

# An implicit high-order hybridizable discontinuous Galerkin method for nonlinear convection–diffusion equations

N.C. Nguyen<sup>a,\*</sup>, J. Peraire<sup>a</sup>, B. Cockburn<sup>b</sup>

<sup>a</sup> Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

<sup>b</sup> School of Mathematics, University of Minnesota, Minneapolis, MN 55455, USA

## ARTICLE INFO

### Article history:

Received 7 January 2009

Received in revised form 9 August 2009

Accepted 28 August 2009

Available online 6 September 2009

### Keywords:

Finite element methods

Discontinuous Galerkin methods

Hybrid/mixed methods

Nonlinear convection–diffusion equations

## ABSTRACT

In this paper, we present hybridizable discontinuous Galerkin methods for the numerical solution of steady and time-dependent nonlinear convection–diffusion equations. The methods are devised by expressing the approximate scalar variable and corresponding flux in terms of an approximate trace of the scalar variable and then explicitly enforcing the jump condition of the numerical fluxes across the element boundary. Applying the Newton–Raphson procedure and the hybridization technique, we obtain a global equation system solely in terms of the approximate trace of the scalar variable at every Newton iteration. The high number of globally coupled degrees of freedom in the discontinuous Galerkin approximation is therefore significantly reduced. We then extend the method to time-dependent problems by approximating the time derivative by means of backward difference formulae. When the time-marching method is  $(p + 1)$ th order accurate and when polynomials of degree  $p \geq 0$  are used to represent the scalar variable, each component of the flux and the approximate trace, we observe that the approximations for the scalar variable and the flux converge with the optimal order of  $p + 1$  in the  $L^2$ -norm. Finally, we apply element-by-element postprocessing schemes to obtain new approximations of the flux and the scalar variable. The new approximate flux, which has a continuous inter-element normal component, is shown to converge with order  $p + 1$  in the  $L^2$ -norm. The new approximate scalar variable is shown to converge with order  $p + 2$  in the  $L^2$ -norm. The postprocessing is performed at the element level and is thus much less expensive than the solution procedure. For the time-dependent case, the postprocessing does not need to be applied at each time step but only at the times for which an enhanced solution is required. Extensive numerical results are provided to demonstrate the performance of the present method.

© 2009 Elsevier Inc. All rights reserved.

## 1. Introduction

This paper is a continuation of our previous work [13] on hybridizable discontinuous Galerkin (HDG) methods for steady-state and time-dependent convection–diffusion equations. The HDG method presented in [13] is in turn an extension to convection–diffusion problems of the class of HDG methods first introduced in [8] for symmetric second-order elliptic problems. This method was then analyzed in [5,9] for second-order elliptic problems and further developed in [6] for convection–diffusion–reaction problems. However, none of the earlier work considers time-dependent and nonlinear problems. Indeed, extension to the time-dependent case was first considered in [13]; therein, however, only linear convection–diffusion

\* Corresponding author. Tel.: +1 617 253 8080.

E-mail address: [cuongng@mit.edu](mailto:cuongng@mit.edu) (N.C. Nguyen).

problems are addressed. As a stepping stone towards the more challenging convection-dominated and purely convective cases, we focus here on the extension of HDG methods presented in [13] to steady-state and time-dependent nonlinear convection–diffusion equations with *smooth* solution.

Other DG methods have been developed and used to solve nonlinear convection–diffusion equations. Examples include the Baumann–Oden method [4], the Bassi–Rebay schemes [2], the local discontinuous Galerkin (LDG) methods [11], and the compact discontinuous Galerkin (CDG) method [14,15]. The unified analysis of several discontinuous Galerkin (DG) methods for elliptic problems is presented in [1]. None of the DG methods considered in the unified framework [1] belongs to the class of HDG methods; for all of them, the convergence rate of the approximate flux variable is only suboptimal. In [5,9] it was proven that the approximate flux provided by the HDG method converges with optimal order for diffusion problems. The optimal convergence of the approximate flux variable was then exploited to construct a new approximation of the scalar variable which was found to converge at a faster rate than the original approximate scalar variable. In [13], extensive numerical evidence was presented indicating that all these convergence properties remain unchanged for linear convection–diffusion problems and even in the time-dependent case.

As it is typical of HDG methods, to carry out the discretization in space, we proceed in two main steps. First, we express the approximate scalar variable and corresponding flux within each element in terms of an approximate trace of the scalar variable and enforce the continuity of the normal component of the numerical fluxes across the element boundary. Then, by application of the hybridization technique, we obtain a global equation system solely in terms of the degrees of freedom of the approximate trace. The HDG methods have some favorable features. First, the final matrix system to be inverted has a significantly smaller number of globally coupled degrees of freedom relative to more standard discontinuous Galerkin methods analyzed in [1]. Moreover, the methods possess a compact discretization which eliminates the non-compact nature of the LDG methods in multi-dimensions [14]. Another advantage is that for linear elliptic problems [5,9] and linear convection–diffusion problems [6,13] the approximate scalar variable and the approximate flux converge optimally at a rate of  $p + 1$  in  $L^2$ -norm when polynomials of order  $p$  are used to represent the approximate variables. In addition, they provide optimally convergent approximations even for the case  $p = 0$ .

In this paper, we first present the HDG methods and describe their implementation for steady nonlinear convection–diffusion equations. We propose a general formula for defining the numerical fluxes which are crucial to maintaining the stability and accuracy of the numerical solution. We then make the extension to time-dependent nonlinear convection–diffusion equations by employing backward difference formulae for the discretization of the time derivative. The resulting method is implicit, stable and high-order accurate and involves significantly less degrees of freedom than standard implicit DG methods [15]. We present numerical results for two particular choices of numerical fluxes indicating that the HDG methods developed here exhibit the above-mentioned convergence properties (observed for the linear convection–diffusion problems) for both steady-state and time-dependent nonlinear convection–diffusion equations. Even though here we only consider backward difference formulae for the time discretization, the HDG methods described can also work with other implicit time-stepping methods such as the fully implicit Runge–Kutta methods and DG methods in time.

Finally, we use a simple element-by-element postprocessing scheme to obtain new approximations of the flux and the scalar variable. The new approximate flux, which has a continuous interelement normal component, is shown to converge with order  $p + 1$  in the  $L^2$ -norm. The new approximate scalar variable is shown to converge with order  $p + 2$  in the  $L^2$ -norm. Since the postprocessing involves solving a linear elliptic partial differential equation (a Poisson problem) at the element level, the new approximations are much less expensive to compute than the original approximate solution. The proposed postprocessing scheme is closely related to the one introduced in [16,17]; it is an extension of the one used in [13], which in turn is an extension of the postprocessing schemes developed in [5,9]; see also the references therein. Let us stress the fact that, unlike the cases considered in [5,9] our local postprocessing scheme does *not* involve the original partial differential equation. It is thus particularly well-suited for both time-dependent and nonlinear problems. Moreover, it does not have to be applied at each timestep, but only at desired times during the simulation.

The paper is organized as follows. We describe the HDG-space discretization methods for the steady-state case in Section 2 and then extend the methods to the time-dependent case in Section 3. In Section 4, we present the local postprocessing procedures to compute a higher-order accurate solution. In Section 5, we provide extensive numerical results to assess the convergence and accuracy of the method. Finally, we end with some concluding remarks in Section 6.

## 2. The hybridizable discontinuous Galerkin methods

### 2.1. Problem statement and notation

In this section, we present HDG methods for steady-state nonlinear convection–diffusion equations of the form

$$\begin{aligned} -\nabla \cdot (\kappa \nabla u) + \nabla \cdot \mathbf{F}(u) &= f, & \text{in } \Omega, \\ u &= g_D, & \text{on } \partial\Omega. \end{aligned} \tag{1}$$

As it is typical of DG discretizations, we introduce a diffusive flux  $\mathbf{q} = -\kappa \nabla u$  and rewrite the above equation as a first-order system of equations

$$\begin{aligned} \mathbf{q} + \kappa \nabla u &= \mathbf{0}, & \text{in } \Omega, \\ \nabla \cdot (\mathbf{q} + \mathbf{F}(u)) &= f, & \text{in } \Omega, \\ u &= g_D, & \text{on } \partial\Omega. \end{aligned} \tag{2}$$

Here  $\Omega \in \mathbb{R}^d$  is the physical domain with Lipschitz boundary  $\partial\Omega$ ,  $f \in L^2(\Omega)$  is a prescribed source term,  $\kappa \in L^\infty(\Omega)$  is a positive diffusion coefficient, and  $\mathbf{F} \in (L^\infty(\Omega))^d$  are vector-valued nonlinear functions of the scalar variable  $u$ .

We first need to introduce some notation. We denote by  $\mathcal{T}_h$  a collection of disjoint regular elements  $K$  that partition  $\Omega$  and set  $\partial\mathcal{T}_h := \{\partial K : K \in \mathcal{T}_h\}$ . For an element  $K$  of the collection  $\mathcal{T}_h$ ,  $e = \partial K \cap \partial\Omega$  is the boundary face if the  $(d - 1)$ -Lebesgue measure of  $e$  is nonzero. For two elements  $K^+$  and  $K^-$  of the collection  $\mathcal{T}_h$ ,  $e = \partial K^+ \cap \partial K^-$  is the interior face between  $K^+$  and  $K^-$  if the  $(d - 1)$ -Lebesgue measure of  $e$  is nonzero. Let  $\mathcal{E}_h^o$  and  $\mathcal{E}_h^b$  denote the set of interior and boundary faces, respectively. We denote by  $\mathcal{E}_h$  the union of  $\mathcal{E}_h^o$  and  $\mathcal{E}_h^b$ .

Let  $\mathbf{n}^+$  and  $\mathbf{n}^-$  be the outward unit normals of  $\partial K^+$  and  $\partial K^-$ , respectively, and let  $(\mathbf{v}^\pm, w^\pm)$  be the traces of  $(\mathbf{v}, w)$  on  $e$  from the interior of  $K^\pm$ . Then, we define the average  $\{\{\cdot\}\}$  and the jump  $[[\cdot]]$  as follows. For  $e \in \mathcal{E}_h^o$ , we set

$$\begin{aligned} \{\{\mathbf{q}\}\} &= (\mathbf{q}^+ + \mathbf{q}^-)/2 & \{\{u\}\} &= (u^+ + u^-)/2, \\ [[\mathbf{q} \cdot \mathbf{n}]] &= \mathbf{q}^+ \cdot \mathbf{n}^+ + \mathbf{q}^- \cdot \mathbf{n}^- & [[u]] &= u^+ \mathbf{n}^+ + u^- \mathbf{n}^-. \end{aligned}$$

For  $e \in \mathcal{E}_h^b$ , the set of boundary edges on which  $\mathbf{q}$  and  $u$  are singled value, we set

$$\begin{aligned} \{\{\mathbf{q}\}\} &= \mathbf{q} & \{\{u\}\} &= u, \\ [[\mathbf{q} \cdot \mathbf{n}]] &= \mathbf{q} \cdot \mathbf{n} & [[u]] &= u, \end{aligned}$$

Note that the jump in  $u$  is a vector, but the jump in  $\mathbf{q}$  is a scalar which only involves the normal component of  $\mathbf{q}$ . Furthermore, the jump will be zero for a continuous function.

Now, let  $\mathcal{P}^p(D)$  denote the set of polynomials of degree at most  $p$  on a domain  $D$  and let  $L^2(D)$  be the space of square integrable functions on  $D$ . For any element  $K$  of the collection  $\mathcal{T}_h$  we denote  $W^p(K) \equiv \mathcal{P}^p(K)$  and  $\mathbf{V}^p(K) \equiv (\mathcal{P}^p(K))^d$ . We introduce discontinuous finite element spaces

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

We also set  $M_h^p(g_D) = \{\mu \in M_h^p : \mu = \text{Pg}_D \text{ on } \Gamma_D\}$ , where  $\text{P}$  denotes the  $L^2$ -projection into the space  $\{\mu|_{\partial\Omega} \mid \forall \mu \in M_h^p\}$ . Note that  $M_h^p$  consists of functions which are continuous inside the faces (or edges)  $e \in \mathcal{E}_h$  and discontinuous at their borders.

For functions  $u$  and  $v$  in  $L^2(D)$ , we denote  $(u, v)_D = \int_D uv$  if  $D$  is a domain in  $\mathbb{R}^d$  and  $\langle u, v \rangle_D = \int_D uv$  if  $D$  is a domain in  $\mathbb{R}^{d-1}$ . We finally introduce

$$(w, v)_{\mathcal{T}_h} = \sum_{K \in \mathcal{T}_h} (w, v)_K, \quad \langle \zeta, \rho \rangle_{\partial\mathcal{T}_h} = \sum_{K \in \mathcal{T}_h} \langle \zeta, \rho \rangle_{\partial K}, \quad \langle \mu, \eta \rangle_{\mathcal{E}_h} = \sum_{e \in \mathcal{E}_h} \langle \mu, \eta \rangle_e,$$

for functions  $w, v$  defined on  $\Omega$ , functions  $\zeta, \rho$  defined on  $\partial\mathcal{T}_h$ , and functions  $\mu, \eta$  defined on  $\mathcal{E}_h$ . Similar notation can be used to define volume and boundary inner products for any pair of vector-valued functions.

### 2.2. The HDG formulation

Multiplying the first two equations of (2) by test functions and integrating by parts, we arrive at the following formulation for determining an approximate solution  $(\mathbf{q}_h, u_h) \in \mathbf{V}_h^p \times W_h^p$  such that for all  $K \in \mathcal{T}_h$ ,

$$(\kappa^{-1} \mathbf{q}_h, \mathbf{v})_K - (u_h, \nabla \cdot \mathbf{v})_K + \langle \hat{u}_h, \mathbf{v} \cdot \mathbf{n} \rangle_{\partial K} = 0, \quad -(\mathbf{q}_h + \mathbf{F}(u_h), \nabla w)_K + \langle (\hat{\mathbf{q}}_h + \hat{\mathbf{F}}_h) \cdot \mathbf{n}, w \rangle_{\partial K} = (f, w)_K, \tag{3}$$

for all  $(\mathbf{v}, w) \in \mathbf{V}^p(K) \times W^p(K)$ . Here the numerical traces  $\hat{\mathbf{q}}_h, \hat{\mathbf{F}}_h$ , and  $\hat{u}_h$  are approximations to  $-\kappa \nabla u, \mathbf{F}(u)$ , and  $u$  over  $\mathcal{E}_h$ , respectively. By adding the contributions of (3) over all the elements and enforcing the continuity of the normal component of the total numerical flux, we obtain the following problem (see [13]): find an approximation  $(\mathbf{q}_h, u_h, \hat{u}_h) \in \mathbf{V}_h^p \times W_h^p \times M_h^p(g_D)$  such that

$$\begin{aligned} (\kappa^{-1} \mathbf{q}_h, \mathbf{v})_{\mathcal{T}_h} - (u_h, \nabla \cdot \mathbf{v})_{\mathcal{T}_h} + \langle \hat{u}_h, \mathbf{v} \cdot \mathbf{n} \rangle_{\partial\mathcal{T}_h} &= 0, \\ -(\mathbf{q}_h + \mathbf{F}(u_h), \nabla w)_{\mathcal{T}_h} + \langle (\hat{\mathbf{q}}_h + \hat{\mathbf{F}}_h) \cdot \mathbf{n}, w \rangle_{\partial\mathcal{T}_h} &= (f, w)_{\mathcal{T}_h}, \\ \langle (\hat{\mathbf{q}}_h + \hat{\mathbf{F}}_h) \cdot \mathbf{n}, \mu \rangle_{\partial\mathcal{T}_h} &= 0, \end{aligned} \tag{4}$$

for all  $(\mathbf{v}, w, \mu) \in \mathbf{V}_h^p \times W_h^p \times M_h^p(0)$ , where

$$\hat{\mathbf{q}}_h + \hat{\mathbf{F}}_h = \mathbf{q}_h + \mathbf{F}(\hat{u}_h) + \tau(u_h, \hat{u}_h)(u_h - \hat{u}_h)\mathbf{n}, \quad \text{on } \mathcal{E}_h. \tag{5}$$

This completes the definition of the HDG method.

Note that the Dirichlet boundary condition has been enforced by requiring that  $\hat{u}_h = \text{Pg}_D$  on  $\mathcal{E}_h \cap \partial\Omega$ . Note also that the choice of the numerical flux  $\hat{\mathbf{q}}_h + \hat{\mathbf{F}}_h$  is an extension of the expression for the numerical flux used for the linear case in [13]. The main difference is that, due to the nonlinearity of the convection, the *stabilization function*  $\tau(\cdot, \cdot) : \partial\mathcal{T}_h \rightarrow \mathbb{R}$  can now be a nonlinear function of  $u_h$  and  $\hat{u}_h$ . This implies that the last Eq. (4) cannot force the normal component of the total flux  $\hat{\mathbf{q}}_h + \hat{\mathbf{F}}_h$  to be single valued on all interior faces  $e \in \mathcal{E}_h^\circ$ ; it only forces the  $L^2$ -projection of the normal component of the total flux into  $M_h(0)$  to be single valued. This is enough to guarantee the local conservativity of the method, as we can see from the second term of the left-hand side of the second Eq. (4).

### 2.3. A condition on the stabilization function $\tau$

We are going to choose  $\tau$  so that the energy associated with the approximate solution is positive; this is an extension to our setting of the approach used for the linear case [13]. Thus, we begin by obtaining what could be called an energy identity. In the purely convective case, this identity gives rise to what is called an entropy inequality for the quadratic entropy  $u^2/2$ .

To do that, we need to introduce the *entropy flux* associated to the entropy, namely,

$$\mathbf{F}(u, c) := u\mathbf{F}(u) - c\mathbf{F}(c) - \int_c^u \mathbf{F}(s)ds,$$

where  $c$  is an arbitrary real number. Recall that this entropy flux is defined by requiring that  $\frac{\partial}{\partial u}\mathcal{F}(u, c) = u\mathbf{F}'(u)$ , and  $\mathcal{F}(u, u) = 0$ . Finally, to shorten the energy identity, we write

$$\hat{\mathcal{F}}_{h,c} := \mathcal{F}(\hat{u}_h, c) + \tau(u_h, \hat{u}_h)(u_h - \hat{u}_h)\hat{u}_h.$$

**Proposition 1** (Energy identity). *For any real number  $c$ , we have*

$$(\kappa^{-1}\mathbf{q}_h, \mathbf{q}_h)_{\mathcal{T}_h} + \langle \theta, (u_h - \hat{u}_h)^2 \rangle_{\partial\mathcal{T}_h} = (f, u_h)_\Omega - \langle \mathbf{q}_h \cdot \mathbf{n}, \hat{u}_h \rangle_{\partial\Omega} - \langle \hat{\mathcal{F}}_{h,c} \cdot \mathbf{n}, \mathbf{1} \rangle_{\partial\Omega},$$

where  $\theta := \tau(u_h, \hat{u}_h) - \tau_F(u_h, \hat{u}_h)$  and

$$\tau_F(u_h, \hat{u}_h) := \frac{1}{(u_h - \hat{u}_h)^2} \int_{\hat{u}_h}^{u_h} (\mathbf{F}(s) - \mathbf{F}(\hat{u}_h)) \cdot \mathbf{n} ds.$$

**Proof.** Taking  $\mathbf{v} := \mathbf{q}_h$ ,  $w := u_h$  and  $\mu|_{\mathcal{E}_h^\circ} := -\hat{u}_h$  in the Eq. (4) and adding them up, we get

$$(\kappa^{-1}\mathbf{q}_h, \mathbf{q}_h)_{\mathcal{T}_h} + \Psi_h = (f, u_h)_\Omega,$$

where

$$\Psi_h := -\langle u_h, \mathbf{q}_h \cdot \mathbf{n} \rangle_{\partial\mathcal{T}_h} + \langle \hat{u}_h, \mathbf{q}_h \cdot \mathbf{n} \rangle_{\partial\mathcal{T}_h} - (\mathbf{F}(u_h), \nabla u_h)_{\mathcal{T}_h} + \langle (\hat{\mathbf{q}}_h + \hat{\mathbf{F}}_h) \cdot \mathbf{n}, u_h \rangle_{\partial\mathcal{T}_h} - \langle (\hat{\mathbf{q}}_h + \hat{\mathbf{F}}_h) \cdot \mathbf{n}, \hat{u}_h \rangle_{\partial\mathcal{T}_h \setminus \partial\Omega}.$$

Then, rearranging terms, we get

$$\Psi_h = -(\mathbf{F}(u_h), \nabla u_h)_{\mathcal{T}_h} + \langle (\hat{\mathbf{q}}_h + \hat{\mathbf{F}}_h - \mathbf{q}_h) \cdot \mathbf{n}, u_h - \hat{u}_h \rangle_{\partial\mathcal{T}_h} + \langle (\hat{\mathbf{q}}_h + \hat{\mathbf{F}}_h) \cdot \mathbf{n}, \hat{u}_h \rangle_{\partial\Omega},$$

and, by the definition of the numerical flux  $(\hat{\mathbf{q}}_h + \hat{\mathbf{F}}_h)$ , (5),

$$\begin{aligned} \Psi_h &= -(\mathbf{F}(u_h), \nabla u_h)_{\mathcal{T}_h} + \langle \mathbf{F}(\hat{u}_h) \cdot \mathbf{n}, u_h - \hat{u}_h \rangle_{\partial\mathcal{T}_h} + \langle \tau(u_h, \hat{u}_h)(u_h - \hat{u}_h), u_h - \hat{u}_h \rangle_{\partial\mathcal{T}_h} + \langle \mathbf{q}_h \cdot \mathbf{n}, \hat{u}_h \rangle_{\partial\Omega} \\ &\quad + \langle \hat{u}_h \mathbf{F}(\hat{u}_h) \cdot \mathbf{n} + \tau(u_h, \hat{u}_h)(\hat{u}_h - u_h)\hat{u}_h, \mathbf{1} \rangle_{\partial\Omega}. \end{aligned}$$

Let  $\mathbf{G}(s)$  be such that  $d\mathbf{G}(s)/ds = \mathbf{F}(s)$ . Since

$$\begin{aligned} -(\mathbf{F}(u_h), \nabla u_h)_{\mathcal{T}_h} &= -(\nabla \cdot \mathbf{G}(u_h), \mathbf{1})_{\mathcal{T}_h} = -\langle \mathbf{G}(u_h) \cdot \mathbf{n}, \mathbf{1} \rangle_{\partial\mathcal{T}_h} + \langle \mathbf{G}(c) \cdot \mathbf{n}, \mathbf{1} \rangle_{\partial\mathcal{T}_h} = -\left\langle \int_c^{u_h} \mathbf{F}(s) \cdot \mathbf{n} ds, \mathbf{1} \right\rangle_{\partial\mathcal{T}_h}, \\ &= -\left\langle \int_{\hat{u}_h}^{u_h} \mathbf{F}(s) \cdot \mathbf{n} ds, \mathbf{1} \right\rangle_{\partial\mathcal{T}_h} - \left\langle \int_c^{\hat{u}_h} \mathbf{F}(s) \cdot \mathbf{n} ds, \mathbf{1} \right\rangle_{\partial\Omega}, \end{aligned}$$

we obtain,

$$\begin{aligned} \Psi_h &= -\left\langle \int_{\hat{u}_h}^{u_h} (\mathbf{F}(s) - \mathbf{F}(\hat{u}_h)) \cdot \mathbf{n} ds, \mathbf{1} \right\rangle_{\partial\mathcal{T}_h} - \left\langle \int_c^{\hat{u}_h} \mathbf{F}(s) \cdot \mathbf{n} ds, \mathbf{1} \right\rangle_{\partial\Omega} + \langle \tau(u_h, \hat{u}_h)(u_h - \hat{u}_h), u_h - \hat{u}_h \rangle_{\partial\mathcal{T}_h} + \langle \mathbf{q}_h \cdot \mathbf{n}, \hat{u}_h \rangle_{\partial\Omega} \\ &\quad + \langle \hat{u}_h \mathbf{F}(\hat{u}_h) \cdot \mathbf{n} + \tau(u_h, \hat{u}_h)(\hat{u}_h - u_h)\hat{u}_h, \mathbf{1} \rangle_{\partial\Omega}. \end{aligned}$$

Now, by definition of  $\theta$ ,

$$\Psi_h = \left\langle \theta, (u_h - \hat{u}_h)^2 \right\rangle_{\partial T_h} - \left\langle \int_c^{\hat{u}_h} \mathbf{F}(s) \cdot \mathbf{n} ds, 1 \right\rangle_{\partial \Omega} + \langle \mathbf{q}_h \cdot \mathbf{n}, \hat{u}_h \rangle_{\partial \Omega} + \langle \hat{u}_h \mathbf{F}(\hat{u}_h) \cdot \mathbf{n} + \tau(u_h, \hat{u}_h)(\hat{u}_h - u_h) \hat{u}_h, 1 \rangle_{\partial \Omega},$$

and by definition of  $\widehat{\mathcal{F}}_{h,c}$ ,

$$\begin{aligned} \Psi_h &= \left\langle \theta, (u_h - \hat{u}_h)^2 \right\rangle_{\partial T_h} + \langle \mathbf{q}_h \cdot \mathbf{n}, \hat{u}_h \rangle_{\partial \Omega} + \left\langle \left( \widehat{\mathcal{F}}_{h,c} + c\mathbf{F}(c) \right) \cdot \mathbf{n}, 1 \right\rangle_{\partial \Omega} \\ &= \left\langle \theta, (u_h - \hat{u}_h)^2 \right\rangle_{\partial T_h} + \langle \mathbf{q}_h \cdot \mathbf{n}, \hat{u}_h \rangle_{\partial \Omega} + \left\langle \widehat{\mathcal{F}}_{h,c} \cdot \mathbf{n}, 1 \right\rangle_{\partial \Omega}, \end{aligned}$$

since  $c$  is a constant. This completes the proof.  $\square$ .

Note that the energy identity of Proposition 1 shows that the energy associated to the discontinuities  $u_h - \hat{u}_h$  on  $\partial T_h$  is strictly positive provided

$$\tau(u_h, \hat{u}_h) > \tau_F(\hat{u}_h, u_h). \tag{6}$$

Note also that since

$$\tau_F(u_h, \hat{u}_h) = \frac{1}{(u_h - \hat{u}_h)^2} \int_{\hat{u}_h}^{u_h} \mathbf{F}'(s) \cdot \mathbf{n}(u_h - s) ds \leq \frac{1}{2} \sup_{s \in J(u_h, \hat{u}_h)} |\mathbf{F}'(s)|,$$

where  $J(u_h, \hat{u}_h) = [\min\{u_h, \hat{u}_h\}, \max\{u_h, \hat{u}_h\}]$ , the function  $\tau(u_h, \hat{u}_h)$  satisfies the positivity condition (6) if

$$\tau(u_h, \hat{u}_h) \geq \frac{1}{2} \sup_{s \in J(u_h, \hat{u}_h)} |\mathbf{F}'(s)|. \tag{7}$$

We next give other choices of  $\tau$  which satisfies the positivity condition (6).

#### 2.4. Examples of stabilization functions $\tau$

The examples we consider here are extensions of the expression proposed in [13] for the linear case. To obtain them, we begin by separating the influence of the diffusion and the convection. Thus we write

$$\tau(u_h, \hat{u}_h) = \tau_{diff} + \tau_{conv}(u_h, \hat{u}_h),$$

and set

$$\hat{\mathbf{q}}_h := \mathbf{q}_h + \tau_{diff}(u_h - \hat{u}_h)\mathbf{n}, \widehat{\mathbf{F}}_h := \mathbf{F}(\hat{u}_h) + \tau_{conv}(u_h, \hat{u}_h)\mathbf{n}.$$

We can consider that  $\tau_{diff}$  is the stabilization function associated to the diffusion and  $\tau_{conv}(u_h, \hat{u}_h)$  the stabilization function associated to the convection. Now the positivity condition (6) would be satisfied if  $\tau_{diff} > 0$  and if  $\tau_{conv}(u_h, \hat{u}_h) \geq \tau_F(u_h, \hat{u}_h)$ . Next, we give examples of such choices.

For the stabilization function associated to the diffusion, we simply take

$$\tau_{diff} = \kappa/\ell,$$

where  $\ell$  is a length associated to the problem. To define the stabilization function associated to the convection, we introduce the numerical fluxes  $\widehat{\mathbf{F}} \cdot \mathbf{n}(\cdot, \cdot)$ , associated with the so-called monotone schemes; see [12], for example.

We require that  $\widehat{\mathbf{F}} \cdot \mathbf{n}(\cdot, \cdot)$  satisfy the following properties:

$$\widehat{\mathbf{F}} \cdot \mathbf{n}(a, a) = \mathbf{F}(a) \cdot \mathbf{n}, \tag{8a}$$

$$\widehat{\mathbf{F}} \cdot \mathbf{n}(a, b) \text{ is non-decreasing in } a, \tag{8b}$$

$$\widehat{\mathbf{F}} \cdot \mathbf{n}(a, b) \text{ is non-decreasing in } b. \tag{8c}$$

The main examples are the Godunov numerical flux,

$$\widehat{\mathbf{F}} \cdot \mathbf{n}^G(a, b) = \begin{cases} \min_{s \in [a, b]} \mathbf{F}(s) \cdot \mathbf{n}, & \text{if } a \leq b, \\ \max_{s \in [b, a]} \mathbf{F}(s) \cdot \mathbf{n}, & \text{if } a > b, \end{cases} \tag{9a}$$

the Engquist–Osher numerical flux,

$$\widehat{\mathbf{F}} \cdot \mathbf{n}^{EO}(a, b) = \frac{1}{2}(\mathbf{F}(a) + \mathbf{F}(b)) \cdot \mathbf{n} - \frac{1}{2} \int_a^b |\mathbf{F}'(s) \cdot \mathbf{n}| ds, \tag{9b}$$

and the Lax–Friedrichs flux:

$$\widehat{\mathbf{F}} \cdot \widehat{\mathbf{n}}^{LF}(a, b) = \frac{1}{2}(\mathbf{F}(a) + \mathbf{F}(b)) \cdot \mathbf{n} - \frac{C}{2}(b - a), \quad (9c)$$

where  $C = C^{LF}(I) := \sup_{s \in I} |\mathbf{F}'(s)|$ . Note that this numerical flux satisfies the conditions (8b) provided  $a$  and  $b$  lie in the interval  $I$ .

So, given a numerical flux  $\widehat{\mathbf{F}} \cdot \mathbf{n}(\cdot, \cdot)$ , we can take, for example,

$$\tau_{conv}(u_h, \hat{u}_h) = \frac{1}{(u_h - \hat{u}_h)} \int_{\hat{u}_h}^{u_h} \frac{(\widehat{\mathbf{F}} \cdot \mathbf{n}(s, \hat{u}_h) - \mathbf{F}(\hat{u}_h) \cdot \mathbf{n})}{(u_h - \hat{u}_h)} ds. \quad (10a)$$

This choice gives rise to the convective numerical flux

$$\widehat{\mathbf{F}}_h \cdot \mathbf{n} = \frac{1}{u_h - \hat{u}_h} \int_{\hat{u}_h}^{u_h} \widehat{\mathbf{F}} \cdot \mathbf{n}(s, \hat{u}_h) ds, \quad (10b)$$

which satisfies the positivity condition (6). Indeed, we have

$$\tau_{conv}(u_h, \hat{u}_h) \geq \frac{1}{(u_h - \hat{u}_h)^2} \int_{\hat{u}_h}^{u_h} (\widehat{\mathbf{F}} \cdot \mathbf{n}(s, s) - \mathbf{F}(\hat{u}_h) \cdot \mathbf{n}) ds$$

by the monotonicity property (8c), and so  $\tau_{conv}(u_h, \hat{u}_h) \geq \tau_{\mathbf{F}}(u_h, \hat{u}_h)$ , by the consistency property (8a).

### 2.5. The general form of the numerical traces

Next we show that the HDG methods just introduced can be thought of a conservative and consistent DG method with some appropriate choices of the numerical traces. Indeed, if the stabilization function  $\tau$  is constant on each interior face, by the last equation in (4), we have that

$$\llbracket (\widehat{\mathbf{F}}_h + \widehat{\mathbf{q}}_h) \cdot \mathbf{n} \rrbracket = 0 \quad \text{on } \mathcal{E}_h^o.$$

Inserting the expression of the numerical flux  $(\widehat{\mathbf{F}}_h + \widehat{\mathbf{q}}_h)$ , (5), we obtain

$$\llbracket \mathbf{q}_h \cdot \mathbf{n} \rrbracket + \tau^+ u_h^+ + \tau^- u_h^- - (\tau^+ + \tau^-) \hat{u}_h = 0 \quad \text{on } \mathcal{E}_h^o,$$

since  $\llbracket \mathbf{F}(\hat{u}_h) \cdot \mathbf{n} \rrbracket = 0$ . Finally, solving for  $\hat{u}_h$  and inserting the result into the expression of the numerical flux  $(\widehat{\mathbf{F}}_h + \widehat{\mathbf{q}}_h)$ , (5), we obtain on  $\mathcal{E}_h^o$

$$\begin{aligned} \hat{u}_h &= \frac{\tau^+}{\tau^+ + \tau^-} u_h^+ + \frac{\tau^-}{\tau^+ + \tau^-} u_h^- + \left( \frac{1}{\tau^+ + \tau^-} \right) \llbracket \mathbf{q}_h \cdot \mathbf{n} \rrbracket, \\ \widehat{\mathbf{F}}_h + \widehat{\mathbf{q}}_h &= \mathbf{F}(\hat{u}_h) + \frac{\tau^-}{\tau^+ + \tau^-} \mathbf{q}_h^+ + \frac{\tau^+}{\tau^+ + \tau^-} \mathbf{q}_h^- + \left( \frac{\tau^+ \tau^-}{\tau^+ + \tau^-} \right) \llbracket u_h \mathbf{n} \rrbracket. \end{aligned} \quad (11)$$

On the set of boundary faces,  $\mathcal{E}_h^g$ , the Dirichlet condition  $u_h \in M_h(g_D)$  and the expression of the numerical flux  $(\widehat{\mathbf{F}}_h + \widehat{\mathbf{q}}_h)$ , (5), give

$$\hat{u}_h = Pg_D, \quad \widehat{\mathbf{F}}_h + \widehat{\mathbf{q}}_h = \mathbf{F}(Pg_D) + \mathbf{q}_h + \tau(u_h - Pg_D). \quad (12)$$

Note that our numerical fluxes are closely related to the numerical fluxes of the LDG methods [11]; the main difference lies in the fact that in the LDG methods  $\hat{u}_h$  does not depend on  $\mathbf{q}_h$ .

Note also that if we take  $\tau = \tau^+ = \tau^- := \kappa/\ell + C/2$ , where  $C$  is associated to the modified Lax–Friedrichs flux, then the numerical traces are

$$\hat{u}_h = \{\{u_h\}\} + \frac{1}{2\tau} \llbracket \mathbf{q}_h \cdot \mathbf{n} \rrbracket, \quad \widehat{\mathbf{F}}_h + \widehat{\mathbf{q}}_h = \mathbf{F}(\hat{u}_h) + \{\{\mathbf{q}_h\}\} + \frac{\tau}{2} \llbracket u_h \mathbf{n} \rrbracket. \quad (13)$$

This is the *centered scheme* proposed and analyzed in [13] for linear convection–diffusion problems.

### 2.6. Implementation

To implement the method, we first insert (5) into (4) and obtain, after some simple manipulations, that  $(\mathbf{q}_h, u_h, \hat{u}_h) \in \mathbf{V}_h^p \times W_h^p \times M_h^p(g_D)$  is the solution of the following system

$$\begin{aligned} (\kappa^{-1} \mathbf{q}_h, \mathbf{v})_{T_h} - (u_h, \nabla \cdot \mathbf{v})_{T_h} + \langle \hat{u}_h, \mathbf{v} \cdot \mathbf{n} \rangle_{\partial T_h} &= 0, \\ (\nabla \cdot \mathbf{q}_h, \mathbf{w})_{T_h} - (\mathbf{F}(u_h), \nabla \mathbf{w})_{T_h} + \langle \mathbf{F}(\hat{u}_h) \cdot \mathbf{n} + \tau(u_h, \hat{u}_h)(u_h - \hat{u}_h), \mathbf{w} \rangle_{\partial T_h} &= (f, \mathbf{w})_{T_h}, \\ \langle (\mathbf{q}_h + \mathbf{F}(\hat{u}_h)) \cdot \mathbf{n} + \tau(u_h, \hat{u}_h)(u_h - \hat{u}_h), \mu \rangle_{\partial T_h} &= 0, \end{aligned}$$

for all  $(\mathbf{v}, w, \mu) \in \mathbf{V}_h^p \times W_h^p \times M_h^p(0)$ . Next, we apply the Newton–Raphson method to solve the above system. Denoting the current iterate by  $(\bar{\mathbf{q}}_h, \bar{u}_h, \bar{u}_h) \in \mathbf{V}_h^p \times W_h^p \times M_h^p(g_D)$ , we then find an increment  $(\delta\mathbf{q}_h, \delta u_h, \delta\hat{u}_h) \in \mathbf{V}_h^p \times W_h^p \times M_h^p(0)$  such that

$$\begin{aligned} a(\delta\mathbf{q}_h, \mathbf{v}) - b(\delta u_h, \mathbf{v}) + c(\delta\hat{u}_h, \mathbf{v}) &= r(\mathbf{v}), \\ b(w, \delta\mathbf{q}_h) + d(\delta u_h, w) + e(\delta\hat{u}_h, w) &= f(w), \\ c(\mu, \delta\mathbf{q}_h) + g(\mu, \delta u_h) + h(\mu, \delta\hat{u}_h) &= \ell(\mu), \end{aligned} \tag{14}$$

for all  $(\mathbf{v}, w, \mu) \in \mathbf{V}_h^p \times W_h^p \times M_h^p(0)$ . Here, the forms are given by

$$\begin{aligned} a(\boldsymbol{\sigma}, \mathbf{v}) &= (\kappa^{-1}\boldsymbol{\sigma}, \mathbf{v})_{\mathcal{T}_h}, \\ b(\zeta, \mathbf{v}) &= (\zeta, \nabla \cdot \mathbf{v})_{\mathcal{T}_h}, \\ c(\rho, \mathbf{v}) &= \langle \rho, \mathbf{v} \cdot \mathbf{n} \rangle_{\partial\mathcal{T}_h}, \\ d(\zeta, w) &= -(\mathbf{F}'(\bar{u}_h)\zeta, \nabla w)_{\mathcal{T}_h} + \left\langle \left( \partial_1 \tau(\bar{u}_h, \bar{u}_h)(\bar{u}_h - \bar{u}_h) + \tau(\bar{u}_h, \bar{u}_h) \right) \zeta, w \right\rangle_{\partial\mathcal{T}_h}, \\ e(\rho, w) &= \left\langle \left( \mathbf{F}'(\bar{u}_h) \cdot \mathbf{n} + \partial_2 \tau(\bar{u}_h, \bar{u}_h)(\bar{u}_h - \bar{u}_h) - \tau(\bar{u}_h, \bar{u}_h) \right) \rho, w \right\rangle_{\partial\mathcal{T}_h}, \\ g(\mu, \zeta) &= \left\langle \left( \partial_1 \tau(\bar{u}_h, \bar{u}_h)(\bar{u}_h - \bar{u}_h) + \tau(\bar{u}_h, \bar{u}_h) \right) \zeta, \mu \right\rangle_{\partial\mathcal{T}_h}, \\ h(\mu, \rho) &= \langle \mathbf{F}'(\bar{u}_h) \cdot \mathbf{n} + \partial_2 \tau(\bar{u}_h, \bar{u}_h)(\bar{u}_h - \bar{u}_h) - \tau(\bar{u}_h, \bar{u}_h) \rangle \rho, \mu \rangle_{\partial\mathcal{T}_h}, \\ f(w) &= (f, w)_{\mathcal{T}_h} - (\nabla \cdot \bar{\mathbf{q}}_h, w)_{\mathcal{T}_h} + (\mathbf{F}(\bar{u}_h), \nabla w)_{\mathcal{T}_h} - \langle \mathbf{F}(\bar{u}_h) \cdot \mathbf{n} + \tau(\bar{u}_h, \bar{u}_h)(\bar{u}_h - \bar{u}_h), w \rangle_{\partial\mathcal{T}_h}, \\ \mathbf{r}(\mathbf{v}) &= -(\kappa^{-1}\bar{\mathbf{q}}_h, \mathbf{v})_{\mathcal{T}_h} + (\bar{u}_h, \nabla \cdot \mathbf{v})_{\mathcal{T}_h} - \langle \bar{u}_h, \mathbf{v} \cdot \mathbf{n} \rangle_{\partial\mathcal{T}_h}, \\ \ell(\mu) &= -\langle (\bar{\mathbf{q}}_h + \mathbf{F}(\bar{u}_h)) \cdot \mathbf{n} + \tau(\bar{u}_h, \bar{u}_h)(\bar{u}_h - \bar{u}_h), \mu \rangle_{\partial\mathcal{T}_h}, \end{aligned} \tag{15}$$

for all  $(\boldsymbol{\sigma}, \zeta, \rho)$  and  $(\mathbf{v}, w, \mu) \in \mathbf{V}_h^p \times W_h^p \times M_h^p$ . Here  $\partial_1 \tau(\cdot, \cdot)$  (respectively,  $\partial_2 \tau(\cdot, \cdot)$ ) denotes the first derivative of  $\tau$  with respect to the first argument (respectively, second argument).

The discretization of the system of Eq. (14) gives rise to a matrix equation of the form

$$\begin{bmatrix} A & -B^T & C^T \\ B & D & E \\ C & G & H \end{bmatrix} \begin{bmatrix} \delta\mathbf{Q} \\ \delta U \\ \delta A \end{bmatrix} = \begin{bmatrix} R \\ F \\ L \end{bmatrix}, \tag{16}$$

where  $\delta\mathbf{Q}$ ,  $\delta U$ , and  $\delta A$  represent the vectors of degrees of freedom for  $\delta\mathbf{q}_h$ ,  $\delta u_h$ , and  $\delta\hat{u}_h$ , respectively. The matrices in (16) correspond to the forms in (14) in the order they appear in the equations. We can write the above system as

$$\begin{bmatrix} \delta\mathbf{Q} \\ \delta U \end{bmatrix} = \begin{bmatrix} A & -B^T \\ B & D \end{bmatrix}^{-1} \left( \begin{bmatrix} R \\ F \end{bmatrix} - \begin{bmatrix} C^T \\ E \end{bmatrix} \delta A \right), \tag{17a}$$

and

$$C\delta\mathbf{Q} + G\delta U + H\delta A = L. \tag{17b}$$

We emphasize that the above inverse can be computed on each element independently of each other since the matrices  $A$ ,  $B$  and  $D$  are block-diagonal owing to the discontinuous nature of the approximation spaces  $W_h^p$  and  $\mathbf{V}_h^p$ . Moreover, the inverse matrix is block-diagonal since it results from applying the LDG method to solve the linearized PDE (2) with Dirichlet conditions at each element [8].

Finally, we insert (17a) into (17b) to obtain a reduced globally coupled matrix equation only for  $\delta A$  as

$$\mathbb{K}\delta A = \mathbb{F}, \tag{18a}$$

where  $\mathbb{K}$  is a sparse matrix given by

$$\mathbb{K} = -[C \ G] \begin{bmatrix} A & -B^T \\ B & D \end{bmatrix}^{-1} \begin{bmatrix} C^T \\ E \end{bmatrix} + H, \tag{18b}$$

and

$$\mathbb{F} = L - [C \ G] \begin{bmatrix} A & -B^T \\ B & D \end{bmatrix}^{-1} \begin{bmatrix} R \\ F \end{bmatrix}. \tag{18c}$$

Once  $\delta A$  is available both  $\delta\mathbf{Q}$  and  $\delta U$  can be obtained from (17a).

### 3. Time-dependent nonlinear problems

#### 3.1. HDG formulation

In this section, we extend the hybridizable DG methods for time-dependent nonlinear convection–diffusion problems written as a system of first-order equations

$$\begin{aligned} \mathbf{q} + \kappa \nabla u &= 0, & \text{in } \Omega, \\ \frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{F}(u) + \mathbf{q}) &= f, & \text{in } \Omega, \\ u &= g_D, & \text{on } \Gamma_D. \end{aligned} \quad (19)$$

The HDG method of lines for the above problem seeks an approximation  $(\mathbf{q}_h, u_h) \in \mathbf{V}_h^p \times W_h^p$  such that for all  $K \in \mathcal{T}_h$ ,

$$\begin{aligned} (\kappa^{-1} \mathbf{q}_h, \mathbf{v})_K - (u_h, \nabla \cdot \mathbf{v})_K + \langle \hat{u}_h, \mathbf{v} \cdot \mathbf{n} \rangle_{\partial K} &= 0, \\ \left( \frac{\partial u_h}{\partial t}, w \right)_K - (\mathbf{F}(u_h) + \mathbf{q}_h, \nabla w)_K + \langle (\hat{\mathbf{F}}_h + \hat{\mathbf{q}}_h) \cdot \mathbf{n}, w \rangle_{\partial K} &= (f, w)_K, \end{aligned} \quad (20)$$

for all  $(\mathbf{v}, w) \in (\mathcal{P}^p(K))^d \times \mathcal{P}^p(K)$ , where

$$\hat{\mathbf{F}}_h + \hat{\mathbf{q}}_h = \mathbf{F}(\hat{u}_h) + \mathbf{q}_h + \tau(u_h, \hat{u}_h)(u_h - \hat{u}_h)\mathbf{n} \quad \text{on } \partial \mathcal{T}_h.$$

Note that the formulation (20) is obtained by adding the unsteady term to the HDG formulation (3) of the steady-state case.

The above HDG formulation (20) can then be discretized in time using an appropriate time-stepping scheme. Here, we consider backward difference formulae (BDF) for the discretization of the time derivative. For instance, using the Backward–Euler scheme at time-level  $t^k$  with timestep  $\Delta t^k$  we obtain the following system

$$\begin{aligned} (\kappa^{-1} \mathbf{q}_h^k, \mathbf{v})_K - (u_h^k, \nabla \cdot \mathbf{v})_K + \langle \hat{u}_h^k, \mathbf{v} \cdot \mathbf{n} \rangle_{\partial K} &= 0, \\ \frac{1}{\Delta t^k} (u_h^k, w)_K - (\mathbf{F}(u_h^k) + \mathbf{q}_h^k, \nabla w)_K + \langle (\hat{\mathbf{F}}_h^k + \hat{\mathbf{q}}_h^k) \cdot \mathbf{n}, w \rangle_{\partial K} &= (f, w)_K + \frac{1}{\Delta t^k} (u_h^{k-1}, w)_K, \end{aligned}$$

for all  $(\mathbf{v}, w) \in (\mathcal{P}^p(K))^d \times \mathcal{P}^p(K)$ . Here, we denote  $u_h^k = u_h(t^k)$  and  $\mathbf{q}_h^k = \mathbf{q}_h(t^k)$ .

The HDG method then seeks an approximation  $(\mathbf{q}_h^k, u_h^k, \hat{u}_h^k) \in \mathbf{V}_h^p \times W_h^p \times M_h^p(g_D)$  such that

$$\begin{aligned} (\kappa^{-1} \mathbf{q}_h^k, \mathbf{v})_{\mathcal{T}_h} - (u_h^k, \nabla \cdot \mathbf{v})_{\mathcal{T}_h} + \langle \hat{u}_h^k, \mathbf{v} \cdot \mathbf{n} \rangle_{\partial \mathcal{T}_h} &= 0, \\ \frac{1}{\Delta t^k} (u_h^k, w)_{\mathcal{T}_h} - (\mathbf{F}(u_h^k) + \mathbf{q}_h^k, \nabla w)_{\mathcal{T}_h} + \langle (\hat{\mathbf{F}}_h^k + \hat{\mathbf{q}}_h^k) \cdot \mathbf{n}, w \rangle_{\partial \mathcal{T}_h} &= (f, w)_{\mathcal{T}_h} + \frac{1}{\Delta t^k} (u_h^{k-1}, w)_{\mathcal{T}_h}, \\ \langle (\hat{\mathbf{F}}_h^k + \hat{\mathbf{q}}_h^k) \cdot \mathbf{n}, \mu \rangle_{\partial \mathcal{T}_h} &= 0, \end{aligned} \quad (21)$$

for all  $(\mathbf{v}, w, \mu) \in \mathbf{V}_h^p \times W_h^p \times M_h^p(0)$ . This is done by adding the contributions over all the elements and enforcing continuity of the numerical fluxes. Here the numerical flux is given by

$$\hat{\mathbf{F}}_h^k + \hat{\mathbf{q}}_h^k = \mathbf{F}(\hat{u}_h^k) + \mathbf{q}_h^k + \tau(u_h^k, \hat{u}_h^k)(u_h^k - \hat{u}_h^k)\mathbf{n}, \quad \text{on } \partial \mathcal{T}_h.$$

We proceed to describe the implementation of the HDG method (21) as follows.

#### 3.2. Implementation

Inserting the expression of the numerical flux into (21) and after few algebraic manipulations we obtain that  $(\mathbf{q}_h^k, u_h^k, \hat{u}_h^k) \in \mathbf{V}_h^p \times W_h^p \times M_h^p(g_D)$  is the solution of the following weak formulation

$$\begin{aligned} (\kappa^{-1} \mathbf{q}_h^k, \mathbf{v})_{\mathcal{T}_h} - (u_h^k, \nabla \cdot \mathbf{v})_{\mathcal{T}_h} + \langle \hat{u}_h^k, \mathbf{v} \cdot \mathbf{n} \rangle_{\partial \mathcal{T}_h} &= 0 \\ \frac{1}{\Delta t^k} (u_h^k, w)_{\mathcal{T}_h} + (\nabla \cdot \mathbf{q}_h^k, w)_{\mathcal{T}_h} - (\mathbf{F}(u_h^k), \nabla w)_{\mathcal{T}_h} + \langle \mathbf{F}(\hat{u}_h^k) \cdot \mathbf{n} + \tau(u_h^k, \hat{u}_h^k)(u_h^k - \hat{u}_h^k), w \rangle_{\partial \mathcal{T}_h} \\ = (f, w)_{\mathcal{T}_h} + \frac{1}{\Delta t^k} (u_h^{k-1}, w)_{\mathcal{T}_h} + \langle (\mathbf{q}_h^k + \mathbf{F}(\hat{u}_h^k)) \cdot \mathbf{n} + \tau(u_h^k, \hat{u}_h^k)(u_h^k - \hat{u}_h^k), \mu \rangle_{\partial \mathcal{T}_h} &= 0, \end{aligned}$$

for all  $(\mathbf{v}, w, \mu) \in \mathbf{V}_h^p \times W_h^p \times M_h^p(0)$ . This discrete system has a similar form as the system (14) for the steady-state case except that there are two additional terms resulting from the discretization of the time derivative by means of the Backward–Euler scheme. Therefore, we can apply exactly the same solution procedure described earlier for the steady-state case to the time-dependent case at every time step.

Of course, a similar procedure can be applied to treat any higher-order backward difference formulae (BDF) method such as the widely used second-order and third-order BDF schemes. The HDG methods can also work with other implicit time-stepping methods such as the fully implicit Runge–Kutta methods and DG methods in time.

### 3.3. Extension to Neumann boundary condition

Let us end this section by extending the methods to treat Neumann boundary conditions. For example, the case when on part of the boundary  $\partial\Omega$ ,  $\partial\Omega_N$ , the Neumann boundary condition  $\mathbf{q} \cdot \mathbf{n} = q_N$  is specified can be treated as follows. First, we require that the approximate trace  $\hat{u}_h$  belongs to

$$M_h(\mathbf{g}_D) = \{\mu \in M_h : \mu = P\mathbf{g}_D \text{ on } \partial\Omega_D\}, \tag{22}$$

where  $\partial\Omega_D = \partial\Omega \setminus \partial\Omega_N$  is the Dirichlet boundary. We then replace the last equation in (3.2) with

$$\langle \mathbf{F}(\hat{u}_h^k) \cdot \mathbf{n} + \tau(u_h^k, \hat{u}_h^k)(u_h^k - \hat{u}_h^k), \mu \rangle_{\partial\mathcal{T}_h \setminus \partial\Omega_N} + \langle \mathbf{q}_h^k \cdot \mathbf{n}, \mu \rangle_{\partial\mathcal{T}_h} = \langle \mathbf{g}_N, \mu \rangle_{\partial\Omega_N}. \tag{23}$$

Hence, in order to deal with the Neumann condition  $\mathbf{q} \cdot \mathbf{n} = \mathbf{g}_N$  on  $\partial\Omega_N$  we need only to redefine the space  $M_h(\mathbf{g})$  according to (22) and modify the jump condition according to (23). The solution procedure proceeds in the same manner as described earlier.

## 4. Local postprocessing

In this section, we propose element-by-element postprocessing procedures to obtain new approximations of the scalar variable and the flux. For the scalar variable, our proposed approach exploits the optimal convergence of  $\mathbf{q}_h$ , and the super-convergence properties of  $u_h$  of the HDG method. Although we choose to discuss the local postprocessing within the HDG framework we wish to emphasize that this postprocessing method can be directly applied to other mixed methods such as the hybridized RT method and the hybridized BDM method [7] provided that these methods have similar convergence properties as the HDG method.

### 4.1. Postprocessing of the flux

We first show that we can postprocess the flux  $\mathbf{q}_h$  and its numerical trace  $\hat{\mathbf{q}}_h = \mathbf{q}_h + \tau(u_h, \hat{u}_h)(u_h - \hat{u}_h)$  with an element-by-element procedure to obtain an approximation of  $\mathbf{q}$ , denoted  $\mathbf{q}_h^*$  that belongs to  $H(\text{div}, \Omega)$  and also converges in an optimal fashion. We follow the postprocessing introduced in [3,10] and later used in [5,9].

On each simplex  $K \in \mathcal{T}_h$ , we define the new numerical flux  $\mathbf{q}_h^*$  as the only element of  $(\mathcal{P}^p(K))^d + \boldsymbol{\chi}\mathcal{P}^p(K)$  satisfying, for  $p \geq 0$ ,

$$\langle (\mathbf{q}_h^* - \hat{\mathbf{q}}_h) \cdot \mathbf{n}, \mu \rangle_e = 0, \quad \forall \mu \in \mathcal{P}^p(e), \quad \forall e \in \partial K, \quad (\mathbf{q}_h^* - \mathbf{q}_h, \mathbf{v})_K = 0, \quad \forall \mathbf{v} \in (\mathcal{P}^{p-1}(K))^d \quad \text{if } p \geq 1. \tag{24}$$

It is clear that the function  $\mathbf{q}_h^*$  belongs to  $H(\text{div}, \Omega)$ , thanks to the single valuedness of the normal component of the numerical trace  $\hat{\mathbf{q}}_h$ . It is shown in [5,9] that  $\mathbf{q}_h^*$  converges with the same order as  $\mathbf{q}_h$ . It is, however, worth noting that  $\mathbf{q}_h^*$  is an  $H(\text{div}, \Omega)$ -conforming function, whereas  $\mathbf{q}_h$  is discontinuous over  $\mathcal{T}_h$ .

### 4.2. Postprocessing of the scalar variable

Next, we postprocess  $u_h, \mathbf{q}_h$ , and  $\hat{\mathbf{q}}_h$  to obtain the new approximate scalar variable  $u_h^*$  of  $u$  which can converge at a faster rate than the original approximation  $u_h$ . Towards this end, we introduce  $\mathcal{P}^{p^*}(\partial K)$  with  $p^* = p + 1$ . We follow the proposal in [13] to seek  $u_h^* \in \mathcal{P}^{p^*}(K)$  on the simplex  $K \in \mathcal{T}_h$  such that

$$(\kappa \nabla u_h^* - \nabla w)_K = -(\mathbf{q}_h^* - \nabla w)_K, \quad \forall w \in \mathcal{P}^p(K), \quad (u_h^*, 1)_K = (u_h, 1)_K. \tag{25}$$

To compute  $u_h^*$  we need to solve the local Poisson problem (25) on each simplex  $K$  of the triangulation  $\mathcal{T}_h$ . Since the local postprocessing can be done in parallel, the new scalar variable is significantly less expensive to compute than the original approximate scalar variable.

We note that the local postprocessing (25) is the discretization of the following Poisson problem with Neumann condition

$$-\nabla \cdot (\kappa \nabla u) = \nabla \cdot \mathbf{q}_h^*, \quad \text{in } K, \quad -\kappa \nabla u \cdot \mathbf{n} = \mathbf{q}_h^* \cdot \mathbf{n}, \quad \text{on } \partial K, \quad (u, 1)_K = (u_h, 1)_K, \tag{26}$$

for each simplex  $K \in \mathcal{T}_h$ , where  $\mathbf{q}_h^*$  is the postprocessed flux and  $u_h$  is the original approximate scalar variable.

### 4.3. Remarks

The postprocessing method used here is closely related to the one introduced in [16,17]. This procedure in turn is an extension of the one introduced in [5] for a particular HDG method, and then studied in [9] for a wider class of HDG and DG methods, when the symmetric second-order elliptic problem treated therein does not have a reaction term. We must however emphasize the main differences between the local postprocessing method proposed here and the scheme suggested in [5,9]. First, the latter relies on solving the original PDE at element level, whereas the former *does not*; it only exploits the relation between the scalar variable and the flux. Therefore, the new local postprocessing method is particularly well-suited for the time-dependent problems since it can compute the new approximations at any desired timestep without the need for

computing them at the previous timesteps. Second, since our postprocessing approach only solves a simple Poisson problem on each element, it is easily extended to systems of nonlinear convection–diffusion equations.

To be successful, our postprocessing procedures rely on the optimal convergence of  $\mathbf{q}_h$  and on the superconvergence of the average of the approximate scalar variable  $u_h$ . In fact, these properties for the HDG method been theoretically analyzed and confirmed by numerical experiments for linear elliptic problems [5,9], linear convection–diffusion problems [6,13]: both  $\mathbf{q}_h^*$  and  $\mathbf{q}_h$  converge with order  $p + 1$ , while  $(u_h, 1)_K$  superconverges with order  $p + 2$ . Numerical experiments presented in Section 4 show that the HDG method developed here for the nonlinear convection–diffusion equations inherits the above-mentioned convergence properties even for the time-dependent case. As a result, we expect that the new approximate scalar variable  $u_h^*$  converges with order  $p + 2$  in  $L^2$ -norm and  $p + 1$  in  $H^1$ -norm even in the time-dependent case assuming that sufficiently time accurate integration schemes are used.

### 5. Numerical results

In this section, we present numerical examples to demonstrate the convergence and accuracy of the proposed HDG methods. We shall compare the performance of two HDG schemes. The first, called the HDG-I scheme, takes  $\tau(u_h, \hat{u}_h) := C_\tau$  and so its numerical trace in Eq. (5) becomes

$$\hat{\mathbf{q}}_h + \hat{\mathbf{F}}_h = \mathbf{q}_h + \mathbf{F}(\hat{u}_h) + C_\tau(u_h - \hat{u}_h)\mathbf{n}, \quad \text{on } \mathcal{E}_h, \tag{27a}$$

where  $C_\tau$  is a suitably chosen constant. The second, named the HDG-II scheme, takes  $\tau(u_h, \hat{u}_h) := C_\tau + (\mathbf{F}(u_h) - \mathbf{F}(\hat{u}_h)) \cdot \mathbf{n} / (u_h - \hat{u}_h)$  so that the numerical trace in Eq. (5) becomes

$$\hat{\mathbf{q}}_h + \hat{\mathbf{F}}_h = \mathbf{q}_h + \mathbf{F}(u_h) + C_\tau(u_h - \hat{u}_h)\mathbf{n}, \quad \text{on } \mathcal{E}_h. \tag{27b}$$

The first example serves as validation for the steady problem; and the second example, as validation for the time-dependent case. The examples are steady and unsteady viscous Burgers problems with small Reynolds number.

#### 5.1. Steady viscous Burgers problem

Our first example is a steady viscous Burger problem of the form (1) in which  $\Omega = (0, 1) \times (0, 1)$ ,  $\kappa = 0.1$ ,  $\mathbf{F}(u) = (u^2/2, u^2/2)$ , and  $\mathbf{g}_D = 0$  on  $\partial\Omega$ . The source term  $f$  is chosen such that the exact solution is

$$u = xy \tanh\left(\frac{1-x}{\kappa}\right) \tanh\left(\frac{1-y}{\kappa}\right).$$

We consider triangular meshes obtained by splitting a regular  $n \times n$  Cartesian grid into a total of  $2n^2$  triangles, giving uniform element sizes of  $h = 1/n$ . On these meshes, we consider solutions of polynomial degree  $p$  represented using a nodal basis within each element, with the nodes uniformly distributed.

**Table 1**  
Example 1: History of convergence for the HDG-I scheme.

Degree	Mesh	$\ u - u_h\ _{L^2(\Omega)}$		$\ \mathbf{q} - \mathbf{q}_h\ _{L^2(\Omega)}$		$\ \mathbf{q} - \mathbf{q}_h^*\ _{L^2(\Omega)}$		$\ u - u_h^*\ _{L^2(\Omega)}$	
		Error	Order	Error	Order	Error	Order	Error	Order
0	4	4.29e-2	-	5.58e-2	-	1.78e-1	-	1.70e-1	-
	8	3.66e-2	0.23	5.03e-2	0.15	8.76e-2	1.02	5.60e-2	1.60
	16	2.21e-2	0.73	3.72e-2	0.44	4.13e-2	1.08	2.16e-2	1.38
	32	1.17e-2	0.92	2.05e-2	0.86	2.13e-2	0.95	1.10e-2	0.97
	64	6.01e-3	0.96	1.08e-2	0.92	1.09e-2	0.97	5.78e-3	0.93
1	4	7.76e-2	-	6.90e-2	-	5.06e-2	-	2.51e-2	-
	8	2.52e-2	1.62	2.55e-2	1.44	1.62e-2	1.64	5.50e-3	2.19
	16	6.72e-3	1.91	1.05e-2	1.27	5.06e-3	1.68	8.14e-4	2.76
	32	1.76e-3	1.93	2.87e-3	1.88	1.32e-3	1.94	1.08e-4	2.92
	64	4.49e-4	1.97	7.37e-4	1.96	3.35e-4	1.98	1.38e-5	2.97
2	4	1.44e-2	-	2.39e-2	-	2.12e-2	-	7.35e-3	-
	8	2.73e-3	2.40	7.84e-3	1.61	4.75e-3	2.16	9.18e-4	3.00
	16	4.49e-4	2.61	1.08e-3	2.87	6.10e-4	2.96	5.48e-5	4.07
	32	5.87e-5	2.94	1.36e-4	2.99	7.66e-5	2.99	3.34e-6	4.04
	64	7.44e-6	2.98	1.73e-5	2.97	9.65e-6	2.99	2.08e-7	4.00
3	4	4.55e-3	-	1.23e-2	-	9.21e-3	-	2.86e-3	-
	8	5.97e-4	2.93	1.23e-3	3.32	8.17e-4	3.49	1.11e-4	4.68
	16	4.14e-5	3.85	9.04e-5	3.76	6.16e-5	3.73	4.21e-6	4.73
	32	2.79e-6	3.89	7.03e-6	3.69	4.38e-6	3.81	1.52e-7	4.80
	64	1.77e-7	3.98	4.49e-7	3.97	2.77e-7	3.98	4.76e-9	4.99

In this example, we study the performance of the HDG-I scheme (with the numerical flux (27a)) and the HDG-II scheme (with the numerical flux (27b)), and the effectiveness of our local postprocessing procedure. For both schemes, we take

$$C_\tau \geq \kappa/\ell + C^{LF}(I)/2,$$

where  $I$  is an interval including the ranges of  $u_h$  and  $\hat{u}_h$ ; in this way, the condition (7) is satisfied and the positivity property (6) follows. Since we are solving our system of equations in an iterative way, we can compute the constant  $C_\tau$  also in an iterative manner. How to do this in an efficient way will be considered elsewhere. Here we simply take

$$C_\tau = 1,$$

given that  $|\mathbf{F}'(u(\mathbf{x}))| \leq \sqrt{2}\nu \mathbf{x} \in \Omega$ , and  $\kappa/\ell + \max_{\mathbf{x} \in \Omega} |\mathbf{F}'(u(\mathbf{x}))|/2 = 0.1/1 + \sqrt{2}/2 \approx 0.8$ . This choice works well in the numerical experiments as we discuss next.

We present the error and order of convergence in  $L^2$ -norm in Table 1 for the HDG-I scheme and Table 2 for the HDG-II scheme. We see that in all cases the approximate scalar variable  $u_h$  and the approximate total flux  $\mathbf{q}_h$  converge optimally with order  $p + 1$  for  $p = 0, 1, 2$ , and 3. It is important to point out that some other DG methods such as the LDG method may not produce optimal convergent approximations for the special case  $p = 0$ , and that  $p = 0$  solution may be useful in certain cases, for example, when one want to construct a multi-grid solver. We also observe that the HDG-I and HDG-II yield very similar results as far as the error and convergence rate are concerned.

As for results of the local postprocessing, the postprocessed flux  $\mathbf{q}_h^*$  converges with the same order as  $\mathbf{q}_h$ ; however, the former has slightly smaller error than the latter. Most notably, the postprocessed scalar variable  $u_h^*$  converges with order  $p + 2$  in  $L^2$ -norm for  $p \geq 1$ , which is one order higher than the convergence rate of the original scalar variable  $u_h$ . For the special case  $p = 0$ , however,  $u_h^*$  converges with order  $p + 1$  just like  $u_h$  since the average of  $u_h$  on each element converges with order  $p + 1$  for  $p = 0$ . These results are similar to those obtained for linear convection–diffusion problems [6,13]. Finally, we plot in Figs. 1 and 2  $u_h$  and  $u_h^*$  on the meshes  $h = 1/4$  and  $h = 1/8$ , respectively. In all these of figures, we see a clear improvement of the approximations as we increase from  $k = 1$  to  $k = 3$ . Furthermore, the local postprocessing improves the approximate solution significantly for  $k = 1$  and  $k = 2$ , as the postprocessed scalar variable  $u_h^*$  is clearly superior to the original scalar variable  $u_h$ .

### 5.2. Unsteady viscous Burgers problem

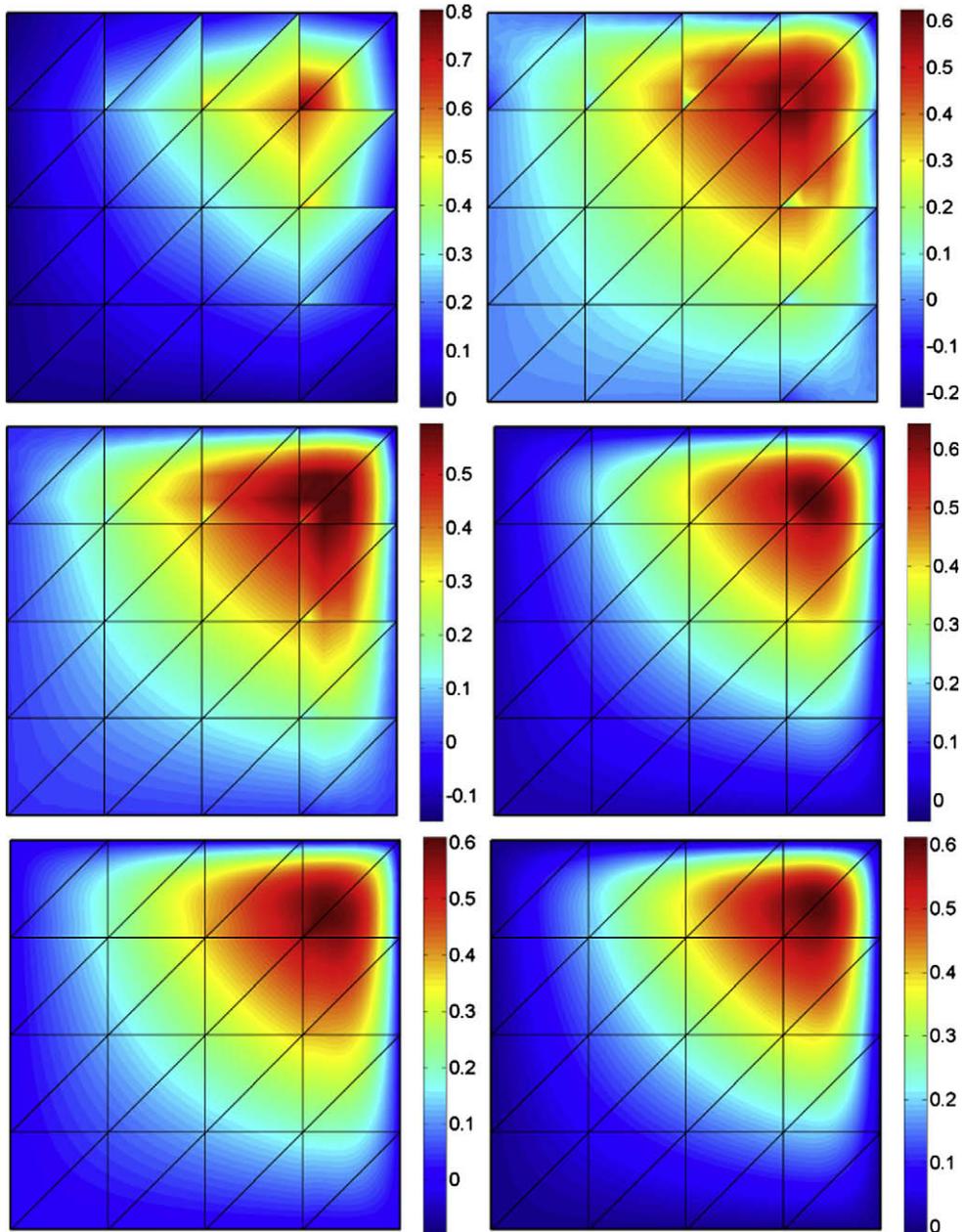
This example is the unsteady version of the first example with  $\Omega = (0, 1) \times (0, 1)$ ,  $\mathbf{F}(u) = (u^2/2, u^2/2)$ ,  $g_D = 0$  on  $\partial\Omega$  and  $u(\cdot, 0) = 0$ . The exact solution is given by

$$u = (e^t - 1)xy \tanh\left(\frac{1-x}{\kappa}\right) \tanh\left(\frac{1-y}{\kappa}\right).$$

The source term  $f$  is then determined from this exact solution with  $\kappa = 0.1$ . For the temporal discretization, we use the third-order backward difference formula (BDF3) to discretize the time derivative with a constant timestep  $\Delta t$ . As in the steady-state case, we take the stabilization parameter  $C_\tau := 1$  on  $\partial\mathcal{T}_h$ . This works well for the numerical example we consider

**Table 2**  
Example 1: History of convergence for the HDG-II scheme.

Degree $p$	Mesh $n$	$\ u - u_h\ _{L^2(\Omega)}$		$\ \mathbf{q} - \mathbf{q}_h\ _{L^2(\Omega)}$		$\ \mathbf{q} - \mathbf{q}_h^*\ _{L^2(\Omega)}$		$\ u - u_h^*\ _{L^2(\Omega)}$	
		Error	Order	Error	Order	Error	Order	Error	Order
0	4	4.66e-2	-	5.58e-2	-	1.59e-1	-	1.49e-1	-
	8	4.24e-2	0.14	5.35e-2	0.06	8.36e-2	0.93	5.52e-2	1.43
	16	2.59e-2	0.71	3.73e-2	0.52	3.87e-2	1.11	2.40e-2	1.20
	32	1.39e-2	0.90	2.05e-2	0.86	1.97e-2	0.98	1.30e-2	0.89
	64	7.19e-3	0.95	1.08e-2	0.92	9.99e-3	0.98	6.91e-3	0.91
1	4	7.75e-2	-	6.95e-2	-	5.11e-2	-	2.46e-2	-
	8	2.54e-2	1.61	2.53e-2	1.46	1.65e-2	1.63	5.03e-3	2.29
	16	6.73e-3	1.92	1.05e-2	1.26	5.00e-3	1.73	7.88e-4	2.68
	32	1.76e-3	1.94	2.87e-3	1.88	1.29e-3	1.95	1.06e-4	2.90
	64	4.47e-4	1.97	7.35e-4	1.96	3.25e-4	1.99	1.35e-5	2.96
2	4	1.51e-2	-	2.37e-2	-	2.20e-2	-	7.15e-3	-
	8	2.90e-3	2.38	7.91e-3	1.58	4.88e-3	2.17	9.17e-4	2.96
	16	4.56e-4	2.67	1.08e-3	2.87	6.13e-4	2.99	5.46e-5	4.07
	32	5.87e-5	2.96	1.36e-4	2.99	7.47e-5	3.04	3.25e-6	4.07
	64	7.39e-6	2.99	1.73e-5	2.97	9.29e-6	3.01	2.01e-7	4.01
3	4	4.76e-3	-	1.22e-2	-	9.46e-3	-	2.85e-3	-
	8	6.07e-4	2.97	1.23e-3	3.31	8.28e-4	3.52	1.11e-4	4.68
	16	4.21e-5	3.85	8.97e-5	3.78	6.11e-5	3.76	4.07e-6	4.76
	32	2.80e-6	3.91	7.03e-6	3.67	4.30e-6	3.83	1.48e-7	4.78
	64	1.76e-7	3.99	4.50e-7	3.97	2.69e-7	4.00	4.65e-9	5.00

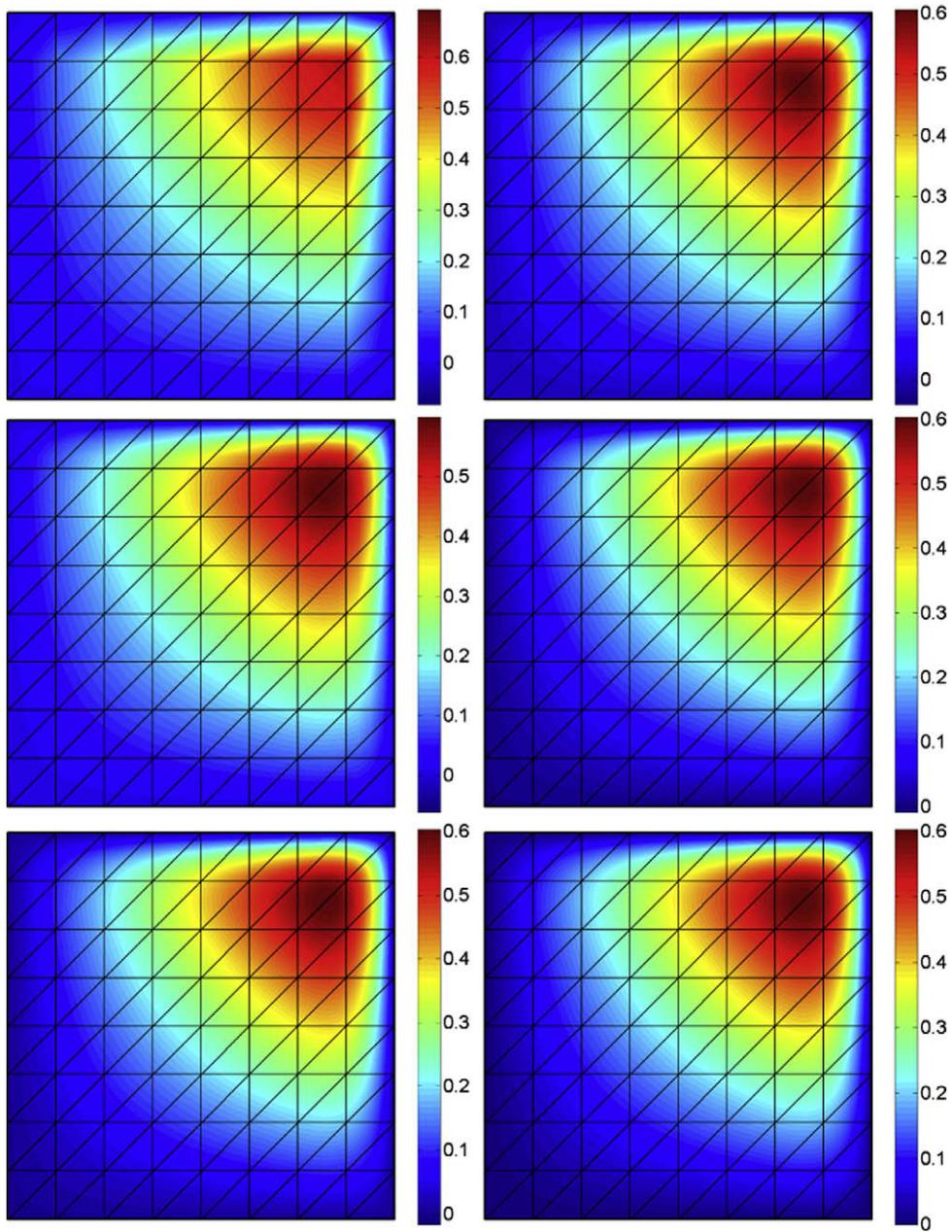


**Fig. 1.** Example 1: Comparison of  $u_h$  (left) and  $u_h^*$  (right) computed by the HDG-I scheme on the mesh  $h = 1/4$  for  $p = 1$  (top),  $p = 2$  (middle), and  $p = 3$  (bottom). See Table 1 for the corresponding  $L^2(\Omega)$  errors.

here, but a more complete analysis would require the extension of the energy equality (1) to the time dependent case in order to find the new positivity condition (6) on the stabilization function  $\tau(u_h, \hat{u}_h)$ . This will be done elsewhere.

We first look at the convergence of the numerical solution with respect to time discretization. For this purpose, we use a very fine triangulation of size  $h = 1/64$  and polynomials of order  $p = 3$  so that the overall error is governed by the temporal error alone. We show in Table 3 the error and order of convergence as a function of  $\Delta t$  for the approximate scalar variable  $u_h$  at the final time  $T = 1$ . The accuracy is third-order in time and it is optimal for the BDF3 scheme.

We now look at the spatial convergence of the HDG methods. For this purpose we consider a very small timestep  $\Delta t = 5 \times 10^{-3}$  so that the overall error is governed by the spatial error. We present the history of convergence in Tables 4 and 5 for the HDG-I and HDG-II, respectively. We see that both the approximate scalar variable  $u_h$  and total flux  $\mathbf{q}_h$  converge optimally with order  $p + 1$ . The postprocessed total flux  $\mathbf{q}_h^*$  converges with the optimal order  $p + 1$  just like  $\mathbf{q}_h$ ; unlike  $\mathbf{q}_h$ , however,  $\mathbf{q}_h^*$  has the normal component continuous across the element interfaces. The postprocessed scalar variable  $u_h^*$



**Fig. 2.** Example 1: Comparison of  $u_h$  (left) and  $u_h^*$  (right) computed by the HDG-I scheme on the mesh  $h = 1/8$  for  $p = 1$  (top),  $p = 2$  (middle), and  $p = 3$  (bottom). See Table 1 for the corresponding  $L^2(\Omega)$  errors.

**Table 3**

Example 2: History of convergence of the  $L^2$ -error of  $u_h$  at the final time computed by the HDG-I and HDG-II as a function of timestep  $\Delta t$ . The BDF3 scheme is used for time integration.

$\Delta t$	HDG-I		HDG-II	
	Error	Order	Error	Order
0.2	1.90e-4	-	1.90e-4	-
0.1	2.95e-5	2.68	2.95e-5	2.68
0.05	4.01e-6	2.88	4.01e-6	2.88
0.025	5.37e-7	2.90	5.30e-7	2.92

**Table 4**

Example 2: History of convergence for the HDG-I scheme.

Degree $p$	Mesh $n$	$\ u - u_h\ _{L^2(\Omega)}$		$\ q - q_h\ _{L^2(\Omega)}$		$\ q - q_h^*\ _{L^2(\Omega)}$		$\ u - u_h^*\ _{L^2(\Omega)}$	
		Error	Order	Error	Order	Error	Order	Error	Order
0	4	4.98e-2	–	9.35e-2	–	2.96e-1	–	2.86e-1	–
	8	4.16e-2	0.26	8.43e-2	0.15	1.49e-1	0.99	8.45e-2	1.76
	16	2.45e-2	0.77	6.17e-2	0.45	7.39e-2	1.01	2.60e-2	1.70
	32	1.29e-2	0.93	3.41e-2	0.86	3.85e-2	0.94	1.20e-2	1.12
	64	6.60e-3	0.96	1.80e-2	0.92	1.98e-2	0.96	6.22e-3	0.95
1	4	1.38e-1	–	1.19e-1	–	8.97e-2	–	3.96e-2	–
	8	4.39e-2	1.66	4.41e-2	1.43	2.81e-2	1.67	9.04e-3	2.13
	16	1.16e-2	1.92	1.80e-2	1.29	8.65e-3	1.70	1.38e-3	2.71
	32	3.03e-3	1.93	4.92e-3	1.87	2.27e-3	1.93	1.85e-4	2.90
	64	7.71e-4	1.97	1.26e-3	1.96	5.74e-4	1.98	2.38e-5	2.96
2	4	2.52e-2	–	4.09e-2	–	3.65e-2	–	1.24e-2	–
	8	4.60e-3	2.45	1.32e-2	1.63	8.01e-3	2.19	1.59e-3	2.97
	16	7.64e-4	2.59	1.82e-3	2.86	1.03e-3	2.95	9.46e-5	4.07
	32	1.00e-4	2.93	2.32e-4	2.97	1.31e-4	2.98	5.74e-6	4.04
	64	1.28e-5	2.98	2.97e-5	2.97	1.65e-5	2.98	3.57e-7	4.01

**Table 5**

Example 2: History of convergence for the HDG-II scheme.

Degree $p$	Mesh $n$	$\ u - u_h\ _{L^2(\Omega)}$		$\ q - q_h\ _{L^2(\Omega)}$		$\ q - q_h^*\ _{L^2(\Omega)}$		$\ u - u_h^*\ _{L^2(\Omega)}$	
		Error	Order	Error	Order	Error	Order	Error	Order
0	4	6.65e-2	–	1.02e-2	–	2.61e-1	–	2.43e-1	–
	8	5.22e-2	0.35	8.70e-2	0.23	1.40e-1	0.89	7.77e-2	1.65
	16	3.16e-2	0.72	5.86e-2	0.57	6.70e-2	1.07	2.81e-2	1.47
	32	1.70e-2	0.90	3.21e-2	0.87	3.40e-2	0.98	1.50e-2	0.90
	64	8.79e-3	0.95	1.69e-2	0.93	1.73e-2	0.98	8.20e-3	0.87
1	4	1.38e-1	–	1.21e-1	–	9.47e-2	–	3.84e-2	–
	8	4.46e-2	1.63	4.33e-2	1.48	3.05e-2	1.63	7.80e-3	2.30
	16	1.16e-2	1.94	1.79e-2	1.28	8.81e-3	1.79	1.30e-3	2.58
	32	3.03e-3	1.94	4.89e-3	1.87	2.24e-3	1.97	1.77e-4	2.88
	64	7.69e-4	1.98	1.26e-3	1.96	5.62e-4	2.00	2.28e-5	2.96
2	4	2.70e-2	–	4.03e-2	–	3.90e-2	–	1.19e-2	–
	8	5.09e-3	2.41	1.34e-2	1.59	8.50e-3	2.20	1.60e-3	2.89
	16	7.86e-4	2.69	1.84e-3	2.87	1.05e-3	3.01	9.46e-5	4.08
	32	1.01e-4	2.97	2.32e-4	2.98	1.27e-4	3.05	5.54e-6	4.09
	64	1.26e-5	2.99	2.97e-5	2.97	1.57e-5	3.01	3.41e-7	4.02

superconverges with order  $p + 2$  for  $p \geq 1$ , which is one order higher than the original approximation  $u_h$ . The presented results indicate that when polynomials of degree  $p \geq 1$  are used in the HDG-spatial discretization, all the approximate variables converge with the optimal order. Moreover, the local postprocessing procedure is effective in improving the convergence order of the scalar variable even for the time-dependent nonlinear case.

## 6. Conclusion

In this paper, we present implicit high-order hybridizable DG (HDG) methods for steady-state and time-dependent nonlinear convection–diffusion equations. The main motivation is the reduction in the number of the globally coupled degrees of freedom of the DG approximations. The methods developed achieve this objective by expressing the approximate scalar variable and flux in terms of the approximate trace of the scalar variable and enforcing flux continuity explicitly. This allows us to eliminate both the approximate solution and flux to obtain a matrix equation involving only the numerical trace at every Newton iteration. We propose two different flux formulas and extend the methods to time-dependent convection–diffusion problems. Numerical results indicate that both the approximate scalar variable and flux converge with the optimal order  $p + 1$ . We note that for many other DG methods the approximate flux variable converges with the suboptimal order  $p$ .

Based on the optimal convergence of the HDG methods, we employ a local postprocessing procedure to obtain new approximations of the scalar variable and flux. The postprocessed flux converges with the same order as the original flux; however, the normal component of the postprocessed flux is continuous, while that of the original flux is discontinuous. The postprocessed scalar variable converges with order  $p + 2$  for  $p \geq 1$ , which is one order higher than the original scalar

variable. Moreover, the postprocessing procedure is less expensive than the solution procedure, since it solves a simple Poisson problem at the element level. Thus we obtain  $p + 2$  convergent solution with using the polynomial degree  $p$ .

We end this paper by pointing out that the extension of this work to the purely convective case and then to nonlinear hyperbolic systems of conservation constitutes the subject of ongoing research.

## Acknowledgments

J. Peraire and N. C. Nguyen would like to acknowledge the Singapore-MIT Alliance for partially supporting this work. B. Cockburn would like to acknowledge the National Science Foundation for partially supporting this work through Grant DMS-0712955.

## References

- [1] D.N. Arnold, F. Brezzi, B. Cockburn, L.D. Marini, Unified analysis of discontinuous Galerkin methods for elliptic problems, *SIAM J. Numer. Anal.* 39 (5) (2001) 1749–1779.
- [2] F. Bassi, S. Rebay, A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations, *J. Comput. Phys.* 131 (1997) 267–279.
- [3] P. Bastian, B. Rivière, Superconvergence and  $H(\text{div})$  projection for discontinuous Galerkin methods, *Int. J. Numer. Meth. Fluids* 42 (2003) 1043–1057.
- [4] C. Baumann, J. Oden, A discontinuous hp finite element method for convection–diffusion problems, *Comput. Meth. Appl. Mech. Eng.* 175 (1999) 311–341.
- [5] B. Cockburn, B. Dong, J. Guzmán, A superconvergent LDG-hybridizable Galerkin method for second-order elliptic problems, *Math. Comp.* 77 (2008) 1887–1916.
- [6] B. Cockburn, B. Dong, J. Guzmán, M. Restelli, R. Sacco, Superconvergent and optimally convergent LDG-hybridizable discontinuous Galerkin methods for convection–diffusion–reaction problems, *SIAM J. Sci. Comput.*, in press.
- [7] B. Cockburn, J. Gopalakrishnan, A characterization of hybridized mixed methods for second order elliptic problems, *SIAM J. Numer. Anal.* 42 (1) (2004) 283–301.
- [8] B. Cockburn, J. Gopalakrishnan, R. Lazarov, Unified hybridization of discontinuous Galerkin, mixed and continuous Galerkin methods for second-order elliptic problems, *SIAM J. Numer. Anal.* 47 (2009) 1319–1365.
- [9] B. Cockburn, J. Guzmán, H. Wang, Superconvergent discontinuous Galerkin methods for second-order elliptic problems, *Math. Comp.* 78 (2009) 1–24.
- [10] B. Cockburn, G. Kanschat, D. Schötzau, A locally conservative LDG method for the incompressible Navier–Stokes equations, *Math. Comp.* 74 (2005) 1067–1095.
- [11] B. Cockburn, C.W. Shu, The local discontinuous Galerkin method for convection–diffusion systems, *SIAM J. Numer. Anal.* 35 (1998) 2440–2463.
- [12] R.J. LeVeque, *Numerical Methods for Conservation Laws*. Lectures in Mathematics, ETH-Zurich, Birkhauser-Verlag, Basel, 1990.
- [13] N.C. Nguyen, J. Peraire, B. Cockburn, An implicit high-order hybridizable discontinuous Galerkin method for linear convection–diffusion equations, *J. Comput. Phys.* 228 (2009) 3232–3254.
- [14] J. Peraire, P.O. Persson, The compact discontinuous Galerkin (CDG) method for elliptic problems, *SIAM J. Sci. Comput.* 30 (4) (2008) 1806–1824.
- [15] P.O. Persson, J. Peraire, Newton-GMRES preconditioning for discontinuous Galerkin discretizations of the Navier–Stokes equations, *SIAM J. Sci. Comput.* 30 (6) (2008) 2709–2733.
- [16] R. Stenberg, A family of mixed finite elements for the elasticity problem, *Numer. Math.* 53 (1988) 513–538.
- [17] R. Stenberg, Postprocessing schemes for some mixed finite elements, *RAIRO Modél. Math. Anal. Numér.* 25 (1991) 151–167.