Journal of Computational Physics 228 (2009) 3232–3254

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

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

N.C. Nguyen [a,*], J. Peraire [a], B. Cockburn [b]

[a] Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Rm 37-435, 77 Massachusetts Avenue, Cambridge, MA 02139, USA
[b] School of Mathematics, University of Minnesota, Minneapolis, MN 55455, USA

ARTICLE INFO

Article history:
Received 7 August 2008
Received in revised form 11 December 2008
Accepted 14 January 2009
Available online 6 February 2009

Keywords:
Finite element methods
Discontinuous Galerkin methods
Hybrid/mixed methods
Convection–diffusion equations

ABSTRACT

We present a hybridizable discontinuous Galerkin method for the numerical solution of steady and time-dependent linear convection–diffusion equations. We devise the method as follows. First, we express the approximate scalar variable and corresponding flux within each element in terms of an approximate trace of the scalar variable along the element boundary. We then define a unique value for the approximate trace by enforcing the continuity of the normal component of the flux across the element boundary; a global equation system solely in terms of the approximate trace is thus obtained. The high number of globally coupled degrees of freedom in the discontinuous Galerkin approximation is therefore significantly reduced. If the problem is time-dependent, we discretize the time derivative by means of backward difference formulae. This results in efficient schemes capable of producing high-order accurate solutions in space and time. Indeed, when the time-marching method is $(p+1)$th order accurate and when polynomials of degree $p \geqslant 0$ are used to represent the scalar variable, the flux and the approximate trace, we observe that the approximations for the scalar variable, the flux and the trace of the scalar variable converge with the optimal order of $p+1$ in the L^2-norm. Finally, we introduce 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 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. 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. Moreover, the postprocessing procedure is less expensive than the solution procedure, since it is performed at the element level. Extensive numerical results are presented to demonstrate the convergence properties of the method.

Published by Elsevier Inc.

1. Introduction

In this paper, we present a hybridizable discontinuous Galerkin (HDG) method for the solution of steady and time-dependent convection–diffusion equations. The method is an extension to time-dependent convection–diffusion problems of the class of HDG methods first introduced in [10] for symmetric second-order elliptic problems. This method was then extended to steady-state convection–diffusion in [8] using a different approximation for the total flux than the one considered here.

As it is typical of HDG methods, to carry out the discretization in space, we proceed in two main steps. First, the approximate scalar variable and flux are expressed in an element-by-element fashion in terms of an approximate trace of the scalar

* Corresponding author. Tel.: +1 617 253 8080.
E-mail address: cuongng@mit.edu (N.C. Nguyen).

0021-9991/$ - see front matter Published by Elsevier Inc.
doi:10.1016/j.jcp.2009.01.030

variable along the element boundary. Then, a unique value for the trace at inter-element boundaries is obtained by enforcing flux continuity. This leads to a global equation system in terms of the approximate boundary traces only. This system has a significantly smaller number of globally coupled degrees of freedom relative to more standard discontinuous Galerkin methods analyzed in [1]. Moreover, the stiffness matrices are of the same size and structure to those associated with the corresponding hybridized version of the mixed methods; see [10]. It also results in more accurate approximations when the stabilization parameters are suitably chosen; see [7,11].

It turns out that several well known DG methods, such as some particular cases of the Local Discontinuous Galerkin (LDG) method [13], as well as some stabilized interior penalty methods [14], can be cast as HDG methods; see [10]. In the case of those special LDG methods, their hybridized formulation can be interpreted as a re-parametrization of the discrete solution resulting in a reduced number of globally coupled unknowns. In addition, the method possesses a compact discretization which eliminates the non-compact nature of the standard LDG method in multi-dimensions [17,20]. Another advantage of the HDG methods introduced in [10] is that they include a wide family of schemes in which both the approximate scalar variable and the approximate flux converge at a rate of $p+1$ in L^2-norm when polynomials of order p are used to represent the scalar variable, the flux and the approximate trace. For $p \geqslant 1$, a simple local postprocessing procedure can then be used to obtain a solution that converges at a rate of $p+2$ in the L^2-norm [7,11]. In addition, unlike the standard LDG methods, it is possible to obtain optimally convergent approximations for elliptic problems even for $p=0$.

We consider HDG methods built by using, on each element, LDG methods to express the approximate scalar variable and the flux in terms of an approximate trace along the element boundary. Our methods are thus LDG-hybridizable, according to the terminology introduced in [10], but we are going to call them HDG methods for simplicity. (Note that all the standard LDG schemes which have a zero constant C_{22} cannot be HDG schemes. To be an HDG scheme, an LDG scheme must have a positive value for C_{22} and $|\boldsymbol{C}_{12}\cdot\boldsymbol{n}|=\frac{1}{2}$ [10]; for example, only the LDG method introduced in [5] falls into this category. The Compact DG scheme introduced in [17] is not an HDG method; it is a modification of an LDG method.) First, we present the HDG methods and describe their implementation for steady-state convection–diffusion equations. We next derive a weak formulation that characterizes the numerical solution in terms of the numerical trace only and prove the existence and uniqueness of the numerical trace. In addition, we propose two different choices of the stabilization parameters which are shown to provide an accurate and stable solution even in the diffusive and convective limits. We provide numerical examples indicating that both the approximate scalar variable and the approximate flux converge optimally with order $p+1$ in the L^2-norm even for the special case $p=0$.

We then extend our methods to time-dependent convection–diffusion equations by employing backward difference formulaes for the discretization of the time derivative in Section 3.7. The resulting methods are implicit, stable and high-order accurate and involve significantly less degrees of freedom than standard implicit DG methods [18]. We present numerical results indicating that the HDG methods developed here exhibit the above-mentioned convergence properties in the time-dependent case.

Finally, we introduce a simple element-by-element postprocessing scheme to obtain new approximations of the flux and the scalar variable. The new approximate flux is obtained by employing the projection used in [8]. This projection, and various variations, have already been used in several different situations; see [3,7,11,12]. Unlike the original approximate flux, the postprocessed flux has a continuous inter-element normal component. Moreover, it converges with the optimal order of $p+1$ in the L^2-norm, which is the same as the original flux. The new approximate scalar variable is obtained by using a postprocessing which is an extension of the one introduced in [7], and then used in [11] for the symmetric second-order elliptic problem. It is shown to converge with order $p+2$ in L^2-norm which is one-order higher than the original approximate scalar variable. Since the local postprocessing is performed at element level, the new approximations are even less expensive to compute than the approximate solution. Let us stress the fact that, unlike the cases considered in [7,8,11], the local postprocessing scheme does *not* involve the original partial differential equation. It is thus particularly well-suited for time-dependent problems. Moreover, it does not have to be applied at each time-step, but only at desired times during the simulation. Therefore, compared with the more established DG methods, the proposed approach can be more efficient: the $p+2$ convergent solution can be computed at the cost of a DG approximation using polynomials of degree p.

The paper is organized as follows: In Section 2, we introduce the steady-state convection–diffusion model equation and the notation. In Section 3, we describe the HDG-space discretization method and then extend it to the time-dependent case (see Section 3.7). In Section 4, we introduce the local postprocessing procedure to compute a higher-order accurate solution. In Section 5, we provide extensive numerical results to assess the convergence and accuracy of the method. Although our examples are in two dimension, the present method can be applied to problems in one and three dimensions. Finally, we present some concluding remarks in Section 6.

2. Problem statement and notation

2.1. The convection–diffusion model equation

We consider the following convection–diffusion model problem

$$\nabla\cdot(\boldsymbol{c}u)-\nabla\cdot(\kappa\nabla u)=f,\quad \text{in } \Omega, \tag{1}$$

with boundary conditions

$$\begin{aligned} &u = g_D, \quad \text{on } \Gamma_D, \\ &(-\kappa\nabla u + \boldsymbol{c}u)\cdot\boldsymbol{n} = g_N, \quad \text{on } \Gamma_N. \end{aligned} \tag{2}$$

Here $\Omega \in \mathbb{R}^d$ is the physical domain with boundary $\partial\Omega$, $f \in L^2(\Omega)$ is a source term, $\kappa \in L^\infty(\Omega)$ is a positive diffusivity coefficient, and $\boldsymbol{c} \in (L^\infty(\Omega))^d$ is a smooth velocity vector field. As usual, we assume that the $(d-1)$-Lebesgue measure of Γ_D is not zero and that $\partial\Omega = \overline{\Gamma}_D \cup \overline{\Gamma}_N$ and $\Gamma_D \cap \Gamma_N = \emptyset$.

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

$$\begin{aligned} &\boldsymbol{q} + \kappa\nabla u = 0, \quad \text{in } \Omega, \\ &\nabla\cdot(\boldsymbol{c}u + \boldsymbol{q}) = f, \quad \text{in } \Omega, \end{aligned} \tag{3}$$

with boundary conditions

$$\begin{aligned} &u = g_D, \quad \text{on } \Gamma_D, \\ &(\boldsymbol{q} + \boldsymbol{c}u)\cdot\boldsymbol{n} = g_N, \quad \text{on } \Gamma_N. \end{aligned} \tag{4}$$

Next, we introduce the notation we are going to use to describe the HDG method.

2.2. Mesh and trace operators

Let $\mathcal{T}_h$ be a collection of disjoint elements that partition Ω. We denote by $\partial\mathcal{T}_h$ the set $\{\partial K : K \in \mathcal{T}_h\}$. 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 non-zero. 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 non-zero. Let $\mathcal{E}_h^o$ and $\mathcal{E}_h^\partial$ denote the set of interior and boundary faces, respectively. We denote by $\mathcal{E}_h$ the union of $\mathcal{E}_h^o$ and $\mathcal{E}_h^\partial$.

Let $\boldsymbol{n}^+$ and $\boldsymbol{n}^-$ be the outward unit normals of ∂K^+ and ∂K^-, respectively, and let $(\boldsymbol{q}^\pm, u^\pm)$ be the traces of $(\boldsymbol{q}, u)$ on e from the interior of $K^\pm$. Then, we define the mean values $\{\{\cdot\}\}$ and jumps $[\![\cdot]\!]$ as follows. For $e \in \mathcal{E}_h^i$, we set

$$\{\{\boldsymbol{q}\}\} = (\boldsymbol{q}^+ + \boldsymbol{q}^-)/2 \quad \{\{u\}\} = (u^+ + u^-)/2,$$

$$[\![\boldsymbol{q}\cdot\boldsymbol{n}]\!] = \boldsymbol{q}^+\cdot\boldsymbol{n}^+ + \boldsymbol{q}^-\cdot\boldsymbol{n}^- \qquad [\![u\boldsymbol{n}]\!] = u^+\boldsymbol{n}^+ + u^-\boldsymbol{n}^-.$$

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

$$\{\{\boldsymbol{q}\}\} = \boldsymbol{q} \quad \{\{u\}\} = u,$$

$$[\![\boldsymbol{q}\cdot\boldsymbol{n}]\!] = \boldsymbol{q}\cdot\boldsymbol{n} \quad [\![u\boldsymbol{n}]\!] = u\boldsymbol{n},$$

where $\boldsymbol{n}$ is the outward normal to $\partial\Omega$. Note that the jump in u is a vector, but the jump in $\boldsymbol{q}$ is a scalar which only involves the normal component of $\boldsymbol{q}$. Furthermore, the jump will be zero for a continuous function.

2.3. Approximation spaces

Let $\mathcal{P}^p(D)$ denote the set of polynomials of degree at most p on a domain D. For any element K of the collection $\mathcal{T}_h$ we denote $W^p(K) \equiv \mathcal{P}^p(K)$ and $\boldsymbol{V}^p(K) \equiv (\mathcal{P}^p(K))^d$. We introduce discontinuous finite element spaces

$$W_h^p = \{w \in L^2(\Omega) \ : \ w|_K \in W^p(K)\ \forall K \in \mathcal{T}_h\},$$

and

$$\boldsymbol{V}_h^p = \{\boldsymbol{v} \in (L^2(\Omega))^d \ : \ \boldsymbol{v}|_K \in \boldsymbol{V}^p(K)\ \forall K \in \mathcal{T}_h\}.$$

Here $L^2(D)$ is the space of square integrable functions on D. In addition, we introduce a traced finite element space

$$M_h^p = \{\mu \in L^2(\mathcal{E}_h) \ : \ \mu|_e \in \mathcal{P}^p(e), \forall e \in \mathcal{E}_h\}.$$

We also set $M_h^p(g_D) = \{\mu \in M_h^p : \mu = \mathsf{P}g_D \text{ on } \Gamma_D\}$, where P denotes the L^2-projection into the space $\{\mu|_{\partial\Omega}\ \forall \mu \in M_h^p\}$. 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 w and v in $(L^2(D))^d$, we denote $(\mathsf{w},\mathsf{v})_D = \int_D \mathsf{w}\cdot\mathsf{v}$. 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

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

for functions w, v defined on $\mathcal{T}_h, \zeta, \rho$ defined on $\partial\mathcal{T}_h$, and μ, η defined on $\mathcal{E}_h$.

3. The hybridizable DG method

3.1. Formulation

We seek an approximation $(\mathsf{q}_h, u_h) \in \mathsf{V}_h^p \times W_h^p$ such that for all $K \in \mathcal{T}_h$,

$$\begin{aligned} &(\kappa^{-1}\mathsf{q}_h, \mathsf{v})_K - (u_h, \nabla\cdot\mathsf{v})_K + \langle \hat{u}_h, \mathsf{v}\cdot\mathsf{n}\rangle_{\partial K} = 0, \quad \forall \mathsf{v} \in (\mathcal{P}^p(K))^d, \\ &-(\mathsf{c}u_h + \mathsf{q}_h, \nabla w)_K + \langle (\widehat{\mathsf{c}u}_h + \widehat{\mathsf{q}}_h)\cdot\mathsf{n}, w\rangle_{\partial K} = (f, w)_K, \quad \forall w \in \mathcal{P}^p(K). \end{aligned} \tag{5}$$

Here, the numerical traces $\widehat{\mathsf{c}u}_h + \widehat{\mathsf{q}}_h$ and $\hat{u}_h$ are approximations to $\mathsf{c}u - \kappa\nabla u$ and u over ∂K, respectively. Next, we express (q_h, u_h) in terms of $\hat{u}_h$ only. To this end, we consider numerical traces $\widehat{\mathsf{c}u}_h + \widehat{\mathsf{q}}_h$ of the form

$$\widehat{\mathsf{c}u}_h + \widehat{\mathsf{q}}_h = \mathsf{c}\hat{u}_h + \mathsf{q}_h + \tau(u_h - \hat{u}_h)\mathsf{n}, \quad \text{on } \partial K. \tag{6}$$

Here, τ is the so-called *local stabilization parameter*; it has an important effect on both the stability and accuracy of the resulting scheme. The selection of the value of the parameter τ will be described below. Note that both $\widehat{\mathsf{c}u}_h$ and $\mathsf{c}\hat{u}_h$ are different approximations to the same quantity $\mathsf{c}u$ and that the former is defined in terms of the latter as later shown in Section 3.6.

We next express $\hat{u}_h$ in terms of the boundary data g_D and a new variable $\lambda_h \in M_h^p(0)$ as

$$\hat{u}_h = \begin{cases} \mathsf{P}g_D, & \text{on } \mathcal{E}_h \cap \Gamma_D, \\ \lambda_h, & \text{on } \mathcal{E}_h \setminus \Gamma_D. \end{cases}$$

By adding the contributions of (5) over all the elements and enforcing the continuity of the normal component of the numerical flux, we arrive at the following problem: find an approximation $(\mathsf{q}_h, u_h, \lambda_h) \in \mathsf{V}_h^p \times W_h^p \times M_h^p(0)$ such that

$$\begin{aligned} &(\kappa^{-1}\mathsf{q}_h, \mathsf{v})_{\mathcal{T}_h} - (u_h, \nabla\cdot\mathsf{v})_{\mathcal{T}_h} + \langle \lambda_h, \mathsf{v}\cdot\mathsf{n}\rangle_{\partial\mathcal{T}_h} = -\langle g_D, \mathsf{v}\cdot\mathsf{n}\rangle_{\Gamma_D}, \quad \forall \mathsf{v} \in \mathsf{V}_h^p, \\ &-(\mathsf{c}u_h + \mathsf{q}_h, \nabla w)_{\mathcal{T}_h} + \langle (\widehat{\mathsf{c}u}_h + \widehat{\mathsf{q}}_h)\cdot\mathsf{n}, w\rangle_{\partial\mathcal{T}_h} = (f, w)_{\mathcal{T}_h}, \quad \forall w \in W_h^p, \\ &\langle [\![(\widehat{\mathsf{c}u}_h + \widehat{\mathsf{q}}_h)\cdot\mathsf{n}]\!], \mu\rangle_{\mathcal{E}_h} = \langle g_N, \mu\rangle_{\Gamma_N}, \quad \forall \mu \in M_h^p(0). \end{aligned} \tag{7}$$

Note that the Dirichlet boundary condition has been enforced by requiring that $\hat{u}_h = \mathsf{P}g_D$ on $\mathcal{E}_h \cap \Gamma_D$, whereas the continuity of the normal component of $\widehat{\mathsf{c}u}_h + \widehat{\mathsf{q}}_h$ is enforced explicitly by the last equation. This type of method is sometimes called a hybrid dual-mixed method. It is called mixed because we seek approximations for both q_h and u_h. It is called hybrid dual because the approximate trace, λ_h, associated to the conservativity condition is an approximation for the trace of u_h on the boundaries of the elements.

We observe that λ_h is uniquely defined over each edge since λ_h belongs to M_h^p. Furthermore, if $[\![(\widehat{\mathsf{c}u}_h + \widehat{\mathsf{q}}_h)\cdot\mathsf{n}]\!]$ belongs to M_h^p, then the last Eq. (7) simply states that $[\![(\widehat{\mathsf{c}u}_h + \widehat{\mathsf{q}}_h)\cdot\mathsf{n}]\!] = 0$ pointwise over $\mathcal{E}_h \setminus \Gamma_N$ and that $(\widehat{\mathsf{c}u}_h + \widehat{\mathsf{q}}_h)\cdot\mathsf{n} = \mathsf{P}g_N$ on Γ_N; in other words, the normal component of the numerical trace $\widehat{\mathsf{c}u}_h + \widehat{\mathsf{q}}_h$ is single-valued. Hence, both λ_h and $\widehat{\mathsf{c}u}_h + \widehat{\mathsf{q}}_h$ are conservative fluxes according to the definition in [1]. Note that our numerical traces remain conservative even when the diffusion coefficient κ is discontinuous at the interior element interface.

Let us point out that the difference between this HDG method and the one considered in [8] is that therein the total flux $\mathsf{c}u + \mathsf{q}$ is approximated by a single variable whereas here we approximate its two components separately. As a result, our method can deal with the purely convective case, unlike the approach presented in [8]. Finally, we note that, due to the discontinuous nature of both V_h^p and W_h^p, the first two equations in (7) can be used to eliminate both q_h and u_h to obtain a weak formulation in terms of λ_h only and thus a global system of equations involving the degrees of freedom of λ_h, as described below.

3.2. Implementation

We first insert (6) into (7) and obtain, after some simple manipulations, that $(\mathsf{q}_h, u_h, \lambda_h) \in \mathsf{V}_h^p \times W_h^p \times M_h^p(0)$ is the solution of the following weak formulation:

$$\begin{aligned} &a(\mathsf{q}_h, \mathsf{v}) - b(u_h, \mathsf{v}) + c(\lambda_h, \mathsf{v}) = r(\mathsf{v}), \\ &b(w, \mathsf{q}_h) + d(u_h, w) + e(\lambda_h, w) = f(w), \\ &c(\mu, \mathsf{q}_h) + g(\mu, u_h) + h(\mu, \lambda_h) = \ell(\mu), \end{aligned} \tag{8}$$

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

$$
\begin{aligned}
&a(\mathsf{q},\mathsf{v}) = (\kappa^{-1}\mathsf{q},\mathsf{v})_{\mathcal{T}_h},\\
&b(u,\mathsf{v}) = (u,\nabla\cdot\mathsf{v})_{\mathcal{T}_h},\\
&c(\lambda,\mathsf{v}) = \langle\lambda,\mathsf{v}\cdot\mathsf{n}\rangle_{\partial\mathcal{T}_h},\\
&d(u,w) = -(\mathsf{c}u,\nabla w)_{\mathcal{T}_h} + \langle w,\tau u\rangle_{\partial\mathcal{T}_h},\\
&e(\lambda,w) = \langle w,(\mathsf{c}\cdot\mathsf{n}-\tau)\lambda\rangle_{\partial\mathcal{T}_h},\\
&g(\mu,u) = \langle\mu,\tau u\rangle_{\partial\mathcal{T}_h},\\
&h(\mu,\lambda) = \langle\mu,(\mathsf{c}\cdot\mathsf{n}-\tau)\lambda\rangle_{\partial\mathcal{T}_h},\\
&f(w) = (f,w)_{\mathcal{T}_h},\\
&r(\mathsf{v}) = -\langle g_D,\mathsf{v}\cdot\mathsf{n}\rangle_{\Gamma_D},\\
&\ell(\mu) = \langle g_N,\mu\rangle_{\Gamma_N},
\end{aligned}
\tag{9}
$$

for all (q,u,λ) and (v,w,μ) in $\mathsf{V}_h^p\times W_h^p\times M_h^p$.

The discretization of the system of equations (8) gives rise to a matrix equation of the form

$$
\begin{bmatrix} A & -B^T & C^T\\ B & D & E\\ C & G & H\end{bmatrix}\begin{bmatrix}\mathsf{Q}\\ U\\ \Lambda\end{bmatrix} = \begin{bmatrix} R\\ F\\ L\end{bmatrix}. \tag{10}
$$

Here Q, U, and Λ represent the vector of degrees of freedom for q_h, u_h, and λ_h, respectively. The matrices in (10) correspond to the bilinear forms in (8) in the order they appear in the equations.

As pointed out above, the first two equations in (7) can be used to eliminate both q_h and u_h in an element-by-element fashion. As a consequence, we can write the above system as

$$
\begin{bmatrix}\mathsf{Q}\\ U\end{bmatrix} = \begin{bmatrix} A & -B^T\\ B & D\end{bmatrix}^{-1}\left(\begin{bmatrix} R\\ F\end{bmatrix} - \begin{bmatrix} C^T\\ E\end{bmatrix}\Lambda\right), \tag{11a}
$$

and

$$
C\mathsf{Q} + GU + H\Lambda = L. \tag{11b}
$$

Let us stress once again that the above inverse can be computed on each element independently of each other owing to the discontinuous nature of the approximation spaces W_h^p and V_h^p. Moreover, the inverse matrix is well defined and bock-diagonal since the it results from applying the LDG method to solve the original PDE (3) with Dirichlet conditions at each element [10].

Now, we eliminate both Q and U to obtain a reduced globally coupled matrix equation only for Λ as

$$
\mathbb{K}\,\Lambda = \mathbb{F}, \tag{12a}
$$

where

$$
\mathbb{K} = -\begin{bmatrix} C & G\end{bmatrix}\begin{bmatrix} A & -B^T\\ B & D\end{bmatrix}^{-1}\begin{bmatrix} C^T\\ E\end{bmatrix} + H, \tag{12b}
$$

and

$$
\mathbb{F} = L - \begin{bmatrix} C & G\end{bmatrix}\begin{bmatrix} A & -B^T\\ B & D\end{bmatrix}^{-1}\begin{bmatrix} R\\ F\end{bmatrix}. \tag{12c}
$$

Once Λ is available both Q and U can be obtained from (11a).

3.3. Characterization of the numerical trace λ_h

Let us now shed light into the nature and structure of the matrix Eq. (12a). To do that, let us introduce the so-called local solver. Thus, we associate to each function $(\mathsf{m},f)\in M_h^p\times L^2(\Omega)$, the pair $(\mathsf{q}_h^{\mathsf{m},f},u_h^{\mathsf{m},f})$ on Ω whose restriction to each element K is in $\mathsf{V}^p(K)\times W^p(K)$ and satisfies

$$
(\kappa^{-1}\mathsf{q}_h^{\mathsf{m},f},\mathsf{v})_K - (u_h^{\mathsf{m},f},\nabla\cdot\mathsf{v})_K = -\langle\mathsf{m},\mathsf{v}\cdot\mathsf{n}\rangle_{\partial K}, \tag{13a}
$$

$$
-(\mathsf{c}u_h^{\mathsf{m},f} + \mathsf{q}_h^{\mathsf{m},f},\nabla w)_K + \langle(\widehat{\mathsf{c}u}_h^{\mathsf{m},f} + \widehat{\mathsf{q}}_h^{\mathsf{m},f})\cdot\mathsf{n},w\rangle_{\partial K} = (f,w)_K, \tag{13b}
$$

for all $(\mathsf{v},w)\in\mathsf{V}^p\times W^p$, where

$$
\widehat{\mathsf{c}u}_h^{\mathsf{m},f} + \widehat{\mathsf{q}}_h^{\mathsf{m},f} = \mathsf{c}\mathsf{m} + \mathsf{q}_h^{\mathsf{m},f} + \tau(u_h^{\mathsf{m},f} - \mathsf{m})\mathsf{n}. \tag{13c}
$$

It is now clear, see equations (7), that the approximate solution $(\mathsf{q}_h, u_h) \in \mathbb{V}_h^p \times W_h^p$ satisfies

$$\mathsf{q}_h = \mathsf{q}_h^{\lambda_h f}, \quad u_h = u_h^{\lambda_h f}, \tag{14a}$$

where $\lambda_h \in M_h^p(0)$ is such that

$$\langle [\![(\widehat{\mathsf{c}u}_h^{\lambda_h f} + \widehat{\mathsf{q}}_h^{\lambda_h f}) \cdot \mathsf{n}]\!], \mu \rangle_{\mathcal{E}_h} = \langle g_N, \mu \rangle_{\Gamma_N}, \quad \forall \mu \in M_h^p(0). \tag{14b}$$

We can now see that Eq. (11a) is the matrix equation associated to the weak formulation (14a), and that Eq. (11b) is the matrix equation associated to the weak formulation (14b). We next show that we can eliminate q_h and u_h from the above equations to obtain a weak formulation in terms of λ_h only.

Let $\left(\mathsf{q}_h^{\mathsf{m},0}, u_h^{\mathsf{m},0}\right)$ (respectively, $\left(\mathsf{q}_h^{0,f}, u_h^{0,f}\right)$) solve (13b) when we set $f = 0$ (respectively, $\mathsf{m} = 0$). If, for all η and $\mu \in M_h^p$, we set

$$a_h(\eta, \mu) = -\langle [\![(\widehat{\mathsf{c}u}_h^{\eta,0} + \widehat{\mathsf{q}}_h^{\eta,0}) \cdot \mathsf{n}]\!], \mu \rangle_{\mathcal{E}_h},$$
$$b_h(\mu) = \langle [\![(\widehat{\mathsf{c}u}_h^{0,f} + \widehat{\mathsf{q}}_h^{0,f}) \cdot \mathsf{n}]\!], \mu \rangle_{\mathcal{E}_h},$$

we have from (14b) and linearity of the problem (13b) that the function $\lambda_h \in M_h^p(0)$ is the solution of the variational formulation

$$a_h(\lambda_h, \mu) = b_h(\mu) - \langle g_N, \mu \rangle_{\Gamma_N}, \quad \forall \mu \in M_h^p(0). \tag{15}$$

Thus, $\mathbb{K}$ is the matrix associated with the bilinear form $a_h(\cdot,\cdot)$ and $\mathbb{F}$ the matrix associated with the linear form $b_h(\cdot) - \langle g_N, \cdot \rangle_{\Gamma_N}$. Note that since

$$a_h(\eta, \mu) = -\langle (\widehat{\mathsf{c}u}_h^{\eta,0} + \widehat{\mathsf{q}}_h^{\eta,0}) \cdot \mathsf{n}, \mu \rangle_{\partial \mathcal{T}_h},$$

we can easily deduce that if the support of η is the interior face $e = \partial K^+ \cap \partial K^-$, or the boundary face $e = \partial K \cap \partial \Omega$, then $a_h(\eta, \mu) = 0$ when the support of μ does not intersect $\partial K^+ \cup \partial K^-$, or ∂K, respectively. Thus, the matrix $\mathbb{K}$ has a block-structure of blocks of square matrices of order $\dim \mathcal{P}^p$. In eack block-row or block-column, there are at most five non-zero blocks when the elements are triangles, and at most seven non-zero blocks in three space dimension.

It remains to prove the well-posedness of the local solver.

Lemma 3.1. *Assume that $\nabla \cdot \mathsf{c} \geqslant 0$ in Ω. Then, if the stabilization parameter τ satisfies the condition*

$$\tau > \frac{1}{2} \mathsf{c} \cdot \mathsf{n} \quad \text{on } \partial \mathcal{T}_h, \tag{16}$$

we have that the solution $\left(\mathsf{q}_h^{\mathsf{m},f}, u_h^{\mathsf{m},f}\right)$ of the local solver (13b) exists and is unique.

Proof 1. By integration by parts, we can write (13b) as

$$(\kappa^{-1} \mathsf{q}_h^{\mathsf{m},f}, \mathsf{v})_K - (u_h^{\mathsf{m},f}, \nabla \cdot \mathsf{v})_K = -\langle \mathsf{m}, \mathsf{v} \cdot \mathsf{n} \rangle_{\partial K}, \tag{17a}$$

$$(\nabla \cdot \mathsf{q}_h^{\mathsf{m},f}, w)_K - (\mathsf{c} u_h^{\mathsf{m},f}, \nabla w)_K + \langle \tau u_h^{\mathsf{m},f}, w \rangle_{\partial K} = (f, w)_K + \langle (\tau - \mathsf{c} \cdot \mathsf{n}) \mathsf{m}, w \rangle_{\partial K}, \tag{17b}$$

for all $(\mathsf{v}, w) \in \mathbb{V}^p(K) \times W^p(K)$. Due to the linearity, finite dimensionality, and to the fact that this is a square system, it is sufficient to show that the only solution of (17b) for $\mathsf{m} = 0$ and $f = 0$ is $\mathsf{q}_h^{\mathsf{m},f} = 0$ and $u_h^{\mathsf{m},f} = 0$. Indeed, taking $\mathsf{v} = \mathsf{q}_h^{\mathsf{m},f}$ and $w = u_h^{\mathsf{m},f}$ and adding the two equations, we get

$$(\kappa^{-1} \mathsf{q}_h^{\mathsf{m},f}, \mathsf{q}_h^{\mathsf{m},f})_K - (\mathsf{c} u_h^{\mathsf{m},f}, \nabla u_h^{\mathsf{m},f})_K + \langle \tau u_h^{\mathsf{m},f}, u_h^{\mathsf{m},f} \rangle_K = 0,$$

which, after integrating by parts, yields

$$\left(\kappa^{-1} \mathsf{q}_h^{\mathsf{m},f}, \mathsf{q}_h^{\mathsf{m},f} \right)_K + \frac{1}{2} \left((\nabla \cdot \mathsf{c})\, u_h^{\mathsf{m},f}, u_h^{\mathsf{m},f} \right)_K + \left\langle (\tau - \frac{1}{2} \mathsf{c} \cdot \mathsf{n}) u_h^{\mathsf{m},f}, u_h^{\mathsf{m},f} \right\rangle_K = 0.$$

This equation implies that $\mathsf{q}_h^{\mathsf{m},f} = 0$ over the simplex K and that $u_h^{\mathsf{m},f} = 0$ on ∂K since we assumed $\nabla \cdot \mathsf{c} \geqslant 0$ and $\tau > \frac{1}{2} \mathsf{c} \cdot \mathsf{n}$. It thus follows from (17a) that

$$\left(\nabla u_h^{\mathsf{m},f}, \mathsf{v} \right)_K = 0, \quad \forall \mathsf{v} \in \mathbb{V}^p(K).$$

Since $u_h^{\mathsf{m},f} \in W^p(K)$ the above equation implies $\nabla u_h^{\mathsf{m},f}$ is constant over K. As a consequence, $u_h^{\mathsf{m},f} = 0$ over K since $u_h^{\mathsf{m},f} = 0$ on ∂K. This completes the proof. □

Therefore, for the uniqueness and thus also stability of the numerical solution we only require τ to satisfy the condition (16) – not necessarily the fully upwinding condition $\tau \geqslant \mathsf{c} \cdot \mathsf{n}$; see also [15].

3.4. A sufficient condition for the existence and uniqueness of λ_h

Next, we show that Eq. (15) uniquely determines λ_h and, by (14a) and Lemma 3.1, q_h and u_h.

Theorem 3.1. *Assume that $\nabla\cdot\mathsf{c}\geqslant 0$ in Ω. Then, if the stabilization parameter τ satisfies the condition (16) we have that the solution λ_h of the variational formulation (15) exists and is unique.*

To prove this theorem, we are going to use the following auxiliary result.

Lemma 3.2. *We have, for all $\eta,\mu\in M_h^p$, that*

$$a_h(\eta,\mu)=\left(\kappa^{-1}\mathsf{q}_h^{\mu,0},\mathsf{q}_h^{\eta,0}\right)_{\mathcal{T}_h}-(\mathsf{c}u_h^{\eta,0},\nabla u_h^{\mu,0})_{\mathcal{T}_h}+\left\langle u_h^{\mu,0}-\mu,\mathsf{c}\cdot\mathsf{n}\,u_h^{\eta,0}\right\rangle_{\mathcal{T}_h}+\left\langle u_h^{\mu,0}-\mu,(\tau-\mathsf{c}\cdot\mathsf{n})\,(u_h^{\eta,0}-\eta)\right\rangle_{\mathcal{T}_h},$$

Proof 2. To prove this result, we rewrite the equations (13b) with $f=0$ as follows:

$$\left(\kappa^{-1}\mathsf{q}_h^{\mu,0},\mathsf{v}\right)_{\mathcal{T}_h}-(u_h^{\mu,0},\nabla\cdot\mathsf{v})_{\mathcal{T}_h}=-\langle\mu,\mathsf{v}\cdot\mathsf{n}\rangle_{\partial\mathcal{T}_h},\tag{18}$$

$$\left(\nabla\cdot\left(\mathsf{c}u_h^{\eta,0}+\mathsf{q}_h^{\eta,0}\right),w\right)_{\mathcal{T}_h}=-\left\langle w,(\tau-\mathsf{c}\cdot\mathsf{n})\left(u_h^{\eta,0}-\eta\right)\right\rangle_{\partial\mathcal{T}_h},\tag{19}$$

for all $(\mathsf{v},w)\in\mathsf{V}_h^p\times W_h^p$. The first equation is obtained from the first equation in (13b) by simply replacing f by 0 and summing over the elements. The second is obtained from the second equation in (13b) by replacing μ by η, f by 0, integrating by parts, summing the resulting equations over all elements and inserting the definition of the numerical trace (13c).

Thus, we have

$$a_h(\eta,\mu)=-\left\langle\mu,[\![(\widehat{\mathsf{c}u}_h^{\eta,0}+\widehat{\mathsf{q}}_h^{\eta,0})\cdot\mathsf{n}]\!]\right\rangle_{\mathcal{E}_h}=-\left\langle\mu,(\widehat{\mathsf{c}u}_h^{\eta,0}+\widehat{\mathsf{q}}_h^{\eta,0})\cdot\mathsf{n}\right\rangle_{\partial\mathcal{T}_h}=-\left\langle\mu,\mathsf{c}\cdot\mathsf{n}\,\eta+\mathsf{q}_h^{\eta,0}\cdot\mathsf{n}\right\rangle_{\mathcal{T}_h}-\left\langle\mu,\tau\,(u_h^{\eta,0}-\eta)\right\rangle_{\mathcal{T}_h},$$

by (13c). Rearranging terms, we get

$$a_h(\eta,\mu)=-\left\langle\mu,\mathsf{q}_h^{\eta,0}\cdot\mathsf{n}\right\rangle_{\mathcal{T}_h}-\left\langle\mu,\mathsf{c}\cdot\mathsf{n}\,u_h^{\eta,0}\right\rangle_{\mathcal{T}_h}-\left\langle\mu,(\tau-\mathsf{c}\cdot\mathsf{n})\left(u_h^{\eta,0}-\eta\right)\right\rangle_{\mathcal{T}_h},$$

and, by (18) with $\mathsf{v}=q_h^{\eta,0}$,

$$a_h(\eta,\mu)=\left(\kappa^{-1}\mathsf{q}_h^{\mu,0},\mathsf{q}_h^{\eta,0}\right)_{\mathcal{T}_h}-\left(u_h^{\mu,0},\nabla\cdot\mathsf{q}_h^{\eta,0}\right)_{\mathcal{T}_h}-\left\langle\mu,\mathsf{c}\cdot\mathsf{n}\,u_h^{\eta,0}\right\rangle_{\mathcal{T}_h}-\left\langle\mu,(\tau-\mathsf{c}\cdot\mathsf{n})\left(u_h^{\eta,0}-\eta\right)\right\rangle_{\mathcal{T}_h}.$$

Finally, by (19) with $w=u_h^{\mu,0}$,

$$a_h(\eta,\mu)=\left(\kappa^{-1}\mathsf{q}_h^{\mu,0},\mathsf{q}_h^{\eta,0}\right)_{\mathcal{T}_h}+\left(\nabla\cdot\left(\mathsf{c}u_h^{\eta,0}\right),u_h^{\mu,0}\right)_{\mathcal{T}_h}-\left\langle\mu,\mathsf{c}\cdot\mathsf{n}\,u_h^{\eta,0}\right\rangle_{\mathcal{T}_h}+\left\langle u_h^{\mu,0}-\mu,(\tau-\mathsf{c}\cdot\mathsf{n})\left(u_h^{\eta,0}-\eta\right)\right\rangle_{\mathcal{T}_h},$$

and the result follows by integration by parts. This completes the proof. □

We are now ready to prove Theorem 3.1.

Proof 3. The existence and uniqueness of λ_h follows if we show that the only solution $\eta\in M_h^p(0)$ of the problem

$$a_h(\eta,\mu)=0,\quad\forall\mu\in M_h^p(0),$$

is $\eta=0$. Taking $\mu=\eta$, by the previous lemma, we then have that

$$0=\left(\kappa^{-1}\mathsf{q}_h^{\eta,0},\mathsf{q}_h^{\eta,0}\right)_{\mathcal{T}_h}-\left(\mathsf{c}u_h^{\eta,0},\nabla u_h^{\eta,0}\right)_{\mathcal{T}_h}+\left\langle u_h^{\eta,0}-\eta,\mathsf{c}\cdot\mathsf{n}\,u_h^{\eta,0}\right\rangle_{\mathcal{T}_h}+\left\langle u_h^{\eta,0}-\eta,(\tau-\mathsf{c}\cdot\mathsf{n})\left(u_h^{\eta,0}-\eta\right)\right\rangle_{\mathcal{T}_h},$$

and, by integration by parts,

$$\begin{aligned}0&=\left(\kappa^{-1}\mathsf{q}_h^{\eta,0},\mathsf{q}_h^{\eta,0}\right)_{\mathcal{T}_h}+\frac{1}{2}\left(\nabla\cdot\mathsf{c},\left(u_h^{\eta,0}\right)^2\right)_{\mathcal{T}_h}-\frac{1}{2}\left\langle u_h^{\eta,0},\mathsf{c}\cdot\mathsf{n}\,u_h^{\eta,0}\right\rangle_{\partial\mathcal{T}_h}+\left\langle u_h^{\eta,0}-\eta,\mathsf{c}\cdot\mathsf{n}\,u_h^{\eta,0}\right\rangle_{\mathcal{T}_h}\\&\quad+\left\langle u_h^{\eta,0}-\eta,(\tau-\mathsf{c}\cdot\mathsf{n})\left(u_h^{\eta,0}-\eta\right)\right\rangle_{\partial\mathcal{T}_h}\\&=\left(\kappa^{-1}\mathsf{q}_h^{\eta,0},\mathsf{q}_h^{\eta,0}\right)_{\mathcal{T}_h}+\frac{1}{2}\left(\nabla\cdot\mathsf{c},\left(u_h^{\eta,0}\right)^2\right)_{\mathcal{T}_h}+\frac{1}{2}\left\langle u_h^{\eta,0}-\eta,\mathsf{c}\cdot\mathsf{n}\left(u_h^{\eta,0}-\eta\right)\right\rangle_{\partial\mathcal{T}_h}-\frac{1}{2}\langle\eta,\mathsf{c}\cdot\mathsf{n}\,\eta\rangle_{\partial\mathcal{T}_h}\\&\quad+\left\langle u_h^{\eta,0}-\eta,(\tau-\mathsf{c}\cdot\mathsf{n})\left(u_h^{\eta,0}-\eta\right)\right\rangle_{\partial\mathcal{T}_h},\end{aligned}$$

after some simple algebraic manipulations. Finally, since $\eta\in M_h^p(0)$, we have that $\langle\eta,\mathsf{c}\cdot\mathsf{n}\,\eta\rangle_{\partial\mathcal{T}_h}=0$ and we get

$$0=\left(\kappa^{-1}\mathsf{q}_h^{\eta,0},\mathsf{q}_h^{\eta,0}\right)_{\mathcal{T}_h}+\frac{1}{2}\left(\nabla\cdot\mathsf{c},\left(u_h^{\eta,0}\right)^2\right)_{\mathcal{T}_h}+\left\langle u_h^{\eta,0}-\eta,\left(\tau-\frac{1}{2}\mathsf{c}\cdot\mathsf{n}\right)\left(u_h^{\eta,0}-\eta\right)\right\rangle_{\partial\mathcal{T}_h}.$$

As a consequence, we can conclude that $\mathsf{q}_h^{\eta,0} = 0$ and $u_h^{\eta,0} = \eta$, since we assumed that $\nabla \cdot \mathsf{c} \geqslant 0$ and $\tau > \frac{1}{2}\mathsf{c} \cdot \mathsf{n}$.

After a simple integration by parts, Eq. (18), with μ replaced by η, now reads

$$\left(\nabla u_h^{\eta,0}, \mathsf{v}\right)_{\mathcal{T}_h} = 0, \quad \forall \mathsf{v} \in \mathsf{V}_h^p.$$

This implies that $u_h^{\eta,0}$ is a constant on Ω. As a consequence η is also a constant on $\mathcal{E}_h$ and, since $\eta \in M_h(0)$, we must have that $\eta = 0$ on $\mathcal{E}_h$. This completes the proof of Theorem 3.1. □

3.5. The general form of the numerical traces

In order to derive an explicit expression for the numerical traces in terms of (u_h, q_h), we proceed as follows. Since the conservativity condition implies $[\![(\widehat{\mathsf{c}u_h} + \widehat{\mathsf{q}}_h) \cdot \mathsf{n}]\!] = 0$ pointwise, we have, using expression (6), that

$$[\![\mathsf{q}_h \cdot \mathsf{n}]\!] + \tau^+ u_h^+ + \tau^- u_h^- - (\tau^+ + \tau^-)\lambda_h = 0, \quad \text{on } \mathcal{E}_h^o.$$

Solving for λ_h and inserting the result into the expression for $\widehat{\mathsf{c}u_h} + \widehat{\mathsf{q}}_h$ (6), we obtain on $\mathcal{E}_h^o$

$$\begin{aligned} \lambda_h &= \frac{\tau^+}{\tau^+ + \tau^-} u_h^+ + \frac{\tau^-}{\tau^+ + \tau^-} u_h^- + \left(\frac{1}{\tau^+ + \tau^-}\right)[\![\mathsf{q}_h \cdot \mathsf{n}]\!], \\ \widehat{\mathsf{c}u}_h + \widehat{\mathsf{q}}_h &= \mathsf{c}\lambda_h + \frac{\tau^-}{\tau^+ + \tau^-}\mathsf{q}_h^+ + \frac{\tau^+}{\tau^+ + \tau^-}\mathsf{q}_h^- + \left(\frac{\tau^+\tau^-}{\tau^+ + \tau^-}\right)[\![u_h \mathsf{n}]\!]. \end{aligned} \tag{20}$$

These expressions for the numerical traces highlight the relationship between the HDG method and the more standard DG methods, as discussed below.

3.6. The local stabilization parameter τ

In order to get some guidance for choosing the stabilization parameter τ, we consider the two limiting cases where only convection or diffusion occur.

3.6.1. Convective limit

In the convective limit we have $\kappa = 0$ and consequently $\mathsf{q}_h = 0$. In this case, the expressions (20) become

$$\begin{aligned} \lambda_h &= \frac{\tau^+}{\tau^+ + \tau^-} u_h^+ + \frac{\tau^-}{\tau^+ + \tau^-} u_h^-, \\ \widehat{\mathsf{c}u}_h \cdot \mathsf{n}^+ &= \frac{\tau^+}{\tau^+ + \tau^-}(\mathsf{c} \cdot \mathsf{n}^+ + \tau^-) u_h^+ + \frac{\tau^-}{\tau^+ + \tau^-}(\mathsf{c} \cdot \mathsf{n}^+ - \tau^+) u_h^-, \end{aligned} \tag{21}$$

where we have multiplied the second equation by n^+ to obtain the normal flux. For the flux to be fully upwinded, we want

$$\widehat{\mathsf{c}u}_h \cdot \mathsf{n}^+ = \begin{cases} (\mathsf{c} \cdot \mathsf{n}^+) u_h^+ & \text{if } \mathsf{c} \cdot \mathsf{n}^+ \geqslant 0, \\ (\mathsf{c} \cdot \mathsf{n}^+) u_h^- & \text{if } \mathsf{c} \cdot \mathsf{n}^+ < 0 \end{cases} \tag{22}$$

or, equivalently,

$$\widehat{\mathsf{c}u}_h \cdot \mathsf{n}^+ = \frac{1}{2}(\mathsf{c} \cdot \mathsf{n}^+ + |\mathsf{c} \cdot \mathsf{n}^+|) u_h^+ + \frac{1}{2}(\mathsf{c} \cdot \mathsf{n}^+ - |\mathsf{c} \cdot \mathsf{n}^+|) u_h^-. \tag{23}$$

Comparing the desired expression (23) with Eq. (21), we observe that *two* choices for the parameter τ are possible. Either, $\tau^\pm = |\mathsf{c} \cdot \mathsf{n}|$ or $\tau^\pm = \alpha^\pm(|\mathsf{c} \cdot \mathsf{n}^+| \pm \mathsf{c} \cdot \mathsf{n}^+)$, where α^+ and α^- are two positive constants. Note that in both cases, the condition in Theorem 3.1 on the stabilization parameter τ, (16), is satified; so the existence and uniqueness of the approximate solution is guaranteed.Note also that for the first choice we get the *centered* trace $\lambda_h = \{\{u_h\}\} = (u_h^+ + u_h^-)/2$ whereas for the second choice we obtain the *upwinded* trace, namely, $\lambda_h = u_h^+$ if $\mathsf{c} \cdot \mathsf{n}^+ > 0$ and $\lambda_h = u_h^-$ otherwise. Note that for the second choice, the resulting DG method is nothing but the original DG method [19], whereas for the first choice, the resulting HDG scheme seems to be new.

3.6.2. Diffusive limit

In the diffusive limit $\mathsf{c} = 0$, expressions (20) become

$$\begin{aligned} \lambda_h &= \frac{\tau^+}{\tau^+ + \tau^-} u_h^+ + \frac{\tau^-}{\tau^+ + \tau^-} u_h^- + \left(\frac{1}{\tau^+ + \tau^-}\right)[\![\mathsf{q}_h \cdot \mathsf{n}]\!], \\ \widehat{\mathsf{q}}_h &= \frac{\tau^-}{\tau^+ + \tau^-}\mathsf{q}_h^+ + \frac{\tau^+}{\tau^+ + \tau^-}\mathsf{q}_h^- + \left(\frac{\tau^+\tau^-}{\tau^+ + \tau^-}\right)[\![u_h \mathsf{n}]\!]. \end{aligned} \tag{24}$$

This case has been originally studied in [5]; see also [7,10,11]. By rearranging terms these expressions can be transformed into the more standard form considered in [5],

$$\begin{aligned}\widehat{\mathsf{q}}_h &= \{\{\mathsf{q}_h\}\} + C_{11}[\![u_h \mathsf{n}]\!] + C_{12}[\![\mathsf{q}_h \cdot \mathsf{n}]\!],\\ \lambda_h = \hat{u}_h &= \{\{u_h\}\} - C_{12} \cdot [\![u_h \mathsf{n}]\!] + C_{22}[\![\mathsf{q}_h \cdot \mathsf{n}]\!],\end{aligned} \tag{25}$$

where

$$C_{11} = \left(\frac{\tau^+\tau^-}{\tau^+ + \tau^-}\right), \quad C_{12} = \frac{1}{2}\left(\frac{[\![\tau \mathsf{n}]\!]}{\tau^+ + \tau^-}\right), \quad C_{22} = \left(\frac{1}{\tau^+ + \tau^-}\right).$$

Therein, it was shown that the method is well defined provided $C_{11} > 0$ and $C_{22} \geqslant 0$, which is a condition implies by the condition (16) in Theorem 3.1, namely, $\tau > 0$. We point out that in the LDG method, the trace λ_h is chosen to be independent of q_h, that is $C_{22} = 0$. This has the advantage of allowing the degrees of freedom associated with the q_h to be locally eliminated and a global system involving only the degrees of freedom associated to u_h is thus solved.

It is interesting to note that for some choices of the parameter $\tau^\pm$ one can obtain schemes that converge at a rate of $p+1$ for both the scalar variable and the flux, and that they display superconvergence properties of the scalar variable; see [7,11] for symmetric problems and [8] for convection–diffusion-reaction problems. It is also shown [11] that these superconvergent schemes require that C_{22} be non-zero. While this presents a serious inconvenience for LDG methods, for HDG methods this represents no difficulty. Choosingτ equal to zero in d faces of the simplex while taking τ to be of order one in the remaining face, gives superconvergence; see [7]. It turns out that the simple choice of $\tau^\pm$ of order unity everywhere also results in a scheme that possesses the above-mentioned superconvergence properties; see [11].

3.6.3. Some practical choices for τ

Guided by the desired values for τ in the above limiting cases, we now give two expressions that generalize the above expressions over the convection–diffusion range. Both of them satisfy the condition (16) in Theorem 3.1. To account for the diffusion and convection effects our local stabilization parameter τ will take the following form:

$$\tau = \tau_d + \tau_c,$$

where τ_d and τ_c are the local stabilization parameters related to the diffusion and convection, respectively. This allows us to write each component of the numerical trace $\widehat{\mathsf{q}}_h + \widehat{c u}_h$ as

$$\begin{aligned}\widehat{\mathsf{q}}_h &= \mathsf{q}_h + \tau_d(u_h - \lambda_h)\mathsf{n},\\ \widehat{c u}_h &= \mathsf{c}\lambda_h + \tau_c(u_h - \lambda_h)\mathsf{n}.\end{aligned}$$

The expressions for both τ_c and τ_d are given below.

Centered scheme. The first option considered is to take on each edge $\tau_c^+ = \tau_c^- = \eta_c$ and $\tau_d^+ = \tau_d^- = \eta_d$, where

$$\eta_c = |\mathsf{c} \cdot \mathsf{n}|, \quad \eta_d = \frac{\kappa}{\ell}, \tag{26}$$

where ℓ denotes a representative diffusive length scale. In this case, the expressions for the numerical traces becomes

$$\begin{aligned}\lambda_h &= \{\{u_h\}\} + \frac{1}{2\tau}[\![\mathsf{q}_h \cdot \mathsf{n}]\!],\\ \widehat{c u}_h + \widehat{\mathsf{q}}_h &= \mathsf{c}\,\lambda_h + \{\{\mathsf{q}_h\}\} + \frac{\tau}{2}[\![u_h \mathsf{n}]\!].\end{aligned}$$

Clearly, our stabilization parameter τ is a real-valued function on the element edges since it depends on c and κ. Furthermore, it is single-valued since $\tau^+ = \tau^-$. We note from the above expression that the centered scheme is a Lax–Friedrich type flux.

Upwinded Scheme. Here, we choose $\tau_c^\pm$ and $\tau_d^\pm$ according to the following expression

$$(\tau_c^\pm, \tau_d^\pm) = (\eta_c, \eta_d)\ \frac{|\mathsf{c} \cdot \mathsf{n}^+| \pm \mathsf{c} \cdot \mathsf{n}^+}{2|\mathsf{c} \cdot \mathsf{n}^+|}, \tag{27}$$

where η_c and η_d are given by Eq. (26). In this case, the numerical fluxes are given by

$$\begin{cases}\lambda_h = u_h^+ + \tau^{-1}[\![\mathsf{q}_h \cdot \mathsf{n}]\!],\\ \widehat{c u}_h + \widehat{\mathsf{q}}_h = \mathsf{c}\lambda_h + \mathsf{q}_h^-,\end{cases} \quad \text{if } \mathsf{c} \cdot \mathsf{n}^+ \geqslant 0,$$

and

$$\begin{cases}\lambda_h = u_h^- + \tau^{-1}[\![\mathsf{q}_h \cdot \mathsf{n}]\!],\\ \widehat{c u}_h + \widehat{\mathsf{q}}_h = \mathsf{c}\lambda_h + \mathsf{q}_h^+,\end{cases} \quad \text{if } \mathsf{c} \cdot \mathsf{n}^+ < 0.$$

Note that unlike the centered scheme, the stabilization parameter τ of the upwinded scheme is double-valued since $\tau^+ \neq \tau^-$.

As pointed out in [10], the above numerical traces are not those of an LDG methodfor any $\tau \in [0, \infty)$. The only way to recover an LDG method is to formally set one branch of τ equal to infinity for every interior face, while maintaining a fixed finite value on the other branch of τ; see Corollary 3.3 of [10]. However, this comes at the expense of losing the supercon-

vergence properties of the method. The minimum dissipation LDG method [6] is a special case of the HDG method when we let τ^+ go to infinity and set τ^- to be zero.

3.7. Extension to time-dependent problems

We end this section by extending the hybridizable DG method for time-dependent convection–diffusion problems. We consider the time-dependent convection–diffusion model written as a system of first-order equations

$$\begin{aligned}
&\mathrm{q} + \kappa\nabla u = 0, \quad \text{in } \Omega\times(0,T],\\
&\frac{\partial u}{\partial t} + \nabla\cdot(\mathrm{c}u + \mathrm{q}) = f, \quad \text{in } \Omega\times(0,T],\\
&u = g_D, \quad \text{on } \Gamma_D\times(0,T],\\
&(\mathrm{q}+\mathrm{c}u)\cdot\mathrm{n} = g_N, \quad \text{on } \Gamma_N\times(0,T],\\
&u = u_0, \quad \text{in } \Omega \text{ for } t = 0.
\end{aligned} \tag{28}$$

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

$$\begin{aligned}
&(\kappa^{-1}\mathrm{q}_h,\mathrm{v})_K - (u_h,\nabla\cdot\mathrm{v})_K + \langle\widehat{u}_h,\mathrm{v}\cdot\mathrm{n}\rangle_{\partial K} = 0,\\
&\left(\frac{\partial u_h}{\partial t},w\right)_K - (\mathrm{c}u_h+\mathrm{q}_h,\nabla w)_K + \langle(\widehat{\mathrm{c}u}_h+\widehat{\mathrm{q}}_h)\cdot\mathrm{n},w\rangle_{\partial K} = (f,w)_K,
\end{aligned} \tag{29}$$

for all $(\mathrm{v},w)\in(\mathcal{P}^p(K))^d\times\mathcal{P}^p(K)$ and for all $t\in(0,T]$. Here, the numerical traces $\widehat{\mathrm{c}u}_h+\widehat{\mathrm{q}}_h$ and $\widehat{u}_h$ are approximations to $\mathrm{c}u-\kappa\nabla u$ and u over ∂K, respectively.

The above HDG formulation (29) can then be discretized in time using an appropriate time-stepping scheme. Here we consider backward difference formulaes (BDF) for the discretization of the time derivative. For instance, using the Backward–Euler scheme at time-level t^k with time-step Δt^k we obtain the following system:

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

for all $(\mathrm{v},w)\in(\mathcal{P}^p(K))^d\times\mathcal{P}^p(K)$. Here we denote $u_h^k = u_h(t^k)$ and $\mathrm{q}_h^k = \mathrm{q}_h(t^k)$, and u_h^0 as the L^2 projection of u_0 into W_h^p.

The HDG method then seeks an approximation $(\mathrm{q}_h^k,u_h^k,\lambda_h^k)\in\mathrm{V}_h^p\times W_h^p\times M_h^p(0)$ such that

$$\begin{aligned}
&\left(\kappa^{-1}\mathrm{q}_h^k,\mathrm{v}\right)_{\mathcal{T}_h} - \left(u_h^k,\nabla\cdot\mathrm{v}\right)_{\mathcal{T}_h} + \langle\lambda_h^k,\mathrm{v}\cdot\mathrm{n}\rangle_{\partial\mathcal{T}_h} = -\langle g_D,\mathrm{v}\cdot\mathrm{n}\rangle_{\Gamma_D},\\
&\frac{1}{\Delta t^k}\left(u_h^k,w\right)_{\mathcal{T}_h} - \left(\mathrm{c}u_h^k+\mathrm{q}_h^k,\nabla w\right)_{\mathcal{T}_h} + \langle(\widehat{\mathrm{c}u}_h^k+\widehat{\mathrm{q}}_h^k)\cdot\mathrm{n},w\rangle_{\partial\mathcal{T}_h}\\
&\quad = (f,w)_{\mathcal{T}_h} + \frac{1}{\Delta t^k}\left(u_h^{k-1},w\right)_{\mathcal{T}_h},\\
&\langle[\![(\widehat{\mathrm{c}u}_h^k+\widehat{\mathrm{q}}_h^k)\cdot\mathrm{n}]\!],\mu\rangle_{\mathcal{E}_h} = \langle g_N,\mu\rangle_{\Gamma_N}
\end{aligned} \tag{30}$$

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

Next, as we did for the steady-state case, we express (q_h^k,u_h^k) in terms of λ_h^k only. To this end, we consider numerical traces $\widehat{\mathrm{c}u}_h^k+\widehat{\mathrm{q}}_h^k$ of the form

$$\widehat{\mathrm{c}u}_h^k+\widehat{\mathrm{q}}_h^k = \mathrm{c}\,\widehat{u}_h^k + \mathrm{q}_h^k + \tau(u_h^k-\widehat{u}_h^k)\mathrm{n}, \quad \text{on } \partial K.$$

Inserting this expression into (30) and after few algebraic manipulations we obtain that $(\mathrm{q}_h^k,u_h^k,\lambda_h^k)\in\mathrm{V}_h^p\times W_h^p\times M_h^p(0)$ is the solution of the following weak formulation

$$\begin{aligned}
&a(\mathrm{q}_h^k,\mathrm{v}) - b(u_h^k,\mathrm{v}) + c\left(\lambda_h^k,\mathrm{v}\right) = r(\mathrm{v}),\\
&\frac{1}{\Delta t^k}m(u_h^k,w) + b(w,\mathrm{q}_h^k) + d(u_h^k,w) + e\left(\lambda_h^k,w\right) = f(w) + \frac{1}{\Delta t^k}m(u_h^{k-1},w),\\
&c(\mu,\mathrm{q}_h^k) + g(\mu,u_h^k) + h\left(\mu,\lambda_h^k\right) = \ell(\mu),
\end{aligned} \tag{31}$$

for all $(\mathrm{v},w,\mu)\in\mathrm{V}_h^p\times W_h^p\times M_h^p(0)$, where $m(u,w) = (u,w)_{\mathcal{T}_h}$ and the other forms are already introduced in (9). This discrete system has a similar form as the system (8) for the steady case except that the right-hand side contains the information at the previous time-steps. Hence, we can apply exactly the same solution procedure described earlier for the steady case to the time-dependent case at every time-step.

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

4. Local postprocessing

In this section, we propose simple 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 q_h and the superconvergence 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 [9,8] provided that these methods have similar convergence properties as the HDG method.

4.1. Postprocessing of the total flux

We first show that we can postprocess the total approximate flux $\mathrm{q}_h^T = \mathrm{q}_h + \mathrm{c}u_h$ and its numerical trace $\widehat{\mathrm{q}}_h^T = \widehat{\mathrm{q}}_h + \widehat{\mathrm{c}u}_h$ with an element-by-element procedure to obtain an approximation of $\mathrm{q} + \mathrm{c}u$, denoted q_h^{T*} that belongs to $H(\mathrm{div},\Omega)$ and also converges in an optimal fashion. We follow the postprocessing proposed in [8], which is a straightforward extension of the postprocessing procedure introduced in [3,12] and later used in [7,11].

On each simplex $K \in \mathcal{T}_h$, we define the new total flux q_h^{T*} as the only element of $(\mathcal{P}^p(K))^d + \mathrm{x}\mathcal{P}^p(K)$ satisfying, for $p \geqslant 0$,

$$\begin{aligned} &\langle (\mathrm{q}_h^{T*} - \widehat{\mathrm{q}}_h^T)\cdot \mathrm{n}, \mu\rangle_e = 0, \quad \forall \mu \in \mathcal{P}^p(e), \ \ \forall e \in \partial K, \\ &(\mathrm{q}_h^{T*} - \mathrm{q}_h^T, \mathrm{v})_K = 0, \quad \forall \mathrm{v} \in (\mathcal{P}^{p-1}(K))^d \ \text{ if } p \geqslant 1. \end{aligned} \tag{32}$$

It is clear that the function q_h^{T*} belongs to $H(\mathrm{div},\Omega)$, thanks to the single-valuedness of the normal component of the numerical trace $\widehat{\mathrm{q}}_h + \widehat{\mathrm{c}u}_h$.

It is shown in [8] that q_h^{T*} converges optimally with the same order as $\mathrm{q}_h^T = \mathrm{q}_h + \mathrm{c}u_h$. Numerical experiments presented in the next section demonstrate that the new approximation q_h^{T*} has smaller $L^2(\Omega)$ error than q_h^T. It is, however, worth noting that q_h^{T*} is an $H(\mathrm{div},\Omega)$-conforming function, while q_h^T is completely discontinuous over $\mathcal{T}_h$.

4.2. Postprocessing of the scalar variable

Next, we consider postprocessing u_h, q_h, and $\widehat{\mathrm{q}}_h$ to obtain the new approximate scalar variable u_h^* of u. Towards this end, we introduce $\mathcal{P}^{p^*}(\partial K)$ with $p^* = p+1$. We find $(u_h^*, \mathrm{q}_h^*, \lambda_h^*) \in \mathcal{P}^{p^*}(K) \times (\mathcal{P}^{p^*}(K))^d \times (\mathcal{P}^{p^*}(e))^{d+1}$ on the simplex $K \in \mathcal{T}_h$ such that

$$\begin{aligned} &(\kappa^{-1}\nabla \mathrm{q}_h^*, \mathrm{v})_K - (u_h^*, \nabla\cdot \mathrm{v})_K + \langle \lambda_h^*, \mathrm{v}\cdot \mathrm{n}\rangle_{\partial K} = 0 \\ &-(\mathrm{q}_h^* + \mathrm{c}u_h^*, \nabla w)_K + \langle (\widehat{\mathrm{q}}_h^* + \widehat{\mathrm{c}u}_h^*)\cdot \mathrm{n}, w\rangle_{\partial K} = (\nabla\cdot \mathrm{q}_h^{T*}, w)_K, \\ &\langle (\widehat{\mathrm{q}}_h^* + \widehat{\mathrm{c}u}_h^*)\cdot \mathrm{n}, \mu\rangle_{\partial K} = \langle \mathrm{q}_h^{T*}\cdot \mathrm{n}, \mu\rangle_{\partial K}, \\ &(u_h^*, 1)_K = (u_h, 1)_K, \end{aligned} \tag{33}$$

for all $(\mathrm{v}, w, \mu) \in (\mathcal{P}^{p^*}(K))^d \times \mathcal{P}^{p^*}(K) \times (\mathcal{P}^{p^*}(e))^{d+1}$, where

$$\widehat{\mathrm{q}}_h^* + \widehat{\mathrm{c}u}_h^* = \mathrm{q}_h^* + \mathrm{c}\lambda_h^* + \tau(u_h^* - \lambda_h^*)\mathrm{n}.$$

It is a simple matter to verify that this local postprocessing is well defined, namely, the problem (33) has a unique solution u_h^*. This is indeed the case since (*i*) the problem (33) is linear and finite dimensional and (*ii*) $u_h^* = 0$ is its unique solution when we set u_h and q_h^{T*} to zero. To compute u_h^* we need only to invert some matrices of size $\dim(\mathcal{P}^{p^*}(K))$, which are the same for all elements. Since the local postprocessing can also 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 (33) is the discretization by the HDG method of the following convection–diffusion Neumann problem:

$$\begin{aligned} &\nabla\cdot(-\kappa\nabla u + \mathrm{c}u) = \nabla\cdot \mathrm{q}_h^{T*}, \quad \text{in } K, \\ &(-\kappa\nabla u + \mathrm{c}u)\cdot \mathrm{n} = \mathrm{q}_h^{T*}\cdot \mathrm{n}, \quad \text{on } \partial K, \\ &(u,1)_K = (u_h, 1)_K, \end{aligned} \tag{34}$$

for each simplex $K \in \mathcal{T}_h$, where q_h^{T*} is the postprocessed total flux and u_h is the original approximate scalar variable. Therefore, to be successful, our postprocessing procedure relies on the optimal convergence of q_h^{T*} and its divergence $\nabla\cdot \mathrm{q}_h^{T*}$, and on the superconvergence of the average of the approximate scalar variable u_h. In fact, these properties for the HDG method have been theoretically analyzed and confirmed by numerical experiments for the steady symmetric diffusion case in [7,11]: both q_h^{T*} and $\nabla\cdot \mathrm{q}_h^{T*}$ converge with order $p+1$, while $(u_h,1)_K$ superconverges with order $p+2$. Similar results were proven for the HDG method for steady-state convection–diffusion-reaction developed in [8] in the diffusion-dominated regime. Numerical

experiments presented in Section 5 show that the HDG method developed here for the convection–diffusion equations inherits the above-mentioned convergence properties even for the steady and time-dependent convection–diffusion problems. 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.

The postprocessing method proposed here is an extension of the one introduced in [7] for a particular HDG method, and then studied in [11] 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 scheme proposed here and the procedure suggested in [7,11]. 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 time-step without advancing in time. Second, the latter employs the standard CG method to solve the original PDE at element level, whereas the former uses the HDG method to solve (34). The HDG method is employed here because it is more stable than the standard CG method for solving convection–diffusion problems such as (34), especially in the convection-dominated regime.

5. Numerical examples

In this section, we present numerical examples to demonstrate the convergence and accuracy of the proposed HDG method. The first example serves as validation in the diffusion-dominated regime; the second example, as validation in the weakly convection-dominated regime; the third example, as validation in the strongly convection-dominated regime; and the fourth example, as validation in the time-dependent case for a wide range of parabolic and hyperbolic regimes. The implementation is written in Matlab Version 7.1 and direct Gaussian elimination is used to solve the linear system.

5.1. A steady diffusion-dominated problem

In the first example, we study the performance of the centered and upwinded schemes and the effectiveness of our local postprocessing procedure for the diffusion-dominated case. We consider a steady convection–diffusion problem (1) and (2) in which $\Omega = (0,1)\times(0,1)$, $\kappa = 1, \mathbf{c} = (1,1)$, and $g_D = 0$ on $\partial\Omega$. The source term f and Dirichlet boundary condition are chosen such that the exact solution is

$$u = \exp(x+y)\sin(\pi x)\sin(\pi y).$$

Table 1
Example 1: History of convergence for the centered scheme.

Degree	Mesh	Ratio	$\|u-u_h\|_{L^2(\Omega)}$		$\|\mathbf{q}^T-\mathbf{q}_h^T\|_{L^2(\Omega)}$		$\|\mathbf{q}^T-\mathbf{q}_h^{T*}\|_{L^2(\Omega)}$		$\|u-u_h^*\|_{L^2(\Omega)}$	
p	n	R	Error	Order	Error	Order	Error	Order	Error	Order
0	4	1.65	5.62e−1	–	2.43e−0	–	1.43e−0	–	8.04e−1	–
	8	1.72	2.87e−1	0.97	1.14e−0	1.09	7.74e−1	0.88	3.48e−1	1.21
	16	1.76	1.46e−1	0.98	5.56e−1	1.04	4.02e−1	0.95	1.60e−1	1.12
	32	1.78	7.33e−2	0.99	2.75e−1	1.01	2.05e−1	0.97	7.70e−2	1.06
	64	1.79	3.68e−2	0.99	1.37e−1	1.01	1.03e−1	0.99	3.77e−2	1.03
1	4	1.62	2.08e−1	–	8.81e−1	–	4.2e−1	–	1.88e−2	–
	8	1.43	5.47e−2	1.93	2.32e−1	1.93	1.08e−1	1.96	2.51e−3	2.90
	16	1.38	1.39e−2	1.98	5.89e−2	1.97	2.71e−2	1.99	3.24e−4	2.96
	32	1.36	3.49e−3	1.99	1.48e−2	1.99	6.79e−3	2.00	4.11e−5	2.98
	64	1.36	8.74e−4	2.00	3.72e−3	2.00	1.70e−3	2.00	5.17e−6	2.99
2	4	1.15	1.70e−2	–	6.28e−2	–	4.67e−2	–	1.24e−3	–
	8	1.04	2.20e−3	2.95	8.15e−3	2.95	5.95e−3	2.97	7.57e−5	4.04
	16	1.00	2.77e−4	2.99	1.03e−3	2.98	7.47e−4	3.00	4.65e−6	4.02
	32	1.00	3.48e−5	3.00	1.30e−4	2.99	9.33e−5	3.00	2.88e−7	4.01
	64	1.00	4.35e−6	3.00	1.62e−5	3.00	1.17e−5	3.00	1.78e−8	4.01
3	4	1.11	1.40e−3	–	5.08e−3	–	4.04e−3	–	7.73e−5	–
	8	1.04	9.04e−5	3.95	3.29e−4	3.95	2.59e−4	3.96	2.49e−6	4.96
	16	1.02	5.71e−6	3.98	2.07e−5	3.99	1.63e−5	3.99	7.82e−8	4.99
	32	1.02	3.58e−7	3.99	1.30e−6	4.00	1.02e−6	4.00	2.44e−9	5.00
	64	1.02	2.16e−8	4.05	8.07e−8	4.01	6.68e−8	3.93	7.63e−11	5.00
4	4	1.16	1.04e−4	–	3.94e−4	–	3.21e−4	–	5.91e−6	–
	8	1.07	3.34e−6	4.96	1.25e−5	4.98	1.01e−5	4.98	9.32e−8	5.99
	16	1.05	1.05e−7	4.99	3.93e−7	4.99	3.17e−7	5.00	1.47e−9	5.99
	32	1.05	3.32e−9	4.99	1.24e−8	4.99	9.99e−9	4.99	2.31e−11	5.99
	64	1.05	1.04e−10	4.99	3.90e−10	4.99	3.14e−10	4.99	3.64e−13	5.99

Table 2
Example 1: History of convergence for the upwinded scheme.

Degree	Mesh	Ratio	$\|u-u_h\|_{L^2(\Omega)}$		$\|\mathbf{q}^T-\mathbf{q}_h^T\|_{L^2(\Omega)}$		$\|\mathbf{q}^T-\mathbf{q}_h^{T*}\|_{L^2(\Omega)}$		$\|u-u_h^*\|_{L^2(\Omega)}$	
p	n	R	Error	Order	Error	Order	Error	Order	Error	Order
0	4	1.72	7.04e−1	–	2.55e−0	–	1.49e−0	–	9.43e−1	–
	8	1.77	3.59e−1	0.97	1.20e−0	1.09	7.93e−1	0.91	4.20e−1	1.17
	16	1.79	1.82e−1	0.98	5.84e−1	1.04	4.07e−1	0.96	1.97e−1	1.09
	32	1.80	9.16e−2	0.99	2.89e−1	1.01	2.06e−1	0.98	9.54e−2	1.05
	64	1.80	4.60e−2	0.99	1.44e−1	1.01	1.04e−1	0.99	4.69e−2	1.02
1	4	1.60	2.19e−1	–	8.85e−1	–	4.20e−1	–	1.79e−2	–
	8	1.42	5.72e−2	1.93	2.32e−1	1.93	1.08e−1	1.95	2.39e−3	2.91
	16	1.37	1.45e−2	1.98	5.89e−2	1.98	2.73e−2	1.99	3.06e−4	2.96
	32	1.36	3.64e−3	1.99	1.48e−2	1.99	6.86e−3	2.00	3.88e−5	2.98
	64	1.36	9.10e−4	2.00	3.71e−3	2.00	1.72e−3	2.00	4.87e−6	2.99
2	4	1.16	1.85e−2	–	6.25e−2	–	4.75e−2	–	1.20e−3	–
	8	1.04	2.38e−3	2.96	8.09e−3	2.95	6.07e−3	2.97	7.35e−5	4.03
	16	1.00	3.00e−4	2.99	1.02e−3	2.98	7.63e−4	2.99	4.52e−6	4.02
	32	1.00	3.76e−5	3.00	1.28e−4	2.99	9.54e−5	3.00	2.80e−7	4.01
	64	1.00	4.70e−6	3.00	1.61e−5	3.00	1.19e−5	3.00	1.74e−8	4.01
3	4	1.11	1.54e−3	–	4.99e−3	–	4.12e−3	–	7.53e−5	–
	8	1.04	9.92e−5	3.95	3.22e−4	3.95	2.64e−4	3.96	2.42e−6	4.96
	16	1.02	6.26e−6	3.99	2.03e−5	3.99	1.66e−5	3.99	7.61e−8	4.99
	32	1.02	3.92e−7	4.00	1.27e−6	4.00	1.04e−6	4.00	2.40e−9	4.98
	64	1.02	2.45e−8	4.00	7.94e−8	4.00	6.50e−8	4.00	7.60e−11	4.98
4	4	1.16	1.14e−4	–	3.90e−4	–	3.29e−4	–	5.83e−6	–
	8	1.07	3.68e−6	4.96	1.24e−5	4.98	1.04e−5	4.99	9.17e−8	5.99
	16	1.06	1.16e−7	4.99	3.88e−7	4.99	3.26e−7	5.00	1.43e−9	6.00
	32	1.05	3.64e−9	4.99	1.21e−8	5.00	1.02e−8	5.00	2.23e−11	6.00
	64	1.05	1.15e−10	4.99	3.78e−10	5.00	3.19e−10	5.00	3.52e−13	5.99

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.

We present the error and order of convergence in L^2-norm in Table 1 for the centered scheme and Table 2 for the upwinded scheme. We see that in both cases the approximate scalar variable u_h and the approximate total flux $\mathbf{q}_h^T$ converge optimally with order $p+1$ for $p = 0, 1, 2, 3, 4$. It is interesting to note that the HDG method results in approximations which converge optimally for $p = 0$, while some other DG methods such as the LDG method may fail to deliver optimal convergent approximations in this case. We further observe that the centered scheme and upwinded scheme yield very similar results as far as the error and convergence rate are concerned.

We can also see that the postprocessed total flux $\mathbf{q}_h^{*T}$ converges with the same order as $\mathbf{q}_h^T$; 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 $p \geqslant 1$, 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. This result is consistent with the theoretical analysis and numerical experiments presented in [7,11] for symmetric second-order elliptic problems and in [8] for steady convection–diffusion problems. It is also worth noting that other DG methods such as the standard LDG method may not be stable for $p = 0$ and that $p = 0$ solution may be useful in certain cases, for example, when one want to construct a multi-grid solver.

Finally, in the third column of Tables 1 and 2, we tabulate the ratio R which is defined as

$$R = \frac{C}{n^2(p^{1.8}+1)}.$$

Here C is the condition number of the discrete matrix $\mathbb{K}$ given by (12b). The condition number is defined as the ratio of the largest singular value of $\mathbb{K}$ to the smallest singular value, which are computed by a singular value decomposition of the matrix $\mathbb{K}$. It can be seen that the condition number C grows at the rate of $O(h^{-2})$ for all cases and that the condition number are very similar for the centered scheme and the upwinded scheme. The growth rate of $O(h^{-2})$ is typical of the DG approximation for convection–diffusion problems. Furthermore, the condition number appear to grow at the rate of $O(p^{1.8})$ for $p = 2, 3, 4$.

5.2. A steady convection–diffusion problem

We consider a steady convection–diffusion problem (1) and (2) in which $\Omega = (0,1) \times (0,1), \mathbf{c} = (c_x, c_y)$, and $\kappa = 1$. The source term f and Dirichlet boundary conditions are chosen such that we have the exact solution as follows:

$$u(x,y) = xy\frac{(1-e^{(x-1)c_x})(1-e^{(y-1)c_y})}{(1-e^{-c_x})(1-e^{-c_y})}.$$

The solution develops boundary layers along the boundaries $x = 1$ and $y = 1$ for large values of the velocity c. This example serves to validate the performance of the HDG method and the local postprocessing procedures in the weakly convection-dominated regime. For this purpose we shall consider $\mathrm{c} = (25, 25)$ in our numerical experiments.

We present the error and order of convergence in L^2-norm in Table 3 for the centered scheme and Table 4 for the upwinded scheme. Note that the errors being computed on the reduced domain $\tilde{\Omega} = [0.1, 0.9] \times [0.1, 0.9] \subset \Omega$ to exclude the under-resolved boundary layer. We can see that in both cases the postprocessed scalar variable u_h^* converges with order $p + 2$, while the original approximate scalar variable u_h converges with order $p + 1$. Both the postprocessed and original fluxes converge optimally with order $p + 1$; however, the error of the postprocessed total flux is smaller than that of the original flux for $p = 1$ and $p = 2$. Moreover, unlike the original total flux, the normal component of the postprocessed total flux is continuous – an important quality for many practical applications such as in porous media. These results show the effectiveness of the local postprocessing procedures. It is also interesting to note that the upwinded scheme produces the original and postprocessed approximations which have slightly smaller errors than those provided by the centered scheme.

Finally, we study the effectiveness of the postprocessing for unstructured and anisotropic meshes. To this end, we consider an unstructured mesh of 56 elements in Fig. 1(a) and an anisotropic mesh of 72 elements in Fig. 1(b). In anticipation of the boundary layer the two meshes are refined toward the upper right corner of the domain. We present the numerical results in Fig. 2 for the unstructured mesh and in Fig. 3 for the anisotropic mesh. These results are obtained by using the centered scheme for $p = 2$. For both meshes the postprocessed quantity u_h^* is clearly superior to the original approximation u_h since the errors in the max norm are $\|u - u_h\|_{L^\infty(\Omega)} = 1.43 \times 10^{-1}$ and $\|u - u_h^*\|_{L^\infty(\Omega)} = 5.70 \times 10^{-2}$ for the unstructured case, and $\|u - u_h\|_{L^\infty(\Omega)} = 2.25 \times 10^{-2}$ and $\|u - u_h^*\|_{L^\infty(\Omega)} = 6.63 \times 10^{-3}$ for the anisotropic case. The results demonstrate the effectiveness of the postprocessing on both unstructured meshes and anisotropic meshes. Similar results are obtained for the upwinded scheme.

Table 3
Example 2: History of convergence in the L^2-norm for $\mathrm{c} = (25, 25)$. The results are obtained using the centered scheme.

Degree	Mesh	$\|u - u_h\|_{L^2(\Omega)}$		$\|\mathrm{q}^T - \mathrm{q}_h^T\|_{L^2(\Omega)}$		$\|\mathrm{q}^T - \mathrm{q}_h^{T*}\|_{L^2(\Omega)}$		$\|u - u_h^*\|_{L^2(\Omega)}$	
p	n	Error	Order	Error	Order	Error	Order	Error	Order
0	10	1.08e−2	–	4.05e−1	–	6.86e−1	–	1.65e−2	–
	20	5.96e−3	0.85	2.16e−1	0.91	3.48e−1	0.98	6.58e−3	1.33
	40	3.10e−3	0.94	1.11e−1	0.96	1.75e−1	0.99	3.10e−3	1.08
	80	1.57e−3	0.98	5.64e−2	0.98	8.80e−2	1.00	1.55e−3	1.00
	160	7.87e−4	0.99	2.84e−2	0.99	4.40e−2	1.00	7.80e−4	0.99
1	10	6.04e−3	–	1.31e−1	–	8.64e−2	–	3.09e−3	–
	20	1.37e−3	2.14	2.95e−2	2.15	1.64e−2	2.40	3.43e−4	3.17
	40	3.28e−4	2.06	7.08e−3	2.06	3.91e−3	2.07	4.26e−5	3.01
	80	7.94e−5	2.05	1.71e−3	2.05	9.69e−4	2.01	5.33e−6	3.00
	160	1.95e−5	2.03	4.20e−4	2.03	2.42e−4	2.00	6.68e−7	3.00
2	10	4.29e−4	–	1.08e−2	–	1.12e−2	–	1.56e−4	–
	20	8.02e−5	2.42	1.87e−3	2.53	1.51e−3	2.89	7.22e−6	4.44
	40	1.14e−5	2.82	2.63e−4	2.83	2.04e−4	2.89	5.17e−7	3.80
	80	1.49e−6	2.94	3.42e−5	2.94	2.62e−5	2.96	3.52e−8	3.88
	160	1.89e−7	2.98	4.33e−6	2.98	3.30e−6	2.99	2.30e−9	3.94

Table 4
Example 2: History of convergence in the L^2-norm for $\mathrm{c} = (25, 25)$. The results are obtained using the upwinded scheme.

Degree	Mesh	$\|u - u_h\|_{L^2(\Omega)}$		$\|\mathrm{q}^T - \mathrm{q}_h^T\|_{L^2(\Omega)}$		$\|\mathrm{q}^T - \mathrm{q}_h^{T*}\|_{L^2(\Omega)}$		$\|u - u_h^*\|_{L^2(\Omega)}$	
p	n	Error	Order	Error	Order	Error	Order	Error	Order
0	10	1.53e−2	–	6.19e−1	–	6.22e−1	–	2.01e−2	–
	20	7.89e−3	0.95	3.12e−1	0.99	3.16e−1	0.98	8.51e−3	1.24
	40	4.02e−3	0.97	1.57e−1	0.99	1.60e−1	0.98	4.07e−3	1.07
	80	2.03e−3	0.99	7.89e−2	0.99	8.04e−2	0.99	2.02e−3	1.01
	160	1.02e−3	0.99	3.95e−2	1.00	4.03e−2	1.00	1.01e−3	1.00
1	10	5.85e−3	–	1.27e−1	–	8.29e−2	–	2.84e−3	–
	20	1.35e−3	2.12	2.92e−2	2.13	1.61e−2	2.36	3.16e−4	3.17
	40	3.25e−4	2.05	7.04e−3	2.05	3.89e−3	2.05	3.90e−5	3.02
	80	7.92e−5	2.04	1.71e−3	2.04	9.64e−4	2.01	4.88e−6	3.00
	160	1.95e−5	2.02	4.20e−4	2.03	2.41e−4	2.00	6.10e−7	3.00
2	10	4.31e−4	–	1.07e−2	–	1.09e−2	–	1.46e−4	–
	20	7.95e−5	2.44	1.85e−3	2.53	1.47e−3	2.89	6.99e−6	4.38
	40	1.13e−5	2.82	2.59e−4	2.83	1.98e−4	2.89	5.05e−7	3.79
	80	1.47e−6	2.94	3.36e−5	2.95	2.54e−5	2.96	3.43e−8	3.88
	160	1.86e−7	2.98	4.26e−6	2.98	3.20e−6	2.99	2.24e−9	3.94

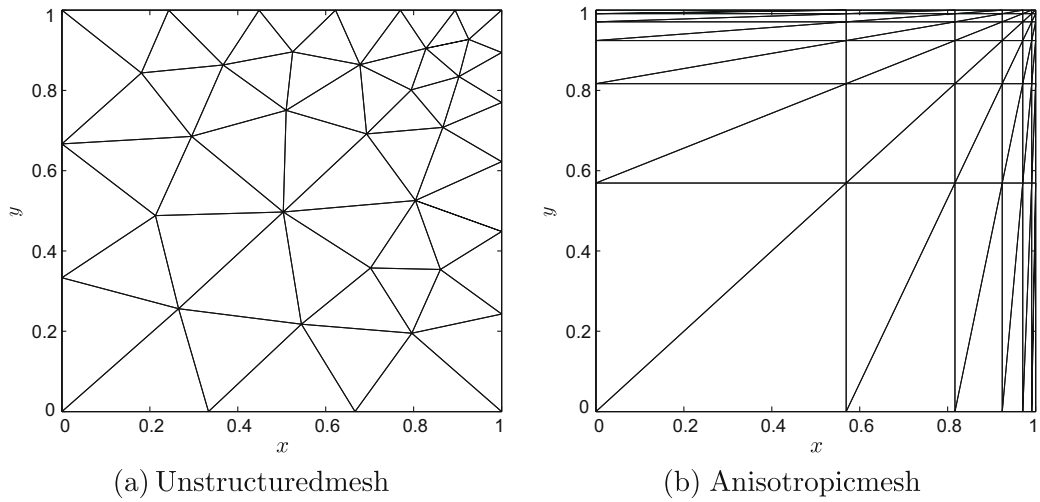

Fig. 1. Example 2: Unstructured and anisotropic meshes used to study the effectiveness of the postprocessing.

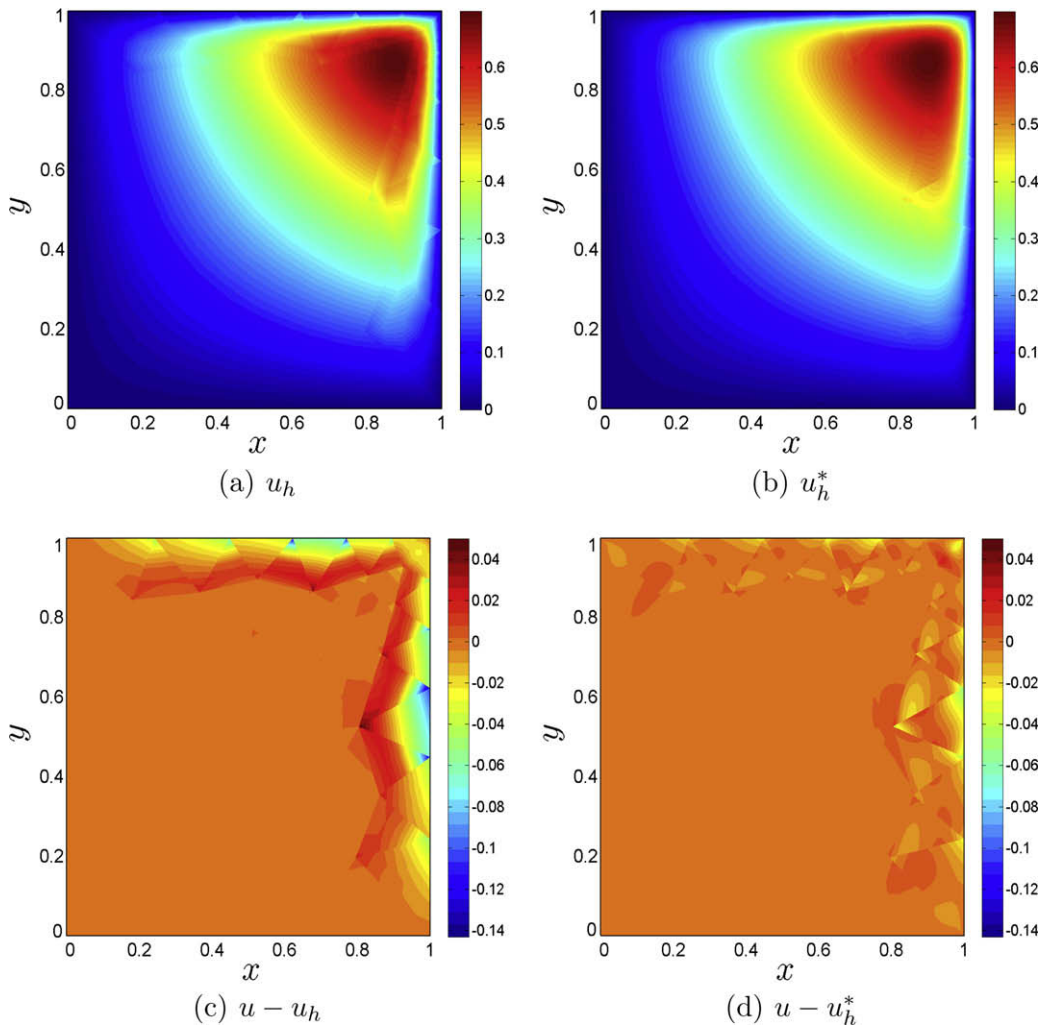

Fig. 2. Example 2: Numerical results computed by the centered scheme for $p = 2$ on the unstructured mesh of Fig. 1(a). Note that the errors in the max norm are $\|u - u_h\|_{L^\infty(\Omega)} = 1.43 \times 10^{-1}$ and $\|u - u_h^*\|_{L^\infty(\Omega)} = 5.70 \times 10^{-2}$, respectively.

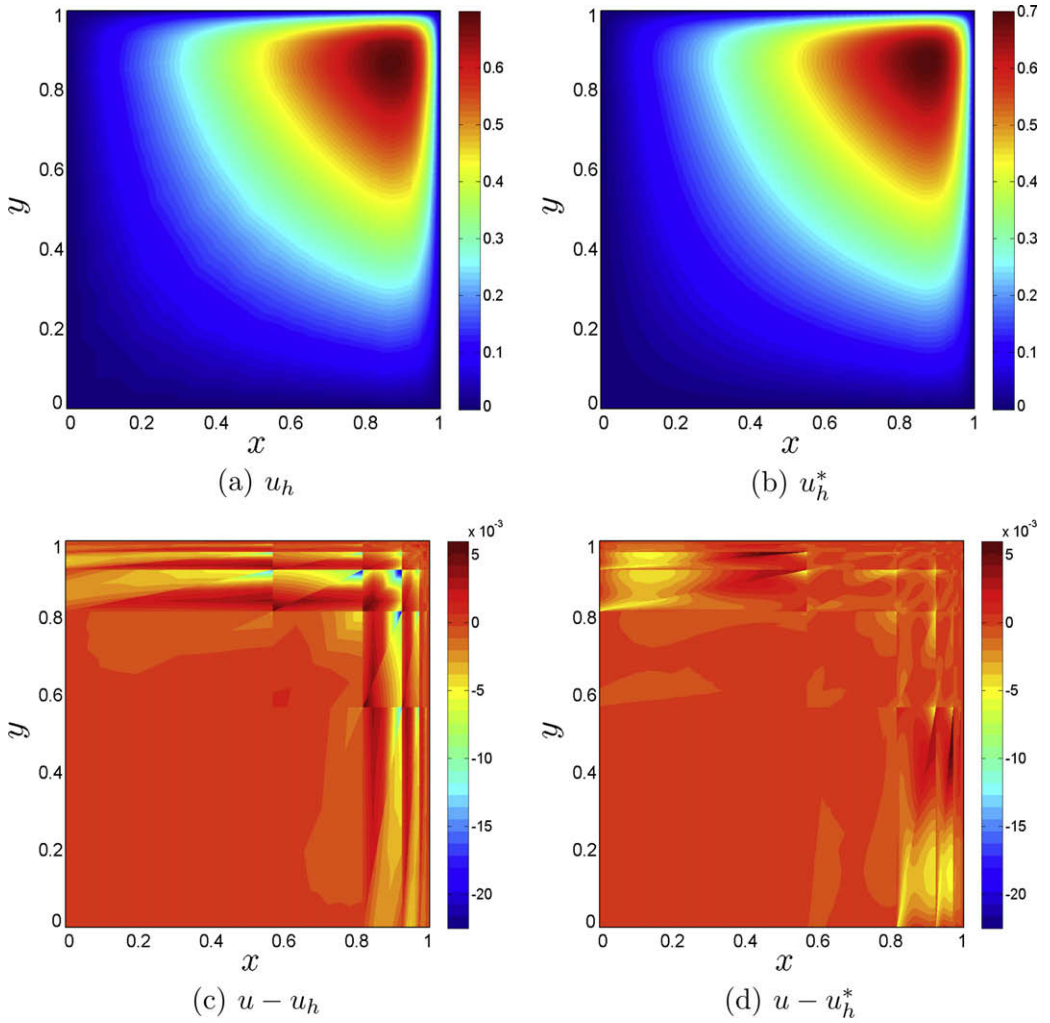

Fig. 3. Example 2: Numerical results computed by the centered scheme for $p = 2$ on the anisotropic mesh of Fig. 1(b). Note that the errors in the max norm are $\|u - u_h\|_{L^\infty(\Omega)} = 2.25 \times 10^{-2}$ and $\|u - u_h^*\|_{L^\infty(\Omega)} = 6.63 \times 10^{-3}$, respectively.

5.3. A steady convection-dominated problem

We consider the rotating flow problem widely used in the literature [4] as a test case to assess numerical schemes in the convection-dominated regime. This is a relevant example of the model problem (1) and (2) with variable coefficients and boundary conditions of mixed type.

The domain is $\Omega = (0,1) \times (0,1) \setminus \Gamma$ with $\Gamma = (0, \frac{1}{2}) \times \{\frac{1}{2}\}$. The (divergence-free) convective field $\mathbf{c} = (y - \frac{1}{2}, \frac{1}{2} - x)$ so that it represents a vortex around the midpoint of the unit square in the clockwise direction. Therefore, Γ represents an inflow boundary denoted by Γ^{in} if we approach Γ from below; but it also represents an outflow boundary denoted by Γ^{out} if we approach it from above. The boundary conditions are specified as $g_D = 1 - \tanh(10(1 - 4x))$ at the inflow boundary Γ^{in}, as $g_D = 0$ at the outer boundary $\partial[(0,1)^2]$, and as $g_N = (\frac{1}{2} - x)(1 - \tanh(10(1 - 4x)))$ (which is equivalent to the homogeneous Neumann condition $\nabla u \cdot \mathbf{n} = 0$) at the outflow boundary Γ^{out}. Furthermore, we set $f = 0$ and $\kappa = 10^{-7}$.

In Fig. 4 we present u_h and λ_h computed by the centered scheme for the case $p = 4$ on a uniform triangulation of 16×16 subdivisions. We further show in Fig. 5 the numerical approximation compared with the boundary data g_D at the outflow boundary Γ^{out}. The results show that the numerical solution is accurate and stable even for the convection-dominated case.

5.4. A time-dependent convection–diffusion problem

This example involves the transport of a two-dimensional rotating Gaussian pulse. The physical domain is $\Omega = (-0.5, 0.5) \times (-0.5, 0.5)$; the rotating velocity field and the forcing term are prescribed as $\mathbf{c} = (-4y, 4x)$ and $f = 0$, respectively; and the final time is $T = \pi/4$ which is the time period for one-half rotation of the Gaussian pulse. The initial condition is given by

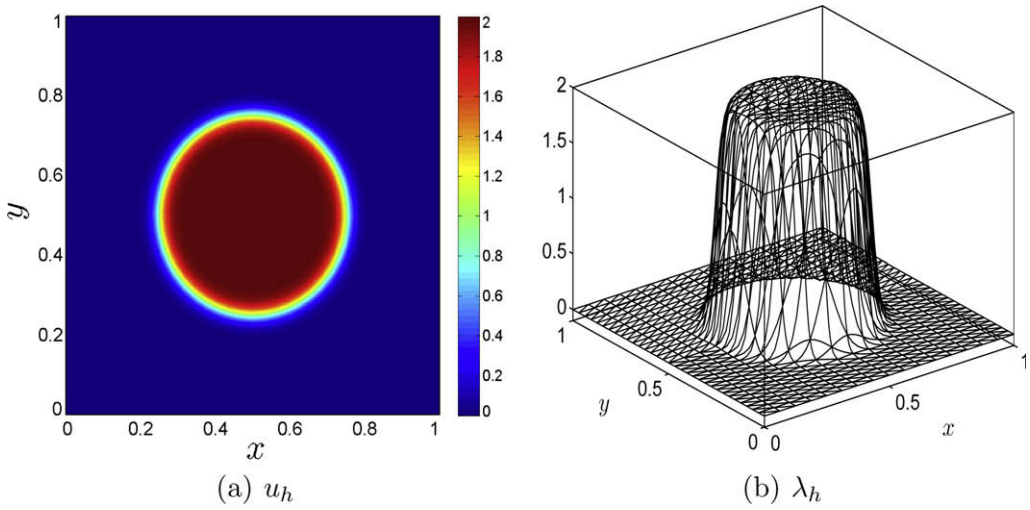

Fig. 4. Example 3: The approximate scalar variable u_h and the numerical trace λ_h computed by the centered scheme for $p = 4$ and $h = 1/16$.

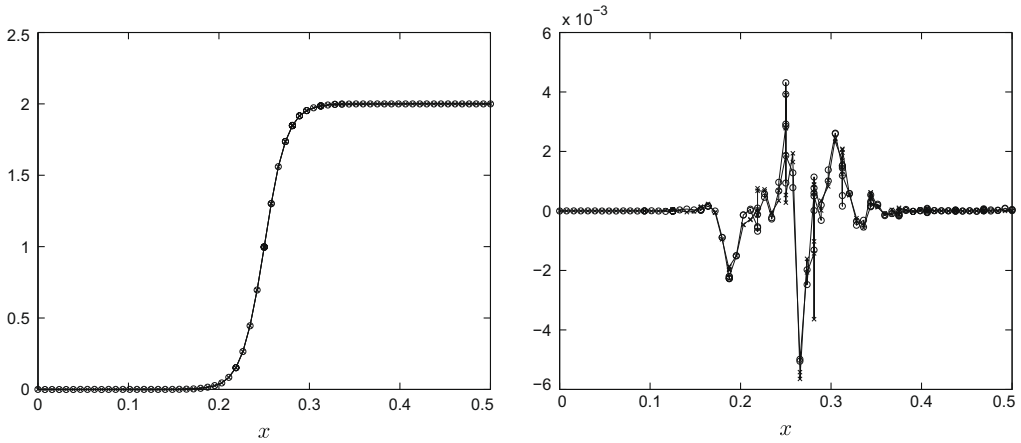

Fig. 5. Example 3: Comparison between u_h and g_D (solid line) at the outflow boundary Γ^{out} for the centered scheme (circle symbol) and the upwinded scheme (cross symbol). The figure on the right shows the difference $g_D - u_h$.

$$u_0(x,y) = \exp\left(-\frac{(x-x_c)^2 + (y-y_c)^2}{2\sigma^2}\right),$$

where (x_c, y_c) is the center and σ is the standard deviation. The exact solution for the problem (28) with a constant diffusivity coefficient κ is then given by

$$u(x,y) = \frac{2\sigma^2}{2\sigma^2 + 4\kappa t} \exp\left(-\frac{(\bar{x}-x_c)^2 + (\bar{y}-y_c)^2}{2\sigma^2 + 4\kappa t}\right),$$

where $\bar{x} = x\cos(4t) + y\sin(4t)$ and $\bar{y} = -x\sin(4t) + y\cos(4t)$. The Dirichlet boundary condition is deduced from the exact solution.

The physics of this problem is rich in the sense that the convection dominates in most of the domain, while the diffusion dominates in the region close to the origin. These types of problems often arise in many important applications and are more difficult to simulate compared with purely convection-dominated or purely diffusion-dominated problems. This problem is suitable to assess many important properties of a numerical scheme, such as stability, convergence, numerical diffusion, spurious oscillations, and phase errors.

In the numerical experiments, the data are chosen as follows $x_c = -0.2, y_c = 0$, and $\sigma = 0.1$. Furthermore, we consider two different values of the viscosity constant $\kappa = 0.01$ and $\kappa = 0.001$. These values of the viscosity represent a wide range of parabolic and hyperbolic regimes in this particular problem. Below we present numerical results.

Table 5
Example 4: History of convergence of the L^2-error of u_h at the final time computed with $p = 3$ and BDF3 as a function of time-step Δt for the centered scheme and upwinded scheme.

Δt	Centered scheme				Upwinded scheme			
	κ=0.01		κ=0.001		κ=0.01		κ=0.001	
	Error	Order	Error	Order	Error	Order	Error	Order
0.08	2.58e−02	–	1.25e−01	–	2.71e−02	–	1.35e−01	–
0.04	4.53e−03	2.51	2.47e−02	2.34	4.82e−03	2.49	2.81e−02	2.27
0.02	5.57e−04	3.03	3.09e−03	3.00	5.95e−04	3.02	3.42e−03	3.04
0.01	6.77e−05	3.04	3.85e−04	3.01	7.38e−05	3.01	4.21e−04	3.02

First, we look at the temporal convergence properties of the numerical solution computed by the HDG method when the third-order backward difference formulae (BDF3) is employed to discretize the time derivative with a fixed time-step Δt. The numerical solution is computed on 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. We show in Table 5 the error and order of convergence as a function of Δt for the approximate scalar variable u_h at the final time. We observe that the accuracy is third-order in time.

We now look at the spatial convergence properties of the HDG method in the time-dependent case. For this purpose we consider a very small time-step $\Delta t = 5 \times 10^{-4}$ so that the overall error is governed by the spatial error. We present the history of convergence in Tables 6–9 for both the centered scheme and the upwinded scheme. We see that the approximate scalar variable u_h and total flux q_h^T converge optimally with order $p+1$ and that the postprocessed total flux q_h^{T*} also converges with order $p+1$. These results indicate that the numerical solution has optimal convergence rates even for the special case $p = 0$. Moreover, the postprocessed scalar variable converges with order $p+1$ for $p = 0$ and with order $p+2$ for $p \geqslant 1$. This shows that when polynomials of degree $p \geqslant 1$ are used the HDG method possesses superconvergence properties and that the local postprocessing procedures are effective even in the time-dependent case. In short, the convergence properties seen in the previous (steady) examples are also observed in this (time-dependent) example.

5.5. The contaminant transport problem

We consider the transport of the contaminant concentration with kinematic viscosity κ in a two-dimensional domain $\Omega = (-1.25, -1.25) \times (0, 10)$ driven by a convective velocity field c. Here the velocity field c is taken to be the analytical solution of the Kovasznay flow [16]

$$\mathrm{c} = \left(1 - e^{\gamma x}\cos(2\pi y), \frac{\gamma}{2\pi} e^{\gamma x}\sin(2\pi y)\right),$$

where $\gamma = \frac{Re}{2} - \sqrt{\frac{Re^2}{4} + 4\pi^2}$ and we take $Re = 100$ for the Reynolds number. Since the Kovasznay solution models laminar flow behind a two-dimensional grid, this example can be considered as an application of the present method to study the contaminant transport behind a two-dimensional grid.

At time $t = 0$ the initial contaminant concentration is a superposition of three Gaussian distributions

$$w_0 = e^{-\frac{(x-1)^2+y^2}{0.5^2}} + e^{-\frac{(x-1)^2+(y-0.5)^2}{0.5^2}} + e^{-\frac{(x-1)^2+(y+0.5)^2}{0.5^2}}.$$

Table 6
Example 4: History of convergence of the approximate solution at the final time for $\kappa = 0.01$, computed by the centered scheme and the BDF3 scheme.

Degree	Mesh	$\|u - u_h\|_{L^2(\Omega)}$		$\|\mathrm{q}^T - \mathrm{q}_h^T\|_{L^2(\Omega)}$		$\|\mathrm{q}^T - \mathrm{q}_h^{T*}\|_{L^2(\Omega)}$		$\|u - u_h^*\|_{L^2(\Omega)}$	
p	n	Error	Order	Error	Order	Error	Order	Error	Order
0	8	5.24e−2	–	5.55e−2	–	5.11e−2	–	6.53e−2	–
	16	3.43e−2	0.61	3.71e−2	0.58	3.66e−2	0.48	3.48e−2	0.91
	32	2.07e−2	0.73	2.26e−2	0.72	2.29e−2	0.68	2.05e−2	0.76
	64	1.17e−2	0.83	1.27e−2	0.83	1.30e−2	0.81	1.16e−2	0.83
	128	6.24e−3	0.90	6.80e−3	0.90	6.98e−3	0.90	6.21e−3	0.90
1	8	4.48e−3	–	5.83e−3	–	8.86e−3	–	8.52e−3	–
	16	1.44e−3	1.64	1.65e−3	1.82	1.79e−3	2.31	1.21e−3	2.81
	32	4.41e−4	1.71	4.81e−4	1.78	3.93e−4	2.18	1.61e−4	2.92
	64	1.22e−4	1.86	1.31e−4	1.88	9.41e−5	2.06	2.12e−5	2.92
	128	3.20e−5	1.93	3.42e−5	1.94	2.33e−5	2.01	2.88e−6	2.88
2	8	5.58e−4	–	7.27e−4	–	8.63e−4	–	6.64e−4	–
	16	7.21e−5	2.95	9.41e−5	2.95	1.03e−4	3.07	4.17e−5	3.99
	32	9.50e−6	2.92	1.21e−5	2.95	1.28e−5	3.01	2.48e−6	4.07
	64	1.24e−6	2.94	1.55e−6	2.97	1.60e−6	3.00	1.56e−7	3.99
	128	1.63e−7	2.93	2.02e−7	2.94	2.05e−7	2.96	9.82e−9	3.99

Table 7
Example 4: History of convergence of the approximate solution at the final time for $\kappa = 0.01$, computed by the upwinded scheme and the third-order BDF scheme.

Degree	Mesh	$\|u - u_h\|_{L^2(\Omega)}$		$\|\mathbf{q}^T - \mathbf{q}_h^T\|_{L^2(\Omega)}$		$\|\mathbf{q}^T - \mathbf{q}_h^{T*}\|_{L^2(\Omega)}$		$\|u - u_h^*\|_{L^2(\Omega)}$	
p	n	Error	Order	Error	Order	Error	Order	Error	Order
0	8	4.91e−2	–	5.29e−2	–	4.80e−2	–	6.63e−2	–
	16	3.07e−2	0.68	3.39e−2	0.64	3.32e−2	0.53	3.13e−2	1.08
	32	1.77e−2	0.79	1.97e−2	0.78	1.99e−2	0.74	1.75e−2	0.84
	64	9.55e−3	0.89	1.07e−2	0.88	1.09e−2	0.87	9.47e−3	0.89
	128	4.96e−3	0.94	5.57e−3	0.94	5.73e−3	0.93	4.94e−3	0.94
1	8	5.33e−3	–	6.27e−3	–	7.53e−3	–	7.31e−3	–
	16	1.66e−3	1.68	1.80e−3	1.80	1.62e−3	2.21	1.01e−3	2.85
	32	4.80e−4	1.79	5.08e−4	1.82	3.69e−4	2.14	1.33e−4	2.93
	64	1.29e−4	1.90	1.35e−4	1.91	8.90e−5	2.05	1.76e−5	2.91
	128	3.33e−5	1.95	3.50e−5	1.95	2.19e−5	2.02	2.47e−6	2.83
2	8	6.79e−4	–	7.71e−4	–	8.21e−4	–	5.70e−4	–
	16	8.78e−5	2.95	9.95e−5	2.95	9.91e−5	3.05	3.45e−5	4.05
	32	1.11e−5	2.99	1.26e−5	2.98	1.23e−5	3.01	2.11e−6	4.03
	64	1.39e−6	2.99	1.59e−6	2.99	1.53e−6	3.01	1.37e−7	3.95
	128	1.78e−7	2.97	2.05e−7	2.96	1.96e−7	2.96	8.50e−9	4.01

Table 8
Example 4: History of convergence of the approximate solution at the final time for $\kappa = 0.001$, computed by the centered scheme and the third-order BDF scheme.

Degree	Mesh	$\|u - u_h\|_{L^2(\Omega)}$		$\|\mathbf{q}^T - \mathbf{q}_h^T\|_{L^2(\Omega)}$		$\|\mathbf{q}^T - \mathbf{q}_h^{T*}\|_{L^2(\Omega)}$		$\|u - u_h^*\|_{L^2(\Omega)}$	
p	n	Error	Order	Error	Order	Error	Order	Error	Order
0	8	1.12e−1	–	1.02e−1	–	8.69e−2	–	2.17e−1	–
	16	8.80e−2	0.49	7.58e−2	0.42	7.20e−2	0.27	1.20e−1	0.85
	32	5.33e−2	0.59	5.19e−2	0.55	5.15e−2	0.48	6.64e−2	0.86
	64	3.24e−2	0.72	3.21e−2	0.69	3.23e−2	0.67	3.63e−2	0.87
	128	1.83e−2	0.83	1.83e−2	0.81	1.85e−2	0.81	1.88e−2	0.95
1	8	1.68e−2	–	1.85e−2	–	3.01e−2	–	9.49e−2	–
	16	4.39e−3	1.93	4.40e−3	2.08	6.44e−3	2.22	1.39e−2	2.77
	32	1.32e−3	1.73	1.27e−3	1.79	1.25e−3	2.37	1.97e−3	2.82
	64	3.83e−4	1.79	3.67e−4	1.79	2.69e−4	2.21	2.82e−4	2.81
	128	1.03e−4	1.89	9.92e−5	1.89	6.33e−5	2.09	4.17e−5	2.76
2	8	2.96e−3	–	2.96e−3	–	3.83e−3	–	1.22e−2	–
	16	3.99e−4	2.89	4.07e−4	2.87	4.27e−4	3.16	8.09e−3	3.91
	32	4.91e−5	3.02	5.15e−5	2.98	5.16e−5	3.05	5.09e−5	3.99
	64	5.94e−6	3.05	6.43e−6	3.00	6.3e−6	3.03	3.16e−6	4.01
	128	7.61e−7	2.96	8.43e−7	2.93	8.05e−7	2.97	2.00e−7	3.98

Table 9
Example 4: History of convergence of the approximatesolution at the final time for $\kappa = 0.001$, computed by the upwinded scheme and the third-order BDF scheme.

Degree	Mesh	$\|u - u_h\|_{L^2(\Omega)}$		$\|\mathbf{q}^T - \mathbf{q}_h^T\|_{L^2(\Omega)}$		$\|\mathbf{q}^T - \mathbf{q}_h^{T*}\|_{L^2(\Omega)}$		$\|u - u_h^*\|_{L^2(\Omega)}$	
p	n	Error	Order	Error	Order	Error	Order	Error	Order
0	8	1.11e−1	–	1.02e−1	–	8.62e−2	–	1.70e−1	–
	16	7.84e−2	0.50	7.51e−2	0.44	7.12e−2	0.28	1.08e−1	0.66
	32	5.11e−2	0.62	5.01e−2	0.58	4.97e−2	0.52	6.11e−2	0.82
	64	3.02e−2	0.76	2.99e−2	0.75	3.01e−2	0.72	3.39e−2	0.85
	128	1.64e−2	0.88	1.63e−2	0.88	1.65e−2	0.87	1.69e−2	1.01
1	8	1.73e−2	–	1.77e−2	–	2.65e−2	–	9.54e−2	–
	16	4.80e−3	1.85	4.62e−3	1.94	6.11e−3	2.12	1.37e−2	2.80
	32	1.42e−3	1.75	1.33e−3	1.80	1.22e−3	2.33	1.87e−3	2.87
	64	4.05e−4	1.82	3.79e−4	1.81	2.64e−4	2.21	2.59e−4	2.85
	128	1.08e−4	1.91	1.02e−4	1.90	6.15e−5	2.10	3.70e−5	2.81
2	8	3.24e−3	–	3.16e−3	–	3.80e−3	–	1.06e−2	–
	16	4.34e−4	2.90	4.22e−4	2.91	4.11e−4	3.21	7.00e−4	3.92
	32	5.55e−5	2.97	5.43e−5	2.96	4.98e−5	3.05	4.40e−5	3.99
	64	6.91e−6	3.01	6.81e−6	2.99	6.09e−6	3.03	2.75e−6	4.00
	128	8.80e−7	2.97	8.82e−7	2.95	7.78e−7	2.97	1.76e−7	3.97

Every two seconds $T = 2$ after that the same contaminant concentration w_0 is injected into the flow field, while the contaminant concentration u is being transported downstream. We consider the contaminant transport over the time horizon of $T_f = 10$ s. Therefore, within the jth period $[(j-1)T, jT]$ for $j = 1, \ldots, 5$, the physical process is modeled by the time-dependent convection–diffusion equation

$$\frac{\partial u}{\partial t} + \nabla \cdot (\mathrm{c}u) - \kappa \nabla^2 u = 0, \quad \text{in } \Omega \times ((j-1)T, jT],$$
$$u = u_0, \quad \text{in } \Omega \text{ for } t = (j-1)T.$$

In order to model the injection of the new contaminant concentration w_0 into the flow field at the beginning of the jth period, the initial data is given by

$$u_0 = \begin{cases} w_0, & \text{if } j = 1, \\ w_0 + u((j-1)T), & \text{if } j > 1. \end{cases}$$

Here $u((j-1)T)$ is the contaminant concentration at time $t = (j-1)T$ obtained from the $(j-1)$th period.

On the inflow boundary Γ_D, which is defined by $x = 0$ and $-1.25 \leqslant y \leqslant 1.25$, the contaminant concentration u satisfies a homogenous Dirichlet condition

$$u = 0, \quad \text{on } \Gamma_D.$$

On the remaining boundary $\Gamma_N = \Gamma_D \setminus \partial\Omega$, it satisfies a homogenous Neumann condition

$$\nabla u \cdot \mathrm{n} = 0, \quad \text{on } \Gamma_N.$$

We note that a similar contaminant transport problem has been studied in the model reduction context [2].

We shall consider the case $\kappa = 0.01$. The reference Peclet number is defined and calculated as

$$Pe = \frac{v_{\max} D}{\kappa} = 200,$$

where $v_{\max} = 2$ is the maximum velocity magnitude and $D = 1$ is the distance between the two consecutive cylinders in the Kovasznay flow. For the spatial discretization we consider triangular meshes obtained by splitting an $n \times n$ Cartesian grid into $2n^2$ triangles, which yields the aspect ratio of $4/1$ for all triangles. For the temporal discretization, we consider the BDF2 scheme with a constant time-step $\Delta t = 0.025$. Below we present the numerical results obtained using the centered scheme.

We present in Fig. 6 the computed contaminant concentration u_h at the end of each period for the grid $n = 20$ and polynomial degree $p = 3$. We further show in Fig. 7(a) u_h at the spatial point (3.0,0.0) as a function of time t for the grid $n = 20$

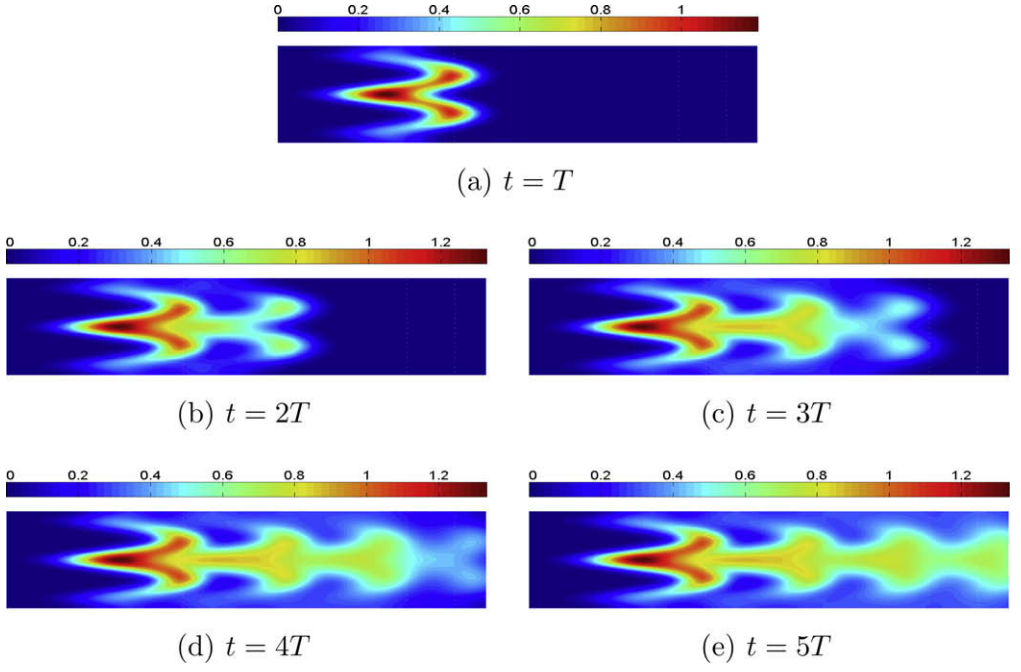

Fig. 6. Example 5: u_h at the end of each period for $n = 20$ and $p = 3$.

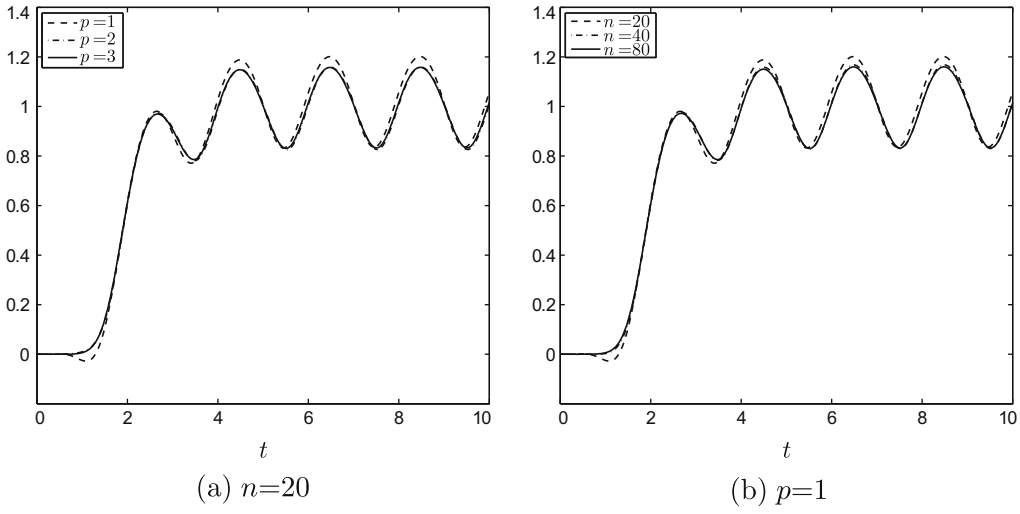

Fig. 7. Example 5: u_h at the spatial point $(3.0, 0.0)$ as a function of time t for: (a) the grid $n = 20$ for $p = 1, 2$, and 3, and (b) the polynomial degree $p = 1$ for $n = 20, 40$, and 80.

and for $p = 1, 2$, and 3. Fig. 7(b) plots the same quantity for $p = 1$ for different values of $n = 20, 40$, and 80. The results in Fig. 7 demonstrates $h - p$ convergence of the approximate solution. To illustrate how the postprocessing can improve the approximation we present in Fig. 8 u_h and u_h^* for $p = 1$ on the grid $n = 20$. Comparing Fig. 8 with Fig. 6, we see that u_h^* appears a better approximation than u_h. For this highly convective flow the postprocessing procedure is defined as in Section 4 but excluding the convective contributions in Eqs. (32) and (33). The effectiveness of the postprocessing can be seen more clearly by plotting in Fig. 9 u_h and u_h^* at the spatial point (3.0, 0.0) as a function of t and comparing them with u_h for $p = 2$ on the same grid.

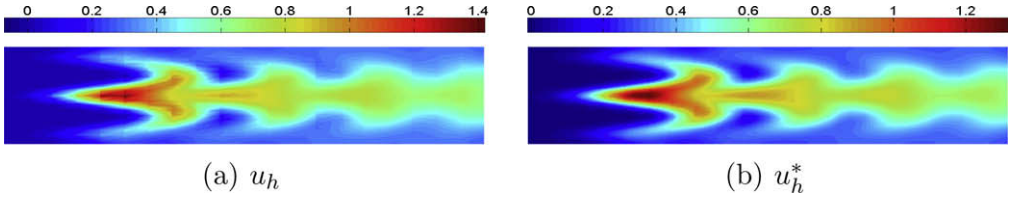

Fig. 8. Example 5: Comparison of u_h and u_h^* at the final time $t = 5$ T for $n = 20$ and $p = 1$.

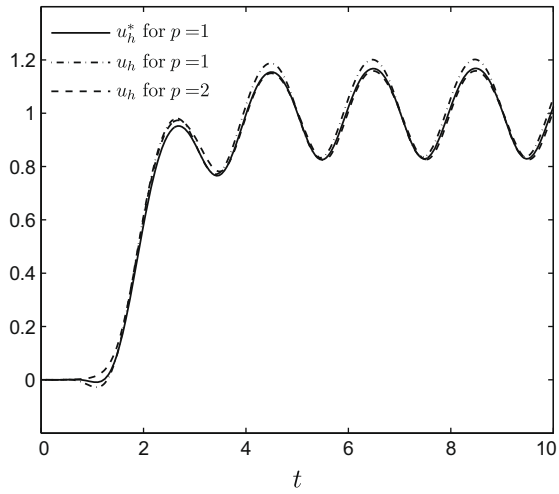

Fig. 9. Example 5: Comparison of u_h and u_h^* at the spatial point (3.0, 0.0) as a function t for $p = 1$ and $n = 20$. The reference value is u_h for $p = 2$ on the same grid.

In summary, the HDG method and postprocessing procedure work well for both the steady and time-dependent problems in a wide range of elliptic, parabolic and hyperbolic regimes. The proposed approach is thus well-suited for the high-order accurate and stable solution of convection–diffusion equations.

6. Conclusions

In this paper, we present an implicit high-order hybridizable DG (HDG) method for steady and time-dependent convection–diffusion equations. The main motivation is the reduction in the number of the globally coupled degrees of freedom of the DG approximations. The method developed achieves this objective by expressing the approximate scalar variable and flux in terms of the approximate trace of the scalar variable and enforcing flux continuity in an explicit manner. This allows us to eliminate both the approximate solution and flux to obtain a matrix equation involving only the numerical trace.

We have shown that the HDG method is a DG method with the special choice of the numerical traces. For steady-state problems, the HDG method differs from the class of DG methods analyzed in [1] in that the approximate flux q_h cannot be expressed in terms of the approximate scalar variable u_h in an element-by-element fashion. This is just a direct consequence of the choice of the numerical trace λ_h which depends on both q_h and u_h. The HDG method is consistent and stable, and achieves optimal convergence rates even for $p = 0$. In fact, thanks to the dependence of the numerical trace λ_h on q_h, the approximate flux q_h converges optimally with order $p + 1$ in L^2 norm, while the average of the approximate scalar variable superconverges when $p \geqslant 1$. We propose the centered scheme and the upwinded scheme for choosing the stabilization parameters. These schemes are shown to provide accurate and stable approximations for a wide range of convective–diffusive regimes. We extend the HDG method to treat time-dependent problems. The HDG method also exhibits the above-mentioned convergence properties in the time-dependent case.

We have also developed simple local postprocessing procedures that exploit the relation between the scalar variable and the flux as well as the superconvergence properties of the HDG method to obtain new approximations of the scalar variable and total flux. The postprocessed total flux converges with the same order as the original total flux; however, the normal component of the postprocessed total flux is continuous, while that of the original flux is discontinuous. The postprocessed scalar variable converges with order $p + 2$ for $p \geqslant 1$, which is one order higher the original scalar variable. Moreover, the postprocessing procedures are less expensive than the solution procedure, since they involve operations at element level. Therefore, compared with the more established DG methods such as the LDG, the proposed approach can be more efficient: the $p + 2$ convergent solution can be computed at the cost of a DG approximation using polynomials of degree p.

Several numerical examples are presented to demonstrate the accuracy and stability of the HDG method for convection–diffusion equations from diffusion-dominated to convection-dominated regimes. The obtained results appear to confirm the earlier conclusions.

We end this paper by pointing out that the extension of this work to linear and non-linear hyperbolic systems of conservation laws constitutes the subject of ongoing research.

Acknowledgments

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

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] O. Bashir, K. Willcox, O. Ghattas, B. van Bloemen Waanders, J. Hill, Hessian-based model reduction for large-scale systems with initial-condition inputs, Int. J. Numer. Methods Eng. 73 (6) (2008) 844–868.
[3] P. Bastian, B. Rivière, Superconvergence and $H(\mathrm{div})$ projection for discontinuous Galerkin methods, Int. J. Numer. Methods Fluids 42 (2003) 1043–1057.
[4] A.N. Brooks, T.J.R. Hughes, Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Eng. 32 (1982) 199–259.
[5] P. Castillo, B. Cockburn, I. Perugia, D. Schötzau, An a priori error analysis of the local discontinuous Galerkin method for elliptic problems, SIAM J. Numer. Anal. 38 (5) (2001) 1676–1706.
[6] B. Cockburn, B. Dong, An analysis of the minimal dissipation local discontinuous Galerkin method for convection–diffusion problems, J. Sci. Comput. 32 (2) (2007) 233–262.
[7] B. Cockburn, B. Dong, J. Guzmán, A superconvergent LDG-hybridizable Galerkin method for second-order elliptic problems, Math. Comput. 77 (2008) 1887–1916.
[8] B. Cockburn, B. Dong, J. Guzmán, M. Restelli, R. Sacco, A hybridizable discontinuous Galerkin method for steady-state convection-diffusion-reaction problems, submitted for publication.
[9] B. Cockburn, J. Gopalakrishnan, A characterization of hybridized mixed methods for second order elliptic problems, SIAM J. Numer. Anal. 42 (1) (2004) 283–301.
[10] 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., in press.
[11] B. Cockburn, J. Guzmán, H. Wang, Superconvergent discontinuous Galerkin methods for second-order elliptic problems, Math. Comput. 78 (2009) 1–24.
[12] B. Cockburn, G. Kanschat, D. Schötzau, A locally conservative LDG method for the incompressible Navier–Stokes equations, Math. Comput. 74 (2005) 1067–1095.

[13] B. Cockburn, C.W. Shu, The local discontinuous Galerkin method for convection–diffusion systems, SIAM J. Numer. Anal. 35 (1998) 2440–2463.
[14] R. Ewing, J. Wang, Y. Yang, A stabilized discontinuous finite element method for elliptic problems, Numer. Linear Alg. Appl. 10 (2003) 83–104.
[15] J. Giannakouros, G.E. Karniadakis, Spectral element-FCT method for scalar hyperbolic conservation laws problems, Int. J. Numer. Methods Fluids 14 (6) (1992) 707–727.
[16] L.I.G. Kovasznay, Laminar flow behind a two-dimensional grid, Proc. Camb. Philos. Soc. 44 (1948) 58–62.
[17] J. Peraire, P.O. Persson, The compact discontinuous Galerkin (CDG) method for elliptic problems, SIAM J. Sci. Comput. 30 (4) (2008) 1806–1824.
[18] 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.
[19] W.H. Reed, T.R. Hill, Triangular mesh methods for the neutron transport equation, Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.
[20] S.J. Sherwin, R.M. Kirby, J. Peiró, R.L. Taylor, O.C. Zienkiewicz, On 2D elliptic discontinuous Galerkin methods, Int. J. Numer. Methods Eng. 65 (5) (2006) 752–784.