Optimal Hessian recovery using a biorthogonal system with an application to adaptive refinement

Jordan Shaw-Carmody[1] Bishnu P. Lamichhane[2]

(Received 31 January 2023; revised 12 September 2023)

Abstract

We present a method for recovering the Hessian from a linear finite element approach to achieve a higher rate of convergence. This method uses an L^2-based projection as well as boundary modification to achieve and improve the convergence rate. The projection uses a biorthogonal system to make the computation more numerically efficient. We present numerical examples to illustrate the efficiency and optimality of our approach on different meshes. The performance of our approach on adaptively refined meshes is briefly explored.

DOI:10.21914/anziamj.v64.17971, © Austral. Mathematical Soc. 2024. Published 2024-03-17, as part of the Proceedings of the 20th Biennial Computational Techniques and Applications Conference. ISSN 1445-8810. (Print two pages per sheet of paper.) Copies of this article must not be made otherwise available on the internet; instead link directly to the DOI for this article.

Contents

1 Introduction

The Hessian matrix is important in the adaptive finite element method since it is used to inform and direct a refinement of an adaptive mesh. We propose a computationally effective Hessian recovery method by using biorthogonal projections with boundary modifications to recover the gradient [8, 11], and then applying the biorthogonal projection with a new boundary modification to recover the Hessian. Two direct applications of the gradient recovery lead to a rate of convergence of $O\left(h^{1.5}\right)$ for the recovered Hessian, where h is the mesh size. We therefore propose a simple modification of the approach, which leads to an improved rate of convergence of $O\left(h^{2}\right)$. The higher convergence rate is beneficial because it means that the number of refinements to reach a more accurate Hessian is reduced.

Second order derivatives such as the Hessian are recovered by post-processing techniques. Post-processing is important in applied and computational mathematics because it allows further information to be obtained from what is

already known. This has been demonstrated in applications of the finite element method by techniques developed to perform gradient recovery, such as those that use local or global least square fittings [7, 12, 13, 14], global or local projections [2, 6], or averaging methods [3, 10].

The numerical results obtained in Section 5 agree with the rates obtained in the literature [5] for the uniformly refined meshes. Our method of projection is an efficient computation, since the only matrix that needs to be inverted is a diagonal matrix, as described in Section 2.1.

Let our domain $\Omega \subset \mathbb{R}^2$ be bounded and have polygonal boundary $\partial\Omega$. Let $\mathcal{T}_h$ be a quasi-uniform partition of Ω into triangles. We define $\mathcal{N}$ to be the set of all N nodes of the partition $\mathcal{T}_h$. We use standard Sobolev spaces

$$L^2(\Omega) = \left\{ f : \int_\Omega |f(x)|^2 \, dx < \infty \right\},$$
$$H^1(\Omega) = \left\{ f : f \in L^2(\Omega), \nabla f \in \left[L^2(\Omega) \right]^2 \right\}.$$

We denote the conforming standard element space as

$$V_h = \left\{ v \in C^0(\Omega) : v|_T \in \mathcal{P}_1(T), T \in \mathcal{T}_h \right\},$$

with $\{\phi_1, \ldots, \phi_N\}$ being a set of basis functions of V_h. For this article, we use u_h to denote a finite element solution.

2 Gradient recovery method

2.1 Biorthogonal projection

Our choice of gradient recovery method employs a biorthogonal projection operator Π_h to project ∇u_h onto $[V_h]^2$. The projection is performed by finding $g_1 = \Pi_h \left(\frac{\partial u_h}{\partial x_1} \right) \in V_h$ and $g_2 = \Pi_h \left(\frac{\partial u_h}{\partial x_2} \right) \in V_h$ such that

$$\int_\Omega g_1 \mu_j \, dx = \int_\Omega \frac{\partial u_h}{\partial x_1} \mu_j \, dx \qquad \text{for } j \in \{1, 2, \ldots, N\}, \tag{1}$$

$$\int_\Omega g_2\mu_j\,dx = \int_\Omega \frac{\partial u_h}{\partial x_2}\mu_j\,dx \qquad \text{for } j \in \{1,2,\dots,N\}, \tag{2}$$

where $\mu_j \in M_h$ and M_h is a piecewise polynomial space whose basis is constructed to satisfy a biorthogonal relation with the basis for V_h.

Take the basis of V_h, $\{\phi_1,\dots,\phi_N\}$, to make the basis of M_h, $\{\mu_1,...,\mu_N\}$, by satisfying the biorthogonality relation

$$\int_\Omega \phi_i\mu_j\,dx = c_j\delta_{i,j}\,, \qquad c_j \neq 0\,, \quad i,j \in \{1,2,\dots,N\}.$$

Since $g_1, g_2 \in V_h$, we express them as $g_1 = \sum_{i=1}^N a_i\phi_i$ and $g_2 = \sum_{i=1}^N b_i\phi_i$, where $a_i, b_i \in \mathbb{R}$. Substituting into (1) and (2) gives

$$\sum_{i=1}^N a_i \int_\Omega \phi_i\mu_j\,dx = \int_\Omega \frac{\partial u_h}{\partial x_1}\mu_j\,dx \qquad \text{for } j \in \{1,2,\dots,N\}, \tag{3}$$

$$\sum_{i=1}^N b_i \int_\Omega \phi_i\mu_j\,dx = \int_\Omega \frac{\partial u_h}{\partial x_2}\mu_j\,dx \qquad \text{for } j \in \{1,2,\dots,N\}. \tag{4}$$

Writing (3) in matrix form gives

$$\begin{bmatrix} \int_\Omega \phi_1\mu_1\,dx & \dots & \int_\Omega \phi_N\mu_1\,dx \\ \vdots & \ddots & \vdots \\ \int_\Omega \phi_1\mu_N\,dx & \dots & \int_\Omega \phi_N\mu_N\,dx \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} \int_\Omega \frac{\partial u_h}{\partial x_1}\mu_1\,dx \\ \vdots \\ \int_\Omega \frac{\partial u_h}{\partial x_1}\mu_N\,dx \end{bmatrix},$$

which gives

$$D\vec{a} = \vec{f_1}\,,$$

where D is a diagonal matrix. This is a computationally inexpensive process since D is a diagonal matrix. The same process is then applied to equation (4) to find $\vec{b} = (b_1, b_2, \dots, b_N)$. This gives the functions g_1 and g_2.

2.2 Boundary modification for gradient

A higher convergence rate is desirable because it requires fewer mesh refinements to obtain a more accurate result. Nodes positioned on the boundary achieve a rate of convergence of $O\left(h^{1.5}\right)$ when biorthogonal projection is applied to recover the gradient [8, 11]. With boundary modification, the rate is raised to $O\left(h^2\right)$ over the entire domain [8, 11].

This boundary modification is implemented by first defining the set of nodes on the boundary and interior nodes, respectively, as

$$\mathcal{N}^{out} = \{x : x \in \mathcal{N}, x \in \partial\Omega\}, \qquad \mathcal{N}^{in} = \mathcal{N} \setminus \mathcal{N}^{out}.$$

We then expand this notion to the elements, such that the set of elements that lie inside of the domain and that lie on the boundary, respectively, are

$$\mathcal{T}^{in} = \{T : T \in \mathcal{T}_h, x \notin \overline{T}, \forall x \in \mathcal{N}^{out}\}, \qquad \mathcal{T}^{out} = \mathcal{T}_h \setminus \mathcal{T}^{in}.$$

For each $x_i \in \mathcal{N}^{out}$ we find the closest element $T_i \in \mathcal{T}^{in}$, where we judge the distance from the element as the distance between x_i and the centre of the element. We denote the nodes in T_i as x_{i_j} and their corresponding functions as ϕ_{i_j}, where $j \in \{1, 2, 3\}$. The modification to the basis functions is:

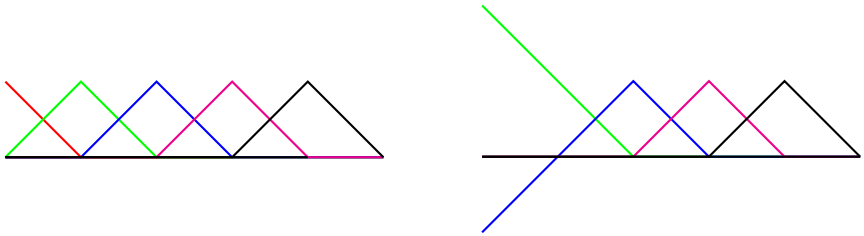
1. remove the basis function ϕ_i;
2. replace the basis functions ϕ_{i_j} with the functions

$$\tilde{\phi}_{i_j} = \phi_{i_j} + \alpha_j \phi_i, \qquad j \in \{1, 2, 3\},$$

where the α_j are scalars satisfying

$$\sum_{j=1}^{2} \alpha_j p\left(x_{i_j}\right) = p\left(x_i\right), \qquad p \in \mathcal{P}_1(\Omega).$$

The calculation of α_j is equivalent to finding the barycentric coordinates of x_i in relation to T_i. We denote this boundary modified projection as Π_h^{G*} that is proven to have an $O\left(h^2\right)$ approximation property for a uniform mesh [8].



(a) Before boundary modification (b) After boundary modification

Figure 1: 1D version of the boundary modification process, where the red basis function is removed and the blue and green functions are modified.

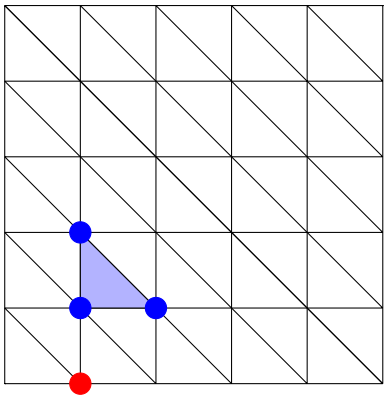


Figure 2: The red node corresponds to the function being removed and the blue triangle is the closest element that belongs to $\mathcal{T}^{\text{in}}$. The three blue nodes correspond to the functions that are being modified.

A visual representation of the function modification is shown in Figure 1 and the closest element selection in Figure 2.

3 Hessian recovery method

To recover the Hessian, we present three methods. The first method takes the derivative of the gradient recovery approximation, which we denote as

$$\Pi_h^{GD}(u) = \begin{bmatrix} \frac{\partial}{\partial x}\Pi_h^{G*}\left(\frac{\partial u}{\partial x}\right) & \frac{\partial}{\partial x}\Pi_h^{G*}\left(\frac{\partial u}{\partial y}\right) \\ \frac{\partial}{\partial y}\Pi_h^{G*}\left(\frac{\partial u}{\partial x}\right) & \frac{\partial}{\partial y}\Pi_h^{G*}\left(\frac{\partial u}{\partial y}\right) \end{bmatrix}.$$

The second method applies the gradient recovery with boundary modification to the first method:

$$\Pi_h^{GG}(u) = \begin{bmatrix} \Pi_h^{G*}\left(\frac{\partial}{\partial x}\Pi_h^{G*}\left(\frac{\partial u}{\partial x}\right)\right) & \Pi_h^{G*}\left(\frac{\partial}{\partial x}\Pi_h^{G*}\left(\frac{\partial u}{\partial y}\right)\right) \\ \Pi_h^{G*}\left(\frac{\partial}{\partial y}\Pi_h^{G*}\left(\frac{\partial u}{\partial x}\right)\right) & \Pi_h^{G*}\left(\frac{\partial}{\partial y}\Pi_h^{G*}\left(\frac{\partial u}{\partial y}\right)\right) \end{bmatrix}.$$

When testing this second method, we observe that the Hessian error converges at a faster rate when the boundary nodes and nodes that are directly connected to a boundary node are removed, as shown in Figure 3. The third Hessian recovery method uses a new boundary modification discussed in the next section.

3.1 New boundary modification

The new boundary modification for the Hessian recovery is similar to that used for gradient recovery. For the boundary modification, we define the set of nodes on the outer two levels of the boundary to be $\mathcal{N}_2^{out}$, and the second level interior nodes $\mathcal{N}_2^{in}$:

$$\mathcal{N}_2^{out} = \left\{p : p \in \mathcal{N}, \exists T \in \mathcal{T}^{out} \text{ s.t. } p \in \bar{T}\right\}, \qquad \mathcal{N}_2^{in} = \mathcal{N} \setminus \mathcal{N}_2^{out}.$$

Similarly for the elements:

$$\mathcal{T}_2^{in} = \left\{T : T \in \mathcal{T}_h, p \notin \bar{T}, \forall p \in \mathcal{N}_2^{out}\right\}, \qquad \mathcal{T}_2^{out} = \mathcal{T}_h \setminus \mathcal{T}_2^{in}.$$

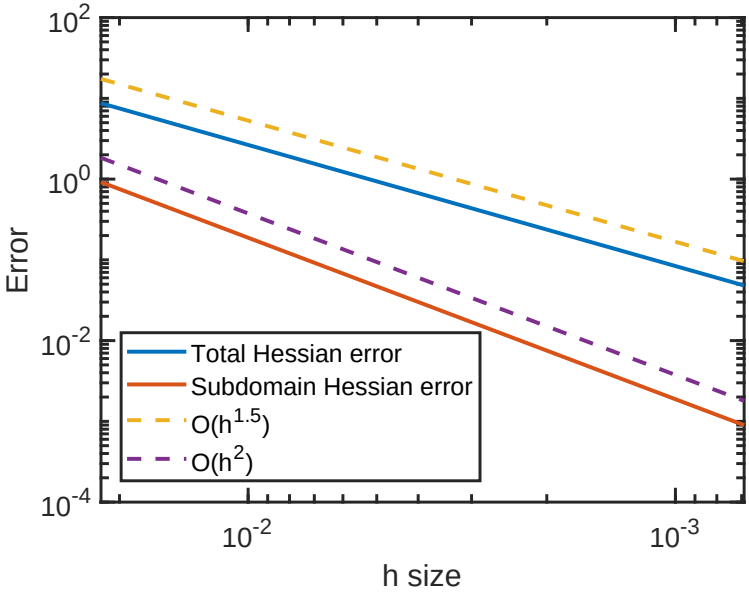


Figure 3: The Hessian error over the entire domain against the Hessian error over the domain where the two outermost boundary layers are not included.

We now remove all degrees of freedom in $\mathcal{N}_2^{\text{out}}$. The modification of the basis functions follows the same procedure as the gradient modification outlined in Section 2.2. We denote this boundary modified projection as Π_h^{H*}.

When calculating the approximate Hessian with this boundary modification, we use

$$\Pi_h^{GH}(u) = \begin{bmatrix} \Pi_h^{H*}\left(\frac{\partial}{\partial x}\Pi_h^{G*}\left(\frac{\partial u}{\partial x}\right)\right) & \Pi_h^{H*}\left(\frac{\partial}{\partial x}\Pi_h^{G*}\left(\frac{\partial u}{\partial y}\right)\right) \\ \Pi_h^{H*}\left(\frac{\partial}{\partial y}\Pi_h^{G*}\left(\frac{\partial u}{\partial x}\right)\right) & \Pi_h^{H*}\left(\frac{\partial}{\partial y}\Pi_h^{G*}\left(\frac{\partial u}{\partial y}\right)\right) \end{bmatrix}.$$

4 Adaptive mesh refinement

Adaptive mesh refinement works in two steps, the first being element selection, and the second being mesh refinement. Element selection is accomplished by applying a local error estimator $\eta_T(\mathcal{T}_h; u_h) \geqslant 0$ to each element $T \in \mathcal{T}_h$, and a global error estimator

$$\eta(\mathcal{T}; u_h) = \sqrt{\sum_{T \in \mathcal{T}} \eta_T(\mathcal{T}; u_h)^2},$$

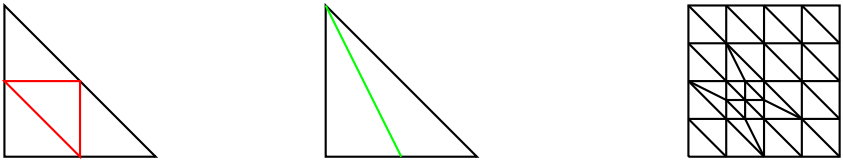
over the whole domain Ω. The results of the error estimators are then used with the Dörfler marking scheme to select which elements are refined. The Dörfler marking scheme creates $\mathcal{M} \subset \mathcal{T}$ to have minimal cardinality while also satisfying

$$\theta\eta(\mathcal{T}; u_h)^2 \leqslant \sum_{T \in \mathcal{M}} \eta_T(\mathcal{T}; u_h)^2,$$

where $\theta \in [0, 1]$ is a fixed parameter. The set $\mathcal{M}$ is the list of elements that require refinement. With this list of elements $\mathcal{M}$, we apply Red-Green refinement to refine the mesh [1, 4]. A red refinement is first applied to all the elements selected. Those elements that have a common edge with a red refined element, but are not red refined themselves, are marked for green refinement. If an element is marked for green refinement more than once, then a red refinement is applied to that element. Once all red refinements are completed, then those elements marked for green refinement are refined.

5 Results

Here we compare how the Hessian recovery methods work with and without the additional boundary modification in the uniform mesh refinement setting as well as the adaptive mesh refinement setting. To demonstrate the effectiveness of the methods, the sample problem is the Poisson problem in 2D: find $u(x, y)$



(a) Red refinement (b) Green refinement (c) Adaptive mesh refinement

Figure 4: Adaptive refinements.

such that

$$\begin{aligned} -\Delta u &= f \quad \text{in } \Omega\,, \\ u &= 0 \quad \text{on } \partial\Omega\,, \end{aligned} \tag{5}$$

where $f \in L^2(\Omega)$. For the adaptive refinement, we use a modified residual as the local error estimator:

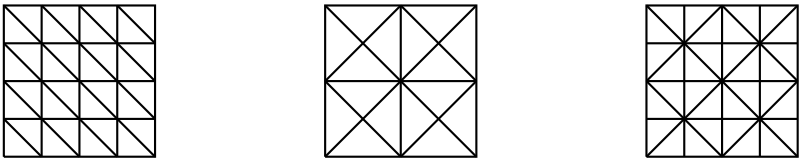
$$\eta_T(\mathcal{T}; u_h) = \sqrt{h^2 \int_T \left[H_{1,1}(u_h) + H_{2,2}(u_h) + f\right]^2 dx}\,,$$

where H is the recovered Hessian. The error of the recovered Hessian is $E(u_h) = \|\Delta u - H\|_F$. The error of the three recovery approaches E^D, E^G and E^H correspond to $H = \Pi_h^{GD}$, Π_h^{GG} and Π_h^{GH}, respectively. The norm used is the Frobenius norm

$$\|A\|_F = \sqrt{\sum_i^n \sum_j^m \|a_{i,j}\|^2_{L^2(\Omega)}}\,, \quad A \in \mathbb{R}^{n\times m}\,.$$

When testing in the uniform refinement setting, the three Hessian recovery methods are applied to uniform, criss cross and union jack meshes. These are standard meshes that have uniform elements shown in Figure 5. The finite element solution used while testing the uniformly refined meshes is

$$u_h = \sum_{i=1}^{N} u(x_i)\,\phi_i\,. \tag{6}$$



(a) Uniform mesh (b) Criss cross mesh (c) Union Jack mesh

Figure 5: The three mesh types used to test the proposed recovery methods, all have uniformly sized elements.

This is used to test the rate of convergence of the recovered Hessian in the uniform refinement setting of Sections 5.1 and 5.2.

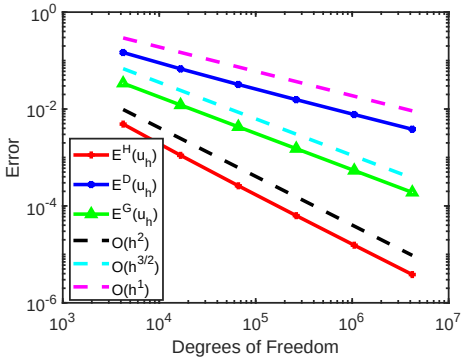
The Hessian recovery is also used in the setting of adaptive mesh refinement, where the approximate solution u_h is obtained from a Poisson FEM solver. The u_h has the Hessian recovery applied, and the recovered Hessian is used to direct the refinement of the mesh. The examples in Sections 5.1 and 5.2 demonstrate the viability of the recovered Hessian for directing the mesh refinement to maintain the L^2 and H^1 convergence rates of the obtained approximate solution u_h.

5.1 Example 1

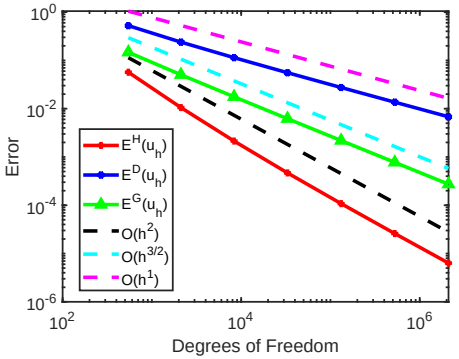
For this example, the domain is $\Omega = [0,1]^2$. The exact solution to the Poisson problem (5) is

$$u(x,y) = e^x\left(x^2+y^2\right) + y^2\cos(xy) + x^2\sin(xy).$$

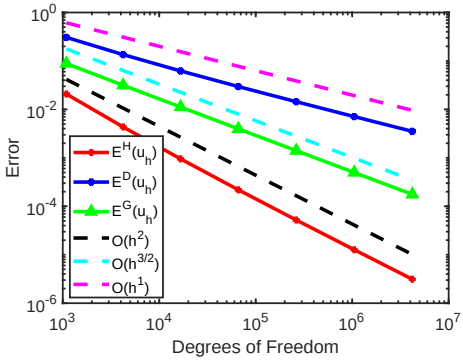
First we apply the Hessian recovery to the uniform refinement setting where refinement of the meshes is uniform. Figure 6 shows the Hessian error of the three recovery methods when applied to the finite element solution u_h for uniformly refined meshes. The Hessian error is consistent across the uniform, criss cross and Union Jack meshes for each of the recovery methods. The modified projection operators Π_h^{GD}, Π_h^{GG} and Π_h^{GH} yield convergence rates



(a) Uniform mesh



(b) Criss cross mesh

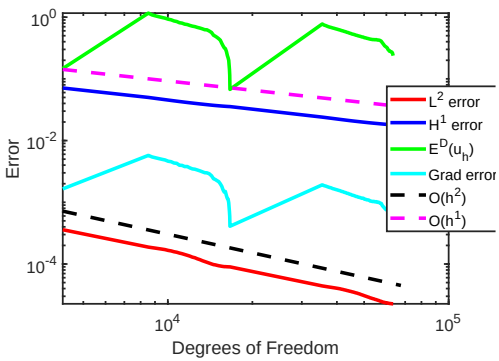


(c) Union Jack mesh

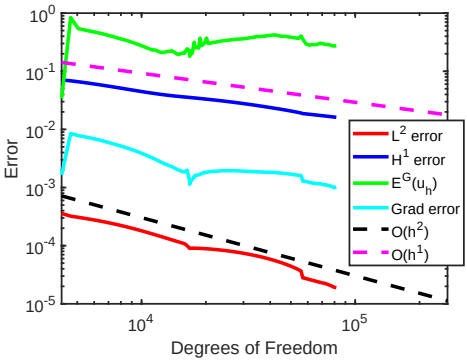
Figure 6: The performance of the proposed Hessian recovery methods on each type of uniformly refined mesh for Example 1 of Section 5.1.

of $O(h)$, $O(h^{1.5})$ and $O(h^2)$, respectively. This shows that the modified boundary method presented for the Hessian recovery performs the best of the three methods, but it requires that the element sizes be uniform across the mesh.

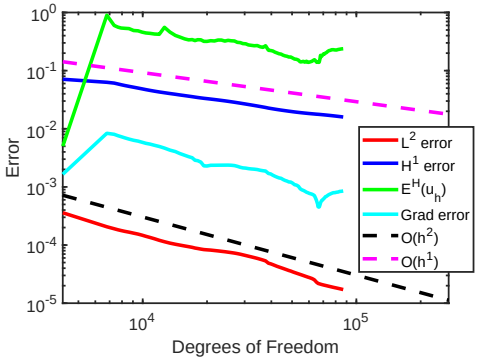
Second, we apply the Hessian recovery in the adaptive refinement setting where the mesh refinement is directed by the recovered Hessian and the finite element solution u_h is obtained from a finite element method (FEM) Poisson solver at each iteration. In Figure 7 we observe that the recovered



(a)



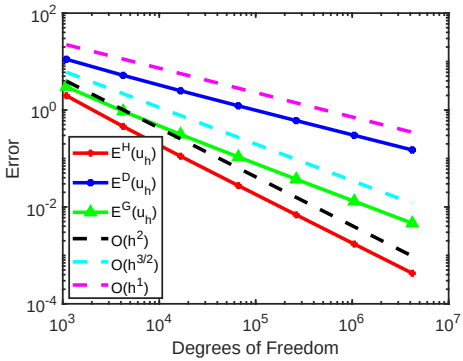
(b)



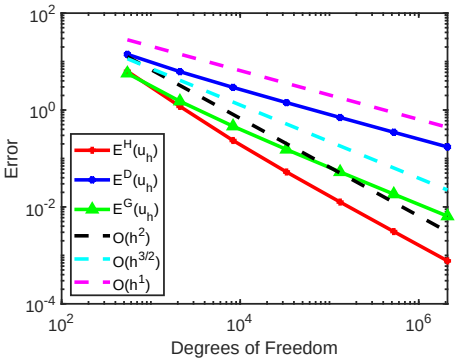
(c)

Figure 7: Adaptive refinement results for Example 1 of Section 5.1. The errors of the Hessians recovered by (a) Π_h^{GD}, (b) Π_h^{GG} and (c) Π_h^{GH}. The Dörfler marking scheme was used with $\theta = 0.75$ for element selection.

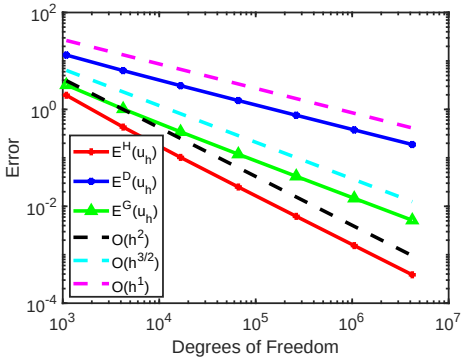
gradient and recovered Hessian have no guarantee to decrease as the degrees of freedom increase [9]. The recovered Hessian and recovered gradient have a similar error pattern, which is expected since the recovered Hessian is obtained from the recovered gradient. Despite this non-convergence of the recovered Hessian, when it is used to choose elements for refinement, the L^2 and H^1 errors decrease with optimal convergence rates $O\left(h^2\right)$ and $O\left(h\right)$, respectively. This demonstrates that using the recovered Hessian for directing the adaptive mesh refinement is viable.



(a) Uniform mesh



(b) Criss cross mesh



(c) Union Jack mesh

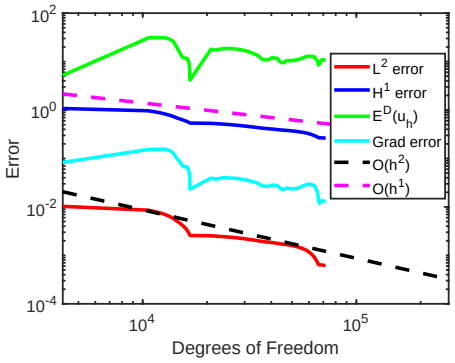
Figure 8: The performance of the proposed Hessian recovery methods on each type of uniformly refined mesh for Example 2 of Section 5.2.

5.2 Example 2

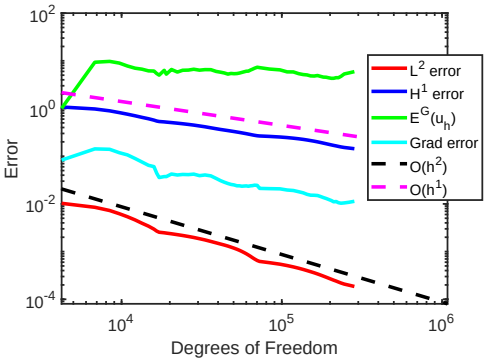
For this example, the domain is $\Omega = [0,1]^2$. The exact solution to the Poisson problem (5) is

$$u(x,y) = 2\pi^2 \sin(\pi x)\sin(\pi y).$$

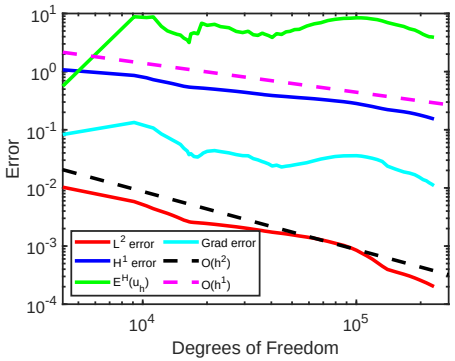
Results presented in Figures 8 and 9 are consistent with those in Example 1 of Section 5.1. All the Hessian recovery methods preserve the $O(h^2)$ and $O(h)$



(a)



(b)



(c)

Figure 9: Adaptive refinement results for Example 2 of Section 5.2. The errors of the Hessians recovered by (a) Π_h^{GD}, (b) Π_h^{GG} and (c) Π_h^{GH}. The Dörfler marking scheme was used with $\theta = 0.75$ for element selection.

convergence rates for the L^2 and H^1 errors in the adaptive mesh refinement setting. The uniform refinement results also show that Π_h^{GH} has the best convergence rate of $O\left(h^2\right)$ and is consistent over the three different mesh types.

References

[1] R. E. Bank, A. H. Sherman, and A. Weiser. "Some refinement algorithms and data structures for regular local mesh refinement". In: *Scientific computing: Applications of mathematics and computing to the physical sciences*. Ed. by R. S. Stepleman. North-Holland Publishing, 1983, pp. 3–17 (cit. on p. C138).

[2] R. E. Bank and J. Xu. "Asymptotically exact a posteriori error estimators, Part I: Grids with superconvergence". In: *SIAM J. Numer. Anal.* 41 (2003), pp. 2294–2312. DOI: 10.1137/S003614290139874X (cit. on p. C132).

[3] J. H. Bramble and A. H. Schatz. "Higher order local accuracy by averaging in the finite element method". In: *Math. Comput.* 31.137 (1977), pp. 94–111. DOI: 10.2307/2005782. (Cit. on p. C132).

[4] S. A. Funken and A. Schmidt. "Adaptive mesh refinement in 2D—An efficient implementation in Matlab". In: *Comput. Meth. Appl. Math.* 20.3 (2020), pp. 459–479. DOI: doi:10.1515/cmam-2018-0220. (Cit. on p. C138).

[5] H. Guo, Z. Zhang, and R. Zhao. "Hessian recovery for finite element methods". In: *Math. Comput.* 86.306 (2017), pp. 1671–1692. URL: https://www.jstor.org/stable/90004689 (cit. on p. C132).

[6] B.-O. Heimsund, X.-C. Tai, and J. Wang. "Superconvergence for the gradient of finite element approximations by L2 projections". In: *SIAM J. Numer. Anal.* 40.4 (2002), pp. 1263–1280. DOI: 10.1137/S003614290037410X. (Cit. on p. C132).

[7] Y. Huang and N. Yi. "The superconvergent cluster recovery method". In: *J. Sci. Comput.* 44 (2010), pp. 301–322. DOI: 10.1007/s10915-010-9379-9 (cit. on p. C132).

[8] M. Ilyas, B. P. Lamichhane, and M. H. Meylan. "A gradient recovery method based on an oblique projection and boundary modification". In: *Proceedings of the 18th Biennial Computational Techniques and Applications Conference, CTAC-2016*. Ed. by J. Droniou, M. Page, and S. Clarke. Vol. 58. ANZIAM J. Aug. 2017, pp. C34–C45. DOI: 10.21914/anziamj.v58i0.11730 (cit. on pp. C131, C134).

[9] L. Kamenski and W. Huang. "How A nonconvergent recovered Hessian works in mesh adaptation". In: *SIAM J. Numer. Anal.* 52.4 (2014), pp. 1692–1708. URL: http://www.jstor.org/stable/24512164 (cit. on p. C142).

[10] M. Křížek and P. Neittaanmäki. "Superconvergence phenomenon in the finite element method arising from averaging gradients". In: *Numer. Math.* 45 (1984), pp. 105–116. DOI: 10.1007/BF01379664 (cit. on p. C132).

[11] B. P. Lamichhane and J. Shaw-Carmody. "A gradient recovery approach for nonconforming finite element methods with boundary modification". In: *Proceedings of the 19th Biennial Computational Techniques and Applications Conference, CTAC-2020*. Ed. by W. McLean, S. Macnamara, and J. Bunder. Vol. 62. ANZIAM J. Feb. 2022, pp. C163–C175. DOI: 10.21914/anziamj.v62.16032 (cit. on pp. C131, C134).

[12] A. Naga and Z. Zhang. "A posteriori error estimates based on the polynomial preserving recovery". In: *SIAM J. Numer. Anal.* 42.4 (2004), pp. 1780–1800. DOI: 10.1137/S0036142903413002 (cit. on p. C132).

[13] Z. Zhang and A. Naga. "A new finite element gradient recovery method: Superconvergence property". In: *SIAM J. Sci. Comput.* 26.4 (2005), pp. 1192–1213. DOI: 10.1137/S1064827503402837 (cit. on p. C132).

[14] O. C. Zienkiewicz and J. Z. Zhu. "The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique". In: *Int. J. Numer. Meth. Eng.* 33 (1992), pp. 1331–1364. DOI: 10.1002/nme.1620330702 (cit. on p. C132).

Author addresses

1. **Jordan Shaw-Carmody**, School of Information & Physical Sciences (Mathematics), University of Newcastle, University Drive, Callaghan, NSW 2308, Australia
mailto:Jordan.Shaw-Carmody@uon.edu.au
orcid:0000-0002-6034-4644

2. **Bishnu P. Lamichhane**, School of Information & Physical Sciences (Mathematics), University of Newcastle, University Drive, Callaghan, NSW 2308, Australia
mailto:Bishnu.Lamichhane@newcastle.edu.au
orcid:0000-0002-9184-8941