Journal of Computational and Applied Mathematics 273 (2015) 346–362

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

A comparative study on the weak Galerkin, discontinuous Galerkin, and mixed finite element methods

Guang Lin [a,b,*], Jiangguo Liu [c], Farrah Sadre-Marandi [c]

[a] Department of Mathematics, School of Mechanical Engineering, Purdue University, West Lafayette, IN 47907, USA

[b] Computational Sciences and Mathematics Division, Pacific Northwest National Lab, Richland, WA 99352, USA

[c] Department of Mathematics, Colorado State University, Fort Collins, CO 80523-1874, USA

ARTICLE INFO

Article history:
Received 9 December 2013
Received in revised form 5 May 2014

Keywords:
Discontinuous Galerkin
Elliptic boundary value problems
Flux
Local conservation
Mixed methods
Weak Galerkin

ABSTRACT

This paper presents a comparative study on the newly introduced weak Galerkin finite element methods (WGFEMs) with the widely accepted discontinuous Galerkin finite element methods (DGFEMs) and the classical mixed finite element methods (MFEMs) for solving second-order elliptic boundary value problems. We examine the differences, similarities, and connection among these methods in scheme formulations, implementation strategies, accuracy, and computational cost. The comparison and numerical experiments demonstrate that WGFEMs are viable alternatives to MFEMs and hold some advantages over DGFEMs, due to their properties of local conservation, normal flux continuity, no need for penalty factor, and definiteness of discrete linear systems.

Published by Elsevier B.V.

1. Introduction

For convenience of presentation, we concentrate on two-dimensional elliptic boundary value problems formulated as

$$\begin{cases} \nabla \cdot (-\mathbf{K}\nabla p) \equiv \nabla \cdot \mathbf{u} = f, & \mathbf{x} \in \Omega, \\ p = p_D, & \mathbf{x} \in \Gamma^D, \\ \mathbf{u} \cdot \mathbf{n} = u_N, & \mathbf{x} \in \Gamma^N, \end{cases} \tag{1}$$

where $\Omega \subset \mathbb{R}^2$ is a bounded polygonal domain, p the primal unknown, $\mathbf{K}$ a conductivity or permeability tensor that is uniformly symmetric positive-definite, f a source-term, p_D, u_N respectively Dirichlet and Neumann boundary data, $\mathbf{n}$ the unit outward normal vector on $\partial\Omega$, which has a non-overlapping decomposition $\Gamma^D \cup \Gamma^N$.

When $\Gamma^D \neq \emptyset$, the problem has a unique solution. When $\Gamma^D = \emptyset$, a consistency condition

$$\int_\Omega f d\mathbf{x} = \int_{\Gamma^N} u_N ds$$

is specified to ensure uniqueness of the solution.

The model problem (1) arises from many practical problems, for example, flow in porous media, heat or electrical conduction in composite materials. In the context of porous medium flow, p is the pressure for a single-phase steady flow, $\mathbf{K}$ is the ratio of permeability and fluid viscosity, and $\mathbf{u} = -\mathbf{K}\nabla p$ is the Darcy velocity. In the context of heat (electric)

* Corresponding author at: Computational Sciences and Mathematics Division, Pacific Northwest National Lab, Richland, WA 99352, USA. Tel.: +1 509 372 6596.

E-mail addresses: guang.lin@pnnl.gov, glin75@yahoo.com, guanglin@purdue.edu (G. Lin), liu@math.colostate.edu (J. Liu), sadre@math.colostate.edu (F. Sadre-Marandi).

http://dx.doi.org/10.1016/j.cam.2014.06.024
0377-0427/Published by Elsevier B.V.

conduction, p is the temperature (electric potential), $\mathbf{K}$ is the thermal (electric) conductivity, and $\mathbf{u} = -\mathbf{K}\nabla p$ is heat (electric) flux. All these applications call for accurate, efficient, and robust numerical approximations of not only the primal variable (pressure, temperature, or electrical potential) but also the flux (Darcy velocity, heat or electrical flux).

There have been a variety of numerical methods for the model problem (1): the continuous Galerkin finite element methods (CGFEMs), the DGFEMs, and the MFEMs, in addition to the finite difference methods and finite volume methods. All these numerical methods result in large-scale discrete linear systems, which are solved directly or iteratively. Besides accuracy, efficiency, and robustness of numerical methods, physical properties such as local conservation and flux continuity are also major concerns in practical applications mentioned above.

One could say there are already plenty of numerical methods for even just a simple elliptic boundary value problem like (1). *Is there any need for developing new numerical methods for an already well studied model problem? What is it good for with the new WGFEMs?*

To answer the above questions, let us briefly examine the main features of the existing numerical methods.

(i) The CGFEMs are known as lacking of "local conservation", even though they are conceptually simple and have relatively less unknowns [1,2].
(ii) The DGFEMs are locally conservative but *there is no continuity in the DG flux* [3]. DGFEMs have flexibility in handling complicated geometry but proliferate in numbers of unknowns. Choosing problem-dependent penalty factors is a drawback for practical use of DGFEMs.
(iii) The MFEMs approximate the primal variable and flux simultaneously using finite element pairs that satisfy the inf–sup condition. But the resulting indefinite linear systems (saddle-point problems) require special solvers [4,5].

The weak Galerkin finite element methods introduced in [6] adopt a completely different approach. They rely on novel concepts such as the weak gradient and discrete weak gradients. As is well known, the variational formulation of a second-order elliptic problem relies on the duality of the classical gradient operator. Locality of a finite element space implies local conservation and relative independence of elementwise shape functions. However, shape functions in element interiors are related to their values on the element interfaces (through integration by parts). The weak gradient operator characterizes exactly this connection, see Eq. (12) in Section 3. Discrete weak gradients inherit the connection for finite dimensional (Galerkin type) polynomial approximation subspaces, see Eq. (14) in Section 3.

As novel WGFEMs are being developed for different types of problems, e.g., second order elliptic problems, bi-harmonic problems, and Stokes flow, there arises a need for comparing WGFEMs with the existing numerical methods, especially the classical mixed finite element methods and the intensively investigated DGFEMs. This paper addresses such a need by providing fair comparison of these three types of methods. We compare these methods on scheme formulation, accuracy and error estimates, numbers of unknowns and condition numbers, implementation issues, and desired physical properties.

The rest of this paper is organized as follows. Section 2 presents preliminaries about RT_0, BDM_1 finite elements that are common to MFEMs, WGFEMs, and DGFEMs postprocessing, especially for flux calculations. Section 3 presents WGFEMs, DGFEMs, MFEMs and their implementations. Section 4 presents detailed comparison of WGFEMs with DGFEMs and MFEMs. Section 5 presents numerical results to further examine the differences among these methods. Section 6 concludes the paper with some remarks.

2. Preliminaries: bases for *RT*, *BDM* finite element spaces

The divergence form of the second order elliptic problem in (1) indicates the importance of the space $H(\mathrm{div}, \Omega)$ and $H(\mathrm{div})$-conforming finite element spaces. We define

$$H(\mathrm{div};\Omega) = \left\{\mathbf{v} \in L^2(\Omega)^2 : \nabla\cdot\mathbf{v} \in L^2(\Omega)\right\},$$
$$H_{0,N}(\mathrm{div};\Omega) = \left\{\mathbf{v} \in H(\mathrm{div};\Omega) : \mathbf{v}\cdot\mathbf{n} = 0 \text{ on } \Gamma^N\right\},$$
$$H_{u_N,N}(\mathrm{div};\Omega) = \left\{\mathbf{v} \in H(\mathrm{div};\Omega) : \mathbf{v}\cdot\mathbf{n} = u_N \text{ on } \Gamma^N\right\}.$$

Raviart–Thomas (RT) elements and Brezzi–Douglas–Marini (BDM) elements are among the most frequently used [7,3,5,8,9,6]. In this section, we briefly discuss some interesting properties of the basis functions for these elements.

2.1. Edge-based bases for RT_0 and BDM_1

Barycentric coordinates are enjoyed by practitioners of FEMs. Let $T = \Delta P_1P_2P_3$ be a triangle oriented counterclockwise and $|T|$ be its area. For any point $P(x, y)$ on the triangle, let $|T_i|(i = 1, 2, 3)$ be the areas of the small triangles when $P_i(x_i, y_i)$ are respectively replaced by P (see Fig. 1). Then $\lambda_i = |T_i|/|T|(i = 1, 2, 3)$ are the barycentric coordinates. Clearly $0 \le \lambda_i \le 1$ and $\lambda_1 + \lambda_2 + \lambda_3 = 1$.

It is clear that $\lambda_i(x, y)(i = 1, 2, 3)$ are also the Lagrangian P1 basis functions, whose gradients are

$$\begin{aligned}\nabla\lambda_1 &= \langle y_2 - y_3, x_3 - x_2\rangle/(2|T|),\\ \nabla\lambda_2 &= \langle y_3 - y_1, x_1 - x_3\rangle/(2|T|),\\ \nabla\lambda_3 &= \langle y_1 - y_2, x_2 - x_1\rangle/(2|T|).\end{aligned} \tag{2}$$

Their integrals are nicely expressed in the following lemma.

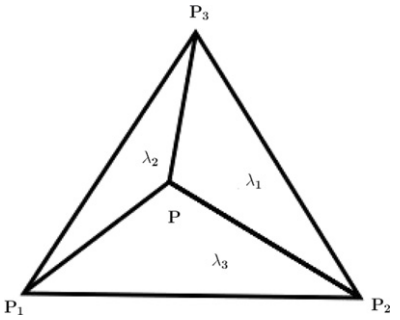

Fig. 1. Barycentric coordinates $\lambda_i (i = 1, 2, 3)$ for a generic triangle.

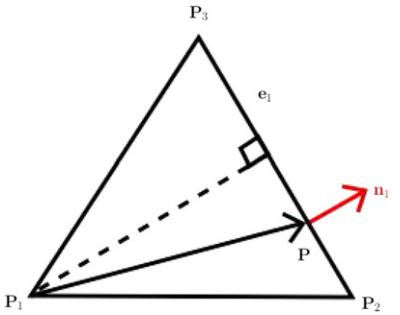

Fig. 2. RT_0 edge-based local basis functions.

Lemma. *Let α, β, γ be nonnegative integers. Then*

$$\int_T \lambda_1^\alpha \lambda_2^\beta \lambda_3^\gamma dT = \frac{2|T|\alpha!\beta!\gamma!}{(\alpha+\beta+\gamma+2)!}. \tag{3}$$

Geometric information of triangles is used to construct local and global basis functions for RT_0, RT_1, BDM_1 elements.

RT_0 **edge-based basis functions** use areas and edge lengths of triangles (see Fig. 2). Let $e_i (i = 1, 2, 3)$ be the edge opposite to vertex P_i and

$$\phi_i = \frac{|e_i|}{2|T|}(P - P_i), \quad i = 1, 2, 3. \tag{4}$$

One can easily prove that [7]

$$\phi_i \cdot \mathbf{n}_j = \delta_{i,j}, \tag{5}$$

$$\nabla \cdot \phi_i = |e_i|/|T|, \quad i = 1, 2, 3, \tag{6}$$

where $\mathbf{n}_j$ is the unit outer normal on edge e_j and δ_{ij} is the Kronecker symbol.

BDM_1 **edge-based basis functions** use internal angles, tangential vectors, and barycentric coordinates of triangles. Let θ_i be the internal angle at vertex P_i and $\mathbf{t}_i$ be the unit tangential vector on edge e_i following the counterclockwise orientation. Clearly, $\mathbf{n}_i$ can be obtained by a 90° clockwise rotation of $\mathbf{t}_i$. We have six edge-based basis local functions for BDM_1 (see Fig. 3):

$$\begin{aligned} \Phi_{1,s} &= \frac{\lambda_2 \mathbf{t}_3}{\sin(\theta_2)}, & \Phi_{1,e} &= -\frac{\lambda_3 \mathbf{t}_2}{\sin(\theta_3)}, \\ \Phi_{2,s} &= \frac{\lambda_3 \mathbf{t}_1}{\sin(\theta_3)}, & \Phi_{2,e} &= -\frac{\lambda_1 \mathbf{t}_3}{\sin(\theta_1)}, \\ \Phi_{3,s} &= \frac{\lambda_1 \mathbf{t}_2}{\sin(\theta_1)}, & \Phi_{3,e} &= -\frac{\lambda_2 \mathbf{t}_1}{\sin(\theta_2)}. \end{aligned} \tag{7}$$

It is clear that

(i) $\Phi_{i,s} \cdot \mathbf{n}_j = 0$ and $\Phi_{i,e} \cdot \mathbf{n}_j = 0$ for $i \neq j$.
(ii) $\Phi_{i,s} \cdot \mathbf{n}_i$ and $\Phi_{i,e} \cdot \mathbf{n}_i$ are linear on edge e_i.
(iii) $\nabla \cdot \Phi_{i,s}$, $\nabla \cdot \Phi_{i,e}$ can be calculated using (2) and (7).

Items (i) and (ii) are due to either orthogonality of the normal and tangential vectors or vanishing of a barycentric coordinate on the opposite edge.

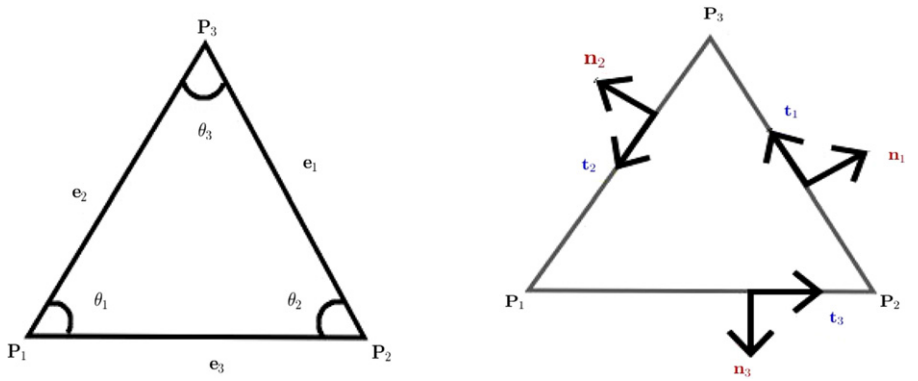

Fig. 3. Triangle geometric information used for BDM_1 edge-based local basis functions.

Theorem 0. *There exists hierarchy among RT_0, RT_1, BDM_1 spaces as shown below* [5]

$$RT_0 \subset BDM_1 \subset RT_1 \subset \cdots.$$

This is further reflected in the edge-based local basis functions. In particular, the following holds

$$\phi_i^{RT_0} = \frac{1}{2}\left(\Phi_{i,s}^{BDM_1} + \Phi_{i,e}^{BDM_1}\right), \quad i = 1, 2, 3. \tag{8}$$

Proof. We consider edge e_1. It is clear that

$$\frac{|e_1|}{\sin(\theta_1)} = \frac{|e_2|}{\sin(\theta_2)} = \frac{|e_3|}{\sin(\theta_3)}.$$

So we have $|e_2|\sin(\theta_3) = |e_3|\sin(\theta_2) = c$, where for convenience c is used to name the same constant. Note that $|T| = |e_3||e_1|\sin(\theta_2) = |e_1||e_2|\sin(\theta_3)$. Taking coefficients $\frac{1}{2} = c_s = c_e = \frac{|e_1|}{2|T|}c$ and using the facts that $|e_3|\mathbf{t}_3 = P_2 - P_1$, $-|e_2|\mathbf{t}_2 = P_3 - P_1$, we have

$$\begin{aligned} c_s\Phi_{1,s}^{BDM_1} + c_e\Phi_{1,e}^{BDM_1} &= \frac{|e_1|}{2|T|}(\lambda_2 P_2 + \lambda_3 P_3 - \lambda_2 P_1 - \lambda_3 P_1) \\ &= \frac{|e_1|}{2|T|}(\lambda_2 P_2 + \lambda_3 P_3 + \lambda_1 P_1 - P_1) = \frac{|e_1|}{2|T|}(P - P_1) = \phi_1^{RT_0}, \end{aligned}$$

where the facts $\lambda_1 + \lambda_2 + \lambda_3 = 1$ and $\lambda_1 P_1 + \lambda_2 P_2 + \lambda_3 P_3 = P$ have been used.

2.2. Normalized bases for RT_0, BDM_1, RT_1 finite element spaces

Let (x_c, y_c) be the element center and $X = x - x_c$, $Y = y - y_c$. The normalized local basis for triangular RT_0 elements refers to [9,10]

$$\begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}X\\Y\end{bmatrix}. \tag{9}$$

The normalized local basis functions for triangular BDM_1 elements are

$$\begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}X\\0\end{bmatrix}, \begin{bmatrix}Y\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}0\\X\end{bmatrix}, \begin{bmatrix}0\\Y\end{bmatrix}. \tag{10}$$

Similarly, the normalized local basis for triangular RT_1 elements is

$$\begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}X\\0\end{bmatrix}, \begin{bmatrix}Y\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}0\\X\end{bmatrix}, \begin{bmatrix}0\\Y\end{bmatrix}, \begin{bmatrix}X^2\\XY\end{bmatrix}, \begin{bmatrix}XY\\Y^2\end{bmatrix}. \tag{11}$$

The hierarchy among these basis functions are trivial. It has been observed in [9] that the edge-based functions are convenient for implementing MFEMs, whereas the normalized basis functions are convenient for implementing WGFEMs.

3. WG, DG, MFEM and their implementations

Let $\mathcal{E}_h$ be a conforming mesh on $\Omega \subset \mathbb{R}^2$. We use Γ_h to denote the set of all edges, Γ_h^I the set of all interior edges, Γ_h^D the set of edges on the Dirichlet boundary, Γ_h^N the set of edges on the Neumann boundary. Accordingly we have $\Gamma_h = \Gamma_h^I \cup \Gamma_h^D \cup \Gamma_h^N$.

3.1. WGFEMs and implementation

The weak Galerkin finite element methods were first developed in [6] based on discrete weak gradients, which approximate the weak gradient operator.

3.1.1. Discrete weak gradients and weak finite element spaces

Let E be a triangle or rectangle with interior E° and boundary $E^\partial := \partial E$.

A **weak function** on E refers to a pair of scalar-valued functions $v = \{v^\circ, v^\partial\}$ such that $v^\circ \in L^2(E^\circ)$ and $v^\partial \in H^{\frac{1}{2}}(\partial E)$. Here v° can be understood as the value of v in the interior of E, whereas v^∂ is the value of v on the boundary of E. Note that v^∂ might not be the trace of v°, should a trace be defined.

For any weak function v, its **weak gradient** $\nabla_w v$ is defined (interpreted) as a linear functional on $H(\mathrm{div}, E)$:

$$(\nabla_w v, \mathbf{w}) = \int_{\partial E} v^\partial (\mathbf{w}\cdot\mathbf{n}) ds - \int_E v^\circ (\nabla\cdot\mathbf{w}) dE, \quad \forall \mathbf{w} \in H(\mathrm{div}, E). \tag{12}$$

Let $P^l(E^\circ)$ be the space of polynomials on E° with degree at most $l \geq 0$ and $P^m(\partial E)$ be the space of polynomials on ∂E with degree at most $m \geq 0$. A **discrete weak function** is a weak function $v = \{v^\circ, v^\partial\}$ such that $v^\circ \in P^l(E^\circ)$ and $v^\partial \in P^m(\partial E)$. A discrete weak function space on E is defined as

$$W(E, l, m) = \{v = \{v^\circ, v^\partial\} : v^\circ \in P^l(E^\circ), v^\partial \in P^m(\partial E)\}. \tag{13}$$

Let n be any nonnegative integer and $P^n(E)^2$ be the space of vector-valued polynomials with degree at most n. Let $V(E, n)$ be a subspace of $P^n(E)^2$. For any $v \in W(E, l, m)$, its **discrete weak gradient** is so defined that $\nabla_{w,d} v \in V(E, n)$ is the unique solution of

$$\int_E \nabla_{w,d} v \cdot \mathbf{w} dE = \int_{\partial E} v^\partial (\mathbf{w}\cdot\mathbf{n}) ds - \int_E v^\circ (\nabla\cdot\mathbf{w}) dE, \quad \forall \mathbf{w} \in V(E, n). \tag{14}$$

For later use, we define a projection operator $Q_h = (Q_h^\circ, Q_h^\partial)$ into a discrete weak function space, where Q_h° is the L^2-projection into the function space on E° and Q_h^∂ is the L^2-projection into the function space on E^∂.

Patching all elementwise discrete weak function spaces over the mesh $\mathcal{E}_h$, we obtain **weak finite element spaces**

$$S_h(l, m) = \{v = \{v^\circ, v^\partial\} : v|_E \in W(E, l, m)\ \forall E \in \mathcal{E}_h\}, \tag{15}$$

$$S_h^0(l, m) = \{v = \{v^\circ, v^\partial\} \in S_h(l, m), v^\partial|_{\partial E \cap \Gamma^D} = 0\ \forall E \in \mathcal{E}_h\}. \tag{16}$$

For the WGFEMs to work well, two properties should be satisfied [6]:

- **Property P1**. For any $v \in S_h(l, m)$ and any $E \in \mathcal{E}_h$, if $\nabla_{w,d} v = 0$ on E, then there must be $v^\circ = \text{const}$ on E° and $v^\partial = \text{const}$ on ∂E.
- **Property P2**. For $u \in H^s(\Omega), s \geq 1$, let $Q_h u \in S_h(l, m)$ be an interpolation or projection of u. Then the discrete weak gradient of $Q_h u$ should be a good approximation of ∇u.

Here is a list of weak finite element spaces that satisfy the above properties [9,11,6].

- The lowest order WG (P_0, P_0, RT_0) for triangular meshes consists of weak shape functions that are constants on elements in a triangular mesh and constants on edges in the mesh. Their discrete weak gradients are in the Raviart–Thomas space $RT_0(E)$ for each triangle E in the mesh.
- WG $(P_0, P_0, RT_{[0]})$ consists of weak shape functions that are constants on elements in a rectangular mesh and constants on edges in the mesh. Their discrete weak gradients are in the Raviart–Thomas space $RT_{[0]}(E)$ for each rectangle E in the mesh.
- WG (P_0, P_1, BDM_1) for triangular meshes.
- Higher order WG (P_l, P_l, RT_l) on triangular meshes and WG $(P_l, P_l, RT_{[l]})$ on rectangular meshes for any integer $l \geq 1$.
- WG (P_0, P_0, RT_0) for tetrahedral meshes in 3-dim as used in [12]. Such an element has constant basis functions on the interior of a tetrahedron and each of its four triangular faces but has discrete weak gradients in the 3-dim RT_0 space. Similar higher order WG elements can be established for tetrahedral and 3-dim rectangular meshes.

3.1.2. Weak Galerkin finite element scheme for elliptic BVPs

We define bilinear and linear forms on the WG finite element spaces as

$$\mathcal{A}_h(p_h, q) := \sum_{E\in\mathcal{E}_h} \int_E \mathbf{K}\nabla_{w,d} p_h \cdot \nabla_{w,d} q dE, \tag{17}$$

$$\mathcal{F}(q) := \sum_{E\in\mathcal{E}_h} \int f q^\circ dE - \sum_{\gamma\in\Gamma_h^N} \int_\gamma u_N q ds. \tag{18}$$

Weak Galerkin Finite Element Scheme reads as: Seek $p_h = \{p_h^\circ, p_h^\partial\} \in S_h(l, m)$ such that $p_h^\partial|_{\Gamma^D} = Q_h^\partial p_D$ and

$$\mathcal{A}_h(p_h, q) = \mathcal{F}(q), \quad \forall q = \{q^\circ, q^\partial\} \in S_h^0(l, m). \tag{19}$$

After p_h is obtained from solving (19), one computes the WG flux as follows

$$\mathbf{u}_h = R_h(-\mathbf{K}\nabla_{w,d} p_h), \tag{20}$$

where R_h is the local L^2 projection onto $V(E, n)$.

We define norms of the error as follows

$$\|p - p_h^\circ\|^2_{L^2(\Omega)} := \sum_{E\in\mathcal{E}_h} \int_E |p - p_h^\circ|^2 dE, \tag{21}$$

$$\|Q_h^\circ p - p_h^\circ\|^2_{L^2(\Omega)} := \sum_{E\in\mathcal{E}_h} \int_E |Q_h^\circ p - p_h^\circ|^2 dE, \tag{22}$$

$$\|\mathbf{u} - \mathbf{u}_h\|^2_{L^2(\Omega)} = \sum_{E\in\mathcal{E}_h} \int_E |\mathbf{u} - \mathbf{u}_h|^2 dE. \tag{23}$$

Theorem 1. *Let $(p, \mathbf{u})$ be the exact solutions to (1). If the solutions are sufficiently smooth, i.e., there is an integer $m \geq 2$ such that $p \in H^m(\Omega)$ and $\mathbf{u} \in H^{m-1}(\Omega)^2$, then*

$$\|p - p_h^\circ\|_{L^2(\Omega)} = \mathcal{O}(h^{r_0}), \qquad \|Q_h^\circ p - p_h^\circ\|_{L^2(\Omega)} = \mathcal{O}(h^{r_0+1}), \tag{24}$$

$$\|\mathbf{u} - \mathbf{u}_h\|_{L^2(\Omega)} = \mathcal{O}(h^{r_1}), \tag{25}$$

where $r_0 = \min(m, k_0), r_1 = \min(m - 1, k_1)$ and k_0, k_1 are integers related to the WG finite elements being used. For (P_0, P_0, RT_0) and $(P_0, P_0, RT_{[0]})$, we have $k_0 = k_1 = 1$; for (P_1, P_1, RT_1) and $(Q_1, P_1, RT_{[1]})$, we have $k_0 = k_1 = 2$; for (P_0, P_1, BDM_1) and $(Q_0, P_1, BDM_{[1]})$, we have $k_0 = 1, k_1 = 2$. If the solutions have low regularity, that is, $p \in H^{1+s}(\Omega), \mathbf{u} \in H^s(\Omega)^2$ for some $s \in (\frac{1}{2}, 1)$, then

$$\|p - p_h^\circ\|_{L^2(\Omega)} = \mathcal{O}(h^{r_0}), \qquad \|Q_h^\circ p - p_h^\circ\|_{L^2(\Omega)} = \mathcal{O}(h^{2r_0}), \tag{26}$$

$$\|\mathbf{u} - \mathbf{u}_h\|_{L^2(\Omega)} = \mathcal{O}(h^{r_1}), \tag{27}$$

where $r_0 = \min(s, k_0), r_1 = \min(s, k_1)$ but k_0, k_1 are the same as above.

Theorem 2 (*Local Conservation*)**.** *Let E be any element in the mesh $\mathcal{E}_h$. Then*

$$\int_{\partial E} \mathbf{u}_h \cdot \mathbf{n} ds = \int_E f dE. \tag{28}$$

Theorem 3 (*Continuity of Normal Flux*)**.** *Let $\gamma \in \Gamma_h^I$ be an interior edge shared by two elements E_1, E_2, $\mathbf{n}_i$ be the outer normal of element $E_i (i = 1, 2)$ on γ, and $\mathbf{u}_h^{(i)}$ be the numerical flux on element $E_i (i = 1, 2)$. Then*

$$\int_\gamma \mathbf{u}_h^{(1)} \cdot \mathbf{n}_1 ds = \int_\gamma \mathbf{u}_h^{(2)} \cdot \mathbf{n}_2 ds. \tag{29}$$

For Theorem 1, when the solutions are sufficiently smooth, the error estimates can be derived from the analysis in [6]. When the solutions have low regularity, the error estimates can be derived using the techniques presented in [13]. The proof of Theorem 3 involves the duality of the basis functions on edges. This is one of the places where we see admissibility of WG finite elements.

3.1.3. Implementation of WGFEMs

Here is a typical procedure for implementing WGFEMs.

(1) Choosing an appropriate type of WG finite elements;
(2) Computing the discrete weak gradients of the chosen WG basis functions;
(3) Computing local stiffness matrices;
(4) Assemble the elementwise stiffness matrices into a global stiffness matrix to obtain a symmetric positive-definite discrete linear system;
(5) Solving the linear system to obtain a numerical approximation to the primal variable and then use (20) to compute the WG flux.

For more details, see [9,11]. Note that when $\mathbf{K}$ is a constant scalar on each element, the local L^2 projection R_h in (20) is not needed.

Notice that for higher order WG elements, one has to solve small-size dense linear systems to obtain the discrete weak gradients of the higher order WG basis functions. For example, when WG (P_2, P_2, RT_2) elements are used, there will be 12 WG basis functions for each triangle, and their discrete weak gradients are in RT_2, which in turn has 15 basis functions. Then twelve order 15 linear systems need to be solved.

To address the above issue, the Schur complement technique can be used to reduce the computational cost of solving the global sparse discrete linear system. Theoretically, hybridization can be applied when formulating the WGFEM schemes so that interior unknowns are cast onto the interface unknowns and the degrees of freedom is reduced for higher order WG elements [14].

3.2. MFEMs and implementation

The classical mixed finite element methods (MFEMs) have been used for many problems including the prototype (1) for second-order elliptic boundary value problems. In MFEMs, the primal variable and its flux are approximated simultaneously. This is based on rewriting (1) as a system of first-order partial differential equations as follows [5]

$$\begin{cases} \mathbf{K}^{-1}\mathbf{u} + \nabla p = 0, \quad \mathbf{x} \in \Omega, \\ \nabla \cdot \mathbf{u} = f, \quad \mathbf{x} \in \Omega, \\ p = p_D, \quad \mathbf{x} \in \Gamma^D, \\ \mathbf{u} \cdot \mathbf{n} = u_N, \quad \mathbf{x} \in \Gamma^N. \end{cases} \tag{30}$$

The mixed weak formulation for (1) is: Seek $\mathbf{u} \in H_{u_N,N}(\mathrm{div}, \Omega)$ and $p \in L^2(\Omega)$ such that the following hold

$$\begin{cases} \displaystyle\int_\Omega (\mathbf{K}^{-1}\mathbf{u}) \cdot \mathbf{v} - \int_\Omega p(\nabla \cdot \mathbf{v}) = -\int_{\Gamma_D} p_D \mathbf{v} \cdot \mathbf{n}, & \forall \mathbf{v} \in H_{0,N}(\mathrm{div}, \Omega), \\ \displaystyle -\int_\Omega (\nabla \cdot \mathbf{u}) q = -\int_\Omega fq, & \forall q \in L^2(\Omega). \end{cases} \tag{31}$$

MFEMs are based on $H(\mathrm{div})$-conforming finite elements and the inf–sup condition. Denote $V = H(\mathrm{div}; \Omega)$, $W = L^2(\Omega)$. Let $V_h \subset V$ and $W_h \subset W$ be a pair of finite element spaces for the flux and primal variable respectively. The inf–sup condition requires that [5]

$$\inf_{q \in W_h} \sup_{v \in V_h} \frac{B(v, q)}{\|v\|_V \|q\|_W} > 0. \tag{32}$$

Let $U_h = V_h \cap H_{u_N,N}(\mathrm{div}; \Omega)$ and $V_h^0 = V_h \cap H_{0,N}(\mathrm{div}; \Omega)$. A mixed finite element scheme can be formulated as: Seek $\mathbf{u}_h \in U_h$ and $p_h \in W_h$ such that

$$\begin{cases} \displaystyle\sum_{E \in \mathcal{E}_h} \int_E (\mathbf{K}^{-1}\mathbf{u}_h) \cdot \mathbf{v} dE - \sum_{E \in \mathcal{E}_h} \int_E p_h \nabla \cdot \mathbf{v} dE = -\sum_{\gamma \in \Gamma_h^D} \int_\gamma p_D \mathbf{v} \cdot \mathbf{n} ds, & \forall \mathbf{v} \in V_h^0, \\ \displaystyle -\sum_{E \in \mathcal{E}_h} \int_E (\nabla \cdot \mathbf{u}_h) q dE = -\sum_{E \in \mathcal{E}_h} \int_E fq dE, & \forall q \in W_h. \end{cases} \tag{33}$$

Some error estimates for MFEMs are given in [5] when p has integer-order regularity. However, one can derive error estimates for $p \in H^{1+s}(\Omega)(s > \frac{1}{2})$ as follows

$$\|p - p_h\|_{L^2} = \mathcal{O}(h^s), \tag{34}$$

$$\|\mathbf{u} - \mathbf{u}_h\|_{L^2} = \mathcal{O}(h^s). \tag{35}$$

It is well known that MFEMs result in indefinite discrete linear systems that require special solvers, e.g., the Uzawa algorithm [4,5].

There are many element pairs that satisfy the inf–sup condition and hence could be used in the above mixed finite element scheme. The commonly used are the lowest-order Raviart–Thomas RT_0 elements for triangular meshes and $RT_{[0]}$ elements for rectangular meshes. In each case, the pressure is approximated by elementwise constants, whereas the flux is approximated from the RT_0 or $RT_{[0]}$ space.

MFEM RT_0 implementation on triangular meshes exemplifies the strategies needed for efficient implementation. As discussed in Section 2, one could use formula (4) to construct three edge-based local basis functions ϕ_i on each triangular element T. There are signs $\sigma_i = \pm 1$ $(i = 1, 2, 3)$ for these edges. These local basis functions glued together yield the global edge-based basis functions for $RT_0(\mathcal{E}_h)$.

For ease of implementation, we assume the permeability $\mathbf{K}$ is a constant 2×2 matrix on each element. Thus so is the inverse matrix $\mathbf{K}^{-1}$. Corresponding to the first term in the first equation of (33) is a local stiffness matrix M on each triangular element:

$$M = \left[\int_T (\mathbf{K}^{-1}\phi_j) \cdot \phi_i dT \right]_{1 \le i,j \le 3}.$$

Then we have

$$M = \frac{1}{48|T|} C'B'(A \otimes \mathbf{K}^{-1})BC, \tag{36}$$

where

$$A = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}, \qquad B = \begin{bmatrix} P_1 - P_1 & P_1 - P_2 & P_1 - P_3 \\ P_2 - P_1 & P_2 - P_2 & P_2 - P_3 \\ P_3 - P_1 & P_3 - P_2 & P_3 - P_3 \end{bmatrix},$$

$$C = \operatorname{diag}\left(\sigma_1|e_1|, \sigma_2|e_2|, \sigma_3|e_3|\right),$$

P_i $(i = 1, 2, 3)$ are the three vertices oriented counterclockwise, $|T|$ the triangle area, $|e_i|$ the edge length, σ_i the aforementioned sign. Clearly, the 3-by-3 matrix A is related to the Gram matrix of $\lambda_i (i = 1, 2, 3)$ for the triangle T. Formula (36) includes the special case $\mathbf{K} = \mathbf{I}_2$ discussed in [7].

MFEM BDM_1 **implementation** on triangular meshes is similar to MFEM RT_0 implementation. But attention needs to be paid to the assembly process. For an interior edge shared by two elements, we assume its normal vector points from the 1st element to the 2nd element. Note that there are two normal flux unknowns on this edge. If the counterclockwise orientation is adopted on each element, then for the two rows and two columns in the stiffness matrix of the 2nd element that correspond to the two flux unknowns on this edge, we swap their positions and change their signs $(1 \to -1, -1 \to 1)$, while assembling them into the global stiffness matrix.

3.3. DGFEMs, projected flux, and implementation

The discontinuous Galerkin finite element methods have been intensively investigated recently. In DGFEMs, discontinuous shape functions are adopted on different elements, so interior penalty is usually introduced to enforce weak continuity across the inter-element boundaries.

Let $k \geq 1$ be an integer and $E \in \mathcal{E}_h$. By convention, $P^k(E)$ consists of polynomials on E with degree at most k. We define

$$V_h^k = \{v : v|_E \in P^k(E)\ \forall E \in \mathcal{E}_h\}$$

and use it as a DG finite element space. Averages and jumps of scalar and vector quantities are defined in the usual way, see [3,2].

3.3.1. DG schemes for the primal variable

For the boundary value problem (1), a DG scheme is to seek $p_h \in V_h^k$ such that [2]

$$\mathcal{A}_h(p_h, q) = \mathcal{F}(q), \quad \forall q \in V_h^k, \tag{37}$$

where

$$\begin{aligned} \mathcal{A}_h(p_h, q) \ = \ & \sum_{E \in \mathcal{E}_h} \int_E \mathbf{K}\nabla p_h \nabla q dE - \sum_{\gamma \in \Gamma_h^I \cup \Gamma_h^D} \int_\gamma \{\mathbf{K}\nabla p_h \cdot \mathbf{n}\}[q] ds + \beta \sum_{\gamma \in \Gamma_h^I \cup \Gamma_h^D} \int_\gamma \{\mathbf{K}\nabla q \cdot \mathbf{n}\}[p_h] ds \\ & + \sum_{\gamma \in \Gamma_h^I} \frac{\alpha_\gamma}{h_\gamma} \int_\gamma [p_h][q] ds + \sum_{\gamma \in \Gamma_h^D} \frac{\alpha_\gamma}{h_\gamma} \int_\gamma p_h q ds, \end{aligned} \tag{38}$$

$$\mathcal{F}(q) = \sum_{E \in \mathcal{E}_h} \int_E f q dE - \sum_{\gamma \in \Gamma_h^N} \int_\gamma u_N q ds + \beta \sum_{\gamma \in \Gamma_h^D} \int_\gamma \mathbf{K}\nabla q \cdot \mathbf{n} p_D ds + \sum_{\gamma \in \Gamma_h^D} \frac{\alpha_\gamma}{h_\gamma} \int_\gamma p_D q ds. \tag{39}$$

Here $\alpha_\gamma > 0$ is a penalty factor on an edge $\gamma \in \Gamma_h$ and β is a formulation parameter [2]. Depending on the choice of β, one ends up with the symmetric interior penalty Galerkin (SIPG) for $\beta = -1$, the nonsymmetric interior penalty Galerkin (NIPG) for $\beta = 1$, and the incomplete interior penalty Galerkin (IIPG) for $\beta = 0$.

It is well known that the NIPG scheme is stable for any positive penalty factors. For the SIPG scheme to be stable, the penalty factors need to be large enough. This is one drawback of the DGFEMs, since choosing penalty factors are usually problem-dependent [15]. Note also that SIPG has optimal order convergence in the L^2-norm, whereas NIPG has only suboptimal order convergence (in the L^2-norm) when the approximating polynomials are even order [16].

To avoid the inconvenience of choosing penalty factors in the DGFEMs, the weak over-penalization technique is investigated in [17] and a series of follow-up work. In particular, the weakly over-penalized symmetric interior penalty (WOPSIP) formulation maintains the symmetric positive-definiteness in the discrete linear system. This results in large condition numbers, but some nice and simple preconditioners can be constructed [17].

3.3.2. Projected flux for DGFEMs

The **elementwise DG numerical flux** is obtained via

$$\mathbf{U}_h^{DG} = -\mathbf{K}\nabla p_h^{DG}, \quad \forall E \in \mathcal{E}_h. \tag{40}$$

It is locally conservative on each element. This can be proved by simply setting $q = 1$ on an element E and zero on all other elements in (38) and (39). However, *the numerical flux defined in (40) does not have a continuous normal component on element interfaces*, as pointed out in [3]. This leads to nonphysical oscillations, if the DG flux is coupled to a transport solver in a straightforward manner. This could make particle tracking difficult or impossible, if the DG velocity is used in a characteristic-based transport solver.

In view of the features and usefulness of the $H(\mathrm{div})$ finite elements, it is natural to establish a postprocessing procedure to project a DG flux into any $H(\mathrm{div})$ finite element space, e.g., a RT or BDM type space.

Local RT_{k-1}-projection of the DG flux. Let $E \in \mathcal{E}_h$ be a triangular element with three edges $\gamma_i (i = 1, 2, 3)$. For an integer $k \geq 1$, seek $\mathbf{U}_h^* \in RT_{k-1}(E)$ such that

$$\int_{\gamma_i} (\mathbf{U}_h^* \cdot \mathbf{n}_{\gamma_i})\phi = \int_{\gamma_i} (\{\mathbf{U}_h^{DG}\} \cdot \mathbf{n}_{\gamma_i})\phi, \quad \forall \phi \in P_{k-1}(\gamma_i),\ i = 1, 2, 3, \tag{41}$$

$$\int_E \mathbf{U}_h^* \cdot \Phi = \int_E \mathbf{U}_h^{DG} \cdot \Phi, \quad \forall \Phi \in P_{k-2}(E)^2 \ (\text{if } k \geq 2). \tag{42}$$

Local BDM_{k-1}-projection of the DG flux [3]. Let $E \in \mathcal{E}_h$ be a triangular element with three edges $\gamma_i (i = 1, 2, 3)$. For an integer $k \geq 2$, define

$$M_k(E) = \{\phi \in P_k(E) : \phi|_{\partial E} = 0\},$$
$$\nabla^\perp \phi = (\partial_y \phi, -\partial_x \phi).$$

Seek $\mathbf{U}_h^* \in BDM_{k-1}(E)$ such that

$$\int_{\gamma_i} (\mathbf{U}_h^* \cdot \mathbf{n}_{\gamma_i})\phi = \int_{\gamma_i} (\{\mathbf{U}_h^{DG}\} \cdot \mathbf{n}_{\gamma_i})\phi, \quad \forall \phi \in P_{k-1}(\gamma_i),\ i = 1, 2, 3, \tag{43}$$

$$\int_E \mathbf{U}_h^* \cdot \nabla \phi = \int_E \mathbf{U}_h^{DG} \cdot \nabla \phi, \quad \forall \phi \in P_{k-2}(E), \tag{44}$$

$$\int_E \mathbf{U}_h^* \cdot \nabla^\perp \phi = \int_E \mathbf{U}_h^{DG} \cdot \nabla^\perp \phi, \quad \forall \phi \in M_k(E). \tag{45}$$

For post-processing of the DG flux, the BDM finite element spaces are preferred, "because they nicely match in dimension with the space of velocities generated by the DG method" [3]. It was shown in [3] that the new numerical flux $\mathbf{U}_h^*$ obtained from solving (43)–(45) has the following properties:

(i) It is locally (elementwise) conservative

$$\int_{\partial E} \mathbf{U}_h^* \cdot \mathbf{n}_E = \int_E f, \quad \forall E \in \mathcal{E}_h.$$

(ii) It has a continuous normal component on element interfaces.
(iii) The new numerical flux identically reproduces the averaged normal component of the DG flux.
(iv) It has the same accuracy and convergence order as the original DG flux. See Theorem 4.

Theorem 4. *Assume the exact solution to (1) has regularity $p \in H^{1+s}(\Omega), s > \frac{1}{2}$. Regarding the elementwise and edgewise accuracy of the post-processed DG flux obtained from solving (43)–(45), the following hold* [3]

$$\|\mathbf{U}_h^* - \mathbf{U}_h^{DG}\|_{L^2(E)} = \mathcal{O}(h^r) \quad \forall E \in \mathcal{E}_h, \tag{46}$$

$$\|\mathbf{U}_h^* \cdot \mathbf{n} - \mathbf{u} \cdot \mathbf{n}\|_{L^2(\gamma)} = \mathcal{O}(h^{r-\frac{1}{2}}) \quad \forall \gamma \subset \partial E\ \forall E \in \mathcal{E}_h, \tag{47}$$

where $r = \min(k, s)$ and k is the degree of polynomials used for approximating the primal variable.

3.3.3. DG implementation: the primal variable and flux

DG implementation is easy in some aspects but needs careful treatments in some other aspects. Some C-type pseudo-code is provided in [16]. We have developed `Matlab` code modules for DGP1, DGP2 for testing the numerical examples in this paper.

The formulation parameter β allows one to easily switch between SIPG, NIPG, IIPG and compare their performances. Regarding the penalty factor α, even though it can vary with edges theoretically, the common practice is to choose just a uniform penalty factor for all edges. For SIPG, the dependency of scheme stability, accuracy, and condition numbers on penalty factor is obvious. A good penalty factor should ensure the DG scheme working, but is not too large and hence results in small condition numbers. This indeed depends on the physical problems being solved.

Well-organized mesh information facilitates DG implementation. Here are some strategies (tips).

(i) Information on edges versus their neighboring elements is needed.
(ii) Set the second element to zero for boundary edges.

(iii) Let the normal vector on each edge point to the 2nd element from the 1st element. This way, the normal vector on a boundary edge will automatically agree with the outward normal vector on the domain boundary.
(iv) Treat interior edges separately from the boundary edges. The former involve both neighboring elements, whereas the latter need only the 1st elements.

For `Matlab` implementation of DG, adopt the "vectorization" technique, that is, avoid assembly by a loop through each element, instead try to compute the element stiffness matrices simultaneously for the entire mesh. For `C/C++` implementation, element–element adjacency information needs to be generated for memory allocation of the sparse global coefficient matrix. Actually, one will be dealing with a sparse block matrix. Each block is a small size full matrix. The block size equals to the degrees of freedom inside each element. For a conforming triangular mesh, there are at most four blocks in each "row/column", since one triangular element interacts with itself and three neighboring elements.

Implementation of DGP1, DGP2. A delicate piece in DG implementation is the handling of averages and jumps across interior edges, as reflected in Formula (38) Terms 2, 3 and 4. Actually, Formula (38) Term 2 (ignoring the negative sign in front of it) and Term 3 could be implemented simultaneously.

Let $\gamma \in \Gamma_h^I$ be an interior edge shared by two elements E_1 and E_2. By the convention for mesh data structure, the unit normal vector $\mathbf{n}$ points (from E_1) to E_2, and hence

$$[q] = q|_{E_1} - q|_{E_2} =: q^{(1)} - q^{(2)},$$
$$[p_h] = p_h|_{E_1} - p_h|_{E_2} =: p_h^{(1)} - p_h^{(2)},$$

whereas

$$\{\mathbf{K}\nabla p_h \cdot \mathbf{n}\} = \frac{1}{2}\mathbf{K}|_{E_1}\nabla p_h^{(1)} \cdot \mathbf{n} + \frac{1}{2}\mathbf{K}|_{E_2}\nabla p_h^{(2)} \cdot \mathbf{n},$$
$$\{\mathbf{K}\nabla q \cdot \mathbf{n}\} = \frac{1}{2}\mathbf{K}|_{E_1}\nabla q^{(1)} \cdot \mathbf{n} + \frac{1}{2}\mathbf{K}|_{E_2}\nabla q^{(2)} \cdot \mathbf{n}.$$

So Terms 2 and 3 each results in four small-size matrices that reflect the interaction of trial and test basis functions on element E_1 and E_2 as follows

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \qquad \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}.$$

For example,

$$A_{12} = -\frac{1}{2}\left[\int_\gamma q^{(1)}\mathbf{K}_{E_2}\nabla p_h^{(2)} \cdot \mathbf{n} ds\right], \qquad B_{11} = \frac{1}{2}\left[\int_\gamma q^{(1)}\mathbf{K}_{E_1}\nabla p_h^{(1)} \cdot \mathbf{n} ds\right]$$

are $m \times m$ matrices ($m = 3$ for DGP1 and $m = 6$ for DGP2). Noticing the transpose relationship among these matrices will ensure code correctness and efficiency:

$$B_{11} = A'_{11}, \qquad B_{12} = A'_{21}, \qquad B_{21} = A'_{12}, \qquad B_{22} = A'_{22}.$$

DGP1 postprocessing for RT_0 flux. Assume discontinuous linear polynomials are used for approximating the primal variable on a triangular mesh. Here we describe a practical and simple postprocessing procedure for obtaining a flux in RT_0, assuming the permeability $\mathbf{K} = K_E\mathbf{I}_2$ and K_E is a constant scalar on each element.

(1) After the numerical approximation p_h is obtained, compute its numerical flux $\mathbf{u}_h = -K_E\nabla p_h$, which is a constant vector on each triangular element.
(2) On each edge, compute the numerical flux average, which is also a constant vector.
(3) For each of the three edges of a triangular element, compute the normal flux (a constant scalar on each edge). Use these three normal fluxes to derive a RT_0 flux $\mathbf{U}_h^*$ on the triangular element.
(4) Note that for an interior edge shared by two triangular elements, the RT_0 coefficients (related to this edge) from two sides have opposite signs.

DGP2 postprocessing for BDM_1 flux. Assume discontinuous quadratic polynomials are used for approximating the primal variable on a triangular mesh. Here we describe a practical and simple postprocessing procedure for obtaining a flux in BDM_1, assuming the permeability $\mathbf{K} = K_E\mathbf{I}_2$ and K_E is a constant scalar on each element.

(1) After the numerical approximation p_h is obtained, compute its numerical flux $\mathbf{u}_h = -K_E\nabla p_h$, which is a vector-valued linear polynomial on each triangular element.
(2) On each edge, compute the numerical flux average, which is also a vector-valued linear polynomial.
(3) For each of the three edges of a triangular element, compute the normal fluxes at the two endpoints. Use these six normal fluxes to derive a BDM_1 flux $\mathbf{U}_h^*$ on the triangular element.
(4) Note that for an interior edge shared by two triangular elements, the BDM_1 coefficients (related to this edge) from two sides have opposite signs and reversed positions.

Table 1
Comparison of degrees of freedom (DOFs) and condition numbers (CondNum) for WGFEMs, MFEMs, and DGFEMs on triangular and rectangular meshes, assuming a rectangular domain has n partitions in both x- and y-directions.

Method	Triangular meshes		Rectangular meshes	
	DOFs	CondNum	DOFs	CondNum
CGP1/CGQ1	n^2+2n+1	$\mathcal{O}(n^2)$	n^2+2n+1	$\mathcal{O}(n^2)$
SIPG(P1/Q1)	$6n^2$	$\mathcal{O}(n^2)$	$4n^2$	$\mathcal{O}(n^2)$
SIPG(P2/Q2)	$12n^2$	$\mathcal{O}(n^2)$	$9n^2$	$\mathcal{O}(n^2)$
WOPSIP(P1)	$6n^2$	$\mathcal{O}(n^4)$	N/A	N/A
MFEM RT_0 or $RT_{[0]}$	$5n^2+2n$	N/A	$3n^2+2n$	N/A
WG(P_0, P_0, RT_0) or WG$(Q_0, P_0, RT_{[0]})$	$5n^2+2n$	$\mathcal{O}(n^2)$	$3n^2+2n$	$\mathcal{O}(n^2)$
WG(P_1, P_1, RT_1) or WG$(Q_1, P_1, RT_{[1]})$	$12n^2+4n$	$\mathcal{O}(n^2)$	$8n^2+4n$	$\mathcal{O}(n^2)$

4. Comparison of WGFEMs with MFEMs and DGFEMs

This section presents comparison of the newly introduced weak Galerkin finite element methods with the widely accepted discontinuous Galerkin methods and the classical mixed finite element methods. In particular, we shall make comparison on these aspects:

- Scheme formulation;
- Approximating finite element spaces;
- Accuracy and convergence rates;
- Features and solving of the discrete linear systems;
- Computational costs including degrees of freedom;
- Implementation issues.

It should be pointed out that DGFEMs offer complete independence of elementwise shape functions and result in two sets of values and hence jumps on each inter-element boundary. The discontinuity is controlled through penalty on these jumps. The MFEMs consider unknowns in both element interiors and on inter-element interfaces. The connection between these two sets of unknowns is enforced through the inf–sup condition and the first equation in (33). While WGFEMs share some features of the above methods, the discrete weak gradient operator used in WGFEMs establishes intrinsic connection of the shape functions in element interiors and those on inter-element boundaries.

4.1. Comparison of WGFEMs with MFEMs

MFEMs rely on mixed formulations, need to satisfy the inf–sup condition, result in indefinite linear systems which require special algorithms but produce numerical fluxes directly along with approximation for the primal variable. WGFEMs rely on usual weak formulations, have plenty of choices for approximating finite element spaces, result in definite linear systems that are easier to solve. WG numerical fluxes can be obtained via simple calculations or a local projection. (See Table 2.)

There are some interesting connection and differences between MFEM RT_0 (respectively MFEM $RT_{[0]}$) and WG (P_0, P_0, RT_0) (respectively WG $(Q_0, P_0, RT_{[0]})$).

(i) The MFEM RT_0 on a triangular mesh uses RT_0 for approximating the flux $-\mathbf{K}\nabla p$; whereas the weak Galerkin (P_0, P_0, RT_0) on a triangular mesh uses RT_0 for approximating the gradient ∇p.
(ii) For both methods, the flux is locally conservative on elements, the norm flux is continuous across element interfaces.
(iii) Both methods have the same number of unknowns. MFEM RT_0 has one pressure unknown on each element and one normal flux unknown on each edge; whereas WG (P_0, P_0, RT_0) has one pressure unknown on each element and one pressure unknown on each edge.
(iv) However, the major difference is that MFEMs result in saddle-point problems which are difficult to solve; whereas WGFEMs result in symmetric definite linear systems that are easier to solve.

4.2. Comparison of WGFEMs with DGFEMs

DGFEMs rely on formulations involving jumps and averages, which need to be carefully dealt in implementation. DGFEMs have more degrees of freedom and DG flux needs post-processing to be continuous in its normal component. WGFEMs rely on novel concepts of weak differential operators and have wide choices for approximating finite element spaces.

The "vectorization" technique for `Matlab` implementation discussed in Section 3.3.3 applies to both DGFEMs and WGFEMs. However, for WGFEMs, there is no need for handling jumps across element interfaces, the implementation is easier than that of DGFEMs.

Table 2
Highlights in comparison of the mixed and weak Galerkin finite element methods.

	Mixed methods	Weak Galerkin
Formulation	Dirichlet BCs natural, Neumann BCs essential	Dirichlet BCs essential, Neumann BCs natural
Shape functions	On edges for flux, In elements for primal	In elements & on edges for primal
Error estimates	Optimal order	Optimal order
Main features(+)	Flux obtained directly	SPD discrete linear systems
Main features(−)	Saddle-point problems	Discrete weak gradients for higher order elements
Two methods produce close results when **K** is piecewise constant		

Table 3
Highlights in comparison of the discontinuous and weak Galerkin finite element methods.

	Discontinuous Galerkin	Weak Galerkin
Formulation	Dirichlet BCs weakly, Neumann BCs natural	Dirichlet BCs essential, Neumann BCs natural
Shape functions	Inside elements, totally discontinuous, weak continuity by penalty	Inside elements & on edges
Error estimates	Optimal/suboptimal order	Optimal order
Main features(+)	Flexible in element geometry	Locally conservative flux, Continuous normal flux
Main features (−)	Penalty factor, Proliferation in unknowns, Flux is not continuous	Solving small linear systems for discrete weak gradients of higher order elements

Table 4
Example 1: Convergence rates of the WGFEM with (P_0, P_0, RT_0) elements on uniform triangular meshes.

$1/h$	$\|Q_h^\circ p - p_h^\circ\|_{L^2}$	Rate	$\|p - p_h^\circ\|_{L^2}$	Rate	$\|\mathbf{u} - \mathbf{u}_h\|_{L^2}$	Rate
8	2.2312e−3	–	6.5174e−2	–	2.5164e−1	–
16	5.6777e−4	1.97	3.2690e−2	0.99	1.2589e−1	0.99
32	1.4257e−4	1.99	1.6358e−2	0.99	6.2954e−2	0.99
64	3.5682e−5	1.99	8.1807e−3	0.99	3.1478e−2	0.99
128	8.9229e−6	1.99	4.0905e−3	0.99	1.5739e−2	1.00

Table 5
Example 1: Convergence rates of the WGFEM with (P_1, P_1, RT_1) elements on uniform triangular meshes.

$n = 1/h$	$\|Q_h^\circ p - p_h^\circ\|_{L^2}$	$\|\mathbf{u} - \mathbf{u}_h\|_{L^2}$
8	7.7758e−05	6.1772e−03
16	9.9421e−06	1.5615e−03
32	1.2572e−06	3.9265e−04
64	1.5804e−07	9.8454e−05
Conv. rate	2.98	1.99

5. Numerical experiments

In this section, we conduct numerical experiments on three benchmark problems and examine the performance of WGFEMs, MFEMs, and DGFEMs. Some mesh data structures recommended in the *iFEM* package [18] are used in our testing code. Some of the techniques discussed in [7] for implementing the lowest order Raviart–Thomas elements in `Matlab` are also adopted.

Example 1 (*A Very Smooth Solution*)**.** Here $\Omega = (0, 1)^2$, a known exact solution is specified as $p(x, y) = \sin(\pi x)\sin(\pi y)$, $\mathbf{K} = \mathbf{I}_2$ (the order 2 identity matrix). Accordingly, the source term $f(x, y) = 2\pi^2 \sin(\pi x)\sin(\pi y)$. This smooth example has been tested widely. A similar example was tested in [11].

For Example 1, WGFEMs demonstrate optimal order convergence in the primal variable and its flux, as demonstrated by the numerical results in Tables 4 and 5. These results agree with the theoretical estimates in Theorem 1.

Table 6 lists the numerical results of WG(P_0, P_1, BDM_1) and MFEM(BDM_1, P_0). For both methods, piecewise constants P_0 are used to approximate the primal variable (pressure), whereas degree-one vector polynomials BDM_1 are used to approximate the flux (velocity). The two methods have the same degrees of freedom and produce numerical results that are very close. For both methods, the pressure error L^2-norm has 1st-order convergence, and the velocity error L^2-norm has 2nd-order convergence, as expected from the theoretical estimates.

However, choosing a good penalty factor is an issue for the DGFEMs, as demonstrated by such a simple problem. Listed in Table 7 are the results of SIPG P_1 with various values of penalty factor α. For $\alpha = 1$, $\|p - p_h\|_{L^2}$ does not show 2nd-order

Table 6
Example 1: Numerical results of WG and MFEM in the primal variable (P_0) and flux (BDM_1).

	WG(P_0, P_1, BDM_1)		MFEM(BDM_1, P_0)	
$1/h$	$\|p - p_h^\circ\|_{L^2}$	$\|\mathbf{u} - \mathbf{u}_h\|_{L^2}$	$\|p - p_h\|_{L^2}$	$\|\mathbf{u} - \mathbf{u}_h\|_{L^2}$
2^3	6.5669e−02	4.7799e−02	6.5687e−02	4.8425e−02
2^4	3.2755e−02	1.2080e−02	3.2757e−02	1.2228e−02
2^5	1.6366e−02	3.0292e−03	1.6367e−02	3.0654e−03
2^6	8.1817e−03	7.5799e−04	8.1817e−03	7.6692e−04
Rate	1.00	1.99	1.00	1.99

Table 7
Example 1: Numerical results of the SIPG scheme with P_1 elements on uniform triangular meshes.

$1/h$	$\alpha = 1$		$\alpha = 3$		$\alpha = 30$	
	Cond#	$\|p - p_h\|_{L^2}$	Cond#	$\|p - p_h\|_{L^2}$	Cond#	$\|p - p_h\|_{L^2}$
8	1.162e+6	3.408e−2	1.374e+5	9.498e−3	3.977e+5	1.836e−2
16	1.571e+7	9.677e−3	6.272e+5	2.504e−3	2.260e+6	4.765e−3
32	1.995e+9	4.183e−3	2.782e+6	6.449e−4	1.137e+7	1.207e−3
64	2.525e+9	1.972e−3	1.197e+7	1.636e−4	4.714e+7	3.036e−4

Table 8
Example 1: Convergence rates of DG P_2 solutions in the primal variable and post-processed flux (BDM_1).

$1/h$	SIPG ($\alpha = 10$)		NIPG ($\alpha = 1$)	
	$\|p - p_h\|_{L^2}$	$\|\mathbf{u} - \mathbf{U}_h^*\|_{L^2}$	$\|p - p_h\|_{L^2}$	$\|\mathbf{u} - \mathbf{U}_h^*\|_{L^2}$
2^3	2.6171e−04	2.9680e−02	3.4790e−03	2.8047e−02
2^4	3.2066e−05	7.3149e−03	8.9717e−04	6.9903e−03
2^5	3.9820e−06	1.8134e−03	2.2712e−04	1.7427e−03
2^6	4.9672e−07	4.5128e−04	5.7091e−05	4.3492e−04
Rate	3.01	2.01	1.97	2.00

Table 9
Example 2: Numerical results of MFEM RT_0.

$1/h$	$\|\mathbf{u} - \mathbf{u}_h\|_{L^2}$	Rate	$\|p - p_h\|_{L^2}$	Rate
2^4	7.601703590820497e−02	–	1.669244690634372e−02	–
2^5	4.840110231960586e−02	0.65	8.307545391465606e−03	1.00
2^6	3.068878633444810e−02	0.66	4.140421356097081e−03	1.00
2^7	1.940984536108590e−02	0.66	2.065773522660539e−03	1.00

convergence. For $\alpha = 3$, $\|p - p_h\|_{L^2}$ does exhibit close to 2nd-order convergence. When the penalty factor increases ten fold to $\alpha = 30$, the condition number becomes much larger but the error in the L^2-norm becomes worse. It is unclear what the optimal penalty factor value is.

Shown in Table 8 are the results of DG P_2 numerical solutions with symmetric formulation (penalty factor $\alpha = 10$) and nonsymmetric formulation (penalty factor $\alpha = 1$). Post-processing is performed to obtain a numerical flux in BDM_1. These numerical results agree with the DG theory [16]. For DG approximation with even-order polynomials, the L^2-norm of the error in the primal variable has only suboptimal order convergence for the nonsymmetric formulation. However, the post-processed BDM_1 flux for this example has 2nd-order convergence for both SIPG and NIPG.

Example 2 (*A Low-Regularity Solution*)**.** Now we consider a frequently tested problem on an L-shaped domain $\Omega = (-1, 1)^2 \setminus ([0, 1] \times [-1, 1])$. We have $\mathbf{K} = \mathbf{I}_2$, $p(x, y) = p(r, \theta) = r^{\frac{2}{3}} \sin\left(\frac{2}{3}\theta\right)$ and $f(x, y) = 0$. A Dirichlet boundary condition is specified using the value of $p(x, y)$ on $\partial\Omega$. It is known that $p \in H^{1+s}(\Omega)$ with $s = \frac{2}{3} - \varepsilon$ for any small positive ε.

Tables 9 and 10 present respectively the numerical results from using MFEM RT_0 and WG (P_0, P_0, RT_0). For MFEM RT_0, one can clearly observe $\frac{2}{3}$-order convergence in the flux, which agrees with the theory in [5]. The convergence rate in the primal variable is actually 1, which is better than the theoretical estimate $\frac{2}{3}$. For WG (P_0, P_0, RT_0), the convergence rates for the flux and primal variable are the same as those for MFEM RT_0. Actually the WG numerical solution is very close to that of the mixed method and their differences are negligible. Additionally, we observe $\frac{4}{3}$-order convergence in $\|Q_h^\circ p - p_h\|_{L^2}$, as stated in Theorem 1.

Shown in Table 11 are the results of applying DG with P_1 polynomials: both the symmetric and nonsymmetric formulation with a penalty factor $\alpha = 4$. Even though the approximation to the primal variable exhibits a 1st order convergence for both SIPG and NIPG, the post-processed RT_0 flux shows no convergence. One might try DG P_2 approximation and

Table 10
Example 2: Numerical results of WG (P_0, P_0, RT_0).

$1/h$	$\|\mathbf{u}-\mathbf{u}_h\|_{L^2}$	Rate	$\|p-p_h\|_{L^2}$	Rate	$\|Q_h^\circ p - p_h\|_{L^2}$	Rate
2^4	7.6017e−2	–	1.6692e−2	–	2.8622e−3	–
2^5	4.8401e−2	0.65	8.3075e−3	1.00	1.1486e−3	1.32
2^6	3.0688e−2	0.66	4.1404e−3	1.00	4.5882e−4	1.32
2^7	1.9409e−2	0.66	2.0657e−3	1.00	1.8281e−4	1.33

Table 11
Example 2: Convergence rates of DG P_1 solutions in the primal variable and post-processed flux (RT_0).

$1/h$	SIPG ($\alpha = 4$)		NIPG ($\alpha = 4$)	
	$\|p-p_h\|_{L^2}$	$\|\mathbf{u}-\mathbf{U}_h^*\|_{L^2}$	$\|p-p_h\|_{L^2}$	$\|\mathbf{u}-\mathbf{U}_h^*\|_{L^2}$
2^4	3.0867e−02	5.9658e−01	3.4854e−02	9.5868e−01
2^5	1.5518e−02	5.9672e−01	1.7525e−02	9.6227e−01
2^6	7.7797e−03	5.9718e−01	8.7849e−03	9.6427e−01
2^7	3.8950e−03	5.9755e−01	4.3976e−03	9.6534e−01
Rate	0.99	0	0.99	0

Table 12
Example 3: Comparison of MFEMs and WGFEMs using triangular RT_0 elements and rectangular $RT_{[0]}$ elements.

$1/h$	Triangular meshes $\max_{\varepsilon_h} \lvert p_h^{MFEM} - p_h^{WG}\rvert$	Rectangular meshes $\max_{\varepsilon_h} \lvert p_h^{MFEM} - p_h^{WG}\rvert$
20	2.252642516964443e−13	1.609823385706477e−14
40	7.779332733548472e−13	2.565725409908737e−13
60	9.641176745844859e−13	5.186961971048731e−13
80	2.427724687947830e−12	1.314948150366035e−12

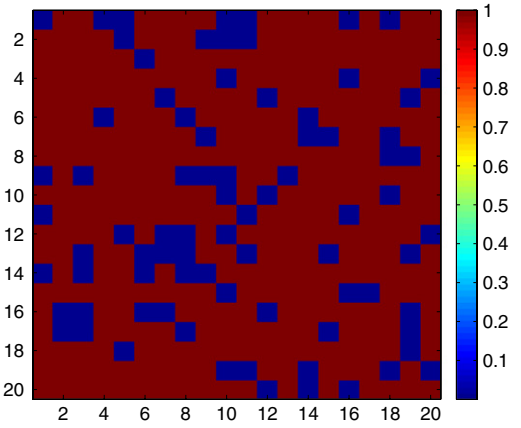

Fig. 4. Example 3: A heterogeneous permeability field on 20×20 gridblocks: permeability is 1 on red locks but only 10^{-6} on blue blocks. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

postprocessing for BDM_1 flux, but that results in significantly more unknowns. There will not be much gain in convergence in the approximations of the primal variable and flux, since the problem possesses low regularity.

Example 3 (*A Heterogeneous Permeability Field*)**.** We solve the boundary value problem (1) with a heterogeneous permeability field given in [19] (see Fig. 4). Dirichlet and Neumann boundary conditions are specified as follows

$$p = 1, \quad \text{left}; \qquad p = 0, \quad \text{right}; \qquad -(\mathbf{K}\nabla p)\cdot\mathbf{n} = 0, \quad \text{top or bottom}.$$

No analytical pressure solution is known for this example. We run MFEM RT_0 and WG (P_0, P_0, RT_0) on triangular meshes and MFEM $RT_{[0]}$ and WG $(Q_0, P_0, RT_{[0]})$ on rectangular meshes. The mesh size varies from $h = 1/20$ to $h = 1/80$. Shown in Fig. 5 are the profiles of the numerical pressure (in color) and velocity vector (in black arrow vectors) obtained from using MFEM RT_0 on a triangular mesh and WGFEM $(Q_0, P_0, RT_{[0]})$ on a rectangular mesh. Note that both the mixed method and weak Galerkin method use piecewise constants to approximate the primal variable pressure. The results listed in Table 12 show that **MFEMs and WGFEMs produce very close results when the permeability is a piecewise constant**, see [9].

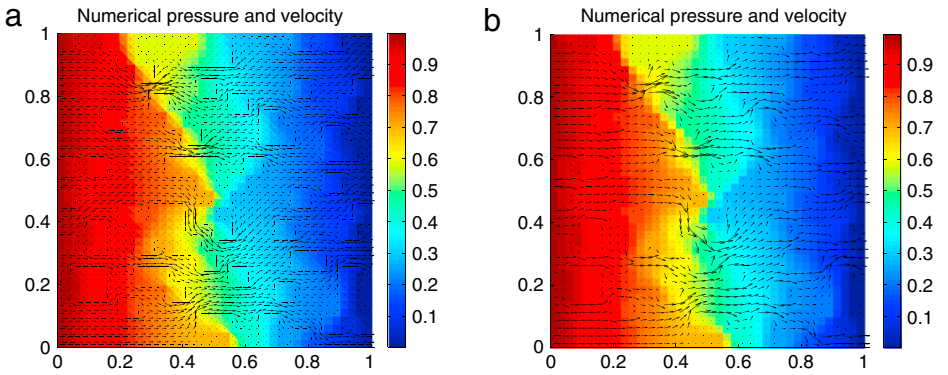

Fig. 5. Example 3: Profiles of numerical pressure (in color) and velocity vector (in black arrow vectors): (a) MFEM RT_0 with $h = 1/40$, (b) WG (Q_0, P_0, $RT_{[0]}$) with $h = 1/40$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

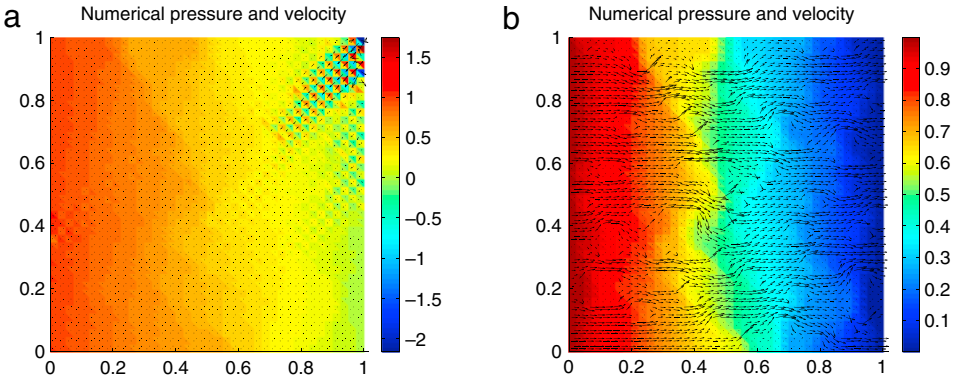

Fig. 6. Example 3: Profiles of numerical pressure (in color) and velocity vector (in black arrow vectors): (a) SIPG P_1, $\alpha = 1$ with $h = 1/40$, (b) SIPG P_1, $\alpha = 3$ with $h = 1/40$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

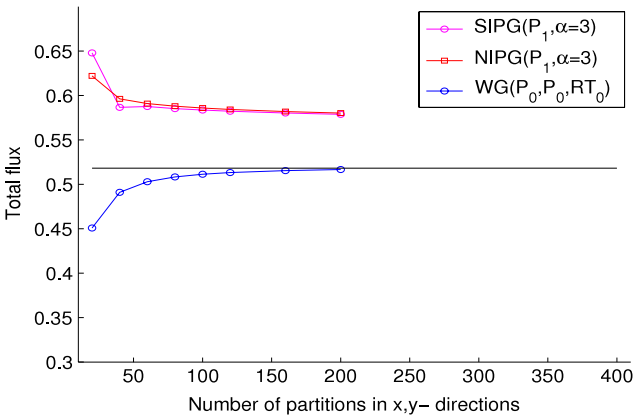

Fig. 7. Example 3: The total fluxes of WG (P_0, P_0, RT_0) and DG (SIPG,NIPG) with penalty factor $\alpha = 3$.

This example was also studied in [3]. Quadratic and cubic polynomials were used in the DG framework to obtain numerical pressures. The polynomials ought to have degree at least 2 in order to project the DG flux into a BDM finite element space. Such a DG scheme results in a significant number of unknowns, see Table 1.

Here we test DG with P_1 polynomials and the post-processing procedure for projection into RT_0, as discussed in Section 3.3.3. As shown in Fig. 6, for the SIPG with penalty factor $\alpha = 1$ on a triangular mesh with mesh size $h = 1/40$, the numerical pressure and post-processed velocity are not useful. However, the results for $\alpha = 3$ are good.

To further examine the performance of these numerical methods, we calculate the total flux as suggested in [19]. As shown in Fig. 7, the total flux obtained from using the cell-centered finite difference method [20,9] on a rectangular mesh with $h = 1/400$ is used as a reference. The results from MFEM RT_0 are not shown in the figure, since they are basically

equivalent to those of WG (P_0, P_0, RT_0). The results of WG (P_0, P_0, RT_0) show good approximation. They exhibit the feature "approximating from below", which is typical for the $\mathbf{K}^{-1}$-methods [21]. The results of NIPG and SIPG with P_1 polynomials and penalty factor $\alpha = 3$ are also shown in Fig. 7. They exhibit the feature "approximating from above", which is typical for the $\mathbf{K}$-methods [21].

6. Concluding remarks

In this paper, we have conducted a comparative study on the newly introduced WGFEMs, the widely accepted DGFEMs, and the classical MFEMs. Here is a summary:

(i) WGFEMs are viable alternatives of MFEMs;
(ii) Compared to DGFEMs, WGFEMs are easier to use and performs better with respect to continuous normal flux across element interfaces, less unknowns, and no need for choosing penalty factors.

The following main features of the WGFEMs have been clearly observed:

- Locally conservative by design;
- Normal flux is continuous across element interfaces;
- Less unknowns (compared to DGFEMs);
- Definite discrete linear systems (compared to MFEMs);
- No need for choosing penalty factors (compared to DGFEMs).

It is accepted that CGFEMs are not locally conservative, although there are different opinions and debates on this statement. There are at least two remedies for rendering CG locally conservative:

(i) Postprocessing as presented in [8];
(ii) Enhancement by the piecewise constant space as investigated in [2]. This is known as the enhanced Galerkin (EG).

It should be interesting to compare WG with CG postprocessing and EG, examining the connection and differences among these methods.

This paper focuses on the comparison of WG with DG and MFEM in 2-dim. But conclusions presented in Tables 2 and 3 still hold for the 3-dimensional versions of these methods.

There are already a variety of numerical methods for the prototype problem (1), including the recently developed weak Galerkin methods. Different methods have different features. This paper intends to offer some guidelines for making choices. We have developed a `Matlab` code package that includes WGFEMs, DGFEMs with post-processing, and MFEMs. This package will be further expanded to include CGFEMs with post-processing and EG.

Acknowledgments

G. Lin would like to acknowledge support by the Applied Mathematics program of the US DOE Office of Advanced Scientific Computing Research and Pacific Northwest National Laboratory's Carbon Sequestration Initiative, which is part of the Laboratory Directed Research and Development Program. J. Liu and F. Sadre-Marandi were partially supported by the National Science Foundation under Grant No. DMS-0915253. Computations were performed using the computational resources of Pacific Northwest National Laboratory (PNNL) Institutional Computing cluster systems and the National Energy Research Scientific Computing Center at Lawrence Berkeley National Laboratory. The PNNL is operated by Battelle for the US Department of Energy under Contract DE-AC05-76RL01830.

References

[1] S.C. Brenner, L.R. Scott, The Mathematical Theory of Finite Element Methods, third ed., Springer, 2008.
[2] S. Sun, J. Liu, A locally conservative finite element method based on piecewise constant enrichment of the continuous Galerkin method, SIAM J. Sci. Comput. 31 (2009) 2528–2548.
[3] P. Bastian, B. Riviére, Superconvergence and H(div) projection for discontinuous Galerkin methods, Int. J. Numer. Methods Fluids 42 (2003) 1043–1057.
[4] M. Benzi, G.H. Golub, J. Liesen, Numerical solution of saddle point problems, Acta Numer. 14 (2005) 1–137.
[5] F. Brezzi, M. Fortin, Mixed and Hybrid Finite Element Methods, Springer-Verlag, 1991.
[6] J. Wang, X. Ye, A weak Galerkin finite element method for second order elliptic problems, J. Comput. Appl. Math. 241 (2013) 103–115.
[7] C. Bahriawati, C. Carstensen, Three matlab implementations of the lowest-order Raviart–Thomas MFEM with a posteriori error control, Comput. Methods Appl. Math. 5 (2005) 333–361.
[8] B. Cockburn, J. Gopalakrishnan, H. Wang, Locally conservative fluxes for the continuous Galerkin method, SIAM J. Numer. Anal. 45 (2007) 1742–1776.
[9] G. Lin, J. Liu, L. Mu, X. Ye, Weak Galerkin finite element methods for Darcy flow: Anisotropy and heterogeneity, J. Comput. Phys. (2013) in press.
[10] J. Liu, R. Cali, A note on the approximation properties of the locally divergence-free finite elements, Int. J. Numer. Anal. Model. 5 (2008) 693–703.
[11] L. Mu, J. Wang, Y. Wang, X. Ye, A computational study of the weak Galerkin method for second-order elliptic equations, Numer. Algebra 63 (2013) 753–777.
[12] L. Chen, J. Wang, X. Ye, A posteriori error estimates for weak Galerkin finite element methods for second order elliptic problems, J. Sci. Comput. 59 (2014) 496–511.
[13] J. Liu, L. Mu, X. Ye, L2 Error estimation for DGFEM for elliptic problems with low regularity, Appl. Math. Lett. 25 (2012) 1614–1618.
[14] C. Wang, J. Wang, A hybridized weak Galerkin finite element method for the biharmonic equation, arXiv:1402.1157 2014.
[15] Y. Epshteyn, B. Riviere, Estimation of penalty parameters for symmetric interior penalty Galerkin methods, J. Comput. Appl. Math. 206 (2007) 843–872.

[16] B. Riviere, Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations: Theory and Implementation, SIAM, 2008.
[17] S.C. Brenner, L. Owens, L.Y. Sung, A weakly over-penalized symmetric interior penalty method, Electron. Trans. Numer. Anal. 30 (2008) 107–127.
[18] L. Chen, iFEM: An innovative finite element method package in Matlab.
[19] L.J. Durlofsky, Accuracy of mixed and control volume finite element approximations to Darcy velocity and related quantities, Water Resour. Res. 30 (1994) 965–973.
[20] R.E. Ewing, The Mathematics of Reservior Simulation, SIAM, 1983.
[21] T.F. Russell, Relationship Among Some Conservative Discretizaation Methods, in: Lec. Notes Phys., vol. 552, Springer, 2000.