, 66 [440, 4679]	21, 6266, 124 [1111, 9673]
BACOLI/LE	11, 1253, 66 [153, 2042]	14, 2981, 103 [356, 4637]	22, 6637, 164 [685, 9717]
BACOLRI/ST	11, 301, 45 [365, 1675] {319, 1269}	15, 586, 60 [354, 2829] {293, 2119}	18, 1334, 145 [504, 6403] {358, 4778}
BACOLRI/LE	13, 252, 66 [339, 1560] {272, 1175}	15, 588, 95 [376, 2975] {280, 2196}	22, 1324, 143 [309, 6296] {165, 4685}
p	9		
BACOLI/ST	10, 1715, 42 [218, 2740]	12, 3562, 67 [521, 5451]	15, 7200, 118 [1189, 10903]
BACOLI/LE	10, 1366, 56 [149, 2213]	15, 2801, 69 [200, 3994]	15, 6600, 125 [1075, 10032]
BACOLRI/ST	10, 399, 36 [467, 2049] {430, 1538}	11, 613, 71 [473, 3154] {401, 2343}	15, 1302, 94 [590, 6193] {495, 4691}
BACOLRI/LE	11, 253, 56 [348, 1537] {291, 1171}	15, 578, 68 [331, 2830] {262, 2115}	14, 1342, 124 [357, 6409] {232, 4818}

Table 5: *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		
BACOLI/ST	16, 13127, 741 [2370, 23725]	27, 30970, 861 [5616, 51026]	47, 66987, 939 [9475, 97420]
BACOLI/LE	20, 10870, 1213 [2550, 21539]	40, 24401, 1409 [3000, 38617]	99, 47919, 3652 [7433, 81388]
BACOLRI/ST	18, 2610, 730 [3589, 16219] {2858, 11817}	25, 5963, 842 [3816, 29299] {2973, 21546}	48, 13750, 1018 [3073, 64313] {2054, 48526}
BACOLRI/LE	22, 2504, 1133 [4068, 17797] {2934, 13026}	40, 5885, 1325 [3554, 31379] {2228, 22843}	115, 13151, 3621 [7253, 67086] {3631, 46692}
p	5		
BACOLI/ST	14, 13819, 655 [2444, 24438]	20, 35856, 1011 [6175, 54453]	31, 72055, 1067 [9926, 106168]
BACOLI/LE	15, 18120, 997 [2384, 23825]	25, 29474, 1173 [2854, 45325]	49, 64649, 1214 [3954, 91821]
BACOLRI/ST	15, 2810, 627 [3493, 16606] {2865, 12197}	20, 5850, 933 [4412, 30144] {3478, 22313}	31, 13246, 1043 [4814, 63447] {3770, 48114}
BACOLRI/LE	13, 2579, 951 [4198, 17578] {3246, 12989}	26, 5813, 1094 [3396, 30018] {2301, 22016}	47, 13338, 1234 [2492, 62925] {1257, 47118}
p	7		
BACOLI/ST	15, 14141, 429 [1396, 22721]	15, 36086, 971 [7176, 59778]	20, 92859, 1188 [11506, 102134]
BACOLI/LE	15, 12780, 637 [1781, 21625]	15, 72997, 1247 [6745, 59040]	22, 62140, 1368 [9047, 93262]
BACOLRI/ST	14, 3367, 359 [3770, 17577] {3410, 13216}	16, 5876, 776 [4218, 30214] {3441, 22228}	20, 12809, 1318 [7628, 63290] {6309, 47686}
BACOLRI/LE	15, 2620, 577 [3617, 16006] {3039, 12194}	15, 5821, 1260 [5024, 31798] {3763, 23282}	23, 12994, 1320 [2937, 62802] {1616, 47167}
p	9		
BACOLI/ST	14, 15041, 386 [1434, 23981]	15, 35909, 534 [5057, 54055]	17, 72175, 1312 [12887, 110479]
BACOLI/LE	14, 12835, 548 [1436, 21268]	15, 32975, 631 [4558, 50830]	15, 120274, 1732 [15849, 121737]
BACOLRI/ST	13, 4927, 281 [5140, 22749] {4858, 16701}	15, 6241, 463 [4710, 30922] {4246, 23289}	15, 12870, 1398 [6391, 64297] {4992, 47719}
BACOLRI/LE	12, 3839, 437 [4623, 20273] {4185, 15121}	15, 5771, 621 [3921, 29186] {3299, 21850}	16, 12824, 1667 [7497, 64164] {5829, 47913}

Table 6: *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	
BACOLI/ST	15, 926, 85 [212, 1685]	— —	50, 4946, 328 [1277, 7908]
BACOLI/LE	18, 792, 88 [189, 1559]	44, 1705, 124 [267, 2772]	135, 3384, 407 [838, 7026]
BACOLRI/ST	15, 199, 88 [301, 1281] {212, 905}	— — —	51, 969, 142 [390, 4570] {247, 3316}
BACOLRI/LE	18, 197, 76 [239, 1233] {162, 883}	45, 446, 115 [320, 2416] {204, 1739}	134, 978, 366 [862, 5216] {495, 3505}
p		5	
BACOLI/ST	15, 868, 55 [128, 1443]	17, 2240, 72 [255, 3366]	34, 4777, 90 [685, 6977]
BACOLI/LE	13, 883, 75 [165, 1585]	23, 1792, 96 [216, 2735]	51, 4121, 104 [263, 5432]
BACOLRI/ST	15, 201, 53 [256, 1145] {202, 837}	18, 444, 73 [263, 2171] {189, 1580}	30, 964, 98 [282, 4328] {183, 3167}
BACOLRI/LE	14, 197, 64 [240, 1185] {175, 859}	25, 444, 87 [279, 2261] {191, 1642}	55, 972, 101 [315, 4426] {213, 3251}
p		7	
BACOLI/ST	13, 940, 38 [92, 1477]	15, 2330, 68 [332, 3644]	21, 4667, 102 [780, 7164]
BACOLI/LE	15, 887, 52 [118, 1441]	15, 2260, 82 [222, 3380]	21, 4690, 105 [378, 6683]
BACOLRI/ST	14, 211, 34 [230, 1118] {195, 837}	14, 453, 68 [278, 2223] {209, 1629}	18, 976, 103 [290, 4428] {186, 3245}
BACOLRI/LE	15, 200, 46 [216, 1122] {169, 829}	15, 448, 72 [237, 2163] {164, 1570}	18, 963, 108 [302, 4415] {193, 3235}
p		9	
BACOLI/ST	11, 1124, 37 [134, 1800]	15, 2553, 52 [284, 3685]	15, 5293, 78 [780, 7800]
BACOLI/LE	12, 964, 45 [105, 1570]	15, 2181, 65 [230, 3228]	14, 4924, 115 [775, 7558]
BACOLRI/ST	15, 201, 34 [218, 1103] {183, 833}	14, 461, 50 [309, 2220] {258, 1657}	15, 983, 81 [334, 4498] {252, 3349}
BACOLRI/LE	9, 265, 41 [326, 1400] {284, 1035}	15, 448, 60 [250, 2129] {189, 1559}	15, 973, 93 [267, 4451] {173, 3291}

Table 7: *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	
BACOLI/ST	15, 10962, 737 [1895, 17637]	27, 22623, 791 [3338, 36524]	47, 47809, 857 [6807, 70543]
BACOLI/LE	19, 7978, 969 [1984, 16370]	42, 16482, 1203 [2475, 27034]	100, 38827, 3274 [6605, 68291]
BACOLRI/ST	15, 1965, 635 [2560, 11803] {1924, 8486}	26, 4486, 728 [2481, 21657] {1752, 15704}	51, 9978, 860 [1818, 44399] {957, 32700}
BACOLRI/LE	20, 1894, 863 [2600, 12675] {1736, 9054}	38, 4379, 1099 [2711, 23618] {1611, 17040}	102, 9691, 2768 [5676, 48505] {2907, 33277}
p		5	
BACOLI/ST	15, 9678, 489 [1349, 16543]	19, 23087, 746 [3704, 37536]	33, 48713, 760 [6523, 70496]
BACOLI/LE	15, 9090, 775 [1714, 17141]	24, 20179, 957 [2106, 31047]	43, 46580, 1090 [3104, 65542]
BACOLRI/ST	15, 2122, 480 [2430, 11650] {1949, 8470}	19, 4386, 704 [2681, 21557] {1976, 15742}	30, 9749, 824 [2414, 43624] {1589, 32226}
BACOLRI/LE	15, 1961, 713 [2807, 12305] {2093, 8885}	21, 4340, 878 [2338, 22079] {1459, 15982}	45, 9608, 1008 [2114, 43738] {1105, 32113}
p		7	
BACOLI/ST	14, 10938, 402 [1250, 17857]	15, 26671, 741 [4786, 43232]	17, 89230, 908 [8397, 78062]
BACOLI/LE	15, 9792, 547 [1320, 16743]	15, 75411, 985 [4205, 41097]	21, 83805, 1099 [5500, 70180]
BACOLRI/ST	15, 2386, 419 [2798, 12700] {2378, 9374}	15, 4459, 650 [3093, 22473] {2442, 16532}	19, 9580, 944 [4057, 44093] {3112, 32595}
BACOLRI/LE	14, 2306, 515 [2845, 12820] {2329, 9397}	14, 4396, 955 [3366, 23154] {2410, 16790}	22, 9551, 1042 [2164, 43881] {1121, 32245}
p		9	
BACOLI/ST	14, 12513, 362 [1635, 20162]	15, 28289, 590 [4413, 43590]	14, 55178, 926 [9393, 83551]
BACOLI/LE	14, 12062, 424 [1652, 19852]	14, 27605, 755 [4638, 43702]	15, 64341, 1216 [10240, 87680]
BACOLRI/ST	14, 3185, 355 [3612, 15636] {3256, 11449}	14, 4939, 633 [4224, 25049] {3590, 18521}	15, 9545, 966 [4415, 44962] {3448, 33189}
BACOLRI/LE	15, 2683, 441 [3250, 14073] {2808, 10302}	15, 4525, 777 [3787, 23620] {3009, 17334}	15, 9605, 1219 [4174, 45021] {2954, 32955}

Table 8: *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	
BACOLI/ST	14, 882, 80 [197, 1629]	— —	51, 5062, 224 [927, 7630]
BACOLI/LE	17, 756, 93 [199, 1540]	44, 1544, 117 [254, 2550]	136, 3384, 420 [864, 7142]
BACOLRI/ST	15, 200, 124 [375, 1432] {250, 983}	— — —	50, 970, 215 [525, 4921] {309, 3520}
BACOLRI/LE	18, 196, 79 [246, 1242] {166, 887}	42, 446, 114 [321, 2412] {206, 1737}	131, 978, 351 [856, 5151] {504, 3470}
p		5	
BACOLI/ST	15, 870, 54 [129, 1445]	19, 2122, 73 [258, 3235]	35, 4621, 82 [625, 6621]
BACOLI/LE	15, 854, 74 [166, 1537]	24, 1790, 92 [209, 2679]	50, 4161, 115 [277, 5503]
BACOLRI/ST	13, 208, 46 [252, 1131] {205, 830}	18, 444, 78 [268, 2192] {189, 1591}	32, 963, 88 [253, 4335] {164, 3195}
BACOLRI/LE	14, 198, 68 [249, 1213] {180, 878}	24, 444, 92 [295, 2282] {202, 1653}	54, 972, 94 [286, 4403] {191, 3242}
p		7	
BACOLI/ST	13, 945, 36 [85, 1433]	15, 2259, 63 [307, 3504]	18, 4792, 104 [810, 7432]
BACOLI/LE	15, 884, 54 [124, 1447]	15, 2235, 77 [215, 3331]	22, 4816, 120 [409, 6861]
BACOLRI/ST	14, 208, 35 [232, 1109] {196, 830}	15, 450, 66 [271, 2183] {204, 1599}	18, 979, 119 [342, 4498] {222, 3280}
BACOLRI/LE	15, 199, 48 [219, 1128] {170, 832}	15, 446, 75 [251, 2183] {175, 1586}	21, 964, 117 [325, 4415] {207, 3216}
p		9	
BACOLI/ST	11, 1141, 33 [113, 1779]	15, 2408, 54 [302, 3623]	15, 5297, 84 [802, 7968]
BACOLI/LE	11, 1012, 44 [120, 1650]	15, 2225, 64 [227, 3277]	15, 5030, 109 [660, 7452]
BACOLRI/ST	15, 199, 34 [221, 1088] {186, 820}	14, 462, 50 [287, 2208] {236, 1644}	15, 982, 83 [373, 4522] {289, 3373}
BACOLRI/LE	11, 241, 39 [272, 1268] {232, 942}	15, 450, 59 [249, 2138] {189, 1569}	15, 971, 97 [267, 4478] {169, 3312}

Table 9: *Machine independent results for the Two Layer Burgers equation*$\times 6$ *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	
BACOLI/ST	14, 9319, 660 [1650, 17261]	26, 22537, 786 [3466, 36452]	50, 48081, 833 [6482, 70085]
BACOLI/LE	21, 8257, 940 [1928, 16449]	41, 16179, 1214 [2495, 26615]	108, 36176, 3170 [6366, 64669]
BACOLRI/ST	17, 1964, 617 [2503, 11794] {1885, 8499}	25, 4464, 729 [2509, 21761] {1779, 15821}	50, 9920, 874 [1927, 44107] {1052, 32438}
BACOLRI/LE	20, 1897, 868 [2605, 12704] {1736, 9070}	36, 4385, 1137 [2744, 23828] {1606, 17168}	101, 9685, 2836 [5801, 48714] {2964, 33356}
p		5	
BACOLI/ST	14, 9743, 510 [1406, 16803]	20, 22933, 749 [3611, 37399]	30, 48644, 787 [6776, 71101]
BACOLI/LE	15, 9132, 754 [1719, 16957]	24, 19917, 932 [2097, 30618]	48, 43528, 1016 [2792, 60358]
BACOLRI/ST	14, 2112, 479 [2394, 11653] {1914, 8473}	20, 4371, 679 [2643, 21366] {1963, 15603}	32, 9747, 858 [2510, 43714] {1651, 32250}
BACOLRI/LE	14, 1960, 707 [2784, 12269] {2076, 8867}	22, 4335, 857 [2309, 21946] {1451, 15896}	46, 9626, 1014 [2120, 43683] {1105, 32028}
p		7	
BACOLI/ST	15, 12749, 439 [1220, 17230]	15, 37942, 795 [4984, 43924]	20, 132940, 914 [8522, 78680]
BACOLI/LE	15, 10305, 537 [1435, 17606]	15, 54848, 973 [4234, 41459]	21, 46917, 1037 [5384, 68288]
BACOLRI/ST	14, 2625, 407 [2973, 13432] {2565, 9810}	15, 4477, 684 [3127, 22728] {2442, 16681}	20, 9610, 986 [4162, 44383] {3175, 32771}
BACOLRI/LE	14, 2383, 533 [2993, 13179] {2459, 9607}	13, 4401, 929 [3248, 22993] {2318, 16674}	22, 9496, 1060 [2217, 43875] {1156, 32258}
p		9	
BACOLI/ST	15, 11478, 384 [1527, 18777]	15, 28576, 583 [4451, 43971]	15, 96843, 954 [9518, 83362]
BACOLI/LE	14, 11704, 422 [1540, 19211]	14, 29830, 739 [4797, 44656]	15, 75917, 1195 [10156, 88050]
BACOLRI/ST	15, 2942, 367 [3442, 14855] {3074, 10920}	15, 4788, 634 [3972, 24343] {3337, 18021}	15, 9535, 943 [4502, 44873] {3558, 33166}
BACOLRI/LE	14, 2922, 446 [3577, 15157] {3130, 11035}	14, 4549, 806 [3903, 23904] {3096, 17476}	14, 9604, 1254 [4262, 45197] {3007, 33069}

Table 10: *Machine independent results for the Two Layer Burgers equation*$\times 6$ *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	
BACOLI/ST	13, 940, 102 [260, 1781]	— —	52, 5014, 741 [2527, 8964]
BACOLI/LE	19, 760, 95 [204, 1580]	44, 1819, 118 [252, 2823]	136, 3384, 418 [860, 7125]
BACOLRI/ST	15, 201, 90 [309, 1301] {218, 918}	— — —	53, 970, 141 [379, 4596] {237, 3343}
BACOLRI/LE	19, 197, 84 [252, 1267] {167, 901}	45, 446, 106 [308, 2365] {201, 1706}	131, 978, 353 [848, 5165] {494, 3480}
p		5	
BACOLI/ST	14, 901, 51 [125, 1480]	18, 2193, 75 [258, 3321]	31, 4869, 90 [695, 7124]
BACOLI/LE	15, 844, 72 [157, 1504]	25, 1856, 104 [233, 2876]	53, 4194, 106 [252, 5368]
BACOLRI/ST	13, 206, 51 [253, 1160] {201, 849}	18, 444, 69 [248, 2140] {178, 1557}	31, 963, 84 [258, 4341] {173, 3209}
BACOLRI/LE	15, 197, 62 [236, 1183] {173, 861}	21, 445, 88 [287, 2267] {198, 1645}	53, 972, 105 [302, 4439] {196, 3256}
p		7	
BACOLI/ST	13, 926, 38 [92, 1459]	14, 2439, 70 [360, 3780]	19, 4699, 106 [799, 7253]
BACOLI/LE	15, 825, 51 [117, 1372]	15, 2207, 79 [212, 3269]	21, 4913, 114 [420, 7007]
BACOLRI/ST	14, 208, 35 [230, 1112] {194, 832}	14, 453, 68 [279, 2213] {210, 1622}	17, 971, 103 [319, 4417] {215, 3239}
BACOLRI/LE	15, 199, 50 [218, 1135] {167, 835}	13, 452, 86 [285, 2280] {198, 1655}	22, 963, 108 [301, 4396] {192, 3216}
p		9	
BACOLI/ST	11, 1123, 35 [114, 1783]	15, 2330, 51 [234, 3427]	15, 5332, 76 [774, 7818]
BACOLI/LE	12, 1000, 46 [128, 1653]	15, 2163, 63 [226, 3239]	14, 4981, 100 [687, 7406]
BACOLRI/ST	15, 198, 31 [220, 1078] {188, 817}	14, 462, 47 [296, 2212] {248, 1654}	15, 979, 85 [366, 4511] {280, 3359}
BACOLRI/LE	12, 235, 41 [264, 1266] {222, 936}	15, 447, 59 [245, 2128] {185, 1561}	14, 971, 108 [293, 4493] {184, 3305}

Table 11: *Machine independent results for the Two Layer Burgers equation*×12 *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	
BACOLI/ST	15, 9306, 663 [1682, 17278]	24, 25868, 756 [3306, 36352]	45, 47558, 848 [6737, 69823]
BACOLI/LE	20, 8195, 980 [2011, 16689]	42, 16236, 1215 [2486, 26804]	100, 38716, 3250 [6533, 67984]
BACOLRI/ST	17, 1966, 638 [2537, 11859] {1898, 8524}	25, 4481, 695 [2445, 21559] {1749, 15673}	46, 9919, 1163 [2473, 45480] {1309, 33234}
BACOLRI/LE	20, 1897, 893 [2679, 12799] {1785, 9115}	40, 4393, 1105 [2697, 23680] {1591, 17076}	102, 9682, 2827 [5806, 48640] {2978, 33303}
p		5	
BACOLI/ST	15, 9663, 473 [1270, 16323]	20, 22965, 780 [3684, 37418]	33, 49047, 807 [6838, 71577]
BACOLI/LE	13, 9714, 804 [1838, 17430]	23, 19986, 946 [2139, 30812]	46, 45034, 1043 [2991, 63229]
BACOLRI/ST	15, 2113, 476 [2429, 11577] {1952, 8418}	18, 4388, 753 [2809, 21800] {2055, 15879}	33, 9727, 773 [2276, 43478] {1502, 32204}
BACOLRI/LE	15, 1965, 761 [2884, 12577] {2122, 9060}	24, 4329, 894 [2378, 22074] {1483, 15956}	46, 9620, 986 [2079, 43855] {1092, 32262}
p		7	
BACOLI/ST	15, 10282, 440 [1180, 16983]	14, 26729, 736 [4740, 43158]	21, 96947, 911 [8469, 78484]
BACOLI/LE	13, 10469, 554 [1364, 17279]	14, 24283, 973 [4278, 40996]	21, 46944, 1041 [5468, 68724]
BACOLRI/ST	15, 2341, 415 [2770, 12515] {2354, 9272}	15, 4489, 720 [3222, 22942] {2501, 16799}	17, 9609, 976 [4147, 44306] {3170, 32715}
BACOLRI/LE	15, 2379, 513 [2976, 13054] {2462, 9517}	14, 4396, 974 [3350, 23262] {2375, 16860}	21, 9472, 1042 [2166, 43791] {1123, 32234}
p		9	
BACOLI/ST	14, 12371, 351 [1569, 19749]	15, 31020, 590 [4511, 43894]	14, 54847, 951 [9519, 83518]
BACOLI/LE	14, 11555, 433 [1563, 19208]	15, 27058, 744 [4629, 43575]	15, 85222, 1235 [10449, 88444]
BACOLRI/ST	14, 3144, 352 [3600, 15519] {3247, 11372}	14, 4923, 606 [4091, 24814] {3484, 18383}	15, 9579, 950 [4267, 45063] {3316, 33275}
BACOLRI/LE	14, 2849, 421 [3421, 14704] {2999, 10765}	15, 4567, 786 [3897, 23887] {3110, 17471}	14, 9617, 1172 [3974, 44895] {2801, 32917}

Table 12: *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		
BACOLI/ST	19, 5691, 173 [477, 8590]	43, 11876, 332 [889, 17253]	92, 24745, 413 [1546, 33523]
BACOLI/LE	50, 3669, 432 [931, 7385]	137, 11248, 723 [1585, 18444]	227, 24371, 1200 [2603, 36137]
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		
BACOLI/ST	15, 6708, 167 [499, 9803]	31, 12519, 327 [983, 18224]	50, 27114, 437 [1952, 37200]
BACOLI/LE	19, 4276, 240 [573, 7096]	40, 12352, 394 [1040, 17830]	93, 24097, 427 [1110, 31144]
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		
BACOLI/ST	16, 7505, 116 [495, 10723]	20, 14881, 328 [1683, 22580]	27, 29530, 418 [2800, 41872]
BACOLI/LE	15, 5789, 183 [552, 9020]	24, 12708, 412 [1222, 19198]	33, 26190, 438 [1564, 35322]
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		
BACOLI/ST	15, 9863, 136 [791, 14212]	18, 28626, 314 [2194, 26569]	16, 31905, 431 [3543, 45734]
BACOLI/LE	15, 8196, 152 [805, 12053]	16, 16248, 404 [2515, 25306]	21, 29803, 514 [3344, 43622]
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 13: *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		
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		
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		
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		
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 14: *Machine independent results for the Schrödinger Equation. 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}.*

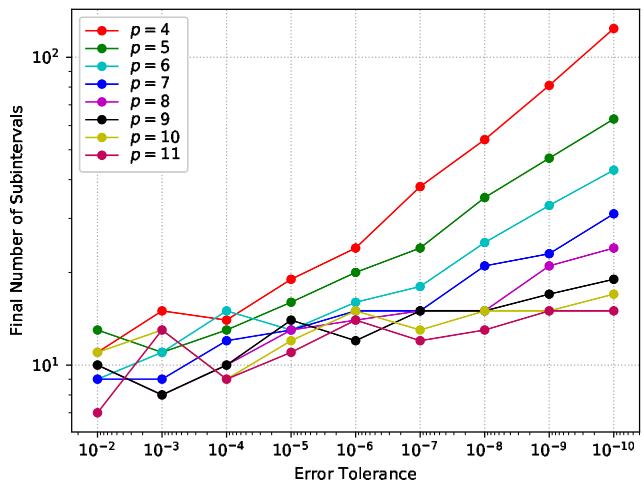

Figure 1: BACOLI/ST Number of Subintervals vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

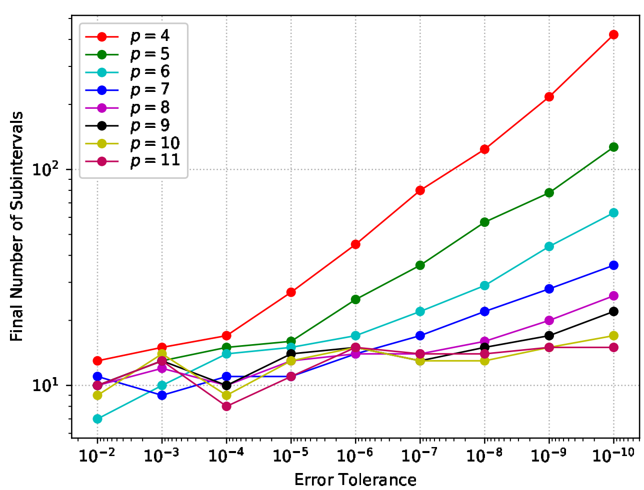

Figure 2: BACOLI/LE Number of Subintervals vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

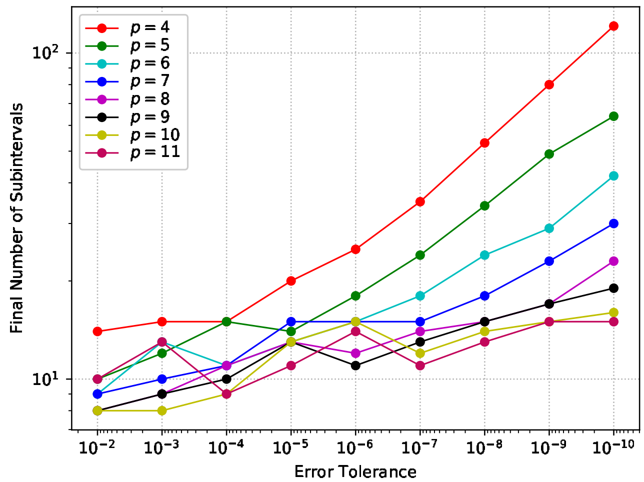

Figure 3: BACOLRI/ST Number of Subintervals vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

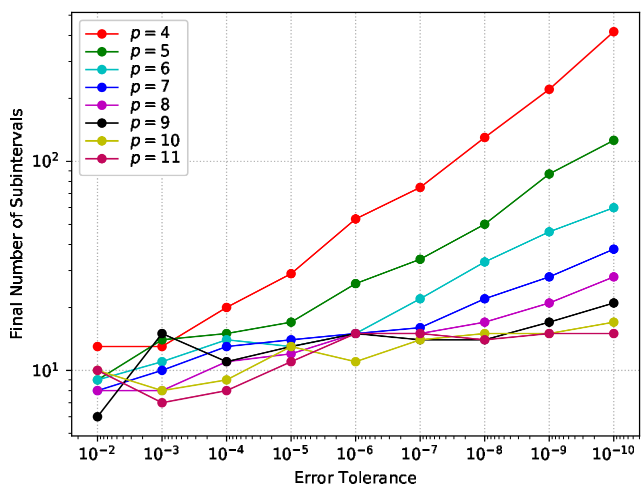

Figure 4: BACOLRI/LE Number of Subintervals vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

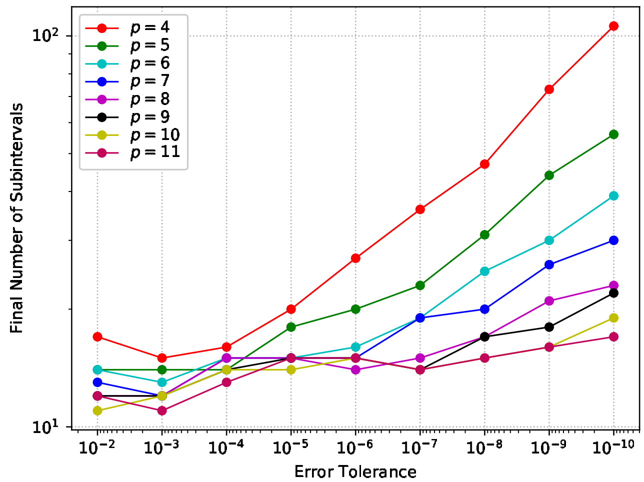

Figure 5: BACOLI/ST Number of Subintervals vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

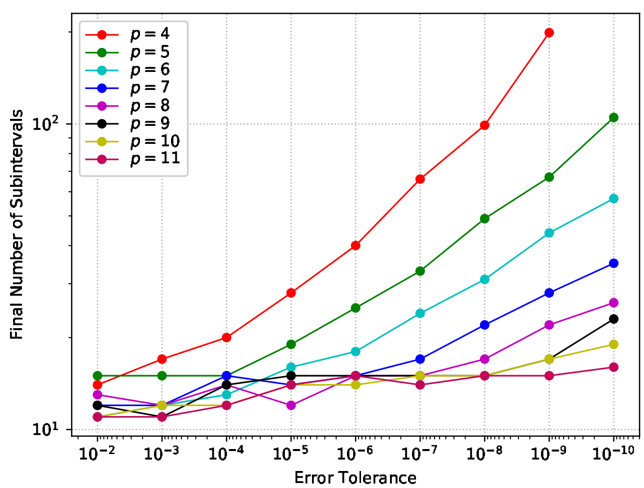

Figure 6: BACOLI/LE Number of Subintervals vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

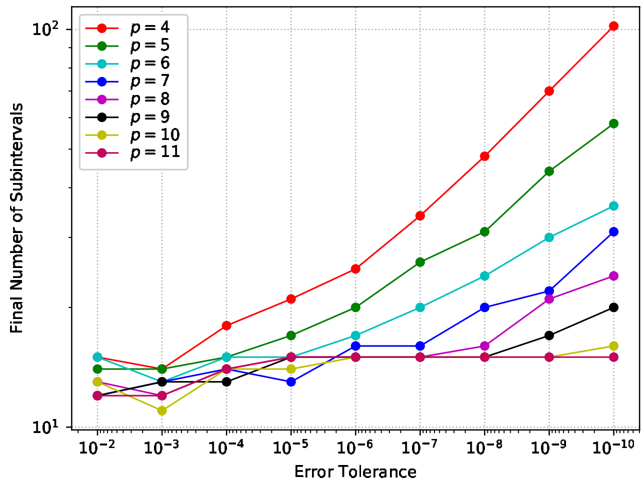

Figure 7: BACOLRI/ST Number of Subintervals vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

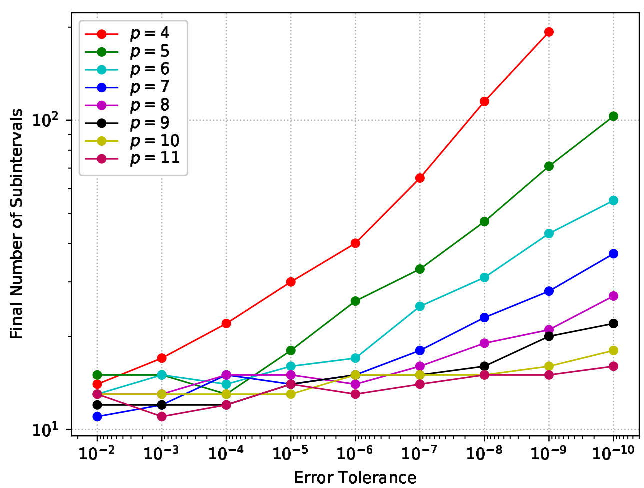

Figure 8: BACOLRI/LE Number of Subintervals vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

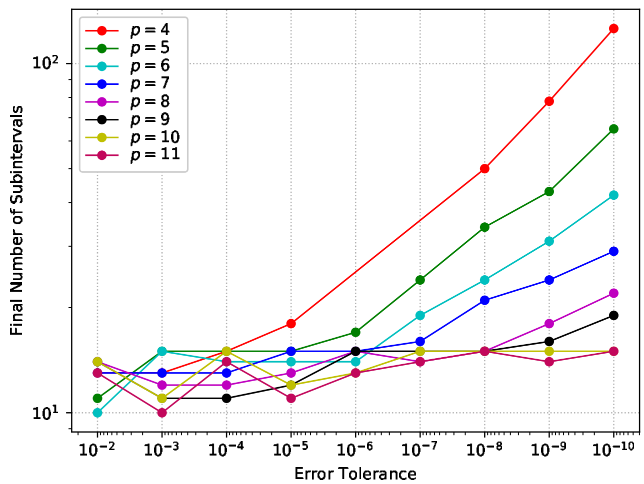

Figure 9: BACOLI/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

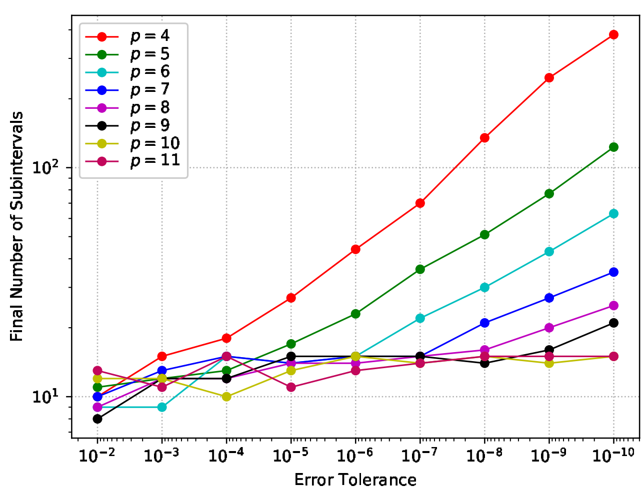

Figure 10: BACOLI/LE Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

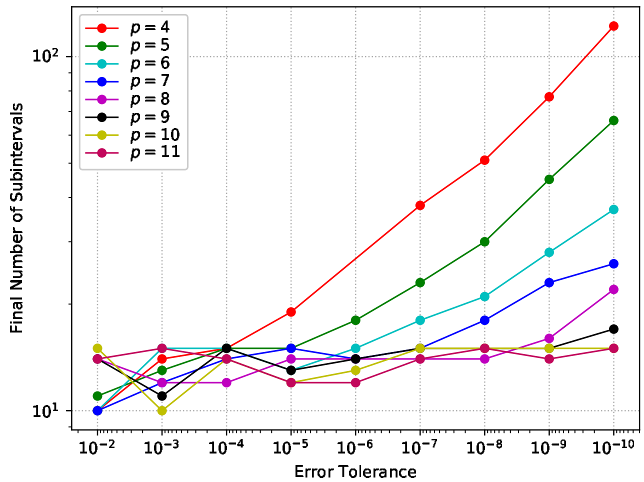

Figure 11: BACOLRI/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

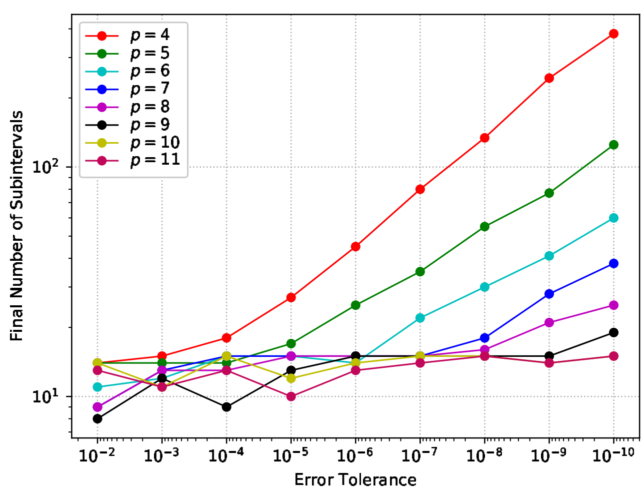

Figure 12: BACOLRI/LE Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

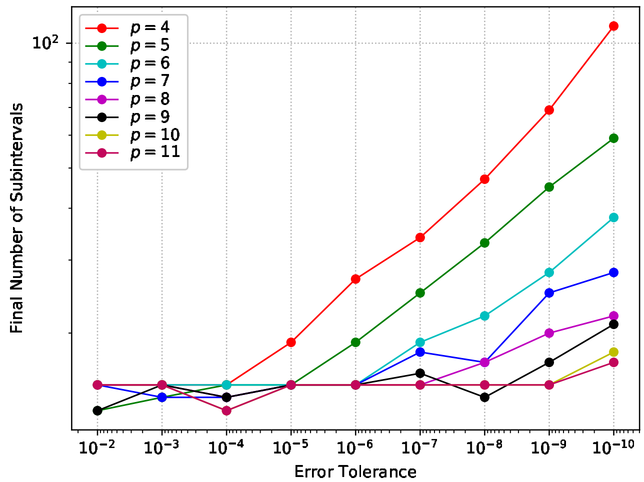

Figure 13: BACOLI/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

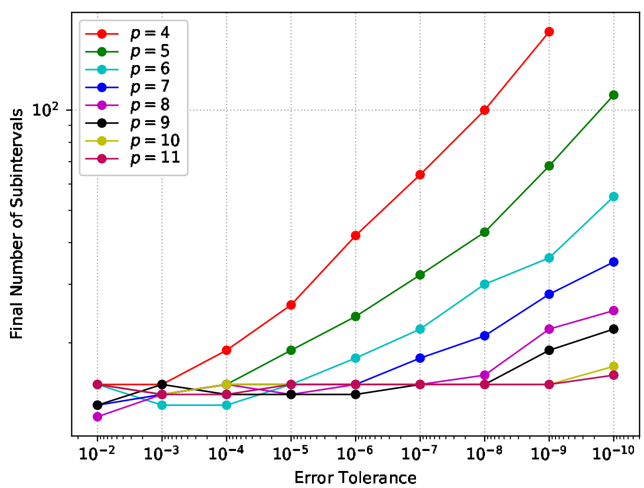

Figure 14: BACOLI/LE Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

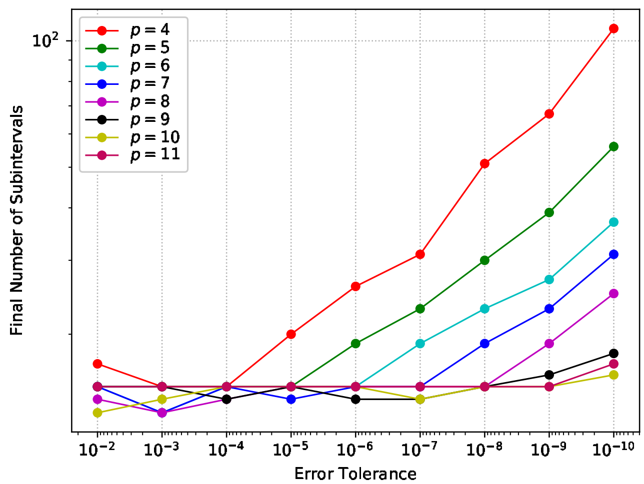

Figure 15: BACOLRI/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

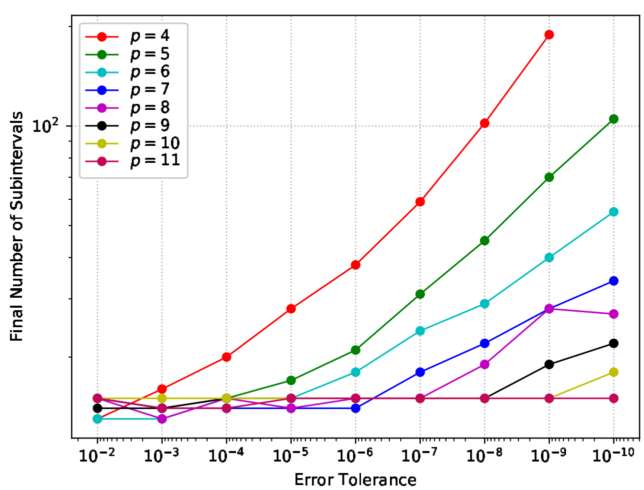

Figure 16: BACOLRI/LE Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

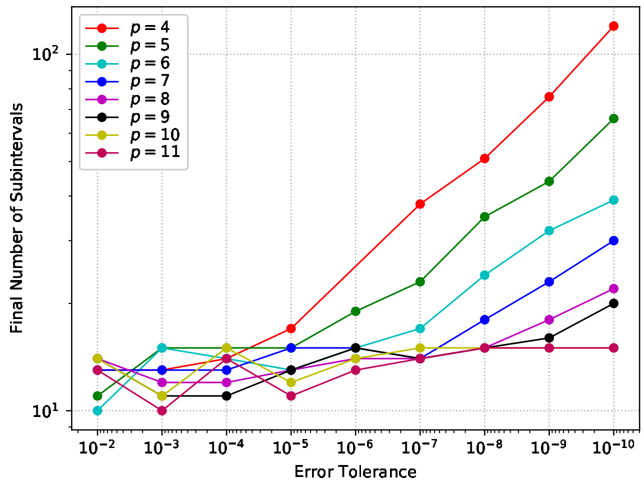

Figure 17: BACOLI/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

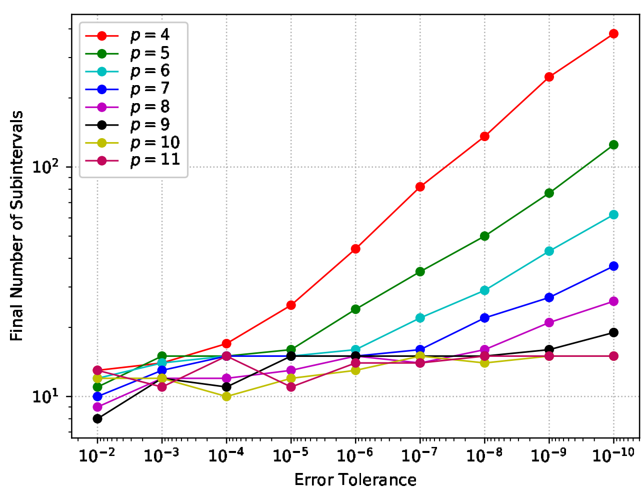

Figure 18: BACOLI/LE Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

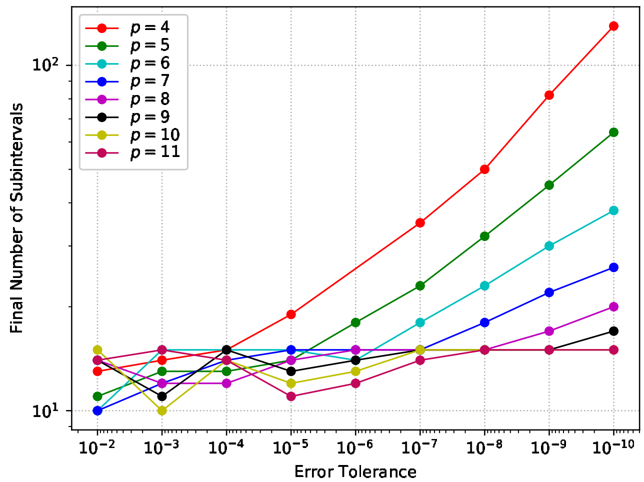

Figure 19: BACOLRI/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

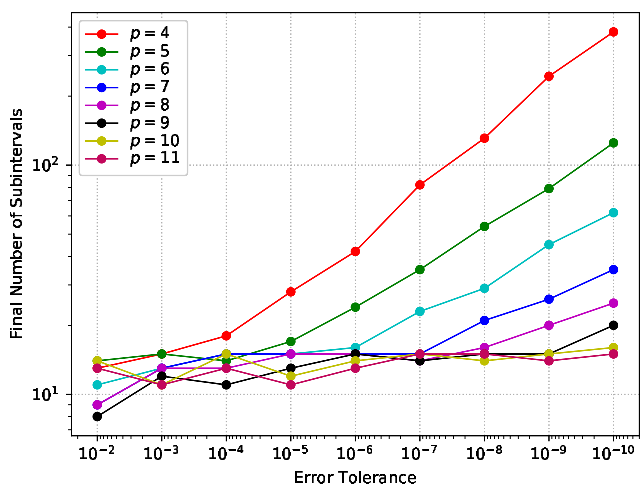

Figure 20: BACOLRI/LE Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

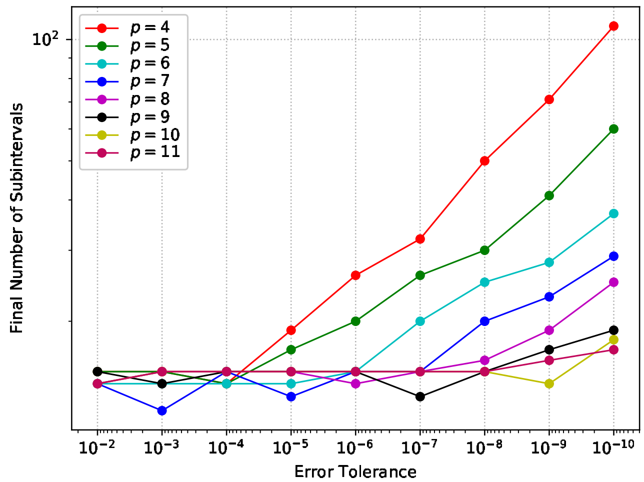

Figure 21: BACOLI/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

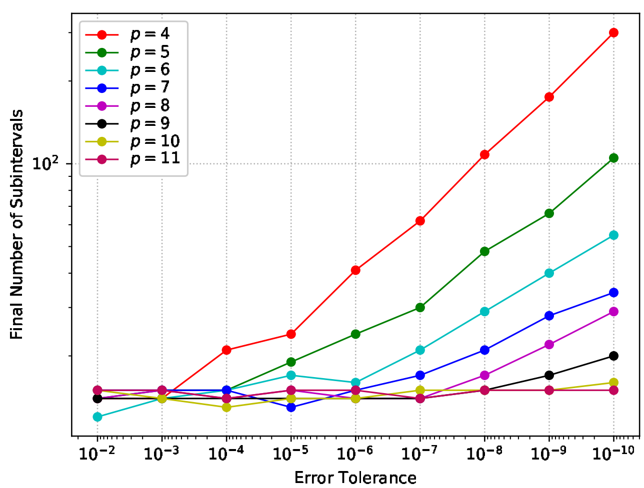

Figure 22: BACOLI/LE Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

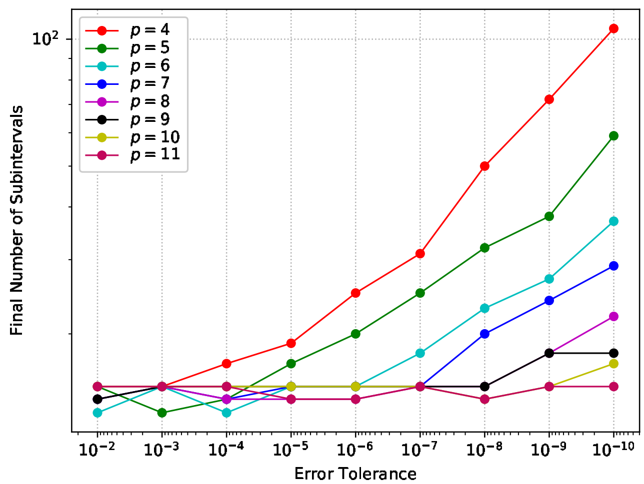

Figure 23: BACOLRI/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

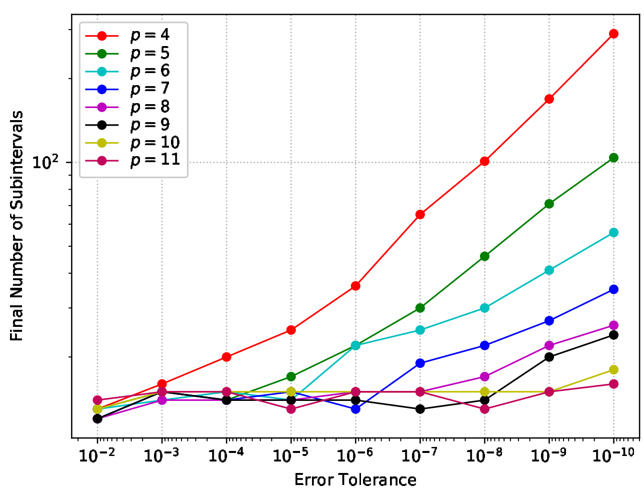

Figure 24: BACOLRI/LE Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

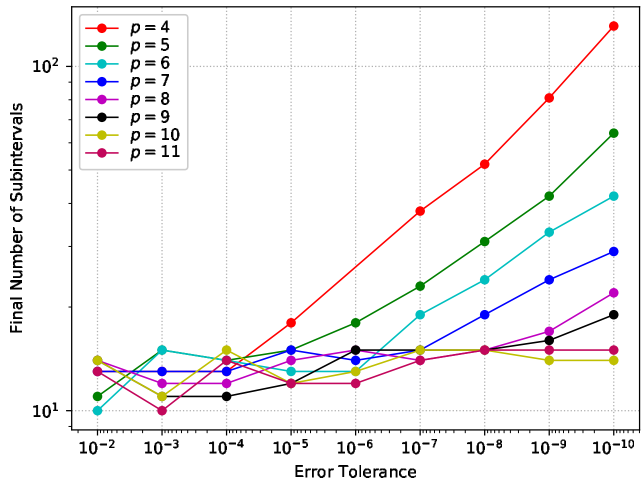

Figure 25: BACOLI/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

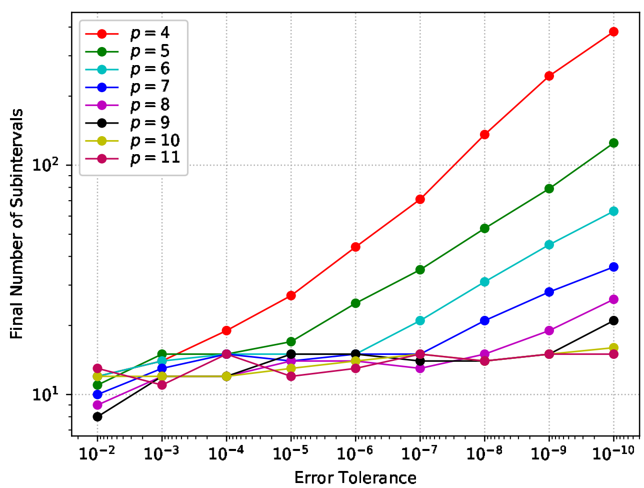

Figure 26: BACOLI/LE Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

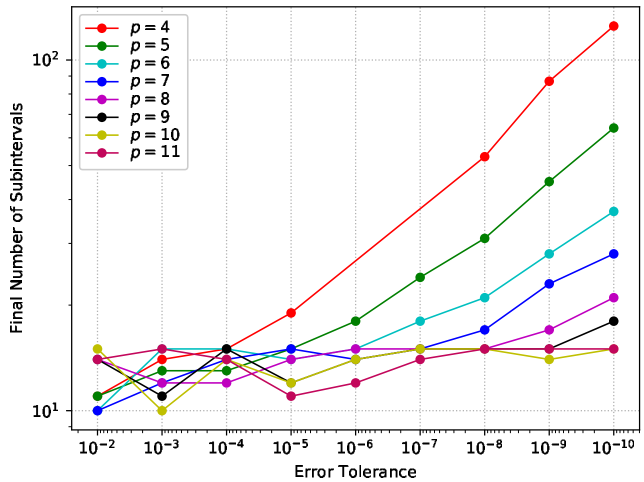

Figure 27: BACOLRI/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

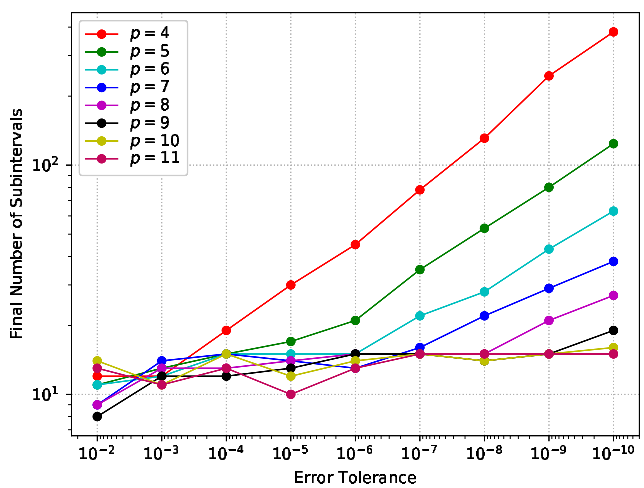

Figure 28: BACOLRI/LE Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

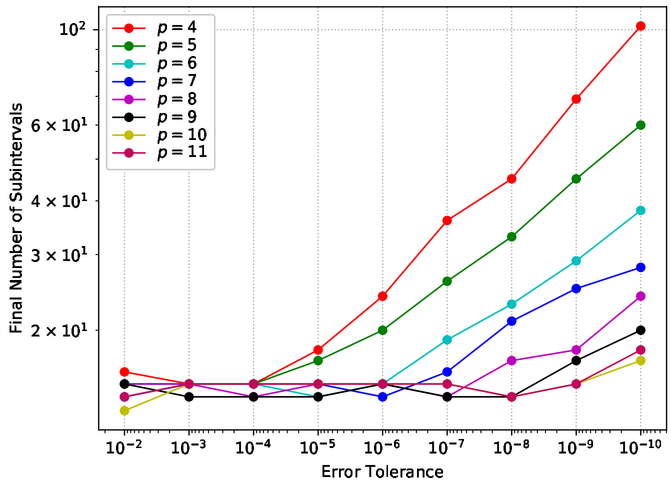

Figure 29: BACOLI/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

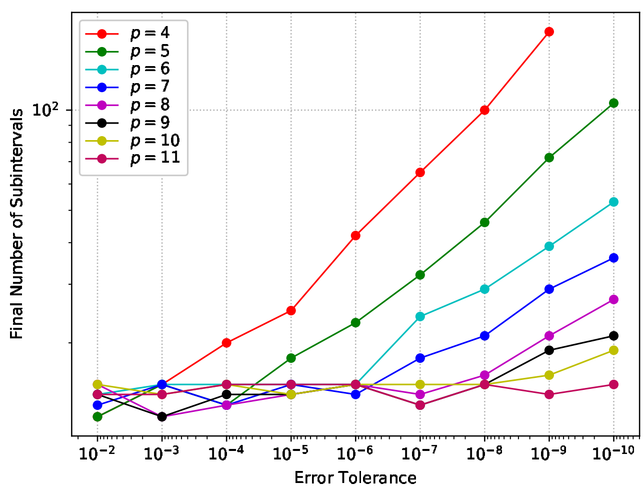

Figure 30: BACOLI/LE Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

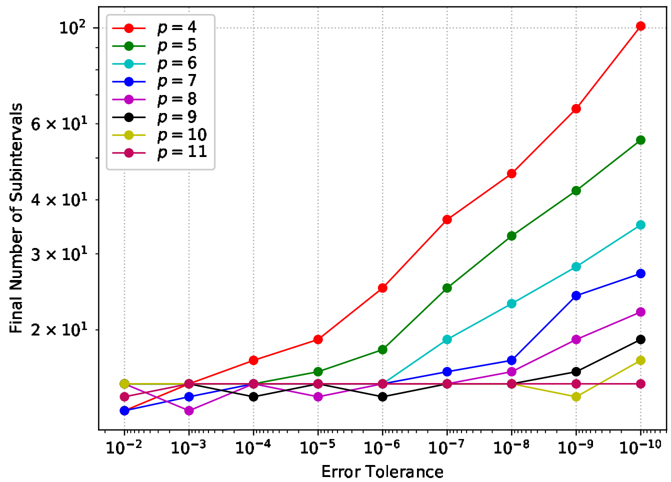

Figure 31: BACOLRI/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

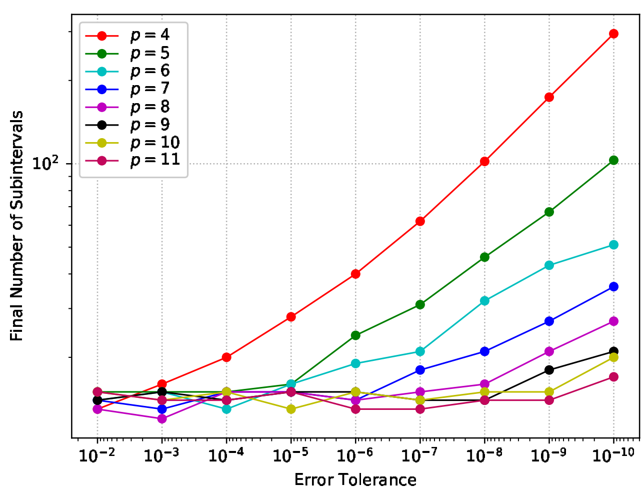

Figure 32: BACOLRI/LE Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

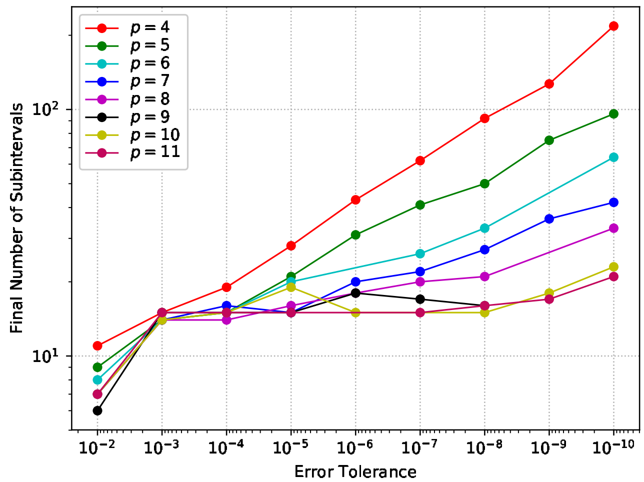

Figure 33: BACOLI/ST Number of Subintervals vs. Error Tolerance: Catalytic Surface Reaction Model with $p = 4 \ldots 11$

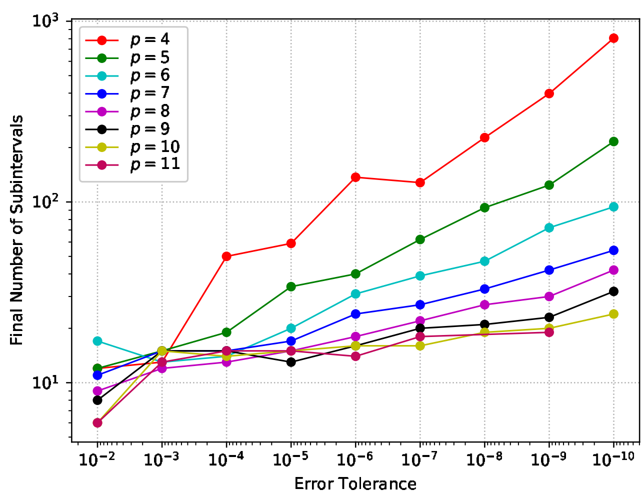

Figure 34: BACOLI/LE Number of Subintervals vs. Error Tolerance: Catalytic Surface Reaction Model with $p = 4 \ldots 11$

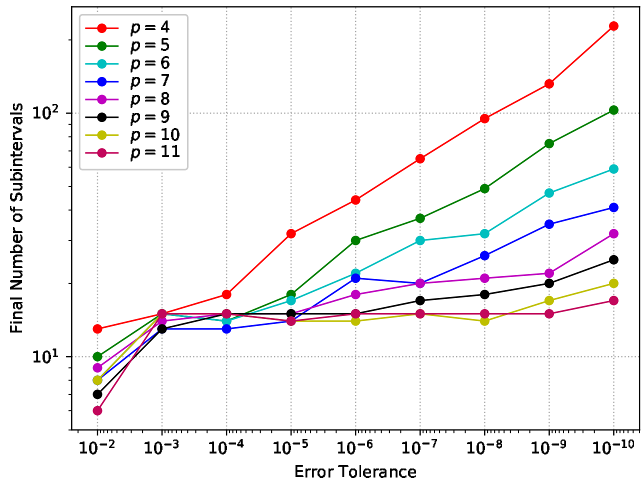

Figure 35: BACOLRI/ST Number of Subintervals vs. Error Tolerance: Catalytic Surface Reaction Model with $p = 4 \ldots 11$

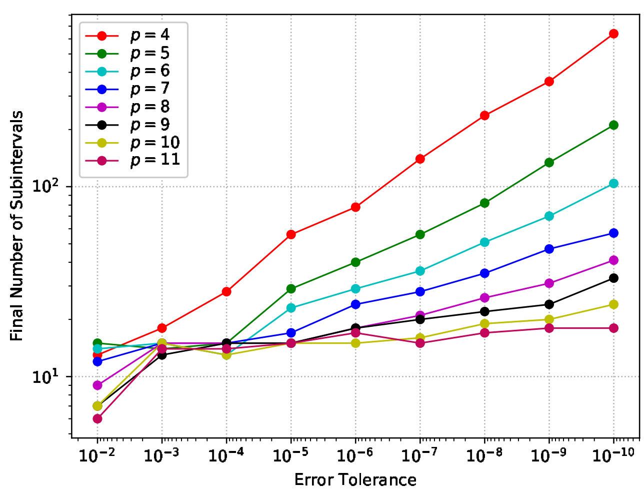

Figure 36: BACOLRI/LE Number of Subintervals vs. Error Tolerance: Catalytic Surface Reaction Model with $p = 4 \ldots 11$

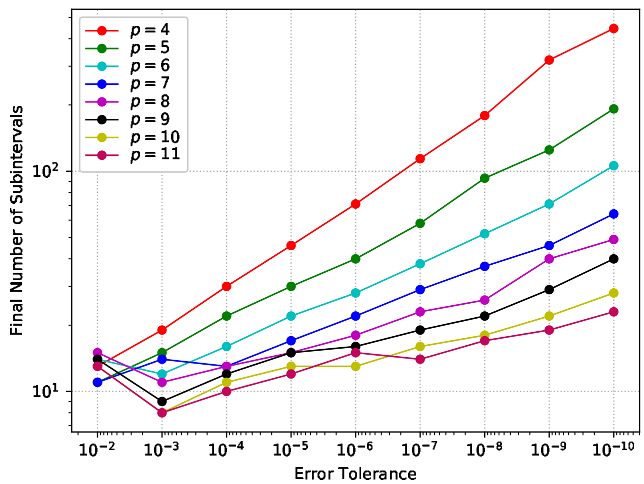

Figure 37: BACOLRI/ST Number of Subintervals vs. Error Tolerance: Schrödinger System with $p = 4 \ldots 11$

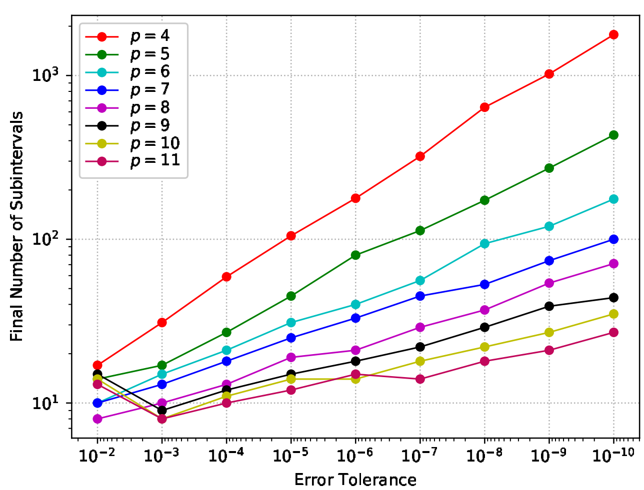

Figure 38: BACOLRI/LE Number of Subintervals vs. Error Tolerance: Schrödinger System with $p = 4 \ldots 11$

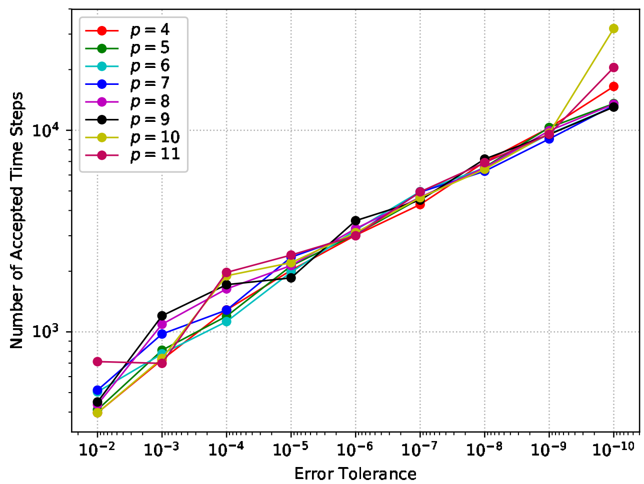

Figure 39: BACOLI/ST Number of Accepted Time Steps vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

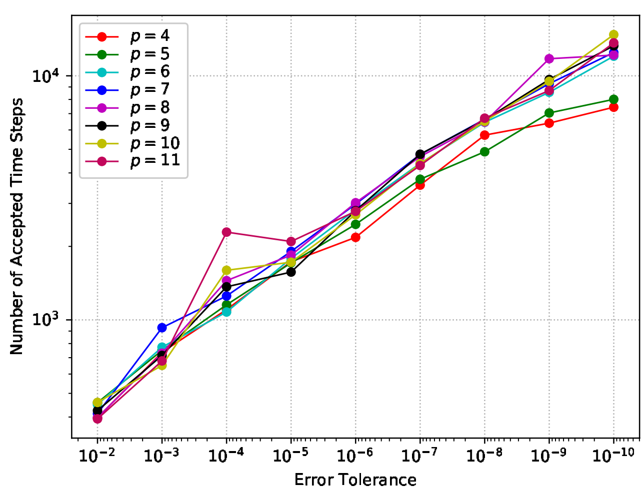

Figure 40: BACOLI/LE Number of Accepted Time Steps vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

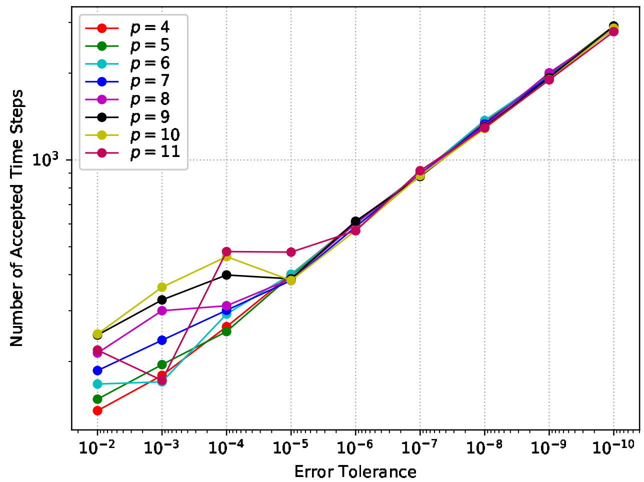

Figure 41: BACOLRI/ST Number of Accepted Time Steps vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

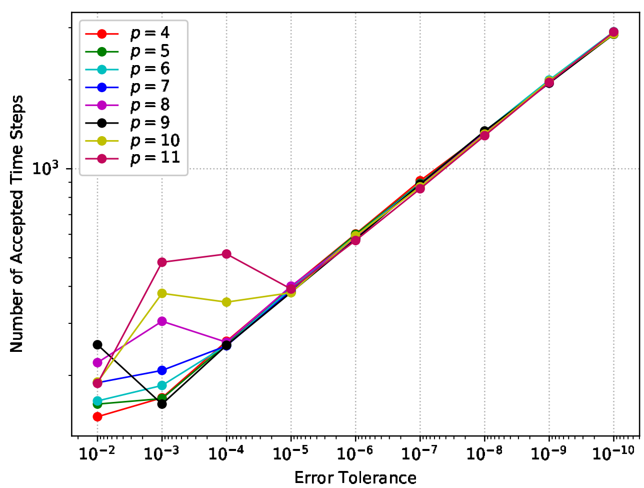

Figure 42: BACOLRI/LE Number of Accepted Time Steps vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

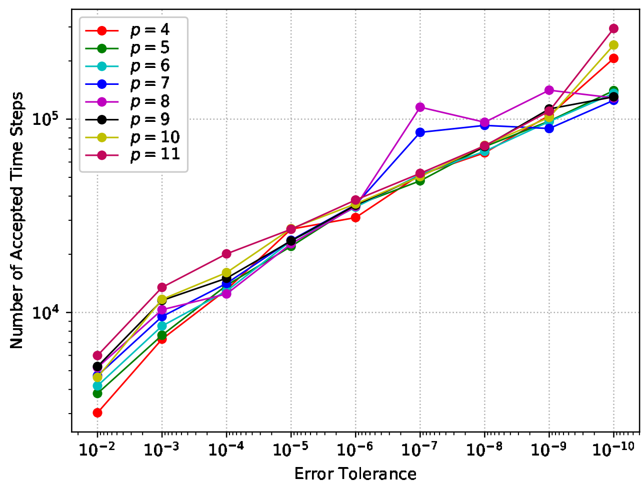

Figure 43: BACOLI/ST Number of Accepted Time Steps vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

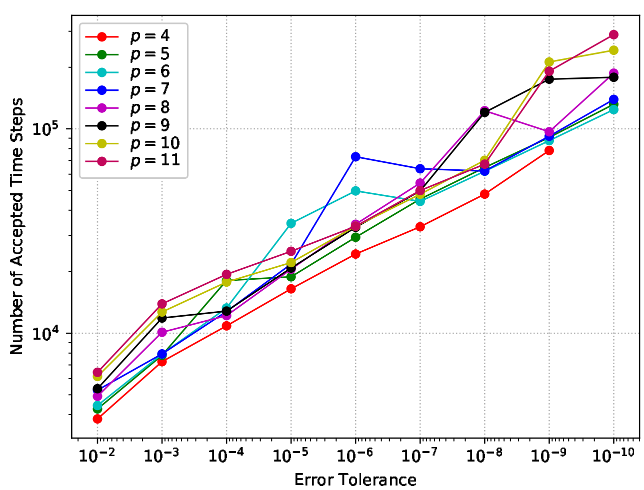

Figure 44: BACOLI/LE Number of Accepted Time Steps vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

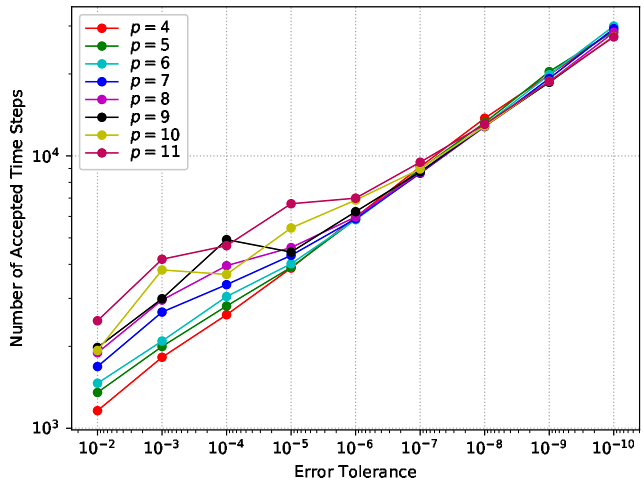

Figure 45: BACOLRI/ST Number of Accepted Time Steps vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

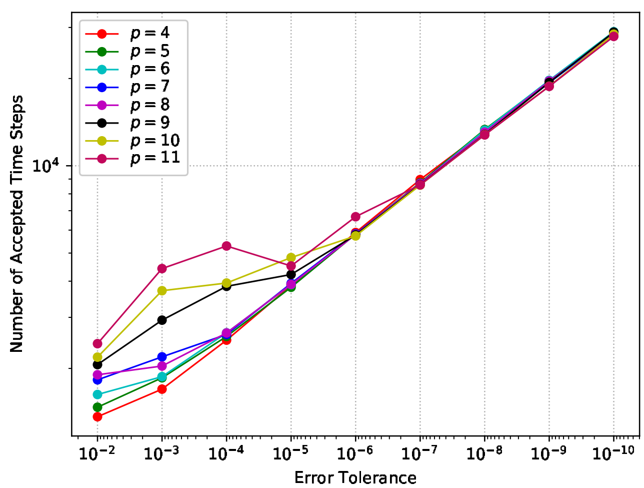

Figure 46: BACOLRI/LE Number of Accepted Time Steps vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

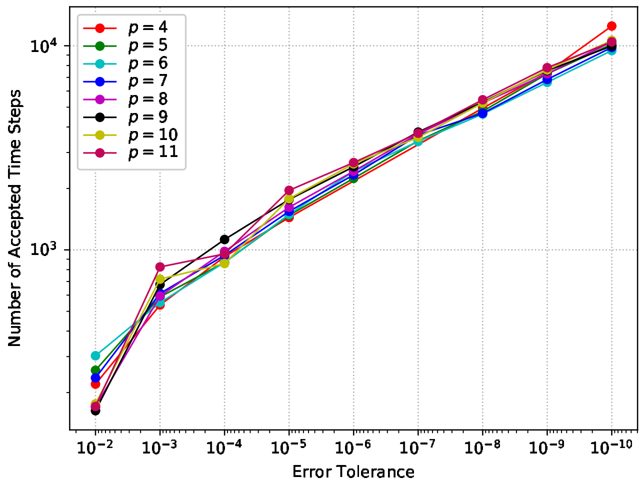

Figure 47: BACOLI/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

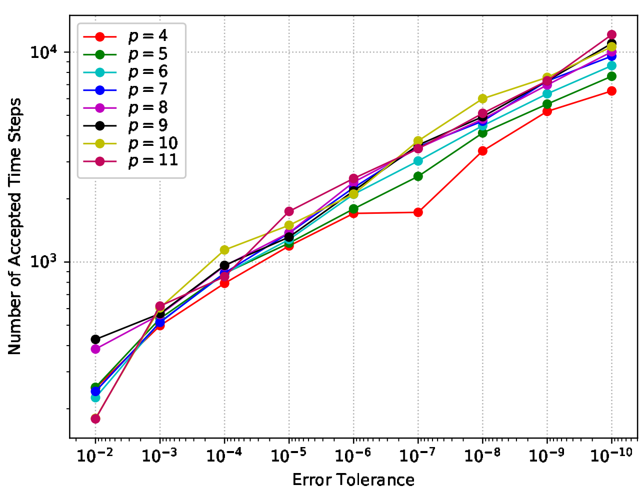

Figure 48: BACOLI/LE Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

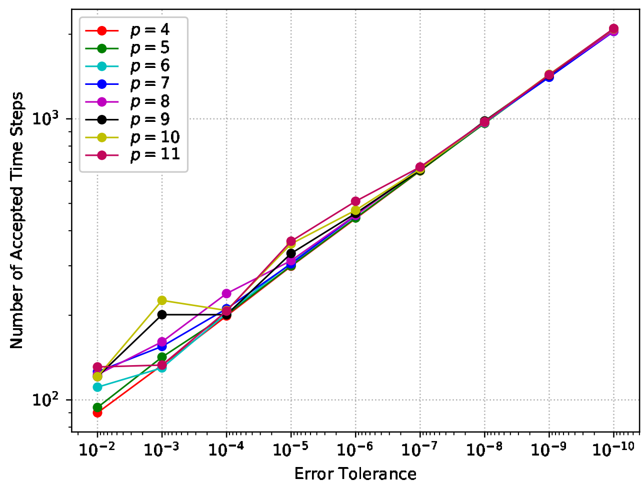

Figure 49: BACOLRI/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

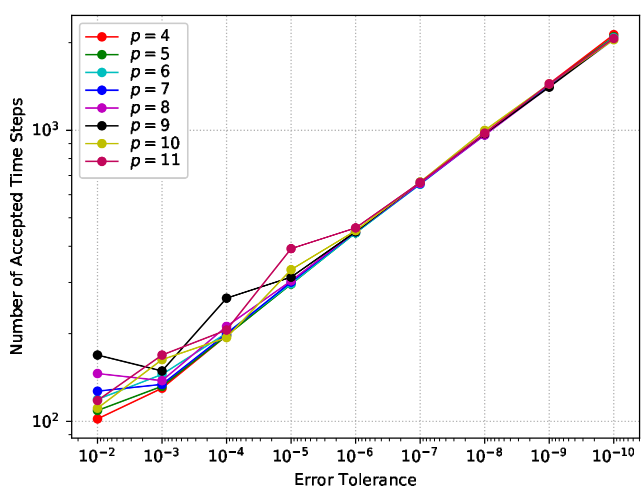

Figure 50: BACOLRI/LE Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

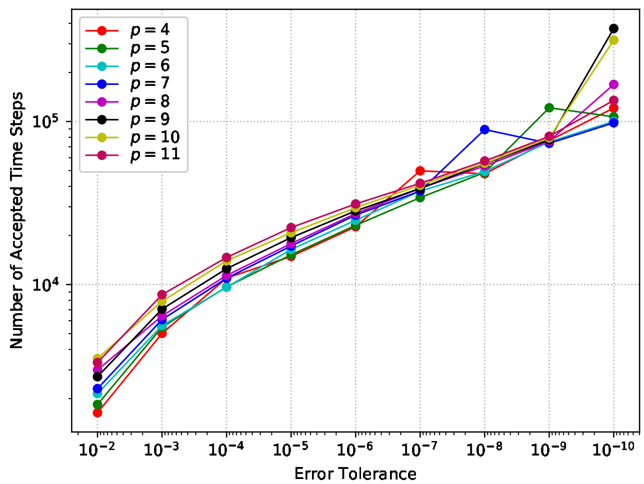

Figure 51: BACOLI/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

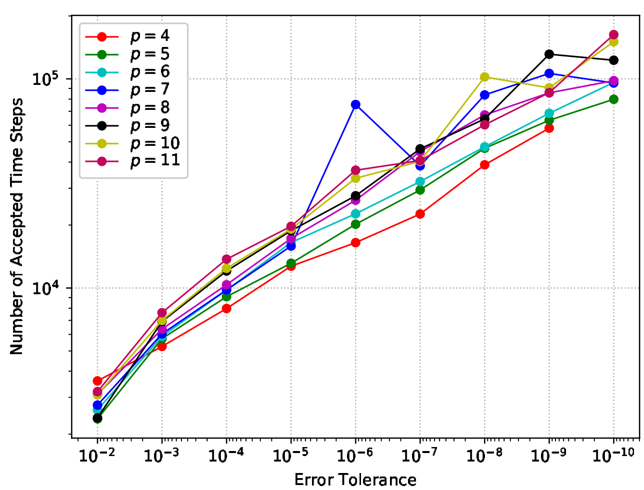

Figure 52: BACOLI/LE Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

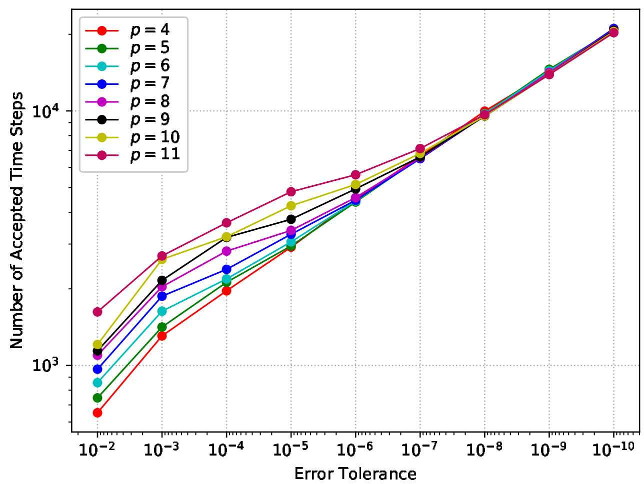

Figure 53: BACOLRI/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

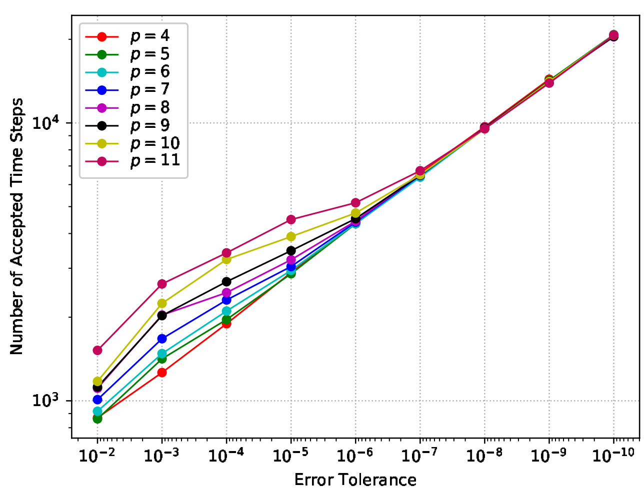

Figure 54: BACOLRI/LE Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

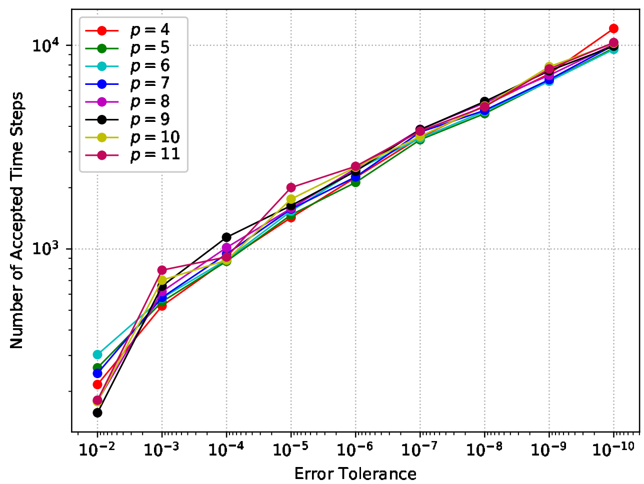

Figure 55: BACOLI/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

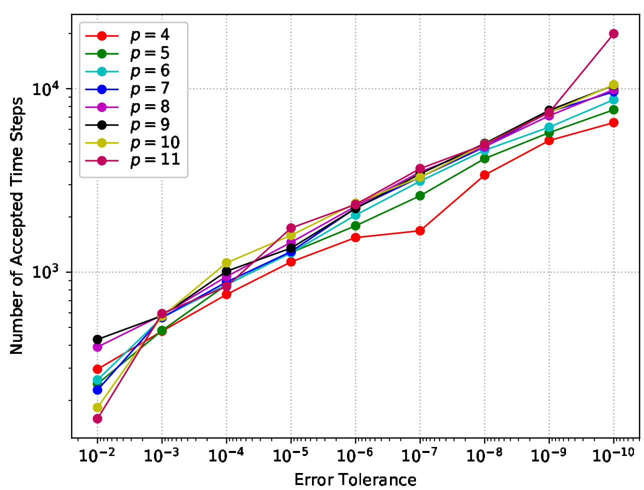

Figure 56: BACOLI/LE Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

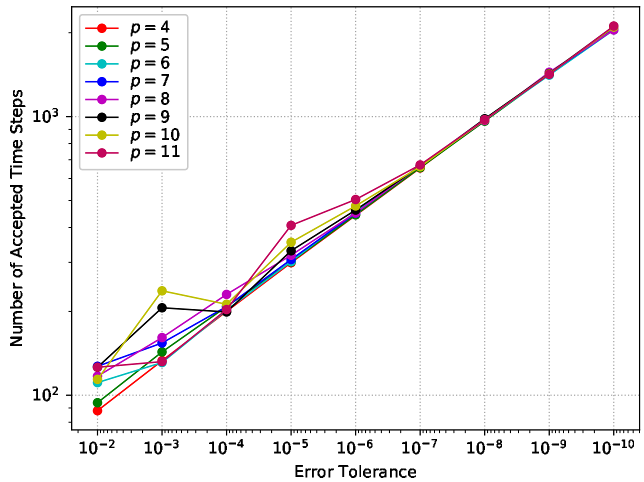

Figure 57: BACOLRI/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 4 \dots 11$

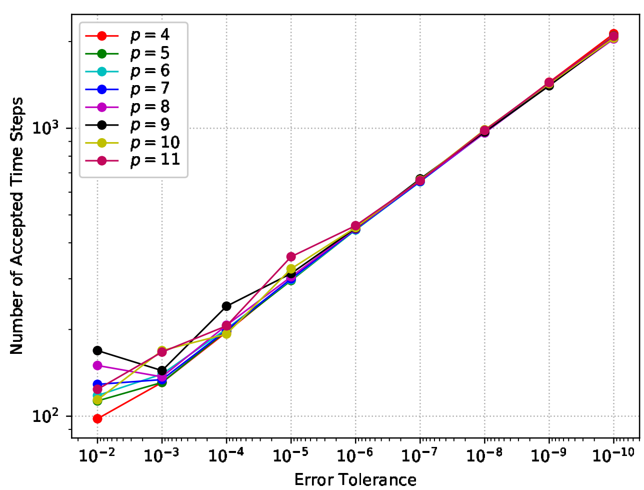

Figure 58: BACOLRI/LE Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 4 \dots 11$

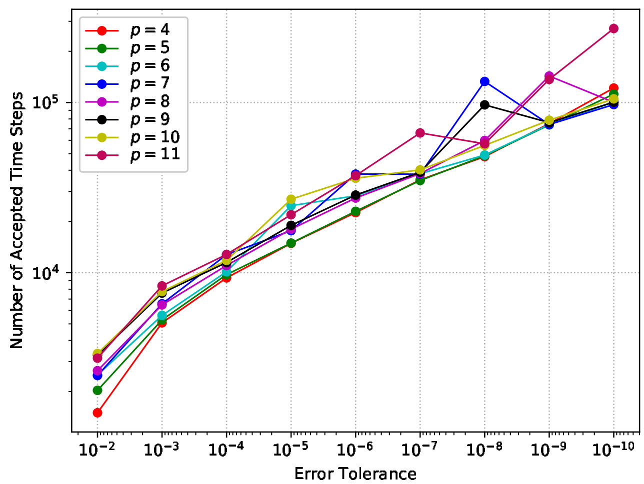

Figure 59: BACOLI/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

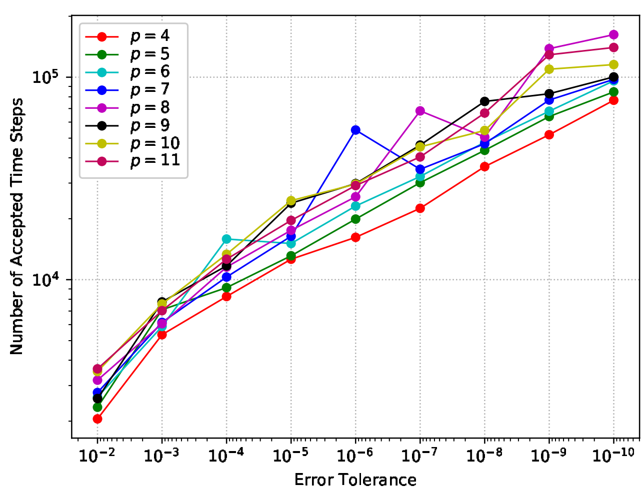

Figure 60: BACOLI/LE Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

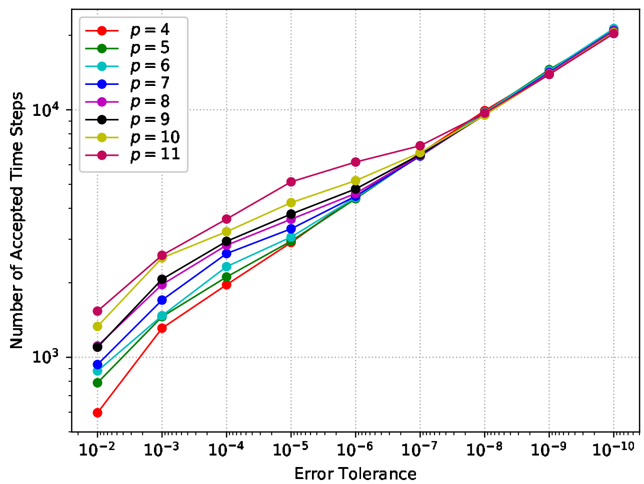

Figure 61: BACOLRI/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

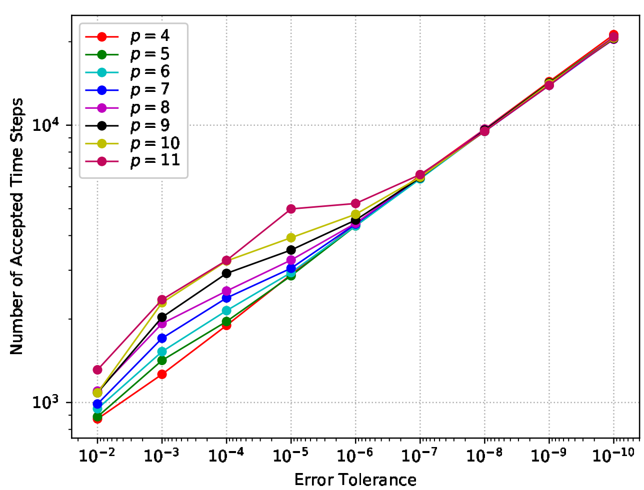

Figure 62: BACOLRI/LE Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

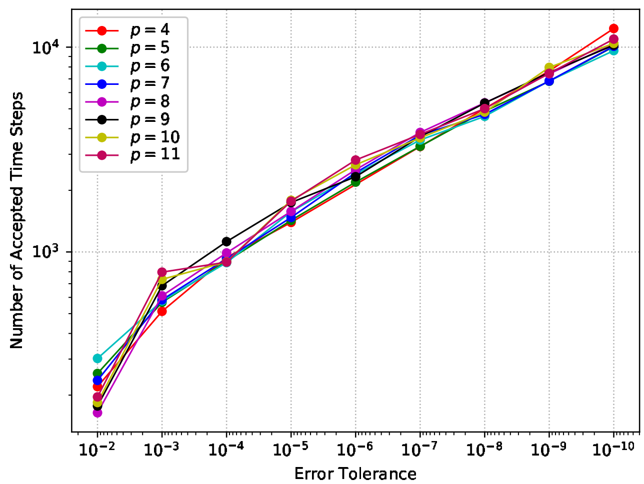

Figure 63: BACOLI/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

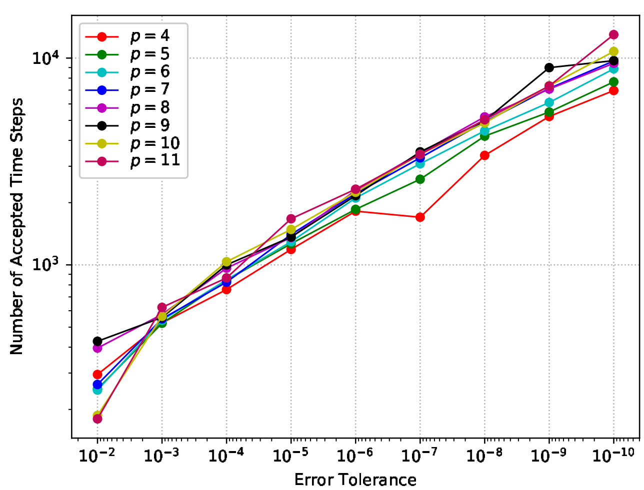

Figure 64: BACOLI/LE Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

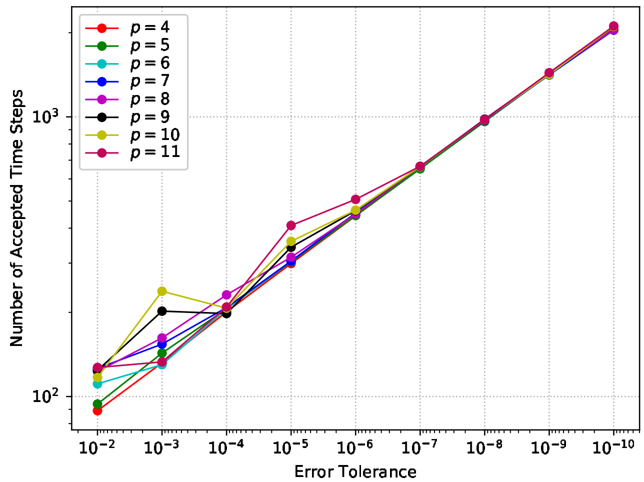

Figure 65: BACOLRI/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

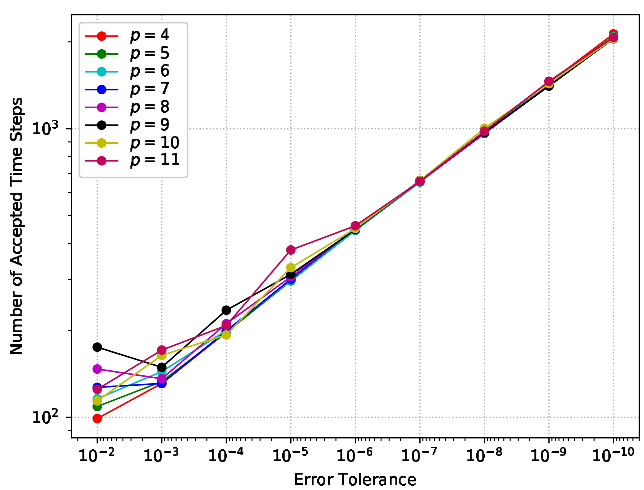

Figure 66: BACOLRI/LE Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

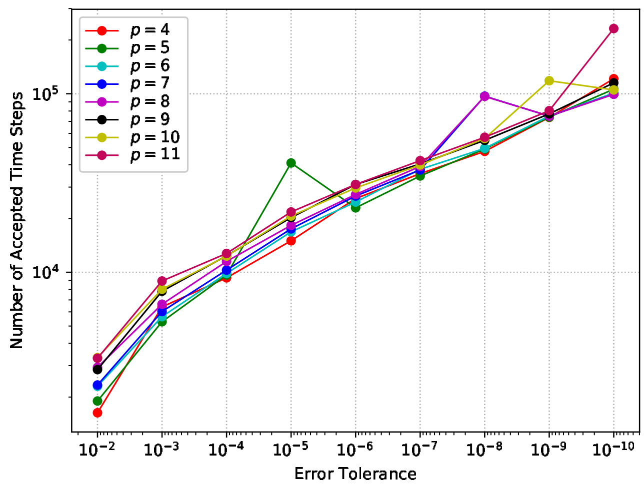

Figure 67: BACOLI/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

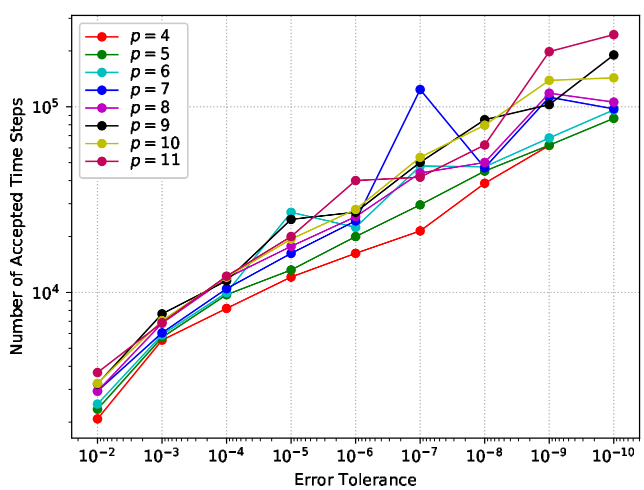

Figure 68: BACOLI/LE Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

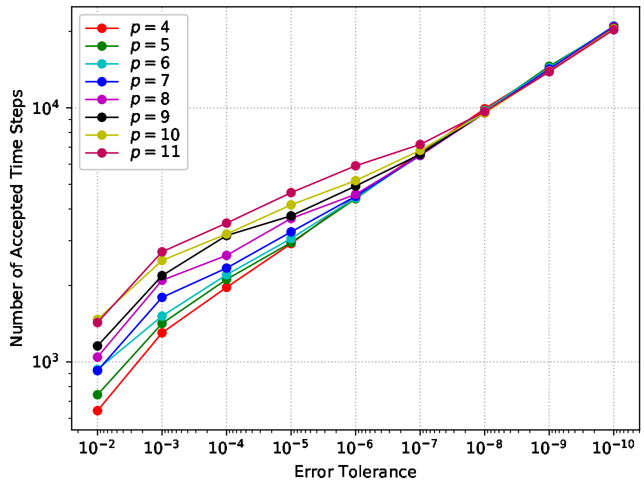

Figure 69: BACOLRI/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

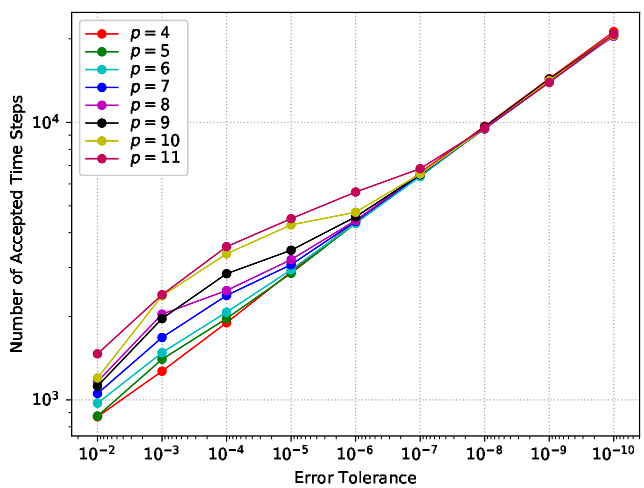

Figure 70: BACOLRI/LE Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

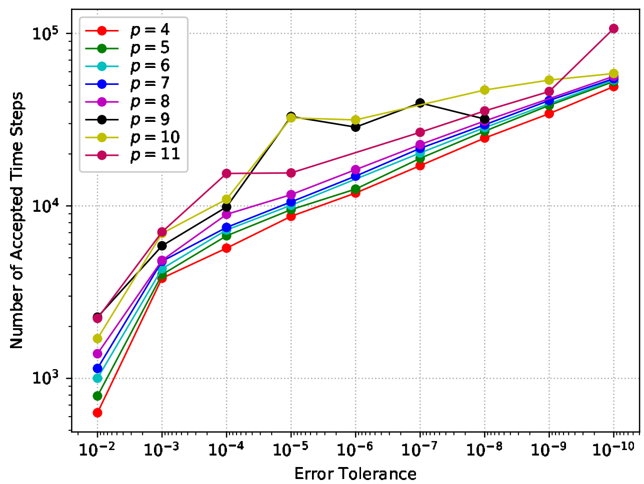

Figure 71: BACOLI/ST Number of Accepted Time Steps vs. Error Tolerance: Catalytic Surface Reaction Model with $p = 4 \ldots 11$

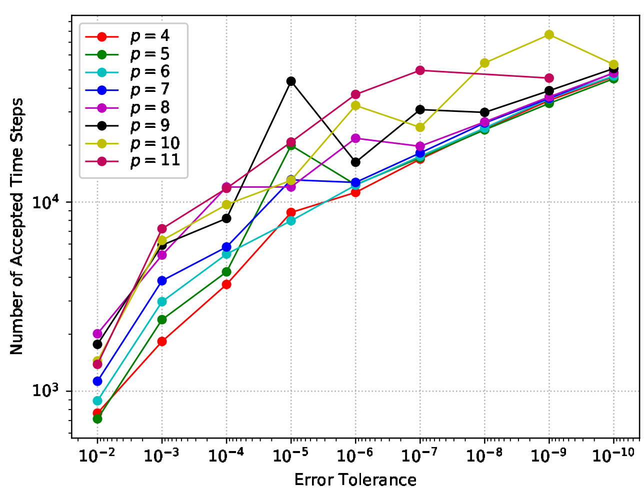

Figure 72: BACOLI/LE Number of Accepted Time Steps vs. Error Tolerance: Catalytic Surface Reaction Model with $p = 4 \ldots 11$

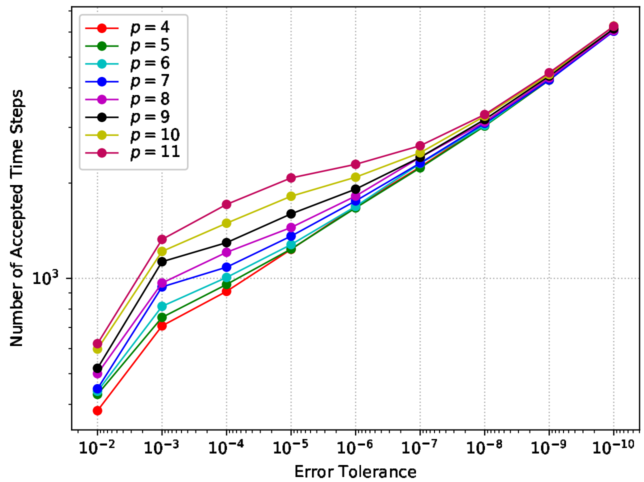

Figure 73: BACOLRI/ST Number of Accepted Time Steps vs. Error Tolerance: Catalytic Surface Reaction Model with $p = 4 \ldots 11$

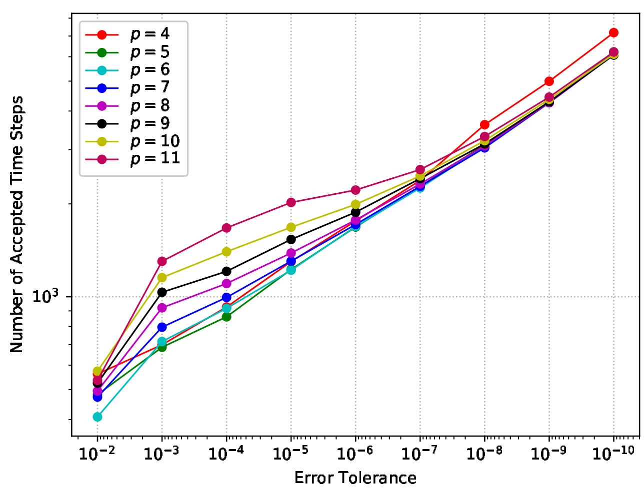

Figure 74: BACOLRI/LE Number of Accepted Time Steps vs. Error Tolerance: Catalytic Surface Reaction Model with $p = 4 \ldots 11$

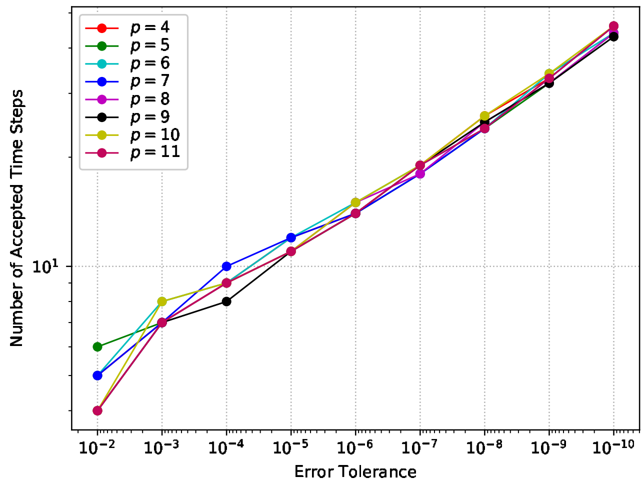

Figure 75: BACOLRI/ST Number of Accepted Time Steps vs. Error Tolerance: Schrödinger System with $p = 4 \ldots 11$

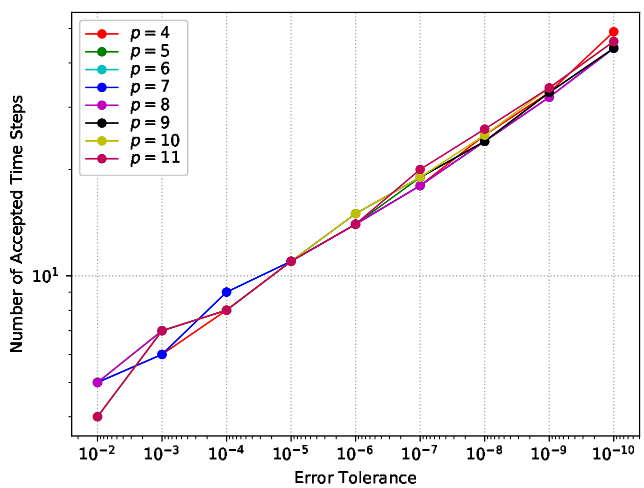

Figure 76: BACOLRI/LE Number of Accepted Time Steps vs. Error Tolerance: Schrödinger System with $p = 4 \ldots 11$

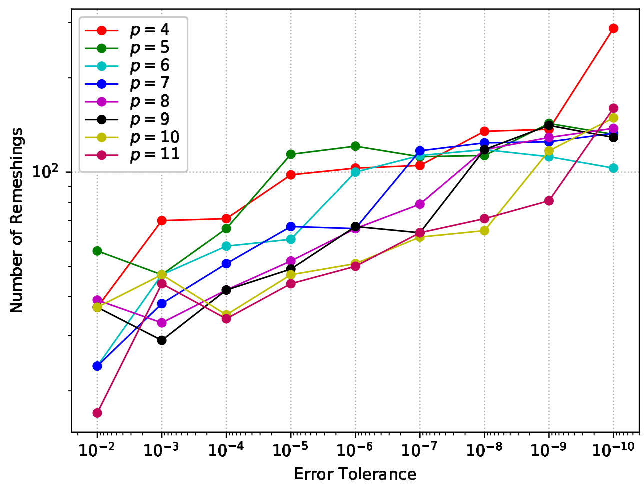

Figure 77: BACOLI/ST Number of Remeshings vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

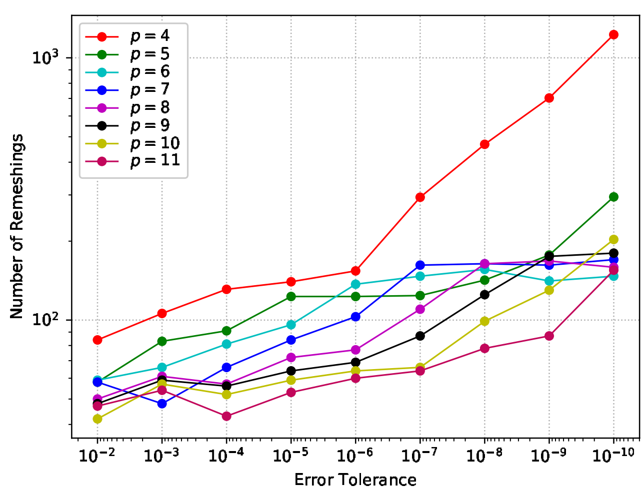

Figure 78: BACOLI/LE Number of Remeshings vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

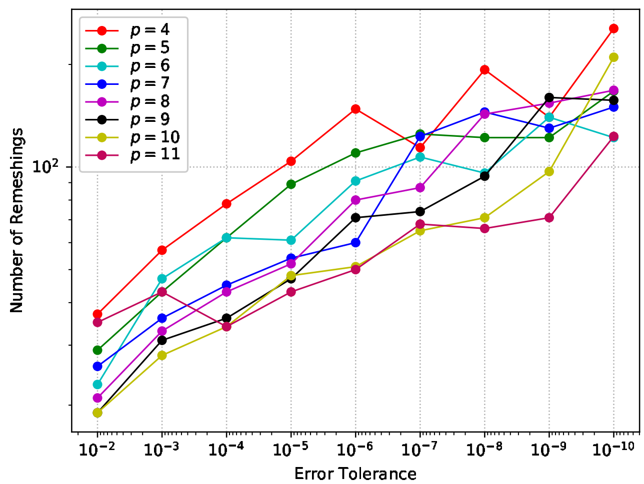

Figure 79: BACOLRI/ST Number of Remeshings vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \dots 11$

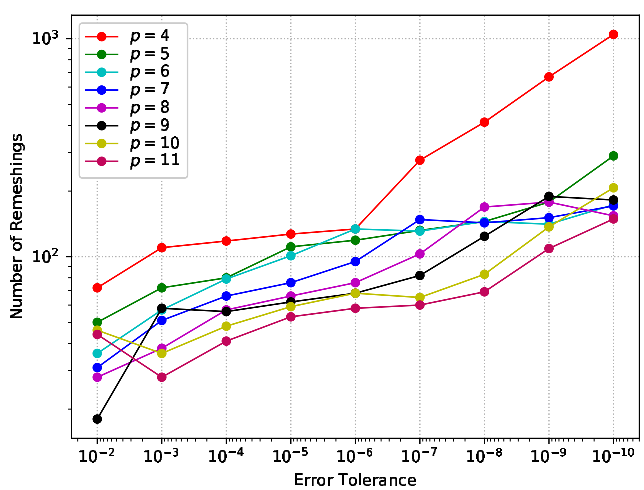

Figure 80: BACOLRI/LE Number of Remeshings vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \dots 11$

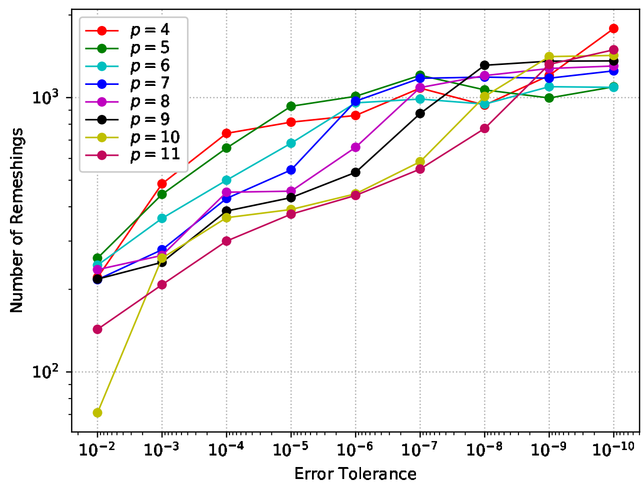

Figure 81: BACOLI/ST Number of Remeshings vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

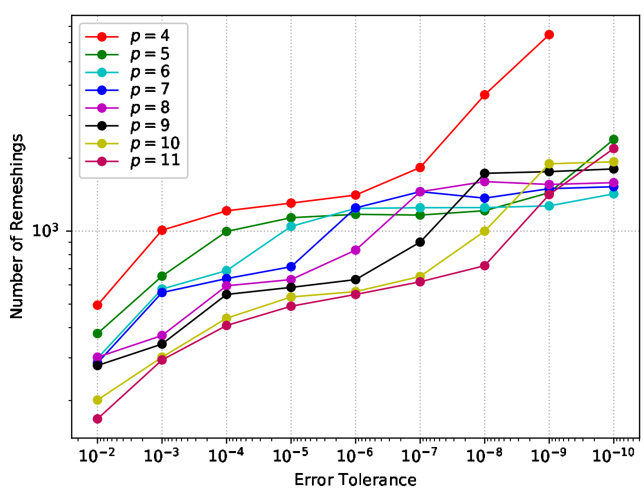

Figure 82: BACOLI/LE Number of Remeshings vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

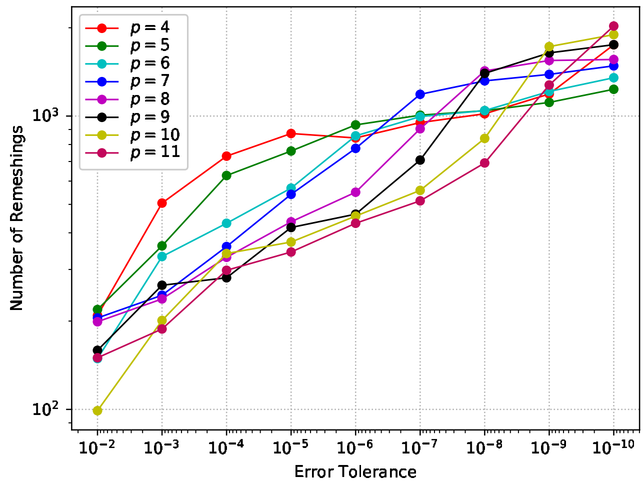

Figure 83: BACOLRI/ST Number of Remeshings vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

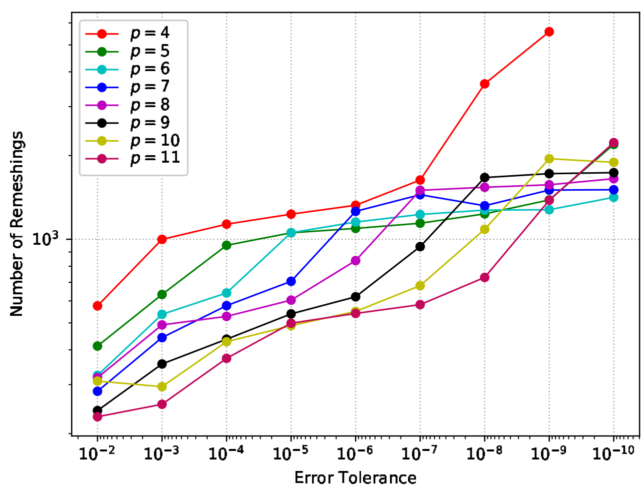

Figure 84: BACOLRI/LE Number of Remeshings vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

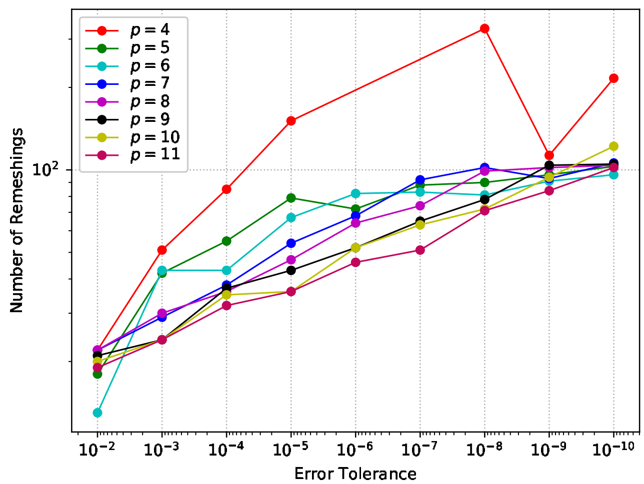

Figure 85: BACOLI/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

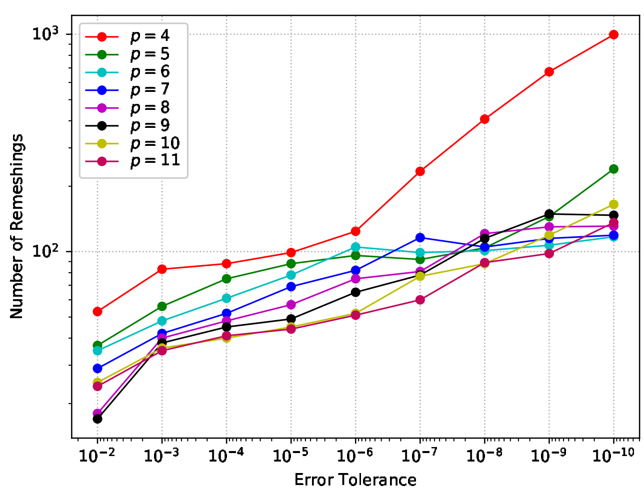

Figure 86: BACOLI/LE Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

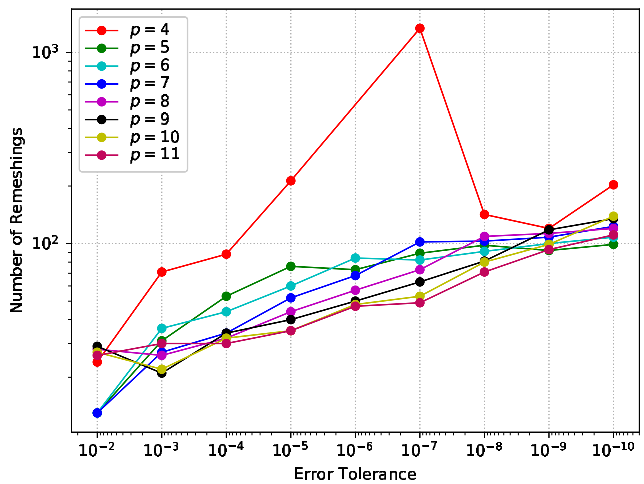

Figure 87: BACOLRI/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

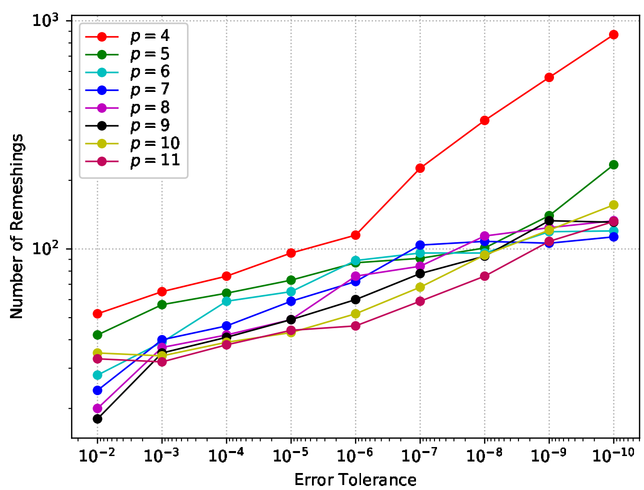

Figure 88: BACOLRI/LE Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

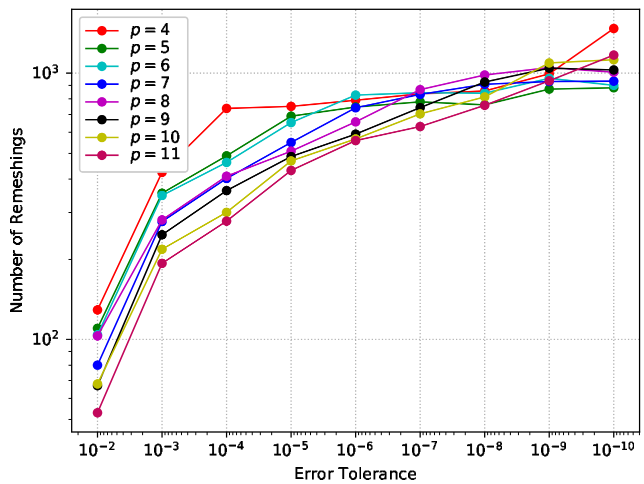

Figure 89: BACOLI/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

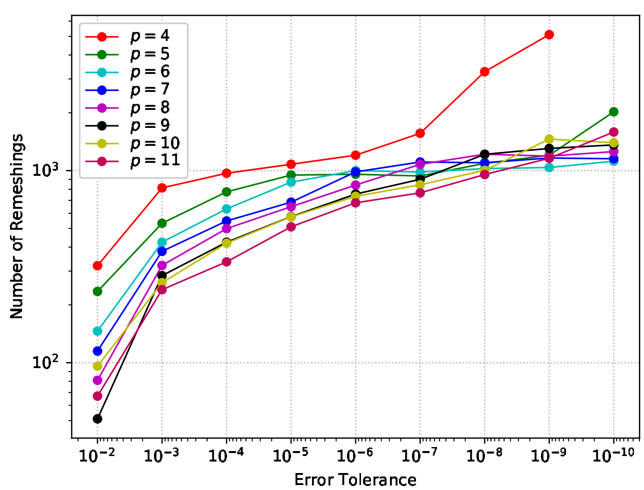

Figure 90: BACOLI/LE Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

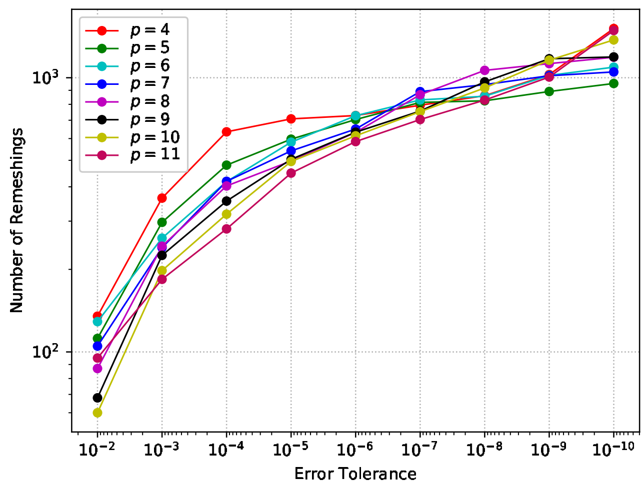

Figure 91: BACOLRI/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

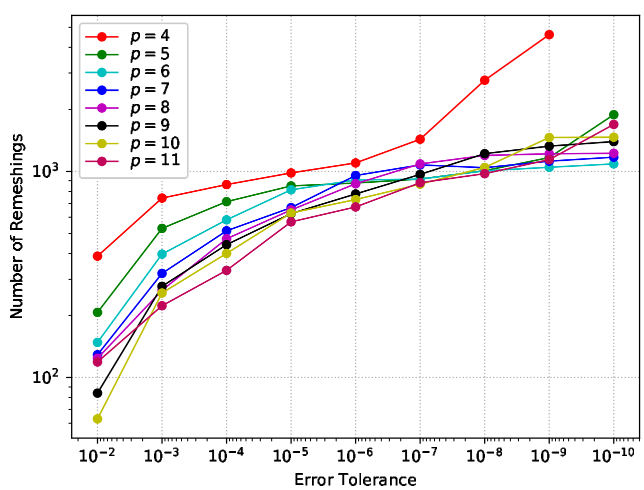

Figure 92: BACOLRI/LE Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

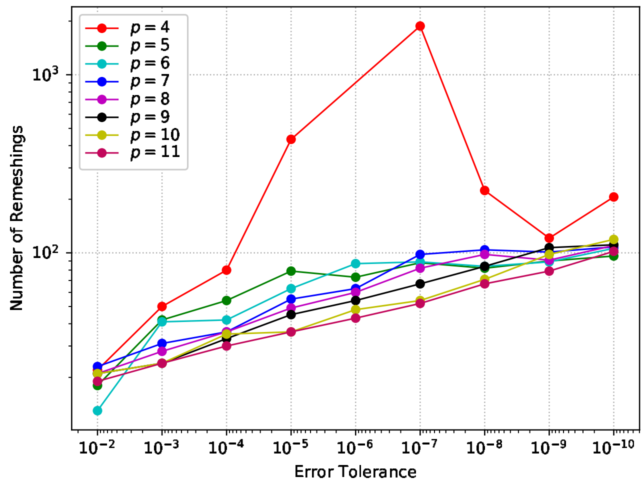

Figure 93: BACOLI/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

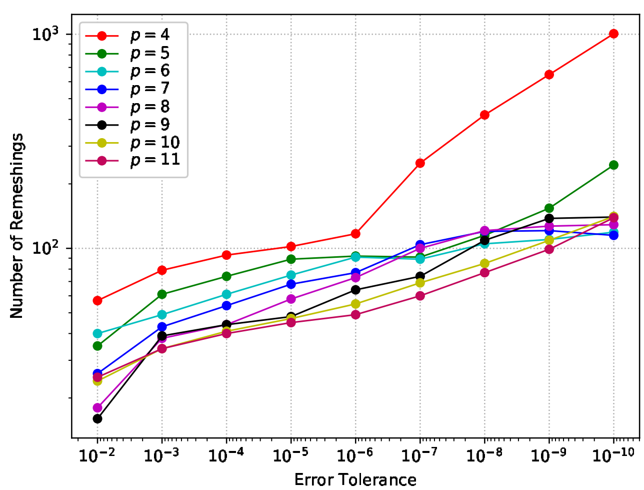

Figure 94: BACOLI/LE Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

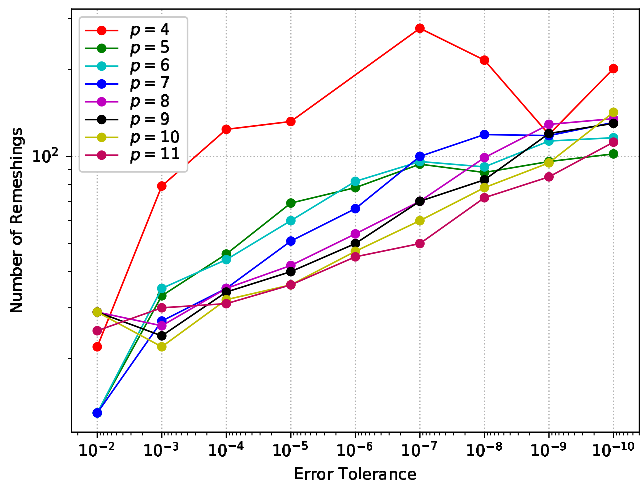

Figure 95: BACOLRI/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

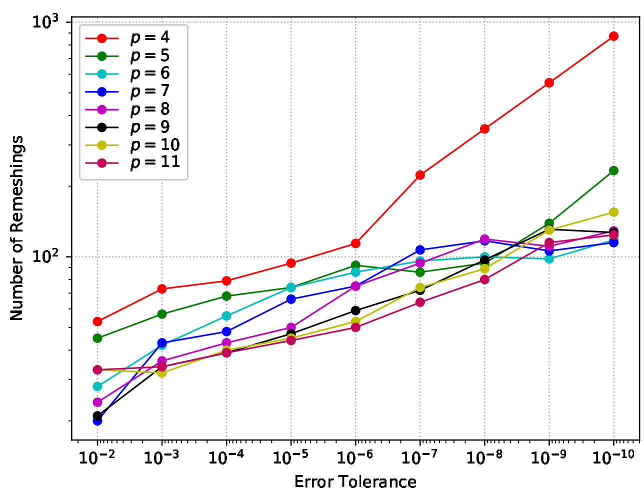

Figure 96: BACOLRI/LE Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

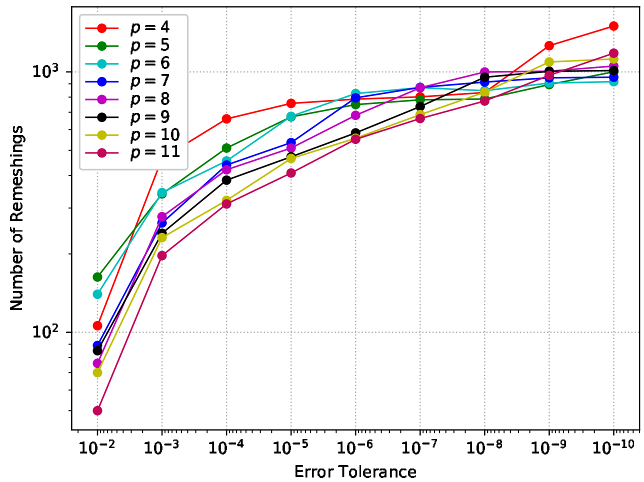

Figure 97: BACOLI/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

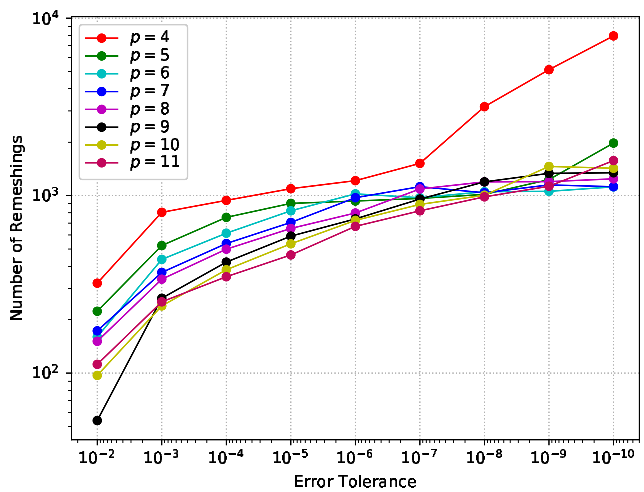

Figure 98: BACOLI/LE Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

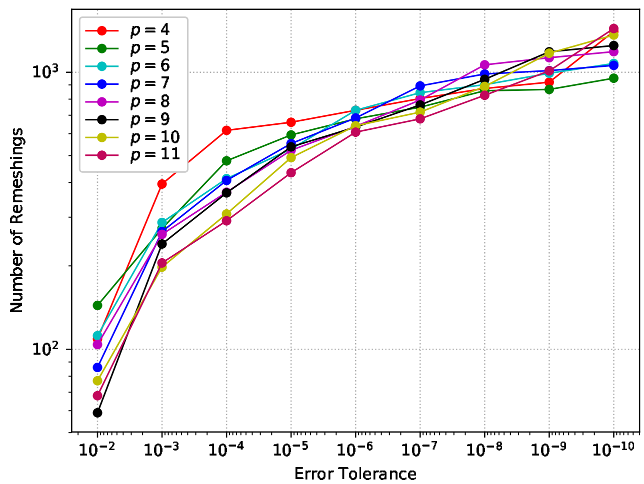

Figure 99: BACOLRI/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

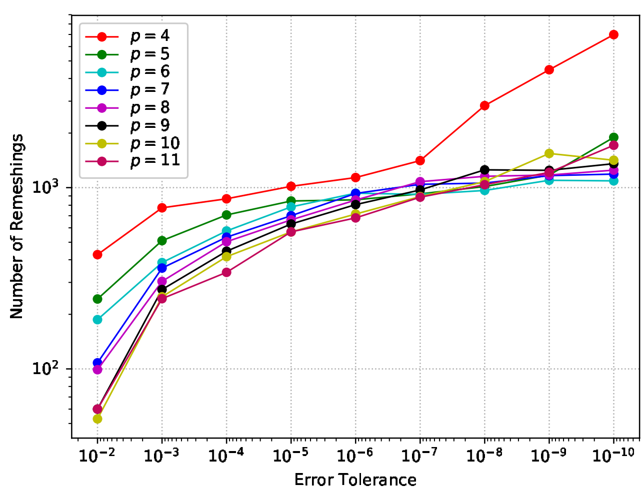

Figure 100: BACOLRI/LE Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

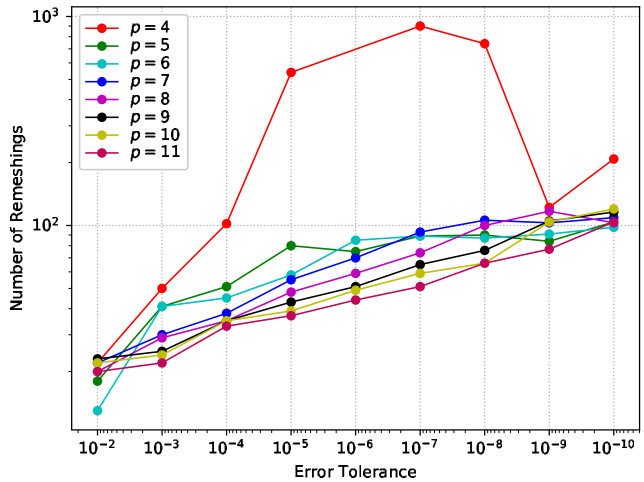

Figure 101: BACOLI/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 4 \dots 11$

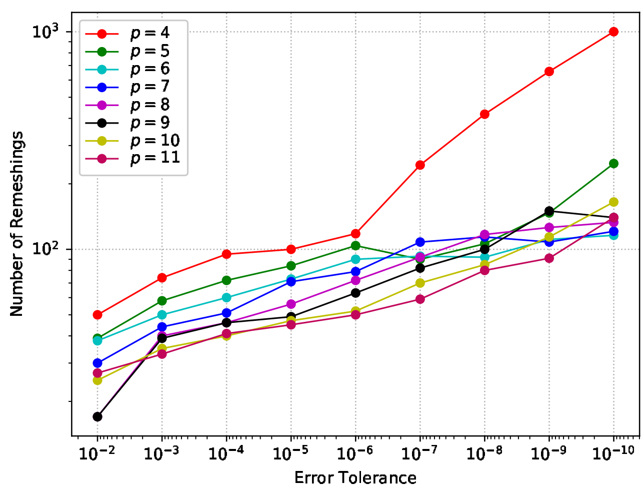

Figure 102: BACOLI/LE Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 4 \dots 11$

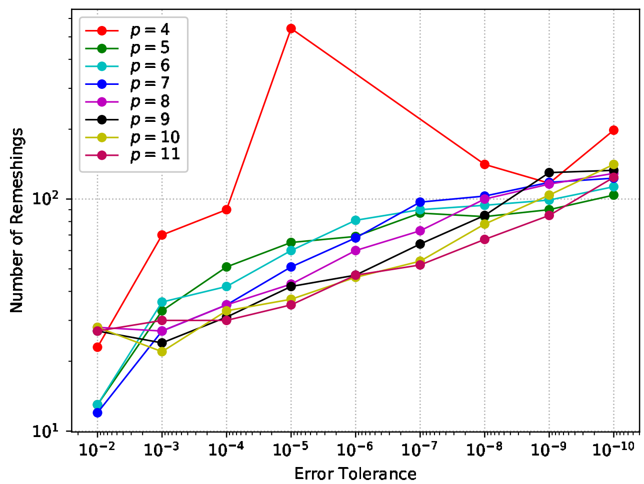

Figure 103: BACOLRI/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

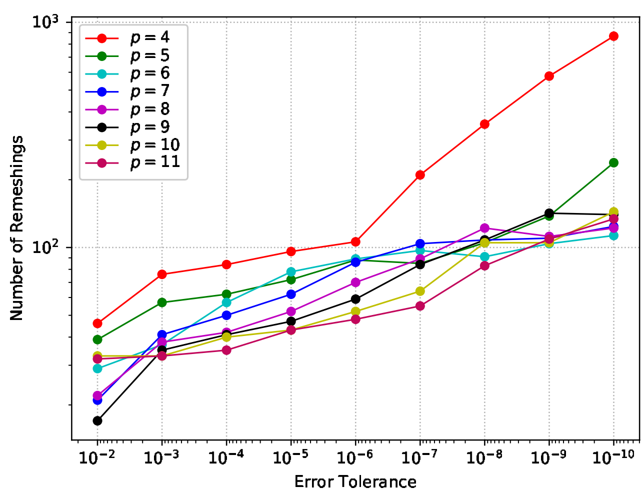

Figure 104: BACOLRI/LE Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

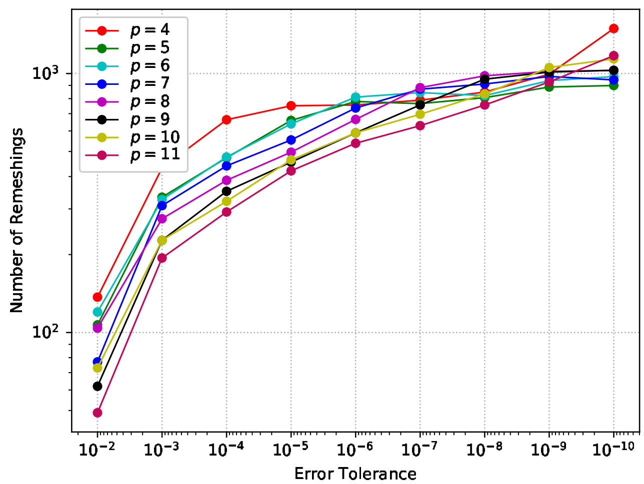

Figure 105: BACOLI/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

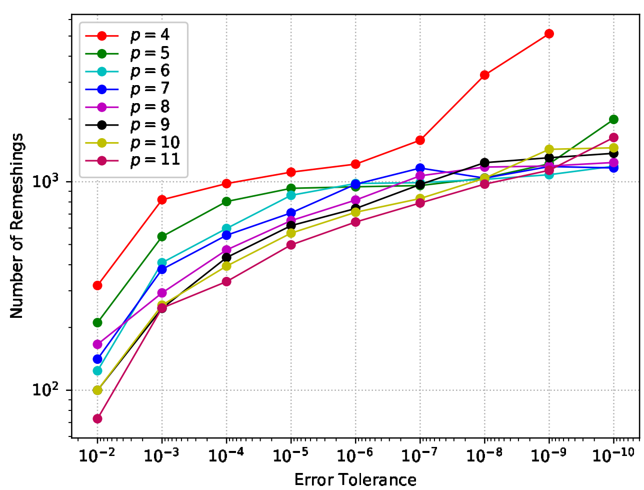

Figure 106: BACOLI/LE Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

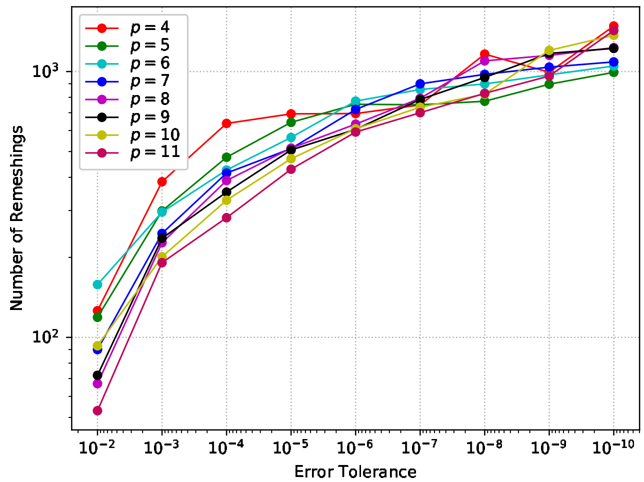

Figure 107: BACOLRI/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

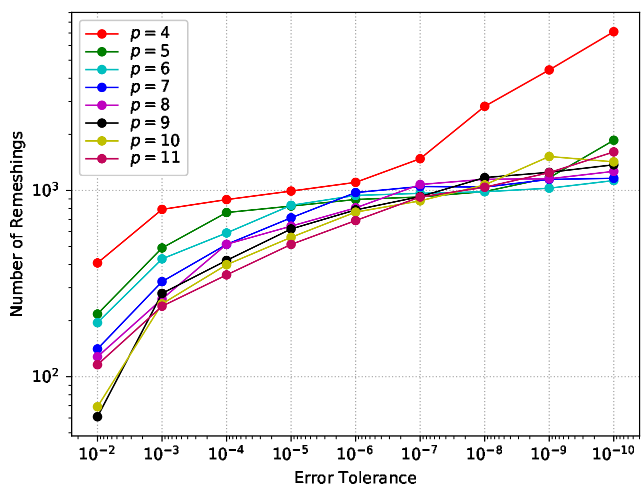

Figure 108: BACOLRI/LE Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

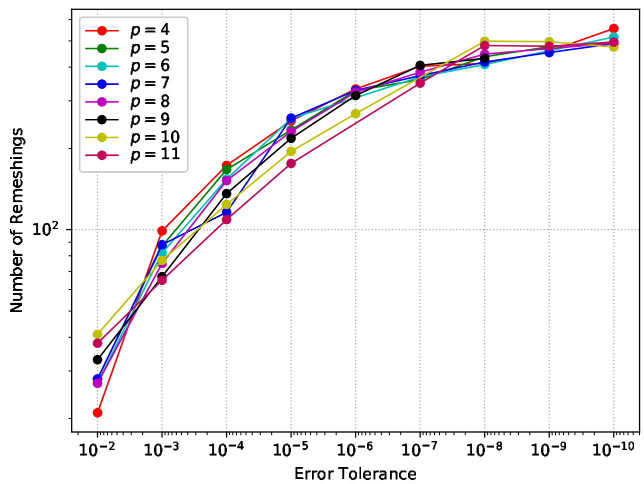

Figure 109: BACOLI/ST Number of Remeshings vs. Error Tolerance: Catalytic Surface Reaction Model with $p = 4 \ldots 11$

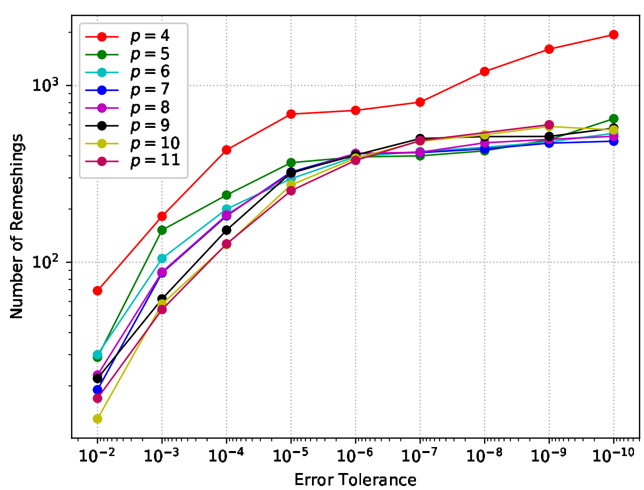

Figure 110: BACOLI/LE Number of Remeshings vs. Error Tolerance: Catalytic Surface Reaction Model with $p = 4 \ldots 11$

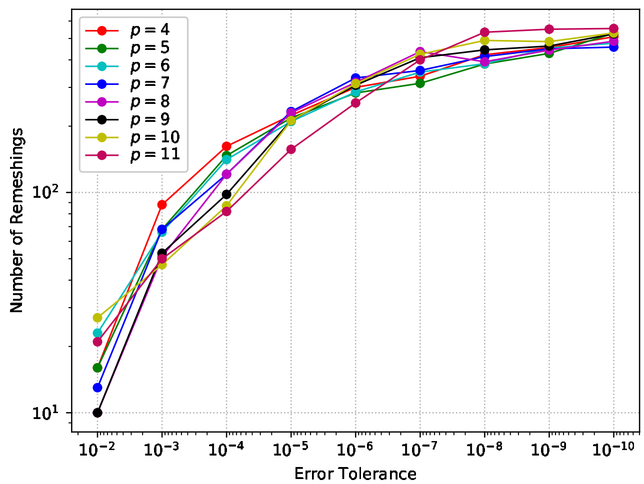

Figure 111: BACOLRI/ST Number of Remeshings vs. Error Tolerance: Catalytic Surface Reaction Model with $p = 4 \ldots 11$

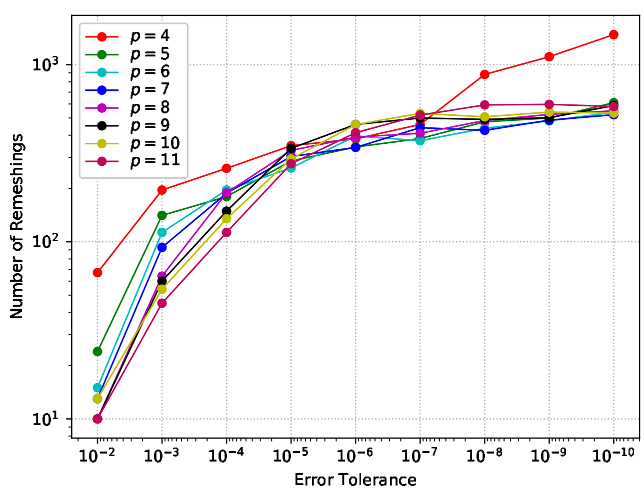

Figure 112: BACOLRI/LE Number of Remeshings vs. Error Tolerance: Catalytic Surface Reaction Model with $p = 4 \ldots 11$

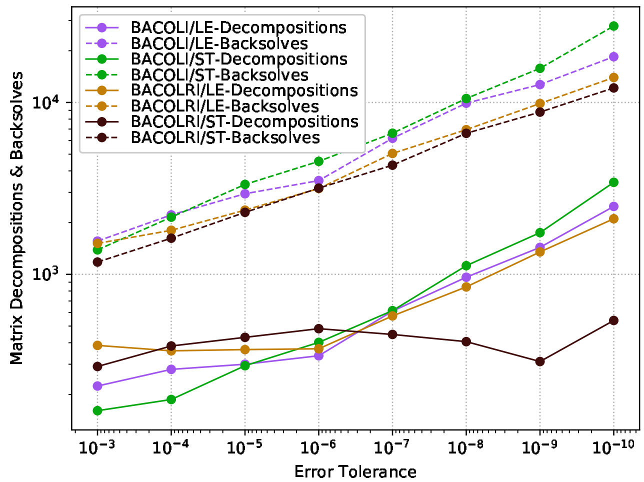

Figure 113: Number of Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4$

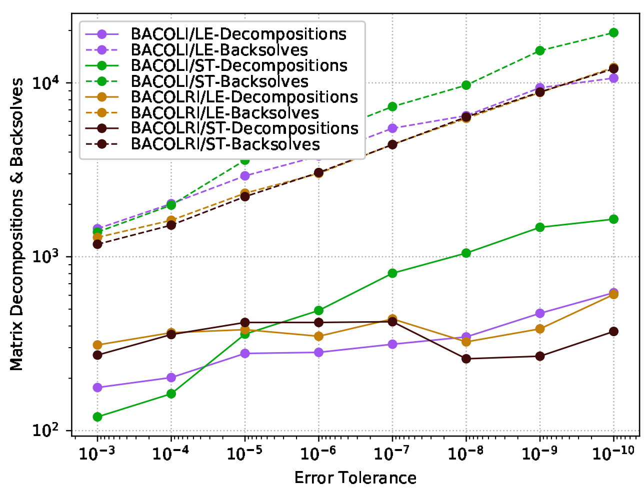

Figure 114: Number of Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 5$

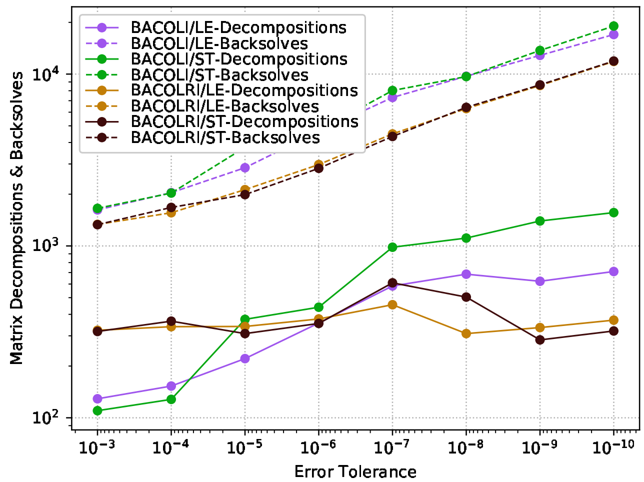

Figure 115: Number of Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 7$

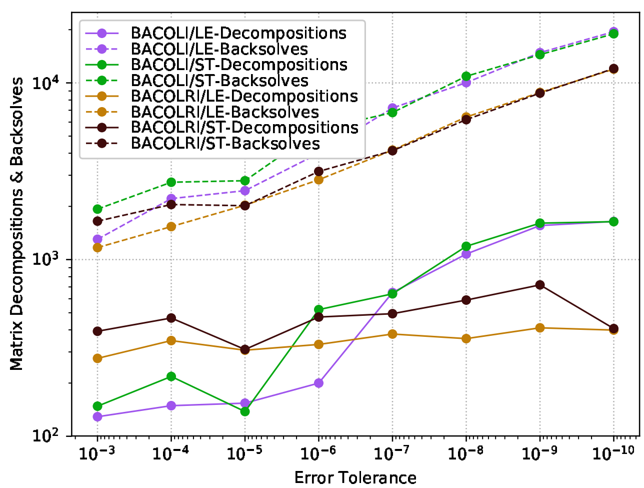

Figure 116: Number of Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 9$

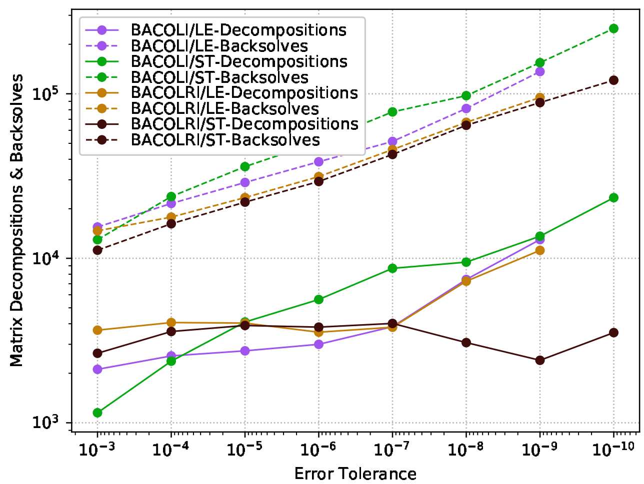

Figure 117: Number of Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4$

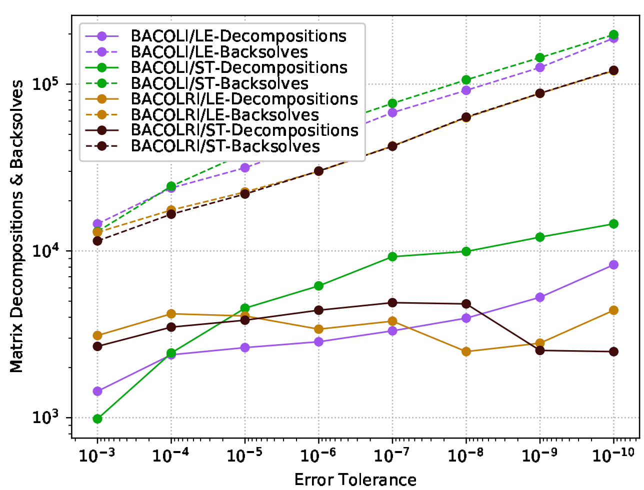

Figure 118: Number of Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 5$

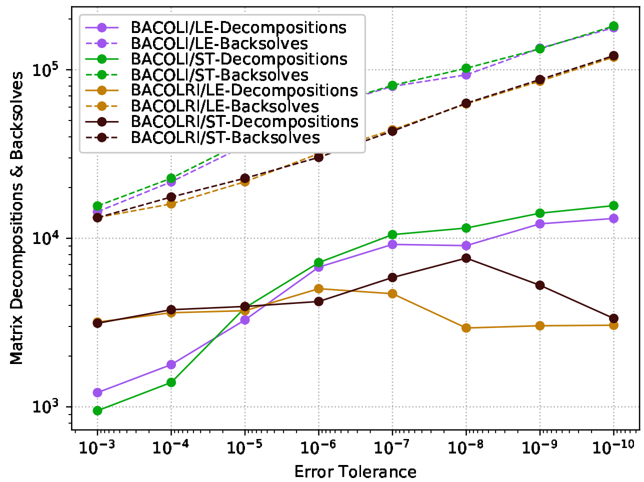

Figure 119: Number of Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 7$

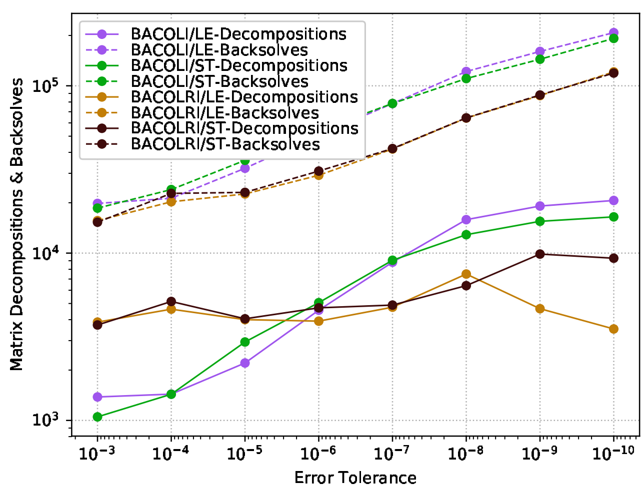

Figure 120: Number of Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 9$

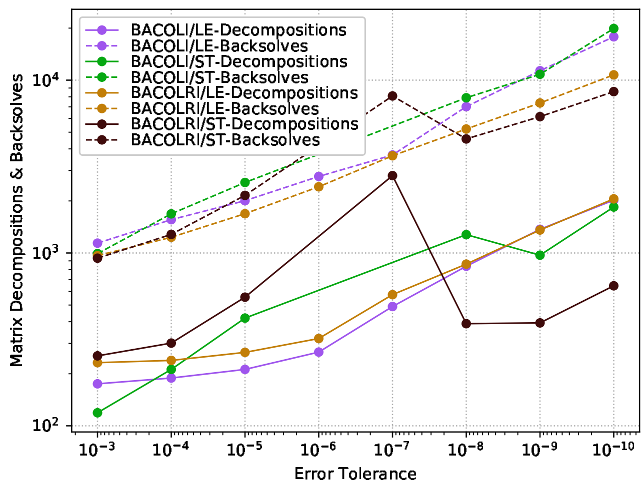

Figure 121: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4$

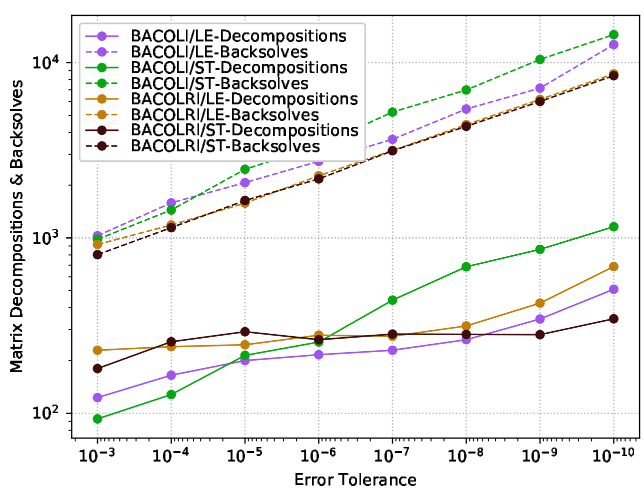

Figure 122: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 5$

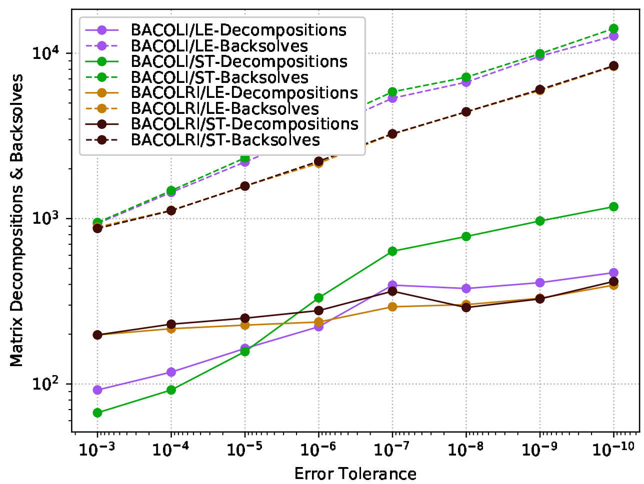

Figure 123: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 7$

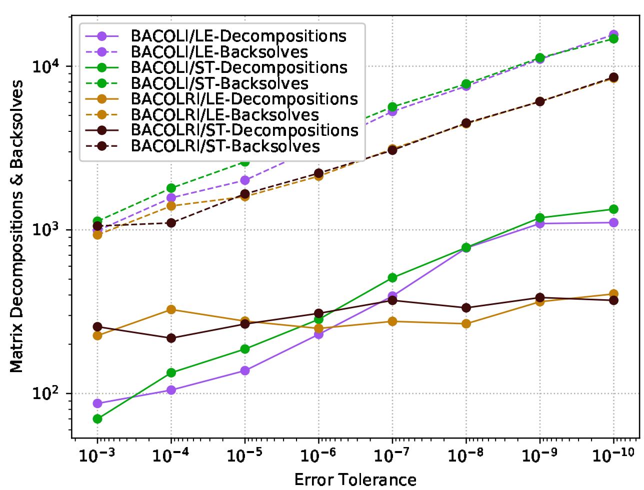

Figure 124: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 9$

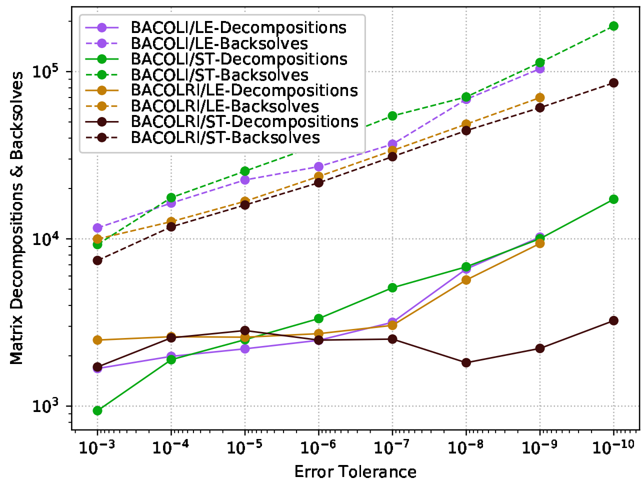

Figure 125: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4$

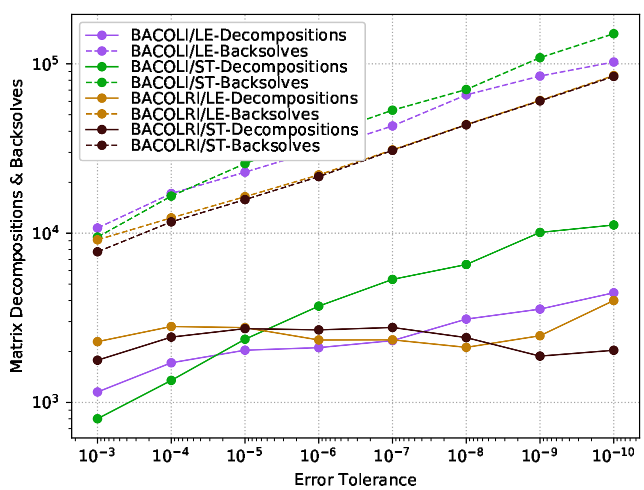

Figure 126: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 5$

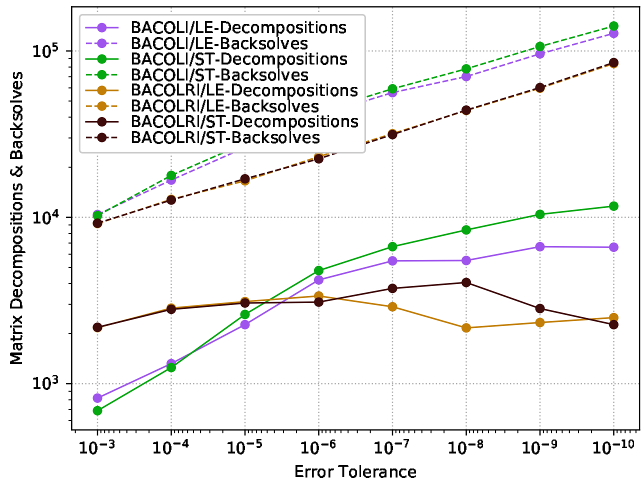

Figure 127: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 7$

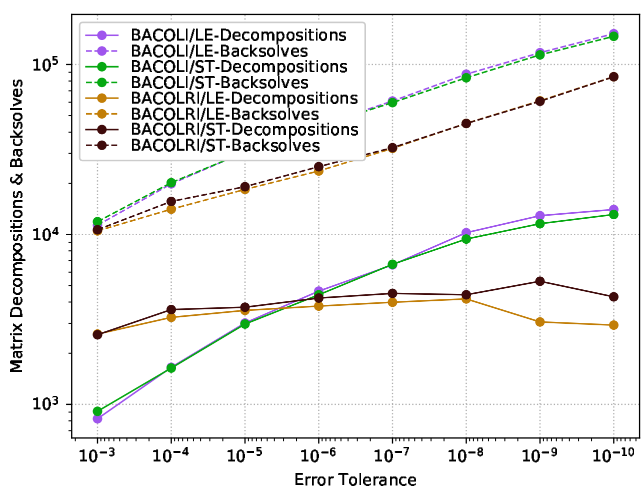

Figure 128: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 9$

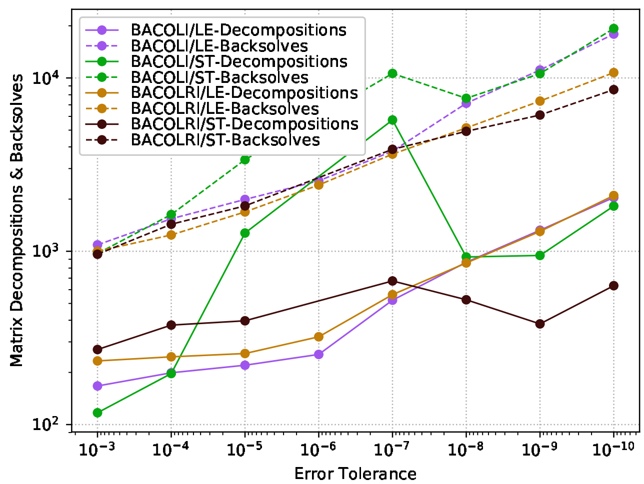

Figure 129: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 4$

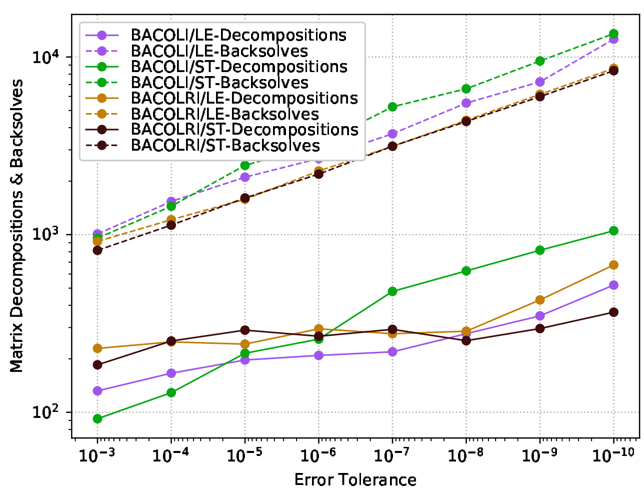

Figure 130: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 5$

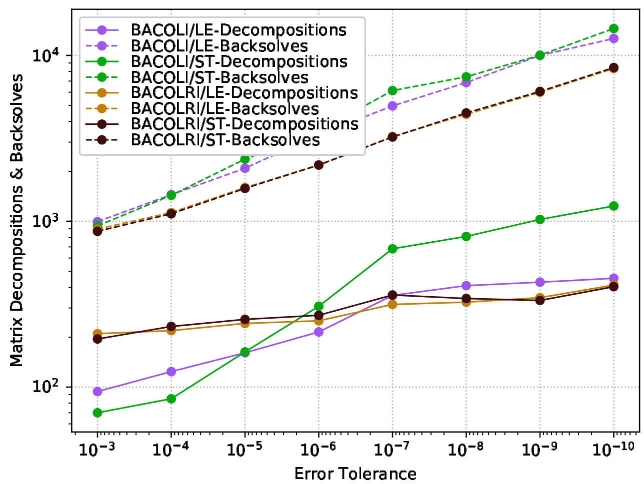

Figure 131: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 7$

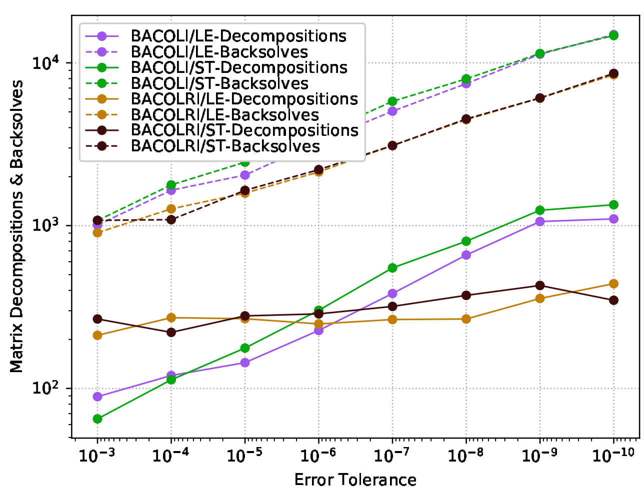

Figure 132: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 9$

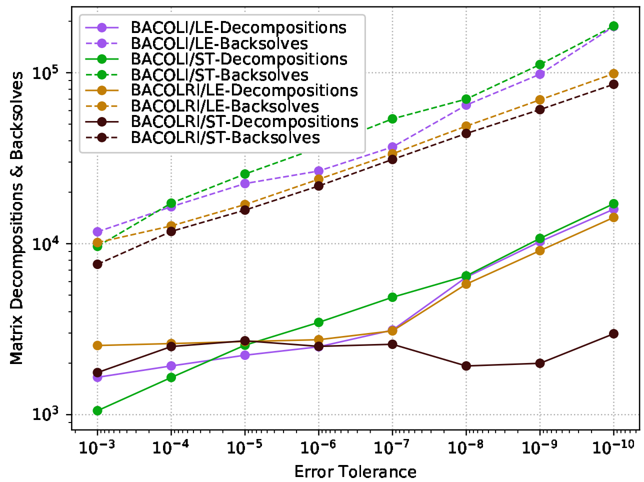

Figure 133: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 4$

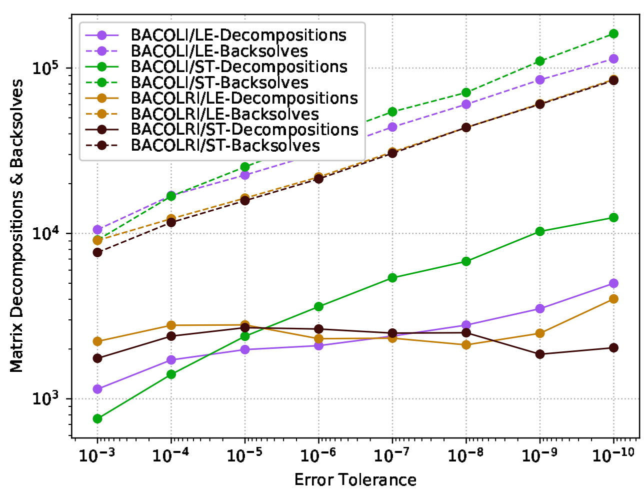

Figure 134: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 5$

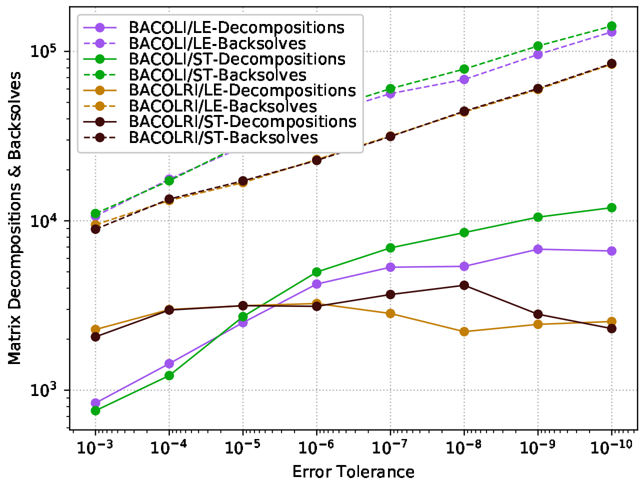

Figure 135: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 7$

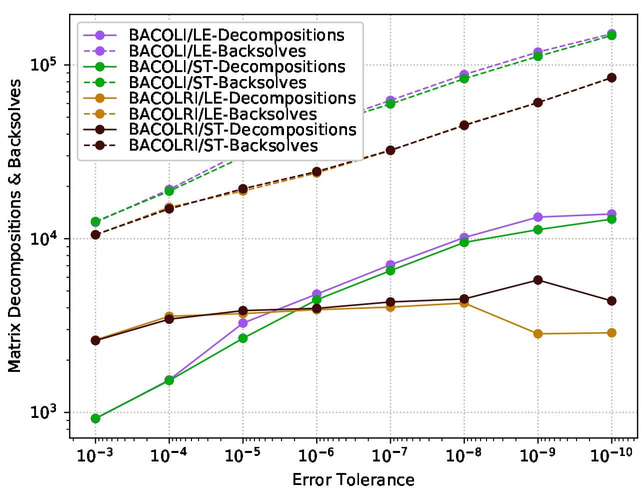

Figure 136: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 9$

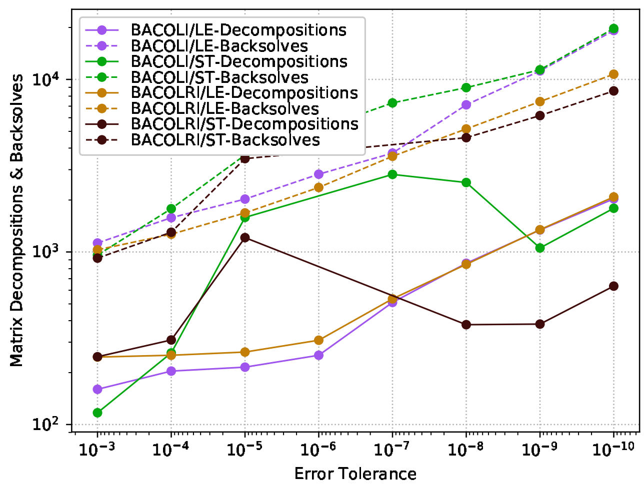

Figure 137: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 4$

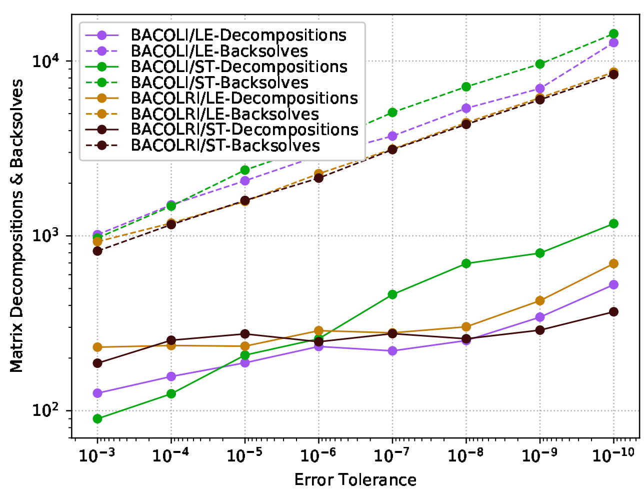

Figure 138: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 5$

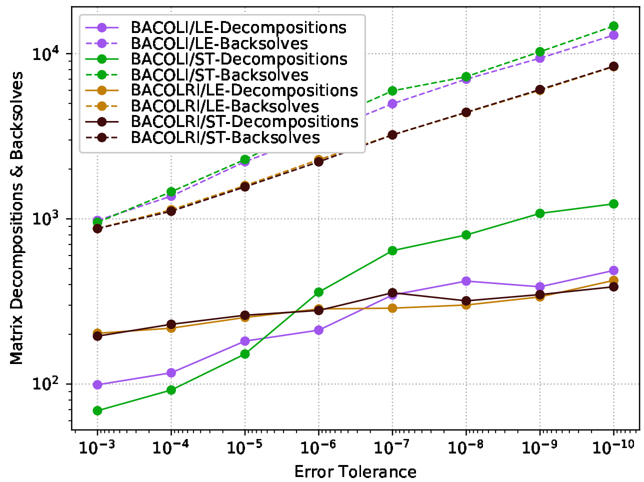

Figure 139: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 7$

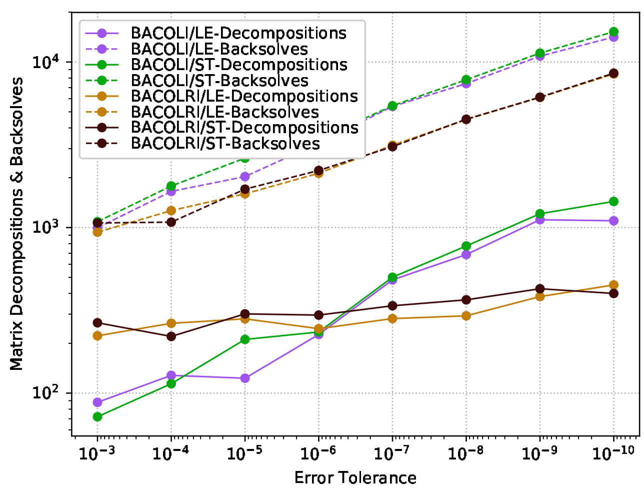

Figure 140: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 9$

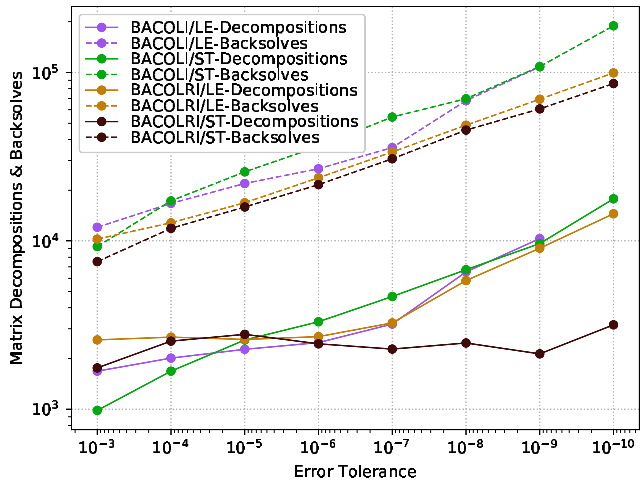

Figure 141: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 4$

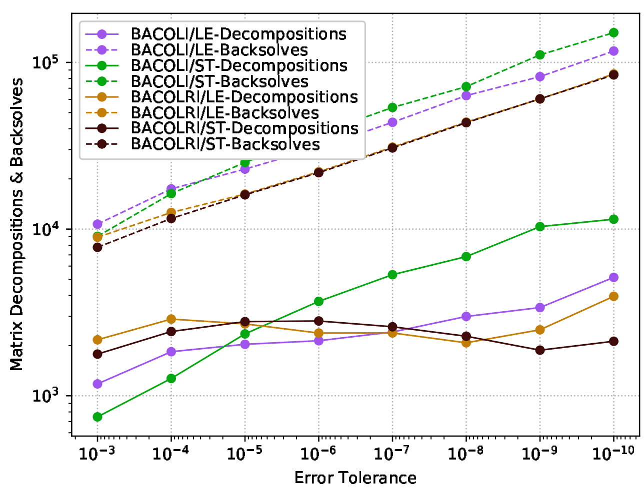

Figure 142: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 5$

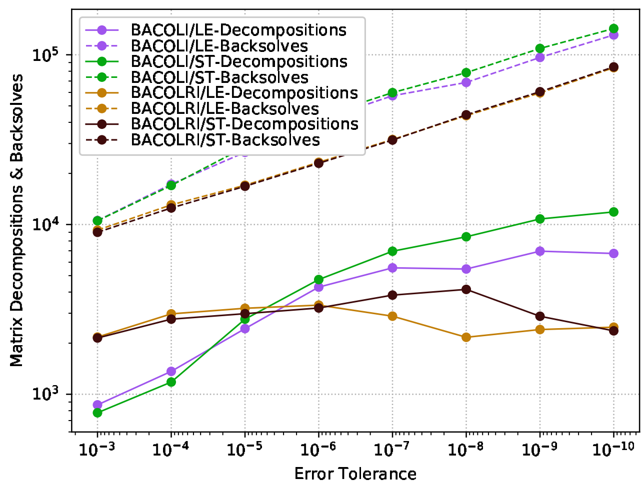

Figure 143: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 7$

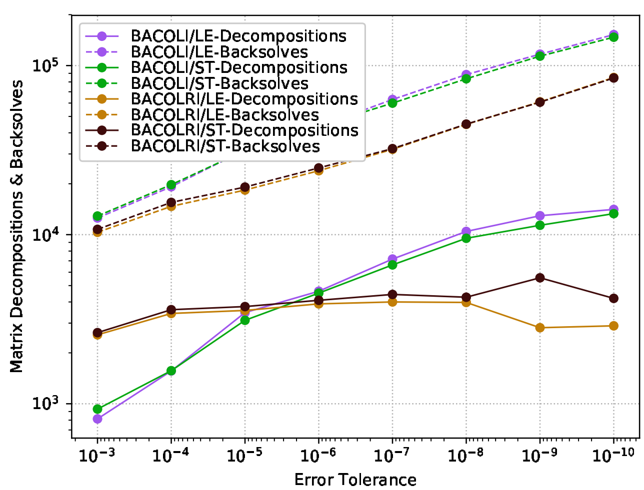

Figure 144: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 9$

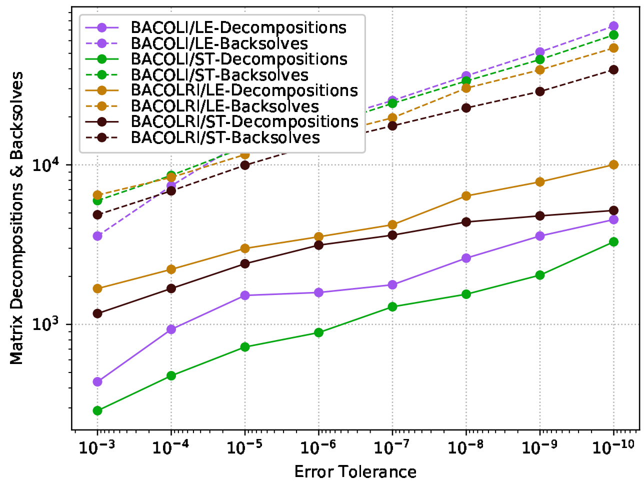

Figure 145: Number of Matrix Factorizations and Backsolves vs. Tolerance: Catalytic Surface Reaction Model with $p = 4$

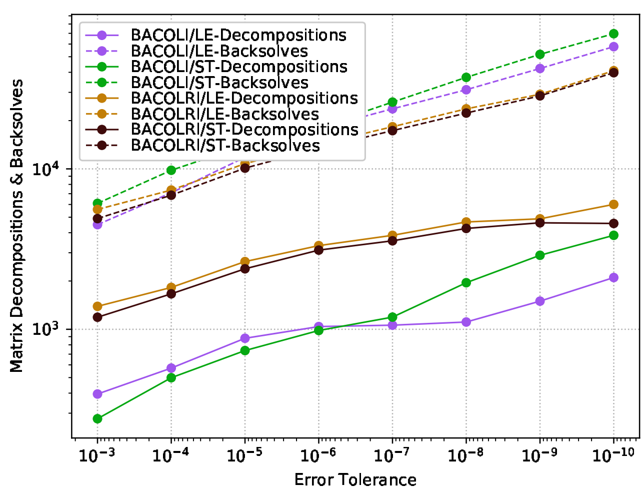

Figure 146: Number of Matrix Factorizations and Backsolves vs. Tolerance: Catalytic Surface Reaction Model with $p = 5$

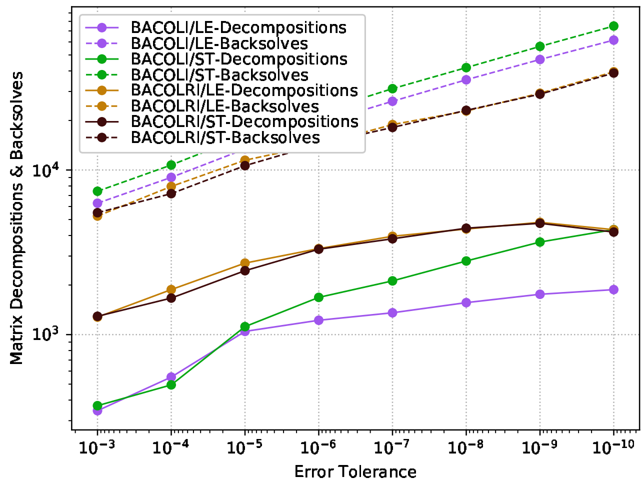

Figure 147: Number of Matrix Factorizations and Backsolves vs. Tolerance: Catalytic Surface Reaction Model with $p = 7$

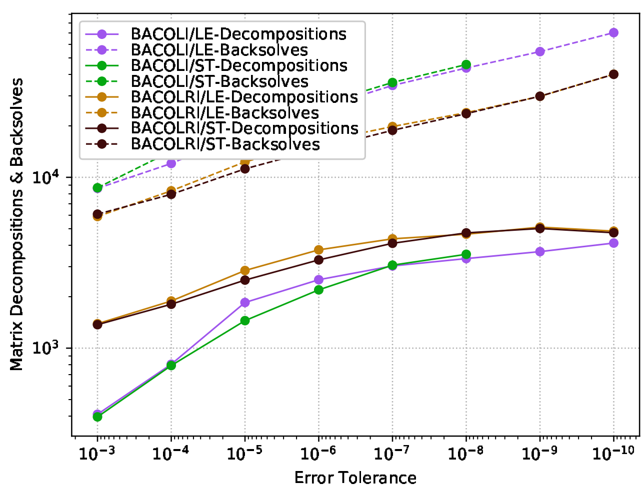

Figure 148: Number of Matrix Factorizations and Backsolves vs. Tolerance: Catalytic Surface Reaction Model with $p = 9$

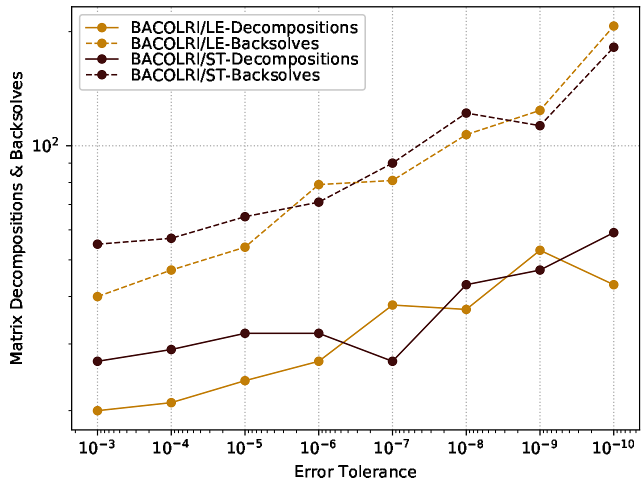

Figure 149: Number of Matrix Factorizations and Backsolves vs. Tolerance: Schrödinger System with $p = 4$

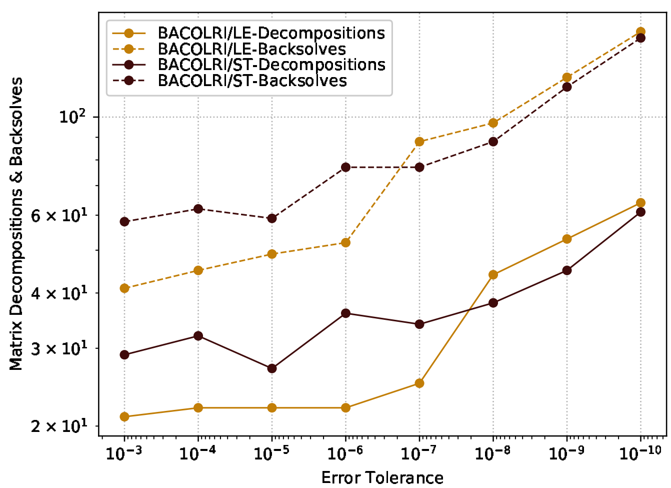

Figure 150: Number of Matrix Factorizations and Backsolves vs. Tolerance: Schrödinger System with $p = 5$

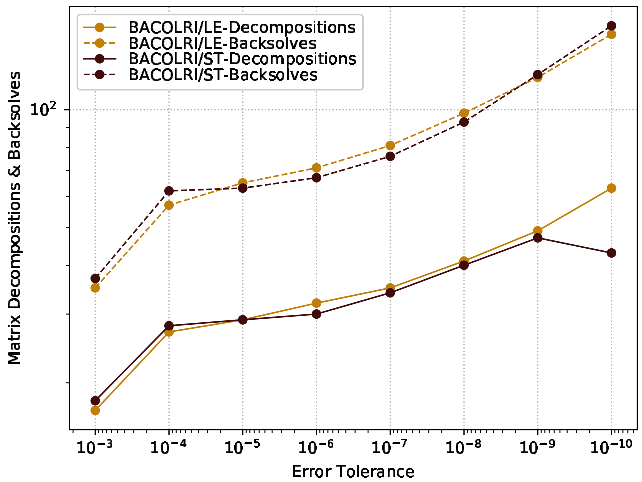

Figure 151: Number of Matrix Factorizations and Backsolves vs. Tolerance: Schrödinger System with $p = 7$

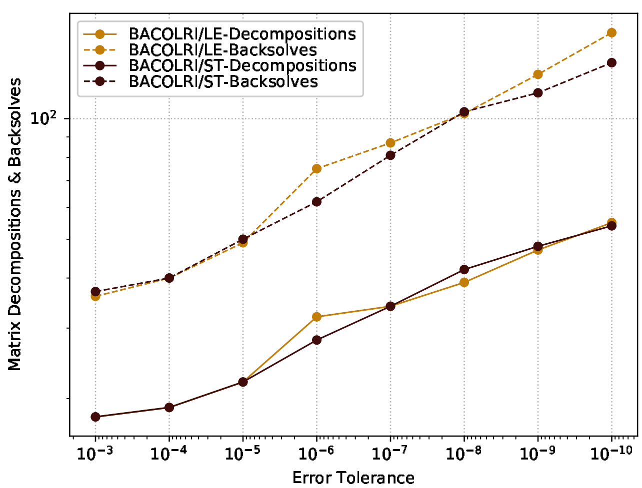

Figure 152: Number of Matrix Factorizations and Backsolves vs. Tolerance: Schrödinger System with $p = 9$

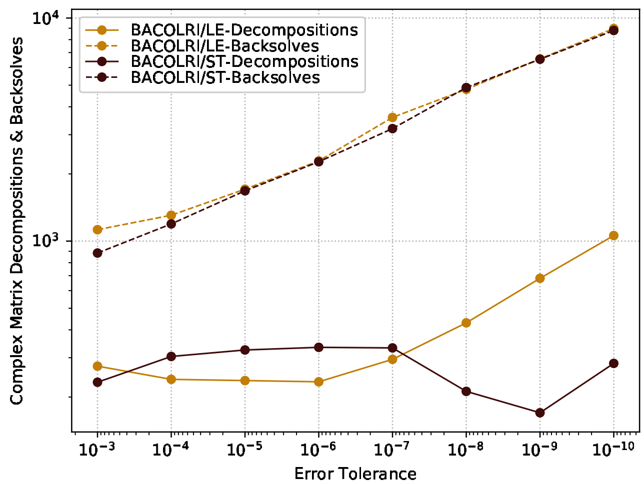

Figure 153: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4$

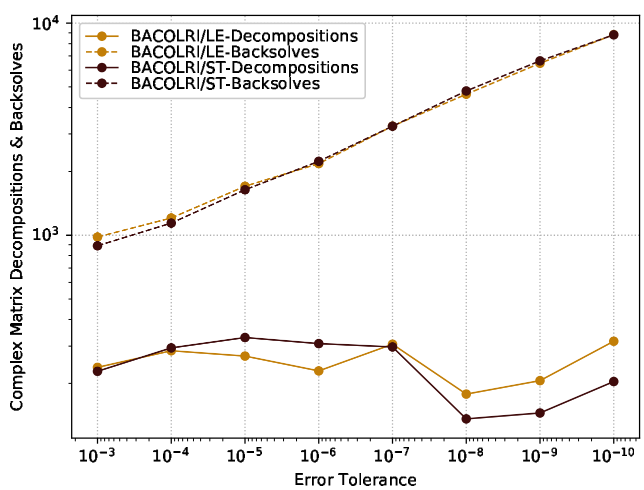

Figure 154: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 5$

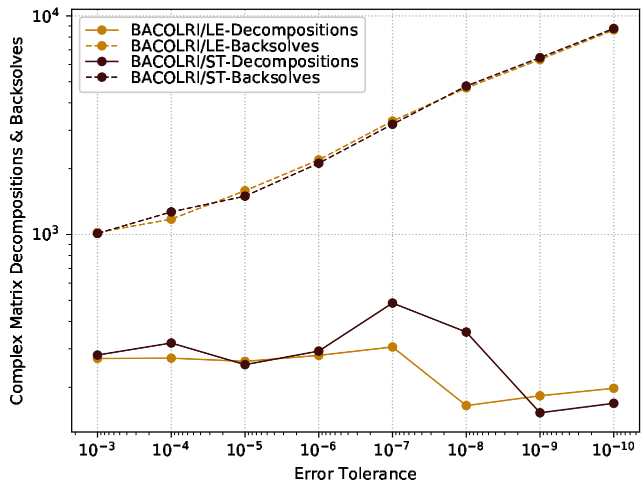

Figure 155: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 7$

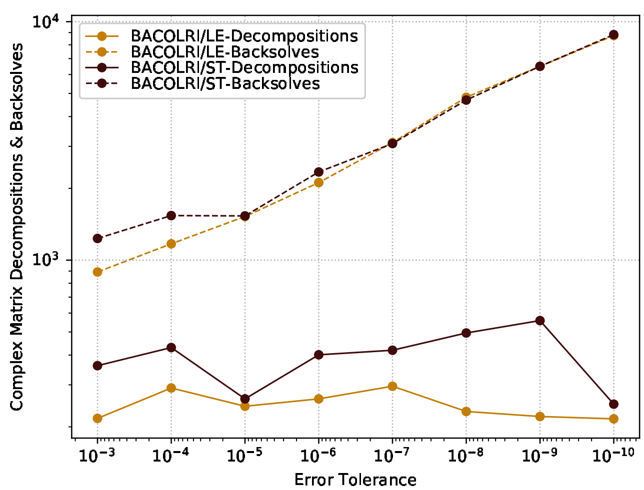

Figure 156: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 9$

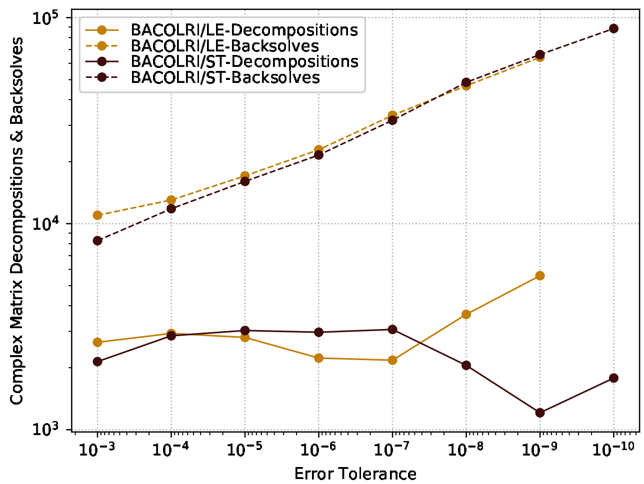

Figure 157: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4$

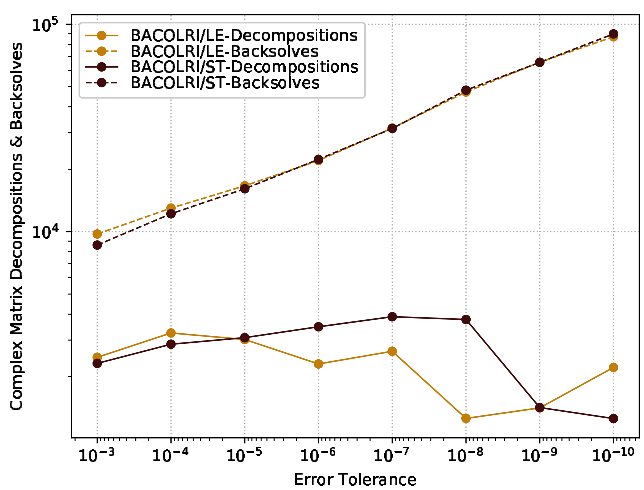

Figure 158: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 5$

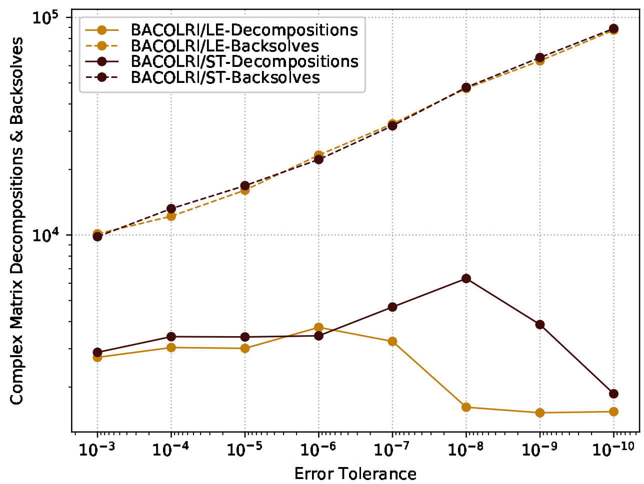

Figure 159: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 7$

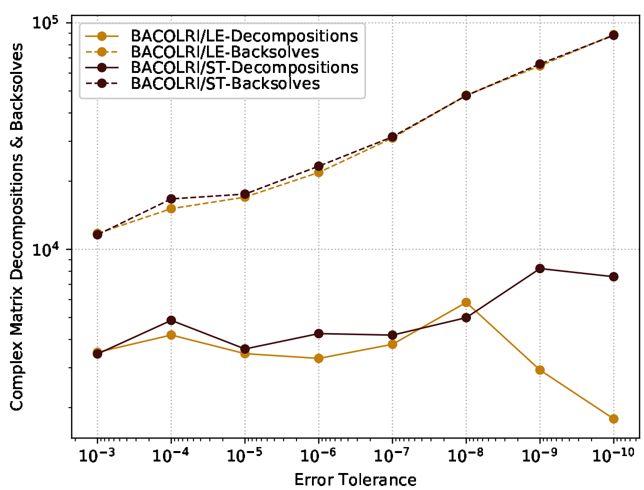

Figure 160: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 9$

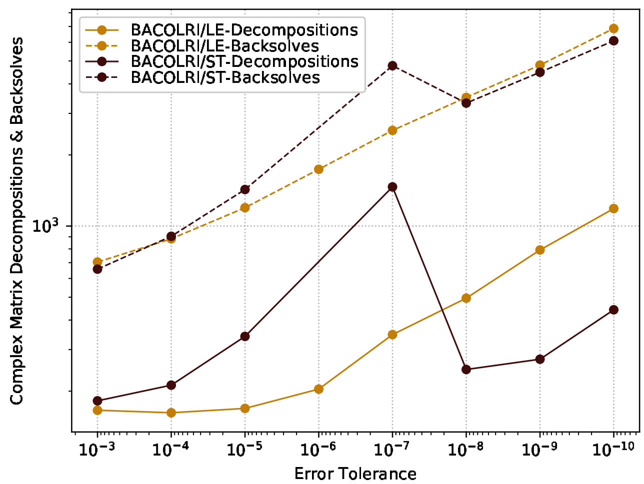

Figure 161: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4$

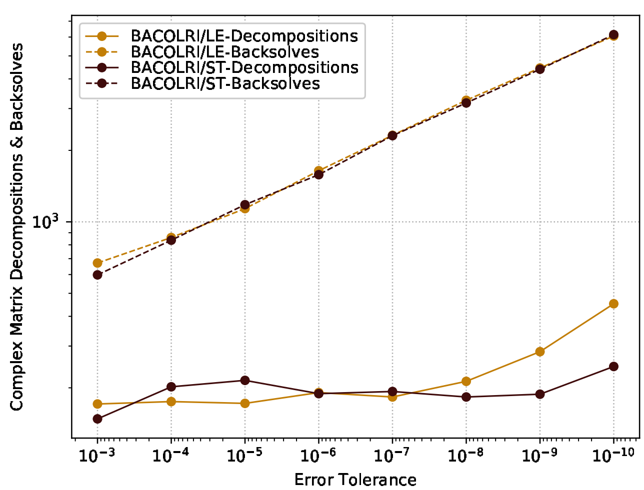

Figure 162: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 5$

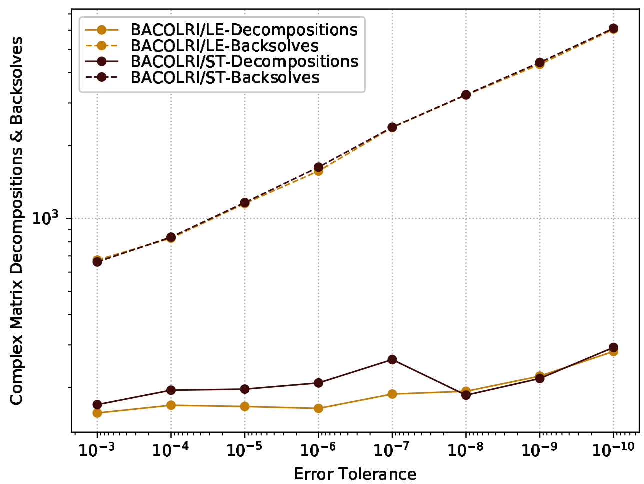

Figure 163: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 7$

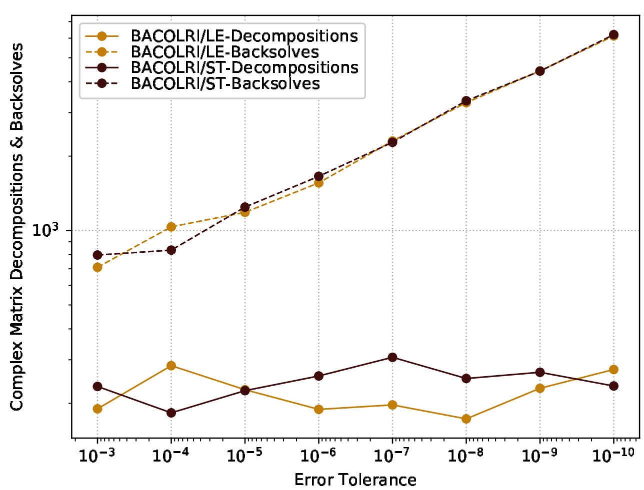

Figure 164: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 9$

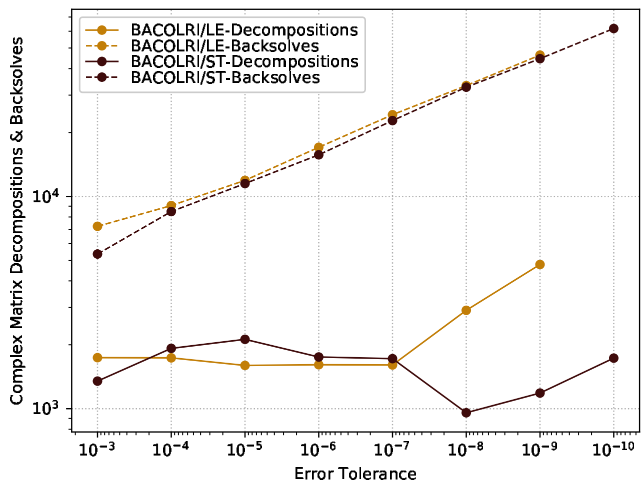

Figure 165: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4$

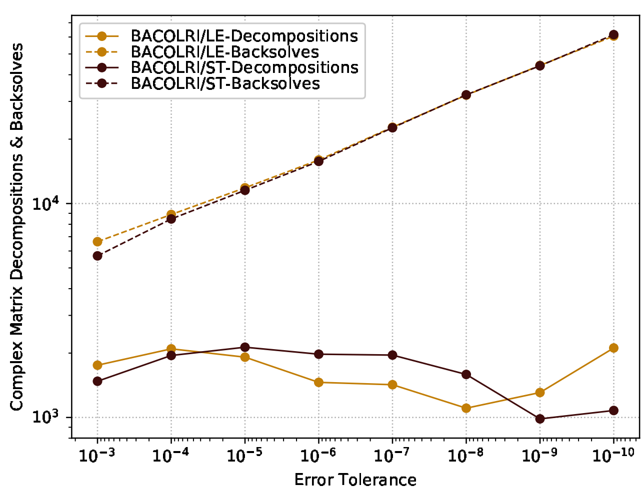

Figure 166: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 5$

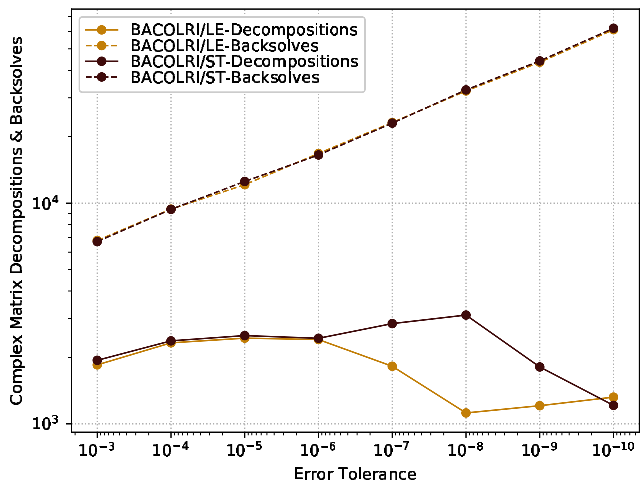

Figure 167: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 7$

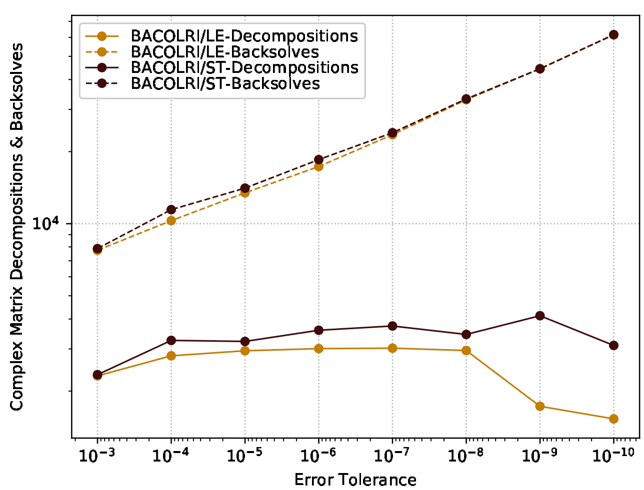

Figure 168: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 9$

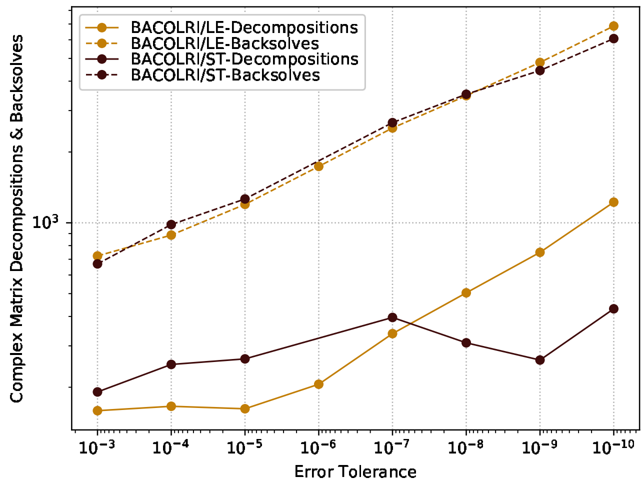

Figure 169: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 4$

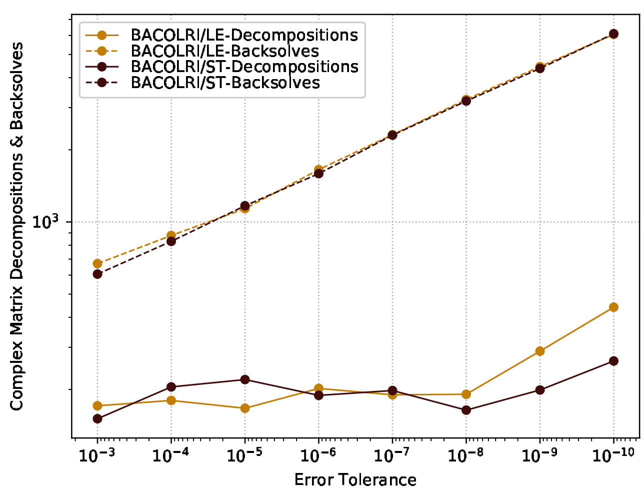

Figure 170: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 5$

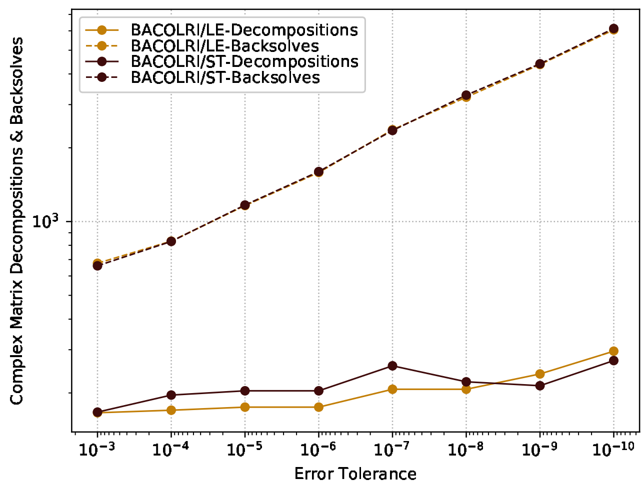

Figure 171: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 7$

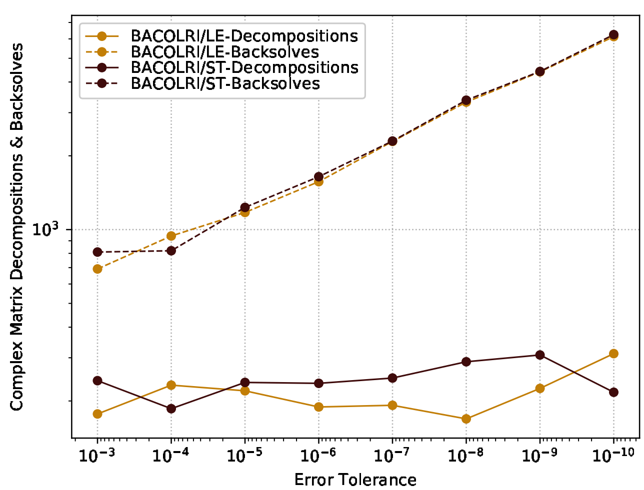

Figure 172: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 9$

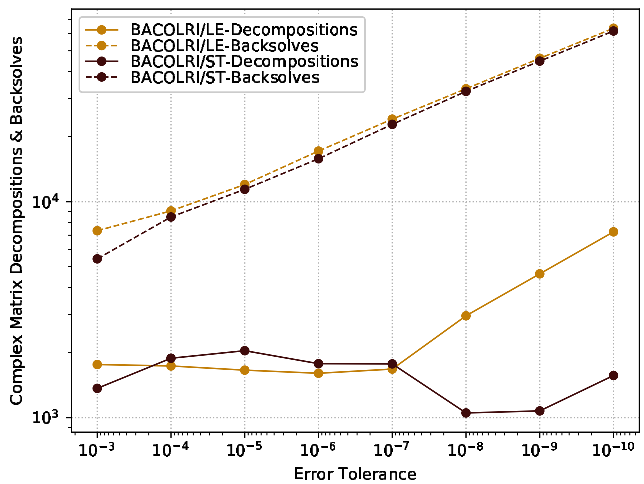

Figure 173: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 4$

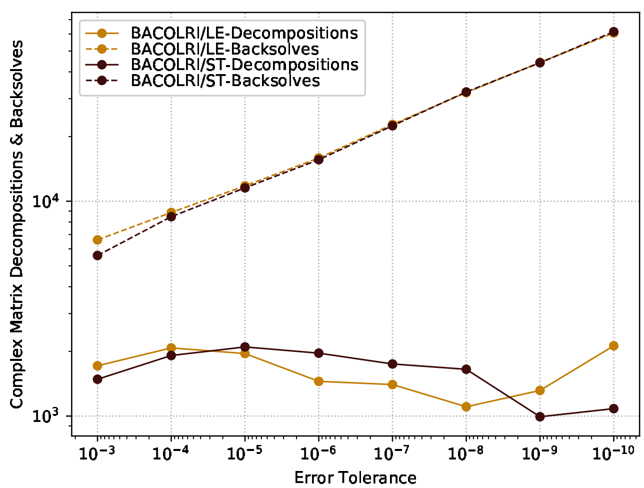

Figure 174: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 5$

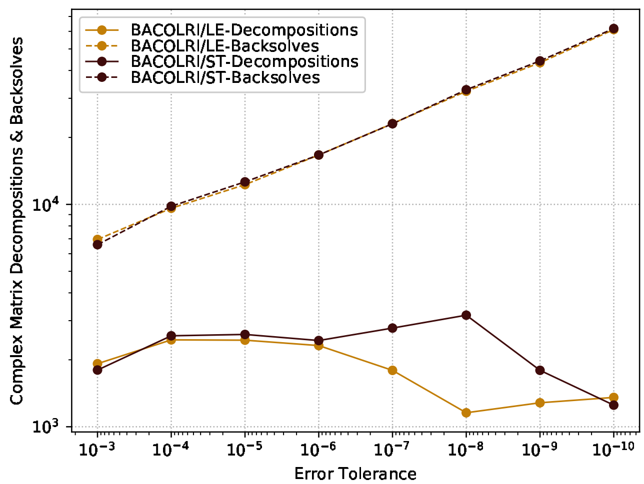

Figure 175: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 7$

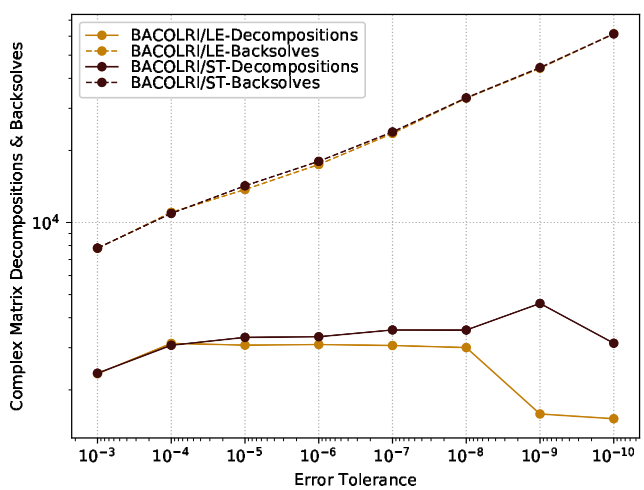

Figure 176: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 9$

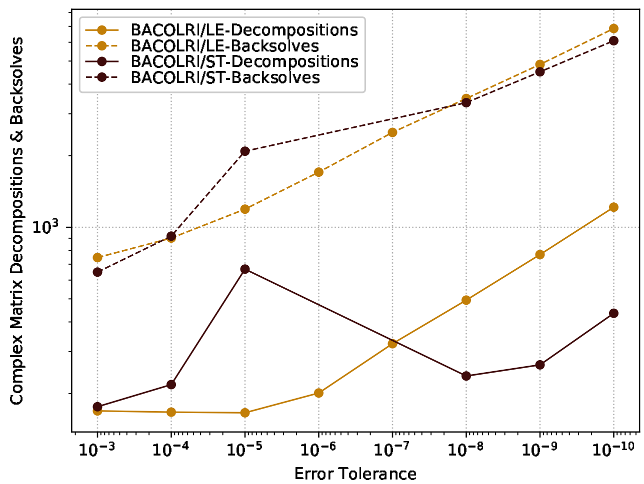

Figure 177: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 4$

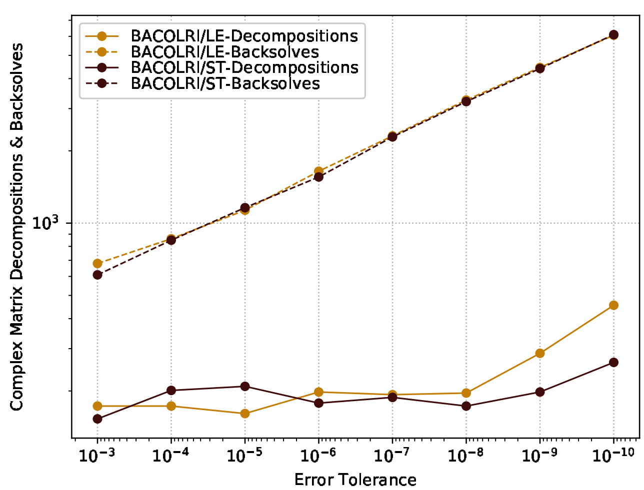

Figure 178: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 5$

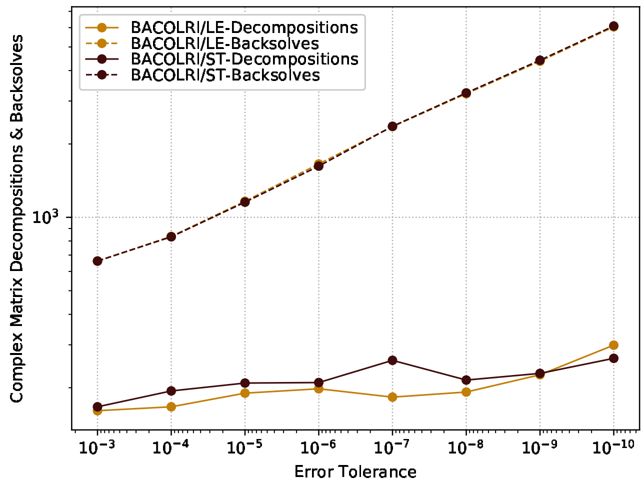

Figure 179: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 7$

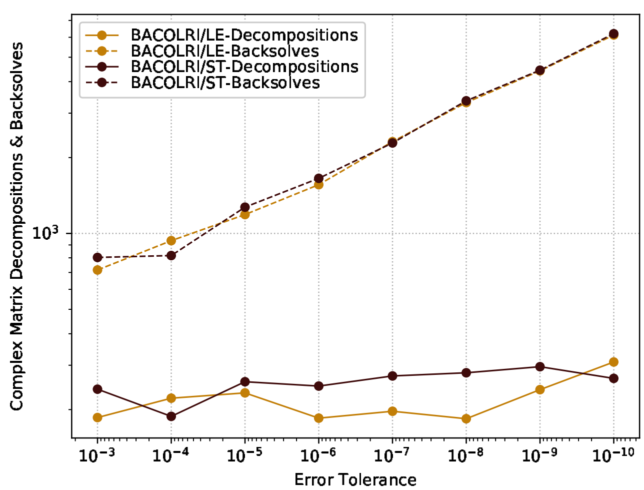

Figure 180: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 9$

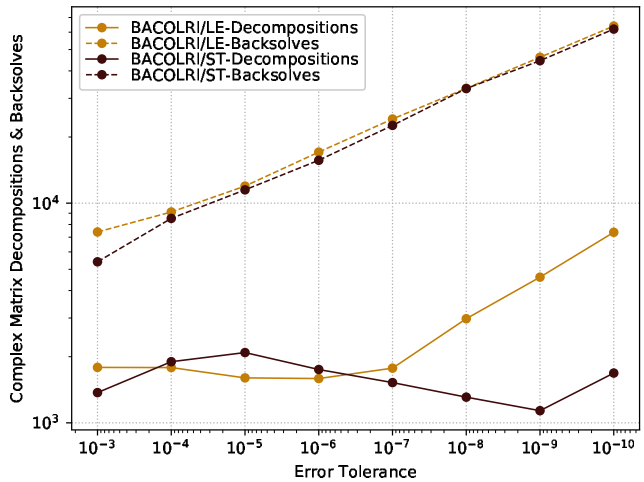

Figure 181: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 4$

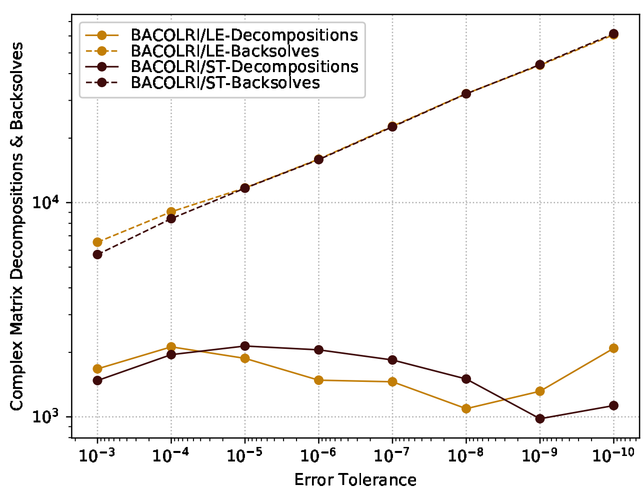

Figure 182: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 5$

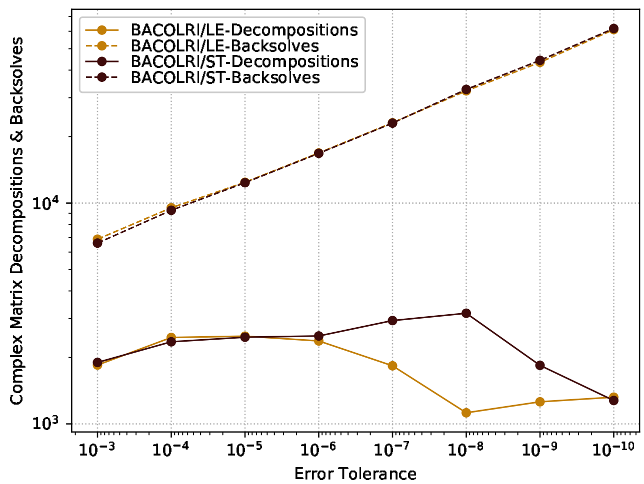

Figure 183: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 7$

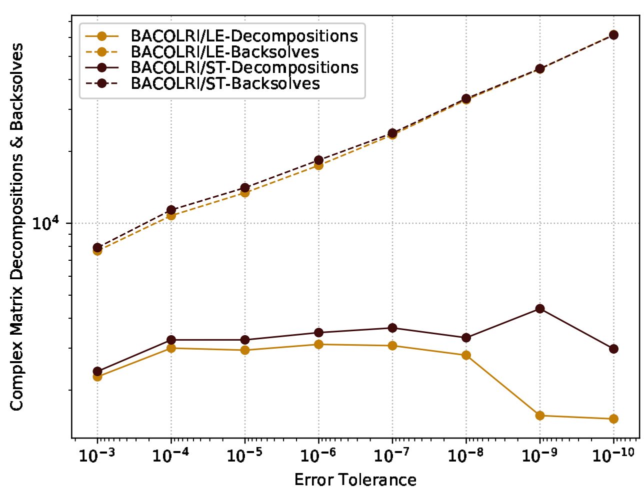

Figure 184: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 9$

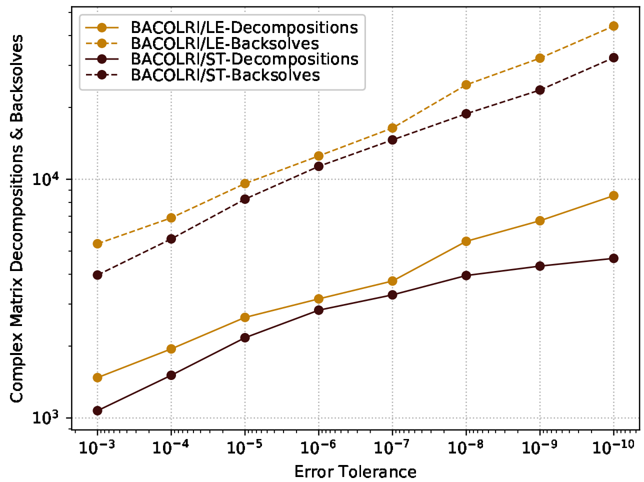

Figure 185: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Catalytic Surface Reaction Model with $p = 4$

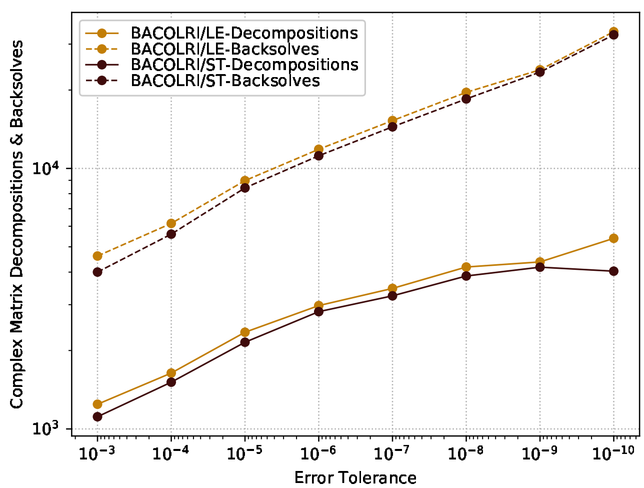

Figure 186: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Catalytic Surface Reaction Model with $p = 5$

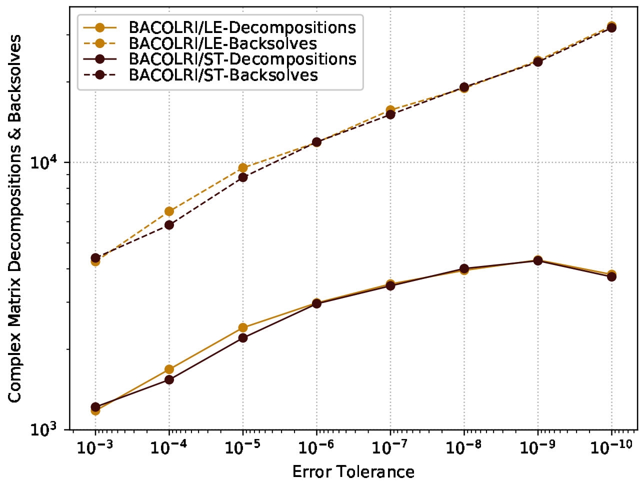

Figure 187: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Catalytic Surface Reaction Model with $p = 7$

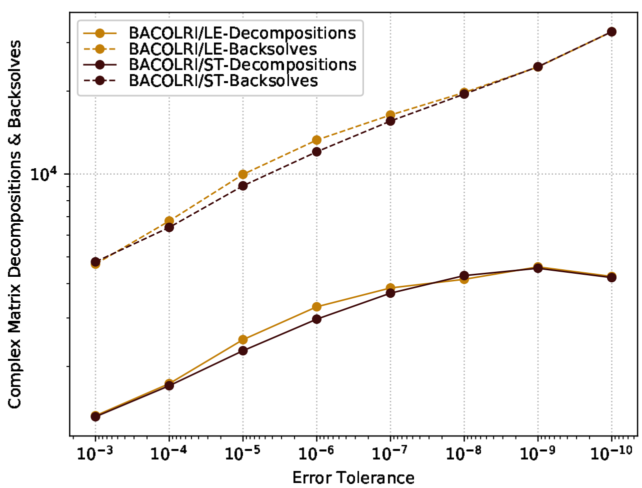

Figure 188: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Catalytic Surface Reaction Model with $p = 9$

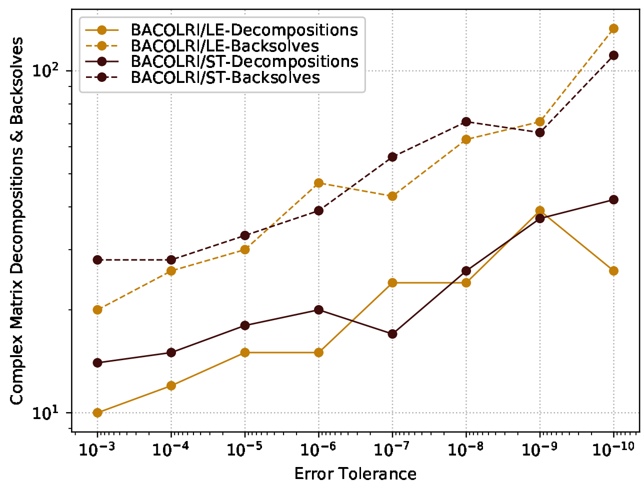

Figure 189: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Schrödinger System with $p = 4$

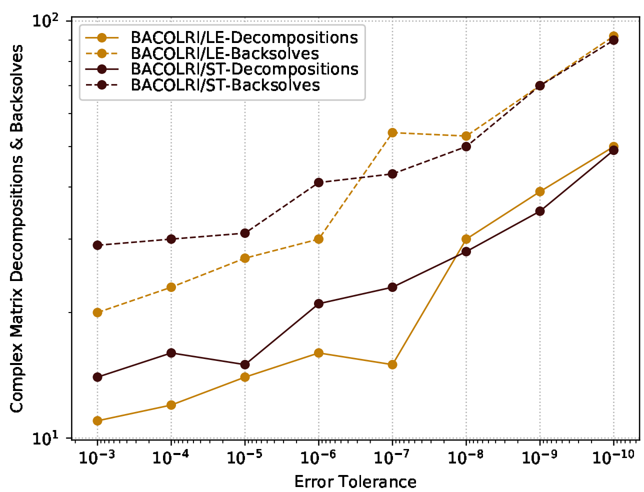

Figure 190: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Schrödinger System with $p = 5$

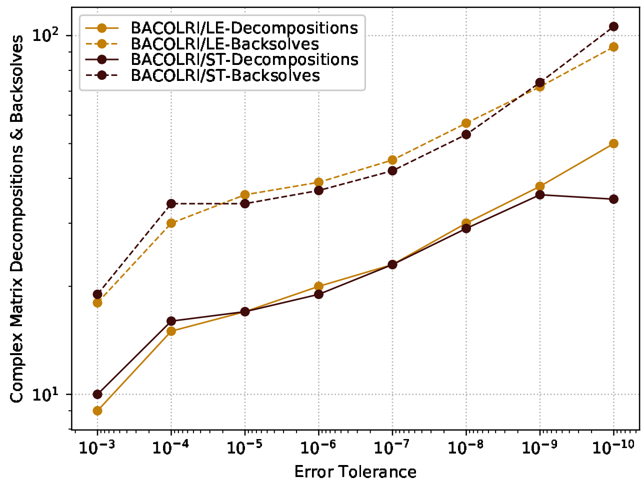

Figure 191: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Schrödinger System with $p = 7$

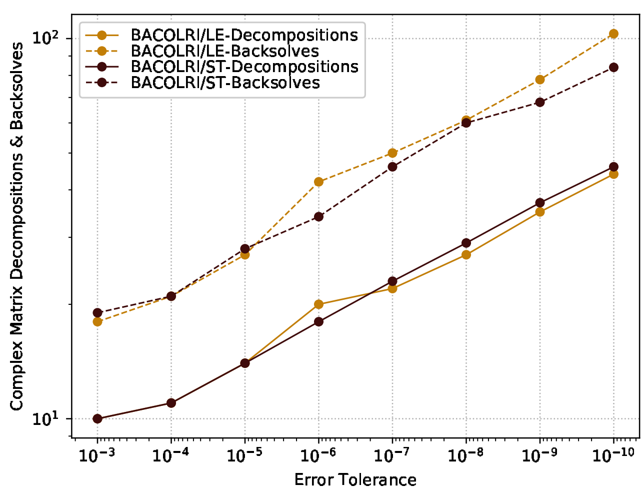

Figure 192: Number of Complex Matrix Factorizations and Backsolves vs. Tolerance: Schrödinger System with $p = 9$

$tol = 10^{-4}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	0.02	0.03	0.04	0.04	0.05	0.06	0.07	0.08
BACOLI/LE	0.03	0.03	0.04	0.04	0.05	0.06	0.07	0.09
BACOLRI/ST	0.03	0.04	0.04	0.05	0.06	0.07	0.08	0.09
BACOLRI/LE	0.04	0.04	0.04	0.05	0.05	0.06	0.07	0.09
$tol = 10^{-6}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	0.09	0.10	0.10	0.11	0.12	0.14	0.17	0.18
BACOLI/LE	0.11	0.09	0.10	0.10	0.12	0.13	0.15	0.18
BACOLRI/ST	0.07	0.07	0.07	0.08	0.09	0.10	0.13	0.15
BACOLRI/LE	0.10	0.08	0.09	0.09	0.10	0.12	0.12	0.15
$tol = 10^{-8}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	0.38	0.30	0.27	0.28	0.29	0.34	0.32	0.36
BACOLI/LE	0.84	0.34	0.32	0.32	0.30	0.32	0.31	0.36
BACOLRI/ST	0.24	0.20	0.18	0.19	0.20	0.22	0.23	0.24
BACOLRI/LE	0.62	0.28	0.23	0.21	0.22	0.22	0.23	0.26
$tol = 10^{-10}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	2.16	1.10	0.84	0.79	0.76	0.77	0.79	0.88
BACOLI/LE	5.21	1.30	1.07	0.89	0.76	0.82	0.89	0.83
BACOLRI/ST	0.96	0.63	0.53	0.48	0.47	0.48	0.53	0.53
BACOLRI/LE	4.17	1.29	0.77	0.61	0.54	0.51	0.52	0.50

Table 15: *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
BACOLI/ST	0.28	0.35	0.39	0.49	0.58	0.74	0.90	1.13
BACOLI/LE	0.32	0.36	0.40	0.49	0.57	0.71	0.86	1.06
BACOLRI/ST	0.23	0.28	0.33	0.44	0.54	0.71	0.90	1.19
BACOLRI/LE	0.31	0.33	0.36	0.45	0.54	0.67	0.83	1.07
$tol = 10^{-6}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	0.91	1.04	1.11	1.36	1.41	1.66	1.92	2.28
BACOLI/LE	1.07	1.04	1.20	1.37	1.41	1.52	1.72	2.11
BACOLRI/ST	0.53	0.62	0.64	0.76	0.89	1.13	1.41	1.70
BACOLRI/LE	0.88	0.71	0.77	0.89	0.90	1.06	1.24	1.51
$tol = 10^{-8}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	3.31	3.14	2.76	3.04	3.28	3.85	4.01	4.36
BACOLI/LE	5.57	3.85	3.10	3.07	3.63	4.17	3.76	3.89
BACOLRI/ST	1.87	1.78	1.69	1.95	2.05	2.31	2.37	2.79
BACOLRI/LE	4.96	2.45	2.14	2.07	2.22	2.50	2.44	2.55
$tol = 10^{-10}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	16.71	10.19	8.26	7.49	7.57	8.14	8.82	9.75
BACOLI/LE	—	16.07	9.98	8.77	8.88	9.23	9.73	10.75
BACOLRI/ST	7.54	5.64	4.76	4.72	4.72	5.20	5.52	6.26
BACOLRI/LE	—	9.90	6.86	5.81	5.38	5.44	5.64	6.30

Table 16: *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
BACOLI/ST	0.02	0.03	0.03	0.03	0.04	0.05	0.06	0.07
BACOLI/LE	0.03	0.03	0.03	0.04	0.04	0.05	0.05	0.08
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	0.05	0.07	0.08
$tol = 10^{-6}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	—	0.08	0.08	0.09	0.11	0.12	0.14	0.16
BACOLI/LE	0.11	0.08	0.09	0.09	0.11	0.12	0.13	0.14
BACOLRI/ST	—	0.06	0.07	0.07	0.08	0.09	0.10	0.12
BACOLRI/LE	0.11	0.08	0.07	0.07	0.08	0.09	0.10	0.11
$tol = 10^{-8}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	0.45	0.27	0.23	0.25	0.27	0.29	0.31	0.35
BACOLI/LE	0.89	0.34	0.27	0.26	0.27	0.29	0.30	0.35
BACOLRI/ST	0.22	0.17	0.15	0.16	0.17	0.18	0.21	0.23
BACOLRI/LE	0.67	0.27	0.20	0.18	0.18	0.18	0.21	0.23
$tol = 10^{-10}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	2.13	1.03	0.76	0.73	0.71	0.69	0.74	0.74
BACOLI/LE	6.35	1.83	0.97	0.83	0.77	0.81	0.82	0.80
BACOLRI/ST	0.92	0.58	0.46	0.43	0.40	0.42	0.42	0.45
BACOLRI/LE	4.40	1.23	0.71	0.55	0.48	0.46	0.46	0.46

Table 17: *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
BACOLI/ST	0.27	0.29	0.35	0.44	0.55	0.68	0.80	0.92
BACOLI/LE	0.30	0.32	0.38	0.44	0.57	0.70	0.87	0.95
BACOLRI/ST	0.22	0.24	0.30	0.39	0.51	0.66	0.79	1.00
BACOLRI/LE	0.27	0.28	0.32	0.40	0.51	0.63	0.85	0.93
$tol = 10^{-6}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	0.89	0.91	1.01	1.22	1.42	1.68	1.98	2.33
BACOLI/LE	0.99	0.92	1.05	1.23	1.44	1.67	2.00	2.30
BACOLRI/ST	0.51	0.56	0.61	0.73	0.90	1.12	1.38	1.70
BACOLRI/LE	0.87	0.67	0.70	0.81	0.93	1.13	1.32	1.63
$tol = 10^{-8}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	3.12	2.78	2.69	2.94	3.21	3.58	3.89	4.41
BACOLI/LE	7.08	3.46	3.08	3.01	3.44	3.81	4.05	4.53
BACOLRI/ST	1.77	1.57	1.57	1.67	1.82	2.05	2.39	2.81
BACOLRI/LE	4.74	2.32	1.94	1.88	2.04	2.17	2.39	2.77
$tol = 10^{-10}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	17.58	10.48	8.19	7.78	7.64	7.89	8.66	9.62
BACOLI/LE	—	13.76	10.32	8.71	8.42	8.77	9.20	10.30
BACOLRI/ST	7.38	5.27	4.51	4.28	4.30	4.60	4.94	5.48
BACOLRI/LE	—	10.02	6.53	5.43	5.05	5.14	5.19	5.71

Table 18: *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
BACOLI/ST	0.11	0.13	0.14	0.18	0.23	0.29	0.39	0.44
BACOLI/LE	0.12	0.14	0.16	0.22	0.22	0.28	0.34	0.47
BACOLRI/ST	0.34	0.34	0.45	0.59	0.72	1.05	1.33	1.67
BACOLRI/LE	0.32	0.36	0.45	0.59	0.74	0.93	1.30	1.58
$tol = 10^{-6}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	—	0.41	0.52	0.58	0.72	0.86	1.02	1.19
BACOLI/LE	0.55	0.42	0.46	0.53	0.63	0.78	0.92	1.06
BACOLRI/ST	—	0.59	0.71	0.87	1.11	1.40	1.83	2.44
BACOLRI/LE	0.95	0.79	0.76	0.84	1.07	1.33	1.74	2.09
$tol = 10^{-8}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	2.31	1.62	1.54	1.84	2.11	2.39	2.63	2.95
BACOLI/LE	5.26	1.97	1.63	1.69	1.89	2.21	2.52	2.89
BACOLRI/ST	2.34	1.47	1.47	1.70	1.98	2.44	3.12	3.91
BACOLRI/LE	7.28	2.57	2.07	2.00	2.12	2.18	2.64	3.22
$tol = 10^{-10}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	12.59	6.42	5.19	5.55	5.76	6.04	6.57	6.93
BACOLI/LE	49.65	12.16	6.15	5.38	5.16	6.09	6.65	7.65
BACOLRI/ST	9.39	5.39	4.55	4.50	4.63	4.67	5.12	6.27
BACOLRI/LE	64.15	15.62	8.50	6.56	5.95	5.79	5.87	5.93

Table 19: *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
BACOLI/ST	1.26	1.55	1.92	2.59	3.73	4.61	5.34	6.86
BACOLI/LE	1.47	1.67	2.07	2.81	3.90	4.59	6.19	6.65
BACOLRI/ST	2.01	2.71	4.02	6.31	9.91	14.29	19.96	28.05
BACOLRI/LE	2.49	3.15	4.52	6.52	9.64	14.30	21.05	25.90
$tol = 10^{-6}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	4.55	5.20	6.37	8.75	10.85	13.20	16.25	20.40
BACOLI/LE	5.30	4.89	6.08	8.06	10.15	13.12	16.87	20.21
BACOLRI/ST	4.27	5.41	6.78	9.62	14.03	20.21	29.24	41.97
BACOLRI/LE	6.88	6.17	7.65	9.89	13.83	19.40	26.40	35.91
$tol = 10^{-8}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	17.70	17.38	18.27	22.13	26.14	30.80	35.51	41.76
BACOLI/LE	35.76	19.18	18.89	19.73	27.04	32.07	36.56	44.52
BACOLRI/ST	12.72	13.31	14.35	19.45	23.65	30.11	39.99	55.44
BACOLRI/LE	47.71	20.01	17.61	18.62	21.43	28.65	37.52	49.79
$tol = 10^{-10}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	107.31	69.57	57.48	57.39	63.74	68.06	80.16	92.34
BACOLI/LE	407.62	89.69	66.72	59.15	64.25	75.68	87.43	103.68
BACOLRI/ST	63.49	44.52	40.16	40.57	44.95	55.19	68.47	86.81
BACOLRI/LE	397.64	116.26	68.83	56.74	56.29	58.10	63.43	77.18

Table 20: *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
BACOLI/ST	0.41	0.39	0.50	0.61	0.77	1.07	1.47	1.79
BACOLI/LE	0.43	0.46	0.58	0.74	0.82	1.17	1.49	2.02
BACOLRI/ST	1.37	1.62	2.30	3.20	4.28	6.28	7.92	10.47
BACOLRI/LE	1.47	1.66	2.27	3.13	4.24	5.68	8.00	9.79
$tol = 10^{-6}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	—	1.32	1.77	2.29	2.67	3.06	4.33	5.55
BACOLI/LE	1.88	1.48	1.56	1.82	2.26	2.92	3.33	4.10
BACOLRI/ST	—	2.66	3.44	4.42	6.08	8.50	11.34	15.69
BACOLRI/LE	4.62	3.75	3.73	4.29	5.78	7.61	9.94	13.05
$tol = 10^{-8}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	17.66	5.79	5.69	7.21	8.75	10.14	11.09	13.67
BACOLI/LE	21.13	6.57	5.42	5.89	7.37	8.94	10.99	13.65
BACOLRI/ST	9.33	7.26	7.31	8.76	10.65	14.15	19.50	24.73
BACOLRI/LE	38.94	14.65	11.06	10.65	11.69	12.61	16.67	21.33
$tol = 10^{-10}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	50.94	25.57	20.52	22.91	24.10	27.34	30.52	33.96
BACOLI/LE	208.28	52.12	24.95	21.54	20.12	26.13	35.30	35.99
BACOLRI/ST	46.38	29.85	25.98	25.71	26.77	29.66	32.14	40.00
BACOLRI/LE	288.35	78.36	46.28	38.49	34.83	36.44	38.13	40.05

Table 21: *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
BACOLI/ST	3.80	4.79	6.55	8.91	12.97	18.30	22.37	27.48
BACOLI/LE	4.87	5.91	7.57	9.75	14.32	18.28	24.82	27.18
BACOLRI/ST	9.79	14.44	23.29	38.00	59.66	91.59	127.96	177.82
BACOLRI/LE	12.17	17.35	25.84	39.07	60.61	89.03	139.27	178.23
$tol = 10^{-6}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	13.42	17.76	23.44	33.44	44.06	55.97	73.88	92.37
BACOLI/LE	16.18	15.70	21.47	31.94	42.96	57.49	72.90	92.60
BACOLRI/ST	18.54	27.38	36.61	54.74	84.52	127.04	181.78	260.99
BACOLRI/LE	35.13	31.11	40.81	58.25	80.70	115.06	165.88	243.05
$tol = 10^{-8}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	55.00	61.81	69.12	88.96	110.07	135.71	162.07	199.16
BACOLI/LE	151.97	64.20	66.00	75.16	108.15	149.44	173.97	217.83
BACOLRI/ST	75.20	63.64	75.98	107.21	142.44	175.69	241.93	363.33
BACOLRI/LE	253.68	105.34	95.75	98.99	124.39	169.30	236.01	321.72
$tol = 10^{-10}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	419.25	256.06	238.42	242.63	275.18	315.42	384.20	464.07
BACOLI/LE	—	359.34	269.52	244.37	279.38	358.78	434.57	542.50
BACOLRI/ST	326.85	239.56	221.16	232.70	273.21	340.68	437.58	555.79
BACOLRI/LE	1790.23	554.68	377.45	326.85	333.52	354.91	398.94	483.24

Table 22: *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
BACOLI/ST	0.01	0.01	0.02	0.02	0.03	0.03	0.05	0.06
BACOLI/LE	0.02	0.02	0.02	0.03	0.03	0.04	0.04	0.05
BACOLRI/ST	0.08	0.08	0.09	0.10	0.11	0.14	0.15	0.18
BACOLRI/LE	0.10	0.10	0.10	0.11	0.12	0.13	0.15	0.18
$tol = 10^{-6}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	0.05	0.05	—	0.07	0.09	0.11	0.14	—
BACOLI/LE	0.09	0.09	0.08	0.08	0.09	0.10	0.12	0.16
BACOLRI/ST	0.13	0.13	0.15	0.17	0.20	0.22	0.28	0.31
BACOLRI/LE	0.21	0.18	0.17	0.20	0.21	0.24	0.27	0.32
$tol = 10^{-8}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	0.23	0.19	0.21	0.22	0.26	0.31	0.32	0.41
BACOLI/LE	0.54	0.27	0.25	0.26	0.25	0.32	0.35	—
BACOLRI/ST	0.33	0.27	0.30	0.32	0.36	0.41	0.47	0.55
BACOLRI/LE	0.86	0.50	0.41	0.44	0.42	0.43	0.48	0.56
$tol = 10^{-10}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	1.06	0.82	0.69	0.68	0.77	—	0.79	0.93
BACOLI/LE	3.55	1.20	0.82	0.69	0.74	0.76	0.78	—
BACOLRI/ST	1.14	0.84	0.79	0.80	0.88	0.92	0.95	1.10
BACOLRI/LE	4.76	1.81	1.31	1.14	1.12	1.16	1.11	1.09

Table 23: *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
BACOLRI/ST	0.13	0.14	0.14	0.14	0.14	0.15	0.16	0.16
BACOLRI/LE	0.13	0.13	0.14	0.14	0.14	0.15	0.16	0.16
$tol = 10^{-6}/p =$	4	5	6	7	8	9	10	11
BACOLRI/ST	0.16	0.15	0.15	0.16	0.17	0.17	0.18	0.19
BACOLRI/LE	0.18	0.15	0.15	0.16	0.17	0.17	0.18	0.19
$tol = 10^{-8}/p =$	4	5	6	7	8	9	10	11
BACOLRI/ST	0.22	0.19	0.19	0.20	0.20	0.21	0.22	0.24
BACOLRI/LE	0.45	0.28	0.22	0.22	0.23	0.22	0.22	0.23
$tol = 10^{-10}/p =$	4	5	6	7	8	9	10	11
BACOLRI/ST	0.56	0.40	0.35	0.32	0.34	0.35	0.32	0.34
BACOLRI/LE	1.89	0.85	0.56	0.46	0.38	0.37	0.35	0.35

Table 24: *Machine dependent timings (in seconds), Schrödinger Equation,* $p = 4, \ldots, 11$, $tol = 10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}$.

$error = 10^{-4}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	0.02	0.02	0.02	0.03	0.04	0.04	0.05	0.06
BACOLI/LE	0.02	0.02	0.02	0.03	0.03	0.04	0.04	0.05
BACOLRI/ST	0.03	0.03	0.03	0.03	0.04	0.05	0.05	0.06
BACOLRI/LE	0.03	0.02	0.03	0.03	0.03	0.04	0.05	0.05
$error = 10^{-6}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	0.09	0.08	0.08	0.08	0.09	0.11	0.11	0.13
BACOLI/LE	0.08	0.06	0.07	0.07	0.08	0.09	0.10	0.12
BACOLRI/ST	0.07	0.07	0.07	0.07	0.08	0.09	0.10	0.12
BACOLRI/LE	0.10	0.07	0.07	0.07	0.07	0.08	0.09	0.10
$error = 10^{-8}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	0.37	0.28	0.24	0.24	0.25	0.26	0.27	0.29
BACOLI/LE	0.46	0.21	0.21	0.20	0.21	0.25	0.25	0.27
BACOLRI/ST	0.19	0.17	0.15	0.16	0.16	0.18	0.19	0.21
BACOLRI/LE	0.42	0.20	0.17	0.16	0.16	0.16	0.17	0.19
$error = 10^{-10}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	1.50	0.93	0.76	0.66	0.65	0.66	0.64	0.64
BACOLI/LE	2.47	0.72	0.68	0.57	0.58	0.64	0.60	0.59
BACOLRI/ST	0.52	0.43	0.34	0.33	0.32	0.34	0.35	0.38
BACOLRI/LE	1.67	0.59	0.43	0.36	0.33	0.32	0.32	0.35

Table 25: *Fitted CPU time (in seconds) for obtained errors, One Layer Burgers equation,* $\epsilon = 10^{-3}$, $p = 4, \ldots, 11$, $error = 10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}$.

$error = 10^{-4}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	0.44	0.45	0.44	0.51	0.56	0.65	0.73	0.91
BACOLI/LE	0.35	0.38	0.41	0.45	0.50	0.60	0.73	0.82
BACOLRI/ST	0.31	0.34	0.37	0.46	0.53	0.68	0.76	1.00
BACOLRI/LE	0.36	0.34	0.37	0.42	0.48	0.58	0.71	0.85
$error = 10^{-6}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	1.30	1.16	1.08	1.20	1.25	1.42	1.57	1.85
BACOLI/LE	1.54	1.24	1.12	1.25	1.27	1.41	1.58	1.68
BACOLRI/ST	0.78	0.78	0.76	0.91	0.99	1.21	1.30	1.63
BACOLRI/LE	1.42	0.95	0.87	0.94	0.97	1.10	1.24	1.40
$error = 10^{-8}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	3.84	3.00	2.65	2.78	2.83	3.12	3.37	3.78
BACOLI/LE	6.75	4.03	3.03	3.42	3.25	3.31	3.41	3.44
BACOLRI/ST	1.96	1.76	1.54	1.78	1.82	2.14	2.25	2.65
BACOLRI/LE	5.65	2.68	2.08	2.10	1.96	2.09	2.18	2.30
$error = 10^{-10}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	11.34	7.79	6.51	6.48	6.37	6.82	7.21	7.71
BACOLI/LE	29.66	13.15	8.20	9.38	8.32	7.78	7.39	7.05
BACOLRI/ST	4.92	3.99	3.15	3.49	3.36	3.80	3.88	4.31
BACOLRI/LE	22.52	7.53	4.97	4.68	3.99	3.95	3.83	3.77

Table 26: *Fitted CPU time (in seconds) for obtained errors, One Layer Burgers equation,* $\epsilon = 10^{-4}$, $p = 4, \ldots, 11$, $error = 10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}$.

$error = 10^{-4}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	0.02	0.02	0.02	0.02	0.03	0.03	0.04	0.04
BACOLI/LE	0.01	0.01	0.02	0.02	0.02	0.03	0.03	0.04
BACOLRI/ST	0.03	0.02	0.03	0.03	0.03	0.04	0.04	0.05
BACOLRI/LE	0.02	0.02	0.02	0.02	0.03	0.03	0.04	0.05
$error = 10^{-6}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	0.10	0.07	0.07	0.07	0.08	0.09	0.10	0.11
BACOLI/LE	0.08	0.05	0.05	0.06	0.07	0.08	0.09	0.10
BACOLRI/ST	0.08	0.06	0.06	0.06	0.06	0.07	0.08	0.09
BACOLRI/LE	0.08	0.06	0.05	0.05	0.06	0.07	0.08	0.09
$error = 10^{-8}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	0.43	0.24	0.21	0.22	0.22	0.23	0.25	0.26
BACOLI/LE	0.50	0.20	0.17	0.17	0.18	0.21	0.23	0.24
BACOLRI/ST	0.23	0.15	0.13	0.13	0.13	0.15	0.16	0.18
BACOLRI/LE	0.37	0.18	0.14	0.13	0.13	0.14	0.15	0.16
$error = 10^{-10}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	1.82	0.84	0.69	0.69	0.64	0.64	0.62	0.64
BACOLI/LE	3.12	0.72	0.57	0.51	0.52	0.57	0.62	0.58
BACOLRI/ST	0.64	0.37	0.30	0.29	0.28	0.30	0.30	0.34
BACOLRI/LE	1.61	0.54	0.36	0.30	0.28	0.27	0.28	0.30

Table 27: *Fitted CPU time (in seconds) for obtained errors, Two Layer Burgers equation,* $\epsilon = 10^{-3}$, $p = 4, \ldots, 11$, $error = 10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}$.

$error = 10^{-4}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	0.37	0.37	0.41	0.47	0.56	0.66	0.76	0.88
BACOLI/LE	0.29	0.34	0.38	0.45	0.54	0.63	0.76	0.90
BACOLRI/ST	0.26	0.28	0.33	0.40	0.49	0.61	0.75	0.92
BACOLRI/LE	0.31	0.30	0.34	0.40	0.50	0.62	0.75	0.94
$error = 10^{-6}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	1.21	1.05	1.04	1.13	1.28	1.45	1.63	1.85
BACOLI/LE	1.46	1.09	1.05	1.17	1.29	1.47	1.69	1.93
BACOLRI/ST	0.68	0.67	0.72	0.79	0.91	1.10	1.29	1.55
BACOLRI/LE	1.29	0.84	0.78	0.86	0.96	1.12	1.30	1.56
$error = 10^{-8}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	3.92	2.98	2.63	2.73	2.91	3.21	3.49	3.90
BACOLI/LE	7.29	3.56	2.89	3.08	3.05	3.40	3.77	4.12
BACOLRI/ST	1.79	1.60	1.55	1.57	1.70	1.98	2.24	2.61
BACOLRI/LE	5.45	2.38	1.80	1.83	1.86	2.03	2.25	2.61
$error = 10^{-10}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	12.75	8.42	6.66	6.57	6.63	7.09	7.51	8.20
BACOLI/LE	36.34	11.58	7.95	8.08	7.21	7.89	8.42	8.80
BACOLRI/ST	4.67	3.81	3.34	3.13	3.18	3.57	3.87	4.41
BACOLRI/LE	23.03	6.69	4.16	3.91	3.61	3.67	3.89	4.36

Table 28: *Fitted CPU time (in seconds) for obtained errors, Two Layer Burgers equation,* $\epsilon = 10^{-4}$, $p = 4, \ldots, 11$, $error = 10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}$.

$error = 10^{-4}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	0.40	0.30	0.37	0.40	0.52	0.71	0.84	1.08
BACOLI/LE	0.20	0.24	0.29	0.36	0.49	0.58	0.75	0.95
BACOLRI/ST	1.38	1.20	1.52	2.14	2.93	4.18	5.77	7.50
BACOLRI/LE	0.82	0.98	1.31	1.66	2.47	3.58	5.08	6.85
$error = 10^{-6}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	1.76	1.15	1.28	1.44	1.77	2.21	2.50	3.04
BACOLI/LE	1.38	0.89	0.96	1.11	1.39	1.75	2.23	2.65
BACOLRI/ST	3.57	2.68	3.01	3.86	4.93	6.59	8.87	11.61
BACOLRI/LE	3.87	2.86	3.09	3.44	4.45	5.85	7.67	10.11
$error = 10^{-8}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	7.64	4.37	4.42	5.26	6.00	6.83	7.47	8.57
BACOLI/LE	9.50	3.23	3.22	3.37	3.93	5.27	6.68	7.40
BACOLRI/ST	9.21	5.97	5.94	6.98	8.29	10.39	13.65	17.97
BACOLRI/LE	18.31	8.33	7.28	7.12	8.03	9.57	11.59	14.93
$error = 10^{-10}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	33.20	16.56	15.28	19.13	20.36	21.14	22.25	24.10
BACOLI/LE	65.53	11.76	10.80	10.27	11.09	15.88	19.97	20.67
BACOLRI/ST	23.77	13.33	11.73	12.62	13.93	16.38	20.99	27.82
BACOLRI/LE	86.64	24.29	17.14	14.75	14.50	15.64	17.50	22.04

Table 29: *Fitted CPU time (in seconds) for obtained errors, Two Layer Burgers equation* $\times 12$, $\epsilon = 10^{-3}$, $p = 4, \ldots, 11$, *error* $= 10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}$.

$error = 10^{-4}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	5.79	6.29	7.80	9.90	13.43	17.88	22.33	26.49
BACOLI/LE	4.71	5.65	6.97	9.47	12.58	16.94	21.99	27.70
BACOLRI/ST	9.99	14.17	20.95	31.82	48.31	72.37	106.75	153.56
BACOLRI/LE	12.26	15.10	21.23	31.07	47.61	70.96	104.58	147.54
$error = 10^{-6}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	20.16	18.93	21.58	26.00	33.48	43.42	52.80	61.48
BACOLI/LE	25.27	18.52	18.98	24.86	32.77	43.25	53.27	65.96
BACOLRI/ST	23.06	28.63	36.66	51.46	74.41	108.07	155.89	220.72
BACOLRI/LE	54.57	38.43	41.14	53.46	73.12	101.41	146.08	203.97
$error = 10^{-8}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	70.15	57.01	59.67	68.32	83.46	105.42	124.86	142.71
BACOLI/LE	135.46	60.73	51.68	65.23	85.39	110.43	129.02	157.04
BACOLRI/ST	53.24	57.85	64.16	83.23	114.61	161.37	227.63	317.26
BACOLRI/LE	242.90	97.84	79.74	91.99	112.31	144.92	204.03	281.99
$error = 10^{-10}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	244.07	171.63	165.00	179.50	208.05	255.96	295.26	331.25
BACOLI/LE	726.29	199.11	140.68	171.19	222.48	282.00	312.53	373.90
BACOLRI/ST	122.93	116.89	112.26	134.61	176.53	240.96	332.39	456.03
BACOLRI/LE	1081.15	249.08	154.54	158.28	172.51	207.12	284.98	389.85

Table 30: *Fitted CPU time (in seconds) for obtained errors, Two Layer Burgers equation* $\times 12$, $\epsilon = 10^{-4}$, $p = 4, \ldots, 11$, *error* $= 10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}$.

$error = 10^{-4}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	0.06491	0.04877	0.05487	0.06183	0.07139	0.08022	0.09410	0.11119
BACOLI/LE	0.09407	0.07001	0.06639	0.07083	0.07743	0.08629	0.09886	0.11056
BACOLRI/ST	0.05309	0.05308	0.06012	0.07039	0.08229	0.09343	0.10477	0.12085
BACOLRI/LE	0.04706	0.04941	0.06237	0.07537	0.08351	0.09518	0.10907	0.12459
$error = 10^{-6}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	0.53660	0.32317	0.28044	0.25413	0.26098	0.26262	0.28506	0.32694
BACOLI/LE	2.17087	0.68538	0.41854	0.32192	0.30028	0.29721	0.30750	0.30991
BACOLRI/ST	0.10694	0.09636	0.10059	0.11116	0.12714	0.14266	0.16071	0.18186
BACOLRI/LE	0.15202	0.11969	0.11859	0.12940	0.13785	0.15253	0.16725	0.18797
$error = 10^{-8}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	—	—	—	—	—	—	—	—
BACOLI/LE	—	—	—	—	—	—	—	—
BACOLRI/ST	0.21540	0.17492	0.16831	0.17554	0.19645	0.21782	0.24654	0.27369
BACOLRI/LE	0.49104	0.28994	0.22546	0.22217	0.22754	0.24446	0.25645	0.28360
$error = 10^{-10}/p =$	4	5	6	7	8	9	10	11
BACOLI/ST	—	—	—	—	—	—	—	—
BACOLI/LE	—	—	—	—	—	—	—	—
BACOLRI/ST	0.43389	0.31754	0.28161	0.27721	0.30354	0.33259	0.37820	0.41188
BACOLRI/LE	—	0.70237	0.42863	0.38144	0.37559	0.39178	0.39324	0.42787

Table 31: *Fitted CPU time (in seconds) for obtained errors, Schrödinger System,* $p = 4, \ldots, 11$, $error = 10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}$.

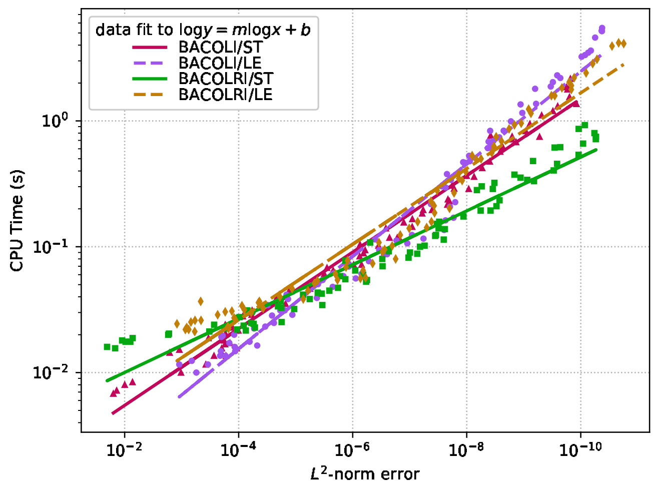

Figure 193: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 4$

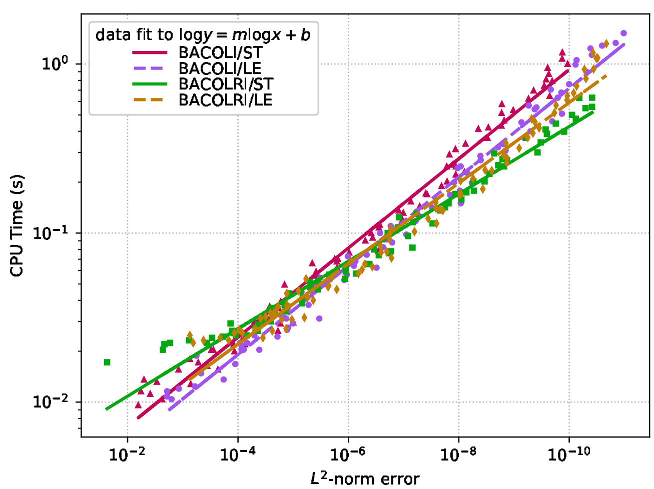

Figure 194: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 5$

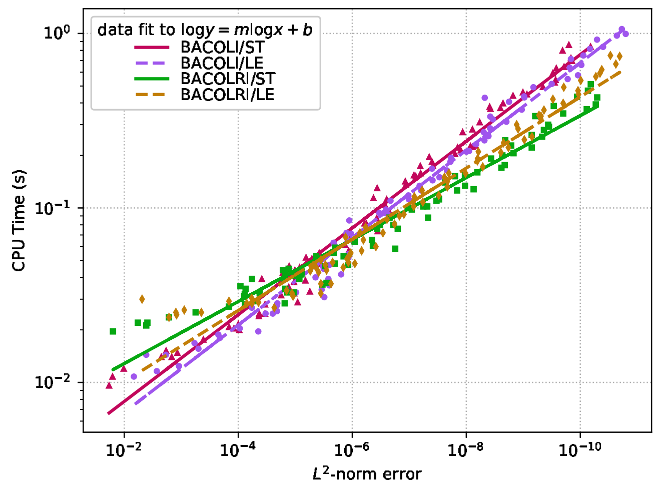

Figure 195: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 6$

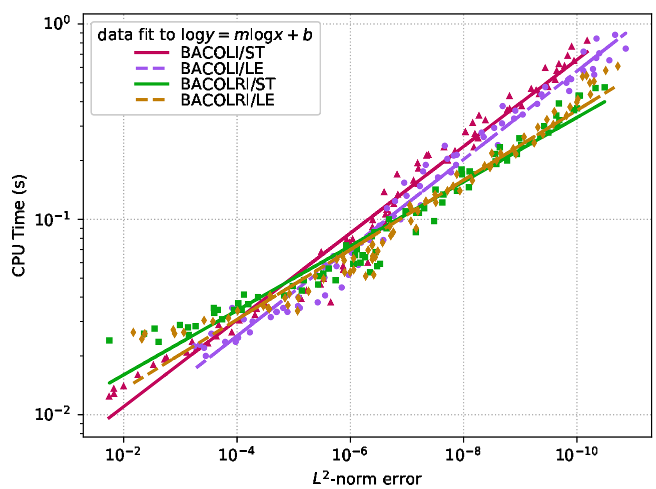

Figure 196: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 7$

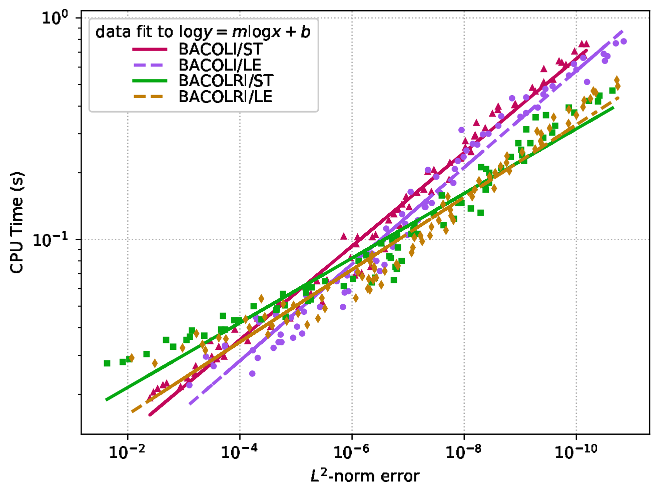

Figure 197: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 8$

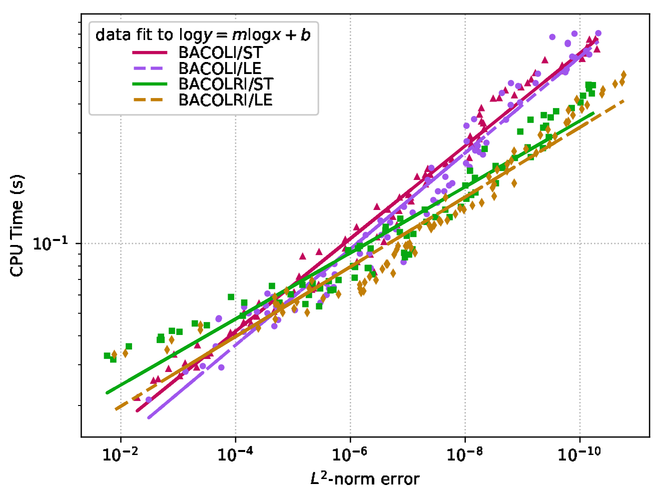

Figure 198: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 9$

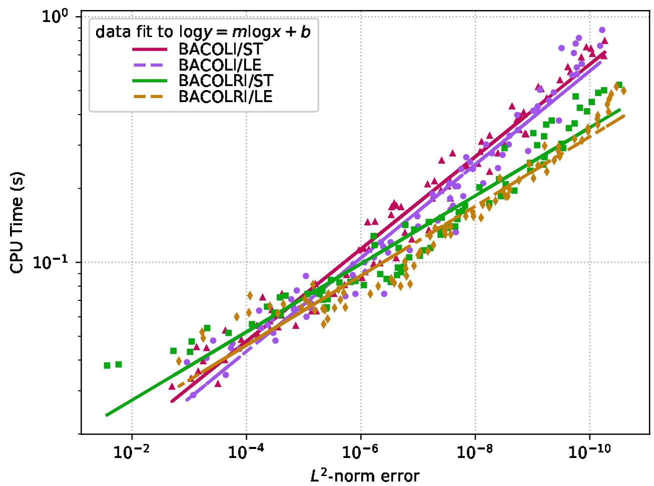

Figure 199: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 10$

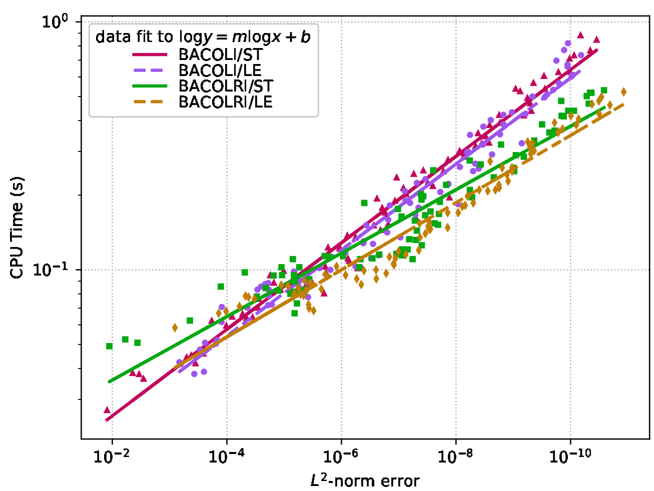

Figure 200: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 11$

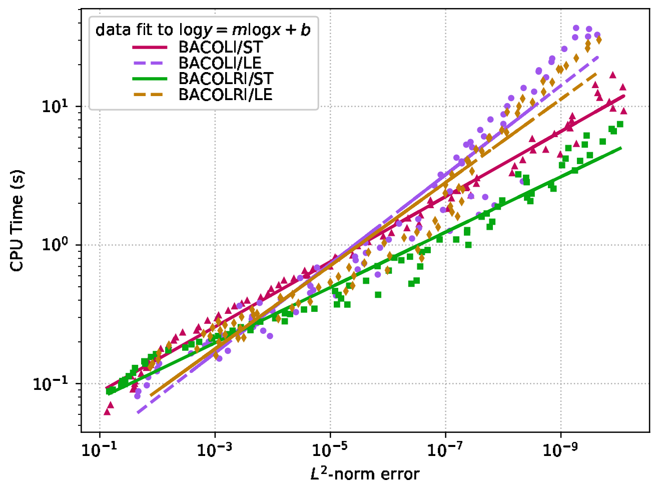

Figure 201: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 4$

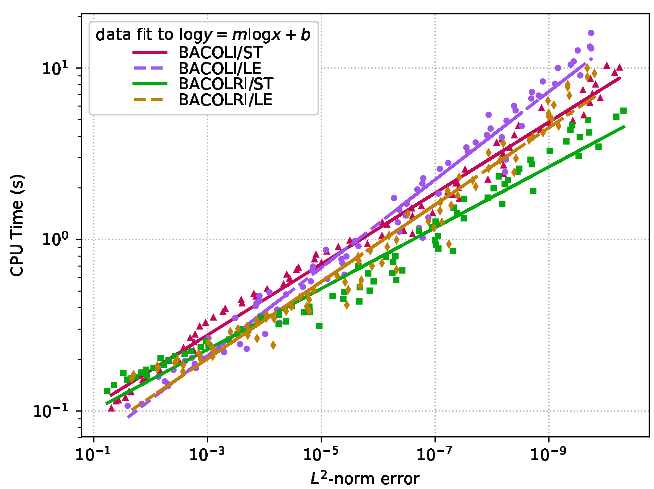

Figure 202: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 5$

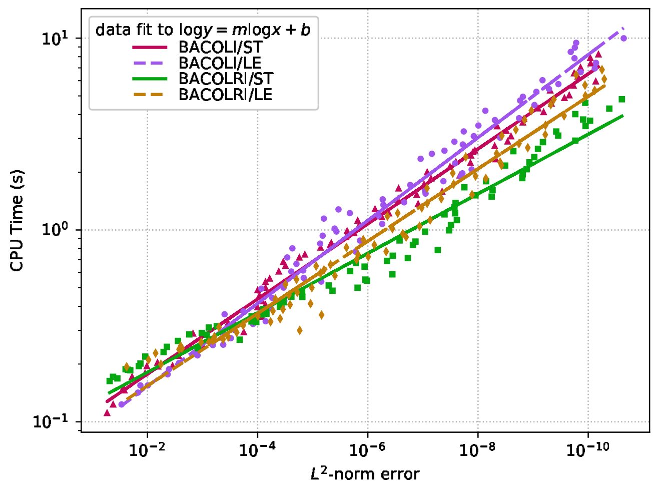

Figure 203: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 6$

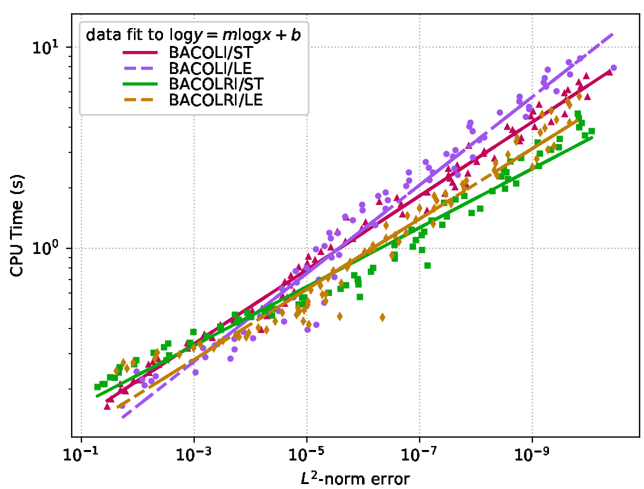

Figure 204: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 7$

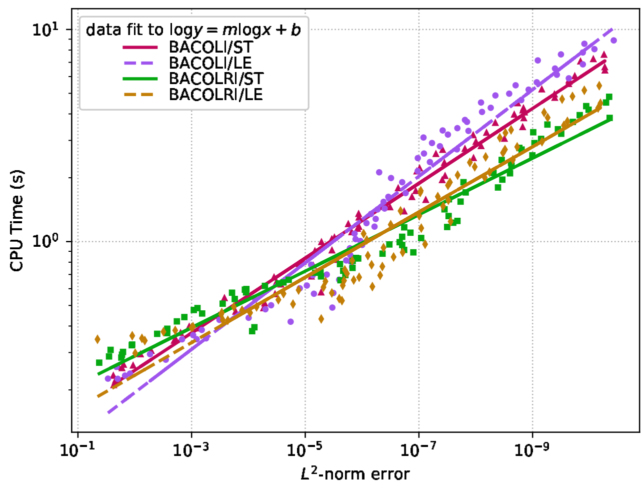

Figure 205: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 8$

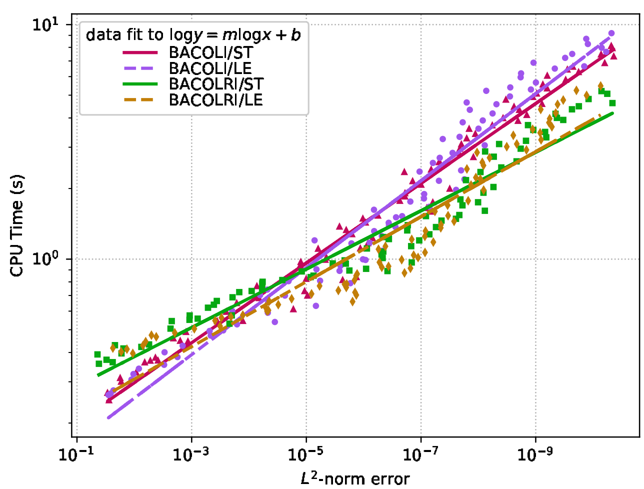

Figure 206: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 9$

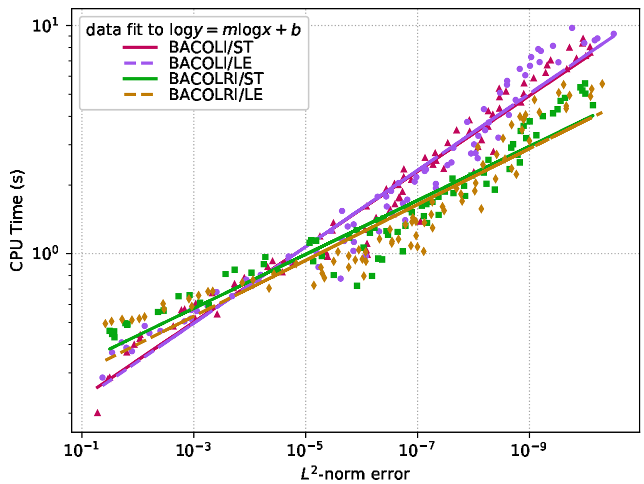

Figure 207: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 10$

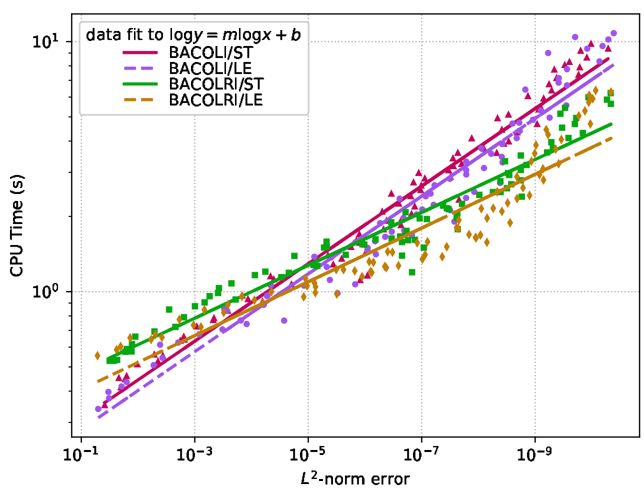

Figure 208: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 11$

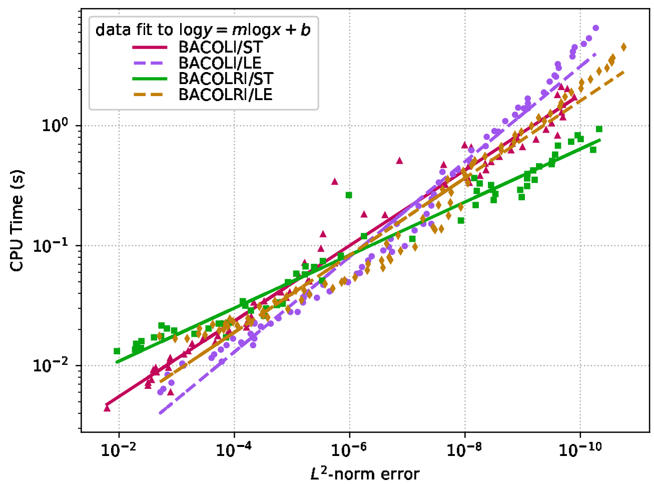

Figure 209: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 4$

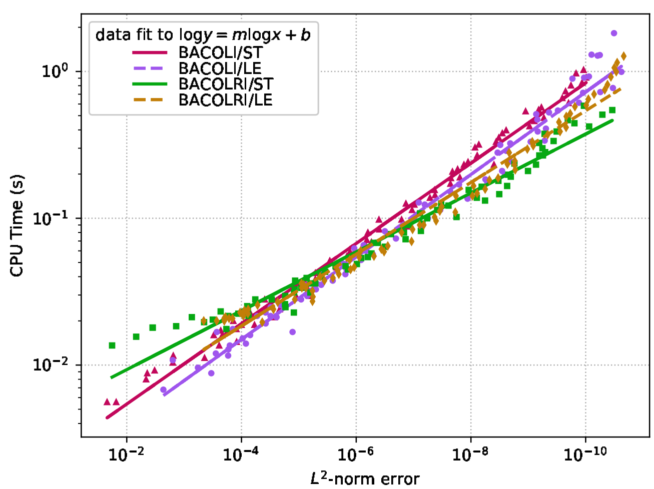

Figure 210: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 5$

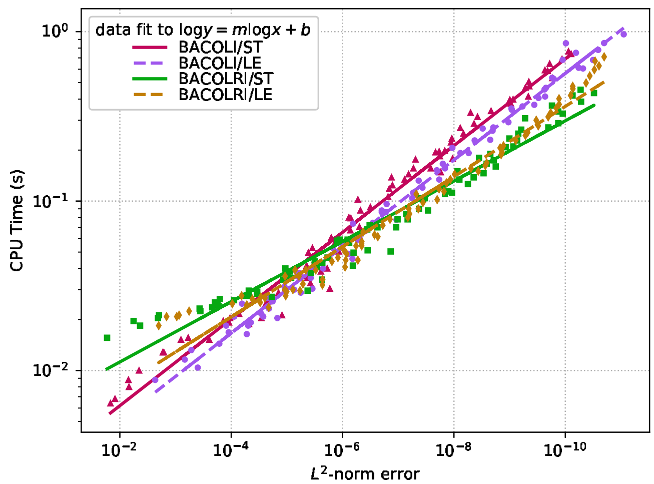

Figure 211: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 6$

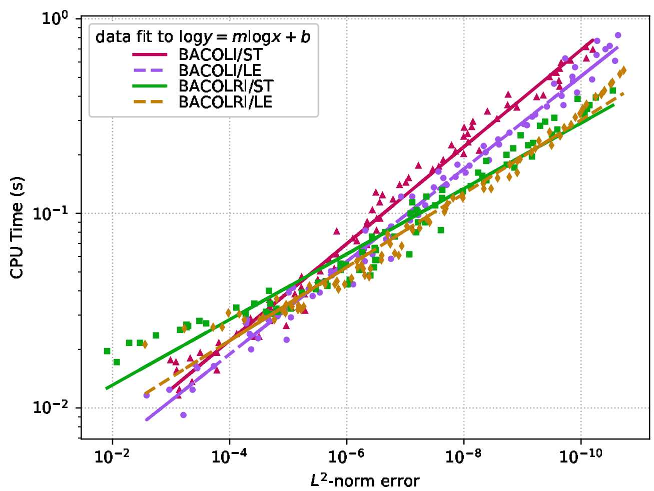

Figure 212: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 7$

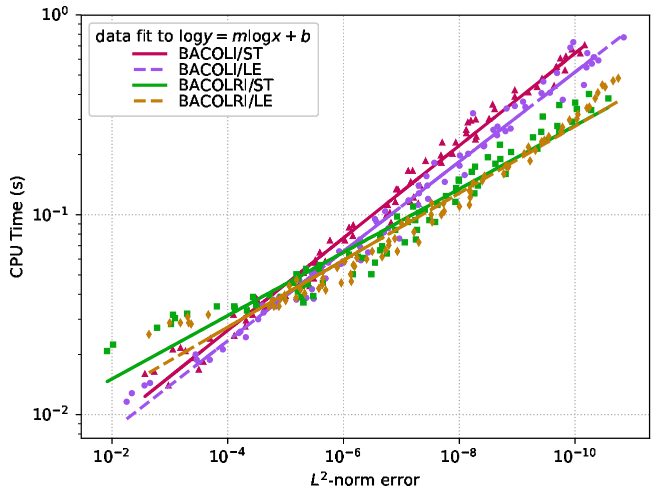

Figure 213: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 8$

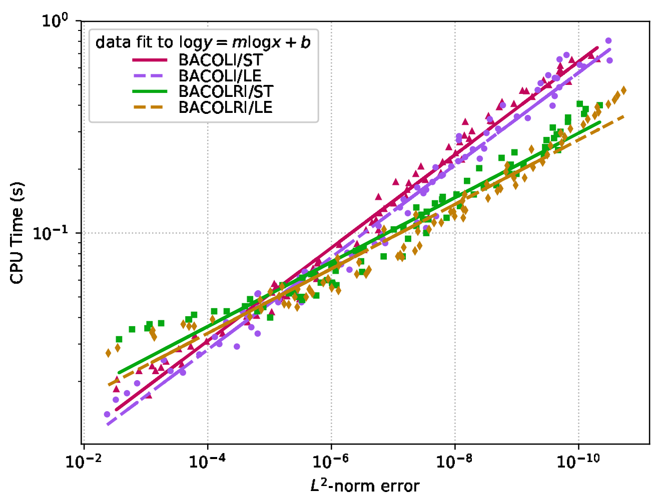

Figure 214: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 9$

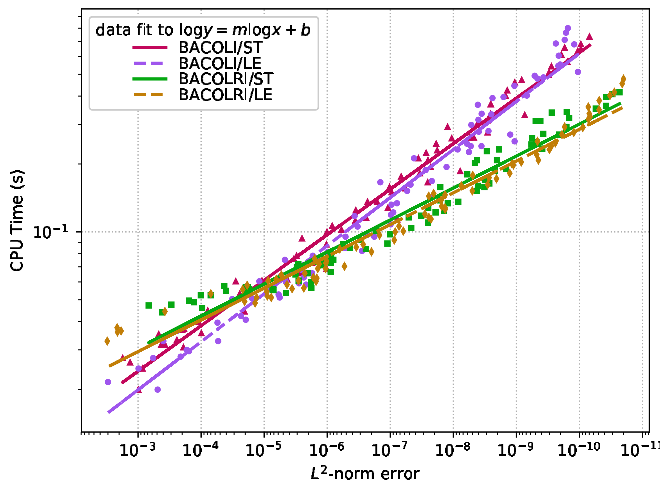

Figure 215: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 10$

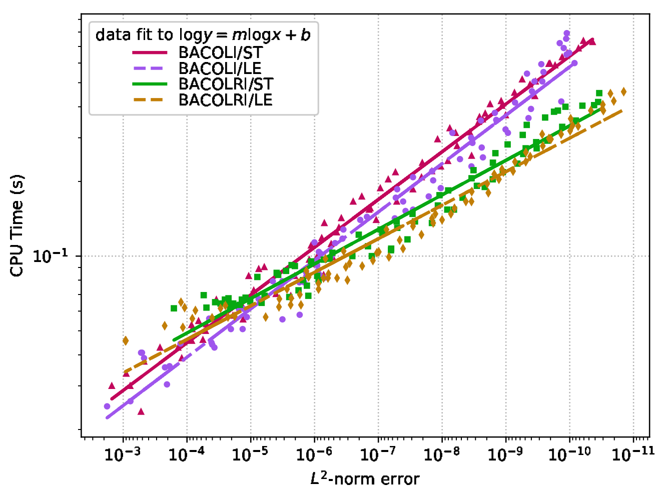

Figure 216: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 11$

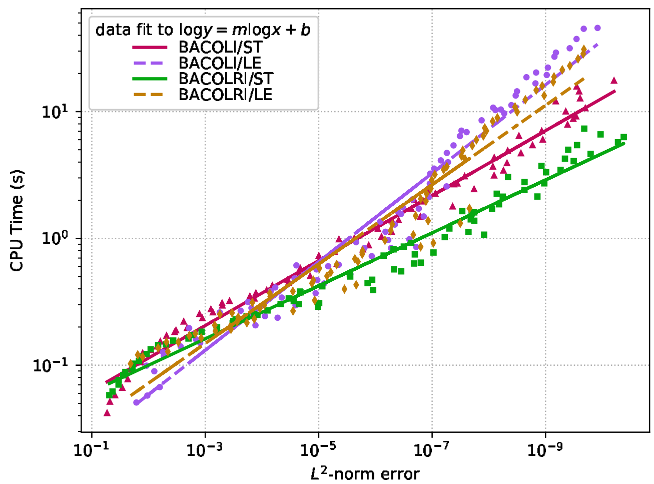

Figure 217: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 4$

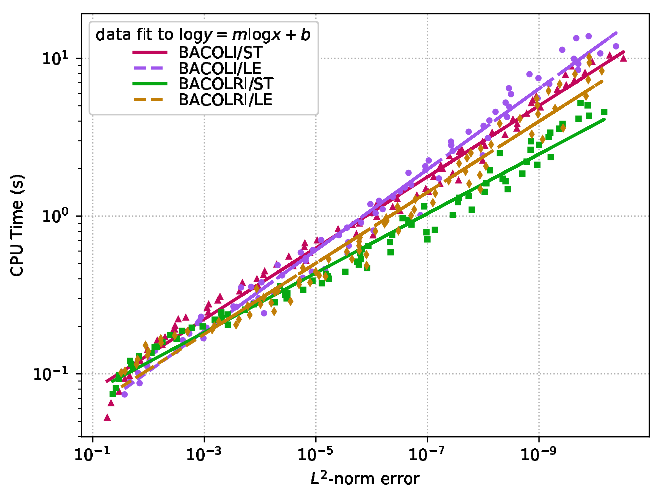

Figure 218: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 5$

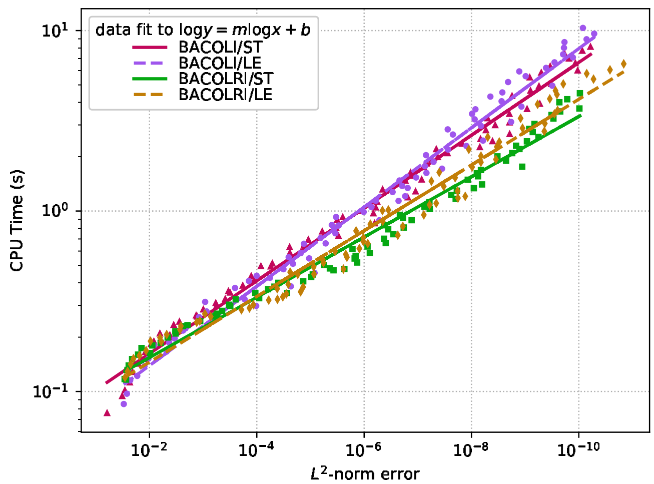

Figure 219: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 6$

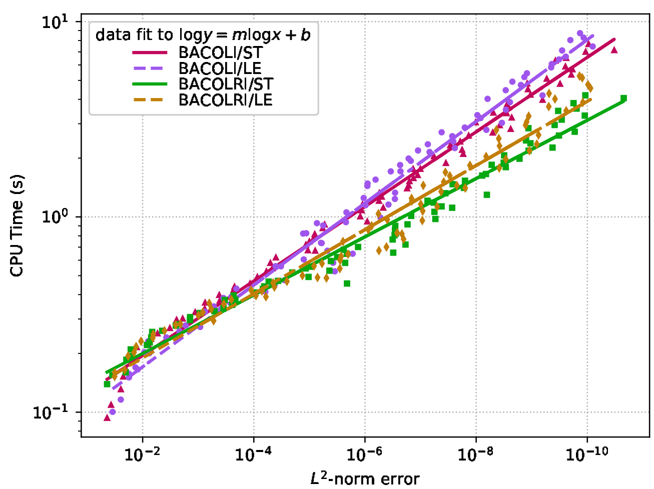

Figure 220: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 7$

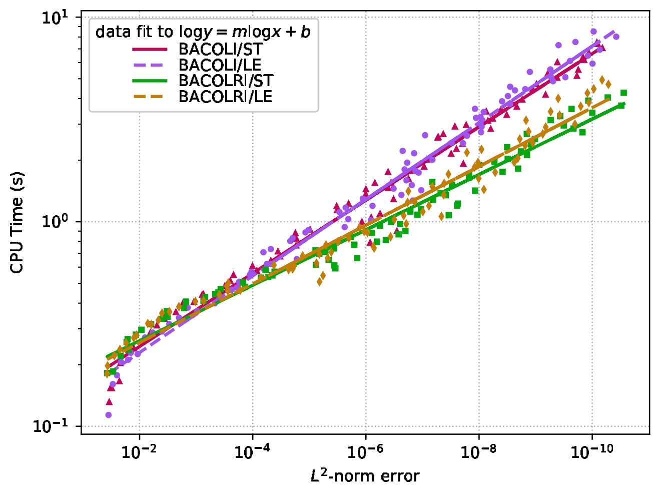

Figure 221: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 8$

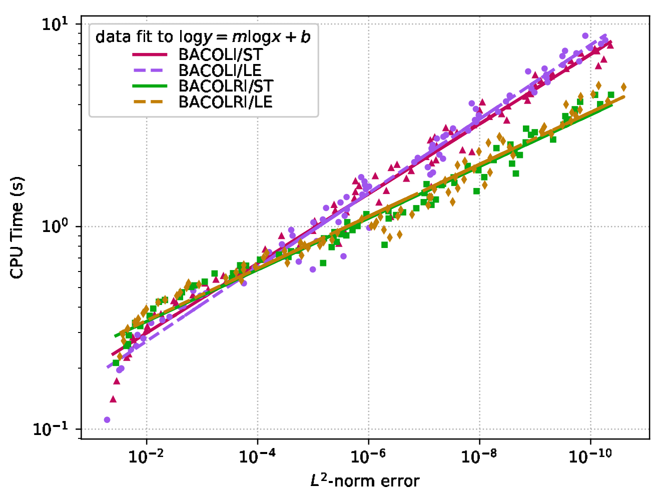

Figure 222: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 9$

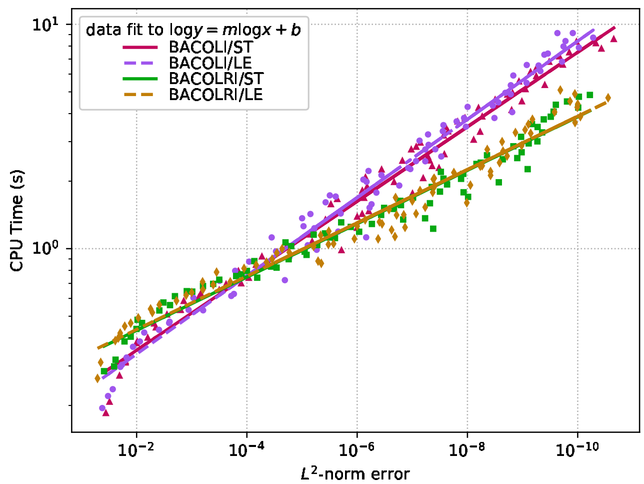

Figure 223: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 10$

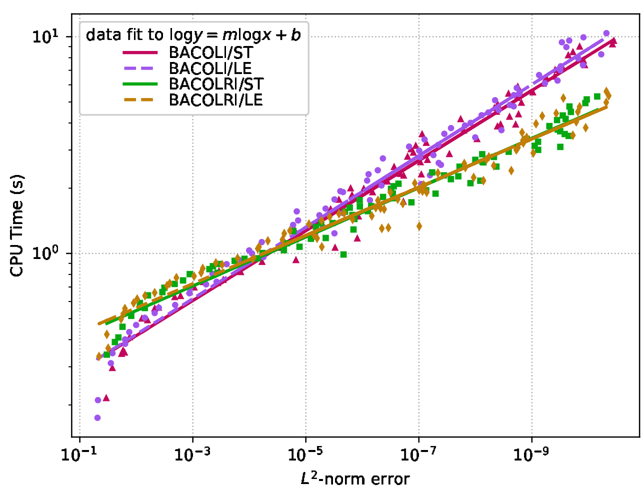

Figure 224: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 11$

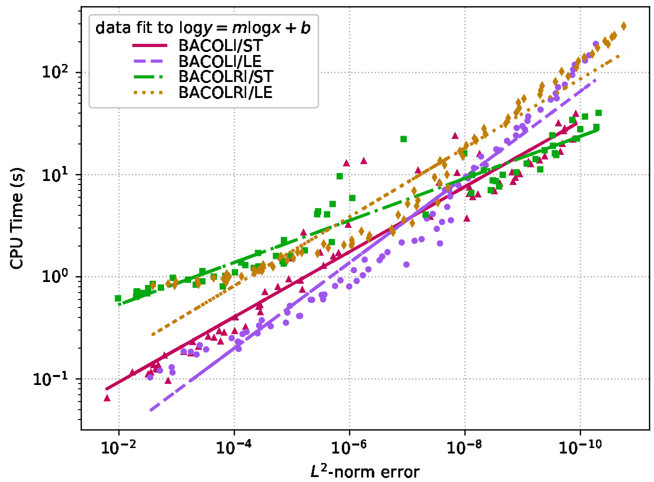

Figure 225: Work vs. Accuracy: Two Layer Burgers equation ×12, $\epsilon = 10^{-3}, p = 4$

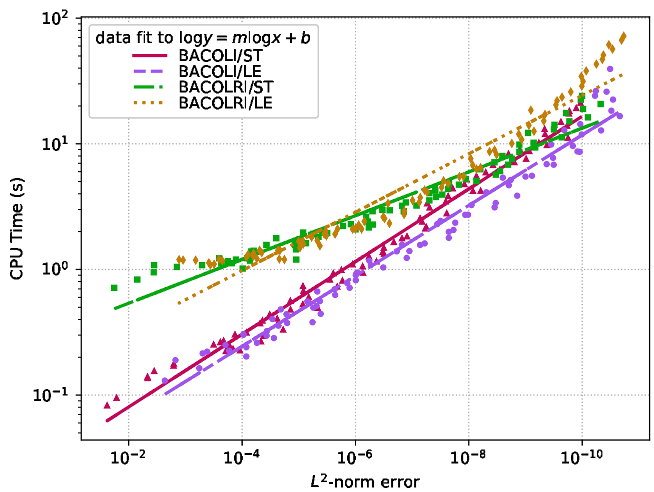

Figure 226: Work vs. Accuracy: Two Layer Burgers equation ×12, $\epsilon = 10^{-3}, p = 5$

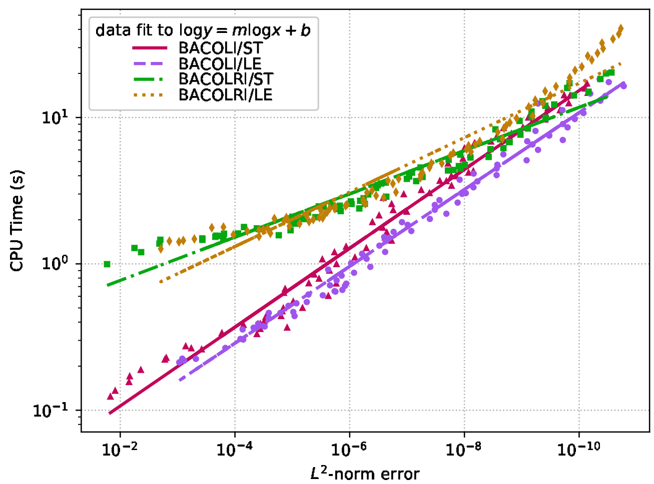

Figure 227: Work vs. Accuracy: Two Layer Burgers equation $\times 12$, $\epsilon = 10^{-3}, p = 6$

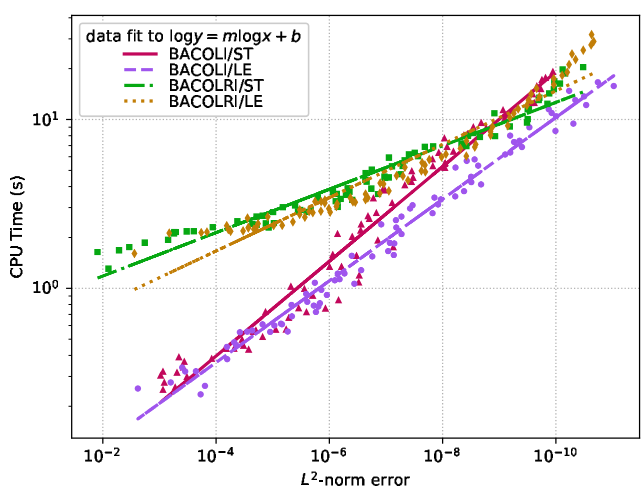

Figure 228: Work vs. Accuracy: Two Layer Burgers equation $\times 12$, $\epsilon = 10^{-3}, p = 7$

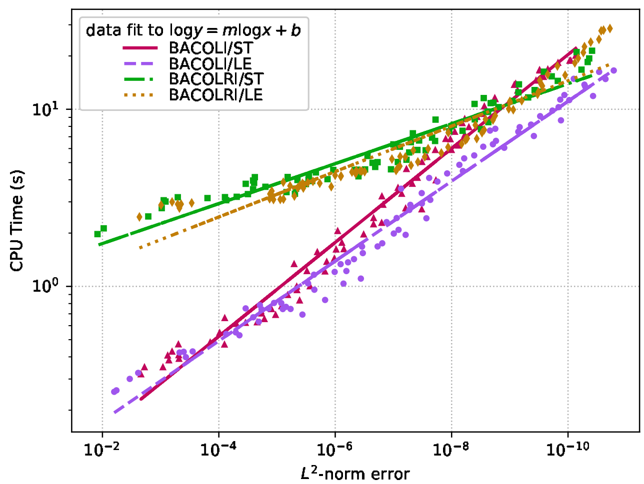

Figure 229: Work vs. Accuracy: Two Layer Burgers equation $\times 12$, $\epsilon = 10^{-3}, p = 8$

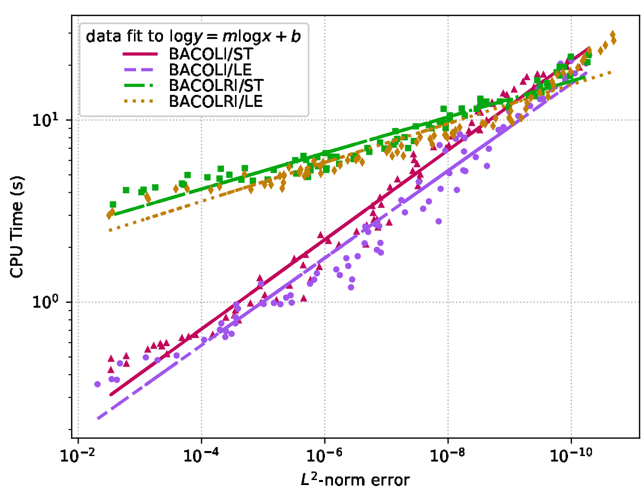

Figure 230: Work vs. Accuracy: Two Layer Burgers equation $\times 12$, $\epsilon = 10^{-3}, p = 9$

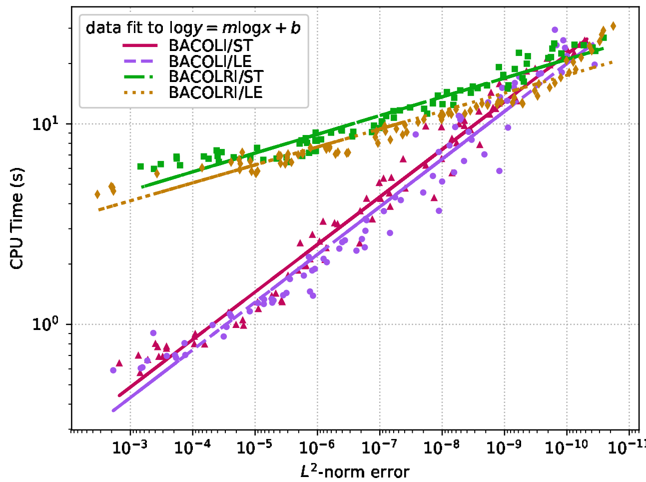

Figure 231: Work vs. Accuracy: Two Layer Burgers equation $\times 12$, $\epsilon = 10^{-3}, p = 10$

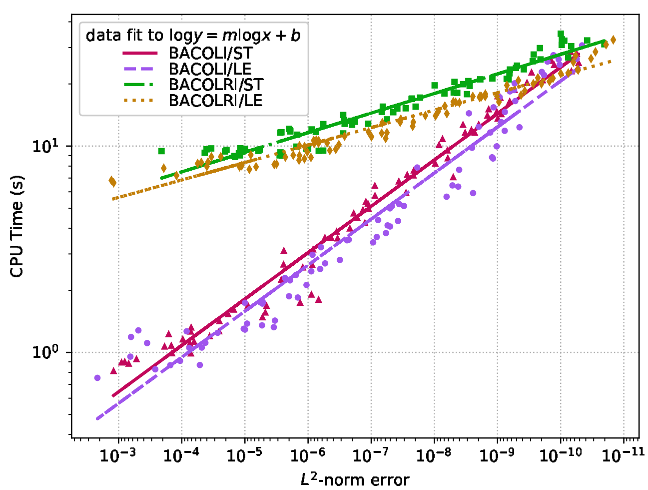

Figure 232: Work vs. Accuracy: Two Layer Burgers equation $\times 12$, $\epsilon = 10^{-3}, p = 11$

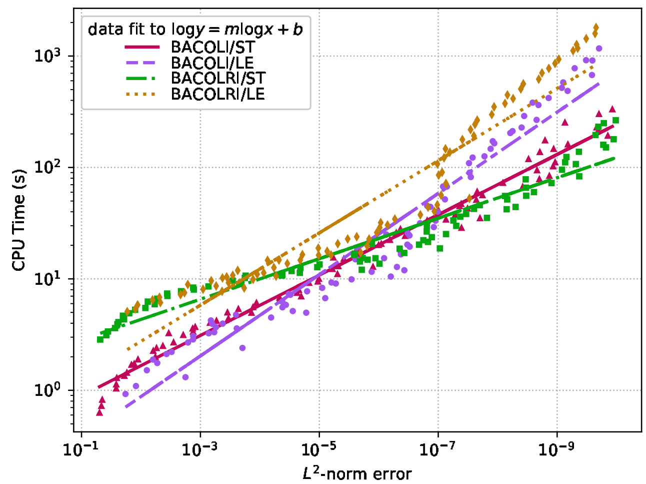

Figure 233: Work vs. Accuracy: Two Layer Burgers equation ×12, $\epsilon = 10^{-4}, p = 4$

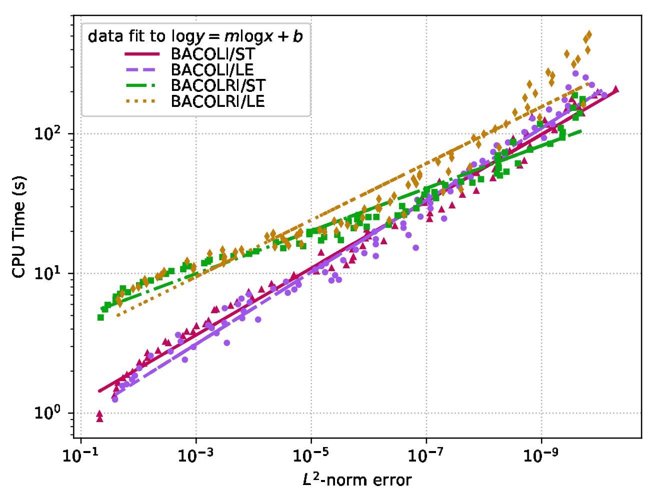

Figure 234: Work vs. Accuracy: Two Layer Burgers equation ×12, $\epsilon = 10^{-4}, p = 5$

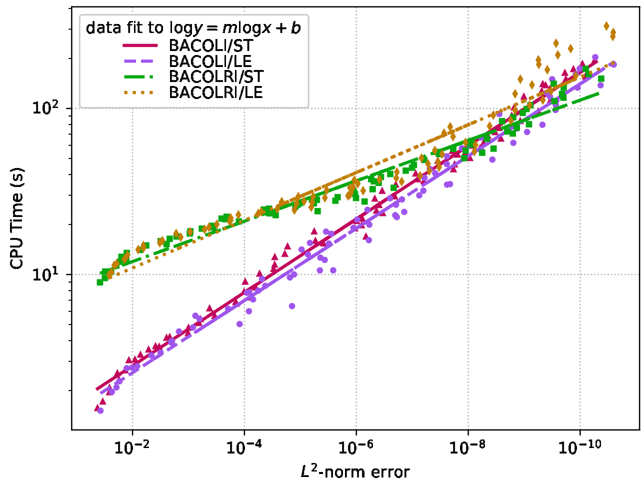

Figure 235: Work vs. Accuracy: Two Layer Burgers equation $\times 12$, $\epsilon = 10^{-4}, p = 6$

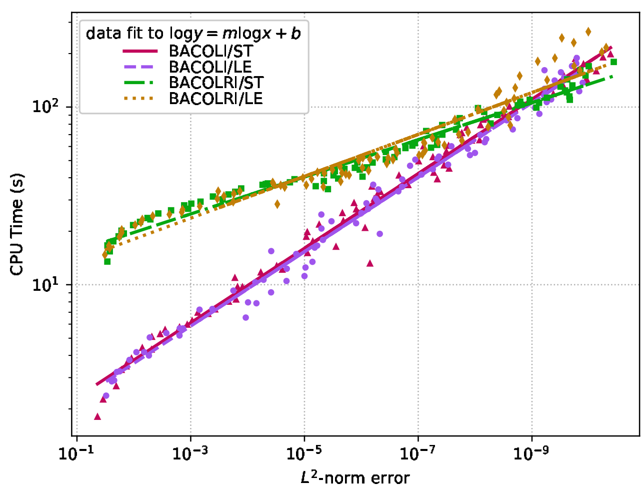

Figure 236: Work vs. Accuracy: Two Layer Burgers equation $\times 12$, $\epsilon = 10^{-4}, p = 7$

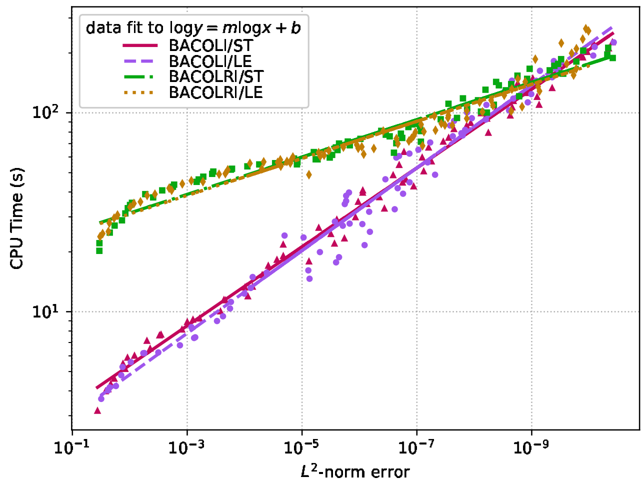

Figure 237: Work vs. Accuracy: Two Layer Burgers equation $\times 12$, $\epsilon = 10^{-4}, p = 8$

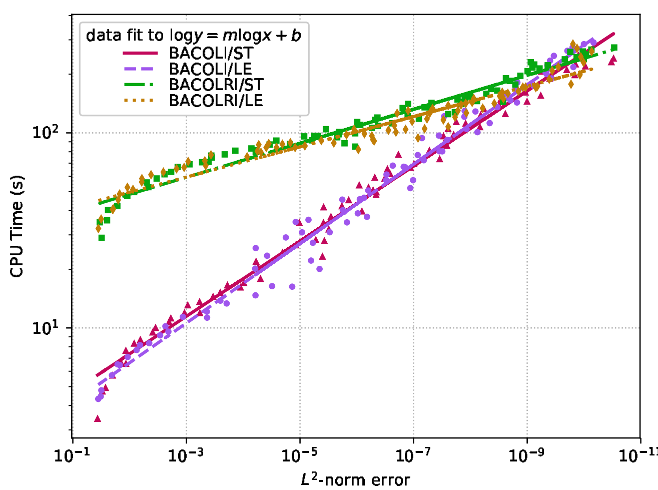

Figure 238: Work vs. Accuracy: Two Layer Burgers equation $\times 12$, $\epsilon = 10^{-4}, p = 9$

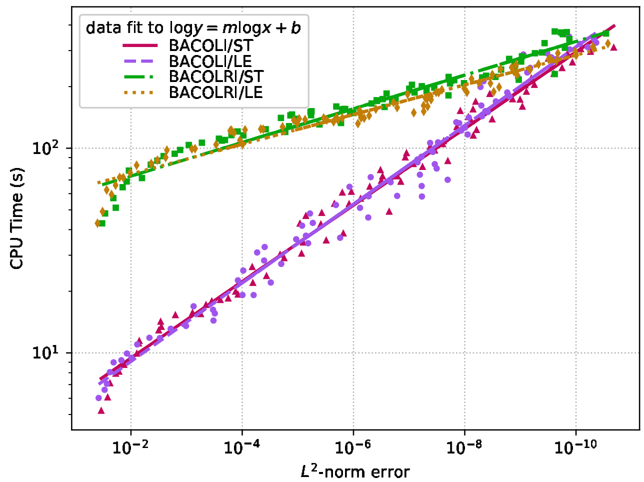

Figure 239: Work vs. Accuracy: Two Layer Burgers equation ×12, $\epsilon = 10^{-4}, p = 10$

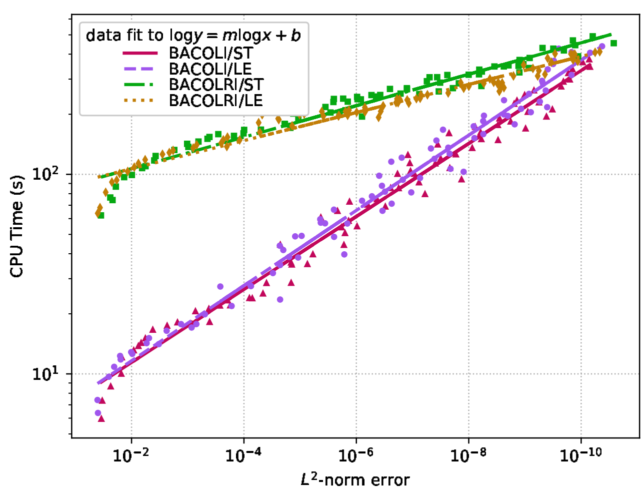

Figure 240: Work vs. Accuracy: Two Layer Burgers equation ×12, $\epsilon = 10^{-4}, p = 11$

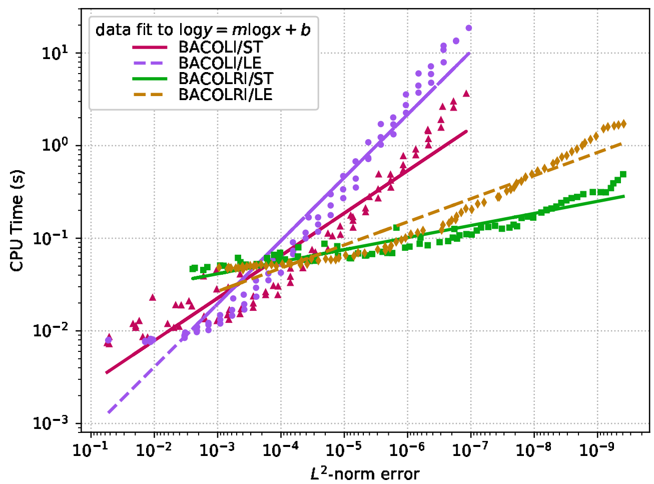

Figure 241: Work vs. Accuracy: Schrödinger System, $p = 4$

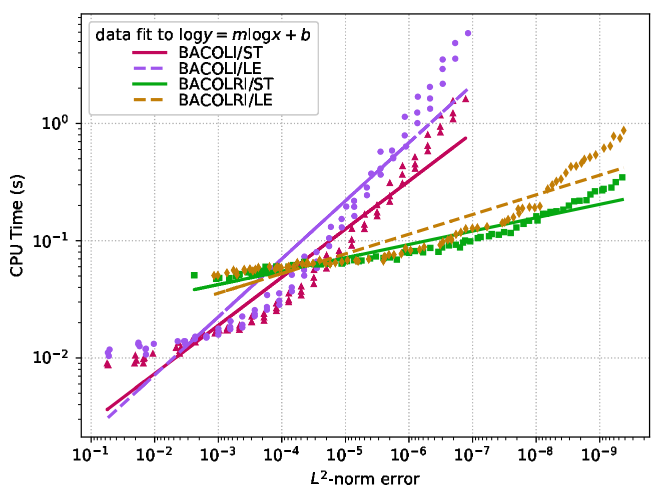

Figure 242: Work vs. Accuracy: Schrödinger System, $p = 5$

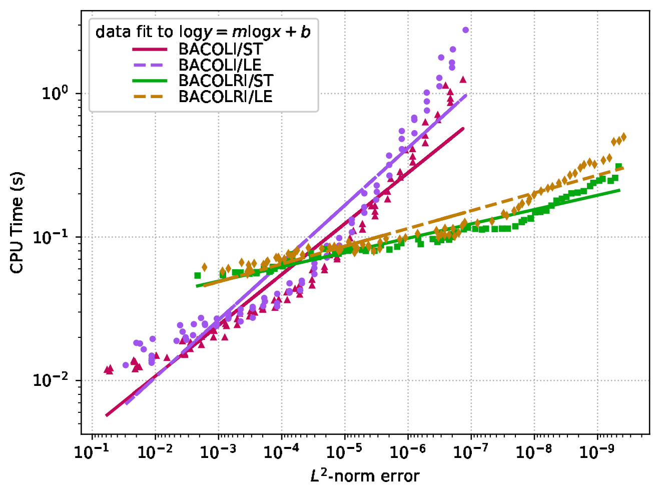

Figure 243: Work vs. Accuracy: Schrödinger System, $p = 6$

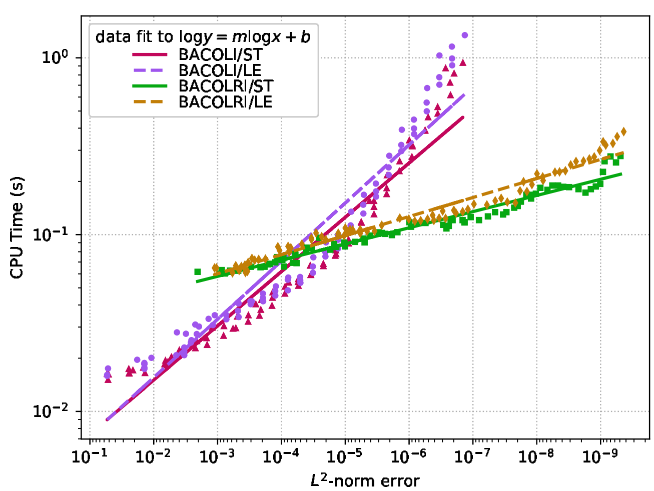

Figure 244: Work vs. Accuracy: Schrödinger System, $p = 7$

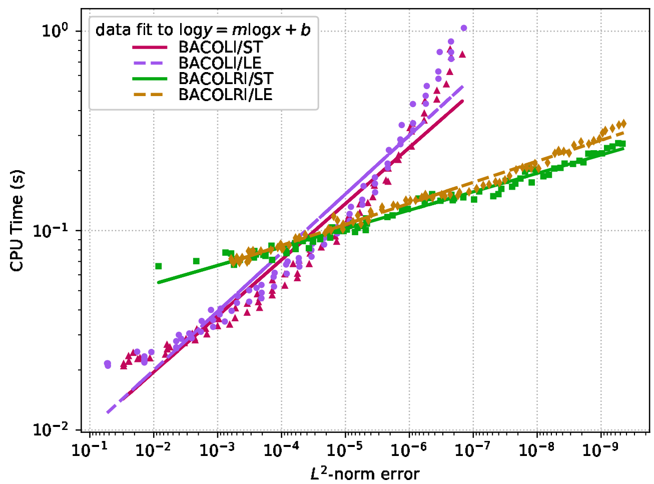

Figure 245: Work vs. Accuracy: Schrödinger System, $p = 8$

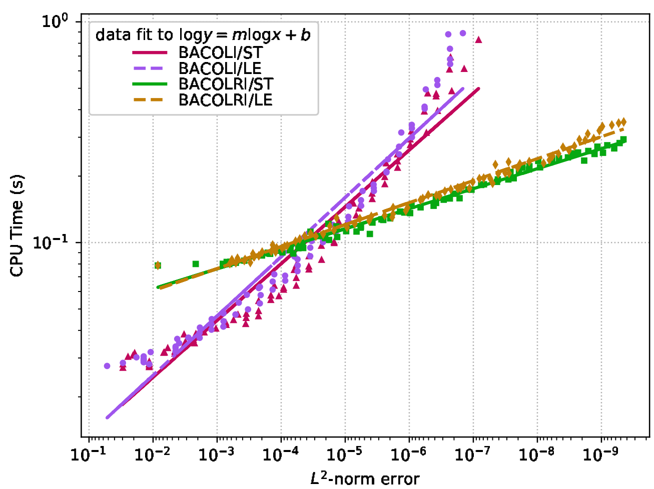

Figure 246: Work vs. Accuracy: Schrödinger System, $p = 9$

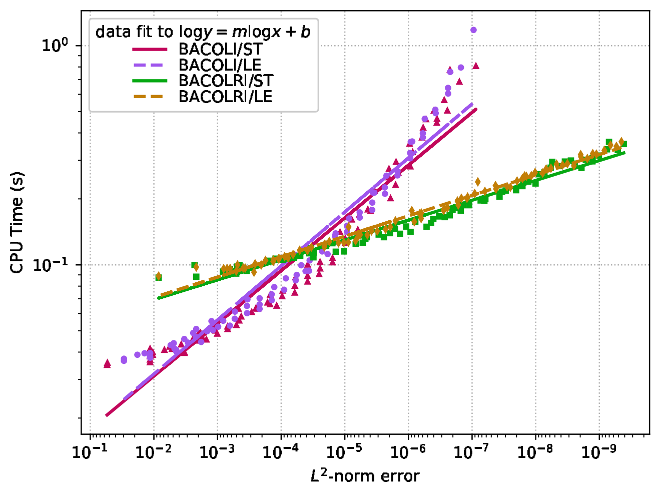

Figure 247: Work vs. Accuracy: Schrödinger System, $p = 10$

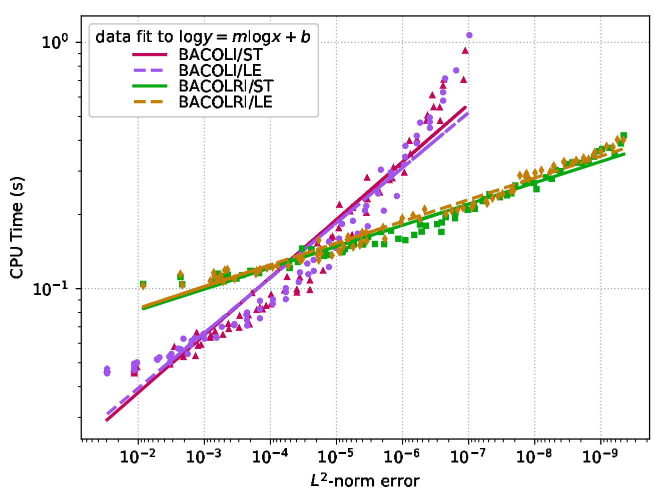

Figure 248: Work vs. Accuracy: Schrödinger System, $p = 11$

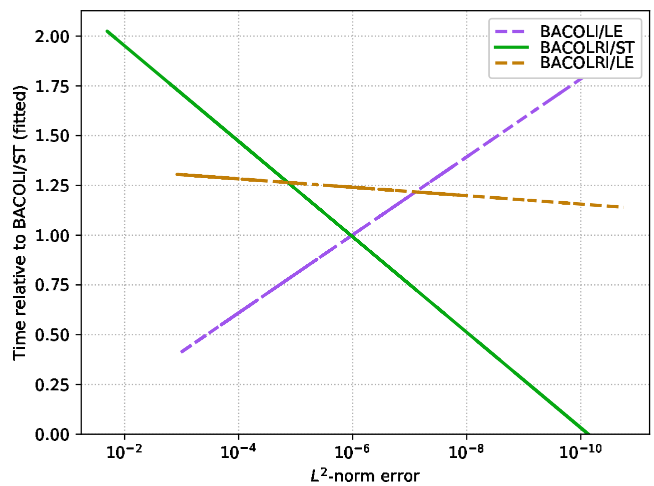

Figure 249: Rel. Work-Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 4$

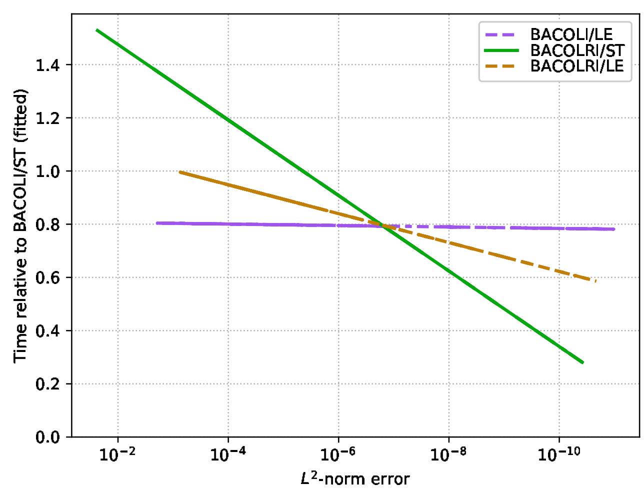

Figure 250: Rel. Work-Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 5$

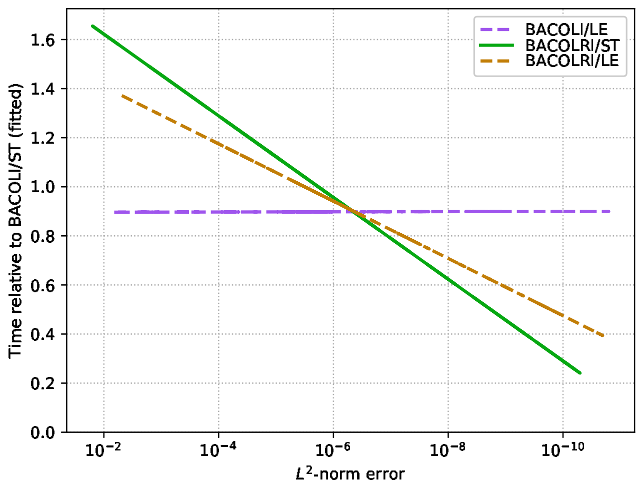

Figure 251: Rel. Work-Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 6$

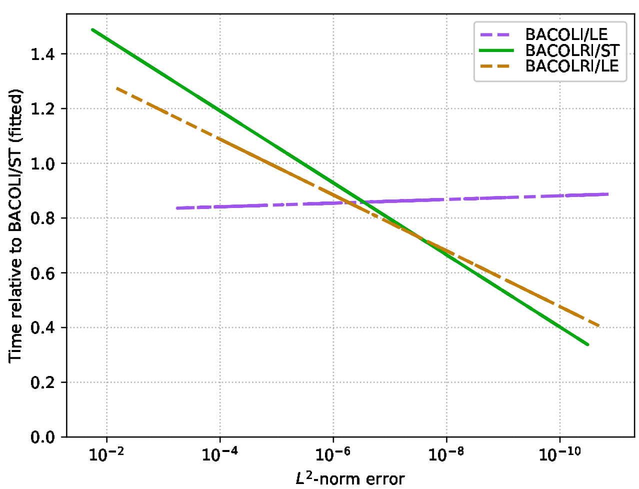

Figure 252: Rel. Work-Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 7$

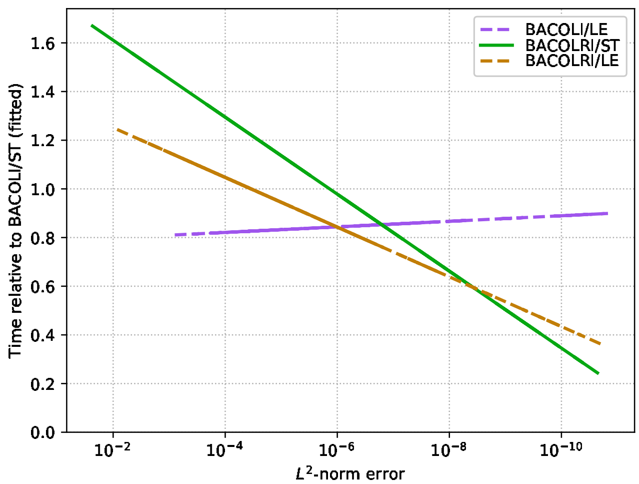

Figure 253: Rel. Work-Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 8$

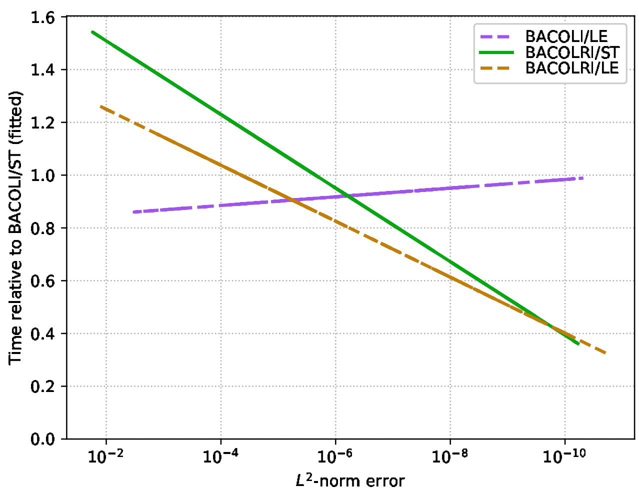

Figure 254: Rel. Work-Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 9$

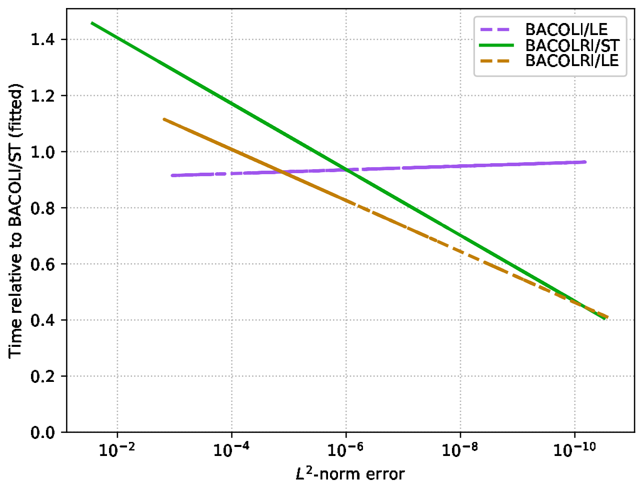

Figure 255: Rel. Work-Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 10$

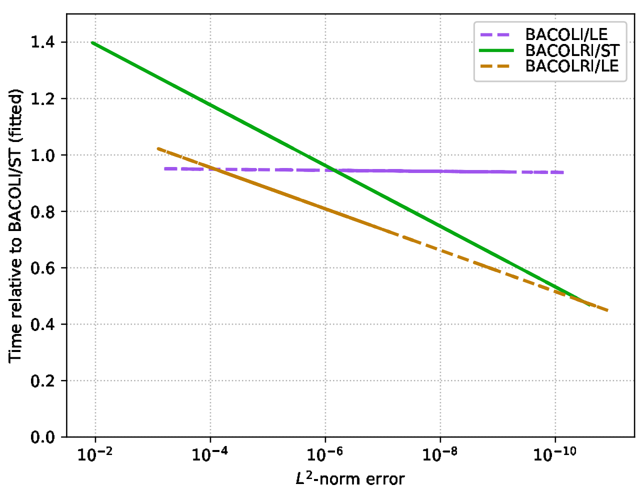

Figure 256: Rel. Work-Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 11$

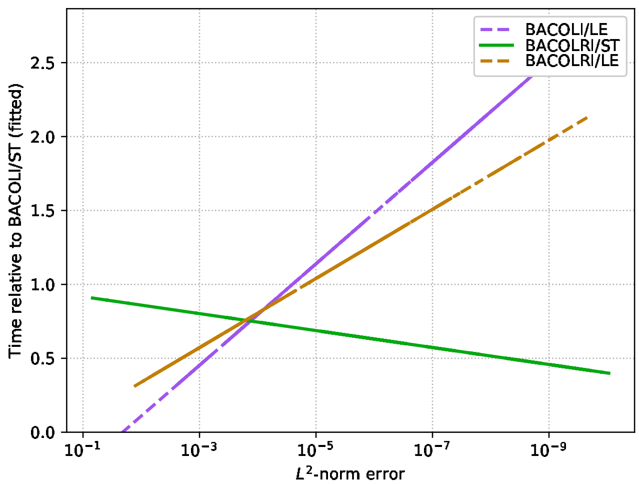

Figure 257: Rel. Work-Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 4$

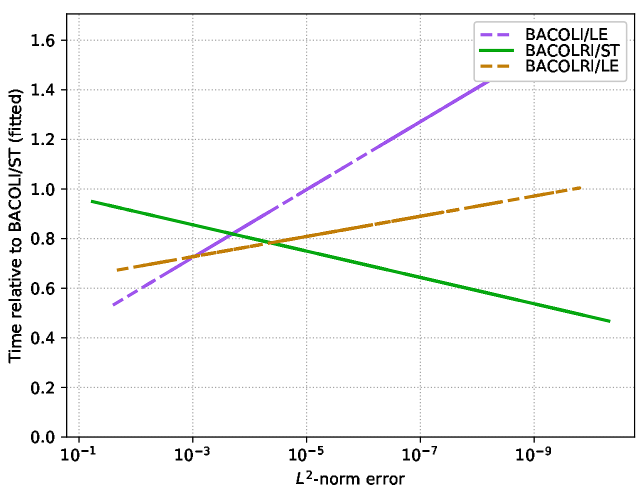

Figure 258: Rel. Work-Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 5$

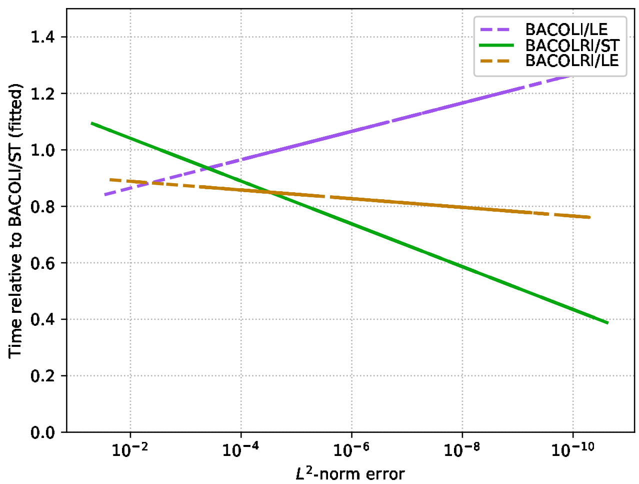

Figure 259: Rel. Work-Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 6$

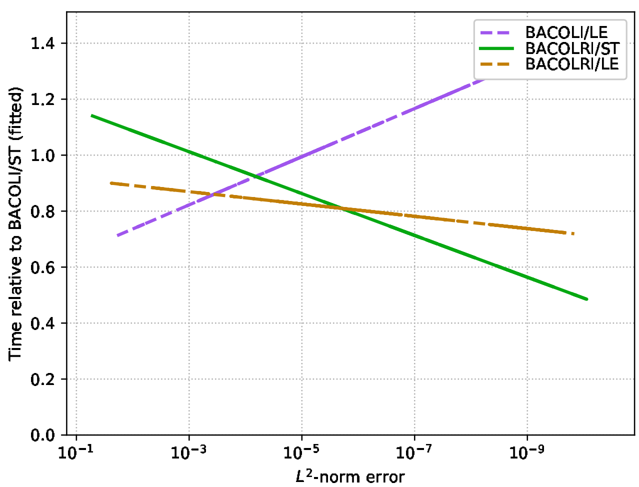

Figure 260: Rel. Work-Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 7$

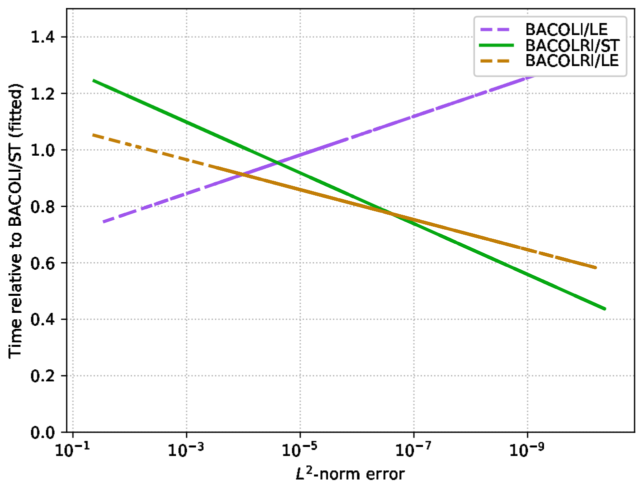

Figure 261: Rel. Work-Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 8$

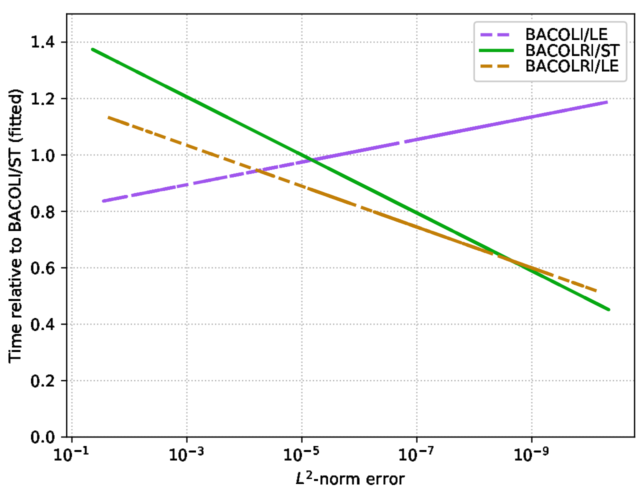

Figure 262: Rel. Work-Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 9$

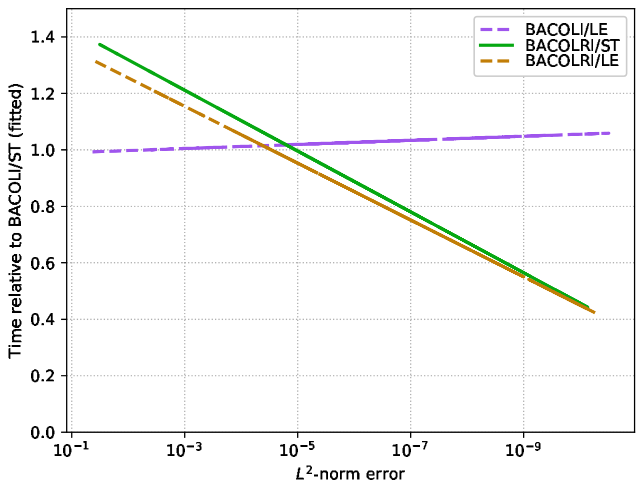

Figure 263: Rel. Work-Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 10$

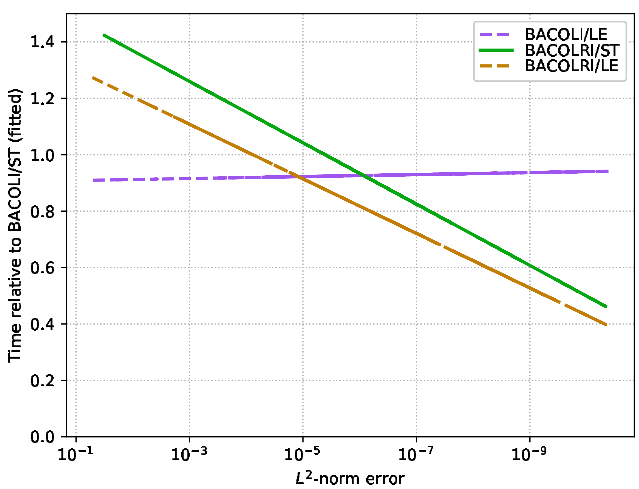

Figure 264: Rel. Work-Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 11$

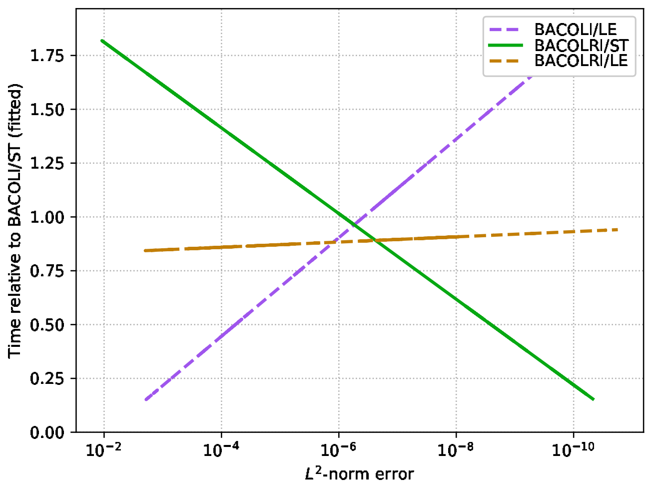

Figure 265: Rel. Work-Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 4$

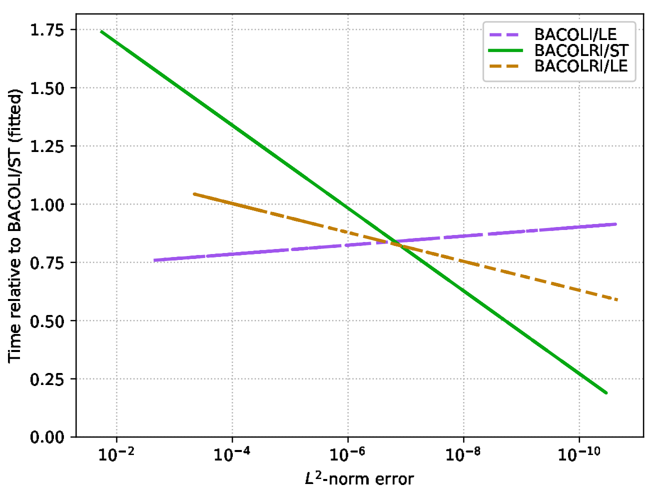

Figure 266: Rel. Work-Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 5$

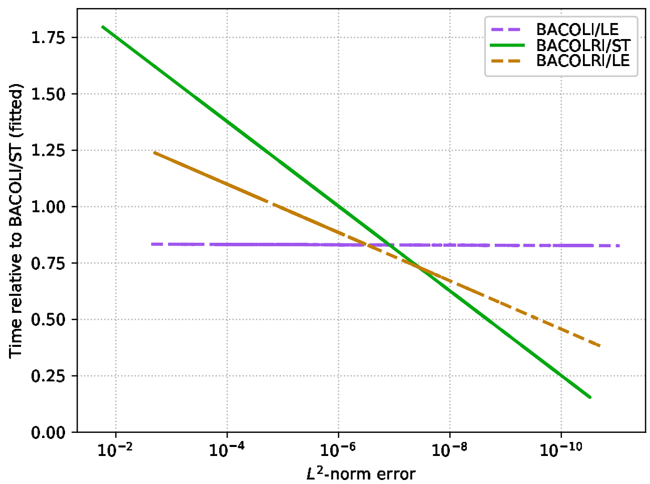

Figure 267: Rel. Work-Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 6$

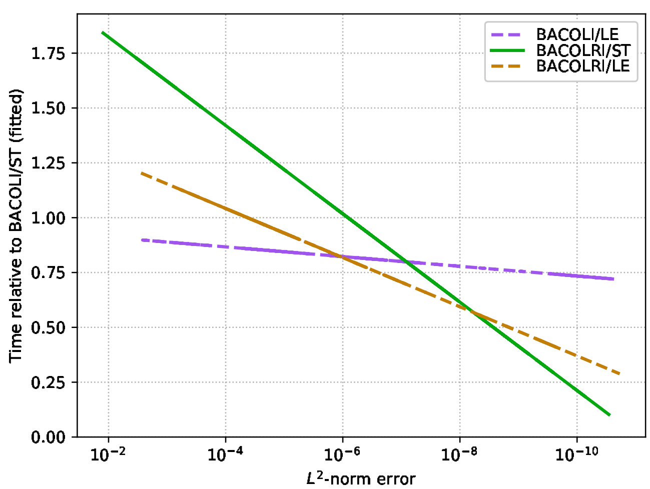

Figure 268: Rel. Work-Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 7$

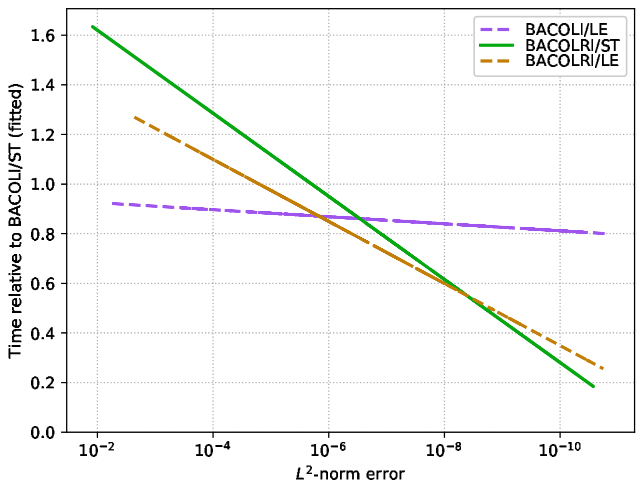

Figure 269: Rel. Work-Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 8$

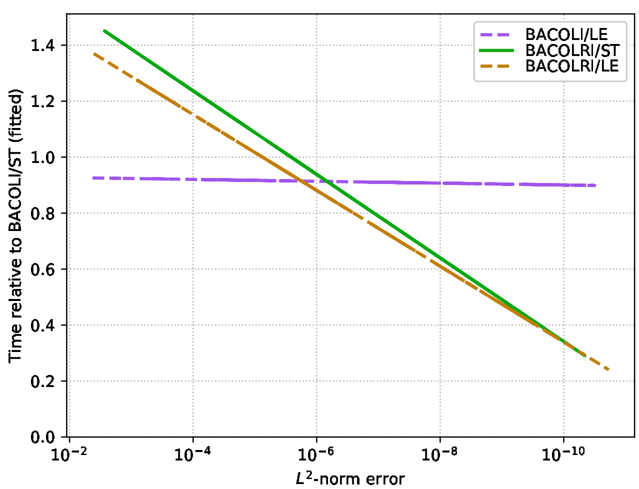

Figure 270: Rel. Work-Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 9$

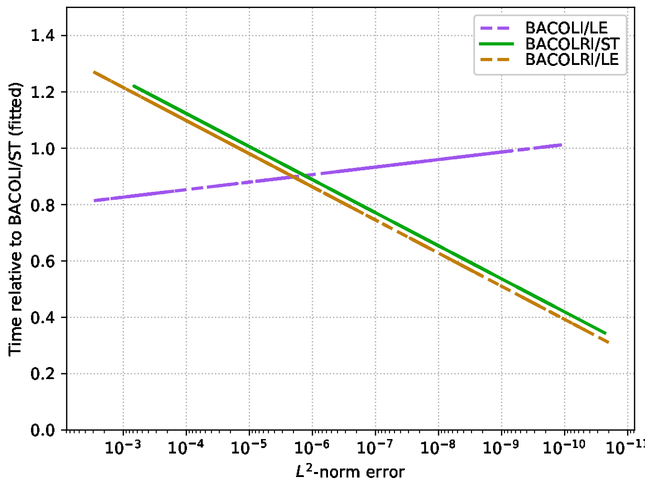

Figure 271: Rel. Work-Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 10$

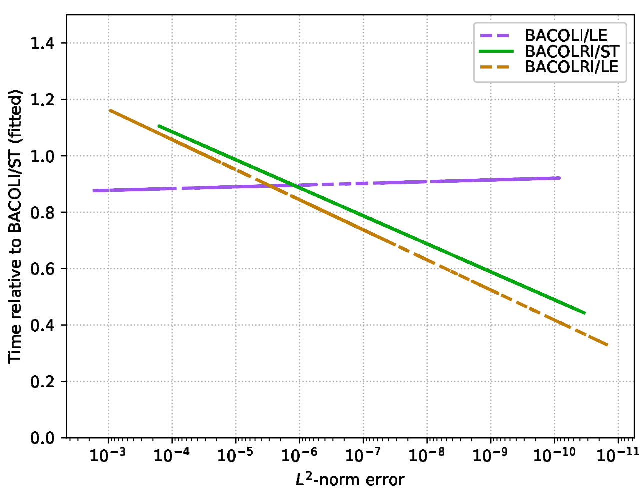

Figure 272: Rel. Work-Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 11$

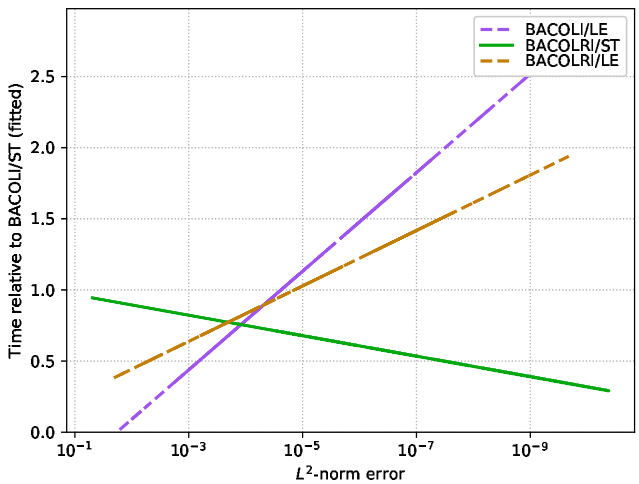

Figure 273: Rel. Work-Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 4$

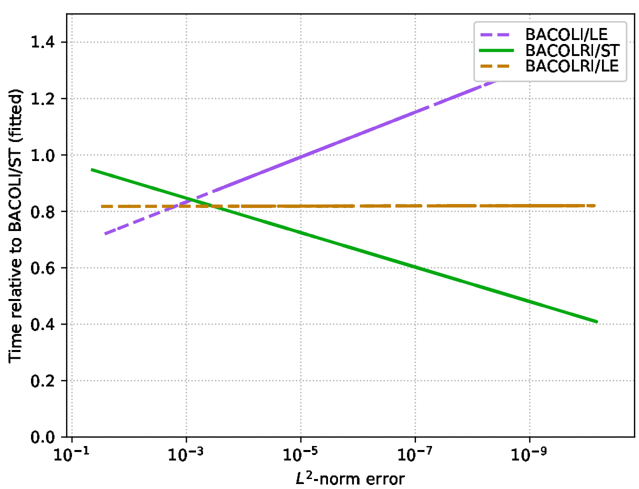

Figure 274: Rel. Work-Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 5$

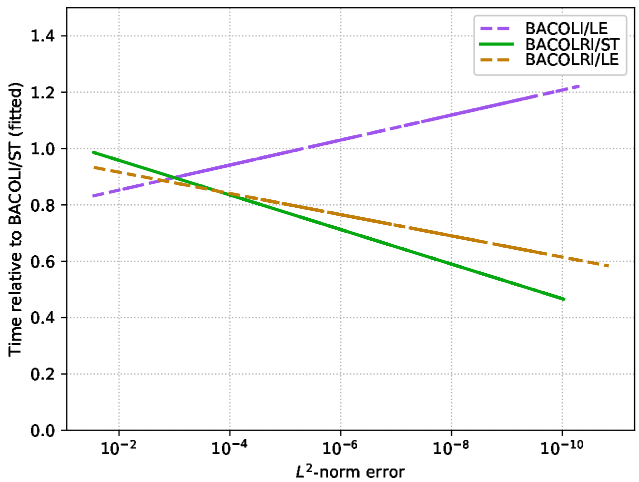

Figure 275: Rel. Work-Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 6$

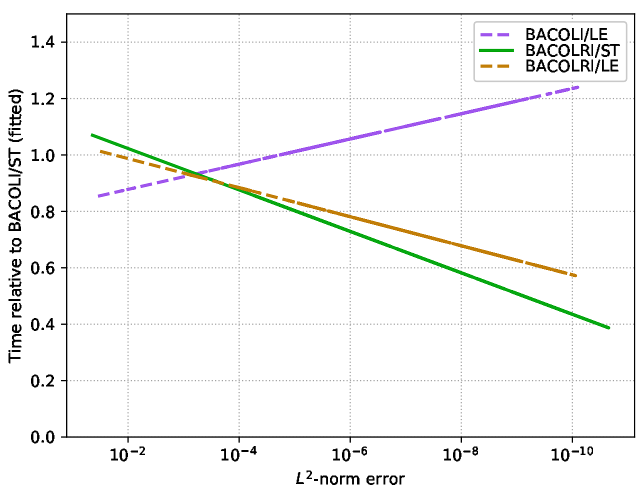

Figure 276: Rel. Work-Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 7$

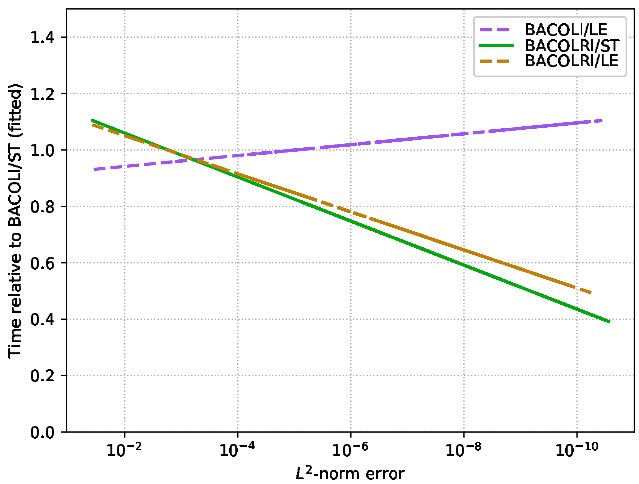

Figure 277: Rel. Work-Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 8$

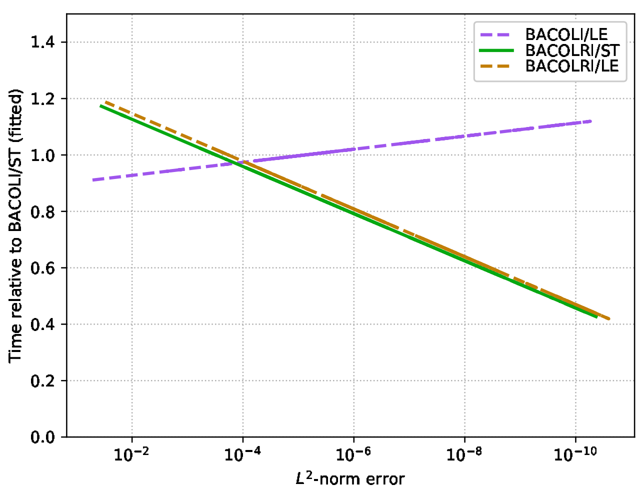

Figure 278: Rel. Work-Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 9$

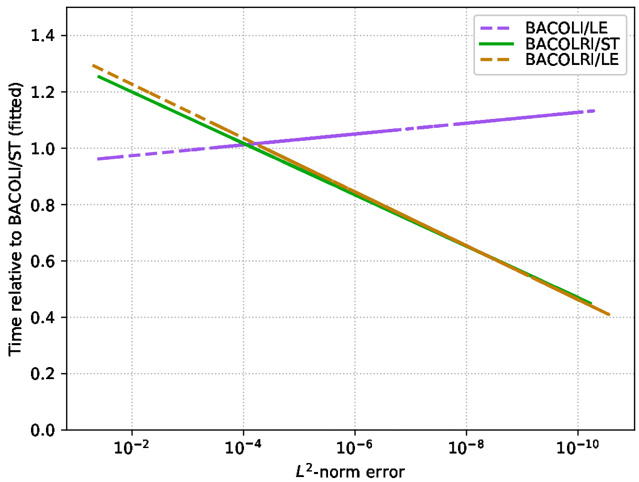

Figure 279: Rel. Work-Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 10$

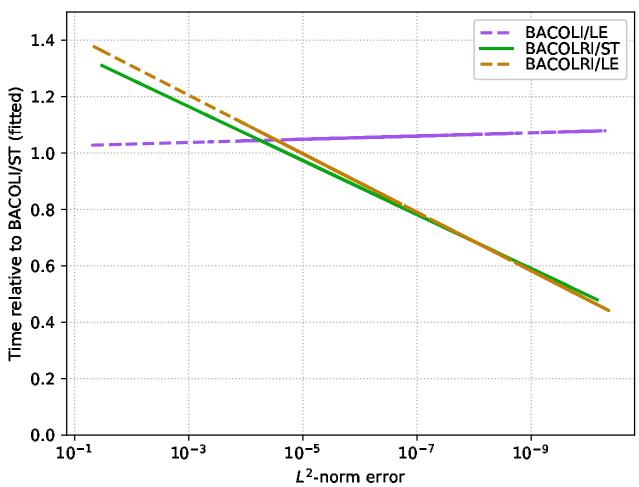

Figure 280: Rel. Work-Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 11$

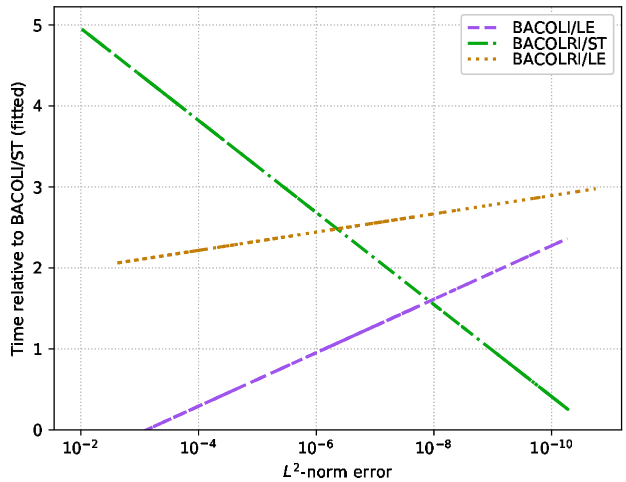

Figure 281: Rel. Work-Accuracy: Two Layer Burgers equation $\times 12$, $\epsilon = 10^{-3}, p = 4$

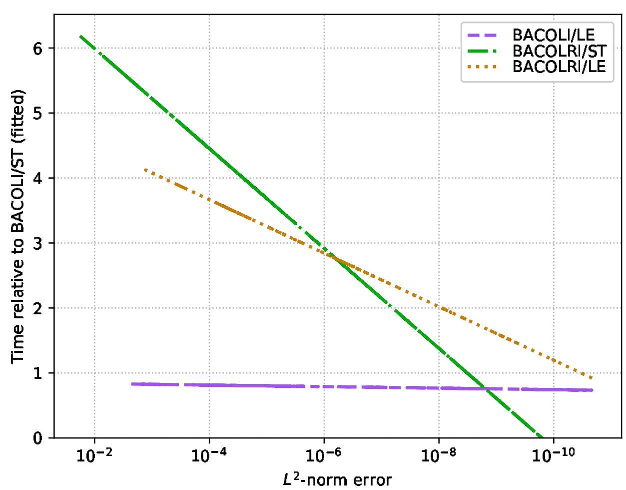

Figure 282: Rel. Work-Accuracy: Two Layer Burgers equation $\times 12$, $\epsilon = 10^{-3}, p = 5$

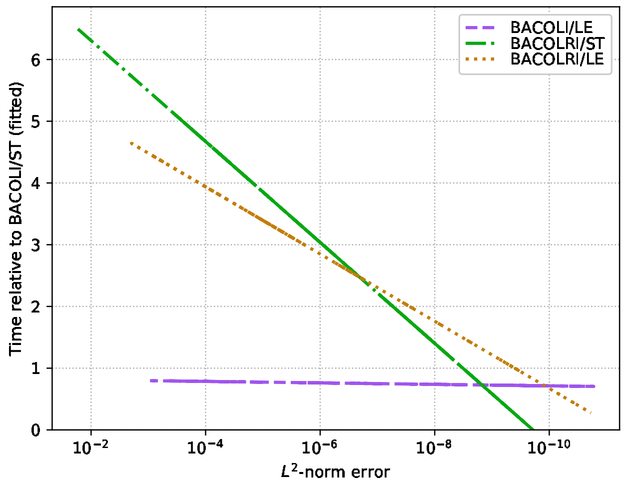

Figure 283: Rel. Work-Accuracy: Two Layer Burgers equation ×12, $\epsilon = 10^{-3}, p = 6$

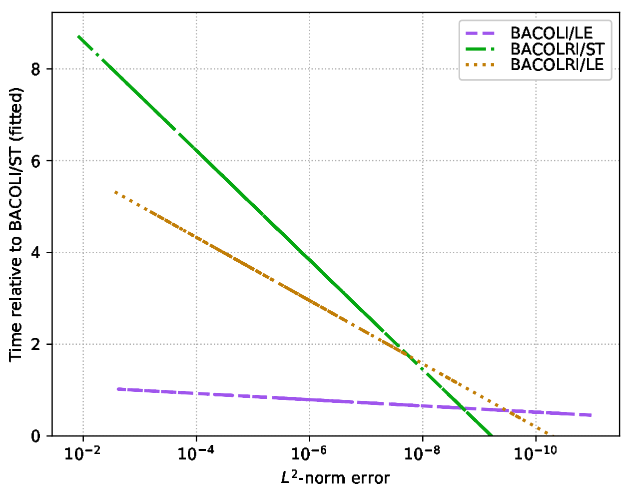

Figure 284: Rel. Work-Accuracy: Two Layer Burgers equation ×12, $\epsilon = 10^{-3}, p = 7$

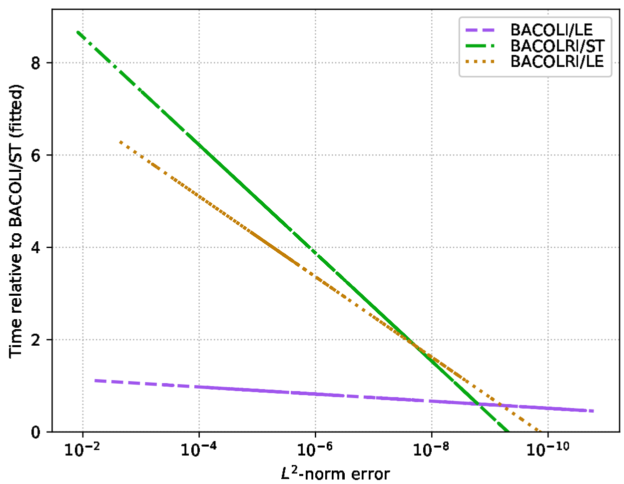

Figure 285: Rel. Work-Accuracy: Two Layer Burgers equation $\times 12$, $\epsilon = 10^{-3}, p = 8$

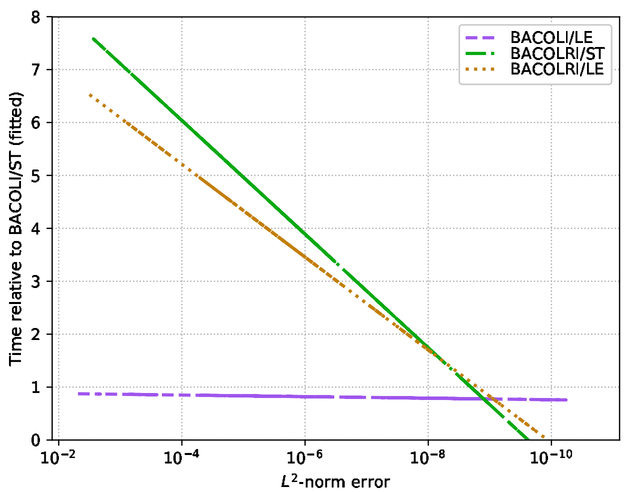

Figure 286: Rel. Work-Accuracy: Two Layer Burgers equation $\times 12$, $\epsilon = 10^{-3}, p = 9$

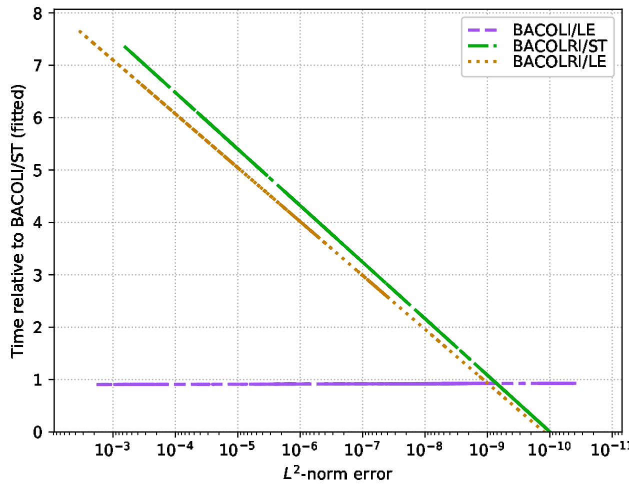

Figure 287: Rel. Work-Accuracy: Two Layer Burgers equation ×12, $\epsilon = 10^{-3}, p = 10$

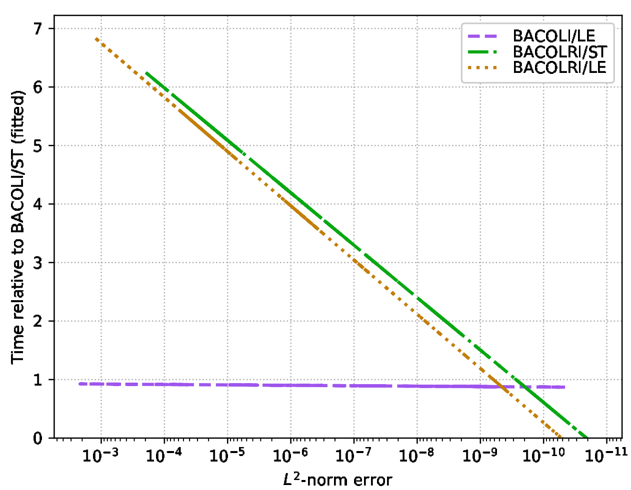

Figure 288: Rel. Work-Accuracy: Two Layer Burgers equation ×12, $\epsilon = 10^{-3}, p = 11$

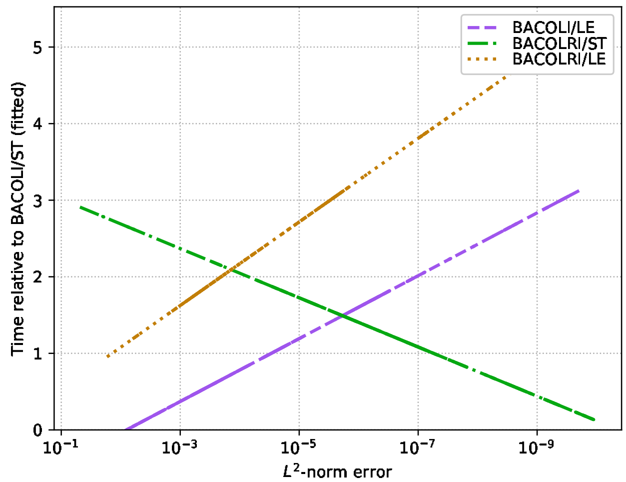

Figure 289: Rel. Work-Accuracy: Two Layer Burgers equation $\times 12$, $\epsilon = 10^{-4}, p = 4$

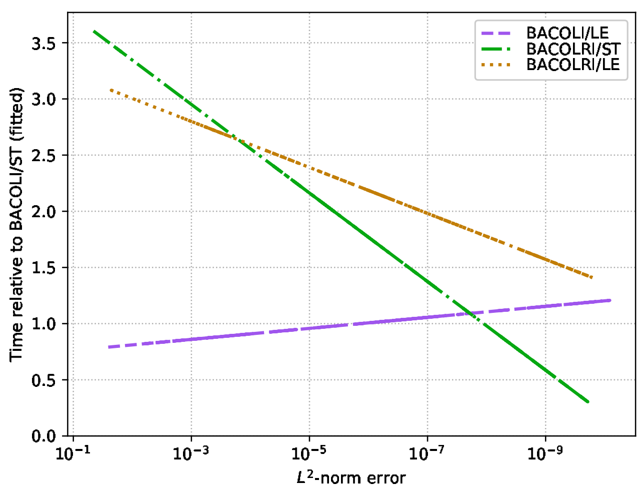

Figure 290: Rel. Work-Accuracy: Two Layer Burgers equation $\times 12$, $\epsilon = 10^{-4}, p = 5$

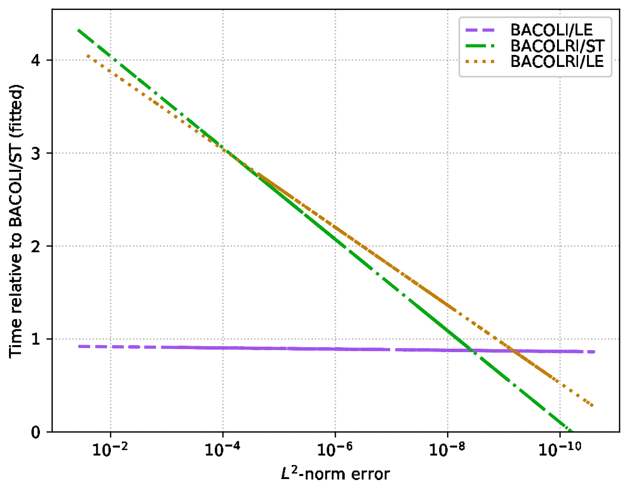

Figure 291: Rel. Work-Accuracy: Two Layer Burgers equation $\times 12$, $\epsilon = 10^{-4}, p = 6$

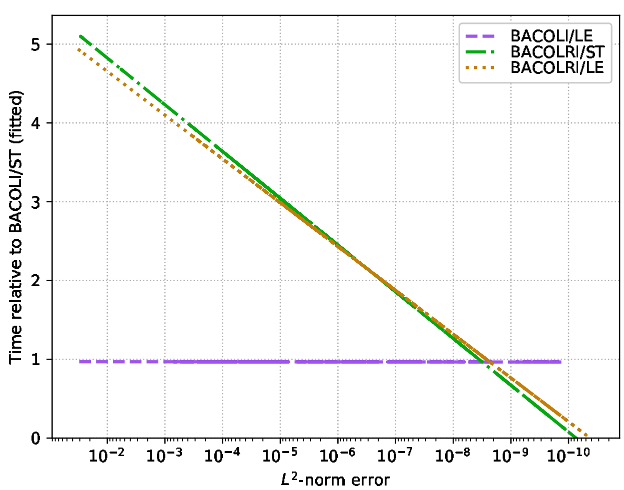

Figure 292: Rel. Work-Accuracy: Two Layer Burgers equation $\times 12$, $\epsilon = 10^{-4}, p = 7$

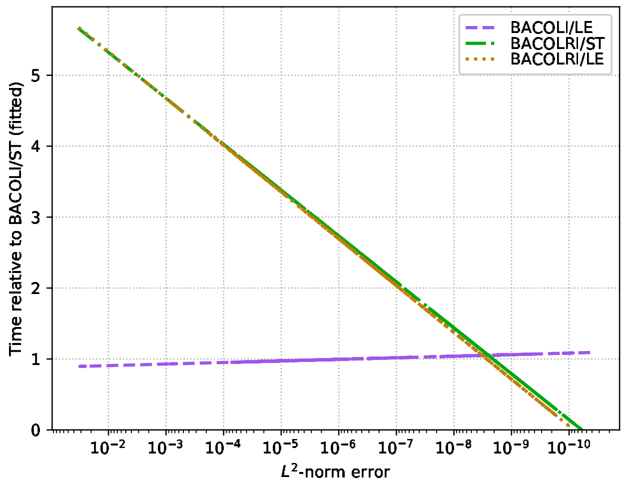

Figure 293: Rel. Work-Accuracy: Two Layer Burgers equation $\times 12$, $\epsilon = 10^{-4}, p = 8$

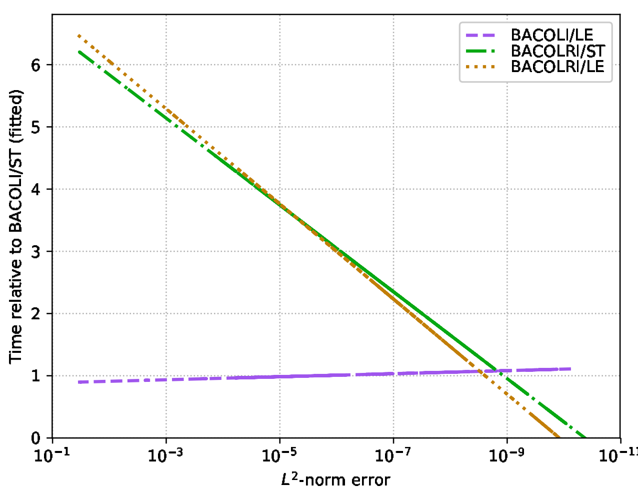

Figure 294: Rel. Work-Accuracy: Two Layer Burgers equation $\times 12$, $\epsilon = 10^{-4}, p = 9$

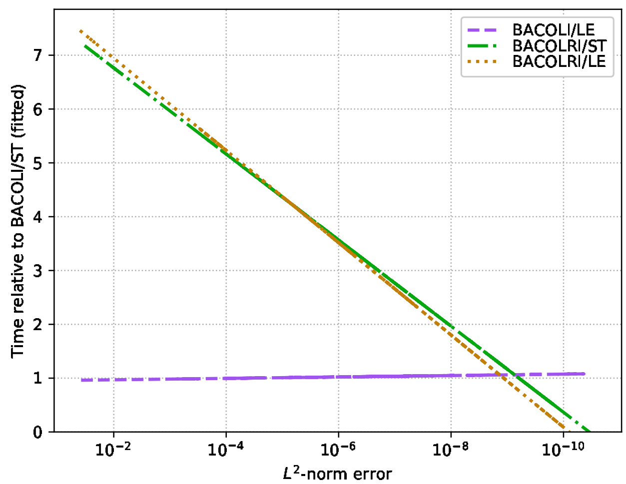

Figure 295: Rel. Work-Accuracy: Two Layer Burgers equation $\times 12$, $\epsilon = 10^{-4}, p = 10$

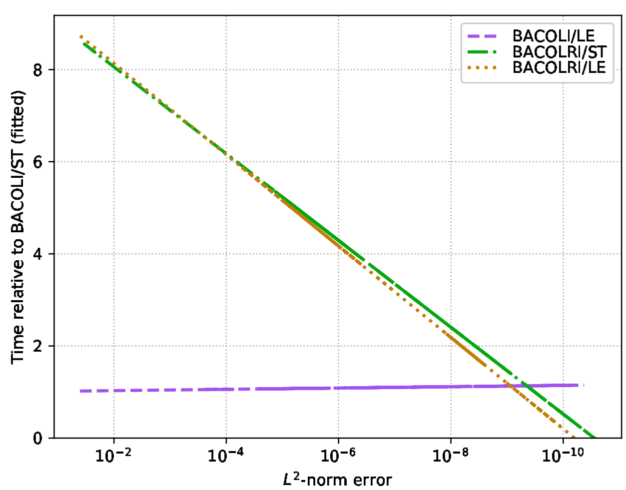

Figure 296: Rel. Work-Accuracy: Two Layer Burgers equation $\times 12$, $\epsilon = 10^{-4}, p = 11$

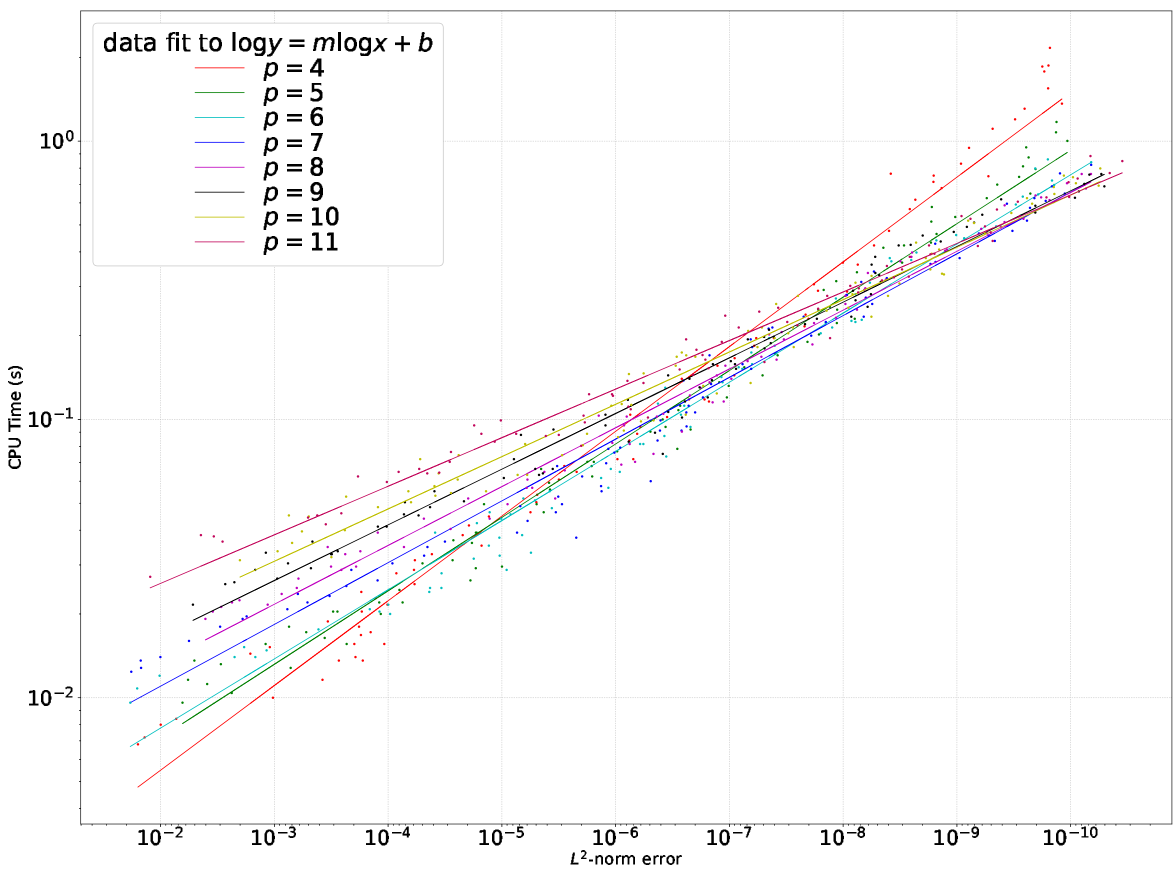

Figure 297: BACOLI/ST Work vs. Accuracy: One Layer Burgers equation $\epsilon = 10^{-3}$; $p = 4 \ldots 11$

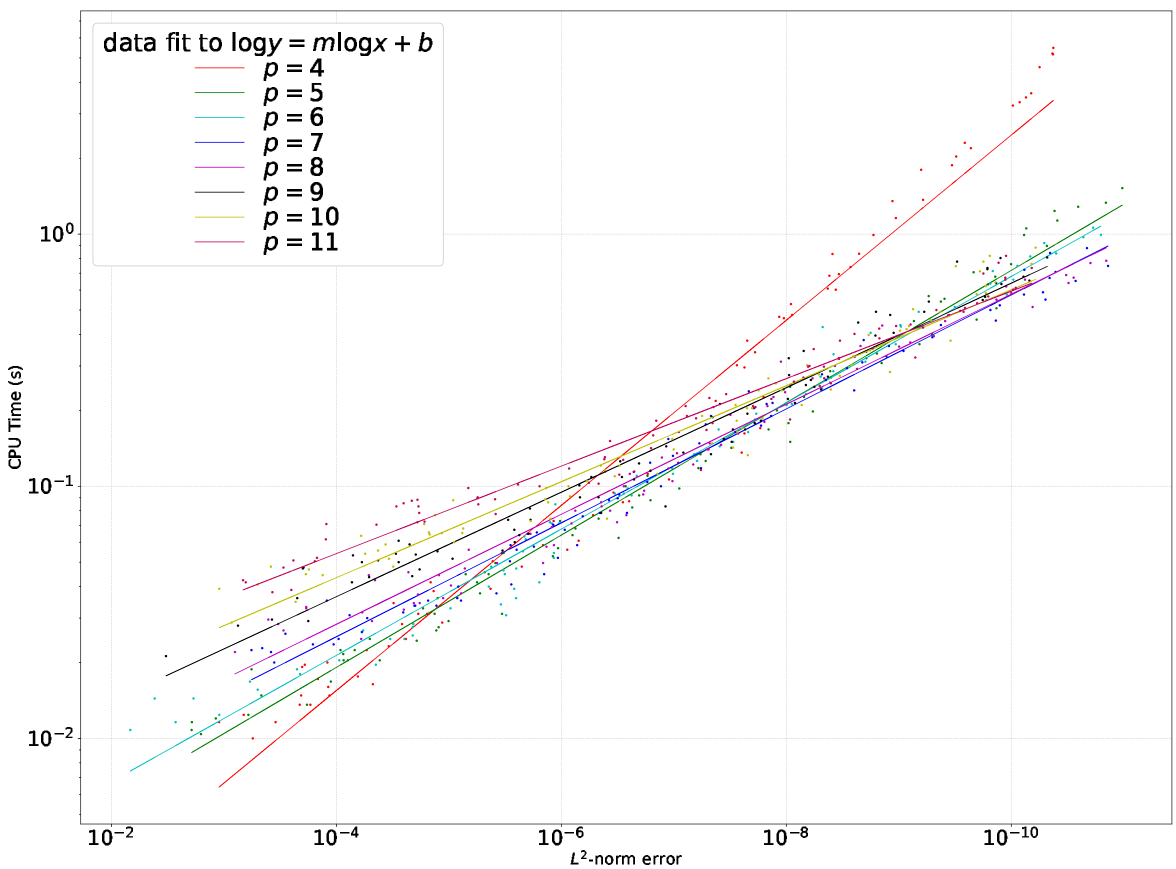

Figure 298: BACOLI/LE Work vs. Accuracy: One Layer Burgers equation $\epsilon = 10^{-3}$; $p = 4 \ldots 11$

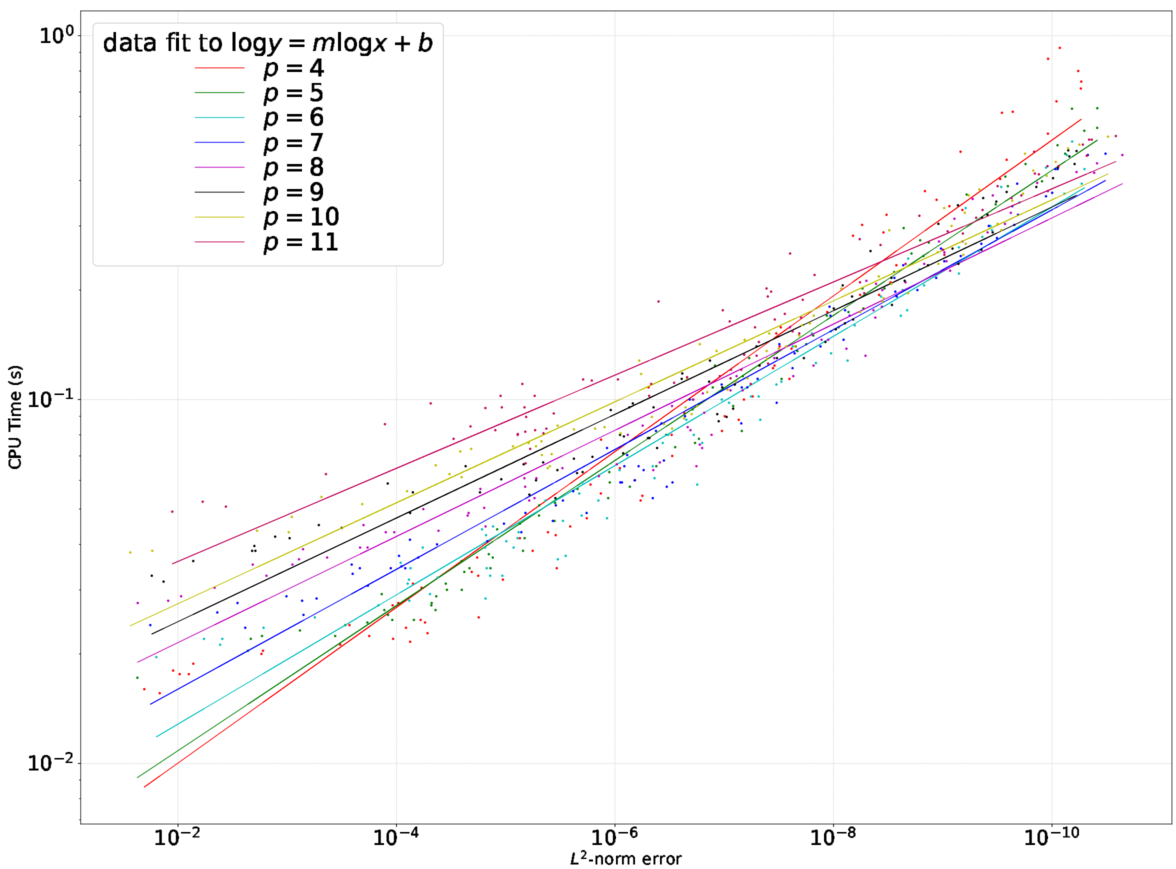

Figure 299: BACOLRI/ST Work vs. Accuracy: One Layer Burgers equation $\epsilon = 10^{-3}$; $p = 4 \ldots 11$

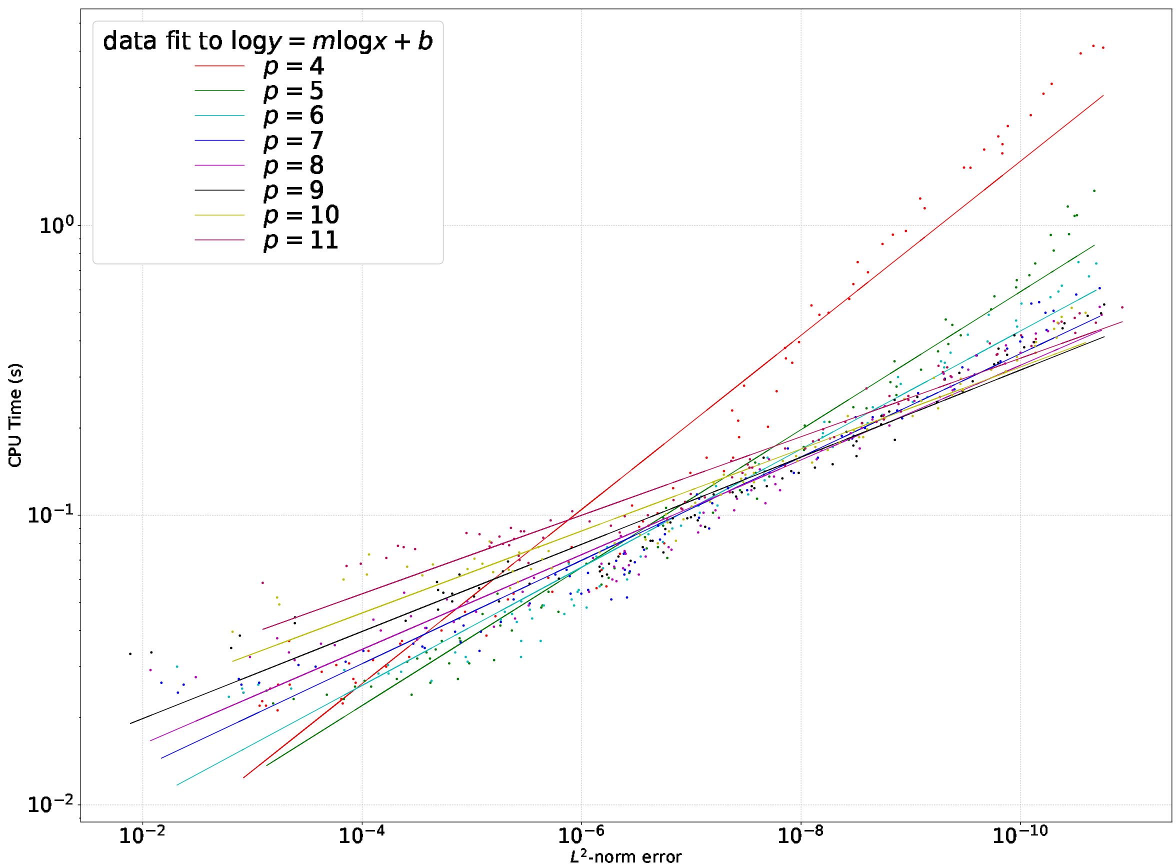

Figure 300: BACOLRI/LE Work vs. Accuracy: One Layer Burgers equation $\epsilon = 10^{-3}$; $p = 4 \ldots 11$

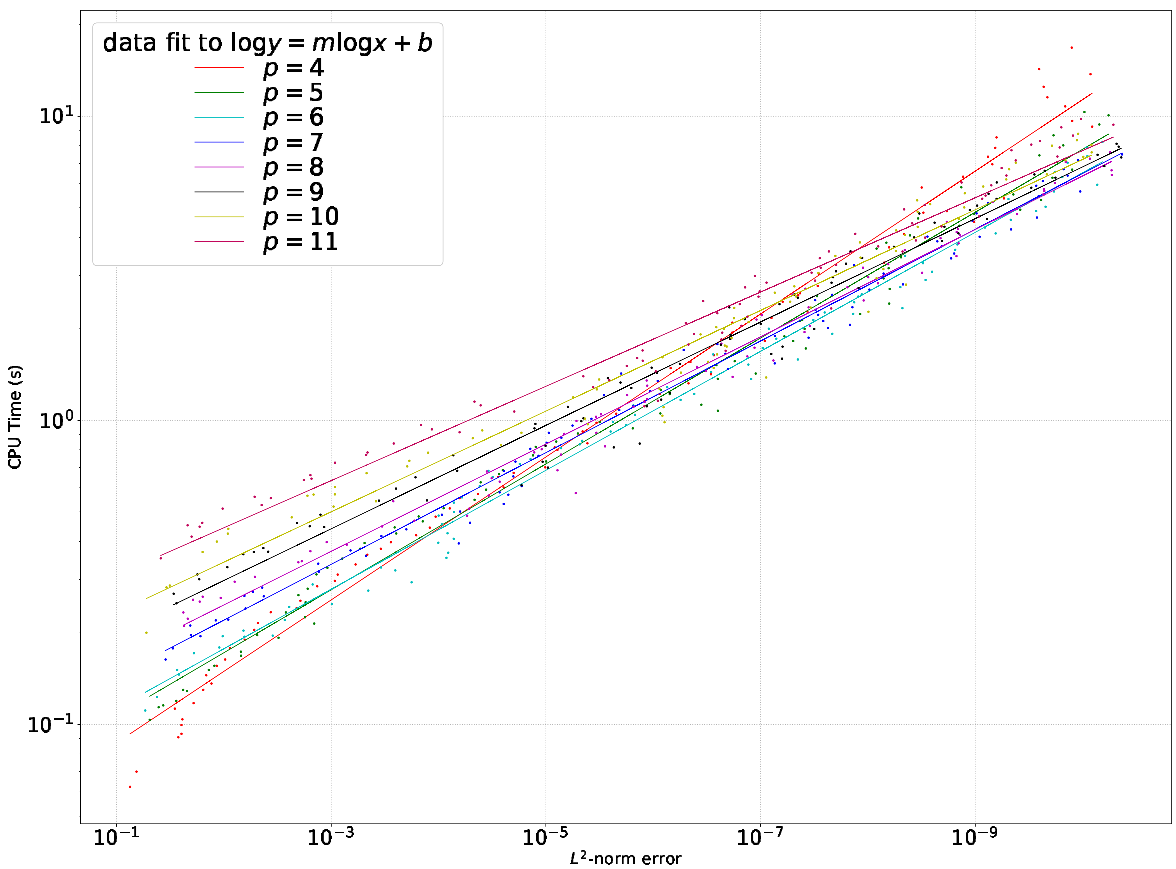

Figure 301: BACOLI/ST Work vs. Accuracy: One Layer Burgers equation $\epsilon = 10^{-4}$; $p = 4 \ldots 11$

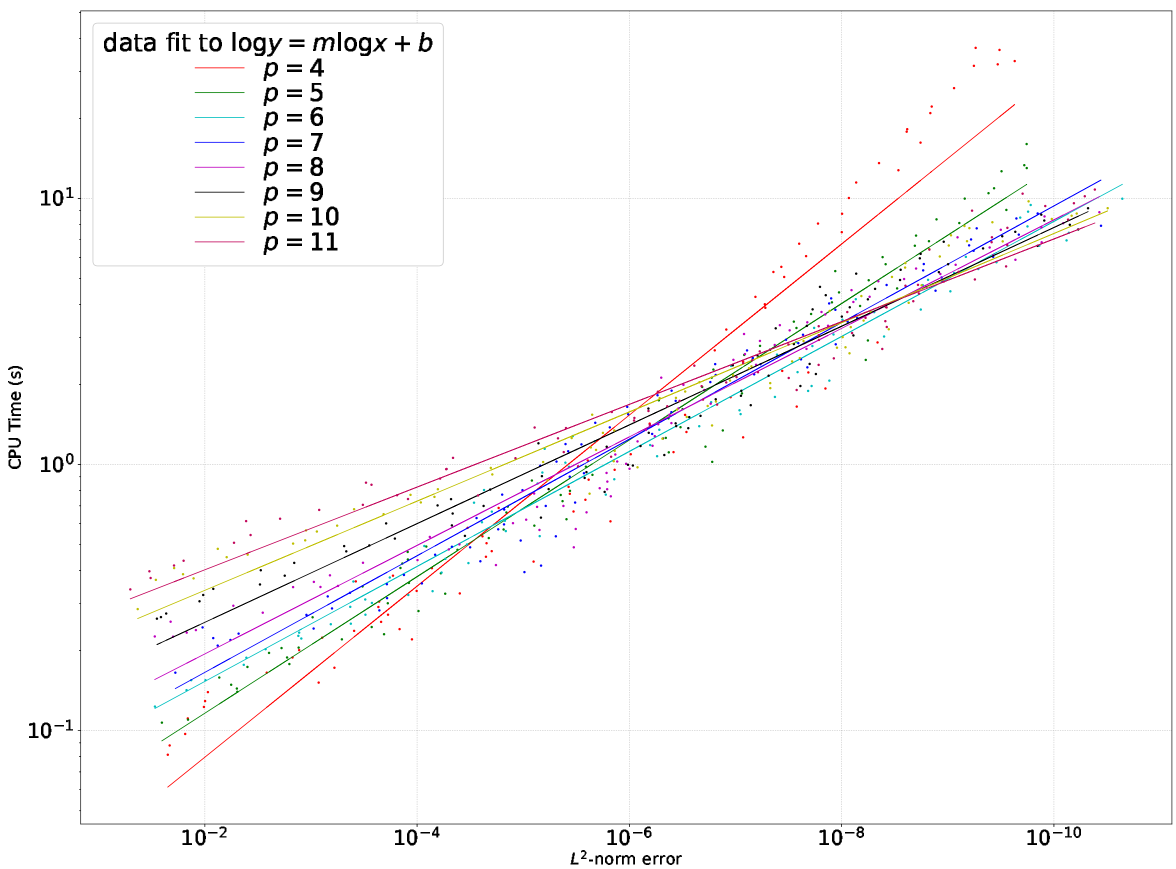

Figure 302: BACOLI/LE Work vs. Accuracy: One Layer Burgers equation $\epsilon = 10^{-4}$; $p = 4 \ldots 11$

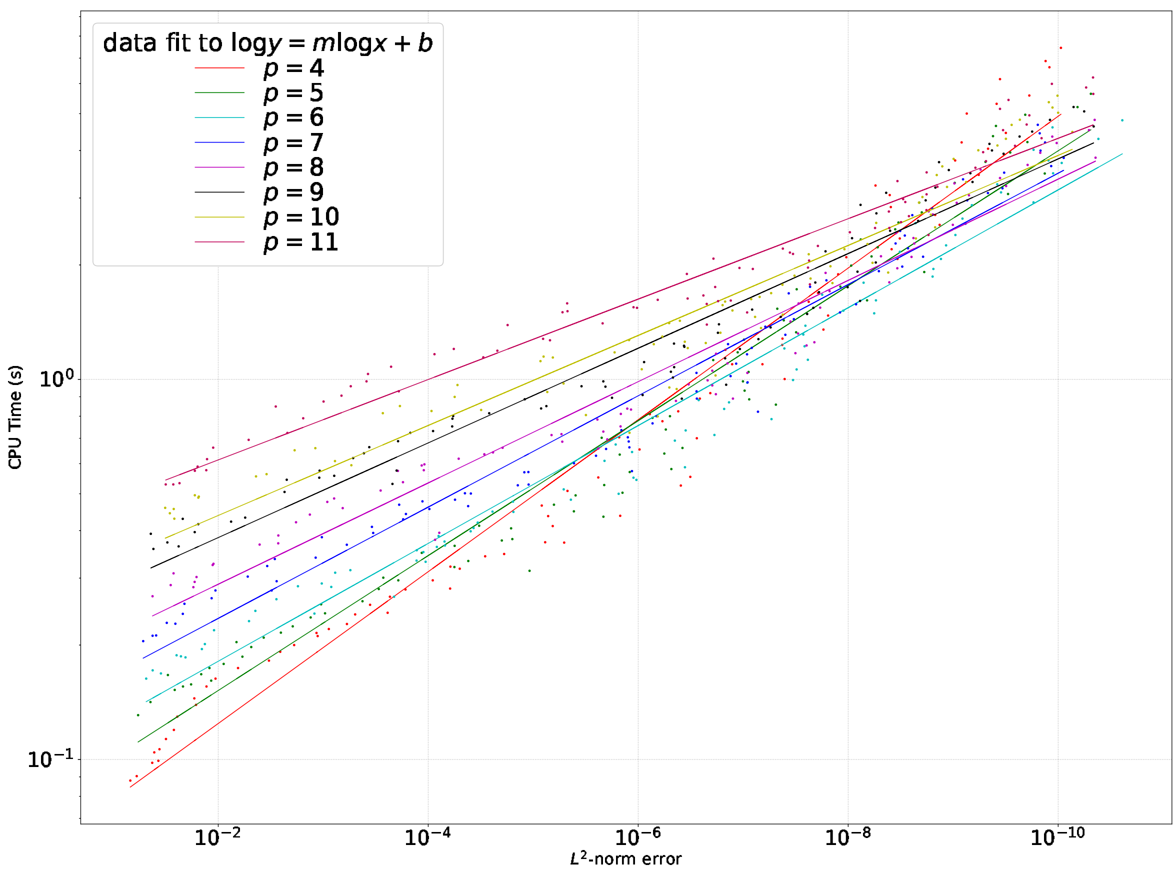

Figure 303: BACOLRI/ST Work vs. Accuracy: One Layer Burgers equation $\epsilon = 10^{-4}$; $p = 4 \ldots 11$

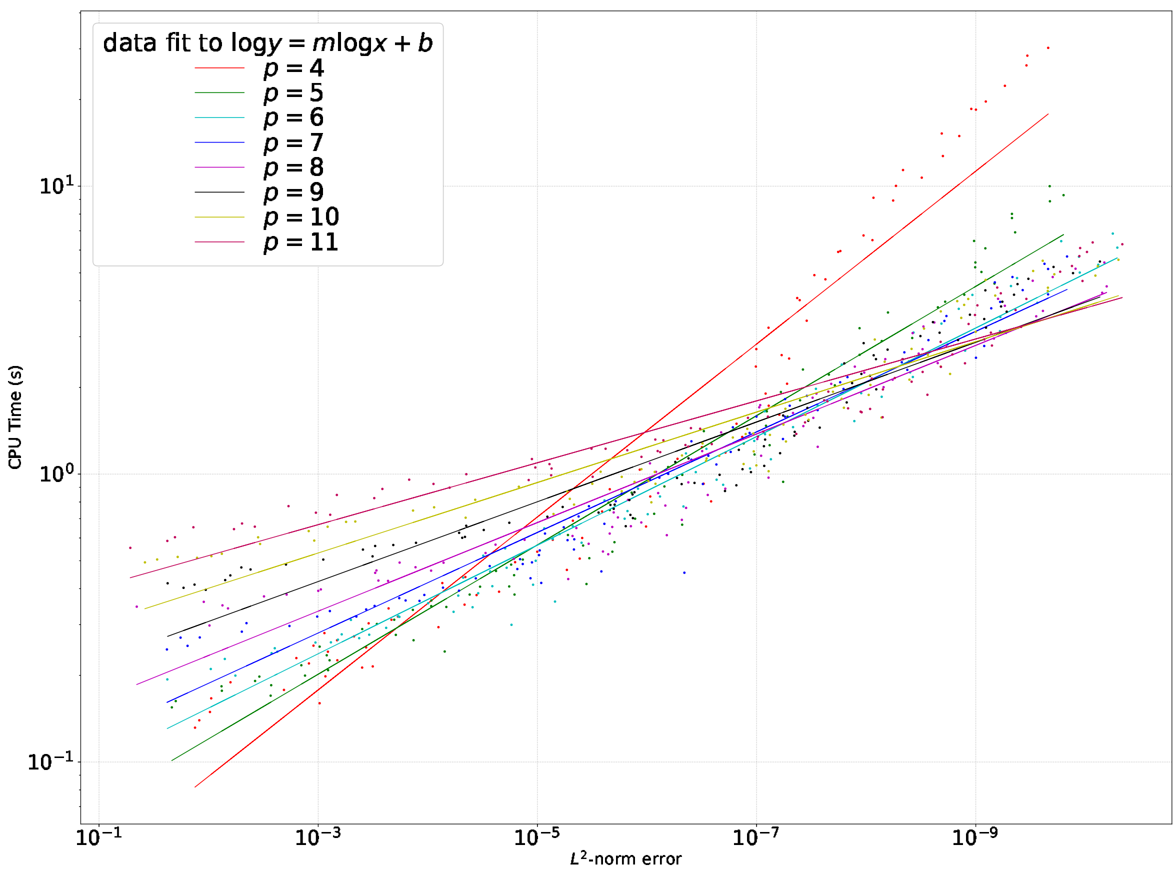

Figure 304: BACOLRI/LE Work vs. Accuracy: One Layer Burgers equation $\epsilon = 10^{-4}$; $p = 4 \ldots 11$

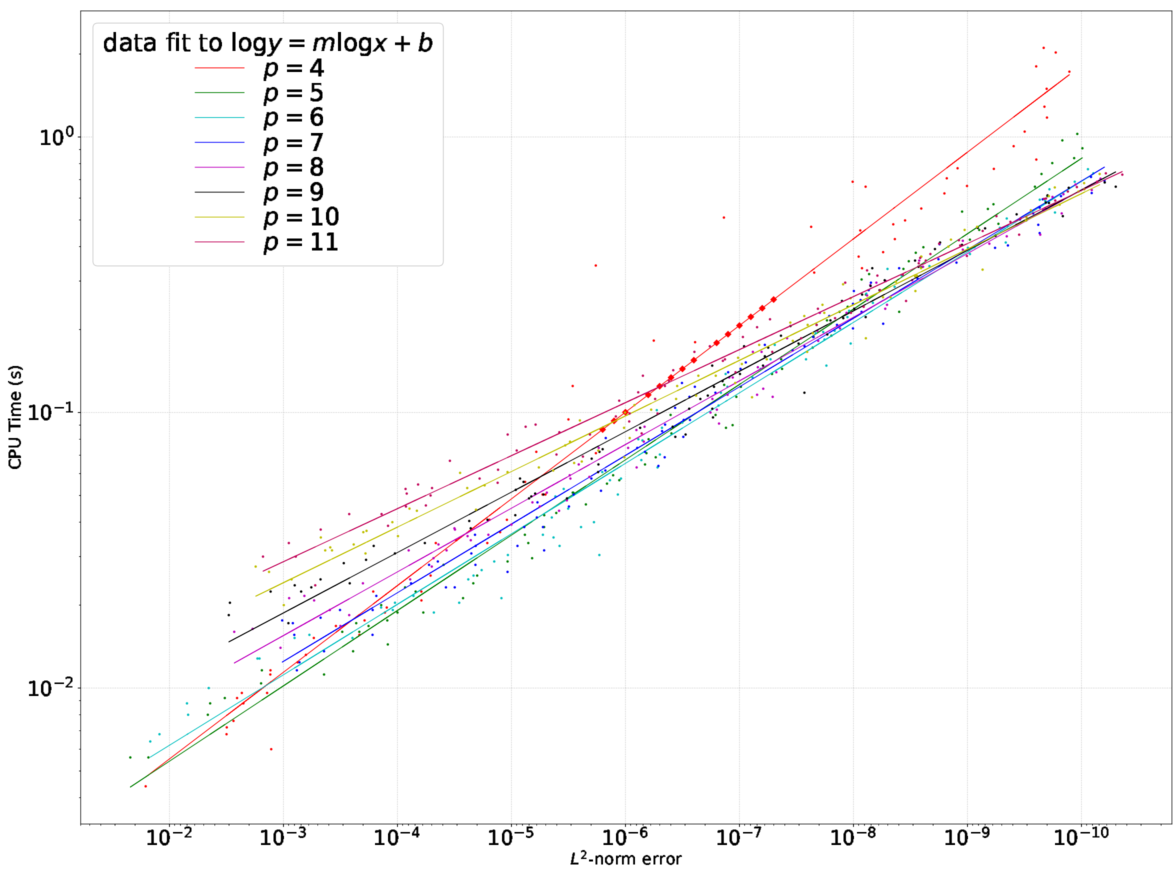

Figure 305: BACOLI/ST Work vs. Accuracy: Two Layer Burgers equation $\epsilon = 10^{-3}$; $p = 4 \ldots 11$

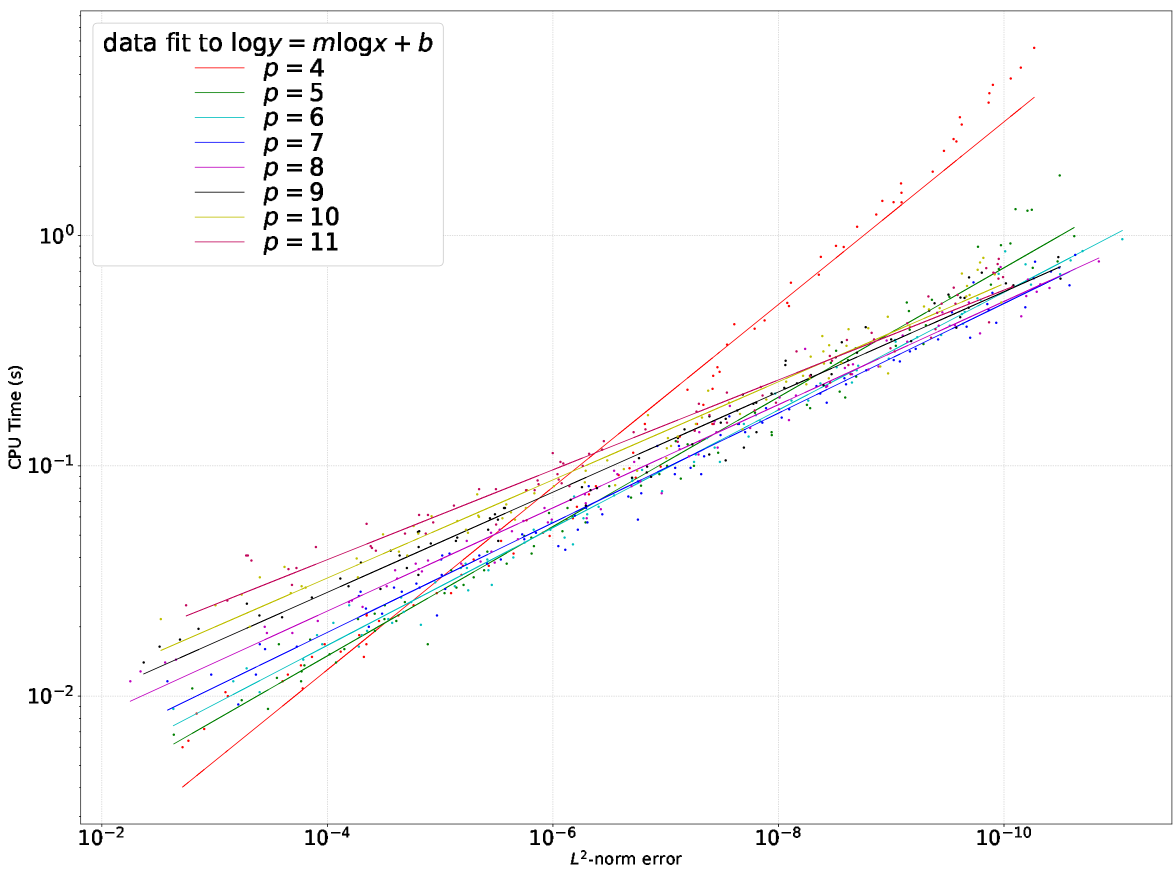

Figure 306: BACOLI/LE Work vs. Accuracy: Two Layer Burgers equation $\epsilon = 10^{-3}$; $p = 4 \ldots 11$

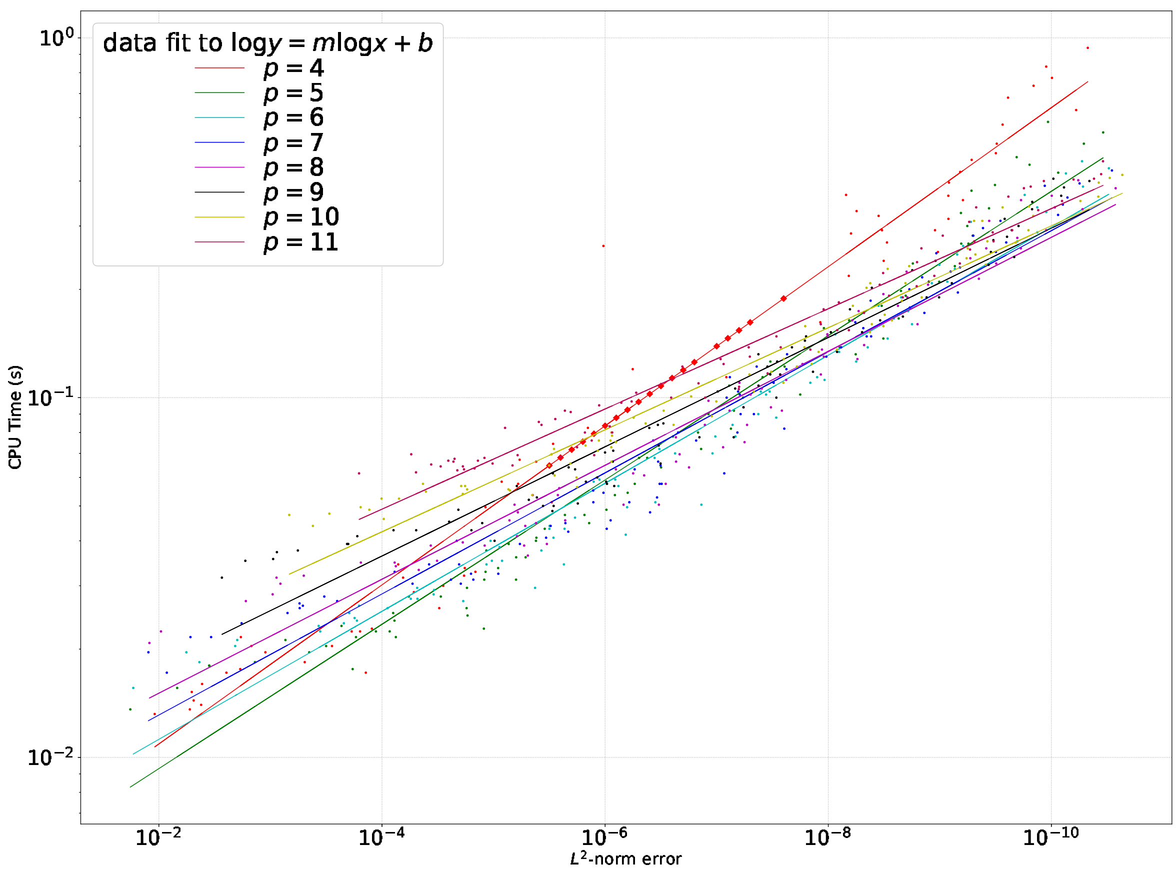

Figure 307: BACOLRI/ST Work vs. Accuracy: Two Layer Burgers equation $\epsilon = 10^{-3}$; $p = 4 \ldots 11$

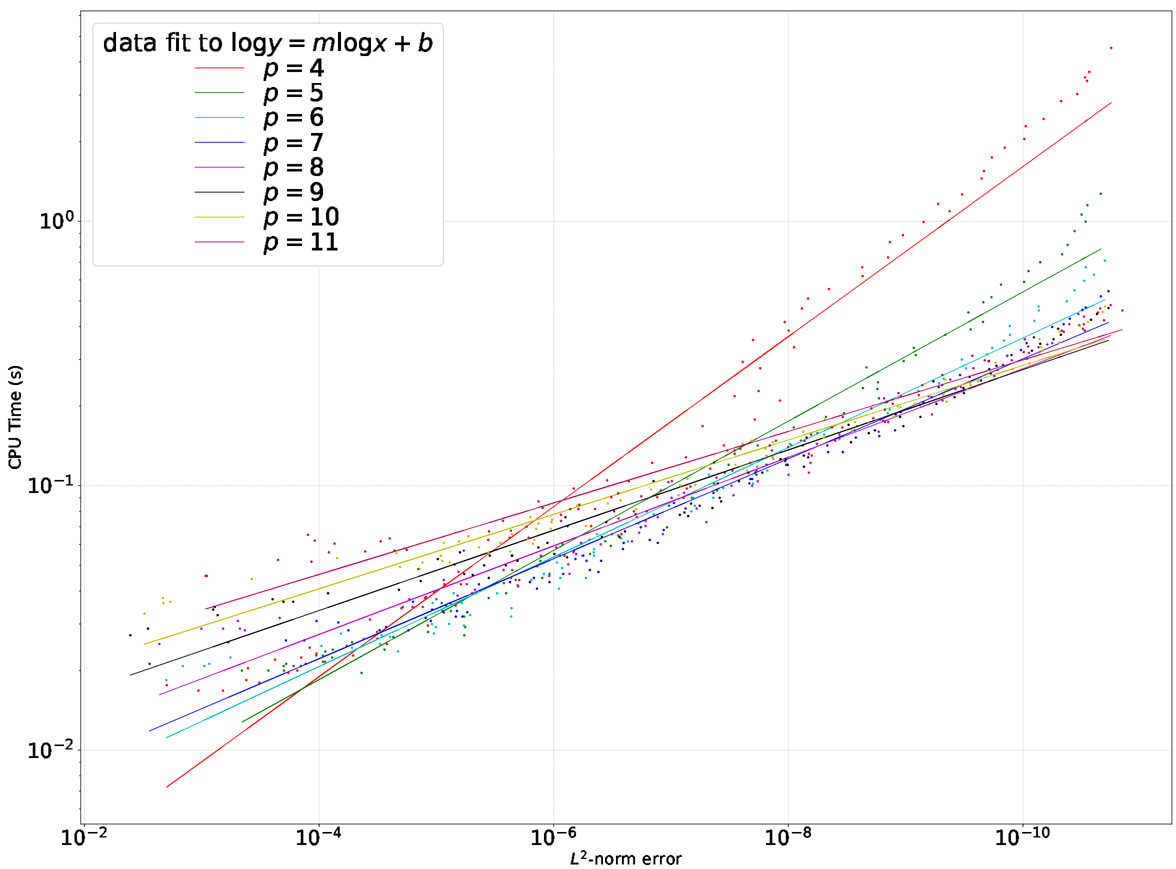

Figure 308: BACOLRI/LE Work vs. Accuracy: Two Layer Burgers equation $\epsilon = 10^{-3}$; $p = 4 \ldots 11$

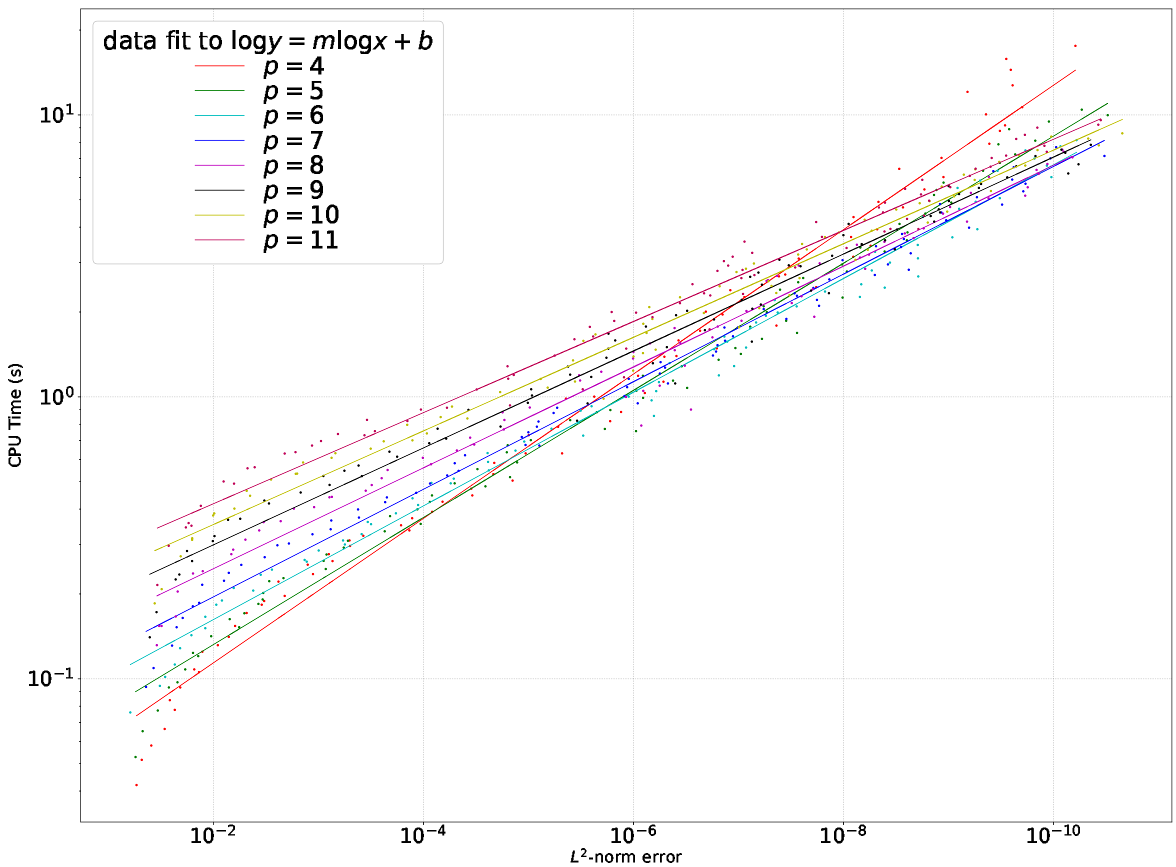

Figure 309: BACOLI/ST Work vs. Accuracy: Two Layer Burgers equation $\epsilon = 10^{-4}$; $p = 4 \ldots 11$

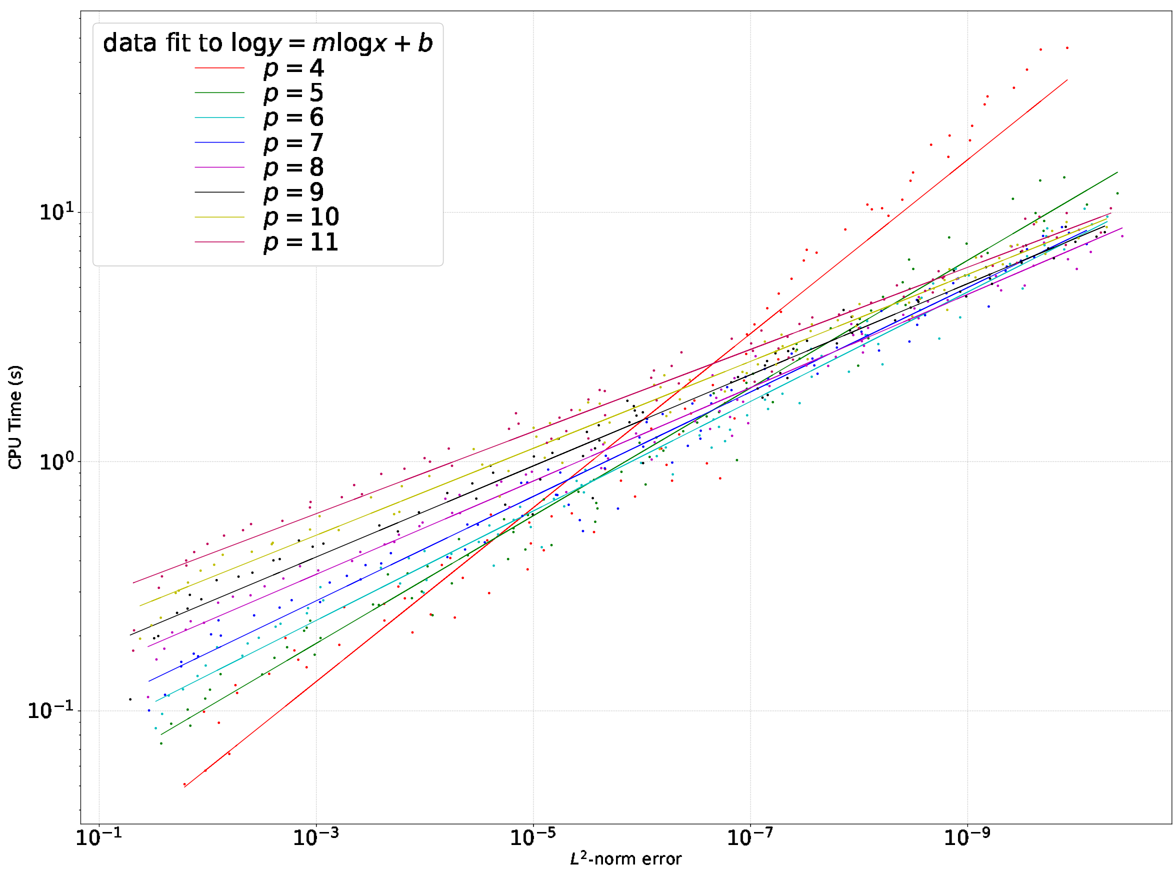

Figure 310: BACOLI/LE Work vs. Accuracy: Two Layer Burgers equation $\epsilon = 10^{-4}$; $p = 4 \ldots 11$

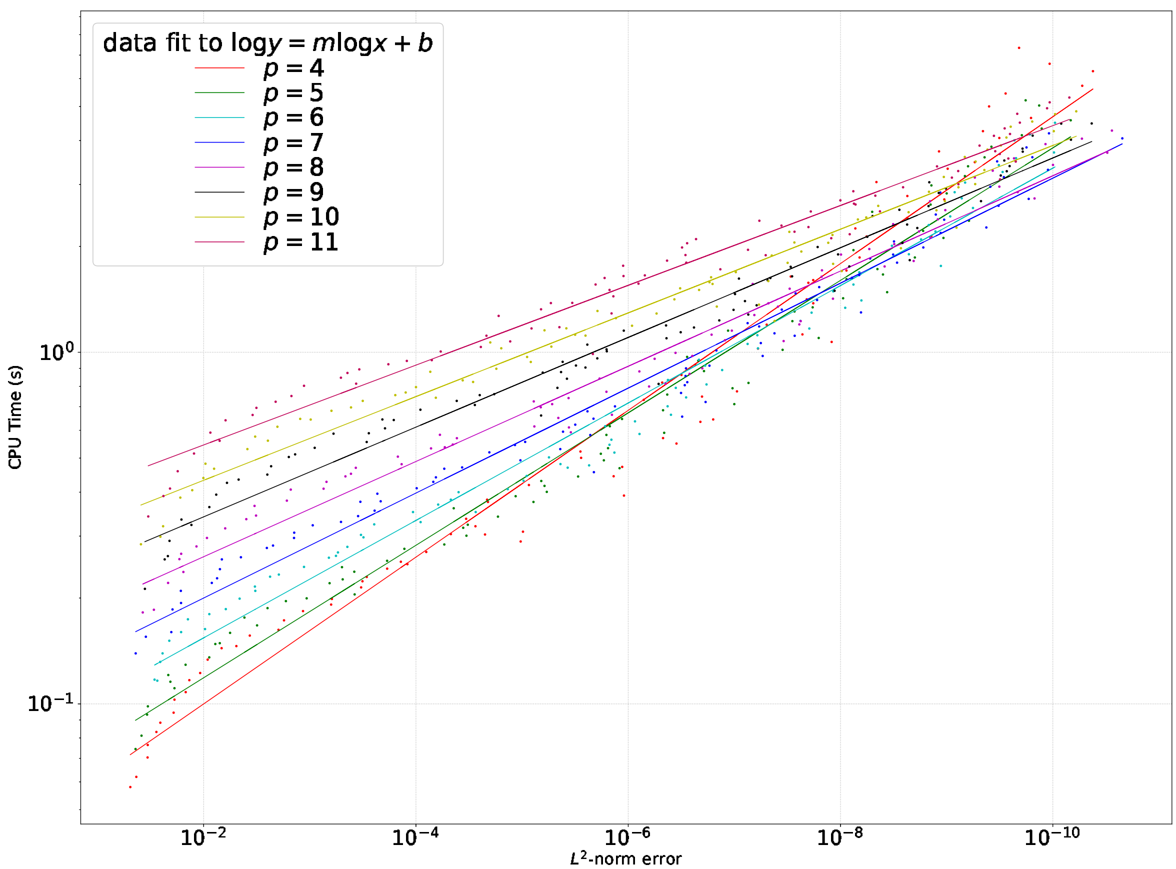

Figure 311: BACOLRI/ST Work vs. Accuracy: Two Layer Burgers equation $\epsilon = 10^{-4}$; $p = 4 \ldots 11$

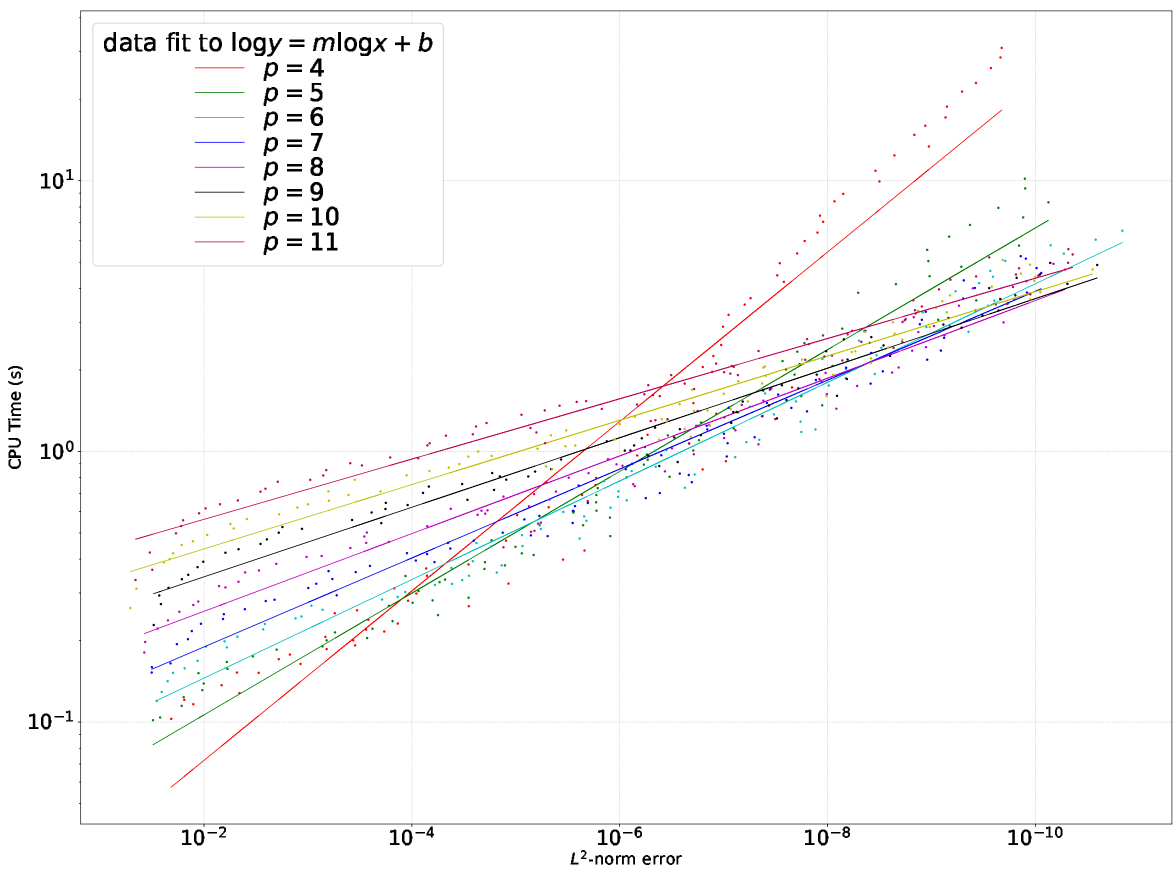

Figure 312: BACOLRI/LE Work vs. Accuracy: Two Layer Burgers equation $\epsilon = 10^{-4}$; $p = 4 \ldots 11$

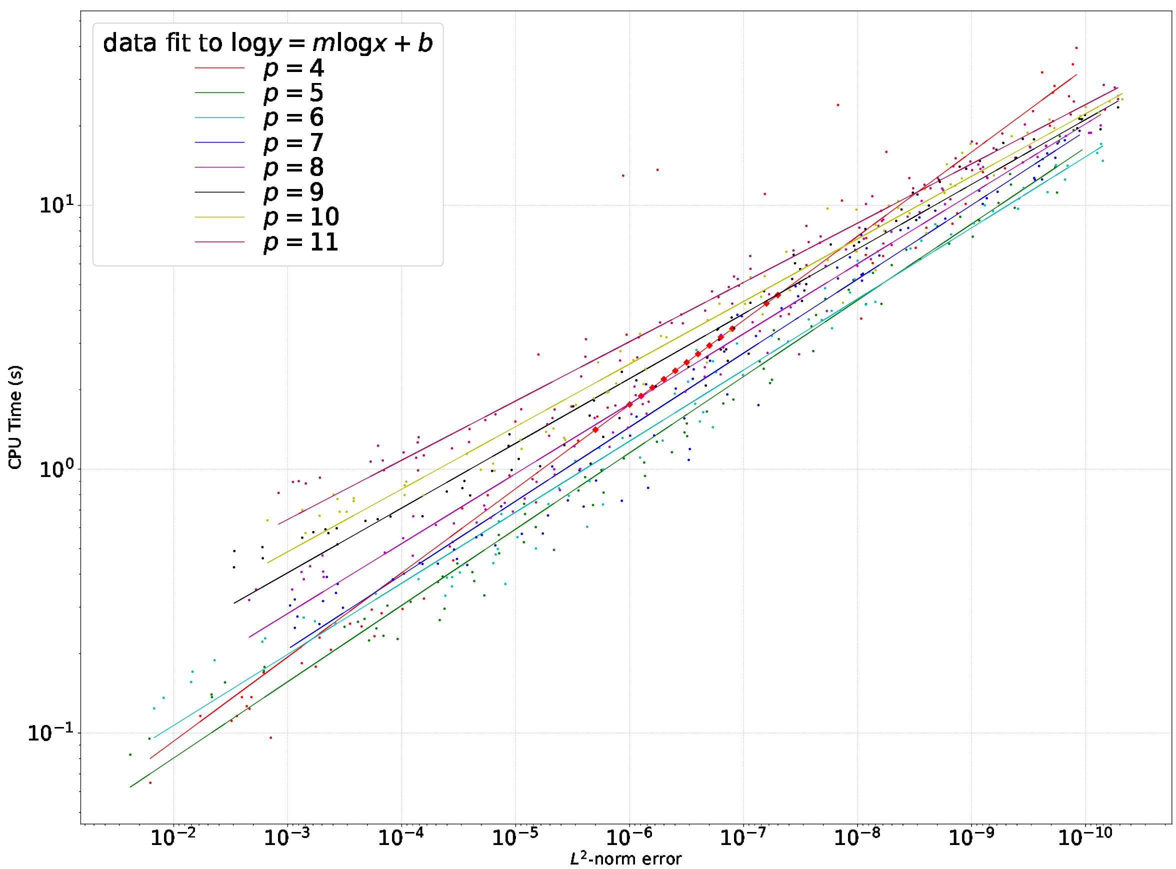

Figure 313: BACOLI/ST Work vs. Accuracy: Two Layer Burgers equation $\times 12$ $\epsilon = 10^{-3}$; $p = 4 \ldots 11$

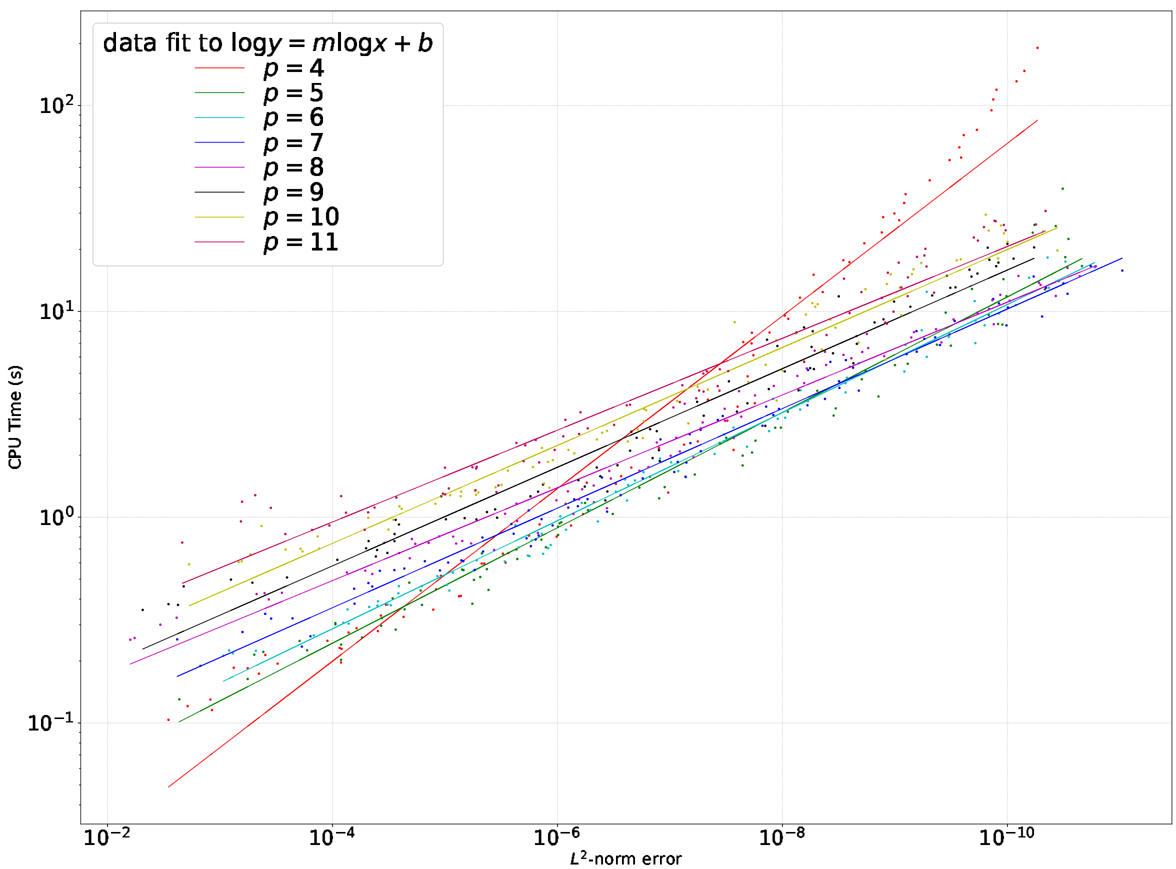

Figure 314: BACOLI/LE Work vs. Accuracy: Two Layer Burgers equation $\times 12$ $\epsilon = 10^{-3}$; $p = 4 \ldots 11$

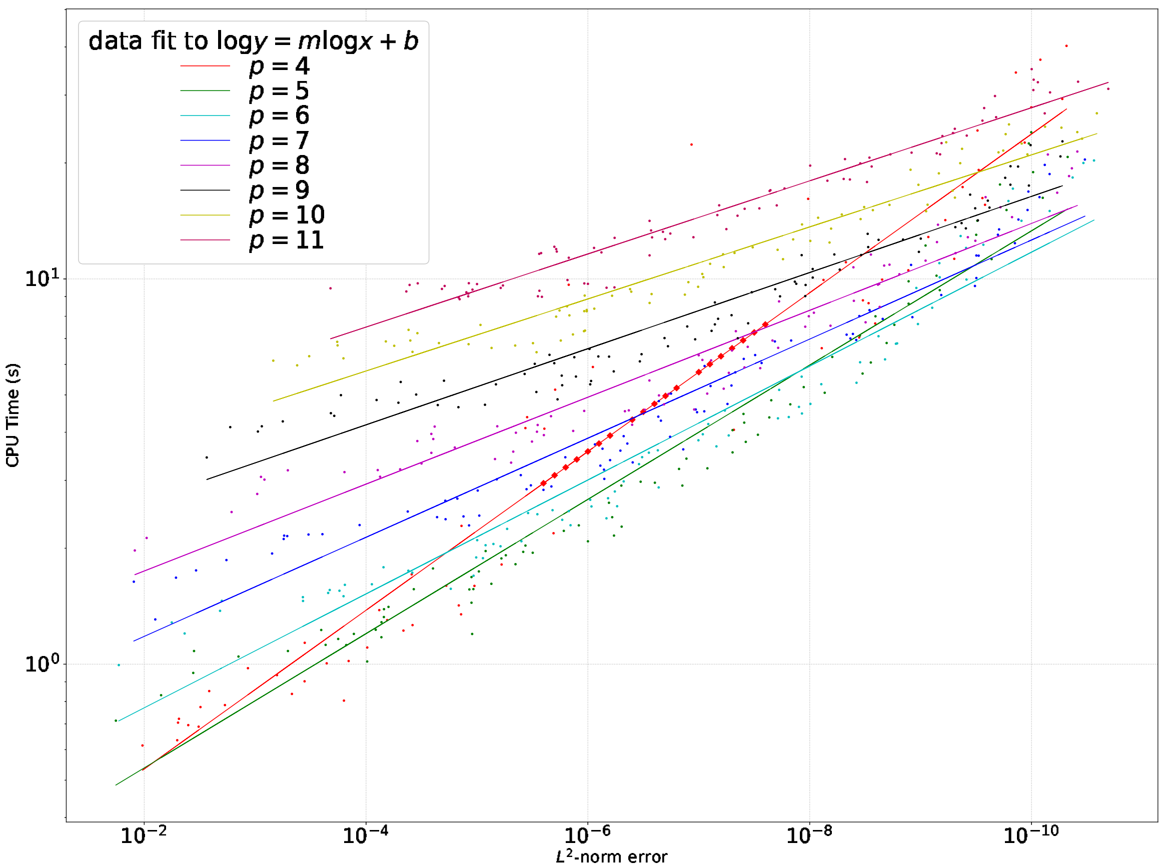

Figure 315: BACOLRI/ST Work vs. Accuracy: Two Layer Burgers equation $\times 12$ $\epsilon = 10^{-3}$; $p = 4 \ldots 11$

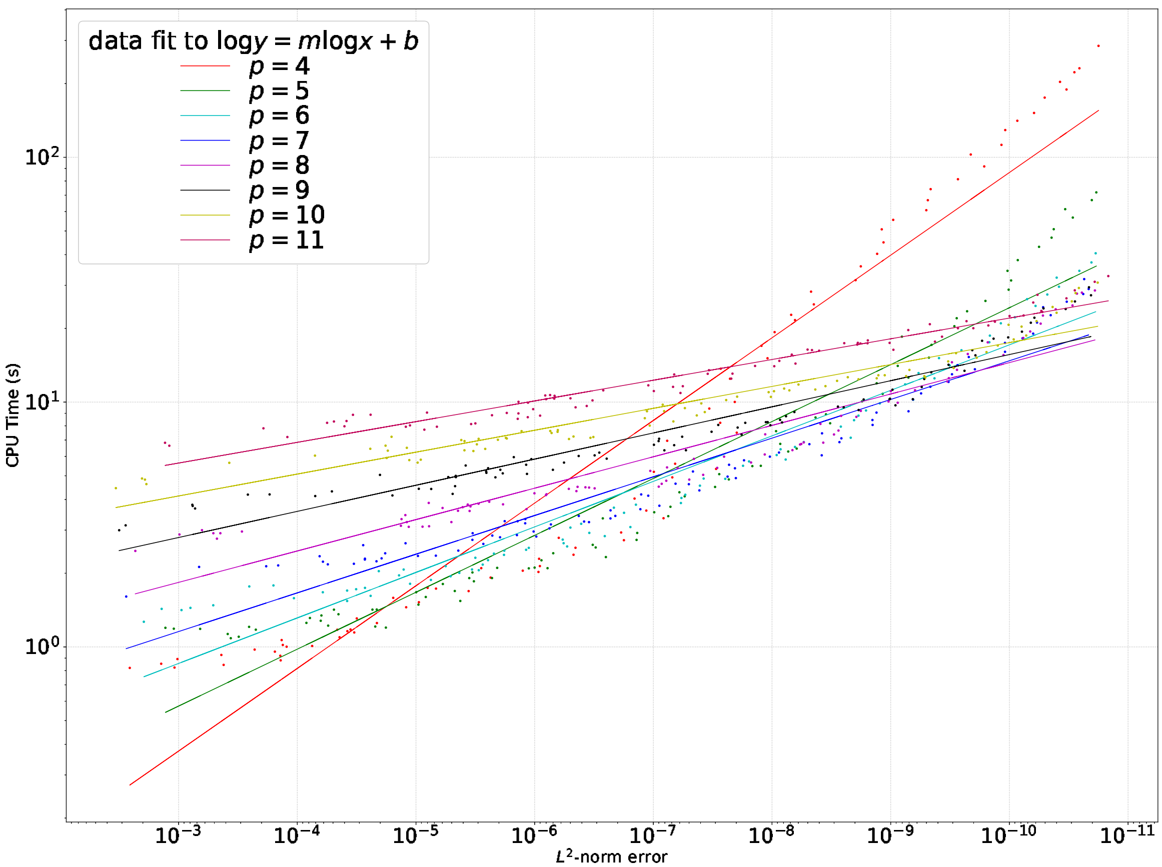

Figure 316: BACOLRI/LE Work vs. Accuracy: Two Layer Burgers equation $\times 12$ $\epsilon = 10^{-3}$; $p = 4 \ldots 11$

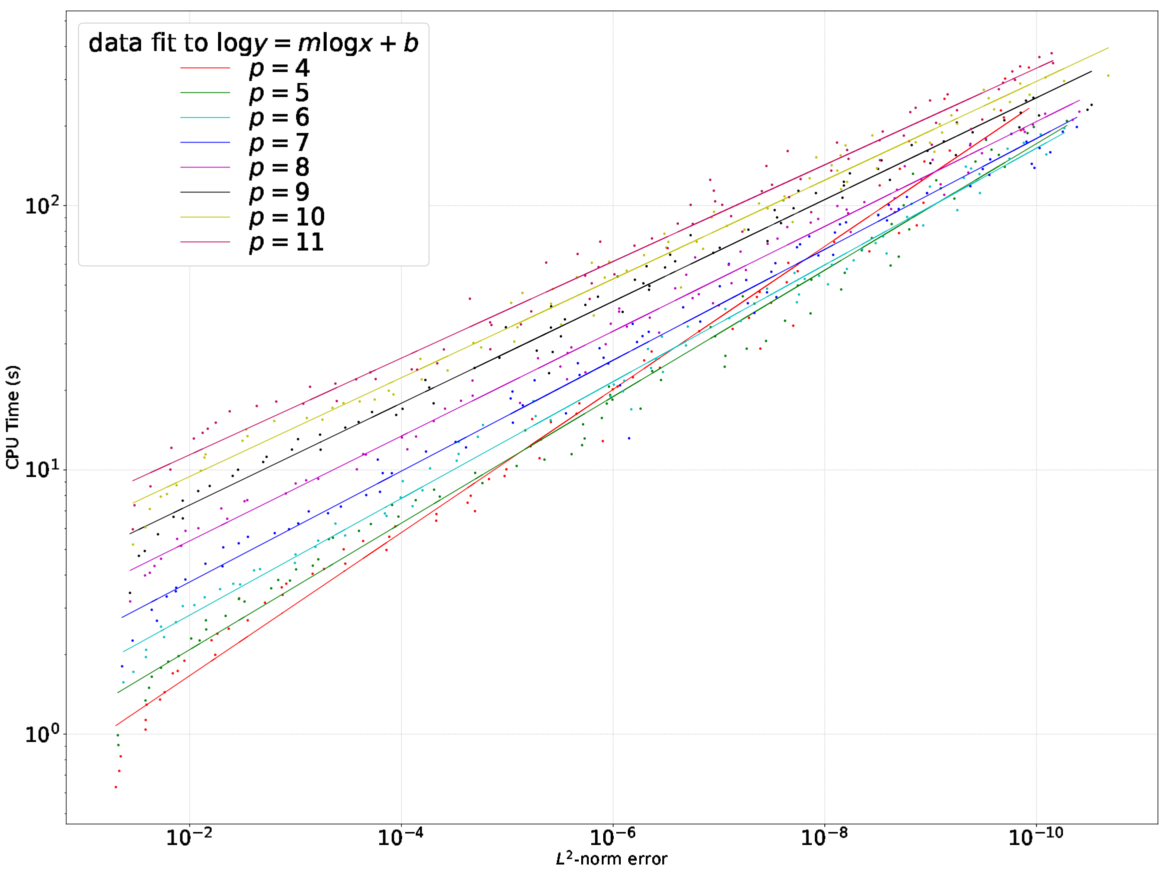

Figure 317: BACOLI/ST Work vs. Accuracy: Two Layer Burgers equation $\times 12$
$\epsilon = 10^{-4}$; $p = 4 \ldots 11$

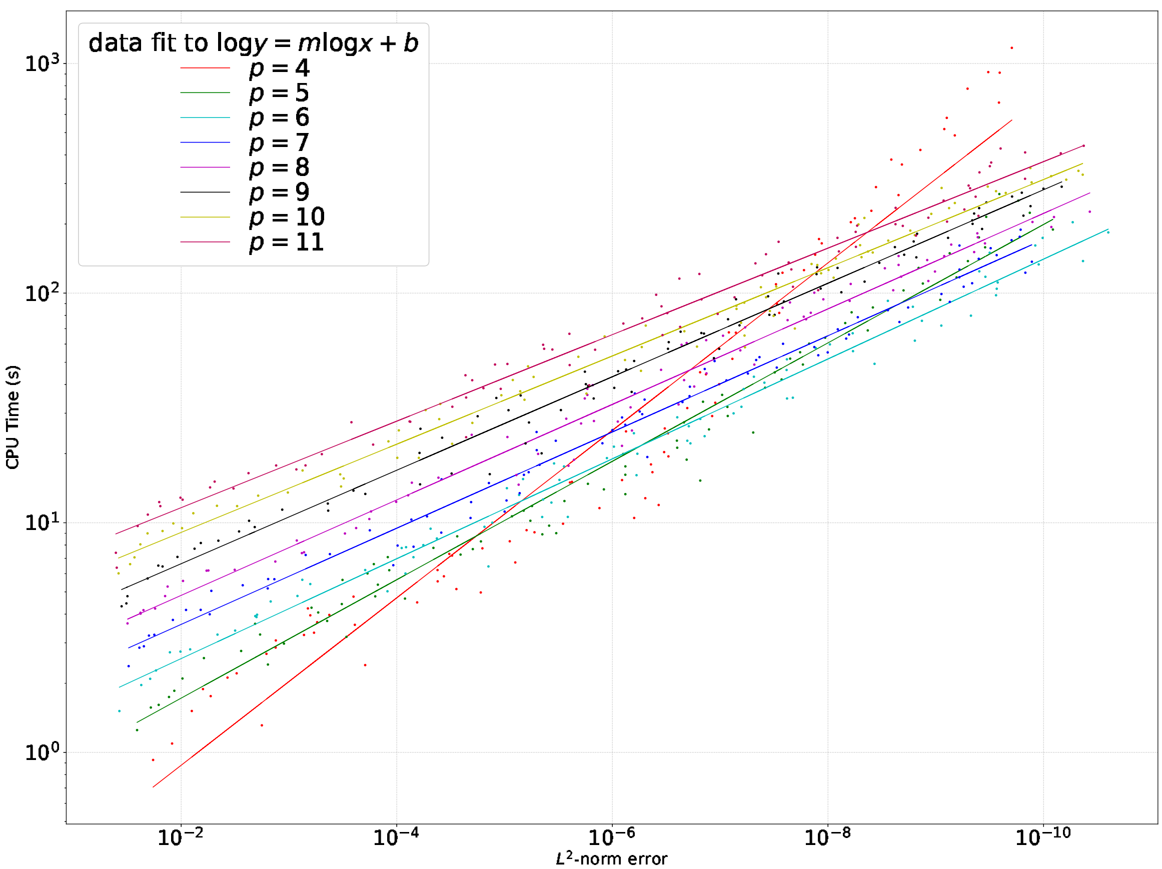

Figure 318: BACOLI/LE Work vs. Accuracy: Two Layer Burgers equation $\times 12$ $\epsilon = 10^{-4}$; $p = 4 \dots 11$

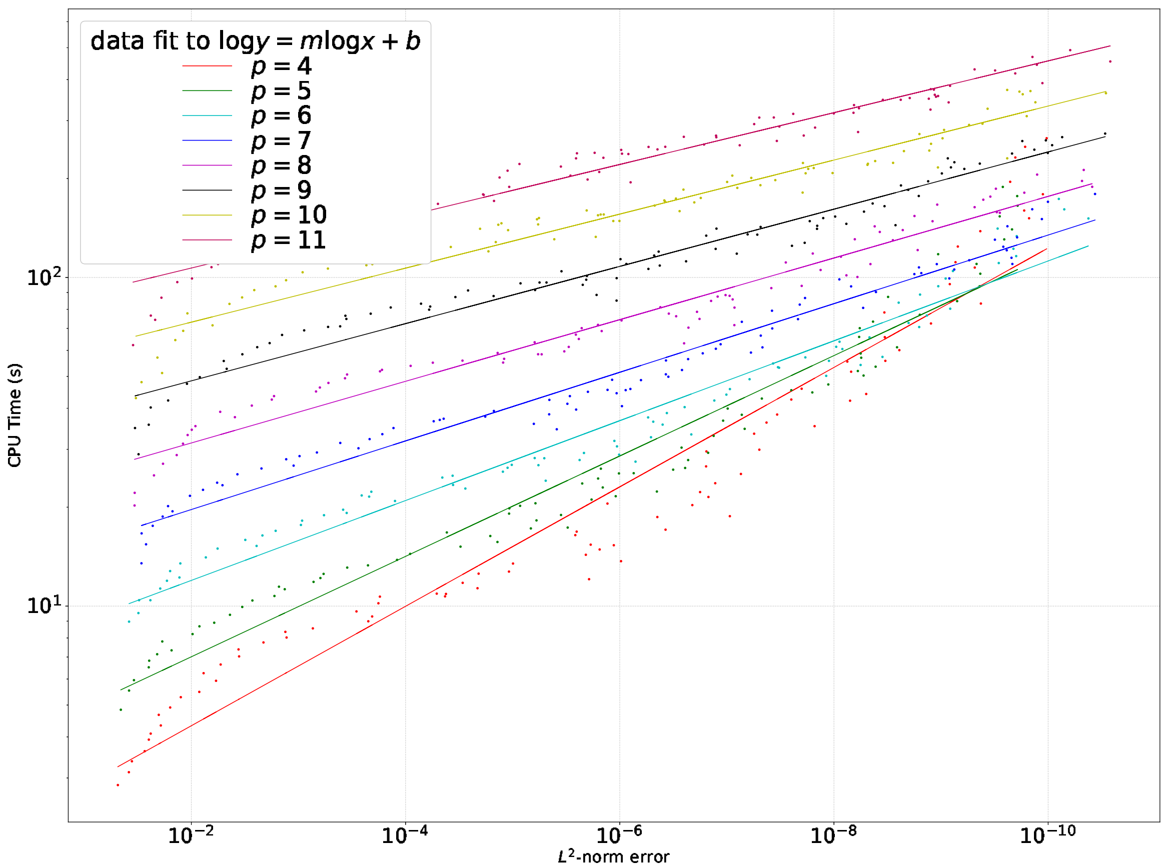

Figure 319: BACOLRI/ST Work vs. Accuracy: Two Layer Burgers equation $\times 12$ $\epsilon = 10^{-4}$; $p = 4 \ldots 11$

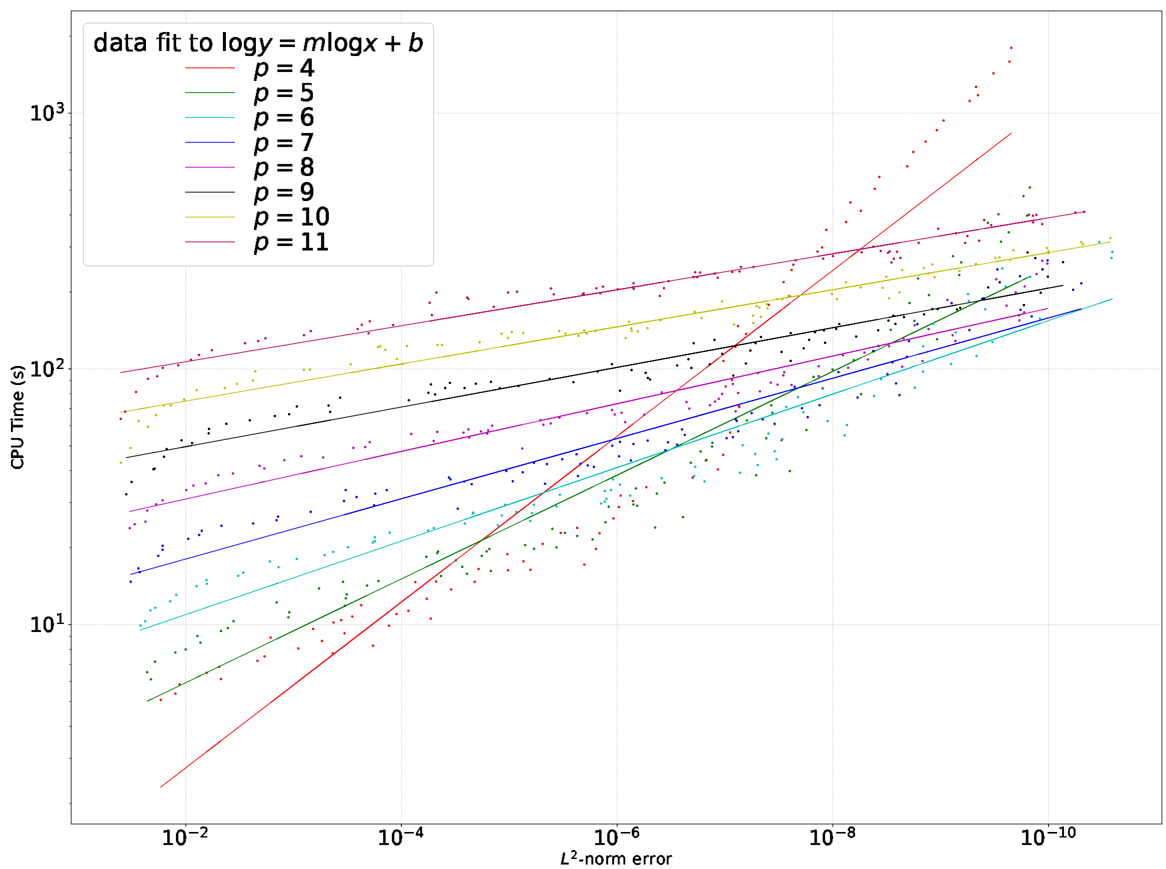

Figure 320: BACOLRI/LE Work vs. Accuracy: Two Layer Burgers equation $\times 12$ $\epsilon = 10^{-4}$; $p = 4 \ldots 11$

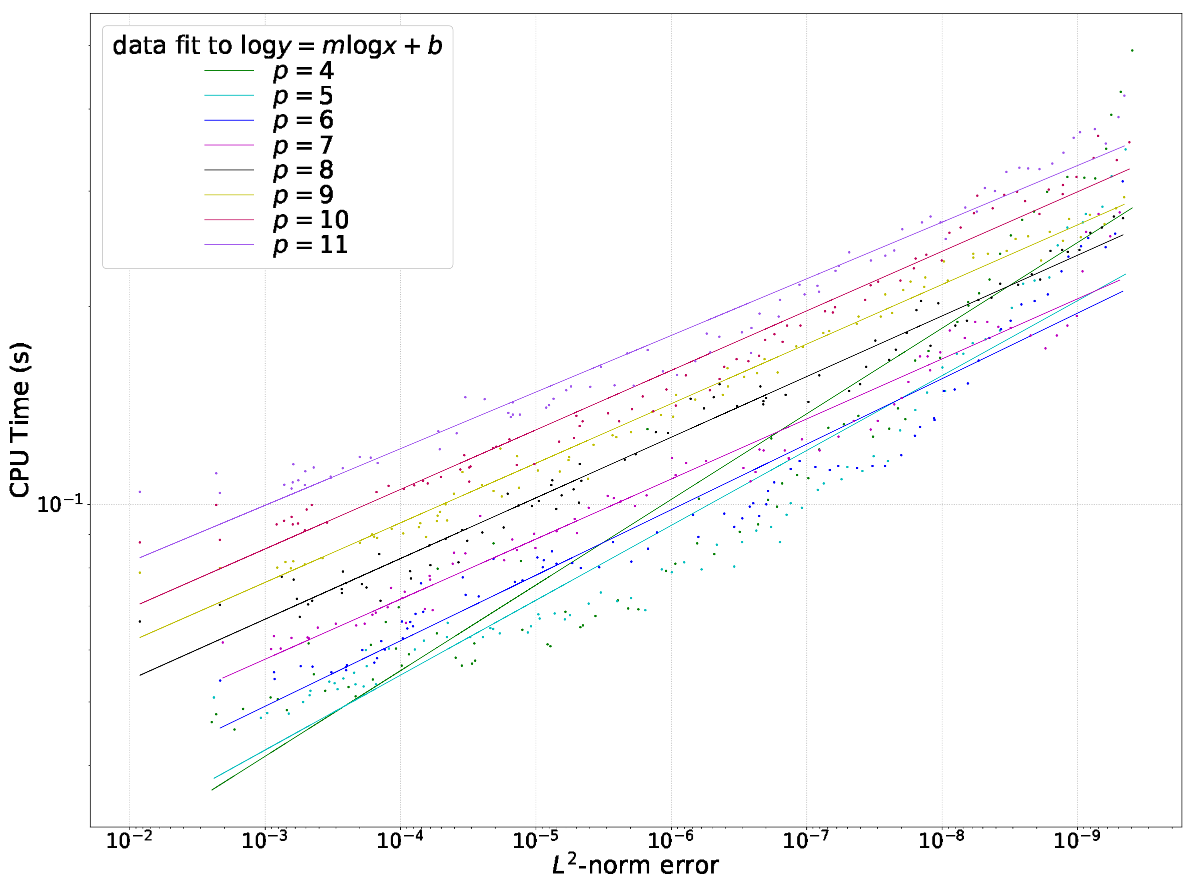

Figure 321: BACOLRI/ST Work vs. Accuracy: Schrödinger System; $p = 4 \ldots 11$

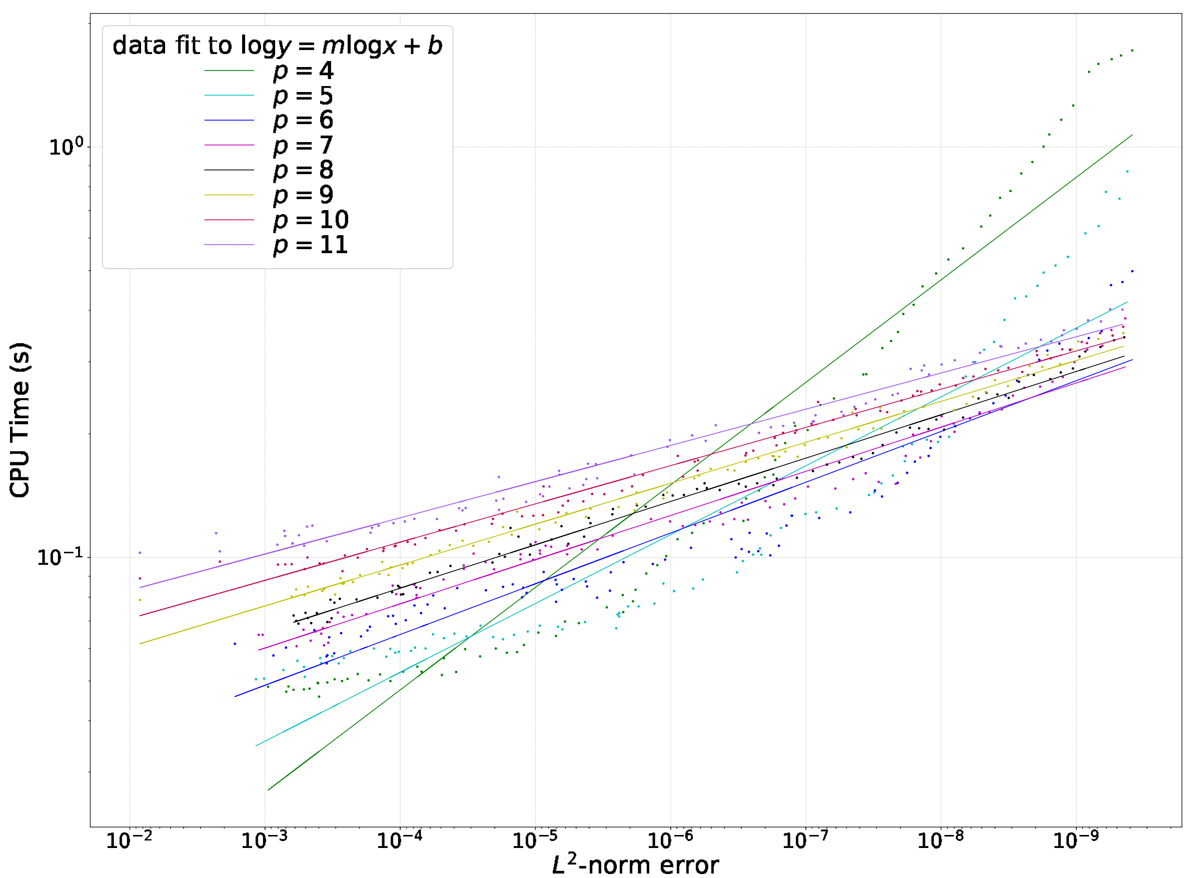

Figure 322: BACOLRI/LE Work vs. Accuracy: Schrödinger System; $p = 4 \ldots 11$