From Rankine-Hugoniot Condition to a Constructive Derivation of HDG Methods

Tan Bui-Thanh

Abstract This chapter presents a constructive derivation of HDG methods for convection-diffusion-reaction equation using the Rankine-Hugoniot condition. This is possible due to the fact that, in the first order form, convection-diffusion-reaction equation is a hyperbolic system. As such it can be discretized using the standard upwind DG method. The key is to realize that the Rankine-Hugoniot condition naturally provides an upwind HDG framework. The chief idea is to first break the uniqueness of the upwind flux across element boundaries by introducing single-valued new trace unknowns on the mesh skeleton, and then re-enforce the uniqueness via algebraic conservation constraints. Essentially, the HDG framework is a redesign of the standard DG approach to reduce the number of coupled unknowns. In this work, an upwind HDG method with one trace unknown is systematically constructed, and then extended to a family of penalty HDG schemes. Various existing HDG methods are rediscovered using the proposed framework.

1 Introduction

The high-order discontinuous Galerkin (DG) method was originally developed by Reed and Hill [2] for the neutron transport equation, first analyzed in [3, 4], and then has been extended to other problems governed by partial differential equations (PDEs) [11]. Roughly speaking, DG combines advantages of classical finite volume and finite element methods. 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 approaches, and hence more expensive in general.

Tan Bui-Thanh
Department of Aerospace Engineering and Engineering Mechanics, and
Institute for Computational Engineering and Sciences, Austin TX 78712 USA e-mail: tanbui@ices.utexas.edu

Recently, Cockburn and his coworkers have introduced a hybridizable (also known as hybridized) discontinuous Galerkin (HDG) methods for various type of PDEs including Poisson equation [9, 12], and convection-diffusion equation [8, 13]. 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 counterpart. The coupled unknowns are in fact unknown traces introduced on the mesh skeleton, i.e. the faces, to hybridize the numerical flux. Once they are solved for, the usual DG unknowns can be recovered in an element-by-element fashion, completely independent of each other. Thus, the HDG methods are well suited for current and future supercomputing systems. 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.

In this chapter we seek to develop a systematic and constructive hybridized discontinuous Galerkin (HDG) methods for partial differential equations. For concreteness and clarity of the exposition we choose to present our development for convection-diffusion-reaction equation, though it can be extended to other PDEs. This paper is a continuation of our recent effort [7] on unifying the construction and theory HDG method. Unlike [7], in which we construct HDG schemes from the Godunov approach with upwind flux, in this work we discover a new way to unify HDG methods using the Rankine-Hugoniot jump condition. In fact, we shall show that Rankine-Hugoniot jump condition is, perhaps, the most natural way to construct HDG schemes. In the following, we provide step-by-step the construction of our new unified HDG framework and we refer the readers to [9, 12, 8, 13, 7] for a complete description of HDG methodology, its novelties, and its efficiency.

2 Upwind HDG method and its variants for convection-diffusion-reaction equation

In this section we will systematically devise an upwind HDG scheme for convection-diffusion-reaction in the following first order form

$$\varepsilon^{-1}\sigma + \nabla u = 0 \quad \text{in } \Omega, \quad \text{and} \quad \nabla\cdot\sigma + \nabla\cdot(\beta\cdot u) + \nu u = f \quad \text{in } \Omega \tag{1}$$

where $\Omega \subset \mathbb{R}^d$, and we take $d = 3$ for concreteness; the velocity field β is assumed to be continuous; ε is the diffusion coefficient; ν is the reaction parameter; and f is the forcing term. Since the boundary condition plays no role in the basic construction and understanding of our upwind HDG framework, it will be ignored.

If we define $\mathbf{u} := [\sigma, u]$ we can rewrite (1) in a more compact form as

$$\nabla\cdot\mathscr{F}(\mathbf{u}) + \mathbf{C}\mathbf{u} = \mathbf{f}, \quad \text{in } \Omega, \tag{2}$$

where $\mathbf{f} := [\mathbf{0}, f]$, and $\mathbf{C}$ is a 4×4 matrix with $\mathbf{C}(1,1) = \mathbf{C}(2,2) = \mathbf{C}(3,3) = \varepsilon$, $\mathbf{C}(4,4) = \nu$ and $C(i,j) = 0$ otherwise. Here, the flux tensor $\mathscr{F}$ is given by $\mathscr{F}(\mathbf{u}) := \mathscr{A}\mathbf{u}$ and $\mathscr{A}$ is a tensor with three components defined as

$$\mathscr{A}^1 := \left[\begin{array}{ccc|c} 0&0&0&1\\ 0&0&0&0\\ 0&0&0&0\\ \hline 1&0&0&\beta^1 \end{array}\right], \quad \mathscr{A}^2 := \left[\begin{array}{ccc|c} 0&0&0&0\\ 0&0&0&1\\ 0&0&0&0\\ \hline 0&1&0&\beta^2 \end{array}\right], \text{ and } \quad \mathscr{A}^3 := \left[\begin{array}{ccc|c} 0&0&0&0\\ 0&0&0&0\\ 0&0&0&1\\ \hline 0&0&1&\beta^3 \end{array}\right].$$

Now, let $\mathbf{n} := \left[\mathbf{n}^1, \mathbf{n}^2, \mathbf{n}^3\right]$ be an arbitrary unit vector, we observe that

$$\mathbf{A} := \mathscr{A}\cdot\mathbf{n} = \left[\begin{array}{ccc|c} 0&0&0&\mathbf{n}^1\\ 0&0&0&\mathbf{n}^2\\ 0&0&0&\mathbf{n}^3\\ \hline \mathbf{n}^1&\mathbf{n}^2&\mathbf{n}^3&\beta\cdot\mathbf{n} \end{array}\right] \tag{3}$$

has four eigenvalues $[c_1, c_2, c_2, c_3]$:

$$[c_1, c_2, c_2, c_3] := \left[\frac{\beta\cdot\mathbf{n}}{2} - \frac{\sqrt{|\beta\cdot\mathbf{n}|^2+4}}{2}, 0, 0, \frac{\beta\cdot\mathbf{n}}{2} + \frac{\sqrt{|\beta\cdot\mathbf{n}|^2+4}}{2}\right].$$

It can be inspected that the eigen-values are real and eigen-vectors are independent. Consequently, (1) is a steady state hyperbolic system (see, e.g., [1] for definition of hyperbolicity), though the original convection-diffusion-reaction is not purely hyperbolic (in fact elliptic if $\beta = \mathbf{0}$). As such, it can be discretized and solved using upwind numerical methods such as DG.

The *goal* of this section is to provide a systematic construction of an upwind HDG framework for convection-diffusion-reaction equation (1). Let us begin by introducing some notations and conventions. The domain Ω is partitioned into N_{el} non-overlapping elements $K_j, j = 1,\ldots,N_{\text{el}}$ with Lipschitz boundaries such that $\Omega_h := \cup_{j=1}^{N_{\text{el}}} K_j$ and $\overline{\Omega} = \overline{\Omega}_h$. We denote the skeleton of the mesh by $\mathscr{E}_h := \cup_{j=1}^{N_{\text{el}}} \partial K_j$; it is the set of all (uniquely defined) faces e. We conventionally identify the normal vector $\mathbf{n}^-$ on the boundary ∂K of the element K under consideration (also denoted as K^-) and $\mathbf{n}^+ = -\mathbf{n}^-$ as the normal of the boundary of a neighboring element (also denoted as K^+). On the other hand, we use $\mathbf{n}$ to denote either $\mathbf{n}^-$ or $\mathbf{n}^+$ in an expression that is valid for both cases, and this convention is also used for other quantities (restricted) on $e \in \mathscr{E}_h$. For the sake of convenience, we denote by $\mathscr{E}_h^\partial$ the sets of all boundary faces and define $\mathscr{E}_h^o := \mathscr{E}_h \setminus \mathscr{E}_h^\partial$ the set of all interior faces.

For simplicity in writing we define $(\cdot,\cdot)_K$ as the L^2-inner product on a domain $K \in \mathbb{R}^d$ and $\langle\cdot,\cdot\rangle_K$ as the L^2-inner product on a domain K if $K \in \mathbb{R}^{d-1}$. We shall use bold-face lowercase/uppercase letters for vector-valued functions and in that case the inner product is defined as $(\mathbf{u},\mathbf{v})_K := \sum_{i=1}^m \left(\mathbf{u}^i, \mathbf{v}^i\right)_K$, and similarly as $\langle\mathbf{u},\mathbf{v}\rangle_K := \sum_{i=1}^m \left\langle\mathbf{u}^i, \mathbf{v}^i\right\rangle_K$, where m is the number of components ($\mathbf{u}^i, i = 1,\ldots,m$) of $\mathbf{u}$. We also employ upper case calligraphic letter, e.g. $\mathscr{F}$, to denote tensors. It is our convention

that superscripts are used to denote the components of vector, matrix, and tensor. We shall not distinguish row and column vectors in what follows.

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

$$\mathbf{V}_h(\Omega_h) := \left\{ \mathbf{v} \in \left[L^2(\Omega)\right]^m : \mathbf{v}|_K \in [\mathscr{P}^p(K)]^m, \forall K \in \Omega_h \right\},$$
$$\Lambda_h(\mathscr{E}_h) := \left\{ \lambda \in L^2(\mathscr{E}_h) : \lambda|_e \in \mathscr{P}^p(e), \forall e \in \mathscr{E}_h \right\},$$

and similarly for $\mathbf{V}_h(K)$, and $\Lambda_h(e)$ by replacing Ω_h with K and $\mathscr{E}_h$ with e. If $m = 1$, i.e. scalar-valued functions, we define

$$V_h(\Omega_h) := \left\{ v \in L^2(\Omega) : v|_K \in \mathscr{P}^p(K), \forall K \in \Omega_h \right\}.$$

From now on we conventionally use $\mathbf{u}$ for DG solution. We would like to find local finite element solution $\mathbf{u} \in \mathbf{V}_h(K)$ on each element $K \in \Omega_h$. To that end, multiplying (2) by $\mathbf{v}$ and integrating by parts we have

$$-(\mathscr{F}(\mathbf{u}), \nabla \mathbf{v})_K + \langle \mathscr{F}(\mathbf{u}) \cdot \mathbf{n}, \mathbf{v} \rangle_{\partial K} + (\mathbf{C}\mathbf{u}, \mathbf{v})_K = (\mathbf{f}, \mathbf{v})_K, \quad \forall \mathbf{v} \in \mathbf{V}_h(K). \tag{4}$$

At this point, the flux $\mathscr{F}(\mathbf{u}) \cdot \mathbf{n}$ on $e \in \partial K$ is not well-defined since the traces of both $\mathbf{u}^-$ of element K^- and $\mathbf{u}^+$ of element K^+ co-exist on e. Godunov's type methods [5] resolves this by introducing some (typically upwind, see e.g. [1, 6]) numerical flux $\mathscr{F}^*(\mathbf{u}^-, \mathbf{u}^+)$ to replace $\mathscr{F}(\mathbf{u})$ on the boundary term in (4) so that (4) becomes

$$-(\mathscr{F}(\mathbf{u}), \nabla \mathbf{v})_K + \left\langle \mathscr{F}^*\left(\mathbf{u}^-, \mathbf{u}^+\right) \cdot \mathbf{n}, \mathbf{v} \right\rangle_{\partial K} + (\mathbf{C}\mathbf{u}, \mathbf{v})_K = (\mathbf{f}, \mathbf{v})_K. \tag{5}$$

It should be pointed out that for simplicity in writing we have ignored the fact (5) must hold for all test functions $\mathbf{v} \in \mathbf{V}_h(K)$; to the end of the chapter, this should be implicitly understood.

It is the upwind numerical flux $\mathscr{F}^*$ that couples local unknowns on elements K^+ and K^- that share a face $e \in \partial K$. Consequently, local unknowns on all elements are coupled (for steady state problems or time-dependent problem with implicit time-integrators), and they must be solved together. This leads to the "usual complaint" that DG has so many coupled unknowns, and hence is expensive, though it has many attractive properties.

What we are going to do next is to remove this coupling by introducing new trace unknowns that live on the mesh skeleton. The beauty of this approach is that the actual globally coupled unknowns are those newly introduced trace unknowns, and hence the resulting system is substantially smaller and sparser. Once the trace unknowns are computed, the local DG unknown $\mathbf{u}$ is computed locally element-by-element independent of each other. *We shall show that the Rankine-Hugoniot condition (see, e.g. [1]) provides all the necessary ingredients for accomplishing this decoupling task*. To that end, let us sketch in Figure 1 the wave structure of the Riemann problem for the first order PDE system (1) along the normal direction of the interface between K^- and K^+. Here, τ *is the pseudo-time*.

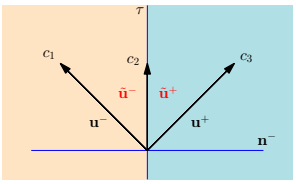

Fig. 1 The wave structure in the Riemann problem for first order form of convection-diffusion-reaction equation (1) with pseudo-time τ.

Applying the Rankine-Hugoniot condition across each wave we obtain

$$\left(\mathscr{A}^- \cdot \mathbf{n}^-\right) \tilde{\mathbf{u}}^- - \left(\mathscr{A}^- \cdot \mathbf{n}^-\right) \mathbf{u}^- = c_1 \left(\tilde{\mathbf{u}}^- - \mathbf{u}^-\right), \tag{6a}$$

$$\left(\mathscr{A}^+ \cdot \mathbf{n}^-\right) \tilde{\mathbf{u}}^+ - \left(\mathscr{A}^- \cdot \mathbf{n}^-\right) \tilde{\mathbf{u}}^- = 0, \tag{6b}$$

$$\left(\mathscr{A}^+ \cdot \mathbf{n}^-\right) \mathbf{u}^+ - \left(\mathscr{A}^+ \cdot \mathbf{n}^-\right) \tilde{\mathbf{u}}^+ = c_3 \left(\mathbf{u}^+ - \tilde{\mathbf{u}}^+\right). \tag{6c}$$

On the other hand, from definition of $\mathscr{A}$ and the continuity of β we have

$$\mathscr{A}^- = \mathscr{A}^+ = \mathscr{A}, \quad \text{and } (\mathscr{A} \cdot \mathbf{n})\,\mathbf{u} = [u\mathbf{n}, \sigma \cdot \mathbf{n} + \beta \cdot \mathbf{n} u]$$

which, together with (6b), imply

$$u^* := \tilde{u}^- = \tilde{u}^+, \quad \text{and} \quad \sigma^* \cdot \mathbf{n} := \tilde{\sigma}^- \cdot \mathbf{n} = \tilde{\sigma}^+ \cdot \mathbf{n}$$

where $\mathbf{u}^* := [\sigma^*, u^*]$ is defined as the upwind state, which is also the Riemann solution in this case (see (10a) and (10b)). The upwind flux is then defined as

$$\mathscr{F}^* \cdot \mathbf{n} := (\mathscr{A} \cdot \mathbf{n})\,\mathbf{u}^*.$$

Using the definition of c_1, c_3, and $\mathscr{A}$, we can rewrite both (6a) and (6c) in a general form, referring to either K^- or K^+, as

$$\mathscr{F}^* \cdot \mathbf{n} = \begin{bmatrix} u\mathbf{n} + \frac{1}{2}(\alpha - \beta \cdot \mathbf{n})(\sigma - \sigma^*) \\ \beta \cdot \mathbf{n} u + \sigma \cdot \mathbf{n} + \frac{1}{2}(\alpha - \beta \cdot \mathbf{n})(u - u^*) \end{bmatrix}, \tag{7}$$

with α given by $\alpha := \sqrt{|\beta \cdot \mathbf{n}|^2 + 4}$. Since the first three components of left hand sides of (7) is a vector parallel to $\mathbf{n}$, the tangent component of the corresponding

vector consisting of the first three components of right hand side must vanish. This observation allows us to rewrite the Rankine-Hugoniot conditions (7) as

$$\mathscr{F}^* \cdot \mathbf{n} = \begin{bmatrix} u\mathbf{n} + \frac{1}{2}(\alpha - \beta \cdot \mathbf{n})(\sigma - \sigma^*) \cdot \mathbf{n}\,\mathbf{n} \\ \beta \cdot \mathbf{n} u + \sigma \cdot \mathbf{n} + \frac{1}{2}(\alpha - \beta \cdot \mathbf{n})(u - u^*) \end{bmatrix}. \tag{8}$$

Since $\mathscr{F}^* \cdot \mathbf{n}$ is the upwind flux, it obviously satisfies

$$[\![\mathscr{F}^* \cdot \mathbf{n}]\!] = \mathbf{0}, \tag{9}$$

where we have defined the "jump" operator as $[\![(\cdot)]\!] := (\cdot)^- + (\cdot)^+$. We also define "average" operator as $\{\{(\cdot)\}\} := \frac{1}{2}[\![(\cdot)]\!]$.

Lemma 1. *The following hold true:*

i) The upwind state $\mathbf{u}^*$ *satisfies*

$$u^* = \{\{u\}\} + \frac{\beta \cdot \mathbf{n}}{2\alpha}[\![u\mathbf{n}]\!] \cdot \mathbf{n} + \frac{1}{\alpha}[\![\sigma \cdot \mathbf{n}]\!], \tag{10a}$$

$$\sigma^* \cdot \mathbf{n} = \{\{\sigma\}\} \cdot \mathbf{n} + \frac{1}{\alpha}[\![u\mathbf{n}]\!] \cdot \mathbf{n} - \frac{\beta \cdot \mathbf{n}}{2\alpha}[\![\sigma \cdot \mathbf{n}]\!], \tag{10b}$$

ii) The upwind flux is given by

$$\mathscr{F}^* \cdot \mathbf{n} = \left[u^*\mathbf{n}_1,\, u^*\mathbf{n}_2,\, u^*\mathbf{n}_3,\, \beta \cdot \mathbf{n} u + \sigma \cdot \mathbf{n} + \frac{1}{2}(\alpha - \beta \cdot \mathbf{n})(u - u^*) \right], \tag{11}$$

where

$$u^* = u + \frac{2}{\alpha}(\sigma - \sigma^*) \cdot \mathbf{n} + \frac{\beta \cdot \mathbf{n}}{\alpha}(u - u^*). \tag{12}$$

Proof. We know that the conservation (9) gives us four equations for the upwind state $\mathbf{u}^*$. Solving for u^* and $\sigma^* \cdot \mathbf{n}$ in terms of u and σ we obtain the desired result. The second assertion immediately follows by substituting (10) into (8) and inspecting that (12) is true.

Up to this point, we have used the exact upwind state $\mathbf{u}^*$ and the upwind flux $\mathscr{F}^* \cdot \mathbf{n}$ to derive identities in Lemma 1. In particular, we have shown that the upwind flux of the form (11) naturally arises from the Rankine-Hugoniot condition. *The appealing feature of this form is that the upwind flux depends on the DG unknowns of only one side of a face $e \in \partial K$ and the single-valued upwind state u^*.* As such, it is completely determined using only information from either side (K^- or K^+) 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 equation (5) for $\mathbf{u}$ element-by-element independent of each other. This observation suggests that we should treat u^* as the extra unknown and solve for it on the skeleton of the mesh instead of using the upwind value which couples the local unknown $\mathbf{u}$ on elements. To signify this step, let us rename u^* to $\hat{u}$ and $\mathscr{F}^*$ to $\hat{\mathscr{F}}$, i.e.,

$$\hat{\mathscr{F}}\cdot\mathbf{n}=\left[\hat{u}\mathbf{n}_1,\quad \hat{u}\mathbf{n}_2,\quad \hat{u}\mathbf{n}_3,\quad \beta\cdot\mathbf{n}u+\sigma\cdot\mathbf{n}+\frac{1}{2}\left(\alpha-\beta\cdot\mathbf{n}\right)\left(u-\hat{u}\right)\right], \tag{13}$$

where $\hat{u}$ is the single-valued trace unknown on the mesh skeleton that needs to be solve for.

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 ∂K so that (5) 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 ∂K. Clearly, at this point $\hat{u}$ is not the upwind state and hence identity (9) is in general no longer satisfied for $\hat{\mathscr{F}}$. It is therefore natural to use (9) as the extra equation. This additional algebraic equation ensures that what coming out from element K through its boundary ∂K must enter the neighboring elements that share (part of) the boundary ∂K. This is the statement of conservation and it is exactly conveyed by (9). Due to the single-valued nature of $\hat{u}$, the first three components of our HDG flux (13) automatically satisfy the conservation condition (9). For the fourth one, enforcing (9) weakly is sufficient for local conservation, i.e., $\forall e\in\mathscr{E}_h^o$:

$$\left\langle [\![\beta\cdot\mathbf{n}u+\sigma\cdot\mathbf{n}+\frac{1}{2}\left(\alpha-\beta\cdot\mathbf{n}\right)\left(u-\hat{u}\right)]\!],\mu\right\rangle_e=0,\quad \forall\mu\in\Lambda_h(e). \tag{14}$$

In summary, *we define an upwind HDG method by hybridizing the upwind flux of the standard DG scheme*. In particular, it has the usual DG local unknown $\mathbf{u}$ and the extra "trace" unknown $\hat{u}$. These unknowns can be solver for using the global conservation constraint (14) and the local solver (5) with $\mathscr{F}^*$ replaced by $\hat{\mathscr{F}}$.

We now generalize our upwind HDG approach to a class of penalty HDG schemes, a member of which is the upwind HDG itself. To that end, we first observe that $u-\hat{u}$ is the mismatch between the volume unknown restricted on the mesh skeleton and trace unknown. This mismatch vanishes for the exact solution, but converges to zero for the HDG solution as the mesh (or solution order) is refined. This suggests that one can control the mismatch by introducing a penalty parameter λ to form a penalized family of HDG fluxes as follows

$$\hat{\mathscr{F}}\cdot\mathbf{n}=\left[\hat{u}\mathbf{n}_1,\quad \hat{u}\mathbf{n}_2,\quad \hat{u}\mathbf{n}_3,\quad \beta\cdot\mathbf{n}u+\sigma\cdot\mathbf{n}+\lambda\left(u-\hat{u}\right)\right]. \tag{15}$$

Clearly, when $\lambda=\frac{1}{2}\left(\alpha-\beta\cdot\mathbf{n}\right)$ we recover the proposed upwind HDG scheme.

Next, we discuss the relation of our penalty HDG family, and hence upwind HDG, with other existing HDG ones. It is necessary brief since a more detailed discussion can be found in our previous work [7]. To begin, we observe that, for general convection-diffusion-reaction problem (and similarly for pure convection problem), if we replace λ by $\lambda-\beta\cdot\mathbf{n}$ in the HDG flux (15), we obtain

$$\hat{\mathscr{F}}\cdot\mathbf{n}=\left[\hat{u}\mathbf{n}_1,\quad \hat{u}\mathbf{n}_2,\quad \hat{u}\mathbf{n}_3,\quad \beta\cdot\mathbf{n}\hat{u}+\sigma\cdot\mathbf{n}+\lambda\left(u-\hat{u}\right)\right].$$

This is exactly the HDG scheme proposed in [8].

For the Poisson equation, our penalty HDG flux (15) simplifies to

$$\hat{\mathscr{F}} \cdot \mathbf{n} = \left[\hat{u}\mathbf{n}_1, \quad \hat{u}\mathbf{n}_2, \quad \hat{u}\mathbf{n}_3, \quad \boldsymbol{\sigma} \cdot \mathbf{n} + \lambda\,(u - \hat{u})\right], \tag{16}$$

which is exactly the HDG method originally proposed in [9]. It is important to point out that since the differential part of the Helmholtz equation is the same as that of the Poisson equation, HDG flux for the Helmholtz equation using our framework is identical to (16). That is, we have also recovered the HDG scheme for Helmholtz equation proposed in [10]. Finally, we refer to [7, 9, 8, 10, 12, 13] for a rigorous analysis of all HDG methods presented in this chapter.

3 Conclusions

We have presented a constructive methodology to derive HDG methods for convection-diffusion-reaction equation. In particular, we have shown that the Rankine-Hugoniot condition, in its primitive form, is already a hybridization of the upwind flux. The chief idea is to first break the uniqueness of the upwind flux across element boundaries by introducing single-valued trace unknowns on the mesh skeleton, and then re-enforce the uniqueness via algebraic conservation constraints. We have devised in details the construction of our upwind HDG method and extended it to a family of penalty HDG schemes. The proposed framework allows one to rediscover many existing HDG methods. Ongoing work is to apply the proposed framework to constructively derive HDG methods for other PDEs.

References

1. Toro, E. F.: Riemann Solvers and Numerical Methods for Fluid Dynamics, Springer (1999)
2. Reed, W. H. and Hill, T. R.: Triangular mesh methods for the Neutron Transport Equation, Los Alamos Scientific Laboratory (1973)
3. LeSaint, P. and Raviart, P. A.: On a Finite Element Method for Solving the Neutron Transport Equation. In Mathematical Aspects of Finite Element Methods in Partial Differential Equations, Academic Press (1974)
4. Johnson, C. and Pitkäranta, J.: An Analysis of the Discontinuous Galerkin Method for a Scalar Hyperbolic Equation, Mathematics of Computation, 1–26 (1986)
5. Godunov, S. K.: A finite difference method for the computation of discontinuous solutions of the equations of fluid dynamics, Mat. Sb., 357–393 (1959)
6. LeVeque, R. J.: Finite Volume Methods for Hyperbolic Problems, Cambridge University Press (2002)
7. Bui-Thanh, T.: From Godunov to a Unified Hybridized Discontinuous Galerkin Framework for Partial Differential Equations, Submitted (2014)
8. N. C. Nguyen and J. Peraire and B. Cockburn: An implicit high-order hybridizable discontinuous Galerkin method for linear convection-diffusion equations, Journal Computational Physics, 3232–3254 (2009)
9. Bernardo Cockburn and Jay Gopalakrishnan and Raytcho Lazarov: Unified Hybridization of Discontinuous Galerkin, Mixed, and Continuous Galerkin Methods for Second Order Elliptic Problem, SIAM J. Numer. Ana., 1319-1365 (2009)
10. R. Griesmaier and P. Monk: Error analysis for a hybridizable discontinous Galerkin method for the Helmholtz equation, J. Sci. Comput., 291–310 (2011)

11. Bernardo Cockburn and George E. Karniadakis and Chi-Wang Shu: Discontinuous Galerkin Methods: Theory, Computation and Applications. Lecture Notes in Computational Science and Engineering, Vol. 11, Springer (2000)
12. Bernardo Cockburn and Jay Gopalakrishnan and Francisco-Javier Sayas: A projection-based error analysis of HDG methods, Mathematics Of Computation, 1351-136 (2010)
13. B. Cockburn and B. Dong and J. Guzman and M. Restelli and R. Sacco: A hybridizable discontinuous Galerkin method for steady state convection-diffusion-reaction problems, SIAM J. Sci. Comput., 3827–3846 (2009)