1-1-2017

Polynomial Preserving Recovery For Weak Galerkin Methods And Their Applications

Ren Zhao
Wayne State University,

POLYNOMIAL PRESERVING RECOVERY FOR WEAK GALERKIN METHODS AND THEIR APPLICATIONS

by

REN ZHAO

DISSERTATION

Submitted to the Graduate School

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

2017

MAJOR: MATHEMATICS

Approved By:

Advisor Date

DEDICATION

To My Parents

ACKNOWLEDGEMENTS

First, I would like to sincerely thank my dissertation advisor Professor Zhimin Zhang. Without his inspirational guidance, professional supervision, and extraordinary patience, I could never have finished this dissertation. I am eternally grateful for your kind consideration, support and help.

I would like to thank my former advisor Professor Yinnian He from Xi'an Jiaotong University, who introduced me to my research area – numerical methods on PDE, and whose enthusiasm and dedication have always been encouraging me in my research and study. Thanks to Professor He's invaluable help and support, I had this opportunity to pursue the doctorate degree at Wayne State University.

I would like to thank Prof. Fatih Celiker, Prof. Hengguang Li, Prof. Weisong Shi and Prof. Pei-Yong Wang for serving my dissertation committee. Thank you very much for your precious time and invaluable help in this very important period of my life.

To Dr. Ahmed Naga, who had spent days and nights working with me for two years, your help is very much appreciated and will always be. Thank you very much for inviting me into your gorgeous home and thank you for your patience and valuable time on discussing algorithms, checking codes and advising my reports. It is always a joy to work with such a professional researcher.

To Department of Mathematics at Wayne State University, thank you for awarding me Rumble fellowship in 2009, which enabled me come to the United States and totally changed my life. I wouldn't have gone this far without the department's financial support and generous assistance during my study. I appreciate all the classes, trainings and teaching experiences I received from the department. They are invaluable treasure and will affect me in the rest of my life.

I would like to extend my sincere gratitude and appreciation to my fellow collaborators and friends Prof. Feng Shi, Dr. Hailong Guo, Dr. Li Fan, Prof. Can Huang, Prof. Jiaping Yu, Prof. Haibiao Zheng and Prof. Wenming He for their kind help, valuable discussion, and collaboration. It was beyond a pleasure to collaborate with all of you these past several years.

Finally, I would like to thank my beloved parents for their unconditional love, support, and encouragement for all the past years. I won't be this stronger without both of you as my inspiration. I love you all so much.

TABLE OF CONTENTS

DEDICATION **ii**

ACKNOWLEDGEMENTS **iii**

LIST OF TABLES **vii**

LIST OF FIGURES **ix**

Chapter 1: INTRODUCTION **1**

Chapter 2: BOUNDARY STRATEGIES **8**

2.1 Polynomial Preserving Recovery 9

2.2 PPR on boundary 11

2.3 Numercial Examples 24

2.4 Conclusion remarks 30

Chapter 3: WGFEMS FOR 2ND ORDER ELLIPTIC PROBLEMS **33**

3.1 WG scheme for (P_0, P_0, RT_0) element 34

3.2 WG scheme for (P_1, P_1, P_0) element 36

3.3 Supercloseness Analysis 37

Chapter 4: PPR FOR WGFEMS **41**

4.1 WGPPR for (P_0,P_0,RT_0) element 41

4.2 WGPPR for (P_1,P_1,P_0) element 50

4.3 Property of the Gradient Recovery Operator 58

4.4 Numerical Examples 59

Chapter 5: WGPPR FOR INTERFACE PROBLEM **63**

5.1 The weak Galerkin scheme 64

5.2 WGPPR for Interface Problems 66

5.3 Numerical Examples 67

Chapter 6: APPLICATIONS OF WGPPR **88**

6.1 Application to adaptive methods 88

6.2 3D Problem 92

6.3 Stokes Problem 92

REFERENCES **98**

ABSTRACT **112**

AUTOBIOGRAPHICAL STATEMENT **114**

LIST OF TABLES

Table 2.3.1 Example 1: $\|\cdot\|_{L_\infty(\mathcal{N}_{h,1})}$ on Delaunay Triangulation 25

Table 2.3.2 Example 1: $\|\cdot\|_{L_\infty(\mathcal{N}_{h,2})}$ on Delaunay Triangulation 26

Table 2.3.3 Example 1: $\|\cdot\|_{L_2(\Omega_{h,1})}$ on Delaunay Triangulation 26

Table 2.3.4 Example 1: $\|\cdot\|_{L_2(\Omega_{h,2})}$ on Delaunay Triangulation 26

Table 2.3.5 Example 2: $\|\cdot\|_{L_\infty(\mathcal{N}_{h,1})}$ on Delaunay Triangluation 27

Table 2.3.6 Example 2: $\|\cdot\|_{L_\infty(\mathcal{N}_{h,2})}$ on Delaunay Triangulation 28

Table 2.3.7 Example 2: $\|\cdot\|_{L_2(\Omega_{h,1})}$ on Delaunay Triangulation 28

Table 2.3.8 Example 2: $\|\cdot\|_{L_2(\Omega_{h,2})}$ on Delaunay Triangulation 28

Table 2.3.9 Example 3: $\|\cdot\|_{L_\infty(\mathcal{N}_{h,1})}$ on Delaunay Triangulation 29

Table 2.3.10 Example 3: $\|\cdot\|_{L_\infty(\mathcal{N}_{h,2})}$ on Delaunay Triangulation 29

Table 2.3.11 Example 3: $\|\cdot\|_{L_2(\Omega_{h,1})}$ on Delaunay Triangulation 29

Table 2.3.12 Example 3: $\|\cdot\|_{L_2(\Omega_{h,2})}$ on Delaunay Triangulation 30

Table 4.4.1 Poisson Problem: WG using $(P_1(T), P_1(e), P_0(T))$ 62

Table 4.4.2 Poisson Problem: Interior error vs Boundary error 62

Table 5.3.1 Example 1: Comparison of H^1 error of Gradient recovery . . . 69

Table 5.3.2 Example 2: Comparison of H^1 error of Gradient recovery . . . 72

Table 5.3.3 Example 3: Comparison of H^1 error of Gradient recovery . . . 73

Table 5.3.4 Example 4: Comparison of H^1 error of Gradient recovery . . . 74

Table 5.3.5 Example 5: Comparison of H^1 error of Gradient recovery, $\omega = 5$ 76

Table 5.3.6 Example 5: Comparison of H^1 error of Gradient recovery, $\omega = 10$ 76

Table 5.3.7 Example 5: Comparison of H^1 error of Gradient recovery, $\omega = 20$ 76

Table 6.1.1 Error table of L-shape problem 90

Table 6.1.2 Error table of Crack problem 91

Table 6.2.1 Poisson 3D 92

Table 6.3.1 Stokes Problem: WG using $(P_0(T), P_0(e), RT_0(T))$ 94

Table 6.3.2 Stokes Problem: WG using $(P_0(T), P_0(e), RT_0(T))$ 94

Table 6.3.3 Stokes Problem: WG using $(P_1(T), P_1(e), P_0(T))$ 95

Table 6.3.4 Stokes Problem: WG using $(P_1(T), P_1(e), P_0(T))$ 95

Table 6.3.5 Stokes Problem: Interior error vs Boundary error 95

LIST OF FIGURES

Figure 2.2.1 Examples for patch used in Strategy 1 13

Figure 2.2.2 Examples for patch used in Strategy 2 15

Figure 2.2.3 Patch of isolated corner vertex in strategy 1 16

Figure 2.2.4 Patch of isolated corner vertex in strategy 2 16

Figure 2.2.5 Denominator $42h$ 19

Figure 2.2.6 Denominator $6h$ 19

Figure 2.2.7 Denominator $12h$ 20

Figure 2.2.8 Denominator $140h$ 21

Figure 2.2.9 Denominator $36h$ 22

Figure 2.2.10 Denominator $60h$ 23

Figure 2.3.1 Initial mesh for Example 4 31

Figure 2.3.2 Comparison of decay of error among different strategies 31

Figure 4.1.1 regular pattern 42

Figure 4.1.2 Regular pattern: (Left) $G_h^x : \frac{1}{16h}$. (Right) $G_h^y : \frac{1}{16h}$. 45

Figure 4.1.3 Chevron pattern: (Left) $G_h^x : \frac{1}{25390h}$. (Right) $G_h^y : \frac{1}{4979980h}$. 47

Figure 4.1.4 Unionjack pattern: (Left) $G_h^x : \frac{1}{157h}$. (Right) $G_h^y : \frac{1}{157h}$. 48

Figure 4.1.5 Criss-cross pattern:(Left) $G_h^x : \frac{3}{70h}$. (Right) $G_h^y : \frac{3}{70h}$. 49

Figure 4.2.1 regular pattern 51

Figure 4.2.2 chevron pattern 51

Figure 4.2.3 unionjack pattern 52

Figure 4.2.4 crisscross pattern . . . 52

Figure 4.4.1 Regular mesh: $||\nabla u - G_h u_h||_{L_2(\Omega)}$ and $||\nabla u - \nabla_w u_h||_{L_2(\Omega)}$ . . . 60

Figure 4.4.2 Chevron mesh: $||\nabla u - G_h u_h||_{L_2(\Omega)}$ and $||\nabla u - \nabla_w u_h||_{L_2(\Omega)}$ . . . 60

Figure 4.4.3 Unionjack mesh: $||\nabla u - G_h u_h||_{L_2(\Omega)}$ and $||\nabla u - \nabla_w u_h||_{L_2(\Omega)}$ . . 61

Figure 4.4.4 Criss-cross mesh: $||\nabla u - G_h u_h||_{L_2(\Omega)}$ and $||\nabla u - \nabla_w u_h||_{L_2(\Omega)}$ . . 61

Figure 5.3.1 Example 1. (a) Shape of interface; (b) Body fitted initial mesh. 69

Figure 5.3.2 WG solution u_h . . . 70

Figure 5.3.3 Recovered solution $R_h u_h$ . . . 70

Figure 5.3.4 Graph of $G_h^x(u_h)$ at Level 2 . . . 70

Figure 5.3.5 Graph of $G_h^y(u_h)$ at Level 2 . . . 70

Figure 5.3.6 Example 2. (a) Shape of interface; (b) Body fitted initial mesh. 71

Figure 5.3.7 WG solution u_h . . . 77

Figure 5.3.8 Recovered solution $R_h u_h$ . . . 77

Figure 5.3.9 Example 2: $G_h^x(u_h)$ . . . 77

Figure 5.3.10 Example 2: $G_h^y(u_h)$ . . . 77

Figure 5.3.11 Example 3. (a) Shape of interface; (b) Body fitted initial mesh. 78

Figure 5.3.12 WG solution u_h . . . 79

Figure 5.3.13 Recovered solution $R_h u_h$ . . . 79

Figure 5.3.14 Example 3: $G_h^x(u_h)$ . . . 79

Figure 5.3.15 Example 3: $G_h^y(u_h)$ . . . 79

Figure 5.3.16 Example 4. (a) Shape of interface; (b) Body fitted initial mesh. 80

Figure 5.3.17 WG solution u_h . . . 81

Figure 5.3.18 Recovered solution $R_h u_h$ 81

Figure 5.3.19 Example 4: $G_h^x(u_h)$ 81

Figure 5.3.20 Example 4: $G_h^y(u_h)$ 81

Figure 5.3.21 $\omega = 5$ (a) Shape of interface; (b) Body fitted initial mesh. . . . 82

Figure 5.3.22 WG solution u_h 83

Figure 5.3.23 Recovered solution $R_h u_h$ 83

Figure 5.3.24 $G_h^x(u_h)$ when $\omega = 5$ 83

Figure 5.3.25 $G_h^y(u_h)$ when $\omega = 5$ 83

Figure 5.3.26 $\omega = 10$ (a) Shape of interface; (b) Body fitted initial mesh. . . 84

Figure 5.3.27 WG solution u_h 85

Figure 5.3.28 Recovered solution $R_h u_h$ 85

Figure 5.3.29 $G_h^x(u_h)$ when $\omega = 10$ 85

Figure 5.3.30 $G_h^y(u_h)$ when $\omega = 10$ 85

Figure 5.3.31 $\omega = 20$ (a) Shape of interface; (b) Body fitted initial mesh. . . 86

Figure 5.3.32 WG solution u_h 87

Figure 5.3.33 Recovered solution $R_h u_h$ 87

Figure 5.3.34 $G_h^x(u_h)$ when $\omega = 20$ 87

Figure 5.3.35 $G_h^y(u_h)$ when $\omega = 20$ 87

Figure 6.1.1 L-shape problem: Initial mesh 96

Figure 6.1.2 Adaptive mesh 96

Figure 6.1.3 Decay of recovery error 96

Figure 6.1.4 efficient index κ 96

Figure 6.1.5 Crack Problem: Initial mesh 97

Figure 6.1.6 Adaptive mesh 97

Figure 6.1.7 Decay of gradient error 97

Figure 6.1.8 efficient index κ 97

CHAPTER 1 INTRODUCTION

Gradient recovery [13, 16, 18, 26, 29, 30, 31, 32, 33, 34] is an effective and widely used post-processing technique in scientific and engineering computation. The main purpose of this technique is to reconstruct a better numerical gradient from a finite element solution. It can be used for mesh smoothing, a posteriori error estimate [18, 31, 32, 34, 29, 38], and adaptive finite element method [35] even with anisotropic meshes [44, 15, 17, 25].

An efficient gradient recovery technique must be fast, easy to implement, and accurate in approximating the exact gradient. Some popular post-processing techniques include the celebrated Zienkiewicz-Zhu superconvergent patch recovery (SPR) [33], polynomial preserving recovery (PPR)[30, 18] and edge based recovery [26], which were proposed to obtain accurate gradients with reasonable cost. The SPR recovers the gradient at vertices by local least-squares fitting to the finite element gradient in an associated patch, while the PPR recovers the gradient at a vertex by local least-squares fitting to the finite element solution in an associated patch and then taking the gradient of the least-squares fitted polynomial. The Superconvergent Patch Recovery (SPR) and Polynomial Preserving Recovery (PPR) are two popular methods which have been adopted by commercial software such as ANSYS, Abaqus, COMSOL Multiphysics [7], Diffpack, LS-DYNA, etc.

More recently, the gradient recovery technique was applied to improve the accuracy of eigenvalue approximation. In [27], Shen and Zhou introduced a defect correction scheme based on averaging recovery, like a global L^2 projection and a Clément-type operator. In [21], Naga, Zhang and Zhou used Polynomial Preserving Recovery in eigenvalue approximation and superconvergence results is achieved. In [22], Wu and Zhang further showed polynomial preserving recovery can also enhance eigenvalue approximation on adaptive meshes. The idea was further studied in [23, 24]. Later in [20], Naga and Zhang introduced the function recovery technique and applied it on eigenvalue approximation. In our recent work [14], we propose some fast and efficient solvers for elliptic eigenvalue problems. Our first algorithm is a combination of the shifted-inverse power based two-grid scheme [39, 40] and polynomial recovery enhancing technique [21]. The second algorithm can be viewed as a combination of the two-grid scheme [39, 40] and the two-space method [42, 41]. Both of our methods inherit the superconvergence property of the previous methods but have much lower computational cost.

Post-processing for second order derivatives, which are related to physical quantities such as momentum and Hessian, are also desirable. In adaptive mesh design, Hessian matrix can indicate the direction where the function changes the most and hence it could be used to construct anisotropic meshes to cope with the anisotropic properties of the solution of the underlying PDEs [43, 44]. It is also widely employed in FEM approximation of second order nonvariational elliptic problems [46] and nonlinear equations like Monge-Ampère equation [47, 48]. Moreover, it is used in designing a nonlocal finite element technique [45] as well. In our recent work [36], an effective Hessian recovery method is proposed, both theoretical analysis and numerical experiment have validated the superconvergence property of our algorithm. Our work is not targeted in the direction of adaptive mesh refinement; instead, our emphasis is to obtain accurate Hessian matrices via recovery techniques. This idea is natural: apply PPR twice to the primarily computed data. However, the mathematical theory behind it is non-trivial and quite involved, especially in the ultraconvergence analysis of the recovered Hessian. A direct calculation of the gradient from the linear finite element space has linear convergent rate and the Hessian has no convergence at all. Our Hessian recovery method can achieve second order convergence under some uniform meshes, which is a very surprising result!

The PPR often forms a higher-order approximation of the gradient on a patch of mesh elements around each mesh vertex. For regular meshes, the convergence rate of the recovered gradient is $O(h^{p+1})$-the same as for the solution itself [5, p.471] [6, p.1061]. However, the accuracy of PPR near boundaries is not as good as that away from the boundaries. It might even be worse than without recovery. [5, p.471][6, p.1061]. Some special treatments are needed to improve the accuracy of PPR on the boundary. In this thesis, we present two boundary recovery strategies to resolve the problem caused by boundaries. Our first strategy to recover the gradient at a boundary vertex is as follows. First, by using the standard PPR local least-squares fitting procedure for interior vertex, we construct a polynomial for each selected interior vertices close to the target boundary vertex. Then we take the average of all quantities evaluating the gradient of the obtained polynomials at the target boundary vertex as the recovered gradient. The second recovery strategy is as below: We construct a relatively large element patch by merging all the element patches of some selected interior vertices near the target point.

Then we select all mesh nodes in the above patch as sampling points to fit a polynomial in least-squares sense and define the recovered gradient by the gradient of the constructed polynomial at the target point. The basic idea behind our two strategies is: the classic PPR method cannot achieve a good approximation on boundary comparable to that in the interior of the domain since the classic selected boundary patch does not contain sufficient information. Therefore, we should replace the boundary patch by the interior patches which has more information than the boundary patch and which has a certain symmetric property. Both the above proposed methods use more information than the classic PPR methods. Our two methods are numerically tested and compared with standard implementation in COMSOL Multiphysics. The numerical results in L_2 norm validate that both our methods lead to superconvergent recovered gradient up to boundary. The numerical errors in L_∞ norm show improved accuracy over the classical PPR method near boundary.

The weak Galerkin finite element methods is a novel numerical method that was first proposed and analyzed by Wang and Ye in [118] for general second order elliptic problems on simplicial grids, and later on in [101, 98, 108] for shape regular polytopal meshes. The main idea of weak Galerkin finite element methods is the use of weak functions where the differential operators, such as gradient, divergence, curl, Laplacian, are approximated by their weak forms as distributions. Different algorithms and improvement have been made for solving second order elliptic equations [68, 80, 85, 117, 55]. By replacing the differential operators in the weak form of different problem, the weak Galerkin finite element methods have been successfully applied to various problems. A weak Galerkin method was introduced in [116] for the elliptic interface problems by using a Lagrange multiplier to handle the interface conditions. Later in [69], a new weak Galerkin method has been developed for the same interface problem, the main difference is the use of a parameter free stabilizer term which makes the new WGFEMs more flexible in handling complicated interface geometries. By introducing the weak Laplacian operator, biharmonic equations have been studied in [92, 93, 101, 108, 110, 115, 54]. With the definition of weak divergence and weak curl, the div-curl system is discretised by the WGFEMs in [71]. Under the same concept, the weak Galerkin methods have been further developed to solve many other problems, including Helmholtz equations [56, 111, 88], Maxwell equations [89], Wave equations [60], Stokes equations [67, 81, 83, 87, 99, 57], Brinkman equations [74, 76, 103], Oseen equations [66], Darcy-Stokes equations [78, 102, 58]. Furthermore, fourth order problem have been solved by WGFEMs in [79, 59]. Besides the success in solving different problems with WGFEMs, there are also a lot of modified versions of WGFEMs to serve different purpose. In [106], Gao and Wang proposed a modified WGFEMs for a class of parabolic problems. In [90], Gao et. al. proposed a modified WGFEMs for convection-diffusion problems in 2D. The Sobolev equation has been studied by Gao and Wang in [94], using a modified weak Galerkin finite element method as well. Mu and her group introduced the modified WGFEMs for the Stokes equations in [99]. The advantage of these modified weak finite element method is its lower global degrees of freedom. Most of these work on weak Galerkin finite element methods concern only a priori error estimates for the corresponding numerical solutions. The superconvergence of weak Galerkin finite element method is still lack of attention. Recently, Chen et. al. [109] presented a residual type a posteriori error estimator and analyzed its convergence property. This is the first article concerning about the a posteriori error estimation and the adaptivity of weak Galerkin method. In [64], Zhang et. al. presented an a posteriori error estimator for the modified weak Galerkin finite element methods.

Due to its problem independent and method independent feature, PPR can be generalized to finite volume methods, finite difference methods and non-conforming finite element methods. In this thesis, we will apply PPR on the information generated by different weak Galerkin scheme and denote it by WGPPR. Detailed framework of WGPPR and several numerical experiments will be provided to show this process. Boundary recovery technique can be used at an interface, where the solution or its gradient has jumps. In other words, we treat an interface (if the location is known a priori) as a boundary when performing gradient recovery or function recovery. In this thesis, we will present the detail on applying WGPPR to interface problems and this is the first appearance of this approach. Furthermore, since we have seen success in applying PPR to adaptive methods for standard Galerkin methods, it is natural for us to apply the same idea to the adaptivity of weak Galerkin method. In addition, WGFEMs for Stokes problem is considered as well. WGPPR is employed to recover the gradient information and superconvergent phenomenon is again observed.

The rest of this dissertation is organized as follows:

Chapter 2 introduces the polynomial preserving recovery technique with its boundary strategies. We present two strategies to improve the performance of PPR gradient recovery on the boundary. Several numerical experiments are provided to validate our methods. This chapter is based on our published paper [37].

Chapter 3 is devoted to the weak Galerkin methods on second order elliptic problem. Two different schemes of weak Galerkin are defined, and the supercloseness property is analyzed.

Chapter 4 is about the gradient recovery technique for the weak Galerkin methods: WGPPR. We gather the information obtained from WGFEMs solution in different schemes and perform the polynomial preserving recovery process. Superconvergence phenomenon are observed from numerical tests which verify the superconvergence property of our proposed algorithm.

Chapter 5 will focus on interface problem and WGPPR will be applied to perform gradient recovery and function recovery. Numerical experiments are performed to prove the superconvergence property of the proposed method.

Chapter 6 studies the adaptive method for weak Galerkin which uses the recovery type posteriori error estimator based on WGPPR. Furthermore, our proposed recovery algorithm is applied to different problems, including 3D Poisson problem and Stokes problem using WGFEMs.

CHAPTER 2 BOUNDARY STRATEGIES

Consider the following model second order elliptic problem

$$\begin{cases} -\Delta u = f, & \text{in } \Omega, \\ \\ u = g, & \text{on } \partial\Omega; \end{cases} \tag{2.0.1}$$

where Ω is a bounded polygonal domain with Lipschitz boundary $\partial\Omega$ in $\mathbb{R}^2$. In this thesis, we adopt the standard notations for Sobolev space and their associate norms [4].

A multi-index α is a 2-tuple of non-negative integers $\alpha_i, i = 1, 2$ with length $|\alpha| = \sum_{i=1}^{2} \alpha_i$. Define the weak partial derivative $D^\alpha v = (\frac{\partial}{\partial x})^{\alpha_1}(\frac{\partial}{\partial y})^{\alpha_2}$ [1, 2, 4] and denote $D^k v$ with $|\alpha| = k$ the vector of all partial derivatives of order r. $W_p^k(\Omega)$ denotes the Sobolev space $W_p^k(\Omega) = \{v : D^\alpha v \in L^p(\Omega), |\alpha| \le k\}$ equipped with the norm

$$\|v\|_{k,p,\Omega} = (\sum_{|\alpha|\le k} \int_\Omega |D^\alpha v(z)|^p dz)^{\frac{1}{p}},\ 1 \le p < \infty,$$

$$\|v\|_{k,\infty,\Omega} = \text{ess} \sup_{|\alpha|\le k, z\in\Omega} |D^\alpha v(z)|,\ p = \infty;$$

and seminorm

$$|v|_{k,p,\Omega} = (\sum_{|\alpha|=k} \int_\Omega |D^\alpha v(z)|^p dz)^{\frac{1}{p}},\ 1 \le p < \infty,$$

$$|v|_{k,\infty,\Omega} = \text{ess} \sup_{|\alpha|=k, z\in\Omega} |D^\alpha v(z)|,\ p = \infty.$$

When $p = 2$, we denote simply $H^k(\Omega) = W_2^k(\Omega)$. The space $H(\text{div}; \Omega)$ is defined as the set of vector-valued functions on Ω which, together with their divergence, are square integrable, i.e.,

$$H(\text{div}; \Omega) = \{v : v \in [L^2(\Omega)]^2, \nabla \cdot u \in L^2(\Omega)\}.$$

The norm in $H(\text{div}; \Omega)$ is defined by $\|v\|_{H(\text{div};\Omega)} = (\|v\|^2 + \|\nabla \cdot v\|^2)^{\frac{1}{2}}$. For any $0 < h < \frac{1}{2}$, let $\mathcal{T}_h$ be a shape regular triangulation of $\overline{\Omega}$ with mesh size at most h, i.e.

$$\overline{\Omega} = \bigcup_{T \in \mathcal{T}_h} T,$$

where T is a triangle. For any positive integer r, define the continuous finite element space S_h of order r as

$$S_h = \{v \in C(\overline{\Omega}) : v|_T \in \mathbb{P}_r(T), \quad \forall T \in \mathcal{T}_h\} \subset H^1(\Omega),$$

where $\mathbb{P}_r$ denote the space of polynomial defined on T with degree less than or equal to r. Denote the finite element solution in S_h by u_h, and the set of mesh nodes and interior mesh nodes by $\mathcal{N}_h$ and $\mathring{\mathcal{N}}_h$, respectively.

The standard Lagrange basis of S_h is denoted by $\{\phi_z : z \in \mathcal{N}_h\}$ with $\phi_z(z') = \delta_{zz'}$ for all $z, z' \in \mathcal{N}_h$. For any continuous function u, let $I_h u$ be the standard interpolation of u into the finite element space S_h, i.e. $I_h u = \sum_{z \in \mathcal{N}_h} u(z)\phi_z$.

Throughout this thesis, we denote $u \leq Cv$ by $u \lesssim v$ where the letter C or c denotes a constant which is independent of h and may not necessarily be the same at each occurrence.

2.1 Polynomial Preserving Recovery

In this section, we will give a brief introduction to the polynomial preserving recovery method. For the sake of clarity, only C^0 finite element methods will be considered.

Given a vertex $z \in \mathcal{N}_h$, let $\mathcal{L}(z, n)$ denote the union of mesh elements in the first n layers around z, i.e.,

$$\mathcal{L}(z,n) = \begin{cases} z, & \text{if } n = 0, \\ \bigcup\{\tau : \tau \in \mathcal{T}_h,\ \tau \cap \mathcal{L}(z,0) \neq \phi\}, & \text{if } n = 1, \\ \bigcup\{\tau : \tau \in \mathcal{T}_h,\ \tau \cap \mathcal{L}(z,n-1) \text{ is a } (d-1)\text{-simplex}\}, & \text{if } n \geq 2. \end{cases} \tag{2.1.1}$$

An element patch $\mathcal{K}_z$ around an interior vertex z is defined based on $\mathcal{L}(z, n)$, which contains n_z nodes. For details construction of $\mathcal{K}_z$, readers are referred to [30, 19]. We select all mesh nodes $z_j \in \mathcal{N}_h$, $j = 1, 2, \cdots, n_z$ in this element patch $\mathcal{K}_z$ as sampling points, and fit a polynomial of degree $r + 1$ in the least-squares sense, i.e., we seek $p_z \in \mathbb{P}_{r+1}(\mathcal{K}_z)$ such that

$$\sum_{j=1}^{n_z} (p_z - u_h)^2(z_j) = \min_{q \in \mathbb{P}_{r+1}} \sum_{j=1}^{n_z} (q - u_h)^2(z_j). \tag{2.1.2}$$

The recovered gradient at node z is then defined as

$$(G_h u_h)(z) := \nabla p_z(z). \tag{2.1.3}$$

If $r = 1$, all sampling points are vertices and $G_h u_h$ is completely defined. If $r > 1$, sampling points may contain the following two types of node:

- Edge node: if z lies on an edge e formed by two vertices z_1 and z_2, we define

  $$(G_h u_h)(z) = \lambda \nabla p_{z_1}(z) + (1 - \lambda) \nabla p_{z_2}(z) \tag{2.1.4}$$

  where λ is determined by the ratio of distances of z to z_1 and z_2.

- Interior node: if z lies in the interior of a triangle T formed by three vertices z_1, z_2, and z_3, we define

  $$(G_h u_h)(z) = \sum_{j=1}^{3} \lambda_j \nabla p_{z_j}(z), \tag{2.1.5}$$

  where λ_j is the barycentric coordinate of z.

With all nodal values of $G_h u_h$ determined, the gradient recovery operator: $G_h : S_h \to S_h^d$ is then well defined.

It was proved in [30] that the least-squares fitting procedure has a unique solution under certain geometric conditions. As for linear element, we need at least six nodes to fit a quadratic polynomial and those sampling points should not be on a conic curve. In addition, the gradient recovery operator G_h has the following properties [19, 30]:

1. G_h is a bounded operator in the sense that there exists a constant C, independent of h, such that $||G_h v||_{L_2(\Omega)} \leq C|v|_{H^1(\Omega)}, \quad \forall v \in S_h$.

2. For any nodal point z, if $p \in P_{r+1}(\mathcal{K}_z)$, $(G_h p)(z) = \nabla p(z)$.

Furthermore, the following superconvergence results hold [30].

Theorem 2.1.1. *Let $\mathcal{T}_h$ be an arbitrary mesh. Then, G_h preserves polynomials of degrees up to $r+1$ in Ω. Furthermore, if the nodes involved in PPR at a mesh vertex $z \in \mathcal{N}_h$ are symmetrically distributed around z, and if r is even, then G_h preserves polynomials of degree up to $r+2$ at z.*

Theorem 2.1.2. *Let z be a mesh node and $\mathcal{K}_z$ be the corresponding patch. If $u \in W_\infty^{r+2}(\mathcal{K}_z)$, then*

$$||\nabla u - G_h u||_{L_\infty(\mathcal{K}_z)} \leq Ch^{r+1}|u|_{W_\infty^{r+2}(\omega_z)},$$

where ω_z is a larger element patch which contains $\mathcal{K}_z$.

2.2 PPR on boundary

If not handled properly, gradient recovery techniques may deteriorate near boundary [30, 19]. High performance near/on boundary is one of the key characteristics of a good gradient recovery technique. In this section, we present two systematic strategies to construct robust PPR operator up to boundary. Both strategies have comparable accuracy near boundary $\partial\Omega$ as in the interior of Ω. Only linear element is considered here. Extension to higher-order elements can be done by combining ideas in this work with PPR for higher-order cases. In the sequel, we denote z as a mesh vertex on boundary, i.e., $z \in \mathcal{N}_h \cap \partial\Omega$.

2.2.1 Strategy 1

Simple averaging of the recovered gradient from PPR under uniform triangular mesh of the regular pattern produces ultra-convergence (two orders higher) gradient recovery for quadratic element at element edge centers [30]. In light of this fact, our first strategy is to treat $z \in \mathcal{N}_h \cap \partial\Omega$ similarly as an edge center in quadratic element.

For any boundary vertex z, define

$$\mathcal{K}_z = \mathcal{L}(z, n_0), \tag{2.2.1}$$

where n_0 is the smallest integer such that $\mathcal{L}(z, n_0)$ contains at least one interior vertex.

Let $z_0, z_1, \ldots, z_{n_z}$ be all the interior vertices in $\mathcal{K}_z$. Then our recovered gradient at z is defined as

$$(G_h u_h)(z) = \frac{1}{n_z + 1} \sum_{j=0}^{n_z} \nabla p_{z_j}(z), \tag{2.2.2}$$

where p_{z_j} is the polynomial that fits u_h at the interior vertex z_j in $\mathcal{K}_{z_j}$, a well defined element patch according to [30].

To describe how to construct $\mathcal{K}_z$, consider a typical Delaunay unstructured mesh on rectangle $[0, 2] \times [0, 1]$ which is obtained using **Triangle** [28], see Fig 2.2.1. Boundary vertices can be grouped into those connecting with one interior vertex, two interior vertices, three interior vertices, and so on. It is worth to mention that the first group usually contains only corner vertices. Fig 2.2.1 depicts three types of boundary vertices and their corresponding patches.

(1) The left upper corner z is contained in two elements that share the same interior vertex z_0. According to definition , $\mathcal{K}_z$ is the element patch which consists of two triangles. We then define $(G_h u_h)(z) = \nabla p_{z_0}(z)$.

(2) The bottom z is contained in three elements that have two interior vertices z_0 and z_1. According to definition $\mathcal{K}_z$ is the element patch which consists of three

triangles. We then define $(G_h u_h)(z) = \frac{1}{2}(\nabla p_{z_0}(z) + \nabla p_{z_1}(z))$.

(3) The upper z is contained in four elements that have three interior vertices z_0, z_1, and z_2. According to definition , $\mathcal{K}_z$ is the element patch which consists of four triangles. The recovered gradient at z is then defined as $(G_h u_h)(z) = \frac{1}{3}(\nabla p_{z_0}(z) + \nabla p_{z_1}(z) + \nabla p_{z_2}(z))$.

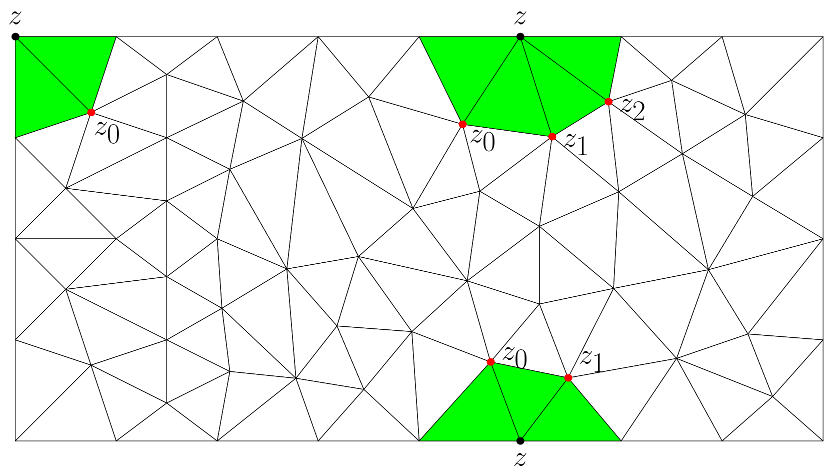

Figure 2.2.1: Examples for patch used in Strategy 1

2.2.2 Strategy 2

Here we treat z just like an interior vertex. However, the definition of $\mathcal{K}_z$ is more delicate and deserves special consideration. $\mathcal{K}_z$ is constructed in two steps. In the first step, we define a temporary patch $\tilde{\mathcal{K}}_z$ as $\mathcal{K}_z$ in (2.2.1). After constructing the temporary patch $\tilde{\mathcal{K}}_z$, we define

$$\mathcal{K}_z = \left(\bigcup_{\tilde{z} \in \tilde{\mathcal{K}}_z \cap \mathring{\mathcal{N}}_h} \mathcal{K}_{\tilde{z}} \right) \bigcup \left(\bigcup_{\tilde{z} \in \tilde{\mathcal{K}}_z \cap \mathcal{N}_h \cap \partial\Omega} \mathcal{L}(\tilde{z}, 1) \right), \tag{2.2.3}$$

where $\mathcal{K}_{\tilde{z}}$ is defined in reference [30] for $\tilde{z} \in \mathring{\mathcal{N}}_h$. Note that we distinguish between interior vertices and boundary vertices in the temporary patch $\tilde{\mathcal{K}}_z$. For a boundary vertex z', only triangles having z' as a vertex is added to $\mathcal{K}_z$; but for an interior vertex z'', its own patch $\mathcal{K}_{z''}$ is adding to $\mathcal{K}_z$. Let $p_z \in \mathbb{P}_2(\mathcal{K}_z)$ be the polynomial that best fits u_h at the mesh nodes in K_z in discrete least squares sense, i.e.,

$$p_z = \arg \min_{p \in \mathbb{P}_2(\mathcal{K}_z)} \sum_{\tilde{z} \in \mathcal{N}_h \cap \mathcal{K}_z} |(u_h - p)(\tilde{z})|^2. \tag{2.2.4}$$

Then define the gradient recovery operator at vertex z as $(G_h u_h)(z) = \nabla p_z(z)$.

To demonstrate the process of constructing $\mathcal{K}_z$ in Strategy 2, we use the same Delaunay mesh as in Strategy 1. All three types boundary vertices are described in previous subsection. Note that we construct $\mathcal{K}_z$ in two steps. Firstly, we construct $\tilde{\mathcal{K}}_z$ which is shown in Fig 2.2.1. Then $\mathcal{K}_z$ can be constructed which is illustrated in Fig 2.2.2.

(1) The left upper corner z is contained in two elements that share the same interior vertex z_0. Therefore, $\tilde{\mathcal{K}}_z$ is the union of $\mathcal{K}_{z_0}$ and the patches corresponding to the other two boundary vertices near z. Hence, the two red triangles are added to $\mathcal{K}_{z_0}$ and this completes the construction of $\tilde{\mathcal{K}}_z$.

(2) The bottom z is contained in three elements that have two interior vertices z_0 and z_1. $\mathcal{K}_z$ is constructed as shown previous in Fig 2.2.1 and $\tilde{\mathcal{K}}_z$ contains $\mathcal{K}_{z_0}$ and $\mathcal{K}_{z_1}$. The union of $\mathcal{K}_{z_0}$ and $\mathcal{K}_{z_1}$ are all green triangles near bottom edge in Fig 2.2.2. For the other two boundary vertices in $\tilde{\mathcal{K}}_z$, we will add triangles containing them into $\mathcal{K}_z$, i.e. the three red triangles near the bottom edge.

(3) Finally, we look at the boundary vertex connecting with 3 interior vertices; see the solid dot point on the top edge in Fig 2.2.2. We first construct $\tilde{\mathcal{K}}_z$ which consists of four triangles having z as a vertex; see Fig 2.2.1 for detail. z_0, z_1 and z_2 are all interior vertices in $\tilde{\mathcal{K}}_z$. According to (2.2.3), $\mathcal{K}_z$ contains $\mathcal{K}_{z_0}$, $\mathcal{K}_{z_1}$ and $\mathcal{K}_{z_2}$. The union of $\mathcal{K}_{z_0}$, $\mathcal{K}_{z_1}$ and $\mathcal{K}_{z_2}$ are all green triangles near the top edge in Fig 2.2.2. For other boundary vertices in $\tilde{\mathcal{K}}_z$, we only add triangles containing them into $\mathcal{K}_z$, i.e. the two red triangles near the top edge. Thus $\mathcal{K}_z$ is the element patch consisting of sixteen triangles.

Remark. Definition of $\mathcal{K}_z$ in (2.2.3) always guarantees the existence and uniqueness

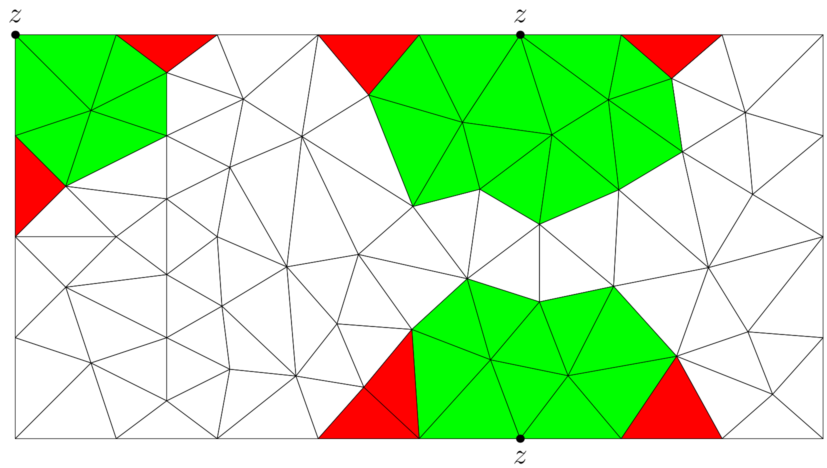

Figure 2.2.2: Examples for patch used in Strategy 2

of p_z. The construction procedure is systematic and works for arbitrary mesh.

Remark. Comparing with the boundary recovery methods proposed in [30], the points involved in our procedure are more symmetric. Hence, this strategy is more stable and robust.

Before ending this subsection, we consider a special situation. For mesh generated by engineering procedure such as Delaunay mesh generator, any vertex connects with at least one interior vertex, i.e. $\mathcal{L}(z,1) \cap \mathring{\mathcal{N}}_h \neq \emptyset$; see Fig 2.2.1 or 2.2.2. But it may occur that $\mathcal{L}(z,1) \cap \mathring{\mathcal{N}}_h = \emptyset$, such as regular and chevon pattern of uniform mesh. Even in this case, both our strategies can be applied without any change. One typical example is shown in Fig 2.2.3 or 2.2.4. For strategy 1, $\mathcal{K}_z$ should be defined as $\mathcal{L}(z,2)$ instead of $\mathcal{L}(z,1)$. In other words, $\mathcal{K}_z$ are two green triangles in Fig 2.2.3. Then the recovered gradient at z is defined as $(G_h u_h)(z) = \nabla p_{z_0}(z)$. In order to define $\mathcal{K}_z$ in strategy 2, we first construct $\tilde{\mathcal{K}}_z$ containing one interior vertex z_0; see the second sub-figure of Fig 2.2.4. According to (2.2.3), $\mathcal{K}_z$ contains K_{z_0}, i.e. all green triangles in the third sub-figure of Fig 2.2.4. Similarly, all triangles containing z' are added to $\mathcal{K}_z$ for each boundary vertex z' in $\tilde{\mathcal{K}}_z$.

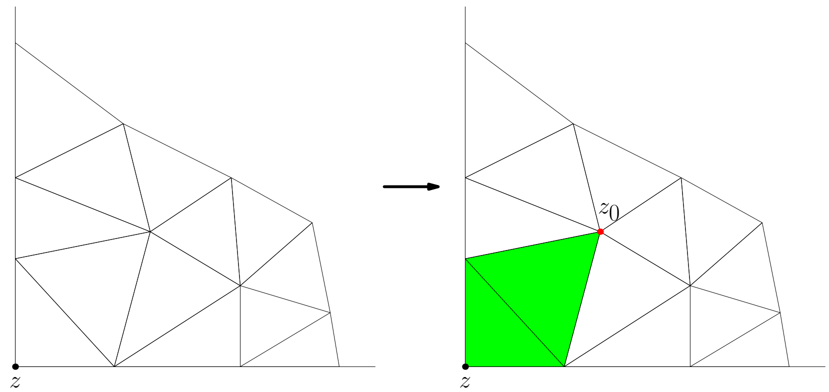

Figure 2.2.3: Patch of isolated corner vertex in strategy 1

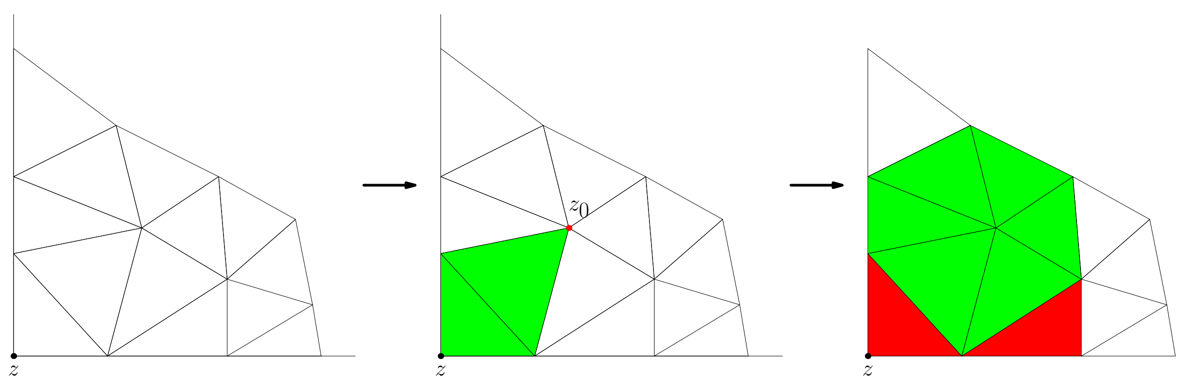

Figure 2.2.4: Patch of isolated corner vertex in strategy 2

2.2.3 Some illustrations

In this subsection, we use three examples of uniform mesh to demonstrate superconvergence and robustness of our two gradient recovery strategies on boundary. Let G_h^1 and G_h^2 denote boundary recovery operator defined by Strategy 1 and Strategy 2, respectively.

Example 1. We consider a typical corner vertex in regular pattern, see the solid dot point in Fig 2.2.6. In this case, the corner point belongs to only one element, to which there is no interior vertex attached. According to strategy 1, we fit a quadratic polynomial $p_{\tilde{z}}(x,y)$ at $\tilde{z}$ instead of fitting a quadratic polynomial of $p_z(x,y)$ at z, where $\tilde{z}$ is the closest interior vertex to z, i.e. the solid dot point in Fig 2.2.5. Note that Fig 2.2.5 shows the patch of the interior vertex $\tilde{z}$ instead of z. Applying the least squares fitting procedure described in [30], we obtain

$$
\begin{aligned}
p_{\tilde{z}}(x,y) =& u_0 + \frac{1}{6h}(2u_1 + u_2 - u_3 - 2u_4 - u_5 + u_6)x \\
&+ \frac{1}{6h}(-u_1 + u_2 + 2u_3 + u_4 - u_5 - 2u_6)y \\
&+ \frac{1}{6h^2}(-6u_0 + 3u_1 + 3u_4)x^2 + \frac{1}{6h^2}(-6u_0 + 3u_3 + 3u_6)y^2 \\
&+ \frac{1}{6h^2}(6u_0 - 3u_1 + 3u_2 - 3u_3 - 3u_4 + 3u_5 - 3u_6)xy.
\end{aligned}
$$

Differentiating with respect to x and y, we get

$$
\begin{aligned}
\frac{\partial p_{\tilde{z}}}{\partial x} =& \frac{1}{6h}(2u_1 + u_2 - u_3 - 2u_4 - u_5 + u_6) + \frac{1}{3h^2}(-6u_0 + 3u_1 + 3u_4)x + \\
&+ \frac{1}{6h^2}(6u_0 - 3u_1 + 3u_2 - 3u_3 - 3u_4 + 3u_5 - 3u_6)y; \\
\frac{\partial p_{\tilde{z}}}{\partial y} =& \frac{1}{6h}(-u_1 + u_2 + 2u_3 + u_4 - u_5 - 2u_6) + \frac{1}{3h^2}(-6u_0 + 3u_3 + 3u_6)y \\
&+ \frac{1}{6h^2}(6u_0 - 3u_1 + 3u_2 - 3u_3 - 3u_4 + 3u_5 - 3u_6)x.
\end{aligned}
$$

Evaluating $\frac{\partial p_{\tilde{z}}}{\partial x}$ and $\frac{\partial p_{\tilde{z}}}{\partial y}$ at z yields

$$G_h^1 u(z) = \frac{1}{6h} \begin{pmatrix} -18u_0 + 11u_1 - 2u_2 + 2u_3 + 7u_4 - 4u_5 + 4u_6 \\ \\ 18u_0 - 4u_1 + 4u_2 - 7u_3 - 2u_4 + 2u_5 - 11u_6 \end{pmatrix}, \tag{2.2.5}$$

as depicted in Fig 2.2.5. Using **Mathematica**, we can easily calculate the Taylor expansion:

$$G_h^1 u(z) = \begin{pmatrix} u_x(z) - \frac{h^2}{6}(2u_{xxx}(z) - 7u_{xxy}(z) + 2u_{xyy}(z)) + O(h^3) \\ \\ u_y(z) - \frac{h^2}{6}(2u_{xxy}(z) - 7u_{xyy}(z) + 2u_{yyy}(z)) + O(h^3) \end{pmatrix}, \tag{2.2.6}$$

which is a second order finite difference scheme approximating $\nabla u(z)$.

Now we turn to Strategy 2. It fits a quadratic polynomial

$$\hat{p}_z(\xi, \eta) = (1, \xi, \eta, \xi^2, \xi\eta, \eta^2)(\hat{a}_1, \ldots, \hat{a}_6)^T,$$

in the least-squares sense at z, see the solid dot point in Fig 2.2.6, with respect to eight nodal values in (ξ, η) coordinates

$$\vec{\xi} = (0, 0, 0, -1, -2, -1, -1, -2)^T, \quad \vec{\eta} = (0, 1, 2, 1, 0, 0, 2, 1).$$

We obtain

$$\begin{aligned} p_z(x, y) = &\frac{1}{42}(38u_0 + 6u_1 - 2u_2 - 8u_3 - 2u_4 + 6u_5 + 2u_6 + 2u_7) \\ &\frac{1}{42h}(44u_0 + 11u_1 + 8u_2 - 38u_3 - 6u_4 - 38u_5 - 8u_6 + 27u_7)x \\ &\frac{1}{42h}(-44u_0 + 38u_1 + 6u_2 + 38u_3 - 8u_4 - 11u_5 - 27u_6 + 8u_7)y \\ &\frac{1}{42h^2}(12u_0 + 3u_1 + 6u_2 - 18u_3 + 6u_4 - 18u_5 - 6u_6 + 15u_7)x^2 \\ &\frac{1}{42h^2}(-18u_0 + 6u_1 + 12u_2 + 6u_3 + 12u_4 + 6u_5 - 12u_6 - 12u_7)xy \\ &\frac{1}{42h^2}(12u_0 - 18u_1 + 6u_2 - 18u_3 + 6u_4 + 3u_5 + 15u_6 - 6u_7)y^2. \end{aligned}$$

It indicates that

$$
\begin{aligned}
\frac{\partial p_z}{\partial x} =& \frac{1}{42h}(44u_0 + 11u_1 + 8u_2 - 38u_3 - 6u_4 - 38u_5 - 8u_6 + 27u_7) \\
& \frac{1}{21h^2}(12u0 + 3u_1 + 6u_2 - 18u_3 + 6u_4 - 18u_5 - 6u_6 + 15u_7)x \\
& \frac{1}{42h^2}(-18u_0 + 6u_1 + 12u_2 + 6u_3 + 12u_4 + 6u_5 - 12u_6 - 12u_7)y; \\
\frac{\partial p_z}{\partial y} =& \frac{1}{42h}(-44u_0 + 38u_1 + 6u_2 + 38u_3 - 8u_4 - 11u_5 - 27u_6 + 8u_7) \\
& \frac{1}{42h^2}(-18u_0 + 6u_1 + 12u_2 + 6u_3 + 12u_4 + 6u_5 - 12u_6 - 12u_7)x \\
& \frac{1}{21h^2}(12u_0 - 18u_1 + 6u_2 - 18u_3 + 6u_4 + 3u_5 + 15u_6 - 6u_7)y.
\end{aligned}
$$

Then we obtain the recovered gradient at boundary vertex z (see Fig 2.2.6)

$$
G_h^2 u(z) = \frac{1}{42h} \begin{pmatrix} 44u_0 + 11u_1 + 8u_2 - 38u_3 - 6u_4 - 38u_5 - 8u_6 + 27u_7 \\ \\ -44u_0 + 38u_1 + 6u_2 + 38u_3 - 8u_4 - 11u_5 - 27u_6 + 8u_7 \end{pmatrix}. \tag{2.2.7}
$$

The following Taylor expansion is computed in **Mathematica** as well:

$$
G_h^2 u(z) = \begin{pmatrix} u_x(z) - \frac{h^2}{42}(14u_{xxx}(z) - 27u_{xxy}(z) - 8u_{xyy}(z)) + O(h^3) \\ \\ u_y(z) + \frac{h^2}{42}(8u_{xxy}(z) + 27u_{xyy}(z) - 14u_{yyy}(z)) + O(h^3) \end{pmatrix}; \tag{2.2.8}
$$

which again is a second-order finite difference schem.

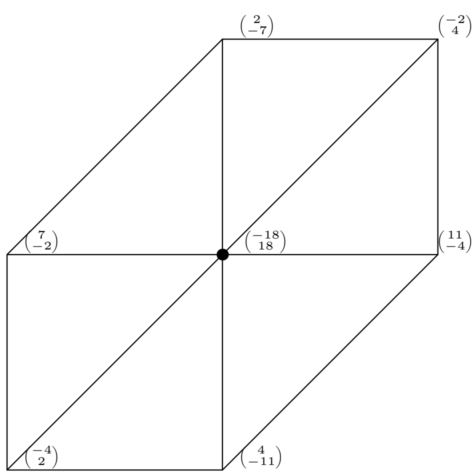

Figure 2.2.5: Denominator $42h$

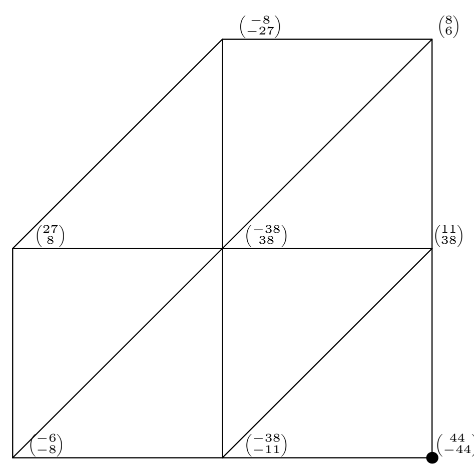

Figure 2.2.6: Denominator $6h$

Remark. The main difference between Strategy 1 and Strategy 2 is that the former fits quadratic polynomials at some interior vertices near z but the later fits a quadratic polynomial at the very boundary vertex z.

Example 2. In this example, a typical boundary vertex, as plotted in Fig 2.2.7, in chevron pattern mesh is considered. Firstly, we employ Strategy 1 to this case. Repeating the same procedure as in *Example 1*, we find that

$$G_h^1 u(z) = \frac{1}{12h} \begin{pmatrix} -6u_4 + 6u_6 \\ \\ 10u_0 + 7u_1 - 6u_2 + 7u_3 - 7u_4 - 4u_5 - 7u_6 \end{pmatrix}. \tag{2.2.9}$$

as shown in Fig 2.2.7. It is easy to verify in **Mathematica** that

$$G_h^1 u(z) = \begin{pmatrix} u_x(z) - \frac{h^2}{6} u_{xxx}(z) + O(h^3) \\ \\ u_y(z) + \frac{h^2}{12}(7u_{xxy}(z) - 4u_{yyy}(z)) + O(h^3) \end{pmatrix}; \tag{2.2.10}$$

which provides a second-order approximation to the exact gradient ∇u. Then we con-

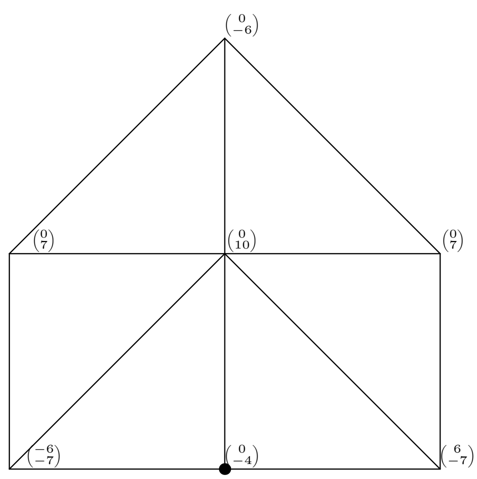

Figure 2.2.7: Denominator $12h$

sider Strategy 2. The patch $\mathcal{K}_z$ of z is showed in Fig 2.2.8. Following the same procedure as *Example 1*, we derive that

$$G_h^{2x}u(z) = \frac{1}{140h}\left(-28u_5 - 14u_6 + 14u_8 + 28u_9\right),$$

and

$$G_h^{2y}u(z) = \frac{1}{140h}\left(66u_0 + 61u_1 - 70u_2 + 61u_3 + 46u_4 - 52u_5\right.$$
$$\left. -37u_6 - 37u_7 - 37u_8 - 52u_9 + 46u_{10}\right);$$

where G_h^{2x} and G_h^{2y} represent the first and second row of G_h^2 respectively. Note that Strategy 2 uses larger patch, see Fig 2.2.8, but it also produces a second-order finite difference scheme. Actually, we have

$$G_h^2 u(z) = \begin{pmatrix} u_x(z) - \frac{17h^2}{30}u_{xxx}(z) + O(h^3) \\ \\ u_y(z) + \frac{h^2}{12}(21u_{xxy}(z) - 4u_{yyy}(z)) + O(h^3) \end{pmatrix}. \tag{2.2.11}$$

Example 3. This example demonstrates that G_h^1 and G_h^2 may involve the same vertices

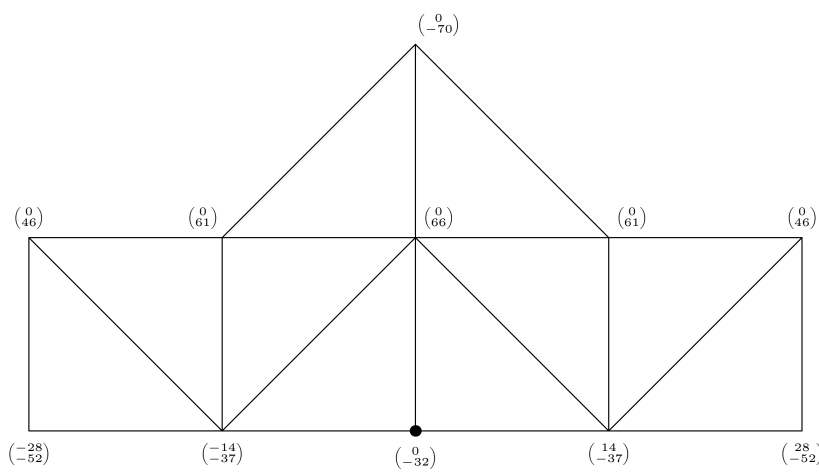

Figure 2.2.8: Denominator $140h$

but produce different finite difference schemes. Let z be a boundary vertex as plotted in Fig 2.2.9. As for Strategy 1, we need to fit three least square polynomials at three interior vertices z_0, z_1 and z_2 connecting z and then take average. It is not hard to compute that

$$G_h^{1x}u(z) = \frac{1}{36h}\left(-10u_0 - 7u_3 + 5u_4 + 2u_5 - u_6 - 13u_7 \right.$$
$$\left. +10u_9 - 2u_{10} - 5u_{11} + 7u_{12} + 13u_{13} + u_{14}\right),$$

and

$$G_h^{1y}u(z) = \frac{1}{36h}\left(-10u_0 + 24u_1 - 4u_3 - 5u_4 + 8u_5 - 3u_6 - 12u_7 \right.$$
$$\left. -24u_8 + 16u_9 + 8u_{10} - 5u_{11} - 4u_{12} - 12u_{13} - 3u_{14}\right),$$

where G_h^{1x} and G_h^{1y} are two rows of G_h^1. Using **Mathematica** to compute the Taylor expansion, we obtain

$$G_h^1 u(z) = \begin{pmatrix} u_x(z) - \frac{h^2}{6}(u_{xxx}(z) + u_{xyy}(z)) + O(h^3) \\ \\ u_y(z) - \frac{h^2}{3}u_{yyy}(z) + O(h^3) \end{pmatrix}. \tag{2.2.12}$$

which clearly indicates that G_h^1 provides a second order approximation to the exact gradient $\nabla u(z)$.

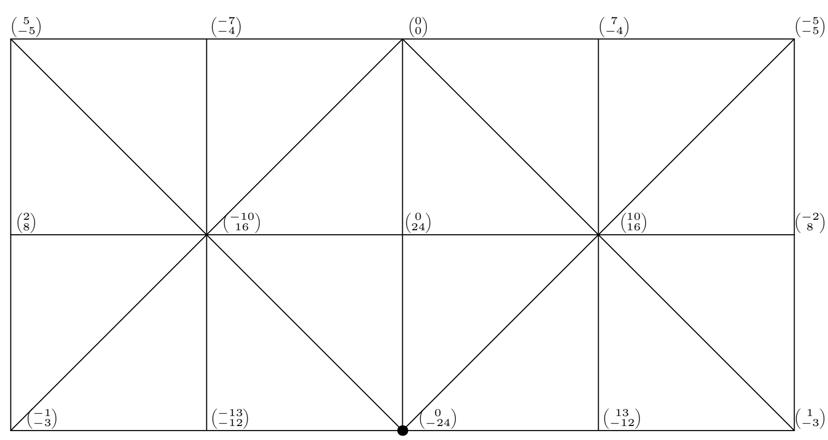

Figure 2.2.9: Denominator $36h$

To see how Strategy 2 works, we construct patch $\mathcal{K}_z$, as shown in Fig 2.2.10, in two steps. Using all vertices in $\mathcal{K}_z$, fit a quadratic polynomial at z which yields

$$G_h^{2x}u(z) = \frac{1}{60h}\left(2u_1 + 4u_2 - 2u_3 - u_4 + u_6 + 2u_7 - 4u_8 - 2u_9 \right.$$
$$\left. -10u_{10} - 5u_{11} + 5u_{13} + 10u_{14}\right),$$

and

$$G_h^{2y}u(z) = \frac{1}{10h}\left(4u_0 + 4u_1 + 4u_2 - u_3 - u_4 - u_5 - u_6 - u_7 \right.$$
$$\left. +4u_8 + 4u_9 - 3u_{10} - 3u_{11} - 3u_{12} - 3u_{13} - 3u_{14}\right),$$

where G_h^{2x} and G_h^{2y} have the same meaning as previous example. Taylor expansion results in

$$G_h^2 u(z) = \begin{pmatrix} u_x(z) + \frac{h^2}{30}(17u_{xxx}(z) - 5u_{xyy}(z)) + O(h^3) \\ \\ u_y(z) + \frac{h^2}{3}(3u_{xxy}(z) - u_{yyy}(z)) + O(h^3) \end{pmatrix}. \tag{2.2.13}$$

This means that $G_h^2 u(z)$ is also a second order approximation of the exact gradient $\nabla u(z)$.

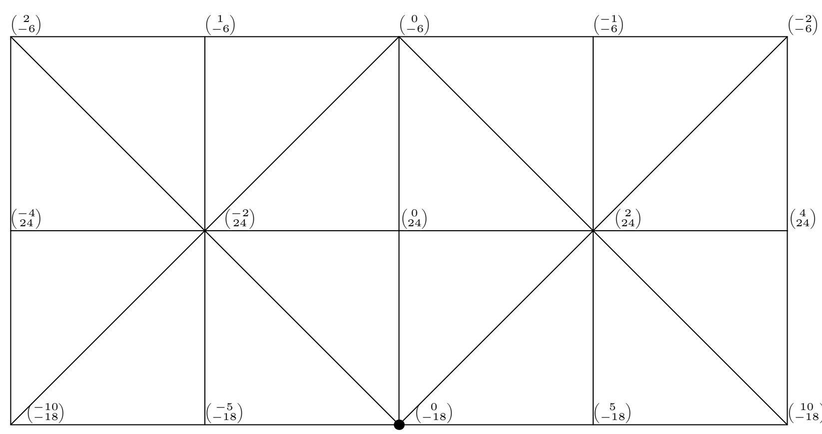

Figure 2.2.10: Denominator $60h$

Remark. Comparing the computational complexity of Strategy 1 and Strategy 2, we see that Strategy 1 needs to perform three least-squares fittings with three 9×6 matrices. On the other hand, Strategy 2 does one least-squares fittings with one 15×6 matrix. Thus the computational cost of those two strategies are comparable.

Remark. We have discussed three cases to illustrate proposed two strategies for PPR on boundary. Indeed, both $G_h^1 u$ and $G_h^2 u$ converge to ∇u with second-order rate for all boundary vertices of arbitrary mesh due to the polynomial preserving property.

2.3 Numercial Examples

In this section, we provide four numerical examples to verify superconvergence and robustness of our boundary recovery strategies and also compare the results with COMSOL Multiphysics integrated $'ppr'$ command. In order to detect boundary influence, define $\mathcal{N}_{h,2} = \{z \in \mathcal{N}_h : \text{dist}(z, \partial\Omega) \leq L\}$ be the set of all near boundary nodes and let $\mathcal{N}_{h,1} = \mathcal{N}_h \setminus \mathcal{N}_{h,2}$ denote the set of nodes away from boundary. Now, the domain Ω_h is splitted into $\Omega_{h,1}$ and $\Omega_{h,2}$ where

$$\Omega_{h,1} = \bigcup \{\tau \in \mathcal{T}_h : \text{all vertices in } \tau \in \mathcal{N}_{h,1}\}, \tag{2.3.1}$$

and

$$\Omega_{h,2} = \Omega \setminus \Omega_{h,1}, \tag{2.3.2}$$

where L is some small quantity to indicates the *width* of the boundary. In this section, the width of the boundary is chosen as $L = 0.1$.

The notations used are the following:

$De = \nabla(u - u_h)$, where u_h is the finite element solution.

$De^1 = \nabla u - G_h^1 u_h$, where $G_h^1 u_h$ is defined by PPR using Strategy 1.

$De^2 = \nabla u - G_h^2 u_h$, where $G_h^2 u_h$ is defined by PPR using Strategy 2.

$De^3 = \nabla u - G_h^3 u_h$, where $G_h^3 u_h$ is defined by PPR using COMSOL Multiphysics integrated $'ppr'$ command.

All computations are carried out in COMSOL Multiphysics 3.5a on Delaunay triangulation. We perform three levels mesh refinement by connecting midpoints of each triangles.

Example 1. We first consider a symmetric and infinitely smooth case:

$$-\Delta u = 2\pi^2 \sin \pi x \sin \pi y, \quad \text{in} \quad \Omega = [0,1]^2,$$

with $u = 0$ on $\partial\Omega$. The exact solution is $u(x,y) = \sin \pi x \sin \pi y$.

The maximum error of $\nabla u - G_h u_h$ for interior nodes and near boundary nodes are depicted in Table 2.3.1 and Table 2.3.2, respectively. It can be observed that after performing PPR by any of the three methods, the maximum error decreases significantly comparing to that without performing gradient recovery processing. In Table 2.3.1, the L^∞ norm of De^1 and De^2 are identical since they have the same strategy for the interior nodes and only differ on the boundary. It is worth to point out that to achieve the same accuracy, PPR 1 or PPR 2 requires approximately only $\frac{1}{4}$ degrees of freedom (DOF) of COMSOL Multiphysics integrated $'ppr'$ command.

In Table 2.3.2, we observe clearly superconvergence phenomena. Before recovery, De shows a convergence rate $O(N^{-\frac{1}{2}})$. After PPR, our second strategy converges at a rate of $O(N^{-1})$. Moreover, to achieve the same level of accuracy, PPR 1 requires approximately $\frac{1}{4}$ degrees of freedom of COMSOL Multihphysics.

Table 2.3.1: Example 1: $\|\cdot\|_{L_\infty(\mathcal{N}_{h,1})}$ on Delaunay Triangulation

DOF	De	order	De^1	order	De^2	order	De^3	order
1241	2.37e-01	–	1.54e-02	–	1.54e-02	–	2.58e-02	–
4841	1.27e-01	0.46	5.48e-03	0.76	5.48e-03	0.76	1.66e-02	0.32
19121	6.55e-02	0.48	2.34e-03	0.62	2.34e-03	0.62	8.02e-03	0.52
76001	3.33e-02	0.49	1.11e-03	0.54	1.11e-03	0.54	3.73e-03	0.56

In addition, we report the L_2 error in Table 2.3.3 and Table 2.3.4. As expected, it is observed that $\nabla(u - u_h)$ is $O(N^{-\frac{1}{2}})$. Concerning the convergence of recovered gradients, all three strategies show superconvergence at rate of $O(N^{-1})$ in the interior domain and near the boundary region.

Table 2.3.2: Example 1: $\|\cdot\|_{L_\infty(\mathcal{N}_{h,2})}$ on Delaunay Triangulation

DOF	De	order	De^1	order	De^2	order	De^3	order
1241	2.87e-01	–	8.08e-03	–	2.83e-02	–	2.76e-02	–
4841	1.45e-01	0.50	2.36e-03	0.90	7.08e-03	1.02	7.03e-03	1.00
19121	7.25e-02	0.50	9.85e-04	0.64	1.77e-03	1.01	2.23e-03	0.83
76001	3.63e-02	0.50	4.48e-04	0.57	4.48e-04	0.99	8.08e-04	0.74

Table 2.3.3: Example 1: $\|\cdot\|_{L_2(\Omega_{h,1})}$ on Delaunay Triangulation

DOF	De	order	De^1	order	De^2	order	De^3	order
1241	5.70e-02	–	6.92e-03	–	6.92e-03	–	5.57e-03	–
4841	2.86e-02	0.505	1.80e-03	0.98	1.82e-03	0.98	1.63e-03	0.90
19121	1.45e-02	0.497	4.83e-04	0.98	4.83e-04	0.97	4.10e-04	0.99
76001	7.27e-03	0.500	1.26e-04	0.97	1.26e-04	0.97	1.09e-04	0.97

Table 2.3.4: Example 1: $\|\cdot\|_{L_2(\Omega_{h,2})}$ on Delaunay Triangulation

DOF	De	order	De^1	order	De^2	order	De^3	order
1241	5.17e-02	–	4.98e-03	–	7.07e-03	–	4.04e-03	–
4841	2.58e-02	0.51	1.32e-03	0.97	1.59e-03	1.09	1.01e-03	1.01
19121	1.27e-02	0.51	3.34e-04	1.00	3.69e-04	1.07	2.53e-04	1.01
76001	6.33e-03	0.51	8.48e-05	0.99	8.92e-05	1.03	6.26e-05	1.01

Example 2. Our second example is:

$$-\Delta u = 1, \quad \text{in} \quad \Omega = [0,1]^2,$$

with $u = 0$ on $\partial\Omega$. The exact solution is given by the infinite series

$$\begin{aligned} u(x,y) =& \frac{x(1-x)+y(1-y)}{4} - \frac{2}{\pi^3}\sum_{m=0}^{\infty}\frac{1}{(2m+1)^3(1+e^{-(2m+1)\pi})} \\ &\cdot \{[e^{-(2m+1)\pi y} + e^{-(2m+1)\pi(1-y)}]\sin((2m+1)\pi x) \\ &+ [e^{-(2m+1)\pi x} + e^{-(2m+1)\pi(1-x)}]\sin((2m+1)\pi y)\}. \end{aligned}$$

This problem has weak singularities at four corners. In order to observe the asymptotic behavior of numerical approximations, we start from the second mesh level in the previous example and perform one more level mesh refinement. The maximum error of gradient and convergence rates are reported in Table 2.3.5 and Table 2.3.6. Due to the corner singularities, the maximum error occurs near the boundary and it is observed in Table 2.3.6. It can be seen that all strategies have enhanced the maximum error of gradient as expected. In Table 2.3.5, we can also observe that De^1 and De^2 on level 2 are comparable to De^3 on level 4. In Table 2.3.6, De^1 in level 3 is even smaller than De^2 and De^3 on level 4.

The L_2 errors are displayed in Table 2.3.7 and Table 2.3.8. Inside the domain, Strategy 1 and Strategy 2 superconverges at rate $\approx O(N^{-0.9})$ while COMSOL Multiphysics integrated $'ppr'$ command superconverges at rate $\approx O(N^{-1})$. However, we can observe smaller errors in both of our strategies than in COMSOL Multiphysics. Concerning the performing PPR near boundary, all three strategies are comparable and superconvergent.

Table 2.3.5: Example 2: $\|\cdot\|_{L_\infty(\mathcal{N}_{h,1})}$ on Delaunay Triangluation

DOF	De	order	De^1	order	De^2	order	De^3	order
4841	1.16e-02	–	3.34e-04	–	3.33e-04	–	1.27e-03	–
19121	5.99e-03	0.48	1.58e-04	0.55	1.58e-04	0.55	5.41e-04	0.62
76001	2.98e-03	0.51	8.03e-05	0.49	8.03e-05	0.49	2.57e-04	0.54
303041	1.49e-03	0.50	4.04e-05	0.50	4.05e-05	0.50	1.30e-04	0.50

Example 3. We now consider an anisotropic diffusion problem defined in the unit square $\Omega = (0,1)^2$ as follows

$$\begin{cases} -\nabla \cdot (\mathcal{A}\nabla u) = f, \quad \text{in} \quad \Omega \\ u = 0, \quad \text{on} \quad \partial\Omega \end{cases}$$

Table 2.3.6: Example 2: $\|\cdot\|_{L_\infty(\mathcal{N}_{h,2})}$ on Delaunay Triangulation

DOF	De	order	De^1	order	De^2	order	De^3	order
4841	3.07e-02	–	3.42e-03	–	8.51e-03	–	6.99e-03	–
19121	1.77e-02	0.40	1.54e-03	0.58	4.22e-03	0.51	3.45e-03	0.51
76001	9.95e-03	0.42	7.73e-04	0.50	2.18e-03	0.48	1.80e-03	0.47
303041	5.44e-03	0.44	3.41e-04	0.59	1.04e-03	0.53	8.53e-04	0.54

Table 2.3.7: Example 2: $\|\cdot\|_{L_2(\Omega_{h,1})}$ on Delaunay Triangulation

DOF	De	order	De^1	order	De^2	order	De^3	order
4841	2.24e-03	–	6.21e-05	–	6.21e-05	–	1.26e-04	–
19121	1.14e-03	0.49	1.91e-05	0.86	1.91e-05	0.86	3.33e-05	0.97
76001	5.75e-04	0.50	5.54e-06	0.90	5.54e-06	0.90	8.76e-06	0.97
303041	2.89e-04	0.50	1.56e-06	0.92	1.56e-06	0.92	2.31e-06	0.96

Table 2.3.8: Example 2: $\|\cdot\|_{L_2(\Omega_{h,2})}$ on Delaunay Triangulation

DOF	De	order	De^1	order	De^2	order	De^3	order
4841	3.09e-03	–	1.22e-04	–	3.11e-04	–	2.47e-04	–
19121	1.53e-03	0.51	3.33e-05	0.94	7.94e-05	0.99	6.71e-05	0.95
76001	7.63e-04	0.50	9.04e-06	0.94	2.02e-05	0.99	1.75e-05	0.98
303041	3.80e-04	0.50	2.44e-06	0.95	5.15e-06	0.99	4.59e-06	0.97

where the diffusion matrix is given by

$$\mathcal{A} = \begin{pmatrix} k^2 & 0 \\ \\ 0 & 1 \end{pmatrix},$$

and $f(x)$ is chosen such that the exact solution is $u = sin(\pi x)sin(k\pi y)$. We test the case $k = 10$. For anisotropic problems, it is more suitable to use anisotropic meshes or adaptive meshes. Nevertheless, for the sake of identifying the performance of PPR, the same Delaunay meshes as in Example 1 would serve the purpose. The results are listed in Table 2.3.9 and Table 2.3.10. The numerical results indicate that all three PPR strategies have improved the error on each mesh level.

As for L_2 error, it can observed from Table 2.3.11 and Table 2.3.12 that all three strategies superconverge at rate of $O(N^{-1})$ asymptotically.

Table 2.3.9: Example 3: $\|\cdot\|_{L_\infty(\mathcal{N}_{h,1})}$ on Delaunay Triangulation

DOF	De	order	De^1	order	De^2	order	De^3	order
1241	1.82e+01	–	1.63e+01	–	1.63e+01	–	1.59e+01	–
4841	1.08e+01	0.38	6.89e+00	0.63	6.89e+00	0.63	6.89e+00	0.62
19121	5.10e+00	0.55	2.35e+00	0.78	2.35e+00	0.78	2.39e+00	0.77
76001	2.44e+00	0.53	8.21e-01	0.76	8.21e-01	0.76	1.05e+00	0.60

Table 2.3.10: Example 3: $\|\cdot\|_{L_\infty(\mathcal{N}_{h,2})}$ on Delaunay Triangulation

DOF	De	order	De^1	order	De^2	order	De^3	order
1241	1.27e+01	–	4.84e+00	–	4.56e+00	–	5.14e+00	–
4841	6.63e+00	0.48	1.92e+00	0.68	1.92e+00	0.63	1.92e+00	0.72
19121	3.35e+00	0.50	6.02e-01	0.84	6.03e-01	0.84	6.99e-01	0.74
76001	1.68e+00	0.50	2.38e-01	0.67	2.38e-01	0.67	4.55e-01	0.31

Table 2.3.11: Example 3: $\|\cdot\|_{L_2(\Omega_{h,1})}$ on Delaunay Triangulation

DOF	De	order	De^1	order	De^2	order	De^3	order
1241	5.16e+00	–	5.38e+00	–	5.38e+00	–	4.38e+00	–
4841	2.29e+00	0.60	1.96e+00	0.74	1.96e+00	0.74	1.69e+00	0.70
19121	9.79e-01	0.62	5.81e-01	0.89	5.81e-01	0.89	4.83e-01	0.91
76001	4.50e-01	0.56	1.58e-01	0.94	1.58e-01	0.94	1.40e-01	0.90

Example 4. In all previous examples, solutions are analytic. Let us consider the Laplace equation on the L-shaped domain $\Omega = (-1,1)\times(-1,1)\setminus(0,1)\times(-1,0)$. The Dirichlet boundary condition is imposed so that the true solution $u = r^{2/3}sin(2\theta/3)$ in

Table 2.3.12: Example 3: $\|\cdot\|_{L_2(\Omega_{h,2})}$ on Delaunay Triangulation

DOF	De	order	De^1	order	De^2	order	De^3	order
1241	1.94e+00	–	1.55e+00	–	1.58e+00	–	1.13e+00	–
4841	9.54e-01	0.52	4.94e-01	0.84	5.00e-01	0.84	3.92e-01	0.78
19121	4.63e-01	0.53	1.31e-01	0.97	1.32e-01	0.97	9.96e-02	1.00
76001	2.29e-01	0.51	3.48e-02	0.96	3.49e-02	0.96	2.85e-02	0.91

polar coordinates. In order to remove the pollution caused by the corner singularity, recovery based adaptive method [18] is employed. We start with an initial mesh shown in Fig 2.3.1 and use Dörfler marking strategy [8] with $\theta = 0.3$.

Due to the corner singularity, the maximum error of $\nabla u - \nabla u_h$ is divergent. Hence we track $||\nabla u - \nabla u_h||_{0,\Omega}$ and $||\nabla u - G_h u_h||_{0,\Omega}$ instead. The numerical results are depicted in Fig 2.3.2. For PPR with both Strategy 1 and Strategy 2, a superconvergence rate $O(N^{-1})$ is observed, where N represents the total degrees of freedom. We also test the '*ppr*' command in COMSOL Multiphysics and obtain a superconvergence rate $O(N^{-0.9})$. In Fig 2.3.2, a comparison among different strategies is made. It is observed that to achieve the same level of accuracy, both Strategy 1 and Strategy 2 require less degrees of freedom than PPR in COMSOL Multiphysics, and De^1 needs almost half less degrees of freedom than De^3.

2.4 Conclusion remarks

In this chapter, we have introduced two strategies to improve performance of PPR gradient recovery on boundary. Numerical tests provide convincing evidence that our methods inherit the superconvergence property of PPR in the interior of solution domains.

It is also worth to emphasize that both strategies are problem independent and method independent just as PPR itself. In order to obtain recovered gradient on the

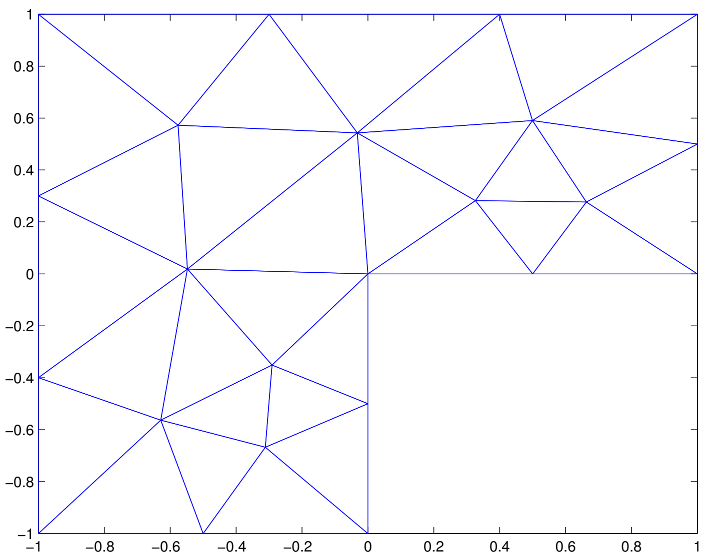

Figure 2.3.1: Initial mesh for Example 4

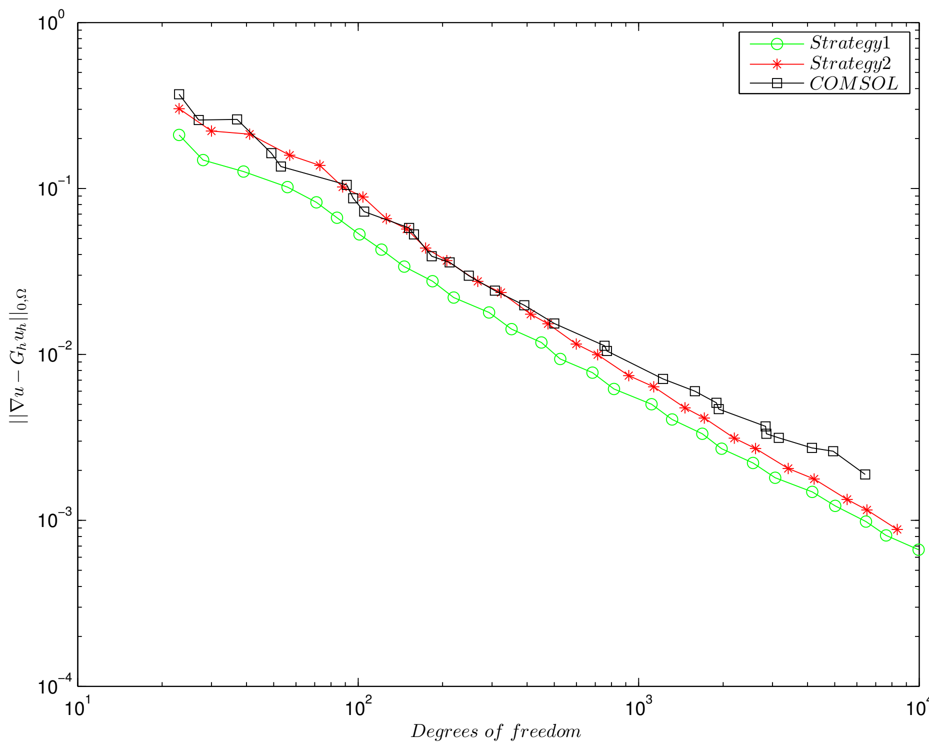

Figure 2.3.2: Comparison of decay of error among different strategies

boundary, all we need are numerical data nearby. It does not matter what the original problem is, even though the quality of the recovery might be influenced by the underlying problem, the method itself is universal. Although our technique is demonstrated for the finite element method, it can be well applied to other methods, such as finite difference method and finite volume method, as long as numerical data are provided at some sampling points. In later sections, we will extend this technique to the newly proposed WGFEMs.

Finally, boundary recovery technique can be used at an interface, where the solution or its gradient has jumps. In other words, we treat an interface (if the location is known *a priori*) as a boundary when performing gradient recovery. We will apply this idea on the weak Galerkin method for interface problem in chapter 5.

CHAPTER 3 WGFEMS FOR 2ND ORDER ELLIPTIC PROBLEMS

WGFEMs refers to finite element techniques for partial differential equations in which differential operators are approximated by weak forms as distributions. Let K be any polygonal domain with boundary ∂K. A weak function v on K refers to a function $v = \{v_0, v_b\}$ such that $v_0 \in L^2(K)$ and $v_b \in H^{\frac{1}{2}}(K)$. Denote by $W(K)$ the space of weak function on K:

$$W(K) := \{v = \{v_0, v_b\} : \ v_0 \in L^2(K), v_b \in H^{\frac{1}{2}}(K)\}. \tag{3.0.1}$$

One can treat v_0 as the value of v in K, and v_b as the value of v on ∂K. Note that v_b may not necessarily be related to the trace of v_0 on ∂K should a trace be well-defined.

Definition 3.0.1. *For any $v \in W(K)$, the weak gradient of v is defined as a linear functional $\nabla_w v$ in the dual space of $H(div, K)$ whose action on each $q \in H(div, K)$ is given by*

$$(\nabla_w v, q)_K := -(v_0, \nabla \cdot q)_K + < v_b, q \cdot \mathbf{n} >_{\partial K}, \tag{3.0.2}$$

where $\mathbf{n}$ is the outward normal direction to ∂K, $(v_0, \nabla \cdot q)_K = \int_K v_0(\nabla \cdot q) dK$ is the action of v_0 on $\nabla \cdot q$, and $< v_b, q \cdot \mathbf{n} >_{\partial K}$ is the action of $q \cdot \mathbf{n}$ on $v_b \in H^{\frac{1}{2}}(\partial K)$.

By choosing a finite element subspace of $H(\text{div}, K)$, we obtain a discrete weak gradient. When K is a domain such as triangles, tetrahedron, rectangles and cubes, we choose Raviart-Thomas element or BDM element.

Let $P_r(K)$ be the set of polynomials on K with degree no more than r and $\hat{P}_k(K)$ be the set of homogeneous polynomials of order k in the variable $\mathbf{x} = (x_1, \cdots, x_d)^T$. Let $G_k(K)$ be either $[P_k(K)]^d$ or $RT_k(K) = [P_k(K)]^d + \hat{P}_k(K)\mathbf{x}$. For this thesis, we choose $d = 2$.

Definition 3.0.2. *The discrete weak gradient of v denoted by $\nabla_{w,k,K} v$ is defined as* *the unique polynomial* $(\nabla_{w,k,K} v) \in G_k(K)$ *satisfying the following equation*

$$(\nabla_{w,k,K} v, q) = -(v_0, \nabla \cdot q)_K + < v_b, q \cdot \mathbf{n} >_{\partial K}, \ \forall q \in G_k(K). \tag{3.0.3}$$

For the weak function space $W(K)$*, we discretize it by* $W_{j,l}(K)$ *given as follows*

$$W_{j,l}(K) := \{v = \{v_0, v_b\} : v_0 \in P_j(K), v_b \in P_l(\partial K)\}.$$

Note that if $v \in H^1(K)$ and $\nabla v \in G_k(K)$, then $\nabla_{w,k,K} v = \nabla v$.

Different weak Galerkin finite element methods can be derived by choosing $W_{j,l}(K)$ and $G_k(K)$ with various combinations of the indices j, l and k. Please refer to [118] for details. In this thesis, we will consider the case (P_0, P_0, RT_0) and (P_1, P_1, P_0) element.

For any given integer $k \geq 1$, denote by $W_k(T)$ the discrete weak function space consisting of polynomials of degree k in T and piecewise polynomials of degree k on each flat spaces of ∂T, that is

$$W_k(T) := \{v = \{v_0, v_b\} : v_o \in \mathbb{P}_k(T), v_b|_e \in \mathbb{P}_k(e), e \in \partial T\}. \tag{3.0.4}$$

Patching together $W_k(T)$ over all elements $T \in \mathcal{T}_h$, the weak Galerkin finite element spaces W_h is given by

$$W_h := \prod_{T \in \mathcal{T}_h} W_k(T). \tag{3.0.5}$$

3.1 WG scheme for (P_0, P_0, RT_0) element

Let $\mathcal{T}_h$ be a shape-regular, quasi-uniform mesh of the domain Ω, with mesh size h. Denote by $\mathcal{E}_h$ the set of all edges or faces in $\mathcal{T}_h$, and let $\mathcal{E}_h^0 = \mathcal{E}_h \setminus \partial\Omega$ be the set of all interior edges or faces. We now define global weak Galerkin finite element spaces

$$V_h := \{v = \{v_0, v_b\} : \{v_0, v_b\}|_T \in W_k(T)\},$$

$$V_h^0 := \{v : v \in V_h, v_b = 0 \text{ on } \partial\Omega\}.$$

The component v_0 is defined element-wise and totally discontinuous. The component v_b is defined on edges/faces which glue v_0 in different elements to be a reasonable approximation of a function in $H_0^1(\Omega)$.

Denote by $\nabla_{w,k}$ the discrete weak gradient operator on V_h computed on each element T, i.e.,

$$(\nabla_{w,k} v)|_T := \nabla_{w,k,T}(v|_T), \ \forall v \in V_h.$$

For simplicity of notation, we shall drop the subscript k from now on in the notation $\nabla_{w,k}$ for the discrete weak gradient.

A numerical approximation for the model problem (2.0.1) can be obtained by seeking $u_h = \{u_0, u_b\} \in V_h^0$ satisfying $u_b = Q_b g$ on $\partial\Omega$ and the following equation:

$$a(u_h, v) = (f, v_0), \ \forall v = \{v_0, v_b\} \in V_h^0, \tag{3.1.1}$$

where

$$a(w, v) = (\nabla_w w, \nabla_w v) := \sum_{T \in \mathfrak{T}_h} (\nabla_w w, \nabla_w v)_T. \tag{3.1.2}$$

and $Q_b g$ is an approximation of the boundary value in the polynomial space $\mathbb{P}_l(\partial T \cap \partial\Omega)$. For simplicity, $Q_b g$ is taken as the standard L^2 projection for each boundary segment; other approximations of the boundary value $u = g$ can also be employed in (3.1.1).

Lemma 3.1.1. *Let $u \in H^1(\Omega)$ be the solution of (2.0.1) and $u_h \in S_h(j, j+1)$ be the weak Galerkin approximation of u obtained from (3.1.1). Let $Q_h u$ be the L^2 projection of the exact solution u, then there exists a constant C and positive constant K such that*

$$\frac{\alpha_1}{2}(\|\nabla_w(u_h - Q_h u)\|^2 + \|u_0 - Q_0 u\|^2) \leq C(\|\Pi_h(\nabla u) - R_h(\nabla u)\|^2 + K\|u_0 - Q_0 u\|^2.$$

The following approximation theorem holds true and was proved in [118].

Theorem 3.1.2. *Assume that the dual of the problem (2.0.1) has the H^{1+s} regularity,*

$s \in (0,1]$. Let $u \in H^1(\Omega)$ be the solution of (2.0.1) and assume it is sufficiently smooth such that $u \in H^{m+1}(\Omega)$ with $0 \le m \le j+1$. Let u_h be a weak Galerkin approximation of u from (3.1.1) by using the weak finite element space $S_h(j,j)$. Let $Q_h u$ be the L^2 projection of u in the corresponding finite element space. Then, there exists a constant C such that

$$\|\nabla_w(u_h - Q_h u)\| + \|u_0 + Q_0 u\| \le C(h^m\|u\|_{m+1} + h^{1+s}\|f - Q_0 f\|), \tag{3.1.3}$$

$$\|u_h - Q_h u\| \le C(h^{1+s}\|f - Q_0 f\| + h^{m+s}\|u\|_{m+1}). \tag{3.1.4}$$

3.2 WG scheme for (P_1, P_1, P_0) element

Denote by V_h a finite element space of V consisting of functions of W_h which are continuous across each interior edge:

$$V_h = \{v \in W_h : [v]_e = 0, \forall e \in \mathcal{E}_h^0\}.$$

and let V_h^0 be a subspace of V_h consisting of functions with vanishing boundary value

$$V_h^0 = \{v \in V_h, v = 0 \text{ on } \partial\Omega\}.$$

On V_h, we define the two following forms:

$$a(v,w) = \sum_{T \in \mathcal{T}_h} \int_T \nabla_w v \cdot \nabla_w w dT, \tag{3.2.1}$$

$$s(v,w) = \sum_{T \in \mathcal{T}_h} h_T^{-\beta} < v_0 - v_b, w_0 - w_b >_{\partial T} . \tag{3.2.2}$$

Denote by $a_s(\cdot,\cdot)$ a stabilization of $a(\cdot,\cdot)$ given by

$$a_s(v,w) = a(v,w) + s(v,w). \tag{3.2.3}$$

The weak galerkin scheme is then given as following: seek $u_h = \{u_0, u_b\} \in V_h$ satisfying both $u_b = Q_b g$ on $\partial\Omega$ and the following equation:

$$a_s(u_h, v) = (f, v_0), \tag{3.2.4}$$

$\forall v = \{v_0, v_b\} \in V_h^0$, where $Q_b g$ is an approximation of the Dirichlet boundary value in the polynomial space $\mathbb{P}_k(\partial T \cap \partial\Omega)$. For simplicity, we take $Q_b g$ as the standard L^2 projection of the boundary value g on each boundary segment.

The following approximation estimates hold and was proved in [98].

Theorem 3.2.1. *Let $u_h \in V_h$ be the weak Galerkin finite element solution of the problem (2.0.1) arising from (3.2.4). Assume that the exact solution is so regular that $u \in H^{k+1}(\Omega)$. Then there exists a constant C such that*

$$|||u_h - Q_h u||| \leq Ch^k \|u\|_{k+1}. \tag{3.2.5}$$

Furthermore, onsider the dual problem that seeks $\Phi \in H_0^1(\Omega)$ satisfying

$$-\nabla \cdot (\nabla \Phi) = e_0, \text{in } \Omega, \tag{3.2.6}$$

we assume that the usual H^2-regularity is satisfied for the dual problem. Then we have the following estimates:

Theorem 3.2.2. *In addition to the assumptions of Theorem (3.2.1), we also assume that the dual problem (3.2.6) has the usual H^2-regularity. Then there exists a constant C such that*

$$\|Q_0 u - u_0\| \leq Ch^{k+1} \|u\|_{k+1}. \tag{3.2.7}$$

3.3 Supercloseness Analysis

We first introduce the definition of the mesh structure which guarantees the supercloseness result.

Definition 3.3.1. *[38] Two adjacent triangles are said to form an $\mathcal{O}(h^{1+\alpha})$ approximate parallelogram if the lengths of any two opposite edges differ only by $\mathcal{O}(h^{1+\alpha})$.*

Definition 3.3.2. *[38] The triangluation $\mathcal{T}_h$ is called to satisfy Condition (α, σ) if there exists a partition $\mathcal{T}_{1,h} \cup \mathcal{T}_{2,h}$ of $\mathcal{T}_h$ and positive constants α and σ such that every two adjacent triangles in $\mathcal{T}_{1,h}$ form an $\mathcal{O}(h^{1+\alpha})$ parallelogram and*

$$\sum_{T \in \mathcal{T}_{2,h}} |T| = \mathcal{O}(h^{\sigma}).$$

The following lemma is proved in [38] by Xu and Zhang.

Lemma 3.3.3. *Assume that $\mathcal{T}_h$ satisfy Condition (α, σ), then for any $v_h \in S_h$,*

$$|\sum_{T \in \mathcal{T}_h} \int_T \nabla(I_h u - u) \cdot \nabla v_h| \leq ch^{1+\rho}(\|u\|_{3,\Omega} + |u|_{2,\infty,\Omega})|v|_{1,\Omega}, \tag{3.3.1}$$

where $\rho = \min(\alpha, \frac{\sigma}{2}, \frac{1}{2})$ and $I_h u \in S_h$ is the interpolation of u.

Lemma 3.3.4. *The interpolation operator I_h satisfies*

$$(\nabla_w I_h v, \vec{q})_h = (\nabla I_h v, \vec{q})_h, \forall v \in C^0(\Omega), q \in W_h, \tag{3.3.2}$$

where $(\cdot, \cdot)_h = \sum_{T \in \mathcal{T}_h} (\cdot, \cdot)_T$.

The following lemma is proved by Wang et. al. in [119].

Lemma 3.3.5. *The functional $||| \cdot ||| : V_h \to \mathbb{R}$ defined by*

$$|||v|||^2 = a_s(v, v), \forall v \in V_h, \tag{3.3.3}$$

is a norm on the space V_h^0. Furthermore, the following inequalities hold true:

$$\sum_{T \in \mathcal{T}_h} \|\nabla v_0\|_T^2 \lesssim |||v|||^2, \forall v \in V_h, \tag{3.3.4}$$

$$\sum_{T \in \mathcal{T}_h} h_T^{-1} \|v_0 - v_b\|_{\partial T}^2 \lesssim |||v|||^2, \forall v \in V_h. \tag{3.3.5}$$

Now we are ready to derive an error estimate for $|||I_h u - u_h|||$, where u_h is the solution of the weak Galerkin method (3.2.4) and $I_h u$ is the interploation of the exact solution of problem (2.0.1).

Theorem 3.3.6. *Let $u \in H^3(\Omega)$ be the solution of (2.0.1) and $u_h \in V_h$ be solution of weak Galerkin method (3.2.4), we have the following error estimate:*

$$|||I_h u - u_h||| \leq ch^{\min\{1+\rho, \frac{\beta+1}{2}\}}(\|u\|_{3,\Omega} + |u|_{2,\infty,\Omega}). \tag{3.3.6}$$

Proof. From Lemma 3.3.4 and Cauchy-Swarchtz inequality, we have

$$\begin{aligned}
|||I_h u - u_h|||^2 &= a_s(I_h u - u_h, I_h u - u_h)\\
&= a_s(I_h u, I_h u - u_h) - a_s(u_h, I_h u - u_h)\\
&= \sum_{T\in\mathfrak{T}_h} (\nabla_w I_h u, \nabla_w(I_h u - u_h))_T - (f, I_h u - u_0)\\
&= \sum_{T\in\mathfrak{T}_h} (\nabla I_h u, \nabla_w(I_h u - u_h))_T - \sum_{T\in\mathfrak{T}_h} (\nabla u, \nabla(I_h u - u_0)_T\\
&+ \sum_{T\in\mathfrak{T}_h} < \nabla u\cdot\vec{n}, I_h u - u_0 - (I_h - u_b) >_{\partial\Omega}\\
&= \sum_{T\in\mathfrak{T}_h} (\nabla(I_h u - u_h), \nabla(I_h u - u_0))_T\\
&- \sum_{T\in\mathfrak{T}_h} < \nabla(I_h u - u)\cdot\vec{n}, I_h u - u_0 - (I_h - u_b) >_{\partial\Omega}\\
&\leq \sum_{T\in\mathfrak{T}_h} (\nabla(I_h u - u), \nabla(I_h u - u_0))_T\\
&+ (\sum_{T\in\mathfrak{T}_h} h_T^{\beta}\|\nabla(I_h u - u)\|_{\partial T}^2)^{\frac{1}{2}} (\sum_{T\in\mathfrak{T}_h} h_T^{-\beta}\|I_h u - u_0 - (I_h u - u_b)\|_{\partial T}^2)^{\frac{1}{2}}
\end{aligned} \tag{3.3.7}$$

For $u \in H^3 \cap W_{2,\infty}$, Lemma 3.3.3 implies

$$\sum_{T\in\mathfrak{T}_h} (\nabla(I_h u - u), \nabla(I_h u - u_0))_T \leq h^{1+\rho}(\|u\|_{3,\Omega} + |u|_{2,\infty,\Omega})|I_h u - u_0|_{1,\Omega}. \tag{3.3.8}$$

By definition of $|||\cdot|||$, we have

$$(\sum_{T\in\mathfrak{T}_h} h_T^{-\beta}\|I_h u - u_0 - (I_h u - u_b)\|_{\partial T}^2)^{\frac{1}{2}} \leq |||I_h u - u_h\ ||| \,. \tag{3.3.9}$$

Furthermore, we have

$$
\begin{aligned}
(\sum_{T\in\mathfrak{T}_h} h_T^{\beta}\|\nabla(I_hu-u)\|_{\partial T}^2)^{\frac{1}{2}} &\le (\sum_{T\in\mathfrak{T}_h} h_T^{\beta}(h^{-1}\|u-I_hu\|_{1,T}^2+h\|\nabla(u-I_hu)\|_{1,T}^2)^{\frac{1}{2}}\\
&\le (\sum_{T\in\mathfrak{T}_h} h_T^{\beta}(h^{-1}\cdot h^2\|u\|_{3,T}^2+h\|u\|_{3,T}^2))^{\frac{1}{2}}\\
&\le h^{\frac{\beta+1}{2}}\|u\|_{3,\Omega}^2.
\end{aligned}
\tag{3.3.10}
$$

Hence, we have

$$
\begin{aligned}
|||I_hu-u_h|||^2 \le\, & h^{1+\rho}(\|u\|_{3,\Omega}+|u|_{2,\infty,\Omega})|I_hu-u_0|_{1,\Omega}\\
&+h^{\frac{\beta+1}{2}}\|u\|_{3,\Omega}^2\,|||\,I_hu-u_h|||\\
&\le Ch^{\min\{1+\rho,\frac{\beta+1}{2}\}}(\|u\|_{3,\Omega}+|u|_{2,\infty,\Omega})\,|||\,I_hu-u_h\,|||\,.
\end{aligned}
\tag{3.3.11}
$$

This yields the desired result and completes the proof. □

CHAPTER 4 PPR FOR WGFEMS

As an intermediate product, the weak gradient could be computed and obtained by the weak Galerkin finite element methods. However, we are more interested in the gradient information at the mesh grids. Gradient recovery technique serves this purpose well and provides a better approximation of ∇u.

In a recent work of Wang et. al. in [119], they develped an modified form of PPR scheme. Different from standard FEMs approximation, WG solution is discontinuous across boundary of elements which leads to multiple values of a nodal points. Their strategy is to take an appropriate weighted average to unify these values and then apply the standard PPR scheme for nodal points. In this thesis, we employ the main concept of PPR and generalize it to WGFEMs, and call it by WGPPR. In the rest of this chapter, we will introduce a detailed framework of WGPPR for WGFEMs using (P_0, P_0, RT_0) element and (P_1, P_1, P_0) element.

4.1 WGPPR for (P_0,P_0,RT_0) element

Different from PPR for C^0 finite element method, the sampling points for the vertex z are not vertices anymore in WGPPR. Instead, we take the degree of freedom as assembly points. For (P_0, P_0, RT_0) element, barycenters and edge centers are employed as sampling points. Let $\mathcal{C}_h$ and $\mathcal{M}_h$ denote the set of degree of freedom inside the elements and the set of degree of freedom on the edges, respectively. Denote the set of sampling points for z by $\mathcal{L}_z$, and define it as

$$\mathcal{L}_z = \{\zeta : \zeta \in \mathcal{C}_h \cap \mathcal{K}_z\} \bigcup \{\zeta : \zeta \in \mathcal{M}_h \cap \mathcal{K}_z\}. \tag{4.1.1}$$

Inspired by the idea of PPR for C^0 Lagrange element, we fit a quadratic polynomial on each patch $\mathcal{K}_z$. Define the least-squares fitting polynomial p_z as follows:

$$p_z = \arg\min_{p \in P_2(\mathcal{K}_z)} \sum_{\zeta \in \mathcal{L}_z} |(u_h - p)(\zeta)|^2, \tag{4.1.2}$$

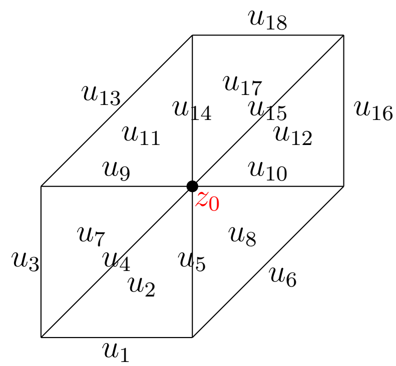

Figure 4.1.1: regular pattern

and define the recovered gradient at z as

$$G_h u_h(z) = \nabla p_z(x, y; z). \tag{4.1.3}$$

To illustrate the idea, we look at the two-dimensional problem and employ the weak Galerkin method defined on a regular mesh using $(P_0(K_0), P_0(F), RT_0(K))$. In other words, the weak Galerkin method uses piecewise constants on both the triangles and the edges. Different from the C^0 Lagrange element, the degree of freedom lies in the triangles and edges. Fig 4.1.1 shows a distribution of the degree of freedom for (P_0, P_0, RT_0) on regular pattern mesh. In what follows, we will demonstrate how to recover the gradient information at z_0, which is not one of the degrees of freedom for WGFEMs.

To avoid computational instability resulting from small h, we introduce the coordinate transformation

$$F : (x, y) \to (\xi, \eta) = \frac{(x - y) - (x_0, y_0)}{h}, \tag{4.1.4}$$

where $h = \max\{|z - \zeta| : \zeta \in \mathcal{L}_z\}$. All computations are then carried out on the local element patch $\hat{\mathcal{K}}_z = F(\mathcal{K}_z)$. Thus, the fitted polynomial can be written as

$$p_z(x, y) = P^T a = \hat{P}^T \hat{a}, \tag{4.1.5}$$

with

$$P^T = (1, x, y, x^2, xy, y^2),\ \hat{P}^T = (1, \xi, \eta, \xi^2, \xi\eta, \eta^2); \tag{4.1.6}$$

$$a^T = (a_0, a_1, a_2, a_3, a_4, a_5),\ \hat{a}^T = (a_0, ha_1, ha_2, h^2a_3, h^2a_4, h^2a_5). \tag{4.1.7}$$

The coefficient vector $\hat{a}$ is then uniquely determined by solving the system

$$A^T A\hat{a} = A^T b \tag{4.1.8}$$

where $b = (u_h(\zeta_1), u_h(\zeta_z), \cdots, u_h(\zeta_m))^T$ and

$$A = \begin{pmatrix} 1 & \xi_0 & \eta_0 & \xi_0^2 & \xi_0\eta_0 & \eta_0^2 \\ 1 & \xi_1 & \eta_1 & \xi_1^2 & \xi_1\eta_1 & \eta_1^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \xi_m & \eta_m & \xi_m^2 & \xi_m\eta_m & \eta_m^2 \end{pmatrix}. \tag{4.1.9}$$

Here m denotes the number of degree of freedom in $\mathcal{L}_z$. Consequently, the recovered gradient at z is given by

$$G_h u_h = \nabla p_z(0, 0; z) = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \frac{1}{h} \begin{pmatrix} \hat{a}_1 \\ \hat{a}_2 \end{pmatrix}. \tag{4.1.10}$$

Remark. In order to solve (4.1.8), $\mathcal{L}_z$ must contain at least 6 points, i.e., $m \geq 6$. This condition is mostly satisfied since two adjoint elements will provide sufficient degree of freedom. For some extreme boundary cases, please refer to Chapter 2 regarding the boundary strategies.

Remark. In the PPR process, if we evaluate the least-squares fitting polynomial $p_2(x, y; z_0)$ alone at the node z_0, we get a recovered function value at node z_0, i.e., $R_h u_h(z_0) = p_2(0, 0; z_0)$. It is worth to point out that WGFEMs provide totally discontinuous solution across elements. With the recovered solution $R_h u_h$, we will actually obtain a global continuous approximation to the exact solution u.

4.1.1 Regular pattern

To demonstrate the above procedure, we apply WGPPR to recover the gradient information at z on the uniform regular pattern mesh (see Fig 4.1.1) in detail. Similar to previous chapters, we use the exact solution u here instead of the WGFEM solution u_h to demonstrate the superconvergence property of WGPPR.

Given

$$\xi = (-\frac{1}{2}, -\frac{1}{3}, -1, -\frac{1}{2}, 0, \frac{1}{2}, -\frac{2}{3}, \frac{1}{3}, -\frac{1}{2}, \frac{1}{2}, -\frac{1}{3}, \frac{2}{3}, -\frac{1}{2}, 0, \frac{1}{2}, 1, \frac{1}{3}, \frac{1}{2})^T;$$
$$\eta = (-1, -\frac{2}{3}, -\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}, -\frac{1}{3}, -\frac{1}{3}, 0, 0, \frac{1}{3}, \frac{1}{3}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{2}{3}, 1)^T;$$
$$b = (u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8, u_9, u_{10}, u_{11}, u_{12}, u_{13}, u_{14}, u_{15}, u_{16}, u_{17}, u_{18})^T;$$

where $u_i = u(F^{-1}(\xi_i, \eta_i))$ for $i = 1, \cdots, 18$. The least square fitting polynomial with respect to (ξ, η) is then in the form of

$$\hat{p}_2(\xi, \eta) = (1, \xi, \eta, \xi^2, \xi\eta, \eta^2)(\hat{a}_0, \hat{a}_1, \hat{a}_2, \hat{a}_3, \hat{a}_4, \hat{a}_5)^T.$$

Let $\vec{e} = (1, 1, 1, 1, 1, 1)^T$, and

$$A = (\vec{e}, \vec{\xi}, \vec{\eta}, \vec{\xi} \circ \vec{\xi}, \vec{\xi} \circ \vec{\eta}, \vec{\eta} \circ \vec{\eta}), \tag{4.1.11}$$

where $\circ$ is the Hadamard product for matricies. Let $S = (A^T A)^{-1} A^T$, then simple calculation yields $\hat{a} = Sb$. Since

$$(\hat{a}_0, \hat{a}_1, \hat{a}_2, \hat{a}_3, \hat{a}_4, \hat{a}_5) = (a_0, ha_1, ha_2, h^2 a_3, h^2 a_4, h^2 a_5),$$

we must have

$$p_2(x, y) = \hat{a}_0 + \frac{1}{h}(\hat{a}_1 x + \hat{a}_2 y) + \frac{1}{h^2}(\hat{a}_3 x^2 + \hat{a}_4 xy + \hat{a} y^2).$$

Differentiating $p_2(x, y)$ with respect to x and y respectively and evaluate at $(0, 0)$, the recovered gradient at a vertex z_0 is given by

$$G_h^x u = \frac{1}{16h}(-3u_3 - u_4 + u_5 + 3u_6 - 2u_7 + 2u_8 - 2u_9 + 2u_{10} - 2u_{11} + 2u_{12} - 3u_{13} - 1u_{14} + u_{15} + 3u_{16}), \tag{4.1.12}$$

and

$$G_h^y u = \frac{1}{16h}(-3u_1 - 2u_2 - u_4 - 2u_5 - 3u_6 - 2u_8 + u_9 - u_{10} + 2u_{11} + 3u_{13} + 2u_{14} + u_{15} + 2u_{17} + 3u_{18}), \tag{4.1.13}$$

where $G_h^x(z)$ and $G_h^y(z)$ are x-component and y-component of $G_h u(z)$ respectively.

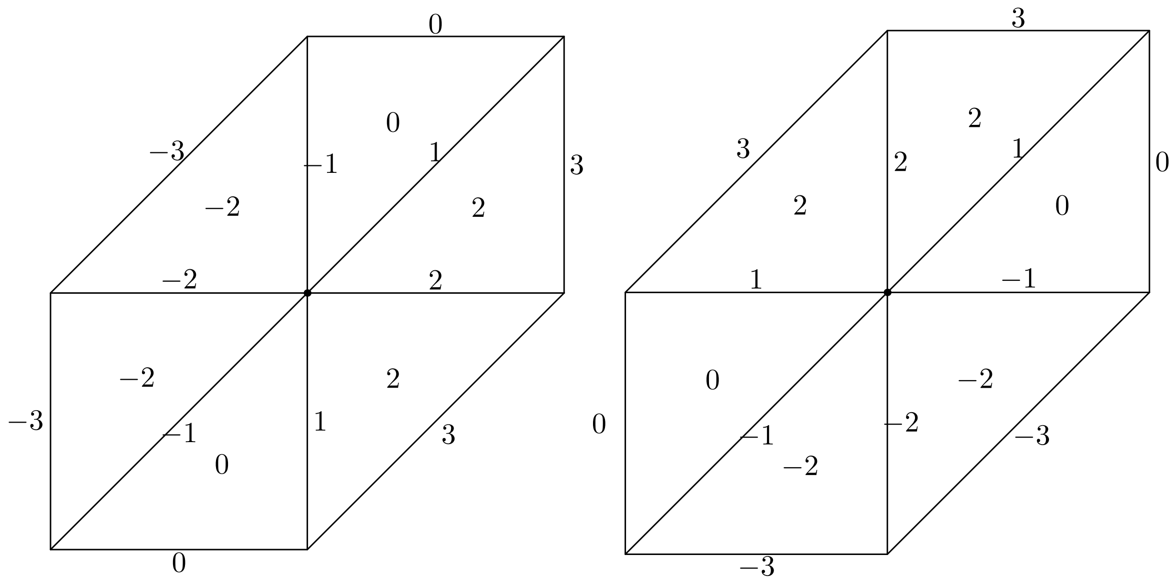

Figure 4.1.2: Regular pattern: (Left) $G_h^x : \frac{1}{16h}$. (Right) $G_h^y : \frac{1}{16h}$.

By using **Mathematica**, we get the following Taylor expansion

$$G_h^x u_h(z) = u^{(1,0)}(z) + \frac{53h^2}{576}(u^{(1,2)}(z) + u^{(2,1)}(z) + u^{(3,0)}(z)) + O(h^4),$$

$$G_h^y u_h(z) = u^{(0,1)}(z) + \frac{53h^2}{576}(u^{(1,2)}(z) + u^{(2,1)}(z) + u^{(0,3)}(z)) + O(h^4),$$

which is a second-order finite difference scheme. With $G_h u$ given at each vertex, the recovered gradient field can be obtained by linear interpolation.

Remark. In the PPR process, if we evaluate the least-squares fitting polynomial $p_2(x, y; z_0)$ alone at the node z_0, we get a recovered function value at node z_0, i.e.,

$$R_h u_h(z_0) = p_2(0, 0; z_0).$$

The following expression can also be derived at the vertex z_0:

$$R_h u = \frac{1}{186}(- 19u_1 + 21u_2 - 19u_3 + 29u_4 + 29u_5 - 19u_6 + 21u_7 + 21u_8 + 29u_9 + 29u_{10} + 21u_{11} + 21u_{12} - 19u_{13} + 29u_{14} + 29u_{15} - 19u_{16} + 21u_{17} - 19u_{18}),$$

and by Taylor Expansion we can obtain the following:

$$R_h u_h(z) = u(z) - \frac{157h^3}{26784}(u^{(0,4)}(z) + 2u^{(1,3)}(z) + 3u^{(2,2)}(z) + 2u^{(3,1)}(z) + u^{(4,0)}(z)) + O(h^6).$$

4.1.2 Chevron pattern

For Chevron pattern mesh, the procedure is the same as regular pattern. At first we can compute

$$\begin{aligned} G_h^x u =& \frac{1}{25390h}(1161u_1 - 1161u_2 - 438u_3 + 438u_4 - 3132u_5 - 1566u_6 \\ &+ 1566u_8 + 3132u_9 - 3300u_{10} + 3300u_{11} - 4293u_{12} \\ &+ 4293u_{13} - 4074u_{14} + 4074u_{15} - 7020u_{16} + 7020u_{18}), \end{aligned} \tag{4.1.14}$$

and

$$\begin{aligned} G_h^y u =& \frac{1}{4979980h}(115743u_1 + 115743u_2 - 809526u_3 - 809526u_4 + 126732u_5 \\ &- 788163u_6 - 1093128u_7 - 788163u_8 + 126732u_9 - 545790u_{10} \\ &- 545790u_{11} - 312909u_{12} - 312909u_{13} + 600702u_{14} + 600702u_{15} \\ &+ 1541505u_{16} + 1236540u_{17} + 1541505u_{18}). \end{aligned} \tag{4.1.15}$$

It is straightforward to verify the following taylor expansion in **Mathematica**:

$$G_h^x u_h(z) = u^{(1,0)}(z) + (\frac{12499}{182808}u^{(1,2)}(z) + \frac{206699}{2742120}u^{(3,0)}(z))h^2 + O(h^3)$$

$$G_h^y u_h(z) = u^{(0,1)}(z) + (\frac{21151817}{537837840} u^{(0,3)}(z) + \frac{2606239}{35855856} u^{(2,1)}(z))h^2 + O(h^3)$$

which are second order difference schemes.

Remark. Similarly, we could obtain the recovered function value at the vertices

$$\begin{aligned} R_h u = \frac{1}{29879880}(& -3157489u_1 - 3157489u_2 + 3209238u_3 + 3209238u_4 - 3174976u_5 \\ & + 3566789u_6 + 5814044u_7 + 3566789u_8 - 3174976u_9 + 2789550u_{10} \\ & + 2789550u_{11} + 4572467u_{12} + 4572467u_{13} + 3314394u_{14} + 3314394u_{15} \\ & - 140455u_{16} + 2106800u_{17} - 140455u_{18}), \end{aligned} \tag{4.1.16}$$

and the taylor expansion gives

$$R_h u_h(z) = u(z) + \frac{h^3}{1075675680}(23133673u^{(0,3)}(z) + 49867995u^{(2,1)}(z)) + O(h^4).$$

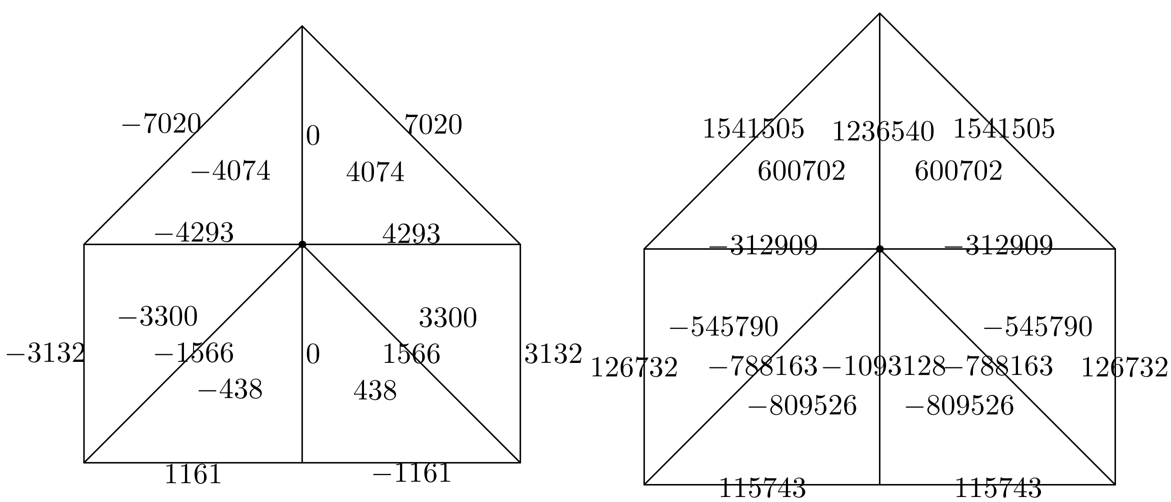

Figure 4.1.3: Chevron pattern: (Left) $G_h^x : \frac{1}{25390h}$. (Right) $G_h^y : \frac{1}{4979980h}$.

4.1.3 Unionjack pattern

Then we consider the Unionjack pattern mesh.

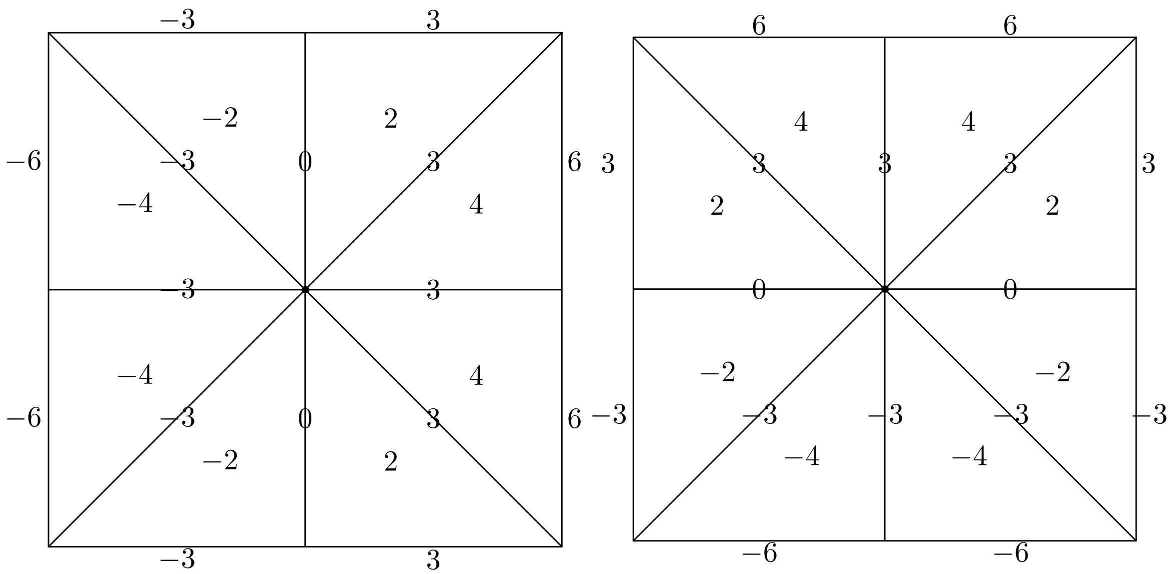

Figure 4.1.4: Unionjack pattern: (Left) $G_h^x : \frac{1}{157h}$. (Right) $G_h^y : \frac{1}{157h}$.

The recovered gradient are computed by

$$\begin{aligned} G_h^x u =& \frac{1}{157h}(-3u_1 + 3u_2 - 2u_3 + 2u_4 - 6u_5 - 3u_6 + 3u_8 + 6u_9 \\ & - 4u_{10} + 4u_{11} - 3u_{12} + 3u_{13} - 4u_{14} + 4u_{15} - 6u_{16} - 3u_{17} + 3u_{19} \\ & + 6u_{20} - 2u_{21} + 2u_{22} - 3u_{23} + 3u_{24}) \end{aligned} \tag{4.1.17}$$

and

$$\begin{aligned} G_h^y u =& \frac{1}{157h}(-6u_1 - 6u_2 - 4u_3 - 4u_4 - 3u_5 - 3u_6 - 3u_7 - 3u_8 - 3u_9 \\ & - 2_{10} - 2u_{11} + 2u_{14} + 2u_{15} + 3u_{16} + 3u_{17} + 3u_{18} + 3u_{19} \\ & + 3u_{20} + 4u_{21} + 4u_{22} + 6u_{23} + 6u_{24}) \end{aligned} \tag{4.1.18}$$

With **Mathematica**, the taylor expansion is given as:

$$G_h^x u_h(z) = u^{(1,0)}(z) + (\frac{857}{5652}u^{(1,2)}(z) + \frac{3541}{33912}u^{(3,0)}(z))h^2 + O(h^3)$$

$$G_h^y u_h(z) = u^{(0,1)}(z) + (\frac{3541}{33912}u^{(0,3)}(z) + \frac{857}{5652}u^{(2,1)}(z))h^2 + O(h^3)$$

which are second order difference schemes.

Remark. Similarly, we could obtain the recovered function value at the vertices

$$R_h u = \frac{1}{27524}(-1810u_1 - 1810u_2 + 2115u_3 + 2115u_4 - 1810u_5 + 2429u_6 + 3842u_7 + 2429u_8 - 1810u_9 + 2115u_{10} + 2115u_{11} + 3842u_{12} + 3842u_{13} + 2115u_{14} + 2115u_{15} - 1810u_{16} + 2429u_{17} + 3842u_{18} + 2429u_{19} - 1810u_{20} + 2115u_{21} + 21150u_{22} - 1810u_{23} - 1810u_{24})$$

and the taylor expansion gives

$$R_h u_h(z) = u(z) + \frac{h^4}{11890368}(-86930u^{(0,4)}(z) - 235137u^{(2,2)}(z) - 86930u^{(4,0)}(z)) + O(h^6).$$

4.1.4 Criss-cross pattern

For Criss-cross pattern, the recovered gradient is computed as

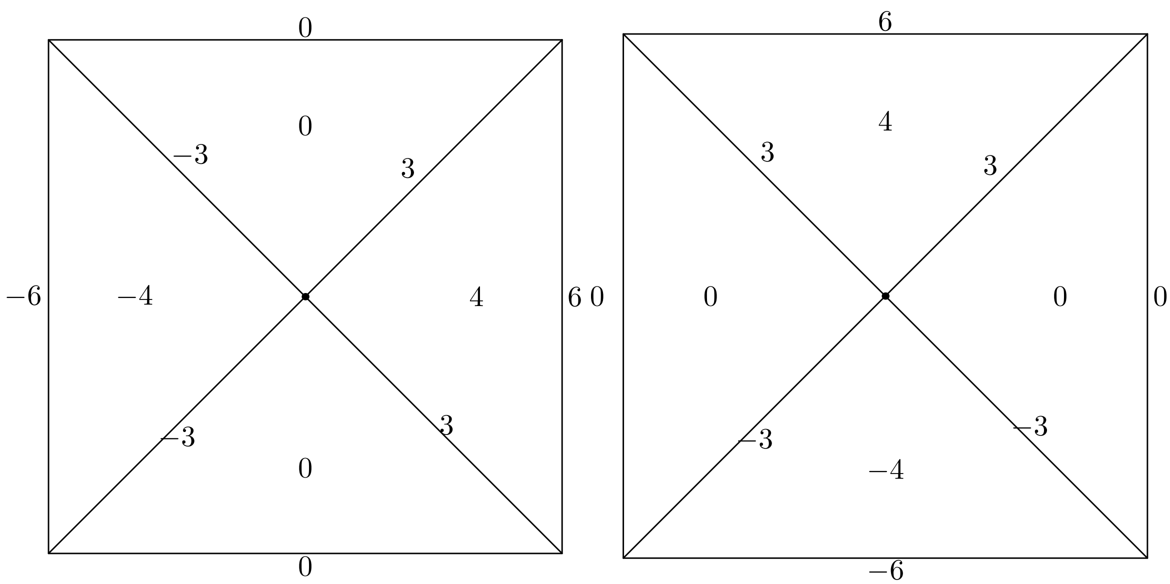

Figure 4.1.5: Criss-cross pattern:(Left) $G_h^x : \frac{3}{70h}$. (Right) $G_h^y : \frac{3}{70h}$.

$$G_h^x u(z) = \frac{3}{70h}(-3(u_3 - u_4 + u_9 - u_{10}) - 6(u_5 - u_8) - 4(u_6 - u_7)),$$

and

$$G_h^y u(z) = \frac{3}{70h}(-3(u_3 + u_4 - u_9 - u_{10}) - 6(u_1 - u_{12}) - 4(u_2 - u_{11})).$$

The taylor expansion gives

$$G_h^x u_h(z) = u^{(1,0)}(z) + (\frac{9}{280}u^{(1,2)}(z) + \frac{857}{7560}u^{(3,0)}(z))h^2 + O(h^4),$$

and

$$G_h^y u_h(z) = u^{(0,1)}(z) + (\frac{857}{7560}u^{(0,3)}(z) + \frac{9}{280}u^{(2,1)}(z))h^2 + O(h^4).$$

This also means $G_h u$ is a second order approximation to ∇u.

Remark. Similarly, the recovered funtion value at the vertices are given by

$$\begin{aligned} R_h u(z) = \frac{1}{104}(&-23(u_1 + u_5 + u_8 + u_{12}) + 27(u_2 + u_6 + u_7 + u_{11}) \\ &+ 22(u_3 + u_4 + u_9 + u_{10})), \end{aligned}$$

and taylor expansion gives

$$R_h u_h(z) = u(z) + \frac{h^4}{14976}(-179u^{(0,4)}(z) + 198u^{(2,2)}(z) - 179u^{(4,0)}(z)) + O(h^6).$$

4.2 WGPPR for (P_1, P_1, P_0) element

Now, let us look at the two-dimensional problem and employ the weak Galerkin method using $(P_1(T), P_1(e), P_0(T))$. That is, WG uses 3 degrees of freedom in each element T and uses 2 degrees of freedoms on each edge. Here we choose the three vertices from each triangle and two gaussian points from each edge, then the total number of degrees of freedom is given by $DOF = 3 * N_T + 2 * N_E$. Since the WG solution u_h is piecewise on each element but continuous across the edges, for each $z \in \mathcal{N}_z$ we choose all the degrees of freedom on the edges to be the sampling points for the PPR process on each patch $\mathcal{K}_z$, i.e.

$$\mathcal{L}_z = \{\zeta : \zeta \in \mathcal{M}_z \cap \mathcal{K}_z\}. \tag{4.2.1}$$

On each patch $\mathcal{K}_z$, we fit a quadratic polynomial

$$p_z = \arg \min_{p \in P_2(\mathcal{K}_z)} \sum_{\zeta \in \mathcal{L}_z} |(u_h - p)(\zeta)|^2, \tag{4.2.2}$$

and define the recovered gradient at z as

$$G_h u_h(z) = \nabla p_z(x, y; z). \tag{4.2.3}$$

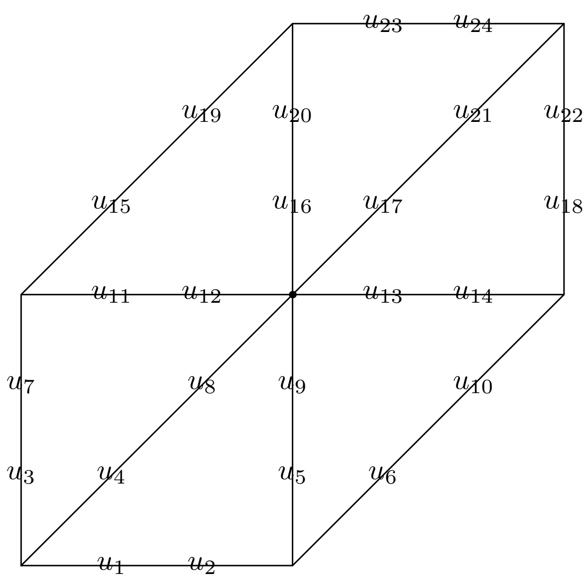

Figure 4.2.1: regular pattern

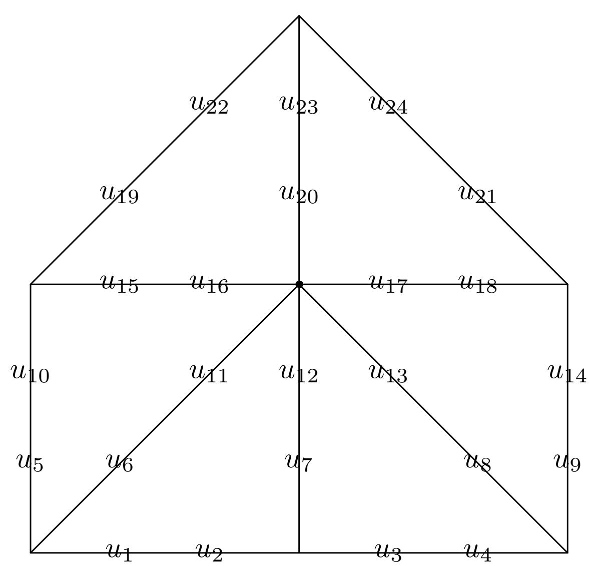

Figure 4.2.2: chevron pattern

To illustrate the idea of choosing sampling points, we still perform WGPPR on the four different types of meshes: reguar pattern, chevron pattern, unionjack pattern and crisscross pattern. The sampling points are displayed in Fig (4.2.1)-(4.2.4), respectively. Again, we will also recover the function value $R_h u_h$ as a byproduct of gradient recovery process since it can be easily recorded from the gradient recovery matrix. Similar to previous section, the exact solution u is used instead of the WGFEM solution u_h.

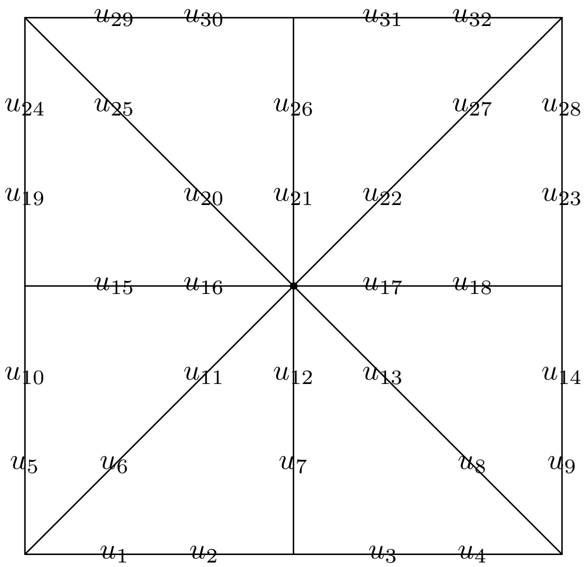

Figure 4.2.3: unionjack pattern

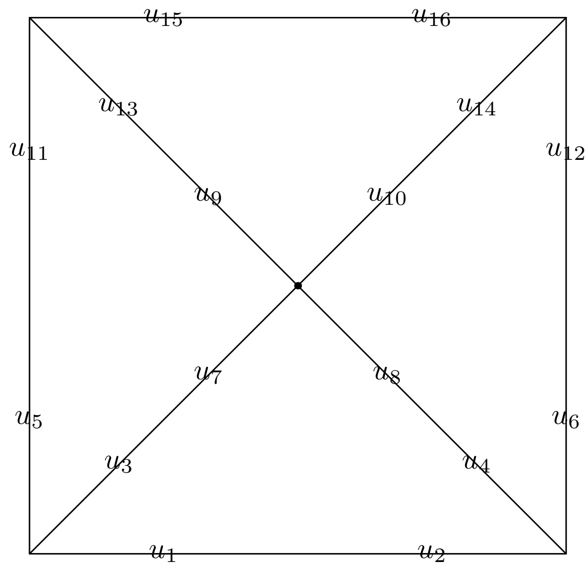

Figure 4.2.4: crisscross pattern

4.2.1 Regular pattern

By using sampling points chosen in Fig (4.2.1) and applying PPR, we can easily obtain the recovered gradient $G_h u$ at a mesh grid z, where

$$
\begin{aligned}
G_h^x u(z) = \frac{1}{84h}(& -2\sqrt{3}u_1 + 2\sqrt{3}u_2 + (-9+\sqrt{3})u_3 + (3+\sqrt{3})u_4 \\
& - (3+\sqrt{3})u_5 + (9-\sqrt{3})u_6 + (-9-\sqrt{3})u_7 + (-3+\sqrt{3})u_8 \\
& + (3-\sqrt{3})u_9 + (9+\sqrt{3})u_{10} - 2(3+\sqrt{3})u_{11} + 2(-3+\sqrt{3})u_{12} \\
& + (6-2\sqrt{3})u_{13} + 2(3+\sqrt{3})u_{14} + (-9-\sqrt{3})u_{15} + (-3+\sqrt{3})u_{16} \\
& + (3-\sqrt{3})u_{17} + (9+\sqrt{3})u_{18} + (-9+\sqrt{3})u_{19} + (-3-\sqrt{3})u_{20} \\
& + (3+\sqrt{3})u_{21} + (9-\sqrt{3})u_{22} - 2\sqrt{3}u_{23} + 2\sqrt{3}u_{24}),
\end{aligned}
$$

and

$$G_h^y u(z) = \frac{1}{84h}((-9+\sqrt{3})u_1 + (-9-\sqrt{3})u_2 - 2\sqrt{3}u_3 - 2(3+\sqrt{3})u_4$$
$$- (3+\sqrt{3})u_5 - (9+\sqrt{3})u_6 + 2\sqrt{3}u_7 + (-3+\sqrt{3})u_8$$
$$+ 2(-3+\sqrt{3})u_9 + (-9+\sqrt{3})u_{10} + (3+\sqrt{3})u_{11} + (3-\sqrt{3})u_{12}$$
$$+ (-3+\sqrt{3})u_{13} - (3+\sqrt{3})u_{14} + (9-\sqrt{3})u_{15} + (6-2\sqrt{3})u_{16}$$
$$+ (3-\sqrt{3})u_{17} - 2\sqrt{3}u_{18} + (9+\sqrt{3})u_{19} + (6+2\sqrt{3})u_{20}$$
$$+ (3+\sqrt{3})u_{21} + 2\sqrt{3}u_{22} + (9+\sqrt{3})u_{23} + (9-\sqrt{3})u_{24}).$$

The recovered function value at node z is given by

$$R_h u(z) = \frac{1}{1080}(-18(u_1 + u_2 + u_3 + u_6 + u_7 + u_{10}$$
$$+ u_{15} + u_{18} + u_{19} + u_{22} + u_{23} + u_{24})$$
$$+ (108 - 42\sqrt{3})(u_4 + u_5 + u_{11} + u_{14} + u_{20} + u_{21})$$
$$+ (108 + 42\sqrt{3}(u_8 + u_9 + u_{12} + u_{13} + u_{16} + u_{17}).$$

By using computer algebra system **Mathematica**, it is easy to verify the following Taylor expansion

$$R_h u(z) = u^{(0,0)}(z) - \frac{11h^4}{6480}(u^{(0,4)}(z) + 2u^{(1,3)}(z) + 3u^{(2,2)}(z) + 2u^{(3,1)}(z) + u^{(4,0)}(z)) + O(h^5),$$

$$G_h^x u(z) = u^{(1,0)}(z) + \frac{8h^2}{63}(u^{(1,2)}(z) + u^{(2,1)}(z) + u^{(3,0)}(z)) + O(h^4),$$

$$G_h^y u(z) = u^{(0,1)}(z) + \frac{8h^2}{63}(u^{(1,2)}(z) + u^{(2,1)}(z) + u^{(0,3)}(z)) + O(h^4),$$

which is a second-order finite difference scheme.

4.2.2 Chevron pattern

Similar to regular pattern, we choose the sampling points as in Fig (4.2.2) and the recovered gradient is then computed as

$$
\begin{aligned}
G_h^x u(z) = \frac{1}{232h}&((3+\sqrt{3})(u_1-u_4)+(3-\sqrt{3})(u_2-u_3)\\
&+(-21+9\sqrt{3})(u_5-u_9)+(-6+\sqrt{3})(u_6-u_8)\\
&+(-21-9\sqrt{3})(u_{10}-u_{14})+(-6-\sqrt{3})(u_{11}-u_{13})\\
&+(-24-8\sqrt{3})(u_{15}-u_{18})+(-24+8\sqrt{3})(u_{16}-u_{17})\\
&+(-33-8\sqrt{3})(u_{19}-u_{21})+(-33+8\sqrt{3})(u_{22}-u_{24})),
\end{aligned}
$$

and

$$
\begin{aligned}
G_h^y u(z) = \frac{1}{43606}&((-2676+729\sqrt{3})(u_1+u_4)+(-2676-729\sqrt{3})(u_2+u_3)\\
&+(84-329\sqrt{3})(u_5+u_9)+(-2832+400\sqrt{3})(u_6+u_8)\\
&+(84-329\sqrt{3})(u_{10}+u_{14})+(-2832-400\sqrt{3})(u_{11}+u_{13})\\
&+(-4290-329\sqrt{3})u_7+(-4290+329\sqrt{3})u_{12}\\
&+(-702+729\sqrt{3})(u_{15}+u_{18})+(-702-729\sqrt{3})(u_{16}+u_{17})\\
&+(6000-1886\sqrt{3})(u_{19}+u_{21})+(4542-2615\sqrt{3})u_{20})\\
&+(6000+1886\sqrt{3})(u_{22}+u_{24})+(4542+2615\sqrt{3})u_{23}).
\end{aligned}
$$

The recovered function value is given as

$$\begin{aligned}
R_h u(z) = \frac{1}{174424}&((-1174 - 4031\sqrt{3})(u_1 + u_4) + (-1174 + 4031\sqrt{3})(u_2 + u_3) \\
&+ (-3874 - 2996\sqrt{3})(u_5 + u_9) + (12250 - 7027\sqrt{3})(u_6 + u_8) \\
&+ (-3874 + 2996\sqrt{3})(u_{10} + u_{14}) + (12250 + 7027\sqrt{3})(u_{11} + u_{13}) \\
&+ (20312 - 2996\sqrt{3})u_7 + (20312 + 2996\sqrt{3})u_{12} \\
&+ (16802 - 4031\sqrt{3})(u_{15} + u_{18}) + (16802 + 4031\sqrt{3})(u_{16} + u_{17}) \\
&+ (3610 + 1845\sqrt{3})(u_{19} + u_{21}) + (11672 + 5876\sqrt{3})u_{20}) \\
&+ (3610 - 1845\sqrt{3})(u_{22} + u_{24}) + (11672 - 5876\sqrt{3})u_{23}).
\end{aligned}$$

It is straightforward to verify that

$$R_h u(z) = u^{(0,0)}(z) + \frac{h^3}{523272}(3420u^{(0,3)}(z) + 36971u^{(2,1)}(z) + O(h^4),$$

$$G_h^x u(z) = u^{(1,0)}(z) + \frac{h^2}{2088}(66u^{(1,2)}(z) + 251u^{(3,0)}(z)) + O(h^3),$$

$$G_h^y u(z) = u^{(0,1)}(z) + \frac{7h^2}{784908}(7890u^{(2,1)}(z) + 12577u^{(0,3)}(z)) + O(h^3).$$

And this provides a second order approximation to ∇u.

4.2.3 Unionjack pattern

For unionjack pattern, the sampling points are displayed in Fig (4.2.3). The recovered gradient is given by

$$\begin{aligned}
G_h^x u(z) = \frac{1}{88h}&((-3 - \sqrt{3})(u_1 + u_6 + u_{15} + u_{25} + u_{29}) \\
&+ (-3 + \sqrt{3})(u_2 + u_{11} + u_{16} + u_{20} + u_{30}) \\
&+ (3 - \sqrt{3})(u_3 + u_{13} + u_{17} + u_{22} + u_{31}) \\
&+ (3 + \sqrt{3})(u_4 + u_8 + u_{18} + u_{27} + u_{32}) \\
&- 6(u_5 + u_{10} + u_{19} + u_{24}) + 6(u_9 + u_{14} + u_{23} + u_{28})),
\end{aligned}$$

and

$$G_h^y u(z) = \frac{1}{88h}(-6(u_1 + u_2 + u_3 + u_4) + 6(u_{29} + u_{30} + u_{31} + u_{32}))$$
$$(-3 - \sqrt{3})(u_5 + u_6 + u_7 + u_8 + u_9)$$
$$+ (-3 + \sqrt{3})(u_{10} + u_{11} + u_{12} + u_{13} + u_{14})$$
$$+ (3 - \sqrt{3})(u_{19} + u_{20} + u_{21} + u_{22} + u_{23})$$
$$+ (3 + \sqrt{3})(u_{24} + u_{25} + u_{26} + u_{27} + u_{28}).$$

The recovered function value is given by

$$R_h u(z) = \frac{1}{1536}((-7 - 22\sqrt{3})(u_1 + u_4 + u_5 + u_9 + u_{24} + u_{28} + u_{29} + u_{32})$$
$$+ (-7 + 22\sqrt{3})(u_2 + u_3 + u_{10} + u_{14} + u_{19} + u_{23} + u_{30} + u_{31})$$
$$+ (81 - 44\sqrt{3})(u_6 + u_8 + u_{25} + u_{27})$$
$$+ (81 + 44\sqrt{3})(u_{11} + u_{13} + u_{20} + u_{22})$$
$$+ (125 - 22\sqrt{3})(u_7 + u_{15} + u_{18} + u_{26})$$
$$+ (125 + 22\sqrt{3})(u_{12} + u_{16} + u_{17} + u_{21})).$$

By using **Mathematica**, it is easy to verify the following Taylor expansion

$$R_h u(z) = u^{(0,0)}(z) + \frac{h^4}{36864}(-49u^{(0,4)}(z) - 1228u^{(2,2)}(z) - 49u^{(4,0)}(z)) + O(h^5),$$

$$G_h^x u(z) = u^{(1,0)}(z) + \frac{h^2}{792}(186u^{(1,2)}(z) + 107u^{(3,0)}(z)) + O(h^4),$$

$$G_h^y u(z) = u^{(0,1)}(z) + \frac{h^2}{792}(107u^{(0,3)}(z) + 186u^{(2,1)}(z)) + O(h^4).$$

which again is second order convergence to ∇u.

4.2.4 Criss-cross pattern

The sampling points for criss-cross pattern as shown in Fig (4.2.4). Following the same procedure as previously, we obtain the recovered gradient as

$$\begin{aligned}G_h^x u(z) = \frac{1}{48h}(&-2\sqrt{3}(u_1 - u_2 + u_{15} - u_{16}) - 6(u_5 - u_6 + u_{11} - u_{12})\\ &- (3+\sqrt{3})(u_3 - u_4 + u_{13} - u_{14})\\ &- (3-\sqrt{3})(u_7 - u_8 + u_9 - u_{10})),\end{aligned}$$

and

$$\begin{aligned}G_h^y u(z) = \frac{1}{48h}(&-2\sqrt{3}(u_5 + u_6 - u_{11} - u_{12}) - 6(u_1 + u_2 - u_{15} - u_{16})\\ &- (3+\sqrt{3})(u_3 + u_4 - u_{13} - u_{14})\\ &- (3-\sqrt{3})(u_7 + u_8 - u_9 - u_{10})).\end{aligned}$$

The recovered function value is

$$\begin{aligned}R_h u(z) = \frac{1}{80}(&-(u_1 + u_2 + u_5 + u_6 + u_{11} + u_{12} + u_{15} + u_{16})\\ &+ (11 - 6\sqrt{3})(u_3 + u_4 + u_{13} + u_{14})\\ &+ (11 + 6\sqrt{3})(u_7 + u_8 + u_9 + u_{10})).\end{aligned}$$

The taylor expansion obtained from **Mathematica** are:

$$\begin{aligned}R_h u(z) &= u^{(0,0)}(z) + \frac{h^4}{2880}(-5u^{(0,4)}(z) - 14u^{(2,2)}(z) - 5u^{(4,0)}(z)) + O(h^5),\\ G_h^x u(z) &= u^{(1,0)}(z) + \frac{h^2}{72}(19u^{(1,2)}(z) + 9u^{(3,0)}(z)) + O(h^4),\\ G_h^y u(z) &= u^{(0,1)}(z) + \frac{h^2}{72}(9u^{(0,3)}(z) + 19u^{(2,1)}(z)) + O(h^4).\end{aligned}$$

This verifies that $G_h u$ is a second order appoximation to ∇u.

4.3 Property of the Gradient Recovery Operator

Theorem 4.3.1. *The gradient recovery operator G_h preserves polynomial up to second order.*

Proof. Suppose $\vec{z}_1, \vec{z}_2, \cdots, \vec{z}_n$ are all the sampling points. Let $b_0(\vec{z}), b_1(\vec{z}), \cdots, b_5(\vec{z})$ be a basis of $\mathcal{P}_2(\mathcal{K}_z)$. Then the least square fitting is to find

$$p_z = \arg \min_{p \in P_2(\mathcal{K}_z)} \sum_{i=1}^{n} |(u(\vec{z}_i) - p(\vec{z}_i))|^2, \tag{4.3.1}$$

Without loss of generality, let $p = \alpha_0 b_0(\vec{z}) + \alpha_1 b_1(\vec{z}) + \cdots + \alpha_5 b_5(\vec{z})$, then it is suffice to find $\vec{\alpha} = (\alpha_0, \alpha_1, \cdots, \alpha_5)$. Let

$$A = \begin{pmatrix} b_0(\vec{z}_1) & b_1(\vec{z}_1) & b_2(\vec{z}_1) & b_3(\vec{z}_1) & b_4(\vec{z}_1) & b_5(\vec{z}_1) \\ b_0(\vec{z}_2) & b_1(\vec{z}_2) & b_2(\vec{z}_2) & b_3(\vec{z}_2) & b_4(\vec{z}_2) & b_5(\vec{z}_2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ b_0(\vec{z}_n) & b_1(\vec{z}_n) & b_2(\vec{z}_n) & b_3(\vec{z}_n) & b_4(\vec{z}_n) & b_5(\vec{z}_n) \end{pmatrix} \tag{4.3.2}$$

and

$$F = (f(\vec{z}_1), \cdots, f(\vec{z}_n)^T,$$

then it is equivalent to solve the linear matrix system

$$A^T A \vec{\alpha} = A^T F.$$

Let $f = b_j(\vec{z}), 0 \le j \le 5$, then it is easy to see that $\alpha = e_j$, which also implies $p = b_j$. Thus the least-square process preserves polynomial up to second order and we have $G_h p = \nabla p$. □

Using the polynomial preserving property above, we can show the following approximation theorem.

Theorem 4.3.2. *Suppose $u \in H^3(\mathcal{K}_z)$, then we have*

$$\|G_h u - \nabla u\|_{0,\mathcal{K}_z} \leq ch^2 \|u\|_{3,\mathcal{K}_z}.$$

Proof. Define $F(u) = \|G_h u - \nabla u\|_{0,\mathcal{K}_z}$, it is easy to see

$$\begin{aligned} F(u) &\leq \|G_h u\|_{0,\mathcal{K}_z} + \|\nabla u\|_{0,\mathcal{K}_z} \\ &\leq c\|\nabla u\|_{0,\mathcal{K}_z} \\ &\leq c\|u\|_{1,\mathcal{K}_z}. \end{aligned}$$

The polynomial property of the gradient recovery operator G_h implies $G_h p = \nabla p$ for any $p \in \mathcal{P}_2(\mathcal{K}_z)$. Thus we have $F(u+p) = F(u)$. By the Brambler-Hilbert Lemma, we obtain $F(u) \leq ch^2\|u\|_{3,\mathcal{K}_z}$. □

4.4 Numerical Examples

Consider the Laplace equation with zero boundary condition on unit square $\Omega = [0,1] \times [0,1]$ and the exact solution is given by $u = \sin(\pi x)\sin(\pi y)$.

We will use four different triangular mesh: regular pattern, chevron pattern, union-jack pattern and criss-cross pattern. The weak Galerkin method with the element $(P_0(K), P_0(F), RT_0(K))$ is employed to solve the problem. The convergence curves of the L^2 error of the recovered gradient and weak gradient are depicted in Fig 4.4.1 – 4.4.4. From the plots, we can clearly observe the optimal convergence rate for the weak gradient at $O(h)$, and superconvergence for the recovered gradient at approximately $O(h^2)$ on these four different meshes. We also test the second approach on the same mesh and superconvergence phenomenon is again observed for the recovered gradient in all figures.

Next, we employ a uniform triangular mesh with regular pattern and (P_1, P_1, P_0) element is used in the weak Galerkin discretization.

In Table 4.4.1, we compare the L_2 error of the weak gradient and the recovered gra-

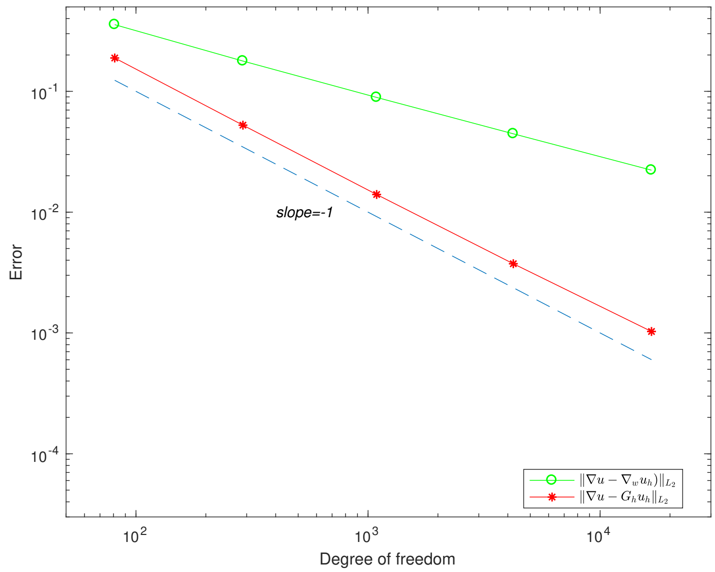

Figure 4.4.1: Regular mesh: $||\nabla u - G_h u_h||_{L_2(\Omega)}$ and $||\nabla u - \nabla_w u_h||_{L_2(\Omega)}$

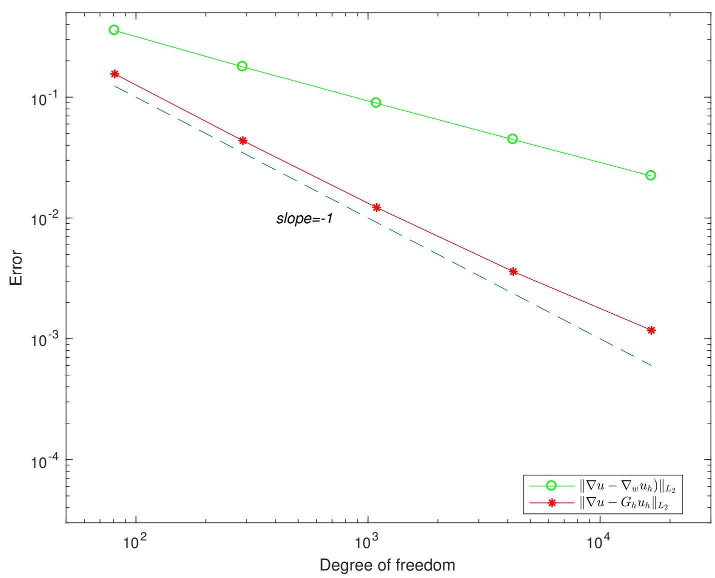

Figure 4.4.2: Chevron mesh: $||\nabla u - G_h u_h||_{L_2(\Omega)}$ and $||\nabla u - \nabla_w u_h||_{L_2(\Omega)}$

dient. The weak gradient converges at the rate of $O(h)$ as expected while the recovered gradient superconverges at the order of $O(h^2)$ which matches our theoretical results. In order to track the behavor of PPR near boundary, we split the domain Ω into interior domain and boundary domain, denote by Ω_1 and Ω_2 as defined in (2.3.1) and (2.3.2), respectively. The nuemrical results are shown in Table 4.4.2, it is not suprise that we

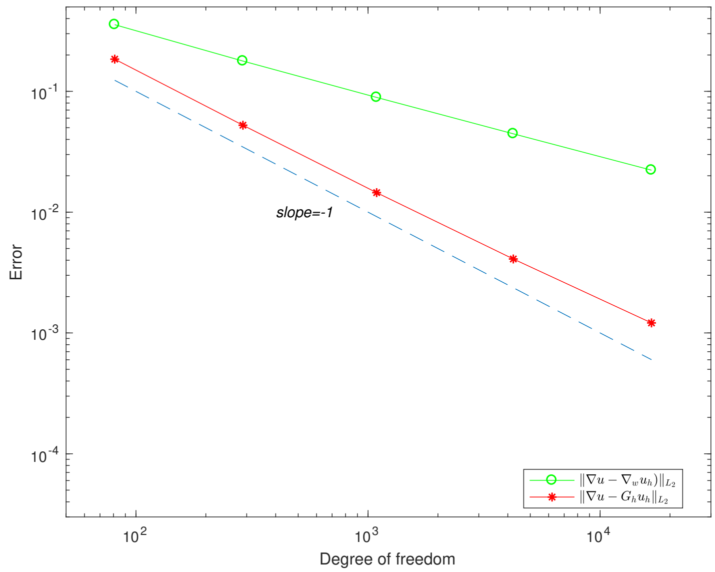

Figure 4.4.3: Unionjack mesh: $||\nabla u - G_h u_h||_{L_2(\Omega)}$ and $||\nabla u - \nabla_w u_h||_{L_2(\Omega)}$

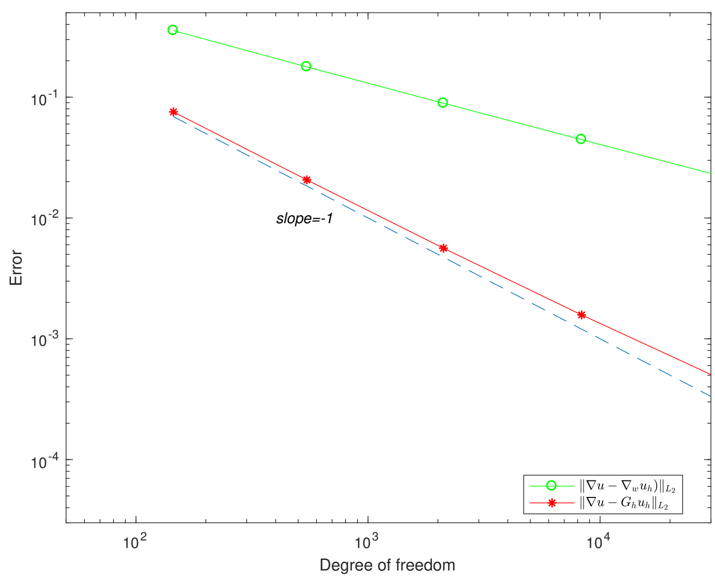

Figure 4.4.4: Criss-cross mesh: $||\nabla u - G_h u_h||_{L_2(\Omega)}$ and $||\nabla u - \nabla_w u_h||_{L_2(\Omega)}$

obtain superconvergent results in both interior domain and boundary domain which again validates the effectiveness and robustness of our boundary strategies.

Table 4.4.1: Poisson Problem: WG using $(P_1(T), P_1(e), P_0(T))$

$1/h$	$\|\nabla_w u_h - \nabla u\|_\Omega$	order	$\|G_h u_h - \nabla u\|_\Omega$	order
8	6.8908e-01	–	1.8434e-01	–
16	3.4608e-01	0.9935	4.9900e-02	1.8852
32	1.7323e-01	0.9984	1.2916e-02	1.9499
64	8.6641e-02	0.9996	3.2915e-03	1.9723
128	4.3323e-02	0.9999	8.3599e-04	1.9772
256	2.1662e-02	1.0000	2.1335e-04	1.9702

Table 4.4.2: Poisson Problem: Interior error vs Boundary error

$1/h$	$\|G_h u_h - \nabla u\|_{\Omega_1}$	order	$\|G_h u_h - \nabla u\|_{\Omega_2}$	order
8	1.4609e-01	–	1.1242e-01	–
16	3.7166e-02	1.9748	3.3297e-02	1.7555
32	9.3394e-03	1.9926	8.9215e-03	1.9000
64	2.4710e-03	1.9183	2.1744e-03	2.0366
128	6.3434e-04	1.9617	5.4452e-04	1.9976
256	1.5860e-04	1.9999	1.4271e-04	1.9319

CHAPTER 5 WGPPR FOR INTERFACE PROBLEM

Let Ω be a bounded polygonal domain with Lipschitz boundary $\partial\Omega$ in $\mathbb{R}^2$. A curve $\Gamma \in C^2$ divides Ω into two disajoint subdomains Ω^- and Ω^+. The interface curve Γ is often characterized by zero level set of some level set function ϕ [120, 121], therefore we have $\Omega^- = \{z \in \Omega | \phi(z) \leq 0\}$ and $\Omega^+ = \{z \in \Omega | \phi(z) \geq 0\}$. Consider the following elliptic interface problem

$$\begin{cases} -\nabla \cdot (\beta(z)\nabla u(z)) = f(z), \quad z \text{ in } \Omega \setminus \Gamma, \\ \\ u = g, \quad z \text{ on } \partial\Omega; \end{cases} \tag{5.0.1}$$

where the diffusion coefficient $\beta(z) \geq \beta_0$ is a piecewise smooth function defined as

$$\beta(z) = \begin{cases} \beta^-(z), & \text{if } z \in \Omega^-, \\ \\ \beta^+(z), & \text{if } z \in \Omega^+, \end{cases}$$

which has a finite jump of function value across the interface Γ. The source term $f(z)$ may be singular at the interface and is defined by

$$f(z) = \begin{cases} f^-(z), & \text{if } z \in \Omega^-, \\ \\ f^+(z), & \text{if } z \in \Omega^+, \end{cases}$$

The dirichlet boundary condition is also defined by

$$g(z) = \begin{cases} g^-(z), & \text{on } \partial\Omega^- \setminus \Gamma, \\ \\ g^+(z), & \text{on } \partial\Omega^+ \setminus \Gamma. \end{cases}$$

The elliptic interface problem is otherwise unsolvable unless it is supplemented from the underlying physics with two jump conditions across the interface Γ: $[u]_\Gamma = u^+ - u^- = \phi$ and $[\beta u_n]_\Gamma = \beta^+ u_n^+ - \beta^- u_n^- = \psi$ where u_n denotes the normal flux $\nabla u \cdot n$ with n being the unit outer normal vector of the interface Γ.

5.1 The weak Galerkin scheme

Standard notations for Sobolev spaces and their associate norms given in chapter 3 are adopted again in this section. Furthermore, we denote $W^{k,p}(\Omega^- \cap \Omega^+)$ as the function space consisting of piecewise function w such that $w|_{\Omega^-} \in W^{k,p}(\Omega^-)$ and $w|_{\Omega^+} \in W^{k,p}(\Omega^+)$. For the function space $W^{k,p}(\Omega^- \cup \Omega^+)$, the associated norm is defined as $\|w\|_{k,p,\Omega^-\cup\Omega^+} = (\|w\|^p_{k,p,\Omega^-} + \|w\|^p_{k,p,\Omega^+})^{1/p}$, and the seminorm is defined as $|w|_{k,p,\Omega^-\cup\Omega^+} = (|w|^p_{k,p,\Omega^-} + |w|^p_{k,p,\Omega^+})^{1/p}$.

Let $\mathcal{T}_h$ be a body-fitted triangulation of Ω. For each triangle $T \in \mathcal{T}_h$, it can be classified into the following three types:

1. $T \subset \overline{\Omega^-}$;

2. $T \subset \overline{\Omega^+}$;

3. $T \cap \Omega^- \neq \emptyset$ and $T \cap \Omega^+ \neq \emptyset$, then the two vertices of T lie on the interface Γ.

Denote Γ_h as an approximation of Γ which consists of the edges with both endpoints lying on Γ. The domain Ω is now divided into two parts $\Omega_{1,h}$ and $\Omega_{2,h}$, which are the approximation of Ω_1 and Ω_2, respectively. We can now define

$$\mathcal{T}_h^- := \{T \in \mathcal{T}_h |\ \text{all three vertices of T are in } \overline{\Omega^-}\},$$

$$\mathcal{T}_h^+ := \{T \in \mathcal{T}_h |\ \text{all three vertices of T are in } \overline{\Omega^+}\},$$

$$\mathcal{T}_h^0 := \{T \in \mathcal{T}_h |\ \text{T has two vertices on } \Gamma\}.$$

For simplicity, we denote $(v,w)_T := \int_T vwdT, \quad < v, w >_{\partial T} = \int_{\partial T} vwds$. For each triangle $T \in \mathcal{T}_h$, let T^0 and ∂T denote the interior and boundary of T respectively. Denote by $\mathbb{P}_j(T^0)$ the set of polynomials in T^0 with degree no more than j, and $P_l(e)$ the set of polynomials on each segment(edge of face) e, $e \in \partial T$ with degree no more than l. A discrete function $w = \{w_0, w_b\}$ refers to a polynomial with two components in which the first component w_0 is associated with the interior T^0 and w_b is defined on each edge or face e, $e \in \partial T$. Please note that w_b may or may not equal to w_0 on ∂T. Now we introduce three trial finite element spaces as follows:

$$V_h^- := \{w = \{w_0, w_b\} : \{w_0, w_b\}|_T \in P_j(T^0) \times P_l(e), e \in \partial T, \forall T \in \mathcal{T}_h^-\},$$

$$V_h^+ := \{\rho = \{\rho_0, \rho_b\} : \{\rho_0, \rho_b\}|_T \in P_j(T^0) \times P_l(e), e \in \partial T, \forall T \in \mathcal{T}_h^+\},$$

$$\Lambda_h := \{\mu : \mu|_e \in P_m(e), e \in \Gamma_h\}.$$

Define two test spaces by

$$V_h^{0,-} = \{w = \{w_0, w_b\} \in V_h^- : w_b|_e = 0,\ e \in \partial\Omega^- \setminus \Gamma\},$$

$$V_h^{0,+} = \{\rho = \{\rho_0, \rho_b\} \in V_h^+ : \rho_b|_e = 0,\ e \in \partial\Omega^+ \setminus \Gamma\}.$$

For each $w = \{w_0, w_b\} \in V_h^-$ or V_h^+, the discrete gradient of w, denoted by $\nabla_d w \in V_r(T)$ on each element T, is defined by the following equation:

$$\int_T \nabla_d w \cdot q dT = -\int_T w_0(\nabla \cdot q) dT + \int_{\partial T} w_b (q \cdot \mathbf{n}) ds, \forall q \in V_r(T),$$

where $V_r(T)$ is a subspace of the set of vector-valued polynomials of degree no more than r on T.

The selection of the indices j, l, m, and r is critical in the design of weak Galerkin finite element methods. Please refer to [104] for a detailed discussion on the selection of those indices. In this part for interface problem, let $j = l = m = k \geq 0$ and choose the Raviart-Thomas element for $V_r(T) := RT_k(T)$. These elements are referred as $\{P_k(T^0)^2, P_k(e)^2, P_k(\Gamma)\}$ element in the numercial test. Recall that the Raviart-Thomas element $RT_k(K)$ of order k is of the following form $RT_k(T) = P_k(T)^2 + \tilde{P}_k(T)x$, where $\tilde{P}_k(T)$ is the set of homogeneous polynomials of degree k and $x = (x_1, x_2)$.

A numerical approximation of the model problem can be obtained by seeking $u_h =$

$\{u_0, u_b\} \in V_h = V_h^- \cup V_h^+$ satisfying $u_b = Q_b g$, and $\lambda_h \in \Lambda_h$ such that

$$
\begin{aligned}
(A\nabla_d u_h^-, \nabla_d w) - < \lambda_h, w_b >_\Gamma &= (f^-, w_0),\ \forall w \in V_h^{0,-} \\
(A\nabla_d u_h^+, \nabla_d \rho) + < \lambda_h, \rho_b >_\Gamma &= (f^+, \rho_0) + < \psi, \rho_b >_\Gamma,\ \forall \rho \in V_h^{0,+} \\
< u_b^- - u_b^+, \mu >_\Gamma &= < \phi, \mu >_\Gamma,\ \forall \mu \in \Lambda_h.
\end{aligned}
\tag{5.1.1}
$$

Here $Q_b g$ is the standard L^2 projection of the Dirichlet boundary data in $P_k(e)$ for any edge/face $e \in \partial\Omega$.

Denote by $Q_h = \{Q_0, Q_b\}$ a local L^2 projection operator where $Q_0 : H^1(T^0) \to P_k(T^0)$, and $Q_b : H^{\frac{1}{2}}(e) \to P_k(e), e \in \partial T$ are the usual L^2 projections into the corresponding spaces. The following error estimates hold true [116].

Theorem 5.1.1. *Let $(u_h, \lambda_h) \in V_h \times \Lambda_h$ be the solution arising from the weak Galerkin finite element scheme. Then,*

$$||\nabla_d(Q_h u^- - u_h^-)|| + ||\nabla_d(Q_h u^+ - u_h^+)|| \lesssim h^{k+1}(||u^-||_{k+2} + ||u^+||_{k+2}), \tag{5.1.2}$$

$$||A\nabla u \cdot n - \lambda_h||_\Gamma \lesssim h^{k+\frac{1}{2}}(||u^-||_{k+2} + ||u^+||_{k+2}). \tag{5.1.3}$$

5.2 WGPPR for Interface Problems

The standard PPR process works as a smoothing operator since it provides continuous gradient approximation to ∇u. Due to the discontinuity of ∇u across the interface, the original flavor of PPR will not work as expected for the elliptic interface problem. In practice, the two components of u ($u|_{\Omega_h^-}$ and $u|_{\Omega_h^+}$) are smooth in their corresponding domain, even though u has low global regularity due to the effect of the interface. This motivates us to recover $u|_{\Omega_h^-}$ and $u|_{\Omega_h^+}$ in Ω_h^- and Ω_h^+ separately and consequently a piecewise continuous gradient approximation could be obtained and yet a good approximation to ∇u.

Let $G_h^- : V_h^- \to S_h^- \times S_h^-$ and $G_h^+ : V_h^+ \to S_h^+ \times S_h^+$ be the PPR gradient recovery operator defined on S_h^- and S_h^+, respectively. For any $u_h \in S_h$, we define the global gradient recovery operator $G_h : V_h \to (S_h^- \cup S_h^+) \times (S_h^- \cup S_h^+)$ as

$$(G_h u_h)(z) = \begin{cases} (G_h^- u_h)(z), & \text{if } z \in \overline{\Omega_h^-}, \\ (G_h^+ u_h)(z), & \text{if } z \in \overline{\Omega_h^+}, \end{cases} \tag{5.2.1}$$

Remark. If z is away from the interface Γ_h, $(G_h u_h)(z)$ is the stardard WGPPR process at z in its corresponding domain.

Remark. If z is near the interface Γ_h, $(G_h u_h)(z)$ is computed by fitting a quadratic polynomial in the least-squares sense that only employs sampling points from either $\mathcal{T}_h^-$ or $\mathcal{T}_h^+$.

Remark. If z is on the interface Γ_h, $(G_h u_h)(z)$ will be computed in $\mathcal{T}_h^-$ and $\mathcal{T}_h^+$ seperately, yet for all points on interface, there are two values of the gradient: $(G_h^- u_h)(z)$ and $(G_h^+ u_h)(z)$.

Remark. Furthermore, if we perform the function recovery of z in a similar way, which in fact can be recorded from PPR process, and denote the value of recovered function by $R_h^- u_h$ and $R_h^+ u_h$, respectively. The jump of u across interface Γ can be captured by the difference between $R_h^- u_h$ and $R_h^+ u_h$.

5.3 Numerical Examples

In this section, we present several numerical examples to verify the robustness and superconvergence of the function recovery and gradient recovery algorithms. The computational domain of our examples are chosen as $\Omega = [-1, 1] \times [-1, 1]$. Note that all convergence rate will be computed against the degree of freedom (Dof), and since $Dof \approx h^{-2}$ for a two-dimensional quasi-uniform mesh, the corresponding convergence rate in mesh size h is twice as much as what we present in the tables.

Example 1. The interface problem is defined in a square $[-1, 1] \times [-1, 1]$ with a circular interface $r^2 = x^2 + y^2 = \frac{1}{4}$. The analytical solution to the equation, the coefficient β, and the inhomogeneous term of the equation are given as follows

$$
\begin{aligned}
u(x,y) &= \begin{cases} x^2+y^2-1, & \text{if } r \le 0.5, \\ \frac{1}{4}(1-\frac{1}{8b}-\frac{1}{b}) + (\frac{r^4}{2}+r^2)/b, & \text{otherwise} \end{cases} \\
\beta(x,y) &= \begin{cases} 2, & \text{if } r \le 0.5, \\ b, & \text{otherwise} \end{cases} \\
f(x,y) &= \begin{cases} 8, & \text{if } r \le 0.5, \\ 8(x^2+y^2)+4, & \text{otherwise} \end{cases}
\end{aligned} \tag{5.3.1}
$$

By choosing $b = 10$, it can be checked that on the interface $[u] = 1$ and $[\beta u_n] = -0.75$. The interface is shown in Fig 5.3.1 and a body-fitted initial mesh is given in Fig 5.3.1. In Fig 5.3.2, the WG solution based on mesh level 2 is depicted. The function jump is constant across the circular interface. However, the graph of the WG solution is piecewise constant in each element since the function value at the vertices are absent. Thus, we can obtain $R_h u_h$ at each vertex during the WGPPR process and the recovered solution is displayed in Fig 5.3.3, the graph of the recovered gradient $G_h^x(u_h)$ and $G_h^y(u_h)$ are presented in Fig 5.3.4 and Fig 5.3.5, respectively. We can see that the recovery process is able to capture the jump information along the interface. It can be seen that WGPPR works well on these two subdomains correspondingly and the flux jump is captured. Furthermore, the numerical result is reported in Table 5.3.1. Optimal convergence rate is achieved by WGFEM while we observed superconvergence for WGPPR at the rate of $O(h^{1.5})$. This validates the effectiveness and robustness of our proposed algorithm.

Example 2. In this example, we consider the elliptic interface problem in the square domain $\Omega = (-1,1) \times (-1,1)$ with a circular interface of radius $r_0 = 0.5$. The

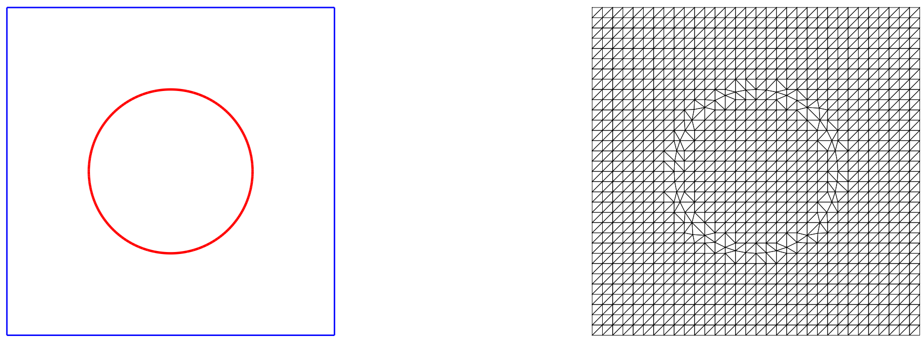

Figure 5.3.1: Example 1. (a) Shape of interface; (b) Body fitted initial mesh.

Table 5.3.1: Example 1: Comparison of H^1 error of Gradient recovery

Dof	$\|\nabla u - \nabla_w u_h\|$	order	$\|\nabla u - G_h u_h\|$	order
20608	1.83e-03	–	1.60e-03	–
82176	9.12e-04	0.50	5.67e-04	0.75
328192	4.55e-04	0.50	2.01e-04	0.75
1311744	2.28e-04	0.50	7.09e-05	0.75

exact solution is

$$u(x,y) = \begin{cases} \frac{r^3}{\beta^-}, & \text{if } z \in \Omega^-, \\ \frac{r^3}{\beta^+} + (\frac{1}{\beta^-} - \frac{1}{\beta^+}) r_0^3, & \text{if } z \in \Omega^+, \end{cases} \tag{5.3.2}$$

Here we choose $\beta^- = 1$ and $\beta^+ = 10$. The shape of interface is shown in Fig 5.3.6. A body-fitted initial mesh is depicted in Fig 5.3.6. The WGFEM solution is plotted

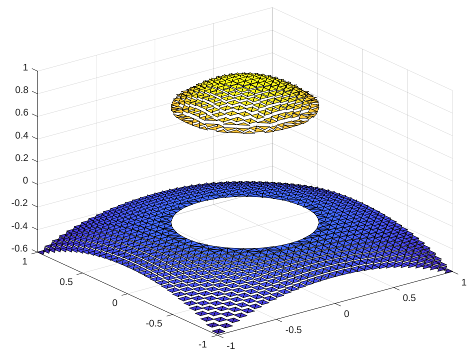

Figure 5.3.2: WG solution u_h

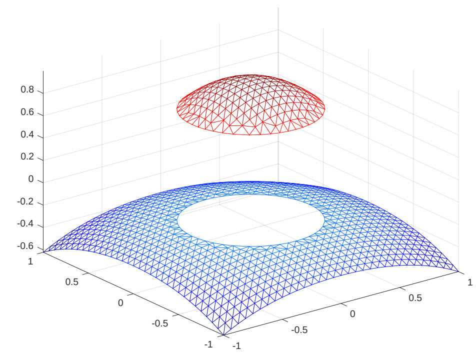

Figure 5.3.3: Recovered solution $R_h u_h$

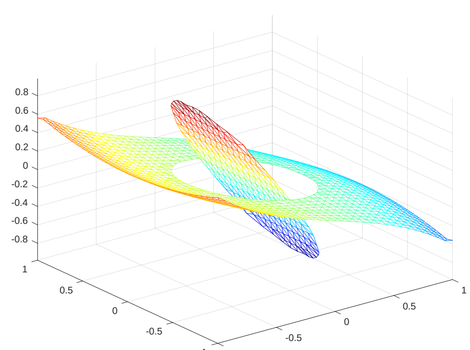

Figure 5.3.4: Graph of $G_h^x(u_h)$ at Level 2

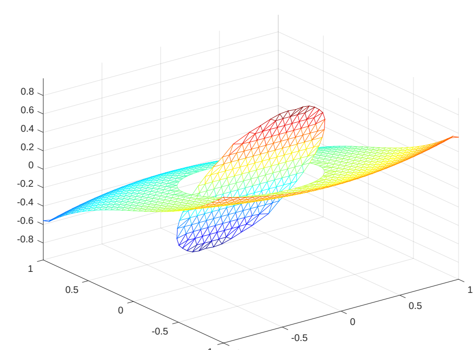

Figure 5.3.5: Graph of $G_h^y(u_h)$ at Level 2

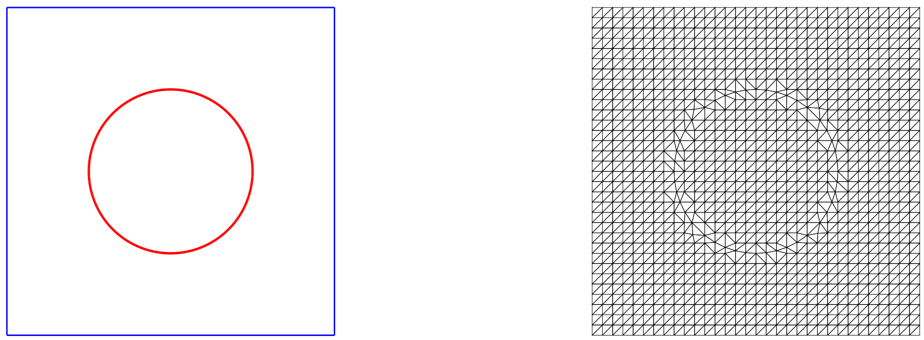

Figure 5.3.6: Example 2. (a) Shape of interface; (b) Body fitted initial mesh.

in Fig 5.3.7 and the recovered solution $R_h u_h$ is presented in Fig 5.3.8. Since the WG scheme uses piecewise constant on each element, the discontuity is obviously observed in Fig 5.3.7. The recovered function $R_h u_h$ produces a continuous function in different subdomains, therefore we can see a piecewise continuous function well presented in Fig 5.3.8.The recovered gradient function $G_h^x u_h$ and $G_h^y u_h$ are plotted in Fig 5.3.9 and 5.3.10. The numerical errors are displayed in Table 5.3.2. An optimal convergence in the H^1-seminorm is observed. The recovered gradient $G_h u_h$ superconverges to ∇u at the rate of $O(h^{1.5})$.

Example 3. Cardioid Interface Problem In this example, we consider the interface problem with a cardioid interface as in [53]. The interface Γ is the zero level of the

Table 5.3.2: Example 2: Comparison of H^1 error of Gradient recovery

Dof	$\|\nabla u - \nabla_w u_h\|$	order	$\|\nabla u - G_h u_h\|$	order
20608	3.14e-03	–	3.37e-03	–
82176	1.57e-03	0.50	1.06e-03	0.84
328192	7.85e-04	0.50	3.54e-04	0.79
1311744	3.93e-04	0.50	1.21e-04	0.78
5244928	1.96e-04	0.50	4.19e-05	0.76

function

$$\phi(x,y) = (3(x^2+y^2)-x)^2 - x^2 - y^2, \tag{5.3.3}$$

and the exact solution

$$u(x,y) = \phi(x,y)/\beta(x,y), \tag{5.3.4}$$

where

$$\beta(x,y) = \begin{cases} xy+3, & \text{if } (x,y) \in \Omega^-, \\ 100, & \text{if } (x,y) \in \Omega^+; \end{cases} \tag{5.3.5}$$

The contour of the interface is shown in Fig 5.3.11. The interface is not Lipschitz-continuous and has singular point at the origin. A body-fitted initial mesh is given in Fig 5.3.11. In Fig 5.3.12 and 5.3.13, we present the WGFEM solution u_h and the recovered solution $R_h u_h$. The recovered gradient function $G_h^x u_h$ and $G_h^y u_h$ are shown in Fig 5.3.14 and 5.3.15. The numerical result is reported in Table 5.3.3, we can observe optimal convergence rate $O(h)$ for WGFEM and superconvergence for WGPPR at the rate of $O(h^{1.6})$ even though the interface is not Lipschitz-continuous.

Example 4. In this example, we consider the interface problem with complex geometrical structure as in [69], the arbitrarily shaped interface in polar coordinates is given by

$$r = 0.40178(1+\cos(2\theta)\sin(6\theta))\cos(\theta), \tag{5.3.6}$$

Table 5.3.3: Example 3: Comparison of H^1 error of Gradient recovery

Dof	$\|\nabla u - \nabla_w u_h\|$	order	$\|\nabla u - G_h u_h\|$	order
1312	3.55e-02	–	9.22e-02	–
5184	1.70e-02	0.54	3.29e-02	0.75
20608	8.36e-03	0.51	1.03e-02	0.84
82176	4.17e-03	0.50	3.18e-03	0.85
328192	2.09e-03	0.50	1.02e-03	0.82
1311744	1.05e-03	0.50	3.46e-04	0.78

for $\theta \in [0, 2\pi]$. The interface and subdomains are displayed in Fig (5.3). The coefficient function is chosen as

$$\beta(x,y) = \begin{cases} (x^2 - y^2 + 3)/7, & \text{if } (x,y) \in \Omega^-, \\ (xy+2)/5, & \text{if } (x,y) \in \Omega^+; \end{cases} \tag{5.3.7}$$

and the exact solution is

$$u(x,y) = \begin{cases} \sin(x+y) + \cos(x+y) + 1, & \text{if } (x,y) \in \Omega^-, \\ x+y+1, & \text{if } (x,y) \in \Omega^+; \end{cases} \tag{5.3.8}$$

Due to the complex geometrical structure of the interface shown in Fig 5.3.16, we adopt the adaptive strategy to generate an initial body-fitteed mesh [51]. The initial mesh is displayed in Fig 5.3.16. It is obvious that the mesh is refined around the interface with high curvature. The WGFEM solution is shown in Fig 5.3.17 and the recovered solution $R_h u_h$ is plotted in Fig 5.3.18. The visible discontinuities of WGFEM solution is again observed while the recovered solution provides a piecewise continuous function which better approximates the exact solution u. The recovered gradient function $G_h^x u_h$ and $G_h^y u_h$ are depicted in Fig (5.3.19) and Fig (5.3.20), respectively. We can see clear continuity in both subdomains of the gradient function.

To track the convergence of our proposed algorithm, we perfom PPR on the other four level finer meshes as well. The refinements are obtained by uniform refinement while keeping the mesh along the interface. The numerical result is shown in Table 5.3.4, the convergence rates are listed with respect to the degree of freedom (DOF). While the mesh getting finer, the weak gradient of the WG solution tends to converge to the exact gradient at the rate of $O(h)$ while the convergence rate of recovered gradient is increasing to $O(h^{1.4})$ asymptotically. Since the gradient recovery technique is based on the numerical solution, the recovered gradient $G_h u_h$ relies greatly on u_h. When the numerical solution captures the exact solution well, the gradient recovery algorithm will generate good result. With a denser mesh, WG's performance is getting better and so does WGPPR. The phonomenon is clearly observed in Table (5.3.4).

Since $R_h u_h$ is obtained at each vertex, together with the given mesh information, we can perform the original PPR to get the gradient information and denote it by $G_h(R_h u_h)$. The numerical result is also shown in Table 5.3.4. To achieve the same level of accuracy, this approaches requires $\frac{1}{4}$ DOFs of $G_h u_h$. This may introduce extra computing time since a second level PPR is performed. However, the saving of computing time in WGFEM solving and WGPPR gradient recovery process are significant.

Table 5.3.4: Example 4: Comparison of H^1 error of Gradient recovery

Dof	$\|\nabla u - \nabla_w u_h\|$	order	$\|\nabla u - G_h u_h\|$	order	$\|\nabla u - G_h(R_h u_h)\|$	order
30452	2.67e-02	–	2.86e-02	–	1.18e-02	–
121784	1.69e-02	0.33	1.64e-02	0.40	4.83e-03	0.64
487088	9.22e-03	0.44	7.79e-03	0.54	2.68e-03	0.42
1948256	4.74e-03	0.48	3.26e-03	0.63	1.30e-03	0.52
7792832	2.39e-03	0.49	1.27e-03	0.68	5.50e-04	0.62

Example 5. In this example, we consider the interface problem as in [50][51]. The interface Γ in parametric form is defined by

$$\begin{cases} x(t) & = r(\theta)\cos(\theta) + x_c, \\ y(t) & = r(\theta)\sin(\theta) + y_c; \end{cases}$$

where $r(\theta) = r_0 + r_1 \sin(\omega\theta), 0 \le \theta < 2\pi$.

The exact solution is given by

$$u(x,y) = \begin{cases} \frac{r^2}{\beta^-}, & \text{if } (x,y) \in \Omega^-, \\ \frac{r^4 + C_0 \log(2r)}{\beta^+} + C_1\left(\frac{r_0^2}{\beta^-} - \frac{r_0^4 + C_0 \log(2r_0)}{\beta^+}\right), & \text{if } (x,y) \in \Omega^+; \end{cases} \tag{5.3.9}$$

where $r = \sqrt{x^2 + y^2}$. The source term is then determined accordingly:

$$f(x,y) = \begin{cases} \frac{4}{\beta^-}, & \text{if } (x,y) \in \Omega^-, \\ \frac{16r^2}{\beta^+}, & \text{if } (x,y) \in \Omega^+; \end{cases} \tag{5.3.10}$$

In this example, we take $r_0 = 0.4$, $r_1 = 0.2$ and $x_c = y_c = 0.02\sqrt{5}$. The coefficient β is a piecewise constant with $\beta^- = 1$ and $\beta^+ = 10$. To track the performance of WG and WGPPR, we choose $\omega = 5, 10, 20$ and the contour of the interface are shown in Fig 5.3.21, Fig 5.3.26 and Fig 5.3.31 respectively. Similar to previous example, adaptive meshes are employed as the initial mesh to capture the interface, see Fig 5.3.21, Fig 5.3.26 and Fig 5.3.31 . In Fig 5.3.22, Fig 5.3.27 and Fig 5.3.32, we present the WGFEM solution which is obviously piecewise constant in each element. Piecewise continuous function $R_h u_h$ is shown in Fig 5.3.23, Fig 5.3.28 and Fig 5.3.33 for differnent ω. Their corresponding gradient function $G_h^x u_h$ and $G_h^y u_h$ are plotted in Fig 5.3.24 - 5.3.25, Fig 5.3.29 - 5.3.30 and Fig 5.3.34 - 5.3.35.

With the refinement of the meshes, WGFEM solution u_h provides a better approximation to u and the convergence rate of the weak gradient is increased to $O(h)$ while the recovered gradient superconverges asymptotically at the rate of $O(h^{1.6})$. The numerical results are shown in Table 5.3.5, Table 5.3.6 and Table 5.3.7. Furthermore, we compute $G_h(R_h u_h)$ for $\omega = 5, 10, 20$ as well. The results are displayed in Table 5.3.5, Table 5.3.6 and Table 5.3.7 as well. It is clear that $G_h(R_h u_h)$ not only reduces the error, but also has a better convergence rate which is superconvergent at the rate of $O(h^{1.8})$. This again verifies the effectiveness and robustness for both PPR and WGPPR.

Table 5.3.5: Example 5: Comparison of H^1 error of Gradient recovery, $\omega = 5$

Dof	$\|\nabla u - \nabla_w u_h\|$	order	$\|\nabla u - G_h u_h\|$	order	$\|\nabla u - G_h(R_h u_h)\|$	order
9485	3.82e-02	–	1.20e-01	–	1.30e-01	–
37920	2.14e-02	0.42	4.54e-02	0.70	3.60e-02	0.92
151640	1.11e-02	0.48	1.55e-02	0.77	1.11e-02	0.85
606480	5.58e-03	0.49	5.10e-03	0.80	3.15e-03	0.91
2425760	2.80e-03	0.50	1.70e-03	0.79	9.18e-04	0.89

Table 5.3.6: Example 5: Comparison of H^1 error of Gradient recovery, $\omega = 10$

Dof	$\|\nabla u - \nabla_w u_h\|$	order	$\|\nabla u - G_h u_h\|$	order	$\|\nabla u - G_h(R_h u_h)\|$	order
31552	3.67e-02	–	9.96e-02	–	8.71e-02	–
126184	2.06e-02	0.42	3.90e-02	0.68	2.72e-02	0.84
504688	1.06e-02	0.48	1.37e-02	0.75	9.05e-03	0.79
2018656	5.34e-03	0.49	4.54e-03	0.80	2.63e-03	0.89

Table 5.3.7: Example 5: Comparison of H^1 error of Gradient recovery, $\omega = 20$

Dof	$\|\nabla u - \nabla_w u_h\|$	order	$\|\nabla u - G_h u_h\|$	order	$\|\nabla u - G_h(R_h u_h)\|$	order
78647	3.65e-02	–	9.80e-02	–	8.24e-02	–
314564	2.05e-02	0.42	3.87e-02	0.67	2.68e-02	0.81
1258208	1.05e-02	0.48	1.36e-02	0.75	8.96e-03	0.79
5032736	5.30e-03	0.49	4.52e-03	0.80	2.61e-03	0.89

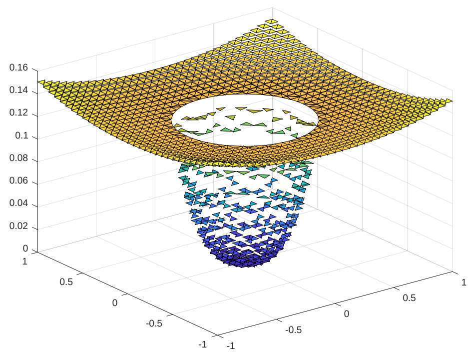

Figure 5.3.7: WG solution u_h

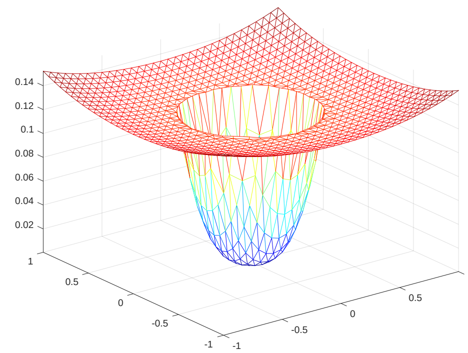

Figure 5.3.8: Recovered solution $R_h u_h$

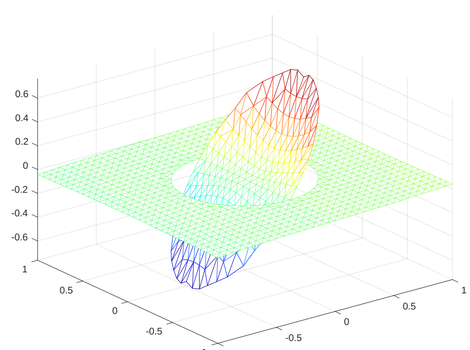

Figure 5.3.9: Example 2: $G_h^x(u_h)$

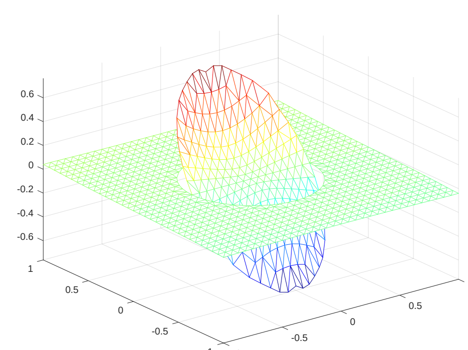

Figure 5.3.10: Example 2: $G_h^y(u_h)$

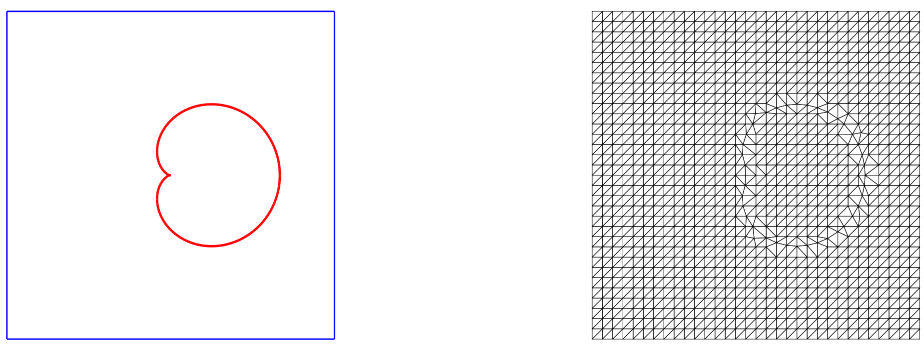

Figure 5.3.11: Example 3. (a) Shape of interface; (b) Body fitted initial mesh.

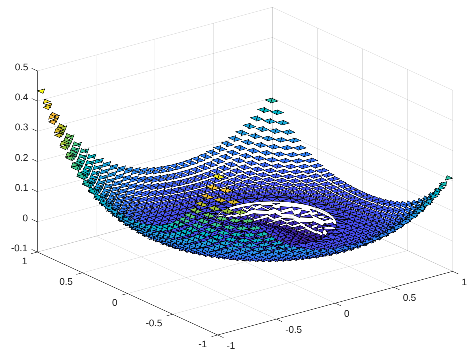

Figure 5.3.12: WG solution u_h

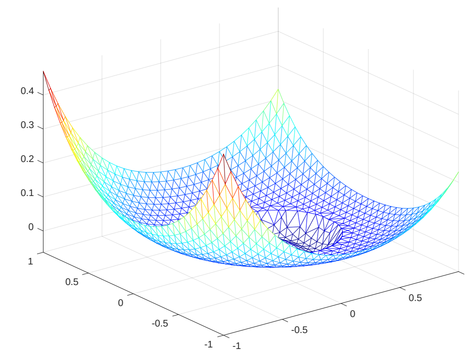

Figure 5.3.13: Recovered solution $R_h u_h$

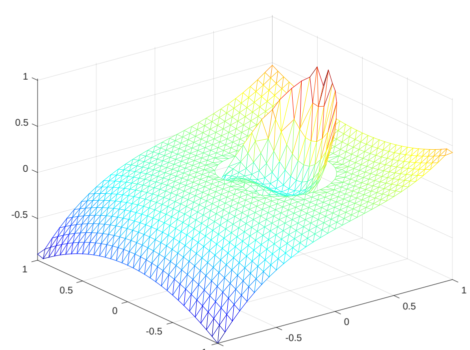

Figure 5.3.14: Example 3: $G_h^x(u_h)$

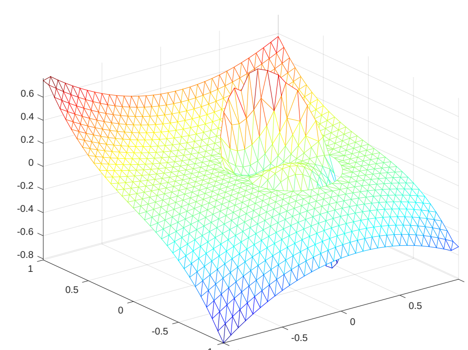

Figure 5.3.15: Example 3: $G_h^y(u_h)$

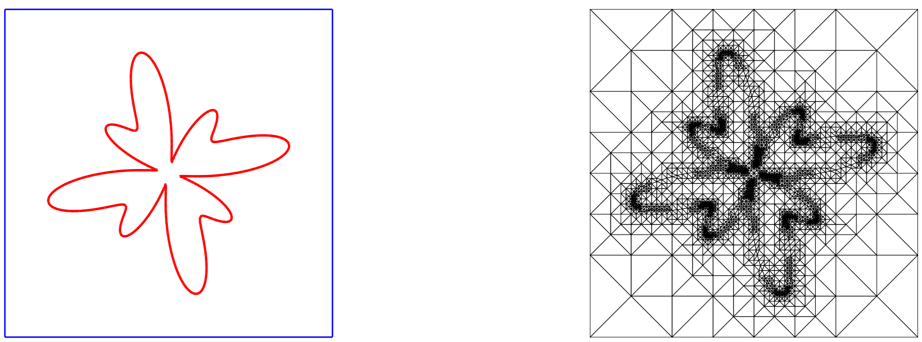

Figure 5.3.16: Example 4. (a) Shape of interface; (b) Body fitted initial mesh.

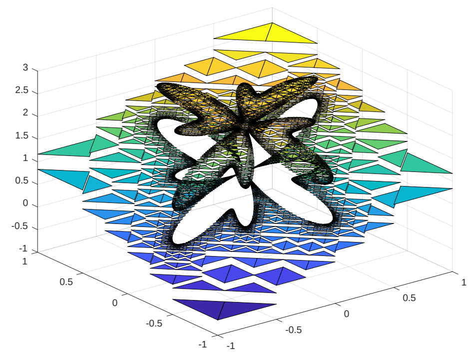

Figure 5.3.17: WG solution u_h

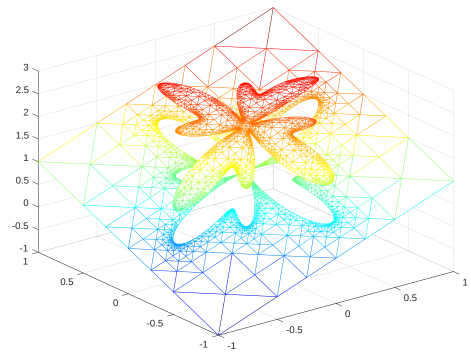

Figure 5.3.18: Recovered solution $R_h u_h$

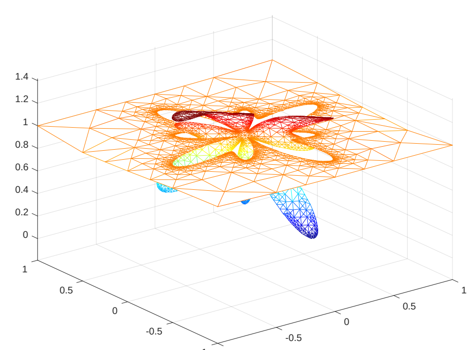

Figure 5.3.19: Example 4: $G_h^x(u_h)$

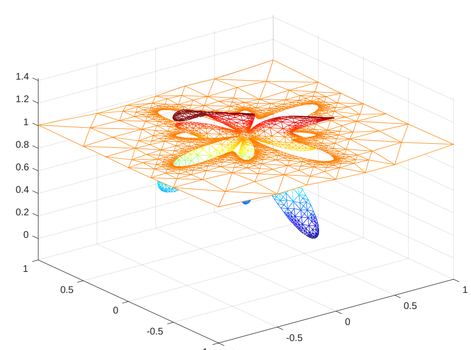

Figure 5.3.20: Example 4: $G_h^y(u_h)$

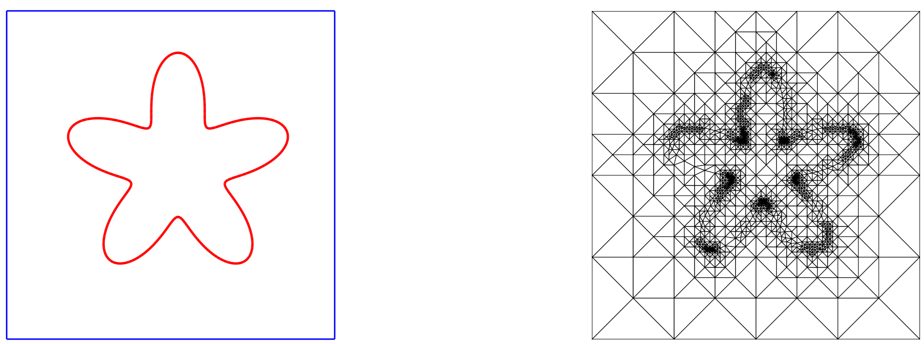

Figure 5.3.21: $\omega = 5$ (a) Shape of interface; (b) Body fitted initial mesh.

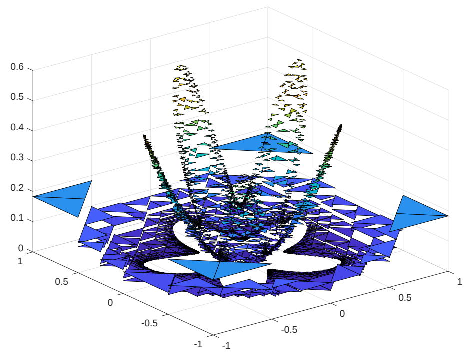

Figure 5.3.22: WG solution u_h

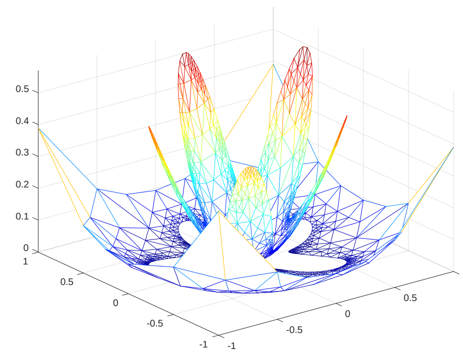

Figure 5.3.23: Recovered solution $R_h u_h$

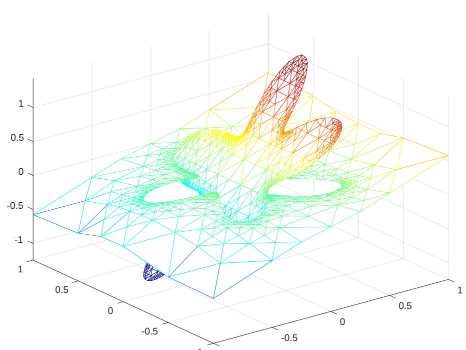

Figure 5.3.24: $G_h^x(u_h)$ when $\omega = 5$

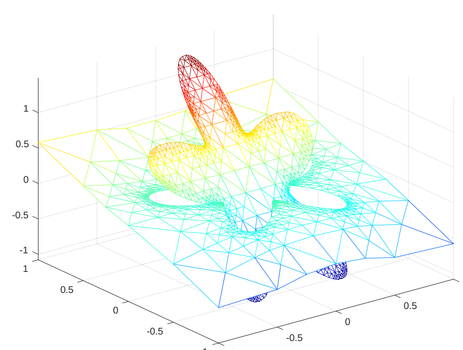

Figure 5.3.25: $G_h^y(u_h)$ when $\omega = 5$

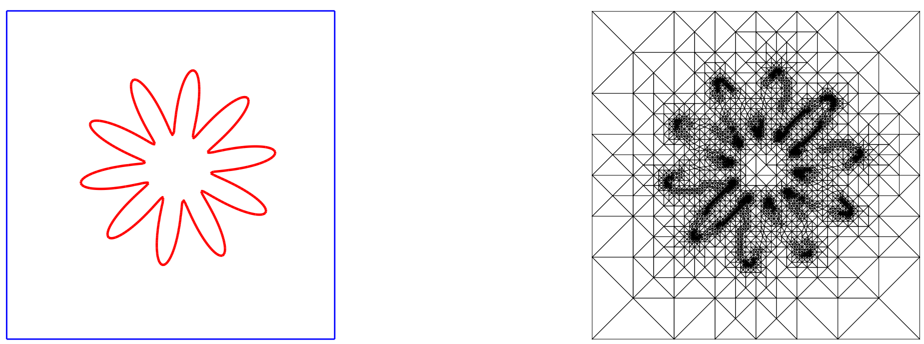

Figure 5.3.26: $\omega = 10$ (a) Shape of interface; (b) Body fitted initial mesh.

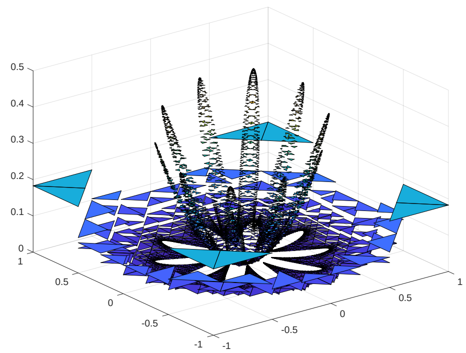

Figure 5.3.27: WG solution u_h

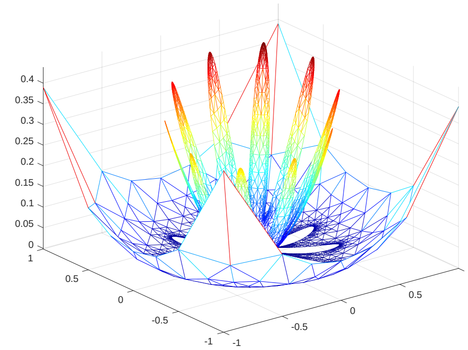

Figure 5.3.28: Recovered solution $R_h u_h$

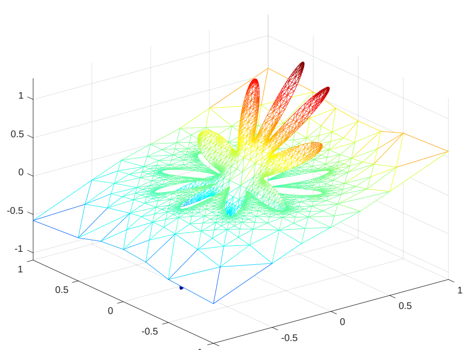

Figure 5.3.29: $G_h^x(u_h)$ when $\omega = 10$

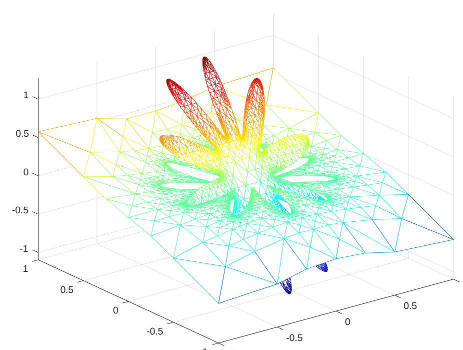

Figure 5.3.30: $G_h^y(u_h)$ when $\omega = 10$

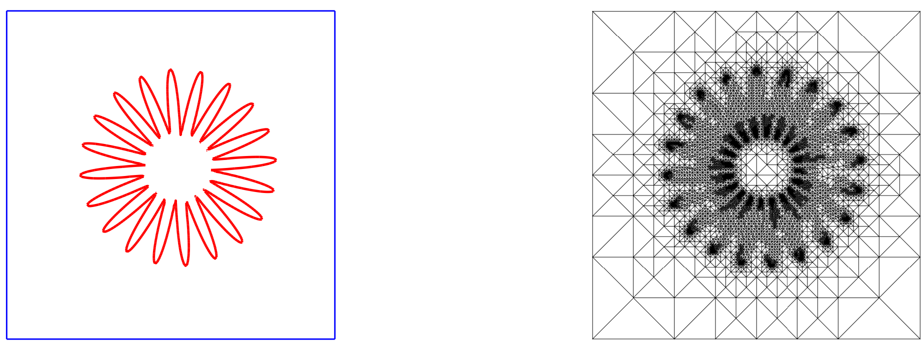

Figure 5.3.31: $\omega = 20$ (a) Shape of interface; (b) Body fitted initial mesh.

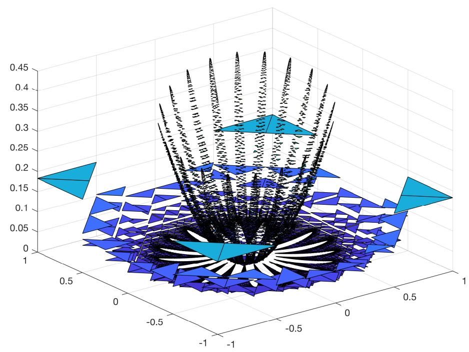

Figure 5.3.32: WG solution u_h

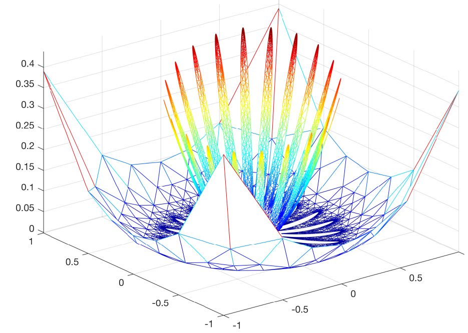

Figure 5.3.33: Recovered solution $R_h u_h$

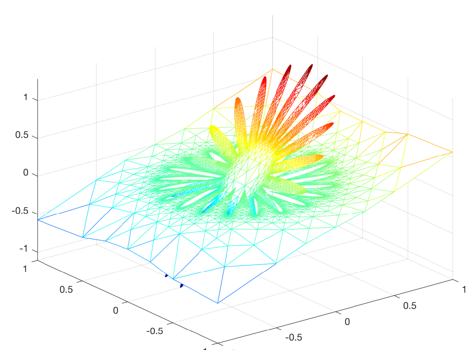

Figure 5.3.34: $G_h^x(u_h)$ when $\omega = 20$

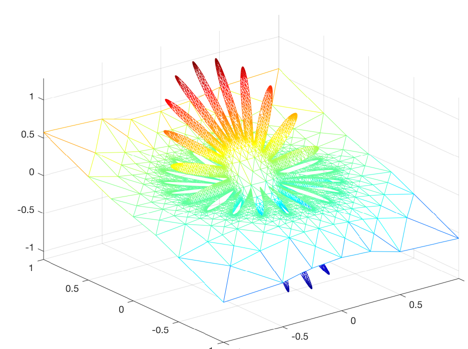

Figure 5.3.35: $G_h^y(u_h)$ when $\omega = 20$

CHAPTER 6 APPLICATIONS OF WGPPR

6.1 Application to adaptive methods

Adaptive finite element method (AFEM) based on local mesh refinement can be characterized in the loops of the form [10, 11]:

SOLVE → ESTIMATE → MARK → REFINE

In the **ESTIMATE** step, the a posteriori error estimators are of significant importance and are used to make local modifications. There are two types of a posteriori estimators: residual type and recovery type. For conforming FEM, the residual type a posteriori error estimators have been studies in [9, 123, 124, 125, 10, 126, 11, 127, 128]. For WGFEMs, the residual type a posteriori error estimator is firstly proposed and analyzed by Chen et. al. [109]. Later in [64], Zhang et. al. presented an a posteriori error estimator for the modified WGFEMs. For conforming finite element method, recovery type a posteriori error estimators have been studied in [9, 129, 18, 130, 38, 131, 132, 133]. In particular, Zhang and Naga introduced PPR and proposed a recovery type a posteriori error estimator in [18]. Since we have seen success in applying PPR to adaptive methods for standard Galerkin methods, it is natural for us to apply the same idea to the adaptivity of weak Galerkin method. In this section, we apply the proposed WGPPR to a recovery type a posteriori error estimator.

The local a posteriori error estimator on the element T as:

$$\eta(u_h, T) = \|G_h u_h - \nabla_w u_h\|_{0,T}, \tag{6.1.1}$$

where $G_h u_h$ is the recovered gradient using WGPPR and $\nabla_w uh$ is the weak gradient of the WG solution. The global error estimator is defined as

$$\eta(u_h, \Omega) = (\sum_{T\in\mathcal{T}_h} \eta(u_h, T))^{\frac{1}{2}}. \tag{6.1.2}$$

The adaptive algorithm can be summarized as following:

Given any initial mesh $\mathcal{T}_0$ and set $k = 0$:

- **SOLVE.** Compute the weak Galerkin solution u_h and the weak gradient $\nabla_w u_h$ of model problem (3.1.1) using proper WGFEM on the mesh $\mathcal{T}_k$.

- **ESTIMATE.** Compute the recovery gradient $G_h u_h$ using WGPPR and then compute the local error estimator $\eta(u_h, T)$ on $\mathcal{T}_k$.

- **MARK.** The marking set $\mathcal{M}_k \subset \mathcal{T}_k$ is defined by a set of element satisfying bulk marking strategy [8]:

$$\eta^2(u_h, T) \geq \theta \eta^2(u_h, \Omega), \tag{6.1.3}$$

  for some $\theta \in (0, 1)$.

- **REFINE** Refine $\mathcal{T}_k$ into $\mathcal{T}_{k+1}$ by using bisection method [126, 10, 11] which guarantees $\mathcal{T}_{k+1}$ is still a shape regular and conforming mesh. Set $k = k + 1$ and iterate.

For measuring the quality of the proposed error estimator, we define the effectivity index κ [9, 125] as the ratio between the estimated error and the weak Galerkin approximation error, that is

$$\kappa = \frac{\|G_h u_h - \nabla_w u_h\|_{0,\Omega}}{\|\nabla u - \nabla_w u_h\|_{0,\Omega}}. \tag{6.1.4}$$

To test the robustness of the error estimator (6.1.2), we use three examples as our benchmark problems: L-shape problem, Crack problem and Kellogg problem.

L-shape Problem. Let $\Omega := (-1, 1)^2 \setminus \{[0, 1) \times (-1, 0]\}$ be a L-shaped domain with a reentrant corner. Consider the Laplace equation on the L-shaped domain Ω and $u = g$ on $\partial\Omega$. We choose the Dirichlet boundary condition g such that the exact solution is $u(r, \theta) = r^{\frac{2}{3}} sin(\frac{2}{3}\theta)$ in polar coordinates.

We use the lowest order WG method, i.e., (P_0, P_0, RT_0) element, and expect the first order convergence of the energy error $||\nabla u - \nabla_w u_h|| \leq CN^{-\frac{1}{2}}$. The initial mesh is given in Fig 6.1.1. The bulk marking strategy by Dörfler [8] with $\theta = 0.5$ is adopted in our simulation for marking. Marked elements are refined by the newest vertex bisection. We present the adaptive grid genertated by our algorithm in Fig 6.1.2 and the error table is displayed in Table 6.1.1. The decay of energy error is shown in Fig 6.1.3, it meets our expectation. In Fig 6.1.3, we can observe $||G_h u_h - \nabla u||_{L^2(\Omega)}$ is superconvergent with order $O(N^{-0.72})$. In Fig 6.1.4, we depict the curve of effectivity index versus number of DOFs. It can be clearly seen that it converges to 1 quickly after the several iterations which indicates the proposed a posteriori error estimator (6.1.2) is asymptotically exact.

Table 6.1.1: Error table of L-shape problem

N	$\|\|\nabla u - \nabla_{w,h} u_h\|\|$	η	κ	N	$\|\|\nabla u - \nabla_{w,h} u_h\|\|$	η	κ
19	4.166218e-01	4.665876e-01	1.12	454	9.309483e-02	9.500979e-02	1.02
24	3.817037e-01	3.684954e-01	0.97	609	7.908910e-02	7.911142e-02	1.00
29	3.464294e-01	2.763590e-01	0.80	877	6.687090e-02	6.767188e-02	1.01
35	2.868965e-01	3.193751e-01	1.11	1240	5.440797e-02	5.456889e-02	1.00
50	2.612274e-01	2.683002e-01	1.03	1686	4.722625e-02	4.778079e-02	1.01
60	2.378536e-01	2.224138e-01	0.94	2364	3.955613e-02	3.965601e-02	1.00
76	2.228042e-01	2.252268e-01	1.01	3311	3.333611e-02	3.370858e-02	1.01
91	1.925436e-01	1.791509e-01	0.93	4506	2.792213e-02	2.786410e-02	1.00
136	1.713096e-01	1.705006e-01	1.00	6214	2.407998e-02	2.412359e-02	1.00
165	1.477008e-01	1.435043e-01	0.97	8402	2.026700e-02	2.012421e-02	0.99
251	1.272253e-01	1.291997e-01	1.02	11504	1.756064e-02	1.761035e-02	1.00

Crack Problem Let us now consider the elliptic problem (2.0.1) on the crack domain $\Omega = \{|x|+|y| < 1\} \setminus \{0 \leq x \leq 1, y = 0\}$. The right hand side function is chosen as $f = 1$ and the exact solution u in polar coordinates is given as $u(r, \theta) = r^{\frac{1}{2}} \sin \frac{\theta}{2} - \frac{1}{4} r^2$.

The initial mesh is plotted in Fig 6.1.5. We employ the WGFEM with (P_0, P_0, RT_0)

element to solve the crack problem on the initial mesh and the bulking marking strategy [8] with $\theta = 0.4$ is adpoted. Marked elements are refined by the newest vertex bisection. The adaptive refined mesh is displayed in Fig 6.1.6. Fig 6.1.7 shows that the L^2 error of weak derivatives is optimal while the recovery gradient error superconverges at rate of $O(h^{1.4})$. Again the effectivity index κ converges to 1 quickly which implies the error estimator is asymptotically exact, see Fig 6.1.8. In Table 6.1.2, we display the error of the crack problem.

Table 6.1.2: Error table of Crack problem

N	$\|\|\nabla u - \nabla_{w,h} u_h\|\|$	η	κ	N	$\|\|\nabla u - \nabla_{w,h} u_h\|\|$	η	κ
13	5.591752e-01	8.136789e-01	1.46	285	1.815918e-01	1.913780e-01	1.05
25	4.637481e-01	7.373226e-01	1.59	348	1.606301e-01	1.665744e-01	1.04
28	4.519550e-01	6.711667e-01	1.49	452	1.457635e-01	1.525263e-01	1.05
54	4.171312e-01	5.216470e-01	1.25	553	1.294438e-01	1.345086e-01	1.04
59	3.809660e-01	4.334415e-01	1.14	724	1.194848e-01	1.248537e-01	1.04
62	3.671047e-01	4.386639e-01	1.19	931	1.040062e-01	1.063679e-01	1.02
77	3.325325e-01	3.765405e-01	1.13	1215	9.143555e-02	9.373127e-02	1.03
80	3.189069e-01	3.802232e-01	1.19	1548	8.075016e-02	8.168402e-02	1.01
95	2.940545e-01	3.292018e-01	1.12	2067	6.882329e-02	7.094008e-02	1.03
101	2.729706e-01	3.399531e-01	1.25	2724	6.048520e-02	6.097953e-02	1.01
136	2.501901e-01	2.703636e-01	1.08	3530	5.234802e-02	5.370433e-02	1.03
157	2.432314e-01	2.726219e-01	1.12	4555	4.637032e-02	4.677240e-02	1.01
177	2.220768e-01	2.334317e-01	1.05	5975	3.968880e-02	4.053908e-02	1.02
216	2.132452e-01	2.259377e-01	1.06	7764	3.511732e-02	3.528910e-02	1.00
234	1.991539e-01	2.038693e-01	1.02	10059	3.070675e-02	3.112878e-02	1.01

6.2 3D Problem

Let us now consider a 3D example. The model problem is

$$-\Delta u = 3\pi^2 \sin \pi x \sin \pi y \sin \pi z,$$

in the domain $\Omega = (0,1)^3$, and $u = 0$ on $\partial\Omega$. The solution of this problem is $u(x, y, z) = \sin \pi x \sin \pi y \sin \pi z$. An initial mesh T_0 is obtained by partitioning $x - axis$, $y - axis$ and $z - axis$ into 4 equally distributed subintervals, then dividing one cube into six tetrahedron. Here we employ WGFEM with (P_0, P_0, RT_0) element. The numerical results are displayed in Table 6.2.1 and we can observe $\|\nabla u - G_h u_h\|$ is superconvergent with order $O(h^{-2/3})$. We define the interior domain $\Omega_{h,1}$ and boundary domain $\Omega_{h,2}$ in a similar way as in (2.3.1) and (2.3.2). The numerical results indicate that WGPPR is a second order approximation to ∇u.

Table 6.2.1: Poisson 3D

Dof	$\|\nabla u - G_h u_h\|_\Omega$	order	$\|\nabla u - G_h u_h\|_{\Omega_1}$	order	$\|\nabla u - G_h u_h\|_{\Omega_2}$	order
9600	1.51e-01	–	9.75e-02	–	1.15e-01	–
75264	4.25e-02	0.62	2.57e-02	0.65	3.38e-02	0.60
595968	1.12e-02	0.64	6.52e-03	0.66	9.13e-03	0.63
4743168	2.92e-03	0.65	1.77e-03	0.63	2.32e-03	0.66

6.3 Stokes Problem

Consider the Stokes problem which seeks unknown function $\mathbf{u}$ and p satisfying

$$-\Delta u + \nabla p = f \quad \text{in } \Omega \tag{6.3.1}$$

$$\nabla \cdot u = 0 \quad \text{in } \Omega \tag{6.3.2}$$

$$u = \mathbf{g} \quad \text{on } \partial\Omega \tag{6.3.3}$$

where Ω is a polygonal domain in $\mathbb{R}^2$.

For any integer $k \geq 1$, we define a weak Galerkin finite element space for the velocity variable as $V_h = \{v = \{v_0, v_b\} : \{v_0, v_b\} \in [P_k(T)]^2 \times [P_k(e)]^2, e \subset \partial T\}$. For the pressure variable, we have the following finite element space $W_h = \{q : q \in L_0^2(\Omega), q|_T \in P_{k-1}(T)\}$. Denote by V_h^0 the subspace of V_h consisting of discrete weak functions with vanishing boundary values; i.e., $V_h^0 = \{v = \{v_0, v_b\} \in V_h, v_b = 0 \text{ on } \partial\Omega\}$.

The discrete weak gradient operator, denoted by $\nabla_{w,K}$, is defined as the unique polynomial $\nabla_{w,K} v \in [P_{k-1}(K)]^2$ satisfying the following equation,

$$(\nabla_{w,K} v, q)_K = -(v_0, \nabla \cdot q)_K + < v_b, q \cdot \mathbf{n} >_{\partial K}, \quad \forall q \in [P_{k-1}(K)]^2. \tag{6.3.4}$$

Similarly, the discrete weak divergence operator, denoted by $\nabla_{w,K} \cdot$, is defined as the unique polynomial $\nabla_{w,K} \cdot v \in [P_{k-1}(K)]^2$ that satisfies the following equation

$$(\nabla_{w,K} \cdot v, \phi)_K = -(v_0, \nabla \phi)_K + < v_b \cdot \mathbf{n}, \phi >_{\partial K}, \quad \forall \phi \in P_{k-1}(K). \tag{6.3.5}$$

The weak method for the Stokes problem is: find $u_h = \{u_0, u_b\} \in V_h$ and $p_h \in W_h$ such that $u_b = Q_b g$ on $\partial\Omega$ and

$$a(u_h, v) - b(v, p_h) = (f, v_0), \tag{6.3.6}$$

$$b(u_h, q) = 0 \tag{6.3.7}$$

for all $v = \{v_0, v_b\} \in V_h^0$ and $q \in W_h$, where $s(v, w) = \sum_{T \in \mathbb{T}} h_T^{-1} < v_0 - v_b, w_0 - w_b >_{\partial T}$, $a(v, w) = (\nabla_w v, \nabla_w w) + s(v, w)$, and $b(v, q) = (\nabla_w \cdot v, q)$.

In the test, the exact solution is

$$u(x, y) = \begin{pmatrix} \pi \sin^2(\pi x) \sin(2\pi y) \\ \\ -\pi \sin(2\pi x) \sin^2(\pi y) \end{pmatrix}$$

and $p(x, y) = \cos(\pi x) \cos(\pi y)$. We use the weak Galerkin method with (P_0, P_0, RT_0) element and (P_1, P_1, P_0) element to solve the model problem. The numerical results are reported in Table 6.3.1 and Table 6.3.3. Optimal convergence rate is achieved by weak Galerkin methods for both elements. Then we use WGPPR for these two different WG scheme to recover the gradient information for u_h, numerical results are shown in Table 6.3.2 and Table 6.3.4, respetively. It is obviously that superconvergence phenomenon is observed for (P_0, P_0, RT_0) element at the rate of $O(h^2)$ in Table 6.3.2. For the (P_1, P_1, P_0) element, we can see a convergence rate of $O(h^{1.8})$. To further explore the superconvergence behavior, we split the domain to interior domain $\Omega_{h,1}$ and $\Omega_{h,2}$ as defined in (2.3.1) and (2.3.2). The H_1 error of the gradient recovery in $\Omega_{h,1}$ and $\Omega_{h,2}$ are displayed in Table 6.3.5. It clearly shows the error in the interior domain has second order convergence rate which is superconvergent.

Table 6.3.1: Stokes Problem: WG using $(P_0(T), P_0(e), RT_0(T))$

$1/h$	$\|u - u_h\|$	order	$\|p_h - p\|_\Omega$	order
8	9.8624e-02	–	8.7513e-01	–
16	2.5276e-02	1.9642	4.1211e-01	1.0865
32	6.3793e-03	1.9863	2.0019e-01	1.0416
64	1.5992e-03	1.9960	9.9207e-02	1.0129
128	4.0009e-04	1.9990	4.9486e-02	1.0034

Table 6.3.2: Stokes Problem: WG using $(P_0(T), P_0(e), RT_0(T))$

$1/h$	$\|\nabla_w u_h - \nabla u\|_\Omega$	order	$\|G_h u_h - \nabla u\|_\Omega$	order
8	1.8723e+00	–	3.5219e+00	–
16	9.1907e-01	1.0266	1.0174e+00	1.7915
32	4.5785e-01	1.0053	2.6807e-01	1.9242
64	2.2874e-01	1.0012	6.9002e-02	1.9579
128	1.1435e-01	1.0003	1.7803e-02	1.9545

Table 6.3.3: Stokes Problem: WG using $(P_1(T), P_1(e), P_0(T))$

$1/h$	$\|u - u_h\|$	order	$\|p_h - p\|_\Omega$	order
8	4.2921e-01	–	6.6931e-01	–
16	1.0944e-01	1.9716	3.3393e-01	1.0031
32	2.7497e-02	1.9928	1.6636e-01	1.0053
64	6.8828e-03	1.9982	8.3075e-02	1.0018
128	1.7213e-03	1.9995	4.1523e-02	1.0005

Table 6.3.4: Stokes Problem: WG using $(P_1(T), P_1(e), P_0(T))$

$1/h$	$\|\nabla_w u_h - \nabla u\|_\Omega$	order	$\|G_h u_h - \nabla u\|_\Omega$	order
8	7.4400e+00	–	3.2748e+00	–
16	3.7693e+00	0.9810	9.3912e-01	1.8020
32	1.8908e+00	0.9953	2.6406e-01	1.8304
64	9.4619e-01	0.9988	7.6686e-02	1.7838
128	4.7319e-01	0.9997	2.3460e-02	1.7088

Table 6.3.5: Stokes Problem: Interior error vs Boundary error

$1/h$	$\|G_h u_h - \nabla u\|_{\Omega_1}$	order	$\|G_h u_h - \nabla u\|_{\Omega_2}$	order
8	2.5992e+00	–	1.8760e+00	–
16	6.9950e-01	1.8937	6.0179e-01	1.6403
32	1.7811e-01	1.9735	1.8991e-01	1.6639
64	4.6152e-02	1.9483	6.0246e-02	1.6564
128	1.1725e-02	1.9768	2.0133e-02	1.5813

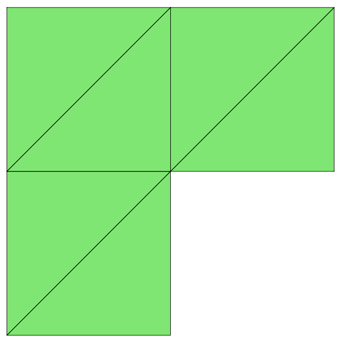

Figure 6.1.1: L-shape problem: Initial mesh

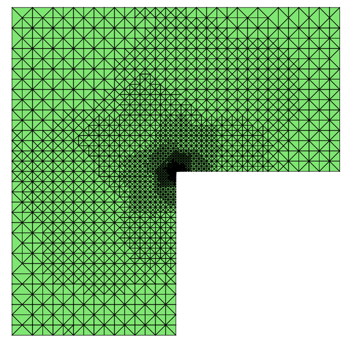

Figure 6.1.2: Adaptive mesh

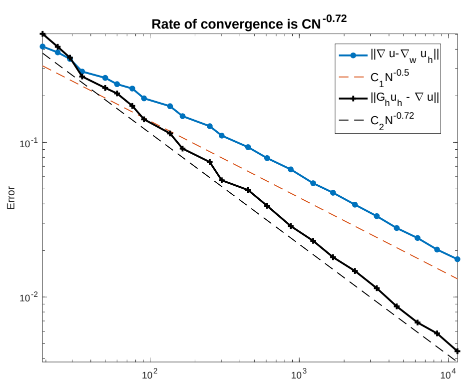

Figure 6.1.3: Decay of recovery error

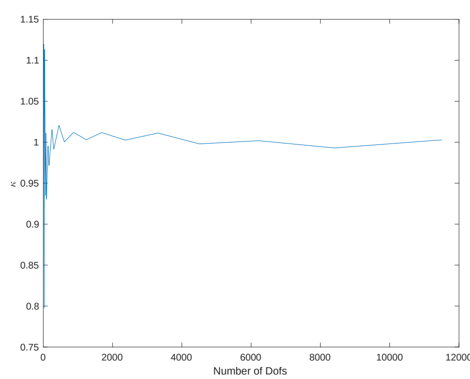

Figure 6.1.4: efficient index κ

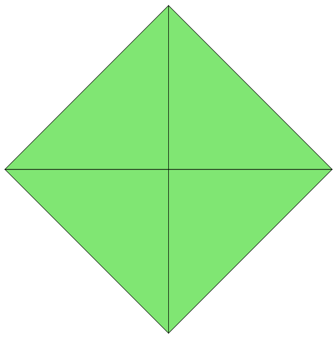

Figure 6.1.5: Crack Problem: Initial mesh

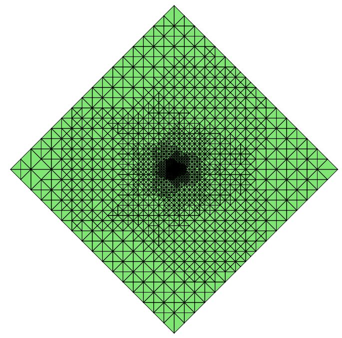

Figure 6.1.6: Adaptive mesh

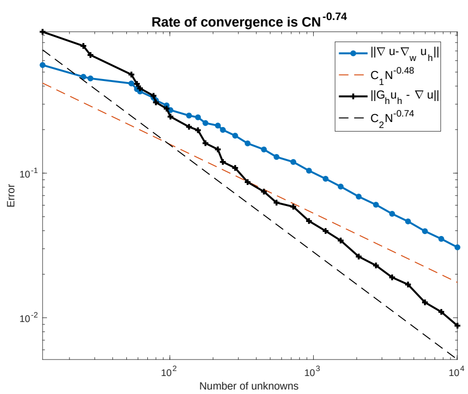

Figure 6.1.7: Decay of gradient error

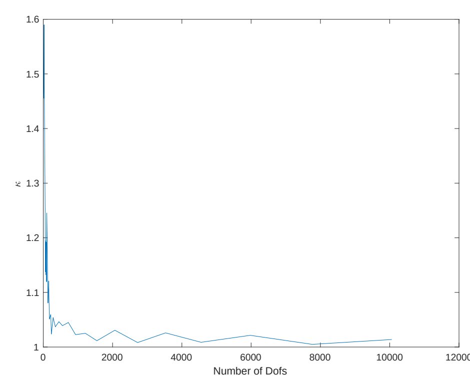

Figure 6.1.8: efficient index κ

REFERENCES

[1] R.A. Adams and J.J.F. Fournier, *Sobolev spaces*, Second edition, Elsevier/Academic Press, Amsterdam, 2003.

[2] S.C. Brenner and L.R. Scott, *The mathematical theory of finite element methods*, Third edition, Texts in Applied Mathematics, Springer, New York, 2008.

[3] W. Cao, *Superconvergence analysis of the linear finite element method and a gradient recovery postprocessing on anisotropic meshes*, Math. Comp. 84 (2015), no. 291, 89–117.

[4] P.G. Ciarlet, *The Finite Element Method for Elliptic Problems*, North-Holland, Amsterdam, 1978.

[5] *COMSOL Multiphysics User's Guide, Version 3.5a*, 2008.

[6] *COMSOL Multiphysics User's Guide, Version 4.3*, 2012.

[7] *Introduction to COMSOL Multiphysics, Version 5.3*, 2017.

[8] W. Dörfler, *A convergent adaptive algorithm for Poisson's equation*, SIAM J. Numer. Anal., 33(1996), 1106-1024.

[9] M. Ainsworth and J. T. Oden, *A posteriori error estimation in finite element analysis*, Wiley-Interscience [John Wiley & Sons], New York, 2000.

[10] R. H. Nochetto, K. G. Siebert, and A. Veeser, *Theory of adaptive finite element methods: an introduction*, Multiscale, nonlinear and adaptive approximation, 409–542, Springer, Berlin, 2009.

[11] , R. H. Nochetto, A. Veeser, *Primer of adaptive finite element methods*, Multiscale and adaptivity: modeling, numerics and applications, 125–225, Springer, Heidelberg, 2012.

[12] P. Binev, W. Dahmen, and R. DeVore, *Adaptive finite element methods with convergence rates*, Numer. Math., 97 (2004), 219–268.

[13] H. Guo and Z. Zhang, *Gradient recovery for the Crouzeix-Raviart element*, J. Sci. Comput. 64 (2015), no. 2, 456–476.

[14] H. Guo, Z. Zhang, and R. Zhao, *Superconvergent Two-grid Methods For Elliptic Eigenvalue Problems*, Journal of Scientific Computing(2016), DOI 10.1007/s10915-016-0245-2.

[15] W. Huang and R. D. Russell, *Adaptive moving mesh methods*, Springer, New York, 2011.

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

[17] L. Kamenski and W. Huang, *How a nonconvergent recovered Hessian works in mesh adaptation*, SIAM J. Numer. Anal. 52 (2014), 1692–1708.

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

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

[20] A. Naga and Z. Zhang, *Function value recovery and its application in eigenvalue problems*, SIAM J. Numer. Anal., 50-1 (2012), 272–286.

[21] A. Naga, Z. Zhang, and A. Zhou, *Enhancing eigenvalue approximation by gradient recovery*, SIAM J. Sci. Comput. 28 (2006), 1289–1300.

[22] H. Wu and Z. Zhang, *Enhancing eigenvalue approximation by gradient recovery on adaptive meshes*, IMA J. Numer. Anal. 29, 1008-1022(2009)

[23] J. Fang, X. Gao and A. Zhou, *A finite element recovery approach to eigenvalue approximations with application to electronic structure calculations*, J. Sci. Comput. 55, 432-454(2013)

[24] L. Meng and Z. Zhang, *The ultraconvergence of eigenvalues for bi-quadrature finite elements*, J. Comput. Math.30, 555-564(2012)

[25] G. M. Porta, S. Perotto, and F. Ballio, *Anisotropic mesh adaptation driven by a recovery-based error estimator for shallow water flow modeling*, Internat. J. Numer. Methods Fluids , 70 (2012), 269–299.

[26] B. Pouliot, M. Fortin, A. Fortin and E. Chamberland, *On a new edge-base gradient recovery technique*, Int. J. Numer. Meth. Engng., 93-1 (2013), 52–65.

[27] L. Shen and A. Zhou, *A defect correction scheme for finite element eigenvalues with applications to quantum chemistry*, SIAM J. Sci. Comput. 28 (2006), 321–338.

[28] J. R. Shewchuk, *Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator*, in Applied Computational Geometry: Towards Geometric Engineering, Lecture Notes in Comput. Sci. 1148, M.C. Lin and D. Manocha, eds, Springer-Verlag, Berlin, 1996, 203-222.

[29] L. Song and Z. Zhang, *Superconvergence property of an over-penalized discontinuous Galerkin finite element gradient recovery method*, J. Comput. Phys. 299 (2015), 1004–1020.

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

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

[32] O. C. Zienkiewicz and J. Z. Zhu, *The superconvergent patch recovery (SPR) and adaptive finite element refinement*, Comput. Methods Appl. Mech. Engrg. 101 (1992), 207–224.

[33] 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 (1992), 1331–1364.

[34] 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 (1992), 1365–1382.

[35] H. Wu and Z. Zhang, *Can We Have Superconvergent Gradient Recovery Under Adaptive Meshes?*, SIAM Journal on Numerical Analysis 45-4 (2007), 1701-1722.

[36] H. Guo, Z. Zhang, and R. Zhao, *Hessian Recovery for Finite Element Methods*, Math. Comp. 86 (2017), no. 306, 1671–1692.

[37] H Guo, Z Zhang, R Zhao, and Q Zou, *Polynomial preserving recovery on boundary*, J. Comput. Appl. Math. 307 (2016), 119–133.

[38] J. Xu and Z. Zhang, *Analysis of recovery type a posteriori error estimators for mildly structured grids*, Mathematics of computation, Volume 73, Number 247, 1139–1152.

[39] X. Hu and X. Cheng, *Acceleration of a two-grid method for eigenvalue problems*, Math. Comput. 80, 1287-1301(2011).

[40] Y. Yang and H. Bi, *Two-grid finite element discretization schemes based on shifted-inverse power method for elliptic eigenvalue problems*, SIAM J. Numer. Anal. 49, 1602-1624(2011).

[41] A.B. Andreev, R.D. Lazarov and M.R. Racheva, *Postprocessing and higher order convergence of the mixed finite element approximations of biharmonic eigenvalue problems*, J. Comput. Appl. Math. 182, 333-349(2005).

[42] M.R. Racheva and A.B. Andreev, *Superconvergence postprocessing for eigenvalues*, Comput. Methods Appl. Math. 2, 171-185(2002).

[43] R.E. Bank and J. Xu, *Asymptotically exact a posteriori error estimators. I. Grids with superconvergence*, SIAM J. Numer. Anal. 41(2003), no. 6, 22942312 (electronic), DOI 10.1137/S003614290139874X.

[44] W. Cao, *Superconvergence analysis of the linear finite element method and a gradient recovery postprocessing on anisotropic meshes*, Math. Comp. 84(2015), no. 291, 89117, DOI 10.1090/S0025-5718-2014-02846-9.

[45] X. Gan and J.E. Akin, *Superconvergent second order derivative recovery technique and its application in a nonlocal damage mechanics model*, Finite Elements in Analysis and Design 35 (2014), 118127.

[46] O. Lakkis and T. Pryer, *A finite element method for second order nonvariational elliptic problems*, SIAM J. Sci. Comput. 33 (2011), no. 2, 786801, DOI 10.1137/100787672.

[47] O. Lakkis and T. Pryer, *A finite element method for nonlinear elliptic problems*, SIAM J. Sci. Comput. 35 (2013), no. 4, A2025A2045, DOI 10.1137/120887655.

[48] M. Neilan, *Finite element methods for fully nonlinear second order PDEs based on a discrete Hessian with applications to the Monge-Ampère equation*, J. Comput. Appl. Math. 263(2014), 351369, DOI 10.1016/j.cam.2013.12.027.

[49] J. Bedrossian, J.H. von Brecht, S. Zhu, E. Sifakis, and J.M.Teran, *A second order virtual node method for elliptic problems with interfaces and irregular domains*, J. Comput. Phys., 229(18):6405-6426, 2010. ISSN 0021-9991

[50] Z. Li, *A fast iterative algorithm for elliptic interface problems*, SIAM J. Numer. Anal., 35(1):230-254, 1998. ISSN 0036-1429.

[51] H. Wei, L. Chen, Y. Huang, and B. Zheng, *Adaptive mesh refinement and superconvergence for two-dimensional interface problems*, SIAM J. Sci. Comput., 36(4):A1478-A1499, 2014. ISSN 1064-8275

[52] C. Börgers, *A triangulation algorithm for fast elliptic solvers based on domain imbedding*, SIAM J. Numer. Anal. 27(1990), no.5, 1187-1196. MR 1061125

[53] S. Hou, X.D. Liu, *A numerical method for solving variable coefficent elliptic equation with interfaces*, J. Comput. Phys., 202(2)(2005), 411-445. ISSN 0021-9991

[54] Lin Mu, Junping Wang and Xiu Ye, *Effective implementation of the weak Galerkin finite element methods for the biharmonic equation*, Comput. Math. Appl. 74 (2017), no. 6, 1215–1222.

[55] Lin Mu, Junping Wang and Xiu Ye, *A Least-Squares-Based Weak Galerkin Finite Element Method for Second Order Elliptic Equations*, SIAM J. Sci. Comput. 39 (2017), no. 4, A1531–A1557.

[56] Yu Du and Zhimin Zhang, *A Numerical Analysis of the Weak Galerkin Method for the Helmholtz Equation with High Wave Number*, Commun. Comput. Phys. 22 (2017), no. 1, 133–156.

[57] Xiaobo Zheng, Gang Chen and Xiaoping Xie, *A divergence-free weak Galerkin method for quasi-Newtonian Stokes flows*, Sci. China Math. 60 (2017), no. 8, 1515–1528.

[58] Rui Li, Jian Li, Xin Liu and Zhangxin Chen, *A weak Galerkin finite element method for a coupled Stokes-Darcy problem*, Numer. Methods Partial Differential Equations 33 (2017), no. 4, 1352–1373.

[59] Chunmei Wang and Haomin Zhou, *A weak Galerkin finite element method for a type of fourth order problem arising from fluorescence tomography*, J. Sci. Comput. 71 (2017), no. 3, 897–918.

[60] Yunqing Huang, Jichun Li and Dan Li, *Developing weak Galerkin finite element methods for the wave equation*, Numer. Methods Partial Differential Equations 33 (2017), no. 3, 868–884.

[61] Lunji Song, Kaifang Liu and Shan Zhao, *A weak Galerkin method with an over-relaxed stabilization for low regularity elliptic problems*, J. Sci. Comput. 71 (2017), no. 1, 195–218.

[62] Fuzheng Gao, Jintao Cui and Guoqun Zhao, *Weak Galerkin finite element methods for Sobolev equation*, J. Comput. Appl. Math. 317 (2017), 188–202.

[63] J.C. Zhang, K. Zhang and J.Z.Li, *A weak Galerkin method for diffraction gratings*, Appl. Anal. 96 (2017), no. 2, 190–214.

[64] T. Zhang and T. Lin, *A posteriori error estimate for a modified weak Galerkin method solving elliptic problems*, Numer. Methods Partial Differential Equations 33 (2017), no. 1, 381–398.

[65] J. Wang, X. Ye and R. Zhang, *Basics of weak Galerkin finite element methods*, (Chinese) Math. Numer. Sin. 38 (2016), no. 3, 289–308.

[66] X. Liu, J. Li and Z. Chen, *A weak Galerkin finite element method for the Oseen equations*, Adv. Comput. Math. 42 (2016), no. 6, 1473–1490.

[67] G. Chen, M. Feng and X. Xie, *Robust globally divergence-free weak Galerkin methods for Stokes equations*, J. Comput. Math. 34 (2016), no. 5, 549–572.

[68] Q. Zhang and R. Zhang, *A weak Galerkin mixed finite element method for second-order elliptic equations with Robin boundary conditions*, J. Comput. Math. 34 (2016), no. 5, 532–548.

[69] L. Mu, J. Wang, X. Ye and S. Zhao, *A new weak Galerkin finite element method for elliptic interface problems*, J. Comput. Phys. 325 (2016), 157–173.

[70] A. Zhu, T. Xu and Q. Xu, *Weak Galerkin finite element methods for linear parabolic integro-differential equations*, Numer. Methods Partial Differential Equations 32 (2016), no. 5, 1357–1377.

[71] C. Wang and J. Wang, *Discretization of div-curl systems by weak Galerkin finite element methods on polyhedral partitions*, J. Sci. Comput. 68 (2016), no. 3, 1144–1171.

[72] G. Chen and X. Xie, *A robust weak Galerkin finite element method for linear elasticity with strong symmetric stresses*, Comput. Methods Appl. Math. 16 (2016), no. 3, 389–408.

[73] L. Mu, J. Wang and X. Ye, *A hybridized formulation for the weak Galerkin mixed finite element method*, J. Comput. Appl. Math. 307 (2016), 335–345.

[74] X. Wang, Q. Zhai and R. Zhang, *The weak Galerkin method for solving the incompressible Brinkman flow*, J. Comput. Appl. Math. 307 (2016), 13–24.

[75] H. Zhang, Y. Zou, Y. Xu and Q. Zhai, *Weak Galerkin finite element method for second order parabolic equations*, Int. J. Numer. Anal. Model. 13 (2016), no. 4, 525–544.

[76] Q. Zhai, R. Zhang and L. Mu, *A new weak Galerkin finite element scheme for the Brinkman model*, Commun. Comput. Phys. 19 (2016), no. 5, 1409–1434.

[77] L. Mu, J. Wang and X. Ye, *A weak Galerkin generalized multiscale finite element method*, J. Comput. Appl. Math. 305 (2016), 68–81.

[78] W. Chen, F. Wang and Y. Wang, *Weak Galerkin method for the coupled Darcy-Stokes flow*, IMA J. Numer. Anal. 36 (2016), no. 2, 897–921.

[79] G. Chen and M. Feng, *A C^0-weak Galerkin finite element method for fourth-order elliptic problems.*, Numer. Methods Partial Differential Equations 32 (2016), no. 3, 1090–1104.

[80] F. Qin, M. Zha and F. Wang, *A two-level additive Schwarz preconditioning algorithm for the weak Galerkin method for the second-order elliptic equation* , Math. Probl. Eng. 2016, Art. ID 2685659, 6 pp.

[81] R. Wang, X. Wang and Q. Zhai, *A weak Galerkin finite element scheme for solving the stationary Stokes equations*, J. Comput. Appl. Math. 302 (2016), 171–185.

[82] F. Gao and G. Zhao, *Weak Galerkin finite element method for time dependent reaction-diffusion equation*, J. Comput. Anal. Appl. 21 (2016), no. 6, 1086–1102.

[83] J. Wang and X. Ye, *A weak Galerkin finite element method for the stokes equations*, Adv. Comput. Math. 42 (2016), no. 1, 155–174.

[84] V. Ginting, G. Lin and J. Liu, *On application of the weak Galerkin finite element method to a two-phase model for subsurface flow*, J. Sci. Comput. 66 (2016), no. 1, 225–239.

[85] J. Li, X. Wang and K. Zhang, *Multi-level Monte Carlo weak Galerkin method for elliptic equations with stochastic jump coefficients*, Appl. Math. Comput. 275 (2016), 181–194.

[86] H. Zhang, Y. Zou and S. Chai, *Weak Galerkin method with (r,r1,r1)-order finite elements for second order parabolic equations*, Appl. Math. Comput. 275 (2016), 24–40.

[87] Q. Zhai, R. Zhang and X. Wang, *A hybridized weak Galerkin finite element scheme for the Stokes equations*, Sci. China Math. 58 (2015), no. 11, 2455–2472.

[88] L. Mu, J. Wang and X. Ye, *A new weak Galerkin finite element method for the Helmholtz equation*, IMA J. Numer. Anal. 35 (2015), no. 3, 1228–1255.

[89] L. Mu, J. Wang, X. Ye and S. Zhang, *A weak Galerkin finite element method for the Maxwell equations*, J. Sci. Comput. 65 (2015), no. 1, 363–386.

[90] F. Gao, X. Wang and L. Mu, *A modified weak Galerkin finite element methods for convection-diffusion problems in 2D*, J. Appl. Math. Comput. 49 (2015), no. 1-2, 493–511.

[91] L. Chen, J. Wang and Y. Wang, *An auxiliary space multigrid preconditioner for the weak Galerkin method*, Comput. Math. Appl. 70 (2015), no. 4, 330–344.

[92] R. Zhang and Q. Zhai, *A weak Galerkin finite element scheme for the biharmonic equations by using polynomials of reduced order*, J. Sci. Comput. 64 (2015), no. 2, 559–585.

[93] C. Wang and J. Wang, *A hybridized weak Galerkin finite element method for the biharmonic equation*, Int. J. Numer. Anal. Model. 12 (2015), no. 2, 302–317.

[94] F. Gao and X. Wang, *A modified weak Galerkin finite element method for Sobolev equation*, J. Comput. Math. 33 (2015), no. 3, 307–322.

[95] B. Li and X. Xie, *A two-level algorithm for the weak Galerkin discretization of diffusion problems*, J. Comput. Appl. Math. 287 (2015), 179–195.

[96] L. Mu, J. Wang and X. Ye, *A weak Galerkin finite element method with polynomial reduction*, J. Comput. Appl. Math. 285 (2015), 45–58.

[97] M. Yang, *Couplings of mixed finite element and weak Galerkin methods for elliptic problems*, J. Appl. Math. Comput. 47 (2015), no. 1-2, 327–343.

[98] L. Mu, J. Wang and X. Ye, *Weak Galerkin finite element methods on polytopal meshes*, Int. J. Numer. Anal. Model. 12 (2015), no. 1, 31–53.

[99] L. Mu, X. Wang and X. Ye, *A modified weak Galerkin finite element method for the Stokes equations*, J. Comput. Appl. Math. 275 (2015), 79–90.

[100] G. Lin, J. Liu and F. Sadre-Marandi, *A comparative study on the weak Galerkin, discontinuous Galerkin, and mixed finite element methods*, J. Comput. Appl. Math. 273 (2015), 346–362.

[101] C. Wang and J. Wang, *An efficient numerical scheme for the biharmonic equation by weak Galerkin finite element methods on polygonal or polyhedral meshes*, Comput. Math. Appl. 68 (2014), no. 12, part B, 2314–2330.

[102] G. Lin, J. Liu , L. Mu and X. Ye, *Weak Galerkin finite element methods for Darcy flow: anisotropy and heterogeneity*, J. Comput. Phys. 276 (2014), 422–437.

[103] L. Mu, J. Wang and X. Ye, *A stable numerical algorithm for the Brinkman equations by weak Galerkin finite element methods.*, J. Comput. Phys. 273 (2014), 327–342.

[104] J. Wang and X. Ye, *A weak Galerkin mixed finite element method for second order elliptic problems*, Math. Comp. 83 (2014), no. 289, 2101–2126.

[105] A. Zhu, Q. Xu and Z. Jiang, *Characteristics weak Galerkin finite element methods for convection-dominated diffusion problems*, Abstr. Appl. Anal. 2014, Art. ID 102940, 8 pp.

[106] F. Gao and X. Wang, *A modified weak Galerkin finite element method for a class of parabolic problems*, J. Comput. Appl. Math. 271 (2014), 1–19.

[107] F. Gao and L. Mu, *On L^2 error estimate for weak Galerkin finite element methods for parabolic problems*, J. Comput. Math. 32 (2014), no. 2, 195–204.

[108] L. Mu, J. Wang and X. Ye, *Weak Galerkin finite element methods for the biharmonic equation on polytopal meshes*, Numer. Methods Partial Differential Equations 30 (2014), no. 3, 1003–1029.

[109] L. Chen, J. Wang and X. Ye, *A posteriori error estimates for weak Galerkin finite element methods for second order elliptic problems*, J. Sci. Comput. 59 (2014), no. 2, 496–511.

[110] L. Mu, J. Wang, X. Ye and S. Zhang, *A C^0-weak Galerkin finite element method for the biharmonic equation*, J. Sci. Comput. 59 (2014), no. 2, 473–495.

[111] L. Mu, J. Wang, X. Ye and S. Zhao, *A numerical study on the weak Galerkin method for the Helmholtz equation*, Commun. Comput. Phys. 15 (2014), no. 5, 1461–1479.

[112] R. Zhang, H. Song and N. Luan, *Weak Galerkin finite element method for valuation of American options*, Front. Math. China 9 (2014), no. 2, 455–476.

[113] A. Harris and S. Harris, *Superconvergence of weak Galerkin finite element approximation for second order elliptic problems by L^2-projections*, Appl. Math. Comput. 227 (2014), 610–621.

[114] Q. Li and J. Wang, *Weak Galerkin finite element methods for parabolic equations*, Numer. Methods Partial Differential Equations 29 (2013), no. 6, 2004–2024.

[115] L. Mu, J. Wang, Y. Wang and X. Ye, *A weak Galerkin mixed finite element method for biharmonic equations*, Numerical solution of partial differential equations: theory, algorithms, and their applications, 247–277, Springer Proc. Math. Stat., 45, Springer, New York, 2013.

[116] L. Mu, J. Wang, G. Wei and X. Ye, *Weak Galerkin methods for second order elliptic interface problems*, J. Comput. Phys. 250 (2013), 106–125.

[117] L. Mu, J. Wang, Y. Wang and X. Ye, *A computational study of the weak Galerkin method for second-order elliptic equations*, Numer. Algorithms 63 (2013), no. 4, 753–777.

[118] J. Wang and X. Ye, *A weak Galerkin finite element method for second-order elliptic problems*, J. Comput. Appl. Math. 241 (2013), 103–115.

[119] R Wang, R Zhang, X Zhang, Z Zhang, *Supercloseness analysis and polynomial preserving recovery for a class of weak Galerkin method*, arXiv preprint arXiv:1610.05626.

[120] S. Osher and R. Fedkiw, *Level set methods and dynamic implicit surfaces*, Applied Mathematical Sciences, vol. 153, Springer-Verlag, New York, 2003.

[121] J. A. Sethian, *Level set methods*, Cambridge Monographs on Applied and Computational Mathematics, vol. 3, Cambridge University Press, Cambridge, 1996, Evolving interfaces in geometry, fluid mechanics, computer vision, and materials science.

[122] M. G. Armentano and R. G. Durán, *Asymptotic lower bounds for eigenvalues by nonconforming finite element methods*, Electron. Trans. Numer. Anal., 17 (2004), 93–101.

[123] I. Babuška and W. C. Rheinboldt, *Error estimates for adaptive finite element computations*, SIAM J. Numer. Anal., 15 (1978), 736–754.

[124] I. Babuška and W. C. Rheinboldt, *A-posteriori error estimates for the finite element method*, Int. J. Numer. Meth. Engng., 12 (1978), 1597–1615.

[125] I. Babuška and T. Strouboulis, *The finite element method and its reliability*, University Press, New York, 2001.

[126] P. Binev, W. Dahmen, and R. DeVore, *Adaptive finite element methods with convergence rates*, Numer. Math., 97 (2004), 219–268.

[127] R. Verfürth, *A review of a posteriori error estimation and adaptive mesh-refinement techniques*, Wiley and Teubner, New York, 1996.

[128] R. Verfürth, *A posteriori error estimation techniques for finite element methods*, Oxford University Press, Oxford, 2013.

[129] C. Carstensen and S. Bartels, *Each averaging technique yields reliable a posteriori error control in FEM on unstructured grids. I. Low order conforming, nonconforming, and mixed FEM*, Math. Comp., 71 (2002), 945–969.

[130] H. Wu and Z. Zhang, *Can we have superconvergent gradient recovery under adaptive meshes?*, SIAM J. Numer. Anal., 45 (2007), 1701–1722.

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

[132] 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 (1992), 1331–1364.

[133] 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 (1992), 1365–1382.

[134] Z. Cai and S. Zhang, *Recovery-based error estimator for interface problems: conforming linear elements*, SIAM J. Numer. Anal. 47 (2009), no. 3, 21322156.

[135] Z. Chen and S. Dai, *On the efficiency of adaptive finite element methods for elliptic problems with discontinuous coefficients*, SIAM J. Sci. Comput. 24 (2002), no. 2, 443462 (electronic).

[136] R. B. Kellogg, *On the Poisson equation with intersecting interfaces*, Applicable Anal. 4 (1974/75), 101129, Collection of articles dedicated to Nikolai Ivanovich Muskhelishvili.

[137] P. Morin, R. H. Nochetto, and K. G. Siebert, *Convergence of adaptive finite element methods*, SIAM Rev. 44 (2002), no. 4, 631658 (electronic) (2003), Revised reprint of Data oscillation and convergence of adaptive FEM [SIAM J. Numer. Anal. 38 (2000), no. 2, 466488 (electronic); MR1770058 (2001g:65157)].

ABSTRACT

POLYNOMIAL PRESERVING RECOVERY FOR WEAK GALERKIN METHODS AND THEIR APPLICATIONS

by

REN ZHAO

December 2017

Advisor: Dr. Zhimin Zhang

Major: Mathematics

Degree: Doctor of Philosophy

Gradient recovery technique is widely used to reconstruct a better numerical gradient from a finite element solution, for mesh smoothing, a posteriori error estimate and adaptive finite element methods. The PPR technique generates a higher order approximation of the gradient on a patch of mesh elements around each mesh vertex. It can be used for different finite element methods for different problems. This dissertation presents recovery techniques for the weak Galerkin methods and as well as applications of gradient recovery on various of problems, including elliptic problems, interface problems, and Stokes problems.

Our first target is to develop a boundary strategy for the current PPR algorithm. The current accuracy of PPR near boundaries is not as good as that in the interior of the domain. It might be even worse than without recovery. Some special treatments are needed to improve the accuracy of PPR on the boundary. In this thesis, we present two boundary recovery strategies to resolve the problem caused by boundaries. Numerical experiments indicate that both of the newly proposed strategies made an improvement to the original PPR.

Our second target is to generalize PPR to the weak Galerkin methods. Different from the standard finite element methods, the weak Galerkin methods use a different set of degrees of freedom. Instead of the weak gradient information, we are able to obtain the recovered gradient information for the numerical solution in the generalization of PPR. In the PPR process, we are also able to recover the function value at the nodal points which will produce a global continuous solution instead of piecewise continuous function u_h.

Our third target is to apply our proposed strategy and WGPPR to interface problems. We treat an interface as a boundary when performing gradient recovery, and the jump condition on the interface can be well captured by the function recovery process.

In addition, adaptive methods based on WGPPR recovery type a posteriori error estimator is proposed and numerically tested in this thesis. Application on the numerical examples validate the effectiveness and robustness of our algorithm. Furthermore, WGPPR has been applied to 3D problem and Stokes problem as well. Superconvergent phenomenon is again observed.

AUTOBIOGRAPHICAL STATEMENT

Ren Zhao

Education

- Ph.D. in Mathematics, 2017
  Wayne State University, Detroit, Michigan, USA
- M.A. in Statistics, 2012
  Wayne State University, Detroit, Michigan, USA
- B.S. in Information and Computing Science, 2006
  Xi'an Jiaotong University, China

Selected List of Awards and Scholarships

1. Rumble Fellowship, Department of Mathematics, Wayne State University, 2009-2010.
2. Graduate Assistant Excellence in Teaching Award, Wayne State University, 2012-2013.

Selected List of Publications

1. K Wang, Y Shang, and R Zhao, Optimal error estimates of the penalty method for the linearized viscoelastic flows, International Journal of Computer Mathematics 87 (14), 3236-3253.
2. H Zheng, F Shi, Y Hou, J Zhao, Y Cao, and R Zhao, New local and parallel finite element algorithm based on the partition of unity, Journal of Mathematical Analysis and Applications 435 (2016), no. 1, 1–19.
3. W He, Z Zhang, and R Zhao, The Highest Superconvergence of the Tri-linear Element for Schrödinger Operator with Singularity, Journal of Scientific Computing 66 (2016), no. 1, 1–18.
4. H Guo, Z Zhang, R Zhao, and Q Zou, Polynomial preserving recovery on boundary, Journal of Computational and Applied Mathematics 307(2016), 119-133.
5. H. Guo, Z. Zhang, and R. Zhao, Hessian Recovery for Finite Element Methods, Mathematics of Computation 86 (2017), no. 306, 1671–1692.
6. H. Guo, Z. Zhang and R. Zhao, Superconvergent two-grid schemes for Elliptic eigenvalue problems, Journal of Scientific Computing 70 (2017), no. 1, 125–148.
7. J Yu, F Shi, H Zheng and R. Zhao, Two-grid finite element method for the stabilization of mixed Stokes-Darcy model, Discrete and Continuous Dynamical Systems, Series B, 2017, accepted.
8. F Shi, H Zheng, Y Cao, J Li and R Zhao, A fast numerical method for solving coupled Burgers' equations, Numerical Methods for Partial Differential Equations. (2017) doi:10.1002/num.22160.