Research Article

Mathematical Methods in the Applied Sciences

Received 19 February 2013

Published online 20 June 2013 in Wiley Online Library

(wileyonlinelibrary.com) DOI: 10.1002/mma.2856
MOS subject classification: 65N30; 65N50; 35J15

A selective immersed discontinuous Galerkin method for elliptic interface problems

Xiaoming He[a*†], Tao Lin[b] and Yanping Lin[c,d]

Communicated by Q. Wang

This article proposes a selective immersed discontinuous Galerkin method based on bilinear immersed finite elements (IFE) for solving second-order elliptic interface problems. This method applies the discontinuous Galerkin formulation wherever selected, such as those elements around an interface or a singular source, but the regular Galerkin formulation everywhere else. A selective bilinear IFE space is constructed and applied to the selective immersed discontinuous Galerkin method based on either the symmetric or nonsymmetric interior penalty discontinuous Galerkin formulation. The new method can solve an interface problem by a rectangular mesh with local mesh refinement independent of the interface even if its geometry is nontrivial. Meanwhile, if desired, its computational cost can be maintained very close to that of the standard Galerkin IFE method. It is shown that the selective bilinear IFE space has the optimal approximation capability expected from piecewise bilinear polynomials. Numerical examples are provided to demonstrate features of this method, including the effectiveness of local mesh refinement around the interface and the sensitivity to the penalty parameters. Copyright © 2013 John Wiley & Sons, Ltd.

Keywords: interface problems; immersed finite elements; selective discontinuous Galerkin; penalty; mesh refinement

1. Introduction

Because of its flexibility for mesh/polynomial refinement, localizability, stability, parallelizability, and less numerical diffusion/dipersion, the discontinuous Galerkin(DG) method has been widely extended and applied to solve many partial differential equations. Since its introduction in the 1970s [1], numerous DG methods have been developed, such as the total variation bounded Runge–Kutta discontinuous Galerkin method [2–8], the local discontinuous Galerkin method [9–16], the interior penalty discontinuous Galerkin (IPDG) method [17–23], the mixed DG method [24–26], the central DG method [27–29], the hybridizable DG method [30–32], the space-time DG method [33, 34], the positivity-preserving DG method [35, 36], and many others [37–47]. Some recent developments and analysis of DG methods can be also found in [48–58] and references therein. For more details, we refer readers to the survey papers [59–61].

The IPDG method [17–21, 23] was introduced to solve second-order elliptic equations. One important feature of this method is its capability to easily incorporate p and h local refinement. Therefore, it is natural to apply them to elliptic interface problems because local refinement is usually needed around an interface. However, the IPDG method needs much more finite element basis functions than the standard Galerkin finite element method on the same mesh. This leads to a much larger algebraic system to be solved with a significantly higher computational cost. Hence, it is of great importance to find out how to take advantage of the local refinement feature of DG methods while keeping the computational cost as close as possible to that of standard Galerkin finite element methods. Coupling [62–64] or hybridization [65, 66] of DG methods and conforming finite element methods are efficient ways to resolve this challenge. Following this idea, we will introduce a selective immersed discontinuous Galerkin (SIDG) method for solving a second-order elliptic interface problem on a mesh independent of the interface. The basic idea is to use the IPDG formulation wherever local refinement is needed, such as around an interface or a singular source, but the standard Galerkin formulation everywhere else. The SIDG is originated from the DG methods, but it has the potential to be more efficient than the standard DG methods in the literature because the degree of freedom is increased only in the selected area for the mesh refinement.

[a] *Department of Mathematics and Statistics, Missouri University of Science and Technology, Rolla, MO 65409, USA*
[b] *Department of Mathematics, Virginia Tech, Blacksburg, VA 24061, USA*
[c] *Department of Applied Mathematics, Hong Kong Polytechnic University, Hung Hom, Hong Kong*
[d] *Department of Mathematical and Statistics Science, University of Alberta, Edmonton AB, T6G 2G1, Canada*
**Correspondence to: Xiaoming He, Department of Mathematics and Statistics, Missouri University of Science and Technology, Rolla, MO 65409, USA.*
[†] *E-mail: hex@mst.edu*

Copyright © 2013 John Wiley & Sons, Ltd.

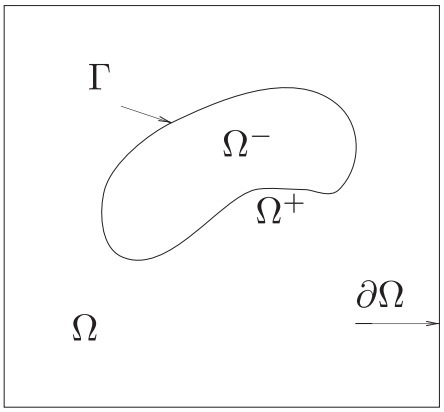

Figure 1. A sketch of the domain for the interface problem.

Consider the following typical boundary value problem

$$-\nabla \cdot (\beta \nabla u) = f, \quad (x, y) \in \Omega, \tag{1.1}$$

$$u|_{\partial\Omega} = g, \tag{1.2}$$

together with the jump conditions on the interface Γ:

$$[u]|_{\Gamma} = 0, \tag{1.3}$$

$$\left[\beta \frac{\partial u}{\partial n}\right]|_{\Gamma} = 0. \tag{1.4}$$

As illustrated in Figure 1, without loss of generality, we assume that $\Omega \subset \boldsymbol{R}^2$ is a rectangular domain, the interface Γ is a curve separating Ω into two subdomains Ω^-, Ω^+ such that $\overline{\Omega} = \overline{\Omega^-} \cup \overline{\Omega^+} \cup \Gamma$, and the coefficient $\beta(x, y)$ is a piecewise constant function defined by

$$\beta(x, y) = \begin{cases} \beta^-, & (x, y) \in \Omega^-, \\ \beta^+, & (x, y) \in \Omega^+. \end{cases}$$

Interface problem (1.1)–(1.4) can be solved by conventional finite difference or finite element methods, provided that a so-called body-fitting mesh is utilized [67–71]. However, there are many applications, such as moving interface problems [72] and Particle-In-Cell method for plasma particle simulation [73–77], in which it is preferable to solve the interface problem on a structured Cartesian mesh.

Therefore, different numerical methods on Cartesian mesh have been developed for interface problems, including immersed boundary method [78–80], immersed interface method [81–83], matched interface and boundary methods [84, 85], cut-cell methods [86, 87], embedded boundary methods [88], Cartesian grid methods [89, 90], generalized finite element methods [91–94], extended finite element methods [95–97], and Eulerian–Lagrangian localized adjoint methods [98–100].

Those methods based on finite difference formulation on Cartesian mesh, such as the immersed interface method, is efficient for solving the interface problems. But it has no guaranty of a symmetric positive definite linear system for the interface problem (1.1)–(1.4). It is also not straightforward to carry out the mesh refinement for this method. With a similar 'immersed' idea under the framework of finite element methods, the immersed finite elements (IFEs) have been developed [72, 83, 101–126]. Similar to the aforementioned numerical methods, the IFEs allow the elements to be cut through by the interface; hence, a mesh independent of the interface, such as a Cartesian mesh, can be used for solving interface problems. Furthermore, the linear system arising from the IFEs is naturally symmetric positive definite. The combination of the proposed selective DG formulation and the IFEs allows an adaptive structured mesh to be used for solving interface problems while keeping the computational cost close to that of the standard Galerkin method. In this article, we will focus on applying a bilinear IFE [109, 117] space in the SIDG method.

The rest of this article is organized as follows. In Section 2, we discuss the formulation of the SIDG method. In Section 3, we form a selective bilinear IFE space and apply it in the SIDG method for solving elliptic interface problems. In Section 4, we carry out the interpolation error estimation for the selective bilinear IFE space. In Section 5, we present some numerical examples to illustrate the features of this SIDG method.

2. Formulation of the selective immersed discontinuous Galerkin method

In this section, we will formulate the SIDG method for second-order elliptic interface problems described by (1.1)–(1.4). The basic idea can be extended to other problems.

Copyright © 2013 John Wiley & Sons, Ltd.

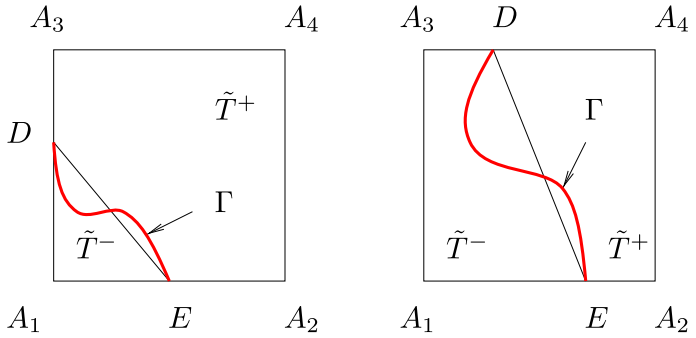

Figure 2. Two typical rectangular interface elements.

First, we introduce some conventions. For a set $\Lambda \subset \Omega$, we let $\Lambda^s = \Lambda \cap \Omega^s, s = \pm$. Let $\mathcal{T}_h$ ($h > 0$) be a mesh of the solution domain Ω, and let $\mathcal{E}_h$ be the set of edges of elements in $\mathcal{T}_h$. The mesh consists of interface elements whose interiors are cut by interfaces and the rest of elements called noninterface elements. Interface edges and noninterface edges have a similar meaning. Consider a typical rectangular element $T \in \mathcal{T}_h$. Assume that the four vertices of T are $A_i, i = 1, 2, 3, 4$ in the counterclockwise order, with $A_i = (x_i, y_i)^t$. If T is an interface element, then we use $D = (x_D, y_D)^T$ and $E = (x_E, y_E)^T$ to denote the interface points on its edges. The line $\overline{DE}$ separates T into two subelements: $\tilde{T}^-$ and $\tilde{T}^+$. Here, $\tilde{T}^-$ is the polygon contained in T sharing at least one vertex of T^- inside Ω^-, and the other subelement is $\tilde{T}^+$. There are two types of interface elements. We call an interface element a Type I interface element if its interface points, D and E, are on two adjacent edges; otherwise, we call this element a Type II interface element. Related illustrations can be seen in Figure 2.

We let $\mathcal{E}_h^o$ denote the collection of all interior element edges of mesh $\mathcal{T}_h$ and let $\mathcal{E}_S \subset \mathcal{E}_h^o$ denote the collection of selected element edges where a discontinuity will be allowed by the DG formulation. Also, we let $\mathcal{E}_h^D$ denote the collection of element edges on the boundary and let Ω_S be the union of elements in $\mathcal{T}_h$ having at least one of its edges in $\mathcal{E}_S$. The key point of the SIDG method is to properly select these two sets based on the features of the problem to be solved. For our model interface problem, the discontinuity of the coefficient results in the lack of smoothness across the interface. In order to apply local mesh refinement in the vicinity of the interface to obtain a better approximation, we can choose Ω_S to be the union of all interface elements, and $\mathcal{E}_S$ to be the set of all edges of elements in $\overline{\Omega_S}$.

To describe the formulation, we will use the following spaces:

$$PH_S^1(\mathcal{T}_h) = \left\{ v : [v]|_e = 0 \ \forall e \in \mathcal{E}_h / \mathcal{E}_S;\ v|_T \in C(T) \cap H^1(T) \ \forall T \in \mathcal{T}_h \right\},$$
$$PH_{\text{int}}^2(\Lambda) = \left\{ u \in C(\Lambda), u|_{\Lambda^s} \in H^2(\Lambda^s), s = -, +, \left[\beta \frac{\partial u}{\partial \mathbf{n}_\Gamma} \right]_{\Gamma \cap \Lambda} = 0 \right\}.$$

Here, $\Lambda \subset \Omega$, whose interior is cut through by Γ.

Following the usual DG framework [26], we assume that the solution u of the model interface problem (1.1)–(1.4) is in $PH_{\text{int}}^2(\Omega)$. Let $(\cdot, \cdot)_\Lambda$ denote the usual L^2 inner product of two functions on a set Λ. For each $e \in \mathcal{E}_h$, we let T_1, T_2 be the two elements in $\mathcal{T}_h$ such that $e = \partial T_1 \cap \partial T_2$ and let ν be the unit normal vector of e exterior to T_2. For a rectangular mesh, we define T_2 to be the element on the right hand side of e if e is vertical, and the element over e if e is horizontal. Also, $\forall e \in \mathcal{E}_h^D$, we let ν be the unit normal vector of e exterior to Ω. On each $e \in \mathcal{E}_S$, we define the usual jump and average functions as follows

$$[v] = (v|_{T_2})|_e - (v|_{T_1})|_e, \quad \{v\} = \frac{1}{2}((v|_{T_2})|_e + (v|_{T_1})|_e).$$

$\forall e \in \mathcal{E}_h^D$, the jump and average functions are just the function itself.

Multiplying (1.1) by $v \in PH_S^1(\mathcal{T}_h)$, integrating over each element $T \in \mathcal{T}_h$, applying Green's formula, and summing over $T \in \mathcal{T}_h$, we have

$$-\sum_{e \in \mathcal{E}_h^D} (\beta \nabla u \cdot \nu, v)_e - \sum_{e \in \mathcal{E}_S} [(\beta \nabla u \cdot \nu, v)_{\partial T_2 \cap e} - (\beta \nabla u \cdot \nu, v)_{\partial T_1 \cap e}] + \sum_{T \in \mathcal{T}_h} (\beta \nabla u, \nabla v)_T = \sum_{T \in \mathcal{T}_h} (f, v)_T. \tag{2.5}$$

In addition, assuming $\beta \nabla u \cdot \nu$ is continuous almost everywhere on each $e \in \mathcal{E}_S$, we have

$$\sum_{e \in \mathcal{E}_S} ([\beta \nabla u \cdot \nu], \{v\})_e = 0.$$

Then, applying the algebraic identity

$$ab - cd = \frac{1}{2}(a + c)(b - d) + \frac{1}{2}(a - c)(b + d)$$

Copyright © 2013 John Wiley & Sons, Ltd.

to the second term in (2.5), we get

$$\sum_{T\in\mathcal{T}_h}(\beta\nabla u,\nabla v)_T-\sum_{e\in\mathcal{E}_S}(\{\beta\nabla u\cdot\nu\},[v])_e-\sum_{e\in\mathcal{E}_h^D}(\beta\nabla u\cdot\nu,v)_e=\sum_{T\in\mathcal{T}_h}(f,v)_T.$$

Define

$$b(u,v)=\sum_{T\in\mathcal{T}_h}(\beta\nabla u,\nabla v)_T, \tag{2.6}$$

$$J_S(u,v)=\sum_{e\in\mathcal{E}_S}(\{\beta\nabla u\cdot\nu\},[v])_e+\sum_{e\in\mathcal{E}_h^D}(\beta\nabla u\cdot\nu,v)_e, \tag{2.7}$$

$$a_S(u,v)=b(u,v)-J_S(u,v), \tag{2.8}$$

$$L(v)=\sum_{T\in\mathcal{T}_h}(f,v)_T. \tag{2.9}$$

Then, a weak formulation of (1.1) is to find a $u\in PH^2_{\text{int}}(\Omega)$ such that

$$a_S(u,v)=L(v),\quad\forall v\in PH^1_S(\mathcal{T}_h). \tag{2.10}$$

Remark 2.1
Because $u\in PH^2_{\text{int}}(\Omega)$, we have $(\beta\nabla u\cdot\nu)|_e\in H^{1/2}(e)$. In fact, we can use the space $H^1(\Omega)$ instead of the space $PH^2_{\text{int}}(\Omega)$ for u to reduce the smoothness requirement. That is, we can find a $u\in H^1(\Omega)$ such that (2.10) is true. Then we have $(\beta\nabla u\cdot\nu)|_e\in H^{-1/2}(e)$.

Because the bilinear form of the above weak formulation is neither symmetric nor positive definite, we follow [17, 19, 23, 26, 127] to generate a symmetric positive definite discontinuous weak formulation as follows. Consider the following penalty term

$$J_{S\theta}(u,v)=\sum_{e\in\mathcal{E}_S}\theta_e([u],[v])_e+\sum_{e\in\mathcal{E}_h^D}\theta_e(u,v)_e,$$

where θ_e is the penalty parameter. Define

$$\begin{aligned}a^-_{S\theta}(u,v)&=b(u,v)-J_S(u,v)-J_S(v,u)+J_{S\theta}(u,v),\\ L^-_\theta(v)&=L(v)-\sum_{e\in\mathcal{E}_h^D}(\beta\nabla v\cdot\nu,g)_e+\sum_{e\in\mathcal{E}_h^D}\theta_e(g,v)_e.\end{aligned}$$

Then, the symmetric selective discontinuous weak formulation is to find a $u\in PH^2_{int}(\Omega)$ such that

$$a^-_{S\theta}(u,v)=L^-_\theta(v),\quad\forall v\in PH^1_S(\mathcal{T}_h). \tag{2.11}$$

On the basis of the convergence analysis for the related DG finite element method in [17, 19, 23, 26], we usually need to choose the penalty parameter such that

$$\theta_e=\frac{C_*}{h}, \tag{2.12}$$

where C_* is a constant such that $C_*>C_0$ for some positive constant C_0. This restriction can be relaxed by the nonsymmetric formulation [18, 20, 21, 26]: Find a $u\in PH^2_{int}(\Omega)$ such that

$$a^+_{S\theta}(u,v)=L^+_\theta(v),\quad\forall v\in PH^1_S(\mathcal{T}_h), \tag{2.13}$$

where

$$\begin{aligned}a^+_{S\theta}(u,v)&=b(u,v)-J_S(u,v)+J_S(v,u)+J_{S\theta}(u,v),\\ L^+_\theta(v)&=L(v)+\sum_{e\in\mathcal{E}_h^D}(\beta\nabla v\cdot\nu,g)_e+\sum_{e\in\mathcal{E}_h^D}\theta_e(g,v)_e.\end{aligned}$$

The analysis in [18, 20, 21, 26] suggests the following choice for the penalty parameter:

$$\theta_e=\frac{C_{**}}{h}, \tag{2.14}$$

where C_{**} is a just positive constant.

Copyright © 2013 John Wiley & Sons, Ltd.

Then, we can use the IFEs [109, 110, 117–119] and the weak formulations aforementioned to solve an interface problem on a rectangular mesh or a structured triangular mesh. Suppose we have constructed a selective IFE space $S_h^S(\Omega) \subset PH_S^1(\mathcal{T}_h)$ according to a certain selecting rule for $\mathcal{E}_S$, then our symmetric SIDG method is to find $u_h \in S_h^S(\Omega)$ such that

$$a_{S\theta}^{-}(u_h, v_h) = L_{\theta}^{-}(v_h), \quad \forall v_h \in S_h^S(\Omega), \tag{2.15}$$

and our nonsymmetric SIDG method is to find $u_h \in S_h^S(\Omega)$ such that

$$a_{S\theta}^{+}(u_h, v_h) = L_{\theta}^{+}(v_h), \quad \forall v_h \in S_h^S(\Omega). \tag{2.16}$$

Remark 2.2
The SIDG method depends on two key components: $\mathcal{E}_S$ and $S_h^S(\Omega)$. $\mathcal{E}_S$ and/or Ω_S depend on the problem and our desire to use the DG formulation at the locations needed. The selective immersed finite element space $S_h^S(\Omega)$ can then be defined according to $\mathcal{E}_S, \Omega_S$ and our desire to include other features, such as the need to control computational cost. As we can see later, a suitable choice of these components can generate a convergent method for solving an interface problem on a rectangular mesh with local mesh refinement capability at a reduced computational cost.

Remark 2.3
If we choose $\mathcal{E}_S$ to be the set of all interior edges, then Ω_S is the union of all elements in $\mathcal{T}_h$, and the SIDG method becomes the usual IPDG method with IFEs [111].

3. Selective immersed discontinuous Galerkin method with a bilinear immersed finite elements space

In this section, we will first use the bilinear IFE functions introduced in [109, 119, 128] to form a selective bilinear IFE space $S_h^{Sb}(\Omega)$. Then, we apply this space to the SIDG method for solving the interface problem and discuss its advantages.

3.1. A selective bilinear immersed finite elements space

First, we briefly recall the piecewise bilinear IFE functions discussed in [109, 119, 128]. On a typical interface rectangular element $T \in \mathcal{T}_h$ with 4 vertices $A_i = (x_i, y_i)^t$, $i = 1, 2, 3, 4$, see Figure 2 for the illustration, we define the piecewise bilinear IFE functions as follows:

$$\phi(x,y) = \begin{cases} \phi^-(x,y) = a^-x + b^-y + c^- + \mathrm{d}^-xy, & (x,y) \in \tilde{T}^-, \\ \phi^+(x,y) = a^+x + b^+y + c^+ + \mathrm{d}^+xy, & (x,y) \in \tilde{T}^+, \\ \phi^-(D) = \phi^+(D), \ \ \phi^-(E) = \phi^+(E), \ \ \mathrm{d}^- = \mathrm{d}^+, & \\ \int_{\overline{DE}} \left(\beta^- \frac{\partial \phi^-}{\partial \mathbf{n}_{\overline{DE}}} - \beta^+ \frac{\partial \phi^+}{\partial \mathbf{n}_{\overline{DE}}} \right) \mathrm{d}s = 0. & \end{cases} \tag{3.17}$$

We let $\phi_i(X)$ be the bilinear IFE function such that

$$\phi_i(x_j, y_j) = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j, \end{cases}$$

for $1 \leq i, j \leq 4$, and we call them the bilinear IFE nodal basis functions on an interface element T. For every element $T \in \mathcal{T}_h$, we let $S_h(T) = \mathrm{span}\{\phi_i, \ i = 1, 2, 3, 4\}$, where ϕ_i, $i = 1, 2, 3, 4$, are the standard bilinear nodal basis functions for a noninterface element T; otherwise, ϕ_i, $i = 1, 2, 3, 4$, are the immersed bilinear basis functions defined previously.

Remark 3.1
On an interface element T, according to [109], every function $\phi \in S_h(T)$ satisfies the flux jump condition on $\Gamma \cap T$ exactly in the following weak sense:

$$\int_{\Gamma \cap T} \left[\beta \frac{\partial \phi}{\partial \mathbf{n}_\Gamma} \right] |_\Gamma \mathrm{d}s = 0.$$

We need to introduce a few terminologies for describing the selective bilinear IFE space. Assume that we start from a rectangular mesh $\mathcal{T}_h^0$ of Ω without any hanging nodes, and we call its elements the 0th level elements. If needed, we refine $\mathcal{T}_h^0$ once by dividing each of a collection of selected elements in $\mathcal{T}_h^0$ into 4 congruent subrectangles to generate a new mesh $\mathcal{T}_h^1$. We call those new smaller rectangles in $\mathcal{T}_h^1$ the first level elements. Repeating this procedure to the nth level, we can generate a refined mesh $\mathcal{T}_h = \mathcal{T}_h^n$, which can contain elements from level 0 to level n.

For each node $(x_N, y_N)^t$ in a mesh $\mathcal{T}_h$, we define its associated elements to be those elements in which $(x_N, y_N)^t$ is a vertex. For example, the associated elements of node K in Figure 3 are $\square NKIL$ and $\square JEKN$, but $\square EBFI$ is not an associated element of node K even though it is a neighboring element of K. The associated elements of node I in Figure 3 are $\square NKIL$, $\square EBFI$, $\square IFCG$, and $\square HIGD$. We then use $\mathcal{T}_{h,N} \subset \mathcal{T}_h$ to represent the collection of all associated elements of a node $(x_N, y_N)^t$.

Copyright © 2013 John Wiley & Sons, Ltd.

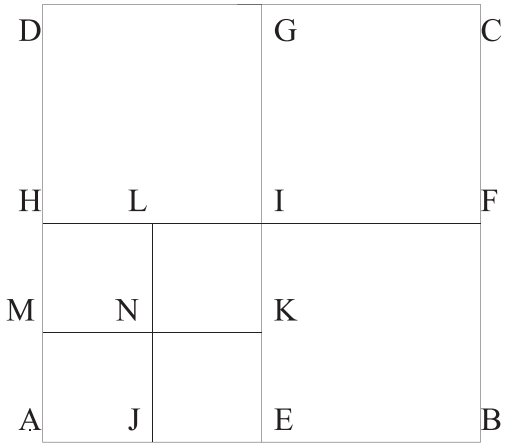

Figure 3. A sketch for the associated elements, coarsenable sets, and hanging nodes.

At each node $(x_N, y_N)^t$ of $\mathcal{T}_h$, we also define a coarsenable set to be a subset $\mathcal{T}_{h,N,c} \subset \mathcal{T}_{h,N}$ such that

(1) $\left|\mathcal{T}_{h,N,c}\right| \geq 2$, that is, $\mathcal{T}_{h,N,c}$ must contain at least two associated elements of $(x_N, y_N)^t$.
(2) All elements in $\mathcal{T}_{h,N,c}$ are in the same level.
(3) Each element in $\mathcal{T}_{h,N,c}$ shares at least one element edge with another element in $\mathcal{T}_{h,N,c}$.

For example, for the node *I* in Figure 3,

$$\{\square EBFI, \square IFCG, \square HIGD\},\ \{\square EBFI, \square IFCG\},\ \{\square IFCG, \square HIGD\}$$

are coarsenable sets, but $\{\square EBFI,\ \square HIGD\}$ is not a coarsenable set. For node *I*, there is no coarsenable set containing $\square NKIL$.

For each coarsenable set $\mathcal{T}_{h,N,c}$ at a node $(x_N, y_N)^t$, we can define a piecewise bilinear function $\Psi_{N,c}$ as follows:

$$\Psi_{N,c}|_{T\in\mathcal{T}_{h,N,c}} \in S_h(T),\ \ \Psi_{N,c}(x_N, y_N) = 1 \text{ but } \Psi_{N,c} \text{ is zero at other nodes,}$$
$$\Psi_{N,c}|_{T\notin\mathcal{T}_{h,N,c}} = 0,\ \ \Psi_{N,c}|_{\cup\mathcal{T}_{h,N,c}} \text{ is continuous at all nodes in } \mathcal{T}_{h,N,c}.$$

Then, we define the selective global bilinear IFE basis functions associated with each node $(x_N, y_N)^t \in \mathcal{T}_h$ as follows:

Step 1: For each associated element $T_{N,i} \in \mathcal{T}_{h,N}$, we define a piecewise bilinear function $\Psi_{N,i}$ associated with $(x_N, y_N)^t$ by the zero extension of the local bilinear IFE or FE nodal basis function $\phi(x, y)$ on $T_{N,i}$ such that $\Psi_{N,i}(x_N, y_N) = 1$.
Step 2: We select a collection of coarsenable sets $\mathcal{T}_{h,N,c_j}, j = 1, 2, \cdots, m_N$ and introduce piecewise bilinear functions $\Psi_{N,c_j}, j = 1, 2, \cdots, m_N$.
Step 3: We discard every $\Psi_{N,i}$ defined earlier if $T_{N,i}$ is contained in one of the selected coarsenable sets $\mathcal{T}_{h,N,c_j}, j = 1, 2, \cdots, m_N$ and call all the remaining piecewise bilinear functions the selective global bilinear IFE basis functions associated with the $(x_N, y_N)^t$.

Finally, we define the selective bilinear IFE space $S_h^{Sb}(\Omega)$ to be the space spanned by the selective global bilinear IFE basis functions associated with all the nodes in $\mathcal{T}_h$.

Remark 3.2
We note that in forming the global nodal basis functions associated with a node $(x_N, y_N)^t$, *Step 3* actually replaces those functions $\Psi_{N,i}$ with smaller supports by a function Ψ_{N,c_j} with a larger support provided that $T_{N,i} \in \mathcal{T}_{h,N,c_j}$. This makes it possible to construct $S_h^{Sb}(\Omega)$ of a lower dimension. Also, this step is very similar to the coarsening step in an adaptive mesh refinement procedure.

The selection of coarsenable sets at each node in the *Step 2* is up to the user of this method. A user can use this opportunity to include desirable features in the method. One necessary requirement is to make sure that the global bilinear IFE basis functions associated with each node are linearly independent. We describe two specific rules for selecting coarsenable sets at each node so that this necessary requirement is fulfilled.

Selective rule 1: At each node $(x_N, y_N)^t$ of $\mathcal{T}_h$, we select only the largest coarsenable set that doesn't include any interface element.

For example, in Figure 3, suppose $\square AJNM$ is the only interface element around node *N*, then the coarsenable set to be used *N* is $\{\square JEKN, \square NKIL, \square MNLH\}$ according to **Selective rule 1**.

Copyright © 2013 John Wiley & Sons, Ltd.

Selective rule 2: At each node $(x_N, y_N)^t$ of $\mathcal{T}_h$, if $(x_N, y_N)^t$ is also a node of $\mathcal{T}_h^0$, then we select only the largest coarsenable set which doesn't include any interface element. If $(x_N, y_N)^t$ is not a node of $\mathcal{T}_h^0$, we don't select any coarsenable set.

For example, suppose the $\mathcal{T}_h^0$ consists of elements $\square AEIH$, $\square EBFI$, $\square IFCG$, and $\square HIGD$ in Figure 3, and we refine $\square AEIH$ into four congruent rectangular elements to form the mesh $\mathcal{T}_h$. If $\square AJNM$ is the only interface element, then we don't select any coarsenable set for nodes J, K, L, M, N but select the coarsenable set $\{\square EBFI, \square IFCG, \square HIGD\}$ for node I.

Now, we use the selective bilinear IFE space $S_h^{Sb}(\Omega) \subset PH_S^1(\mathcal{T}_h)$ and the selective DG formulations introduced in the previous section to construct the SIDG method with bilinear IFE as follows: find $u_h \in S_h^{Sb}(\Omega)$ such that

$$a_{S\theta}^-(u_h, v_h) = L_\theta^-(v_h), \quad \forall v_h \in S_h^{Sb}(\Omega) \tag{3.18}$$

or find $u_h \in S_h^{Sb}(\Omega)$ such that

$$a_{S\theta}^+(u_h, v_h) = L_\theta^+(v_h), \quad \forall v_h \in S_h^{Sb}(\Omega). \tag{3.19}$$

3.2. Advantages of the selective immersed discontinuous Galerkin method

In this section, we will first point out a limitation of Galerkin finite element method for using an adaptive Cartesian mesh. Then we will explain how the combination of the SIDG method and bilinear IFE allows us to form an efficient numerical method with adaptive Cartesian mesh for solving interface problems.

In a Galerkin method based on Lagrange type finite elements, the involved finite element space can be described by the global nodal basis functions. These global basis functions are often required to have certain continuity at each node in the mesh. When a global basis function is restricted to an element, it is either the zero function or becomes one of the local basis function in that element. If we refine the mesh, new nodes usually need to be introduced. A new node becomes a so called hanging node if it is in the interior of an edge of an element. At a hanging node, the global basis function of the original type usually cannot be defined. As remarked in [129], the construction of a conforming finite element space on a mesh with hanging nodes is a nontrivial matter. Techniques using continuity constraints to handle hanging nodes in finite element methods based on the conforming Galerkin formulation have been developed in [129, 130]. However, IFEs are nonconforming because of their discontinuity across edges between two interface elements, and this makes the ideas in [129, 130] not directly applicable to hanging nodes in the IFE method with local refinement. On the other hand, the DG formulation can be naturally employed in local refinement without the concern of hanging nodes because it allows interelement discontinuity. This motivates us to combine the DG formulation with IFEs for solving interface problems on Cartesian meshes with local refinement.

For a hanging nodes in a SIDG method, we can find the element in which this hanging node is a vertex and use the corresponding local nodal basis to introduce a global basis at this hanging node. For example, for the hanging node K in Figure 3, we can define a global nodal basis from the local nodal basis at K in element $\square NKIL$ and another global nodal basis from the local nodal basis at K in element $\square JEKN$. Because there is no local nodal basis at K in element $\square EBFI$, no global nodal basis is introduced at K with respect to this element. This is why the SIDG method does not have the limitation of hanging nodes and can allow us to employ the mesh refinement locally at the places needed.

Meanwhile, the IFE feature allows us to use structured meshes for interface problems; hence, the SIDG method based on IFE spaces will allow us to use locally adaptive structured meshes for solving interface problems with nontrivial interfaces. That is, in a structured mesh, we can refine any region again and again while keeping the mesh in the remaining region coarse. For example, we can repeatedly refine the interface elements along the interface locally, see Figure 4, if the solution/physics changes drastically across the interface. This is very important for maintaining the efficiency of mesh refinement.

We would like to note that the local refinement capability is a common feature for all DG methods. Therefore, the usual IPDG method with IFEs also has this advantage. However, as mentioned in the introduction section, the traditional IPDG formulation increases the computational cost by a significant amount compared with the standard Galerkin formulation. In the following, we will explain how the selective feature in the SIDG method can help to keep the computational cost low while retaining the local mesh refinement with structured mesh.

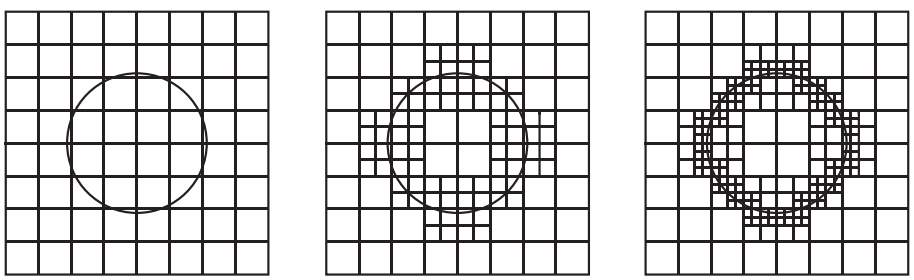

Figure 4. The left plot is a usual rectangular mesh without refinement. In the middle plot, each interface element in the left mesh is refined into four congruent rectangular elements. In the right plot, each interface element in the middle graph is refined into four congruent rectangular elements.

Copyright © 2013 John Wiley & Sons, Ltd.

Table I. Comparison of the number of global basis functions used by the Galerkin method, the interior penalty discontinuous Galerkin (IPDG) method, and the selective immersed discontinuous Galerkin (SIDG) method.

h	N_G	N_{IPDG}	N_{SIDG}
1/16	961	4092	1357
1/32	3721	16 380	4749
1/64	15 625	65 532	17 681
1/128	64 009	262 140	68 113
1/256	259 081	1 048 572	267 281

Assuming we use the SIDG method to solve an interface problem with a rectangular mesh. The SIDG method will use only one global nodal basis function at each node outside of $\overline{\Omega_S}$, which is the same as that used in the IFE method based on the usual Galerkin formulation. The SIDG method only increases the number of global nodal basis functions at those nodes inside $\overline{\Omega_S}$. If the nodes inside $\overline{\Omega_S}$ are much less than those outside of $\overline{\Omega_S}$, then the SIDG method doesn't increase the number of global nodal basis functions significantly compared with the method based on the Galerkin formulation. To be specific, let us consider a rectangular mesh $\mathcal{T}_h$ on Ω with $N \times N$ nodes, and assume that L of its nodes are in $\overline{\Omega_S}$. On this mesh, the Galerkin IFE method will use $(N-2)^2$ global nodal basis functions; the IPDG method with bilinear IFEs needs $4N^2 - 8N$ global nodal basis functions. However, the SIDG method will use at most $(N-2)^2 + 3L$ global nodal basis functions. Assume that the SIDG method uses the DG formulation only in interface elements, and without loss of generality, we can assume $L = CN$ for some constant C. In this case, the number of global nodal basis functions for the SIDG method is at most $(N-2)^2 + 3CN$, which is much closer to $(N-2)^2$ than $4N^2 - 8N$ when N is large.

We can also observe the efficiency of the SIDG method in actual computations. Tables 5.1, 7.1, and 7.6 in [128], Table II and Table X in Section 5 present the solution errors of Galerkin IFE method, the IPDG IFE method, and the SIDG method for the same example described in Section 5 with $\beta^- = 1$ and $\beta^+ = 10$. From these tables, we can see that all of these methods achieve comparable accuracy on the same mesh. To be specific, we have used the **Selective rule 2** in Section 3.1 for the SIDG method.

However, the dimensions of these three methods are not at the same level. For example, let us consider a domain $\Omega = [-1,1]\times[-1,1]$ with the interface curve $\Gamma = \{(x,y) : x^2 + y^2 = (\pi/6.28)^2\}$. Note that a rectangular Cartesian mesh with a step size h has $(\frac{2}{h}+1)^2$ nodes. Let N_G, N_{IPDG}, and N_{SIDG} denote the number of global basis functions used by Galerkin method, the IPDG method, and the SIDG method separately. Then, we get the following table:

The datum in Table I shows that $\frac{N_G}{N_{SIDG}}$ becomes closer and closer to 1 when the mesh size h becomes smaller, but N_{IPDG} stays around four times of N_G. This means that to solve an interface problem by the IPDG IFE method with a comparable accuracy, we need to solve a linear system whose number of unknowns is about four times that of the SIDG method. Therefore, the selective feature can be used to effectively reduce the computational cost in the SIDG method without the loss of its local refinement capability on structured meshes. Also, we note that the SIDG method needs less computations in assembling its algebraic system because it only computes the jump terms for the edges in $\mathcal{E}_S$ instead of $\mathcal{E}_h$ in the IPDG IFE method. Even though the Galerkin method uses less number of global basis functions than the SIDG method, it does have a much more stringent requirement on mesh refinement.

4. Approximation capability of the selective bilinear immersed finite elements space

In this section, we will analyze the approximation capability of the selective bilinear IFE space by estimating its interpolation error. This is also an important preparation for the convergence analysis of the selective DG methods, which is an interesting future work. We use C to represent a generic constant whose value might be different from line to line. Unless otherwise specified, all the generic constants C in the presentation in the succeeding text are independent of the interface and mesh. Define the following space

$$PC^m_{\text{int}}(\Lambda) = \left\{ u \in C(\Lambda), u|_{\Lambda^s} \in C^m(\Lambda^s), s = -,+, \left[\beta \frac{\partial u}{\partial \mathbf{n}_\Gamma}\right]_{\Gamma\cap\Lambda} = 0 \right\}, \ m \geq 1,$$

where $\Lambda \subset \Omega$ whose interior is cut through by Γ. Consider a mesh $\mathcal{T}_h$ with step size h. In the discussion in the succeeding text, we need three assumptions on the interface curve Γ and the mesh $\mathcal{T}_h$.

- (H_1): The family of meshes $\mathcal{T}_h$ with $h > 0$ is regular.
- (H_2): The interface curve Γ is defined by a piecewise C^2 function, and the mesh $\mathcal{T}_h$ is formed such that the subset of Γ in any interface element is C^2.
- (H_3): The interface Γ is smooth enough so that $PC^3_{\text{int}}(T)$ is dense in $PH^2_{\text{int}}(T)$ for any interface element T of $\mathcal{T}_h$.

Copyright © 2013 John Wiley & Sons, Ltd.

To describe the errors in our analysis in the succeeding text, we need to use the following slightly modified definitions for the norms involving functions of $S_h(T)$ for interface elements $T \in \mathcal{T}_h$. First, for each $w_h \in S_h(T)$, we define

$$\|w_h\|_{s,T}^2 = \|w_h\|_{s,\tilde{T}^+}^2 + \|w_h\|_{s,\tilde{T}^-}^2,\ s = 0,1,2,$$
$$|w_h|_{s,T}^2 = |w_h|_{s,\tilde{T}^+}^2 + |w_h|_{s,\tilde{T}^-}^2,\ s = 0,1,2.$$

For each $w_h \in S_h(T)$ and $u \in PH_{\text{int}}^2(T)$, $w_h \pm u$ is also a piecewise H^2 function on T, and we define

$$\|w_h \pm u\|_{s,T}^2 = \|w_h \pm u\|_{s,\tilde{T}^+\cap T^+}^2 + \|w_h \pm u\|_{s,\tilde{T}^+\cap T^-}^2 + \|w_h \pm u\|_{s,\tilde{T}^-\cap T^+}^2 + \|w_h \pm u\|_{s,\tilde{T}^-\cap T^-}^2,\ s = 0,1,2,$$
$$|w_h \pm u|_{s,T}^2 = |w_h \pm u|_{s,\tilde{T}^+\cap T^+}^2 + |w_h \pm u|_{s,\tilde{T}^+\cap T^-}^2 + |w_h \pm u|_{s,\tilde{T}^-\cap T^+}^2 + |w_h \pm u|_{s,\tilde{T}^-\cap T^-}^2,\ s = 0,1,2.$$

Then, for each $w_h \in S_h^{Sb}(\Omega)$ and $u \in PH_{\text{int}}^2(\Omega)$, we define

$$\|u\|_{s,\mathcal{T}_h}^2 = \sum_{T\in\mathcal{T}_h} \|u\|_{s,T}^2,\ s = 0,1,2,$$
$$|u|_{s,\mathcal{T}_h}^2 = \sum_{T\in\mathcal{T}_h} |u|_{s,T}^2,\ s = 0,1,2,$$
$$\|w_h \pm u\|_{s,\mathcal{T}_h}^2 = \sum_{T\in\mathcal{T}_h} \|w_h \pm u\|_{s,T}^2,\ s = 0,1,2,$$
$$|w_h \pm u|_{s,\mathcal{T}_h}^2 = \sum_{T\in\mathcal{T}_h} |w_h \pm u|_{s,T}^2,\ s = 0,1,2.$$

Now, we define the bilinear IFE interpolation. For a function $u \in PH_{\text{int}}^2(T)$, $T \in \mathcal{T}_h$, we let $I_{h,T}u \in S_h(T)$ be its interpolant such that $I_{h,T}u(X) = u(X)$ when X is a vertex of T. In general, for an element T with vertices A_1, A_2, A_3, A_4, we have

$$I_{h,T}u(X) = u(A_1)\phi_1(X) + u(A_2)\phi_2(X) + u(A_3)\phi_3(X) + u(A_4)\phi_4(X).$$

Accordingly, for a function $u \in PH_{\text{int}}^2(\Omega)$, we let $I_h u \in S_h^{Sb}(\Omega)$ be its interpolant such that $I_h u|_T = I_{h,T}(u|_T)$ for any $T \in \mathcal{T}_h$. For more details about the bilinear IFE interpolation, we refer readers to [109, 128]. Now, we recall the following theorem from [109, 128].

Theorem 4.1
There exists a constant C independent of the interfaces such that

$$\|I_h u - u\|_{0,\mathcal{T}_h} \le Ch^2 \|u\|_{2,\mathcal{T}_h}, \tag{4.20}$$

$$|I_h u - u|_{1,\mathcal{T}_h} \le Ch \|u\|_{2,\mathcal{T}_h}. \tag{4.21}$$

for any $u \in PH_{\text{int}}^2(\Omega)$ and $h > 0$ small enough.

Now, we estimate the interpolation error in the H^2 seminorm for bilinear IFE interpolation. Because the arguments used in this estimation are rather similar to that of [109, 128], we only show the major steps of the proof here. We will also recall and directly use some conclusions in [109, 128]. We would like to note that the two pieces of local IFE functions in [109] are defined on T^- and T^+, but the two pieces of local IFE functions in this article and [128] are defined on $\tilde{T}^-$ and $\tilde{T}^+$. Even though this difference causes a slight difference in the proof, the arguments in [109] can still go through for the IFE functions defined in this article and [128]. Instead of discussing only T^- and T^+ as in [109], we need to discuss $\tilde{T}^- \cap T^-$, $\tilde{T}^- \cap T^+$, $\tilde{T}^+ \cap T^-$, and $\tilde{T}^+ \cap T^+$ with the same arguments as used in this article and in [128].

We call a point $X = (x, y)^T$ in an interface element T an *obscure point* if one of the four line segments connecting X and the vertices of T intersects the interface more than once. Without loss of generality, we discuss interface elements that do not contain any obscure point because the arguments used in the succeeding text can be readily extended to handle the interface elements with obscure points.

Without loss of generality, we assume $T \in \mathcal{T}_h$ is a Type I interface element(see Figure 2) with vertices $A_i = (x_i, y_i)$, $i = 1,2,3,4$, such that $A_1 \in T^+$ and $A_i \in T^-$, $i = 2,3,4$, see Figure 5. We start with the estimation on $\tilde{T}^- \cap T^-$. Consider a point $X = (x, y) \in \tilde{T}^- \cap T^-$ and assume that line segments $\overline{XA_i}$, $i = 2,3,4$ do not intersect with the interface and $\overline{DE}$, whereas line segment $\overline{XA_1}$ meets Γ at $\widetilde{A}_1$ (see Figure 5) with

$$\widetilde{A}_1 = \tilde{t}A_1 + (1-\tilde{t})X = (\widetilde{x}_1, \widetilde{y}_1)$$

Copyright © 2013 John Wiley & Sons, Ltd.

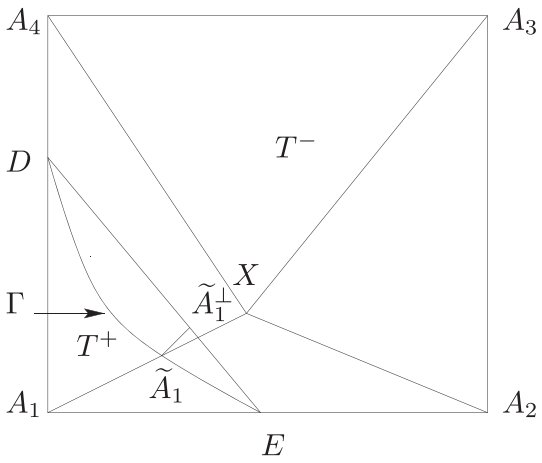

Figure 5. An interface rectangle element with no obscure point. A point $X \in \tilde{T}^- \cap T^-$ is connected to the four vertices by line segments in a Type I interface element.

for a certain $\tilde{t}$. For any point $\widetilde{A} \in \Gamma$, let $\widetilde{A}^{\perp}$ be the orthogonal projection of $\tilde{A}$ onto $\overline{DE}$ (see Figure 5). Let $\rho = \frac{\beta^-}{\beta^+}$, $\mathbf{n}(\widetilde{A}) = (n_x(\widetilde{A}), n_y(\widetilde{A}))$ be the unit normal vector of Γ at $\widetilde{A}$, $\mathbf{n}(\overline{DE}) = (\bar{n}_x, \bar{n}_y)$ be the unit normal vector of $\overline{DE}$, $X_{\overline{DE}} = (\bar{x}, \bar{y})$ be a point on $\overline{DE}$, $X^s_{\overline{DE}} = (\bar{y}, \bar{x})$ and $X^s = (y, x)$. For every function $u(x, y)$ satisfying the interface jump conditions (1.3) and (1.4), we define

$$N^-(\widetilde{A}) = \begin{pmatrix} n_y(\widetilde{A})^2 + \rho n_x(\widetilde{A})^2 & (\rho-1)n_x(\widetilde{A})n_y(\widetilde{A}) \\ (\rho-1)n_x(\widetilde{A})n_y(\widetilde{A}) & n_x(\widetilde{A})^2 + \rho n_y(\widetilde{A})^2 \end{pmatrix},$$

$$N^-_{\overline{DE}} = \begin{pmatrix} \bar{n}_y^2 + \rho \bar{n}_x^2 & (\rho-1)\bar{n}_x\bar{n}_y \\ (\rho-1)\bar{n}_x\bar{n}_y & \bar{n}_x^2 + \rho \bar{n}_y^2 \end{pmatrix}.$$

In the discussion in the succeeding text, we let I_i, $i = 1, 2, 3, 4$ be the integral terms involving the vertices A_i, $i = 1, 2, 3, 4$ of Theorem 3.1 in [109]. With the same arguments as those used for Theorems 3.1 and 3.3 in [109], we can prove the following lemma.

Lemma 4.1
For any $u \in PC^3_{\text{int}}(T)$ and $X = (x, y) \in \tilde{T}^- \cap T^-$, we have

$$\begin{aligned} \frac{\partial^2 \left(I_{h,T}u(X) - u(X)\right)}{\partial x \partial y} &= (N^-\left(\widetilde{A}_1\right) - N^-_{\overline{DE}})\nabla u(X)\cdot(A_1 - \widetilde{A}_1)\frac{\partial^2\phi_1(X)}{\partial x\partial y} - (N^-_{\overline{DE}} - I)\nabla u(X)\cdot(\widetilde{A}_1 - X_{\overline{DE}})\frac{\partial^2\phi_1(X)}{\partial x\partial y} \\ &\quad - \frac{\partial^2 u^-(X)}{\partial x\partial y}\left[N^-_{\overline{DE}}\left(X^s_{\overline{DE}} - X^s\right)\cdot(A_1 - X_{\overline{DE}})\frac{\partial^2\phi_1(X)}{\partial x\partial y} + N^-_{\overline{DE}}(-1,0)\cdot(A_1 - X_{\overline{DE}})\frac{\partial\phi_1(X)}{\partial x}\right. \\ &\quad \left. + N^-_{\overline{DE}}(0,-1)\cdot(A_1 - X_{\overline{DE}})\frac{\partial\phi_1(X)}{\partial y} + (x_1 - \bar{x})(y_1 - \bar{y})\frac{\partial^2\phi_1(X)}{\partial x\partial y} + 2\phi_1(X) - (\bar{y} - y)\frac{\partial\phi_1(X)}{\partial y}\right. \\ &\quad - (\bar{x} - x)\frac{\partial\phi_1(X)}{\partial x} + (\bar{x} - x)(\bar{y} - y)\frac{\partial^2\phi_1(X)}{\partial x\partial y} + \sum_{i=2}^{4}\left[2\phi_i(X) - (y_i - y)\frac{\partial\phi_i(X)}{\partial y}\right. \\ &\quad \left.\left. - (x_i - x)\frac{\partial\phi_i(X)}{\partial x} + (x_i - x)(y_i - y)\frac{\partial^2\phi_i(X)}{\partial x\partial y}\right]\right] + \sum_{i=1}^{4} I_i\frac{\partial^2\phi_i(X)}{\partial x\partial y}. \end{aligned} \tag{4.22}$$

Copyright © 2013 John Wiley & Sons, Ltd.

Proof

From Theorem 3.3 in [109], we can get

$$\begin{aligned}\frac{\partial^2(I_{h,T}u(X)-u(X))}{\partial x\partial y} &= \frac{\partial}{\partial y}\left[\left(N^-\left(\widetilde{A}_1\right)-N^-_{\overline{DE}}\right)\nabla u(X)\cdot\left(A_1-\widetilde{A}_1\right)\right]\frac{\partial\phi_1(X)}{\partial x}+\left(N^-\left(\widetilde{A}_1\right)-N^-_{\overline{DE}}\right)\nabla u(X)\cdot\left(A_1-\widetilde{A}_1\right)\frac{\partial^2\phi_1(X)}{\partial x\partial y}\\ &-\frac{\partial}{\partial y}\left[\left(N^-_{\overline{DE}}-I\right)\nabla u(X)\cdot\left(\widetilde{A}_1-X_{\overline{DE}}\right)\right]\frac{\partial\phi_1(X)}{\partial x}-\left(N^-_{\overline{DE}}-I\right)\nabla u(X)\cdot\left(\widetilde{A}_1-X_{\overline{DE}}\right)\frac{\partial^2\phi_1(X)}{\partial x\partial y}\\ &-\frac{\partial^3u^-(X)}{\partial x\partial y^2}\left[N^-_{\overline{DE}}\left(X^s_{\overline{DE}}-X^s\right)\cdot\left(A_1-X_{\overline{DE}}\right)\frac{\partial\phi_1(X)}{\partial x}+N^-_{\overline{DE}}(0,-1)\cdot\left(A_1-X_{\overline{DE}}\right)\phi_1(X)\right.\\ &+(x_1-\bar{x})(y_1-\bar{y})\frac{\partial\phi_1(X)}{\partial x}-(\bar{y}-y)\phi_1(X)+(\bar{x}-x)(\bar{y}-y)\frac{\partial\phi_1(X)}{\partial x}+\sum_{i=2}^{4}\left[-(y_i-y)\phi_i(X)\right.\\ &\left.\left.+(x_i-x)(y_i-y)\frac{\partial\phi_i(X)}{\partial x}\right]\right]-\frac{\partial^2u^-(X)}{\partial x\partial y}\left[N^-_{\overline{DE}}\left(X^s_{\overline{DE}}-X^s\right)\cdot\left(A_1-X_{\overline{DE}}\right)\frac{\partial^2\phi_1(X)}{\partial x\partial y}\right.\\ &+N^-_{\overline{DE}}(-1,0)\cdot\left(A_1-X_{\overline{DE}}\right)\frac{\partial\phi_1(X)}{\partial x}+N^-_{\overline{DE}}(0,-1)\cdot\left(A_1-X_{\overline{DE}}\right)\frac{\partial\phi_1(X)}{\partial y}\\ &+(x_1-\bar{x})(y_1-\bar{y})\frac{\partial^2\phi_1(X)}{\partial x\partial y}+\phi_1(X)-(\bar{y}-y)\frac{\partial\phi_1(X)}{\partial y}-(\bar{x}-x)\frac{\partial\phi_1(X)}{\partial x}+(\bar{x}-x)(\bar{y}-y)\frac{\partial^2\phi_1(X)}{\partial x\partial y}\\ &+\sum_{i=2}^{4}\left[\phi_i(X)-(y_i-y)\frac{\partial\phi_i(X)}{\partial y}-(x_i-x)\frac{\partial\phi_i(X)}{\partial x}+(x_i-x)(y_i-y)\frac{\partial^2\phi_i(X)}{\partial x\partial y}\right]\Bigg]\\ &+\sum_{i=1}^{4}\frac{\partial I_i}{\partial y}\frac{\partial\phi_i(X)}{\partial x}+\sum_{i=1}^{4}I_i\frac{\partial^2\phi_i(X)}{\partial x\partial y}.\end{aligned}\tag{4.23}$$

Taking the first derivative with respect to y on both sides of (3.27) and (3.28) in [109], we can get

$$\begin{aligned}\frac{\partial I_i}{\partial y} &= -P\cdot(A_i-X),\ i=2,3,4 \text{ where } P=\frac{\partial}{\partial y}\nabla u(X),\\ \frac{\partial I_1}{\partial y} &= -P\cdot(A_1-X)-\frac{\partial}{\partial y}\left[(N^-\left(\widetilde{A}_1\right)-I)\nabla u(X)\left(A_1-\widetilde{A}_1\right)\right].\end{aligned}$$

Hence,

$$\sum_{i=1}^{4}\frac{\partial I_i}{\partial y}\frac{\partial\phi_i(X)}{\partial x}=-P\cdot\sum_{i=1}^{4}(A_i-X)\frac{\partial\phi_i(X)}{\partial x}-\frac{\partial}{\partial y}\left[\left(N^-\left(\widetilde{A}_1\right)-I\right)\nabla u(X)\left(A_1-\widetilde{A}_1\right)\right]\frac{\partial\phi_1(X)}{\partial x}.\tag{4.24}$$

Taking the derivative for x on both sides of the conclusion of Lemma 3.6 in [109], we get

$$\begin{aligned}\mathbf{q}\cdot\sum_{i=1}^{4}(A_i-X)\frac{\partial\phi_i(X)}{\partial x} &= -\mathbf{q}\cdot\sum_{i=1}^{4}(-1,0)\phi_i(X)-\left(N^-_{\overline{DE}}-I\right)\mathbf{q}\cdot\left(A_1-\widetilde{A}_1\right)\frac{\partial\phi_1(X)}{\partial x}-\left(N^-_{\overline{DE}}-I\right)\mathbf{q}\cdot\left(\widetilde{A}_1-X_{\overline{DE}}\right)\frac{\partial\phi_1(X)}{\partial x}\\ &-d^-\left[N^-_{\overline{DE}}\left(X^s_{\overline{DE}}-X^s\right)\cdot\left(A_1-X_{\overline{DE}}\right)\frac{\partial\phi_1(X)}{\partial x}+N^-_{\overline{DE}}(0,-1)\cdot\left(A_1-X_{\overline{DE}}\right)\phi_1(X)\right.\\ &+(x_1-\bar{x})(y_1-\bar{y})\frac{\partial\phi_1(X)}{\partial x}-(\bar{y}-y)\phi_1(X)+(\bar{x}-x)(\bar{y}-y)\frac{\partial\phi_1(X)}{\partial x}\\ &\left.+\sum_{i=2}^{4}\left[-(y_i-y)\phi_i(X)+(x_i-x)(y_i-y)\frac{\partial\phi_i(X)}{\partial x}\right]\right],\end{aligned}\tag{4.25}$$

where $\mathbf{q}$ is any given two-dimensional vector and d^- is any given real number.

Copyright © 2013 John Wiley & Sons, Ltd.

Letting $\mathbf{q} = P, d^- = \frac{\partial^3 u^-(X)}{\partial x \partial y^2}$ in (4.25) and applying it to the first term on the right hand side of (4.24), we have

$$\begin{aligned}\sum_{i=1}^{4} \frac{\partial I_i}{\partial y}\frac{\partial \phi_i(X)}{\partial x} = &-\frac{\partial^2 u^-(X)}{\partial x \partial y}\sum_{i=1}^{4}\phi_i(X) + \left(N_{\overline{DE}}^- - I\right) P \cdot \left(A_1 - \widetilde{A}_1\right)\frac{\partial \phi_1(X)}{\partial x} + \left(N_{\overline{DE}}^- - I\right) P \cdot \left(\widetilde{A}_1 - X_{\overline{DE}}\right)\frac{\partial \phi_1(X)}{\partial x} \\ &+ \frac{\partial^3 u^-(X)}{\partial x \partial y^2}\left[N_{\overline{DE}}^-\left(X_{\overline{DE}}^s - X^s\right)\cdot\left(A_1 - X_{\overline{DE}}\right)\frac{\partial \phi_1(X)}{\partial x} + N_{\overline{DE}}^-(0,-1)\cdot\left(A_1 - X_{\overline{DE}}\right)\phi_1(X)\right. \\ &+ (x_1 - \bar{x})(y_1 - \bar{y})\frac{\partial \phi_1(X)}{\partial x} - (\bar{y} - y)\phi_1(X) + (\bar{x} - x)(\bar{y} - y)\frac{\partial \phi_1(X)}{\partial x} \\ &\left.+ \sum_{i=2}^{4}\left[-(y_i - y)\phi_i(X) + (x_i - x)(y_i - y)\frac{\partial \phi_i(X)}{\partial x}\right]\right] \\ &- \frac{\partial}{\partial y}\left[\left(N^-\left(\widetilde{A}_1\right) - I\right)\nabla u(X)\left(A_1 - \widetilde{A}_1\right)\right]\frac{\partial \phi_1(X)}{\partial x}.\end{aligned} \tag{4.26}$$

By direct calculations, we also have

$$\begin{aligned}&\frac{\partial}{\partial y}\left(N^-\left(\widetilde{A}_1\right) - N_{\overline{DE}}^-\right)\nabla u(X)\cdot\left(A_1 - \widetilde{A}_1\right)\frac{\partial \phi_1(X)}{\partial x} - \frac{\partial}{\partial y}\left[\left(N_{\overline{DE}}^- - I\right)\nabla u(X)\cdot\left(\widetilde{A}_1 - X_{\overline{DE}}\right)\right]\frac{\partial \phi_1(X)}{\partial x} \\ &- \frac{\partial}{\partial y}\left[\left(N^-\left(\widetilde{A}_1\right) - I\right)\nabla u(X)\left(A_1 - \widetilde{A}_1\right)\right]\frac{\partial \phi_1(X)}{\partial x} + \left(N_{\overline{DE}}^- - I\right) P \cdot \left(A_1 - \widetilde{A}_1\right)\frac{\partial \phi_1(X)}{\partial x} \\ &+ \left(N_{\overline{DE}}^- - I\right) P \cdot \left(\widetilde{A}_1 - X_{\overline{DE}}\right)\frac{\partial \phi_1(X)}{\partial x} = 0.\end{aligned} \tag{4.27}$$

Plugging (4.26) and (4.27) into (4.23), we finish the proof of (4.22). □

Lemma 4.2
There exists a constant C such that

$$\left\|I_{h,T}u - u\right\|_{2,\tilde{T}^- \cap T^-} \le C\left\|u\right\|_{2,T} \tag{4.28}$$

for any $u \in PH^2_{\text{int}}(T)$ where T is a Type I interface element.

Proof
Because of assumption (H_3), we only need to show that (4.28) is true for any $u \in PC^3_{\text{int}}(T)$. Assume that the interface element T is related to the reference element $\hat{T} = [0,1]\times[0,1]$ by the usual affine mapping:

$$X = F(\hat{X}) = B + M\hat{X},\ X = \begin{pmatrix} x \\ y \end{pmatrix},\ \hat{X} = \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix}. \tag{4.29}$$

Then, define $\hat{\phi}_i(\hat{X})$ to be a nodal bilinear IFE basis function in the reference element $\hat{T}$, which is corresponding to the local nodal bilinear IFE basis function $\phi_i(X)$. Assume

$$M^{-1} = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix}.$$

Then,

$$\frac{\partial^2 \phi_i(X)}{\partial x \partial y} = \frac{\partial^2 \hat{\phi}_i(\hat{X})}{\partial \hat{x}\partial \hat{y}}(m_{12}m_{21} + m_{11}m_{22}).$$

Because the mesh is regular, we have $\|M\| \le Ch^{-1}$, then $|m_{ij}| \le Ch^{-1}$; hence, $|m_{12}m_{21} + m_{11}m_{22}| \le Ch^{-2}$. Note that $\frac{\partial^2 \hat{\phi}_i(\hat{X})}{\partial \hat{x}\partial \hat{y}}$ is a bounded constant, then

$$\left|\frac{\partial^2 \phi_i(X)}{\partial x \partial y}\right| \le Ch^{-2}.$$

Because $\frac{\partial^2 (I_{h,T}u)}{\partial x^2} = \frac{\partial^2 (I_{h,T}u)}{\partial y^2} = 0$, we complete the proof by applying the same arguments of Theorem 3.2 and Theorem 3.4 in [109] to Lemma 4.1. □

We can get similar conclusions for $\tilde{T}^- \cap T^+$, $\tilde{T}^+ \cap T^-$, and $\tilde{T}^+ \cap T^+$ of the Type I interface elements. The conclusions on each Type II interface element are also similar. Finally, with the same idea as Theorem 3.14 in [109], we get the following theorem.

Theorem 4.2
$\forall u \in PH^2_{int}(\Omega)$, there exists a constant C independent of the interfaces such that

$$|I_h u - u|_{2,\mathcal{T}_h} \le C\|u\|_{2,\mathcal{T}_h}. \tag{4.30}$$

Copyright © 2013 John Wiley & Sons, Ltd.

5. Numerical examples

In this section, we will provide numerical examples to demonstrate features of the SIDG method with bilinear IFEs. Because large errors usually appear in interface elements, we will try to reduce the error by refining interface elements as follows. With a chosen mesh size h, we start from a rectangular Cartesian mesh $\mathcal{T}_h^0$ without any hanging nodes, see the left plot of Figure 4. Then, we divide each interface element of $\mathcal{T}_h^0$ into four congruent rectangular elements to obtain the first-level refined mesh $\mathcal{T}_h^1$, see the middle plot of Figure 4. We continue to refine the mesh $\mathcal{T}_h^1$ by dividing each interface element of $\mathcal{T}_h^1$ into four congruent rectangular elements to obtain the second-level refined mesh $\mathcal{T}_h^2$, see the right plot of Figure 4. Repeating this process, we can obtain the n^{th} level refined mesh $\mathcal{T}_h = \mathcal{T}_h^n$ to be used for solving the interface problem. How many levels of refinements to be used in $\mathcal{T}_h$ depends on the refinement rule chosen in the computation. All the numerical results in this section are generated by the SIDG method with **Selecting rule 2** described in Section 3.1. In this case, Ω_S is actually the union of all interface elements in $\mathcal{T}_h^0$, and $\mathcal{E}_S$ is the collection of all element edges of $\mathcal{T}_h$ that are in $\overline{\Omega_S}$.

To generate the numerical results, we consider the interface problem defined by (1.1)–(1.4) on $\Omega = [-1,1]\times[-1,1]$. The interface curve Γ is a circle with radius $r_0 = \pi/6.28$ which separates Ω into two subdomains Ω^- and Ω^+ with $\Omega^- = \{(x,y) \mid x^2+y^2 \le r_0^2\}$. The coefficient function is

$$\beta(x,y) = \begin{cases} \beta^- = 1, & (x,y)\in\Omega^-, \\ \beta^+, & (x,y)\in\Omega^+, \end{cases}$$

with β^+ to be chosen to reflect different jump ratios in the coefficient. The boundary condition function $g(x,y)$, and the source term $f(x,y)$ are chosen such that the following function u is the exact solution.

$$u(x,y) = \begin{cases} \frac{r^\alpha}{\beta^-}, & \text{if } r \le r_0, \\ \frac{r^\alpha}{\beta^+} + \left(\frac{1}{\beta^-} - \frac{1}{\beta^+}\right) r_0^\alpha, & \text{otherwise,} \end{cases}$$

with $\alpha = 3, r = \sqrt{x^2+y^2}$. We will present numerical results for two cases with $\beta^+ = 10$ and $\beta^+ = 1\,000\,000$, which represent a moderate and large jumps in the coefficient, respectively.

5.1. *Numerical results for the symmetric selective immersed discontinuous Galerkin method without local mesh refinement*

In this section, we will consider the convergence of the symmetric SIDG solutions with bilinear IFEs in the L^2 and H^1 norms on rectangular meshes $\mathcal{T}_h = \mathcal{T}_h^0$ without local refinement. Table II contains the errors of the solutions u_h with various mesh sizes h, $\beta^+ = 10$ and $C_* = 1000$. Table III and IV contain the errors of the solutions u_h with various mesh sizes h, $\beta^+ = 1\,000\,000$ and $C_* = 1000, 0.0001$, respectively.

Using linear regression, we can see that the data in Table II obey

$$\|u_h - u\|_0 \approx 0.2309\, h^{1.9933}, \quad |u_h - u|_1 \approx 0.8318\, h^{1.0431},$$

Table II. Errors of the symmetric selective immersed discontinuous Galerkin (SIDG) solutions without mesh refinement for $\beta^+ = 10$ and C_*=1000.

h	$\|u_h - u\|_0$	$\|u_h - u\|_1$
1/16	9.3598×10^{-4}	4.6983×10^{-2}
1/32	2.2376×10^{-4}	2.2336×10^{-2}
1/64	5.8624×10^{-5}	1.0568×10^{-2}
1/128	1.4462×10^{-5}	5.2162×10^{-3}
1/256	3.6792×10^{-6}	2.6168×10^{-3}

Table III. Errors of the symmetric selective immersed discontinuous Galerkin (SIDG) solutions without mesh refinement for $\beta^+ = 1\,000\,000$ and C_*=1000.

h	$\|u_h - u\|_0$	$\|u_h - u\|_1$
1/16	9.5450×10^{-4}	4.5462×10^{-2}
1/32	2.4804×10^{-4}	2.1547×10^{-2}
1/64	5.9188×10^{-5}	9.5477×10^{-3}
1/128	1.4644×10^{-5}	4.6827×10^{-3}
1/256	3.8419×10^{-6}	2.3697×10^{-3}

Copyright © 2013 John Wiley & Sons, Ltd.

Table IV. Errors of the symmetric selective immersed discontinuous Galerkin (SIDG) solutions without mesh refinement for $\beta^+ = 1\,000\,000$ and C_*=0.0001.

h	$\|u_h - u\|_0$	$\|u_h - u\|_1$
1/16	7.0060×10^{-3}	2.2073×10^{0}
1/32	1.0443×10^{-2}	3.0078×10^{0}
1/64	3.0155×10^{-4}	3.6226×10^{-1}
1/128	2.3456×10^{-3}	6.1921×10^{-1}
1/256	5.4700×10^{-5}	2.8866×10^{-1}

Table V. L^2 norm errors of the symmetric selective immersed discontinuous Galerkin (SIDG) solutions on different meshes for $\beta^+ = 10$ and C_*=100 000.

h	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$
Original mesh	2.1366×10^{-2}	4.7971×10^{-3}	7.2254×10^{-4}
First-level refined mesh	4.3277×10^{-3}	8.2111×10^{-4}	1.7169×10^{-4}
Second-level refined mesh	2.2492×10^{-3}	5.9412×10^{-4}	1.5710×10^{-4}
Third-level refined mesh	2.0740×10^{-3}	5.8111×10^{-4}	1.5364×10^{-4}
Fourth-level refined mesh	2.0634×10^{-3}	5.7744×10^{-4}	1.5281×10^{-4}

Table VI. H^1 norm errors of the symmetric selective immersed discontinuous Galerkin (SIDG) solutions on different meshes for $\beta^+ = 10$ and C_*=100 000.

h	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$
Original mesh	1.7413×10^{-1}	7.2458×10^{-2}	2.7376×10^{-2}
First-level refined mesh	7.8639×10^{-2}	3.9678×10^{-2}	1.9964×10^{-2}
Second-level refined mesh	6.6543×10^{-2}	3.7186×10^{-2}	1.9708×10^{-2}
Third-level refined mesh	6.5186×10^{-2}	3.7008×10^{-2}	1.9664×10^{-2}
Fourth-level refined mesh	6.5084×10^{-2}	3.6968×10^{-2}	1.9656×10^{-2}

and the data in Table III obey

$$\|u_h - u\|_0 \approx 0.2460\, h^{1.9996}, \quad |u_h - u|_1 \approx 0.8720\, h^{1.0726}.$$

These data indicate that, with a large penalty constant C_*, the symmetric SIDG solutions converge to the exact solution with convergence rates $O(h^2)$ in the L^2 norm and $O(h)$ in the H^1 norm.

The analysis for the IPDG method suggests that the symmetric SIDG requires a large enough penalty constant C_* to perform satisfactorily. The issue here is that there is no specific guide for the choice of the penalty parameter C_*. Furthermore, in using the symmetric SIDG method, an arbitrarily chosen value for C_* may hinder its performance, and the oscillating errors in Table IV corroborates this numerically.

5.2. Numerical results for the symmetric selective immersed discontinuous Galerkin method with local mesh refinement

For the symmetric SIDG method with bilinear IFE, we now observe the effect of local mesh refinement by comparing the numerical errors in the L^2, H^1, and discrete infinity norms on meshes with different refinement levels using the step sizes $h = \frac{1}{8}, \frac{1}{16}, and \frac{1}{32}$. Tables V to VII contain numerical results generated on meshes from the 0th level to the fourth level. Table V is for the L^2 norm errors, Table VI contains the H^1 norm errors, and Table VII contains the discrete infinity norm errors. Note that h is the mesh size for $\mathcal{T}_h^0$. From these tables, we can see that the first-level and second-level mesh refinements dramatically reduce the global errors. Table VIII shows the discrete infinity norm errors in noninterface elements and interface elements separately for the third-level and the fourth-level mesh refinements. Because the error in the noninterface area becomes more and more dominant during the repeated local refinements in interface elements, the third-level and fourth-level mesh refinements do not reduce the global error much any more.

We can also observe the effect of local mesh refinement from the error on the interface elements. Table IX contains the discrete infinity norm errors of the solutions u_h on the interface elements for meshes with the step sizes $h = \frac{1}{8}, \frac{1}{16}, \frac{1}{32}$. From this table, we can see that when interface elements are refined, the error around the interface decreases quickly. Therefore, the SIDG method can efficiently control the error across the interface where interesting physics happens in many applications.

Copyright © 2013 John Wiley & Sons, Ltd.

Table VII. Discrete infinity norm errors of the symmetric selective immersed discontinuous Galerkin (SIDG) solutions on different meshes for $\beta^+ = 10$ and C_*=100 000.

h	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$
Original mesh	3.0810×10^{-2}	8.0050×10^{-3}	3.2539×10^{-3}
First-level refined mesh	8.3872×10^{-3}	1.9109×10^{-3}	4.6486×10^{-4}
Second-level refined mesh	5.0352×10^{-3}	1.4862×10^{-3}	4.0208×10^{-4}
Third-level refined mesh	4.6433×10^{-3}	1.4302×10^{-3}	3.7831×10^{-4}
Fourth-level refined mesh	4.6150×10^{-3}	1.4105×10^{-3}	3.7724×10^{-4}

Table VIII. Discrete infinity norm errors of the symmetric selective immersed discontinuous Galerkin (SIDG) solutions on interface elements and noninterface elements for $\beta^+ = 10$ and C_*=100 000.

	element type	$h = \frac{1}{8}$	$h = \frac{1}{16}$	$h = \frac{1}{32}$
Third-level refined mesh	noninterface elements	4.6433×10^{-3}	1.4302×10^{-3}	3.7831×10^{-4}
	interface elements	3.7265×10^{-4}	1.9026×10^{-4}	8.3144×10^{-5}
Fourth-level refined mesh	noninterface elements	4.6150×10^{-3}	1.4105×10^{-3}	3.7724×10^{-4}
	interface elements	2.5030×10^{-4}	9.3606×10^{-5}	3.1081×10^{-5}

Table IX. Discrete infinity norm errors of the symmetric selective immersed discontinuous Galerkin (SIDG) solutions on interface elements for $\beta^+ = 10$ and C_*=100 000.

h	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$
Original mesh	2.7754×10^{-2}	8.0050×10^{-3}	3.2539×10^{-3}
First-level refined mesh	5.4905×10^{-3}	1.7120×10^{-3}	3.1164×10^{-4}
Second-level refined mesh	1.7126×10^{-3}	6.2751×10^{-4}	1.6193×10^{-4}
Third-level refined mesh	3.7265×10^{-4}	1.9026×10^{-4}	8.3144×10^{-5}
Fourth-level refined mesh	2.5030×10^{-4}	9.3606×10^{-5}	3.1081×10^{-5}

Table X. Errors of the nonsymmetric selective immersed discontinuous Galerkin (SIDG) solutions without mesh refinement for $\beta^+ = 10$ and C_{**}=1000.

h	$\|u_h - u\|_0$	$\|u_h - u\|_1$
1/8	4.8417×10^{-3}	9.4182×10^{-2}
1/16	1.2938×10^{-3}	4.6984×10^{-2}
1/32	3.1571×10^{-4}	2.2336×10^{-2}
1/64	8.1634×10^{-5}	1.0568×10^{-2}
1/128	2.0260×10^{-5}	5.2162×10^{-3}
1/256	5.1259×10^{-6}	2.6168×10^{-3}

5.3. *Numerical results for the nonsymmetric selective immersed discontinuous Galerkin method*

In this section, we will numerically demonstrate the convergence of the nonsymmetric SIDG method and its insensitivity on the choice of the penalty parameter C_{**}. Table X contains the error in the solution u_h for various mesh sizes h, $\beta^+ = 10$, and the penalty constant $C_{**} = 1000$. Table XI and XII contain the error in the solution u_h for various mesh sizes h, $\beta^+ = 1000000$, and penalty constants $C_{**} = 1000$ and 0.0001 separately. All of these data are generated on meshes $\mathcal{T}_h = \mathcal{T}_h^0$ with the specified mesh size h.

Using linear regression, we can also see that the data in Table X obey

$$\|u_h - u\|_0 \approx 0.3056\, h^{1.9817},\ |u_h - u|_1 \approx 0.8244\, h^{1.0412},$$

the data in Table XI obey

$$\|u_h - u\|_0 \approx 0.2176\, h^{1.9756},\ |u_h - u|_1 \approx 0.8532\, h^{1.0682},$$

and the data in Table XII obey

$$\|u_h - u\|_0 \approx 0.1753\, h^{1.9831},\ |u_h - u|_1 \approx 0.8303\, h^{1.0634},$$

Copyright © 2013 John Wiley & Sons, Ltd.

Table XI. Errors of the nonsymmetric selective immersed discontinuous Galerkin (SIDG) solutions without mesh refinement for $\beta^+ = 1\,000\,000$ and C_{**}=1000.

h	$\|u_h - u\|_0$	$\|u_h - u\|_1$
1/8	3.3558×10^{-3}	9.0901×10^{-2}
1/16	9.4698×10^{-4}	4.5722×10^{-2}
1/32	2.4617×10^{-4}	2.1560×10^{-2}
1/64	5.8874×10^{-5}	9.5491×10^{-3}
1/128	1.4523×10^{-5}	4.6839×10^{-3}
1/256	3.7641×10^{-6}	2.3559×10^{-3}

Table XII. Errors of the nonsymmetric selective immersed discontinuous Galerkin (SIDG) solutions without mesh refinement for $\beta^+ = 1\,000\,000$ and C_{**}=0.0001.

h	$\|u_h - u\|_0$	$\|u_h - u\|_1$
1/8	4.1938×10^{-3}	8.9981×10^{-2}
1/16	5.6185×10^{-4}	4.5020×10^{-2}
1/32	1.1305×10^{-4}	2.0943×10^{-2}
1/64	5.4539×10^{-5}	9.5756×10^{-3}
1/128	1.2871×10^{-5}	4.6820×10^{-3}
1/256	3.0983×10^{-6}	2.3497×10^{-3}

and these data indicate the optimal convergence of the nonsymmetric SIDG solutions in the L^2 norm and the H^1 norm for the penalty constants $C_{**} = 1000$ and 0.0001. This numerically suggests that the nonsymmetric SIDG is insensitive to the choice of the penalty parameter C_{**}. Assigning an arbitrary positive value to C_{**} seems to be enough for the nonsymmetric SIDG method to produce satisfactory numerical solutions to an interface problem.

In our numerical experiments, we have also observed that the effect of the local mesh refinement on the nonsymmetric SIDG method is very similar to that reported for the symmetric SIDG method in the previous section; hence, we omit the related datum here to reduce the presentation.

6. Conclusion

In this paper, following the framework of the IPDG method, we have developed the SIDG method for solving interface problems in which the DG formulation is applied only at selected locations. We use the nodal basis functions of the bilinear IFE to construct a selective bilinear IFE space by imposing a certain selective rule. We also analyze the approximation capability of this space and apply it to the SIDG method.

The combination of a DG formulation and the IFE makes it possible to solve interface problems efficiently by a rectangular mesh with local mesh refinement, even if the interface has a nontrivial geometry. The selective feature of this method can be used to reduce the computational cost and/or incorporate desirable features in the numerical solver. Our interpolation error estimates and numerical examples show that the finite element solutions generated by this method have the optimal approximation capability. The numerical results also demonstrate the efficiency of the local mesh refinement.

Acknowledgements

This work is partially supported by the National Science Foundation (NSF) grant DMS-0713763, the National Science Foundation of China(10875034, 11175052), General Research Fund (GRF) of Hong Kong Special Administrative Region (HKSAR) #501012 and The Natural Sciences and Engineering of Canada (NSERC).

References

1. Reed WH, Hill TR. Triangular mesh methods for the neutron transport equation. *Tech. Report No. LA-UR-73-479*, Los Alamos Scientific Laboratory, Los Alamos, New Mexico, 1973.
2. Billet G, Ryan J. A Runge–Kutta discontinuous Galerkin approach to solve reactive flows: the hyperbolic operator. *Journal of Computational Physics* 2011; **230**(4):1064–1083.
3. Cockburn B, Hou SC, Shu CW. The Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV. The multidimensional case. *Mathematics of Computation* 1990; **54**(190):545–581.

Copyright © 2013 John Wiley & Sons, Ltd.

4. Cockburn B, Lin SY, Shu C-W. TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws. III. One-dimensional systems. *Journal of Computational Physics* 1989; **84**(1):90–113.
5. Cockburn B, Shu C-W. TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws. II. General framework. *Mathematics of Computation* 1989; **52**(186):411–435.
6. Cockburn B, Shu C-W. The Runge–Kutta local projection p_1-discontinuous-Galerkin finite element method for scalar conservation laws. *RAIRO-Mathematical Modelling and Numerical Analysis* 1991; **25**(3):337–361.
7. Cockburn B, Shu C-W. The Runge–Kutta discontinuous Galerkin method for conservation laws. V. Multidimensional systems. *Journal of Computational Physics* 1998; **141**(2):199–224.
8. Zhang Q, Shu C-W. Stability analysis and a priori error estimates of the third order explicit Runge–Kutta discontinuous Galerkin method for scalar conservation laws. *SIAM Journal on Numerical Analysis* 2010; **48**(3):1038–1063.
9. Chen Y, Huang J, Huang X, Xu Y. On the local discontinuous Galerkin method for linear elasticity. *Mathematical Problems in Engineering* 2010; **Art. ID 759547**:20.
10. Cockburn B, Shu C-W. The local discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM Journal on Numerical Analysis* 1998; **35**(6):2440–2463.
11. Ji L, Xu Y. Optimal error estimates of the local discontinuous Galerkin method for Willmore flow of graphs on Cartesian meshes. *International Journal of Numerical Analysis and Modeling* 2011; **8**(2):252–283.
12. Kanschat G. Block preconditioners for LDG discretizations of linear incompressible flow problems. *Journal of Scientific Computing* 2005; **22/23**:371–384.
13. Xia Y, Xu Y, Shu C-W. Local discontinuous Galerkin methods for the generalized Zakharov system. *Journal of Computational Physics* 2010; **229**(4):1238–1259.
14. Xu Y, Shu C-W. Local discontinuous Galerkin methods for high-order time-dependent partial differential equations. *Communications in Computational Physics* 2010; **7**(1):1–46.
15. Yan J, Osher S. A local discontinuous Galerkin method for directly solving Hamilton–Jacobi equations. *Journal of Computational Physics* 2011; **230**(1):232–244.
16. Zhou Z, Yan N. The local discontinuous Galerkin method for optimal control problem governed by convection diffusion equations. *International Journal of Numerical Analysis and Modeling* 2010; **7**(4):681–699.
17. Arnold DN. An interior penalty finite element method with discontinuous elements. *SIAM Journal on Numerical Analysis* 1982; **19**:742–760.
18. Babuska I, Baumann CE, Oden JT. A discontinuous *hp* finite element method for diffusion problems:1D analysis. *Computers & Mathematcs with Applications* 1999; **37**:103–122.
19. Douglas J, Dupont T. Interior penalty procedures for elliptic and parabolic Galerkin methods. *Lecture Notes in Physics* 1976; **58**:207–216.
20. Oden JT, Babuska I, Baumann CE. A discontinuous *hp* finite element method for diffusion problems. *Journal of Computational Physics* 1998; **146**:491–519.
21. Rivière B, Wheeler MF, Girault V. Improved energy estimates for interior penalty, constrained and discontinuous Galerkin methods for elliptic problems Part 1. *Computers & Geosciences* 1999; **3**:337–360.
22. Sun T. Discontinuous Galerkin finite element method with interior penalties for convection diffusion optimal control problem. *International Journal of Numerical Analysis and Modeling* 2010; **7**(1):87–107.
23. Wheeler MF. An elliptic collocation-finite element method with interior penalties. *SIAM Journal on Numerical Analysis* 1978; **15**:152–161.
24. Chen H, Chen Z. Stability and convergence of mixed discontinuous finite element methods for second-order differential problems. *Journal of Numerical Mathematics* 2003; **11**(4):253–287.
25. Chen H, Chen Z, Li B. Numerical study of the *hp* version of mixed discontinuous finite element methods for reaction-diffusion problems: the 1D case. *Numerical Methods Partial Differential Equations* 2003; **19**(4):525–553.
26. Chen Z. *Finite Element Methods and Their Applications*, Scientific Computation. Springer-Verlag: Berlin, 2005.
27. Li F, Xu L, Yakovlev S. Central discontinuous Galerkin methods for ideal MHD equations with the exactly divergence-free magnetic field. *Journal of Computational Physics* 2011; **230**:4828–4847.
28. Li F, Yakovlev S. A central discontinuous Galerkin method for Hamilton–Jacobi equations. *Journal of Scientific Computing* 2010; **45**(1–3): 404–428.
29. Liu Y, Shu C-W, Tadmor E, Zhang M. L^2L2 stability analysis of the central discontinuous Galerkin method and a comparison between the central and regular discontinuous Galerkin methods. *M2AN Mathematical Modelling and Numerical Analysis* 2008; **42**(4):593–607.
30. Nguyen NC, Peraire J, Cockburn B. An implicit high-order hybridizable discontinuous Galerkin method for linear convection-diffusion equations. *Journal of Computational Physics* 2009; **228**(9):3232–3254.
31. Nguyen NC, Peraire J, Cockburn B. An implicit high-order hybridizable discontinuous Galerkin method for the incompressible Navier–Stokes equations. *Journal of Computational Physics* 2011; **230**(4):1147–1170.
32. Nguyen NC, Peraire J, Cockburn B. High-order implicit hybridizable discontinuous galerkin methods for acoustics and elastodynamics. *Journal of Computational Physics* 2011; **230**(10):3695–3718.
33. Feistauer M, Kučera V, Najzar K, Prokopová J. Analysis of space-time discontinuous Galerkin method for nonlinear convection-diffusion problems. *Numerische Mathematik* 2011; **117**(2):251–288.
34. Sollie WEH, Bokhove O, van der Vegt JJW. Space-time discontinuous Galerkin finite element method for two-fluid flows. *Journal of Computational Physics* 2011; **230**(3):789–817.
35. Zhang X, Shu C-W. On positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes. *Journal of Computational Physics* 2010; **229**(23):8918–8934.
36. Zhang X, Shu C-W. Positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations with source terms. *Journal of Computational Physics* 2011; **230**(4):1238–1248.
37. Chidyagwai P, Mishev I, Rivière B. On the coupling of finite volume and discontinuous Galerkin method for elliptic problems. *Journal of Computational and Applied Mathematics* 2011; **235**(8):2193–2204.
38. Guo W, Li F, Qiu J. Local-structure-preserving discontinuous Galerkin methods with Lax-Wendroff type time discretizations for Hamilton–Jacobi equations. *Journal of Scientific Computing* 2011; **47**(2):239–257.
39. Jia Z, Zhang S. A new high-order discontinuous Galerkin spectral finite element method for Lagrangian gas dynamics in two-dimensions. *Journal of Computational Physics* 2011; **230**(7):2496–2522.

Copyright © 2013 John Wiley & Sons, Ltd.

40. Liu H, Yan J. The direct discontinuous Galerkin (DDG) method for diffusion with interface corrections. *Communications in Computational Physics* 2010; **8**(3):541–564.
41. Loverich J, Hakim A, Shumlak U. A discontinuous Galerkin method for ideal two-fluid plasma equations. *Communications in Computational Physics* 2011; **9**(2):240–268.
42. Luo H, Luo L, Ali A, Nourgaliev R, Cai C. A parallel, reconstructed discontinuous Galerkin method for the compressible flows on arbitrary grids. *Communications in Computational Physics* 2011; **9**(2):363–389.
43. Min M, Lee T. A spectral-element discontinuous Galerkin lattice Boltzmann method for nearly incompressible flows. *Journal of Computational Physics* 2011; **230**(1):245–259.
44. Minoli CAA, Kopriva DA. Discontinuous Galerkin spectral element approximations on moving meshes. *Journal of Computational Physics* 2011; **230**(5):1876–1902.
45. Peraire J, Persson P-O. The compact discontinuous Galerkin (CDG) method for elliptic problems. *SIAM Journal of Scientific Computing* 2008; **30**(4):1806–1824.
46. Wang W, Guzmán J, Shu C-W. The multiscale discontinuous Galerkin method for solving a class of second order elliptic problems with rough coefficients. *International Journal of Numerical Analysis and Modeling* 2011; **8**(1):28–47.
47. Zheng Y, Li C, Zhao Z. A fully discrete discontinuous Galerkin method for nonlinear fractional Fokker–Planck equation. *Mathematical Problems in Engineering* 2010; **Art. ID 279038**:26.
48. Adjerid S, Weinhart T. Discontinuous Galerkin error estimation for linear symmetrizable hyperbolic systems. *Mathematics of Computation* 2011; **80**(275):1335–1367.
49. Antonietti PF, Houston P. A class of domain decomposition preconditioners for *hp*hp-discontinuous Galerkin finite element methods. *Journal of Scientific Computing* 2011; **46**(1):124–149.
50. Brix K, Massjung R, Voss A. Refinement and connectivity algorithms for adaptive discontinuous Galerkin methods. *SIAM Journal of Scientific Computing* 2011; **33**(1):66–101.
51. Cheng Y, Shu C-W. Superconvergence of discontinuous Galerkin and local discontinuous Galerkin schemes for linear hyperbolic and convection-diffusion equations in one space dimension. *SIAM Journal on Numerical Analysis* 2010; **47**(6):4044–4072.
52. Duan H, Tan RCE. On the Poincaré–Friedrichs inequality for piecewise H^1 functions in anisotropic discontinuous Galerkin finite element methods. *Mathematics of Computation* 2011; **80**(273):119–140.
53. Georgoulis EH, Lakkis O, Virtanen JM. A posteriori error control for discontinuous Galerkin methods for parabolic problems. *SIAM Journal on Numerical Analysis* 2011; **49**(2):427–458.
54. Hiptmair R, Moiola A, Perugia I. Plane wave discontinuous Galerkin methods for the 2D Helmholtz equation: analysis of the *p*-version. *SIAM Journal on Numerical Analysis* 2011; **49**(1):264–284.
55. May G. On the connection between the spectral difference method and the discontinuous Galerkin method. *Communications in Computational Physics* 2011; **9**(4):1071–1080.
56. Wihler TP, Rivière B. Discontinuous Galerkin methods for second-order elliptic PDE with low-regularity solutions. *Journal of Scientific Computing* 2011; **46**(2):151–165.
57. Xie Z, Zhang Z. Uniform superconvergence analysis of the discontinuous Galerkin method for a singularly perturbed problem in 1-D. *Mathematics of Computation* 2010; **79**(269):35–45.
58. Xu J, Ostroumov PN, Mustapha B, Nolen J. Scalable direct Vlasov solver with discontinuous Galerkin method on unstructured mesh. *SIAM Journal of Scientific Computing* 2010; **32**(6):3476–3494.
59. Arnold DN, Brezzi F, Marini B, Cockburn LD. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM Journal on Numerical Analysis* 2001/02; **39**(5):1749–1779.
60. Chen Z. On the relationship of various discontinuous finite element methods for second-order elliptic equations. *East-West Journal of Numerical Mathematics* 2001; **9**(2):99–122.
61. Cockburn B, Karniadakis GE, Shu C-W. The development of discontinuous Galerkin methods. In *Discontinuous Galerkin Methods (Newport, RI, 1999)*, Lect. Notes Comput. Sci. Eng. Springer: Berlin, 2000; 3–50.
62. Girault V, Sun S, Wheeler MF, Yotov I. Coupling discontinuous Galerkin and mixed finite element discretizations using mortar finite elements. *SIAM Journal on Numerical Analysis* 2008; **46**(2):949–979.
63. Lian X, Rui H. A discontinuous Galerkin method combined with mixed finite element for seawater intrusion problem. *Journal of Systems Science & Complexity* 2010; **23**(4):830–845.
64. Perugia I, Schötzau D. On the coupling of local discontinuous Galerkin and conforming finite element methods. *Journal of Scientific Computing* 2002; **16(2001)**(4):411–433.
65. Cockburn B, Dong B, Guzmán J. A hybridizable and superconvergent discontinuous Galerkin method for biharmonic problems. *Journal of Scientific Computing* 2009; **40**(1-3):141–187.
66. Cockburn B, Gopalakrishnan J, Lazarov R. Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems. *SIAM Journal on Numerical Analysis* 2009; **47**(2):1319–1365.
67. Babuška I. The finite element method for elliptic equations with discontinuous coefficients. *Computing* 1970; **5**:207–213.
68. Babuška I, Osborn JE. Can a finite element method perform arbitrarily badly? *Mathematics of Computation* 2000; **69**(230):443–462.
69. Bramble JH, King JT. A finite element method for interface problems in domains with smooth boundary and interfaces. *Advances in Computational Mathematics* 1996; **6**:109–138.
70. Chen Z, Zou J. Finite element methods and their convergence for elliptic and parabolic interface problems. *Numerische Mathematik* 1998; **79**:175–202.
71. Heinrich B. *Finite Difference Methods on Irregular Networks*, International Series of Numerical Mathematics, Vol. 82. Birkhäuser: Boston, 1987.
72. He X-M, Lin T, Lin Y, Zhang X. Immersed finite element methods for parabolic equations with moving interface. *Numerical Methods Partial Differential Equations* 2013; **29**(2):619–646.
73. Kafafy R, Wang J, Lin T. A hybrid-grid immersed-finite-element particle-in-cell simulation model of ion optics plasma dynamics. *Dynamics of Continuous Discrete and Impulsive Systems-Series B-Applications & Algorithms* 2005; **12**:1–16.
74. Lin T, Wang J. The immersed finite element method for plasma particle simulation. *Proceedings of AIAA Aerospace Sciences Meeting*, Reno, NV, 2003, AIAA, 2003-0842.

Copyright © 2013 John Wiley & Sons, Ltd.

75. Lin T, Wang J. An immersed finite element electric field solver for ion optics modeling. *Proceedings of AIAA Joint Propulsion Conference*, Indianapolis, IN, july, 2002, AIAA, 2002-4263.
76. Wang J, He X-M, Cao Y. Modeling spacecraft charging and charged dust particle interactions on lunar surface. *Proceedings of the 10th Spacecraft Charging Technology Conference*, Biarritz, France, 2007.
77. Wang J, He X-M, Cao Y. Modeling electrostatic levitation of dusts on lunar surface. *IEEE Transactions on Plasma Science* 2008; **36**(5):2459–2466.
78. Lai M, Huang C, Huang Y. Simulating the axisymmetric interfacial flows with insoluble surfactant by immersed boundary method. *International Journal of Numerical Analysis and Modeling* 2011; **8**(1):105–117.
79. Peskin CS. Flow patterns around heart valves. *Journal of Computational Physics* 1972; **10**:252–271.
80. Peskin CS. Numerical analysis of blood flow in the heart. *Journal of Computational Physics* 1977; **25**:220–252.
81. Feng X, Li Z. Simplified immersed interface methods for elliptic interface problems with straight interfaces. *Numerical Methods Partial Differential Equations* 2012; **28**(1):188–203.
82. LeVeque RJ, Li Z. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM Journal on Numerical Analysis* 1994; **34**:1019–1044.
83. Li Z, Ito K. *The Immersed Interface Method: Numerical Solutions of PDEs Involving Interfaces and Irregular Domains*, Frontiers in Applied Mathematics, Vol. 33. Society for Industrial and Applied Mathematics (SIAM): Philadelphia, PA, 2006.
84. Zhou YC, Wei GW. On the fictitious-domain and interpolation formulations of the matched interface and boundary (MIB) method. *Journal of Computational Physics* 2006; **219**(1):228–246.
85. Zhou YC, Zhao S, Feig M, Wei GW. High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources. *Journal of Computational Physics* 2006; **213**(1):1–30.
86. Ingram DM, Causon DM, Mingham CG. Developments in Cartesian cut cell methods. *Mathematics and Computers in Simulation* 2003; **61** (3-6):561–572.
87. Ji H, Lien F-S, Yee E. An efficient second-order accurate cut-cell method for solving the variable coefficient Poisson equation with jump conditions on irregular domains. *International Journal for Numerical Methods in Fluids* 2006; **52**:723–748.
88. Hewitt DW. The embedded curved boundary method for orthogonal simulation meshes. *Journal of Computational Physics* 1997; **138**:585–616.
89. Almgre A, Bell JB, Collela P, Marthaler T. A Cartesian grid method for imcompressible Euler equations in complex geometries. *SIAM Journal of Scientific Computing* 1997; **18**:1289–1390.
90. Calhoun D. A Cartesian grid method for solving the two-dimensional Streamfunction-Vorticity equations in irregular regions. *Journal of Computational Physics* 2002; **176**:231–275.
91. Babuška I, Caloz G, Osborn JE. Special finite element methods for a class of second order elliptic problems with rough coefficients. *SIAM Journal on Numerical Analysis* 1994; **31**:945–981.
92. Babuška I, Melenk J. The partition of unity method. *International Journal for Numerical Methods in Engineering* 1997; **40**:727–758.
93. Babuška I, Osborn JE. Generalized finite element methods: their performance and relation to mixed methods. *SIAM Journal on Numerical Analysis* 1983; **20**(3):510–536.
94. Babuška I, Osborn JE. Finite element methods for the solution of problems with rough input data. In *Singular and Constructive Methods for their Treatment, Lecture Notes in Mathematics, #1121*, Grisvard P, Wendland W, Whiteman JR (eds). Springer-Verlag: New York, 1985; 1–18.
95. Belytschko T, Moës N, Usui S, Primi C. Arbitrary discontinuities in finite elements. *International Journal for Numerical Methods in Engineering* 2001; **50**:993–1013.
96. Moës N, Dolbow J, Belytschko T. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering* 1999; **46**(1):131–150.
97. Sukemar N, Chopp DL, Moës N, Belytschko T. Modeling holes and inclusions by level set in the extended finite-element method. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**:6183–6200.
98. Ewing RE, Wang H. A summary of numerical methods for time-dependent advection-dominated partial differential equations. *Journal of Computational and Applied Mathematics* 2001; **128**:423–445.
99. Ewing RE, Wang H, Russell TF. Eulerian–Lagrangian localized adjoint methods of convection-diffusion equations and their convergence analysis. *IMA Journal of Applied Mathematics* 1995; **15**:405–495.
100. Wang K, Wang H, Yu X. An immersed Eulerian–Lagrangian localized adjoint method for transient advection-diffusion equations with interfaces. *International Journal of Numerical Analysis and Modeling* 2012; **9**(1):29–42.
101. Adjerid S, Lin T. Higher-order immersed discontinuous Galerkin methods. *International Journal of Information & Systems Sciences* 2007; **3**(4):555–568.
102. Adjerid S, Lin T. *p*-th degree immersed finite element for boundary value problems with discontinuous coefficients. *Applied Numerical Mathematics* 2009; **59**(6):1303–1321.
103. Camp B, Lin T, Lin Y, Sun W. Quadratic immersed finite element spaces and their approximation capabilities. *Advances in Computational Mathematics* 2006; **24**(1-4):81–112.
104. Chou S, Kwak DY, Wee KT. Optimal convergence analysis of an immersed interface finite element method. *Advances in Computational Mathematics* 2010; **33**(2):149–168.
105. Chu Y, Cao Y, He X-M, Luo M. Asymptotic boundary conditions for two-dimensional electrostatic field problems with immersed finite elements. *Computer Physics Communications* 2011; **182**(11):2331–2338.
106. Ewing RE, Li Z, Lin T, Lin Y. The immersed finite volume element methods for the elliptic interface problems. Modelling '98 (Prague). *Mathematics and Computers in Simulation* 1999; **50**(1-4):63–76.
107. Gong Y, Li B, Li Z. Immersed-interface finite-element methods for elliptic interface problems with non-homogeneous jump conditions. *SIAM Journal on Numerical Analysis* 2008; **46**:472–495.
108. Gong Y, Li Z. Immersed interface finite element methods for elasticity interface problems with non-homogeneous jump conditions. *Numerical Mathematics: Theory, Methods and Applications* 2010; **3**(1):23–39.
109. He X-M, Lin T, Lin Y. Approximation capability of a bilinear immersed finite element space. *Numerical Methods Partial Differential Equations* 2008; **24**(5):1265–1300.
110. He X-M, Lin T, Lin Y. A bilinear immersed finite volume element method for the diffusion equation with discontinuous coefficients. *Communications in Computational Physics* 2009; **6**(1):185–202.
111. He X-M, Lin T, Lin Y. Interior penalty discontinuous Galerkin methods with bilinear IFE for a second order elliptic equation with discontinuous coefficient, dedicated to Professor David Russell's 70th birthday. *Journal of Systems Science & Complexity* 2010; **23**(3):467–483.

Copyright © 2013 John Wiley & Sons, Ltd.

112. He X-M, Lin T, Lin Y. Immersed finite element methods for elliptic interface problems with non-homogeneous jump conditions. *International Journal of Numerical Analysis and Modeling* 2011; **8**(2):284–301.
113. He X-M, Lin T, Lin Y. The convergence of the bilinear and linear immersed finite element solutions to interface problems. *Numerical Methods Partial Differential Equations* 2012; **28**(1):312–330.
114. Kafafy R, Lin T, Lin Y, Wang J. Three-dimensional immersed finite element methods for electric field simulation in composite materials. *International Journal for Numerical Methods in Engineering* 2005; **64**(7):940–972.
115. Kwak DY, Wee KT, Chang KS. An analysis of a broken p_1-nonconforming finite element method for interface problems. *SIAM Journal on Numerical Analysis* 2010; **48**(6):2117–2134.
116. Li Z. The immersed interface method using a finite element formulation. *Applied Numerical Mathematics* 1997; **27**(3):253–267.
117. Li Z, Lin T, Lin Y, Rogers RC. An immersed finite element space and its approximation capability. *Numerical Methods Partial Differential Equations* 2004; **20**(3):338–367.
118. Li Z, Lin T, Wu X. New Cartesian grid methods for interface problems using the finite element formulation. *Numerische Mathematik* 2003; **96**(1):61–98.
119. Lin T, Lin Y, Rogers RC, Ryan LM. A rectangular immersed finite element method for interface problems. In *Advances in Computation: Theory and Practice*, Vol. 7, Minev P, Lin Y (eds). Nova Science Publishers, Inc.: Commack, NY, 2001; 107–114.
120. Lin T, Lin Y, Sun W. Error estimation of a class of quadratic immersed finite element methods for elliptic interface problems. *Discrete and Continuous Dynamical Systems-Series B* 2007; **7**(4):807–823.
121. Lin T, Lin Y, Sun W, Wang Z. Immersed finite element methods for 4th order differential equations. *Journal of Computational and Applied Mathematics* 2011; **235**(13):3953–3964.
122. Lin T, Sheen D. The immersed finite element method for parabolic problems with the Laplace transformation in time discretization. *International Journal of Numerical Analysis and Modeling* 2013; **10**(2):298–313.
123. Sauter SA, Warnke R. Composite finite elements for elliptic boundary value problems with discontinuous coefficients. *Computing* 2006; **77**(1):29–55.
124. Vallaghè S, Papadopoulo T. A trilinear immersed finite element method for solving the electroencephalography forward problem. *SIAM Journal of Scientific Computing* 2010; **32**(4):2379–2394.
125. Wu C, Li Z, Lai M. Adaptive mesh refinement for elliptic interface problems using the non-conforming immersed finite element method. *International Journal of Numerical Analysis and Modeling* 2011; **8**(3):466–483.
126. Xie H, Li Z, Qiao Z. A finite element method for elasticity interface problems with locally modified triangulations. *International Journal of Numerical Analysis and Modeling* 2011; **8**(2):189–200.
127. Delves M, Hall CA. An implicit matching principle for global element calculations. *Journal of the Institute of Mathematics and its Applications* 1979; **23**:223–234.
128. He X-M. Bilinear immersed finite elements for interface problems. *Ph.D. dissertation*, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 2009.
129. Ainsworth M, Senior B. Aspects of an adaptive hp-finite element method: adaptive strategy conforming approximation and efficient solvers. *Computer Methods in Applied Mechanics and Engineering* 1997; **150**:65–87.
130. Demkowicz L, Oden J, Rachowicz W, Hardy O. Toward a universal h_p. *Computer Methods in Applied Mechanics and Engineering* 1989; **77**:79–112.

Copyright © 2013 John Wiley & Sons, Ltd.