From Godunov to a unified hybridized discontinuous Galerkin framework for partial differential equations

Tan Bui-Thanh
Department of Aerospace Engineering and Engineering Mechanics, Institute for Computational Engineering & Sciences, The University of Texas at Austin, Austin, TX 78712, USA

ABSTRACT

By revisiting the basic Godunov approach for system of linear hyperbolic Partial Differential Equations (PDEs) we show that it is hybridizable. As such, it is a natural recipe for us to constructively and systematically establish a unified hybridized discontinuous Galerkin (HDG) framework for a large class of PDEs including those of Friedrichs’ type. The unification is fourfold. First, it provides a single constructive procedure to devise HDG schemes for elliptic, parabolic, hyperbolic, and mixed-type PDEs. The key that we exploit is the fact that, for many PDEs, irrespective of their type, the first order form is a hyperbolic system. Second, it reveals the nature of the trace unknowns as the upwind states. Third, it provides a parameter-free HDG framework, and hence eliminating the “usual complaint” that HDG is a parameter-dependent method. Fourth, it allows us to rediscover most existing HDG methods and furthermore discover new ones.

We apply the proposed unified framework to three different PDEs: the convection-diffusion-reaction equation, the Maxwell equation in frequency domain, and the Stokes equation. The purpose is to present a step-by-step construction of various HDG methods, including the most economic ones with least trace unknowns, by exploiting the particular structure of the underlying PDEs. The well-posedness of the resulting HDG schemes, i.e. the existence and uniqueness of the HDG solutions, is proved. The well-posedness result is also extended and proved for abstract Friedrichs’ systems. We also discuss variants of the proposed unified framework and extend them to the popular Lax–Friedrichs flux and to nonlinear PDEs. Numerical results for transport equation, convection-diffusion equation, compressible Euler equation, and shallow water equation are presented to support the unification framework.

© 2015 Published by Elsevier Inc.

1. Introduction

The discontinuous Galerkin (DG) method was originally developed by Reed and Hill [1] for the neutron transport equation, first analyzed in [2,3], and then has been extended to other problems governed by partial differential equations (PDEs) [4]. Roughly speaking, DG combines advantages of classical finite volume and finite element methods. In particular, it has the ability to treat solutions with large gradients including shocks, it provides the flexibility to deal with complex geometries, and it is highly parallelizable due to its compact stencil. As such, it has been adopted to solve large-scale forward [5,6] and inverse [7] problems. However, for steady state problems or time-dependent ones that require implicit
time-integrators, DG methods typically have many more (coupled) unknowns compared to the other existing numerical methods, and hence more expensive in general.

Recently, Cockburn, his coworkers, and others have introduced hybridizable (also known as hybridized) discontinuous Galerkin (HDG) methods for various types of PDE including Poisson equation [8], convection–diffusion equation [9,10], Stokes equations [11–18], Euler and Navier–Stokes equations [19–21], Maxwell equation [22–24], acoustics and elastodynamics [25], Helmholtz equation [26,27], and eigenvalue problem [28], to name a few. The beauty of the HDG method is that it reduces the number of coupled unknowns substantially while retaining all other attractive properties of the DG method. The coupled unknowns are in fact new unknown traces introduced on the mesh skeleton, i.e. the faces, to hybridize the DG method. Once they are solved for, the usual DG unknowns can be recovered in an element-by-element fashion, completely independent of each other. Existing HDG constructions however vary from one type of PDE to another, though they do share some similarities. Moreover, they are parameter-dependent method. Consequently, practitioners may be wary of deriving/applying the HDG approach to a new PDE.

Our aim is to construct a unified HDG framework that can be applied to a wider class of PDEs and that rediscovers most, if not all, the existing HDG methods when applied to aforementioned PDEs. More importantly, the unified framework should be able to provide a constructive approach to discover new HDG methods. It should be pointed out that similar efforts for constructing a unified DG framework have been done recently in [29–32]. We have also successfully devised a unified framework for the discontinuous Petrov–Galerkin method [33]. Following this light, this paper is an attempt to provide a unified framework for the HDG approach.

In the following we summarize the contributions of this article. We start by revisiting the Godunov approach for solving a general system of linear PDEs in Section 2. This reveals the fact that the Godunov (upwind) flux on a face can be hybridized. That is, it can be computed using information from either of elements sharing that face and the single-valued upwind state. This motivates us to replace the upwind state by a trace unknown living on the mesh skeleton and solve for it as part of the discrete solution. At the first sight, this seems to be more expensive than the standard (upwind) DG. However, by a Schur complement approach (“static condensation”) we show that the actual coupled unknowns are the traces, and hence the number of globally coupled unknowns are substantially reduced. In other words, instead of evaluating the upwind state first and then the Godunov flux as usually done, we propose to treat the upwind state as unknown and compute it. We essentially suggest to redesign or, equivalently, to hybridize the Godunov flux (in this paper, we use Godunov flux and upwind flux interchangeably).

The purpose of our systematic construction of the upwind HDG framework is threefold. First, we will show that the basic Godunov approach can be used as a natural recipe to constructively derive HDG fluxes. Thus, Godunov flux is hybridizable. Second, we provide a new insight of the trace unknown, that is, it is nothing more than the upwind state. Third, the upwind HDG framework is parameter-free by construction. Nevertheless, from the upwind HDG framework we will show how one can naturally and constructively derive most of the existing parameter-dependent HDG methods.

We then apply our abstract upwind HDG framework to convection–diffusion–reaction equation in Section 3, Maxwell’s equations in frequency domain in Section 4, and Stokes equation in Section 5. The purpose of these sections is to show that by exploiting the structure of the underlying PDEs, one can systematically construct HDG methods with minimal number of trace unknowns. In particular, we shall demonstrate the power of the unified framework in rediscovers most, if not all, of the existing HDG methods and in discovering new ones. In each section (or its subsections) the well-posedness of the HDG global and local systems will be analyzed.

Section 6 is our attempt to generalize the analysis in Sections 3, 4, and 5 to a large class of PDEs of Friedrichs’ type. Next, Section 7 shows how to hybridize the popular Lax–Friedrichs flux, whose advantages and disadvantages compared to the upwind HDG flux are discussed in detail. After that, Section 8 constructs variants of the upwind HDG method, among which is a popular class of parameter-dependent HDG schemes. Section 9 extends the HDG framework to PDEs with discontinuous coefficients, and Section 10 presents an extrapolation to nonlinear PDEs. To validate our developments we presents numerical results in Section 11 for various types of PDE including transport equation, convection–diffusion equation, compressible Euler equation, and shallow water equation. Finally, Section 12 concludes the paper.

2. A constructive derivation of a unified HDG framework

In this section we are interested in constructing a single HDG framework for general system of linear PDEs of the following form

$$
\sum_{k=1}^{d} \partial_k F_k (u) + Cu := \sum_{k=1}^{d} \partial_k (A_k u) + Cu = f, \quad \text{in } \Omega,
$$

(1)

where $d$ is the spatial dimension (which, for the clarity of the exposition, is assumed to be $d = 3$ whenever a particular value of dimension is of concern), $F_k$ the $k$th component of the flux vector (or tensor) $F$, $u$ the unknown solution with

---

[1] This is the paper that invented the HDG method.
values in $\mathbb{R}^m$, and $f$ the forcing term. The matrices $A_k$ and $C$ are assumed to be constant and continuous\(^2\) across $\Omega$. Here, $\partial_k$ is understood as the $k$th partial derivative. Since the boundary condition plays no role in the basic construction and understanding of our unified framework, it is ignored until the well-definedness is concerned. At that point we will provide the boundary condition, which is then incorporated via the trace unknown.

We are going to solve \((1)\) using the discontinuous Galerkin finite element method. To begin, let us partition the domain $\Omega$ into $N_{el}$ non-overlapping elements $K_j$, $j = 1, \ldots, N_{el}$ with Lipschitz boundaries such that $\Omega_h := \bigcup_{j=1}^{N_{el}} K_j$ and $\partial \Omega := \bigcup_{j=1}^{N_{el}} \partial K_j$. Here, $h$ is defined as $h := \max_{1 \leq j \leq N_{el}} \text{diam} (K_j)$. We denote the skeleton of the mesh by $\mathcal{E}_h := \bigcup_{j=1}^{N_{el}} \partial K_j$; it is the set of all (uniquely defined) faces $e$, each of which comes with a normal vector $n$. We conventionally identify $n^\ast$ as the normal vector on the boundary $\partial K$ of element $K$ (also denoted as $K^\ast$) and $n^\ast = -n$ as the normal vector of the boundary of a neighboring element (also denoted as $K^\ast$). Furthermore, we use $\mathcal{n}$ to denote either $n^\ast$ or $n$ in an expression that is valid for both cases, and this convention is also used for other quantities (restricted) on a face $e \in \mathcal{E}_h$. For the sake of convenience, we denote by $\mathcal{E}_h^0$ the sets of all boundary faces on $\partial \Omega$, by $\mathcal{E}_h^\ast := \mathcal{E}_h \setminus \mathcal{E}_h^0$ the set of all interior faces, and by $\partial \Omega_h := \{K : K \in \Omega_h\}$.

For simplicity in writing we define $(\cdot, \cdot)_K$ as the $L^2$-inner product on a domain $K \in \mathbb{R}^d$ and $(\cdot, \cdot)_K$ as the $L^2$-inner product on a domain $K$ if $K \in \mathbb{R}^{d-1}$. We shall use $\| \cdot \|_K := \| \cdot \|_{L^2(K)}$ as the induced norm for both cases and the particular value of $K$ in a context will indicate which inner product the norm is coming from. We also denote the $e$-weighted norm of a function $u$ as $\| u \|_{K, e} := \| \sqrt{e} u \|_K$ for any positive $\sqrt{e}$. We shall use boldface lowercase letters for vector-valued functions and in that case the inner product is defined as $(u, v)_K := \sum_{i=1}^{m} (u_i, v_i)_K$, and similarly $(u, v)_K := \sum_{i=1}^{m} (u_i, v_i)_K$, where $m$ is the number of components of $u$. Moreover, we define $(u, v)_n := \sum_{e \in \mathcal{E}_h} (u, v)_e$ and $(u, v)_{e_h} := \sum_{e \in \mathcal{E}_h} (u, v)_e$ whose induced (weighted) norms are clear, and hence their definitions are omitted. We also employ boldface uppercase letter, e.g. $L$, to denote both matrices and tensors. It is our convention that subscripts are used to denote the components of vector, matrix, and tensor.

We define $\mathcal{T}^p (K)$ as the space of polynomials of degree at most $p$ on a domain $K$. Next, we introduce three discontinuous piecewise polynomial spaces

$$
\mathcal{V}_h (\Omega_h) := \left\{ v \in \left[ L^2 (\Omega_h) \right]^m : v|_K \in \left[ \mathcal{T}^p (K) \right]^m, \forall K \in \Omega_h \right\},
$$

$$
\mathcal{A}_h (\mathcal{E}_h) := \left\{ \lambda \in \left[ L^2 (\mathcal{E}_h) \right]^m : \lambda|_e \in \left[ \mathcal{T}^p (\mathcal{e}) \right]^m, \forall e \in \mathcal{E}_h \right\},
$$

and similarly for $\mathcal{V}_h (K)$, and $\mathcal{A}_h (e)$ by replacing $\Omega_h$ with $K$ and $\mathcal{E}_h$ with $e$. If $m = 1$, i.e. scalar-valued functions, we denote these spaces as

$$
\mathcal{V}_h (\Omega_h) := \left\{ v \in \left[ L^2 (\Omega_h) \right] : v|_K \in \mathcal{T}^p (K), \forall K \in \Omega_h \right\},
$$

$$
\mathcal{A}_h (\mathcal{E}_h) := \left\{ \lambda \in \left[ L^2 (\mathcal{E}_h) \right] : \lambda|_e \in \mathcal{T}^p (\mathcal{e}), \forall e \in \mathcal{E}_h \right\}.
$$

From now on we conventionally use $u$ for the numerical solution and $u^\ast$ for the exact solution. We would like to find a local finite element solution $u \in \mathcal{V}_h (K)$ on each element $K \in \Omega_h$. To that end, multiplying \((1)\) by $v$ and integrating by parts we have

$$
- (F (u), \nabla v)_K + (F (u) \cdot n, v)_{\partial K} + (C u, v)_K = (f, v)_K, \quad \forall v \in \mathcal{V}_h (K). \quad (2)
$$

At this point, the flux $F (u)$ is not well-defined since the traces of both $u^\ast$ of element $K^\ast$ and $u^\ast$ of element $K^\ast$ co-exist on $e$. Godunov-type methods \cite{godunov} resolve this by first solving, either exactly or approximately, for the upwind state at the interface $e$ and then introducing the upwind numerical flux $F^\ast (u^\ast, u^\ast)$ so that \((2)\) becomes

$$
- (F (u), \nabla v)_K + (F^\ast (u^\ast, u^\ast) \cdot n, v)_{\partial K} + (C u, v)_K = (f, v)_K, \quad \forall v \in \mathcal{V}_h (K). \quad (3)
$$

It is this numerical flux that couples local unknowns on elements $K^\ast$ and $K^\ast$ that share a face $e \in \partial K$. Consequently, the local unknowns on all elements are coupled, and they must be solved together. This leads to the “usual complaint” that DG has many more coupled unknowns, and hence more expensive in general. The chief goal of this paper is to break this coupling and this can be accomplished by hybridizing the classical Godunov flux, as we now show.

The upwind numerical flux is typically constructed by solving the Riemann problem \cite{riemann} locally along the normal $n$ of $\partial K$: find $u (n, t)$ such that

$$
\frac{\partial u}{\partial t} + \frac{\partial (A u)}{\partial n} = 0, \quad (4)
$$

with the initial condition $u (n, 0) = u^\ast$ for $n < 0$, $u (n, 0) = u^\ast$ for $n \geq 0$, and $A = \sum_{k=1}^{d} A_k n_k$. Here, (artificial) time $t$ is introduced to help understand the Godunov flux via the Riemann problem, but it is otherwise not necessary in the

\(^2\) The assumption will be relaxed later.
following development. Though the detailed solution of Riemann problem can be found, for example, in [35], we review the main points here for completeness. We begin with the eigenvalue decomposition of $A$

$$A := \sum A_k n_k = RSR^{-1},$$

(5)

where $S := \text{diag}(\theta_1, \ldots, \theta_m)$ and $\theta_i$ are eigenvalues of $A$. To the end of this section we assume that $A$ has real eigenvalues and independent eigenvectors, i.e., the system of linear PDEs (1) is hyperbolic [35].

Since $R$ forms a complete basis of $\mathbb{R}^m$, we can decompose the initial condition as

$$u^- = \sum_{i=1}^l \alpha_i^- R_i, \quad \text{and} \quad u^+ = \sum_{i=1}^m \alpha_i^+ R_i.$$ 

The solution of the Riemann problem (4) for small time $t$ can be shown to be

$$u^* = \sum_{i=1}^l \alpha_i^- R_i + \sum_{i=l+1}^m \alpha_i^+ R_i,$$

(6)

where $l$ is such that $\theta_l < 0$ and $\theta_{l+1} \geq 0$. In terms of the “left” (−) and the “right” (+) states, we have

$$u^* = u^- + \sum_{i=1}^l (\alpha_i^+ - \alpha_i^-) R_i,$$

(7)

and

$$u^* = u^+ - \sum_{i=l+1}^m (\alpha_i^+ - \alpha_i^-) R_i.$$ 

(8)

The Godunov, or equivalently upwind, flux $F^* \cdot n^-$ on the boundary of element $K^-$ is found by multiplying both sides of (7) by $A$, i.e.,

$$F^* \cdot n^- := Au^- + A \sum_{i=1}^l (\alpha_i^+ - \alpha_i^-) R_i.$$ 

(9)

Since the positive spectrum of $A$ has no contribution to the second term in (9), we can write

$$F^* \cdot n^- = Au^- - |A| \sum_{i=1}^l (\alpha_i^+ - \alpha_i^-) R_i,$$ 

(10)

where $|A| := R |S| R^{-1}$. Now substituting (7) into (10) gives

$$F^* \cdot n^- = F (u^-) \cdot n^- + |A| (u^- - u^*).$$ 

(11)

Similarly, the Godunov flux for the neighboring element $K^+$ on the same face $e$ is given by

$$F^* \cdot n^+ := F (u^+) \cdot n^+ + |A| (u^+ - u^*).$$ 

(12)

The following identity obviously holds

$$\ll F^* \cdot n \rr = 0,$$ 

(13)

where we have defined the “jump” operator $\ll (\cdot) \rr := (\cdot)^- + (\cdot)^+.$

Next, combining (11) and (12) we have the usual “symmetric” form of the Godunov flux

$$F^* \cdot n^- = \frac{1}{2} \ll F (u^-) + F (u^+) \rr \cdot n^- + \frac{1}{2} |A| (u^- - u^*),$$ 

(14)

which is usually used in upwind numerical methods such as upwind DG. It is this nicely symmetric form that couples the unknowns of the neighboring elements, and hence the unknowns of all elements.

Can we avoid this “standard” global coupling? The above re-derivation of the Godunov flux with two new expressions in (11) and (12) in fact provides an answer. We first observe that the upwind fluxes (11) and (12) depend on the DG unknowns of only one side of a face $e \in \partial K$ and the single-valued solution $u^*$ of the Riemann problem. The numerical flux is then completely determined using only information from either side (left or right) of the face $e \in \partial K$, as long as $u^*$ is (either exactly or approximately) provided. More importantly, this in turn shows that we can solve (3) for $u$ element-by-element independent of each other.
The above discussion suggests that we should treat $u^*$ as the extra unknown and solve for it on the skeleton of the mesh instead of using the Riemann/upwind state which couples the local unknown $u$. To signify this step, let us rename $u^*$ to $\hat{u}$ and $F^*$ to $\hat{F}$, i.e.,
\[
\hat{F} \cdot n = F(u) \cdot n + |A|(u - \hat{u}),
\]
and we refer to this as the hybridized *upwind flux* or HDG flux. Conventionally, “hatted” quantities are computed using the trace unknown $\hat{u}$ while “starred” quantities are associated with the upwind state $u^*$. An immediate question that arises is how to compute $\hat{u}$. To answer this question, we note that $\hat{u}$ is a new unknown that is introduced on $\partial K$ so that (3) can be solved in an element-by-element fashion. To ensure the well-posedness of our formulation, we need to introduce an extra equation on $\partial K$. Clearly, at this point $\hat{u}$ is not the upwind state and hence the identity (13) is in general no longer satisfied for $\hat{F}$. It is therefore natural to use (13) as the extra equation. This additional algebraic equation ensures that what is coming out from element $K$ through its boundary $\partial K$ must enter the neighboring elements that share (part of) the boundary $\partial K$. This is the statement of conservation and it is exactly conveyed by (13).

In summary, for each element $K$, the DG local unknown $u$ and the extra "trace" unknown $\hat{u}$ need to satisfy
\[
- (F(u), \nabla v)_K + \left[ \hat{F}(u, \hat{u}) \cdot n, v \right]_{\partial K} + (Cu, v)_K = (f, v)_K, \quad \forall v \in V_h(K),
\]
\[
\left[ \left[ \hat{F}(u, \hat{u}) \cdot n \right], \mu \right]_{\partial K} = 0, \quad \forall \mu \in \Lambda_h,
\]
where we have enforced the uniqueness of the HDG flux weakly which is sufficient for local conservation. We have constructively devised an upwind HDG formulation which consists of (16a), (16b), and the HDG flux (15).

By revisiting the Godunov method we have constructively derived the HDG method for a general system of linear hyperbolic PDEs. The procedure for computing HDG solution requires three steps. We first solve (16a) for the local solution $u$ as a function of $\hat{u}$. It is then substituted into the conservative algebraic equation (16b) on the mesh skeleton to solve for the unknown $\hat{u}$. Finally, the local unknown $u$ is computed, as in the first step, using $\hat{u}$ from the second step. To signify the nature of each step, we conventionally name the element-by-element elimination in the first and the third step as the *local solver*, and the process for computing the trace on the mesh skeleton in the second step as the *global solver*.

Let us emphasize some other crucial points. First, we have shown that the Godunov method is hybridizable. Second, the trace unknown in our upwind HDG, once computed, is nothing more than the upwind state; we will verify this fact for various linear PDEs in the following. Third, the appearance of $|A|$, and hence $|A|(u - \hat{u})$, is to ensure the upwinding mechanism, and hence stability. In the HDG literature $|A|$ is a free stabilization parameter/matrix that must be appropriately chosen for stability, whereas in our upwind HDG framework $|A|$ arises naturally from the hybridization of the Godunov method. Thus, it automatically ensures the stability of the upwind HDG method if the original Godunov method is stable.

At this level of generality, it appears that we have $m$ trace unknowns $\hat{u}_i, i = 1, \ldots, m$ to solve for on the mesh skeleton. However, when we apply our upwind HDG methodology to a particular PDE—including convection–diffusion–reaction equation in Section 3, Maxwell equation in Section 4, and Stokes equation in Section 5—we can eliminate most of them so that there are in fact only a few trace unknowns that need to be solved for. For all examples considered, we will assume that all the boundary conditions are homogeneous for the clarity of the exposition. The extension to non-homogeneous cases is straightforward and hence omitted.

3. HDG schemes for convection–diffusion–reaction equation

In this section we will apply the upwind HDG methodology in Section 2 to convection–diffusion–reaction equation and its subsets. As shall be shown, we recover several existing HDG methods and discover new ones.

The problem of interest in this section is the convection–diffusion–reaction equation written in the first order form
\[
\begin{align}
\nabla \cdot \sigma + \nabla u &= 0 \quad \text{in } \Omega, \\
\n\nabla \cdot \sigma + \beta \cdot \nabla u + \nu u &= f \quad \text{in } \Omega,
\end{align}
\]
where we assume $\beta \in L^\infty(\Omega)^d$, $\nabla \cdot \beta \in L^\infty(\Omega)$. Simple algebraic manipulation shows that
\[
|A| = \frac{1}{\alpha} \begin{bmatrix}
2 n_1^2 & 2 n_1 n_2 & 2 n_1 n_3 & \beta \cdot n_1 \\
2 n_1 n_2 & 2 n_2^2 & 2 n_2 n_3 & \beta \cdot n_2 \\
2 n_1 n_3 & 2 n_2 n_3 & 2 n_3^2 & \beta \cdot n_3 \\
\beta \cdot n_1 & \beta \cdot n_2 & \beta \cdot n_3 & \gamma
\end{bmatrix},
\]
with $\alpha := \sqrt{\beta \cdot n_1^2 + 4}$ and $\gamma := |\beta \cdot n|^2 + 2$. It is important to point out that though the original PDE is of mixed hyperbolic–elliptic type, the system (17) is hyperbolic ($A$ is symmetric, and hence having real eigenvalues and independent eigenvectors) in the group variable $\mathbf{u} = (u, \sigma)$. As such, it falls under our unified treatment. In particular, the Godunov flux (15) becomes
where the upwind state in (6) is defined as \( u^* = (u^*, \sigma^* = (\sigma^*_1, \sigma^*_2, \sigma^*_3) = (u^*, \sigma^*) \).

If we stop here and construct an HDG method by replacing starred quantities by hatted ones as in the abstract setting presented in Section 2, we see that there are four trace unknowns: three from \( \sigma^* \) and one from \( u^* \). However, as can be seen from (18), there are only two independent quantities, namely, the traces \( [\sigma^* \cdot n] \) and \( u^* \) that are needed to compute the upwind flux. In fact, even these two quantities are not independent of each other, as will be shown, and hence either of them can be eliminated. This observation is the key leading to the most economic HDG methods: first removing all the dependencies, and then constructing HDG methods.

3.1. Constructing HDG scheme by eliminating \( \sigma^* \cdot n \)

We first show that the Godunov flux can be written as a function of only \( u^* \).

**Lemma 3.1.** The following hold true:

i) The upwind state \( u^* = (u^*, \sigma^*) \) satisfies

\[
\begin{align*}
    u^* &= ([u] + \frac{\beta \cdot n}{2\alpha} [u n] \cdot n + \frac{1}{\alpha} [\sigma \cdot n]), \\
    \sigma^* \cdot n &= ([\sigma]) \cdot n + \frac{1}{\alpha} [u n] \cdot n - \frac{\beta \cdot n}{2\alpha} [\sigma \cdot n].
\end{align*}
\]

(19a) (19b)

ii) The Godunov flux is given by

\[
F^* \cdot n = \begin{bmatrix}
    u^* n_1 \\
    u^* n_2 \\
    u^* n_3 \\
\end{bmatrix},
\]

\[
\begin{bmatrix}
    \beta \cdot v + \sigma \cdot n + \frac{1}{2} (\alpha - \beta \cdot n) (u - u^*)
\end{bmatrix}
\]

(20)

where

\[
    u^* = u + \frac{2}{\alpha} (\sigma - \sigma^*) \cdot n + \frac{\beta \cdot n}{\alpha} (u - u^*).
\]

(21)

**Proof.** To prove the first assertion we can use popular techniques such as those in [35]. Here, we take a different approach using the conservation equation (similar to that of Proposition 3.4 of [8]). Specifically, we note that for the convection–diffusion–reaction problem, the conservation (13) gives us four equations for the upwind state \( u^* \). Solving for \( u^* \) and \( \sigma^* \cdot n \) in terms of \( u \) and \( \sigma \) we obtain the desired result. The second assertion immediately follows by substituting (19) into (18) and inspecting that (21) is true. \( \square \)

Up to this point, we have used the exact Riemann solution and the Godunov flux to derive identities in Lemma 3.1. These simple results pave the way for constructing an HDG method with single trace unknown. In particular, we start by replacing the starred quantities in (20) with hatted ones, i.e.,

\[
\hat{F} \cdot n = \begin{bmatrix}
    \hat{n} n_1 \\
    \hat{n} n_2 \\
    \hat{n} n_3 \\
\end{bmatrix},
\]

\[
\begin{bmatrix}
    \beta \cdot v + \sigma \cdot n + \frac{1}{2} (\alpha - \beta \cdot n) (u - \hat{u})
\end{bmatrix}
\]

(22)

and then declaring that \( \hat{u} \) is unknown, that is, \( \hat{u} \) is not assumed to satisfy (19) and (21). Clearly, the first three components of the HDG flux (22) automatically satisfy the conservation condition (16b), and thus we need to weakly enforce only the fourth one, i.e.,

\[
\left[ [\beta \cdot v + \sigma \cdot n + \frac{1}{2} (\alpha - \beta \cdot n) (u - \hat{u})] \right]_e = 0, \quad \forall \mu \in \Lambda_h (e), \quad \forall v \in \mathcal{E}_h^0.
\]

(23)

Taking \( v = (\kappa, v) \in V_h (K) \otimes V_h (K) \) as the test function, the local equation (16a), when applied to (17), becomes: find \( (\sigma, u) \in V_h (K) \otimes V_h (K) \) such that
\[
\begin{align*}
\left( e^{-1} \sigma_i, k_i \right)_K &= -(u, \partial_t k_i)_K + \left( \left[ \mathbf{F} \cdot \mathbf{n} \right]_{i} \right)_K, \quad i = 1, 2, 3, \\
&\quad - (\sigma, \nabla v)_K - (u, \nabla \cdot (\beta v))_K + (\nu u, v)_K + \left( \left[ \mathbf{F} \cdot \mathbf{n} \right]_{4} \right)_K = (f, v)_K, 
\end{align*}
\]

(24a) 

(24b)

with \( \left[ \mathbf{F} \cdot \mathbf{n} \right]_i \) denoting the \( i \)th component of numerical flux (22).

In summary, we have proposed an HDG scheme consisting of local equations (24), HDG flux (22), and conservation condition (23). A question immediately arises is whether the trace solution \( \hat{u} \) is identical to the upwind state (19a) and whether the HDG flux (22) is in fact the Godunov one. The answer is, not surprisingly, affirmative.

**Lemma 3.2.** The trace solution \( \hat{u} \) of the HDG scheme (24), (22), and (23) coincides with the upwind state. Furthermore, the HDG flux (22) is identical to the Godunov upwind flux (18).

**Proof.** We proceed with the conservation condition (23) which shows that

\[
\langle \alpha \hat{u}, \mu \rangle_e = \left\langle \alpha \langle u \rangle + \frac{\beta}{2} \mathbf{[un]} \cdot \mathbf{n} + \left[ \sigma \cdot \mathbf{n} \right], \mu \right\rangle_e = \left\langle \alpha u^*, \mu \right\rangle_e, \quad \forall \mu \in \Lambda_h(e),
\]

where we have used (19a) to obtain the second equality. Now, taking \( \mu = \hat{u} - u^* \) yields

\[
\|\hat{u} - u^*\|_{\varepsilon, \alpha}^2 = 0,
\]

which in turn shows that \( \hat{u} \) is exactly the same as \( u^* \) since \( \alpha \) is constant and \( u^* \in \Lambda_h(\varepsilon_h) \) by our assumption. The second assertion is clear by defining \( \mathbf{\hat{\sigma}} \cdot \mathbf{n} \) in place of \( \mathbf{\sigma}^* \cdot \mathbf{n} \) in (19b) and substituting \( \hat{u} \) from (19a) into (22).

3.2. Constructing HDG scheme by eliminating \( u^* \)

One way to achieve the goal of eliminating \( u^* \) is to define the unique flux \( \theta^* \)

\[
\theta^* \text{sgn}(\mathbf{n}) := \beta \cdot v + \sigma \cdot \mathbf{n} + \frac{\beta}{\alpha} (\sigma - \sigma^*) \cdot \mathbf{n} + \frac{\gamma}{\alpha} (u - u^*),
\]

(25)

so that the fourth component of the Godunov flux (18) can be written simply as

\[
\left( \mathbf{F}^* \cdot \mathbf{n} \right)_4 = \theta^* \text{sgn}(\mathbf{n}).
\]

Here, we have defined, \( \forall e \in E_h \),

\[
\text{sgn}(\mathbf{n}) := \begin{cases}
1 & \text{if } \mathbf{n} = \mathbf{n}^e \\
-1 & \text{if } \mathbf{n} = -\mathbf{n}^e
\end{cases}
\]

Then, we have a result parallel to Lemma 3.1.

**Proposition 3.3.** The Godunov flux (18) is equivalent to

\[
\mathbf{F}^* \cdot \mathbf{n} = \begin{bmatrix}
\hat{u} \mathbf{n}_1 \\
\hat{u} \mathbf{n}_2 \\
\hat{u} \mathbf{n}_3 \\
\theta^* \text{sgn}(\mathbf{n})
\end{bmatrix},
\]

(26)

where

\[
\hat{u} = \left( 1 + \frac{2\alpha \beta \cdot \mathbf{n}}{\alpha - \beta \cdot \mathbf{n}} \right) u + \frac{2\alpha}{\alpha - \beta \cdot \mathbf{n}} \sigma \cdot \mathbf{n} - \frac{2\alpha}{\alpha - \beta \cdot \mathbf{n}} \theta^* \text{sgn}(\mathbf{n}).
\]

(27)

**Proof.** Combining (21) and (25) gives

\[
(\sigma - \sigma^*) \cdot \mathbf{n} = -\frac{\alpha}{2} \left( 1 + \frac{\beta \cdot \mathbf{n}}{2} \right) (u - u^*),
\]

(28a)

\[
(u - u^*) = \frac{2}{\alpha - \beta \cdot \mathbf{n}} \left[ \theta^* \text{sgn}(\mathbf{n}) - \beta \cdot \mathbf{n} - \sigma \cdot \mathbf{n} \right].
\]

(28b)

Substituting (28) and (25) into (18) ends the proof.

**Proposition 3.3** shows that the upwind flux (18) can be written as a function of only \( \theta^* \) instead of \( u^* \) and \( \sigma^* \). This suggests that we can construct a new HDG scheme consisting of local solver (24), and with \( \mathbf{F}^* \cdot \mathbf{n} \) and \( \theta^* \) in (26) replaced
by \( \hat{\mathbf{F}} \cdot \mathbf{n} \) and \( \hat{\theta} \). Here, \( \hat{\theta} \) is now the single trace unknown. As can be seen, the fourth equation in the conservation condition (16b) is automatically satisfied since \( \hat{\theta} \) is single-valued. We need to weakly enforce the conservation for the first three components, but they are the same up to a constant \( \mathbf{n} \). Consequently, only one conservation condition on the skeleton is necessary and this ensures that the number of equations is the same as the number of unknowns, i.e.,

\[
\langle [\mathbf{u} \, \text{sgn}(\mathbf{n})], \mu \rangle_e = 0, \quad \forall \mu \in \Lambda_h(e), \quad \forall e \in \mathcal{E}_h^o.
\]  

Similar to Lemma 3.2, one can show that the new HDG scheme with \( \hat{\theta} \) as the trace unknown is equivalent to the standard upwind DG scheme.

3.3. HDG method for convection–reaction equation

This is a special case of (17) in which the diffusion coefficient is zero, i.e. \( \epsilon = 0 \). The construction of the HDG method following the general procedure in Section 2 is substantially simplified. In particular, we have

\[
|A| = |\beta \cdot \mathbf{n}|,
\]

and the HDG flux (14) turns out to be

\[
\hat{\mathbf{F}} \cdot \mathbf{n} = \beta \cdot \mathbf{v} + |\beta \cdot \mathbf{n}| (u - \hat{u}).
\]  

The local HDG equation (16a) now becomes: find \( u \in V_h(K) \) such that

\[
-(\mathbf{u}, \nabla (\beta \mathbf{v}))_K + (\mathbf{v}, \mathbf{u})_K + \langle \hat{\mathbf{F}} \cdot \mathbf{n}, \mathbf{v} \rangle_{\partial K} = (f, \mathbf{v})_K, \quad \forall \mathbf{v} \in V_h(K)
\]  

with the HDG flux given in (30), and with the conservation equation given as

\[
\langle [\beta \cdot \mathbf{v} + |\beta \cdot \mathbf{n}| (u - \hat{u})], \mu \rangle_e = 0, \quad \forall \mu \in \Lambda_h(e), \quad \forall e \in \mathcal{E}_h^o.
\]  

Lemma 3.4. The trace solution \( \hat{u} \) of the HDG scheme (31), (30), and (32) coincides with the upwind state. Furthermore, the HDG flux \( \hat{\mathbf{F}} \cdot \mathbf{n} \) (30) is identical to the standard Godunov upwind flux for convection–reaction problem.

Proof. To begin, we note that, similar to Lemma 3.1, we can use the conservation equation to deduce that

\[
\hat{u} = \frac{1}{2} \lVert \mathbf{u} \, \text{sgn}(\beta \cdot \mathbf{n}) \rVert + \lVert \mathbf{u} \rVert,
\]

which is exactly the upwind state [1,35]. Next, \( \hat{u} \) can be substituted into the HDG flux (30) to give

\[
\hat{\mathbf{F}} \cdot \mathbf{n} = \beta \cdot \mathbf{n} \lVert \mathbf{u} \rVert + \frac{1}{2} \text{sgn}(\beta \cdot \mathbf{n}) \lVert \beta \cdot \mathbf{v} \rVert,
\]

which is exactly the usual upwind flux [1,35] for the convection equation. Clearly, we can also bypass the computation of \( \hat{u} \) to compute the upwind flux directly by

\[
2\hat{\mathbf{F}} \cdot \mathbf{n}^- = \hat{\mathbf{F}} \cdot \mathbf{n}^- - \hat{\mathbf{F}} \cdot \mathbf{n}^+ = 2\beta \cdot \mathbf{n} \lVert \mathbf{u} \rVert + \text{sgn}(\beta \cdot \mathbf{n}) \lVert \beta \cdot \mathbf{v} \rVert,
\]

and hence obtaining the same result. □

The next result shows that our upwind HDG method is the same as the standard upwind scheme [3,1,2]. This is not surprising since the former is constructed from the latter by hybridizing the upwind flux.

Corollary 3.5. The HDG scheme (31), (30), and (32) is equivalent to the standard upwind DG scheme.

Proof. We first show that the HDG solution \( u \) is a solution of the DG scheme. But this is clear from the proof of Lemma 3.4. In particular, by substituting (34) into (31), the latter becomes the standard upwind DG [3,1,2].

Conversely, let \( u \) be the solution of the upwind DG method, i.e., \( u \) satisfies (31) with the upwind flux (34). Now, first defining \( \hat{u} \) as in (33), then expressing the upwind flux (34) as a function of \( \hat{u} \) we recover (31). We also observe that the upwind flux satisfies the conservation equation (32), and this concludes the proof. □

3.4. HDG method for Poisson equation

This is a special case of the HDG scheme in Section 3.1 in which the velocity field \( \beta \) is zero, i.e. \( \beta = 0 \). We again emphasize that though the original Poisson equation is of elliptic type, its first order form with group variable \( \mathbf{u} = (u, \sigma) \) is hyperbolic, and hence our unified theory is applicable. In particular, the HDG flux (22) now simplifies to
\[ \mathbf{\hat{F}} \cdot \mathbf{n} = \begin{bmatrix} \hat{u}n_1 \\ \hat{u}n_2 \\ \hat{u}n_3 \\ \mathbf{\sigma} \cdot \mathbf{n} + (u - \hat{u}) \end{bmatrix}, \]

As shown above, \( \hat{u} \) is identical to the upwind state and hence approximating the trace of the exact solution on the mesh skeleton. This implies that the discrepancy/error \( u - \hat{u} \) converges to zero as the mesh (or solution order) is refined for PDE with smooth solution. This observation suggests that one can control this discrepancy by penalizing \( u - \hat{u} \) using a penalty parameter \( \tau \), i.e.,

\[ \mathbf{\hat{F}} \cdot \mathbf{n} = \begin{bmatrix} \hat{u}n_1 \\ \hat{u}n_2 \\ \hat{u}n_3 \\ \mathbf{\sigma} \cdot \mathbf{n} + \tau (u - \hat{u}) \end{bmatrix}, \quad (35) \]

which recovers our upwind HDG when \( \tau = 1 \). On the other hand, if \( \tau \) approaches infinity the trace unknown \( \hat{u} \) is forced to match the trace of the DG solution \( u \). One can realize that, by hybridizing the Godunov flux, we have constructively and intuitively recovered the HDG methods for elliptic equations proposed in [36].

**Remark 3.6.** It is important to point out that since the differential part of the Helmholtz equation is the same as that of the Poisson equation, the upwind HDG flux, or its penalized variant, for the Helmholtz equation is identical to (35). That is, we have also recovered the HDG scheme for Helmholtz equation in [26,27].

### 3.5. Analysis

In this section, we discuss the relation of the our proposed HDG schemes to the existing ones. Since the upwind HDG scheme with (24), (26), and (29) is equivalent to that with (24), (22), and (23), we study only the latter. We will also discuss the HDG schemes for transport and Poisson equations.

It is important to point out that our upwind HDG scheme for convection–diffusion–reaction problem (17) is different from that proposed in [11]. In particular, their HDG flux \( \mathbf{\hat{\sigma}}_{\text{NPC}} \cdot \mathbf{n} \) is

\[ \mathbf{\hat{\sigma}}_{\text{NPC}} \cdot \mathbf{n} := \mathbf{\beta} \cdot \mathbf{\hat{u}} + \mathbf{\sigma} \cdot \mathbf{n} + \tau (u - \hat{u}), \]

where \( \tau \) is the so-called stabilization parameter. On the other hand, by hybridizing the Godunov upwind method, ours is

\[ \mathbf{\hat{\sigma}} \cdot \mathbf{n} := \mathbf{\beta} \cdot \mathbf{v} + \mathbf{\sigma} \cdot \mathbf{n} + \frac{1}{2} (\mathbf{\alpha} - \mathbf{\beta} \cdot \mathbf{n}) (u - \hat{u}). \]

Nevertheless, we can rewrite our upwind HDG flux as

\[ \mathbf{\hat{\sigma}} \cdot \mathbf{n} := \mathbf{\beta} \cdot \mathbf{\hat{u}} + \mathbf{\sigma} \cdot \mathbf{n} + \frac{1}{2} (\mathbf{\alpha} + \mathbf{\beta} \cdot \mathbf{n}) (u - \hat{u}), \]

and if we take \( \tau = \frac{1}{2} (\mathbf{\alpha} + \mathbf{\beta} \cdot \mathbf{n}) \) the two fluxes are identical. Moreover, in this case \( \tau = \beta \cdot \mathbf{n}/2 = \alpha > 0 \), and therefore all results in [11] hold for our upwind HDG method as well. Let us summarize some important points for mixed Dirichlet–Neumann boundary condition (enforcing boundary condition via the trace unknown is straightforward and can be found in [11]).

**Theorem 3.7.** Assume \( \nu - \frac{1}{2} \mathbf{\nabla} \cdot \mathbf{\beta} \geq 0 \). Then, there exists a unique solution \((u, \sigma, \hat{u})\) for the upwind HDG scheme (24), (22), and (23).

**Proof.** See [11] for a proof. \( \square \)

When \( \mathbf{\beta} = \mathbf{0} \) the HDG scheme is much simplified as shown in Section 3.4. Consequently, more results can be obtained for this case. Indeed, the work in [9] presents a complete analysis, including optimal convergent rate, of the HDG method for the Poisson equation with Dirichlet boundary condition. Thus our upwind HDG method also converges optimally. A question that immediately arises is whether it is consistent with the local discontinuous Galerkin (LDG) approach [37] in which the upwinding is done in the opposite direction, i.e. upwinding for (24a) and downwinding for (24b) or the other way around. For clarity in the exposition, we present the answer for the Poisson equation.

Due to Lemma 3.2, we see that since \( \hat{u} = u^* \), and hence \( \mathbf{\hat{\sigma}} = \mathbf{\sigma}^\ast \), they also satisfy (21), i.e.,

\[ \hat{u} - u = - (\mathbf{\hat{\sigma}} \cdot \mathbf{n} - \mathbf{\sigma} \cdot \mathbf{n}), \quad (36) \]

which says that the unbalanced (or mismatched) components of the flux due to our upwind HDG discretization are opposite. More specifically, \( \hat{u} - u \) and \( \mathbf{\hat{\sigma}} \cdot \mathbf{n} - \mathbf{\sigma} \cdot \mathbf{n} \) are the flux mismatch of quantities \( u \) and \( \sigma \), respectively, per unit area/length.
Identity (36) implies that the mismatches are opposite. In particular, if \( \hat{u} = u^\tau \) then from (19a)-(19b) one can see that \( \hat{\sigma} \cdot n = n^\tau \cdot n \), i.e., they are upwind fluxes in opposite directions. Thus, we have rediscovered the LDG upwinding mechanism.

We next discuss the upwind HDG scheme for transport equation in Section 3.3. From Corollary 3.5, the upwind HDG scheme is equivalent to the standard upwind DG method, and hence all the results in [3] hold for the former as well. Let us present some results that are special for our HDG scheme. These are simpler versions of the results in Section 6, and hence we postpone the boundary condition enforcement until then.

**Lemma 3.8.** If \( v - \frac{1}{2} \nabla \cdot \beta > 0 \), there exists a unique solution for the local solver (31).

**Proof.** Since the local problem is in finite dimensional space, it is sufficient to show that \( u = 0 \) when \( f = 0 \) and \( \hat{u} = 0 \). Indeed, testing (31) with \( v = u \) we have

\[
-(u, \nabla \cdot (\beta u))_K + (\nu u, u)_K + ((|\beta \cdot n| + \beta \cdot n) u, u)_{\partial K} = 0,
\]

which, after integrating the first term by parts gives

\[
(u, \beta \cdot \nabla u)_K + (\nu u, u)_K + ((|\beta \cdot n| + \beta \cdot n) u, u)_{\partial K} = 0.
\]

Finally adding the above two equations gives

\[
(2v - \nabla \cdot \beta) u, u)_K + ((2|\beta \cdot n| + \beta \cdot n) u, u)_{\partial K} = 0.
\]

Since \( v = \frac{1}{2} \nabla \cdot \beta > 0 \) and \( 2|\beta \cdot n| + \beta \cdot n \geq 0 \), we conclude that \( u = 0 \). \( \square \)

**Lemma 3.9.** If \( v - \frac{1}{2} \nabla \cdot \beta > 0 \), there exists a unique solution \( (u, \hat{u}) \) for the upwind HDG scheme.

**Proof.** Similar to the proof of Lemma 3.8, it is sufficient to show that \( f = 0 \) and \( g = 0 \) imply \( u = 0 \) and \( \hat{u} = 0 \). Indeed, eliminating \( \hat{u} \) as in the proof of Lemma 3.4 we recover the standard upwind DG scheme [3] which shows that \( u = 0 \) since \( f = 0 \) and \( g = 0 \). This in turn implies that \( \hat{u} = 0 \). \( \square \)

We have used the equivalence (Corollary 3.5) between our upwind HDG method and the standard upwind HDG to obtain the uniqueness of \( \hat{u} \) with no effort. We can also obtain the convergence rate of our upwind HDG method.

**Theorem 3.10.** Suppose the exact solution \( u^e \) is sufficiently smooth, i.e. \( u^e|_K \in H^{p+1}(K) \), then the HDG solution \( (u, \hat{u}) \) satisfies the following error bounds

\[
\| u - u^e \|_{L^2(\Omega_h)} \leq ch^{p+\frac{1}{2}} \| u^e \|_{H^{k+1}(\Omega_h)},
\]

\[
\| \hat{u} - u^e \|_{L^2(\Omega_h^e)} \leq ch^{p+\frac{1}{2}} \| u^e \|_{H^{k+1}(\Omega_h)},
\]

**Proof.** Since the upwind HDG is equivalent to the upwind DG scheme, the first assertion is immediate by the existing convergence theory of the latter (see, e.g., [3,38]). For the second assertion, subtracting \( u^e \) from both sides of (33) yields

\[
\| \hat{u} - u^e \|_{L^2(\epsilon)} \leq \frac{1}{2} \| u - u^e \|_{L^2(\epsilon)} \text{sgn}(\beta \cdot n) + \| (u - u^e) \|_{L^2(\epsilon)}
\]

\[
\leq c \| u - u^e \|_{L^2(\epsilon)} \leq ch^{p+\frac{1}{2}} \| u^e \|_{H^{k+1}(K)}.
\]

where the last inequality is from the upwind DG (see [3,38] and the references therein). Summing over all faces \( \epsilon \), and hence elements, ends the proof. \( \square \)

**4. HDG schemes for the Maxwell equations**

In this section, we apply the single abstract HDG framework in Section 2 to the Maxwell equation in frequency\(^3\) (time harmonic) domain. We will show that by hybridizing the upwind DG for the first order Maxwell equation we can derive families of HDG schemes from which we recover many existing methods and discover new ones.

Similar to [22] we consider the following time-harmonic equations

\[
vE - \nabla \times H = 0 \text{ in } \Omega,
\]

\[
\nabla \times E - \varepsilon \omega^2 H + \varepsilon \nabla q = f \text{ in } \Omega,
\]

\[
\nabla \cdot (\varepsilon H) = 0 \text{ in } \Omega.
\]

\(^3\) Construction of upwind HDG schemes for the time domain Maxwell equation is simpler and hence omitted here.
In this case, \( A \) reads
\[
A = \begin{bmatrix}
0 & B & 0 \\
B & 0 & D \\
0 & D^T & 0
\end{bmatrix},
\]
where \( B = \begin{bmatrix}
0 & n_3 & -n_2 \\
-n_3 & 0 & n_1 \\
n_2 & -n_1 & 0
\end{bmatrix}, \quad D = \varepsilon \begin{bmatrix}
n_1 \\
n_2 \\
n_3
\end{bmatrix},
\]
and \( 0 \)'s are zero matrices with appropriate size. After some simple algebraic manipulations one can show that \( |A| \) is given by
\[
|A| = \begin{bmatrix}
\tilde{B} & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & \varepsilon
\end{bmatrix},
\]
where \( \tilde{B} = \begin{bmatrix}
(1 - n_1^2) & -n_1 n_2 & -n_1 n_3 \\
-n_1 n_2 & (1 - n_2^2) & -n_2 n_3 \\
-n_1 n_3 & -n_2 n_3 & (1 - n_3^2)
\end{bmatrix},
\]
and
\[
\tilde{B} = \begin{bmatrix}
(\varepsilon - 1) n_1^2 + 1 & (\varepsilon - 1) n_1 n_2 & (\varepsilon - 1) n_1 n_3 \\
(\varepsilon - 1) n_1 n_2 & (\varepsilon - 1) n_2^2 + 1 & (\varepsilon - 1) n_2 n_3 \\
(\varepsilon - 1) n_1 n_3 & (\varepsilon - 1) n_2 n_3 & (\varepsilon - 1) n_3^2 + 1
\end{bmatrix}.
\]
Consequently, the Godunov fluxes (11)-(12) in this case read
\[
F^+ \cdot n = \begin{bmatrix}
-n \times H + n \times n \times (E^+ - E) \\
\varepsilon H \cdot n + \varepsilon (q^* - q)
\end{bmatrix},
\]
where \( E^+, H^+ \), and \( q^* \) (two vectors and one scalar) are the upwind states.

Similar to previous sections, we seek the relationships between the upwind states to motivate the construction of HDG methods. Proposition 4.1 collects some important identities.

**Proposition 4.1.** The following hold for the upwind states \( E^+, H^+ \) and \( q^* \):
\[
\begin{align*}
n \times n \times (E^+ - E) &= n \times (H - H^+) , \\
(H - H^+) \cdot n &= (q^* - q) .
\end{align*}
\]
and
\[
\begin{align*}
n \times n \times E^+ &= n \times n \times (|E|) + \frac{1}{2} [n \times H] , \\
n \times n \times H^+ &= -(|n \times E|) + n \times n \times (|H|) , \\
\hat{H} \cdot n &= (|q n + H|) \cdot n , \\
q^* &= |q| + \frac{1}{2} [H \cdot n] .
\end{align*}
\]
Using the results in Proposition 4.1, the upwind flux (38) can be written either as
\[
F^+ \cdot n = \begin{bmatrix}
-n \times H + n \times n \times (E^+ - E) \\
\varepsilon n \times E^+ + \varepsilon q^* n \\
\varepsilon H \cdot n + \varepsilon (q^* - q)
\end{bmatrix},
\]
(39)
or
\[
F^+ \cdot n = \begin{bmatrix}
n \times E + n \times n \times (H^+ - H) + \varepsilon q^* n \\
\varepsilon H \cdot n + \varepsilon (q^* - q)
\end{bmatrix},
\]
(40)
or
\[
F^+ \cdot n = \begin{bmatrix}
n \times E + \varepsilon q n + n \times n \times (H^+ - H) + \varepsilon [(H - H^+) \cdot n] n \\
\varepsilon H^+ \cdot n
\end{bmatrix}.
\]
(41)

Let us also define the tangent component of the electric and magnetic fields as \( E^t = -n \times n \times E \) and \( H^t = -n \times n \times H \). It can be observed from (39)-(41) that we in fact need only either the pair \((E^t, q^*)\) or the pair \((H^t, q^*)\) or \(H^+\) to compute the upwind flux. This motivates us to define the following two\(^4\) (families of) HDG schemes based on two different, but equivalent, Godunov fluxes (39)-(40).

\(^4\) Our analysis shows that the HDG local solver using the HDG flux (41) does not have a unique solution for \( q \) for the interior elements. A remedy similar to Section 5 can be done but we skip its discussion in this paper to keep the length of the paper reasonable.
4.1. HDG scheme based on \( \left( \bar{E}, \hat{q} \right) \)

The starting point of the first HDG scheme is to define the HDG flux based on (39), i.e.,

\[
\hat{F} \cdot \mathbf{n} = \begin{bmatrix}
-\mathbf{n} \times \mathbf{H} + (\mathbf{E} - \bar{E}) \\
\mathbf{n} \times \mathbf{E} + \varepsilon \hat{q} \mathbf{n}
\end{bmatrix}
\]

(42)

Thus, the actual single-valued trace unknowns in our first HDG scheme are the tangent vector \( \mathbf{E} \) and the scalar \( \hat{q} \). Now, since \( \mathbf{n} \times \mathbf{E} + \varepsilon \hat{q} \mathbf{n} \) automatically fulfills the conservation property and \( -\mathbf{n} \times \mathbf{H} + (\mathbf{E} - \bar{E}) \) lives in the tangent plane, (16b) reduces to

\[
\left( -\mathbf{n} \times \mathbf{H} + (\mathbf{E} - \bar{E}) \right) = \mathbf{0} \\
\varepsilon \left( \mathbf{n} \times \mathbf{H} + (\mathbf{E} - \bar{E}) \right) = \mathbf{0}
\]

(43)

where the tangent trace space \( \mathbf{A}_h \) is defined as

\[
\mathbf{A}_h : \lambda \in \left( \mathcal{L}^2 (\mathcal{E}_h) \right)^m : \lambda_e \in \left( \mathcal{P}^p (e) \right)^m, \lambda \cdot \mathbf{n}|_e = \mathbf{0}, \forall e \in \mathcal{E}_h
\]

(44)

Similar to Section 3.4, one can control the mismatches \( \mathbf{E} \) and \( \hat{q} \) by introducing two penalty parameters \( \tau^e \) and \( \tau^n \) to form a penalized family of HDG fluxes as

\[
\hat{F} \cdot \mathbf{n} = \begin{bmatrix}
-\mathbf{n} \times \mathbf{H} + \tau^e (\mathbf{E} - \bar{E}) \\
\mathbf{n} \times \mathbf{E} + \varepsilon \hat{q} \mathbf{n}
\end{bmatrix}
\]

(45)

The conservation algebraic constraints in this case read

\[
\left( -\mathbf{n} \times \mathbf{H} + \tau^e (\mathbf{E} - \bar{E}) \right) = \mathbf{0} \\
\varepsilon \left( \mathbf{n} \times \mathbf{H} + \tau^e (\mathbf{E} - \bar{E}) \right) = \mathbf{0}
\]

(46)

Clearly, we recover the upwind HDG scheme by setting \( \tau^e = \tau^n = 1 \).

Next is the description of the local solver. Taking \( \mathbf{v} = (\mathbf{e}, \mathbf{h}, v) \in \mathbf{V}_h (K) \otimes \mathbf{V}_h (K) \otimes \mathbf{V}_h (K) \) as the test function, the local equation (16a) when applied to (37) becomes: find \( (\mathbf{E}, \mathbf{H}, q) \in \mathbf{V}_h (K) \otimes \mathbf{V}_h (K) \otimes \mathbf{V}_h (K) \), such that

\[
(\nu \mathbf{E}, \mathbf{e})_K + (\mathbf{H}, \nabla \times \mathbf{e})_K + \sum_{i=1}^{3} \left( \left( \hat{F} \cdot \mathbf{n} \right)_i, \mathbf{e} \right)_K = 0
\]

(47)

with \( \left( \hat{F} \cdot \mathbf{n} \right)_i, i = 1, \ldots, 7 \) denoting the ith component of numerical flux (44). Thus, the first HDG scheme consists of the local solver (46), the global solver (45), and the HDG flux (44).

4.2. HDG scheme based on \( \left( \bar{H}, \hat{q} \right) \)

In the second HDG method, we define the HDG flux based on (40), i.e.,

\[
\hat{F} \cdot \mathbf{n} = \begin{bmatrix}
-\mathbf{n} \times \mathbf{H} \\
\mathbf{n} \times \mathbf{E} + \varepsilon \hat{q} \mathbf{n}
\end{bmatrix}
\]

(48)

As can be seen, this HDG flux (47) is the same as that in [22] with unity stabilization parameters. On the one hand, unlike [22], our upwind HDG flux is constructed by hybridizing the Godunov flux, and hence (stabilization) parameter-free. On the other hand, one can view our approach as a constructive way to devise the family of parameter-dependent HDG
fluxes in [22]. In particular, introducing two parameters \( \tau^f \) and \( \tau^n \) to penalize the deviation of \( \hat{H}^f \) and \( \hat{q} \) from \( H^f \) and \( q \), respectively, gives

\[
\tilde{F} \cdot n = \begin{bmatrix}
-\mathbf{n} \times \hat{H}^f \\
\mathbf{n} \times \mathbf{E} + \tau^f (H^f - \hat{H}^f) + \epsilon \hat{q} \mathbf{n} \\
\epsilon \mathbf{H} \cdot \mathbf{n} + \tau^n \epsilon (q - \hat{q})
\end{bmatrix}.
\] (48)

Clearly, when \( \tau^f = \tau^n = 1 \) the upwind HDG flux (47) is recovered.

It is obvious that the first three components of the HDG flux (48) automatically satisfy the conservation constraint (16b). For the next three components, note that \( \mathbf{n} \times \mathbf{E} + \tau^f (H^f - \hat{H}^f) \) is a vector in the tangent plane while \( \epsilon \hat{q} \mathbf{n} \) is a vector along the normal vector. The latter automatically satisfies the conservation condition. Thus, (16b) when applied to the HDG flux (48) reduces to the following algebraic constraints

\[
\begin{align*}
\left( \left\| \mathbf{n} \times \mathbf{E} + \tau^f (H^f - \hat{H}^f) \right\| , \mu^t \right)_e & = 0, \quad \forall \mu^t \in \Lambda^t_h(e), \forall e \in \mathcal{E}_h^n, \\
\left( \left\| \epsilon \mathbf{H} \cdot \mathbf{n} + \tau^n \epsilon (q - \hat{q}) \right\| , \mu^s \right)_e & = 0, \quad \forall e \in \mathcal{E}_h^n, \quad \forall \mu^s \in \Lambda_s_h.
\end{align*}
\] (49a, 49b)

The second HDG scheme consists of the local solver (46), the global solver (49), and the HDG flux (48).

**Remark 4.2.** Given the understanding on the nature of HDG schemes established in this paper, one can introduce a new HDG scheme based on the HDG flux (40) by defining

\[
\hat{q} \mathbf{sgn}(\mathbf{n}) := \hat{q} \mathbf{n} + \tau^n \epsilon (q - \hat{q}),
\]

which transforms the HDG flux (40) into

\[
\tilde{F} \cdot n = \begin{bmatrix}
-\mathbf{n} \times \hat{H}^f \\
\mathbf{n} \times \mathbf{E} + \tau^f (H^f - \hat{H}^f) + \epsilon (\mathbf{H} \cdot \mathbf{n}) \mathbf{n} + \tau^n \epsilon (q - \hat{q} \mathbf{sgn}(\mathbf{n})) \mathbf{n} \\
\epsilon \hat{q} \mathbf{sgn}(\mathbf{n})
\end{bmatrix}.
\] (50)

For this HDG flux, the first three and the seventh conservation conditions are automatically satisfied. We are left with three algebraic constraints, i.e. \( \forall \mu \in \Lambda^t_h(e), e \in \mathcal{E}_h^n, \)

\[
\left( \left\| \mathbf{n} \times \mathbf{E} + \tau^f (H^f - \hat{H}^f) + \epsilon (\mathbf{H} \cdot \mathbf{n}) \mathbf{n} + \tau^n \epsilon (q - \hat{q} \mathbf{sgn}(\mathbf{n})) \mathbf{n} \right\| , \mu \right)_e = 0.
\] (51)

By construction, the HDG fluxes (40) and (50) are equivalent. The number of algebraic constraints in (49) is also the same as that in (51) since the tangent trace space \( \Lambda^t_h \) has two components while it is three for \( \Lambda_s_h \).

Parallel to Lemma 3.4 and Corollary 3.5, the following results hold the proposed HDG families.

**Lemma 4.3.** Assume \( \epsilon, \nu \) are positive, and \( \tau^f = \tau^n = 1 \).

i) The tangent component of the trace solutions \( \hat{H}^f \) and \( \hat{E} \) of the two proposed HDG schemes coincides with that of the upwind states in Proposition 4.1. Furthermore, the HDG fluxes are identical to the Godunov upwind flux (38).

ii) The two proposed HDG schemes are equivalent to the standard upwind DG scheme.

**Proof.** The proofs of these assertions are similar to those of Lemma 3.4 and Corollary 3.5, so we leave out the details. \( \square \)

4.3. Analysis

For simplicity in the exposition we will analyze our HDG scheme for the fully upwind case, i.e. \( \tau^f = \tau^n = 1 \). For the general case with \( \tau^f, \tau^n \geq 0 \), all the proofs will go through except that some terms will carry \( \tau^f \) and \( \tau^n \). Since our second HDG scheme is the same as that proposed in [22], its analysis is ignored and the readers are referred to [22]. Thus, the analysis of our first HDG scheme, a new HDG scheme for time harmonic Maxwell equation, is the main focus of this section. By construction, it is automatically locally conservative and consistent since it is equivalent to the upwind DG method (Lemma 4.3).

In order to study the well-posedness of the local solver (solving for \( \mathbf{E}, \mathbf{H}, q \) as a function of \( \hat{E} \) and \( \hat{q} \)) and global solver (solving for \( \hat{E} \) and \( \hat{q} \) on the mesh skeleton), we need to supply boundary conditions. For simplicity, perfect electric conduction (PEC) boundary condition is considered, i.e.,

\[
\mathbf{n} \times \mathbf{E} = \mathbf{0}, \quad \text{and} \quad q = 0 \text{ on } \partial \Omega.
\]
which is enforced through the hybrid variables, i.e.,
$$ \mathbf{n} \times \mathbf{E}^i = \mathbf{0}, \text{ and } \mathbf{q} = \mathbf{0} \text{ on } \partial \Omega. $$

The local solver (46) with the HDG flux (39) equipped with the boundary conditions reads

$$ (\nu \mathbf{E}, \mathbf{e})_K + (\mathbf{n} \times \mathbf{E} + (\mathbf{E}^i - 1)_{\partial \Omega} \mathbf{E}^i, \mathbf{e})_{\partial \Omega} = 0, \quad \text{(52a)} $$

$$ - (\varepsilon \omega^2 \mathbf{H}, \mathbf{h})_K + (\mathbf{E}, \nabla \times \mathbf{E})_K - (\mathbf{q}, \nabla \cdot (\mathbf{E}^i \mathbf{h}))_K + \left(1_{\partial \Omega} \mathbf{n} \times \mathbf{E}^i, \mathbf{h}\right)_{\partial \Omega} + \left(1_{\partial \Omega} \mathbf{E}^i \mathbf{q}, \mathbf{h}\right)_{\partial \Omega} = (\mathbf{f}, \mathbf{h})_K, \quad \text{(52b)} $$

$$ - (\varepsilon \mathbf{H}, \nabla \mathbf{v})_K + (\varepsilon \mathbf{H} \cdot \mathbf{n} + \varepsilon (\mathbf{q} - 1_{\partial \Omega} \mathbf{q}^i), \mathbf{v})_{\partial \Omega} = 0. \quad \text{(52c)} $$

Similar to [22] we also assume that $\varepsilon \omega^2$ is not an eigenvalue for an eigenvalue problem defined by local solver (52) or by our HDG scheme with $\mathbf{f} = \mathbf{0}$. This assumption is implicitly understood from now on. We are now in the position to study the well-posedness of the proposed upwind HDG method.

**Theorem 4.4.** Suppose $\varepsilon, \nu > 0$. There is a unique solution $(\mathbf{E}^i, \mathbf{q})$ for the local solver and a unique solution $(\mathbf{E}, \mathbf{H}, \mathbf{q})$ for the global solver.

**Proof.** First, we exploit the finite dimensionality and linearity of the square system determining $\mathbf{E}^i$ and $\mathbf{q}$. Proving the well-posedness of global solver is equivalent to show that $\mathbf{E}^i = \mathbf{0}$ and $\mathbf{q} = \mathbf{0}$ on $\mathcal{E}^0_h$ are the only solution if $\mathbf{f} = \mathbf{0}$. We proceed by taking $\mathbf{e} = \mathbf{E}, \mathbf{h} = \mathbf{H}$ and $\nu = \mathbf{q}$ in the local solver (52). Next, integrating by parts the second term in (52a) and the first term in (52c), and adding the resulting equations, all together, we obtain

$$ \|\mathbf{E}\|_{\Omega_h, \nu}^2 + \sum_K \left( \left(\mathbf{E}^i - 1_{\partial \Omega} \mathbf{E}^i\right) \cdot \mathbf{E}^i \right)_{\partial \Omega} + \left(1_{\partial \Omega} \mathbf{n} \times \mathbf{E}^i, \mathbf{H}\right)_{\partial \Omega} $$

$$ + \sum_K \left(1_{\partial \Omega} \mathbf{E}^i \mathbf{q}, \mathbf{H}\right)_{\partial \Omega} = \|\mathbf{H}\|_{\Omega_h, \varepsilon \omega^2}. \quad \text{(53)} $$

On the other hand, taking $\mathbf{u}^i = \mathbf{E}^i$ and $\lambda = \mathbf{q}$ in (43) and then adding the resulting equations lead to

$$ \sum_K \left(1_{\partial \Omega} \mathbf{n} \times \mathbf{H} \cdot \mathbf{E}^i \right)_{\partial \Omega} + \left(1_{\partial \Omega} \mathbf{E}^i \mathbf{E}^i \right)_{\partial \Omega} $$

$$ + \sum_K \left(1_{\partial \Omega} \mathbf{E}^i \mathbf{q}, \mathbf{H}\right)_{\partial \Omega} + \left(1_{\partial \Omega} \mathbf{E}^i \mathbf{q}, \mathbf{H}\right)_{\partial \Omega} = 0. \quad \text{(54)} $$

Next subtracting (54) from (53) yields

$$ \|\mathbf{E}\|_{\Omega_h, \nu}^2 + \sum_K \left(1_{\partial \Omega} \mathbf{E}^i \mathbf{E}^i \right)_{\partial \Omega} + \left(1_{\partial \Omega} \mathbf{E}^i \mathbf{E}^i \right)_{\partial \Omega} + \left(1_{\partial \Omega} \mathbf{E}^i \mathbf{E}^i \right)_{\partial \Omega} $$

$$ + \|\mathbf{E}\|_{\Omega_h, \varepsilon \omega^2}^2 = \|\mathbf{H}\|_{\Omega_h, \varepsilon \omega^2}. \quad \text{(55)} $$

Since $\varepsilon \omega^2$ is not an eigenvalue of the eigenvalue problem defined by the HDG scheme with $\mathbf{f} = \mathbf{0}$, $\mathbf{H}$ must vanish. As a result, from (55), we infer that $\mathbf{E} = \mathbf{0}$ on $\Omega_h$, $\mathbf{E} = \mathbf{E}^i = \mathbf{0}$ on $\mathcal{E}^0_h$, $\mathbf{q} = \mathbf{q}$ on $\mathcal{E}^0_h$, and $\mathbf{q} = \mathbf{0}$ on $\partial \Omega$. It follows that $\mathbf{q}$ is continuous and has zero trace on $\partial \Omega$.

Now taking $\nu = \mathbf{q}$ in (52c), using the fact that $\mathbf{H} = \mathbf{0}$, $\mathbf{q} = \mathbf{0}$ on $\mathcal{E}^0_h$, and $\mathbf{q} = \mathbf{0}$ on $\partial \Omega$, and then summing the resulting equations all together we have

$$ (\varepsilon \mathbf{H}, \nabla \mathbf{q})_{\Omega_h} = 0, \quad \forall \mathbf{H} \in \mathbf{V}_h(\Omega_h). $$

which implies that $\mathbf{q}$ is piecewise constant. However, $\mathbf{q}$ is continuous and vanishes on $\partial \Omega$. This can only be true if $\mathbf{q}$ is identically zero. Consequently, $\mathbf{q} = \mathbf{0}$ on $\mathcal{E}^0_h$ as well. \( \square \)

### 4.4. HDG schemes for vector wave equation

We have explicitly imposed the divergent free condition for the magnetic field in (37c), and hence it is necessary to introduce the Lagrange multiplier $\mathbf{q}$ in (37b) [22]. Though this formulation is more stable, especially when $\omega$ is small, we have one extra PDE with one extra unknown. If this is not desirable one can study the vector wave equation directly, i.e.,

$$ \nu \mathbf{E} - \nabla \times \mathbf{H} = \mathbf{0} \text{ in } \Omega, \quad \text{(56a)} $$

$$ \nabla \times \mathbf{E} - \varepsilon \omega^2 \mathbf{H} = \mathbf{f} \text{ in } \Omega, \quad \text{(56b)} $$

which is a simplified version of (37). Indeed, one can show that the Godunov flux in this case reads
\[
\mathbf{F}^* \cdot \mathbf{n} = \left[ -\mathbf{n} \times \mathbf{H} + \mathbf{n} \times \mathbf{n} \times (\mathbf{E}^* - \mathbf{E}) \right] \\
+ \left[ \mathbf{n} \times \mathbf{E} + \mathbf{n} \times \mathbf{n} \times (\mathbf{H}^* - \mathbf{H}) \right],
\]

which is, as expected, a simplified version of (38) by removing the last component and all the terms associated with the normal direction. This observation immediately gives us two HDG fluxes without further effort, i.e.,

\[
\hat{\mathbf{F}} \cdot \mathbf{n} = \left[ -\mathbf{n} \times \mathbf{H} + \tau^t \left( \mathbf{E}^* - \hat{\mathbf{E}} \right) \right] \hat{\mathbf{n}} \times \hat{\mathbf{E}},
\]

and

\[
\hat{\mathbf{F}} \cdot \mathbf{n} = \left[ -\mathbf{n} \times \hat{\mathbf{H}} + \mathbf{n} \times \mathbf{E} + \tau^t \left( \mathbf{H}^* - \hat{\mathbf{H}} \right) \right] \hat{\mathbf{n}} \times \hat{\mathbf{E}}.
\]

We can construct two HDG schemes corresponding to these two HDG fluxes, the former of which is a new HDG scheme for vector wave equation (56) while the latter coincides with that proposed in [22]. Clearly, the analysis for time harmonic Maxwell equation in previous sections still holds by removing appropriate terms, and for that reason we leave out the unnecessary details here.

5. HDG schemes for Stokes equation

Let us consider the Stokes equation in the following first order velocity-pressure-gradient form (see, e.g, [17,15,18]):

\[
\begin{align*}
\nu^{-1} \mathbf{L} - \nabla \mathbf{u} &= 0, \quad \text{in } \Omega \\
-\nabla \cdot \mathbf{L} + \nabla q &= \mathbf{f}, \quad \text{in } \Omega \\
\nabla \cdot \mathbf{u} &= 0, \quad \text{in } \Omega, \\
\int q \, d\Omega &= 0.
\end{align*}
\]

Though we leave out the detailed derivations here, one can show that, as a system of first order linear PDEs, (58) is of hyperbolic type since the corresponding matrix \( \mathbf{A} \) has real eigenvalues and independent eigenvectors. Indeed, the upwind Godunov flux in this case reads

\[
\hat{\mathbf{F}} \cdot \mathbf{n} = \left[ -\mathbf{n} \otimes \mathbf{u} + \Sigma \\
-\mathbf{L} \cdot \mathbf{n} + q \mathbf{n} + \gamma \\
\mathbf{u} \cdot \mathbf{n} + \alpha \right]
\]

where we have defined

\[
\begin{align*}
\Sigma &:= \mathbf{n} \otimes \left[ (\mathbf{L} - \mathbf{L}^*) \cdot \mathbf{n} \right] + \frac{(\sqrt{2} - 2)}{2} \left[ \mathbf{n} \cdot (\mathbf{L} - \mathbf{L}^*) \cdot \mathbf{n} \right] \mathbf{n} \otimes \mathbf{n} - \frac{\sqrt{2}}{2} \mathbf{n} \otimes \mathbf{n} (q - q^*), \\
\gamma &:= \frac{\sqrt{2}}{2} \left[ (\mathbf{u} - \mathbf{u}^*) \cdot \mathbf{n} \right] \mathbf{n} \times \mathbf{n} \times (\mathbf{u} - \mathbf{u}^*), \\
\alpha &:= -\frac{\sqrt{2}}{2} \mathbf{n} \cdot (\mathbf{L} - \mathbf{L}^*) \cdot \mathbf{n} + \frac{\sqrt{2}}{2} (q - q^*),
\end{align*}
\]

with \((\mathbf{u}^*, \mathbf{L}^*, q^*)\) as the upwind state. By inspection, the following results are straightforward.

**Proposition 5.1.** There hold:

\[
\begin{align*}
\Sigma &= \mathbf{n} \otimes \left( \gamma + \frac{(\sqrt{2} - 2)}{2} \mathbf{n} \otimes \mathbf{n} \cdot \gamma \right) = \mathbf{n} \otimes (\mathbf{u} - \mathbf{u}^*), \\
\alpha &= -\frac{\sqrt{2}}{2} \gamma \cdot \mathbf{n} = - (\mathbf{u} - \mathbf{u}^*) \cdot \mathbf{n}, \\
\Sigma \cdot \mathbf{n} &= \frac{\sqrt{2}}{2} \mathbf{n} \otimes \mathbf{n} \cdot \gamma = -\alpha \mathbf{n},
\end{align*}
\]

and the Godunov flux can be written as

\[
\mathbf{F}^* \cdot \mathbf{n} = \left[ -\mathbf{L} \cdot \mathbf{n} + q \mathbf{n} + \sqrt{2} (\mathbf{u} - \mathbf{u}^*) \cdot \mathbf{n} \mathbf{n} - \mathbf{n} \times \mathbf{u} - \mathbf{u}^* \cdot \mathbf{n} \right].
\]
As for other PDEs discussed in this paper, we start the construction of HDG method by using (61) to define the HDG flux, i.e.,
\[
\hat{F} \cdot \mathbf{n} = \begin{bmatrix} -\mathbf{n} \otimes \hat{u} \\ -L \cdot \mathbf{n} + q \mathbf{n} + \sqrt{2} \left( (\mathbf{u} - \hat{u}) \cdot \mathbf{n} \right) \mathbf{n} - \mathbf{n} \times \left( \mathbf{u} - \hat{u} \right) \end{bmatrix},
\] (62)
with \( \hat{u} \) as the trace unknown that needs to be solved for.

Unlike the other PDEs for which we have started defining first the conservation condition and then the local solver, for Stokes case it is useful to discuss the local solver first as we will see. The reason is that additional condition has to be introduced to make the HDG method for Stokes equation well-defined, and this is best motivated through the local solver.

To that end, taking \( \mathbf{v} = (\mathbf{G}, \mathbf{v}, r) \) as the test function, the local equation (16a) when applied to (58) becomes: find \((\mathbf{L}, \mathbf{u}, q)\) such that
\[
\begin{align*}
(v \mathbf{L}, \mathbf{G})_K + (\mathbf{u}, \nabla \cdot \mathbf{G})_K - (1_{\partial K \setminus \partial \Omega} \mathbf{n} \otimes \hat{u}, \mathbf{G})_{\partial K} &= 0, \\
(L, \nabla \mathbf{v})_K - (q, \nabla \cdot \mathbf{v})_K + (-L \cdot \mathbf{n} + q \mathbf{n}, \mathbf{v})_{\partial K} \\
+ (\sqrt{2} \left( (\mathbf{u} - 1_{\partial K \setminus \partial \Omega} \hat{u}) \cdot \mathbf{n} \right) \mathbf{n} - \mathbf{n} \times (\mathbf{u} - 1_{\partial K \setminus \partial \Omega} \hat{u}) \cdot \mathbf{n} - \mathbf{n} \times (\mathbf{u} - \hat{u}), \mathbf{v})_{\partial K} &= (f, \mathbf{v})_K, \\
- (\mathbf{u}, \nabla r) + (1_{\partial K \setminus \partial \Omega} \hat{u} \cdot \mathbf{n}, r)_{\partial K} &= 0,
\end{align*}
\] (63a–b)
where we have imposed the homogeneous Dirichlet boundary condition, \( \mathbf{u} = 0 \) on \( \partial \Omega \), through the trace unknown \( \hat{u} \). Next, we use the energy method to study the well-posedness of the local solver. Since it is linear and finite dimensional, it is sufficient to show that \( \mathbf{L} = 0, \mathbf{u} = 0 \) and \( q = 0 \) are the only solution if \( f = 0 \) and \( \hat{u} = 0 \). To begin, taking \( \mathbf{G} = \mathbf{L}, \mathbf{v} = \mathbf{u} \) and \( r = q \) in (63), integrating by parts the second term of (63a) and (63b), and then adding the resulting equations of (63) all together yield
\[
\| \mathbf{L} \|_{Ka}^2 + \| \mathbf{u} \cdot \mathbf{n} \|_{\partial K}^2 + \| \mathbf{n} \times \mathbf{u} \|_{\partial K}^2 = 0,
\]
which implies that \( \mathbf{L} = 0 \) on \( K, \mathbf{u} \cdot \mathbf{n} = 0 \) and \( \mathbf{n} \times \mathbf{u} \) on \( \partial K \), and hence \( \mathbf{u} = 0 \) on \( \partial K \). It follows that (63a), after the second term is integrated by parts, becomes
\[
(\nabla \mathbf{u}, \mathbf{G})_K = 0, \quad \forall \mathbf{G} \in \mathbf{V}_h(K),
\]
from which we infer that \( \mathbf{u} \) must be a constant vector on \( K \), in fact null vector, since it vanishes on the boundary.

Similarly, integrating the second term in (63b) by parts and using the results \( \mathbf{L} = 0 \) and \( \mathbf{u} = 0 \) we obtain
\[
(\nabla q, \mathbf{v})_K = 0, \quad \forall \mathbf{v} \in \mathbf{V}_h(K),
\]
which shows that \( q \) must be a constant. At this point, one can observe that there is no extra information to prove the uniqueness of \( q \), or equivalently \( q = 0 \). To resolve this issue, a natural idea is to equip the local solver (63) with a minimal condition to enforce the uniqueness. Similar to [17,15], we choose
\[
\bar{q} = \rho, \text{ on } \partial K,
\] (64)
where \( \bar{q} \) is the average of \( q \) on \( \partial K \) defined as \( \bar{q} := \frac{1}{|\partial K|} \langle q, 1 \rangle_{\partial K} \), and \( \rho \) is a constant function belonging to \( \mathcal{A}_h(\partial \Omega_h) := \left\{ \lambda \in L^2(\partial \Omega_h) : \lambda|_{\partial K} \in P^0(\partial K), \forall K \in \Omega_h \right\} \).

Let us state the well-posedness result in Lemma 5.2.

**Lemma 5.2.** Suppose \( \nu > 0 \). Given \( \hat{u} \) and \( \rho \), there exists a unique solution \((\mathbf{L}, \mathbf{u}, q)\) for the local solver (63)–(64).

**Proof.** It is enough to show that \( \hat{u} = 0 \) and \( \rho = 0 \) implies \( \mathbf{L} = 0, \mathbf{u} = 0 \) and \( q = 0 \). Using the above motivation for the augmented local solver (63)–(64) we need to only show \( q = 0 \), but it is clear since \( q \) is constant and have zero average on \( \partial K \) according to (64). □

We next construct the global system of equations on the mesh skeleton to solve for \( \hat{u} \) and \( \rho \). To this end, it is important to realize that the well-posedness of the augmented local solver (63)–(64) comes at an expense of introducing a new unknown \( \rho \) constant on each element boundary in addition to \( \hat{u} \). Since the HDG flux components corresponding to (58a) and (58c) automatically satisfy the conservation constraint (16b), we are left with three algebraic constraints for the trace unknown \( \hat{u} \), i.e., \( \forall \mathbf{e} \in A_h(\mathbb{E}), \forall e \in \mathcal{E}_h^\mathbb{E} \),
\[
\left\{ \left\{ -L \cdot \mathbf{n} + q \mathbf{n} + \sqrt{2} \left( (\mathbf{u} - \hat{u}) \cdot \mathbf{n} \right) \mathbf{n} - \mathbf{n} \times (\mathbf{u} - \hat{u}) \right\}_e, \mu \right\}_e = 0.
\] (65)
We still need one extra equation for \( \rho \). To seek this equation we note that if we take \( r \) as a constant function so that \( r|_{\partial K} = \bar{r}|_{\partial K} \in \mathcal{A}_h(\partial K) \), then (63c) reduces to
\[ (1_{\partial K \setminus \partial \Omega_1} \hat{u} \cdot n, \tau)_{\partial K} = 0, \]

which implies

\[ (1_{\partial K \setminus \partial \Omega_1} \hat{u} \cdot n, \xi)_{\partial K} = 0, \quad \forall \xi \in \overline{K} (\partial K), \tag{66} \]

since the trace operator from the space of constant functions on \( K \) to \( \overline{K} (\partial K) \) is surjective. Since \( \hat{u} \) is viewed as one of the inputs (the forcing) for the local solver, (66) is the compatibility condition for \( \hat{u} \) and this is exactly the extra equation we are looking for. Thus, not only the usual conservation condition (65) but also the compatibility condition (66) constitutes the HDG global solver whose well-posedness is now discussed.

**Theorem 5.3.** Assume \( \nu > 0 \). The HDG scheme consists of the local solver (63)–(64), the global solver (65)–(66), and the HDG flux (59) has a unique solution \((L, u, q, \hat{u}, \rho)\).

**Proof.** We again employ the energy method as in previous sections. The task at hand is to show that \( f = 0 \) implies \( L = 0, u = 0, q = 0, \hat{u} = 0 \) and \( \rho = 0 \). Integrating by parts the second term of (63a) and (63b), adding the resulting equations of (63) all together, then summing over all elements give

\[
\begin{align*}
\|L\|_{\Omega_h, \nu}^2 & - \sum_K (1_{\partial K \setminus \partial \Omega_1} n \otimes \hat{u}, L)_{\partial K} - \sqrt{2} \left( (u - 1_{\partial K \setminus \partial \Omega_1} \hat{u}) \cdot n \right)_{\partial K} \\
& - \sum_K (n \times n \times (u - 1_{\partial K \setminus \partial \Omega_1} \hat{u}) \cdot n)_{\partial K} - (1_{\partial K \setminus \partial \Omega_1} \hat{u} \cdot n, q)_{\partial K} = 0.
\end{align*}
\]

(67)

On the other hand, taking \( \mu = \hat{u} \) in (65) and subtracting the resulting equation from (67) we arrive at

\[
\begin{align*}
\|L\|_{\Omega_h, \nu}^2 & - \sum_K (1_{\partial K \setminus \partial \Omega_1} n \otimes \hat{u}, L)_{\partial K} - \sqrt{2} \left( (u - 1_{\partial K \setminus \partial \Omega_1} \hat{u}) \cdot n \right)_{\partial K} \\
& - \sum_K (n \times n \times (u - 1_{\partial K \setminus \partial \Omega_1} \hat{u}) \cdot n)_{\partial K} = 0,
\end{align*}
\]

which shows that \( L \) must vanish, i.e. \( L = 0 \), on \( K \). It also implies that \( u \cdot n = 1_{\partial K \setminus \partial \Omega_1} \hat{u} \cdot n \) and \( n \times n \times u = 1_{\partial K \setminus \partial \Omega_1} n \times n \times \hat{u} \) on \( \partial K \). Thus, \( u = 1_{\partial K \setminus \partial \Omega_1} \hat{u} \) on \( \partial K \). In other words, \( L = 0 \) on \( \Omega_h \), \( u = \hat{u} \) on \( \mathcal{E}_h^0 \), and \( u = 0 \) on \( \mathcal{E}_h^1 \). Since \( \hat{u} \) is single-valued on \( \mathcal{E}_h^1 \), \( u \) must be continuous across \( \Omega_h \).

Now integrating the second term of (63a) by parts we obtain

\[
(\nabla u, G)_{\Omega_h} = 0, \quad \forall G \in V_h (\Omega_h),
\]

and hence \( u \) must be a constant. But \( u \) is continuous and vanishes on \( \mathcal{E}_h^0 \), it must vanish over \( \Omega_h \). As a result, \( \hat{u} = 0 \) on \( \mathcal{E}_h^0 \).

Next, integrating the second term of (63b) by parts and using the results \( L = 0 \) and \( u = 0 \) we obtain

\[
(\nabla q, \psi)_{\partial K} = 0, \quad \forall \psi \in V_h (K),
\]

which shows that \( q \) must be piecewise constant. On the other hand, if \( \psi \) is taken as a constant vector, it is clear from (63b) that \( \hat{q} = 0 \). Thus, \( q = 0 \) on \( K \). It follows that \( \rho \) also vanishes on \( \partial \Omega_h \) due to (64) and this ends the proof. \( \square \)

**Remark 5.4.** As can be observed, though (64) is introduced to enforce the uniqueness of \( q \) locally on each element, and hence the local solver, it automatically enforces the uniqueness of \( q \) globally. This is the reason why we do not need to explicitly enforce (58d) in our HDG scheme.

Though there are other HDG schemes such as those in [16], let us discuss only the similarity and difference between ours and that proposed in [17,15] since they are related. Indeed, let us derive the HDG family in [17,15] from ours. To that end, we first simplify our HDG flux (62) by removing \( \sqrt{2} \left( (u - \hat{u}) \cdot n \right) \). Next, let us introduce a stabilization matrix \( \mathcal{S} \) to penalize the deviation of the tangent velocity \( n \times n \times (u - \hat{u}) \) so that the HDG flux (62) now becomes

\[
\hat{F} \cdot n = \begin{bmatrix}
-n \otimes \hat{u} \\
-L \cdot n + qn - \mathcal{S} n \times n \times (u - \hat{u}) \\
\hat{u} \cdot n
\end{bmatrix}.
\]

This is exactly the penalized HDG flux proposed in [17,15] for the velocity-pressure-gradient form of the Stokes flow. Clearly, by taking \( \mathcal{S} \) as the identity we obtain a simplified version of our upwind HDG flux (62) in which \( \sqrt{2} \left( (u - \hat{u}) \cdot n \right) \) is omitted.

**Remark 5.5.** We note that there are other forms of the Stokes equations, see [15] for example. Following our HDG construction, we can similarly derive HDG methods for these forms as well. Since not much more benefit can be gained, we omit the details.
6. A HDG framework for Friedrichs’ systems

We have used our unified HDG framework to constructively derive HDG schemes/families for three special PDEs in Sections 3, 4, and 5. Our intention has been to show that by employing our single abstract framework, namely hybridizing the upwind Godunov flux, one could derive different classes of HDG methods by exploiting the special structure of the underlying PDEs. This is then followed by the well-posedness analysis. Though there are differences in the detailed manipulations, the construction and analysis are similar for all PDEs that we have considered. A natural question is whether we can unify not only the construction, as we have done, but also the analysis for a large family of PDEs. The answer to this question is inspired by our previous work on a unified discontinuous Petrov–Galerkin framework for Friedrichs’ systems [33].

It is the purpose of this section that we generalize the analysis of our proposed upwind HDG framework to a large class of linear differential operators of Friedrichs’ type. We are interested in Friedrichs’ systems since they embrace a large class of elliptic, parabolic, hyperbolic, and mixed-type PDEs. We start with the standard assumptions (see, e.g., [39,30,29]):

\[ C \in \left[ L^\infty (\Omega) \right]^{m,m}, \] (68a)

\[ A_k \in \left[ L^\infty (\Omega) \right]^{m,m}, \quad k = 1, \ldots, d, \quad \text{and} \quad \sum_{k=1}^{d} \partial_k A_k \in \left[ L^\infty (\Omega) \right]^{m,m}, \] (68b)

\[ A_k = (A_k)^T \text{ a.e. in } \Omega, \quad k = 1, \ldots, d, \] (68c)

\[ C + C^T + \sum_{k=1}^{d} \partial_k A_k \geq 2 \alpha_0 I \text{ a.e. in } \Omega, \] (68d)

where \( \alpha_0 > 0 \) is some coercivity constant. In this paper, we restrict our analysis to the boundary condition defined by

\[ (A - |A|) u = 0 \text{ on } \partial \Omega, \] (68e)

and we assume that

\[ \mathcal{N}(A - |A|) + \mathcal{N}(A + |A|) = \mathbb{R}^m, \] (68f)

with \( \mathcal{N} \) denoting the nullspace of its argument.

With the simple setting given by (68), one can show that the abstract PDE (1) is well-posed, i.e., the exact solution \( u^e \) exists and it is unique [40]. The well-posedness for more general settings with other boundary conditions can be found in [41–44].

The abstract HDG framework in Section 2 is still applicable for the Friedrichs’ setting. In particular, the upwind HDG method for PDEs of Friedrichs’ type consists of the local solver (16a), the conservation condition (16b) and the HDG flux (15). In particular: find \( u \in V_h(K) \) and \( \hat{u} \in A_h(\mathcal{E}_h^e) \) such that for all \( v \in V_h(K) \) we have

\[ - \sum_{k=1}^{d} (A_k u, \partial_k v)_K + (C u, v)_K + (A u + |A| (u - \hat{u}), v)_{\partial K} = (f, v)_K, \] (69)

and

\[ \left[ [A u + |A| (u - \hat{u})], \mu \right]_e = 0, \quad \forall \mu \in A_h(e), e \in \mathcal{E}_h^e. \] (70)

We next describe how to enforce the boundary condition (68e) in our abstract HDG setting. In previous sections, we have first reduced the number of trace unknowns by exploiting exact relationships of the upwind states and the structure of the PDE under consideration, and then enforced the boundary condition directly on the reduced trace unknowns. However, for the abstract HDG flux (15), it is not clear how to do so and we have to resort to the full trace unknown \( \hat{u} \). In particular, we still enforce the boundary condition (68e) through the trace unknown, i.e.,

\[ (A - |A|) \hat{u} = 0 \text{ on } \mathcal{E}_h^e. \] (71)

Recall that \( A = R S R^{-1} = R S^T \) and \( |A| = R |S| R^T \) since \( A \) is symmetric. Consequently, (71) is equivalent to \( (R^T)^T \hat{u} = 0 \) on \( \mathcal{E}_h^e \), where \( R^\pm \) are the collection of eigenvectors corresponding to non-negative (outflow) and negative (inflow) eigenvalues, respectively. That is, (71) is equivalent to the condition that the inflow components of \( \hat{u} \) must vanish. Clearly, it is not enough to determine \( \hat{u} \), thus we further require that

\[ (A u + |A| (u - \hat{u}), \mu) = (A \hat{u}, \mu)_e, \quad \forall \mu \in A_h(e), e \in \mathcal{E}_h^e. \] (72)

In other words, the HDG flux must be the same as the flux determined by the trace \( \hat{u} \) on the boundary \( \partial \Omega \). We can rewrite (72) as

\[ The exact solution \( u^e \) belongs to a subset of the graph space that satisfies the boundary condition (68e).
\[
\langle (A + |A|) (u - \hat{u}) , \mu \rangle_e = 0, \quad \forall \mu \in \Lambda_h(e), e \in \mathcal{E}_h^\partial,
\]
which is equivalent to
\[
(A + |A|)(u - \hat{u}) = 0 \text{ on } \mathcal{E}_h^\partial,
\]
(73)
or
\[
(R^\top)(u - \hat{u}) = 0 \text{ on } \mathcal{E}_h^\partial.
\]
That is, the outflow components of \( \hat{u} \) must be the same as those of \( u \) on the boundary.

We first analyze the local solver.

**Lemma 6.1.** The local solver (69) is well-posed.

**Proof.** Since the local solver is linear and \( u \) belongs to finite dimensional space \( \mathbf{V}_h(K) \), it is sufficient to prove that \( f = 0 \) and \( \hat{u} = 0 \) implies \( u = 0 \). To begin, taking \( v = u \) in (69) and integrating the first term by parts we obtain
\[
\frac{1}{2} \left( \left[ C + C^T + \sum_{k=1}^{d} \partial_k A_k \right] u , u \right)_K + \left( \frac{1}{2} A + |A| \right) u , u \right)_{\partial K} = 0
\]
which yields \( u = 0 \) in \( K \) since the left hand side is non-negative owing to the coercivity condition (68d), and \( \frac{1}{2} A + |A| \geq 0 \). \( \square \)

The well-posedness of the global solver for \( \hat{u} \) is now established.

**Theorem 6.2.** Suppose \( \mathcal{N}(A) = \{0\} \). There exists a unique trace solution \( \hat{u} \) for the HDG method defined by the local solver (69), the global solver (70), and the boundary conditions (71) and (73).

**Proof.** Due to the finiteness and linearity of the global system, it is sufficient to prove that \( \hat{u} = 0 \) is the only solution if \( f = 0 \). To that end, we again employ the energy approach by taking \( v = u \) in (69), integrating the first term by parts, and summing over all elements to arrive at
\[
\frac{1}{2} \left( \left[ C + C^T + \sum_{k=1}^{d} \partial_k A_k \right] u , u \right)_\Omega_h + \sum_{K} \left( \frac{1}{2} Au + |A| (u - \hat{u}) , u \right)_{\partial K} = 0.
\]
(74)
On the other hand, taking \( \mu = \hat{u} \) in (70), summing over all \( e \in \mathcal{E}_h^\partial \), and then subtracting the resulting equation from (74) we obtain
\[
\frac{1}{2} \left( \left[ C + C^T + \sum_{k=1}^{d} \partial_k A_k \right] u , u \right)_\Omega_h + \sum_{e \in \mathcal{E}_h^\partial} \left( \frac{1}{2} Au + A (u - \hat{u}) , u \right)_e + \sum_{K \in \partial K \cap \mathcal{E}_h^\partial} \sum_{e \in \partial K \cap \mathcal{E}_h^\partial} \left( \frac{1}{2} Au + A (u - \hat{u}) , u - \hat{u} \right)_e + \langle |A| (u - \hat{u}) , u - \hat{u} \rangle_e = 0.
\]
(75)
Now, by inspection, the following identity is true for any \( e \):
\[
\langle A (u - \hat{u}) , \hat{u} \rangle_e = \frac{1}{2} \left[ \langle Au , \hat{u} \rangle_e + \langle A (u - \hat{u}) , u - \hat{u} \rangle_e - \langle Au , u \rangle_e \right].
\]
If we apply this identity to internal faces, i.e. \( e \in \partial K \cap \mathcal{E}_h^\partial \), (75) becomes
\[
\frac{1}{2} \left( \left[ C + C^T + \sum_{k=1}^{d} \partial_k A_k \right] u , u \right)_\Omega_h + \sum_{e \in \mathcal{E}_h^\partial} \left( \frac{1}{2} Au + A (u - \hat{u}) , u \right)_e
\]
\[
+ \sum_{K \in \partial K \cap \mathcal{E}_h^\partial} \sum_{e \in \partial K \cap \mathcal{E}_h^\partial} \left( \langle |A| + \frac{1}{2} A \rangle (u - \hat{u}) , u - \hat{u} \right)_e = 0.
\]
For boundary faces, i.e. \( e \in \partial K \cap \mathcal{E}_h^\partial \), using (71) and (73) one can show that
\[
\left( \frac{1}{2} Au + A (u - \hat{u}) , u \right)_e = \frac{1}{2} \langle |A| u , u \rangle_e, \quad \forall e \in \mathcal{E}_h^\partial.
\]
Combining the last two equations gives
\[
\frac{1}{2} \left( \left[ C + C^T + \sum_{k=1}^d \delta_k A_k \right] \mathbf{u}, \mathbf{u} \right)_{\Omega_b} + \frac{1}{2} \sum_{e \in \mathcal{E}_h^0} \langle |A| \mathbf{u}, \mathbf{u} \rangle_e + \sum_{k} \sum_{e \in K \cap \mathcal{E}_h^0} \left( \langle |A| + \frac{1}{2} A \rangle (\mathbf{u} - \hat{\mathbf{u}}), \mathbf{u} - \hat{\mathbf{u}} \right)_e = 0
\]
whose left hand side is non-negative due to the coercivity condition (68d), $|A| \geq 0$, and $|A| + \frac{1}{2} A \geq 0$. Consequently, we conclude that $\mathbf{u} = \mathbf{0}$ on $\Omega_b$ and $\langle |A| \hat{\mathbf{u}}, \hat{\mathbf{u}} \rangle_e = 0$ for all $e \in \mathcal{E}_h^0$. Since $\mathcal{N}(A) = \{0\}$, we conclude that $\hat{\mathbf{u}} = \mathbf{0}$. □

**Remark 6.3.** Note that $\mathcal{N}(A) = \{0\}$, a sufficient condition, implies condition (68f), and therefore the exact solution $\mathbf{u}'$ is a priori unique. Thus, it is (not surprisingly) the key to ensure the uniqueness of $\hat{\mathbf{u}}$ though it is not needed for the uniqueness of $\mathbf{u}$. It should be also pointed out that we have analyzed our upwind HDG framework for Friedrichs’ systems with full coercivity condition (68d). The analysis for multifield systems with partial coercivity [32] is more delicate and we leave the details for future work.

Following the previous sections one can define a penalized HDG flux as
\[
\mathbf{F}^* \cdot \mathbf{n} = \mathbf{A} \mathbf{u} + \mathcal{T} (\mathbf{u} - \hat{\mathbf{u}}),
\]
where $\mathcal{T}$ is some positive definite stabilization matrix. Clearly, when $\mathcal{T} = |A|$, we recover our upwind HDG scheme.

**Remark 6.4.** The aim of this section is to show that an abstract upwind HDG framework and its analysis are possible for a large class of PDEs. In practice one typically deals with a concrete PDE, and in that case we suggest to follow our systematic construction of HDG schemes in Sections 3–5 to exploit the structure of the problem. This most likely minimizes the number of trace unknowns, and hence yielding HDG methods with least number of trace unknowns.

7. Hybridization of other numerical fluxes

Perhaps the most popular numerical flux is that of Godunov with exact Riemann solutions, and we have used it to motivate a constructive derivation of the upwind HDG framework. At the heart of our construction is the exploration of various identities satisfied by the upwind states to devise HDG methods with least number of trace unknowns. A question that immediately arises is whether we can construct HDG method for other existing numerical fluxes, for example, those documented in [35,45].

In this section, we restrict our derivation of HDG method for the Lax–Friedrichs (LF) numerical flux whose standard form is given by
\[
\mathbf{F}^* \cdot \mathbf{n} = \frac{1}{2} \left[ \mathbf{F} (\mathbf{u}^-) + \mathbf{F} (\mathbf{u}^+) \right] \cdot \mathbf{n}^- + \frac{1}{2} \max_i |\partial_i| (\mathbf{u}^- - \mathbf{u}^+).
\]
Clearly, for scalar problem, i.e. $m = 1$, LF flux is identical to the Godunov one. This suggests we hybridize the former in a similar manner as we have done for the latter. A natural idea is to consider the following hybridized version of the LF flux
\[
\mathbf{F}^* \cdot \mathbf{n} = \mathbf{A} \mathbf{u} + \max_i |\partial_i| (\mathbf{u} - \mathbf{u}^+).
\]
Clearly, the hybridized LF flux (77) coincides with the upwind HDG flux for scalar problem. Moreover, it is equivalent to the original LF flux (76).

Let us now apply the hybridized LF flux (77) to the convection–diffusion–reaction PDE and compare it with the hybridized Godunov counterpart. (The construction for the Maxwell and Stokes equations is similar, and hence omitted.) For this case, it is easy to see that
\[
\theta_{\text{max}} := \max_i |\partial_i| = \frac{1}{2} \sqrt{|\mathbf{\beta} \cdot \mathbf{n}|^2 + 4},
\]
and (77) becomes
\[
\mathbf{F}^* \cdot \mathbf{n} = \begin{bmatrix}
\mathbf{u} \mathbf{n} + \theta_{\text{max}} (\mathbf{\sigma} - \mathbf{\sigma}^*) \\
\mathbf{\beta} \cdot \mathbf{v} + \mathbf{\sigma} \cdot \mathbf{n} + \theta_{\text{max}} (\mathbf{u} - \mathbf{u}^*)
\end{bmatrix}.
\]
Here, $\mathbf{u}^*, \mathbf{\sigma}^*$ are no longer the exact upwind states. Indeed, conservation equation (13) allows us to obtain the following identities
\[
\theta_{\text{max}} (\mathbf{\sigma} - \mathbf{\sigma}^*) = -\frac{\theta_{\text{max}}}{2} \left[ |\mathbf{\sigma}| + \frac{1}{2} \left[ \mathbf{u} \mathbf{n} \right] \right],
\]
\[
\theta_{\text{max}} (\mathbf{u}^* - \mathbf{u}) = \frac{\theta_{\text{max}} \left[ \mathbf{\beta} \cdot \mathbf{n} - \theta_{\text{max}} \right]}{2} \left[ \mathbf{u} \right] + \frac{1}{2} \left[ \mathbf{\sigma} \cdot \mathbf{n} \right].
\]
As can be seen, unlike the hybridized Godunov flux, it is not clear how to eliminate \( \sigma^* - \sigma \) in terms of \( u^* - u \) and vice versa. Thus, the hybridized LF flux (78), though being much simpler than the Godunov counterpart (18), yields more trace unknowns, and hence more expensive. Nevertheless, for nonlinear PDEs, it may be a better choice since we, in general, cannot reduce the number of trace unknowns for the Godunov flux either.

8. Construction of other HDG variants

In this section, we will construct a variant of our HDG framework that can be used to recover many existing HDG methods. To that end, notice that we can write

\[
Au + |A| (u - \hat{u}) = A\hat{u} + (A + |A|) (u - \hat{u}).
\]

This leads us to define a new family of HDG fluxes

\[
\hat{F} \cdot n = A\hat{u} + \mathcal{T} (u - \hat{u}),
\]

which becomes the upwind HDG flux when \( \mathcal{T} = (A + |A|) \). Clearly, we can revisit all examples in this paper with this HDG flux form to derive variants of HDG schemes for various type of PDEs. For example, we can rewrite the HDG flux (22) for the convection–diffusion–reaction PDE as

\[
\hat{F} \cdot n = \begin{bmatrix}
\hat{u}n_1 \\
\hat{u}n_2 \\
\hat{u}n_3 \\
\beta \cdot \hat{u}n + \sigma \cdot n + \frac{1}{2} (\alpha + \beta \cdot n) (u - \hat{u})
\end{bmatrix},
\]

and deduce its penalized family as follows

\[
\hat{F} \cdot n = \begin{bmatrix}
\hat{u}n_1 \\
\hat{u}n_2 \\
\hat{u}n_3 \\
\beta \cdot \hat{u}n + \sigma \cdot n + \tau (u - \hat{u})
\end{bmatrix}.
\]

A careful look at (79) shows that we have rediscovered the HDG family proposed in [11]. One can follow the same exercise to derive families of HDG methods of this type for other PDEs in a straightforward manner, so let us skip the details here.

9. HDG schemes for discontinuous material

In this section we present an extension of our HDG framework for linear PDEs with discontinuous material, that is, \( A_h \)’s are possibly discontinuous across \( e \in \mathcal{E}_h \). In this case, the Godunov flux still has exactly the same form as (11) or (12), but \( A \) is computed using information from the appropriate element under consideration. For example,

\[
F^+ \cdot n^- = F (u^-) \cdot n^- + |A^-| (u^- - u^+).
\]

One can then go through all the steps presented in this paper to derive the corresponding HDG family for each PDE of interest since the idea remains the same. For example, one first seeks various relations between components of the upwind state \( u^+ \) to eliminate their dependencies. One then defines the HDG flux based on the final form (e.g. with least number of trace unknowns) of the Godunov flux. Together with the local solver (16a) and conservation constraint (16b), one obtains desirable HDG schemes. However, the arithmetic is more involved since we need to distinguish materials between neighbor elements. We omit the details here to avoid unnecessarily lengthy paper.

It is important to realize that most of the theory for Friedrichs also holds for this case. In particular, the well-posedness of the local solver remains true, and the global solver is well-posed if we additionally assume that \( A^- + A^+ \leq 0 \) on \( e \in \mathcal{E}_h \). At this point, this assumption is merely mathematically convenient since using it allows us to conclude

\[
- \sum_K \left( \frac{1}{2} A \hat{u}, \hat{u} \right)_{|K \cap \mathcal{E}_h} \geq 0,
\]

and hence the proof of Theorem 6.2 goes through. Whether this assumption is practical is out of the scope of the paper. Clearly, one may find a proof that does not require (81) and in that case the assumption \( A^- + A^+ \leq 0 \) can be completely discarded.

One can similarly define a penalized family of HDG fluxes with double-valued stabilization matrix \( \mathcal{T} \), e.g.,

\[
\hat{F} \cdot n^- = F (u^-) \cdot n^- + \mathcal{T}^- (u^- - \hat{u}).
\]

HDG methods with double-valued stabilization parameters can be found, for example, in [16,11].
10. HDG schemes for nonlinear PDEs

In this section we extrapolate our upwind HDG schemes to nonlinear PDEs. Specifically, we define HDG flux for nonlinear PDEs as

$$\hat{F} \cdot \mathbf{n} = F(u) \cdot \mathbf{n} + |A|(u - \hat{u}),$$

where $A$ is now a function of $u$ given by

$$A := \sum_{k=1}^{d} \partial_k F_k(u) n_k.$$

Alternatively, one can define HDG flux following the spirit of Roe flux [46], i.e.,

$$\hat{F} \cdot \mathbf{n} = F(u) \cdot \mathbf{n} + |A(\hat{u})|(u - \hat{u}),$$

in which $A(\hat{u})$ is computed using the trace unknown $\hat{u}$, i.e.,

$$A(\hat{u}) = \sum_{k=1}^{d} \partial_k F_k(\hat{u}) n_k.$$  \hfill (82)

On the other hand, based on (77) and (82), we can define a hybridized LF flux as

$$\hat{F} \cdot \mathbf{n} = F(u) \cdot \mathbf{n} + \max_i |\partial_i (A(\hat{u}))|(u - \hat{u}),$$

where we denote by $\partial_i (A(\hat{u}))$ the $i$th eigenvalue of $A(\hat{u})$.

Finally, HDG fluxes can be extrapolated from results in Section 8, i.e.,

$$\hat{F} \cdot \mathbf{n} = F(u) \cdot \mathbf{n} + \max_i |\partial_i (A(\hat{u}))|(u - \hat{u}),$$

or

$$\hat{F} \cdot \mathbf{n} = F(u) \cdot \mathbf{n} + |A(\hat{u})|(u - \hat{u}),$$

and these are exactly those proposed in [20,47] for compressible Euler and Navier–Stokes equations.

11. Numerical results

We start this section by pointing out that there are vast existing numerical results for HDG methods on Laplace/Poisson equation [10], convection–diffusion equation [11–14], Stokes equation [15–17], Euler and Navier–Stokes equations [19–21], Maxwell equation [22–24], acoustics and elastodynamics [25], Helmholtz equation [26,27], eigenvalue problem [28], etc. Instead of reproducing these results, let us present numerical results comparing our upwind HDG framework with some existing HDG schemes on transport equation, convection–diffusion equation, compressible Euler equation, and shallow water equation.

11.1. Steady convection (transport) equation

In this section, we apply our upwind HDG method to convection equation of the form

$$\nabla \cdot (\beta u) + \nu u = f, \quad \text{in } \Omega,$$

and compare its solutions with those of the upwind DG scheme. As a numerical demonstration, we take an example from [48] in which $\Omega = (0,2) \times (0,1)$, $\beta = (1 + \sin(\pi y/2), 2)$, $\nu = 0$, $f = 0$, and the inflow boundary condition

$$g = \begin{cases} 1 & x = 0, 0 \leq y \leq 1 \\ \sin^6(\pi x) & 0 < x \leq 1, y = 0 \\ 0 & 1 \leq x \leq 2, y = 0 \end{cases}$$

This problem admits an exact solution using the method of characteristics.

Fig. 1(a) shows the mesh that we use for both HDG and DG schemes. To verify the theoretical result in Corollary 3.5, we compute the $L^2$-norm of the difference between the HDG trace solution $\hat{u}$ and the upwind state $u^*$, and of the difference between the HDG solution $u_{\text{HDG}}$ and the DG solution $u_{\text{DG}}$. Table 1 shows these differences for various solution orders $p = (1, 2, 3, 4, 5)$. As can be seen, the differences are (machine) zeros for all solution orders as predicted by the theory.

We also show the HDG solution $u$ with $p = 3$ in Fig. 1(b) and its convergence to the exact solution at the outflow boundary $y = 1$ in Fig. 1(c). As expected, as the solution order increases the HDG solution is more accurate in the smooth regions, but oscillates in regions with high gradient since we do not employ any limiter or artificial diffusion.
Recall that we have constructed HDG method by hybridizing the DG method (in fact the Godunov flux). This procedure could be understood as a redesign of the DG method. The beauty of this redesign is that the number of coupled unknowns is substantially reduced and the usual DG unknowns can be computed element-by-element independent of each other. It is therefore useful to compare the number of coupled unknowns and the execution time for both methods. We do so for the above two dimensional transport equation on triangular meshes. The result is shown in Fig. 1(d) which plots the ratios of DG/HDG coupled unknowns and DG/HDG execution time. As can be seen, as the solution order increases these ratios increase as well. For the mesh \( h = 0.2 \) in Fig. 1(a) and its one level of refinement \( h = 0.1 \), we gain (almost) an order of magnitude in solution time and in number of coupled unknowns for \( p = 15 \); also the finer the mesh is, the more savings the HDG method provides. Note that, we have computed solution \( u \) in our HDG scheme in a serial manner, element-after-element. The time gain would be even more if we did it in parallel. These gains are expected to be more substantial for three dimensional problems.

11.2. Convection–diffusion equation

In this section we compare our upwind HDG scheme against both upwind and centered trace HDG methods proposed in [11, Section 3.6.3] for convection–diffusion equation, i.e. (17) with \( v = 0 \). We take \( \beta = (1, 2) \) and \( \Omega = [0, 1]^2 \). The right hand side \( f \) is chosen so that the exact solution is given by

\[
u_{e} = \exp(x + y) \sin(\pi x) \sin(\pi y),
\]

and Dirichlet boundary condition on \( \partial \Omega \) is imposed using the exact solution. In Fig. 2(a) we show a plot of \( \nabla \cdot \beta \) inside elements and \( \beta \) on element boundaries with \( h = 0.2 \), accompanied by third order \( (p = 3) \) upwind HDG solution \( u \) in Fig. 2(b), \( \sigma_x \) in Fig. 2(c), and \( \sigma_y \) in Fig. 2(d).

What we are interested in this example is the behavior of the aforementioned HDG methods for varying diffusion coefficients \( \epsilon \) as the mesh is refined. Let us first study the convergence rate as \( \epsilon \) decreases. In Figs. 3(a), 3(c), and 3(e) we present the observed convergence order with \( p = 3 \) as the diffusion coefficient \( \epsilon \) changes. As can be seen, the convergence for both \( u \) and \( \bar{u} \) is optimal, with rate 4 and 3.5, respectively, for all methods when \( \epsilon > 10^{-2} \). When the equation is in convection-dominated regime, the behavior of these methods is different. In particular, both upwind trace and centered...
trace HDG methods are unstable for $\epsilon < 10^{-2}$, while it is for the upwind HDG. This behavior is expected for the upwind HDG since it is equivalent to the standard upwind DG by construction.

To gain some understanding on why the unstability may happen for the others let us recall that the stabilization for the upwind trace HDG scheme is given by

$$\tau = (\epsilon + |\beta \cdot n|) \frac{|\beta \cdot n| + \beta \cdot n}{2|\beta \cdot n|} = (\epsilon + |\beta \cdot n|) \frac{1 + \text{sgn}(\beta \cdot n)}{2}$$

and

$$\tau = (\epsilon + |\beta \cdot n|)$$

for the centered trace HDG scheme. In Fig. 2(a), for edge $e$ with $\beta \cdot n \leq 0$ (negative but machine zero), then $\tau \approx 0$. On the other hand, though $\nabla \cdot \beta$ is identically zero analytically, it is not in our nodal implementation [49], but in fact machine zero with positive and negative signs, as shown in Fig. 2(a). Now recall from [11, Lemma 3.1 and Theorem 3.1] that a sufficient condition for the well-posedness is that $\nabla \cdot \beta \geq 0$ and $\tau > \frac{1}{2} \beta \cdot n$. It follows that the well-posedness is not guaranteed in our implementation of the upwind and centered trace HDG methods. Nevertheless, when $\epsilon$ is sufficiently large, it does not seem to be a problem.

Next, let us examine the dependence of $L^2$-errors on $\epsilon$, especially when $\epsilon$ is large. As an example, we take $p = 3$ and $h = 0.05$ (two levels of refinement of the mesh in Fig. 2(a)). In Figs. 3(b), 3(d), and 3(f) are the $L^2$-errors for $u$ and $\hat{u}$ of the three methods as $\epsilon$ increases. As can be observed, the error in the trace unknown $\hat{u}$ for all methods is robust in $\epsilon$, i.e., it is independent of $\epsilon$, and so is the error in $u$ for the upwind and centered trace HDG methods. The error in $u$ for the upwind HDG increases though maintaining the optimal convergence order.

In summary, our numerical results seem to suggest that either upwind or centered trace HDG method is preferred if the robustness with large $\epsilon$ (diffusion-dominated) is of importance. On the other hand, if stability for a wide range of $\epsilon$ is desired, the upwind HDG is likely to be a better choice.

11.3. Compressible Euler equation

The purpose of this section is to compare our proposed HDG flux (84) and Nguyen et al. flux (85) for nonlinear PDEs. To that end, we next consider the unsteady Euler equation in the conservative form

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{u}) = \mathbf{f},$$

(86)
where the conservative state $\mathbf{u}$, and the $x$- and $y$-components of the flux $\mathbf{F}$ are given by

$$
\mathbf{u} := \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix}, \quad 
\mathbf{F}_x := \begin{bmatrix} \rho u \\ \rho u^2 + P \\ \rho uv \\ u(E + P) \end{bmatrix}, \quad 
\mathbf{F}_y := \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + P \\ v(E + P) \end{bmatrix}.
$$

We assume that the pressure $P$ satisfies the state equation

$$
E = \frac{p}{\gamma - 1} + \frac{1}{2p} \left( (\rho u)^2 + (\rho v)^2 \right),
$$
Fig. 4. Compressible Euler equation: plotted are the HDG solutions for conservative variables ($\rho, \rho u, \rho v, E$) with Crank–Nicholson temporal discretization with time stepsize of $10^{-6}$ and 100 time steps.

Fig. 5. Compressible Euler equation: one-dimensional cuts along the $x$-axis to show the evolution of ($\rho, \rho u, \rho v$) in time. Red-dash are HDG results and solid-black are the exact ones. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 6. Compressible Euler equation: convergence rate for HDG methods with our HDG flux (84) and Nguyen et al. flux (85).
**Fig. 7.** Shallow water equation: convergence rate for HDG methods with our HDG flux (84) and Nguyen et al. flux (85).

**Fig. 8.** Evolution of the water depth $H$ for flat bathymetry case.

with $y = 1.4$ for monoatomic gas. Here, $\rho$ is the density, $u$ and $v$ are the $x$- and $y$-components of velocity, and $E$ denotes the total energy. The domain under consideration is $\Omega = [3.5, 5.5] \times [-1, 1]$. For numerical example, we choose the forcing function $f$ so that the exact solution is the following vortex [49]

$$
\rho = \left[1 - \frac{(y - 1)}{16y^2\pi^2} \beta^2 e^{2(1-R^2)}\right]^{\frac{1}{2}}, \quad u = 1 - \beta e^{(1-R^2)} \frac{(y - y_0)}{2\pi},
$$

$$
v = \beta e^{(1-R^2)} \frac{(x - x_0)}{2\pi}, \quad P = \rho^\gamma,
$$

with $R^2 = (x-t-x_0)^2 + (y-y_0)^2, x_0 = 0.5, y_0 = 0,$ and $\beta = 5$. For simplicity, the exact solution is used to enforce the boundary condition.

Since the Euler equation is nonlinear, both the local solver (16a) and the conservative constraints are nonlinear too. For time discretization, we use the simple Crank-Nicholson method and it is enough for our purpose of testing the HDG spatial discretization. Similar to [12,47] we use Newton method to solve the resulting nonlinear algebraic equations. To see the structure of the flow fields, in Fig. 4 we plot the conservative variables ($\rho, \rho u, \rho v, E$) after 100 time steps with stepsize $10^{-6}$ using our HDG method with (84). To compare the HDG solutions and the exact ones as the vortex moving out of the domain, we also present the evolution of the flow fields on a one-dimensional cut along the $x$-axis in Fig. 5 at $t = \{0.2, 0.6, 1.0, 1.4\}$; as can be seen the HDG and exact solutions are on top of each other. Note that the small step size is
deliberately chosen so that the error is dominated by the spatial discretization using HDG methods. Also, 100 time steps is sufficient for convergence test that is discussed next.

In Fig. 6 are the convergence rates for HDG methods using fluxes (84) and (85) when the mesh is refined with \( h = \{0.125, 0.25, 0.5\} \). It can be seen that both methods give almost identical results. Moreover, the convergence is \( p + 1 \) for all \( p = \{2, 3, 4\} \), and hence optimal. We conclude that the two methods are numerically the same for problems with smooth solutions.

It should be pointed out that our test case is by no means exclusive, but rather simple to demonstrate the convergence rate and to compare with existing methods [19-21]. More extensive test cases such as those in [50,51] are left for future work. Clearly, for cases with shocks our HDG method may need to be equipped with some kind of limiter or artificial diffusion. Some successful attempts have been done in this direction and we refer the readers to the work in [19-21].

11.4. Shallow water equation

The last PDE considered in this paper is the shallow water equation written in the conservative form similar to (86), but now the flux \( F \) and the forcing function \( f \) are given by

\[
F_x := \begin{pmatrix} H u \\ H u^2 + \frac{1}{2} g H^2 \end{pmatrix}, \quad F_y := \begin{pmatrix} H v \\ H u v \\ H v^2 + \frac{1}{2} g H^2 \end{pmatrix}, \quad \text{and} \quad f := \begin{pmatrix} 0 \\ -g b x \\ -g b y \end{pmatrix},
\]
while the conservative variables \( \mathbf{u} \) are defined as
\[
\mathbf{u} := (H, Hu, Hv)^T.
\]
Here, \( H \) is the water depth, \( u \) the depth average velocity component in \( x \)-direction, \( v \) the depth average velocity component in \( y \)-direction, \( b \) the bathymetry, and \( g \) the gravity acceleration.

Following [52] we take \( y = g = 2 \) so that the vortex solution in Section 11.3 is also the exact solution for the shallow water equation with flat bathymetry \( (b = b_x = b_y = 0) \). We choose this example to again compare the HDG methods with our HDG flux (84) and Nguyen et al. flux (85). The rest of the setting is same as that of the Euler equation. For example, we take \( \Omega = [3.5, 5.5] \times [-1, 1] \), and use Crank–Nicholson temporal discretization with 100 time steps of stepsize \( 10^{-6} \). In Fig. 7, we again observe that both methods yield almost identical results. Furthermore, the convergence rate in this case is \( p + \frac{1}{2} \), which is typical for hyperbolic equations using fluxes of Lax–Friedrichs’ type.

The next example we like to apply our HDG method to is that considered in [53,54], namely, the water drop problem with the initial water depth
\[
H(x, y, 0) := 1 + 0.1 \exp \left[ -100 (x - 0.5)^2 - 100 (y - 0.5)^2 \right],
\]
and the flow is initial at rest, i.e.,
\[
Hu(x, y, 0) = Hv(x, y, 0) = 0.
\]

We first consider the case with flat bottom, i.e., \( b = b_x = b_y = 0 \) and the domain of interest is \( \Omega = [0, 1]^2 \). An unstructured triangular mesh with the mesh size \( h = 0.12 \) is generated and we take \( p = 6 \). Wall (slip) boundary conditions are applied to the entire boundary \( \partial \Omega \). Again, the Crank–Nicholson temporal discretization is employed with time step equal to 0.0005 and the simulation is run up to 2000 time steps. Presented in Fig. 8 are the simulated water depths at various time instances \( t = \{0, 0.15, 0.3, 0.5, 0.7, 1\} \). As can be seen, the water depth evolution is very complex. This is also confirmed in Fig. 9 in which we plot the distribution of \( v \) (\( u \) is the same but rotated by 90 degrees) as a function of time. Though there is no analytical solution to compare with, our HDG results look comparable to those in [53,54]. Note that we don’t employ any special techniques to ensure the positivity-preserving and well-balanced properties.
Next we consider the case with non-flat bathymetry proposed in [54] with
\[ b(x, y) := 0.5 \exp \left[ -10(x - 0.75)^2 - 10(y - 0.5)^2 \right]. \]
The initial condition, boundary conditions, and time stepping scheme are the same, but the gravity constant is now \( g = 9.812 \). We show the evolution of the water depth \( H \) after 600 time steps (\( t = 0.3 \)) in Fig. 10, while the velocities \( u \) and \( v \) are shown in Figs. 11 and 12, respectively. As can be seen, due to the bathymetry, the \( x \)- and \( y \)-components of the velocity vector are completely different and very complex.

Again, the examples that we chose are by no means representative but rather simple to demonstrate the convergence rate and to introduce the HDG method as a new numerical method to solve shallow water equations. Clearly, the simple HDG method, a product of our unified framework, is not sufficient to tackle many issues posed by shallow water equations including positivity-preservation, well-balancedness, etc. (see, e.g., [55–61]). Designing HDG methods to address these issues constitutes one of our main future research directions.

12. Conclusions

We have presented a unified framework for the emerging hybridized discontinuous Galerkin (HDG) methods. At the heart of our construction is the hybridization of the Godunov flux. In fact, we have shown that it is a natural recipe to constructively and systematically establish a unified HDG framework for a large class of PDEs including those of Friedrichs’ type.
The unification is fourfold. First, it provides a single constructive procedure to devise HDG schemes for elliptic, parabolic, hyperbolic, and mixed-type PDEs. The key that we exploit is the fact that, for many PDEs, irrespective of their type, the first order form is a hyperbolic system. Second, it reveals the nature of the trace unknowns as the upwind states. Third, it provides a parameter-free HDG framework, and hence eliminating the “usual complaint” that HDG is a parameter-dependent method. Fourth, it allows us to construct many existing HDG methods in a natural manner. In particular, using the unified framework we can rediscover most existing HDG methods and furthermore discover new ones.

We apply the proposed unified framework to three different PDEs: the convection–diffusion–reaction equation, the Maxwell equation in frequency domain, and the Stokes equation. The goal is to present a step-by-step construction of various HDG methods, including the most economic ones with least trace unknowns, by exploiting the particular structure of the underlying PDEs. The well-posedness of the resulting HDG schemes, i.e. the existence and uniqueness of the HDG solutions, is proved. The well-posedness results are also extended and proved for an abstract Friedrichs’ system. We also discuss variants of the proposed unified framework and extend them to the popular Lax–Friedrichs’ flux and to nonlinear PDEs. Numerical results for transport equation, convection–diffusion equation, compressible Euler equation, and shallow water equation are presented to support the proposed unified HDG framework.
Acknowledgements

We are indebted to the anonymous referees for their critical and useful comments that improved the paper substantially. This research is supported by Department of Energy grants DE-SC0010518 and DE-SC0011118, and we are grateful for the support.

References