High-order enforcement of jumps conditions between compressible viscous phases: an extended interior penalty discontinuous Galerkin method for sharp interface simulation

Preprint · July 2023

DOI: 10.1016/j.cma.2023.116215

4 authors:

David Henneaux
Université Catholique de Louvain - UCLouvain
5 PUBLICATIONS **8** CITATIONS
SEE PROFILE

Pierre Schrooyen
Cenaero
15 PUBLICATIONS **70** CITATIONS
SEE PROFILE

Philippe Chatelain
Université Catholique de Louvain - UCLouvain
169 PUBLICATIONS **2,041** CITATIONS
SEE PROFILE

Thierry Magin
von Karman Institute for Fluid Dynamics
262 PUBLICATIONS **3,566** CITATIONS
SEE PROFILE

High-order enforcement of jumps conditions between compressible viscous phases: an extended interior penalty discontinuous Galerkin method for sharp interface simulation

David Henneaux[a,c,*], Pierre Schrooyen[b], Philippe Chatelain[c], Thierry Magin[a,d]

[a] *von Karman Institute for Fluid Dynamics, Aeronautics and Aerospace Department, Waterloosesteenweg 72, B-1640, Sint-Genesius-Rode, Belgium*
[b] *Cenaero, Rue des Freres Wright 29, B-6041, Gosselies, Belgium*
[c] *Universite catholique de Louvain, Institute of Mechanical, Materials and Civil Engineering, Place du Levant 2, B-1348, Louvain-la-Neuve, Belgium*
[d] *Universite Libre de Bruxelles, Service d'Aero-Thermo-Mecanique, 50 Avenue F.D. Roosevelt, B-1050, Bruxelles, Belgium*

Abstract

In this paper, we develop a high-order method for the steady two-phase compressible Navier-Stokes equations closed by generic equations of state, adapted to gas-liquid flows. Our framework relies on the extended discontinuous Galerkin method that is tailored here to a general viscous compressible two-phase configuration. While the imposition of interface jumps conditions is commonly recognised as one of the main difficulties to overcome in the context of geometrically unfitted methods, the setting considered in this work makes this aspect even more challenging. To enforce the non-linear coupling conditions at the interface in a sharp manner, we devise a novel strategy combining multiphase Riemann solvers to handle the convective fluxes across the interface and a weighted stabilized Nitsche's method for the viscous jumps. The weak enforcement of the jumps conditions along with implicit steady-

*Corresponding author
Email addresses: david.henneaux@vki.ac.be (David Henneaux), pierre.schrooyen@cenaero.be (Pierre Schrooyen), philippe.chatelain@uclouvain.be (Philippe Chatelain), thierry.magin@vki.ac.be (Thierry Magin)

state iterations and cell-agglomeration procedure make the approach robust to treat arbitrary large contrast phase interface problems governed by stiff models, irrespective of the interface location on the mesh. The study is here restricted to static interfaces. Based on the method of manufactured solution, reference solutions are designed. This allows to conduct a detailed numerical study investigating the influence of several numerical parameters. It is shown that to reach optimal error convergence, the use of pressure-velocity-temperature variables set, appropriate all-speed bulk Riemann solver and Symmetric Interior Penalty method combined with tensorial penalty is necessary.

Keywords: two-phase flow, viscous compressible, sharp interface, extended, unfitted, interior penalty discontinuous Galerkin

---

1. Introduction

1.1. Context and challenges

Many of the relevant phenomena to study involve the mutual interactions between different phases[1], from the heat conduction between two solids with different thermal conductivities to the deformation of a solid structure by an impacting fluid flow or the spraying of a liquid jet in a surrounding gas. Such problems are usually casted as a system of coupled partial differential equations (PDEs). Often the PDEs are posed on disjoint sub-domains separated by evolving interfaces across which the bulk solutions are glued together by interface conditions. Compared to the simulation of isolated single phase behavior, three important challenges associated to the numerical simulation of such physical problems need to be faced: (i) the interface dynamics, (ii) the discontinuities of the fields across the interface, and (iii) the coupling between the phases. The present work focuses on the two latter aspects, i.e. the discretization of the discontinuous fields and the enforcement of the interface coupling conditions, in the field of separated immiscible two-phase compressible flows. This field is effervescent in the way of numerically dealing with those two difficulties. On the one hand, in the diffuse interface approach, the numerous multi-fluid methods available have in common that

[1] "Phase" is here to be taken in the most general sense: two different phases can refer to two different chemical substances (air and water e.g.) or to the same chemical substance with two different phases of matter (steam and liquid water e.g.).

they solve a single set of equations over the whole domain derived from averaging procedures of the one-fluid Euler or Navier-Stokes equations. Thanks to the averaging (usually a volume average), fluid variables exhibit a smooth transition in a mixture region around the interface. The jumps conditions are fulfilled within the artificial mixture zone through different relaxation operators. Recent reviews about this vast area of multi-fluid models and methods can be found in [1, 2]. On the other hand, in the sharp interface community, most of the existing methods for compressible two-phase flows belong to the Ghost Fluid Methods (GFM) [3–7], the conservative cut-cell method [8–13] or the Arbitrary Lagrangian-Eulerian (ALE) framework [14–18]. All of them are both used with Finite Volume (FVM) and Finite Element (FEM) type discretizations. In this work, we rely instead on a high-order fixed-mesh geometrically unfitted discontinuous Galerkin (DG) FEM approach. Although FE-type cut-cell methods also allow an interface to freely cut across the underlying mesh, they still require local cut-cell reconstruction, which distinguishes them from the proposed approach which do not need any mesh manipulations by combining high-order interface representation and high-order quadrature rules on implicitly-defined domains. We borrow some ideas from the GFM by extending the approximation space on cut elements to be able to represent sharp discontinuities and by using multiphase Riemann solvers to handle the interface convective fluxes. Contrary to the standard GFM, a weak imposition of the jumps conditions is here followed. The review of the state-of-the-art in section 1.2 is directed on this class of methods which adopt an unfitted discretization on a fixed mesh with both the governing equations and the interface discretized by a high-order FE-type method. In this specific scope, a lot of work carried out so far has been concentrated on model problems like scalar hyperbolic or elliptic equations. In the context of two-phase flows, the focus has been put on the incompressible equations. The originality of this work is to extend and apply the concept of geometrically unfitted FE discretization to the sharp interface treatment of viscous compressible two-phase flows. We believe that this class of flows can greatly benefit from the unique features offered by unfitted FE discretizations, which are put into perspective in relation to the previously mentioned methods in section 1.3.

1.2. Previous work about FE-type geometrically unfitted methods

Several denominations can be found in the literature to identify one or the other geometry independent FE discretization. Here the term “geomet-

rically unfitted FEM" embraces all of those "extended/ cut-cell/ unfitted/ embedded/ immersed" methods. The common idea of the geometrically unfitted FE-type methods is to embed a complex geometry into a geometrically simple domain that can easily be meshed. To take advantage of the potential benefits offered by these methods, three computational challenges in comparison to standard mesh-fitting FEM must be overcome [19], [20, Ch. 1.2.1].
The first one is about the critical intersections between the interface and the mesh. Because the mesh is not fitted to the interface, unfortunate cuts resulting in elements with very small intersections or so-called sliver elements can cause stability and conditioning issues and sometimes affect accuracy. For FEM, the most widely used strategy to cope with those issues all at once is the ghost-penalty method [21]: weakly consistent stabilization operators are added in the interface region to control the solution outside of the physical domain. Recent geometrically unfitted FEM resorting to this stabilisation strategy are usually referred as *CutFEM* [22]. Instead of altering the variational form, the recently proposed aggregated FEM (*AgFEM*) [23] modifies the finite element space in troubled cells by constructing a discrete extension operator between well-posed and ill-posed degrees of freedom. The main difficulty is to build this operator in such a way that the continuity between adjacent elements is maintained. In the context of DGM, this kind of extensions is very natural as the inter-element continuity is enforced only weakly, thus each local basis function can be chosen independently. This has lead to the development of the unfitted DGM (*UDG*) [24, 25] in which problematic cut cells are removed from the discretization and merged with incident ones. Similarly, the eXtended DGM (*XDGM*) uses cell-agglomeration by injecting the solution on the agglomerated basis thanks to cell-local extension operations [26, 27]. Ghost-penalty stabilisation has also been applied to DGM under the name of *CutDG* [28]. Worth mentioning is the Shifted Boundary Method (*SBM*) [29, 30] which consists in imposing on a surrogate boundary an accurately modified boundary condition such that the small cut-cell issue is avoided.
The second challenge is related to the integration on cut-cells. Indeed, evaluating integrals on arbitrary shaped cut elements defined by surfaces that do not have an explicit parametrization requires dedicated quadrature techniques. The major challenge is to develop schemes which are robust in their construction, ensure a sufficiently accurate approximation and at the same time minimize the number of quadrature points to retain computational efficiency. There exist several major groups of methods to perform numerical

integration over implictly-defined geometries; an overview of some them can be found in [31]. The choice of one method or another will determine the position of the quadrature points (strictly inside the domain of integration or not) and the quadrature weights (strictly positive or not).
The last challenge has to do with the imposition of the interface jumps. To preserve the stability of the standard FEM or DGM and thus ensure well-posedness of the final discretization, the imposition of the jumps conditions on the unfitted interface must be carefully carried out, both for diffusive and convective effects. Regarding the diffusive effects for FEM, the unfitted discretization precludes the strong imposition of Dirichlet boundary conditions as in the boundary-fitted case. A weak imposition of the interface jumps is therefore usually followed, which requires special measures to control induced instabilities. Among the different strategies available (see e.g. [32] for an overview), Nitsche's method is very appropriate for interface conditions imposition [33] because it is easy to implement and it ensures fundamental properties like inf-sup stability and optimal *a priori* estimates. The dependence of the stability parameter to cut cells scenarios appearing in Nitsche's method can be accommodated with the techniques described above. Likewise, UDGM [25], XDGM [26] and CutDGM [28] also use penalty interface terms to impose jumps conditions related to elliptic operators. There are also work extending Nitsche's technique to Robin-type interface conditions, in the context of CutFEM [34] and DGM [35]. With regard to the convective effects, interface instabilities which emanate from the non-linear convective terms need also to be dealt with. In FEM, they can be balanced by consistently adding stabilising terms allowing to control the inflow from one phase to the other depending on the flow direction [36]. In a DG context, the naturally inherited upwind flux term can simply be adapted on the interface to couple both phases [26] (XDGM). When ghost-penalty is used (CutDGM), the additional stabilisation terms take into account flow direction [37, 38].

1.3. Motivations to use the eXtended DG (XDG) method

The common points and assets that the proposed high-order fixed-mesh geometrically unfitted DG method has with respect to the others mentioned above is summarized in the following. The choice of the eXtended DG method among the available ones covered in section 1.2 is also discussed.
In comparison to standard low-order methods, the well-known advantages of high-order methods to discretize the equations in the bulk of a phase are at least as valuable when it comes to discretize flows with sharp interfaces,

in particular involving compressible effects. Indeed, as pointed out in [39] and [8], local approximation errors across material discontinuities are critical because they will propagate into the rest of the flow field in a cumulative manner, hence impacting the predictive capability of the simulation tool.
Compared to diffusive interface models (see e.g. [40] for a DG method applied to the Baer-Nunziato equations), the general benefits of using a sharp interface treatment are that (i) the singular nature of immiscible, fluid-fluid interfaces (at the continuum scale) is exactly resolved, (ii) the treatment of non-conservative products and the modeling of interfacial source terms are avoided, and (iii) arbitrary high discontinuities can be captured without the need to rely on any complex limiting and interface-sharpening procedures.
Methods using interface-fitted moving meshes in an ALE formalism become particularly complex for significant interface deformations. They require either mesh regularization procedures followed by a remapping or for the direct ALE approach the construction of space-time control volumes [41]. For the interested reader, a complete overview of ALE methods with a focus on compressible multi-material flows is given in [15]. In this respect, the advantage of unfitted methods like the one we are using here is to relieve the coupling between the meshing process and the discretization of the equations.
At this point, the reasons to rely on a high-order unfitted method for our targeted applications should appear clearly. As far as the choice of the discretization technique is concerned, it turns out that discontinuous FEM are particularly well suited compared to their continuous counterparts. The additional freedom which stems from the weak coupling between elements through inter-element flux, jump and/or penalization terms yields a scheme which can naturally handle advection terms thanks to upwinding and ensures the local conservation of physical quantities. Our choice fell on the DG method but other methods like the hybridized DG (HDG) method would have been equally valid candidates. In fact, the HDG method has already been applied to interface problems under the name of eXtended HDG [42] or unfitted HDG [43]. Two other DG-based methods that fall outside the scope of geometrically unfitted methods are still important to mention here as they both achieve a sharp interface treatment of viscous compressible two-phase flows. The first one relies on a reconstructed DG (rDG) discretization and takes advantage of the reconstruction procedure to enforce the interface jumps conditions via a least-squares approach [44]. The second work makes use of a discontinuous Galerkin spectral element method (DGSEM) combined with a local FVM in subdivided cut cells and imposes the jumps

conditions by means of a diffusive generalized Riemann problem [45].
The goal of the present work is to build a geometrically unfitted DG solver for two-phase flows. In this regard, four notable methods designed for two-phase flows are here identified: a space-time unfitted DGM [46, 47], an unfitted DGM [25, 48], a DGM with implicitly defined mesh [49] borrowing some ideas from the UDG and the eXtended DGM (XDGM) [26, 50, 51]. Among the previously mentioned *UDGM*, *CutDGM* and *XDGM*, our work builds upon the XDGM because (i) it offers a way to both discretize bulk equations and enforce interfacial jumps with high-order accuracy by using implicitly defined interface and high-order quadrature rules, (ii) it provides a simple strategy to extend existing DG solver to two-phase sharp interface discretization by considering two separated sets of degrees of freedom on cut cells without the need to locally adapt the mesh topology (contrary to cut-cell type methods), (iii) it makes use of a convenient and efficient agglomeration procedure whose implementation is rather non-intrusive.

1.4. Contributions

Based on the research work identified above, the salient contributions of this paper are:

- the formulation of the XDG discretization for a two-phase compressible Navier-Stokes system of equations,
- the development of a novel strategy to enforce in a high-order manner physical jumps conditions coupling compressible viscous phases in the framework of a geometrically unfitted discontinuous Galerkin method,
- the thorough analysis on several mesh types of different features of the approach concerning the definition of interface convective and diffusive fluxes, the geometrical robustness and the choice of the solved-for variables set,
- the construction of reference verification test cases based on the Method of Manufactured Solution for steady two-phase Navier-Stokes systems with arbitrary specified interface jumps.

In the remaining, the emphasis is put on presenting the various aspects of the spatial discretization of the governing equations, thus focusing on stationary interface test cases. Details about temporal discretization and coupling with

the level-set advection are beyond the scope of this paper.
The outline of the paper is as follows. In section 2, the governing equations in the bulk and the jumps conditions at the interface are established. Section 3 starts by introducing the computational mesh and the extended DG functional space. The weak formulation is then written in a general form. In section 4, a detailed presentation of the strategy developed to enforce the jumps conditions is given. The treatment of the interface convective fluxes is first addressed and then the way to specify the diffusive fluxes is elucidated. In section 5, the framework is applied to manufactured verification test cases in order to study its robustness, its sensitivity with respect to some numerical parameters and its convergence rate. Finally, section 6 summarizes the main findings of this work and discusses its possible perspectives.

2. Problem setting and governing equations

Let $\tilde{\Omega} \subset \mathbb{R}^2$ be the two-dimensional open and bounded physical domain with smooth external boundaries $\partial\tilde{\Omega}$ that are one-dimensional manifold. Let $\tilde{\Gamma} \subset \mathbb{R}$ be a one-dimensional smooth internal interface partitioning $\tilde{\Omega}$ into two non-overlapping subdomains $\tilde{\Omega}_1$ and $\tilde{\Omega}_2$ such that $\tilde{\Omega} \coloneqq \tilde{\Omega}_1 \cup \tilde{\Omega}_2 \cup \tilde{\Gamma}$ with $\tilde{\Gamma} \coloneqq \overline{\tilde{\Omega}}_1 \cap \overline{\tilde{\Omega}}_2$. Each subdomain $\tilde{\Omega}_k$ is occupied by the phase k. The domain boundary $\partial\tilde{\Omega}$ is decomposed into two disjoint subsets $\partial\tilde{\Omega}_{\text{Dir}}$ and $\partial\tilde{\Omega}_{\text{Neu}}$ on which Dirichlet and Neumann boundary conditions, respectively, will be prescribed. Subdomains boundaries associated to phase k, $\partial\tilde{\Omega}_k^*$, are the union of the external boundaries $\partial\tilde{\Omega}_k$ and the interface, i.e. $\partial\tilde{\Omega}_k^* \coloneqq \partial\tilde{\Omega}_k \cup \tilde{\Gamma}$ with $\partial\tilde{\Omega}_k \subset \partial\tilde{\Omega}$. Outward-pointing unit normal vectors defined on the external boundaries are denoted $\mathbf{n}$ while $\mathbf{n}_\Gamma$ is the interface normal pointing outwards from phase $k = 1$ into phase $k = 2$. This setting is sketched on Fig. 1. The notations employed after are summarized in Table 1.

2.1. Equations in the bulk of each phase

In the present work, hydrodynamic equations assuming a single mixture with constant dynamic viscosity, thermal conductivity and heat capacities, with no turbulence model and no source term are considered. Moreover, unsteady effects are not taken into account and the flow is restricted to two dimensions. The two-dimensional steady compressible Navier-Stokes equations that are solved by the solver in the bulk of each k^{th}-phase can therefore

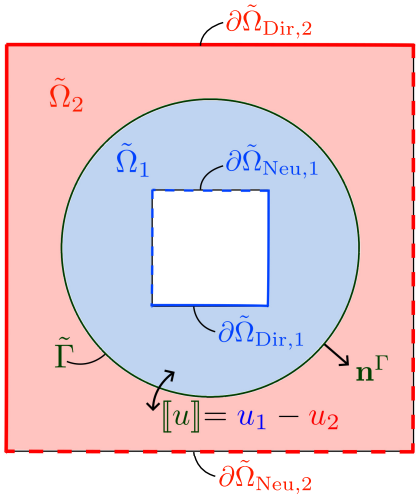


Figure 1: Some notations used for the two-phase interface problem considered in this work.

Notation	Meaning
$\tilde{q}$	exact continuous quantity (VS discretized quantity)
q	scalar quantity
$\mathbf{q}$	vector quantity
$\underline{\boldsymbol{q}}$	tensor quantity
q_k	$= q_{\lvert\tilde{\Omega}_k}$, i.e. quantity q restricted to subdomain $\tilde{\Omega}_k$
$[\![q]\!]$	jump across interface (see Eq. 6)

Table 1: Some notations used in this section

be written, by splitting the convective and the diffusive parts, as

$$\tilde{\nabla} \cdot \left(\underline{\boldsymbol{F}}_k^c(\tilde{\mathbf{u}}_k) \; - \; \underline{\boldsymbol{F}}_k^d(\tilde{\mathbf{u}}_k, \, \tilde{\nabla}\tilde{\mathbf{u}}_k) \right) = \mathbf{S}_k \quad \text{ in } \tilde{\Omega}_k \, , \tag{1}$$

with the continuous gradient operator in two dimensions defined as $\tilde{\nabla} = \left[\frac{\partial}{\partial x} \, , \, \frac{\partial}{\partial y} \right]$. Note that the notation $\tilde{\bullet}$ indicates exact continuous quantities, making the distinction with the discretized ones introduced later. For the sake of clarity, this notation is only used on the global solution vector $\tilde{\mathbf{u}}$ and the gradient. The conservative variables, the convective fluxes and the

diffusive fluxes for phase k are given respectively by

$$\tilde{\mathbf{u}}_k = \begin{bmatrix} \rho_k \\ \rho_k \mathbf{v}_k \\ \rho_k E_k \end{bmatrix} , \ \underline{\boldsymbol{F}}_k^c = \begin{bmatrix} \rho_k \mathbf{v}_k \\ \rho_k \mathbf{v}_k \otimes \mathbf{v}_k + p_k \underline{\boldsymbol{I}} \\ \rho_k \mathbf{v}_k H_k \end{bmatrix} , \ \underline{\boldsymbol{F}}_k^d = \begin{bmatrix} 0 \\ \underline{\boldsymbol{\tau}}_k \\ \underline{\boldsymbol{\tau}}_k \cdot \mathbf{v}_k - \mathbf{q}_k \end{bmatrix} , \tag{2}$$

in which the density of phase k is noted ρ_k, the velocity vector in (x, y) Cartesian coordinates system $\mathbf{v}_k = [v_{x,k}, v_{y,k}]$, the static pressure p_k and the identity matrix $\underline{\boldsymbol{I}}$. The total energy E_k and the total enthalpy H_k per unit mass both include thermal and kinetic contributions, $E_k = e_k + \frac{\mathbf{v}_k \cdot \mathbf{v}_k}{2}$ and $H_k = E_k + \frac{p_k}{\rho_k}$, with the specific internal energy e_k. The constitutive equations for the shear stress tensor $\underline{\boldsymbol{\tau}}_k = \mu_k \left[\tilde{\nabla} \mathbf{v}_k + (\tilde{\nabla} \mathbf{v}_k)^T - \frac{2}{3} \tilde{\nabla} \cdot \mathbf{v}_k \, \underline{\boldsymbol{I}} \right]$ and the heat flux $\mathbf{q}_k = -\lambda_k \tilde{\nabla} T_k$ follow the assumption of a Newtonian fluid and the Fourier law, respectively, with the dynamic viscosity μ_k, the thermal conductivity λ_k, and the temperature T_k. The source term vector $\mathbf{S}$ is set to zero by default, except when the manufactured source terms will be inserted in the equations for the verification test cases in Sec. 5. In addition to the conservative variables, the primitive "p-$\mathbf{v}$-T"variables set will also be considered:

$$\tilde{\mathbf{u}}_k^{\text{prim}} = [p_k, \, \mathbf{v}_k, \, T_k]^T \ . \tag{3}$$

To close the system of Eqs. (1), it must be supplemented by constitutive relations relating intensive thermodynamic properties at equilibrium. For this purpose and as one of the targeted applications of the proposed approach concern gas-liquid flows, a stiffened gas model is used for the volumetric and caloric equations of state that unequivocally determine the state of the fluid inside each phase k:

$$\begin{aligned} p_k(\rho_k, T_k) &= \rho_k(\gamma_k - 1) e_k - \gamma_k p_{\infty,k} \, , \\ e_k(\rho_k, T_k) &= C_{v,k} T_k + \frac{p_{\infty,k}}{\rho_k} \, , \end{aligned} \tag{4}$$

where γ_k is the heat capacity ratio, $C_{v,k}$ is the heat capacity at constant volume and $p_{\infty,k}$ models molecular attractive effects and leads to the "stiffened" properties compared to an ideal gas for which $p_\infty = 0$. The larger the value of p_∞, the closer the flow will behave as nearly incompressible. We can note that more advanced equations of state could readily be used instead since the thermodynamic models for both phases are completely decoupled.

2.2. Jumps conditions at the phase interface

The use of a sharp interface method in this work leads to a so-called local instance formulation requiring the resolution of some jump conditions at the interface. These conditions specify the exchanges of mass, momentum, and energy through the interface and stand as matching conditions between the two phases. They can be derived by writing the balance laws in a control volume around the interface moving with the interface velocity $\mathbf{v}_\Gamma$. Their derivation and the associated assumptions can be found e.g. in [52, Ch. 1.2], [53, Ch. 1.2]. They are written under a generic form as follows:

$$-v_{n,\Gamma}[\![\tilde{\mathbf{u}}]\!] + [\![\underline{\boldsymbol{F}}^c(\tilde{\mathbf{u}}) - \underline{\boldsymbol{F}}^d(\tilde{\mathbf{u}}, \tilde{\nabla}\tilde{\mathbf{u}})]\!] \cdot \mathbf{n}_\Gamma = \tilde{\nabla}_{s,\Gamma} \cdot \underline{\boldsymbol{F}}_\Gamma + \mathbf{S}_\Gamma \,, \tag{5}$$

where $v_{n,\Gamma} = \mathbf{v}_\Gamma \cdot \mathbf{n}_\Gamma$ is the normal interface velocity with the unit normal vector $\mathbf{n}_\Gamma = [n_{x,\Gamma}\,,\ n_{y,\Gamma}]$. The jump operator across the interface $\tilde{\Gamma}$ at a position $\mathbf{x} \in \tilde{\Gamma}$ for some quantity η defined in the bulk of each phase is given by

$$[\![\eta]\!] = (\eta_{1|_\Gamma} - \eta_{2|_\Gamma}) \,, \quad \text{with} \quad \eta_{k|_\Gamma} = \lim_{\epsilon \to 0, \epsilon > 0} \eta(\mathbf{x} \pm \epsilon \mathbf{n}_\Gamma) \,, \tag{6}$$

where the adequate sign is related to the phase considered according to the determined orientation of the normal. On the right-hand side of Eq. (5), $\underline{\boldsymbol{F}}_\Gamma$ is the line flux along the interface while $\mathbf{S}_\Gamma$ represents the surface source. Finally, the surface divergence operator is defined as $\tilde{\nabla}_{s,\Gamma} = \tilde{\nabla} - \mathbf{n}_\Gamma(\mathbf{n}_\Gamma \cdot \tilde{\nabla})$. In this work, the only contribution to the line fluxes is due to surface tension effects while a surface source is present in the energy balance but not elsewhere. Developing Eq. (5) terms by terms for a two dimensional domain with reference Cartesian coordinates system (x, y) yields to:

$$\begin{aligned}
[\![\dot{m}_\Gamma]\!] &= 0 \\
\dot{m}_\Gamma\,[\![v_x]\!] + [\![p]\!]\,n_{x,\Gamma} - \Delta p\,n_{x,\Gamma} - \Delta\tau_{nt}\,t_{x,\Gamma} &= [\![\tau_{xx}]\!]\,n_{x,\Gamma} + [\![\tau_{xy}]\!]\,n_{y,\Gamma} \\
\dot{m}_\Gamma\,[\![v_y]\!] + [\![p]\!]\,n_{y,\Gamma} - \Delta p\,n_{y,\Gamma} - \Delta\tau_{nt}\,t_{y,\Gamma} &= [\![\tau_{yx}]\!]\,n_{x,\Gamma} + [\![\tau_{yy}]\!]\,n_{y,\Gamma} \\
\dot{m}_\Gamma\,[\![E]\!] + [\![p\,v_n]\!] - \Delta p\,v_{n,\Gamma} - \Delta\tau_{nt}\,v_{t,\Gamma} - \Delta q &= [\![\tau_{xx}\,v_x]\!]\,n_{x,\Gamma} + [\![\tau_{yx}\,v_y]\!]\,n_{x,\Gamma} \\
&\quad + [\![\tau_{xy}\,v_x]\!]\,n_{y,\Gamma} + [\![\tau_{yy}\,v_y]\!]\,n_{y,\Gamma} \\
&\quad - [\![q_x]\!]\,n_{x,\Gamma} - [\![q_y]\!]\,n_{y,\Gamma}
\end{aligned} \tag{7}$$

with the net mass flux across the interface defined as $\dot{m}_\Gamma \coloneqq \rho_k(v_{n,k} - v_{n,\Gamma})$ and the meaning of the "Δ" quantities to be found in Table 2. Then, considering

the projections of the momentum jump conditions along the normal $\mathbf{n}_\Gamma$ and the tangent $\mathbf{t}_\Gamma = [t_{x,\Gamma}\,,\ t_{y,\Gamma}]$ to the interface, Eq. (5) becomes:

$$\begin{array}{lllll} & & \llbracket \dot{m}_\Gamma \rrbracket & = & 0 \\ \dot{m}_\Gamma \llbracket v_n \rrbracket + \llbracket p \rrbracket - \Delta p & & & = & \llbracket \tau_{nn} \rrbracket \\ \dot{m}_\Gamma \llbracket v_t \rrbracket & & - \Delta \tau_{nt} & = & \llbracket \tau_{nt} \rrbracket \\ \dot{m}_\Gamma \llbracket E \rrbracket + \llbracket p\, v_n \rrbracket - \Delta p\, v_{n,\Gamma} & & - \Delta \tau_{nt}\, v_{t,\Gamma} - \Delta q & = & \llbracket \tau_{nn}\, v_n \rrbracket + \llbracket \tau_{nt}\, v_t \rrbracket \\ & & & - & \llbracket q_n \rrbracket \end{array} \tag{8}$$

with the normal and tangent bulk fluid velocities at the interface, $v_n = \mathbf{v} \cdot \mathbf{n}_\Gamma$ and $v_t = \mathbf{v} \cdot \mathbf{t}_\Gamma$, respectively, the scalar normal and tangential projections of the shear stress tensor with respect to the normal, $\tau_{nn} = \underline{\boldsymbol{\tau}} \cdot \mathbf{n}_\Gamma \cdot \mathbf{n}_\Gamma$ and $\tau_{nt} = \underline{\boldsymbol{\tau}} \cdot \mathbf{n}_\Gamma \cdot \mathbf{t}_\Gamma$, and the normal heat flux $q_n = \mathbf{q} \cdot \mathbf{n}_\Gamma$. The two sets of jumps conditions Eqs. (7) and Eqs. (8), which both express exactly the same physical conservation laws but according to different coordinate systems, will be useful in the following. Those equations do not constitute sufficient matching conditions to define the problem uniquely. Consequently, they should be supplemented by additional relations. Two common kinematic conditions that are imposed in addition to the jumps concern the jumps on temperature and tangential velocity across the interface:

$$\llbracket v_t \rrbracket = \Delta v_t\,, \tag{9}$$

$$\llbracket T \rrbracket = \Delta T\,. \tag{10}$$

Combining Eq. (9) with the mass balance (first equation in Eqs. (7) and Eqs. (8)), relations on velocity jumps across the interface can be written as:

$$\begin{array}{l} \llbracket v_n \rrbracket = \dot{m}_\Gamma \llbracket 1/\rho \rrbracket \\ \llbracket v_t \rrbracket = \Delta v_t \end{array} \iff \begin{array}{l} \llbracket v_x \rrbracket = \dot{m}_\Gamma \llbracket 1/\rho \rrbracket\, n_{x,\Gamma} + \Delta v_t\, t_{x,\Gamma} \\ \llbracket v_y \rrbracket = \dot{m}_\Gamma \llbracket 1/\rho \rrbracket\, n_{y,\Gamma} + \Delta v_t\, t_{y,\Gamma}\ . \end{array} \tag{11}$$

Moreover, for problems involving mass transfer between the phases, a phase transition model $\dot{m}^{\text{model}}$ must be provided to specify a relation between the definition of $\dot{m}_\Gamma$ and a physical law, hence closing the full set of jumps conditions:

$$\dot{m}_\Gamma \coloneqq \rho_k (v_{n,k} - v_{n,\Gamma}) = \dot{m}_\Gamma^{\text{model}}(\tilde{\mathbf{u}}_1, \tilde{\mathbf{u}}_2)\ , \tag{12}$$

where $\dot{m}_\Gamma^{\text{model}}$ is typically a model derived from kinetic theory and modeling non-equilibrium interfacial resistance to mass transfer [54]. In this work, we

consider all the "Δ" jumps appearing in the relations derived above to be constant, i.e. independent of the fluid quantities and the interface geometry. Their link with physical phenomena they could model are given in Table 2. The jumps conditions that will be used in the remaining of this paper are Eqs. (7) *or* Eqs. (8) (depending on the fluxes to be computed as explained after), Eq. (12), Eq. (9) and Eq. (10). Together with the equations in the bulk Eq. (1), they form the continuous strong form of the problem.

Parameters	Physical meaning	In this work
$\dot{m}_\Gamma$	net mass flux across interface, non zero when evaporation or condensation occurs at the phase interface	= constant
Δp	$= \sigma\kappa$, normal surface stress due to surface tension effects with surface tension coefficient σ and interface curvature κ	= constant
$\Delta\tau_{nt}$	$= \nabla\sigma \cdot \mathbf{t}_\Gamma$, tangential surface stress due to surface tension gradients known as Marangoni effects	= constant
Δq	surface energy flux, due for example to surface radiation or heterogeneous chemical reactions	= constant
ΔT	temperature jump at the interface due to molecular effects, e.g. the presence of the Knudsen layer during evaporation	= constant
Δv_t	partial-slip at the interface, e.g. due to non-continuum flow regimes effects	= constant

Table 2: Overview of the different parameters used in the jumps conditions

3. Discontinuous Galerkin discretization

3.1. Background mesh definition

The first step is the geometrical discretization of the continuous domain $\tilde{\Omega}$ on which the PDE is posed. Let $\mathtt{T}$ be a finite collection of N_T non-overlapping shape-regular simplices or quadrangles $\mathtt{T} = \{\mathtt{T}_m\}_{m=1,\dots,N_T}$ forming an approximate partition of $\tilde{\Omega}$: $\overline{\tilde{\Omega}} \approx \overline{\Omega} = \bigcup_{m=1}^{N_T} \overline{\mathtt{T}}_m$ with $\mathtt{T}_a \cap \mathtt{T}_b = \varnothing$ if $a \neq b$. The

external boundary of $\mathtt{T}$ is noted $\partial\mathtt{T}$. Let $\mathtt{B}$ be the set of N_B boundary faces and $\mathtt{E}$ the set of N_E internal faces such that

$$\mathtt{B} = \left\{ \{\mathtt{B}_b\}_{b=1,\dots,N_B} \,|\, \exists\, \mathtt{T}_m \in \mathtt{T} \text{ s.t. } \mathtt{B}_b = \partial\mathtt{T}_m \cap \partial\mathtt{T} \right\} ,$$
$$\mathtt{E} = \left\{ \{\mathtt{E}_e\}_{e=1,\dots,N_E} \,|\, \exists\, \mathtt{T}_{m1},\, \mathtt{T}_{m2} \in \mathtt{T} \text{ s.t. } \mathtt{E}_e = \partial\mathtt{T}_{m1} \cap \partial\mathtt{T}_{m2} \right\} .$$

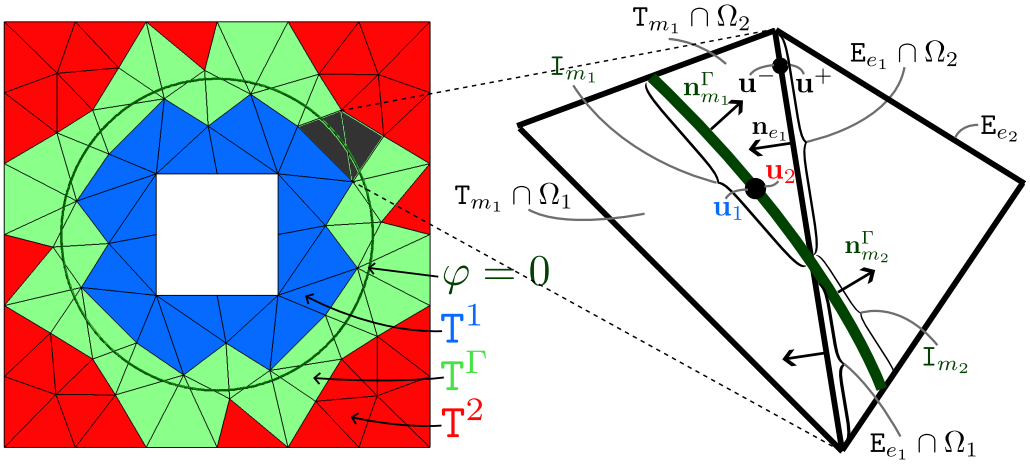
For each $\mathtt{B}_b$ and $\mathtt{E}_e$, $\mathbf{n}_b$ and $\mathbf{n}_e$ are their respective unit outward normal vector. The notation $\mathbf{u}_e^+$ and $\mathbf{u}_e^-$ is used to distinguish traces of the discontinuous quantities on either side of a given $\mathtt{E}_e$, depending on the element from which the internal face is approached. The $\bullet^+$ sign corresponds to the element of which its exterior normal $\mathbf{n}_e$ corresponds to that of the oriented face, whilst $\bullet^-$ corresponds to the other. For a boundary face, $\mathbf{n}_b$ points outward to $\mathtt{T}$ and the traces taken from within the interior of the boundary element are noted $\mathbf{u}_b^+$. Finally, is is assumed that each $\mathtt{T}_m \in \mathtt{T}$ in the coordinate system $\mathbf{x} = (x, y)$ is an image of a fixed reference element $\hat{\mathtt{T}}$ in parametric coordinates $\boldsymbol{\xi} = (\xi_1, \xi_2)$ through a smooth bijective mapping $\chi_m \colon \mathbb{R}^2 \to \mathbb{R}^2,\ \boldsymbol{\xi} \mapsto \mathbf{x}$, i.e. $\mathtt{T}_m = \chi_m(\hat{\mathtt{T}})$.

3.2. Cut mesh definition

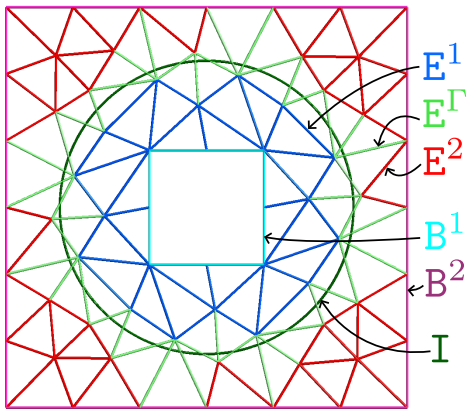
In this work, the continuous phase interface $\tilde{\Gamma}$ is implicitly represented by the zero iso-0 contour of a scalar level-set function φ separating the domain into two non-overlapping regions such that

$$\tilde{\Gamma} \approx \Gamma \coloneqq \{\mathbf{x} \in \Omega : \varphi(\mathbf{x}) = 0\} ,$$
$$\Omega_1 \coloneqq \{\mathbf{x} \in \Omega : \varphi(\mathbf{x}) < 0\} ,\ \Omega_2 \coloneqq \{\mathbf{x} \in \Omega : \varphi(\mathbf{x}) > 0\} .$$

From this definition, the set of elements intersected by the interface is identified by $\mathtt{T}^\Gamma \coloneqq \{\mathtt{T}_m \colon \mathtt{T}_m \cap \Gamma \neq \varnothing\ , \forall\ \mathtt{T}_m \in \mathtt{T}\}$, as well as the active elements for each phase k as $\mathtt{T}^{k,*} \coloneqq \{\mathtt{T}_m \colon \mathtt{T}_m \cap \Omega_k \neq \varnothing\ , \forall\ \mathtt{T}_m \in \mathtt{T}\}$. The set of elements containing only a single phase k can then be defined as $\mathtt{T}^k \coloneqq \mathtt{T}^{k,*} \setminus \mathtt{T}^\Gamma$, such that $\mathtt{T} = \mathtt{T}^1 \cup \mathtt{T}^2 \cup \mathtt{T}^\Gamma$. The sets of internal faces and boundary faces intersected by the interface, $\mathtt{E}^\Gamma$ and $\mathtt{B}^\Gamma$, and the ones contained strictly in a single phase k, $\mathtt{E}^k$ and $\mathtt{B}^k$, are defined in a similar way. Finally, the set of interfaces per element is defined by $\mathtt{I} \coloneqq \left\{\mathtt{I}_m \colon \mathtt{I}_m = \mathtt{T}_m \cap \Gamma\ , \forall\ \mathtt{T}_m \in \mathtt{T}^\Gamma\right\}$, with the associated interface normal calculated by $\mathbf{n}_m^\Gamma = \frac{\nabla\varphi}{\|\nabla\varphi\|}$, where ∇ is the broken gradient operator. The various notations used for the discretization are gathered in Fig. 2.



(a) Illustration of the different sets of elements $\mathtt{T}$ (left) and zoom on two cut elements showing the definition of local components (right).



(b) Illustration of the different sets of internal faces $\mathtt{E}$, domain boundaries $\mathtt{B}$ and phase interfaces $\mathtt{I}$.

Figure 2: Geometrical discretization of the continuous domain in Fig.1.

3.3. Discontinuities representation: extension of the function space

In order to enable independent solution approximations on two sides of the interface within one cut finite element, hence allowing arbitrary discontinuities to be represented in an unfitted manner, it is necessary to consider independent sets of degrees of freedom on those elements. To do so, the active finite element space for each phase k, $\mathcal{W}_{p,k}$, is defined on the corresponding

active elements $\mathtt{T}^{k,*}$ as

$$\mathcal{W}_{p,k}(\varphi,\mathtt{T}) \coloneqq \left\{ \mathbf{w} \in \left[L^2(\Omega)\right]^4 : \mathbf{w}_{|\mathtt{T}_m} \circ \chi_m^{-1} \in \left[\mathcal{P}_p(\hat{\mathtt{T}})\right]^4, \forall\, \mathtt{T}_m \in \mathtt{T}^{k,*} \right\} \ , \quad (13)$$

where $\mathbf{w}_{|\mathtt{T}_m}$ indicates a function whose support is restricted to element $\mathtt{T}_m$, L^2 is an Hilbert space equipped with L^2-inner product $\int_\Omega fg \,\mathrm{d}\mathbf{x}$ and $\mathcal{P}_p(\hat{\mathtt{T}})$ is a function space of degree $p \geq 0$ spanned by the N_p shape functions $\boldsymbol{\phi}_m = \{\phi_{m,1}, ..., \phi_{m,N_p}\}$. The extended DG space is then introduced as

$$\mathcal{W}_p^X(\varphi,\mathtt{T}) = \mathcal{W}_{p,1}(\varphi,\mathtt{T}) \times \mathcal{W}_{p,2}(\varphi,\mathtt{T}) \ . \quad (14)$$

This space depends on the level-set field φ as it is composed of two spaces defined on the respective active meshes $\mathtt{T}^{k,*}$ that depend on the position of the level-set iso-0. Moreover, since the cut elements are shared by both active meshes, the degrees of freedom are doubled on those elements. Hence in the following, $\mathbf{u} \in \mathcal{W}_p^X$ will mean that on cut elements $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2)$ with $\mathbf{u}_k \in \mathcal{W}_{p,k}$ such that for a given element $\mathtt{T}_m \in \mathtt{T}^\Gamma$

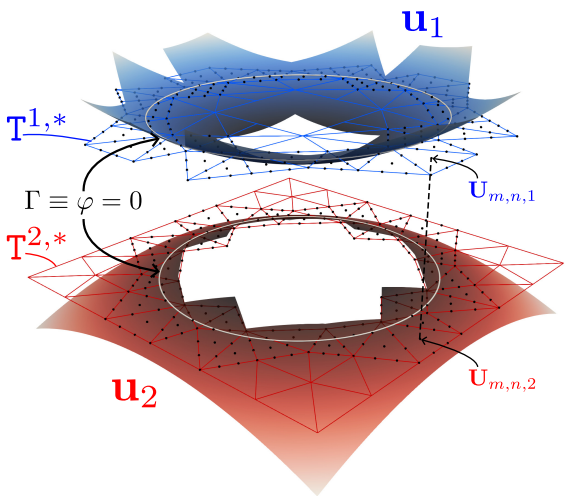
$$\mathbf{u}_m(\mathbf{x}) = \sum_{k=1}^{2} \mathbf{u}_{m,k}(\mathbf{x}) = \sum_{k=1}^{2} \sum_{n=1}^{N_p} \phi_{m,n}(\mathbf{x}) \mathbf{U}_{m,n,k} \ , \quad (15)$$

where $\mathbf{U}_{m,n,k} \in \mathbb{R}^{1\times 4}$ is the nodal solution unknown in element m corresponding to the n^{th}-degree of freedom associated to phase k. This is illustrated in Fig. 3. It is worth mentioning here that the final solution of the coupled interface problem, noted $\mathbf{u}_m^{\mathrm{F}}$, is in fact given by the sum of the solutions in each phase restricted to their corresponding subdomains,

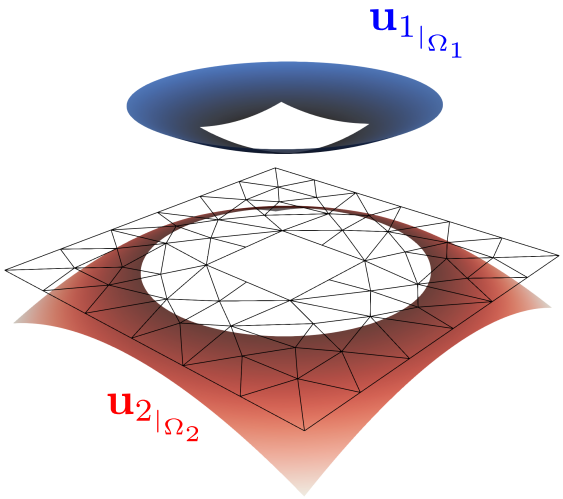
$$\mathbf{u}_m^{\mathrm{F}}(\mathbf{x}) = \sum_{k=1}^{2} \mathbf{u}_{m,k}(\mathbf{x}) \Big|_{\Omega_k} = \mathbf{u}_{m,1}\, \mathrm{H}(-\varphi) + \mathbf{u}_{m,2}\, \mathrm{H}(\varphi) \ ,$$

with the Heaviside function $H(\varphi)$ truncating the solution at the level-set iso-0, as shown in Fig. 3. From an implementation point of view however, the extended function space on the cut elements does not contain truncated functions on either side of the interface. The only modification compared to non-cut elements is thus the extension of the standard DG space by the addition of another decoupled one, without altering the reference shape functions spanning the original space. The cut-off at the interface is realized during the assembly of the variational form by integrating on the respective subdomains. To evaluate those integrals, $\mathbf{u}_k(\mathbf{x})$ only needs to be computed at

locations $\mathbf{x} \in \Omega_k$, so that the solution is never evaluated outside of its corresponding subdomain. For visualization purposes, the resulting solutions in each phase can then be truncated at the interface in a post-processing step to obtain $\mathbf{u}^{\mathrm{F}}$.



(a) The interface Γ is located by the zero iso-contour of a level-set function φ. The n^{th} nodal DOF (identified by the black dots) in element m for phase k is noted $\mathbf{U}_{m,n,k}$, they are doubled on cut elements. The set of active elements for phase k is $\mathbb{T}^{k,*}$.



(b) The restriction of the solutions in their respective domain Ω_k, $\mathbf{u}_{k_{|\Omega_k}}$, gives the sharp interface treatment.

Figure 3: Illustration of the XDGM based on the example of Fig. 2.

3.4. Weak form discretized by the extended DGM

For the purposes of discretization, the Jacobian of the diffusive fluxes with respect to the solution gradients, $\underline{\boldsymbol{G}}(\tilde{\mathbf{u}})$, is introduced such that

$$\underline{\boldsymbol{F}}_k^d(\tilde{\mathbf{u}}_k,\, \tilde{\nabla}\tilde{\mathbf{u}}_k) = \underline{\boldsymbol{G}}_k(\tilde{\mathbf{u}}_k)\tilde{\nabla}\tilde{\mathbf{u}}_k\,. \tag{16}$$

Going from the continuous strong form of the compressible Navier-Stokes equations (in the bulk) to a discrete weak form suited to be implemented inside a solver requires several steps. In this work, the framework and formalism presented in [55] is followed and applied to the problem formed by Eqs. (1,5). This results in the following non-linearly coupled weak form to be solved: find $\mathbf{u} \in \mathcal{W}_p^X$ such that

$$\hat{\mathcal{N}}(\mathbf{u},\mathbf{w}) = \hat{\mathcal{N}}^1(\mathbf{u}_1,\mathbf{w}) + \hat{\mathcal{N}}^2(\mathbf{u}_2,\mathbf{w}) + \hat{\mathcal{N}}^\Gamma(\mathbf{u}_1,\mathbf{u}_2,\mathbf{w}) = 0 \quad \forall \mathbf{w} \in \mathcal{W}_p^X\,, \tag{17}$$

with the contributions from the bulk of each phase k unchanged compared to the classical DGM, i.e.

$$\hat{\mathcal{N}}^k(\mathbf{u}_k,\mathbf{w}) \equiv \sum_{m\in\mathtt{T}^k} \mathcal{V}_{m,k}(\mathbf{u}_k,\mathbf{w}) + \sum_{e\in\mathtt{E}^k} \hat{\mathcal{E}}_{e,k}(\mathbf{u}_k,\mathbf{w}) + \sum_{b\in\mathtt{B}^k} \hat{\mathcal{B}}_{b,k}(\mathbf{u}_k,\mathbf{w})\,, \tag{18}$$

and the contributions from the cut elements, faces and boundaries, and the interface given by

$$\begin{aligned}\hat{\mathcal{N}}^\Gamma(\mathbf{u}_1,\mathbf{u}_2,\mathbf{w}) \equiv \sum_{k\in\{1,2\}} \{ &\sum_{m\in\mathtt{T}^\Gamma} \mathcal{V}_{m,k}(\mathbf{u}_k,\mathbf{w}) + \sum_{e\in\mathtt{E}^\Gamma} \hat{\mathcal{E}}_{e,k}(\mathbf{u}_k,\mathbf{w}) \\ &+ \sum_{b\in\mathtt{B}^\Gamma} \hat{\mathcal{B}}_{b,k}(\mathbf{u}_k,\mathbf{w}) + \sum_{m\in\mathtt{I}} \hat{\mathcal{I}}_{m,k}(\mathbf{u}_1,\mathbf{u}_2,\mathbf{w}) \}\,,\end{aligned} \tag{19}$$

with the test functions $\mathbf{w}$ taken from the same space as the trial functions $\mathbf{u}$. The different contributions to the global variational formulation $\hat{\mathcal{N}}$ are expressed as:

$$\begin{aligned}\mathcal{V}_{m,k}(\mathbf{u},\mathbf{w}) = \int_{\mathtt{T}_m\cap\Omega_k} [\, &-\underline{\boldsymbol{F}}_k^c(\mathbf{u}) \\ &+ \underline{\boldsymbol{F}}_k^d(\mathbf{u},\nabla\mathbf{u})\,] : \nabla\mathbf{w}\,\mathrm{d}\mathbf{x},\end{aligned} \tag{20}$$

$$\begin{aligned}\hat{\mathcal{E}}_{e,k}(\mathbf{u},\mathbf{w}) = &\int_{\mathtt{E}_e\cap\Omega_k} \left(\hat{\underline{\boldsymbol{F}}}_{\mathtt{E},k}^c - \hat{\underline{\boldsymbol{F}}}_{\mathtt{E},k}^d\right)\cdot\mathbf{n}_e\left(\mathbf{w}^+ - \mathbf{w}^-\right)\,\mathrm{d}s \\ &+ \int_{\mathtt{E}_e\cap\Omega_k} [\,(\hat{\mathbf{u}}_{\mathtt{E},k} - \mathbf{u}^+)(\underline{\boldsymbol{G}}_k^T(\mathbf{u}^+)\nabla\mathbf{w}^+) \\ &- (\hat{\mathbf{u}}_{\mathtt{E},k} - \mathbf{u}^-)(\underline{\boldsymbol{G}}_k^T(\mathbf{u}^-)\nabla\mathbf{w}^-)\,]\cdot\mathbf{n}_e\,\mathrm{d}s\,,\end{aligned} \tag{21}$$

$$\hat{\mathcal{B}}_{b,k}(\mathbf{u},\mathbf{w}) = \int_{\mathtt{B}_b\cap\Omega_k} \left(\hat{\underline{\boldsymbol{F}}}^c_{\mathtt{B},k} - \hat{\underline{\boldsymbol{F}}}^d_{\mathtt{B},k}\right)\cdot\mathbf{n}_b\mathbf{w}^+\,\mathrm{d}s + \int_{\mathtt{B}_b\cap\Omega_k} (\hat{\mathbf{u}}_{\mathtt{B},k}-\mathbf{u}^+)(\underline{\boldsymbol{G}}^T_k(\mathbf{u}^+)\nabla\mathbf{w}^+)\cdot\mathbf{n}_b\,\mathrm{d}s\,. \tag{22}$$

$$\hat{\mathcal{I}}_{m,k}(\mathbf{u}_1,\mathbf{u}_2,\mathbf{w}) = \int_{\mathtt{I}_m} \left(\hat{\underline{\boldsymbol{F}}}^c_{\mathtt{I},k} - \hat{\underline{\boldsymbol{F}}}^d_{\mathtt{I},k}\right)\cdot\mathbf{n}^\Gamma_{m,k}\mathbf{w}\,\mathrm{d}s + \int_{\mathtt{I}_m} (\hat{\mathbf{u}}_{\mathtt{I},k}-\mathbf{u}_k)(\underline{\boldsymbol{G}}^T_k(\mathbf{u})\nabla\mathbf{w})\cdot\mathbf{n}^\Gamma_{m,k}\,\mathrm{d}s\,, \tag{23}$$

Because of the discontinuous solution approximation, the flux terms are not uniquely defined across element faces and boundaries and must therefore be replaced by numerical flux functions identified by $\hat{\bullet}$ in Eqs. (20, 21, 22, 23). A few comments are in order at this stage. First, the assembly of the volume, face and boundary contributions, $\mathcal{V},\hat{\mathcal{E}},\hat{\mathcal{B}}$ respectively, in the cut and non-cut elements, faces or boundaries differs only by the integration domain. On non-cut entities, classical Gaussian quadrature rules can be applied to evaluate the integrals. On the cut faces and boundaries, the one-dimensional integrals on $\mathtt{E}_e\cap\Omega_k$ and $\mathtt{B}_b\cap\Omega_k$ respectively can easily be recast as classical one-dimensional Gaussian quadratures on reference segments after having computed the intersection between the level-set iso-0 and the segment. On the other hand, dedicated quadrature techniques are required to evaluate the integrals over the implicitly-defined two-dimensional surfaces $\mathtt{T}_m\cap\Omega_k$ and one-dimensional surfaces $\mathtt{I}_m$. Three different approaches have been included in the solver: (i) a hierarchical moment-fitting (HMF) method proposed in [56] and implemented into the solver by Schrooyen et al. [57], (ii) a high-order quadrature method for domains implicitly-defined by multivariate polynomials [31], included through the external library *Algoim* [58], and (iii) high-order quadrature techniques for triangles and rectangles [59], via the interfacing with the Parallel Hierarchical Grid (*PHG)* toolbox [60]. Note that the additional computational effort coming from the doubling of the degrees of freedom and the integration on either side of the interface is expected to be negligible in comparison to the whole residual assembly. This is because the quadrature rules construction is very fast and the integration of the fluxes is realized via efficient local matrix-matrix multiplications.

We can also remark that since the interface is free to cross the elements of the underlying mesh in any manner, there exist pathological intersection scenarios in which one of the phase occupies only a small volume fraction of the cut element. This results in conditioning issues of the resulting system,

intersection-dependent instabilities and suboptimal error estimates [20]. In the XDGM, a cell agglomeration technique is applied to overcome those issues. It relies on the definition of a threshold parameter being equal to the ratio between the cut volume fraction and the full cell volume, noted δ_{agg}, and from which it is decided if an element has to be agglomerated or not. The technique is based on the work of [26] and [27] and its implementation in the present solver was first presented in [57, 61].
Finally, in the same way as numerical fluxes across faces couple the solution between elements inside a given phase, the coupling at the interface between the phases is achieved by the addition of the terms $\hat{\mathcal{I}}$. Those terms include numerical fluxes whose role is to impose the interface jumps conditions. The specification of those fluxes (which are let unspecified at this stage, hence the use of the notation $\hat{\bullet}$) is explained in Sec. 4..

4. Enforcement of jumps conditions into the weak form

This section addresses the main novelty of the present work, i.e. the computation of interfacial numerical fluxes in order to couple through complex jumps conditions two general sets of governing equations involving inviscid and viscous effects. Capitalizing on the way the fluxes are specified in the bulk of each phase, the interfacial jumps are weakly imposed into the coupled variational formulation Eq. (17). The key idea of weak constraint imposition is to relax the strength of imposing the constraints to the benefit of improved accuracy, stability and robustness [20]. These advantages are all the more marked at a phase interface across which huge contrasts in material properties potentially exist. Before detailing the computation of the convective and then diffusive fluxes, some notations are first introduced. The normal to the interface $\mathbf{n}_m^\Gamma$ has the same direction as the normal vector emanating from phase 1, $\mathbf{n}_{m,1}^\Gamma$, and is opposite to the normal associated to phase 2: $\mathbf{n}_m^\Gamma = \mathbf{n}_{m,1}^\Gamma = -\mathbf{n}_{m,2}^\Gamma$. The "swapped" phase index k^* is simply defined as

$$k^* = \begin{cases} 1 & \text{if } k = 2\,, \\ 2 & \text{if } k = 1\,. \end{cases} \tag{24}$$

Finally, a tag t_k is associated to each phase such that

$$t_k = \begin{cases} 1 & \text{if } k = 1\,, \\ -1 & \text{if } k = 2\,. \end{cases} \tag{25}$$

4.1. Interface convective numerical fluxes

The specification of the convective fluxes across the interface relies on the concept of multiphase Riemann solvers and reads

$$\underline{\hat{\boldsymbol{F}}}^{c}_{\mathtt{I},k} \cdot \mathbf{n}^{\Gamma}_{m,k} = \mathcal{H}_{\mathtt{I},k}(\mathbf{u}_k, \mathbf{u}_{k^*}, \mathbf{n}^{\Gamma}_{m,k}) \quad \text{on } \mathtt{I}\,. \tag{26}$$

The difference with the Riemann solvers used at elements faces is that the multiphase solver considered here returns two distinct fluxes across the iso-0, i.e. $\mathcal{H}_{\mathtt{I},1} \neq \mathcal{H}_{\mathtt{I},2}$, such that the jumps are incorporated into the resulting fluxes. At the same time, waves propagation structure associated to the hyperbolic part of the system is taken into account such as to provide upwinding to stabilize the non-linear convective effects. More precisely, two different approximate multiphase Riemann solvers, the HLLC and HLLCP, are considered here. Those have been shown to produce accurate results, with the advantages of being much faster and more robust than exact Riemann solvers [62]. They are both based on the extension of the classical single-phase HLLC solver. The waves structures from which they are built are sketched in Fig. 4.

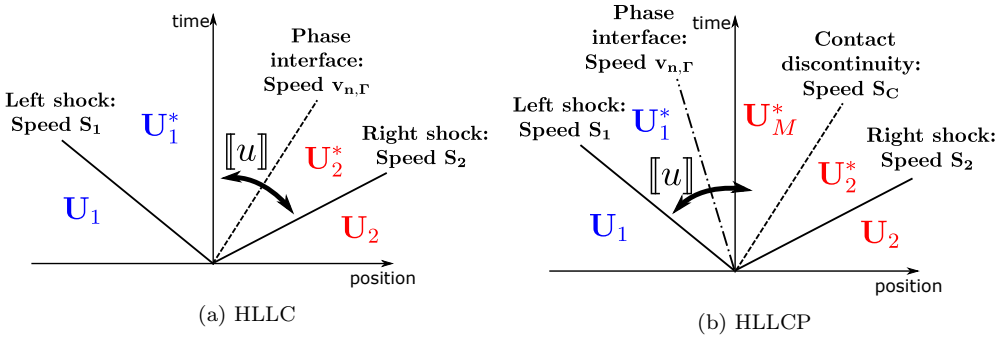


Figure 4: Waves structure of the two-phase Riemann solvers considered.

Since the goal of the solvers is to solve (approximately) the one dimensional Riemann problem along the normal to the interface, the jumps conditions to consider are given by Eqs. (8), without the tangential momentum jump. An important remark should be made at this point. The jumps on the viscous stresses and heat fluxes across the interface are not taken into account for solving the Riemann problem. Indeed, including their contributions would lead to a system of equations that would not only depend on the states on both sides of the interface but also on their gradients. This would

change the hyperbolic nature of the problem and would require to solve a generalized Riemann problem including both convective and viscous fluxes (see e.g. [45, 63, 64]). The approach followed here is thus similar to the one in [3–5] where the normal momentum jump is split into one pressure jump and one jump on the normal viscous stress:

$$\dot{m}_\Gamma \left[\!\left[v_n\right]\!\right] + \left[\!\left[p\right]\!\right] - \Delta p = \left[\!\left[\tau_{nn}\right]\!\right] \Rightarrow \begin{cases} \left[\!\left[p\right]\!\right] = \Delta p - \dot{m}_\Gamma \left[\!\left[v_n\right]\!\right] , \\ \left[\!\left[\tau_{nn}\right]\!\right] = 0 . \end{cases} \tag{27}$$

Moreover, if $\dot{m}_\Gamma \neq 0$, the energy jump without the diffusive contributions is not equivalent to the normal momentum jump and it could therefore be envisaged that this jump is also taken into account in the Riemann problem by splitting the energy jump in the following way:

$$\begin{aligned} \dot{m}_\Gamma \left[\!\left[E\right]\!\right] + \left[\!\left[p\, v_n\right]\!\right] - \Delta p\, v_{n,\Gamma} - \Delta\tau_{nt}\, v_{t,\Gamma} - \Delta q = \left[\!\left[\tau_{nn}\, v_n\right]\!\right] + \left[\!\left[\tau_{nt}\, v_t\right]\!\right] - \left[\!\left[q_n\right]\!\right] \\ \Rightarrow \begin{cases} \dot{m}_\Gamma \left[\!\left[E\right]\!\right] + \left[\!\left[p\, v_n\right]\!\right] = \Delta p\, v_{n,\Gamma} , \\ \left[\!\left[q_n\right]\!\right] - \left[\!\left[\tau_{nn}\, v_n\right]\!\right] - \left[\!\left[\tau_{nt}\, v_t\right]\!\right] = \Delta\tau_{nt}\, v_{t,\Gamma} + \Delta q , \end{cases} \end{aligned} \tag{28}$$

with the first condition handled by the Riemann solver and the second by the diffusive flux functions. Because the choice of the Riemann solver (HLLC VS HLLCP) determines the type of splitting, thus also the conditions to be imposed by the diffusive numerical flux functions (discussed in Sec. 4.2), Table 3 compiles the two possibilities considered in this work.

	Splitting #1	Splitting #2
Convective fluxes	$\left[\!\left[p\right]\!\right] = \Delta p - \dot{m}_\Gamma \left[\!\left[v_n\right]\!\right]$ /	$\left[\!\left[p\right]\!\right] = \Delta p - \dot{m}_\Gamma \left[\!\left[v_n\right]\!\right]$ $\dot{m}_\Gamma \left[\!\left[E\right]\!\right] + \left[\!\left[p\, v_n\right]\!\right] = \Delta p\, v_{n,\Gamma}$
Diffusive fluxes	$\left[\!\left[\tau_{nn}\right]\!\right] = 0$ $\left[\!\left[q_n\right]\!\right] - \left[\!\left[\tau_{nn}\, v_n\right]\!\right] - \left[\!\left[\tau_{nt}\, v_t\right]\!\right]$ $= -\dot{m}_\Gamma \left[\!\left[E\right]\!\right] - \left[\!\left[p\, v_n\right]\!\right]$ $+ \Delta p\, v_{n,\Gamma} + \Delta\tau_{nt}\, v_{t,\Gamma} + \Delta q$	$\left[\!\left[\tau_{nn}\right]\!\right] = 0$ $\left[\!\left[q_n\right]\!\right] - \left[\!\left[\tau_{nn}\, v_n\right]\!\right] - \left[\!\left[\tau_{nt}\, v_t\right]\!\right]$ $= \Delta\tau_{nt}\, v_{t,\Gamma} + \Delta q$

Table 3: The two strategies considered to split the jumps conditions between convective and diffusive interface numerical fluxes

Both solvers follow the same global steps to compute the numerical fluxes. The procedure is basically the same as the one described in [65] and it is

here restated for completeness. Given $\mathbf{u}_1(\boldsymbol{\xi}_q) = \mathbf{u}_1$ and $\mathbf{u}_2(\boldsymbol{\xi}_q) = \mathbf{u}_2$, the solution vectors from both phases interpolated at a given quadrature point with parametric coordinates $\boldsymbol{\xi}_q$ on the iso-0 contour of the level-set, the normal $\mathbf{n}^\Gamma(\boldsymbol{\xi}_q) = \mathbf{n}^\Gamma$ and the tangent $\mathbf{t}^\Gamma(\boldsymbol{\xi}_q) = \mathbf{t}^\Gamma$ vectors computed from the level-set, do:

Step 1. Project $\mathbf{u}_1$ and $\mathbf{u}_2$ onto the interface-attached frame and assemble the vectors $\mathbf{V}_1$ and $\mathbf{V}_2$ as:

$$\mathbf{V}_k = (\rho_k\,,\ v_{n,k}\,,\ e_k\,,\ p_k)^T\ ,$$

with $v_{n,k} = \mathbf{v}_k \cdot \mathbf{n}^\Gamma$.

Step 2. Compute an approximation of the two external shock wave speeds S_1 and S_2 based on the linearization of the Lax curves [62]. The tangent to the wave curves in terms of the variables $\tilde{\mathbf{V}}_k = (\tilde{\rho}_k, \tilde{v}_{n,k}, \tilde{p}_k)$ can be expressed as

$$\tilde{\mathbf{V}}_1 = \begin{bmatrix} \rho_1 \\ v_{n,1} \\ p_1 \end{bmatrix} + \eta_1 \begin{bmatrix} 1 \\ -c_1/\rho_1 \\ c_1^2 \end{bmatrix}\ , \quad \tilde{\mathbf{V}}_2 = \begin{bmatrix} \rho_2 \\ v_{n,2} \\ p_2 \end{bmatrix} + \eta_2 \begin{bmatrix} 1 \\ c_2/\rho_2 \\ c_2^2 \end{bmatrix}\ , \quad (29)$$

where $c_k = c_k(\mathbf{V}_k)$ is the speed of sound in phase k obtained from an appropriate thermodynamic model and η_k are the slopes given by

$$\begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} = \frac{1}{c_1 c_2(c_2/\rho_1 + c_1/\rho_2)} \begin{bmatrix} c_2^2 & -c_2/\rho_2 \\ c_1^2 & c_1/\rho_1 \end{bmatrix} \begin{bmatrix} v_{n,1} - v_{n,2} - \Delta v_n \\ p_1 - p_2 - \Delta p \end{bmatrix}\ , \quad (30)$$

with the interface jumps on normal velocity Eq. (11), Δv_n, and pressure Eq. (8), Δp. The estimation of the shock wave speeds is finally obtained by

$$\begin{aligned} S_0 &= \min\left(v_{n,1} - c_1, \tilde{v}_{n,1} - \tilde{c}_1\right)\ , \\ S_1 &= \min\left(v_{n,2} + c_2, \tilde{v}_{n,2} + \tilde{c}_2\right)\ , \end{aligned} \quad (31)$$

with $\tilde{c}_k = c_k(\tilde{\mathbf{V}}_k)$.

Step 3. Solve the Riemann problem to obtain the solution states $\mathbf{V}_1^*$ and $\mathbf{V}_2^*$, and the interface normal velocity $v_{n,\Gamma}$:

HLLC: Solve Eq. (32).
HLLCP: Solve Eq. (33), including Eq. (34) when $\dot{m}_\Gamma = 0$.

Step 4. Reproject the interface aligned solution states to the reference frame and assemble the set of conservative variables $\mathbf{u}_1^*$ and $\mathbf{u}_2^*$:

$$\mathbf{v}_k^* = v_{n,k}^* \mathbf{n}^\Gamma + v_{t,k}^* \mathbf{t}^\Gamma \ , \quad v_{t,k}^* = \mathbf{v}_k \cdot \mathbf{t}^\Gamma \ ,$$
$$\mathbf{u}_k^* = \rho_k^* \left(1 \, , \ \mathbf{v}_k^* \, , \ \mathbf{e}_k^* + 1/2 \mathbf{v}_k^* \cdot \mathbf{v}_k^*\right)^T \ .$$

An important remark is that the tangential component of the velocity, $v_{t,k}^*$, is set equal to the tangential velocity from the external state. No jump is therefore imposed on the tangential velocity through the interface convective fluxes, it is the role of the interface diffusive fluxes covered in the next section.

Step 5. Compute the numerical fluxes normal to the interface:

$$\mathcal{H}_{\mathrm{I},1} = \begin{cases} \underline{\boldsymbol{F}}_1^c(\mathbf{u}_1) \cdot \mathbf{n}^\Gamma & \text{if } S_1 > 0 \\ \underline{\boldsymbol{F}}_1^c(\mathbf{u}_1) \cdot \mathbf{n}^\Gamma + S_1(\mathbf{u}_1^* - \mathbf{u}_1) & \text{otherwise} \end{cases}$$
$$\mathcal{H}_{\mathrm{I},2} = \begin{cases} \underline{\boldsymbol{F}}_2^c(\mathbf{u}_2) \cdot \mathbf{n}^\Gamma & \text{if } S_2 \leq 0 \\ \underline{\boldsymbol{F}}_2^c(\mathbf{u}_2) \cdot \mathbf{n}^\Gamma + S_2(\mathbf{u}_2^* - \mathbf{u}_2) & \text{otherwise} \end{cases}$$

4.1.1. HLLC multiphase Riemann solver

The two-phase HLLC multiphase solver implemented in the solver is based on the one presented in [66, Ch. 6.3.2.1] and its extension to consider mass transfer effects in [67]. This extension includes the possibility to consider non-zero mass flux across the interface, i.e. considering a discontinuity in the interface normal velocities and the recoil pressure effect in the pressure jump. Because the structure of the HLLC solver has three waves, there are too few equations to be able to simultaneously impose the jumps on the normal momentum and total energy (if $\dot{m}_\Gamma \neq 0$). This solver is therefore part of the first splitting strategy (Table 3). Combining the Rankine-Hugoniot relations across the two external shocks with the interface jumps, the system to solve reads:

$$S_1(\rho_1^* - \rho_1) = \rho_1^* v_{n,1}^* - \rho_1 v_{n,1} \tag{32a}$$
$$S_1(\rho_1^* v_{n,1}^* - \rho_1 v_{n,1}) = \rho_1^* v_{n,1}^{*,2} - \rho_1 v_{n,1}^2 + p_1^* - p_1 \tag{32b}$$
$$S_1(\rho_1^* E_1^* - \rho_1 E_1) = \rho_1^* v_{n,1}^* E_1^* - \rho_1 v_{n,1} E_1 + p_1^* v_{n,1}^* - p_1 v_{n,1} \tag{32c}$$
$$S_2(\rho_2^* - \rho_2) = \rho_2^* v_{n,2}^* - \rho_2 v_{n,2} \tag{32d}$$
$$S_2(\rho_2^* v_{n,2}^* - \rho_2 v_{n,2}) = \rho_2^* v_{n,2}^{*,2} - \rho_2 v_{n,2}^2 + p_2^* - p_2 \tag{32e}$$

$$S_2(\rho_2^* E_2^* - \rho_2 E_2) = \rho_2^* v_{n,2}^* E_2^* - \rho_2 v_{n,2} E_2 + p_2^* v_{n,2}^* - p_2 v_{n,2} \tag{32f}$$
$$v_{n,1}^* - v_{n,2}^* = \dot{m}_\Gamma \left(1/\rho_1^* - 1/\rho_2^*\right) \tag{32g}$$
$$p_1^* - p_2^* = \Delta p - \dot{m}_\Gamma^2 \left(1/\rho_1^* - 1/\rho_2^*\right) \tag{32h}$$
$$\dot{m}_\Gamma = \dot{m}_\Gamma^{\text{model}}(\mathbf{V}_1^*, \mathbf{V}_2^*) \tag{32i}$$

The last equation corresponds to Eq. (12) for the closure of the interface mass flux. The strategy to reduce this system to a single non-linear equation is as follows. The quantities ρ_k^*, p_k^* and e_k^* are first expressed as a function of $v_{n,k}^*$ through the Rankine-Hugoniot relations (the first six equations). They are then plugged into the mass and normal momentum jumps to express $v_{n,1}^*$ and $v_{n,2}^*$ as a function of $\dot{m}_\Gamma$ and the known $\mathbf{V}_k$ states. The last equation for the mass flux model is then solved by a Newton's method with the $\mathbf{V}_k^*$ states now expressed entirely based on $\dot{m}_\Gamma$.

4.1.2. HLLCP multiphase Riemann solver

This second solver considers, in addition to the contact discontinuity and the two shock waves, a fourth non-classical wave modeling the phase interface. The distinction between the contact (C) and phase (P) discontinuities has physical roots: during phase transition a Knudsen layer containing the newly created vapor is formed, separating the liquid and gas equilibrium regions. This has been considered in [66, Ch. 6.4.6.2] and [13]. With the structure of the HLLCP solver, it is possible to take into account the energy balance across the phase interface. The use of this solver is therefore linked to the second splitting strategy (Table 3). If it is assumed, as in Fig. 4b, that the phase interface moving with a normal velocity $v_{n,\Gamma}$ is located between the left star and the middle states, $\mathbf{V}_1^*$ and $\mathbf{V}_M^*$, while the contact discontinuity with speed S_C separates the middle and the right state $\mathbf{V}_2^*$, then the equations to solve are:

$$S_1(\rho_1^* - \rho_1) = \rho_1^* v_{n,1}^* - \rho_1 v_{n,1} \tag{33a}$$
$$S_1(\rho_1^* v_{n,1}^* - \rho_1 v_{n,1}) = \rho_1^* v_{n,1}^{*,2} - \rho_1 v_{n,1}^2 + p_1^* - p_1 \tag{33b}$$
$$S_1(\rho_1^* E_1^* - \rho_1 E_1) = \rho_1^* v_{n,1}^* E_1^* - \rho_1 v_{n,1} E_1 + p_1^* v_{n,1}^* - p_1 v_{n,1} \tag{33c}$$
$$S_2(\rho_2^* - \rho_2) = \rho_2^* v_{n,2}^* - \rho_2 v_{n,2} \tag{33d}$$
$$S_2(\rho_2^* v_{n,2}^* - \rho_2 v_{n,2}) = \rho_2^* v_{n,2}^{*,2} - \rho_2 v_{n,2}^2 + p_2^* - p_2 \tag{33e}$$
$$S_2(\rho_2^* E_2^* - \rho_2 E_2) = \rho_2^* v_{n,2}^* E_2^* - \rho_2 v_{n,2} E_2 + p_2^* v_{n,2}^* - p_2 v_{n,2} \tag{33f}$$
$$v_{n,M}^* = v_{n,2}^* \tag{33g}$$

$$p_M^* = p_2^* \tag{33h}$$

$$\rho_1^*(v_{n,1}^* - v_{n,\Gamma}) = \rho_M^*(v_{n,M}^* - v_{n,\Gamma}) \tag{33i}$$

$$p_1^* - p_M^* - \Delta p = -\dot{m}_\Gamma(v_{n,1}^* - v_{n,M}^*) \tag{33j}$$

$$\begin{aligned} p_1^*\, v_{n,1}^* - p_M^*\, v_{n,M}^* - \Delta p v_{n,\Gamma} = -\dot{m}_\Gamma[e_1^* + 0.5\,(v_{n,1}^{*,2} + v_{t,1}^2) \\ - e_M^* - 0.5\,(v_{M,1}^{*,2} + v_{t,2}^2)] \end{aligned} \tag{33k}$$

$$S_C = v_{n,M}^* \tag{33l}$$

$$\dot{m}_\Gamma = \rho_1^*(v_{n,1}^* - v_{n,\Gamma}) \tag{33m}$$

$$\dot{m}_\Gamma = \dot{m}_\Gamma^{\text{model}}(\mathbf{V}_1^*, \mathbf{V}_M^*) \tag{33n}$$

$$\frac{\rho_1^*}{\rho_M^*} \approx \frac{\rho_1}{\rho_2} \tag{33o}$$

Equations (33a-33f) are the two Rankine-Hugoniot relations across the two external shock waves, Eqs. (33g-33h) are the equality of normal velocity and pressure across the contact discontinuity, Eqs. (33i-33k) are the interface jumps conditions (without the diffusive contributions), Eq. (33l) is a direct consequence of the equality of normal velocity across the contact discontinuity, Eq. (33m) is the definition of the interface mass flux and Eq. (33n) corresponds to Eq. (12) for the closure for the interface mass flux. This represents 14 equations for 15 unknowns: $\mathbf{V}_1^*$ (4), $\mathbf{V}_2^*$ (4), $\mathbf{V}_M^*$ (4), $v_{n,\Gamma}$, S_C and $\dot{m}_\Gamma$. There are two possibilities at this stage. The first one is to solve this non-linear undertermined system of equations with an iterative procedure, which can be quite costly. The second one is to introduce an additional equation, which will in any case come from an assumption as all the physical relations have already been used in the 14 equations. This is the choice made here with the introduction of Eq. (33o) based on [13]. It can be noticed that the implementation in the solver allows also to treat the situation where the phase interface and the contact discontinuity are permuted by checking the sign of $\dot{m}_\Gamma$. The procedure described in [13, App. A] to reduce the system to a single non-linear equation is followed: all the unknowns are expressed as a function of $\dot{m}_\Gamma$ and the non-linear equation to solve is given by Eq. (33n). At the end of the procedure, $\mathbf{V}_2^* = \mathbf{V}_M^*$ if $\dot{m}_\Gamma \geq 0$ or $\mathbf{V}_1^* = \mathbf{V}_M^*$ if $\dot{m}_\Gamma < 0$ are set. The interface convective fluxes can then be assembled at the step 5 explained above.

Although this solver is mostly designed to deal with phase transition problems, the situation of vanishing $\dot{m}_\Gamma$ can also be treated. In this case, the phase interface and contact discontinuity coincide. For such cases, a mod-

ification to the HLLCP solver of [13] is proposed here. When $\dot{m}_\Gamma^{\text{model}} = 0$, the mass jump Eq. (33i) and the energy jump Eq. (33k) become redundant. The idea is to replace those two equations by two other ones. Those new equations should ensure that the temperature returned by the solver satisfies the kinematic condition Eq. (10). Indeed, the HLLC solver and the original HLLCP solver described above do not take into account this condition. This means that the interface convective fluxes that are assembled at the end do not incorporate this condition either. As this condition is anyway enforced as a Dirichlet jump condition by the interface diffusive fluxes, the global procedure should still converge towards a solution that satisfies the interface temperature jump. But to improve the consistency between the convective and diffusive fluxes, a strategy to impose the temperature jump inside the Riemann solver is pursued. This is achieved by replacing Eq. (33i) and Eq. (33k) by a 2×2 internal non-linear system to be solved for the unknowns ρ_M^* and e_M^* such that the jumps on temperature and pressure are simultaneously satisfied:

$$\left\{ \begin{array}{l} T_1^*(e_1^*, p_1^*) - T_M^*(e_M^*, p_M^*) = \Delta T \\ p_1^*(\rho_1^*, e_1^*) - p_M^*(\rho_M^*, e_M^*) = \Delta p \end{array} \right. \tag{34}$$

where thermodynamic relations specific to each phase are used to compute the temperatures and pressures as a function of other quantities.

4.2. Interface diffusive numerical fluxes

The way to incorporate the Dirichlet jumps on the solutions and the Robin jumps on the diffusive fluxes is inspired by the strategies developed for incompressible two-phase flows, e.g. in [20, Ch. 4.2] and [68, App. A]: when considering the assembling of the interface term for a given element, the solution, resp. the flux, on the other side of the interface is compensated by the jump condition to correctly account for the intended discontinuity in the solution, resp. the flux. In addition, those convex combinations between both sides are weighted by a parameter α defined by

$$\alpha_k = \frac{(q_k)^a}{(q_1)^a + (q_2)^a}\,, \quad \alpha_{k^*} = 1 - \alpha_k\,, \tag{35}$$

where a is a user-defined power ($a = 1$ gives the well-known harmonic weighting) and q is a quantity related to the diffusion process. Here we set $q_k = 2\mu_k + \lambda_k / C_{v,k}$. Biasing the numerical flux towards one phase or the other, depending on the rate of diffusion of the interface phenomena involved, may both improve solution accuracy and conditioning of the system [33].

4.2.1. Numerical flux function for the direct term

The numerical flux function $\hat{\underline{\boldsymbol{F}}}^d_{\mathtt{I},k}$ in Eq. (23) is specified based on the Interior Penalty (IP) method as

$$\hat{\underline{\boldsymbol{F}}}^d_{\mathtt{I},k}(\mathbf{u}_k, \nabla\mathbf{u}_k, \mathbf{u}_{k^*}, \nabla\mathbf{u}_{k^*}) = \alpha_{k^*}\underline{\boldsymbol{F}}^d_k + \alpha_k\left(\underline{\boldsymbol{F}}^d_{k^*} + \mathbf{J}^{d,\mathrm{Rob}} \otimes \mathbf{n}^\Gamma_k\right) - \underline{\boldsymbol{\delta}}_{\mathtt{I},k}\,, \quad (36)$$

where $\mathbf{J}^{d,\mathrm{Rob}}$ is a vector incorporating the Robin jumps conditions on the fluxes whose explicit expression is given just after. The focus here is on the two first terms, the third term (penalty term) is discussed in Sec. 4.2.3. The role of two first terms is to take into account via a multivalued interfacial flux the physical jumps prescribed on the diffusive fluxes. It can be noticed the multiplication of $\mathbf{J}^{d,\mathrm{Rob}}$ by the normal $\mathbf{n}^\Gamma_k$ associated to phase k such that the jumps values are either added to or subtracted from the diffusive fluxes in the other phase. The term $\mathbf{J}^{d,\mathrm{Rob}}$ embodies the non-linear Robin jumps conditions to be imposed and its expression depends on the splitting strategy (Table 3) adopted. In both cases, the condition $[\![\tau_{nn}]\!] = 0$ is taken into account to reformulate the x- and y-momentum jumps of Eq. (7). Depending on the splitting, the energy jump differs so that:

$$\begin{aligned}
&\mathbf{J}^{d,\mathrm{Rob}}(\mathbf{u}_1, \mathbf{u}_2) \\
&= -\begin{cases}
\begin{bmatrix} 0 \\ \dot{m}_\Gamma\,[\![v_t]\!]\,t^\Gamma_x - \Delta\tau_{nt}\,t^\Gamma_x \\ \dot{m}_\Gamma\,[\![v_t]\!]\,t^\Gamma_y - \Delta\tau_{nt}\,t^\Gamma_y \\ \dot{m}_\Gamma\,[\![E]\!] + [\![p\,v_n]\!] - \Delta p\,v_{n,\Gamma} - \Delta\tau_{nt}\,v_{t,\Gamma} - \Delta q \end{bmatrix} & \text{if splitting \#1}\,, \\
\begin{bmatrix} 0 \\ \dot{m}_\Gamma\,[\![v_t]\!]\,t^\Gamma_x - \Delta\tau_{nt}\,t^\Gamma_x \\ \dot{m}_\Gamma\,[\![v_t]\!]\,t^\Gamma_y - \Delta\tau_{nt}\,t^\Gamma_y \\ -\Delta\tau_{nt}\,v_{t,\Gamma} - \Delta q \end{bmatrix} & \text{if splitting \#2}\,.
\end{cases}
\end{aligned} \quad (37)$$

The negative sign in front of the bracket stands from the fact that the jumps need to be imposed on the opposite of the diffusive fluxes (see Eq. (23)).
In the most general case, the jumps on the diffusive fluxes could depend both on the solutions and their gradients. This is the case for example if a jump on the heat flux has to be prescribed. In this situation, the function $\mathbf{J}^{d,\mathrm{Rob}}$ must compensate in the energy balance the contributions from the viscous dissipation terms since they are included in the diffusive fluxes $\underline{\boldsymbol{F}}^d_k$

by definition but the goal is to impose a jump solely on the heat fluxes. This yields

$$
\begin{aligned}
&J^{d,\mathrm{Rob}}_{\mathrm{energy}}(\mathbf{u}_1, \nabla\mathbf{u}_1, \mathbf{u}_2, \nabla\mathbf{u}_2) \\
&= -\begin{cases} -[\![\underline{\boldsymbol{\tau}} \cdot \mathbf{v}]\!] \cdot \mathbf{n}^\Gamma + \dot{m}_\Gamma \, [\![E]\!] + [\![p \, v_n]\!] \\ \qquad -\Delta p \, v_{n,\Gamma} - \Delta\tau_{nt} \, v_{t,\Gamma} - \Delta q \end{cases} \quad \text{if splitting } \#1\,, \\
&\qquad\quad \begin{cases} -[\![\underline{\boldsymbol{\tau}} \cdot \mathbf{v}]\!] \cdot \mathbf{n}^\Gamma - \Delta\tau_{nt} \, v_{t,\Gamma} - \Delta q \end{cases} \quad \text{if splitting } \#2\,.
\end{aligned}
\tag{38}
$$

The solver is designed to treat the most general case of $\mathbf{J}^{d,\mathrm{Rob}}$ depending both on the solutions and their gradients coming from both phases.

4.2.2. Numerical flux function for the transpose term

The numerical flux function $\hat{\mathbf{u}}_{\mathrm{I},k}$ in Eq. (23) is specified as

$$
\hat{\mathbf{u}}_{\mathrm{I},k}(\mathbf{u}_k, \mathbf{u}_{k^*}) = (1 - \theta\alpha_{k^*})\mathbf{u}_k + \theta\alpha_{k^*}\mathbf{u}\mathbf{J}_k \;, \tag{39}
$$

with

$$
\mathbf{u}\mathbf{J}_k(\mathbf{u}_k, \mathbf{u}_{k^*}) \coloneqq \left(\mathbf{u}_{k^*} + t_k \mathbf{J}^{d,\mathrm{Dir}}\right) \;, \tag{40}
$$

where t_k is given in Eq. (25), the choice of θ selects the symmetric (SIPG) or non-symmetric (NIPG) interior penalty method ($\theta = 1$ for SIPG, $\theta = -1$ for NIPG) and $\mathbf{J}^{d,\mathrm{Dir}}$ is a vector incorporating the Dirichlet jumps conditions on the solutions whose explicit expression is given just after. The flux function is also multivalued such that when the trace of the solution on the other side of the interface is evaluated, this trace is compensated by the jump data, either positively or negatively depending on the sign of t_k, which results in the imposition of the prescribed jumps on the solutions. The term $\mathbf{J}^{d,\mathrm{Dir}}$ should enclose the jumps on the solutions. The distinction should be made here between schemes that employ the primitive "p-$\mathbf{v}$-T"variables set, denoted $\mathbf{u}_k^{\mathrm{prim}}$, and the ones that solve for another variables set, e.g. the conservative variables $\mathbf{u}_k$, which will be labelled as $\mathbf{u}_k^{\mathrm{cons}}$ for the sake of clarity. There exists a correspondence between both sets through transformations $u_k^{\mathrm{p}\to\mathrm{c}}$ and $u_k^{\mathrm{c}\to\mathrm{p}}$ based on the equations of state associated to each phase k:

$$
\begin{aligned}
\mathbf{u}_k^{\mathrm{prim}} &= u_k^{\mathrm{c}\to\mathrm{p}}(\mathbf{u}_k^{\mathrm{cons}}) \qquad \text{Conservative to primitive sets}\,, \\
\mathbf{u}_k^{\mathrm{cons}} &= u_k^{\mathrm{p}\to\mathrm{c}}(\mathbf{u}_k^{\mathrm{prim}}) \qquad \text{Primitive to conservative sets}\,.
\end{aligned}
\tag{41}
$$

When the solver solves for $\mathbf{u}_k^{\text{prim}}$, the term $\mathbf{J}^{d,\text{Dir}}$ can readily be assembled by combining Eq. (27), Eq. (11) and Eq. (10):

$$\mathbf{J}^{d,\text{Dir}}_{\text{prim}}(\mathbf{u}_1^{\text{prim}}, \mathbf{u}_2^{\text{prim}}) = \begin{bmatrix} \Delta p - \dot{m}_\Gamma [\![v_n]\!] \\ \dot{m}_\Gamma \, [\![1/\rho]\!] \, n_{x,\Gamma} + \Delta v_t \, t_{x,\Gamma} \\ \dot{m}_\Gamma \, [\![1/\rho]\!] \, n_{y,\Gamma} + \Delta v_t \, t_{y,\Gamma} \\ \Delta T \end{bmatrix} \text{ if primitive variables set .} \tag{42}$$

When a variables set different from "p-$\mathbf{v}$-T"is used, there exists no explicit and unique expression to be assigned to $\mathbf{J}^{d,\text{Dir}}$ because since both phases can be governed by different thermodynamic models, it is not possible to convert $\mathbf{J}^{d,\text{Dir}}_{\text{prim}}$ with a transformation $u_k^{\text{p}\to\text{c}}$. The approach chosen here is to convert the conservative sets to "p-$\mathbf{v}$-T"primitive ones for both phases, add or subtract from those vectors the known $\mathbf{J}^{d,\text{Dir}}_{\text{prim}}$ vector, convert the results back to conservative variables and set $\mathbf{J}^{d,\text{Dir}}_{\text{cons}}$ as being equal to the jumps between those compensated conservative sets:

$$\begin{aligned} \mathbf{J}^{d,\text{Dir}}_{\text{cons}} &= u_1^{\text{p}\to\text{c}}(\mathbf{u}_1^{\text{prim}}) - u_2^{\text{p}\to\text{c}}(\mathbf{u}_2^{\text{prim}}) \\ &= u_1^{\text{p}\to\text{c}}(\mathbf{u}_2^{\text{prim}} + \mathbf{J}^{d,\text{Dir}}_{\text{prim}}) - u_2^{\text{p}\to\text{c}}(\mathbf{u}_1^{\text{prim}} - \mathbf{J}^{d,\text{Dir}}_{\text{prim}}) \quad \text{if conservative var. set .} \end{aligned} \tag{43}$$

Both the conservative and "p-$\mathbf{v}$-T"primitive variables sets are tested in this work.

4.2.3. Nitsche's penalization term

The last numerical flux function that remains to be specified is the penalty term in Eq. (36). The idea of the IP method is that instead of a rough and direct imposition, this term is consistently added to the discretization to minimize the average interpolation errors between the solutions on either sides of the interface. Similarly to the penalty in the bulk, the penalization term is either scaled with a scalar diffusivity or with a tensorial one according to [69]:

$$\underline{\boldsymbol{\delta}}_{\mathtt{I},k}(\mathbf{u}_k, \mathbf{u}_{k^*}) = \sigma_{\mathtt{I}} \begin{cases} \beta_{\mathtt{I}} \, [\mathbf{u}_k - \mathbf{u}\mathbf{J}_k] \otimes \mathbf{n}_k^\Gamma \,, & \text{standard IP,} \\ 0.5 \, [\underline{\boldsymbol{G}}_k(\mathbf{u}_k) + \underline{\boldsymbol{G}}_{k^*}(\mathbf{u}_{k^*})] \, [\mathbf{u}_k - \mathbf{u}\mathbf{J}_k] \otimes \mathbf{n}_k^\Gamma \,, & \text{tensorial IP,} \end{cases} \tag{44}$$

where $\mathbf{u}\mathbf{J}_k$ is given by Eq. (40) and the diffusivity scaling $\beta_{\mathtt{I}}$ in the scalar case is computed from a weighted average, effectively scaling the penalty parameters with the minimum diffusion coefficient between the two phases

$\beta_{\mathtt{I}} = \alpha_2\,\beta_1 + \alpha_1\,\beta_2$. The expression of the stabilisation parameter $\sigma_{\mathtt{I}}$ is crucial to ensure the well-posedness of the discrete problem. In [70], bounds on σ in the bulk for different element and mesh types are derived from trace inverse inequalities. The parameter at the interface takes a similar expression as in the bulk,

$$\sigma_{\mathtt{I}} = \max\left(\frac{s \times C_{IP}}{h_1}\,,\ \frac{s \times C_{IP}}{h_2}\right)\,, \tag{45a}$$

$$C_{IP} = \begin{cases} (2p+1)(2p+2)/2 & \text{for 2-D tri}\,, \\ (2p+1)^2 & \text{for 2-D quad}\,, \end{cases} \tag{45b}$$

$$h_k = \frac{|\check{\mathtt{T}}_{m,k}|}{|\partial\check{\mathtt{T}}_{m,k}|} = \frac{\text{“Agglomerated element surface in phase k”}}{\text{“Agglomerated face length in phase k”}}\,. \tag{45c}$$

When tensorial IP is used, a safety factor of $s = 2$ was found to produce convergent results in all the different test cases considered as part of this work. When scalar penalty is used, this factor has to be increased. This is discussed in Sec. 5.2.2. Rather than relying on the heuristic approach adopted here, we can mention the recent work [71] presenting a rigorous framework to study the trace inverse inequality for cut-cells. Compared to bulk penalty, the fundamental difference is on the choice of the geometric factor h_k which follows the reasoning in [26]. To avoid stability issues with small cut cells, the ratio between element area and perimeter is here computed from the agglomerated topology, i.e. by using the ratio between agglomerated element area in phase k, noted $|\check{\mathtt{T}}_{m,k}|$, and its perimeter, $|\partial\check{\mathtt{T}}_{m,k}|$. For an element $\mathtt{T}_m$ which is a target element associated to a set of source elements $\mathtt{T}_m^{\text{src}}$, the agglomerated area in phase k is computed as

$$|\check{\mathtt{T}}_{m,k}| = |\mathtt{T}_m \cap \Omega_k| + \sum_{j \in \mathtt{T}_m^{\text{src}}} |\mathtt{T}_j \cap \Omega_k|\ . \tag{46}$$

The perimeter is the sum of the cut edges lengths (contained in phase k) defining the target and the associated source elements and the lengths of the interface segments crossing the target and the sources. From this sum, the contributions of the set of cut edges which are shared between the target and the sources, noted as $\mathtt{E}_m^{\text{shared}}$, are removed to obtain the perimeter of the agglomerated element:

$$|\partial\check{\mathtt{T}}_{m,k}| = |\partial\mathtt{T}_m \cap \Omega_k| + |\mathtt{I}_m| + \sum_{j \in \mathtt{T}_m^{\text{src}}} \left(\,|\partial\mathtt{T}_j \cap \Omega_k| + |\mathtt{I}_j|\,\right) - \sum_{l \in \mathtt{E}_m^{\text{shared}}} |\mathtt{E}_l \cap \Omega_k|\ . \tag{47}$$

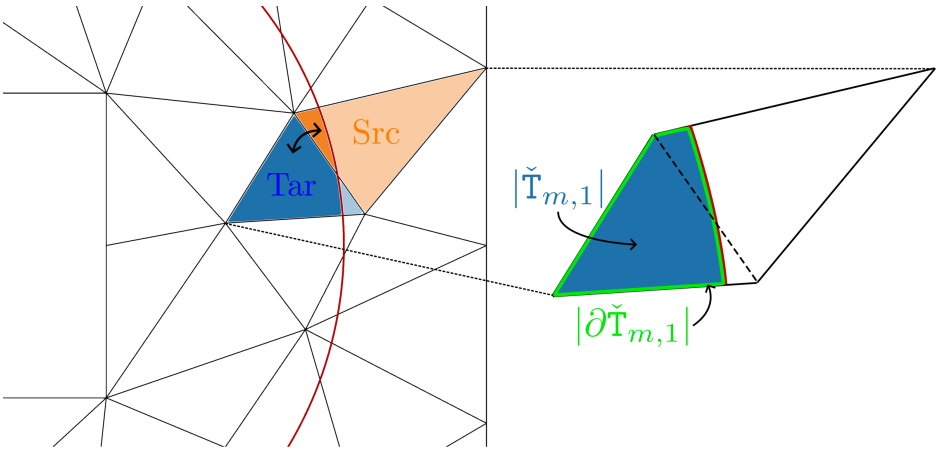


Figure 5: Illustration of the computation of the geometric factor h of the penalization term for an agglomerated cells pair in phase 1.

This is illustrated in Fig. 5. It can be noticed that the quadrature rules to compute the cut areas and the lengths of the interface are available from the construction of the cut-cell quadratures.

4.3. Jacobians of interface terms

In addition to the appropriate specification of the interface numerical flux functions, a key ingredient of the solver is the computation of the Jacobians of those fluxes. Indeed, the non-linear procedure to solve the resulting system of equations require the Jacobian of the global residual and the contributions of the interface fluxes to this global Jacobian are therefore needed. In order to identify the Jacobians that need to be assembled, the contribution of the interface term for phase k in an element m is first written by using the specifications of the numerical fluxes in Eqs. (26, 36, 39, 44) (for the tensorial penalty version):

$$\begin{aligned}
\mathtt{I}_{m,k}(\mathbf{u}_k, \mathbf{u}_{k^*}, \mathbf{w}) = \int_{\mathtt{I}_m} & \left[\mathcal{H}_{\mathtt{I},k}(\mathbf{u}_{k,\Gamma}, \mathbf{u}_{k^*,\Gamma}, \mathbf{n}_k^\Gamma)\right] \cdot \mathbf{w} \\
& - \left[\{\{\underline{\boldsymbol{F}}^d(\mathbf{u}_\Gamma, \nabla\mathbf{u}_\Gamma)\}\}_{\alpha^*} \cdot \mathbf{n}_k^\Gamma + \alpha_k \mathbf{J}^{d,\mathrm{Rob}}(\mathbf{u}_{k,\Gamma}, \nabla\mathbf{u}_{k,\Gamma}, \mathbf{u}_{k^*,\Gamma}, \nabla\mathbf{u}_{k^*,\Gamma})\right] \cdot \mathbf{w} \\
& + \sigma_\mathtt{I} \left(\mathbf{u}_{1,\Gamma} - \mathbf{u}_{2,\Gamma} - \mathbf{J}^{d,\mathrm{Dir}}(\mathbf{u}_{k,\Gamma}, \mathbf{u}_{k^*,\Gamma})\right) \left(\mathbf{n}^\Gamma \cdot \{\{\underline{\boldsymbol{G}}^T(\mathbf{u}_\Gamma)\}\} \cdot \mathbf{n}_k^\Gamma\right) \cdot \mathbf{w}\, \mathrm{d}s \\
& - \theta \int_{\mathtt{I}_m} \alpha_{k^*} \left(\mathbf{u}_{1,\Gamma} - \mathbf{u}_{2,\Gamma} - \mathbf{J}^{d,\mathrm{Dir}}(\mathbf{u}_{k,\Gamma}, \mathbf{u}_{k^*,\Gamma})\right) \cdot \left(\mathbf{n}^\Gamma \cdot \underline{\boldsymbol{G}}_k^T(\mathbf{u}_{k,\Gamma}) \nabla\mathbf{w}\right)\, \mathrm{d}s\ ,
\end{aligned} \tag{48}$$

where the weighted average of the diffusive fluxes across the interface is obtained via the expression $\{\{\underline{\boldsymbol{q}}\}\}_{\alpha^*} = \alpha_2\,\underline{\boldsymbol{q}}_1 + \alpha_1\,\underline{\boldsymbol{q}}_2$. The vector $\mathtt{I}_{m,k}$ will be added to the global residual vector at positions corresponding to the nodal degrees of freedom associated to phase k in element m. To explain how the Jacobians are assembled per phase, let us set $k = 1$. The corresponding contributions to the global Jacobian matrix for $\mathtt{I}_{m,1}$ requires the computation of the local matrices

$$\frac{\partial \mathtt{I}_{m,1}}{\partial \mathbf{u}_1} \text{ and } \frac{\partial \mathtt{I}_{m,1}}{\partial \mathbf{u}_2} \ ,$$

since the interface fluxes for phase $k = 1$ depend on the solution in both phases. Those are computed via forward finite differences by perturbing the fluxes with respect to the solutions or the gradients of the solutions.

5. Numerical experiments

Three test cases with a stationary interface are considered to verify the implemented solver: a compressible Couette test case (Sec. 5.1), a manufactured solution inclined interface test case (Sec. 5.2) and a manufactured solution spinning bubble test case (Sec. 5.3). An overview of the various parameters selected for each of them is provided in Table 4. It should be pointed out at this stage that the second test case cannot be considered as a high contrast interface problem, unlike the other two. This is done on purpose so that a sensibility analysis about various numerical parameters can be conducted without being influenced by potential effects associated to high properties jumps. The two other test cases are there to complement the analysis. The reference solutions for each test case as well as the manufactured source terms for the two last ones can be found in the supplementary material accompanying this paper (available online). Moreover, our symbolic *Python* software that has been developed for generating complex manufactured solutions satisfying prescribed conditions is available at [72] and is described in [73]. It must be noted that for each test case, the initial solutions given to the solver are slightly perturbed compared to the exact ones so that the initial value of the residual is not small and the capacity of the solver to bring it to a prescribed tolerance level can be evaluated. For each of the test case, an order-of-accuracy study is conducted by evaluating the experimental order of convergence (EOC), which depends both on the grid level of refinement h

and the polynomial order p, as

$$\mathrm{EOC}(h,p) = \frac{\log\left(E^p_{h_{i-1}}/E^p_{h_i}\right)}{\log\left(h_{i-1}/h_i\right)}\ , \tag{49}$$

where h_{i-1}/h_i is the ratio of grid sizes between two consecutive grids and $E^p_{h_i} = \left\|\mathbf{u}^p_{h_i} - \tilde{\mathbf{u}}\right\|$ denotes the error of the numerical approximation of degree p on the grid with spacing h_i, noted $\mathbf{u}^p_{h_i}$, with respect to the reference analytical solution, $\tilde{\mathbf{u}}$. The errors used in the following are measured in terms of two different norms:

$$\|\mathbf{u} - \tilde{\mathbf{u}}\|^2_{L_2(\Omega)} = \sum_{k\in\{1,2\}} \sum_{m\in\mathrm{T}^{k,*}} \int_{\mathrm{T}_m\cap\Omega_k} (\mathbf{u}_k - \tilde{\mathbf{u}}_k)^2 \ \mathrm{d}\mathbf{x}\,, \tag{50}$$

$$\|\mathbf{u} - \tilde{\mathbf{u}}\|_{L_\infty(\Omega)} = \max_{\forall k\in\{1,2\}} \left(\max_{\forall q\in Q_k} \left|\mathbf{u}_k(\mathbf{x}_q) - \tilde{\mathbf{u}}_k(\mathbf{x}_q)\right| \right)\ , \tag{51}$$

where the integrals in Eq. (50) use the special volume quadrature rules in the cut elements to integrate on both sides of the interface and Q_k in Eq. (51) labels the set of volume quadrature points belonging to phase k with their corresponding coordinates $\mathbf{x}_q$.

		Compressible Couette	Inclined interface (V1)	Spinning bubble (V1)
Bulk properties (Sec. 2.1)	$\gamma_1\,,\ \gamma_2$	3.0 , 1.4	1.4 , 1.6	1.911 , 1.290
	$C_{v,1}\,,\ C_{v,2}$	3.229 , 0.708	2.5 , 2.0	977.704, 991.20
	$p_{\infty,1}\,,\ p_{\infty,2}$	1.0e5 , 0.0	0.0 , 0.0	7.16e9 , 0.0
	$\mu_1\,,\ \mu_2$	10.0 , 1.0	1.0 , 0.25	4.45e-3 , 6.98e-5
	$\lambda_1\,,\ \lambda_2$	10.0 , 1.376	1.0 , 0.25	35.24 , 0.127
Jumps values (Tab. 2)	$\dot{m}_\Gamma$	0.0	0.0	-0.001
	Δp	0.0	-0.2	200.0
	$\Delta\tau_{nt}$	0.0	0.2	0.2
	Δq	1000.0	1.0	1000.0
	ΔT	0.0	0.5	100.0
	Δv_t	0.0	-0.5	-50.0

Numerics - General	Variables sets Iterative solver	$\rho - \rho\mathbf{v} - \rho E$ Newton +Gauss	$p - \mathbf{v} - T$ Newton +Gauss	$p - \mathbf{v} - T$ Newton +GMRES (ILU)
Numerics - Convective	Riemann bulk Riemann iso-0 (Sec. 4.1)	SLAU2 HLLC	HLLC HLLCP	SLAU2 HLLC
Numerics - Diffusive	IP Penalty (Eqs. (44))	SIPG Tensorial	SIPG Tensorial	SIPG Tensorial
Numerics - XDG	Splitting (Tab. 3) δ_{agg} (Sec. 3.4) Quadrature Exponent a (Eq. (35))	# 1 (jump on heat flux!) 0.2 *Algoim* 0.0	# 2 0.3 *PHG* 1.0	# 1 0.3 *PHG* 1.0

Table 4: Parameters of the test cases used for grid convergence study. Some parameters of the "Inclined interface" and "Spinning bubble" test cases are modified for some sensitivity analysis study afterwards, hence the "V1" denomination here.

5.1. Compressible Couette test case

This test case is taken from [74] and is slightly modified. An horizontal interface separates a stiff fluid on the top and a gas-like fluid on the bottom inside an infinite two-dimensional channel, as represented in Fig. 6. At the lower channel boundary, a no-slip condition and a temperature of $310\,K$ are imposed while the upper boundary moves at a speed of $10\,m/s$ and has a temperature of $330\,K$. All the interface jumps are set to zero except a heat flux source equal to $\Delta q = 1000.0\,W/m^2$. A jump on the heat flux (and not on the difference between the heat flux and the viscous dissipation) is imposed here, i.e. Eq. (38) is considered for the energy jump condition. Since the solution is assumed to be independent of the horizontal coordinate, it is possible to compute an analytical solution. Because the analytical solution is composed of second-order polynomials for the temperature and velocity fields, this test case is solved in terms of the conservative variables such that higher order terms are involved and a convergence analysis for $p > 1$ is possible. Two of

the structured quadrilateral grids used for the convergence study are shown in Fig. 7. In Table 5, the errors on the pressure and horizontal velocity fields are listed for several interpolation orders and grid sizes. For $p = 4$, the refinement stops earlier than for the lower polynomial orders because the absolute values of the errors quickly become negligible. Overall, the data indicate that the solver produces optimally convergent results in the sense that the orders

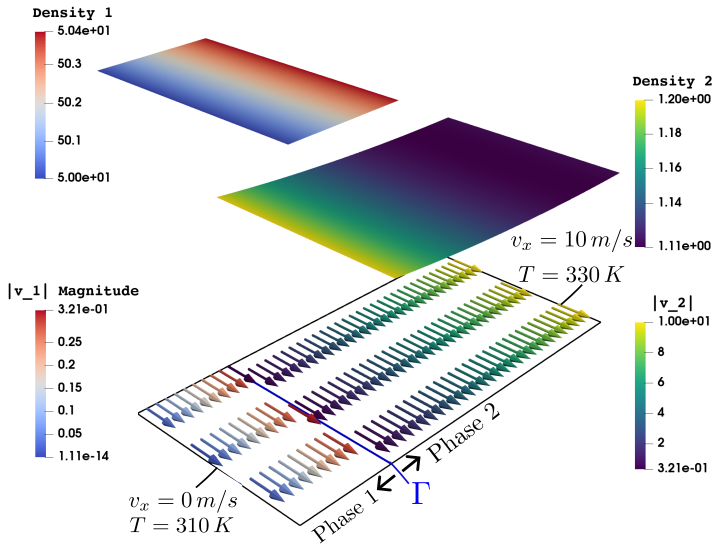


Figure 6: Domain of the **compressible Couette** test case with the position of the phase interface Γ. Arrows represent direction of the velocity field for both phases. The elevated density field (not to scale) is also drawn.

of convergence measured in L_2 and L_∞ error norms behave roughly as $p+1$. However, some deviations from this theoretical rate can be observed which might be explained by the use of the conservative variables set rather than the "p-**v**-T"set. This is further illustrated in Sec. 5.3.

p	n	$\boxed{P}$ L_2	EOC	L_∞	EOC	$\boxed{v_x}$ L_2	EOC	L_∞	EOC
1	20	2.41e-03	**1.6**	1.10e-01	**1.6**	4.18e-05	**2.0**	1.27e-03	**2.6**
	40	6.97e-04	**1.8**	3.34e-02	**1.7**	1.07e-05	**2.0**	3.67e-04	**1.8**
	80	2.06e-04	**1.8**	9.02e-03	**1.9**	2.70e-06	**2.0**	1.12e-04	**1.7**
	160	5.95e-05	**1.8**	2.48e-03	**1.9**	6.82e-07	**2.0**	3.74e-05	**1.6**

2	20	8.87e-06	**2.5**	5.82e-04	**2.1**	3.40e-07	**3.6**	2.13e-05	**2.9**
	40	1.03e-06	**3.1**	6.74e-05	**3.1**	4.88e-08	**2.8**	2.92e-06	**2.9**
	80	1.55e-07	**2.7**	1.94e-05	**1.8**	7.27e-09	**2.7**	4.32e-07	**2.8**
	160	2.80e-08	**2.5**	5.68e-06	**1.8**	1.03e-09	**2.8**	6.87e-08	**2.7**
3	20	6.37e-08	**3.4**	4.67e-06	**3.1**	3.28e-09	**4.0**	1.35e-07	**3.7**
	40	3.18e-09	**4.3**	2.56e-07	**4.2**	2.10e-10	**4.0**	1.02e-08	**3.7**
	80	1.60e-10	**4.3**	1.84e-08	**3.8**	1.33e-11	**4.0**	8.04e-10	**3.7**
	160	1.09e-11	**3.9**	1.52e-09	**3.6**	8.49e-13	**4.0**	5.99e-11	**3.7**
4	10	9.60e-08	**-**	9.90e-06	**-**	5.36e-09	**-**	8.18e-07	**-**
	20	6.59e-10	**7.2**	9.56e-08	**6.7**	1.36e-11	**8.6**	1.33e-09	**9.3**
	40	2.50e-11	**4.7**	4.02e-09	**4.6**	3.94e-13	**5.1**	4.91e-11	**4.8**

Table 5: **Compressible Couette** test case (Tab. 4). Discretization errors and experimental orders of convergence (EOC) in L_2 and L_∞ error norms for polynomial degrees p on **structured quadrilateral** grids of $3 \times n$ elements.

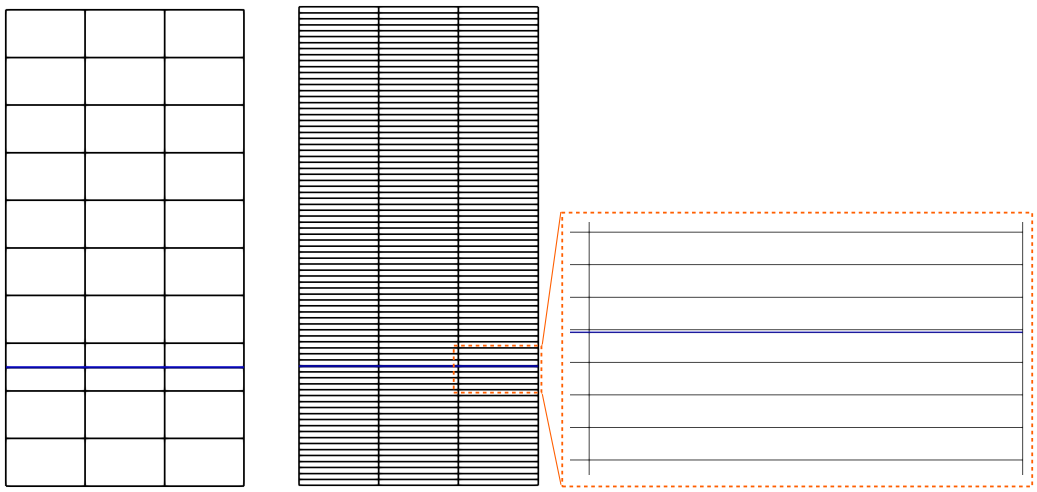


Figure 7: Grids 3×10 (left) and 3×80 (middle) used for the **compressible Couette** test case. The zoom on the interface region (right) shows the proximity of the interface with the element edge.

5.2. Manufactured inclined interface test case

This second test case is built from the method of manufactured solutions. The manufactured solution is first found for an horizontal interface and is

then rotated by 15° to obtain the inclined interface configuration in Fig. 8. Two fluids governed by the perfect gas EOS with different heat capacity ratios and heat capacities at constant volume and ratios of dynamic viscosities and thermal conductivities equal to four are considered. To test the ability of the solver to impose all the different jumps treated in this work, those are all given a value different from zero, except the interface mass flux which is imposed to zero. The HLLCP multiphase Riemann solver with splitting strategy 2 is used.

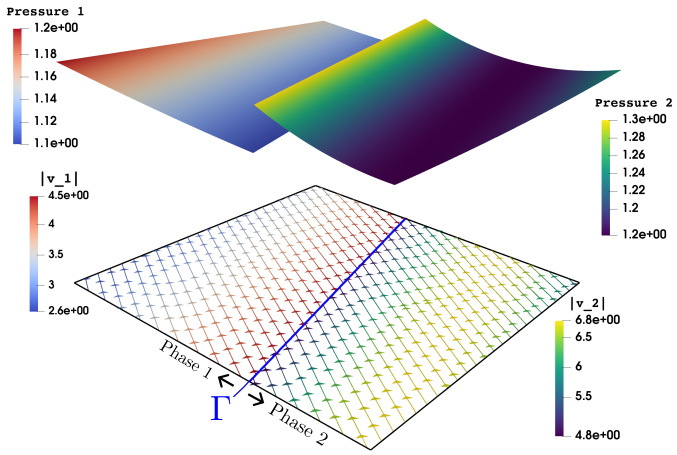


Figure 8: Domain of the **inclined interface** test case with the position of the phase interface Γ. Arrows represent direction of the velocity field for both phases. The elevated pressure field (not to scale) is also drawn.

5.2.1. Grid convergence study

Two of the Cartesian quadrilateral grids used for the convergence study are shown in Fig. 9. Table 6 presents the discretization orders of accuracy on primitive variables. The theoretical $p+1$ convergence rates are here also reached. Note that for $p = 4$, the refinement stops earlier than for the lower polynomial orders because the absolute values of the errors become negligible more quickly. Because the position of the inclined interface is kept unchanged on the various grids considered, it can be concluded that the discrete errors are insensitive to the way the interface intersects the elements on this particular configuration. It shall be demonstrated with the remaining test cases that this observation holds also true for more complex interface geometries.

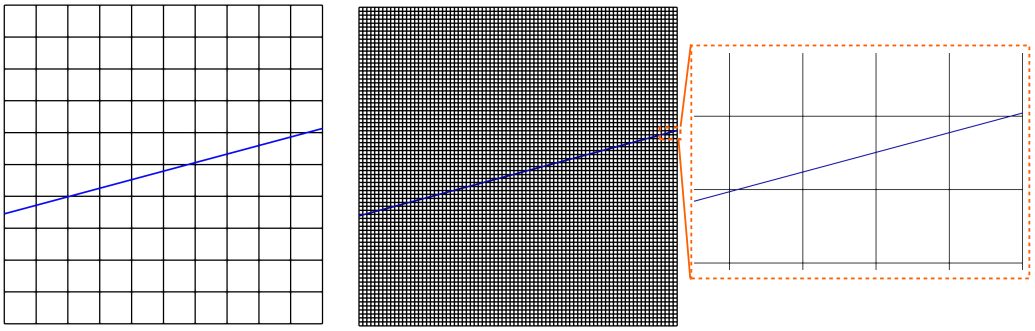


Figure 9: Grids 10×10 (left) and 80×80 (middle) used for the **inclined interface** test case. The zoom on the interface region (right) shows the variety of intersections between the interface and the grid: from normal cuts to (very) small ones.

p	n	$\boxed{P}$ L_2	EOC	L_∞	EOC	$\boxed{T}$ L_2	EOC	L_∞	EOC
2	10	1.12e-04	-	1.08e-03	-	1.79e-04	-	1.08e-03	-
	20	1.41e-05	**3.0**	1.48e-04	**2.9**	2.04e-05	**3.1**	9.53e-05	**3.5**
	40	1.77e-06	**3.0**	1.91e-05	**3.0**	2.49e-06	**3.0**	1.02e-05	**3.2**
	80	2.18e-07	**3.0**	2.26e-06	**3.1**	3.09e-07	**3.0**	1.24e-06	**3.0**
3	10	4.97e-06	-	4.21e-05	-	1.03e-05	-	7.01e-05	-
	20	3.57e-07	**3.8**	5.77e-06	**2.9**	7.22e-07	**3.8**	7.55e-06	**3.2**
	40	2.43e-08	**3.9**	5.31e-07	**3.4**	4.74e-08	**3.9**	6.44e-07	**3.5**
	80	1.47e-09	**4.0**	3.75e-08	**3.8**	2.86e-09	**4.1**	4.39e-08	**3.9**
4	10	3.00e-07	-	7.04e-06	-	6.17e-07	-	6.90e-06	-
	20	9.52e-09	**5.0**	3.15e-07	**4.5**	1.81e-08	**5.1**	2.89e-07	**4.6**
	40	3.06e-10	**5.0**	1.23e-08	**4.7**	5.47e-10	**5.0**	1.08e-08	**4.7**

p	n	$\boxed{v_x}$ L_2	EOC	L_∞	EOC	$\boxed{v_y}$ L_2	EOC	L_∞	EOC
2	10	1.40e-04	-	9.48e-04	-	2.16e-04	-	1.56e-03	-
	20	1.69e-05	**3.1**	1.57e-04	**2.6**	2.60e-05	**3.1**	2.30e-04	**2.8**
	40	2.07e-06	**3.0**	2.03e-05	**2.9**	3.18e-06	**3.0**	3.26e-05	**2.8**
	80	2.48e-07	**3.1**	2.60e-06	**3.0**	3.81e-07	**3.1**	4.17e-06	**3.0**
3	10	5.80e-06	-	1.89e-05	-	8.81e-06	-	2.92e-05	-

	20	3.65e-07	**4.0**	1.13e-06	**4.1**	5.54e-07	**4.0**	1.74e-06	**4.1**
	40	2.29e-08	**4.0**	7.01e-08	**4.0**	3.48e-08	**4.0**	1.06e-07	**4.0**
	80	1.43e-09	**4.0**	4.38e-09	**4.0**	2.17e-09	**4.0**	6.64e-09	**4.0**
4	10	7.36e-08	**-**	7.60e-07	**-**	1.12e-07	**-**	1.21e-06	**-**
	20	2.18e-09	**5.1**	3.14e-08	**4.6**	3.28e-09	**5.1**	4.97e-08	**4.6**
	40	6.66e-11	**5.0**	1.19e-09	**4.7**	1.00e-10	**5.0**	1.86e-09	**4.7**

Table 6: **Inclined interface** test case, **V1** (Tab. 4). Discretization errors and experimental orders of convergence (EOC) in L_2 and L_∞ error norms for polynomial degrees p on **Cartesian quadrilateral** grids of $n \times n$ elements.

5.2.2. Sensitivity analysis

In this second part, some of the numerical parameters listed in Table 4 are modified to conduct an analysis of their influence on the convergence behavior of the solver.

Influence of the interior penalty method choice. Table 7 compares the results obtained with the three different interior penalty methods available. As expected, the $NIPG$ and $IIPG$ methods exhibits a reduced p convergence rates with respect to the L2-norm for even polynomial orders [75, Ch. 1.5]. The $SIPG$ also allows to reach slightly lower errors on the finest grids.

p	n	$\boxed{SIPG}$		$\boxed{NIPG}$		$\boxed{IIPG}$	
		L_2	EOC	L_2	EOC	L_2	EOC
2	10	1.12e-04	**-**	4.13e-04	**-**	2.70e-04	**-**
	20	1.41e-05	**3.0**	1.13e-04	**1.9**	6.89e-05	**2.0**
	40	1.77e-06	**3.0**	3.00e-05	**1.9**	1.78e-05	**2.0**
	80	2.18e-07	**3.0**	7.74e-06	**2.0**	4.54e-06	**2.0**
3	10	4.97e-06	**-**	7.18e-06	**-**	6.20e-06	**-**
	20	3.57e-07	**3.8**	4.91e-07	**3.9**	4.32e-07	**3.8**
	40	2.43e-08	**3.9**	3.26e-08	**3.9**	2.89e-08	**3.9**
	80	1.47e-09	**4.0**	1.98e-09	**4.0**	1.76e-09	**4.0**
4	10	3.00e-07	**-**	4.70e-07	**-**	3.90e-07	**-**
	20	9.52e-09	**5.0**	2.46e-08	**4.3**	1.76e-08	**4.5**
	40	3.06e-10	**5.0**	1.49e-09	**4.0**	9.76e-10	**4.2**

Table 7: **Inclined interface** test case with same parameters as **V1** (Tab. 4) except the choice of Interior Penalty (IP) method. Discretization errors and experimental orders of convergence (EOC) in L_2 error norms for polynomial degrees p on **Cartesian quadrilateral** grids of $n \times n$ elements for the **pressure** field. Comparison between the three different interior penalty methods considered: $SIPG$, $NIPG$, $IIPG$.

Comparison between scalar and tensorial penalty. The second comparison is about the use of scalar and tensorial penalty forms for the discretization of the diffusive fluxes, both in the bulk and at the phase interface. From Table 8, it can be noticed that a sub-optimal p convergence rate is obtained for even polynomial orders for the scalar penalty form. This observation was also made in [69, Sec. 5.1]. Moreover, the errors values are higher with the scalar penalty. Finally, it is important to remark that the safety factor s of the penalty term, Eq. (45), had to be increased by a factor of five to ten compared to the tensorial penalty to obtain convergent results.

p	n	$\boxed{Tensorial}$		$\boxed{Scalar}$	
		L_2	EOC	L_2	EOC
2	10	1.79e-04	**-**	1.42e-03	**-**
	20	2.04e-05	**3.1**	7.45e-04	**0.9**
	40	2.49e-06	**3.0**	2.15e-04	**1.8**
	80	3.09e-07	**3.0**	4.48e-05	**2.3**
3	10	1.03e-05	**-**	2.08e-05	**-**
	20	7.22e-07	**3.8**	8.35e-07	**4.6**
	40	4.74e-08	**3.9**	5.43e-08	**3.9**
	80	2.86e-09	**4.1**	5.61e-09	**3.3**
4	10	6.17e-07	**-**	2.83e-06	**-**
	20	1.81e-08	**5.1**	2.32e-07	**3.6**
	40	5.47e-10	**5.0**	1.51e-08	**3.9**

Table 8: **Inclined interface** test case with same parameters as **V1** (Tab. 4) except the choice of the penalty form. Discretization errors and experimental orders of convergence (EOC) in L_2 error norms for polynomial degrees p on **Cartesian quadrilateral** grids of $n \times n$ elements for the **temperature** field. Comparison between the two different forms of penalty terms considered: scalar and tensorial penalty (Eq. (44)).

5.3. Manufactured spinning bubble test case

This last test case is inspired from [44]. A static fluid bubble is placed at the middle of a squared domain. A manufactured solution is assembled in cylindrical coordinates and then converted to Cartesian coordinates to be passed to the solver. Here real physical conditions are selected that could correspond to a bubble of molten liquid titanium interacting with surrounding air heated around $2000\,K$ (high-temperature effects are however not taken into account). The liquid titanium is almost incompressible with stiffened gas EOS parameters chosen such that the Mach number inside the bubble is of the order of 5×10^{-4}. There are high jumps in properties between the two phases with ratios of densities $\rho_1/\rho_2 \sim 10^5$, of dynamic viscosities $\mu_1/\mu_2 \sim 10^2$ and of thermal conductivities $\lambda_1/\lambda_2 \sim 10^2$. All the jumps are considered different from zero, including the interface mass flux. It can be noted that at the center of the domain, a small empty squared region is added to avoid singularities of the manufactured source terms which is converted from cylindrical to Cartesian coordinates before being fed to the solver. This is shown in Fig. 10.

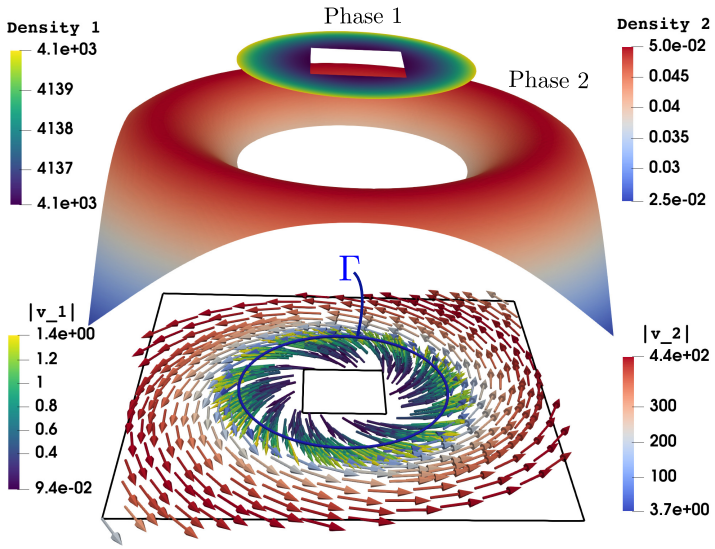


Figure 10: Domain of the **spinning bubble** test case with the position of the phase interface Γ. Arrows represent direction of the velocity field for both phases. The elevated density field (not to scale) is also drawn.

5.3.1. Grid convergence study

Two of the Cartesian triangular grids used for the convergence study are shown in Fig. 11. Table 9 provides the results obtained on Cartesian trian-

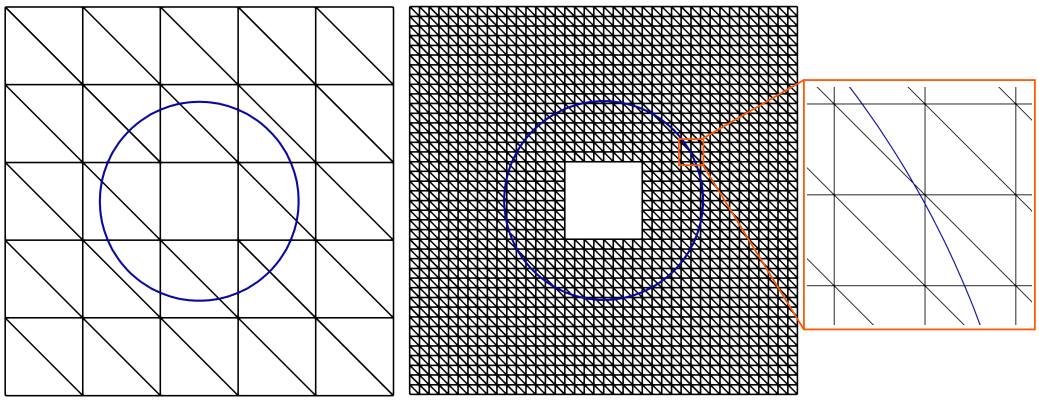


Figure 11: Grids 10×10 (left) and 80×80 (middle) used for the **spinning bubble** test case. The zoom on the interface region (right) shows the variety of intersections between the interface and the grid: from normal cuts to (very) small ones.

gular grids. Due to the geometry considered, the real physical parameters and the stiffness of the models involved, the values of the errors are higher than for the two first test cases. Overall, in the L_2 norm, the targeted $p+1$ convergence rates for the density and vertical velocity and p rates for the heat flux and vorticity are achieved. In the L_∞ norm, this trend is a bit less marked with some rates below or above the expected ones. This is probably due to the stiffness of the models which impacts the accuracy with which the solver can discretize some quantities in different ways. In particular, the treatment of the convective fluxes is very sensitive and critical, as discussed in the first paragraph of Sec. 5.3.2.

p	n	$\boxed{\rho}$				$\boxed{v_y}$			
		L_2	EOC	L_∞	EOC	L_2	EOC	L_∞	EOC
2	10	4.48e-02	-	5.13e-01	-	2.49e+00	-	1.52e+01	-
	20	6.01e-03	**2.9**	1.44e-01	**1.8**	3.17e-01	**3.0**	7.22e+00	**1.1**
	40	5.18e-04	**3.5**	1.23e-02	**3.6**	4.58e-02	**2.8**	7.59e-01	**3.3**
	80	6.11e-05	**3.1**	1.52e-03	**3.0**	5.22e-03	**3.1**	1.65e-01	**2.2**
	160	7.41e-06	**3.0**	2.17e-04	**2.8**	5.83e-04	**3.2**	2.24e-02	**2.9**

3	10	5.28e-03	**-**	6.74e-02	**-**	5.07e-01	**-**	1.04e+01	**-**
	20	6.19e-04	**3.1**	1.94e-02	**1.8**	5.21e-02	**3.3**	2.22e+00	**2.2**
	40	1.61e-05	**5.3**	7.37e-04	**4.7**	2.57e-03	**4.3**	7.03e-02	**5.0**
	80	8.22e-07	**4.3**	4.69e-05	**4.0**	1.84e-04	**3.8**	1.36e-02	**2.4**
	160	4.74e-08	**4.1**	7.72e-06	**2.6**	8.49e-06	**4.4**	8.25e-04	**4.0**
4	10	1.28e+00	**-**	3.29e+01	**-**	1.05e-01	**-**	2.50e+00	**-**
	20	1.59e-04	**13.0**	6.23e-03	**12.4**	6.53e-03	**4.0**	2.01e-01	**3.6**
	40	6.10e-07	**8.0**	3.75e-05	**7.4**	1.83e-04	**5.2**	7.57e-03	**4.7**
	80	1.48e-08	**5.4**	1.04e-06	**5.2**	6.24e-06	**4.9**	5.19e-04	**3.9**
	160	4.58e-10	**5.0**	3.29e-08	**5.0**	1.36e-07	**5.5**	1.23e-05	**5.4**

p	n	$\boxed{q_x}$				$\boxed{\omega}$			
		L_2	EOC	L_∞	EOC	L_2	EOC	L_∞	EOC
2	10	3.98e+01	**-**	3.52e+02	**-**	1.41e+02	**-**	6.76e+02	**-**
	20	1.06e+01	**1.9**	1.02e+02	**1.8**	3.68e+01	**1.9**	2.77e+02	**1.3**
	40	2.86e+00	**1.9**	4.77e+01	**1.1**	1.04e+01	**1.8**	1.57e+02	**0.8**
	80	6.91e-01	**2.0**	1.31e+01	**1.9**	2.56e+00	**2.0**	4.62e+01	**1.8**
	160	1.61e-01	**2.1**	3.34e+00	**2.0**	6.12e-01	**2.1**	1.14e+01	**2.0**
3	10	7.43e+00	**-**	9.17e+01	**-**	4.10e+01	**-**	6.16e+02	**-**
	20	1.40e+00	**2.4**	2.71e+01	**1.8**	6.79e+00	**2.6**	1.77e+02	**1.8**
	40	1.43e-01	**3.3**	3.15e+00	**3.1**	7.79e-01	**3.1**	1.75e+01	**3.3**
	80	1.85e-02	**3.0**	1.20e+00	**1.4**	1.20e-01	**2.7**	5.43e+00	**1.7**
	160	1.94e-03	**3.3**	1.51e-01	**3.0**	1.21e-02	**3.3**	5.46e-01	**3.3**
4	10	2.21e+00	**-**	5.72e+01	**-**	2.37e+01	**-**	1.51e+03	**-**
	20	1.98e-01	**3.5**	1.06e+01	**2.4**	1.20e+00	**4.3**	6.57e+01	**4.5**
	40	9.15e-03	**4.4**	4.46e-01	**4.6**	7.02e-02	**4.1**	2.93e+00	**4.5**
	80	7.12e-04	**3.7**	7.78e-02	**2.5**	4.84e-03	**3.9**	2.99e-01	**3.3**
	160	3.57e-05	**4.3**	5.86e-03	**3.7**	2.51e-04	**4.3**	1.58e-02	**4.2**

Table 9: **Spinning bubble** test case, **V1** (Tab. 4). Discretization errors and experimental orders of convergence (EOC) in L_2 and L_∞ error norms for polynomial degrees p on **Cartesian triangular** grids of $n \times n$ elements. Results for density, y-component of velocity, x-component of heat flux and vorticity ω fields.

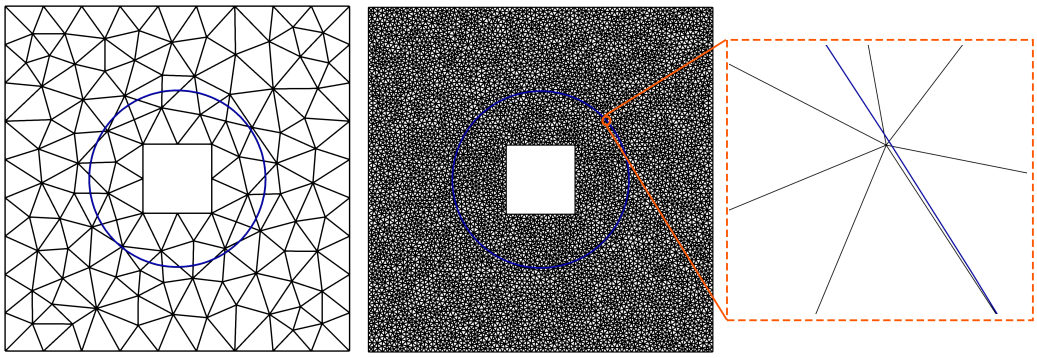
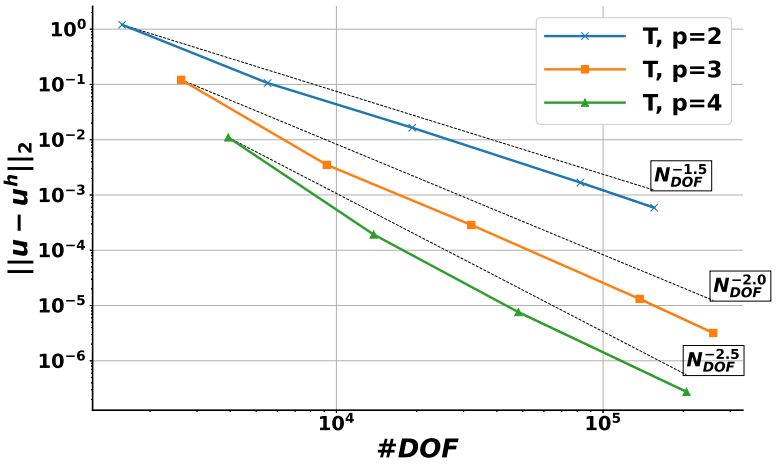


Figure 12: Coarsest (left) and second finest (middle) unstructured triangular grids used for the **spinning bubble** test case. The zoom on the interface region (right) shows critical small cuts intersections.

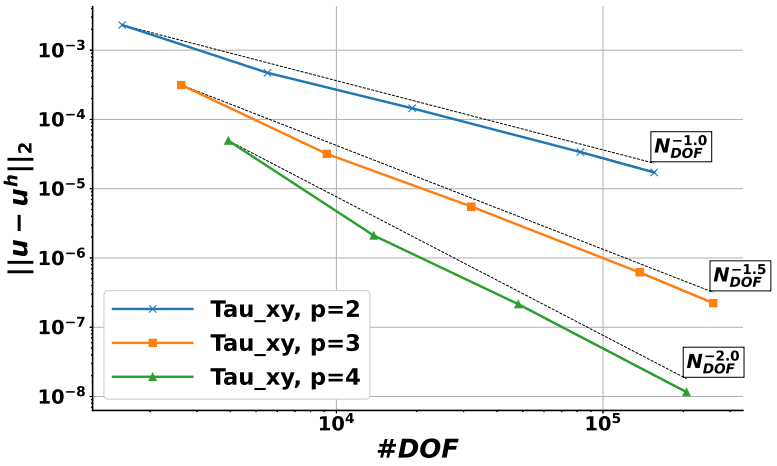
To close this grid convergence study section, the spinning bubble test case is repeated with the same physical and numerical parameters on a set of unstructured triangular grids. Two of them are shown in Fig. 12. Figure 13 shows the L_2 convergence graphs for the temperature field and the xy-component of the shear stress tensor in terms of the number of degrees of freedom (N_{DOF}) on the mesh. The results are in agreement with the theoretical rate $\sim N_{\text{DOF}}^{-(p+1)/2}$ and $\sim N_{\text{DOF}}^{-p/2}$ for the pressure and shear stress tensor, respectively [75]. Those graphs emphasize also the benefits of high-order methods: for the same number of degrees of freedom, the errors are becoming smaller for increasing polynomial orders.

5.3.2. Sensitivity analysis

Influence of the bulk Riemann solver choice. The very low Mach number regime of this situation brings additional numerical challenges linked to the disparity between the convective and acoustic time scales. Thus, using a standard bulk Riemann solver like HLLC [76] on this spinning bubble test case (V1) leads to errors curves that saturate. This behavior is all the more marked for increasing polynomial interpolation orders and finer grids. Saturation is observed from mesh 160×160 for $p = 2$ and from mesh 80×80 for $p = 3$ and $p = 4$. That is why the SLAU2 Riemann solver [77] is selected here for its ability to provide accurate solution over a wide range of Mach numbers.



(a) Temperature field



(b) xy-component of the shear stress tensor $\underline{\tau}$

Figure 13: **Spinning bubble** test case, **V1** (Tab. 4) on **unstructured triangular** grids. Discretization errors in L_2 error norms for different polynomial degrees p as a function of the number of degrees of freedom (DOF). Dashed lines indicate theoretical order of convergence.

Influence of the variables set choice. Solving the spinning bubble test case in terms of conservative variables was found to be impossible as the solver could not converge (keeping the same numerical parameters). This reflects the fact

that for this particular case involving a very stiff phase, the Jacobian of the residual computed with respect to the conservative variables yields a system to be solved which is extremely ill-conditioned. To be able to still perform the analysis, a less stiff second version (without mass transfer) of the spinning bubble test case is considered in Table 10. In this case, the solver is able to make the residual converge to the prescribed tolerance, except for $p = 4$ on the finest grid for which it saturates. As Table 11 reveals, the convergence and the level of accuracy are negatively affected by this new choice of variable. This behavior can be associated to the imposition of the solution jumps by the transpose (Sec. 4.2.2) and penalty terms (Sec. 4.2.3). Since there is no explicit conversion to convert the physical jumps on primitive variables to another variables set, the jumps on the solution vector enforced through conservative variables are only an approximation of the exact prescribed jumps. This discrepancy does not seem to impact too much the solutions accuracy for the first "toy" inclined interface problem but has a non-negligible effect when harsh physical parameters are considered.

		Spinning bubble (V2)
Bulk properties	γ_1	1.0
	$p_{\infty,1}$	1e6
Jumps values (Tab. 2)	$\dot{m}_\Gamma$	0.0
	Δq	100.0
	ΔT	10.0
	Δv_t	-0.5

Table 10: Changes in parameters for the **spinning bubble** test case **V2** compared to V1 in Tab. 4

p	n	$\boxed{p\text{-}\mathbf{v}\text{-}T}$ L_2	EOC	L_∞	EOC	$\boxed{\rho\text{-}\rho\mathbf{v}\text{-}\rho E}$ L_2	EOC	L_∞	EOC
2	10	6.59e+01	-	9.02e+02	-	2.53e+02	-	3.45e+03	-
	20	1.13e+01	**2.5**	2.37e+02	**1.9**	2.78e+01	**3.2**	5.55e+02	**2.6**
	40	7.29e-01	**4.0**	1.70e+01	**3.8**	2.23e+00	**3.6**	7.34e+01	**2.9**

	80	8.35e-02	**3.1**	3.00e+00	**2.5**	1.01e+00	**1.1**	1.37e+02	**0.9**
3	10	5.18e+00	**-**	5.48e+01	**-**	2.24e+02	**-**	1.85e+03	**-**
	20	3.51e-01	**3.9**	9.81e+00	**2.5**	2.55e+00	**6.5**	3.36e+01	**5.8**
	40	1.72e-02	**4.3**	3.58e-01	**4.8**	8.85e-02	**4.8**	6.38e+00	**2.4**
	80	7.91e-04	**4.4**	3.09e-02	**3.5**	1.26e-02	**2.8**	1.93e+00	**1.7**
4	10	9.37e-01	**-**	1.09e+01	**-**	1.63e+03	**-**	1.67e+04	**-**
	20	4.58e-02	**4.4**	7.21e-01	**3.9**	2.14e-01	**12.9**	1.16e+01	**10.5**
	40	9.72e-04	**5.6**	2.02e-02	**5.2**	4.90e-03	**5.4**	2.62e-01	**5.5**
	80	2.09e-05	**5.5**	8.20e-04	**4.6**	5.07e-04	**3.3**	1.15e-01	**1.2**

Table 11: **Spinning bubble V2** (Tab. 10) with different choices of solved-for variables sets. Discretization errors and experimental orders of convergence (EOC) in L_2 error norms for polynomial degrees p on **Cartesian triangular** grids of $n \times n$ elements for the **pressure** field. Comparison between the use of primitive and conservative variables sets.

5.3.3. Flower-like geometry

The spinning bubble test case is finally adapted to a more complex interface geometry, namely a flower (or starfish) shape represented in Fig. 14 and whose equation is given by

$$\varphi_0 \equiv \sqrt{(x-1)^2+(y-1)^2} - 0.06\cos\left(5.0\,\mathrm{atan2}(y-1,x-1)\right) - 0.25625 = 0\ . \tag{52}$$

The properties selected are the same as the ones in Table 4. The analytical solutions and source terms provided by the method of manufactured solution (MMS) are also identical. However, it is important to note that the imposition of the jumps into our solver is adapted for this configuration. Instead of giving as inputs the values of the jumps parameters (Table 2) used during the MMS process, we instead directly evaluate the analytical solution and its gradients along the interface to set the jumps we want to impose on the pressure, the velocities, the temperature and the diffusive fluxes. This means that the physical jumps conditions Eqs. (7) or Eqs. (8) are not respected on the flower-shaped interface, they are strictly respected only on the circular interface. The reason is that we could not manufacture a solution which would satisfy the complex jumps conditions on an interface with an arbitrary shape. But that does not make this test any less interesting since the ability of the

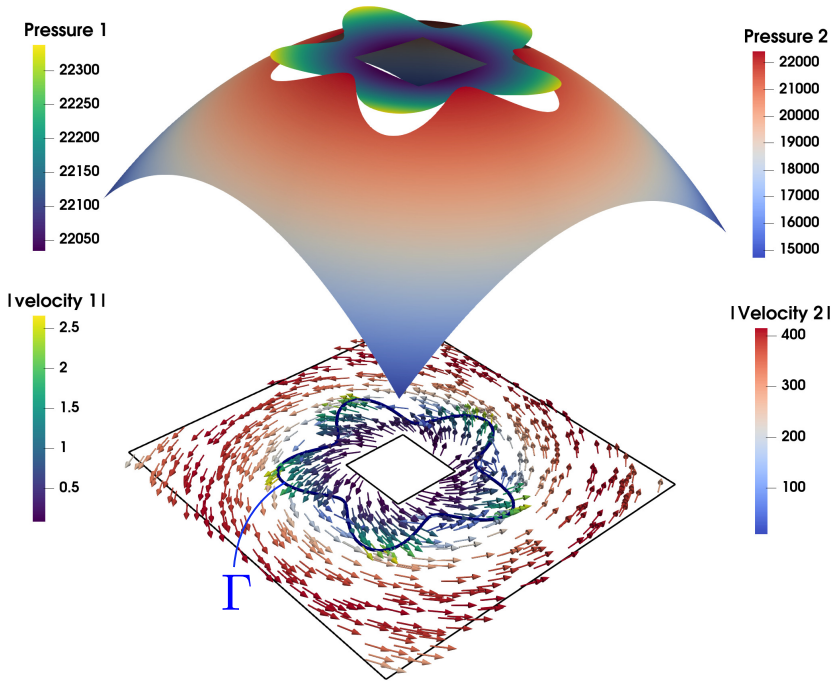


Figure 14: Domain of the **flower shape** test case with the position of the phase interface Γ. Arrows represent direction of the velocity field for both phases. The elevated pressure field (not to scale) is also drawn.

solver to impose prescribed jumps, through the multiphase Riemann solver and the interior penalty method, is still exercised. In order to present the results from a slightly different point of view, the errors are here measured by means of the L_2 norm restricted to the level-set iso-0, with an error for each phase, i.e.

$$\|\mathbf{u}_k - \tilde{\mathbf{u}}_k\|^2_{L_{2,\Gamma(\Omega)}} = \sum_{m \in \mathtt{I}} \int_{\mathtt{I}_m} (\mathbf{u}_k - \tilde{\mathbf{u}}_k)^2 \, \mathrm{d}s \,. \tag{53}$$

Moreover, instead of evaluating the accuracy based on solution variables, the normal projections of the convective and diffusive fluxes along the interface are here selected as error measures. The errors are listed in Table 12. Overall, the errors are decreasing as expected with the refinement of the grids and the superiority of higher order discretizations for comparable number of degrees of freedom manifests itself again. Irregularities in the values of the convergence rates can nonetheless be noticed. Those probably come from the

use of the specific $L_{2,\Gamma}$ error norm (Eq. (53)). It was checked that in terms of the L_2 norm (Eq. (50)), the theoretical convergence rates of $N_{\mathrm{DOF}}^{-(p+1)/2}$ on the primitive variables were indeed retrieved as in the case of the circular interface. The high values of the absolute errors on the normal energy convective fluxes can be explained by the high value of the product between the density and the total enthalpy. This last test case emphasizes the ability of the solver to reach high-order accuracy on non-trivial interface shapes for quantities involving products of solutions variables or their gradients.

p	N_{DOF}	$\boxed{\mathbf{F}^c_{k,2}\cdot\mathbf{n}_\Gamma}$				$\boxed{\mathbf{F}^c_{k,4}\cdot\mathbf{n}_\Gamma}$			
		$k=1$	EOC	$k=2$	EOC	$k=1$	EOC	$k=2$	EOC
3	2620	9.34e+00	-	1.92e+00	-	9.05e+06	-	1.76e+04	-
	9200	2.05e-01	**2.2**	1.11e-01	**2.2**	8.42e+05	**2.4**	6.80e+02	**2.2**
	32080	2.02e-02	**1.8**	7.96e-03	**1.7**	6.71e+04	**1.8**	5.59e+01	**2.2**
	137080	1.43e-03	**1.9**	6.55e-04	**2.1**	4.82e+03	**2.0**	2.15e+00	**2.0**
	259200	3.42e-04	**3.0**	1.61e-04	**2.3**	1.07e+03	**1.9**	5.28e-01	**2.6**
4	3930	3.14e+00	-	4.64e-01	-	2.23e+06	-	4.95e+03	-
	13800	1.73e-02	**2.4**	7.55e-03	**2.2**	8.00e+04	**2.5**	5.52e+01	**2.7**
	48120	1.10e-03	**2.2**	4.38e-04	**2.3**	3.83e+03	**2.4**	1.98e+00	**2.7**
	205620	3.48e-05	**4.1**	1.71e-05	**3.3**	9.79e+01	**2.7**	4.16e-02	**3.6**

p	N_{DOF}	$\boxed{\mathbf{F}^d_{k,2}\cdot\mathbf{n}_\Gamma}$				$\boxed{\mathbf{F}^d_{k,4}\cdot\mathbf{n}_\Gamma}$			
		$k=1$	EOC	$k=2$	EOC	$k=1$	EOC	$k=2$	EOC
3	2620	3.87e-04	-	3.42e-03	-	8.57e+00	-	1.81e+01	-
	9200	6.54e-05	**1.4**	4.24e-04	**1.3**	4.10e+00	**1.9**	6.50e+00	**1.1**
	32080	1.29e-05	**1.2**	6.34e-05	**1.7**	3.06e-01	**1.0**	5.97e-01	**2.1**
	137080	2.20e-06	**1.3**	5.41e-06	**1.5**	6.76e-02	**2.1**	2.83e-02	**1.9**
	259200	8.95e-07	**1.4**	2.38e-06	**1.7**	1.98e-02	**0.6**	1.38e-02	**0.8**
4	3930	2.02e-04	-	1.98e-03	-	5.99e+00	-	8.63e+00	-
	13800	1.01e-05	**2.1**	4.69e-05	**2.1**	5.59e-01	**1.9**	9.98e-01	**2.4**
	48120	1.52e-06	**1.5**	3.13e-06	**2.2**	3.25e-02	**2.3**	3.11e-02	**2.8**
	205620	7.37e-08	**2.4**	1.43e-07	**3.0**	2.03e-03	**1.9**	8.96e-04	**1.7**

Table 12: **Flower shape** test case. Discretization errors and experimental orders of convergence (EOC) in the $L_{2,\Gamma}$ error norm for polynomial degrees p on **unstructured triangular** meshes containing an increasing number of degrees of freedom (N_{DOF}). Results per phase k for the interface normal projection of the x-momentum ($\mathbf{F}_{k,2}$) and total energy ($\mathbf{F}_{k,4}$) components of the convective and diffusive fluxes.

5.4. Additional comments about other parameters

The results section is closed by providing some further analyses about the influence of some other numerical parameters:

- Geometrical robustness: it was found empirically that an agglomeration threshold value $\delta_{\mathrm{agg}} \in [0.2, 0.3]$ was suitable to ensure convergence of all the simulations run in this work without impacting the accuracy of the solver. A similar range is mentioned in [27]. It was noticed that using too low thresholds could potentially lead to a degradation of the convergence rates and/or divergence of the non-linear iterations. As far as the Nitsche's penalty term is concerned, the stability parameter in Eq. (45) combined with a geometrical factor based on the agglomerated topology was appropriate to make all the simulations converge. However, as noted before, the safety factor associated to the penalty term had to be increased to guarantee convergence of simulations relying on scalar penalty.

- Viscous weighting: the inclined interface test case was modified such that a ratio of dynamic viscosities of the order of 10^4 was set. Whether in this configuration or the one of the spinning bubble test case (also involving important jumps in transport properties), only marginal effects of the viscous weighting, by varying the a exponent in Eq. (35), on the convergence behavior of the non-linear solver could be noticed. No impact on the accuracy of the results could be observed.

- Splitting strategy: both splitting strategies (Table 3) and so both the HLLC and HLLCP multiphase Riemann solvers (because they are linked) were tested on the inclined interface test case and the version 2 of the spinning bubble test case without interface mass transfer. No appreciable differences were observed. This is expected since for a zero net interface mass flux, $\dot{m}_\Gamma = 0$, the jumps on convective and diffusive fluxes become the same between the two splittings. Considering

mass transfer effects may lead to a different conclusion. Unfortunately, we were not able to manufacture a solution complying with the jumps conditions involved in splitting 2 when $\dot{m}_\Gamma \neq 0$. That is why only the splitting 1 with mass transfer was tested here.

6. Conclusions and outlook

We have established a high-order numerical framework able to sharply discretize a steady two-phase compressible Navier-Stokes system closed by general equations of state. The interface compatibility conditions considered include, through constant imposed parameters, most of the jumps that can arise in real physical configurations, providing the solver with the potential of treating a wide variety of interfacial phenomena. From the literature review, we have identified the extended discontinuous Galerkin (XDG) method as being the ideal candidate to reach the targeted objective. We have derived a new weak formulation for XDG relying on multiphase Riemann solvers for the interface convective fluxes and a revised interior penalty method for the diffusive fluxes. The novelty of the proposed approach is the building of a bridge between two related but not necessarily connected research areas. On the one hand, the modeling aspects and the multiphase Riemann solvers are inspired from previous work using FVM or high-order cut-cell and ALE methods to discretize two-phase Euler (for most of them) or Navier-Stokes equations. On the other hand, the discretization techniques are borrowed from the geometrically unfitted FE-type community in which most of the work deal with reduced systems of equations (principally of elliptic nature) or two-phase incompressible flows.

Through the design of appropriate reference solutions on different configurations, the implementation has been verified. A sensitivity analysis about various numerical parameters and methods has shed the light on the best combination among the ones available. It was found that when the p-$\mathbf{v}$-T variables are employed such that the jumps on the solutions can be exactly imposed, all-speed bulk Riemann solvers are used when needed and SIPG method combined with tensorial penalty is applied, the solver is optimal order accurate, showing $p+1$ convergence rates in both L_2 and L_∞ norms. This conclusion holds both for the regular "gas-gas"-like compressible situation and the quite harsh "gas-liquid" system combining high properties jumps and stiff models. Moreover, a test case with a flower-shaped interface has been designed to exhibit the potential of our approach to deliver high-

order accuracy for interface fluxes on complex geometries. Those features position the present solver as an ideal tool to study (extremely) high contrast interface problems involving phases with very distinct properties. The cell-agglomeration procedure and external high-order quadrature libraries for implicitly-defined domains support the ability of the proposed approach to seamlessly work on different kinds of grids (structured quadrilateral and triangular grids as well as unstructured triangular grids, as shown in the test cases), irrespective of the way the interface intersects the underlying mesh. This makes it an appropriate candidate to be applied to (evolving) intricate geometries in combination with mesh adaptation.
The focus of the present work has been put on steady test cases and spatial accuracy. Although the robustness of the proposed tool with respect to the magnitude of the jumps and the cut-cell scenarios has been exercised on several configurations, additional studies would probably be required to fully certify its adequacy to any situations that can be encountered in the field of separated compressible two-phase flows. Future research should also concentrate on the coupling of the XDG solver with the interface transport scheme and investigate its temporal accuracy in the context of moving interface problems. Further developments include the extension to solution dependent properties, the computation of surface tension effects by means of appropriate interface curvature filtering strategies and the simulation of more test cases involving physically consistent interface mass flux models with the goal of comparing the two splitting strategies in this scenario. Finally, the extension of the framework to three-dimensional applications is a logical next step for which we do not expect any particular difficulties.

Acknowledgements

The first author is supported by the Fonds de la Recherche Scientifique (FNRS) of Belgium, under a FRIA grant. Computational resources have been provided by the supercomputing facilities of the Université catholique de Louvain (CISM UCL) and the Consortium des Équipements de Calcul Intensif en Fédération Wallonie Bruxelles (CÉCI) funded by the Fond de la Recherche Scientifique de Belgique (F.R.S.-FNRS) under convention 2.5020.11 and by the Walloon Region.

References

[1] R. Saurel, C. Pantano, Diffuse-interface capturing methods for compressible two-phase flows, Annual Review of Fluid Mechanics 50 (2018).

[2] V. Maltsev, M. Skote, P. Tsoutsanis, High-order methods for diffuse-interface models in compressible multi-medium flows: A review, Physics of Fluids 34 (2022) 021301.

[3] R. W. Houim, K. K. Kuo, A ghost fluid method for compressible reacting flows with phase change, Journal of Computational Physics 235 (2013) 865–900.

[4] S. Majidi, A. Afshari, A ghost fluid method for sharp interface simulations of compressible multiphase flows, Journal of Mechanical Science and Technology 30 (2016) 1581–1593.

[5] P. Das, H. Udaykumar, A sharp-interface method for the simulation of shock-induced vaporization of droplets, Journal of Computational Physics 405 (2020) 109005.

[6] Y.-L. Liu, C.-W. Shu, A.-M. Zhang, Weighted ghost fluid discontinuous galerkin method for two-medium problems, Journal of Computational Physics 426 (2021) 109956.

[7] L. Xu, W. Yang, T. Liu, An interface treatment for two-material multi-species flows involving thermally perfect gases with chemical reactions, Journal of Computational Physics 448 (2022) 110707.

[8] C.-H. Chang, X. Deng, T. G. Theofanous, Direct numerical simulation of interfacial instabilities: a consistent, conservative, all-speed, sharp-interface method, Journal of Computational Physics 242 (2013) 946–990.

[9] J. Luo, X. Hu, N. A. Adams, A conservative sharp interface method for incompressible multiphase flows, Journal of Computational Physics 284 (2015) 547–565.

[10] J.-Y. Lin, Y. Shen, H. Ding, N.-S. Liu, X.-Y. Lu, Simulation of compressible two-phase flows with topology change of fluid–fluid interface by a robust cut-cell method, Journal of Computational Physics 328 (2017) 140–159.

[11] X.-L. Deng, M. Li, Simulating compressible two-medium flows with sharp-interface adaptive runge–kutta discontinuous galerkin methods, Journal of Scientific Computing 74 (2018) 1347–1368.

[12] Y. Shen, Y. Ren, H. Ding, A 3d conservative sharp interface method for simulation of compressible two-phase flows, Journal of Computational Physics 403 (2020) 109107.

[13] T. Long, J. Cai, S. Pan, An accelerated conservative sharp-interface method for multiphase flows simulations, Journal of Computational Physics 429 (2021) 110021.

[14] V.-T. Nguyen, J. Peraire, B. C. Khoo, P.-O. Persson, A discontinuous galerkin front tracking method for two-phase flows with surface tension, Computers & Fluids 39 (2010) 1–14.

[15] A. J. Barlow, P.-H. Maire, W. J. Rider, R. N. Rieben, M. J. Shashkov, Arbitrary lagrangian–eulerian methods for modeling high-speed compressible multimaterial flows, Journal of Computational Physics 322 (2016) 603–665.

[16] C. Chalons, C. Rohde, M. Wiebe, A finite volume method for undercompressive shock waves in two space dimensions, ESAIM: Mathematical Modelling and Numerical Analysis 51 (2017) 1987–2015.

[17] Y. Zhang, A. Chandra, F. Yang, E. Shams, O. Sahni, M. Shephard, A. A. Oberai, A locally discontinuous ale finite element formulation for compressible phase change problems, Journal of Computational Physics 393 (2019) 438–464.

[18] N. Potghan, C. S. Jog, An ale-based finite element strategy for modeling compressible two-phase flows, International Journal for Numerical Methods in Fluids (2022).

[19] F. de Prenter, C. Verhoosel, H. van Brummelen, M. Larson, S. Badia, Stability and conditioning of immersed finite element methods: analysis and remedies, arXiv preprint arXiv:2208.08538 (2022).

[20] B. Schott, Stabilized cut finite element methods for complex interface coupled flow problems, Ph.D. thesis, Technische Universität München, 2017.

[21] E. Burman, Ghost penalty, Comptes Rendus Mathematique 348 (2010) 1217–1220.

[22] E. Burman, S. Claus, P. Hansbo, M. G. Larson, A. Massing, Cutfem: discretizing geometry and partial differential equations, International Journal for Numerical Methods in Engineering 104 (2015) 472–501.

[23] S. Badia, F. Verdugo, A. F. Martín, The aggregated unfitted finite element method for elliptic problems, Computer Methods in Applied Mechanics and Engineering 336 (2018) 533–553.

[24] P. Bastian, C. Engwer, An unfitted finite element method using discontinuous galerkin, International journal for numerical methods in engineering 79 (2009) 1557–1576.

[25] F. Heimann, C. Engwer, O. Ippisch, P. Bastian, An unfitted interior penalty discontinuous galerkin method for incompressible navier–stokes two-phase flow, International Journal for Numerical Methods in Fluids 71 (2013) 269–293.

[26] F. Kummer, Extended discontinuous galerkin methods for two-phase flows: the spatial discretization, International Journal for Numerical Methods in Engineering 109 (2017) 259–289.

[27] B. Müller, S. Krämer-Eis, F. Kummer, M. Oberlack, A high-order discontinuous galerkin method for compressible flows with immersed boundaries, Internal Journal For Numerical Methods in Engineering 110 (2017) 3–30.

[28] C. Gürkan, A. Massing, A stabilized cut discontinuous galerkin framework for elliptic boundary value and interface problems, Computer Methods in Applied Mechanics and Engineering 348 (2019) 466–499.

[29] A. Main, G. Scovazzi, The shifted boundary method for embedded domain computations. part ii: Linear advection–diffusion and incompressible navier–stokes equations, Journal of Computational Physics 372 (2018) 996–1026.

[30] N. M. Atallah, C. Canuto, G. Scovazzi, The high-order shifted boundary method and its analysis, Computer Methods in Applied Mechanics and Engineering 394 (2022) 114885.

[31] R. I. Saye, High-order quadrature on multi-component domains implicitly defined by multivariate polynomials, Journal of Computational Physics 448 (2022) 110720.

[32] F. Chouly, A review on some discrete variational techniques for the approximation of essential boundary conditions, 2022. Preprint.

[33] E. Burman, P. Zunino, Numerical approximation of large contrast problems with the unfitted nitsche method, in: Frontiers in Numerical Analysis-Durham 2010, Springer, 2011, pp. 227–282.

[34] M. Winter, B. Schott, A. Massing, W. A. Wall, A nitsche cut finite element method for the oseen problem with general navier boundary conditions, Computer Methods in Applied Mechanics and Engineering 330 (2018) 220–252.

[35] C. Annavarapu, M. Hautefeuille, J. E. Dolbow, A robust nitsche's formulation for interface problems, Computer Methods in Applied Mechanics and Engineering 225 (2012) 44–54.

[36] B. Schott, S. Shahmiri, R. Kruse, W. Wall, A stabilized nitsche-type extended embedding mesh approach for 3d low-and high-reynolds-number flows, International Journal for Numerical Methods in Fluids 82 (2016) 289–315.

[37] C. Gurkan, S. Sticko, A. Massing, Stabilized cut discontinuous galerkin methods for advection-reaction problems, SIAM Journal on Scientific Computing 42 (2020) A2620–A2654.

[38] C. Engwer, S. May, A. Nüßing, F. Streitbürger, A stabilized dg cut cell method for discretizing the linear transport equation, SIAM Journal on Scientific Computing 42 (2020) A3677–A3703.

[39] J. Glimm, X. L. Li, Y. Liu, N. Zhao, Conservative front tracking and level set algorithms, Proceedings of the National Academy of Sciences 98 (2001) 14198–14201.

[40] E. Franquet, V. Perrier, Runge–kutta discontinuous galerkin method for the approximation of baer and nunziato type multiphase models, Journal of Computational Physics 231 (2012) 4096–4141.

[41] E. Gaburro, W. Boscheri, S. Chiocchetti, C. Klingenberg, V. Springel, M. Dumbser, High order direct arbitrary-lagrangian-eulerian schemes on moving voronoi meshes with topology changes, Journal of Computational Physics 407 (2020) 109167.

[42] C. Gürkan, M. Kronbichler, S. Fernández-Méndez, extended hybridizable discontinuous galerkin for incompressible flow problems with unfitted meshes and interfaces, International Journal for Numerical Methods in Engineering 117 (2019) 756–777.

[43] E. Burman, M. Cicuttin, G. Delay, A. Ern, An unfitted hybrid high-order method with cell agglomeration for elliptic interface problems, SIAM Journal on Scientific Computing 43 (2021) A859–A882.

[44] R. Nourgaliev, P. T. Greene, S. P. Schofield, Physics-based reconstruction of interfacial jump conditions in all-speed multi-fluid dynamics, in: Fluids Engineering Division Summer Meeting, volume 58059, American Society of Mechanical Engineers, 2017, p. V01BT11A017.

[45] S. Jöns, C. Müller, J. Hintz, A. Beck, C.-D. Munz, A viscous and heat conducting ghost fluid method for multi-fluid simulations, arXiv preprint arXiv:2305.10278 (2023).

[46] W. Sollie, J. J. van der Vegt, O. Bokhove, A space-time discontinuous galerkin finite element method for two-fluid problems, Journal of Computational Physics 141 (2006) 46–77.

[47] W. E. H. Sollie, O. Bokhove, J. J. van der Vegt, Space–time discontinuous galerkin finite element method for two-fluid flows, Journal of computational physics 230 (2011) 789–817.

[48] C. Engwer, F. Heimann, Dune-udg: A cut-cell framework for unfitted discontinuous galerkin methods, in: Advances in DUNE, Springer, 2012, pp. 89–100.

[49] R. Saye, Implicit mesh discontinuous galerkin methods and interfacial gauge methods for high-order accurate interface dynamics, with applications to surface tension dynamics, rigid body fluid–structure interaction, and free surface flow: Part i, Journal of Computational Physics 344 (2017) 647–682.

[50] M. Smuda, Direct Numerical Simulation of Multi-Phase Flows using Extended Discontinuous Galerkin Methods, Ph.D. thesis, Technische Universität Darmstadt, 2021.

[51] F. Kummer, J. Weber, M. Smuda, Bosss: A package for multigrid extended discontinuous galerkin methods, Computers & Mathematics with Applications 81 (2021) 237–257.

[52] V. Boniou, On the numerical simulation of evaporating two-phase flows using sharp interface capturing methods, Ph.D. thesis, Université Paris-Saclay, 2021.

[53] M. Ishii, T. Hibiki, Thermo-fluid dynamics of two-phase flow, Springer Science & Business Media, 2010.

[54] M. Bond, H. Struchtrup, Mean evaporation and condensation coefficients based on energy dependent condensation probability, Physical Review E 70 (2004) 061605.

[55] R. Hartmann, Numerical analysis of higher order discontinuous galerkin finite element methods, in: D. H. (Ed.), 35th CFD / ADIGMA Course on Very High-Order Discretization Methods, von Karman Institute for Fluid Dynamics, 2008, pp. 1–107.

[56] B. Müller, F. Kummer, M. Oberlack, Highly accurate surface and volume integration on implicit domains by means of moment-fitting, International Journal for Numerical Methods in Engineering 96 (2013) 512–528.

[57] P. Schrooyen, J.-S. Cagnogne, N. Poletz, K. Hillewaert, A high-order extended discontinuous galerkin method for moving immersed interface problem, 2017. XDMS Conference.

[58] R. Saye, Algoim (v2), 2022-11-16. URL: `https://algoim.github.io/`.

[59] T. Cui, W. Leng, H. Liu, L. Zhang, W. Zheng, High-order numerical quadratures in a tetrahedron with an implicitly defined curved interface, ACM Transactions on Mathematical Software (TOMS) 46 (2020) 1–18.

[60] T. Cui, W. Leng, H. Liu, L. Zhang, W. Zheng, Phg (0.9.7), 2022-11-16. URL: `http://lsec.cc.ac.cn/phg/download/`.

[61] P. Schrooyen, L. Arbaoui, J.-S. Cagnone, N. Poletz, K. Hillewaert, A high-order extended discontinuous galerkin method to treat hydrodynamics problems, 2018. URL: http://congress.cimne.com/eccm_ecfd2018/admin/files/fileabstract/a479.pdf, eCCM-ECFD Conference.

[62] S. Fechter, F. Jaegle, V. Schleper, Exact and approximate riemann solvers at phase boundaries, Computers & Fluids 75 (2013) 112–126.

[63] G. Gassner, F. Lörcher, C.-D. Munz, A contribution to the construction of diffusion fluxes for finite volume and discontinuous galerkin schemes, Journal of Computational Physics 224 (2007) 1049–1063.

[64] S. Jöns, C.-D. Munz, An approximate riemann solver for advection–diffusion based on the generalized riemann problem, Communications on Applied Mathematics and Computation 2 (2020) 515–539.

[65] S. Fechter, C.-D. Munz, C. Rohde, C. Zeiler, Approximate riemann solver for compressible liquid vapor flow with phase transition and surface tension, Computers & Fluids 169 (2018) 169–185.

[66] S. Fechter, Compressible multi-phase simulation at extreme conditions using a discontinuous Galerkin scheme, Ph.D. thesis, Universitat Stuttgart, 2015.

[67] S. Jöns, C.-D. Munz, Riemann solvers for phase transition in a compressible sharp-interface method, Applied Mathematics and Computation 440 (2023) 127624.

[68] R. Saye, Fast multigrid solution of high-order accurate multiphase stokes problems, Communications in Applied Mathematics and Computational Science 15 (2020) 147–196.

[69] R. Hartmann, P. Houston, An optimal order interior penalty discontinuous galerkin discretization of the compressible navier–stokes equations, Journal of Computational Physics 227 (2008) 9670–9685.

[70] K. Hillewaert, Development of the discontinuous Galerkin method for high-resolution, large scale CFD and acoustics in industrial geometries, Ph.D. thesis, Université catholique de Louvain, 2013.

[71] N. Levaux, A. Bilocq, P. Schrooyen, V. Terrapon, K. Hillewaert, Interior penalty method for a cartesian discontinuous galerkin solver with immersed boundaries, 2023. URL: https://orbi.uliege.be/bitstream/2268/302551/2/CFC2023_abstract.pdf, iACM Computational Fluids Conference.

[72] D. Henneaux, PyManufSol, https://github.com/henneauxd/PyManufSol, 2023. doi:10.5281/zenodo.1234, v1.0.0.

[73] D. Henneaux, P. Schrooyen, P. Chatelain, T. Magin, A general methodology for symbolically generating manufactured solutions satisfying prescribed conditions - application to two-phase flows equations, in: 6th International Conference on Numerical and Symbolic Computation, ECCOMAS Thematic Conference, 2023, p. ...

[74] S. Peluchon, G. Gallice, L. Mieussens, Development of numerical methods to simulate the melting of a thermal protection system, Journal of Computational Physics 448 (2022) 110753.

[75] B. Rivière, Discontinuous Galerkin methods for solving elliptic and parabolic equations: theory and implementation, SIAM, 2008.

[76] E. F. Toro, The hllc riemann solver, Shock Waves 29 (2019) 1065–1082.

[77] K. Kitamura, E. Shima, Towards shock-stable and accurate hypersonic heating computations: A new pressure flux for ausm-family schemes, Journal of Computational Physics 245 (2013) 62–83.