Max-Planck-Institut
für Mathematik
in den Naturwissenschaften
Leipzig

Fixing a "Bug" in Recovery-Type A Posteriori Error Estimators

by

Jeffrey Ovall

Preprint no.: 25 2006

Fixing a "Bug" in Recovery-Type *A Posteriori* Error Estimators *

Jeffrey S. Ovall†

February 28, 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 paper we highlight a difficulty that such error estimators have in the case of problems with a jumping coefficient on the diffusion term of the differential operator, explain why this difficulty exists and offer a fairly straight-forward way to fix it.

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

1 Introduction

A posteriori error estimation via gradient recovery methods is cheap in terms of computational cost, 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 [3, 4, 5, 6, 8, 9, 10, 11, 12] 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 piecewise linear 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 patch-wise discrete least-squares fitting of Zienkiewicz and Zhu [11, 12]. An example of a very good global recovery technique is that of Bank and Xu [3, 4],

*Submitted to SINUM, Feb. 8, 2006

†Max Planck Institute for Mathematics in the Sciences, D-04103 Leipzig, Germany (ovall@mis.mpg.de).

in which a componentwise L^2-projection of ∇u_h into the space of continuous piecewise linear functions on the given mesh 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 [8] provide a general framework for the analysis of several of these approaches. All gradient recovery techniques can be viewed as performing some sort of averaging of the piecewise constant gradient ∇u_h. The purpose of this brief paper is to highlight a difficulty which arises when gradient recovery is used for elliptic problems with a jumping coefficient a on the diffusion term, $-\nabla \cdot (a\nabla u) + \cdots$, and no care is taken to avoid averaging across the interfaces on which a has a jump discontinuity. Although this difficulty has been noted elsewhere before (see [4], for example), here we demonstrate explicity that the effectivity such estimators can be arbitrarily bad and that the local error estimators can lead to adaptive refinement which completely fails to reduce the global error. We also explain why global weighted-gradient recovery schemes, $\nabla u_h \hookrightarrow a^{-1}\mathcal{R}a\nabla u_h$, will not work; so we are really stuck with procedures which treat each subdomain separately.

We used quotations around "bug" in the title, because problems with jumping coefficients lead to solutions which do not meet the global regularity assumptions specified in the theoretical justification of gradient recovery methods. In fact, $u \notin H^2(\Omega)$, although u might be smooth almost everywhere. However, gradient recovery error estimation and adaptive refinement has been empirically shown to be amazingly effective in situations in which the current theory does **not** apply - its practical utility is actually **based** on its ability to accurately identify and resolve locally singular behavior. In light of this, we consider poor performance of the estimator, explained and demonstrated by example in Sections 2 and 3 to be a "bug" for which we seek a simple fix.

2 The Model Problem

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

$$-a\Delta u = 0 \text{ in } \Omega \tag{1}$$

$$u = 0 \text{ on } \Gamma_1 \tag{2}$$

$$u = b_1 \sin \alpha\theta + c_1 \cos \alpha\theta \text{ on } \Gamma_2 \tag{3}$$

$$u = b_2 \sin \alpha\theta + c_2 \cos \alpha\theta \text{ on } \Gamma_3 \tag{4}$$

$$\nabla u \cdot \mathbf{n} = 0 \text{ on } \Gamma_4 \tag{5}$$

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.

Clearly, $u = 0$ satisfies conditions (1)-(5) trivially. If we want a nontrivial solution

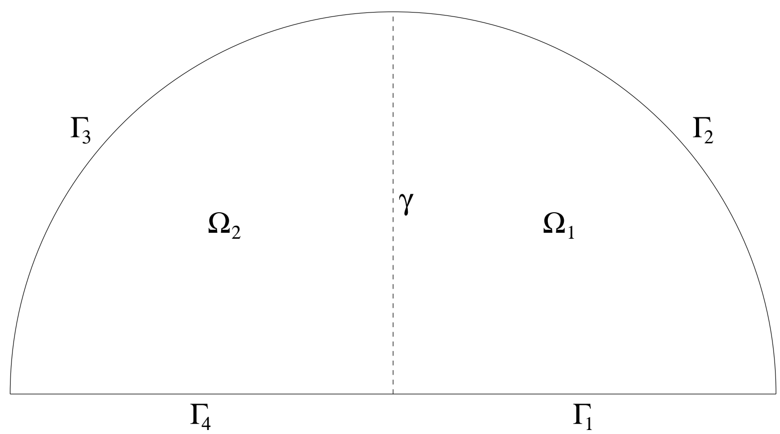

Figure 1: The domain Ω, with labeled subdomains, boundary and interface.

u, we must choose α so that the linear system

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \cos\alpha\pi & -\sin\alpha\pi \\ \sin\frac{\alpha\pi}{2} & \cos\frac{\alpha\pi}{2} & -\sin\frac{\alpha\pi}{2} & -\cos\frac{\alpha\pi}{2} \\ \beta\cos\frac{\alpha\pi}{2} & -\beta\sin\frac{\alpha\pi}{2} & \cos\frac{\alpha\pi}{2} & \sin\frac{\alpha\pi}{2} \end{pmatrix} \begin{pmatrix} b_1 \\ c_1 \\ b_2 \\ c_2 \end{pmatrix} = \mathbf{0} \tag{6}$$

has a nontrivial solution. The first two rows of this system correspond to the boundary conditions on Γ_1 and Γ_4, and the last two correspond to the continuity conditions on γ. By checking the determinant, it is clear that choosing

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

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

$$u = \begin{cases} r^\alpha \sin\alpha\theta & , \quad 0 \le \theta \le \frac{\pi}{2} \\ r^\alpha\left(\frac{2\beta}{\beta+1}\sin\alpha\theta - \frac{\sqrt{\beta}(\beta-1)}{\beta+1}\cos\alpha\theta\right) & , \quad \frac{\pi}{2} < \theta \le \pi \end{cases} \tag{8}$$

$$\alpha \in \frac{\arccos\frac{1-\beta}{1+\beta}}{\pi} + 2\mathbb{Z}. \tag{9}$$

The gradient near the interface γ is given by

$$\nabla u\left(r, \frac{\pi}{2}^-\right) = \alpha r^{\alpha-1}\begin{pmatrix} \frac{-1}{\sqrt{\beta+1}} \\ \\ \sqrt{\frac{\beta}{\beta+1}} \end{pmatrix} \quad , \quad \nabla u\left(r, \frac{\pi}{2}^+\right) = \alpha r^{\alpha-1}\begin{pmatrix} \frac{-\beta}{\sqrt{\beta+1}} \\ \\ \sqrt{\frac{\beta}{\beta+1}} \end{pmatrix}. \tag{10}$$

It is clear from (10) 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. Herein lies the problem:

> If $\|\nabla u - \nabla u_h\|_{0,\tau}$ is small on an element τ near γ, and $\mathcal{R}$ is a gradient recovery operator which averages across γ, then the error estimate $\|\mathcal{R}\nabla u_h - \nabla u_h\|_{0,\tau}$ can be quite large in comparison.

Refinement based on such local estimates is expected to concentrate heavily on the region near γ, even if this is not an optimal strategy. In fact, if $\|\nabla u - \nabla u_h\|_{\infty,\tau} \sim h$ for a triangle τ which has an edge touching γ, we expect $\|\nabla u - \nabla u_h\|_{0,\tau} \sim h^2$ and $\|\mathcal{R}\nabla u_h - \nabla u_h\|_{0,\tau} \sim \sqrt{\beta}\, h$! 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 comments above directly refer to our model problem, but this general difficulty is present for **any** problem with a jumping diffusion coefficient a. Because $a\nabla u \cdot \mathbf{n}$ must be continuous along the interfaces between regions where a jumps, both ∇u and $a\nabla u$ will almost certainly jump across these interfaces. Therefore, if the recovery operator $\mathcal{R}$ averages across these interfaces, the corresponding local error estimates may be quite large in comparison to the actual local errors. The implication is obvious:

> To improve the reliability and efficiency of gradient recovery error estimators for problems with jumping diffusion coefficients, it is **necessary** to avoid averaging across the interfaces between regions with different coefficients.

We do not argue here that avoiding averaging across the interfaces is also **sufficient** for improving the performance of the estimator, although it appears to be the only reasonable approach. The lack of even H^2 global regularity of the solution u does not bode well for a general proof of this sort. However, patchwise recovery methods have proven effective in practice on a wide variety of problems, and we are merely adding the additional requirement that none of the patches overlap any interface between regions where the diffusion coefficient a jumps. In Section 3 below we numerically demonstrate on the model problem both the difficulty described above when averaging takes place across the interface, and the near optimal performance that results from avoiding such averaging.

3 Numerically Demonstrating the Bug and its Fix

We consider our model problem with the choice $\beta = 10^6$. To five significant digits, this gives us

$$\alpha = 0.99936 \quad , \quad b_2 = \frac{2\beta}{\beta+1} = 2.0000 \quad , \quad c_2 = -\frac{\sqrt{\beta}(\beta-1)}{\beta+1} = -1000.0\ . \tag{11}$$

We see from (8) then, that the solution u varies much more dramatically in Ω_2 than in Ω_1. Intuitively, we should expect a well-adapted mesh to be relatively fine in Ω_2 and coarse in Ω_1. We compare the performance of two gradient recovery operators. The recovery operator here labeled $\mathcal{R}$, due to Bank and Xu [3, 4], uses a componentwise

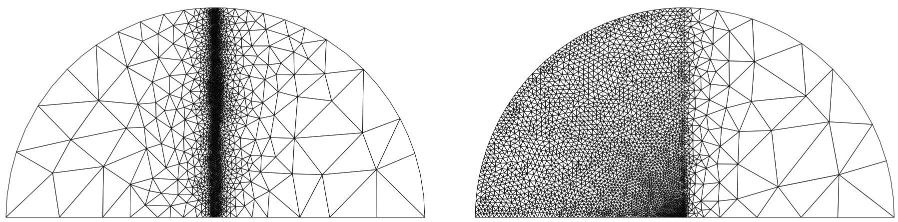

Figure 2: The meshes for the jumping coefficient problem after three stages of adaptive refinement, using $\mathcal{R}$ gradient recovery error estimates (left) and $\hat{\mathcal{R}}$ gradient recovery error estimates (right). The mesh on the left has 6390 triangles, and the mesh on the right has 6287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

global L^2-projection of ∇u_h into the space of continuous, piecewise linear functions on the mesh followed by a few iterations of a multigrid-like smoother. It is global in nature, so it will average across γ. The recovery operator here labeled $\hat{\mathcal{R}}$ is essentially the same as the first, but the projection and smoothing are carried out independently on the two subdomains Ω_1 and Ω_2.

In Figure 2 we see a comparison of meshes with roughly the same number of elements, which were generated by adaptive refinement based on both types gradient recovery error estimates. The difference is striking. The $\mathcal{R}$ gradient recovery refinement is concentrated heavily around the interface γ, as predicted in Section 2. Refinement based on $\hat{\mathcal{R}}$ error estimates seems to correspond more closely to what was predicted at the beginning of this section, but we must look at numbers before making any real judgment.

In Table 1 we provide the number of elements 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.

4 Conclusions

We have seen that gradient recovery operators which average across interfaces between regions where the diffusion coefficient a jumps can perform arbitrarily poorly, and that variants which seek to recover the weighted gradient $a\nabla u_h$ will also tend to

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

perform poorly. It appears necessary that we avoid averaging across such interfaces - which rules out global recovery schemes. We demonstrated here by example that recovery schemes that independently treat subdomains between which a has a jump discontinuity have a reasonable hope of achieving near optimal performance.

A difficulty may arise in practice, and could be difficult to address in this way. Suppose the domain is decomposed into subdomains in which the diffusion coefficient a is continuous - with jump discontinuities between these subdomains. If one or more of these subdomains is small in area or narrow, there may not be any reasonable gradient recovery which can take place - imagine a narrow strip with very few or no interior triangles. In such cases it may be necessary to either force the initial triangulation to be sufficiently fine in each subdomain, or to forego the use of gradient recovery error estimators in favor of hierarchical basis error estimators such as those described in [1, 2, 7] which tend to be quite robust and have been shown (empirically) to work well on problems with jumping coefficients.

The author thanks Steffen Börm of the Max Planck Institute in Leipzig, Markus Melenk of the Technische Universität Wien, and Randy Bank of the University of California at San Diego for conversations and comments which helped to shape the direction of this paper.

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

[3] 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.

[4] 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.

[5] 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.

[6] 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.

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

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.