A Survey of *hp*-Adaptive Strategies for Elliptic Partial Differential Equations

William F. Mitchell*
Marjorie A. McClain
Mathematical and Computational Sciences 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 demonstrate the exponential convergence rates with two classic test problems.

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. Self-adaptive methods to determine a quasi-optimal grid are a critical component of the improvements, and have been studied for nearly 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

*Contribution of NIST, not subject to copyright. Preprint, submitted 2009. See http://math.nist.gov/∼WMitchell for the current publication status.

approximate solution is a polynomial over each element. Most of the work has focused on h-adaptive methods. In these methods, the mesh size, 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 are approximately equal over all elements.

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 indicates 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. In this paper we summarize 15 such hp-adaptive strategies.

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 an hp-adaptive finite element algorithm. Section 4 defines the hp-adaptive strategies. Section 5 contains numerical results to demonstrate the convergence achieved by the different strategies. Finally, we draw our conclusions in Section 6.

2 The Finite Element Method

For simplicity, consider the Poisson boundary value problem

$$-\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = f(x,y) \quad \text{in } \Omega \tag{1}$$

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

where Ω is a bounded, connected, open region in $\mathbb{R}^2$. Note, however, that everything in this paper applies equally well to a general second order elliptic PDE with mixed boundary conditions. The data in Equations 1-2 are assumed to satisfy the usual ellipticity and regularity assumptions.

Denote by $L^2(\Omega)$ the space of square integrable functions over Ω with inner product

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

and norm

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

$H^m(\Omega)$ denotes the usual Sobolev spaces of functions whose derivatives up to order m are in $L^2(\Omega)$. The Sobolev spaces have inner products

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

and norms

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

where

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

Let $H^m_0(\Omega) = \{v \in H^m(\Omega) : v = 0 \text{ on } \partial\Omega\}$. Let $\tilde{u}_D$ be a lift function satisfying the Dirichlet boundary conditions in Equation 2 and define the affine space $\tilde{u}_D + H^1_0(\Omega) = \{\tilde{u}_D + v : v \in H^1_0(\Omega)\}$. Define the bilinear form

$$B(u, v) = \iint_\Omega \frac{\partial u}{\partial x}\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\frac{\partial v}{\partial y}$$

and the linear form

$$L(v) = \iint_\Omega f v$$

Then the variational form of the problem is to find the unique $u \in \tilde{u}_D + H^1_0(\Omega)$ that satisfies

$$B(u, v) = L(v) \quad \forall v \in H^1_0(\Omega).$$

The energy norm of $v \in H^1_0$ is defined by $||v||^2_{E(\Omega)} = B(v, v)$.

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 the minimum rule, i.e. the edge is assigned the minimum of the degrees of the adjacent elements.

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 $\{\phi_i\}_{i=1}^{N_{dof}}$ that span $\tilde{u}_D + V_{hp}$,

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

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

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

N_{dof} is the number of degrees of freedom in the solution. The p-hierarchical basis of Szabo and Babuška [31], which is based on Legendre polynomials, is used in the program used for most of the results in Section 5. The basis functions are hierarchical in the sense that the basis functions for a space of degree p are a subset of the basis functions for a space of degree $p+1$. For an element of degree p_i with edge degrees $p_{i,j}$, $j = 1, 2, 3$ there is one linear basis function associated with each vertex, $p_{i,j} - 1$ basis functions, of degree $2, 3 \ldots p_{i,j}$, associated with edge j, and $q - 2$ basis functions of degree q for $q = 3, 4 \ldots p_i$ (for a total of $(p_i - 1)(p_i - 2)/2$) whose support is the interior of the triangle.

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

$$Ax = b \tag{3}$$

where the matrix A is given by $A_{ij} = B(\phi_i, \phi_j)$ and the right hand side is given by $b_i = L(\phi_i)$. The solution x consists of the α_i's.

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]

$$||e_{hp}||_{H^1(\Omega)} \leq C \frac{h^\mu}{p^{m-1}} ||u||_{H^m(\Omega)} \tag{4}$$

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 [13] to converge exponentially in the number of degrees of freedom,

$$||e_{hp}||_{H^1(\Omega)} \leq C_1 e^{-C_2 N_{dof}^{1/3}} \tag{5}$$

for some C_1, $C_2 > 0$ independent of N_{dof}.

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. In this paper, the following approaches are used.

Triangles are h-refined by the newest node bisection method [18]. 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 1 assigned to the triangles of the initial coarse grid. p-refinement is performed by increasing the degree of the element by one, followed by enforcing the minimum rule for the edges. Coarsening of elements means reversing the refinement.

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, 32]. For this paper, the error indicator for element T_i is given by solving a local Neumann residual problem:

$$-\frac{\partial^2 e_i}{\partial x^2} - \frac{\partial^2 e_i}{\partial y^2} = f - \frac{\partial^2 u_{hp}}{\partial x^2} - \frac{\partial^2 u_{hp}}{\partial y^2} \qquad \text{in } T_i \tag{6}$$

$$e_i = 0 \qquad \text{on } \partial T_i \cap \partial\Omega \tag{7}$$

$$\frac{\partial e_i}{\partial n} = -\frac{1}{2}\left[\frac{\partial u_{hp}}{\partial n}\right] \qquad \text{on } \partial T_i \setminus \partial\Omega \tag{8}$$

where $\frac{\partial}{\partial n}$ is the unit outward normal derivative and $\left[\frac{\partial u_{hp}}{\partial n}\right]$ is the jump in the outward normal derivative of u_{hp} across the element boundary. The approximate solution, $e_{i,hp}$ of Equations 6-8 is computed using the hierarchical bases of exact degree $p_i + 1$, where p_i is the degree of T_i. 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

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

The criterion for program termination is that the relative error estimate be smaller than a prescribed error tolerance τ, i.e. $\eta/||u_{hp}||_{E(\Omega)} < \tau$. Elements are selected for coarsening if $\eta_i < \max_i \eta_i/100$ and for refinement if $\eta_i > \tau||u_{hp}||_{E(\Omega)}/\sqrt{N_T}$. 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.

4 The hp-Adaptive Strategies

In this section, the hp-adaptive strategies that have been proposed in the literature are presented. 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 bounds in Equations 4 and 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. For example, it is known that linear elliptic PDEs with smooth coefficients and piecewise analytic boundary data will have point singularities only near reentrant corners of the boundary and where boundary conditions change [4]. Another example would be a situation where one knows the approximate location of a shock in the interior of the domain.

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.

We extend this strategy to allow more general regions of irregularity, and to provide the strength of the irregularity. The user provides a function that, given an element T_i as input, returns a regularity value for that element. For true singularities, it would ideally return the maximum value of m such that $u \in H^m(T_i)$. But it can also indicate that a triangle intersects an area that is considered to be nearly irregular, like a boundary layer or sharp wave front. Based on the definition of μ in Equation 4, if the current degree of the triangle is p_i and the returned regularity value is m_i, we do p-refinement if $p_i \leq m_i - 1$ and h-refinement otherwise. The same approach is used in all the strategies that estimate the local regularity m_i.

4.2 Estimate Regularity Using Smaller p Estimates

Süli, Houston and Schwab [30] presented a strategy based on Equation 4 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. This requires $p_i \geq 3$, so we always use p-refinement in elements of degree 1 and 2.

Suppose the error estimate in Equation 4 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.

Thanks to the p-hierarchical basis, the computation of the error estimates is very inexpensive. For $1 \leq j < p_i$,

$$u_{hp}|_{T_i} = \sum_{\text{supp}(\phi_k)\cap T_i \neq \emptyset} \alpha_k \phi_k = \sum_{\substack{\text{supp}(\phi_k)\cap T_i \neq \emptyset \\ \deg(\phi_k) \leq p_i - j}} \alpha_k \phi_k + \sum_{\substack{\text{supp}(\phi_k)\cap T_i \neq \emptyset \\ \deg(\phi_k) > p_i - j}} \alpha_k \phi_k$$

where $\text{supp}(\phi_k)$ is the support of ϕ_k and $\deg(\phi_k)$ is the degree of ϕ_k. The last sum is the amount by which the solution changed when the degree of the element was increased from $p_i - j$ to p_i, and provides an estimate of the error in the partial approximate solution of degree $p_i - j$ given in the next to last sum. (Indeed, the local Neumann error estimator of Equations 6-8 approximates this quantity for the increase from degree p_i to p_{i+1}.) Thus the error estimates are

$$\eta_{i,p_i-j} = ||\sum_{\substack{\text{supp}(\phi_k)\cap T_i \neq \emptyset \\ \deg(\phi_k) > p_i - j}} \alpha_k \phi_k||_{H^1(T_i)}$$

which only involves computing the norm of known quantities.

4.3 Type parameter

Gui and Babuška [12] 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 h-refined if $R(T_i) > \gamma$, and p-refined if $R(T_i) \leq \gamma$. Note that $\gamma = 0$ gives pure h-refinement and $\gamma = \infty$ gives pure p-refinement.

For the error estimates, we use the local Neumann error estimate of Equations 6-8 for η_{i,p_i}, and the η_{i,p_i-1} of Section 4.2. We use $\gamma = 0.3$ in the results of Section 5.

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 6-8 in which Equation 8 is replaced by

$$\frac{\partial e_i}{\partial n} = g_i \quad \text{on } \partial T_i \setminus \partial\Omega$$

where g_i is an approximation of $\frac{\partial u}{\partial n}$ 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 1-2, 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

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

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 4, 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, 20, 21] 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. This may be accomplished by performing uniform h-refinements of some very coarse initial mesh until the asymptotic range is reached. The resulting grid has N_0 elements

with sizes h_i, degrees p_i and *a posteriori* error estimates η_i, and approximate solution u_0. The results in Section 5 begin with $p = 1$ and assume the initial grid is sufficiently fine 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 the formula

$$n_i = \left[\frac{\Lambda_i^2 N_I h_i^{2\mu_i}}{p_i^{2(m_i-1)} \eta_I^2} \right]^{\frac{1}{\beta\mu_i+1}} \tag{9}$$

where $N_I = \sum n_i$ is the number of elements in the intermediate grid, m_i is the local regularity of the solution, $\mu_i = \min(p_i, m_i - 1)$, $\eta_I = \gamma\tau||u_0||_{E(\Omega)}$, $\beta = 1$ for 2D problems, $\eta_0^2 = \sum \eta_i^2$ and

$$\Lambda_i = \frac{\eta_i \Lambda}{\eta_0}$$

where

$$\Lambda = \frac{\eta_0 p_i^{m_i-1}}{h_i^{\mu_i}}$$

See any of the above references for the derivation of this formula. It is based on the *a priori* error estimate in Equation 4. Inserting the expression for Λ_i into Equation 9 and using $\beta = 1$ we arrive at

$$n_i = \left[\frac{\eta_i^2 N_I}{\eta_I^2} \right]^{\frac{1}{\mu_i+1}}$$

N_I is not known at this point, since it is the sum of the n_i. Successive iterations are used to solve for n_i and N_I simultaneously. We use 5 iterations, which preliminary experiments showed to be sufficient (convergence was usually achieved in 3 or 4 iterations). Once the n_i have been determined, we perform $\lfloor 0.5 + \log_2 n_i \rfloor$ uniform h-refinements (bisections) of each element T_i to generate approximately n_i children, and solve the discrete problem on the intermediate grid.

The third step is to perform adaptive p-refinement to reduce the error to the desired tolerance τ. The new degree for each element is given by

$$\hat{p}_i = p_i \left[\frac{\eta_{I,i}\sqrt{N_I}}{\eta_T} \right]^{\frac{1}{m_i-1}}$$

where $\eta_{I,i}$ is the *a posteriori* error estimate for element T_i of the intermediate grid and $\eta_T = \tau||u_0||_{E(\Omega)}$. Again, the formula is a simple reduction of the equations derived in the references. p-refinement is performed to increase the degree of each element T_i to $\hat{p}_i$, and the discrete problem is solved on the final grid.

In the results of Section 5, if $n_i < 2$ or $\hat{p}_i < p_i$ then refinement is not performed. Also, to avoid excessive refinement, the number of h-refinements done to any element in Step 2 and number of p-refinements in Step 3 is limited to 3.

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) [22]. 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 $\gamma = 6$. For the local regularity m_i we use the same routine as the APRIORI strategy (Section 4.1).

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 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 [23] proposed a strategy for hp mesh design using nonlinear programming. We refer to this strategy as NLP. They presented it in the context of quadrilaterals 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 $\{l_i\}$, error estimates $\{\eta_i\}$, and element diameters

$$h_i = \left(\frac{1}{\sqrt{2}}\right)^{l_i} H_{0,i}$$

where $H_{0,i}$ is the diameter of the element in the initial grid that contains T_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. The new grid is obtained by refining T_i $\hat{l}_i - l_i$

times (or coarsening if $\hat{l}_i < l_i$) and assigning degree $\hat{p}_i$ to the $2^{\hat{l}_i - l_i}$ children. The size of the children of T_i is

$$\hat{h}_i = \left(\frac{1}{\sqrt{2}}\right)^{\hat{l}_i} H_{0,i}.$$

There are two forms of the optimization problem, which can be informally stated as 1) minimize the number of degrees of freedom (or some other measure of grid size) subject to the error being less than a given tolerance and other constraints, and 2) minimize the error subject to the number of degrees of freedom being less than a given limit and other constraints. We will only consider the first form here; the second form simply reverses the objective function and constraint.

Computationally, the square of the error is approximated by $\sum_{i=0}^{N_T} \hat{\eta}_i^2$ where $\hat{\eta}_i$, to be defined later, is an estimate of the error in the refined grid over the region covered by T_i. The number of degrees of freedom associated with a triangle of degree p is taken to be 3/6 (one for each vertex with an average of six triangles sharing a vertex) plus $3(p-1)/2$ ($p-1$ for each edge with two triangles sharing an edge) plus $(p-1)(p-2)/2$ (for the interior), which is $p^2/2$. Thus the number of degrees of freedom over the children of T_i is $2^{\hat{l}_i - l_i}\hat{p}_i^2/2$. We can now formally state the optimization problem as

$$\underset{\{\hat{l}_i\},\{\hat{p}_i\}}{\text{minimize}} \quad \sum_{i=1}^{N_T} 2^{\hat{l}_i - l_i} \frac{\hat{p}_i^2}{2} \tag{10}$$

$$\text{s.t.} \quad \sum_{i=1}^{N_T} \hat{\eta}_i^2 \leq \hat{\tau}^2 \tag{11}$$

$$\hat{l}_j - 1 \leq \hat{l}_i \leq \hat{l}_j + 1 \ \forall j \text{ such that } T_j \text{ shares an edge with } T_i \tag{12}$$

$$1 \leq \hat{l}_i \leq l_{max} \tag{13}$$

$$1 \leq \hat{p}_i \leq p_{max} \tag{14}$$

$$l_i - \delta l_{dec} \leq \hat{l}_i \leq l_i + \delta l_{inc} \tag{15}$$

$$p_i - \delta p_{dec} \leq \hat{p}_i \leq p_i + \delta p_{inc} \tag{16}$$

where $\hat{\tau}$ is the error tolerance for this refinement phase. We use $\hat{\tau} = \eta/4$ where η is the global error estimate on the current grid. The divisor 4 is arbitrary and could be replaced by some other value. In practice, Equation 11 is divided through by τ^2 so that the numbers are $O(1)$. Equation 12 is a necessary condition for compatibility of the grid (in [23] it enforces one level of hanging nodes). It is not a sufficient condition, however any violations of compatibility while this condition is met are cases where only one triangle of a compatibly divisible pair was refined, and it is a slight adjustment to the optimal solution to also refine the other one to maintain compatibility. Equation 13 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 14 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 15 and 16 restrict the amount of decrease in l and p to prescribed limits δl_{dec} and δp_{dec}, and the amount of increase to δl_{inc} and δp_{inc}. In the results in Section 5 we used $\delta l_{dec} = \delta p_{dec} = 1$, $\delta l_{inc} = 5$, and $\delta p_{inc} = 2$.

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 [1] 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.

It remains to define $\hat{\eta}_i$, the estimate of the error in the refined grid over the region covered by T_i. Assuming approximate equality in the *a priori* error estimate of Equation 4, we have

$$\eta_i \approx C \frac{h_i^{\mu_i}}{p_i^{m_i-1}} ||u||_{H^m(T_i)}$$

and

$$\hat{\eta}_i \approx C \frac{\hat{h}_i^{\mu_i}}{\hat{p}_i^{m_i-1}} ||u||_{H^m(T_i)}$$

where m_i is the local regularity over T_i and $\mu_i = \min(p_i, m_i - 1)$. Combining these leads to

$$\hat{\eta}_i \approx \frac{\hat{h}_i^{\mu_i}}{\hat{p}_i^{m_i-1}} \frac{p_i^{m_i-1}}{h_i^{\mu_i}} \eta_i = \left(\frac{1}{\sqrt{2}}\right)^{\mu_i(\hat{l}_i - l_i)} \left(\frac{p_i}{\hat{p}_i}\right)^{m_i-1} \eta_i$$

and thus the constraint in Equation 11 is

$$\sum_{i=1}^{N_T} \left(\frac{1}{2}\right)^{\min(p_i, m_i-1)(\hat{l}_i - l_i)} \left(\frac{p_i}{\hat{p}_i}\right)^{2(m_i-1)} \eta_i^2 < \hat{\tau}^2$$

in which the only remaining quantity to be determined is m_i. Patra and Gupta suggest estimating m_i by using the observed convergence rate from two grids, with a formula very similar to that used in the PRIOR2P strategy of Section 4.2. However, this requires that p_i be at least three in every element, so instead we use the estimate of m_i from the NEXT3P strategy of Section 4.4 which allows $p_i = 1$.

[1] 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.

4.8 Another Optimization Strategy

Another strategy based on the formulation and solution of an optimization problem is given in Novotny et al. [19]. However, it turns out that 1) the optimization does not work near singularities, so *a priori* knowledge of singularities must be used to force h-refinement near singularities, and 2) for the finite element method and class of problems considered in this paper, the strategy always chooses p-refinement except for extremely large elements. Thus, this strategy is (nearly) identical to the APRIORI strategy, and will not be considered further in this paper.

4.9 Predict Error Estimate on Assumption of Smoothness

Melenk and Wohlmuth [16] 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 4. 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 $\gamma_h = 4$ and $\gamma_p = \sqrt{0.4}$.

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

In 1D, Schmidt and Siebert [25] 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 6-8 is actually an estimate of how much the error will be reduced 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.

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

$$-\frac{\partial^2 e_i}{\partial x^2} - \frac{\partial^2 e_i}{\partial y^2} = f - \frac{\partial^2 u_{hp}}{\partial x^2} - \frac{\partial^2 u_{hp}}{\partial y^2} \qquad \text{in } T_i \cup T_i^{\text{mate}} \tag{17}$$

$$e_i = g - u_{hp} \qquad \text{on } \partial(T_i \cup T_i^{\text{mate}}) \cap \partial\Omega \tag{18}$$

$$e_i = 0 \qquad \text{on } \partial(T_i \cup T_i^{\text{mate}}) \setminus \partial\Omega \tag{19}$$

where T_i^{mate} is the element that is refined along with T_i in the newest node bisection method [18]. 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.

The two error estimates can be divided by the associated increase in the number of degrees of freedom to obtain an approximate error reduction per degree of freedom, and/or 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, which seemed to work best on the largest number of test problems.

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

4.11 Legendre coefficient strategies

There are three hp-adaptive strategies that are based on the coefficients in an expansion of the solution in Legendre polynomials. In 1D, 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 jth degree Legendre polynomial scaled to the interval of element T_i.

Mavriplis [15] 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. When four coefficients are not available, we fit to whatever is available. If only one coefficient is available, we use p-refinement.

Houston et al. [14] present the other two approaches which use the Legendre coefficients to estimate the regularity of the solution. One approach estimates the regularity 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.

They also present a second way of estimating the regularity from the Legendre coefficients using the ratio test. However, they determined the ratio test is inferior to the root test, so it will not be considered further in this paper.

Both Mavriplis and Houston et al. presented the strategies in the context of one dimension and used the Legendre polynomials as the local basis so the coefficients are readily available. In [14] 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. In this paper we are using triangular elements which have a basis that is based on Legendre polynomials [31]. 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.

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

4.12 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, 24, 29]. In its full glory, the method is quite complicated. Here we present only the basic ideas of the algorithm and how we have adapted it for bisected triangles (it is usually presented in the context of rectangular elements with some reference to quadrisected triangles), and refer to the references for further details. 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 (coarse) mesh, $G_{h,p} := G_{hp}$, and current solution, $u_{h,p} := u_{hp}$, 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 norm of the difference between the current solution and reference solution is used as the global error estimate, i.e.,

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

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 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, computed by solving a local Dirichlet problem, and likewise $\Pi_{new,i}$ is the projection onto the resulting elements from any one of the candidate refinements. $|\cdot|_{H^1(T_i)}$ is the H^1 seminorm over T_i. 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, although the factor 1/3 is somewhat arbitrary.

The 2D algorithm also begins by computing a reference solution on the globally hp-refined grid $G_{h/2,p+1}$. (For bisected triangles, we should use the subscript $h/\sqrt{2}, p+1$ for the fine grid and solution, but for simplicity we will use the original notation.) 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 by increasing edge degrees as necessary.

A related, but simpler, approach was developed by Šolín et al. [28]. We refer to this strategy as REFSOLN_ELEM since it also begins by computing a reference solution, $u_{h/2,p+1}$, on $G_{h/2,p+1}$, 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 [28], 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. Šolín et al. allow an unlimited number of hanging nodes, so they have no issue of how to assign the degrees of children that were created to maintain compatibility or one level of hanging nodes. For the newest node bisection of triangles algorithm, we assign $\lfloor (p + 1)/\sqrt{2} \rfloor$ to both children of a triangle of degree p that is refined only for the sake of compatibility.

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.

The selection of which candidate to use is not simply the candidate with the smallest projection error [27]. Let N_i be the number of degrees of freedom in the space corresponding to T_i, and $N_{\text{candidate}}$ be the number of degrees of freedom in the space corresponding to a candidate. For simplicity, when computing N_i and $N_{\text{candidate}}$ we apply the minimum rule for edge degree ignoring the degrees of the neighbors of T_i, e.g. $N_i = (p_i + 1)(p_i + 2)/2$ regardless of what the actual edge degrees of T_i are.

Candidates with $\zeta_{\text{candidate}} > \zeta_i$ are discarded. We also discard any of the h-refined candidates for which the degrees are both greater than p_i since the reference solution is (locally) in that space. Let n be the number of remaining candidates. Compute the average and standard deviation of the base 10 logarithms of the ζ's

$$\bar{\zeta} = \frac{1}{n} \sum_{\text{candidates}} \log \zeta_{\text{candidate}}$$

$$\sigma = \sqrt{\frac{1}{n} \sum_{\text{candidates}} (\log \zeta_{\text{candidate}})^2 - \bar{\zeta}^2}$$

Finally, to determine which candidate to use, select an above-average candidate with the steepest error decrease, i.e., from among the candidates with $\log \zeta_{\text{candidate}} < \bar{\zeta} + \sigma$ and $N_{\text{candidate}} > N_i$, select the candidate that maximizes

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

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

5 Numerical Results

In this section we demonstrate the performance of the *hp*-adaptive strategies using two problems that are commonly used in the adaptive refinement literature. It is not the intention of this paper to compare the strategies against each other. In this paper, we merely demonstrate the ability of the strategies to achieve exponential convergence rates on a problem with a regular, but nearly irregular, solution and a problem with a point singularity.

The computations were performed on an Intel Core 2 based PC operating under the 32 bit CentOS 4 distribution of Linux with kernel 2.6.18-128.1.10.el5. Programs were compiled with Intel Fortran Version 10.1 and gcc Version 4.1.2.

Results for REFSOLN_EDGE were computed using Demkowicz's code hp2d, which was obtained from the CD in Demkowicz's book [10]. For h-refinement of triangles this code uses quadrisection with one level of hanging nodes. The maximum degree for the polynomials is 7. Results for REFSOLN_ELEM were computed using Šolín's code Hermes Version 0.99 [26]. For h-refinement of triangles this code uses quadrisection with unlimited levels of hanging nodes. The maximum degree for the polynomials is 9. Results for all other strategies were computed using PHAML Version 1.6 [17]. This code uses newest node bisection of triangles. The maximum h-refinement level was set to 53 and the maximum degree was set to 21.

To observe the convergence rates, we apply the algorithm in Figure 1 with a series of tolerances, $\tau = 0.1, 0.05, 0.025, 0.01, 0.005, \ldots, 10^{-8}$. For each run we record N_{dof} and $||e_{hp}||_{E(\Omega)}$ for the final grid and solution. A least squares fit to the exponential form

$$||e_{hp}||_{E(\Omega)} = Ae^{-BN_{dof}^{C}}$$

is computed to determine the rate of convergence. According to Equation 5, C is optimally 1/3. Slightly smaller values of C still indicate exponential convergence, although not quite optimal, but very small values of C indicate that exponential convergence was not obtained.

The first test problem is Poisson's equation given in Equations 1-2 on the unit square with the right hand sides f and g chosen so the solution is

$$u(x, y) = \tan^{-1}(\alpha(\sqrt{(x - x_c)^2 + (y - y_c)^2 - r_0})).$$

The solution has a sharp circular wave front of radius r_0 centered at (x_c, y_c) as shown in Figure 2. α determines the sharpness of the wave front. For this paper we use $\alpha = 200$, $(x_c, y_c) = (-.05, -.05)$ and $r_0 = 0.7$. The center of the circle is taken to be slightly outside the domain because the solution has a mild singularity at the center of the circle and we want to see how the strategies handle the wave front, not the singularity. For the regularity function for the APRIORI strategy we return 3.0 if the element touches the circle on which the wave front is centered, and a very large number otherwise. This causes h-refinement with cubic elements along the wave front and p-refinement elsewhere. The choice of cubics was arbitrary.

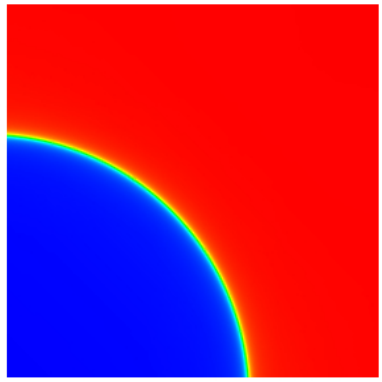

Figure 2: The solution of the wave front problem. Colors represent the function value, with blue the minimum value and red the maximum.

The convergence results are shown in Figures 3 and 4 where the norm of the error is plotted against the number of degrees of freedom on a log-log scale. The circles show the actual results for the sequence of values of τ, and the lines are the exponential least squares fit to that data. The curvature of the lines is indicative of the exponential rate of convergence. Higher curvature indicates a larger exponent on N_{dof}, and a straight line would indicate a polynomial rate of convergence.

Table 1 contains the exponents C from the exponential least squares fit. All strategies exhibit exponential rates of convergence, as none of the exponents are far from the theoretical 1/3. The differences from 1/3, both smaller and larger, may be due to effects such as suboptimal performance of the strategy, data points that are not in the asymptotic range of convergence, etc. Note that if the early (coarse grid, low accuracy) data points are suboptimal, this causes an increased curvature as the accuracy "catches up" in the finer grids, which can create exponents larger than 1/3.

The second test problem is Laplace's equation, i.e. Equation 1 with the right hand side $f = 0$, on the L-shaped domain of Figure 5. The reentrant corner induces a singularity such that the exact solution, which is also shown in Figure 5, in polar coordinates is

$$u(r, \theta) = r^{2/3} \sin(2\theta/3).$$

Dirichlet boundary conditions are set accordingly. The solution is known to be in $H^{1+2/3}$ in any neighborhood of the reentrant corner, so the regularity function for APRIORI returns $1 + 2/3$ if the element touches the reentrant corner and a very large number otherwise. This results in h-refinement with linear elements at the reentrant corner and p-refinement elsewhere.

The convergence results are shown in Figures 6 and 7, and the exponents from the least squares fit are given in Table 2. Again, all strategies achieved

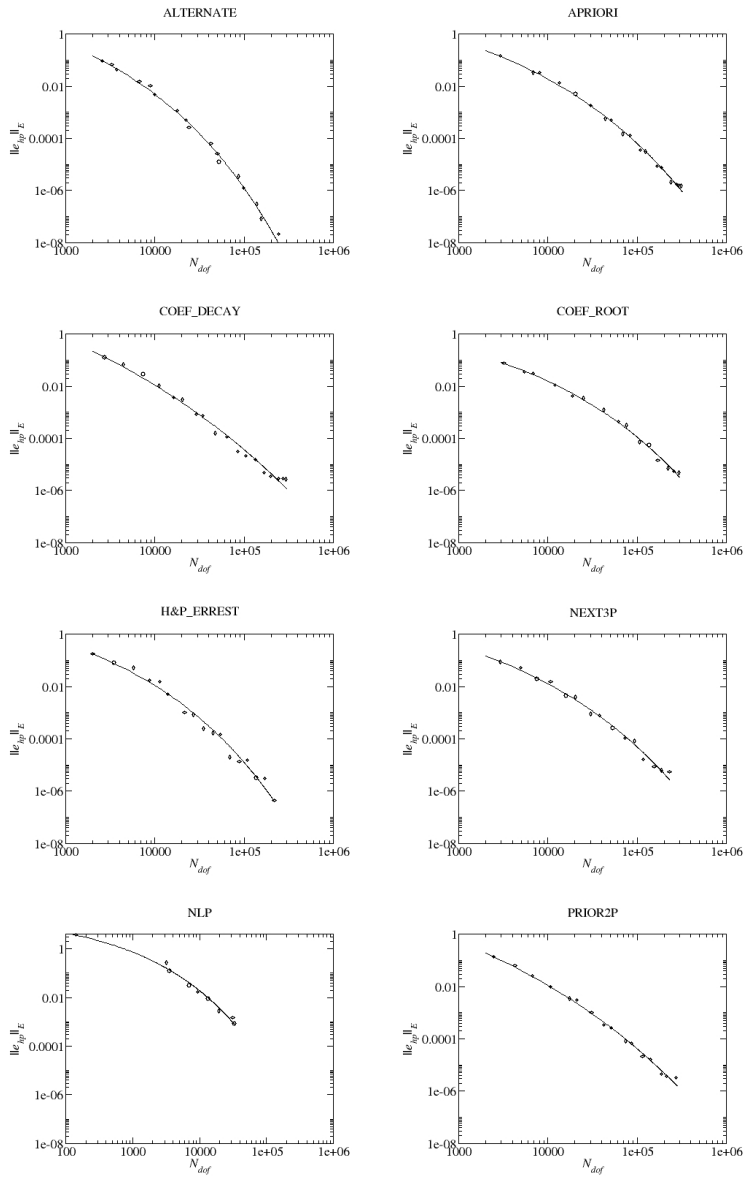

Figure 3: Convergence plots for the wave front problem.

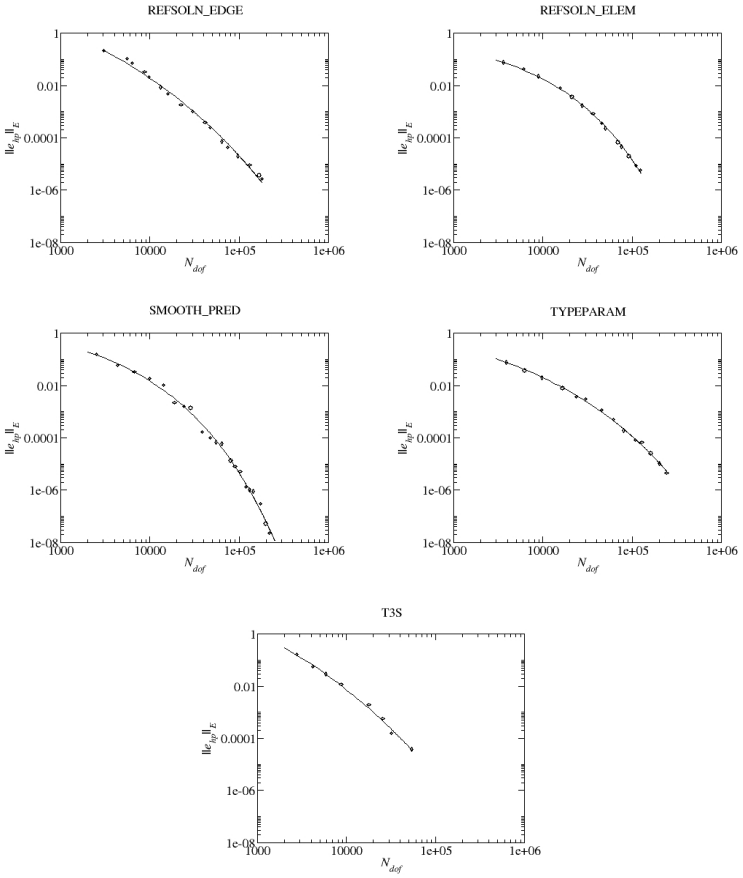

Figure 4: Convergence plots for the wave front problem (continued).

strategy	exponent C
ALTERNATE	0.27
APRIORI	0.23
COEF_DECAY	0.14
COEF_ROOT	0.25
H&P_ERREST	0.27
NEXT3P	0.23
NLP	0.30
PRIOR2P	0.16
REFSOLN_EDGE	0.21
REFSOLN_ELEM	0.44
SMOOTH_PRED	0.40
TYPEPARAM	0.28
T3S	0.18

Table 1: Exponent on N_{dof} from the exponential least squares fit to the convergence data for the wave front problem.

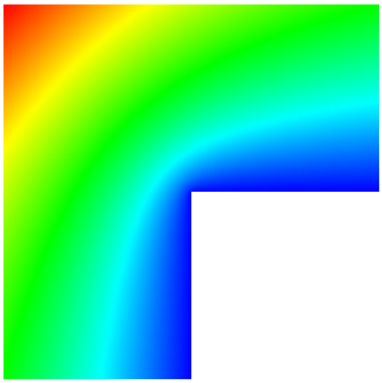

Figure 5: The solution of the L-domain problem.

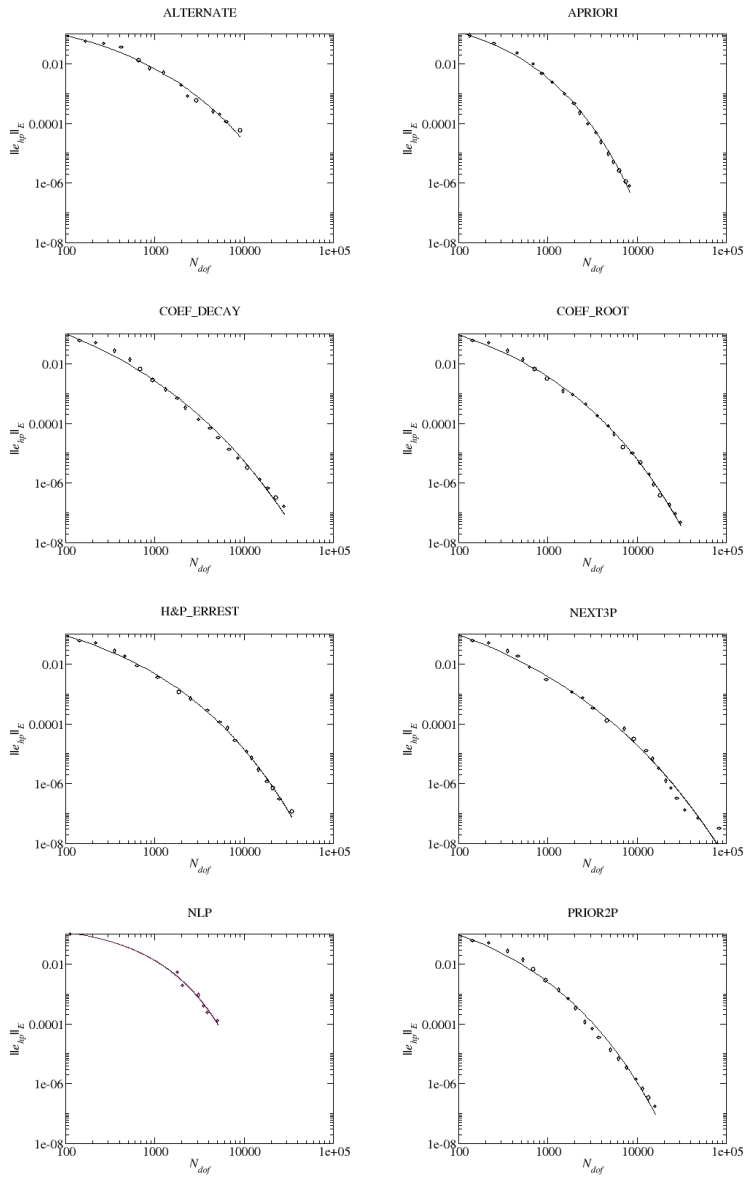

Figure 6: Convergence plots for the L-domain problem.

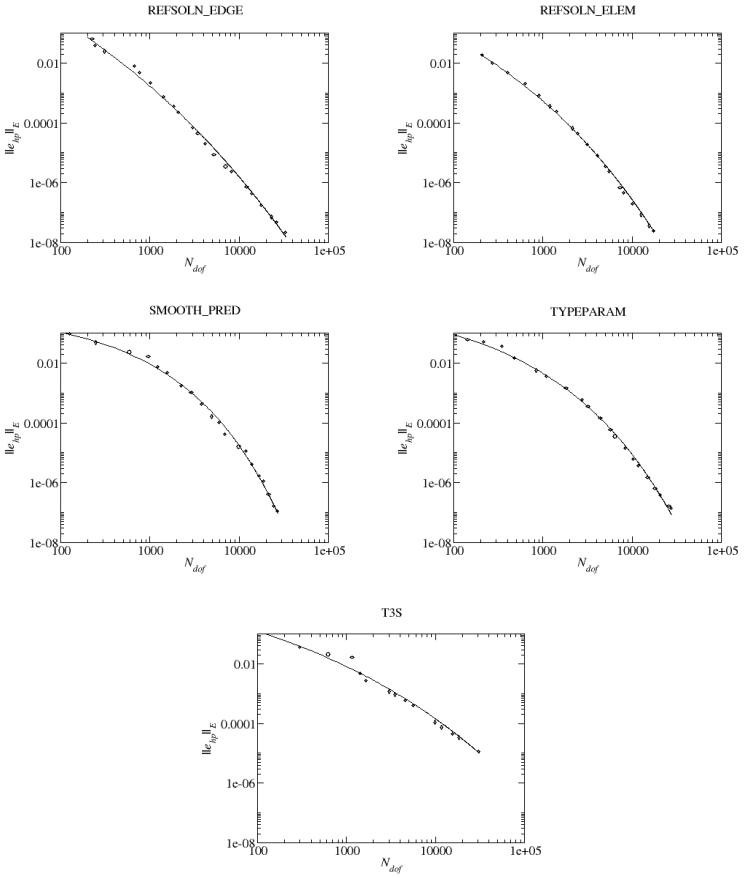

Figure 7: Convergence plots for the L-domain problem (continued).

strategy	exponent C
ALTERNATE	0.34
APRIORI	0.45
COEF_DECAY	0.23
COEF_ROOT	0.30
H&P_ERREST	0.30
NEXT3P	0.22
NLP	0.61
PRIOR2P	0.32
REFSOLN_EDGE	0.13
REFSOLN_ELEM	0.20
SMOOTH_PRED	0.41
TYPEPARAM	0.32
T3S	0.17

Table 2: Exponent on N_{dof} from the exponential least squares fit to the convergence data for the L-domain problem.

exponential rates of convergence with a few of them achieving an exponent of about 1/3 or more.

6 Conclusion and Future Work

Several *hp*-adaptive strategies have been presented in this paper. Although they were presented in the context of Poisson's equation in 2D, the strategies either apply directly to other classes of PDEs or are easily modified for other classes. Numerical results with two classic test problems demonstrate that all of the strategies can achieve exponential rates of convergence, although the 1/3 in the theoretical $N^{1/3}$ is not always achieved.

The purpose of this paper is to summarize the proposed strategies in one source and demonstrate that exponential rates of convergence can be achieved. It would be of interest to know which, if any, of the strategies consistently outperform the other strategies. Toward this end, future research involves a numerical experiment using a large collection of 2D elliptic PDEs that exhibit several types of difficulties, a uniform software base, and a consistent methodology.

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] 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.

[13] 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.

[14] 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.

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

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

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

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

[19] A. A. Novotny, J. T. Pereira, E. A. Fancello, and C. S. de Barcellos, *A fast hp adaptive finite element mesh design for 2D elliptic boundary value problems*, Comput. Methods Appl. Mech. Engrg. **190** (2000), 133–148.

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

[21] 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.

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

[23] 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.

[24] 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.

[25] 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.

[26] P. Šolín, *Hermes home page*, `http://hpfem.org`.

[27] ———, *private communication*.

[28] 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.

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

[30] 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.

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

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