Coupling of hybridisable discontinuous Galerkin and finite volumes for transient compressible flows

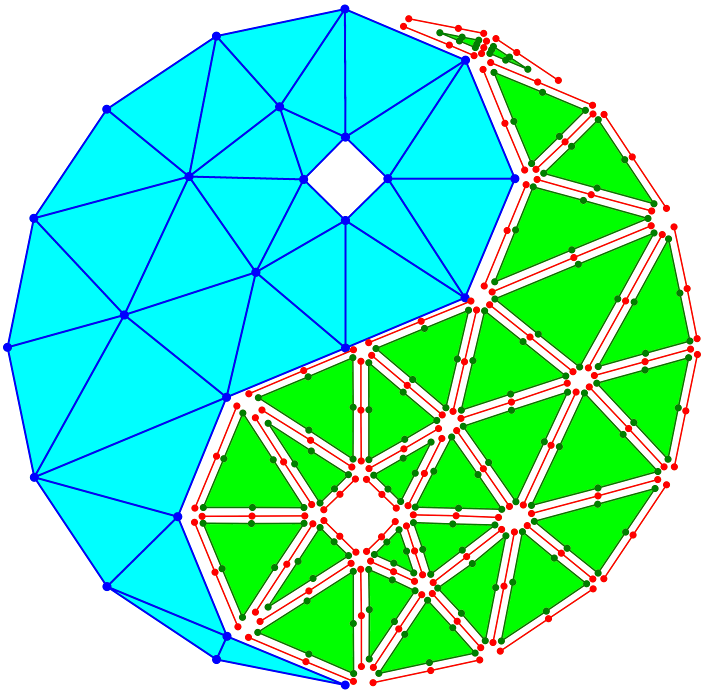
Swansea University
Prifysgol Abertawe

Sanjay Komala Sheshachala

College of Engineering
Swansea University

This dissertation is submitted for the degree of
Doctor of Philosophy

March 2021



“Light is the left hand of darkness
and darkness the right hand of light.”

- Ursula K. Le Guin

Dedicated to *Ajji*

Declaration

Statement 1

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.
Signed: [redacted] (candidate)
Date: 21 December 2020

Statement 2

This thesis is a result of my own independent work/investigation, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.
Signed: [redacted] (candidate)
Date: 21 December 2020

Statement 3

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.
Signed: [redacted] (candidate)
Date: 21 December 2020

Acknowledgements

First and foremost, I would like to express my gratitude to my supervisors, Ruben Sevilla and Oubay Hassan, for their guidance and support throughout the PhD program. They have been highly inspirational figures, I consider myself fortunate to have had the opportunity to learn a great deal of knowledge and insights from them.

I am indebted to all my friends in Swansea for making it a wonderful stay. I shall miss the countless evenings spent with Tan and Emilio; very few experiences were as enriching as the discussions we had over warm and comforting Chinese food. I'm specially thankful to German for his funny yet insightful words of wisdom, Ken for his kindness and compassion, Mai for being the sensitive and kind-hearted person that she is, Nidhal for his unrelenting brotherly love and Quang for his limitless optimism. Life in Swansea was brightened by countless amazing people, many thanks to Davide, Diwali, Gniewomir, Kayal, Leandro, Marianna, Osama and Sunag. It would be remiss of me not to mention Aunty Molly for her unbounded kindness and Dan and his team for their warm cheerfulness and hot falafels.

On the other side of the globe, I count myself blessed to have such a supportive and caring family. I am forever indebted to *Amma* and *Appa* for their love and affection. They were always there to cheer me up after a stressful day at work. A huge shout-out to *Aunty* and *Dodappa*, Anu, Sri, Sush and Sup for being such steadfast supporters throughout. Big hugs to Apu and Kiru for their perspectives and advices on life. And a special place is always reserved for Silvia, for her comforting companionship during life's successes and struggles.

The present work would not have been possible without the financial support of the Zienkiewicz Scholarship awarded by the Zienkiewicz Centre for Computational Engineering at Swansea University, I am deeply thankful for it.

Abstract

Fast, high-fidelity solution workflows for transient flow phenomena is an important challenge in the computational fluid dynamics (CFD) community. Current low-order methodologies suffer from large dissipation and dispersion errors and require large mesh sizes for unsteady flow simulations. Recently, on the other hand, high-order methods have gained popularity offering high solution accuracy. But they suffer from the lack of robust, curvilinear mesh generators.

A novel methodology that combines the advantages of the classical vertex-centred finite volume (FV) method and high-order hybridisable discontinuous Galerkin (HDG) method is presented for the simulation of transient inviscid compressible flows. The resulting method is capable of simulating the transient effects on coarse, unstructured 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.

FV and HDG methods for compressible inviscid flows with an implicit time-stepping method and capable of handling flow discontinuities is developed. A two-way coupling of the methods in a monolithic manner was achieved by the consistent application of the so-called *transmission conditions* at the FV-HDG interface. Numerical tests highlight the optimal convergence properties of the coupled HDG-FV scheme. Numerical examples demonstrate the potential and suitability of the developed methodology for unsteady 2D and 3D flows in the context of simulating the wind gust effect on aerodynamic shapes.

Keywords: transient flows, hybridisable discontinuous Galerkin, finite volumes, coarse meshes, coupling, transmission conditions

Research output

Journal

1. **S Komala-Sheshachala**, R Sevilla, O Hassan, A coupled HDG-FV method for the simulation of transient inviscid compressible flows, *Computers & Fluids*, Vol. 202, 2020. https://doi.org/10.1016/j.compfluid.2020.104495

2. **S Komala-Sheshachala**, R Sevilla, O Hassan, A coupled HDG-FV method for 3D gust simulations. *In preparation.*

Conference talks, posters and proceedings

1. **S Komala-Sheshachala**, R Sevilla, O Hassan, A combined HDG-FV scheme for inviscid wind gust simulations, *27th Conference of the UK Association of Computational Mechanics*, UK, 2020. https://doi.org/10.17028/rd.lboro.12097644.v1

   - *Mike Crisfield Prize* for best presentation at the inaugural research highlight competition.

2. **S Komala-Sheshachala**, R Sevilla, O Hassan, A complementary low and high order scheme for unsteady flow simulations in coarse meshes, *26th Conference of the UK Association of Computational Mechanics*, UK, 2019.

3. **S Komala-Sheshachala**, R Sevilla, O Hassan, Can a mesh optimised for steady flow cater to unsteady flow simulations? *6th European Conference on Computational Mechanics & 7th European Conference on Computational Fluid Dynamics*, UK, 2018.

4. *2nd prize* (twice, in 2018 and 2019) at the Annual Postgraduate Poster Competition, Zienkiewicz Centre for Computational Engineering, Swansea, UK.

Table of contents

List of figures

List of Algorithms

Nomenclature

Roman Symbols

$\boldsymbol{B}$	Generic boundary flux
$\boldsymbol{F}$	Inviscid flux
$\boldsymbol{S}$	Volumetric source term
$\boldsymbol{\mathcal{U}}$	Conservative variable
$\boldsymbol{V}$	Conservative variable in the FV domain
$\widehat{\boldsymbol{U}}$	HDG hybrid conservative variable
$\boldsymbol{U}$	Conservative variable in the HDG domain
E	Specific total energy
H	Specific total enthalpy
h	Characteristic mesh size
k	Order of the polynomial of approximation in HDG
M	Mach number
$\boldsymbol{n}$	Unit outward normal vector
p	Pressure
$\boldsymbol{v}$	Fluid velocity
$\boldsymbol{x}$	Coordinate system in the physical space

Greek Symbols

ϵ	Solution error
η	FV dissipation factor
Γ^1	Internal interface of the HDG domain
Γ^I	HDG-FV interface
Ω	Domain of interest
$\partial\Omega$	Boundary of the domain
ρ	Fluid density
Υ	Control volume facets
$\boldsymbol{\xi}$	Coordinate system in the reference space

Superscripts

∞	Quantities associated with the far-field condition
n	BDF time step
1	Quantities associated with the HDG domain
∂	Quantities associated with the domain boundary
q	Newton-Raphson iteration step
$*$	Non-dimensional terms
2	Quantities associated with the FV domain
$\hat{\cdot}$	Quantities associated with the HDG hybrid variable

Subscripts

0	Quantities associated with the initial condition
e	Quantities associated with the HDG element e
f	Quantities associated with the HDG face f
i	Quantities associated with the FV control volume i
t	Partial derivative with respect to time

Other Symbols

$\mathtt{m}_{\mathtt{sd}}$	Number of components of the conservative variable
$\mathtt{n}_{\mathtt{cv}}$	Number of control volumes
$\mathtt{n}_{\mathtt{ef}}$	Number of element faces on the mesh skeleton
$\mathtt{n}_{\mathtt{el}}$	Number of elements
$\mathtt{n}_{\mathtt{en}}$	Number of element nodes
$\mathtt{n}_{\mathtt{me}}$	Number of edges of the FV mesh
$\mathtt{n}_{\mathtt{sd}}$	Number of spatial dimensions
$\mathtt{n}_{\mathtt{vf}}$	Number of facets of a FV vertex

Acronyms / Abbreviations

BDF	Backward Differentiation Formula
CDG	Compact Discontinuous Galerkin
CFD	Computational Fluid Dynamics
DG	Discontinuous Galerkin
FD	Finite Difference
FE	Finite Element
FV	Finite Volume
HDG	Hybridisable Discontinuous Galerkin
HPC	High-Performance Computing
IP	Interior Penalty
JST	Jameson-Schmitt-Turkel
LDG	Local Discontinuous Galerkin
PDE	Partial Differential Equation

Chapter 1

Introduction

The whole is greater than the sum of its parts.

Aristotle

1.1 Motivation

Physics-based simulation of processes have seen increasing demand over the past few decades. Computational approaches to problem solving play a decisive role not only in the design of engineering products, but in their entire life-cycle. Advances in computational capabilities like hardware acceleration are being made in conjunction with vast strides in the numerical modelling of physical phenomena. This has allowed for an ever-increasing capacity to tackle large, complex and dynamical problems.

The computational fluid dynamics (CFD) study of aerodynamic flows has evolved to embrace increasingly advanced techniques over the past few decades; shown in Figure 1.1 is the advancement in the uptake of CFD tools for commercial aircraft development at Boeing. Such progress in simulation capabilities provides new opportunities for improving aerospace vehicle designs, lowers costs in testing and broadens our understanding of the underlying physical phenomena. In pursuit of accurate numerical solutions to the Navier-Stokes equations, the recent report on NASA's CFD vision for 2030 has identified strategic areas requiring high-priority research necessary to advance the field [141]. Some of the key findings, relevant to the present work, are discussed below.

- **High-fidelity transient simulations:** High resolution simulations with improved accuracy and efficiency are essential. Particular emphasis is placed on the need for CFD in the design phase for predicting unsteady flow features with

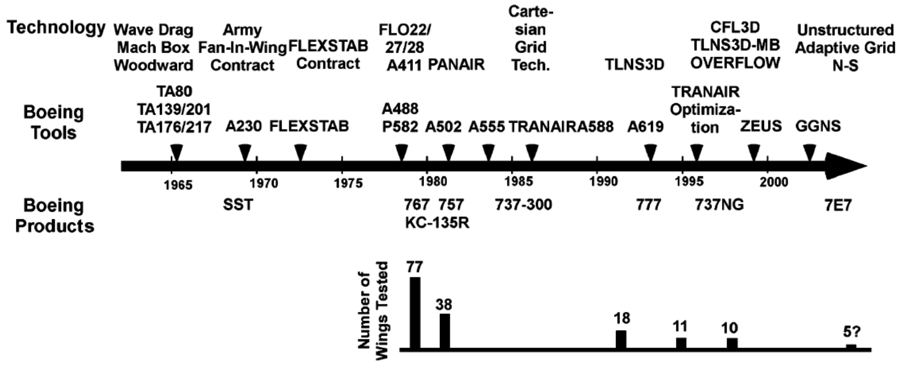


Figure. 1.1 An illustration of the advances in the adoption of CFD tools for commercial aircraft development at Boeing [78].

turbulence and significant regions of separation. The need for capability-updated CFD tools, with algorithmic improvements in discretisation methods and solvers, among other techniques, for an efficient uptake of the advances in computer hardware is highlighted.

- **Meshing:** For high-fidelity simulations on high performance computing (HPC) hardware, advancements in the individual components of the CFD workflow need to be undertaken. At present, mesh generation and adaptivity form a large bottleneck in the simulation workflow and they require more research investment.
- **Automation:** For design efforts where components of similar configurations are simulated, certain processes benefit from specialised automation workflows. Geometry preparation and meshing, which often dictate the pace of the CFD workflow, are automated for steady flow simulations. In the aerospace industry, an accurate geometrical representation of the aircraft configuration and precise calculations of the aerodynamic loads are of importance in steady flow conditions. Such automation is not feasible for the meshing of transient flows, where unsteady features are not known a-priori [147].

Low-order methods up to second-order accuracy dominate the simulation landscape in both industry and academia. Despite significant improvements over the past few decades, aircraft flow simulations have been reliable in only a restricted design space. This is, in-part, due to the large dispersion and dissipation errors of the low-order methodologies. Moreover, in viscous boundary layer meshes, elements with very large changes in length-scales are encountered. In such meshes, accurately propagating the vortical structures in an unsteady flow up to large distances would be extremely expensive with current low-order methods. The problem sizes become prohibitively large when high speed flows of industrial applications are considered. Furthermore, with expanding design space requirements such as high-lift, take-off and landing configurations, edge of flight envelope conditions, turbulence and gust simulations, fast high-resolution methods and CFD workflows are the need of the day [92, 85, 151, 152, 91, 130].

1.2 Challenges and opportunities

The path towards fast, high-fidelity solutions to unsteady phenomena is paved with attractive opportunities in the form of improved numerical techniques and methodologies. But, also challenges, both new and old, need to be overcome.

- **Low-order methods:** Low-order methods have been the workhorse of industrial and large-scale simulations over the past few decades. Second-order finite volume (FV) methods are still the predominant technique for industrial CFD applications due to their robustness and efficacy [88, 154]. 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 [103, 49, 16]. They provide with an accuracy, at most $\epsilon \propto h^2$, where ϵ is a measure of the solution error and h is a measure of the mesh size. 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.

- **Low-order mesh generation:** The meshes used for the simulations of steady-state phenomena 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.

- **High-order methods:** High-order methods in recent years have garnered attention for providing with highly accuracy solutions to the Navier-Stokes equations, especially in applications such as direct numerical simulation, large-eddy simulation and aero-acoustics [83, 159, 84]. These methods employ a high-order approximation in the discrete representation of the partial differential equations (PDE) for the spatial terms, i.e., $\epsilon \propto h^k$ where $k > 2$. Low dissipation and dispersion errors are the hallmarks of the high-order methods. When suitable meshes are available, these methods have been shown to be advantageous when compared to a low-order solution in terms of the CPU time for smooth flows and competitive for non-smooth solutions [159].

- **High-order mesh generation:** 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 [35, 14, 81, 134, 43, 99]. In fact, a low fidelity description of the geometry is known to have a major impact on the solution accuracy [167, 135, 142, 140] and, in some cases, to degrade or even prevent convergence to the correct solution [14, 136, 46, 36]. In other words, high-fidelity solutions are possible when high-order accurate representation of the geometry is available. 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 [98, 123, 150, 166, 47, 45, 106]. 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 [157, 21, 79].

1.3 Background

A brief background of the high-order methods employed in CFD, along with a detailed discussion on the discontinuous Galerkin methods relevant to the current work is presented.

1.3.1 High-order methods in CFD

High-order methods in CFD have witnessed a major increase in interest over the past few years [83, 84]. Listed below are some of the high-order extensions to prevalent low-order schemes.

In the context of finite difference (FD) methods, schemes with high-resolution compact differences for improved representation of a wide-range of scales have been proposed [87]. Also, high-order difference approximations that satisfy the summation-by-parts rule are available [149]. In the framework of finite volume (FV) methods, high-order approximation is obtained with a control volume averaged solution and high-order reconstruction of the fluxes [114, 18]. Although more widely adopted to structured or cartesian grids [60, 95], some recent advances are being made for unstructured grids, see [169] for a review. Also, developments to reduce dissipation and dispersion errors have been attempted [97]. Galerkin methods, which use piece-wise continuous approximations, have been extended to a high order context. These include the classical finite element (FE) method which employs a polynomial basis [170] and the spectral element method which utilises a trigonometric basis [118]. High-order stabilised finite element formulations have been shown to be effective [163, 19, 138, 56]. Also several other techniques have been proposed to obtain high-order accurate numerical schemes [71, 156, 160].

1.3.2 High-order discontinuous Galerkin methods

A popular family of high-order methods with demonstrated ability to accurately capture transient effects with minimum dissipation and dispersion are the discontinuous Galerkin (DG) methods [23].

A DG formulation combines the advantages of the FV and the classical continuous Galerkin (CG) methods. Both methods allow for geometric flexibility when compared to FD schemes. But they differ in the way the solution is approximated and how these approximations satisfy the underlying PDE [63]. In FV methods, the solution is approximated as piecewise constant for each element and the governing PDE is conserved locally for each element. The solution can be reconstructed up to an arbitrary order through appropriate numerical approximation of the inter-element fluxes. Although it allows for flexibility in the definition of the inter-element fluxes, extensions to a high-order context on unstructured meshes are non-trivial and a subject of active research. On the other hand, CG methods offer piecewise definition of the solution lying in a certain function space for every element. The solution is continuous over the entire domain and the PDE is satisfied in a global sense. As a result, the nodes are shared along the element boundaries as illustrated in Figure 1.2(a). Extensions to higher order approximations are relatively straight-forward, since each element can be approximated with an arbitrary order. However, for convection-dominated flows where instabilities arise due to the discretisation, the selection of proper stabilisation is not

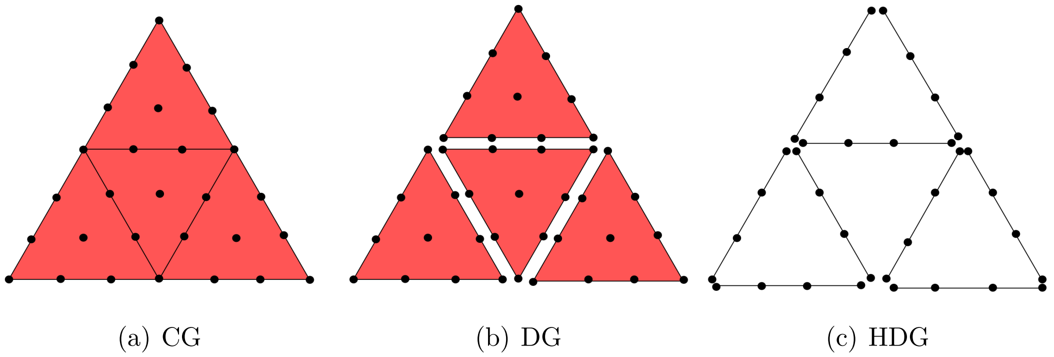
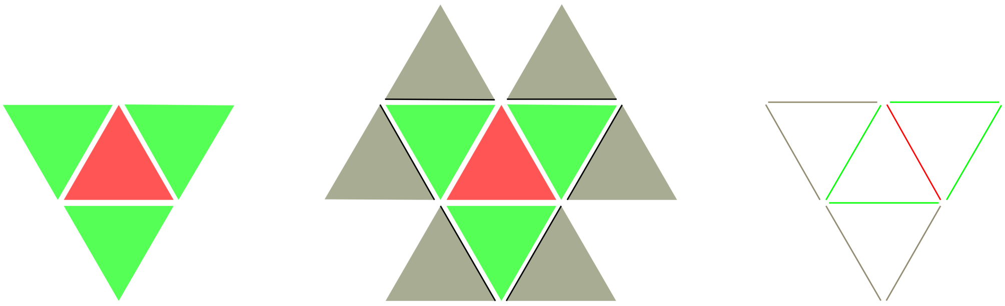


Figure. 1.2 An illustration of the solution unknowns for various discretisation methods. (a) CG elements share unknowns at the element boundaries, (b) DG methods have duplicating degrees of freedom along the element boundaries. (c) HDG method involves solving globally for the hybrid variable defined at the element boundaries.

straight-forward. Optimal stabilisation is known only for linear scalar problems in 1D with linear elements [39]. However, new approaches to stabilise convection-dominated flows through appropriate use of boundary conditions have recently been developed [1]. DG methods strive to overcome the disadvantages of both FV and CG methods, while retaining their benefits. A DG solution is constructed from a piecewise definition lying in a certain functional basis, like the CG method, but discontinuous across element boundaries. Here, duplication of the degrees of freedom lying on the element boundaries becomes necessary, as shown in Figure 1.2(b). The solution satisfies the equation in a local manner similar to the FV method. If limiters are not required, for wave-like behaviour of the PDE, stabilisation can be introduced to the inter-element fluxes without losing the compactness [30, 23].

The DG method was initially developed to solve the neutron transport problem [125]. Later it saw applications in many hyperbolic systems [7, 31, 38, 41]. Extensions to high-order PDEs were introduced through the so-called mixed formulation, where the high-order operators are reduced to a system consisting of first-order operators [13]. The major criticism faced by the DG methods has been the high computational cost. Large number of node duplications at the element boundaries result in a large system of unknowns to be solved for, especially for solutions in 3D domains. Although the issue is somewhat alleviated for high-order approximations as the proportion of the number of repeated degrees of freedom relative to the total is smaller, it is nevertheless significant. New DG methodologies have been introduced in order to reduce the computational requirements. Listed below is an overview of some of the advances:



(a) IPM

(b) LDG

(c) HDG

Figure. 1.3 An illustration of the stencils for different DG methods. (a) IP method for the element in red includes contribution from its neighbouring elements in green. (b) LDG method for the element in red includes the contributions from its immediate neighbours in green as well as the face unknowns in black for elements in grey. (c) HDG method has a compact stencil. For the hybrid variable on the face in red, its stencil includes only the unknowns on green.

- **Interior Penalty (IP) method:** The idea of introducing penalising terms to enforce inter-element continuity was proposed in [3]. This was inspired by the method of imposition of weak Dirichlet boundary conditions by Nitsche [113]. Symmetric and a non-symmetric variants are possible, which show an optimal convergence of $k{+}1$ and k in the $\mathcal{L}_2(\Omega)$ respectively when polynomial of degree k is utilised. This allowed for second-order PDEs to be handled without decomposing into two first-order PDEs, but the disadvantage is the requirement to fine-tune the penalty parameter for different polynomial approximations and element sizes. The stencil for the IP method is illustrated in Figure 1.3(a). See [128, 48] for a detailed analysis of the IP method.

- **Local DG (LDG) method:** The LDG method utilises the mixed formulation with various choice of numerical fluxes available at the inter-element boundaries, as detailed in [4]. After introducing the so-called lifting operators, it can be shown that the LDG method can be written as a combination of the IP method and a few additional terms. Unlike the IP method, LDG method is less sensitive to the penalty parameter. As indicated in Figure 1.3(b), the stencil for the LDG method not only involves the unknowns of its immediate neighbours, but some from their neighbours as well. This loss in compactness affects the sparsity of the resulting linear system. The LDG method converges at the optimal rate of $k+1$ in the $\mathcal{L}_2(\Omega)$.

- **Compact DG (CDG) method:** As suggested by the name, the CDG method was introduced as a remedy to address the large stencil in the LDG method [121]. The computational cost in the form of memory and CPU time is reduced when compared to the LDG method, with only a small overhead in computing the lifting operators. The CDG method retains the convergence rate of $k+1$ in the $\mathcal{L}_2(\Omega)$, similar to the LDG method.

- **Hybridisable DG (HDG) method:** Recently, a particular DG method, called hybridisable DG (HDG) method [25, 26, 28, 29], has 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 [69, 168]. The method introduces a hybrid variable defined uniquely on the inter-element boundaries, thus making the formulation amenable to the static condensation technique. The reduced total number of unknowns include the fewest repetition; duplication of the degrees of freedom occurs only at the vertices in 2D and along the edge in 3D, as illustrated in Figure 1.2(c). The compactness of the method, as illustrated in Figure 1.3(c), shows the stencil for a given face only includes the unknowns of its constituent elements. The HDG method converges at the optimal rate of $k+1$ in the $\mathcal{L}_2(\Omega)$ for first-order PDEs. For second order PDEs, a mixed formulation yields a solution converging optimally for both the primal and dual variable at a rate of $k+1$ in the $\mathcal{L}_2(\Omega)$. Moreover, a so-called *super-convergent* solution for the primal variable can be constructed through a simple post-processing technique which results in the convergence of the solution at a rate $k+2$ in the $\mathcal{L}_2(\Omega)$. A further improvement to reduce the number of repeated nodes in the HDG method has had limited success, suffering from a loss of accuracy in mixed formulations [119]. The application and performance of HDG for CFD applications have been studied by many authors, see for instance [112, 120, 54, 165, 42, 127]. A detailed description of the HDG method is provided in Chapter 3.

1.4 Scope of the thesis

With relevant background and motivation highlighted in the previous sections, the discussion leads to the key proposal introduced in the current work. The rest of the chapter is dedicated to presenting the proposed methodology and the outline of the thesis.

1.4.1 Proposed methodology

The present 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 and three dimensional aerodynamic shapes using the 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 FV method in the vicinity of the aerodynamic shape ensures a minimum number of degrees of freedom when compared to cell-centred or face-centred FV scheme for unstructured meshes [137] 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. Also, the approach circumvents the necessity to generate multiple, tailor-made meshes for transient simulations. Moreover, the proposed methodology would enable cheap and easy degree adaptivity in the HDG subdomain, which will eliminate the need for adaptive re-meshing in many applications. 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 [20], where FVs are coupled to high-order finite differences for capturing trailing vortical structures and in [40], where FVs are coupled to DG methods for addressing discontinuities. The coupling of HDG with other techniques has also been proposed. For instance, HDG is coupled with a boundary element method in [32] and HDG is coupled to standard finite elements in the framework of second-order elliptic problems in [115, 148].

1.4.2 Outline

A brief outline of the structure of the thesis is given below:

- **Chapter 2:** Euler equations in the conservation form for the ideal fluid in a compressible, inviscid flow is introduced. The hyperbolic properties for the quasi-linear form of the Euler equations and their relevance to the imposition of boundary conditions are discussed. A strategy to generate time-harmonic gusts in the computational domain in the form of a source term to the Euler equations is presented.

- **Chapter 3:** In the beginning, a brief discussion on the development of the HDG method is provided. A succinct framework, with relevant preliminary details, is presented. This is followed by a two-stage weak formulation for the Euler equations. Shock-capturing method, weak-form imposition of the boundary conditions, spatial and temporal discretisations are detailed. A linearisation method for the non-linear residual, resulting in a set of discrete system of sparse linear equations, is presented. Key implementation details are provided. The Ringleb flow problem as a suitable candidate for spatial convergence tests is described. The chapter concludes with a suite of 2D and 3D test cases for the spatial and temporal convergence properties of the HDG method.

- **Chapter 4:** The chapter details the vertex-centred FV method applied to the Euler equations. Preliminary information regarding the dual-mesh construction and edge-based data structure is presented. Second-order spatial and temporal discretisations, along with biharmonic artificial dissipation and harmonic shock capturing terms are presented. An implicit formulation, leading to a discrete linearised system of equations, is derived. Treatment of the relevant boundary conditions is highlighted. Remarks on key implementation details are provided. Finally, the spatial and temporal convergence properties are presented with a set of test cases.

- **Chapter 5:** A novel coupling of the HDG and FV methods, individually detailed in the previous chapters, is carried out. A set of *transmission conditions* across the HDG-FV interface is introduced and incorporated into the preceding weak formulations of the HDG and FV methods. A resulting monolithic weak form of the coupled method is derived. Special attention highlighting the issue of the interface treatment is provided along with the implementation details. Numerical tests of the coupled method, demonstrating the shock capturing capabilities and the optimal spatial and temporal convergence properties, concludes the chapter.

- **Chapter 6:** Numerical examples are presented to demonstrate the advantages of the proposed HDG-FV method over the standard second-order FV scheme. Unsteady gust flow simulations are carried out for a single aerofoil and a two-aerofoil configurations in 2D domains and for the ONERA M6 wing in a 3D domain. Selection of mesh designed for steady-state simulation for use in the gust flow example is highlighted. A methodology for partitioning the computational domain into HDG and FV subdomains based on element size is detailed. Also, performance benefits of the coupled scheme are analysed.
- **Chapter 7:** The thesis ends with some concluding remarks. It recalls the contribution of the present work and comments on the future outlook for the coupling strategy presented.

Chapter 2

Governing equations

> *"The breaking of the wave cannot explain the whole sea."*
>
> Vladimir Nabokov

The present chapter is devoted to a review of the Euler equations of gas dynamics which govern compressible and inviscid flows. A brief description of the governing equations and its non-dimensional form is provided. The wave-like nature of the equations is discussed and the resulting influence on the imposition of boundary conditions is explained. Furthermore, a detailed setup for the sinusoidal gust generation, encountered in the examples in Chapter 6, is introduced.

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,

Mass:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho v_1}{\partial x_1} + \frac{\partial \rho v_2}{\partial x_2} + \frac{\partial \rho v_3}{\partial x_3} = 0, \tag{2.1.1a}$$

Linear momentum:

$$\frac{\partial \rho v_1}{\partial t} + \frac{\partial \rho v_1^2}{\partial x_1} + \frac{\partial \rho v_1 v_2}{\partial x_2} + \frac{\partial \rho v_1 v_3}{\partial x_3} = -\frac{\partial p}{\partial x_1} + \rho b_1, \tag{2.1.1b}$$

$$\frac{\partial \rho v_2}{\partial t} + \frac{\partial \rho v_2 v_1}{\partial x_1} + \frac{\partial \rho v_2^2}{\partial x_2} + \frac{\partial \rho v_2 v_3}{\partial x_3} = -\frac{\partial p}{\partial x_2} + \rho b_2, \tag{2.1.1c}$$

$$\frac{\partial \rho v_3}{\partial t}+\frac{\partial \rho v_3 v_1}{\partial x_1}+\frac{\partial \rho v_3 v_2}{\partial x_2}+\frac{\partial \rho v_3^2}{\partial x_3} = -\frac{\partial p}{\partial x_3}+\rho b_3, \tag{2.1.1d}$$

Total energy:

$$\frac{\partial \rho E}{\partial t}+\frac{\partial \rho H v_1}{\partial x_1}+\frac{\partial \rho H v_2}{\partial x_2}+\frac{\partial \rho H v_3}{\partial x_3} = \rho(v_1 b_1+v_2 b_2+v_3 b_3). \tag{2.1.1e}$$

Here, ρ is the density, v_1, v_2 and v_3 are the components of the velocity vector $\boldsymbol{v}$, E is the specific total energy, p is the pressure, $H = E + p/\rho$ is the specific total enthalpy of the fluid and b_1, b_2 and b_3 are the components of the specific body force vector $\boldsymbol{b}$ acting on the fluid.

As in classical gas dynamics, the only influences on the momentum equation are its redistribution and pressure. Similarly, the only influences on the energy equation are its redistribution, conversion of energy to or from momentum and pressure [86]. The coordinate invariant form of the above equation can be written as,

$$\frac{\partial \rho}{\partial t}+\boldsymbol{\nabla}\cdot(\rho\boldsymbol{v}) = 0, \tag{2.1.2a}$$

$$\frac{\partial \rho\boldsymbol{v}}{\partial t}+\boldsymbol{\nabla}\cdot(\rho\boldsymbol{v}\otimes\boldsymbol{v}+p\mathbf{I}_{\mathtt{n_{sd}}}) = \rho\boldsymbol{b}, \tag{2.1.2b}$$

$$\frac{\partial \rho E}{\partial t}+\boldsymbol{\nabla}\cdot(\rho H\boldsymbol{v}) = \boldsymbol{v}\cdot\rho\boldsymbol{b}. \tag{2.1.2c}$$

Using Einstein summation convention, the nabla operator $\boldsymbol{\nabla} = \dfrac{\partial}{\partial x_l}\boldsymbol{e}_l$, where $\{\boldsymbol{e}_1,\ldots,\boldsymbol{e}_{\mathtt{n_{sd}}}\}$ is the standard basis and $\mathtt{n_{sd}}$ is the number of spatial dimensions. The vector outer product $\boldsymbol{v}\otimes\boldsymbol{v} = v_i v_j$ and $\mathbf{I}_{\mathtt{n_{sd}}}$ is the identity matrix of dimension $\mathtt{n_{sd}}$. Writing the Euler equations in a compact notation along with initial and boundary conditions, we get,

$$\boldsymbol{\mathcal{U}}_t+\boldsymbol{\nabla}\cdot\boldsymbol{F}(\boldsymbol{\mathcal{U}}) = \boldsymbol{S} \quad \text{in } \Omega\times(0,T], \tag{2.1.3a}$$

$$\boldsymbol{\mathcal{U}} = \boldsymbol{\mathcal{U}}_0 \quad \text{in } \Omega\times\{0\}, \tag{2.1.3b}$$

$$\boldsymbol{B}(\boldsymbol{\mathcal{U}},\boldsymbol{\mathcal{U}}^\infty) = \mathbf{0} \quad \text{on } \partial\Omega\times(0,T], \tag{2.1.3c}$$

where $\boldsymbol{\mathcal{U}}$ denotes the vector of conservation variables, the tensor $\boldsymbol{F}$ contains the hyperbolic flux vector for each spatial dimension x_l, $(l = 1,\ldots,\mathtt{n_{sd}})$, T is the final time, $\boldsymbol{\mathcal{U}}_0$ denotes the initial condition, $\boldsymbol{B}$ is a generic flux used to define the boundary conditions over the inflow, outflow and wall boundaries and $\boldsymbol{\mathcal{U}}^\infty$ denotes the free-stream state. The source term $\boldsymbol{S}$ in Equation (2.1.6) usually accounts for the external volume forces. The vectors $\boldsymbol{\mathcal{U}}$ and $\boldsymbol{S}$ of size $\mathtt{m_{sd}} = \mathtt{n_{sd}}+2$ and tensor $\boldsymbol{F}$ of size $\mathtt{m_{sd}}\times\mathtt{n_{sd}}$ are given by

$$\boldsymbol{\mathcal{U}} = \begin{Bmatrix} \rho \\ \rho\boldsymbol{v} \\ \rho E \end{Bmatrix}, \quad \boldsymbol{F} = \begin{bmatrix} \boldsymbol{F}_1 & \dots & \boldsymbol{F}_{\mathrm{n_{sd}}} \end{bmatrix} = \begin{bmatrix} \rho\,\boldsymbol{v}^T \\ \rho\boldsymbol{v}\otimes\boldsymbol{v} + p\mathbf{I}_{\mathrm{n_{sd}}} \\ (\rho E + p)\boldsymbol{v}^T \end{bmatrix}, \quad \boldsymbol{S} = \begin{Bmatrix} 0 \\ \rho\boldsymbol{b} \\ \boldsymbol{v}\cdot\rho\boldsymbol{b} \end{Bmatrix}. \tag{2.1.4}$$

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

$$p = (\gamma - 1)\rho\left(E - \frac{1}{2}\|\boldsymbol{v}\|^2\right), \tag{2.1.5}$$

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

A usual quantity for post-process 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. In general, the flow regimes can be classified as subsonic ($0.1 < M < 0.7$), transonic ($0.7 < M < 1.3$) and supersonic ($1.3 < M < 3$).

2.1.1 The dimensionless form

Typically, the Euler equations are programmed to be solved in a dimensionless form. The non-dimensional terms $(\cdot)^*$ are obtained using reference values of the variables $(\cdot)_{\mathrm{ref}}$. Firstly, a reference length scale x_{ref} and reference velocity scale v_{ref} are chosen. Now, either a scaling for the pressure p_{ref} or the density ρ_{ref} is required since they are related by $\dim(p_{\mathrm{ref}}) = \dim(\rho_{\mathrm{ref}}\, v_{\mathrm{ref}}^2)$. Now, the reference total energy is invoked by either $\rho_{\mathrm{ref}}\, v_{\mathrm{ref}}^2$ or p_{ref} and the reference time is given by $x_{\mathrm{ref}}/v_{\mathrm{ref}}$. In the presence of external body forces given by the source $\boldsymbol{S}$ in Equation (2.1.4), an external force scaling term b_{ref} is applied.

The resulting non-dimensional form of Equation 2.1.3 is given as,

$$\boldsymbol{\mathcal{U}}^*{}_{t^*} + \boldsymbol{\nabla}^* \boldsymbol{\cdot} \boldsymbol{F}^*\left(\boldsymbol{\mathcal{U}}^*\right) = \mathrm{Fr}^{-2}\, \boldsymbol{S}^* \qquad \text{in } \Omega\times(0,T] \tag{2.1.6a}$$

$$\boldsymbol{\mathcal{U}}^* = \boldsymbol{\mathcal{U}}_0^* \qquad \text{in } \Omega\times\{0\} \tag{2.1.6b}$$

$$\boldsymbol{B}\left(\boldsymbol{\mathcal{U}}^*, \boldsymbol{\mathcal{U}}^{\infty,*}\right) = \boldsymbol{0} \qquad \text{on } \partial\Omega\times(0,T] \tag{2.1.6c}$$

where, the Froude's number, $\mathrm{Fr} = \dfrac{v_{\mathrm{ref}}}{\sqrt{x_{\mathrm{ref}}\, b_{\mathrm{ref}}}}$, is a measure of the relative strengths of inertial forces to body forces respectively [162]. In the present work, no body forces are considered and the source term only accounts for the generated gust, see Section 2.2. Also, for the sake of brevity, henceforth, the superscripts in the non-dimensional equations are dropped.

2.1.2 The quasi-linear form

Important properties of the Euler equations are revealed in its quasi-linear form. For a perfect polytropic fluid, the components of the inviscid flux $\boldsymbol{F}_l$, for $l = 1, \ldots, \mathtt{n_{sd}}$, are shown to be *homogeneous functions of degree* 1 [39]. This allows for the flux components to be written as,

$$\boldsymbol{F}_l\left(\boldsymbol{\mathcal{U}}\right) = \boldsymbol{A}_l\left(\boldsymbol{\mathcal{U}}\right) \boldsymbol{\mathcal{U}}, \tag{2.1.7}$$

where the tensor,

$$\boldsymbol{A}_l\left(\boldsymbol{\mathcal{U}}\right) = \frac{\partial \boldsymbol{F}_l}{\partial \boldsymbol{\mathcal{U}}}, \tag{2.1.8}$$

for $l = 1, \ldots, \mathtt{n_{sd}}$ is the jacobian of the inviscid flux. The Equation (2.1.6a) can be expressed in its quasi-linear form as,

$$\frac{\partial \boldsymbol{\mathcal{U}}}{\partial t} + \boldsymbol{A}_l\left(\boldsymbol{\mathcal{U}}\right) \frac{\partial \boldsymbol{\mathcal{U}}}{\partial x_l} = \boldsymbol{S}. \tag{2.1.9}$$

The equation admits wave-like properties when the eigenvalues of the projected jacobian matrix $\boldsymbol{A}_n\left(\boldsymbol{\mathcal{U}}\right) = \boldsymbol{A}_l\left(\boldsymbol{\mathcal{U}}\right) n_l$ along any direction $\boldsymbol{n}$ are real and linearly independent with its corresponding right eigenvectors [66]. Consider the eigenvalue decomposition of the projected matrix,

$$\boldsymbol{A}_n\left(\boldsymbol{\mathcal{U}}\right) = \boldsymbol{R}\left(\boldsymbol{\mathcal{U}}\right)\ \boldsymbol{\Lambda}_n\left(\boldsymbol{\mathcal{U}}\right)\ \boldsymbol{L}\left(\boldsymbol{\mathcal{U}}\right), \tag{2.1.10}$$

where, $\boldsymbol{\Lambda}_n := \mathrm{diag}(\lambda_j), j = 1, \cdots, \mathtt{m_{sd}}$, is the diagonal matrix of eigenvalues, $\boldsymbol{R}$ is the vector of corresponding right eigenvectors of $\boldsymbol{A}_n$ and $\boldsymbol{L} = \boldsymbol{R}^{-1}$ is the vector of left eigenvectors of $\boldsymbol{A}_n$. It is important to note that the projection of the jacobian matrix along an arbitrary direction is diagonalisable but not the jacobian matrix itself. The general form of the matrices [129] are given below,

$$\boldsymbol{A}_n = \begin{bmatrix} 0 & n_1 & n_2 & n_3 & 0 \\ (\gamma-1)E_k n_1 - v_1 v_n & v_n - (\gamma-2)v_1 n_1 & v_1 n_2 - (\gamma-1)v_2 n_1 & v_1 n_3 - (\gamma-1)v_3 n_1 & (\gamma-1)n_1 \\ (\gamma-1)E_k n_2 - v_2 v_n & v_2 n_1 - (\gamma-1)v_1 n_2 & v_n - (\gamma-2)v_2 n_2 & v_2 n_3 - (\gamma-1)v_3 n_2 & (\gamma-1)n_2 \\ (\gamma-1)E_k n_3 - v_3 v_n & v_3 n_1 - (\gamma-1)v_1 n_3 & v_3 n_2 - (\gamma-1)v_2 n_3 & v_n - (\gamma-2)v_3 n_3 & (\gamma-1)n_3 \\ \left((\gamma-1)E_k - H\right) v_n & H n_1 - (\gamma-1)v_1 v_n & H n_2 - (\gamma-1)v_2 v_n & H n_3 - (\gamma-1)v_3 v_n & \gamma v_n \end{bmatrix}, \tag{2.1.11}$$

$$\boldsymbol{R} = \begin{bmatrix}\boldsymbol{R}_1 & \boldsymbol{R}_2 & \boldsymbol{R}_3 & \boldsymbol{R}_4 & \boldsymbol{R}_5\end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ v_1 - cn_1 & v_1 & v_1 + cn_1 & n_2 & -n_3 \\ v_2 - cn_2 & v_2 & v_2 + cn_2 & -n_1 & 0 \\ v_3 - cn_3 & v_3 & v_3 + cn_3 & 0 & n_1 \\ H - cv_n & E_k & H + cv_n & v_1 n_2 - v_2 n_1 & v_3 n_1 - v_1 n_3 \end{bmatrix}, \tag{2.1.12}$$

$$\boldsymbol{\Lambda}_n = \begin{bmatrix} v_n - c & 0 & 0 & 0 & 0 \\ 0 & v_n & 0 & 0 & 0 \\ 0 & 0 & v_n + c & 0 & 0 \\ 0 & 0 & 0 & v_n & 0 \\ 0 & 0 & 0 & 0 & v_n \end{bmatrix}, \tag{2.1.13}$$

$$\boldsymbol{L} = \begin{bmatrix} \boldsymbol{L}_1^T \\ \boldsymbol{L}_2^T \\ \boldsymbol{L}_3^T \\ \boldsymbol{L}_4^T \\ \boldsymbol{L}_5^T \end{bmatrix} = \begin{bmatrix} \frac{(\gamma-1)E_k + cv_n}{2c^2} & \frac{(1-\gamma)v_1 - cn_1}{2c^2} & \frac{(1-\gamma)v_2 - cn_2}{2c^2} & \frac{(1-\gamma)v_3 - cn_3}{2c^2} & \frac{\gamma-1}{2c^2} \\ \frac{c^2 - (\gamma-1)E_k}{c^2} & \frac{(\gamma-1)v_1}{c^2} & \frac{(\gamma-1)v_2}{c^2} & \frac{(\gamma-1)v_3}{c^2} & \frac{1-\gamma}{c^2} \\ \frac{(\gamma-1)E_k - cv_n}{2c^2} & \frac{(1-\gamma)v_1 + cn_1}{2c^2} & \frac{(1-\gamma)v_2 + cn_2}{2c^2} & \frac{(1-\gamma)v_3 + cn_3}{2c^2} & \frac{\gamma-1}{2c^2} \\ \frac{v_2 - v_n n_2}{n_1} & n_2 & \frac{n_2^2 - 1}{n_1} & \tfrac{n_2 n_3}{n_1} & 0 \\ \tfrac{v_n n_3 - v_3}{n_1} & -n_3 & \frac{-n_2 n_3}{n_1} & \tfrac{1-n_3^2}{n_1} & 0 \end{bmatrix}. \tag{2.1.14}$$

Here, n_1, n_2 and n_3 are the components of the unit vector $\boldsymbol{n}$ along the projected direction, $E_k = \dfrac{||\boldsymbol{v}||^2}{2}$ is the specific kinetic energy and $v_n = \boldsymbol{v} \cdot \boldsymbol{n}$ is the normal velocity. The jacobian matrices projected along the unit vectors of the coordinate axes can be deduced as,

$$\boldsymbol{A}_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ (\gamma-1)E_k - v_1^2 & (3-\gamma)v_1 & (1-\gamma)v_2 & (1-\gamma)v_3 & (\gamma-1) \\ -v_2 v_1 & v_2 & v_1 & 0 & 0 \\ -v_3 v_1 & v_3 & 0 & v_1 & 0 \\ \left((\gamma-1)E_k - H\right) v_1 & H - (\gamma-1)v_1^2 & (1-\gamma)v_2 v_1 & (1-\gamma)v_3 v_1 & \gamma v_1 \end{bmatrix}, \tag{2.1.15a}$$

$$\boldsymbol{A}_2 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ -v_1 v_2 & v_2 & v_1 & 0 & 0 \\ (\gamma-1)E_k - v_2^2 & (1-\gamma)v_1 & (3-\gamma)v_2 & (1-\gamma)v_3 & (\gamma-1) \\ -v_3 v_2 & 0 & v_3 & v_2 & 0 \\ \left((\gamma-1)E_k - H\right) v_2 & (1-\gamma)v_1 v_2 & H-(\gamma-1)v_2^2 & (1-\gamma)v_3 v_2 & \gamma v_2 \end{bmatrix}, \tag{2.1.15b}$$

$$\boldsymbol{A}_3 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ -v_1 v_3 & v_3 & 0 & v_1 & 0 \\ -v_2 v_3 & 0 & v_3 & v_2 & 0 \\ (\gamma-1)E_k - v_3^2 & (1-\gamma)v_1 & (1-\gamma)v_2 & (3-\gamma)v_3 & (\gamma-1) \\ \left((\gamma-1)E_k - H\right) v_3 & (1-\gamma)v_1 v_3 & (1-\gamma)v_2 v_3 & H-(\gamma-1)v_3^2 & \gamma v_3 \end{bmatrix}. \tag{2.1.15c}$$

Note that, for 2D flows, the necessary forms of the matrices above can be obtained by removing appropriate rows and columns and setting $n_3 = v_3 = 0$.

2.1.3 Imposition of boundary conditions

A careful application of boundary conditions is critical for obtaining accurate solutions to Euler equations [124]. The hyperbolic nature of the solution requires the imposition of boundary conditions to respect the direction of information travel for the different characteristic waves. An overview of the considerations are provided here and detailed implementations of the procedure is described in the upcoming chapters for the HDG and FV methods.

For slip wall conditions, no mass or energy transport occurs across the boundary. Only linear momentum transport in the tangential direction to the wall is allowed. Hence, only the momentum equations are modified. This is straight-forward and takes the form of imposing $v_n = 0$. It is a widely-accepted standard practice to apply a similar procedure to the symmetry boundary [145]. However, additional constraints may be applied at the symmetry boundary, see for instance [126].

In two and three dimensional flows, the jacobian in Equation (2.1.8) cannot be diagonalised since information can travel in infinite directions and decomposition into independent scalar waves is not possible. This is only achievable in the one dimensional case. Consider the one dimensional quasi-linear form of Equation (2.1.9), premultiplied by $\boldsymbol{L}$,

$$\boldsymbol{L}\left(\frac{\partial \boldsymbol{\mathcal{U}}}{\partial t} + \boldsymbol{A}_1(\boldsymbol{\mathcal{U}})\frac{\partial \boldsymbol{\mathcal{U}}}{\partial x_1}\right) = \boldsymbol{L}\,\boldsymbol{S}. \tag{2.1.16}$$

Defining the so-called characteristic variable or Riemann variable $\boldsymbol{W} = \boldsymbol{L}\boldsymbol{\mathcal{U}}$, one can verify that the following holds true,

$$\frac{\partial \boldsymbol{W}}{\partial t} + \boldsymbol{\Lambda}_1 \left(\boldsymbol{\mathcal{U}}\right) \frac{\partial \boldsymbol{W}}{\partial x_1} = \boldsymbol{L}\, \boldsymbol{S}, \tag{2.1.17}$$

where,

$$\boldsymbol{\Lambda}_1 = \begin{bmatrix} v_1 & 0 & 0 \\ 0 & v_1 + c & 0 \\ 0 & 0 & v_1 - c \end{bmatrix}. \tag{2.1.18}$$

Now, the Euler equations can be decoupled into three scalar convection equations. The characteristic equations define three distinct speeds and direction of information travel. They are given by,

$$C_0 : \qquad \frac{dx_1}{dt} = v_1, \tag{2.1.19a}$$

$$C_+ : \qquad \frac{dx_1}{dt} = v_1 + c, \tag{2.1.19b}$$

$$C_- : \qquad \frac{dx_1}{dt} = v_1 - c. \tag{2.1.19c}$$

The expressions for the characteristic variable that is being transported is possible in the restrictive case of one dimensional flows only and is given by,

$$\boldsymbol{W} = \begin{Bmatrix} s \\ v_1 + \dfrac{2c}{\gamma - 1} \\ v_1 - \dfrac{2c}{\gamma - 1} \end{Bmatrix}. \tag{2.1.20}$$

At an external boundary, the advection of information along the direction normal to the boundary is of importance. Thus the diagonalisation can be performed in the normal direction. Here, the direction of the outward unit normal $\boldsymbol{n}$, the flow velocity and the local speed of sound, together dictate the speed of information travel according to Equation (2.1.13). Three distinct wave speeds are encountered, v_n, $v_n + c$ and $v_n - c$. Negative characteristics signify information travelling into the domain and positive characteristics denote information leaving the domain. The jacobian matrix associated with the outgoing (+) and incoming (−) characteristics is given by,

$$\boldsymbol{A}_n^{\pm} \left(\boldsymbol{\mathcal{U}}\right) = \frac{1}{2} \left(\boldsymbol{A}_n \left(\boldsymbol{\mathcal{U}}\right) \pm \left|\boldsymbol{A}_n \left(\boldsymbol{\mathcal{U}}\right)\right|\right), \tag{2.1.21}$$

where,

$$|\boldsymbol{A}_n(\boldsymbol{\mathcal{U}})| = \boldsymbol{R}(\boldsymbol{\mathcal{U}})\ |\boldsymbol{\Lambda}_n(\boldsymbol{\mathcal{U}})|\,\boldsymbol{L}(\boldsymbol{\mathcal{U}})\,. \tag{2.1.22}$$

Hence, prescribed boundary conditions correspond to negative characteristics only. The boundary conditions are applied on the characteristic or Riemann variable $\boldsymbol{W}$ and then transformed to the conservative variable $\boldsymbol{\mathcal{U}}$.
For supersonic flow cases, where local Mach number $M > 1$, the following considerations need to be undertaken.

- At inflow boundaries, $v_n < 0$ and $|v_n| > c$. This implies that all characteristic waves enter the domain and need to be prescribed for the Riemann variables $\boldsymbol{W}$ on the boundary.

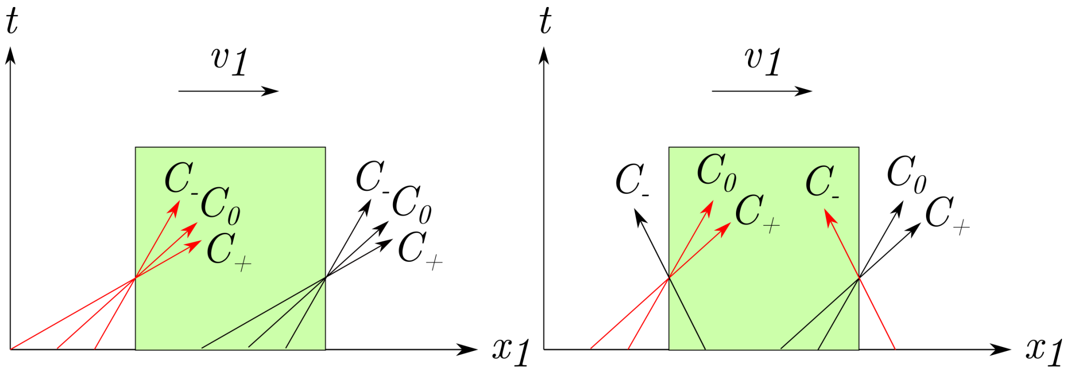
- At outflow boundaries, $v_n > 0$ and $|v_n| > c$ and all the characteristics are positive. Hence, no prescription is necessary since information travels to the boundary from the interior of the domain.

Subsonic flow conditions $M < 1$ require that the following conditions are satisfied.

- At inflow boundaries, $v_n < 0$ and $|v_n| < c$. This implies that $\texttt{n}_{\texttt{sd}} + 1$ characteristic waves enter the domain and need to be prescribed at the boundary. Riemann variables corresponding to the wave speeds v_n and $v_n - c$ are obtained from the external conditions and the one corresponding to the wave speed $v_n + c$ is obtained from the domain interior. Generally, density and velocity are prescribed here.

- At outflow boundaries, $v_n > 0$ and $|v_n| < c$ and all characteristics except one are positive. Characteristic variable corresponding to the wave speed $v_n - c$ is prescribed by the exterior condition, whereas the other variables corresponding to the wave speeds v_n and $v_n + c$ are obtained from the domain interior. Usually pressure is prescribed at these boundaries.

In practise, unit outward normal to the inflow/outflow boundaries are well-defined. In the FV implementation, the outward pointing normal is defined for each facet of the edge connecting two boundary nodes. Similarly, in the HDG implementation, unit outward normal may be evaluated at each gauss point location on the element face. At sharp edges such as the aerofoil trailing edges, contributions arising from the FV facets or HDG faces with well-defined normals are treated naturally. Non-physical, numerical perturbations were not encountered.



(a) (b)

Figure. 2.1 A space-time plot of the characteristics, given in Equation (2.1.19), for (a) supersonic and (b) subsonic 1D flows. The incoming characteristics are denoted in red and outgoing characteristics are in black. The green region represents the space-time domain

Figure 2.1 illustrates the characteristics for supersonic and subsonic flows in a 1D domain, where the inflow and outflow characteristics are highlighted in red and black lines respectively.

2.2 Gust problem setup

Gust phenomena plays a major role in aircraft design and are crucial in determining critical load cases. They represent unsteady flow phenomena that have been traditionally modelled through different approaches. One popular method is the generation of desired vortical gusts through the imposition of time-dependent inflow/outflow boundary conditions [67, 58]. It is critical that a fluctuating mean flow is maintained and that the reflection of disturbances at the boundary is minimal. Moreover, the path from the far-field boundary to the aerodynamic configuration needs to propagate the gust fluctuations without large dispersion and dissipation errors, necessitating a large zone of mesh refinement, which increases the possibility of spurious boundary reflections.

An alternative methodology involves introducing the gust inside the domain as a source term in the governing equations. It was first proposed in [96] and is the procedure of choice in the present work. The harmonic perturbation of the velocity field is incompressible and vortical in nature and thus only affects the momentum equation [57, 96]. Contrary to the previous approach, this allows for the use of stretched grids in the far-field to minimise wave reflections. Moreover, unlike forcing a perturbation in the conservative variables, introducing the sinusoidal variation as a source term would not decouple the effect of the fluid flow on the the domain lying in the region of gust generation [70].

Consider a flow in the $x_1 - x_2$ plane. Two-dimensional vortical perturbation of the velocity field $\{v_1^g \;\; v_2^g\}^T$ is introduced in the form described below,

$$v_1^g = \delta_1 \cos(\alpha x_1 + \beta x_2 - \omega^g t), \tag{2.2.1a}$$

$$v_2^g = \delta_2 \cos(\alpha x_1 + \beta x_2 - \omega^g t), \tag{2.2.1b}$$

where the gust amplitudes,

$$\delta_1 = -\frac{\delta^g \beta ||\boldsymbol{v}^\infty||}{\sqrt{\alpha^2 + \beta^2}}, \qquad \delta_2 = \frac{\delta^g \alpha ||\boldsymbol{v}^\infty||}{\sqrt{\alpha^2 + \beta^2}}. \tag{2.2.2}$$

Here, δ^g is the gust intensity relative to the mean flow, $\alpha = \omega^g / ||\boldsymbol{v}^\infty||$ and $\beta = \alpha \tan\theta$ represent the wave numbers of the sinusoidal gust in the x_1 and x_2 direction respectively, ω^g is the imposed gust frequency, $||\boldsymbol{v}^\infty||$ is the freestream velocity and θ is the angle of propagation of the gust front with respect to the x_1 axis. Note that the gust field satisfies the incompressibility condition, that is, $\boldsymbol{\nabla} \cdot \{v_1^g \;\; v_2^g\}^T = 0$

The source term $\boldsymbol{S}(\boldsymbol{x}, t)$ to generate the desired gust is composed of two non-zero terms among $\mathtt{m_{sd}}$ components, namely S_1 and S_2. These terms act on the momentum equation along the x_1 and x_2 directions respectively and are given by,

$$S_1(\boldsymbol{x}, t) = \beta K g(x_1) \lambda(x_2) \cos\left(\omega^g t - \beta x_2 - \alpha x_1^g\right), \tag{2.2.3a}$$

$$S_2(\boldsymbol{x}, t) = K g'(x_1) \lambda(x_2) \sin\left(\omega^g t - \beta x_2 - \alpha x_1^g\right), \tag{2.2.3b}$$

where $(x_1^g, 0)$ being the centre of the rectangle of dimension $a \times b$ where the gust is generated. The constant K is defined as,

$$K = \delta^g \frac{\alpha\left(\alpha^2 - \hat{a}^2\right) ||\boldsymbol{v}^\infty||^2}{\hat{a}^2 \sqrt{\alpha^2 + \beta^2} \sin\left(\frac{\omega^g \pi}{\hat{a} ||\boldsymbol{v}^\infty||}\right)}$$

where $\hat{a}$, 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

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

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

Chapter 3

The hybridisable discontinuous Galerkin method

"Science is a way of thinking, much more than it is a body of knowledge."

Carl Sagan

Ciaret [22] defines a hybrid method as "any finite element method based on a formulation where one unknown is a function, or some of its derivatives, on the set Ω, and the other unknown is the trace of some of the derivatives of the same function, or the trace of the function itself, along the boundaries of the set K". Fix [44] generalises a method to be hybrid when any kind of duality technique is used to treat troublesome constraints, the use of Lagrange multipliers for handling boundary conditions being one of them. The HDG method was born with the works of Cockburn and Gopalakrishnan [25–28], Cockburn et al. [29], when the idea was applied to relax the continuity requirements on the inter-elemental boundaries.

The HDG framework begins with expressing the approximating variable inside each element and the corresponding flux in-terms of the trace of the approximating variable defined over the element boundaries. Uniqueness of the trace variable is then obtained by imposing the continuity of the normal component of the flux across the inter-element boundaries. This results in only globally coupled degrees of freedom for the solution approximation, which are singularly defined on the element boundaries (faces/edges) only. Efforts have been undertaken to further reduce the duplication of the degrees of freedom with limited success [119].

The HDG method, while retaining the advantages of DG methods over CG methods, offers attractive numerical properties, especially in CFD applications. Compared to the other DG methods analysed in [5], fewer degrees of freedom are duplicated, thus resulting in a significantly smaller discrete system. Both the approximating solution and the flux exhibit optimal convergence properties of the order of h^{k+1}, where h is the characteristic mesh size and k is the order of the polynomial of approximation. Additionally, for second-order PDEs, when approximating the primal and the dual variables in a mixed formulation setting with a polynomial of order k, optimal convergence at a rate of $k+1$ in the $\mathcal{L}_2$ norm is possible. A simple, element-by-element post-processing can then be used to obtain the so-called *super-convergent* solution for the primal variable that converges at the rate of $k+2$ in the $\mathcal{L}_2$ norm. Moreover, for hyperbolic equations, numerical stabilisation can be intuitively added with appropriate definitions of the numerical flux across element faces, wherein, the dependency on only the values of approximating variables on the faces is often advantageous. Numerical fluxes to define both the inviscid and the viscous fluxes in a single framework is also possible. Additionally, a unified treatment of the boundary conditions can be achieved with appropriate definitions of numerical fluxes on the domain boundaries. High accuracy, suitability for anisotropic meshes and degree adaptivity are hallmarks of the HDG method.

The HDG method is a topic of active research with applications for equations of linear convection-diffusion [110], non-linear convection-diffusion [111], Stokes flow [108], incompressible Navier-Stokes flow [109, 52], compressible Euler and Navier-Stokes flow [120], wave propagation [55], linear elasticity [133] and electro-magnetics [93]. See [51] for an overview of the developments in the HDG method and an open-source implementation.

This chapter applies the HDG method to the Euler equations previously described in Chapter 2 and is organised as follows. A brief setting for the HDG framework is provided as preliminaries, wherein, the *broken* HDG domain, the hybrid variable and the relevant functional spaces for the approximating variables are defined. A two-stage HDG weak formulation for the governing equations along with the treatment of the boundary conditions is presented. Additionally, the shock capturing method employed is discussed. The exposition then moves on to present the choice of the temporal and spatial discretisation. The strategy to solve the resulting discrete non-linear system of equations with the Newton-Raphson linearisation method concludes the discussion on HDG. A few notes on the implementation details are provided to supplement the formulations. A short description of the Ringleb flow problem is provided. It is a popular example with an analytical solution to the Euler equations, which is employed to demonstrate the spatial accuracy of discretisation methods. The chapter ends with

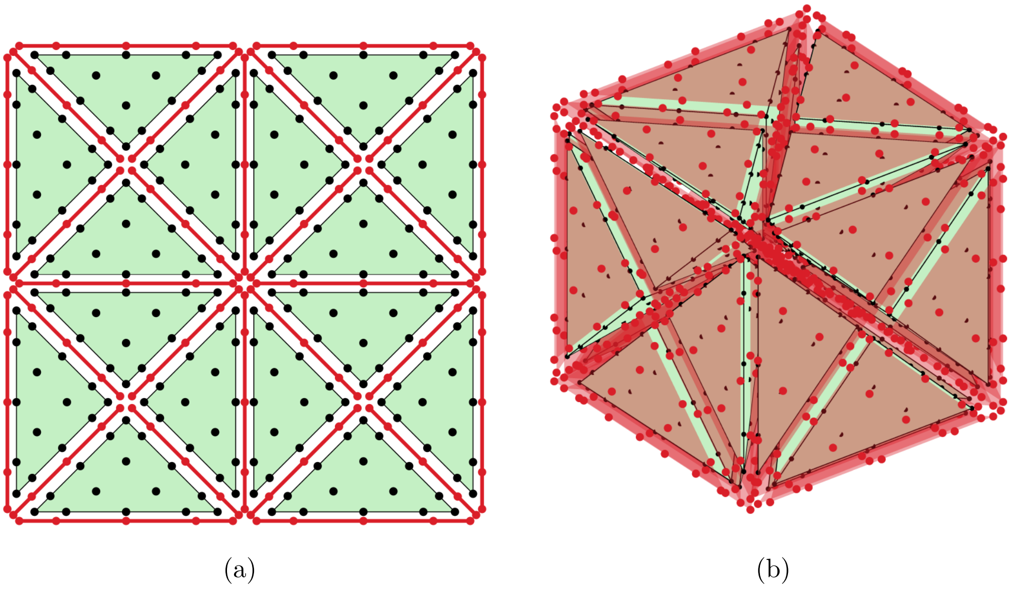


Figure. 3.1 A schematic of the partition of the HDG domain into (a) straight-sided triangular elements in 2D and (b) planar-faced tetrahedral elements in 3D. Fekette nodal distribution of the polynomial of order $k = 4$ for the solution unknowns in the elements (in black) and for the trace of the solution unknowns on the inter-element boundaries (in red) are illustrated.

results from the numerical tests designed to check the methodology for consistency and accuracy.

3.1 Preliminaries

Consider an open bounded domain $\Omega^1 \in \mathbb{R}^{\mathtt{n_{sd}}}$, with boundary $\partial\Omega^1$ and $\mathtt{n_{sd}}$ the number of spatial dimensions, to be discretised with the HDG method. As a first step, assume that Ω^1 is partitioned into $\mathtt{n_{el}}$ disjoint elements Ω^1_e such that,

$$\overline{\Omega}^1 = \bigcup_{e=1}^{\mathtt{n_{el}}} \overline{\Omega}^1_e, \quad \Omega^1_e \cap \Omega^1_f = \emptyset \quad \text{for } e \neq f. \tag{3.1.1}$$

The elemental boundaries $\partial\Omega^1_e$ define the internal interface Γ^1 as,

$$\Gamma^1 := \left[\bigcup_{e=1}^{\mathtt{n_{el}}} \partial\Omega^1_e \right] \setminus \partial\Omega^1. \tag{3.1.2}$$

The partition of the HDG domain is illustrated in Figure 3.1.

The strong form of the governing equations in Equation (2.1.6) applied to the *broken* HDG domain is given as,

$$\boldsymbol{U}_t + \boldsymbol{\nabla} \cdot \boldsymbol{F}(\boldsymbol{U}) = \boldsymbol{S} \qquad \text{in } \Omega_e^1 \times (0,T], \tag{3.1.3a}$$

$$\left.\begin{aligned} \llbracket \rho \boldsymbol{n} \rrbracket &= \mathbf{0}_{\mathtt{n_{sd}}} \\ \llbracket \rho \boldsymbol{v} \otimes \boldsymbol{n} \rrbracket &= \mathbf{0}_{\mathtt{n_{sd}} \times \mathtt{n_{sd}}} \\ \llbracket \rho E \boldsymbol{n} \rrbracket &= \mathbf{0}_{\mathtt{n_{sd}}} \end{aligned}\right\} \Longleftrightarrow \llbracket \boldsymbol{U} \otimes \boldsymbol{n} \rrbracket = \mathbf{0}_{\mathtt{m_{sd}} \times \mathtt{n_{sd}}} \qquad \text{on } \Gamma^1 \times (0,T], \tag{3.1.3b}$$

$$\llbracket \boldsymbol{F}_k(\boldsymbol{U}) n_k \rrbracket = \mathbf{0}_{\mathtt{m_{sd}}} \Longleftrightarrow \llbracket \boldsymbol{F}(\boldsymbol{U}) \cdot \boldsymbol{n} \rrbracket = \mathbf{0}_{\mathtt{m_{sd}}} \qquad \text{on } \Gamma^1 \times (0,T], \tag{3.1.3c}$$

$$\boldsymbol{U} = \boldsymbol{\mathcal{U}}_0 \qquad \text{in } \Omega_e^1 \times \{0\}, \tag{3.1.3d}$$

$$\boldsymbol{B}(\boldsymbol{U}, \boldsymbol{\mathcal{U}}^\infty) = \mathbf{0}_{\mathtt{m_{sd}}} \qquad \text{on } \left(\partial\Omega_e^1 \cap \partial\Omega\right) \times (0,T], \tag{3.1.3e}$$

for $e = 1, \ldots, \mathtt{n_{el}}$, where $\boldsymbol{U} = \boldsymbol{\mathcal{U}}|_{\Omega^1}$, Equations (3.1.3b) and (3.1.3c) impose the continuity of the conservative variable and the normal fluxes respectively, the so-called *transmission* conditions, across the internal interface Γ^1.

The jump operator $\llbracket \cdot \rrbracket$ at the interface, following the definition introduced in [101], sums the values from the right and the left of the interface, say Ω_i and Ω_j as

$$\llbracket \odot \rrbracket = \odot|_{\Omega_i^1} + \odot|_{\Omega_j^1}. \tag{3.1.4}$$

It is to be noted that the corresponding unit outward normal vector $\boldsymbol{n}$ is used on each side of the interface.

The HDG method describes the solution process for Equation (3.1.3) in two phases. Firstly an element-by-element local problem is defined to find $\boldsymbol{U}$ for each element $e = 1, \ldots, \mathtt{n_{el}}$ with $\widehat{\boldsymbol{U}}$, which is the trace of $\boldsymbol{U}$ defined on the mesh skeleton $(\Gamma^1 \cup \partial\Omega^1)$, imposed as Dirichlet condition. Secondly, a global problem is setup to determine the trace $\widehat{\boldsymbol{U}}$. Note that $\widehat{\boldsymbol{U}}$, which is referred to as the hybrid variable, is an independent variable to be solved for. This leads to the final strong form as,

$$\boldsymbol{U}_t + \boldsymbol{\nabla} \cdot \boldsymbol{F}(\boldsymbol{U}) = \boldsymbol{S} \qquad \text{in } \Omega_e^1 \times (0,T], \tag{3.1.5a}$$

$$\boldsymbol{U} = \widehat{\boldsymbol{U}} \qquad \text{on } \partial\Omega_e^1 \times (0,T], \tag{3.1.5b}$$

$$\llbracket \boldsymbol{F}(\boldsymbol{U}) \cdot \boldsymbol{n} \rrbracket = \mathbf{0}_{\mathtt{m_{sd}}} \qquad \text{on } \Gamma^1 \times (0,T], \tag{3.1.5c}$$

$$\boldsymbol{U} = \boldsymbol{\mathcal{U}}_0 \qquad \text{in } \Omega_e^1 \times \{0\}, \tag{3.1.5d}$$

$$\boldsymbol{B}(\boldsymbol{U}, \widehat{\boldsymbol{U}}, \boldsymbol{\mathcal{U}}^\infty) = \mathbf{0}_{\mathtt{m_{sd}}} \qquad \text{on } \left(\partial\Omega_e^1 \cap \partial\Omega\right) \times (0,T], \tag{3.1.5e}$$

for $e = 1, \dots, \mathtt{n_{el}}$. It is worth noting that Equation (3.1.3b) is satisfied by the Dirichlet condition in Equation (3.1.5b) and the unique definition of the hybrid variable on each face of the mesh skeleton.

What follows in this section are some of the definitions that supplement the upcoming HDG formulation. The $\mathcal{L}_2$ inner product in a generic domain D is defined as,

$$(\boldsymbol{u}, \boldsymbol{v})_D = \int_D \boldsymbol{u} \cdot \boldsymbol{v} \, d\Omega. \tag{3.1.6}$$

Similarly, the $\mathcal{L}_2$ inner product in any domain $S \subset (\Gamma^1 \cup \partial\Omega^1)$ is given by,

$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle_S = \int_S \boldsymbol{u} \cdot \boldsymbol{v} \, d\Gamma. \tag{3.1.7}$$

Utilising the notation in [139], the following discontinuous finite element functional spaces are defined,

$$\mathcal{W}^{h,k_e}(\Omega_e^1) := \left\{ v \in \mathcal{L}_2(\Omega^1) : v|_{\Omega_e^1} \in \mathcal{P}^{k_e}(\Omega_e^1) \; \forall \Omega_e^1 , \, e = 1, \dots, \mathtt{n_{el}} \right\}, \tag{3.1.8a}$$

$$\widehat{\mathcal{W}}^{h,k_j}(\Gamma_j^1) := \left\{ \hat{v} \in \mathcal{L}_2(\Gamma^1 \cup \partial\Omega^1) : \hat{v}|_{\Gamma_j^1} \in \mathcal{P}^{k_j}(\Gamma_j^1) \; \forall \Gamma_j^1 \subset \left(\Gamma^1 \cup \partial\Omega^1\right) \right\}, \tag{3.1.8b}$$

$$\mathcal{W}_t^{h,k_e}(\Omega_e^1) := \left\{ v : v(\cdot, t) \in \mathcal{W}^{h,k_e}(\Omega_e^1), \forall t \in (0, T] \right\}, \tag{3.1.8c}$$

$$\widehat{\mathcal{W}}_t^{h,k_j}(\Gamma_j^1) := \left\{ \hat{v} : \hat{v}(\cdot, t) \in \widehat{\mathcal{W}}^{h,k_j}(\Gamma_j^1), \forall t \in (0, T] \right\}, \tag{3.1.8d}$$

where $\mathcal{P}^{k_e}(\Omega_e^1)$ and $\mathcal{P}^{k_j}(\Gamma_j^1)$ are the spaces of polynomial functions of complete degree at most k_e in Ω_e and k_j on Γ_j respectively. Note that while $\mathcal{W}^{h,k_e}$ and $\mathcal{W}_t^{h,k_e}$ are defined on a per-element basis, $\widehat{\mathcal{W}}^{h,k_j}$ and $\widehat{\mathcal{W}}_t^{h,k_j}$ are defined on a per-face (in 3D) or per-edge (in 2D) basis on the mesh skeleton which includes the mesh internal interface and the external faces/edges.

3.2 The HDG weak formulation

The HDG method as the choice of spatial discretisation, introduced in the previous section, is elaborated here. The strong form in Equation (3.1.5) is solved in two phases: the so-called local and the global problems [29, 24, 110]. Following the application of the method of lines [39], a semi-discrete weak form is derived.

3.2.1 The local problem

The weak form of the local problem involves the integration of the Equation (3.1.5a) in each element Ω_e^1. The problem can be stated as: given $\widehat{\boldsymbol{U}}$ on $(\Gamma^1 \cup \partial\Omega^1)$, find $\boldsymbol{U} \in [\mathcal{W}_t^{h,k_e}(\Omega_e^1)]^{\mathtt{m_{sd}}}$ such that,

$$\Big(\delta\boldsymbol{U}, \boldsymbol{U}_t\Big)_{\Omega_e^1} + \Big(\delta\boldsymbol{U}, \boldsymbol{\nabla\cdot F}(\boldsymbol{U})\Big)_{\Omega_e^1} = \Big(\delta\boldsymbol{U}, \boldsymbol{S}\Big)_{\Omega_e^1}, \tag{3.2.1}$$

for all $\delta\boldsymbol{U} \in [\mathcal{W}^{h,k_e}(\Omega_e^1)]^{\mathtt{m_{sd}}}$ and $e = 1, \dots, \mathtt{n_{el}}$. As usual with the Galerkin method, the spaces of test functions are chosen to be the space spanned by the approximations. Applying the divergence theorem the local problem becomes,

$$\Big(\delta\boldsymbol{U}, \boldsymbol{U}_t\Big)_{\Omega_e^1} - \Big(\boldsymbol{\nabla}\delta\boldsymbol{U}, \boldsymbol{F}(\boldsymbol{U})\Big)_{\Omega_e^1} + \Big\langle\delta\boldsymbol{U}, \boldsymbol{F}(\boldsymbol{U})\cdot\boldsymbol{n}\Big\rangle_{\partial\Omega_e^1} = \Big(\delta\boldsymbol{U}, \boldsymbol{S}\Big)_{\Omega_e^1}. \tag{3.2.2}$$

Approximating the normal flux on the element boundaries with a numerical normal flux, we introduce a dependency on the trace of the solution $\widehat{\boldsymbol{U}}$ as,

$$\widehat{\boldsymbol{F}}(\boldsymbol{U},\widehat{\boldsymbol{U}})\cdot\boldsymbol{n} := \boldsymbol{F}(\widehat{\boldsymbol{U}})\cdot\boldsymbol{n} + \boldsymbol{\tau}\cdot(\boldsymbol{U} - \widehat{\boldsymbol{U}}), \tag{3.2.3}$$

where, the selection of the stabilisation tensor $\boldsymbol{\tau}$ plays an important role in the stability, accuracy and convergence properties of the resulting HDG method [29, 24, 110, 111, 108, 112]. As noted earlier, the continuity in the conservative variables are introduced through the definition of a unique value of the hybrid variable. The definition in Equation (3.2.3) is consistent, that is, when $\boldsymbol{U} = \widehat{\boldsymbol{U}}$, $\widehat{\boldsymbol{F}}\cdot\boldsymbol{n} = \boldsymbol{F}\cdot\boldsymbol{n}$. The condition imposed penalises the continuity constraint through the stabilisation term. The weak imposition thus ensures the balance of the fluxes across element boundaries, although not necessarily in a point-wise manner, but in its $\mathcal{L}_2$ projection onto $\widehat{\mathcal{W}}^{h,k_j}(\Gamma_j^1)$. Hence, any jumps or sharp gradients in quantities such as density across a shock, which are allowed by the physics, may be accounted without the loss of stability in the scheme. The flexibility of defining the stabilisation tensor provides an advantage in tailoring it based on the physics of the problem being solved, see [110, 111]. Various definitions have been explored for Euler equations [120, 155], some of which are listed below:

$$\boldsymbol{\tau}^{\mathrm{ROE}}\left(\widehat{\boldsymbol{U}}\right) := \boldsymbol{L}\left(\widehat{\boldsymbol{U}}\right)|\boldsymbol{\Lambda}_n\left(\widehat{\boldsymbol{U}}\right)|\boldsymbol{R}\left(\widehat{\boldsymbol{U}}\right), \tag{3.2.4a}$$

$$\boldsymbol{\tau}^{\mathrm{LLF}}\left(\widehat{\boldsymbol{U}}\right) := \left(|\hat{v}_n| + \hat{c}\right)\mathbf{I}_{\mathtt{m_{sd}}}, \tag{3.2.4b}$$

$$\boldsymbol{\tau}^{\mathrm{GLF}}\left(\boldsymbol{\mathcal{U}}^{\infty}\right) := \left(||\boldsymbol{v}^{\infty}|| + c^{\infty}\right)\mathbf{I}_{\mathtt{m_{sd}}}. \tag{3.2.4c}$$

Equation (3.2.4a) results in a Roe's scheme for the numerical flux, where, $\boldsymbol{\Lambda}_n\left(\widehat{\boldsymbol{U}}\right)$, given in Equation (2.1.13), is the diagonal matrix of eigenvalues of normal convective jacobian $\boldsymbol{A}_n\left(\widehat{\boldsymbol{U}}\right)$ in Equation 2.1.11. $\boldsymbol{L}\left(\widehat{\boldsymbol{U}}\right)$ and $\boldsymbol{R}\left(\widehat{\boldsymbol{U}}\right)$ are the corresponding matrices of left and right eigenvectors given in Equation (2.1.14) and Equation (2.1.12). $|\boldsymbol{\Lambda}_n\left(\widehat{\boldsymbol{U}}\right)|$ has all the diagonal entries as absolute values of that corresponding to the entries of $\boldsymbol{\Lambda}_n\left(\widehat{\boldsymbol{U}}\right)$. Equation (3.2.4b) leads to a Local Lax-Friedrichs scheme, where the local normal velocity $\hat{v}_n = \hat{\boldsymbol{v}} \cdot \boldsymbol{n}$ and local speed of sound $\hat{c}$ are computed using the hybrid variable $\widehat{\boldsymbol{U}}$. Equation (3.2.4c) defines a global Lax-Freidrichs method where free-stream value of the conservative variable $\boldsymbol{\mathcal{U}}^{\infty}$ is utilised.

For its ease of computation and suitable accuracy, $\boldsymbol{\tau}^{\text{LLF}}$ has been used throughout in the present work. Moreover it has been reported [120] that the choice of the stabilization matrix becomes less critical for high-order Discontinuous Galerkin methods as numerical dissipation of the order h^{k+1} vanishes rapidly with increasing k.

Finally, the semi-discrete weak form of the local problem is: given $\widehat{\boldsymbol{U}}$ on $(\Gamma^1 \cup \partial\Omega^1)$, find $\boldsymbol{U} \in [\mathcal{W}_t^{h,k_e}(\Omega_e^1)]^{\mathtt{msd}}$ such that,

$$\Big(\delta\boldsymbol{U}, \boldsymbol{U}_t\Big)_{\Omega_e^1} - \Big(\boldsymbol{\nabla}\delta\boldsymbol{U}, \boldsymbol{F}(\boldsymbol{U})\Big)_{\Omega_e^1} + \Big\langle\delta\boldsymbol{U}, \widehat{\boldsymbol{F}}(\boldsymbol{U}, \widehat{\boldsymbol{U}}) \cdot \boldsymbol{n}\Big\rangle_{\partial\Omega_e^1} = \Big(\delta\boldsymbol{U}, \boldsymbol{S}\Big)_{\Omega_e^1}, \tag{3.2.5}$$

for all $\delta\boldsymbol{U} \in [\mathcal{W}^{h,k_e}(\Omega_e^1)]^{\mathtt{msd}}$ and $e = 1, \dots, \mathtt{n_{el}}$.

3.2.2 The global problem

The global problem solves for the hybrid variable globally. Utilising the definition of the numerical normal flux in Equation (3.1.5c) for the balance of normal fluxes on the mesh skeleton and Equation (3.1.3e), the global problem is given as: find $\widehat{\boldsymbol{U}} \in [\widehat{\mathcal{W}}_t^{h,k_j}(\Gamma^1 \cup \partial\Omega^1)]^{\mathtt{msd}}$ such that

$$\sum_{e=1}^{\mathtt{n_{el}}} \left\{ \Big\langle\delta\widehat{\boldsymbol{U}}, \widehat{\boldsymbol{F}}(\boldsymbol{U}, \widehat{\boldsymbol{U}}) \cdot \boldsymbol{n}\Big\rangle_{\partial\Omega_e^1 \setminus \partial\Omega^1} + \Big\langle\delta\widehat{\boldsymbol{U}}, \widehat{\boldsymbol{B}}(\boldsymbol{U}, \widehat{\boldsymbol{U}}, \boldsymbol{\mathcal{U}}^{\infty})\Big\rangle_{\partial\Omega_e^1 \cap \partial\Omega^1} \right\} = 0, \tag{3.2.6}$$

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

The local problem, given by Equation (3.2.5) is used to write the primal solution $\boldsymbol{U}$ as a function of the hybrid variable $\widehat{\boldsymbol{U}}$. Introducing this expression into the global problem, given by Equation (3.2.6), leads to a problem only for the hybrid variable.

3.2.3 Boundary conditions

Treatment of the boundary conditions, given in form of the boundary flux in Equation (3.1.5e), is of a vital concern [15]. For external flow phenomena, generally two distinct conditions arise, the free-stream/far-field boundary and the inviscid/slip wall boundary [100]. Special considerations are necessary because of the presence of the hybrid variable $\widehat{\boldsymbol{U}}$. Various approaches in the HDG context are available, see [120, 76]. The following definitions of the boundary fluxes are referenced from [120].
The far-field boundary receives information from the external free-stream conditions (prescribed) or the domain interior values based on the flow characteristics, which is taken into account with the boundary flux defined as,

$$\widehat{\boldsymbol{B}}^{\infty}(\boldsymbol{U},\widehat{\boldsymbol{U}},\boldsymbol{\mathcal{U}}^{\infty}) = \boldsymbol{A}_n^{+}\left(\widehat{\boldsymbol{U}}\right)\left(\boldsymbol{U}-\widehat{\boldsymbol{U}}\right) - \boldsymbol{A}_n^{-}\left(\widehat{\boldsymbol{U}}\right)\left(\boldsymbol{\mathcal{U}}^{\infty}-\widehat{\boldsymbol{U}}\right). \tag{3.2.7}$$

Here, the jacobian matrices associated with positive and negative characteristics $\boldsymbol{A}_n^{\pm}\left(\widehat{\boldsymbol{U}}\right)$ is given in Equation (2.1.21).
At the inviscid wall boundary, the flow is assumed to 'slip', wherein the momentum is tangential to the boundary. In other words, the condition imposes zero normal component to the momentum equations as below,

$$\widehat{\boldsymbol{B}}^{W}(\boldsymbol{U},\widehat{\boldsymbol{U}}) = \boldsymbol{b}^{W}\boldsymbol{U} - \widehat{\boldsymbol{U}}, \tag{3.2.8}$$

where, the $\mathtt{m_{sd}} \times \mathtt{m_{sd}}$ matrix $\boldsymbol{b}^W$ is,

$$\boldsymbol{b}^W = \begin{bmatrix} 1 & \mathbf{0}_{\mathtt{n_{sd}}}^T & 0 \\ 0 & \mathbf{I}_{\mathtt{n_{sd}}} - \boldsymbol{n}\otimes\boldsymbol{n} & 0 \\ 0 & \mathbf{0}_{\mathtt{n_{sd}}}^T & 1 \end{bmatrix}. \tag{3.2.9}$$

3.3 Shock capturing method

High speed flows governed by the Euler equations admit regions of sharp gradients in its solution which take the form of shock waves and contact discontinuities. Special treatment is required for the numerical approximation techniques in the vicinity of such discontinuities in order to avoid excessive Gibbs oscillations appearing in the solution and to thus maintain the stability of the discretisation method.
Two classes of methods, namely, slope limiters and artificial dissipation methods are generally available. The current work adopts the latter for resolving the shocks.

Slope limiting methods reconstruct the solution in the regions detected as shocks with a low order approximation [33, 82, 95]. They have not been widely adopted for practical applications because of the restrictions on the time step size which make them unsuitable for implicit methods. Artificial dissipation methods on the other hand rely on applying additional viscous terms to the equations in the region of the shocks in order to smear the discontinuity across a length scale such that they may then be resolved by the approximating functions [158, 73, 74]. For high-order DG methods it is argued that the high accuracy afforded by the high-order approximating functions allow for sub-cell resolution of shocks that are thinner than the element size [122, 11]. A key ingredient in the shock capturing methods is the so-called shock sensor, which determines the location and quantifies the magnitude of the shock. Previously, a sensor based on the rate of decay of the expansion coefficients of the polynomials of approximation in every element has been used [122]. In the current work, the sensor is based on a non-dimensional form of the divergence of the velocity field (dilatation) introduced in [107] and later expanded in [104]. The performance of the dilatation-based sensor is shown to be robust for complex flows in highly anisotropic meshes

The artificial dissipation method introduces a discrete form of the Laplacian-type term $\boldsymbol{\nabla}\cdot\left(\kappa\boldsymbol{\nabla}\boldsymbol{U}\right)$ into the governing equations, which gets activated in the region of the shocks. Here, κ determines the amount of artificial viscosity added and is a function of the dilatation, mesh size and state of the fluid. The mesh size is given by a reference length scale equal to $\left(l_h h\left(\boldsymbol{x}\right)\right)/k$, where the constant $l_h = 1.5$ and $h\left(\boldsymbol{x}\right)$ is the piecewise linear reconstruction of the element size, which is obtained by the averaging of element sizes surrounding a vertex. The non-dimensional, dilatation-based sensor s is given by,

$$s = -\left(\frac{l_h h\left(\boldsymbol{x}\right)}{k}\right)\frac{\boldsymbol{\nabla}\cdot\boldsymbol{v}}{c^*}, \tag{3.3.1}$$

where, the critical speed of sound c^* is,

$$c^* = c\sqrt{\frac{2+\left(\gamma-1\right)M^2}{\gamma+1}}. \tag{3.3.2}$$

Finally, κ is given as,

$$\kappa = -\left(\frac{l_h h\left(\boldsymbol{x}\right)}{k}\right)\sqrt{\boldsymbol{v}\cdot\boldsymbol{v}+c^2}\ f(s). \tag{3.3.3}$$

A smoothing $f(s)$ is applied to the ramp function, $\max(0, s-b)$, to avoid convergence issues as,

$$f\left(s\right) = \frac{\log\left(1+e^{a(s-b)}\right)}{a}, \tag{3.3.4}$$

where (a) controls the shape of the smoothing and (b) determines the value beyond which the sensor ramps up. Their values, $a = 10^4$ and $b = 0.01$, are obtained from [104].

Thus, the semi-discrete weak form of the local problem in Equation (3.2.5) is updated as: given $\widehat{\boldsymbol{U}}$ on $(\Gamma^1 \cup \partial\Omega^1)$, find $\boldsymbol{U} \in [\mathcal{W}_t^{h,k_e}(\Omega_e^1)]^{\mathtt{m_{sd}}}$ such that,

$$\left(\delta\boldsymbol{U},\boldsymbol{U}_t\right)_{\Omega_e^1} - \left(\boldsymbol{\nabla}\delta\boldsymbol{U},\boldsymbol{F}(\boldsymbol{U})\right)_{\Omega_e^1} + \left\langle\delta\boldsymbol{U},\widehat{\boldsymbol{F}}(\boldsymbol{U},\widehat{\boldsymbol{U}})\cdot\boldsymbol{n}\right\rangle_{\partial\Omega_e^1} = \left(\delta\boldsymbol{U},\boldsymbol{S}\right)_{\Omega_e^1} - \left(\boldsymbol{\nabla}\delta\boldsymbol{U},\kappa\boldsymbol{\nabla}\boldsymbol{U}\right)_{\Omega_e^1}, \tag{3.3.5}$$

for all $\delta\boldsymbol{U} \in [\mathcal{W}^{h,k_e}(\Omega_e^1)]^{\mathtt{m_{sd}}}$ and $e = 1,\ldots,\mathtt{n_{el}}$.

3.4 Temporal discretisation

The semi-discrete weak forms of the local problem in Equation (3.3.5) and the global problem in Equation (3.2.6) represents a differential algebraic system of equations [59, 6]. Implicit Runge-Kutta method, in particular diagonally implicit R-K (DIRK) methods have been previously used for HDG [77, 76].

Currently, the time integration is performed using backward differentiation formulae (BDF) [6]. This popular family of implicit multi-step time marching algorithms approximate the first-order time derivative as

$$u_t \approx \sum_{s=0}^{\mathtt{n_{tg}}} a_s u^{n+1-s} \tag{3.4.1}$$

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 $\mathtt{n_{tg}} = 1$, $a_0 = 1/\Delta t$ and $a_1 = -1/\Delta t$. For transient computations, a second-order BDF method (BDF2) is employed, corresponding to $\mathtt{n_{tg}} = 2$, $a_0 = 3/(2\Delta t)$, $a_1 = -2/\Delta t$ and $a_2 = 1/(2\Delta t)$. Note that only BDF1 and BDF2 are A-stable.

The semi-discrete weak formulation for HDG with BDF time integration can be written as,

$$
\begin{aligned}
&\sum_{s=0}^{\mathtt{n_{tg}}} a_s\Big(\delta\boldsymbol{U},\boldsymbol{U}^{n+1-s}\Big)_{\Omega_e^1} - \Big(\boldsymbol{\nabla}\delta\boldsymbol{U},\boldsymbol{F}(\boldsymbol{U}^{n+1})\Big)_{\Omega_e^1} + \Big\langle\delta\boldsymbol{U},\boldsymbol{F}(\widehat{\boldsymbol{U}}^{n+1})\cdot\boldsymbol{n}\Big\rangle_{\partial\Omega_e^1}\\
&\quad + \Big\langle\delta\boldsymbol{U},\boldsymbol{\tau}^{n+1}\cdot\boldsymbol{U}^{n+1}\Big\rangle_{\partial\Omega_e^1} - \Big\langle\delta\boldsymbol{U},\boldsymbol{\tau}^{n+1}\cdot\widehat{\boldsymbol{U}}^{n+1}\Big\rangle_{\partial\Omega_e^1} - \Big(\delta\boldsymbol{U},\boldsymbol{S}^{n+1}\Big)_{\Omega_e^1}\\
&\quad + \Big(\boldsymbol{\nabla}\delta\boldsymbol{U},\kappa^n\boldsymbol{\nabla}\boldsymbol{U}^{n+1}\Big)_{\Omega_e^1} = 0, \qquad (3.4.2a)\\
&\sum_{e=1}^{\mathtt{n_{el}}}\Big\{\Big\langle\delta\widehat{\boldsymbol{U}},\boldsymbol{F}(\widehat{\boldsymbol{U}}^{n+1})\cdot\boldsymbol{n}\Big\rangle_{\partial\Omega_e^1\backslash\partial\Omega^1} + \Big\langle\delta\widehat{\boldsymbol{U}},\boldsymbol{\tau}^{n+1}\cdot\boldsymbol{U}^{n+1}\Big\rangle_{\partial\Omega_e^1\backslash\partial\Omega^1}\\
&\quad - \Big\langle\delta\widehat{\boldsymbol{U}},\boldsymbol{\tau}^{n+1}\cdot\widehat{\boldsymbol{U}}^{n+1}\Big\rangle_{\partial\Omega_e^1\backslash\partial\Omega^1} + \Big\langle\delta\widehat{\boldsymbol{U}},\widehat{\boldsymbol{B}}(\boldsymbol{U}^{n+1},\widehat{\boldsymbol{U}}^{n+1},\boldsymbol{\mathcal{U}}^{\infty})\Big\rangle_{\partial\Omega_e^1\cap\partial\Omega^1}\Big\} = 0. \qquad (3.4.2b)
\end{aligned}
$$

3.5 Spatial discretisation

The definitions of functional spaces in Equations (3.1.8) lead to the following nodal interpolations of the conservative variable on an element-by-element and face-by-face basis. Solution $\boldsymbol{U}$ is defined on a reference element, with local coordinates $\boldsymbol{\xi} = (\xi_1, \dots, \xi_{\mathtt{n_{sd}}})$ as, for $\boldsymbol{U}(\boldsymbol{\xi}, t) \in \mathcal{W}_t^{h,k_e}(\Omega_e^1)$,

$$
\boldsymbol{U}(\boldsymbol{\xi},t) \approx \boldsymbol{U}_h(\boldsymbol{\xi},t) := \sum_{J=1}^{\mathtt{n_{en}}} N_J(\boldsymbol{\xi})\,\mathbf{U}_J(t), \qquad (3.5.1)
$$

for all $e = 1, \dots, \mathtt{n_{el}}$. Here, $\mathbf{U}_J$ is the nodal value of $\boldsymbol{U}$, N_J is the nodal polynomial shape function of order k_e and $\mathtt{n_{en}}$ is the number of element nodes.

Analogously, the hybrid variable $\widehat{\boldsymbol{U}}$ is approximated on a reference face, with local coordinates $\widehat{\boldsymbol{\xi}} = (\widehat{\xi}_1, \dots, \widehat{\xi}_{\mathtt{n_{sd}}-1})$ as, for $\widehat{\boldsymbol{U}}(\widehat{\boldsymbol{\xi}}, t) \in \widehat{\mathcal{W}}^{h,k_j}(\Gamma_j^1)$,

$$
\widehat{\boldsymbol{U}}(\widehat{\boldsymbol{\xi}},t) \approx \widehat{\boldsymbol{U}}_h(\widehat{\boldsymbol{\xi}},t) := \sum_{J=1}^{\mathtt{n_{fn}}} \widehat{N}_J(\widehat{\boldsymbol{\xi}})\widehat{\mathbf{U}}_J(t), \qquad (3.5.2)
$$

for all $\Gamma_j^1 \subset (\Gamma^1 \cup \partial\Omega^1)$, where $\widehat{\mathbf{U}}_J$ is the nodal value of $\boldsymbol{U}$, $\widehat{N}_J$ is the nodal polynomial shape function of order k_j and $\mathtt{n_{fn}}$ is the number of face nodes.

A standard isoparametric formulation [143] is considered, where local and Cartesian coordinates are linked via the so-called isoparametric mapping

$$\boldsymbol{x}(\boldsymbol{\xi}) = \sum_{J=1}^{\mathtt{n_{en}}} N_J(\boldsymbol{\xi})\, \mathbf{x}_J, \tag{3.5.3a}$$

$$\boldsymbol{x}(\widehat{\boldsymbol{\xi}}) = \sum_{J=1}^{\mathtt{n_{fn}}} \widehat{N}_J(\widehat{\boldsymbol{\xi}})\, \mathbf{x}_J, \tag{3.5.3b}$$

where $\{\mathbf{x}_J\}_{J=1,\dots,\mathtt{n_{en}} \text{ or } \mathtt{n_{fn}}}$ are the nodal coordinates of a generic element or face. Naturally, this leads to an element-by-element definition for $\boldsymbol{U}$ and a global definition over the mesh skeleton for $\widehat{\boldsymbol{U}}$. Consequently, for each element $e = 1, \dots, \mathtt{n_{el}}$, the nodal values $\mathbf{U}$ as described in Equation (3.5.1) are of the size $\mathtt{m_{sd}}\mathtt{n_{en}}$. And nodal values $\widehat{\mathbf{U}}$ described in Equation (3.5.2) is of the size,

$$\dim\left(\widehat{\mathbf{U}}\right) = \left(\sum_{l=1}^{\mathtt{n_{ef}}} \mathtt{n_{fn}}|_l\right) \mathtt{m_{sd}}, \tag{3.5.4}$$

where, $\mathtt{n_{ef}}$ is the number of element faces/edges on the mesh skeleton and $\mathtt{n_{fn}}|_l$ is the number of face nodes on the l-th face, which in-turn depends on the order of polynomial approximation.

Remark 1 (Polynomial approximation in elements)**.** *The order of the polynomial of approximation for each element is not restricted to be the same; in fact they can be chosen arbitrarily and independent of each other. The flexibility of uneven degree afforded by the discontinuous framework facilitates the tailoring of solution accuracy in the regions of interest. This facility has been previously used to devise degree adaptive algorithms, see [53, 54, 140, 50].*

Remark 2 (Polynomial approximation on the mesh skeleton)**.** *One of the consequences of the variable polynomial degree for elements is the necessity to choose the appropriate polynomial degree for the mesh skeleton. For each face/edge on the skeleton shared by elements of degree k_e and k_f, the largest of the two, $max(k_e, k_f)$, is chosen for the nodal approximation. This choice ensures that, for each element e, all of its faces have a degree of at least k_e, thus preserving the accuracy of approximation in each element.*

3.6 Discrete system

Introducing the discrete approximations of $\boldsymbol{U}$ in Equation (3.5.1) and $\widehat{\boldsymbol{U}}$ in Equation (3.5.2) into the semi-discrete weak forms given in Equation (3.4.2) result in a discrete non-linear system of equations. The non-linear system of equations are given as,

$$\boldsymbol{\mathcal{R}}(\boldsymbol{U}_h^{n+1},\dots,\boldsymbol{U}_h^{n+1-\mathtt{n_{tg}}},\widehat{\boldsymbol{U}}_h^{n+1}) = \boldsymbol{0}, \tag{3.6.1}$$

obtained from the assembly of contributions to the local and global system of equations, that is,

$$\boldsymbol{\mathcal{R}}^e := \begin{Bmatrix} \boldsymbol{\mathcal{R}}^e(\boldsymbol{U}_h^{n+1},\dots,\boldsymbol{U}_h^{n+1-\mathtt{n_{tg}}},\widehat{\boldsymbol{U}}_h^{n+1}) \\ \widehat{\boldsymbol{\mathcal{R}}}^e(\boldsymbol{U}_h^{n+1},\dots,\boldsymbol{U}_h^{n+1-\mathtt{n_{tg}}},\widehat{\boldsymbol{U}}_h^{n+1}) \end{Bmatrix}, \tag{3.6.2}$$

where,

$$\begin{aligned}
\boldsymbol{\mathcal{R}}_I^e :=& \sum_{s=0}^{\mathtt{n_{tg}}} a_s\Big(N_I, \boldsymbol{U}_h^{n+1-s}\Big)_{\Omega_e^1} - \Big(\boldsymbol{F}(\boldsymbol{U}_h^{n+1}), \boldsymbol{\nabla}^T N_I\Big)_{\Omega_e^1} + \Big\langle N_I, \boldsymbol{F}(\widehat{\boldsymbol{U}}_h^{n+1})\cdot\boldsymbol{n}\Big\rangle_{\partial\Omega_e^1} \\
&+ \Big\langle N_I, \boldsymbol{\tau}^{n+1}\cdot\boldsymbol{U}_h^{n+1}\Big\rangle_{\partial\Omega_e^1} - \Big\langle N_I, \boldsymbol{\tau}^{n+1}\cdot\widehat{\boldsymbol{U}}_h^{n+1}\Big\rangle_{\partial\Omega_e^1} - \Big(N_I, \boldsymbol{S}^{n+1}\Big)_{\Omega_e^1} \\
&+ \Big(\kappa^n \boldsymbol{\nabla}\boldsymbol{U}_h^{n+1}, \boldsymbol{\nabla}^T N_I\Big)_{\Omega_e^1} = 0, && \text{(3.6.3a)} \\
\widehat{\boldsymbol{\mathcal{R}}}_I^e :=& \sum_{e=1}^{\mathtt{n_{el}}} \Big\{ \Big\langle \widehat{N}_I, \boldsymbol{F}(\widehat{\boldsymbol{U}}_h^{n+1})\cdot\boldsymbol{n}\Big\rangle_{\partial\Omega_e^1\setminus\partial\Omega^1} + \Big\langle \widehat{N}_I, \boldsymbol{\tau}^{n+1}\cdot\boldsymbol{U}_h^{n+1}\Big\rangle_{\partial\Omega_e^1\setminus\partial\Omega^1} \\
&- \Big\langle \widehat{N}_I, \boldsymbol{\tau}^{n+1}\cdot\widehat{\boldsymbol{U}}_h^{n+1}\Big\rangle_{\partial\Omega_e^1\setminus\partial\Omega^1} + \Big\langle \widehat{N}_I, \widehat{\boldsymbol{B}}(\boldsymbol{U}_h^{n+1},\widehat{\boldsymbol{U}}_h^{n+1},\boldsymbol{\mathcal{U}}^\infty)\Big\rangle_{\partial\Omega_e^1\cap\partial\Omega^1}\Big\} = 0, && \text{(3.6.3b)}
\end{aligned}$$

for $e = 1,\dots,\mathtt{n_{el}}$.

3.7 Linearisation method

The Newton-Raphson method is applied to linearise the non-linear residual of Equation (3.6.1) and, by truncating the Taylor expansion at first order, the non-symmetric sparse linear system to be solved at each iteration (q) of the Newton-Raphson is obtained, namely

$$\begin{bmatrix} \mathbf{T}_{uu} & \mathbf{T}_{u\hat{u}} \\ \mathbf{T}_{\hat{u}u} & \mathbf{T}_{\hat{u}\hat{u}} \end{bmatrix}^{n+1,q} \begin{Bmatrix} \Delta\mathrm{U} \\ \Delta\widehat{\mathrm{U}} \end{Bmatrix}^{n+1,q} = \begin{Bmatrix} \mathbf{f}_u \\ \mathbf{f}_{\hat{u}} \end{Bmatrix}^{n+1,q}, \tag{3.7.1}$$

where $\Delta\circledcirc^{n+1,q} = \circledcirc^{n+1,q+1} - \circledcirc^{n+1,q}$ denotes the increment of the vector of degrees of freedom. The detailed expression of the tangent matrices $\mathbf{T}_{uu}$, $\mathbf{T}_{u\hat{u}}$, $\mathbf{T}_{\hat{u}u}$ and $\mathbf{T}_{\hat{u}\hat{u}}$ and the right hand side vectors $\mathbf{f}_u$ and $\mathbf{f}_{\hat{u}}$ result from the linearisation using a

Newton-Raphson method, that is,

$$(\mathbf{T}_{uu})_{IJ}^{e,n+1,q} := \frac{\partial \boldsymbol{\mathcal{R}}_I^e}{\partial \mathbf{U}_J^{n+1,q}}, \qquad (\mathbf{T}_{u\hat{u}})_{IJ}^{e,n+1,q} := \frac{\partial \boldsymbol{\mathcal{R}}_I^e}{\partial \widehat{\mathbf{U}}_J^{n+1,q}}, \tag{3.7.2a}$$

$$(\mathbf{T}_{\hat{u}u})_{IJ}^{e,n+1,q} := \frac{\partial \widehat{\boldsymbol{\mathcal{R}}}_I^e}{\partial \mathbf{U}_J^{n+1,q}}, \qquad (\mathbf{T}_{\hat{u}\hat{u}})_{IJ}^{e,n+1,q} := \frac{\partial \widehat{\boldsymbol{\mathcal{R}}}_I^e}{\partial \widehat{\mathbf{U}}_J^{n+1,q}}, \tag{3.7.2b}$$

and

$$(\mathbf{f}_u^e)_I^{n+1,q} := -\boldsymbol{\mathcal{R}}_I^e, \qquad (\mathbf{f}_{\hat{u}}^e)_I^{n+1,q} := -\widehat{\boldsymbol{\mathcal{R}}}_I^e. \tag{3.7.3a}$$

The block diagonal structure of $\mathbf{T}_{uu}$ makes the linear system in Equation (3.7.1) amenable for static condensation, resulting in a reduced system as follows,

$$\left[\widetilde{\mathbf{T}}_{\hat{u}\hat{u}}\right]^{n+1,q} \left\{\Delta\widehat{\mathbf{U}}\right\}^{n+1,q} = \left\{\tilde{\mathbf{f}}_{\hat{u}}\right\}^{n+1,q}, \tag{3.7.4}$$

where,

$$\widetilde{\mathbf{T}}_{\hat{u}\hat{u}} := \mathbf{T}_{\hat{u}\hat{u}} - \mathbf{T}_{\hat{u}u}\mathbf{T}_{uu}^{-1}\mathbf{T}_{u\hat{u}}, \tag{3.7.5a}$$

$$\tilde{\mathbf{f}}_{\hat{u}} := \mathbf{f}_{\hat{u}} - \mathbf{T}_{\hat{u}u}\mathbf{T}_{uu}^{-1}\mathbf{f}_u. \tag{3.7.5b}$$

Solving the linear system for $\Delta\widehat{\mathbf{U}}^{n+1,q}$, the solution in the element interior $\Delta\mathbf{U}^{n+1,q}$ can then be recovered element-by-element using,

$$[\mathbf{T}_{uu}]^{n+1,q}\{\Delta\mathbf{U}\}^{n+1,q} = \{\mathbf{f}_u\}^{n+1,q} - [\mathbf{T}_{u\hat{u}}]^{n+1,q}\Delta\left\{\widehat{\mathbf{U}}\right\}^{n+1,q}. \tag{3.7.6}$$

The number of contributions to the tangent matrix given by the Equation (3.7.4) for any node in the mesh skeleton depends on the dimensionality of the problem $\mathtt{m_{sd}}$ and the degree of approximation k. However, for an individual node on the mesh skeleton, non-zero contributions are only made by those nodes that lie on the mesh skeleton faces of the neighbouring elements. In a 2D problem with triangular elements, the tangent matrix for each of the $\mathtt{m_{sd}} = 4$ unknown degrees of freedom at a node, for $k = 1$ to $k = 4$ degree of approximations, receive a total of $40, 50, 80$ and 100 non-zero contributions respectively. Similarly, for 3D problems with tetrahedral elements, the tangent matrix for each of the $\mathtt{m_{sd}} = 5$ degrees of freedom at a node, for $k = 1$ to $k = 4$ degrees of approximation, receive a total of $105, 210, 350$ and 525 non-zero contributions respectively.

Remark 3 (Imposition of initial conditions)**.** *Time marching requires the computation of the discrete initial approximations of the solutions* $\boldsymbol{U}_h^0$ *and* $\widehat{\boldsymbol{U}}_h^0$*. For spatially continuous fields, the initial condition in Equation* (3.1.5d) *is imposed strongly in a straightforward manner. In the case of spatially discontinuous fields, for convergence and stability of the HDG method both* $\boldsymbol{U}^0$ *and* $\widehat{\boldsymbol{U}}^0$ *need to be in equilibrium. In other words, given* $\widehat{\boldsymbol{U}}^0$ *on* $(\Gamma^1 \cup \partial\Omega^1)$*, find* $\boldsymbol{U}^0 \in [\mathcal{W}^{h,k_e}(\Omega_e^1)]^{\mathtt{m_{sd}}}$ *such that,*

$$\left\langle \delta\widehat{\boldsymbol{U}}, \boldsymbol{\tau}\left(\boldsymbol{U}^0 - \widehat{\boldsymbol{U}}^0\right)\right\rangle_{\partial\Omega_e^1\setminus\partial\Omega} = 0, \tag{3.7.7}$$

for all $\delta\widehat{\boldsymbol{U}} \in [\widehat{\mathcal{W}}^{h,k_j}(\Gamma^1 \cup \partial\Omega^1)]^{\mathtt{m_{sd}}}$ *and* $e = 1, \dots, \mathtt{n_{el}}$*. This critical procedure is necessary whenever solutions are transferred between different 'k-maps', such as during a degree adaptivity procedure.*

3.8 HDG vectors and matrices

The current section is dedicated to detailing the expressions related to the matrices and vectors appearing in the HDG formulation for both the global and the local problems. Following the spatial discretisation introduced in Section 3.5, consider the compact form of the interpolating functions,

$$\mathbf{N} = \begin{bmatrix} N_1\mathbf{I}_{\mathtt{m_{sd}}} & N_2\mathbf{I}_{\mathtt{m_{sd}}} & \dots & N_{\mathtt{n_{en}}}\mathbf{I}_{\mathtt{m_{sd}}} \end{bmatrix}^T, \tag{3.8.1a}$$

$$\widehat{\mathbf{N}} = \begin{bmatrix} \widehat{N}_1\mathbf{I}_{\mathtt{m_{sd}}} & \widehat{N}_2\mathbf{I}_{\mathtt{m_{sd}}} & \dots & \widehat{N}_{\mathtt{n_{fn}}}\mathbf{I}_{\mathtt{m_{sd}}} \end{bmatrix}^T, \tag{3.8.1b}$$

where $\mathbf{N}$ is of size $\mathtt{n_{en}}\mathtt{m_{sd}}\times\mathtt{m_{sd}}$ and $\widehat{\mathbf{N}}$ is of size $\mathtt{n_{fn}}\mathtt{m_{sd}}\times\mathtt{m_{sd}}$. Let the vector of dimension $\mathtt{n_{sd}}$ denoting the gradient of the interpolating function N_1 in the physical/spatial coordinates be $\boldsymbol{\nabla}^{\mathsf{s}}N_1 = \mathbf{J}^{-1}\boldsymbol{\nabla}N_1$, where $\mathbf{J}$ is the jacobian of the isoparametric transformation. The matrix representing each component of the gradient of interpolating functions are given by,

$$\boldsymbol{\nabla}^{\mathsf{s}}\mathbf{N}|_I = \begin{bmatrix} \boldsymbol{\nabla}^{\mathsf{s}}N_1|_I\mathbf{I}_{\mathtt{m_{sd}}} & \boldsymbol{\nabla}^{\mathsf{s}}N_2|_I\mathbf{I}_{\mathtt{m_{sd}}} & \dots & \boldsymbol{\nabla}^{\mathsf{s}}N_{\mathtt{n_{en}}}|_I\mathbf{I}_{\mathtt{m_{sd}}} \end{bmatrix}^T, \tag{3.8.2}$$

for $I = 1.\dots.\mathtt{n_{sd}}$, where $\boldsymbol{\nabla}^{\mathsf{s}}\mathbf{N}|_I$ is of size $\mathtt{n_{en}}\mathtt{m_{sd}} \times \mathtt{m_{sd}}$.

The vectors in Equation (3.7.3) for each element $e = 1, \dots, \mathtt{n_{el}}$ and is given by,

$$
\begin{aligned}
(\mathbf{f}_u^e)^{n+1,q} = & - \sum_{s=0}^{\mathtt{n_{tg}}} a_s \sum_{\mathrm{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\mathtt{e}}} \mathbf{N}(\boldsymbol{\xi}_{\mathrm{g}}^{\mathtt{e}}) \mathbf{U}^{n+1-s,q}(\boldsymbol{\xi}_{\mathrm{g}}^{\mathtt{e}}) w_{\mathrm{g}}^{\mathtt{e}} \\
& + \sum_{\mathrm{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\mathtt{e}}} \sum_{I=1}^{\mathtt{n_{sd}}} \boldsymbol{\nabla}^{\mathrm{s}} \mathbf{N}(\boldsymbol{\xi}_{\mathrm{g}}^{\mathtt{e}})|_I \left(\mathbf{F}(\mathbf{U}^{n+1,q}(\boldsymbol{\xi}_{\mathrm{g}}^{\mathtt{e}}))|_I \right) w_{\mathrm{g}}^{\mathtt{e}} \\
& - \sum_{\partial\Omega_e^1} \sum_{\mathrm{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\mathtt{f}}} \mathbf{N}(\boldsymbol{\xi}_{\mathrm{g}}^{\mathtt{f}}) \left(\mathbf{F}(\widehat{\mathbf{U}}^{n+1,q}(\boldsymbol{\xi}_{\mathrm{g}}^{\mathtt{f}})) \cdot \boldsymbol{n}(\boldsymbol{\xi}_{\mathrm{g}}^{\mathtt{f}}) \right) w_{\mathrm{g}}^{\mathtt{f}} \\
& - \sum_{\partial\Omega_e^1} \sum_{\mathrm{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\mathtt{f}}} \mathbf{N}(\boldsymbol{\xi}_{\mathrm{g}}^{\mathtt{f}}) \boldsymbol{\tau}^{n+1}(\boldsymbol{\xi}_{\mathrm{g}}^{\mathtt{f}}) \mathbf{U}^{n+1,q}(\boldsymbol{\xi}_{\mathrm{g}}^{\mathtt{f}}) w_{\mathrm{g}}^{\mathtt{f}} \\
& + \sum_{\partial\Omega_e^1} \sum_{\mathrm{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\mathtt{f}}} \mathbf{N}(\boldsymbol{\xi}_{\mathrm{g}}^{\mathtt{f}}) \boldsymbol{\tau}^{n+1}(\boldsymbol{\xi}_{\mathrm{g}}^{\mathtt{f}}) \widehat{\mathbf{U}}^{n+1,q}(\boldsymbol{\xi}_{\mathrm{g}}^{\mathtt{f}}) w_{\mathrm{g}}^{\mathtt{f}} \\
& + \sum_{\mathrm{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\mathtt{e}}} \mathbf{N}(\boldsymbol{\xi}_{\mathrm{g}}^{\mathtt{e}}) \mathbf{S}^{n+1}(\boldsymbol{x}(\boldsymbol{\xi}_{\mathrm{g}}^{\mathtt{e}})) w_{\mathrm{g}}^{\mathtt{e}} \\
& - \kappa^n \sum_{\mathrm{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\mathtt{e}}} \sum_{I=1}^{\mathtt{n_{sd}}} \boldsymbol{\nabla}^{\mathrm{s}} \mathbf{N}(\boldsymbol{\xi}_{\mathrm{g}}^{\mathtt{e}})|_I \boldsymbol{\nabla}^{\mathrm{s}} \mathbf{N}^T(\boldsymbol{\xi}_{\mathrm{g}}^{\mathtt{e}})|_I (\mathbf{U}^e)^{n+1,q} w_{\mathrm{g}}^{\mathtt{e}}, \qquad (3.8.3\text{a}) \\
(\mathbf{f}_{\hat{u}}^e)^{n+1,q} = & - \sum_{\partial\Omega_e^1 \setminus \partial\Omega^1} \sum_{\mathrm{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\mathtt{f}}} \widehat{\mathbf{N}}(\boldsymbol{\xi}_{\mathrm{g}}^{\mathtt{f}}) \left(\mathbf{F}(\widehat{\mathbf{U}}^{n+1,q}(\boldsymbol{\xi}_{\mathrm{g}}^{\mathtt{f}})) \cdot \boldsymbol{n}(\boldsymbol{\xi}_{\mathrm{g}}^{\mathtt{f}}) \right) w_{\mathrm{g}}^{\mathtt{f}} \\
& - \sum_{\partial\Omega_e^1 \setminus \partial\Omega^1} \sum_{\mathrm{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\mathtt{f}}} \widehat{\mathbf{N}}(\boldsymbol{\xi}_{\mathrm{g}}^{\mathtt{f}}) \boldsymbol{\tau}^{n+1}(\boldsymbol{\xi}_{\mathrm{g}}^{\mathtt{f}}) \mathbf{U}^{n+1,q}(\boldsymbol{\xi}_{\mathrm{g}}^{\mathtt{f}}) w_{\mathrm{g}}^{\mathtt{f}} \\
& + \sum_{\partial\Omega_e^1 \setminus \partial\Omega^1} \sum_{\mathrm{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\mathtt{f}}} \widehat{\mathbf{N}}(\boldsymbol{\xi}_{\mathrm{g}}^{\mathtt{f}}) \boldsymbol{\tau}^{n+1}(\boldsymbol{\xi}_{\mathrm{g}}^{\mathtt{f}}) \widehat{\mathbf{U}}^{n+1,q}(\boldsymbol{\xi}_{\mathrm{g}}^{\mathtt{f}}) w_{\mathrm{g}}^{\mathtt{f}} \\
& - \sum_{\partial\Omega_e^1 \setminus \partial\Omega^1} \sum_{\mathrm{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\mathtt{f}}} \widehat{\mathbf{N}}(\boldsymbol{\xi}_{\mathrm{g}}^{\mathtt{f}}) \widehat{\boldsymbol{B}}(\boldsymbol{\xi}_{\mathrm{g}}^{\mathtt{f}}) w_{\mathrm{g}}^{\mathtt{f}}, \qquad (3.8.3\text{b})
\end{aligned}
$$

and the matrices in Equation (3.7.2) for each element $e = 1, \dots, \mathtt{n}_{\mathtt{el}}$ and is given by,

$$\begin{aligned}
(\mathbf{T}_{uu})^{e,n+1,q} &= a_0 \sum_{\mathtt{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\mathtt{e}}} \mathbf{N}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{e}})\mathbf{N}^T(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{e}}) w_{\mathtt{g}}^{\mathtt{e}} \\
&\quad - \sum_{\mathtt{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\mathtt{e}}} \sum_{I=1}^{\mathtt{n}_{\mathtt{sd}}} \mathbf{N}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{e}}) \left(\mathbf{A}(\mathbf{U}^{n+1,q}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{e}}))|_I\right) \boldsymbol{\nabla}^{\mathtt{s}}\mathbf{N}^T(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{e}})|_I w_{\mathtt{g}}^{\mathtt{e}} \\
&\quad + \sum_{\partial\Omega_e^1} \sum_{\mathtt{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\mathtt{f}}} \mathbf{N}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) \boldsymbol{\tau}^{n+1}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) \mathbf{N}^T(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) w_{\mathtt{g}}^{\mathtt{f}} \\
&\quad + \sum_{\partial\Omega_e^1} \sum_{\mathtt{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\mathtt{f}}} \mathbf{N}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) \left(\frac{\partial \boldsymbol{\tau}^{n+1}}{\partial \mathbf{U}^{n+1}}|_{\mathbf{U}^{n+1,q}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}})} \mathbf{U}^{n+1,q}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) \right) \mathbf{N}^T(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) w_{\mathtt{g}}^{\mathtt{f}} \\
&\quad + \kappa^n \sum_{\mathtt{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\mathtt{e}}} \sum_{I=1}^{\mathtt{n}_{\mathtt{sd}}} \boldsymbol{\nabla}^{\mathtt{s}}\mathbf{N}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{e}})|_I \boldsymbol{\nabla}^{\mathtt{s}}\mathbf{N}^T(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{e}})|_I w_{\mathtt{g}}^{\mathtt{e}},
\end{aligned} \tag{3.8.4a}$$

$$\begin{aligned}
(\mathbf{T}_{u\hat{u}})^{e,n+1,q} &= \sum_{\partial\Omega_e^1} \sum_{\mathtt{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\mathtt{f}}} \mathbf{N}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) \left(\mathbf{A}(\widehat{\mathbf{U}}^{n+1,q}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}})) \cdot \boldsymbol{n}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}})\right) \widehat{\mathbf{N}}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) w_{\mathtt{g}}^{\mathtt{f}} \\
&\quad - \sum_{\partial\Omega_e^1} \sum_{\mathtt{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\mathtt{f}}} \mathbf{N}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) \boldsymbol{\tau}^{n+1}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) \widehat{\mathbf{N}}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) w_{\mathtt{g}}^{\mathtt{f}} \\
&\quad - \sum_{\partial\Omega_e^1} \sum_{\mathtt{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\mathtt{f}}} \mathbf{N}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) \left(\frac{\partial \boldsymbol{\tau}^{n+1}}{\partial \widehat{\mathbf{U}}^{n+1}}|_{\widehat{\mathbf{U}}^{n+1,q}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}})} \widehat{\mathbf{U}}^{n+1,q}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) \right) \widehat{\mathbf{N}}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) w_{\mathtt{g}}^{\mathtt{f}},
\end{aligned} \tag{3.8.4b}$$

$$\begin{aligned}
(\mathbf{T}_{\hat{u}u})^{e,n+1,q} &= \sum_{\partial\Omega_e^1 \setminus \partial\Omega^1} \sum_{\mathtt{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\mathtt{f}}} \widehat{\mathbf{N}}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) \boldsymbol{\tau}^{n+1}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) \mathbf{N}^T(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) w_{\mathtt{g}}^{\mathtt{f}} \\
&\quad + \sum_{\partial\Omega_e^1 \setminus \partial\Omega^1} \sum_{\mathtt{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\mathtt{f}}} \widehat{\mathbf{N}}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) \left(\frac{\partial \boldsymbol{\tau}^{n+1}}{\partial \mathbf{U}^{n+1}}|_{\mathbf{U}^{n+1,q}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}})} \mathbf{U}^{n+1,q}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) \right) \mathbf{N}^T(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) w_{\mathtt{g}}^{\mathtt{f}} \\
&\quad + \sum_{\partial\Omega_e^1 \cap \partial\Omega^1} \sum_{\mathtt{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\mathtt{f}}} \widehat{\mathbf{N}}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) \left(\frac{\partial \widehat{\boldsymbol{B}}}{\partial \mathbf{U}^{n+1}}|_{\mathbf{U}^{n+1,q}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}})} \right) \mathbf{N}^T(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) w_{\mathtt{g}}^{\mathtt{f}},
\end{aligned} \tag{3.8.4c}$$

$$\begin{aligned}
(\mathbf{T}_{\hat{u}\hat{u}})^{e,n+1,q} &= \sum_{\partial\Omega_e^1 \setminus \partial\Omega^1} \sum_{\mathtt{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\mathtt{f}}} \widehat{\mathbf{N}}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) \left(\mathbf{A}(\widehat{\mathbf{U}}^{n+1,q}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}})) \cdot \boldsymbol{n}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}})\right) \widehat{\mathbf{N}}^T(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) w_{\mathtt{g}}^{\mathtt{f}} \\
&\quad - \sum_{\partial\Omega_e^1 \setminus \partial\Omega^1} \sum_{\mathtt{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\mathtt{f}}} \widehat{\mathbf{N}}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) \boldsymbol{\tau}^{n+1}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) \widehat{\mathbf{N}}^T(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) w_{\mathtt{g}}^{\mathtt{f}} \\
&\quad - \sum_{\partial\Omega_e^1 \setminus \partial\Omega^1} \sum_{\mathtt{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\mathtt{f}}} \widehat{\mathbf{N}}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) \left(\frac{\partial \boldsymbol{\tau}^{n+1}}{\partial \widehat{\mathbf{U}}^{n+1}}|_{\widehat{\mathbf{U}}^{n+1,q}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}})} \widehat{\mathbf{U}}^{n+1,q}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) \right) \widehat{\mathbf{N}}^T(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) w_{\mathtt{g}}^{\mathtt{f}} \\
&\quad + \sum_{\partial\Omega_e^1 \cap \partial\Omega^1} \sum_{\mathtt{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\mathtt{f}}} \widehat{\mathbf{N}}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) \left(\frac{\partial \widehat{\boldsymbol{B}}}{\partial \widehat{\mathbf{U}}^{n+1}}|_{\widehat{\mathbf{U}}^{n+1,q}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}})} \right) \widehat{\mathbf{N}}^T(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) w_{\mathtt{g}}^{\mathtt{f}},
\end{aligned} \tag{3.8.4d}$$

where, $\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{e}}$ and $w_{\mathtt{g}}^{\mathtt{e}}$ are the $\mathtt{n}_{\mathtt{ip}}^{\mathtt{e}}$ integration points and weights defined in the reference element and $\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}$ and $w_{\mathtt{g}}^{\mathtt{f}}$ are the $\mathtt{n}_{\mathtt{ip}}^{\mathtt{f}}$ integration points defined on the reference edge/face.

3.9 Implementation details

A brief description of the important details regarding the implementation of the HDG method is presented. Alongside, the default values of parameters concerning these details, wherever relevant, are provided.

The work utilises meshes with straight-sided triangles in 2D and planar-faced tetrahedral elements in 3D. These are chosen for their suitability for the Euler problem, ease of mesh generation, efficiency of storage (only vertex coordinates and connectivities are required) and ease of transformation of element quantities between the reference and the physical element using the jacobian of its transformation. To avoid ill-conditioning of elemental matrices, Fekette nodal distributions in the reference elements are adopted [62], as illustrated in Figure 3.1. High-order numerical integration is performed using the recently proposed symmetric quadrature rules in [164]. The number of gauss points for each element with a known polynomial approximation is chosen such that the consistent mass matrix is integrated exactly.

The convergence of the Newton-Raphson linearisation method is monitored through the values of the residual and the solution increments in each iteration. In particular, the following $\mathcal{L}_2$ norms of the residual and the solution error, both in the domain interior and the mesh skeleton, are defined as,

$$||\boldsymbol{\epsilon}_{R^e}||_{\mathcal{L}_2(\Omega^1)} := \sqrt{\sum_{e=1}^{\mathtt{n_{el}}} \int_{\Omega_e^1} \boldsymbol{\mathcal{R}}^e \cdot \boldsymbol{\mathcal{R}}^e \, \partial\Omega}, \tag{3.9.1a}$$

$$||\boldsymbol{\epsilon}_{\Delta U}||_{\mathcal{L}_2(\Omega^1)} := \frac{\sqrt{\int_{\Omega^1} \Delta \boldsymbol{U}^{n+1,q} \cdot \Delta \boldsymbol{U}^{n+1,q} \, \partial\Omega}}{\sqrt{\int_{\Omega^1} \boldsymbol{U}^{n+1,q+1} \cdot \boldsymbol{U}^{n+1,q+1} \, \partial\Omega}}, \tag{3.9.1b}$$

$$||\boldsymbol{\epsilon}_{\widehat{R}^e}||_{\mathcal{L}_2(\Gamma^1)} := \sqrt{\sum_{e=1}^{\mathtt{n_{el}}} \int_{\Gamma^1} \widehat{\boldsymbol{\mathcal{R}}}^e \cdot \widehat{\boldsymbol{\mathcal{R}}}^e \, \partial\Omega}, \tag{3.9.1c}$$

$$||\boldsymbol{\epsilon}_{\Delta \widehat{U}}||_{\mathcal{L}_2(\Gamma^1)} := \frac{\sqrt{\int_{\Gamma^1} \Delta \widehat{\boldsymbol{U}}^{n+1,q} \cdot \Delta \widehat{\boldsymbol{U}}^{n+1,q} \, \partial\Omega}}{\sqrt{\int_{\Gamma^1} \widehat{\boldsymbol{U}}^{n+1,q+1} \cdot \widehat{\boldsymbol{U}}^{n+1,q+1} \, \partial\Omega}}, \tag{3.9.1d}$$

and whose quadratic convergence of the quantities are expected. In particular cases, normalisation of the quantities in Equation (3.9.1) with respect to time step size may be carried out. The monitoring of the residual ensures correctness of the converged

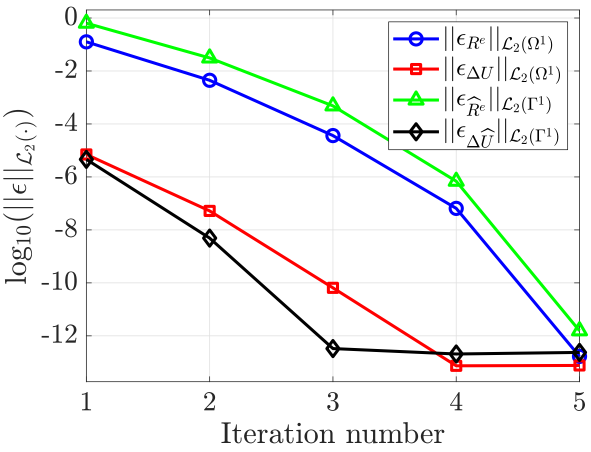


Figure. 3.2 Quadratic convergence of Newton-Raphson method demonstrating consistent linearisation of the non-linear system of equations for HDG method.

solution. The linearisation method is said to have converged when the quantities fall below a pre-determined tolerance. In all the cases, the value of the tolerance for the Newton-Raphson method is set to 1.0×10^{-10}, unless specified otherwise. Figure 3.2 shows the typical quadratic convergence behaviour, where the maximum of the $\mathtt{m}_{\mathtt{sd}}$ components of the errors in Equation (3.9.1) are plotted against the iteration number, which confirms a consistent linearisation of the non-linear terms.

The work utilises both direct and iterative sparse linear system solvers. Direct methods are favourable for their robustness of operation and handling of ill-conditioning but they incur high memory and numerical costs. On the other hand, iterative solvers are more scalable and may require less memory and solution time but have its convergence depending on the conditioning of the matrix. In the present work, direct methods are favoured for linear system less than a million degrees of freedom. Multi-frontal method for sparse unsymmetrical systems of the HSL library [68, 94] and multi-threaded PARDISO library [37, 153, 80] have been used in the examples. For larger linear systems, the multi-core PETSc implementation [8–10] of the Generalised Minimal Residual (GMRES) iterative method with incomplete LU pre-conditioner has been used. The tolerance for convergence of the linear system residual is set to 1.0×10^{-15} by default.

A pseudo-code laying out the solution procedure of the HDG method is provided in Algorithm 1.

```
Initial solution: $\boldsymbol{U}^{n=0}$ and $\widehat{\boldsymbol{U}}^{n=0}$

*Time stepping*
for $t^{n+1}$, $n = 0$ to $end - 1$ do
    Data: $\boldsymbol{U}^{n}$ and $\widehat{\boldsymbol{U}}^{n}$
    Initial guess: $\boldsymbol{U}^{n+1,q=0} = \boldsymbol{U}^{n}$, $\widehat{\boldsymbol{U}}^{n+1,q=0} = \widehat{\boldsymbol{U}}^{n}$
    Compute residuals: $\boldsymbol{\mathcal{R}}^{n+1,0}$ from Equation (3.6.1)

    *Newton-Raphson linearisation*
    repeat
        *Global problem*
        Data: $\boldsymbol{U}^{n+1,q}, \dots, \boldsymbol{U}^{n+1-\mathtt{n_{tg}}}, \widehat{\boldsymbol{U}}^{n+1,q}$
        for $e = 1$ to $\boldsymbol{n_{el}}$ do
            Compute matrices:
             $(\mathbf{T}_{uu})_{IJ}^{e,n+1,q}, (\mathbf{T}_{u\hat{u}})_{IJ}^{e,n+1,q}, (\mathbf{T}_{\hat{u}u})_{IJ}^{e,n+1,q}$ and $(\mathbf{T}_{\hat{u}\hat{u}})_{IJ}^{e,n+1,q}$ in
             Equation (3.7.2)
            Compute vectors: $(\mathbf{f}_{u}^{e})_{I}^{n+1,q}$ and $(\mathbf{f}_{\hat{u}}^{e})_{I}^{n+1,q}$ in Equation (3.7.3)
            Assemble: $[\widetilde{\mathbf{T}}_{\hat{u}\hat{u}}]^{e,n+1,q}$ and $\{\tilde{\mathbf{f}}_{\hat{u}}\}^{e,n+1,q}$ to Equation (3.7.4)
        end

        Solve linear system, obtain $\{\Delta\widehat{\boldsymbol{U}}\}^{n+1,q}$ in Equation (3.7.4)
        Update: $\widehat{\boldsymbol{U}}^{n+1,q+1} = \widehat{\boldsymbol{U}}^{n+1,q} + \{\Delta\widehat{\boldsymbol{U}}\}^{n+1,q}$

        *Local problem*
        Obtain $\{\Delta\boldsymbol{U}\}^{n+1,q}$ in Equation (3.7.6)
        Update: $\boldsymbol{U}^{n+1,q+1} = \boldsymbol{U}^{n+1,q} + \{\Delta\boldsymbol{U}\}^{n+1,q}$

        Update residuals: $\boldsymbol{\mathcal{R}}^{n+1,q+1}$ from Equation (3.6.1)

        Update: $q = q + 1$
        Compute errors in Equation (3.9.1)
    until $||\boldsymbol{\epsilon}_{R^e}||_{\mathcal{L}_2(\Omega^1)}, ||\boldsymbol{\epsilon}_{\Delta U}||_{\mathcal{L}_2(\Omega^1)}, ||\boldsymbol{\epsilon}_{\widehat{R}^e}||_{\mathcal{L}_2(\Gamma^1)}$ *and* $||\boldsymbol{\epsilon}_{\Delta\widehat{U}}||_{\mathcal{L}_2(\Gamma^1)} \leq$ *tolerance*;

    Update solution: $\widehat{\boldsymbol{U}}^{n+1} = \widehat{\boldsymbol{U}}^{n+1,q+1}$ and $\boldsymbol{U}^{n+1} = \boldsymbol{U}^{n+1,q+1}$
end
```

Algorithm 1: HDG solution procedure

3.10 Ringleb flow problem

The Ringleb flow is a widely adopted test case to test the spatial accuracy of high-order methods for 2D Euler equations. It provides an exact, smooth and steady solution in the transonic regime via the hodograph technique. Here, the solution is obtained by the separation of variables when the coordinate system is transformed to the so-called hodograph plane, see [61].

An exact solution as a function of the spatial coordinates $\boldsymbol{x}$ is given by solving the following non-linear equation for the speed of sound c,

$$\left(x + \frac{J}{2}\right)^2 + y^2 = \frac{1}{4\rho^2 v^4}, \tag{3.10.1}$$

where the quantity J, velocity magnitude v and density ρ are functions of c as,

$$J = \frac{1}{c} + \frac{1}{3c^3} + \frac{1}{5c^5} - \frac{1}{2}\ln\left(\frac{1+c}{1-c}\right), \tag{3.10.2}$$

$$v = \sqrt{\frac{2(1-c^2)}{\gamma - 1}}, \tag{3.10.3}$$

$$\rho = c^{\frac{2}{\gamma-1}}. \tag{3.10.4}$$

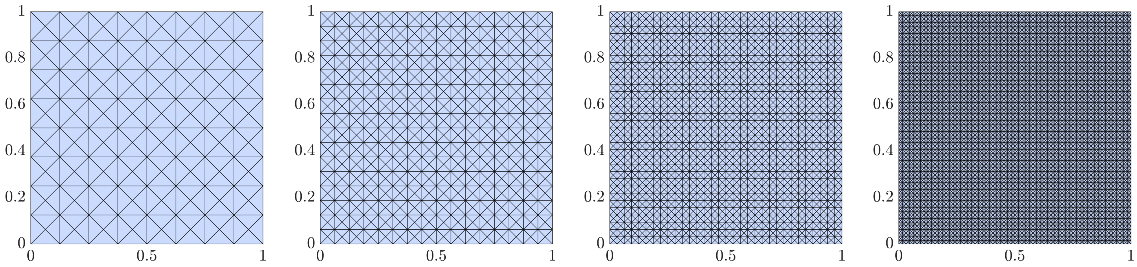
With the known value of c, flow variables such as pressure and velocity can be calculated using the relations,

$$p = \frac{1}{\gamma} c^{\frac{2\gamma}{\gamma-1}}, \; \boldsymbol{v} = \begin{Bmatrix} \pm v \sin\theta \\ v\cos\theta \end{Bmatrix}, \tag{3.10.5}$$

where,

$$\theta = \sin^{-1}(v\psi), \; \psi = \sqrt{\frac{1}{2v^2} - p\left(x + \frac{J}{2}\right)}. \tag{3.10.6}$$

Traditionally, the Ringleb flow is setup with a domain bounded by a pair of streamlines and inflow-outflow boundaries [159]. This however is not necessary, since an arbitrary domain bounded by consistently enforced inflow-outflow conditions is sufficient [119]. Hence in all the cases presented in this work, the boundaries are treated as inflow/outflow conditions given by the analytical solution.



(a) Mesh 1 (b) Mesh 2 (c) Mesh 3 (d) Mesh 4

Figure. 3.3 Triangular meshes of the domain $\Omega = [0, 1]^2$ used to test the optimal convergence properties of the HDG method.

3.11 Numerical tests

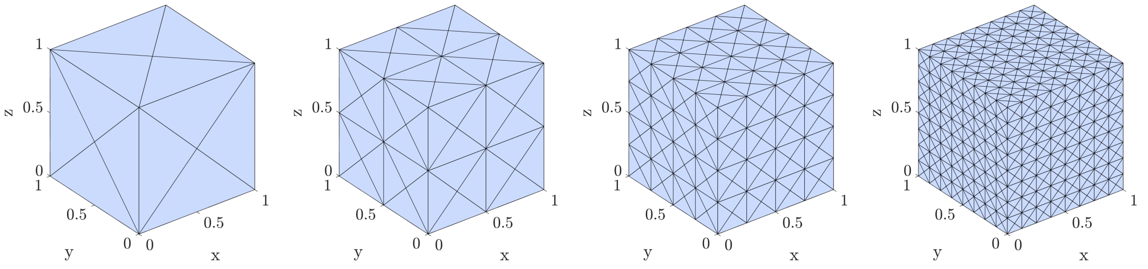
Convergence properties of the error in the HDG solution, both in space and time, are presented in this section. While Ringleb flow problem described in Section 3.10 presents a suitable test case for the spatial convergence test in 2D domains, the method of manufactured solutions is the preferred choice for the spatial convergence test in 3D domains and temporal convergence tests in both 2D and 3D domains. In the manufactured solution test, a suitable analytical solution $\boldsymbol{\mathcal{U}}_a$ is designed for the Euler equations by providing an appropriate source term such that the equations are satisfied. These are then solved numerically, whose solution can then be compared with the designed analytical solution. The following definition of the solution error $\boldsymbol{\epsilon}$ in the $\mathcal{L}_2(\Omega^1)$ given by,

$$||\boldsymbol{\epsilon}||_{\mathcal{L}_2(\Omega^1)} := \frac{\sqrt{\int_{\Omega^1} \left(\boldsymbol{U} - \boldsymbol{\mathcal{U}}_a\right)^2 \, d\Omega}}{\sqrt{\int_{\Omega^1} \boldsymbol{\mathcal{U}}_a^2 \, d\Omega}}, \tag{3.11.1}$$

is utilised to quantify the discretisation error. These tests confirm the solution accuracy, predictability of error behaviour and demonstrate a bug-free implementation of the method.

3.11.1 Spatial convergence tests

For spatial convergence tests in 2D, four uniform meshes of the domain $\Omega^1 = [0\;1]^2$ are considered, with $256, 1024, 4096$ and $16\,384$ triangular elements respectively, as shown in Figure 3.3.



(a) Mesh 1 (b) Mesh 2 (c) Mesh 3 (d) Mesh 4

Figure. 3.4 Tetrahedral meshes of the domain $\Omega = [0, 1]^3$ used to test the optimal convergence properties of the HDG method.

For spatial convergence tests in 3D, the analytical solution is constructed with the following variation in the density, velocity and specific energy,

$$\rho := 2 + \sin(\pi(x_1 + x_2 + x_3)/3), \tag{3.11.2a}$$

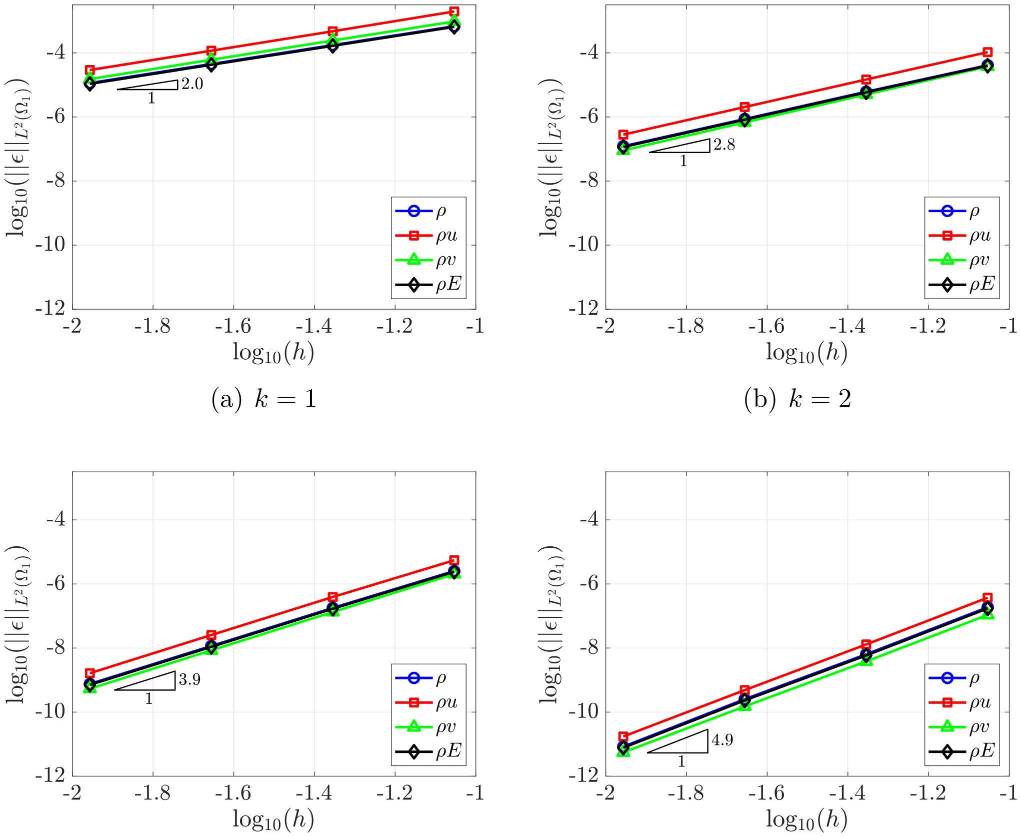
$$\boldsymbol{v} := \begin{pmatrix} 3 + \sqrt{3}\ \cos(\pi x_2)/4 \\ 3 + 3\ \sin(\pi x_3)/4 \\ 3 + \cos(\pi x_1)/2 \end{pmatrix}, \tag{3.11.2b}$$

$$E := 25 + \sin(\pi(x_1 + x_2 + x_3)/5). \tag{3.11.2c}$$

Four uniform meshes of the domain $\Omega^1 = [0\ 1]^3$ are considered, with 24, 192, 1536 and 12 288 tetrahedral elements respectively, as shown in Figure 3.4.

For both 2D and 3D test cases, values of the polynomial order k up to 4 is considered. A nodally exact initial solution is constructed and time marched until a steady solution is obtained. The solution error in Equation (3.11.1) is expected to be proportional to h^{k+1}. This can be clearly seen in the nearly optimal convergence rates with asymptotic slope of $k + 1$ for various mesh sizes and polynomial orders for 2D in Figure 3.5 and 3D in Figure 3.6 respectively.

It is interesting to observe, in Figure 3.5, the mesh size required to reach a certain value of the solution accuracy. For instance, for the solution error equal to 1.0×10^{-6}, the HDG method with $k = 2$ requires 4096 elements, with $k = 3$ requires 1024 elements and with $k = 4$ requires 256 elements. Figure 3.7 shows the Ma number plot for the these meshes. These correspond to linear system sizes of $74\,496, 25\,088$ and 8000 respectively. Normalised with respect to the CPU time for the largest problem size (corresponding to $k = 2$), the CPU time for high-order approximations with $k = 3$ and $k = 4$ are 0.57 and 0.47 respectively. This highlights the advantage of high-order methods over



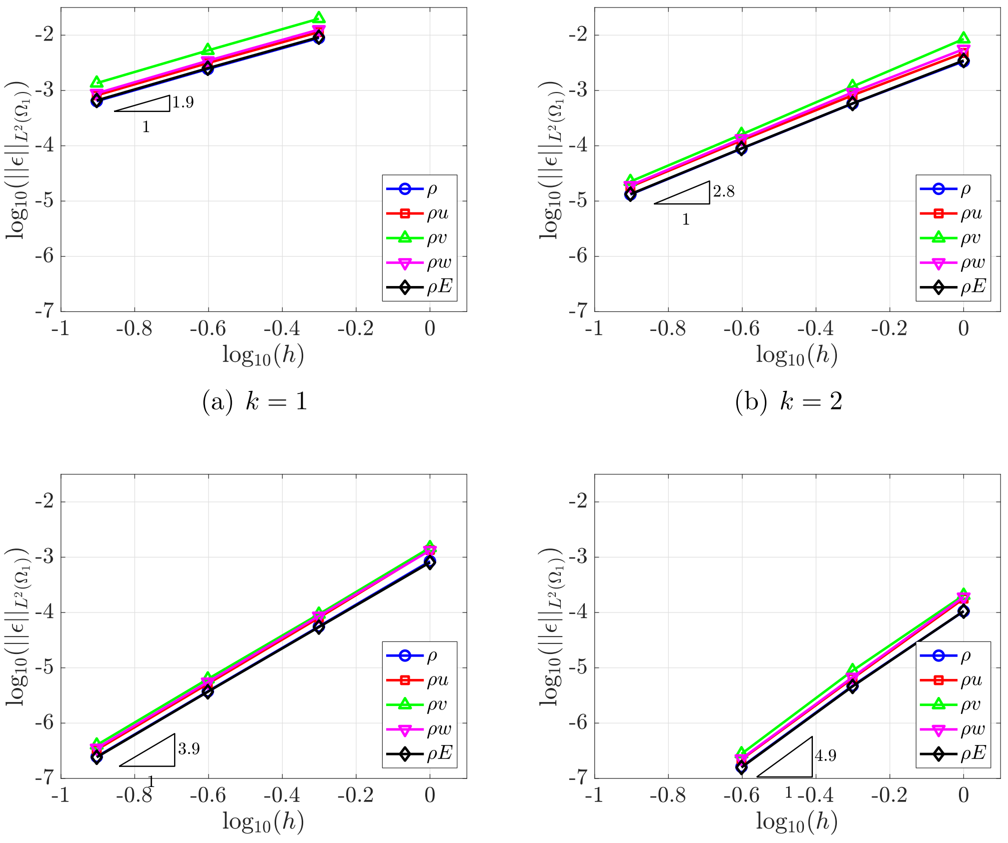
(a) $k = 1$

(b) $k = 2$

(c) $k = 3$

(d) $k = 4$

Figure. 3.5 The optimal rates of convergence in the plots of the $\mathcal{L}_2(\Omega^1)$ of the error in all the conservative variables versus the mesh size h, for different orders of approximation k of the HDG method. The example demonstrated here is the 2D Ringleb flow.



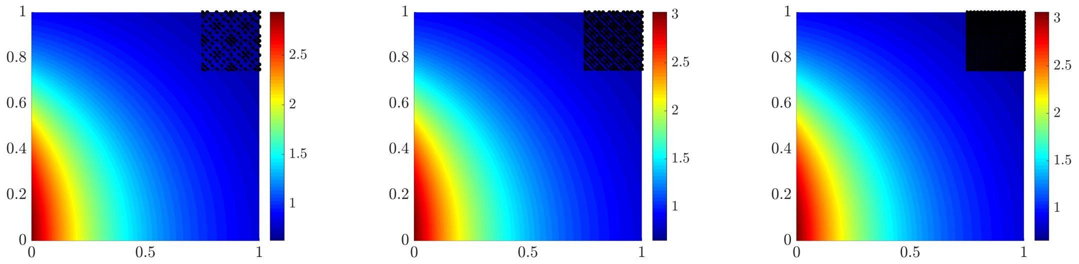
(a) $k = 1$

(b) $k = 2$

(c) $k = 3$

(d) $k = 4$

Figure. 3.6 The optimal rates of convergence in the plots of the $\mathcal{L}_2(\Omega^1)$ of the error in all the conservative variables versus the mesh size h, for different orders of approximation k of the HDG method. The example demonstrated here is the 3D manufactured solution for spatial convergence given in Equation (3.11.2).



(a) 256 $k = 1$ elements

(b) 1024 $k = 2$ elements

(c) 4096 $k = 3$ elements

Figure. 3.7 Mach number plots using various meshes and different k's for the HDG solution of the Ringleb flow problem corresponding to the solution error less than 1.0×10^{-6}

low-order methods in providing high-fidelity solution for a lower number of degrees of freedom and CPU time. The Mach number plot of the solutions on these meshes are shown in Figure 3.7. This trend is observed for the 3D test case in Figure 3.6 as well.

3.11.2 Temporal convergence tests

As mentioned in Section 3.4, BDF1 and BDF2 time marching methods have been implemented.

For temporal convergence tests in 2D, the analytical solution is constructed with the following variation in the density, velocity and specific energy,

$$\rho := 2 + 0.1\ \cos(\pi t), \tag{3.11.3a}$$

$$\boldsymbol{v} := \begin{pmatrix} 4 + 0.1\ \sin(2\pi t) \\ 4 + 0.1\ \cos(3\pi t) \end{pmatrix}, \tag{3.11.3b}$$

$$E := 50 + 0.1\ \sin(4\pi t). \tag{3.11.3c}$$

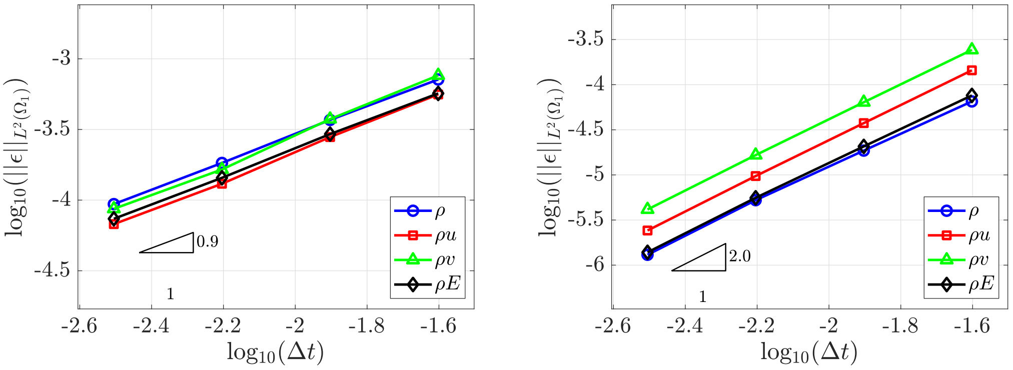
For temporal convergence tests in 3D, the analytical solution is constructed with the following variation in the density, velocity and specific energy,

$$\rho := 2 + 0.1\ \cos(\pi t), \tag{3.11.4a}$$

$$\boldsymbol{v} := \begin{pmatrix} 4 + 0.1\ \sin(2\pi t) \\ 4 + 0.1\ \cos(3\pi t) \\ 4 + 0.1\ \sin(4\pi t) \end{pmatrix}, \tag{3.11.4b}$$

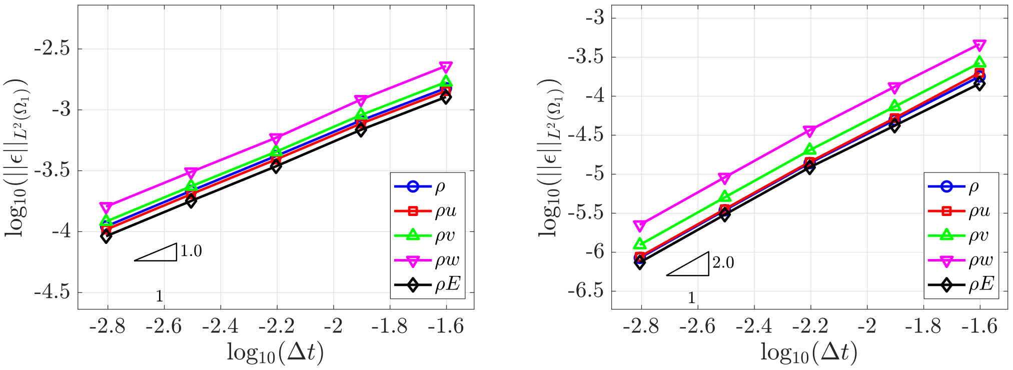
$$E := 50 + 0.1\ \cos(5\pi t). \tag{3.11.4c}$$

For both 2D and 3D test cases, a fine mesh or a high-order approximation is chosen in order to minimize the influence of the spatial discretisation error. The initial solution, chosen to be the nodally exact analytical solution, is time marched until the same physical time is reached, with decreasing time step sizes. Figures 3.8 and 3.9 demonstrate that the error in Equation (3.11.1) decreases at a rate given by the order of the BDF method for both 2D and 3D cases respectively.



(a) BDF1 (b) BDF2

Figure. 3.8 The optimal rates of convergence for HDG method in the plots of the $\mathcal{L}_2(\Omega^1)$ of the error in all the conservative variables versus the time step size Δt, for different orders of the BDF method. The example demonstrated here is the 2D manufactured solution for temporal convergence given in Equation (3.11.3).



(a) BDF1 (b) BSF2

Figure. 3.9 The optimal rates of convergence for HDG method in the plots of the $\mathcal{L}_2(\Omega^1)$ of the error in all the conservative variables versus the time step size Δt, for different orders of the BDF method. The example demonstrated here is the 3D manufactured solution for temporal convergence given in Equation (3.11.4).

Chapter 4

The vertex-centred finite volume method

"Before researchers become researchers, they should become philosophers. They should consider what the human goal is, what it is that humanity should create."
Masanobu Fukuoka
The one-straw revolution

The finite volume method for conservation equations can be broken down into two steps. Firstly, the PDEs for the conservative variables are transformed into integration of balance equations in discrete elements called control volumes. Volume and surface integrals are numerically solved through quadrature rules of suitable accuracy over the control volume and its boundaries respectively. Second step involves obtaining discrete algebraic equations by choosing suitable interpolation profiles for approximating the values on the control volume surfaces based on discrete control volume values of conservative variables. Finite volume methods are popular, in part due to the automatic satisfaction of conservative variables over each control volume and the whole domain independent of the number of control volumes, not just in a limiting sense when the number of control volumes become large. Its appeal is enhanced by the ability to carry out discretisation in the physical space without having to transform into the reference space.

Vertex-centred finite volume framework has been robustly applied to the Euler and Navier-Stokes equations as an efficient discretisation method on unstructured meshes [103, 72, 34, 146, 161, 145, 144]. The simplicity afforded by the central-difference type discretisation with its compact stencil is unrivalled. Moreover a vertex-centred approach can be shown to be equivalent to linear finite elements[12]. For 2D grids, the number of calculations in an element-based approach, typically in cell-centred methods and an edge-based method, in vertex-centred methods do not vary significantly. In tetrahedral meshes, the ratio of the number of faces to the number of edges is 6.5/11. Hence, edge-based approaches are more favourable computationally and in storage requirements [102, 170].

The chapter begins with some preliminaries regarding the domain discretisation and notation for the edge-based data structure. The vertex-centred finite volume method is presented as the solution methodology for the set of integral equations on discrete control volumes. Suitable treatment for the volume and surface integrals to achieve a second-order scheme in space is laid out. An accompanying stabilisation method through artificial dissipation along with the shock capturing method is described. Implicit time integration and the solution procedure for the resulting discrete non-linear system of equations is explained. Key details regarding the implementation are described. This is followed by a concluding section on numerical tests for optimal convergence properties of the FV method.

4.1 Preliminaries

Consider an open bounded domain $\Omega^2 \in \mathbb{R}^{\mathtt{n_{sd}}}$, with boundary $\partial\Omega^2$ and $\mathtt{n_{sd}}$ the number of spatial dimensions, to be discretised with the FV method. As a first step, assume that Ω^2 is partitioned into $\mathtt{n_{el}}$ disjoint elements Ω^2_e such that,

$$\overline{\Omega}^2 = \bigcup_{e=1}^{\mathtt{n_{el}}} \overline{\Omega}^2_e, \quad \Omega^2_e \cap \Omega^2_f = \emptyset \quad \text{for } e \neq f. \tag{4.1.1}$$

Once the domain is discretised into elements, control volumes can be defined, generally with two approaches: element-based and node-based [105]. The element-based approach has the unknown degrees of freedom located inside the element, typically at the element centroid with the control volume corresponding to the discrete element. This is the classical cell-centred approach. On the other hand, node-based schemes have unknown degrees of freedom at the element vertices with many possible choices of the control volume surrounding it. In the so-called vertex-centred approach, a dual mesh is constructed. The median dual principle is applied, where the control volumes are generated by connecting edge centres, face and the volume centroids of the elements, see Figure 4.1. In 3D cases, for hybrid meshes generated by merging tetrahedral elements, modifications need to be made to the median dual principle in order to obtain a good mesh, see for instance [144]. Vertex-centred approaches perform better

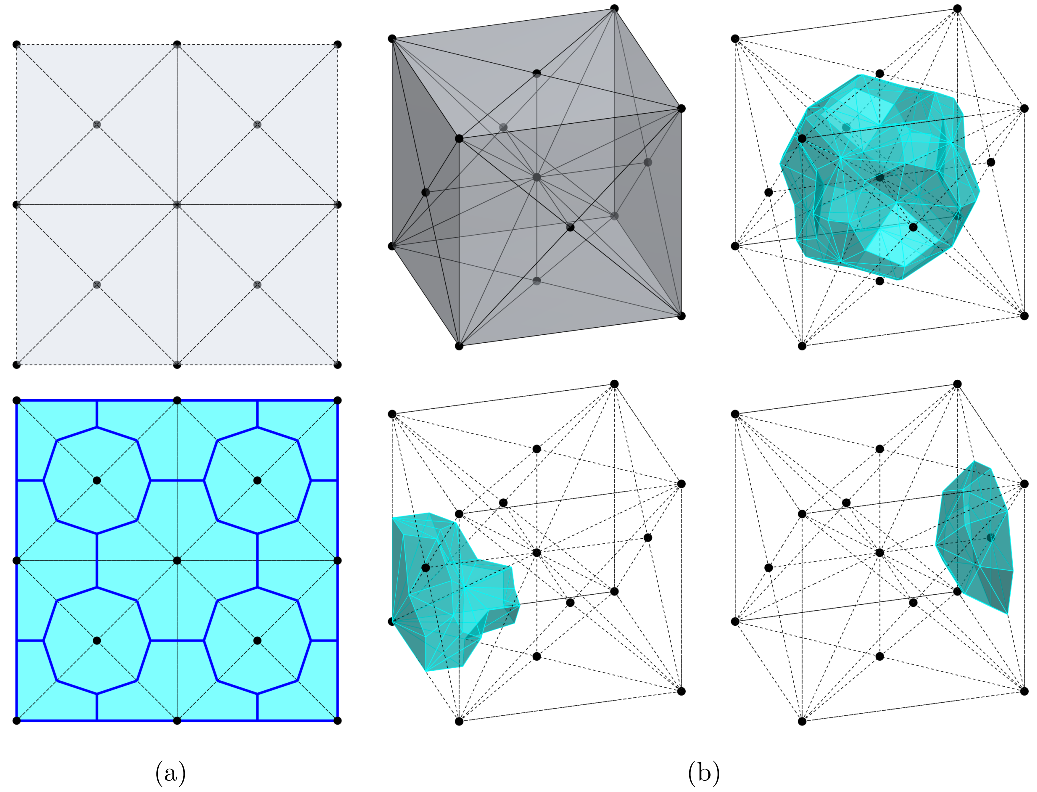


Figure. 4.1 A schematic of the partition of the FV domain into (a) straight-sided triangular elements (top) and control volumes (bottom) in 2D and (b) planar-faced tetrahedral elements (top left) and a representation of various control volumes corresponding to nodes in the interior and on the boundaries of the domain in 3D.

than their cell-centred counterparts for high anisotropy or distortion in the mesh. Also, it does not suffer from non-orthogonality in the discretisation of second-order terms. Moreover, with the nodes lying on the boundaries, vertex-centred approach makes the imposition of boundary conditions straight-forward. Unique definition of unit outward normal vector on each boundary facet implies a natural imposition of the inflow/outflow boundary conditions. Additionally, for tetrahedral meshes the number of elements is usually 5.5 times the number of nodes. And with dual mesh construction, the scheme is efficient in memory usage since coefficients/weights need to be computed and stored only once per edge.

Following the construction of the dual mesh, a few nomenclatures related to the edge-based formulation are introduced. Let Ω_i^2 be the control volume of an arbitrary node i among the $\mathtt{n}_{\mathtt{cv}}$ mesh nodes of the FV domain. The $\mathtt{n}_{\mathtt{cv}}$ disjoint control volumes are then a partition of Ω^2,

$$\overline{\Omega}^2 = \bigcup_{i=1}^{\mathtt{n_{cv}}} \overline{\Omega}_i^2, \quad \Omega_i^2 \cap \Omega_j^2 = \emptyset \quad \text{for } i \neq j. \tag{4.1.2}$$

As illustrated in Figure 4.1, a set of facets $\{\Upsilon_i\}$ of size $\mathtt{n_{vf}}$, made up of straight edges in 2D and planar triangular faces in 3D, define the control volume boundary $\partial\Omega_i^2$ such that

$$\partial\Omega_i^2 = \bigcup_{k=1}^{\mathtt{n_{vf}}} \Upsilon_i|_k. \tag{4.1.3}$$

In 2D, each straight edge facet is obtained by connecting the edge centre and the element centroid. In 3D, each triangular facet is obtained by connecting the edge centre, face centroid and the element centroid. Note that each facet in the dual mesh can be associated with a particular edge connecting two nodes.

Let Λ_i be the set of all nodes connected to node i through an edge. For a boundary node, a subset Λ_i^∂ denotes the set of nodes connected to i through an edge lying on the boundary $\partial\Omega^2$. A set of facets $\{\Upsilon_{i,j}\}$ and $\{\Upsilon_{i,j}^\partial\}$ can be identified as those associated with the edge connecting nodes i and j and lying in the interior and the boundary of the FV domain respectively. Figure 4.2 illustrates interior facets (in cyan and magenta) and boundary facets (in green) for an interior edge (in dotted blue) and a boundary edge (in dotted red) respectively. In particular,

$$\Upsilon_{i,j} = \left\{\Upsilon,\ \Upsilon \in \{\Upsilon_i\},\ \Upsilon \in \{\Upsilon_j\}, \Upsilon \cap \partial\Omega^2 = \emptyset\right\}, \tag{4.1.4a}$$

$$\Upsilon_{i,j}^\partial = \left\{\Upsilon,\ \Upsilon \in \{\Upsilon_i, \Upsilon_j\}, \Upsilon \in \partial\Omega^2\right\}. \tag{4.1.4b}$$

4.2 The vertex-centred finite volume method

The strong form in Equation (2.1.6) of the Euler equations in the FV domain is written as,

$$\begin{aligned} \boldsymbol{V}_t + \boldsymbol{\nabla}\cdot\boldsymbol{F}(\boldsymbol{V}) &= \boldsymbol{S} && \text{in } \Omega^2 \times (0,T], && (4.2.1a)\\ \boldsymbol{V} &= \boldsymbol{\mathcal{U}}_0 && \text{in } \Omega^2 \times \{0\}, && (4.2.1b)\\ \boldsymbol{B}(\boldsymbol{V},\boldsymbol{\mathcal{U}}^\infty) &= \boldsymbol{0} && \text{on } \left(\partial\Omega^2 \cap \partial\Omega\right) \times (0,T], && (4.2.1c) \end{aligned}$$

where, $\boldsymbol{V} = \boldsymbol{\mathcal{U}}|_{\Omega^2}$. Using the definitions of the functional spaces defined in Equation (3.1.8), the weak form of the FV problem in Equation (4.2.1) after applying the

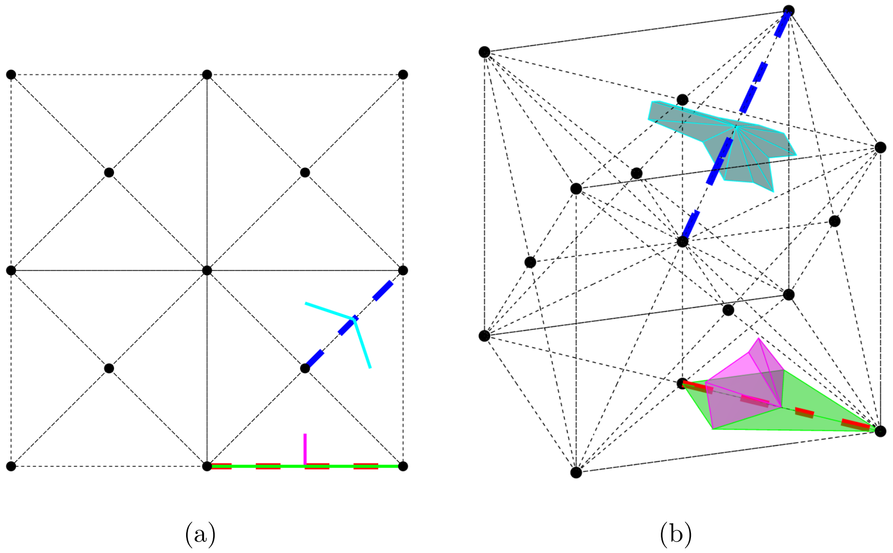


Figure. 4.2 An illustration of interior facets $\{\Upsilon_{i,j}\}$ (in cyan and magenta) and boundary facets $\{\Upsilon_{i,j,\partial}\}$ (in green) for an interior edge (in dotted blue) and a boundary edge (in dotted red) for (a) 2D and (b) 3D FV domains respectively.

divergence theorem is: find $\boldsymbol{V} \in [\mathcal{W}_t^{h,0}(\Omega_i^2)]^{\mathtt{m_{sd}}}$ that satisfies

$$\Big(\delta\boldsymbol{V}, \boldsymbol{V}_t\Big)_{\Omega_i^2} + \Big\langle\delta\boldsymbol{V}, \boldsymbol{F}(\boldsymbol{V})\cdot\boldsymbol{n}\Big\rangle_{\partial\Omega_i^2\setminus\partial\Omega} = \Big(\delta\boldsymbol{V}, \boldsymbol{S}\Big)_{\Omega_i^2}, \tag{4.2.2}$$

for all $\delta\boldsymbol{V} \in [\mathcal{W}^{h,0}(\Omega_i^2)]^{\mathtt{m_{sd}}}$ and for $i = 1, \ldots, \mathtt{n_{cv}}$. Now the test functions are chosen to be of unit value inside each control volume and zero elsewhere in the domain, which is equivalent to performing the integration of the Euler equations in each of the $\mathtt{n_{cv}}$ control volumes. Thus conservation of variables are expressed in finite control volumes, just as the strong form in Equation (4.2.1) is the expression for conservation in an infinitesimal volume in Ω^2. Also, the boundary conditions in Equation (4.2.1c) are enforced strongly, see Section 4.7.

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

$$\boldsymbol{V}(\boldsymbol{x},t) \approx \boldsymbol{V}_h(\boldsymbol{x},t) := \begin{cases} \mathbf{V}_i(t) & \text{if } \boldsymbol{x} \in \Omega_i^2 \\ 0 & \text{otherwise} \end{cases}, \tag{4.2.3}$$

thus transforming the weak form in Equation (4.2.2) to, find $\boldsymbol{V} \in [\mathcal{W}_t^{h,0}(\Omega_i^2)]^{\mathtt{m_{sd}}}$ that satisfies

$$\Big(1, \boldsymbol{V}_t\Big)_{\Omega_i^2} + \Big\langle 1, \boldsymbol{F}(\boldsymbol{V}) \cdot \boldsymbol{n} \Big\rangle_{\partial\Omega_i^2 \backslash \partial\Omega} = \Big(1, \boldsymbol{S}\Big)_{\Omega_i^2}, \tag{4.2.4}$$

for $i = 1, \dots, \mathtt{n_{cv}}$. The vector of unknowns $\mathbf{V}$ which is composed of nodal unknowns $\mathbf{V}_i$ is of the size $\mathtt{n_{cv}} \times \mathtt{m_{sd}}$.

The treatment of volume integrals is now straight-forward:

$$\Big(1, \boldsymbol{V}_t\Big)_{\Omega_i^2} = |\Omega_i^2| \, (\mathbf{V}_t)_i \tag{4.2.5a}$$

$$\Big(1, \boldsymbol{S}\Big)_{\Omega_i^2} = |\Omega_i^2| \, \mathbf{S}_i \tag{4.2.5b}$$

4.2.1 Treatment of the inviscid fluxes

The boundary integral for each control volume i is performed over its facets $\{\Upsilon_i\}$. Since an edge-based formulation is undertaken in the present study, the surface integrals of the invisid fluxes are computed by looping over the edges of the mesh. For each edge, the contributions are expressed as a sum of the integration over internal facets $\Upsilon_{i,j}$ and boundary facets $\Upsilon_{i,j}^{\partial}$. The inviscid flux in the weak form in Equation (4.2.2) is,

$$\Big\langle 1, \boldsymbol{F}_l(\boldsymbol{V})\, n_l \Big\rangle_{\partial\Omega_i^2 \backslash \partial\Omega} \approx \sum_{j \in \Lambda_i} \Big(C_l^{i,j} \, \tilde{\mathbf{F}}_l^{i,j} \Big) + \sum_{j \in \Lambda_i^{\partial}} \Big(C_l^{i,j,\partial} \, \tilde{\mathbf{F}}_l^{i,j,\partial} \Big), \tag{4.2.6}$$

for $i = 1, \dots, \mathtt{n_{cv}}$. Here the flux coefficients $C_l^{i,j}$ and $C_l^{i,j,\partial}$ are given by,

$$C_l^{i,j} = \sum_{\Upsilon_i \in \Upsilon_{i,j}} |\Upsilon_i| \, n_l^{\Upsilon_i}, \tag{4.2.7a}$$

$$C_l^{i,j,\partial} = \sum_{\Upsilon_i \in \Upsilon_{i,j}^{\partial}} |\Upsilon_i| \, n_l^{\Upsilon_i}, \tag{4.2.7b}$$

where $\boldsymbol{n}^{\Upsilon_i}$ is the unit outward normal to Υ_i. The flux coefficients are purely geometrical constants and thus only need to be recomputed after re-meshing or mesh movement. The discrete interior flux $\tilde{\mathbf{F}}_l^{i,j}$ and boundary flux $\tilde{\mathbf{F}}_l^{i,j,\partial}$ need to be expressed in terms of discrete conservation variables $\mathbf{V}$. Various definitions are available in the literature, popular ones being Lax-Friedrich, Lax-Wendroff, upwind methods, whose choice directly decides the stability and accuracy of the method. To obtain a second-order central difference type scheme, we use trapezoidal rule across the edge as,

$$\tilde{\mathbf{F}}_l^{i,j} = \frac{\boldsymbol{F}_l\left(\mathbf{V}_i\right) + \boldsymbol{F}_l\left(\mathbf{V}_j\right)}{2}, \tag{4.2.8a}$$

$$\tilde{\mathbf{F}}_l^{i,j,\partial} = \frac{3\boldsymbol{F}_l\left(\mathbf{V}_i\right) + \boldsymbol{F}_l\left(\mathbf{V}_j\right)}{4}. \tag{4.2.8b}$$

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.

Remark 4 (Consistency and conservation)**.** *It is clear from the definitions of the flux coefficients in Equation* (4.2.7) *that* $C_l^{i,j} = -C_l^{j,i}$ *and* $C_l^{i,j,\partial} = C_l^{j,i,\partial}$*. These important properties highlight the advantage of edge-based formulation where memory requirement for storage is minimal and the integration of surface fluxes over large number of facets for a pair of nodes can be performed efficiently. Moreover, the scheme is consistent, i.e., for constant flux, the surface integral is*

$$\sum_{j\in\Lambda_i} C_l^{i,j} + \sum_{j\in\Lambda_i^\partial} C_l^{i,j,\partial} = 0, \tag{4.2.9}$$

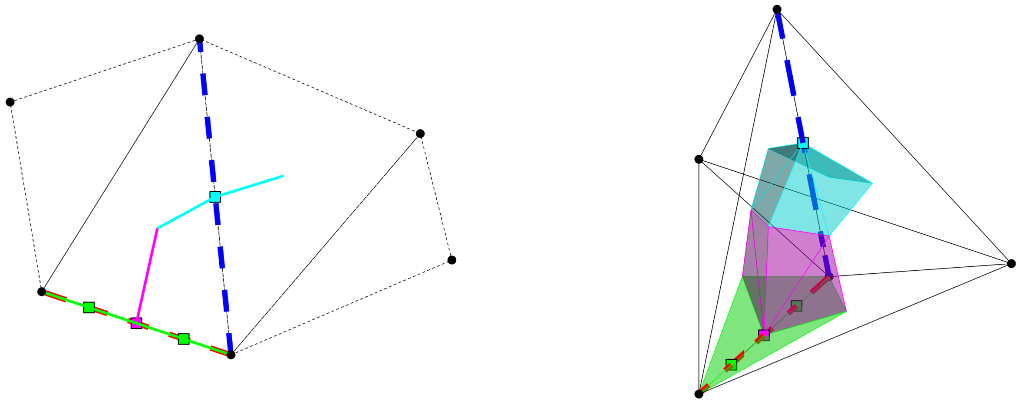
and that conservation of inviscid fluxes is maintained, i.e.,

$$\sum_{i=1}^{n_{cv}}\left(\sum_{j\in\Lambda_i}\left(C_l^{i,j}\,\tilde{\mathbf{F}}_l^{i,j}\right) + \sum_{j\in\Lambda_i^\partial}\left(C_l^{i,j,\partial}\,\tilde{\mathbf{F}}_l^{i,j,\partial}\right)\right) = \sum_{i=1}^{n_{cv}}\sum_{j\in\Lambda_i^\partial}\left(C_l^{i,j,\partial}\,\tilde{\mathbf{F}}_l^{i,j,\partial}\right). \tag{4.2.10}$$

Remark 5 (Alternative method of boundary flux integration)**.** *An alternative to the boundary flux definition for edge* i,j *in Equation* (4.2.8b)*, where the flux is assumed to be constant per* $(\Upsilon_i \cap \Upsilon_{i,j,\partial})$*, is to simplify by assuming the flux to be constant per* $(\Upsilon_i \cap \partial\Omega_i^2)$*, resulting in the expression,*

$$\tilde{\mathbf{F}}_l^{i,j,\partial} = \boldsymbol{F}_l\left(\mathbf{V}_i\right). \tag{4.2.11}$$

Remark 6 (Edge-based flux interpolation)**.** *The edge-based formulation leads to the flux being interpolated using only the values at the edge nodes. In other words, as indicated in Figure 4.3, across the length/over the surface of all the internal facets (shown in blue and magenta), the flux is assumed to be constant and equal to the edge-wise interpolated flux at the edge centre (blue and magenta □). Similarly for boundary facets (shown in green), the flux is also assumed to be constant and equal to the edge-wise interpolated flux at the midpoint (green □) connecting the node and*



(a)

(b)

Figure. 4.3 An illustration of the edge-based flux interpolation (located at □) for interior facets (in cyan and magenta) and boundary facets (in green) for an interior edge (in dotted blue) and a boundary edge (in dotted red) for (a) 2D and (b) 3D FV domains respectively.

the edge centre. Thus, the flux coefficients need to be stored only once per edge, thus enabling the edge-based formulation.

Second-order central difference schemes are computationally cheaper and more accurate compared to upwind methods but allow for odd-even decoupling of solution (two independent discrete solutions) and oscillations for sharp gradients [66, 117]. It is thus unstable for hyperbolic problems and stabilisation is necessary while retaining the second-order accuracy. This is achieved by the addition of artificial dissipation. New developments for stabilising hyperbolic equations using boundary conditions have been reported [1].

4.3 Artificial dissipation and shock capturing method

Solving hyperbolic Euler equations with second-order central difference schemes generates severe oscillations in the regions of sharp gradients such as shocks and discontinuities. Central difference stencil is unable to resolve changes in wave propagation direction that occur at such discontinuities. Remedy to this stems from the understanding that by locally reducing the order of approximation, one can smooth out high-order frequencies of oscillation. But one needs to be mindful of excessively smoothing the solution resulting in the loss of flow features. Lax Friedrich scheme of first order is an example of such a situation where large smoothing in the whole domain is observed since the procedure resembles the addition of second-order viscous term which is introduced as a consequence of the discretisation. Artificial dissipation involves the addition of a new term that mimics the effect of viscosity to smooth the solution. But the term is of such order that it becomes negligible in regions away from discontinuities. This can be ensured by taking the order to be higher than that of the scheme (> 2 for central differences). Upwind and TVD schemes have been shown to be equivalent in form to central difference schemes with artificial dissipation [64, 65].

Artificial dissipation of the Jameson-Schmidt-Turkel (JST) kind is applied [75]. A discrete third-order bi-harmonic operator term resulting in fourth-order differences as shown below is introduced,

$$\mathbf{P}_i\left(\mathbf{V}\right) = \sum_{j \in \Lambda_i} \mathbf{M}_{i,j}\left(\mathbf{E}_j\left(\mathbf{V}\right) - \mathbf{E}_i\left(\mathbf{V}\right)\right), \tag{4.3.1}$$

where, the difference in discrete nodal conservative variables $\mathbf{E}\left(\mathbf{V}\right)$ is ,

$$\mathbf{E}_i\left(\mathbf{V}\right) = \frac{1}{|\Lambda_i|} \sum_{k \in \Lambda_i} \left(\mathbf{V}_k - \mathbf{V}_i\right), \tag{4.3.2}$$

and the coefficient $\mathbf{M}_{i,j}$ is given by,

$$\mathbf{M}_{i,j} = \mathbf{m}_{i,j}\alpha_{i,j}, \tag{4.3.3}$$

where,

$$\mathbf{m}_{i,j} = \max\left(\eta_4 - \eta_2 s_{i,j}, 0\right)\mathbf{I}_{\mathtt{msd}}, \tag{4.3.4}$$

$$\alpha_{i,j} = \frac{1}{|\Lambda_i| + |\Lambda_j|}\min\left(\frac{\Omega_i^2}{\Delta\tau^i}, \frac{\Omega_j^2}{\Delta\tau^j}\right). \tag{4.3.5}$$

The user-defined scalar constants η_2, η_4 are second order (harmonic) and fourth order (biharmonic) dissipation factors and taken to be 0.2 throughout this work. The values for the scalar coefficients are industry-standard and widely-utilised. They are obtained from validation studies for external compressible flow simulations [146, 144]. $\Delta\tau_i$ is the local time step size for the node i and the pressure sensor $s_{i,j}$ is given by,

$$s_{i,j} = \max\left(|\Delta p_i|, |\Delta p_j|\right), \tag{4.3.6}$$

$$\Delta p_i = 12\,\frac{\sum_{k \in \Lambda_i}\left(p_k - p_i\right)}{\sum_{k \in \Lambda_i}\left(p_k + p_i\right)}, \tag{4.3.7}$$

where p_i is the pressure at node i. Note that $\mathbf{m}_{i,j} = \mathbf{m}_{j,i}$. In smooth convective flows, only the fourth-order differences are active, providing stability to the central difference scheme. For strong shocks and discontinuities, dissipation introduced by Equation (4.3.1) is not sufficient to stabilise the method. Hence a shock capturing method is made available. This gets activated by the pressure sensor in Equation (4.3.6) close to sharp gradients. Note that the fourth-order difference term is suppressed, since they cause perturbations if active near shocks.

Oscillations near shocks, where $s_{i,j}$ is large, can be mitigated with a local first-order harmonic shock capturing term. This results in a second-order differences term,

$$\mathbf{Q}_i\left(\mathbf{V}\right) = \sum_{j\in\Lambda_i} \eta_2 \alpha_{i,j} s_{i,j}\left(\mathbf{V}_j - \mathbf{V}_i\right). \tag{4.3.8}$$

Close to shocks, the scheme locally reduces to being first-order accurate, thus suppressing high-order oscillation inducing terms.

4.4 Time integration method

The discrete system in the present work is time marched in an implicit manner. In the FV community, for unsteady flows, dual time stepping is a popular choice for temporal integration[72]. It involves a physical time step, marched usually with a BDF or Runge-Kutta method, within which a fictitious time stepping is undertaken using acceleration methods designed for steady state solutions such as the local time stepping method. In the present study, BDF method of the first and second orders, described previously in Section 3.4, is chosen for its simplicity, accuracy and stability, as well the ease for combining with the HDG method later on. Following the expression for BDF approximation in Equation (3.4.1), the temporal derivative in Equation (4.2.5a) becomes,

$$\left(1, \boldsymbol{V}_t\right)_{\Omega_i^2} \approx |\Omega_i^2| \sum_{s=0}^{\mathrm{n_{tg}}} a_s \mathbf{V}_i^{n+1-s}. \tag{4.4.1}$$

4.5 Discrete system

The fully discrete system of non-linear equations for Equation (4.2.4) with volume integrals in Equation (4.2.5) and surface integrals in Equation (4.2.6) along with artificial dissipation in Equation (4.3.1) and shock capturing term in Equation (4.3.8) is expressed as a non-linear residual,

$$\boldsymbol{\mathcal{R}}(\mathbf{V}^{n+1},\ldots,\mathbf{V}^{n+1-\mathtt{n_{tg}}}) = \mathbf{0}, \tag{4.5.1}$$

which is obtained by the assembly of nodal contributions $\boldsymbol{\mathcal{R}}_i$ given by,

$$\begin{aligned}\boldsymbol{\mathcal{R}}_i :=\ & |\Omega_i^2| \sum_{s=0}^{\mathtt{n_{tg}}} a_s \mathbf{V}_i^{n+1-s} + \sum_{j\in\Lambda_i} \left(C_l^{i,j} \left(\frac{\boldsymbol{F}_l\left(\mathbf{V}_i^{n+1}\right) + \boldsymbol{F}_l\left(\mathbf{V}_j^{n+1}\right)}{2} \right)\right) \\ & + \sum_{j\in\Lambda_i^{\partial}} \left(C_l^{i,j,\partial} \left(\frac{3\boldsymbol{F}_l\left(\mathbf{V}_i^{n+1}\right) + \boldsymbol{F}_l\left(\mathbf{V}_j^{n+1}\right)}{4} \right)\right) \\ & + \sum_{j\in\Lambda_i} \mathbf{M}_{i,j}^n \left(\mathbf{E}_j\left(\mathbf{V}^{n+1}\right) - \mathbf{E}_i\left(\mathbf{V}^{n+1}\right)\right) \\ & - \sum_{j\in\Lambda_i} \eta_2 \alpha_{i,j}^n s_{i,j}^n \left(\mathbf{V}_j^{n+1} - \mathbf{V}_i^{n+1}\right) - |\Omega_i^2|\, \mathbf{S}_i^{n+1},\end{aligned} \tag{4.5.2a}$$

for $i = 1,\ldots,\mathtt{n_{cv}}$. Note that certain terms in the artificial dissipation and discontinuity capturing expressions are evaluated at the known time step n. This reduces the complexity of the linearisation terms and was observed not to adversely affect the stability and solution accuracy, even with large time step sizes for the examples demonstrated in the present work.

4.6 Linearisation method

As previously explained in Section (3.7) for the HDG method, linearisation of the non-linear system of equations in Equation (4.5.1) is carried out using the Newton-Raphson method. Due to the compact support of FV stencil, a sparse linear system of equations to be solved at each Newton iteration (q) arise, given by,

$$\left[\mathbf{T}_{vv}\right]^{n+1,q} \left\{\Delta\mathbf{V}\right\}^{n+1,q} = \left\{\mathbf{f}_v\right\}^{n+1,q}, \tag{4.6.1}$$

where, as usual, $\Delta\circledcirc^{n+1,q} = \circledcirc^{n+1,q+1} - \circledcirc^{n+1,q}$ denotes the increment of the vector of degrees of freedom. The right hand side vector is given by,

$$\left(\mathbf{f}_v^i\right)^{n+1,q} := -\boldsymbol{\mathcal{R}}^i. \tag{4.6.2}$$

The tangent matrix is given by the expression,

$$
\begin{aligned}
\left(\mathbf{T}_{vv}\right)_J^{i,n+1,q} := \frac{\partial \boldsymbol{\mathcal{R}}^i}{\partial \mathbf{V}_J^{n+1,q}} &= |\Omega_i^2| \sum_{s=0}^{\mathtt{n_{tg}}} a_s \frac{\partial \mathbf{V}_i^{n+1-s}}{\partial \mathbf{V}_J^{n+1,q}} \\
&+ \sum_{j\in\Lambda_i} \left(C_l^{i,j} \left(\frac{\boldsymbol{A}_l\left(\mathbf{V}_i^{n+1}\right) \dfrac{\partial \mathbf{V}_i^{n+1}}{\partial \mathbf{V}_J^{n+1,q}} + \boldsymbol{A}_l\left(\mathbf{V}_j^{n+1}\right) \dfrac{\partial \mathbf{V}_j^{n+1}}{\partial \mathbf{V}_J^{n+1,q}}}{2} \right) \right) \\
&+ \sum_{j\in\Lambda_i^{\partial}} \left(C_l^{i,j,\partial} \left(\frac{3\boldsymbol{A}_l\left(\mathbf{V}_i^{n+1}\right) \dfrac{\partial \mathbf{V}_i^{n+1}}{\partial \mathbf{V}_J^{n+1,q}} + \boldsymbol{A}_l\left(\mathbf{V}_j^{n+1}\right) \dfrac{\partial \mathbf{V}_j^{n+1}}{\partial \mathbf{V}_J^{n+1,q}}}{4} \right) \right) \\
&+ \sum_{j\in\Lambda_i} \mathbf{M}_{i,j}^n \left(\frac{\partial \mathbf{E}_j\left(\mathbf{V}^{n+1}\right)}{\partial \mathbf{V}_J^{n+1,q}} - \frac{\partial \mathbf{E}_i\left(\mathbf{V}^{n+1}\right)}{\partial \mathbf{V}_J^{n+1,q}} \right) \\
&- \sum_{j\in\Lambda_i} \eta_2 \alpha_{i,j}^n s_{i,j}^n \left(\frac{\partial \mathbf{V}_j^{n+1}}{\partial \mathbf{V}_J^{n+1,q}} - \frac{\partial \mathbf{V}_i^{n+1}}{\partial \mathbf{V}_J^{n+1,q}} \right) - |\Omega_i^2| \frac{\partial \mathbf{S}_i^{n+1}}{\partial \mathbf{V}_J^{n+1,q}},
\end{aligned}
\tag{4.6.3a}
$$

where, $\boldsymbol{A}_l$ for $l = 1, \ldots, \mathtt{n_{sd}}$ is the jacobian of the inviscid flux, as given in Equation (2.1.15). And $\dfrac{\partial \mathbf{V}_i}{\partial \mathbf{V}_J} = \delta_{iJ} \mathbf{I}_{\mathtt{m_{sd}}}$, where δ_{iJ} is the Kronecker delta.

The number of contributions to the tangent matrix given by the Equation (4.6.3) for any node depends on the dimensionality of the problem $\mathtt{m_{sd}}$. However, for an individual node, immediate neighbouring nodes along with their neighbours provide non-zero contributions. In a 2D problem with triangular elements, the tangent matrix for each of the $\mathtt{m_{sd}} = 4$ unknown degrees of freedom at a node receives a total of 76 non-zero contributions. Similarly, for 3D problems with tetrahedral elements, the tangent matrix for each of the $\mathtt{m_{sd}} = 5$ degrees of freedom at a node receives a total of 705 non-zero contributions.

4.7 Boundary conditions

Boundary conditions in the FV method are imposed strongly on the updated solution field $\mathbf{V}^{n+1}$. As explained earlier in Section 3.2.3, two distinct types of boundaries, namely, far-field and slip wall boundaries, need to be treated.

On a far-field boundary, the local Mach number M and the flow direction normal to the boundary influence the manner of imposition of the boundary condition, which is based on the characteristics entering or leaving the domain [89]. For supersonic flows

$(M > 1)$, at the inflow boundary $(v_n \leq 0)$,

$$\mathbf{V}_i^{n+1} = \boldsymbol{\mathcal{U}}^{\infty}(\boldsymbol{x}_i, t^{n+1}) \tag{4.7.1}$$

and at an outflow boundary $(v_n > 0)$, no conditions are necessary since the solution depends only on the interior nodes. For subsonic flows $(M \leq 1)$, at the inflow boundaries,

$$\left.\begin{aligned} \rho &= \rho^{\infty}, \\ \boldsymbol{v} &= \boldsymbol{v}^{\infty}, \\ E &= \tfrac{p}{\rho^{\infty}(\gamma-1)} + \tfrac{\boldsymbol{v}^{\infty}\cdot\boldsymbol{v}^{\infty}}{2}, \end{aligned}\right\} \tag{4.7.2}$$

and at outflow boundaries,

$$E = \frac{p^{\infty}}{\rho(\gamma - 1)} + \frac{\boldsymbol{v}\cdot\boldsymbol{v}}{2}. \tag{4.7.3}$$

Slip wall condition dictates that the flow remain tangential to the wall and is imposed by modifying the equations for the updated solution as,

$$\mathbf{V} = \begin{Bmatrix} \rho \\ \rho(\boldsymbol{v} - v_n\boldsymbol{n}) \\ \rho E \end{Bmatrix}. \tag{4.7.4}$$

4.8 Implementation details

The convergence of the Newton-Raphson linearisation is monitored through the $\mathcal{L}_2$ norms of the residual and the solution increment as follows,

$$||\boldsymbol{\epsilon}_{R^i}||_{\mathcal{L}_2(\Omega^2)} := \sqrt{\sum_{i=1}^{\mathtt{n_{cv}}} \int_{\Omega_i^2} \boldsymbol{\mathcal{R}}^i \cdot \boldsymbol{\mathcal{R}}^i \, \partial\Omega}, \tag{4.8.1a}$$

$$||\boldsymbol{\epsilon}_{\Delta V}||_{\mathcal{L}_2(\Omega^2)} := \frac{\sqrt{\int_{\Omega^2} \Delta\boldsymbol{V}^{n+1,q} \cdot \Delta\boldsymbol{V}^{n+1,q} \, \partial\Omega}}{\sqrt{\int_{\Omega^2} \boldsymbol{V}^{n+1,q+1} \cdot \boldsymbol{V}^{n+1,q+1} \, \partial\Omega}}. \tag{4.8.1b}$$

The typical quadratic convergence of the iterations errors is illustrated in Figure 4.4 where the maximum of the $\mathtt{m_{sd}}$ components of the error norms in Equation (4.8.1) is plotted against the iteration number. This demonstrates consistent linearisation of the non-linear terms for the FV method. In all the examples, the tolerance for convergence is set to 1.0×10^{-10}, unless specified otherwise.

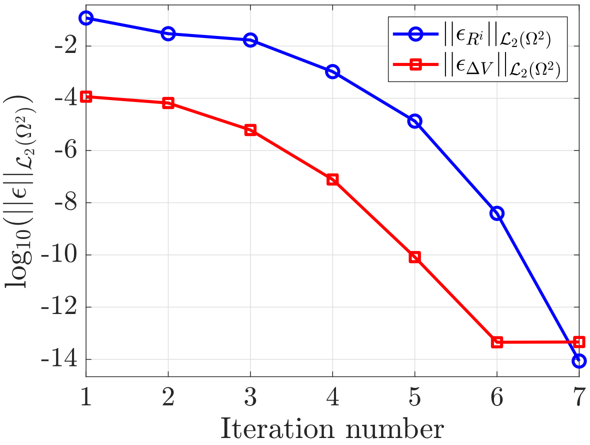


Figure. 4.4 Quadratic convergence of Newton-Raphson method demonstrating consistent linearisation of the non-linear system of equations for FV method.

The choice of the solution procedure for the sparse linear system of equations follows the discussion previously described in Section 3.9. A pseudo-code laying out the solution procedure of the FV method is provided in Algorithm 2.

4.9 Numerical tests

Testing for the convergence of the discretisation error is the standard methodology to assess the solution accuracy of the discretisation scheme, both in spatial and temporal domains. Manufactured solutions is a popular method for such tests. As applied to the HDG problem in Section 3.11, both manufactured solution test and the Ringleb flow problem are used to test the FV discretisation. The solution error $\boldsymbol{\epsilon}$ in the $\mathcal{L}_2(\Omega^2)$ is computed using the relation,

$$||\boldsymbol{\epsilon}||_{\mathcal{L}_2(\Omega^2)} := \frac{\sqrt{\int_{\Omega^2} \left(\boldsymbol{V} - \boldsymbol{\mathcal{U}}_a\right)^2 \, d\Omega}}{\sqrt{\int_{\Omega^2} \boldsymbol{\mathcal{U}}_a^2 \, d\Omega}}, \tag{4.9.1}$$

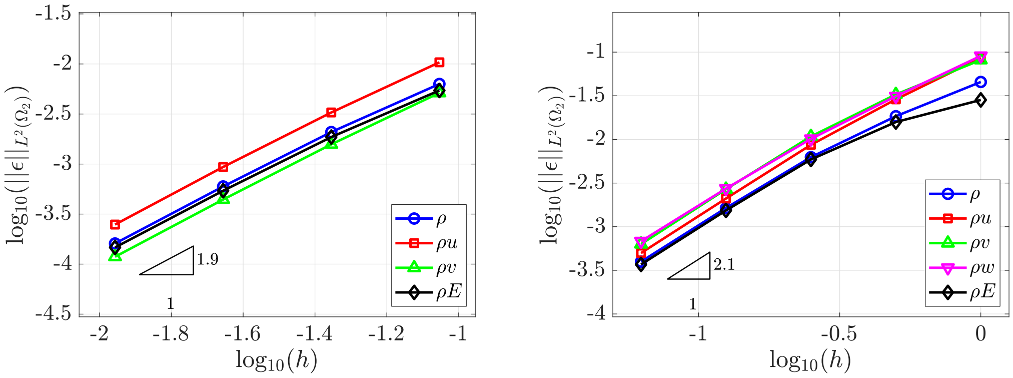
where $\boldsymbol{\mathcal{U}}_a$ is a suitably designed analytical solution.

4.9.1 Spatial convergence tests

The spatial convergence test in 2D involves solving the Ringleb problem in four uniform meshes of the domain $\Omega^2 = [0\ 1]^2$ with 256, 1024, 4096 and 16 384 triangular elements respectively, as shown in Figure 3.3. For the spatial convergence test in

```
Initial solution: $\boldsymbol{V}^{n=0}$
Time stepping
for $t^{n+1}$, $n = 0$ to $end - 1$ do
    Data: $\boldsymbol{V}^{n}$
    Initial guess: $\boldsymbol{V}^{n+1,q=0} = \boldsymbol{V}^{n}$
    Newton-Raphson linearisation
    repeat
        Data: $\boldsymbol{V}^{n+1,q}, \ldots, \boldsymbol{V}^{n+1-\mathtt{n}_{\mathtt{tg}}}$
        Edge-based approach
        for $e = 1$ to $\mathsf{n}_{\mathsf{me}}$ do
            for $i = 1$ to $2$ do
                Compute matrix: $(\mathbf{T}_{vv})_{J}^{i,n+1,q}$ in Equation (4.6.3)
                Compute vector: $(\mathbf{f}_{v}^{i})^{n+1,q}$ in Equation (4.6.2)
                Assemble to Equation (4.6.1)
            end
        end
        Solve linear system, obtain $\{\Delta \boldsymbol{V}\}^{n+1,q}$ in Equation (4.6.1)
        Update: $\boldsymbol{V}^{n+1,q+1} = \boldsymbol{V}^{n+1,q} + \{\Delta \boldsymbol{V}\}^{n+1,q}$
        Compute residuals: $\boldsymbol{\mathcal{R}}^{n+1,q+1}$ from Equation (4.5.1)
        Update: $q = q + 1$
        Compute errors in Equation (4.8.1)
    until $||\boldsymbol{\epsilon}_{R^{i}}||_{\mathcal{L}_2(\Omega^2)}$ and $||\boldsymbol{\epsilon}_{\Delta V}||_{\mathcal{L}_2(\Omega^2)}, \leq$ tolerance;
    Update solution: $\boldsymbol{V}^{n+1} = \boldsymbol{V}^{n+1,q+1}$.
end
```

Algorithm 2: FV solution procedure



(a) (b)

Figure. 4.5 The optimal rates of convergence for the FV method as shown in the plots of the $\mathcal{L}_2(\Omega^2)$ of the error in all the conservative variables versus the mesh size h, for the (a) Ringleb flow in 2D and (b) manufactured solution in Equation (3.11.2).

3D, a manufactured solution is constructed as indicated in Equation (3.11.2). Five uniform meshes of the domain $\Omega^2 = [0\ 1]^3$ are considered, with 24, 192, 1536, 12 288 and 98 304 tetrahedral elements respectively, as shown in Figure 3.4 for the first four meshes. As usual, nodally exact initial solution is constructed and time marched until the steady state is reached. The solution error in Equation (3.11.1) clearly observed to be decreasing in the nearly optimal convergence rate for second-order central difference scheme with an asymptotic slope of 2 for decreasing mesh sizes in both 2D and 3D, see Figure 4.5.

4.9.2 Temporal convergence tests

Convergence properties of the FV method for the BDF method of time integrations is presented. As the total discretisation error includes the contributions of both spatial and temporal discretisations, a finer mesh is employed to minimise the influence of the spatial terms.

For the temporal convergence tests, the analytical solution in Equation (3.11.3) and Equation (3.11.4), for 2D and 3D cases respectively, were constructed. Initial solution in the form of nodally exact analytical solution was time marched up-to the same physical time with decreasing sizes of the time step size Δt. Figure 4.6 for 2D and Figure 4.7 demonstrate the convergence in the solution error at an optimal rate given by the order of the BDF method in all conservative variables, thus confirming an error-free implementation of the time marching method. Moreover, an order of magnitude

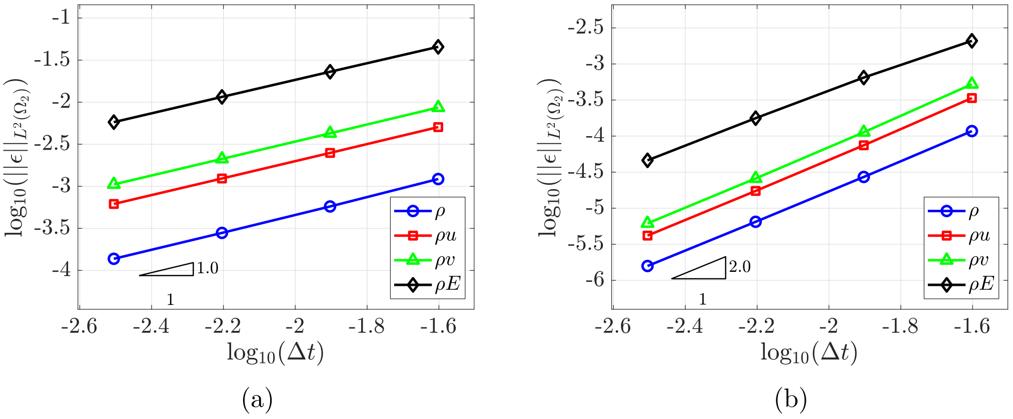
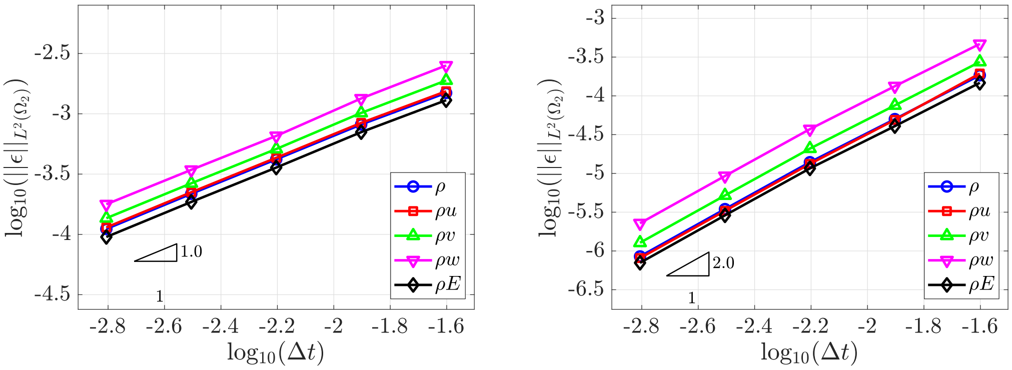


Figure. 4.6 The optimal rates of convergence for the FV method as shown in the plots of the $\mathcal{L}_2(\Omega^2)$ of the error in all the conservative variables versus the time step size Δt, for different orders of the BDF method, (a) BDF1 and (b) BDF2. The example demonstrated here is the 2D manufactured solution for temporal convergence given in Equation (3.11.3).

increase in the solution accuracy is obtained for BDF2 as compared to BDF1, thus highlighting the benefits gained from increasing the order of the temporal integration schemes.



(a)

(b)

Figure. 4.7 The optimal rates of convergence for the FV method as shown in the plots of the $\mathcal{L}_2(\Omega^2)$ of the error in all the conservative variables versus the time step size Δt, for different orders of the BDF method, (a) BDF1 and (b) BDF2. The example demonstrated here is the 3D manufactured solution for temporal convergence given in Equation (3.11.4).

Chapter 5

The coupled HDG-FV method

> *In the hard cases, the best algorithms are all about doing what makes the most sense in the least amount of time … These aren't the concessions we make when we can't be rational. They're what being rational means.*
>
> Brian Christian & Tom Griffiths
> Algorithms to live by

A coupling of the HDG method described in Chapter 3 and the vertex-centred FV method in Chapter 4 is presented. A partition of the computational domain Ω is carried out to define the HDG and FV subdomains. Euler equations are solved in both subdomains in a monolithic fashion. Continuity of the conservative variables and their fluxes across the interface between the subdomains, the *transmission conditions* in other words, define the coupling between the constitutive methods. The conditions are imposed, one per subdomain, thus creating a two-way coupling. Imposition of the *transmission conditions* is made easy by the use of matching meshes at the interface. The chapter begins with a preliminary description of the partitioning of the computational domain. The strong form of the Euler equation applied to the HDG and FV subdomains along with the *transmission conditions* is presented. In the next stage, a monolithic weak formulation is obtained. HDG and FV spatial discretisations and BDF time integration are introduced, resulting in the final non-linear residuals. A linearisation procedure which results in a linear system of equations, comprised of the assembly of various tangent matrices and right hand side vectors, is presented. Important details of the implementation, specifically regarding the handling of interface terms, are explained. Towards the end, the behaviour of coupled HDG-FV method in several numerical tests are discussed.

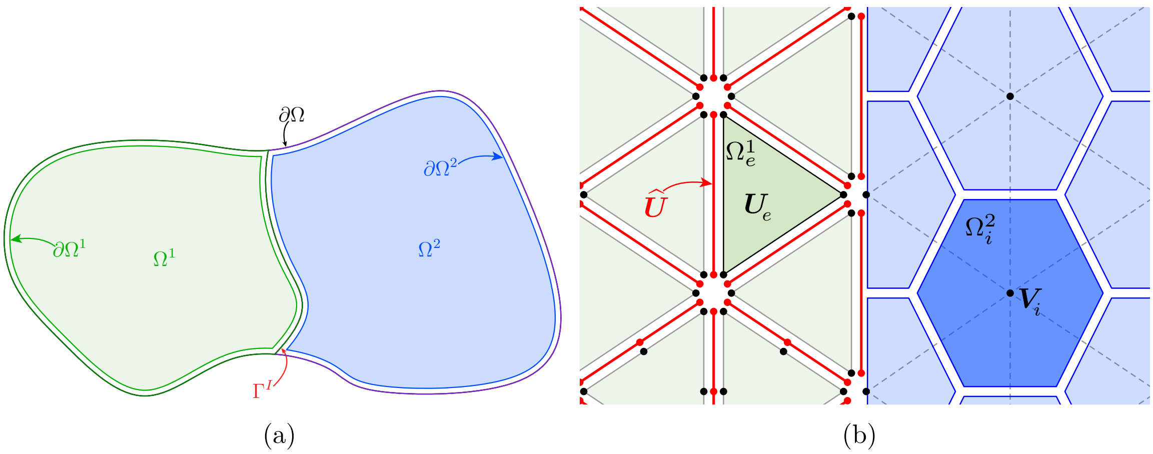


Figure. 5.1 A schematic of (a) the partition of the domain into the HDG and FV subdomains and (b) partition of the HDG subdomain in elements and the FV subdomain in medial dual control volumes.

5.1 Preliminaries

Consider an open bounded domain $\Omega \in \mathbb{R}^{\mathtt{n_{sd}}}$, with boundary $\partial\Omega$ and $\mathtt{n_{sd}}$ the number of spatial dimensions, 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 5.1(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^1_e, as given in Equation (3.1.1), with boundaries $\partial\Omega^1_e$, which define an internal interface Γ^1 in Equation (3.1.2). The subdomain Ω^2 is partitioned in $\mathtt{n_{cv}}$ disjoint control volumes Ω^2_i, as given in Equation (4.1.2), with the median dual approach. The partition of each subdomain into elements and control volumes respectively is shown in Figure 5.1(b) near the interface between the HDG and FV subdomains.

5.2 Coupled HDG-FV formulation

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

$$\boldsymbol{U}_t + \boldsymbol{\nabla\cdot F}(\boldsymbol{U}) = \boldsymbol{S} \qquad \text{in } \Omega^1_e \times (0,T], \tag{5.2.1a}$$

$$[\![\boldsymbol{U} \otimes \boldsymbol{n}]\!] = \boldsymbol{0} \qquad \text{on } \Gamma^1 \times (0,T], \tag{5.2.1b}$$

$$[\![\boldsymbol{F}(\boldsymbol{U}) \cdot \boldsymbol{n}]\!] = \boldsymbol{0} \qquad \text{on } \Gamma^1 \times (0,T], \tag{5.2.1c}$$

$$\boldsymbol{V}_t + \boldsymbol{\nabla\cdot F}(\boldsymbol{V}) = \boldsymbol{S} \qquad \text{in } \Omega^2 \times (0,T], \tag{5.2.1d}$$

$$[\![\boldsymbol{\mathcal{U}} \otimes \boldsymbol{n}]\!] = \boldsymbol{0} \qquad \text{on } \Gamma^I \times (0,T], \tag{5.2.1e}$$

$$[\![\boldsymbol{F}(\boldsymbol{\mathcal{U}}) \cdot \boldsymbol{n}]\!] = \boldsymbol{0} \qquad \text{on } \Gamma^I \times (0,T], \tag{5.2.1f}$$

$$\boldsymbol{U} = \boldsymbol{\mathcal{U}}_0 \qquad \text{in } \Omega^1_e \times \{0\}, \tag{5.2.1g}$$

$$\boldsymbol{V} = \boldsymbol{\mathcal{U}}_0 \qquad \text{in } \Omega^2 \times \{0\}, \tag{5.2.1h}$$

$$\boldsymbol{B}(\boldsymbol{U},\boldsymbol{\mathcal{U}}^\infty) = \boldsymbol{0} \qquad \text{on } \left(\partial\Omega^1_e \cap \partial\Omega\right) \times (0,T], \tag{5.2.1i}$$

$$\boldsymbol{B}(\boldsymbol{V},\boldsymbol{\mathcal{U}}^\infty) = \boldsymbol{0} \qquad \text{on } \left(\partial\Omega^2 \cap \partial\Omega\right) \times (0,T], \tag{5.2.1j}$$

for $e = 1, \dots, \mathtt{n_{el}}$, where $\boldsymbol{U} = \boldsymbol{\mathcal{U}}|_{\Omega^1}$, $\boldsymbol{V} = \boldsymbol{\mathcal{U}}|_{\Omega^2}$, Equations (5.2.1e) and (5.2.1f) are the so-called *transmission* conditions across the subdomain interfaces, which impose the continuity of the solution and the normal fluxes between the two subdomains Ω^1 and Ω^2, respectively.

Introducing the hybrid HDG variable $\widehat{\boldsymbol{U}}$ to the final strong form,

$$\boldsymbol{U}_t + \boldsymbol{\nabla\cdot F}(\boldsymbol{U}) = \boldsymbol{S} \qquad \text{in } \Omega^1_e \times (0,T], \tag{5.2.2a}$$

$$\boldsymbol{U} = \widehat{\boldsymbol{U}} \qquad \text{on } \partial\Omega^1_e \times (0,T], \tag{5.2.2b}$$

$$[\![\boldsymbol{F}(\boldsymbol{U}) \cdot \boldsymbol{n}]\!] = \boldsymbol{0} \qquad \text{on } \Gamma^1 \times (0,T], \tag{5.2.2c}$$

$$\boldsymbol{V}_t + \boldsymbol{\nabla\cdot F}(\boldsymbol{V}) = \boldsymbol{S} \qquad \text{in } \Omega^2 \times (0,T], \tag{5.2.2d}$$

$$[\![\![\boldsymbol{\mathcal{U}} \otimes \boldsymbol{n}]\!]\!] = \boldsymbol{0} \qquad \text{on } \Gamma^I \times (0,T], \tag{5.2.2e}$$

$$[\![\boldsymbol{F}(\boldsymbol{\mathcal{U}}) \cdot \boldsymbol{n}]\!] = \boldsymbol{0} \qquad \text{on } \Gamma^I \times (0,T], \tag{5.2.2f}$$

$$\boldsymbol{U} = \boldsymbol{\mathcal{U}}_0 \qquad \text{in } \Omega^1_e \times \{0\}, \tag{5.2.2g}$$

$$\boldsymbol{V} = \boldsymbol{\mathcal{U}}_0 \qquad \text{in } \Omega^2 \times \{0\}, \tag{5.2.2h}$$

$$\widehat{\boldsymbol{B}}(\boldsymbol{U},\widehat{\boldsymbol{U}},\boldsymbol{\mathcal{U}}^\infty) = \boldsymbol{0} \qquad \text{on } \left(\partial\Omega^1_e \cap \partial\Omega\right) \times (0,T], \tag{5.2.2i}$$

$$\boldsymbol{B}(\boldsymbol{V},\boldsymbol{\mathcal{U}}^\infty) = \boldsymbol{0} \qquad \text{on } \left(\partial\Omega^2 \cap \partial\Omega\right) \times (0,T], \tag{5.2.2j}$$

for $e = 1, \dots, \mathtt{n_{el}}$, where $[\![\![\circledcirc]\!]\!] = \widehat{\circledcirc} + \circledcirc|_{\Omega^2}$.

Remark 7 (HDG-FV interface devoid of hybrid variables)**.** *The strong form of Equation* (5.2.2) *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* (5.2.1e) *by Equation* (5.2.2e)*. 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* (5.2.1e)*). 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. Also, note that for the preferred use of LLF stabilisation matrix given in Equation* (3.2.4)*, the presence of the hybrid variables on each face in Γ^1 is necessary. 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.*

5.3 HDG-FV weak formulation

The current section begins with a recapitulation of the weak forms of the FV method presented in Section 4.2 and the HDG local and global problems described in Section 3.2. For each element in the HDG domain, Ω^1_e, $e = 1, \dots, \mathtt{n_{el}}$, the weak formulation of the HDG local problem presented in Equations (5.2.2a)-(5.2.2b), along with the shock capturing term in Equation (3.3.5) is: given $\widehat{\boldsymbol{U}}$ on $(\Gamma^1 \cup \partial\Omega^1)$, find $\boldsymbol{U} \in [\mathcal{W}_t^{h,k_e}(\Omega^1_e)]^{\mathtt{m_{sd}}}$ that satisfies

$$\Big(\delta\boldsymbol{U}, \boldsymbol{U}_t\Big)_{\Omega^1_e} - \Big(\boldsymbol{\nabla}\delta\boldsymbol{U}, \boldsymbol{F}(\boldsymbol{U})\Big)_{\Omega^1_e} + \Big\langle\delta\boldsymbol{U}, \widehat{\boldsymbol{F}}(\boldsymbol{U},\widehat{\boldsymbol{U}})\cdot\boldsymbol{n}\Big\rangle_{\partial\Omega^1_e} = \Big(\delta\boldsymbol{U}, \boldsymbol{S}\Big)_{\Omega^1_e} - \Big(\boldsymbol{\nabla}\delta\boldsymbol{U}, \kappa\boldsymbol{\nabla}\boldsymbol{U}\Big)_{\Omega^1_e}, \tag{5.3.1}$$

for all $\delta\boldsymbol{U} \in [\mathcal{W}^{h,k_e}(\Omega^1_e)]^{\mathtt{m_{sd}}}$. Similarly, the weak formulation of the HDG global problem presented in Equations (5.2.2c) and (5.2.2i) is: find $\widehat{\boldsymbol{U}} \in [\widehat{\mathcal{W}}_t^{h,k_j}(\Gamma^1 \cup \partial\Omega^1)]^{\mathtt{m_{sd}}}$ such that,

$$\sum_{e=1}^{\mathtt{n_{el}}} \Big\{ \Big\langle \delta\widehat{\boldsymbol{U}}, \widehat{\boldsymbol{F}}(\boldsymbol{U},\widehat{\boldsymbol{U}})\cdot\boldsymbol{n} \Big\rangle_{\partial\Omega^1_e\setminus\partial\Omega} + \Big\langle \delta\widehat{\boldsymbol{U}}, \widehat{\boldsymbol{B}}(\boldsymbol{U},\widehat{\boldsymbol{U}},\boldsymbol{\mathcal{U}}^\infty) \Big\rangle_{\partial\Omega^1_e\cap\partial\Omega} \Big\} = 0, \tag{5.3.2}$$

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

The weak formulation of the FV problem in Equations (5.2.2d) and (5.2.2j) is: find $\boldsymbol{V} \in [\mathcal{W}_t^{h,0}(\Omega_i^2)]^{\mathtt{msd}}$ that satisfies

$$\Big(\delta\boldsymbol{V}, \boldsymbol{V}_t\Big)_{\Omega_i^2} + \Big\langle\delta\boldsymbol{V}, \boldsymbol{F}(\boldsymbol{V})\cdot\boldsymbol{n}\Big\rangle_{\partial\Omega_i^2\setminus\partial\Omega} + \Big\langle\delta\boldsymbol{V}, \boldsymbol{B}(\boldsymbol{V}, \boldsymbol{\mathcal{U}}^\infty)\Big\rangle_{\partial\Omega_i^2\cap\partial\Omega} = \Big(\delta\boldsymbol{V}, \boldsymbol{S}\Big)_{\Omega_i^2}, \quad (5.3.3)$$

for all $\delta\boldsymbol{V} \in [\mathcal{W}^{h,0}(\Omega_i^2)]^{\mathtt{msd}}$ and for $i = 1, \dots, \mathtt{n_{cv}}$.

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 (5.3.2) and the continuity of the fluxes is imposed in the FV weak formulation given by Equation (5.3.3). The weak form of the coupled problem is: find $(\boldsymbol{U}, \widehat{\boldsymbol{U}}, \boldsymbol{V}) \in [\mathcal{W}_t^{h,k_e}(\Omega_e^1)]^{\mathtt{msd}} \times [\widehat{\mathcal{W}}_t^{h,k_j}(\Gamma^1 \cup \partial\Omega^1)]^{\mathtt{msd}} \times [\mathcal{W}^{h,0}(\Omega_i^2)]^{\mathtt{msd}}$ such that

$$\begin{aligned}
&\Big(\delta\boldsymbol{U}, \boldsymbol{U}_t\Big)_{\Omega_e^1} - \Big(\boldsymbol{\nabla}\delta\boldsymbol{U}, \boldsymbol{F}(\boldsymbol{U})\Big)_{\Omega_e^1} + \Big\langle\delta\boldsymbol{U}, \boldsymbol{F}(\widehat{\boldsymbol{U}})\cdot\boldsymbol{n}\Big\rangle_{\partial\Omega_e^1} \\
&\quad + \Big\langle\delta\boldsymbol{U}, \boldsymbol{\tau}\cdot\boldsymbol{U}\Big\rangle_{\partial\Omega_e^1} - \Big\langle\delta\boldsymbol{U}, \boldsymbol{\tau}\cdot\widehat{\boldsymbol{U}}\Big\rangle_{\partial\Omega_e^1} = \Big(\delta\boldsymbol{U}, \boldsymbol{S}\Big)_{\Omega_e^1} - \Big(\boldsymbol{\nabla}\delta\boldsymbol{U}, \kappa\boldsymbol{\nabla}\boldsymbol{U}\Big)_{\Omega_e^1}, && (5.3.4a)\\
&\sum_{e=1}^{\mathtt{n_{el}}}\Big\{\Big\langle\delta\widehat{\boldsymbol{U}}, \boldsymbol{F}(\widehat{\boldsymbol{U}})\cdot\boldsymbol{n}\Big\rangle_{\partial\Omega_e^1\setminus\partial\Omega} + \Big\langle\delta\widehat{\boldsymbol{U}}, \boldsymbol{\tau}\cdot\boldsymbol{U}\Big\rangle_{\partial\Omega_e^1\setminus\partial\Omega} - \Big\langle\delta\widehat{\boldsymbol{U}}, \boldsymbol{\tau}\cdot\widehat{\boldsymbol{U}}\Big\rangle_{\partial\Omega_e^1\setminus\partial\Omega^1} \\
&\quad - \Big\langle\delta\widehat{\boldsymbol{U}}, \boldsymbol{\tau}\cdot\boldsymbol{V}\Big\rangle_{\partial\Omega_e^1\cap\Gamma^I} + \Big\langle\delta\widehat{\boldsymbol{U}}, \widehat{\boldsymbol{B}}(\boldsymbol{U}, \widehat{\boldsymbol{U}}, \boldsymbol{\mathcal{U}}^\infty)\Big\rangle_{\partial\Omega_e^1\cap\partial\Omega}\Big\} = 0, && (5.3.4b)\\
&\Big(\delta\boldsymbol{V}, \boldsymbol{V}_t\Big)_{\Omega_i^2} + \Big\langle\delta\boldsymbol{V}, \boldsymbol{F}(\boldsymbol{V})\cdot\boldsymbol{n}\Big\rangle_{\partial\Omega_i^2\setminus\Gamma^I} + \Big\langle\delta\boldsymbol{V}, \boldsymbol{F}(\widehat{\boldsymbol{U}})\cdot\boldsymbol{n}\Big\rangle_{\partial\Omega_i^2\cap\Gamma^I} \\
&\quad - \Big\langle\delta\boldsymbol{V}, \boldsymbol{\tau}\cdot\boldsymbol{U}\Big\rangle_{\partial\Omega_i^2\cap\Gamma^I} + \Big\langle\delta\boldsymbol{V}, \boldsymbol{\tau}\cdot\widehat{\boldsymbol{U}}\Big\rangle_{\partial\Omega_i^2\cap\Gamma^I} \\
&\quad + \Big\langle\delta\boldsymbol{V}, \boldsymbol{B}(\boldsymbol{V}, \boldsymbol{\mathcal{U}}^\infty)\Big\rangle_{\partial\Omega_i^2\cap\partial\Omega} = \Big(\delta\boldsymbol{V}, \boldsymbol{S}\Big)_{\Omega_i^2}, && (5.3.4c)
\end{aligned}$$

for all $\delta\boldsymbol{U} \in [\mathcal{W}^{h,k_e}(\Omega_e^1)]^{\mathtt{msd}}$, $\delta\widehat{\boldsymbol{U}} \in [\widehat{\mathcal{W}}^{h,k_j}(\Gamma^1 \cup \partial\Omega^1)]^{\mathtt{msd}}$, $\delta\boldsymbol{V} \in [\mathcal{W}^{h,0}(\Omega_i^2)]^{\mathtt{msd}}$ and for $e = 1, \dots, \mathtt{n_{el}}$ and $i = 1, \dots, \mathtt{n_{cv}}$.

The semi-discrete weak formulation for the proposed HDG-FV, with BDF time integration given in Equation (3.4.1), can be written as: find $(\boldsymbol{U}, \widehat{\boldsymbol{U}}, \boldsymbol{V}) \in [\mathcal{W}_t^{h,k_e}(\Omega_e^1)]^{\mathtt{msd}} \times$

$[\widehat{\mathcal{W}}_t^{h,k_j}(\Gamma^1 \cup \partial\Omega^1)]^{\mathtt{msd}} \times [\mathcal{W}_t^{h,k_e}[0](\Omega_i^2)]^{\mathtt{msd}}$ such that

$$
\begin{aligned}
&\sum_{s=0}^{\mathtt{n_{tg}}} 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} \\
&\quad + \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} \\
&\quad + \Big(\boldsymbol{\nabla}\delta\boldsymbol{U}, \kappa^n \boldsymbol{\nabla}\boldsymbol{U}^{n+1}\Big)_{\Omega_e^1} = 0, \qquad\qquad (5.3.5a) \\
&\sum_{e=1}^{\mathtt{n_{el}}} \Big\{ \Big\langle \delta\widehat{\boldsymbol{U}}, \boldsymbol{F}(\widehat{\boldsymbol{U}}^{n+1}) \cdot \boldsymbol{n} \Big\rangle_{\partial\Omega_e^1 \setminus \partial\Omega} + \Big\langle \delta\widehat{\boldsymbol{U}}, \boldsymbol{\tau}^{n+1} \cdot \boldsymbol{U}^{n+1} \Big\rangle_{\partial\Omega_e^1 \setminus \partial\Omega} \\
&\quad - \Big\langle \delta\widehat{\boldsymbol{U}}, \boldsymbol{\tau}^{n+1} \cdot \widehat{\boldsymbol{U}}^{n+1} \Big\rangle_{\partial\Omega_e^1 \setminus \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} \\
&\quad + \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} \Big\} = 0, \qquad\qquad (5.3.5b) \\
&\sum_{s=0}^{\mathtt{n_{tg}}} 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 \setminus \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} \\
&\quad - \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} \\
&\quad + \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\qquad (5.3.5c)
\end{aligned}
$$

for all $\delta\boldsymbol{U} \in [\mathcal{W}_t^{h,k_e}(\Omega_e^1)]^{\mathtt{msd}}$, $\delta\widehat{\boldsymbol{U}} \in [\widehat{\mathcal{W}}_t^{h,k_j}(\Gamma^1 \cup \partial\Omega^1)]^{\mathtt{msd}}$, $\delta\boldsymbol{V} \in [\mathcal{W}_t^{h,k_e}[0](\Omega_i^2)]^{\mathtt{msd}}$ and for $e = 1, \dots, \mathtt{n_{el}}$ and $i = 1, \dots, \mathtt{n_{cv}}$.

5.4 Discrete system

The approximations for the solution in the HDG and FV subdomains, given by Equations (3.5.1) and (4.2.3) respectively, and the approximation of the HDG hybrid variable given by Equation (3.5.2) are introduced in the semi-discrete system of Equation (5.3.5). Selecting the spaces of weighting functions as the space spanned by the shape functions for HDG, leads to the non-linear system of equations

$$
\boldsymbol{\mathcal{R}}(\boldsymbol{U}_h^{n+1}, \dots, \boldsymbol{U}_h^{n+1-\mathtt{n_{tg}}}, \widehat{\boldsymbol{U}}^{n+1}, \boldsymbol{V}_h^{n+1}, \dots, \boldsymbol{V}_h^{n+1-\mathtt{n_{tg}}}) = \boldsymbol{0}, \tag{5.4.1}
$$

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

$$
\boldsymbol{\mathcal{R}}^{e,i} := \left\{ \begin{matrix} \boldsymbol{\mathcal{R}}^{e}(\boldsymbol{U}_h^{n+1}, \dots, \boldsymbol{U}_h^{n+1-\mathtt{n_{tg}}}, \widehat{\boldsymbol{U}}_h^{n+1}) \\ \widehat{\boldsymbol{\mathcal{R}}}^{e}(\boldsymbol{U}_h^{n+1}, \dots, \boldsymbol{U}_h^{n+1-\mathtt{n_{tg}}}, \widehat{\boldsymbol{U}}_h^{n+1}, \boldsymbol{V}_h^{n+1}, \dots, \boldsymbol{V}_h^{n+1-\mathtt{n_{tg}}}) \\ \boldsymbol{\mathcal{R}}^{i}(\boldsymbol{U}_h^{n+1}, \dots, \boldsymbol{U}_h^{n+1-\mathtt{n_{tg}}}, \widehat{\boldsymbol{U}}_h^{n+1}, \boldsymbol{V}_h^{n+1}, \dots, \boldsymbol{V}_h^{n+1-\mathtt{n_{tg}}}) \end{matrix} \right\} \tag{5.4.2}
$$

with

$$\begin{aligned}
\boldsymbol{\mathcal{R}}_I^e :=& \sum_{s=0}^{\mathtt{n_{tg}}} a_s \Big(N_I, \boldsymbol{U}_h^{n+1-s} \Big)_{\Omega_e^1} - \Big(\boldsymbol{F}(\boldsymbol{U}_h^{n+1}), \boldsymbol{\nabla}^T N_I \Big)_{\Omega_e^1} + \Big\langle N_I, \boldsymbol{F}(\widehat{\boldsymbol{U}}_h^{n+1}) \cdot \boldsymbol{n} \Big\rangle_{\partial\Omega_e^1} \\
&+ \Big\langle N_I, \boldsymbol{\tau}^{n+1} \cdot \boldsymbol{U}_h^{n+1} \Big\rangle_{\partial\Omega_e^1} - \Big\langle N_I, \boldsymbol{\tau}^{n+1} \cdot \widehat{\boldsymbol{U}}_h^{n+1} \Big\rangle_{\partial\Omega_e^1} - \Big(N_I, \boldsymbol{S}^{n+1} \Big)_{\Omega_e^1} \\
&+ \Big(\kappa^n \boldsymbol{\nabla} \boldsymbol{U}_h^{n+1}, \boldsymbol{\nabla}^T N_I \Big)_{\Omega_e^1} = 0, && (5.4.3a) \\
\widehat{\boldsymbol{\mathcal{R}}}_I^e :=& \sum_{e=1}^{\mathtt{n_{el}}} \Big\{ \Big\langle \widehat{N}_I, \boldsymbol{F}(\widehat{\boldsymbol{U}}_h^{n+1}) \cdot \boldsymbol{n} \Big\rangle_{\partial\Omega_e^1 \setminus \partial\Omega} + \Big\langle \widehat{N}_I, \boldsymbol{\tau}^{n+1} \cdot \boldsymbol{U}_h^{n+1} \Big\rangle_{\partial\Omega_e^1 \setminus \partial\Omega} \\
&- \Big\langle \widehat{N}_I, \boldsymbol{\tau}^{n+1} \cdot \widehat{\boldsymbol{U}}_h^{n+1} \Big\rangle_{\partial\Omega_e^1 \setminus \partial\Omega^1} - \Big\langle \widehat{N}_I, \boldsymbol{\tau}^{n+1} \cdot \boldsymbol{V}_h^{n+1} \Big\rangle_{\partial\Omega_e^1 \cap \Gamma^I} \\
&+ \Big\langle \widehat{N}_I, \widehat{\boldsymbol{B}}(\boldsymbol{U}_h^{n+1}, \widehat{\boldsymbol{U}}_h^{n+1}, \boldsymbol{\mathcal{U}}^\infty) \Big\rangle_{\partial\Omega_e^1 \cap \partial\Omega} \Big\} = 0, && (5.4.3b) \\
\boldsymbol{\mathcal{R}}^i :=& \sum_{s=0}^{\mathtt{n_{tg}}} a_s \Big(1, \boldsymbol{V}_h^{n+1-s} \Big)_{\Omega_i^2} + \Big\langle 1, \boldsymbol{F}(\boldsymbol{V}_h^{n+1}) \cdot \boldsymbol{n} \Big\rangle_{\partial\Omega_i^2 \setminus \Gamma^I} + \Big\langle 1, \boldsymbol{F}(\widehat{\boldsymbol{U}}_h^{n+1}) \cdot \boldsymbol{n} \Big\rangle_{\partial\Omega_i^2 \cap \Gamma^I} \\
&- \Big\langle 1, \boldsymbol{\tau}^{n+1} \cdot \boldsymbol{U}_h^{n+1} \Big\rangle_{\partial\Omega_i^2 \cap \Gamma^I} + \Big\langle 1, \boldsymbol{\tau}^{n+1} \cdot \widehat{\boldsymbol{U}}_h^{n+1} \Big\rangle_{\partial\Omega_i^2 \cap \Gamma^I} \\
&+ \Big\langle 1, \boldsymbol{B}(\boldsymbol{V}_h^{n+1}, \boldsymbol{\mathcal{U}}^\infty) \Big\rangle_{\partial\Omega_i^2 \cap \partial\Omega} - \Big(1, \boldsymbol{S}^{n+1} \Big)_{\Omega_i^2} = 0, && (5.4.3c)
\end{aligned}$$

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

5.5 Linearisation method

As previously discussed for HDG and FV methods, the Newton-Raphson method is applied to linearise the non-linear residual of Equation (5.4.1). The algebraic system to be solved at each iteration (q) of the Newton-Raphson is

$$\begin{bmatrix} \mathbf{T}_{uu} & \mathbf{T}_{u\hat{u}} & \mathbf{0} \\ \mathbf{T}_{\hat{u}u} & \mathbf{T}_{\hat{u}\hat{u}} & \mathbf{T}_{\hat{u}v} \\ \mathbf{T}_{vu} & \mathbf{T}_{v\hat{u}} & \mathbf{T}_{vv} \end{bmatrix}^{n+1,q} \begin{Bmatrix} \Delta\mathrm{U} \\ \Delta\widehat{\mathrm{U}} \\ \Delta\mathbf{V} \end{Bmatrix}^{n+1,q} = \begin{Bmatrix} \mathbf{f}_u \\ \mathbf{f}_{\hat{u}} \\ \mathbf{f}_v \end{Bmatrix}^{n+1,q}, \qquad (5.5.1)$$

where $\Delta\circledcirc^{n+1,q} = \circledcirc^{n+1,q+1} - \circledcirc^{n+1,q}$ denote the increment of the solution vector. 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 $\mathbf{f}_v$ resulting from the linearisation are expressed as,

$$(\mathbf{T}_{uu})_{IJ}^{e,n+1,q} := \frac{\partial \boldsymbol{\mathcal{R}}_I^e}{\partial \mathbf{U}_J^{n+1,q}}, \qquad (\mathbf{T}_{u\hat{u}})_{IJ}^{e,n+1,q} := \frac{\partial \boldsymbol{\mathcal{R}}_I^e}{\partial \widehat{\mathbf{U}}_J^{n+1,q}}, \tag{5.5.2a}$$

$$(\mathbf{T}_{\hat{u}u})_{IJ}^{e,n+1,q} := \frac{\partial \widehat{\boldsymbol{\mathcal{R}}}_I^e}{\partial \mathbf{U}_J^{n+1,q}}, \qquad (\mathbf{T}_{\hat{u}\hat{u}})_{IJ}^{e,n+1,q} := \frac{\partial \widehat{\boldsymbol{\mathcal{R}}}_I^e}{\partial \widehat{\mathbf{U}}_J^{n+1,q}}, \tag{5.5.2b}$$

$$(\mathbf{T}_{\hat{u}v})_{IJ}^{e,n+1,q} := \frac{\partial \widehat{\boldsymbol{\mathcal{R}}}_I^e}{\partial \mathbf{V}_J^{n+1,q}}, \qquad (\mathbf{T}_{vu})_{J}^{i,n+1,q} := \frac{\partial \boldsymbol{\mathcal{R}}^i}{\partial \mathbf{U}_J^{n+1,q}}, \tag{5.5.2c}$$

$$(\mathbf{T}_{v\hat{u}})_{J}^{i,n+1,q} := \frac{\partial \boldsymbol{\mathcal{R}}^i}{\partial \widehat{\mathbf{U}}_J^{n+1,q}}, \qquad (\mathbf{T}_{vv})_{J}^{i,n+1,q} := \frac{\partial \boldsymbol{\mathcal{R}}^i}{\partial \mathbf{V}_J^{n+1,q}}, \tag{5.5.2d}$$

and

$$(\mathbf{f}_u^e)_I^{n+1,q} := -\boldsymbol{\mathcal{R}}_I^e, \qquad (\mathbf{f}_{\hat{u}}^e)_I^{n+1,q} := -\widehat{\boldsymbol{\mathcal{R}}}_I^e, \qquad \left(\mathbf{f}_v^i\right)^{n+1,q} := -\boldsymbol{\mathcal{R}}^i. \tag{5.5.3a}$$

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 through static condensation, i.e.,

$$\begin{bmatrix} \widetilde{\mathbf{T}}_{\hat{u}\hat{u}} & \mathbf{T}_{\hat{u}v} \\ \widetilde{\mathbf{T}}_{v\hat{u}} & \mathbf{T}_{vv} \end{bmatrix}^{n+1,q} \begin{Bmatrix} \Delta\widehat{\mathbf{U}} \\ \Delta\mathbf{V} \end{Bmatrix}^{n+1,q} = \begin{Bmatrix} \tilde{\mathbf{f}}_{\hat{u}} \\ \tilde{\mathbf{f}}_v \end{Bmatrix}^{n+1,q}, \tag{5.5.4}$$

where,

$$\widetilde{\mathbf{T}}_{\hat{u}\hat{u}} : = \mathbf{T}_{\hat{u}\hat{u}} - \mathbf{T}_{\hat{u}u}\mathbf{T}_{uu}^{-1}\mathbf{T}_{u\hat{u}}, \qquad \widetilde{\mathbf{T}}_{v\hat{u}} : = \mathbf{T}_{v\hat{u}} - \mathbf{T}_{vu}\mathbf{T}_{uu}^{-1}\mathbf{T}_{u\hat{u}} \tag{5.5.5a}$$

$$\tilde{\mathbf{f}}_{\hat{u}} : = \mathbf{f}_{\hat{u}} - \mathbf{T}_{\hat{u}u}\mathbf{T}_{uu}^{-1}\mathbf{f}_u, \qquad \tilde{\mathbf{f}}_v : = \mathbf{f}_v - \mathbf{T}_{vu}\mathbf{T}_{uu}^{-1}\mathbf{f}_u. \tag{5.5.5b}$$

Solving the linear system, the solution in the HDG domain is recovered by solving a set of independent local problems in each element, namely,

$$\mathbf{T}_{uu}^{n+1,q}\Delta\mathbf{U}^{n+1,q} = \mathbf{f}_u^{n+1,q} - \mathbf{T}_{u\hat{u}}^{n+1,q}\Delta\widehat{\mathbf{U}}^{n+1,q}. \tag{5.5.6}$$

As compared to the HDG residuals in Equation (3.6.3) and the FV residual in Equation (4.5.1), the coupled residual in Equation (5.4.3) has additional terms related to the transmission conditions. The RHS vector for the global problem in Equation (3.8.3b) is modified as,

$$
\begin{aligned}
\left(\mathbf{f}^{e}_{\hat{u}}\right)^{n+1,q} = &- \sum_{\partial\Omega^1_e \setminus \partial\Omega} \sum_{\mathsf{g}=1}^{\mathtt{n}^{\mathsf{f}}_{\mathtt{ip}}} \widehat{\mathbf{N}}(\boldsymbol{\xi}^{\mathsf{f}}_{\mathsf{g}}) \left(\mathbf{F}(\widehat{\mathbf{U}}^{n+1,q}(\boldsymbol{\xi}^{\mathsf{f}}_{\mathsf{g}})) \cdot \boldsymbol{n}(\boldsymbol{\xi}^{\mathsf{f}}_{\mathsf{g}}) \right) w^{\mathsf{f}}_{\mathsf{g}} \\
&- \sum_{\partial\Omega^1_e \setminus \partial\Omega} \sum_{\mathsf{g}=1}^{\mathtt{n}^{\mathsf{f}}_{\mathtt{ip}}} \widehat{\mathbf{N}}(\boldsymbol{\xi}^{\mathsf{f}}_{\mathsf{g}}) \boldsymbol{\tau}^{n+1}(\boldsymbol{\xi}^{\mathsf{f}}_{\mathsf{g}}) \mathbf{U}^{n+1,q}(\boldsymbol{\xi}^{\mathsf{f}}_{\mathsf{g}}) w^{\mathsf{f}}_{\mathsf{g}} \\
&+ \sum_{\partial\Omega^1_e \setminus \partial\Omega^1} \sum_{\mathsf{g}=1}^{\mathtt{n}^{\mathsf{f}}_{\mathtt{ip}}} \widehat{\mathbf{N}}(\boldsymbol{\xi}^{\mathsf{f}}_{\mathsf{g}}) \boldsymbol{\tau}^{n+1}(\boldsymbol{\xi}^{\mathsf{f}}_{\mathsf{g}}) \widehat{\mathbf{U}}^{n+1,q}(\boldsymbol{\xi}^{\mathsf{f}}_{\mathsf{g}}) w^{\mathsf{f}}_{\mathsf{g}} \\
&- \sum_{\partial\Omega^1_e \cap \Gamma^I} \sum_{\mathsf{g}=1}^{\mathtt{n}^{\mathsf{f}}_{\mathtt{ip}}} \widehat{\mathbf{N}}(\boldsymbol{\xi}^{\mathsf{f}}_{\mathsf{g}}) \boldsymbol{\tau}^{n+1}(\boldsymbol{\xi}^{\mathsf{f}}_{\mathsf{g}}) \mathbf{V}^{n+1,q}(\boldsymbol{\xi}^{\mathsf{f}}_{\mathsf{g}}) w^{\mathsf{f}}_{\mathsf{g}} \\
&- \sum_{\partial\Omega^1_e \setminus \partial\Omega} \sum_{\mathsf{g}=1}^{\mathtt{n}^{\mathsf{f}}_{\mathtt{ip}}} \widehat{\mathbf{N}}(\boldsymbol{\xi}^{\mathsf{f}}_{\mathsf{g}}) \widehat{\boldsymbol{B}}(\boldsymbol{\xi}^{\mathsf{f}}_{\mathsf{g}}) w^{\mathsf{f}}_{\mathsf{g}}.
\end{aligned}
\tag{5.5.7a}
$$

Similarly, the RHS vector for the FV problem in Equation (4.6.2) is updated as,

$$
\begin{aligned}
\left(\mathbf{f}^{i}_{v}\right)^{n+1,q} := &-|\Omega^2_i| \sum_{s=0}^{\mathtt{n}_{\mathtt{tg}}} a_s \mathbf{V}_i^{n+1-s} + \sum_{j\in\Lambda_i} \left(C_l^{i,j} \left(\frac{\boldsymbol{F}_l\left(\mathbf{V}_i^{n+1}\right) + \boldsymbol{F}_l\left(\mathbf{V}_j^{n+1}\right)}{2} \right)\right) \\
&- \sum_{j\in\Lambda_i^{\partial}} \left(C_l^{i,j,\partial} \left(\frac{3\boldsymbol{F}_l\left(\mathbf{V}_i^{n+1}\right) + \boldsymbol{F}_l\left(\mathbf{V}_j^{n+1}\right)}{4} \right)\right) \\
&- \sum_{\Upsilon_i\in\Gamma^I} \sum_{\mathsf{g}=1}^{\mathtt{n}^{\Upsilon_i}_{\mathtt{ip}}} \left(\mathbf{F}\left(\widehat{\mathbf{U}}^{n+1,q}(\boldsymbol{\xi}^{\Upsilon_i}_{\mathsf{g}}) \right) \cdot \boldsymbol{n}(\boldsymbol{\xi}^{\Upsilon_i}_{\mathsf{g}}) \right) w^{\Upsilon_i}_{\mathsf{g}} \\
&+ \sum_{\Upsilon_i\in\Gamma^I} \sum_{\mathsf{g}=1}^{\mathtt{n}^{\Upsilon_i}_{\mathtt{ip}}} \boldsymbol{\tau}^{n+1} \mathbf{U}^{n+1,q}(\boldsymbol{\xi}^{\Upsilon_i}_{\mathsf{g}}) w^{\Upsilon_i}_{\mathsf{g}} - \sum_{\Upsilon_i\in\Gamma^I} \sum_{\mathsf{g}=1}^{\mathtt{n}^{\Upsilon_i}_{\mathtt{ip}}} \boldsymbol{\tau}^{n+1} \widehat{\mathbf{U}}^{n+1,q}(\boldsymbol{\xi}^{\Upsilon_i}_{\mathsf{g}}) w^{\Upsilon_i}_{\mathsf{g}} \\
&- \sum_{j\in\Lambda_i} \mathbf{M}^n_{i,j} \left(\mathbf{E}_j\left(\mathbf{V}^{n+1}\right) - \mathbf{E}_i\left(\mathbf{V}^{n+1}\right) \right) \\
&+ \sum_{j\in\Lambda_i} \eta_2 \alpha^n_{i,j} s^n_{i,j} \left(\mathbf{V}_j^{n+1} - \mathbf{V}_i^{n+1} \right) + |\Omega^2_i|\, \mathbf{S}_i^{n+1},
\end{aligned}
\tag{5.5.8a}
$$

where, $\boldsymbol{\xi}^{\Upsilon_i}_{\mathsf{g}}$ and $w^{\Upsilon_i}_{\mathsf{g}}$ are the $\mathtt{n}^{\Upsilon_i}_{\mathtt{ip}}$ integrations points and weights defined in the reference facet, which is a triangular element in 3D or a straight edge in 2D. The number of integration points for an interface facet ($\Upsilon_i \in \Gamma^I$) is based on the order of the adjacent HDG element to ensure accurate integration of the HDG variables.

Consequently, together with the tangent matrices for the HDG global problem in Equation (3.8.4d), the following tangent matrices are computed,

$$
\begin{aligned}
(\mathbf{T}_{\hat{u}u})^{e,n+1,q} &= \sum_{\partial\Omega_e^1\setminus\partial\Omega} \sum_{\mathtt{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\mathtt{f}}} \widehat{\mathbf{N}}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) \boldsymbol{\tau}^{n+1}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) \mathbf{N}^T(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) w_{\mathtt{g}}^{\mathtt{f}} \\
&+ \sum_{\partial\Omega_e^1\setminus\partial\Omega} \sum_{\mathtt{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\mathtt{f}}} \widehat{\mathbf{N}}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) \left(\frac{\partial \boldsymbol{\tau}^{n+1}}{\partial \mathbf{U}^{n+1}} \Big|_{\mathbf{U}^{n+1,q}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}})} \mathbf{U}^{n+1,q}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) \right) \mathbf{N}^T(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) w_{\mathtt{g}}^{\mathtt{f}} \\
&+ \sum_{\partial\Omega_e^1\cap\partial\Omega} \sum_{\mathtt{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\mathtt{f}}} \widehat{\mathbf{N}}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) \left(\frac{\partial \widehat{\boldsymbol{B}}}{\partial \mathbf{U}^{n+1}} \Big|_{\mathbf{U}^{n+1,q}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}})} \right) \mathbf{N}^T(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) w_{\mathtt{g}}^{\mathtt{f}}, \qquad (5.5.9a) \\
(\mathbf{T}_{\hat{u}v})^{e,n+1,q} &= \sum_{\partial\Omega_e^1\cap\Gamma^I} \sum_{\mathtt{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\mathtt{f}}} \widehat{\mathbf{N}}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) \boldsymbol{\tau}^{n+1}(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) \widetilde{\mathbf{N}}^T(\boldsymbol{\xi}_{\mathtt{g}}^{\mathtt{f}}) w_{\mathtt{g}}^{\mathtt{f}}, \qquad (5.5.9b)
\end{aligned}
$$

where, following the definitions in Equation (3.8.1) for HDG, $\widetilde{\mathbf{N}}$ is constructed for a FV face with $\mathtt{n}_{\mathtt{fn}}$ nodes such that,

$$
\widetilde{\mathbf{N}} = \left[\widetilde{N}_1 \mathbf{I}_{\mathtt{m}_{\mathtt{sd}}} \quad \widetilde{N}_2 \mathbf{I}_{\mathtt{m}_{\mathtt{sd}}} \quad \dots \quad \widetilde{N}_{\mathtt{n}_{\mathtt{fn}}} \mathbf{I}_{\mathtt{m}_{\mathtt{sd}}} \right]^T . \qquad (5.5.10)
$$

Similarly, in addition to the tangent matrices of the FV residual in Equation (4.6.3), the new tangent matrices are introduced as,

$$
\begin{aligned}
(\mathbf{T}_{vu})_J^{i,n+1,q} &= \sum_{\Upsilon_i\in\Gamma^I} \sum_{\mathtt{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\Upsilon_i}} \boldsymbol{\tau}^{n+1} \mathbf{N}^T(\boldsymbol{\xi}_{\mathtt{g}}^{\Upsilon_i}) w_{\mathtt{g}}^{\Upsilon_i} \\
&+ \sum_{\Upsilon_i\in\Gamma^I} \sum_{\mathtt{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\Upsilon_i}} \left(\frac{\partial \boldsymbol{\tau}^{n+1}}{\partial \mathbf{U}^{n+1}} \Big|_{\mathbf{U}^{n+1,q}(\boldsymbol{\xi}_{\mathtt{g}}^{\Upsilon_i})} \mathbf{U}^{n+1,q}(\boldsymbol{\xi}_{\mathtt{g}}^{\Upsilon_i}) \right) \mathbf{N}^T(\boldsymbol{\xi}_{\mathtt{g}}^{\Upsilon_i}) w_{\mathtt{g}}^{\Upsilon_i}, \qquad (5.5.11a) \\
(\mathbf{T}_{v\hat{u}})_J^{i,n+1,q} &= - \sum_{\Upsilon_i\in\Gamma^I} \sum_{\mathtt{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\Upsilon_i}} \left(\mathbf{A}\left(\widehat{\mathbf{U}}^{n+1,q}(\boldsymbol{\xi}_{\mathtt{g}}^{\Upsilon_i}) \right) \cdot \boldsymbol{n}(\boldsymbol{\xi}_{\mathtt{g}}^{\Upsilon_i}) \right) \widehat{\mathbf{N}}^T(\boldsymbol{\xi}_{\mathtt{g}}^{\Upsilon_i}) w_{\mathtt{g}}^{\Upsilon_i} \\
&- \sum_{\Upsilon_i\in\Gamma^I} \sum_{\mathtt{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\Upsilon_i}} \boldsymbol{\tau}^{n+1} \widehat{\mathbf{N}}^T(\boldsymbol{\xi}_{\mathtt{g}}^{\Upsilon_i}) w_{\mathtt{g}}^{\Upsilon_i} \\
&- \sum_{\Upsilon_i\in\Gamma^I} \sum_{\mathtt{g}=1}^{\mathtt{n}_{\mathtt{ip}}^{\Upsilon_i}} \left(\frac{\partial \boldsymbol{\tau}^{n+1}}{\partial \widehat{\mathbf{U}}^{n+1}} \Big|_{\widehat{\mathbf{U}}^{n+1,q}(\boldsymbol{\xi}_{\mathtt{g}}^{\Upsilon_i})} \widehat{\mathbf{U}}^{n+1,q}(\boldsymbol{\xi}_{\mathtt{g}}^{\Upsilon_i}) \right) \widehat{\mathbf{N}}^T(\boldsymbol{\xi}_{\mathtt{g}}^{\Upsilon_i}) w_{\mathtt{g}}^{\Upsilon_i}. \qquad (5.5.11b)
\end{aligned}
$$

Remark 8 (Algebraic system for HDG-FV interface devoid of hybrid variables)**.** *As mentioned in Remark 7, 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*

$$\begin{bmatrix} \mathbf{T}_{uu} & \mathbf{T}_{u\hat{u}} & \mathbf{T}_{uv} \\ \mathbf{T}_{\hat{u}u} & \mathbf{T}_{\hat{u}\hat{u}} & \mathbf{0} \\ \mathbf{T}_{vu} & \mathbf{T}_{v\hat{u}} & \mathbf{T}_{vv} \end{bmatrix}. \tag{5.5.12}$$

This alternative formulation leads to a reduced system

$$\begin{bmatrix} \widetilde{\mathbf{T}}_{\hat{u}\hat{u}} & \widetilde{\mathbf{T}}_{\hat{u}v} \\ \widetilde{\mathbf{T}}_{v\hat{u}} & \widetilde{\mathbf{T}}_{vv} \end{bmatrix}^{n+1,q} \begin{Bmatrix} \Delta\widehat{\mathbf{U}} \\ \Delta\mathbf{V} \end{Bmatrix}^{n+1,q} = \begin{Bmatrix} \tilde{\mathbf{f}}_{\hat{u}} \\ \tilde{\mathbf{f}}_{v} \end{Bmatrix}^{n+1,q}, \tag{5.5.13}$$

where

$$\widetilde{\mathbf{T}}_{\hat{u}v} := -\mathbf{T}_{\hat{u}u}\mathbf{T}_{uu}^{-1}\mathbf{T}_{uv}, \qquad \widetilde{\mathbf{T}}_{vv} := \mathbf{T}_{vv} - \mathbf{T}_{vu}\mathbf{T}_{uu}^{-1}\mathbf{T}_{uv} \tag{5.5.14}$$

When compared to the system of Equation (5.5.4)*, the system of the alternative formulation given by Equation* (5.5.13) *is slightly smaller but the hybridisation process requires the extra operations detailed in Equation* (5.5.14) *for each time step and each non-linear iteration. Reiterating, the LLF stabilisation cannot be realised in the absence of skeleton nodes on the HDG-FV interface.*

5.6 Implementation details

Off-diagonal tangent matrices corresponding to the interface terms need special considerations. It is of value to preserve the element-based assembly of matrices in HDG and the edge-based assembly in FV, when possible, for the interface contributions.

In the HDG domain, the interface terms in the residual and the tangent matrices appear in Equation (5.5.7a) and Equation (5.5.9b) respectively. Therefore, in the assembly over each HDG face lying on the interface, the connectivities of corresponding FV nodes on the interface are required. It is worth noting that the number of integration points is decided by the order of the HDG element. This ensures accurate numerical integration. Illustrated in Figure 5.2, the HDG face on the interface being integrated is highlighted. The FV nodes in blue denote the face connectivity required for the numerical integration.

On the other hand, the interface terms appear in the FV residuals and tangent matrices given in Equation (5.5.8) and Equation (5.5.11) respectively. They are assembled in an edge-based manner. The integration is performed on a per-facet basis for every interface edge. Here, the contributions are assembled to the respective edge nodes.

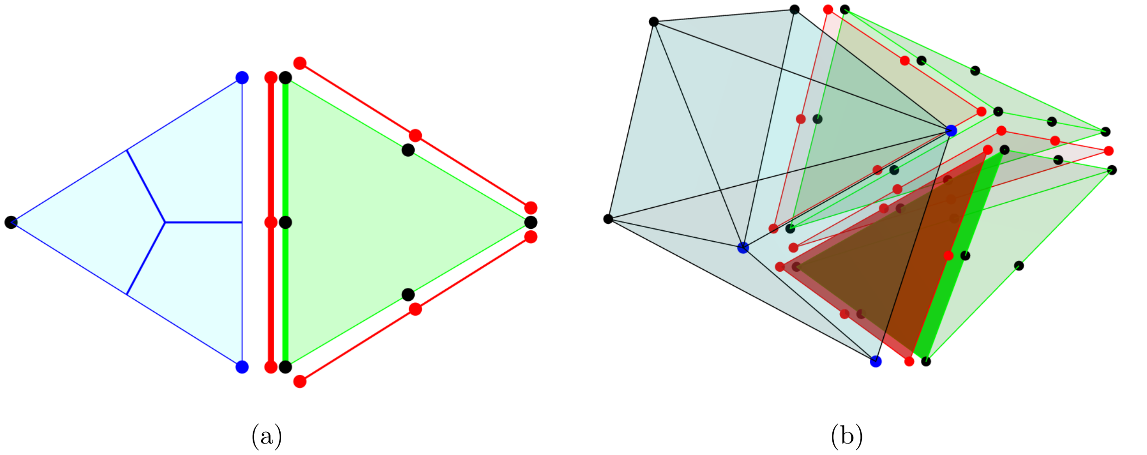


Figure. 5.2 A schematic for the interface integration of the HDG residual and its tangent matrices in (a) 2D and (b) 3D respectively. HDG face being integrated is highlighted. FV nodes in blue denote the corresponding face connectivity required for integration.

This is illustrated in Figure 5.3 for an interface edge connected by nodes (in magenta and blue). Here, the integration over highlighted triangular interface facets in 3D or linear interface facets in 2D are performed. Again, since high-order HDG variables $\boldsymbol{U}$ and $\widehat{\boldsymbol{U}}$ are involved in the interface terms, to ensure accuracy of the integration, the number of integration points is decided by the order of the HDG element. However, note that, the integration terms involve the HDG variables related to the skeleton and the element associated with the facet. The FV unknowns are not required.

Remark 9 (Consistent surface integration for interface FV control volumes)**.** *Balancing of the surface fluxes in control volumes is achieved with consistently defined edge weights as described in the Remark 4. However, the flux integration in FV control volumes with interface facets is now performed with HDG variables (as a result of the flux transmission condition) in Equation* (5.5.8) *and Equation* (5.5.11)*. As noted before, this integration is performed up to high-order accuracy on a per-facet basis. Recalling Remark 6, the assumption of flux taken to be constant over facets associated with an edge and interpolated using only edge values is no longer consistent for the interface control volumes. To regain the consistency, the trapezoidal rule for flux interpolation needs to be performed on a per-facet basis as well, see illustration in Figure 5.4. This implies that the surface weights involve the unknowns at not just the edge nodes, but the entire FV element in which the facet lies. It is worth noting that, only a very small number of control volumes associated with the interface deviates from the edge-based*

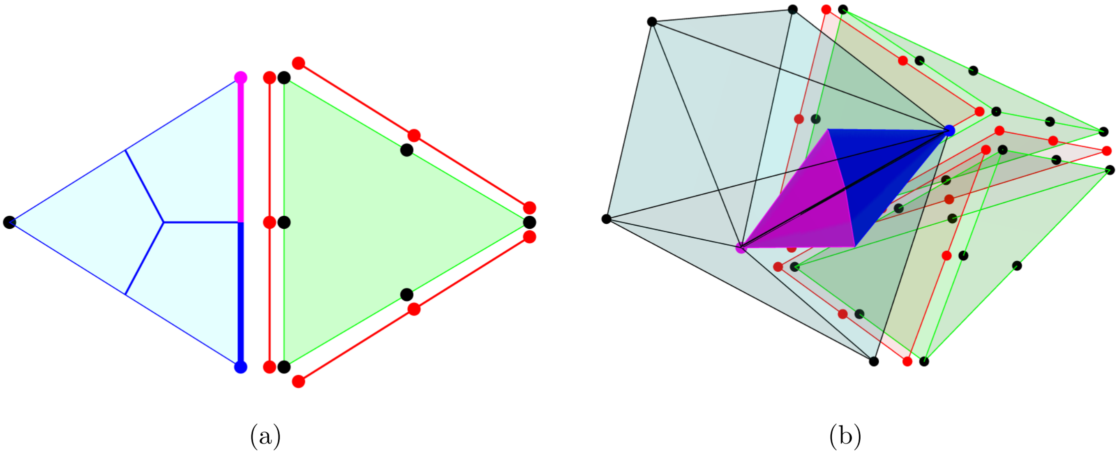


Figure. 5.3 A schematic for the interface integration of the FV residual and its tangent matrices in (a) 2D and (b) 3D respectively. Interface facets associated with a typical interface edge (connecting the blue and magenta nodes), where integration is performed on a per-facet basis are highlighted.

formulation and thus require modification. The rest of the FV domain remains adhered to the edge-based formulation. The correction implies that the flux coefficients in Equation (4.2.7) *are modified to compute on a per-facet basis as,*

$$C_l^{i,j}|_{\Upsilon_i} = |\Upsilon_i|\, n_l^{\Upsilon_i}, \qquad \forall \Upsilon_i \in \Upsilon_{i,j}, \tag{5.6.1a}$$

$$C_l^{i,j,\partial}|_{\Upsilon_i} = |\Upsilon_i|\, n_l^{\Upsilon_i}, \qquad \forall \Upsilon_i \in \Upsilon_{i,j}^{\partial}. \tag{5.6.1b}$$

The scheme remains consistent, i.e., for constant flux, the surface integral is provided

$$\sum_{j\in\Lambda_i} \sum_{\Upsilon_i\in\Upsilon_{i,j}} C_l^{i,j}|_{\Upsilon_i} + \sum_{j\in\Lambda_i^{\partial}} \sum_{\Upsilon_i\in\Upsilon_{i,j}^{\partial}} C_l^{i,j,\partial}|_{\Upsilon_i} = 0. \tag{5.6.2}$$

It is to be noted that, in the absence of the consistency correction, spurious solution jumps arise at the interface. A demonstration of the behaviour of the correction in reproducing the passage of shocks and discontinuities across the HDG-FV interface in provided in the upcoming Section 5.7.1.

Remark 10 (Considerations for artificial dissipation terms close to the HDG-FV interface)**.** *The artificial dissipation term in Equation* (4.3.1) *for a node involves not just its adjacent nodes, but the neighbours of its adjacent nodes as well. In one dimensional domain, this results in a 5-point stencil around the node. For the FV nodes close to and lying on the interface, the stencil extends into the HDG domain.*

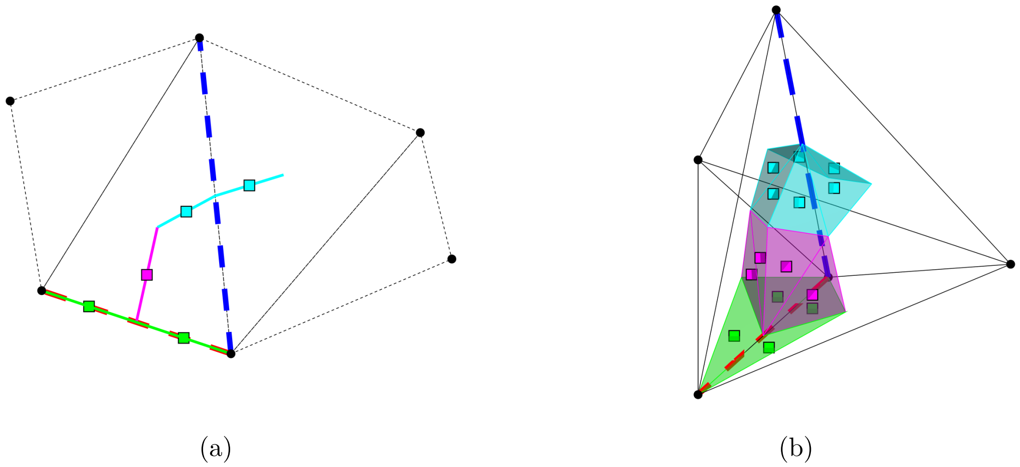


Figure. 5.4 An illustration of consistent flux interpolation for interface elements (located at □) for interior facets (in cyan and magenta) and boundary facets (in green) for an interior edge (in dotted blue) and a boundary edge (in dotted red) for (a) 2D and (b) 3D FV domains respectively.

However, multiple nodes define the solution at any given element corner. To overcome this difficulty, a weighted average based on the element volume is considered in order to obtain a singular value for the purposes of dissipation calculation. Note that this increases the bandwidth in the linear system for such FV nodes.

The convergence behaviour of the Newton-Raphson method is monitored with the $\mathcal{L}_2(\Omega)$ of the residual and solution errors in both HDG and FV domains, which are defined as below,

$$||\boldsymbol{\epsilon}_{R^{e,i}}||_{\mathcal{L}_2(\Omega)} := \sqrt{\sum_{e=1}^{\mathtt{n_{el}}} \int_{\Omega_e^1} \boldsymbol{\mathcal{R}}^e \cdot \boldsymbol{\mathcal{R}}^e \, \partial\Omega + \sum_{i=1}^{\mathtt{n_{cv}}} \int_{\Omega_i^2} \boldsymbol{\mathcal{R}}^i \cdot \boldsymbol{\mathcal{R}}^i \, \partial\Omega}, \tag{5.6.3a}$$

$$||\boldsymbol{\epsilon}_{\Delta U,\Delta V}||_{\mathcal{L}_2(\Omega)} := \frac{\sqrt{\int_{\Omega^1} \Delta \boldsymbol{U}^{n+1,q} \cdot \Delta \boldsymbol{U}^{n+1,q} \, \partial\Omega + \int_{\Omega^2} \Delta \boldsymbol{V}^{n+1,q} \cdot \Delta \boldsymbol{V}^{n+1,q} \, \partial\Omega}}{\sqrt{\int_{\Omega^1} \boldsymbol{U}^{n+1,q+1} \cdot \boldsymbol{U}^{n+1,q+1} \, \partial\Omega + \int_{\Omega^2} \boldsymbol{V}^{n+1,q+1} \cdot \boldsymbol{V}^{n+1,q+1} \, \partial\Omega}}. \tag{5.6.3b}$$

Additionally, $\mathcal{L}_2$ norms for the hybrid variable given in Equation (3.9.1c) and Equation (3.9.1d) are computed. As previously defined for HDG and FV methods, the linearisation iterations are considered to have converged when the value of the norms fall below a predefined tolerance, whose default value is set to 1.0×10^{-10}. Figure 5.5 shows the typical quadratic convergence of the $\mathcal{L}_2$ norms, thus demonstrating a consistent linearisation.

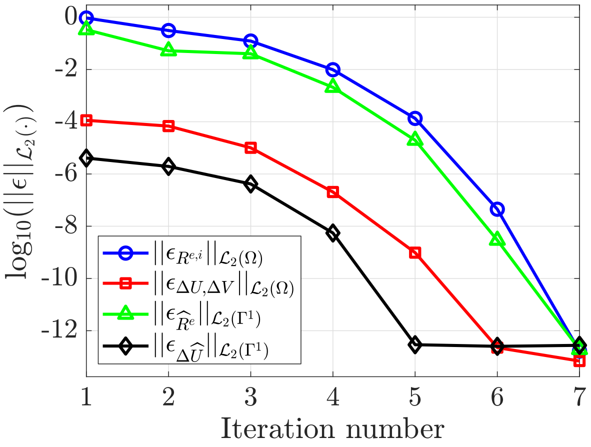


Figure. 5.5 Quadratic convergence of Newton-Raphson method demonstrating consistent linearisation of the non-linear system of equations for FV method.

Solution of the sparse linear system of equations follows the discussion previously described in Section 3.9. A pseudo-code laying out the solution procedure of the coupled HDG-FV method is provided in Algorithm 3.

5.7 Numerical tests

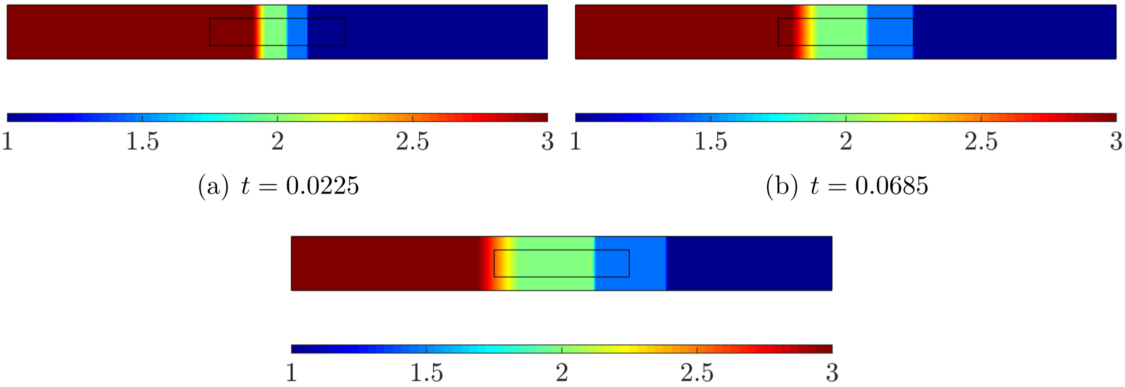
Several numerical tests are conducted on the coupled HDG-FV method. Sod's classical shock tube problem is solved to highlight the handling of shocks across and aligned with the interface. Later, tests for the convergence of spatial and temporal errors for the coupled scheme is presented. As previously demonstrated for HDG and FV methods separately, results from the Ringleb flow problem for 2D spatial tests and manufactured solutions for 3D spatial and 2D and 3D temporal tests are presented. For the manufactured solutions problem, the solution error $\boldsymbol{\epsilon}$ in the $\mathcal{L}_2(\Omega)$ is computed using the relation,

$$||\boldsymbol{\epsilon}||_{\mathcal{L}_2(\Omega)} := \frac{\sqrt{\int_{\Omega^1} \left(\boldsymbol{U} - \boldsymbol{\mathcal{U}}_a\right)^2 \, d\Omega + \int_{\Omega^2} \left(\boldsymbol{V} - \boldsymbol{\mathcal{U}}_a\right)^2 \, d\Omega}}{\sqrt{\int_\Omega \boldsymbol{\mathcal{U}}_a^2 \, d\Omega}}, \tag{5.7.1}$$

where $\boldsymbol{\mathcal{U}}_a$ is a suitably designed analytical solution.

```
Initial solution: $\boldsymbol{U}^{n=0}$, $\widehat{\boldsymbol{U}}^{n=0}$ and $\boldsymbol{V}^{n=0}$

Time stepping
for $t^{n+1}$, $n = 0$ to $end - 1$ do
    Data: $\boldsymbol{U}^{n}$,$\widehat{\boldsymbol{U}}^{n}$ and $\boldsymbol{V}^{n}$
    Initial guess: $\boldsymbol{U}^{n+1,q=0} = \boldsymbol{U}^{n}$, $\widehat{\boldsymbol{U}}^{n+1,q=0} = \widehat{\boldsymbol{U}}^{n}$ and $\boldsymbol{V}^{n+1,q=0} = \boldsymbol{V}^{n}$
    Compute residuals: $\boldsymbol{\mathcal{R}}^{n+1,0}$ from Equation (5.4.1)

    Newton-Raphson linearisation
    repeat
        Data: $\boldsymbol{U}^{n+1,q},\ldots,\boldsymbol{U}^{n+1-\mathtt{n_{tg}}},\widehat{\boldsymbol{U}}^{n+1,q},\boldsymbol{V}^{n+1,q},\ldots,\boldsymbol{V}^{n+1-\mathtt{n_{tg}}}$
        for $e = 1$ to $\boldsymbol{n}_{el}$ do
            Compute $(\mathbf{T}_{uu})_{IJ}^{e,n+1,q}$, $(\mathbf{T}_{u\hat{u}})_{IJ}^{e,n+1,q}$, $(\mathbf{T}_{\hat{u}u})_{IJ}^{e,n+1,q}$, $(\mathbf{T}_{\hat{u}\hat{u}})_{IJ}^{e,n+1,q}$, and
             $(\mathbf{T}_{\hat{u}v})_{IJ}^{e,n+1,q}$ in Equation (5.5.2) $(\mathbf{f}_{u}^{e})_{I}^{n+1,q}$ and $(\mathbf{f}_{\hat{u}}^{e})_{I}^{n+1,q}$ in
             Equation (5.5.3) and assemble to Equation (5.5.4)
        end
        for $e = 1$ to $\boldsymbol{n}_{me}$ do
            for $i = 1$ to $2$ do
                Compute $(\mathbf{T}_{vu})_{J}^{i,n+1,q}$, $(\mathbf{T}_{v\hat{u}})_{J}^{i,n+1,q}$, $(\mathbf{T}_{vv})_{J}^{i,n+1,q}$ in Equation (5.5.2)
                 $(\mathbf{f}_{v}^{i})^{n+1,q}$ in Equation (5.5.3) and assemble to Equation (5.5.4)
            end
        end

        Solve linear system, obtain $\{\Delta\widehat{\boldsymbol{U}}\}^{n+1,q}$, $\{\Delta\boldsymbol{V}\}^{n+1,q}$ in Equation (5.5.4)
        Update:
         $\widehat{\boldsymbol{U}}^{n+1,q+1} = \widehat{\boldsymbol{U}}^{n+1,q} + \{\Delta\widehat{\boldsymbol{U}}\}^{n+1,q}$, $\boldsymbol{V}^{n+1,q+1} = \boldsymbol{V}^{n+1,q} + \{\Delta\boldsymbol{V}\}^{n+1,q}$

        Local problem
        Obtain $\{\Delta\boldsymbol{U}\}^{n+1,q}$ in Equation (5.5.6)
        Update: $\boldsymbol{U}^{n+1,q+1} = \boldsymbol{U}^{n+1,q} + \{\Delta\boldsymbol{U}\}^{n+1,q}$

        Update residuals: $\boldsymbol{\mathcal{R}}^{n+1,q+1}$ from Equation (5.4.1)

        Update: $q = q + 1$
        Compute errors in Equation (5.6.3)
    until $||\boldsymbol{\epsilon}_{R^{e,i}}||_{\mathcal{L}_2(\Omega)}, ||\boldsymbol{\epsilon}_{\Delta U,\Delta V}||_{\mathcal{L}_2(\Omega)}, ||\boldsymbol{\epsilon}_{\widehat{R}^{e}}||_{\mathcal{L}_2(\Gamma^1)}$ and $||\boldsymbol{\epsilon}_{\Delta\widehat{U}}||_{\mathcal{L}_2(\Gamma^1)} \leq$ tolerance;

    Update solution: $\widehat{\boldsymbol{U}}^{n+1} = \widehat{\boldsymbol{U}}^{n+1,q+1}$, $\boldsymbol{U}^{n+1} = \boldsymbol{U}^{n+1,q+1}$ and $\boldsymbol{V}^{n+1} = \boldsymbol{V}^{n+1,q+1}$
end
```

Algorithm 3: Coupled HDG-FV solution procedure



(a) $t = 0.0225$

(b) $t = 0.0685$

(c) $t = 0.1145$

Figure. 5.6 Density at three different instants computed with the proposed HDG-FV scheme showing the ability to handle the shock at different stages.

5.7.1 Shock tube problem

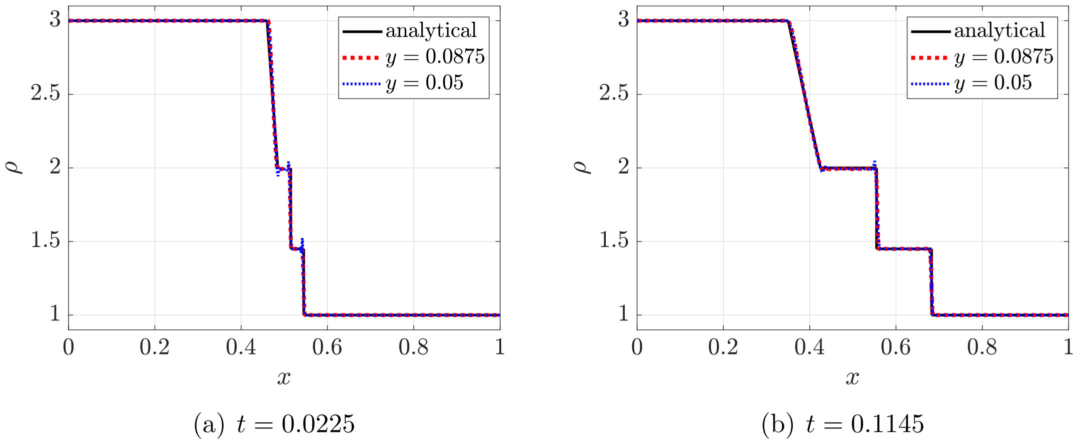
The classical shock tube problem is considered here to show the ability of the proposed coupling strategy to handle shocks that cross and that are aligned with the HDG-FV interface.

The computational domain is $\Omega = [0,1] \times [0,0.1]$ and the initial condition is defined as [90]

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

The FV subdomain is $\Omega^2 = [0.375, 0.625] \times [0.025, 0.075]$ and discretised into 15 584 triangular elements. The HDG subdomain is $\Omega^1 = \Omega \setminus \Omega^2$ and discretised into 80 640 triangular elements. 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 with $k = 1$, at three different instants, is displayed in Figure 5.6. 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 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. A comparison of density variation at



(a) $t = 0.0225$

(b) $t = 0.1145$

Figure. 5.7 Variation of density as a function of x coordinate.

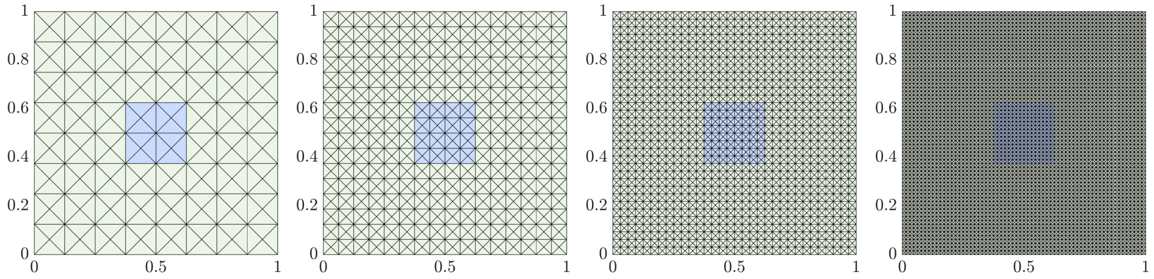
different sections are presented along with the analytical solution in Figure 5.7. The numerical and analytical solutions show good agreement.

It is worth mentioning that the focus here is on 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.

5.7.2 Spatial convergence tests

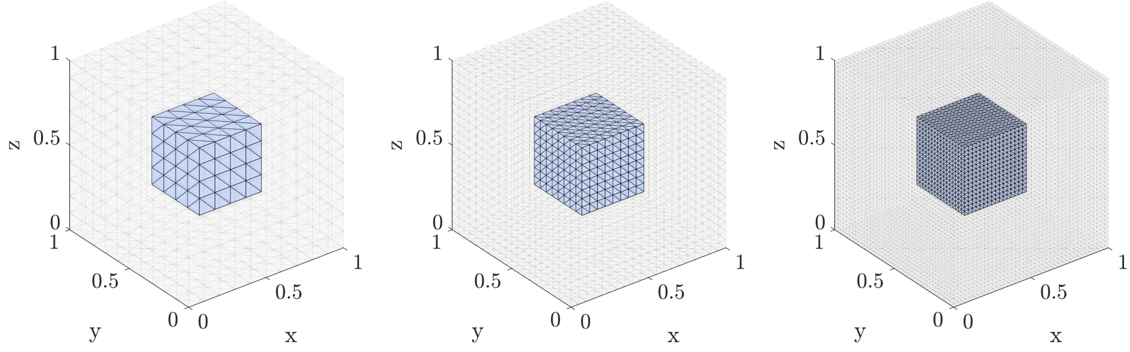
First, the convergence of the coupled HDG-FV method under mesh refinement is studied. In 2D, four uniform meshes of the domain $\Omega = [0,1]^2$ are considered for the Ringleb flow problem, with 256, 1024, 4096 and 16 384 triangular elements, respectively. The meshes are partitioned as $\Omega^2 = [0.375, 0.625]^2$ and $\Omega^1 = \Omega \setminus \Omega^2$ as shown in Figure 5.8. In 3D, three uniform meshes of the domain $\Omega = [0,1]^3$ with 6000, 48 000 and 384 000 tetrahedral elements, are considered for the manufactured solution problem given by Equation 3.11.2. The subdomains are created as $\Omega^2 = [0.3, 0.7]^3$ and $\Omega^1 = \Omega \setminus \Omega^2$ as illustrated in Figure 5.9.

Coupling of second-order finite volume and HDG with linear elements is considered. As previously for HDG and FV methods, a nodally exact initial solution is constructed and time marched until steady state solution. Figure 5.10 shows the nearly second-order convergence of the $\mathcal{L}_2(\Omega)$ of the errors versus mesh size h for 2D and 3D problems respectively. In addition, the results for both HDG with linear approximation and FV have been added to Figure 5.11 to enable a visual comparison of the gain in accuracy



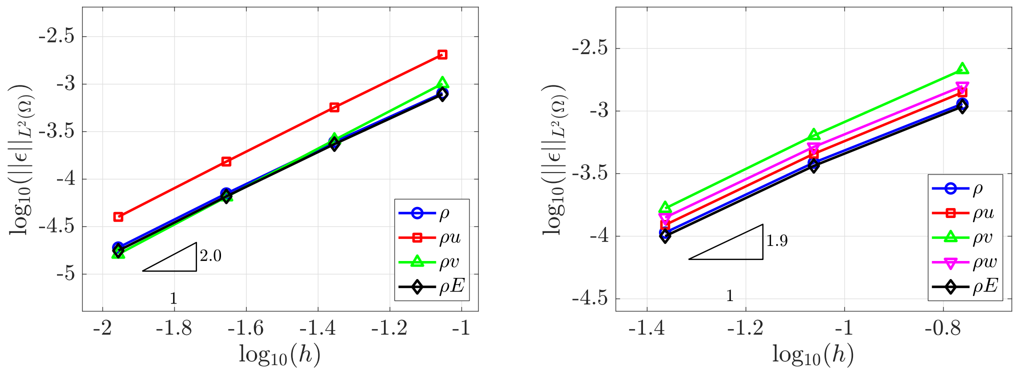
(a) Mesh 1 (b) Mesh 2 (c) Mesh 3 (d) Mesh 4

Figure. 5.8 Triangular meshes of the domain $\Omega = [0, 1]^2$ used to test the optimal convergence properties of the coupled HDG-FV method.



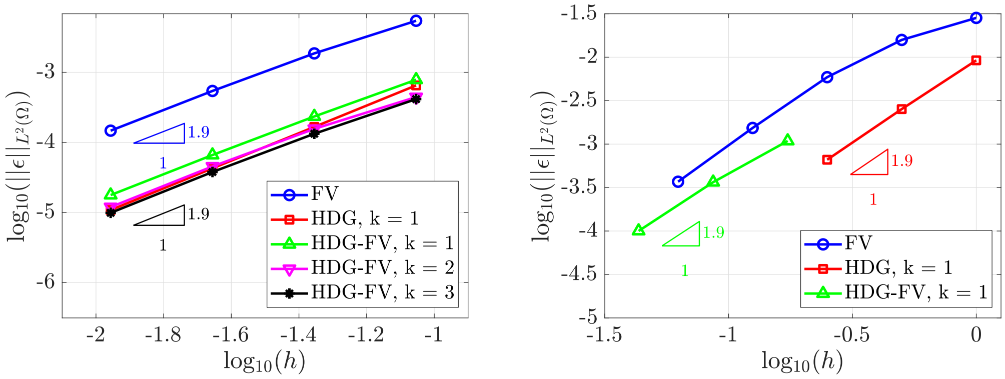
(a) Mesh 1 (b) Mesh 2 (c) Mesh 3

Figure. 5.9 Tetrahedral meshes of the domain $\Omega = [0, 1]^3$ used to test the optimal convergence properties of the coupled HDG-FV method.



(a) (b)

Figure. 5.10 The optimal rates of convergence for the coupled method as shown in the plots of the $\mathcal{L}_2(\Omega)$ of the error in all the conservative variables versus the mesh size h, for the (a) Ringleb flow in 2D and (b) manufactured solution in Equation (3.11.2).



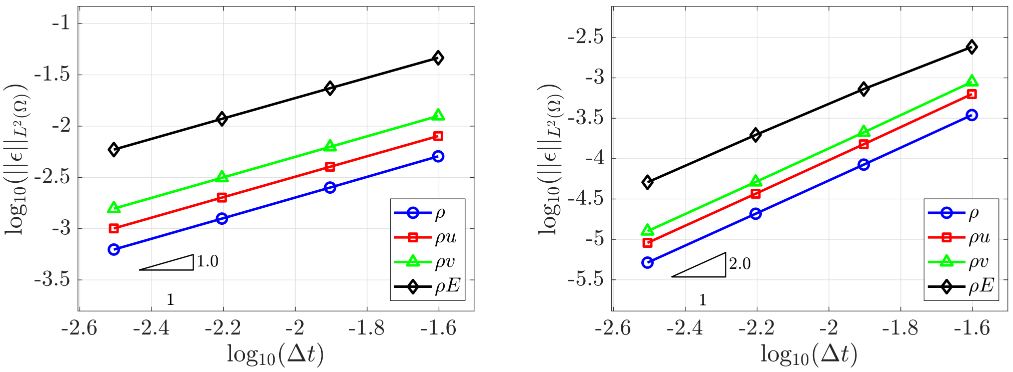
(a) (b)

Figure. 5.11 Convergence properties of FV, HDG and the coupled scheme for the (a) Ringleb flow problem in 2D and (b) manufactured solution problem in Equation (3.11.2) respectively.

induced by the HDG formulation with linear approximation when compared to the standard second-order FV method. In the Ringleb flow example, the accuracy of the combined scheme, as depicted in Figure 5.11(a), is almost identical to the accuracy of the standard HDG solver. Also, the solution accuracy improves when high-order approximations are employed in the HDG subdomain, but, the rate of convergence is dictated by the solution error in the second-order accurate FV subdomain. For the 3D manufactured solution problem, the solution accuracy of the coupled scheme, as shown in Figure 5.11(b), lies between those of its constituent methods.

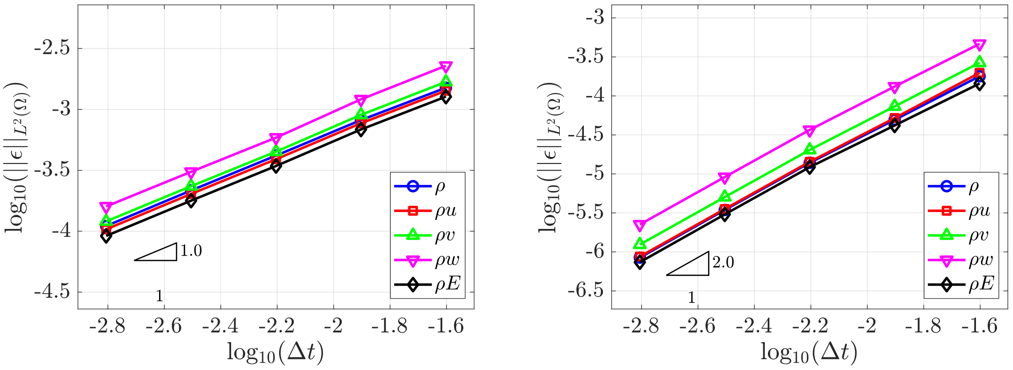
5.7.3 Temporal convergence tests

Finally, for the temporal convergence properties of the coupled method with BDF method of time integration, the same partition of domain Ω as described in the previous section is carried out. Also a finer mesh is employed to minimise the influence of the spatial error terms. The analytical solution are based on Equation (3.11.3) and Equation (3.11.4), for 2D and 3D cases respectively. Initial solution in the form of nodally exact analytical solution was time marched up-to the same physical time with decreasing sizes of the time step size Δt. Figure 5.12 for 2D and Figure 5.13 demonstrate the convergence in the solution error at an optimal rate given by the order of the BDF method in all conservative variables.



(a) (b)

Figure. 5.12 The optimal rates of convergence for the coupled method as shown in the plots of the $\mathcal{L}_2(\Omega)$ of the error in all the conservative variables versus the time step size Δt, for different orders of the BDF method, (a) BDF1 and (b) BDF2. The example demonstrated here is the 2D manufactured solution for temporal convergence given in Equation (3.11.3).



(a) (b)

Figure. 5.13 The optimal rates of convergence for the FV method as shown in the plots of the $\mathcal{L}_2(\Omega)$ of the error in all the conservative variables versus the time step size Δt, for different orders of the BDF method, (a) BDF1 and (b) BDF2. The example demonstrated here is the 3D manufactured solution for temporal convergence given in Equation (3.11.4).

Chapter 6

Numerical examples

"Nothing ever comes to one, that is worth having, except as a result of hard work."

Booker T. Washington

The present chapter demonstrates the application of the coupled HDG-FV method for transient problems. Examples of inviscid gust flow over various aerodynamic configurations in both 2D and 3D domains are considered. Attention is drawn towards comparing and contrasting the coupled scheme with the low-order FV method. Special emphasis is placed on highlighting the suitability of the proposed scheme for transient simulations.

In both 2D and 3D examples, the discussion begins with a succinct description of the problem setup. A mesh convergence study is carried out to select a suitable mesh refinement for a steady-state problem. This mesh is then used to perform transient gust simulations using the coupled method. The details for the partitioning into HDG and FV subdomains in the coupled scheme are provided. Finally, the results are compared with standard FV method applied to meshes specifically designed for the gust problem.

6.1 Gust flow in 2D: NACA aerofoil

6.1.1 Problem setup

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 6.1, showing the aerofoil of chord length $c = 1$ and the rectangular box of dimension $a \times b$

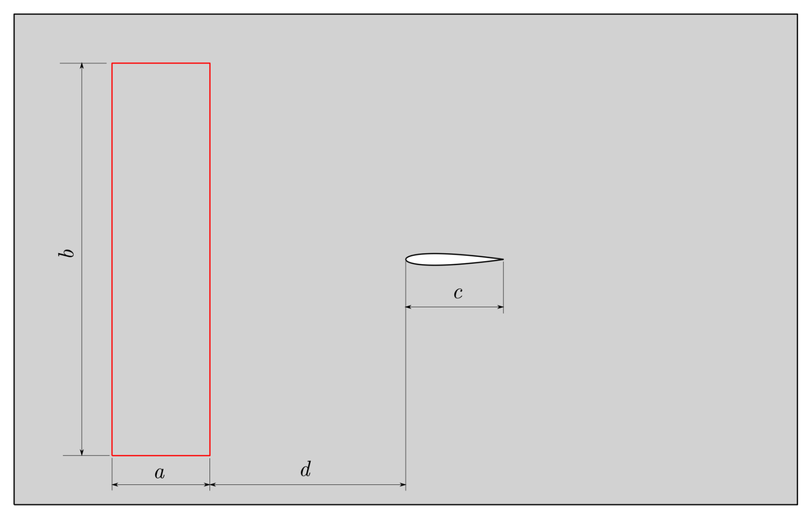


Figure. 6.1 An illustration of the problem setup for the simulation of wind gust impinging on a NACA0012 aerofoil. A sinusoidal gust is generated in the region enclosed by the box of width a and height b, which is located at a distance d upstream to the aerofoil of chord length c.

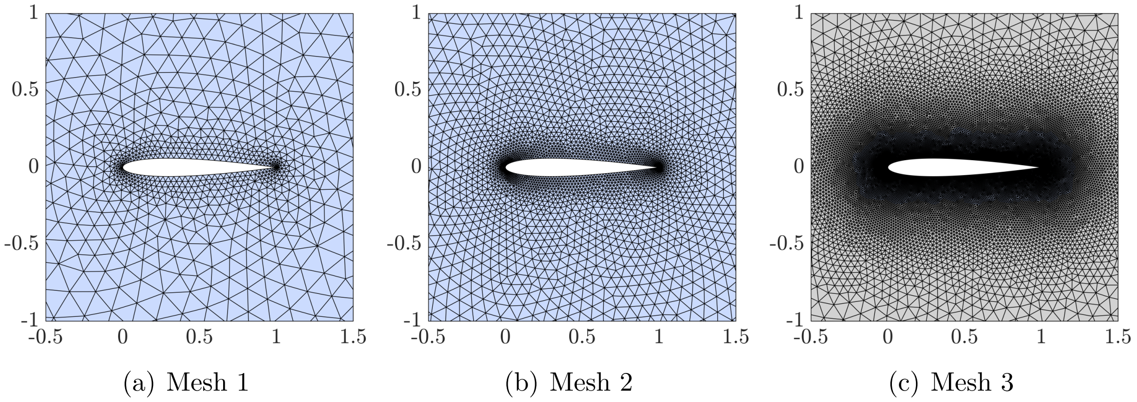
at a distance d from the aerofoil where the gust is introduced as a source term. The far-field boundary is situated at 10 chord lengths from the aerofoil.

6.1.2 Mesh convergence study

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 6.2. The meshes are generated such that the geometry is discretised to be represented accurately and that the domain near the aerofoil is discretised to obtain a high-fidelity representation of the flow features, such as the large gradients near the leading and trailing edges. The generated meshes contain 2295, 7701, 35 425 and 133 459 elements, respectively, and the aerofoil is discretised with 101, 179, 375 and 725 points in each case. The minimum mesh spacing at the leading edge is 0.008, 0.004, 0.002 and 0.001 and the spacing at the trailing edge is 0.012, 0.006, 0.003 and 0.0015 respectively.

Figure 6.3 shows the computed non-dimensional lift coefficient C_L, as a function of the number of elements, where

$$C_L = \frac{2F_L}{\rho||\boldsymbol{v}||^2 S}, \tag{6.1.1}$$



(a) Mesh 1

(b) Mesh 2

(c) Mesh 3

Figure. 6.2 Meshes used in the selection of the level of mesh refinement required to accurately capture the steady state solution.

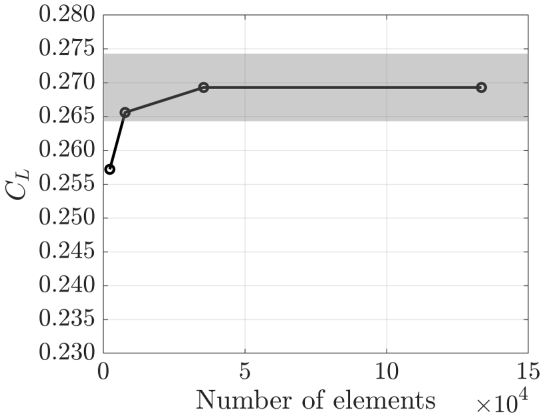
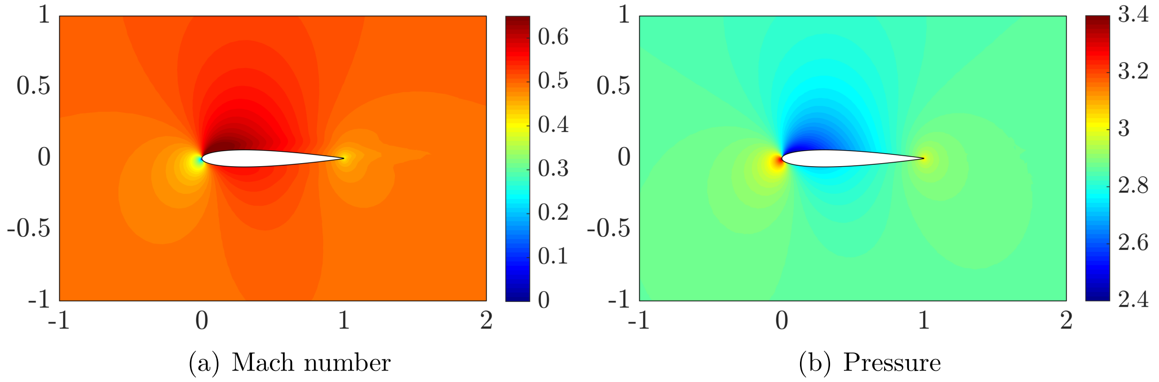


Figure. 6.3 Convergence plot of the lift coefficient C_L as a function of the number of elements. The shaded area represents the region with an error within five lift counts compared to the reference solution.

lift force, F_L, is the component of the total force acting on the aerofoil orthogonal to the flow direction. The total force is obtained by integrating the pressure over the aerofoil surface. Here, S is the reference area, taken to be unity. From this study, it can be concluded that the second mesh provides the required 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 6.4.

6.1.3 Gust problem specification

The simulation of the sinusoidal gust impinging on the NACA aerofoil is considered. To speed up the convergence to the time harmonic steady state, the computed steady



(a) Mach number

(b) Pressure

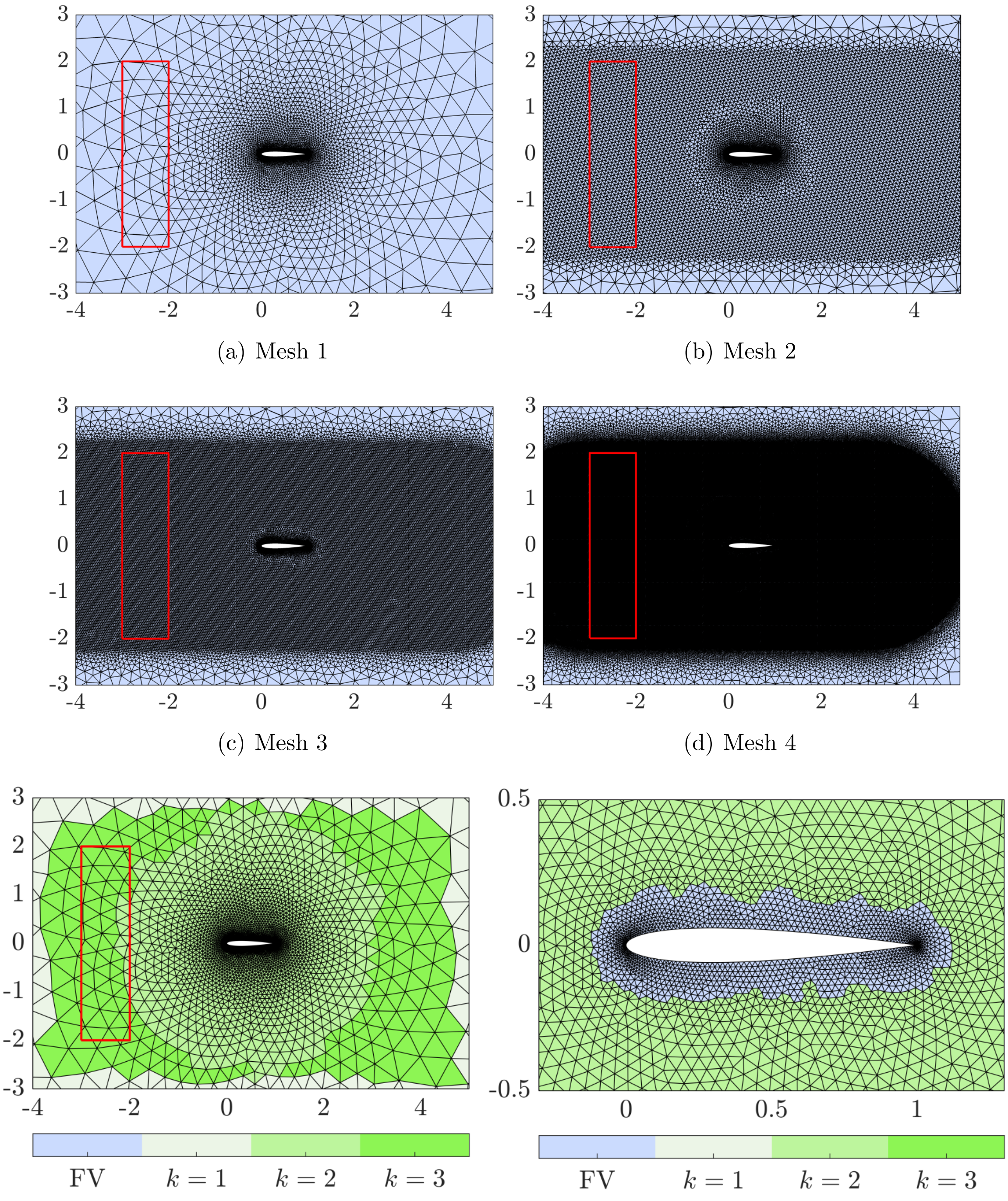
Figure. 6.4 Steady state solution computed on the mesh displayed in Figure 6.2 (b).

state solution is used as the initial condition for the transient gust simulation. The non-dimensional parameters of the source term in Equation (2.2.3) required to introduce the gust are the angular frequency, $\omega^g = 4$, the angle of propagation of the gust front, $\theta = 45°$, the gust intensity, $\delta^g = 0.1$, the dimensions of the box where the gust is generated, $a = 1$ and $b = 4$, and the 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 6.5 (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 $h^\star = 0.08$. This corresponds to having 18 points per wavelength of the impinging gust. The resulting mesh, displayed in Figure 6.5 (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 $h^\star = 0.04$ and the resulting mesh has 36 points per wavelength. The mesh, displayed in Figure 6.5 (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 edge on the aerofoil used in the steady state simulation. The resulting mesh, displayed in Figure 6.5 (d), has 265 237 elements.

6.1.3.1 Domain partitioning for HDG-FV method

Finally, the same mesh considered in the steady state simulation is partitioned into two regions as follows. Elements from the discretised domain Ω are included into the FV domain Ω^2 layer-by-layer, beginning from the elements surrounding the aerofoil. In this manner, the FV domain continues to grow until elements larger than a pre-determined



(a) Mesh 1

(b) Mesh 2

(c) Mesh 3

(d) Mesh 4

(e) Mesh 5

Figure. 6.5 Unstructured triangular meshes employed to simulate the wind gust impinging on a NACA0012 aerofoil. (e) A detailed view of the partitioned mesh close to the aerofoil is presented on the right, highlighting the HDG and FV subdomains.

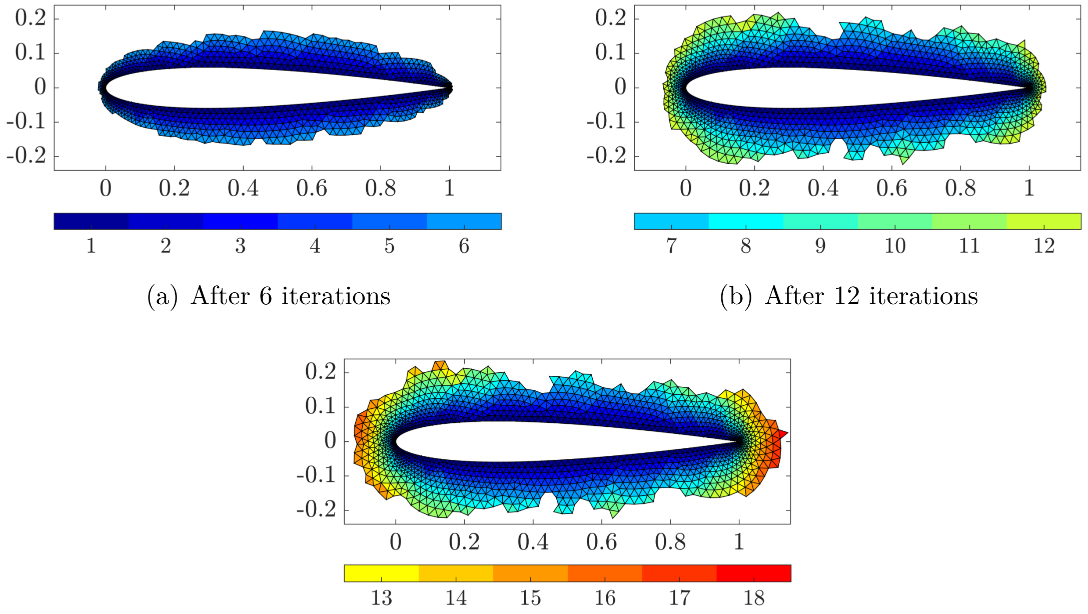
size are encountered. Thus, all elements with a size less than or equal to the desired element size and close to the aerodynamic configuration lie in the FV domain. See Algorithm 4 for an overview of the implementation of the partitioning procedure. Figure 6.6 depicts the growth of the FV domain where elements are added over the span of several iterations. For 2D cases, elements of sizes upto $3h^\star/2$ are contained in the FV domain Ω^2, whereas the rest of the mesh defines the HDG region Ω^1.

```
Mark all nodes lying on the aerofoil
repeat
    Initialise: number of new FV elements = 0
    for all elements previously not identified as lying in FV domain and having
      atleast one marked node do
        for element size <= desired element size do
            Identify element as lying in FV domain
            Mark all the element nodes
            Increment by 1 the number of new FV elements
        end
    end
    for all elements previously not identified as lying in FV domain do
        if all element nodes are marked then
            Identify element as lying in FV domain
            Increment by 1 the number of new FV elements
        end
    end
until until no new FV elements are identified;
```

Algorithm 4: Partitioning of Ω for the coupled HDG-FV method

In the HDG elements lying in the region of interest, the degree of functional approximation is adapted based on the wavelength of the impinging gust, $\lambda = \pi/2 \approx 80h^\star$. 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 6.5 (e), including the degree of approximation used in the different elements.

The time step size Δt is decided based on the time period of the gust T. A comparative study is performed with 16, 32 and 64 time steps per cycle of the gust. Figure 6.7 shows the variation in the lift coefficient observed for different time step sizes. The $\Delta t = T/32$ was selected since it provides a solution to within 5 lift counts when compared to $\Delta t = T/64$. This corresponds to a CFL number of approximately 442 in the coarsest mesh and 478 in the finest mesh. The fully-implicit time-stepping procedure allows for arbitrarily large time step sizes that are permissible with respect to the time scale of the



(a) After 6 iterations

(b) After 12 iterations

(c) After 18 iterations

Figure. 6.6 An illustration of the workflow of the partitioning algorithm depicting the growth of the FV domain around the aerodynamic configuration over several iterations.

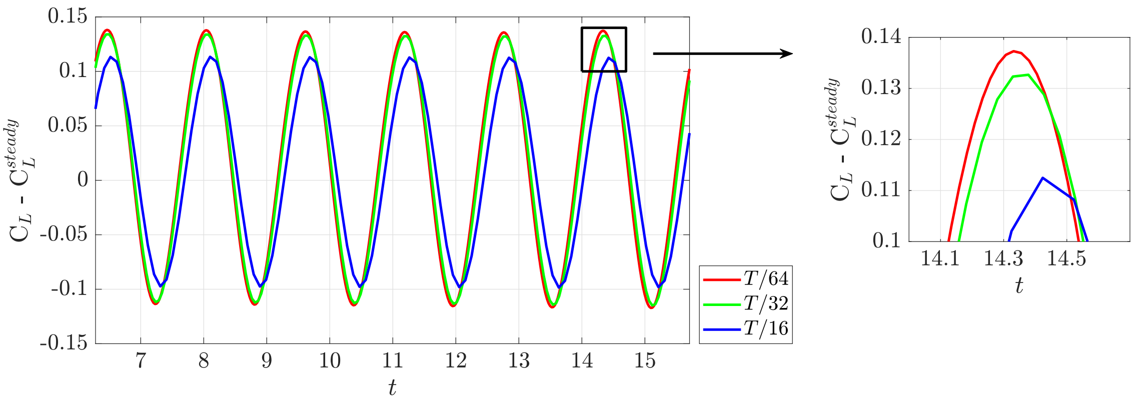


Figure. 6.7 Variation of the lift coefficient with respect to the steady state solution as a function of the non-dimensional time for the simulations computed with various time step sizes.

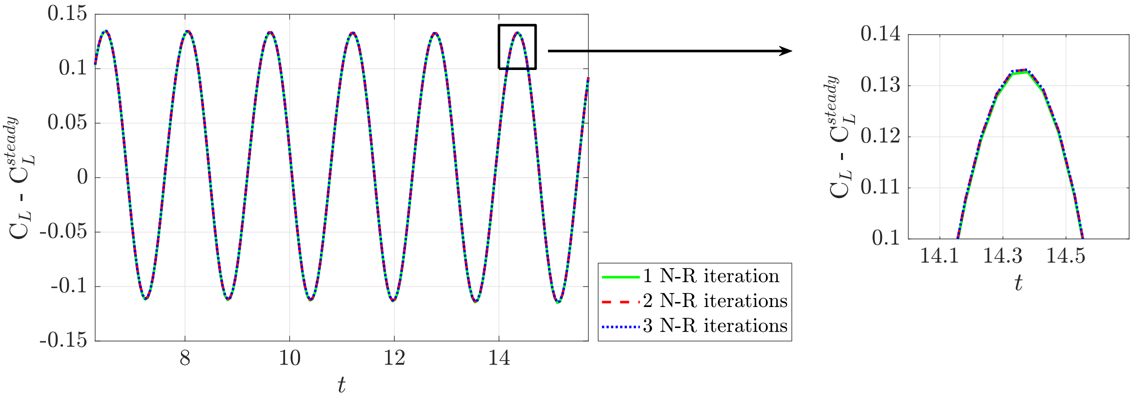
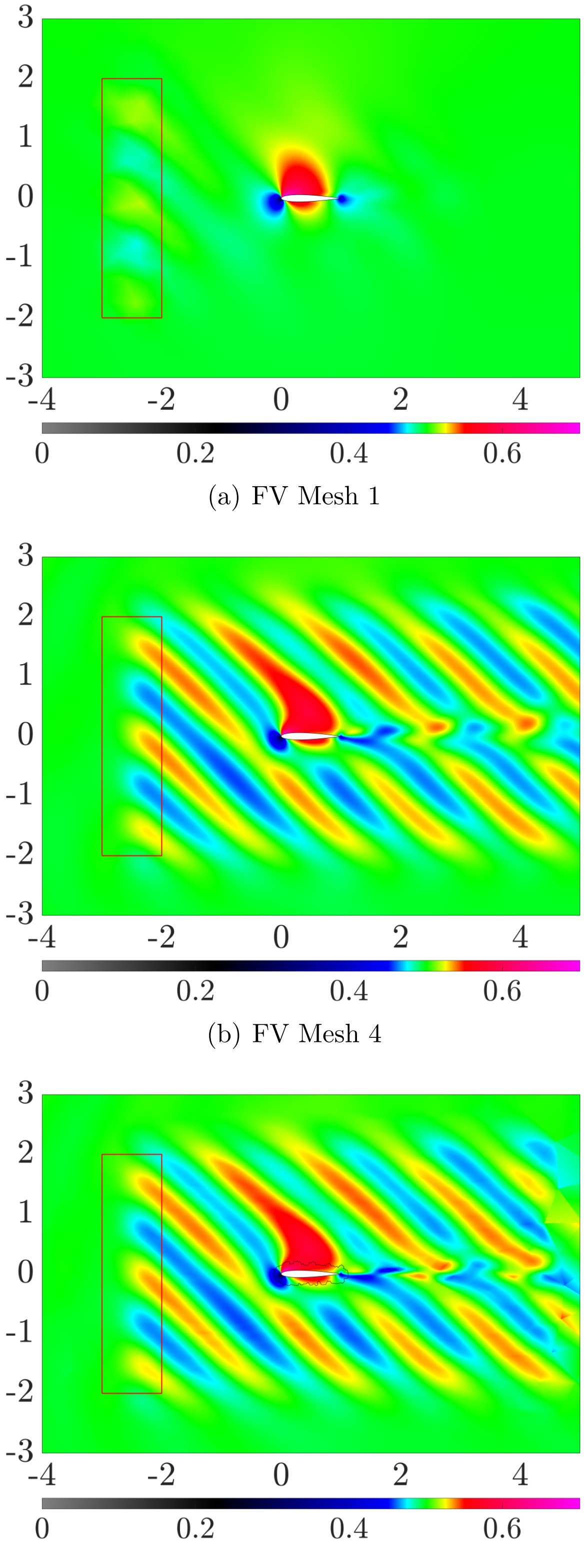


Figure. 6.8 Variation of the lift coefficient with respect to the steady state solution as a function of the non-dimensional time for the simulations computed with different number of linearisation iterations performed in each time step calculation.

fluid phenomena. In all cases, the second order BDF2 time integrator is employed. The tolerance for the Newton-Raphson linearisation was set to 1.0×10^{-2}. This resulted in the linearisation procedure performed once per time step. A comparative study was carried out to assess the influence of the number of linearisation iterations. Figure 6.8 depicts the variation of the lift coefficient as a function of non-dimensional time for different iteration counts. Clearly, one iteration per time step is found to provide accurate solutions.

6.1.4 Results and discussion

Figure 6.9 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 6.5. The solution obtained with the FV method on the mesh of Figure 6.5 (a) shows, as expected, the large dissipation introduced by the traditional FV scheme on coarse meshes. This experiment confirms that meshes designed for steady state simulations are not suitable for transient simulations in a low-order framework. Figure 6.9 (b) shows the FV solution computed on the fine mesh of Figure 6.5 (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 6.5 (e), showing a good agreement with the reference solution computed with FV in the finest mesh.



(a) FV Mesh 1

(b) FV Mesh 4

(c) HDG-FV Mesh 5

Figure. 6.9 Mach number distribution for the simulation of the wind gust impinging in a NACA0012 aerofoil after the time harmonic steady state is reached.

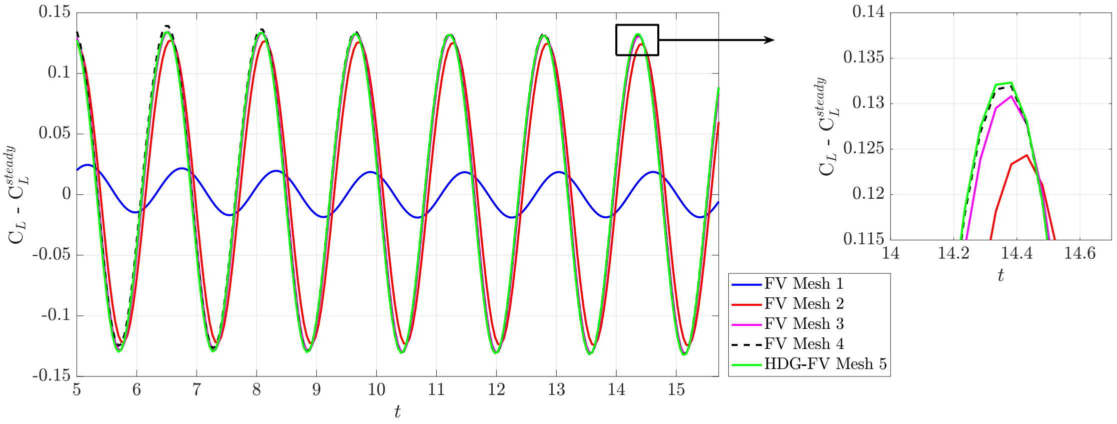


Figure. 6.10 Variation of the lift coefficient with respect to the steady state solution as a function of the non-dimensional time for the simulations computed on the five meshes shown in Figure 6.5.

6.1.4.1 Solution accuracy

To better illustrate the accuracy of the proposed scheme, Figure 6.10 shows the evolution of the lift coefficient in time for the solutions computed on the five meshes shown in Figure 6.5. 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 errors.

The dissipation error is estimated by comparing the amplitude of the oscillations in the lift coefficient against the reference results. For the FV 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 6.11 shows a one dimensional section, at $y = c/2$, of the vertical velocity field v for the five simulations computed on the five meshes shown in Figure 6.5. 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

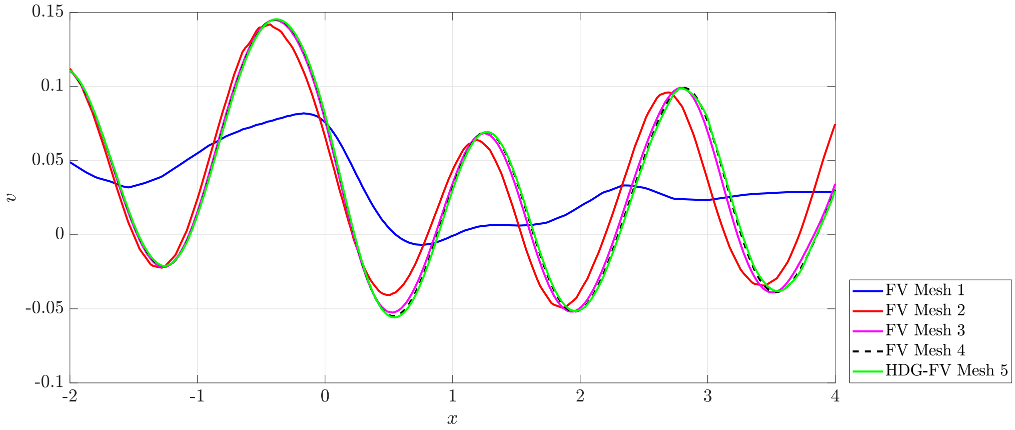


Figure. 6.11 One dimensional section, at $y = c/2$, of the vertical velocity field v for the five simulations computed on the five meshes shown in Figure 6.5.

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

6.1.4.2 Computational efficiency

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 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.

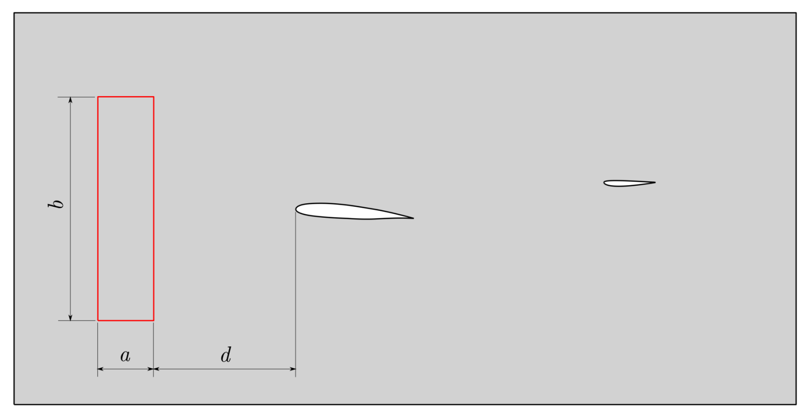


Figure. 6.12 Illustration of the problem setup for the simulation of wind gust impinging on the transverse section of wing and tail configuration. A sinusoidal gust is generated in the region enclosed by the box of width a and height b, which is located at a distance d upstream to the wing.

6.2 Gust flow in 2D: 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 [116]. 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.

6.2.1 Problem setup and gust specification

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 6.12, 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 $d = 3.04$.

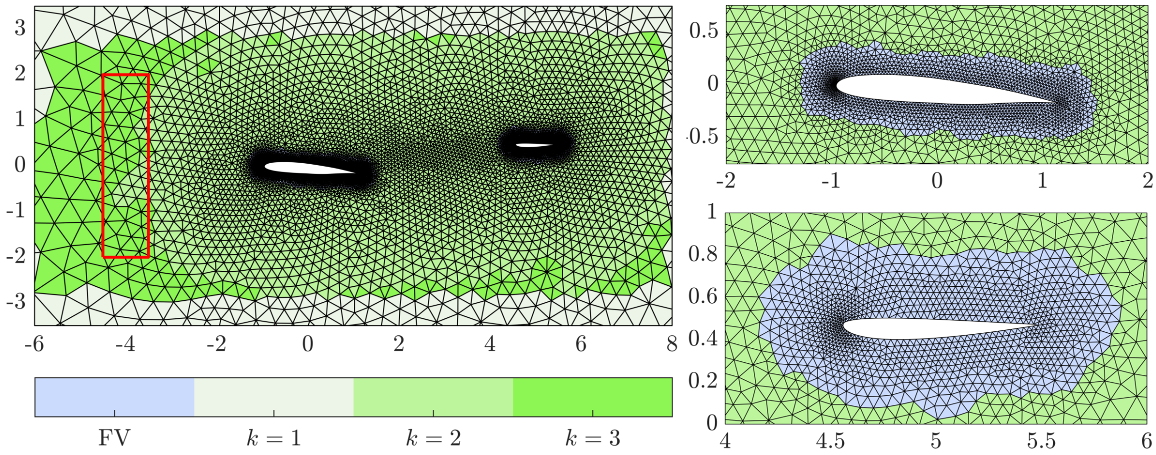


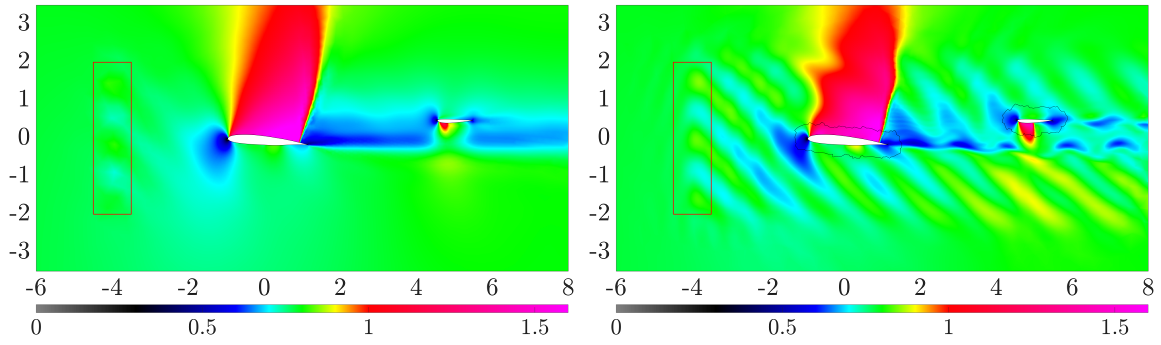
Figure. 6.13 Unstructured triangular mesh employed to simulate the wind gust impinging in a wing-tail configuration. The two detailed views around the wing and tail show the partition in HDG and FV regions.

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 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 6.13. 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. Also similar to the previous example, one linearisation iteration is performed per time step.

6.2.2 Results and discussion

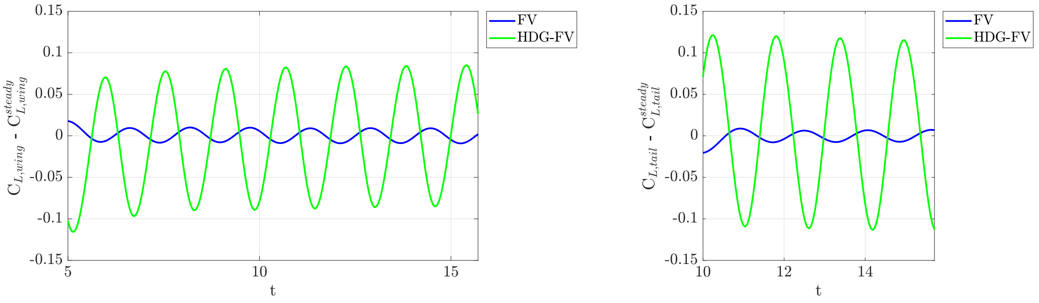
The solution after the time harmonic steady state is achieved for both the standard FV scheme and the proposed HDG-FV method are shown in Figure 6.14. The results illustrate the substantial dissipation introduced by the FV scheme when the coarse mesh, suitable for a steady state simulation, is used. Instead, the solution with the proposed HDG-FV scheme is able to capture the perturbation of the velocity induced by



(a) FV

(b) HDG-FV

Figure. 6.14 Mach number distribution for the simulation of the wind gust impinging in a wing-tail configuration after the time harmonic steady state is reached.



(a) Wing

(b) Tail

Figure. 6.15 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 6.14 for the wing and the tail.

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.

To further illustrate the benefits of the proposed approach, Figure 6.15 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.

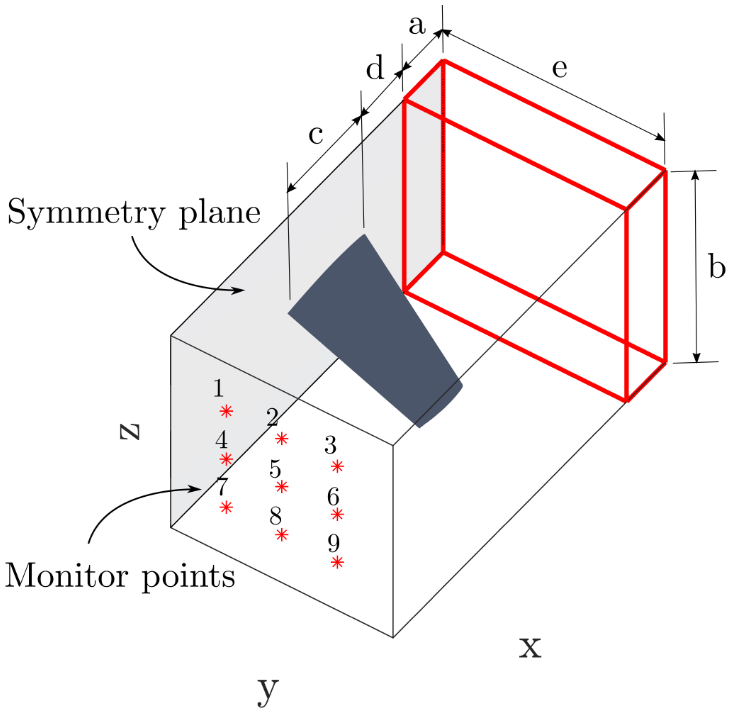


Figure. 6.16 An illustration of the problem setup for the wind gust simulation of ONERA M6 wing. The gust is generated in the region enclosed by the rectangular box depicted in red. Also indicated are the region of interest for the gust flow problem, enclosed by the rectangular box in black and the locations of monitor points in *

6.3 Gust flow in 3D: ONERA M6 wing

6.3.1 Problem setup

A popular model designed to be used for three dimensional flows from low to transonic speeds is the ONERA M6 wing[132]. The geometry set-up for the gust flow testing of this symmetrical, swept-back wing is shown in Figure 6.16. The wing is immersed in an inviscid transonic flow with free-stream Mach number $M^{\infty} = 0.84$ and angle of attack 3.06 degrees. The free-stream flow is taken to be parallel to the $x-z$ plane, thus enabling the use of a symmetry plane as indicated. The rectangular box of dimensions $a \times e \times b$ at a distance d from the leading edge of the wing of chord length $c = 10$ denotes the region of gust generation. The far-field boundary is located at 12.5 chord lengths from the wing.

6.3.2 Mesh convergence study

The first step is to obtain a mesh size suitable for steady-state simulations. Four unstructured tetrahedral meshes are considered. Successive refinements close to the wing is performed to in order to obtain an accurate representation of the geometry and quantities of interest i.e., lift and drag. The surface triangulation of the first three meshes are presented in Figure 6.17. The generated meshes contain 236 682, 720 667, 3 562 293, and 5 044 644 elements and the wing is represented by 11 592, 15 214, 54 372, and 105 326 faces respectively. The minimum mesh spacing at the leading edge is 0.095, 0.052, 0.023 and 0.018 and the spacing at the trailing edge is 0.099, 0.071, 0.037 and 0.024 respectively. The second and the third meshes involve a stretching along the span of the wing by a factor of 5.

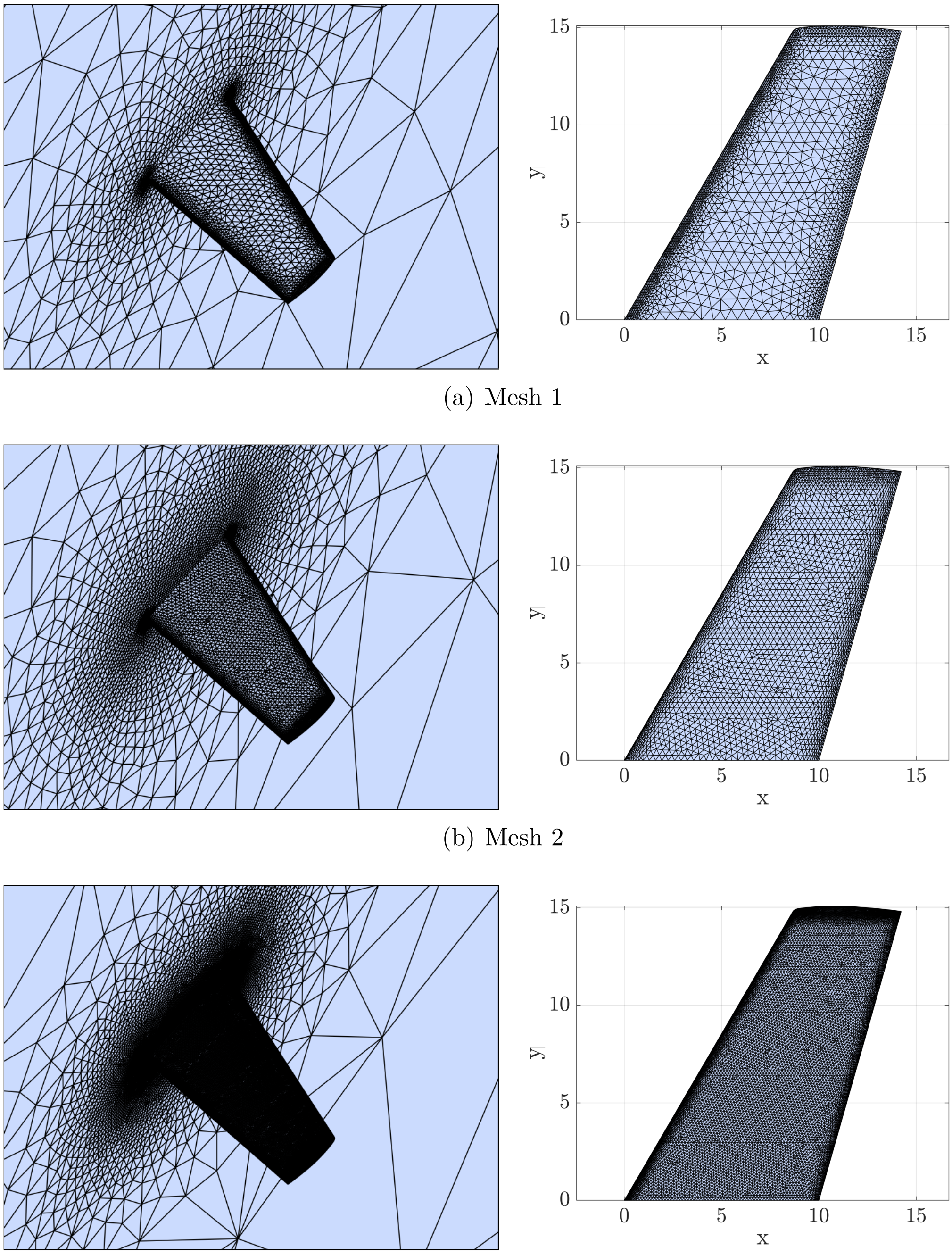
The non-dimensional lift coefficient as a function of the number of elements is plotted in Figure 6.18. Here, C_L is computed using Equation (6.1.1), with reference area of 239.12. The convergence of the lift coefficient shows that the second mesh in Figure 6.17 provides an accurate value to within 2.5 lift counts from the reference value and chosen to perform the gust simulations with the coupled HDG-FV method. The value of 2.5 lift counts represents the same allowable limit to the C_L variation in the 2D example in-terms of the percentage change in C_L. The steady state Mach number distribution on the wing, shown in Figure 6.19(a), indicates the present of strong shocks. Further, Figures 6.19(b-d) show the variation of the non-dimensional quantity, the pressure coefficient C_p, which is a measure of the relative pressure at a location, given by,

$$C_p = \frac{2(p - p^\infty)}{\rho^\infty ||\boldsymbol{v}^\infty||^2}, \tag{6.3.1}$$

for different cross-sections of the wing along with experimental measurements reported in [132]. The location of the shocks are captured well but they appear smeared. In practice, when necessary, sharp resolution of the shocks is obtained by designing refined mesh elements at the location of the shock. The solutions are obtained using the FV method, which provides efficient computations for steady simulations as compared to the coupled HDG-FV method. It is to be noted that the solutions obtained by both methods quantitatively match each other.

6.3.3 Gust problem specification

Transient gust flow simulations are considered next. The dimensions of the rectangular box where the gust is generated is chosen to be $a = 5$, $e = 20$ and $b = 20$ and is at a



(a) Mesh 1

(b) Mesh 2

(c) Mesh 3

Figure. 6.17 Surface triangulation of the first three meshes for the mesh convergence study of the ONERA M6 wing.

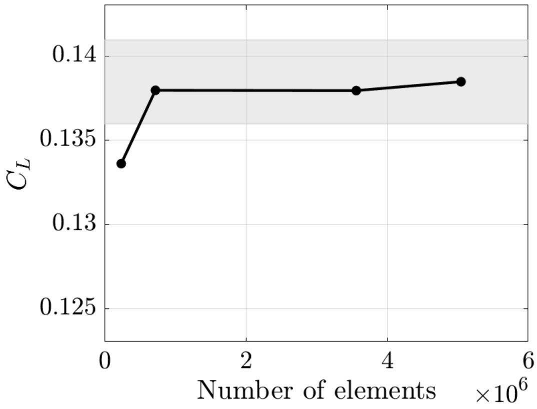
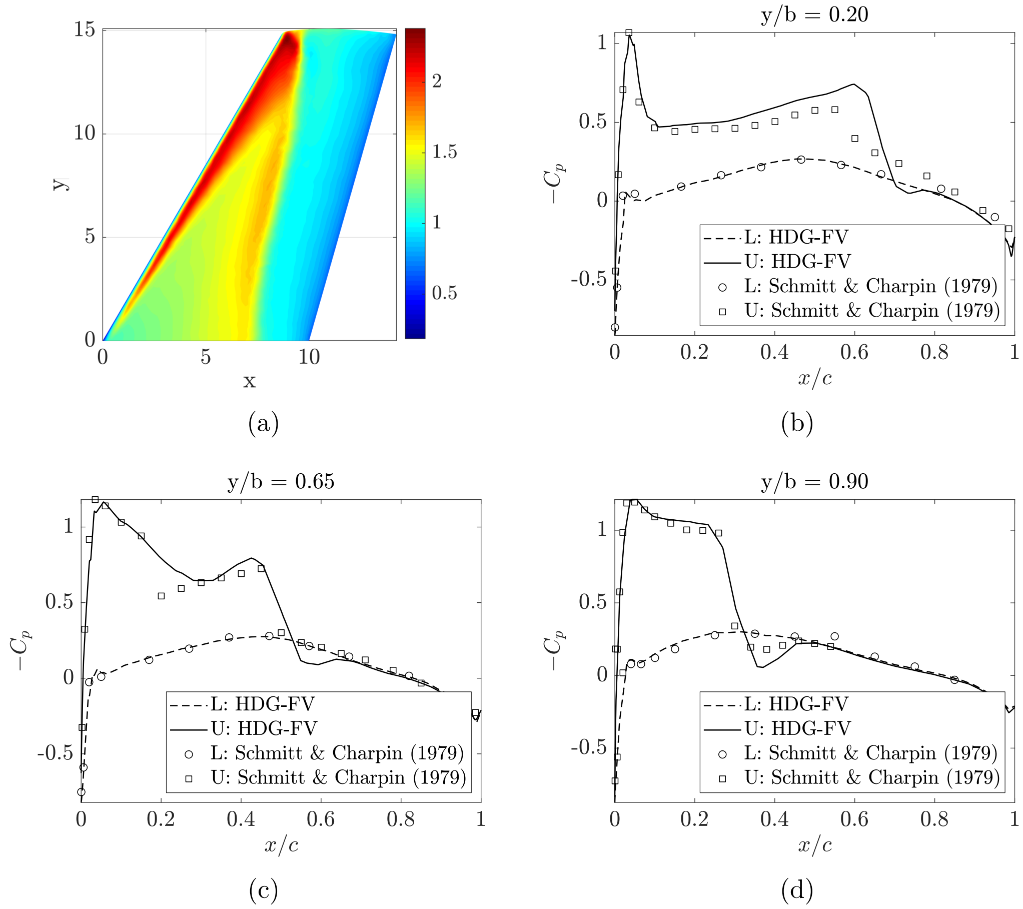


Figure. 6.18 Convergence plot of lift coefficient C_L as a function of the number of elements. The shaded region depicts the region within 2.5 lift counts of the reference value.



(a)

(b)

(c)

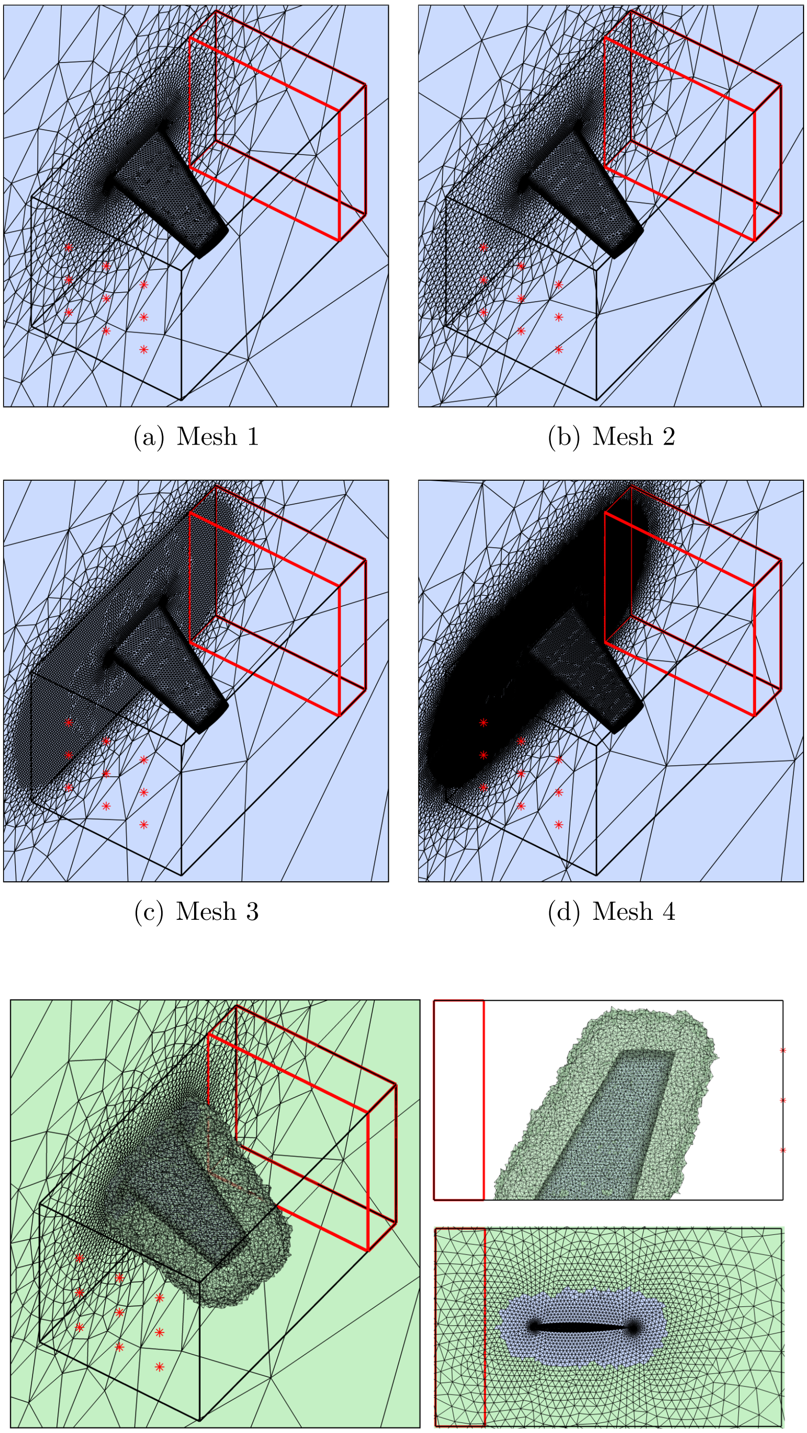
(d)

Figure. 6.19 Steady state (a) Mach number distribution on the top of the wing and (b-d) the variation of pressure coefficient C_p at different cross-sections along the wing span on the lower (L) and the upper (U) surfaces.

distance $d = 5$ upstream of the leading edge. The parameters of the gust specification in Equation (2.2.3) are, angular frequency, $\omega^g = 0.4$, inclination of the wave front, $\theta = 45°$, wavelength, $\lambda = 5\pi$, and the gust intensity, $\delta^g = 0.1$. Note that the wavelength is scaled with respect to the chord length of the wing to be similar to that of the 2D gust flow over NACA0012 in the previous section. The steady state solution is used as the initial condition to accelerate to a time-harmonic solution for the gust problem. The region of interest for the gust flow problem is taken to be $\widetilde{\Omega} = [-10, 25] \times [0, 20] \times [-10, 10]$. The solution is monitored at various locations downstream of the wing. These monitor points, as shown in Figure 6.16, are located at the corners, edge centres and centroid of the rectangular region $25 \times [5, 15] \times [-5, 5]$.

The coupled HDG-FV method performs the simulation using the same mesh determined to be suitable for steady simulations in the previous step. On the other hand, three additional meshes are designed for the FV method. The first mesh, as shown in Figure 6.20(a), considers the same mesh employed for the steady-state solution. The second mesh, as displayed in Figure 6.20(b), considers the entire region of interest $\widetilde{\Omega}$ to be refined with a desired element size of $h^\star = 1.0$. The third mesh considers $\widetilde{\Omega}$ to be refined with elements of size $h^\star = 0.5$ as depicted in Figure 6.20(c). The fourth mesh considers a further refinement with element size $h^\star = 0.25$ in $\widetilde{\Omega}$ and is as shown in Figure 6.20(d). The refinements applied to the latter three meshes lead to 874 045, 2 599 755 and 17 961 056 elements and correspond to having 16, 32 and 64 points per wavelength of the sinusoidal gust, respectively.

The partitioning of the mesh into FV and HDG subdomains follows the procedure outlined in Algorithm 4. The desired max element size in the FV domain is chosen to be $1.1h^\star$, where $h^\star$ is the largest element on the wing. With consideration for the presence of stretched elements, the element size is measured by the radius of inscribed spheres, in contrast to using the largest edge length as a measure of element size in 2D elements. Figure 6.20(e) depicts the resulting interface between the subdomains consisting of triangular faces of arbitrary orientation. The interface is handled as described previously in Chapter 5. Furthermore, the HDG elements in the region of interest are assigned the degree of approximation k based on the wavelength of the impinging gust. For elements of sizes less than $\lambda/5$, a quadratic approximation is used. The remaining elements of up to size $\lambda/3$ are assigned a cubic approximation. Rest of the elements are given a quartic approximation. This results in 189 320 quadratic, 841 cubic and 58 quartic elements as depicted in Figure 6.21. As with 2D cases, linear approximation is employed in the HDG domain outside the region of interest, resulting in 26 994 linear elements.



(a) Mesh 1

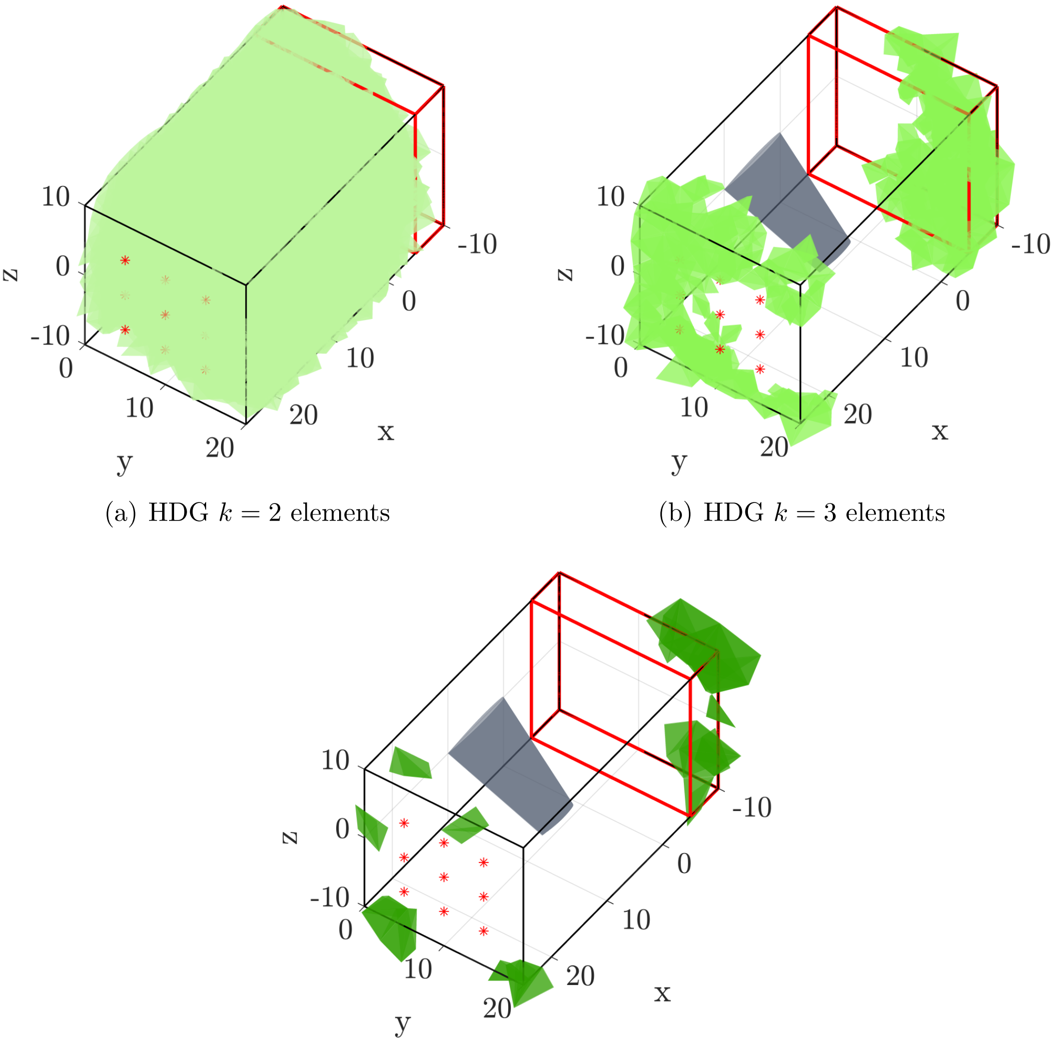
(b) Mesh 2

(c) Mesh 3

(d) Mesh 4

(e) Mesh 5

Figure. 6.20 Surface triangulation of the unstructured tetrahedral volume meshes employed for the wind gust simulations for the ONERA M6 wing. (e) A detailed view of the HDG-FV interface.



(a) HDG $k = 2$ elements

(b) HDG $k = 3$ elements

(c) HDG $k = 4$ elements

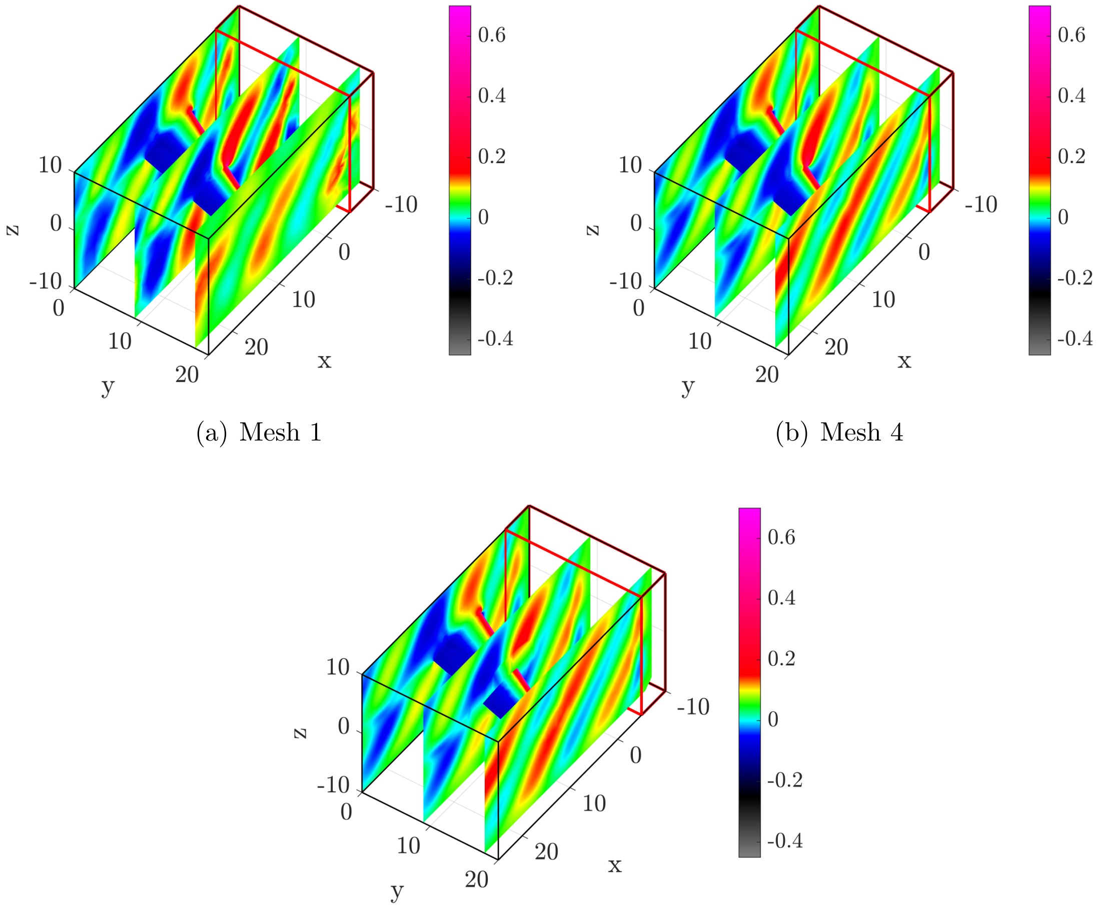
Figure. 6.21 Distribution of HDG elements of various orders in the domain of interest for the gust simulation for the ONERA M6 wing.

In all the gust simulations, the time step size Δt corresponds to performing 32 time steps per cycle of the gust. For the coupled HDG-FV method, BDF2 was employed and the Δt corresponds to a CFL number of 185. Three iterations of the Newton-Raphson linearisation performed at each time step calculation was found to result in sufficiently accurate solution, which corresponded to a tolerance of 1×10^{-4}. The coupled HDG-FV method on Mesh 5 results in a sparse linear system of size 12 958 555. This was solved at each Newton iteration using the GMRES solver with ILU pre-conditioner [131]. The tolerance for convergence of the residual of the linear solver was set to 1×10^{-9}. In order to handle the large problem sizes encountered for the FV simulation on the meshes designed for the gust problem, a fast in-house code, the ‘FLITE system’, was utilised. The code performs a fully-parallel, second-order FV discretisation with multigrid acceleration and dual time-stepping procedure [146, 144]. Hence, a comparison of computational costs between the FV and the proposed method was not feasible.

6.3.4 Results and discussion

Figure 6.22 illustrates the variation in the z-momentum, ρw, induced by the sinusoidal gust after a time-harmonic steady-state has been reached for three chosen meshes depicted in Figure 6.20. In practice, for the gust phenomena on airframes, the sinusoidal perturbations are generated ahead of the nose of the aircraft, several chord lengths upstream from the lift generating structures, as demonstrated in the 2D cases. This ensures that the effect of the fuselage on the gust is taken into account before impinging on the wing. Here, the gust is generated closer to the wing leading to a smaller region of interest in-order to reduce the problem size.

In-part due to the proximity of the region of gust generation to the wing and availability of finer elements in the region of interest, as compared to 2D examples in the previous sections, the dissipation is not drastic but noticeable. Figure 6.22(a) is obtained for the FV method on the mesh designed for the steady-state simulations. Qualitative loss of accuracy can be observed, particularly along the section at $y = 18$ as compared to the FV solution on the finest mesh shown in Figure 6.22(b), which is taken to be the reference solution. Hence. the low-order FV solution to unsteady flows on a mesh designed for steady simulations is not satisfactory, even in the presence of a reduced number of coarse elements. On the other hand, the solution in Figure 6.22(c) for the coupled schemes closely matches with that of the reference solution. Thus the coupling of HDG and FV methods to re-use the meshes generated for steady state solutions in a transient simulation is feasible and advantageous.



(a) Mesh 1

(b) Mesh 4

(c) Mesh 5

Figure. 6.22 Plots of z-component of momentum, ρw, obtained for different meshes of the gust flow problem for the ONERA M6 wing immersed in a transonic inviscid flow.

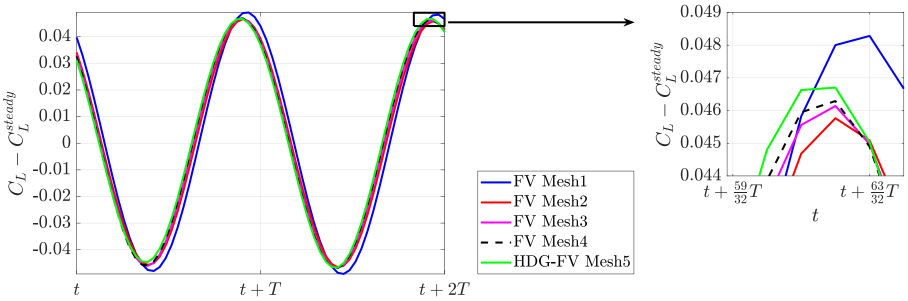


Figure. 6.23 Variation of the lift coefficient with respect to the steady state solution as a function of the non-dimensional time for the ONERA M6 wing immersed in a sinusoidal gust flow.

6.3.4.1 Solution accuracy

To better illustrate the solution accuracy of the proposed method, Figure 6.23 shows the evolution of the lift coefficient for the FV solution on different meshes, along with the reference solution and the coupled HDG-FV scheme with non-uniform degree approximations. Dissipation and dispersion errors are computed to quantify the solution accuracy by comparing the amplitude and phase of the lift coefficient variation against the reference solution. For the FV simulations on the first three meshes, the dissipation was found to be 4.5%, 1.0% and 0.3%. The dissipation error for the combined scheme was 0.4%. FV solution on the second and the third mesh along with the combined scheme solution lie within one lift count of the reference method, whereas the FV solution on the first mesh shows a larger variation and thus demonstrates the unsuitability of FV method on meshes designed for steady-state solution. Also, dispersion error computed for the FV simulations are 10.7°, 4.7° and 1.5°, whereas it is 9.5° for the combined scheme. The superior properties of the coupled method to repurpose meshes routinely designed for steady simulations to transient processes are demonstrated. As commented earlier, the dispersion and dissipation errors as measured by the lift coefficient is not large as compared to the 2D examples in the previous examples since the gust is generated very close to the wing where the relatively smaller elements are capable of resolving the solution to a large extent.

The high-fidelity solution produced by the coupled scheme can be better demonstrated when the gust has traversed distances several times its wavelength. As shown for the two-aerofoil configuration in Section 6.2, it is critical to preserve the transient features upto large distances in practical examples. Monitor points at the downstream of the wing track the different components of the solution field over time. Figures 6.24-

6.28 show variation induced by the gust on normalised differences in the primitive variables, $\rho, \boldsymbol{v} = \{u \;\; v \;\; w\}^T$ and p, plotted in the logarithmic scale as a function of the non-dimensional time. Here, the differences are calculated with the reference value taken to be the solution on the finest mesh for FV in Figure 6.20. In the plot the for density variations given by Figure 6.24, the FV solution on first mesh shows large dispersion errors. The errors in the coupled HDG-FV solution and the FV solutions on the refined meshes are similar. The gust source term produces sinusoidal perturbations in the momentum field in the $x - z$ plane. In Figure 6.25 for the variations in the x-component of the velocity, u, FV solution on the coarse mesh produces consistently large dispersion and dissipation errors. The effect of mesh refinement in reducing the solution error is clearly visible for the second and the third meshes. The coupled HDG-FV solution consistently produces the least error across all the monitor points. Figure 6.26 depicts the variations induced in the y-component of the velocity field, v by the wing, since the gust does not produce variations in y-component of the momentum. Monitor points 4,5 and 6, situated downstream of the wing, register large errors in the FV solutions, even on the refined meshes as compared to that of the coupled method. At other monitor points, the FV solution on the coarse mesh registers the largest error, as previously observed for other variables. For the z-component of the velocity, w, in Figure 6.27, the coupled method offers the smallest solution error and the FV solution on the coarse mesh produces the largest errors. Again, the benefit of mesh refinement in improving the solution accuracy is observed for the FV solution on the second and third meshes. The vortical gust does not induce large variations in the pressure field, see Figure 6.28. Here, the scale of the errors in pressure variation in much smaller than the other primitive variables and no clear superior solution candidate may be inferred. In conclusion, unsteady FV solutions on meshes designed for steady-state problems are not feasible. The low order method requires tailored mesh refinements which in-turn leads to large problem sizes. On the other hand, the coupled FV-HDG method offers significantly accurate solutions with low computational costs to unsteady problems on the very same mesh generated routinely for steady-state calculations.

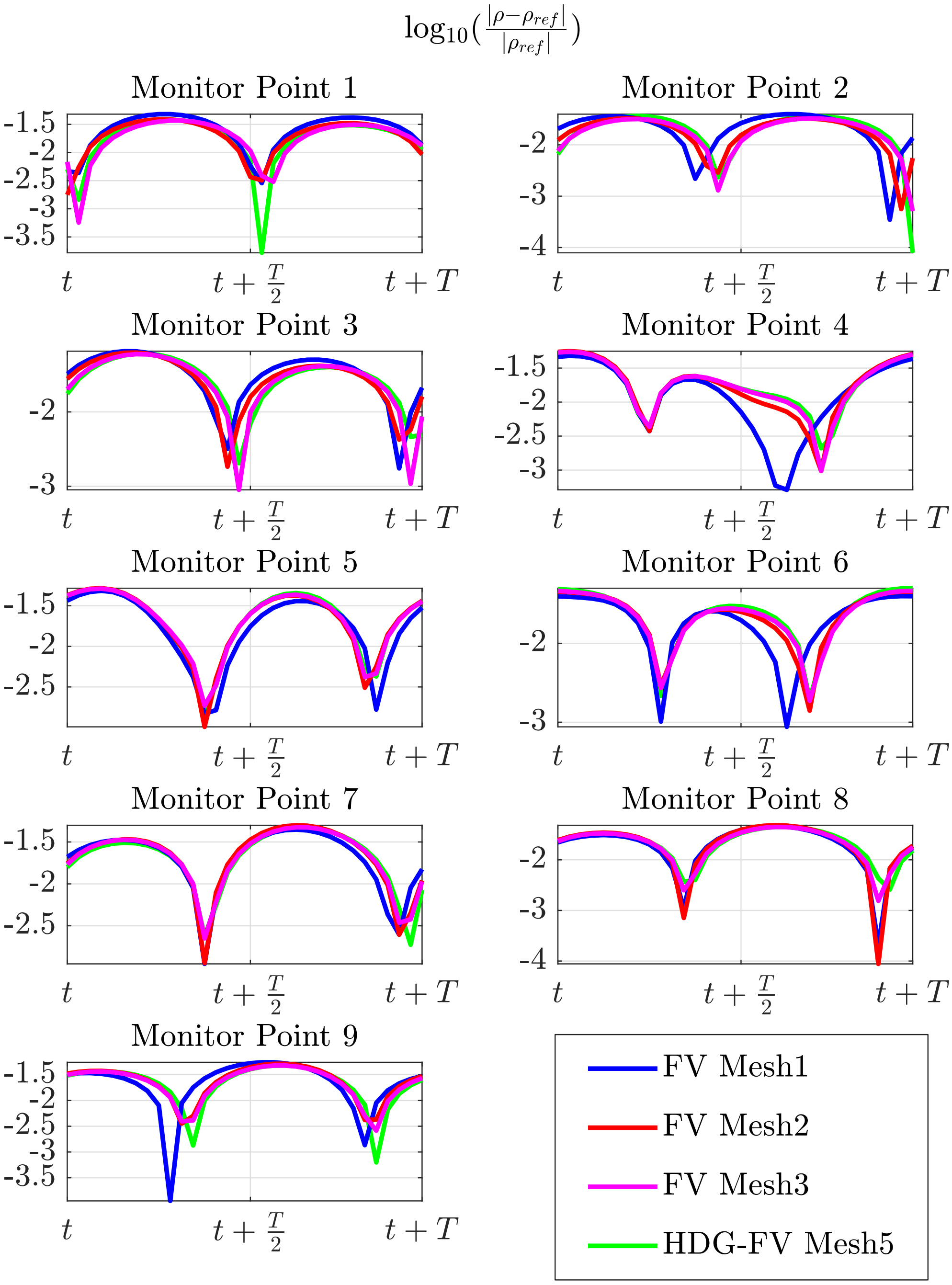


Figure. 6.24 Variation in the logarithmic scale of the normalised density at various monitor points located downstream as depicted in Figure 6.16 of an ONERA M6 wing immersed in a sinusoidal gust flow.

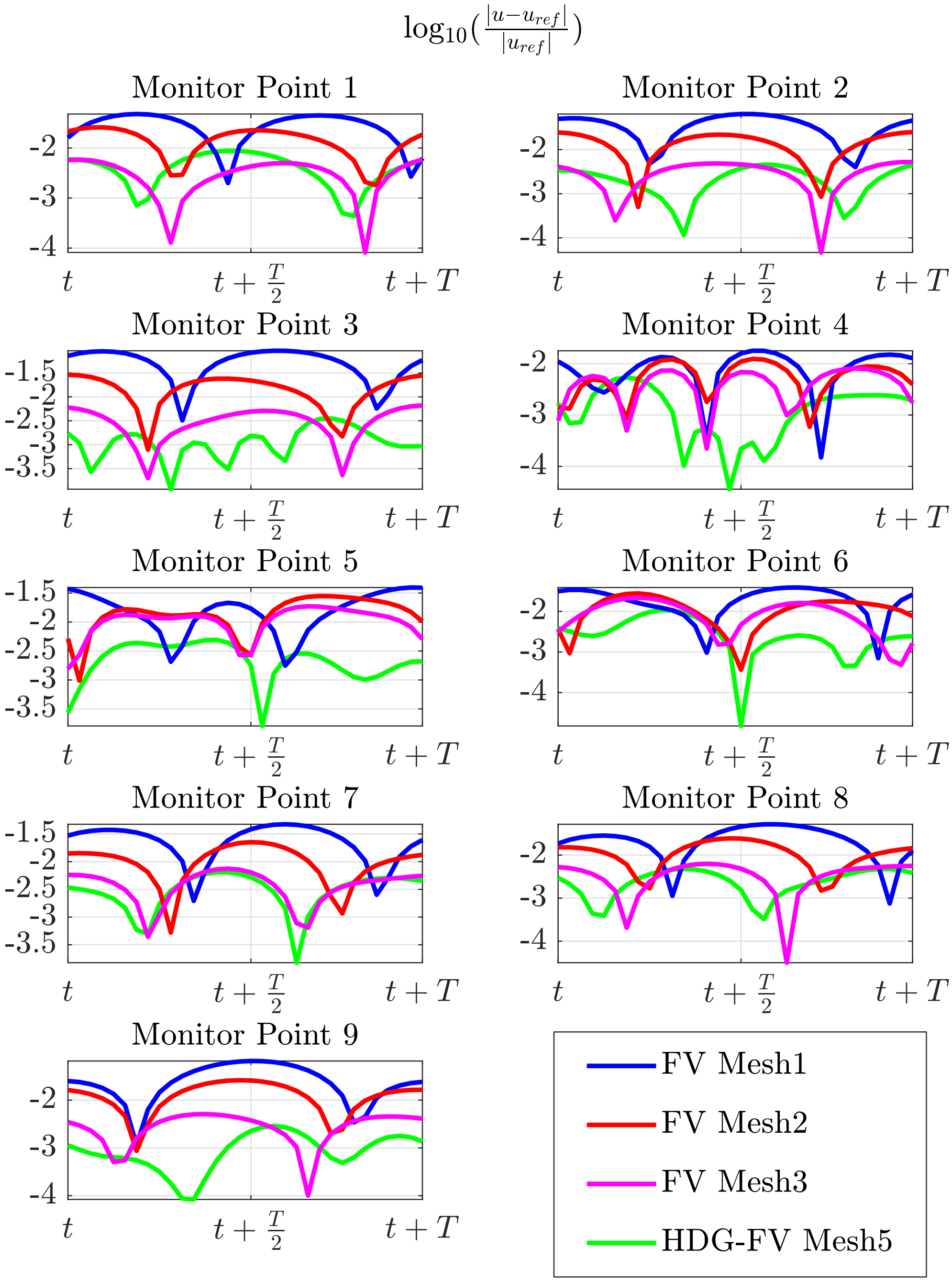


Figure. 6.25 Variation in the logarithmic scale of the normalised x-component of the velocity at various monitor points located downstream as depicted in Figure 6.16 of an ONERA M6 wing immersed in a sinusoidal gust flow.

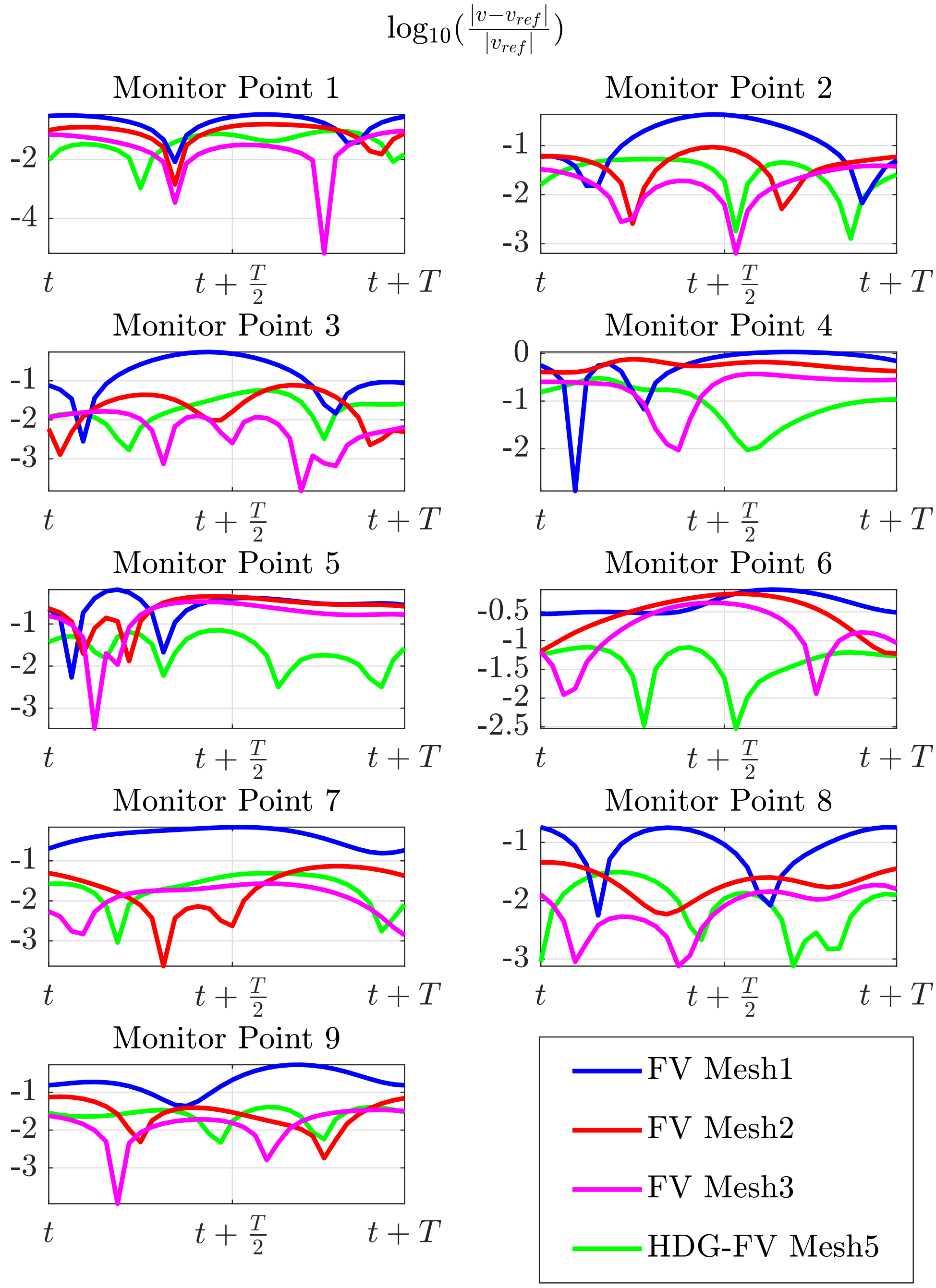


Figure. 6.26 Variation in the logarithmic scale of the normalised y-component of the velocity at various monitor points located downstream as depicted in Figure 6.16 of an ONERA M6 wing immersed in a sinusoidal gust flow.

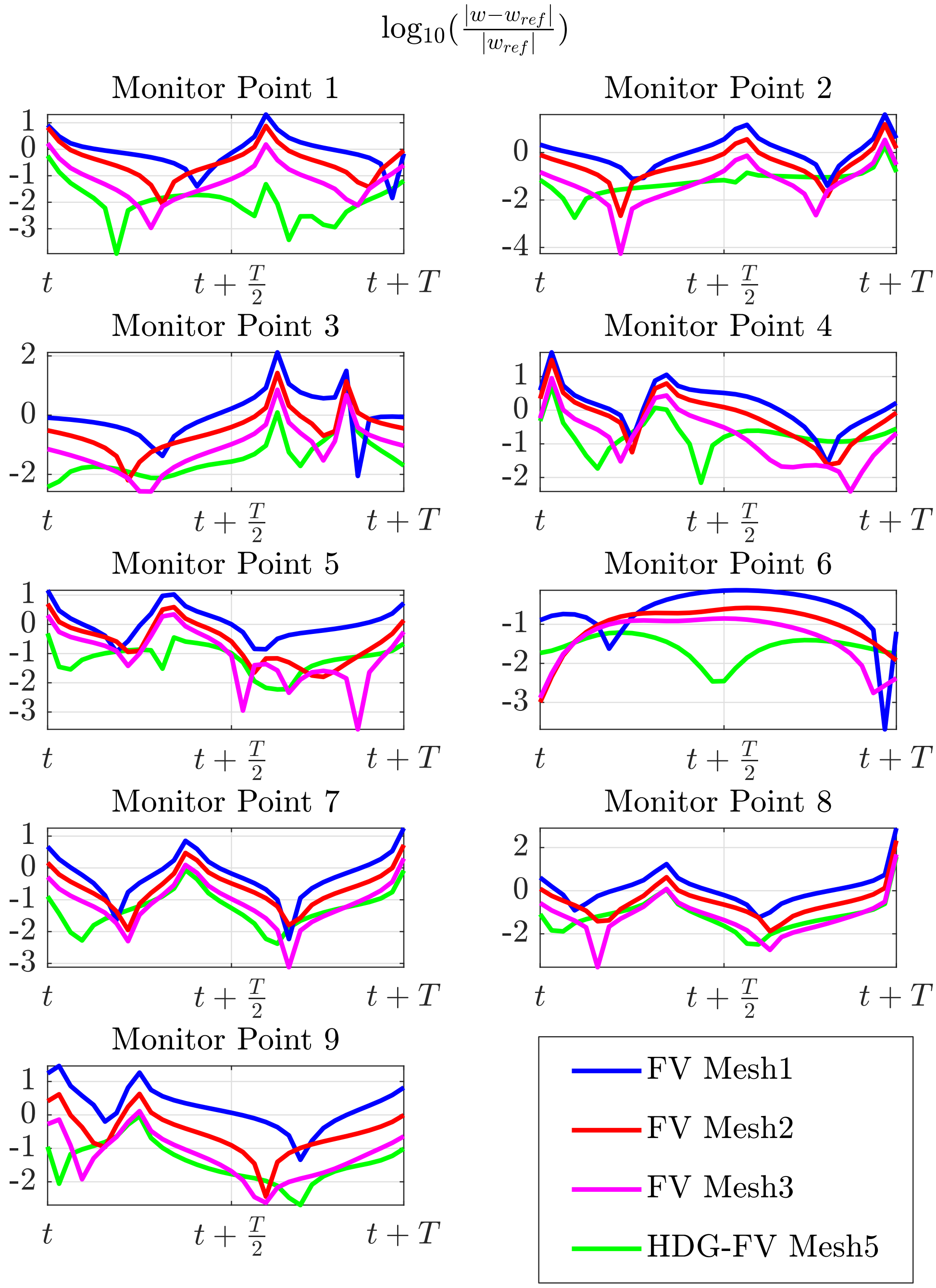


Figure. 6.27 Variation in the logarithmic scale of the normalised z-component of the velocity at various monitor points located downstream as depicted in Figure 6.16 of an ONERA M6 wing immersed in a sinusoidal gust flow.

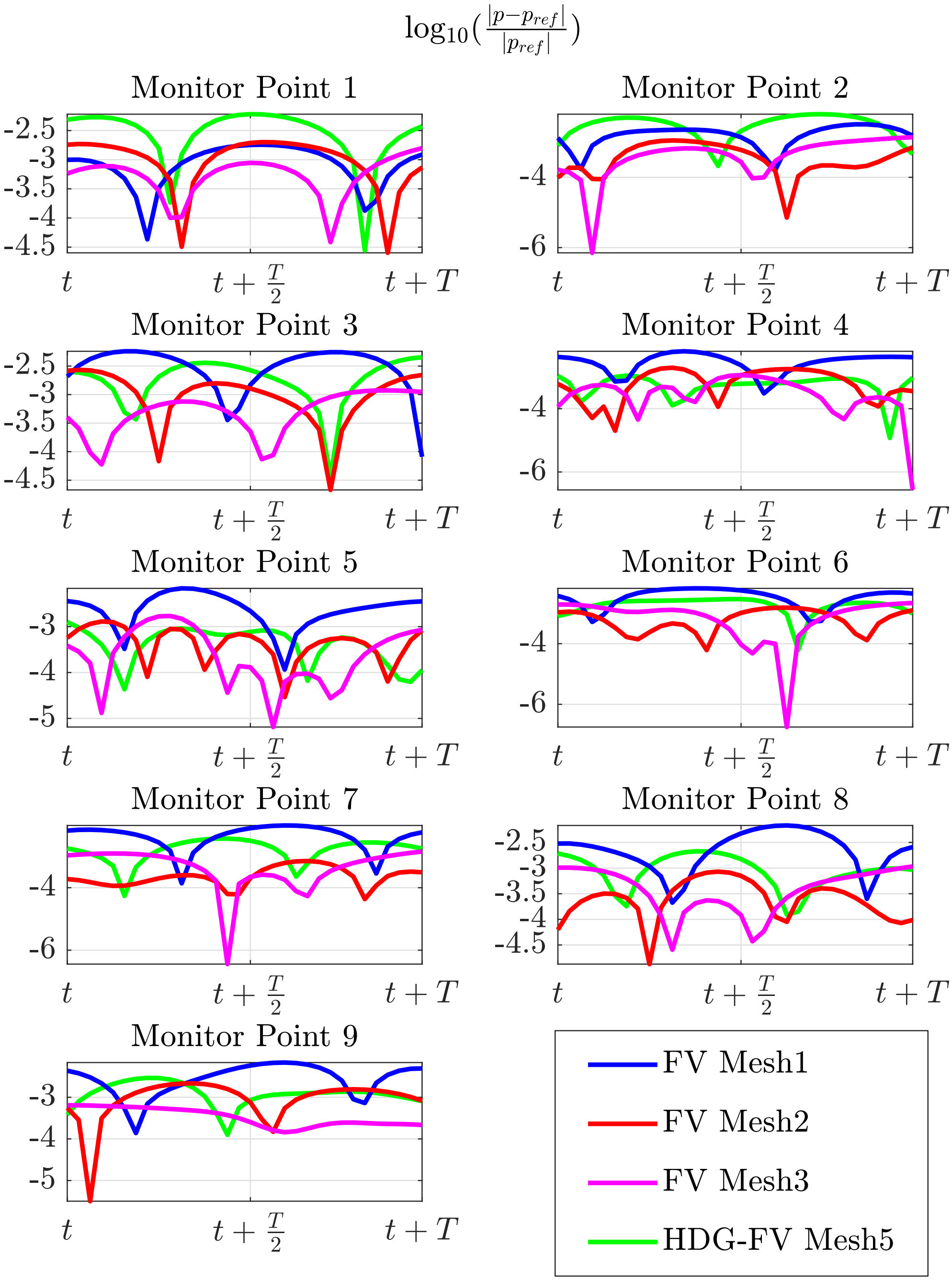


Figure. 6.28 Variation in the logarithmic scale of the normalised pressure at various monitor points located downstream as depicted in Figure 6.16 of an ONERA M6 wing immersed in a sinusoidal gust flow.

Chapter 7

Concluding remarks

"End? No, the journey doesn't end here."

Gandalf to Pippin

The present chapter concludes the thesis with a succinct recollection of the key ideas proposed and the outcomes achieved during the course of the current work. Also, a brief outlook along various research lines is presented for the progression and further development of the ideas proposed in this work.

7.1 Summary

The current work aims to address one of the vital needs of the CFD landscape, which is the development of fast, efficient and high-fidelity numerical solution workflow for unsteady flow problems.

The current techniques rely on low-order FV methods, which show large dissipation and dispersion errors on coarse meshes. They require a highly refined discretisation of the domain, resulting in very high computational costs due to the problem sizes. On the other hand, high-order methods provide with high-resolution solutions at computationally competitive costs in comparison to low-order methods.

In unsteady flows, the mitigation strategy to the large errors induced by the standard FV method is h-refinement. The computational domain needs to be discretised with suitably fine elements in order to capture the transient features in an adaptive manner. Apart from leading to large problem sizes, the generation of tailored meshes for transient flows is a non-trivial, problem-dependent and time-consuming process. Contrastingly, high-fidelity solutions are possible with high-order methods. These methods offer non-uniform order of approximations and degree-adaptivity, which greatly reduces the computational requirements and reduces the need for the modification of the mesh topology. However, close to the boundaries of the domain, high-order curvilinear meshes are necessary in-order to represent the geometry accurately. High-order meshing technologies have not yet reached maturity for adoption in the industrial workflows, specially for complex geometries in viscous simulations.

The present work is a proof-of-concept of the coupling of high-order and low-order methods for their favourable properties to improve the computational efficiency for transient phenomena. A novel coupling of second-order, industry standard FV and a new class of high-resolution schemes, the HDG method is set forth, wherein meshes routinely generated for steady-state simulations are repurposed for unsteady phenomena. It allows for efficient FV solution procedure on elements capable of accurately representing the solution with the low-order method and a high-fidelity HDG solution elsewhere. Also, the need to generate tailor-made fine grid or curvilinear meshes is circumvent. The following highlights the important achievements realised during the course of the present work.

- **Chapter 3:** A numerical formulation is developed for the time-dependent Euler equations with HDG spatial discretisation and BDF time integration. The HDG method with variable order of approximation on unstructured triangular and tetrahedral meshes is implemented. A shock capturing method to handle discontinuities in the flow is introduced. The resulting non-linear terms in the discrete equation for the residual are fully linearised using the Newton-Raphson method. The solution procedure for the linear system of equations is detailed. The numerical solution shows expected convergence behaviour for both 2D and 3D test cases using different polynomial orders of approximation.

- **Chapter 4:** A vertex-centred FV approach to the solution of transient Euler equations is implemented on 2D and 3D unstructured grids. An efficient edge-based formulation is developed. Second-order accurate, central difference type scheme with stabilisation through JST artificial dissipation is introduced. Implementations of additional shock-capturing, harmonic term activated by a pressure-based shock sensor is carried out. BDF implicit time-stepping up-to second order is performed on the discrete non-linear residual. Newton-Raphson linearisation is performed to obtain a linear system of equations. Numerical tests are performed and optimal convergence behaviour of the numerical solution is observed.

- **Chapter 5:** A novel combination of variable-order HDG and second-order FV methods is proposed. A set of transmission conditions across the HDG-FV interface, which imposes the continuity of the solution and the normal fluxes, is implemented. The resulting coupled scheme is solved in a monolithic manner with implicit BDF time stepping. Fast and parallel computations for the solution of the linear system of equations for both direct and iterative methods are realized through the use of external libraries. OpenMP and MPI programming models, for shared-memory and distributed-memory architectures respectively, are utilised in this context. Detailed information regarding the handling of the interface conditions is presented. Test cases are carried out depicting accurate resolution of discontinuities across the interface and optimal convergence behaviour of the coupled scheme.
- **Chapter 6:** Sinusoidal gust flow phenomena as the case study for simulating transient flow features is applied to various aerodynamic configurations. The advantages in terms of computational efficiency, both memory and the total CPU time required, is demonstrated for the coupled scheme on meshes designed for steady-state simulations as compared with the standard FV method on meshes customised for the unsteady phenomena. Significant savings are achieved when meshes for steady-state simulations are repurposed for transient processes. The partitioning of such meshes into HDG and FV subdomains based on element size is demonstrated. Also, assignment of the polynomial order of approximation based on element size and wavelength of the impinging gust is effective in obtaining an accurate solution. The procedure is shown to perform satisfactorily for 3D unstructured meshes as well.

7.2 Outlook

Further development of the proof-of-concept presented in the current work can be expanded along multiple aspects as highlighted below.

- **Extensions of the methodology** The next step would be to consider viscous effects for realistic flow simulations. The advantages demonstrated for the inviscid case will be carried over multiple-fold for the compressible Navier-Stokes equation with boundary layer meshes. Challenging flows such as simulating turbulence and unsteady flows with separation would become tenable with the proposed methodology. Additionally, the coupling can be applied to incompressible flow regimes. Applying Arbitrary Lagrangian-Eulerian (ALE) technique enables simulations with mesh motions to be performed. Furthermore, with multi-physics couplings, new areas of applications such as aero-acoustics and fluid-structure interaction studies become feasible.

- **Formulation improvements:** The coupled method would benefit from a degree-adaptive strategy for unsteady flows [54, 140]. Explorations into various time-integration strategies such as high-order and adaptive methodologies would be highly beneficial. An efficient edge-based strategy for the currently element-based HDG discretisation would lead to a performant formulation. The high-order requirement imposes severe restrictions in such a development since one requires information from the degrees of freedom in the element interiors. Advances in high-order serendipity elements would make an edge-based formulation tenable.

- **Algorithmic improvements:** Partitioning of HDG and FV subdomains can be performed including the influence of other metrics such as distance from aerodynamic configuration and CPU load balancing. A fast solution would result from effective parallel strategies utilising techniques such as multi-grid methods [17] and domain decomposition methodologies [2]. Solution to the sparse linear system of equations traditionally take up a majority of the CPU time in CFD simulations. In this regard, efforts for designing effective pre-conditioners for the iterative solution methods would be beneficial [131]. Improvements in solution process of several orders of magnitude can be realised with efficient register and cache management, pre-fetching, pipe-lining and instruction-level parallelism. With evolving computing hardware and architectures, considerations for effective use of heterogeneous computing accelerators is key to scalability in current and upcoming generations of HPC systems.

References

[1] Abgrall, R., Nordstrom, J., Offner, P., and Tokareva, S. (2020). Analysis of the SBP-SAT stabilization for finite element methods part i: Linear problems. *Journal of Scientific Computing*, 85(2).

[2] Andrea Toselli, O. W. (2004). *Domain Decomposition Methods - Algorithms and Theory*. Springer Berlin Heidelberg.

[3] Arnold, D. N. (1982). An interior penalty finite element method with discontinuous elements. *SIAM Journal on Numerical Analysis*, 19(4):742–760.

[4] Arnold, D. N., Brezzi, F., Cockburn, B., and Marini, D. (2000). Discontinuous galerkin methods for elliptic problems. In *Lecture Notes in Computational Science and Engineering*, pages 89–101. Springer Berlin Heidelberg.

[5] Arnold, D. N., Brezzi, F., Cockburn, B., and Marini, L. D. (2002). Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM Journal on Numerical Analysis*, 39(5):1749–1779.

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

[7] Atkins, H. L. and Shu, C.-W. (1998). Quadrature-free implementation of discontinuous galerkin method for hyperbolic equations. *AIAA Journal*, 36(5):775–782.

[8] Balay, S., Abhyankar, S., Adams, M. F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Dener, A., Eijkhout, V., Gropp, W. D., Karpeyev, D., Kaushik, D., Knepley, M. G., May, D. A., McInnes, L. C., Mills, R. T., Munson, T., Rupp, K., Sanan, P., Smith, B. F., Zampini, S., Zhang, H., and Zhang, H. (2019). PETSc Web page. https://www.mcs.anl.gov/petsc.

[9] Balay, S., Abhyankar, S., Adams, M. F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Dener, A., Eijkhout, V., Gropp, W. D., Karpeyev, D., Kaushik, D., Knepley, M. G., May, D. A., McInnes, L. C., Mills, R. T., Munson, T., Rupp, K., Sanan, P., Smith, B. F., Zampini, S., Zhang, H., and Zhang, H. (2020). PETSc users manual. Technical Report ANL-95/11 - Revision 3.14, Argonne National Laboratory.

[10] Balay, S., Gropp, W. D., McInnes, L. C., and Smith, B. F. (1997). Efficient management of parallelism in object oriented numerical software libraries. In Arge, E., Bruaset, A. M., and Langtangen, H. P., editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhäuser Press.

[11] Barter, G. E. and Darmofal, D. L. (2010). Shock capturing with PDE-based artificial viscosity for DGFEM: Part i. formulation. *Journal of Computational Physics*, 229(5):1810–1827.

[12] Barth, T. (1994). Aspects of unstructured grids and finitevolume solvers for the euler and navier-stokes equations, agard, special course on unstructured grid methods for advection dominated flows.

[13] Bassi, F. and Rebay, S. (1997a). A high-order accurate discontinuous finite element method for the numerical solution of the compressible navier–stokes equations. *Journal of Computational Physics*, 131(2):267–279.

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

[15] Benzoni-Gavage, S., Coulombel, J.-F., and Aubert, S. (2003). Boundary conditions for euler equations. *AIAA Journal*, 41(1):56–63.

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

[17] Briggs, W. (2000). *A multigrid tutorial.* Society for Industrial and Applied Mathematics, Philadelphia, PA.

[18] Campagne, G., Hassan, O., Morgan, K., and Sørensen, K. (2010). 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*, pages 309–325. Springer.

[19] Chalot, F. and Normand, P.-E. (2010). 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*, pages 145–165. Springer.

[20] Changfoot, D. M., Malan, A. G., and Nordström, J. (2018). Hybrid computational-fluid-dynamics platform to investigate aircraft trailing vortices. *Journal of Aircraft*, 56(1):344–355.

[21] Chawner, J. R., Dannenhoffer, J., Dey, S., Jones, W., Slotnick, J. P., and Taylor, N. J. (2015). The path to and state of geometry and meshing in 2030: Panel summary. In *22nd AIAA Computational Fluid Dynamics Conference*, page 3409.

[22] Ciarlet, P. G. (2002). *The finite element method for elliptic problems*, volume 40 of *Classics in Applied Mathematics.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA. Reprint of the 1978 original [North-Holland, Amsterdam].

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

[24] Cockburn, B., Dong, B., and Guzmán, J. (2008). A superconvergent LDG-hybridizable Galerkin method for second-order elliptic problems. *Mathematics of Computation*, 77(264):1887–1916.

[25] Cockburn, B. and Gopalakrishnan, J. (2004). A characterization of hybridized mixed methods for second order elliptic problems. *SIAM Journal on Numerical Analysis*, 42(1):283–301.

[26] Cockburn, B. and Gopalakrishnan, J. (2005a). Incompressible finite elements via hybridization. I. The Stokes system in two space dimensions. *SIAM Journal on Numerical Analysis*, 43(4):1627–1650.

[27] Cockburn, B. and Gopalakrishnan, J. (2005b). Incompressible finite elements via hybridization. II. The Stokes system in three space dimensions. *SIAM Journal on Numerical Analysis*, 43(4):1651–1672.

[28] Cockburn, B. and Gopalakrishnan, J. (2005c). New hybridization techniques. *GAMM-Mitt.*, 28(2):154–182.

[29] Cockburn, B., Gopalakrishnan, J., and Lazarov, R. (2009). Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems. *SIAM Journal on Numerical Analysis*, 47(2):1319–1365.

[30] Cockburn, B., Karniadakis, G. E., and Shu, C.-W. (2000). The development of discontinuous Galerkin methods. In *Discontinuous Galerkin methods (Newport, RI, 1999)*, volume 11 of *Lect. Notes Comput. Sci. Eng.*, pages 3–50. Springer, Berlin.

[31] Cockburn, B., Li, F., and Shu, C.-W. (2004). Locally divergence-free discontinuous galerkin methods for the maxwell equations. *Journal of Computational Physics*, 194(2):588–610.

[32] Cockburn, B., Sayas, F.-J., and Solano, M. (2012). Coupling at a distance HDG and BEM. *SIAM Journal on Scientific Computing*, 34(1):A28–A47.

[33] Cockburn, B. and Shu, C.-W. (1998). The runge–kutta discontinuous galerkin method for conservation laws v. *Journal of Computational Physics*, 141(2):199–224.

[34] Crumpton, P., Moinier, P., and Giles, M. (1998). An unstructured algorithm for high reynolds number flows on highly-stretched grids.

[35] Dadone, A. and Grossman, B. (1994). Surface boundary conditions for the numerical solution of the Euler equations. *AIAA journal*, 32(2):285–293.

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

[37] De Coninck, A., De Baets, B., Kourounis, D., Verbosio, F., Schenk, O., Maenhout, S., and Fostier, J. (2016). Needles: Toward large-scale genomic prediction with marker-by-environment interaction. 203(1):543–555.

[38] Dolejsi, V. and Feistauer, M. (2004). A semi-implicit discontinuous galerkin finite element method for the numerical solution of inviscid compressible flow. *Journal of Computational Physics*, 198(2):727–746.

[39] Donea, J. and Huerta, A. (2003). *Finite Element Methods for Flow Problems*. John Wiley & Sons.

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

[41] Engsig-Karup, A. P., Hesthaven, J. S., Bingham, H. B., and Madsen, P. A. (2006). Nodal DG-FEM solution of high-order boussinesq-type equations. *Journal of Engineering Mathematics*, 56(3):351–370.

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

[43] Ferrer, E. and Willden, R. H. (2012). A high order discontinuous Galerkin–Fourier incompressible 3D Navier–Stokes solver with rotating sliding meshes. *Journal of Computational Physics*, 231(21):7037–7056.

[44] Fix, G. M. (1976). Hybrid finite element methods. *SIAM Review*, 18(3):460–484.

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

[46] Gao, H., Wang, Z. J., and Liu, Y. (2010). A study of curved boundary representations for 2D high order Euler solvers. *Journal of Scientific Computing*, 44(3):323–336.

[47] Gargallo-Peiró, A., Roca, X., Peraire, J., and Sarrate, J. (2015). Distortion and quality measures for validating and generating high-order tetrahedral meshes. *Engineering with Computers*, 31(3):423–437.

[48] Georgoulis, E. H. (2005). Optimal error estimates for the hp-version interior penalty discontinuous galerkin finite element method. *IMA Journal of Numerical Analysis*, 25(1):205–220.

[49] Gerhold, T. (2005). Overview of the hybrid RANS code TAU. In *MEGAFLOW-Numerical Flow Simulation for Aircraft Design*, pages 81–92. Springer.

[50] Giacomini, M. and Sevilla, R. (2019). Discontinuous galerkin approximations in computational mechanics: hybridization, exact geometry and degree adaptivity. *SN Applied Sciences*, 1(9).

[51] Giacomini, M., Sevilla, R., and Huerta, A. (2020a). HDGlab: An open-source implementation of the hybridisable discontinuous galerkin method in MATLAB. *Archives of Computational Methods in Engineering*.

[52] Giacomini, M., Sevilla, R., and Huerta, A. (2020b). Tutorial on hybridizable discontinuous galerkin (HDG) formulation for incompressible flow problems. In *Modeling in Engineering Using Innovative Numerical Methods for Solids and Fluids*, pages 163–201. Springer International Publishing.

[53] Giorgiani, G., Fernández-Méndez, S., and Huerta, A. (2013a). Hybridizable discontinuous Galerkin p-adaptivity for wave propagation problems. *International Journal for Numerical Methods in Fluids*, 72(12):1244–1262.

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

[55] Giorgiani, G., Modesto, D., Fernández-Méndez, S., and Huerta, A. (2013b). High-order continuous and discontinuous galerkin methods for wave problems. *International Journal for Numerical Methods in Fluids*, pages n/a–n/a.

[56] Glasby, R., Burgess, N., Anderson, K., Wang, L., Allmaras, S., and Mavriplis, D. (2013). 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*, page 691.

[57] Golubev, V., Dreyer, B., Hollenshade, T., and Visbal, M. (2009). High-accuracy viscous simulation of gust-airfoil nonlinear aeroelastic interaction. In *39th aiaa fluid dynamics conference*, page 4200.

[58] Golubev, V. V., Mankbadi, R. R., and Hixon, R. (2005). Space-time mapping analysis of airfoil nonlinear interaction with unsteady inviscid flow. *AIAA Journal*, 43(10):2147–2156.

[59] Hairer, E. and Wanner, G. (1996). *Solving Ordinary Differential Equations II*. Springer Berlin Heidelberg.

[60] Harten, A., Osher, S., Engquist, B., and Chakravarthy, S. R. (1986). Some results on uniformly high-order accurate essentially nonoscillatory schemes. *Applied Numerical Mathematics*, 2(3-5):347–377.

[61] Hartmann, R. (2002). Adaptive finite element methods for the compressible euler equations.

[62] Hesthaven, J. and Warburton, T. (2002). Nodal high-order methods on unstructured grids. *Journal of Computational Physics*, 181(1):186–221.

[63] Hesthaven, J. S. and Warburton, T. (2008). *Nodal Discontinuous Galerkin Methods*. Springer-Verlag GmbH.

[64] Hirsch, C. (1988). *Numerical computation of internal and external flows. Volume 1, Fundamentals of numerical discretization*. Wiley series in numerical methods in engineering. John Wiley & sons, Chichester.

[65] Hirsch, C. (1990). *Numerical computation of internal and external flows. Volume 2, Computational Methods for Inviscid and Viscous Flows.* Wiley series in numerical methods in engineering. John Wiley & sons, Chichester.

[66] Hirsch, C. (2007). *Numerical Computation of Internal and External Flows, Volume 1.* Elsevier LTD, Oxford.

[67] Hixon, R., Golubev, V., Mankbadi, R. R., Scott, J. R., Sawyer, S., and Nallasamy, M. (2006). Application of a nonlinear computational aeroacoustics code to the gust-airfoil problem. *AIAA Journal*, 44(2):323–328.

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

[69] Huerta, A., Angeloski, A., Roca, X., and Peraire, J. (2013). Efficiency of high-order elements for continuous and discontinuous Galerkin methods. *International Journal for Numerical Methods in Engineering*, 96(9):529–560.

[70] Huntley, S. J., Jones, D., and Gaitonde, A. (2016). 2d and 3d gust response using a prescribed velocity method in viscous flows. In *46th AIAA Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics.

[71] Huynh, H., Wang, Z., and Vincent, P. (2014). High-order methods for computational fluid dynamics: A brief review of compact differential formulations on unstructured grids. *Computers & Fluids*, 98:209–220.

[72] Jameson, A. (1991). Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. In *10th Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics.

[73] Jameson, A. (1995a). Analysis and design of numerical schemes for gas dynamics, 1: artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence. *International Journal of Computational Fluid Dynamics*, 4(3-4):171–218.

[74] Jameson, A. (1995b). Analysis and design of numerical schemes for gas dynamics, 2: artificial diffusion and discrete shock structure. *International Journal of Computational Fluid Dynamics*, 5(1-2):1–38.

[75] Jameson, A., Schmidt, W., and Turkel, E. (1981). Numerical solution of the Euler equations by finite volume methods using Runge Kutta time stepping schemes. In *14th fluid and plasma dynamics conference*, page 1259.

[76] Jaust, A. and Schuetz, J. (2014). A temporally adaptive hybridized discontinuous galerkin method for time-dependent compressible flows. *Computers & Fluids*, 98:177–185.

[77] Jaust, A., Schuetz, J., and Woopen, M. (2014). A hybridized discontinuous galerkin method for unsteady flows with shock-capturing. In *44th AIAA Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics.

[78] Johnson, F. T., Tinoco, E. N., and Yu, N. J. (2005). Thirty years of development and application of CFD at boeing commercial airplanes, seattle. *Computers & Fluids*, 34(10):1115–1151.

[79] Karman, S. L., Wyman, N., and Steinbrenner, J. P. (2017). Mesh generation challenges: A commercial software perspective. In *23rd AIAA Computational Fluid Dynamics Conference*, page 3790.

[80] Kourounis, D., Fuchs, A., and Schenk, O. (2018). Towards the next generation of multiperiod optimal power flow solvers. *IEEE Transactions on Power Systems*, PP(99):1–10.

[81] Krivodonova, L. and Berger, M. (2006). High-order accurate implementation of solid wall boundary conditions in curved geometries. *Journal of computational physics*, 211(2):492–512.

[82] Krivodonova, L., Xin, J., Remacle, J.-F., Chevaugeon, N., and Flaherty, J. (2004). Shock detection and limiting with discontinuous galerkin methods for hyperbolic conservation laws. *Applied Numerical Mathematics*, 48(3-4):323–338.

[83] Kroll, N., Bieler, H., Deconinck, H., Couaillier, V., van der Ven, H., and Sorensen, K. (2010). *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.

[84] Kroll, N., Hirsch, C., Bassi, F., Johnston, C., and Hillewaert, K. (2015). *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.

[85] Laflin, K. R., Klausmeyer, S. M., Zickuhr, T., Vassberg, J. C., Wahls, R. A., Morrison, J. H., Brodersen, O. P., Rakowitz, M. E., Tinoco, E. N., and Godard, J.-L. (2005). Data summary from second AIAA computational fluid dynamics drag prediction workshop. *Journal of Aircraft*, 42(5):1165–1178.

[86] Laney, C. B. (1998). *Computational Gasdynamics*. Cambridge University Press.

[87] Lele, S. K. (1992). Compact finite difference schemes with spectral-like resolution. *Journal of Computational Physics*, 103(1):16–42.

[88] LeVeque, R. J. (2002a). *Finite volume methods for hyperbolic problems*, volume 31. Cambridge university press.

[89] LeVeque, R. J. (2002b). *Finite volume methods for hyperbolic problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press, Cambridge.

[90] LeVeque, R. J. and Leveque, R. J. (1992). *Numerical methods for conservation laws*, volume 132. Springer.

[91] Levy, D., Laflin, K., Vassberg, J., Tinoco, E., Mani, M., Rider, B., Brodersen, O., Crippa, S., Rumsey, C., Wahls, R., Morrison, J., Mavriplis, D., and Murayama, M. (2013). Summary of data from the fifth AIAA CFD drag prediction workshop. In *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. American Institute of Aeronautics and Astronautics.

[92] Levy, D. W., Zickuhr, T., Vassberg, J., Agrawal, S., Wahls, R. A., Pirzadeh, S., and Hemsch, M. J. (2003). Data summary from the first AIAA computational fluid dynamics drag prediction workshop. *Journal of Aircraft*, 40(5):875–882.

[93] Li, L., Lanteri, S., and Perrussel, R. (2014). A hybridizable discontinuous galerkin method combined to a schwarz algorithm for the solution of 3d time-harmonic maxwell's equation. *Journal of Computational Physics*, 256:563–581.

[94] Liu, J. W. (1992). The multifrontal method for sparse matrix solution: Theory and practice. *SIAM review*, 34(1):82–109.

[95] Liu, X.-D., Osher, S., and Chan, T. (1994). Weighted essentially non-oscillatory schemes. *Journal of Computational Physics*, 115(1):200–212.

[96] Lockard, D. P. and Morris, P. J. (1998). Radiated noise from airfoils in realistic mean flows. *AIAA Journal*, 36(6):907–914.

[97] Löwe, J., Probst, A., Knopp, T., and Kessler, R. (2015). A low-dissipation low-dispersion second-order scheme for unstructured finite-volume flow solvers. In *53rd AIAA Aerospace Sciences Meeting*, page 0815.

[98] Luo, X.-J., Shephard, M. S., O'bara, R. M., Nastasia, R., and Beall, M. W. (2004). Automatic *p*-version mesh generation for curved domains. *Engineering with Computers*, 20(3):273–285.

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

[100] Mengaldo, G., Grazia, D. D., Witherden, F., Farrington, A., Vincent, P., Sherwin, S., and Peiro, J. (2014). A guide to the implementation of boundary conditions in compact high-order methods for compressible aerodynamics. In *7th AIAA Theoretical Fluid Mechanics Conference*. American Institute of Aeronautics and Astronautics.

[101] Montlaur, A., Fernández-Méndez, S., and Huerta, A. (2008). Discontinuous Galerkin methods for the Stokes equations using divergence-free approximations. *International Journal for Numerical Methods in Fluids*, 57(9):1071–1092.

[102] Morgan, K. and Peraire, J. (1998). Unstructured grid finite-element methods for fluid mechanics. *Reports on Progress in Physics*, 61(6):569.

[103] Morgan, K., Peraire, J., Peiro, J., and Hassan, O. (1991). The computation of three-dimensional flows using unstructured grids. *Computer Methods in Applied Mechanics and Engineering*, 87(2-3):335–352.

[104] Moro, D., Nguyen, N. C., and Peraire, J. (2016). Dilation-based shock capturing for high-order methods. *International Journal for Numerical Methods in Fluids*, 82(7):398–416.

[105] Moukalled, F., Mangani, L., and Darwish, M. (2015). *The Finite Volume Method in Computational Fluid Dynamics*. Springer-Verlag GmbH.

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

[107] Nguyen, C. and Peraire, J. (2011). An adaptive shock-capturing HDG method for compressible flows. In *20th AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics.

[108] Nguyen, N., Peraire, J., and Cockburn, B. (2010a). A hybridizable discontinuous Galerkin method for Stokes flow. *Computer Methods in Applied Mechanics and Engineering*, 199(9-12):582–597.

[109] Nguyen, N., Peraire, J., and Cockburn, B. (2010b). A hybridizable discontinuous galerkin method for the incompressible navier-stokes equations. In *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*. American Institute of Aeronautics and Astronautics.

[110] Nguyen, N. C., Peraire, J., and Cockburn, B. (2009a). An implicit high-order hybridizable discontinuous Galerkin method for linear convection-diffusion equations. *Journal of Computational Physics*, 228(9):3232–3254.

[111] Nguyen, N. C., Peraire, J., and Cockburn, B. (2009b). An implicit high-order hybridizable discontinuous Galerkin method for nonlinear convection-diffusion equations. *Journal of Computational Physics*, 228(23):8841–8855.

[112] Nguyen, N. C., Peraire, J., and Cockburn, B. (2011). An implicit high-order hybridizable discontinuous Galerkin method for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, 230(4):1147–1170.

[113] Nitsche, J. (1971). Uber ein variationsprinzip zur losung von dirichlet-problemen bei verwendung von teilraumen, die keinen randbedingungen unterworfen sind. *Abhandlungen aus dem Mathematischen Seminar der Universitat Hamburg*, 36(1):9–15.

[114] Nogueira, X., Colominas, I., Cueto-Felgueroso, L., and Khelladi, S. (2010). 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(23-24):1471–1490.

[115] Paipuri, M., Tiago, C., and Fernández-Méndez, S. (2019). Coupling of continuous and hybridizable discontinuous Galerkin methods: Application to conjugate heat transfer problem. *Journal of Scientific Computing*, 78(1):321–350.

[116] Pan, D. and Cheng, J.-C. (1995). Unstructured Euler flutter analysis of two-dimensional wing-tail configuration. *Journal of Aircraft*, 32(5):1152–1155.

[117] Patankar, S. (1980). *Numerical heat transfer and fluid flow.* CRC press.

[118] Patera, A. T. (1984). A spectral element method for fluid dynamics: Laminar flow in a channel expansion. *Journal of Computational Physics*, 54(3):468–488.

[119] Peraire, J., Nguyen, C., and Cockburn, B. (2011). An embedded discontinuous galerkin method for the compressible euler and navier-stokes equations. In *20th AIAA Computational Fluid Dynamics Conference*, page 3228.

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

[121] Peraire, J. and Persson, P.-O. (2008). The compact discontinuous galerkin (CDG) method for elliptic problems. *SIAM Journal on Scientific Computing*, 30(4):1806–1824.

[122] Persson, P.-O. and Peraire, J. (2006). Sub-cell shock capturing for discontinuous Galerkin methods. In *44th AIAA Aerospace Sciences Meeting and Exhibit*, pages 112–124.

[123] Persson, P.-O. and Peraire, J. (2009). Curved mesh generation and mesh refinement using lagrangian solid mechanics. In *Proceedings of the 47th AIAA Aerospace Sciences Meeting and Exhibit.* AIAA.

[124] Poinsot, T. and Lelef, S. (1992). Boundary conditions for direct simulations of compressible viscous flows. *Journal of Computational Physics*, 101(1):104–129.

[125] Reed, W. H. and Hill, T. R. (1973). Triangular mesh methods for the neutron transport equation.

[126] Reports, T. P., Rumsey, C. L., Poirier, D. M. A., Bush, R. H., and Towne, C. E. (2001). Nasa/tm-2001-211236 a users guide to cgns.

[127] Rhebergen, S. and Wells, G. N. (2018). A hybridizable discontinuous Galerkin method for the Navier–Stokes equations with pointwise divergence-free velocity field. *Journal of Scientific Computing*, 76(3):1484–1501.

[128] Rivière, B., Wheeler, M. F., and Girault, V. (2001). A priori error estimates for finite element methods based on discontinuous approximation spaces for elliptic problems. *SIAM Journal on Numerical Analysis*, 39(3):902–931.

[129] Rohde, A. (2001). Eigenvalues and eigenvectors of the euler equations in general geometries. In *15th AIAA Computational Fluid Dynamics Conference.* American Institute of Aeronautics and Astronautics.

[130] Roy, C. J., Rumsey, C. L., and Tinoco, E. N. (2018). Summary data from the sixth AIAA computational fluid dynamics drag prediction workshop: Code verification. *Journal of Aircraft*, 55(4):1338–1351.

[131] Saad, Y. (2003). *Iterative methods for sparse linear systems.* Society for Industrial and Applied Mathematics.

[132] Schmitt, V. and Charpin, F. (1979). Pressure distributions on the onera m6 wing at transonic mach numbers. In *Experimental Data Base for Computer Program Assessment*. Report of the Fluid Dynamics Panel Working Group 04, AGARD AR 138.

[133] Sevilla, R. (2019). HDG-NEFEM for two dimensional linear elasticity. *Computers & Structures*, 220:69–80.

[134] Sevilla, R., Fernández-Méndez, S., and Huerta, A. (2008). NURBS-enhanced finite element method (NEFEM) for Euler equations. *International Journal for Numerical Methods in Fluids*, 57(9):1051–1069.

[135] Sevilla, R., Fernández−Méndez, S., and Huerta, A. (2011a). Comparison of high-order curved finite elements. *International Journal for Numerical Methods in Engineering*, 87(8):719–734.

[136] Sevilla, R., Fernández−Méndez, S., and Huerta, A. (2011b). NURBS–Enhanced Finite Element Method (NEFEM): a seamless bridge between CAD and FEM. *Archives of Computational Methods in Engineering*, 18(4):441–484.

[137] Sevilla, R., Giacomini, M., and Huerta, A. (2018). A face-centred finite volume method for second-order elliptic problems. *International Journal for Numerical Methods in Engineering*, 115(8):986–1014.

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

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

[140] Sevilla, R. and Huerta, A. (2018). HDG-NEFEM with degree adaptivity for stokes flows. *Journal of Scientific Computing*, 77(3):1953–1980.

[141] Slotnick, J., Khodadoust, A., Alonso, J., Darmofal, D., Gropp, W., Lurie, E., and Mavriplis, D. (2014). Cfd vision 2030 study: A path to revolutionary computational aerosciences. NASA/CR-2014-218178.

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

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

[144] Sørensen, K., Hassan, O., Morgan, K., and Weatherill, N. (2003a). A multigrid accelerated hybrid unstructured mesh method for 3D compressible turbulent flow. *Computational Mechanics*, 31(1-2):101–114.

[145] Sørensen, K., Hassan, O., Morgan, K., and Weatherill, N. (2003b). 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(5):517–536.

[146] Sørensen, K. A., Hassan, O., Morgan, K., and Weatherill, N. P. (2002). Agglomerated multigrid on hybrid unstructured meshes for compressible flow. *International Journal for Numerical Methods in Fluids*, 40(3-4):593–603.

[147] Spalart, P. R. and Venkatakrishnan, V. (2016). On the role and challenges of CFD in the aerospace industry. *The Aeronautical Journal*, 120(1223):209–232.

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

[149] Strand, B. (1994). Summation by parts for finite difference approximations for d/dx. *Journal of Computational Physics*, 110(1):47–67.

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

[151] Vassberg, J., Tinoco, E., Mani, M., Brodersen, O., Eisfeld, B., Wahls, R., Morrison, J., Zickuhr, T., Laflin, K., and Mavriplis, D. (2007). Summary of the third AIAA CFD drag prediction workshop. In *45th AIAA Aerospace Sciences Meeting and Exhibit*. American Institute of Aeronautics and Astronautics.

[152] Vassberg, J., Tinoco, E., Mani, M., Rider, B., Zickuhr, T., Levy, D., Brodersen, O., Eisfeld, B., Crippa, S., Wahls, R., Morrison, J., Mavriplis, D., and Murayama, M. (2010). Summary of the fourth AIAA CFD drag prediction workshop. In *28th AIAA Applied Aerodynamics Conference*. American Institute of Aeronautics and Astronautics.

[153] Verbosio, F., Coninck, A. D., Kourounis, D., and Schenk, O. (2017). Enhancing the scalability of selected inversion factorization algorithms in genomic prediction. *Journal of Computational Science*, 22(Supplement C):99 – 108.

[154] Versteeg, H. K. and Malalasekera, W. (2007). *An introduction to computational fluid dynamics: the finite volume method*. Pearson education.

[155] Vila-Perez, J., Giacomini, M., Sevilla, R., and Huerta, A. (2020). Hybridisable discontinuous galerkin formulation of compressible flows.

[156] Vincent, P. E., Castonguay, P., and Jameson, A. (2010). A new class of high-order energy stable flux reconstruction schemes. *Journal of Scientific Computing*, 47(1):50–72.

[157] Vincent, P. E. and Jameson, A. (2011). Facilitating the adoption of unstructured high-order methods amongst a wider community of fluid dynamicists. *Mathematical Modelling of Natural Phenomena*, 6(3):97–140.

[158] VonNeumann, J. and Richtmyer, R. D. (1950). A method for the numerical calculation of hydrodynamic shocks. *Journal of Applied Physics*, 21(3):232–237.

[159] Wang, Z., Fidkowski, K., Abgrall, R., Bassi, F., Caraeni, D., Cary, A., Deconinck, H., Hartmann, R., Hillewaert, K., Huynh, H., Kroll, N., May, G., Persson, P.-O., van Leer, B., and Visbal, M. (2013). High-order CFD methods: current status and perspective. *International Journal for Numerical Methods in Fluids*, 72(8):811–845.

[160] Wang, Z. and Gao, H. (2009). A unifying lifting collocation penalty formulation for the euler equations on mixed grids. In *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*. American Institute of Aeronautics and Astronautics.

[161] Weatherill, N. P., Hassan, O., Morgan, K., Jones, J. W., Larwood, B. G., and Sorenson, K. (2002). Aerospace simulations on parallel computers using unstructured grids. *International Journal for Numerical Methods in Fluids*, 40(1-2):171–187.

[162] White, F. (2003). *Fluid mechanics.* McGraw-Hill, Boston.

[163] Whiting, C. H. and Jansen, K. E. (2001). A stabilized finite element method for the incompressible Navier–Stokes equations using a hierarchical basis. *International Journal for Numerical Methods in Fluids*, 35(1):93–116.

[164] Witherden, F. D. and Vincent, P. E. (2015). On the identification of symmetric quadrature rules for finite element methods. *Comput. Math. Appl.*, 69(10):1232–1241.

[165] Woopen, M. and May, G. (2015). An anisotropic adjoint-based *hp*-adaptive HDG method for compressible turbulent flow. In *53rd AIAA Aerospace Sciences Meeting*, page 2042.

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

[167] Xue, D., Demkowicz, L., et al. (2005). 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(3):283–300.

[168] Yakovlev, S., Moxey, D., Kirby, R. M., and Sherwin, S. J. (2016). To CG or to HDG: a comparative study in 3d. *Journal of Scientific Computing*, 67(1):192–220.

[169] Zhang, Y.-T. and Shu, C.-W. (2016). ENO and WENO schemes. In *Handbook of Numerical Analysis*, pages 103–122. Elsevier.

[170] Zienkiewicz, O. and Taylor, R. (2000). *The Finite Element Method: The basis.* Fluid Dynamics. Butterworth-Heinemann.