ZIBELINE INTERNATIONAL PUBLISHING

ISSN: 2521-0831(Print)
ISSN: 2521-084X (Online)
CODEN : MSMADH

Matrix Science Mathematic (MSMK)

DOI : http://doi.org/10.26480/msmk.02.2019.27.31

RESEARCH ARTICLE

WEIGHTED TECHNIQUE FOR FINITE ELEMENT GRADIENT RECOVERY AT BOUNDARY

Y. Kashwaa, A. Elsaid*, M. El-Agamy

Mathematics and Engineering Physics Department, Faculty of Engineering, Mansoura University, P.O. 35516, Mansoura, Egypt.
**Corresponding Author E-mail: a_elsaid@mans.edu.eg*

This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

ARTICLE DETAILS

Article History:

Received 27 December 2019
Accepted 28 January 2020
Available online 07 February 2020

ABSTRACT

In this paper, an improved technique is presented to recover the finite element gradient at boundaries. The proposed technique begins by evaluating the recovered gradient at the interior nodes using polynomial preserving recovery technique. Then we propose formula for weights to the recovered gradient at the interior nodes attached to boundary nodes. The sum of these weighted recovered gradients is utilized as an approximation for the gradient at the attached boundary node. The validity of the proposed technique is illustrated by some two-dimensional numerical examples.

KEYWORDS

Gradient at boundary, Polynomial preserving recovery, Finite element method, Poisson equation.

1. INTRODUCTION

Gradient recovery techniques are post-processing methods that reconstruct numedical approximations from finite element solution to obtain a better numerical gradient (recovered gradient). Several ideas for gradient recovery techniques were proposed by researchers as the work in [1-3]. But these techniques suffered some drawbacks which include long execution time, complexity, requiring meshes with special structure, or the necessity of using low-order finite elements. The first technique to overcome the above-mentioned drawbacks was developed in [4] and called the super convergent patch recovery (SPR). The advantages that are noticed in SPR include super convergence of the recovered gradients, its robustness as a posteriori error estimator, and its efficient execution [5-11].

In some studies the polynomial preserving recovery (PPR) technique was proposed and it was demonstrated that the PPR is better than the SPR [7,11]. The idea behind recovering the gradient using the PPR technique starts by construction a patch of elements around the targeted node. Then, solution at the nodes of this patch of elements is fitted to a second-order polynomial using least-squares approach. The gradient of this fitting polynomial is the recovered gradient at this node.

The results obtained using PPR often yields a higher-order gradient approximation on the patch of mesh elements around each mesh vertex. On the other hand, the accuracy of PPR near boundaries of the considered domain is not as good as that obtained at nodes in the interior of the domain. So, some treatments are required to improve the accuracy of PPR on boundaries. A gradient recovery technique on boundary node was proposed [13]. The authors utilized the PPR least squares fitting technique to construct a polynomial for each selected interior node attached to the target boundary node. Then they computed the recovered gradient at this boundary node as the gradient of the polynomial obtained by averaging the polynomials constructed at the attached internal nodes. The numerical examples they presented show that in some cases the recovered gradient evaluated using this technique has better accuracy than the classical PPR.

In this paper, we propose a weighted technique to obtain a better recovered gradient at nodes on the boundary. Our technique is performed in two stages: First the standard PPR is applied to each selected interior node attached to the target boundary node. Then we utilize an error estimator to assign weights for the obtained recovered gradients and substitute with the boundary node in the sum of these weighted recovered gradients. The rest of this article is organized as follows. In section 2, the PPR method and the procedure of adaptive finite element method are discussed. In section 3, we introduce the mathematical formulation of our technique. Section 4 contains some numerical examples for applying the proposed technique to Poisson equation in two-dimensional domains. Polygonal domains with different types of corners are considered and results are shown for both uniform and adaptive refinement. Finally section 5 presents the conclusion of this work.

2. MATHEMATICAL PRELIMINARIES

2.1 Basic concepts of PPR

Let v denote a mesh node, n denote a positive integer, and $\ell(v, n)$ denote the union of elements in the first n layers around v, i.e.,

$$\ell(v, n) = \cup\{\tau \in \tau_h, \tau \cap \ell(v, n-1) = \emptyset\}, \quad (1)$$

where $\ell(v,0) = \{v\}$ and τ_h denotes consequence of triangulation of the finite element mesh.

Let the space of continuous linear finite element space and the set of all mesh nodes be denoted by V_h and N_h, respectively. The standard Lagrange basis of V_h is denoted by $\{\varphi_v, v \in N_h\}$ with $\{\varphi_v(v') = \delta_{vv'}\}$. Let the gradient recovery PPR operator be denoted by G_h such that $G_h: V_h \rightarrow V_h \times V_h$. For a mesh node v, let α_v be a patch of elements around v. The polynomial $p_v \in P_{n+1}(\alpha_v)$ is to be fitted in the least square sense at sampling points that consist of the set of all nodes in $N_h \cap \alpha_v$, i.e.

$$p_v = \arg \min_{p_v \in P_{n+1}(\alpha_v)} \sum_{\bar{v} \in N_h \cap \alpha_v} (u_h - p)^2(\bar{v}), \quad (2)$$

where $P_k(\alpha_v)$ is the space of polynomials defined on α_v with a degree less than or equal to k. Then

$$(G_h u_h)(v) = \nabla P_v(v). \quad (3)$$

The finite element representation of the recovered gradient on the entire domain is given by

$$(G_h u_h) = \sum_{v \in N_h} (G_h u_h)(v)\, \varphi_v. \quad (4)$$

For an interior node v, we define ϑ_v as the smallest $\ell(v, n)$ that ensures the uniqueness of the fitting polynomial p_v [6,7]. In the case that v is a boundary node, let n_0 be the smallest positive integer such that $\ell(v, n_0)$ has at least one interior mesh node. Then, we define

$\vartheta_v = \ell(v, n_0) \cup \{\vartheta_{\tilde{v}} : \tilde{v} \in \ell(v, n_0) : \text{ and } \tilde{v} \text{ an interior vertex}\}$.

In a study the authors proved, that is if $u \in W_\infty^{n+2}(\vartheta_v)$, then [12]

$$\|\nabla u - G_h u_h\|_{L\infty} \le C h^{n+2} \|u\|_{W_\infty^{n+2}(\vartheta v)} \quad (5)$$

where W_∞^{n+2} denotes the standard Sobolev space of order n + 2.

Then G_h preserves polynomials of degrees up to n + 1 in the domain Ω. For even n, v is a vertex with symmetric mesh, and $u \in W_\infty^{n+3}(\vartheta_v)$, then

$$\|\nabla u - G_h u_h\|_{L\infty} \le C h^{n+2} \|u\|_{W_\infty^{n+3}(\vartheta v)} \quad (6)$$

and G_h preserves the gradient of polynomials of degree up to n + 2.

2.2 Adaptive finite element method

In finite element adaptive techniques, the refinement process is controlled via a chosen error estimator. The iterative post-processing procedures are carried out until this estimator reaches a predefined tolerance. Some adaptive procedures based on finite element recovery techniques were utilized with different types of equations [14-16]. The adaptive technique begins by solving the considered problem on an initial mesh and a chosen gradient recovery technique is applied. Then for every element k, we calculate err_k which denotes the error between the recovered gradient and the finite element gradient on that element. A marking strategy is prescribed to refine some marked elements where the error estimator is greater than the prescribed tolerance. The mesh is then adapted, and the process is repeated until the tolerance condition is satisfied. Marking the elements is characterized by the error (err_k) as follows:

If

$$(err_k > \mu \max(err)) \quad (7)$$

for the kth element then mark the kth element,

where μ is a ratio to be specified and max(err) is maximum local error

$$\max(err) = \max_{1 \le k \ge N_n} \{err_k\} \quad (8)$$

3. RECOVERY GRADIENT FOR BOUNDARY NODE

For any boundary node v, define

$$\alpha_v = \ell(v, n_0), \quad (9)$$

as the patch of all elements attached to v with no as the smallest positive integer that ensures that $\ell(v, n_0)$ contains at least one interior vertex. Let v1, ..., v_n be all the interior vertices in α_v. Then we propose that the recovered gradient calculated at v be computed by

$$G_h u_h(v) = \sum_{i=1}^{n} w_i \nabla P_{v_i}(v) \quad (10)$$

where p_{vi} is the polynomial that fits u_h at the interior vertex v_i that belongs to α_{vi}, a well-defined element patch according to [17].

The idea of weight formula in recovery techniques has been employed before as in the work reported in [18]. The authors proposed weights that depend on a certain error estimator for improving the accuracy of flux computations in elliptic interface problems. In this work, the weight formula is chosen in one of two techniques: weighted average or minimum error technique.

3.1 Weighted average technique

At every interior node vi attached to the boundary node v, we calculate an error estimator $\epsilon(v_i)$ as

$$\epsilon(v_i) = \|G_h u_h(v_i) - \nabla u_h(v_i)\|, \quad (11)$$

where $G_h u_h(v_i)$ is the recovered gradient at node v_i and $\nabla u_h(v_i)$ is finite element gradient at node v_i for i=1, 2, ..., n. Then the weights w_i in (9) is defined as

$$w_i = \frac{2}{n} - \frac{\epsilon(v_i)}{\sum_{j=1}^{n} \epsilon(v_j)} \quad (12)$$

By this formula we aim to give higher weights to nodes with better FEM gradient, i.e. when the difference between $G_h u_h(v_i)$ and $\nabla u_h(v_i)$ is small.

3.2 Minimum error technique

In this technique we evaluate the error estimator defined by (11). Then we define ϵ_{min} as the minimum error estimator on all nodes v_i as

$$\epsilon_{min} = \min_{1 \le i \le n} \{\epsilon(v_i)\}. \quad (13)$$

Let K be the number of nodes v_i such that

$$\epsilon(v_i) = \epsilon_{min}, \text{ i=1, 2, ..., K} \quad (14)$$

$$w_i = \begin{cases} \frac{1}{K} & \epsilon(v_i) = \epsilon_{min,} \\ 0 & \epsilon(v_i) > \epsilon_{min}. \end{cases} \quad (15)$$

In case only one interior node v_j has the minimum value of error estimator ϵ_{min}, then the recovered gradient at boundary node v is simply the substitution of this boundary node in the recovered gradient at v_j.

4. NUMERICAL EXAMPLES

The Poisson equation is solved on polygonal domains with different types of corners. Different measures of error are presented in the following tables to compare the different recovery techniques namely the standard PPR technique, the weighted average technique (WE), the minimum error technique (MIN), and the average technique (AVG) presented by using uniform refinement. Figures are also given to illustrate error on boundary when using adaptive technique presented in section (2.2) where we utilize the error estimator defined by (11). We denote degrees of freedom by DOFs. Let $E_1(x_0, y_0)$ denote L_2 error computed at the patch of elements attached to corner (x_0, y_0), E_2 denotes L_2 error computed around all elements attached to boundary, and $E_3(x_0, y_0)$ denotes L_∞ error around corner (x_0, y_0).

Example 1

Consider the Poisson equation

$$-\nabla^2 u = f, \; u = 0, \text{ on } \partial\Omega, \quad (16)$$

and f is chosen such that the exact solution u(x, y) is given by

$$u(x, y) = xy(x^2 - 1)(y^2 - 1), \quad (17)$$

where Ω is the L-shaped domain [-1,1] × [-1,1] – [-1,0] ×[-1,0]. Figure (1) shows the domain with the initial mesh of Example (1).

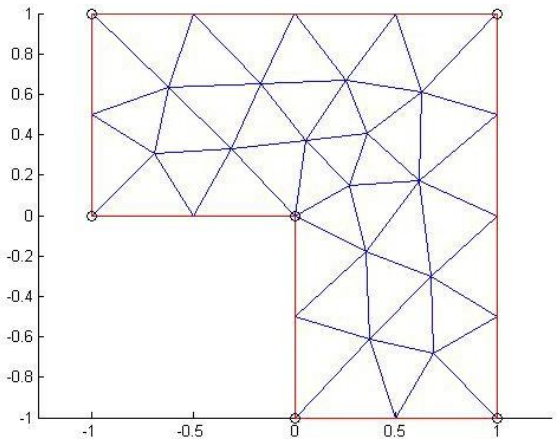

Figure 1: Domain and initial mesh of Example 1.

Table 1 shows that the four techniques present alternating performance at patch of elements attached to node (0,0) whereas Table 2 shows that the weighted average technique gives better results along the entire boundary elements. It can be seen in Table 3 that all strategies have enhanced the maximum error of gradient again with favorable results for the weighted average technique and alternating results among the other three techniques. Figure 2 and Figure 3 illustrate that all techniques nearly have the same convergence rate. Also a slight increase of accuracy is noticed for the average and the weighted average techniques over the minimum error technique.

Table 1: $E_1(0, 0)$ error difference of Example 1 using uniform refinement.

DOFs	WE-AVG	WE-MIN	WE-PPR	MIN-PPR	MIN-AVG
30	-1.66E-02	-1.8E-02	4.69E-03	2.29E-02	1.57E-03
101	-8.74E-04	2.24E-03	-1.39E-02	-1.61E-02	-3.11E-03
369	2.75E-05	-5.29E-05	5.15E-04	5.68E-04	8.05E-05

1409	-6.47E-07	-2.93E-06	4.91E-05	5.20E-05	2.28E-06
5505	8.95E-08	-5.58E-07	3.32E-06	3.88E-06	6.47E-07
21761	3.93E-08	-9.89E-08	2.19E-07	3.18E-07	1.38E-07

Table 2: E_2 error difference of Example 1 using uniform refinement.

DOFs	WE-AVG	WE-MIN	WE-PPR	MIN-PPR	MIN-AVG
30	1.68E-01	-3.806E-02	-5.705E-02	-1.89E-02	2.06E-01
101	4.43E-02	-3.42E-02	-3.74E-02	-3.209E-03	7.85E-02
369	-2.43E-03	-2.57E-02	-2.509E-02	6.58E-04	2.33E-02
1409	-1.21E-02	-1.85E-02	-1.84E-02	1.05E-04	6.41E-03
5505	-1.02E-02	-1.2E-02	-1.19E-02	1.64E-05	1.71E-03
21761	-6.65E-03	-7.102E-03	-7.1E-03	2.10E-06	4.43E-04

Table 3: $E_3(0, 0)$ of Example 1 using uniform refinement.

DOFs	WE-AVG	WE-MIN	WE-PPR	MIN-PPR	MIN-AVG
30	2.01E-02	-3.38E-02	6.83E-02	1.02E-01	5.401E-02
101	1.35E-03	-1.38E-02	1.94E-03	1.57E-02	1.24E-02
369	-3.35E-04	-2.03E-03	-4.05E-04	1.62E-03	1.69E-03
1409	-4.61E-05	1.04E-04	-2.15298E-05	-1.25E-04	-1.502E-04
5505	-1.26E-05	8.46E-05	1.1E-05	-7.36E-05	-9.72E-05
21761	-3.86E-06	2.12E-05	5.87E-06	-1.53E-05	-2.51E-05

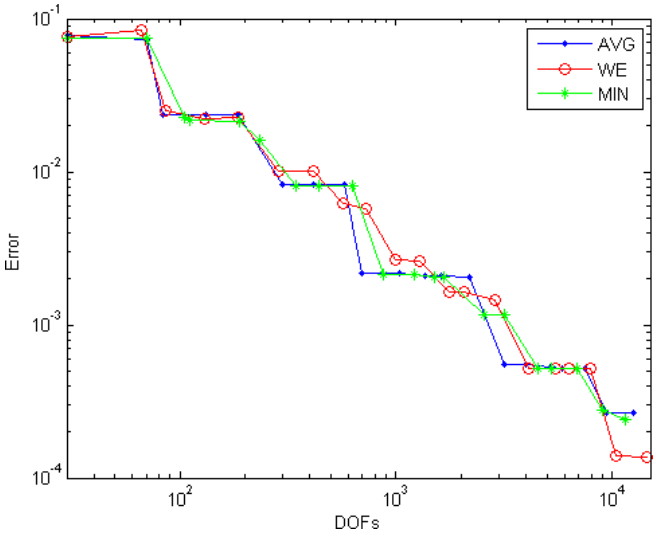

Figure 2: $E_1(0, 0)$ of Example 1 using adaptive refinement.

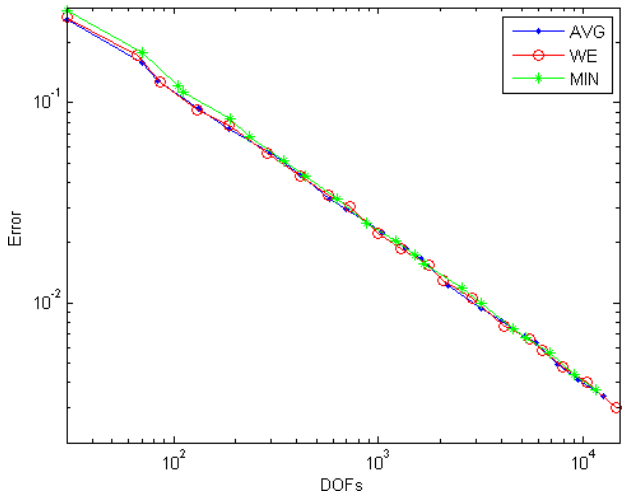

Figure 3: E_2 of Example 1 using adaptive refinement.

Example 2

Consider the following equation

$$-\nabla^2 u = f, \quad u = 0, \text{ on } \partial\Omega, \tag{18}$$

and f is chosen such that the exact solution u(x, y) is given by

$$u(x, y) = \sin(\pi x)\sin(\pi y), \tag{19}$$

where Ω denotes the domain [0,3] × [0,2] – [1,2] × [0,1]. Figure 4 shows the domain with the initial mesh of Example 2.

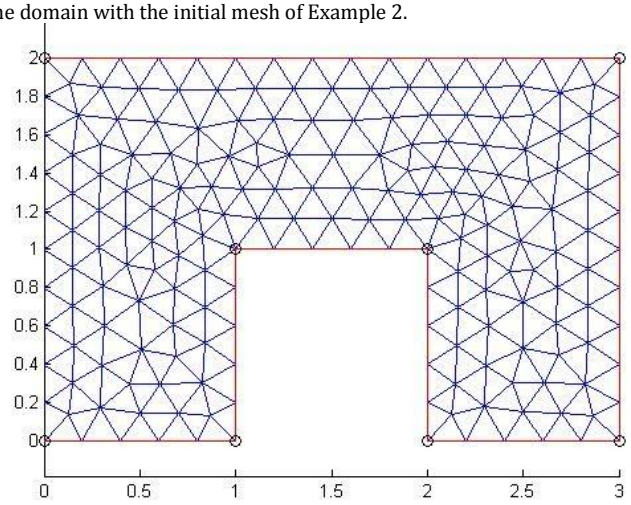

Figure 4: Domain and initial mesh of Example 2.

Table 4: $E_1(1, 1)$ error difference of Example 2 using uniform refinement.

Dofs	WE-AVG	WE-MIN	WE-PPR	MIN-PPR	MIN-AVG
203	-1.26E-03	1.61E-03	2.69E-03	1.07E-03	-2.88E-03
749	-6.78E-05	-1.25E-04	2.49E-04	3.75E-04	5.77E-05
2873	-1.64E-05	1.24E-05	5.09E-06	-7.31E-06	-2.88E-05
11249	2.26E-08	1.42E-07	1.44E-06	1.3E-06	-1.19E-07
44513	-7.49E-08	5.95E-08	2.14E-08	-3.81E-08	-1.34E-07
177089	-2.23E-09	6.84E-11	4.91E-09	4.84E-09	-2.3E-09

Table 5: $E_1(2, 1)$ error difference of Example 2 using uniform refinement.

Dofs	WE-AVG	WE-MIN	WE-PPR	MIN-PPR	MIN-AVG
203	-1.24E-03	1.16E-03	2.62E-03	1.46E-03	-2.41E-03
749	3.05E-05	-2.89E-04	3.46E-04	6.36E-04	3.2E-04
2873	-1.59E-05	9.14E-06	5.34E-06	-3.8E-06	-2.5E-05
11249	5.74E-08	1.22E-07	1.43E-06	1.3E-06	-6.54E-08
44513	-6.81E-08	3.12E-08	1.96E-08	-1.15E-08	-9.93E-08
177089	-7.1E-10	7.1E-11	5.01E-09	4.94E-09	-7.82E-10

Table 6: E_2 error difference of Example 2 using uniform refinement.

Dofs	WE-AVG	WE-MIN	WE-PPR	MIN-PPR	MIN-AVG
203	4.08E-03	-2.62E-02	-2.63E-03	-4.76E-05	3.03E-02
749	4.807E-04	-2.84E-03	-2.05E-03	7.95E-04	3.32E-03
2873	5.78E-05	-2.46E-04	-1.97E-04	4.86E-05	3.03E-04
11249	5.38E-06	-1.82E-05	-2.54E-05	-7.21E-06	2.36E-05
44513	6.68E-07	-1.45E-06	-3.59E-06	-2.14E-06	2.12E-06
177089	7.59E-08	-1.26E-07	-5.75E-07	-4.49E-07	2.01E-07

Table 7: $E_3(2, 1)$ of Example 2 using uniform refinement.

Dofs	WE-AVG	WE-MIN	WE-PPR	MIN-PPR	MIN-AVG
203	1.78E-01	1.37E-02	2.26E-01	2.13E-01	1.65E-01
749	-2.58E-03	-2.62E-02	-1.9E-03	2.43E-02	2.36E-02
2873	2.68E-03	-1.74E-03	1.779E-03	3.52E-03	4.43E-03
11249	-3.77E-04	-2.36E-04	-3.44E-04	-1.07E-04	-1.41E-04
44513	-9.06E-05	9.18E-05	-3.89E-05	-1.307E-04	-1.82E-04
177089	-3.43E-05	-2.99E-06	-1.28E-05	-9.89E-06	-3.13E-05

Table 8: $E_3(1, 1)$ of Example 2 using uniform refinement.

Dofs	WE-AVG	WE-MIN	WE-PPR	MIN-PPR	MIN-AVG
203	1.83E-01	1.13E-02	2.506E-01	2.39E-01	1.71E-01
749	2.45E-02	-7.105E-03	3.14E-02	3.85E-02	3.16E-02
2873	3.42E-03	-1.77E-03	2.42E-03	4.19E-03	5.19E-03

11249	-6.91E-05	-1.56E-04	-1.35E-04	2.12E-05	8.71E-05
44513	-7.23E-05	4.08E-05	-4.007E-05	-8.09E-05	-1.13E-04
177089	-2.7E-05	1.87E-06	-1.01E-05	-1.2E-05	-2.89E-05

Tables 4 and 5 show that the weighted average technique and minimum error technique presented better results than the average technique on the patch of elements attached to nodes (1,1) and (1,2), respectively. Yet the tables also indicate a better performance by the classical PPR technique. Table 6 shows that average technique gives better results along the entire boundary and again the weighted technique yields better accuracy than the classical PPR. In Table 7 and Table 8 we can see that weighted technique gives better maximum error of gradient. Figure 5 illustrates that weighted average technique has better results than minimum technique at elements attached to corner (2,1) but the three techniques have same convergence rate along the boundary shown in Figure 6.

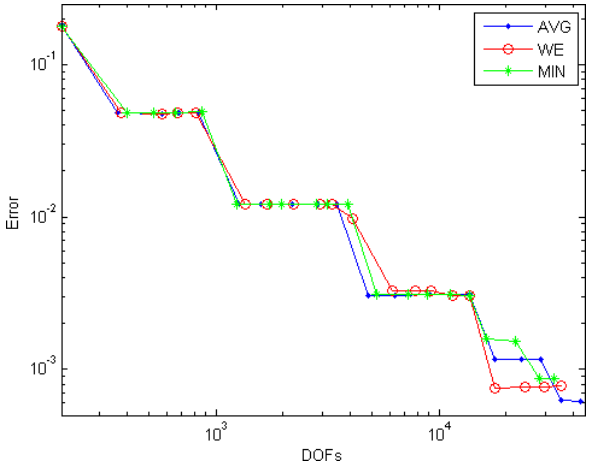

Figure 5: $E_1(2, 1)$ of Example 2 using adaptive refinement

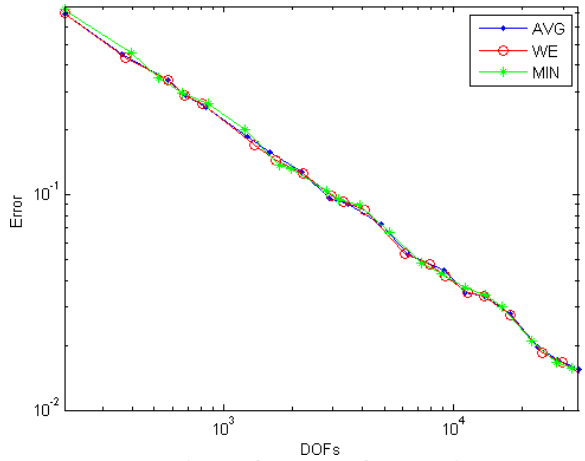

Figure 6: E_2 of Example 2 using adaptive refinement.

Example 3

Consider the following equation

$$-\nabla^2 u = f, \quad u = 0, \text{ on } \partial\Omega, \tag{20}$$

and f is chosen such that the exact solution u(x, y) is given by

$$u(x, y) = (y^2 - x^2)(x^2 - 4)(y^2 - 1), \tag{21}$$

where Ω denotes the domain $[-1,2] \times [-1,1] - \{[-1,1] \times [-1,0] \ \& \ |x| < y\}$.

Figure 7 shows the domain with the initial mesh of Example 3.

Table 9, Table 10 and Table 11 show that PPR technique presents better results along elements attached to corners. Table 12 shows that PPR and the two weighted techniques give results better than average technique along the boundary. Whereas Table 13 and Table 14 show that weighted techniques give better maximum error of gradient whereas Table 15 shows that average technique gives better maximum error at corner (0,0). Figure 8 shows that the weighted techniques give better results at elements attached to corner (0,0) but the three techniques have same convergence rate along the boundary shown in Figure 9.

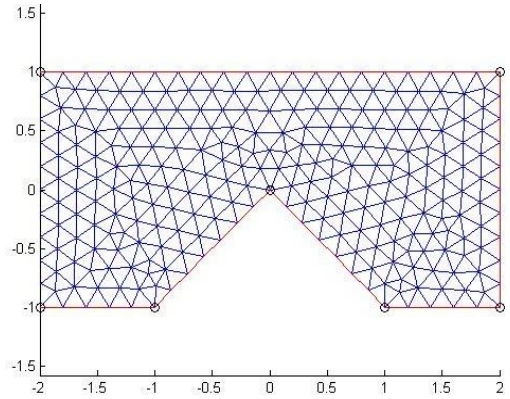

Figure 7: Domain and initial mesh of Example 3.

Table 9: $E_1(-1, -1)$ error difference of Example 3 using uniform refinement.

DOFs	WE-AVG	WE-MIN	WE-PPR	MIN-PPR	MIN-AVG
144	-7.7E-05	-2.32E-03	1.26E-04	2.44E-03	2.24E-03
527	7.24E-04	3.01E-03	8.19E-04	5.17 E-04	4.22E-04
2013	1.48E-04	5.52E-05	1.15E-04	6.007E-05	9.36E-05
7865	2.19E-05	7.46E-06	1.83E-05	1.08E-05	1.45E-05
31089	2.95E-06	9.58E-07	2.54E-06	1.58E-06	1.99E-06
123617	3.82E-07	1.2E-07	3.33E-07	2.12E-07	2.61E-07

Table 10: $E_1(1, -1)$ error difference of Example 3 using uniform refinement.

DOFs	WE-AVG	WE-MIN	WE-PPR	MIN-PPR	MIN-AVG
144	-3.27E-03	-6.66E-03	1.21E-03	7.87E-03	3.39E-03
527	1.19E-04	5.05E-04	6.52E-04	1.47E-04	-3.85E-04
2013	2.46E-06	-7.33E-05	7.37E-05	1.47E-04	7.58E-05
7856	-4.93E-07	-1.13E-05	1.16E-05	2.29E-05	1.08E-05
31089	-1.48E-07	-1.59E-06	1.59E-06	3.18E-06	1.44E-06
123617	-2.49E-08	-2.11E-07	2.07E-07	4.19E-07	1.86E-07

Table 11: $E_1(0, 0)$ error difference of Example 3 using uniform refinement.

DOFs	WE-AVG	WE-MIN	WE-PPR	MIN-PPR	MIN-AVG
144	3.67E-03	-4.53E-03	7.34E-03	1.18 E-02	8.201E-03
527	1.39E-04	-2.94E-04	4.13E-04	7.07E-04	4.34E-04
2013	6.79E-06	-1.28E-05	2.44E-05	3.73E-05	1.96E-05
7865	4.15E-07	-7.77E-07	1.47E-06	2.25E-06	1.19E-06
31089	2.73E-08	-4.79E-08	8.78E-08	1.35E-07	7.53E-08
123617	1.95E-09	-2.91E-09	4.81E-09	7.73E-09	4.86E-09

Table 12: E_2 error difference of Example 3 using uniform refinement.

Dofs	WE-AVG	WE-MIN	WE-PPR	MIN-PPR	MIN-AVG
144	-1.15E-02	-4.78E-02	-3.97E-02	8.068E-03	03.63E-02
527	-1.43E-03	-2.66E-03	4.46E-03	7.12E-03	1.22E-03
2013	-1.34E-04	1.06E-03	1.84E-03	7.83E-04	-1.19E-03
7865	3.34E-07	3.25E-04	4.38E-04	1.12E-04	-3.25E-04
31089	-9.09E-08	6.99E-05	8.48E-05	1.48E-05	-7.007E-05
123617	1.62E-08	1.42E-05	1.56E-05	1.43E-06	-1.420E-05

Table 13: $E_3(-1, -1)$ of Example 2 using uniform refinement.

Dofs	WE-AVG	WE-MIN	WE-PPR	MIN-PPR	MIN-AVG
144	-1.21E-01	4.07E-02	-3.04E-01	-3.45E-01	-1.61E-01
527	-6.82E-02	1.33E-02	-1.12E-01	-1.26E-01	-8.16E-02
2013	-1.87E-02	3.71E-03	-2.94E-02	-3.31E-02	-2.24E-02
7865	-4.88E-03	9.69E-04	-7.51E-03	-8.48E-03	-5.85E-03
31089	-1.24E-03	2.49E-04	-1.9E-03	-2.15E-03	-1.49E-03
123617	-3.15 E-04	6.35E-05	-4.78E-04	-5.41E-04	-3.78E-04

Table 14: $E_3(1, -1)$ of Example 2 using uniform refinement.

Dofs	WE-AVG	WE-MIN	WE-PPR	MIN-PPR	MIN-AVG
144	-2.05E-02	3.22E-02	-4.98E-01	-5.306E-01	-5.28E-02
527	-1.903E-02	6.67E-03	-9.905E-02	0.1.05E-01	-2.57E-02
2013	-5.01E-03	2.84E-03	-2.606E-02	-2.89E-02	-7.86E-03
7865	-1.28E-03	8.41E-04	-6.83E-03	-7.67E-03	-2.12E-03
31089	-3.25E-04	2.29E-04	-1.75E-03	-1.98E-03	-5.55E-04
123617	-8.23E-05	6.01E-05	-4.45E-04	-5.05E-04	-1.42E-04

Table 15: $E_3(0, 0)$ of Example 2 using uniform refinement.

Dofs	WE-AVG	WE-MIN	WE-PPR	MIN-PPR	MIN-AVG
144	1.03E-01	-2.27E-02	2.88E-01	3.11E-01	1.26E-01
527	8.108E-03	-4.37E-02	3.72E-02	8.105E-02	5.18E-02
2013	8.03E-04	-1.84E-03	4.64E-03	6.49E-03	2.65E-03
7865	9.89E-05	-2.55E-04	6.42E-04	8.98E-04	3.54E-04
31089	1.44E-05	-3.91E-05	1.02E-04	1.41E-04	5.35E-05
123617	2.54E-06	-7.59E-06	1.97E-05	2.73E-05	1.01E-05

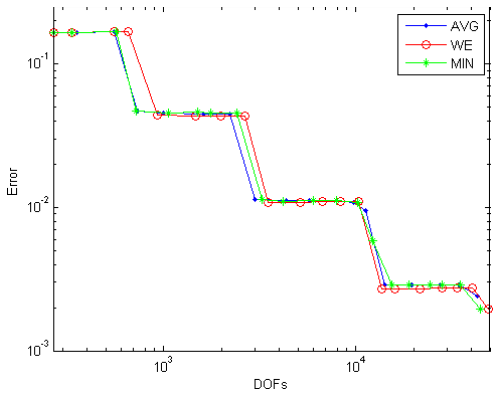

Figure 8: $E_1(0, 0)$ of Example 3 using adaptive refinement.

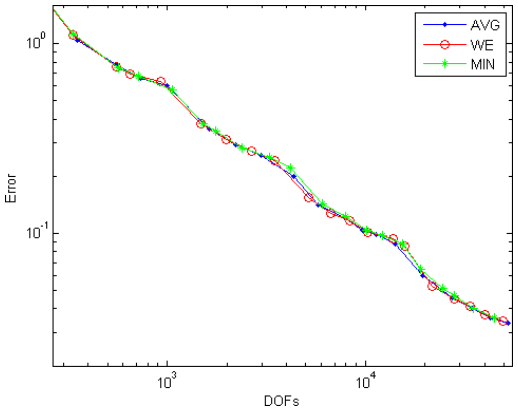

Figure 9: E_2 of Example 3 using adaptive refinement.

5. CONCLUSION

In this work, we have presented the idea of a weighted technique for gradient recovery at boundary nodes when using finite element method. The weights are proposed in two formulas based on an error estimator that measures the difference between the recovered gradient and the finite element gradient at a given node. The first formula, weighted average formula, takes the sum of recovered gradient at all interior nodes attached to the boundary node but with assigning higher weights to nodes with smaller error estimator value. The second formula only substitutes the boundary node in the average gradient of the nodes with minimum error estimator value.

The numerical examples considered Poisson equation on polygonal domains with different types of corners. Tables were presented to compare some error measures when applying different recovery techniques and using uniform refinement. In many cases, but not all, the weighted average technique yields better accuracy than the classical PPR and average technique. Also graphs were shown to illustrate how these techniques perform using adaptive refinement. Though weighted average technique presented higher accuracy especially with higher degrees of freedom, all techniques have almost the same convergence rate. It is shown that weighted technique may increase the accuracy but evidently all linear combinations of recovered gradients at interior nodes yields the same convergence rate as the degrees of freedom increase.

REFERENCES

[1] Levine, N., 1985. Superconvergent recovery of the gradient from piecewise linear Finite element approximations. IMA J.Numer. Anal., 5, 407-427.

[2] Oden, J., Brauchli, H., 1971. On the calculation of consistent stress distributions in Fnite element applications. Internat. J.Numer. Methods Engrg., 3, 317-325.

[3] Bramble, J., Schatz, A., 1977. Higher order local accuracy by averaging in the finite element method. Math. Comp., 31 (137), 94-111.

[4] Zienkiewicz, O., Zhu, J., 1992. The super convergent patch recovery and a posteriori error estimates, part I: the recovery technique. Internat. J. Numer. Methods Engrg., 33 (7), 1331-1364.

[5] Ainsworth, M., Oden, J., 2000. A Posteriori Error Estimation in Finite Element Analysis. Wiley Interscience, 37.

[6] Babuska, I., Strouboulis, T., 2001. The Finite Element Method and its Reliability, Oxford University Press, Oxford.

[7] Naga, A., Zhang, Z., 2005b. The polynomial-preserving recovery for higher order finite element methods in 2D and 3D, Discrete Contin. Dyn. Syst. Ser. B., 5 (3), 769–798.

[8] Repin, S., 2008. A Posteriori Estimates for Partial Di erential Equations. Walter de Gruyter, Berlin.

[9] Xu, J., Zhang, Z., 2004. Analysis of recovery type a posteriori error estimators for mildly structured grids. Math. Comp., 73(247), 1139-1152.

[10] Bank, R., Xu, J., 2004. Asymptotically exact a posteriori error estimators, part I: grids with superconvergence. SIAM J. Numer. Anal., 2294-2312.

[11] Zhang, Z., Naga, A., 2005. A new Fnite element gradient recovery method: super- convergence property. SIAM J. Sci. Comput., 26 (4), 1192-1213.

[12] Guo, H., Zhang, Z., Zhao, R., Zou, Q., 2016. Polynomial preserving recovery on boundary. J. Comput. Appl. Math, 119–133,.

[13] Adel, E., Elsaid, A., El-Agamy, M., 2016. Adaptive finite element method for Fred- holm integral equation. South Asian J Math, 6 (5), 239-248.

[14] Sameeh, M., Elsaid, A., 2016. Chebyshev Collocation Method for Parabolic Partial Integrodifferential Equations, Advances in Mathematical Physics, Article ID 7854806, 7 pages.

[15] Wu, H., Zhang, Z., 2007. Can we have super convergent gradient recovery under adaptive meshes. Society for Industrial and Applied Mathematics, 1701–1722.

[16] Huang, Y., Yi, N., 2010. The super convergent cluster recovery method. J. Sci. Comput., 44 (3), 301–322.

[17] Naga, A., Zhang, Z., 2005a. A posteriori error estimates based on the polynomial preserving recovery. SIAM J. Numer. Anal., 1780-1800.

[18] Essam, R., El-Agamy, M., Elsaid, A., 2019. Heat flux recovery in a multilayer model for skin tissues in the presence of a tumor. Eur. Phys. J. Plus, 134, 285