Journal of Scientific Computing (2024) 100:44
https://doi.org/10.1007/s10915-024-02604-3

Hybridizable Discontinuous Galerkin Methods for the Two-Dimensional Monge–Ampère Equation

Ngoc Cuong Nguyen[1] · **Jaime Peraire**[1]

Received: 3 January 2024 / Revised: 13 June 2024 / Accepted: 17 June 2024 /
Published online: 27 June 2024

Abstract
We introduce two hybridizable discontinuous Galerkin (HDG) methods for numerically solving the two-dimensional Monge–Ampère equation. The first HDG method is devised to solve the nonlinear elliptic Monge–Ampère equation by using Newton's method. The second HDG method is devised to solve a sequence of the Poisson equation until convergence to a fixed-point solution of the Monge–Ampère equation is reached. Numerical examples are presented to demonstrate the convergence and accuracy of the HDG methods. Furthermore, the HDG methods are applied to r-adaptive mesh generation by redistributing a given scalar density function via the optimal transport theory. This r-adaptivity methodology leads to the Monge–Ampère equation with a nonlinear Neumann boundary condition arising from the optimal transport of the density function to conform the resulting high-order mesh to the boundary. Hence, we extend the HDG methods to treat the nonlinear Neumann boundary condition. Numerical experiments are presented to illustrate the generation of r-adaptive high-order meshes on planar and curved domains.

Keywords Monge–Ampère equation · Hybridizable discontinuous Galerkin methods · Grid adaptivity · r-adaptivity · Elliptic equations

1 Introduction

The Monge–Ampère equation has its root in optimal transport theory and arises from many areas in science and engineering such as astrophysics, differential geometry, geostrophic fluid dynamics, image processing, mesh generation, optimal transportation, statistical inference, and stochastic control; see [1] and references therein. The equation belongs to a class of fully nonlinear second-order elliptic partial differential equations (PDEs). Due to the wide range of above-mentioned applications, the Monge–Ampère equation has attracted significant attention from mathematicians and scientists [2–12]. Significant progress has been made in the development of numerical methods for solving the Monge–Ampère equation [1, 4, 5, 7, 9,

✉ Ngoc Cuong Nguyen
cuongng@mit.edu

[1] Department of Aeronautics and Astronautics, Center for Computational Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA

13–25]. A number of finite element methods have been proposed for the Monge–Ampère equation. In [7], Dean and Glowinski presented an augmented Lagrange multiplier method and a least squares method for the Monge–Ampère equation by treating the nonlinear equations as a constraint and using finite elements. Böhmer [14] introduced a projection method using C^1 finite element functions for a class of fully nonlinear second order elliptic PDEs and analyzed the method using consistency and stability arguments. In [15], Brenner et al. proposed C^0 finite element methods and discontinuous Galerkin (DG) methods for the Monge–Ampère equation, which were extended to the three dimensional Monge–Ampère equation [16]. In [4], Benamou et al. proposed a fixed-point method that only requires the solution of a sequence of Poisson equations for the two-dimensional Monge–Ampère equation. In [25], a linearization-then-descretization approach consists in applying Newton's method to a nonlinear elliptic PDE to produce a sequence of linear nonvariational elliptic PDEs that can be dealt with using nonvariational finite element method. In [20], Feng and Lewis developed mixed interior penalty DG methods for fully nonlinear second order elliptic and parabolic equations. Recently, a finite element/operator-splitting method is introduced for the Monge–Ampère equation [24, 26] by using an equivalent divergence formulation of the Monge–Ampère equation through the cofactor matrix of the Hessian of the solution.

In this paper, two hybridizable discontinuous Galerkin (HDG) methods are considered for the numerical solution of the Monge–Ampère equation in two dimensions. The first HDG method is devised to solve the Monge–Ampère equation by using Newton's method. The second HDG method is devised to solve a sequence of the Poisson equation until convergence to a fixed-point solution of the Monge–Ampère equation is reached. Numerical examples are presented to compare the convergence and accuracy of the HDG methods. It is found out that while the two methods yield similar orders of accuracy for the numerical solution, the Newton-HDG method requires considerably less number of linear solves than the fixed-point HDG method. However, an advantage of the fixed-point HDG method is that the discretization of the Poisson equation results in linear systems that can be solved more efficiently than those resulting from the Newton-HDG method.

The HDG methods have some unique features which distinguish themselves from other finite element methods for the Monge–Ampère equation. First, the global degrees of freedom are those of the approximate trace of the scalar variable defined on element faces. This translates to computational efficiency for the solution of nonlinear and linear systems arising from the HDG discretization of the Monge–Ampère equation. Second, the approximate gradient and Hessian converge with the same order as the approximate scalar variable. For most other finite element methods, the convergence rates of the approximate gradient and Hessian are lower than that of the approximate scalar variable.

There has been considerable interest in r-adaptive mesh generation by the optimal transport theory via solving the Monge–Ampère equation [27–35]. In r-adaptivity, mesh points are neither created nor destroyed, data structures do not need to be modified in-place, and complicated load-balancing is not necessary [36]. Furthermore, the r-adaptivity approach based on the Monge–Ampère equation has the ability to avoid mesh entanglement and sharp changes in mesh resolution [28, 29]. This r-adaptivity approach has its root from the optimal transport theory since it seeks to minimize a deformation functional subject to equidistributing a given scalar monitor function which controls the local density of mesh points. If the functional is defined as the L_2 norm of the grid deformation, the optimal transport theory results in a mesh mapping as the gradient of the solution of the Monge–Ampère equation [27, 31]. To determine the mesh mapping we need to impose a boundary condition that mesh mapping conforms to the boundary of the physical domain. This boundary condition can be characterized as a Hamilton–Jacobi equation on the boundary [5] or a nonlinear Neumann

boundary condition [30]. We extend the HDG methods to the Monge–Ampère equation with nonlinear boundary conditions. The methods are used to generate r-adaptive high-order meshes based on an equidistribution of a density function via the optimal transport theory.

The paper is organized as follows. In Sect. 2, we introduce HDG methods for the Monge–Ampère equation and present numerical results to demonstrate their performance. In Sect. 3, we extend these HDG methods to r-adaptivity based on the optimal transport theory and present numerical experiments to illustrate the generation of r-adaptive high-order meshes. Finally, in Sect. 4, we make a number of concluding remarks on the results as well as future work.

2 The Hybridizable Discontinuous Galerkin Methods

HDG methods were first introduced in [37] for elliptic problems and subsequently extended to a wide variety of PDEs: linear convection-diffusion problems [38, 39], nonlinear convection-diffusion problems [40–42], Stokes problems [43–46], incompressible flows equations [47–53], compressible flows [54–61], Maxwell's equations [62–64], linear elasticity [65–69], and nonlinear elasticity [70–73]. To the best of our knowledge, however, HDG methods have not been considered for solving the Monge–Ampère equation prior to this work. In this section, we describe HDG methods for numerically solving the Monge–Ampère equation. The proposed HDG methods simultaneously compute approximations to the scalar variable, the gradient variables, and Hessian variables. The HDG methods are computationally efficient owing to a hybridization procedure that eliminates the degrees of freedom of those approximate variables to obtain global systems for the degrees of freedom of an approximate trace defined on the element faces.

2.1 Governing Equations and Approximation Spaces

We consider the Monge–Ampère equation with a Dirichlet boundary condition

$$\begin{aligned} \det(D^2u) &= f, \text{ in } \Omega, \\ u &= g, \text{ on } \partial\Omega. \end{aligned} \tag{1}$$

Here $\Omega \in \mathbb{R}^2$ is the physical domain with Lipschitz boundary $\partial\Omega$, the source term f is a strictly positive function, and g is a smooth function. Note that D^2u is the Hessian matrix of the exact solution u, and $\det(D^2u) = u_{xx}u_{yy} - u_{xy}^2$ is its determinant. The solution u must be convex in order for the equation to be elliptic. Without the convexity constraint, the equation does not have a unique solution. In two dimensions [4], the Monge–Ampère Eq. (1) can be rewritten as

$$\Delta u = \sqrt{u_{xx}^2 + u_{yy}^2 + 2u_{xy}^2 + 2f} \quad \text{in } \Omega. \tag{2}$$

We introduce $\boldsymbol{q} = \nabla u$ and $\boldsymbol{H} = \nabla \boldsymbol{q}$, and rewrite the above equation as a first-order system of equations

$$\begin{aligned} \boldsymbol{H} - \nabla \boldsymbol{q} &= 0, \text{ in } \Omega, \\ \boldsymbol{q} - \nabla u &= 0, \text{ in } \Omega, \\ s(\boldsymbol{H}, f) - \nabla \cdot \boldsymbol{q} &= 0, \text{ in } \Omega, \\ u - g &= 0, \text{ on } \partial\Omega, \end{aligned} \tag{3}$$

where $s(\boldsymbol{H}, f) = \sqrt{H_{11}^2 + H_{22}^2 + H_{12}^2 + H_{21}^2 + 2f}$. While we consider Dirichlet boundary condition in this section, we will treat Neumann boundary condition in the next section.

We denote by $\mathcal{T}_h$ a collection of disjoint regular elements K that partition Ω and set $\partial\mathcal{T}_h := \{\partial K : K \in \mathcal{T}_h\}$. For an element K of the collection $\mathcal{T}_h$, $e = \partial K \cap \partial\Omega$ is the boundary face if the $(d-1)$-Lebesgue measure of e is nonzero. For two elements K^+ and K^- of the collection $\mathcal{T}_h$, $e = \partial K^+ \cap \partial K^-$ is the interior face between K^+ and K^- if the $(d-1)$-Lebesgue measure of e is nonzero. Let $\mathcal{E}_h^i$ and $\mathcal{E}_h^\partial$ denote the set of interior and boundary faces, respectively. We denote by $\mathcal{E}_h$ the union of $\mathcal{E}_h^i$ and $\mathcal{E}_h^\partial$. Let $\boldsymbol{n}^+$ and $\boldsymbol{n}^-$ be the outward unit normal of ∂K^+ and ∂K^-, respectively. Let $\mathcal{P}^p(D)$ denote the set of polynomials of degree at most p on a domain D and let $L^2(D)$ be the space of square integrable functions on D. We introduce discontinuous finite element spaces

$$
\begin{aligned}
\boldsymbol{W}_h^p &= \{\boldsymbol{v} \in \boldsymbol{L}^2(\Omega) \;\; \boldsymbol{v}|_K \in (\mathcal{P}^p(K))^{2\times 2} \quad \forall K \in \mathcal{T}_h\},\\
\boldsymbol{V}_h^p &= \{\boldsymbol{v} \in \boldsymbol{L}^2(\Omega) \;\; \boldsymbol{v}|_K \in (\mathcal{P}^p(K))^{2} \quad \forall K \in \mathcal{T}_h\},\\
U_h^p &= \{w \in L^2(\Omega) \;\; w|_K \in \mathcal{P}^p(K) \quad \forall K \in \mathcal{T}_h\},\\
M_h^p &= \{\mu \in L^2(\mathcal{E}_h) \;\; \mu|_e \in \mathcal{P}^p(e), \quad \forall e \in \mathcal{E}_h\}.
\end{aligned}
$$

Note that M_h^p consists of functions which are continuous inside the faces (or edges) $e \in \mathcal{E}_h$ and discontinuous at their borders.

For functions u and v in $L^2(D)$, we denote $(u,v)_D = \int_D uv$ if D is a domain in $\mathbb{R}^d$ and $\langle u, v\rangle_D = \int_D uv$ if D is a domain in $\mathbb{R}^{d-1}$. The inner produces associated with the above approximation spaces are defined as follows

$$
\begin{aligned}
(u,v)_K &:= \int_K uv\,\mathrm{d}\Omega, & (u,v)_{\mathcal{T}_h} &:= \sum_{K\in\mathcal{T}_h} (u,v)_K, & \forall u, v \in L^2(\Omega),\\
(\boldsymbol{u},\boldsymbol{v})_K &:= \int_K \boldsymbol{u}\cdot\boldsymbol{v}\,\mathrm{d}\Omega, & (\boldsymbol{u},\boldsymbol{v})_{\mathcal{T}_h} &:= \sum_{K\in\mathcal{T}_h} (\boldsymbol{u},\boldsymbol{v})_K, & \forall \boldsymbol{u}, \boldsymbol{v} \in L^2(\Omega)^d,\\
(\boldsymbol{G},\boldsymbol{H})_K &:= \int_K \boldsymbol{G} : \boldsymbol{H}\,\mathrm{d}\Omega, & (\boldsymbol{G},\boldsymbol{H})_{\mathcal{T}_h} &:= \sum_{K\in\mathcal{T}_h} (\boldsymbol{G},\boldsymbol{H})_K, & \forall \boldsymbol{G}, \boldsymbol{H} \in L^2(\Omega)^{d\times d},\\
\langle \mu,\eta\rangle_{\partial K} &:= \int_{\partial K} \mu\eta\,\mathrm{d}\Gamma, & \langle \mu,\eta\rangle_{\partial\mathcal{T}_h} &:= \sum_{K\in\mathcal{T}_h} \langle \mu,\eta\rangle_{\partial K}, & \forall \mu, \eta \in L^2(\partial\mathcal{T}_h).
\end{aligned}
$$

We are ready to describe the HDG methods for the Monge–Ampère problem (3).

2.2 The Newton-HDG Formulation

To numerically solve (3) with the HDG method, we find $(\boldsymbol{H}_h, \boldsymbol{q}_h, u_h, \widehat{u}_h) \in \boldsymbol{W}_h^p \times \boldsymbol{V}_h^p \times U_h^p \times M_h^p$ such that

$$
\begin{aligned}
(\boldsymbol{H}_h, \boldsymbol{G})_{\mathcal{T}_h} + (\boldsymbol{q}_h, \nabla\cdot\boldsymbol{G})_{\mathcal{T}_h} - \langle \widehat{\boldsymbol{q}}_h, \boldsymbol{G}\cdot\boldsymbol{n}\rangle_{\partial\mathcal{T}_h} &= 0,\\
(\boldsymbol{q}_h, \boldsymbol{v})_{\mathcal{T}_h} + (u_h, \nabla\cdot\boldsymbol{v})_{\mathcal{T}_h} - \langle \widehat{u}_h, \boldsymbol{v}\cdot\boldsymbol{n}\rangle_{\partial\mathcal{T}_h} &= 0,\\
(\boldsymbol{q}_h, \nabla w)_{\mathcal{T}_h} - \langle \widehat{\boldsymbol{q}}_h\cdot\boldsymbol{n}, w\rangle_{\partial\mathcal{T}_h} + (s(\boldsymbol{H}_h, f), w)_{\mathcal{T}_h} &= 0,\\
\langle \widehat{\boldsymbol{q}}_h\cdot\boldsymbol{n}, \mu\rangle_{\partial\mathcal{T}_h\setminus\partial\Omega} + \langle \widehat{u}_h - g, \mu\rangle_{\partial\Omega} &= 0,
\end{aligned}
\tag{4}
$$

for all $(\boldsymbol{G}, \boldsymbol{v}, w, \mu) \in \boldsymbol{W}_h^p \times \boldsymbol{V}_h^p \times U_h^p \times M_h^p$, where

$$
\widehat{\boldsymbol{q}}_h = \boldsymbol{q}_h - \tau(u_h - \widehat{u}_h)\boldsymbol{n}, \quad \text{on } \mathcal{E}_h. \tag{5}
$$

Here τ is the stabilization parameter which, in general, can be a function of $\boldsymbol{H}_h, \boldsymbol{q}_h, u_h$ and $\widehat{u}_h$. The choice of the stabilization τ plays a crucial role in both the accuracy and stability of the resulting method. We shall study the effects of the stabilization on the convergence and stability of the numerical solutions for one of the examples presented later.

Newton's method is used to solve the nonlinear system (4). The procedure evaluates successive approximations $(\boldsymbol{H}_h^l, \boldsymbol{q}_h^l, u_h^l, \widehat{u}_h^l)$ starting from an initial guess $(\boldsymbol{H}_h^0, \boldsymbol{q}_h^0, u_h^0, \widehat{u}_h^0)$. For each Newton step l, the system of Eq. (4) is linearized with respect to the Newton increments $\left(\delta\boldsymbol{H}_h^l, \delta\boldsymbol{q}_h^l, \delta u_h^l, \delta\widehat{u}_h^l\right) \in \boldsymbol{W}_h^p \times \boldsymbol{V}_h^p \times U_h^p \times M_h^p$ that satisfy

$$\left(\delta\boldsymbol{H}_h^l, \boldsymbol{G}\right)_{\mathcal{T}_h} + \left(\delta\boldsymbol{q}_h^l, \nabla\cdot\boldsymbol{G}\right)_{\mathcal{T}_h} - \left\langle\delta\widehat{\boldsymbol{q}}_h^l, \boldsymbol{G}\cdot\boldsymbol{n}\right\rangle_{\partial\mathcal{T}_h} = r_1(\boldsymbol{G}), \tag{6a}$$

$$\left(\delta\boldsymbol{q}_h^l, \boldsymbol{v}\right)_{\mathcal{T}_h} + \left(\delta u_h^l, \nabla\cdot\boldsymbol{v}\right)_{\mathcal{T}_h} - \left\langle\delta\widehat{u}_h^l, \boldsymbol{v}\cdot\boldsymbol{n}\right\rangle_{\partial\mathcal{T}_h} = r_2(\boldsymbol{v}), \tag{6b}$$

$$(\partial s_{\boldsymbol{H}}(\boldsymbol{H}_h^l, f)\delta\boldsymbol{H}_h^l, w)_{\mathcal{T}_h} + \left(\delta\boldsymbol{q}_h^l, \nabla w\right)_{\mathcal{T}_h} - \left\langle\delta\widehat{\boldsymbol{q}}_h^l\cdot\boldsymbol{n}, w\right\rangle_{\partial\mathcal{T}_h} = r_3(w), \tag{6c}$$

$$\left\langle\delta\widehat{\boldsymbol{q}}_h^l\cdot\boldsymbol{n}, \mu\right\rangle_{\partial\mathcal{T}_h\setminus\partial\Omega} + \left\langle\delta\widehat{u}_h^l, \mu\right\rangle_{\partial\Omega} = r_4(\mu), \tag{6d}$$

for all $(\boldsymbol{G}, \boldsymbol{v}, w, \mu) \in \boldsymbol{W}_h^p \times \boldsymbol{V}_h^p \times U_h^p \times M_h^p$, the right-hand side residuals are given by

$$r_1(\boldsymbol{G}) = -\left(\boldsymbol{H}_h^l, \boldsymbol{G}\right)_{\mathcal{T}_h} - \left(\boldsymbol{q}_h^l, \nabla\cdot\boldsymbol{G}\right)_{\mathcal{T}_h} + \left\langle\widehat{\boldsymbol{q}}_h^l, \boldsymbol{G}\cdot\boldsymbol{n}\right\rangle_{\partial\mathcal{T}_h} \tag{7a}$$

$$r_2(\boldsymbol{v}) = -\left(\boldsymbol{q}_h^l, \boldsymbol{v}\right)_{\mathcal{T}_h} - \left(u_h^l, \nabla\cdot\boldsymbol{v}\right)_{\mathcal{T}_h} + \left\langle\widehat{u}_h^l, \boldsymbol{v}\cdot\boldsymbol{n}\right\rangle_{\partial\mathcal{T}_h} \tag{7b}$$

$$r_3(w) = -\left(\boldsymbol{q}_h^l, \nabla w\right)_{\mathcal{T}_h} + \left\langle\widehat{\boldsymbol{q}}_h^l\cdot\boldsymbol{n}, w\right\rangle_{\partial\mathcal{T}_h} - (s(\boldsymbol{H}_h^l, f), w)_{\mathcal{T}_h} \tag{7c}$$

$$r_4(\mu) = -\left\langle\widehat{\boldsymbol{q}}_h^l\cdot\boldsymbol{n}, \mu\right\rangle_{\partial\mathcal{T}_h\setminus\partial\Omega} - \left\langle\widehat{u}_h^l - g, \mu\right\rangle_{\partial\Omega}. \tag{7d}$$

Here $\partial s_{\boldsymbol{H}}(\boldsymbol{H}, f)$ denotes the partial derivatives of $s(\boldsymbol{H}, f)$ with respect to $\boldsymbol{H}$. After solving (6), the numerical approximations are then updated

$$\left(\boldsymbol{H}_h^{l+1}, \boldsymbol{q}_h^{l+1}, u_h^{l+1}, \widehat{u}_h^{l+1}\right) := \left(\boldsymbol{H}_h^l, \boldsymbol{q}_h^l, u_h^l, \widehat{u}_h^l\right) + \alpha\left(\delta\boldsymbol{H}_h^l, \delta\boldsymbol{q}_h^l, \delta u_h^l, \delta\widehat{u}_h^l\right), \tag{8}$$

where the coefficient α is determined by a line-search algorithm in order to optimally decrease the residual. This process is repeated until the residual norm is smaller than a given tolerance, typically 10^{-8}.

At each step of the Newton method, the linearization (6) gives the following matrix system to be solved

$$\begin{pmatrix}\mathbb{A}^l & \mathbb{B}^l\\ \mathbb{C}^l & \mathbb{D}^l\end{pmatrix}\begin{pmatrix}\delta U^l\\ \delta\widehat{U}^l\end{pmatrix} = \begin{pmatrix}R_{123}^l\\ R_4^l\end{pmatrix}, \tag{9}$$

where δU^l and $\delta\widehat{U}^l$ are the vectors of degrees of freedom of $(\delta\boldsymbol{H}_h^l, \delta\boldsymbol{q}_h^l, \delta u_h^l)$ and $\delta\widehat{u}_h^l$, respectively. The system (9) is first solved for the traces only $\delta\widehat{U}^l$ as

$$\mathbb{K}^l\delta\widehat{U}^l = R^l, \tag{10}$$

where $\mathbb{K}^l$ is the Schur complement of the block $\mathbb{A}^l$ and R^l is the reduced residual

$$\mathbb{K}^l = \mathbb{D}^l - \mathbb{C}^l\left(\mathbb{A}^l\right)^{-1}\mathbb{B}^l, \qquad R^l = R_4^l - \mathbb{C}^l\left(\mathbb{A}^l\right)^{-1}R_{123}^l. \tag{11}$$

The reduced system (10) involves fewer degrees of freedom than the full system (9). Moreover due to the discontinuous nature of the approximate solution, the matrix $\mathbb{A}^l$ and its inverse are

block diagonal, and can be computed elementwise. Once $\delta\widehat{U}^l$ is known, the other unknowns δU^l are then retrieved element-wise. Therefore, the full system (9) is never explicitly built, and the reduced matrix $\mathbb{K}^l$ is built directly in an elementwise fashion, thus reducing the memory storage.

2.3 The Fixed Point-HDG Method

In addition to using Newton's method, we employ the fixed-point method to solve the nonlinear system (4). We note that the nonlinear source term s in (4) depends on $\boldsymbol{H}_h$. Hence, starting from an initial guess $\boldsymbol{H}_h^0$ we repeatedly find $(\boldsymbol{q}_h^l, u_h^l, \widehat{u}_h^l) \in \boldsymbol{V}_h^p \times U_h^p \times M_h^p$ such that

$$\begin{aligned} (\boldsymbol{q}_h^l, \boldsymbol{v})_{\mathcal{T}_h} + (u_h^l, \nabla\cdot\boldsymbol{v})_{\mathcal{T}_h} - \langle \widehat{u}_h^l, \boldsymbol{v}\cdot\boldsymbol{n}\rangle_{\partial\mathcal{T}_h} &= 0, \\ (\boldsymbol{q}_h^l, \nabla w)_{\mathcal{T}_h} - \langle \widehat{\boldsymbol{q}}_h^l\cdot\boldsymbol{n}, w\rangle_{\partial\mathcal{T}_h} &= -(s(\boldsymbol{H}_h^{l-1}, f), w)_{\mathcal{T}_h}, \\ \langle \widehat{\boldsymbol{q}}_h^l\cdot\boldsymbol{n}, \mu\rangle_{\partial\mathcal{T}_h\setminus\partial\Omega} + \langle \widehat{u}_h^l, \mu\rangle_{\partial\Omega} &= \langle g, \mu\rangle_{\partial\Omega}, \end{aligned} \tag{12}$$

for all $(\boldsymbol{v}, w, \mu) \in \boldsymbol{V}_h^p \times U_h^p \times M_h^p$, and then compute $\boldsymbol{H}_h^l \in \boldsymbol{W}_h^p$ such that

$$\left(\boldsymbol{H}_h^l, \boldsymbol{G}\right)_{\mathcal{T}_h} = -\left(\boldsymbol{q}_h^l, \nabla\cdot\boldsymbol{G}\right)_{\mathcal{T}_h} + \left\langle \widehat{\boldsymbol{q}}_h^l, \boldsymbol{G}\cdot\boldsymbol{n}\right\rangle_{\partial\mathcal{T}_h}, \quad \forall \boldsymbol{G} \in \boldsymbol{W}_h^p \tag{13}$$

until $\|\boldsymbol{H}_h^l - \boldsymbol{H}_h^{l-1}\|$ is less than a specified tolerance, typically 10^{-6}.

We note that the weak formulation (12) is nothing but the HDG method for the Poisson equation $-\Delta u = -s$ in Ω with $u = g$ on $\partial\Omega$. Applying the solution strategy described earlier to the system (12), we obtain the following global system

$$\mathbb{G}\,\widehat{U}^l = F^l, \tag{14}$$

for the degrees of freedom of $\widehat{u}_h^l$. As the global matrix $\mathbb{G}$ is unchanged, it can be computed once prior to carrying out the fixed-point algorithm. However, the right hand side vector F^l has to be computed at every fixed-point iteration because the source term depends on the previous iterate. Once $\widehat{U}^l$ is obtained by solving the linear system (14), the degrees of freedom of both $\boldsymbol{q}_h^l$ and u_h^l are then retrieved element-wise. Finally, the degrees of freedom of $\boldsymbol{H}_h^l$ are also computed by solving (13) element-wise.

We see that the fixed-point HDG method for the Monge–Ampère equation means to solving the Poisson problem with a sequence of right-hand side vectors. Hence, the fixed-point HDG method is much easier to implement the Newton-HDG method. Furthermore, its global matrix is computed only once, whereas that of the Newton-HDG method has to be computed for each Newton step. However, the Newton-HDG method can converge considerably faster than the fixed-point HDG method.

2.4 Numerical Examples

In this section we provide examples to demonstrate the HDG methods described above. In the first example we compare the convergence and accuracy of the methods for several polynomial degrees. In the second example we focus on a problem with a nearly singular solution. These examples demonstrate the relative advantages and disadvantages of the HDG methods. For the first problem, the Newton-HDG method has robust convergence in the nonlinear iteration, while the fixed-point HDG method requires more iterations. The second example illustrates the influence of regularity on the performance of the methods. For these

examples, we choose $\mathcal{T}_h$ as a uniform triangulation of Ω with mesh size $h = 1/n$ and use $u^0 = (x^2 + y^2)/2$ as the initial guess.

Example 1 We consider the Monge–Ampère equation in $\Omega = (0, 1)^2$ with $f = (1 + x^2 + y^2)e^{x^2+y^2}$ and $g = e^{0.5(x^2+y^2)}$, which yields the exact solution $u = e^{0.5(x^2+y^2)}$. We use triangular and quadrilateral meshes with uniform mesh size $h = 1/n$ for $n = 4, 8, 16, 32, 64$, and polynomial degrees of $p = 1, 2, 3$. We consider three different values for the stabilization parameter $\tau = 1$, $\tau = 1/h$, and $\tau = h$.

We report the L^2 errors and orders of convergence of the numerical solutions on the triangular meshes in Table 1 and the quadrilateral meshes in Table 2 for $\tau = 1$. We see that both the Newton-HDG method and the fixed-point HDG method have the same errors and convergence rates. These results are expected because the two methods should converge to the same numerical solution. We observe interestingly that the convergence rates of u_h, $\boldsymbol{q}_h$, and $\boldsymbol{H}_h$ are $O(h^p)$ on triangular and quadrilateral meshes. The convergence rate of $O(h^p)$ for the computed Hessian $\boldsymbol{H}_h$ is a good news for the HDG methods since the Hessian H are the second-order partial derivatives of the scalar variable u. However, the convergence rate of $O(h^p)$ for both u_h and $\boldsymbol{q}_h$ are suboptimal. The suboptimal convergence of u_h and $\boldsymbol{q}_h$ for the Monge–Ampère equation can be attributed to the fact that the source term s depends on the Hessian $\boldsymbol{H}$. As the computed Hessian $\boldsymbol{H}_h$ converges with order $O(h^p)$, the L^2 projection of the source term $s(\boldsymbol{H}_h, f)$ also converges with order $O(h^p)$. In contrast, for linear second-order elliptic problems with smooth data and regular geometries, the L^2 projection of the source term converges with order $O(h^{p+1})$, the HDG method can yield optimal convergence rate of $O(h^{p+1})$ for both u_h and $\boldsymbol{q}_h$.

We show the L^2 errors and orders of convergence of the numerical solutions on the triangular meshes in Table 3 and the quadrilateral meshes in Table 4 for $\tau = 1/h$. We observe that the approximate Hessian $\boldsymbol{H}_h$ for $\tau = 1/h$ converges with order $p - 1$, which is one order less than that for $\tau = 1$. The approximate scalar u_h and gradient $\boldsymbol{q}_h$ converge with order $p - 1$ for $p = 1, 3$ and with order p for $p = 2, 4$. Hence, it appears that u_h and $\boldsymbol{q}_h$ have the convergence rate of $O(h^{p-1})$ for odd polynomial degrees and $O(h^p)$ for even polynomial degrees when the stabilization parameter is chosen as $\tau = 1/h$.

We show the L^2 errors and orders of convergence of the numerical solutions on the triangular meshes in Table 5 and the quadrilateral meshes in Table 6 for $\tau = h$. We observe that the approximate Hessian $\boldsymbol{H}_h$ converges with order p on triangular meshes, but with order $p - 1$ on quadrilateral meshes. The approximate scalar u_h converges with order p on triangular meshes, but with order $p-1$ for $p = 1, 3$ on quadrilateral meshes. The approximate gradient $\boldsymbol{q}_h$ converges with order $p+1$ for $p = 1, 3$ and p for $p = 2, 4$ on triangular meshes, but with order $p - 1$ for $p = 1, 3$ and p for $p = 2, 4$ on quadrilateral meshes.

Table 7 summarizes the convergence rates of the numerical solutions. We see that the convergence rates depend on the stabilization parameter τ, the type of meshes, and the polynomial degree p. For $\tau = 1$, the convergence rates of all the approximate variables are $O(h^p)$ for all polynomial degrees considered on both types of meshes. However, for the other two choices of the stabilization parameter, the convergence rates of the approximate variables are no longer the same. Therefore, we shall use $\tau = 1$ in the other examples.

Lastly, in Table 8, we show the number of iterations required to reach convergence for $\tau = 1$. As expected, the Newton-HDG method requires considerably less iterations (about six times less) than the fixed-point HDG method. Furthermore, the number of iterations appears consistent for all values of n and p on both triangular and quadrilateral meshes.

Table 1 Error and convergence rate of the numerical solutions computed on the triangular meshes for $\tau = 1$ for the Example 1

Degree	Mesh	Newton-HDG method						Fixed point-HDG method					
		$\|u - u_h\|$		$\|\boldsymbol{q} - \boldsymbol{q}_h\|$		$\|\boldsymbol{H} - \boldsymbol{H}_h\|$		$\|u - u_h\|$		$\|\boldsymbol{q} - \boldsymbol{q}_h\|$		$\|\boldsymbol{H} - \boldsymbol{H}_h\|$	
p	n	Error	Order	Error	Order	Error	Order	Error	Order	Error	Order	Error	Order
1	4	1.37e−3	–	1.12e−2	–	5.75e−1	–	1.37e−3	–	1.12e−2	–	5.75e−1	–
	8	1.13e−4	1.96	2.73e−3	2.04	2.95e−1	0.97	1.13e−4	1.96	2.73e−3	2.04	2.95e−1	0.97
	16	1.75e−4	1.80	1.16e−3	1.23	1.49e−1	0.98	1.75e−4	1.80	1.16e−3	1.23	1.49e−1	0.98
	32	1.37e−4	1.25	7.30e−4	0.67	7.49e−2	0.99	1.37e−4	1.25	7.30e−4	0.67	7.49e−2	0.99
	64	8.13e−5	0.94	4.21e−4	0.79	3.76e−2	1.00	8.13e−5	0.94	4.21e−4	0.79	3.76e−2	1.00
2	4	1.29e−4	–	1.01e−3	–	5.82e−2	–	1.29e−4	–	1.01e−3	–	5.82e−2	–
	8	4.28e−5	2.51	2.44e−4	2.05	1.49e−2	1.96	4.28e−5	2.51	2.44e−4	2.05	1.49e−2	1.96
	16	1.17e−5	2.21	6.26e−5	1.96	3.78e−3	1.98	1.17e−5	2.21	6.26e−5	1.96	3.78e−3	1.98
	32	3.03e−6	2.05	1.60e−5	1.97	9.49e−4	1.99	3.03e−6	2.05	1.60e−5	1.97	9.49e−4	1.99
	64	7.67e−7	2.01	4.03e−6	1.98	2.38e−4	2.00	7.67e−7	2.01	4.03e−6	1.98	2.38e−4	2.00
3	4	1.62e−6	–	4.22e−5	–	4.37e−3	–	1.62e−6	–	4.22e−5	–	4.37e−3	–
	8	3.13e−7	3.82	3.23e−6	3.71	5.60e−4	2.96	3.13e−7	3.82	3.23e−6	3.71	5.60e−4	2.96
	16	5.35e−8	3.60	3.34e−7	3.27	7.04e−5	2.99	5.35e−8	3.60	3.34e−7	3.27	7.04e−5	2.99
	32	7.59e−9	3.23	4.17e−8	3.00	8.80e−6	3.00	7.59e−9	3.23	4.17e−8	3.00	8.80e−6	3.00
	64	9.9e−10	3.05	5.31e−9	2.98	1.10e−6	3.00	9.9e−10	3.05	5.31e−9	2.98	1.10e−6	3.00
4	4	3.54e−7	–	2.74e−6	–	2.82e−4	–	3.54e−7	–	2.74e−6	–	2.82e−4	–
	8	2.55e−8	4.51	1.52e−7	4.17	1.81e−5	3.96	2.55e−8	4.51	1.52e−7	4.17	1.81e−5	3.96
	16	1.70e−9	4.23	9.36e−9	4.03	1.13e−6	4.00	1.7e−9	4.24	9.33e−9	4.03	1.13e−6	4.00
	32	1.06e−10	4.10	5.73e−10	4.03	7.06e−8	4.00	1.05e−10	4.12	5.64e−10	4.05	7.09e−8	4.00

Table 2 Error and convergence rate of the numerical solutions computed on the quadrilateral meshes for $\tau = 1$ for the Example 1

Degree	Mesh	Newton-HDG method						Fixed point-HDG method					
		$\|u - u_h\|$		$\|\boldsymbol{q} - \boldsymbol{q}_h\|$		$\|\boldsymbol{H} - \boldsymbol{H}_h\|$		$\|u - u_h\|$		$\|\boldsymbol{q} - \boldsymbol{q}_h\|$		$\|\boldsymbol{H} - \boldsymbol{H}_h\|$	
p	n	Error	Order	Error	Order	Error	Order	Error	Order	Error	Order	Error	Order
1	4	2.20e−3	–	2.64e−2	–	5.03e−1	–	2.20e−3	–	2.64e−2	–	5.03e−1	–
	8	4.68e−4	1.73	8.85e−3	1.58	2.92e−1	0.78	4.68e−4	1.73	8.85e−3	1.58	2.92e−1	0.78
	16	1.64e−4	1.83	3.05e−3	1.54	1.66e−1	0.82	1.64e−4	1.83	3.05e−3	1.54	1.66e−1	0.82
	32	1.04e−4	1.69	1.19e−3	1.36	9.43e−2	0.82	1.04e−4	1.69	1.19e−3	1.36	9.43e−2	0.82
	64	6.45e−5	1.27	5.28e−4	1.17	5.40e−2	0.80	6.45e−5	1.27	5.28e−4	1.17	5.40e−2	0.80
2	4	1.04e−4	–	1.62e−3	–	5.49e−2	–	1.04e−4	–	1.62e−3	–	5.49e−2	–
	8	3.20e−5	2.63	2.89e−4	2.49	1.62e−2	1.76	3.20e−5	2.63	2.89e−4	2.49	1.62e−2	1.76
	16	8.56e−6	2.48	5.70e−5	2.34	4.8e−3	1.76	8.56e−6	2.48	5.70e−5	2.34	4.8e−3	1.76
	32	2.20e−6	2.24	1.26e−5	2.18	1.42e−3	1.75	2.20e−6	2.24	1.26e−5	2.18	1.42e−3	1.75
	64	5.55e−7	2.08	2.97e−6	2.08	4.24e−4	1.75	5.55e−7	2.08	2.97e−6	2.08	4.24e−4	1.75
3	4	1.93e−6	–	5.88e−5	–	3.24e−3	–	1.93e−6	–	5.88e−5	–	3.24e−3	–
	8	2.11e−7	3.80	5.09e−6	3.53	4.93e−4	2.72	2.11e−7	3.80	5.09e−6	3.53	4.93e−4	2.72
	16	2.95e−8	3.79	4.65e−7	3.45	7.41e−5	2.73	2.95e−8	3.79	4.65e−7	3.45	7.41e−5	2.73
	32	4.32e−9	3.55	4.67e−8	3.31	1.11e−5	2.74	4.32e−9	3.55	4.67e−8	3.31	1.11e−5	2.74
	64	6.0e−10	3.19	5.15e−9	3.18	1.65e−6	2.74	6.0e−10	3.19	5.14e−9	3.18	1.66e−6	2.74
4	4	1.92e−7	–	2.47e−6	–	1.84e−4	–	1.92e−7	–	2.47e−6	–	1.84e−4	–
	8	1.33e−8	4.62	1.13e−7	4.45	1.41e−5	3.71	1.33e−8	4.62	1.13e−7	4.45	1.41e−5	3.71
	16	8.59e−10	4.42	5.65e−9	4.32	1.06e−6	3.73	8.59e−10	4.42	5.64e−9	4.32	1.07e−6	3.73
	32	5.51e−11	4.17	3.16e−10	4.16	7.99e−8	3.73	5.43e−11	4.19	3.22e−10	4.13	8.45e−8	3.66

Table 3 Error and convergence rate of the numerical solutions computed on the triangular meshes for $\tau = 1/h$ for the Example 1

Degree	Mesh	Newton-HDG method						Fixed point-HDG method					
		$\|u - u_h\|$		$\|\boldsymbol{q} - \boldsymbol{q}_h\|$		$\|\boldsymbol{H} - \boldsymbol{H}_h\|$		$\|u - u_h\|$		$\|\boldsymbol{q} - \boldsymbol{q}_h\|$		$\|\boldsymbol{H} - \boldsymbol{H}_h\|$	
p	n	Error	Order	Error	Order	Error	Order	Error	Order	Error	Order	Error	Order
1	4	5.62e−3	–	3.82e−2	–	1.24e−0	–	5.62e−3	–	3.82e−2	–	1.24e−0	–
	8	4.83e−3	0.21	2.68e−2	0.51	1.15e−0	0.11	4.83e−3	0.21	2.68e−2	0.51	1.15e−0	0.11
	16	4.64e−3	0.05	2.27e−2	0.24	1.12e−0	0.03	4.64e−3	0.05	2.27e−2	0.24	1.12e−0	0.03
	32	4.59e−3	0.01	2.15e−2	0.08	1.12e−0	0.01	4.59e−3	0.01	2.15e−2	0.08	1.12e−0	0.01
	64	4.58e−3	0.00	2.12e−2	0.02	1.11e−0	0.00	4.58e−3	0.00	2.12e−2	0.02	1.11e−0	0.00
2	4	8.6e−5	–	2.30e−3	–	1.09e−1	–	8.6e−5	–	2.3e−3	–	1.09e−1	–
	8	1.13e−5	3.09	5.91e−4	1.96	5.03e−2	1.11	1.13e−5	3.09	5.91e−4	1.96	5.03e−2	1.11
	16	1.76e−6	2.74	1.49e−4	1.99	2.46e−2	1.04	1.76e−6	2.74	1.49e−4	1.99	2.46e−2	1.04
	32	3.47e−7	2.36	3.73e−5	2.00	1.22e−2	1.01	3.47e−7	2.36	3.73e−5	2.00	1.22e−2	1.01
	64	8.01e−8	2.12	9.33e−6	2.00	6.09e−3	1.00	8.01e−8	2.12	9.33e−6	2.00	6.09e−3	1.00
3	4	1.13e−5	–	1.45e−4	–	8.67e−3	–	1.13e−5	–	1.45e−4	–	8.67e−3	–
	8	3.55e−6	1.71	2.53e−5	2.52	2.05e−3	2.08	3.55e−6	1.71	2.53e−5	2.52	2.05e−3	2.08
	16	9.45e−7	1.91	5.43e−6	2.22	5.03e−4	2.02	9.45e−7	1.91	5.43e−6	2.22	5.03e−4	2.02
	32	2.40e−7	1.98	1.30e−6	2.07	1.25e−4	2.01	2.40e−7	1.98	1.30e−6	2.07	1.25e−4	2.01
	64	6.02e−8	1.99	3.20e−7	2.02	3.13e−5	2.00	6.02e−8	1.99	3.20e−7	2.02	3.13e−5	2.00
4	4	2.15e−7	–	6.44e−6	–	5.39e−4	–	2.15e−7	–	6.44e−6	–	5.39e−4	–
	8	9.52e−9	4.84	4.29e−7	3.91	6.40e−5	3.07	9.51e−9	4.84	4.29e−7	3.91	6.40e−5	3.07
	16	4.9e−10	4.64	2.77e−8	3.95	7.92e−6	3.01	4.9e−10	4.64	2.77e−8	3.95	7.92e−6	3.01
	32	2.5e−11	4.50	1.76e−9	3.98	9.91e−7	3.00	2.5e−11	4.50	1.76e−9	3.97	9.91e−7	3.00

Table 4 Error and convergence rate of the numerical solutions computed on the quadrilateral meshes for $\tau = 1/h$ for the Example 1

Degree	Mesh	Newton-HDG method						Fixed point-HDG method					
		$\|u - u_h\|$		$\|\boldsymbol{q} - \boldsymbol{q}_h\|$		$\|\boldsymbol{H} - \boldsymbol{H}_h\|$		$\|u - u_h\|$		$\|\boldsymbol{q} - \boldsymbol{q}_h\|$		$\|\boldsymbol{H} - \boldsymbol{H}_h\|$	
p	n	Error	Order	Error	Order	Error	Order	Error	Order	Error	Order	Error	Order
	4	3.43e−3	–	3.52e−2	–	5.81e−1	–	3.43e−3	–	3.52e−2	–	5.81e−1	–
	8	3.42e−3	0.18	2.41e−2	0.55	4.8e−1	0.27	3.42e−3	0.18	2.41e−2	0.55	4.8e−1	0.27
1	16	3.42e−3	0.01	1.98e−2	0.28	4.5e−1	0.09	3.42e−3	0.01	1.98e−2	0.28	4.5e−1	0.09
	32	3.42e−3	0.00	1.85e−2	0.10	4.42e−1	0.03	3.42e−3	0.00	1.85e−2	0.10	4.42e−1	0.03
	64	3.42e−3	0.00	1.82e−2	0.03	4.4e−1	0.01	3.42e−3	0.00	1.82e−2	0.03	4.4e−1	0.01
	4	7.16e−5	–	1.36e−3	–	4.68e−2	–	7.16e−5	–	1.36e−3	–	4.68e−2	–
	8	1.63e−5	2.72	3.42e−4	1.99	1.87e−2	1.32	1.63e−5	2.72	3.42e−4	1.99	1.87e−2	1.32
2	16	3.93e−6	2.12	8.64e−5	1.99	8.67e−3	1.11	3.93e−6	2.12	8.64e−5	1.99	8.67e−3	1.11
	32	9.71e−7	2.02	2.17e−5	2.00	4.24e−3	1.03	9.71e−7	2.02	2.17e−5	2.00	4.24e−3	1.03
	64	2.42e−7	2.00	5.42e−6	2.00	2.1e−3	1.01	2.42e−7	2.00	5.42e−6	2.00	2.1e−3	1.01
	4	7.45e−6	–	6.38e−5	–	2.87e−3	–	7.45e−6	–	6.38e−5	–	2.87e−3	–
	8	2.29e−6	1.82	1.35e−5	2.24	6.1e−4	2.23	2.29e−6	1.82	1.35e−5	2.24	6.1e−4	2.23
3	16	6.02e−7	1.93	3.21e−6	2.07	1.45e−4	2.07	6.02e−7	1.93	3.21e−6	2.07	1.45e−4	2.07
	32	1.52e−7	1.98	7.92e−7	2.02	3.59e−5	2.02	1.52e−7	1.98	7.92e−7	2.02	3.59e−5	2.02
	64	3.82e−8	2.00	1.97e−7	2.01	8.93e−6	2.01	3.82e−8	2.00	1.97e−7	2.01	8.93e−6	2.01
	4	1.06e−7	–	1.71e−6	–	1.32e−4	–	1.06e−7	–	1.71e−6	–	1.32e−4	–
	8	6.52e−9	4.60	1.05e−7	4.03	1.33e−5	3.31	6.52e−9	4.60	1.05e−7	4.03	1.33e−5	3.31
4	16	4.06e−10	4.07	6.6e−9	3.99	1.56e−6	3.09	4.05e−10	4.08	6.6e−9	3.99	1.56e−6	3.09
	32	2.58e−11	3.98	4.15e−10	3.99	1.91e−7	3.03	2.57e−11	3.98	4.22e−10	3.97	1.91e−7	3.03

Table 5 Error and convergence rate of the numerical solutions computed on the triangular meshes for $\tau = h$ for the Example 1

Degree	Mesh	Newton-HDG method						Fixed point-HDG method					
		$\|u - u_h\|$		$\|\boldsymbol{q} - \boldsymbol{q}_h\|$		$\|\boldsymbol{H} - \boldsymbol{H}_h\|$		$\|u - u_h\|$		$\|\boldsymbol{q} - \boldsymbol{q}_h\|$		$\|\boldsymbol{H} - \boldsymbol{H}_h\|$	
p	n	Error	Order	Error	Order	Error	Order	Error	Order	Error	Order	Error	Order
1	4	2.02e−3	–	1.03e−2	–	4.9e−1	–	2.02e−3	–	1.03e−2	–	4.9e−1	–
	8	5.47e−4	0.97	2.86e−3	1.86	2.51e−1	0.96	5.47e−4	0.97	2.86e−3	1.86	2.51e−1	0.96
	16	1.4e−4	0.99	7.42e−4	1.95	1.27e−1	0.98	1.4e−4	0.99	7.42e−4	1.95	1.27e−1	0.98
	32	3.54e−5	1.00	1.88e−4	1.98	6.41e−2	0.99	3.54e−5	1.00	1.88e−4	1.98	6.41e−2	0.99
	64	8.87e−6	1.00	4.73e−5	1.99	3.22e−2	0.99	8.87e−6	1.00	4.73e−5	1.99	3.22e−2	0.99
2	4	1.51e−4	–	9.04e−4	–	5.19e−2	–	1.51e−4	–	9.04e−4	–	5.19e−2	–
	8	4.66e−5	1.97	2.49e−4	1.86	1.32e−2	1.97	4.66e−5	1.97	2.49e−4	1.86	1.32e−2	1.97
	16	1.22e−5	1.99	6.44e−5	1.95	3.35e−3	1.98	1.22e−5	1.99	6.44e−5	1.95	3.35e−3	1.98
	32	3.09e−6	2.00	1.63e−5	1.99	8.42e−4	1.99	3.09e−6	2.00	1.63e−5	1.99	8.42e−4	1.99
	64	7.76e−7	2.00	4.07e−6	2.00	2.11e−4	2.00	7.76e−7	2.00	4.07e−6	2.00	2.11e−4	2.00
3	4	2.42e−6	–	2.51e−5	–	3.8e−3	–	2.42e−6	–	2.51e−5	–	3.8e−3	–
	8	1.50e−7	2.95	1.64e−6	3.94	4.82e−4	2.98	1.50e−7	2.95	1.64e−6	3.94	4.82e−4	2.98
	16	9.24e−9	2.99	1.05e−7	3.96	6.04e−5	2.99	9.24e−9	2.99	1.05e−7	3.96	6.04e−5	2.99
	32	5.7e−10	3.00	6.66e−9	3.98	7.6e−6	3.00	5.7e−10	3.00	6.67e−9	3.98	7.6e−6	3.00
	64	4.9e−11	3.00	4.6e−10	3.87	9.47e−7	3.00	5.5e−11	3.00	4.8e−10	3.79	9.47e−7	3.00
4	4	4.07e−7	–	2.40e−6	–	2.48e−4	–	4.07e−7	–	2.40e−6	–	2.48e−4	–
	8	2.79e−8	3.95	1.53e−7	3.97	1.57e−5	3.99	2.79e−8	3.95	1.53e−7	3.97	1.57e−5	3.99
	16	1.78e−9	3.99	9.62e−9	3.99	9.78e−7	4.00	1.78e−9	3.99	9.6e−9	4.00	9.78e−7	4.00
	32	1.1e−10	4.00	5.8e−10	4.04	6.09e−8	4.01	1.1e−10	4.00	5.8e−10	4.06	6.10e−8	4.00

Table 6 Error and convergence rate of the numerical solutions computed on the quadrilateral meshes for $\tau = h$ for the Example 1

Degree	Mesh	Newton-HDG method						Fixed point-HDG method					
		$\|u - u_h\|$		$\|\boldsymbol{q} - \boldsymbol{q}_h\|$		$\|\boldsymbol{H} - \boldsymbol{H}_h\|$		$\|u - u_h\|$		$\|\boldsymbol{q} - \boldsymbol{q}_h\|$		$\|\boldsymbol{H} - \boldsymbol{H}_h\|$	
p	n	Error	Order	Error	Order	Error	Order	Error	Order	Error	Order	Error	Order
1	4	9.37e−3	–	7.75e−2	–	9.68e−1	–	9.37e−3	–	7.75e−2	–	9.68e−1	–
	8	6.3e−3	0.82	5.26e−2	0.56	9.78e−1	−0.01	6.3e−3	0.82	5.26e−2	0.56	9.78e−1	−0.01
	16	5.47e−3	0.63	4.22e−2	0.32	9.85e−1	−0.01	5.47e−3	0.63	4.22e−2	0.32	9.85e−1	−0.01
	32	5.26e−3	0.31	3.9e−2	0.12	9.89e−1	−0.01	5.26e−3	0.31	3.9e−2	0.12	9.89e−1	−0.01
	64	5.2e−3	0.11	3.81e−2	0.03	9.91e−1	0.00	5.2e−3	0.11	3.81e−2	0.03	9.91e−1	0.00
2	4	1.12e−4	–	4e−3	–	1.25e−1	–	1.12e−4	–	4e−3	–	1.25e−1	–
	8	2.85e−5	1.95	1.02e−3	1.96	6.24e−2	1	2.85e−5	1.95	1.02e−3	1.96	6.24e−2	1
	16	7.31e−6	1.97	2.62e−4	1.97	3.17e−2	0.98	7.31e−6	1.97	2.62e−4	1.97	3.17e−2	0.98
	32	1.83e−6	1.98	6.7e−5	1.97	1.62e−2	0.97	1.83e−6	1.98	6.7e−5	1.97	1.62e−2	0.97
	64	4.55e−7	1.98	1.73e−5	1.95	8.26e−3	0.97	4.55e−7	1.98	1.73e−5	1.95	8.26e−3	0.97
3	4	9.15e−6	–	1.68e−4	–	7.72e−3	–	9.15e−6	–	1.68e−4	–	7.72e−3	–
	8	2.27e−6	2.87	2.86e−5	2.55	1.97e−3	1.97	2.27e−6	2.87	2.86e−5	2.55	1.97e−3	1.97
	16	5.65e−7	2.71	5.88e−6	2.28	4.98e−4	1.98	5.65e−7	2.71	5.88e−6	2.28	4.98e−4	1.98
	32	1.41e−7	2.4	1.37e−6	2.1	1.25e−4	1.99	1.41e−7	2.4	1.37e−6	2.1	1.25e−4	1.99
	64	3.53e−8	2.15	3.36e−7	2.03	3.15e−5	1.99	3.53e−8	2.15	3.36e−7	2.03	3.15e−5	1.99
4	4	2.6e−7	–	6.36e−6	–	4.69e−4	–	2.6e−7	–	6.36e−6	–	4.69e−4	–
	8	1.77e−8	3.94	4.18e−7	3.93	6.06e−5	2.95	1.77e−8	3.94	4.18e−7	3.93	6.06e−5	2.95
	16	1.11e−9	3.98	2.67e−8	3.97	7.72e−6	2.97	1.11e−9	3.98	2.67e−8	3.97	7.72e−6	2.97
	32	6.95e−11	3.99	1.71e−9	3.97	9.8e−7	2.98	6.87e−11	3.99	1.7e−9	3.97	9.78e−7	2.98

Table 7 Convergence rate of the numerical solutions for the Example 1

Degree	Triangular meshes									Quadrilateral meshes								
	$\tau = 1$			$\tau = 1/h$			$\tau = h$			$\tau = 1$			$\tau = 1/h$			$\tau = h$		
p	u_h	$\boldsymbol{q}_h$	$\boldsymbol{H}_h$	u_h	$\boldsymbol{q}_h$	$\boldsymbol{H}_h$	u_h	$\boldsymbol{q}_h$	$\boldsymbol{H}_h$	u_h	$\boldsymbol{q}_h$	$\boldsymbol{H}_h$	u_h	$\boldsymbol{q}_h$	$\boldsymbol{H}_h$	u_h	$\boldsymbol{q}_h$	$\boldsymbol{H}_h$
1	p	p	p	$p-1$	$p-1$	$p-1$	p	$p+1$	p	p	p	p	$p-1$	$p-1$	$p-1$	$p-1$	$p-1$	$p-1$
2	p	p	p	p	p	$p-1$	p	p	p	p	p	p	p	p	$p-1$	p	p	$p-1$
3	p	p	p	$p-1$	$p-1$	$p-1$	p	$p+1$	p	p	p	p	$p-1$	$p-1$	$p-1$	$p-1$	$p-1$	$p-1$
4	p	p	p	p	p	$p-1$	p	p	p	p	p	p	p	p	$p-1$	p	p	$p-1$

Table 8 Number of iterations required to reach convergence for the HDG methods with $\tau = 1$ for the Example 1

Mesh	Newton-HDG method						Fixed point-HDG method					
	p=1		p=2		p=3		p=1		p=2		p=3	
n	Tri	Quad	Tri	Quad	Tri	Quad	Tri	Quad	Tri	Quad	Tri	Quad
4	6	6	6	6	6	6	30	37	38	38	40	42
8	6	6	6	6	6	6	36	40	41	41	41	43
16	7	6	6	6	6	6	40	43	42	43	42	44
32	7	7	6	6	6	6	42	44	43	44	44	45
64	7	7	6	6	6	6	44	44	45	45	45	46

Here "tri" columns refer to triangular meshes, while "quad" columns refer to quadrilateral meshes

Example 2 We consider the Monge–Ampère equation in $\Omega = (0, 1)^2$ with $f = R^2/(R^2 - x^2 - y^2)^2$ and $g = -\sqrt{R^2 - x^2 - y^2}$, which yields the exact solution $u = -\sqrt{R^2 - x^2 - y^2}$ for $R > \sqrt{2}$. Note that the regularity of the exact solution decreases as R decreases toward $\sqrt{2}$. Indeed, the gradient and Hessian of the exact solution are singular at the corner $(1, 1)$ for $R = \sqrt{2}$. In this example, we would like to see how the HDG methods perform as the regularity of the solution decreases. Hence, we will report numerical results for $R = 2$, $R = \sqrt{2} + 0.1$, and $R = \sqrt{2} + 0.01$. We consider triangular meshes with size $h = 1/n$, and stabilization parameter $\tau = 1$.

We report the L^2 errors and orders of convergence of the computed solutions in Table 9 for $R = 2$, Table 10 for $R = \sqrt{2}+0.1$, and Table 11 for $R = \sqrt{2}+0.01$. The Newton-HDG method and the fixed-point HDG method have very similar errors and convergence rates. We observe again that u_h, $\boldsymbol{q}_h$, and $\boldsymbol{H}_h$ converge with order $O(h^p)$, except for $p = 3$ and $R = \sqrt{2}+0.1$ where both u_h and $\boldsymbol{q}_h$ appear to converge like $O(h^{p+1})$. As R decreases, the L^2 errors tend to increase because both $\boldsymbol{q}$ and $\boldsymbol{H}$ become less smooth. Hence, we need to increase the mesh resolution in order to observe the expected convergence rates for $R = \sqrt{2} + 0.01$. Table 12 lists the number of iterations required to reach convergence. The Newton-HDG method requires considerably less iterations than the fixed-point HDG method. The number of iterations for the Newton-HDG method remains consistent, whereas that for the fixed-point HDG method tends to increase as R decreases.

3 Optimal Transport for *r*-adaptive Mesh Generation

In this section, we review optimal transport theory and describe how it can be applied to r-adaptive mesh generation. This r-adaptive mesh generation approach results in the Monge–Ampère equation with nonlinear Neumann boundary condition, which is solved by extending the HDG methods described in the previous section. We present numerical experiments to demonstrate the performance of the HDG methods for r-adaptive mesh generation.

3.1 Optimal Transport Theory

The optimal transport (OT) problem is described as follows. Suppose we are given two probability densities: $\rho(\boldsymbol{x})$ supported on $\Omega \in \mathbb{R}^d$ and $\rho'(\boldsymbol{x}')$ supported on $\Omega' \in \mathbb{R}^d$. The

Table 9 Error and convergence rate for the HDG methods for the Example 2 with $R = 2$

Degree	Mesh	Newton-HDG method						Fixed point-HDG method					
		$\|u - u_h\|$		$\|\boldsymbol{q} - \boldsymbol{q}_h\|$		$\|\boldsymbol{H} - \boldsymbol{H}_h\|$		$\|u - u_h\|$		$\|\boldsymbol{q} - \boldsymbol{q}_h\|$		$\|\boldsymbol{H} - \boldsymbol{H}_h\|$	
p	n	Error	Order	Error	Order	Error	Order	Error	Order	Error	Order	Error	Order
1	4	1.23e−4	–	2.30e−3	–	9.93e−2	–	1.23e−4	–	2.3e−3	–	9.93e−2	–
	8	1.96e−5	1.96	6.25e−4	1.88	5.05e−2	0.98	1.96e−5	1.96	6.25e−4	1.88	5.05e−2	0.98
	16	1.99e−5	1.89	1.92e−4	1.70	2.54e−2	0.99	1.99e−5	1.89	1.92e−4	1.70	2.54e−2	0.99
	32	1.43e−5	1.54	8.47e−5	1.18	1.28e−2	0.99	1.43e−5	1.54	8.47e−5	1.18	1.28e−2	0.99
	64	8.35e−6	1.12	4.43e−5	0.94	6.40e−3	1.00	8.35e−6	1.12	4.43e−5	0.94	6.40e−3	1.00
2	4	8.31e−6	–	9.82e−5	–	8.54e−3	–	8.31e−6	–	9.82e−5	–	8.54e−3	–
	8	2.57e−6	2.83	1.91e−5	2.36	2.24e−3	1.93	2.57e−6	2.83	1.91e−5	2.36	2.24e−3	1.93
	16	7.00e−7	2.71	4.43e−6	2.11	5.69e−4	1.97	7.00e−7	2.71	4.43e−6	2.11	5.69e−4	1.97
	32	1.81e−7	2.42	1.10e−6	2.01	1.43e−4	1.99	1.81e−7	2.42	1.10e−6	2.01	1.43e−4	1.99
	64	4.57e−8	2.16	2.75e−7	2.00	3.60e−5	2.00	4.57e−8	2.16	2.75e−7	2.00	3.60e−5	2.00
3	4	3.16e−7	–	5.56e−6	–	9.39e−4	–	3.16e−7	–	5.56e−6	–	9.39e−4	–
	8	1.01e−8	3.83	3.95e−7	3.81	1.28e−4	2.87	1.01e−8	3.83	3.95e−7	3.81	1.28e−4	2.87
	16	1.57e−9	3.93	2.72e−8	3.86	1.65e−5	2.96	1.57e−9	3.93	2.72e−8	3.86	1.65e−5	2.96
	32	3.19e−10	3.91	2.46e−9	3.47	2.07e−6	2.99	3.2e−10	3.91	2.47e−9	3.46	2.07e−6	2.99
	64	5.28e−11	3.67	3.24e−10	2.92	2.60e−7	2.99	5.32e−11	3.67	3.33e−10	2.89	2.60e−7	2.99

Table 10 Error and convergence rate for the HDG methods for the Example 2 with $R = \sqrt{2} + 0.1$

Degree	Mesh	Newton-HDG method						Fixed point-HDG method					
		$\|u - u_h\|$		$\|\boldsymbol{q} - \boldsymbol{q}_h\|$		$\|\boldsymbol{H} - \boldsymbol{H}_h\|$		$\|u - u_h\|$		$\|\boldsymbol{q} - \boldsymbol{q}_h\|$		$\|\boldsymbol{H} - \boldsymbol{H}_h\|$	
p	n	Error	Order	Error	Order	Error	Order	Error	Order	Error	Order	Error	Order
1	4	1.79e−3	–	1.14e−2	–	5.27e−1	–	1.79e−3	–	1.14e−2	–	5.27e−1	–
	8	4.49e−4	1.84	4.01e−3	1.50	3.46e−1	0.61	4.49e−4	1.84	4.01e−3	1.50	3.46e−1	0.61
	16	6.87e−5	1.98	1.13e−3	1.83	2.01e−1	0.78	6.87e−5	1.98	1.13e−3	1.83	2.01e−1	0.78
	32	3.31e−5	1.82	3.13e−4	1.85	1.09e−1	0.89	3.31e−5	1.82	3.13e−4	1.85	1.09e−1	0.89
	64	2.70e−5	1.20	1.51e−4	1.05	5.64e−2	0.95	2.70e−5	1.20	1.51e−4	1.05	5.64e−2	0.95
2	4	1.35e−4	–	1.76e−3	–	2.10e−1	–	1.35e−4	–	1.76e−3	–	2.10e−1	–
	8	6.99e−5	2.39	6.67e−4	1.40	8.49e−2	1.31	6.99e−5	2.39	6.67e−4	1.40	8.49e−2	1.31
	16	2.04e−5	2.33	2.13e−4	1.65	2.77e−2	1.62	2.04e−5	2.33	2.13e−4	1.65	2.77e−2	1.62
	32	5.23e−6	2.18	5.75e−5	1.89	7.88e−3	1.81	5.23e−6	2.18	5.75e−5	1.89	7.88e−3	1.81
	64	1.32e−6	2.06	1.47e−5	1.96	2.09e−3	1.91	1.32e−6	2.06	1.47e−5	1.96	2.09e−3	1.91
3	4	8.68e−5	–	6.98e−4	–	8.27e−2	–	8.68e−5	–	6.98e−4	–	8.27e−2	–
	8	8.09e−6	3.16	1.12e−4	2.64	2.28e−2	1.86	8.09e−6	3.16	1.12e−4	2.64	2.28e−2	1.86
	16	5.52e−7	3.57	1.23e−5	3.18	4.37e−3	2.38	5.52e−7	3.57	1.23e−5	3.18	4.37e−3	2.38
	32	3.02e−8	3.83	9.56e−7	3.69	6.53e−4	2.74	3.02e−8	3.83	9.56e−7	3.69	6.53e−4	2.74
	64	1.33e−9	3.95	5.98e−8	4.00	8.66e−5	2.91	1.33e−9	3.95	5.98e−8	4.00	8.66e−5	2.91

Table 11 Error and convergence rate for the HDG methods for the Example 2 with $R = \sqrt{2} + 0.01$

Degree	Mesh	Newton-HDG method						Fixed point-HDG method					
		$\|u - u_h\|$		$\|\boldsymbol{q} - \boldsymbol{q}_h\|$		$\|\boldsymbol{H} - \boldsymbol{H}_h\|$		$\|u - u_h\|$		$\|\boldsymbol{q} - \boldsymbol{q}_h\|$		$\|\boldsymbol{H} - \boldsymbol{H}_h\|$	
p	n	Error	Order	Error	Order	Error	Order	Error	Order	Error	Order	Error	Order
1	16	9.17e−4	–	1.62e−2	–	1.8e−0	–	9.17e−4	–	1.62e−2	–	1.8e−0	–
	32	2.41e−4	1.85	7.71e−3	1.07	1.61e−0	0.16	2.41e−4	1.85	7.71e−3	1.07	1.61e−0	0.16
	64	6.28e−5	1.88	3.01e−3	1.36	1.18e−0	0.44	6.28e−5	1.88	3.01e−3	1.36	1.18e−0	0.44
	128	2.45e−5	1.51	9.79e−4	1.62	7.35e−1	0.68	2.45e−5	1.51	9.79e−4	1.62	7.35e−1	0.68
	256	1.31e−5	1.04	2.83e−4	1.79	4.10e−1	0.84	1.31e−5	1.04	2.83e−4	1.79	4.10e−1	0.84
2	16	9.57e−5	–	3.19e−3	–	1.21e−0	–	9.57e−5	–	3.19e−3	–	1.21e−0	–
	32	3.60e−5	2.45	1.33e−3	1.27	7.23e−1	0.74	3.60e−5	2.45	1.33e−3	1.27	7.23e−1	0.74
	64	9.89e−6	2.67	5.37e−4	1.30	3.35e−1	1.11	9.89e−6	2.67	5.37e−4	1.30	3.35e−1	1.11
	128	2.44e−6	2.43	1.56e−4	1.78	1.21e−1	1.47	2.44e−6	2.43	1.56e−4	1.78	1.21e−1	1.47
	256	6.03e−7	2.15	4.02e−5	1.96	3.66e−2	1.73	6.03e−7	2.15	4.02e−5	1.96	3.66e−2	1.73
3	16	1.04e−4	–	3.95e−3	–	6.84e−1	–	1.04e−4	–	3.95e−3	–	6.84e−1	–
	32	1.05e−5	3.12	9.06e−4	2.12	3.28e−1	1.06	1.05e−5	3.12	9.06e−4	2.12	3.28e−1	1.06
	64	8.19e−7	3.31	1.39e−4	2.70	1.06e−1	1.63	8.19e−7	3.31	1.39e−4	2.70	1.06e−1	1.63
	128	5.29e−8	3.53	1.40e−5	3.31	2.29e−2	2.21	5.29e−8	3.53	1.40e−5	3.31	2.28e−2	2.21
	256	3.22e−9	3.77	1.00e−6	3.80	3.63e−3	2.65	3.22e−9	3.77	1.00e−6	3.80	3.63e−3	2.65

Table 12 Number of iterations required to reach convergence for the HDG methods for the Example 2

Mesh	Newton-HDG method									Fixed point-HDG method								
	R=2			$R=\sqrt{2}+0.1$			$R=\sqrt{2}+0.01$			R=2			$R=\sqrt{2}+0.1$			$R=\sqrt{2}+0.01$		
Level	P1	P2	P3	P1	P2	P3	P1	P2	P3	P1	P2	P3	P1	P2	P3	P1	P2	P3
1	6	6	6	6	6	6	7	7	7	27	30	31	29	34	39	35	59	72
2	6	6	6	6	6	6	7	7	7	29	31	32	33	38	43	39	54	84
3	6	6	6	7	6	6	7	7	7	30	32	33	37	42	43	47	66	97
4	7	6	6	7	6	6	7	7	7	32	33	34	40	44	46	54	89	121
5	7	6	6	7	6	6	7	7	7	33	34	35	43	46	49	60	112	146

source density $\rho(\boldsymbol{x})$ may be discontinuous and even vanish. The target density $\rho'(\boldsymbol{x}')$ must be strictly positive and Lipschitz continuous. The OT problem is to find a map $\boldsymbol{\phi} : \Omega \to \Omega'$ such that it minimizes the following functional

$$\inf_{\boldsymbol{\phi}\in\mathcal{M}} \int_{\Omega} \|\boldsymbol{x} - \boldsymbol{\phi}(\boldsymbol{x})\|^2 \rho(\boldsymbol{x}) d\boldsymbol{x}, \tag{15}$$

where

$$\mathcal{M} = \{\boldsymbol{\phi} : \Omega \to \Omega',\ \rho'(\boldsymbol{\phi}(\boldsymbol{x})) \det(\nabla\boldsymbol{\phi}(\boldsymbol{x})) = \rho(\boldsymbol{x}),\ \forall \boldsymbol{x} \in \Omega\}, \tag{16}$$

is the set of mappings which map the source density $\rho(\boldsymbol{x})$ onto the target density $\rho'(\boldsymbol{x}')$.

In [2], Brenier gave the proof of the existence and uniqueness of the solution of the OT problem. Furthermore, the optimal map $\boldsymbol{\phi}$ can be written as the gradient of a unique (up to a constant) convex potential u, so that $\boldsymbol{\phi}(\boldsymbol{x}) = \nabla u(\boldsymbol{x})$, $\Delta u(\boldsymbol{x}) > 0$. Substituting $\boldsymbol{\phi}(\boldsymbol{x}) = \nabla u(\boldsymbol{x})$ into (16) results in the Monge–Ampère equation

$$\rho'(\nabla u(\boldsymbol{x})) \det(D^2 u(\boldsymbol{x})) = \rho(\boldsymbol{x}) \quad \text{in } \Omega, \tag{17}$$

along with the restriction that u is convex. The equation lacks standard boundary conditions. However, it is geometrically constrained by the fact that the gradient map takes $\partial\Omega$ to $\partial\Omega'$:

$$\nabla u(\boldsymbol{x}) \in \partial\Omega', \quad \forall \boldsymbol{x} \in \Omega. \tag{18}$$

This constraint is referred to as the second boundary value problem for the Monge–Ampère equation. In [5], the constraint (18) is replaced with a Hamilton–Jacobi equation on the boundary

$$H(\nabla u(\boldsymbol{x})) := \operatorname{dist}(\nabla u(\boldsymbol{x}), \partial\Omega') = 0, \quad \forall \boldsymbol{x} \in \partial\Omega. \tag{19}$$

If the boundary $\partial\Omega'$ can be expressed by $\partial\Omega' = \{\boldsymbol{x}' \in \Omega' : g(\boldsymbol{x}') = 0\}$ then the Hamilton–Jacobi equation reduces to the following Neumann boundary condition

$$g(\nabla u(\boldsymbol{x})) = 0, \quad \forall \boldsymbol{x} \in \partial\Omega. \tag{20}$$

This boundary condition simplifies to a linear Neumann boundary condition when $g(\cdot)$ is a linear function, that is, when the boundary $\partial\Omega'$ is flat. For certain problems where densities are periodic, it is natural and convenient to use periodic boundary conditions instead.

3.2 Adaptive Mesh Redistribution

One approach to mesh adaptation is the equidistribution principle that equidistributes the target density function ρ' so that the source density ρ is uniform on Ω [27, 31]. The equidistribution principle leads to a constant source density $\rho(\boldsymbol{x}) = \theta$, where the constant θ is given by $\theta = \int_{\Omega'} \rho'(\boldsymbol{x}')d\boldsymbol{x}' / \int_{\Omega} d\boldsymbol{x}$. Using the optimal transport theory, the optimal mesh is sought by solving the Monge–Ampère equation with the Neumann boundary condition:

$$\begin{aligned} \det(D^2 u(\boldsymbol{x})) &= f(\nabla u(\boldsymbol{x})), \quad \text{in } \Omega, \\ g(\nabla u(\boldsymbol{x})) &= 0, \qquad\quad \text{on } \partial\Omega, \\ \int_{\Omega} u(\boldsymbol{x}) d\boldsymbol{x} &= 0, \end{aligned} \tag{21}$$

where $f(\nabla u(\boldsymbol{x})) = \theta / \rho'(\nabla u(\boldsymbol{x}))$. The gradient of u gives us the desired mesh.

In two dimensions, we can rewrite the above equation as a first-order system of equations

$$\begin{aligned} \boldsymbol{H} - \nabla \boldsymbol{q} &= 0, \text{ in } \Omega, \\ \boldsymbol{q} - \nabla u &= 0, \text{ in } \Omega, \\ s(\boldsymbol{H}, \boldsymbol{q}) - \nabla \cdot \boldsymbol{q} &= 0, \text{ in } \Omega, \\ g(\boldsymbol{q}) &= 0, \text{ on } \partial\Omega, \\ \int_\Omega u(\boldsymbol{x}) d\boldsymbol{x} &= 0, \end{aligned} \tag{22}$$

where $s(\boldsymbol{H}, \boldsymbol{q}) = \sqrt{H_{11}^2 + H_{22}^2 + H_{12}^2 + H_{21}^2 + 2f(\boldsymbol{q})}$. The system (22) differs from (3) in the source term and the boundary condition. In order to solve (22), we will need to extend the HDG methods described in the previous section.

3.3 HDG Formulation

The HDG discretization of the system (22) is to find $(\boldsymbol{H}_h, \boldsymbol{q}_h, u_h, \widehat{u}_h) \in \boldsymbol{W}_h^p \times \boldsymbol{V}_h^p \times U_h^p \times M_h^p$ such that

$$\begin{aligned} (\boldsymbol{H}_h, \boldsymbol{G})_{\mathcal{T}_h} + \left(\boldsymbol{q}_h, \nabla \cdot \boldsymbol{G}\right)_{\mathcal{T}_h} - \langle \widehat{\boldsymbol{q}}_h, \boldsymbol{G} \cdot \boldsymbol{n} \rangle_{\partial\mathcal{T}_h} &= 0, \\ \left(\boldsymbol{q}_h, \boldsymbol{v}\right)_{\mathcal{T}_h} + (u_h, \nabla \cdot \boldsymbol{v})_{\mathcal{T}_h} - \langle \widehat{u}_h, \boldsymbol{v} \cdot \boldsymbol{n} \rangle_{\partial\mathcal{T}_h} &= 0, \\ \left(\boldsymbol{q}_h, \nabla w\right)_{\mathcal{T}_h} - \langle \widehat{\boldsymbol{q}}_h \cdot \boldsymbol{n}, w \rangle_{\partial\mathcal{T}_h} + (s(\boldsymbol{H}_h, \boldsymbol{q}_h), w)_{\mathcal{T}_h} &= 0, \\ \langle \widehat{\boldsymbol{q}}_h \cdot \boldsymbol{n}, \mu \rangle_{\partial\mathcal{T}_h \setminus \partial\Omega} + \langle g(\boldsymbol{q}_h) + \tau(\widehat{u}_h - u_h), \mu \rangle_{\partial\Omega} &= 0, \\ (u_h, 1)_{\mathcal{T}_h} &= 0, \end{aligned} \tag{23}$$

for all $(\boldsymbol{G}, \boldsymbol{v}, w, \mu) \in \boldsymbol{W}_h^p \times \boldsymbol{V}_h^p \times U_h^p \times M_h^p$, where

$$\widehat{\boldsymbol{q}}_h = \boldsymbol{q}_h - \tau(u_h - \widehat{u}_h)\boldsymbol{n}, \quad \text{on } \mathcal{E}_h. \tag{24}$$

The approximate gradient $\boldsymbol{q}_h$ represents a mesh that adapts to the target density function ρ'. Since $\boldsymbol{q}_h$ is a discontinuous field, we need to postprocess it to obtain a conformal continuous mesh. In particular, we simply average the duplicate degrees of freedom of a DG scalar field to obtain a continuous scalar field of the same polynomial degree. This postprocessing is efficient and easy to implement. We are going to describe the Newton method and the fixed-point method to solve the HDG formulation.

3.4 Newton Method

For each Newton step l, we compute $\left(\delta\boldsymbol{H}_h^l, \delta\boldsymbol{q}_h^l, \delta u_h^l, \delta\widehat{u}_h^l\right) \in \boldsymbol{W}_h^p \times \boldsymbol{V}_h^p \times U_h^p \times M_h^p$ by solving the following linear system

$$\left(\delta\boldsymbol{H}_h^l, \boldsymbol{G}\right)_{\mathcal{T}_h} + \left(\delta\boldsymbol{q}_h^l, \nabla \cdot \boldsymbol{G}\right)_{\mathcal{T}_h} - \left\langle \delta\widehat{\boldsymbol{q}}_h^l, \boldsymbol{G} \cdot \boldsymbol{n} \right\rangle_{\partial\mathcal{T}_h} = r_1(\boldsymbol{G}), \tag{25a}$$

$$\left(\delta\boldsymbol{q}_h^l, \boldsymbol{v}\right)_{\mathcal{T}_h} + \left(\delta u_h^l, \nabla \cdot \boldsymbol{v}\right)_{\mathcal{T}_h} - \left\langle \delta\widehat{u}_h^l, \boldsymbol{v} \cdot \boldsymbol{n} \right\rangle_{\partial\mathcal{T}_h} = r_2(\boldsymbol{v}), \tag{25b}$$

$$\begin{aligned} &(\partial s_{\boldsymbol{H}}(\boldsymbol{H}_h^l, \boldsymbol{q}_h^l)\delta\boldsymbol{H}_h^l, w)_{\mathcal{T}_h} + (\partial s_{\boldsymbol{q}}(\boldsymbol{H}_h^l, \boldsymbol{q}_h^l)\delta\boldsymbol{q}_h^l, w)_{\mathcal{T}_h} \\ &+ \left(\delta\boldsymbol{q}_h^l, \nabla w\right)_{\mathcal{T}_h} - \left\langle \delta\widehat{\boldsymbol{q}}_h^l \cdot \boldsymbol{n}, w \right\rangle_{\partial\mathcal{T}_h} = r_3(w), \end{aligned} \tag{25c}$$

$$\left\langle \delta\widehat{\boldsymbol{q}}_h^l \cdot \boldsymbol{n}, \mu \right\rangle_{\partial\mathcal{T}_h \setminus \partial\Omega} + \left\langle \partial g_{\boldsymbol{q}}(\boldsymbol{q}_h^l) \cdot \delta\boldsymbol{q}_h^l + \tau(\delta\widehat{u}_h^l - \delta u_h^l), \mu \right\rangle_{\partial\Omega} = r_4(\mu), \tag{25d}$$

$$(\delta u_h^l, 1)_{\mathcal{T}_h} = r_5 \tag{25e}$$

for all $(\boldsymbol{G}, \boldsymbol{v}, w, \mu) \in \boldsymbol{W}_h^p \times \boldsymbol{V}_h^p \times U_h^p \times M_h^p$, the right-hand side residuals are given by

$$r_1(\boldsymbol{G}) = -\left(\boldsymbol{H}_h^l, \boldsymbol{G}\right)_{\mathcal{T}_h} - \left(\boldsymbol{q}_h^l, \nabla \cdot \boldsymbol{G}\right)_{\mathcal{T}_h} + \left\langle \widehat{\boldsymbol{q}}_h^l, \boldsymbol{G} \cdot \boldsymbol{n}\right\rangle_{\partial \mathcal{T}_h} \tag{26a}$$

$$r_2(\boldsymbol{v}) = -\left(\boldsymbol{q}_h^l, \boldsymbol{v}\right)_{\mathcal{T}_h} - \left(u_h^l, \nabla \cdot \boldsymbol{v}\right)_{\mathcal{T}_h} + \left\langle \widehat{u}_h^l, \boldsymbol{v} \cdot \boldsymbol{n}\right\rangle_{\partial \mathcal{T}_h} \tag{26b}$$

$$r_3(w) = -\left(\boldsymbol{q}_h^l, \nabla w\right)_{\mathcal{T}_h} + \left\langle \widehat{\boldsymbol{q}}_h^l \cdot \boldsymbol{n}, w\right\rangle_{\partial \mathcal{T}_h} - (s(\boldsymbol{H}_h^l, \boldsymbol{q}_h^l), w)_{\mathcal{T}_h} \tag{26c}$$

$$r_4(\mu) = -\left\langle \widehat{\boldsymbol{q}}_h^l \cdot \boldsymbol{n}, \mu\right\rangle_{\partial \mathcal{T}_h \setminus \partial \Omega} - \left\langle g(\boldsymbol{q}_h^l) + \tau(\widehat{u}_h^l - u_h^l), \mu\right\rangle_{\partial \Omega} \tag{26d}$$

$$r_5 = -(u_h^l, 1)_{\mathcal{T}_h}. \tag{26e}$$

Here $\partial s_{\boldsymbol{H}}(\boldsymbol{H}, \boldsymbol{q})$ and $\partial s_{\boldsymbol{q}}(\boldsymbol{H}, \boldsymbol{q})$ denote the partial derivatives of $s(\boldsymbol{H}, \boldsymbol{q})$ with respect to $\boldsymbol{H}$ and $\boldsymbol{q}$, respectively. And $\partial g_{\boldsymbol{q}}(\boldsymbol{q})$ denotes the partial derivatives of $g(\boldsymbol{q})$ with respect to $\boldsymbol{q}$.

At each step of the Newton method, the linearization (25) gives the following matrix system to be solved

$$\begin{pmatrix} \mathbb{A}^l & \mathbb{B}^l \\ \mathbb{C}^l & \mathbb{D}^l \\ \mathbb{E}^l & 0 \end{pmatrix} \begin{pmatrix} \delta U^l \\ \delta \widehat{U}^l \end{pmatrix} = \begin{pmatrix} R_{123}^l \\ R_4^l \\ r_5 \end{pmatrix} \tag{27}$$

where δU^l and $\delta \widehat{U}^l$ are the vectors of degrees of freedom of $(\delta \boldsymbol{H}_h^l, \delta \boldsymbol{q}_h^l, \delta u_h^l)$ and $\delta \widehat{u}_h^l$, respectively. The system (27) is first solved for the traces only $\delta \widehat{U}^l$:

$$\begin{pmatrix} \mathbb{K}^l \\ \mathbb{M}^l \end{pmatrix} \delta \widehat{U}^l = \begin{pmatrix} R^l \\ r_5 \end{pmatrix} \tag{28}$$

where

$$\mathbb{K}^l = \mathbb{D}^l - \mathbb{C}^l \left(\mathbb{A}^l\right)^{-1} \mathbb{B}^l, \qquad \mathbb{M}^l = -\mathbb{E}^l \left(\mathbb{A}^l\right)^{-1} \mathbb{B}^l, \qquad R^l = R_4^l - \mathbb{C}^l \left(\mathbb{A}^l\right)^{-1} R_{123}^l \tag{29}$$

Once $\delta \widehat{U}^l$ is known, the other unknowns δU^l are then retrieved element-wise.

3.5 Fixed-Point Method

To devise the fixed-point method for solving (22), we describe how to deal with the boundary condition $g(\boldsymbol{q}) = 0$. For any given $\boldsymbol{q}^{\ell-1}$, we linearize it around $\boldsymbol{q}^{\ell-1}$ to obtain

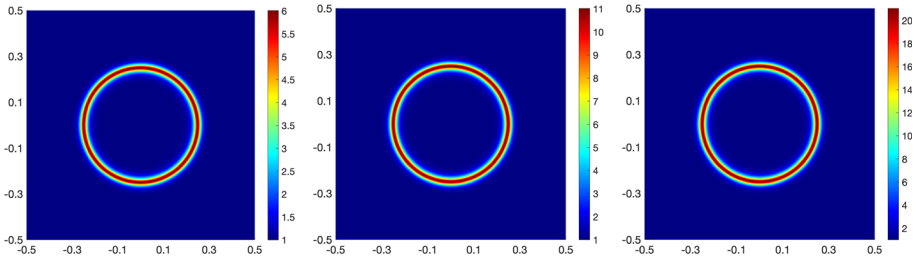
$$g(\boldsymbol{q}^{l-1}) + \partial g_{\boldsymbol{q}}(\boldsymbol{q}^{l-1}) \cdot \left(\boldsymbol{q}^l - \boldsymbol{q}^{l-1}\right) = 0. \tag{30}$$

Starting from an initial guess $(\boldsymbol{H}_h^0, \boldsymbol{q}_h^0, u_h^0)$ we find $(\boldsymbol{q}_h^l, u_h^l, \widehat{u}_h^l) \in \boldsymbol{V}_h^p \times U_h^p \times M_h^p$ such that

$$\begin{aligned} \left(\boldsymbol{q}_h^l, \boldsymbol{v}\right)_{\mathcal{T}_h} + \left(u_h^l, \nabla \cdot \boldsymbol{v}\right)_{\mathcal{T}_h} - \left\langle \widehat{u}_h^l, \boldsymbol{v} \cdot \boldsymbol{n}\right\rangle_{\partial \mathcal{T}_h} &= 0, \\ \left(\boldsymbol{q}_h^l, \nabla w\right)_{\mathcal{T}_h} - \left\langle \widehat{\boldsymbol{q}}_h^l \cdot \boldsymbol{n}, w\right\rangle_{\partial \mathcal{T}_h} &= -(s(\boldsymbol{H}_h^{l-1}, \boldsymbol{q}_h^{l-1}), w)_{\mathcal{T}_h}, \\ \left\langle \widehat{\boldsymbol{q}}_h^l \cdot \boldsymbol{n}, \mu\right\rangle_{\partial \mathcal{T}_h \setminus \partial \Omega} + \left\langle \partial g_{\boldsymbol{q}}(\boldsymbol{q}_h^{l-1}) \cdot \boldsymbol{q}_h^l + \tau(\widehat{u}_h^l - u_h^l), \mu\right\rangle_{\partial \Omega} &= -\left\langle b(\boldsymbol{q}^{l-1}), \mu\right\rangle_{\partial \Omega}, \\ (u_h^l, 1)_{\mathcal{T}_h} &= 0, \end{aligned} \tag{31}$$

for all $(\boldsymbol{v}, w, \mu) \in \boldsymbol{V}_h^p \times U_h^p \times M_h^p$, and then compute $\boldsymbol{H}_h^l \in \boldsymbol{W}_h^p$ such that

$$\left(\boldsymbol{H}_h^l, \boldsymbol{G}\right)_{\mathcal{T}_h} = -\left(\boldsymbol{q}_h^l, \nabla \cdot \boldsymbol{G}\right)_{\mathcal{T}_h} + \left\langle \widehat{\boldsymbol{q}}_h^l, \boldsymbol{G} \cdot \boldsymbol{n}\right\rangle_{\partial \mathcal{T}_h}, \quad \forall \boldsymbol{G} \in \boldsymbol{W}_h^p. \tag{32}$$



(a) $\boldsymbol{\alpha}_1 = (5, 200, 0.25)$ (b) $\boldsymbol{\alpha}_2 = (10, 200, 0.25)$ (c) $\boldsymbol{\alpha}_3 = (20, 200, 0.25)$

Fig. 1 Three instances of the density function (33)

Note here that $b(\boldsymbol{q}_h^{l-1}) = g(\boldsymbol{q}_h^{l-1}) - \partial g_{\boldsymbol{q}}(\boldsymbol{q}_h^{l-1}) \cdot \boldsymbol{q}_h^{l-1}$, and that the numerical flux $\widehat{\boldsymbol{q}}_h^l$ is defined by (24).

At each step of the fixed-point HDG method, the weak formulation (31) yields the matrix system similar to (27). The global matrix for the degrees of freedom of $\widehat{u}_h^l$ is changed at each step because of the linearization of the nonlinear boundary condition $g(\boldsymbol{q}) = 0$. In this case, the fixed-point HDG method is no longer competitive to the Newton-HDG method. We note however that if $g(\cdot)$ is a linear function, then the global matrix will be unchanged at each fixed-point step. In this case, the global matrix can be formed prior to performing the fixed-point iteration. In any case, the Newton-HDG method is more efficient than the fixed-point HDG method since the former requires considerably less number of iterations than the latter.

3.6 Numerical Experiments

We give several examples of high-order meshes generated using the HDG methods for analytical density functions on both planar and curved domains. We compare the convergence of the HDG methods for these examples. We also demonstrate that our methods can generate smooth high-order meshes even at very high mesh resolutions.

Ring meshes on a square domain. We wish to generate meshes on a square domain $\Omega' = (-0.5, 0.5)^2$ with the following target density function:

$$\rho'(x', y') = 1 + \alpha_1 \operatorname{sech}^2\left(\alpha_2\left(x'^2 + y'^2 - \alpha_3^2\right)\right), \tag{33}$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3)$ determines the density function. This density function was introduced in [29]. We consider three instances of the density function (33) corresponding to $\boldsymbol{\alpha}_1 = (5, 200, 0.25)$, $\boldsymbol{\alpha}_2 = (10, 200, 0.25)$, and $\boldsymbol{\alpha}_3 = (20, 200, 0.25)$, as shown in Fig. 1.

First, a background mesh on which the optimal transport meshes are generated is a uniform grid of $50 \times 50 \times 2$ triangles on the square domain $\Omega = (-0.5, 0.5)^2$. Here we use polynomial degree $p = 3$ to represent the numerical solution. Figure 2 depicts the three high-order meshes generated for these density instances. We observe that as α_1 increases from 5, 10, to 20, it results in more elements concentrating into the ring.

To demonstrate that our methods can generate smooth high-order meshes even at very high mesh resolutions, we consider a background mesh as a uniform grid of 100×100 quadrilaterals with polynomial degree $p = 3$. Figure 3 depicts the three high-order meshes generated on this background mesh, while Fig. 4 shows the close-up view near the ring of

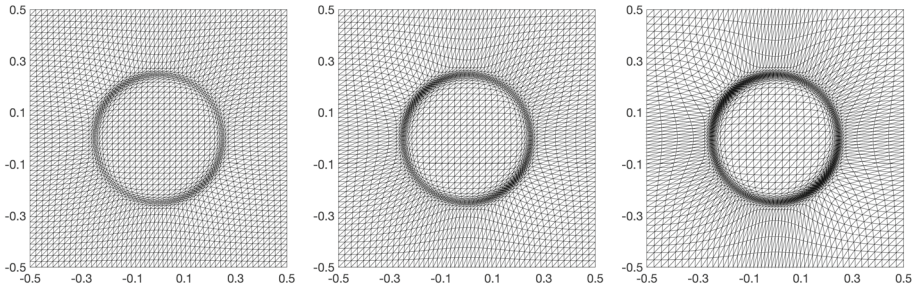


Fig. 2 Three high-order meshes are generated for the density functions shown in Fig. 1 using a uniform background mesh of $50 \times 50 \times 2$ triangles with $p = 3$

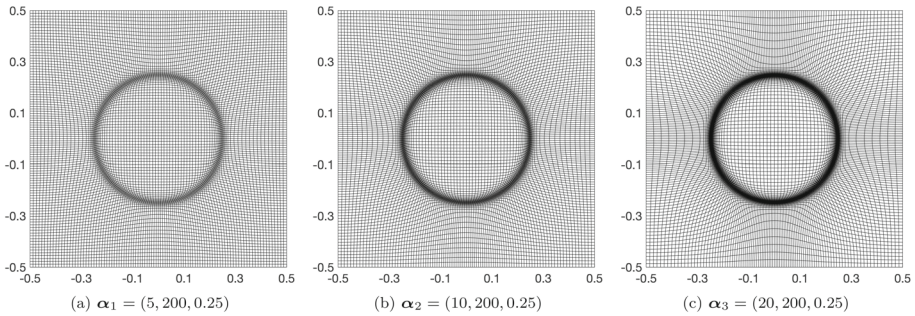


Fig. 3 Three high-order meshes are generated for the density functions shown in Fig. 1 using a uniform background mesh of 100×100 quadrilaterals with $p = 3$

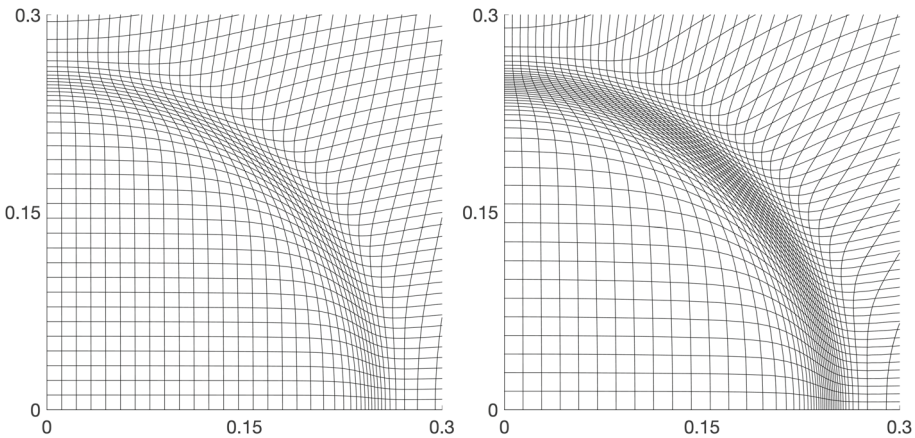
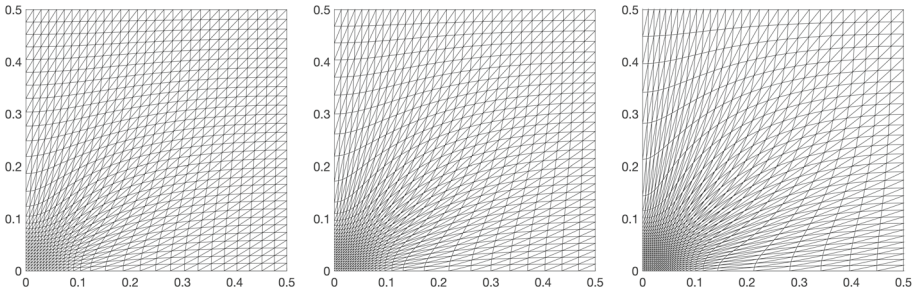


Fig. 4 Close-up view near the ring of the meshes shown in Fig. 3a (left) and in Fig. 3c (right)

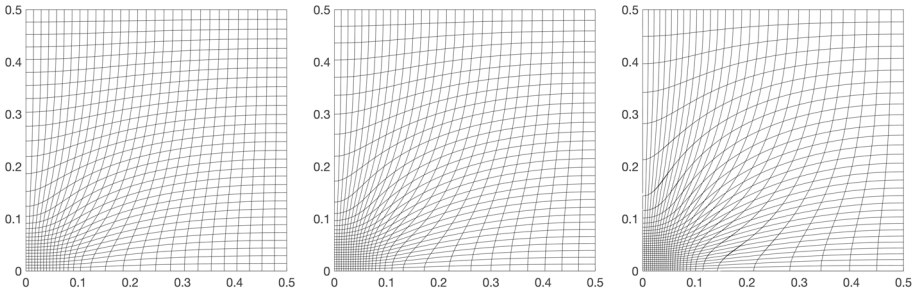
the first and third meshes. Despite there are many elements concentrating into the ring, the meshes are smooth and non-tangled.

Bell meshes on a square domain. We consider three new instances of the density function (33) corresponding to $\boldsymbol{\alpha}_4 = (10, 200, 0)$, $\boldsymbol{\alpha}_5 = (20, 200, 0)$, and $\boldsymbol{\alpha}_6 = (40, 200, 0)$. Figure 5



(a) $\boldsymbol{\alpha}_4 = (10, 200, 0)$ (b) $\boldsymbol{\alpha}_5 = (20, 200, 0)$ (c) $\boldsymbol{\alpha}_6 = (40, 200, 0)$

Fig. 5 Three high-order meshes are generated for three instances of the density function (33) using a uniform background mesh of $60 \times 60 \times 2$ triangles with $p = 3$



(a) $\boldsymbol{\alpha}_4 = (10, 200, 0)$ (b) $\boldsymbol{\alpha}_5 = (20, 200, 0)$ (c) $\boldsymbol{\alpha}_6 = (40, 200, 0)$

Fig. 6 Three high-order meshes are generated for three instances of the density function (33) using a uniform background mesh of 60×60 quadrilaterals with $p = 3$

Table 13 Number of iterations required to reach convergence for the HDG methods for different instances of the density function (33) on uniform triangular and quadrilateral grids

Case	Newton-HDG method		Fixed point-HDG method	
	Triangular grid	Quadrilateral grid	Triangular grid	Quadrilateral grid
$\boldsymbol{\alpha}_1$	6	6	40	55
$\boldsymbol{\alpha}_2$	9	9	57	92
$\boldsymbol{\alpha}_3$	16	11	79	138
$\boldsymbol{\alpha}_4$	8	7	65	65
$\boldsymbol{\alpha}_5$	9	10	88	87
$\boldsymbol{\alpha}_6$	12	15	126	122

depicts the three high-order meshes that are generated for these instances on a uniform mesh of $60 \times 60 \times 2$ triangles with $p = 3$, while Fig. 6 shows the meshes on a uniform mesh of 60×60 quadrilaterals. We see that as α_1 increases from 10, 20, to 40, it results in more elements concentrating into the origin (0, 0).

Shock-aligned meshes on a cylindrical domain. High speed flows past a unit circular cylinder is a popular test case in computational fluid dynamics [74–80]. For high Mach numbers, a strong bow shock forms in front of the cylinder. Therefore, it is important to

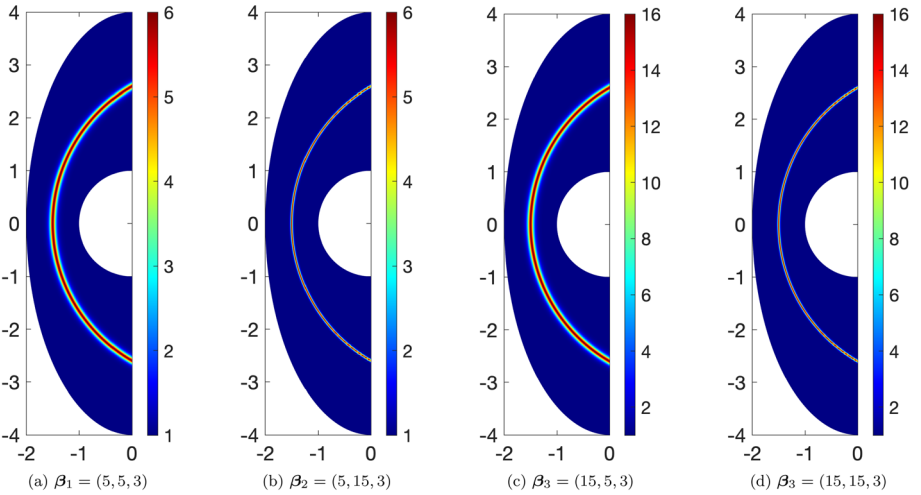


Fig. 7 Four instances of the density function

generate high-quality meshes to align the bow shock. The geometry is described by a half unit cylinder $x^2 + y^2 = 1$ and a half elliptical boundary $x^2/2^2 + y^2/4^2 = 1$. To represent a bow shock, the following target density function is considered

$$\rho'(x', y') = 1 + \beta_1 \text{sech}^2\left(\beta_2\left((x' - 1.5)^2 + y'^2 - \beta_3^2\right)\right), \tag{34}$$

where $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3)$ determines the density function. We consider four instances of the density function (34) corresponding to $\boldsymbol{\beta}_1 = (5, 5, 3)$, $\boldsymbol{\beta}_2 = (5, 15, 3)$, $\boldsymbol{\beta}_3 = (15, 5, 3)$, and $\boldsymbol{\beta}_4 = (15, 15, 3)$, as shown in Fig. 7.

The background mesh on which the optimal transport meshes are generated is a quadrilateral grid of 40×60 elements with polynomial degree $p = 4$. Figure 8 depicts the four high-order meshes generated for these density instances shown in Fig. 7, while Fig. 9 shows the close-up view near the shock region of the last two meshes. We see that increasing the amplitude of the density function in the shock region results in more elements concentrating into the shock region. Furthermore, widening the density function increases the thickness of the shock region. We emphasize that the generated meshes are high-order, smooth, non-tangled, and conforming to the curved boundary of the physical domain.

4 Conclusion

We have presented two hybridizable discontinuous Galerkin methods for numerically solving the Monge–Ampère equation. The first HDG method is based on the Newton method, whereas the second HDG method is based on the fixed-point method. Numerical results were presented to demonstrate that the convergence and accuracy of the HDG methods. The Newton-HDG method is more efficient since it requires less iterations than the fixed-point HDG method. The numerical results showed that the HDG methods yield the convergence rate of $O(h^p)$ for the approximate scalar variable u_h, the approximate gradient $\boldsymbol{q}_h$, and the approximate Hessian $\boldsymbol{H}_h$. The convergence rate of $O(h^p)$ for the approximate gradient and Hessian is an attractive feature of the HDG methods. Furthermore, the HDG methods were extended

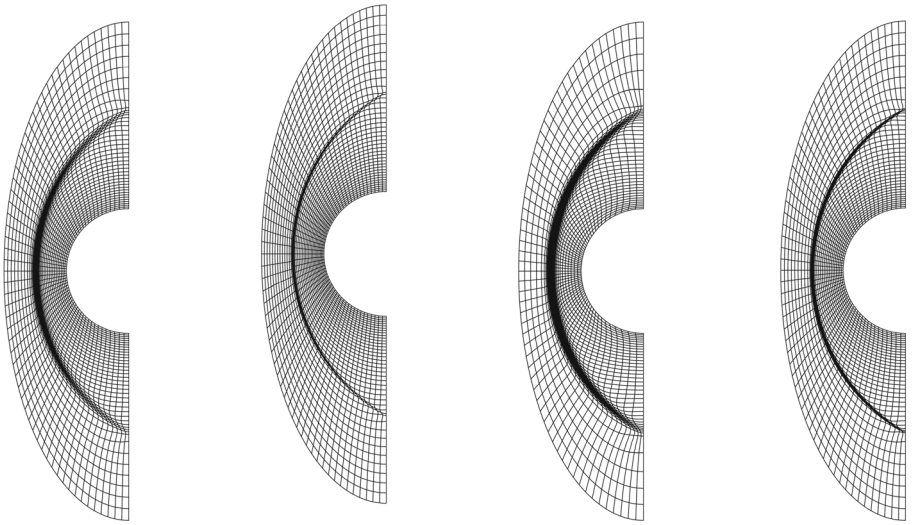


Fig. 8 High-order quadrilateral meshes correspond to the four density instances shown in Fig. 7

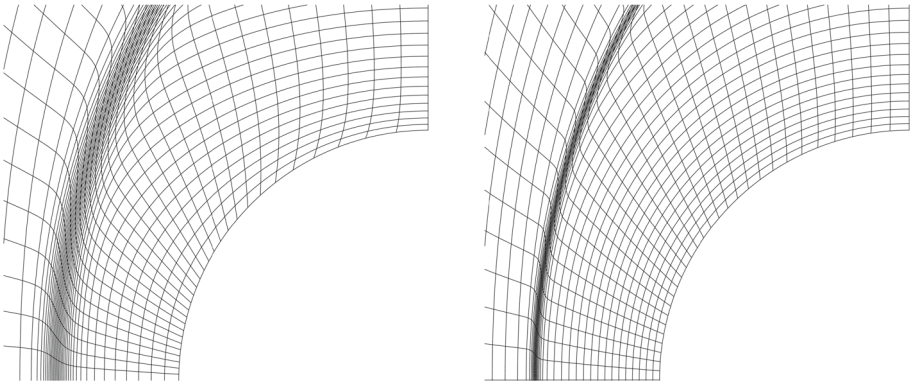


Fig. 9 Close-up view near the cylinder of the meshes shown in Fig. 8c (left) and in Fig. 8d (right)

to generate r-adaptive high-order meshes based on equidistribution of a given scalar density function via the optimal transport theory. Several numerical experiments were presented to illustrate the generation of smooth high-order meshes on planar and curved domains.

We would like to construct a stability analysis of the proposed HDG method, that would allow us to properly define the stabilization function. Choosing the optimal stabilization function is still a theoretical issue to be investigated for nonlinear PDEs [73]. It is also important to analyze the HDG methods to understand the convergence rates observed in this paper. Furthermore, we would like to extend our methodology to generate r-adaptive meshes for flow problems in computational fluid dynamics. Therefore, three-dimensional

mesh generation based on the optimal transport theory is an important topic to be addressed in future work.

Funding This work was supported by the United States Department of Energy under contract DE-NA0003965, the National Science Foundation under grant number NSF-PHY-2028125, the Air Force Office of Scientific Research under Grant No. FA9550-22-1-0356, and the MIT Portugal program under the seed grant number 6950138.

Data Availability The datasets generated in this study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Feng, X., Glowinski, R., Neilan, M.: Recent developments in numerical methods for fully nonlinear second order partial differential equations. SIAM Rev. **55**(2), 205–267 (2013). https://doi.org/10.1137/110825960
2. Brenier, Y.: Polar factorization and monotone rearrangement of vector-valued functions. Commun. Pure Appl. Math. **44**(4), 375–417 (1991). https://doi.org/10.1002/cpa.3160440402
3. Benamou, J.D., Brenier, Y.: A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem. Numer. Math. **84**(3), 375–393 (2000). https://doi.org/10.1007/s002110050002
4. Benamou, J.D., Froese, B.D., Oberman, A.M.: Two numerical methods for the elliptic Monge–Ampère equation. ESAIM Math. Modell. Numer. Anal. **44**(4), 737–758 (2010). https://doi.org/10.1051/m2an/2010017
5. Benamou, J.D., Froese, B.D., Oberman, A.M.: Numerical solution of the optimal transportation problem using the Monge–Ampère equation. J. Comput. Phys. **260**, 107–126 (2014). https://doi.org/10.1016/j.jcp.2013.12.015
6. Caffarelli, L.A.: Interior W^2, p estimates for solutions of the Monge–Ampere equation. Ann. Math. **131**(1), 135 (1990). https://doi.org/10.2307/1971510
7. Dean, E.J., Glowinski, R.: Numerical methods for fully nonlinear elliptic equations of the Monge–Ampère type. Comput. Methods Appl. Mech. Eng. **195**(13–16), 1344–1386 (2006). https://doi.org/10.1016/j.cma.2005.05.023
8. Frisch, U., Matarrese, S., Mohayaee, R.: A reconstruction of the initial conditions of the universe by optimal mass transportation. Nature **417**(6886), 260–262 (2002). https://doi.org/10.1038/417260a. arXiv:astro-ph/0109483
9. Froese, B.D., Oberman, A.M.: Fast finite difference solvers for singular solutions of the elliptic Monge–Ampère equation. J. Comput. Phys. **230**(3), 818–834 (2011). https://doi.org/10.1016/j.jcp.2010.10.020
10. Oliker, V.: Mathematical aspects of design of beam shaping surfaces in geometrical optics. In: Trends in nonlinear analysis, pp. 193–224. Springer, Berlin (2003). https://doi.org/10.1007/978-3-662-05281-54
11. Prins, C.R., Ten Thije Boonkkamp, J.H., Van Roosmalen, J., Ijzerman, W.L., Tukker, T.W.: A Monge–Ampère-solver for free-form reflector design. SIAM J. Sci. Comput. **36**(3), B640–B660 (2014). https://doi.org/10.1137/130938876
12. Trudinger, N.S., Wang, X.-J.: The Monge–Ampere equation and its geometric applications. Handb. Geom. Anal. **I**, 467–524 (2008)
13. Awanou, G.: Standard finite elements for the numerical resolution of the elliptic Monge–Ampère equation: classical solutions. IMA J. Numer. Anal. **35**(3), 1150–1166 (2015). https://doi.org/10.1093/imanum/dru028
14. Böhmer, K.: On finite element methods for fully nonlinear elliptic equations of second order. SIAM J. Numer. Anal. **46**(3), 1212–1249 (2008). https://doi.org/10.1137/040621740
15. Brenner, S., Gudi, T., Neilan, M., Sung, L.-Y.: C0 penalty methods for the fully nonlinear Monge–Ampère equation. Math. Comput. **80**(276), 1979–1995 (2011). https://doi.org/10.1090/s0025-5718-2011-02487-7

16. Brenner, S.C., Neilan, M.: Finite element approximations of the three dimensional Monge–Ampère equation. ESAIM Math. Modell. Numer. Anal. **46**(5), 979–1001 (2012). https://doi.org/10.1051/m2an/2011067
17. Caboussat, A., Glowinski, R., Sorensen, D.C.: A least-squares method for the numerical solution of the Dirichlet problem for the elliptic Monge–Ampère equation in dimension two. ESAIM Control Optim. Calc. Var. **19**(3), 780–810 (2013). https://doi.org/10.1051/cocv/2012033
18. Feng, X., Neilan, M.: Finite element approximations of general fully nonlinear second order elliptic partial differential equations based on the vanishing moment method. Comput. Math. Appl. **68**(12), 2182–2204 (2014). https://doi.org/10.1016/j.camwa.2014.07.023
19. Feng, X., Neilan, M.: Mixed finite element methods for the fully nonlinear Monge-Ampère equation based on the vanishing moment method. SIAM J. Numer. Anal. **47**(2), 1226–1250 (2009). https://doi.org/10.1137/070710378. arXiv:0712.1241
20. Feng, X., Lewis, T.: Mixed interior penalty discontinuous Galerkin methods for fully nonlinear second order elliptic and parabolic equations in high dimensions. Numer. Methods Partial Differ. Equ. **30**(5), 1538–1557 (2014). https://doi.org/10.1002/num.21856
21. Feng, X., Jensen, M.: Convergent semi-Lagrangian methods for the Monge–Ampère equation on unstructured grids. SIAM J. Numer. Anal. **55**(2), 691–712 (2017). https://doi.org/10.1137/16M1061709
22. Feng, X., Lewis, T.: Nonstandard local discontinuous Galerkin methods for fully nonlinear second order elliptic and parabolic equations in high dimensions. J. Sci. Comput. **77**(3), 1534–1565 (2018). https://doi.org/10.1007/s10915-018-0765-z. arXiv:1801.05877
23. Froese, B.D.: A numerical method for the elliptic Monge–Ampère equation with transport boundary conditions. SIAM J. Sci. Comput. **34**(3), A1432–A1459 (2012). https://doi.org/10.1137/110822372. arXiv:1101.4981
24. Liu, H., Glowinski, R., Leung, S., Qian, J.: A finite element/operator-splitting method for the numerical solution of the three dimensional Monge–Ampère equation. J. Sci. Comput. **81**(3), 2271–2302 (2019). https://doi.org/10.1007/s10915-019-01080-4
25. Lakkis, O., Pryer, T.: A finite element method for nonlinear elliptic problems. SIAM J. Sci. Comput. **35**(4), A2025–A2045 (2013). https://doi.org/10.1137/120887655
26. Glowinski, R., Liu, H., Leung, S., Qian, J.: A finite element/operator-splitting method for the numerical solution of the two dimensional elliptic Monge–Ampère equation. J. Sci. Comput. **79**(1), 1–47 (2019). https://doi.org/10.1007/s10915-018-0839-y
27. Delzanno, G.L., Chacón, L., Finn, J.M., Chung, Y., Lapenta, G.: An optimal robust equidistribution method for two-dimensional grid adaptation based on Monge–Kantorovich optimization. J. Comput. Phys. **227**(23), 9841–9864 (2008). https://doi.org/10.1016/j.jcp.2008.07.020
28. Budd, C.J., Williams, J.F.: Moving mesh generation using the parabolic Monge–Ampère equation. SIAM J. Sci. Comput. **31**(5), 3438–3465 (2009). https://doi.org/10.1137/080716773
29. Budd, C.J., Russell, R.D., Walsh, E.: The geometry of r-adaptive meshes generated using optimal transport methods. J. Comput. Phys. **282**, 113–137 (2015). https://doi.org/10.1016/j.jcp.2014.11.007. arXiv:1409.5361
30. Browne, P.A., Budd, C.J., Piccolo, C., Cullen, M.: Fast three dimensional r-adaptive mesh redistribution. J. Comput. Phys. **275**, 174–196 (2014). https://doi.org/10.1016/j.jcp.2014.06.009
31. Chacón, L., Delzanno, G.L., Finn, J.M.: Robust, multidimensional mesh-motion based on Monge-Kantorovich equidistribution. J. Comput. Phys. **230**(1), 87–103 (2011). https://doi.org/10.1016/j.jcp.2010.09.013
32. Weller, H., Browne, P., Budd, C., Cullen, M.: Mesh adaptation on the sphere using optimal transport and the numerical solution of a Monge–Ampère type equation. J. Comput. Phys. **308**, 102–123 (2016). https://doi.org/10.1016/j.jcp.2015.12.018. arXiv:1512.02935
33. McRae, A.T., Cotter, C.J., Budd, C.J.: Optimal-transport-based mesh adaptivity on the plane and sphere using finite elements. SIAM J. Sci. Comput. **40**(2), A1121–A1148 (2018). https://doi.org/10.1137/16M1109515. arXiv:1612.08077
34. Sulman, M., Williams, J.F., Russell, R.D.: Optimal mass transport for higher dimensional adaptive grid generation. J. Comput. Phys. **230**(9), 3302–3330 (2011). https://doi.org/10.1016/j.jcp.2011.01.025
35. Sulman, M.H., Nguyen, T.B., Haynes, R.D., Huang, W.: Domain decomposition parabolic Monge–Ampère approach for fast generation of adaptive moving meshes. Comput. Math. Appl. **84**, 97–111 (2021). https://doi.org/10.1016/j.camwa.2020.12.007
36. Aparicio-Estrems, G., Gargallo-Peiró, A., Roca, X.: Combining high-order metric interpolation and geometry implicitization for curved r-adaption. CAD Comput. Aided Des. **157**, 103478 (2023). https://doi.org/10.1016/j.cad.2023.103478

37. Cockburn, B., Gopalakrishnan, J., Lazarov, R.: Unified hybridization of discontinuous Galerkin, mixed and continuous Galerkin methods for second order elliptic problems. SIAM J. Numer. Anal. **47**, 1319–1365 (2009)
38. Cockburn, B., Dong, B., Guzmán, J., Restelli, M., Sacco, R.: A hybridizable discontinuous Galerkin method for steady-state convection-diffusion-reaction problems. SIAM J. Sci. Comput. **31**(5), 3827–3846 (2009)
39. Nguyen, N.C., Peraire, J., Cockburn, B.: An implicit high-order hybridizable discontinuous Galerkin method for linear convection diffusion equations. J. Comput. Phys. **228**(9), 3232–3254 (2009). https://doi.org/10.1016/j.jcp.2009.01.030
40. Cockburn, B., Mustapha, K.: A hybridizable discontinuous Galerkin method for fractional diffusion problems. Numer. Math. **130**(2), 293–314 (2015). https://doi.org/10.1007/s00211-014-0661-x. arXiv:1409.7383
41. Nguyen, N.C., Peraire, J., Cockburn, B.: An implicit high-order hybridizable discontinuous Galerkin method for nonlinear convection diffusion equations. J. Comput. Phys. **228**(23), 8841–8855 (2009). https://doi.org/10.1016/j.jcp.2009.08.030
42. Ueckermann, M.P., Lermusiaux, P.F.: High-order schemes for 2D unsteady biogeochemical ocean models. Ocean Dyn. **60**(6), 1415–1445 (2010). https://doi.org/10.1007/s10236-010-0351-x
43. Cockburn, B., Gopalakrishnan, J.: The derivation of hybridizable discontinuous Galerkin methods for Stokes flow. SIAM J. Numer. Anal. **47**(2), 1092–1125 (2009). https://doi.org/10.1137/080726653
44. Cockburn, B., Gopalakrishnan, J., Nguyen, N.C., Peraire, J., Sayas, F.J.: Analysis of an HDG method for Stokes flow. Math. Comp. **80**, 723–760 (2011)
45. Cockburn, B., Nguyen, N.C., Peraire, J.: A comparison of HDG methods for Stokes flow. J. Sci. Comput. **45**(1–3), 215–237 (2010). https://doi.org/10.1007/s10915-010-9359-0
46. Nguyen, N.C., Peraire, J., Cockburn, B.: A hybridizable discontinuous Galerkin method for Stokes flow. Comput. Methods Appl. Mech. Eng. **199**(9–12), 582–597 (2010). https://doi.org/10.1016/j.cma.2009.10.007
47. Ahnert, T., Bärwolff, G.: Numerical comparison of hybridized discontinuous Galerkin and finite volume methods for incompressible flow. Int. J. Numer. Meth. Fluids **76**(5), 267–281 (2014). https://doi.org/10.1002/fld.3938
48. Nguyen, N.C., Peraire, J., Cockburn, B.: An implicit high-order hybridizable discontinuous Galerkin method for the incompressible Navier-Stokes equations. J. Comput. Phys. **230**(4), 1147–1170 (2011). https://doi.org/10.1016/j.jcp.2010.10.032
49. Nguyen, N. C., Peraire, J., Cockburn, B.: Hybridizable discontinuous Galerkin methods. In: Proceedings of the international conference on spectral and high order methods, pp. 63–84. Springer, Berlin (2009)
50. Nguyen, N. C., Peraire, J., Cockburn, B.: A hybridizable discontinuous Galerkin method for the incompressible Navier-Stokes equations. In: Proceedings of the 48th AIAA Aerospace Sciences Meeting and Exhibit, pp. AIAA 2010–362. Orlando, Florida (2010)
51. Rhebergen, S., Cockburn, B.: A space-time hybridizable discontinuous Galerkin method for incompressible flows on deforming domains. J. Comput. Phys. **231**(11), 4185–4204 (2012). https://doi.org/10.1016/j.jcp.2012.02.011
52. Rhebergen, S., Wells, G.N.: A hybridizable discontinuous Galerkin method for the Navier-Stokes equations with pointwise divergence-free velocity field. J. Sci. Comput. **76**(3), 1484–1501 (2018). https://doi.org/10.1007/s10915-018-0671-4. arXiv:1704.07569
53. Ueckermann, M.P., Lermusiaux, P.F.: Hybridizable discontinuous Galerkin projection methods for Navier-Stokes and Boussinesq equations. J. Comput. Phys. **306**, 390–421 (2016). https://doi.org/10.1016/j.jcp.2015.11.028
54. Ciucă, C., Fernandez, P., Christophe, A., Nguyen, N.C., Peraire, J.: Implicit hybridized discontinuous Galerkin methods for compressible magnetohydrodynamics. J. Comput. Phys. X **5**, 100042 (2020). https://doi.org/10.1016/j.jcpx.2019.100042
55. Fernandez, P., Nguyen, N.C., Peraire, J.: The hybridized discontinuous Galerkin method for Implicit Large-Eddy simulation of transitional turbulent flows. J. Comput. Phys. **336**, 308–329 (2017). https://doi.org/10.1016/j.jcp.2017.02.015
56. Franciolini, M., Fidkowski, K.J., Crivellini, A.: Efficient discontinuous Galerkin implementations and preconditioners for implicit unsteady compressible flow simulations. Comput. Fluids **203**, 104542 (2020). https://doi.org/10.1016/j.compfluid.2020.104542. arXiv:1812.04789
57. Moro, D., Nguyeny, N. C., Peraire, J.: Navier-stokes solution using Hybridizable discontinuous Galerkin methods. In: 20th AIAA computational fluid dynamics conference 2011, American Institute of Aeronautics and Astronautics, Honolulu, Hawaii, pp. AIAA–2011–3407 (2011). https://doi.org/10.2514/6.2011-3407. http://arc.aiaa.org/doi/abs/10.2514/6.2011-3407

58. Nguyen, N.C., Peraire, J.: Hybridizable discontinuous Galerkin methods for partial differential equations in continuum mechanics. J. Comput. Phys. **231**(18), 5955–5988 (2012). https://doi.org/10.1016/j.jcp.2012.02.033
59. Nguyen, N.C., Peraire, J., Cockburn, B.: A class of embedded discontinuous Galerkin methods for computational fluid dynamics. J. Comput. Phys. **302**, 674–692 (2015). https://doi.org/10.1016/j.jcp.2015.09.024
60. Vila-Pérez, J., Giacomini, M., Sevilla, R., Huerta, A.: Hybridisable discontinuous Galerkin formulation of compressible flows. Arch. Comput. Methods Eng. **28**(2), 753–784 (2021). https://doi.org/10.1007/s11831-020-09508-z
61. Vila-Pérez, J., Van Heyningen, R.L., Nguyen, N.-C., Peraire, J.: Exasim: generating discontinuous Galerkin codes for numerical solutions of partial differential equations on graphics processors. SoftwareX **20**, 101212 (2022). https://doi.org/10.1016/j.softx.2022.101212
62. Nguyen, N.C., Peraire, J., Cockburn, B.: Hybridizable discontinuous Galerkin methods for the time-harmonic Maxwell's equations. J. Comput. Phys. **230**(19), 7151–7175 (2011). https://doi.org/10.1016/j.jcp.2011.05.018
63. Sánchez, M.A., Du, S., Cockburn, B., Nguyen, N.-C., Peraire, J.: Symplectic Hamiltonian finite element methods for electromagnetics. Comput. Methods Appl. Mech. Eng. **396**, 114969 (2022). https://doi.org/10.1016/j.cma.2022.114969
64. Li, L., Lanteri, S., Perrussel, R.: A hybridizable discontinuous Galerkin method combined to a Schwarz algorithm for the solution of 3D time-harmonic Maxwell's equations. J. Comput. Phys. **256**, 563–581 (2014). https://doi.org/10.1016/j.jcp.2013.09.003
65. Soon, S.-C., Cockburn, B., Stolarski, H.K.: A hybridizable discontinuous Galerkin method for linear elasticity. Int. J. Numer. Meth. Eng. **80**(8), 1058–1092 (2009)
66. Cockburn, B., Shi, K.: Superconvergent HDG methods for linear elasticity with weakly symmetric stresses. IMA J. Numer. Anal. **33**(3), 747–770 (2013). https://doi.org/10.1093/imanum/drs020
67. Fu, G., Cockburn, B., Stolarski, H.: Analysis of an HDG method for linear elasticity. Int. J. Numer. Meth. Eng. **102**(3–4), 551–575 (2015)
68. Qiu, W., Shen, J., Shi, K.: An HDG method for linear elasticity with strong symmetric stresses. Math. Comput. **87**(309), 69–93 (2018)
69. Sánchez, M.A., Cockburn, B., Nguyen, N.C., Peraire, J.: Symplectic Hamiltonian finite element methods for linear elastodynamics. Comput. Methods Appl. Mech. Eng. **381**, 113843 (2021). https://doi.org/10.1016/j.cma.2021.113843
70. Cockburn, B., Shen, J.: An algorithm for stabilizing hybridizable discontinuous Galerkin methods for nonlinear elasticity. Results Appl. Math. **1**, 100001 (2019). https://doi.org/10.1016/j.rinam.2019.01.001
71. Fernandez, P., Christophe, A., Terrana, S., Nguyen, N.C., Peraire, J.: Hybridized discontinuous Galerkin methods for wave propagation. J. Sci. Comput. **77**(3), 1566–1604 (2018). https://doi.org/10.1007/s10915-018-0811-x
72. Kabaria, H., Lew, A., Cockburn, B.: A hybridizable discontinuous Galerkin formulation for non-linear elasticity. Comput. Methods Appl. Mech. Eng. **283**, 303–329 (2015)
73. Terrana, S., Nguyen, N.C., Bonet, J., Peraire, J.: A hybridizable discontinuous Galerkin method for both thin and 3D nonlinear elastic structures. Comput. Methods Appl. Mech. Eng. **352**, 561–585 (2019). https://doi.org/10.1016/J.CMA.2019.04.029
74. Bai, Y., Fidkowski, K.J.: Continuous artificial-viscosity shock capturing for hybrid discontinuous Galerkin on adapted meshes. AIAA J. **60**(10), 5678–5691 (2022). https://doi.org/10.2514/1.J061783
75. Barter, G.E., Darmofal, D.L.: Shock capturing with PDE-based artificial viscosity for DGFEM: Part I. Formulation. J. Comput. Phys. **229**(5), 1810–1827 (2010). https://doi.org/10.1016/j.jcp.2009.11.010
76. Ching, E.J., Lv, Y., Gnoffo, P., Barnhardt, M., Ihme, M.: Shock capturing for discontinuous Galerkin methods with application to predicting heat transfer in hypersonic flows. J. Comput. Phys. **376**, 54–75 (2019). https://doi.org/10.1016/j.jcp.2018.09.016
77. Nguyen, N. C., Perairey, J.: An adaptive shock-capturing HDG method for compressible flows. In: 20th AIAA computational fluid dynamics conference 2011, American Institute of Aeronautics and Astronautics, Reston, Virigina, 2011, pp. AIAA 2011–3060. https://doi.org/10.2514/6.2011-3060. http://arc.aiaa.org/doi/abs/10.2514/6.2011-3060
78. Moro, D., Nguyen, N.C., Peraire, J.: Dilation-based shock capturing for high-order methods. Int. J. Numer. Meth. Fluids **82**(7), 398–416 (2016). https://doi.org/10.1002/fld.4223
79. Persson, P. O., Peraire, J.: Sub-cell shock capturing for discontinuous Galerkin methods. In: Collection of Technical Papers-44th AIAA Aerospace Sciences Meeting, Vol. 2, pp. 1408–1420. Reno, Neveda (2006). https://doi.org/10.2514/6.2006-112

80. Persson, P. O.: Shock capturing for high-order discontinuous Galerkin simulation of transient flow problems. In: 21st AIAA computational fluid dynamics conference, p. 3061. San Diego, CA (2013). https://doi.org/10.2514/6.2013-3061