Journal of Computational Mathematics
Vol.36, No.6, 2018, 866–880.
http://www.global-sci.org/jcm
doi:10.4208/jcm.1705-m2016-0744

AN OVER-PENALIZED WEAK GALERKIN METHOD FOR SECOND-ORDER ELLIPTIC PROBLEMS*

Kaifang Liu, Lunji Song and Shuangfeng Zhou
School of Mathematics and Statistics, and Key Laboratory of Applied Mathematics and Complex Systems (Gansu Province), Lanzhou University, Lanzhou 730000, China
Email: liukf17@lzu.edu.cn, song@lzu.edu.cn, zhoushf14@lzu.edu.cn

Abstract

The weak Galerkin (WG) finite element method was first introduced by Wang and Ye for solving second order elliptic equations, with the use of weak functions and their weak gradients. The basis function spaces depend on different combinations of polynomial spaces in the interior subdomains and edges of elements, which makes the WG methods flexible and robust in many applications. Different from the definition of jump in discontinuous Galerkin (DG) methods, we can define a new weaker jump from weak functions defined on edges. Those functions have double values on the interior edges shared by two elements rather than a limit of functions defined in an element tending to its edge. Naturally, the weak jump comes from the difference between two weak functions defined on the same edge. We introduce an over-penalized weak Galerkin (OPWG) method, which has two sets of edge-wise and element-wise shape functions, and adds a penalty term to control weak jumps on the interior edges. Furthermore, optimal *a priori* error estimates in H^1 and L^2 norms are established for the finite element $(\mathbb{P}_k(K), \mathbb{P}_k(e), RT_k(K))$. In addition, some numerical experiments are given to validate theoretical results, and an incomplete LU decomposition has been used as a preconditioner to reduce iterations from the GMRES, CG, and BICGSTAB iterative methods.

Mathematics subject classification: 65N15, 65N30.
Key words: Weak Galerkin, Over-penalized, Finite element methods, Second-order elliptic equation.

1. Introduction

A weak Galerkin (WG) method was first introduced by Wang and Ye [1] for solving general second order elliptic equations, and a series of related numerical analysis and numerical applications to the method are conducted in Ref. [2], which show the WG method efficient and reliable in scientific computing. In general, the WG method refers to a finite element method where differential operators can be approximated by the linear space of vector polynomial functions. The original WG schemes include the polynomial combination $(\mathbb{P}_k(K), \mathbb{P}_k(e), RT_k(K))$ and $(\mathbb{P}_k(K), \mathbb{P}_{k+1}(e), [\mathbb{P}_{k+1}(K)]^d)$ for $k \geq 0$, where $RT_k(K)$ represents the kth order Raviart-Thomas elements [3], $[\mathbb{P}_k(K)]^d$ is a set of polynomials of order no more than k, and d is a dimension of space. To get flexible basis functions and to maintain some kind of weak continuity, Wang and Ye [4, 5] add a stabilizer in variational forms of PDEs. Moreover, WG has been developed for solving more applications, such as elliptic interface problems, Stokes, Helmholtz, Maxwell, etc [6–10].

* Received September 19, 2016 / Revised version received February 25, 2017 / Accepted May 27, 2017 / Published online August 7, 2018 /

For second-order elliptic problems, the weak functions in weak finite element spaces are expressed in a form of $v = (v_0, v_b)$ with v_0 representing the value of v in the interior of each element and v_b on the edges of each element. Generally, polynomial combination $\mathbb{P}_l(K) \times \mathbb{P}_m(e)$ were chosen to be weak finite element spaces, where e denotes the edges or faces of element K, l and m are non-negative integers. The approximation spaces $RT_k(K)$ or $[\mathbb{P}_k(K)]^d$ are chosen for weak differential operators. As far as we know, all WG schemes are based on the fact that weak function v_b along every interior edge is single-valued, however, in this work, we consider the function v_b double-valued on interior edges and for every element and its corresponding edges, the pair of functions (v_0, v_b) are separately defined well.

Due to the treatment of the jumps appearing in the DG methods, we introduce an over-penalized weak Galerkin (OPWG) method for second-order elliptic problems by using a new stabilized term of weak jumps. In other words, shape functions have two traces along each interior edge shared by two neighboring elements, where a weak jump could be generated. Different from the definition of jump in interior-penalty discontinuous Galerkin (IPDG) [11] finite element methods, weak jump comes from the weak functions rather than a limit passing from an interior domain to its edges. We introduce a penalty on the weak jumps, characterizing a new WG method and strengthening the stability and analysis. Therefore, we can also present a new DG method with the use of the definitions of weak functions, because the functions have discontinuity just on the interior edges.

Our main idea is to connect WG with DG methods, and investigate the possibility of penalized methods. In the present work, we do not modify the definition of weak gradient. To change the definition of weak gradient, the reader is referred to a modified WG method [12], in which Wang and Malluwawadu developed a new weak gradient operator defined on piecewise polynomial spaces. We keep the weak finite element spaces and weak gradient operator unchanged except that the shape functions along the interior edges are double-valued. Therefore, many primary results about the WG methods developed before can be easily applied to the present penalized WG method. Furthermore, due to the complete independence of elementwises shape functions similar as in DG, OPWG seems more convenient in parallel computing than the WG methods.

For the sake of simple and easy presentation of the new method, we consider the following second order elliptic equation with nonhomogeneous Dirichlet boundary condition:

$$-\nabla \cdot A\nabla u = f, \qquad \text{in } \Omega, \tag{1.1}$$

$$u = g, \qquad \text{on } \partial\Omega, \tag{1.2}$$

where Ω is a polygonal or polyhedral domain in $\mathbb{R}^d\,(d = 2, 3)$, $f \in L^2(\Omega)$ and A is a symmetric and positive definite matrix-valued function in Ω, i.e., there exist two positive numbers $\lambda_1, \lambda_2 > 0$ such that

$$\lambda_1 \xi^t \xi \le \xi^t A \xi \le \lambda_2 \xi^t \xi, \qquad \forall\, \xi \in \mathbb{R}^d,$$

where ξ is a column vector and ξ^t means the transpose of ξ.

The paper is organized as follows. In Section 2, we give some preliminary notations and definitions. In Section 3, the OPWG scheme was introduced. In Section 4, optimal error analysis in H^1 and L^2 norms is established. In Section 5, numerical experiments are conducted to confirm the theoretical results. Some conclusions and remarks are given in the final section.

2. Notations and Preliminaries

We begin with some basic notations and definitions. Let $\mathcal{T}_h$ be a shape regular partition [13] of the domain Ω, $\mathcal{E}$ be all edges(or faces), and $\mathcal{E}_\mathcal{I}$ be all interior edges. If there is no further specification, we always assume elements in $\mathcal{T}_h$ are simplex. The sign h_K means the diameter of an element $K \in \mathcal{T}_h$, and $h := \max_K h_K$ represents the maximum mesh size for $\mathcal{T}_h$. Let $\mathbb{P}_k(K)$ be a space of piecewise polynomials whose degrees are no more than k in each $K \in \mathcal{T}_h$. Similarly, $\mathbb{P}_k(e)$ is denoted by a space of piecewise polynomials on e whose degrees are no more than k on each $e \in \mathcal{E}$.

Now on any element K, we define a weak function by $v = \{v_0, v_b\}$ such that $v_0 \in L^2(K)$ and $v_b \in L^2(\partial K)$. The first part v_0 can be understood as the value of v in K, and the second part v_b is the boundary value of K. Define by $W(K)$ the space of weak functions on K, i.e.,

$$W(K) = \Big\{ v = (v_0, v_b) : v_0 \in L^2(K), v_b \in L^2(\partial K) \Big\}.$$

As in [1], we define the same local weak gradient operator for any $v \in W(K)$.

Definition 2.1. *For any $v \in W(K)$, the weak gradient operator, denoted by ∇_w, is defined as the unique vector-value function $(\nabla_w v) \in [H^1(K)]^d$ satisfying the following equation:*

$$(\nabla_w v,\ \boldsymbol{q})_K = -(v_0,\ \nabla \cdot \boldsymbol{q})_K + \langle v_b, \boldsymbol{q} \cdot \boldsymbol{n} \rangle_{\partial K}, \quad \forall\ \boldsymbol{q} \in [H^1(K)]^d,$$

where $\boldsymbol{n}$ is the outward normal direction to ∂K, $(\cdot,\cdot)_K$ is the inner product in $L^2(K)$, and $\langle \cdot,\cdot \rangle_{\partial K}$ is the inner product in $L^2(\partial K)$.

We can define a discrete weak gradient operator by approximating ∇_w in a space of vector polynomials.

Definition 2.2. *For any $v \in W(K)$ and $K \in \mathcal{T}_h$, the discrete weak gradient operator, denoted by $\nabla_{w,k,K}$, is defined as the unique polynomial $(\nabla_{w,k,K} v) \in \boldsymbol{V}_k(K)$ satisfying the following equation:*

$$(\nabla_{w,k,K} v,\ \boldsymbol{q})_K = -(v_0,\ \nabla \cdot \boldsymbol{q})_K + \langle v_b, \boldsymbol{q} \cdot \boldsymbol{n} \rangle_{\partial K}, \quad \forall\ \boldsymbol{q} \in \boldsymbol{V}_k(K),$$

where $\boldsymbol{V}_k(K)$ is a subspace of vector-valued polynomials of degree no more than k in element K. For simplicity of notation, we always denote by ∇_w the discrete weak gradient operator $\nabla_{w,k,K}$. Here we take $\boldsymbol{V}_k(K) = RT_k(K)$.

Define the weak Galerkin finite element space associated with $\mathcal{T}_h$ as follows

$$\begin{aligned} V_h = \Big\{ &(v_0,\ v_b) : v_0|_K \in \mathbb{P}_j(K),\ K \in \mathcal{T}_h, \\ &v_b|_e \in \mathbb{P}_l(e) \times \mathbb{P}_l(e), e \in \mathcal{E}_\mathcal{I};\ v_b|_e \in \mathbb{P}_l(e), e \in \partial\Omega,\ j,\ l \geq 0 \Big\}, \end{aligned} \tag{2.1}$$

$$V_h^0 = \Big\{ (v_0,\ v_b) : v \in V_h,\ v_b = 0 \text{ on } \partial\Omega \Big\}, \tag{2.2}$$

where v_b is a double-valued function on each interior edge. Since elements $RT_k(k \geq 0)$ are mainly analyzed, the shape of element K shall be triangle or tetrahedron. Here we set $j = l = k$.

Let the normal vector space $\mathcal{N}$ consist of all unit normal vectors $\boldsymbol{n}_e$ associated with each edge $e \in \mathcal{E}$. When $e \in \mathcal{E}_\mathcal{I}$, $\boldsymbol{n}_e$ can be any normal pointing from one element to its neighboring element; when $e \in \partial\Omega$, $\boldsymbol{n}_e$ is a unit normal vector exterior to the boundary. Assume elements

K_1^e and K_2^e are neighbors and share one common edge e, so there are two traces v_b along e. We denote by $[\![v_b]\!]$ the weak jump for v_b:

$$[\![v_b]\!] = v_b|_{K_1^e} - v_b|_{K_2^e}, \qquad \forall\ e = \partial K_1^e \cap \partial K_2^e.$$

We also extend the definition of jump to sides that belong to the boundary, i.e., $[\![v_b]\!] = v_b|_{K_1^e}$, $\forall\ e = \partial K_1^e \cap \partial\Omega$. In addition, we denote by $|e|$ length or area of edge/face.

Let $(\bar{x}, \bar{y})$ be a barycenter of any element K and $X = x - \bar{x}$, $Y = y - \bar{y}$. Then for two-dimensional problems, the local basis for $RT_0(K)$ elements are

$$\begin{bmatrix}1\\0\end{bmatrix}, \quad \begin{bmatrix}0\\1\end{bmatrix}, \quad \begin{bmatrix}X\\Y\end{bmatrix}, \tag{2.3}$$

and the local basis for $RT_1(K)$ element are

$$\begin{bmatrix}1\\0\end{bmatrix}, \quad \begin{bmatrix}0\\1\end{bmatrix}, \quad \begin{bmatrix}X\\0\end{bmatrix}, \quad \begin{bmatrix}0\\X\end{bmatrix}, \quad \begin{bmatrix}Y\\0\end{bmatrix}, \quad \begin{bmatrix}0\\Y\end{bmatrix}, \quad \begin{bmatrix}X^2\\XY\end{bmatrix}, \quad \begin{bmatrix}XY\\Y^2\end{bmatrix}. \tag{2.4}$$

To investigate the approximation properties of the finite element space V_h and weak gradient space $\boldsymbol{V}_k(K)$, we introduce the following three standard L^2 projections:

$$\begin{aligned}
&Q_0 : L^2(K) \to \mathbb{P}_k(K), && \forall\, K \in \mathcal{T}_h,\\
&Q_b : L^2(e) \to \mathbb{P}_k(e), && \forall\, e \in \mathcal{E},\\
&\mathbb{Q}_h : [L^2(K)]^2 \to RT_k(K), && \forall\, K \in \mathcal{T}_h.
\end{aligned}$$

We combine Q_0 with Q_b by writing as $Q_h = \{Q_0, Q_b\}$. To investigate error estimates, we introduce a projection operator Π_h, which is widely used in the mixed finite element method, and satisfies the following property: for any $\boldsymbol{\tau} \in H(\mathrm{div}, \Omega)$, $\Pi_h\boldsymbol{\tau} \in H(\mathrm{div}, \Omega)$; and on each element $K \in \mathcal{T}_h$, one has $\Pi_h\boldsymbol{\tau} \in RT_k(K)$ satisfying

$$(\nabla \cdot \boldsymbol{\tau}, v_0) = (\nabla \cdot \Pi_h\boldsymbol{\tau}, v_0), \quad \forall\, v_0 \in \mathbb{P}_k(K).$$

Moreover, the following identity holds (see [1]):

$$\nabla_w(Q_h v) = \mathbb{Q}_h(\nabla v), \quad \forall\, v \in H^1(K). \tag{2.5}$$

3. The Over-penalized Weak Galerkin Scheme

For any w, $v \in V_h$, we define the following bilinear form

$$a(w, v) := (A\nabla_w w_h, \nabla_w v) + \sum_{e\in\mathcal{E}_{\mathcal{I}}} \frac{1}{|e|^{\beta_0}} \int_e [\![w_b]\!][\![v_b]\!],$$

where $\beta_0 \geq 1$ is selected as conventionally named after an over-penalized parameter. Note that this bilinear form does not need a penalty factor, which often appears in the interior penalty discontinuous Galerkin methods.

Algorithm 3.1. *A weak Galerkin approximation for* (1.1)-(1.2) *is to seek* $u_h = (u_0, u_b) \in V_h$ *satisfying* $u_b = Q_b g$ *on* $\partial\Omega$ *and such that*

$$a(u_h, v) = (f, v_0), \quad \forall\ v = (v_0, v_b) \in V_h^0. \tag{3.1}$$

Next, we justify the well-posedness of the scheme (3.1). For any $v \in V_h$, an energy norm is written as

$$|||v||| := \sqrt{a(v,v)}. \tag{3.2}$$

It is easy to see that $|||\cdot|||$ define a semi-norm in V_h. Moreover, it is a norm in V_h^0. Indeed, it suffices to check the positivity property for $|||\cdot|||$. To this end, assume that $v \in V_h^0$ and $|||v||| = 0$. It follows that

$$(A\nabla_w v, \nabla_w v) + \sum_{e\in\mathcal{E}_\mathcal{I}} \frac{1}{|e|^{\beta_0}} \int_e [\![v_b]\!]^2 = 0,$$

which implies that $\nabla_w v = 0$ on each element K and $[\![v_b]\!]_e = 0$ on e. Thus, $v_0 = v_b = const$ on every $K \in \mathcal{T}_h$. On a common edge e shared by two neighboring elements K_1 and K_2, we have $v_b|_{e\cap\partial K_1} = v_b|_{e\cap\partial K_2}$. Together with $v_b = 0$ on $\partial\Omega$, it follows that $v_0 = v_b = 0$.

Lemma 3.1. *The over-penalized weak Galerkin finite element scheme* (3.1) *has one and only one solution.*

Proof. It suffices to prove the uniqueness. If $u_h^{(1)}$ and $u_h^{(2)}$ are two solutions of (3.1), then $e_h = u_h^{(1)} - u_h^{(2)}$ would satisfy the following equation

$$a(e_h, v) = 0, \quad \forall\, v \in V_h^0.$$

Note that $e_h \in V_h^0$. Then by taking $v = e_h$ in the above equation we arrive at

$$|||e_h|||^2 = a(e_h, e_h) = 0.$$

It follows that $e_h \equiv 0$, or equivalently, $u_h^{(1)} \equiv u_h^{(2)}$. This completes the proof of the lemma. □

Remark 3.1. The OPWG method preserves local mass conservation. The model problem (1.1) can be rewritten in a conservative form as follows:

$$\nabla \cdot q = f, \quad q = -A\nabla u.$$

Let K be any control volume. Integrating the first equation over K yields the following integral form of mass conservation:

$$\int_{\partial K} q \cdot \boldsymbol{n} = \int_K f. \tag{3.3}$$

Following a similar proof in [1], one can show that the numerical approximation from the over-penalized weak Galerkin finite element method for (1.1) retains the mass conversation property (3.3) with a numerical flux $q_h = \mathbb{Q}_h(A\nabla_w u_h)$.

4. Convergence Theory

The goal of this section is to establish some error estimates for the weak Galerkin finite element solution u_h arising from (3.1). The error estimates will be conducted in two natural norms: the energy norm as defined in (3.2) and the standard L^2 norm. We first state an approximation property of the operator Π_h, which has been proved in [1].

Lemma 4.1. *Assume that Π_h be the local projection operator satisfying $\Pi_h \boldsymbol{\tau} \in RT_k(K)$, $\boldsymbol{\tau} \in H(div, \Omega)$, and $Q_h = \{Q_0, Q_b\}$ defined in Section 2. For $u \in H^{k+2}(\Omega)$ with $k \geq 0$, we have*

$$\|\Pi_h(A\nabla u) - A\nabla_w(Q_h u)\| \leq Ch^{k+1}\|u\|_{k+2}. \tag{4.1}$$

Next, we state an important result for the projection operator Π_h.

Lemma 4.2. *Let $\boldsymbol{\tau} \in H(div, \Omega)$ be a smooth vector-valued function and Π_h be the local projection operator defined in Section 2. Then, the following identity holds true*

$$\sum_{K\in\mathcal{T}_h}(-\nabla\cdot\boldsymbol{\tau}, v_0)_K = \sum_{K\in\mathcal{T}_h}(\Pi_h\boldsymbol{\tau}, \nabla_w v_h)_K - \sum_{e\in\mathcal{E}_\mathcal{I}}\langle[\![v_b]\!], \Pi_h\boldsymbol{\tau}\cdot\boldsymbol{n}_e\rangle_e, \quad v_h \in V_h^0. \tag{4.2}$$

Proof. It follows from the definitions of Π_h and ∇_w that

$$\begin{aligned}
\sum_{K\in\mathcal{T}_h}(-\nabla\cdot\boldsymbol{\tau}, v_0)_K &= \sum_{K\in\mathcal{T}_h}(-\nabla\cdot\Pi_h\boldsymbol{\tau}, v_0)_K \\
&= \sum_{K\in\mathcal{T}_h}(\Pi_h\boldsymbol{\tau}, \nabla_w v_h)_K - \sum_{K\in\mathcal{T}_h}\langle v_b, \Pi_h\boldsymbol{\tau}\cdot\boldsymbol{n}\rangle_{\partial K} \\
&= \sum_{K\in\mathcal{T}_h}(\Pi_h\boldsymbol{\tau}, \nabla_w v_h)_K - \sum_{e\in\mathcal{E}_\mathcal{I}}\langle[\![v_b]\!], \Pi_h\boldsymbol{\tau}\cdot\boldsymbol{n}_e\rangle_e,
\end{aligned}$$

where the last equality results from the definition of weak jump and unit outer normal vector $\boldsymbol{n_e}$ in Section 2, thus the proof is completed. □

Testing (1.1) with v_0 of $v = (v_0, v_b) \in V_h^0$ and using (4.2) leads to

$$\sum_{K\in\mathcal{T}_h}(\Pi_h A\nabla u, \nabla_w v_h)_K - \sum_{e\in\mathcal{E}_\mathcal{I}}\langle[\![v_b]\!], \Pi_h A\nabla u\cdot\boldsymbol{n}_e\rangle_e = (f, v_0), \tag{4.3}$$

which can be rewritten as

$$\begin{aligned}
&\sum_{K\in\mathcal{T}_h}(A\nabla_w Q_h u, \nabla_w v_h)_K + \sum_{K\in\mathcal{T}_h}(\Pi_h A\nabla u - A\nabla_w Q_h u, \nabla_w v_h)_K \\
&\qquad - \sum_{e\in\mathcal{E}_\mathcal{I}}\langle[\![v_b]\!], \Pi_h A\nabla u\cdot\boldsymbol{n}_e\rangle_e = (f, v_0).
\end{aligned} \tag{4.4}$$

Set $e_h := (e_0, e_b) = (Q_0 u - u_0, Q_b u - u_b)$. Subtracting (3.1) from (4.4) leads to

$$\begin{aligned}
a(e_h, v_h) &= (A\nabla_w e_h, \nabla_w v_h) + \sum_{e\in\mathcal{E}}\frac{1}{|e|^{\beta_0}}\int_e [\![e_b]\!][\![v_b]\!] \\
&= \sum_{e\in\mathcal{E}_\mathcal{I}}\langle[\![v_b]\!], \Pi_h A\nabla u\cdot\boldsymbol{n}_e\rangle_e - \sum_{K\in\mathcal{T}_h}(\Pi_h A\nabla u - A\nabla_w Q_h u, \nabla_w v_h)_K,
\end{aligned} \tag{4.5}$$

which is called the *error equation* for the over-penalized weak Galerkin approximation (3.1). Then we give a convergence theorem for the error in the energy norm as follows.

Theorem 4.1. *Suppose that the exact solution u of* (1.1)-(1.2) *belongs to $H^{k+2}(\Omega)$ with $k \geq 0$. Then the error e_h between the L^2 projection of exact solution $Q_h u \in V_h$ and WG solution $u_h \in V_h$ satisfies*

$$|\!|\!| e_h |\!|\!| \leq C\left(h^{k+1} + h^{\frac{\beta_0-1}{2}}\right)\|u\|_{k+2,\Omega}, \tag{4.6}$$

where C is a positive constant independent of h and $\beta_0 > 1$.

Proof. From equation (4.5), we obtain by the Cauchy-Schwarz inequality that for any $v_h \in V_h^0$,

$$
\begin{aligned}
|a(e_h, v_h)| &\le \left|\sum_{e\in\mathcal{E}_\mathcal{I}} \langle [\![v_b]\!], \Pi_h A\nabla u\cdot \boldsymbol{n}_e\rangle_e\right| + \left|\sum_{K\in\mathcal{T}_h} (\Pi_h A\nabla u - A\nabla_w Q_h u, \nabla_w v_h)_K\right| \\
&\le \sum_{e\in\mathcal{E}_\mathcal{I}} \|[\![v_b]\!]\|_e \|\Pi_h A\nabla u\cdot \boldsymbol{n}_e\|_e + \sum_{K\in\mathcal{T}_h} \|\Pi_h A\nabla u - A\nabla_w Q_h u\|_K \|\nabla_w v_h\|_K \\
&\le \sum_{e\in\mathcal{E}_\mathcal{I}} \|[\![v_b]\!]\|_e \|\Pi_h A\nabla u\cdot \boldsymbol{n}_e\|_e + Ch^{k+1}\|u\|_{k+2,\Omega} \,|\!|\!|\, v_h \,|\!|\!|\, .
\end{aligned}
$$

We only need to estimate the first term in the above inequality. By using trace inequalities in [13], we arrive at

$$
\begin{aligned}
&\sum_{e\in\mathcal{E}_\mathcal{I}} \|[\![v_b]\!]\|_e \|\Pi_h A\nabla u\cdot \boldsymbol{n}_e\|_e \le \sum_{K\in\mathcal{T}_h} C h_K^{\frac{\beta_0}{2}} \,|\!|\!|\, v_h \,|\!|\!|_K \, h_K^{-\frac{1}{2}} \|\nabla u\|_K \\
&\le \sum_{K\in\mathcal{T}_h} C h_K^{\frac{\beta_0-1}{2}} \,|\!|\!|\, v_h \,|\!|\!|_K \, \|u\|_{k+2,K} \le Ch^{\frac{\beta_0-1}{2}} \|u\|_{k+2} \,|\!|\!|\, v_h |\!|\!|,
\end{aligned}
$$

which completes the proof of (4.6) by replacing the v_h by e_h. □

In the rest of the section, we shall derive an optimal-order error estimate for the over-penalized weak Galerkin finite element scheme (3.1) in the L^2-norm by using a duality argument as in the standard Galerkin finite element methods. To this end, we consider a dual problem with a homogeneous Dirichlet boundary condition to seek a solution $w \in H_0^1(\Omega)$ satisfying

$$
-\nabla\cdot(A\nabla w) = e_0, \quad \text{in } \Omega, \tag{4.7}
$$

Assume that the above dual problem has the usual H^2-regularity. This means that there exists a constant C such that

$$
\|w\|_2 \le C\|e_0\|. \tag{4.8}
$$

Theorem 4.2. *Under the condition of Theorem* 4.1*, and assume that the dual problem* (4.7) *has the* H^2*-regularity. Then, the error* $e_h \in V_h$ *in the* L^2*-norm has the following estimate*

$$
\|e_0\| \le Ch^{k+2}\|f\|_k + C\left(h^{k+2} + h^{1+\frac{\beta_0-1}{2}} + h^{\beta_0-1}\right)\|u\|_{k+2}, \tag{4.9}
$$

where C *is a positive constant independent of* h *and* $\beta_0 > 1$*.*

Proof. By testing (4.7) with the weak function e_0 we obtain

$$
\begin{aligned}
\|e_0\|^2 &= \sum_{K\in\mathcal{T}_h} (-\nabla\cdot(A\nabla w), e_0)_K \\
&= \sum_{K\in\mathcal{T}_h} (\Pi_h(A\nabla w), \nabla_w e_h)_K - \sum_{e\in\mathcal{E}_\mathcal{I}} \langle [\![e_b]\!], \Pi_h(A\nabla w)\cdot \boldsymbol{n}_e\rangle_e \\
&=: I + II,
\end{aligned} \tag{4.10}
$$

where

$$
\begin{aligned}
I &= (\Pi_h(A\nabla w), \nabla_w(Q_h u - u_h)) \\
&= (\Pi_h(A\nabla w), \mathbb{Q}_h(\nabla u) - \nabla_w u_h) = (\Pi_h(A\nabla w), \nabla u - \nabla_w u_h) \\
&= (\Pi_h(A\nabla w) - A\nabla w, \nabla u - \nabla_w u_h) + (A\nabla w, \nabla u - \nabla_w u_h).
\end{aligned}
$$

From Theorem 4.1 and inequality (4.8), the two terms in the above equation can be bounded as follows

$$
\begin{aligned}
&|(\Pi_h(A\nabla w)-A\nabla w,\nabla u-\nabla_w u_h)|\\
\le&|(\Pi_h(A\nabla w)-A\nabla w,\nabla u-\nabla_w(Q_h u))|\\
&+|(\Pi_h(A\nabla w)-A\nabla w,\nabla_w(Q_h u)-\nabla_w u_h)|\\
\le&\|\Pi_h(A\nabla w)-A\nabla w\|(\|\nabla u-\nabla_w(Q_h u)\|+\|\nabla_w(Q_h u)-\nabla_w u_h\|)\\
\le&Ch(\|\nabla u-\nabla_w(Q_h u)\|+\|\nabla_w e_h\|)\|e_0\|\\
\le&C\left(h^{k+2}+h^{\frac{\beta_0-1}{2}+1}\right)\|u\|_{k+2}\|e_0\|.
\end{aligned}
$$

Applying the fact $[\![Q_b w]\!]_e=0$ for $w\in H^1(\Omega)$ in the following term, we have

$$
\begin{aligned}
&|(A\nabla w,\nabla u-\nabla_w u_h)|\\
=&|(A\nabla w,\nabla u)-(A\nabla w,\nabla_w uh)|\\
=&|(A\nabla w,\nabla u)-(A(\nabla w-\mathbb{Q}_h(\nabla w)),\nabla_w u_h)-(A\nabla_w(Q_h w),\nabla_w u_h)|\\
=&|(A\nabla w,\nabla u)-(A\nabla_w(Q_h w),\nabla_w u_h)-(A(\nabla w-\mathbb{Q}_h(\nabla w)),\nabla_w u_h-\nabla u)\\
&-(\nabla w-\mathbb{Q}_h(\nabla w),A\nabla u-\mathbb{Q}_h(A\nabla u))|\\
=&|(A\nabla w,\nabla u)-(A\nabla_w(Q_h w),\nabla_w u_h)-\sum_{e\in\mathcal{E}}\frac{1}{|e|^{\beta_0}}\int_e[\![u_b]\!][\![Q_b w]\!]\\
&-(A(\nabla w-\mathbb{Q}_h(\nabla w)),\nabla_w u_h-\nabla u)-(\nabla w-\mathbb{Q}_h(\nabla w),A\nabla u-\mathbb{Q}_h(A\nabla u))|.
\end{aligned}
$$

Thanks to the facts that $(A\nabla w,\nabla u)=(f,w)$ and $a(u_h,Q_h w)=(f,Q_0 w)$, the above equation can be written as

$$
\begin{aligned}
&|(A\nabla w,\nabla u-\nabla_w u_h)|\\
=&|(f,w)-(f,Q_0 w)-(A(\nabla w-\mathbb{Q}_h(\nabla w)),\nabla_w u_h-\nabla u)\\
&-(\nabla w-\mathbb{Q}_h(\nabla w),A\nabla u-\mathbb{Q}_h(A\nabla u))|\\
\le&|(f-Q_0 f,w-Q_0 w)|+|(A(\nabla w-\mathbb{Q}_h(\nabla w)),\nabla_w u_h-\nabla u)|\\
&+|(\nabla w-\mathbb{Q}_h(\nabla w),A\nabla u-\mathbb{Q}_h(A\nabla u))|\\
\le&Ch^{k+2}\|f\|_k\|e_0\|+C\left(h^{k+2}+h^{\frac{\beta_0-1}{2}+1}\right)\|u\|_{k+2}\|e_0\|,
\end{aligned}
$$

where we have used the following estimates in the last inequality

$$
\begin{aligned}
&|(f-Q_0 f,w-Q_0 w)|\\
&\qquad\le Ch^2\|f-Q_0 f\|\|e_0\|\le Ch^{k+2}\|f\|_k\|e_0\|,\\
&|(A(\nabla w-\mathbb{Q}_h(\nabla w)),\nabla_w u_h-\nabla u)|\\
&\qquad\le Ch(\|\nabla u-\mathbb{Q}_h(\nabla u)\|+\|\nabla_w(Q_h u-u_h)\|)\|e_0\|\\
&\qquad\le C\left(h^{k+2}+h^{\frac{\beta-1}{2}+1}\right)\|u\|_{k+2}\|e_0\|,\\
&|(\nabla w-\mathbb{Q}_h(\nabla w),A\nabla u-\mathbb{Q}_h(A\nabla u))|\\
&\qquad\le Ch\|A\nabla u-\mathbb{Q}_h(A\nabla u)\|\|e_0\|\le Ch^{k+2}\|u\|_{k+2}\|e_0\|.
\end{aligned}
$$

Consequently, it arrives at

$$|I| \leq Ch^{k+2}\|f\|_k\|e_0\| + C\left(h^{k+2} + h^{\frac{\beta_0-1}{2}+1}\right)\|u\|_{k+2}\|e_0\|. \tag{4.11}$$

Furthermore, following the proof of Theorem 4.1, we have

$$\begin{aligned}|II| &= \left|\sum_{e\in\mathcal{E}_\mathcal{I}} \langle [\![e_b]\!], \Pi_h(A\nabla w)\cdot \boldsymbol{n}_e\rangle_e\right| \leq h^{\frac{\beta_0-1}{2}}\|w\|_2 \,|\!|\!|\, e_h |\!|\!| \\ &\leq C\left(h^{k+1+\frac{\beta_0-1}{2}} + h^{\beta_0-1}\right)\|u\|_{k+2}\|e_0\|.\end{aligned}$$

From (4.10), the sum of I and II is bounded

$$\begin{aligned}\|e_0\|^2 &\leq |I| + |II| \\ &\leq Ch^{k+2}\|f\|_k\|e_0\| + C\left(h^{k+2} + h^{\frac{\beta_0-1}{2}+1} + h^{\beta_0-1}\right)\|u\|_{k+2}\|e_0\|,\end{aligned} \tag{4.12}$$

which completes the proof of (4.9). □

5. Numerical Experiments

In this section, we give some numerical results using scheme (3.1) in section 3 to verify the error estimates in Theorems 4.1 and 4.2.

In all numerical examples, we take $\Omega = (0,1)\times(0,1)$ and employ RT_k $(k = 0,1)$ elements in the weak Galerkin discretization. Multilevel uniform triangular meshes are generated by the following ways. First, we partition the square domain into $N \times N$ subsquares uniformly; then we divide each subsquare into two triangles by the diagonal line with a negative slope, completing the construction of uniformly refined triangular meshes.

Example 5.1. We consider the Poisson's equation

$$\begin{aligned} -\Delta u &= f, \qquad \text{in} \quad \Omega,\\ u &= g, \qquad \text{on} \quad \partial\Omega.\end{aligned}$$

Here let the exact solution be $u(x,y) = e^{-x-y^2}$, which admits high regularity.

The errors are listed in Tables 5.1-5.2 for RT_0 elements and in Tables 5.3-5.5 for RT_1 elements. On the one hand, it can be seen for RT_0 elements from Table 5.1 that the over-penalized WG solution converges poorly for $\beta_0 = 1$, and has low-order convergence rates about 0.48 in

Table 5.1: Errors for Example 5.1 with $(\mathbb{P}_0, \mathbb{P}_0, RT_0)$ and $\beta_0 = 1, 2$.

h	$\beta_0 = 1$		$\beta_0 = 2$	
	$\|\|\|e_h\|\|\|$	$\|e_0\|$	$\|\|\|e_h\|\|\|$	$\|e_0\|$
1/16	5.7902e-01	1.8504e-02	2.8636e-01	3.2236e-03
1/32	5.8036e-01	1.8448e-02	2.1333e-01	1.6640e-03
1/64	5.8099e-01	1.8475e-02	1.5517e-01	8.4869e-04
1/128	5.8129e-01	1.8503e-02	1.1135e-01	4.2907e-04
Rate	-7.4476e-04	-2.1848e-03	0.4787	0.9840

Table 5.2: Errors for Example 5.1 with $(\mathbb{P}_0, \mathbb{P}_0, RT_0)$ and $\beta_0 = 3, 4$.

h	$\beta_0 = 3$		$\beta_0 = 4$	
	$\vert\vert\vert e_h \vert\vert\vert$	$\Vert e_0 \Vert$	$\vert\vert\vert e_h \vert\vert\vert$	$\Vert e_0 \Vert$
1/16	9.0716e-02	4.2043e-04	2.9669e-02	2.0304e-04
1/32	4.5839e-02	1.0617e-04	1.1694e-02	4.9946e-05
1/64	2.3000e-02	2.6652e-05	4.8595e-03	1.2358e-05
1/128	1.1515e-02	6.6741e-06	2.1383e-03	3.0775e-06
Rate	0.9981	1.9976	1.1843	2.0155

Table 5.3: Errors for Example 5.1 with $(\mathbb{P}_1, \mathbb{P}_1, RT_1)$ and $\beta_0 = 2, 3$.

h	$\beta_0 = 2$		$\beta_0 = 3$	
	$\vert\vert\vert e_h \vert\vert\vert$	$\Vert e_0 \Vert$	$\vert\vert\vert e_h \vert\vert\vert$	$\Vert e_0 \Vert$
1/4	1.0169e-01	7.9897e-04	5.9035e-02	2.9159e-04
1/8	7.3594e-02	2.6721e-04	2.9946e-02	4.7429e-05
1/16	5.2662e-02	1.0785e-04	1.5077e-02	9.4341e-06
1/32	3.7461e-02	4.9846e-05	7.5645e-03	2.1713e-06
1/64	2.6568e-02	2.4347e-05	3.7886e-03	5.2843e-07
Rate	0.4956	1.0337	0.9976	2.0388

Table 5.4: Errors for Example 5.1 with $(\mathbb{P}_1, \mathbb{P}_1, RT_1)$ and $\beta_0 = 4$.

h	$\beta_0 = 4$			
	$\vert\vert\vert e_h \vert\vert\vert$	Rate.	$\Vert e_0 \Vert$	Rate.
1/4	3.4462e-02		1.4098e-04	
1/8	1.2276e-02	1.4892	1.5700e-05	3.1667
1/16	4.3545e-03	1.4953	1.9271e-06	3.0263
1/32	1.5420e-03	1.4977	2.4322e-07	2.9861
1/64	5.4559e-04	1.4989	4.8043e-08	2.3399

Table 5.5: Errors for Example 5.1 with $(\mathbb{P}_1, \mathbb{P}_1, RT_1)$ and $\beta_0 = 5$.

h	$\beta_0 = 5$			
	$\vert\vert\vert e_h \vert\vert\vert$	Rate.	$\Vert e_0 \Vert$	Rate.
1/4	2.0385e-02		1.0749e-04	
1/8	5.1426e-03	1.9869	1.3559e-05	2.9869
1/16	1.2916e-03	1.9933	1.7487e-06	2.9549
1/32	3.2365e-04	1.9967	2.2262e-07	2.9736
1/64	9.2389e-05	1.8086	9.6887e-06	- 5.4436

the triple-bar norm and 0.98 in the L^2-norm for $\beta_0 = 2$. When $\beta_0 = 3$ (see Table 5.2), the numerical solution has optimal convergent rates in the triple-bar and L^2-norms. On the other hand, for RT_1 elements, similar convergence results can be found in Tables 5.3-5.5, producing the optimal convergence as $\beta_0 = 5$. However, convergence rates seem to be influenced by the condition numbers of stiff matrix when the finest mesh size has been used in the case $\beta_0 = 5$. Thus, a preconditioning technique shall be considered to modify the ill-conditioned systems as β_0 increases.

Table 5.6: Comparison of iterations and CPU time for CG with IC preconditioner for Example 5.1.

h	Without preconditioning			With IC preconditioning		
	Cond.	Iter.	Time(s)	Cond.	Iter.	Time(s)
1/8	5.4388e+03	149	0.0100	1.1778e+03	38	0.0100
1/16	7.9701e+04	313	0.0500	4.9571e+03	72	0.0300
1/32	1.2458e+06	634	0.6400	2.0088e+04	132	0.3300
1/64	1.9816e+07	1226	5.8400	8.0612e+04	252	1.8800

Table 5.7: Comparison of iterations and CPU time for GMRES(restart=100) with IC preconditioner for Example 5.1.

h	Without preconditioning				With IC preconditioning			
	Cond.	Outer	Inner	Time(s)	Cond.	Outer	Inner	Time(s)
1/8	5.4388e+03	2	88	0.1092	1.1778e+03	1	4	0.0624
1/16	7.9701e+04	5	27	0.5616	4.9571e+03	1	75	0.3744
1/32	1.2458e+06	14	49	41.6520	2.0088e+04	2	44	4.0404
1/64	1.9816e+07	45	2	306.2800	8.0612e+04	4	56	24.8670

For RT_0 elements and refined meshes, we compare the Conjugate Gradient (CG) method, Generalized Minimum Residual method (GMRES) and Biconjugate Gradient Stabilized method (BICGSTAB) [14] with a preconditioner to solve the linear algebraic systems. Note that we choose the incomplete Cholesky (IC) factorization [14] as a preconditioner. In this section, all the tests start from a zero vector and terminate when the residual $r^{(k)}$ at the k-th iteration satisfies $\|r^{(k)}\|_2/\|r^{(0)}\|_2 \leq 10^{-6}$. The stiff matrices are given as $\beta_0 = 3$ in Algorithm 3.1. Comparisons of the three iterative methods are listed in the Tables 5.6-5.8 respectively. The conditon number (Cond.), iterations (Iter.) and CPU time (Time(s)) are listed in Table 5.7, where the terminologies "Outer" and "Inner" mean the total (100*Outer+Inner) iterations required by GMRES, respectively. We observe that the linear system can be handled well by the three iterative methods with IC preconditioning.

In addition, we first propose a symmetric positive-definite bilinear form $b(\cdot,\cdot)$

$$b(w,v) := \sum_{K\in\mathcal{T}_h}\sum_{e\in\partial K}(\langle w_0,v_0\rangle_e + \langle w_b,v_b\rangle_e) + \sum_{e\in\mathcal{E}_{\mathcal{I}}}\frac{1}{|e|^{\beta_0}}\int_e [\![w_b]\!][\![v_b]\!], \tag{5.1}$$

which can be used to construct a suitable *block-diagonal* preconditioner as β_0 increases for any $w,\, v \in V_h$. Table 5.9 states that the new preconditioner works well and reduces the orders of condition number from $O(h^{-4})$ to $O(h^{-2})$ when $k = 0, \beta_0 = 3$, and from $O(h^{-6})$ to $O(h^{-3})$ when $k = 1, \beta_0 = 5$.

Example 5.2. Consider the problem (1.1)-(1.2) with the following analytical solution

$$u(x,y) = \sin(\pi x)\cos(\pi y),$$

and the diffusion matrix $A = \begin{bmatrix} x^2+y^2+1 & xy \\ xy & x^2+y^2+1 \end{bmatrix}$, which was also investigated in [4].

The errors e_h in the triple-bar and L^2 norms are listed in the Table 5.10 with $\beta_0 = 2, 3$ for RT_0 elements, illustrating the optimal convergence rates of the OPWG method as $\beta_0 = 3$. For

Table 5.8: Comparison of iterations and CPU time for BICGSTAB with IC preconditioner for Example 5.1.

h	Without preconditioning			With IC preconditioning		
	Cond.	Iter.	Time(s)	Cond.	Iter.	Time(s)
1/8	5.4388e+03	127	0.0468	1.1778e+03	21	0.00
1/16	7.9701e+04	473	0.3120	4.9571e+03	40	0.0624
1/32	1.2458e+06	1539	4.3368	2.0088e+04	79	0.3588
1/64	1.9816e+07	4914	36.9570	8.0612e+04	156	2.4336

Table 5.9: Comparison of condition number with $k = 0, \beta_0 = 3$ and $k = 1, \beta_0 = 5$ for Example 5.1.

h	Without preconditioning		With *block-diagonal* preconditioning	
	$k = 0, \beta_0 = 3$	$k = 1, \beta_0 = 5$	$k = 0, \beta_0 = 3$	$k = 1, \beta_0 = 5$
1/4	4.5487e+02	2.1535e+05	8.9443e+02	8.2693e+03
1/8	5.4388e+03	1.2758e+07	3.6572e+03	6.6793e+04
1/16	7.9701e+04	8.0454e+08	1.4799e+04	5.2943e+05
1/32	1.2458e+06	5.1315e+10	5.9440e+04	4.1972e+06
1/64	1.9816e+07	3.2815e+12	2.3801e+05	3.3384e+07
Rate	3.9915	5.9988	2.0015	2.9917

Table 5.10: Errors for Example 5.2 with $(\mathbb{P}_0, \mathbb{P}_0, RT_0)$ and $\beta_0 = 2, 3$.

h	$\beta_0 = 2$		$\beta_0 = 3$	
	$\vert\vert\vert e_h\vert\vert\vert$	$\Vert e_0\Vert$	$\vert\vert\vert e_h\vert\vert\vert$	$\Vert e_0\Vert$
1/16	1.4183e+00	5.8015e-02	4.5651e-01	5.5156e-03
1/32	1.0614e+00	3.1045e-02	2.3150e-01	1.3987e-03
1/64	7.7559e-01	1.6126e-02	1.1633e-01	3.5167e-04
1/128	5.5832e-01	8.2305e-03	5.8281e-02	8.8137e-05
Rate	0.4742	0.9703	0.9971	1.9964

Table 5.11: Errors for Example 5.2 with $(\mathbb{P}_1, \mathbb{P}_1, RT_1)$ and $\beta_0 = 2, 3$.

h	$\beta_0 = 2$		$\beta_0 = 3$	
	$\vert\vert\vert e_h\vert\vert\vert$	$\Vert e_0\Vert$	$\vert\vert\vert e_h\vert\vert\vert$	$\Vert e_0\Vert$
1/4	4.9460e-01	7.5216e-03	2.8824e-01	2.6080e-03
1/8	3.6435e-01	3.6510e-03	1.4757e-01	6.1349e-04
1/16	2.6297e-01	1.8325e-03	7.4658e-02	1.5111e-04
1/32	1.8783e-01	9.2338e-04	3.7552e-02	3.7770e-05
1/64	1.3347e-01	4.6422e-04	1.8832e-02	9.4626e-06
Rate	0.4929	0.9921	0.9957	1.9969

RT_1 elements, the results are listed in the Tables 5.11-5.13, illustrating the optimal convergence orders as $\beta_0 = 5$. Moreover, when RT_0 elements are employed, we compare iterations and CPU time required for the CG, GMRES and BICGSTAB methods with/without the IC preconditioning in Tables 5.14-5.16, respectively.

Table 5.12: Errors for Example 5.2 with $(\mathbb{P}_1, \mathbb{P}_1, RT_1)$ and $\beta_0 = 4$.

h	$\beta_0 = 4$			
	$\lVert\lvert e_h \rvert\rVert$	Rate.	$\lVert e_0 \rVert$	Rate.
1/4	1.7175e-01		1.1953e-03	
1/8	6.1053e-02	1.4921	1.5828e-04	2.9168
1/16	2.1578e-02	1.5005	2.0334e-05	2.9605
1/32	7.6232e-03	1.5010	2.5738e-06	2.9819
1/64	2.6938e-03	1.5007	3.2419e-07	2.9889

Table 5.13: Errors for Example 5.2 with $(\mathbb{P}_1, \mathbb{P}_1, RT_1)$ and $\beta_0 = 5$.

h	$\beta_0 = 5$			
	$\lVert\lvert e_h \rvert\rVert$	Rate.	$\lVert e_0 \rVert$	Rate.
1/4	1.0807e-01		9.5709e-04	
1/8	2.7655e-02	1.9663	1.2735e-04	2.9098
1/16	6.9782e-03	1.9866	1.6310e-05	2.9649
1/32	1.7518e-03	1.9940	2.0575e-06	2.9867
1/64	4.3883e-04	1.9971	2.9945e-07	2.7805

Table 5.14: Comparison of iterations and CPU time for CG with IC preconditioner for Example 5.2.

h	Without preconditioning			With IC preconditioning		
	Cond.	Iter.	Time(s)	Cond.	Iter.	Time(s)
1/8	4.1111e+03	199	0.0200	1.5287e+03	37	0.0100
1/16	5.3413e+04	512	0.0800	6.6883e+03	76	0.0300
1/32	8.0446e+05	1173	1.1800	2.7536e+04	153	0.3500
1/64	1.2664e+07	2482	11.6100	1.1097e+05	301	2.2500

Table 5.15: Comparison of iterations and CPU time for GMRES (restart=100) with IC preconditioner for Example 5.2.

h	Without preconditioning				With IC preconditioning			
	Cond.	Outer	Inner	Time(s)	Cond.	Outer	Inner	Time(s)
1/8	4.1111e+03	3	13	0.2600	1.5287e+03	1	38	0.0800
1/16	5.3413e+04	8	51	1.5100	6.6883e+03	1	78	0.1400
1/32	8.0446e+05	28	31	63.8100	2.7536e+04	2	76	3.8700
1/64	1.2664e+07	94	22	643.2200	1.1097e+05	4	74	23.8900

Table 5.16: Comparison of iterations and CPU time for BICGSTAB with IC for Example 5.2.

h	Without preconditioning			With IC preconditioning		
	Cond.	Iter.	Time(s)	Cond.	Iter.	Time(s)
1/8	4.1111e+03	142	0.0300	1.5287e+03	23	0.0100
1/16	5.3413e+04	466	0.1400	6.6883e+03	49	0.0300
1/32	8.0446e+05	1682	3.2600	2.7536e+04	97	0.4200
1/64	1.2664e+07	5785	54.0900	1.1097e+05	196	2.7700

6. Summary and Forecasting

In this paper, we have introduced the over-penalized weak Galerkin (OPWG) finite element method and prove the optimal convergence rates in the H^1 and L^2 norms. The OPWG method offers complete independence of elementwises shape functions and results in two sets of values on interior edges, which can be handled conveniently by the penalty terms on the jumps to maintain a weak continuity. This method is given as an efficient WG approximation with the penalized jump term. Based on our analysis, the optimal choices of the penalty parameters β_0 can be derived such that the OPWG method has optimal convergence orders for the second order elliptic problems, with RT_k elements employed. Some experiments in the piecewise constant and linear approximation spaces with RT_k elements ($k = 0, 1$) have been conducted and the numerical results are in good agreement with the theoretical analysis in Section 4. Based on our analysis, the present method is flexible in selecting penalty parameters, simple shape functions, and easy-to-implement stiff matrix. Furthermore, due to the discontinuity between neighboring elements, the novel WG is promising in adaptive approximation, because different orders of polynomials can be used for any elements. Furthermore, the new method has potentials as discontinuous Galerkin methods in parallel computing [15] and in discretizations of other PDEs with interface conditions [6, 16]. Nevertheless, there are open questions to be considered, such as more degrees of freedom generated on edges than those in the conventional WG methods.

Acknowledgements. The research of the second author is supported in part by the Natural Science Foundation of Gansu Province, China (Grant 18JR3RA290), and Special Program for Applied Research on Super Computation of the NSFC-Guangdong Joint Fund (the second phase).

References

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

[2] L. Mu, J. Wang, Y. Wang and X. Ye, A computational study of the weak Galerkin method for second-order elliptic equations, *Numer. Algo.*, **63**:4 (2012), 753–777.

[3] P.A. Raviart and J.M. Thomas, A mixed finite element method for 2nd order elliptic problems, Mathematical aspects of the finite element method, Lecture Notes in Mathematics, Vol. 606, Springer, New York, 1977.

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

[5] L. Mu, J. Wang and X. Ye, Weak Galerkin finite element methods on polytopal meshes, *Int. J. Numer. Anal. Model.*, **12**:1 (2015), 31–53.

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

[7] J. Wang and X. Ye, A weak Galerkin finite element method for the stokes equations, *Adv. Comput. Math.*, **42**:1 (2016), 155–174.

[8] L. Mu, J. Wang and X. Ye, A new weak Galerkin finite element method for the Helmholtz equation, *IMA J. Numer. Anal.*, **35**:3 (2015), 1228–1255.

[9] L. Mu, J. Wang, X. Ye and S. Zhang, A weak Galerkin finite element method for the Maxwell equations, *J. Sci. Comput.*, **65**:1 (2015), 363–386.

[10] 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**:12 (2014), 2314–2330.

[11] B. Riviere, Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations: Theory and Implementation, SIAM, Philadelphia, PA, USA, 2008.

[12] X. Wang, N. Malluwawadu, F. Gao and T. McMillan, A modified weak Galerkin finite element method, *J. Comput. Appl. Math.*, **271** (2014), 319–327.

[13] J. Wang and X. Ye, A weak Galerkin mixed finite element method for second order elliptic problems, *Math. Comput.*, **83**:6 (2014), 2101–2126.

[14] Y. Saad, Iterative Methods for Sparse Linear System, Philadelphia, SIAM, second edition, 2003.

[15] A. Baggag, H. Atkins and D. Keyes, Parallel implementation of the discontinuous Galerkin method, Technical report, 1999.

[16] 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.