INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2013; **93**:183–200
Published online 14 June 2012 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/nme.4382

A high-order hybridizable discontinuous Galerkin method for elliptic interface problems

L. N. T. Huynh[1,*,†], N. C. Nguyen[2], J. Peraire[2] and B. C. Khoo[1]

[1]*Singapore-MIT Alliance, National University of Singapore, Singapore 119077*
[2]*Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA*

SUMMARY

We present a high-order hybridizable discontinuous Galerkin method for solving elliptic interface problems in which the solution and gradient are nonsmooth because of jump conditions across the interface. The hybridizable discontinuous Galerkin method is endowed with several distinct characteristics. First, they reduce the globally coupled unknowns to the approximate trace of the solution on element boundaries, thereby leading to a significant reduction in the global degrees of freedom. Second, they provide, for elliptic problems with polygonal interfaces, approximations of all the variables that converge with the optimal order of $k+1$ in the $L^2(\Omega)$-norm where k denotes the polynomial order of the approximation spaces. Third, they possess some superconvergence properties that allow the use of an inexpensive element-by-element postprocessing to compute a new approximate solution that converges with order $k+2$. However, for elliptic problems with finite jumps in the solution across the curvilinear interface, the approximate solution and gradient do not converge optimally if the elements at the interface are isoparametric. The discrepancy between the exact geometry and the approximate triangulation near the curved interfaces results in lower order convergence. To recover the optimal convergence for the approximate solution and gradient, we propose to use superparametric elements at the interface.

Received 4 January 2012; Revised 8 May 2012; Accepted 31 May 2012

KEY WORDS: discontinuous Galerkin; mixed/hybrid method; elliptic partial differential equation; curvilinear interface; jump condition; superparametric element

1. INTRODUCTION

Partial differential equations with discontinuous coefficients and/or discontinuous solutions are often encountered in fluid dynamics and solid mechanics. For instance, many problems in engineering and science involve multiple distinct materials or fluids with different densities, diffusions, conductivities, Young's modulus, or Poisson ratio. Such problems include multiphase flows, composite structures, and heat transfer in porous media. Another class of problems of engineering interest involves solutions having a finite jump across an interface. This class of problems arises when there is a singular force applied on the interface [1–3] and is more difficult to deal with than problems with discontinuous coefficients.

For PDEs with interface jump conditions, the smoothness of the solution and the curvature of the interface may have some important implications. The solution of the interface problem might be smooth in individual regions occupied by different materials or fluids, but the global solution distribution might be nonsmooth because of jump conditions across the interface. The shape and

*Correspondence to: L. N. T. Huynh, Singapore-MIT Alliance, National University of Singapore, Singapore 119077.
†E-mail: hlengocthanh@gmail.com

the curvature of the interface can add an extra difficulty for numerical methods used to solve the interface problem as we shall point it out later in the paper. The discontinuity in the solution and the high curvature of the interface are a considerable challenge for any numerical method to compute solution with high-order accuracy.

Several methods are available in the literature for solving interface problems. They include immersed interface method [1], fitted finite element method [4], unfitted finite element method [5], and embedded finite element method [6]. Although these methods can treat both stationary and moving interfaces through the use of fitted or unfitted grids, they are only second-order accurate. Moreover, it is not clear that these methods can achieve second-order accuracy for problems with discontinuous solutions on general curved interfaces.

Recently, in [7], the local discontinuous Galerkin (LDG) method has been extended to solve elliptic interface problems. The LDG method possesses some advantages over other continuous Galerkin finite element methods (CG-FEMs) for solving interface problems. First, the resulting stiffness matrix is the same as the matrix obtained by using the standard LDG method for elliptic problems without jump conditions because all the discontinuity terms appear on the right-hand side. Second, the LDG method can provide high-order accurate solutions for problems in which the domain, boundary data, interface, and jumps are smooth enough. However, the stiffness matrix of the LDG method is considerably larger than that of CG-FEMs because the LDG solution has nodal duplications along element boundaries. Therefore, the LDG method is more computationally expensive than CG-FEMs. Moreover, the previous work [7] does not deal with general curved interfaces effectively.

In this paper, we present a high-order hybridizable discontinuous Galerkin (HDG) method for numerically solving elliptic interface problems. The method developed here is an extension of the HDG method introduced in [8] and analyzed in [9–11] for elliptic problems without jump conditions. The essential ingredients of the method are: (1) a local Galerkin projection of the underlying PDE at the element level onto spaces of polynomials of degree k to capture the jump in the solution and parametrize the numerical solution in terms of the numerical trace; (2) a judicious choice of the numerical flux to provide stability and consistency; and (3) a global Galerkin projection that weakly enforces the continuity of the numerical flux and weakly imposes boundary conditions and jump conditions in the flux. Although only elliptic interface problems are considered in this paper, the method can be extended to convection–diffusion, Stokes, and incompressible Navier–Stokes problems with jump conditions [12, 13] by combining the present idea with the HDG methods for incompressible flows introduced in [14–19] and the interface stabilized finite element method introduced in [20, 21].

The HDG method possesses all the advantages of the LDG method and offers additional distinct features. First, they reduce the globally coupled unknowns to the approximate trace of the solution on element boundaries, thereby leading to a significant reduction in the global degrees of freedom. Second, they provide, for elliptic problems with polygonal interfaces, approximations of all the variables that converge with the optimal order of $k + 1$ in the $L^2(\Omega)$-norm. And third, they possess some superconvergence properties that allow us, by means of local post-processing, to obtain a new approximate solution that converges with order $k + 2$. However, for elliptic problems with general curvilinear interfaces, the approximate solution and gradient do not converge optimally if the elements at the interface are isoparametric. The discrepancy between the exact geometry and the approximate triangulation near the curved interfaces results in lower order convergence [7]. To recover the optimal convergence of order $k + 1$ for the approximate solution and gradient, we propose to use superparametric elements at the interface. We present numerical results to demonstrate the convergence properties and accuracy of the HDG method.

The paper is organized as follows. In Section 2, we introduce the problem statement and notation used throughout the paper. In Section 3, we describe the HDG method for solving elliptic interface problems. In Section 4, we discuss the effect of curved interfaces on accuracy and provide an effective treatment by utilizing superparametric elements near the interface. In Section 5, we provide numerical examples to demonstrate the proposed method. Finally, we present some concluding remarks in Section 6.

DOI: 10.1002/nme

2. PROBLEM STATEMENT AND NOTATION

2.1. Elliptic interface problem

Let Ω be a bounded domain in $\mathbb{R}^d$ with Lipschitz boundary $\partial\Omega$. Let $\partial\Omega_{\mathrm{D}}$ and $\partial\Omega_{\mathrm{N}}$ be two disjoint portions of $\partial\Omega$. Let Γ be an interior interface that splits Ω into two disjoint subdomains Ω^1 and Ω^2. We define

$$\Gamma^1 \equiv \left\{ \boldsymbol{x} - \epsilon \boldsymbol{n}^1 \ : \ \boldsymbol{x} \in \Gamma \text{ and } \epsilon \to 0 \right\},$$
$$\Gamma^2 \equiv \left\{ \boldsymbol{x} - \epsilon \boldsymbol{n}^2 \ : \ \boldsymbol{x} \in \Gamma \text{ and } \epsilon \to 0 \right\},$$

where $\boldsymbol{n}^1$ (respectively, $\boldsymbol{n}^2$) is the normal unit vector that points outward the subdomain Ω^1 (respectively, Ω^2). Note that Γ^1 and Γ^2 are considered as the interior boundary of Ω^1 and Ω^2, respectively. Figure 1 shows the geometric setting of the problem.

We consider solving the following elliptic problem

$$\nabla \cdot (-\nu \nabla u) = f, \text{ in } \Omega, \tag{1}$$

with boundary conditions

$$\begin{aligned} u &= g_D, \quad \text{on } \partial\Omega_{\mathrm{D}}, \\ -\nu_2 \nabla u \cdot \boldsymbol{n} &= g_N, \quad \text{on } \partial\Omega_{\mathrm{N}}, \end{aligned} \tag{2}$$

and jump conditions on the interface

$$\begin{aligned} u|_{\Gamma^1} - u|_{\Gamma^2} &= s_D, \quad \text{on } \Gamma, \\ -\left(\nu_1 \nabla u|_{\Gamma^1} \cdot \boldsymbol{n}^1 + \nu_2 \nabla u|_{\Gamma^2} \cdot \boldsymbol{n}^2\right) &= s_N, \quad \text{on } \Gamma. \end{aligned} \tag{3}$$

The governing equation (1) and its conditions can be rewritten as

$$\begin{aligned} \boldsymbol{q} - \nabla u &= 0, && \text{in } \Omega, \\ \nabla \cdot (-\nu \boldsymbol{q}) &= f, && \text{in } \Omega, \\ u &= g_D, && \text{on } \partial\Omega_{\mathrm{D}}, \\ -\nu \boldsymbol{q} \cdot \boldsymbol{n} &= g_N, && \text{on } \partial\Omega_{\mathrm{N}}, \\ u|_{\Gamma^1} - u|_{\Gamma^2} &= s_D, && \text{on } \Gamma, \\ -\left(\nu_1 \boldsymbol{q}|_{\Gamma^1} \cdot \boldsymbol{n}^1 + \nu_2 \boldsymbol{q}|_{\Gamma^2} \cdot \boldsymbol{n}^2\right) &= s_N, && \text{on } \Gamma. \end{aligned} \tag{4}$$

Here, f is a source term, g_D and g_N are boundary data, and s_D and s_N are interfacial jumps. We assume that these functions are given and smooth.

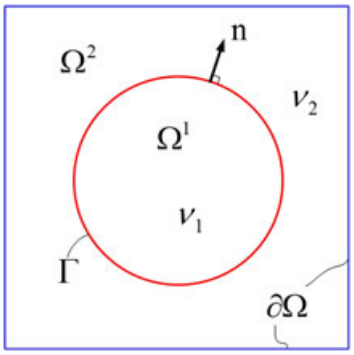


Figure 1. A circular interface Γ is immersed inside Ω. Material coefficients ν_1 and ν_2 are not the same in Ω^1 (inside the circle) and Ω^2 (outside the circle).

DOI: 10.1002/nme

In general, the aforementioned elliptic problem results in nonsmooth solution because of the jump conditions (3) on the interface. In particular, when both s_D and s_N are nonzero, the solution u does not belong to the Hilbert space $H^1(\Omega)$ but to the Lebesgue space $L^2(\Omega)$. These spaces are defined as follows:

$$L^2(\Omega) = \left\{ v \ : \ \int_\Omega |v|^2 < \infty \right\}, \tag{5}$$

$$H^1(\Omega) = \left\{ v \ : \int_\Omega |v|^2 < \infty, \int_\Omega |\nabla v|^2 < \infty \right\}. \tag{6}$$

The curved interface and the nonsmooth solution of the problem represent a considerable challenge for any numerical method to compute solution with high-order accuracy. Our objective in this paper is to develop an HDG method with high-order of accuracy for the approximate solution and gradient. To describe the HDG method, we need to introduce some notation.

2.2. *Finite element mesh*

We denote by $\mathcal{T}_h^j$ $(j = 1, 2)$ a collection of disjoint elements K that partition Ω^j. We assume that $\mathcal{T}_h^1 \cap \mathcal{T}_h^2 = \emptyset$ and that $\mathcal{T}_h = \Omega$, where $\mathcal{T}_h$ denotes the union of $\mathcal{T}_h^1$ and $\mathcal{T}_h^2$. We then set

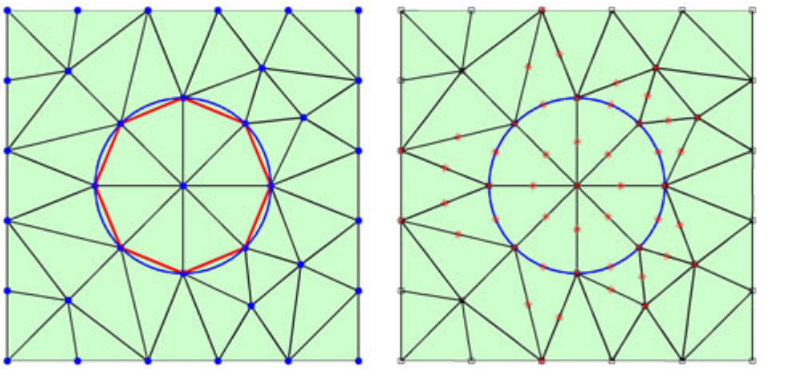
$$\partial\mathcal{T}_h := \{\partial K : \ K \in \mathcal{T}_h\}, \tag{7}$$

where ∂K denotes the boundary of the element K. We also introduce

$$A_h := \left(\Omega^1 \cap \mathcal{T}_h^2\right) \cup \left(\Omega^2 \cap \mathcal{T}_h^1\right). \tag{8}$$

Note that A_h is an empty region only if $\mathcal{T}_h^j = \Omega^j, j = 1, 2$, that is, if our finite element mesh represents the interface Γ exactly. However, in general, we may have $\mathcal{T}_h^j \neq \Omega^j, j = 1, 2$. In such case, the region A_h is nonempty. For example, Figure 2(a) shows the approximation of a circular interface inside a square using linear isoparametric elements. We observe that $\Omega^2 \cap \mathcal{T}_h^1 = 0$ and $\Omega^1 \cap \mathcal{T}_h^2 \neq 0$. Sum of the curved strips in between the red segments and the blue circle in Figure 2(a) is A_h. The convergence rate of the L^2-norm error inside A_h will be discussed in Section 4.

For an element $K \in \mathcal{T}_h$, $F = \partial K \cap \partial\Omega$ is the boundary face if the $d-1$ measure of F is nonzero. We denote by $\mathcal{E}_h^\partial$ the union of all boundary faces. For two elements K^+ and K^- of the collection $\mathcal{T}_h$, $F = \partial K^+ \cap \partial K^-$ is the interior face between K^+ and K^- if the $d-1$ measure of F is nonzero. Those interior faces whose vertices lie on Γ shall be called interfacial faces. The remaining interior faces shall be called interelement faces. We denote by Γ_h the union of all interfacial faces and by $\mathcal{E}_h^o$ the union of all interelement faces.



(a) Iso-parametric elements with $k = 1$ (b) Super-parametric elements with $k^* = 2$

Figure 2. Mesh generation of a circular interface inside a square. (a) Isoparametric elements with $k = 1$ and (b) superparametric elements with $k^* = 2$.

DOI: 10.1002/nme

Finally, we denote by $\mathcal{E}_h$ the set of all boundary faces, interelement faces, and interfacial faces, namely,

$$\mathcal{E}_h := \mathcal{E}_h^{\partial} \cup \mathcal{E}_h^{o} \cup \Gamma_h. \tag{9}$$

We also introduce a switch function δ_{Γ_h} that is defined on $\partial\mathcal{T}_h$ and satisfies

$$\delta_{\Gamma_h} = \begin{cases} 1, & \text{if } \partial K \cap \Gamma_h \neq \emptyset \text{ and } K \in \mathcal{T}_h^1 \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

In words, δ_{Γ_h} is equal to 1 on the intersection $\partial K \cap \Gamma_h$ only if (i) $K \in \mathcal{T}_h^1$ and (ii) the intersection is nonempty. This function is introduced to capture the discontinuity in the solution across the interface.

2.3. Approximation spaces

Let $\mathcal{P}^k(D)$ denote the space of polynomials of degree at most k on a domain D. We introduce the following discontinuous approximation spaces

$$\mathcal{W}_h^k = \left\{ w \in L^2(\mathcal{T}_h) \,:\, w|_K \in \mathcal{P}^k(K),\ \forall\ K \in \mathcal{T}_h \right\},$$
$$\boldsymbol{\mathcal{V}}_h^k = \left\{ \boldsymbol{v} \in [L^2(\mathcal{T}_h)]^d \,:\, \boldsymbol{v}|_K \in [\mathcal{P}^k(K)]^d,\ \forall\ K \in \mathcal{T}_h \right\}.$$

In addition, we introduce a finite element approximation space for the approximate trace of the solution

$$\mathcal{M}_h^k = \left\{ \mu \in L^2(\mathcal{E}_h) \,:\, \mu|_F \in \mathcal{P}^k(F),\ \forall\ F \in \mathcal{E}_h \right\}.$$

Note that $\mathcal{M}_h^k$ consists of functions that are continuous inside the facets and discontinuous at their borders.

Finally, we define various inner products for our finite element spaces. For functions a and b in $L^2(D)$, we denote $(a,b)_D = \int_D ab$ if D is a domain in $\mathbb{R}^d$ and $\langle a,b\rangle_D = \int_D ab$ if D is a domain in $\mathbb{R}^{d-1}$. For functions $\boldsymbol{a}$ and $\boldsymbol{b}$ in $[L^2(D)]^d$, we denote $(\boldsymbol{a},\boldsymbol{b})_D = \int_D \boldsymbol{a}\cdot\boldsymbol{b}$ if D is a domain in $\mathbb{R}^d$ and $\langle \boldsymbol{a},\boldsymbol{b}\rangle_D = \int_D \boldsymbol{a}\cdot\boldsymbol{b}$ if D is a domain in $\mathbb{R}^{d-1}$. We define the volume inner products as

$$(a,b)_{\mathcal{T}_h} = \sum_{K\in\mathcal{T}_h} (a,b)_K, \quad (\boldsymbol{a},\boldsymbol{b})_{\mathcal{T}_h} = \sum_{K\in\mathcal{T}_h} (\boldsymbol{a},\boldsymbol{b})_K,$$

and boundary inner products as

$$\langle a,b\rangle_{\partial\mathcal{T}_h} = \sum_{K\in\mathcal{T}_h} \langle a,b\rangle_{\partial K}, \quad \langle \boldsymbol{a},\boldsymbol{b}\rangle_{\partial\mathcal{T}_h} = \sum_{K\in\mathcal{T}_h} \langle \boldsymbol{a},\boldsymbol{b}\rangle_{\partial K}.$$

We are now ready to describe the HDG method for solving the system (4).

3. HYBRIDIZABLE DISCONTINUOUS GALERKIN METHOD

3.1. Formulation

In the HDG method, we aim to solve a global matrix system involving the degrees of freedom of an *approximate trace* $\widehat{u}_h \in \mathcal{M}_h$, which is an approximation to the exact solution u on $\mathcal{E}_h$. We note, however, that u is double-valued on Γ, whereas $\widehat{u}_h$ is single-valued on any given face $F \in \Gamma_h$. To resolve this issue, we let the value of $\widehat{u}_h$ on Γ_h, $\widehat{u}_h|_{\Gamma_h}$, be an approximation to $u|_{\Gamma^2}$. We then take $\widehat{u}_h|_{\Gamma_h} + s_D$ to be an approximation to $u|_{\Gamma^1}$ as derived from (3).

We next consider the governing equations (4) on one element K of $\mathcal{T}_h$, multiply with test functions $\boldsymbol{v}$ and w, respectively, and integrate the resulting equations by parts to obtain an approximation $(\boldsymbol{q}_h, u_h) \in \boldsymbol{\mathcal{V}}_h^k \times \mathcal{W}_h^k$ that satisfies

$$(\boldsymbol{q}_h,\boldsymbol{v})_K + (u_h, \nabla\cdot\boldsymbol{v})_K - \langle \widetilde{u}_h, \boldsymbol{v}\cdot\boldsymbol{n}\rangle_{\partial K} = 0, \tag{11a}$$
$$(\nu\boldsymbol{q}_h, \nabla w)_K - \langle \nu\widehat{\boldsymbol{q}}_h\cdot\boldsymbol{n}, w\rangle_{\partial K} = (f,w)_K, \tag{11b}$$

DOI: 10.1002/nme

for all $(\boldsymbol{v}, w) \in [\mathcal{P}^k(K)]^d \times \mathcal{P}^k(K)$, where the numerical flux $\nu\widehat{\boldsymbol{q}}_h$ is defined as

$$\nu\widehat{\boldsymbol{q}}_h = \nu\boldsymbol{q}_h - \tau(u_h - \widetilde{u}_h)\boldsymbol{n}. \tag{12}$$

The stabilization parameter τ is defined in (19). Here, to capture the jump condition in the solution across the interface, the function $\widetilde{u}_h$ is defined in terms of $\widehat{u}_h$ as

$$\widetilde{u}_h = \begin{cases} \widehat{u}_h + s_D, & \text{if } \partial K \cap \Gamma_h \neq \emptyset \text{ and } K \in \mathcal{T}_h^1 \\ \widehat{u}_h, & \text{otherwise.} \end{cases} \tag{13}$$

Note that $\widetilde{u}_h$ incorporates the jump condition in the solution across the interface. In fact, it is an approximation to u on $\mathcal{E}_h$ and on both sides of the interface Γ_h.

By using the definition of δ_{Γ_h} in (10), we can rewrite (11)–(13) as

$$(\boldsymbol{q}_h, \boldsymbol{v})_K + (u_h, \nabla\cdot\boldsymbol{v})_K - \left\langle \widehat{u}_h + \delta_{\Gamma_h} s_D, \boldsymbol{v}\cdot\boldsymbol{n}\right\rangle_{\partial K} = 0, \tag{14a}$$

$$(\nu\boldsymbol{q}_h, \nabla w)_K - \langle \nu\widehat{\boldsymbol{q}}_h\cdot\boldsymbol{n}, w\rangle_{\partial K} = (f, w)_K, \tag{14b}$$

where

$$\nu\widehat{\boldsymbol{q}}_h = \nu\boldsymbol{q}_h - \tau(u_h - \widehat{u}_h - \delta_{\Gamma_h} s_D)\boldsymbol{n}. \tag{15}$$

We note that if $\widehat{u}_h$ is available, then we can substitute (15) into (14) and solve the resulting system for $(\boldsymbol{q}_h, u_h)$ in an element-by-element fashion. Hence, (14) defines a *local problem* that determines $(\boldsymbol{q}_h, u_h)$ as a function of $\widehat{u}_h$.

To determine $\widehat{u}_h$, we enforce weakly the jump condition in the flux across interior faces and impose weakly the boundary conditions. In particular, we require that $\widehat{u}_h \in \mathcal{M}_h^k$ satisfies

$$\langle \nu\widehat{\boldsymbol{q}}_h\cdot\boldsymbol{n}, \mu\rangle_{\partial\mathcal{T}_h\setminus(\partial\Omega_D\cup\partial\Omega_N)} - \langle s_N, \mu\rangle_{\Gamma_h} + \langle \widehat{u}_h - g_D, \mu\rangle_{\partial\Omega_D} + \langle \nu\widehat{\boldsymbol{q}}_h\cdot\boldsymbol{n} - g_N, \mu\rangle_{\partial\Omega_N} = 0,$$

for all $\mu \in \mathcal{M}_h^k$. The earlier equation can be written as $\widehat{u}_h \in \mathcal{M}_h^k$ satisfies

$$\begin{aligned} \langle \nu\boldsymbol{q}_h\cdot\boldsymbol{n} - \tau(u_h - \widehat{u}_h), \mu\rangle_{\partial\mathcal{T}_h\setminus\partial\Omega_D} + \langle \widehat{u}_h, \mu\rangle_{\partial\Omega_D} &= \langle g_D, \mu\rangle_{\partial\Omega_D} + \langle g_N, \mu\rangle_{\partial\Omega_N} \\ &\quad + \langle s_N - \tau s_D, \mu\rangle_{\Gamma_h}, \end{aligned} \tag{16}$$

for all $\mu \in \mathcal{M}_h^k$. This equation can be considered as a *global weak formulation* in terms of $\widehat{u}_h$ because $(\boldsymbol{q}_h, u_h)$ as defined by (14)–(15) is in fact a function of $\widehat{u}_h$.

Finally, by substituting (15) into (14), adding the resulting equations over all elements and rearranging some terms, we obtain the HDG formulation: find $(\boldsymbol{q}_h, u_h, \widehat{u}_h) \in \boldsymbol{\mathcal{V}}_h^k \times \mathcal{W}_h^k \times \mathcal{M}_h^k$ such that

$$(\boldsymbol{q}_h, \boldsymbol{v})_{\mathcal{T}_h} + (u_h, \nabla\cdot\boldsymbol{v})_{\mathcal{T}_h} - \langle \widehat{u}_h, \boldsymbol{v}\cdot\boldsymbol{n}\rangle_{\partial\mathcal{T}_h} = \left\langle \delta_{\Gamma_h} s_D, \boldsymbol{v}\cdot\boldsymbol{n}\right\rangle_{\partial\mathcal{T}_h}, \tag{17a}$$

$$(\nu\boldsymbol{q}_h, \nabla w)_{\mathcal{T}_h} - \langle \nu\boldsymbol{q}_h\cdot\boldsymbol{n} - \tau(u_h - \widehat{u}_h), w\rangle_{\partial\mathcal{T}_h} = (f, w)_{\mathcal{T}_h} + \left\langle \tau\delta_{\Gamma_h} s_D, w\right\rangle_{\partial\mathcal{T}_h}, \tag{17b}$$

$$\begin{aligned} \langle \nu\boldsymbol{q}_h\cdot\boldsymbol{n} - \tau(u_h - \widehat{u}_h), \mu\rangle_{\partial\mathcal{T}_h\setminus\partial\Omega_D} + \langle \widehat{u}_h, \mu\rangle_{\partial\Omega_D} &= \langle g_D, \mu\rangle_{\partial\Omega_D} + \langle g_N, \mu\rangle_{\partial\Omega_N} \\ &\quad + \langle s_N - \tau s_D, \mu\rangle_{\Gamma_h}, \end{aligned} \tag{17c}$$

for all $(\boldsymbol{v}, w, \mu) \in \boldsymbol{\mathcal{V}}_h^k \times \mathcal{W}_h^k \times \mathcal{M}_h^k$. This system completely defines the HDG method.

3.2. Relation to the hybridizable discontinuous Galerkin method for standard elliptic problems

We recall from [8–10] that the HDG method for standard elliptic problems without interface conditions is defined as follows: find $(\boldsymbol{q}_h, u_h, \widehat{u}_h) \in \boldsymbol{\mathcal{V}}_h^k \times \mathcal{W}_h^k \times \mathcal{M}_h^k$ such that

$$(\boldsymbol{q}_h, \boldsymbol{v})_{\mathcal{T}_h} + (u_h, \nabla\cdot\boldsymbol{v})_{\mathcal{T}_h} - \langle \widehat{u}_h, \boldsymbol{v}\cdot\boldsymbol{n}\rangle_{\partial\mathcal{T}_h} = 0, \tag{18a}$$

$$(\nu\boldsymbol{q}_h, \nabla w)_{\mathcal{T}_h} - \langle \nu\boldsymbol{q}_h\cdot\boldsymbol{n} - \tau(u_h - \widehat{u}_h), w\rangle_{\partial\mathcal{T}_h} = (f, w)_{\mathcal{T}_h}, \tag{18b}$$

$$\langle \nu\boldsymbol{q}_h\cdot\boldsymbol{n} - \tau(u_h - \widehat{u}_h), \mu\rangle_{\partial\mathcal{T}_h\setminus\partial\Omega_D} + \langle \widehat{u}_h, \mu\rangle_{\partial\Omega_D} = \langle g_D, \mu\rangle_{\partial\Omega_D} + \langle g_N, \mu\rangle_{\partial\Omega_N}, \tag{18c}$$

for all $(\boldsymbol{v}, w, \mu) \in \boldsymbol{\mathcal{V}}_h^k \times \mathcal{W}_h^k \times \mathcal{M}_h^k$.

DOI: 10.1002/nme

Several remarks are in order. First, the present method (17) is a natural extension of the method (18) to elliptic interface problems. Indeed, the formulation (18) is a particular case of the more general formulation (17) for $s_D = 0$ and $s_N = 0$. As a result, the latter inherits all the convergence properties of the former. In particular, for regular interface problems in which the data and geometry are sufficiently smooth, both the approximate solution and gradient converge with the optimal order $k+1$ in the $L^2(\Omega)$-norm; see [9, 10]. Second, the HDG method provides an elegant treatment of the interface conditions so that all the discontinuity terms appear as extra terms on the right-hand side. Therefore, the global stiffness matrix does not change, and the matrix is symmetric positive-definite. Finally, we note that the stabilization parameter is chosen as

$$\tau = \frac{\nu}{\ell}, \tag{19}$$

where ℓ is a characteristic length scale. The optimal value of τ turns out to be a unit length [18].

3.3. Expression of the numerical traces

We now derive an explicit expression for $\widehat{u}_h$ and $\widehat{\boldsymbol{q}}_h$ in terms of u_h and $\boldsymbol{q}_h$ that can shed light on the nature of the HDG method. It is clear from (16) that

$$\widehat{u}_h = g_D, \text{ on } \partial\Omega_D, \tag{20}$$

$$\nu_2 \boldsymbol{q}_h \cdot \boldsymbol{n} + \tau(u_h - \widehat{u}_h) = g_N, \text{ on } \partial\Omega_N. \tag{21}$$

The latter equation yields

$$\widehat{u}_h = \frac{1}{\tau}(\nu_2 \boldsymbol{q}_h \cdot \boldsymbol{n} - g_N) + u_h, \text{ on } \partial\Omega_N. \tag{22}$$

Furthermore, on any interelement face $F \in \mathcal{E}_h^o$ shared by two elements K^- and K^+, we obtain from (15) and (16) that

$$\nu^+ \boldsymbol{q}_h^+ \cdot \boldsymbol{n}^+ + \tau\left(u_h^+ - \widehat{u}_h\right) + \nu^- \boldsymbol{q}_h^- \cdot \boldsymbol{n}^- + \tau\left(u_h^- - \widehat{u}_h\right) = 0, \tag{23}$$

which gives

$$\widehat{u}_h = \frac{1}{2}\left(u_h^+ + u_h^-\right) + \frac{1}{2\tau}\left(\nu^+ \boldsymbol{q}_h^+ - \nu^- \boldsymbol{q}_h^-\right) \cdot \boldsymbol{n}^+, \text{ on } \mathcal{E}_h^o, \tag{24}$$

where $\nu^\pm = \nu|_{K^\pm}$, $u_h^\pm = u_h|_{K^\pm}$, and $\boldsymbol{q}_h^\pm = \boldsymbol{q}_h|_{K^\pm}$. Similarly, we have

$$\widehat{u}_h = \frac{1}{2}\left(u_h^+ + u_h^- - s_D\right) + \frac{1}{2\tau}\left(\left(\nu^+ \boldsymbol{q}_h^+ - \nu^- \boldsymbol{q}_h^-\right) \cdot \boldsymbol{n}^+ - s_N\right), \text{ on } \Gamma_h. \tag{25}$$

Therefore, we obtain the following formula for $\widehat{u}_h$ as

$$\widehat{u}_h = \begin{cases} g_D, & \text{on } \partial\Omega_D, \\ \frac{1}{\tau}(\nu_2 \boldsymbol{q}_h \cdot \boldsymbol{n} - g_N) + u_h, & \text{on } \partial\Omega_N, \\ \frac{1}{2}\left(u_h^+ + u_h^-\right) + \frac{1}{2\tau}\left(\nu^+ \boldsymbol{q}_h^+ - \nu^- \boldsymbol{q}_h^-\right) \cdot \boldsymbol{n}^+, & \text{on } \mathcal{E}_h^o, \\ \frac{1}{2}\left(u_h^+ + u_h^- - s_D\right) + \frac{1}{2\tau}\left(\left(\nu^+ \boldsymbol{q}_h^+ - \nu^- \boldsymbol{q}_h^-\right) \cdot \boldsymbol{n}^+ - s_N\right), & \text{on } \Gamma_h. \end{cases} \tag{26}$$

We next substitute the earlier expression into (15) to obtain

$$\nu\widehat{\boldsymbol{q}}_h = \begin{cases} \nu_2 \boldsymbol{q}_h + \tau(u_h - g_D)\boldsymbol{n}, & \text{on } \partial\Omega_D, \\ g_N \boldsymbol{n}, & \text{on } \partial\Omega_N, \\ \frac{1}{2}\left(\nu^+ \boldsymbol{q}_h^+ + \nu^- \boldsymbol{q}_h^-\right) + \frac{\tau}{2}\left(u_h^+ - u_h^-\right)\boldsymbol{n}^+, & \text{on } \mathcal{E}_h^o, \\ \frac{1}{2}\left(\nu^+ \boldsymbol{q}_h^+ + \nu^- \boldsymbol{q}_h^- + s_N \boldsymbol{n}^+\right) + \frac{\tau}{2}\left(u_h^+ - u_h^- + s_D\right)\boldsymbol{n}^+, & \text{on } \Gamma_h. \end{cases} \tag{27}$$

These expressions reveal the relationship between the HDG method and several DG methods analyzed in [22]. In particular, the HDG method can be viewed as a DG method defined by (14) with the aforementioned choice of the numerical traces.

DOI: 10.1002/nme

3.4. Implementation

There are two different strategies for implementing the HDG method. We briefly describe the direct implementation in this paper and refer the readers to [23] for details of the second strategy. We begin by noting that the HDG formulation (17) yields a linear system of the form

$$\begin{bmatrix} \mathbf{A} & -\mathbf{B}^T & -\mathbf{C}^T \\ \mathbf{B} & \mathbf{D} & \mathbf{E}^T \\ \mathbf{C} & \mathbf{E} & \mathbf{F} \end{bmatrix} \begin{bmatrix} \mathbf{q} \\ \mathbf{u} \\ \widehat{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{g} \\ \mathbf{f} \\ \mathbf{h} \end{bmatrix}. \tag{28}$$

Here, $\mathbf{q}$, $\mathbf{u}$, and $\widehat{\mathbf{u}}$ represent the vector of degrees of freedom for $\boldsymbol{q}_h$, u_h, and $\widehat{u}_h$, respectively. We next eliminate $\mathbf{q}$ and $\mathbf{u}$ to obtain a reduced globally coupled matrix equation only for $\widehat{\mathbf{u}}$ as

$$\mathbf{K}\,\widehat{\mathbf{u}} = \mathbf{r}, \tag{29a}$$

where

$$\mathbf{K} = -\begin{bmatrix} \mathbf{C} & \mathbf{E} \end{bmatrix} \begin{bmatrix} \mathbf{A} & -\mathbf{B}^T \\ \mathbf{B} & \mathbf{D} \end{bmatrix}^{-1} \begin{bmatrix} -\mathbf{C}^T \\ \mathbf{E} \end{bmatrix} + \mathbf{F}, \tag{29b}$$

and

$$\mathbf{r} = \mathbf{h} - \begin{bmatrix} \mathbf{C} & \mathbf{E} \end{bmatrix} \begin{bmatrix} \mathbf{A} & -\mathbf{B}^T \\ \mathbf{B} & \mathbf{D} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{g} \\ \mathbf{f} \end{bmatrix}. \tag{29c}$$

Because of the discontinuous nature of the approximation space $\mathcal{V}_h^k \times \mathcal{W}_h^k$, the matrix $[\mathbf{A}\ -\mathbf{B}^T; \mathbf{B}\ \mathbf{D}]$ is block-diagonal and inverted in an element-by-element fashion to yield a block-diagonal inverse. Therefore, the stiffness matrix $\mathbf{K}$ and vector $\mathbf{r}$ can be formed efficiently.

3.5. Local postprocessing

We use a local postprocessing proposed in [24] to improve the accuracy of the numerical solution. On every element $K \in \mathcal{T}_h$, we define a new approximate solution $u_h^* \in \mathcal{P}^{k+1}(K)$ such that it satisfies

$$\begin{aligned} \left(\nabla u_h^*, \nabla w\right)_K &= (\boldsymbol{q}_h, \nabla w)_K, \quad \forall\, w \in \mathcal{P}^{k+1}(K), \\ \left(u_h^*, 1\right)_K &= (u_h, 1)_K. \end{aligned} \tag{30}$$

The new approximation u_h^* will converge with order $k+2$ whenever the approximate gradient q_h converges with the optimal order $k+1$ and the average of the original approximation superconverges with order $k+2$ [9, 10]. We note, however, that this local postprocessing is not effective for the LDG method [7] because the approximate gradient of the LDG method converges with the suboptimal order k.

4. SUPERPARAMETRIC TREATMENT OF CURVED INTERFACES

4.1. Definition of the numerical error

We measure the numerical error using the $L^2(\Omega)$-norm as follows:

$$\|u - u_h\|_\Omega := \left(\int_\Omega (u-u_h)^2\right)^{1/2} = \left(\int_{\Omega\setminus A_h} (u-u_h)^2 + \int_{A_h} (u-u_h)^2\right)^{1/2}. \tag{31}$$

We recall that A_h is the region in which our approximate solution u_h is *not* an accurate approximation to the exact solution u if the amount of the jump s_D is nonzero and finite. In particular, in the region A_h, the absolute value $|u-u_h|$ is in the order of $|s_D|$ regardless of h and k. The mismatch between the numerical solution and exact one may have a negative effect on accuracy as discussed in the succeeding paragraphs. This problem also appears in the $L^2(\Omega)$-error norm of the approximate flux, $\|\boldsymbol{q} - \boldsymbol{q}_h\|_\Omega$, which is defined in the same fashion as the error norm of the approximate solution.

DOI: 10.1002/nme

4.2. *Effect of the curved interface on accuracy*

In this paper, we are interested in elliptic interface problems in which the interface Γ is curved and the amount of the jump is finite. It turns out for such problems that the geometric representation of the interface Γ has a great influence on the accuracy. In particular, we shall show that the use of isoparametric elements may reduce the optimal convergence rate of the approximate solution. The optimal convergence rate can be retained by using superparametric elements.

For simplicity of exposition, we assume that $s_D = 1$ and that $|u - u_h| = s_D = 1$ in the region A_h. In such case, the $L^2(\Omega)$-error norm is given by

$$\|u - u_h\|_\Omega = \left(\int_{\Omega \setminus A_h} (u - u_h)^2 + \text{volume}(A_h) \right)^{1/2}. \tag{32}$$

To estimate the volume of A_h, we assume that polynomials of degree n are used to represent all the elements $K \in \mathcal{T}_h$ near the interface Γ. It is also reasonable to assume that the number of elements near the interface is $O(1/h^{d-1})$. Now, considering any given element K whose face F lies on Γ_h, let E be a *straight* face (edge in two dimensions or triangle in three dimensions) formed by the vertices of F. Furthermore, we denote by $\boldsymbol{s} = (s_1, \ldots, s_{d-1})$ a Cartesian coordinate system attached to E. We further assume that the interface Γ is exactly represented by a function $g(\boldsymbol{s}) \in C^{n+1}(E)$. Then, the face F can be considered as an interpolating polynomial $p_n(\boldsymbol{s})$ of degree n, which approximates $g(\boldsymbol{s})$ on E. It follows from the standard interpolation theory [25] that

$$|g(s) - p_n(s)| \simeq O\left(h_E^{n+1}\right), \tag{33}$$

where $h_E = O(h)$ denotes the length of E in two dimensions or the length of the longest edge of E in three dimensions. We thus have

$$\int_E |g(\boldsymbol{s}) - p_n(\boldsymbol{s})| \simeq O\left(h^{n+d}\right), \tag{34}$$

because $m(E) \simeq O(h^{d-1})$, where $m(E)$ denotes the measure of E. Because we assume that there are $O(1/h^{d-1})$ elements near the interface Γ, we obtain

$$\text{volume}(A_h) \simeq O\left(h^{n+1}\right). \tag{35}$$

We see that the volume of A_h converges with order $n + 1$ in general.

Furthermore, because u_h converges optimally to u in the region $\Omega \setminus A_h$, we have

$$\int_{\Omega \setminus A_h} (u - u_h)^2 \simeq O\left(h^{2(k+1)}\right). \tag{36}$$

As a result, for general curved interface problems with $O(1)$ jump magnitude, we obtain the following convergence rate

$$\|u - u_h\|_\Omega \simeq O\left(h^{\min(k+1,(n+1)/2)}\right). \tag{37}$$

This implies that u_h will converge with order $(k + 1)/2$ if isoparametric elements are used to represent the interface provided that the solution is not smooth across the interface. (Of course, if either the volume of A_h is zero or the amount of the jump s_D is zero, then the approximate solution u_h will converge with the optimal order $k + 1$ for isoparametric elements.) To recover the optimal convergence rate of $k + 1$ for u_h, we need to use polynomials of degree $2k + 1$ to represent (superparametric) elements near the interface.

5. NUMERICAL EXAMPLES

In this section, we present several numerical examples to demonstrate the performance of the proposed method. We use the same quadrature rules for both superparametric and isoparametric elements. More specifically, we need to use the number of quadrature points that yields an accuracy of at least order $O(h^{k+2})$. Hence, in practice, we choose $2k + 2$ integration points to achieve this goal.

DOI: 10.1002/nme

5.1. Linear interface

We solve the Poisson equation (1) with $\nu = 1$, and Γ is a star-shaped interface as shown in Figure 3. The jump conditions across the interface are given as

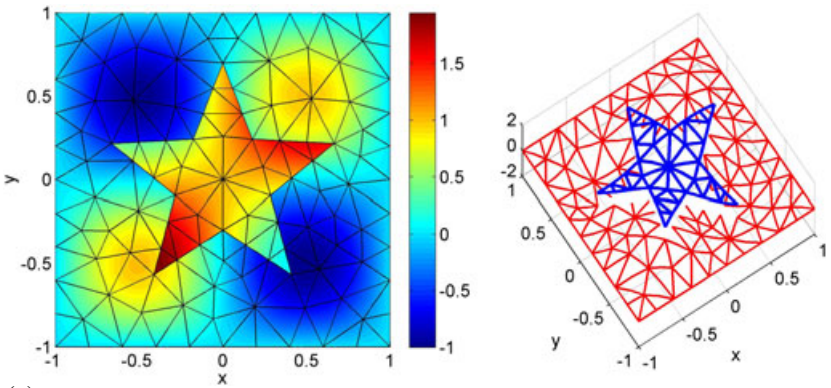
$$u|_{\Gamma^1} - u|_{\Gamma^2} = 1, \text{ on } \Gamma,$$
$$-\left(\nabla u|_{\Gamma^1} \cdot \boldsymbol{n}^1 + \nabla u|_{\Gamma^2} \cdot \boldsymbol{n}^2\right) = 0, \text{ on } \Gamma.$$

Homogeneous Dirichlet-type boundary condition is imposed on $\partial\Omega$. The source term is chosen such that the problem has the following analytical solution

$$u = \begin{cases} \sin(\pi x)\sin(\pi y) + 1 & \text{if } \boldsymbol{x} \in \Omega^1, \\ \sin(\pi x)\sin(\pi y) & \text{if } \boldsymbol{x} \in \Omega^2. \end{cases}$$

To study the convergence of the HDG method, we compute the numerical solution on a sequence of refined meshes. The initial mesh is shown in Figure 3. Each refinement is obtained by subdividing each triangle into four smaller triangles. We say that the mesh has level N_{ref}. It is obtained from the initial mesh by N_{ref} of these refinements. On these meshes, we consider polynomials of degree k to represent all the approximate variables using a nodal basis within each element [26]. We note that the interface is exactly represented by our finite element meshes.

Figure 3 shows the computed solution on the initial mesh. We further present in Table I the order of convergence and error in the approximate solution u_h, approximate gradient $\boldsymbol{q}_h$, and postprocessed solution u_h^*. We observe that the approximate solution and gradient converge optimally with



(a) u_h (b) $\widehat{u}_h$

Figure 3. (a) Numerical solution u_h and (b) numerical trace $\widehat{u}_h$ computed on the initial mesh in Example 5.1 of the linear interface.

Table I. Convergence history of the numerical approximations for Example 5.1 of the linear interface. The initial mesh is refined by $N_{\text{ref}} = 0, 1, 2$ times.

k	N_{ref}	$\| u - u_h \|_\Omega$	Order	$\| \boldsymbol{q} - \boldsymbol{q}_h \|_\Omega$	Order	$\| u - u_h^* \|_\Omega$	Order
1	0	7.27×10^{-02}	–	1.29×10^{-01}	–	3.99×10^{-03}	–
	1	1.84×10^{-02}	1.98	3.26×10^{-02}	1.98	4.89×10^{-04}	3.03
	2	4.63×10^{-04}	1.99	8.19×10^{-03}	1.99	6.02×10^{-05}	3.02
2	0	5.63×10^{-03}	–	1.15×10^{-02}	–	2.76×10^{-04}	–
	1	7.20×10^{-04}	2.97	1.44×10^{-03}	3.00	1.70×10^{-05}	4.02
	2	9.06×10^{-05}	2.99	1.81×10^{-04}	2.99	1.05×10^{-06}	4.02
3	0	3.75×10^{-04}	–	7.19×10^{-04}	–	1.35×10^{-05}	–
	1	2.37×10^{-05}	3.98	4.54×10^{-05}	3.99	4.21×10^{-07}	5.00
	2	1.48×10^{-06}	4.00	2.84×10^{-06}	4.00	1.31×10^{-08}	5.01

DOI: 10.1002/nme

order $k+1$. Moreover, the postprocessed solution converges with order $k+2$ that is one-order higher than the convergence rate of the original solution. This is one important advantage of the HDG method for interface problems.

5.2. *Crossing of the interface and the boundary*

In this example, we consider a Poisson problem with a straight interface cutting the boundary as shown in Figure 4(a). The interface divides the entire square domain $\Omega = [0,1]\times[0,1]$ into Ω^1 and Ω^2. The level set function of the interface is $\phi = x - 0.75$. Dirichlet and Neumann conditions on the boundary are given as follows.

$$\begin{aligned} u &= 0, && \text{on } \partial\Omega_{\mathrm{D}},\\ -\nu\nabla u\cdot\boldsymbol{n} &= -2\pi\sin(2\pi x) - 2x, && \text{on } \partial\Omega_{\mathrm{N}_1},\\ -\nu\nabla u\cdot\boldsymbol{n} &= -2\pi\sin(2\pi x), && \text{on } \partial\Omega_{\mathrm{N}_2},\\ -\nu\nabla u\cdot\boldsymbol{n} &= -2\pi\sin(2\pi y), && \text{on } \partial\Omega_{\mathrm{N}_3}, \end{aligned}$$

where the material property $\nu = 1$. The jump conditions on the interface are given as follows.

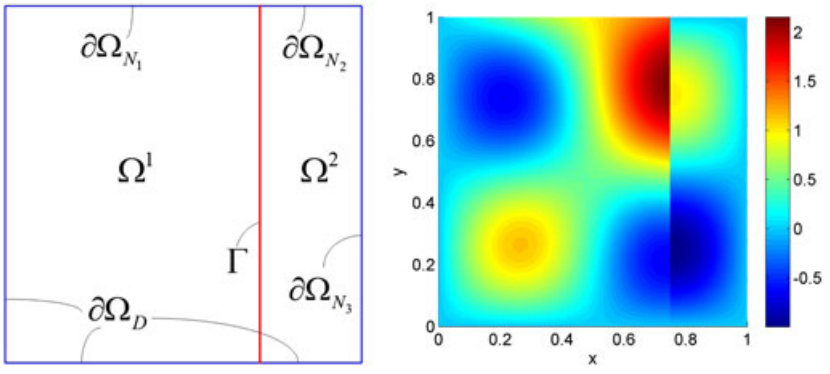
$$\begin{aligned} u|_{\Gamma^1} - u|_{\Gamma^2} &= 2xy, \quad \text{on } \Gamma,\\ -\left(\nabla u|_{\Gamma^1}\cdot\boldsymbol{n}^1 + \nabla u|_{\Gamma^2}\cdot\boldsymbol{n}^2\right) &= 2y, \quad \text{on } \Gamma. \end{aligned}$$

The numerical solution, as shown in the figure, will be compared with the following expression

$$u = \begin{cases} \sin(2\pi x)\sin(2\pi y) + 2xy & \text{if } \boldsymbol{x}\in\Omega^1,\\ \sin(2\pi x)\sin(2\pi y) & \text{if } \boldsymbol{x}\in\Omega^2. \end{cases}$$

Similar to Example 5.1, the convergence of the HDG method is estimated via a sequence of the refined meshes. Figures 4(b) and 5 plot the approximate solution u_h and the approximate gradient $\boldsymbol{q}_h$, respectively. Table II shows the optimal convergence rates for u_h and $\boldsymbol{q}_h$ and the superconvergence rate for u_h^*. The complexity at the intersection area between the interface and the boundary is resolved by the proposed approach without any extra modificiation.

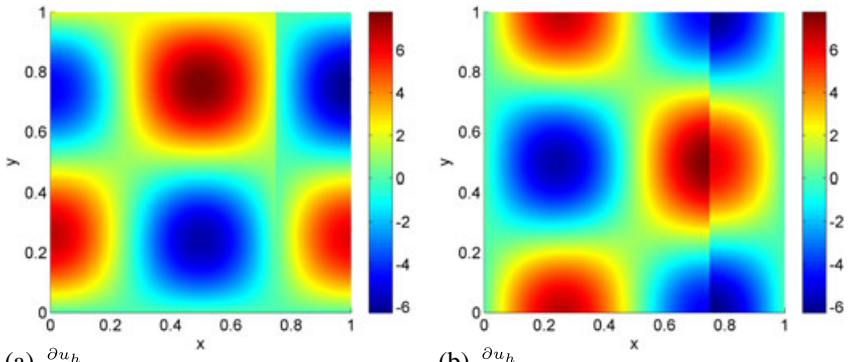
We note that the solution across the interface is nonsmooth in Examples 5.1 and 5.2. However, the use of superparametric elements at the interface is not compulsary to maintain the optimal convergence properties of the HDG method because the interface is made of straight lines.



(a) Domain (b) u_h

Figure 4. (a) Geometry arrangement and (b) the solution u_h computed on the initial mesh $h = 0.125$ with $k = 3$ in Example 5.2.

DOI: 10.1002/nme



(a) $\frac{\partial u_h}{\partial x}$ (b) $\frac{\partial u_h}{\partial y}$

Figure 5. The approximate flux $\boldsymbol{q}_h$ computed on the initial mesh $h = 0.125$ with $k = 3$ in Example 5.2. (a) $\partial u_h \,/\, \partial x$ and (b) $\partial u_h \,/\, \partial y$.

Table II. Convergence history of the numerical approximations for Example 5.2. The initial mesh is refined by $N_{\text{ref}} = 0, 1, 2$ times.

k	N_{ref}	$\| u - u_h \|_\Omega$	Order	$\| \boldsymbol{q} - \boldsymbol{q}_h \|_\Omega$	Order	$\| u - u_h^* \|_\Omega$	Order
1	0	9.21×10^{-02}	–	1.96×10^{-01}	–	3.10×10^{-03}	–
	1	2.38×10^{-02}	1.95	4.96×10^{-02}	1.98	3.75×10^{-04}	3.05
	2	6.01×10^{-03}	1.99	1.24×10^{-02}	2.00	4.59×10^{-05}	3.03
2	0	9.45×10^{-03}	–	2.17×10^{-02}	–	2.61×10^{-04}	–
	1	1.21×10^{-03}	2.97	2.74×10^{-03}	2.99	1.63×10^{-05}	4.00
	2	1.53×10^{-04}	2.98	3.44×10^{-04}	2.99	1.02×10^{-06}	4.00
3	0	7.92×10^{-04}	–	1.88×10^{-03}	–	1.64×10^{-05}	–
	1	5.07×10^{-05}	3.97	1.19×10^{-04}	3.98	5.08×10^{-07}	5.01
	2	3.19×10^{-06}	3.99	7.46×10^{-06}	4.00	1.58×10^{-08}	5.01

5.3. Dual thermal conductivity problem

In this example, we simulate the heat distribution at steady state over a plate made of two types of materials with different thermal conductivities. The governing equation reads

$$-\nabla \cdot (\nu \nabla u) = s, \quad \boldsymbol{x} \in [-1, 1] \times [-1, 1], \quad \nu = \begin{cases} 1 & \text{if } \boldsymbol{x} \in \Omega^1 \\ 100 & \text{if } \boldsymbol{x} \in \Omega^2 \end{cases}, \tag{38}$$

where the source term s is given by

$$s = -10\left(x_1^2 + x_2^2\right)^{3/2} - 15x_1^2\left(x_1^2 + x_2^2\right)^{1/2} - 15x_2^2\left(x_1^2 + x_2^2\right)^{1/2}, \quad \text{in } \Omega.$$

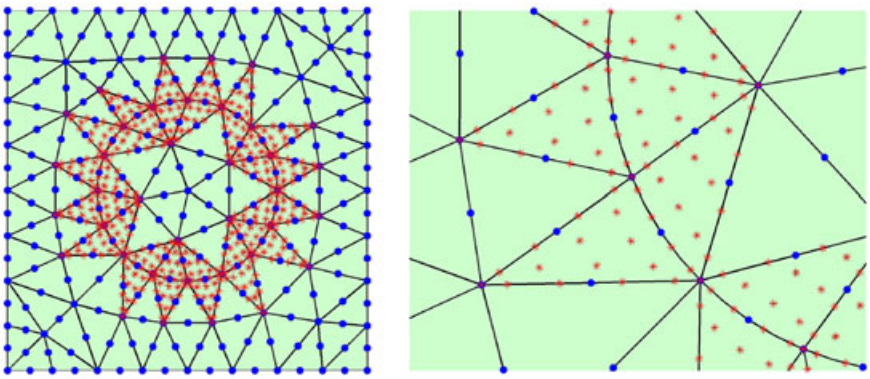
The circular interface Γ of radius $R = 0.5$ divides the entire domain $\Omega = [-1, 1] \times [-1, 1]$ into two separate regions Ω^1 and Ω^2 as shown in Figure 6. The numerical solution is compared with the following analytical solution

$$u = \begin{cases} \frac{1}{\nu_1}\left(x_1^2 + x_2^2\right)^{5/2} & \text{if } \boldsymbol{x} \in \Omega^1, \\ \frac{1}{\nu_2}\left(x_1^2 + x_2^2\right)^{5/2} + \left(\frac{1}{\nu_1} - \frac{1}{\nu_2}\right) R^5 & \text{if } \boldsymbol{x} \in \Omega^2, \end{cases} \tag{39}$$

where $\nu_1 = 1$ and $\nu_2 = 100$. Dirichlet boundary condition on the outer boundary $\partial\Omega$ is derived from (39). The jump in the solution as well as the jump in the flux are prescribed as follows

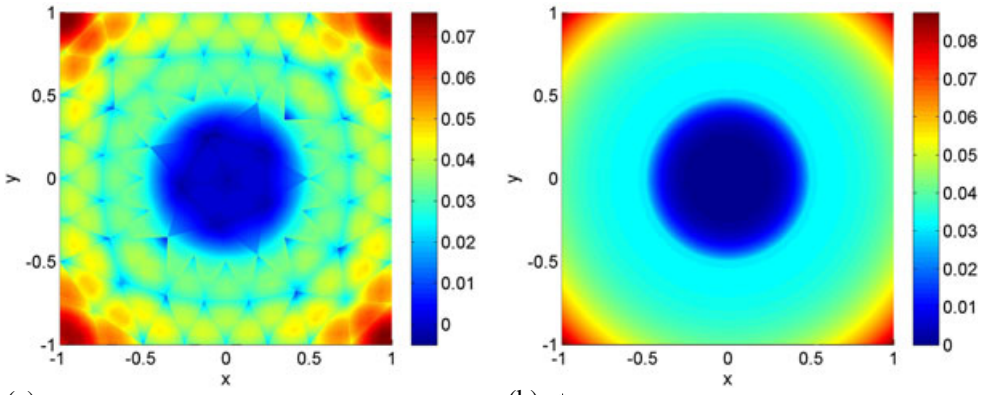
$$u|_{\Gamma^1} - u|_{\Gamma^2} = 0, \text{ on } \Gamma,$$
$$-\left(\nu_1 \nabla u|_{\Gamma^1} \cdot \boldsymbol{n}^1 + \nu_2 \nabla u|_{\Gamma^2} \cdot \boldsymbol{n}^2\right) = 0, \text{ on } \Gamma.$$

DOI: 10.1002/nme



(a) Entire mesh (b) Elements near the interface

Figure 6. The initial mesh uses superparametric elements near the interface and isoparametric elements away from the interface in Example 5.3 of the dual thermal conductivities.



(a) u_h (b) u_h^*

Figure 7. The original (a) u_h and the postprocessed solutions (b) u_h^* computed on the initial mesh for $k = 2$ in Example 5.3 of the dual thermal conductivities.

We compute the numerical solution on a sequence of refined meshes that is similar to Example 5.1. The initial mesh is shown in Figure 6(a). On these meshes, we consider polynomials of degree k to represent all the approximate variables using a nodal basis within each element [26]. However, we use polynomials of degree $2k + 1$ to represent the elements near the interface as shown in Figure 6.

Figure 7 plots the solution u_h and the postprocessed solution u_h^* on the initial mesh for $k = 2$. The solution is continuous but nonsmooth across the circular interface because of different values of ν. We see that u_h^* is significantly superior to u_h as depicted in Figure 7. This demonstrates the effectiveness of the local postprocessing for interface problems. We further present in Table III the order of convergence and error in the approximate solution u_h, approximate gradient $\boldsymbol{q}_h$, and postprocessed solution u_h^*. We observe again that the approximate solution and gradient converge optimally with order $k + 1$ and that the postprocessed solution converges with order $k + 2$.

5.4. Circular interface with non-zero jump conditions

We consider solving a Poisson equation ($\nu = 1$) with a circular interface of radius $R = 0.5$ immersed inside Ω that is a square $[-1, 1] \times [-1, 1]$. Jump boundary conditions across Γ are given as

DOI: 10.1002/nme

Table III. Convergence history of the numerical approximations for Example 5.3 of the dual thermal conductivities. Superparametric elements with the order of $(2k+1)$ are used to approximate the interface.

k	N_{ref}	$\|u-u_h\|_\Omega$	Order	$\|\boldsymbol{q}-\boldsymbol{q}_h\|_\Omega$	Order	$\|u_h^*-u\|_\Omega$	Order
1	0	1.44×10^{-01}	–	1.62×10^{-02}	–	6.31×10^{-04}	–
	1	3.65×10^{-02}	1.98	4.65×10^{-03}	1.80	8.95×10^{-05}	2.82
	2	9.16×10^{-03}	2.00	1.22×10^{-03}	1.93	1.18×10^{-05}	2.92
2	0	8.63×10^{-03}	–	2.22×10^{-03}	–	6.17×10^{-05}	–
	1	1.08×10^{-03}	3.00	3.04×10^{-04}	2.86	4.33×10^{-06}	3.83
	2	1.35×10^{-04}	3.00	3.88×10^{-05}	2.97	2.77×10^{-07}	3.97
3	0	2.42×10^{-04}	–	1.62×10^{-04}	–	3.82×10^{-06}	–
	1	1.48×10^{-05}	4.03	1.06×10^{-05}	3.93	1.26×10^{-07}	4.92
	2	9.14×10^{-07}	4.02	6.67×10^{-07}	4.00	3.98×10^{-09}	4.99

Table IV. Convergence history of the numerical approximations for Example 5.4 of the circular interface with $k=1$ for the isoparametric $k^*=1$ and the superparametric cases $k^*=2$.

N_{ref}	$\|u-u_h\|_\Omega$	Order	$\|\boldsymbol{q}-\boldsymbol{q}_h\|_\Omega$	Order
		Isoparametric case		
0	3.51×10^{-01}	–	8.23×10^{-01}	–
1	1.68×10^{-01}	1.06	3.91×10^{-01}	1.07
2	7.40×10^{-02}	1.18	1.75×10^{-01}	1.16
		Superparametric case		
0	1.50×10^{-01}	–	2.77×10^{-01}	–
1	3.89×10^{-02}	1.95	6.77×10^{-02}	2.03
2	9.80×10^{-03}	1.99	1.66×10^{-02}	2.03

$$
\begin{aligned}
u|_{\Gamma^1}-u|_{\Gamma^2} &= \sin(\pi x)\sin(\pi y)-e^x\cos y,\\
-\left(\nabla u|_{\Gamma^1}\cdot\boldsymbol{n}^1+\nabla u|_{\Gamma^2}\cdot\boldsymbol{n}^2\right) &= 2x(\pi\cos(\pi x)\sin(\pi y)\\
&\quad -e^x\cos y)+2y(\pi\sin(\pi x)\cos(\pi y)-e^x\sin y).
\end{aligned}
$$

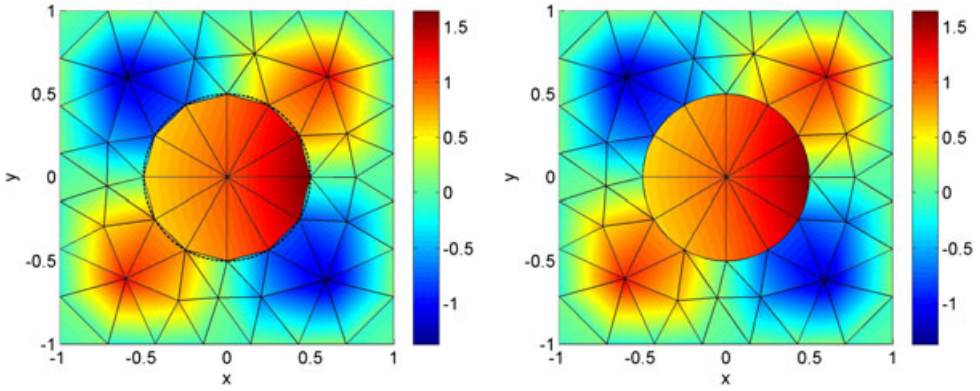
Dirichlet boundary conditions along the outside rectangular domain $\partial\Omega$ and the source term are derived from the following analytical expression

$$
u=\begin{cases} e^x\cos y & \text{if } \boldsymbol{x}\in\Omega^1,\\ \sin(\pi x)\sin(\pi y) & \text{if } \boldsymbol{x}\in\Omega^2.\end{cases}
$$

With this example, we aim to demonstrate the benefit of using superparametric elements for curved interfaces.

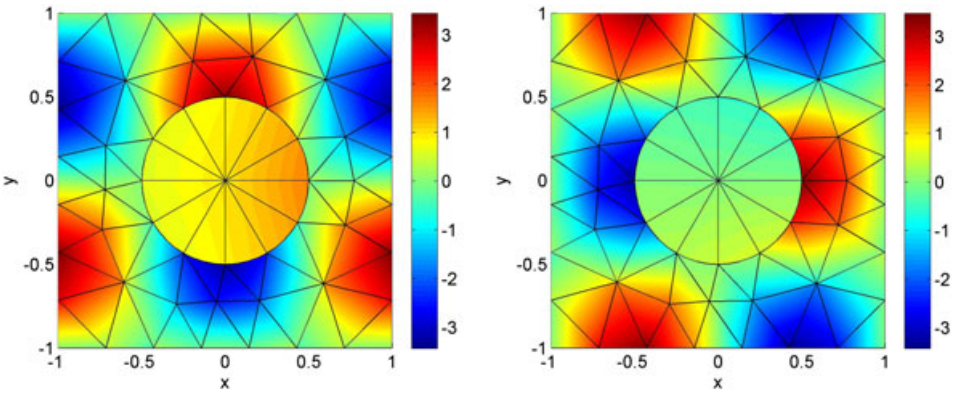
We present in Table IV the convergence rate and error for the numerical approximations computed using $k=1$ for both isoparametric $k^*=1$ and superparametric elements $k^*=2$ for the interface approximation. We see that when isoparametric elements are applied along the circular interface, both u_h and $\boldsymbol{q}_h$ converge suboptimally with order one, whereas they converge optimally with order $k+1$ when superparametric elements are used along the interface. Note that using $k^*=2$ suffices to guarantee optimal convergence because the interface is circular. Figure 8 depicts the approximate solution u_h in both cases. Note that the black-dashed line in Figure 8(a) represents the exact circular interface. Because of the effect of the jump in the solution, using isoparametric elements leads to suboptimal convergence rates. By using superparametric elements, we can recover optimal

DOI: 10.1002/nme



(a) Iso-parametric elements $k^* = 1$ (b) Super-parametric elements $k^* = 2$

Figure 8. The approximate solution u_h computed using (a) isoparametric and (b) superparametric elements for elements near the circular interface in Example 5.4. These solutions are obtained on the initial mesh shown herein.



(a) $\frac{\partial u_h}{\partial x}$ (b) $\frac{\partial u_h}{\partial y}$

Figure 9. The approximate flux $\boldsymbol{q}_h$ computed using superparametric elements for elements near the circular interface in Example 5.4. (a) $\partial u_h \,/\, \partial x$ and (b) $\partial u_h \,/\, \partial y$.

Table V. Convergence history of the numerical approximations for Example 5.4 of the circular interface on using superparametric elements $k^* = k+1$. The original mesh size $h = 0.3$ is reduced by $N_{\text{ref}} = 0, 1, 2$ times.

k	N_{ref}	$\| u - u_h \|_\Omega$	Order	$\| \boldsymbol{q} - \boldsymbol{q}_h \|_\Omega$	Order	$\|u_h^* - u\|_\Omega$	Order
1	0	1.50×10^{-01}	–	2.77×10^{-01}	–	1.50×10^{-02}	–
	1	3.89×10^{-02}	1.95	6.77×10^{-02}	2.03	1.64×10^{-03}	3.19
	2	9.80×10^{-03}	1.99	1.66×10^{-02}	2.03	1.92×10^{-04}	3.09
2	0	1.89×10^{-02}	–	3.60×10^{-02}	–	1.03×10^{-03}	–
	1	2.29×10^{-03}	3.04	4.24×10^{-03}	3.09	5.89×10^{-05}	4.13
	2	2.83×10^{-04}	3.02	5.16×10^{-04}	3.04	3.52×10^{-06}	4.06
3	0	1.73×10^{-03}	–	4.02×10^{-03}	–	8.73×10^{-05}	–
	1	1.12×10^{-04}	3.95	2.28×10^{-04}	4.14	2.40×10^{-06}	5.18
	2	6.95×10^{-06}	4.01	1.33×10^{-05}	4.10	6.93×10^{-08}	5.11

convergence rates for both the approximate solution and gradient. Figure 9 illustrates the approximate flux $\boldsymbol{q}_h$ obtained using superparametric elements. Table V shows optimal convergence rates of u_h and $\boldsymbol{q}_h$ for higher order elements with $k = \{1, 2, 3\}$.

DOI: 10.1002/nme

5.5. Kidney-shaped interface

In the last example, we consider a kidney-shaped interface governed by the following level set function

$$\phi = \left(3\left((x+0.5)^2+y^2\right)-x-0.5\right)^2-\left((x+0.5)^2+y^2\right)+0.1.$$

The Dirichlet boundary data and source term are derived from the following exact solution

$$u = \begin{cases} \sin(2x^2+y^2+2)+x & \text{if } \boldsymbol{x}\in\Omega^1, \\ \frac{1}{\nu_2}\cos(1-x^2-y^2) & \text{if } \boldsymbol{x}\in\Omega^2. \end{cases}$$

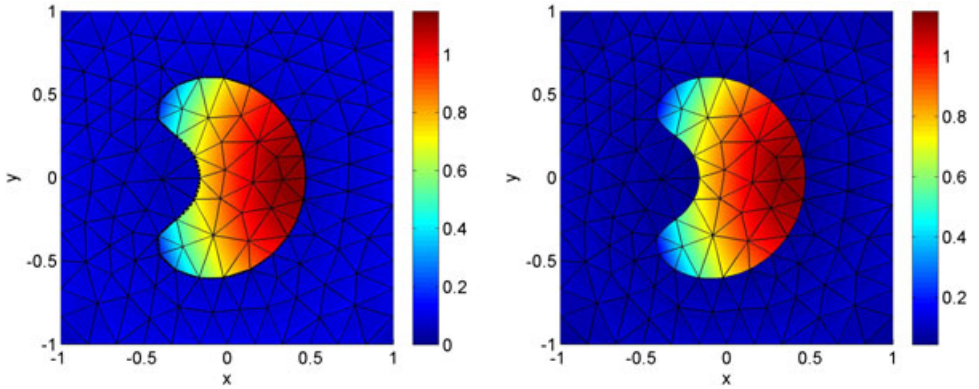
Note that the diffusion coefficient ν is equal to $\nu_1 = 1$ in Ω^1 and equal to $\nu_2 = 10$ in Ω^2. Here, $\overline{\Omega} = [-1,1]^2 \equiv \overline{\Omega}_1 \cup \overline{\Omega}_2$, where the subdomain Ω^1 is the region closed by the aforementioned level set function.

Table VI. Convergence history of the numerical approximations for Example 5.5 of the kidney-shaped interface on using isoparametric elements.

k	N_{ref}	$\| u-u_h \|_\Omega$	Order	$\| \boldsymbol{q}-\boldsymbol{q}_h \|_\Omega$	Order
1	0	4.05×10^{-02}	–	2.41×10^{-01}	–
	1	2.36×10^{-02}	0.78	1.30×10^{-01}	0.89
	2	1.19×10^{-02}	0.99	6.66×10^{-02}	0.96

Table VII. Convergence history of the numerical approximations for Example 5.5 of the kidney-shaped interface on using superparametric elements $k^* = k+1$.

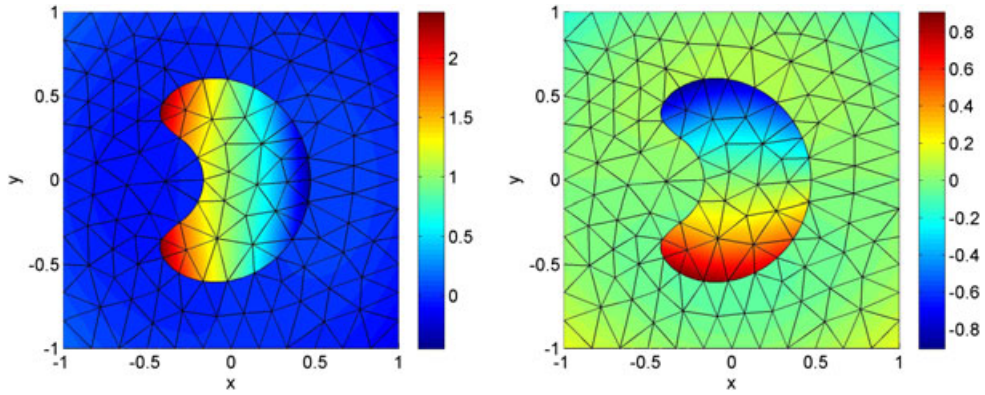
k	N_{ref}	$\| u-u_h \|_\Omega$	Order	$\| \boldsymbol{q}-\boldsymbol{q}_h \|_\Omega$	Order	$\|u_h^*-u\|_\Omega$	Order
1	0	1.66×10^{-02}	–	1.17×10^{-02}	–	2.80×10^{-04}	–
	1	4.06×10^{-03}	2.03	2.68×10^{-03}	2.13	3.36×10^{-05}	3.06
	2	1.00×10^{-03}	2.02	6.20×10^{-04}	2.11	4.11×10^{-06}	3.03
2	0	8.54×10^{-04}	–	1.87×10^{-03}	–	2.41×10^{-05}	–
	1	1.08×10^{-04}	2.98	1.88×10^{-04}	3.31	1.56×10^{-06}	3.95
	2	1.22×10^{-05}	3.15	1.88×10^{-05}	3.32	9.42×10^{-08}	4.05
3	0	1.50×10^{-04}	–	3.30×10^{-04}	–	5.24×10^{-06}	–
	1	8.54×10^{-06}	4.13	2.24×10^{-05}	3.88	1.90×10^{-07}	4.79
	2	4.18×10^{-07}	4.35	1.16×10^{-06}	4.27	4.57×10^{-09}	5.38



(a) Iso-parametric elements $k^* = 1$ (b) Super-parametric elements $k^* = 2$

Figure 10. The approximate solution u_h computed using (a)isoparametric and (b) superparametric elements for elements near the kidney-shaped interface in Example 5.5. These solutions are obtained on the initial mesh shown herein.

DOI: 10.1002/nme



(a) $\frac{\partial u_h}{\partial x}$ (b) $\frac{\partial u_h}{\partial y}$

Figure 11. The approximate flux $\boldsymbol{q}_h$ computed using superparametric elements for elements near the kidney-shaped interface in Example 5.5. (a) $\partial u_h / \partial x$ and (b) $\partial u_h / \partial y$.

We present numerical results in Table VI for the case where isoparametric elements $k^* = k$ are used along the interface and in Table VII for the case where superparametric elements $k^* = k+1$ are used along the interface. The results are similar to those presented in the previous example. Figure 10 shows the approximate solution u_h in both cases, whereas Figure 11 illustrates the approximate flux $\boldsymbol{q}_h$ obtained using superparametric elements.

6. CONCLUSION

We have developed an HDG method for numerically solving elliptic interface problems. All the discontinuous terms appear on the right-hand side vector of the global matrix system, and thus, the positive definite properties of the stiffness matrix are preserved. The use of superparametric elements is essential to remedy the numerical errors arising from the inexact approximation of the interface geometry and the nonsmoothness of the solution across the interface. The order $k^* = 2k + 1$ of superparametric elements is recommended for elliptic problems with the finite jump in the solution across the high curvature interface. Numerical results have shown that the method provides optimal convergence of order $k + 1$ for both the approximate solution and gradient. Furthermore, the post-processed solution converges with order $k + 2$. The extension of this work to the incompressible Stokes and Navier–Stokes equations constitutes the subject of ongoing research.

ACKNOWLEDGEMENTS

We are grateful to Professor Bernardo Cockburn of UMN for his fruitful discussion on this work. We would like to thank the Singapore-MIT Alliance for funding this work.

REFERENCES

1. Leveque RJ, Li Z. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM Journal on Numerical Analysis* 1994; **31**(4):1019–1044.
2. Leveque RJ, Li Z. Immersed interface methods for Stokes flow with elastic boundaries or surface tension. *SIAM Journal on Scientific Computing* 1997; **18**:709–735.
3. Li Z, Ito K. The immersed interface method: numerical solutions of PDEs involving interfaces and irregular domains. *Frontiers in Applied Mathematics.* SIAM, Philadelphia 2006. ISBN: 978-0-898-71609-2.
4. Bramble JH, King JT. A finite element method for interface problems in domains with smooth boundaries and interfaces. *Advances in Computational Mathematics* 1996; **6**:109–138.
5. Hansbo A, Hansbo P. An unfitted finite element method, based on Nitsches method, for elliptic interface problems. *Computer Methods in Applied Mechanics and Engineering* 2002; **191**:5537–5552.
6. Dolbow J, Harari I. An efficient finite element method for embedded interface problems. *International Journal for Numerical Methods in Engineering* 2009; **78**:229–252.

DOI: 10.1002/nme

7. Guyomarc'h G, Lee CO, Jeon K. A discontinuous Galerkin methods for elliptic interface problems with application to electroporation. *Communications in Numerical Methods in Engineering* 2009; **25**:991–1008.
8. 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**:1319–1365.
9. Cockburn B, Dong B, Guzmán J. A superconvergent LDG-hybridizable Galerkin method for second-order elliptic problems. *Mathematics of Computation* 2008; **77**:1887–1916.
10. Cockburn B, Gopalakrishnan J, Sayas FJ. A projection-based error analysis of HDG methods. *Mathematics of Computation* 2010; **79**:1351–1367.
11. Cockburn B, Guzmán J, Wang H. Superconvergent discontinuous Galerkin methods for second-order elliptic problems. *Mathematics of Computation* 2009; **78**:1–24.
12. Tan Z, Lim KM, Khoo BC. An immersed interface method for Stokes flows with fixed/moving interfaces and rigid boundaries. *Journal of Computational Physics* 2009; **228**:6855–6881.
13. Tan Z, Le DV, Lim KM, Khoo BC. An immersed interface method for the incompressible Navier–Stokes equations with discontinuous viscosity across the interfaces. *SIAM Journal on Scientific Computing* 2009; **31**:1798–1819.
14. Cockburn B, Nguyen NC, Peraire J. A comparison of HDG methods for Stokes flow. *Journal of Scientific Computing* 2010; **45**:215–237.
15. Cockburn B, Gopalakrishnan J, Nguyen NC, Peraire J, Sayas FJ. Analysis of HDG methods for Stokes flow. *Mathematics of Computation* 2011; **80**:723–760.
16. Nguyen NC, Peraire J, Cockburn B. A hybridizable discontinuous Galerkin method for Stokes flow. *Computer Methods in Applied Mechanics and Engineering* 2009; **199**:582–597.
17. Nguyen NC, Peraire J, Cockburn B. A hybridizable discontinuous Galerkin method for the incompressible Navier–Stokes equations. *Journal of Computational Physics* 2011; **230**:1147–1170.
18. Nguyen NC, Peraire J, Cockburn B. An implicit high-order hybridazable discontinuous Galerkin method for linear convection–diffusion equations. *Journal of Computational Physics* 2009; **228**:3232–3254.
19. Nguyen NC, Peraire J, Cockburn B. An implicit high-order hybridizable discontinuous Galerkin method for nonlinear convection–diffusion equations. *Journal of Computational Physics* 2009; **228**:8841–8855.
20. Labeur RJ, Wells GN. A Galerkin interface stabilisation method for advection–diffusion and incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 2007; **196**:4985–5000.
21. Labeur RJ, Wells GN. Interface stabilised finite element method for moving domains and free surface flows. *Computer Methods in Applied Mechanics and Engineering* 2009; **198**:615–630.
22. Arnold DN, Brezzi F, Cockburn B, Marini LD. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM Journal on Numerical Analysis* 2001; **39**:1749–1779.
23. Huynh LNT. Hybridizable discontinuous Galerkin method and fast solver FFT for multi-viscosity incompressible Navier–Stokes flows on irregular domains. *Ph.D Dissertation*, Singapore-MIT Alliance, National University of Singapore, 2010.
24. Stenberg R. Some new families of finite elements for the Stokes equations. *Numerische Mathematik* 1990; **56**:827–838.
25. Phillips GM. *Interpolation and Approximation by Polynomials.* CMS Books in Mathematics. Springer: New York, 2003. ISBN: 978-0-387-00215-6.
26. Hesthaven JS, Warburton T. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*. Springer: New York, 2008. ISBN: 978-0-387-72065-4.

DOI: 10.1002/nme