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

Adventures in Adaptivity

Randolph E. Bank · **Chris Deotte**

Received: April 12, 2016 / Accepted: October 21, 2016

Abstract In this work, we compare and contrast a few finite element h-adaptive and hp-adaptive algorithms. We test these schemes on three example PDE problems and we utilize and evaluate an a posteriori error estimate. In the process, we introduce a new framework to study adaptive algorithms and a posteriori error estimators. Our innovative environment begins with a solution u and then uses interpolation to simulate solving a corresponding PDE. As a result, we always know the exact error and we avoid the noise associated with solving. Using an effort indicator, we evaluate the relationship between accuracy and computational work. We report the order of convergence of different approaches. And we evaluate the accuracy and effectiveness of an a posteriori error estimator.

Keywords h-adaptive, hp-adaptive, finite elements, a posteriori error estimation, interpolation, convergence order, PDE

Mathematics Subject Classification (2010) 65N30, 65N15, 65N50

1 Introduction

Let $H(\Omega)$ denote a Sobolev space of interest, equipped with norm $\|\cdot\|_\Omega$, and let $\mathcal{S}_h \subset H$ denote a finite element space, associated with a tessellation $\mathcal{T}_h$ of Ω. In [4], it was shown that under weak assumptions

$$\|u-\mathcal{I}u\|_\Omega \le C\|u-v\|_\Omega \tag{1}$$
$$\|u-\mathcal{I}u\|_t \le C\|u-v\|_t \tag{2}$$

for all $v \in \mathcal{S}_h$. Here $u \in H$, $\mathcal{I}u \in \mathcal{S}_h$ is the usual interpolant of u, $t \in \mathcal{T}_h$, and $\|\cdot\|_t$ is the norm restricted to element t. In finite element analysis, interpolation error $u-\mathcal{I}u$ is often used as an upper bound for the error in various finite element approximations, here denoted $u_h \in \mathcal{S}_h$. Estimates (1)-(2) provide lower bounds for the error for *any* finite element approximation. Thus these bounds can be combined with the usual a priori estimates for u_h to see that

$$C_1\|u-\mathcal{I}u\|_\Omega \le \|u-u_h\|_\Omega \le C_2\|u-\mathcal{I}u\|_\Omega \tag{3}$$
$$C_3\|u-\mathcal{I}u\|_t \le \|u-u_h\|_t \tag{4}$$

In this work, we explore the practical implications of (3)-(4) in the context of adaptive finite element methods, and in particular h-adaptive and hp-adaptive feedback loops.

Inequalities (3)-(4) show that interpolation error is both efficient and reliable [13,14] for controlling an adaptive feedback loop of the type commonly employed in adaptive finite element calculations for solving partial differential equations. While interpolation error cannot in general be used for an a posteriori error estimate, in this work we create an environment based on interpolation error that allows us to study and evaluate various adaptive approaches independently of the PDE. Aspects such as approximate solution of linear and nonlinear systems, unknown exact solutions, and the a posteriori error estimation procedure make it difficult to focus exclusively on the adaptive procedure itself. Using interpolation error, we construct idealized "reference" adaptive procedures. We then employ these reference procedures to compare and evaluate various approaches. Indeed, during the course of our investigations, we analyzed and improved the adaptive approach used in the PLTMG software package

Bank: The work of this author was supported by the National Science Foundation under contract DMS-1318480, and DMS 1345013.

Deotte: The work of this author was supported by the National Science Foundation under contract DMS 1345013.

Bank: Department of Mathematics, University of California, San Diego, La Jolla, California 92093-0112. Email:rbank@ucsd.edu
Deotte: Department of Mathematics, University of California, San Diego, La Jolla, California 92093-0112. Email:cdeotte@ucsd.edu

[1]. The version described in this summary work is not the procedure in the currently available version of PLTMG (although it is quite similar), but it or its descendant will appear in future versions of the package.

We are also able to study the effectiveness of certain a posteriori error estimates within this environment. In particular, we compare the behavior of some fixed adaptive algorithm using interpolation error for the local error indicators with the same adaptive algorithm using some computable a posteriori error estimate for the the local error indicators. In this case, the observed differences can be attributed to the a posteriori error estimate.

While we think the results for various adaptive approaches are by themselves quite interesting, and reveal the advantages and disadvantages of the given approaches, we also believe that our general methodology for evaluating the different approaches is an equally important contribution. Indeed, while the PLTMG adaptive algorithms described here are the best that we have developed at this point in time, we expect further improvements to be made through the use of this environment.

The remainder of this paper is organized as follows. In Section 2, we describe the the model problems we chose to include in our testing environment. In Section 3, we present results for several h-adaptive strategies, for several values of p, the degree of the piecewise polynomial approximation space. In Section 4, we present results for several hp-adaptive strategies. In Section 5, we consider more carefully the effort of various adaptive approaches on the total work required to solve the problem, and in Section 6, were study the effectiveness of the recovered derivative a posteriori error estimate used in some of our experiments.

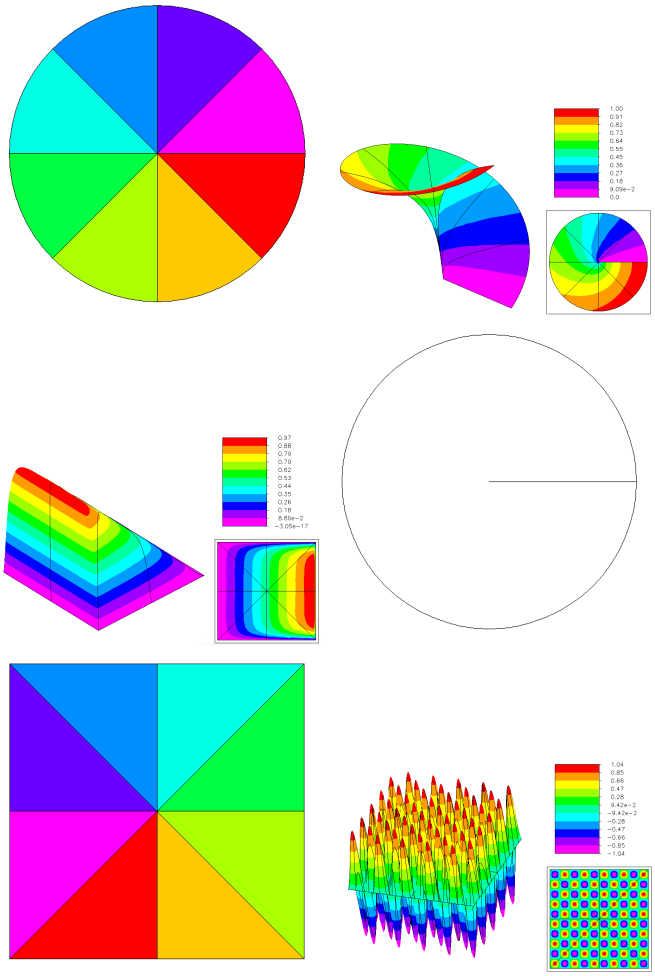

Fig. 1 The domains, the initial triangulations, and the function u's for the singular problem, boundary layer problem, and isotropic problem.

2 Preliminaries and Example Problems

We ran experiments on three different problems, chosen to exhibit typical challenges encountered by adaptive methods. In the first problem, the test domain Ω is the unit circle with a crack along the positive x axis. The function u is given by

$$u = r^{1/4}\sin(\theta/4)$$

(see Figure 1 top right). This function u is chosen to mimic a typical solution of the Laplace equation with singularity. For example, the elliptic PDE could be

$$-\Delta u = 0$$

in Ω. Homogeneous Dirichlet boundary conditions are imposed on the top of the crack, and homogeneous Neumann boundary conditions are imposed on the bottom. The combination of an internal angle of 2π and the change of boundary conditions at the crack tip leads to a singularity with leading term $r^{1/4}\sin(\theta/4)$. Imposing the Dirichlet boundary condition $u = r^{1/4}\sin(\theta/4)$ on the circle $r = 1$ makes this the exact solution of the boundary value problem.

In the second problem, the test domain is the unit square where the function u is given by

$$u = \left(x - \frac{e^{\beta(x-1)} - e^{-\beta}}{1 - e^{-\beta}}\right)\left(1 - \frac{e^{\gamma(y-1)} - e^{-\gamma}}{1 - e^{-\gamma}} - \frac{e^{-\gamma y} - e^{-\gamma}}{1 - e^{-\gamma}}\right)$$

where we took $\beta = 200$ and $\gamma = 20$ (see Figure 1 middle left). The first term is the solution of the two point boundary value problem

$$-u'' + \beta u' = \beta \quad \text{for } 0 \le x \le 1$$
$$u(0) = u(1) = 0.$$

This creates a boundary layer at $x = 1$. The second term creates additional (but weaker) boundary layers at $y = 0$ and $y = 1$, and enforces homogeneous Dirichlet boundary conditions on u. This problem is quite distinct from the first example, in that the refinement region is now a one-dimensional

curve rather than a single point. Also, while u has increasingly large higher derivatives, they are not singular as in the first example.

In the third problem, the test domain is the unit square where the function u is given by

$$u = \sin(10\pi x)\sin(10\pi y)$$

(see Figure 1 bottom right). This problem is quite distinct from the first two examples, in that the refinement region is now the entire two-dimensional domain. As with the boundary layer problem, this solution exhibits increasingly large but smooth higher derivatives.

Overall, we expect the singular problem to be the most challenging for adaptive methods, followed by the boundary layer problem, with the isotropic problem the least challenging. See Mitchell [10] for other possible test examples.

In these experiments we enriched the finite element subspace using several h and hp adaptive strategies. The local error indicators used in these adaptive methods were either the interpolation error for the exact solution [1] or error indicators generated by recovering derivatives using the interpolant as a proxy for the finite element solution. The recovery procedure is described in detail in [3]. In all experiments we started from the initial mesh of eight elements shown in Figure 1 and refined the mesh to one containing approximately 250,000 degrees of freedom.

We have selected only three example problems, a few example h-adaptive and hp-adaptive algorithms, and a single a posteriori error estimate for study in this work. While we think these are all interesting in their own ways for study, an important component of this work is to demonstrate through example the power of this approach as an experimental workbench. The environment of using the interpolant in place of the finite element solution allows us to discover and improve refinement procedures. Different refinement schemes can be compared to see which produces the most efficient reduction of error per computational cost without other aspects of the finite element solution confusing the data. And different error indicators can be compared and contrasted using this environment. See [11, 12] for some alternative approaches.

3 Experiments with h Refinement

We cannot compute an optimal finite element space with 250,000 degrees of freedom for purposes of comparison. Instead, we choose an adaptive procedure in which we use exact interpolation errors at every step, and use it as a "reference" adaptive procedure. For the case of h adaptive refinement, we employ an idealized version of the adaptive procedure used in PLTMG . In particular, the feedback loop is a three step process

$$\text{solve} \rightarrow \text{estimate} \rightarrow \text{refine}. \tag{5}$$

In this case the "solve" steps are skipped and we use the interpolant in place of the finite element solution. In the "estimate" step, we compute the interpolation error. This is given elementwise by

$$\|\nabla\{u - \mathcal{I}_p u\}\|_t \tag{6}$$

where $\mathcal{I}_p$ is the interpolation operator based on the usual nodes for the finite element space of continuous piecewise polynomials of degree p. In the "refine" step, we begin with a space of N_k degrees of freedom and compute a mesh with a target number of degrees of freedom given by

$$N_{k+1} \approx N_{trgt} = \min(4N_k, 250000) \tag{7}$$

degrees of freedom. The refinement process itself is a relaxed version of longest edge bisection as implemented in PLTMG . All elements are placed on a heap according to their errors, with an element with largest error at the root. The root element t is selected for refinement. Its longest edge neighbor (and perhaps a few others according to the relaxed longest edge criterion) are refined by bisection. The refined elements are removed from the heap, and their refined children elements are added to the heap. Errors for the children elements are given by interpolation errors for the exact solution. This procedure is repeated until the target value N_{trgt} has been achieved. Then the existing mesh is improved with respect to shape regularity using a combination of edge-swapping and mesh smoothing as described in the PLTMG users guide [1]. The adaptive loop is then repeated with the newly created mesh with N_{k+1} degrees of freedom; the outer feedback loop terminates when a mesh with (approximately) 250,000 degrees of freedom has been created.

We remark on some generic limits for our scheme. In the case of h-refinement, we check the proposed size of refined elements in terms of round-off error. If the elements become too small, many typical finite element calculations will suffer ill effects. For example, in computing the affine map from the reference element to an actual element in the mesh, one performs calculations of the form $v_i - v_j$, where v_i and v_j are vectors containing coordinates of triangle vertices. If $v_i \approx v_j$ due to the small size of the element, this and similar calculations will result in catastrophic cancellations that will create a cascade of undesirable consequences in the matrix and right hand side assembly, and subsequent linear system solution. Thus, if the proposed refined elements are too small, if

$$\sqrt{\frac{\|v_i - v_j\|^2}{\|v_i\|^2 + \|v_j\|^2}} < \varepsilon$$

[1] The interpolation error is computed exactly up to round off and numerical quadrature errors.

where ε is machine epsilon, then h-refinement is disallowed. In all of our reported experiments we used double precision floating point arithmetic, but we also did several quadruple precision calculations (not reported here) to verify that round-off effects were minimal for our reported results. Nonetheless, we think it remarkable that adaptive methods have become so effective that such issues even need to be addressed at the level of implementation.

To examine the effect of our recovery error estimator, we have also employed the reference adaptive procedure with error indicators rather than interpolation errors. In this scenario, in solve steps the finite element solution is again replaced by the interpolant. If we are using elements of degree p, derivatives of order $p+1$ are recovered from the interpolant using the same procedure normally applied to the finite element solution. We now summarize this procedure for approximating the interpolation error on element t. Our starting point is the representation

$$u - \mathcal{I}_p u = \sum_j \mathcal{F}_j(\partial^{p+1} u)\psi_j \tag{8}$$

where the ψ_j form a basis for the space of polynomials of degree $p+1$ that are zero at all nodes for degree p nodal basis functions on t and the coefficient functions $\mathcal{F}_j$ depend in an explicitly known and computationally accessible way on potentially all derivatives of u of order $p+1$, generically denoted $\partial^{p+1}u$. This error representation can be derived from Sobolev's counterpart (see [8], for example) of Taylor's theorem for weakly differentiable functions. Approximations to the derivatives $\partial^{p+1}u$ on element t are given by constants computed by a superconvergent recovery procedure that we now summarize. The derivatives of order p of the interpolant $\mathcal{I}_p u$, denoted $\partial^p \mathcal{I}_p u$, are piecewise constant. The recovery operator $\mathcal{R}\partial^p \mathcal{I}_p u$ consists of projecting these piecewise constant functions onto the space of continuous piecewise linear finite element functions using L_2-projection, followed by a smoothing step. This results in a globally superconvergent piecewise linear approximation of the order p derivatives $\partial^p u$. Then $\partial\mathcal{R}\partial^p\mathcal{I}_p u$ is a piecewise constant approximation of $\partial^{p+1}u$. The local error indicators η_t are given by

$$\eta_t = \left\| \nabla \left\{ \sum_j \mathcal{F}_j(\partial\mathcal{R}\partial^p\mathcal{I}_p u)\psi_j \right\} \right\|_t . \tag{9}$$

They depend only on $\mathcal{I}_p u$, and on the shape and size of the finite elements. See [3] for a more detailed discussion.

The refinement procedure itself is the same as for the interpolation error case described above, with one important difference. When an element is refined, its children elements inherit their derivative information from the parent, rather than computing it anew. This is how the procedure behaves in PLTMG , where the recovered derivatives are based on the finite element solution, and the recovery procedure is employed only in the estimate step.

We also studied the standard PLTMG refinement procedure using the recovered derivatives. This is similar to our reference procedure except we restrict the growth in the number of degrees of freedom on a given refinement step for reasons that will become clear below. The target value for a given refine step is still given by (7) but we exit the refine step if the current root element t satisfies

$$\eta_t^2 < \eta_{ave}^2/3 \tag{10}$$

where

$$\eta_{ave}^2 = \frac{\sum_t \eta_t^2}{T_k} \tag{11}$$

and the sum is taken over all elements existing at the beginning of the current refinement step, denoted by T_k.

We have also studied a four step adaptive procedure

$$\text{solve} \rightarrow \text{estimate} \rightarrow \text{mark} \rightarrow \text{refine} \tag{12}$$

which has become a common paradigm in adaptive methods [5,7,6,13,14]. In this procedure, a set of elements $\mathcal{S}$ are marked for refinement. The set of marked elements satisfies

$$\sum_{t\in\mathcal{S}} \eta_t^2 \geq \theta \sum_{t\in\mathcal{T}} \eta_t^2 \tag{13}$$

where $\mathcal{T}$ is the complete set of elements at the mark stage, and $0 < \theta \leq 1$ is a parameter. A similar formula holds if recovered error indicators are replaced by interpolation errors. In our setting we simulate this procedure as follows. Elements are placed in a heap as before. When an element is refined, it is added to the set $\mathcal{S}$; it is removed from the heap and its children are not added to the heap. When the errors in the set $\mathcal{S}$ exceed the threshold value, the refine procedure exits. In this way the set $\mathcal{S}$ will tend to contain elements with largest errors or error indicators.

In the first series of experiments, we employed these algorithms for h adaptive refinement for the the cases $p = 1,2,4$. For each experiment we provide four data items. First, we provide the error

$$\text{error} = \frac{\|\nabla(u - \mathcal{I}_p u)\|_\Omega}{\|\nabla u\|_\Omega}$$

for the finest mesh with (approximately) 250,000 degrees of freedom. We also provide the number

$$\text{loops} = L$$

where L is the number of outer loops through the adaptive feedback loop. We also fit the errors for some of L subspaces via least squares to a function of the form

$$AN_k^{-B/2}$$

and report the value

$$\text{order} = B.$$

Subspaces for smaller values of N_k were not included in order to obtain a better idea of the asymptotic behavior of each experiment. For piecewise polynomials of degree p, optimal behavior would be $B \approx p$. Based on the singularity, behavior for quasiuniform meshes for the circle problem would be $B \approx .25$, while for the boundary layer and isotropic examples, we expect $B \approx p$. Finally, we computed an effort indicator W using the formula

$$W_\alpha = \sum_{k=1}^{L} N_k^\alpha$$
$$\text{effort}_\alpha = W_\alpha / 10^6$$

The case $\alpha = 1$ would correspond to an implementation that exhibits optimal complexity $O(N)$ in every respect. We also consider the case $\alpha = 1.5$; this could arise, for example, if one employs an optimal order direct method with complexity $O(N^{3/2})$ for solving linear systems arising in the solve phase of the feedback loop. Occasionally, during our experiments, the number of degrees of freedom in the last two iterations were similar, $N_{L-1} \approx N_L \approx 250,000$, due to our use of (7). In this case our work estimate is artificially inflated because the two largest spaces are approximately the same size. In these cases, we provide an extrapolated effort value (ignoring N_L) in parenthesis before the actual value, that we believe provides a more accurate figure for purposes of comparison.

In Table 1 we summarize some experiments for the case $p = 1$. Some of these results are displayed graphically in Figure 2. The x axis is $\log_{10} N$ and the y axis is $\log_{10}(\text{error})$. We note that the reference procedure behaves in an optimal fashion using the exact interpolation errors. The PLTMG procedure and the reference procedure using recovered derivatives both track the optimal reference procedure quite well, although both are slightly less effective.

The marking procedure for $\theta = .5$ with either interpolation error or recovered derivatives tracks the optimal reference procedure almost exactly in terms of accuracy. However, it requires many more steps through the adaptive feedback loop in order to achieve this. Generally speaking, for such a marking procedure, one would expect that as θ is increased, the result would be more refinement during each loop, at the potential expense of decreasing the rate of convergence. We had originally planned to illustrate this point using, e.g., $\theta = 0.5, 0.25, 0.1$, but all give similar results. In the case of the singular problem, for example, the error is very large near the crack tip and relatively small everywhere else. Thus for a large range of θ, the marking scheme refines just a few elements near the crack tip on many adaptive steps, resulting in very slow growth in subspace dimension.

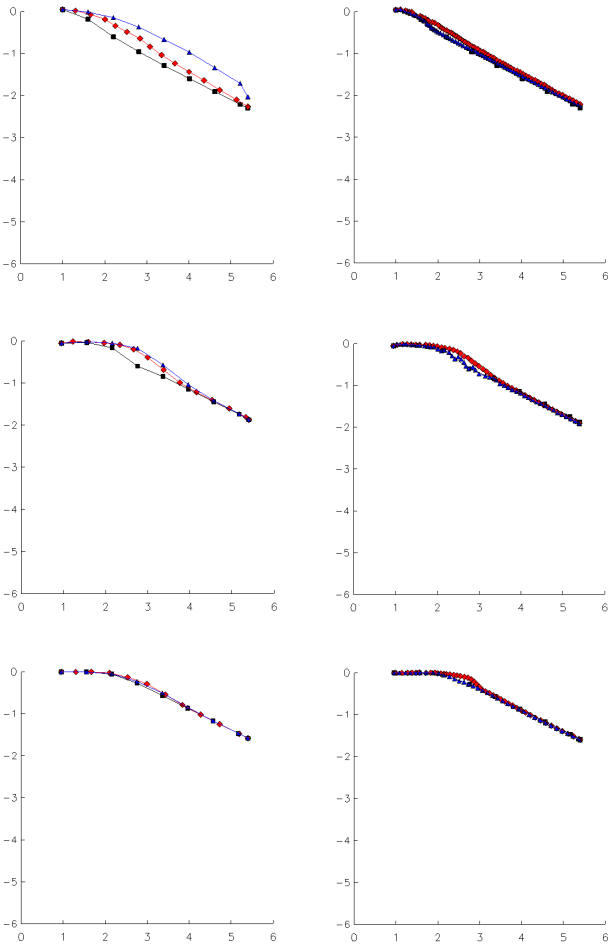

Fig. 2 h refinement for $p = 1$. Top row: Singular problem. Middle row: Boundary layer problem. Bottom row: Isotropic problem. Left column: reference method with interpolation error (black), reference method with recovered derivatives (blue), PLTMG method with recovered derivatives (red). Right column: reference method with interpolation error (black), marking method with interpolation error $\theta = .5$ (blue), marking method with recovered derivatives $\theta = .5$ (red).

We employed the marking strategy with $\theta = .99$, which reduced the number of adaptive steps without much sacrifice in terms of accuracy. We also show the result for $\theta = 1$, which refines every element on every loop, resulting in quasi uniform meshes and relatively poor accuracy for the singular problem. The reduction in convergence order was less severe for the boundary layer and isotropic problems, in part due to the smoothness of the functions in these cases.

This illustrates two issues with the marking paradigm. First, choosing the parameter θ is quite likely problem dependent, as is the case in these examples, and could be very sensitive to small changes in some ranges (e.g. $.99 \leq \theta \leq 1.0$ in the singular example). Second, without allowing for any element to be refined more than once based on its error indicator, the growth in subspace dimension will tend to be slow. For our relaxed longest edge bisection and similar schemes,

Singular problem					
h Refinement, $p=1$	Error	Order	Loops	$10^2 \cdot \text{Effort}_{1.0}$	$\text{Effort}_{1.5}$
reference-interpolation	5.06e-03	1.03	9	46.9	201
reference-recovered	9.29e-03	1.22	9	46.8	200
pltmg-recovered	5.45e-03	1.18	15	48.1	191
mark-recovered, $\theta = .5$	6.12e-03	1.03	68	148	505
mark-interpolation, $\theta = .5$	5.61e-03	1.05	70	149	522
mark-interpolation, $\theta = .99$	3.70e-02	0.93	38	148	536
mark-interpolation, $\theta = 1.0$	3.05e-01	0.25	14	60.0	252
Boundary layer problem					
h Refinement, $p=1$	Error	Order	Loops	$10^2 \cdot \text{Effort}_{1.0}$	$\text{Effort}_{1.5}$
reference-interpolation	1.35e-02	0.97	9	44.6	189
reference-recovered	1.37e-02	1.28	9	44.6	189
pltmg-recovered	1.34e-02	1.05	14	(48.1) 60.8	(207) 257
mark-recovered, $\theta = .5$	1.24e-02	1.03	45	114	406
mark-interpolation, $\theta = .5$	1.22e-02	1.02	52	(124) 142	(435) 531
mark-interpolation, $\theta = .99$	1.46e-02	1.06	21	54.6	213
mark-interpolation, $\theta = 1.0$	1.09e-01	0.84	17	51.4	199
Isotropic problem					
h Refinement, $p=1$	Error	Order	Loops	$10^2 \cdot \text{Effort}_{1.0}$	$\text{Effort}_{1.5}$
reference-interpolation	2.62e-02	0.99	9	44.6	189
reference-recovered	2.63e-02	1.04	9	44.6	189
pltmg-recovered	2.60e-02	1.05	12	48.9	202
mark-recovered, $\theta = .5$	2.51e-02	0.98	47	(100) 116	(359) 437
mark-interpolation, $\theta = .5$	2.51e-02	0.98	37	(99.6) 112	(361) 423
mark-interpolation, $\theta = .99$	2.65e-02	1.00	16	46.8	185
mark-interpolation, $\theta = 1.0$	3.06e-02	0.96	17	51.4	199

Table 1 h refinement for $p = 1$.

the upper limit will be around growth ≈ 2, since for $\theta = 1$, most elements would be refined with two child elements. A few could be refined more than once due to the rules of the particular bisection scheme. Thus we should expect growth < 2 for most values of $\theta < 1$.

Finally, as a general remark, we observe that the effort for an optimal calculation ($\text{Effort}_{1.0}$) is uniformly more than 100 times less than the corresponding calculation using suboptimal algorithms ($\text{Effort}_{1.5}$). We also note that methods that use many loops require more effort. We will return to this point and explore it more completely in Section 5.

We now consider the cases for h refinement for $p = 2$ and $p = 4$ together as they illustrate a trend for increasing p. We first note that the reference method using interpolation errors appears optimal with order of convergence of order $\approx p$. However, we note that for the first small values of N_j, the method is not behaving in an optimal fashion. This is likely due to limitations of the bisection process. For the case $p = 1$, 500 degrees of freedom correspond to roughly 1000 elements, which is enough for bisection to produce a smoothly graded mesh exhibiting an optimal rate of convergence for piecewise linear elements. On the other hand, in the case $p = 4$, 500 degrees of freedom correspond to approximately $1000/16 \approx 63$ elements. In the case of the singular problem, the bisection process is asked to create a smooth and more steeply graded mesh with just a few elements, which cannot be accomplished within the rules of longest edge bisection. Thus the convergence is not initially optimal, but becomes optimal as a larger number of elements are introduced into the mesh. Note the case $p = 4$ is worse than $p = 2$, and in general one can expect this effect to increase with increasing p.

Next we look at the reference method using the recovered derivative error indicators. Here we see that in both the $p = 2$ and $p = 4$ cases, an optimal convergence rate is not obtained for the singular problem. Indeed, the order of convergence is linear or worse. This illustrates a potential issue in allowing multiple refinements of an element during a single loop. When the recovered derivatives are inherited by children elements and subsequently used to compute error indicators, they also inherit the errors associated with the calculation of those recovered derivatives. These errors remain the same as in the original element for which they were computed, and are not reduced as they are inherited by the child elements. Thus an element that is refined many times in a single loop will tend to create a patch of quasiuniform elements as descendants. In the case $p = 1$, these patches were close enough to the optimal grading that they did not significantly affect the order of convergence. For $p = 2$ and even more for $p = 4$ they did have a severe adverse effect, and we expect this trend to continue with increasing p. This explains our use of (10) in PLTMG to restrict the growth of

Singular problem					
h Refinement, $p=2$	Error	Order	Loops	$10^2 \cdot \text{Effort}_{1.0}$	$\text{Effort}_{1.5}$
reference-interpolation	1.14e-04	2.19	8	39.6	166
reference-recovered	2.15e-02	1.01	8	39.8	167
pltmg-recovered	4.39e-04	2.20	26	77.6	277
mark-recovered, $\theta = .5$	3.00e-04	2.13	103	290	988
mark-interpolation, $\theta = .5$	1.31e-04	2.16	115	374	1301
mark-interpolation, $\theta = .99$	8.81e-04	1.93	78	292	1046
mark-interpolation, $\theta = 1.0$	3.21e-01	0.20	12	48.5	195
Boundary layer problem					
h Refinement, $p=2$	Error	Order	Loops	$10^2 \cdot \text{Effort}_{1.0}$	$\text{Effort}_{1.5}$
reference-interpolation	4.74e-04	2.03	8	39.2	165
reference-recovered	4.81e-04	2.41	8	38.8	163
pltmg-recovered	4.25e-04	2.15	15	57.8	226
mark-recovered, $\theta = .5$	3.55e-04	2.09	54	(158) 174	(557) 634
mark-interpolation, $\theta = .5$	3.50e-04	1.96	56	(159) 177	(540) 644
mark-interpolation, $\theta = .99$	4.07e-04	2.08	21	57.6	222
mark-interpolation, $\theta = 1.0$	2.13e-02	1.55	15	51.4	199
Isotropic problem					
h Refinement, $p=2$	Error	Order	Loops	$10^2 \cdot \text{Effort}_{1.0}$	$\text{Effort}_{1.5}$
reference-interpolation	1.24e-03	1.97	8	38.8	163
reference-recovered	1.25e-03	1.99	8	38.5	162
pltmg-recovered	1.18e-03	2.14	13	(47.4) 58.0	(189) 233
mark-recovered, $\theta = .5$	1.12e-03	2.01	51	(134) 156	(459) 568
mark-interpolation, $\theta = .5$	1.13e-03	2.00	39	124	440
mark-interpolation, $\theta = .99$	1.14e-03	2.03	15	(49.5) 64.6	(205) 271
mark-interpolation, $\theta = 1.0$	1.48e-03	1.96	15	51.4	199

Table 2 h refinement for $p = 2$.

Singular problem					
h Refinement, $p=4$	Error	Order	Loops	$10^2 \cdot \text{Effort}_{1.0}$	$\text{Effort}_{1.5}$
reference-interpolation	4.42e-07	4.42	7	37.1	156
reference-recovered	4.38e-02	0.73	7	36.7	154
pltmg-recovered	8.19e-06	4.36	60	246	794
mark-recovered, $\theta = .5$	1.59e-06	4.25	319	866	2667
mark-interpolation, $\theta = .5$	4.76e-07	4.11	167	628	2094
mark-interpolation, $\theta = .99$	3.27e-06	4.15	134	654	2256
mark-interpolation, $\theta = 1.0$	2.40e-01	0.20	11	62.0	264
Boundary layer problem					
h Refinement, $p=4$	Error	Order	Loops	$10^2 \cdot \text{Effort}_{1.0}$	$\text{Effort}_{1.5}$
reference-interpolation	2.22e-06	3.76	7	35.9	151
reference-recovered	2.21e-06	4.00	7	35.9	151
pltmg-recovered	1.27e-06	4.32	18	69.5	258
mark-recovered, $\theta = .5$	8.08e-07	4.23	90	306	1064
mark-interpolation, $\theta = .5$	7.68e-07	4.16	79	291	1010
mark-interpolation, $\theta = .99$	8.95e-07	4.19	20	(62.2) 78.6	(263) 319
mark-interpolation, $\theta = 1.0$	1.06e-03	2.82	13	51.4	199
Isotropic problem					
h Refinement, $p=4$	Error	Order	Loops	$10^2 \cdot \text{Effort}_{1.0}$	$\text{Effort}_{1.5}$
reference-interpolation	5.07e-06	3.75	7	35.9	151
reference-recovered	5.47e-06	3.74	7	35.9	151
pltmg-recovered	4.34e-06	4.35	17	(67.4) 81.0	(259) 321
mark-recovered, $\theta = .5$	3.76e-06	4.22	75	279	950
mark-interpolation, $\theta = .5$	3.80e-06	4.11	57	(212) 232	(724) 825
mark-interpolation, $\theta = .99$	4.96e-06	3.98	13	50.3	195
mark-interpolation, $\theta = 1.0$	6.14e-06	3.87	13	51.4	199

Table 3 h refinement for $p = 4$.

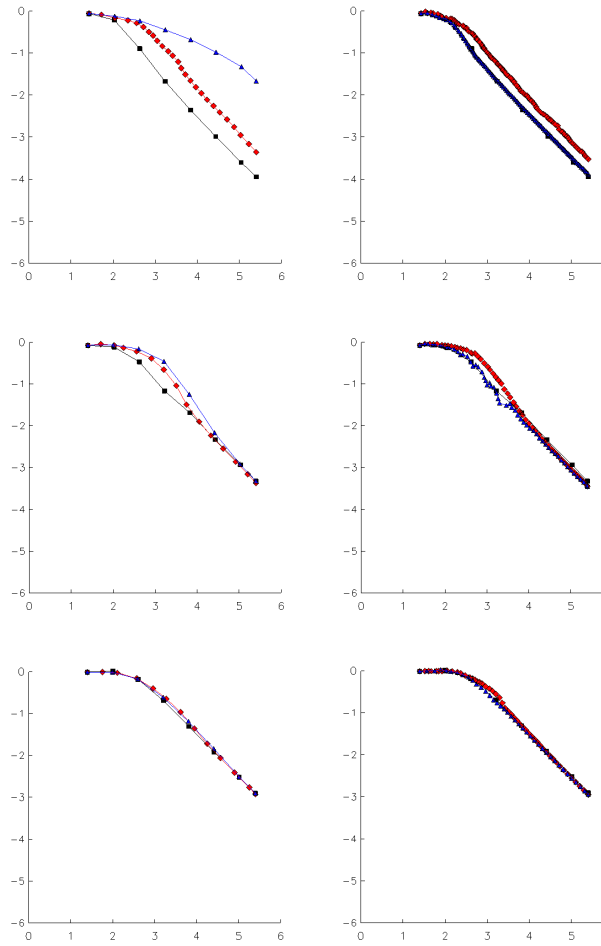

Fig. 3 h refinement for $p = 2$. Top row: Singular problem. Middle row: Boundary layer problem. Bottom row: Isotropic problem. Left column: reference method with interpolation error (black), reference method with recovered derivatives (blue), PLTMG method with recovered derivatives (red). Right column: reference method with interpolation error (black), marking method with interpolation error $\theta = .5$ (blue), marking method with recovered derivatives $\theta = .5$ (red).

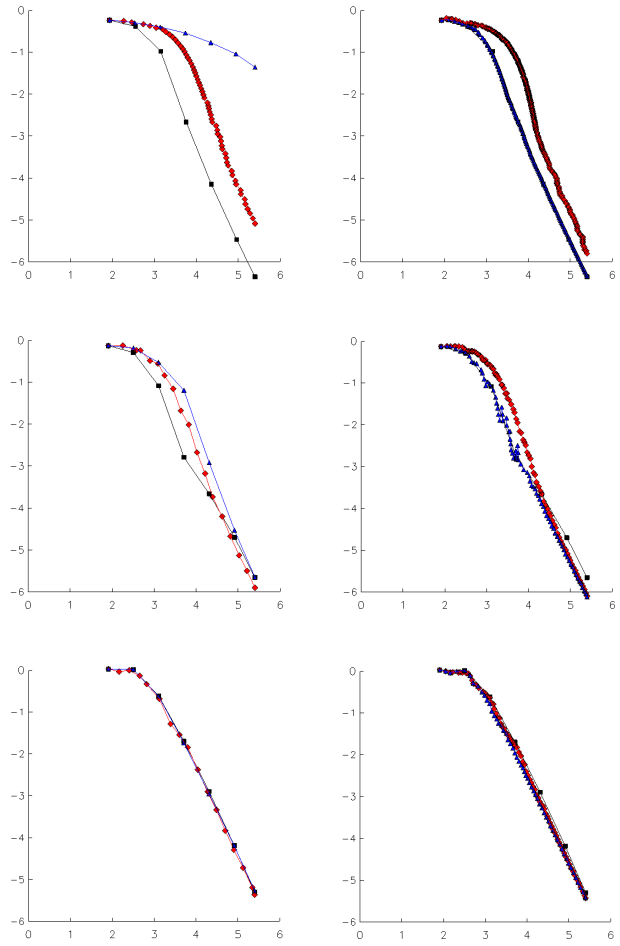

Fig. 4 h refinement for $p = 4$. Top row: Singular problem. Middle row: Boundary layer problem. Bottom row: Isotropic problem. Left column: reference method with interpolation error (black), reference method with recovered derivatives (blue), PLTMG method with recovered derivatives (red). Right column: reference method with interpolation error (black), marking method with interpolation error $\theta = .5$ (blue), marking method with recovered derivatives $\theta = .5$ (red).

subspace dimensions as p increases. However, we note that in the boundary layer and isotropic problems, this was less of an issue due to the smoothness of the solutions and the less severe grading of the mesh.

For the PLTMG experiments, we see that our use of (10) had the desired effect of producing optimal rates of convergence, but at the cost of an increasing number of adaptive feedback loops, especially in the case of the circle problem. We also note that the convergence is not optimal for the initial loops, with the effect becoming worse with increasing p. This is partly explained by the limitations of the bisection process noted above, but it also exposes an issue with the derivative recovery process. In particular, while the ansatz (8) seems appropriate in regions where the function u is very smooth, it is clearly inappropriate near the crack trip. However, the recovery process is purely algebraic, and so recovers approximate derivatives regardless of whether they actually exist. Fortunately, it turns out to be easy to spot these bogus derivatives when they occur, and that provides an important ingredient of our hp adaptive algorithm described below. In the h adaptive setting, the recovered derivatives are wrong, but still lead to very large error indicators for elements near the crack tip. At the beginning, all of the elements are near the crack tip, and hence all of the elements have error indicators strongly influenced by this "regularity gap." As the mesh is refined, the regularity gap strongly influences only elements near the crack tip (in terms of graph distance), and that number remains approximately constant as the total number of elements increases. Here again this effect becomes more pronounced with increasing p as the regularity gap increases with increasing p.

In viewing the marking strategy results, we see that for the cases where interpolation errors were used and $\theta = .5$, they behave in a similar fashion to the case $p = 1$. The main difference is that the number of loops shows an undesirable increase with increasing p, especially in the case of the circle problem. When recovered derivative error indicators are used, they show suboptimal convergence for small values of N_j consistent with the explanation above for similar behavior in the PLTMG method.

For the case $\theta = .99$ using interpolation errors, the number of loops is reduced somewhat, with little change in the order of convergence for the case $p = 4$. When $\theta = 1$, the meshes become quasiuniform and while loops decreases drastically for the circle problem, the approximation order $\approx .25$. Now there is also a less drastic reduction of order for the boundary layer problem, but still fairly good approximation for the isotropic problem.

4 Experiments with hp Refinement

We now consider the case of hp refinement. With elements of differing p values, special "transition elements" with one or two edges of higher degree than the overall element are needed to insure a conforming finite element space. These elements are described in detail in [2] but the details are not so important in the current context.

The reference hp method follows the basic lines of the h method. The subspace dimension increases according to (7) and is limited by (10). The new feature is a test for deciding whether to refine the root element in the heap by bisection (h refinement) or increasing the polynomial degree by one (p refinement). Using interpolation error, we can directly evaluate the effect of h refining our root element into four child triangles created by connecting the root triangle's edge midpoints. Let η_h be the error resulting from this h refinement and η_t be the original error. Given that the root element has polynomial degree p, then we choose p refinement if

$$\eta_h < \eta_t \left(\frac{1}{2}\right)^{p/2}$$

Otherwise, we choose h refinement. We expect this to be satisfied if the solution is smooth, whereas this should fail if the solution is not sufficiently regular.

Our p-refinement is limited to elements of at most $p = 9$. This is due to accuracy limitations of the suite of quadrature formulas used in the PLTMG software. As a practical matter, a maximum element degree is needed in any event due to round-off and stability considerations. In our hp method, if a proposed h-refinement is disallowed due to round-off considerations, it is considered for p-refinement instead. Similarly, if a proposed p-refinement is disallowed due to our degree limitation, it is considered for h-refinement. It is possible to construct sufficiently small elements of maximal degree that can no longer be refined at all. This possibility did not influence our experiments here, but in large parallel adaptive calculations involving millions of degrees of freedom, this limit has been achieved.

The hp method in PLTMG has several new features relative to the h method. First, when an element is refined in p it is removed from the heap. If it had degree p and is increased to degree $p+1$, it would require recovered derivatives of order $p+2$ to compute an error indicator as in (9). The existing recovered derivatives are of order $p+1$, and thus have no value for the p refined element. At present, there is no computationally inexpensive way to create derivatives of order $p+2$, so p refinement is restricted to just one level in any refine step.

Second, given an element t with degree p, if

$$\frac{(\sum_t \eta_t{}^2)^{1/2}}{\|\nabla \mathcal{I}u\|_\Omega} > \left(\frac{1}{3}\right)^p \tag{14}$$

only h refinement is allowed. This partially addresses the issue of potentially poor quality recovered derivatives for higher degree elements on very coarse meshes.

In addition to these controls, the main new issue arising in hp refinement is the choice of h or p refinement in the generic situation when both options are allowed. This is done by detecting when the derivative recovery failed for the given element; the cause is assumed to be a regularity gap. In particular we compute the ratio

$$\left(\frac{\sum_k \|\partial_k^p \mathcal{I}_p u - \mathcal{R}\partial_k^p \mathcal{I}_p u\|_t^2}{\sum_k \|\partial_k^p \sum_j \mathcal{F}_j(\partial \mathcal{R}\partial^p \mathcal{I}_p u)\psi_j\|_t^2}\right)^{1/2} = \alpha_t \tag{15}$$

where ∂_k^p sums over all derivatives of order p. If the solution u is smooth in t and has all derivatives of order p, then $\alpha_t \approx 1$. When this fails, typically α_t becomes quite large. In PLTMG , if $\alpha_t > 2$ we choose h refinement; otherwise we choose p refinement. In either case, we rescale the the coefficients as $\mathcal{F}_j(\partial \mathcal{R}\partial^p \mathcal{I}_p u)\alpha_t$, such that when (15) is evaluated with the rescaled coefficients, the new $\alpha_t = 1$.

hp refinement is supposed to exhibit an exponential rate of convergence. Rather than estimate the order of convergence, in our hp experiments we fit data for most of the L subspaces via least squares to a function of the form

$$A\exp(BN^C)$$

and report the value

$$\text{exponent} = C$$

rather than the order. For problems with singular behavior, the optimal value of exponent $= 1/3$ [9].

The results for some hp experiments are reported in Table 4 and Figure 5. The x axis is $\log_{10} N$ and the y axis is

Singular problem					
hp Refinement	Error	Exponent	Loops	$10^2 \cdot \text{Effort}_{1.0}$	$\text{Effort}_{1.5}$
reference-interpolation	3.50e-10	0.23	9	46.5	199
pltmg-recovered	7.97e-08	0.30	38	225	831
pltmg-recovered $p = 1,2,4$	1.71e-06	0.28	28	117	408
Boundary layer problem					
hp Refinement	Error	Exponent	Loops	$10^2 \cdot \text{Effort}_{1.0}$	$\text{Effort}_{1.5}$
reference-interpolation	1.03e-10	0.19	9	44.2	187
pltmg-recovered	1.14e-11	0.33	31	185	675
pltmg-recovered $p = 1,2,4$	1.19e-06	0.22	17	73.0	278
Isotropic problem					
hp Refinement	Error	Exponent	Loops	$10^2 \cdot \text{Effort}_{1.0}$	$\text{Effort}_{1.5}$
reference-interpolation	4.50e-11	0.29	9	44.3	187
pltmg-recovered	4.72e-11	0.60	22	157	603
pltmg-recovered $p = 1,2,4$	4.66e-06	0.35	15	74.7	285

Table 4 *hp* refinement.

$\log_{10}$(error). Note that the range of the y axis is different than the range of the y axis in the figures for the h experiments. We first discuss the top row images in Figure 5, where convergence histories for the hp reference method is displayed along with the h reference method for $p = 1, 2, 4$. The h reference convergence histories for $p = 2$ and $p = 4$ exhibit the problems on coarse grids previously discussed. The hp reference method may encounter some of the same issues on coarse meshes as the h refinement methods, however, on the finer meshes, the reference hp method does exhibit exponential convergence. As expected, the hp curve lies below all the h refinement curves.

The PLTMG hp adaptive method does a reasonable job. It takes 38 loops for the circle problem, and the final error is larger than the reference method, but it does exhibit exponential convergence. As a point of contrast we also tested the PLTMG adaptive method but allowed only the elements of degrees $p = 1, 2, 4$ to participate. When an element of degree two is p refined, it becomes an element of degree four. Elements of degree four can only be h refined. There are several possible advantages for this approach. First, since these elements are node-nested, there is a simple and obvious hierarchical basis associated with the spaces, simplifying basis construction and assembly procedures. There are also classical hierarchical basis multigrid schemes that could be used to solve the associated linear systems. The hierarchical structure also allows the possibility of simple hierarchical a posteriori error estimates to be employed. On the negative side, such a scheme can at best asymptotically behave like an h adaptive method for $p = 4$. But we note that up to 250,000 degrees of freedom, it tracks the PLTMG hp adaptive scheme very closely. It is less accurate for 250,000 degrees of freedom; by this point the hp method has created elements of degree larger than four. But it might prove to be an attractive alternative to more complex hp adaptive procedures. Also note that for the circle problem, in 28 loops it produced an error similar to the h-adaptive strategy for $p = 4$ that required 60 loops and about twice the effort. For the smoother boundary layer and isotropic problems, it performed comparably to the h-adaptive PLTMG strategy for $p = 4$.

5 Refinement Efficiency

It's difficult to compare the efficiency of different adaptive strategies by only looking at the final error and effort. A better comparison can be made by plotting the entire curve $\log_{10}(10^6 \times \text{effort}_{1.5})$ versus $\log_{10}$(error).

From these plots, there are three interesting things to note. First, on all three problems, the PLTMG refinement scheme is more efficient than the marking scheme. Second, for the point singularity and boundary layer problem (Fig 6), the PLTMG hp refinement scheme and the $hp-1,2,4$ refinement scheme are more efficient than corresponding h refinement schemes as should be. Third, for the isotropic problem (Fig 6), the PLTMG hp refinement scheme is less efficient than corresponding h refinement schemes until the error becomes smaller than 10^{-4}. This is due to (14) that limits the speed for p refinement. For this very smooth problem, allowing earlier p refinement would have been desirable. This indicates that the PLTMG hp refinement scheme has room for improvement.

The reference hp refinement scheme is more efficient than the corresponding reference h refinement schemes on all problems as it should be. These plots are not pictured, but they would appear the same as Figure 5 with the axis relabeled (since all the reference schemes use the same number of adaptive loops with the same sequence of $N_1, N_2, \ldots N_L$).

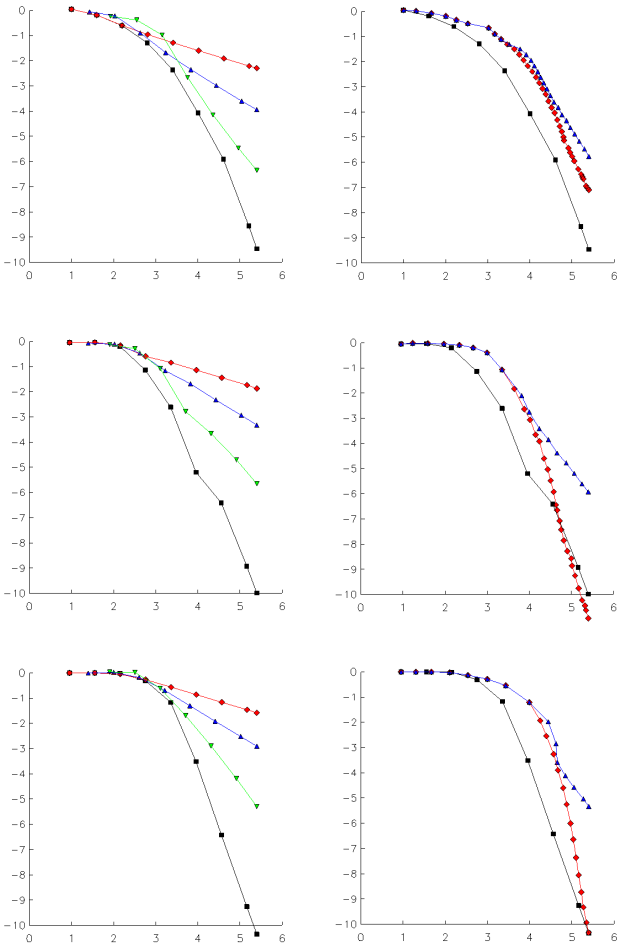

Fig. 5 *hp* refinement. Top row: Singular problem. Middle row: Boundary layer problem. Bottom row: Isotropic problem. Left column: *hp* reference method with interpolation error (black), *h* reference method with interpolation error for $p = 1$ (red), *h* reference method with interpolation error for $p = 2$ (blue), *h* reference method with interpolation error for $p = 4$ (green). Right column: *hp* reference method with interpolation error (black), PLTMG *hp* method with recovered derivatives (red), (1,2,4) PLTMG *hp* method with recovered derivatives (blue).

6 Error Indicator Accuracy

In our final experiment, we illustrate how this workbench can be used to evaluate various a posteriori error estimates. Here we use the reference *h*-adaptive and reference *hp*-adaptive schemes to compute a sequence of meshes in the usual way. On each mesh, we computed an a posteriori error estimate using the interpolant associated with that mesh. Note that the a posteriori error estimate was not used for subsequent adaptive refinement, but was only compared with the exact interpolation error. The results for the three test problems are shown in Figure 7. In these figure the curves representing reference methods are identical to those shown in Figure 5.

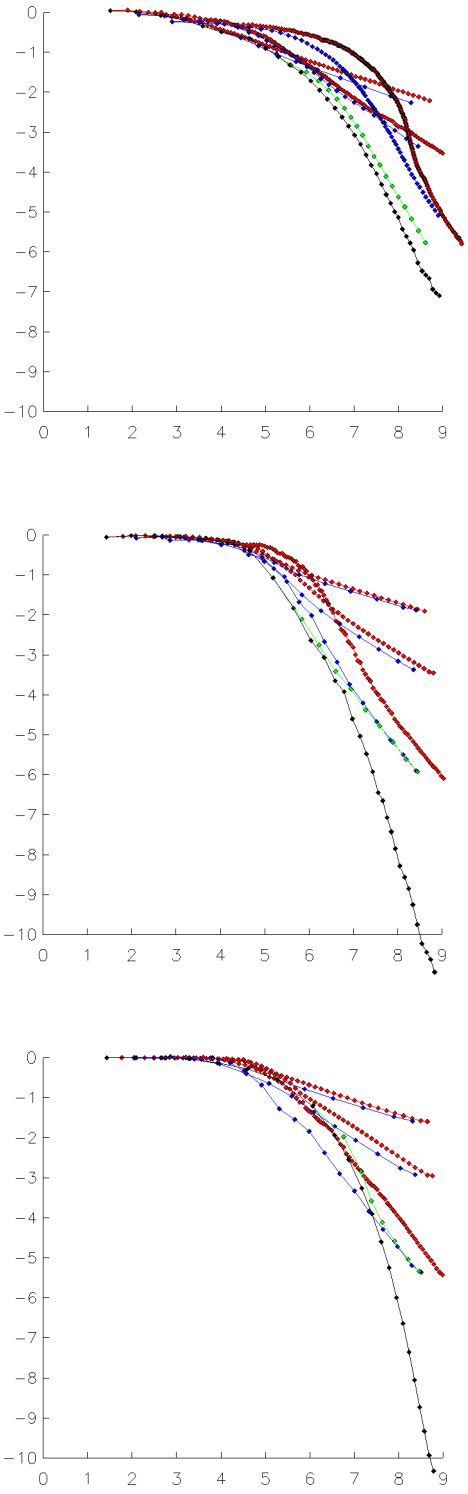

Fig. 6 Refinement Efficiency. Top: Singular problem. Middle: Boundary layer problem. Bottom: Isotropic problem. PLTMG *hp* method with recovered derivatives (black), (1,2,4) PLTMG *hp* method with recovered derivatives (green), PLTMG *h* method with recovered derivatives for $p = 1,2,4$ (blue), marking method with recovered derivatives $\theta = .5$ for $p = 1,2,4$ (red).

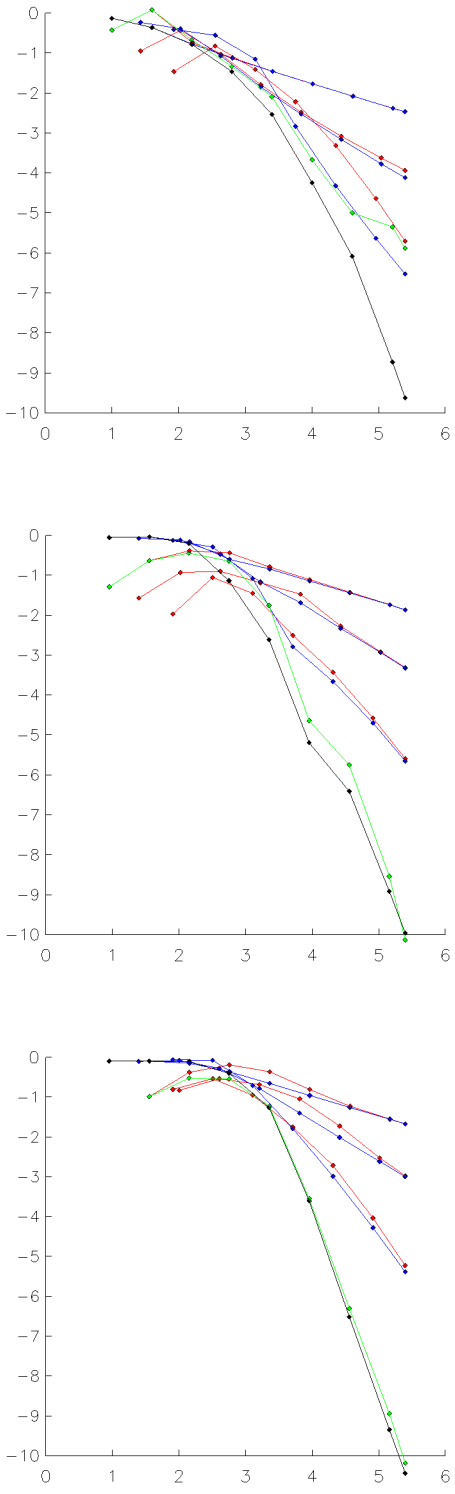

Fig. 7 Error Indicator Accuracy. Top: Singular problem. Middle: Boundary layer problem. Bottom: Isotropic problem. *hp* reference method displaying interpolation error (black), *hp* reference method displaying recovered derivatives (green), *h* reference method with p=1,2,4 displaying interpolation error (blue), *h* reference method with p=1,2,4 displaying recovered derivatives (red).

The a posteriori error estimate studied is recovered derivative estimator used in our other experiments. We first examine results for the *h*-adaptive method for $p = 1, 2, 4$. Asymptotically, these estimates look very good, apparently even asymptotically exact in some cases. On coarse meshes the situation is quite different, where they often under estimate the true error.

We next consider the behavior of the error estimator for the *hp*-adaptive algorithm. In this case, the recovery process proceeds patchwise, with all elements of current degree p plus some fringe elements forming a patch on which derivatives of order p are recovered. Recovered derivatives for the fringe elements are discarded, as the fringe elements are included for the purpose of improving the recovery process for the interior elements. This scheme is largely heuristic, as the analysis in [3] does not directly cover meshes of variable p. For the isotropic problem the recovered derivative estimator track the actual error very closely, except for very coarse grids, similar to the case of *h*-adaptive refinement. This is likely because the polynomial degree is more uniform than in the other examples. In the other examples, the recovered derivatives track less well, but still follow the trend of the exact error.

Overall, this experiment exposes an issue on coarse meshes for this indicator which likely can be addressed with further research. The regularity gap issue illustrated in the singular problem might also be addressed with further research, or may prove to be a fundamental feature of this recovery scheme.

References

1. Bank, R.E.: PLTMG: A software package for solving elliptic partial differential equations, users' guide 12.0. Tech. rep., Department of Mathematics, University of California at San Diego (2016)
2. Bank, R.E., Nguyen, H.: *hp* adaptive finite elements based on derivative recovery and superconvergence. Comput. Vis. Sci. **14**(6), 287–299 (2011). DOI 10.1007/s00791-012-0179-7. URL http://dx.doi.org/10.1007/s00791-012-0179-7
3. 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
4. Bank, R.E., Yserentant, H.: A note on interpolation, best approximation, and the saturation property. Numer. Math. **131**(1), 199–203 (2015). DOI 10.1007/s00211-014-0687-0. URL http://dx.doi.org/10.1007/s00211-014-0687-0
5. 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
6. Demlow, A., Stevenson, R.: Convergence and quasi-optimality of an adaptive finite element method for controlling L_2 errors. Numer. Math. **117**(2), 185–218 (2011). DOI 10.1007/s00211-010-0349-9. URL http://dx.doi.org/10.1007/s00211-010-0349-9

7. Dörfler, W.: A convergent adaptive algorithm for Poisson's equation. SIAM J. Numer. Anal. **33**(3), 1106–1124 (1996). DOI 10.1137/0733054. URL http://dx.doi.org/10.1137/0733054
8. Dupont, T., Scott, R.: Polynomial approximation of functions in Sobolev spaces. Math. Comp. **34**(150), 441–463 (1980). DOI 10.2307/2006095. URL http://dx.doi.org/10.2307/2006095
9. Guo, B., Babuška, I.: The *hp* version of the finite element method. part 1: the basic approximation results. Computational Mechanics **1**(1), 21–41 (1986)
10. Mitchell, W.F.: A collection of 2D elliptic problems for testing adaptive grid refinement algorithms. Appl. Math. Comput. **220**, 350–364 (2013). DOI 10.1016/j.amc.2013.05.068. URL http://dx.doi.org/10.1016/j.amc.2013.05.068
11. Mitchell, W.F., McClain, M.A.: A survey of *hp*-adaptive strategies for elliptic partial differential equations. In: T.E. Simos (ed.) Recent advances in computational and applied mathematics, pp. 227–258. Springer, Dordrecht (2011). DOI 10.1007/978-90-481-9981-5_10. URL http://dx.doi.org/10.1007/978-90-481-9981-5_10
12. Mitchell, W.F., McClain, M.A.: A comparison of *hp*-adaptive strategies for elliptic partial differential equations. ACM Trans. Math. Software **41**(1), Art. 2, 39 (2014). DOI 10.1145/2629459. URL http://dx.doi.org/10.1145/2629459
13. Nochetto, R.H., Siebert, K.G., Veeser, A.: Theory of adaptive finite element methods: an introduction. In: R. DeVore, A. Kunoth (eds.) Multiscale, nonlinear and adaptive approximation, pp. 409–542. Springer, Berlin (2009). DOI 10.1007/978-3-642-03413-8_12. URL http://dx.doi.org/10.1007/978-3-642-03413-8_12
14. Nochetto, R.H., Veeser, A.: Primer of adaptive finite element methods. In: G. Naldi, G. Russo (eds.) Multiscale and adaptivity: modeling, numerics and applications, *Lecture Notes in Math.*, vol. 2040, pp. 125–225. Springer, Heidelberg (2012). DOI 10.1007/978-3-642-24079-9. URL http://dx.doi.org/10.1007/978-3-642-24079-9