A coupled HDG-FV scheme for the simulation of transient inviscid compressible flows

Sanjay Komala-Sheshachala, Ruben Sevilla[1], Oubay Hassan

Zienkiewicz Centre for Computational Engineering, College of Engineering, Swansea University, Swansea, SA1 8EN, Wales, UK.

Abstract

A methodology that combines the advantages of the vertex-centred finite volume (FV) method and high-order hybridisable discontinuous Galerkin (HDG) method is presented for the simulation of the transient inviscid two dimensional flows. The resulting method is suitable for simulating the transient effects on coarse meshes that are suitable to perform steady simulations with traditional low-order methods. In the vicinity of the aerodynamic shapes, FVs are used whereas in regions where the size of the element is too large for finite volumes to provide an accurate answer, the high-order HDG approach is employed with a non-uniform degree of approximation. The proposed method circumvents the need to produce tailored meshes for transient simulations, as required in a low-order context, and also the need to produce high-order curvilinear meshes, as required by high-order methods. Numerical examples are used to test the optimal convergence properties of the combined HDG-FV scheme and to demonstrate its potential in the context of simulating the wind gust effect on aerodynamic shapes.

---

[1]Corresponding author: r.sevilla@swansea.ac.uk

1. Introduction

Second-order finite volume (FV) methods are still the predominant technique for industrial computational fluid dynamic (CFD) applications due to their robustness and efficiency [1, 2]. Both vertex-centred and cell-centred FV methods form the basis of many industrial and research codes and they have proved to be extremely competitive when simulating steady flows [3, 4, 5]. However, the need for simulating transient high Reynolds number flows poses a major challenge for low-order methods, due to the excessively large number of degrees of freedom required to accurately capture all the flow features. The meshes used for the simulation of steady state have been automated and are designed to capture the required aerodynamic forces. However, these meshes lack the ability to resolve the unsteady features and this results in high dissipation and dispersion errors if utilised with low-order methods. To address this limitation, additional meshes that are refined along the path of all unsteady features have to be generated. For an aircraft configuration, it is estimated that an order of magnitude increase of the mesh size will be required to ensure an adequate unsteady solution with the traditional FV methods. This drawback has motivated the development of different FV schemes with lower dissipation and dispersion errors [6] as well as a number of high-order FV extensions [7, 8].

Other high-order methods have gained popularity due to their ability to accurately capture transient effects with minimum dissipation and dispersion.

2

Among these approaches, high-order stabilised finite elements [9] and discontinuous Galerkin (DG) methods [10] have shown potential, not only by reducing the number of degrees of freedom, but also the computational time to achieve a required accuracy [11, 12, 13, 14, 15, 16]. A particular DG method, called hybridisable DG (HDG) method [17, 18, 19, 20], has recently become popular due to the ability to produce accurate solutions with a reduced number of degrees of freedom when compared to other high-order methods [21, 22]. The application and performance of HDG for CFD applications have been studied by many authors, see for instance [23, 24, 25, 26, 27, 28].

When simulating the flow around aerodynamic shapes, it is of major importance not only to accurately represent the solution but also the geometric description of the boundary of the computational domain [29, 30, 31, 32, 33, 34]. In fact, a low fidelity description of the geometry is known to have a major impact on the solution accuracy [35, 36, 37, 38] and, in some cases, to degrade or even prevent convergence to the correct solution [30, 39, 40, 41]. In this context, the use of curvilinear elements is mandatory to ensure that the full potential of high-order methods is obtained. This has led to a significant effort by the mesh generation community to produce fast and robust arbitrary order curvilinear mesh generation algorithms [42, 43, 44, 45, 46, 47, 48]. Although some approaches are nowadays available, the generation of high-quality meshes for complex aerodynamic shapes, in particular to resolve the highly anisotropic boundary layer region, is still a major challenge [49, 50, 51].

This work proposes a novel scheme that combines the advantages of both second-order vertex-centred FVs and high-order HDG, to enable the computation of wind gust effects on two dimensional aerodynamic shapes using the

3

same meshes employed to simulate steady flows within a FV framework. The proposed approach partitions the mesh in a subdomain where the element size is small enough for a second-order FV to provide the desired accuracy and a subdomain where the use of high-order methods is required to ensure an acceptable accuracy capturing the transient flow features in coarse elements. A high-order HDG method is employed in the coarser elements, with different degree of approximation in different elements according to their characteristic element size. The use of vertex-centred FVs in the vicinity of the aerodynamic shape ensures a minimum number of degrees of freedom when compared to cell-centred or face-centred FVs [52] and avoids the need for generating a high-order curvilinear mesh. Similarly, the use of HDG guarantees a lower number of degrees of freedom compared to other DG methods. It is worth noting that combining low and high-order FV schemes is also an alternative. This option was not considered here due to the extra difficulty associated to the design of the large stencils in unstructured meshes to guarantee high-order convergence.

The coupling between both techniques is performed by introducing a set of transmission conditions between the FV and HDG subdomains to weakly impose the continuity of the solution and the normal flux, thus ensuring conservation. The time marching is performed using classical backward differentiation formulae (BDF) and the resulting non-linear problem is fully linearised using a Newton-Raphson algorithm, leading to an unconditionally stable method.

It is worth noting that the coupling of FVs with other high-order methods was also utilised in [53], where FVs are coupled to high-order finite differences

4

for capturing trailing vortex structures and in [54], where FVs are coupled to DG methods for addressing discontinuities. The coupling of HDG with other techniques has also been proposed. For instance, in [55] HDG is coupled with a boundary element method and in [56, 57] HDG is coupled to standard finite elements in the framework of second-order elliptic problems.

The structure of the paper is as follows. Section 2 briefly recalls the compressible Euler equations and the strategy to introduce gust using a source term. The proposed coupling is described in detail in Section 3. This includes the derivation of the weak formulation, with particular emphasis on the interface coupling. In Section 4 the temporal and spatial discretisation are detailed for both the FV and HDG schemes and the resulting coupled system of non-linear equations is obtained. The linearisation using Newton-Raphson is briefly outlined. Section 5 presents numerical studies to verify the implementation and the optimal approximation properties of the proposed scheme. In Section 6 two examples involving the simulation of a time-harmonic gust impinging on a wing and on a wing-tail configuration are presented. The advantages of the proposed scheme with respect to the classical FV method are analysed. Finally, Section 7 presents the conclusions of the work that has been presented.

2. Governing equations for the simulation of gust in an inviscid compressible fluid

2.1. Euler equations

The Euler equations of gas dynamics express the conservation of mass, linear momentum and total energy of a compressible inviscid fluid. The

strong form of these conservation laws can be expressed as

$$\mathcal{U}_t + \boldsymbol{\nabla}\cdot\boldsymbol{F}(\mathcal{U}) = \boldsymbol{S} \qquad \text{in } \Omega \times (0, T] \tag{1a}$$

$$\mathcal{U} = \mathcal{U}_0 \qquad \text{in } \Omega \times \{0\} \tag{1b}$$

$$\boldsymbol{B}(\mathcal{U},\mathcal{U}^\infty) = \boldsymbol{0} \qquad \text{on } \partial\Omega \times (0, T] \tag{1c}$$

where $\mathcal{U}$ denotes the vector of conservation variables, the tensor $\boldsymbol{F}$ contains the hyperbolic flux vector for each spatial dimension x_l ($l = 1, \ldots, \mathtt{n_{sd}}$), $\mathtt{n_{sd}}$ is the number of spatial dimensions, T is the final time, $\mathcal{U}_0$ denotes the initial condition, $\boldsymbol{B}$ is a generic flux used to define the boundary conditions over the inflow, outflow and wall boundaries and $\mathcal{U}^\infty$ denotes the free stream state. The vector $\mathcal{U}$ and tensor $\boldsymbol{F}$ are given by

$$\mathcal{U} = \begin{Bmatrix} \rho \\ \rho\boldsymbol{v} \\ \rho E \end{Bmatrix}, \qquad \boldsymbol{F} = \begin{bmatrix} \boldsymbol{F}_1 & \ldots & \boldsymbol{F}_{\mathtt{n_{sd}}} \end{bmatrix} = \begin{bmatrix} \rho\,\boldsymbol{v}^T \\ \rho\boldsymbol{v} \otimes \boldsymbol{v} + p\mathbf{I}_{\mathtt{n_{sd}}} \\ (\rho E + p)\boldsymbol{v}^T \end{bmatrix}, \tag{2}$$

where ρ is the density, $\rho\boldsymbol{v}$ is the momentum, ρE is the total energy per unit volume, p is the pressure and $\mathbf{I}_{\mathtt{n_{sd}}}$ is the identity matrix of dimension $\mathtt{n_{sd}}$.

The source term $\boldsymbol{S}$ in Equation (1) usually accounts for the external volume forces. To simplify the presentation, here it only accounts for the generated gust and it is described in the next Section.

The system of non-linear hyperbolic equations is closed with an equation of state, which for a perfect polytropic gas, is

$$p = (\gamma - 1)\rho \left(E - \frac{1}{2}\|\boldsymbol{v}\|^2 \right), \tag{3}$$

where γ is the specific heat at constant pressure over specific heat at constant volume, with value $\gamma = 1.4$ for air.

6

A usual quantity for postprocess of inviscid flow computations is the Mach number, defined as

$$M = \frac{\|\boldsymbol{v}\|}{c},$$

where $c = \sqrt{\gamma p / \rho}$ is the speed of sound.

2.2. Gust modelling

The harmonic perturbation of the velocity field is introduced via a source term that only affects the momentum equation [58]. Contrary to other approaches where the gust is introduced as a time-dependent boundary condition, the approach followed here does not require a fine mesh to capture the propagation of the gust from the far field until it impinges the aerodynamic shape.

In two dimensions, the source is written as

$$\boldsymbol{S}(\boldsymbol{x}, t) = \left\{ 0, \quad S_1, \quad S_2, \quad 0 \right\}^T \tag{4}$$

with

$$S_1(\boldsymbol{x}, t) = \beta K g(x_1) \lambda(x_2) \cos\left(\omega t - \beta x_2 - \alpha x_1^g\right), \tag{5a}$$

$$S_2(\boldsymbol{x}, t) = K g'(x_1) \lambda(x_2) \sin\left(\omega t - \beta x_2 - \alpha x_1^g\right), \tag{5b}$$

where $(x_1^g, 0)$ being the centre of the rectangle of dimension $a \times b$ where the gust is generated, $\alpha = \omega / v_\infty$ and $\beta = \alpha \tan \theta$ represent the wave numbers of the sinusoidal gust in the horizontal and vertical direction respectively, with θ being the angle of propagation of the gust front with respect to the x_1 axis and v_∞ the magnitude of the free-stream velocity. The constant K is defined as

$$K = \epsilon \frac{\alpha \left(\alpha^2 - \hat{a}^2\right) v_\infty^2}{\hat{a}^2 \sqrt{\alpha^2 + \beta^2} \sin\left(\frac{\omega \pi}{\hat{a} v_\infty}\right)}$$

7

where ϵ denotes the gust intensity relative to the mean flow and $\hat{a}$ is used to define the width of the rectangle where the gust is generated, namely $a = 2\pi/\hat{a}$. Finally, the functions λ and g are given by

$$\lambda(x_2) = \frac{1}{2}\Big(\tanh\big(3(x_2 + b/2)\big) - \tanh\big(3(x_2 - b/2)\big)\Big)$$

and

$$g(x_1) = \begin{cases} \frac{1}{2}\big(1 + \cos(\hat{a}(x_1 - x_1^g))\big) & \text{if} \quad |x_1 - x_1^g| \le \frac{a}{2} \\ 0 & \text{otherwise} \end{cases}$$

and they are used to guarantee a smooth transition of the flow field in the boundary of the gust region.

3. HDG-FV weak formulation

Let us consider a two dimensional open bounded domain Ω with boundary $\partial\Omega$ partitioned in two disjoint subdomains Ω^1 and Ω^2 such that $\overline{\Omega} = \overline{\Omega}^1 \cup \overline{\Omega}^2$ with common interface $\Gamma^I = \overline{\Omega}^1 \cap \overline{\Omega}^2$ as shown in Figure 1(a).

This work considers an HDG discretisation in Ω^1 and a vertex-centred finite volume discretisation in Ω^2. To this end, the subdomain Ω^1 is partitioned in $\mathtt{n_{el}}$ disjoint elements Ω_e^1 such that

$$\overline{\Omega}^1 = \bigcup_{e=1}^{\mathtt{n_{el}}} \overline{\Omega}_e^1, \tag{6}$$

with boundaries $\partial\Omega_e^1$, which define an internal interface Γ^1

$$\Gamma^1 := \Big[\bigcup_{e=1}^{\mathtt{n_{el}}} \partial\Omega_e^1 \Big] \setminus \partial\Omega^1. \tag{7}$$

The subdomain Ω^2 is partitioned in $\mathtt{n_{cv}}$ disjoint control volumes Ω_i^2 such that

$$\overline{\Omega}^2 = \bigcup_{i=1}^{\mathtt{n_{cv}}} \overline{\Omega}_i^2. \tag{8}$$
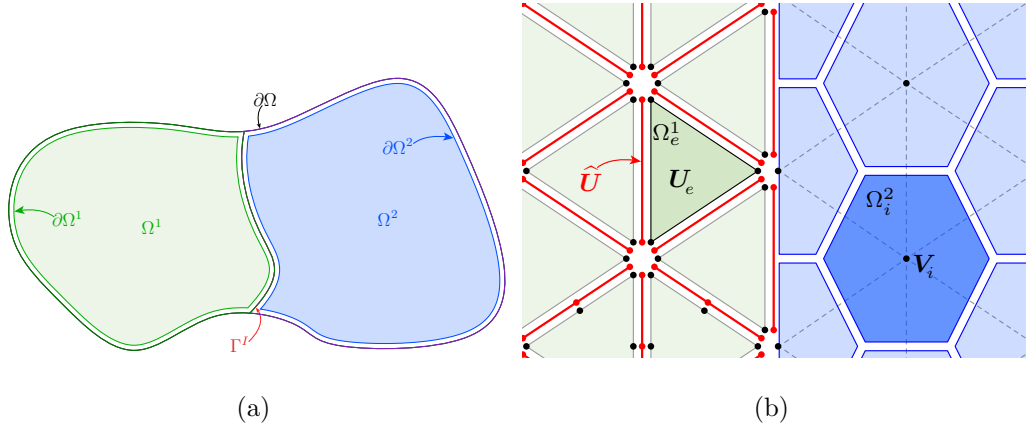
8

Figure 1: (a) Partition of the domain into the HDG and FV subdomains and (b) partition of the HDG subdomain in elements and the FV subdomain in dual cells.

As usual in a finite volume context, the control volume Ω_i^2, associated with the node $\boldsymbol{x}_i$ of Ω^2, is constructed by joining the edge midpoints and the element centroids of the edges connected to node $\boldsymbol{x}_i$ and the elements sharing $\boldsymbol{x}_i$ respectively.

The partition of each subdomain into elements and control volumes respectively is shown in Figure 1(b) near the interface between the HDG and FV subdomains.

3.1. Governing equations

The strong form of the Euler equations in the partitioned domain is written as

$$\boldsymbol{U}_t + \boldsymbol{\nabla} \cdot \boldsymbol{F}(\boldsymbol{U}) = \boldsymbol{S} \qquad \text{in } \Omega_e^1 \times (0, T] \tag{9a}$$

$$[\![\boldsymbol{U} \otimes \boldsymbol{n}]\!] = \boldsymbol{0} \qquad \text{on } \Gamma^1 \times (0, T] \tag{9b}$$

$$[\![\boldsymbol{F}(\boldsymbol{U}) \cdot \boldsymbol{n}]\!] = \boldsymbol{0} \qquad \text{on } \Gamma^1 \times (0, T] \tag{9c}$$

$$\boldsymbol{V}_t + \boldsymbol{\nabla} \cdot \boldsymbol{F}(\boldsymbol{V}) = \boldsymbol{S} \qquad \text{in } \Omega^2 \times (0, T] \tag{9d}$$

$$[\![\boldsymbol{\mathcal{U}} \otimes \boldsymbol{n}]\!] = \boldsymbol{0} \qquad \text{on } \Gamma^I \times (0, T] \tag{9e}$$

$$[\![\boldsymbol{F}(\boldsymbol{\mathcal{U}}) \cdot \boldsymbol{n}]\!] = \boldsymbol{0} \qquad \text{on } \Gamma^I \times (0, T] \tag{9f}$$

$$\boldsymbol{U} = \boldsymbol{\mathcal{U}}_0 \qquad \text{in } \Omega_e^1 \times \{0\} \tag{9g}$$

$$\boldsymbol{V} = \boldsymbol{\mathcal{U}}_0 \qquad \text{in } \Omega^2 \times \{0\} \tag{9h}$$

$$\boldsymbol{B}(\boldsymbol{U}, \boldsymbol{\mathcal{U}}^\infty) = \boldsymbol{0} \qquad \text{on } \left(\partial\Omega_e^1 \cap \partial\Omega\right) \times (0, T] \tag{9i}$$

$$\boldsymbol{B}(\boldsymbol{V}, \boldsymbol{\mathcal{U}}^\infty) = \boldsymbol{0} \qquad \text{on } \left(\partial\Omega^2 \cap \partial\Omega\right) \times (0, T] \tag{9j}$$

for $e = 1, \ldots, \mathtt{n_{el}}$, where $\boldsymbol{U} = \boldsymbol{\mathcal{U}}|_{\Omega^1}$, $\boldsymbol{V} = \boldsymbol{\mathcal{U}}|_{\Omega^2}$, Equations (9b) and (9c) are the transmission conditions introduced in the mesh skeleton of the HDG subdomain Ω^1 [20, 59, 60] and Equations (9e) and (9f) impose the continuity of the solution and the normal fluxes between the two subdomains Ω^1 and Ω^2, respectively. The jump operator is defined at the interface between two elements, Ω_i and Ω_j, as

$$[\![\odot]\!] = \odot|_{\Omega_i} + \odot|_{\Omega_j} \tag{10}$$

and always involving the normal vector $\boldsymbol{n}$, see [61] for more details.

The HDG method introduces the trace of the solution $\boldsymbol{U}$, called hybrid variable and denoted by $\widehat{\boldsymbol{U}}$, as an independent variable leading to the final

10

strong form

$$U_t + \boldsymbol{\nabla}\cdot\boldsymbol{F}(\boldsymbol{U}) = \boldsymbol{S} \qquad \text{in } \Omega_e^1 \times (0,T] \tag{11a}$$

$$\boldsymbol{U} = \widehat{\boldsymbol{U}} \qquad \text{on } \partial\Omega_e^1 \times (0,T] \tag{11b}$$

$$[\![\boldsymbol{F}(\boldsymbol{U})\cdot\boldsymbol{n}]\!] = \boldsymbol{0} \qquad \text{on } \Gamma^1 \times (0,T] \tag{11c}$$

$$\boldsymbol{V}_t + \boldsymbol{\nabla}\cdot\boldsymbol{F}(\boldsymbol{V}) = \boldsymbol{S} \qquad \text{in } \Omega^2 \times (0,T] \tag{11d}$$

$$[\![\boldsymbol{\mathcal{U}}\otimes\boldsymbol{n}]\!] = \boldsymbol{0} \qquad \text{on } \Gamma^I \times (0,T] \tag{11e}$$

$$[\![\boldsymbol{F}(\boldsymbol{\mathcal{U}})\cdot\boldsymbol{n}]\!] = \boldsymbol{0} \qquad \text{on } \Gamma^I \times (0,T] \tag{11f}$$

$$\boldsymbol{U} = \boldsymbol{\mathcal{U}}_0 \qquad \text{in } \Omega_e^1 \times \{0\} \tag{11g}$$

$$\boldsymbol{V} = \boldsymbol{\mathcal{U}}_0 \qquad \text{in } \Omega^2 \times \{0\} \tag{11h}$$

$$\widehat{\boldsymbol{B}}(\boldsymbol{U},\widehat{\boldsymbol{U}},\boldsymbol{\mathcal{U}}^\infty) = \boldsymbol{0} \qquad \text{on } \left(\partial\Omega_e^1 \cap \partial\Omega\right) \times (0,T] \tag{11i}$$

$$\boldsymbol{B}(\boldsymbol{V},\boldsymbol{\mathcal{U}}^\infty) = \boldsymbol{0} \qquad \text{on } \left(\partial\Omega^2 \cap \partial\Omega\right) \times (0,T] \tag{11j}$$

for $e = 1,\ldots,\mathtt{n_{el}}$, where $[\![\odot]\!] = \widehat{\odot} + \odot|_{\Omega^2}$. It is worth noting that Equation (9b) does not feature in the final strong form given by Equation (11) due to the Dirichlet boundary condition of Equation (11b) and the unique definition of the hybrid variable $\widehat{\boldsymbol{U}}$ on each face of Ω^1.

Remark 1. *The strong form of Equation (11) assumes that the hybrid variable $\widehat{\boldsymbol{U}}$ is defined on the interface between the HDG and FV subdomains, Γ^I. This approach weakly enforces the continuity of the solution at Γ^I using the hybrid variable, that is replacing Equation (9e) by Equation (11e). It is also possible to produce a slightly different formulation where the hybrid variable is not defined on Γ^I and the continuity of the solution is weakly imposed directly using the primal variable on the interface (i.e. Equation (9e)). The approach considered here minimises the changes required in an existing HDG*

solver at the expense of introducing the hybrid variable as an extra unknown on the interface Γ^I. It is worth emphasising that this represents a minimum overhead due to the small number of degrees of freedom corresponding to the hybrid variable in the interface compared to the global number of degrees of freedom in the HDG and FV subdomains.

As usual in HDG methods [20, 59, 60], the problem is solved in two stages. The so-called local problem, given by Equations (11a)-(11b) is used to write the primal solution $\boldsymbol{U}$ as a function of the hybrid variable $\widehat{\boldsymbol{U}}$. Introducing this expression into the so-called global problem, given by Equation (11c), leads to a problem only for the hybrid variable.

3.2. Weak formulation

Following the notation in [62], the following discrete functional spaces are introduced

$$\mathcal{W}^{h,k_e}(\Omega_e) := \left\{ v \in \mathcal{L}_2(\Omega) : v|_{\Omega_e} \in \mathcal{P}^{k_e}(\Omega_e) \, \forall \Omega_e \, , \, e = 1, \ldots, \mathtt{n_{el}} \right\}, \quad (12a)$$

$$\widehat{\mathcal{W}}^{h,k_i}(\Gamma_j) := \left\{ \hat{v} \in \mathcal{L}_2(\Gamma \cup \partial\Omega) : \hat{v}|_{\Gamma_j} \in \mathcal{P}^{k_j}(\Gamma_j) \, \forall \Gamma_j \subset \Gamma \cup \partial\Omega \right\}, \quad (12b)$$

$$\mathcal{W}_t^{h,k_e}(\Omega_e) := \left\{ v : v(\cdot,t) \in \mathcal{W}^{h,k_e}(\Omega_e), \forall t \in (0,T] \right\}, \quad (12c)$$

$$\widehat{\mathcal{W}}_t^{h,k_i}(\Gamma_j) := \left\{ \hat{v} : \hat{v}(\cdot,t) \in \widehat{\mathcal{W}}^{h,k_i}(\Gamma_j), \forall t \in (0,T] \right\}, \quad (12d)$$

where $\mathcal{P}^{k_e}(\Omega_e)$ and $\mathcal{P}^{k_j}(\Gamma_j)$ are the spaces of polynomial functions of complete degree at most k_e in Ω_e and k_j on Γ_j respectively. It is worth emphasising that the current implementation of the HDG method allows to use different degree of approximation for the solution in different elements/faces. This capability has been previously exploited to devise degree adaptive procedures within an HDG framework [25, 38].

For each element in the HDG domain, Ω_e^1, $e = 1, \ldots, \mathtt{n_{el}}$, the weak formulation of the HDG local problem, given by Equations (11a)-(11b), is: given $\widehat{\boldsymbol{U}}$ on $\Gamma^1 \cup \partial\Omega^1$, find $\boldsymbol{U} \in [\mathcal{W}_t^{h,k_e}(\Omega_e^1)]^{\mathtt{m_{sd}}}$ that satisfies

$$\left(\delta\boldsymbol{U}, \boldsymbol{U}_t\right)_{\Omega_e^1} - \left(\boldsymbol{\nabla}\delta\boldsymbol{U}, \boldsymbol{F}(\boldsymbol{U})\right)_{\Omega_e^1} + \left\langle\delta\boldsymbol{U}, \widehat{\boldsymbol{F}}(\boldsymbol{U}, \widehat{\boldsymbol{U}}) \cdot \boldsymbol{n}\right\rangle_{\partial\Omega_e^1} = \left(\delta\boldsymbol{U}, \boldsymbol{S}\right)_{\Omega_e^1}, \quad (13)$$

for all $\delta\boldsymbol{U} \in [\mathcal{W}^{h,k_e}(\Omega_e^1)]^{\mathtt{m_{sd}}}$, where $\mathtt{m_{sd}} = \mathtt{n_{sd}} + 2$ is the number of components of $\boldsymbol{U}$, $(\cdot, \cdot)_{\Omega_e}$ and $\langle\cdot, \cdot\rangle_{\partial\Omega_e}$ denote the standard internal products over an element and over an element boundary respectively. The numerical normal flux is defined as

$$\widehat{\boldsymbol{F}}(\boldsymbol{U}, \widehat{\boldsymbol{U}}) \cdot \boldsymbol{n} := \boldsymbol{F}(\widehat{\boldsymbol{U}}) \cdot \boldsymbol{n} + \boldsymbol{\tau} \cdot (\boldsymbol{U} - \widehat{\boldsymbol{U}}). \quad (14)$$

where the stabilisation tensor $\boldsymbol{\tau}$ is selected to ensure the stability, accuracy and convergence properties of the resulting HDG method [20, 59, 60, 63, 64, 23]. Here, the stabilisation tensor is selected as

$$\boldsymbol{\tau}(\boldsymbol{U}, \widehat{\boldsymbol{U}}) = \left(|\hat{v}_n| + \hat{c}\right) \mathbf{I}_{\mathtt{m_{sd}}}. \quad (15)$$

where the normal velocity, $\hat{v}_n = \hat{\boldsymbol{v}} \cdot \boldsymbol{n}$ and $\hat{c}$ are computed from the hybrid variable $\widehat{\boldsymbol{U}}$. This definition of the stabilisation parameter, inspired on the local Lax-Friedrichs method, has been previously used in an HDG context [24]. It is worth noting that other numerical fluxes, e.g. the Roe numerical flux, can be employed but as pointed out in [24], small differences are observed between different Riemann solvers in a high-order context.

Similarly, the weak formulation of the HDG global problem, given by Equations (11c) and (11i) is: find $\widehat{\boldsymbol{U}} \in [\widehat{\mathcal{W}}_t^{h,k_i}(\Gamma^1 \cup \partial\Omega^1)]^{\mathtt{m_{sd}}}$ such that

$$\sum_{e=1}^{\mathtt{n_{el}}} \left\{ \left\langle\delta\widehat{\boldsymbol{U}}, \widehat{\boldsymbol{F}}(\boldsymbol{U}, \widehat{\boldsymbol{U}}) \cdot \boldsymbol{n}\right\rangle_{\partial\Omega_e^1 \backslash \partial\Omega} + \left\langle\delta\widehat{\boldsymbol{U}}, \widehat{\boldsymbol{B}}(\boldsymbol{U}, \widehat{\boldsymbol{U}}, \boldsymbol{\mathcal{U}}^\infty)\right\rangle_{\partial\Omega_e^1 \cap \partial\Omega} \right\} = 0, \quad (16)$$

for all $\delta\widehat{\boldsymbol{U}} \in [\widehat{\mathcal{W}}^{h,k_i}(\Gamma^1 \cup \partial\Omega^1)]^{\mathtt{m}_{\mathtt{sd}}}$.

The weak formulation of the FV problem, given by Equations (11d) and (11j) is: find $\boldsymbol{V} \in [\mathcal{W}_t^{h,0}(\Omega_i^2)]^{\mathtt{m}_{\mathtt{sd}}}$ that satisfies

$$\left(\delta\boldsymbol{V}, \boldsymbol{V}_t\right)_{\Omega_i^2} + \left\langle\delta\boldsymbol{V}, \boldsymbol{F}(\boldsymbol{V})\cdot\boldsymbol{n}\right\rangle_{\partial\Omega_i^2\backslash\partial\Omega} + \left\langle\delta\boldsymbol{V}, \boldsymbol{B}(\boldsymbol{V},\boldsymbol{\mathcal{U}}^\infty)\right\rangle_{\partial\Omega_i^2\cap\partial\Omega} = \left(\delta\boldsymbol{V}, \boldsymbol{S}\right)_{\Omega_i^2}, \tag{17}$$

for all $\delta\boldsymbol{V} \in [\mathcal{W}^{h,0}(\Omega_i^2)]^{\mathtt{m}_{\mathtt{sd}}}$ and for $i = 1, \ldots, \mathtt{n}_{\mathtt{cv}}$.

To account for the transmission conditions at the interface between the HDG and FV subdomains, the continuity of the solution is weakly imposed in the HDG global problem given by Equation (16) and the continuity of the fluxes is imposed in the FV weak formulation given by Equation (17). The weak form of the coupled problem is: find $(\boldsymbol{U}, \widehat{\boldsymbol{U}}, \boldsymbol{V}) \in [\mathcal{W}_t^{h,k_e}(\Omega_e^1)]^{\mathtt{m}_{\mathtt{sd}}} \times [\widehat{\mathcal{W}}_t^{h,k_i}(\Gamma^1 \cup \partial\Omega^1)]^{\mathtt{m}_{\mathtt{sd}}} \times [\mathcal{W}^{h,0}(\Omega_i^2)]^{\mathtt{m}_{\mathtt{sd}}}$ such that

$$\begin{aligned}\left(\delta\boldsymbol{U}, \boldsymbol{U}_t\right)_{\Omega_e^1} &- \left(\boldsymbol{\nabla}\delta\boldsymbol{U}, \boldsymbol{F}(\boldsymbol{U})\right)_{\Omega_e^1} + \left\langle\delta\boldsymbol{U}, \boldsymbol{F}(\widehat{\boldsymbol{U}})\cdot\boldsymbol{n}\right\rangle_{\partial\Omega_e^1} \\ &+ \left\langle\delta\boldsymbol{U}, \boldsymbol{\tau}\cdot\boldsymbol{U}\right\rangle_{\partial\Omega_e^1} - \left\langle\delta\boldsymbol{U}, \boldsymbol{\tau}\cdot\widehat{\boldsymbol{U}}\right\rangle_{\partial\Omega_e^1} = \left(\delta\boldsymbol{U}, \boldsymbol{S}\right)_{\Omega_e^1},\end{aligned} \tag{18a}$$

$$\begin{aligned}\sum_{e=1}^{\mathtt{n}_{\mathtt{el}}} \Big\{ &\left\langle\delta\widehat{\boldsymbol{U}}, \boldsymbol{F}(\widehat{\boldsymbol{U}})\cdot\boldsymbol{n}\right\rangle_{\partial\Omega_e^1\backslash\partial\Omega} + \left\langle\delta\widehat{\boldsymbol{U}}, \boldsymbol{\tau}\cdot\boldsymbol{U}\right\rangle_{\partial\Omega_e^1\backslash\partial\Omega} - \left\langle\delta\widehat{\boldsymbol{U}}, \boldsymbol{\tau}\cdot\widehat{\boldsymbol{U}}\right\rangle_{\partial\Omega_e^1\backslash\partial\Omega^1} \\ &- \left\langle\delta\widehat{\boldsymbol{U}}, \boldsymbol{\tau}\cdot\boldsymbol{V}\right\rangle_{\partial\Omega_e^1\cap\Gamma^I} + \left\langle\delta\widehat{\boldsymbol{U}}, \widehat{\boldsymbol{B}}(\boldsymbol{U},\widehat{\boldsymbol{U}},\boldsymbol{\mathcal{U}}^\infty)\right\rangle_{\partial\Omega_e^1\cap\partial\Omega} \Big\} = 0,\end{aligned} \tag{18b}$$

$$\begin{aligned}\left(\delta\boldsymbol{V}, \boldsymbol{V}_t\right)_{\Omega_i^2} &+ \left\langle\delta\boldsymbol{V}, \boldsymbol{F}(\boldsymbol{V})\cdot\boldsymbol{n}\right\rangle_{\partial\Omega_i^2\backslash\Gamma^I} + \left\langle\delta\boldsymbol{V}, \boldsymbol{F}(\widehat{\boldsymbol{U}})\cdot\boldsymbol{n}\right\rangle_{\partial\Omega_i^2\cap\Gamma^I} \\ &- \left\langle\delta\boldsymbol{V}, \boldsymbol{\tau}\cdot\boldsymbol{U}\right\rangle_{\partial\Omega_i^2\cap\Gamma^I} + \left\langle\delta\boldsymbol{V}, \boldsymbol{\tau}\cdot\widehat{\boldsymbol{U}}\right\rangle_{\partial\Omega_i^2\cap\Gamma^I} \\ &+ \left\langle\delta\boldsymbol{V}, \boldsymbol{B}(\boldsymbol{V},\boldsymbol{\mathcal{U}}^\infty)\right\rangle_{\partial\Omega_i^2\cap\partial\Omega} = \left(\delta\boldsymbol{V}, \boldsymbol{S}\right)_{\Omega_i^2},\end{aligned} \tag{18c}$$

for all $\delta\boldsymbol{U} \in [\mathcal{W}^{h,k_e}(\Omega_e^1)]^{\mathtt{m}_{\mathtt{sd}}}$, $\delta\widehat{\boldsymbol{U}} \in [\widehat{\mathcal{W}}^{h,k_i}(\Gamma^1 \cup \partial\Omega^1)]^{\mathtt{m}_{\mathtt{sd}}}$, $\delta\boldsymbol{V} \in [\mathcal{W}^{h,0}(\Omega_i^2)]^{\mathtt{m}_{\mathtt{sd}}}$ and for $e = 1, \ldots, \mathtt{n}_{\mathtt{el}}$ and $i = 1, \ldots, \mathtt{n}_{\mathtt{cv}}$.

14

4. HDG-FV discretisation

4.1. Temporal discretisation

The time integration is performed using backward differentiation formulae (BDF) [65]. This popular family of implicit multi-step time marching algorithms approximate the first-order time derivative as

$$u_t \approx \sum_{s=0}^{n_s} a_s u^{n+1-s} \tag{19}$$

where $u^r(\boldsymbol{x}) := u(\boldsymbol{x}, t^r)$ and, to shorten the notation, the coefficients a_s include the dependence upon the selected time step Δt.

For steady-state computations, this work employs the first-order BDF method (BDF1), which is equivalent to the backward Euler method, corresponding to $n_s = 1$, $a_0 = 1/\Delta t$ and $a_1 = -1/\Delta t$. For transient computations, a second-order BDF method (BDF2) is employed, corresponding to $n_s = 2$, $a_0 = 3/(2\Delta t)$, $a_1 = -2/\Delta t$ and $a_2 = 1/(2\Delta t)$.

The semi-discrete weak formulation for the proposed HDG-FV with BDF

time integration can be written as

$$
\sum_{s=0}^{n_s} a_s \big(\delta \boldsymbol{U}, \boldsymbol{U}^{n+1-s}\big)_{\Omega_e^1} - \big(\boldsymbol{\nabla}\delta\boldsymbol{U}, \boldsymbol{F}(\boldsymbol{U}^{n+1})\big)_{\Omega_e^1} + \big\langle \delta\boldsymbol{U}, \boldsymbol{F}(\widehat{\boldsymbol{U}}^{n+1})\cdot\boldsymbol{n}\big\rangle_{\partial\Omega_e^1}
$$

$$
+ \big\langle \delta\boldsymbol{U}, \boldsymbol{\tau}^{n+1}\cdot\boldsymbol{U}^{n+1}\big\rangle_{\partial\Omega_e^1} - \big\langle \delta\boldsymbol{U}, \boldsymbol{\tau}^{n+1}\cdot\widehat{\boldsymbol{U}}^{n+1}\big\rangle_{\partial\Omega_e^1} - \big(\delta\boldsymbol{U}, \boldsymbol{S}^{n+1}\big)_{\Omega_e^1} = 0,
$$

$$(20a)$$

$$
\sum_{e=1}^{n_{el}} \bigg\{ \big\langle \delta\widehat{\boldsymbol{U}}, \boldsymbol{F}(\widehat{\boldsymbol{U}}^{n+1})\cdot\boldsymbol{n}\big\rangle_{\partial\Omega_e^1\backslash\partial\Omega} + \big\langle \delta\widehat{\boldsymbol{U}}, \boldsymbol{\tau}^{n+1}\cdot\boldsymbol{U}^{n+1}\big\rangle_{\partial\Omega_e^1\backslash\partial\Omega}
$$

$$
- \big\langle \delta\widehat{\boldsymbol{U}}, \boldsymbol{\tau}^{n+1}\cdot\widehat{\boldsymbol{U}}^{n+1}\big\rangle_{\partial\Omega_e^1\backslash\partial\Omega^1} - \big\langle \delta\widehat{\boldsymbol{U}}, \boldsymbol{\tau}^{n+1}\cdot\boldsymbol{V}^{n+1}\big\rangle_{\partial\Omega_e^1\cap\Gamma^I}
$$

$$
+ \big\langle \delta\widehat{\boldsymbol{U}}, \widehat{\boldsymbol{B}}(\boldsymbol{U}^{n+1},\widehat{\boldsymbol{U}}^{n+1},\boldsymbol{\mathcal{U}}^{\infty})\big\rangle_{\partial\Omega_e^1\cap\partial\Omega} \bigg\} = 0, \qquad\qquad (20b)
$$

$$
\sum_{s=0}^{n_s} a_s \big(\delta\boldsymbol{V}, \boldsymbol{V}^{n+1-s}\big)_{\Omega_i^2} + \big\langle \delta\boldsymbol{V}, \boldsymbol{F}(\boldsymbol{V}^{n+1})\cdot\boldsymbol{n}\big\rangle_{\partial\Omega_i^2\backslash\Gamma^I} + \big\langle \delta\boldsymbol{V}, \boldsymbol{F}(\widehat{\boldsymbol{U}}^{n+1})\cdot\boldsymbol{n}\big\rangle_{\partial\Omega_i^2\cap\Gamma^I}
$$

$$
- \big\langle \delta\boldsymbol{V}, \boldsymbol{\tau}^{n+1}\cdot\boldsymbol{U}^{n+1}\big\rangle_{\partial\Omega_i^2\cap\Gamma^I} + \big\langle \delta\boldsymbol{V}, \boldsymbol{\tau}^{n+1}\cdot\widehat{\boldsymbol{U}}^{n+1}\big\rangle_{\partial\Omega_i^2\cap\Gamma^I}
$$

$$
+ \big\langle \delta\boldsymbol{V}, \boldsymbol{B}(\boldsymbol{V}^{n+1},\boldsymbol{\mathcal{U}}^{\infty})\big\rangle_{\partial\Omega_i^2\cap\partial\Omega} - \big(\delta\boldsymbol{V}, \boldsymbol{S}^{n+1}\big)_{\Omega_i^2} = 0, \qquad (20c)
$$

4.2. Spatial discretisation

The spatial discretisation of the semi-discrete system given by Equation (20) is considered next.

4.2.1. HDG spatial discretisation

For the HDG subdomain Ω^1, an arbitrary order approximation of the solution $\boldsymbol{U}$, using Lagrange polynomials, is defined on a reference element, with local coordinates $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_{n_{sd}})$, namely

$$
\boldsymbol{U}(\boldsymbol{\xi},t) \approx \boldsymbol{U}_h(\boldsymbol{\xi},t) := \sum_{J=1}^{n_{en}} N_J(\boldsymbol{\xi})\mathbf{U}_J(t), \qquad (21)
$$

where n_{en} is the total number of element nodes, $u_J(t)$ are the time-dependent nodal values of the solution and N_J are the shape functions of degree k.

16

Analogously, the hybrid variable $\widehat{\boldsymbol{U}}$ is approximated on a reference face, with local coordinates $\widehat{\boldsymbol{\xi}} = (\widehat{\xi}_1, \ldots, \widehat{\xi}_{\mathbf{n}_{sd}-1})$, as

$$\widehat{\boldsymbol{U}}(\widehat{\boldsymbol{\xi}}, t) \approx \widehat{\boldsymbol{U}}_h(\widehat{\boldsymbol{\xi}}, t) := \sum_{J=1}^{\mathbf{n}_{fn}} \widehat{N}_J(\widehat{\boldsymbol{\xi}}) \widehat{\mathbf{U}}_J(t), \tag{22}$$

where $\mathbf{n}_{fn}$ is the total number of face nodes,

A standard isoparametric formulation [66] is considered, where local and Cartesian coordinates are linked via the so-called isoparametric mapping

$$\boldsymbol{x}(\boldsymbol{\xi}) = \sum_{J=1}^{\mathbf{n}_{en}} N_J(\boldsymbol{\xi}) \boldsymbol{x}_J, \tag{23}$$

where $\{\boldsymbol{x}_J\}_{J=1,\ldots,\mathbf{n}_{en}}$ are the nodal coordinates of a generic element Ω_e^1.

4.2.2. FV spatial discretisation

The vertex-centred finite volume method employs a constant approximation of the solution in each control volume

$$\boldsymbol{V}(\boldsymbol{x}, t) \approx \boldsymbol{V}_h(\boldsymbol{x}, t) := \begin{cases} \mathbf{V}_J(t) & \text{if } \boldsymbol{x} \in \Omega_J^2 \\ 0 & \text{otherwise} \end{cases}. \tag{24}$$

When computing the integral of the normal fluxes over the boundary of the control volume, a linear reconstruction is considered, as classically done in a FV framework, to ensure second-order convergence of the method.

4.2.3. Discrete system

The approximations for the solution in the HDG and FV subdomains, given by Equations (21) and (24) respectively, and the approximation of the HDG hybrid variable given by Equation (22) are introduced in the semi-discrete system of Equation (20). Selecting the spaces of weighting functions

17

as the space spanned by the shape functions, leads to the non-linear system of equations

$$\mathcal{R}(\boldsymbol{U}_h^{n+1}, \ldots, \boldsymbol{U}_h^{n+1-\mathbf{n_s}}, \widehat{\boldsymbol{U}}^{n+1}, \boldsymbol{V}_h^{n+1}, \ldots, \boldsymbol{V}_h^{n+1-\mathbf{n_s}}) = \boldsymbol{0}, \qquad (25)$$

where the global residual of the coupled HDG-FV problem is obtained by assembling the contributions from the HDG global and local problems and the FV problem, namely

$$\mathcal{R}^{e,i} := \left\{ \begin{array}{l} \mathcal{R}^e(\boldsymbol{U}_h^{n+1}, \ldots, \boldsymbol{U}_h^{n+1-\mathbf{n_s}}, \widehat{\boldsymbol{U}}_h^{n+1}) \\ \widehat{\mathcal{R}}^e(\boldsymbol{U}_h^{n+1}, \ldots, \boldsymbol{U}_h^{n+1-\mathbf{n_s}}, \widehat{\boldsymbol{U}}_h^{n+1}, \boldsymbol{V}_h^{n+1}, \ldots, \boldsymbol{V}_h^{n+1-\mathbf{n_s}}) \\ \mathcal{R}^i(\boldsymbol{U}_h^{n+1}, \ldots, \boldsymbol{U}_h^{n+1-\mathbf{n_s}}, \widehat{\boldsymbol{U}}_h^{n+1}, \boldsymbol{V}_h^{n+1}, \ldots, \boldsymbol{V}_h^{n+1-\mathbf{n_s}}) \end{array} \right\} \qquad (26)$$

with

$$\mathcal{R}_I^e := \sum_{s=0}^{\mathbf{n_s}} a_s \left(N_I, \boldsymbol{U}_h^{n+1-s} \right)_{\Omega_e^1} - \left(\boldsymbol{F}(\boldsymbol{U}_h^{n+1}), \boldsymbol{\nabla}^T N_I \right)_{\Omega_e^1} + \left\langle N_I, \boldsymbol{F}(\widehat{\boldsymbol{U}}_h^{n+1}) \cdot \boldsymbol{n} \right\rangle_{\partial \Omega_e^1}$$

$$+ \left\langle N_I, \boldsymbol{\tau}^{n+1} \cdot \boldsymbol{U}_h^{n+1} \right\rangle_{\partial \Omega_e^1} - \left\langle N_I, \boldsymbol{\tau}^{n+1} \cdot \widehat{\boldsymbol{U}}_h^{n+1} \right\rangle_{\partial \Omega_e^1} - \left(N_I, \boldsymbol{S}^{n+1} \right)_{\Omega_e^1} = 0,$$

$$(27a)$$

$$\widehat{\mathcal{R}}_I^e := \sum_{e=1}^{\mathbf{n_{el}}} \left\{ \left\langle \widehat{N}_I, \boldsymbol{F}(\widehat{\boldsymbol{U}}_h^{n+1}) \cdot \boldsymbol{n} \right\rangle_{\partial \Omega_e^1 \backslash \partial \Omega} + \left\langle \widehat{N}_I, \boldsymbol{\tau}^{n+1} \cdot \boldsymbol{U}_h^{n+1} \right\rangle_{\partial \Omega_e^1 \backslash \partial \Omega} \right.$$

$$- \left\langle \widehat{N}_I, \boldsymbol{\tau}^{n+1} \cdot \widehat{\boldsymbol{U}}_h^{n+1} \right\rangle_{\partial \Omega_e^1 \backslash \partial \Omega^1} - \left\langle \widehat{N}_I, \boldsymbol{\tau}^{n+1} \cdot \boldsymbol{V}_h^{n+1} \right\rangle_{\partial \Omega_e^1 \cap \Gamma^I}$$

$$+ \left\langle \widehat{N}_I, \widehat{\boldsymbol{B}}(\boldsymbol{U}_h^{n+1}, \widehat{\boldsymbol{U}}_h^{n+1}, \boldsymbol{\mathcal{U}}^\infty) \right\rangle_{\partial \Omega_e^1 \cap \partial \Omega} \right\} = 0, \qquad (27b)$$

$$\mathcal{R}^i := \sum_{s=0}^{\mathbf{n_s}} a_s \left(1, \boldsymbol{V}_h^{n+1-s} \right)_{\Omega_i^2} + \left\langle 1, \boldsymbol{F}(\boldsymbol{V}_h^{n+1}) \cdot \boldsymbol{n} \right\rangle_{\partial \Omega_i^2 \backslash \Gamma^I} + \left\langle 1, \boldsymbol{F}(\widehat{\boldsymbol{U}}_h^{n+1}) \cdot \boldsymbol{n} \right\rangle_{\partial \Omega_i^2 \cap \Gamma^I}$$

$$- \left\langle 1, \boldsymbol{\tau}^{n+1} \cdot \boldsymbol{U}_h^{n+1} \right\rangle_{\partial \Omega_i^2 \cap \Gamma^I} + \left\langle 1, \boldsymbol{\tau}^{n+1} \cdot \widehat{\boldsymbol{U}}_h^{n+1} \right\rangle_{\partial \Omega_i^2 \cap \Gamma^I}$$

$$+ \left\langle 1, \boldsymbol{B}(\boldsymbol{V}_h^{n+1}, \boldsymbol{\mathcal{U}}^\infty) \right\rangle_{\partial \Omega_i^2 \cap \partial \Omega} - \left(1, \boldsymbol{S}^{n+1} \right)_{\Omega_i^2} = 0, \qquad (27c)$$

for $e = 1, \ldots, \mathbf{n_{el}}$ and $i = 1, \ldots, \mathbf{n_{cv}}$.

18

The FV formulation considered here is stabilised using the so-called Jameson–Schmidt–Turkel scheme. In addition, a shock capturing term based on the traditional pressure switch has been added. The details of both the stabilisation and the shock-capturing can be found in many FV publications [67, 68]. For the HDG approach, the stabilisation is controlled by the stabilisation tensor $\boldsymbol{\tau}$ defined in Equation (15). The shock capturing implemented in the proposed approach follows the traditional sub-cell shock capturing approach introduced for the first time in a DG context in [69].

4.3. Solution strategy

The Newton-Raphson method is applied to linearise the non-linear residual of Equation (25) and, by truncating the Taylor expansion at first order, the non-symmetric sparse linear system to be solved at each iteration (m) of the Newton-Raphson is obtained, namely

$$\begin{bmatrix} \mathbf{T}_{uu} & \mathbf{T}_{u\hat{u}} & \mathbf{0} \\ \mathbf{T}_{\hat{u}u} & \mathbf{T}_{\hat{u}\hat{u}} & \mathbf{T}_{\hat{u}v} \\ \mathbf{T}_{vu} & \mathbf{T}_{v\hat{u}} & \mathbf{T}_{vv} \end{bmatrix}^{n+1,m} \begin{Bmatrix} \Delta\mathbf{U} \\ \Delta\widehat{\mathbf{U}} \\ \Delta\mathbf{V} \end{Bmatrix}^{n+1,m} = \begin{Bmatrix} \mathbf{f}_u \\ \mathbf{f}_{\hat{u}} \\ \mathbf{f}_v \end{Bmatrix}^{n+1,m}, \tag{28}$$

where $\Delta\odot^{n+1,m} = \odot^{n+1,m+1} - \odot^{n+1,m}$ denote the increment of the vector of degrees of freedom. The detailed expression of the tangent matrices $\mathbf{T}_{uu}$, $\mathbf{T}_{u\hat{u}}$, $\mathbf{T}_{\hat{u}u}$, $\mathbf{T}_{\hat{u}\hat{u}}$, $\mathbf{T}_{\hat{u}v}$, $\mathbf{T}_{vu}$, $\mathbf{T}_{v\hat{u}}$ and $\mathbf{T}_{vv}$ and the right hand side vectors $\mathbf{f}_u$, $\mathbf{f}_{\hat{u}}$ and $\mathbf{f}_v$ result from the linearisation using a Newton Raphson method, that is

19

$$(\mathbf{T}_{uu})_{IJ}^{e,n+1,m} := \frac{\partial \boldsymbol{\mathcal{R}}_I^e}{\partial \mathbf{U}_J^{n+1,m}}, \qquad (\mathbf{T}_{u\hat{u}})_{IJ}^{e,n+1,m} := \frac{\partial \boldsymbol{\mathcal{R}}_I^e}{\partial \widehat{\mathbf{U}}_J^{n+1,m}}, \qquad \text{(29a)}$$

$$(\mathbf{T}_{\hat{u}u})_{IJ}^{e,n+1,m} := \frac{\partial \widehat{\boldsymbol{\mathcal{R}}}_I^e}{\partial \mathbf{U}_J^{n+1,m}}, \qquad (\mathbf{T}_{\hat{u}\hat{u}})_{IJ}^{e,n+1,m} := \frac{\partial \widehat{\boldsymbol{\mathcal{R}}}_I^e}{\partial \widehat{\mathbf{U}}_J^{n+1,m}}, \qquad \text{(29b)}$$

$$(\mathbf{T}_{\hat{u}v})_{IJ}^{e,n+1,m} := \frac{\partial \widehat{\boldsymbol{\mathcal{R}}}_I^e}{\partial \mathbf{V}_J^{n+1,m}}, \qquad (\mathbf{T}_{vu})_{J}^{i,n+1,m} := \frac{\partial \boldsymbol{\mathcal{R}}^i}{\partial \mathbf{U}_J^{n+1,m}}, \qquad \text{(29c)}$$

$$(\mathbf{T}_{v\hat{u}})_{J}^{i,n+1,m} := \frac{\partial \boldsymbol{\mathcal{R}}^i}{\partial \widehat{\mathbf{U}}_J^{n+1,m}}, \qquad (\mathbf{T}_{vv})_{J}^{i,n+1,m} := \frac{\partial \boldsymbol{\mathcal{R}}^i}{\partial \mathbf{V}_J^{n+1,m}}, \qquad \text{(29d)}$$

and

$$(\mathbf{f}_u^e)_I^{n+1,m} := -\boldsymbol{\mathcal{R}}_I^e, \qquad (\mathbf{f}_{\hat{u}}^e)_I^{n+1,m} := -\widehat{\boldsymbol{\mathcal{R}}}_I^e, \qquad \left(\mathbf{f}_v^i\right)^{n+1,m} := -\boldsymbol{\mathcal{R}}_v^i. \qquad \text{(30a)}$$

It is worth noting that the tangent matrix $\mathbf{T}_{uu}$ has an element by element block diagonal structure that can be used to obtain a reduced system of equations

$$\begin{bmatrix} \widetilde{\mathbf{T}}_{\hat{u}\hat{u}} & \mathbf{T}_{\hat{u}v} \\ \widetilde{\mathbf{T}}_{v\hat{u}} & \mathbf{T}_{vv} \end{bmatrix}^{n+1,m} \left\{ \begin{array}{c} \Delta \widehat{\mathbf{U}} \\ \Delta \mathbf{V} \end{array} \right\}^{n+1,m} = \left\{ \begin{array}{c} \tilde{\mathbf{f}}_{\hat{u}} \\ \tilde{\mathbf{f}}_v \end{array} \right\}^{n+1,m}, \qquad \text{(31)}$$

where

$$\widetilde{\mathbf{T}}_{\hat{u}\hat{u}} := \mathbf{T}_{\hat{u}\hat{u}} - \mathbf{T}_{\hat{u}u}\mathbf{T}_{uu}^{-1}\mathbf{T}_{u\hat{u}}, \qquad \widetilde{\mathbf{T}}_{v\hat{u}} := \mathbf{T}_{v\hat{u}} - \mathbf{T}_{vu}\mathbf{T}_{uu}^{-1}\mathbf{T}_{u\hat{u}} \qquad \text{(32a)}$$

$$\tilde{\mathbf{f}}_{\hat{u}} := \mathbf{f}_{\hat{u}} - \mathbf{T}_{\hat{u}u}\mathbf{T}_{uu}^{-1}\mathbf{f}_u, \qquad \tilde{\mathbf{f}}_v := \mathbf{f}_v - \mathbf{T}_{vu}\mathbf{T}_{uu}^{-1}\mathbf{f}_u. \qquad \text{(32b)}$$

In the current implementation, the linear system given by Equation (31) is solved using a multi-frontal method for sparse unsymmetric systems [70, 71] and the solution in the HDG domain is recovered by solving a set of independent local problems in each element, namely

$$\mathbf{T}_{uu}^{n+1,m}\Delta \mathbf{U}^{n+1,m} = \mathbf{f}_u^{n+1,m} - \mathbf{T}_{u\hat{u}}^{n+1,m}\Delta \widehat{\mathbf{U}}^{n+1,m}. \qquad \text{(33)}$$

Remark 2. *As mentioned in Remark 1, the continuity of the solution in the interface Γ^I has been imposed on the global problem, through the hybrid variable $\widehat{\boldsymbol{U}}$. If the hybrid variable is not defined on the interface, the alternative formulation would lead to a tangent matrix of the form*

$$\begin{bmatrix} \mathbf{T}_{uu} & \mathbf{T}_{u\hat{u}} & \mathbf{T}_{uv} \\ \mathbf{T}_{\hat{u}u} & \mathbf{T}_{\hat{u}\hat{u}} & \mathbf{0} \\ \mathbf{T}_{vu} & \mathbf{T}_{v\hat{u}} & \mathbf{T}_{vv} \end{bmatrix}. \tag{34}$$

This alternative formulation leads to a reduced system

$$\begin{bmatrix} \widetilde{\mathbf{T}}_{\hat{u}\hat{u}} & \widetilde{\mathbf{T}}_{\hat{u}v} \\ \widetilde{\mathbf{T}}_{v\hat{u}} & \widetilde{\mathbf{T}}_{vv} \end{bmatrix}^{n+1,m} \left\{ \begin{matrix} \Delta\widehat{\mathbf{U}} \\ \Delta\mathbf{V} \end{matrix} \right\}^{n+1,m} = \left\{ \begin{matrix} \tilde{\mathbf{f}}_{\hat{u}} \\ \tilde{\mathbf{f}}_v \end{matrix} \right\}^{n+1,m}, \tag{35}$$

where

$$\widetilde{\mathbf{T}}_{\hat{u}v} := -\mathbf{T}_{\hat{u}u}\mathbf{T}_{uu}^{-1}\mathbf{T}_{uv}, \qquad \widetilde{\mathbf{T}}_{vv} := \mathbf{T}_{vv} - \mathbf{T}_{vu}\mathbf{T}_{uu}^{-1}\mathbf{T}_{uv} \tag{36}$$

When compared to the system of Equation (31), the system of the alternative formulation given by Equation (35) is slightly smaller but the hybridisation process requires the extra operations detailed in Equation (36) for each time step and each non-linear iteration.

5. Numerical studies

This section presents a set of numerical studies to verify the optimal approximation properties of the proposed HDG-FV scheme. First, the developed FV and HDG schemes are tested separately to verify their optimal convergence both in space and time and the coupled scheme is then assessed and compared to the FV and HDG schemes.
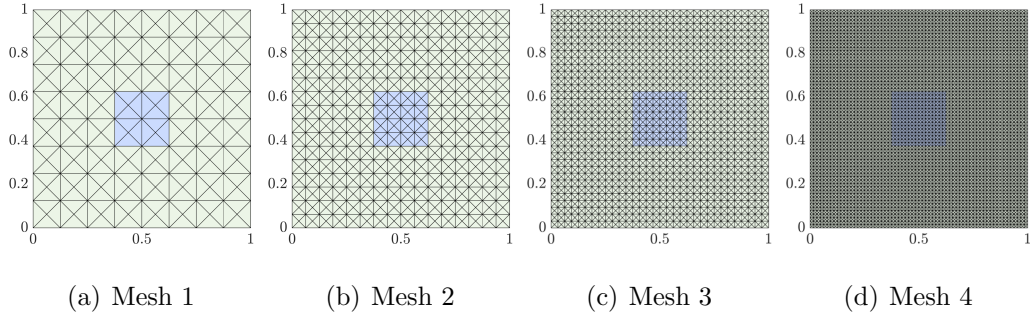
Figure 2: Triangular meshes of the domain $\Omega = [0, 1]^2$ used to test the optimal convergence properties of the FV, HDG and HDG-FV methods.

To test the optimal convergence of the spatial discretisation, the Ringleb flow problem [72] is considered. This classical test case features a steady-state problem that has analytical solution and it is often employed to verify the optimal approximation properties of newly developed spatial discretisation schemes. To test the optimal convergence of the implemented BDF methods, a synthetic transient problem with analytical solution is defined using the method of manufactured solutions. The exact solution is selected to be

$$\boldsymbol{\mathcal{U}} = \left\{ 1, \cos(\pi/6), \sin(\pi/6), 4 + \sin(100t) \right\}^T, \tag{37}$$

and a source term is imposed so that Equation (37) is a solution of the Euler equations.

Four uniform meshes of the domain $\Omega = [0, 1]^2$ are considered to test the spatial convergence, with 256, 1,024, 4,096 and 16,384 triangular elements, respectively, as shown in Figure 2. When the combined scheme is used, the mesh is partitioned as $\Omega^2 = [0.375, 0.625]^2$ and $\Omega^1 = \Omega \setminus \Omega^2$, as illustrated in Figure 2.
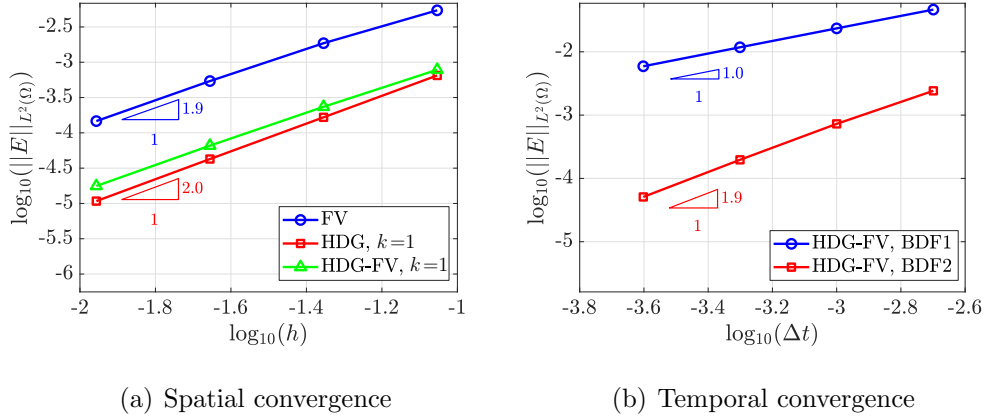
22

(a) Spatial convergence   (b) Temporal convergence
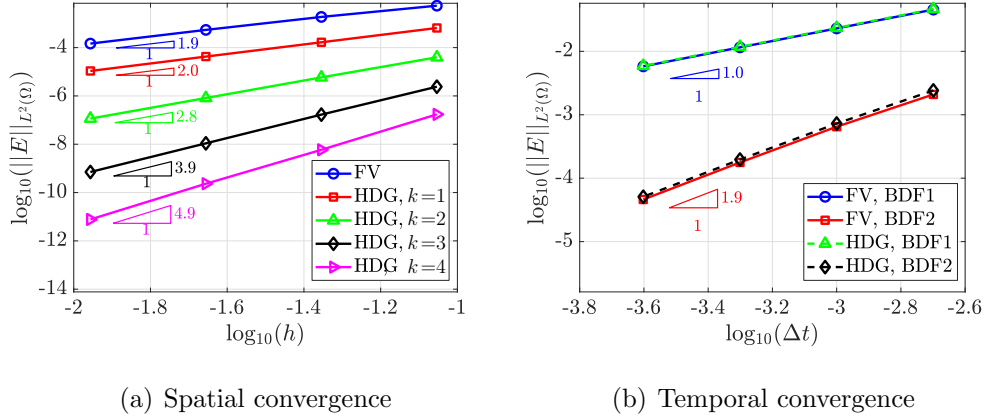
Figure 3: Convergence properties of the FV and HDG methods. (a) Mesh convergence of the error of the density in the $\mathcal{L}_2(\Omega)$ for the Ringleb flow problem and (b) temporal convergence of the error of the density in the $\mathcal{L}_2(\Omega)$ for the manufactured problem.

5.1. Optimal convergence of the FV and HDG methods

First the convergence of the FV and HDG methods under mesh refinement is studied. Figure 3(a) shows the relative error in the $\mathcal{L}_2(\Omega)$ norm as a function of the characteristic element size h. For the HDG scheme a degree of approximation ranging from $k = 1$ up to $k = 4$ is considered.

The results show the expected second-order convergence of the vertex-centred FV scheme and nearly the optimal rate of convergence $k + 1$ for the HDG method. It is worth noting that Figure 3(a) illustrates the extra accuracy of the HDG scheme with linear approximation when compared to the standard second-order FV method.

The temporal convergence for the FV and HDG schemes using BDF1 and BDF2 time integrators is shown in Figure 3(b) for the problem with manufactured solution. The relative error in the $\mathcal{L}_2(\Omega)$ norm is displayed as a function of the time step Δt. The results show the expected first and

23

(a) Spatial convergence   (b) Temporal convergence

Figure 4: Convergence properties of the proposed HDG-FV method. (a) Mesh convergence of the error of the density in the $\mathcal{L}_2(\Omega)$ for the Ringleb flow problem and (b) temporal convergence of the error of the density in the $\mathcal{L}_2(\Omega)$ for the manufactured problem.

second order convergence of the BDF1 and BDF2 methods respectively, for both FV and HDG.

5.2. Optimal convergence of the coupled HDG-FV method

Next, the spatial and temporal convergence properties of the proposed HDG-FV scheme are verified using the same numerical examples described in Section 5.1.

For the spatial convergence, Figure 4(a) shows the relative error in the $\mathcal{L}_2(\Omega)$ norm as a function of the characteristic element size h for the Ringleb flow. The combined scheme shows the expected second order convergence. In addition, the results for both HDG with linear approximation and FV have been added to the figure to enable a visual comparison of the gain of accuracy induced by the HDG formulation when compared to the FV. In this example the accuracy of the combined scheme is almost identical to the

24

accuracy of a standard HDG solver.

Finally, Figure 4(b) shows the relative error in the $\mathcal{L}_2(\Omega)$ norm as a function of the time step Δt for the problem with manufactured solution. The expected order of convergence is observed for both the BDF1 and BDF2 schemes with the combined HDG-FV approach. The accuracy is identical in both cases to the accuracy obtained with HDG or FV. This is expected because for studying the temporal convergence the mesh is selected to be fine enough so that the error is controlled by the temporal integrator.

5.3. Handling shocks across and aligned with the HDG-FV interface

The classical shock tube problem is considered here to show the ability of the proposed coupling strategy to handle shocks that cross and that are aligned with the HDG-FV interface.

The computational domain is $\Omega = [0, 1] \times [0, 0.1]$ and the initial condition is defined as [73]

$$
\boldsymbol{\mathcal{U}}_0 =
\begin{cases}
\{3, 0, 0, 3/(\gamma - 1)\}^T & \text{if } x \leq 0.5 \\
\{1, 0, 0, 1/(\gamma - 1)\}^T & \text{if } x > 0.5.
\end{cases}
$$

The FV subdomain is $\Omega^2 = [0.375, 0.625] \times [0.025, 0.075]$ and the HDG subdomain is $\Omega^1 = \Omega \setminus \Omega^2$. This choice ensures that from $t = 0$ to $t < 0.0685$, the a shock is crossing the HDG-FV interface and at time $t = 0.0685$, the shock is aligned with the HDG-FV interface.

The density, computed with the proposed HDG-FV scheme, at three different instants, is displayed in Figure 5. The thick line denotes the interface between the HDG and FV subdomains. At $t = 0.0225$, the shock is crossing the top and bottom part of the HDG-FV interface and it can be seen that the

25

(a) $t = 0.0225$ (b) $t = 0.0685$

(c) $t = 0.1145$
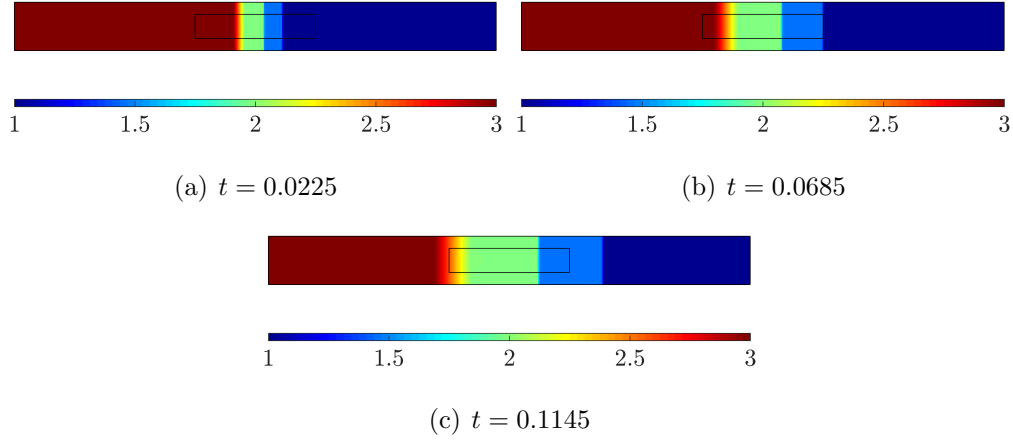
Figure 5: Density at three different instants computed with the proposed HDG-FV scheme showing the ability to handle the shock at different stages.

position of the shock is captured correctly by both schemes, with no artefacts on the interface. At $t = 0.0685$, the shock is perfectly aligned with the interface and again, the solution is captured correctly with no artefacts. Finally, at time $t = 0.1145$ the shock is in the HDG subdomain. No artefacts are present due to the transition between subdomains and all the flow features are well represented.

It is worth mentioning that the focus here is in the ability to handle shocks across the interface between the HDG and FV subdomains and shocks that are perfectly aligned with the interface. In addition, this example shows the ability to also handle contact discontinuities and rarefactions waves across the two subdomains.

6. Numerical examples

Two numerical examples involving the simulation of the gust impinging on aerofoils [74, 75] are considered. The first example is used to show the efficiency of the proposed combined HDG-FV method compared to a traditional FV method and the second example shows the potential and applicability of the proposed method to a more challenging problem involving a two-aerofoil configuration.

6.1. Gust impinging on a NACA0012 aerofoil

The simulation of a sinusoidal gust impinging on a NACA0012 aerofoil immersed in an inviscid subsonic flow is considered. The free stream Mach number is $M_\infty = 0.5$ and the angle of attack is 2 degrees. The problem setup is illustrated in Figure 6, showing the aerofoil of chord length $c = 1$ and the rectangular box of dimension $a \times b$ at a distance d from the aerofoil where the gust is introduced as a source term, as detailed in Section 2.2. The far field boundary is situated at 10 chord lengths from the aerofoil.

First, four unstructured triangular meshes are used to select the level of mesh refinement required to accurately compute the quantities of interest, namely lift and drag, for the steady state solution of the Euler equations. A detailed view of the first three meshes near the aerofoil is depicted in Figure 7. The generated meshes contain 2,295, 7,701, 35,425 and 133,459 elements, respectively, and the aerofoil is discretised with 101, 179, 375 and 725 points in each case.

Figure 8 shows the computed lift as a function of the number of elements. From this study, it can be concluded that the second mesh provides the re-
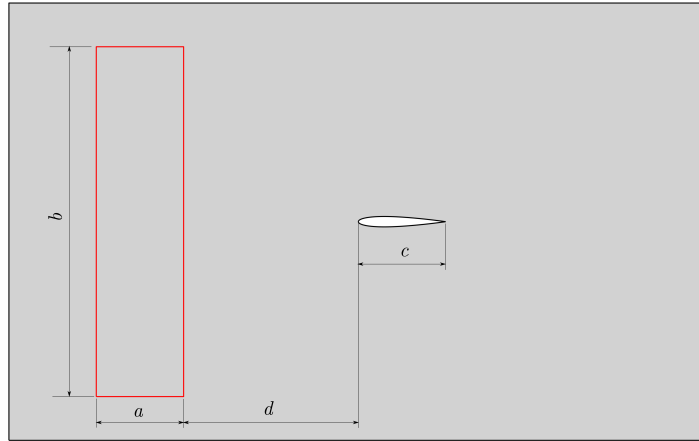
Figure 6: Illustration of the problem setup for the simulation of wind gust impinging on a NACA0012 aerofoil. A sinusoidal gust is generated in the region enclosed by the box of width a and height b, which is located at a distance d upstream to the aerofoil of chord length c.

(a) Mesh 1   (b) Mesh 2   (c) Mesh 3
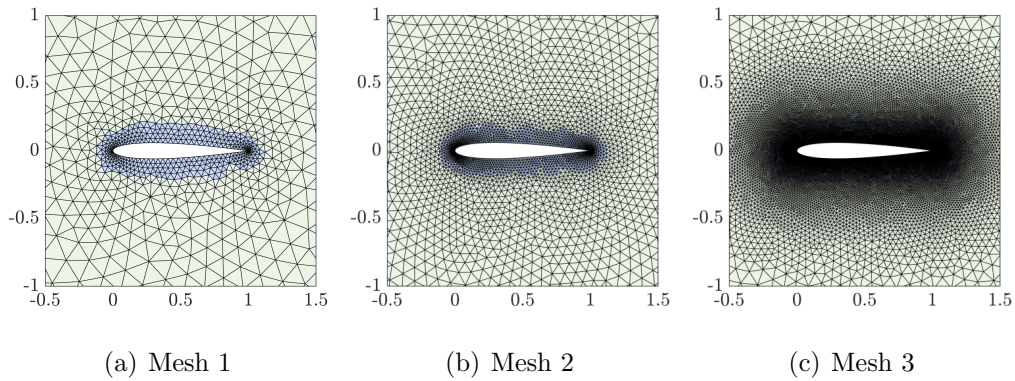
Figure 7: Meshes used in the select the level of mesh refinement required to accurately capture the steady state solution.

quired accuracy as the lift coefficient is within five lift counts of the reference value. A detailed view of the Mach number and pressure distributions near the aerofoil is displayed in Figure 9.
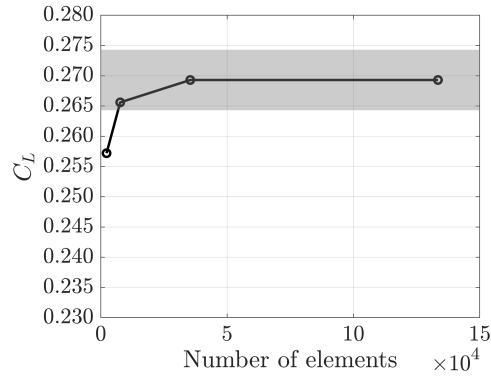
Figure 8: Convergence plot of the lift coefficient C_L as a function of the number of elements. The shaded area represents the region with an error of maximum five lift counts compared to the reference solution.

(a) Mach

(b) Pressure
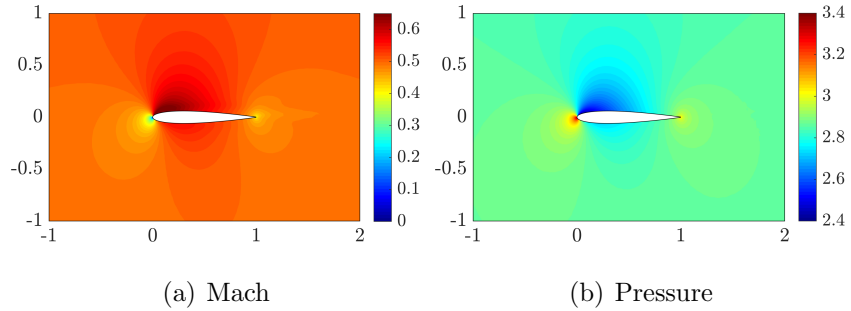
Figure 9: Steady state solution computed on the mesh displayed in Figure 7 (b).

Next, the simulation of the sinusoidal gust impinging in the NACA aerofoil is considered. To speed up the convergence to the time harmonic steady state, the computed steady state solution is used as the initial condition for the transient gust simulation. The non-dimensional parameters of the source term required to introduce the gust are the frequency, $\omega = 4$, the angle of propagation of the gust front, $\theta = 45°$, the gust intensity, $\epsilon = 0.1$, the dimensions of the box where the gust is generated, $a = 1$ and $b = 4$, and the

29

(a) Mesh 1　　　　　　　　(b) Mesh 2　　　　　　　　(c) Mesh 3

(d) Mesh 4　　　　　　　　(e) Mesh 5
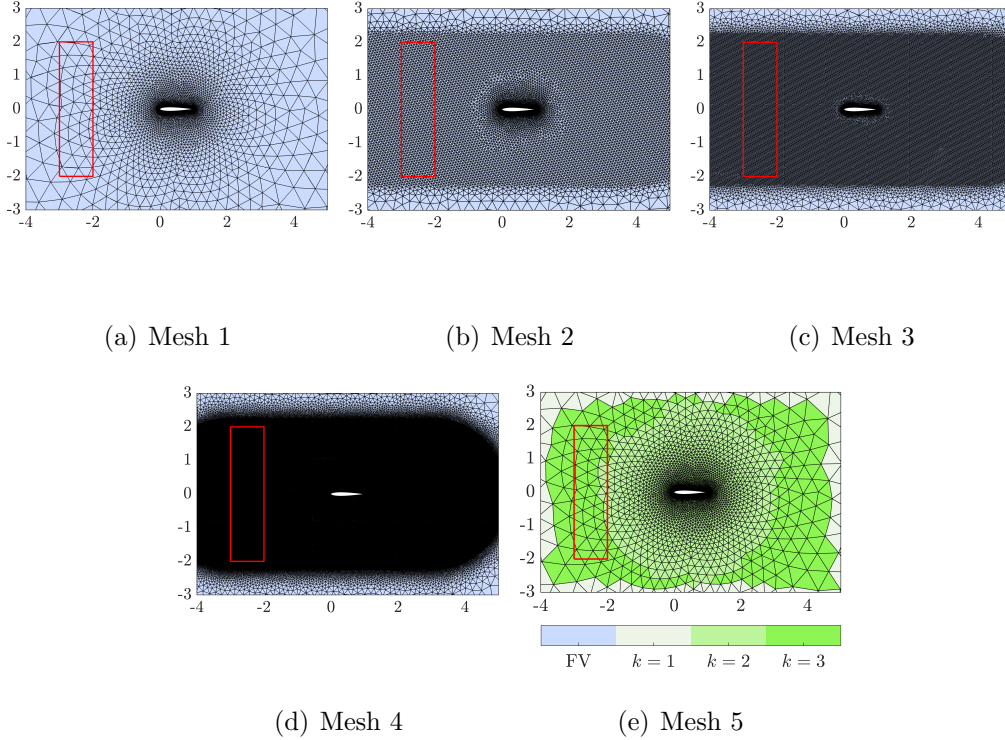
Figure 10: Unstructured triangular meshes employed to simulate the wind gust impinging in a NACA0012 aerofoil.

distance to the aerofoil, $d = 2$.

Five meshes are considered to show the benefits of the proposed HDG-FV approach for capturing the transient gust effect. First, the mesh used to compute the steady state solution, shown in Figure 10 (a), is considered to perform a standard FV simulation. Second, a mesh where the whole region of interest, namely $\widetilde{\Omega} = [-4, 4] \times [-2, 2]$, is refined by using a desired element size equal to $h^\star = 0.08$. The resulting mesh, displayed in Figure 10 (b), has 24,851 elements. The third mesh corresponds to a mesh where the region of interest is refined using a desired element size equal to $h^\star = 0.04$.

30

The resulting mesh, displayed in Figure 10 (c), has 83,229 elements. The fourth mesh corresponds to a mesh where the region of interest is refined using a desired element size equal to $h^\star = 0.02$, which corresponds to the largest element size on the aerofoil used in the steady state simulation.. The resulting mesh, displayed in Figure 10 (d), has 265,237 elements.

Finally, the same mesh considered in the steady state simulation is partitioned in two regions. The FV region Ω_2 consists of elements with size less or equal to $3h^\star/2$, whereas the rest of the mesh defines the HDG region Ω_1. In the HDG region, the degree of the functional approximation employed in elements lying in the region of interest $\widetilde{\Omega}$ is adapted based on the wavelength of the impinging gust, $\lambda = \pi/2$. For elements with size less than or equal to $\lambda/5$, a quadratic approximation is employed, whereas in the remaining elements a cubic approximation is used. For elements outside the region of interest a linear approximation is used. The mesh is depicted in Figure 10 (e), including the degree of approximation used in the different elements.

In all the simulations, 32 time steps per cycle of the gust are considered. This corresponds to a CFL number of approximately 442 in the coarsest mesh and 478 in the finest mesh.

Figure 11 shows the Mach number distribution after the time harmonic steady state has been reached for the three computations using three of the meshes of Figure 10. In all cases the second order BDF2 time integrator described in Section 4.1 is employed.

The solution obtained with the FV method on the mesh of Figure 10 (a) shows, as expected, the large dissipation introduced by the traditional FV scheme on coarse meshes. This experiment confirms that meshes designed for

31

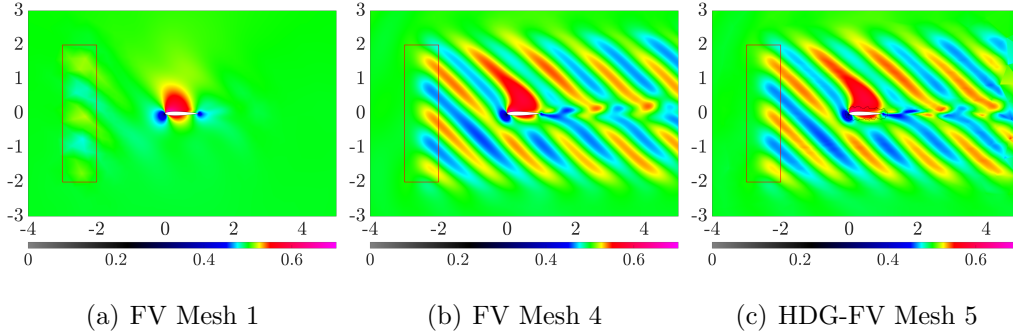(a) FV Mesh 1 (b) FV Mesh 4 (c) HDG-FV Mesh 5

Figure 11: Mach number distribution for the simulation of the wind gust impinging in a NACA0012 aerofoil after the time harmonic steady state is reached.

steady state simulations are not suitable for transient simulations in a low-order framework. Figure 11 (b) shows the FV solution computed on the fine mesh of Figure 10 (d). The solution computed with the proposed HDG-FV scheme on the coarse mesh used for the steady state simulation and with a variable degree of approximation in the HDG region is depicted in Figure 10 (e), showing a good agreement with the reference solution computed with FV in the finest mesh.

To better illustrate the accuracy of the proposed scheme, Figure 12 shows the evolution of the lift coefficient in time for the solutions computed on the five meshes shown in Figure 10. The computation using FV in the finest mesh is taken as the reference solution and the accuracy of the computations, using FV in the first three meshes and the computation using HDG-FV in the coarse mesh with non-uniform degree of approximation, is measured by means of the dissipation and dispersion error.

The dissipation error is estimated by comparing the amplitude of the oscillations in the lift coefficient against the reference results. For the FV
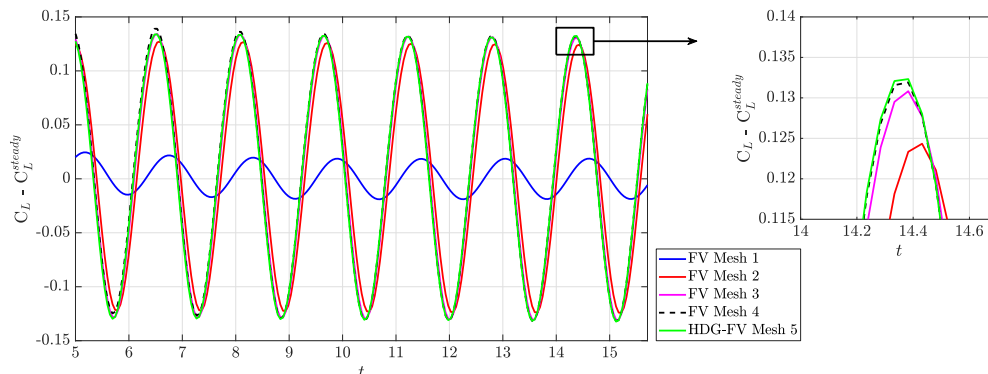
Figure 12: Variation of the lift coefficient with respect to the steady state solution as a function of the non-dimensional time for the simulations computed on the five meshes shown in Figure 10.

simulations, the estimated dissipation error is 85.8%, 5.7% and 0.8% in the first three meshes respectively, whereas the computation with the proposed HDG-FV scheme produces a dissipation error of 0.3%. Similarly, the dispersion error is estimated by comparing the phase of the oscillations in the lift coefficient against the reference value. In this case, the FV computations produce a dispersion error of 57.5°, 13° and 2.5° respectively, whereas the combined HDG-FV approach produces an error below 0.3°.

To further illustrate the benefits of the proposed approach, Figure 13 shows a one dimensional section, at $y = c/2$, of the vertical velocity field for the five simulations computed on the five meshes shown in Figure 10. The results clearly show an excellent agreement between the solution computed using the proposed scheme and the reference solution. Using the coarsest mesh with a FV scheme the flow features are not captured due to an excessive dissipation. The dissipation and dispersion errors when the FV scheme is
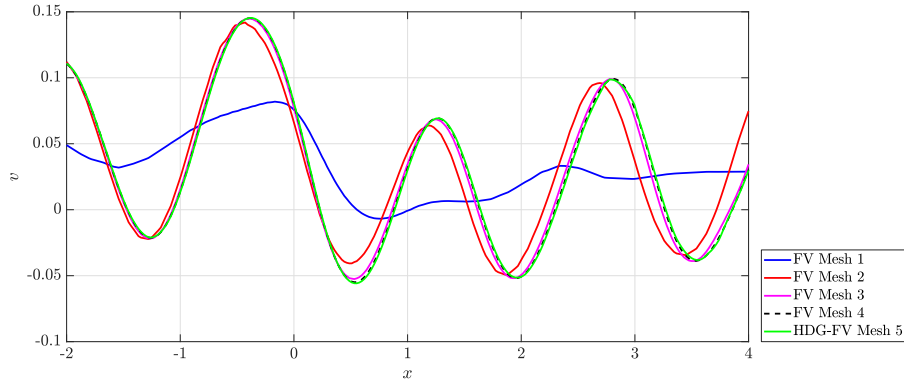
Figure 13: One dimensional section, at $y = c/2$, of the vertical velocity field for the five simulations computed on the five meshes shown in Figure 10.

used in the second mesh are clearly visible, whereas the simulation using FV in the third mesh provides a much better agreement.

From a computational point of view, the simulation using the proposed HDG-FV scheme requires the solution of a linear system of equations with 96,652 degrees of freedom within each Newton-Raphson iteration whereas the solution computed on the reference mesh requires the solution of a linear system of equations with 530,832 degrees of freedom within each Newton-Raphson iteration. It is worth noting that the substantial decrease in number of degrees of freedom also corresponds to a save in computational time. The time required to compute the solution using the proposed HDG-FV approach is almost one order of magnitude lower than using the standard FV method on the fine mesh. Using the FV scheme in the third mesh leads to a linear system of equations with 166,816 degrees of freedom to be solved within each Newton-Raphson iteration. This simulations takes twice the time required by the proposed HDG-FV scheme and, as detailed earlier, produces less accurate

34

results.

It is also worth emphasising that the benefit of the proposed approach, in addition to the save in computational cost, is that it avoids the generation of meshes tailored to specific transient simulations.

6.2. Gust impinging on a two-aerofoil configuration

The simulation of the wind gust effect on two-aerofoil configurations is of major importance as it corresponds to a two dimensional representation of a canard-wing or wing-tail configuration [76]. In this scenario, it is not only important to accurately capture the gust impinging on the first aerofoil but it is also relevant to accurately represent the flow disturbances produced by the first aerofoil that impinge on the second aerofoil.

The simulation of a sinusoidal gust impinging on a wing-tail configuration immersed in an inviscid transonic flow at free stream Mach number $M_\infty = 0.8$ and with angle of attack equal to 4.4844 degrees with respect to the wing is considered. The problem setup is illustrated in Figure 14, showing the aerofoil of chord length c and the rectangular box of dimension $a \times b$ at a distance d from the aerofoil, where the gust is introduced as a source term. As in the previous example, the far field boundary is situated at 10 chord lengths from the aerofoil and the same intensity. The frequency and intensity of the gust and the angle of propagation of the gust front are taken as in the previous example. The dimensions of the box where the gust is generated are $a = 1$ and $b = 4$ and the distance to the aerofoil is $d = 3.04$.

An unstructured triangular mesh with 12,504 elements was generated, with localised mesh refinement around the wing and tail. This mesh, suitable for a steady state simulation is then partitioned in two regions as done in
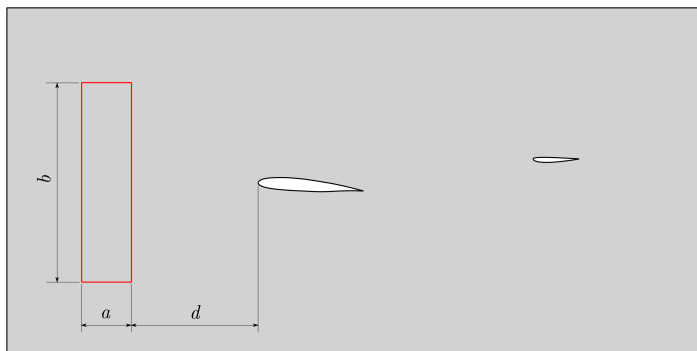
Figure 14: Illustration of the problem setup for the simulation of wind gust impinging on the transverse section of wing and tail configuration. A sinusoidal gust is generated in the region enclosed by the box of width a and height b, which is located at a distance d upstream to the wing.

the previous example. In the region where the elements are small enough to capture the gust perturbation a standard FV scheme is employed whereas in the rest of the domain an HDG approach is used. In the HDG region, the degree of the approximation is adapted following the same strategy as in the previous example. The resulting spatial discretisation, including the degree of approximation used in each element of the HDG region is displayed in Figure 15. Two detailed views of the mesh around the wing and tail are are displayed in order to show the regions where the standard FV scheme is used.

As in the previous example, 32 time steps per cycle of the gust are considered. This corresponds to a CFL number of approximately 56.

The solution after the time harmonic steady state is achieved for both the standard FV scheme and the proposed HDG-FV method are shown in Figure 16. The results illustrate the substantial dissipation introduced by the
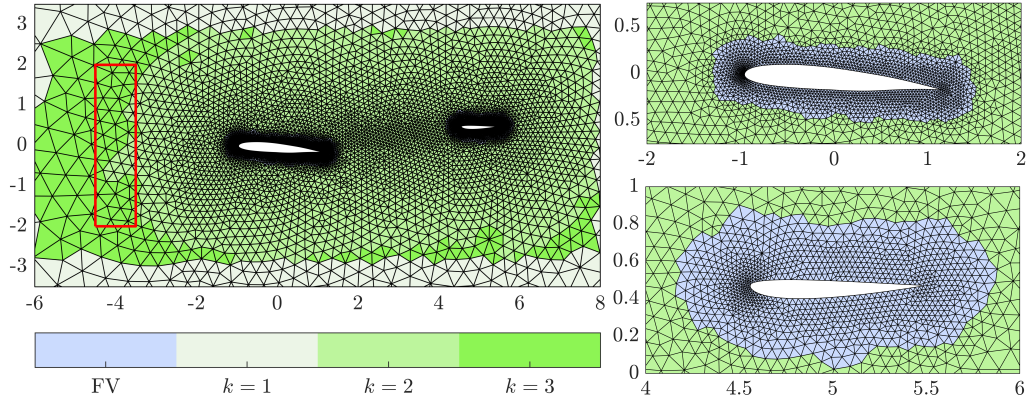
36

Figure 15: Unstructured triangular mesh employed to simulate the wind gust impinging in a wing-tail configuration. The two detailed views around the wing and tail show the partition in HDG and FV regions.
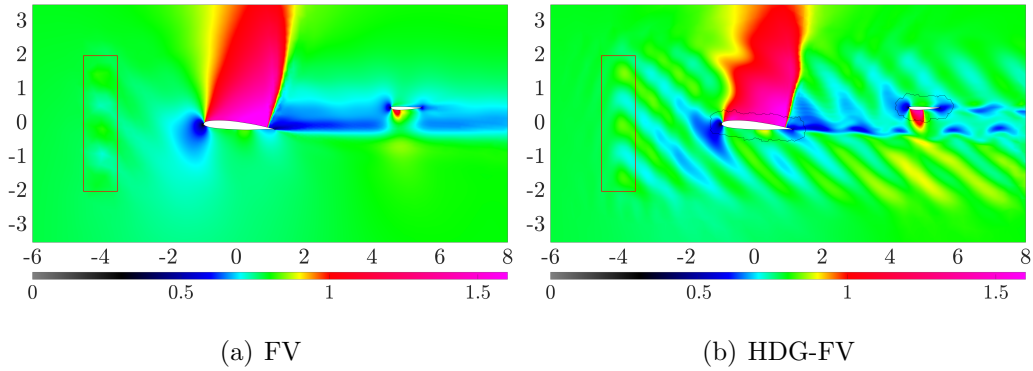

(a) FV

(b) HDG-FV

Figure 16: Mach number distribution for the simulation of the wind gust impinging in a wing-tail configuration after the time harmonic steady state is reached.

FV scheme when the coarse mesh, suitable for a steady state simulation, is used. Instead, the solution with the proposed HDG-scheme is able to capture the perturbation of the velocity induced by the gust not only impinging in the aerofoil but also arriving to the tail and interacting with the strong shocks on both the aerofoil and the tail.

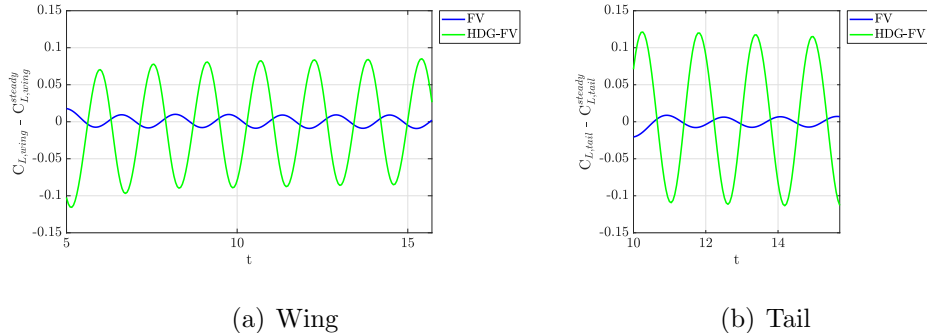(a) Wing   (b) Tail

Figure 17: Variation of the lift coefficient with respect to the steady state solution as a function of the non-dimensional time for the simulations displayed in Figure 16 for the wing and the tail.

To further illustrate the benefits of the proposed approach, Figure 17 shows the evolution of the lift coefficient computed on the aerofoil and tail separately. The results clearly show the dissipation introduced by the FV scheme in coarse meshes and how the proposed scheme is able to capture the amplitude of the oscillations of the lift coefficient on both the aerofoil and the tail without the need to produce a tailored mesh for this application, just re-using the mesh that is generated to perform a steady state simulation.

7. Concluding remarks

A new methodology that combines the advantages of the vertex-centred FV and the HDG methods has been presented and applied to the simulation of the transient wind gust effect on aerodynamic shapes with meshes suitable for steady state simulations.

The method avoids the need to generate high-order curvilinear meshes, required in an HDG context, and to generate meshes tailored for each transient

simulation as required in a FV framework. Instead, the proposed scheme uses low order FV elements in regions where the mesh is fine enough to capture the transient effect and high-order HDG elements in regions where the mesh, suitable for the steady state simulation, is not fine enough to capture the transient effects. Both schemes are coupled through the weak imposition of the continuity of the solution and the normal fluxes. The resulting scheme is fully implicit and linearisation is performed using a Newton Raphson algorithm.

Two numerical test cases have been used to demonstrate the optimal convergence properties of the proposed scheme, both in space and time. Finally, two numerical examples involving the transient simulation of the wind gust effect in an aerofoil and a wing-tail configuration have been presented to show the benefit and potential of the proposed approach. The proposed method is capable of accurately capturing the transient flow effects and reducing the computational cost by an order of magnitude compared to the computation with a standard FV scheme in a sufficiently fine mesh.

Acknowledgements

References

[1] R. J. LeVeque, Finite volume methods for hyperbolic problems, volume 31, Cambridge university press, 2002.

[2] H. K. Versteeg, W. Malalasekera, An introduction to computational fluid dynamics: the finite volume method, Pearson education, 2007.

[3] K. Morgan, J. Peraire, J. Peiro, O. Hassan, The computation of three-dimensional flows using unstructured grids, Computer Methods in Applied Mechanics and Engineering 87 (1991) 335–352.

[4] T. Gerhold, Overview of the hybrid RANS code TAU, in: MEGAFLOW-Numerical Flow Simulation for Aircraft Design, Springer, 2005, pp. 81–92.

[5] R. T. Biedron, J.-R. Carlson, J. M. Derlaga, P. A. Gnoffo, D. P. Hammond, W. T. Jones, B. Kleb, E. M. Lee-Rausch, E. J. Nielsen, M. A. Park, et al., FUN3D Manual: 12.9, Technical Report TM-2016-219012, NASA, 2016.

[6] J. Löwe, A. Probst, T. Knopp, R. Kessler, A low-dissipation low-dispersion second-order scheme for unstructured finite-volume flow solvers, in: 53rd AIAA Aerospace Sciences Meeting, p. 0815.

[7] X. Nogueira, I. Colominas, L. Cueto-Felgueroso, S. Khelladi, On the simulation of wave propagation with a higher-order finite volume scheme based on reproducing kernel methods, Computer Methods in Applied Mechanics and Engineering 199 (2010) 1471–1490.

[8] G. Campagne, O. Hassan, K. Morgan, K. Sørensen, Higher–order aerodynamic computations using an edge based finite volume scheme, in: ADIGMA-A European Initiative on the Development of Adaptive

Higher-Order Variational Methods for Aerospace Applications, Springer, 2010, pp. 309–325.

[9] J. Donea, A. Huerta, Finite Element Methods for Flow Problems, Finite Element Methods for Flow Problems, John Wiley & Sons, 2003.

[10] B. Cockburn, Discontinuous Galerkin methods for computational fluid dynamics, Encyclopedia of Computational Mechanics Second Edition (2018) 1–63.

[11] C. H. Whiting, K. E. Jansen, A stabilized finite element method for the incompressible Navier–Stokes equations using a hierarchical basis, International Journal for Numerical Methods in Fluids 35 (2001) 93–116.

[12] F. Chalot, P.-E. Normand, Higher-order stabilized finite elements in an industrial Navier-Stokes code, in: ADIGMA-A European Initiative on the Development of Adaptive Higher-Order Variational Methods for Aerospace Applications, Springer, 2010, pp. 145–165.

[13] R. Sevilla, O. Hassan, K. Morgan, An analysis of the performance of a high-order stabilised finite element method for simulating compressible flows, Computer Methods in Applied Mechanics and Engineering 253 (2013) 15–27.

[14] R. Glasby, N. Burgess, K. Anderson, L. Wang, S. Allmaras, D. Mavriplis, Comparison of SU/PG and DG finite-element techniques for the compressible Navier-Stokes equations on anisotropic unstructured meshes, in: 51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, p. 691.

[15] N. Kroll, H. Bieler, H. Deconinck, V. Couaillier, H. van der Ven, K. Sorensen, ADIGMA–A European Initiative on the Development of Adaptive Higher-Order Variational Methods for Aerospace Applications: Results of a Collaborative Research Project Funded by the European Union, 2006-2009, volume 113, Springer Science & Business Media, 2010.

[16] N. Kroll, C. Hirsch, F. Bassi, C. Johnston, K. Hillewaert, IDIHOM: Industrialization of High-Order Methods-A Top-Down Approach: Results of a Collaborative Research Project Funded by the European Union, 2010-2014, volume 128, Springer, 2015.

[17] B. Cockburn, J. Gopalakrishnan, A characterization of hybridized mixed methods for second order elliptic problems, SIAM Journal on Numerical Analysis 42 (2004) 283–301.

[18] B. Cockburn, J. Gopalakrishnan, Incompressible finite elements via hybridization. I. The Stokes system in two space dimensions, SIAM Journal on Numerical Analysis 43 (2005) 1627–1650.

[19] B. Cockburn, J. Gopalakrishnan, New hybridization techniques, GAMM-Mitt. 28 (2005) 154–182.

[20] B. Cockburn, J. Gopalakrishnan, R. Lazarov, Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems, SIAM Journal on Numerical Analysis 47 (2009) 1319–1365.

[21] A. Huerta, A. Angeloski, X. Roca, J. Peraire, Efficiency of high-order

elements for continuous and discontinuous Galerkin methods, International Journal for Numerical Methods in Engineering 96 (2013) 529–560.

[22] S. Yakovlev, D. Moxey, R. M. Kirby, S. J. Sherwin, To CG or to HDG: a comparative study in 3d, Journal of Scientific Computing 67 (2016) 192–220.

[23] N. C. Nguyen, J. Peraire, B. Cockburn, An implicit high-order hybridizable discontinuous Galerkin method for the incompressible Navier-Stokes equations, Journal of Computational Physics 230 (2011) 1147–1170.

[24] J. Peraire, N. Nguyen, B. Cockburn, A hybridizable discontinuous Galerkin method for the compressible Euler and Navier-Stokes equations, AIAA paper 363 (2010) 2010.

[25] G. Giorgiani, S. Fernández-Méndez, A. Huerta, Hybridizable discontinuous Galerkin with degree adaptivity for the incompressible Navier-Stokes equations, Computers & Fluids 98 (2014) 196–208.

[26] M. Woopen, G. May, An anisotropic adjoint-based hp-adaptive HDG method for compressible turbulent flow, in: 53rd AIAA Aerospace Sciences Meeting, p. 2042.

[27] P. Fernandez, N. C. Nguyen, J. Peraire, The hybridized discontinuous Galerkin method for implicit large-eddy simulation of transitional turbulent flows, Journal of Computational Physics 336 (2017) 308–329.

[28] S. Rhebergen, G. N. Wells, A hybridizable discontinuous Galerkin

method for the Navier–Stokes equations with pointwise divergence-free velocity field, Journal of Scientific Computing 76 (2018) 1484–1501.

[29] A. Dadone, B. Grossman, Surface boundary conditions for the numerical solution of the Euler equations, AIAA journal 32 (1994) 285–293.

[30] F. Bassi, S. Rebay, High-order accurate discontinuous finite element solution of the 2D Euler equations, Journal of Computational Physics 138 (1997) 251–285.

[31] L. Krivodonova, M. Berger, High-order accurate implementation of solid wall boundary conditions in curved geometries, Journal of computational physics 211 (2006) 492–512.

[32] R. Sevilla, S. Fernández-Méndez, A. Huerta, NURBS-enhanced finite element method (NEFEM) for Euler equations, International Journal for Numerical Methods in Fluids 57 (2008) 1051–1069.

[33] E. Ferrer, R. H. Willden, A high order discontinuous Galerkin–Fourier incompressible 3D Navier–Stokes solver with rotating sliding meshes, Journal of Computational Physics 231 (2012) 7037–7056.

[34] X. Meng, G. Hu, A NURBS-enhanced finite volume solver for steady Euler equations, Journal of Computational Physics 359 (2018) 77–92.

[35] D. Xue, L. Demkowicz, et al., Control of geometry induced error in hp finite element (FE) simulations. I. Evaluation of FE error for curvilinear geometries, International Journal of Numerical Analysis and Modeling 2 (2005) 283–300.

[36] R. Sevilla, S. Fernández−Méndez, A. Huerta, Comparison of high-order curved finite elements, International Journal for Numerical Methods in Engineering 87 (2011) 719–734.

[37] S. Soghrati, R. A. Merel, NURBS enhanced HIFEM: a fully mesh-independent method with zero geometric discretization error, Finite Elements in Analysis and Design 120 (2016) 68–79.

[38] R. Sevilla, A. Huerta, HDG-NEFEM with degree adaptivity for stokes flows, Journal of Scientific Computing 77 (2018) 1953–1980.

[39] R. Sevilla, S. Fernández−Méndez, A. Huerta, NURBS–Enhanced Finite Element Method (NEFEM): a seamless bridge between CAD and FEM, Archives of Computational Methods in Engineering 18 (2011) 441–484.

[40] H. Gao, Z. J. Wang, Y. Liu, A study of curved boundary representations for 2D high order Euler solvers, Journal of Scientific Computing 44 (2010) 323–336.

[41] M. Dawson, R. Sevilla, K. Morgan, The application of a high-order discontinuous Galerkin time-domain method for the computation of electromagnetic resonant modes, Applied Mathematical Modelling 55 (2018) 94–108.

[42] X.-J. Luo, M. S. Shephard, R. M. O'bara, R. Nastasia, M. W. Beall, Automatic p-version mesh generation for curved domains, Engineering with Computers 20 (2004) 273–285.

[43] P.-O. Persson, J. Peraire, Curved mesh generation and mesh refinement

using lagrangian solid mechanics, in: Proceedings of the 47th AIAA Aerospace Sciences Meeting and Exhibit, AIAA.

[44] T. Toulorge, C. Geuzaine, J.-F. Remacle, J. Lambrechts, Robust untangling of curvilinear meshes, Journal of Computational Physics 254 (2013) 8–26.

[45] Z. Q. Xie, R. Sevilla, O. Hassan, K. Morgan, The generation of arbitrary order curved meshes for 3D finite element analysis, Computational Mechanics 51 (2013) 361–374.

[46] A. Gargallo-Peiró, X. Roca, J. Peraire, J. Sarrate, Distortion and quality measures for validating and generating high-order tetrahedral meshes, Engineering with Computers 31 (2015) 423–437.

[47] M. Fortunato, P.-O. Persson, High-order unstructured curved mesh generation using the Winslow equations, Journal of Computational Physics 307 (2016) 1 – 14.

[48] D. Moxey, D. Ekelschot, Ü. Keskin, S. J. Sherwin, J. Peiró, High-order curvilinear meshing using a thermo-elastic analogy, Computer-Aided Design 72 (2016) 130–139.

[49] P. E. Vincent, A. Jameson, Facilitating the adoption of unstructured high-order methods amongst a wider community of fluid dynamicists, Mathematical Modelling of Natural Phenomena 6 (2011) 97–140.

[50] J. R. Chawner, J. Dannenhoffer, S. Dey, W. Jones, J. P. Slotnick, N. J. Taylor, The path to and state of geometry and meshing in 2030: Panel

summary, in: 22nd AIAA Computational Fluid Dynamics Conference, p. 3409.

[51] S. L. Karman, N. Wyman, J. P. Steinbrenner, Mesh generation challenges: A commercial software perspective, in: 23rd AIAA Computational Fluid Dynamics Conference, p. 3790.

[52] R. Sevilla, M. Giacomini, A. Huerta, A face-centred finite volume method for second-order elliptic problems, International Journal for Numerical Methods in Engineering 115 (2018) 986–1014.

[53] D. M. Changfoot, A. G. Malan, J. Nordström, Hybrid computational-fluid-dynamics platform to investigate aircraft trailing vortices, Journal of Aircraft 56 (2018) 344–355.

[54] M. Dumbser, O. Zanotti, R. Loubère, S. Diot, A posteriori subcell limiting of the discontinuous Galerkin finite element method for hyperbolic conservation laws, Journal of Computational Physics 278 (2014) 47–75.

[55] B. Cockburn, F.-J. Sayas, M. Solano, Coupling at a distance HDG and BEM, SIAM Journal on Scientific Computing 34 (2012) A28–A47.

[56] M. Paipuri, C. Tiago, S. Fernández-Méndez, Coupling of continuous and hybridizable discontinuous Galerkin methods: Application to conjugate heat transfer problem, Journal of Scientific Computing 78 (2019) 321–350.

[57] M. G. A. La Spina, A. Huerta, Hybrid coupling of CG and HDG discretizations based on Nitsche's method, Computational Mechanics https://doi.org/10.1007/s00466-019-01770-8 (2019).

[58] V. Golubev, B. Dreyer, T. Hollenshade, M. Visbal, High-accuracy viscous simulation of gust-airfoil nonlinear aeroelastic interaction, in: 39th aiaa fluid dynamics conference, p. 4200.

[59] B. Cockburn, B. Dong, J. Guzmán, A superconvergent LDG-hybridizable Galerkin method for second-order elliptic problems, Mathematics of Computation 77 (2008) 1887–1916.

[60] N. C. Nguyen, J. Peraire, B. Cockburn, An implicit high-order hybridizable discontinuous Galerkin method for linear convection-diffusion equations, Journal of Computational Physics 228 (2009) 3232–3254.

[61] A. Montlaur, S. Fernández-Méndez, A. Huerta, Discontinuous Galerkin methods for the Stokes equations using divergence-free approximations, International Journal for Numerical Methods in Fluids 57 (2008) 1071–1092.

[62] R. Sevilla, A. Huerta, Tutorial on Hybridizable Discontinuous Galerkin (HDG) for second-order elliptic problems, in: J. Schröder, P. Wriggers (Eds.), Advanced Finite Element Technologies, volume 566 of *CISM International Centre for Mechanical Sciences*, Springer International Publishing, 2016, pp. 105–129.

[63] N. C. Nguyen, J. Peraire, B. Cockburn, An implicit high-order hybridizable discontinuous Galerkin method for nonlinear convection-diffusion equations, Journal of Computational Physics 228 (2009) 8841–8855.

[64] N. Nguyen, J. Peraire, B. Cockburn, A hybridizable discontinuous

Galerkin method for Stokes flow, Computer Methods in Applied Mechanics and Engineering 199 (2010) 582–597.

[65] U. M. Ascher, L. R. Petzold, Computer methods for ordinary differential equations and differential-algebraic equations, volume 61, Siam, 1998.

[66] P. Solin, K. Segeth, I. Dolezel, Higher-order finite element methods, Chapman and Hall/CRC, 2003.

[67] A. Jameson, W. Schmidt, E. Turkel, Numerical solution of the Euler equations by finite volume methods using Runge Kutta time stepping schemes, in: 14th fluid and plasma dynamics conference, p. 1259.

[68] K. Sørensen, O. Hassan, K. Morgan, N. Weatherill, A multigrid accelerated time-accurate inviscid compressible fluid flow solution algorithm employing mesh movement and local remeshing, International journal for numerical methods in fluids 43 (2003) 517–536.

[69] P.-O. Persson, J. Peraire, Sub-cell shock capturing for discontinuous Galerkin methods, in: 44th AIAA Aerospace Sciences Meeting and Exhibit, pp. 112–124.

[70] J. W. Liu, The multifrontal method for sparse matrix solution: Theory and practice, SIAM review 34 (1992) 82–109.

[71] HSL, A collection of Fortran codes for large scale scientific computation, http://www.hsl.rl.ac.uk/, 2013.

[72] H. Yoshihara, H. Norstrud, J. Boerstoel, G. Chiocchia, D. Jones, Test Cases for Inviscid Flow Field Methods, Technical Report AGARD-AR-

211, Advisory group for aerospace research and development, France, 1985.

[73] R. J. LeVeque, R. J. Leveque, Numerical methods for conservation laws, volume 132, Springer, 1992.

[74] M. Nallasamy, R. Hixon, S. Sawyer, Solution of unsteady Euler equations: Gust–cascade interaction tones, Computers & Fluids 36 (2007) 724–741.

[75] R. G. Cook, C. Wales, A. Gaitonde, D. Jones, J. E. Cooper, Efficient modelling of a nonlinear gust loads process for uncertainty quantification of highly flexible aircraft, in: 2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, p. 1681.

[76] D. Pan, J.-C. Cheng, Unstructured Euler flutter analysis of two-dimensional wing-tail configuration, Journal of Aircraft 32 (1995) 1152–1155.