Computing and Visualization in Science manuscript No.
(will be inserted by the editor)

Randolph E. Bank · **Hieu Nguyen**

hp Adaptive Finite Elements Based on Derivative Recovery and Superconvergence

Received: April 30, 2012 / Accepted: date

Abstract In this paper, we present a new approach to *hp*-adaptive finite element methods. Our a posteriori error estimates and *hp*-refinement indicator are inspired by the work on gradient/derivative recovery of Bank and Xu [12,13]. For element τ of degree p, $R(\partial^p u_{hp})$, the (piece-wise linear) recovered function of $\partial^p u$ is used to approximate $|\varepsilon|_{1,\tau} = |\hat{u}_{p+1} - u_p|_{1,\tau}$, which serves as our local error indicator. Under sufficient conditions on the smoothness of u, it can be shown that $\|\partial^p(\hat{u}_{p+1} - u_p)\|_{0,\Omega}$ is a superconvergent approximation of $\|(I-R)\partial^p u_{hp}\|_{0,\Omega}$. Based on this, we develop a heuristic *hp*-refinement indicator based on the ratio between the two quantities on each element. Also in this work, we introduce nodal basis functions for special elements where the polynomial degree along edges is allowed to be different from the overall element degree. Several numerical examples are provided to show the effectiveness of our approach.

Keywords *hp* Adaptivity, Finite Elements, *hp*-FEM, Nodal Basis

Mathematics Subject Classification (2000) 65N50, 65N55

1 Introduction

Adaptive *hp* finite elements were introduced by Babuška and his colleagues [3]. Since then, there has been much work on a priori error estimates for this version of finite elements [3, 18,20,21,7,4,19,22,5]. In particular, in [20,21,22,5], it was shown that the *hp* version is able to achieve an exponential rate of convergence, namely (in two-dimensional space)

$$\|u - u_{hp}\|_{H^1(\Omega)} \leq C\exp(-bN^{1/k}), \tag{1}$$

where C and $b > 0$ are independent of number of degrees of freedom N, and $k = 3$ if the solution has singularities and $k = 2$ if the solution is smooth.

Numerical experiments confirm (1) but also indicate that, in general, it is only achieved when the geometry of the mesh and the degrees of elements are properly chosen. This implies that the procedure that guides the adaptivity (in both h and p) is critical to the success of any *hp*-adaptive finite element method. Usually *hp*-adaptivity is guided by a posteriori error estimates and the so-call "*hp*-refinement indicator" that decides whether an element should be refined in h or in p.

Currently, there are several general approaches for deciding whether a given element τ should be refined in geometry or increased in degree. One common approach is to estimate the local regularity, the maximum value of m such that $u \in H^m(\tau)$, and use p-refinement when $p_\tau < [m]$. This can be done by utilizing multiple local error estimates with different values of degree p_τ, either smaller ones as in Süli, Houston, and Schwab [30] or larger ones as in Ainsworth and Senior [2]. Usually these error estimates are computed through solving local Neumann/Dirichlet problems that depend on the PDEs. Another approach is to estimate how the error will be reduced for the two types of refinement and make a decision based on the error reduction per degree of freedom, see Schmidt and Siebert [28], Demkowicz and his collaborators [27,16,15]. This often requires the computation of a so-called reference solution, obtained from refining all current elements in both h and p. For other approaches of formulating an *hp*-refinement indicator we refer to [18,23, 17,29,24,14], and in particular the survey by Mitchell and McClain [25]. For surveys of a posteriori error estimates for finite elements see [32,31,1,6].

In this paper, we introduce an *hp* version of finite element method that is reliable, efficient and versatile. Our overall development has three major components.

Bank: The work of this author was supported by the National Science Foundation under contract DMS-0915220.

Nguyen: The work of this author was supported by the National Science Foundation under contract DMS-0915220 and a grant from the Vietnam Education Foundation (VEF).

Bank: Department of Mathematics, University of California, San Diego, La Jolla, California 92093-0112. Email: rbank@ucsd.edu.
Nguyen: Department of Computer Science, University of California, Davis, Davis, California 95616. Email: htrnguyen@ucdavis.edu.

First, we formulate a posteriori error estimates based on the derivative recovery technique developed in [11,12,13]. In this approach, the local error for element τ of degree p is approximated by $|\hat{u}_{p+1}-u_p|_{1,\tau}$, where u_p is the interpolant of degree p and $\hat{u}_{p+1}$ is the hierarchical extension of degree $p+1$. These have previously been described for the cases of constant $p=1,2$ but the formulae in those cases do not generalize for larger p in a straightforward fashion [12,13]. To construct a general formula for arbitrary p, we introduce a particular basis for the error space $\mathscr{E}_{p+1}(\tau)$. Using this basis, $\hat{u}_{p+1}-u_p|_\tau$ can be expressed by a relatively simple formula in which all things are computable, except for the derivatives $\partial^{p+1}\hat{u}_{p+1}$. Motivated by [12,13], these derivatives are approximated by the superconverged recovered derivatives, $\partial(S^mQ(\partial^p u_{hp}))$. Here S is a smooth operator, Q is a L^2 projection and m is a small integer, typically one or two.

Second, we formulate an hp-refinement indicator. Our hp-refinement indicator is based on the superconvergence result that the recovered derivatives $S^mQ(\partial^p u_{hp})$ for elements of degree p have better than first order accuracy, provided the true solution has the required smoothness. Derivatives of these piece-wise linear polynomials are then used in our local error indicator $\eta_\tau=|\varepsilon_\tau|=|\hat{u}_{p+1}-u_p|_\tau$. To provide additional reliability for our local error indicator, especially in cases where the true solution is not smooth enough to support superconvergence, we require $\|\partial^p\varepsilon_\tau\|_{0,\tau}\equiv\|(I-S^mQ)\partial^p u_{hp}\|_{0,\tau}$, by introducing a scaling parameter α_τ that scales ε_τ to make this so. We have observed empirically that that α_τ is not only a good normalization but also a good hp-refinement indicator. As the superconvergence result only holds under appropriate assumptions on the smoothness of u, we expect $\alpha_\tau\approx 1$ if u is smooth enough on τ, and bigger than one otherwise. Therefore, we use the size of α_τ to decide if we should refine element τ in h or in p.

At first glance, our hp-refinement indicator has some similarities to that of Melenk and Wohlmuth [24]. However, while Melenk and Wohlmuth assume the smoothness to predict error estimates for potential refined elements (according to the optimal rate of convergence) we use smoothness to empirically verify that the superconvergence underlying our error estimate is valid. In addition, [24] relies on the accuracy of the approximate solution on the previous mesh while we use only information from the current mesh.

The third important component of our work is the construction of our basis functions. Unlike much of the available work, we use nodal basis functions with special treatment for transition elements which can have edges of higher degree. Initially, we limited the transition elements of degree p to have a single edge of degree $p+1$ [26]. This proved to be very inflexible for performing several mesh generation tasks, in particular the mesh regularization phase of the Bank-Holst parallel adaptive meshing paradigm [9,10]. Our current construction is more flexible as it allows one or two transition edges, each with (possibly different) degrees greater than p. Our overall space is a conforming C^0 finite element subspace with nodal basis functions and no hanging nodes.

The rest of this paper is organized as follows. In section 2, we construct our basis functions and the formulation of our local error estimates. In section 3, we describe our hp-refinement indicator and the algorithm for hp-refinement. In section 4, we present several numerical experiments that demonstrate the efficiency as well as reliability of our approach for a wide range of problems. In Section 5 we make some concluding remarks. Finally, in an appendix, we sketch the construction of nodal basis functions for transition elements for conforming three-dimensional tetrahedral finite element spaces.

2 Preliminaries

2.1 Basis Functions

In our study, we use nodal basis functions, rather than a more traditional hierarchical family of functions as in [25]. For a standard element of degree p, basis functions are defined by their values at nodal points of degree p as illustrated in Figure 1 (left). Specifically, the basis function of degree p associated with the nodal point having the barycentric coordinates $(\hat{c}_1,\hat{c}_2,\hat{c}_3)=(\hat{i}/p,\hat{j}/p,\hat{k}/p)$ is uniquely determined by

$$\phi(c_1,c_2,c_3)=C\prod_{i=0}^{\hat{i}-1}\left(c_1-\frac{i}{p}\right)\prod_{j=0}^{\hat{j}-1}\left(c_2-\frac{j}{p}\right)\prod_{k=0}^{\hat{k}-1}\left(c_3-\frac{k}{p}\right),$$

where (c_1,c_2,c_3) are the barycentric coordinates, and C is a constant given by

$$C^{-1}=\prod_{i=0}^{\hat{i}-1}\left(\hat{c}_1-\frac{i}{p}\right)\prod_{j=0}^{\hat{j}-1}\left(\hat{c}_2-\frac{j}{p}\right)\prod_{k=0}^{\hat{k}-1}\left(\hat{c}_3-\frac{k}{p}\right).$$

For elements sharing at least one edge with a neighbor of higher degree, called *transition elements*, nodal points of the higher degree element are used on the shared edge (see Figure 1, middle). This is in contrast to the minimum rule used in [16,25].

To illustrate the construction of the nodal basis for transition elements, consider the case of an element τ of degree p with one transition edge (edge three) of degree $p+1$. We define a special polynomial of degree $p+1$, $\tilde{\phi}_{p+1}$ which is zero at all standard nodal points of degree p of τ, and identically zero on edges one and two. At the barycentric coordinate (c_1,c_2,c_3) the value of $\tilde{\phi}_{p+1}$ is given by

$$\begin{cases}\prod_{k=0}^{(p-1)/2}(c_1-k/p)(c_2-k/p), & \text{for } p \text{ odd},\\ (c_1-c_2)\prod_{k=0}^{(p-2)/2}(c_1-k/p)(c_2-k/p), & \text{for } p \text{ even}.\end{cases}$$

The local finite element space of the transition element τ is given by $\tilde{\mathscr{P}}(\tau)=\mathscr{P}_p(\tau)\oplus\tilde{\phi}_{p+1}$, where $\mathscr{P}_p(\tau)$ is the standard finite element space of order p. To form a nodal basis for this space, we create $p+2$ nodal basis functions $\{\hat{\phi}_i\}_{i=1}^{p+2}$

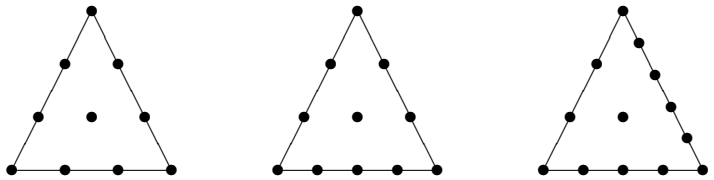


Fig. 1: A standard cubic element (left), a cubic element with one quartic edge (middle) and a cubic element with one quartic and one quintic edge (right).

associated with edge three and use standard basis functions at all other nodal points. The new edge basis functions are formulated as linear combinations of $\tilde{\phi}_{p+1}$ and standard basis functions of degree p, namely

$$\hat{\phi}_i = \sum_{j \in S_{trans}} \alpha_{ij}\phi_j + \beta_i \tilde{\phi}_{p+1}, \tag{2}$$

where S_{trans} is the index set for the $p+1$ nodal basis functions in $\mathscr{P}_p$ associated with the transition edge. In the general case, where the transition edge is of degree $p+k$, for $k>1$, we define the finite element space for τ as

$$\tilde{\mathscr{P}}(\tau) = \mathscr{P}_p(\tau) \oplus \{\tilde{\phi}_{p+1}(c_1-c_2)^m\}_{m=0}^{k-1},$$

and the transition basis functions $\{\hat{\phi}_i\}_{i=1}^{p+k+1}$ are given by

$$\hat{\phi}_i = \sum_{j \in S_{trans}} \alpha_{ij}\phi_j + \sum_{m=0}^{k-1} \beta_{i,m}\,\tilde{\phi}_{p+1}(c_1-c_2)^m. \tag{3}$$

Here α_{ij} are easily determined by evaluating both sides of equation (3) at the nodal points of degree p associated with the transition edge. And $\beta_{i,m}$ can be sequentially computed by taking $(p+m+1)$-th directional derivatives, $m = k-1, k-2, \ldots, 0$, of equation (3) in the tangential direction of the transition edge; the resulting set of linear equations for the $\beta_{i,m}$ is triangular. If a second transition edge is present, we can define nodal basis functions for it using an analogous construction without interfering with existing basis functions of the other transition edge.

Because the functions $\tilde{\phi}_{p+1}(c_1-c_2)^m$ are identically zero on edges one and two, the global finite element space constructed in this way is continuous everywhere, and thus remains conforming. Even though the formulation of nodal basis functions in this section is presented for triangular meshes in two-dimensional space it can be extended for tetrahedral meshes in three-dimensional space. We sketch such a construction in the Appendix.

2.2 Derivative Recovery and Error Estimates

Let Ω be the domain of the PDE and $\mathscr{T}_h$ be a shape regular triangulation of Ω of size h. Denote by $\mathscr{V}_h^{(p)}$ the space of continuous piece-wise polynomials of degree p associated with $\mathscr{T}_h$.

For a given function $u \in L^2(\Omega)$, we define its L^2 projection $Q_h u \in \mathscr{V}_h^{(1)}$ as the solution of the following variational problem

$$(Q_h u, v_h) = (u, v_h),\ \forall v_h \in \mathscr{V}_h^{(1)}. \tag{4}$$

Here $(\cdot,\cdot)$ denotes the inner product on $L^2(\Omega)$.

We also define smoothing operator $S_h : S_h = I - \lambda^{-1}A_h$, where $A_h : \mathscr{V}_h^{(1)} \to \mathscr{V}_h^{(1)}$ uniquely defined by

$$(A_h u_h, v_h) = (\nabla u_h, \nabla v_h) + (u_h, v_h), \quad \forall u_h, v_h \in \mathscr{V}_h^{(1)},$$
$$\lambda \equiv \rho(A_h) \simeq h^{-2}.$$

For the finite element approximation $u_{hp} \in \mathscr{V}_h^{(p)}$, let $\partial^p u_{hp}$ denote any of the (discontinuous piece-wise constant) p-th derivatives. The recovered p-th derivative is denoted by $R(\partial^p u_{hp}) \equiv S^m Q(\partial^p u_{hp})$. Here m is a small integer, typically one or two.

We recall the following superconvergence result from [13].

Theorem 1 *Let $u \in H^{p+2}(\Omega) \cap W^{p+1,\infty}(\Omega)$ and $u_{hp} \in \mathscr{V}_h^{(p)}$ be an approximation of u satisfying*

$$\|u - u_{hp}\|'_{p-1,\Omega} \lesssim h^2 |u|_{p+1,\Omega}, \tag{5a}$$

$$\|u - u_{hp}\|'_{p-1,\infty,\Omega} \lesssim h^2 |\log h|\, |u|_{p+1,\infty,\Omega}. \tag{5b}$$

where $\|\cdot\|'_{\cdot,\Omega}$ is the discrete norm defined by $\|\cdot\|'_{\cdot,\Omega} = \sum_{\tau\in\mathscr{T}_h} \|\cdot\|_{\cdot,\tau}$. Then

$$\|\partial^p u - R(\partial^p u_{hp})\|_{0,\Omega} \lesssim h\left(mh^{1/2} + \varepsilon_m\right) |||u|||, \tag{6}$$

$$\|\partial(\partial^p u - R(\partial^p u_{hp}))\|_{0,\Omega} \lesssim \left(mh^{1/2} + \varepsilon_m\right) |||u|||, \tag{7}$$

here $|||u||| = \|u\|_{p+2,\Omega} + |u|_{p+1,\infty,\Omega}$, $\varepsilon_m = (1-\kappa^{-1})^m$ with $\kappa = (Ch^2)\lambda$, for some constant C and small $m \in \mathbb{N}$.

Theorem 2 *Assume the hypotheses of Theorem 1. Then*

$$\|\partial^p(u - u_{hp})\|_{0,\Omega} \le \|(I-R)\partial^p u_{hp}\|_{0,\Omega} + Ch\left(mh^{1/2} + \varepsilon_m\right) |||u|||, \tag{8}$$

$$\|(I-R)\partial^p u_{hp}\|_{0,\Omega} \le \|\partial^p(u - u_{hp})\|_{0,\Omega} + Ch\left(mh^{1/2} + \varepsilon_m\right) |||u|||. \tag{9}$$

Furthermore, if there exists a positive constant $c_0(u)$ independent of h such that

$$\|\partial^p(u-u_{hp})\|_{0,\Omega} \geq c_0(u)h, \tag{10}$$

then

$$\left|\frac{\|(I-R)\partial^p u_{hp}\|_{0,\Omega}}{\|\partial^p(u-u_{hp})\|_{0,\Omega}} - 1\right| \lesssim \left(mh^{1/2}+\varepsilon_m\right)|||u|||. \tag{11}$$

Proof The proof of (8)-(9) is a simple application of the triangle inequalities

$$\begin{aligned}\|\partial^p(u-u_{hp})\|_{0,\Omega} &\leq \|(I-R)\partial^p u_{hp}\|_{0,\Omega} \\ &\quad + \|\partial^p u - R(\partial^p u_{hp})\|_{0,\Omega},\\ \|(I-R)\partial^p u_{hp}\|_{0,\Omega} &\leq \|\partial^p(u-u_{hp})\|_{0,\Omega} \\ &\quad + \|\partial^p u - R(\partial^p u_{hp})\|_{0,\Omega},\end{aligned}$$

and (6) in Theorem 1. The estimate (11) then follows from (8)-(9) and assumption (10). □

Note that the recovery process always involves the projection of piecewise constant functions into $\mathscr{V}_h^{(1)}$ followed by smoothing within that space; both of these operations are independent of p. Theorem 1 shows that $\partial R(\partial^p u_{hp})$ is a superconvergent approximation of $\partial^{p+1}u$ and Theorem 2 proves that $\|(I-P)\partial^p u_{hp}\|_{0,\Omega}$ is an asymptotically exact approximation of $\|\partial^p(u-u_{hp})\|_{0,\Omega}$. These observations lead to the construction of our local a posteriori error estimates discussed below.

Consider an element τ of degree p, we approximate the error $|u-u_{hp}|_{1,\tau}$ by $|\hat{u}_{p+1}-u_p|_{1,\tau}$, where u_p is the interpolant of degree p and $\hat{u}_{p+1}$ is the hierarchical extension of degree $p+1$. To illustrate our construction in detail, we write

$$\mathscr{P}_{p+1}(\tau) = \mathscr{P}_p(\tau) \oplus \mathscr{E}_{p+1}(\tau),$$

where the hierarchical extension $\mathscr{E}_{p+1}(\tau)$ consists of those polynomials of degree $p+1$ in $\mathscr{P}_{p+1}(\tau)$ that are zero at all degrees of freedom associated with $\mathscr{P}_p(\tau)$. It is not hard to show that the following set of functions form a basis for $\mathscr{E}_{p+1}(\tau)$

$$\psi_{p+1,i}(c_1,c_2,c_3) = \prod_{j=0}^{i-1}(c_1-j/p)\prod_{m=0}^{p-i}(c_2-m/p), \tag{12}$$

where $0 \leq i \leq p+1$.

Let $v_i=(x_i,y_i)$, $1\leq i\leq 3$ be the vertices of τ in counterclockwise orientation. And let $\{t_i\}_{i=1}^3$ denote the unit tangent vectors also with counterclockwise orientation, and $\{\ell_i\}_{i=1}^3$ the edge lengths of τ; see Figure 2. We have the following result.

Theorem 3 *Let $\{\psi_{p+1,i}\}_{i=0}^{p+1}$ be defined as in* (12)*. Then*

$$\hat{u}_{p+1}-u_p\big|_\tau = \sum_{i=0}^{p+1}\ell_2^i(-\ell_1)^{p+i-1}\frac{\partial_{t_2}^i\partial_{t_1}^{p+1-i}\hat{u}_{p+1}}{i!(p+1-i)!}\psi_{p+1,i}. \tag{13}$$

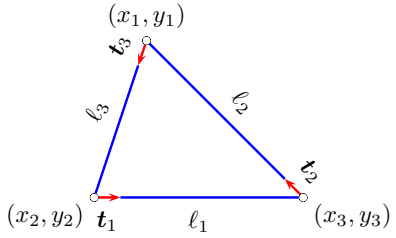


Fig. 2: Parameters associated with element τ.

Moreover, $\hat{u}_{p+1}-u_p\big|_\tau$ can be written as

$$\hat{u}_{p+1}-u_p\big|_\tau = \sum_{i=0}^{p+1}\sum_{j=0}^{p+1-i}\sum_{k=0}^{i}\gamma_{ijk}\,\theta_{ijk}\,\psi_{p+1,i}, \tag{14}$$

where

$$\gamma_{ijk} = (-1)^{p+1-i}\binom{p+1-i}{j}\binom{i}{k}(\partial_x^{j+k}\partial_y^{p+1-j-k}\hat{u}_{p+1}),$$

$$\theta_{ijk} = (x_1-x_3)^k(y_1-y_3)^{i-k}(x_3-x_2)^j(y_3-x_2)^{p+1-i-j},$$

Proof Since $\hat{u}_{p+1}-u_p\big|_\tau \in \mathscr{E}_{p+1}(\tau)$ and $\{\psi_{p+1,i}\}_{i=1}^{p+1}$ is a basis of $\mathscr{E}_{p+1}(\tau)$, we can write

$$\begin{aligned}\hat{u}_{p+1}-u_p\big|_\tau &= \sum_{i=0}^{p+1}C_i\psi_{p+1,i}\\ &= \sum_{i=0}^{p+1}C_i\prod_{j=0}^{i-1}(c_1-j/p)\prod_{m=0}^{p-i}(c_2-m/p).\end{aligned}$$

The coefficient C_i can be computed by taking the $(p+1)$-th derivative $\partial_{t_2}^i\partial_{t_1}^{p+1-i}$ of both sides and using the identity

$$\begin{pmatrix}\ell_1 t_1^t\\ \ell_2 t_2^t\\ \ell_3 t_3^t\end{pmatrix}(\nabla c_1\ \nabla c_2\ \nabla c_3) = \begin{pmatrix}0&-1&1\\1&0&-1\\-1&1&0\end{pmatrix}.$$

The equation (14) and formula for γ_{ijk} and θ_{ijk} then follow from equation (13), the definition of directional derivative, and the observation that:

$$\ell_2 t_2 = \begin{pmatrix}x_1-x_3\\y_1-y_3\end{pmatrix};\quad \ell_1 t_1 = \begin{pmatrix}x_3-x_2\\y_3-y_2\end{pmatrix}.$$

□

Next we define our *local error* $\varepsilon_\tau = \hat{u}_{p+1}-u_p\big|_\tau$. This error can be estimated using equation (14) and approximating the $(p+1)$-th derivatives $\partial_x^i\partial_y^{p+1-i}\hat{u}_{p+1}$ by

$$\begin{cases}\alpha_\tau\,\partial_y R(\partial_y^p u_{hp}), & i=0\\ \alpha_\tau\left\{\partial_x R(\partial_x^{i-1}\partial_y^{p+1-i}u_{hp}) + \partial_y R(\partial_x^i\partial_y^{p-i}u_{hp})\right\}/2, & 1\leq i\leq p\\ \alpha_\tau\,\partial_x R(\partial_x^p u_{hp}), & i=p+1\end{cases}.$$

Inspired by Theorem 2, the *scaling factor* α_τ is defined by

$$\alpha_\tau = \frac{\sum_{i=0}^{p} \binom{p}{i} \|(I-R)(\partial_x^i \partial_y^{p-i} u_{hp})\|_{0,\tau}^2}{\sum_{i=0}^{p} \binom{p}{i} \|\partial_x^i \partial_y^{p-i} \varepsilon_\tau\|_{0,\tau}^2}. \tag{15}$$

More discussion of α_τ follows in Subsection 3.1.

Then our *local error indicator* is defined by

$$\eta_\tau = |\varepsilon_\tau|_{1,\tau}. \tag{16}$$

Since ε_τ is a discontinuous piece-wise polynomial (of degree $p+1$) on the whole of Ω, we can also formally approximate errors in global norms and other functionals using ε_τ. For example,

$$|u-u_{hp}|_{1,\Omega}^2 \approx \sum_{\tau\in\Omega} |\varepsilon_\tau|_{1,\tau}^2 = \sum_{\tau\in\Omega} \eta_\tau^2.$$

3 *hp*-Refinement

With the error estimates established in Section 2, one can easily construct an adaptive *p*-refinement scheme where elements with large errors are selected for *p*-refinement. This scheme can then be used alternately with any *h*-adaptive refinement scheme to create adaptive meshes in both *h* and *p*. In some cases, this approach delivers reasonably good meshes and shows an exponential rate of convergence. However, the performance of such a simple scheme degenerates for problems with singularities or local rapid changes in the solution, see [26]. The reason is that both schemes will tend to refine elements in the critical regions. However, *p*-refinement might be much less effective than *h*-refinement in such regions due to the low local smoothness of the solution. To eliminate these drawbacks we would like to do both types of refinement in one refinement cycle and use *h*-refinement in critical regions and *p*-refinement everywhere else.

3.1 *hp*-Refinement Indicator

The most challenging question in formulating *hp*-adaptive FEMs is whether it is more advantageous to divide a given element in several child elements (*h*-refinement), or to increase its degree (*p*-refinement). In our *hp*-refinement procedure, the answer comes from the the scaling factor defined in (15).

According to Theorem 2 and definition of α_τ in (15), one should normally expect that $\alpha_\tau \approx 1$, which is likely to be the case where $(p+1)$-th derivatives of the true solution u are well-defined and of reasonable size. In regions near singularities or with rapid changes, we anticipate that our recovery scheme will have difficulties approximating the $(p+1)$-th derivatives of u. In these regions, the scaling factor ($\alpha_\tau > 1$) provides a normalization that partially compensates for poor approximation. Thus we can formulate our *hp*-refinement indicator, a Boolean value function PTEST(τ), simply based on the size of α_τ.

In all of the numerical examples provided in this paper, PTEST(τ) is defined as follows:

$$\text{PTEST}(\tau) = \begin{cases} 1 & \text{if } \alpha_\tau < 2\,\alpha_{ave} \quad (p\text{-refinement}) \\ 0 & \text{otherwise} \qquad\quad (h\text{-refinement}) \end{cases}, \tag{17}$$

where α_{ave} is the average of all α_τ in the mesh before refinement.

3.2 Automatic *hp*-Refinement Algorithm

In our strategy, prior to the actual refinement, local error indicators and scaling factors for each element are evaluated. Elements are then put into a *max-heap H* according to their errors. The element at the top of the heap is the one with largest error. It is considered for refinement first.

In many adaptive refinement schemes, elements are first marked for refinement, and then all marked elements are refined together as a group and mesh information is updated. In our scheme, we remove the element at the top of the heap, refine it, and immediately update our mesh data structures. The child elements are then introduced into the heap where they can be considered for further refinement. In *h*-refinement situations, typically two or more elements need to be refined simultaneously to maintain a conforming mesh, and the above procedure is applied to each of the parent elements.

In our study, a relaxed version of longest edge bisection *h*-refinement is used. When an element is refined in *h*, its two children inherit the degree from their parent. Local error indicators for new elements resulting from *h*-refinement are computed from derivative information that is inherited from the parent, as described in [12]. The scaling factor α_τ is also inherited from the parent element. On the other hand, when an element is *p*-refined, its degree is increased by one. We presently have no algorithm to update local error indicators for an element after *p*-refinement; we would need approximate derivatives of order $p+2$, whereas the approximate derivatives available from the parent are of order $p+1$ and are constant. Thus, in our study, elements that are *p*-refined are not allowed to be refined any further in the current adaptive step. A summary of our strategy for automatic *hp*-refinement is given in Algorithm 1.

In Algorithm 1, nd and nd_{trgt} are the current and the target degrees of freedom of the mesh respectively. The value of nd_{trgt} is given by the minimum of a user specified value and nd_{max}. The parameter nd_{max} is chosen to insure a geometric increase in the dimension of the finite element space. At the beginning of each adaptive enrichment step, the average degree of elements in the mesh is estimated by

$$p_{ave} = \left(\frac{nd}{nv}\right)^{1/2},$$

where nv is the number of vertices in the mesh. The value of nd_{max} is then given by

$$nd_{max} = nd \times \left(\frac{1+p_{ave}}{p_{ave}}\right)^r, \tag{18}$$

Algorithm 1 Automatic *hp*-Refinement

Calculate error estimates.
Build a max-heap H of elements according to their error estimates.
while $nd < nd_{trgt}$ **do**
  $\tau \leftarrow H(1)$;
  if $error(\tau) = 0$ **then** *Stop* **endif**
  if $PTEST(\tau)$ **then**
    p-refine τ
    Update mesh status, set $error(\tau) = 0$
  else
    h-refine τ *and neighboring elements as necessary*
    Update mesh status
    Estimate errors for new elements and add them to the heap
  end if
  Update error heap H
end while

where $r > 0$. Typically we chose $1 \le r \le 2$. The choice $r = 2$ gives $nd_{max} = 4 \times nd$ for the case of piece-wise linear elements, $p_{ave} = 1$. We have learned from experience that this growth produces near optimal observed rates of convergence for *h*-adaptive meshes for the case $p = 1$. We have also observed that decreasing geometric growth factors are necessary for increasingly higher degree elements in order for them to demonstrate near optimal convergence rates for *h*-adaptive meshes. Growing the dimensions of the finite element spaces more slowly of course requires more cycles through the adaptive feedback loop and therefore increases the cost in creating a solution for a specified number of degrees of freedom. Thus our choice of nd_{max} in (18) is an empirical attempt to balance the conflicting objectives of minimizing the computation cost while still achieving optimal or near optimal rates of convergence.

4 Numerical Examples

In this section, we illustrate the effectiveness of our error estimator and our automatic *hp*-adaptive strategy on some model problems. These problems have known solutions which possess different characteristics. All of our numerical examples were performed using PLTMG package, see [8], on an Apple MacPro with two quad core 3Ghz Xeon processors, 16gb of memory, and the gfortran and gcc compilers, version 4.4.4.

In these examples, we study the convergence of exact H^1-errors as functions of degrees of freedom N and of elapsed CPU time. The errors in *hp*-adaptivity are compared with that in *h*-adaptivity using elements of uniform degree *p*, $1 \le p \le 4$. For the *hp*-adaptivity, we will compare errors predicted by our error indicator with the exact ones. In addition, the exact errors in *hp*-adaptivity are fitted via least squares to an exponential of the form $C\exp(-bN^{1/k})$ as in (1) to estimate the rate of convergence. The optimal value for $1/k$ in (1) lies between $1/3$ and $1/2$ depending on the smoothness of the solution [20,21,22,5]. The average of scaling factor, α_{avg}, for each *hp*-mesh is also provided for reference. The degrees of freedom growth factor constant *r* in (18) is set $r = 2$ for first two examples and $r = 1$ for the last problem.

The CPU times are based on our preliminary implementation; in particular, the linear system solver, an algebraic multilevel method based on ILU, is one of the main costs, often representing over half the time. It seems likely that a more specialized solver taking advantage of the structure of these linear systems could substantially reduce the time; this will be the topic of a future investigation. However, since all experiments are using the same linear system solver, relative comparisons of time between different approaches should still be of interest.

In the first example, we consider the following nonlinear problem

$$-\nabla \cdot (a\nabla u) + e^u = f \qquad \text{in } \Omega = (0,1)\times(0,1), \tag{19a}$$
$$u = 0 \qquad \text{on } \partial_D\Omega, \tag{19b}$$

where a is the 2×2 diagonal matrix

$$a = \begin{pmatrix} .01 & \\ & 1 \end{pmatrix},$$

and f is chosen so that $u = x(1-x)^3y^5(1-y)$. The solution to this problem is very smooth, and our *hp*-procedure is able to produce optimal meshes with the least squares fits to the critical exponent $1/k = 0.64$, despite the anisotropic diffusion and the non-linearity, see Figure 3 and Table 1. The reason we get better-than-expected fitting result is due to the sub-optimal data points in early steps. These data points increases curvature as the accuracy "catch up" in later steps. This is also seen in [25] where other approaches of *hp*-adaptivity are tested.

Table 1: Errors in automatic *hp*-adaptive - Nonlinear Problem.

N	α_{ave}	Exact $\lvert e\rvert_{1,\Omega}$	Computed $\lvert e\rvert_{1,\Omega}$
9		1.514E-02	
25	2.654	1.269E-02	8.931E-03
100	1.477	4.465E-03	5.212E-03
400	1.405	1.993E-03	2.261E-03
1544	1.414	1.175E-04	1.369E-04
3231	1.421	1.194E-05	1.250E-05
4476	1.431	1.929E-06	1.710E-06
6753	1.415	2.481E-07	1.661E-07
8700	1.414	2.838E-08	2.808E-08
10602	1.415	3.290E-09	2.401E-09
13146	1.415	3.452E-10	1.970E-10
16259	1.414	2.581E-11	2.353E-11
19224	1.415	2.435E-12	2.193E-12
22429	1.415	1.502E-13	1.645E-13
27775	1.414	1.385E-14	2.019E-14
Fitting parameters:		$C = 1.42$, $b = 1.04$, $1/k = 0.64$	

In our second example, we consider an in-homogeneous Dirichlet problem communicated to us by Dr. Stefan Sauter.

$$-\nabla \cdot (a_\varepsilon \nabla u) + u = f \qquad \text{in } \Omega = (0,1)\times(0,1), \tag{20a}$$
$$u = u^{\varepsilon}_{\xi,p} \qquad \text{on } \partial_D\Omega, \tag{20b}$$

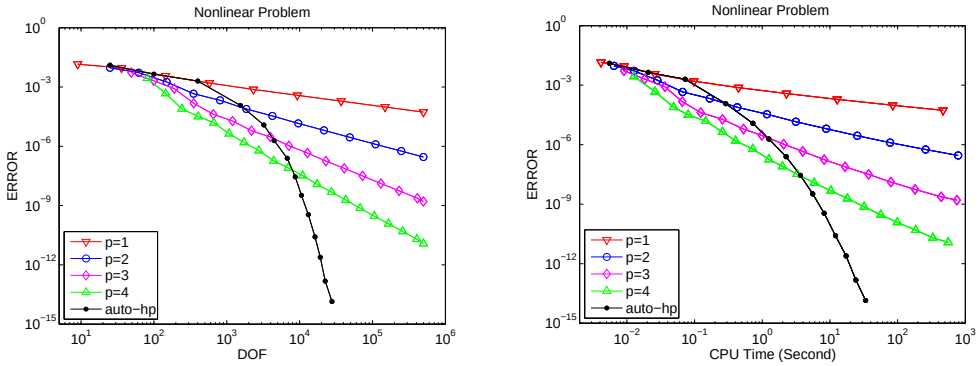


Fig. 3: Automatic *hp*-adaptive versus *h*-adaptive - Nonlinear Problem.

where $\varepsilon > 0$, $\xi = [\xi_1\ \xi_2]^T \in \mathbb{R}^2$, $p \in \mathbb{N}$ are given, and

$$a_\varepsilon = 10 + 2\cos\frac{2\pi x}{\varepsilon} + 2\cos\frac{2\pi y}{\varepsilon} + 2\cos\frac{2\pi(x+y)}{\varepsilon} + 2\cos\frac{2\pi(x-y)}{\varepsilon},$$

$$f(x,y) = \cos 2\pi\langle \xi, [x\ y]^T\rangle = \cos 2\pi(\xi_1 x + \xi_2 y),$$

$$u^\varepsilon_{\xi,p} = \sum_{\kappa\in\mathscr{I}_p} \phi_{\xi,p,\kappa}\, cos 2\pi\langle \xi + \kappa/\varepsilon, [x\ y]^T\rangle,$$

$$\mathscr{I}_p = \{(i,j) \in \mathbb{N}^2 : -p \le i,j \le p\}.$$

The coefficient vector $(\phi_{\xi,p,\kappa})_{\kappa\in\mathscr{I}_p}$ in the definition of $u^\varepsilon_{\xi,p}$ is the solution of the following system of equations

$$\sum_{\kappa\in\mathscr{I}_p} c_{\xi,\lambda,\kappa}\phi_{\xi,\lambda,\kappa} = r_\lambda, \quad \forall\lambda \in \mathscr{I}_p,$$

where

$$c_{\xi,\lambda,\kappa} = 4\pi^2 a_{\lambda-\kappa}\langle \xi + \kappa/\varepsilon, \kappa + \lambda/\varepsilon\rangle + b_{\lambda-\kappa}$$
$$r_\lambda = \delta_{(0,0),\lambda}.$$

The magnitude of the gradient, $|\nabla u|$, of the solution of this problem has a fine structure (see Figure 4) that can be captured only by proper adaptive meshes. Our *hp*-procedure creates meshes that deliver near optimal rate of convergence with the exponential constant $1/k = 0.42$, see Table 2 and Figure 5. The performance in term of time of *hp*-adaptivity is not as good as in the previous problem but is still better than *h*-adaptivity especially when high accuracy is sought.

In our third example, we consider the problem

$$\begin{aligned} -\Delta u &= 0 && \text{in } \Omega,\\ u &= f && \text{on } \partial_D\Omega,\\ \frac{\partial u}{\partial n} &= 0 && \text{on } \partial_N\Omega. \end{aligned}$$

Here Ω is the unit circle with a crack along the positive x axis, $0 \le x \le 1$; $\partial_D\Omega$ is the union of the boundary of the circle and the top of the crack; and $\partial_N\Omega$ is the bottom of the crack. The function g is chosen such that the exact solution is $u = r^{1/4}\sin(\theta/4)$. We note that u is not smooth ($u \in H^{5/4-\varepsilon}(\Omega)$) and it has a singularity at the origin.

Table 2: Errors in automatic *hp*-adaptive - Sauter Problem.

N	α_{ave}	Exact $\lvert e\rvert_{1,\Omega}$	Computed $\lvert e\rvert_{1,\Omega}$
9		1.117E-02	
25	2.166	4.588E-03	5.304E-03
100	1.470	3.033E-03	2.252E-03
400	1.475	2.346E-03	1.380E-03
1514	1.485	1.693E-03	1.739E-04
3275	1.559	8.698E-04	1.742E-04
6272	1.415	4.513E-04	5.821E-05
9897	1.463	2.254E-04	3.894E-05
12871	1.421	1.101E-04	2.205E-05
15940	1.421	5.599E-05	1.299E-05
19079	1.418	2.967E-05	1.029E-05
21817	1.536	1.659E-05	1.642E-05
29242	1.416	8.399E-06	5.306E-06
38082	1.418	3.636E-06	7.820E-07
52323	1.421	1.023E-06	2.631E-07
69701	1.420	2.525E-07	7.728E-08
90870	1.416	6.251E-08	2.429E-08
116550	1.417	2.128E-08	1.181E-08
148088	1.417	5.488E-09	1.996E-09
186929	1.423	1.537E-09	6.863E-10
234632	1.415	5.259E-10	2.252E-10
293551	1.414	2.430E-10	9.311E-11
Fitting parameters:		$C = 1.09$, $b = 1.10$, $1/k = 0.42$	

Our *hp*-procedure is able to detect this singularity and uses small elements of low degrees near the singularity and large elements of high degree elsewhere. This is shown in Figure 6 where degree and element size distributions of an adaptive mesh are presented. The *hp*-meshes also deliver an optimal rate of convergence with fitting exponent constant $1/k = 0.34$, see Table 3 and Figure 7.

5 Concluding Remarks

This project on *hp*-adaptivity in ongoing, and in this section we remark on some open issues and future directions.

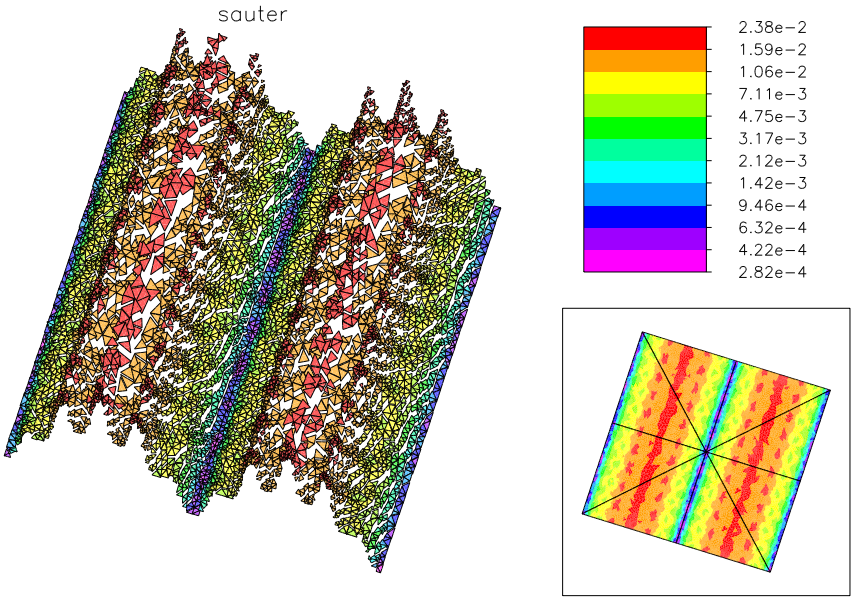


Fig. 4: Side view of $|\nabla u|$ in Sauter Problem.

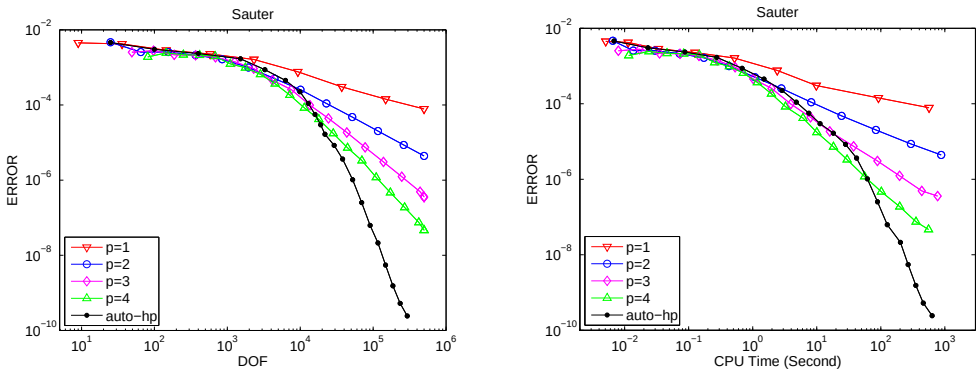
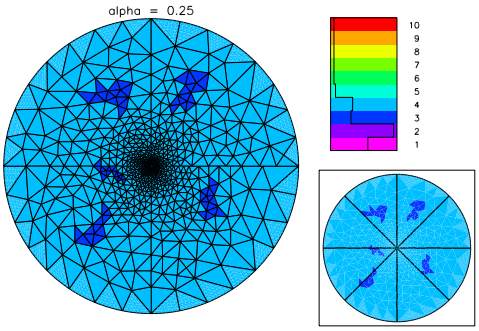


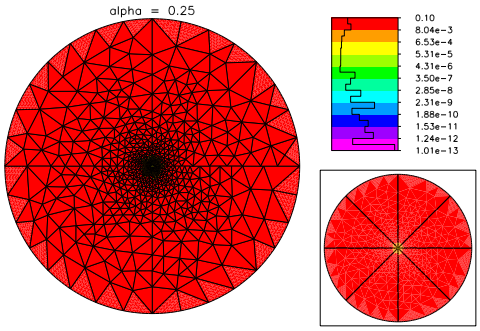
Fig. 5: Automatic hp-adaptive versus h-adaptive - Sauter Problem.

As noted in Section 3, in the case of p-refinement, at present we do not immediately compute a new error estimate for a p-refined element and allow possible further refinement in the current adaptive step. This is because the piece-wise constant approximate derivatives of order $p+1$ used in computing an error estimate for an element of degree p cannot be used to compute an error estimate for a newly refined element of degree $p+1$. In the current version of our code this limits p-refinement of an element to just one level in any single adaptive step. In the case of hp-refinement, this is not a large issue, because p-refinement is directed mainly to regions where the solution is smooth, and one level of p-refinement in a single adaptive step is sufficient in almost all situations. However in our current code, at the discretion of the user, one can specify h-refinement only or p-refinement only in any or all adaptive steps. In the case of p-refinement only, the ability to refine an element more than once (or alternatively, increase its degree from p to $p+k$ for $k>1$) would prove very useful. A procedure for doing this in a reliable and efficient fashion within the framework of our a posteriori error estimation procedure is a topic of current research.

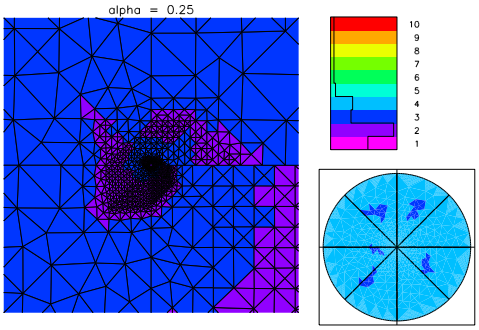
A second remark concerns the solution of linear systems created by the hp-adaptive process. An element of degree p has $O(p^2)$ basis functions, and thus has an element stiffness matrix with $O(p^4)$ nonzeros. With increasing density of the global stiffness matrix direct methods become more attractive as linear solvers, since relatively little fill-in is created if a good ordering of unknowns is used. For example, bubble functions associated with element interiors can be directly eliminated without causing any fill-in, a process sometimes called "static condensation." Matrices arising from hp-adaptive finite element spaces will have quite variable sparsity structures. Thus it seems that methods that combine both (multilevel) iteration and (block) Gaussian Elimination will be effective for solving such systems of equations. This is a topic of current research.



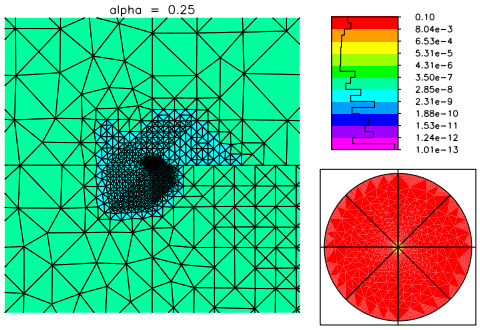
(a) Degree distribution (N=44433)



(c) Element size distribution (N=44433)



(b) Degree distribution near (0,0)



(d) Element size distribution near (0,0)

Fig. 6: *hp*-meshes and their close−ups at the singularity.

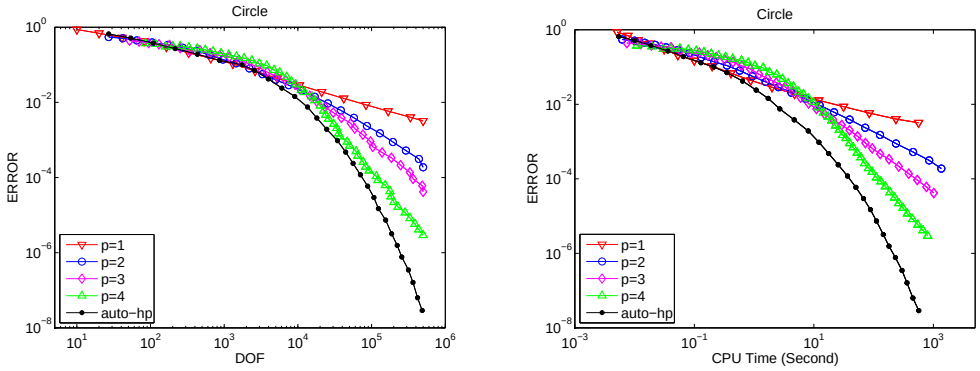


Fig. 7: Automatic *hp*-adaptive versus *h*-adaptive - Circle Problem.

References

1. Ainsworth, M., Oden, J.T.: A posteriori error estimation in finite element analysis. Pure and Applied Mathematics (New York). Wiley-Interscience [John Wiley & Sons], New York (2000)
2. Ainsworth, M., Senior, B.: An adaptive refinement strategy for *hp*-finite element computations. In: Proceedings of the International Centre for Mathematical Sciences Conference on Grid Adaptation in Computational PDEs: Theory and Applications (Edinburgh, 1996), vol. 26, pp. 165–178 (1998). DOI 10.1016/S0168-9274(97)00083-4. URL `http://dx.doi.org/10.1016/S0168-9274(97)00083-4`
3. Babuška, I., Dorr, M.R.: Error estimates for the combined *h* and *p* versions of the finite element method. Numer. Math. **37**(2), 257–277 (1981). DOI 10.1007/BF01398256. URL `http://dx.doi.org/10.1007/BF01398256`
4. Babuška, I., Guo, B.: The hp version of the finite element method for domains with curved boundaries. SIAM Journal on Numerical Analysis pp. 837–861 (1988)
5. Babuška, I., Guo, B.Q.: The *h-p* version of the finite element method for problems with nonhomogeneous essential boundary condition. Comput. Methods Appl. Mech. Engrg. **74**(1), 1–28 (1989). DOI 10.1016/0045-7825(89)90083-2. URL `http://dx.doi.org/10.1016/0045-7825(89)90083-2`
6. Babuška, I., Strouboulis, T.: The finite element method and its reliability. Numerical Mathematics and Scientific Computation. The Clarendon Press Oxford University Press, New York (2001)
7. Babuška, I., Suri, M.: The *h-p* version of the finite element method with quasi-uniform meshes. RAIRO Modél. Math. Anal. Numér. **21**(2), 199–238 (1987)
8. Bank, R.E.: PLTMG: A software package for solving elliptic partial differential equations, users' guide 10.0. Tech. rep., De-

Table 3: Errors in automatic *hp*-adaptive - Circle Problem ($r = 1$ in (18)).

N	α_{ave}	Exact $\lvert e\rvert_{1,\Omega}$	Computed $\lvert e\rvert_{1,\Omega}$
10		7.153E-01	
27	3.141	6.581E-01	7.154E-01
54	1.532	5.110E-01	3.143E-01
108	1.532	3.764E-01	3.524E-01
216	1.470	2.698E-01	2.516E-01
432	1.451	1.904E-01	1.933E-01
864	1.437	1.305E-01	1.451E-01
1747	1.429	9.995E-02	1.084E-01
2569	1.428	7.084E-02	7.735E-02
3958	1.427	4.208E-02	3.677E-02
6058	1.426	2.395E-02	2.118E-02
9060	1.424	1.438E-02	1.360E-02
13503	1.425	7.525E-03	8.938E-03
18072	1.421	3.818E-03	4.464E-03
24630	1.420	1.919E-03	2.291E-03
34397	1.418	9.537E-04	1.135E-03
44433	1.418	4.759E-04	5.717E-04
56103	1.417	2.375E-04	2.882E-04
70435	1.417	1.183E-04	1.434E-04
88372	1.417	5.894E-05	7.186E-05
107451	1.416	2.946E-05	4.557E-05
123756	1.416	1.486E-05	1.819E-05
155284	1.416	7.355E-06	8.041E-06
187098	1.416	3.185E-06	4.061E-06
222540	1.416	1.559E-06	2.228E-06
256925	1.416	7.683E-07	8.382E-07
316650	1.415	3.463E-07	3.824E-07
364793	1.415	1.609E-07	1.776E-07
420959	1.415	6.329E-08	8.833E-08
486234	1.415	2.899E-08	3.797E-08
Fitting parameters:		$C = 1.13,\ b = 1.22,\ 1/k = 0.34$	

partment of Mathematics, University of California at San Diego (2007)
9. Bank, R.E., Nguyen, H.: Domain decomposition and *hp*-adaptive finite elements. In: Domain Decomposition Methods in Science and Engineering XIX, Lecture Notes in Computational Science and Engineering. Springer-Verlag (2011)
10. Bank, R.E., Nguyen, H.: Mesh regularization in bank-holst parallel *hp*-adaptive meshing. In: Domain Decomposition Methods in Science and Engineering XX, Lecture Notes in Computational Science and Engineering. Springer-Verlag (in preparation)
11. Bank, R.E., Xu, J.: Asymptotically exact a posteriori error estimators. I. Grids with superconvergence. SIAM J. Numer. Anal. **41**(6), 2294–2312 (electronic) (2003). DOI 10.1137/S003614290139874X. URL http://dx.doi.org/10.1137/S003614290139874X
12. Bank, R.E., Xu, J.: Asymptotically exact a posteriori error estimators. II. General unstructured grids. SIAM J. Numer. Anal. **41**(6), 2313–2332 (electronic) (2003). DOI 10.1137/S0036142901398751. URL http://dx.doi.org/10.1137/S0036142901398751
13. Bank, R.E., Xu, J., Zheng, B.: Superconvergent derivative recovery for Lagrange triangular elements of degree p on unstructured grids. SIAM J. Numer. Anal. **45**(5), 2032–2046 (electronic) (2007). DOI 10.1137/060675174. URL http://dx.doi.org/10.1137/060675174
14. Bürg, M., Dörfler, W.: Convergence of an adaptive *hp* finite element strategy in higher space-dimensions. Appl. Numer. Math. **61**(11), 1132–1146 (2011). DOI 10.1016/j.apnum.2011.07.008. URL http://dx.doi.org/10.1016/j.apnum.2011.07.008
15. Demkowicz, L.: Computing with *hp*-adaptive finite elements. Vol. 1. Chapman & Hall/CRC Applied Mathematics and Nonlinear Science Series. Chapman & Hall/CRC, Boca Raton, FL (2007)
16. Demkowicz, L., Rachowicz, W., Devloo, P.: A fully automatic *hp*-adaptivity. In: Proceedings of the Fifth International Conference on Spectral and High Order Methods (ICOSAHOM-01) (Uppsala), vol. 17, pp. 117–142 (2002). DOI 10.1023/A:1015192312705. URL http://dx.doi.org/10.1023/A:1015192312705
17. Eibner, T., Melenk, J.M.: An adaptive strategy for *hp*-FEM based on testing for analyticity. Comput. Mech. **39**(5), 575–595 (2007). DOI 10.1007/s00466-006-0107-0. URL http://dx.doi.org/10.1007/s00466-006-0107-0
18. Gui, W., Babuška, I.: The h, p and h-p versions of the finite element method in 1 dimension. II. The error analysis of the h- and h-p versions. Numer. Math. **49**(6), 613–657 (1986). DOI 10.1007/BF01389734. URL http://dx.doi.org/10.1007/BF01389734
19. Guo, B.: The h-p version of the finite element method for elliptic equations of order 2m. Numerische Mathematik **53**(1), 199–224 (1988)
20. Guo, B., Babuška, I.: The h-p version of the finite element method - part 1: The basic approximation results. Computational Mechanics **1**(1), 21–41 (1986). URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0042893223&partnerID=40
21. Guo, B., Babuška, I.: The h-p version of the finite element method - part 2: General results and applications. Computational Mechanics **1**(3), 203–220 (1986). URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0039977843&partnerID=40
22. Guo, B.Q., Babuška, I.: Countable normed spaces and the h-p version of the finite element method. In: Numerical and applied mathematics, Part II (Paris, 1988), *IMACS Ann. Comput. Appl. Math.*, vol. 1, pp. 525–530. Baltzer, Basel (1989)
23. Houston, P., Süli, E.: A note on the design of *hp*-adaptive finite element methods for elliptic partial differential equations. Comput. Methods Appl. Mech. Engrg. **194**(2-5), 229–243 (2005). DOI 10.1016/j.cma.2004.04.009. URL http://dx.doi.org/10.1016/j.cma.2004.04.009
24. Melenk, J.M., Wohlmuth, B.I.: On residual-based a posteriori error estimation in *hp*-FEM. Adv. Comput. Math. **15**(1-4), 311–331 (2002) (2001). DOI 10.1023/A:1014268310921. URL http://dx.doi.org/10.1023/A:1014268310921
25. Mitchell, W.F., McClain, M.A.: A survey of *hp*-adaptive strategies for elliptic partial differential equations. Annals of the European Academy of Science (to appear)
26. Rachowicz, W., Oden, J.T., Demkowicz, L.: Toward a universal h-p adaptive finite element strategy part 3. design of h-p meshes. Computer Methods in Applied Mechanics and Engineering **77**(1-2), 181 – 212 (1989). DOI DOI:10.1016/0045-7825(89)90131-X. URL http://www.sciencedirect.com/science/article/B6V29-48050D5-T/2/d1f1c2fd306c9d472242b4b6ab51298d
27. Schmidt, A., Siebert, K.G.: A posteriori estimators for the h-p version of the finite element method in 1D. Appl. Numer. Math. **35**(1), 43–66 (2000). DOI 10.1016/S0168-9274(99)00046-X. URL http://dx.doi.org/10.1016/S0168-9274(99)00046-X
28. Solin, P., Dubcova, L., Dolezel, I.: Adaptive *hp*-FEM with arbitrary-level hanging nodes for Maxwell's equations. Adv. Appl. Math. Mech. **2**(4), 518–532 (2010)
29. Süli, E., Houston, P., Schwab, C.: *hp*-finite element methods for hyperbolic problems. In: The mathematics of finite elements and applications, X, MAFELAP 1999 (Uxbridge), pp. 143–162. Elsevier, Oxford (2000). DOI 10.1016/B978-008043568-8/50008-0. URL http://dx.doi.org/10.1016/B978-008043568-8/50008-0
30. Verfürth, R.: A review of a posteriori error estimation and adaptive mesh-refinement techniques, vol. 1. Wiley-Teubner (1996)

31. Wahlbin, L.B.: Superconvergence in Galerkin finite element methods, *Lecture Notes in Mathematics*, vol. 1605. Springer-Verlag, Berlin (1995)

Appendix: Three Dimensional Basis Functions

Although the discussion in this paper was restricted to two space dimensions, all the major features extend in straightforward fashion to three dimensional tetrahedral meshes. In this appendix, we consider the construction of continuous Lagrange nodal basis functions for transition elements in such a mesh.

To establish notation, let D denote the index set for degrees of freedom associated with a tetrahedron T and set

$$D_v = D_{v_1} \oplus D_{v_2} \oplus D_{v_3} \oplus D_{v_4}$$
$$D_e = D_{e_{12}} \oplus D_{e_{13}} \oplus D_{e_{14}} \oplus D_{e_{23}} \oplus D_{e_{24}} \oplus D_{e_{34}}$$
$$D_f = D_{f_1} \oplus D_{f_2} \oplus D_{f_3} \oplus D_{f_4}$$
$$D = D_v \oplus D_e \oplus D_f \oplus D_t$$

where D_v are the vertex degrees of freedom, D_e are the edge degrees of freedom, D_f are the face degrees of freedom, and D_t are the degrees of freedom associated with the interior of the element. For our family of C^0 Lagrange elements, a standard element of degree p has $|D| = (p+1)(p+2)(p+3)/6$, $|D_v| = 4$, $|D_e| = 6(p-1)$, $|D_f| = 4((p-1)(p-2)/2)$ and $|D_t| = (p-1)(p-2)(p-3)/6$. In local notation, the k-th face is opposite to the k-th vertex ($1 \le k \le 4$).

A transition element of degree p can have faces of degree $p+k$ for $k \ge 1$, with at least one face of degree p. Each of the six edges can be of degree $p+k$, with $k = \max(k_f, k_{f'})$, where f and f' are the two faces sharing that edge, with degrees $p+k_f$ and $p+k_{f'}$, respectively.

Faces

We first consider the construction of nodal basis functions of degree $q+1$ for a face, assuming we already have a set of nodal basis functions of degree q for that face. We note the most common situation here will be $q = p$, that is, a tetrahedral element of degree p sharing a face with with an element of degree $p+1$. There are $(q-1)(q-2)/2$ existing basis functions associated with the face. Let c_i, $1 \le i \le 4$ denote the barycentric coordinates for the tetrahedron, and suppose the face in question is face 4 (opposite vertex 4). We seek $q-1$ linearly independent polynomials of degree $q+1$ that are zero at all of the current nodes (of degree q). A convenient choice are the bubble functions

$$\eta_j = c_3 \prod_{k=0}^{j} (c_1 - k/q) \prod_{k=0}^{q-2-j} (c_2 - k/q)$$

for $0 \le j \le q-2$. These functions all contain the cubic bubble $c_1 c_2 c_3$ as a factor, and hence are identically zero on all edges and all faces other than face 4. They are also zero at all nodes of degree q of face 4 by construction, and will be zero at interior nodes of the element as well if $p = q$. If $q > p > 3$, then we need an additional step where the η_j are modified to be zero at the nodes corresponding to the set D_t

$$\hat{\eta}_j(x) = \eta_j(x) - \sum_{k \in D_t} \eta_j(x_k)\phi_k(x) \tag{22}$$

The $(q-1)(q-2)/2$ nodal basis functions of degree $q+1$, $\hat{\phi}_m$, $m \in \hat{D}_{f_4}$, are constructed from the current nodal basis for the face and the η_j (or the $\hat{\eta}_j$).

$$\hat{\phi}_m = \sum_{k \in D_{f_4}} \alpha_k \phi_k + \sum_{j=0}^{q-2} \beta_j \eta_j \tag{23}$$

The α_k are easily found by evaluating both sides of (23) at the nodes for the polynomial of degree q on face 4. The β_j can be found by differentiating both sides of (23) $q+1$ times ($\partial^{q+1} / \partial c_1^n \partial c_2^{q+1-n}$, for $1 \le n \le q$) yielding a consistent set of q equations in $q-1$ unknowns, which can be arranged in several ways into a lower triangular, nonsingular, and bidiagonal system that is easy to solve.

Note that the calculation of face basis functions for a given face is independent of the calculation for the other three faces and also independent of the calculation of edge and vertex basis functions. We now consider the issue of continuity of face basis functions constructed as above. Consider an element of degree p with a face of degree $p+k$, $k > 0$. Since our face space is built inductively, starting from the original space of degree p, at every step the space will continue to contain the original face space, so that the resulting transition element will always contain the polynomials of degree p as a subspace. Now consider the inductive step, moving from a face space of degree q to $q+1$. Polynomials in face spaces for all degrees must be identically zero on the other three faces (and as a consequence all edges and vertices). The requirement that members of the extension space be zero at interior degrees of freedom, and also the face degrees of freedom of degree q uniquely characterizes the extension subspace of dimension $q-1$. Thus the new nodal basis functions for the face nodes of degree $q+1$ are also uniquely defined. The restrictions of these basis functions to the given face are just the (unique) 2-dimensional nodal basis functions for a triangular element. Since this is also true for the neighbor element sharing the face, these basis functions, with support in only two tetrahedrons, are continuous.

Edges and Vertices

Once all of the face basis functions have been formed, we can calculate the edge and vertex basis functions. Like the two dimensional case, each edge can be treated independently. Let us consider the edge e_{12} shared by faces 3 and 4, with endpoint vertices 1 and 2 (local notation). Unlike the two dimensional case, we must insure continuity of the basis functions across faces sharing the given edge as well as

the edge itself. Note that a given edge can be shared by elements with several different values of p, and this must be accounted for in constructing a conforming basis. Similar to the construction of the face basis functions, assume we have constructed a nodal basis for a given edge (and its two endpoints) of degree q, and we wish to construct a basis of degree $q+1$

Motivated by the two dimensional case, we define

$$\eta_0 = \begin{cases} \prod_{k=0}^{(q-1)/2}(c_1-k/q)(c_2-k/q), & \text{for } q \text{ odd,} \\ (c_1-c_2)\prod_{k=0}^{(q-2)/2}(c_1-k/q)(c_2-k/q), & \text{for } q \text{ even.} \end{cases}$$

The function η_0 is a polynomial of degree $q+1$ that is identically zero on faces 1 and 2. If $q=p$, then η_0 will also be zero at nodes corresponding to D_t; if $q>p>3$, then η_0 should be modified analogously to (22). We now consider faces 3 and 4; on these faces, η_0 (or $\hat{\eta}_0$ if updated in (22)) will not be identically zero. However if $q=p_{f_3}$, where p_{f_3} is the degree of face 3, then η_0 will be zero at all the nodes. Otherwise, we must update η_0 or $\hat{\eta}_0$ using a formula like (22) with the index set D_t replaced by D_{f_3}. We make a similar modification if face 4 has degree $p_{f_4} \neq q$.

The $q+1$ nodal basis functions $\hat{\phi}_m$, $m \in \hat{D}_{e_{12}} \oplus D_{v_1} \oplus D_{v_2}$

$$\hat{\phi}_m = \sum_{k \in D_{e_{12}} \oplus D_{v_1} \oplus D_{v_2}} \alpha_k \phi_k + \beta \eta_0$$

(replace η_0 by $\hat{\eta}_0$ if necessary.) These α_k and β are computed exactly as in the two dimensional case. The modifications to the vertex basis functions at vertices v_1 and v_2 are independent of other potential modifications associated with other edges in the tetrahedron. As in the case of face basis functions, edge basis functions are computed recursively starting from the basis functions of degree p for the standard element.

We now consider in detail of issue of continuity of edge and vertex basis functions constructed as above. First, let us consider the case of an edge of degree $p+k$, $k>0$ and a face of degree $p+k$. We assume the face basis has already been constructed as described above. As with the face basis, at each step we start from an existing edge/vertex basis of degree q and create one of degree $q+1$ using a 1-dimensional extension space spanned by the function η_0. Since we begin with the original edge/vertex basis for the polynomials of degree p the transition element will always contain the polynomials of degree p as a subspace. For simplicity first consider the case where all three edges sharing the given face have the same degree $p+k$ as the shared face. Then at the end of the calculation for all three edges, the restrictions of the edge, vertex, and face basis functions to that face will be the complete set of nodal basis functions for a polynomial of degree $p+k$ for a 2-dimensional triangular element. Since this is also true for the element sharing the face, these basis functions are continuous across that face (and as a consequence, along edges and at vertices associated with that face as well).

Now suppose that the edge is of degree $p+k' > p+k \geq p$, where the face has degree $p+k$. Then we will start from a conforming and continuous edge/vertex basis of degree $p+k$ and create a basis of degree $p+k'$. At each step, moving from degree q to $q+1$, we create a 1-dimensional extension space spanned by η_0. The restriction of η_0 to the face is the same for both tetrahedrons sharing that face for q odd, but could differ by a minus sign if q is even, since the roles of c_1 and c_2 could be reversed in the neighbor element. However, in either case the restricted subspace is the same. Forming the edge/vertex nodal basis of degree $q+1$ from the basis of degree q and η_0 is an identical (1-dimensional) calculation in both elements (except for the sign of β if q is even and c_1 and c_2 have reversed roles in the neighbor tetrahedron), so edge/face basis functions of degree $q+1$ inherit continuity in the shared face from the continuity of the basis of degree q and η_0.