# Performance Analysis Results for Error Control B-spline Gaussian Collocation PDE Solvers *

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

**Abstract**

B-spline Gaussian collocation software has been widely used for the numerical solution of PDEs in one space dimension (1D) for several decades. Such packages represent the approximate solution as a linear combination of B-spline basis functions, of a given degree $p$, with time-dependent coefficients which are determined by requiring the approximate solution to satisfy the boundary conditions and the differential equations at certain points within the interior of the spatial domain. An essential capability of a high quality numerical software package is that it provide error control. That is, the software must return a numerical solution such that an associated error estimate satisfies the given user tolerance. The 1D PDE solver, BACOL, developed a little over a decade ago, was the first B-spline Gaussian collocation package to provide both temporal and spatial error control. The recently developed package, BACOLI, improves upon the efficiency of the spatial error estimation of BACOL through the use of two new interpolation-based schemes and provides two corresponding types of spatial error control. In this report we investigate the performances of the BACOL and BACOLI packages with respect to a number of important algorithmic assessments and examine the effectiveness of the new error estimation schemes, the new error control strategies, and the influence of the choice of $p$ on the efficiency of the solvers. These results will provide insights for making improvements to BACOLI, for making improvements to Gaussian collocation software for boundary value ODEs, and for the further development of error control B-spline Gaussian collocation software for the numerical solution of 2D PDEs.

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

*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, muir@smu.ca

# 1 Introduction

Gaussian spline collocation algorithms, or orthogonal spline collocation (OSC) algorithms, as they are also commonly known, have been used in the numerical solution of PDEs for many decades. Examples of recent papers that apply OSC in the spatial discretization of PDEs include [5, 6] which apply OSC based on Hermite cubic splines for the spatial discretization and the Crank-Nicholson (C-N) and Extrapolated Crank-Nicholson (EC-N) schemes (respectively) to PDEs in one spatial dimension (1D) that have non-local integral conditions that depend on the solution across the entire spatial domain rather than local boundary conditions. In [10], OSC based on Hermite cubic splines is employed for the spatial discretization of the 1D Sine-Gordon PDE, which involves the second time derivative of the solution; a finite difference scheme is used for the time integration. OSC with Hermite cubic splines is applied in the spatial discretization of PDEs with two spatial dimensions (2D) within the framework of an alternating direction implicit (ADI) approach together with an EC-N scheme for the time-stepping in [13, 12]; in the former a fixed 2D spatial domain is assumed while in latter a moving 2D spatial domain is considered. The papers [31, 30, 18, 11] consider the application of a family of OSC schemes to 2D fractional diffusion PDEs; a major challenge in these problems is treating the Riemann-Liouville or Caputo fractional derivatives that arise in the PDEs; for the time discretization, [31, 30] use finite differences while [18] uses the L1 and Grünwald-Letnikov approximations, and in [11] an ADI C-N scheme is used.

It is also common to represent the spline space employed in a Gaussian collocation algorithm in terms of B-splines [8]. The standard B-spline Gaussian collocation algorithm represents the approximate solution as a linear combination of known B-spline basis functions, of a given degree $p$, with unknown coefficients that are determined by requiring the approximate solution to satisfy (i) the differential equation at a set of points (images of the Gauss points) on each subinterval of a mesh that partitions the spatial domain and (ii) the boundary conditions at the endpoints of the spatial domain.

B-spline Gaussian collocation software packages, designed to solve general classes of problems, have been widely used for the numerical solution of boundary value ordinary differential equations (BVODEs) and 1D PDEs for several decades. Such software packages often employ a framework that provides an error controlled computation of the numerical solution. Error control involves the computation of an error estimate for the computed solution which is used as the basis for an adaptive computation (typically involving the use of an adaptive mesh refinement algorithm) that yields a numerical solution which has an error estimate that satisfies the given user tolerance. Error control provides two important advantages: (i) the user can have reasonable confidence that the numerical solution has an error that is consistent with the requested tolerance, and (ii) the cost of the computation will be consistent with the requested accuracy, i.e., error control software will typically adapt the computation so that the amount of work performed will be approximately proportional to the desired accuracy, using, for example, a spatial mesh with an appropriate number

and distribution of points required to obtain a numerical solution whose error estimate approximately agrees with the requested tolerance.

For systems of BVODEs, the very successful COLSYS package [3], developed about four decades ago, represents the first error control software package based on B-spline Gaussian collocation. The COLSYS package has been very widely used for many decades and has also been implemented in a number of numerical software libraries, e.g., [20], as well as in several problem solving environments such as Scilab (bvode) [23] and Python (scikits.bvp1lg) [25]. (In some cases, the implementation is based on COLNEW [4], a modified version of COLSYS, in which the primary difference is the replacement of the B-spine basis with a monomial basis; the Gaussian collocation and error control algorithms are largely unchanged).

For 1D parabolic PDEs, the B-spline Gaussian collocation packages PDECOL [21], also developed about four decades ago, and EPDCOL [17], developed about three decades ago, are the earliest packages of this type. Both of these packages provide control of an estimate of the temporal error but not the spatial error. The more recently developed packages, BACOL [27, 29] and BACOLR [26], implement B-spline Gaussian collocation for the spatial discretization and provide both temporal and spatial error control. The error controlled computation of the time-dependent B-spline coefficients is performed in BACOL using the DASSL [7] package (which is based on a family of multi-step methods known as Backward Differentiation Formulas (BDFs) [7]), and in BACOLR using the RADAU5 package [15] (which is based on a 5th order implicit Runge-Kutta method of Radau IIA type [15]). Both BACOL and BACOLR obtain a spatial error estimate by computing a second (higher degree) collocation solution, essentially doubling the cost of the computation. 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 tolerance requirements [28]. In [26], numerical comparisons of BACOL and BACOLR show that the two codes perform similarly on several standard test problems and that BACOLR has much superior performance on problems for which the stability of the higher order BDFs is an issue. The stability regions of the higher order BDFs do not include the imaginary axis and thus problems that lead to DAEs having Jacobians with eigenvalues near the imaginary axis, for example, Schrödinger type problems, cannot be treated using the higher order BDFs. The paper [26] shows that BACOL fails on problems of this type unless DASSL is restricted to using only lower order BDFs, in which case the efficiency of the computation is substantially degraded.

A recently released package from this family is BACOLI [22]. This package improves on the efficiency of the spatial error estimation scheme employed by BACOL by introducing two new interpolation-based schemes, each of which employs a different spatial error control mode. We discuss the BACOL and BACOLI packages in more detail later in this report.

The development of B-spline Gaussian collocation software for 2D parabolic PDEs has recently been considered in [19]. (See also earlier related work [24].) The approach assumes a rectangular spatial domain partitioned by a rectangu-

lar grid and the approximate solution is represented as a linear combination of tensor products of known 1D B-spline basis functions for each of the two spatial dimensions, with unknown time dependent coefficients. These coefficients are determined through the solution of a system of Differential-Algebraic Equations (DAEs) obtained through the application of tensor product Gaussian collocation to the PDEs and boundary conditions. Since this software does not provide adaptive error control, current work on this project is focused on (i) the development of efficient spatial error estimation techniques involving the extension of techniques developed for the 1D case, and (ii) the development of spatial mesh adaptation (in order to provide control of the spatial error estimate) based on ideas from moving mesh methods, see, e.g., [16].

The development of error control B-spline Gaussian collocation software for 2D parabolic PDEs depends heavily on the approaches employed in the 1D case. *A primary goal of this report, therefore, is an investigation of the performance of the above mentioned 1D PDE solvers in order to apply the results to the further development of B-spline Gaussian collocation software for the 2D case.*

A common feature of the COLSYS and COLNEW packages and the BACOL family of 1D PDE packages mentioned above is that they all implement a *family* of Gaussian collocation methods over a range of values of $p$ (the degree of the B-spline basis). This feature is available because the B-spline Gaussian collocation algorithm is quite general in this regard and there are no significant additional implementation issues associated with the use of higher degree methods. While the user is required to select a value for an input argument that determines $p$, the packages provide little or no guidance to the user regarding how to choose this parameter, and, to our knowledge, there has been little investigation of the impact of the degree of the B-spline basis on the overall code performance, particularly for the PDE case. *Another primary goal of this report therefore is to investigate the role that the degree of the B-spline basis plays in efficiency of the computation.*

We do not include BACOLR in this investigation. BACOL and BACOLR were compared in [26]. Future work involves the development and analysis of a new version of BACOLR (called BACOLRI) that implements the interpolation-based spatial error estimation schemes that are implemented in BACOLI; that project will involve a comparison of BACOLR and BACOLRI, as well as a comparison of BACOLI and BACOLRI.

Performance analysis of scientific software for differential equations has a long history. While efficient, stable and robust numerical methods are the foundation upon which all high quality numerical software packages are developed, the sophistication of such packages and the complexity of the interaction among the component algorithms of the packages implies that extensive numerical testing together with subsequent analysis of the results of the numerical experiments is an essential factor in the development of such software. For example, the software package BACOLI includes implementations of a number of complex numerical algorithms such as adaptive error-controlled time-stepping and method order selection for the DAEs and adaptive spatial mesh refinement for the discretization of the spatial domain of the PDEs. In addition, the code also has

available two spatial error control schemes and a family of collocation methods. *The ways in which these algorithms interact to affect the efficiency and reliability of the software must be assessed through detailed numerical experimentation and performance analysis.*

This report is organized as follows. In Section 2, we provide an overview of the B-spline Gaussian collocation algorithm for 1D PDEs, followed by a brief review of the BACOL and BACOLI packages. Section 3 provides results from a set of numerical experiments in which the BACOL and BACOLI packages are investigated. The packages are compared with respect to several important machine independent measures that provide insight into the performance of the algorithms implemented within these packages. As well, work-precision results are presented that compare the work required vs. accuracy achieved by the packages. The impact of the spatial error estimation schemes and the spatial error control modes is also investigated. Section 3 also provides results which examine the role that the degree of the B-spline basis plays in the efficiency of the solvers. We close in Section 4 with a discussion of the results and suggestions for future work.

# 2 B-spline Adaptive Gaussian Collocation Error Control Software for PDEs

In this section, we first describe the B-spline Gaussian collocation algorithm for a system of PDEs. We then give a brief description of the BACOL software package which implements this algorithm within a framework that provides temporal and spatial error control. We follow with a brief description of the spatial error estimation scheme employed by BACOL and the two interpolation-based spatial error estimation schemes implemented in the new package, BACOLI, which improve upon the efficiency of the spatial error estimation scheme employed by BACOL. We also describe the two spatial error control modes implemented in BACOLI.

## 2.1 B-spline Gaussian Collocation with Temporal and Spatial Error Control

We assume a PDE system 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

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

Then, for a spatial mesh, $\{x_i\}_{i=0}^{NINT}$, ($NINT$ is the number of subintervals into which the spatial mesh partitions the spatial domain), and assuming degree $p$ for the piecewise polynomials to be used in the representation of the spatial dependence of the approximate solution, we express this approximate solution, $\underline{U}(x,t)$, in 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 determination of the coefficients, $\underline{y}_{p,i}(t)$, involves the imposition of collocation conditions, obtained by requiring $\underline{U}(x,t)$ to satisfy the PDE at $p-1$ points within each spatial mesh subinterval. The collocation conditions are of the form

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

for $l = 2,\ldots,NC_p-1$, where

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

$\{\rho_j\}_{j=1}^{p-1}$, are the images of the order $p-1$ Gauss points on $[0,1]$, and $h_i = x_{i-1} - x_i$. We define $\xi_1 = a$ and $\xi_{NC_p} = b$, corresponding to the left and right endpoints of the spatial domain, and require $\underline{U}(x,t)$ to also satisfy the boundary conditions; this gives the conditions

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

BACOL solves the coupled time-dependent system of collocation conditions (5) and boundary conditions (7), a DAE system, using, as mentioned earlier, the DAE solver DASSL, which provides a temporal error controlled computation of the B-spline coefficients. The computation performed by DASSL requires the solution of linear systems, that, due to the use of B-spline Gaussian collocation, have an *almost block diagonal (ABD)* structure [9], and BACOL makes use of the software package, COLROW [9], which is designed to efficiently handle such systems. Within COLROW, the CRDCMP routine performs factorizations of the ABD coefficient matrices, while the CRSLVE routine solves the factored ABD systems.

The BACOL package also implements the computation and control (through adaptive spatial mesh refinement) of an estimate of the spatial error of the collocation solution. After each accepted time step taken by DASSL, BACOL computes an estimate of the spatial error and checks to see if it satisfies the user tolerance. If the tolerance is not satisfied, the solution obtained at the current time is rejected and a remeshing (i.e., a redistribution and possible refinement of the spatial mesh) is performed. The 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 [29] for further details. Based on the new spatial mesh, the computation of the solution on the current time step is repeated using DASSL in what is referred to as a "warm start" mode; this means that the current time step and method (i.e., BDF of a given order) are used. If, after several failed attempts to obtain a spatial mesh such that the spatial error estimate for the computed solution satisfies the user tolerance, BACOL will restart the time step using DASSL in what is referred to as "cold start" mode; this means that DASSL restarts with a very small time step and with the BDF of order one. Cold starts are often computationally expensive since it can take a substantial number of time steps and method order increases before DASSL is able to return to the larger step size and higher order method that it was using before the cold start was enforced.

## 2.2 Spatial Error Estimation Schemes

The BACOL spatial error estimate is obtained by computing a *second* collocation solution, $\underline{\bar{U}}(x,t)$, on the same spatial mesh and for the same time $t$, but based on B-splines of degree $p+1$. A scaled difference of $\underline{U}(x,t)$ and $\underline{\bar{U}}(x,t)$ is then computed to provide a spatial error estimate for $\underline{U}(x,t)$. To obtain $\underline{\bar{U}}(x,t)$, BACOL must perform a computation similar to that described above for the computation of $\underline{U}(x,t)$; *this essentially doubles the overall cost and represents an obvious inefficiency in the computation.*

This inefficiency has recently been addressed through the development of the BACOLI package which replaces the expensive computation of the two collocation solutions with a more efficient computation in which a piecewise polynomial interpolant is constructed, based on evaluations of the lone collocation solution, $\underline{U}(x,t)$, that is computed. Two types of piecewise polynomial interpolants have been developed. One, called the *SuperConvergent Interpolant (SCI)* [1], is based on evaluations of $\underline{U}(x,t)$ at certain points on each subinterval where it is superconvergent in space, i.e., the rate of convergence of the spatial error at these points is at least one order higher than at an arbitrary point in the spatial domain, and thus $\underline{U}(x,t)$ is expected to be more accurate at these points. A sufficient number of interpolation points are chosen so that the interpolation error is dominated by the error of the superconvergent values and thus the resultant interpolant is of one order of spatial accuracy higher than $\underline{U}(x,t)$. A scaled difference of the SCI and $\underline{U}(x,t)$ is then computed to provide a spatial error estimate for $\underline{U}(x,t)$.

The second type of interpolant, called the *Lower Order Interpolant (LOI)* [2], is based on evaluations of $\underline{U}(x,t)$ at a set of points on each spatial subinterval such that the resultant interpolant has an interpolation error that is asymptotically equivalent to the collocation error for a collocation solution of one order lower than $\underline{U}(x,t)$. In this case the interpolation error of the LOI dominates the error associated with the $\underline{U}(x,t)$ values upon which the interpolant is based. And then a scaled difference of the LOI and $\underline{U}(x,t)$ provides a spatial error

estimate for a collocation solution that is of one lower order than $\underline{U}(x,t)$. That is, the spatial error estimate is not actually for $\underline{U}(x,t)$. We return to this point shortly.

In BACOL, $\underline{U}(x,t)$ is the primary solution returned to the user, and $\bar{\underline{U}}(x,t)$ is computed only to obtain a spatial error estimate for $\underline{U}(x,t)$. Since the spatial error estimate that is computed and then controlled is for the approximate solution, $\underline{U}(x,t)$, that is returned to the user, we refer to this as *Standard (ST) Error Control*, and we will refer to BACOL, when running in this error control mode, as **BACOL/ST**. Since the SCI scheme also computes and then controls a spatial error estimate for the returned collocation solution $\underline{U}(x,t)$, BACOLI, when using the SCI option, is also using ST error control, and we will refer to the BACOLI code when running in this error control mode as **BACOLI/ST**. With a simple modification, it would be possible to have BACOL return $\bar{\underline{U}}(x,t)$, and in this case BACOL would be estimating and controlling a spatial error estimate for $\underline{U}(x,t)$ but $\bar{\underline{U}}(x,t)$ is the approximate solution that would be returned. We will refer to this type of error control as *Local Extrapolation (LE) Error Control* [14], and BACOL, when running in this error control mode, will be referred to as **BACOL/LE**. This type of error control arises in the context of Runge-Kutta formula pairs for the numerical solution of initial value ODEs [14]. Since BACOLI, when using the LOI option, also returns an approximate solution controlled by a spatial error estimate for a collocation solution of one lower order, we can observe that this is also an example of LE error control, and we will refer to BACOLI when running in this error control mode as **BACOLI/LE**.

# 3 Numerical Results

In this section, we present results from numerical experiments which can be used to investigate the performance of the four codes identified in the previous section:

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

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

### 3.1 Test Problems

We consider several standard test problems chosen from the literature. These problems are of interest because they have solutions that feature moving spatial layer regions; this places demands on the spatial error estimation and spatial mesh adaptivity algorithms to adequately detect and track the moving layers as the solution evolves in time. (See [29], Page 251, Figure 2, which shows this layer tracking capability.)

- **OLBE:** The One Layer Burgers Equation [28]:

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

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 $\epsilon$ is a problem-dependent parameter. We will consider two instances of this problem, one with $\epsilon = 10^{-3}$ and one with $\epsilon = 10^{-4}$. When $t = 0$, the solution has a sharp layer region around $x = 0.25$. As $t$ goes from 0 to 1, the layer moves to the right and is located around $x = 0.75$ when $t = t_{end} = 1$. We solve this problem from $t_0 = 0$ to $t_{end} = 1$.

- **TLBE:** The Two Layer Burgers Equation [28]: 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 $\epsilon$ is a problem-dependent parameter. We will consider two instances of this problem involving $\epsilon = 10^{-3}$ and $\epsilon = 10^{-4}$ respectively. When $t = 0$, the solution has two sharp layers around $x = 0.25$ and $x = 0.5$. As $t$ increases, these layers move to the right and eventually merge, forming a single layer around $x = 0.7$ when $t \approx 0.5$. As time increases further, the single layer continues to move to the right, and is located around $x = 0.9$ when $t = t_{end} = 1$. We solve this problem from $t_0 = 0$ to $t_{end} = 1$.

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

- **CSRM:** Catalytic Surface Reaction Model [32]:

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

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 $A_1 = A_2 = 30$. The solution components exhibit rapid variations in time and space. We solve this problem from $t_0 = 0$ to $t_{end} = 1$.

## 3.2 Machine Independent Performance Measures

In this subsection, we compare BACOL/ST, BACOL/LE, BACOLI/ST, and BACOLI/LE with respect to several *machine independent* measures of the algorithms employed in the codes that can contribute significantly to overall performance. These machine independent measures provide an important complement to standard machine dependent timing results; additional insights regarding code performance can be obtained by considering such performance 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 cold starts (**Cold Starts**), and the total number of factorizations (**Calls to CRDCMP**) and solves (**Calls to CRSLVE**) of ABD systems.

We provide machine independent results for nine test problems: (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 $\epsilon = 10^{-4}$, and (ix) **CSRM**. We consider $p = 4, 5, 7, 9$, and $tol = 10^{-4}, 10^{-6}, 10^{-8}$. *For all tests, we choose p so that we are comparing computations that return collocation solutions based on the same number of collocation points per subinterval.* Tables 1-9 give machine

independent performance measures for the four codes applied to the nine test problems identified above. Each table entry consists of two rows; the first row gives **Final NINT, Accepted Time Steps**, and **Remeshings**; the second row gives (**Cold Starts**), and [**Calls to CRDCMP, Calls to CRSLVE**]. (We note that BACOLI/ST fails on **TLBEx6** with $\epsilon = 10^{-3}$ and **TLBEx12** with $\epsilon = 10^{-3}$ when $p = 4$ and $tol = 10^{-6}$; in these two cases the corresponding table entries are blank.)

In order to assist in the analysis of the machine independent results presented in Tables 1-9, we provide a number of figures that provide visualizations of the tabular data:

- In Figures 1-27, we plot **Final NINT** vs. *tol* for a range of $p$ values for three of the codes (BACOL/ST: $p = 3, ..., 11$, BACOLI/LE: $p = 4, ..., 11$, BACOLI/ST: $p = 4, ..., 11$). We consider $tol = 10^{-2}, 10^{-3}, \ldots, 10^{-10}$. We provide results for the nine test problems identified above. There is one plot for each code/problem combination.

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

- In Figures 28-54, we plot **Final NINT** vs. $p$ for the range of tolerances $tol = 10^{-2}, 10^{-3}, \ldots, 10^{-10}$. We provide results for the nine test problems identified above. We consider BACOL/ST, BACOLI/LE, and BACOLI/ST. Each plot shows results for a range of $p$ values; BACOL/ST: $p = 3, ..., 11$, BACOLI/LE: $p = 4, ..., 11$, and BACOLI/ST: $p = 4, ..., 11$. There is one plot for each code/problem combination.

  From these plots we see that **Final NINT** has roughly the same small value for all $p$ values when the tolerance is coarse. This is also true for sharper tolerances when the $p$ value is large. However, for sharper tolerances and smaller $p$ values, the value of **Final NINT** is quite large. That is, when $p$ is small, the value of **Final NINT** grows quite dramatically as the tolerance decreases. We also note that BACOLI/ST has a smaller **Final NINT** value than the other codes do when $p$ is small and the tolerance is sharp.

- In Figures 55-81, we plot **Accepted Time Steps** vs. *tol* for a range of $p$ values for each code/problem combination. The codes and corresponding $p$ value ranges are BACOL/ST: $p = 3, ..., 11$, BACOLI/LE: $p = 4, ..., 11$, and BACOLI/ST: $p = 4, ..., 11$. We consider $tol = 10^{-2}, 10^{-3}, \ldots, 10^{-10}$. We consider all nine test problems. There is one plot for each code/problem combination.

  From these plots we see that over all $p$ values, the **Accepted Time Steps** values grow approximately linearly (on a log-log scale) as the *tol* values

decrease. Furthermore, the results are generally independent of $p$.

- In Figures 82-108, we plot **Remeshings** vs. *tol* for a range of $p$ values, for each code/problem combination. The codes and corresponding $p$ value ranges are BACOL/ST: $p = 3, ..., 11$, BACOLI/LE: $p = 4, ..., 11$, and BACOLI/ST: $p = 4, ..., 11$. We consider $tol = 10^{-2}, 10^{-3}, \ldots, 10^{-10}$. We consider all nine test problems. There is one plot for each code/problem combination.

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

- In Figures 109-144, we plot the number of ABD matrix factorizations, i.e., **Calls to CRDCMP**, and the number of backsolves of ABD systems, i.e., **Calls to CRSLVE**, vs. *tol* for each of the codes BACOL/ST, BACOLI/LE, and BACOLI/ST. We consider $tol = 10^{-2}, 10^{-3}, \ldots, 10^{-10}$. We consider all nine problems and $p$ values 4, 5, 7, and 9. There is one plot for each $p$ value/problem combination.

  From these plots we see that, for a given code, the **Calls to CRDCMP** and **Calls to CRSLVE** grow approximately linearly (on a log-log scale) as the *tol* value decreases. The **Calls to CRDCMP** value is about one order of magnitude smaller than the **Calls to CRSLVE** value. The **Calls to CRDCMP** and **Calls to CRSLVE** values for BACOL/ST are typically double those of BACOLI/ST and BACOLI/LE. The only exception is for some problems when $p$ is small; in such cases, the number of calls to CRDCMP performed by BACOL and BACOLI is about the same.

From an examination of Tables 1-9, coupled with the observations given above that follow from the figures that provide visualizations of the data from the tables, we see from that while the performances of BACOL/ST, BACOL/LE, BACOLI/ST, and BACOLI/LE of course exhibit some relative variations over the $p$ and *tol* values and problems we consider, several general observations regarding all codes can be made:

- **Final NINT:** For smaller $p$ values, **Final NINT** increases as *tol* decreases. In contrast, for larger $p$ values, **Final NINT** is independent of *tol*. For the coarsest *tol* value, **Final NINT** is independent of $p$; however, for the sharper *tol* values, **Final NINT** is larger for the smallest $p$ value, and decreases as $p$ increases. Combining these points, we see that **Final NINT** is largest when $p$ is smallest and *tol* is at its sharpest. As well, we note that when $p$ is small and *tol* is sharp, the LE codes require larger **Final NINT** values than do the ST codes; otherwise, the **Final NINT** values employed by the codes are quite similar.

- **Accepted Time Steps:** The number of **Accepted Time Steps** increases as *tol* decreases, as expected. For given $p$ and *tol* values, the

number of **Accepted Time Steps** used by the four codes is generally about the same and is independent of $p$.

- **Remeshings:** Compared to the total number of **Accepted Time Steps**, there are relatively few steps for which a spatial remeshing is required. The number of **Remeshings** is generally smaller when the tolerance is coarser and decreases as $p$ increases. For given $p$ and $tol$ values, the number of **Remeshings** performed by the four codes is generally about the same but there is a slight tendency for the LE codes to perform more **Remeshings** than the ST codes.

- **Cold Starts:** The number of **Cold Starts** is quite insignificant compared to the total number of **Accepted Time Steps**, across all codes and $p$, $tol$ combinations.

- **Calls to CRDCMP/Calls to CRSLVE:** The total number of ABD matrix factorizations (i.e., calls to CRDCMP) and ABD system back-solves (i.e., calls to CRSLVE) performed by BACOL/ST or BACOL/LE is roughly about twice that of BACOLI/ST or BACOLI/LE. (And this generally leads to timing results that show that BACOLI/ST and BACOLI/LE are about twice as fast as BACOL/ST or BACOL/LE.) The only exception is for the case, $p = 4, tol = 10^{-8}$, where BACOLI/ST and BACOLI/LE use about the same number of **calls to CRDCMP** as BACOL/ST and BACOL/LE; however, even in this case, BACOLI/ST and BACOLI/LE use about half as many **calls to CRSLVE** as BACOL/ST and BACOL/LE. For a given code, the number of **calls to CRDCMP** and **calls to CRSLVE** increases substantially as $tol$ decreases. The number of **calls to CRDCMP** is typically about one tenth of the number of **calls to CRSLVE**. There appears to be no significant difference in the number of **calls to CRDCMP** and **calls to CRSLVE** based on which type of error control is employed.

## 3.3 Machine Dependent Timing Results

We next provide machine dependent timing results for each code applied to the nine test problems identified earlier in this report. For all problems except **CSRM**, these tests were conducted on a system with two Intel(R) Xeon(R) CPU E5420 @ 2.50GHz processors. The operating system was Ubuntu 12.04.5 LTS, and the Fortran compiler was GNU Fortran (Ubuntu/Linaro 4.6.3-lubuntu5) 4.6.3. For the **CSRM** test problem, 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 10-18. Instances where

a given code failed correspond to blank table entries; these occurred for BACOLI/ST on **TLBEx6** and **TLBEx12** with $\epsilon = 10^{-3}$, $p = 4$, $tol = 10^{-6}$, BACOLI/LE on **OLBE** and **TLBEx12** with $\epsilon = 10^{-4}$, $p = 4$, $tol = 10^{-10}$, and for certain runs on **CSRM** as follows: BACOL/ST $p = 10$, $tol = 10^{-8}$ and $p = 8, 9$, $tol = 10^{-10}$; BACOL/LE $p = 11$, $tol = 10^{-8}$ and $p = 9, 10$, $tol = 10^{-10}$; BACOLI/ST $p = 6, 11$, $tol = 10^{-6}$, $p = 9$, $tol = 10^{-10}$; and BACOLI/LE $p = 11$, $tol = 10^{-8}$ and $tol = 10^{-10}$.

From these tables we see that, for midrange to high $p$ values the BACOLI/ST and BACOLI/LE codes are faster - in many cases twice as fast - as the BACOL/ST and BACOL/LE codes. For the smallest $p$ value ($p = 4$), the BACOLI/LE code sometimes is as expensive as the BACOL/ST and BACOL/LE codes. For a given problem, BACOLI/ST always has the fastest time. *For sharper tolerances, the use of BACOLI/ST with a larger p value gives the fastest time.*

As an example, we consider one table (Table 11) in more detail; this table gives timing results for **OLBE** with $\epsilon = 10^{-4}$. Over all $p$ values and $tol$ values, the BACOLI/ST and BACOLI/LE codes are faster - in many cases twice as fast - as the BACOL/ST and BACOL/LE codes, with one exception: BACOLI/LE when $p = 4$ and $tol = 10^{-8}$ where it has the same cost as BACOL/ST.

As mentioned above, over all problems, and over all four tolerances, BACOLI/ST is always the fastest code. (We note that these are tolerance vs. CPU time comparisons; a more relevant comparison is accuracy vs. CPU time; we discuss these types of comparisons later in this report.) For a given tolerance, the fastest run by BACOLI/ST depends on $p$ as follows:

- **OLBE** $\epsilon = 10^{-3}$: $tol = 10^{-4}, 10^{-6} \Rightarrow p = 4, 5$; $tol = 10^{-8} \Rightarrow p = 6$; $tol = 10^{-10} \Rightarrow p = 7, 8, 9$.
- **OLBE** $\epsilon = 10^{-4}$: $tol = 10^{-4}, 10^{-6} \Rightarrow p = 4$; $tol = 10^{-8} \Rightarrow p = 6$; $tol = 10^{-10} \Rightarrow p = 7$.
- **TLBE** $\epsilon = 10^{-3}$: $tol = 10^{-4} \Rightarrow p = 4$; $tol = 10^{-6} \Rightarrow p = 5$; $tol = 10^{-8} \Rightarrow p = 6$; $tol = 10^{-10} \Rightarrow p = 8$.
- **TLBE** $\epsilon = 10^{-4}$: $tol = 10^{-4}, 10^{-6} \Rightarrow p = 4$; $tol = 10^{-8} \Rightarrow p = 6$; $tol = 10^{-10} \Rightarrow p = 7$.
- **TLBEx6** $\epsilon = 10^{-3}$: $tol = 10^{-4}, 10^{-6} \Rightarrow p = 5$; $tol = 10^{-8}, tol = 10^{-10} \Rightarrow p = 6$.
- **TLBEx6** $\epsilon = 10^{-4}$: $tol = 10^{-4}, 10^{-6} \Rightarrow p = 4$; $tol = 10^{-8} \Rightarrow p = 5$; $tol = 10^{-10} \Rightarrow p = 7$.
- **TLBEx12** $\epsilon = 10^{-3}$: $tol = 10^{-4} \Rightarrow p = 4$; $tol = 10^{-6} \Rightarrow p = 5$; $tol = 10^{-8}, 10^{-10} \Rightarrow p = 6$.
- **TLBEx12** $\epsilon = 10^{-4}$: $tol = 10^{-4}, 10^{-6}, 10^{-8} \Rightarrow p = 4$; $tol = 10^{-10} \Rightarrow p = 6$.

- **CSRM**: $tol = 10^{-4}, 10^{-6} \Rightarrow p = 4$; $tol = 10^{-8} \Rightarrow p = 5$; $tol = 10^{-10} \Rightarrow p = 6$.

We note that for sharper tolerances, higher $p$ values lead to the fastest execution times. However, we also note that for larger systems of PDEs, somewhat smaller $p$ values correspond to faster execution times, although even for the largest system considered, **TLBEx12**, we still see that, at the sharpest tolerance, $10^{-10}$, a $p$ value of 6 corresponds to the fastest execution time.

In Table 19, we provide Timing Ratio Averages for the nine test problems. A timing ratio for a given $p$ and $tol$ is the ratio of the time spent by one code divided by the time spent by another. The second column of the table gives, for each problem, the average, over all $p$ and $tol$ combinations, of the BACOL/LE vs. BACOL/ST ratios. The third and fourth columns give the average, over all $p$ and $tol$ combinations, of the BACOLI/ST vs. BACOL/ST ratios and the BACOLI/LE vs. BACOL/ST ratios, respectively. The last column gives the average, over all $p$ and $tol$ combinations, of the BACOLI/LE vs. BACOLI/ST ratios. We see that BACOL/LE is typically slightly faster than BACOL/ST, BACOLI/ST is typically almost twice as fast as BACOL/ST, BACOLI/LE is typically about 60% faster than BACOL/ST, and BACOLI/LE is typically slightly slower than BACOLI/ST.

In Table 20, we provide Timing Ratio Averages for $p = 4, \ldots, 11$. The second column gives the average of the BACOL/LE vs. BACOL/ST ratios over all problems and tolerances. The third, fourth, and fifth columns give corresponding results for the BACOLI/ST vs. BACOL/ST, BACOLI/LE vs. BACOL/ST, and BACOLI/LE vs. BACOLI/ST ratios, respectively. We see that for smaller $p$ values BACOL/LE is slower than BACOL/ST but faster for larger $p$ values, BACOLI/ST is substantially faster than BACOL/ST in general, approaching 50% faster for larger $p$ values, BACOLI/LE is slower than BACOL/ST for $p = 4$ but faster for higher $p$ values, approaching 50% faster for larger $p$ values, and BACOLI/LE is much slower than BACOLI/ST for $p = 4$, but is essentially just as fast as BACOLI/ST for larger $p$ values.

The average of the ratios over all problems, $p$ values, and tolerances are

$$\frac{BACOL/LE}{BACOL/ST} = 0.95, \quad \frac{BACOLI/ST}{BACOL/ST} = 0.51,$$

$$\frac{BACOLI/LE}{BACOL/ST} = 0.60, \quad \frac{BACOLI/LE}{BACOLI/ST} = 1.17.$$

## 3.4 Error vs. Execution Time across Codes

Here we present error vs. execution time results for the **OLBE** and **TLBE**, with $\epsilon = 10^{-3}$ and $10^{-4}$, and the **CSRM**, for BACOL/ST, BACOL/LE, BACOLI/ST, and BACOLI/LE. We consider $p$ values from 4 to 11. Since the OLBE and TLBE problems have known exact solutions it is straightforward to compute the error in the numerical solutions. For the **CSRM**, we computed a high accuracy numerical reference solution using BACOL with a very sharp

tolerance. 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 145-184, 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 the **OLBE** and **TLBE** problems, we see that BACOL/ST and BACOL/LE have comparable execution times, BACOLI/ST and BACOLI/LE have comparable execution times, and BACOLI/ST and BACOLI/LE are consistently less expensive than BACOL/ST and BACOL/LE. We also see that, for large to medium errors, the LE codes are slightly faster than the ST codes. The only exception is BACOLI/LE with $p = 4$; in this case the code is significantly faster than the other codes for large errors but significantly slower for small errors. *We note this last observation is in apparent disagreement with the conclusions drawn earlier from the timing tables, in which it was observed that,* **for a given tolerance***, BACOLI/ST was always faster than BACOLI/LE.* **It is important to note that in this section and the next two we are comparing error achieved rather than tolerance requested with the execution time.** *From the figures considered in this subsection we see that BACOLI/LE delivers a smaller error for a given tolerance request. This is understandable since it is controlling the error in a collocation solution that is one order lower than the collocation solution that is returned. We therefore expect the error of the returned solution to be smaller than the requested tolerance.* **Furthermore, this suggests that the tolerance employed when BACOLI runs in LE should be scaled so that it is somewhat coarser than the requested user tolerance. A simple analysis shows that when BACOLI is running in LE mode, the tolerance should be divided by $h$, where $h$ is the subinterval size. Since $h < 1$, this will imply that a coarser (i.e., larger) tolerance is used to control the error estimate that is in use when the code is running in LE mode. However, our experimental results indicate that this is only relevant for large to medium errors; for small errors the ST error control modes deliver comparable or better results in terms of efficiency.**

For the **CSRM**, we see that for larger $p$ values, BACOLI is faster than BACOL in either error control mode, BACOL/LE is slightly faster than BACOL/ST, and BACOLI/ST and BACOLI/LE have similar performance. For $p = 4$, BACOLI/ST is faster than all the other codes, BACOLI/LE is faster than BACOL/LE, for coarse tolerances, both BACOL/LE and BACOLI/LE are faster than BACOL/ST but for sharp tolerances both BACOL/LE and BACOLI/LE are slower than BACOL/ST. For $p = 5$, BACOL/ST is faster than all the other codes, BACOLI/LE is faster than both BACOL/ST and BACOL/LE, for coarse tolerances, BACOL/LE is faster than BACOL/ST but for sharp tolerances both BACOL/LE is slower than BACOL/ST.

The comparisons among these codes can be seen more clearly if we plot errors vs. execution times of BACOL/LE, BACOLI/ST, and BACOLI/LE relative to that of BACOL/ST. The plots were developed as follows. We describe this

process for the BACOL/LE data.

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

In Figures 185-224, we plot error achieved vs. relative CPU time, for BACOL/LE, BACOLI/ST, and BACOLI/LE relative to BACOL/ST, for a given problem and $p$ value. We can now see more clearly that, generally, BACOLI/ST and BACOLI/LE are generally less expensive than BACOL/ST and BACOL/LE for a given problem and $p$ value. The average costs for BACOLI/ST and BACOLI/LE are about 50% of the cost for BACOL/ST and about 60% of the cost of BACOL/LE. As well, we see that for larger errors, i.e., coarser tolerances, the LE error control codes are less expensive than the ST error control codes but as the error gets smaller, i.e., as the tolerance gets sharper, the LE error control codes become comparable to or sometimes more expensive than the ST error control codes.

The plots that correspond to $p = 4$ deserve specific mention. In this case, for each problem, we see BACOLI/LE is much faster than the other codes for coarser tolerances and much worse than the other codes for sharper tolerances. Other observations can be made on a problem specific basis:

- **OLBE, $\epsilon = 10^{-3}$:** BACOL/LE is slightly faster than BACOL/ST for large errors and gets somewhat faster as the error gets smaller. BACOLI/ST is faster than BACOL/ST and BACOL/LE for large errors and gets relatively faster as the error gets smaller.
- **OLBE, $\epsilon = 10^{-4}$:** BACOL/LE is substantially faster than BACOL/ST for large errors but its performance becomes worse as the error gets smaller to the point where, for the smallest errors it has the same performance as BACOL/ST. BACOLI/ST is faster than BACOL/ST but slower than BACOL/LE for large errors but becomes much faster than either as the error gets smaller.

- **TLBE, $\epsilon = 10^{-3}$:** BACOL/LE is slower than BACOL/ST for large errors and gets somewhat faster as the error gets smaller; it is slightly faster than BACOL/ST for small errors. BACOLI/ST is slightly faster than BACOL/ST for all errors.

- **TLBE, $\epsilon = 10^{-4}$:** BACOL/LE is somewhat faster than BACOL/ST for large errors but its performance becomes worse as the error gets smaller to the point where, for the smallest errors it substantially worse performance than BACOL/ST. BACOLI/ST is faster than BACOL/ST for large errors and becomes much faster as the error gets smaller.

- **CSRM**: BACOLI/ST is about twice as fast as BACOL/ST; for coarse tolerances, both BACOL/LE and BACOLI/LE are about twice as fast as BACOL/ST but for sharp tolerances they are both substantially slower than BACOL/ST. For this problem, similar behaviour is also observed for $p = 5$.

### 3.5 Error vs. Execution Time across $p$ Values

Here we consider error vs. execution time results for BACOL/ST, BACOL/LE, BACOLI/ST, BACOLI/LE, over a range of $p$ values, for the **OLBE** and the **TLBE** with $\epsilon = 10^{-3}$ and $10^{-4}$ and for **CSRM**. *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 225-244, provide plots, for each problem and code, showing the performance of the codes with respect to error vs. execution time, over a range of $p$ values.

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

## 4 Summary, Conclusions and Future Work

B-spline collocation software for the numerical solution of BVODEs and 1D PDEs has been widely used for several decades. The more recently developed members of this family of packages for PDEs feature both spatial and temporal error control. The newest code, BACOLI, improves upon the efficiency of the earlier package, BACOL, by employing new interpolation-based schemes for the computation of spatial error estimates. As well, BACOLI implements, as options through its two error estimation schemes, a standard (ST) error control scheme as well as an alternative error control scheme known as local extrapolation (LE) error control.

This report presents a detailed examination of the performance of BACOL and BACOLI in which several important machine independent performance measures are considered. It is shown that the new error estimation schemes and both error control schemes generally give comparable performance measures, except for number of ABD matrix factorizations and ABD system solves, where the BACOLI/ST and BACOLI/LE codes use approximately half as many as do the BACOL/ST and BACOL/LE codes; this leads to substantial savings in execution time. These results also show that for small $p$ and sharp $tol$, the codes that run in LE error control mode require more spatial subintervals; this is observed to lead to greater execution costs for the LE codes, for sharp $tol$. Conversely, for coarser tolerances, the LE codes are observed to be relatively more efficient that the corresponding ST control codes.

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

There are several directions for future work. The results of this report suggest that it may be worthwhile to modify the BACOLI code in order to have it choose $p$ based on the tolerance requested, and that it may be advisable to vary the $p$ values over only an intermediate range of values since these give the best combination of efficiency and reliability. Also a modification of BACOLI to automate the choice of error control mode may be worthwhile since it appears that LE error control is better for coarser tolerances while ST error control is better for sharper tolerances.

Another project for future work involves the development and analysis of a new version of BACOLR that implements the interpolation-based spatial error estimation schemes that are implemented in BACOLI. An investigation of COLSYS/COLNEW to verify similar performance with respect to the choice of $p$, followed by a modification of these codes to automate the choice of $p$, may be warranted. Finally, the results of this report will be used to inform the ongoing development of B-spline Gaussian collocation software for 2D PDEs.

# References


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

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

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

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

[5] B. Bialecki, G. Fairweather, and J. C. López-Marcos. The Crank-Nicolson Hermite cubic orthogonal spline collocation method for the heat equation with nonlocal boundary conditions. *Adv. Appl. Math. Mech.*, 5(4):442–460, 2013.

[6] B. Bialecki, G. Fairweather, and J. C. López-Marcos. The extrapolated Crank-Nicolson orthogonal spline collocation method for a quasilinear parabolic problem with nonlocal boundary conditions. *J. Sci. Comput.*, 62(1):265–283, 2015.

[7] 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.

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

[9] 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.

[10] X. Ding, Q.-J. Meng, and L.-P. Yin. Discrete-time orthogonal spline collocation method for one-dimensional sine-Gordon equation. *Discrete Dyn. Nat. Soc.*, pages Art. ID 206264, 8, 2015.

[11] G. Fairweather, X. Yang, D. Xu, and H. Zhang. An ADI Crank-Nicolson orthogonal spline collocation method for the two-dimensional fractional diffusion-wave equation. *J. Sci. Comput.*, 65(3):1217–1239, 2015.

[12] R.I. Fernandes, B. Bialecki, and G. Fairweather. An ADI extrapolated Crank-Nicolson orthogonal spline collocation method for nonlinear reaction-diffusion systems on evolving domains. *J. Comput. Phys.*, 299:561–580, 2015.

[13] R.I. Fernandes and G. Fairweather. An ADI extrapolated Crank-Nicolson orthogonal spline collocation method for nonlinear reaction-diffusion systems. *J. Comput. Phys.*, 231(19):6248–6267, 2012.

[14] 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.

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

[16] W. Huang and R.D. Russell. *Adaptive Moving Mesh Methods*, volume 174 of *Applied Mathematical Sciences*. Springer, New York, 2011.

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

[18] C. Li, T. Zhao, W. Deng, and Y. Wu. Orthogonal spline collocation methods for the subdiffusion equation. *J. Comput. Appl. Math.*, 255:517–528, 2014.

[19] Z. Li and P.H. Muir. B-spline Gaussian collocation software for two-dimensional parabolic PDEs. *Adv. Appl. Math. Mech.*, 5:528–547, 2013.

[20] NAG Numerical Algorithms Group Fortran library. *d02tlc*. The Numerical Algorithms Group, Ltd., Wilkinson House, Oxford, UK.

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

[22] 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.

[23] Scilab. *bvode, bvodeS*. Scilab Enterprises, 143 bis rue Yves Le Coz, 78000 Versailles, France.

[24] Weiwei Sun. *B*-spline collocation methods for elasticity problems. In *Scientific computing and applications (Kananaskis, AB, 2000)*, volume 7 of *Adv. Comput. Theory Pract.*, pages 133–141. Nova Sci. Publ., Huntington, NY, 2001.

[25] P. Virtanen. *scikits.bvp1lg 0.2.8., https://pv.github.io/scikits.bvp1lg/.*

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] X. Yang, H. Zhang, and D. Xu. Orthogonal spline collocation method for the two-dimensional fractional sub-diffusion equation. *J. Comput. Phys.*, 256:824–837, 2014.

[31] H. Zhang, X. Yang, and X. Han. Discrete-time orthogonal spline collocation method with application to two-dimensional fractional cable equation. *Comput. Math. Appl.*, 68(12, part A):1710–1722, 2014.

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

| *tol* | $10^{-4}$ | $10^{-6}$ | $10^{-8}$ |
|---|---|---|---|
| *p* | 4 | | |
| BACOL/ST | 14, 1144, 78<br>(0)[378, 3980] | 23, 2528, 114<br>(0)[592, 7934] | 48, 6814, 153<br>(0)[1116, 20328] |
| BACOL/LE | 17, 1107, 105<br>(1)[516, 4172] | 42, 2455, 106<br>(0)[502, 7108] | 111, 5708, 129<br>(0)[606, 13560] |
| BACOLI/ST | 15, 1241, 71<br>(0)[186, 2095] | 26, 3171, 119<br>(6)[553, 4791] | 52, 7349, 151<br>(12)[1127, 10251] |
| BACOLI/LE | 18, 1179, 122<br>(0)[264, 2224] | 45, 2223, 155<br>(0)[339, 3525] | 128, 5715, 478<br>(0)[989, 9999] |
| *p* | 5 | | |
| BACOL/ST | 12, 1159, 62<br>(0)[322, 3952] | 16, 3141, 127<br>(0)[732, 9846] | 30, 6816, 130<br>(1)[934, 18614] |
| BACOL/LE | 14, 1144, 78<br>(0)[378, 3980] | 23, 2528, 114<br>(0)[592, 7934] | 48, 6814, 153<br>(0)[1116, 20328] |
| BACOLI/ST | 15, 1143, 63<br>(0)[153, 1850] | 19, 2999, 108<br>(0)[495, 4814] | 35, 6270, 113<br>(0)[1084, 9377] |
| BACOLI/LE | 15, 1160, 95<br>(0)[217, 2070] | 22, 2432, 126<br>(0)[301, 3823] | 57, 5097, 137<br>(0)[366, 6953] |
| *p* | 7 | | |
| BACOL/ST | 10, 1516, 45<br>(0)[374, 4954] | 12, 3267, 87<br>(1)[866, 9998] | 16, 6287, 149<br>(0)[2002, 18614] |
| BACOL/LE | 14, 1175, 63<br>(0)[300, 3808] | 15, 2978, 83<br>(0)[576, 8694] | 21, 6333, 152<br>(1)[1340, 18136] |
| BACOLI/ST | 12, 1285, 49<br>(0)[123, 2028] | 15, 3139, 63<br>(0)[380, 4628] | 21, 7520, 125<br>(2)[1100, 9482] |
| BACOLI/LE | 11, 1179, 67<br>(0)[167, 2038] | 15, 2947, 90<br>(1)[308, 4381] | 23, 6732, 174<br>(5)[697, 9821] |
| *p* | 9 | | |
| BACOL/ST | 11, 1404, 42<br>(0)[288, 4612] | 15, 2853, 57<br>(0)[388, 7800] | 14, 6379, 99<br>(1)[1978, 17962] |
| BACOL/LE | 10, 1702, 43<br>(0)[446, 5472] | 15, 3008, 61<br>(0)[450, 8542] | 15, 6408, 124<br>(1)[2248, 18608] |
| BACOLI/ST | 10, 1790, 38<br>(0)[202, 2789] | 13, 3418, 61<br>(0)[485, 5193] | 15, 7070, 114<br>(0)[1133, 10502] |
| BACOLI/LE | 10, 1348, 55<br>(0)[137, 2249] | 15, 2767, 69<br>(0)[210, 3915] | 15, 6578, 142<br>(2)[1196, 10194] |

Table 1: *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 (Cold Starts) [Calls to CRDCMP, CRSLVE].*

| $tol$ | $10^{-4}$ | $10^{-6}$ | $10^{-8}$ |
|---|---|---|---|
| $p$ | 4 | | |
| BACOL/ST | 15, 13255, 748<br>(2)[3992, 46068] | 23, 33410, 1148<br>(2)[5962, 100340] | 47, 66476, 1261<br>(0)[11542, 183320] |
| BACOL/LE | 18, 12414, 892<br>(1)[4122, 44662] | 33, 31536, 1207<br>(0)[5688, 97710] | 102, 60684, 1127<br>(0)[6688, 160918] |
| BACOLI/ST | 16, 25839, 748<br>(5)[2365, 23772] | 26, 31936, 887<br>(0)[5896, 52839] | 48, 67746, 975<br>(1)[9837, 99568] |
| BACOLI/LE | 22, 10966, 1177<br>(1)[2497, 21477] | 43, 24046, 1420<br>(0)[3023, 37985] | 102, 47041, 3619<br>(0)[7331, 80019] |
| $p$ | 5 | | |
| BACOL/ST | 15, 12509, 598<br>(1)[3114, 41540] | 17, 33265, 1236<br>(2)[12194, 110806] | 29, 60497, 1248<br>(2)[13752, 169134] |
| BACOL/LE | 15, 13255, 748<br>(2)[3992, 46068] | 23, 33410, 1148<br>(2)[5962, 100340] | 47, 66476, 1261<br>(0)[11542, 183320] |
| BACOLI/ST | 15, 13870, 665<br>(2)[2534, 24732] | 20, 31848, 928<br>(2)[5787, 52629] | 34, 111446, 1015<br>(2)[9672, 105093] |
| BACOLI/LE | 17, 15093, 984<br>(9)[2372, 23732] | 25, 29375, 1160<br>(1)[2849, 45314] | 48, 66399, 1265<br>(6)[4190, 94728] |
| $p$ | 7 | | |
| BACOL/ST | 15, 12039, 526<br>(0)[2560, 39606] | 14, 33137, 714<br>(3)[9822, 101494] | 18, 126175, 1499<br>(10)[27826, 211996] |
| BACOL/LE | 13, 11679, 560<br>(0)[2632, 38986] | 15, 33866, 1017<br>(3)[11672, 109814] | 22, 93925, 1415<br>(7)[22668, 190738] |
| BACOLI/ST | 14, 14793, 408<br>(0)[1642, 23590] | 15, 36456, 977<br>(1)[7237, 60070] | 20, 68939, 1210<br>(2)[11649, 103649] |
| BACOLI/LE | 15, 12279, 637<br>(0)[1501, 20601] | 15, 63413, 1252<br>(16)[6730, 58529] | 22, 61944, 1396<br>(12)[9244, 93449] |
| $p$ | 9 | | |
| BACOL/ST | 13, 17209, 383<br>(1)[3806, 54476] | 15, 31805, 545<br>(1)[6916, 93080] | 15, 68299, 978<br>(6)[21200, 187862] |
| BACOL/LE | 13, 14870, 436<br>(0)[3126, 47820] | 15, 32995, 590<br>(4)[8638, 98676] | 15, 81038, 1467<br>(10)[29082, 223006] |
| BACOLI/ST | 14, 16869, 348<br>(0)[1835, 26241] | 15, 35920, 517<br>(0)[4918, 53757] | 17, 98581, 1331<br>(4)[13004, 109650] |
| BACOLI/LE | 14, 13109, 536<br>(0)[1530, 21757] | 15, 34218, 671<br>(2)[4775, 52107] | 15, 103353, 1680<br>(34)[15393, 119360] |

Table 2: *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 (Cold Starts) [Calls to CRDCMP, CRSLVE].*

| *tol* | $10^{-4}$ | $10^{-6}$ | $10^{-8}$ |
|---|---|---|---|
| *p* | 4 | | |
| BACOL/ST | 15, 908, 62<br>(0)[300, 3064] | 23, 1764, 91<br>(0)[462, 5484] | 52, 4468, 90<br>(0)[548, 11732] |
| BACOL/LE | 15, 839, 83<br>(0)[394, 3220] | 39, 1880, 91<br>(0)[412, 5472] | 125, 3365, 87<br>(0)[420, 8176] |
| BACOLI/ST | 15, 875, 72<br>(0)[200, 1560] | 26, 7510, 304<br>(7)[1157, 4323] | 57, 5106, 311<br>(0)[1408, 8051] |
| BACOLI/LE | 19, 727, 91<br>(0)[199, 1508] | 46, 1687, 126<br>(0)[275, 2748] | 135, 3384, 412<br>(0)[854, 7076] |
| *p* | 5 | | |
| BACOL/ST | 13, 893, 53<br>(0)[264, 3002] | 17, 2194, 103<br>(0)[560, 6900] | 30, 4557, 95<br>(0)[626, 12200] |
| BACOL/LE | 15, 908, 62<br>(0)[300, 3064] | 23, 1764, 91<br>(0)[462, 5484] | 52, 4468, 90<br>(0)[548, 11732] |
| BACOLI/ST | 14, 899, 48<br>(0)[122, 1466] | 19, 2252, 68<br>(0)[227, 3363] | 31, 4857, 82<br>(0)[647, 7000] |
| BACOLI/LE | 15, 885, 73<br>(0)[165, 1586] | 23, 1835, 95<br>(0)[224, 2791] | 52, 4060, 108<br>(0)[278, 5344] |
| *p* | 7 | | |
| BACOL/ST | 14, 920, 41<br>(0)[232, 2980] | 14, 2363, 70<br>(0)[482, 7056] | 15, 4872, 106<br>(0)[1238, 13888] |
| BACOL/LE | 15, 889, 45<br>(0)[228, 2846] | 14, 2189, 80<br>(0)[466, 6600] | 20, 4957, 103<br>(0)[756, 13490] |
| BACOLI/ST | 13, 950, 40<br>(0)[100, 1495] | 15, 2350, 72<br>(0)[373, 3804] | 17, 4851, 106<br>(0)[827, 7523] |
| BACOLI/LE | 15, 859, 54<br>(0)[128, 1424] | 15, 2145, 94<br>(0)[251, 3320] | 21, 4885, 117<br>(0)[423, 6987] |
| *p* | 9 | | |
| BACOL/ST | 15, 912, 34<br>(0)[180, 2810] | 15, 2026, 52<br>(0)[446, 6076] | 15, 4957, 74<br>(0)[1414, 13852] |
| BACOL/LE | 13, 974, 38<br>(0)[200, 3076] | 15, 2284, 56<br>(0)[434, 6666] | 15, 4732, 91<br>(0)[1316, 13378] |
| BACOLI/ST | 11, 1112, 36<br>(0)[138, 1773] | 14, 2582, 51<br>(0)[314, 3805] | 14, 5414, 95<br>(0)[896, 8174] |
| BACOLI/LE | 12, 985, 46<br>(0)[132, 1647] | 15, 2149, 57<br>(0)[214, 3138] | 15, 4899, 92<br>(0)[603, 7010] |

Table 3: *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 (Cold Starts) [Calls to CRDCMP, CRSLVE].*

| *tol* | $10^{-4}$ | $10^{-6}$ | $10^{-8}$ |
|---|---|---|---|
| *p* | 4 | | |
| BACOL/ST | 15, 9190, 631<br>(3)[2978, 32660] | 22, 21463, 895<br>(0)[4080, 65304] | 45, 47906, 878<br>(0)[5934, 129392] |
| BACOL/LE | 17, 8758, 785<br>(2)[3446, 33244] | 35, 19316, 971<br>(0)[4174, 60556] | 101, 43917, 978<br>(0)[4770, 112850] |
| BACOLI/ST | 15, 13408, 664<br>(3)[1695, 17267] | 26, 27659, 805<br>(21)[3480, 36468] | 45, 47795, 850<br>(1)[6668, 70128] |
| BACOLI/LE | 19, 8109, 944<br>(1)[1946, 16363] | 37, 16073, 1194<br>(0)[2463, 26678] | 102, 39866, 3240<br>(0)[6517, 69025] |
| *p* | 5 | | |
| BACOL/ST | 15, 9414, 518<br>(7)[2442, 31566] | 17, 23632, 959<br>(3)[6866, 76586] | 31, 46928, 1016<br>(3)[8870, 131120] |
| BACOL/LE | 15, 9190, 631<br>(3)[2978, 32660] | 22, 21463, 895<br>(0)[4080, 65304] | 45, 47906, 878<br>(0)[5934, 129392] |
| BACOLI/ST | 15, 9788, 529<br>(1)[1490, 17173] | 20, 23009, 736<br>(0)[3501, 36818] | 32, 49856, 824<br>(1)[7096, 72951] |
| BACOLI/LE | 14, 26871, 757<br>(7)[1722, 17077] | 24, 19840, 926<br>(0)[2075, 30449] | 48, 43452, 1016<br>(5)[2805, 60080] |
| *p* | 7 | | |
| BACOL/ST | 15, 11258, 404<br>(4)[2692, 36358] | 15, 25737, 764<br>(0)[8272, 82330] | 16, 67381, 1136<br>(6)[18164, 153674] |
| BACOL/LE | 15, 10406, 426<br>(4)[2360, 33524] | 15, 24377, 828<br>(3)[7360, 77954] | 21, 48146, 1051<br>(1)[12352, 137590] |
| BACOLI/ST | 14, 10948, 443<br>(0)[1306, 18236] | 14, 26589, 749<br>(0)[4773, 42928] | 19, 50641, 910<br>(0)[8487, 77688] |
| BACOLI/LE | 14, 10539, 524<br>(3)[1410, 18058] | 16, 24446, 957<br>(5)[3995, 40319] | 22, 47320, 1068<br>(3)[5610, 69369] |
| *p* | 9 | | |
| BACOL/ST | 14, 11998, 317<br>(0)[2444, 36884] | 15, 27513, 617<br>(3)[8314, 84106] | 15, 53550, 911<br>(4)[17892, 154372] |
| BACOL/LE | 14, 23588, 345<br>(13)[3056, 39262] | 14, 26859, 669<br>(2)[8512, 83750] | 15, 53881, 1105<br>(0)[19586, 161904] |
| BACOLI/ST | 15, 11873, 361<br>(0)[1493, 19034] | 14, 28481, 595<br>(1)[4513, 43823] | 15, 55694, 985<br>(2)[9741, 84509] |
| BACOLI/LE | 14, 13511, 433<br>(3)[1625, 19340] | 15, 27117, 734<br>(10)[4501, 42863] | 14, 75698, 1251<br>(12)[10323, 87784] |

Table 4: *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 (Cold Starts) [Calls to CRDCMP, CRSLVE].*

| *tol* | $10^{-4}$ | $10^{-6}$ | $10^{-8}$ |
|---|---|---|---|
| *p* | 4 | | |
| BACOL/ST | 12, 873, 64<br>(0)[316, 3096] | 21, 1816, 90<br>(0)[444, 5582] | 51, 4377, 94<br>(0)[586, 11714] |
| BACOL/LE | 15, 850, 82<br>(0)[388, 3240] | 39, 1918, 92<br>(0)[416, 5568] | 112, 3365, 90<br>(0)[432, 8224] |
| BACOLI/ST | 13, 891, 111<br>(0)[334, 1773] | —<br>— | 51, 4759, 181<br>(0)[901, 7077] |
| BACOLI/LE | 17, 745, 91<br>(0)[199, 1526] | 44, 1611, 112<br>(0)[245, 2539] | 134, 3384, 412<br>(0)[854, 7075] |
| *p* | 5 | | |
| BACOL/ST | 14, 857, 53<br>(0)[264, 2944] | 14, 2145, 94<br>(0)[514, 6594] | 30, 4593, 91<br>(0)[628, 12286] |
| BACOL/LE | 12, 873, 64<br>(0)[316, 3096] | 21, 1816, 90<br>(0)[444, 5582] | 51, 4377, 94<br>(0)[586, 11714] |
| BACOLI/ST | 14, 872, 51<br>(0)[126, 1437] | 18, 2269, 80<br>(0)[269, 3484] | 33, 5168, 84<br>(0)[670, 7397] |
| BACOLI/LE | 14, 862, 75<br>(0)[168, 1570] | 23, 1761, 98<br>(0)[227, 2734] | 51, 4054, 112<br>(0)[274, 5316] |
| *p* | 7 | | |
| BACOL/ST | 15, 909, 43<br>(0)[238, 2978] | 15, 2356, 67<br>(0)[426, 6858] | 15, 4956, 115<br>(0)[1308, 14328] |
| BACOL/LE | 15, 873, 44<br>(0)[224, 2774] | 15, 2167, 70<br>(0)[406, 6404] | 20, 4711, 102<br>(0)[764, 13040] |
| BACOLI/ST | 12, 969, 39<br>(0)[100, 1531] | 13, 2402, 73<br>(0)[399, 3843] | 18, 4832, 93<br>(0)[770, 7285] |
| BACOLI/LE | 15, 856, 53<br>(0)[127, 1434] | 14, 2194, 88<br>(0)[239, 3371] | 21, 5017, 106<br>(0)[368, 7089] |
| *p* | 9 | | |
| BACOL/ST | 15, 898, 35<br>(0)[198, 2874] | 15, 2312, 52<br>(0)[476, 6584] | 15, 4852, 76<br>(0)[1442, 13710] |
| BACOL/LE | 12, 980, 35<br>(0)[220, 3182] | 15, 2225, 54<br>(0)[412, 6480] | 14, 4956, 99<br>(0)[1440, 14262] |
| BACOLI/ST | 11, 1106, 35<br>(0)[134, 1763] | 15, 2368, 49<br>(0)[263, 3458] | 15, 5341, 82<br>(0)[837, 7959] |
| BACOLI/LE | 13, 977, 43<br>(0)[115, 1572] | 14, 2235, 65<br>(0)[265, 3378] | 15, 4817, 98<br>(0)[690, 7164] |

Table 5: *Machine independent results for the Two Layer Burgers equation*×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 (Cold Starts) [Calls to CRDCMP, CRSLVE].*

| $tol$ | $10^{-4}$ | $10^{-6}$ | $10^{-8}$ |
|---|---|---|---|
| $p$ | 4 | | |
| BACOL/ST | 14, 9346, 613<br>(12)[2896, 32612] | 19, 21873, 943<br>(1)[4308, 67264] | 44, 47092, 943<br>(0)[6616, 128772] |
| BACOL/LE | 17, 8857, 795<br>(2)[3488, 33766] | 32, 20871, 1072<br>(0)[4656, 65938] | 87, 43439, 1020<br>(0)[5724, 116722] |
| BACOLI/ST | 16, 9288, 699<br>(2)[1817, 17447] | 25, 22534, 772<br>(4)[3294, 35933] | 45, 48296, 826<br>(1)[6352, 70053] |
| BACOLI/LE | 19, 8103, 959<br>(2)[1980, 16428] | 39, 16262, 1194<br>(0)[2462, 26666] | 103, 39141, 3226<br>(1)[6511, 68219] |
| $p$ | 5 | | |
| BACOL/ST | 14, 9899, 521<br>(7)[2654, 33308] | 17, 23551, 976<br>(5)[7014, 76450] | 28, 46825, 1064<br>(3)[9346, 131442] |
| BACOL/LE | 14, 9346, 613<br>(12)[2896, 32612] | 19, 21873, 943<br>(1)[4308, 67264] | 44, 47092, 943<br>(0)[6616, 128772] |
| BACOLI/ST | 15, 9816, 523<br>(3)[1501, 17062] | 18, 37008, 756<br>(2)[3675, 37541] | 35, 49545, 882<br>(0)[7473, 73273] |
| BACOLI/LE | 15, 8988, 765<br>(1)[1759, 16817] | 23, 20060, 961<br>(1)[2178, 30941] | 46, 45188, 1067<br>(0)[2963, 63104] |
| $p$ | 7 | | |
| BACOL/ST | 15, 10841, 399<br>(6)[2422, 34782] | 15, 25576, 754<br>(3)[8314, 81886] | 16, 78797, 1130<br>(2)[18004, 154172] |
| BACOL/LE | 15, 10042, 476<br>(3)[2534, 33486] | 15, 24653, 899<br>(1)[7870, 80468] | 23, 95628, 1099<br>(8)[13204, 139426] |
| BACOLI/ST | 14, 10749, 417<br>(0)[1266, 17696] | 16, 26350, 717<br>(1)[4580, 42141] | 19, 50825, 906<br>(0)[8429, 77783] |
| BACOLI/LE | 15, 10569, 540<br>(6)[1457, 17900] | 15, 24456, 942<br>(1)[4032, 40437] | 22, 48071, 1070<br>(7)[5425, 69758] |
| $p$ | 9 | | |
| BACOL/ST | 15, 11863, 325<br>(0)[2466, 36356] | 14, 28769, 638<br>(3)[9344, 88620] | 15, 54217, 950<br>(3)[18700, 156698] |
| BACOL/LE | 14, 12065, 360<br>(6)[2766, 37744] | 15, 26579, 658<br>(0)[8234, 83140] | 15, 54894, 1115<br>(0)[19788, 164190] |
| BACOLI/ST | 15, 11626, 352<br>(0)[1431, 18607] | 15, 28684, 600<br>(0)[4604, 44084] | 14, 54695, 938<br>(1)[9308, 82389] |
| BACOLI/LE | 15, 10968, 425<br>(0)[1400, 17791] | 14, 34408, 765<br>(9)[4714, 44146] | 15, 61190, 1235<br>(6)[10315, 88796] |

Table 6: *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 (Cold Starts) [Calls to CRDCMP, CRSLVE].*

| *tol* | $10^{-4}$ | $10^{-6}$ | $10^{-8}$ |
|---|---|---|---|
| *p* | 4 | | |
| BACOL/ST | 13, 873, 66<br>(0)[330, 3096] | 21, 1845, 92<br>(0)[456, 5608] | 47, 4395, 98<br>(0)[628, 11894] |
| BACOL/LE | 15, 841, 83<br>(0)[394, 3234] | 39, 1918, 91<br>(0)[410, 5548] | 111, 3365, 91<br>(0)[436, 8240] |
| BACOLI/ST | 14, 912, 75<br>(0)[205, 1617] | —<br>— | 50, 5110, 178<br>(0)[892, 7741] |
| BACOLI/LE | 19, 771, 94<br>(0)[205, 1572] | 44, 1827, 119<br>(0)[258, 2814] | 136, 3384, 414<br>(0)[858, 7094] |
| *p* | 5 | | |
| BACOL/ST | 15, 920, 54<br>(0)[268, 3060] | 15, 2143, 96<br>(0)[522, 6660] | 29, 4624, 104<br>(0)[680, 12738] |
| BACOL/LE | 13, 873, 66<br>(0)[330, 3096] | 21, 1845, 92<br>(0)[456, 5608] | 47, 4395, 98<br>(0)[628, 11894] |
| BACOLI/ST | 15, 879, 50<br>(0)[126, 1445] | 18, 2164, 78<br>(0)[266, 3336] | 32, 4993, 94<br>(0)[745, 7416] |
| BACOLI/LE | 14, 856, 79<br>(0)[175, 1585] | 22, 1787, 99<br>(0)[233, 2769] | 52, 4193, 108<br>(0)[272, 5493] |
| *p* | 7 | | |
| BACOL/ST | 14, 929, 40<br>(0)[218, 2994] | 15, 2182, 69<br>(0)[480, 6588] | 15, 4949, 105<br>(0)[1208, 14010] |
| BACOL/LE | 14, 909, 46<br>(0)[234, 2902] | 14, 2212, 73<br>(0)[444, 6706] | 20, 4879, 107<br>(0)[876, 13542] |
| BACOLI/ST | 15, 857, 44<br>(0)[107, 1410] | 15, 2332, 62<br>(0)[334, 3539] | 19, 4874, 100<br>(0)[803, 7454] |
| BACOLI/LE | 15, 833, 55<br>(0)[130, 1413] | 15, 2181, 80<br>(0)[225, 3256] | 21, 4778, 110<br>(0)[395, 6725] |
| *p* | 9 | | |
| BACOL/ST | 15, 900, 34<br>(0)[196, 2828] | 15, 2046, 46<br>(0)[416, 5940] | 15, 4691, 75<br>(0)[1472, 13160] |
| BACOL/LE | 13, 992, 40<br>(0)[216, 3118] | 14, 2383, 59<br>(0)[516, 7136] | 14, 4946, 98<br>(0)[1518, 14252] |
| BACOLI/ST | 11, 1148, 36<br>(0)[133, 1816] | 15, 2363, 47<br>(0)[248, 3419] | 15, 5149, 89<br>(0)[834, 7761] |
| BACOLI/LE | 12, 993, 43<br>(0)[120, 1608] | 15, 2344, 61<br>(1)[228, 3342] | 15, 4946, 108<br>(0)[761, 7517] |

Table 7: *Machine independent results for the Two Layer Burgers equation*$\times 12$ *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 (Cold Starts) [Calls to CRDCMP, CRSLVE].*

| *tol* | $10^{-4}$ | $10^{-6}$ | $10^{-8}$ |
|---|---|---|---|
| $p$ | 4 | | |
| BACOL/ST | 14, 9351, 623<br>(4)[2978, 33070] | 20, 21739, 913<br>(2)[4226, 66354] | 41, 45837, 1000<br>(0)[7532, 125988] |
| BACOL/LE | 17, 8802, 791<br>(2)[3470, 33468] | 37, 18405, 969<br>(0)[4246, 57912] | 85, 43168, 990<br>(0)[4932, 111536] |
| BACOLI/ST | 17, 9374, 743<br>(3)[1973, 17567] | 25, 22267, 796<br>(0)[3538, 35635] | 49, 47682, 837<br>(1)[6560, 69823] |
| BACOLI/LE | 21, 8426, 982<br>(2)[2018, 16940] | 39, 16430, 1189<br>(0)[2455, 26875] | 103, 39509, 3274<br>(0)[6584, 68965] |
| $p$ | 5 | | |
| BACOL/ST | 15, 9389, 556<br>(8)[2580, 31890] | 16, 23893, 970<br>(4)[7034, 77314] | 28, 47294, 1071<br>(3)[9162, 133252] |
| BACOL/LE | 14, 9351, 623<br>(4)[2978, 33070] | 20, 21739, 913<br>(2)[4226, 66354] | 41, 45837, 1000<br>(0)[7532, 125988] |
| BACOLI/ST | 15, 9619, 501<br>(0)[1360, 16441] | 20, 23289, 746<br>(1)[3629, 37646] | 31, 48147, 797<br>(0)[6659, 69906] |
| BACOLI/LE | 15, 11118, 784<br>(7)[1766, 16999] | 23, 20098, 946<br>(2)[2137, 30908] | 44, 46250, 1092<br>(1)[3097, 65072] |
| $p$ | 7 | | |
| BACOL/ST | 15, 12987, 416<br>(5)[2460, 34312] | 15, 25815, 756<br>(9)[8120, 81230] | 16, 76925, 1134<br>(4)[17812, 154340] |
| BACOL/LE | 15, 10055, 470<br>(3)[2388, 33154] | 14, 24787, 902<br>(0)[8106, 81178] | 20, 47845, 1037<br>(4)[12318, 136356] |
| BACOLI/ST | 14, 14305, 443<br>(3)[1266, 17916] | 14, 26320, 727<br>(0)[4592, 42269] | 20, 51931, 920<br>(1)[8651, 79251] |
| BACOLI/LE | 13, 14566, 538<br>(6)[1525, 18499] | 15, 27592, 929<br>(8)[4104, 40606] | 21, 47563, 1048<br>(2)[5205, 69059] |
| $p$ | 9 | | |
| BACOL/ST | 15, 11588, 342<br>(0)[2544, 35870] | 15, 27669, 616<br>(3)[8576, 84756] | 15, 54314, 936<br>(2)[18652, 157012] |
| BACOL/LE | 14, 12076, 330<br>(0)[2716, 37782] | 15, 26529, 674<br>(3)[8348, 83096] | 14, 99242, 1144<br>(2)[20424, 166360] |
| BACOLI/ST | 15, 11588, 376<br>(0)[1501, 18907] | 15, 27863, 571<br>(1)[4329, 42741] | 16, 54521, 949<br>(0)[9509, 82749] |
| BACOLI/LE | 13, 12593, 418<br>(1)[1703, 20538] | 14, 31352, 773<br>(6)[4905, 44957] | 15, 74372, 1246<br>(10)[10259, 87568] |

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

| *tol* | $10^{-4}$ | $10^{-6}$ | $10^{-8}$ |
|---|---|---|---|
| *p* | 4 | | |
| BACOL/ST | 18, 284, 21<br>(3)[172, 972] | 21, 644, 58<br>(4)[360, 2438] | 66, 1490, 133<br>(3)[734, 5508] |
| BACOL/LE | 15, 238, 33<br>(4)[214, 1002] | 45, 623, 88<br>(4)[468, 2734] | 151, 1485, 173<br>(4)[874, 6022] |
| BACOLI/ST | 10, 209, 21<br>(0)[67, 428] | 21, 598, 64<br>(0)[168, 1203] | 48, 1602, 132<br>(0)[332, 2927] |
| BACOLI/LE | 15, 279, 46<br>(3)[133, 624] | 43, 681, 97<br>(4)[242, 1402] | 193, 1572, 191<br>(7)[461, 2985] |
| *p* | 5 | | |
| BACOL/ST | 11, 317, 18<br>(2)[164, 1052] | 14, 634, 41<br>(0)[268, 2258] | 30, 1402, 91<br>(0)[602, 4880] |
| BACOL/LE | 18, 284, 21<br>(3)[172, 972] | 21, 644, 58<br>(4)[360, 2438] | 66, 1490, 133<br>(3)[734, 5508] |
| BACOLI/ST | 9, 230, 19<br>(0)[67, 449] | 15, 599, 58<br>(0)[169, 1175] | 28, 1624, 107<br>(0)[307, 2879] |
| BACOLI/LE | 14, 319, 23<br>(3)[83, 511] | 26, 962, 82<br>(8)[242, 1428] | 57, 1593, 118<br>(0)[299, 2760] |
| *p* | 7 | | |
| BACOL/ST | 8, 421, 14<br>(3)[150, 1136] | 15, 702, 20<br>(0)[234, 2256] | 19, 1679, 56<br>(0)[610, 5398] |
| BACOL/LE | 12, 355, 16<br>(2)[150, 1100] | 14, 708, 31<br>(0)[224, 2342] | 18, 1604, 76<br>(1)[630, 5354] |
| BACOLI/ST | 7, 256, 18<br>(0)[67, 478] | 12, 728, 45<br>(0)[129, 1277] | 15, 1648, 93<br>(0)[325, 2866] |
| BACOLI/LE | 13, 263, 16<br>(0)[57, 463] | 13, 818, 36<br>(3)[128, 1106] | 35, 2559, 75<br>(2)[267, 2555] |
| *p* | 9 | | |
| BACOL/ST | 5, 523, 13<br>(4)[202, 1326] | 14, 913, 21<br>(0)[266, 2826] | 19, 1849, 49<br>(0)[682, 6120] |
| BACOL/LE | 9, 295, 4<br>(0)[102, 966] | 15, 955, 31<br>(2)[364, 2950] | 14, 2069, 58<br>(3)[718, 5986] |
| BACOLI/ST | 6, 333, 14<br>(0)[66, 567] | 10, 840, 32<br>(0)[155, 1430] | 13, 1975, 68<br>(0)[346, 3215] |
| BACOLI/LE | 9, 575, 15<br>(5)[78, 539] | 14, 839, 23<br>(0)[112, 1293] | 22, 1810, 48<br>(0)[288, 2850] |

Table 9: *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 (Cold Starts) [Calls to CRDCMP, CRSLVE].*

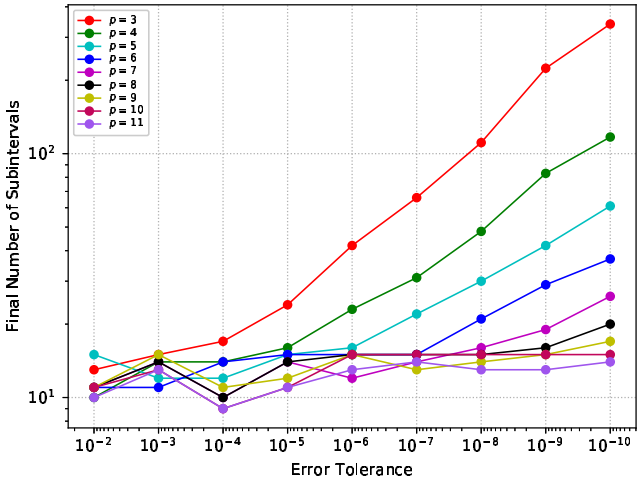


Figure 1: BACOL/ST Number of Subintervals vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 3 \ldots 10$

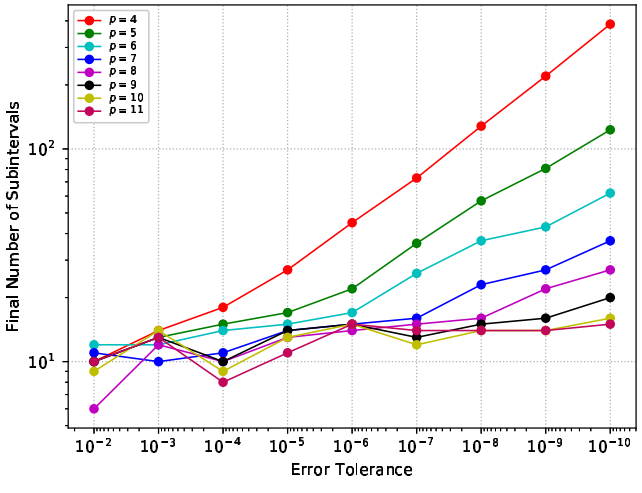


Figure 2: BACOLI/LE Number of Subintervals vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 10$

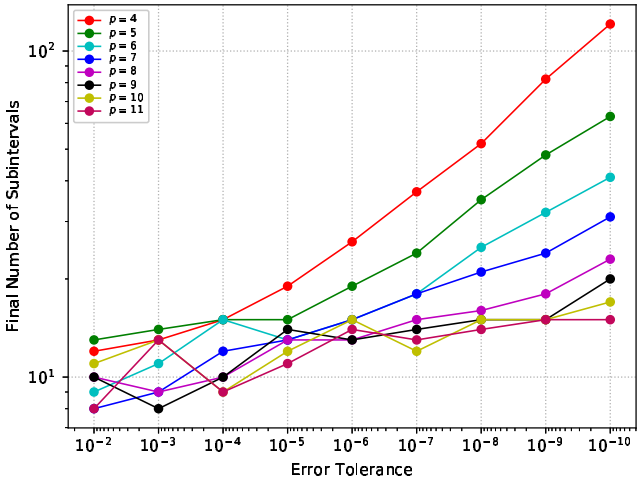


Figure 3: BACOLI/ST Number of Subintervals vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 10$

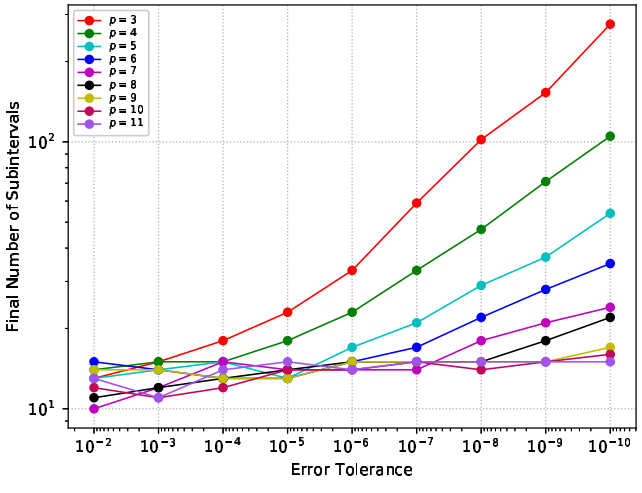


Figure 4: BACOL/ST Number of Subintervals vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 3 \ldots 10$

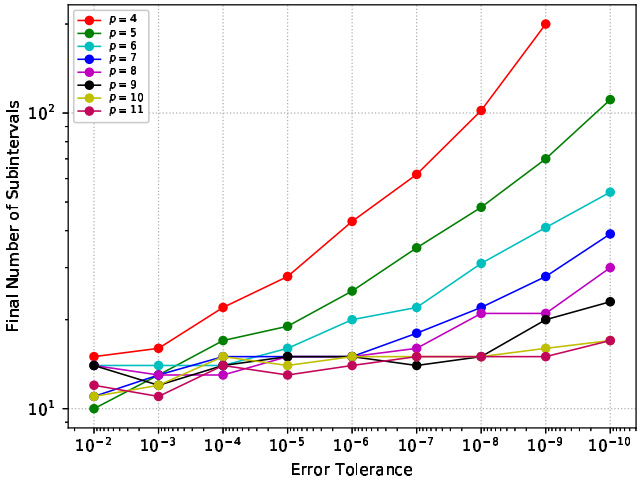


Figure 5: BACOLI/LE Number of Subintervals vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 10$

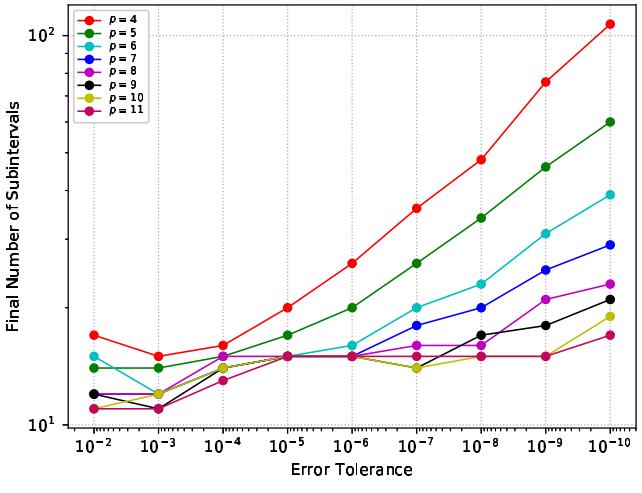


Figure 6: BACOLI/ST Number of Subintervals vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 10$

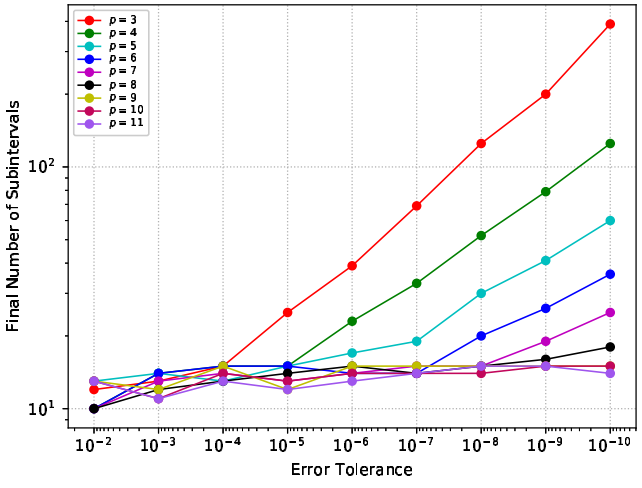


Figure 7: BACOL/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 3 \ldots 10$

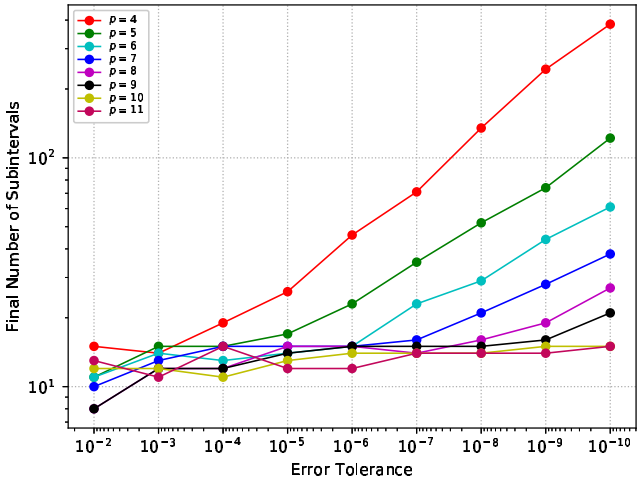


Figure 8: BACOLI/LE Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 10$

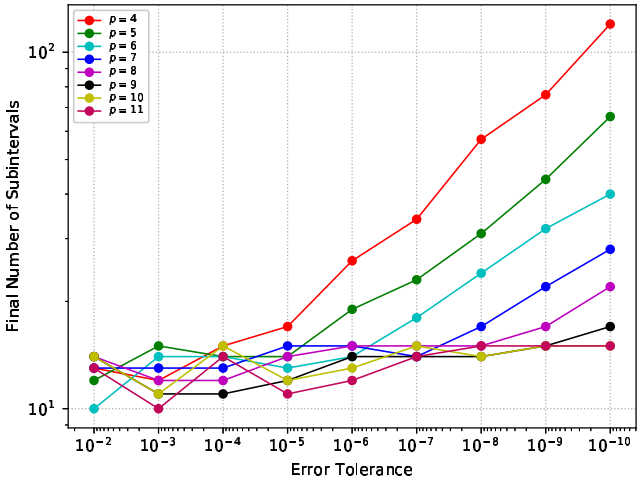


Figure 9: BACOLI/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 10$

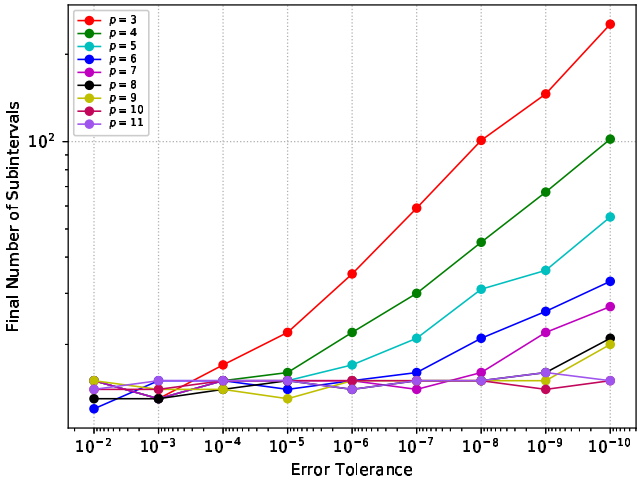


Figure 10: BACOL/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 3 \ldots 10$

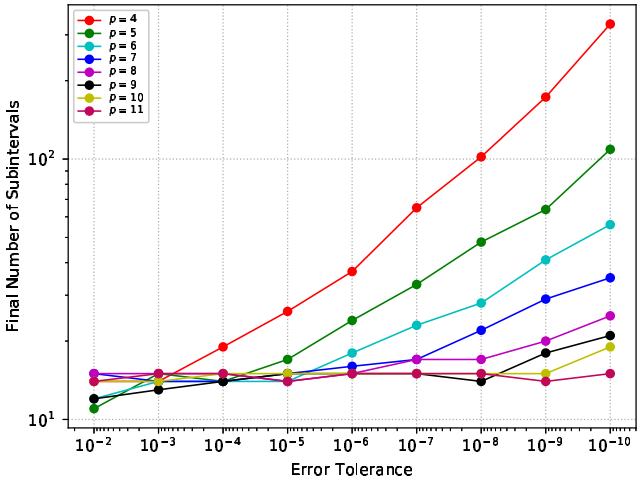


Figure 11: BACOLI/LE Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 10$

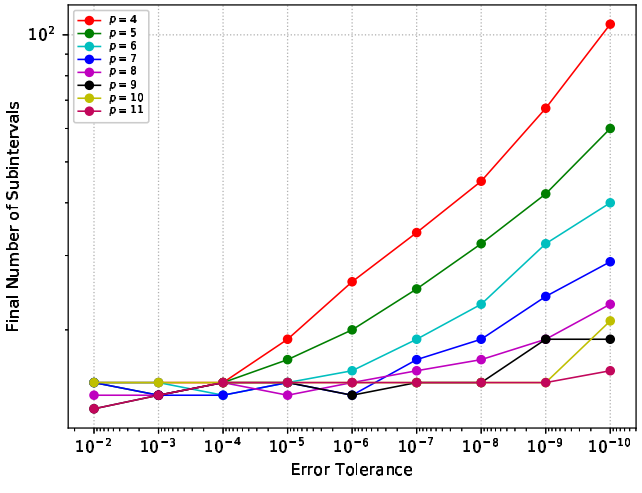


Figure 12: BACOLI/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 10$

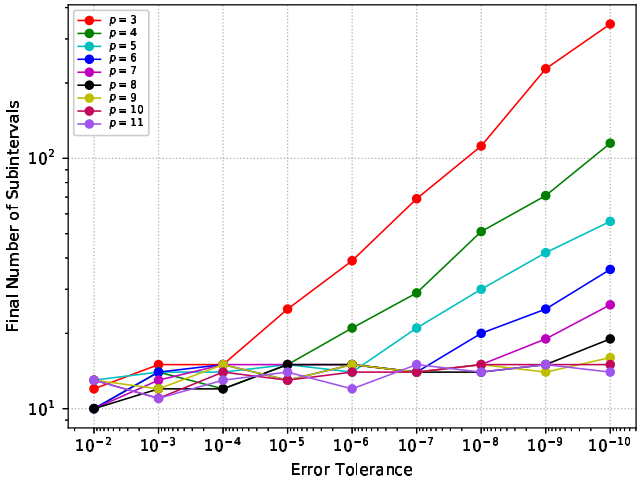


Figure 13: BACOL/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 3 \ldots 10$

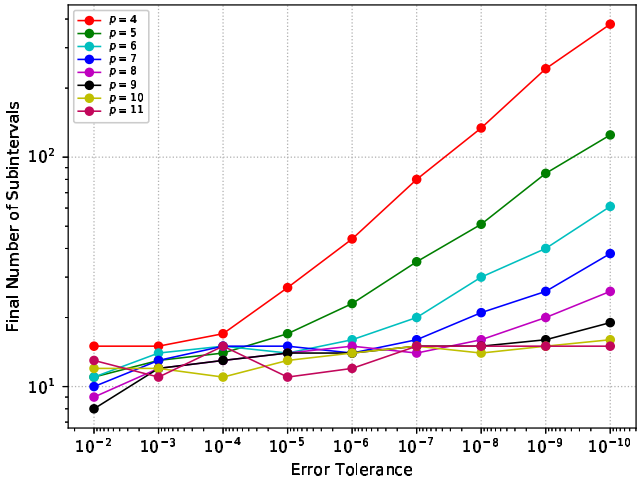


Figure 14: BACOLI/LE Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 4 \ldots 10$

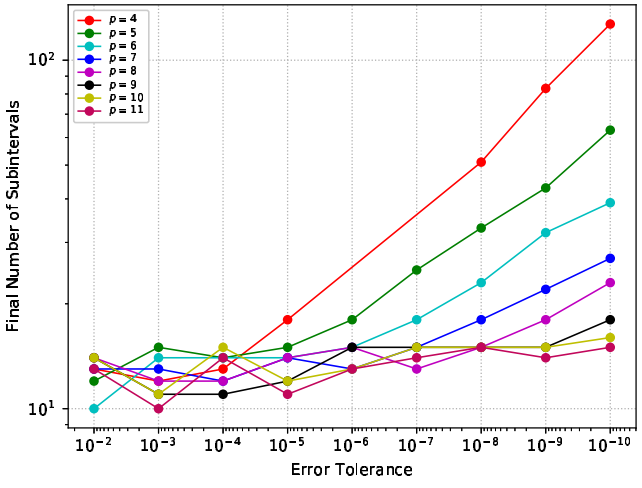


Figure 15: BACOLI/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 4 \ldots 10$

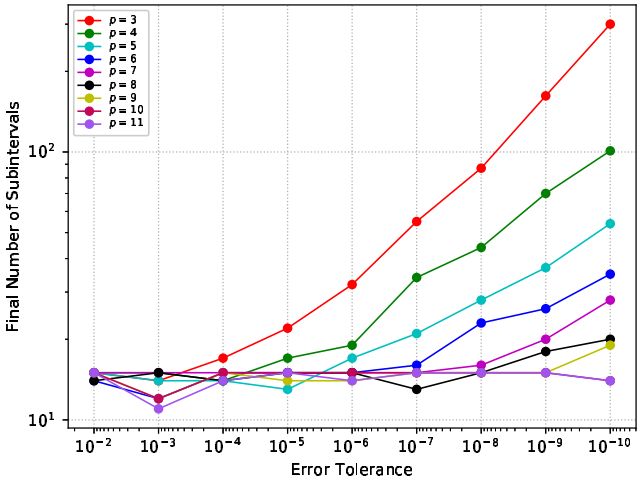


Figure 16: BACOL/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 3 \ldots 10$

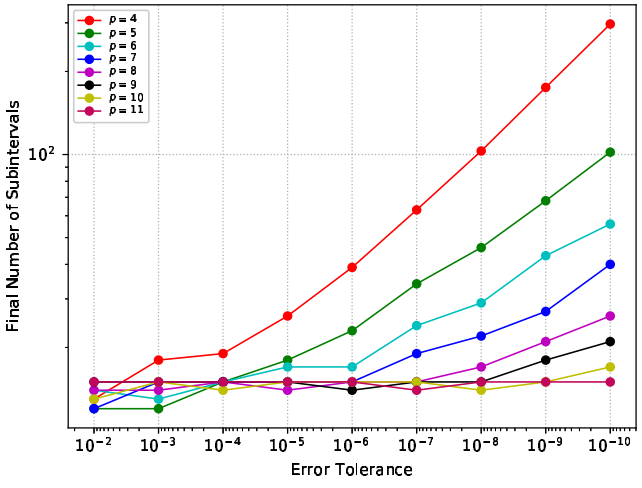


Figure 17: BACOLI/LE Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 4 \ldots 10$

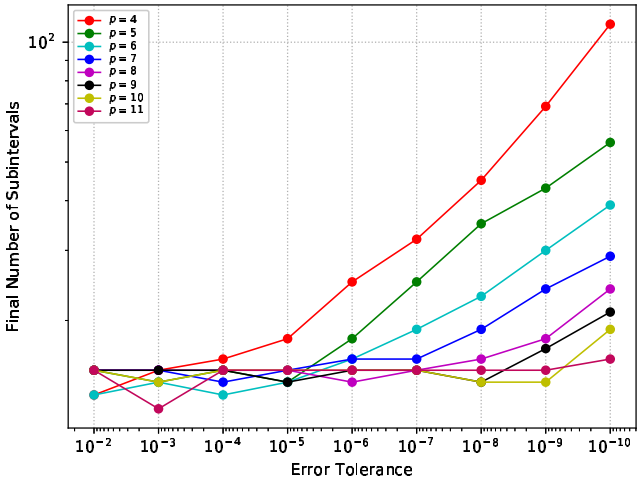


Figure 18: BACOLI/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 4 \ldots 10$

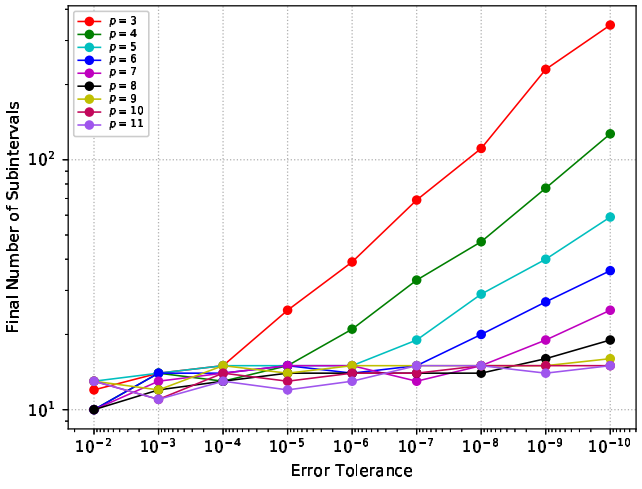


Figure 19: BACOL/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 3 \ldots 10$

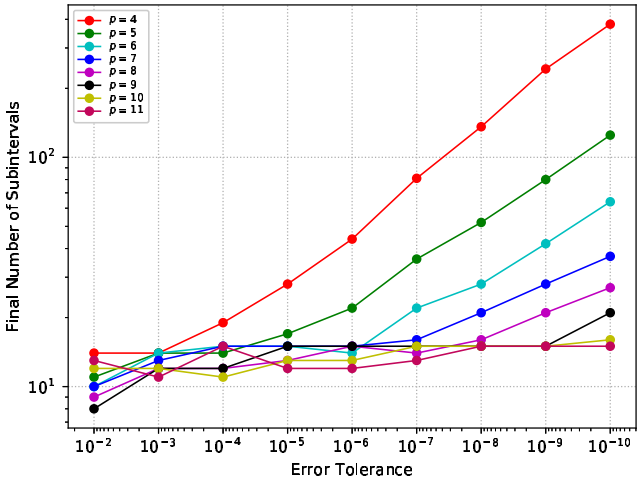


Figure 20: BACOLI/LE Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 4 \ldots 10$

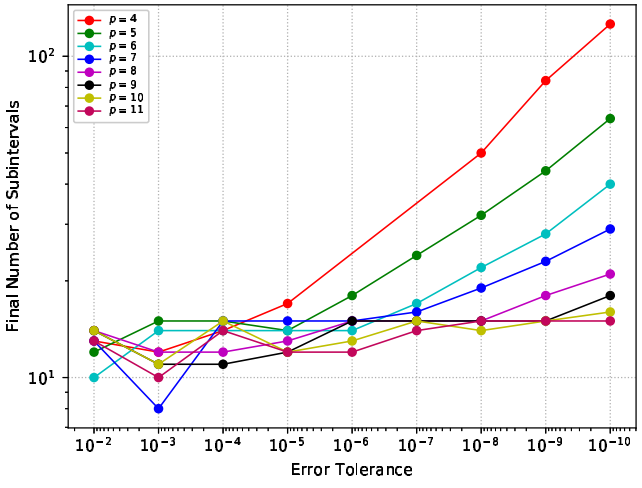


Figure 21: BACOLI/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 4 \ldots 10$

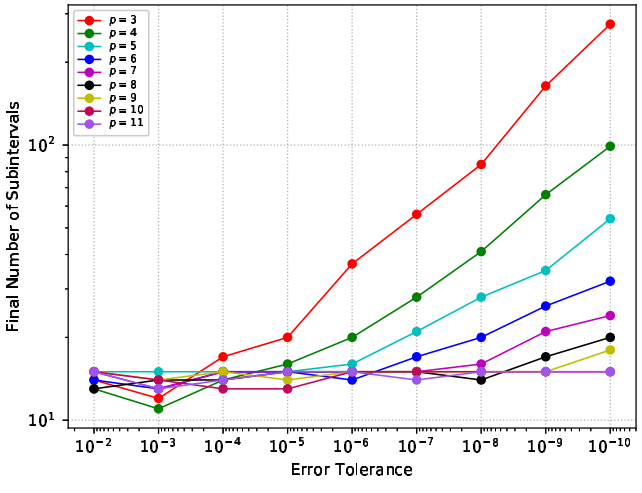


Figure 22: BACOL/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 3 \ldots 10$

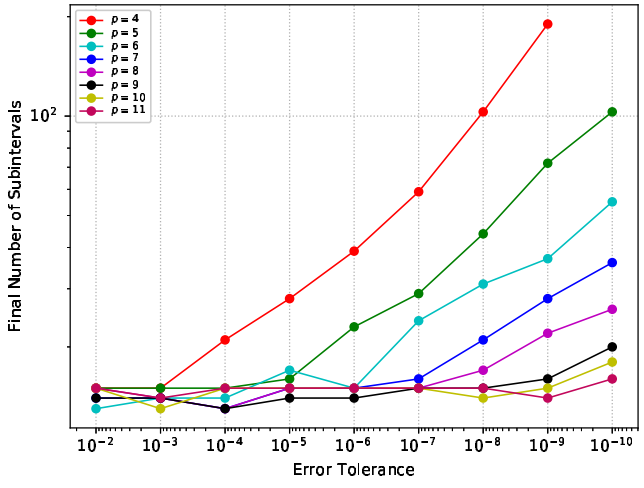


Figure 23: BACOLI/LE Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 4 \ldots 10$

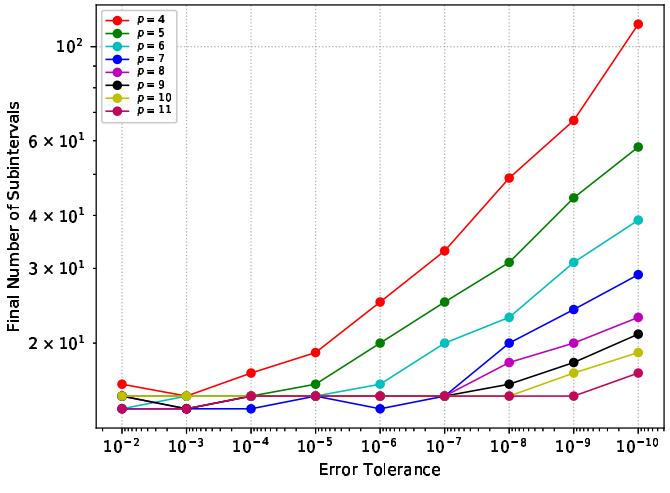


Figure 24: BACOLI/ST Number of Subintervals vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 4 \ldots 10$

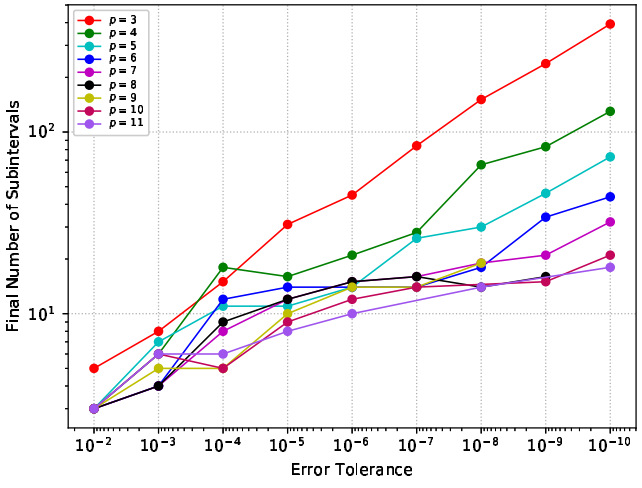


Figure 25: BACOL/ST Number of Subintervals vs. Error Tolerance: Catalytic Surface Reaction Model with $p = 3 \ldots 10$

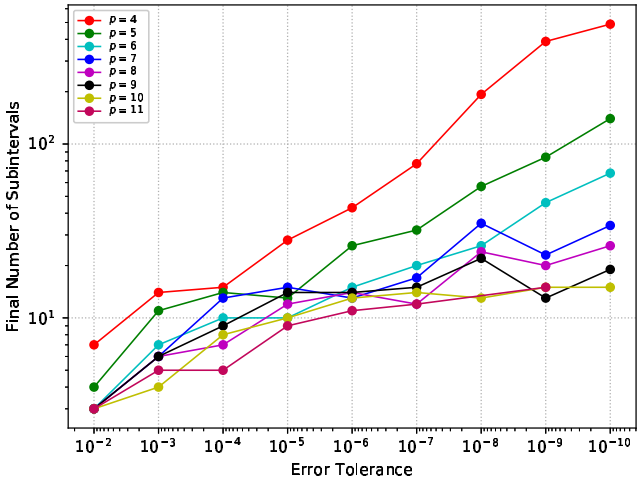


Figure 26: BACOLI/LE Number of Subintervals vs. Error Tolerance: Catalytic Surface Reaction Model with $p = 4 \ldots 10$

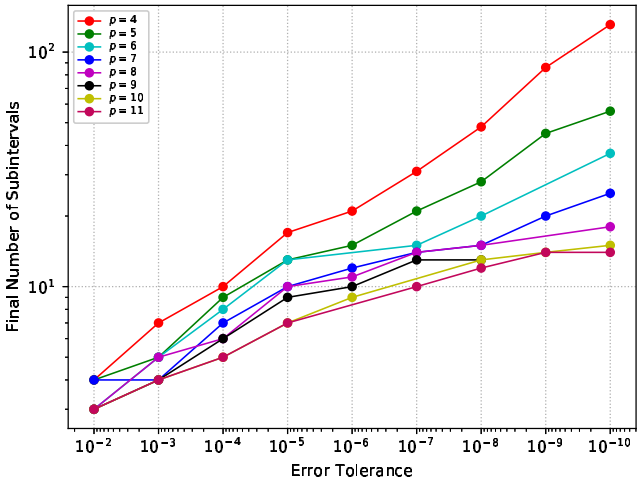


Figure 27: BACOLI/ST Number of Subintervals vs. Error Tolerance: Catalytic Surface Reaction Model with $p = 4 \ldots 10$

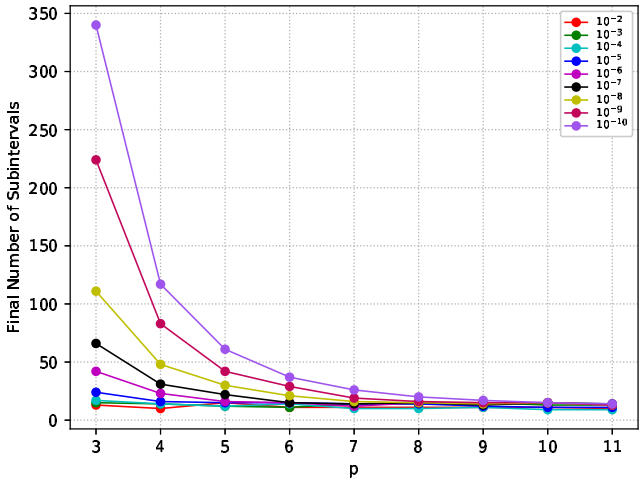


Figure 28: BACOL/ST Number of Subintervals vs $p$: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $tol = 10^{-i}, i = 2 \ldots 10$

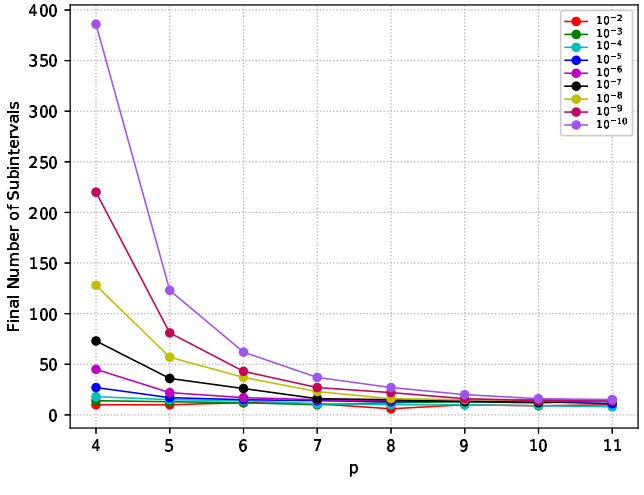


Figure 29: BACOLI/LE Number of Subintervals vs $p$: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $tol = 10^{-i}, i = 2 \ldots 10$

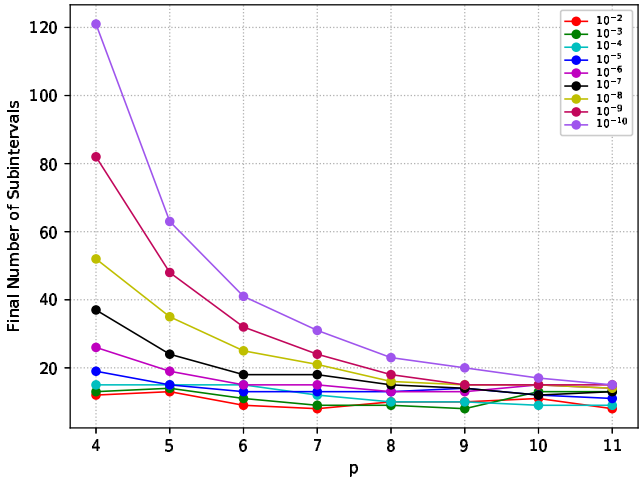


Figure 30: BACOLI/ST Number of Subintervals vs $p$: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $tol = 10^{-i}, i = 2 \ldots 10$

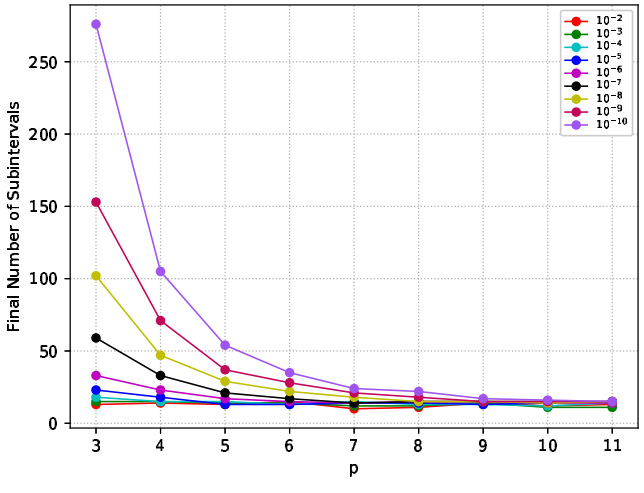


Figure 31: BACOL/ST Number of Subintervals vs $p$: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $tol = 10^{-i}, i = 2 \ldots 10$

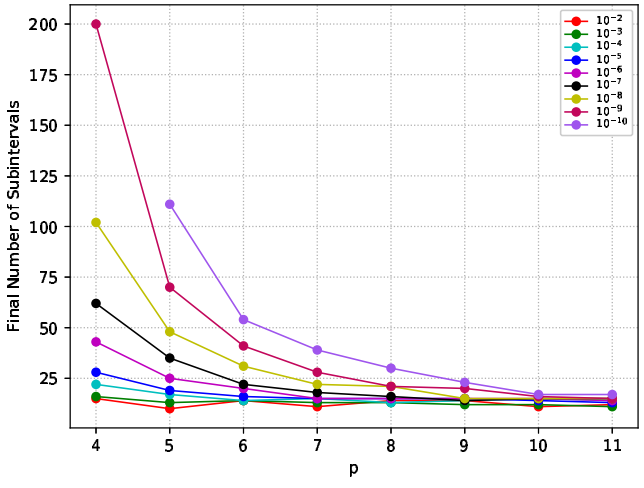


Figure 32: BACOLI/LE Number of Subintervals vs $p$: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $tol = 10^{-i}, i = 2 \ldots 10$

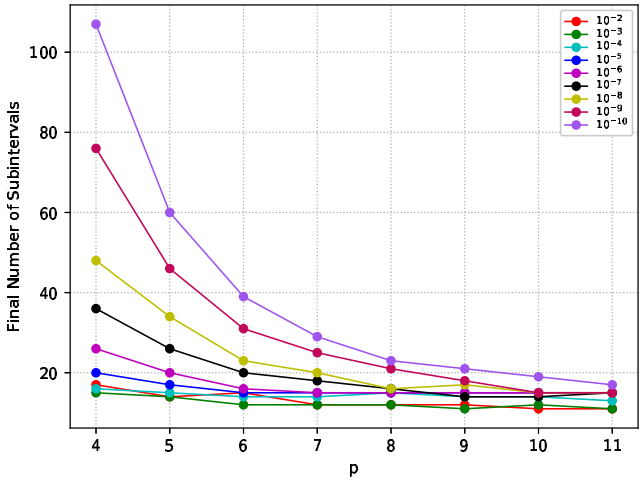


Figure 33: BACOLI/ST Number of Subintervals vs $p$: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $tol = 10^{-i}, i = 2 \ldots 10$

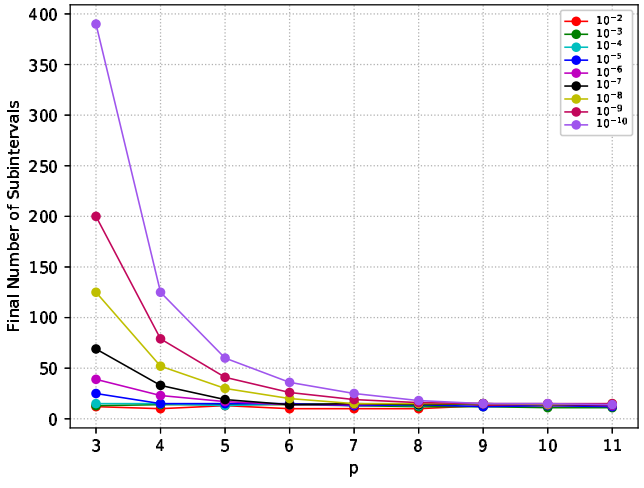


Figure 34: BACOL/ST Number of Subintervals vs $p$: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $tol = 10^{-i}, i = 2 \ldots 10$

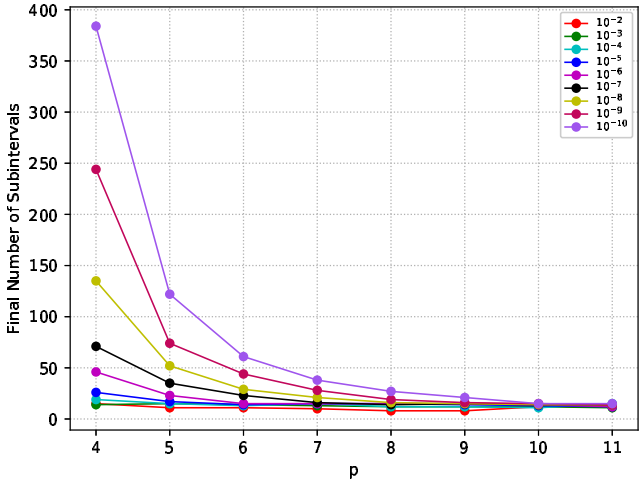


Figure 35: BACOLI/LE Number of Subintervals vs $p$: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $tol = 10^{-i}, i = 2 \ldots 10$

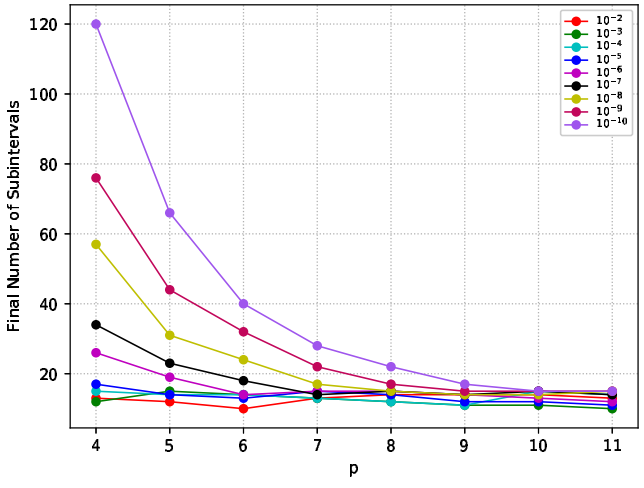


Figure 36: BACOLI/ST Number of Subintervals vs $p$: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $tol = 10^{-i}, i = 2 \ldots 10$

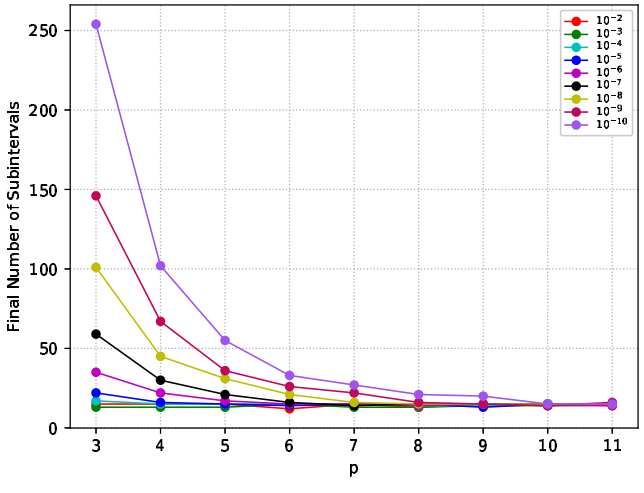


Figure 37: BACOL/ST Number of Subintervals vs $p$: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $tol = 10^{-i}, i = 2 \ldots 10$

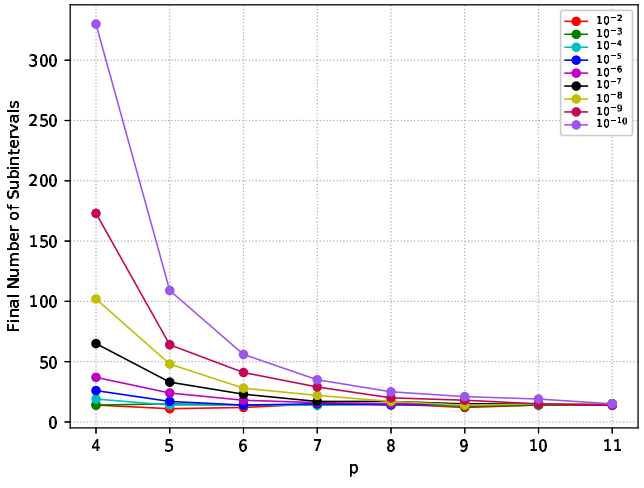


Figure 38: BACOLI/LE Number of Subintervals vs $p$: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $tol = 10^{-i}, i = 2 \ldots 10$

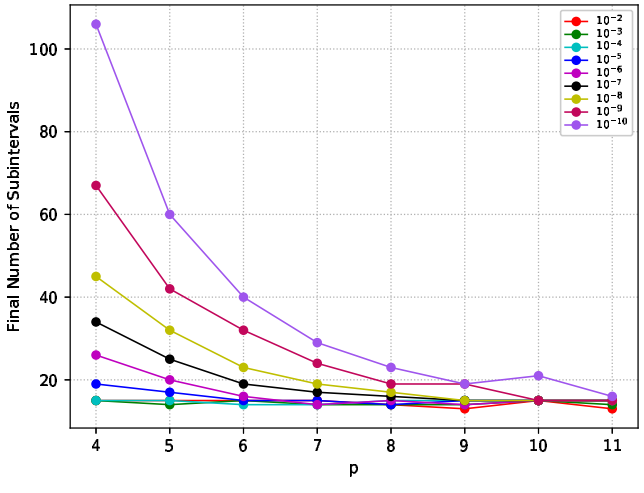


Figure 39: BACOLI/ST Number of Subintervals vs $p$: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $tol = 10^{-i}, i = 2 \ldots 10$

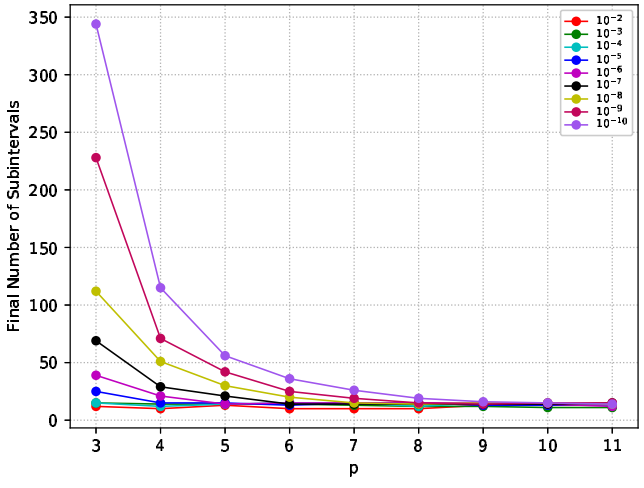


Figure 40: BACOL/ST Number of Subintervals vs $p$: Two Layer Burgers Equation$\times 6$, $\epsilon = 10^{-3}$ with $tol = 10^{-i}, i = 2 \ldots 10$

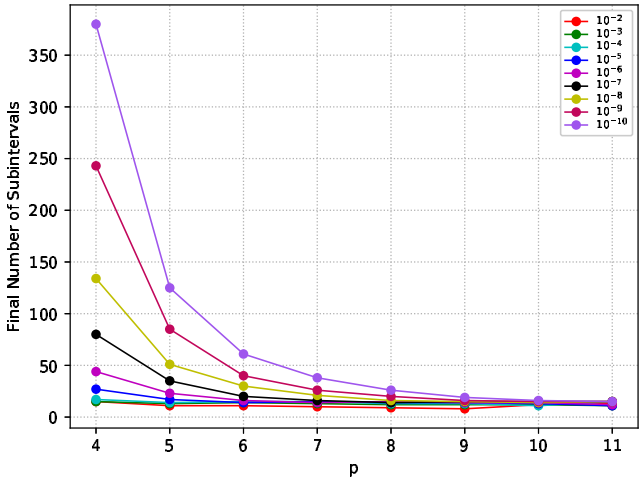


Figure 41: BACOLI/LE Number of Subintervals vs $p$: Two Layer Burgers Equation$\times 6$, $\epsilon = 10^{-3}$ with $tol = 10^{-i}, i = 2 \ldots 10$

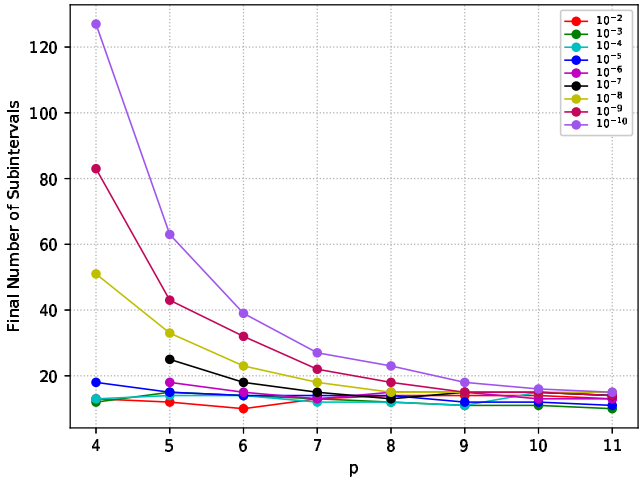


Figure 42: BACOLI/ST Number of Subintervals vs $p$: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $tol = 10^{-i}, i = 2 \ldots 10$

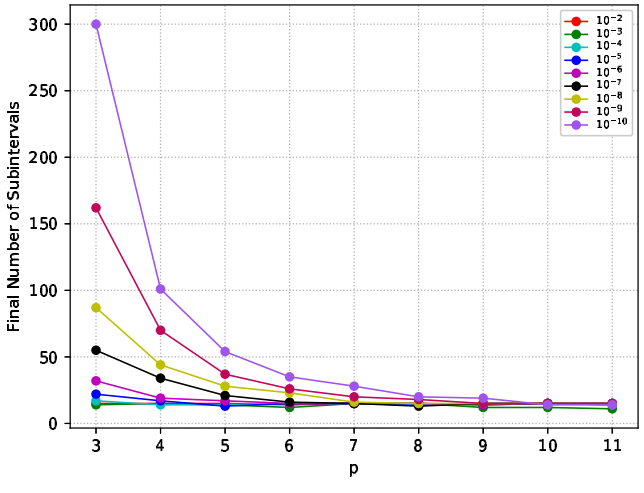


Figure 43: BACOL/ST Number of Subintervals vs $p$: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $tol = 10^{-i}, i = 2 \ldots 10$

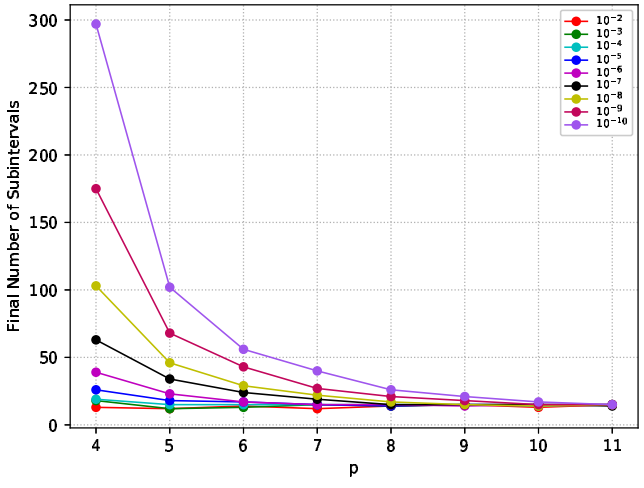


Figure 44: BACOLI/LE Number of Subintervals vs $p$: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $tol = 10^{-i}, i = 2 \ldots 10$

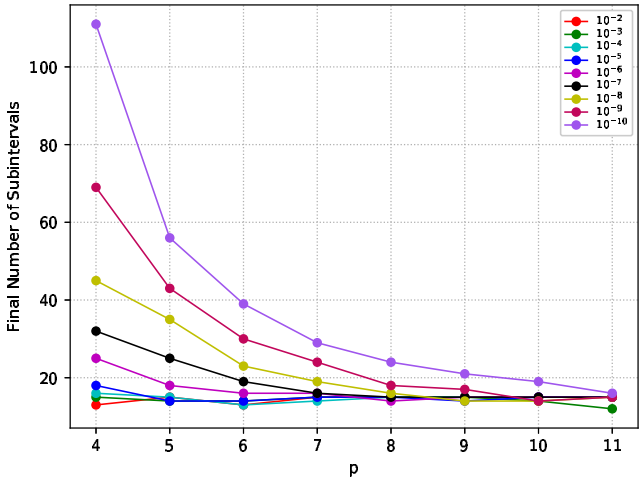


Figure 45: BACOLI/ST Number of Subintervals vs $p$: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $tol = 10^{-i}, i = 2 \ldots 10$

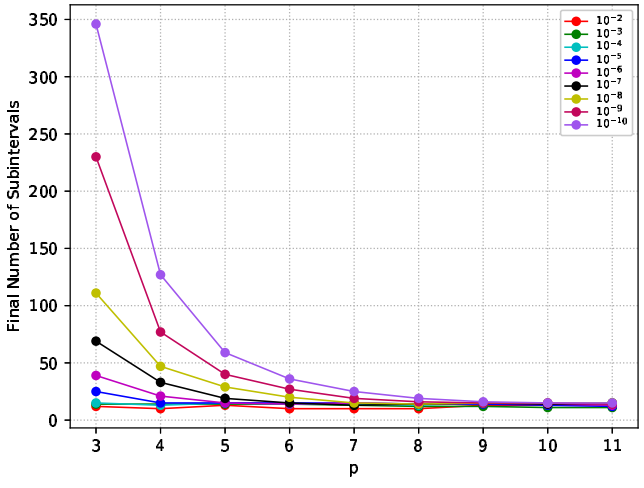


Figure 46: BACOL/ST Number of Subintervals vs $p$: Two Layer Burgers Equation$\times 12$, $\epsilon = 10^{-3}$ with $tol = 10^{-i}, i = 2 \ldots 10$

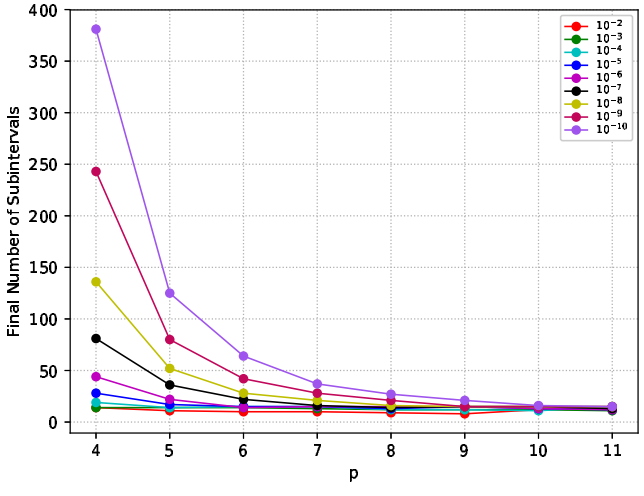


Figure 47: BACOLI/LE Number of Subintervals vs $p$: Two Layer Burgers Equation$\times 12$, $\epsilon = 10^{-3}$ with $tol = 10^{-i}, i = 2 \ldots 10$

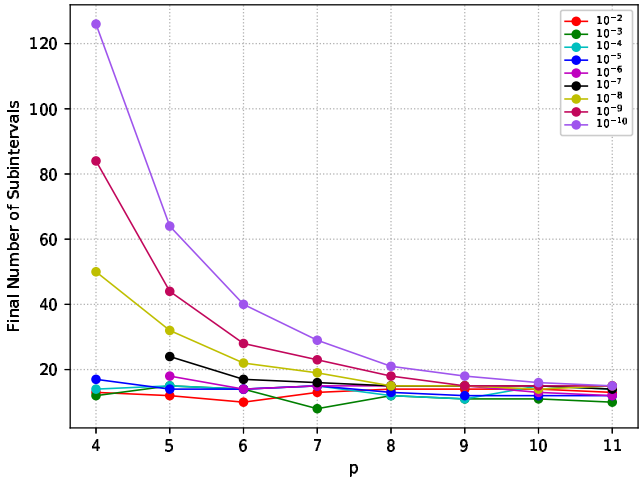


Figure 48: BACOLI/ST Number of Subintervals vs $p$: Two Layer Burgers Equation$\times$12, $\epsilon = 10^{-3}$ with $tol = 10^{-i}, i = 2 \ldots 10$

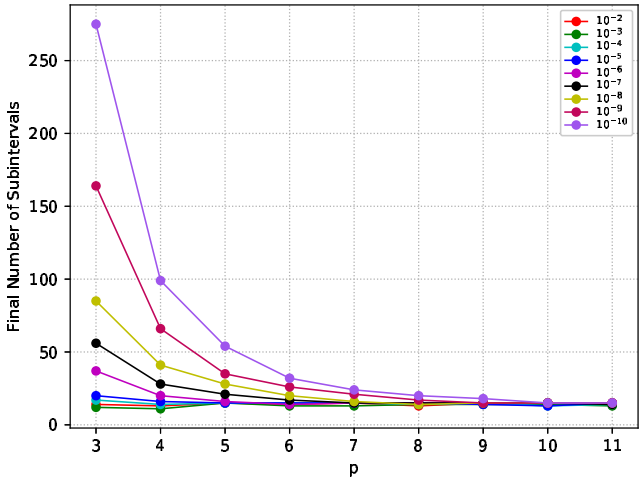


Figure 49: BACOL/ST Number of Subintervals vs $p$: Two Layer Burgers Equation$\times$12, $\epsilon = 10^{-4}$ with $tol = 10^{-i}, i = 2 \ldots 10$

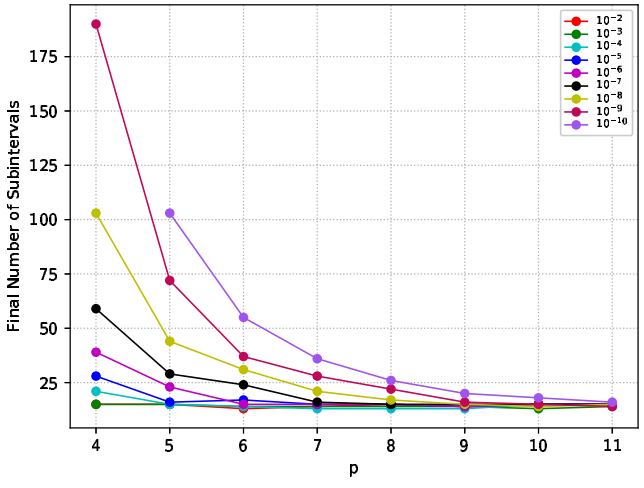


Figure 50: BACOLI/LE Number of Subintervals vs $p$: Two Layer Burgers Equation$\times 12$, $\epsilon = 10^{-4}$ with $tol = 10^{-i}, i = 2 \ldots 10$

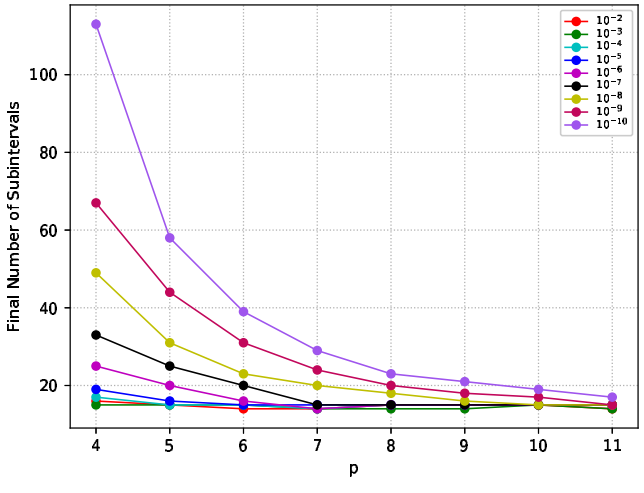


Figure 51: BACOLI/ST Number of Subintervals vs $p$: Two Layer Burgers Equation$\times 12$, $\epsilon = 10^{-4}$ with $tol = 10^{-i}, i = 2 \ldots 10$

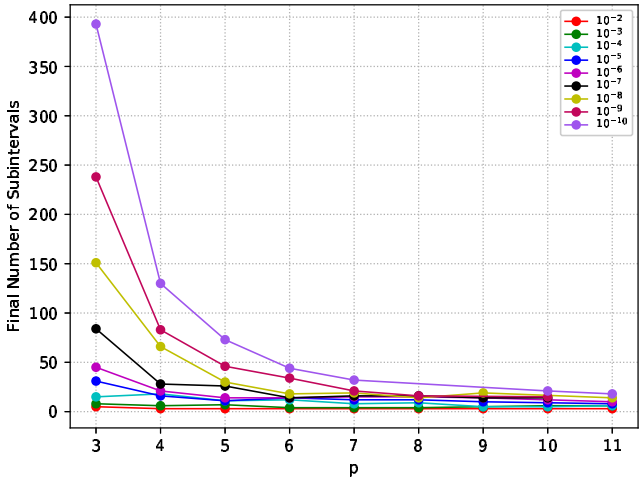


Figure 52: BACOL/ST Number of Subintervals vs $p$: Catalytic Surface Reaction Model with $tol = 10^{-i}, i = 2 \ldots 10$

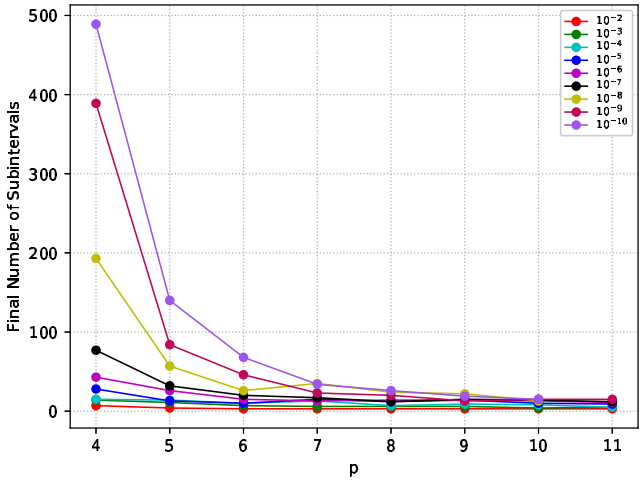


Figure 53: BACOLI/LE Number of Subintervals vs $p$: Catalytic Surface Reaction Model with $tol = 10^{-i}, i = 2 \ldots 10$

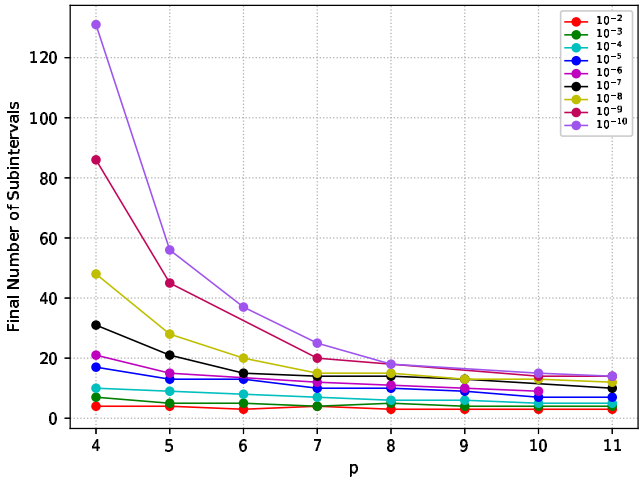


Figure 54: BACOLI/ST Number of Subintervals vs $p$: Catalytic Surface Reaction Model with $tol = 10^{-i}, i = 2 \ldots 10$

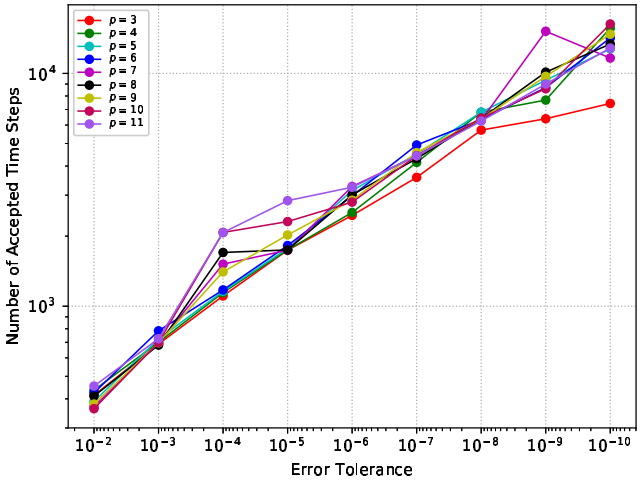


Figure 55: BACOL/ST Number of Accepted Time Steps vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 3 \ldots 11$

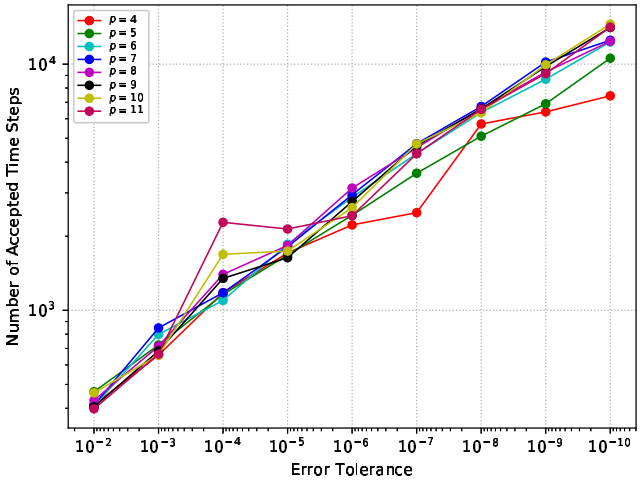


Figure 56: BACOLI/LE Number of Accepted Time Steps vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

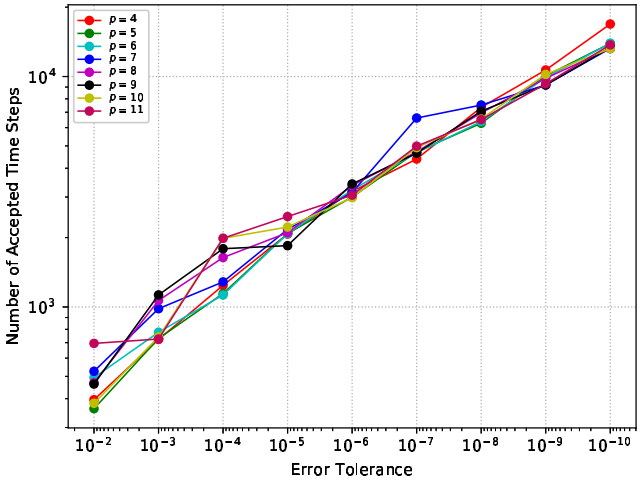


Figure 57: BACOLI/ST Number of Accepted Time Steps vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

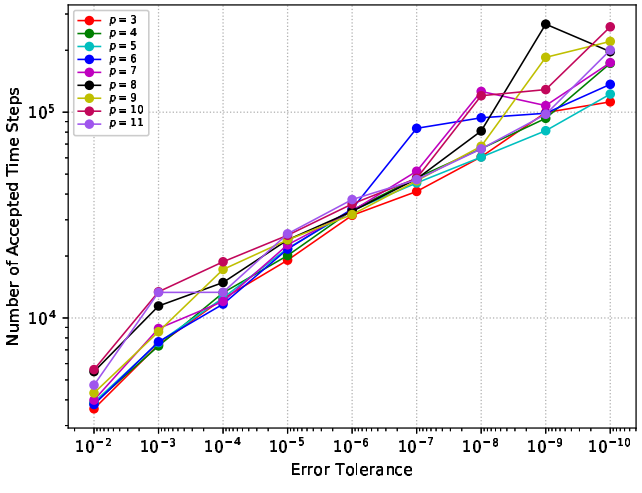


Figure 58: BACOL/ST Number of Accepted Time Steps vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 3 \ldots 11$

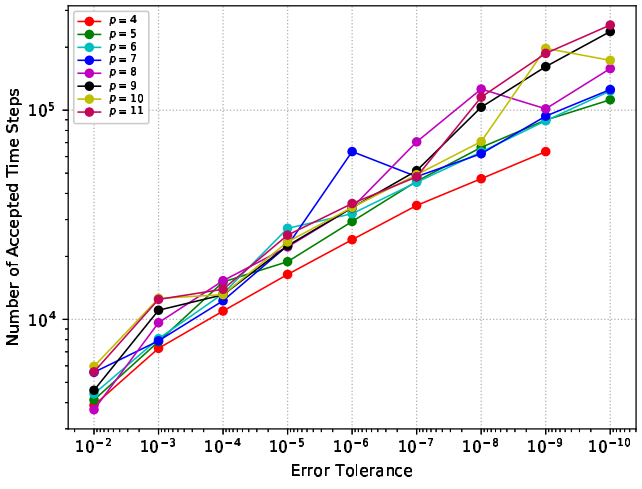


Figure 59: BACOLI/LE Number of Accepted Time Steps vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

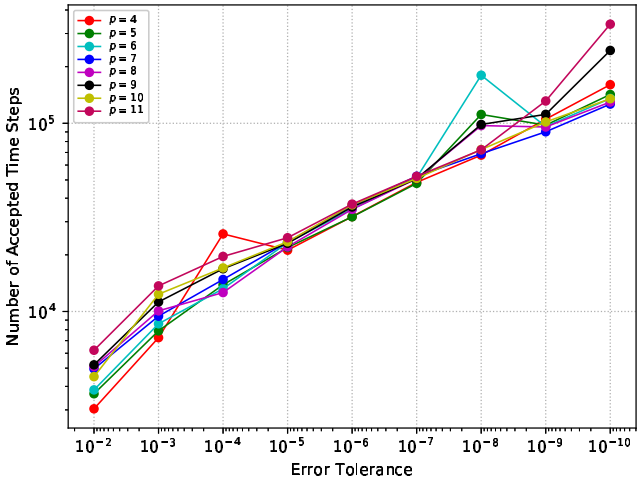


Figure 60: BACOLI/ST Number of Accepted Time Steps vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

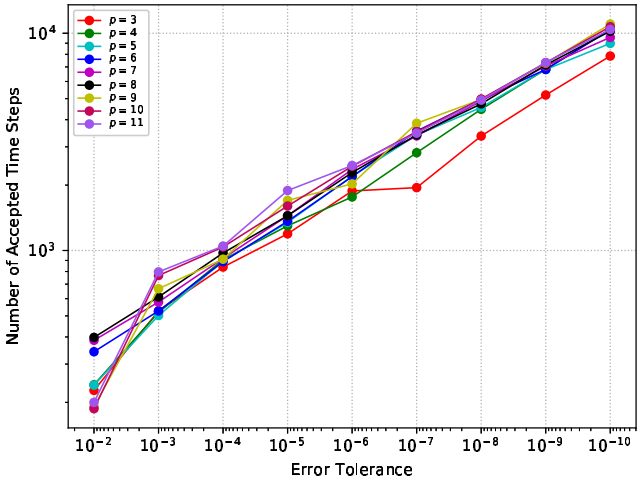


Figure 61: BACOL/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 3 \ldots 11$

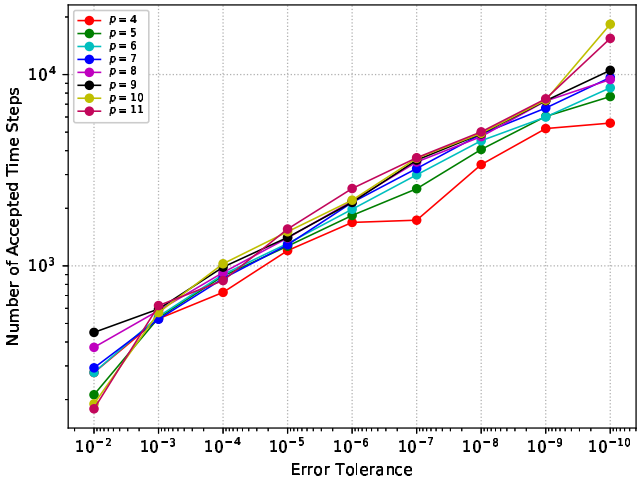


Figure 62: BACOLI/LE Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

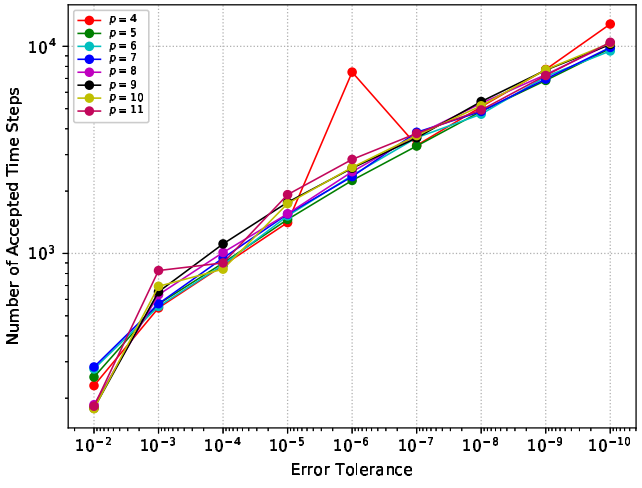


Figure 63: BACOLI/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

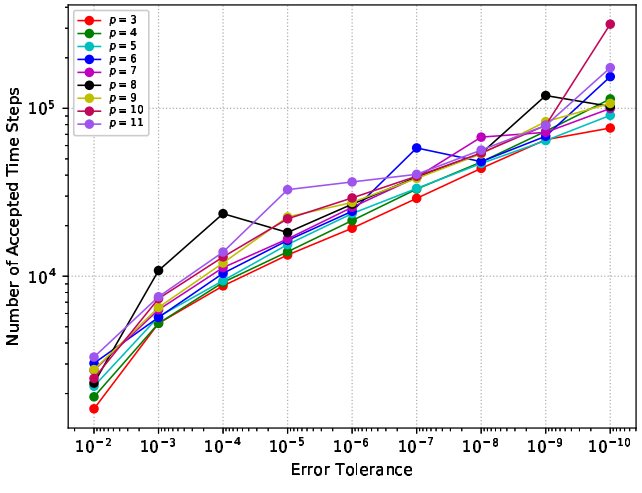


Figure 64: BACOL/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 3 \ldots 11$

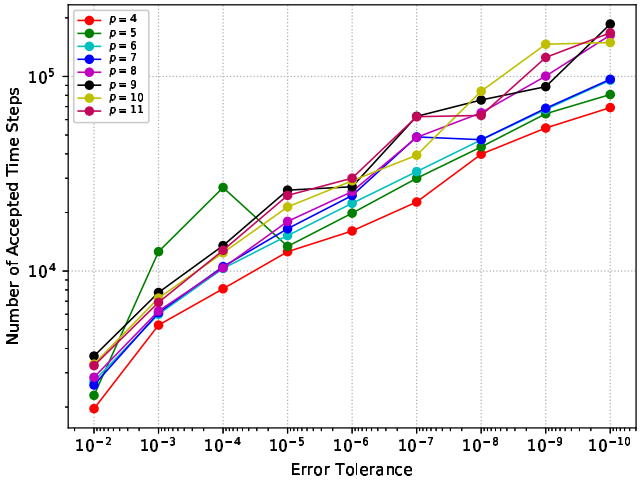


Figure 65: BACOLI/LE Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

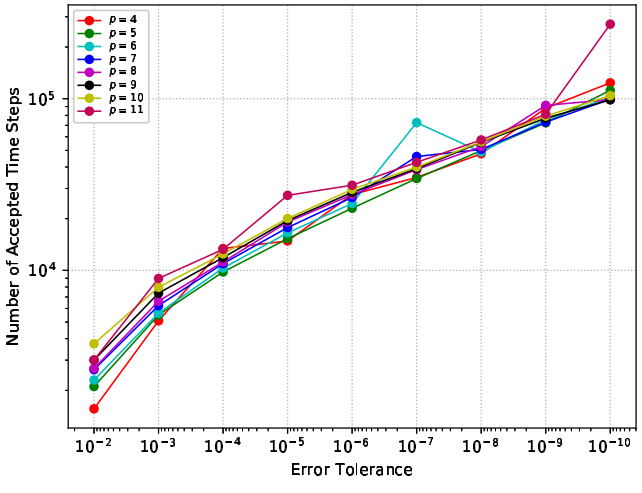


Figure 66: BACOLI/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

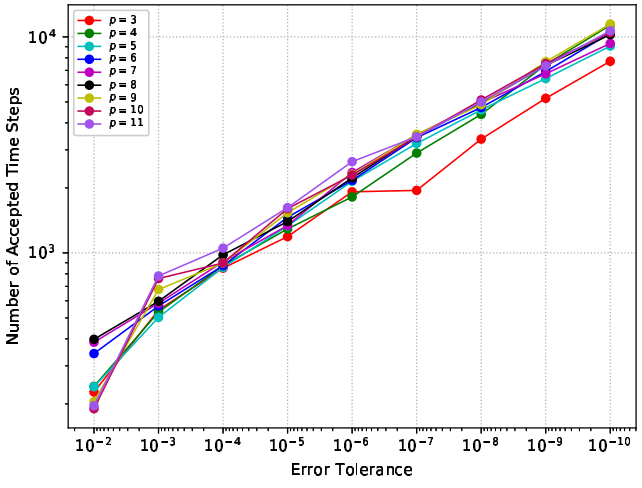


Figure 67: BACOL/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 3 \ldots 11$

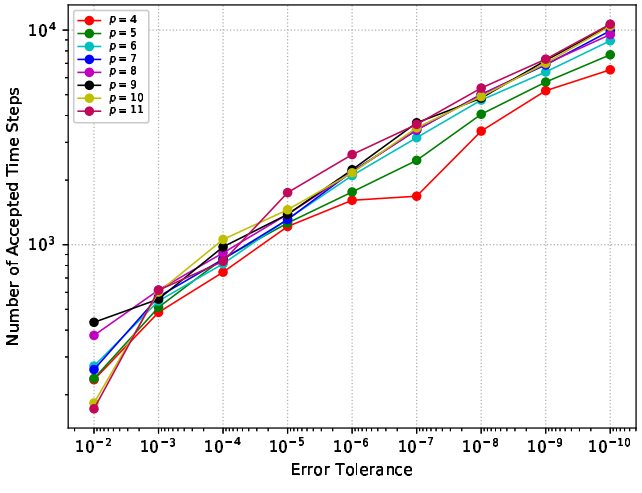


Figure 68: BACOLI/LE Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

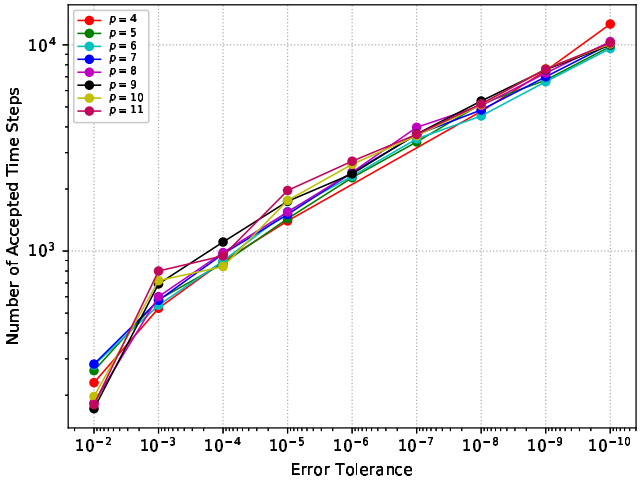


Figure 69: BACOLI/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

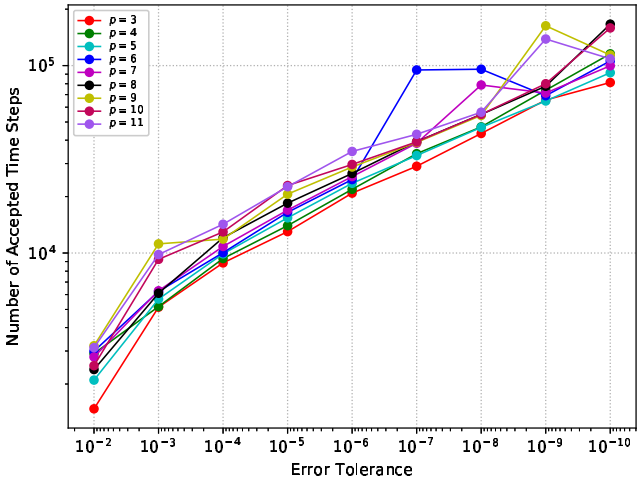


Figure 70: BACOL/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 3 \ldots 11$

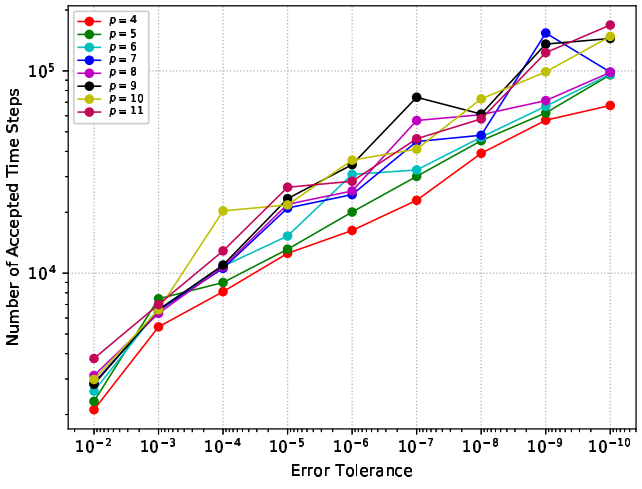


Figure 71: BACOLI/LE Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

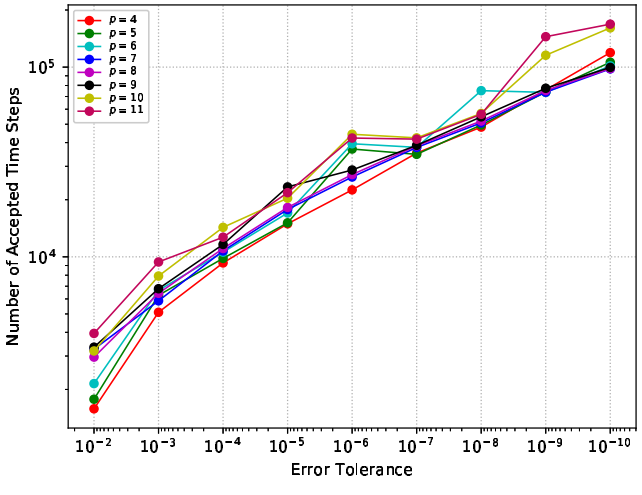


Figure 72: BACOLI/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

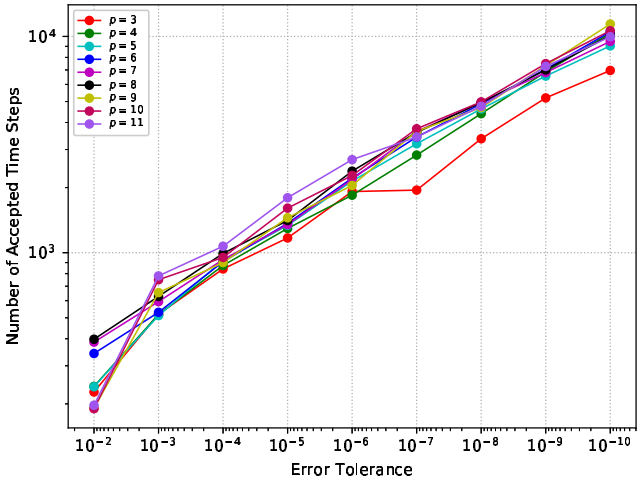


Figure 73: BACOL/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 3 \ldots 11$

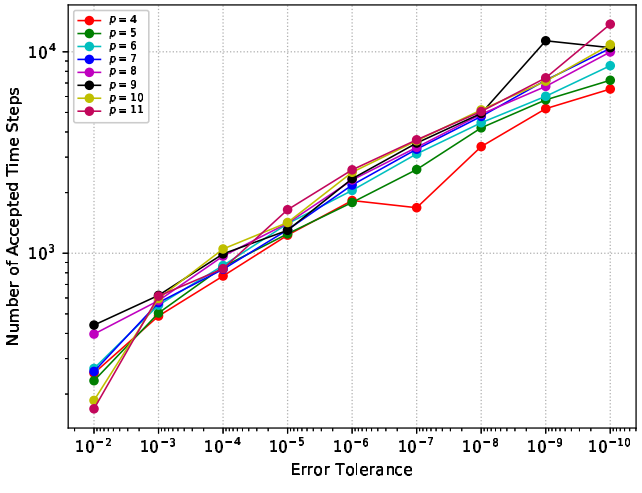


Figure 74: BACOLI/LE Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

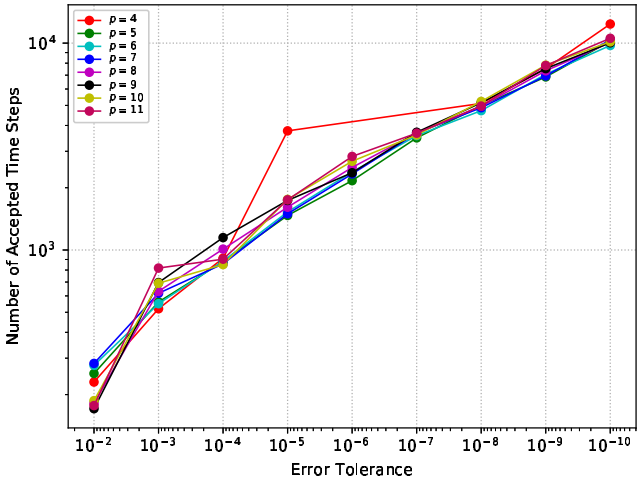


Figure 75: BACOLI/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

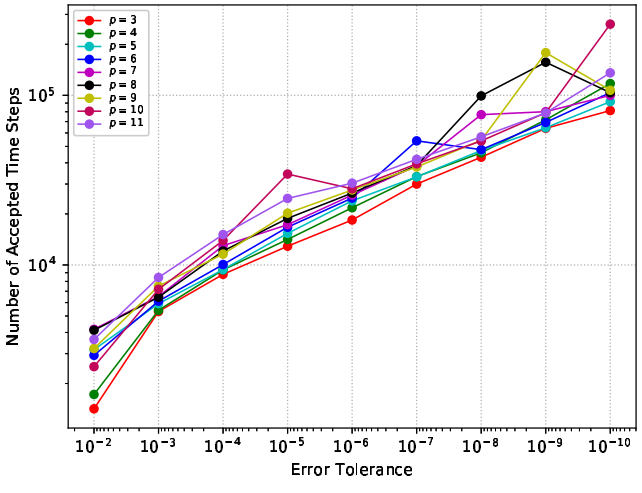


Figure 76: BACOL/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 3 \ldots 11$

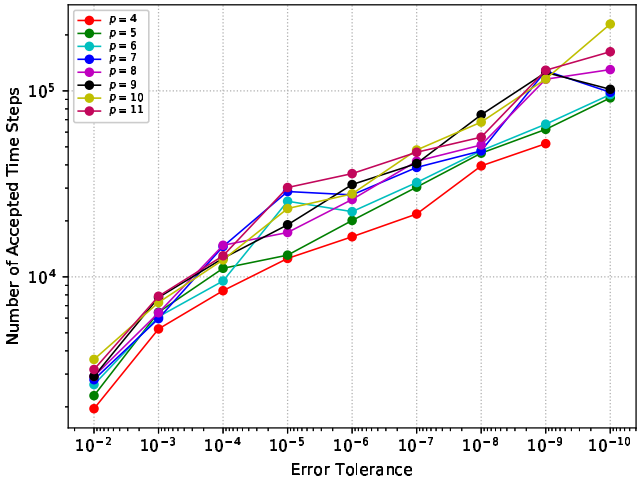


Figure 77: BACOLI/LE Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

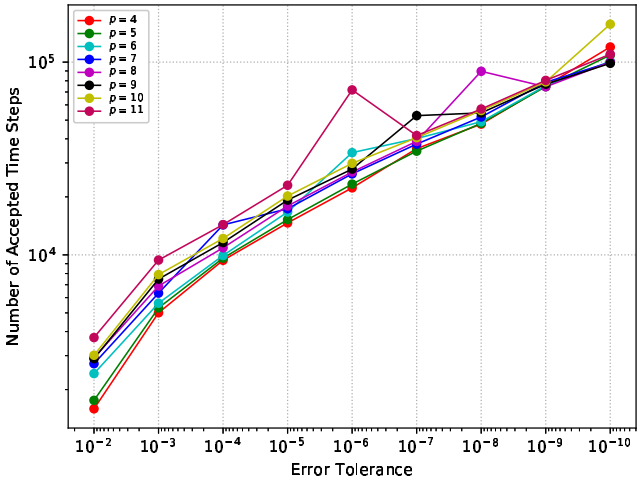


Figure 78: BACOLI/ST Number of Accepted Time Steps vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

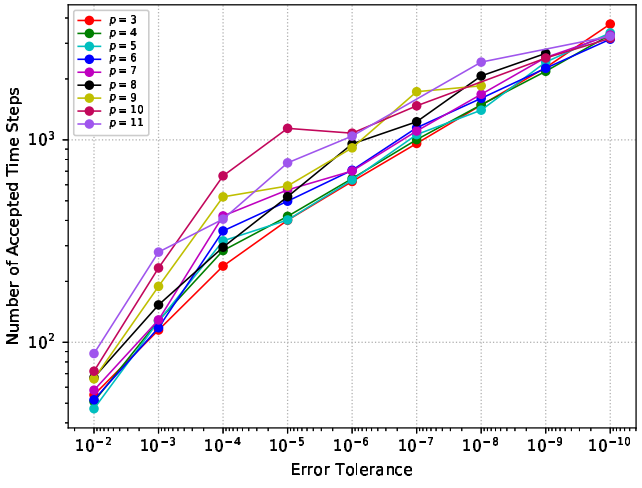


Figure 79: BACOL/ST Number of Accepted Time Steps vs. Error Tolerance: Catalytic Surface Reaction Model with $p = 3 \ldots 11$

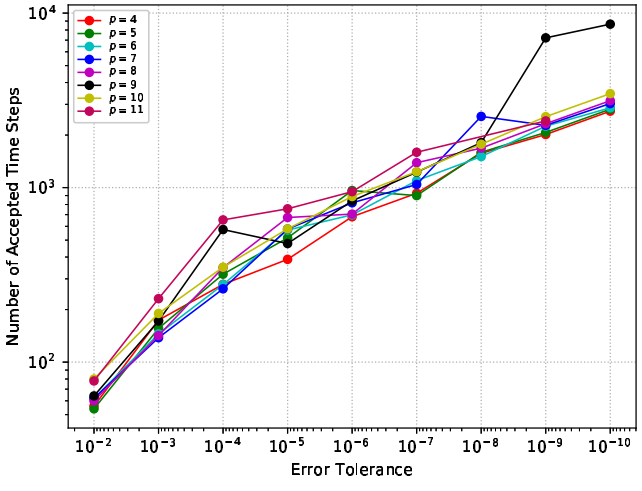


Figure 80: BACOLI/LE Number of Accepted Time Steps vs. Error Tolerance: Catalytic Surface Reaction Model with $p = 4 \ldots 11$

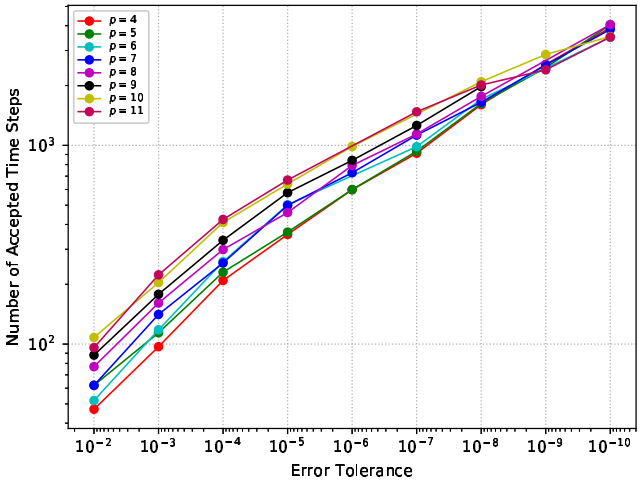


Figure 81: BACOLI/ST Number of Accepted Time Steps vs. Error Tolerance: Catalytic Surface Reaction Model with $p = 4 \ldots 11$

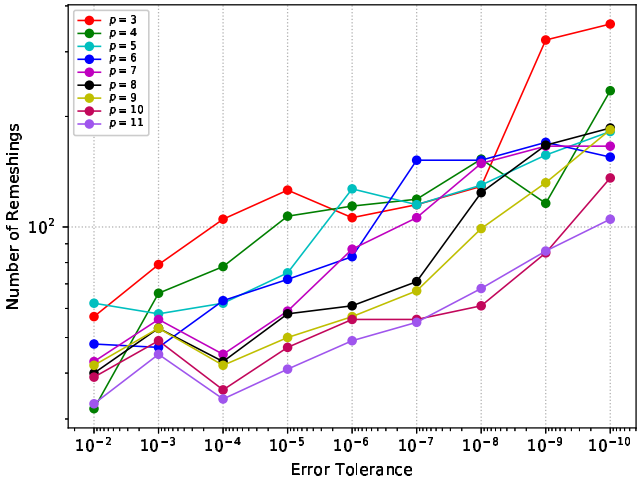


Figure 82: BACOL/ST Number of Remeshings vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 3 \ldots 11$

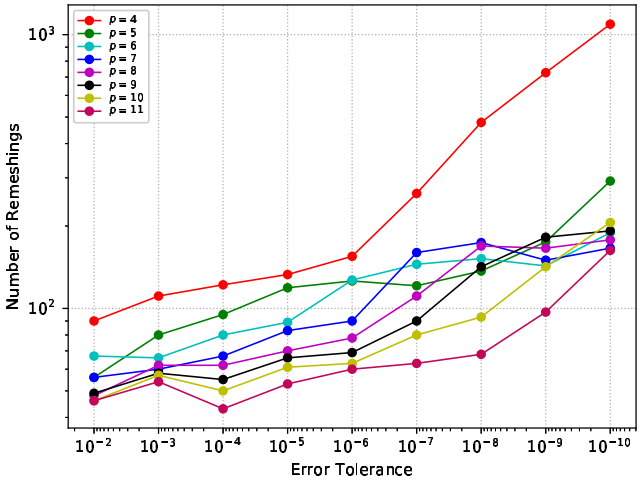


Figure 83: BACOLI/LE Number of Remeshings vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

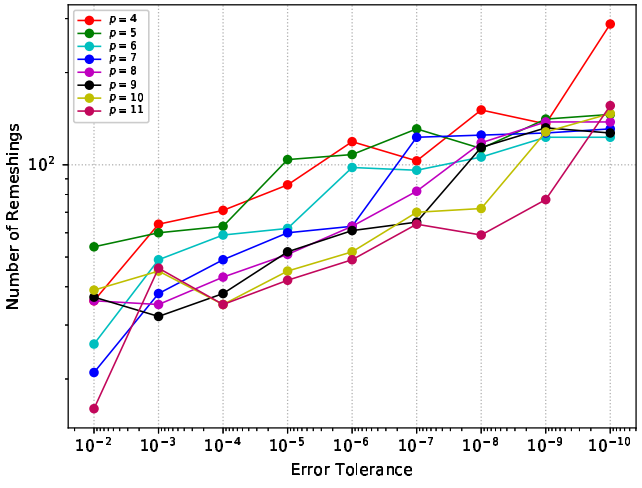


Figure 84: BACOLI/ST Number of Remeshings vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

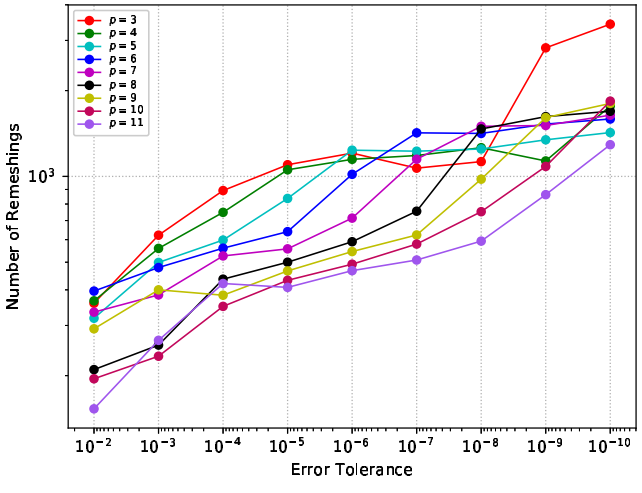


Figure 85: BACOL/ST Number of Remeshings vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 3 \ldots 11$

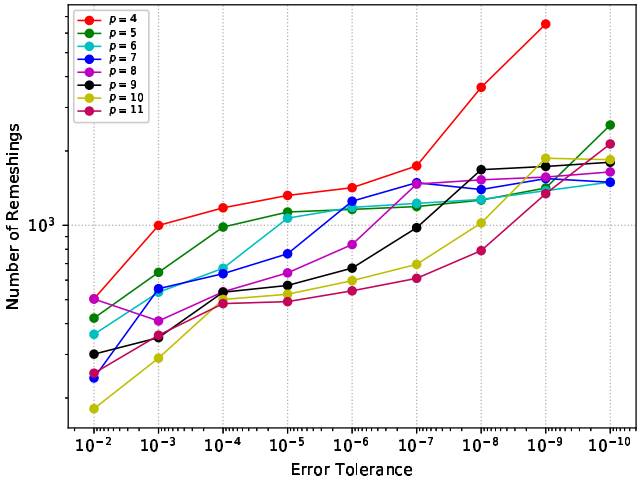


Figure 86: BACOLI/LE Number of Remeshings vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

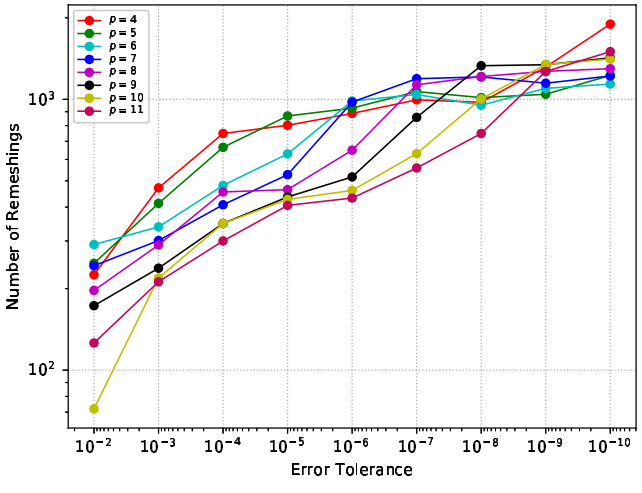


Figure 87: BACOLI/ST Number of Remeshings vs. Error Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

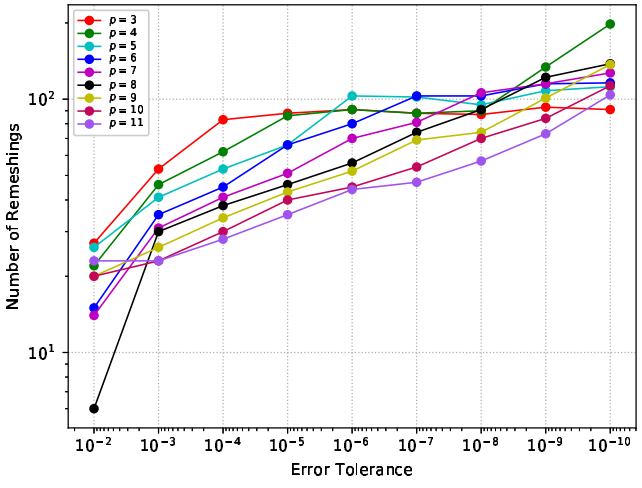


Figure 88: BACOL/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 3 \ldots 11$

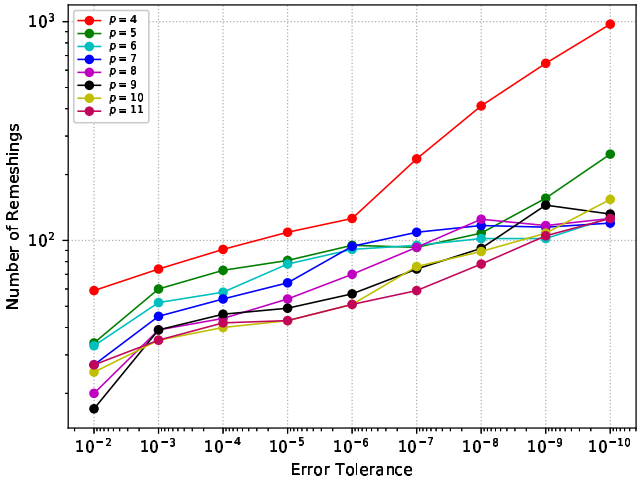


Figure 89: BACOLI/LE Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

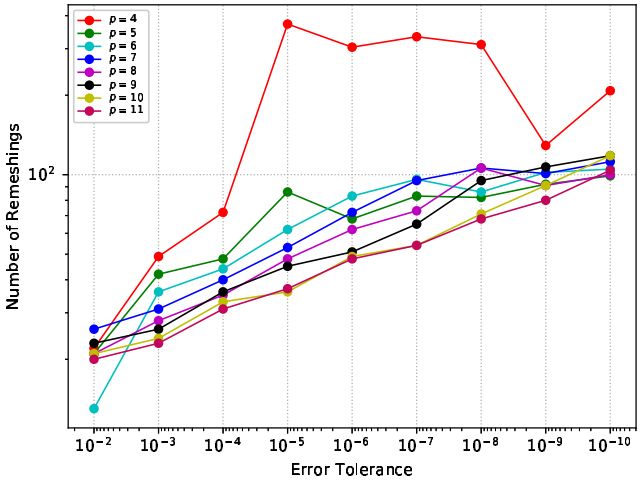


Figure 90: BACOLI/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

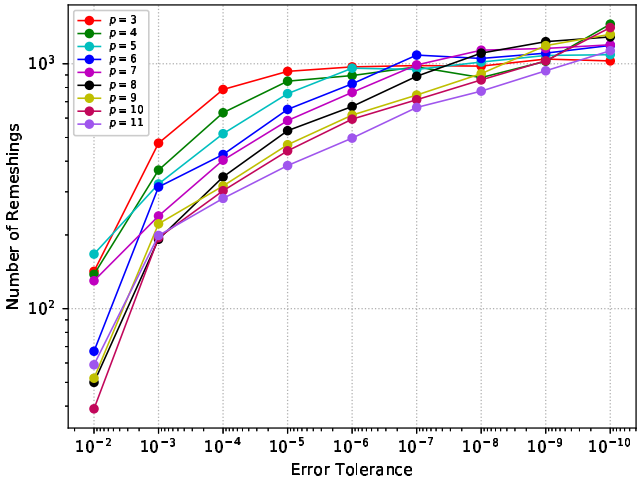


Figure 91: BACOL/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 3 \ldots 11$

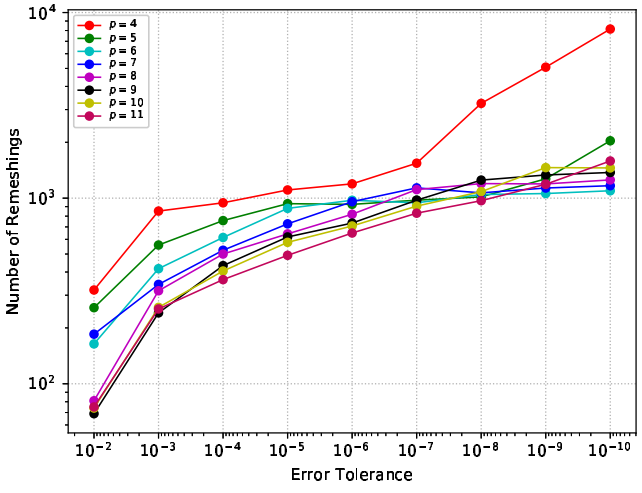


Figure 92: BACOLI/LE Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

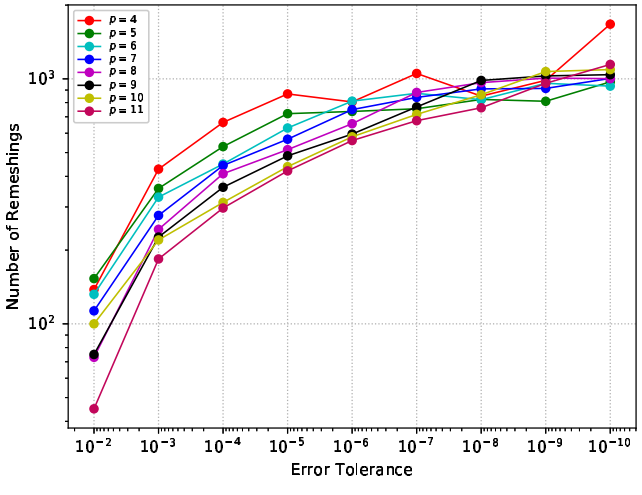


Figure 93: BACOLI/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

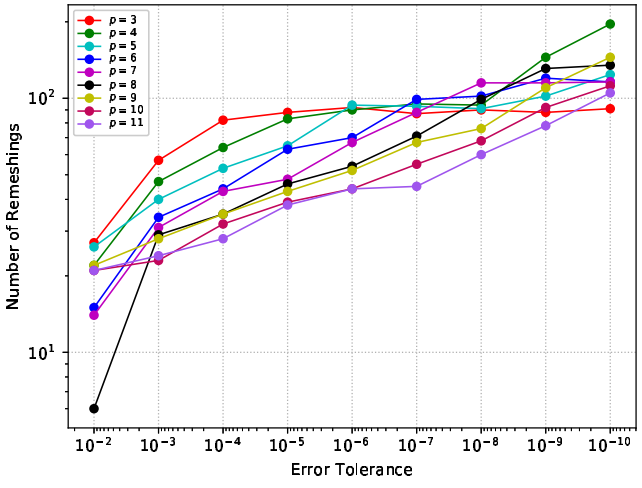


Figure 94: BACOL/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 3 \ldots 11$

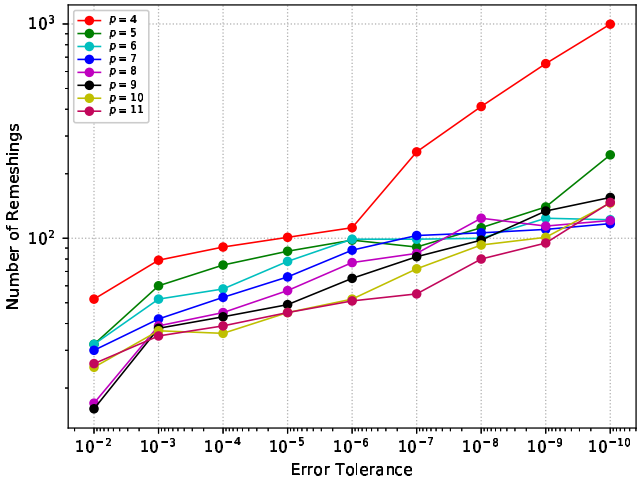


Figure 95: BACOLI/LE Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

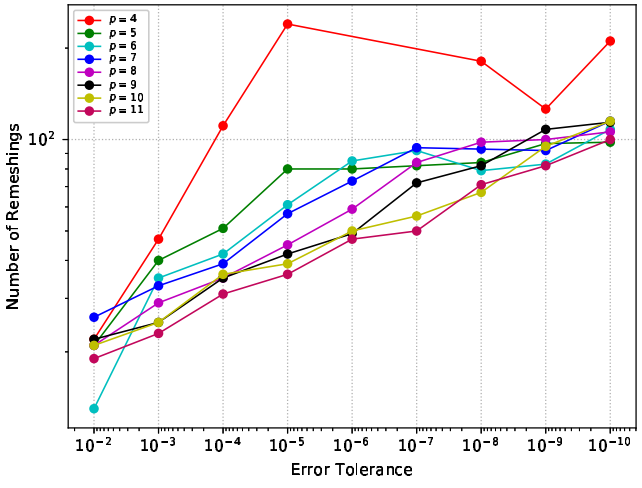


Figure 96: BACOLI/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

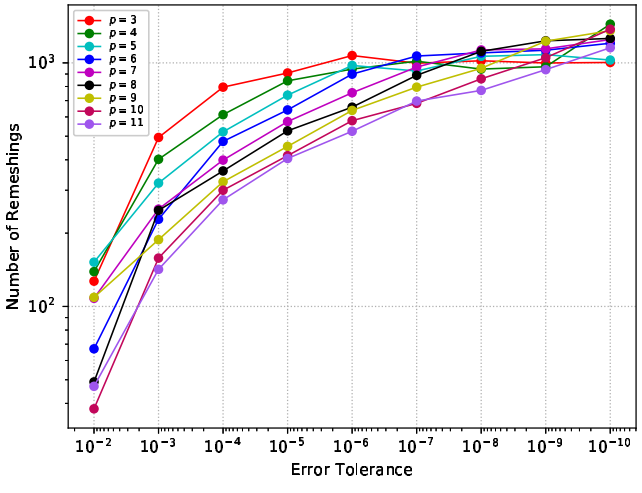


Figure 97: BACOL/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 3 \ldots 11$

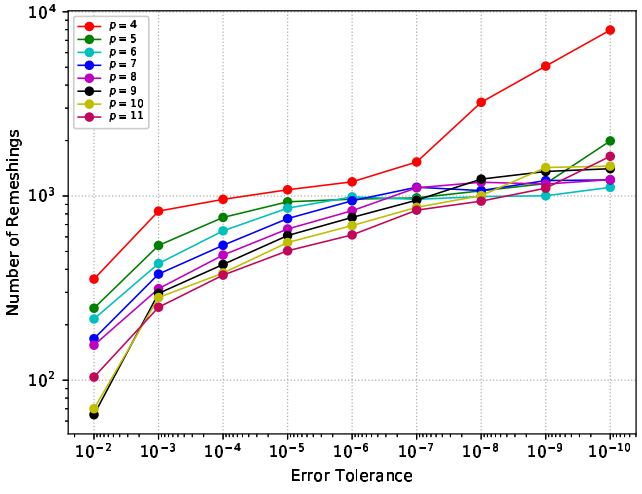


Figure 98: BACOLI/LE Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

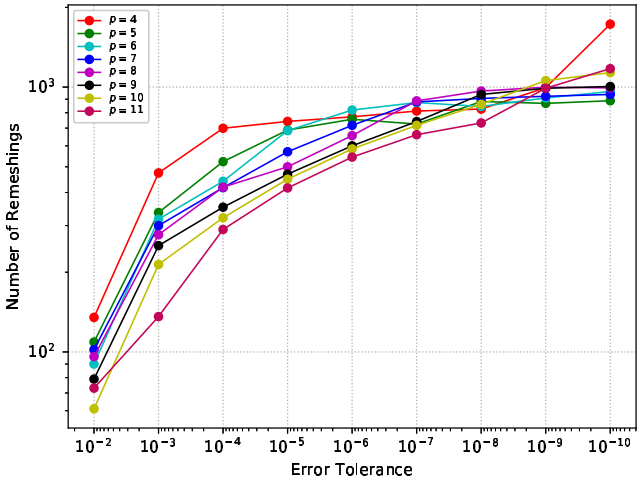


Figure 99: BACOLI/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

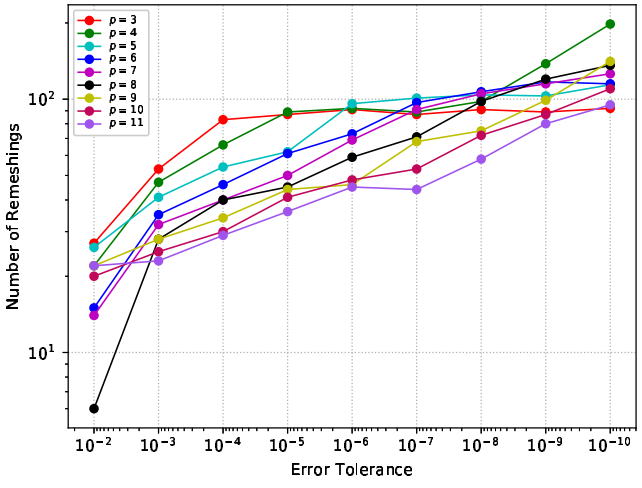


Figure 100: BACOL/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 3 \ldots 11$

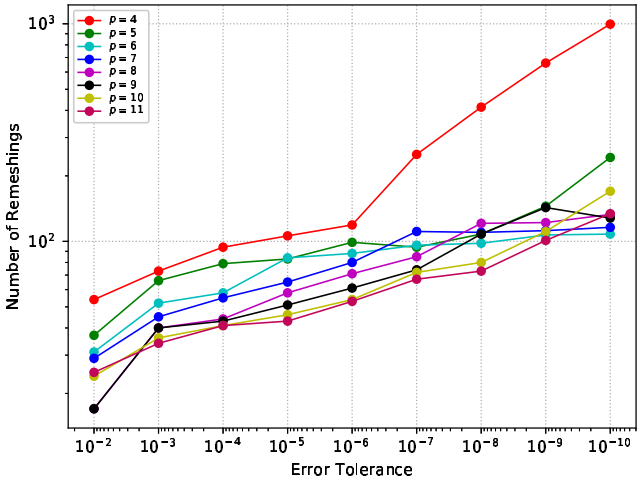


Figure 101: BACOLI/LE Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

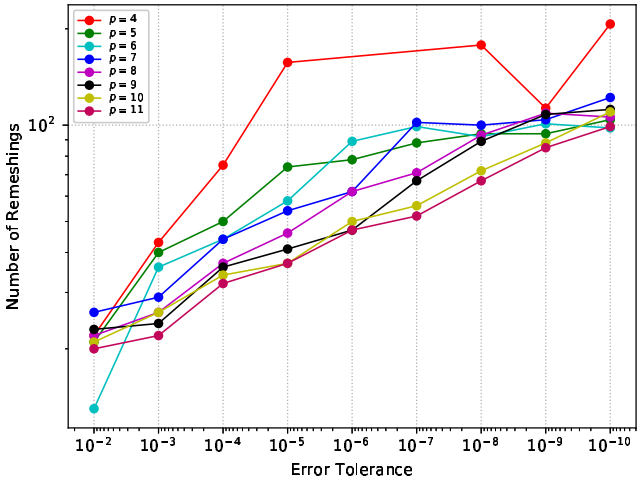


Figure 102: BACOLI/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 4 \ldots 11$

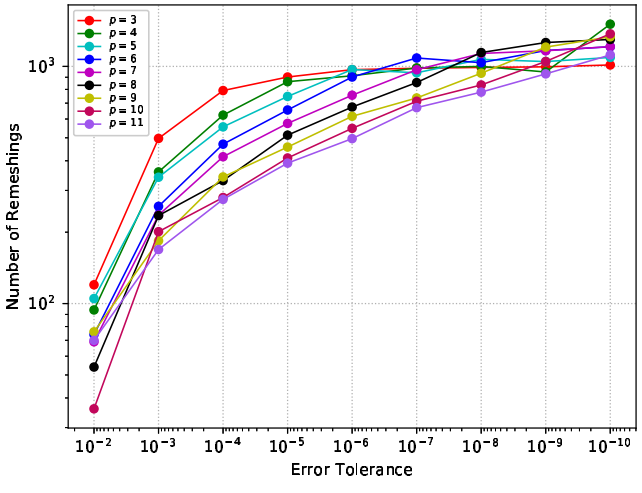


Figure 103: BACOL/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 3 \ldots 11$

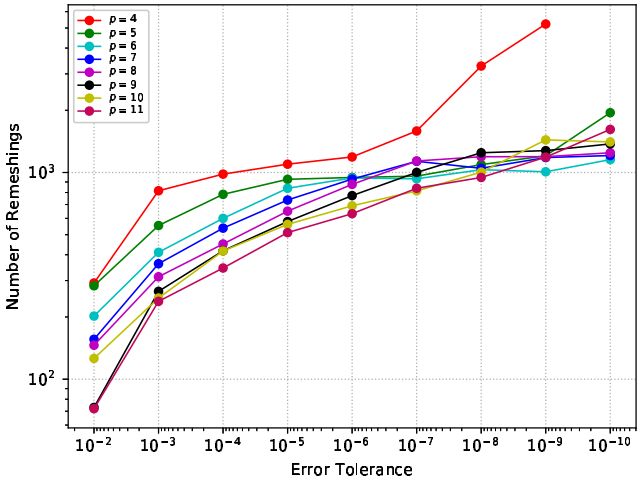


Figure 104: BACOLI/LE Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

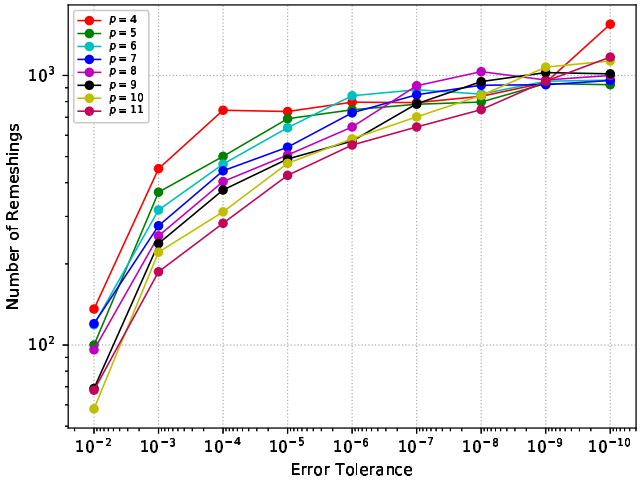


Figure 105: BACOLI/ST Number of Remeshings vs. Error Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 4 \ldots 11$

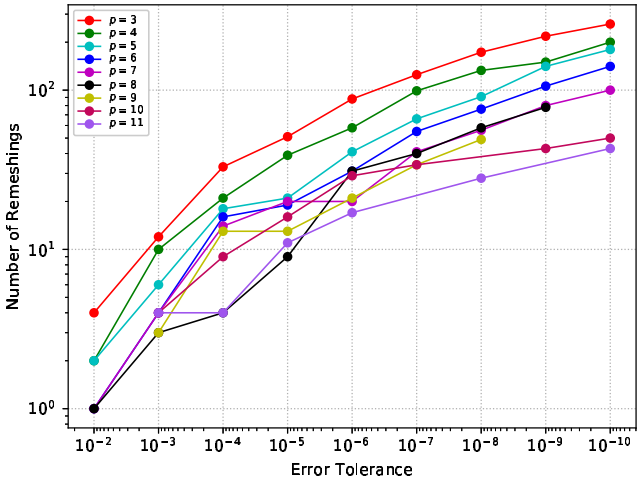


Figure 106: BACOL/ST Number of Remeshings vs. Error Tolerance: Catalytic Surface Reaction Model with $p = 3 \ldots 11$

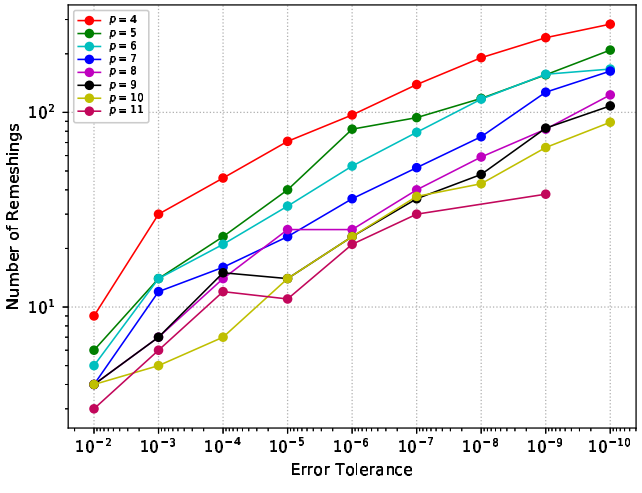


Figure 107: BACOLI/LE Number of Remeshings vs. Error Tolerance: Catalytic Surface Reaction Model with $p = 4 \ldots 11$

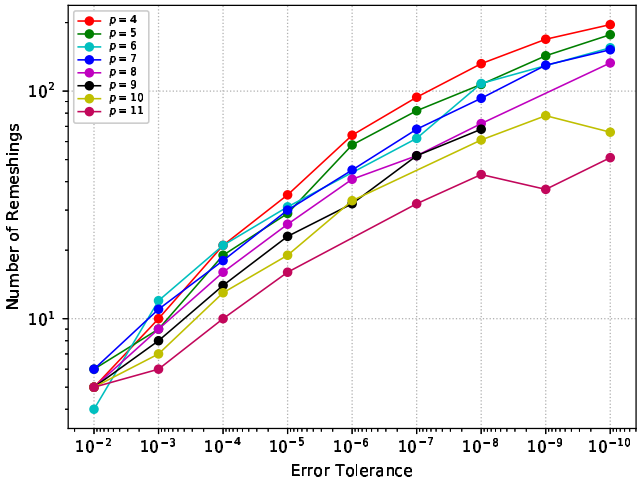


Figure 108: BACOLI/ST Number of Remeshings vs. Error Tolerance: Catalytic Surface Reaction Model with $p = 4 \ldots 11$

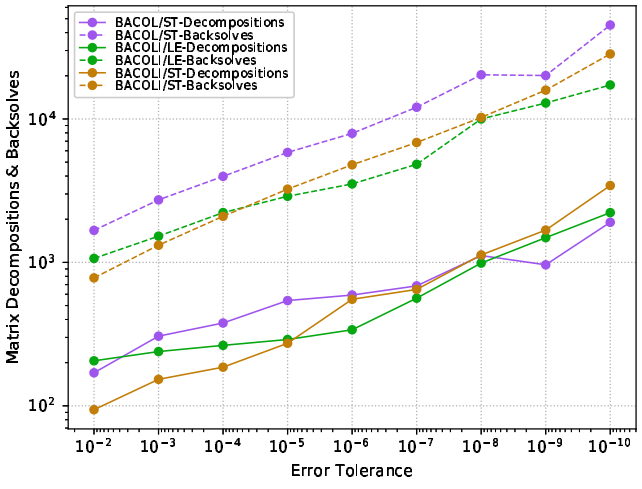


Figure 109: Number of Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4$

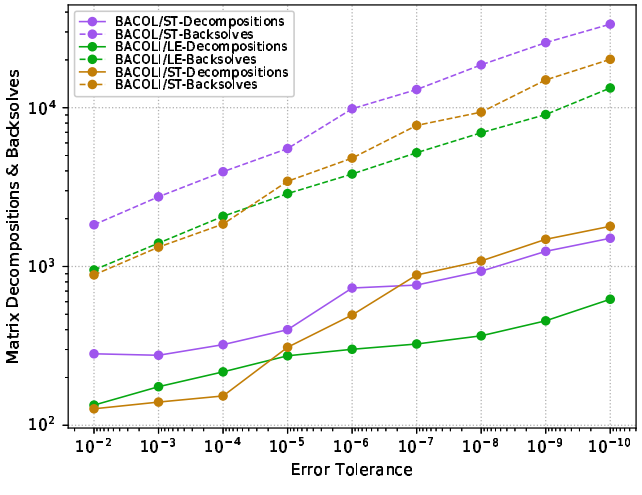


Figure 110: Number of Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 5$

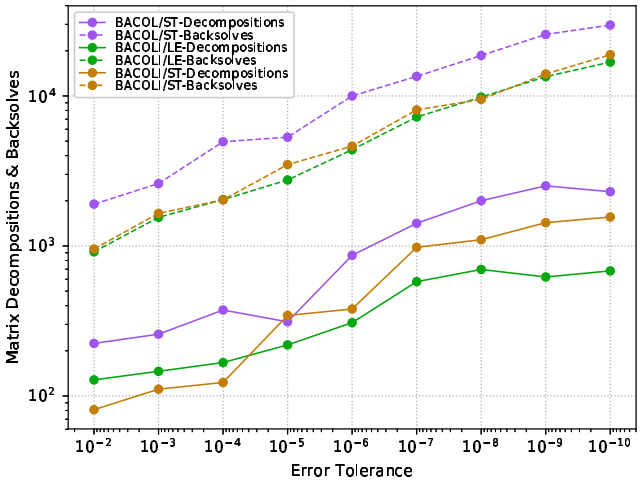


Figure 111: Number of Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 7$

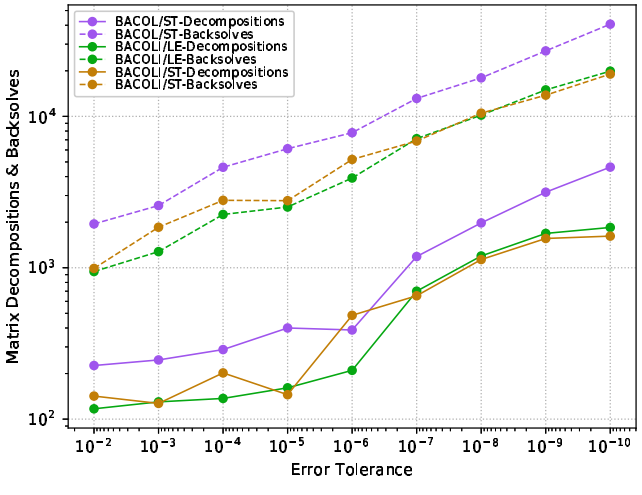


Figure 112: Number of Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 9$

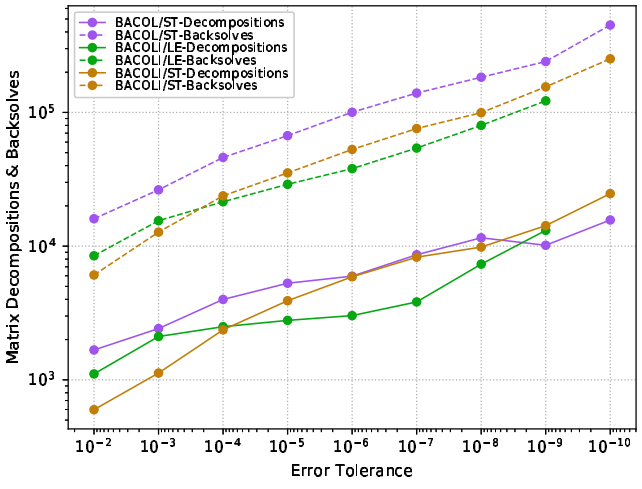


Figure 113: Number of Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4$

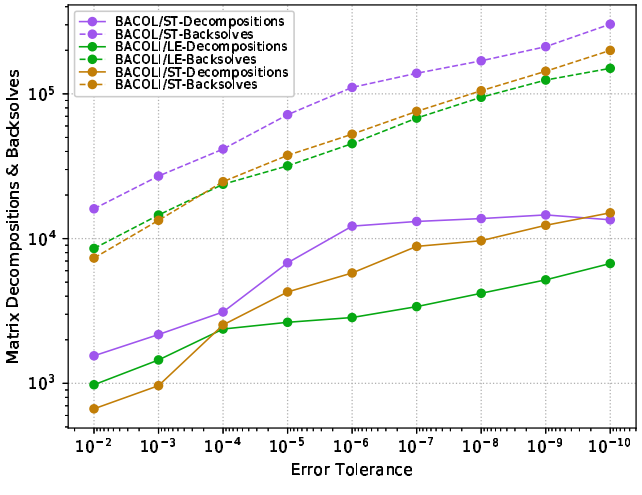


Figure 114: Number of Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 5$

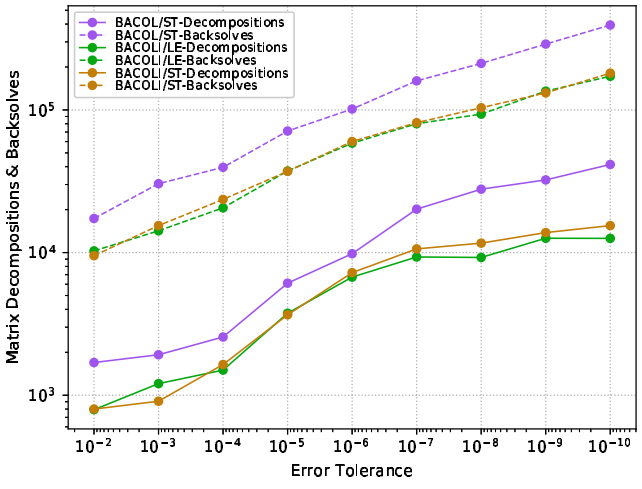


Figure 115: Number of Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 7$

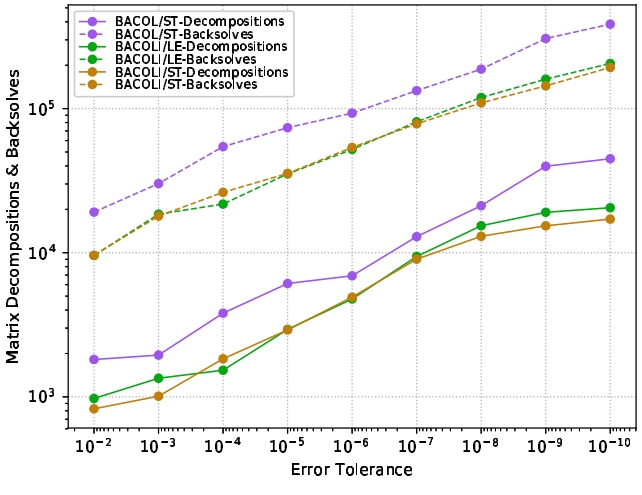


Figure 116: Number of Matrix Factorizations and Backsolves vs. Tolerance: One Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 9$

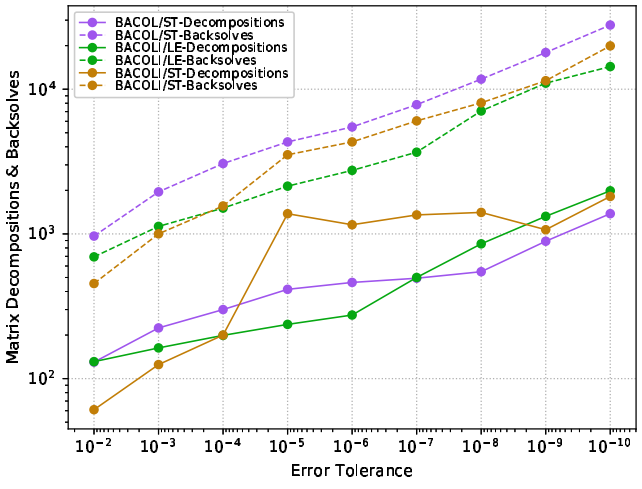


Figure 117: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 4$

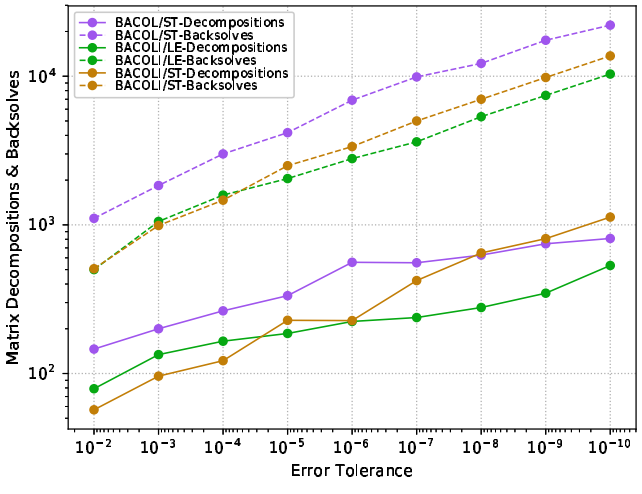


Figure 118: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 5$

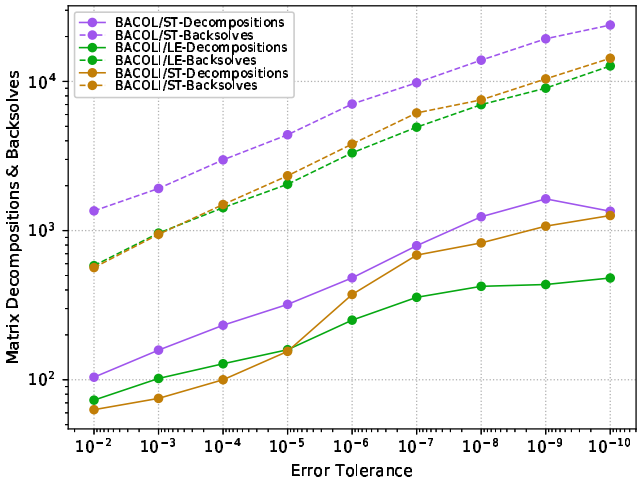


Figure 119: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 7$

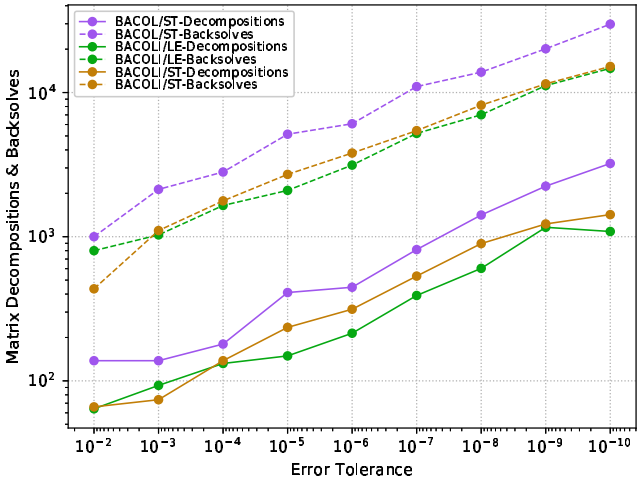


Figure 120: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-3}$ with $p = 9$

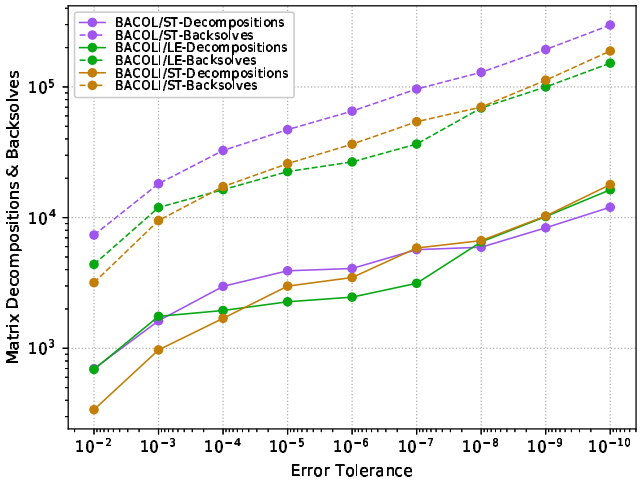


Figure 121: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 4$

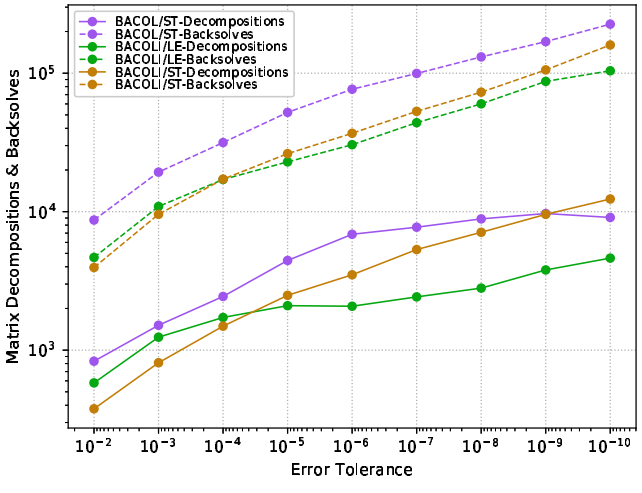


Figure 122: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 5$

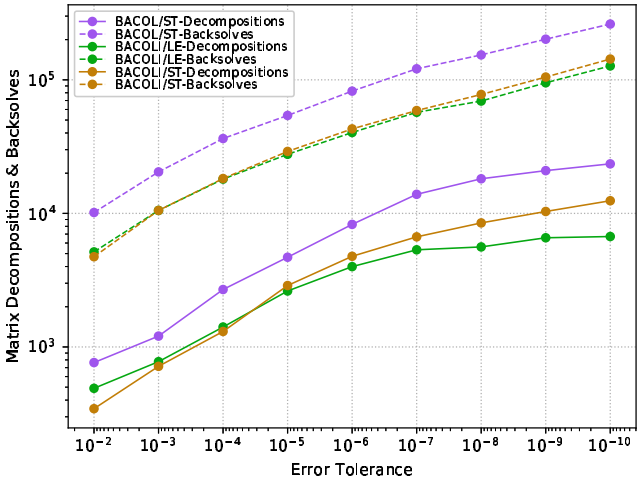


Figure 123: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 7$

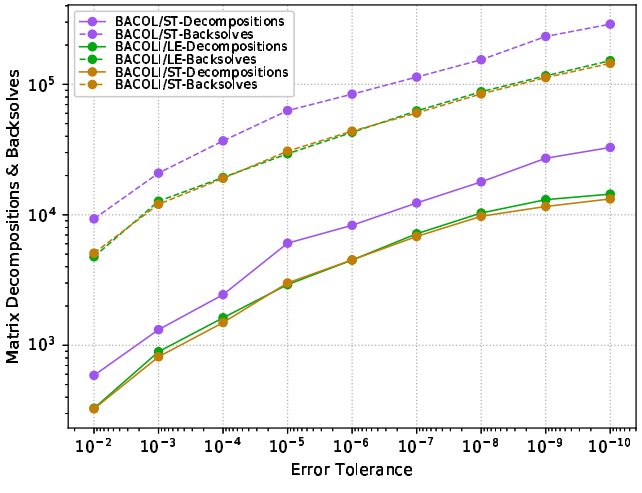


Figure 124: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation, $\epsilon = 10^{-4}$ with $p = 9$

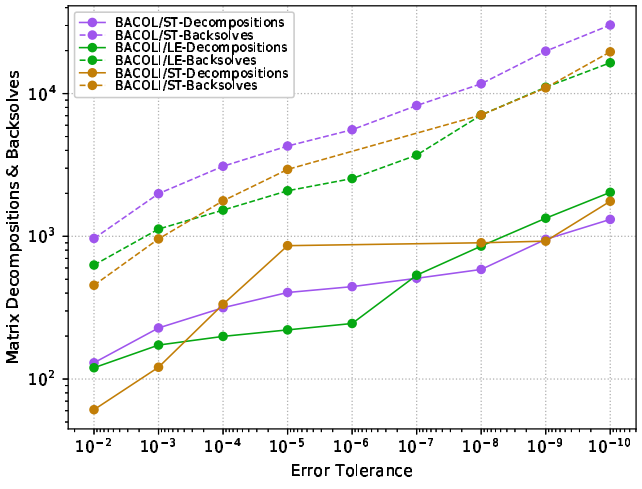


Figure 125: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 4$

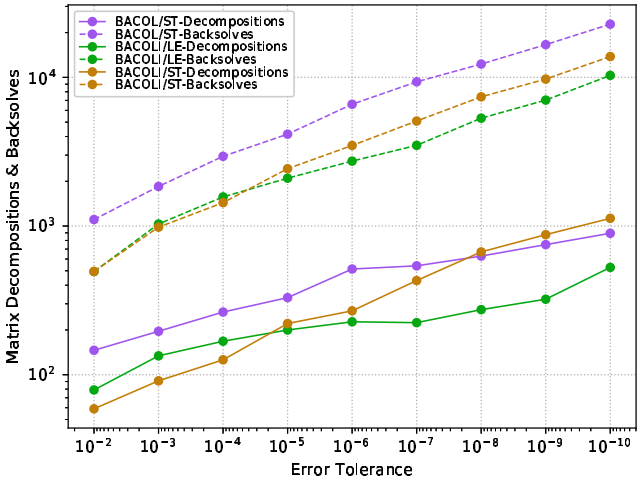


Figure 126: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 5$

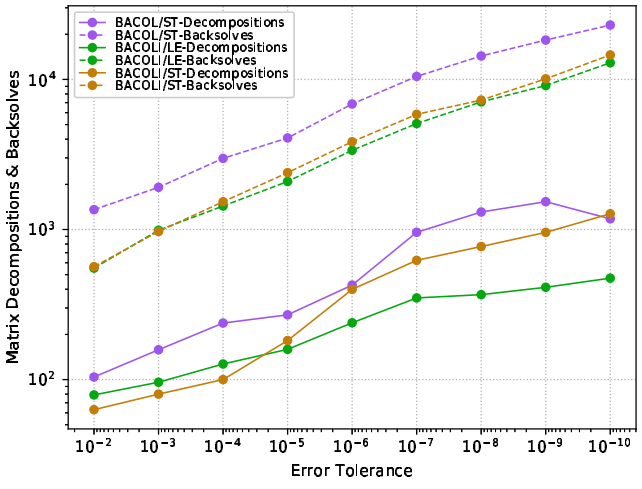


Figure 127: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 7$

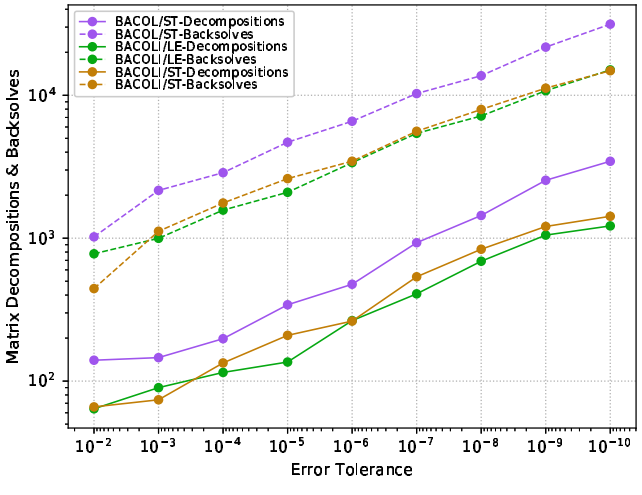


Figure 128: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-3}$ with $p = 9$

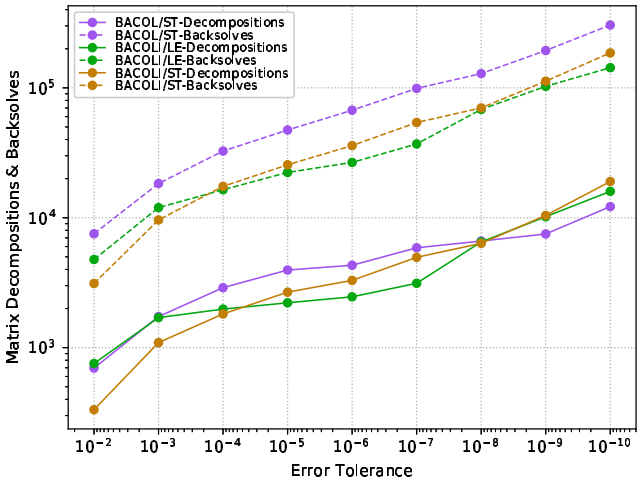


Figure 129: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 4$

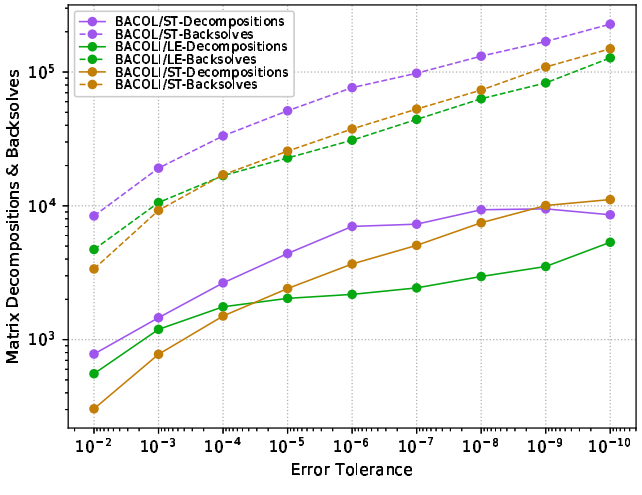


Figure 130: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 5$

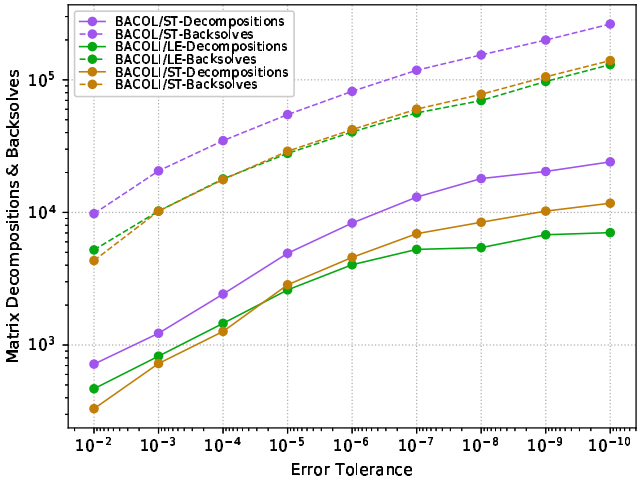


Figure 131: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 7$

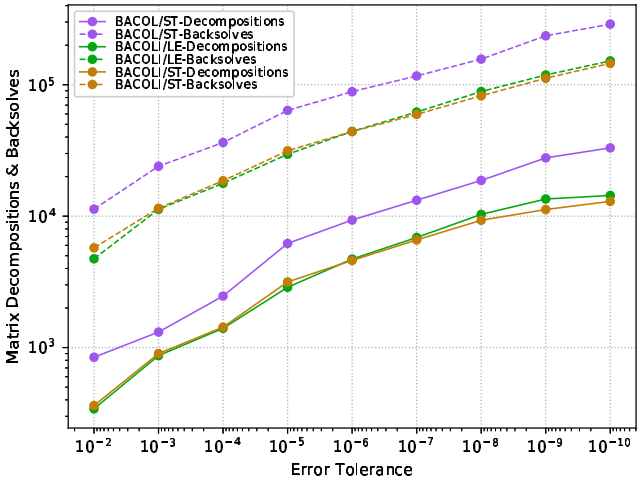


Figure 132: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×6, $\epsilon = 10^{-4}$ with $p = 9$

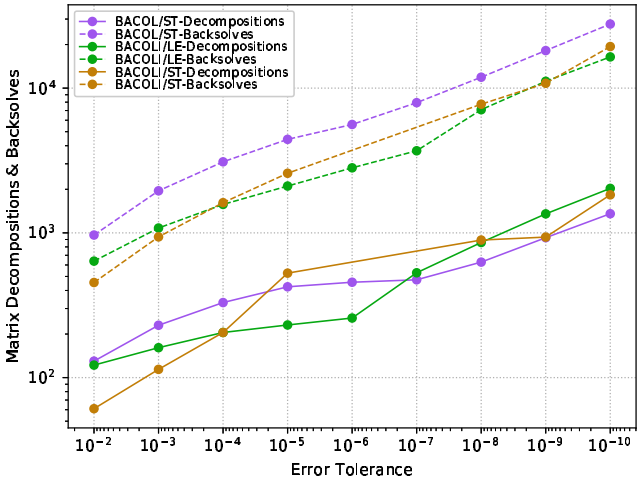


Figure 133: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 4$

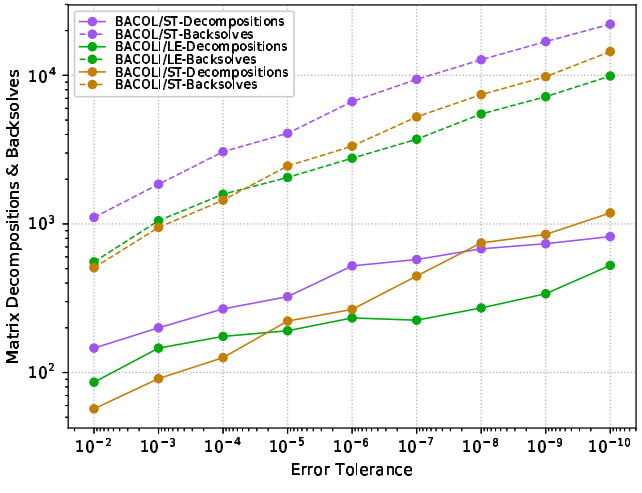


Figure 134: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 5$

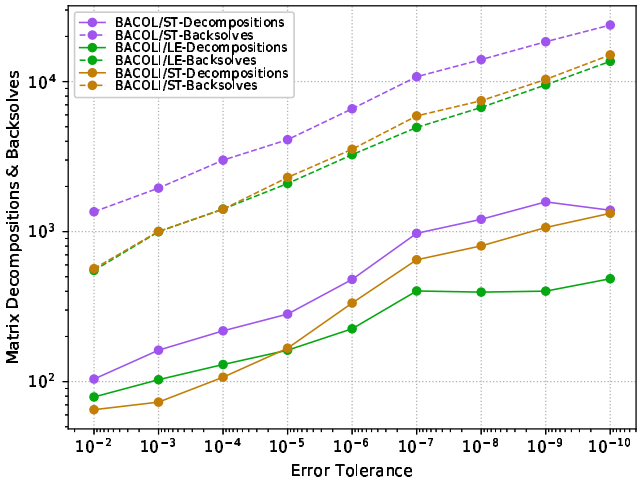


Figure 135: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 7$

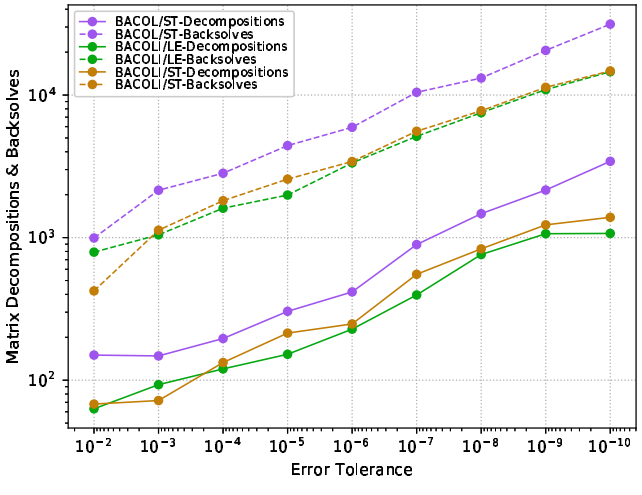


Figure 136: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-3}$ with $p = 9$

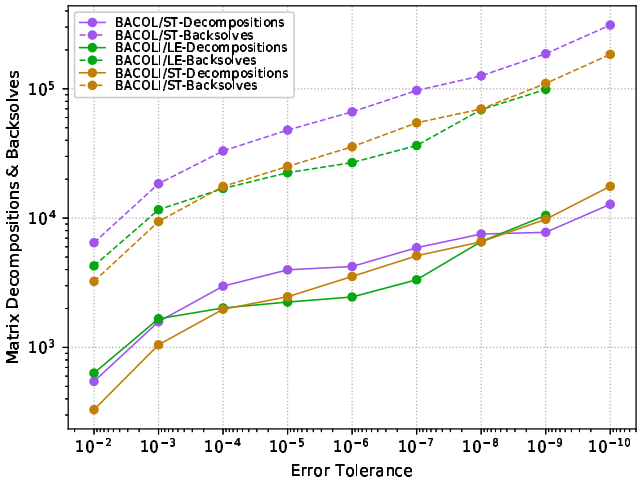


Figure 137: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 4$

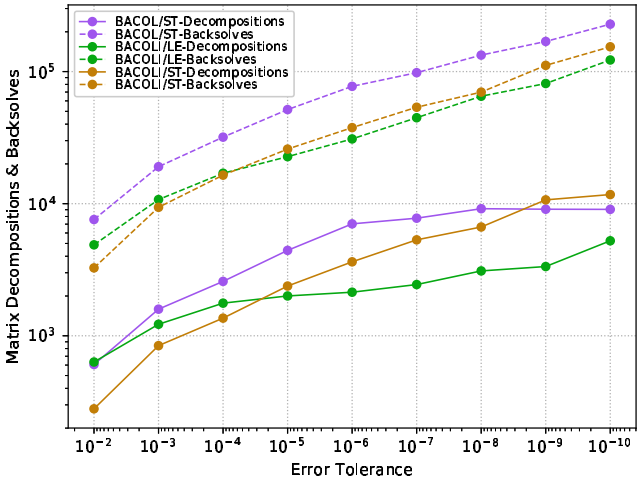


Figure 138: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 5$

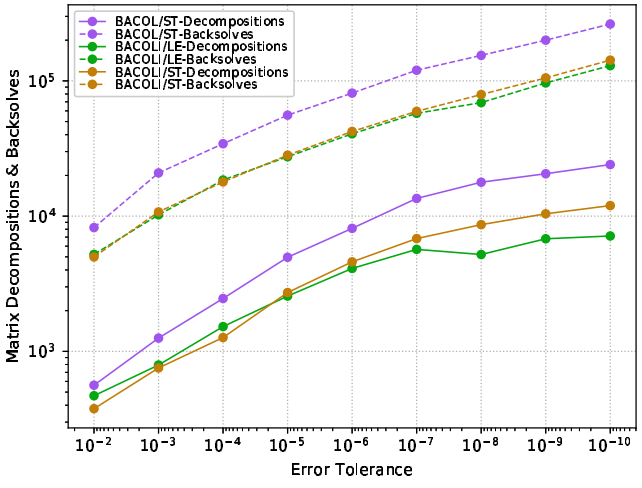


Figure 139: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 7$

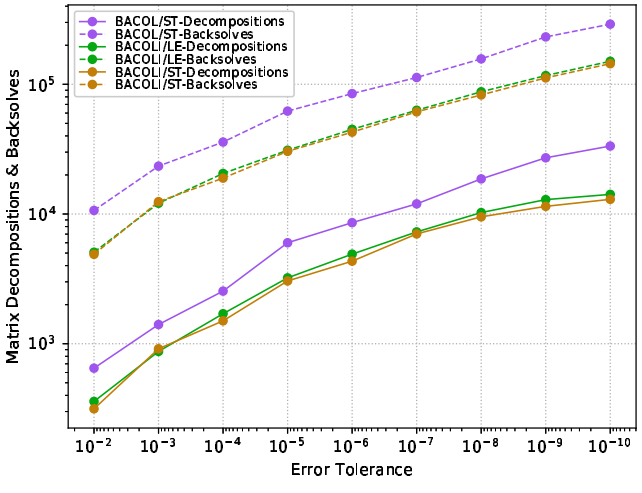


Figure 140: Number of Matrix Factorizations and Backsolves vs. Tolerance: Two Layer Burgers Equation×12, $\epsilon = 10^{-4}$ with $p = 9$

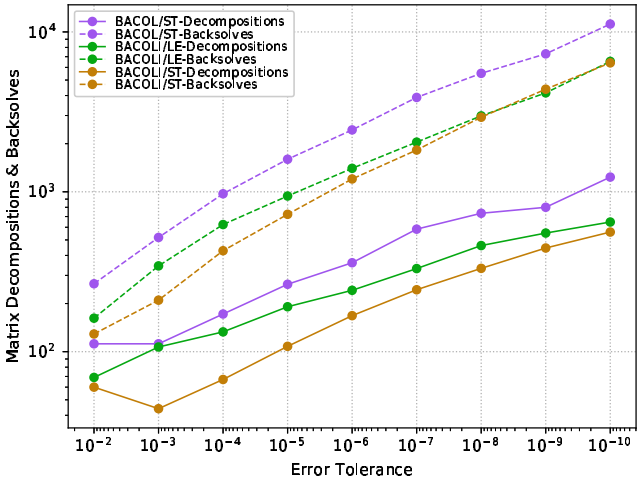


Figure 141: Number of Matrix Factorizations and Backsolves vs. Tolerance: Catalytic Surface Reaction Model with $p = 4$

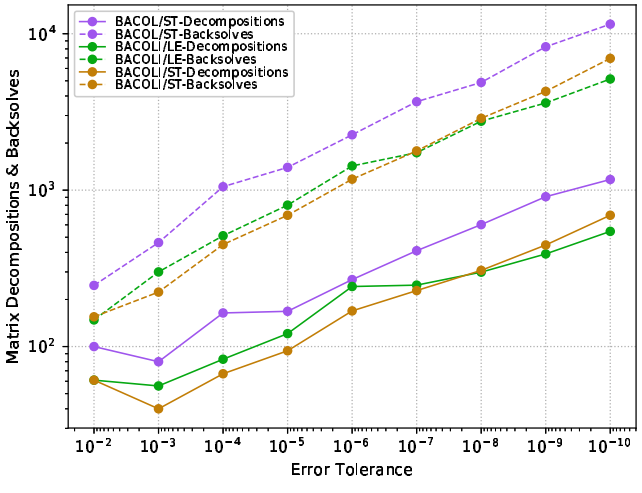


Figure 142: Number of Matrix Factorizations and Backsolves vs. Tolerance: Catalytic Surface Reaction Model with $p = 5$

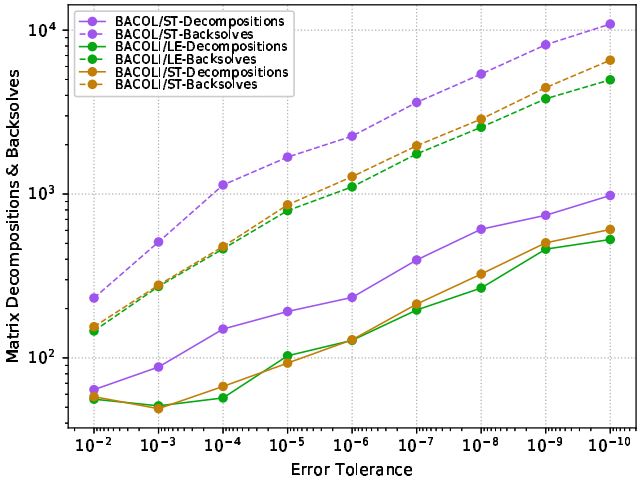


Figure 143: Number of Matrix Factorizations and Backsolves vs. Tolerance: Catalytic Surface Reaction Model with $p = 7$

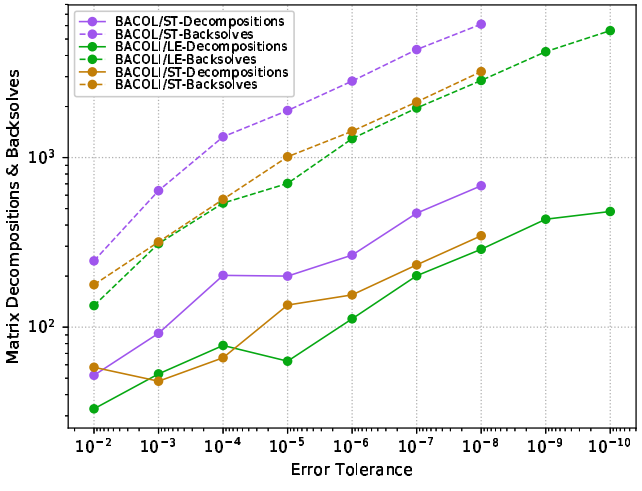


Figure 144: Number of Matrix Factorizations and Backsolves vs. Tolerance: Catalytic Surface Reaction Model with $p = 9$

| $tol = 10^{-4}/p =$ | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|
| BACOL/ST | 0.07 | 0.08 | 0.12 | 0.12 | 0.16 | 0.18 | 0.23 | 0.26 |
| BACOL/LE | 0.06 | 0.07 | 0.08 | 0.12 | 0.12 | 0.16 | 0.18 | 0.23 |
| BACOLI/ST | 0.04 | 0.04 | 0.06 | 0.06 | 0.08 | 0.10 | 0.11 | 0.14 |
| BACOLI/LE | 0.05 | 0.05 | 0.06 | 0.06 | 0.08 | 0.09 | 0.11 | 0.14 |
| $tol = 10^{-6}/p =$ | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| BACOL/ST | 0.21 | 0.25 | 0.26 | 0.30 | 0.37 | 0.41 | 0.49 | 0.54 |
| BACOL/LE | 0.23 | 0.21 | 0.25 | 0.26 | 0.30 | 0.37 | 0.41 | 0.49 |
| BACOLI/ST | 0.14 | 0.14 | 0.15 | 0.16 | 0.19 | 0.22 | 0.26 | 0.30 |
| BACOLI/LE | 0.18 | 0.14 | 0.15 | 0.16 | 0.19 | 0.21 | 0.24 | 0.27 |
| $tol = 10^{-8}/p =$ | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| BACOL/ST | 1.02 | 0.80 | 0.75 | 0.75 | 0.80 | 0.82 | 0.96 | 0.98 |
| BACOL/LE | 1.16 | 1.02 | 0.80 | 0.75 | 0.75 | 0.80 | 0.82 | 0.96 |
| BACOLI/ST | 0.59 | 0.44 | 0.40 | 0.44 | 0.49 | 0.54 | 0.51 | 0.58 |
| BACOLI/LE | 1.36 | 0.51 | 0.50 | 0.50 | 0.50 | 0.52 | 0.50 | 0.56 |
| $tol = 10^{-10}/p =$ | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| BACOL/ST | 4.55 | 2.36 | 2.12 | 1.67 | 1.90 | 2.05 | 1.88 | 1.87 |
| BACOL/LE | 5.12 | 4.55 | 2.36 | 2.12 | 1.67 | 1.90 | 2.05 | 1.88 |
| BACOLI/ST | 2.98 | 1.51 | 1.25 | 1.12 | 1.12 | 1.12 | 1.16 | 1.28 |
| BACOLI/LE | 6.43 | 2.13 | 1.51 | 1.24 | 1.14 | 1.23 | 1.31 | 1.21 |

Table 10: *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 |
|---|---|---|---|---|---|---|---|---|
| BACOL/ST | 0.76 | 0.94 | 1.17 | 1.52 | 1.77 | 2.33 | 2.65 | 2.99 |
| BACOL/LE | 0.60 | 0.76 | 0.94 | 1.17 | 1.52 | 1.77 | 2.33 | 2.65 |
| BACOLI/ST | 0.42 | 0.53 | 0.59 | 0.76 | 0.95 | 1.17 | 1.47 | 1.81 |
| BACOLI/LE | 0.48 | 0.55 | 0.60 | 0.75 | 0.94 | 1.14 | 1.42 | 1.62 |
| $tol = 10^{-6}/p =$ | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| BACOL/ST | 2.50 | 2.85 | 3.12 | 3.37 | 3.97 | 4.57 | 5.56 | 6.59 |
| BACOL/LE | 2.37 | 2.50 | 2.85 | 3.12 | 3.37 | 3.97 | 4.57 | 5.56 |
| BACOLI/ST | 1.39 | 1.49 | 1.73 | 2.13 | 2.28 | 2.64 | 3.10 | 3.71 |
| BACOLI/LE | 1.62 | 1.55 | 1.80 | 2.15 | 2.30 | 2.59 | 2.93 | 3.38 |
| $tol = 10^{-8}/p =$ | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| BACOL/ST | 8.60 | 6.96 | 7.76 | 9.06 | 10.01 | 9.33 | 10.22 | 11.17 |
| BACOL/LE | 11.96 | 8.60 | 6.96 | 7.76 | 9.06 | 10.01 | 9.33 | 10.22 |
| BACOLI/ST | 5.10 | 4.67 | 4.23 | 4.87 | 5.39 | 6.18 | 6.42 | 7.08 |
| BACOLI/LE | 8.60 | 5.78 | 4.84 | 4.89 | 5.76 | 6.63 | 6.05 | 6.42 |
| $tol = 10^{-10}/p =$ | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| BACOL/ST | 41.22 | 20.00 | 20.60 | 20.53 | 20.62 | 21.21 | 23.15 | 21.54 |
| BACOL/LE | 55.33 | 41.22 | 20.00 | 20.60 | 20.53 | 20.62 | 21.21 | 23.15 |
| BACOLI/ST | 23.61 | 13.85 | 11.25 | 10.20 | 10.98 | 11.64 | 12.46 | 14.08 |
| BACOLI/LE | — | 19.86 | 13.73 | 11.90 | 13.03 | 13.08 | 13.59 | 15.54 |

Table 11: *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 |
|---|---|---|---|---|---|---|---|---|
| BACOL/ST | 0.06 | 0.07 | 0.09 | 0.11 | 0.13 | 0.17 | 0.19 | 0.21 |
| BACOL/LE | 0.06 | 0.06 | 0.07 | 0.09 | 0.11 | 0.13 | 0.17 | 0.19 |
| BACOLI/ST | 0.03 | 0.04 | 0.04 | 0.05 | 0.06 | 0.08 | 0.10 | 0.11 |
| BACOLI/LE | 0.04 | 0.04 | 0.05 | 0.06 | 0.06 | 0.08 | 0.09 | 0.12 |
| $tol = 10^{-6}/p =$ | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| BACOL/ST | 0.18 | 0.21 | 0.22 | 0.27 | 0.30 | 0.33 | 0.39 | 0.43 |
| BACOL/LE | 0.23 | 0.18 | 0.21 | 0.22 | 0.27 | 0.30 | 0.33 | 0.39 |
| BACOLI/ST | 0.20 | 0.12 | 0.13 | 0.15 | 0.18 | 0.19 | 0.22 | 0.26 |
| BACOLI/LE | 0.18 | 0.13 | 0.13 | 0.15 | 0.16 | 0.18 | 0.19 | 0.22 |
| $tol = 10^{-8}/p =$ | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| BACOL/ST | 0.81 | 0.66 | 0.66 | 0.68 | 0.70 | 0.77 | 0.83 | 0.91 |
| BACOL/LE | 1.08 | 0.81 | 0.66 | 0.66 | 0.68 | 0.70 | 0.77 | 0.83 |
| BACOLI/ST | 0.73 | 0.41 | 0.37 | 0.41 | 0.45 | 0.48 | 0.48 | 0.54 |
| BACOLI/LE | 1.44 | 0.52 | 0.43 | 0.43 | 0.45 | 0.44 | 0.48 | 0.51 |
| $tol = 10^{-10}/p =$ | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| BACOL/ST | 4.11 | 2.15 | 1.90 | 1.65 | 1.70 | 1.81 | 1.81 | 1.78 |
| BACOL/LE | 5.79 | 4.11 | 2.15 | 1.90 | 1.65 | 1.70 | 1.81 | 1.81 |
| BACOLI/ST | 3.00 | 1.33 | 1.06 | 1.04 | 1.03 | 1.05 | 1.05 | 1.10 |
| BACOLI/LE | 7.82 | 2.29 | 1.31 | 1.19 | 1.05 | 1.11 | 1.18 | 1.14 |

Table 12: *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 |
|---|---|---|---|---|---|---|---|---|
| BACOL/ST | 0.73 | 0.89 | 1.12 | 1.48 | 1.94 | 1.98 | 2.58 | 2.92 |
| BACOL/LE | 0.59 | 0.73 | 0.89 | 1.12 | 1.48 | 1.94 | 1.98 | 2.58 |
| BACOLI/ST | 0.41 | 0.46 | 0.55 | 0.72 | 0.92 | 1.09 | 1.26 | 1.55 |
| BACOLI/LE | 0.47 | 0.50 | 0.60 | 0.74 | 0.92 | 1.11 | 1.35 | 1.56 |
| $tol = 10^{-6}/p =$ | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| BACOL/ST | 2.15 | 2.50 | 2.93 | 3.64 | 4.32 | 5.00 | 5.98 | 6.73 |
| BACOL/LE | 2.18 | 2.15 | 2.50 | 2.93 | 3.64 | 4.32 | 5.00 | 5.98 |
| BACOLI/ST | 1.35 | 1.38 | 1.56 | 1.94 | 2.36 | 2.68 | 3.22 | 3.83 |
| BACOLI/LE | 1.50 | 1.40 | 1.61 | 1.90 | 2.30 | 2.68 | 3.14 | 3.64 |
| $tol = 10^{-8}/p =$ | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| BACOL/ST | 8.08 | 7.13 | 7.44 | 8.41 | 9.47 | 10.11 | 11.36 | 12.53 |
| BACOL/LE | 11.67 | 8.08 | 7.13 | 7.44 | 8.41 | 9.47 | 10.11 | 11.36 |
| BACOLI/ST | 4.83 | 4.26 | 4.13 | 4.63 | 5.22 | 5.90 | 6.47 | 7.28 |
| BACOLI/LE | 11.37 | 5.15 | 4.81 | 4.71 | 5.59 | 6.15 | 6.71 | 7.36 |
| $tol = 10^{-10}/p =$ | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| BACOL/ST | 36.48 | 20.14 | 19.62 | 17.93 | 19.46 | 20.91 | 22.58 | 21.98 |
| BACOL/LE | 49.61 | 36.48 | 20.14 | 19.62 | 17.93 | 19.46 | 20.91 | 22.58 |
| BACOLI/ST | 24.34 | 14.01 | 11.12 | 10.57 | 10.87 | 11.12 | 12.13 | 13.98 |
| BACOLI/LE | 69.02 | 19.38 | 13.86 | 11.92 | 11.98 | 12.68 | 13.36 | 14.66 |

Table 13: *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 |
|---|---|---|---|---|---|---|---|---|
| BACOL/ST | 0.33 | 0.42 | 0.55 | 0.75 | 0.80 | 1.20 | 1.36 | 1.61 |
| BACOL/LE | 0.30 | 0.33 | 0.42 | 0.55 | 0.75 | 0.80 | 1.20 | 1.36 |
| BACOLI/ST | 0.22 | 0.20 | 0.23 | 0.28 | 0.36 | 0.47 | 0.62 | 0.70 |
| BACOLI/LE | 0.21 | 0.23 | 0.26 | 0.34 | 0.38 | 0.49 | 0.53 | 0.73 |
| $tol = 10^{-6}/p =$ | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| BACOL/ST | 1.10 | 1.23 | 1.39 | 1.80 | 2.09 | 2.78 | 3.10 | 3.99 |
| BACOL/LE | 1.39 | 1.10 | 1.23 | 1.39 | 1.80 | 2.09 | 2.78 | 3.10 |
| BACOLI/ST | — | 0.69 | 0.81 | 1.00 | 1.15 | 1.30 | 1.67 | 1.91 |
| BACOLI/LE | 0.94 | 0.69 | 0.76 | 0.83 | 1.02 | 1.22 | 1.34 | 1.67 |
| $tol = 10^{-8}/p =$ | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| BACOL/ST | 5.76 | 4.71 | 4.49 | 5.29 | 6.04 | 7.39 | 8.71 | 9.74 |
| BACOL/LE | 9.64 | 5.76 | 4.71 | 4.49 | 5.29 | 6.04 | 7.39 | 8.71 |
| BACOLI/ST | 3.73 | 2.86 | 2.40 | 2.82 | 3.23 | 3.70 | 4.07 | 4.85 |
| BACOLI/LE | 10.29 | 3.20 | 2.74 | 2.69 | 3.09 | 3.32 | 3.99 | 4.56 |
| $tol = 10^{-10}/p =$ | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| BACOL/ST | 26.15 | 13.11 | 11.72 | 10.37 | 12.68 | 15.83 | 15.37 | 16.52 |
| BACOL/LE | 40.60 | 26.15 | 13.11 | 11.72 | 10.37 | 12.68 | 15.83 | 15.37 |
| BACOLI/ST | 16.71 | 8.47 | 6.89 | 7.15 | 7.51 | 7.65 | 8.36 | 8.60 |
| BACOLI/LE | 53.63 | 12.30 | 7.93 | 7.07 | 6.48 | 7.93 | 8.41 | 9.61 |

Table 14: *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 |
|---|---|---|---|---|---|---|---|---|
| BACOL/ST | 4.16 | 5.40 | 7.38 | 9.96 | 12.74 | 15.68 | 21.24 | 26.38 |
| BACOL/LE | 3.25 | 4.16 | 5.40 | 7.38 | 9.96 | 12.74 | 15.68 | 21.24 |
| BACOLI/ST | 2.18 | 2.59 | 3.16 | 4.16 | 5.73 | 7.02 | 8.89 | 10.46 |
| BACOLI/LE | 2.46 | 2.78 | 3.50 | 4.39 | 5.68 | 6.97 | 10.68 | 11.20 |
| $tol = 10^{-6}/p =$ | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| BACOL/ST | 13.14 | 16.34 | 20.99 | 28.09 | 35.89 | 47.81 | 59.47 | 76.61 |
| BACOL/LE | 13.28 | 13.14 | 16.34 | 20.99 | 28.09 | 35.89 | 47.81 | 59.47 |
| BACOLI/ST | 7.69 | 8.67 | 10.21 | 13.03 | 16.89 | 20.60 | 26.68 | 31.38 |
| BACOLI/LE | 8.36 | 8.03 | 9.89 | 12.36 | 16.25 | 20.80 | 25.01 | 29.22 |
| $tol = 10^{-8}/p =$ | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| BACOL/ST | 58.20 | 52.24 | 56.82 | 73.24 | 90.09 | 106.72 | 133.77 | 155.88 |
| BACOL/LE | 91.12 | 58.20 | 52.24 | 56.82 | 73.24 | 90.09 | 106.72 | 133.77 |
| BACOLI/ST | 29.63 | 29.36 | 29.38 | 34.72 | 40.63 | 47.09 | 57.77 | 64.13 |
| BACOLI/LE | 71.78 | 32.75 | 30.66 | 32.13 | 42.40 | 50.76 | 56.98 | 67.54 |
| $tol = 10^{-10}/p =$ | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| BACOL/ST | 231.94 | 124.86 | 134.63 | 130.26 | 150.47 | 176.64 | 202.04 | 213.03 |
| BACOL/LE | 338.08 | 231.94 | 124.86 | 134.63 | 130.26 | 150.47 | 176.64 | 202.04 |
| BACOLI/ST | 146.50 | 87.59 | 78.55 | 73.45 | 79.87 | 84.08 | 99.86 | 111.83 |
| BACOLI/LE | 356.08 | 125.11 | 82.80 | 79.74 | 80.18 | 97.44 | 108.21 | 128.52 |

Table 15: *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 |
|---|---|---|---|---|---|---|---|---|
| BACOL/ST | 1.14 | 1.70 | 2.26 | 3.35 | 4.75 | 8.62 | 10.96 | 13.74 |
| BACOL/LE | 0.98 | 1.14 | 1.70 | 2.26 | 3.35 | 4.75 | 8.62 | 10.96 |
| BACOLI/ST | 0.57 | 0.63 | 0.79 | 1.22 | 1.53 | 2.26 | 3.11 | 3.80 |
| BACOLI/LE | 0.70 | 0.78 | 0.94 | 1.41 | 1.60 | 2.20 | 3.03 | 4.51 |
| $tol = 10^{-6}/p =$ | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| BACOL/ST | 4.11 | 4.95 | 6.10 | 9.23 | 13.77 | 18.77 | 25.85 | 32.15 |
| BACOL/LE | 5.31 | 4.11 | 4.95 | 6.10 | 9.23 | 13.77 | 18.77 | 25.85 |
| BACOLI/ST | — | 2.14 | 3.05 | 4.06 | 5.56 | 6.20 | 9.43 | 11.89 |
| BACOLI/LE | 3.41 | 2.34 | 2.56 | 3.37 | 4.22 | 5.94 | 7.74 | 10.32 |
| $tol = 10^{-8}/p =$ | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| BACOL/ST | 31.32 | 25.10 | 25.97 | 33.97 | 42.41 | 57.90 | 71.45 | 80.10 |
| BACOL/LE | 43.42 | 31.32 | 25.10 | 25.97 | 33.97 | 42.41 | 57.90 | 71.45 |
| BACOLI/ST | 15.93 | 10.50 | 9.92 | 13.66 | 16.50 | 21.46 | 24.85 | 31.46 |
| BACOLI/LE | 62.57 | 15.49 | 11.23 | 11.47 | 14.59 | 20.51 | 23.99 | 29.55 |
| $tol = 10^{-10}/p =$ | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| BACOL/ST | 104.60 | 50.60 | 43.55 | 41.99 | 54.82 | 74.82 | 78.78 | 88.81 |
| BACOL/LE | 121.89 | 104.60 | 50.60 | 43.55 | 41.99 | 54.82 | 74.82 | 78.78 |
| BACOLI/ST | 50.40 | 26.12 | 21.11 | 26.79 | 27.52 | 31.21 | 34.20 | 37.94 |
| BACOLI/LE | 251.94 | 45.15 | 22.43 | 23.04 | 22.82 | 29.31 | 41.26 | 45.57 |

Table 16: *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 |
|---|---|---|---|---|---|---|---|---|
| BACOL/ST | 14.22 | 20.65 | 32.11 | 54.29 | 84.74 | 118.91 | 163.89 | 231.00 |
| BACOL/LE | 10.41 | 14.22 | 20.65 | 32.11 | 54.29 | 84.74 | 118.91 | 163.89 |
| BACOLI/ST | 7.02 | 7.76 | 10.47 | 16.19 | 25.25 | 36.75 | 46.85 | 75.10 |
| BACOLI/LE | 8.13 | 9.34 | 11.76 | 18.68 | 29.46 | 39.26 | 57.39 | 71.09 |
| $tol = 10^{-6}/p =$ | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| BACOL/ST | 47.56 | 69.26 | 108.63 | 175.68 | 263.87 | 370.65 | 477.66 | 624.35 |
| BACOL/LE | 46.25 | 47.56 | 69.26 | 108.63 | 175.68 | 263.87 | 370.65 | 477.66 |
| BACOLI/ST | 24.81 | 28.03 | 38.23 | 59.36 | 85.46 | 115.32 | 165.34 | 215.99 |
| BACOLI/LE | 29.75 | 25.77 | 35.35 | 55.77 | 87.64 | 121.66 | 161.42 | 210.94 |
| $tol = 10^{-8}/p =$ | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| BACOL/ST | 308.35 | 293.76 | 351.12 | 499.28 | 696.64 | 834.78 | 1026.63 | 1271.17 |
| BACOL/LE | 438.01 | 308.35 | 293.76 | 351.12 | 499.28 | 696.64 | 834.78 | 1026.63 |
| BACOLI/ST | 108.18 | 109.71 | 124.33 | 175.10 | 242.90 | 303.39 | 392.73 | 491.10 |
| BACOLI/LE | 427.54 | 144.74 | 137.51 | 161.30 | 251.19 | 334.52 | 408.79 | 529.60 |
| $tol = 10^{-10}/p =$ | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| BACOL/ST | 898.51 | 481.73 | 548.71 | 606.93 | 761.20 | 964.48 | 1090.47 | 1306.07 |
| BACOL/LE | 1080.43 | 898.51 | 481.73 | 548.71 | 606.93 | 761.20 | 964.48 | 1090.47 |
| BACOLI/ST | 440.97 | 272.94 | 244.36 | 257.04 | 301.63 | 366.45 | 443.08 | 533.97 |
| BACOLI/LE | — | 378.53 | 257.65 | 252.23 | 312.74 | 401.22 | 522.54 | 680.23 |

Table 17: *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 |
|---|---|---|---|---|---|---|---|---|
| BACOL/ST | 0.03 | 0.03 | 0.04 | 0.05 | 0.06 | 0.07 | 0.08 | 0.09 |
| BACOL/LE | 0.02 | 0.03 | 0.03 | 0.04 | 0.05 | 0.06 | 0.07 | 0.08 |
| BACOLI/ST | 0.01 | 0.01 | 0.02 | 0.02 | 0.02 | 0.03 | 0.04 | 0.06 |
| BACOLI/LE | 0.02 | 0.02 | 0.02 | 0.03 | 0.02 | 0.04 | 0.04 | 0.05 |
| $tol = 10^{-6}/p =$ | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| BACOL/ST | 0.10 | 0.12 | 0.14 | 0.16 | 0.19 | 0.24 | 0.32 | 0.32 |
| BACOL/LE | 0.12 | 0.10 | 0.12 | 0.14 | 0.16 | 0.19 | 0.24 | 0.32 |
| BACOLI/ST | 0.05 | 0.05 | — | 0.07 | 0.09 | 0.10 | 0.15 | — |
| BACOLI/LE | 0.08 | 0.08 | 0.07 | 0.07 | 0.09 | 0.11 | 0.13 | 0.15 |
| $tol = 10^{-8}/p =$ | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| BACOL/ST | 0.41 | 0.35 | 0.41 | 0.45 | 0.50 | 0.61 | — | 0.92 |
| BACOL/LE | 0.66 | 0.41 | 0.35 | 0.41 | 0.45 | 0.50 | 0.61 | — |
| BACOLI/ST | 0.21 | 0.19 | 0.21 | 0.21 | 0.24 | 0.29 | 0.37 | 0.39 |
| BACOLI/LE | 0.49 | 0.31 | 0.23 | 0.25 | 0.25 | 0.32 | 0.33 | — |
| $tol = 10^{-10}/p =$ | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| BACOL/ST | 1.77 | 1.47 | 1.21 | 1.26 | — | — | 1.50 | 1.80 |
| BACOL/LE | 4.58 | 1.77 | 1.47 | 1.21 | 1.26 | — | — | 1.50 |
| BACOLI/ST | 0.99 | 0.79 | 0.63 | 0.70 | 0.75 | — | 0.77 | 0.90 |
| BACOLI/LE | 3.14 | 1.14 | 0.78 | 0.70 | 0.67 | 0.74 | 0.80 | — |

Table 18: *Machine dependent timings (in seconds), Catalytic Surface Reaction Model,* $p = 4, \ldots, 11$*,* $tol = 10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}$*.*

| | Timing Ratio Averages for Nine Test Problems | | | |
|---|---|---|---|---|
| Problem | $\frac{\text{BACOL/LE}}{\text{BACOL/ST}}$ | $\frac{\text{BACOLI/ST}}{\text{BACOL/ST}}$ | $\frac{\text{BACOLI/LE}}{\text{BACOL/ST}}$ | $\frac{\text{BACOLI/LE}}{\text{BACOLI/ST}}$ |
| OLBE, $\epsilon = 10^{-3}$ | 0.98 | 0.57 | 0.65 | 1.13 |
| OLBE, $\epsilon = 10^{-4}$ | 0.97 | 0.57 | 0.62 | 1.08 |
| TLBE, $\epsilon = 10^{-3}$ | 1.00 | 0.60 | 0.69 | 1.13 |
| TLBE, $\epsilon = 10^{-4}$ | 0.96 | 0.56 | 0.66 | 1.17 |
| CSRM | 0.99 | 0.49 | 0.63 | 1.30 |
| TLBE×6, $\epsilon = 10^{-3}$ | 0.98 | 0.53 | 0.63 | 1.14 |
| TLBE×6, $\epsilon = 10^{-4}$ | 0.91 | 0.49 | 0.57 | 1.14 |
| TLBE×12, $\epsilon = 10^{-3}$ | 0.90 | 0.41 | 0.54 | 1.26 |
| TLBE×12, $\epsilon = 10^{-4}$ | 0.83 | 0.38 | 0.44 | 1.18 |

Table 19: *Averages of ratios of timings for each problem (see Section 5.1) over* $p = 4, \ldots, 11$ *and* $tol = 10^{-4}, 10^{-6}, 10^{-8}, 10?^{-10}$*. Overall averages over all nine problems are* $\frac{BACOL/LE}{BACOL/ST} = 0.95$*,* $\frac{BACOLI/ST}{BACOL/ST} = 0.51$*,* $\frac{BACOLI/LE}{BACOL/ST} = 0.60$*, and* $\frac{BACOLI/LE}{BACOLI/ST} = 1.17$

| | Timing Ratio Averages for $p = 4, \ldots, 11$ | | | |
|---|---|---|---|---|
| $p$ | $\frac{\text{BACOLI/LE}}{\text{BACOL/ST}}$ | $\frac{\text{BACOLI/ST}}{\text{BACOL/ST}}$ | $\frac{\text{BACOLI/LE}}{\text{BACOL/ST}}$ | $\frac{\text{BACOLI/LE}}{\text{BACOLI/ST}}$ |
| 4 | 1.19 | 0.58 | 1.13 | 2.02 |
| 5 | 1.17 | 0.54 | 0.68 | 1.26 |
| 6 | 0.90 | 0.50 | 0.54 | 1.09 |
| 7 | 0.87 | 0.50 | 0.52 | 1.03 |
| 8 | 0.84 | 0.49 | 0.49 | 0.99 |
| 9 | 0.84 | 0.48 | 0.49 | 1.02 |
| 10 | 0.86 | 0.49 | 0.49 | 0.99 |
| 11 | 0.89 | 0.51 | 0.50 | 1.00 |

Table 20: *Averages of ratios of timings for $p = 4, \ldots, 11$ over all test problems (see Section 5.1) and tol $= 10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}$. Overall averages over all eight kcol values are $\frac{BACOL/LE}{BACOL/ST} = 0.95$, $\frac{BACOLI/ST}{BACOL/ST} = 0.51$, $\frac{BACOLI/LE}{BACOL/ST} = 0.60$, and $\frac{BACOLI/LE}{BACOLI/ST} = 1.17$*

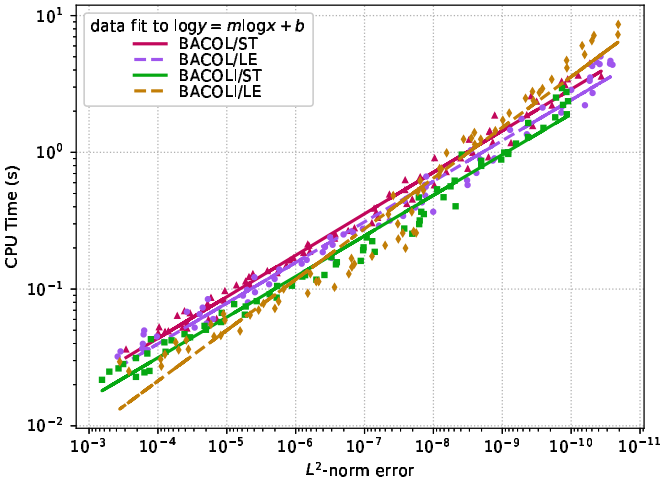


Figure 145: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 4$

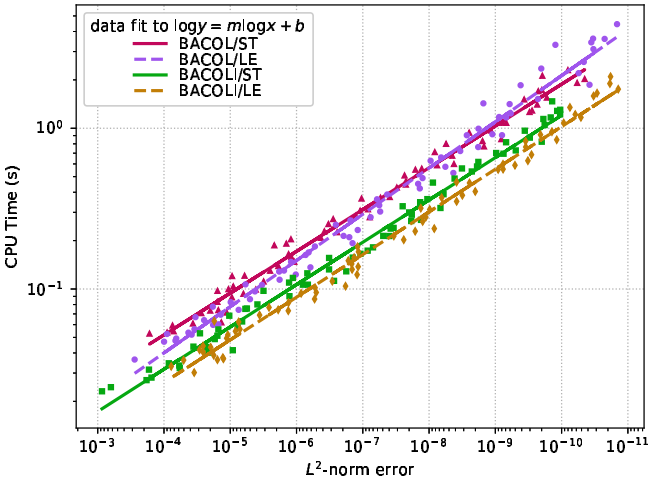


Figure 146: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 5$

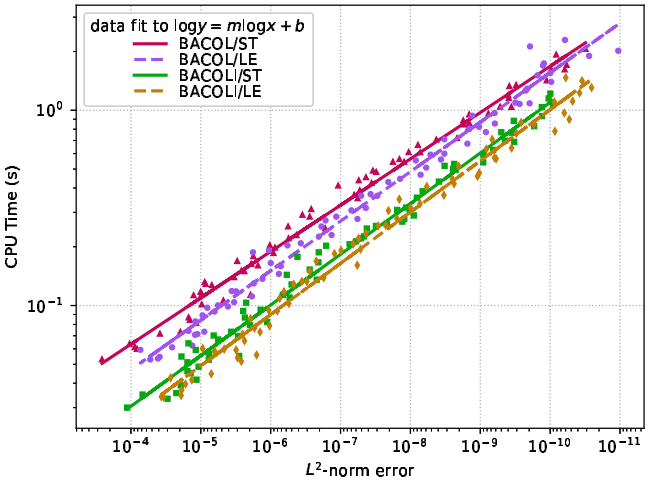


Figure 147: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 6$

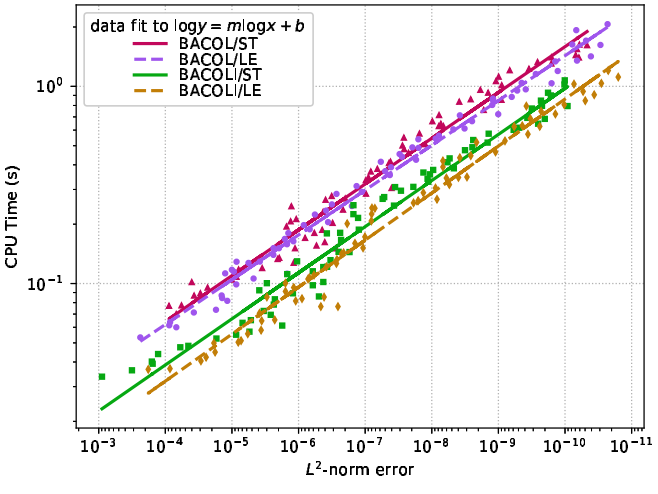


Figure 148: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 7$

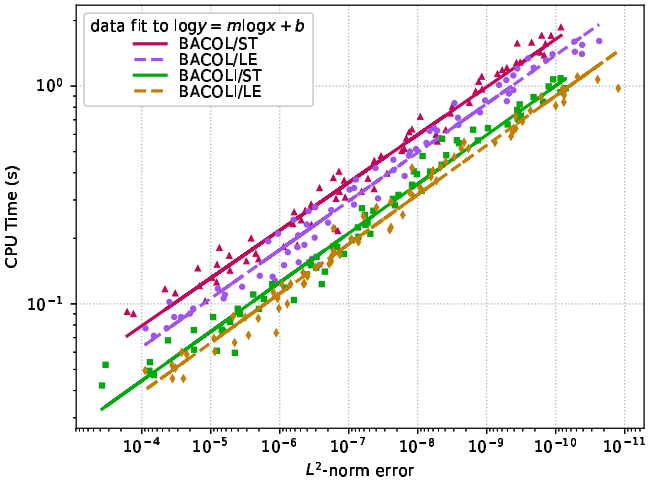


Figure 149: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 8$

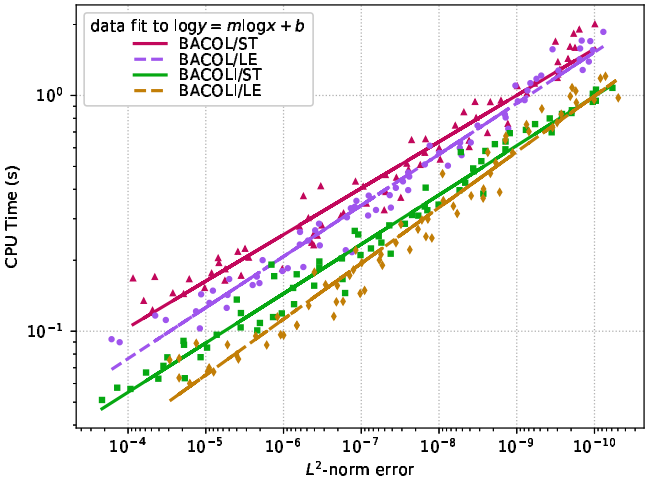


Figure 150: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 9$

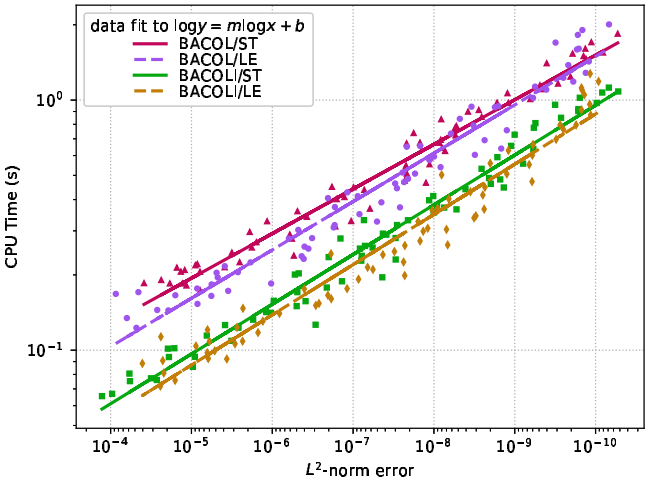


Figure 151: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 10$

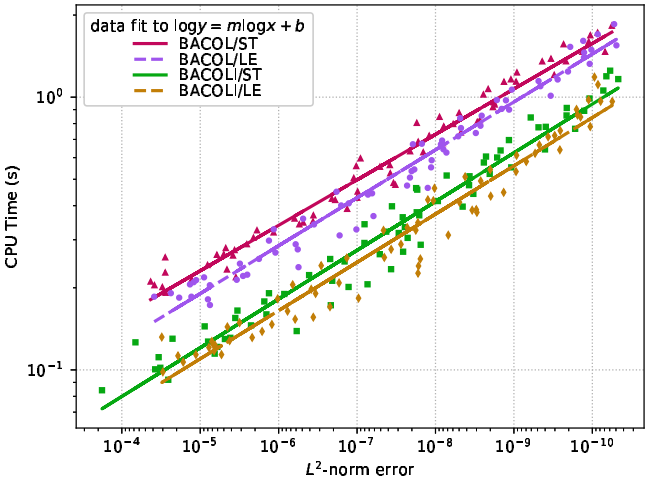


Figure 152: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-3}, p = 11$

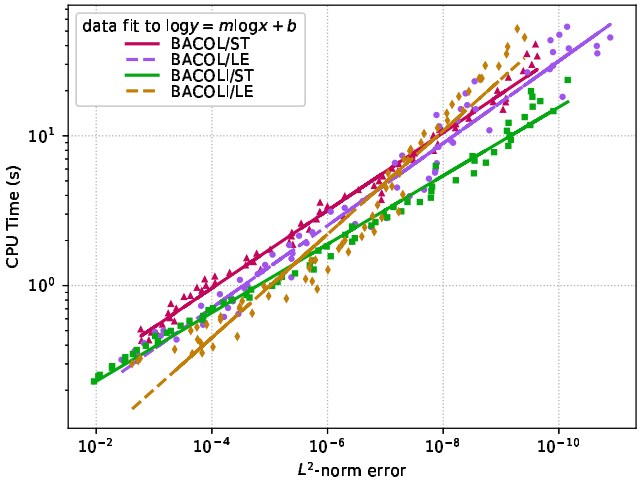


Figure 153: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 4$

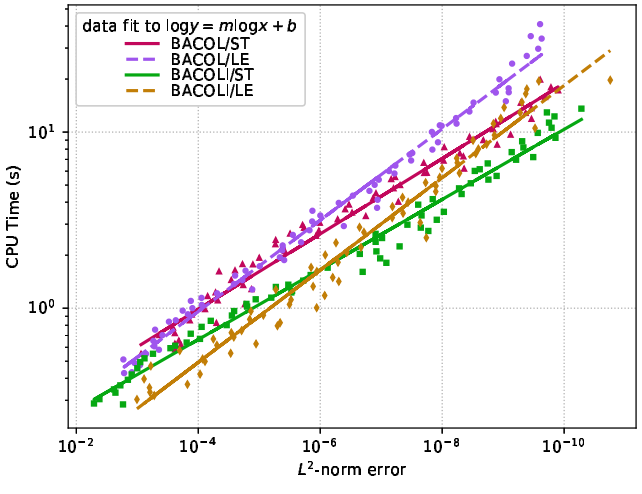


Figure 154: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 5$

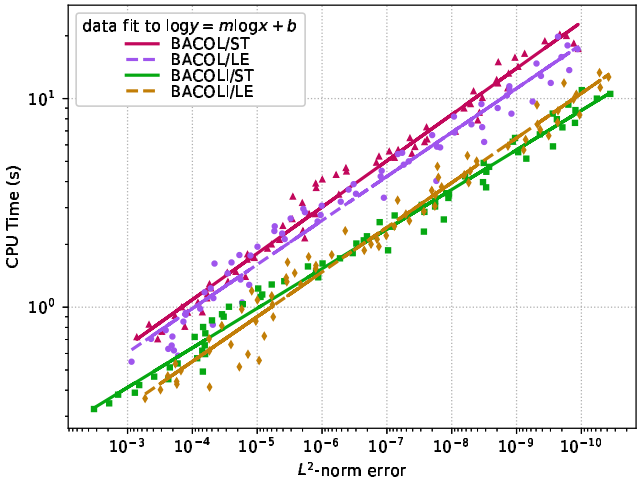


Figure 155: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 6$

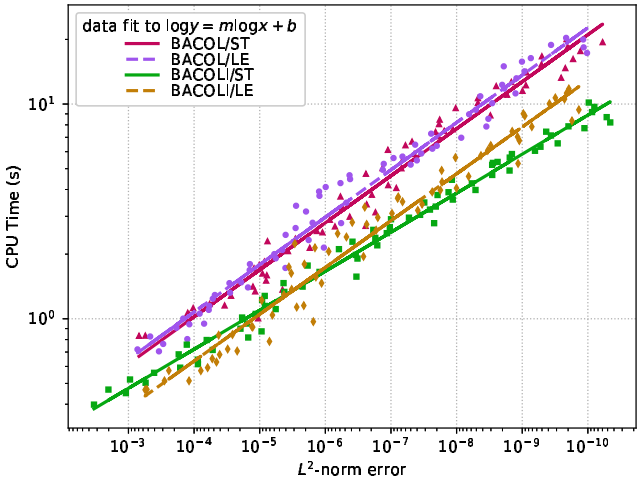


Figure 156: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 7$

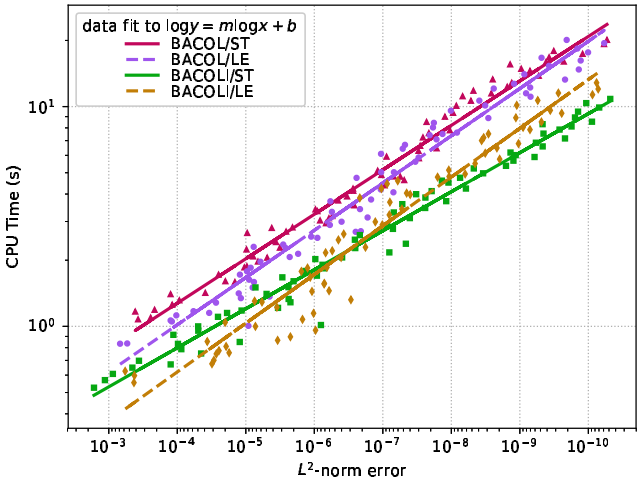


Figure 157: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 8$

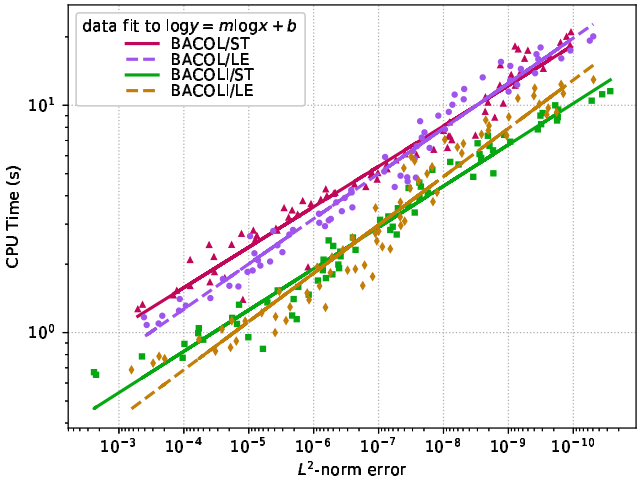


Figure 158: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 9$

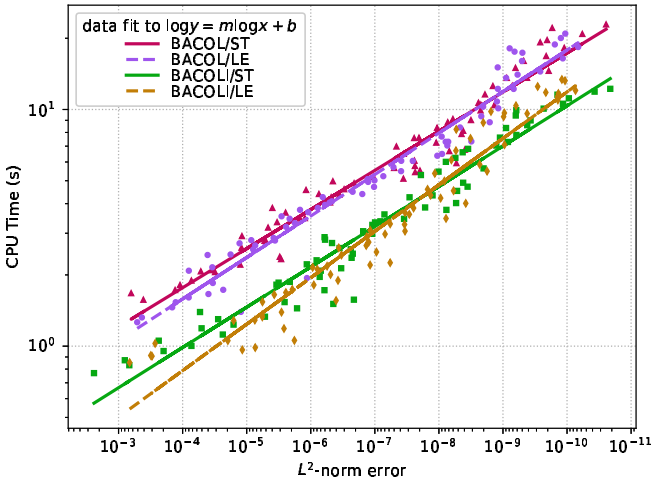


Figure 159: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 10$

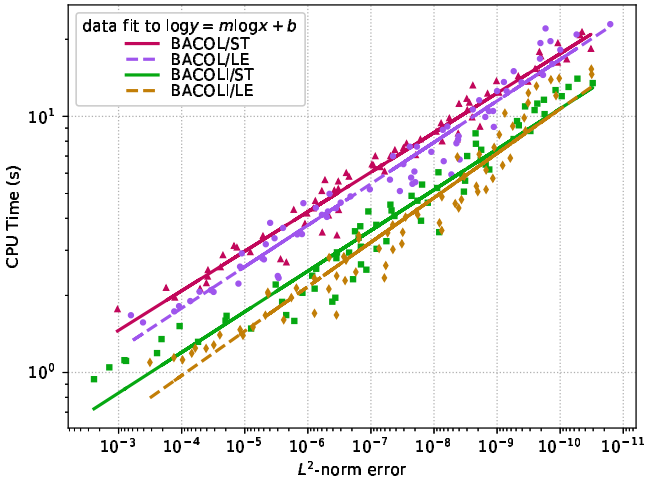


Figure 160: Work vs. Accuracy: One Layer Burgers equation, $\epsilon = 10^{-4}, p = 11$

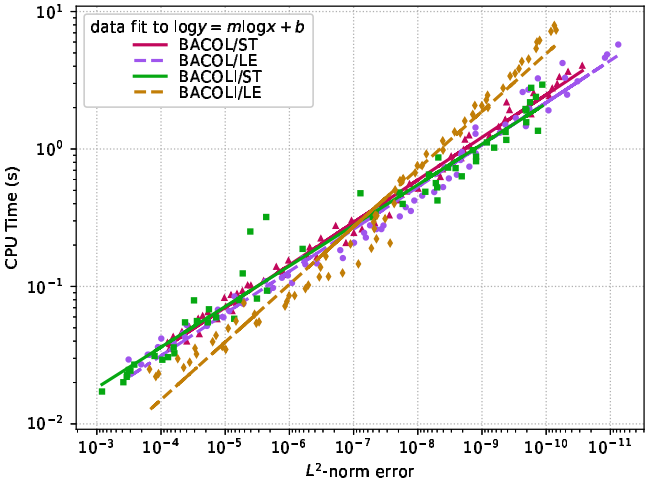


Figure 161: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 4$

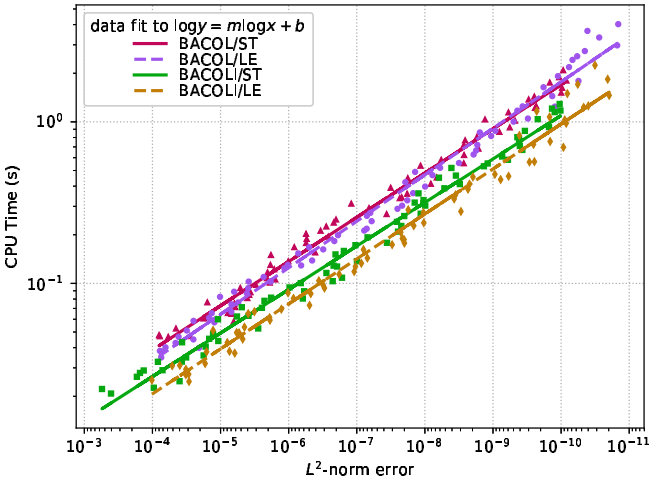


Figure 162: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 5$

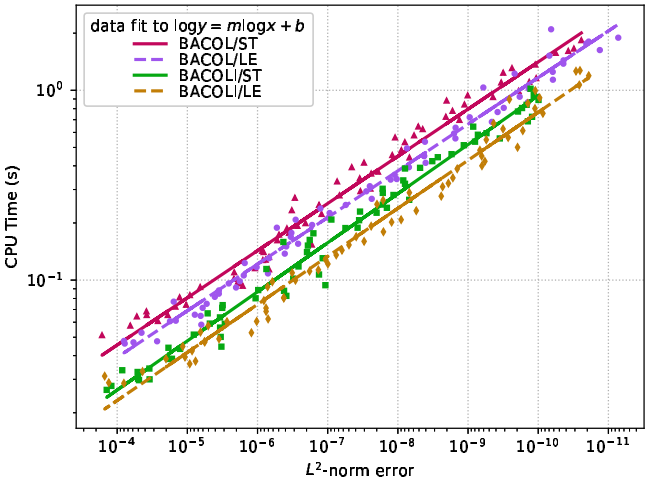


Figure 163: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 6$

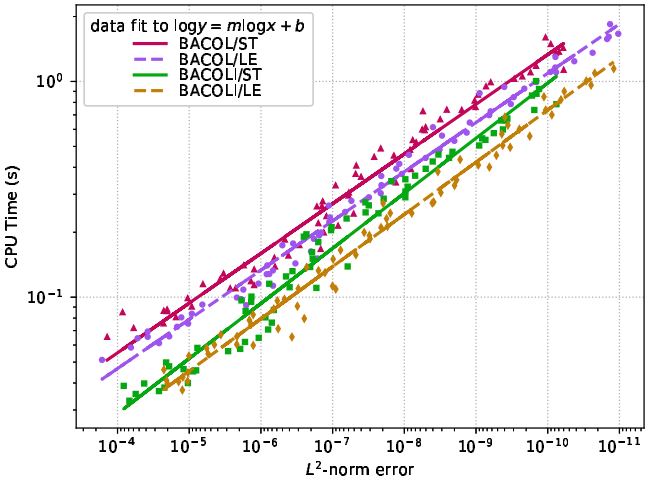


Figure 164: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 7$

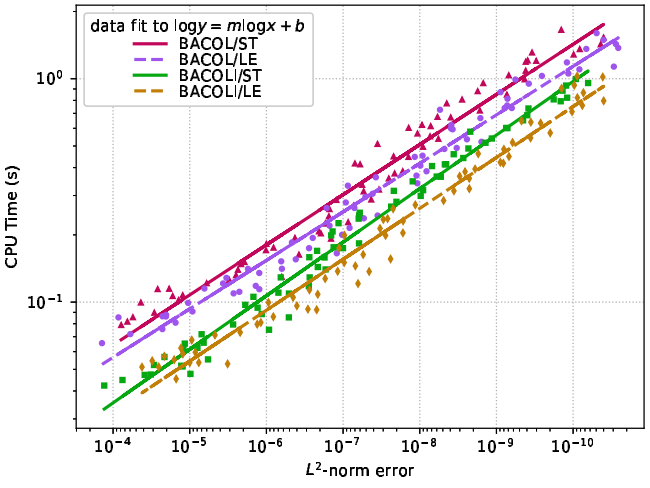


Figure 165: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 8$

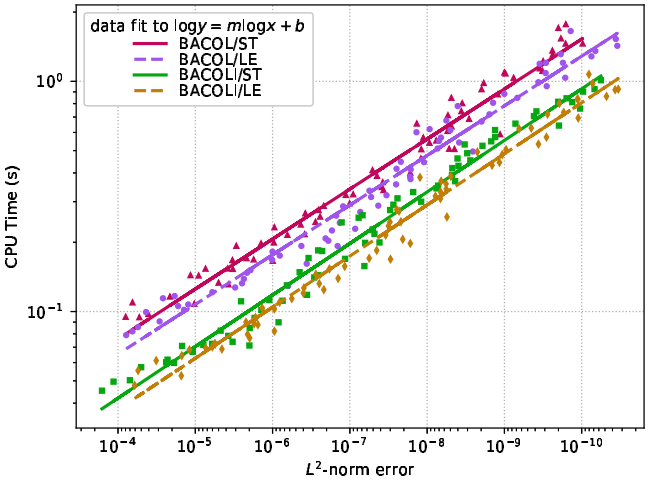


Figure 166: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 9$

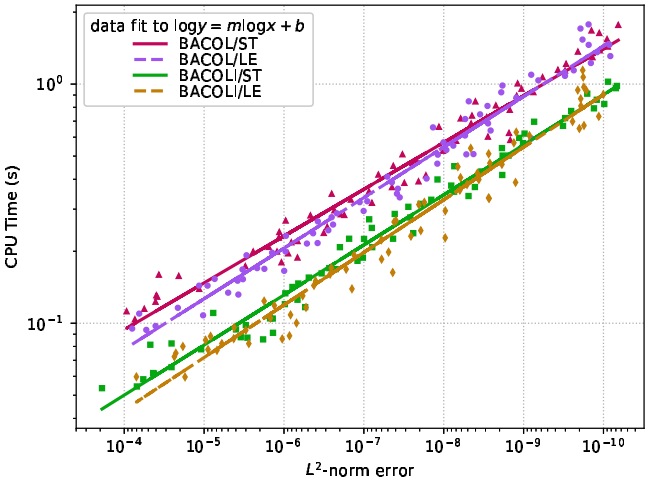


Figure 167: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 10$

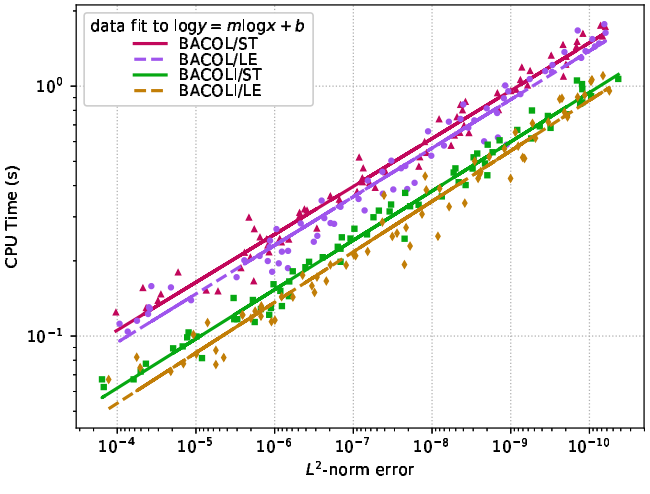


Figure 168: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-3}, p = 11$

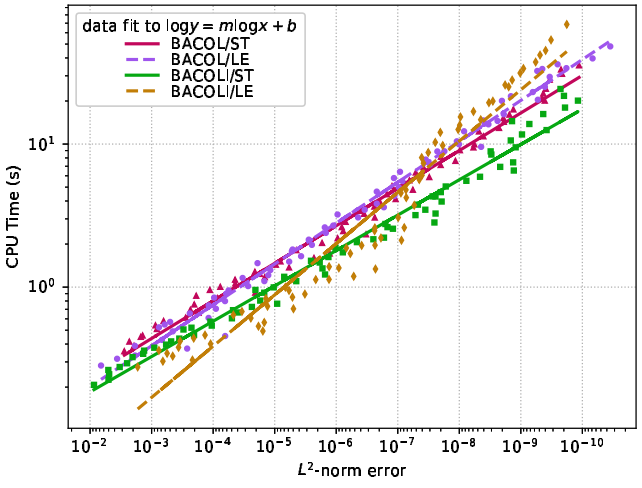


Figure 169: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 4$

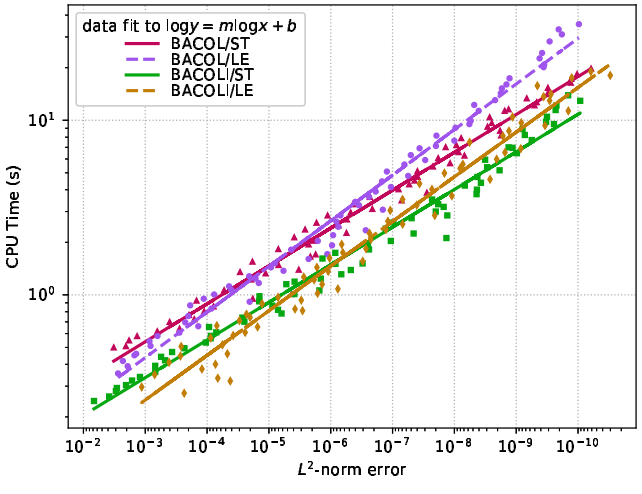


Figure 170: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 5$

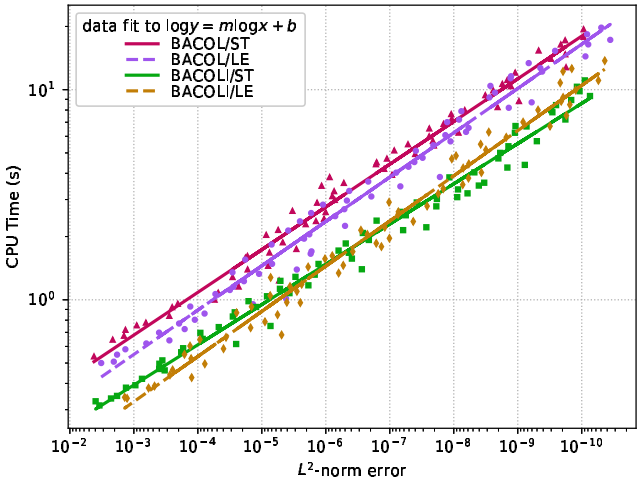


Figure 171: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 6$

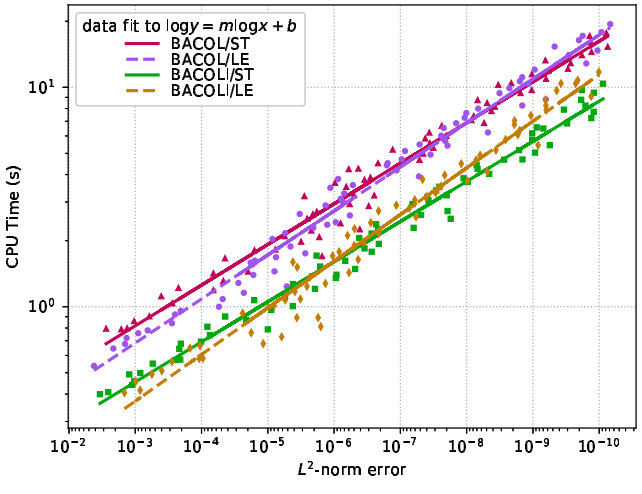


Figure 172: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 7$

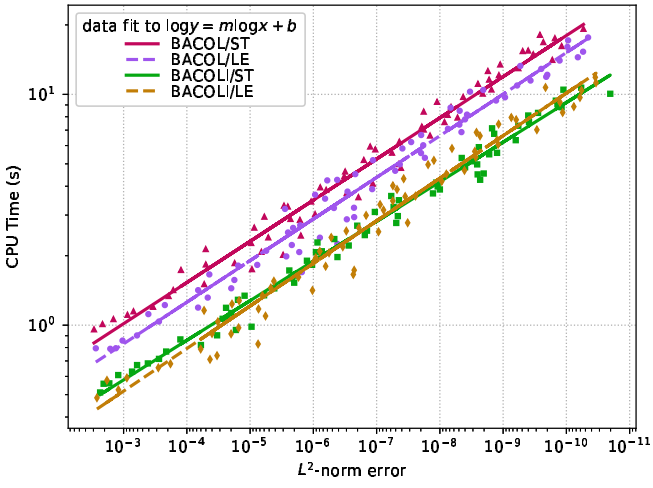


Figure 173: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 8$

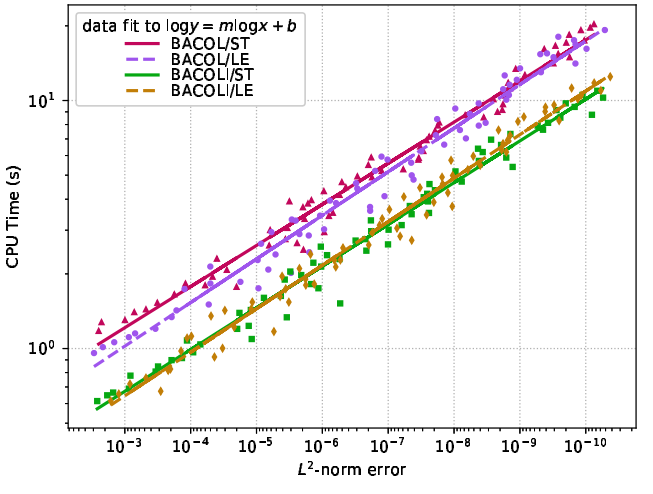


Figure 174: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 9$

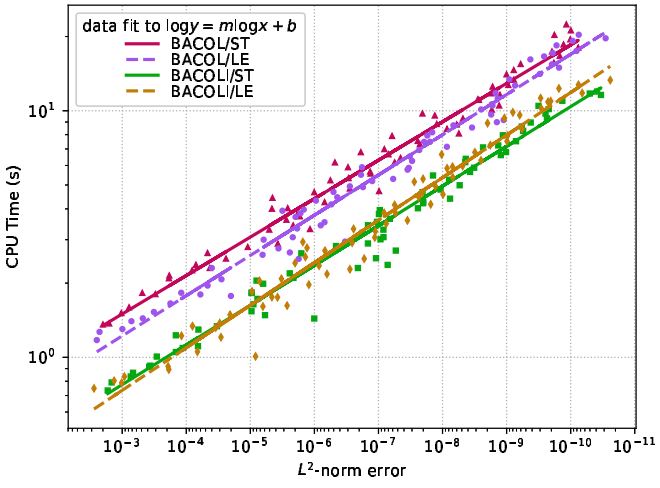


Figure 175: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 10$

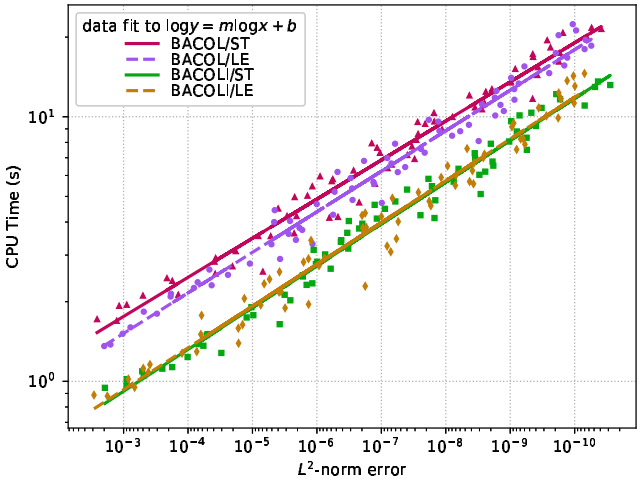


Figure 176: Work vs. Accuracy: Two Layer Burgers equation, $\epsilon = 10^{-4}, p = 11$

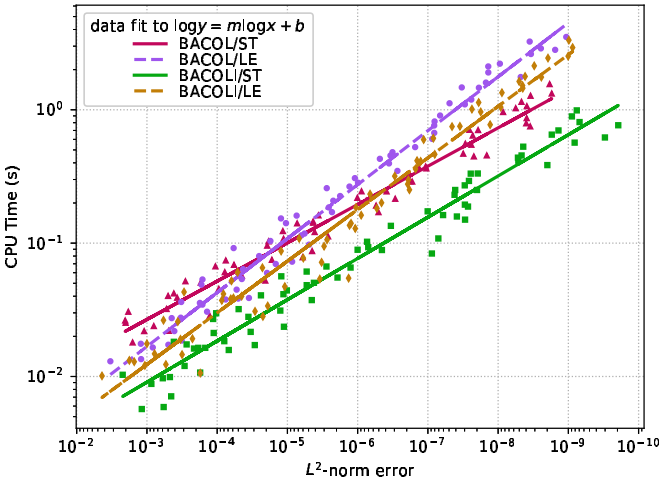


Figure 177: Work vs. Accuracy: Catalytic Surface Reaction Model, $p = 4$

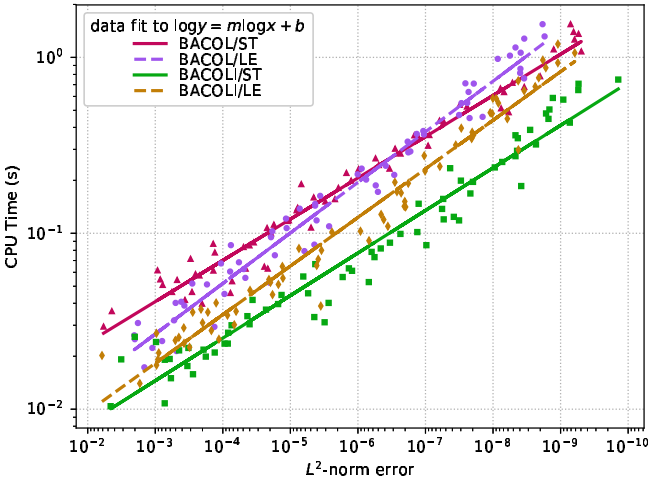


Figure 178: Work vs. Accuracy: Catalytic Surface Reaction Model, $p = 5$

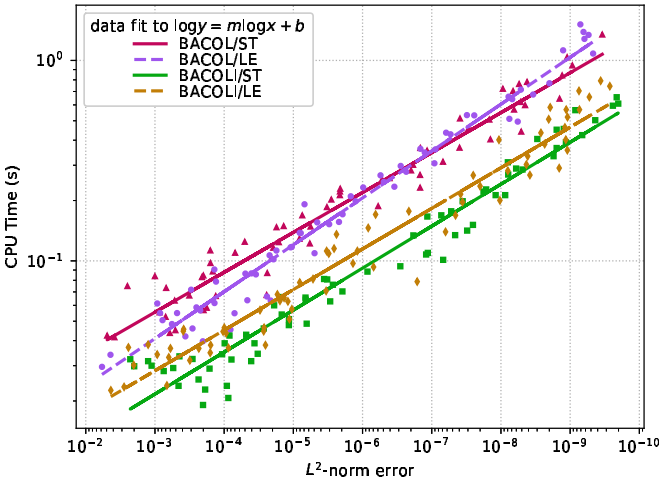


Figure 179: Work vs. Accuracy: Catalytic Surface Reaction Model, $p = 6$

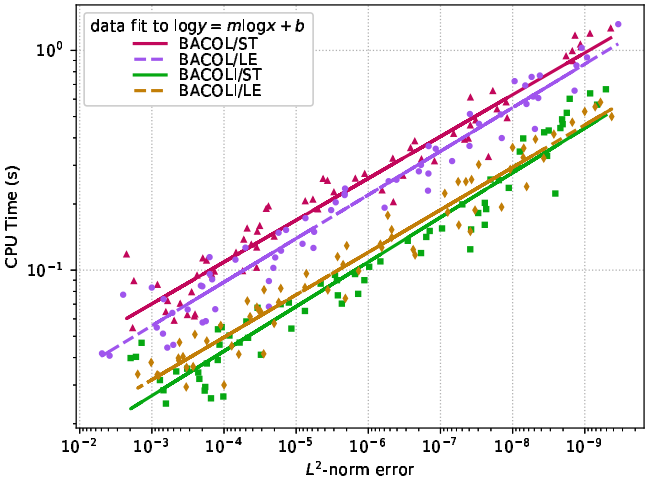


Figure 180: Work vs. Accuracy: Catalytic Surface Reaction Model, $p = 7$

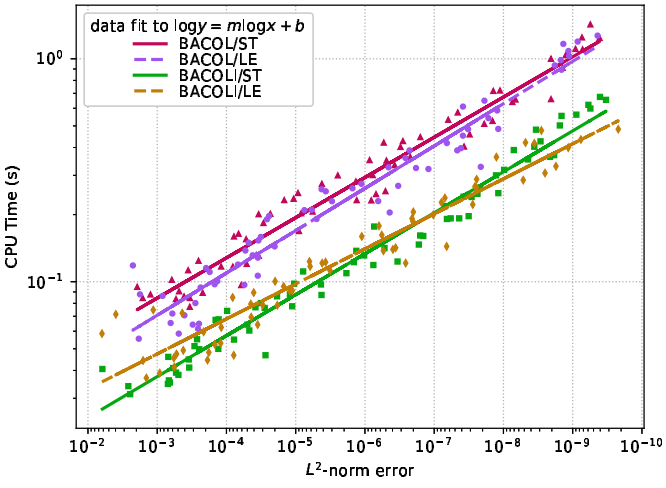


Figure 181: Work vs. Accuracy: Catalytic Surface Reaction Model, $p = 8$

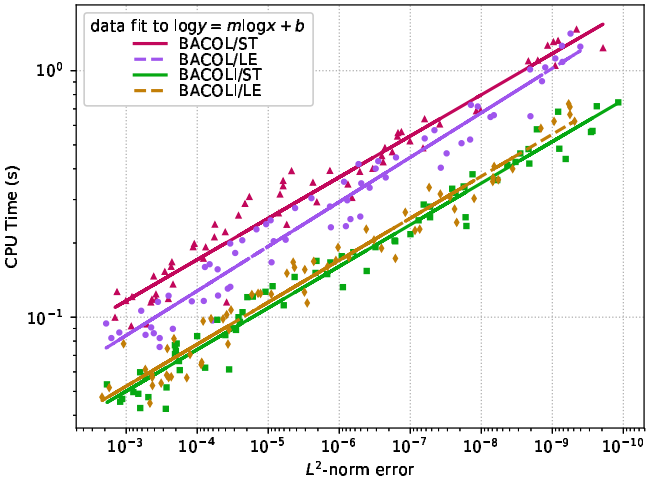


Figure 182: Work vs. Accuracy: Catalytic Surface Reaction Model, $p = 9$

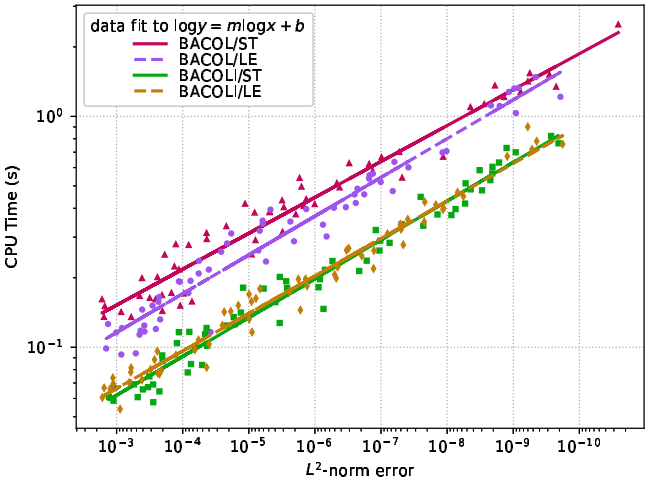


Figure 183: Work vs. Accuracy: Catalytic Surface Reaction Model, $p = 10$

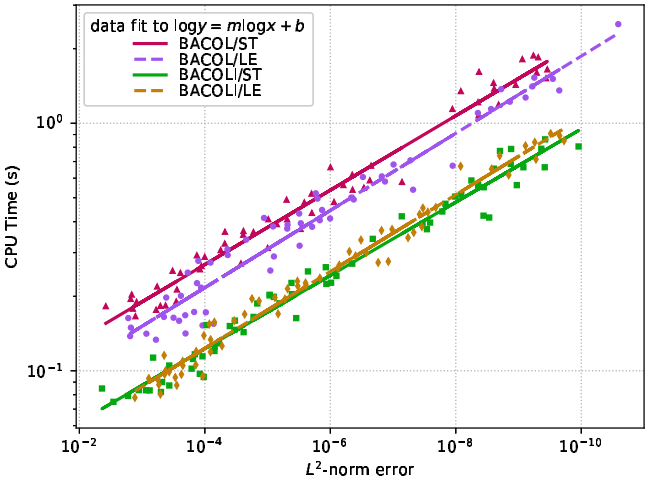


Figure 184: Work vs. Accuracy: Catalytic Surface Reaction Model, $p = 11$

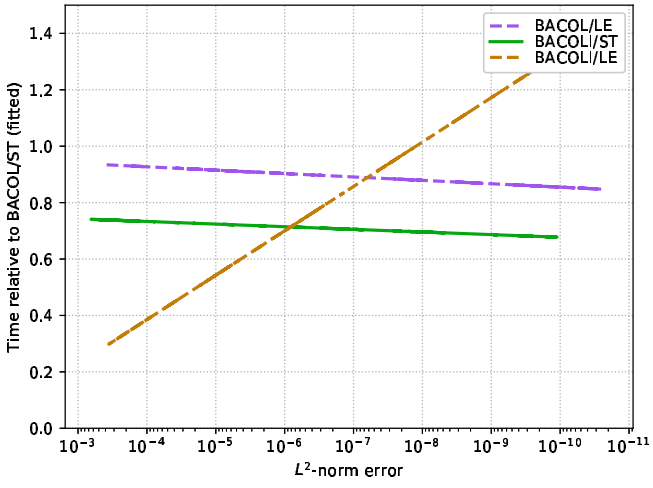


Figure 185: Rel. Work-Accuracy: One Layer Burgers eqn, $\epsilon = 10^{-3}, p = 4$

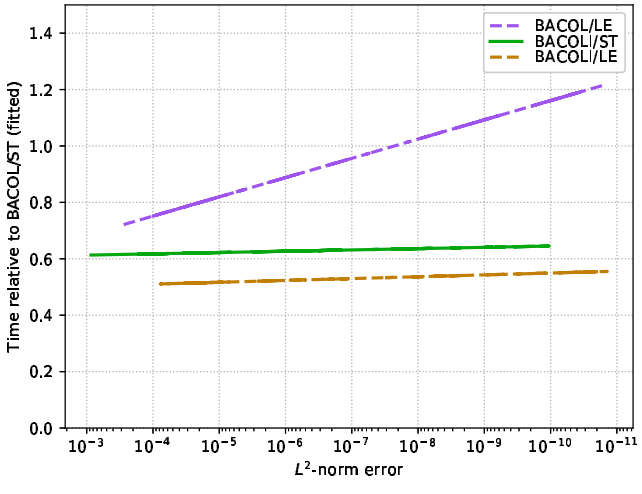


Figure 186: Rel. Work-Accuracy: One Layer Burgers eqn, $\epsilon = 10^{-3}, p = 5$

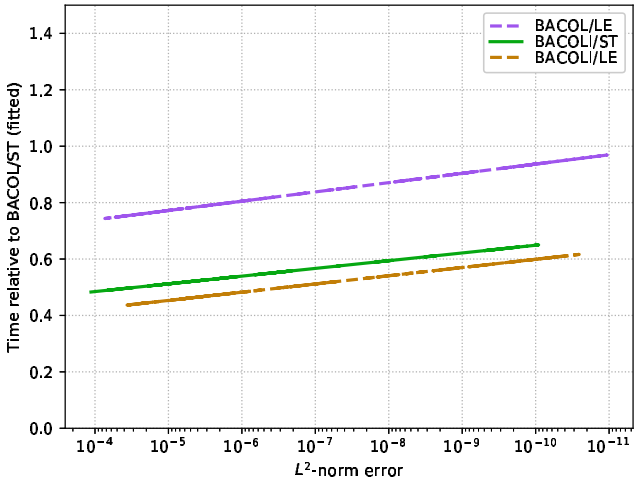


Figure 187: Rel. Work-Accuracy: One Layer Burgers eqn, $\epsilon = 10^{-3}, p = 6$

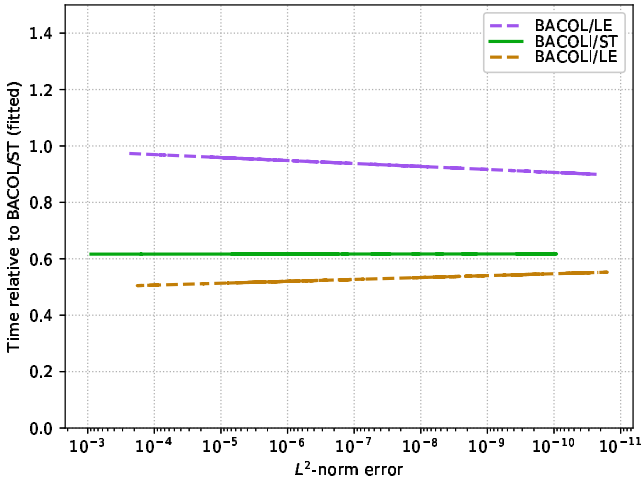


Figure 188: Rel. Work-Accuracy: One Layer Burgers eqn, $\epsilon = 10^{-3}, p = 7$

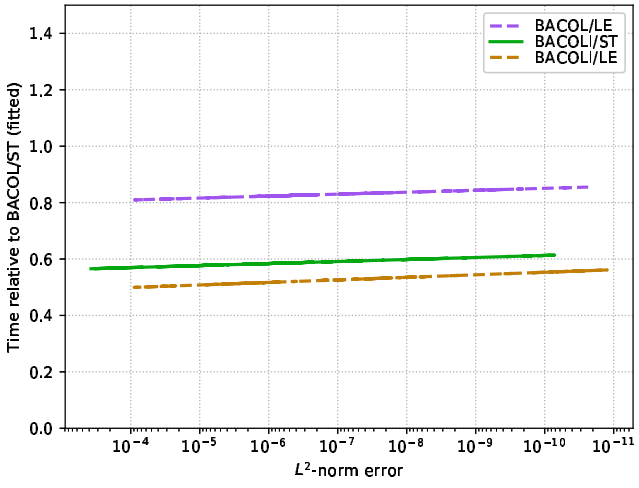


Figure 189: Rel. Work-Accuracy: One Layer Burgers eqn, $\epsilon = 10^{-3}, p = 8$

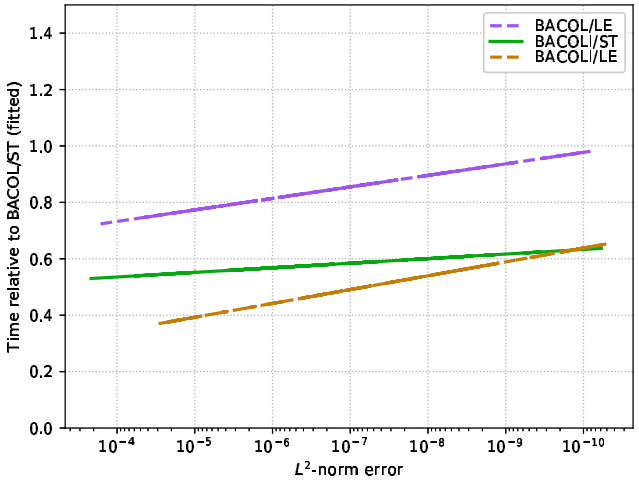


Figure 190: Rel. Work-Accuracy: One Layer Burgers eqn, $\epsilon = 10^{-3}, p = 9$

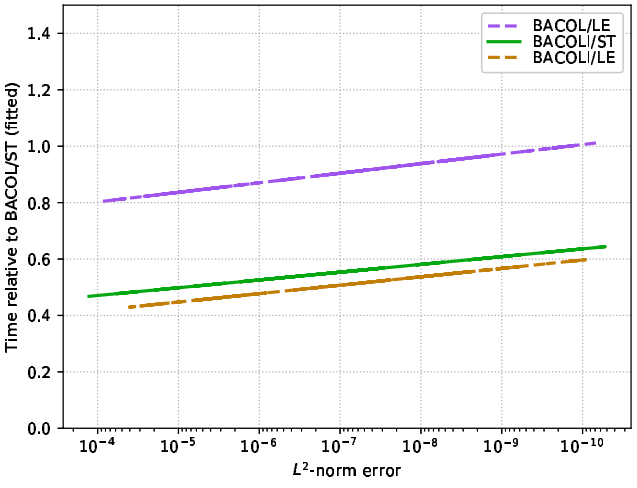


Figure 191: Rel. Work-Accuracy: One Layer Burgers eqn, $\epsilon = 10^{-3}, p = 10$

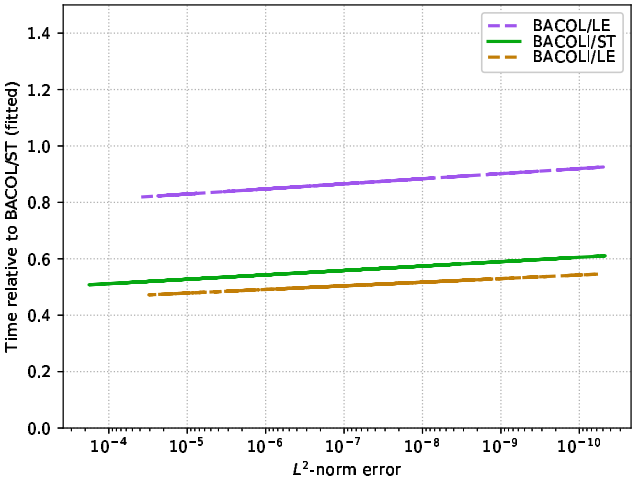


Figure 192: Rel. Work-Accuracy: One Layer Burgers eqn, $\epsilon = 10^{-3}, p = 11$

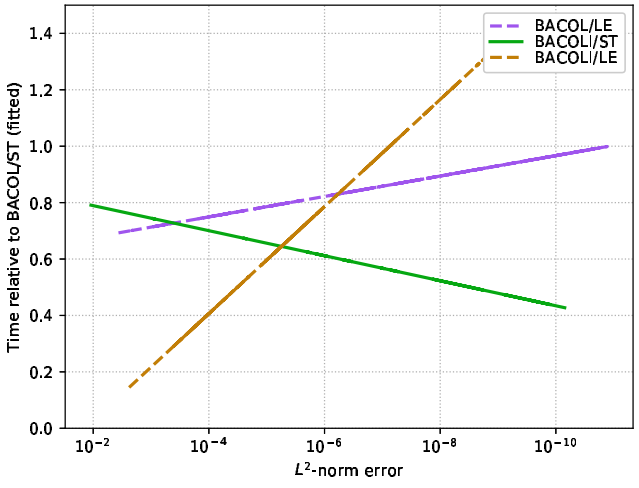


Figure 193: Rel. Work-Accuracy: One Layer Burgers eqn, $\epsilon = 10^{-4}, p = 4$

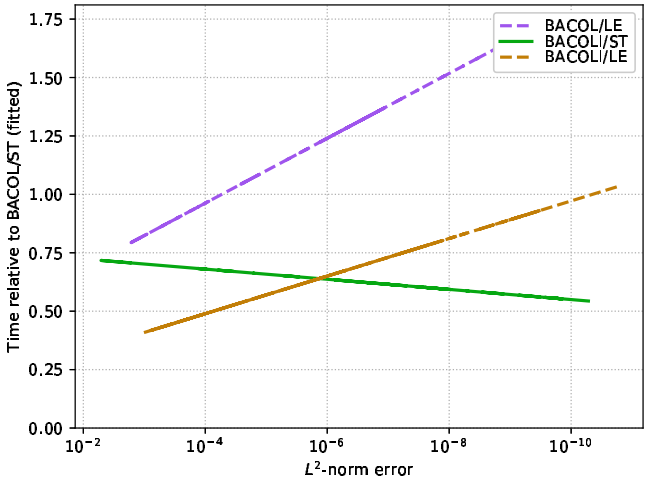


Figure 194: Rel. Work-Accuracy: One Layer Burgers eqn, $\epsilon = 10^{-4}, p = 5$

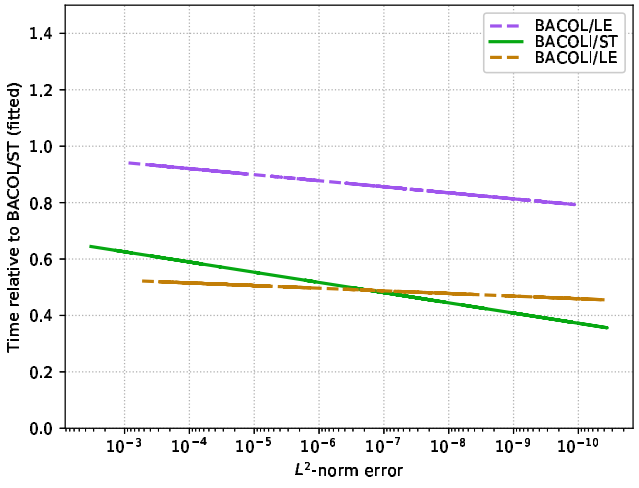


Figure 195: Rel. Work-Accuracy: One Layer Burgers eqn, $\epsilon = 10^{-4}, p = 6$

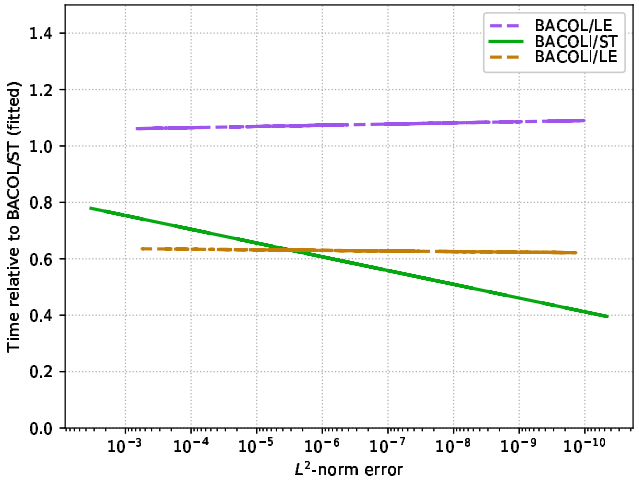


Figure 196: Rel. Work-Accuracy: One Layer Burgers eqn, $\epsilon = 10^{-4}, p = 7$

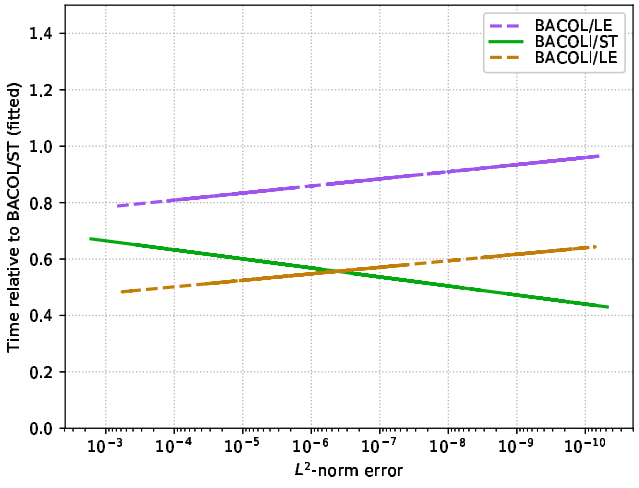


Figure 197: Rel. Work-Accuracy: One Layer Burgers eqn, $\epsilon = 10^{-4}, p = 8$

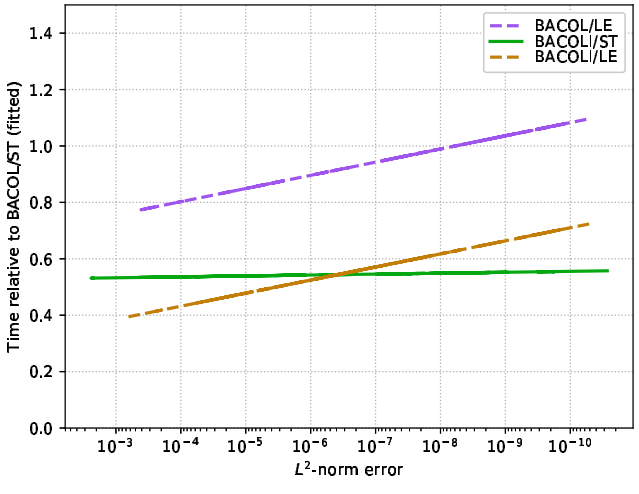


Figure 198: Rel. Work-Accuracy: One Layer Burgers eqn, $\epsilon = 10^{-4}, p = 9$

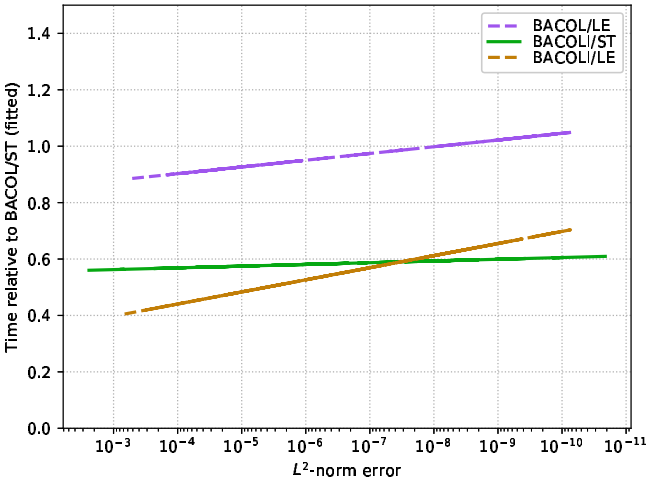


Figure 199: Rel. Work-Accuracy: One Layer Burgers eqn, $\epsilon = 10^{-4}, p = 10$

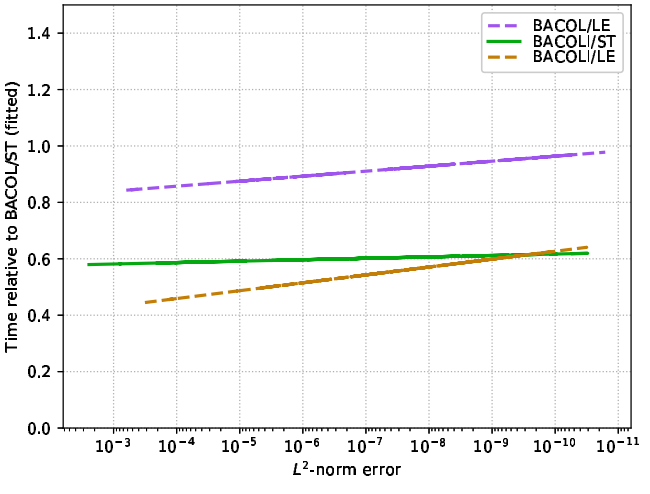


Figure 200: Rel. Work-Accuracy: One Layer Burgers eqn, $\epsilon = 10^{-4}, p = 11$

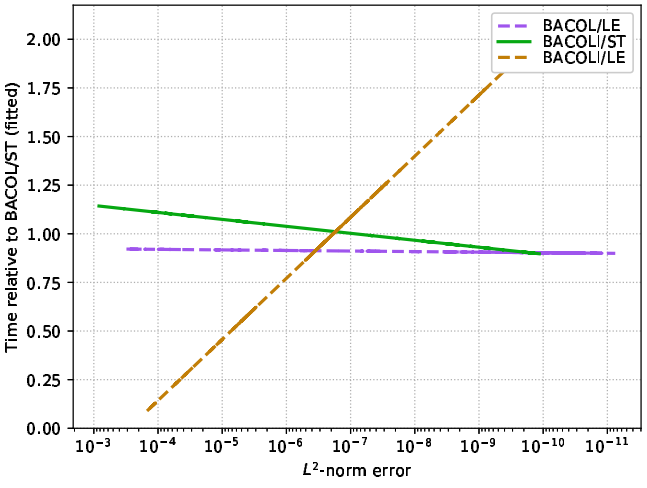


Figure 201: Rel. Work-Accuracy: Two Layer Burgers eqn, $\epsilon = 10^{-3}, p = 4$

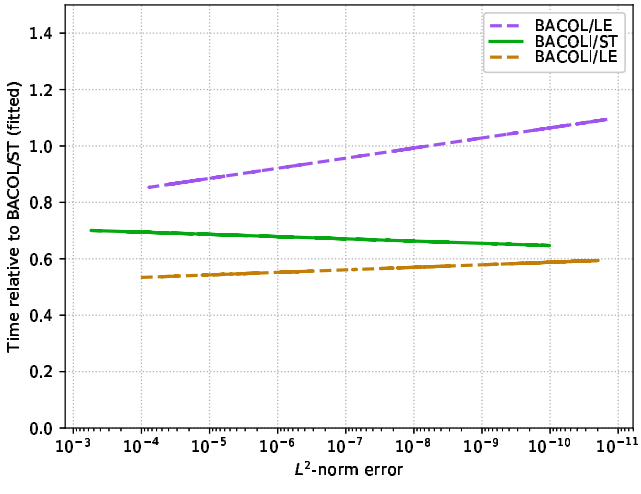


Figure 202: Rel. Work-Accuracy: Two Layer Burgers eqn, $\epsilon = 10^{-3}, p = 5$

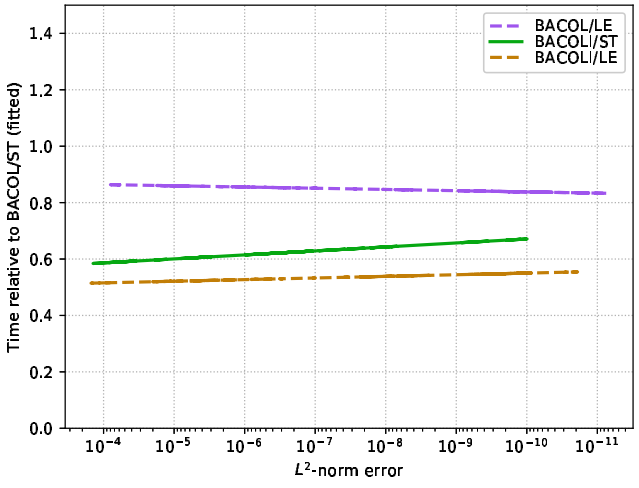


Figure 203: Rel. Work-Accuracy: Two Layer Burgers eqn, $\epsilon = 10^{-3}, p = 6$

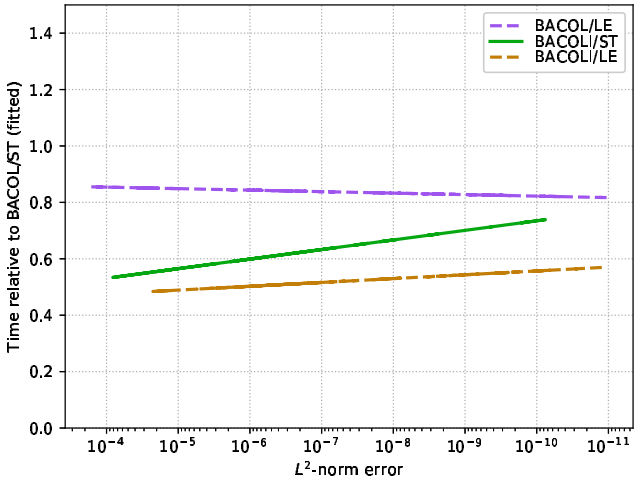


Figure 204: Rel. Work-Accuracy: Two Layer Burgers eqn, $\epsilon = 10^{-3}, p = 7$

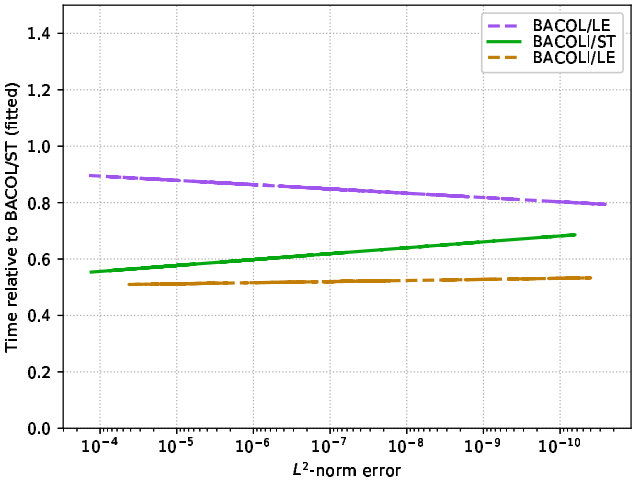


Figure 205: Rel. Work-Accuracy: Two Layer Burgers eqn, $\epsilon = 10^{-3}, p = 8$

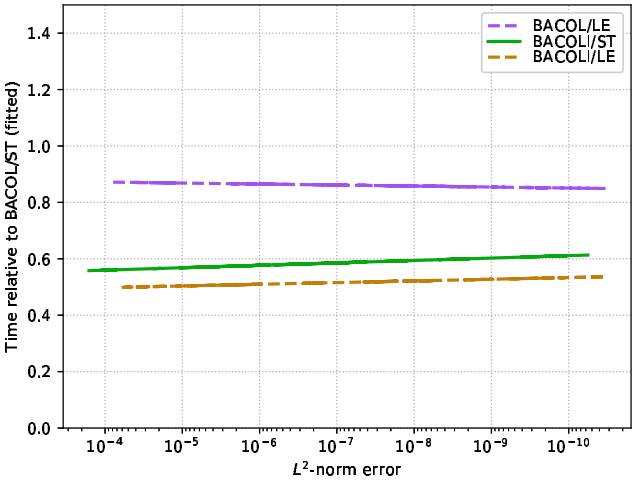


Figure 206: Rel. Work-Accuracy: Two Layer Burgers eqn, $\epsilon = 10^{-3}, p = 9$

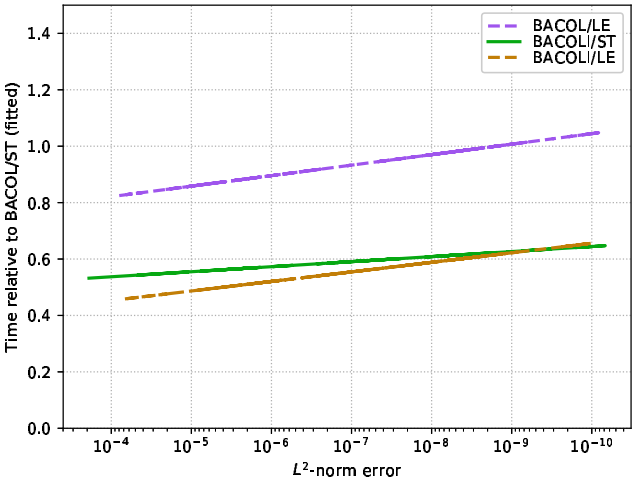


Figure 207: Rel. Work-Accuracy: Two Layer Burgers eqn, $\epsilon = 10^{-3}, p = 10$

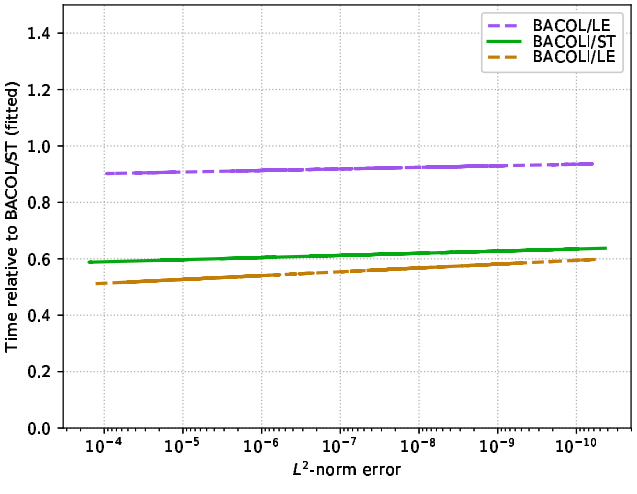


Figure 208: Rel. Work-Accuracy: Two Layer Burgers eqn, $\epsilon = 10^{-3}, p = 11$

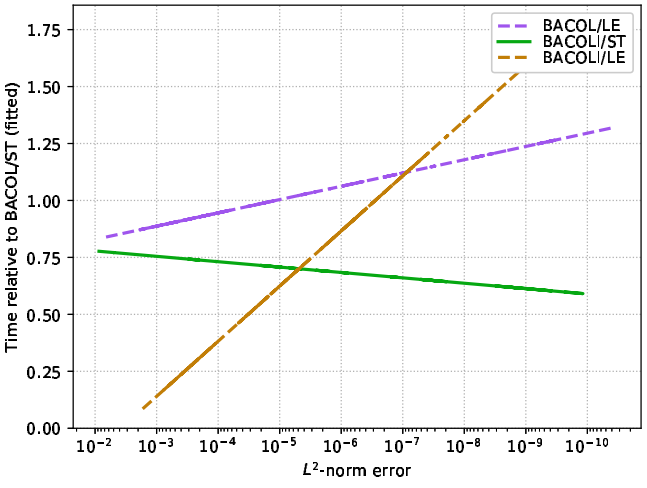


Figure 209: Rel. Work-Accuracy: Two Layer Burgers eqn, $\epsilon = 10^{-4}, p = 4$

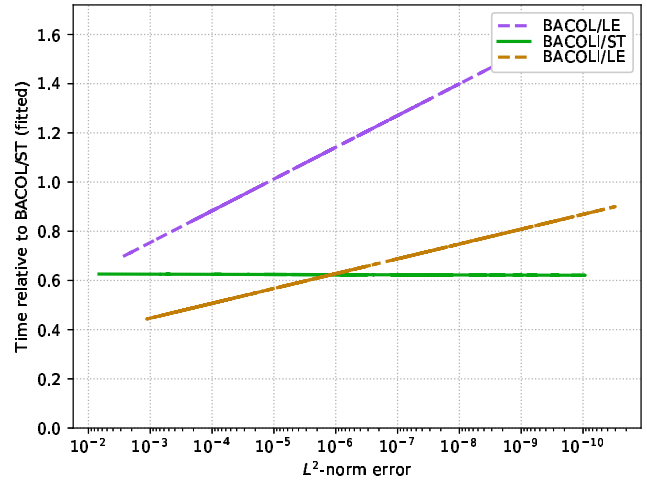


Figure 210: Rel. Work-Accuracy: Two Layer Burgers eqn, $\epsilon = 10^{-4}, p = 5$

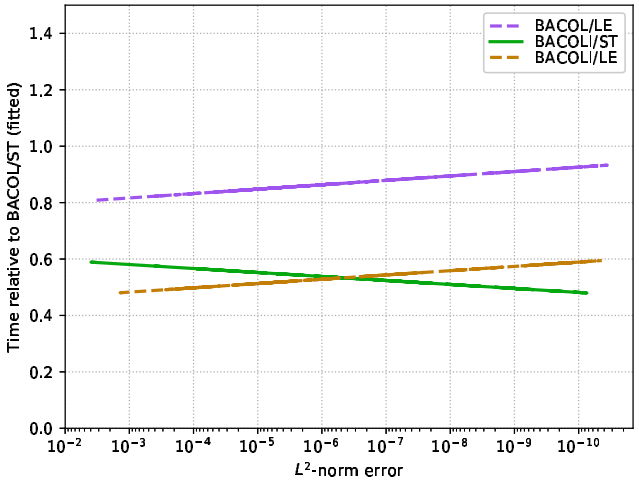


Figure 211: Rel. Work-Accuracy: Two Layer Burgers eqn, $\epsilon = 10^{-4}, p = 6$

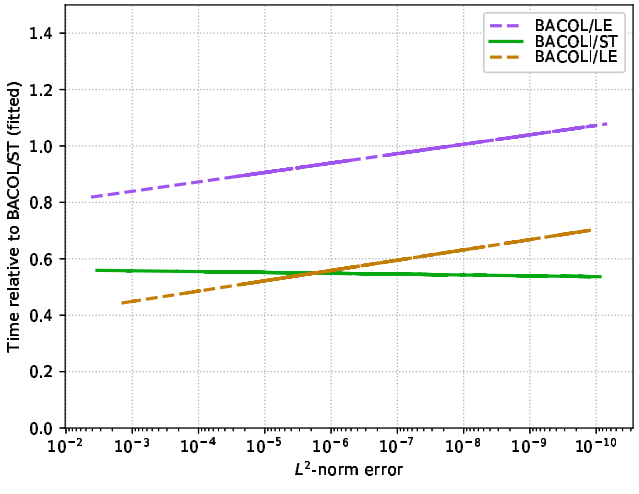


Figure 212: Rel. Work-Accuracy: Two Layer Burgers eqn, $\epsilon = 10^{-4}, p = 7$

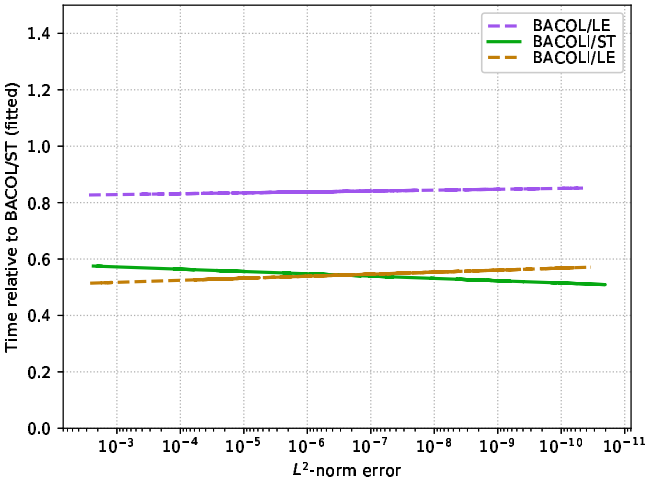


Figure 213: Rel. Work-Accuracy: Two Layer Burgers eqn, $\epsilon = 10^{-4}, p = 8$

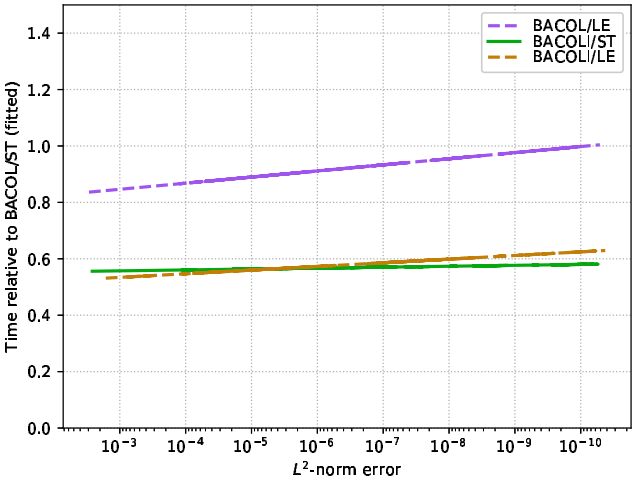


Figure 214: Rel. Work-Accuracy: Two Layer Burgers eqn, $\epsilon = 10^{-4}, p = 9$

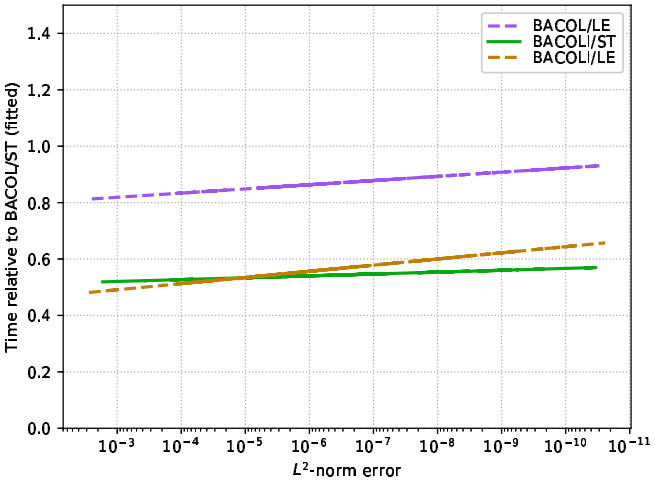


Figure 215: Rel. Work-Accuracy: Two Layer Burgers eqn, $\epsilon = 10^{-4}, p = 10$

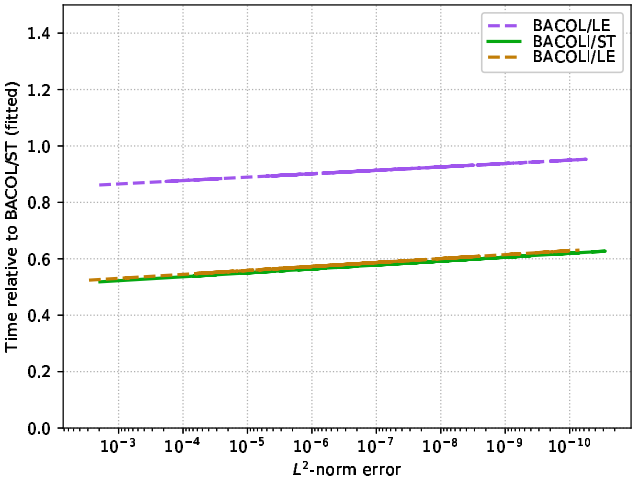


Figure 216: Rel. Work-Accuracy: Two Layer Burgers eqn, $\epsilon = 10^{-4}, p = 11$

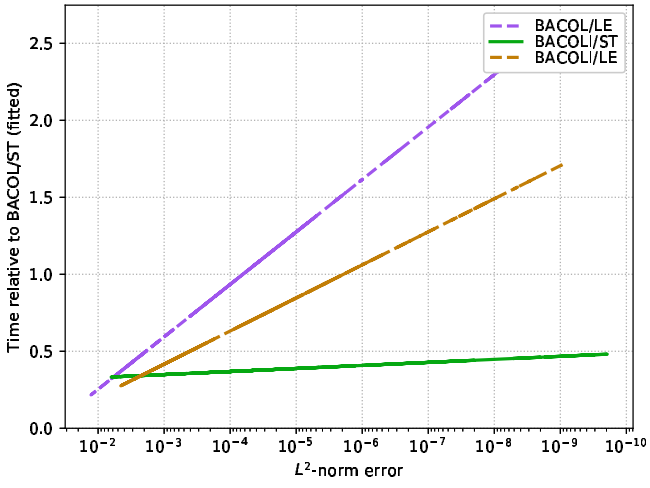


Figure 217: Rel. Work-Accuracy: Catalytic Surface Reaction Model $p = 4$

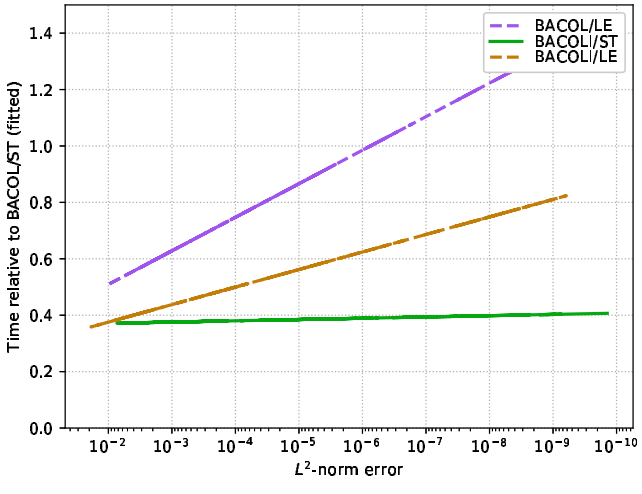


Figure 218: Rel. Work-Accuracy: Catalytic Surface Reaction Model $p = 5$

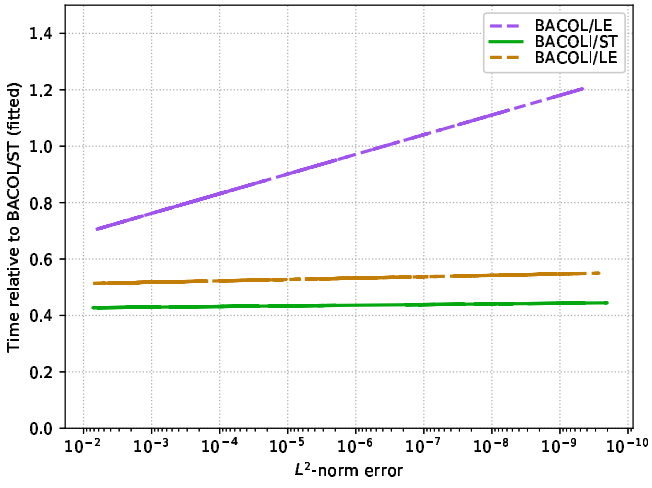


Figure 219: Rel. Work-Accuracy: Catalytic Surface Reaction Model $p = 6$

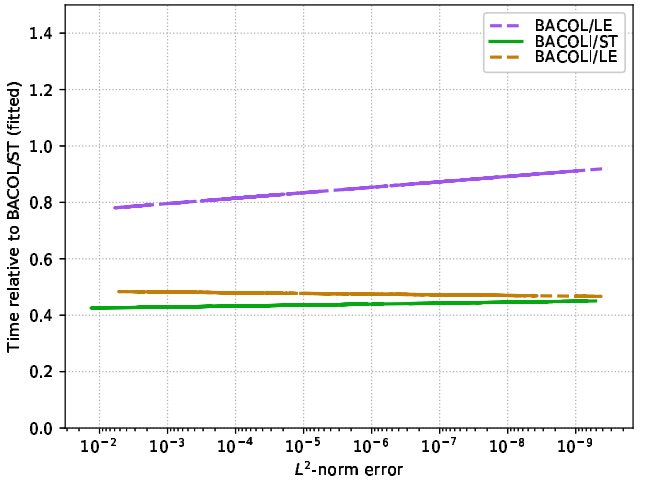


Figure 220: Rel. Work-Accuracy: Catalytic Surface Reaction Model $p = 7$

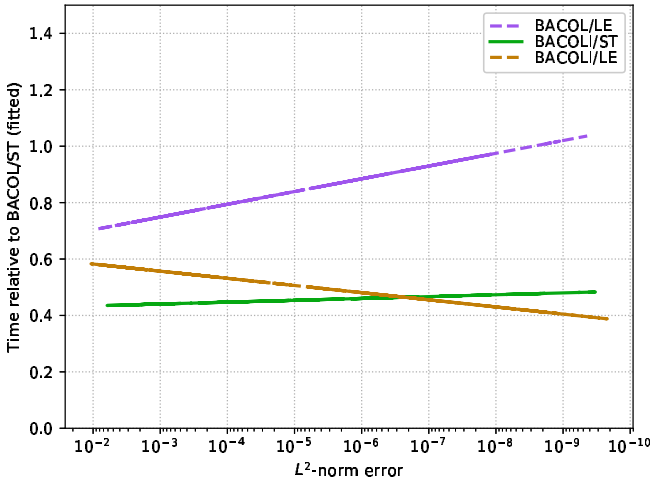


Figure 221: Rel. Work-Accuracy: Catalytic Surface Reaction Model $p = 8$

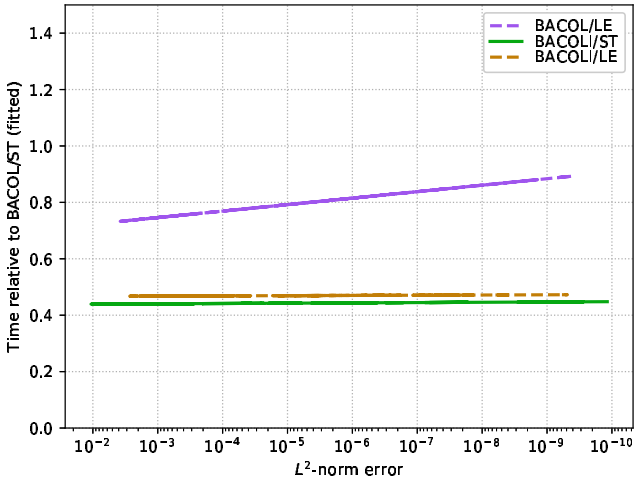


Figure 222: Rel. Work-Accuracy: Catalytic Surface Reaction Model $p = 9$

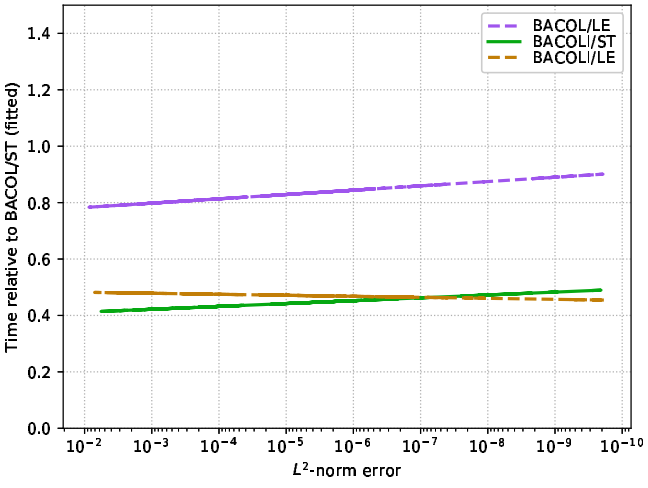


Figure 223: Rel. Work-Accuracy: Catalytic Surface Reaction Model $p = 10$

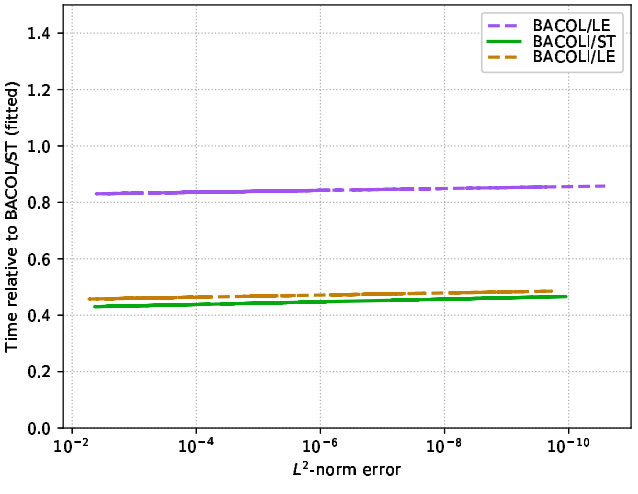


Figure 224: Rel. Work-Accuracy: Catalytic Surface Reaction Model $p = 11$

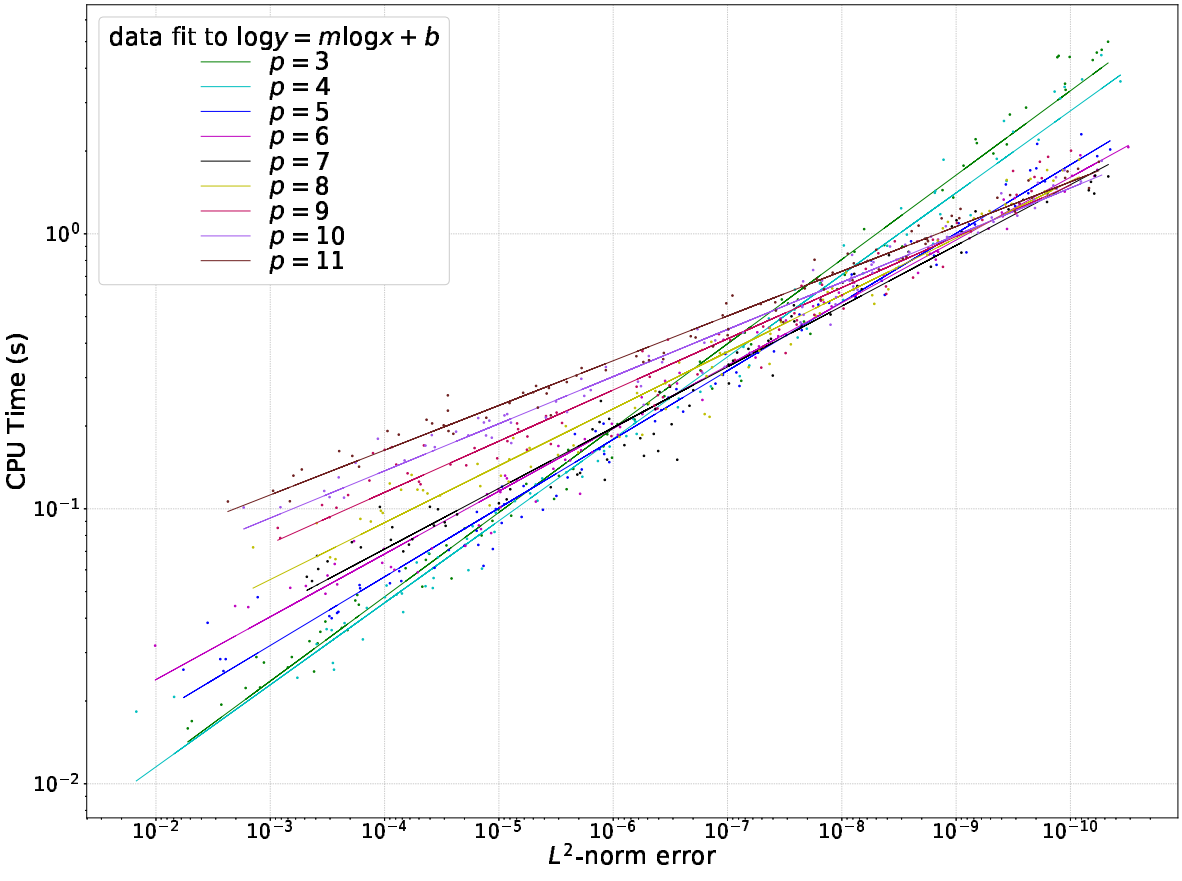


Figure 225: Work vs. Accuracy: One Layer Burgers equation $\epsilon = 10^{-3}$; BACOL/ST for $p = 3 \ldots 11$

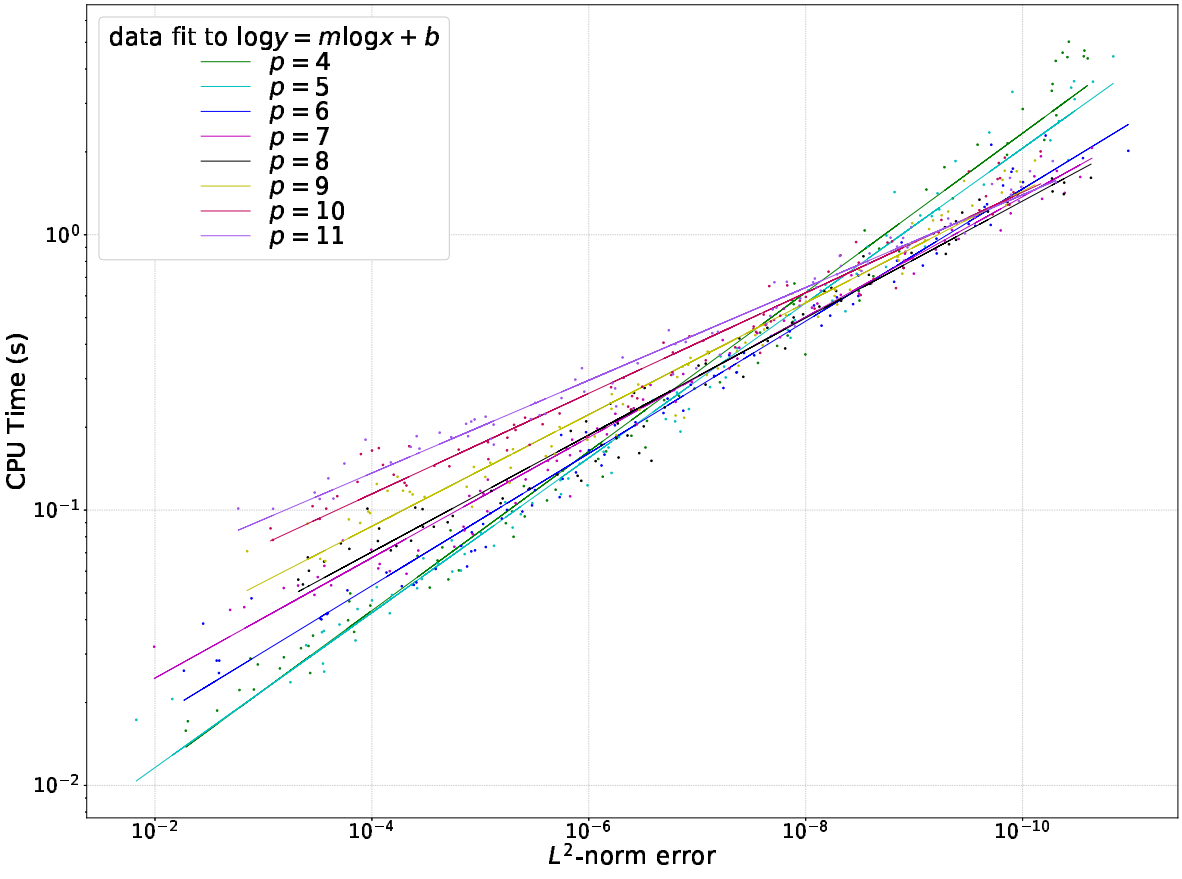


Figure 226: Work vs. Accuracy: One Layer Burgers equation $\epsilon = 10^{-3}$; BACOL/LE for $p = 4 \ldots 11$

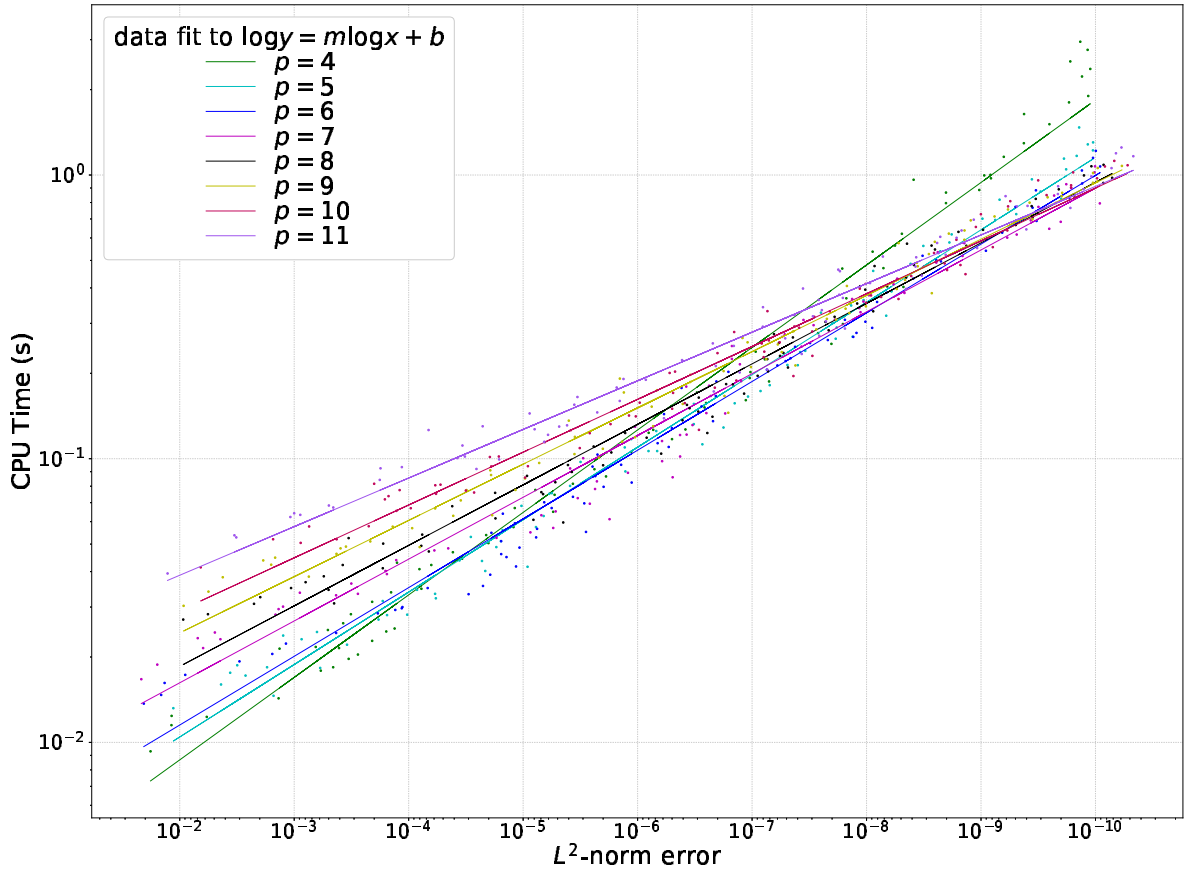


Figure 227: Work vs. Accuracy: One Layer Burgers equation $\epsilon = 10^{-3}$; BACOLI/ST for $p = 4 \dots 11$

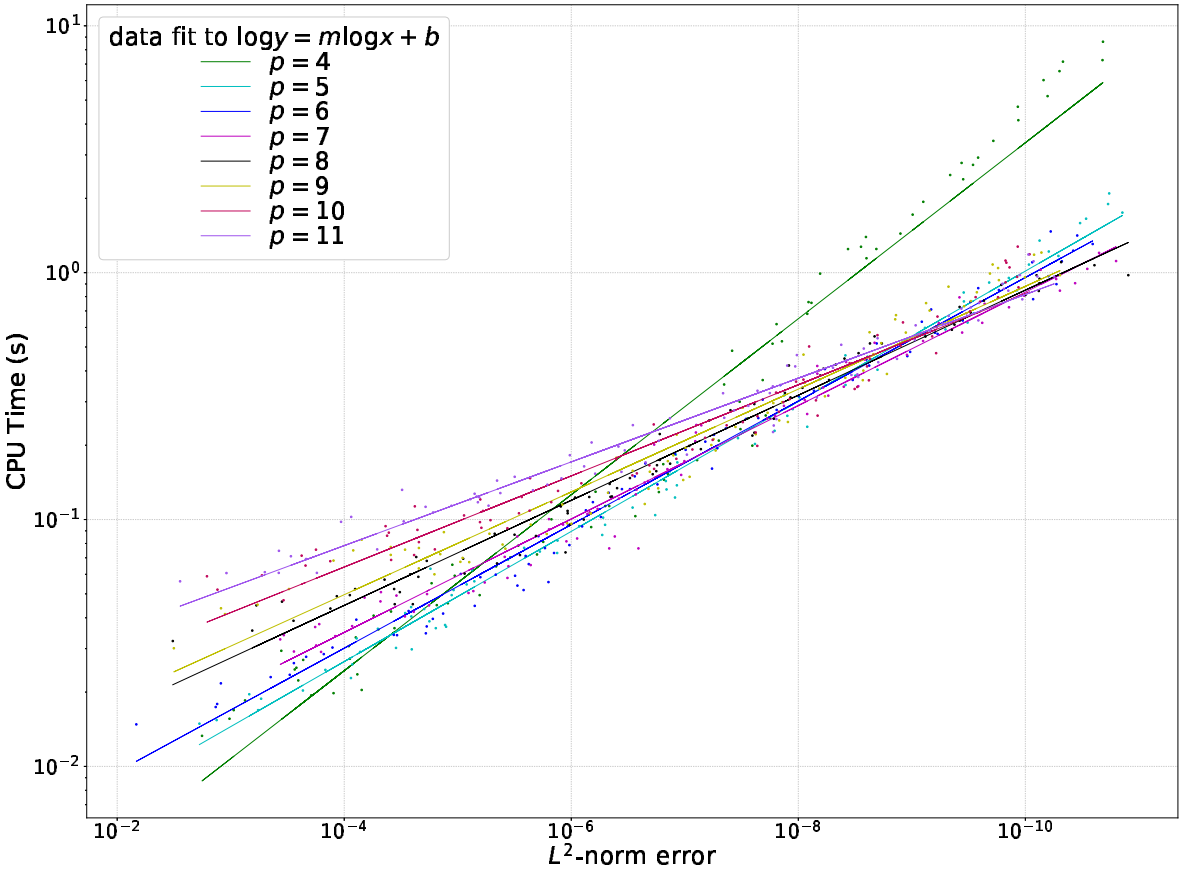


Figure 228: Work vs. Accuracy: One Layer Burgers equation $\epsilon = 10^{-3}$; BACOLI/LE for $p = 4 \ldots 11$

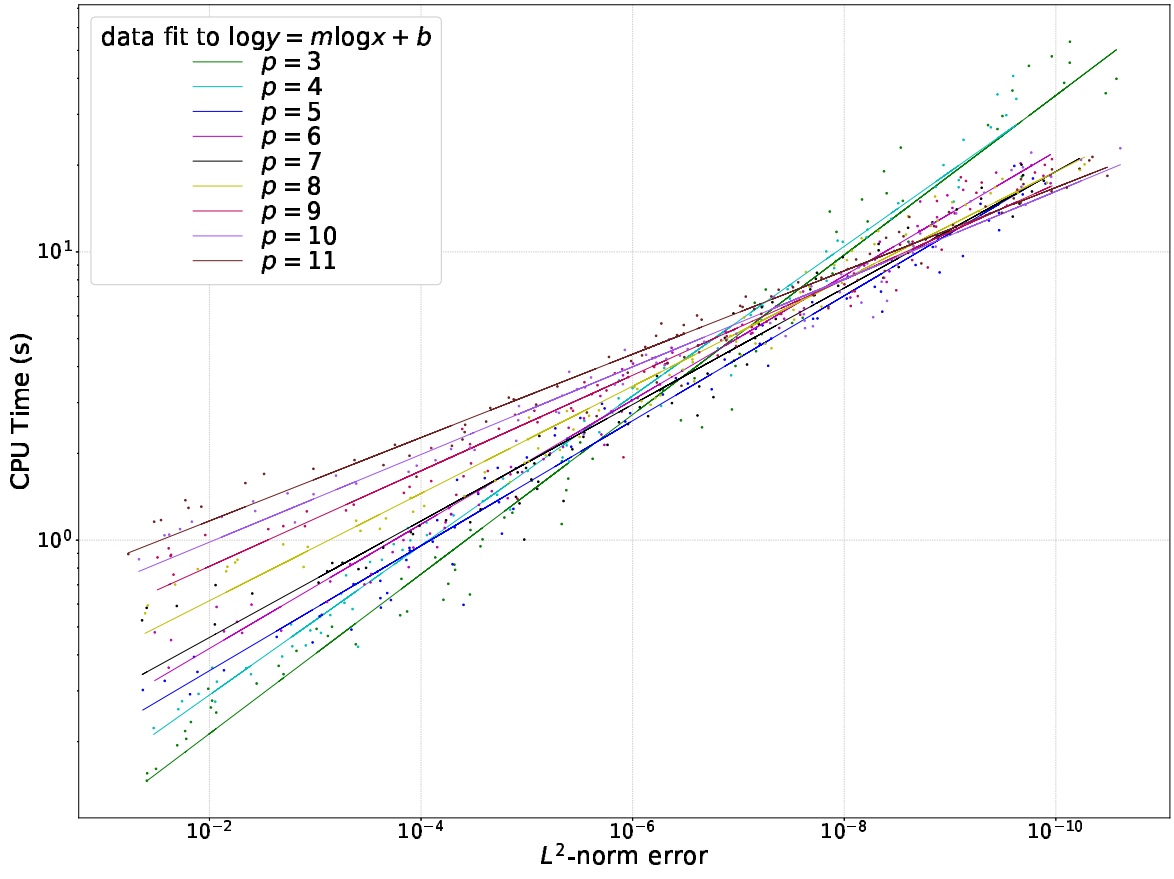


Figure 229: Work vs. Accuracy: One Layer Burgers equation $\epsilon = 10^{-4}$; BACOL/ST for $p = 3 \ldots 11$

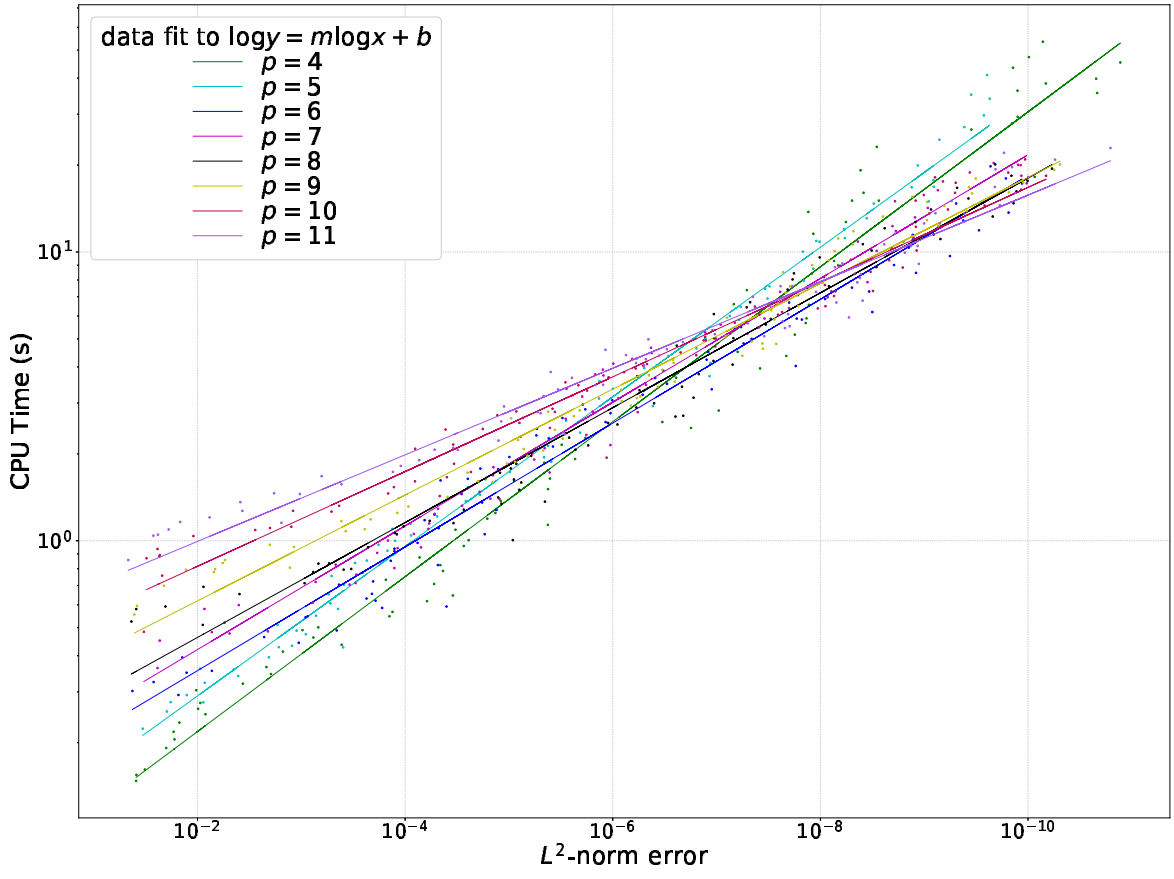


Figure 230: Work vs. Accuracy: One Layer Burgers equation $\epsilon = 10^{-4}$; BACOL/LE for $p = 4 \ldots 11$

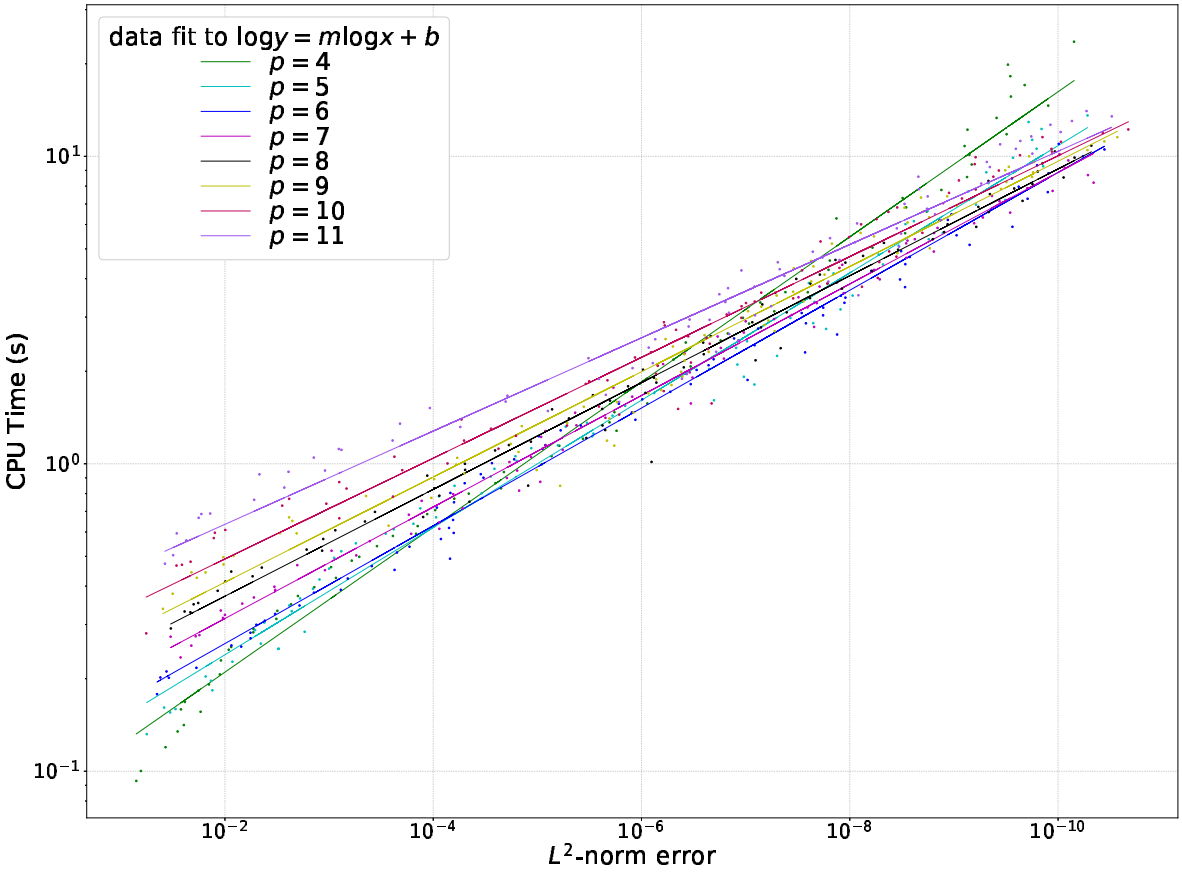


Figure 231: Work vs. Accuracy: One Layer Burgers equation $\epsilon = 10^{-4}$; BACOLI/ST for $p = 4 \dots 11$

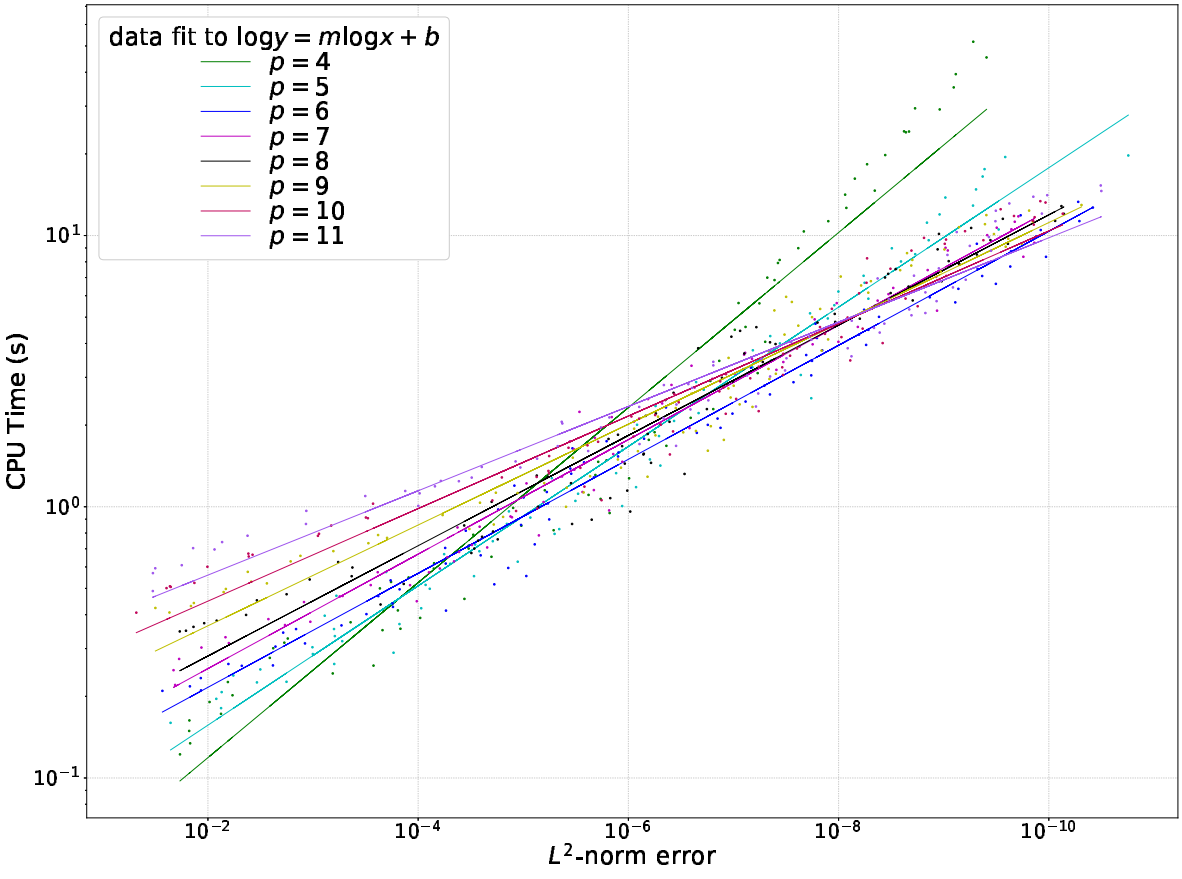


Figure 232: Work vs. Accuracy: One Layer Burgers equation $\epsilon = 10^{-4}$; BACOLI/LE for $p = 4 \ldots 11$

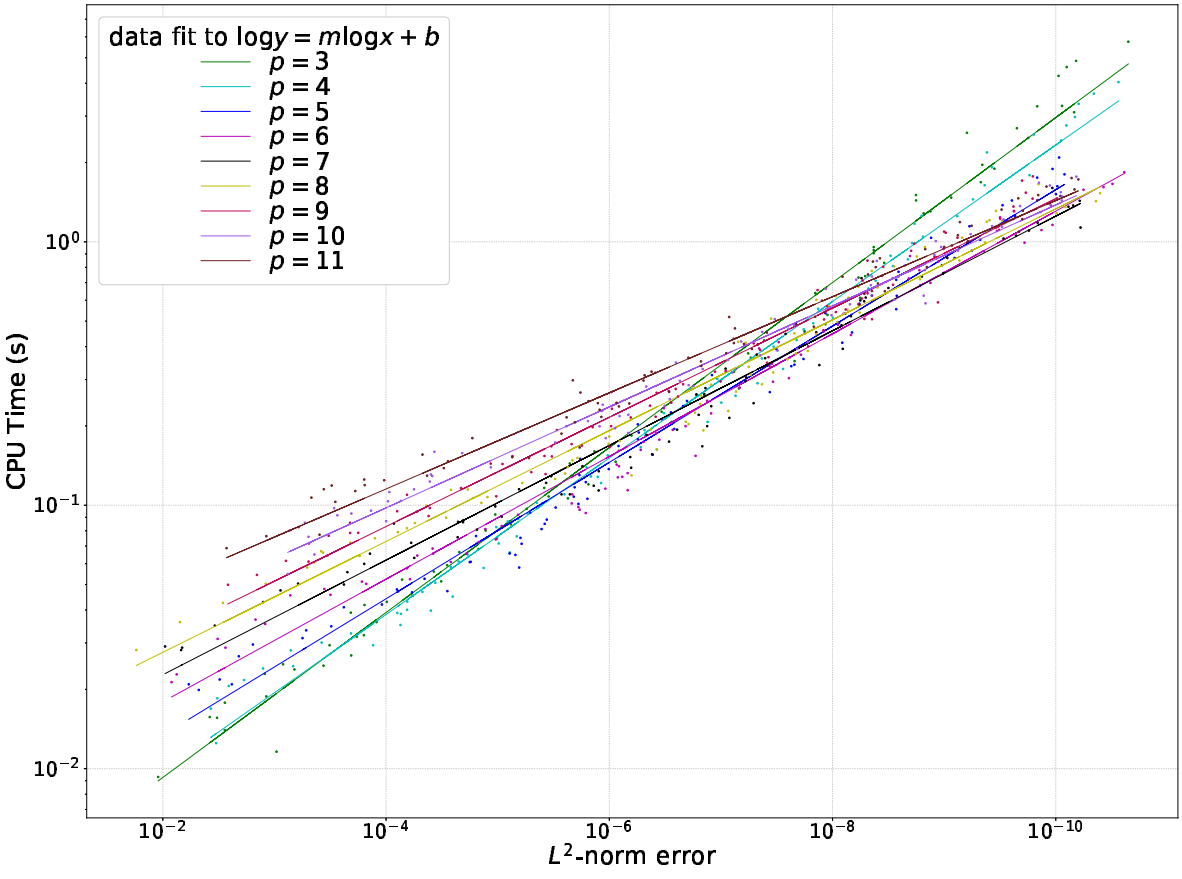


Figure 233: Work vs. Accuracy: Two Layer Burgers equation $\epsilon = 10^{-3}$; BACOL/ST for $p = 3 \ldots 11$

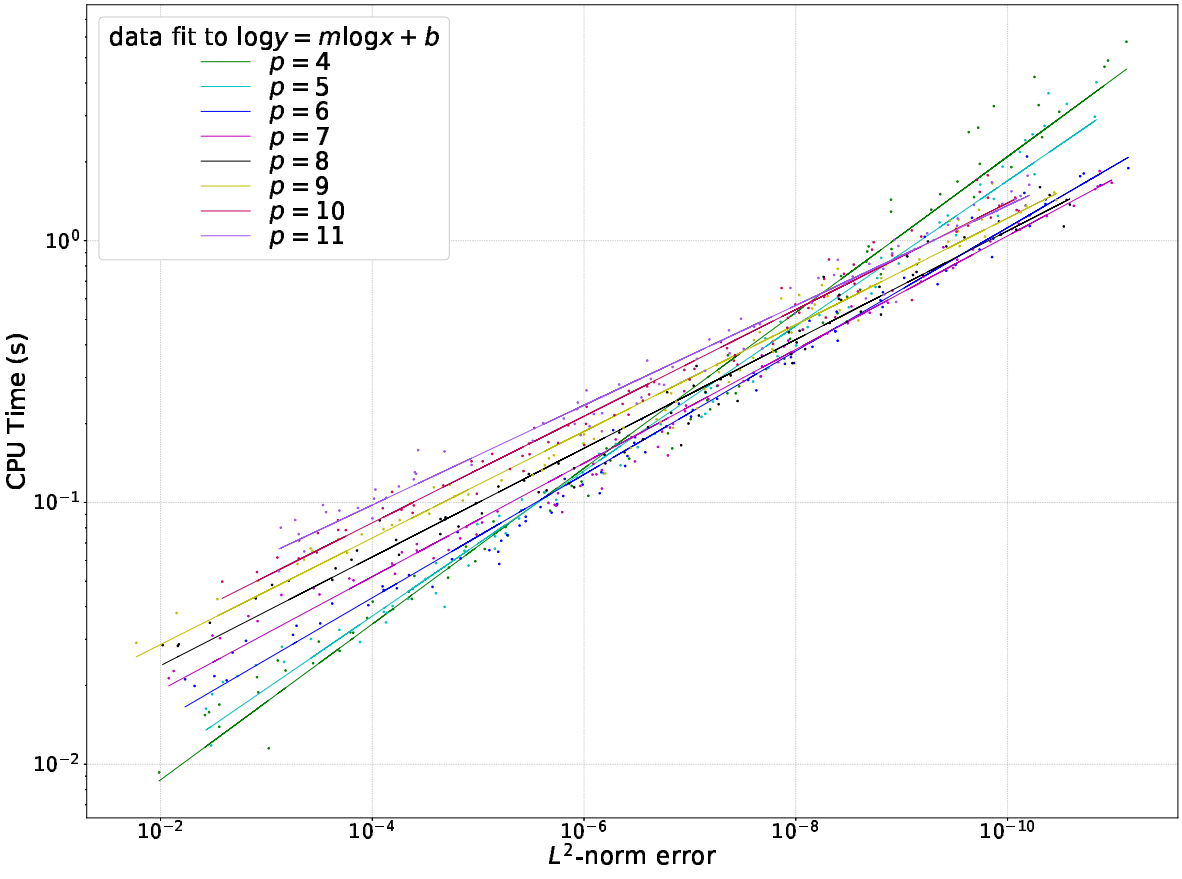


Figure 234: Work vs. Accuracy: Two Layer Burgers equation $\epsilon = 10^{-3}$; BACOL/LE for $p = 4 \ldots 11$

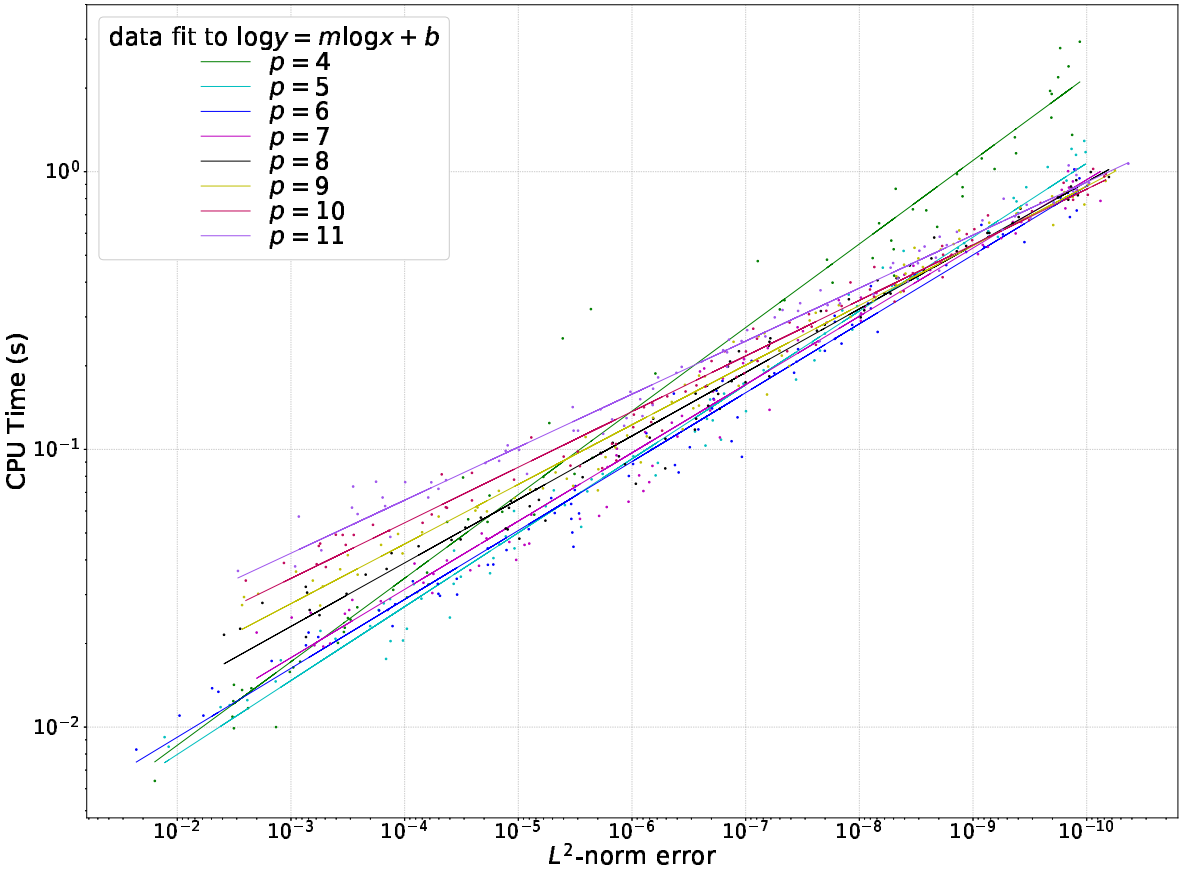


Figure 235: Work vs. Accuracy: Two Layer Burgers equation $\epsilon = 10^{-3}$; BACOLI/ST for $p = 4 \ldots 11$

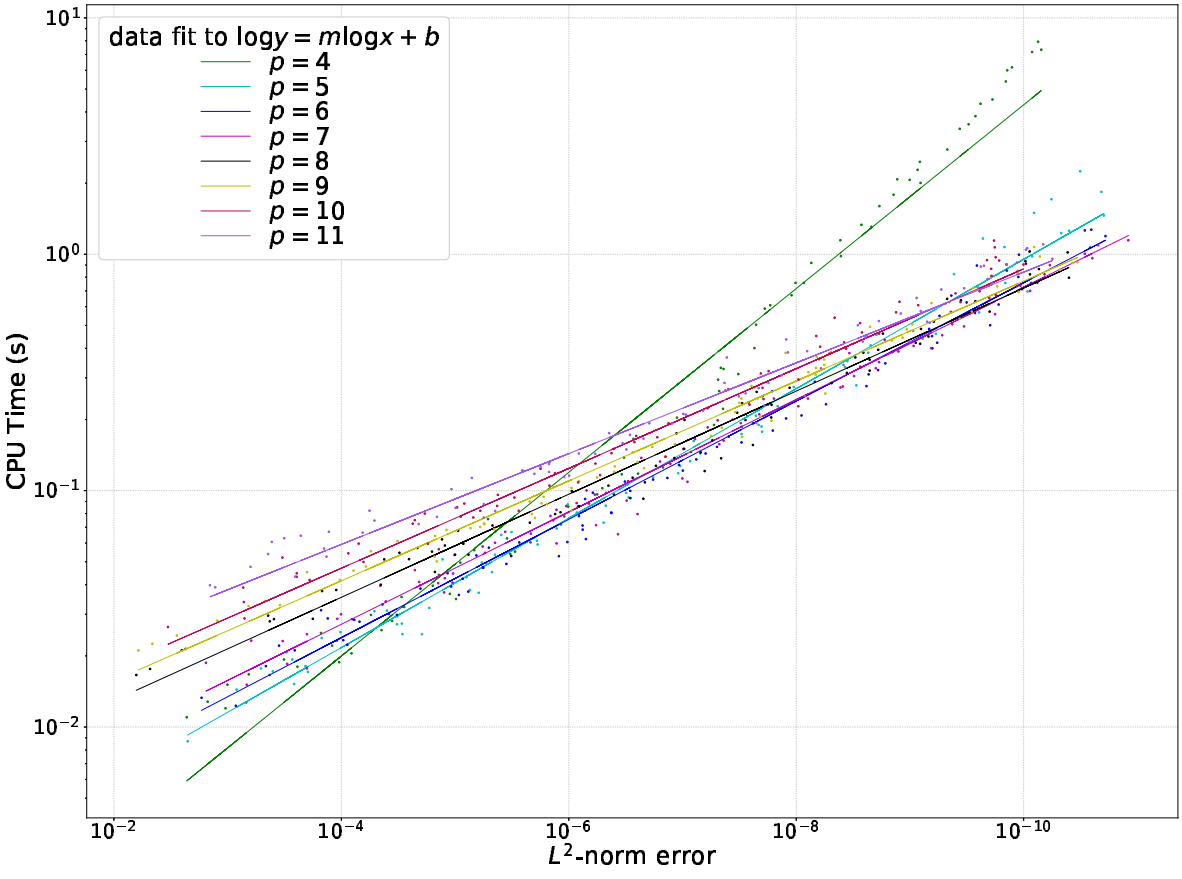


Figure 236: Work vs. Accuracy: Two Layer Burgers equation $\epsilon = 10^{-3}$; BACOLI/LE for $p = 4 \ldots 11$

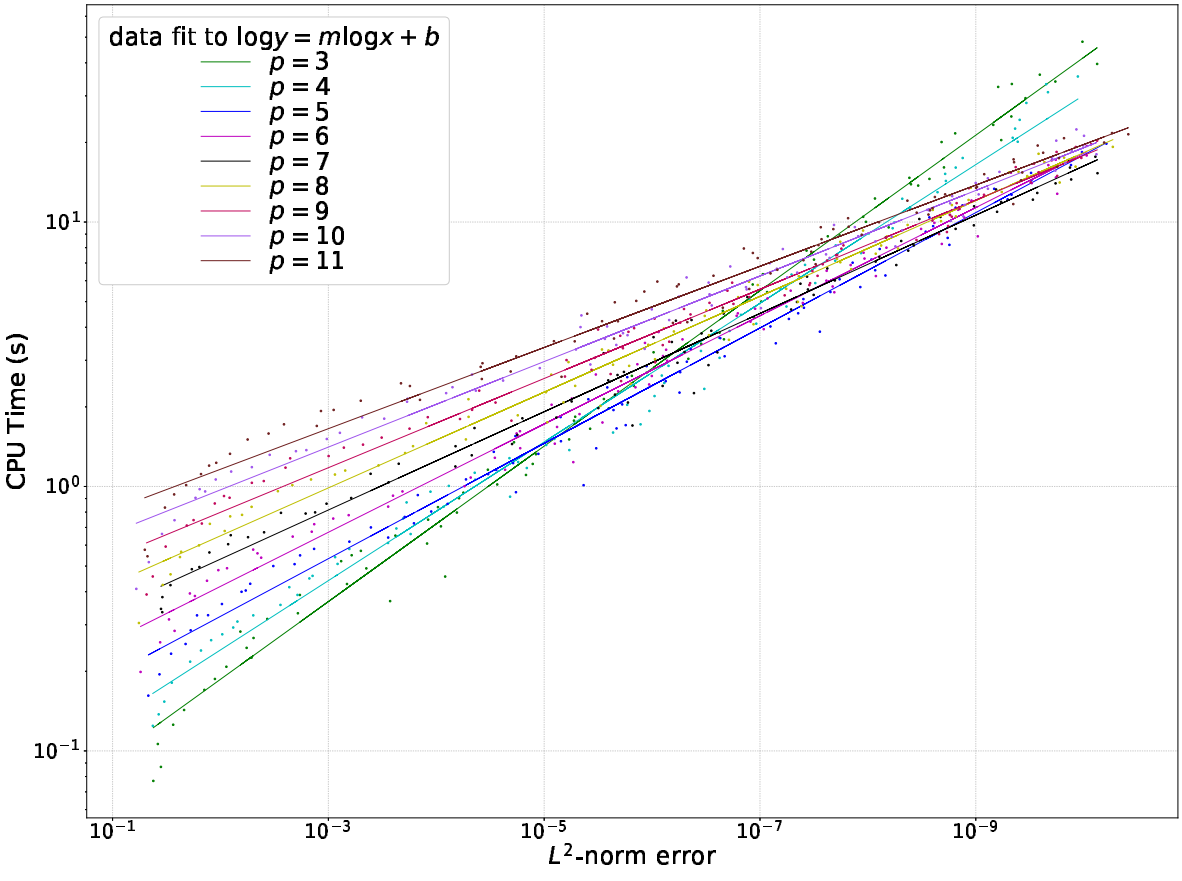


Figure 237: Work vs. Accuracy: Two Layer Burgers equation $\epsilon = 10^{-4}$; BACOL/ST for $p = 3 \ldots 11$

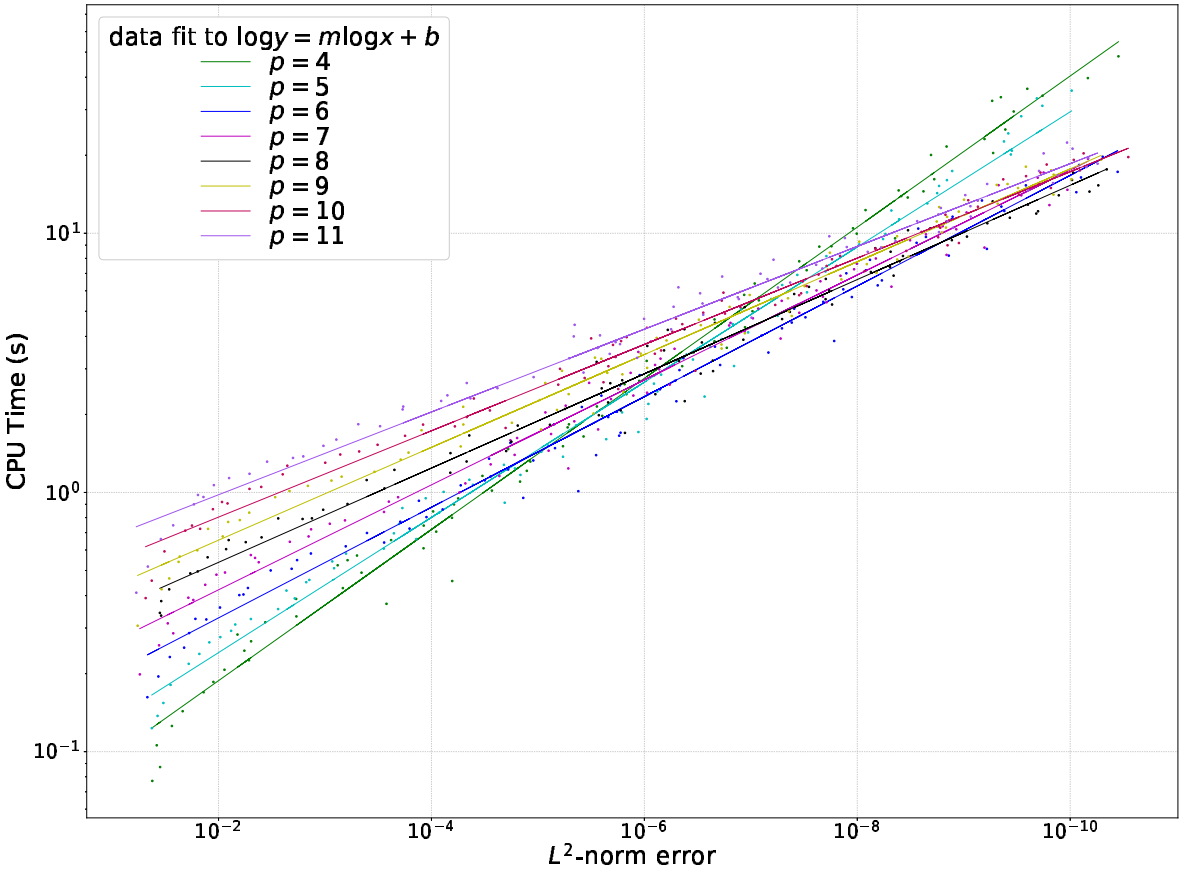


Figure 238: Work vs. Accuracy: Two Layer Burgers equation $\epsilon = 10^{-4}$; BACOL/LE for $p = 4 \ldots 11$

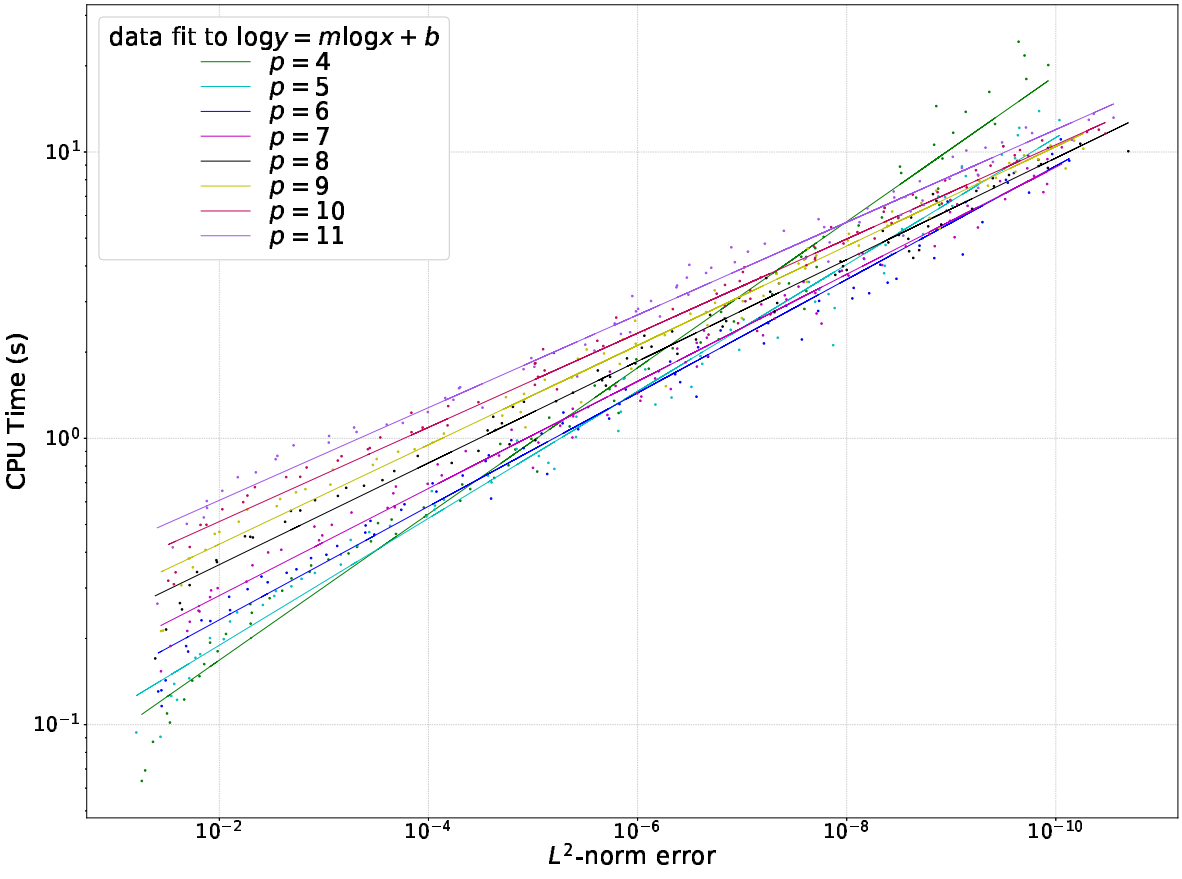


Figure 239: Work vs. Accuracy: Two Layer Burgers equation $\epsilon = 10^{-4}$; BACOLI/ST for $p = 4 \ldots 11$

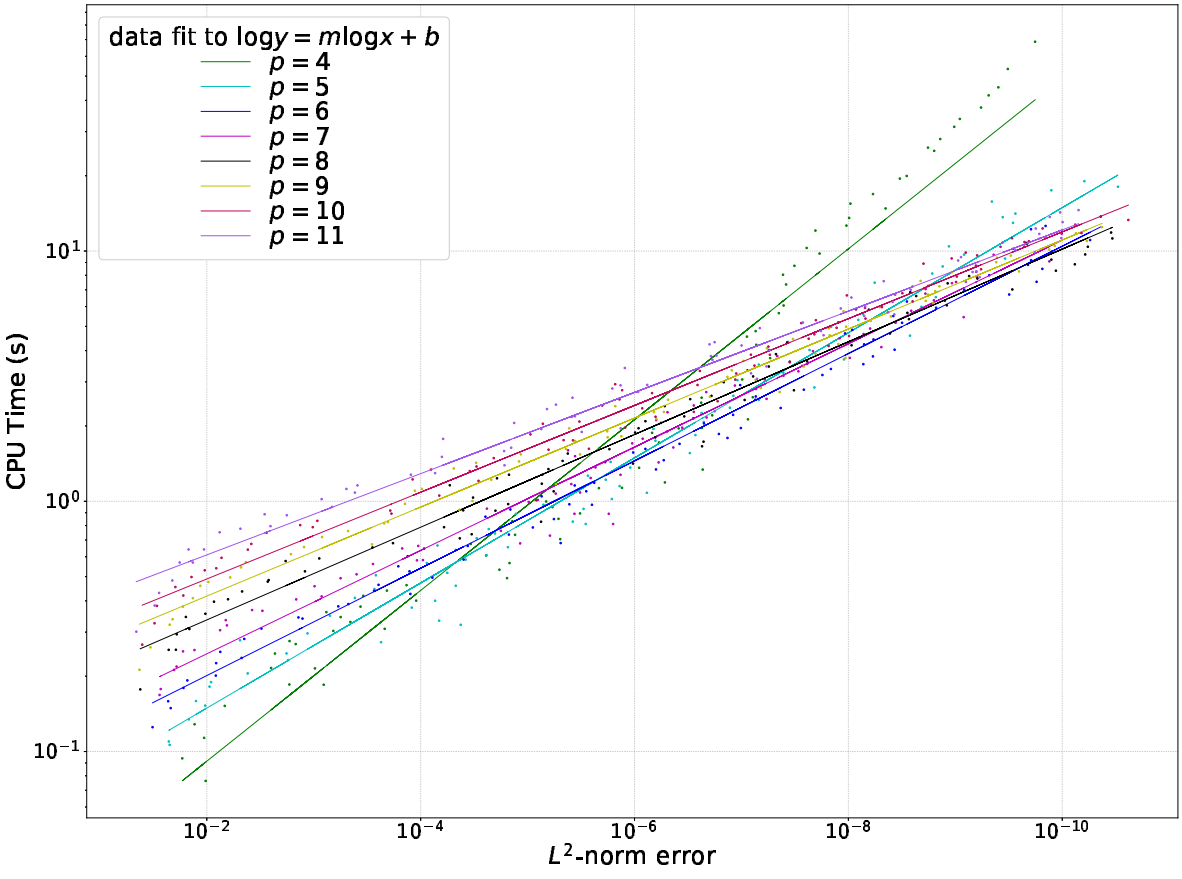


Figure 240: Work vs. Accuracy: Two Layer Burgers equation $\epsilon = 10^{-4}$; BACOLI/LE for $p = 4 \ldots 11$

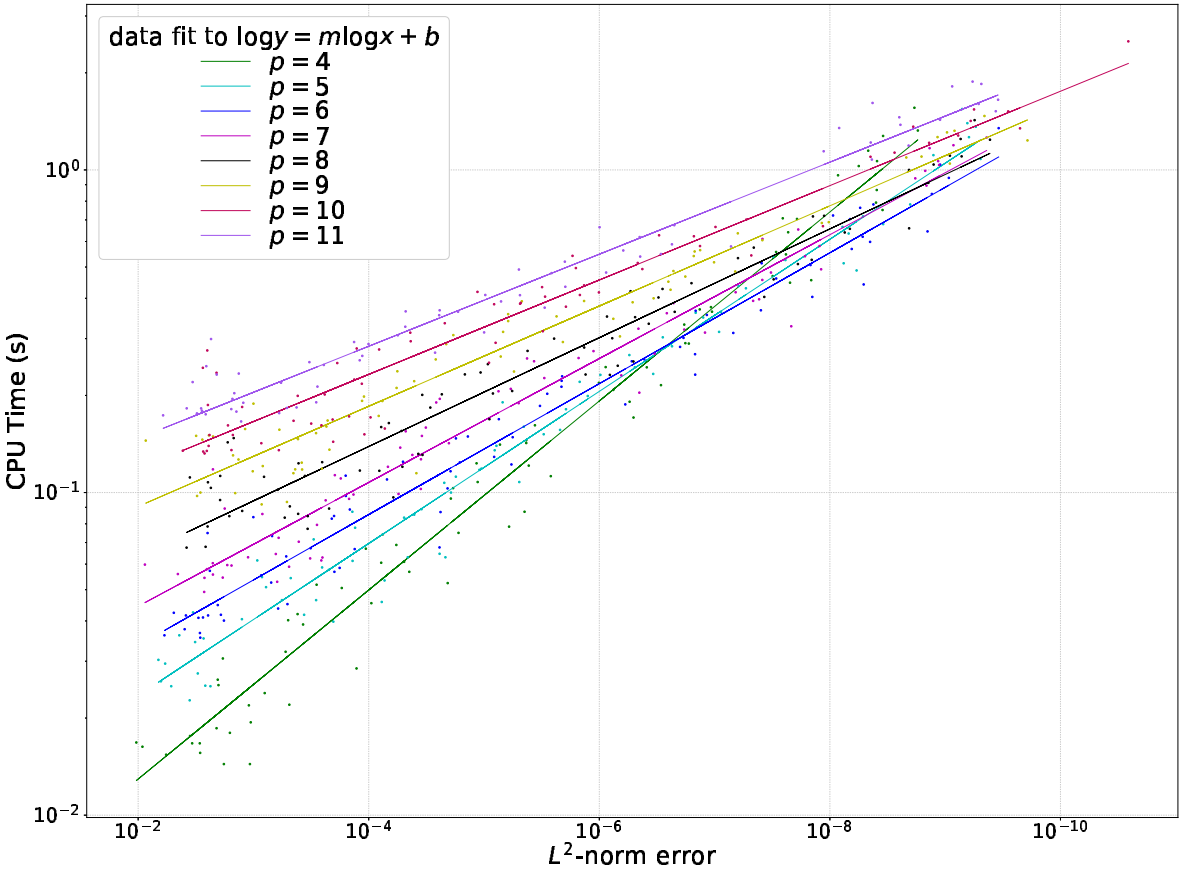


Figure 241: Work vs. Accuracy: Catalytic Surface Reaction Model; BACOL/ST for $p = 3 \ldots 11$

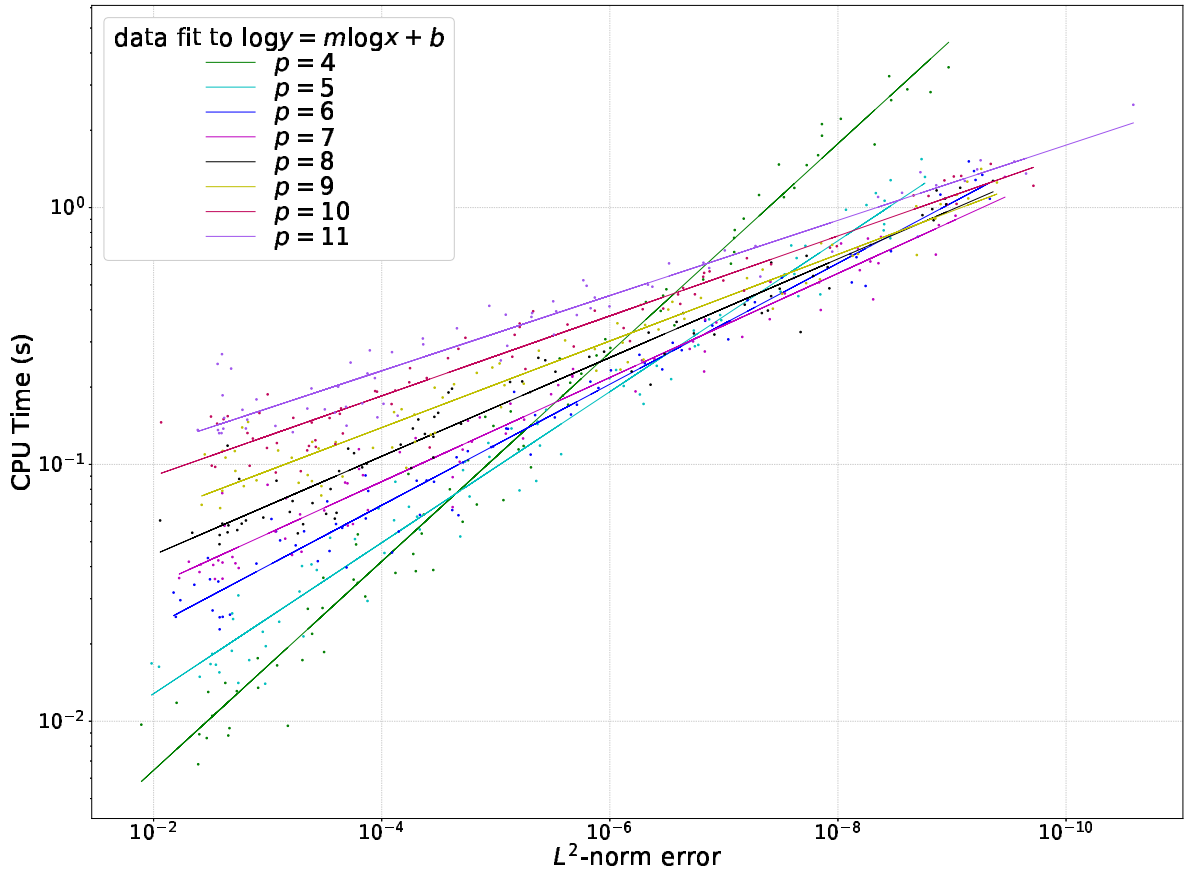


Figure 242: Work vs. Accuracy: Catalytic Surface Reaction Model; BACOL/LE for $p = 4 \ldots 11$

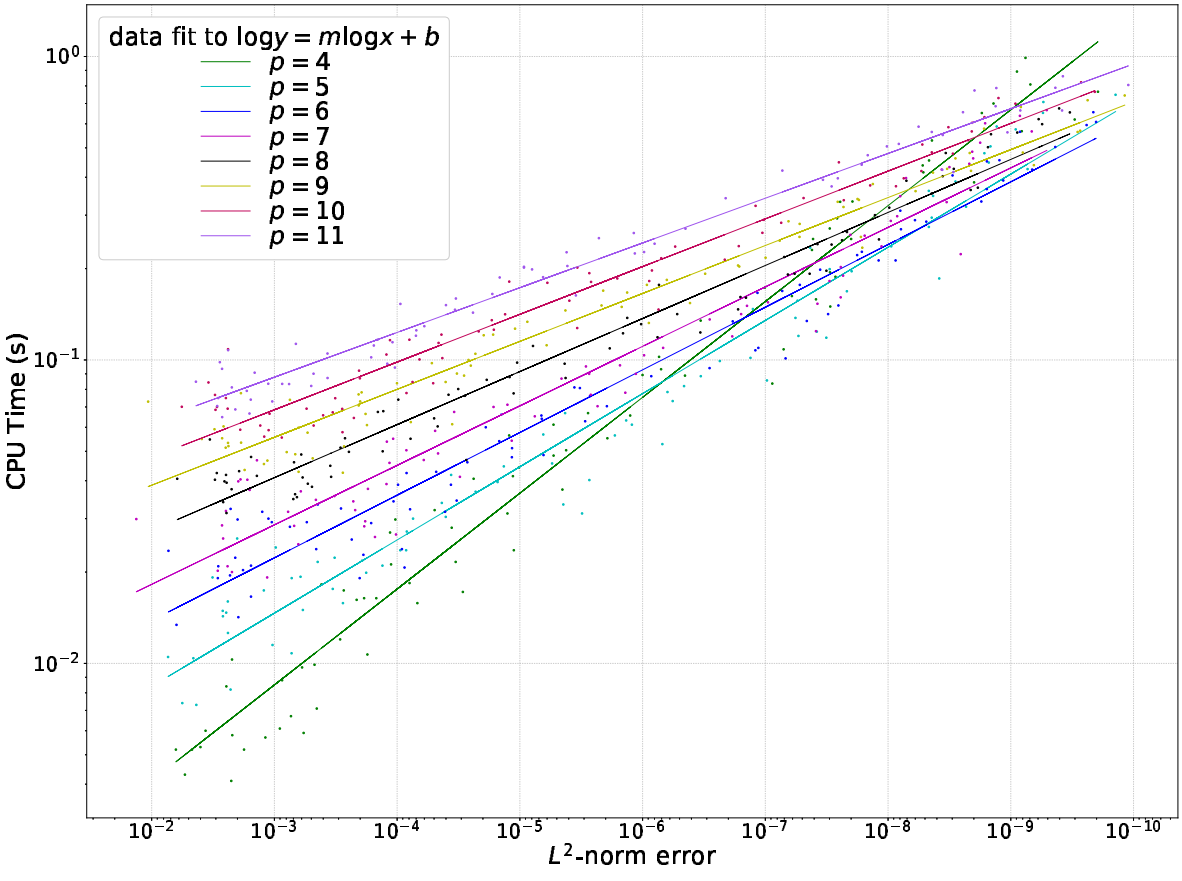


Figure 243: Work vs. Accuracy: Catalytic Surface Reaction Model; BACOLI/ST for $p = 4 \ldots 11$

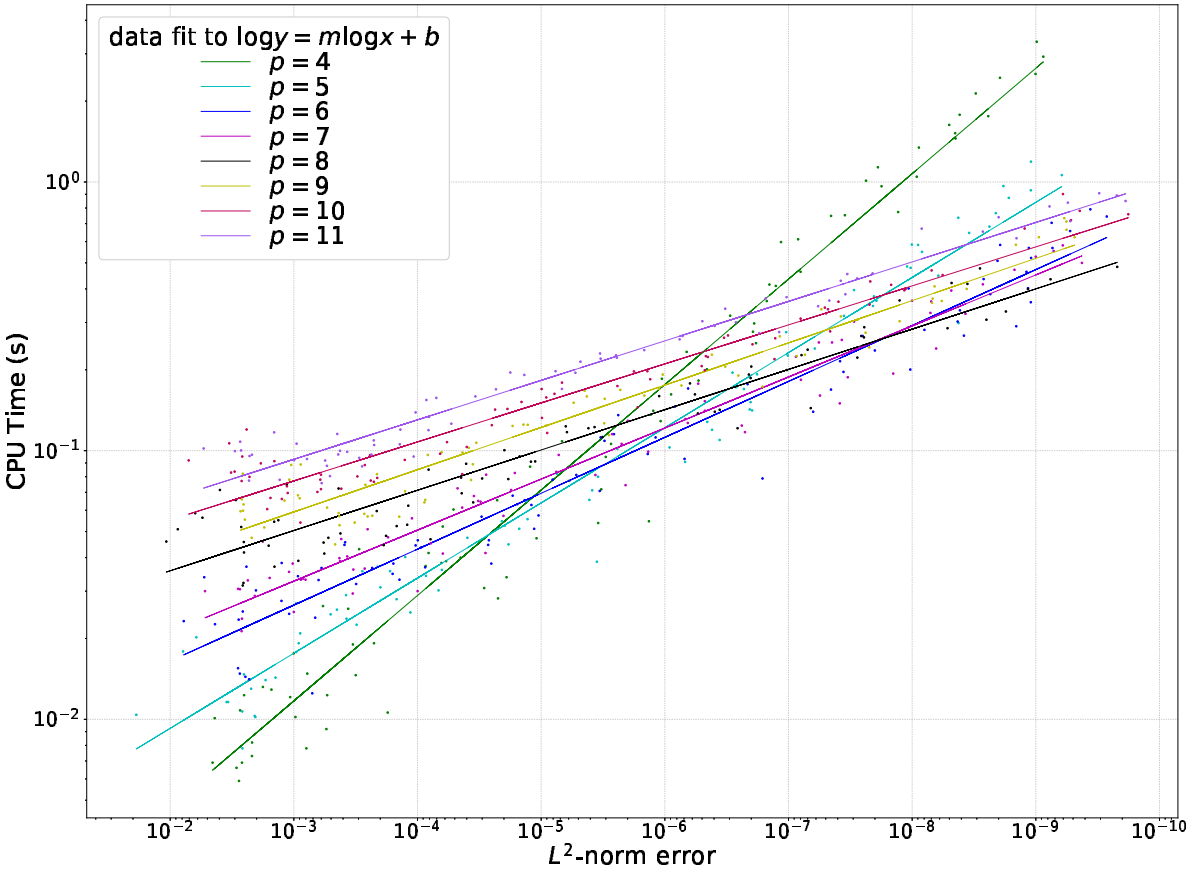


Figure 244: Work vs. Accuracy: Catalytic Surface Reaction Model; BACOLI/LE for $p = 4 \ldots 11$