Journal of Computational Physics 306 (2016) 390–421

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Hybridizable discontinuous Galerkin projection methods for Navier–Stokes and Boussinesq equations

M.P. Ueckermann *, P.F.J. Lermusiaux *

Department of Mechanical Engineering, Massachusetts Institute of Technology, 77 Mass. Avenue, Cambridge, MA 02139, United States

ARTICLE INFO

Article history:
Received 13 January 2015
Received in revised form 7 September 2015
Accepted 10 November 2015
Available online 2 December 2015

Keywords:
Hybridizable DG
Projection methods
IMEX-RK
High-order
Selective limiter
Quadrature-free
Navier–Stokes
Boussinesq
Ocean modeling

ABSTRACT

Schemes for the incompressible Navier–Stokes and Boussinesq equations are formulated and derived combining the novel Hybridizable Discontinuous Galerkin (HDG) method, a projection method, and Implicit–Explicit Runge–Kutta (IMEX-RK) time-integration schemes. We employ an incremental pressure correction and develop the corresponding HDG finite element discretization including consistent edge-space fluxes for the velocity predictor and pressure correction. We then derive the proper forms of the element-local and HDG edge-space final corrections for both velocity and pressure, including the HDG rotational correction. We also find and explain a consistency relation between the HDG stability parameters of the pressure correction and velocity predictor. We discuss and illustrate the effects of the time-splitting error. We then detail how to incorporate the HDG projection method time-split within standard IMEX-RK time-stepping schemes. Our high-order HDG projection schemes are implemented for arbitrary, mixed-element unstructured grids, with both straight-sided and curved meshes. In particular, we provide a quadrature-free integration method for a nodal basis that is consistent with the HDG method. To prevent numerical oscillations, we develop a selective nodal limiting approach. Its applications show that it can stabilize high-order schemes while retaining high-order accuracy in regions where the solution is sufficiently smooth. We perform spatial and temporal convergence studies to evaluate the properties of our integration and selective limiting schemes and to verify that our solvers are properly formulated and implemented. To complete these studies and to illustrate a range of properties for our new schemes, we employ an unsteady tracer advection benchmark, a manufactured solution for the steady diffusion and Stokes equations, and a standard lock-exchange Boussinesq problem.

1. Introduction

Solving systems of equations that govern fluid flows is required for a vast number of applications, from designing microfluid devices to predicting ocean dynamics, the weather, and climate on Earth. These equations are challenging to solve accurately and quickly. A common trend is to attempt larger, more complex, and increasingly important problems that require more accurate answers. Since solutions of multi-scale nonlinear equations are sensitive to small errors, their simulations require accurate numerical schemes. Schemes should be capable of phase-resolved predictions up to predictability

* Corresponding authors. Tel.: +1 617 324 5172.
E-mail addresses: mpuecker@mit.edu (M.P. Ueckermann), pierrel@mit.edu (P.F.J. Lermusiaux).

http://dx.doi.org/10.1016/j.jcp.2015.11.028

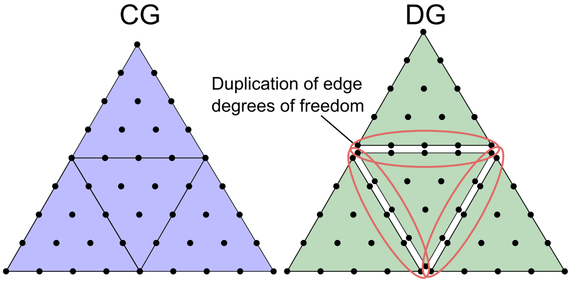

Fig. 1. Discontinuous Galerkin have more degrees of freedom compared to continuous Galerkin on the same mesh.

limits, which can be relatively long durations for certain waves and coherent structures. Schemes should also limit numerical dissipation and other truncation errors that can insidiously alter the system dynamics: for example, global properties of discretized solutions can then differ from those of analytical solutions [59]. Schemes should also be optimized for modern computational architectures. Presently, our focus is on incompressible fluid and ocean flow simulations, for ocean engineering and regional ocean dynamics applications. Our aim is to develop a new class of numerical schemes which combines the recently developed hybridizable discontinuous Galerkin method with the well-studied projection method to obtain new high-order accurate schemes with excellent efficiency for ocean applications.

Projection methods, pioneered by Chorin [7] and Témam [55], decouple the solution of the velocity and pressure. Herein we consider a class of projection methods where the diffusion terms are treated implicitly, while the remaining terms (such as non-linear advection) are treated explicitly, and the pressure is handled through the particular projection scheme. With projection methods, smaller systems can be solved for each velocity component and the pressure separately, instead of a globally coupled solution. The drawback is the splitting errors (e.g. [18]). While many different projection methods exist (e.g. [23]), we focus here on the scheme by Timmermans et al. [56].

Because the solutions of implicit diffusion and pressure Poisson equations often dominate the cost, it is critical to solve them efficiently. Discontinuous Galerkin (DG) methods [27] are attractive because they can be high-order accurate on arbitrary meshes. High-order accurate schemes reach a smaller error tolerance with fewer degrees of freedom than low-order accurate schemes. They also promise to be more efficient on new computational architectures because the computation to memory ratio is higher, and present computations are often limited by the memory bandwidth. Additionally, the DG method is well suited to advection-dominated problems because upwinding can be used to stabilize the scheme. However, Continuous Galerkin (CG) methods [54] are deemed to be less expensive for Poisson equations [32]. All of this, as well as conservative and consistency considerations, led us to the Hybridizable Discontinuous Galerkin (HDG) [40,14] method, aiming for optimal combination of accuracy with cost, hence efficiency.

The development of HDG methods [40,14] was motivated by the desire to improve the computational efficiency of DG compared to CG for elliptic problems. DG has at times been deemed too expensive compared to CG by counting the number of degrees of freedom (DOF) required for DG and CG on the same mesh. The DG discretization duplicates DOFs on the edges of elements (Fig. 1), which means that a larger matrix needs to be inverted compared to the CG case. This comparison is not necessarily fair, since the DG scheme may reach the same level of accuracy with a coarser mesh (or same number of DOFs). To address these issues, HDG parameterizes element-local solutions using a new global edge space, ensuring conservation of normal fluxes. This reduces the effective number of globally coupled DOFs and renders HDG competitive with CG for elliptic problems [62,32].

The HDG method was derived for second order elliptic [14,15] and convection–diffusion equations [40,41]. Next, schemes for Stokes flows were derived [8], applied [42] and analyzed [16]. HDG was then extended to incompressible [43,44] and compressible Navier–Stokes flows [48,49,45,53]. Additionally, HDG was used for elliptic interface problems [30] and [57,47] applied HDG for curved domains using elliptic and convection–diffusion equations. Computational aspects of HDG, such as its implementation and efficiency compared to existing methods, has been explored [59,62,32,1]. While it is still too early to judge whether HDG will be adopted for CFD, results so far show that HDG is competitive with both CG and FV approaches.

A challenge facing the wide-spread adoption of high-order schemes involves shock-capturing or slope limiting [61]. Without stabilizing a high-order numerical scheme, numerical oscillations can lead to instabilities. There is an abundance of literature on slope limiting, and here we do not provide an exhaustive list. For linear elements, critical contributions deal with Total Variation Bounded (TVB) limiters [9,12,13,10,11]. More recently, higher order limiters have received much attention with WENO-type approaches [51,63] and higher-order limiting [28,33,37,29]. Alternatively, stabilization can be added using filtering [26,59] or artificial dissipation [50,4,5]. Efforts to retain the full order of accuracy of the scheme away from the discontinuities have also been pursued [6]. However, there is not yet a standard approach that retains high-order accuracy while capturing sub-cell shocks, and research in this area is active, e.g. [58].

Our goals are to combine Projection methods with HDG Finite Elements, derive consistent projections for the HDG schemes, formulate an efficient numerical implementation using quadrature-free integration, and develop a selective nodal slope limiter to stabilize the method. We start by presenting the particular form of the Projection method we employ §2. We then derive the discrete finite element formulation (§3), obtaining the consistent correction steps from the discrete HDG predictions. We prove the consistency of our velocity correction on the HDG edge-space and derive a consistent HDG stability parameter for the pressure correction. Then, we explain the modifications needed in order to use an Implicit–Explicit Runge Kutta time integrator (§4). Next we address implementation issues concerning quadrature-free integration and slope limiting (§5). The implementation issues are verified in §6, and the overall scheme is evaluated in §7.

2. Governing equations and projection methods

In this section, we provide the equations we solve and the particular form of the projection method that we employ, specifically the rotational incremental pressure correction scheme [56].

We solve the non-dimensionalized unsteady incompressible Navier–Stokes equations on a simply connected domain Ω within a finite time interval $[0, T]$:

$$\begin{aligned} \frac{\partial \mathbf{v}}{\partial t} - \nabla \cdot \frac{1}{\mathrm{Re}} \nabla \mathbf{v} + \nabla p = -\nabla \cdot \mathbf{v}\mathbf{v} + \mathbf{f} \quad &\text{in } \Omega \times [0, T], \\ \nabla \cdot \mathbf{v} = 0 \quad &\text{in } \Omega \times [0, T], \\ \mathbf{v}|_{\partial\Omega} = \mathbf{g}_D \quad &\text{in } \partial\Omega \times [0, T], \\ \mathbf{v}|_{t=0} = \mathbf{v}_0 \quad &\text{in } \Omega, \end{aligned} \tag{1}$$

where $\mathbf{v} = [u, v, w]$ is the velocity, $p = \frac{1}{\rho_0} P$, P is the dynamic pressure, $\mathbf{f}$ are external body forces and ρ_0 is the mean density. The initial velocity conditions are $\mathbf{v}_0$ and we assume for now Dirichlet velocity boundary conditions (BCs) $\mathbf{g}_D$. We will also consider Boussinesq flows using the Boussinesq approximation. The only body forcing is then due to density, i.e. $\mathbf{f} = \mathbf{g}\rho$ where ρ is the density (temperature) perturbation. We then solve an additional equation for ρ, which is obtained from internal energy conservation,

$$\begin{aligned} \frac{\partial \rho}{\partial t} - \nabla \cdot \frac{1}{\mathrm{ReSc}} \nabla \rho = -\nabla \cdot \mathbf{v}\rho \quad &\text{in } \Omega \times [0, T], \\ \rho|_{\partial\Omega} = g_{D_\rho} \quad &\text{in } \partial\Omega \times [0, T], \\ \rho|_{t=0} = \rho_i \quad &\text{in } \Omega, \end{aligned} \tag{2}$$

where $\mathrm{Sc} = \frac{\nu}{\kappa}$ is the Schmidt number (ratio of kinematic viscosity ν to molecular diffusivity κ), g_{D_ρ} are Dirichlet density BCs and ρ_i is the initial condition for density. Alternatively, for ocean applications we can solve tracer equations for temperature and salinity, then calculate density through a state equation. However, if that equation is linear in salinity and temperature and the Schmidt number is the same for both, we can use (2) above. For the remainder of this section, we focus on momentum and continuity.

Now, since the non-linear term will be treated explicitly, we group it with the right-hand-side forcing term and it will not affect the splitting error. As such, we will only consider the Stokes equations henceforth.

$$\begin{aligned} \frac{\partial \mathbf{v}}{\partial t} - \nabla \cdot \frac{1}{\mathrm{Re}} \nabla \mathbf{v} + \nabla p = \mathbf{F}_{\partial t} \quad &\text{in } \Omega \times [0, T], \\ \nabla \cdot \mathbf{v} = 0 \quad &\text{in } \Omega \times [0, T], \\ \mathbf{v}|_{\partial\Omega} = \mathbf{g}_D \quad &\text{in } \partial\Omega \times [0, T], \\ \mathbf{v}|_{t=0} = \mathbf{v}_0 \quad &\text{in } \Omega, \end{aligned} \tag{3}$$

where $\mathbf{F}_{\partial t} = -\nabla \cdot \mathbf{v}\mathbf{v} + \mathbf{f}$.

The un-split time-discretization of the coupled eqs. (3) is given by

$$\begin{aligned} \frac{\mathbf{v}^{k+1}}{a\Delta t} - \nabla \cdot \frac{1}{\mathrm{Re}} \nabla \mathbf{v}^{k+1} + \nabla p^{k+1} &= \mathbf{F}^{k,k+1}, \\ \nabla \cdot \mathbf{v}^{k+1} &= 0 \\ \mathbf{v}|_{\partial\Omega}^{k+1} &= \mathbf{g}_D, \\ \mathbf{v}|_{t=0} &= \mathbf{v}_0, \end{aligned} \tag{4}$$

where a single stage in time is considered, a is some constant associated with it, and $\mathbf{F}^{k,k+1}$ contains the explicitly integrated terms (including old values of $\mathbf{v}$) and the possibly implicit right-hand-side forcing terms $\mathbf{F}_{\partial t}$ (see §4).

The time-split discretization corresponding to (4), with the rotational incremental pressure correction scheme [56,23], starts by solving for the predictor velocity $\bar{\mathbf{v}}^{k+1}$ using an old or guessed value for the pressure gradient:

$$\frac{\bar{\mathbf{v}}^{k+1}}{a\Delta t} - \nabla \cdot \frac{1}{\mathrm{Re}} \nabla \bar{\mathbf{v}}^{k+1} + \nabla p^k = \mathbf{F}^{k,k+1}, \tag{5}$$

$$\bar{\mathbf{v}}|_{\partial\Omega}^{k+1} = \mathbf{g}_D, \tag{6}$$

$$\mathbf{v}|_{t=0} = \mathbf{v}_0, \tag{7}$$

$$p|_{t=0} = p_0 . \tag{8}$$

Note, here we only considered Dirichlet velocity BCs, for simplicity, but the method extends to other BCs [58]. Next, a Poisson equation is solved for the pressure corrector δp^{k+1} (note, negative signs are added so that this derivation matches our numerical implementation in §3):

$$-\nabla^2 \delta p^{k+1} = -\frac{\nabla \cdot \bar{\mathbf{v}}^{k+1}}{a\Delta t}, \tag{9}$$

$$\left.\frac{\partial \delta p^{k+1}}{\partial \hat{\mathbf{n}}}\right|_{\partial\Omega} = 0. \tag{10}$$

Finally, the velocities and pressure need to be corrected

$$\mathbf{v}^{k+1} = \bar{\mathbf{v}}^{k+1} - a\Delta t \nabla \delta p^{k+1}, \tag{11}$$

$$p^{k+1} = p^k + \delta p^{k+1} - \frac{1}{\mathrm{Re}} \nabla \cdot \bar{\mathbf{v}}^{k+1}. \tag{12}$$

We first note that the BC for the pressure-correction (10) comes from (11) and a normal Dirichlet BC on both $\mathbf{v}$ and $\bar{\mathbf{v}}$. Hence, the $\mathbf{v}$ velocity satisfies the normal Dirichlet BCs ($\mathbf{v} \cdot \hat{\mathbf{n}}|_{\partial\Omega}^{k+1} = \mathbf{g}_D \cdot \hat{\mathbf{n}}$), and is divergence free ($\nabla \cdot \mathbf{v} = 0$), while $\bar{\mathbf{v}}$ satisfies both the normal and tangential Dirichlet BCs ($\bar{\mathbf{v}}|_{\partial\Omega}^{k+1} = \mathbf{g}_D$) but is not divergence free ($\nabla \cdot \bar{\mathbf{v}} \neq 0$). We then note that the final term in (12) is known as the rotational-correction term. Eqs. (11)–(12) can be derived by taking the difference between the un-split (implicit pressure) eqs. (4) and split eqs. (5)–(10). Analogous differences of HDG discretizations will be used to obtain the final corrections for the HDG predictor schemes (§3.2.4–§3.2.6).

3. Spatial discretization of time-split equations using HDG

In this section we spatially discretize eqs. (5)–(12). We first define our notation and describe the premise behind HDG methods. Then we complete the basic derivations of the scheme. Later, we obtain and justify the final HDG velocity and pressure corrections, the HDG rotational correction, and a consistent HDG stability parameter for the pressure Poisson equation.

3.1. Finite element definitions and notation

We let $\mathcal{T}_h = \cup K_i$ be a finite collection of non-overlapping elements, K_i, that discretizes the entire computational domain Ω (Fig. 2). Also, let $\partial\mathcal{T}_h = \{\partial K : K \in \mathcal{T}_h\}$ be the set of interfaces of all elements, where ∂K is the boundary of element K. For two elements sharing an edge K^+ and K^-, we define $e = \partial K^+ \cap \partial K^- \neq \emptyset$ as the unique interior interface between elements K^+ and K^-. For a single element K belonging to $\mathcal{T}_h$, if $e = \partial K \cap \partial\Omega \neq \emptyset$ it is a boundary interface. Let ε° and ε^∂ denote the set of unique interior and boundary interfaces, respectively, such that $\varepsilon = \varepsilon^\circ \cup \varepsilon^\partial$. We note that in the interior $\partial\mathcal{T}_h$ contains two interfaces, ∂K^+ and ∂K^-, at the same location (one for each element sharing the edge), whereas the set ε only contains a single interface, e, at the same location.

K^+ and K^- have outward pointing normals $\hat{\mathbf{n}}^+$ and $\hat{\mathbf{n}}^-$, respectively. We then let vector and scalar quantities $[\mathbf{a}^\pm, c^\pm]$ be the traces (i.e. the projections) of $[\mathbf{a}, c]$ on the interface e from the interior of $K^\pm$. The "mean" value $\{\!\{\bullet\}\!\}$ and "jumps" $[\![\bullet]\!]$ on the interior interfaces $e \in \varepsilon^\circ$ for scalar and vector quantities are then defined as

$$\{\!\{\mathbf{a}\}\!\} = (\mathbf{a}^+ + \mathbf{a}^-)/2, \qquad \{\!\{c\}\!\} = (c^+ + c^-)/2,$$

$$[\![\mathbf{a} \cdot \hat{\mathbf{n}}]\!] = \mathbf{a}^+ \cdot \hat{\mathbf{n}}^+ + \mathbf{a}^- \cdot \hat{\mathbf{n}}^-, \qquad [\![c\hat{\mathbf{n}}]\!] = c^+\hat{\mathbf{n}}^+ + c^-\hat{\mathbf{n}}^-.$$

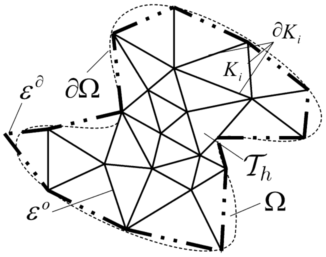

Fig. 2. Notation for domain discretization.

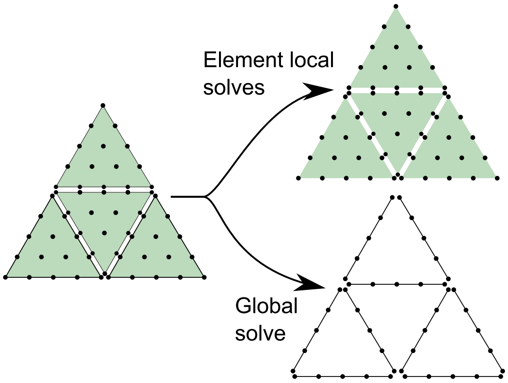

Fig. 3. The HDG method splits the solution of the element local equations from the solution of the globally coupled problem for the boundary conditions.

On the set of boundary interfaces $e \in \varepsilon^\partial$ (with outward facing normal $\hat{\mathbf{n}}$ on $\partial\Omega$) we set these mean and jump quantities as

$$\begin{aligned} \{\{\mathbf{a}\}\} &= \mathbf{a}, & \{\{c\}\} &= c, \\ [\![\mathbf{a}\cdot\hat{\mathbf{n}}]\!] &= \mathbf{a}\cdot\hat{\mathbf{n}}, & [\![c\hat{\mathbf{n}}]\!] &= c\hat{\mathbf{n}}, \end{aligned}$$

since here $\mathbf{a}$ and c are single-valued. Note that the jump will be zero for a continuous function.

A main difference between CG and DG lies in the approximation subspaces used. DG uses bases that are in $L^2(\Omega)$ while CG uses bases that are in $H^1(\Omega)$, that is, the function has to be continuous across elements.

Let $\mathcal{P}^{\mathrm{p}}(D)$ denote the set of polynomials of maximum degree p existing on a domain D. For example, we will be using $\mathrm{p} = 2$ to denote a second degree polynomial basis, which will result in a 3rd order accurate scheme. We define the discontinuous finite element bases we use on the element for scalars, vectors, and tensors, respectively, as

$$\begin{aligned} &\left\{\theta \in L^2(\Omega) : \theta\,|_K \in \mathcal{P}^{\mathrm{p}}(K), \forall K \in \mathcal{T}_h\right\} \\ &\left\{\boldsymbol{\theta} \in (L^2(\Omega))^d : \boldsymbol{\theta}\,|_K \in (\mathcal{P}^{\mathrm{p}}(K))^d, \forall K \in \mathcal{T}_h\right\} \\ &\left\{\boldsymbol{\Theta} \in (L^2(\Omega))^{d\times d} : \boldsymbol{\Theta}\,|_K \in (\mathcal{P}^{\mathrm{p}}(K))^{d\times d}, \forall K \in \mathcal{T}_h\right\}. \end{aligned}$$

To use the HDG method, we will also require the traced finite element spaces existing on the unique interfaces ε

$$\begin{aligned} &\left\{\theta_\varepsilon \in L^2(\Omega) : \theta_\varepsilon\,|_e \in \mathcal{P}^{\mathrm{p}}(e), \forall e \in \varepsilon\right\}, \\ &\left\{\boldsymbol{\theta}_\varepsilon \in (L^2(\Omega))^d : \boldsymbol{\theta}_\varepsilon\,|_e \in (\mathcal{P}^{\mathrm{p}}(e))^d, \forall e \in \varepsilon\right\}, \\ &\left\{\boldsymbol{\Theta}_\varepsilon \in (L^2(\Omega))^{d\times d} : \boldsymbol{\Theta}_\varepsilon\,|_e \in (\mathcal{P}^{\mathrm{p}}(e))^{d\times d}, \forall e \in \varepsilon\right\}. \end{aligned}$$

We also set $\{\theta_\varepsilon = \mathsf{P}\mathbf{g}_D \text{ on } \partial\Omega\}$, where P is the L^2 projection of the BC $\mathbf{g}_D$ into the same space as θ_ε. Note that θ_ε is continuous on the interface, e, shared by K^+ and K^-, but discontinuous at the borders between different interfaces (that is, for a 1D-line-interface, discontinuities are the end-points of the line, see Fig. 3).

Finally we define the inner products over continuous domains $D \in \mathbb{R}^d$ and $\partial D \in \mathbb{R}^{d-1}$ as

$$(\mathbf{a},\mathbf{b})_D = \int_D \mathbf{a}\cdot\mathbf{b}\,\mathrm{d}D \qquad (c,d)_D = \int_D cd\,\mathrm{d}D \tag{13}$$

$$\langle\mathbf{a},\mathbf{b}\rangle_{\partial D} = \int_{\partial D} \mathbf{a}\cdot\mathbf{b}\,\mathrm{d}\partial D \qquad \langle c,d\rangle_{\partial D} = \int_{\partial D} cd\,\mathrm{d}\partial D \tag{14}$$

for vector functions $\mathbf{a}$, $\mathbf{b}$ and scalar functions c, d. Over discontinuous domains we also define

$$(\mathbf{a},\mathbf{b})_{\mathcal{T}_h} = \sum_{K\in\mathcal{T}_h} (\mathbf{a},\mathbf{b})_K, \qquad \langle c,d\rangle_{\partial\mathcal{T}_h} = \sum_{K\in\mathcal{T}_h} \langle c,d\rangle_{\partial K}, \tag{15}$$

for vector functions $\mathbf{a}$, $\mathbf{b}$ defined on $\mathcal{T}_h$, and scalar functions c, d defined on $\partial\mathcal{T}_h$. We also utilize the additional inner product on the HDG edge space

$$\langle \mathbf{a}, \boldsymbol{\theta}_\varepsilon \rangle_\varepsilon = \sum_{e \in \varepsilon} \langle \mathbf{a}, \boldsymbol{\theta}_\varepsilon \rangle_e \qquad \langle c, \theta_\varepsilon \rangle_\varepsilon = \sum_{e \in \varepsilon} \langle c, \theta_\varepsilon \rangle_e \tag{16}$$

for vector or scalar functions $\mathbf{a}$, c defined on ε.

To understand how HDG fluxes are specified over the interfaces ε, it is useful to grasp an underlying premise of HDG methods. The premise is that one can solve the equations of interest locally on an element as long as the initial and boundary conditions are properly specified. While the initial conditions for an element are specified as part of the problem, the BCs on every element edge are not given *a priori*. The HDG method provides a global system of equations for these unknown BCs, rendering them single-valued in the HDG edge-spaces. To do so, the fluxes across element interfaces are first parameterized and made a function of single-valued $\lambda/\boldsymbol{\lambda}$ variables (see §3.2) that live on the HDG edge-spaces. These $\lambda/\boldsymbol{\lambda}$ variables are part of the numerical scheme and are critical to its consistency, stability, and efficiency. In some equations, they are the numerical fluxes or trace functions that connect discontinuous elements, and in others, they are part of the stabilization terms of such fluxes. Since the fluxes belong to the space of continuous edge-space functions (e.g. θ_ε, $\boldsymbol{\theta}_\varepsilon$ or $\boldsymbol{\Theta}_\varepsilon$), one can enforce their conservation across elements, which leads to global flux conservation (i.e. the fluxes on the interior ε° balance the fluxes at the boundary ε^∂). The numerical traces are then conservative in the sense of [2]. To obtain the global system of equations for the HDG edge-space variables, the solution on the interior of an element is expressed in terms of these new variables that represent the BCs for that element. The globally-coupled equations are then found by solving for BCs that give conservative fluxes (that is, the same flux for both elements on either side of an edge). As such, the solutions of the element-local equations are split from those of the globally-coupled equations for the edge degrees of freedom (Fig. 3). The HDG method thus benefits from new implementations, e.g. see [40,57,58].

For our set of equations (5)–(6) and (9)–(10), we need two new variables, $\bar{\boldsymbol{\lambda}}$ for (5) and $\lambda_{\delta p}$ for (9), which live in the same spaces as $\boldsymbol{\theta}_\varepsilon$ and θ_ε, respectively. That is, $\bar{\boldsymbol{\lambda}}$ and $\lambda_{\delta p}$ only exist on the new HDG edge-space, and do not have a value inside the element (i.e. no interior support).

Additionally, we note that the original Stokes system only required knowledge of the velocity initial conditions, average pressure, and velocity BCs to be solvable. However, for the time-split system using projection methods (5)–(12) we need to specify the initial velocity and pressure, and BCs for the velocity and pressure-correction. While this may appear to be an over-specification due to the additional BCs for the pressure-correction, the BCs for velocity and pressure are intimately related, and require careful treatment to be numerically consistent. The proper specification of these HDG edge fluxes using Projection methods is one of our contributions.

3.2. Discrete HDG projection-method equations and their derivations

We first introduce the additional DG unknown variables $\mathbf{Q} = \frac{1}{\text{Re}} \nabla \mathbf{v}$ and $\mathbf{q}_{\delta p} = \nabla \delta p$. We then rewrite (5) as

$$\text{Re}\bar{\mathbf{Q}}^{k+1} - \nabla \bar{\mathbf{v}}^{k+1} = 0, \tag{17}$$

$$\frac{\bar{\mathbf{v}}^{k+1}}{a\Delta t} - \nabla \cdot \bar{\mathbf{Q}}^{k+1} + \nabla p^k = \mathbf{F}^{k,k+1}, \tag{18}$$

(9) as

$$\mathbf{q}_{\delta p}^{k+1} - \nabla \delta p^{k+1} = 0, \tag{19}$$

$$-\nabla \cdot \mathbf{q}_{\delta p}^{k+1} = -\frac{\nabla \cdot \bar{\mathbf{v}}^{k+1}}{a\Delta t}, \tag{20}$$

and expect that final corrections (11) and (12) will remain close to

$$\mathbf{v}^{k+1} = \bar{\mathbf{v}}^{k+1} - a\Delta t \mathbf{q}_{\delta p}^{k+1}, \tag{21}$$

$$p^{k+1} = p^k + \delta p^{k+1} - \frac{1}{\text{Re}} \nabla \cdot \bar{\mathbf{v}}^{k+1}. \tag{22}$$

The time-discretized split equations (17)–(22) are the starting point of one of our main results, the new HDG projection-method discretization.

Next, we obtain these equations with consistent HDG edge-space updates (§3.2.1) and express fluxes in terms of element interior variables (§3.2.2). To complete derivations, we first state an HDG discretization for the un-split case (§3.2.3). We then use its differences with the HDG discretization of the split eqs. (17)–(20) to derive the HDG forms of eqs. (21)–(22) and formally justify these element-local velocity and pressure corrections, and their consistent edge-space updates (§3.2.4–§3.2.5). In §3.2.6, we derive a consistency relation between stability parameters for the split scheme. All equations, including HDG fluxes, are recapped in Fig. 4, in the order in which the systems of equations are solved. The solid boxes give the discretized equations that need to be solved while the dashed boxes give additional information about the fluxes that are not explicitly a part of the numerical discretization. Steps 1 and 2 require the numerical solution of a system of global/local HDG equations. For the solutions steps of HDG systems, we refer to [40,58].

1. Velocity predictor (momentum equations)

Element-Local equations:

$$\left((\mathrm{Re})\bar{\mathbf{Q}}^{k+1}, \boldsymbol{\Theta}\right)_K - \left(\nabla\bar{\mathbf{v}}^{k+1}, \boldsymbol{\Theta}\right)_K + \langle\bar{\mathbf{v}}^{k+1}, \hat{\mathbf{n}}\cdot\boldsymbol{\Theta}\rangle_{\partial K} = \left\langle\bar{\boldsymbol{\lambda}}^{k+1}, \hat{\mathbf{n}}\cdot\boldsymbol{\Theta}\right\rangle_{\partial K} \quad (22)$$

$$\left(\frac{\bar{\mathbf{v}}^{k+1}}{a\Delta t}, \boldsymbol{\theta}\right)_K - \left(\nabla\cdot\bar{\mathbf{Q}}^{k+1}, \boldsymbol{\theta}\right)_K + \langle\tau\bar{\mathbf{v}}^{k+1}, \boldsymbol{\theta}\rangle_{\partial K} = \left\langle\tau\bar{\boldsymbol{\lambda}}^{k+1}, \boldsymbol{\theta}\right\rangle_{\partial K} - \left(\nabla p^k, \boldsymbol{\theta}\right)_K + \left(\mathbf{F}^{k,k+1}, \boldsymbol{\theta}\right)_K \quad (23)$$

Edge-space global flux conservation equations:

$$\left\langle\left[\left[\bar{\mathbf{Q}}^{k+1}\cdot\hat{\mathbf{n}} - \tau\left(\bar{\mathbf{v}}^{k+1} - \bar{\boldsymbol{\lambda}}^{k+1}\right)\right]\right], \boldsymbol{\theta}_\varepsilon\right\rangle_\varepsilon = \left\langle\left[\left[p^k\hat{\mathbf{n}}\right]\right], \boldsymbol{\theta}_\varepsilon\right\rangle_\varepsilon + \langle\mathbf{g}_N, \boldsymbol{\theta}_\varepsilon\rangle_\varepsilon \quad (25)$$

$$\bar{\boldsymbol{\lambda}}\big|^{k+1}_{\varepsilon^\partial_D} = \mathbf{g}_D \quad (26)$$

Flux definitions:

$$\hat{\bar{\mathbf{v}}}^{k+1} = \begin{cases} \bar{\boldsymbol{\lambda}}^{k+1}, & \text{on } \varepsilon^\circ \\ \mathsf{P}\mathbf{g}_D, & \text{on } \varepsilon^\partial_D \end{cases} \quad (36)$$

$$\widehat{\bar{\mathbf{Q}}}^{k+1} - \widehat{p}^k\mathbf{I} = \bar{\mathbf{Q}}^{k+1} - p^k\mathbf{I} - \tau\left(\bar{\mathbf{v}}^{k+1} - \hat{\bar{\mathbf{v}}}^{k+1}\right)\hat{\mathbf{n}} \quad (37)$$

2. Pressure corrector (to enforce continuity)

Element-Local equations:

$$\left(\mathbf{q}^{k+1}_{\delta p}, \boldsymbol{\theta}\right)_K - \left(\nabla\delta p^{k+1}, \boldsymbol{\theta}\right)_K + \langle\delta p^{k+1}, \hat{\mathbf{n}}\cdot\boldsymbol{\theta}\rangle_{\partial K} = \left\langle\lambda^{k+1}_{\delta p}, \hat{\mathbf{n}}\cdot\boldsymbol{\theta}\right\rangle_{\partial K} \quad (27)$$

$$-\left(\nabla\cdot\mathbf{q}^{k+1}_{\delta p}, \theta\right)_K + \langle\tau_p\delta p^{k+1}, \theta\rangle_{\partial K} = \left\langle\tau_p\lambda^{k+1}_{\delta p}, \theta\right\rangle_{\partial K} - \left(\frac{\nabla\cdot\bar{\mathbf{v}}^{k+1}}{a\Delta t}, \theta\right)_K - \left\langle\frac{(\bar{\boldsymbol{\lambda}}^{k+1} - \bar{\mathbf{v}}^{k+1})\cdot\hat{\mathbf{n}}}{a\Delta t}, \theta\right\rangle_{\partial K} \quad (28)$$

Edge-space global flux conservation equations:

$$\left\langle\left[\left[\mathbf{q}^{k+1}_{\delta p}\cdot\hat{\mathbf{n}} - \tau_p\left(\delta p^{k+1} - \lambda^{k+1}_{\delta p}\right)\right]\right], \theta_\varepsilon\right\rangle_\varepsilon = \left\langle g_{N_p}, \theta_\varepsilon\right\rangle_\varepsilon \quad (30)$$

$$\lambda_{\delta p}\big|^{k+1}_{\varepsilon^\partial_D} = g_{D_p} \quad (31)$$

Flux definitions:

$$\widehat{\delta p}^{k+1} = \begin{cases} \lambda^{k+1}_{\delta p}, & \text{on } \varepsilon^\circ \\ \mathsf{P}g_{D_p}, & \text{on } \varepsilon^\partial_{D_p} \end{cases} \quad (38)$$

$$\widehat{\mathbf{q}_{\delta p}}^{k+1} = \mathbf{q}_{\delta p}{}^{k+1} - \tau_p\left(\delta p^{k+1} - \widehat{\delta p}^{k+1}\right)\hat{\mathbf{n}} \quad (39)$$

$$\hat{\bar{\mathbf{v}}}^{k+1}_\star = \bar{\boldsymbol{\lambda}}^{k+1} \quad (40)$$

3. Velocity and pressure corrections

Element-Local correction:

$$\mathbf{v}^{k+1} = \bar{\mathbf{v}}^{k+1} - a\Delta t\mathbf{q}^{k+1}_{\delta p} \quad (32)$$

$$\left(p^{k+1}, \theta\right)_K = \left(p^k + \delta p^{k+1}, \theta\right)_K - \frac{1}{\mathrm{Re}}\left(\nabla\cdot\bar{\mathbf{v}}^{k+1}, \theta\right)_K - \frac{1}{\mathrm{Re}}\left\langle(\boldsymbol{\lambda}^{k+1} - \mathbf{v}^{k+1})\cdot\hat{\mathbf{n}}, \theta\right\rangle_{\partial K} \quad (35)$$

Edge-space correction: $\boldsymbol{\lambda}^{k+1} = \bar{\boldsymbol{\lambda}}^{k+1} - a\Delta t\mathbf{q}^{k+1}_{\delta p} + a\Delta t\tau_p\left(\delta p^{k+1} - \lambda^{k+1}_{\delta p}\right)\hat{\mathbf{n}}$ (34)

Consistent HDG stability parameter:

$$\tau_p = \frac{1}{a\Delta t\tau} \quad (61)$$

Implicit flux definition:

$$\boldsymbol{\lambda}^{k+1} = \left\{\left\{\mathbf{v}^{k+1}\right\}\right\} - \frac{1}{2\tau}\left[\left[\bar{\mathbf{Q}}^{k+1}\cdot\hat{\mathbf{n}}\right]\right] + \frac{1}{2\tau}\left[\left[p^k\hat{\mathbf{n}}\right]\right] + a\Delta t\frac{\tau_p}{2}\left[\left[\delta p^{k+1}\hat{\mathbf{n}}\right]\right] \quad (60)$$

Fig. 4. New HDG–projection method scheme. Plain boxes denote the main equations, while dashed boxes give flux definitions. Element-local equations are arranged such that locally-calculated quantities are on the left of the equal sign, while globally calculated (i.e. $\boldsymbol{\lambda}$'s) and known (or given) quantities are on the right.

3.2.1. HDG spatial discretization: equations

We first multiply eqs. (17)–(20) by their test functions, and integrate over the domain. Following [27], the terms that correspond to fluxes (divergence terms for the conservative form of eqs. (17)–(20)) are integrated by parts twice, substituting non-local and local fluxes sequentially. This leads to edge integrals that contain differences between single-valued HDG edge-space fluxes (denoted by $\hat{\bullet}$) and element interior fluxes. Hence, we utilize the strong form [27], in part for implementation considerations (see §5.1). We then enforce global flux-conservation, i.e. continuity of the normal component of the total fluxes on the interior edge-space (inter-element boundaries) and at domain boundaries (boundary conditions). For our velocity (17)–(18) and pressure corrector (19)–(20) predictions, this implies that, on the edge-space, the total normal stress and pressure corrector are conserved and single-valued functions of $\bar{\boldsymbol{\lambda}}$ and $\lambda_{\delta p}$. The edge integrals are indeed parameterized in terms of HDG stabilization functions (see §3.2.2). Since advection fluxes depend on velocity only, they can be made single-valued on the edge-space by construction: element-local fluxes are then set equal to edge fluxes and they cancel out in the normal flux conservation equations. If the advection term is evaluated explicitly (as we do, see (17)), its strong-form still needs to be utilized to ensure continuity of normal advection fluxes. We give some guidance on different choices of advective fluxes in §3.2.2, but for stability of advection schemes with the HDG framework see for example [41]. With all of the above, we obtain the new element-local equations (with the HDG fluxes substituted) along with their global flux-conservation equations.

The strong-form HDG discretization of (17)–(18) provides the element-local equations for $\bar{\mathbf{v}}^{k+1}$,

$$\left((\mathrm{Re})\bar{\mathbf{Q}}^{k+1},\boldsymbol{\Theta}\right)_K-\left(\nabla\bar{\mathbf{v}}^{k+1},\boldsymbol{\Theta}\right)_K+\left\langle\bar{\mathbf{v}}^{k+1},\hat{\mathbf{n}}\cdot\boldsymbol{\Theta}\right\rangle_{\partial K}=\left\langle\bar{\boldsymbol{\lambda}}^{k+1},\hat{\mathbf{n}}\cdot\boldsymbol{\Theta}\right\rangle_{\partial K}, \tag{23}$$

$$\left(\frac{\bar{\mathbf{v}}^{k+1}}{a\Delta t},\boldsymbol{\theta}\right)_K-\left(\nabla\cdot\bar{\mathbf{Q}}^{k+1},\boldsymbol{\theta}\right)_K+\left\langle\tau\bar{\mathbf{v}}^{k+1},\boldsymbol{\theta}\right\rangle_{\partial K}=\left\langle\tau\bar{\boldsymbol{\lambda}}^{k+1},\boldsymbol{\theta}\right\rangle_{\partial K}-\left(\nabla p^k,\boldsymbol{\theta}\right)_K+\left(\mathbf{F}^{k,k+1},\boldsymbol{\theta}\right)_K, \tag{24}$$

where $\bar{\mathbf{Q}}^{k+1}$, $\bar{\mathbf{v}}^{k+1}$, and p^k live in the spaces $\boldsymbol{\Theta}$, $\boldsymbol{\theta}$, and θ, respectively; $\mathbf{F}^{k,k+1}$ lives in $\boldsymbol{\theta}$; and $\bar{\boldsymbol{\lambda}}^{k+1}$ lives in $\boldsymbol{\theta}_\varepsilon$. In (23)–(24), we defined the single-valued $\widehat{\bar{\mathbf{v}}}^{k+1}=\bar{\boldsymbol{\lambda}}^{k+1}$ and substituted an HDG stabilization vector-valued function for the total normal edge-space flux

$$\left\langle-(\widehat{\bar{\mathbf{Q}}}^{k+1}-\bar{\mathbf{Q}}^{k+1})\cdot\hat{\mathbf{n}}+(\widehat{p}^k-p^k)\hat{\mathbf{n}},\boldsymbol{\theta}\right\rangle_{\partial K}=-\left\langle\tau(\bar{\boldsymbol{\lambda}}^{k+1}-\bar{\mathbf{v}}^{k+1}),\boldsymbol{\theta}\right\rangle_{\partial K}, \tag{25}$$

where τ is the stability parameter, of $O(1/\mathrm{Re})$ for our non-dimensional equations (§3.2.6). The flux defined via (25) is key to reducing the number of global degrees of freedom of the problem (and therefore the computational effort), as well as securing stabilization via the τ parameter. This flux definition leads to the global flux-conservation equations for $\bar{\boldsymbol{\lambda}}^{k+1}$:

$$\begin{aligned}\left\langle\left[\!\left[\widehat{\bar{\mathbf{Q}}}^{k+1}\cdot\hat{\mathbf{n}}\right]\!\right],\boldsymbol{\theta}_\varepsilon\right\rangle_\varepsilon&=\langle\mathbf{g}_N,\boldsymbol{\theta}_\varepsilon\rangle_\varepsilon\\ \Rightarrow\left\langle\left[\!\left[\bar{\mathbf{Q}}^{k+1}\cdot\hat{\mathbf{n}}-\tau\left(\bar{\mathbf{v}}^{k+1}-\bar{\boldsymbol{\lambda}}^{k+1}\right)\right]\!\right],\boldsymbol{\theta}_\varepsilon\right\rangle_\varepsilon&=\left\langle\left[\!\left[p^k\hat{\mathbf{n}}\right]\!\right],\boldsymbol{\theta}_\varepsilon\right\rangle_\varepsilon+\langle\mathbf{g}_N,\boldsymbol{\theta}_\varepsilon\rangle_\varepsilon, &(26)\\ \bar{\boldsymbol{\lambda}}|^{k+1}_{\varepsilon^\partial_D}&=\mathbf{g}_D &(27)\end{aligned}$$

where $\mathbf{g}_D$ and $\mathbf{g}_N$ are the provided Dirichlet and Neumann BCs for the momentum equations, respectively (other BCs can also be used, e.g. [44,58]).

The strong-form HDG discretization of (19)–(20) provides the element-local equations for δp^{k+1},

$$\left(\mathbf{q}^{k+1}_{\delta p},\boldsymbol{\theta}\right)_K-\left(\nabla\delta p^{k+1},\boldsymbol{\theta}\right)_K+\left\langle\delta p^{k+1},\hat{\mathbf{n}}\cdot\boldsymbol{\theta}\right\rangle_{\partial K}=\left\langle\lambda^{k+1}_{\delta p},\hat{\mathbf{n}}\cdot\boldsymbol{\theta}\right\rangle_{\partial K}, \tag{28}$$

$$-\left(\nabla\cdot\mathbf{q}^{k+1}_{\delta p},\theta\right)_K+\left\langle\tau_p\delta p^{k+1},\theta\right\rangle_{\partial K}=\left\langle\tau_p\lambda^{k+1}_{\delta p},\theta\right\rangle_{\partial K}-\left(\frac{\nabla\cdot\bar{\mathbf{v}}^{k+1}}{a\Delta t},\theta\right)_K-\left\langle\frac{(\bar{\boldsymbol{\lambda}}^{k+1}-\bar{\mathbf{v}}^{k+1})\cdot\hat{\mathbf{n}}}{a\Delta t},\theta\right\rangle_{\partial K}, \tag{29}$$

where $\mathbf{q}^{k+1}_{\delta p}$, δp^{k+1}, and $\lambda^{k+1}_{\delta p}$ live in $\boldsymbol{\theta}$, θ, and θ_ε, respectively. In (28)–(29), we defined the single-valued $\widehat{\delta p}^{k+1}=\lambda^{k+1}_{\delta p}$ and substituted an HDG stabilization scalar function for the total normal edge-space flux

$$\left\langle-(\widehat{\mathbf{q}_{\delta p}}^{k+1}-\mathbf{q}^{k+1}_{\delta p})\cdot\hat{\mathbf{n}},\theta\right\rangle_{\partial K}=-\left\langle\tau_p(\lambda^{k+1}_{\delta p}-\delta p^{k+1}),\theta\right\rangle_{\partial K}, \tag{30}$$

where $\tau_p=\frac{1}{a\Delta t\tau}$ is the HDG stability parameter for the pressure-correction (derivation in §3.2.6). In (29), we also used $\widehat{\mathbf{v}}^{k+1}_\star=\bar{\boldsymbol{\lambda}}^{k+1}$ by consistency with $\widehat{\bar{\mathbf{v}}}^{k+1}$ used in (24). The global flux-conservation equations for $\lambda^{k+1}_{\delta p}$ are thus

$$\begin{aligned}\left\langle\left[\!\left[\widehat{\mathbf{q}_{\delta p}}^{k+1}\cdot\hat{\mathbf{n}}\right]\!\right],\theta_\varepsilon\right\rangle_\varepsilon&=\left\langle g_{N_p},\theta_\varepsilon\right\rangle_\varepsilon\\ \Rightarrow\left\langle\left[\!\left[\mathbf{q}^{k+1}_{\delta p}\cdot\hat{\mathbf{n}}-\tau_p\left(\delta p^{k+1}-\lambda^{k+1}_{\delta p}\right)\right]\!\right],\theta_\varepsilon\right\rangle_\varepsilon&=\left\langle g_{N_p},\theta_\varepsilon\right\rangle_\varepsilon, &(31)\\ \lambda_{\delta p}|^{k+1}_{\varepsilon^\partial_D}&=g_{D_p} &(32)\end{aligned}$$

where g_{D_p} and g_{N_p} are the provided Dirichlet and Neumann BCs for the pressure-correction (and these are normally all zero Neumann, see eq. (10)).

We will show in §3.2.4 that the final correction for the element-local velocity remains as (21) (except that spatially discrete HDG variables are used)

$$\mathbf{v}^{k+1}=\bar{\mathbf{v}}^{k+1}-a\Delta t\mathbf{q}^{k+1}_{\delta p}. \tag{33}$$

Logically, a final correction on the velocity edge-space is also required. To be consistent with the HDG pressure-correction (28)–(29), we find that this new correction is $\widehat{\mathbf{v}}^{k+1}_{\mathrm{cor}}=-a\Delta t\widehat{\mathbf{q}_{\delta p}}^{k+1}=-a\Delta t\left(\mathbf{q}^{k+1}_{\delta p}-\tau_p(\delta p^{k+1}-\lambda^{k+1}_{\delta p})\hat{\mathbf{n}}\right)$, where $\widehat{\delta p}^{k+1}=\lambda^{k+1}_{\delta p}$ still. The final corrected edge-space velocity is thus

$$\boldsymbol{\lambda}^{k+1} = \widehat{\bar{\mathbf{v}}}_\star^{k+1} + \widehat{\mathbf{v}}_{\text{cor}}^{k+1} = \bar{\boldsymbol{\lambda}}^{k+1} - a\Delta t\widehat{\mathbf{q}_{\delta p}}^{k+1} \tag{34}$$

$$= \bar{\boldsymbol{\lambda}}^{k+1} - a\Delta t\mathbf{q}_{\delta p}{}^{k+1} + a\Delta t\tau_p\left(\delta p^{k+1} - \lambda_{\delta p}^{k+1}\right)\hat{\mathbf{n}}, \tag{35}$$

where again $\widehat{\bar{\mathbf{v}}}_\star^{k+1} = \bar{\boldsymbol{\lambda}}^{k+1}$. In §3.2.5, the final discretized pressure correction will be shown to be as a standard DG on (12) but with updated HDG fluxes,

$$\left(p^{k+1},\theta\right)_K = \left(p^k + \delta p^{k+1},\theta\right)_K - \frac{1}{\mathrm{Re}}\left(\nabla\cdot\bar{\mathbf{v}}^{k+1},\theta\right)_K - \frac{1}{\mathrm{Re}}\left\langle(\boldsymbol{\lambda}^{k+1} - \mathbf{v}^{k+1})\cdot\hat{\mathbf{n}},\theta\right\rangle_{\partial K}. \tag{36}$$

We note that this final pressure (36) with the HDG rotational correction is only needed for the next stage in the local pressure gradient in (24) and jump terms in (26). The final corrected pressure on the edge-space is not needed in the present scheme: it is within the stabilization term in (24). Variations on the above are provided in [58,35]. All of the HDG fluxes are given next.

3.2.2. HDG spatial discretization: fluxes

To obtain the above HDG discretization, we needed to define the HDG edge-space fluxes (numerical traces) $\widehat{\bar{\mathbf{v}}}^{k+1}$, $\widehat{\bar{\mathbf{v}}}_\star^{k+1}$, $\widehat{\delta p}^{k+1}$, $\widehat{\bar{\mathbf{Q}}}^{k+1}$, $\widehat{p}^k$ and $\widehat{\mathbf{q}_{\delta p}}^{k+1}$. In §3.2.1, these single-valued $\hat{\bullet}$ fluxes arose in the edge integral terms of the strong-form HDG equations. For the equations added to define HDG gradients, the fluxes are simply edge variables and they were set to their $\boldsymbol{\lambda}$'s. For the momentum and pressure equations proper, the differences between the single-valued edge fluxes and interior fluxes were set to stabilization functions, $\tau(\bar{\boldsymbol{\lambda}}^{k+1} - \bar{\mathbf{v}}^{k+1})$ and $\tau_p(\lambda_{\delta p}^{k+1} - \delta p^{k+1})$, respectively. In other words, the total single-valued HDG fluxes were set equal to the corresponding element interior flux plus the stabilization term. These expressions are now given.

Our flux definitions for $\widehat{\bar{\mathbf{v}}}^{k+1}$ and total stress $\widehat{\bar{\mathbf{Q}}}^{k+1} - \widehat{p}^k\mathbf{I}$ that appeared in (23)–(24) at time $k+1$ are:

$$\widehat{\bar{\mathbf{v}}}^{k+1} = \begin{cases} \bar{\boldsymbol{\lambda}}^{k+1}, & \text{on } \varepsilon^\circ \\ \mathsf{P}\mathbf{g}_D, & \text{on } \varepsilon_D^\partial \end{cases} \tag{37}$$

$$\begin{aligned} \widehat{\bar{\mathbf{Q}}}^{k+1} - \widehat{p}^k\mathbf{I} &= \bar{\mathbf{Q}}^{k+1} - p^k\mathbf{I} - \tau\left(\bar{\mathbf{v}}^{k+1} - \widehat{\bar{\mathbf{v}}}^{k+1}\right)\hat{\mathbf{n}}, && \text{on } \varepsilon^\circ \\ \widehat{\bar{\mathbf{Q}}}^{k+1} &= \mathsf{P}\mathbf{g}_N, && \text{on } \varepsilon_N^\partial \end{aligned} \tag{38}$$

where $\mathsf{P}\mathbf{g}_D/\mathsf{P}\mathbf{g}_N$ is the L^2 projection of $\mathbf{g}_D/\mathbf{g}_N$ into the space of $\boldsymbol{\theta}_\varepsilon/\boldsymbol{\Theta}_\varepsilon$ (e.g. tangential stresses), $\mathbf{I}$ is the $d\times d$ identity tensor, and $(\bar{\mathbf{v}}^{k+1} - \widehat{\bar{\mathbf{v}}}^{k+1})\hat{\mathbf{n}}$ is a $d\times d$ tensor. Substitution of these fluxes led to (23)–(24).

Similarly, the fluxes for $\widehat{\delta p}^{k+1}$ and $\widehat{\mathbf{q}_{\delta p}}^{k+1}$ that appeared in (28)–(29) are:

$$\widehat{\delta p}^{k+1} = \begin{cases} \lambda_{\delta p}^{k+1}, & \text{on } \varepsilon^\circ \\ \mathsf{P}g_{D_p}, & \text{on } \varepsilon_{D_p}^\partial \end{cases} \tag{39}$$

$$\begin{aligned} \widehat{\mathbf{q}_{\delta p}}^{k+1} &= \mathbf{q}_{\delta p}{}^{k+1} - \tau_p\left(\delta p^{k+1} - \widehat{\delta p}^{k+1}\right)\hat{\mathbf{n}}, && \text{on } \varepsilon^\circ \\ \widehat{\mathbf{q}_{\delta p}}^{k+1} &= \mathsf{P}g_{N_p}, && \text{on } \varepsilon_{N_p}^\partial \end{aligned} \tag{40}$$

where $\mathsf{P}g_{D_p}/\mathsf{P}g_{N_p}$ is the L^2 projection of g_{D_p}/g_{N_p} into the space of $\theta_\varepsilon/\boldsymbol{\theta}_\varepsilon$ (normally all zero Neumann, see eq. (10)). Substitutions of these fluxes lead to (28)–(29). Finally, the velocity fluxes $\widehat{\bar{\mathbf{v}}}_\star^{k+1}$ stem from continuity on the right-hand-side of (29) and thus ensue from the advective edge-space flux used in (24). A consistent HDG flux is thus to select the same as in (24),

$$\widehat{\bar{\mathbf{v}}}_\star^{k+1} = \bar{\boldsymbol{\lambda}}^{k+1} \tag{41}$$

and in our experience, this choice (41) gives accurate results. However, a central flux or, for improved stability, an upwind flux in the interior yields

$$\widehat{\bar{\mathbf{v}}}_\star^{k+1} = \left\{\!\!\left\{\bar{\mathbf{v}}^{k+1}\right\}\!\!\right\}, \tag{42}$$

$$\widehat{\bar{\mathbf{v}}}_\star^{k+1} = \left\{\!\!\left\{\bar{\mathbf{v}}^{k+1}\right\}\!\!\right\} + \frac{1}{2}\mathrm{sign}(\bar{\mathbf{v}}^{k+1}\cdot\hat{\mathbf{n}})\left[\!\!\left[\bar{\mathbf{v}}^{k+1}\hat{\mathbf{n}}\right]\!\!\right]\cdot\hat{\mathbf{n}}, \tag{43}$$

respectively. Each of these choices then replaces $\bar{\boldsymbol{\lambda}}^{k+1}$ in (29) and (35). Since they originate from (24), they then also need to be used in (24) and thus (37), for consistency.

3.2.3. HDG spatial discretization: un-split equations

We now state the element-local and globally-coupled HDG discretization for the un-split eqs. (4) (for a single time-stage). They are utilized in the proofs that follow. The element-local equations for the un-split system are:

$$\begin{aligned}
\left((\mathrm{Re})\mathbf{Q}^{k+1}, \boldsymbol{\Theta}\right)_K - \left(\nabla \mathbf{v}^{k+1}, \boldsymbol{\Theta}\right)_K + \left\langle \mathbf{v}^{k+1}, \hat{\mathbf{n}} \cdot \boldsymbol{\Theta}\right\rangle_{\partial K} &= \left\langle \boldsymbol{\lambda}^{k+1}, \hat{\mathbf{n}} \cdot \boldsymbol{\Theta}\right\rangle_{\partial K}, \\
\left(\frac{\mathbf{v}^{k+1}}{a\Delta t}, \boldsymbol{\theta}\right)_K - \left(\nabla \cdot \mathbf{Q}^{k+1}, \boldsymbol{\theta}\right)_K + \left\langle \tau \mathbf{v}^{k+1}, \boldsymbol{\theta}\right\rangle_{\partial K} + \left(\nabla p^{k+1}, \boldsymbol{\theta}\right)_K &= \left\langle \tau \boldsymbol{\lambda}^{k+1}, \boldsymbol{\theta}\right\rangle_{\partial K} + \left(\mathbf{F}^{k,k+1}, \boldsymbol{\theta}\right)_K, \\
\left(\nabla \cdot \mathbf{v}^{k+1}, \theta\right)_K - \left\langle \mathbf{v}^{k+1} \cdot \hat{\mathbf{n}}, \theta\right\rangle_{\partial K} &= -\left\langle \boldsymbol{\lambda}^{k+1} \cdot \hat{\mathbf{n}}, \theta\right\rangle_{\partial K}, \\
\left(p^{k+1}, \frac{1}{|K|}\right)_K &= |p|^{k+1}.
\end{aligned} \tag{44}$$

Note that these eqs. (44) are solvable once the velocity BCs and average pressure are specified. The globally coupled equations for $\boldsymbol{\lambda}$ and the average pressure $|p| = \frac{1}{|K|}\int_K p \mathrm{d}K$ on the element (with volume $|K| = \int_K \mathrm{d}K$) are:

$$\begin{aligned}
\left\langle \left[\!\left[-\mathbf{Q}^{k+1} \cdot \hat{\mathbf{n}} + p^{k+1}\hat{\mathbf{n}} + \tau\left(\mathbf{v}^{k+1} - \boldsymbol{\lambda}^{k+1}\right)\right]\!\right], \boldsymbol{\theta}_\varepsilon \right\rangle_\varepsilon &= \langle \mathbf{g}_N, \boldsymbol{\theta}_\varepsilon\rangle_\varepsilon, \\
\left\langle \boldsymbol{\lambda}^{k+1} \cdot \hat{\mathbf{n}}, \frac{1}{|\partial K|}\right\rangle_{\partial \mathcal{T}_h} &= 0, \\
\boldsymbol{\lambda}^{k+1}|_{\varepsilon_D^\partial} &= \mathbf{g}_D.
\end{aligned} \tag{45}$$

3.2.4. Final HDG velocity corrections: derivation

In this section we derive the element-local, (33), and edge-space, (35), final corrections of the predictor velocity. The goal of these final corrections is to obtain velocity fields $\mathbf{v}^{k+1}$ that are numerically divergence-free in the HDG sense, i.e. as in the third equation of the un-split system (44). For the split HDG system, it is the pressure-correction eqs. (28)–(29) that play that role. Hence, by taking the difference of these equations with their un-split counterpart, we can derive the final velocity corrections (33) and (35).

We begin by re-arranging (29), where the goal is to combine terms to make the final element-local and edge-space velocities appear as they are in the continuity equation of the un-split system (44). Multiplying (29) by the scalar $a\Delta t$ and assembling terms, we obtain,

$$\begin{aligned}
&-\left(\nabla \cdot (a\Delta t \mathbf{q}_{\delta p}^{k+1} - \bar{\mathbf{v}}^{k+1}), \theta\right)_K + \left\langle a\Delta t \tau_p (\delta p^{k+1} - \lambda_{\delta p}^{k+1}), \theta\right\rangle_{\partial K} + \left\langle (\bar{\boldsymbol{\lambda}}^{k+1} - \bar{\mathbf{v}}^{k+1}) \cdot \hat{\mathbf{n}}, \theta\right\rangle_{\partial K} = 0 \\
&\quad \Rightarrow -\left(\nabla \cdot (a\Delta t \mathbf{q}_{\delta p}^{k+1} - \bar{\mathbf{v}}^{k+1}), \theta\right)_K + \left\langle (\bar{\boldsymbol{\lambda}}^{k+1} - \bar{\mathbf{v}}^{k+1}) \cdot \hat{\mathbf{n}} + a\Delta t \tau_p (\delta p^{k+1} - \lambda_{\delta p}^{k+1}), \theta\right\rangle_{\partial K} = 0.
\end{aligned}$$

Subtracting this result from the target un-split HDG continuity equation, $\left(\nabla \cdot \mathbf{v}^{k+1}, \theta\right)_K + \left\langle (\boldsymbol{\lambda}^{k+1} - \mathbf{v}^{k+1}) \cdot \hat{\mathbf{n}}, \theta\right\rangle_{\partial K} = 0$ in (44), we obtain

$$\begin{aligned}
&\left(\nabla \cdot \left(\mathbf{v}^{k+1} - (\bar{\mathbf{v}}^{k+1} - a\Delta t \mathbf{q}_{\delta p}^{k+1})\right), \theta\right)_K \\
&+ \left\langle \left(\boldsymbol{\lambda}^{k+1} - \bar{\boldsymbol{\lambda}}^{k+1} - (\mathbf{v}^{k+1} - \bar{\mathbf{v}}^{k+1})\right) \cdot \hat{\mathbf{n}} + a\Delta t \tau_p (\lambda_{\delta p}^{k+1} - \delta p^{k+1}), \theta\right\rangle_{\partial K} = 0.
\end{aligned} \tag{46}$$

Defining $\mathbf{v}_{\mathrm{cor}}^{k+1} = \mathbf{v}^{k+1} - \bar{\mathbf{v}}^{k+1}$ and $\widehat{\mathbf{v}}_{\mathrm{cor}}^{k+1} = \boldsymbol{\lambda}_{\mathrm{cor}}^{k+1} = \boldsymbol{\lambda}^{k+1} - \bar{\boldsymbol{\lambda}}^{k+1}$, and using the flux definition (40) (as stated in (30)), we rewrite (46) as

$$\begin{aligned}
&\left(\nabla \cdot (\mathbf{v}_{\mathrm{cor}}^{k+1} + a\Delta t \mathbf{q}_{\delta p}^{k+1}), \theta\right)_K + \left\langle (\boldsymbol{\lambda}_{\mathrm{cor}}^{k+1} - \mathbf{v}_{\mathrm{cor}}^{k+1}) \cdot \hat{\mathbf{n}} + a\Delta t (\widehat{\mathbf{q}_{\delta p}}^{k+1} - \mathbf{q}_{\delta p}^{k+1}) \cdot \hat{\mathbf{n}}, \theta\right\rangle_{\partial K} = 0, \\
&\Rightarrow \left(\nabla \cdot (\mathbf{v}_{\mathrm{cor}}^{k+1} + a\Delta t \mathbf{q}_{\delta p}^{k+1}), \theta\right)_K + \left\langle (\boldsymbol{\lambda}_{\mathrm{cor}}^{k+1} + a\Delta t \widehat{\mathbf{q}_{\delta p}}^{k+1}) \cdot \hat{\mathbf{n}} - (\mathbf{v}_{\mathrm{cor}}^{k+1} + a\Delta t \mathbf{q}_{\delta p}^{k+1}) \cdot \hat{\mathbf{n}}, \theta\right\rangle_{\partial K} = 0.
\end{aligned} \tag{47}$$

This eq. (47) holds for all h and p discretizations and parameter τ_p. In theory, as long as $\mathbf{v}_{\mathrm{cor}}^{k+1}$ and $\boldsymbol{\lambda}_{\mathrm{cor}}^{k+1}$ satisfy (47), the final corrected velocities $\mathbf{v}^{k+1}$ and $\boldsymbol{\lambda}^{k+1}$ are divergence-free in the discrete HDG sense. However, a logical choice that leads to consistent corrections is to set both interior and edge terms to zero in (47) (and thus also in (46)). Hence, in (47), if we set $\mathbf{v}_{\mathrm{cor}}^{k+1} + a\Delta t \mathbf{q}_{\delta p}^{k+1} = 0$, both the element interior term and last edge-space term cancel, while if $\boldsymbol{\lambda}_{\mathrm{cor}}^{k+1} + a\Delta t \widehat{\mathbf{q}_{\delta p}}^{k+1} = 0$, the first edge-space term cancels and $\boldsymbol{\lambda}_{\mathrm{cor}}^{k+1}$ is then also consistent with $\mathbf{v}_{\mathrm{cor}}^{k+1}$. Summarizing, we derived the final consistent corrections and corresponding $\mathbf{v}^{k+1}$ and $\boldsymbol{\lambda}^{k+1}$, respectively,

$$\mathbf{v}_{\mathrm{cor}}^{k+1} = -a\Delta t \mathbf{q}_{\delta p}^{k+1} \qquad \text{and} \qquad \boldsymbol{\lambda}_{\mathrm{cor}}^{k+1} = -a\Delta t \widehat{\mathbf{q}_{\delta p}}^{k+1} \tag{48}$$

$$\Rightarrow \mathbf{v}^{k+1} = \bar{\mathbf{v}}^{k+1} - a\Delta t \mathbf{q}_{\delta p}^{k+1} \quad \text{and} \quad \boldsymbol{\lambda}^{k+1} = \bar{\boldsymbol{\lambda}}^{k+1} - a\Delta t \mathbf{q}_{\delta p}^{k+1} - a\Delta t \tau_p (\lambda_{\delta p}^{k+1} - \delta p^{k+1})\hat{\mathbf{n}}. \tag{49}$$

This completes the justification of (33) and (35). A few remarks:

1. As mentioned above, other acceptable choices for $\mathbf{v}_{\text{cor}}^{k+1}$ and $\boldsymbol{\lambda}_{\text{cor}}^{k+1}$ are possible, in part because the edge-terms in (46) and (47) only constrain the normal flux (the normal was in essence dropped in (48)–(49)). However, our corrections are logical and consistent with each other and with other relations, including the final HDG pressure correction (§3.2.5) and the τ_p relation (§3.2.6).
2. From (35) (and the above justification), the edge-space pressure-gradient correction leads to a continuous change of the edge-space velocity in the normal direction. Indeed, this change is equal to $\widehat{\mathbf{v}}_{\text{cor}}^{k+1} \cdot \hat{\mathbf{n}} = \boldsymbol{\lambda}_{\text{cor}}^{k+1} \cdot \hat{\mathbf{n}} = -a\Delta t \widehat{\mathbf{q}_{\delta p}}^{k+1} = -a\Delta t {\mathbf{q}_{\delta p}}^{k+1} \cdot \hat{\mathbf{n}} - a\Delta t \tau_p \left(\delta p^{k+1} - \lambda_{\delta p}^{k+1}\right)$, and is continuous because we solved (31). However, the corresponding change in the tangential direction, $\widehat{\mathbf{v}}_{\text{cor}}^{k+1} \cdot \hat{\mathbf{t}} = \boldsymbol{\lambda}_{\text{cor}}^{k+1} \cdot \hat{\mathbf{t}} = -a\Delta t \widehat{\mathbf{q}_{\delta p}}^{k+1} \cdot \hat{\mathbf{t}}$, is not constrained to be continuous. Hence, even though $\widehat{\bar{\mathbf{v}}}^{k+1} = \bar{\boldsymbol{\lambda}}^{k+1}$ is a unique vector on the edge by construction, the final corrected edge velocity $\boldsymbol{\lambda}^{k+1}$ can be discontinuous in the tangential direction. This property is common in projection methods, e.g. see [23].
3. Finally, combining (48) and (49), and using $\tau\, a\Delta t \tau_p = 1$ (to be justified in §3.2.6), we obtain the following relation between HDG stabilization functions for the final velocity correction and pressure-correction,

$$\begin{aligned} \left\langle \tau(\mathbf{v}_{\text{cor}}^{k+1} - \boldsymbol{\lambda}_{\text{cor}}^{k+1}), \boldsymbol{\theta} \right\rangle_{\partial K} &= \left\langle \tau\, a\Delta t \tau_p (\lambda_{\delta p}^{k+1} - \delta p^{k+1}) \hat{\mathbf{n}}, \boldsymbol{\theta} \right\rangle_{\partial K} \\ &= -\left\langle (\delta p^{k+1} - \lambda_{\delta p}^{k+1}) \hat{\mathbf{n}}, \boldsymbol{\theta} \right\rangle_{\partial K} . \end{aligned} \tag{50}$$

This relation will be used next in §3.2.5.

3.2.5. Final HDG pressure correction: derivation

We now derive the final HDG pressure correction (36). For a rotational correction [56,23], the goal is to obtain an equation for the final corrected pressure (gradient) such that the discrete un-split momentum equations are satisfied. Hence, just as in §3.2.4 we used the third equation of (44) to obtain velocities that are HDG divergence-free, here, we employ the second equation of (44) to correct the pressure gradient in (24) such that, at the end of a time-stage, momentum balance is satisfied in the HDG sense. Thus, taking the difference of this equation with the predictor (24), using (25), we obtain

$$\begin{aligned} &\left(\frac{\mathbf{v}_{\text{cor}}^{k+1}}{a\Delta t}, \boldsymbol{\theta}\right)_K - \left(\nabla \cdot \mathbf{Q}_{\text{cor}}^{k+1}, \boldsymbol{\theta}\right)_K + \left(\nabla p_{\text{cor}}^{k+1}, \boldsymbol{\theta}\right)_K + \left\langle -(\widehat{\mathbf{Q}}_{\text{cor}}^{k+1} - \mathbf{Q}_{\text{cor}}^{k+1}) \cdot \hat{\mathbf{n}} + (\widehat{p}_{\text{cor}}^{k+1} - p_{\text{cor}}^{k+1}) \hat{\mathbf{n}}, \boldsymbol{\theta} \right\rangle_{\partial K} = 0 \\ \Leftrightarrow &\left(\frac{\mathbf{v}_{\text{cor}}^{k+1}}{a\Delta t}, \boldsymbol{\theta}\right)_K - \left(\nabla \cdot \mathbf{Q}_{\text{cor}}^{k+1}, \boldsymbol{\theta}\right)_K + \left(\nabla p_{\text{cor}}^{k+1}, \boldsymbol{\theta}\right)_K + \left\langle \tau(\mathbf{v}_{\text{cor}}^{k+1} - \boldsymbol{\lambda}_{\text{cor}}^{k+1}), \boldsymbol{\theta} \right\rangle_{\partial K} = 0 , \end{aligned} \tag{51}$$

where we defined $\mathbf{Q}_{\text{cor}}^{k+1} = \mathbf{Q}^{k+1} - \bar{\mathbf{Q}}^{k+1}$, $\widehat{\mathbf{Q}}_{\text{cor}}^{k+1} = \widehat{\mathbf{Q}}^{k+1} - \widehat{\bar{\mathbf{Q}}}^{k+1}$ and $p_{\text{cor}}^{k+1} = p^{k+1} - p^k$, and where $\mathbf{v}_{\text{cor}}^{k+1}$ and $\boldsymbol{\lambda}_{\text{cor}}^{k+1}$ are as in §3.2.4. Next, we insert in (51) the velocity correction (33) and utilize the HDG pressure gradient corrector (28) and relation (50) between final velocity and pressure corrections, to get

$$\left(\nabla p^{k+1}, \boldsymbol{\theta}\right)_K = \left(\nabla p^k, \boldsymbol{\theta}\right)_K + \left(\nabla \delta p^{k+1}, \boldsymbol{\theta}\right)_K + \left(\nabla \cdot \mathbf{Q}_{\text{cor}}^{k+1}, \boldsymbol{\theta}\right)_K . \tag{52}$$

To obtain (36), we need to express the diffusion term in (52) in terms of a gradient. To do so, we can first start back from (51) and consider the strong form $\left(\nabla \cdot \mathbf{Q}_{\text{cor}}^{k+1}, \boldsymbol{\theta}\right)_K + \langle (\widehat{\mathbf{Q}}_{\text{cor}}^{k+1} - \mathbf{Q}_{\text{cor}}^{k+1}) \cdot \hat{\mathbf{n}}, \boldsymbol{\theta} \rangle_{\partial K}$. We integrate this sum back to the weak form, utilize (33) and then return to the strong form by integration by parts. We then complete the same manipulations as those made to go from (51) to (52), which yields a re-written (52),

$$\left(\nabla p^{k+1}, \boldsymbol{\theta}\right)_K = \left(\nabla p^k, \boldsymbol{\theta}\right)_K + \left(\nabla \delta p^{k+1}, \boldsymbol{\theta}\right)_K - \frac{a\Delta t}{\text{Re}} \left(\nabla \cdot \nabla \mathbf{q}_{\delta p}^{k+1}, \boldsymbol{\theta}\right)_K .$$

Realizing that $\mathbf{q}_{\delta p}^{k+1}$ is a gradient, in the element-local space, numerical operators can be defined such that $\left(\nabla \cdot \nabla \mathbf{q}_{\delta p}^{k+1}, \boldsymbol{\theta}\right)_K = \left(\nabla \nabla \cdot \mathbf{q}_{\delta p}^{k+1}, \boldsymbol{\theta}\right)_K$ i.e. the numerical curl of $\mathbf{q}_{\delta p}^{k+1}$ is null. With this, we obtain

$$\left(\nabla p^{k+1}, \boldsymbol{\theta}\right)_K = \left(\nabla p^k, \boldsymbol{\theta}\right)_K + \left(\nabla \delta p^{k+1}, \boldsymbol{\theta}\right)_K - \frac{a\Delta t}{\text{Re}} \left(\nabla \nabla \cdot \mathbf{q}_{\delta p}^{k+1}, \boldsymbol{\theta}\right)_K . \tag{53}$$

This relation (53) holds for all element-local numerical gradient operator. Thus, up to a constant,

$$\left(p^{k+1}, \theta\right)_K = \left(p^k + \delta p^{k+1}, \theta\right)_K - \frac{a\Delta t}{\text{Re}} \left(\nabla \cdot \mathbf{q}_{\delta p}^{k+1}, \theta\right)_K . \tag{54}$$

Equation (54) can be utilized for computations. A first alternate form is obtained by inserting the velocity correction (33) into (54) to obtain:

$$\left(p^{k+1},\theta\right)_K = \left(p^k+\delta p^{k+1},\theta\right)_K + \frac{1}{\mathrm{Re}}\left(\nabla\cdot\mathbf{v}^{k+1} - \nabla\cdot\bar{\mathbf{v}}^{k+1},\theta\right)_K , \tag{55}$$

which is the HDG element-local version of the classic (22). A second alternate form is obtained by utilizing in (54) the third equation of (44) which, as derived in §3.2.4, is also satisfied by the split HDG scheme. This yields

$$\left(p^{k+1},\theta\right)_K = \left(p^k+\delta p^{k+1},\theta\right)_K - \frac{1}{\mathrm{Re}}\left(\nabla\cdot\bar{\mathbf{v}}^{k+1},\theta\right)_K - \frac{1}{\mathrm{Re}}\left\langle(\boldsymbol{\lambda}^{k+1}-\mathbf{v}^{k+1})\cdot\hat{\mathbf{n}},\theta\right\rangle_{\partial K} . \tag{56}$$

To confirm this, one can also proceed using the results of §3.2.4, inserting (29) in (54) and using the edge-space corrections (33) and (35) to yield again (56). We note that this final (56) is as a standard DG on (12) but with the final updated HDG fluxes. Finally, the three forms (54), (55) and (56) can be derived from each other and are thus theoretically equivalent, but they lead to different implementations, efficiency and round-off errors.

3.2.6. Consistent HDG stability parameter for the pressure-correction: justification

We now justify the magnitudes of the HDG stability parameters τ and τ_p. For diffusive fluxes, the effect of varying the magnitude of τ has been studied [40,14–16,44] and $\tau = O(\nu/\ell)$ was found to yield accurate fluxes and stable solutions (for our non-dimensional equations, $\tau = O(1/\mathrm{Re})$). Hence, what remains is to justify the consistent expression for τ_p, $\tau_p = \frac{1}{a\Delta t\tau}$, used in §3.2.5. To do so, we compare the fluxes of the split equations to these of the un-split equations, through their respective edge-space variables. We first solve for the un-split $\boldsymbol{\lambda}$ in terms of its element-local quantities. To compare the un-split fluxes to the split ones, we then express the split edge-space variables $\bar{\boldsymbol{\lambda}}$ and $\lambda_{\delta p}$ in terms of their element-local quantities, and then form the final edge-space velocity using (35).

To solve for $\boldsymbol{\lambda}^{k+1}$ in the interior of the domain (note $\mathbf{g}_N = 0$ on the interior), we use the first equation in (45), then we expand the "jump" operator in terms of element-local quantities on either side of the edge, and finally we recombine terms using the "jump" and "mean" operators:

$$\begin{aligned}
0 &= \left\llbracket -\mathbf{Q}^{k+1}\cdot\hat{\mathbf{n}} + p^{k+1}\hat{\mathbf{n}} + \tau\left(\mathbf{v}^{k+1}-\boldsymbol{\lambda}^{k+1}\right)\right\rrbracket \\
0 &= -\mathbf{Q}^{+,k+1}\cdot\hat{\mathbf{n}}^+ + \mathbf{Q}^{-,k+1}\cdot\hat{\mathbf{n}}^+ + p^{+,k+1}\hat{\mathbf{n}}^+ - p^{-,k+1}\hat{\mathbf{n}}^+ + \tau(\mathbf{v}^{+,k+1}+\mathbf{v}^{-,k+1}-2\boldsymbol{\lambda}^{k+1}) \\
\Rightarrow \boldsymbol{\lambda}^{k+1} &= \left\{\!\!\left\{\mathbf{v}^{k+1}\right\}\!\!\right\} - \frac{1}{2\tau}\left\llbracket\mathbf{Q}^{k+1}\cdot\hat{\mathbf{n}}\right\rrbracket + \frac{1}{2\tau}\left\llbracket p^{k+1}\hat{\mathbf{n}}\right\rrbracket .
\end{aligned} \tag{57}$$

Now, we want to compare this to the $\boldsymbol{\lambda}^{k+1}$ obtained from the split equations. From the flux-conservation equation for $\bar{\boldsymbol{\lambda}}^{k+1}$, (26) (that is, using our flux definition (38), which includes the explicit pressure contribution), we proceed similarly

$$\begin{aligned}
0 &= \left\llbracket -\bar{\mathbf{Q}}^{k+1}\cdot\hat{\mathbf{n}} + p^{k}\hat{\mathbf{n}} + \tau\left(\bar{\mathbf{v}}^{k+1}-\bar{\boldsymbol{\lambda}}^{k+1}\right)\right\rrbracket \\
0 &= -\bar{\mathbf{Q}}^{+,k+1}\cdot\hat{\mathbf{n}}^+ + \bar{\mathbf{Q}}^{-,k+1}\cdot\hat{\mathbf{n}}^+ + p^{+,k}\hat{\mathbf{n}}^+ - p^{-,k}\hat{\mathbf{n}}^+ + \tau(\bar{\mathbf{v}}^{+,k+1}+\bar{\mathbf{v}}^{-,k+1}-2\bar{\boldsymbol{\lambda}}^{k+1}) \\
\Rightarrow \bar{\boldsymbol{\lambda}}^{k+1} &= \left\{\!\!\left\{\bar{\mathbf{v}}^{k+1}\right\}\!\!\right\} - \frac{1}{2\tau}\left\llbracket\bar{\mathbf{Q}}^{k+1}\cdot\hat{\mathbf{n}}\right\rrbracket + \frac{1}{2\tau}\left\llbracket p^{k}\hat{\mathbf{n}}\right\rrbracket .
\end{aligned} \tag{58}$$

Note that the time-level of the jump in the pressure is different between the split flux (58) and the un-split flux (57). To correct the velocity, we need to know the form of the flux $\widehat{\mathbf{q}_{\delta p}}^{k+1}$ (40). Starting from (31) we can find the form for $\lambda_{\delta p}^{k+1}$ (similar to before)

$$\begin{aligned}
0 &= \left\llbracket -\mathbf{q}_{\delta p}{}^{k+1}\cdot\hat{\mathbf{n}} + \tau_p\left(\delta p^{k+1}-\lambda_{\delta p}{}^{k+1}\right)\right\rrbracket \\
0 &= -\mathbf{q}_{\delta p}^{+,k+1}\cdot\hat{\mathbf{n}}^+ + \mathbf{q}_{\delta p}^{-,k+1}\cdot\hat{\mathbf{n}}^+ + \tau_p(\delta p^{+,k+1}+\delta p^{-,k+1}-2\lambda_{\delta p}^{k+1}) \\
\Rightarrow \lambda_{\delta p}^{k+1} &= \left\{\!\!\left\{\delta p^{k+1}\right\}\!\!\right\} - \frac{1}{2\tau_p}\left\llbracket\mathbf{q}_{\delta p}^{k+1}\cdot\hat{\mathbf{n}}\right\rrbracket .
\end{aligned} \tag{59}$$

Now, substituting (59) into (40) we find:

$$\begin{aligned}
\widehat{\mathbf{q}_{\delta p}}^{k+1}\cdot\hat{\mathbf{n}}^+ &= \mathbf{q}_{\delta p}{}^{+,k+1}\cdot\hat{\mathbf{n}}^+ - \tau_p\left(\delta p^{+,k+1}-\lambda_{\delta p}^{k+1}\right) \\
&= \mathbf{q}_{\delta p}{}^{+,k+1}\cdot\hat{\mathbf{n}}^+ - \tau_p\left(\delta p^{+,k+1} + \left\{\!\!\left\{\delta p^{k+1}\right\}\!\!\right\} - \frac{1}{2\tau}\left\llbracket\mathbf{q}_{\delta p}^{k+1}\cdot\hat{\mathbf{n}}\right\rrbracket\right) \\
&= \mathbf{q}_{\delta p}{}^{+,k+1}\cdot\hat{\mathbf{n}}^+ - \tau_p\left(\delta p^{+,k+1} - \frac{\delta p^{+,k+1}+\delta p^{-,k+1}}{2} - \frac{1}{2\tau_p}(\mathbf{q}_{\delta p}^{+,k+1}-\mathbf{q}_{\delta p}^{-,k+1})\cdot\hat{\mathbf{n}}^+\right) \\
&= \left\{\!\!\left\{\mathbf{q}_{\delta p}{}^{k+1}\right\}\!\!\right\}\cdot\hat{\mathbf{n}}^+ + \frac{\tau_p}{2}\left\llbracket\delta p^{k+1}\hat{\mathbf{n}}\right\rrbracket\cdot\hat{\mathbf{n}}^+
\end{aligned} \tag{60}$$

Finally, we construct the final edge-velocity for the split equations by substituting for (58) and (60) into (35):

$$\begin{aligned}\boldsymbol{\lambda}^{k+1} &= \bar{\boldsymbol{\lambda}}^{k+1} - a\Delta t\widehat{\mathbf{q}_{\delta p}}^{k+1}\\ &= \left\{\!\!\left\{\bar{\mathbf{v}}^{k+1}\right\}\!\!\right\} - \frac{1}{2\tau}\left[\!\!\left[\bar{\mathbf{Q}}^{k+1}\cdot\hat{\mathbf{n}}\right]\!\!\right] + \frac{1}{2\tau}\left[\!\!\left[p^{k}\hat{\mathbf{n}}\right]\!\!\right] - a\Delta t\left\{\!\!\left\{\mathbf{q}_{\delta p}^{k+1}\right\}\!\!\right\} + a\Delta t\frac{\tau_p}{2}\left[\!\!\left[\delta p^{k+1}\hat{\mathbf{n}}\right]\!\!\right],\\ &= \left\{\!\!\left\{\bar{\mathbf{v}}^{k+1} - a\Delta t\mathbf{q}_{\delta p}^{k+1}\right\}\!\!\right\} - \frac{1}{2\tau}\left[\!\!\left[\bar{\mathbf{Q}}^{k+1}\cdot\hat{\mathbf{n}}\right]\!\!\right] + \frac{1}{2\tau}\left[\!\!\left[p^{k}\hat{\mathbf{n}}\right]\!\!\right] + a\Delta t\frac{\tau_p}{2}\left[\!\!\left[\delta p^{k+1}\hat{\mathbf{n}}\right]\!\!\right],\\ &= \left\{\!\!\left\{\mathbf{v}^{k+1}\right\}\!\!\right\} - \frac{1}{2\tau}\left[\!\!\left[\bar{\mathbf{Q}}^{k+1}\cdot\hat{\mathbf{n}}\right]\!\!\right] + \frac{1}{2\tau}\left[\!\!\left[p^{k}\hat{\mathbf{n}}\right]\!\!\right] + a\Delta t\frac{\tau_p}{2}\left[\!\!\left[\delta p^{k+1}\hat{\mathbf{n}}\right]\!\!\right].\end{aligned} \tag{61}$$

We can now equate these split edge-space variables (61) with the un-split ones (57). The result links the split diffusive flux and pressures to the un-split ones:

$$-\frac{1}{2\tau}\left[\!\!\left[\bar{\mathbf{Q}}^{k+1}\cdot\hat{\mathbf{n}}\right]\!\!\right] + \frac{1}{2\tau}\left[\!\!\left[p^{k}\hat{\mathbf{n}}\right]\!\!\right] + \frac{a\Delta t}{2}\tau_p\left[\!\!\left[\delta p^{k+1}\hat{\mathbf{n}}\right]\!\!\right] = -\frac{1}{2\tau}\left[\!\!\left[\mathbf{Q}^{k+1}\cdot\hat{\mathbf{n}}\right]\!\!\right] + \frac{1}{2\tau}\left[\!\!\left[p^{k+1}\hat{\mathbf{n}}\right]\!\!\right].$$

If we multiply both sides by 2τ and re-arrange, we have

$$\left[\!\!\left[-\bar{\mathbf{Q}}^{k+1}\cdot\hat{\mathbf{n}} + \left(p^{k} + (a\Delta t\tau)\tau_p\,\delta p^{k+1}\right)\hat{\mathbf{n}}\right]\!\!\right] = \left[\!\!\left[-\mathbf{Q}^{k+1}\cdot\hat{\mathbf{n}} + p^{k+1}\hat{\mathbf{n}}\right]\!\!\right]. \tag{62}$$

This is the split vs. un-split "consistency of jumps" equation, with the stabilization term (edge-space variables) eliminated. This equation (62) holds for all time-discretization $a\Delta t$, space-discretization h and order p. Hence, for consistency, $\tau_p = \frac{1}{a\Delta t\tau}$, which is the sought-after expression. A few remarks:

1. We can also obtain $(a\Delta t\tau)\tau_p = 1$ by decomposing the un-split pressure p^{k+1} into $p^{k} + \delta p^{k,k+1}$ where $\delta p^{k,k+1}$ is the total un-split pressure correction. This yields the following relation between split and un-split pressure-correction jumps, $(a\Delta t\tau)\tau_p\left[\!\!\left[\delta p^{k+1}\hat{\mathbf{n}}\right]\!\!\right] = \left[\!\!\left[\delta p^{k,k+1}\hat{\mathbf{n}}\right]\!\!\right]$. Hence, we retrieve: $(a\Delta t\tau)\tau_p = 1$.
2. Without the rotational correction, eq. (62) equates the new split pressure $p^{k} + (a\Delta t\tau)\tau_p\,\delta p^{k+1} = p^{k} + \delta p^{k+1}$ to the un-split p^{k+1} and $\bar{\mathbf{Q}}^{k+1}\cdot\hat{\mathbf{n}}$ approximates $\mathbf{Q}^{k+1}\cdot\hat{\mathbf{n}}$.
3. With the rotational correction, e.g. using the form (54), the left-hand-side of (62) yields $p^{k} + (a\Delta t\tau)\tau_p\,\delta p^{k+1} = p^{k} + \delta p^{k+1} = p^{k+1} + \frac{a\Delta t}{\mathrm{Re}}\nabla\cdot\mathbf{q}_{\delta p}^{k+1}$. Equating the new split pressure p^{k+1} with its un-split version, eq. (62) then shows that the rotational correction updates $\bar{\mathbf{Q}}^{k+1}$.
4. The split and un-split fluxes are thus close, but can differ because the diffusive fluxes are not exactly the same; the split equation's diffusive flux still contains a contribution due to the non-divergence of $\bar{\mathbf{v}}^{k+1}$. This contribution is partly removed by the rotational correction, but there are additional terms present in the normal vector $(\mathbf{Q}-\bar{\mathbf{Q}})\cdot\hat{\mathbf{n}}$. As with any projection method scheme, this splitting error is expected to be small, particularly for large Reynolds numbers (as we will show in §7.2). Additional considerations are provided in [58].
5. Finally, the explicit pressure flux present in (38) leads to the constant expression for τ_p. However, that expression for τ_p is also consistent in other cases. For example, if in (24), the edge-continuous $\left(\mathbf{q}_p^{k},\boldsymbol{\theta}\right)_K$ had been utilized instead of $\left(\nabla p^{k},\boldsymbol{\theta}\right)_K$, the pressure difference would not be part of (25) nor in (38) and the pressure jump in (26), (58) and (61) would be replaced by zero. In that case, if we decompose the un-split pressure p^{k+1} into the edge-continuous pressure field at time k and its departure from it, we still obtain $(a\Delta t\tau)\tau_p = 1$ (for details, see [35]).

3.3. Discussion on the effect of varying τ_p

As we have shown in §3.2, to ensure overall consistency and global flux-conservation in the HDG sense, it is important to treat the time-split equations as a single system. Failure to do so can lead to relatively poor numerical results, even when each discrete equation is on its own consistent and stable.

For example, using a value such as $\tau_p = O(1)$ could be thought as adequate if the second step (9) in the split scheme was treated in isolation of the first (that is, not treating the time-split equations as a single system). However, we now show that the consistent value for τ_p derived in §3.2.6 improves solutions, using simple numerical experiments. For the first experiment we use a first order upwind time discretization scheme to calculate one time-step of a lock-exchange flow (described in §7). When $\tau_p = 1$ is used, we see that the solution is discontinuous across the elements bordering the top/bottom boundary when the solution is not well-resolved (Fig. 5(top)). The consistent $\tau_p = \frac{1}{\Delta t\tau}$ does not have this issue (Fig. 5(middle)). When resolution is increased (Fig. 5(bottom)), we find that both choices of τ_p give approximately the same solution. Finally, as $\tau_p \rightarrow \infty$, the solution approaches a continuous Galerkin discretization [14,32].

Overall, this situation arises because the dominant velocity is aligned with the grid. The tangential portion of the velocity-correction is not directly penalized by the numerical scheme. That is, $\mathbf{q}_{\delta p}\cdot\hat{\mathbf{t}}$ is free to vary. Increasing τ_p, then, removes the discontinuity in the pressure field, which indirectly also removes the discontinuity in $\mathbf{q}_{\delta p}\cdot\hat{\mathbf{t}}$ across edges. If we used a mesh of triangles instead of squares (see Fig. 6), the discontinuity is not nearly as severe, since in this case the u-velocity component is not tangent to the diagonal edge of the triangles, and does get penalized.

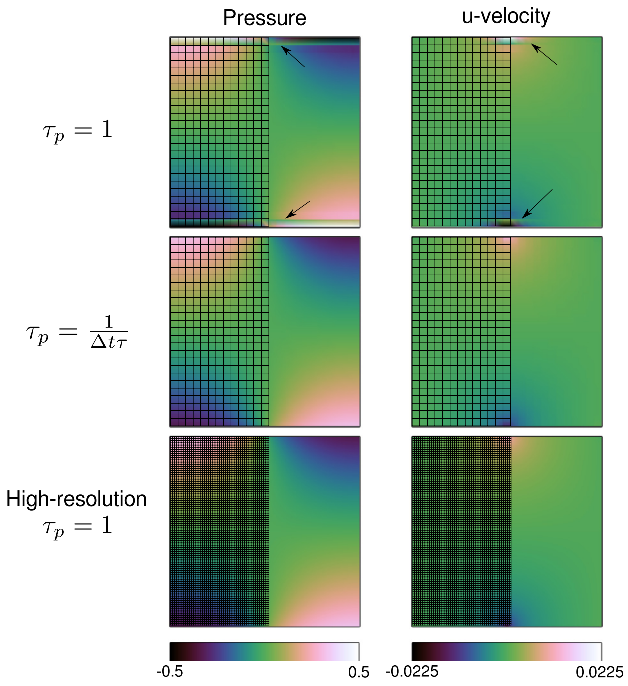

Fig. 5. Pressure and u-velocity after 1 time-step $\Delta t = 0.01$ of the lock-exchange problem (see §7) using a mesh of square elements at low-resolution (p = 1, $\Delta x = 0.04$) with $\tau_p = 1$ (top), $\tau_p = \frac{1}{\Delta t \tau}$ (middle), and at high-resolution (p = 5, $\Delta x = 0.01$). The $\tau_p = \frac{1}{\Delta t \tau}$ case (middle) has a lower u-velocity magnitude and the solution is smooth, while the low resolution $\tau_p = 1$ case (top) does capture the maximum velocity, but the solution has large discontinuities at the locations indicated with arrows. When the solution is further resolved, the $\tau_p = 1$ and $\tau_p = \frac{1}{\Delta t \tau}$ (not shown) cases give essentially the same solution.

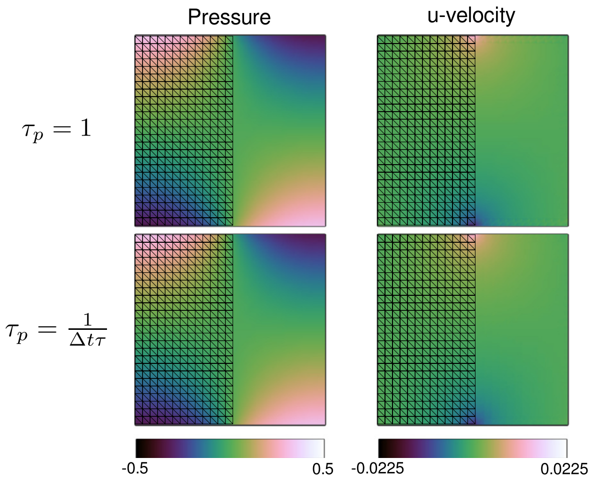

Fig. 6. As in Fig. 5 but with triangular elements. The triangular mesh does not have a large discontinuity for $\tau_p = 1$ because the tangential correction velocity is penalized.

We now use a second numerical experiment to highlight additional repercussions of using an inconsistent τ_p, particularly at low-resolution. We show in Fig. 7 the result of another lock-exchange simulation using our scheme from §3.2 with the consistent $\tau_p = \frac{1}{a\Delta t \tau}$ (see §3.2.6). If we employ another value of τ_p, the proper vortices are not formed (not shown): e.g. with $\tau_p = 1$, Kelvin–Helmholtz instabilities don't develop, while with $\tau_p = \frac{1}{a\Delta t^2 \tau}$, spurious vortices form (see [58]). It is only when the spatial resolution is increased, e.g. to 400 × 100, that all three schemes give a similar answer (not shown).

We have shown that the magnitude of τ_p can significantly affect the numerical solution for HDG schemes when the flow is not well-resolved and the dominant flow is aligned with the element edges. We have derived the consistent magnitude for this parameter, and we have shown how the solution of the lock-exchange problem is affected by using different values. In general, the time-split equations need to be treated as a single system, and consistent penalty terms have to be used in each case.

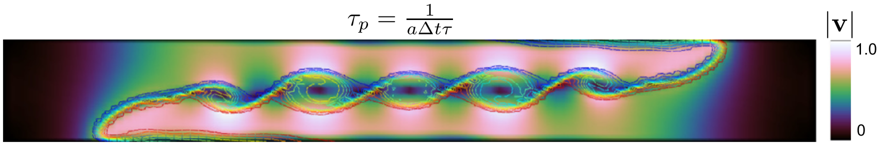

Fig. 7. Density contours over velocity magnitude at time 10 of the lock-exchange problem (see §7) using a time step of $\Delta t = 0.001$, a mesh of 100×25 linear elements and a first-order accurate incremental pressure-correction scheme derived in §3.2.

4. Time discretization using IMEX-RK schemes

Now that the equations are spatially discretized, various ordinary differential equation solvers can be used to advance the equations in time. Runge–Kutta (RK) methods are attractive because they are self-starting, and allow for variable time-step sizes. We thus employ existing Implicit–Explicit (IMEX) RK methods [3,31]. However, these existing methods need to be adapted, due to the nature of projection methods, which already introduce a time-discretization, and HDG methods, which are often implicit. Questions also arise for the treatment of the explicit and implicit pressure updates. As a result, we describe next the modifications required to solve our spatially-discretized equations using IMEX-RK methods. We note that since splitting errors limit to second-order accuracy in time [23], for higher-order integration, an iterative approach would be used, e.g. [22].

We are interested in IMEX-RK time-stepping schemes with s stages that are of the form (using Python/C/C++ indexing):

$$\phi^{k+1} = \phi^k + \Delta t \sum_{i=0}^{s-1} b_i^{im} f^{im}(\phi_i) + \Delta t \sum_{i=0}^{s-1} b_i^{ex} f^{ex}(\phi_i), \tag{63}$$

where the stage variables are solved using

$$\begin{aligned} \phi_i &= \phi^k + \Delta t \sum_{j=0}^{i} a_{i,j}^{im} f^{im}(\phi_j) + \Delta t \sum_{j=0}^{i-1} a_{i,j}^{ex} f^{ex}(\phi_j) \\ \Rightarrow (1 - \Delta t a_{i,i}^{im} f^{im}(\phi_i)/\phi_i)\,\phi_i &= \phi^k + \Delta t \sum_{j=0}^{i-1} a_{i,j}^{im} f^{im}(\phi_j) + \Delta t \sum_{j=0}^{i-1} a_{i,j}^{ex} f^{ex}(\phi_j), \end{aligned} \tag{64}$$

where ϕ is some field satisfying the equation $\frac{\partial \phi}{\partial t} = f^{im} + f^{ex}$ and $\phi_0 = \phi^k$. IMEX schemes treat one part of the right-hand-side implicitly (usually stiff terms such as diffusion) and the other part explicitly (64). As such, we require two Butcher Tableaus, one for the implicit terms, and one for the explicit terms. We only consider schemes with Butcher Tableaus of the form:

$$\begin{array}{c|ccccc} 0 & 0 & \dots & \dots & 0 \\ c_1 & a_{1,0}^{ex} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & 0 \\ c_{s-1} & a_{(s-1),0}^{ex} & \dots & a_{(s-1),(s-2)}^{ex} & 0 \\ \hline & b_0 & \dots & b_{s-2} & a \end{array} \qquad \begin{array}{c|cccccc} 0 & 0 & 0 & \dots & \dots & 0 \\ c_1 & a_{1,0}^{im} & a & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & & \dots & 0 \\ c_{s-1} & a_{(s-1),0}^{im} & \dots & \dots & a_{(s-1),(s-2)}^{im} & a \\ \hline & b_0 & \dots & \dots & b_{s-2} & a \end{array} \tag{65}$$

where these schemes have the following properties:

1. $b_i = b_i^{ex} = b_i^{im}$
2. $c_{s-1} = 1$ (usually)
3. $a_{i,j}^{ex} = 0 \quad \forall j \geq i$
4. $a_{i,j}^{im} = 0 \quad \forall j > i$
5. $a_{i,i}^{im} = a \quad \forall 0 < i < s$

Item 1 is often-used in IMEX-RK schemes (e.g. [3,31]). Item 2 provides a last-stage pressure at end time $k+1$ (useful but not required). Item 3 is a necessary condition for an explicit RK scheme. Finally, items 4 and 5 are important for an efficient implicit method, where zero entries above the diagonal allow different stages to be solved sequentially (as opposed to simultaneously), and a single diagonal entry allows for the creation of a single implicit matrix.

We now obtain the IMEX-RK procedure for the time-split projection method, where the time-splitting happens within every RK stage. For ease of notation, we focus on the time-split projection of §2; the IMEX-RK procedure for our time-split HDG projection of §3.2 is obtained at the end. To start, we write the momentum equations for the true velocity from (3) as

$$\frac{\partial \mathbf{v}}{\partial t} = \mathbf{F}^{im} + \mathbf{F}^{ex} - \nabla p, \tag{66}$$

where $\mathbf{F}^{im} = \nabla \cdot \frac{1}{\mathrm{Re}}\nabla\mathbf{v}$, $\mathbf{F}^{ex} = \mathbf{F}_{\partial t}$. Next, we first consider the un-split projection and write a typical IMEX-RK stage, using items 1–5 above:

$$\begin{aligned}\mathbf{v}_i - a^{im}_{i,i}\Delta t\mathbf{F}^{im}_i &= \left(1 - a^{im}_{i,i}\Delta t\nabla \cdot \frac{1}{\mathrm{Re}}\nabla\right)\mathbf{v}_i \\ &= \mathbf{v}^k + \Delta t\sum_{j=0}^{i-1} a^{im}_{i,j}\mathbf{F}^{im}_j + \Delta t\sum_{j=0}^{i-1} a^{ex}_{i,j}\mathbf{F}^{ex}_j \\ &\quad - \Delta t\sum_{j=0}^{i-1} a^{im}_{i,j}\nabla p_j - \Delta t a^{im}_{i,i}\nabla p_i,\end{aligned} \tag{67}$$

where $\mathbf{F}^{ex}_i$ and $\mathbf{F}^{im}_i$ contain the explicit and implicit terms calculated at previous stages, respectively, and pressure is treated implicitly.

Stage-i, split scheme. For the predictor equation, we estimate ∇p_i at each stage i in (67). For an incremental pressure-correction method, we predict ∇p_i using $\nabla p_{i\star}$ where $p_{i\star}$ is a function of the previously calculated pressures,

$$\nabla p_i \approx \nabla p_{i\star} = \nabla F(p_0, \ldots, p_j),\ j \le i-1.$$

A typical, time-split, IMEX-RK stage calculation for the predictor velocity $\bar{\mathbf{v}}_i$ can then be written as follows, where we have also divided by Δt

$$\begin{aligned}\frac{\bar{\mathbf{v}}_i}{\Delta t} - a^{im}_{i,i}\bar{\mathbf{F}}^{im}_i = \left(\frac{1}{\Delta t} - a^{im}_{i,i}\nabla \cdot \frac{1}{\mathrm{Re}}\nabla\right)\bar{\mathbf{v}}_i = \frac{\mathbf{v}^k}{\Delta t} + \sum_{j=0}^{i-1} a^{im}_{i,j}\mathbf{F}^{im}_j + \sum_{j=0}^{i-1} a^{ex}_{i,j}\mathbf{F}^{ex}_j \\ - \sum_{j=0}^{i-1} a^{im}_{i,j}\nabla p_j - a^{im}_{i,i}\nabla p_{i\star}.\end{aligned} \tag{68}$$

Proceeding similarly to §2, as in (9), we perform the stage i projection step

$$-\nabla^2\delta p_i = -\frac{\nabla \cdot \bar{\mathbf{v}}_i}{a^{im}_{i,i}\Delta t}, \tag{69}$$

and, as in (11), we correct the stage i velocity

$$\mathbf{v}_i = \bar{\mathbf{v}}_i - a^{im}_{i,i}\Delta t\nabla\delta p_i\,. \tag{70}$$

To derive the final pressure correction, we again proceed as in §2 and §3.2. We subtract the IMEX-RK split eq. (68) from the IMEX-RK un-split eq. (67) and substitute the velocity correction at stage i (70). In doing so, the differences between the split and un-split terms from previous stages $j = 0, i-1$ cancel, both for the $\mathbf{F}^{im}_j$ terms and ∇p_j terms (to be discussed later). Hence, since the explicit $\mathbf{F}^{ex}_j$ and $\mathbf{v}^k$ terms also cancel out, we obtain:

$$\left(\frac{1}{\Delta t} - a^{im}_{i,i}\nabla \cdot \frac{1}{\mathrm{Re}}\nabla\right)(-a^{im}_{i,i}\Delta t\nabla\delta p_i) = -a^{im}_{i,i}\nabla p_i + a^{im}_{i,i}\nabla p_{i\star}\,.$$

Now we solve for ∇p_i and then for p_i which we obtain after re-using (70):

$$\nabla p_i = \nabla p_{i\star} + \nabla\delta p_i - a^{im}_{i,i}\nabla \cdot \frac{1}{\mathrm{Re}}\nabla\Delta t\nabla\delta p_i \tag{71}$$

$$\Rightarrow \nabla p_i = \nabla\left\{p_{i\star} + \delta p_i - \nabla\frac{1}{\mathrm{Re}} \cdot a^{im}_{i,i}\Delta t\nabla\delta p_i\right\}$$

$$\Rightarrow p_i = p_{i\star} + \delta p_i - \frac{1}{\mathrm{Re}}\nabla \cdot \bar{\mathbf{v}}_i. \tag{72}$$

Note that this gives the pressure (up to a constant) at the time of stage i.

While this procedure allows us to calculate intermediate divergence-free stage variables, the flux terms are also needed for the next stage, i.e. ∇p_i and $\mathbf{F}^{im}_i$ are needed for stage $i+1$ in (68). First, the guessed values $\nabla p_{i\star}$ need to be replaced by the final corrected ∇p_i. This is assumed in (68) for previous stages $j = 0, i-1$ and was used in the above derivation of the final pressure correction (71): up to splitting errors, the difference between the un-split and split terms ∇p_j at $j = 0, i-1$ cancelled out since the final corrected pressure gradients at $j = 0, i-1$ were used to replace the $\nabla p_{j\star}$'s. Second, the corrected velocities $\mathbf{v}_i$ in (70) need to be used to replace the implicit predicted diffusion $\bar{\mathbf{F}}^{im}_i$ term by the final implicit corrected diffusion $\mathbf{F}^{im}_i$ term. Again, this is assumed in (68) for $j = 0, i-1$ and was used to derive (71): up to splitting errors, the differences between split and un-split $\mathbf{F}^{im}_j$ terms then cancelled out. While we can re-evaluate these implicit diffusion terms using $\mathbf{v}_i$, it is more efficient computationally to solve for $\mathbf{F}^{im}_i$ from (67) as follows

$$\begin{aligned}\mathbf{v}_i - a^{im}_{i,i}\Delta t\mathbf{F}^{im}_i &= \mathbf{v}^k + \Delta t\sum_{j=0}^{i-1} a^{im}_{i,j}\mathbf{F}^{im}_j + \Delta t\sum_{j=0}^{i-1} a^{ex}_{i,j}\mathbf{F}^{ex}_j - \Delta t\sum_{j=0}^{i-1} a^{im}_{i,j}\nabla p_j - \Delta t a^{im}_{i,i}\nabla p_i \\ \Rightarrow \mathbf{F}^{im}_i &= \frac{\mathbf{v}_i - \mathbf{v}^k}{a^{im}_{i,i}\Delta t} - \frac{1}{a^{im}_{i,i}}\left\{\sum_{j=0}^{i-1} a^{im}_{i,j}\mathbf{F}^{im}_j + \sum_{j=0}^{i-1} a^{ex}_{i,j}\mathbf{F}^{ex}_j - \sum_{j=0}^{i} a^{im}_{i,j}\nabla p_j\right\}. \end{aligned} \tag{73}$$

From (69) and (70), the stage-i velocity in (73) is divergence-free: $\nabla\cdot\mathbf{v}_i = 0$. Hence, $\nabla\cdot\left\{\sum_{j=0}^{i} a^{im}_{i,j}\mathbf{F}^{im}_j + \sum_{j=0}^{i-1} a^{ex}_{i,j}\mathbf{F}^{ex}_j - \sum_{j=0}^{i} a^{im}_{i,j}\nabla p_j\right\} = 0$. However, the terms in this sum are in general divergent. At each stage, the pressure p_i thus balances the divergence from the previous stages. This has implications for the final recombination stage. (Note also that if $c_{s-1} = 1$, p_{s-1} is at final time $k+1$, see item 2 above.)

Final recombination, split scheme. The final recombination (63) of the IMEX-RK scheme sums implicit and explicit function evaluations. For our Navier–Stokes equations (66), this yields

$$\frac{\bar{\mathbf{v}}^{k+1}}{\Delta t} = \frac{\mathbf{v}^k}{\Delta t} + \sum_{i=0}^{s-1} b^{im}_i\mathbf{F}^{im}_i + \sum_{i=0}^{s-2} b^{ex}_i\mathbf{F}^{ex}_i + b^{ex}_{s-1}\mathbf{F}^{ex}_{s-1} - \sum_{i=0}^{s-1} b^{im}_i\nabla p_i\,, \tag{74}$$

where we have all the implicit terms $\mathbf{F}^{im}_i\,\forall i\in[0,\,s-1]$, but we still need $\mathbf{F}^{ex}_{s-1} = \mathbf{F}_{ex}(\mathbf{v}_{s-1})$, since $\mathbf{v}_{s-1}$ is newly calculated after solving for all final stage values. This means we have to evaluate the non-linear advection terms $\mathbf{F}^{ex}_{s-1}$, which can be divergent. Critically, the recombination coefficients (b^{im}_i, b^{ex}_i) are not the same as the stage coefficients $(a^{im}_{i,j}, a^{ex}_{i,j})$ and the RHS of (74) is thus divergent. As such, we need to calculate an update to the pressure p_{s-1}, which will balance these divergent terms, leading to an updated divergence-free velocity $\mathbf{v}^{k+1}$. Hence, we solve for the last correction,

$$-\nabla^2\delta p_{k+1} = -\frac{\nabla\cdot\bar{\mathbf{v}}^{k+1}}{b^{im}_{s-1}\Delta t}, \tag{75}$$

leading to the final corrected $\mathbf{v}^{k+1}$ and final corrected pressure p^{k+1},

$$\mathbf{v}^{k+1} = \bar{\mathbf{v}}^{k+1} - b^{im}_{s-1}\Delta t\nabla\delta p_{k+1}\,, \tag{76}$$

$$\nabla p^{k+1} = \nabla p_{s-1} + \nabla\delta p_{k+1} \tag{77}$$

$$\Rightarrow p^{k+1} = p_{s-1} + \delta p_{k+1}. \tag{78}$$

In (77)–(78), the rotational correction is null since the only newly computed term in (74) is $\bar{\mathbf{v}}^{k+1}$ itself [35].

For the HDG recombination step, we have to define $\widehat{\bar{\mathbf{v}}}^{k+1}_\star$ (see §3.2): for the consistent HDG flux $\widehat{\bar{\mathbf{v}}}^{k+1}_\star = \bar{\boldsymbol{\lambda}}^{k+1}$ (41), we have to take additional care. Since we are not solving for the diffusive terms at the final recombination, we need to obtain $\bar{\boldsymbol{\lambda}}^{k+1}$ by another means. The consistent value can be found by considering an HDG discretization of (24) where the diffusive terms are treated explicitly. In that case we only have to define a flux for $-\hat{p}^k\mathbf{I}$ (where the analogous flux (38) contains the diffusive term $\widehat{\mathbf{Q}}^{k+1}$), which gives:

$$-\hat{p}^k\mathbf{I} = -p^k\mathbf{I} - \tau(\bar{\mathbf{v}}^{k+1} - \bar{\boldsymbol{\lambda}}^{k+1})\hat{\mathbf{n}}. \tag{79}$$

Solving for $\bar{\boldsymbol{\lambda}}^{k+1}$ that sets $\llbracket -\hat{p}^k\mathbf{I}\rrbracket = 0$ in the interior, we obtain

$$\bar{\boldsymbol{\lambda}}^{k+1} = \left\{\!\!\left\{\bar{\mathbf{v}}^{k+1}\right\}\!\!\right\} + \frac{1}{2\tau}\left\llbracket p^k\hat{\mathbf{n}}\right\rrbracket\,. \tag{80}$$

The main difference between the stage calculations and the final recombination is that we do not solve for implicit diffusive terms, which requires us to calculate for the intermediate $\bar{\boldsymbol{\lambda}}^{k+1}$. After the velocity predictor step, the final recombination and stage calculations are nearly the same, except that the rotational correction is not applied to the pressure at the final recombination.

In summary, to use IMEX-RK schemes with the HDG projection method discretization, two new modifications are required. First, the projection step has to be carried out at every stage and at the final recombination. Second, at the final recombination, the HDG flux for the predictor velocity needs to be evaluated, and the rotational correction needs to be omitted.

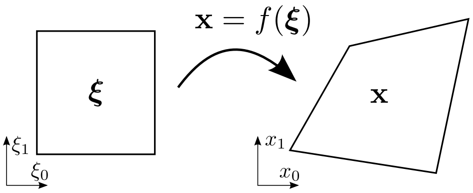

Fig. 8. Coordinate transformation from the reference coordinate system to the physical coordinate system.

5. Quadrature free integration and slope limiting

In this section, we first introduce a new quadrature-free numerical integration scheme which is consistent with the HDG method. Then, we derive our new selective nodal slope limiter which aims to suppress numerical oscillations locally, and so improve the stability of the method. Considerations on nodal and modal HDG implementations are in [58].

5.1. Quadrature free integration consistent with HDG schemes

In §3.2 we formulated the scheme for the strong form of the equations. As we will explain, a reason for this is tied to our quadrature-free implementation of the HDG method on general curved meshes, where the coordinate transformation factors are not constant on the element (but treated as polynomials). This extends the work by Hesthaven and Warburton [27] to HDG implementations where the curved meshes may not be continuous. Next, we first introduce some new notation, then describe our quadrature-free scheme, and finally explain the issues with other choices.

We use $\boldsymbol{\xi}$ for the coordinates in the reference finite element (Fig. 8). The coordinate transformation can then be described through a vector function $\mathbf{x} = f(\boldsymbol{\xi})$, which can be discretized using a truncated polynomial expansion, $\mathbf{x}(\boldsymbol{\xi}) \approx \sum_i \mathbf{x}_i \boldsymbol{\theta}_i(\boldsymbol{\xi})$. In this case, a nodal basis becomes particularly useful since the coefficients of this polynomial will be equal to the real-space coordinate at the nodal points of the reference element, or $\mathbf{x}_i = \mathbf{x}(\boldsymbol{\xi}_i)$. With this polynomial representation of the coordinate transformation, we can perform all needed numerical derivatives, integrals and other operations after having computed the: entries of the $\frac{\partial \boldsymbol{\xi}}{\partial \mathbf{x}}$ matrix for every element; determinant of the $\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}}$ matrix for all elements and edges; and normal vector $\hat{\mathbf{n}}$ for every edge. We shall refer to individual entries in the first matrix as the "Jacobian factors" and to the determinant of the second matrix as the "Jacobian."

It is simple to calculate derivatives $\frac{\partial}{\partial \boldsymbol{\xi}}$ on the master element, but we are interested in calculating derivatives in the physical space $\frac{\partial}{\partial \mathbf{x}}$. Numerically, we always calculate derivatives on the reference element, and then use the chain rule to obtain the desired derivative $\frac{\partial}{\partial \mathbf{x}} = \frac{\partial}{\partial \boldsymbol{\xi}} \frac{\partial \boldsymbol{\xi}}{\partial \mathbf{x}}$. To calculate the Jacobian factors, then, we use the identity:

$$\frac{\partial x_i}{\partial \xi_j} \frac{\partial \xi_j}{\partial x_k} = \delta_{ik} \tag{81}$$

$$\frac{\partial \xi_j}{\partial x_k} = \left[\frac{\partial x_i}{\partial \xi_j} \right]^{-1} \delta_{ik} \tag{82}$$

When using a quadrature-free scheme, it is important to maintain this property (81) discretely. As for the derivations in §3.2, the gradient of a scalar function ϕ (giving a vector function) could be taken discretely either in the strong or weak form using vector $\boldsymbol{\theta}$'s as:

$$(\nabla \phi)_{\text{strong}} \approx \left[\left(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j \right)_K \right]^{-1} \left\{ \left(\nabla \phi, \boldsymbol{\theta}_j \right)_K + \left\langle \hat{\phi} - \phi, \hat{\mathbf{n}} \cdot \boldsymbol{\theta}_j \right\rangle_{\partial K} \right\} \tag{83}$$

$$(\nabla \phi)_{\text{weak}} \approx \left[\left(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j \right)_K \right]^{-1} \left\{ - \left(\phi, \nabla \cdot \boldsymbol{\theta}_j \right)_K + \left\langle \hat{\phi}, \hat{\mathbf{n}} \cdot \boldsymbol{\theta}_j \right\rangle_{\partial K} \right\} \tag{84}$$

To discretize (81) using (83) or (84), let us define the discrete matrices

$$\begin{aligned}
\mathbf{M}_e &= \left(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j \right)_{\partial K^{\text{ref}}}, & \mathbf{M}_\varepsilon &= \left(\boldsymbol{\theta}_{\varepsilon,i}, \boldsymbol{\theta}_{\varepsilon,j} \right)_{\varepsilon^{\text{ref}}}, \\
\mathbf{S} &= \left(\boldsymbol{\theta}_i, \nabla \boldsymbol{\theta}_j \right)_{K^{\text{ref}}}, & \mathbf{M} &= \left(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j \right)_{K^{\text{ref}}}, \\
\mathbf{D} &= \mathbf{M}^{-1} \mathbf{S}, & \mathbf{L} &= \mathbf{M}^{-1} \mathbf{M}_e, \\
\mathbf{J} &= \det \left[\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} \right]_K, & \mathbf{J}_e &= \det \left[\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} \right]_{\partial K},
\end{aligned}$$

where K^{ref} is the reference element. Note that $\mathbf{M}_\varepsilon$ is a matrix of size $N_{b,e} \times N_{b,e}$ and $\mathbf{M}_e$ is a matrix of size $N_b \times (\sum_{i=0}^{i<N_e} N_{b,e,i})$, where N_b is the number of bases on the element, $N_{b,e,i}$ is the number of bases on edge i of the element, and N_e is the number of edges in an element. Also, $\mathbf{J}$ and $\mathbf{J}_e$ are diagonal matrices, where each element of the diagonal is the Jacobian at that particular nodal point.

We can now write the discretization of (81) using the strong form (83) as

$$\begin{aligned}\left(\frac{\partial x_i}{\partial \xi_j}\frac{\partial \xi_j}{\partial x_k}\right)_{\text{strong}} &= \left[\mathbf{D}_j\mathbf{x}_i\right]\frac{\partial \xi_j}{\partial x_k} + \mathbf{J}^{-1}\mathbf{L}(\hat{\mathbf{x}}_i - \mathbf{x}_i)\mathbf{J}_e\hat{\mathbf{n}}_k = \delta_{ik},\\ \left[\mathbf{D}_j\mathbf{x}_i\right]\left(\frac{\partial \xi_j}{\partial x_k}\right)_{\text{strong}} &= \delta_{ik} - \mathbf{J}^{-1}\mathbf{L}(\hat{\mathbf{x}}_i - \mathbf{x}_i)\mathbf{J}_e\hat{\mathbf{n}}_k,\end{aligned}$$

where, the Jacobian factor $\frac{\partial \xi_j}{\partial x_k}$ takes a value at each nodal point in the domain, and $\hat{\mathbf{n}}_k$ is the kth component of the normal. Numerically, $\mathbf{D}_j\mathbf{x}_i$ is a matrix–vector multiplication, while the Jacobian factor is multiplied term-by-term to the result (i.e. Hadamard/Schur product). We could have also solved for the Jacobian factors using the discrete analogue of (82) as:

$$\left(\frac{\partial \xi_j}{\partial x_k}\right)_{\text{strong}} = \left[\mathbf{D}_j\mathbf{x}_i\right]^{-1}\left[\delta_{ik} - \mathbf{J}^{-1}\mathbf{L}(\hat{\mathbf{x}}_i - \mathbf{x}_i)\mathbf{J}_e\hat{\mathbf{n}}_k\right], \tag{85}$$

which simplifies to the form given by Hesthaven and Warburton [27]

$$\left(\frac{\partial \xi_j}{\partial x_k}\right)_{\text{strong}} = \left[\mathbf{D}_j\mathbf{x}_i\right]^{-1} \tag{86}$$

for a continuous mesh. Let's give a few remarks about the form (85):

1. The Jacobians satisfy the identity (81) analytically in the volume term: $\left[\mathbf{D}_j\mathbf{x}_i\right]\frac{\partial \xi_j}{\partial x_k} \equiv \left[(\mathbf{M}^{-1})(\mathbf{S}\mathbf{x}_i)\right]\frac{\partial \xi_j}{\partial x_k}$.
2. The edge-term $\mathbf{J}^{-1}\mathbf{L}(\hat{\mathbf{x}}_i - \mathbf{x}_i)\mathbf{J}_e\hat{\mathbf{n}}_k = \mathbf{J}^{-1}\mathbf{M}^{-1}\mathbf{M}_e(\hat{\mathbf{x}}_i - \mathbf{x}_i)\mathbf{J}_e\hat{\mathbf{n}}_k$ does not require a Jacobian factor since $\hat{\mathbf{n}}$ is the real-space normal, and the edge Jacobian is used.
3. The edge-term calculated here (the element-local equation for the gradient) matches the discrete form of the HDG edge-conservation equation $\left\langle [\![\widehat{\mathbf{q}_{\delta p}} \cdot \hat{\mathbf{n}}]\!], \theta_\varepsilon \right\rangle_e \approx \mathbf{M}_\varepsilon [\![\widehat{\mathbf{q}_{\delta p}} \cdot \hat{\mathbf{n}}]\!]\mathbf{J}_e$. The HDG flux-conservation equation is used to enforce the continuity constraint (31), and needs to be consistent with the discrete continuity constraint in the element-local equations (29) in order to satisfy continuity numerically.

Remark (1) holds analytically for the continuous operators and leads to a convenient simplification for the numerical scheme without loss of accuracy. Remark (2) reflects an important choice made for this scheme, and this choice leads to the result of remark (3). The challenge for the quadrature-free implementation (which remark (3) solves) is maintaining numerically-consistent edge integration terms. Without consistent edge integrals, the conservative flux calculated on the HDG space will not be numerically conserved in the element-local calculations. This often makes the numerical solution of the Navier–Stokes equations unstable.

To appreciate the advantages of (85), consider the same approach but using the weak form (84). This leads to the following for the Jacobian factors:

$$\left(\frac{\partial \xi_j}{\partial x_k}\right)_{\text{weak}} = \left[-\mathbf{M}^{-1}\mathbf{S}_j^T\mathbf{x}_i\right]^{-1}\left[\delta_{ik} - \mathbf{J}^{-1}\mathbf{L}(\hat{\mathbf{x}}_i)\mathbf{J}_e\hat{\mathbf{n}}_k\right].$$

While this may appear reasonable, the operator $\left[-\mathbf{M}^{-1}\mathbf{S}_j^T\mathbf{x}_i\right]$ can be singular, requiring a pseudo-inverse. Using this weak form is thus not as direct. Alternatively, if we evaluate the edge-integrals in the reference space, this would give:

$$\frac{\partial \xi_j}{\partial x_k} = \left[\mathbf{D}_j\mathbf{x}_i + \mathbf{L}(\hat{\mathbf{x}}_i - \mathbf{x}_i)\mathbf{J}_e^{\text{ref}}\hat{\mathbf{n}}_k^{\text{ref}}\right]^{-1}.$$

While this approach works for both weak and strong forms, these edge integrals are no longer consistent with the HDG integrals. Even though the form of the edge-integrals in the HDG method could be modified, the above weak-form approach is less efficient than the strong formulation (85). This approach requires d^2 edge integrals, while the strong form (85) only requires d calculations, where d is the dimension of the problem.

The calculation of the Jacobians and edge normals follow the usual approach in quadrature-free methods. As such, the only challenge was dealing with the Jacobian factors and the quadrature-free derivative terms. We showed that the strong formulation has distinct advantages over the weak formulation in this case. We verify that this approach works in §6, using a simple steady diffusion problem.

5.2. Dealing with oscillations: filtering and limiting

The non-linear advection terms in the Navier–Stokes equations offer unique challenges. In particular, high order methods can develop non-physical oscillations that can lead to numerical instabilities. As such, robust high-order solvers deal with these oscillations. Applying a filter or a limiter are two approaches that can suppress oscillations. A filter damps the modal polynomial coefficients according to a given spectrum, where higher-frequency modes are usually damped more. A limiter ensures that the solution remains within calculated bounds, usually determined from neighboring elements or nodes. While

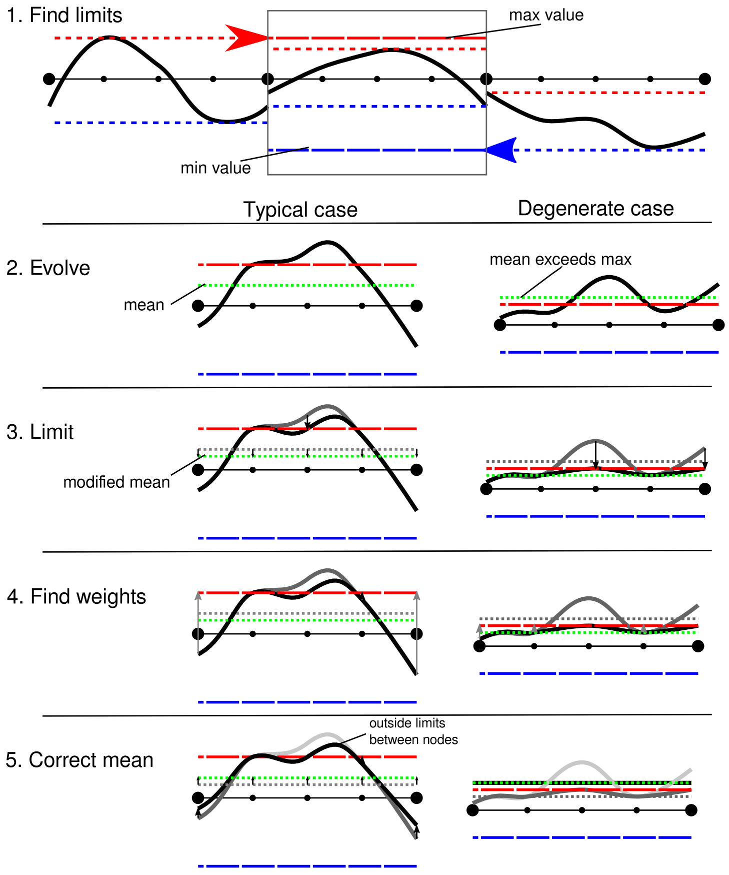

Fig. 9. Graphical representation of the nodal limiting procedure from our high-order selective nodal limiter. The nodal limiting procedure is sketched for a typical and degenerate case in the left and right columns, respectively.

limiting traditionally focuses on reconstructing completely oscillation-free solutions, we are primarily interested in enhancing the stability of our numerical solutions. The selective filters and limiters that we have evaluated and developed for this purpose are described next.

Our initial selective filtering approach is described in [59]. We used an exponential filter [27,26], where the modal coefficients are modified using a function σ that decays exponentially with the polynomial degree [36]. The selectivity of the filter was determined by comparing the decay of the modal coefficient to a reference spectrum. In [59], there was only a filter. Here, we first present our new nodal limiting procedure and then our improved selective filter.

(i) Nodal limiter: Our limiting procedure is based on previous nodal limiting methods used for second-order accurate methods [28]. We have modified this method for high-order nodal DG. There are some drawbacks to our modifications, in particular our procedure is Total Variation Bounded (TVB) as opposed to TVD, and without the selectivity, it does not remain high-order accurate. However, it does successfully stabilize the numerical solution by suppressing spurious oscillations, and with the selectivity criterion the solution does remain high-order accurate.

Our procedure can be understood in 5 steps (Fig. 9). The first step is to find the limits, or the initial total variation of the solution. That is, we determine the allowed maximum and minimum values for each element. Presently, this is done by finding the maximum and minimum values of the solution in the present and neighboring elements.

$$\phi_{\max} = \max(\phi)_{K^\pm}, \qquad \phi_{\min} = \min(\phi)_{K^\pm}, \tag{87}$$

where $K^\pm$ includes the element K and all its immediate neighbors. Using the terminology of [28], this is similar to choosing $\alpha = 1$. While our present step 1 may cause the nodal limiter to fail the Hoteit et al. [28] “stair-step” numerical example, a small modification should guarantee the correct solution: if the maximum and minimum values of the function is determined solely by the present and upwind neighboring elements, then our nodal limiter should correctly solve the “stair-step” problem. Another possible modification is to determine the maximum and minimum values for individual nodes by looking at the values of neighboring nodes. This reduces the allowed variation for each node, which could also improve accuracy. Nonetheless, our primary concern is stability, so these questions are left for future research. The present nodal limiter is efficient, simple to implement, and guarantees that the solution will remain bounded. For example, if the density is positive everywhere in the domain, these limits will never be negative. Once the bounds have been determined, the limiting procedure can continue.

The first step determined the limits or total variation of the field before evolving it in time, and this was the only operation that requires information from neighboring elements. The remaining steps are all element-local.

In the second step of the limiting procedure, the solution is evolved using the right-hand-side forcing without limiting

$$\bar{\phi}^{k+1} = \phi^k + \Delta t F_\phi. \tag{88}$$

The right-hand-side forcing terms, particularly the advection terms, can introduce oscillations. Thus, this new solution may exceed the limits calculated in step 1.

The third step, limits the nodal values. That is, we find nodes where the evolved solution exceeds the limits determined in step 1, and we calculate a forcing $\bar{F}_\phi^{\text{limit}}$ (which is the first predictor for the limiter forcing) that sets those nodes equal to the appropriate maximum or minimum values.

$$\bar{\bar{\phi}}^{k+1} = \phi^k + \Delta t F_\phi + \Delta t \bar{F}_\phi^{\text{limit}} \tag{89}$$

In this step, however, the mean of the initial solution in the element could be modified. As it is important to conserve mass, the mean in the element has to be re-adjusted. Thus, the change in the value of the mean in the element K caused by the adjustment is calculated

$$\Delta \text{mean}\left(\bar{\bar{\phi}}\right)_K = \text{mean}(\bar{F}_\phi^{\text{limit}})_K. \tag{90}$$

In other words, we want the final limiter forcing to have: $\text{mean}(F_\phi^{\text{limit}})_K = 0$.

The fourth step finds weights that determine by how much different nodes can move to help with the re-adjustment of the mean in the element. While previous researchers have developed sophisticated ways to minimize the error of this adjustment, here we use a heuristic approach. If the mean was lowered or raised, we calculate the maximum upward or downward adjustment allowed for each node, respectively, which we denote $F_{\text{max, adjust}}$. For example, if a node is already at the maximum value, it cannot be adjusted upwards, and will therefore have a zero effective weight. The nodes furthest away from the bounds will have the largest weight.

In the fifth and final step, we scale the maximum adjustment weight calculated in step four by the required amount to correct the mean of the limiter forcing. That is, we can now calculate the final limiter forcing

$$F_\phi^{\text{limit}} = \bar{F}_\phi^{\text{limit}} - \frac{\text{mean}(\bar{F}_\phi^{\text{limit}})_K}{\text{mean}(F_{\text{max, adjust}})_K} F_{\text{max, adjust}}, \tag{91}$$

where the sign of the final adjustment depends on the sign of the calculated maximum adjustment.

In the illustration of these steps (Fig. 9), we sketch a normal and degenerate case. For the normal case, note that while the solution is limited at the nodes, between nodes the polynomial is allowed to exceed the bounds. The issue with the degenerate case is that the initial mean of the solution exceeds the calculated bounds. As such the final solution is a constant equal to the original mean; in other words, all oscillations are removed. This situation can occur for the advection operator if it is treated explicitly and the Courant–Friedrichs–Lewy (CFL) condition is violated. In this situation, our nodal limiter can actually stabilize the solution and prevent instability if the CFL condition is not violated by too large a margin. The forcing due to implicit diffusion can also create a degenerate situation and this case is discussed in [58]. In general, different bounds and limiting can be used for different terms in the governing equations. In particular, some terms introduce new variation in the solution and this should not be limited. However, these terms should be well-resolved, or appropriately smoothed to avoid oscillations.

(ii) Filter: We have described how to calculate the forcing term F_ϕ^{limit} that will appropriately limit the solution to be within the calculated bounds. A similar function for the filter can also be calculated, F_ϕ^{filter}. For details on this exponential filter forcing, see Ueckermann and Lermusiaux [59].

(iii) Selective limiting/filtering: At this stage both the nodal limiter and filter would be fully applied everywhere in the domain. However, our aim is to selectively restrict this limiting/filtering to only the parts of the domain where required. What remains is thus an appropriate weighting function, $\alpha(\mathbf{x}, t)$. For this, we use a discontinuity sensor similar to the one used by Huerta et al. [29], which was proposed in Persson and Peraire [50] and Nguyen et al. [39]. The difference is that we do not include the coefficient of the zero-degree polynomial in the denominator of the sensor (see below).

The discontinuity sensor works as follows. First, the coefficients of the nodal basis are transformed to modal-basis coefficients. To do so, we need to form the Vandermonde matrix $\mathcal{V}_{ij} = \theta_j^M(\mathbf{x}_i)$, where θ_j^M is the jth modal polynomial, and $\mathbf{x}_i$ is the ith nodal point. The modal coefficients can then be found as $\phi_i^M = \mathcal{V}^{-1}\phi_i$. We can then compare the decay of the modal coefficients to the decay of reference spectra. To do so, we follow Huerta et al. [29], and define the weight as:

$$\alpha^\star = \frac{1}{\beta^{\text{top}} - \beta^{\text{bot}}} \left\{ \log_{10} \left(\frac{\sum\limits_{\{i:\theta_i^M \in \mathcal{P}^{\mathrm{p}^\star}\}} (\phi_i^M)^2}{\sum\limits_{i>0} (\phi_i^M)^2} \right) - \beta^{\text{bot}} \right\}, \tag{92}$$

$$\alpha = \min(\max(\alpha^\star, 0), 1) \tag{93}$$

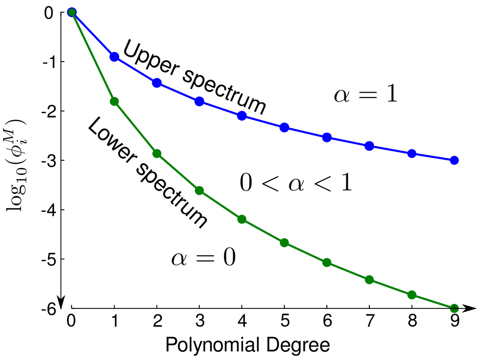

Fig. 10. Sketch of the selectivity criterion. The solution is fully limited, partially limited, or unmodified if the modal-polynomial-coefficient-decay is slower than the top reference spectrum $\log_{10}([\mathrm{p}+1]^{-3})$, between the two reference spectra, and faster than the bottom spectrum $\log_{10}([\mathrm{p}+1]^{-6})$, respectively.

where $\mathrm{p}^\star \geq \mathrm{p}$, $\beta^\star = \log_{10}\left(\frac{\sum_{\{i:\theta_i^M \in \mathcal{P}^{\mathrm{p}^\star}\}}(\phi_i^{M,\star})^2}{\sum_{i>0}(\phi_i^{M,\star})^2}\right)$ and $\phi_i^{M,\star}$ are the modal coefficients for the $\star$ reference spectrum. What this indicator does is compare the sum of the coefficients for the highest degree polynomial to the sum of the coefficients for the polynomials of degree greater than zero. A notable difference between our indicator and that defined in Huerta et al. [29] is that we do not include the coefficient of the zero-degree polynomial in the denominator. This is because the constant term can be arbitrarily scaled (based on non-dimensionalization, for example) and should not have an impact on the smoothness indicator. Note that α is variable in time and space: it is a scalar in every element. Two reference spectra and the ranges for α are sketched in Fig. 10. Depending on the smoothness of the modal coefficients of the numerical solution, the weights for α can be anywhere in [0, 1]. Our approach is different from that of Huerta et al. [29] as we do not use the Mach number for the discontinuity sensor, and we do not decompose the high-order element into low-order sub-domains.

Note that we calculate the selectivity index by examining the field before the advection term is added (that is, at the start of the IMEX-RK stage). At each subsequent stage in the IMEX-RK time-stepping procedure, the selectivity index is updated. Alternatively, we could calculate the selectivity index after the advection is added (that is, advanced in time due to advection at that IMEX-RK stage). Another option is calculating the smoothness index based on the advection term itself, or any combination of the above-mentioned options. Additionally, to increase efficiency, the selectivity index could only be updated at the first or final stage of the IMEX-RK time-stepping procedure. Here, we do not examine the effect of these choices.

Finally, the selective nodal limiter is applied as a weighted forcing term, which gives the final evolution of ϕ as:

$$\phi^{k+1} = \phi^k + \Delta t F_\phi + \Delta t \alpha^s F_\phi^{\mathrm{limit}}, \tag{94}$$

where $s = 0$ gives a non-selective nodal limiter, and $s = 1$ gives a linear weighting between the reference spectra. For $s > 1$, the solution is weakly limited close to the lower spectrum, and for $0 < s < 1$ the weight quickly increases. The same selectivity criterion can also be applied to the forcing calculated from our filter [59], leading to a new selective filter.

In summary, we derived a new selective nodal-limiting/filtering procedure. It is based on existing nodal-based limiters, but extends to higher-order polynomials and employs an inexpensive heuristic to ensure mass conservation. We combined this nodal limiter with a smoothness indicator to selectively limit the solution spatially and temporally: the selectivity index is used to switch off the lower-order limiter and so preserve higher-order accuracy. The resulting selective nodal limiter is tested in §6 and high-order convergence is indeed observed when the solution is sufficiently resolved.

6. Verification of HDG diffusion and selectively-limited advection

Verification of a new code is necessary to ensure that it solves the intended equations [46,52]. To verify that the proposed schemes work, we perform convergence studies on simple equations. To show that our quadrature-free scheme works for HDG schemes, we perform a convergence study on straight and curved meshes. Then we verify that the selective nodal limiter recovers high-order convergence rates when the solution is adequately resolved.

6.1. Verification of quadrature-free hybridizable discontinuous Galerkin scheme

To verify that our HDG implementation works on curved meshes, we perform a convergence study on a steady diffusion problem

$$\nabla^2\phi = f \ \text{ on } \Omega, \tag{95}$$

$$\phi = g_D \ \text{ on } \partial\Omega_D, \tag{96}$$

$$(\nabla\phi)\cdot\hat{\mathbf{n}} = g_N \ \text{ on } \partial\Omega_N, \tag{97}$$

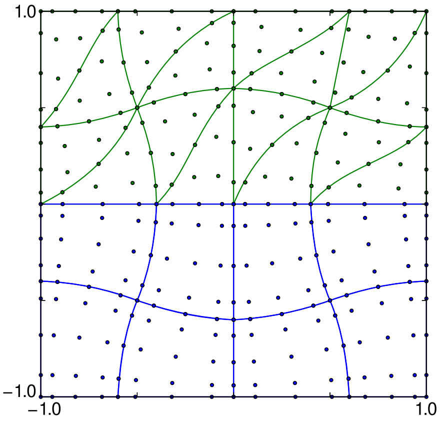

Fig. 11. Curved mesh for the $\Delta x = 0.5$, $\mathrm{p} = 4$ simulation. Triangular and rectangular elements are colored green and blue, respectively.

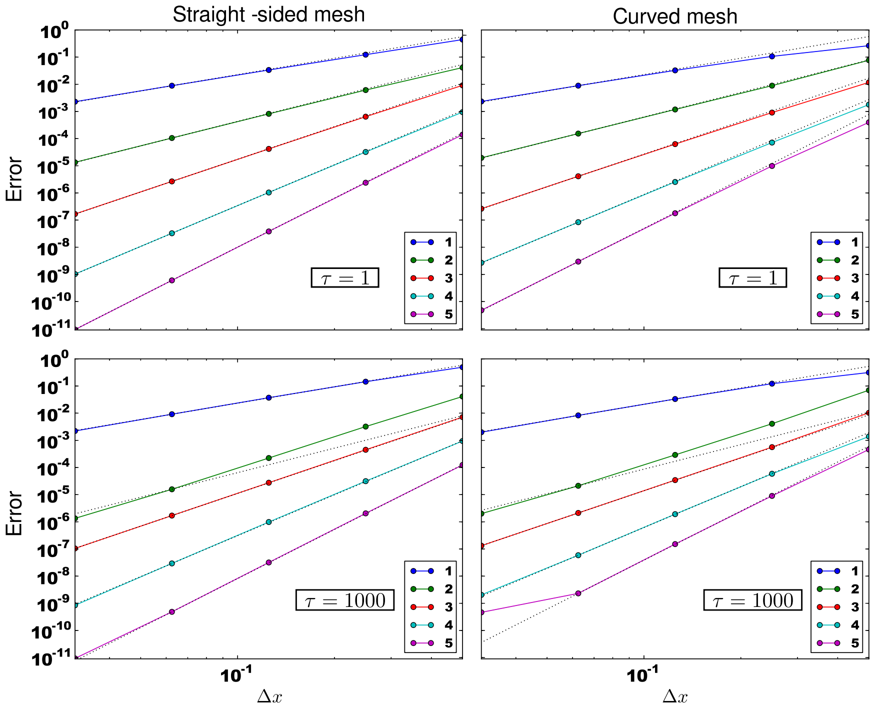

Fig. 12. Spatial convergence (L^2-norm) of diffusion on straight (left) and curved (right) meshes. Dashed lines give the optimal convergence rates.

where

$$f = \sin(\pi(x + x_0))\sin(\pi(y + y_0)), \tag{98}$$

$x_0 = y_0 = 0.3$, and the bottom and right boundaries are Dirichlet ($\partial\Omega_D$), while the top and left boundaries are Neumann ($\partial\Omega_N$) on the domain $\Omega \in [-1, 1] \times [-1, 1]$. We use both straight and curved meshes. The curved mesh for $\Delta x = 0.5$, $\mathrm{p} = 4$ is shown in Fig. 11, and it is made up of a mixture of triangular and rectangular elements. We also perform the convergence study for two different values of the HDG stability parameter: $\tau = 1$ and $\tau = 1000$.

Both the straight-sided and curved mesh simulations converge near-optimally for both values of τ (Fig. 12). The error level (L^2-norm) is generally lower for the straight-sided mesh. While the error levels are generally similar for the different values of τ, the $\mathrm{p} = 2$ result using $\tau = 1000$ seems to converge faster for both meshes. However, the $\mathrm{p} = 5$ result using $\tau = 1000$ suggests the larger value of τ reaches machine precision earlier, possibly due to a larger condition number in the matrix.

These results verify that our scheme works for straight and curved meshes with mixed element types.

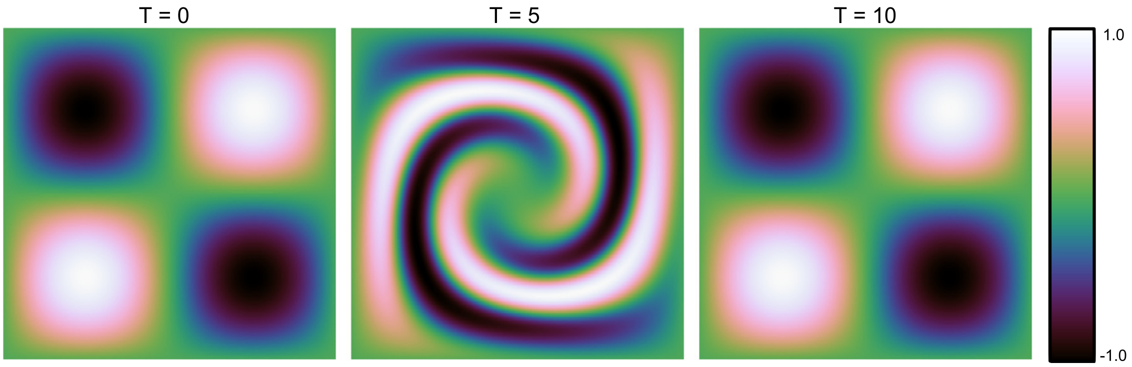

Fig. 13. Tracer concentration at $T = 0,\ 5,\ 10$ (left, center, and right, respectively) for the advection benchmark using $\mathrm{p} = 5$, $\Delta x = \frac{1}{64}$.

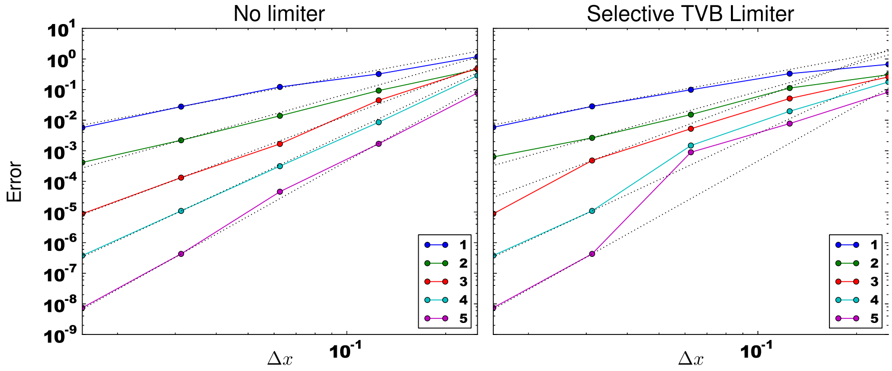

Fig. 14. Spatial convergence (L^2-norm) of advection equation without (left) and with the selective nodal limiter (right). The spatial resolutions used are $\Delta x = \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{64}$. Dashed lines give the optimal convergence rates.

6.2. Verification of selective nodal limiter

In §5.2 we developed a selective nodal limiter and filter, and here we test the effect of the selective nodal limiter. To do so, we study a modification of the swirl problem in chapter 5 of Durran [19]. We do not show results for the filter, see §5.2(ii)–(iii), because the nodal limiter is more robust, accurate, and does not require tuning. We solve the unsteady advection problem

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{v}\phi) = 0 \ \text{ on } \ \Omega,$$

$$\phi = 0 \ \text{ on } \ \partial\Omega_D,$$

on the domain $\Omega = [0,\ 1] \times [0,\ 1]$ with Dirichlet BCs everywhere, over the time interval $T = [0, 10]$. The time-varying velocity is specified as

$$\mathbf{v} = \sin\left(\frac{\pi}{5}t\right)\left[\frac{1}{2}\sin(2\pi y)\sin^2(\pi x),\ -\frac{1}{2}\sin(2\pi x)\sin^2(\pi y)\right],$$

and to have a smooth solution with positive and negative values for the convergence test, we modified the initial condition for the tracer from [19] to

$$\phi(\mathbf{x},\ t = 0) = \sin(2\pi x)\sin(2\pi y).$$

The specified flow field causes the initial tracer concentration to swirl during the interval $T = [0, 5]$ (Fig. 13). In the interval $T = [5, 10]$ the flowfield reverses direction, causing the tracer to "un-swirl." Thus, the final tracer concentration should be the same as the initial tracer concentration. Using this property, we can compute the error by comparing the initial $\phi(\mathbf{x}, t = 0)$ and final $\phi(\mathbf{x}, t = 10)$ fields.

For these simulations, we used a fixed time-step $\Delta t = 10^{-3}$, and a second-order accurate explicit RK time-integrator (with the same coefficients as the IMEX-RK integrators used later). The mesh is composed of uniform quadrilateral elements. The selectivity index, (93), uses $(\mathrm{p}+1)^{-3}$ and $(\mathrm{p}+1)^{-6}$ for the top and bottom reference spectra, respectively.

The simulations without the selective nodal limiter converge near-optimally (Fig. 14). However, with the selective nodal limiter, for too coarse spatial discretizations, the higher order ($\mathrm{p} > 1$) simulations reduce to second-order accuracy. This

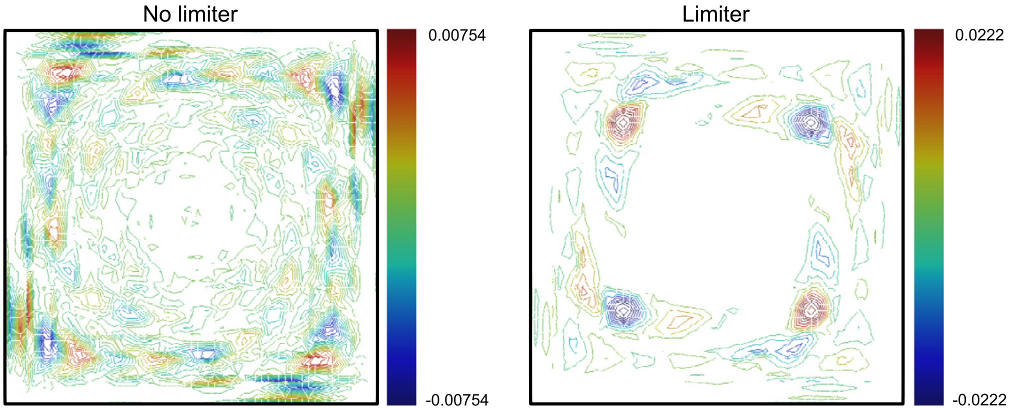

Fig. 15. Errors of the tracer advection test-case for the intermediate resolution, p = 3, $\Delta x = \frac{1}{16}$, case in Fig. 14, without (left) and with the selective nodal limiter (right).

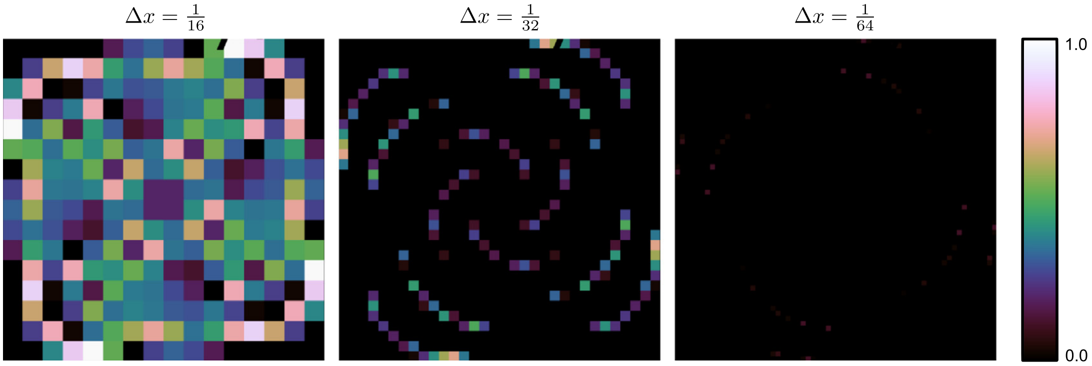

Fig. 16. Selectivity index α (93) for the tracer advection test-case using p = 3 at resolutions of $\Delta x = \frac{1}{16}, \frac{1}{32}, \frac{1}{64}$ on the left, center, and right, respectively.

is because the limiter is fully applied at these resolutions, causing the tops of the sinusoidal tracer concentrations to be chopped (Fig. 15). Once the mesh is sufficiently refined, the effect of the nodal limiter is reduced by our selectivity criterion, and the higher-order convergence rate is recovered. If we examine the value of the selectivity index at $T = 5$ for the p = 3 case (Fig. 16), we see that with increasing resolution the selectivity index α takes a smaller value and is localized in space near sharp gradients. Recall, the nodal limiter is fully applied for $\alpha = 1$ and not applied at $\alpha = 0$. Therefore, at these higher resolutions, the selective nodal limiter is only active in localized regions throughout the domain. This allows the higher-order rates of convergence to be recovered at these resolutions (see Fig. 14).

Overall, these results verify both that our quadrature-free advection scheme is properly implemented, and that our selective nodal limiter can recover higher-order accuracy when the solution is sufficiently resolved.

7. Verification and validation of Stokes/Navier–Stokes HDG solvers

In this section we verify and validate our new algorithm derived in §3.2. We perform detailed convergence studies using a manufactured solution. We then solve a lock-exchange problem, comparing our density contours and Froude numbers to existing literature.

7.1. Definition of analytical benchmark

To evaluate the implementation of our new scheme, we use a manufactured benchmark of Guermond et al. [23]. For this case, consider a domain $\Omega \times [0, T]$ where $\Omega = [-1, 1] \times [-1, 1]$. The solution $[\mathbf{v}, p]$ is defined as

$$\mathbf{v}(x, y, t) = \pi \sin(t) \left[\sin(2\pi y)\sin^2(\pi x), \; -\sin(2\pi x)\sin^2(\pi y)\right], \tag{99}$$

$$p(x, y, t) = \sin(t)\cos(\pi x)\sin(\pi y). \tag{100}$$

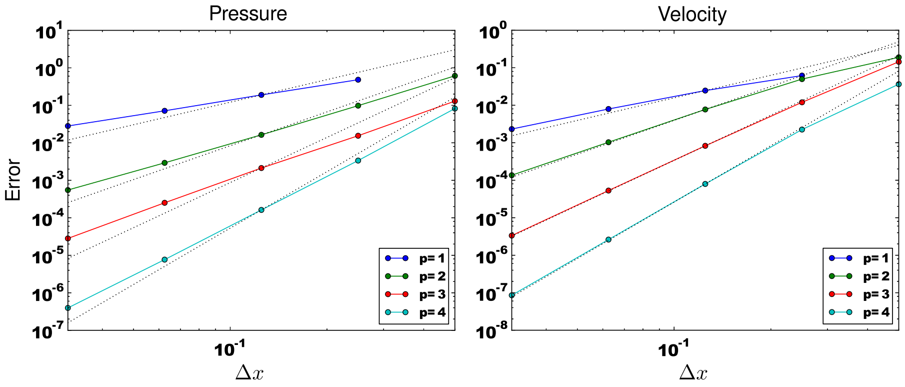

Fig. 17. Spatial convergence (L^2-norm) of pressure (left) and velocity (right) using the analytical Stokes problem with Re = 1, and $\tau_p = \frac{1}{a\Delta t\tau} = 10^5$. A second-order accurate IMEX integrator with time-step fixed at $\Delta t = 10^{-5}$ is used. Dashed lines give the optimal convergence rates.

From these definitions, we can calculate the forcing term $\mathbf{F}_{\partial t}$, which is

$$\begin{aligned}
\mathbf{F}_{\partial t} &= \frac{\partial \mathbf{v}}{\partial t} - \nabla \cdot \frac{1}{\text{Re}} \nabla \mathbf{v} + \nabla p, \\
F^u_{\partial t} &= \pi \cos(t) \sin(2\pi y) \sin^2(\pi x) - \frac{2\pi^3}{\text{Re}} \sin(t) \sin(2\pi y) \cos^2(\pi x) \\
&\quad + \frac{6\pi^3}{\text{Re}} \sin(t) \sin(2\pi y) \sin^2(\pi x) - \pi \sin(t) \sin(\pi x) \sin(\pi y), \\
F^v_{\partial t} &= -\pi \cos(t) \sin(2\pi x) \sin^2(\pi y) + \frac{2\pi^3}{\text{Re}} \sin(t) \sin(2\pi x) \cos^2(\pi y) \\
&\quad - \frac{6\pi^3}{\text{Re}} \sin(t) \sin(2\pi x) \sin^2(\pi y) + \pi \sin(t) \cos(\pi x) \cos(\pi y).
\end{aligned} \tag{101}$$

This provides a smooth analytical solution with which we can verify the spatial and temporal convergence, as well as other numerical results.

7.2. Convergence studies

For the spatial convergence study using the benchmark given in §7.1, we find that the velocity and pressure converge near optimally (Fig. 17). The convergence of the pressure is slightly lower, but this can be attributed to the splitting error (Fig. 19). Note that if an inconsistent value of τ_p is used, the solution can become unstable at coarse resolutions [58].

The temporal convergence is more involved due to the additional complexity introduced by the projection method's time-splitting. As such, we will test convergence with and without the rotational pressure-correction, and for various Reynolds numbers. We always use the consistent value $\tau_p = \frac{1}{\tau a \Delta t}$ in these studies. The time-rates of convergence for the rotational form and standard pressure-correction form show that the rotational correction decreases the error in the pressure field, without as large an impact on the velocity field (Fig. 18). The rotational correction removes part of the pressure error near the boundary of the domain (Fig. 19), as expected from [23].

As seen on Fig. 18, the irreducible splitting error from the projection method can restrict the accuracy to second order in time. Because the splitting error is proportional to $\frac{1}{\text{Re}}$ (see [58]), we can verify the correctness of our time-integration method by considering an infinite Reynolds number. When we do so, we find that pressure and velocity converge optimally (Fig. 20).

To test the effect of the Reynolds number on the time integration accuracy, we calculated the velocity and pressure errors for various Re. We note that the error in the pressure steadily decreases with increasing Re, and then saturates at Re = 10^6 (Fig. 21). The velocity error is not drastically affected for the second and third-order time-integration schemes, but the first-order scheme's error increases for increasing Re before plateauing. For the pressure, the calculated order of convergence approaches second-order, then decreases as the error plateaus. For the velocity, the calculated order is mostly unaffected. The transition from Re = 10^7 to Re = ∞ is not smooth, suggesting that the mere presence of the diffusion operator has an effect numerically. This indicates that to benefit from a time-integration scheme higher than second-order when the implicit diffusion terms are present, iterative HDG schemes would be needed (e.g. [22]). However, this projection-method restriction may not be the limiting factor for the solution accuracy in time. First, since the time-step is restricted by the CFL condition for advection, the temporal dimension is often more finely discretized than the spatial one, particularly when the non-dimensional physical advection-speed exceeds unity. Second, if higher-order temporal accuracy is required, the projection method can indeed be used as an iterative scheme, where the pressure-predictor of the second iteration is the

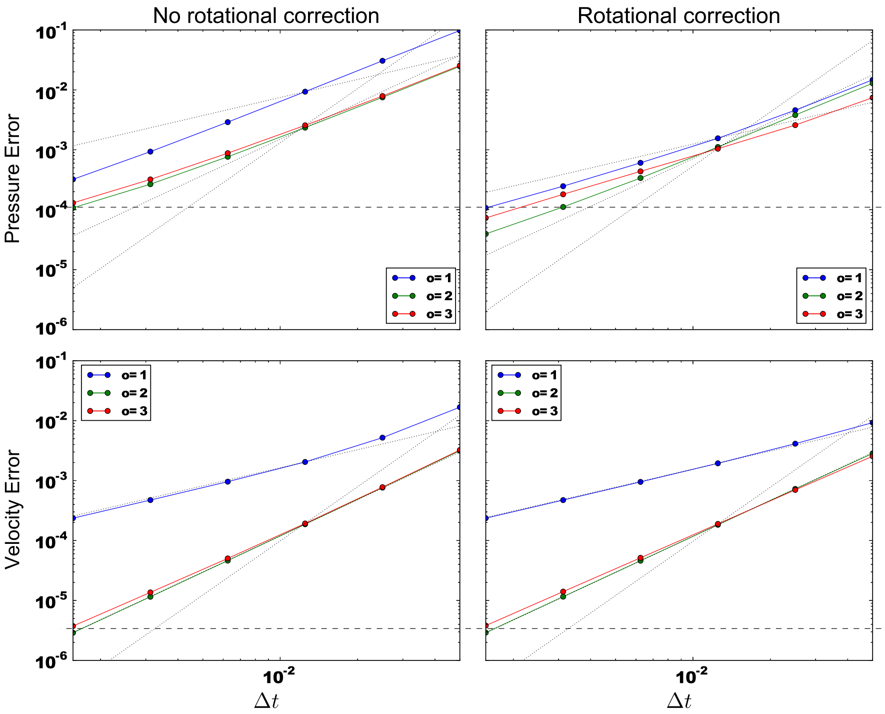

Fig. 18. Temporal convergence (L^2-norm) of pressure (top) and velocity (bottom) using the analytical Stokes problem with $\mathrm{Re} = 1$. A 64 × 64 square mesh with $\mathrm{p} = 6$ was used for space, and first to third order accurate IMEX RK schemes for time. The rotational correction is applied (right), and not applied (left). The rotational correction lowers the absolute pressure-error. Dashed lines give the optimal convergence rates.

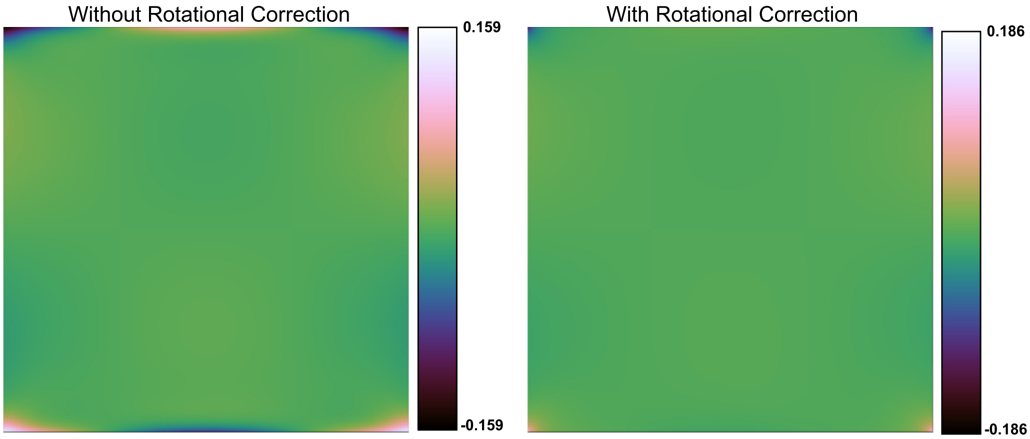

Fig. 19. Pressure error for second order IMEX-RK time integration using $\Delta t = 0.1$ for our standard (left) and rotational (right) HDG pressure corrections. The rotational correction removes errors at the boundary of the domain, but errors at the corners remain.

final pressure from one full execution of the projection method (e.g. [20]). This may still be more computationally efficient than solving a fully coupled system of equations. The fully coupled HDG system requires the inversion of a matrix that is $d + 1$ times larger than our smaller HDG matrices. Since matrix inversion often scales as the square of the number of unknowns, one full execution of the projection method is expected to be roughly $(d + 1)^2$ times more efficient (in $d = 3$, this is a factor of 16). Thirdly, the results from a full execution of the projection method could also be used as a starting guess (or preconditioner) for the fully uncoupled problem.

In this section we evaluated our implementation and convergence rates. We showed that we can obtain near-optimal spatial and temporal rates of convergence when using a manufactured solution. Next we validate our code against a standard benchmark case to ensure that our schemes give the correct solution for an unforced case.

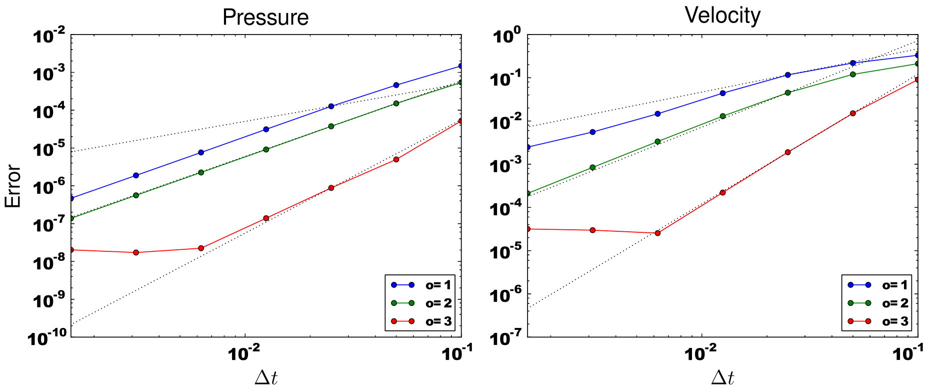

Fig. 20. Temporal convergence (L^2-norm) of pressure (left) and velocity (right) using the analytical Stokes problem with $\mathrm{Re} = \infty$. A 64 × 64 square mesh with $\mathrm{p} = 6$ was used for space, and first to third order accurate IMEX RK schemes for time. Dashed lines give the optimal convergence rates.

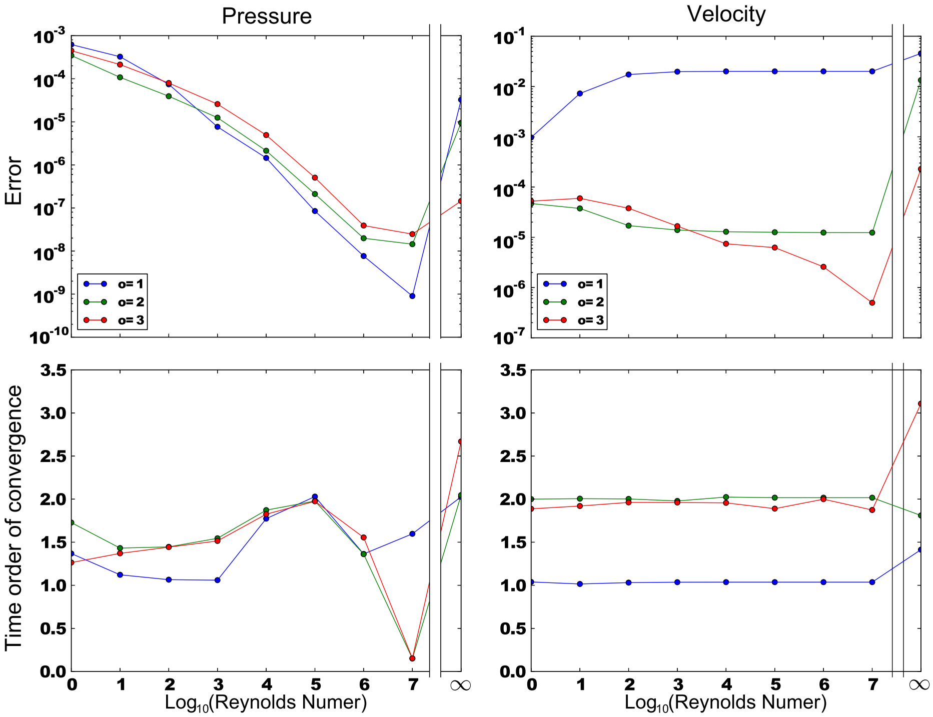

Fig. 21. Error (top) of pressure (left) and velocity (right), and order of temporal convergence (bottom) using the analytical Stokes problem with $\mathrm{Re} = 1$ (L^2-norm). A 64 × 64 square mesh with $\mathrm{p} = 6$ was used for space, and first to third order accurate IMEX RK schemes for time. The error is plotted for $\Delta t = 0.0125$ and the order of convergence was calculated using $\Delta t = 0.025, 0.0125$. As Re increases, the pressure error decreases while the velocity error increases for the first-order IMEX scheme, with less effect for the higher-order schemes. The order of convergence remains unaffected, but when $\mathrm{Re} = \infty$, near optimal convergence is obtained for velocity.

7.3. Validation

We validate our scheme by using a Lock-exchange problem with the same non-dimensional parameters as those of Härtel et al. [25] and Fringer et al. [21]. Our simulation uses a 2D domain of size $[-8, 8] \times [0, 2]$, discretized using uniformly sized and distributed quadrilaterals of various resolutions, and we integrate for $T = [0, 10]$. We use the no-slip boundary condition at all boundaries, a Schmidt number of $\mathrm{Sc} = 1$, and a Grashof number of $\mathrm{Gr} = \frac{g'h^3}{\nu^2} = 1.25 \times 10^6$, where $g' = \frac{\Delta\rho}{\rho_0}$ is the reduced gravity and $h = 1$ is the half-height of the domain. The initial density profile is defined as

$$\rho = \frac{1}{2}\tanh(10^5\mathbf{x}).$$

We use a second-order accurate time-integration scheme, with a fixed time-step of $\Delta t = 0.001$.

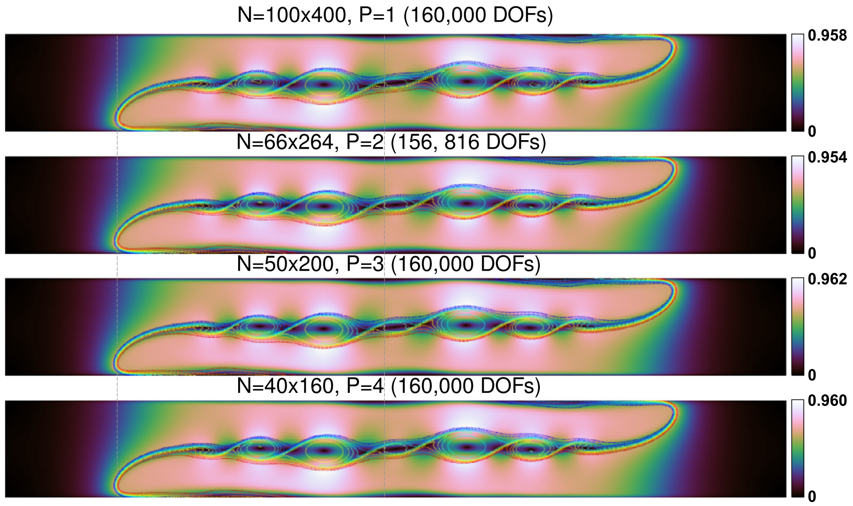

Fig. 22. Density solution at time 10 of the lock-exchange problem ($\text{Gr} = 1.25 \times 10^6$) using various orders of accuracy and spatial resolution, all runs with approximately 160,000 degrees of freedom. There are some minor differences in the front propagation speed and the shape of the Kelvin–Helmholtz instabilities.

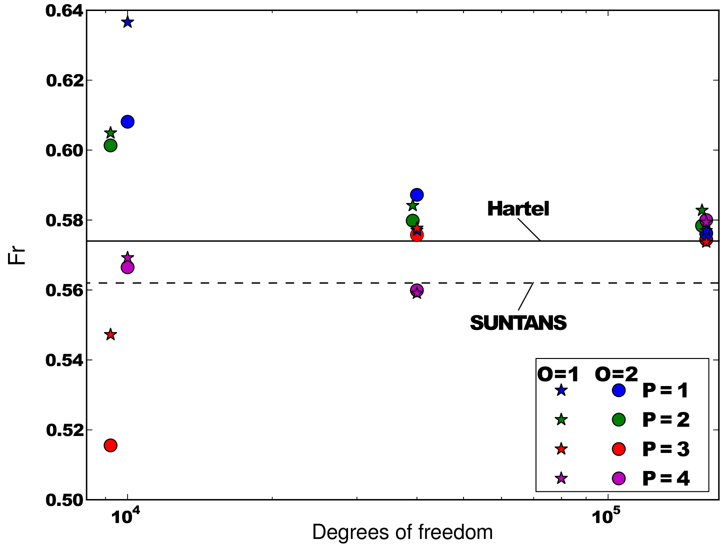

Fig. 23. Density front propagation speed for various resolutions for the no-slip case (⋆: first-order in time; •: second-order in time). Solid and dashed lines indicate the solution obtained by HARTEL and SUNTANS respectively.

To compare our results to [25] and [21], we compute the Froude number $\text{Fr} = \frac{u_f}{u_b}$, where u_f is the speed of the front, defined as the speed at which the foremost point of the front travels, and $u_b = \sqrt{g'h}$ is the buoyancy velocity. To estimate u_f, we find the foremost point of the front at $T = 5$ and $T = 10$, then we simply use $u_f = \frac{\Delta x}{\Delta T} = \frac{\Delta x}{5}$, which gives an average front speed over that time period. Finding the foremost point of the front is non-trivial for cases using higher-order polynomial bases. In those cases we first identify the element that contains the foremost point of the front. Following this, we do an iterative root-find and line-search to find the foremost point (see [58]).

Our density contours are similar to those calculated by Härtel et al. [25] (Fig. 22). The higher-order simulations also match the second-order accurate simulation. Comparing the Froude numbers, as the spatial resolution is refined, our answer approaches the value of Härtel et al. [25], and the spread in our results is on the same order as the difference between Härtel et al. [25] and Fringer et al. [21] (Fig. 23). We note that the first-order time integration scheme performs nearly as well as the second-order time integration scheme for higher spatial resolutions. This suggests that the temporal dimension is well-resolved. Also, as the spatial resolution increases, the agreement among simulations with different spatial order of accuracy increases. For the low spatial resolution cases, there is a larger spread of values between the high and low-order runs. In this case, finding the location of the foremost point of the front may play a role in the error. Nonetheless, our results agree closely with [25] and [21].

It would be interesting to compare the accuracy level and computational effort for the various polynomial degrees of freedom shown in Fig. 22. However, practical numbers can be highly depended on the implementation and the architecture (e.g. parallel vs serial). If the implementation is memory-bandwidth limited, the computational effort should be roughly correlated with the total degrees of freedom of a discretization, in which case Fig. 23 gives an indication of the differences between high- and low-order results.

8. Summary and conclusions

In this manuscript, we formulated and derived new schemes for the incompressible Navier–Stokes and Boussinesq equations combining the novel HDG Method, a projection method, and IMEX-RK time-integration schemes. For the spatial discretization we mathematically derived the proper forms of the element-local corrections and HDG edge-space corrections for both velocity and pressure, including the HDG rotational correction. We also found and explained a consistency relation between the HDG stability parameters for the velocity predictor and pressure correction. We detailed how to incorporate the HDG projection method time-split within standard IMEX-RK time-stepping schemes. Next, we addressed numerical implementation issues of our new solution schemes. In particular, we provided a quadrature-free scheme that is consistent with the HDG method. We also developed a selective nodal limiting approach used to stabilize high-order schemes while retaining high-order accuracy in regions where the solution is smooth.

To verify the quadrature-free approach, we performed a convergence study on a steady diffusion problem using straight-sided and curved meshes. We showed that both of these meshes achieve near-optimal convergence. To verify the selective nodal limiting approach, we completed a convergence study on an unsteady tracer advection problem. We showed that when the selective nodal limiter is applied, the rate of convergence is unaffected for sufficiently resolved solutions: e.g. the rate reduces to second-order for coarse resolutions but the high-order accuracy is retained for fine resolutions.

We performed spatial and temporal convergence studies to verify that our full Navier–Stokes solver based on the new HDG projection method scheme is properly formulated and implemented. We showed that our discretization of the rotational correction term removes pressure errors at the boundary of the domain for the analytical benchmark. We also confirmed that while the splitting error decreases with increasing Reynolds number, it still limits the time-order of accuracy, indicating the need for iterations to achieve higher-order in time. Finally, we evaluated our schemes by comparing our results for a standard lock-exchange benchmark to published literature. We found that our solution closely matched the previous results.

Future opportunities abound for refinement and application of our schemes. We have extended and applied this HDG Projection method to non-hydrostatic physical–biogeochemical ocean equations with a free-surface [24,58,60]. Another research direction is to improve the calculation of bounds for the nodal limiter. Presently the bounds are calculated based on the maximum and minimum values of immediate neighboring nodes. Two potential improvements are to calculate these bounds based on the upwind direction, or based on points sampled between nodes. Finally, our implementation can be further optimized and parallelized to improve efficiency [38], and allow higher resolution required for more realistic and multiscale applications [17,34].

Acknowledgements

We are very thankful to the MSEAS group members, in particular Dr. Patrick Haley Jr. and Dr. Chris Mirabito for many helpful discussions and collaboration. We thank Dr. Ngoc-Cuong Nguyen and Dr. Shirokoff for useful suggestions on HDG and projection methods, and Profs. Flierl, Marshall, Peraire, and Yue for helpful guidance. We are grateful to several agencies for support under the following research grants to the Massachusetts Institute of Technology (MIT): the National Science Foundation for support under the grant OCE-1061160 (ShelfIT); the MIT Sea Grant program and National Oceanic and Atmospheric Administration (Sea Grant) for support under the grant 2013-R/RC-134 (Coastal Bank); and the Office of Naval Research for support under grants N00014-08-1-109 (ONR6.1), N00014-11-1-0701 (MURI-IODA) and N00014-12-1-0944 (ONR6.2). We also thank MIT for awarding MPU with a Pappalardo Fellowship for his first academic year and the Natural Sciences and Engineering Research Council (NSERC) of Canada for partially supporting MPU.

References

[1] T. Ahnert, G. Bärwolff, Numerical comparison of hybridized discontinuous Galerkin and finite volume methods for incompressible flow, Int. J. Numer. Methods Fluids 76 (2014) 267–281.

[2] 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) (2002) 1749–1779.

[3] U. Ascher, S. Ruuth, R. Spiteri, Implicit–explicit Runge–Kutta methods for time-dependent partial differential equations, Appl. Numer. Math. 25 (1997) 151–167.

[4] G.E. Barter, D.L. Darmofal, Shock capturing with higher-order, PDE-based artificial viscosity, AIAA paper 3823:2007, 2007.

[5] G.E. Barter, D.L. Darmofal, Shock capturing with PDE-based artificial viscosity for DGFEM: Part i. formulation, J. Comp. Physiol. 229 (5) (2010) 1810–1827.

[6] P.N. Blossey, D.R. Durran, Selective monotonicity preservation in scalar advection, J. Comput. Phys. 227 (10) (2008) 5160–5183.

[7] A. Chorin, Numerical solution of the Navier–Stokes equations, Math. Comput. 22 (1968) 745–762.

[8] B. Cockburn, J. Gopalakrishnan, The derivation of hybridizable discontinuous Galerkin methods for Stokes flow, SIAM J. Numer. Anal. 47 (2) (2009) 1092–1125, http://dx.doi.org/10.1137/080726653.

[9] B. Cockburn, C.W. Shu, Tvb Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws. II. General framework, Math. Comput. 52 (186) (1989) 411–435.

[10] B. Cockburn, C.W. Shu, The Runge–Kutta discontinuous Galerkin method for conservation laws. V: multidimensional systems, J. Comput. Phys. 141 (2) (1998) 199–224.

[11] B. Cockburn, C.W. Shu, Runge–Kutta discontinuous Galerkin methods for convection-dominated problems, J. Sci. Comput. 16 (3) (2001) 173–261.

[12] B. Cockburn, S.Y. Lin, C.W. Shu, Tvb Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems, J. Comput. Phys. 84 (1) (1989) 90–113.

[13] B. Cockburn, S. Hou, C.W. Shu, The Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws. iv. the multidimensional case, Math. Comput. 54 (190) (1990) 545–581.

[14] B. Cockburn, J. Gopalakrishnan, R. Lazarov, Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems, SIAM J. Numer. Anal. 47 (2) (2009) 1319–1365, http://dx.doi.org/10.1137/070706616.

[15] B. Cockburn, J. Guzmán, H. Wang, Superconvergent discontinuous Galerkin methods for second-order elliptic problems, Math. Comput. 78 (265) (2009) 1–24.

[16] B. Cockburn, J. Gopalakrishnan, N. Nguyen, J. Peraire, F.J. Sayas, Analysis of HDG methods for Stokes flow, Math. Comput. 80 (274) (2011) 723–760.

[17] E. Deleersnijder, V. Legat, P.F.J. Lermusiaux, Multi-scale modelling of coastal, shelf and global ocean dynamics, Ocean Dyn. 60 (6) (2010) 1357–1359, http://dx.doi.org/10.1007/s10236-010-0363-6.

[18] F.M. Denaro, On the application of the Helmholtz–Hodge decomposition in projection methods for incompressible flows with general boundary conditions, Int. J. Numer. Methods Fluids 43 (1) (2003) 43–69.

[19] D.R. Durran, Numerical Methods for Wave Equations in Geophysical Fluid Dynamics, Texts Appl. Math., vol. 32, Springer-Verlag, 1999.

[20] J.H. Ferziger, M. Peric, Computational Methods for Fluid Dynamics, 3rd edn., Springer, New York, NY, 2002.

[21] O. Fringer, M. Gerritsen, R. Street, An unstructured-grid, finite-volume, nonhydrostatic, parallel coastal ocean simulator, Ocean Model. 14 (3) (2006) 139–173.

[22] J. Geiser, Iterative operator-splitting methods with higher-order time integration methods and applications for parabolic partial differential equations, J. Comput. Appl. Math. 217 (1) (2008) 227–242.

[23] J. Guermond, P. Minev, J. Shen, An overview of projection methods for incompressible flows, Comput. Methods Appl. Mech. Eng. 195 (2006) 6011–6045.

[24] P.J. Haley Jr., P.F.J. Lermusiaux, Multiscale two-way embedding schemes for free-surface primitive equations in the “Multidisciplinary Simulation, Estimation and Assimilation System”, Ocean Dyn. 60 (6) (2010) 1497–1537, http://dx.doi.org/10.1007/s10236-010-0349-4.

[25] C. Härtel, E. Meiburg, F. Necker, Analysis and direct numerical simulation of the flow at a gravity-current head. Part 1. Flow topology and front speed for slip and no-slip boundaries, J. Fluid Mech. 418 (2000) 189–212.

[26] J. Hesthaven, R. Kirby, Filtering in Legendre spectral methods, Math. Comput. 77 (263) (2008) 1425–1452.

[27] J. Hesthaven, T. Warburton, Nodal Discontinuous Galerkin Methods, Texts Appl. Math., vol. 54, Springer, New York, NY, 2008.

[28] H. Hoteit, P. Ackerer, R. Mosé, J. Erhel, B. Philippe, New two-dimensional slope limiters for discontinuous Galerkin methods on arbitrary meshes, Int. J. Numer. Methods Eng. 61 (14) (2004) 2566–2593.

[29] A. Huerta, E. Casoni, J. Peraire, A simple shock-capturing technique for high-order discontinuous Galerkin methods, Int. J. Numer. Methods Fluids 69 (10) (2012) 1614–1632.

[30] L. Huynh, N. Nguyen, J. Peraire, B. Khoo, A high-order hybridizable discontinuous Galerkin method for elliptic interface problems, Int. J. Numer. Methods Eng. 93 (2) (2013) 183–200.

[31] C. Kennedy, M. Carpenter, Additive Runge–Kutta schemes for convection–diffusion–reaction equations, Appl. Numer. Math. 44 (2003) 139–181.

[32] R.M. Kirby, S.J. Sherwin, B. Cockburn, To cg or to hdg: a comparative study, J. Sci. Comput. 51 (1) (2012) 183–212.

[33] L. Krivodonova, Limiters for high-order discontinuous Galerkin methods, J. Comput. Phys. 226 (1) (2007) 276–296.

[34] P.F.J. Lermusiaux, J. Schröter, S. Danilov, M. Iskandarani, N. Pinardi, J.J. Westerink, Multiscale modeling of coastal, shelf and global ocean dynamics, Ocean Dyn. 63 (11–12) (2013) 1341–1344, http://dx.doi.org/10.1007/s10236-013-0655-8.

[35] P.F.J. Lermusiaux, M.P. Ueckermann, C. Mirabito, P.J. Haley Jr., J. Aoussou, High order hybridizable discontinuous Galerkin projection schemes for incompressible Navier–Stokes and ocean primitive equations: derivation details, MSEAS Report 20, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, 2014.

[36] C.A. Mavriplis, Nonconforming discretization and a posteriori error estimators for adaptive spectral element techniques, Ph.D. Thesis, MIT, 1989.

[37] C. Michoski, C. Mirabito, C. Dawson, D. Wirasaet, E. Kubatko, J. Westerink, Adaptive hierarchic transformations for dynamically p-enriched slope-limiting over discontinuous Galerkin systems of generalized equations, J. Comput. Phys. 230 (22) (2011) 8028–8056.

[38] C. Mirabito, P.J. Haley Jr., P.F.J. Lermusiaux, Verification and evaluation of hybridizable discontinuous Galerkin schemes for non-hydrostatic ocean modeling using the method of manufactured solutions, 2016, in preparation.

[39] N. Nguyen, P.O. Persson, J. Peraire, Rans solutions using high order discontinuous Galerkin methods, AIAA Paper 914, 2007.

[40] N. Nguyen, J. Peraire, B. Cockburn, An implicit high-order hybridizable discontinuous Galerkin method for linear convection–diffusion equations, J. Comput. Phys. 228 (9) (2009) 3232–3254.

[41] N. Nguyen, J. Peraire, B. Cockburn, An implicit high-order hybridizable discontinuous Galerkin method for nonlinear convection–diffusion equations, J. Comput. Phys. 228 (23) (2009) 8841–8855.

[42] N. Nguyen, J. Peraire, B. Cockburn, A hybridizable discontinuous Galerkin method for Stokes flow, Comput. Methods Appl. Mech. Eng. 199 (9) (2010) 582–597.

[43] N. Nguyen, J. Peraire, B. Cockburn, A hybridizable discontinuous Galerkin method for the incompressible Navier–Stokes equations, in: Proc. of the 48th AIAA Aerospace Sciences Meeting and Exhibit, 2010, Orlando, FL.

[44] N. Nguyen, J. Peraire, B. Cockburn, An implicit high-order hybridizable discontinuous Galerkin method for the incompressible Navier–Stokes equations, J. Comput. Phys. 230 (4) (2011) 1147–1170.

[45] N. Nguyen, X. Roca, D. Moro, J. Peraire, A hybridized multiscale discontinuous Galerkin method for compressible flows, AIAA Paper 689, 2013.

[46] N. Oreskes, K. Shrader-Frechette, K. Belitz, et al., Verification, validation, and confirmation of numerical models in the earth sciences, Science 263 (5147) (1994) 641–646.

[47] M.E. Solano, Hybridizable discontinuous Galerkin method for curved domains, PhD thesis, University of Minnesota, 2012.

[48] J. Peraire, N. Nguyen, B. Cockburn, A hybridizable discontinuous Galerkin method for the compressible Euler and Navier–Stokes equations, AIAA Paper 363:2010, 2010.

[49] J. Peraire, N. Nguyen, B. Cockburn, An embedded discontinuous Galerkin method for the compressible Euler and Navier–Stokes equations, AIAA Paper 3228, 2011.

[50] P.O. Persson, J. Peraire, Sub-cell shock capturing for discontinuous Galerkin methods, AIAA paper 112, 2006.

[51] J. Qiu, C.W. Shu, Runge–Kutta discontinuous Galerkin method using weno limiters, SIAM J. Sci. Comput. 26 (3) (2005) 907–929.

[52] P.J. Roache, Verification and Validation in Computational Science and Engineering, Hermosa, Albuquerque, 1998.

[53] J. Schütz, G. May, A hybrid mixed method for the compressible Navier–Stokes equations, J. Comput. Phys. 240 (2013) 58–75.

[54] G. Strang, G.J. Fix, An Analysis of the Finite Element Method, vol. 212, Prentice Hall, Englewood Cliffs, NJ, 1973.

[55] R. Témam, Sur l'approximation de la solution des équations de Navier–Stokes par la méthode des pas fractionnaires (II), Arch. Ration. Mech. Anal. 33 (1969) 377–385, http://dx.doi.org/10.1007/BF00247696.

[56] L. Timmermans, P. Minev, F. Van De Vosse, An approximate projection scheme for incompressible flow using spectral elements, Int. J. Numer. Methods Fluids 22 (1996) 673–688.

[57] M. Ueckermann, Towards next generation ocean models: novel discontinuous Galerkin schemes for 2D unsteady biogeochemical models, Master's thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering, 2009.

[58] M. Ueckermann, High order hybrid discontinuous Galerkin regional ocean modelling, PhD thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering, 2014.

[59] M.P. Ueckermann, P.F.J. Lermusiaux, High-order schemes for 2D unsteady biogeochemical ocean models, Ocean Dyn. 60 (2010) 1415–1445, http://dx.doi.org/10.1007/s10236-010-0351-x.

[60] M.P. Ueckermann, C. Mirabito, P.J. Haley Jr., P.F.J. Lermusiaux, High order hybridizable discontinuous galerkin projection schemes for non-hydrostatic physical-biogeochemical ocean modeling, 2016, in preparation.

[61] P. Vincent, A. Jameson, Facilitating the adoption of unstructured high-order methods amongst a wider community of fluid dynamicists, Math. Model. Nat. Phenom. 6 (3) (2011) 97–140.

[62] C. Waluga, H. Egger, An implementation of hybrid discontinuous Galerkin methods in dune, in: Advances in DUNE, Springer, 2012, pp. 169–180.

[63] J. Zhu, J. Qiu, C.W. Shu, M. Dumbser, Runge–Kutta discontinuous Galerkin method using weno limiters II: unstructured meshes, J. Comput. Phys. 227 (9) (2008) 4330–4353.