A comparison of *hp*-adaptive strategies for elliptic partial differential equations

William F. Mitchell*
Marjorie A. McClain
Applied and Computational Mathematics Division
National Institute of Standards and Technology
Gaithersburg, MD 20899-8910

Abstract

The *hp* version of the finite element method (*hp*-FEM) combined with adaptive mesh refinement is a particularly efficient method for solving partial differential equations because it can achieve a convergence rate that is exponential in the number of degrees of freedom. *hp*-FEM allows for refinement in both the element size, h, and the polynomial degree, p. Like adaptive refinement for the h version of the finite element method, *a posteriori* error estimates can be used to determine where the mesh needs to be refined, but a single error estimate can not simultaneously determine whether it is better to do the refinement by h or by p. Several strategies for making this determination have been proposed over the years. In this paper we summarize these strategies and present the results of a numerical experiment to study the convergence properties of these strategies.

Keywords: elliptic partial differential equations, finite elements, *hp*-adaptive strategy, *hp*-FEM

1 Introduction

The numerical solution of partial differential equations (PDEs) is the most compute-intensive part of a wide range of scientific and engineering applications. Consequently the development and application of faster and more accurate methods for solving partial differential equations has received much attention in the past fifty years. Many of the applications at the cutting edge of research are extraordinarily challenging. For these problems it is necessary to allocate computing resources in an optimal way in order to have any chance at solving the problem. Determining the best grid and approximation space on which to compute the solution is a central concern in this regard. Unfortunately, it is rarely possible to determine an optimal grid in advance. Thus, developing self-adaptive techniques which lead to optimal resource allocation is critical for future progress in many fields.

Self-adaptive methods have been studied for over 30 years now. They are often cast in the context of finite element methods, where the domain of the PDE is partitioned into a mesh consisting of a number of elements (in two dimensions, usually triangles or rectangles), and the approximate solution is a polynomial over each element. Most of the work has focused on *h*-adaptive methods. In these methods, the mesh size,

 Preprint, submitted for publication October 2011. Check http://math.nist.gov/∼WMitchell for current publication status.

h, is adapted locally by means of a local error estimator with the goal of placing the smallest elements in the areas where they will do the most good. In particular, elements that have a large error estimate get refined so that ultimately the error estimates, and presumably the error, are approximately equal over all elements. h-adaptive methods are quite well understood now, and are beginning to be used in practice.

Recently, the research community has begun to focus more attention on hp-adaptive methods. In these methods, one not only locally adapts the size of the mesh, but also the degree of the polynomials, p. The attraction of hp-adaptivity is that the error converges at an exponential rate in the number of degrees of freedom, as opposed to a polynomial rate for fixed p. Much of the theoretical work showing the advantages of hp-adaptive methods was done in the 1980's, but it wasn't until the 1990's that practical implementation began to be studied. The new complication is that the local error estimator is no longer sufficient to guide the adaptivity. It tells you which elements should be refined, but it does not indicate whether it is better to refine the element by h or by p. A method for making that determination is called an hp-adaptive strategy. A number of strategies have been proposed, but it is not clear which ones perform best under different situations, or even if any of the strategies are good enough to be used as a general purpose solver. In this paper we present an experimental comparison of several hp-adaptive strategies.

Any study of this type is necessarily limited in scope. The comparison will be restricted to steady-state linear elliptic partial differential equations on bounded domains in two dimensions with Dirichlet, natural or mixed boundary conditions. The standard Galerkin finite element method will be used with the space of continuous piecewise polynomial functions over triangles that are refined by the newest node bisection method.

The remainder of the paper is organized as follows. In Section 2 we define the equation to be solved, present the finite element method, and give some *a priori* error estimates. In Section 3 we give the details of the hp-adaptive finite element algorithm used in the experiments. Section 4 defines the hp-adaptive strategies to be compared. Section 5 contains the results of the experiments. Finally, we draw our conclusions in Section 6.

2 The Finite Element Method

We consider the elliptic partial differential equation

$$\mathcal{L}u = -\frac{\partial}{\partial x}\left(p(x,y)\frac{\partial u}{\partial x}\right) - \frac{\partial}{\partial y}\left(q(x,y)\frac{\partial u}{\partial y}\right) + r(x,y)u = f(x,y) \quad \text{in } \Omega \tag{1}$$

$$u = g_D(x,y) \quad \text{on } \partial\Omega_D \tag{2}$$

$$\mathcal{B}u = p(x,y)\frac{\partial u}{\partial x}\frac{\partial y}{\partial s} - q(x,y)\frac{\partial u}{\partial y}\frac{\partial x}{\partial s} + c(x,y)u = g_N(x,y) \quad \text{on } \partial\Omega_N \tag{3}$$

where Ω is a bounded, connected, polygonal, open region in $\mathbb{R}^2$ with boundary $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$, $\partial\Omega_D \cap \partial\Omega_N = \emptyset$. Differentiation with respect to s is with respect to a counterclockwise parameterization of the boundary $(x(s), y(s))$ with $\|(dx/ds\ dy/ds)\| = 1$. If $c = 0$ Equation 3 is the natural boundary condition. If, in addition, $p = q = 1$ or $\partial\Omega_N$ consists of line segments that are parallel to the axes, Equation 3 is the Neumann boundary condition. We assume the data in Equations 1-3 satisfy the usual ellipticity and regularity assumptions. In one of the test problems, we extend the equation to a system of two equations containing a cross derivative term $\partial^2 u/\partial x \partial y$, and in another test problem we include first order derivative terms.

As usual, define the space L^2 by

$$L^2(\Omega) = \{v(x,y) : \iint_\Omega v^2 \, dx \, dy < \infty\}$$

with inner product

$$\langle u, v \rangle_2 = \iint_\Omega uv \, dx \, dy$$

and norm

$$||v||_2^2 = \langle v, v \rangle_2.$$

We denote by $H^m(\Omega)$ the usual Sobolov spaces

$$H^m(\Omega) = \{v \in L^2(\Omega) : D^\alpha v \in L^2(\Omega), |\alpha| \leq m\}$$

where

$$D^\alpha v = \frac{\partial^{|\alpha|} v}{\partial^{\alpha_1} x \partial^{\alpha_2} y}, \alpha = (\alpha_1, \alpha_2), \alpha_i \in \mathbb{N}, |\alpha| = \alpha_1 + \alpha_2.$$

The Sobolov spaces have inner products

$$\langle u, v \rangle_{H^m(\Omega)} = \iint_\Omega \sum_{|\alpha| \leq m} D^\alpha u D^\alpha v \, dx \, dy$$

and norms

$$||v||_{H^m(\Omega)}^2 = \langle v, v \rangle_{H^m(\Omega)}.$$

We will also refer to the seminorm $|v|_{H^m(\Omega)}$ where the sum is over $|\alpha| = m$.

Let $H_0^m(\Omega) = \{v \in H^m(\Omega) : v = 0 \text{ on } \partial\Omega_D\}$. Let $\tilde{u}_D$ be a lift function satisfying the Dirichlet boundary conditions in Equation 2 and define the affine space $\tilde{u}_D + H_0^1(\Omega) = \{\tilde{u}_D + v : v \in H_0^1(\Omega)\}$. Define the bilinear form

$$B(u,v) = \iint_\Omega p \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} + q \frac{\partial u}{\partial y} \frac{\partial v}{\partial y} + ruv \, dx \, dy + \int_{\partial\Omega_N} cuv \, ds$$

and the linear form

$$L(v) = \iint_\Omega fv \, dx \, dy + \int_{\partial\Omega_N} g_N v \, ds$$

Then the variational form of the problem is to find the unique $u \in \tilde{u}_D + H_0^1(\Omega)$ that satisfies

$$B(u,v) = L(v) \quad \forall v \in H_0^1(\Omega).$$

The energy norm of $v \in H_0^1$ is defined by $||v||_{E(\Omega)}^2 = B(v,v)$.

The finite element space is defined by partitioning Ω into a grid (or mesh), G_{hp}, consisting of a set of N_T triangular elements, $\{T_i\}_{i=1}^{N_T}$ with $\bar{\Omega} = \cup_{i=1}^{N_T} \bar{T}_i$. If a vertex of a triangle is contained in the interior of an edge of another triangle, it is called a hanging node. We only consider compatible grids with no hanging nodes, i.e. $\bar{T}_i \cap \bar{T}_j$, $i \neq j$, is either empty, a common edge, or a common vertex. The diameter of the element is denoted h_i. With each element we associate an integer degree $p_i \geq 1$. The finite element space V_{hp} is

the space of continuous piecewise polynomial functions on Ω such that over element T_i it is a polynomial of degree p_i. The degree of an edge is determined by applying either a minimum rule or a maximum rule over G_{hp} in which the edge is assigned the minimum or maximum of the degrees of the adjacent elements, respectively.

The finite element solution is the unique function $u_{hp} \in \tilde{u}_D + V_{hp}$ that satisfies

$$B(u_{hp}, v_{hp}) = L(v_{hp}) \quad \forall v_{hp} \in V_{hp}.$$

The error is defined by $e_{hp} = u - u_{hp}$.

The finite element solution is expressed as a linear combination of basis functions $\{\phi_i\}_{i=1}^N$ that span $\tilde{u}_D + V_{hp}$,

$$u_{hp} = \sum_{i=1}^{N} \alpha_i \phi_i(x, y)$$

N is called the number of degrees of freedom. For high order elements, there are a number of basis sets used in practice. A number of the *hp* strategies of Section 4 rely on the basis being a p-hierarchical basis in which the basis functions for a space of degree p are a subset of the basis functions for a space of degree $p + 1$. In the results of Section 5 the p-hierarchical basis of Szabo and Babuška [33], which is based on Legendre polynomials, is used.

The discrete form of the problem is a linear system of algebraic equations

$$Ax = b \tag{4}$$

where the matrix A is given by $A_{ij} = B(\phi_i, \phi_j)$ and the right hand side is given by $b_i = L(\phi_i)$.

If h and p are uniform over the grid, $u \in H^m(\Omega)$, and the other usual assumptions are met, then the *a priori* error bound is [6, 7]

$$||e_{hp}||_{H^1(\Omega)} \leq C \frac{h^\mu}{p^{m-1}} ||u||_{H^m(\Omega)} \tag{5}$$

where $\mu = \min(p, m-1)$ and C is a constant that is independent of h, p and u, but depends on m.

With a suitably chosen *hp* mesh, and other typical assumptions, the error can be shown [14] to converge exponentially in the number of degrees of freedom,

$$||e_{hp}||_{H^1(\Omega)} \leq C_1 e^{-C_2 N^{1/3}} \tag{6}$$

for some C_1 and $C_2 > 0$ independent of N.

3 hp-Adaptive Refinement Algorithm

One basic form of an *hp*-adaptive algorithm is given in Figure 1. There are a number of approaches to each of the steps of the algorithm. For example, how is an element h-refined? How is an element p-refined? What error indicator is used to guide adaptive refinement? When is the program terminated? How is an element coarsened? How do you determine which elements should be coarsened? How do you determine which elements should be refined? How much refinement should occur before the linear system is formed and solved again? Should an element be refined by h or p?

Complete coverage of the possible answers to these questions is beyond the scope of this paper. We will focus on the choices used for the results given in Section 5, and in some cases briefly mention other

```
begin with a very coarse grid
form and solve the linear system
repeat
  determine which elements to coarsen and whether to coarsen by h or p
  coarsen elements
  repeat
    determine which elements to refine and whether to refine by h or p
    refine elements
  until some criterion on amount of refinement is met
  form and solve the linear system
until some termination criterion is met
```

Figure 1: Basic form of an *hp*-adaptive algorithm.

possibilities or give a reference, but this is not intended to be exhaustive. Note that some of the *hp* strategies in Section 4 require a different choice, or even a modification of the basic algorithm. These exceptions will be noted in Section 4.

There are several ways to refine triangles [19]. In this paper, the newest node bisection method [20] is used. Briefly, a parent triangle is h-refined by connecting one of the vertices to the midpoint of the opposite side to form two new child triangles. The most recently created vertex is chosen as the vertex to use in this bisection. Triangles are always refined in pairs (except when the edge to be refined is on the boundary) to maintain compatibility of the grid. This may require first refining a neighbor triangle to create the second triangle of the pair. The h-refinement level, l_i, of a triangle T_i is one more than the h-refinement level of the parent, with level 0 assigned to the triangles of the initial coarse grid. p-refinement is fairly universally accepted as increasing the degree of the element by one, followed by enforcing either the minimum rule or maximum rule for the edges. We will use the minimum rule.

Adaptive refinement is guided by a local *a posteriori* error indicator computed for each element. There are several choices of error indicators; see for example [2, 34]. For this paper, the error indicator for element T_i is given by solving a local Neumann residual problem:

$$\mathcal{L}e_i = f - \mathcal{L}u_{hp} \qquad \text{in } T_i \tag{7}$$

$$e_i = 0 \qquad \text{on } \partial T_i \cap \partial\Omega_D \tag{8}$$

$$\mathcal{B}e_i = g_N - \mathcal{B}u_{hp} \qquad \text{on } \partial T_i \cap \partial\Omega_N \tag{9}$$

$$\mathcal{B}e_i = -\frac{1}{2}\left[\frac{\partial u_{hp}}{\partial n}\right] \qquad \text{on } (\partial T_i \setminus \partial\Omega_D) \setminus \partial\Omega_N \tag{10}$$

where $\mathcal{L}$, $\mathcal{B}$, f, g_N, $\partial\Omega_D$, and $\partial\Omega_N$ are defined in Equations 1-3, $\left[\frac{\partial u_{hp}}{\partial n}\right]$ is the jump in the outward normal derivative of u_{hp} across the element boundary, including the coefficients of the natural boundary conditions, and in Equation 10 $\mathcal{B}$ is modified by setting $c(x, y) = 0$. If the degree of T_i is p_i, the approximate solution, $e_{i,hp}$ of Equations 7-10 is computed using the hierarchical bases of exact degree $p_i + 1$. The error indicator for element T_i is then given by

$$\eta_i = ||e_{i,hp}||_{E(T_i)}$$

A global energy norm error estimate is given by

$$\eta = \left(\sum_{i=1}^{N_T} \eta_i^2 \right)^{1/2}.$$

One criterion for program termination is that η be smaller than some prescribed error tolerance τ, or, to base it on the relative error rather than the absolute error, $\eta < \tau ||u_{hp}||_{E(\Omega)}$. Other possibilities are to terminate when some quantity, such as number of elements, number of degrees of freedom, amount of memory, amount of computation time, etc., is reached, or combinations of criteria. In this paper, the primary termination criterion is a relative error tolerance, with number of degrees of freedom as a secondary criterion.

Coarsening of elements may be performed to reverse bad decisions about what refinements to perform, or to allow the grid to follow the behavior of the solution in a time dependent problem. Elements are h-coarsened by reversing the h-refinement, i.e., joining the child triangles back together to form the parent triangle. p-coarsening means decreasing the degree of the element by one, and enforcing the minimum or maximum rule for the edges. For steady state problems, one choice of which elements to coarsen is the empty set, i.e., don't perform coarsening. Other than that choice, the most common approach is to coarsen elements that have a sufficiently small error indicator, subject to any requirements for compatibility of the grid. In the numerical results of this paper, an element is coarsened if $\eta_i < \max_i \eta_i/100$. The value 100 is arbitrary.

The elements that are refined are usually those that have a sufficiently large error indicator. Perhaps the most common approach is to refine those with an error indicator that is larger than some fraction, typically between 1/4 and 1/2, of the maximum error indicator. Another approach, which is used in this paper, is to refine those with $\eta_i > \tau ||u_{hp}||_{E(\Omega)}/\sqrt{N_T}$. Note that if every element had $\eta_i = \tau ||u_{hp}||_{E(\Omega)}/\sqrt{N_T}$ then $\eta/||u_{hp}||_{E(\Omega)} = \tau$, hence the $\sqrt{N_T}$ factor.

There are many ways to determine how much refinement to do before forming and solving the linear system. One could refine until the global error estimate has been reduced by some factor, such as 1/2 or 1/4, or one could refine until some quantity, e.g. number of elements or degrees of freedom, has been increased by some factor, such as 2 or 4. Both of these require that reasonable error indicators can be computed on the child elements. The approach taken in this paper is to perform the refine loop once. The downside of this approach is that it requires more passes through the outer loop, which means forming and solving the linear system more times. But for the purpose of this paper, which is to determine the convergence rate of various hp-adaptive strategies with respect to number of degrees of freedom, the excess computation time is not important.

The method for determining whether an element should be refined by h or by p is called an hp-adaptive strategy. Several strategies have been proposed over the years. Many of them will be described in the next section.

4 The hp-Adaptive Strategies

In this section, the hp-adaptive strategies that have been proposed in the literature are briefly described. For brevity, many of the details have been omitted. For a detailed description of the strategies, see [23] or [22]. In some cases, these strategies were developed in the context of 1D problems, rectangular elements, or other settings that are not fully compatible with the context of this paper. In those cases, the strategy is appropriately modified for 2D elliptic PDEs and newest node bisection of triangles.

4.1 Use of *a priori* Knowledge of Solution Regularity

It is well known that for smooth solutions p-refinement will produce an exponential rate of convergence, but near singularities p-refinement is less effective than h-refinement. This is a consequence of the *a priori* error bound in Equation 5. For this reason, many of the hp strategies use h-refinement in areas where the solution is irregular (i.e., locally fails to be in H^m for some finite m, also called nonsmooth) or nearly irregular, and p-refinement elsewhere. The simplest strategy is to use any *a priori* knowledge about irregularities. An hp-adaptive strategy of this type was presented by Ainsworth and Senior [4]. In this approach they simply flag vertices in the initial mesh as being possible trouble spots. During refinement an element is refined by h if it contains a vertex that is so flagged, and by p otherwise. We will refer to this strategy by the name APRIORI.

4.2 Estimate Regularity Using Smaller p Estimates

Süli, Houston and Schwab [32] presented a strategy based on Equation 5 and an estimate of the convergence rate in p using error estimates based on $p_i - 2$ and $p_i - 1$. We will refer to this strategy as PRIOR2P.

Suppose the error estimate in Equation 5 holds on individual elements and that the inequality is an approximate equality. Let η_{i,p_i-2} and η_{i,p_i-1} be *a posteriori* error estimates for partial approximate solutions over triangle T_i using the bases up to degree $p_i - 2$ and $p_i - 1$, respectively. Then

$$\frac{\eta_{i,p_i-1}}{\eta_{i,p_i-2}} \approx \left(\frac{p_i - 1}{p_i - 2}\right)^{-(m_i-1)}$$

and thus the regularity is estimated by

$$m_i \approx 1 - \frac{\log(\eta_{i,p_i-1}/\eta_{i,p_i-2})}{\log((p_i - 1)/(p_i - 2))}$$

Use p-refinement if $p_i \leq m_i - 1$ and h-refinement otherwise.

4.3 Type parameter

Gui and Babuška [13] presented an hp-adaptive strategy using what they call a type parameter, γ. This strategy is also used by Adjerid, Aiffa and Flaherty [1]. We will refer to this strategy as TYPEPARAM.

Given the error estimates η_{i,p_i} and η_{i,p_i-1}, define

$$R(T_i) = \begin{cases} \frac{\eta_{i,p_i}}{\eta_{i,p_i-1}} & \eta_{i,p_i-1} \neq 0 \\ 0 & \eta_{i,p_i-1} = 0 \end{cases}$$

By convention, $\eta_{i,0} = 0$, which forces p-refinement if $p_i = 1$.

R is used to assess the perceived solution smoothness. Given the type parameter, $0 \leq \gamma < \infty$, element T_i is said to be of h-type if $R(T_i) > \gamma$, and of p-type if $R(T_i) \leq \gamma$. If element T_i is selected for refinement, then refine it by h-refinement if it is of h-type and p-refinement if it is of p-type. Note that $\gamma = 0$ gives pure h-refinement and $\gamma = \infty$ gives pure p-refinement.

For the results of Section 5, we use $\gamma = 0.3$ if the solution has a singularity, and $\gamma = 0.6$ otherwise. [1]

[1] The value for this parameter, and the parameters of the other strategies, was determined by a preliminary experiment to determine a single value (or possibly two values dependent on singularness) that generally works best, using a subset of the test problems.

4.4 Estimate Regularity Using Larger p Estimates

Another approach that estimates the regularity is given by Ainsworth and Senior [3]. This strategy uses three error estimates based on spaces of degree $p_i + 1$, $p_i + 2$ and $p_i + 3$, so we refer to it as NEXT3P.

The error estimate used to approximate the regularity is a variation on the local Neumann residual error estimate given by Equations 7-10 in which Equation 10 is replaced by

$$\mathcal{B}e_i = g_i \quad \text{on } (\partial T_i \setminus \partial\Omega_D) \setminus \partial\Omega_N$$

where g_i is an approximation of $\mathcal{B}u$ that satisfies an equilibrium condition. This is the equilibrated residual error estimator in [2].

The local problem is solved on element T_i three times using the spaces of degree $p_i + q$, $q = 1, 2, 3$, to obtain error estimates $e_{i,q}$. In contrast to the local Neumann residual error estimate, the whole space over T_i is used, not just the p-hierarchical bases of degree greater than p_i. These approximations to the error converge to the true solution of the residual problem at the same rate the approximate solution converges to the true solution of Equations 1-3, i.e.

$$||e_i - e_{i,q}||_{E(T_i)} \approx C(p_i + q)^{-\alpha}$$

where C and α are positive constants that are independent of q but depend on T_i. Using the Galerkin orthogonality

$$||e_i - e_{i,q}||^2_{E(T_i)} = ||e_i||^2_{E(T_i)} - ||e_{i,q}||^2_{E(T_i)}$$

this can be rewritten

$$||e_i||^2_{E(T_i)} - ||e_{i,q}||^2_{E(T_i)} \approx C^2(p_i + q)^{-2\alpha}.$$

We can compute $||e_{i,q}||^2_{E(T_i)}$ and $p_i + q$ for $q = 1, 2, 3$ from the approximate solutions, so the three constants $||e_i||_{E(T_i)}$, C and α can be approximated by fitting the data. Then, using the *a priori* error estimate in Equation 5, the approximation of the local regularity is $m_i = 1 + \alpha$. Use p-refinement if $p_i \leq m_i - 1$ and h-refinement otherwise.

4.5 Texas 3 Step

The Texas 3 Step strategy [8, 24, 25] first performs h-refinement to get an intermediate grid, and follows that with p-refinement to reduce the error to some given error tolerance, τ. We will refer to this strategy as T3S. Note that for this strategy the basic form of the hp-adaptive algorithm is different than that in Figure 1.

The first step is to create an initial mesh with uniform p and nearly uniform h such that the solution is in the asymptotic range of convergence in h. The second step is to perform adaptive h-refinement to reach an intermediate error tolerance $\gamma\tau$ where γ is a given parameter. In the references, γ is in the range $5 - 10$, usually 6 in the numerical results. This intermediate grid is created by computing a desired number of children for each element T_i by a formula that is based on the *a priori* error estimate in Equation 5. The discrete problem is then solved on the intermediate grid. The third step is to perform adaptive p-refinement to reduce the error to the desired tolerance τ. Again, a formula is used to determine the new degree for each element, p-refinement is performed to increase the degree of each element to the desired degree, and the discrete problem is solved on the final grid.

The strategy of performing all the h-refinement in one step and all the p-refinement in one step is adequate for low accuracy solutions (e.g. 1%), but is not likely to work well with high accuracy solution (e.g. 10^{-8}

relative error) [26]. We extend the Texas 3 Step strategy to high accuracy by cycling through steps 2 and 3 until the final tolerance τ_{final} is met. τ in the algorithm above is now the factor by which one cycle of steps 2 and 3 should reduce the error. Toward this end, before step 2 the error estimate η_0 is computed for the current grid. The final (for this cycle) and intermediate targets are now given by $\eta_T = \tau\eta_0$ and $\eta_I = \gamma\eta_T$. In the results of Section 5 we use $\tau = 0.1$ and $\gamma = 6$.

4.6 Alternate h and p

This strategy, which will be referred to as ALTERNATE, is a variation on T3S that is more like the algorithm of Figure 1. The difference from T3S is that instead of predicting the number of refinements needed to reduce the error to the next target, the usual adaptive refinement is performed until the target is reached. Thus in step 2 all elements with an error indicator larger than $\eta_I/\sqrt{N_0}$ are h-refined. The discrete problem is solved and the new error estimate compared to η_I. This is repeated until the error estimate is smaller than η_I. Step 3 is similar except adaptive p-refinement is performed and the target is η_T. Steps 2 and 3 are repeated until the final error tolerance is achieved.

4.7 Nonlinear Programming

Patra and Gupta [27] proposed a strategy for hp mesh design using nonlinear programming. We refer to this strategy as NLP. They presented it in the context of quadrilateral elements with one level of hanging nodes, i.e., an element edge is allowed to have at most one hanging node. Here it is modified slightly for newest node bisection of triangles with no hanging nodes. This is another approach that does not strictly follow the algorithm in Figure 1.

Given a current grid with triangles $\{T_i\}$, degrees $\{p_i\}$, h-refinement levels $\{l_i\}$, and error estimates $\{\eta_i\}$, the object is to determine new mesh parameters $\{\hat{p}_i\}$ and $\{\hat{l}_i\}$, $i = 1..N_T$, by solving an optimization problem which can be informally stated as: minimize the number of degrees of freedom subject to the error being less than a given tolerance and other constraints. Computationally, the square of the error is approximated by $\sum_{i=0}^{N_T} \hat{\eta}_i^2$ where $\hat{\eta}_i$ is an estimate of the error in the refined grid over the region covered by T_i, and the number of degrees of freedom over the children of T_i is $2^{\hat{l}_i - l_i}\hat{p}_i^2/2$. Thus the optimization problem is

$$\underset{\{\hat{l}_i\},\{\hat{p}_i\}}{\text{minimize}} \quad \sum_{i=1}^{N_T} 2^{\hat{l}_i - l_i}\frac{\hat{p}_i^2}{2} \tag{11}$$

$$\text{s.t.} \quad \sum_{i=1}^{N_T} \hat{\eta}_i^2 < \hat{\tau}^2 \tag{12}$$

$$\hat{l}_j - 1 \le \hat{l}_i \le \hat{l}_j + 1 \;\; \forall j \text{ such that } T_j \text{ shares an edge with } T_i \tag{13}$$

$$0 \le \hat{l}_i \le l_{max} \tag{14}$$

$$1 \le \hat{p}_i \le p_{max} \tag{15}$$

$$l_i - \Delta l_{\text{dec}} \le \hat{l}_i \le l_i + \Delta l_{\text{inc}} \tag{16}$$

$$p_i - \Delta p_{\text{dec}} \le \hat{p}_i \le p_i + \Delta p_{\text{inc}} \tag{17}$$

where $\hat{\tau}$ is the error tolerance for this refinement phase. Equation 13 is a necessary condition for compatibility of the grid (in [27] it enforces one level of hanging nodes). Equation 14 insures that coarsening does not go beyond the initial grid, and that the refinement level of an element does not exceed a prescribed limit l_{max}.

Similarly, Equation 15 insures that element degrees do not go below one or exceed a prescribed limit p_{max}. Also, because many quantities are only approximate, it is wise to limit the amount of change that occurs to any element during one phase of refinement. Equations 16 and 17 restrict the amount of change that can occur at one time.

Since the $\hat{l}_i$ and $\hat{p}_i$ are naturally integers, the optimization problem is a mixed integer nonlinear program, which is known to be NP-hard. To make the problem tractable, the integer requirement is dropped to give a nonlinear program which can be solved by one of several software packages. For the results in Section 5, we used the program ALGENCAN [2] Version 2.2.1 [5, 9]. Following solution of the nonlinear program, the $\hat{l}_i$ and $\hat{p}_i$ are rounded to the nearest integer.

4.8 Predict Error Estimate on Assumption of Smoothness

Melenk and Wohlmuth [17] proposed a strategy based on a prediction of what the error should be if the solution is smooth. We call this strategy SMOOTH_PRED.

When refining element T_i, assume the solution is locally smooth and that the optimal convergence rate is obtained. If h-refinement is performed and the degree of T_i is p_i, then the expected error on the two children of T_i is reduced by a factor of $\sqrt{2}^{\,p_i}$ as indicated by the *a priori* error estimate of Equation 5. Thus if η_i is the error estimate for T_i, predict the error estimate of the children to be $\gamma_h \eta_i / \sqrt{2}^{\,p_i}$ where γ_h is a user specified parameter. If p-refinement is performed on T_i, exponential convergence is expected, so the error should be reduced by some constant factor $\gamma_p \in (0, 1)$, i.e., the predicted error estimate is $\gamma_p \eta_i$. When the actual error estimate of a child becomes available, it is compared to the predicted error estimate. If the error estimate is less than or equal to the predicted error estimate, then p-refinement is indicated for the child. Otherwise, h-refinement is indicated since presumably the assumption of smoothness was wrong. For the results in Section 5 we use $\gamma_h = 2$ and $\gamma_p = \sqrt{0.4}$.

4.9 Larger of h-Based and p-Based Error Indicators

In 1D, Schmidt and Siebert [29] proposed a strategy that uses two *a posteriori* error estimates to predict whether h-refinement or p-refinement will reduce the error more. We extend this strategy to bisected triangles and refer to it as H&P_ERREST.

The local Neumann residual error estimate given by Equations 7-10 is actually an estimate of how much the norm of the solution will change if T_i is p-refined. This is because the solution of the local problem is estimated using the p-hierarchical bases that would be added if T_i is p-refined, so it is an estimate of the actual change that would occur. Using the fact that the current space is a subspace of the refined space and Galerkin orthogonality, it can be shown that

$$||u - \hat{u}_{hp}||^2 = ||u - u_{hp}||^2 - ||\hat{u}_{hp} - u_{hp}||^2$$

where $\hat{u}_{hp}$ is the solution in the refined space. Thus the change in the solution indicates how much the error will be reduced.

[2] The mention of specific products, trademarks, or brand names is for purposes of identification only. Such mention is not to be interpreted in any way as an endorsement or certification of such products or brands by the National Institute of Standards and Technology. All trademarks mentioned herein belong to their respective owners.

A second error estimate for T_i can be computed by solving a local Dirichlet problem

$$\mathcal{L}e_i = f - \mathcal{L}u_{hp} \qquad \text{in } T_i \cup T_i^{\text{mate}} \tag{18}$$

$$e_i = g_D - u_{hp} \qquad \text{on } \partial(T_i \cup T_i^{\text{mate}}) \cap \partial\Omega_D \tag{19}$$

$$\mathcal{B}e_i = g_N - \mathcal{B}u_{hp} \qquad \text{on } \partial(T_i \cup T_i^{\text{mate}}) \cap \partial\Omega_N \tag{20}$$

$$e_i = 0 \qquad \text{on } \left(\partial(T_i \cup T_i^{\text{mate}}) \setminus \partial\Omega_D\right) \setminus \partial\Omega_N \tag{21}$$

where T_i^{mate} is the element that is refined along with T_i in the newest node bisection method [20]. The solution to this problem is approximated by an h-refinement of the two elements using only the new basis functions. The error estimate obtained by taking the norm of this approximate solution is actually an estimate of how much the solution will change, or the error will be reduced, if h-refinement is performed.

Schmidt and Siebert divide the two error estimates by the associated increase in the number of degrees of freedom to obtain an approximate error reduction per degree of freedom. In addition or instead, one of the error estimates can be multiplied by a user specified constant to bias the refinement toward h- or p-refinement. In the results of Section 5 the p-based error estimate is multiplied by 2.

The type of refinement that is used is the one that corresponds to the larger of the two modified error estimates.

4.10 Legendre coefficient strategies

We consider two hp-adaptive strategies that are based on the coefficients in an expansion of the solution in Legendre polynomials. In one dimension, the approximate solution in element T_i with degree p_i can be written

$$u_i(x) = \sum_{j=0}^{p_i} a_j P_j(x)$$

where P_j is the j^{th} degree Legendre polynomial scaled to the interval of element T_i.

Mavriplis [16] estimates the decay rate of the coefficients by a least squares fit of the the last four coefficients a_j to $Ce^{-\sigma j}$. Elements are refined by p-refinement where $\sigma > 1$ and by h-refinement where $\sigma \leq 1$. We refer to this strategy as COEF_DECAY.

Houston et al. [15] present an approach which uses the Legendre coefficients to estimate the regularity of the solution. The regularity is estimated using the root test yielding

$$m_i = \frac{\log\left(\frac{2p_i+1}{2a_{p_i}^2}\right)}{2\log p_i}.$$

If $p_i = 1$, use p-refinement. Otherwise, use p-refinement if $p_i \leq m_i - 1$ and h-refinement if $p_i > m_i - 1$. We refer to this strategy as COEF_ROOT.

Both Mavriplis and Houston et al. presented the strategies in the context of one dimension and use the Legendre polynomials as the local basis so the coefficients are readily available. In [15] it is extended to 2D for rectangular elements with a tensor product of Legendre polynomials, and the regularity is estimated in each dimension separately, so the coefficients are still readily available. Eibner and Melenk [12] extended the COEF_DECAY strategy to quadrisected triangles with an orthogonal polynomial basis. In this study we are using triangular elements which have a basis that is based on Legendre polynomials [33]. In this basis there are $3 + \max(j-2, 0)$ basis functions of exact degree j over an element, so we don't have a single Legendre

polynomial coefficient to use. Instead, for the coefficients a_j we use the ℓ_1 norm of the coefficients of the degree j basis functions, i.e.

$$a_j = \sum_{\substack{k \text{ s.t. } \deg(\phi_k)=j \\ \text{supp}(\phi_k)\cap T_i \neq \emptyset}} |\alpha_k|$$

4.11 Reference Solution Strategies

Demkowicz and his collaborators developed an *hp*-adaptive strategy over a number of years, presented in several papers and books, e.g. [10, 11, 28, 31]. We refer to this strategy as REFSOLN_EDGE because it relies on computing a reference solution and bases the refinement decisions on edge refinements. Note that for this strategy the basic form of the *hp*-adaptive algorithm is different than that in Figure 1.

The algorithm is first presented for 1D elliptic problems. Given the current existing mesh, $G_{h,p}$, and current solution, $u_{h,p}$, a uniform refinement in both h and p is performed to obtain a fine mesh $G_{h/2,p+1}$. The equation is solved on the fine mesh to obtain a reference solution $u_{h/2,p+1}$.

The next step is to determine the optimal refinement of each element. This is done by considering a p-refinement and all possible (bisection) h-refinements (i.e., all possible assignments of p to the two children of an h-refinement) that give the same increase in the number of degrees of freedom as the p-refinement. In 1D, this means that the sum of the degrees of the two children must be $p+1$, resulting in a total of p h-refinements and one p-refinement to be examined. For each possibility, the error decrease rate is computed as

$$\frac{|u_{h/2,p+1} - \Pi_{hp,i} u_{h/2,p+1}|^2_{H^1(T_i)} - |u_{h/2,p+1} - \Pi_{new,i} u_{h/2,p+1}|^2_{H^1(T_i)}}{N_{new} - N_{hp}}$$

where $\Pi_{hp,i} u_{h/2,p+1}$ is the projection-based interpolant of the reference solution in element T_i, and $\Pi_{new,i}$ is the projection onto the resulting elements from any one of the candidate refinements. The refinement with the largest error decrease rate is selected as the optimal refinement. In the case of h-refinement, the degrees may be increased further by a process known as following the biggest subelement error refinement path, which is also used to determine the guaranteed element rate; see [10] for details.

Elements that have a guaranteed rate larger than 1/3 the maximum guaranteed rate are selected for refinement; the factor 1/3 is arbitrary.

The 2D algorithm also begins by computing a reference solution on the globally *hp*-refined grid $G_{h/2,p+1}$. Then for each edge in the grid, the choice between p- and h-refinement, the determination of the guaranteed edge rate, and the selection of edges to refine are done exactly as in 1D, except that a weighted H^1 seminorm is used instead of the more natural $H^{1/2}$ seminorm which is difficult to work with. In the case of bisected triangles, we only consider edges that would be refined by the bisection of an existing triangle.

The h-refinement of edges determines the h-refinement of elements. It remains to determine the degree of each element. As a starting point, element degrees are assigned to satisfy the minimum rule for edge degrees, using the edge degrees determined in the previous step. Then the biggest subelement error refinement path is followed to determine the guaranteed element rate and assignment of element degrees. We again refer to [10] for details. Finally, the minimum rule for edge degrees is enforced.

A related, but simpler, approach was developed by Šolín et al. [30]. We refer to this strategy as REFSOLN_ELEM since it also begins by computing the reference solution, but bases the refinement on elements. The basic form of the *hp*-adaptive algorithm is different than that in Figure 1 for this strategy, also.

The local error estimate is given by the norm of the difference between the reference solution and the current solution,

$$\eta_i = ||u_{h/2,p+1} - u_{h,p}||_{H^1(T_i)}$$

and the elements with the largest error estimates are refined. If T_i is selected for refinement, let $p_0 = \lfloor (p_i + 1)/2 \rfloor$ and consider the following options:

- p-refine T_i to degree $p_i + 1$,
- p-refine T_i to degree $p_i + 2$,
- h-refine T_i and consider all combinations of degrees p_0, $p_0 + 1$ and $p_0 + 2$ in the children.

In all cases the minimum rule is used to determine edge degrees. In [30], quadrisection of triangles is used leading to 83 options to consider. With bisection of triangles, there are only 11 options. Also, since the object of dividing by two to get p_0 is to make the increase in degrees of freedom from h-refinement comparable to that of p-refinement, we use $p_0 = \lfloor (p_i + 1)/\sqrt{2} \rfloor$ since there are only two children instead of four.

For each candidate, the standard H^1 projection $\Pi^{H^1(T_i)}_{\text{candidate}}$ of $u_{h/2,p+1}$ onto the corresponding space is performed, and the projection error in the H^1 norm, $\zeta_{\text{candidate}}$, is computed,

$$\zeta_{\text{candidate}} = ||u_{h/2,p+1} - \Pi^{H^1(T_i)}_{\text{candidate}} u_{h/2,p+1}||_{H^1(T_i)}$$

as well as the projection error of the projection onto T_i, ζ_i.

Let N_i be the number of degrees of freedom in the space corresponding to T_i, and $N_{\text{candidate}}$ the number of degrees of freedom in the space corresponding to a candidate. After discarding candidates that seem to be outliers, select the candidate that maximizes

$$\frac{\log \zeta_i - \log \zeta_{\text{candidate}}}{N_{\text{candidate}} - N_i} \tag{22}$$

Following the refinement that is indicated by the selected candidate, the minimum rule for edge degrees is applied.

This algorithm can be modified slightly to bias the refinement towards or away from p refinement to improve the performance. Given a parameter p_{bias}, multiply the value from Equation 22 by it for all the p-refinement candidates. $p_{\text{bias}} > 1$ will bias the refinement toward doing p-refinement, and $p_{\text{bias}} < 1$ will bias the refinement toward doing h-refinement. For the results in Section 5 we use $p_{\text{bias}} = 2$ for most problems, and $p_{\text{bias}} = 4$ for the analytic, mild wave front and both peak problems, which are the easiest problems.

5 Numerical Results

This section contains the results of a numerical experiment to compare the hp-adaptive strategies' performance on a suite of 21 test problems with various difficulties that adaptive refinement should locate. The primary criteria for comparing the strategies is the convergence of the relative error in the energy norm as a function of the number of degrees of freedom, N. The test problems and convergence results for each problem are given in Section 5.1, and summary results for comparison of the strategies are given in Section 5.3. We also give some indication of the relative amount of time required to obtain the solution in Section 5.2.

The full details of the test problems can be found in the companion paper [21]. Here we just give a brief description of each problem. Recall that Poisson's equation is $u_{xx} + u_{yy} = f(x, y)$ and Laplace's equation is Poisson's equation with $f = 0$.

Each problem is solved with each hp strategy using the hp-adaptive algorithm of Section 3, except for those strategies that dictate using a variation on that algorithm, as indicated in Section 4. To examine

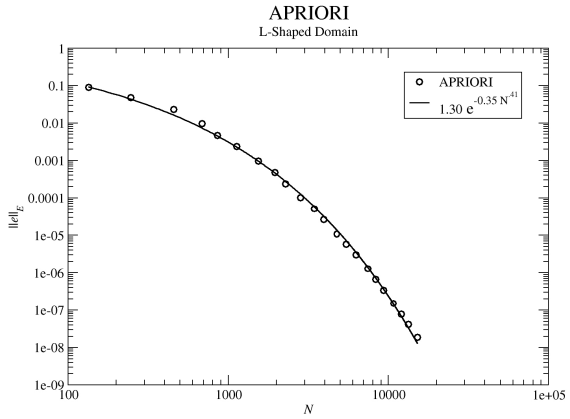


Figure 2: The 3-parameter least squares fit to the convergence data for a typical example.

the convergence of the error as a function of N, each problem is solved using each strategy several times with different values of the termination tolerance τ. The relative energy norm of the error and N are recorded at the end of each run to give a set of points for the convergence data. In most cases we used $\tau = 0.1,\ 0.05,\ 0.02,\ 0.01,\ 0.005\ \ldots\ 2\text{x}10^{-8},\ 10^{-8}$, although some of the more difficult problems required ending the sequence earlier.

In Section 5.1 we give convergence curves of all strategies on one graph for each problem as N vs. the relative energy norm of the error on a log-log scale. The curves are an exponential least squares fit to the data. According to Equation 6 the error should converge like $Ae^{BN^{1/3}}$. In the experiments, the data often exhibits exponential convergence, but with a different exponent on N than 1/3. The curves are a least squares fit to the form Ae^{BN^C}. This 3-parameter least squares fit will be the primary means of comparing the performance of the strategies. Figure 2 illustrates the relationship between the 3-parameter least squares fit (the curve) and the data points (the circles) in a typical example.

Space limitations restrict the presentation of the results to this one form in this paper. A technical report [22] contains a more extensive presentation of the results. In particular, it has

- an image of an example grid for each strategy with each problem, as illustrated in Figure 3,
- the log-log convergence plot for each strategy with each problem, including the data points, 3-parameter least squares fit, and a 2-parameter least squares fit where the exponent on N is set to 1/3,
- cube root of N vs. logarithm of error convergence plots for one problem (the curve is theoretically a straight line with that scale),
- tables of the parameters obtained by the 3-parameter and 2-parameter least squares fits,
- tables ranking the methods according to the 3-parameter fit for each problem at low and high accuracy, and

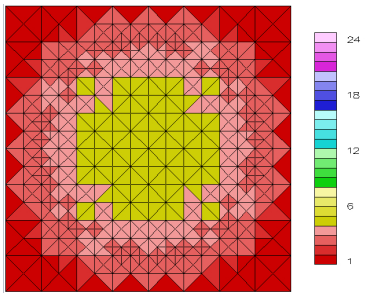


Figure 3: Example image of a grid. The color indicates the degree of the polynomial over each element.

- tables with the factor, for each strategy, by which N is larger than the best strategy for each problem at low and high accuracy.

These computations were performed using the adaptive finite element code PHAML Version 1.8.1 [18] on a single processor. During the period of this investigation there were changes to the available hardware and software, but we do not believe any of these changes would effect the outcome of these computations, except in Section 5.2 where a consistent computational environment is used. The computers were 32-bit and 64-bit x86-class computers operating under CentOS 5.x distributions of Linux. PHAML was compiled with the Intel Fortran compiler.

5.1 Test Problems and Convergence Graphs

Analytic Solution. The analytic problem in [21] is Poisson's equation on the unit square with Dirichlet boundary conditions. The solution is the polynomial

$$2^{4p} x^p (1-x)^p y^p (1-y)^p$$

with $p = 10$. 2^{4p} is a normalization factor so that the L^∞ norm is 1.0. For the APRIORI strategy, we choose to always refine by p, i.e., it is just p-adaptive refinement.

Reentrant Corner, Nearly Straight. For elliptic partial differential equations, a reentrant (concave) corner in the domain, with interior angle ω, causes a point singularity that behaves like r^α where r is the distance from the corner and $\alpha = \pi/\omega$. The larger ω is, the stronger the singularity. The reentrant corner problems of the next five sections are Laplace's equation with Dirichlet boundary conditions on $(-1,1) \times (-1,1)$ with a section of angle $2\pi - \omega$ removed. The solution is

$$r^\alpha \sin(\alpha\theta)$$

where $r = \sqrt{x^2 + y^2}$ and $\theta = \tan^{-1}(y/x)$.

For the nearly straight reentrant corner, $\omega = \pi + .01$. This is a very mild singularity. In all the reentrant corner test problems, the APRIORI strategy refines by h if the element contains the origin and by p otherwise.

Reentrant Corner, Wide Angle. This is the reentrant corner problem with $\omega = 5\pi/4$.

Reentrant Corner, L-Shaped Domain. The reentrant corner problem with $\omega = 3\pi/2$ is the classic "L domain" problem which is used as an example in many papers on adaptive grid refinement.

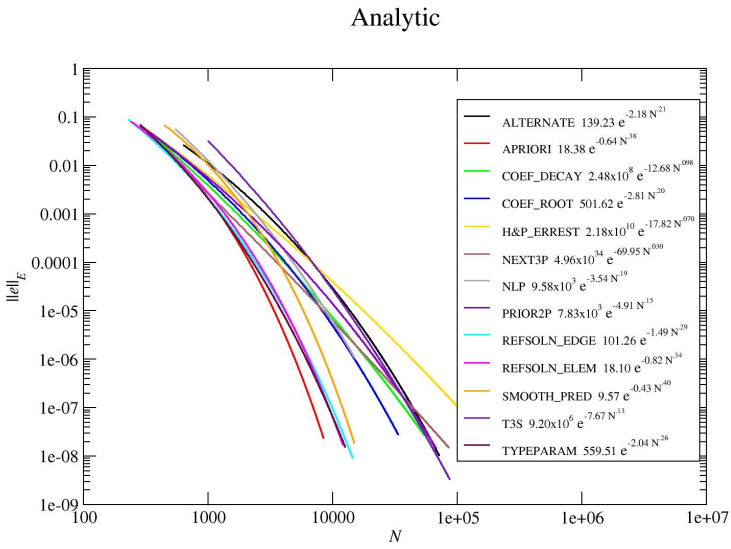


Figure 4: Log-Log plot of the convergence of all strategies with the analytic problem.

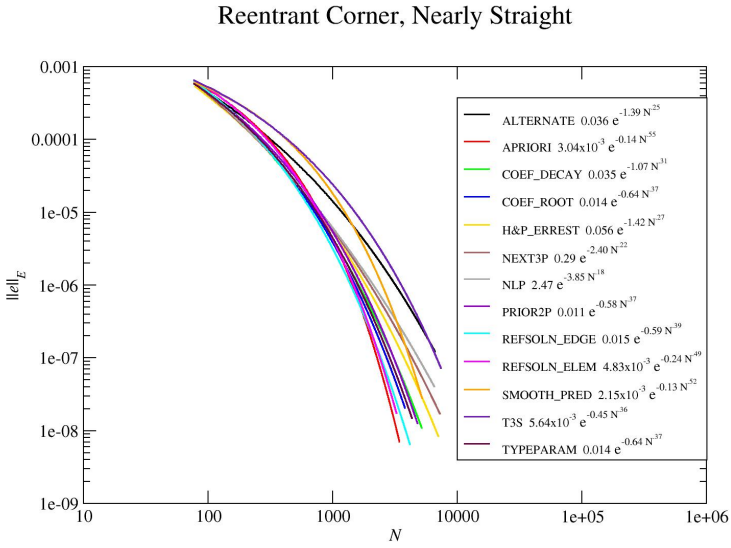


Figure 5: Log-Log plot of the convergence of all strategies with the nearly straight reentrant corner problem.

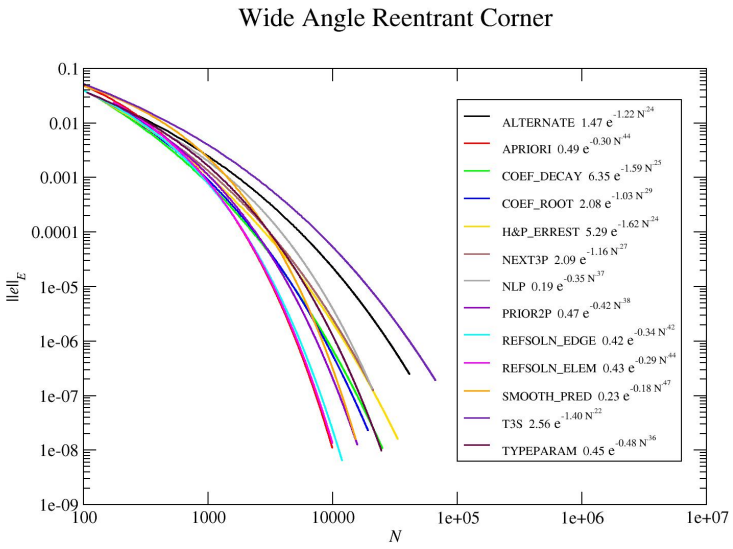


Figure 6: Log-Log plot of the convergence of all strategies with the wide angle reentrant corner problem.

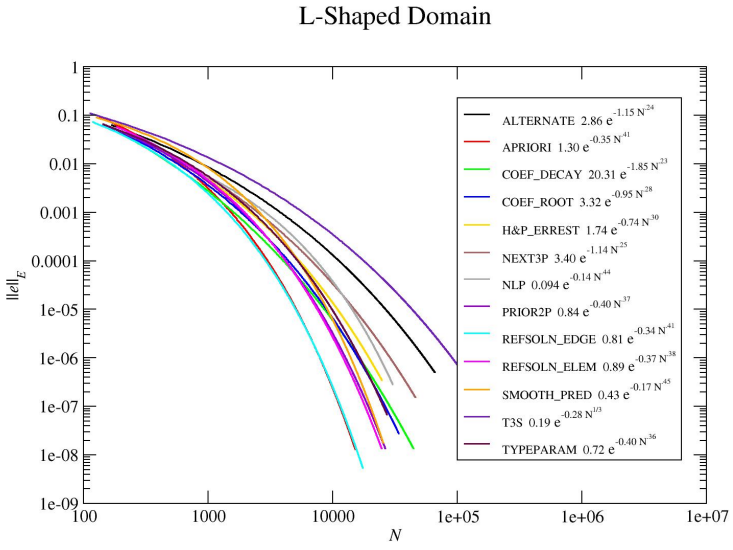


Figure 7: Log-Log plot of the convergence of all strategies with the L-shaped domain problem.

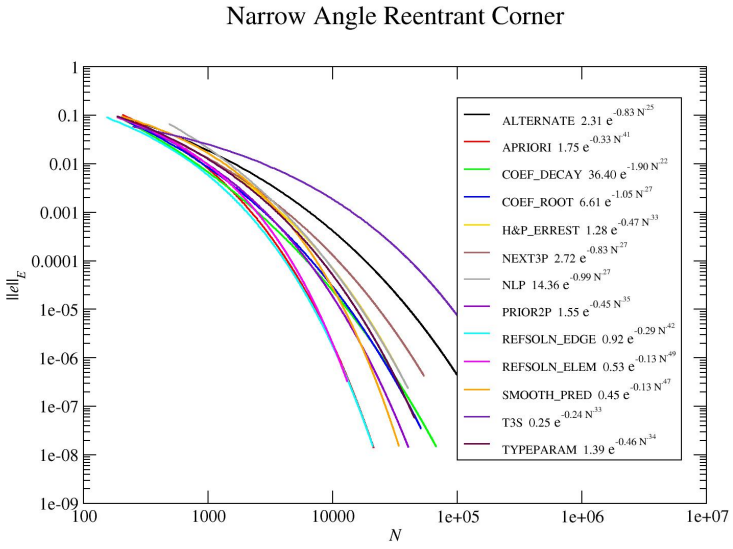


Figure 8: Log-Log plot of the convergence of all strategies with the narrow angle reentrant corner problem.

Reentrant Corner, Narrow Angle. This is the reentrant corner problem with $\omega = 7\pi/4$.

Reentrant Corner, Slit. This is the reentrant corner problem with $\omega = 2\pi$. This results in a domain that has a slit along the positive x axis.

Linear Elasticity, Mode 1. The linear elasticity problem is a coupled system of two equations with a mixed derivative in the coupling term and different coefficients on the second order x and y terms. The domain is a square with a slit, as in the reentrant corner slit domain problem. The boundary conditions are Dirichlet. For further details, see [21]. We consider two solutions, referred to as mode 1 and mode 2, by using different boundary conditions. Both solutions have a singularity at the origin, with the mode 1 solution having the stronger singularity. For both problems, the APRIORI strategy refines by h if the element contains the origin and by p otherwise.

Linear Elasticity, Mode 2. This is the mode 2 solution of the linear elasticity problem.

Mild Peak. The peak problem contains a Gaussian peak in the interior of the domain. It is Poisson's equation on the unit square with Dirichlet boundary conditions. The solution is

$$e^{-\alpha((x-x_c)^2+(y-y_c)^2)}$$

where (x_c, y_c) is the location of the peak, and α determines the strength of the peak. For the easy form of this problem, we use $\alpha = 1000$ and $(x_c, y_c) = (0.5, 0.5)$. The APRIORI strategy refines by h if the element touches the center of the peak and by p otherwise.

Sharp Peak. This is the hard version of the peak problem with $\alpha = 100000$ and $(x_c, y_c) = (.51, .117)$.

Battery. The battery problem is from a model of heat conduction in a battery with nonhomogeneous materials. It has piecewise constant coefficients and right hand side, and mixed boundary conditions on a

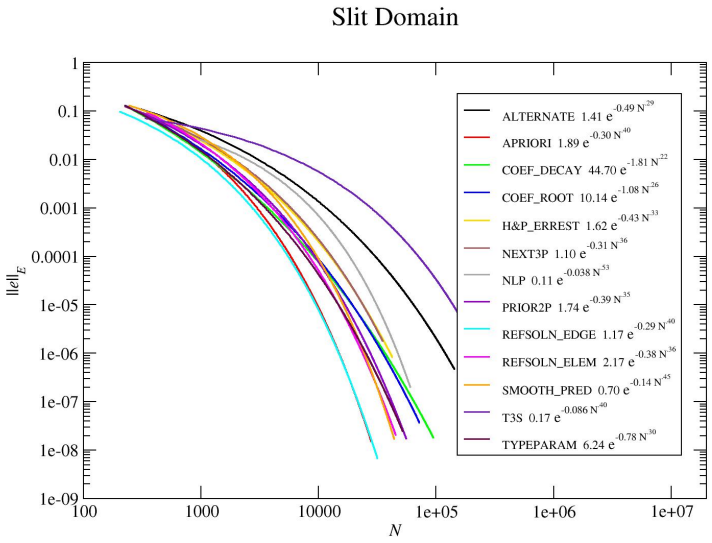


Figure 9: Log-Log plot of the convergence of all strategies with the slit domain problem.

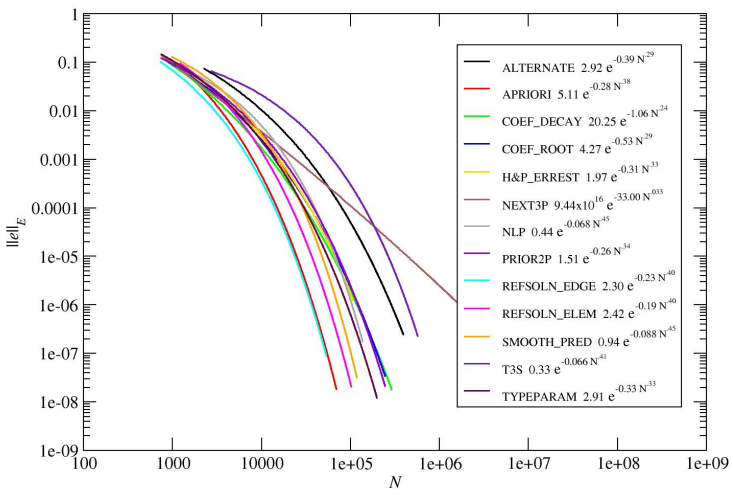


Figure 10: Log-Log plot of the convergence of all strategies with the mode 1 linear elasticity problem.

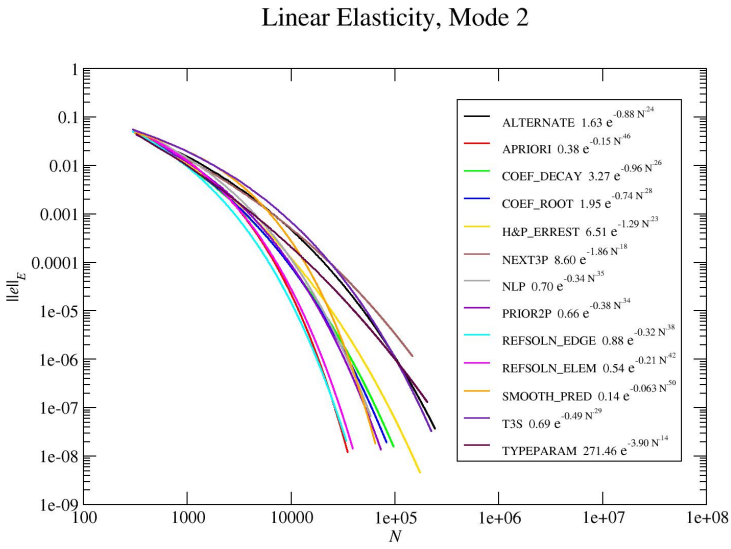


Figure 11: Log-Log plot of the convergence of all strategies with the mode 2 linear elasticity problem.

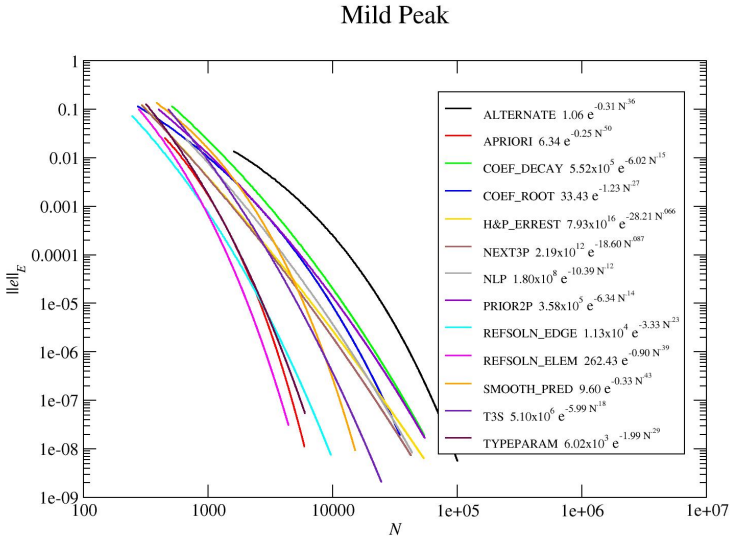


Figure 12: Log-Log plot of the convergence of all strategies with the mild peak problem.

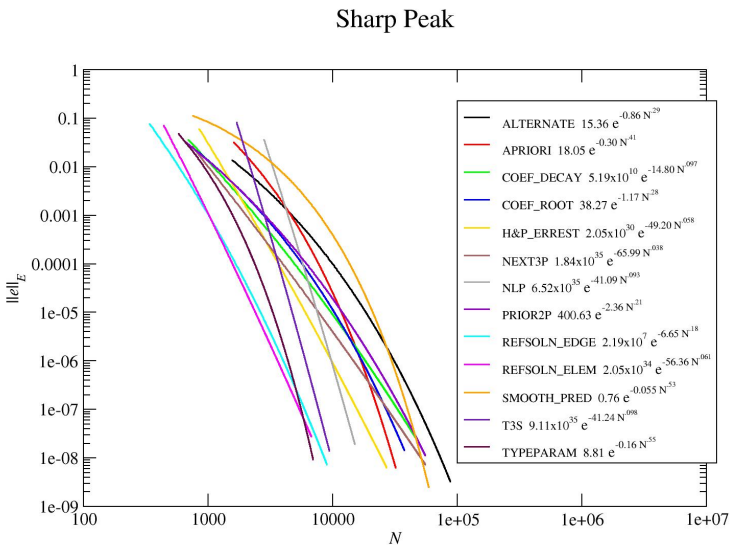


Figure 13: Log-Log plot of the convergence of all strategies with the sharp peak problem.

rectangular domain. The solution has several point singularities in the interior of the domain where three or more materials meet. See [21] for further details. The exact solution of this problem is not known, so the error estimate η (Section 3) is used for the convergence results instead of the error. The APRIORI strategy refines by h if the element touches any of the singularities, and by p otherwise.

Boundary Layer, Mild. The boundary layer problem is a convection-diffusion equation with first order terms and Dirichlet boundary conditions on $(-1, 1) \times (-1, 1)$. The solution is

$$(1 - e^{-(1-x)/\epsilon})(1 - e^{-(1-y)/\epsilon}) \cos(\pi(x + y))$$

where ϵ controls the strength of the boundary layer. In the easy form of this problem we use $\epsilon = 10^{-1}$. In the APRIORI strategy we refine by h if the element touches either of the boundaries with a boundary layer, and by p otherwise.

Boundary Layer, Strong. For the hard version of the boundary layer problem we use $\epsilon = 10^{-3}$.

Oscillatory, Mild. The oscillatory problem contains several circular waves which get closer together as you approach the origin. The PDE is a Helmholtz equation with Dirichlet boundary conditions on the unit square. The solution is

$$\sin(\frac{1}{\alpha + r})$$

where $r = \sqrt{x^2 + y^2}$. The number of oscillations, N, is determined by the parameter $\alpha = \frac{1}{N\pi}$. For the easy form of this problem we use $N = 10.5$. For APRIORI, refine by h if the element touches the origin and by p otherwise.

Oscillatory, Strong. For the strong version of the oscillatory problem we use $N = 50.5$.

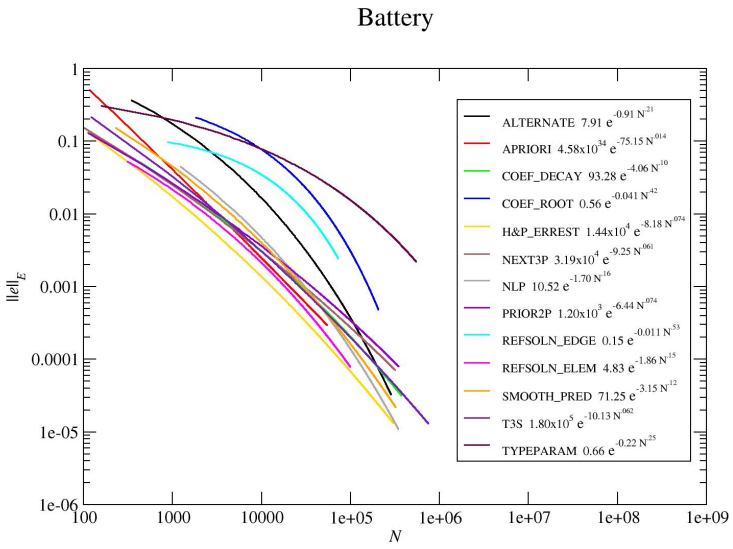


Figure 14: Log-Log plot of the convergence of all strategies with the battery problem.

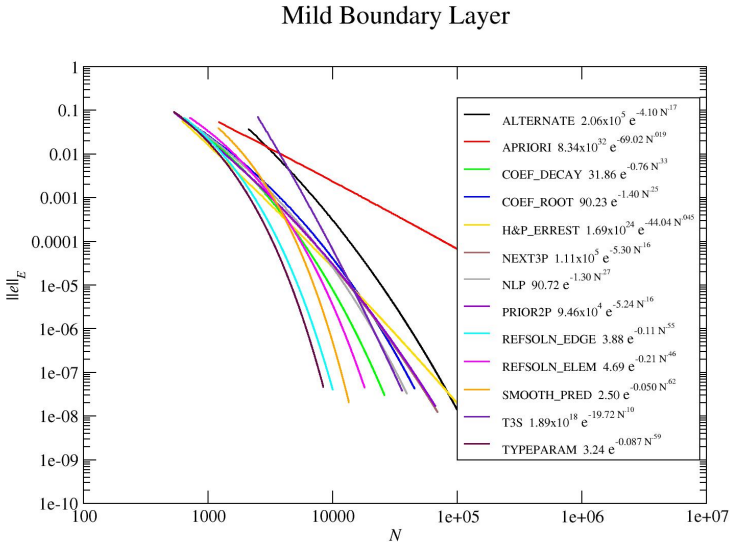


Figure 15: Log-Log plot of the convergence of all strategies with the mild boundary layer problem.

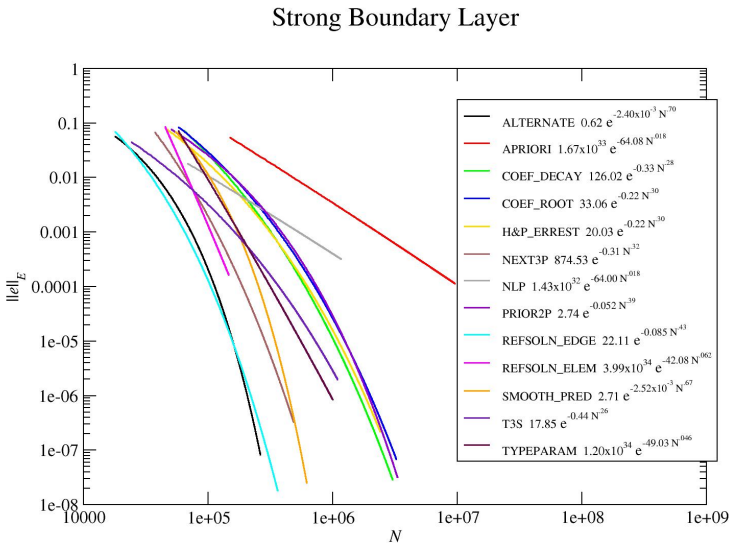


Figure 16: Log-Log plot of the convergence of all strategies with the strong boundary layer problem.

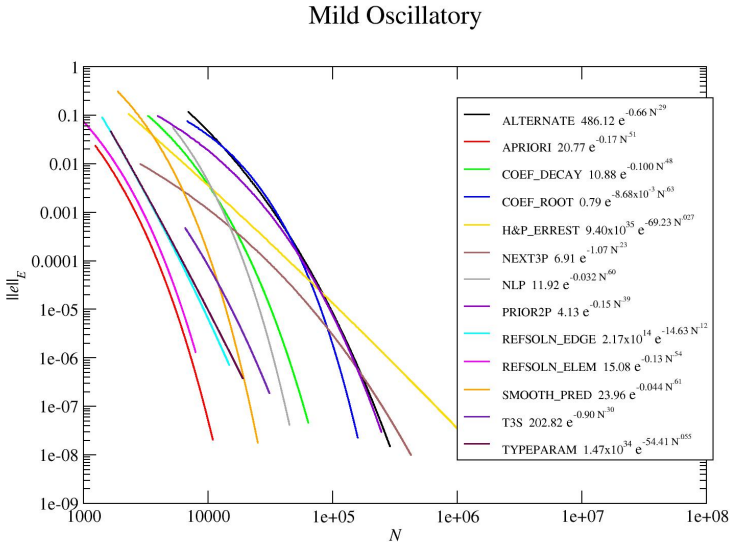


Figure 17: Log-Log plot of the convergence of all strategies with the mild oscillatory problem.

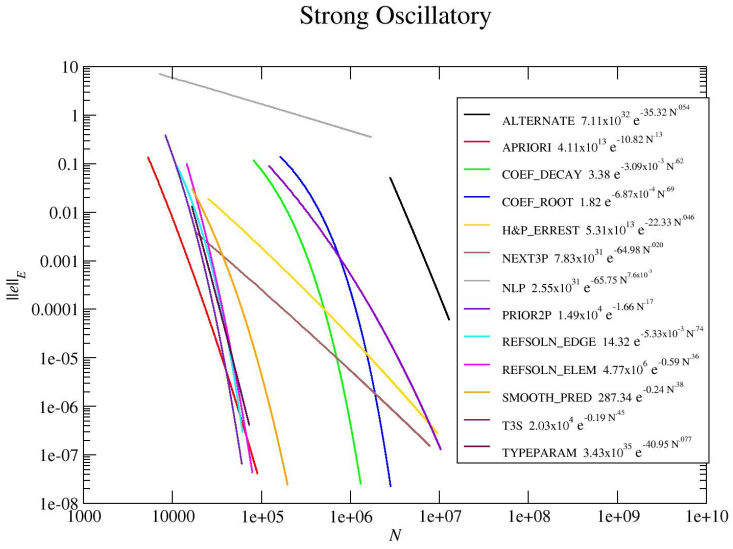


Figure 18: Log-Log plot of the convergence of all strategies with the strong oscillatory problem.

Wave Front, Mild. The circular wave front problem is often used as an example in adaptive grid refinement papers. It is Poisson's equation with Dirichlet boundary conditions on the unit square. The solution is

$$\tan^{-1}(\alpha(r - r_0))$$

where $r = \sqrt{(x - x_c)^2 + (y - y_c)^2}$. The location of the wave front is defined by a circle with radius r_0 and center (x_c, y_c). α determines the steepness of the wave front. In addition to the wave front, the solution has a mild singularity at the center of the circle, if the center is located in the closure of the domain. For the easy form of this problem we use $\alpha = 20$, $(x_c, y_c) = (-.05, -.05)$, and $r_0 = 0.7$. The center is chosen outside the domain so that only the wave front is a factor in the adaptivity, not the singularity. With all the wave front problems, for the APRIORI strategy, refine by h if the element touches the circle that defines the location of the wave front and has degree at least 3 (chosen arbitrarily, but works better than degree 1), and by p otherwise.

Wave Front, Steep. In the hard version of the wave front problem the location of the wave front is the same, but it is much steeper. The parameters are $\alpha = 1000$, $(x_c, y_c) = (-.05, -.05)$, and $r_0 = 0.7$.

Wave Front, Asymmetric. The asymmetric wave front is similar to the steep wave front except the wave front is not symmetric within the domain. The parameters are $\alpha = 1000$, $(x_c, y_c) = (1.5, .25)$, and $r_0 = .92$.

Singular Well. This is the wave front problem with the center of the circle placed at the center of the domain and a relatively mild wave front, effectively creating a well with a mild singularity at the center. $\alpha = 50$, $(x_c, y_c) = (.5, .5)$, and $r_0 = .25$. For the APRIORI strategy, refine by h if the element touches the circle that defines the location of the wave front and has degree at least 3, or touches the center of the circle,

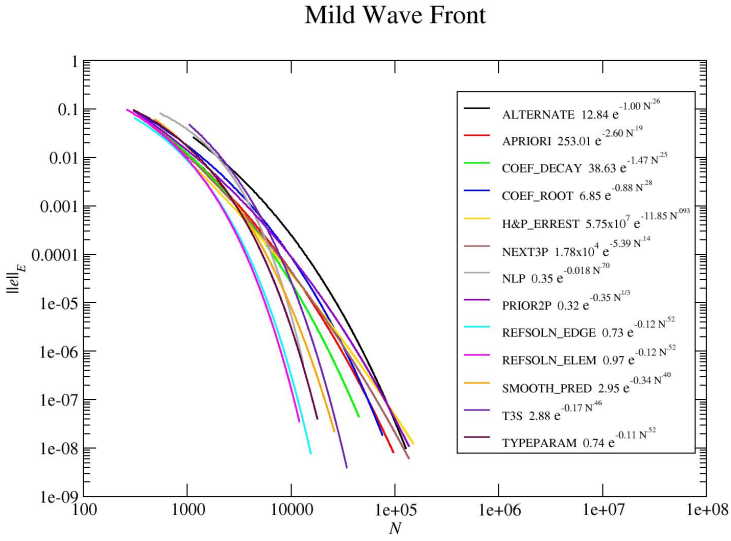


Figure 19: Log-Log plot of the convergence of all strategies with the mild wave front problem.

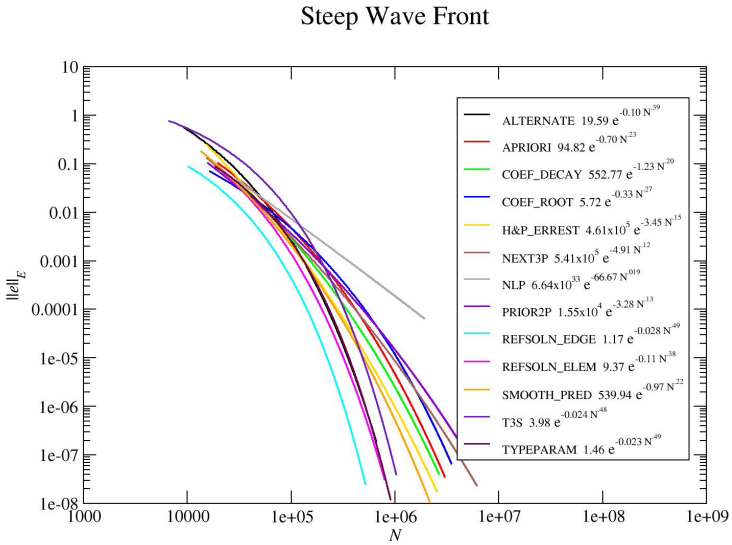


Figure 20: Log-Log plot of the convergence of all strategies with the steep wave front problem.

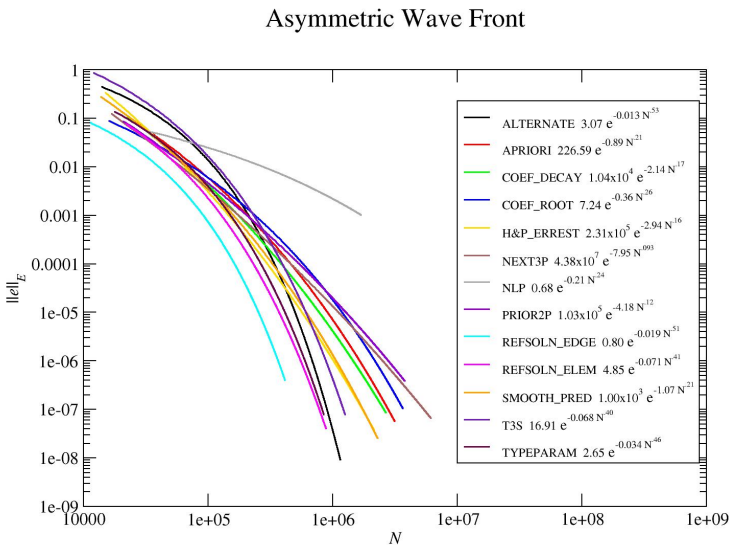


Figure 21: Log-Log plot of the convergence of all strategies with the asymmetric wave front problem.

and by p otherwise.

Intersecting Interfaces. The intersecting interfaces problem has piecewise constant coefficients which create a very strong singularity at the center of the domain and discontinuous derivatives along the x and y axes. The boundary conditions are Dirichlet on the domain $(-1,1)\times(-1,1)$. For the APRIORI strategy, refine by h if the element touches the origin and by p otherwise.

Multiple Difficulties. The multiple difficulties problem combines several of the difficulties of the other problems into a single problem. It contains a reentrant corner point singularity, wave front, peak and boundary layer. For the selected parameters, the peak falls on the wave front, and the wave front intersects the boundary layer and point singularity. The parameters are:

- reentrant corner $\omega = 3\pi/2$
- center of circle for wave front $(0,-3/4)$
- radius of circle for wave front $3/4$
- strength of wave front $\alpha = 200$
- center of peak $(\sqrt{5}/4,-1/4)$
- strength of peak $\alpha = 1000$
- strength of boundary layer $\epsilon = 1/100$

The APRIORI method refines by h in the same cases as it did in the individual problems.

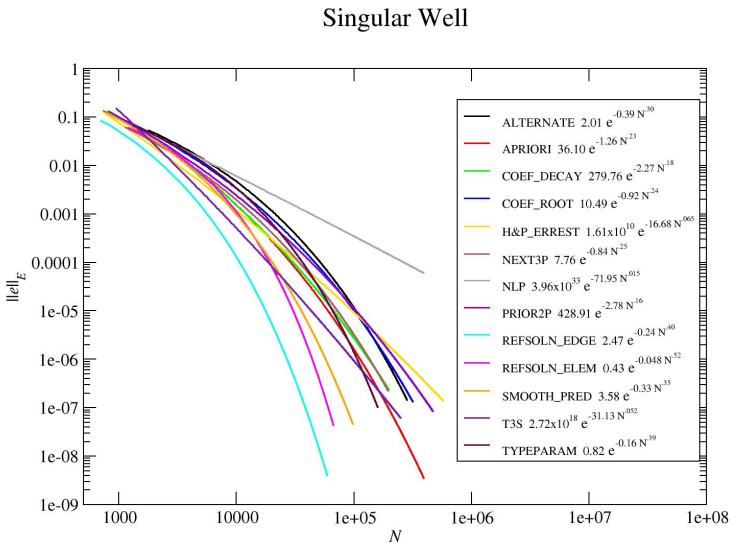


Figure 22: Log-Log plot of the convergence of all strategies with the singular well problem.

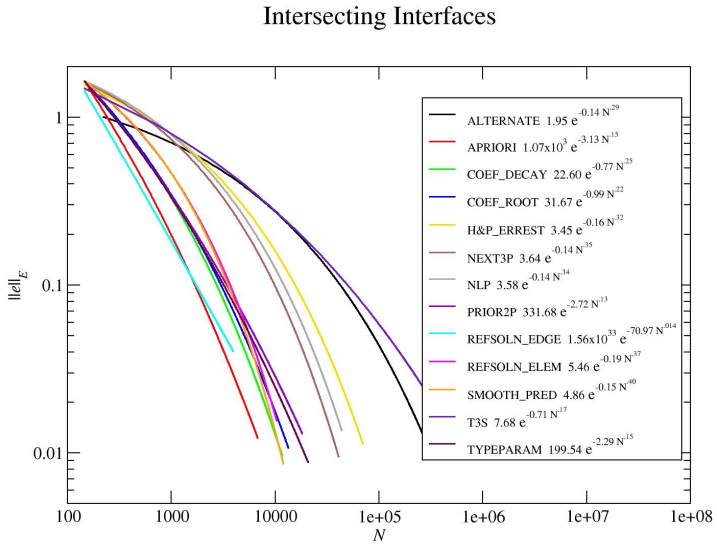


Figure 23: Log-Log plot of the convergence of all strategies with the intersecting interfaces problem.

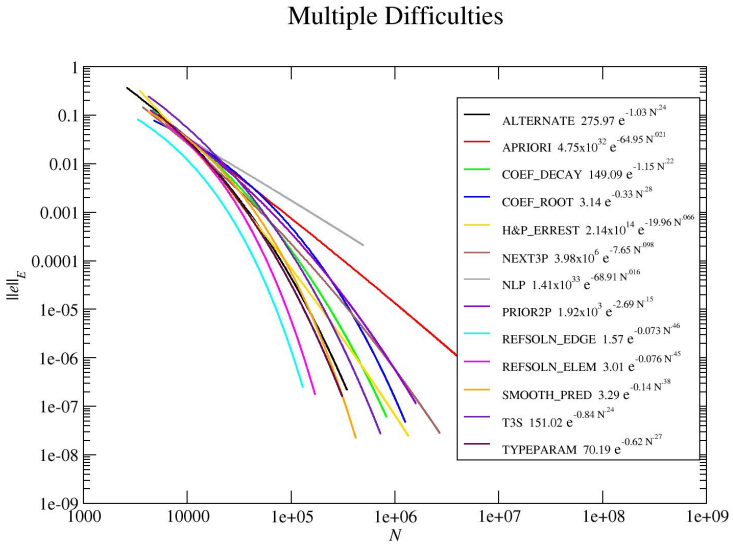


Figure 24: Log-Log plot of the convergence of all strategies with the multiple difficulties problem.

5.2 Computation Time

In the previous section we presented performance results for the *hp*-adaptive strategies on a number of test problems in terms of error vs. the number of degrees of freedom. It would be interesting to also see a comparison in terms of error vs. computation time. However, we do not believe we could perform a fair comparison of that nature at this time for a number of reasons, not the least of which is that the implementation of the strategies in PHAML emphasized correctness of the method and was not tuned for optimal performance. Nevertheless, to satisfy one's curiosity about computation time, we present timing results for one problem, the mild peak problem at a tolerance of 10^{-6}. These times should not be taken too seriously; they should only be viewed as a rough estimate of the relative time required by each of the strategies.

These computations were performed in single user mode on a single core of a Dell Latitude D630 with the Intel Core 2 Duo processor T7700 operating under the CentOS 5.5 distribution of Linux with the 2.6.18 kernel. PHAML Version 1.8.1 was compiled with the Intel Fortran 95 compiler Version 11.1.072 using -O for optimization.

The results are given in Table 1. The first column gives the total wall clock time (in seconds) spent in refinement. There is some variation in the number of times each strategy went through the refine/solve loop making it difficult to compare the time spent in a single refinement phase of the loop using only these numbers. The second column gives the number of refine/solve loops, and the third gives the quotient of the first two columns to obtain the average time spent in a refinement phase. These figures show pretty much what one would expect *a priori*. Most of the strategies use between 0.035 and 0.111 seconds per refinement phase, which, due to the considerations above, should be considered approximately equal in this context,

strategy	total time in refinement (s.)	number of ref/solve loops	average time per refinement (s./loop)
ALTERNATE	1.55	16	0.097
APRIORI	0.95	27	0.035
COEF_DECAY	0.94	11	0.085
COEF_ROOT	0.88	12	0.073
H&P_ERREST	1.44	11	0.131
NEXT3P	7.09	11	0.645
NLP	3969.16	13	305.320
PRIOR2P	1.33	12	0.111
REFSOLN_EDGE	29.38	19	1.546
REFSOLN_ELEM	20.01	12	1.668
SMOOTH_PRED	1.03	11	0.094
T3S	0.38	8	0.048
TYPEPARAM	1.08	15	0.072

Table 1: Wall clock time for the refinement phases of the solution of the mild peak problem with $\tau = 10^{-6}$, the number of refine/solve loops, and the average time for a refinement phase of the loops.

roughly .07 seconds. The H&P_ERREST strategy takes about twice as long, which makes sense because it computes two error indicators instead of one. The NEXT3P strategy takes about ten times longer, which makes sense because, not only is it computing three error indicators, but those error indicators are more expensive than the basic error indicator because they use a higher polynomial degree. The two reference solution strategies are roughly equal and take much longer than most strategies because they solve the expensive reference solution. Finally, NLP is extremely expensive, taking about 5000 times as long as the typical strategy because it has to solve the optimization problem.

5.3 Summary and Observations

In this section, we summarize the results in Section 5.1 to examine the relative performance of the strategies in different situations. The test problems are grouped into six categories: easy problems, hard problems, and singular problems at low accuracy and high accuracy. For low accuracy, which is typical in engineering applications, we use 10^{-2}, or 1% relative error, for most problems. For high accuracy, which is of interest mathematically and useful in some scientific applications, we use 10^{-6} for most problems.

Tables 2–7 give a straightforward ranking of the strategies for each problem based on the 3-parameter least squares fit. The four best strategies for each problem are highlighted in green, and the four worst in red to make it easy to see which strategies are consistently good or bad in a given category.

strategy	analytic	mild peak	mild boundary layer	mild oscillatory	mild wave front
ALTERNATE	11	13	11	12	12
APRIORI	4	3	13	1	6
COEF_DECAY	6	12	6	9	5
COEF_ROOT	7	9	8	13	10
H&P_ERREST	9	6	1	8	2
NEXT3P	2	5	4	6	4
NLP	12	7	7	10	11
PRIOR2P	8	10	5	11	7
REFSOLN_EDGE	1	1	3	3	1
REFSOLN_ELEM	5	2	9	2	3
SMOOTH_PRED	10	11	10	7	8
T3S	13	8	12	5	13
TYPEPARAM	3	4	2	4	9

Table 2: Low accuracy ranking of each strategy for easy problems.

strategy	analytic	mild peak	mild boundary layer	mild oscillatory	mild wave front
ALTERNATE	12	13	12	12	13
APRIORI	1	2	13	1	8
COEF_DECAY	8	12	5	8	7
COEF_ROOT	6	10	8	9	9
H&P_ERREST	13	8	11	13	11
NEXT3P	9	7	9	10	10
NLP	7	9	6	7	4
PRIOR2P	10	11	10	11	12
REFSOLN_EDGE	4	4	2	3	2
REFSOLN_ELEM	3	1	4	2	1
SMOOTH_PRED	5	6	3	5	5
T3S	11	5	7	6	6
TYPEPARAM	2	3	1	4	3

Table 4: High accuracy ranking of each strategy for easy problems.

strategy	sharp peak	strong boundary layer	strong oscillatory	steep wave front	asymmetric wave front
ALTERNATE	9	2	12	10	11
APRIORI	11	13	1	11	10
COEF_DECAY	5	10	9	5	7
COEF_ROOT	7	11	10	9	9
H&P_ERREST	8	9	8	6	4
NEXT3P	4	4	7	4	3
NLP	12	8	13	12	13
PRIOR2P	6	12	11	7	6
REFSOLN_EDGE	1	1	4	1	1
REFSOLN_ELEM	2	5	5	2	2
SMOOTH_PRED	13	7	6	3	5
T3S	10	3	2	13	12
TYPEPARAM	3	6	3	8	8

Table 3: Low accuracy ranking of each strategy for hard problems.

strategy	sharp peak	strong boundary layer	strong oscillatory	steep wave front	asymmetric wave front
ALTERNATE	13	1	12	3	4
APRIORI	8	13	2	9	9
COEF_DECAY	10	8	7	8	8
COEF_ROOT	9	11	8	10	10
H&P_ERREST	6	9	10	7	6
NEXT3P	7	4	9	11	11
NLP	5	12	13	13	13
PRIOR2P	11	10	11	12	12
REFSOLN_EDGE	2	2	3	1	1
REFSOLN_ELEM	1	3	4	2	2
SMOOTH_PRED	12	5	6	6	7
T3S	4	7	1	5	5
TYPEPARAM	3	6	5	4	3

Table 5: High accuracy ranking of each strategy for hard problems.

strategy	nearly straight reentrant corner	wide angle reentrant corner	L-shaped domain	narrow angle reentrant corner	slit domain	mode 1 linear elasticity	mode 2 linear elasticity	battery	singular well	intersecting interfaces	multiple difficulties
ALTERNATE	11	11	11	12	12	12	11	10	12	12	5
APRIORI	10	9	5	3	3	2	8	5	6	2	10
COEF_DECAY	3	1	1	2	2	3	3	4	5	3	8
COEF_ROOT	5	2	3	4	5	5	4	12	10	4	12
H&P_ERREST	1	3	7	8	8	8	5	1	3	11	2
NEXT3P	2	5	8	9	9	6	10	3	7	9	6
NLP	4	7	4	11	11	11	9	9	13	10	13
PRIOR2P	8	6	6	6	7	9	6	6	9	6	9
REFSOLN_EDGE	6	4	2	1	1	1	1	11	1	1	1
REFSOLN_ELEM	9	8	9	5	6	4	7	2	8	8	3
SMOOTH_PRED	12	12	12	10	10	10	12	8	4	7	7
T3S	13	13	13	13	13	13	13	7	2	13	11
TYPEPARAM	7	10	10	7	4	7	2	13	11	5	4

Table 6: Low accuracy ranking of each strategy for singular problems.

The ranking of the strategies indicates which strategies did best, but it does not indicate how much better one strategy is than another (or how close they are to being nearly the same). For this we can examine the factor by which N for a particular strategy is larger than N for the best strategy. For each problem, for each strategy compute the value of N that gives the desired accuracy according to the formula for the 3-parameter least squares fit, as illustrated in Figure 25. Let N_{best} be the minimum such value over all the strategies. For each strategy compute the factor by which N is larger than the best strategy, $N_{\text{strategy}}/N_{\text{best}}$. For example, in Figure 25 the factor for ALTERNATE is $53730/7787 \approx 6.90$.

The factors are illustrated in Figures 26–31. Each circle represents the factor for one problem in the given category. If there is a number at the top of the graph, it indicates the number of factors that are larger than 10. The strategies that performed the best in that category have all the circles near the bottom of the graph, as in REFSOLN_EDGE, REFSOLN_ELEM and TYPEPARAM in Figure 26. To the right of the graph, the strategies are ranked according to the average of the factors for that category.

Based on the tables and figures in this section and Section 5.2, we make the following observations.

- REFSOLN_EDGE and REFSOLN_ELEM are the top two strategies in all categories except singular problems at low accuracy where they are in the top 5 with factors less than 2. Also note that REFSOLN_EDGE would have been the best strategy in that category if it had not performed poorly on the battery problem. The two strategies are equally good with each of them having the better average factor in three categories, and the largest ratio of their average factors being about 1.35. However, these strategies are considerably more expensive than most strategies.

strategy	nearly straight reentrant corner	wide angle reentrant corner	L-shaped domain	narrow angle reentrant corner	slit domain	mode 1 linear elasticity	mode 2 linear elasticity	battery	singular well	intersecting interfaces	multiple difficulties
ALTERNATE	13	12	12	12	12	11	12	10	9	12	5
APRIORI	1	1	1	3	2	2	2	3	6	1	12
COEF_DECAY	7	7	7	7	8	8	8	7	7	3	8
COEF_ROOT	4	6	6	6	7	10	6	12	10	6	9
H&P_ERREST	8	9	9	10	10	7	9	1	12	11	7
NEXT3P	10	10	11	11	9	13	13	8	8	9	10
NLP	11	11	10	9	11	6	5	4	13	10	13
PRIOR2P	6	4	4	5	6	9	4	9	11	8	11
REFSOLN_EDGE	3	3	2	1	1	1	1	11	1	2	1
REFSOLN_ELEM	2	2	3	2	3	3	3	2	2	5	2
SMOOTH_PRED	9	5	5	4	4	4	7	6	3	4	4
T3S	12	13	13	13	13	12	11	5	4	13	6
TYPEPARAM	5	8	8	8	5	5	10	13	5	7	3

Table 7: High accuracy ranking of each strategy for singular problems.

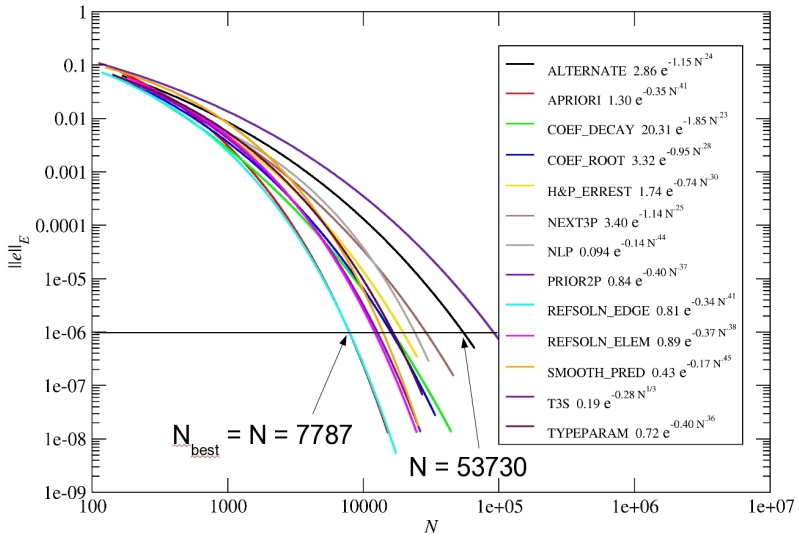
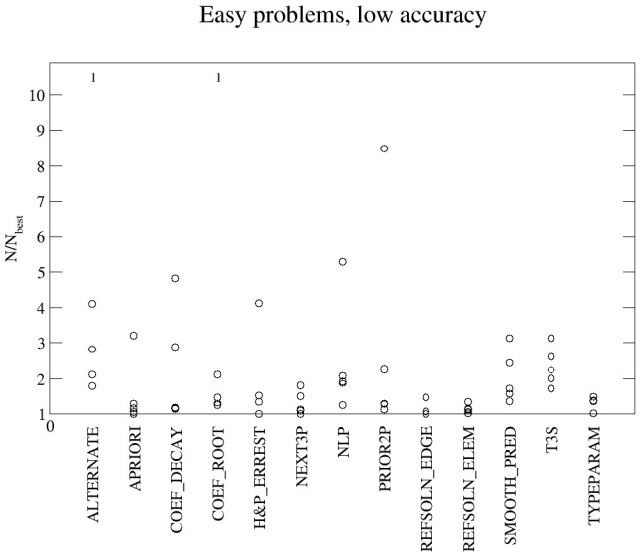
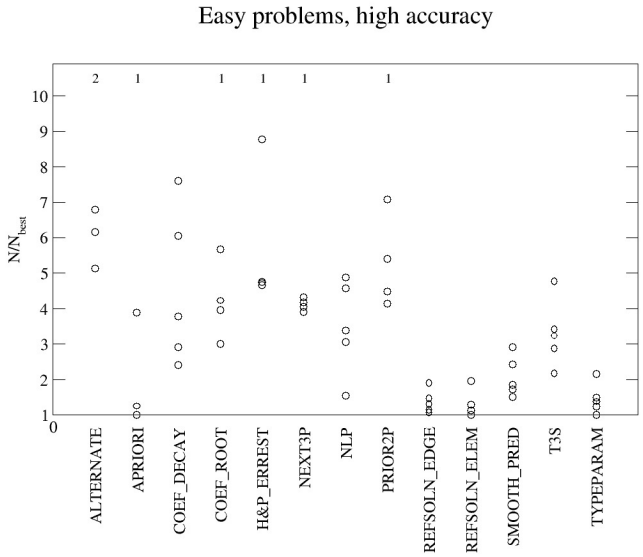


Figure 25: Computation of the factor by which N for a particular strategy is larger than N for the best strategy. In this illustration, for an accuracy of 10^{-6} the factor for ALTERNATE is $53730/7787 \approx 6.90$.



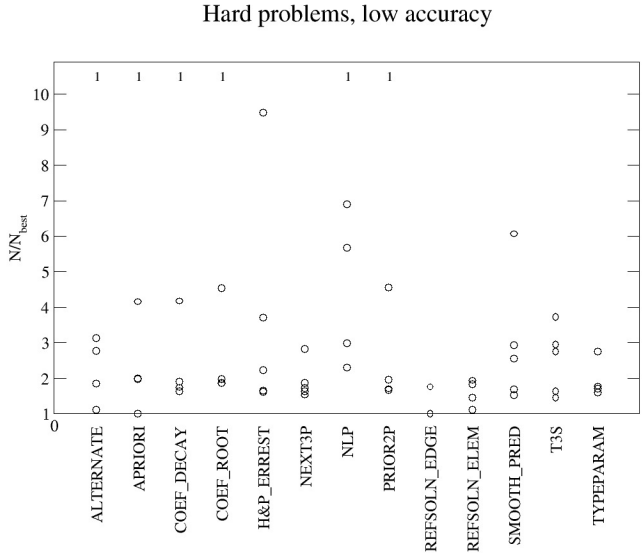
REFSOLN_EDGE	1.11
REFSOLN_ELEM	1.15
TYPEPARAM	1.25
NEXT3P	1.32
APRIORI	1.54
H&P_ERREST	1.80
SMOOTH_PRED	2.05
COEF_DECAY	2.24
T3S	2.35
NLP	2.48
PRIOR2P	2.89
COEF_ROOT	3.23
ALTERNATE	4.17

Figure 26: Factors by which N is larger than the best strategy for each easy problem at low accuracy. The table contains the average over all problems in the category.



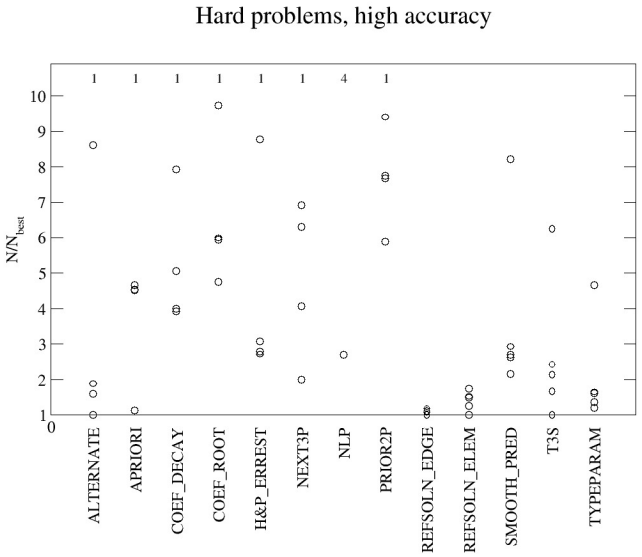
REFSOLN_ELEM	1.27
REFSOLN_EDGE	1.39
TYPEPARAM	1.45
SMOOTH_PRED	2.08
T3S	3.30
APRIORI	3.43
NLP	3.49
COEF_DECAY	4.55
NEXT3P	5.29
COEF_ROOT	5.38
PRIOR2P	6.22
H&P_ERREST	6.58
ALTERNATE	7.62

Figure 27: Factors by which N is larger than the best strategy for each easy problem at high accuracy. The table contains the average over all problems in the category.



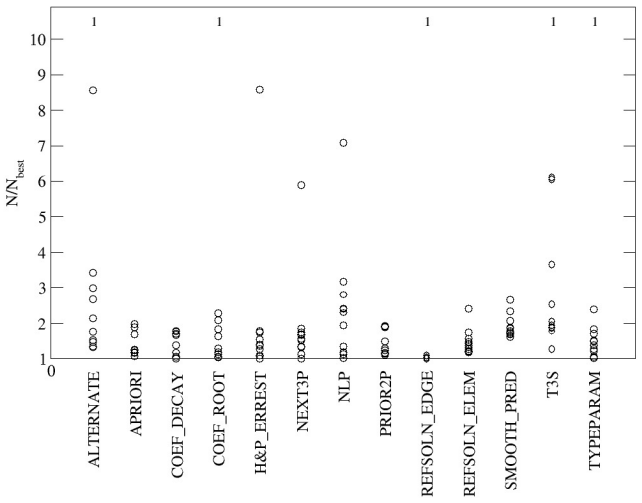
REFSOLN_EDGE	1.15
REFSOLN_ELEM	1.56
TYPEPARAM	1.89
NEXT3P	1.92
T3S	2.51
SMOOTH_PRED	2.95
H&P_ERREST	3.74
ALTERNATE	3.77
APRIORI	3.83
COEF_DECAY	3.89
PRIOR2P	3.98
COEF_ROOT	4.04
NLP	5.57

Figure 28: Factors by which N is larger than the best strategy for each hard problem at low accuracy. The table contains the average over all problems in the category.



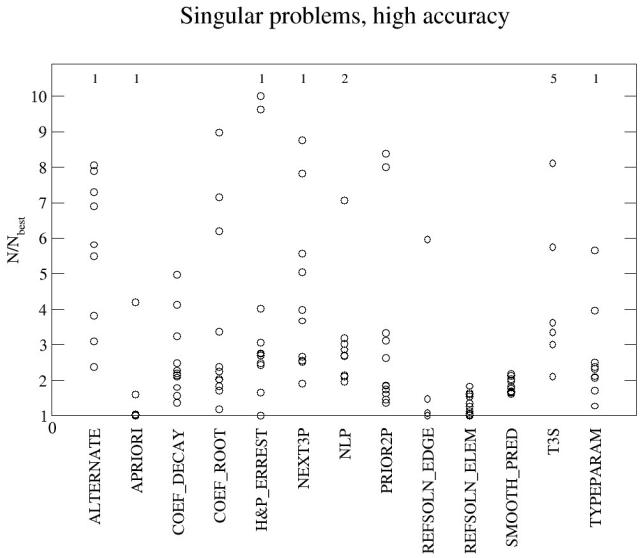
REFSOLN_EDGE	1.07
REFSOLN_ELEM	1.40
TYPEPARAM	2.10
T3S	2.69
SMOOTH_PRED	3.72
ALTERNATE	4.62
APRIORI	4.97
H&P_ERREST	5.47
NEXT3P	5.86
COEF_DECAY	6.19
COEF_ROOT	7.28
PRIOR2P	8.14
NLP	8.54

Figure 29: Factors by which N is larger than the best strategy for each hard problem at high accuracy. The table contains the average over all problems in the category.



COEF_DECAY	1.33
APRIORI	1.37
REFSOLN_ELEM	1.47
PRIOR2P	1.54
REFSOLN_EDGE	1.83
NEXT3P	1.89
SMOOTH_PRED	1.92
H&P_ERREST	1.99
COEF_ROOT	2.24
TYPEPARAM	2.26
NLP	2.43
ALTERNATE	3.38
T3S	3.56

Figure 30: Factors by which N is larger than the best strategy for each singular problem at low accuracy. The table contains the average over all problems in the category.



REFSOLN_ELEM	1.36
REFSOLN_EDGE	1.51
SMOOTH_PRED	1.88
APRIORI	2.17
COEF_DECAY	2.56
PRIOR2P	3.21
TYPEPARAM	3.27
COEF_ROOT	3.55
H&P_ERREST	3.86
NLP	4.34
NEXT3P	4.95
ALTERNATE	6.15
T3S	6.90

Figure 31: Factors by which N is larger than the best strategy for each singular problem at high accuracy. The table contains the average over all problems in the category.

- TYPEPARAM is the third best strategy in all categories of nonsingular problems, and is in the middle of the pack for singular problems where it has an average factor of 2.26 for low accuracy and 3.27 for high accuracy.
- SMOOTH_PRED is in the top 5 in all categories at high accuracy, and is the third best strategy for singular problems at high accuracy. But at low accuracy its average factors are in the middle of the pack and it is in the bottom four for many problems.
- APRIORI performs very well on singular problems with known point singularities and three of the five easy problems at both low and high accuracy. But it performs poorly on the hard problems, except for the strong oscillatory problem, and very poorly with the boundary layer.
- NEXT3P performs very well on nonsingular problems at low accuracy and fairly well on singular problems at low accuracy, but it is a bit more expensive than most strategies. It is below the middle of the pack at high accuracy with average factors around 5.
- T3S performs fairly well on nonsingular problems, but very poorly on singular problems where it has the largest average factor at both low and high accuracy, and is the worst strategy on about half of the singular problems.
- PRIOR2P performs poorly on nonsingular problems, but did very well on singular problems at low accuracy and fairly well on singular problems at high accuracy.
- COEF_DECAY is in the middle of the pack in all categories except for singular problems at low accuracy where it has the smallest average factor and is in the top four for most problems.

- H&P_ERREST is in the middle of the pack in all categories except easy problems at high accuracy where it performed poorly.
- COEF_ROOT performed poorly on nonsingular problems and is in the middle of the pack on singular problems.
- ALTERNATE performs very poorly in all categories, although it did well on a few of the hard problems.
- NLP performs poorly in most cases and is extremely expensive.

6 Conclusion and Future Work

In this paper we presented the results of a study of strategies for the *hp*-adaptive finite element method for 2D linear elliptic partial differential equations using newest node bisection of triangles. The *hp*-strategies are methods for determining how to select between the different possibilities of *h*- and *p*-refinement. Thirteen strategies were described and compared in a numerical experiment using 21 test problems. The primary metric for comparison was the convergence of the relative energy norm of the error vs. the number of degrees of freedom. A rough comparison of computation time was also presented, confirming the *a priori* expectations of the relative expense of the strategies.

We found that the REFSOLN_EDGE and REFSOLN_ELEM strategies performed best overall, in convergence, and are comparable to each other. However, they are considerably more expensive than other viable strategies. For problems with known point singularities and no other significant features, APRIORI appears to be the less expensive method of choice. For nonsingular problems, TYPEPARAM performs very well and is quite inexpensive. Another inexpensive strategy that performed very well at high accuracy is SMOOTH_PRED. Most of the other strategies have their good and bad moments.

Since the determination of what strategies to include in this study, other strategies have come to our attention or have come into existence. For future work we will extend the results of this study to include additional strategies as they are discovered. Also, we hope to use the lessons learned from this study to develop a better general purpose *hp*-strategy. For example, is it possible to get the excellent convergence performance of the reference solution strategies without the expense of computing the reference solution by combining some aspects of the reference solution strategies with some aspects of other strategies? Our conclusion is that, at this time, there is still much opportunity for the development of a general purpose *hp*-adaptive strategy that is both efficient and effective.

References

[1] S. Adjerid, M. Aiffa, and J.E. Flaherty, *Computational methods for singularly perturbed systems*, Singular Perturbation Concepts of Differential Equations (Providence) (J. Cronin and R.E. O'Malley, eds.), AMS, 1998.

[2] M. Ainsworth and J. T. Oden, *a posteriori error estimation in finite element analysis*, John Wiley & Sons, New York, 2000.

[3] M. Ainsworth and B. Senior, *An adaptive refinement strategy for h-p finite element computations*, Appl. Numer. Math. **26** (1997), no. 1-2, 165–178.

[4] ———, *hp-finite element procedures on non-uniform geometric meshes: adaptivity and constrained approximation*, Grid Generation and Adaptive Algorithms (New York) (M. W. Bern, J. E. Flaherty, and M. Luskin, eds.), vol. 113, IMA Volumes in Mathematics and its Applications, Springer-Verlag, 1999, pp. 1–28.

[5] R. Andreani, E. G. Birgin, J. M. Martnez, and M. L. Schuverdt, *On augmented Lagrangian methods with general lower-level constraints*, SIAM J. Optim. **18** (2007), 1286–1309.

[6] I. Babuška and M. Suri, *The h-p version of the finite element method with quasiuniform meshes*, RAIRO Modél. Math. Anal. Numér. **21** (1987), 199–238.

[7] ———, *The p- and h-p versions of the finite element method, an overview*, Comput. Methods Appl. Mech. Engrg. **80** (1990), 5–26.

[8] K. S. Bey, *An hp adaptive discontinuous Galerkin method for hyperbolic conservation laws*, Ph.D. thesis, University of Texas at Austin, Austin, TX, 1994.

[9] E. G. Birgin, *TANGO home page*, `http://www.ime.usp.br/~egbirgin/tango/`.

[10] L. Demkowicz, *Computing with hp-adaptive finite elements, Volume 1, One and two dimensional elliptic and Maxwell problems*, Chapman & Hall/CRC, Boca Raton, FL, 2007.

[11] L. Demkowicz, W. Rachowicz, and Ph. Devloo, *A fully automatic hp-adaptivity*, J. Sci. Comput. **17** (2002), 127–155.

[12] T. Eibner and J. M. Melenk, *An adaptive strategy for hp-FEM based on testing for analyticity*, Comput. Mech. **39** (2007), no. 5, 575–595.

[13] W. Gui and I. Babuška, *The h, p and h-p versions of the finite element method in 1 dimension. Part 3: The adaptive h-p version*, Numer. Math. **49** (1986), 659–683.

[14] B. Guo and I. Babuška, *The h-p version of the finite element method. Part 1: The basic approximation results*, Comput. Mech. **1** (1986), 21–41.

[15] P. Houston, B. Senior, and E. Süli, *Sobolev regularity estimation for hp-adaptive finite element methods*, Numerical Mathematics and Advanced Appplications (Berlin) (F. Brezzi, A Buffa, S. Corsaro, and A. Murli, eds.), Springer-Verlag, 2003, pp. 619–644.

[16] C. Mavriplis, *Adaptive mesh strategies for the spectral element method*, Comput. Methods Appl. Mech. Engrg. **116** (1994), 77–86.

[17] J. M. Melenk and B. I. Wohlmuth, *On residual-based a-posteriori error estimation in hp-FEM*, Adv. Comput. Math. **15** (2001), 311–331.

[18] W. F. Mitchell, *PHAML home page*, `http://math.nist.gov/phaml`.

[19] ———, *A comparison of adaptive refinement techniques for elliptic problems*, ACM Trans. Math. Software **15** (1989), 326–347.

[20] ———, *Adaptive refinement for arbitrary finite element spaces with hierarchical bases*, J. Comput. Appl. Math. **36** (1991), 65–78.

[21] _____, *A collection of 2D elliptic problems for testing adaptive algorithms*, NISTIR 7668, National Institute of Standards and Technology, 2010, also submitted to this journal.

[22] W. F. Mitchell and M. A. McClain, *A comparison of hp-adaptive strategies for elliptic partial differential equations (long version)*, NISTIR 7824, National Institute of Standards and Technology, 2011.

[23] _____, *A survey of hp-adaptive strategies for elliptic partial differential equations*, Recent Advances in Computational and Applied Mathematics (T. E. Simos, ed.), Springer, 2011, pp. 227–258.

[24] J. T. Oden and A. Patra, *A parallel adaptive strategy for hp finite element computations*, Comput. Methods Appl. Mech. Engrg. **121** (1995), 449–470.

[25] J. T. Oden, A. Patra, and Y. Feng, *An hp adaptive strategy*, Adaptive Multilevel and Hierarchical Computational Strategies (A. K. Noor, ed.), vol. 157, ASME Publication, 1992, pp. 23–46.

[26] A. Patra, *private communication*.

[27] A. Patra and A. Gupta, *A systematic strategy for simultaneous adaptive hp finite element mesh modification using nonlinear programming*, Comput. Methods Appl. Mech. Engrg. **190** (2001), 3797–3818.

[28] W. Rachowicz, J. T. Oden, and L. Demkowicz, *Toward a universal h-p adaptive finite element strategy, Part 3. Design of h-p meshes*, Comput. Methods Appl. Mech. Engrg. **77** (1989), 181–212.

[29] A. Schmidt and K. G. Siebert, *a posteriori estimators for the $h-p$ version of the finite element method in 1D*, Appl. Numer. Math. **35** (2000), 43–66.

[30] P. Šolín, J. Červený, and I. Doležel, *Arbitrary-level hanging nodes and automatic adaptivity in the hp-FEM*, Math. Comput. Simulation **77** (2008), 117–132.

[31] P. Šolín, K. Segeth, and I. Doležel, *Higher-order finite element methods*, Chapman & Hall/CRC, New York, 2004.

[32] E. Süli, P. Houston, and Ch. Schwab, *hp-finite element methods for hyperbolic problems*, The Mathematics of Finite Elements and Applications X. MAFELAP (J.R. Whiteman, ed.), Elsevier, 2000, pp. 143–162.

[33] B. Szabo and I. Babuška, *Finite element analysis*, John Wiley and Sons, New York, 1991.

[34] R. Verfürth, *A review of a posteriori error estimation and adaptive mesh-refinement techniques*, Wiley Teubner, Chichester Stuttgart, 1996.