Max-Planck-Institut
für Mathematik
in den Naturwissenschaften
Leipzig

Two Dangers to Avoid When Using Gradient Recovery Methods for Finite Element Error Estimation and Adaptivity

by

Jeffrey Ovall

Technical Report no.: 6 2006

Two Dangers to Avoid When Using Gradient Recovery Methods for Finite Element Error Estimation and Adaptivity

Jeffrey S. Ovall
Max Planck Institute for Mathematics in the Sciences
Inselstr. 22-26, D-04103 Leipzig, Germany.
ovall@mis.mpg.de

August 22, 2006

Abstract

Gradient recovery methods for *a posteriori* error estimation in the finite element method are justifiably popular. They are relatively simple to implement, cheap in terms of storage and computational cost, and generally provide efficient and reliable global and local error estimates for adaptive algorithms. In this report we highlight two practical difficulties which can arise when such techniques are used naively, difficulties which can lead to arbitrarily poor performance of the error estimator both in terms of estimating the true error and in effectively guiding the adaptive refinement process.

AMS Subject Classifications: 65N15, 65N30, 65N50
Key Words: finite elements, gradient recovery, a posteriori estimates, adaptivity

1 Introduction

A posteriori error estimation via gradient recovery methods is computationally inexpensive, relatively simple to implement, and generally both efficient and reliable (often asymptotically exact). It is no surprise then that such methods are popular, particularly in the engineering community, and many good papers have been written on the topic over the past fifteen years or so. We refer the interested reader to [4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17, 18] as a reasonable sampling of the literature.

The basic principle behind these techniques is to apply some inexpensive post-processing to the gradient of the computed finite element solution, $\nabla u_h \hookrightarrow \mathcal{R}\nabla u_h$, so that the recovered gradient $\mathcal{R}\nabla u_h$ provides a better estimate of the true gradient ∇u than ∇u_h does. Global and local error estimates $\|\mathcal{R}\nabla u_h - \nabla u_h\|_{0,\Omega}$, $\|\mathcal{R}\nabla u_h - \nabla u_h\|_{0,\tau}$ provide the basis for the adaptive algorithm. Such recovery operators $\mathcal{R}$ can be either local or global in nature. Perhaps the most popular recovery technique is the patchwise discrete least-squares fitting of Zienkiewicz and Zhu [17, 18]. An example of a very good global recovery technique is that of Bank and Xu [4, 5], in which a componentwise L^2-projection of ∇u_h back into the original finite element space is performed, followed by a few iterations of a multigrid-like smoother. Local L^2-projections and weighted averaging techniques are also popular. Xu and Zhang [13] provide a general framework for the analysis of several of these approaches. All gradient recovery techniques can be viewed as performing some sort of componentwise averaging of the gradient ∇u_h.

The purpose of this brief technical report is to highlight two areas of difficulty which can arise when gradient recovery procedures are used without fully appreciating the principles behind them. We do not claim that most practitioners are currently unaware of such potential difficulties. We merely point out why they exist, demonstrate that performance of the estimators can be arbitrarily bad if no care is taken to avoid these difficulties, and discuss some of the methods to avoid them. For concreteness, we only consider continuous, piecewise linear finite elements in bounded domains in $\mathbb{R}^2$. For the numerical experiments we use the software package PLTMG [2], which uses the global gradient recovery technique of Bank and Xu as its standard means of error estimation.

2 Jumps in the Gradient ∇u

We first consider problems for which the gradient of the true solution ∇u is discontinuous at certain interfaces in the domain. This situation arises quite naturally in elliptic problems having a discontinuous coefficient a on the diffusion term, $-\nabla \cdot (a \nabla u) + etc.$ If the finite element solution u_h is a reasonable approximation of u, we expect there to be a similar jump in ∇u_h at such interfaces. Because gradient recovery methods involve some sort of averaging of the piecewise constant gradient ∇u_h to obtain the recovered gradient $\mathcal{R}\nabla u_h$, if no care is taken to avoid averaging across such interfaces, the resulting local error estimates $\|(\mathcal{R}-I)\nabla u_h\|_{0,\tau}$ near interfaces will almost certainly over-estimate the actual error $\|\nabla(u-u_h)\|_{0,\tau}$ there. Briefly, if gradient recovery is used naively in situations such as this, it is likely that large local errors will be indicated where the errors are actually small, leading to (at best) suboptimal performance of the adaptive algorithm. We demonstrate explicitly in the following example that the performance can be arbitrarily poor.

2.1 A Model Problem for Jumping Gradients

Let Ω denote the open upper half unit disk - see Figure 1 for a labelling of the two subdomains, boundary and interface referred to below. We consider the following model problem:

$$-a\Delta u = 0 \text{ in } \Omega \qquad u = 0 \text{ on } \Gamma_1 \qquad \nabla u \cdot \mathbf{n} = 0 \text{ on } \Gamma_2 \tag{1}$$

$$u = b_1 \sin \alpha\theta + c_1 \cos \alpha\theta \text{ on } \Gamma_3 \qquad u = b_2 \sin \alpha\theta + c_2 \cos \alpha\theta \text{ on } \Gamma_4 \tag{2}$$

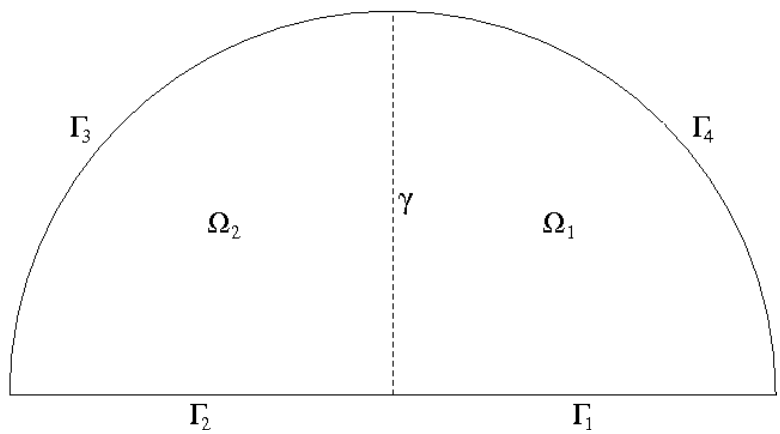

Figure 1: The domain Ω, with labelled subdomains, boundary and interface.

where $a = \beta > 0$ in Ω_1, $a = 1$ in Ω_2, and u and $a\nabla u \cdot \mathbf{n}$ are continuous on the interface γ between Ω_1 and Ω_2. The continuity of $a\nabla u \cdot \mathbf{n}$ on γ is required so that, in the conversion to weak form, the integrals along γ cancel.

We seek a solution of the form

$$u = b_k \sin \alpha\theta + c_k \cos \alpha\theta,\ k = 1,\, 2. \tag{3}$$

Clearly, $u = 0$ satisfies conditions (1-2) trivially with $b_1 = b_2 = c_1 = c_2 = 0$. If we want a nontrivial solution u, we must choose α so that the linear system

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \cos \alpha\pi & -\sin \alpha\pi \\ \sin \frac{\alpha\pi}{2} & \cos \frac{\alpha\pi}{2} & -\sin \frac{\alpha\pi}{2} & -\cos \frac{\alpha\pi}{2} \\ \beta \cos \frac{\alpha\pi}{2} & -\beta \sin \frac{\alpha\pi}{2} & \cos \frac{\alpha\pi}{2} & \sin \frac{\alpha\pi}{2} \end{pmatrix} \begin{pmatrix} b_1 \\ c_1 \\ b_2 \\ c_2 \end{pmatrix} = \mathbf{0} \tag{4}$$

has a nontrivial solution. The first two rows of this system correspond to the boundary conditions on Γ_1 and Γ_2, and the last two correspond to the continuity conditions on γ. By checking the determinant, it is clear that choosing

$$\alpha \in \pm \frac{\arccos \frac{1-\beta}{1+\beta}}{\pi} + 2\mathbb{Z} \tag{5}$$

gives us what we need. We will consider the family of solutions u of the form

$$u = \begin{cases} r^\alpha \sin \alpha\theta & , \quad 0 \le \theta \le \frac{\pi}{2} \\ r^\alpha \left(\frac{2\beta}{\beta+1} \sin \alpha\theta - \frac{\sqrt{\beta}(\beta-1)}{\beta+1} \cos \alpha\theta \right) & , \quad \frac{\pi}{2} < \theta \le \pi \end{cases} \tag{6}$$

$$\alpha \in \frac{\arccos \frac{1-\beta}{1+\beta}}{\pi} + 2\mathbb{Z}. \tag{7}$$

The gradient near the interface γ is given by

$$\nabla u\left(r, \frac{\pi}{2}^-\right) = \alpha r^{\alpha-1}\begin{pmatrix}\frac{-1}{\sqrt{\beta+1}}\\ \\ \sqrt{\frac{\beta}{\beta+1}}\end{pmatrix} \quad , \quad \nabla u\left(r, \frac{\pi}{2}^+\right) = \alpha r^{\alpha-1}\begin{pmatrix}\frac{-\beta}{\sqrt{\beta+1}}\\ \\ \sqrt{\frac{\beta}{\beta+1}}\end{pmatrix}. \tag{8}$$

It is clear from (8) that ∇u has a jump of magnitude $\sqrt{\beta+1}$ across γ. We should expect a similar jump in ∇u_h for any finite element solution u_h which is a moderately good approximation of u. We therefore expect that averaging across the interface γ to recover the gradient will lead to gross over-estimation of error near γ and adaptive refinement which concentrates heavily near there. It is also clear from this example that simply recovering the weighted gradient $\mathcal{R}a\nabla u_h$ for use in error estimates of the form $\|\mathcal{R}a\nabla u_h - a\nabla u_h\|_{0,\tau}$ or $\|a^{-1}\mathcal{R}a\nabla u_h - \nabla u_h\|_{0,\tau}$ will not fix this problem - this merely shifts the difficulty from the x-component of the gradient to the y-component. The most obvious solution is to use the recovery technique of choice in both of the subdomains separately. Bank and Xu note this fix in an example in [5], and many others probably use it as well whether or not they explicitly say so.

To illustrate just how necessary it can be to treat problems gradient jumps appropriately, we consider our model problem with the choice $\beta = 10^6$. To five significant digits, this gives us

$$\alpha = 0.99936 \quad , \quad b_2 = \frac{2\beta}{\beta+1} = 2.0000 \quad , \quad c_2 = -\frac{\sqrt{\beta}(\beta-1)}{\beta+1} = -1000.0\,. \tag{9}$$

In Figure 2 and Table 1, we contrast the behavior of the Bank-Xu global recovery operator $\mathcal{R}$ and its variant $\hat{\mathcal{R}}$ in which the recovery technique is applied to both subdomains separately. The difference is striking. We see in Figure 2 that the $\mathcal{R}$ gradient recovery refinement is concentrated heavily around the interface γ, as predicted, while refinement based on $\hat{\mathcal{R}}$ error estimates seems to correspond more closely to where the solution itself exhibits more interesting behavior. But the real evidence of which technique does a better job lies the quantitative data.

In Table 1 we provide the number of triangles N, the global gradient error estimates $\|(\mathcal{R} - I)\nabla u_h\|_{0,\Omega}$ and $\|(\hat{\mathcal{R}} - I)\nabla u_h\|_{0,\Omega}$, the exact global gradient errors $|e_h|_{1,\Omega}$, and the effectivity EFF of the estimators. Standard scientific notation is abbreviated in the table by giving the base ten exponent as a subscript, e.g., $5.6166_{-2} \equiv 5.6166 \times 10^{-2}$. It stands out that, in the early stages of its adaptive refinement, $\|(\mathcal{R} - I)\nabla u_h\|_{0,\Omega}$ over-estimates the actual gradient error on the mesh by a factor of more than $1000 = \sqrt{\beta}$. Although the effectivity seems to be improving as the mesh is refined, a more careful inspection of the numbers reveals that this is due to the fact that the true error is no longer being reduced by the refinement! By contrast, we see that the performance of the $\hat{\mathcal{R}}$ error estimator is near optimal, with effectivity near 1 and roughly linear convergence in error.

2.2 A More Realistic Example

In this example, we consider orthotropic heat conduction in a thermal battery. The data for this problem originally come from Leszek Demkowicz. The problem is as

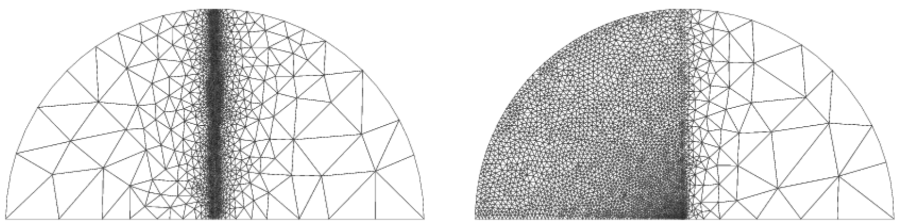

Figure 2: The meshes for the jumping coefficient problem after three stages of adaptive refinement, using $\mathcal{R}$ gradient recovery error estimates (left) and $\hat{\mathcal{R}}$ gradient recovery error estimates (right). The mesh on the left has 6390 triangles, and the mesh on the right has 6287 triangles.

Table 1: Estimates, exact values and effectivity in the H^1-seminorm for the jumping coefficient problem - $\mathcal{R}$ gradient recovery (top) and $\hat{\mathcal{R}}$ gradient recovery (bottom).

N	66	355	1554	6390	25779	103373
$\\|(\mathcal{R}-I)\nabla u_h\\|_{0,\Omega}$	264.12	136.95	72.115	37.474	19.197	9.7403
$\|e_h\|_{1,\Omega}$	0.14956	7.4456_{-2}	6.1226_{-2}	5.9943_{-2}	5.9560_{-2}	6.0322_{-2}
EFF	1766.0	1839.3	1177.8	625.16	322.31	161.47
N	66	347	1526	6287	25549	102992
$\\|(\hat{\mathcal{R}}-I)\nabla u_h\\|_{0,\Omega}$	0.20885	5.7936_{-2}	2.1005_{-2}	9.3772_{-3}	4.3581_{-3}	2.1143_{-3}
$\|e_h\|_{1,\Omega}$	0.14956	4.5009_{-2}	1.8622_{-2}	8.9091_{-3}	4.2268_{-3}	2.0657_{-3}
EFF	1.3964	1.2872	1.1279	1.0525	1.0311	1.0235

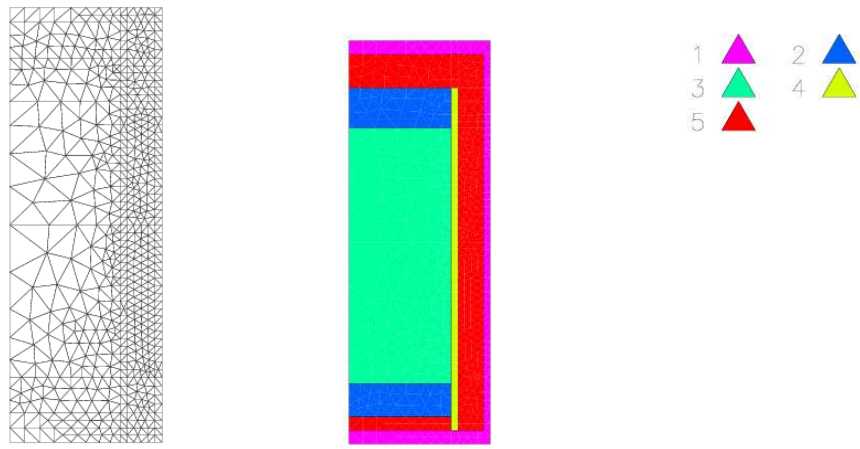

Figure 3: The domain (right) with color-coded regions, and the initial triangulation for the Thermal Battery Problem.

follows:

$$\begin{array}{c} -a_1 u_{xx} - a_2 u_{yy} = f \text{ in } \Omega \\ a\nabla u \cdot \mathbf{n} = c - \alpha u \text{ on } \partial\Omega \end{array}$$

region	a_1	a_2	f
1	25	25	0
2	7	0.8	1
3	5	10^{-4}	1
4	0.2	0.2	0
5	0.05	0.05	0

side	c	α
left	0	0
top	1	3
right	2	2
bottom	3	1

.

The domain, broken into its five regions, together with the initial triangulation computed using an automatic mesh generator based on a skeleton of the domain are given in Figure 3. A qualitatively accurate approximate solution on a very fine mesh is given in Figure 4 .

Again we consider the performance of the Bank/Xu global recovery estimator. Based on the qualitative features of the approximate solution shown in Figure 4, we might expect a well-adapted mesh to be fine near the interior corners of the five subregions, but relatively coarse in regions where the solution is roughly linear - regions 1 and 2, for example, as well as portions of the other regions. In contrast, we expect the recovery estimates to lead to refinement which is concentrated near the interfaces between the subregions. This is precisely what we see in Figure 5. Also in Figure 5 we see the refinement produced using the hierarchical basis error estimator described in [10]. Such estimators are known to be reliable under quite general conditions [1, 3] and have been shown to be asymptotically exact [10] in situations in which gradient recovery methods are asymptotically exact. We will discuss below why we chose to use this estimator here instead of just doing gradient

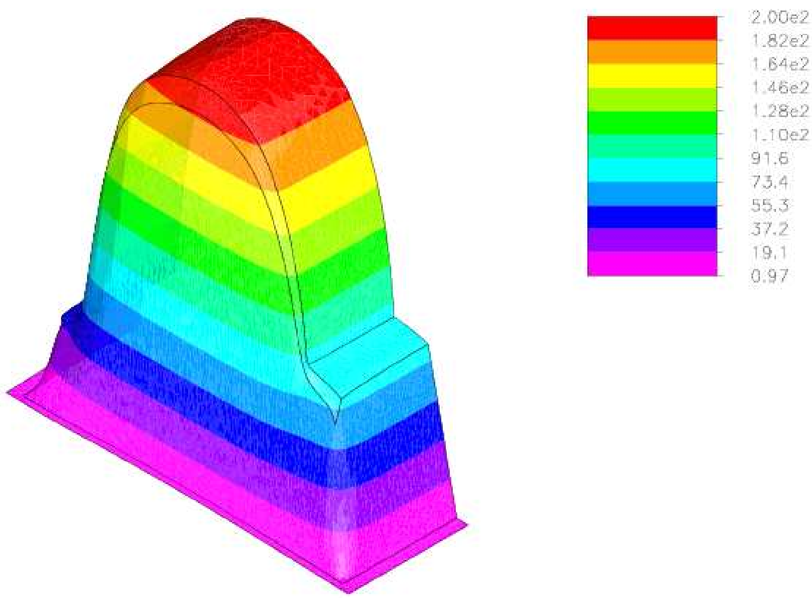

Figure 4: A qualitatively accurate approximate solution for the Thermal Battery Problem.

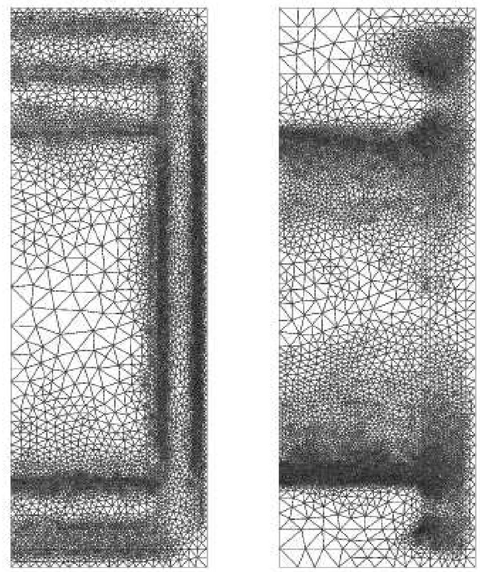

Figure 5: The triangulation after a couple of stages of adaptive refinement using global gradient recovery (left) and hierarchical basis error estimation for the Thermal Battery Problem.

recovery in each subdomain separately, and merely point out that the refinement pattern produced appears to be well-suited for resolving the features of the solution.

We certainly could have used the Bank/Xu recovery method on each subdomain separately, but the necessary modifications to the code to perform the recovery in that way and to accommodate the multiple values of the recovered gradient $\mathcal{R}\nabla u_h$ on the interfaces becomes more cumbersome as the number of subdomains increases. This is primarily why we chose to use a hierarchical basis error estimator for the comparison - it needs no modification for problems with jumping coefficients. Any code which can be used for problems having jumping coefficients will also have some simple way of determining which triangles are in which regions, so using a recovery method separately in each of these regions is not too difficult in principle - some recovery methods may be more difficult to modify than others.

Before moving on, we want to point out another feature of this sort of problem which the aforementioned fix might not adequately address. Recovery methods generally require at least a patch of triangles around each point or triangle to recover the gradient at the point or on the triangle - boundary points or triangles are treated in a “one-sided” fashion. We have seen above that it is necessary to make sure that any such patch does not contain an interface across which the diffusion coefficient jumps. What if some subregions are only one triangle wide? Even after any necessary modifications to the recovery method are made to handle this situation, it still might not be possible to accurately determine the local error, although the local estimates could still be adequate for determining if further refinement is necessary or not. We note that the initial triangulation for the Thermal Battery Problem is precisely of this sort. Recall that this initial triangulation was produced by an automatic mesh generator from data describing the five subregions. Apart from the restriction that no triangle should overlap two subregions, shape-regularity of the triangulation was the key governing principle. We see in Figure 3 that regions 1 and 4 are largely only one triangle in width, and in Figure 5 we see that this is actually quite sufficient for adequate resolution of the solution there.

3 Not Enough Interior Vertices

The discussion at the end of the previous section, concerning recovery over few elements, is closely related to the topic of this section. We saw in the previous section a situation in which the triangulation was sufficiently fine in some subdomains for resolution of the solution (one triangle in width), but perhaps not fine enough to adequately recover the gradient there. We now look at an example where it truly is necessary to refine the triangulation, both for resolution of the solution and for gradient recovery, but where gradient-recovery-based error estimates do not lead to reasonable refinement. The fault in this problem really lies with the initial mesh and not with the gradient recovery technique.

The Spiral Domain problem is to find u such that:

$$-\Delta u = 1 \text{ in } \Omega \quad , \quad u = 0 \text{ on } \partial\Omega, \tag{10}$$

where Ω is a narrow spiral-shaped domain. In Figure 6 we see the initial triangulation of the domain, which was generated by an automatic mesh generator from data describing the boundary. We note that most triangles have all three of their vertices on the boundary. Of course, this means that the finite element solution on this triangulation will be identically zero for much of the domain. We see this in Figure 6, together with a qualitatively accurate finite element solution on a much finer mesh (uniformly refined).

It is clear that the largest error in the approximation on the initial mesh is where the finite element solution is zero - where there were no interior vertices in the triangulation. The problem is that gradient recovery methods have no choice but to estimate zero error there. So in the regions where the error is largest, such methods will estimate small (or no) error, and they will estimate relatively large error where the error is actually the smallest. This leads to adaptive refinement such as that depicted in Figure 7, and very little improvement in the quality of the finite element solution on the refined mesh. We contrast this with the refinement produced by hierarchical basis error estimates, which yield a much more sensible pattern of refinement and a marked improvement in the next finite element solution. Actually, any error estimation technique which somehow takes into account the residual of the finite element solution would yield similar refinement to the hierarchical basis approach.

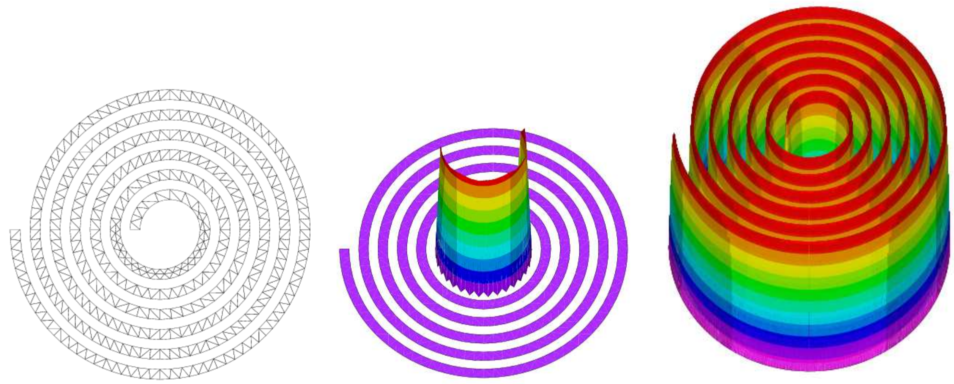

Figure 6: The initial triangulation (left), initial finite element solution (center), and an accurate finite element solution on a fine mesh for the Spiral Domain Problem.

As was stated earlier, the fault really lies with the automatic mesh generator. It should not have produced a mesh in which so many triangles had all of their vertices on the boundary. This defect is actually quite easy to correct - simply mark all triangles which have all of their vertices on the boundary, and either refine or perform edge flips on pairs of adjacent triangles until all of them have at least one interior vertex. This sort of procedure could also be extended to produce interior vertices in specified subdomains for problems such as the Thermal Battery Problem. Some mesh generators may already have this sort of feature built in, but some do not, and one should make sure before using gradient recovery in conjunction with them.

4 Conclusions

We have pointed out two potential dangers to avoid when using gradient recovery methods for error estimation and adaptive refinement. One of them is easily dealt with by ensuring that each triangle in the computational mesh has at least one interior vertex - where we can interpret interior in the sense of internal interfaces as well as the domain boundary. The first of the two dangers is more serious, and can only be addressed by modifying the recovery method directly to handle jumps in the gradient ∇u. In principle the necessary modification is not too difficult, because gradient jumps are most likely to correspond directly to discontinuities in the differential operator and can therefore be predicted *a priori*. In practice, the amount of necessary recoding could vary quite a bit between the recovery methods. Though many engineers are no doubt aware of this danger and have corrected it in their codes, we have found that it has not been sufficiently advertised in the literature for less experienced users, particularly in light of just how bad the performance can be if insufficient care is taken.

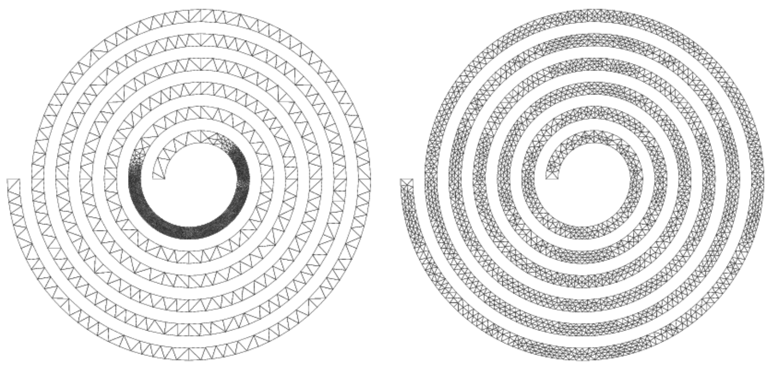

Figure 7: The adaptively refined meshes after one stage of refinement using gradient recovery error estimation (left) and hierarchical basis error estimation for the Spiral Domain Problem.

References

[1] R. E. Bank. Hierarchical bases and the finite element method. In *Acta numerica, 1996*, volume 5 of *Acta Numer.*, pages 1–43. Cambridge Univ. Press, Cambridge, 1996.

[2] R. E. Bank. Pltmg: A software package for solving elliptic partial differential equations, users' guide 9.0. Technical report, University of California, San Diego, 2004.

[3] R. E. Bank and R. K. Smith. A posteriori error estimates based on hierarchical bases. *SIAM J. Numer. Anal.*, 30(4):921–935, 1993.

[4] R. E. Bank and J. Xu. Asymptotically exact a posteriori error estimators. I. Grids with superconvergence. *SIAM J. Numer. Anal.*, 41(6):2294–2312 (electronic), 2003.

[5] R. E. Bank and J. Xu. Asymptotically exact a posteriori error estimators. II. General unstructured grids. *SIAM J. Numer. Anal.*, 41(6):2313–2332 (electronic), 2003.

[6] L. Du and N. Yan. Gradient recovery type a posteriori error estimate for finite element approximation on non-uniform meshes. *Adv. Comput. Math.*, 14(2):175–193, 2001.

[7] R. Durán, M. A. Muschietti, and R. Rodríguez. On the asymptotic exactness of error estimators for linear triangular finite elements. *Numer. Math.*, 59:107–127, 1991.

[8] X. D. Li and N.-E. Wiberg. A posteriori error estimate by element patch post-processing, adaptive analysis in energy and L_2 norms. *Comput. & Structures*, 53(4):907–919, 1994.

[9] A. Naga and Z. Zhang. A posteriori error estimates based on the polynomial preserving recovery. *SIAM J. Numer. Anal.*, 42(4):1780–1800 (electronic), 2004.

[10] J. S. Ovall. Function, gradient and hessian recovery using quadratic edge-bump functions. *Submitted to SINUM*, 2006.

[11] A. H. Schatz and L. B. Wahlbin. Asymptotically exact a posteriori estimators for the pointwise gradient error on each element in irregular meshes. II. The piecewise linear case. *Math. Comp.*, 73(246):517–523 (electronic), 2004.

[12] N.-E. Wiberg and X. D. Li. Superconvergent patch recovery of finite-element solution and a posteriori L_2 norm error estimate. *Comm. Numer. Methods Engrg.*, 10(4):313–320, 1994.

[13] J. Xu and Z. Zhang. Analysis of recovery type a posteriori error estimators for mildly structured grids. *Math. Comp.*, 73(247):1139–1152 (electronic), 2004.

[14] N. Yan and A. Zhou. Gradient recovery type a posteriori error estimates for finite element approximations on irregular meshes. *Comput. Methods Appl. Mech. Engrg.*, 190(32-33):4289–4299, 2001.

[15] Z. Zhang and A. Naga. A new finite element gradient recovery method: superconvergence property. *SIAM J. Sci. Comput.*, 26(4):1192–1213 (electronic), 2005.

[16] J. Z. Zhu and O. C. Zienkiewicz. Superconvergence recovery technique and a posteriori error estimators. *Internat. J. Numer. Methods Engrg.*, 30(7):1321–1339, 1990.

[17] O. C. Zienkiewicz and J. Z. Zhu. The superconvergent patch recovery and a posteriori error estimates. I. The recovery technique. *Internat. J. Numer. Methods Engrg.*, 33(7):1331–1364, 1992.

[18] O. C. Zienkiewicz and J. Z. Zhu. The superconvergent patch recovery and a posteriori error estimates. II. Error estimates and adaptivity. *Internat. J. Numer. Methods Engrg.*, 33(7):1365–1382, 1992.