AIAA AVIATION Forum
June 15-19, 2020, VIRTUAL EVENT
AIAA AVIATION 2020 FORUM

10.2514/6.2020-2971

Extended Discontinuous Galerkin Method for Solving Gas-Liquid Compressible Flows with Phase Transition

David Henneaux*
von Karman Institute for Fluid Dynamics, Rhode-Saint-Genèse, Belgium, 1640

Pierre Schrooyen †
Cenaero, Gosselies, Belgium, 6041

Bruno Dias‡ and Alessandro Turchi§
von Karman Institute for Fluid Dynamics, Rhode-Saint-Genèse, Belgium, 1640

Philippe Chatelain¶
Université catholique de Louvain, Louvain-la-Neuve, Belgium, 1348

Thierry Magin‖
von Karman Institute for Fluid Dynamics, Rhode-Saint-Genèse, Belgium, 1640

The sharp-interface resolution of compressible gas-liquid flows is made particularly difficult by the coupling between the hydrodynamics and the thermodynamics, the stiff equation of state in the liquid phase, the need to represent important discontinuities in the fluid properties, and the modeling of complex interfacial phenomena. In this work, a high-order discontinuous Galerkin method with a level-set interface tracking method is used to handle the moving sharp interface. It is combined with multiphase Riemann solvers to impose the physical jump compatibility relations, with the possibility of modeling phase transition effects. The proposed approach is then tested on some verification test cases showing its potential for future applications.

I. Nomenclature

DGM	=	Discontinous Galerkin Method
FVM	=	Finite Volume Method
FEM	=	Finite Element Method
EOS	=	Equation of State
XDGM	=	eXtended Discontinuous Galerkin Method

II. Introduction

Multiphase flows have become increasingly important in the last decades in a wide variety of engineering applications for their optimum design and safe operations. The mathematical representation and the numerical treatment of this kind of flows are made much more complicated than for single phase flows due to the existence of deformable and moving interfaces. Across those significant discontinuities of fluid properties exist, usually accompanied by phase transitions and complex flow features. While most commercial solvers today rely on the assumption of incompressible multiphase flows, high temperatures and/or high pressures can be encountered which violate this hypothesis. These

*Ph.D. Student, Aeronautics and Aerospace Department, Chaussée de Waterloo 72.
†Senior Research Engineer, Cenaero, Rue des Frères Wright 29, AIAA Member.
‡Ph.D. Student, Aeronautics and Aerospace Department, Chaussée de Waterloo 72, AIAA Student Member.
§Senior Research Engineer, Aeronautics and Aerospace Department, Chaussée de Waterloo 72, AIAA Member.
¶Professor, Institute of Mechanical, Materials and Civil Engineering, Place du Levant 2, AIAA Member.
‖Professor, Aeronautics and Aerospace Department, Chaussée de Waterloo 72, AIAA Member.

conditions can be found in various engineering applications such as fuel injectors, nuclear reactors, rocket motors or gas turbines and heat pumps [1], for which accurate modeling and simulation are crucial. Another important field in which compressible multiphase models are required concerns the prediction of the hydrodynamics of melting materials, such as metallic alloys or glassy materials, surrounded by an external gas. In the manufacturing sector, whether in metallic additive manufacturing [2] or laser-cutting of steel [3], the interaction between the surrounding gas and the melt layer can strongly affect the final manufacturing quality of the component. The ablation of metallic space debris or meteors is another topic of great interest nowadays that requires the development of reliable numerical tools to assess the threat posed by those reentering objects [4]. The very important disparities in the physical properties between the gas and the molten liquid in such situations increase even more the numerical challenges associated to multiphase flows. As the present work is part of a long-term project whose ambition is the development of an unified numerical tool for producing high-fidelity simulations of melting materials, gas-liquid flow configurations are of particular interest.

The treatment of multiphase compressible flows brings additional difficulties compared to the treatment of their incompressible counterparts. Indeed, hydrodynamics and thermodynamics are strongly coupled through equations of state (EOS) in the bulk of each phase (typically the perfect gas EOS for a gas and a more elaborated stiff EOS for a liquid). The main difficulty is therefore related to the elaboration of a consistent numerical strategy to treat the discontinuous nature of the interface together with the related physical interface processes such as surface tension or phase change [5]. Two approaches can be identified to tackle these challenges: diffuse and sharp interface methods. In the first approach, which is the most common one, a set of averaged equations governing mean values of flow quantities are considered. Many variants that are often derived from the Baer-Nunziato model [6] exist. The averaging procedure introduces additional terms to be modeled that represent the interfacial transfer mechanisms between the phases. In order to avoid spurious oscillations near the interface due to the mixing between the phases, various specialized schemes have been developed, like the one in [7].

In the case of dispersed two-phase flows where one phase is scattered in the other (e.g. for fuel injectors), the averaging approach is the only possible one. However, for separated (immiscible) multiphase flows in which clear material interfaces can be identified (e.g. for melting materials surrounded by a hot gas), the use of this technique can lead to an important loss of accuracy in the representation of the interfacial phenomena, due to the smearing introduced by the averaging. In this situation, it is preferable to have a sharp representation in order to model with accuracy the phenomena driving the physics of the interface. To this end, the following general ingredients are needed:

- A method to represent and follow the interface: contrary to averaging approaches identifying implicitly the interface position with a volume fraction, the interface needs to be explicitly represented in the context of sharp interface methods. Three different computational frameworks have been developed in this regard: (i) Lagrangian, (ii) Eulerian, and (iii) Arbitrary Lagrangian-Eulerian (ALE) methods. In the Lagrangian description, the mesh moves and is kept aligned with the material interface. For interfaces undergoing significant topological changes, this approach might suffer from a loss of accuracy and numerical instabilities. In the Eulerian framework in which the mesh is kept fixed, the volume-of-fluid and the level-set techniques are widely used. They both solve an advection equation for a scalar quantity from which the interface position can be determined. The difficulty is then to preserve the resolution of the scheme in the cells cut by the interface. The ALE approach tries to combine the best features of the two previous methods by moving the mesh in some arbitrarily specified way to give a continuous rezoning capability [8].
- A method to handle discontinuous solutions: the discontinuities across material interfaces can naturally be represented with a moving mesh approach for Finite Volume (FVM) [9] or Disontinuous Galerkin (DGM) [10] methods. On the other hand, if the mesh does not conform with the interface, a strategy to represent discontinuous solutions inside cut cells is needed. In the FVM context, the class of ghost fluid schemes has been widely studied, in which ghost values are extrapolated on both sides of the interface to be able to define two different fluxes at the cut cell boundaries ([11]). In the FV cut-cell method, when a mesh cell is passed across by a material interface, it is separated to smaller cells by adaptative mesh refinement. The interface is then represented by piecewise cut faces across which the fluxes are computed ([12]). Both can also be combined as it is done in [13].

  In the Finite Element Method (FEM) community, a popular approach is the extended FEM (XFEM) where the approximation space is enriched with new basis functions containing information about the discontinuities to be represented ([14]).

  Finally in the DGM framework, several variants of unfitted/ cut-cell/ extended methods have been proposed recently, taking inspiration from previous work on the XFEM. In [15], an implicitly defined mesh is constructed based on the intersection between a simplistic background Cartesian grid and the iso-contour of a level-set while cells with small volume fraction are merged with nearby cells. The nodal basis for this implicitly mesh corresponds

to the one of the background mesh, meaning that some interpolation nodes can be located "outside" of the newly defined elements. With the same idea, separate degrees of freedom are considered in cut cells in [16] by extending the initial DG functional space. In this case, a cell-agglomeration procedure is proposed to address the problem of small cut cells. The last example is about an extended hybridizable DGM that adds a trace variable on the material interface in the weak formulation [17]. Doing so, the standard nodal basis is kept, and the solution is therefore not "duplicated" on cut elements. Common to those different methods is the necessity of evaluating accurately integrals on implicitly defined domains.
Worth mentioning is the use in [18] of a DGM for the bulk equations in combination with a local FVM in subdived cut cells.

- A method to resolve the jump conditions: the modeling of the interfacial physicial phenomena relies on the enforcement of jump conditions coupling the field equations valid in the bulk of each phase. For all the aforementioned methods, this is accomplished by defining appropriate numerical fluxes across the material interface (and by subsequently integrating them over the interface in the weak formulation). To find fluid states on both sides of the interface in a ghost-fluid type approach, the situation can be seen as a multiphase Riemann problem [11]. Different exact or approximate solvers able to include surface tension and mass/energy transfer in a thermodynamically consistent way have been designed. Alternatively, sided extrapolation whose polynomial coefficients are determined based on the jumps can be employed for cut-cell FVM [12].
  It is interesting to remark that for DGM dealing with incompressible two-phase flows without phase transition, the interface conditions are usually directly incorporated in the variational formulation [16]. At convergence, those additional terms vanish by satisfying the jump relations. Another worth mentioning approach in the context of moving DGM is the use of separate test functions for the discretization of the conservation law and the interface conditions [10]. Thus the global residual only vanishes upon weak satisfaction of both equations.

In the present work, the extended DGM (XDGM), first presented by [19], is considered. Based on the further developments in [20], the method has been implemented in the existing multi-physics DG platform Argo* [21] to treat general conservation laws with two phases. First illustrating results on simple conservation equations of this novel sharp interface method have been presented by Schrooyen et al. in [22, 23]. The main goals of this work are: (i) to present the theory underlying the method as it has been implemented by Schrooyen et al. in Argo (Sec. IV.A), (ii) to develop a coupling strategy with the multiphase Riemann solvers of [24, 25] (Sec. IV.C), and (iii) to demonstrate its potential on simple inviscid test cases with and without phase transitions (Sec. V). The physical modeling including the governing equations closed by general EOS and the jump conditions is first presented (Sec. III). A detailed description of the different numerical aspects mentioned previously is then given (Sec. IV), before commenting the results (Sec. V) and closing by some perspectives (Sec. VI).

III. Physical modeling

A. Governing equations in the bulk of each phase

In the present work, inviscid single species equations with constant heat capacities and no source terms are considered. The Euler equations that are solved by the solver in the bulk of the k^{th} phase can therefore be written as

$$\frac{\partial \mathbf{u}_k}{\partial t} + \nabla \cdot \mathbf{F}^c = \mathbf{0}\,, \tag{1}$$

where the conservative variables and the convective fluxes are given respectively by

$$\mathbf{u}_k = \begin{bmatrix} \rho_k \\ \rho_k \mathbf{v}_k \\ \rho_k E_k \end{bmatrix}, \quad \mathbf{F}^c(\mathbf{u}_k) = \begin{bmatrix} \rho_k \mathbf{v}_k \\ \rho_k \mathbf{v}_k \mathbf{v}_k + p_k \mathbf{I} \\ \rho_k \mathbf{v}_k H_k \end{bmatrix}, \tag{2}$$

in which the density in the k^{th}-phase is noted ρ_k, the velocity vector $\mathbf{v}_k$ and the static pressure p_k. The total energy E_k and the total enthalpy H_k per unit mass both include thermal and kinetic contributions

$$E_k = e_k + \frac{\mathbf{v}_k \cdot \mathbf{v}_k}{2}\,, \quad H_k = h_k + \frac{\mathbf{v}_k \cdot \mathbf{v}_k}{2} = e_k + \frac{p_k}{\rho_k} + \frac{\mathbf{v}_k \cdot \mathbf{v}_k}{2}\,, \tag{3}$$

*Argo is a proprietary code developed by the Belgian research center Cenaero.

with the specific internal energy e_k and the specific enthalpy h_k.
To close the system of Eqs. (1), it must be supplemented by constitutive relations relating intensive thermodynamic properties at equilibrium. For this purpose, it is needed to provide two so-called incomplete EOS that unequivocally determine the state of the fluid inside each phase k, which are a volumetric, i.e. $p_k = p_k(\rho_k, T_k)$, and a caloric, i.e. $e_k = e_k(\rho_k, T_k)$, EOS [26]. These two relations can be expressed, for example, under a general analytical form, known as the Mie-Grüneisen form:

$$\begin{cases} p_k(\rho_k, T_k) = p_{\text{ref},k}(\rho_k) + \rho_k \, \Gamma_k(\rho_k) \, e_{\text{th},k}(\rho_k, T_k) \, , \\ e_k(\rho_k, T_k) = e_{\text{ref},k}(\rho_k) + e_{\text{th},k}(\rho_k, T_k) \, , \end{cases} \tag{4}$$

where the Grüneisen coefficient is given by

$$\Gamma_k(\rho_k) = \frac{1}{\rho_k} \left(\frac{\partial p_k}{\partial e_k} \right)_{\rho_k = \text{const}} \, , \tag{5}$$

and the thermal part of the internal energy by

$$e_{\text{th},k}(\rho_k, T_k) = e_{0,k} + \int_{T_{0,k}}^{T_k} C_{v,k}(\rho_k, T^{'}) \, \mathrm{d}T^{'} \, , \tag{6}$$

with a reference state $\bullet_{0,k}$ and the specific heat at constant volume $C_{v,k}$. $p_{\text{ref},k}$ and $e_{\text{ref},k}$ are the properly chosen states of the pressure and internal energy along some reference curve (e.g. along an isentrope or a single shock Hugoniot) in order to match the experimental data of the material being examined [27]. This type of EOS is an adequate approximation to a wide variety of materials, as long as the state of the fluid remains near the reference curve [28]. That is why this general EOS is chosen as a basis to derive more specific EOS. The extension of the DG solver to general EOS requires also the definition of some thermodynamic derivatives that are used to express the speed of sound, as well as to build the Jacobian matrices for the implicit time-stepping scheme. More details can be found in [29].

B. Jump conditions at the interface

The use of a sharp interface method in this work leads to a so-called local instant 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. Considering the projections of the momentum jump conditions along the normal $\mathbf{n}_\Gamma$ and the tangent $\mathbf{t}_{\Gamma,d}$ ($d = 1, ..., D-1$ with the number of dimensions D) to the interface, they read, for a single component system:

$$\begin{aligned} &- v_{\Gamma,n} [\![\rho]\!] &&+ [\![\rho v_n]\!] &&= 0 \, , \\ &- v_{\Gamma,n} [\![\rho v_n]\!] &&+ [\![\rho v_n^2 + p]\!] &&= \Delta p_\sigma &&+ [\![\cancel{\boldsymbol{\tau} \cdot \mathbf{n}_\Gamma \cdot \mathbf{n}_\Gamma}]\!] \, , \\ &- v_{\Gamma,n} [\![\rho \mathbf{v}_t]\!] &&+ [\![\rho v_n \mathbf{v}_t]\!] &&= \cancel{\nabla_s \sigma \cdot \mathbf{t}_{\Gamma,d}} &&+ [\![\cancel{\boldsymbol{\tau} \cdot \mathbf{n}_\Gamma \cdot \mathbf{t}_{\Gamma,d}}]\!] \, , \\ &- v_{\Gamma,n} [\![\rho E]\!] &&+ [\![(\rho E + p) \, v_n]\!] &&= \Delta p_\sigma v_{\Gamma,n} &&+ [\![\cancel{\boldsymbol{\tau} \cdot \mathbf{v} \cdot \mathbf{n}_\Gamma}]\!] - [\![\mathbf{q} \cdot \mathbf{n}_\Gamma]\!] \, , \end{aligned} \tag{7}$$

where the jump operator across the interface with position $\mathbf{x}$ for some quantity α defined in the bulk of each phase is given by

$$[\![\alpha]\!] = (\alpha_2 - \alpha_1) \, , \quad \text{with} \quad \alpha_k = \lim_{\epsilon \to 0, \epsilon > 0} \alpha(\mathbf{x} \pm \epsilon \mathbf{n}_\Gamma) \, , \tag{8}$$

$v_{\Gamma,n}$ and v_n are the normal components of the interfacial and the bulk velocities, respectively, $\mathbf{v}_t = (v_{t_1}, v_{t_2})$ are the tangential components of the bulk velocities (which is a scalar in two dimensions), $\Delta p_\sigma = (D-1)\sigma\kappa_\Gamma$ represents the pressure jump induced by surface tension effects, with the surface tension coefficient σ and the interface curvature κ_Γ. The cancelled terms in Eqs. (7) correspond to the viscous contributions with the shear stress tensor $\boldsymbol{\tau}$ (the focus is on inviscid flows here), and to the Marangoni effects involving the derivative of the surface tension coefficient along the interface $\nabla_s \sigma$.
Let remark that although the present model is inviscid, it is not isothermal and it was pointed out in [30] that neglecting the heat flux $\mathbf{q}$ in the interfacial balance equations would lead to thermodynamically inconsistent static solutions. At the same time, $\mathbf{q}$ cannot be modeled with the classical Fourier law as this would make the system not hyperbolic

anymore. The assumption of relating the heat flux across the interface[†] to the latent heat of transition h_{lat} at a reference temperature T_{ref} is therefore formulated:

$$[\![\mathbf{q} \cdot \mathbf{n}_\Gamma]\!] = -\dot{m}_\Gamma h_{lat}(T_{ref}) \,, \tag{9}$$

where $\dot{m}_\Gamma$ is the net mass flux across the interface, defined as

$$\dot{m}_\Gamma \triangleq \rho_k (v_{n,k} - v_{\Gamma,n}) \,. \tag{10}$$

An additional assumption to be made to close the model is the no-slip condition at the interface:

$$[\![\mathbf{v} \cdot \mathbf{t}_{\Gamma,d}]\!] = 0 \,, \quad d = 1, 2 \,, \tag{11}$$

which replaces the tangential momentum jump.
In case of phase transition, $\dot{m}_\Gamma$ is different from zero, which implies that $v_{\Gamma,n}$ is an additional unknown, and that the interfacial jump conditions Eqs. (7) do not constitute sufficient matching conditions which are necessary to define the problem uniquely. Consequently, they must be supplemented by an additional condition. Several possibilities exist and can be roughly categorized in:

- Thermodynamic equilibrium models: these models assume an ideal interface between the phases by neglecting any resistance to mass transfer across it. In most of the cases, the temperatures of the liquid and its vapor at the interface are assumed to be the same and equal to the equilibrium saturation temperature of the system. The mass flux is then usually expressed based on the differences between the energy conducted to and from the interface, as it is done e.g. in [31];
- Thermodynamic non-equilibrium models: if the interfacial resistance to mass transfer due to microscopic effects has to be taken into account, several models can be considered:
  - Entropy production: it is possible to find a relation between the irreversible entropy production at the interface due to phase change and the mass flux, based on some coefficients obtained from kinetic theory [32]. This model is usually chosen for multiphase Riemann solvers in order to ensure an entropy satisfying solution of Eqs. (7);
  - Kinetic relation: starting from kinetic theory and velocity distribution functions at the interface, generalized versions of the Hertz-Knudsen and Schrage laws can be obtained, which express the mass flux as a function of the saturated vapor pressure and the pressure in the vapor phase [33].

In this work, we use a simple thermodynamic non-equilibrium model expressing the maximum possible mass flux at the interface based on the Chapman-Jouguet condition [5]:

$$-\dot{m}_\Gamma^2 = \frac{[\![p]\!]}{[\![1/\rho]\!]} \,. \tag{12}$$

IV. Numerical methods

A. The extended discontinuous Galerkin method

As stated before, the XDGM belongs to the class of sharp interface methods and thus preserves the high-order of convergence of the DGM in the vicinity of the interface by avoiding any degradation of the solution. One of the most interesting features of the XDGM is that the mesh is kept fixed and does not need to conform with the interface. The difficulty is then for the method to be able to handle discontinuous fields represented by polynomial functions inside the elements. In what follows, it is explained how this can be achieved with the XDGM, in the context of two-phase inviscid compressible flows.
Let formulate the discretized weak formulation of the governing Euler Eqs. (1) as follows: find $\mathbf{u}^h \in \mathcal{V}^h$ such that

$$\mathcal{E}(\mathbf{u}^h, \mathbf{v}^h) = \sum_{m=1}^{M} \mathcal{E}_m(\mathbf{u}_m^h, \mathbf{v}_m^h) = 0 \,, \quad \forall \mathbf{v}^h \in \mathcal{V}^h \,,$$

in which the subscript $\bullet_m$ denotes the m^{th} element, and $\mathbf{u}^h$ and $\mathbf{v}^h$ are the approximated discrete solutions and test functions, respectively, both expressed as linear combinations of some shape functions forming a base for the finite

[†]The heat flux inside the bulk of each phase is still set to zero.

dimensional space $\mathcal{V}^h$. Developing the contribution of a single element to the global formulation, the distinction is made between:

- an element Ω_m with edges $\partial\Omega_m$ in the bulk of phase k:

$$\begin{aligned}\mathcal{E}_m = &\int_{\Omega_m} \mathbf{v}_m^h \cdot \frac{\partial \mathbf{u}_m^h}{\partial t}\,\mathrm{d}V \\ &- \int_{\Omega_m} \nabla^h \mathbf{v}_m^h : \mathbf{F}^c(\mathbf{u}_m^h)\,\mathrm{d}V \\ &+ \int_{\partial\Omega_m} \mathbf{v}_m^h \cdot \underbrace{(\mathbf{F}^c(\mathbf{u}_m^h)\cdot\mathbf{n}_{\partial\Omega_m})}_{\mathcal{H}(\mathbf{u}_m^{h,+},\mathbf{u}_m^{h,-})}\,\mathrm{d}S\,,\end{aligned} \tag{13}$$

  where $\mathcal{H}$ is a numerical flux function depending on both the interior and exterior solutions, $\mathbf{u}_m^{h,+}$ and $\mathbf{u}_m^{h,-}$, on the element edges;

- an element $\Omega_m = \Omega_{m,1} \cup \Omega_{m,2}$ with edges $\partial\Omega_m = \partial\Omega_{m,1} \cup \partial\Omega_{m,2}$ cut by the phase interface Γ_m (see Fig. 1b):

$$\begin{aligned}\mathcal{E}_m = \sum_{k=1}^{2} \Bigg[&\int_{\Omega_{m,k}} \mathbf{v}_{m,k}^h \cdot \frac{\partial \mathbf{u}_{m,k}^h}{\partial t}\,\mathrm{d}V \\ &- \int_{\Omega_{m,k}} \nabla^h \mathbf{v}_{m,k}^h : \mathbf{F}^c(\mathbf{u}_{m,k}^h)\,\mathrm{d}V \\ &+ \int_{\partial\Omega_{m,k}} \mathbf{v}_{m,k}^h \cdot \underbrace{(\mathbf{F}^c(\mathbf{u}_{m,k}^h)\cdot\mathbf{n}_{\partial\Omega_m})}_{\mathcal{H}(\mathbf{u}_{m,k}^{h,+},\mathbf{u}_{m,k}^{h,-})}\,\mathrm{d}S \\ &+ \int_{\Gamma_m} \mathbf{v}_{m,k}^h \cdot \underbrace{(\,\mathbf{F}^c(\mathbf{u}_{m,k}^h)\cdot\mathbf{n}_{\Gamma_m})}_{\mathcal{H}_{\Gamma,k}(\mathbf{u}_{m,1}^h,\mathbf{u}_{m,2}^h)}\,\mathrm{d}S \Bigg],\end{aligned} \tag{14}$$

  where $\mathcal{H}_{\Gamma,k}$ is a numerical flux function from the k^{th} phase to the other phase, which depends on the solutions in both phases, $\mathbf{u}_{m,1}^h$ and $\mathbf{u}_{m,2}^h$. Contrary to $\mathcal{H}$ which is single-valued across the element edges, $\mathcal{H}_{\Gamma,k}$ is not.

The following remarks can be made based on Eqs. (13, 14):

- extension of the DG space on cut elements: the number of degrees of freedom on a cut element is doubled in order to have two separate representations of the solution, $\mathbf{u}_{m,1}^h$ and $\mathbf{u}_{m,2}^h$, expressed as linear combinations between Lagrange polynomials $\phi_{m,n}$ of order p and N nodal expansion weights $\mathcal{U}_{n,k}$ (see Fig. 1a):

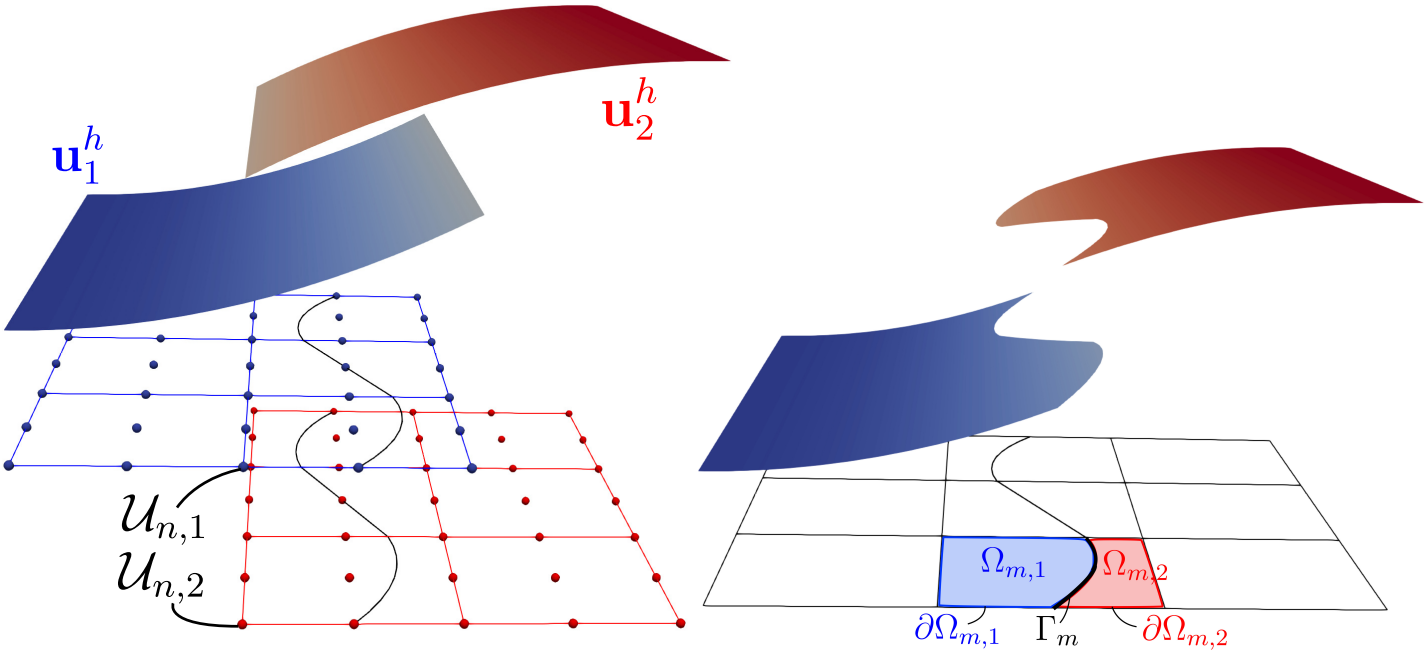
$$\begin{aligned}\mathbf{u}_{m,k}^h(\mathbf{x},t) &= \sum_{n=1}^{N} \phi_{m,n}(\mathbf{x})\mathcal{U}_{n,k}(t)\,, \quad \mathbf{x}\in\Omega_m\,, \\ \mathbf{u}_k^h(\mathbf{x},t) &= \sum_{m=1}^{M} \mathbf{u}_{m,k}^h(\mathbf{x},t)\,.\end{aligned} \tag{15}$$

  Those two representations are then truncated at the interface, resulting in a sharp solution (see Fig. 1b);

- evaluation of integrals on cut elements: the assembling of the weak form in the cut element requires the evaluation of three kind of integrals (see Fig. 2):
  - over the cut volumes $\Omega_{m,k}$ ($k = \{1,2\}$);
  - over the cell edges $\partial\Omega_{m,k}$ ($k = \{1,2\}$);
  - over the phase interface Γ_m.

  The computation of those integrals, which can not be done by classical quadrature rules, is briefly explained in Sec. IV.B;

- computation of interfacial fluxes: the convective numerical flux function across the interface, $\mathcal{H}_\Gamma$, is not based on the classical Riemann solvers used for the flux functions across the element edges, $\mathcal{H}$. The concept of multiphase Riemann solvers on which it relies is presented in Sec. IV.C;
- identification of the interface position: finally, the position of the interface Γ_m, which is needed to perform the different integrals mentioned before, is identified by a signed distance level-set function. A short overview of this technique is given in Sec. IV.D.



(a) Degrees of freedom doubled on cut elements. **(b) Resulting sharp resolution at the interface.**

Fig. 1 Illustration of the XDGM for a discontinuous solution of polynomial order $p = 2$. Underlying mesh with the nodal degrees of freedom (dots) and the iso-0 contour of the level-set (black curve). Note: the degrees of freedom are duplicated on the element edges inside each phase, as in the classical DGM, but it is not represented here for legibility.

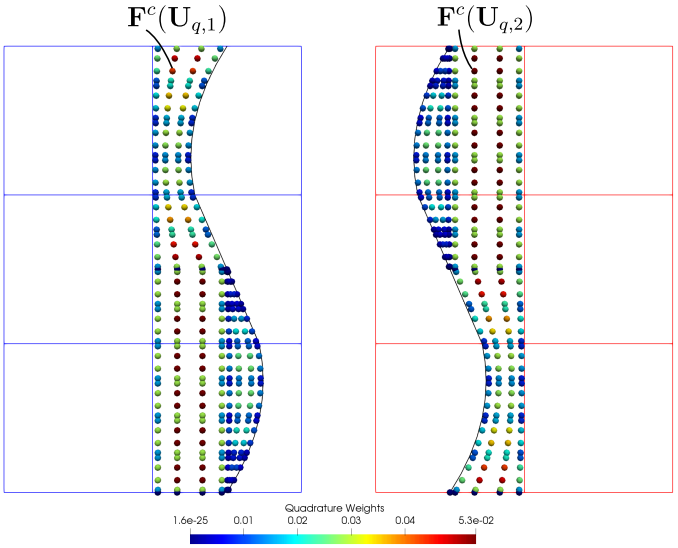
In order to make the link between the different points cited above and the computation of the spatial terms in Eq. (14), the following general algorithm is given:

Algorithm 1: residual assembly in the cut elements For each element cut by the iso-0 contour of the level-set function, loop on both phases ($k = 1, 2$), and do:

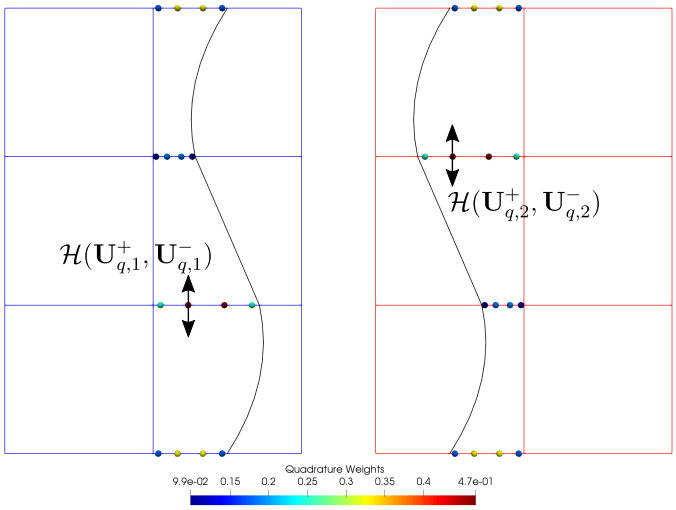
1. *Collocation*: the solution vectors $\mathcal{U}_{n,k}$ (n for the n^{th} degree of freedom) at the interpolation points in the physical element (see Fig. 1a) are used to evaluate the solution vectors $\mathbf{U}_{q,k}$ (q for the q^{th} quadrature point) at the quadrature points in the reference element (see Fig. 2);

2. *Evaluation of the fluxes*: the convective fluxes in the volume, at the element edges and across the interface are then evaluated at the parametric quadrature points: $\mathbf{F}^c(\mathbf{U}_{q,k})$ (Fig. 2a), $\mathcal{H}(\mathbf{U}^+_{q,k}, \mathbf{U}^-_{q,k})$ (Fig. 2b), and $\mathcal{H}_{\Gamma,k}(\mathbf{U}_{q,1}, \mathbf{U}_{q,2})$ (Fig. 2c). The computation of $\mathcal{H}_{\Gamma,k}$ is detailed in Algorithm 2 in Sec. IV.C;

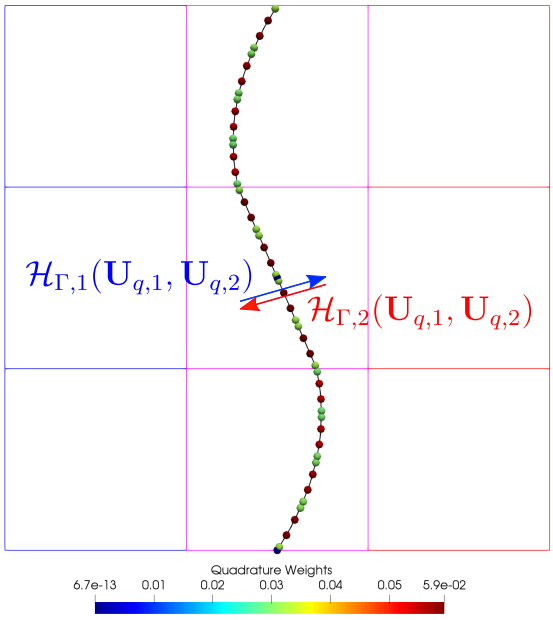
3. *Integration*: those fluxes are finally multiplied by pre-assembled matrices containing the integration weights, the shape functions and their gradients evaluated at the parametric quadrature points and the determinant of the Jacobian of the mapping from physical to reference element.



(a) Volumetric fluxes $\mathbf{F}^c$ integrated over $\Omega_{m,k}$



(b) Element interfacial fluxes $\mathcal{H}$ integrated over $\partial\Omega_{m,k}$.



(c) Double-valued fluxes $\mathcal{H}_{\Gamma,k}$ integrated over the material interface Γ_m.

Fig. 2 Position and weights of the Saye's quadrature [34] of order 4 to integrate the different terms in the weak form Eq. (14).

B. Evaluation of integrals on cut elements

The capacity of the XDGM of reaching high-order accuracy is directly linked to the accuracy with which the integrals in Eq. (14) can be computed. Contrary to normal elements on which standard quadrature rules can be applied, for a cut element the integrals need to be computed over surfaces which are defined implicitly via a fixed isosurface of a smooth scalar function. A special procedure for numerical integration is therefore requested which will build a quadrature for each cut element, i.e. a set of quadrature nodes and weights. Two different approaches have been included in Argo:

- a hierarchical moment-fitting (HMF) method proposed in [35] and implemented into Argo by Schrooyen at al. [23];

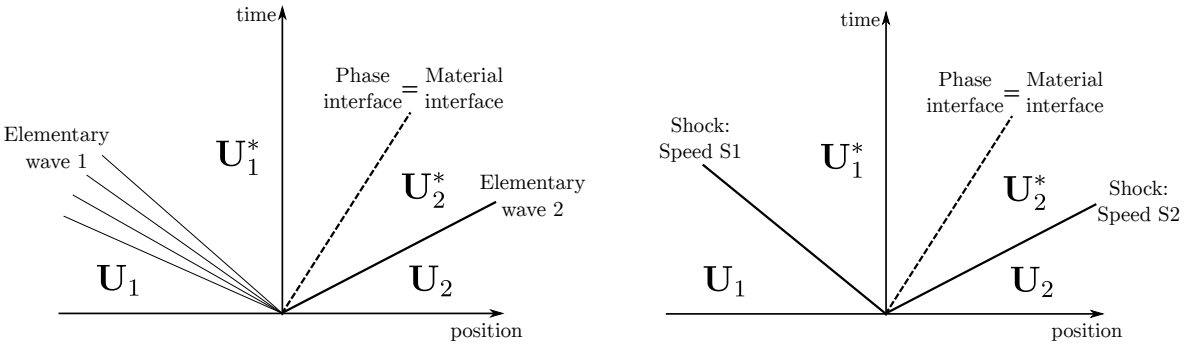
• a high-order quadrature method for implicitly defined domains in hyperrectangles [34], included as a library.

The main difference between both approaches, of interest for the present application, is the following. While the HMF method uses quadrature points on the entire element (for both $\int_{\Omega_{m,k}} \mathrm{d}V$ and $\int_{\Gamma_m} \mathrm{d}S$), the method of Saye defines them strictly inside their respective phase (for $\int_{\Omega_{m,k}} \mathrm{d}V$) or on the zero iso-contour of the level-set (for $\int_{\Gamma_m} \mathrm{d}S$). This is shown in Fig. 2. For two-phase flows, and in particular for compressible flows, evaluating the field variables at positions strictly inside their respective phase is preferable. The second quadrature method is therefore chosen in this work.

In addition to the quadrature rules, a cell-agglomeration strategy, discussed in [20], was included by Schrooyen et al. in the code to circumvent the problem of small cut elements. Indeed, when the ratio between the surface occupied by one phase in an element is much smaller than the surface occupied by the other phase, this can lead to significant numerical issues, such as very ill-conditioned discretisations or extreme time-step restriction. The same agglomeration method is also used to extrapolate the solution from one phase to an element that becomes cut by the interface at a new time step.

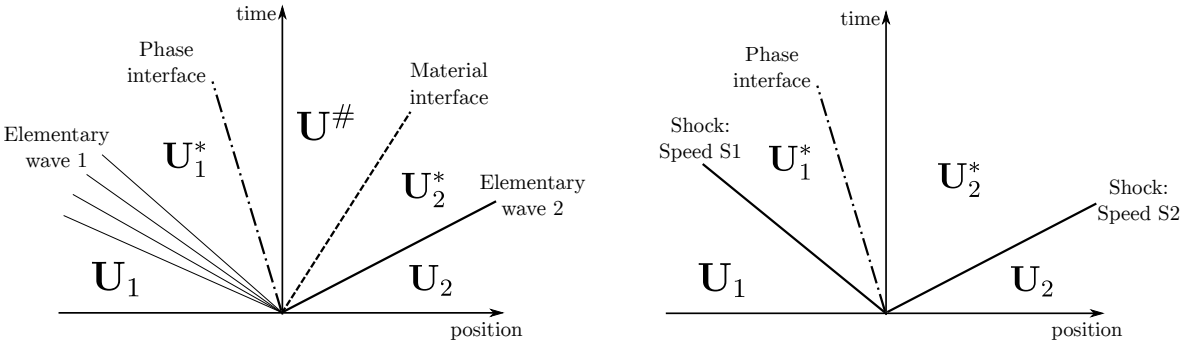
C. Resolution of the jump conditions: multiphase Riemann solver

In order to capture correctly the sharp transition between the two phases, the two distinct solutions on the cut elements must satisfy the jump relations Eqs. (7) at the interface. This is done by computing appropriate fluxes contained in the last terms of Eq. (14). For the inviscid model considered in this work, we propose to rely on the concept of multiphase Riemann solvers. More precisely, we use here two different approximate multiphase Riemann solvers, one without phase transition [24] and the other with phase change effects included [25], both based on the extension of the classical HLLC solver [36]. Those have been shown to produce accurate results, with the advantages of being much faster and more robust than exact solvers (proposed e.g. in [32]). The wave structures from which they are built, with the corresponding exact wave patterns, are illustrated in Fig. 3 and Fig. 4.



(a) Exact **(b) HLLC approximation [24]**

Fig. 3 General Riemann wave structures for an inviscid two-phase flow without phase transition.



(a) Exact **(b) HLLP approximation [25]**

Fig. 4 General Riemann wave structures for an inviscid two-phase flow with phase transition.

The procedure followed to compute the numerical fluxes $\mathcal{H}_{\Gamma,k}$ $(k = 1, 2)$ is the same as the one described in [25] but it

is here restated for convenience:

Algorithm 2: evaluation of interfacial fluxes Given $\mathbf{U}_1$ and $\mathbf{U}_2$, the solution vectors in both phases at one quadrature point on the iso-0 contour of the level-set, and the normal $\mathbf{n}_\Gamma$, the tangents $\mathbf{t}_{\Gamma,d}$ and the curvature κ_Γ approximated from the level-set, do:

Step 1: Project the set of conservative variables onto the normal direction to the interface and assemble the one-dimensional primitive states $\mathbf{V}_1$ and $\mathbf{V}_2$:

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

with $v_{n,k} = \mathbf{v}_k \cdot \mathbf{n}_\Gamma$, $e_k = E_k - 1/2 v_{n,k}^2$, and $p_k = p_k(\rho_k, e_k)$;

Step 2: Solve the Riemann problem to obtain the primitive solution states $\mathbf{V}_1^*$ and $\mathbf{V}_2^*$:

1) Compute the shock wave speeds S_1 and S_2 based on the linearization of the Lax curves [24];
2) Solve the system of equations (Eqs. (16) for HLLC, Eqs. (17) for HLLP) to obtain $\mathbf{V}_1^*$, $\mathbf{V}_2^*$ and the normal velocity of the interface $v_{n,\Gamma}$.

Step 3: Reproject the 1-dimensional solution states to D-dimensions ($D = 2$ or 3) and assemble the set of conservative variables $\mathbf{U}_1^*$ and $\mathbf{U}_2^*$:

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

Step 4: Compute the numerical fluxes normal to the interface:

$$\mathcal{H}_{\Gamma,1} = \begin{cases} \mathbf{F}^c(\mathbf{U}_1)\cdot\mathbf{n}_\Gamma & S_1 > 0 \\ \mathbf{F}^c(\mathbf{U}_1)\cdot\mathbf{n}_\Gamma + S_1(\mathbf{U}_1^* - \mathbf{U}_1) & S_1 \le 0\ \ \&\ \ v_{n,\Gamma} > 0 \\ \mathbf{F}^c(\mathbf{U}_1)\cdot\mathbf{n}_\Gamma + S_1(\mathbf{U}_1^* - \mathbf{U}_1) & S_2 > 0\ \ \&\ \ v_{n,\Gamma} \le 0 \\ \mathbf{F}^c(\mathbf{U}_1)\cdot\mathbf{n}_\Gamma + S_1(\mathbf{U}_1^* - \mathbf{U}_1) & S_2 \le 0 \end{cases} \qquad \mathcal{H}_{\Gamma,2} = \begin{cases} \mathbf{F}^c(\mathbf{U}_2)\cdot\mathbf{n}_\Gamma + S_2(\mathbf{U}_2^* - \mathbf{U}_2) & S_1 > 0 \\ \mathbf{F}^c(\mathbf{U}_2)\cdot\mathbf{n}_\Gamma + S_2(\mathbf{U}_2^* - \mathbf{U}_2) & S_1 \le 0\ \ \&\ \ v_{n,\Gamma} > 0 \\ \mathbf{F}^c(\mathbf{U}_2)\cdot\mathbf{n}_\Gamma + S_2(\mathbf{U}_2^* - \mathbf{U}_2) & S_2 > 0\ \ \&\ \ v_{n,\Gamma} \le 0 \\ \mathbf{F}^c(\mathbf{U}_2)\cdot\mathbf{n}_\Gamma & S_2 \le 0 \end{cases}$$

Step 5: Communicate the interface velocity for the level-set advection (Eq. 18):

$$\mathbf{v}_{LS} = v_{n,\Gamma}\mathbf{n}_\Gamma + \sum_{d=1}^{D-1} v_{t,\Gamma,d}\mathbf{t}_{\Gamma,d}\ , \quad v_{t,\Gamma,d} = \min(|v_{t,k,1}^*|, |v_{t,k,2}^*|)\ .$$

The computation of the interfacial fluxes as described in Algorithm 2 above ensures the satisfaction of the jump conditions at the phase interface. Indeed, the equations that are solved in Step 2.2 involve the jump relations, in addition to the Rankine-Hugoniot relations across the two shock waves. They are written respectively for the HLLC and HLLP solvers as:

- HLLC solver (no phase transition):

$$\begin{aligned} S_1(\rho_1^* - \rho_1) &= \rho_1^* v_{n,1}^* - \rho_1 v_{n,1} \\ S_1(\rho_1^* v_{n,1}^* - \rho_1 v_{n,1}) &= \rho_1^* v_{n,1}^{*,2} - \rho_1 v_{n,1}^2 + p_1^* - p_1 \\ S_1(\rho_1^* E_1^* - \rho_1 E_1) &= \rho_1^* v_{n,1}^* E_1^* - \rho_1 v_{n,1} E_1 + p_1^* v_{n,1}^* - p_1 v_{n,1} \\ S_2(\rho_2^* - \rho_2) &= \rho_2^* v_{n,2}^* - \rho_2 v_{n,2} \\ S_2(\rho_2^* v_{n,2}^* - \rho_2 v_{n,2}) &= \rho_2^* v_{n,2}^{*,2} - \rho_2 v_{n,2}^2 + p_2^* - p_2 \\ S_2(\rho_2^* E_2^* - \rho_2 E_2) &= \rho_2^* v_{n,2}^* E_2^* - \rho_2 v_{n,2} E_2 + p_2^* v_{n,2}^* - p_2 v_{n,2} \\ v_{n,1}^* &= v_{n,2}^* \\ p_1^* &= p_2^* + \Delta p_\sigma \end{aligned} \tag{16}$$

where the interfacial velocity is simply $v_{n,\Gamma} = v^*_{n,1} = v^*_{n,2}$ (hence $\dot{m}_\Gamma = 0$).

- HLLP solver (with phase transition):

$$
\begin{aligned}
S_1(\rho_1^* - \rho_1) &= \rho_1^* v_{n,1}^* - \rho_1 v_{n,1} \\
S_1(\rho_1^* v_{n,1}^* - \rho_1 v_{n,1}) &= \rho_1^* v_{n,1}^{*,2} - \rho_1 v_{n,1}^2 + p_1^* - p_1 \\
\cancel{S_1(\rho_1^* E_1^* - \rho_1 E_1)} &= \cancel{\rho_1^* v_{n,1}^* E_1^* - \rho_1 v_{n,1} E_1 + p_1^* v_{n,1}^* - p_1 v_{n,1}} \\
S_2(\rho_2^* - \rho_2) &= \rho_2^* v_{n,2}^* - \rho_2 v_{n,2} \\
S_2(\rho_2^* v_{n,2}^* - \rho_2 v_{n,2}) &= \rho_2^* v_{n,2}^{*,2} - \rho_2 v_{n,2}^2 + p_2^* - p_2 \\
S_2(\rho_2^* E_2^* - \rho_2 E_2) &= \rho_2^* v_{n,2}^* E_2^* - \rho_2 v_{n,2} E_2 + p_2^* v_{n,2}^* - p_2 v_{n,2} \\
v_{n,\Gamma}(\rho_1^* - \rho_2^*) &= \rho_1^* v_{n,1}^* - \rho_2^* v_{n,2}^* \\
v_{n,\Gamma}(\rho_1^* v_{n,1}^* - \rho_2^* v_{n,2}^*) &= \rho_1^* v_{n,1}^{*,2} - \rho_2^* v_{n,2}^{*,2} + p_1^* - p_2^* - \Delta p_\sigma \\
v_{n,\Gamma}(\rho_1^* E_1^* - \rho_2^* E_2^*) &= \rho_1^* v_{n,1}^* E_1^* - \rho_2^* v_{n,2}^* E_2^* + p_1^* v_{n,1}^* - p_2^* v_{n,2}^* - \dot{m}_\Gamma h_{lat}(T_{ref}) - \Delta p_\sigma v_{n,\Gamma} \\
\dot{m}_\Gamma^{\text{def}} &= \dot{m}_\Gamma^{\text{model}}
\end{aligned}
\tag{17}
$$

where, in the last equation, $\dot{m}_\Gamma^{\text{def}}$ refers to the definition of the mass flux Eq. (10) while $\dot{m}_\Gamma^{\text{model}}$ corresponds to the selected phase transition model, i.e. Eq. (12) in this work. One must point out here that the original system has 10 equations for only 9 unknowns ($\mathbf{V}_1^*$, $\mathbf{V}_2^*$ and $v_{n,\Gamma}$). In order to avoid having to solve an overdetermined system, the choice is made to drop the energy coupling equation in one of the phase [25]. This can be justified if the internal energy at the interface of the phase under consideration is dominated by the phase transition effects and the influence of the surrounding bulk flow is low.

While it is possible to simplify Eqs. (16) in order to find analytical expression for the solution states $\mathbf{V}_k^*$ (see [5]), the system of Eqs. (17) needs to be solved iteratively. Two non-linear solvers were implemented for this purpose: a Broyden method for the simplest test cases and a global Newton method with residual based convergence criterion and adaptive trust region strategy (NLEQ-RES) [37] for the cases where the local Newton method does not converge.
Finally, let note that contrary to single-valued intercell fluxes coupling the solutions at the element edges, two different fluxes are defined in this case, in the idea of the ghost-fluid approach: one from phase 1 to phase 2, the other one from phase 2 to phase 1. Doing so, the mixing between the two fluids obeying possibly dissimilar thermodynamic models is avoided.

D. Interface position and representation: the level-set method

In the XDGM, the interface is represented implicitly as the zero iso-contour of a smooth level-set function φ, for which an advection equation is solved at each time step to update its position. If one writes the level-set equation under a conservative form, a source term appears that is non-zero for compressible flows:

$$
\frac{\partial \varphi}{\partial t} + \nabla \cdot (\varphi \mathbf{v}_{LS}) = \varphi \nabla \cdot (\mathbf{v}_{LS}) \,, \tag{18}
$$

where $\varphi = 0$ indicates the current position of the interface between the two phases, $\varphi \gtrless 0$ signals the presence of one or the other phase and $\mathbf{v}_{LS}$ is the velocity of the interface obtained from the jump conditions (see algorithm 2 in Sec. IV.C). The reinitialization of the level-set combines a recursive contouring algorithm with a fast search tree method [38]. In addition to restore the distance property of the level-set function, the procedure is also useful here to ensure the continuity of the level-set at the element edges. This is indeed necessary for the XDGM in order to be able to define the same quadrature on both sides of the edges cut by the iso-0 (Fig. 2b).
The computation of the normal to the interface and its curvature are needed in the jump conditions and are related to the level-set function via the Bonnet's formula:

$$
\mathbf{n}_\Gamma = \frac{\nabla \varphi}{|\nabla \varphi|} \,, \quad \kappa_\Gamma = \nabla \cdot \mathbf{n}_\Gamma \,. \tag{19}
$$

Following the approach in [39], a lifting procedure is used to compute the gradient of the level-set $\mathbf{G}_\varphi$ inside each element

$$\int_{\Omega_m} \left(\mathbf{G}_\varphi - \nabla\varphi\right) \mathbf{v}_m^h \, \mathrm{d}V = 0 \,, \tag{20}$$

that is subsequently integrated by parts to transform the unknown gradient term $\nabla\varphi$. The curvature is also multiplied by a test function $\mathbf{v}^h$, integrated over the element volume and integrated by parts.
It is important to remark that the governing equations of the flow problem Eqs. (1), discretized by the XDGM , and the level-set Eq. (18), discretized with a classical DG formulation, are not solved simultaneously. The level-set equation is first advanced in time and then the governing equations, before starting a new time step. Within a time step, the interface position can be kept constant or its motion can be taken into account with a cell-agglomeration procedure [40].

E. Artificial viscosity method coupled with the XDGM

As compressible regimes are considered, a special technique is required to remove spurious oscillations that might appear in the vicinity of regions with high gradients of flow quantities. Therefore, the last point about the numerical methods used in this work concerns the extension of a shock-capturing artificial viscosity technique for the elements cut by a material interface. The method is largely based on the one by [41] and the improvement proposed by [42]. It has been considered in this work because of its availability in Argo and its ease of extension to the XDGM. A brief overview is given in what follows.
Contrary to material interfaces that are tracked by a technique like the level-set, one needs to identify the *a priori* unknown position of the shock by an appropriate detector. The detector of Persson and Peraire works on the principle that polynomial expansions of the solutions of two different orders will give similar values in regions where the solution is smooth but will be different when it contains discontinuities. In each element, the polynomial expansion of order p is expressed as

$$\mathbf{u}_m^h(\mathbf{x}) = \sum_{n=1}^{N_m} \phi_{m,n}(\mathbf{x}) \mathcal{U}_{m,n} \,, \quad \mathbf{x} \in \Omega_m \,, \tag{21}$$

where the subscripts $\bullet_m$ and $\bullet_n$ refer to the m^{th} element and the n^{th} degree of freedom on this element, respectively. The shape functions $\phi_{m,n}$ are in this case elementwise Lagrange polynomials of order p. A truncated solution $\hat{\mathbf{u}}_m^h$ is also introduced by projecting the solution on a function space of order $p-1$. Discontinuities can then be detected by comparing the two expansions of different orders, which gives the following cell-local sensor:

$$S_{\Omega_m} = \frac{\int_{\Omega_m} (\mathbf{u}_m^h - \hat{\mathbf{u}}_m^h) \cdot (\mathbf{u}_m^h - \hat{\mathbf{u}}_m^h) \, \mathrm{d}V}{\int_{\Omega_m} (\mathbf{u}_m^h \cdot \mathbf{u}_m^h) \, \mathrm{d}V} \tag{22}$$

Following [43], the sensor is reformulated on cells cut by the interface. In the k^{th} phase on the m^{th} element, the sensor is simply computed by

$$S_{\Omega_{m,k}} = \frac{\int_{\Omega_{m,k}} (\mathbf{u}_{m,k}^h - \hat{\mathbf{u}}_{m,k}^h) \cdot (\mathbf{u}_{m,k}^h - \hat{\mathbf{u}}_{m,k}^h) \, \mathrm{d}V}{\int_{\Omega_{m,k}} (\mathbf{u}_{m,k}^h \cdot \mathbf{u}_{m,k}^h) \, \mathrm{d}V} \,, \tag{23}$$

using the available quadrature rules on cut cells presented in Sec. IV.B.
When a troubled cell has been detected, artificial viscosity has to be added depending on the strength of the shock through a parameter ϵ. It is finally injected in the equations in the form of a Laplacian term in the diffusive fluxes:

$$\mathbf{F}^d(\mathbf{u}, \nabla\mathbf{u}) = \epsilon \nabla\mathbf{u} \,. \tag{24}$$

Although the Euler equations are solved in this work, the diffusive fluxes must be taken into account in the discretization due to the addition of the artificial viscosity. Those are treated with an interior penalty method (IPM) [21].
Let remark finally that for the cut elements, the artificial viscosity is not applied on the phase interface, but only in the volume and on the element edges. It was indeed observed that adding this smoothing term leads to a small smearing of the discontinuity at the interface, which should be avoided.

V. Verification test cases

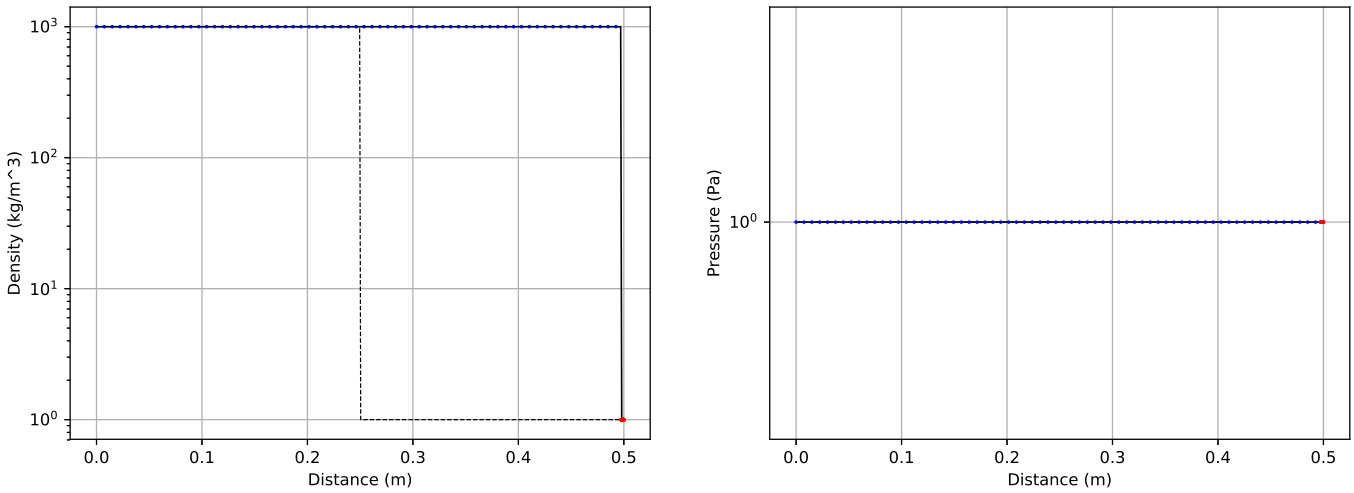
The different test cases proposed in the following are limited to simple theoretical situations. They are considered here more as verification test cases than physical applications, although for some of them physical properties that could represent real gas-liquid systems are used. They were all run on relatively coarse meshes with an interpolation polynomial order equal to two.

A. 1-D translating interface

This test case comes from [44] with the initial values given in Table 1. It concerns a column of dense perfect gas pushing a column of 1000 times less dense perfect gas from the left to the right, at uniformly constant speed and pressure. The result in Fig. 5 shows that the important jump in density is perfectly preserved after the translation of the interface until the end of the domain and that the pressure continuity is maintained. This simple test case illustrates also the capacity of the agglomeration technique to accommodate tiny cut cells without degrading the solution.

Table 1 Initial values for the two-fluid translating interface problem

	ρ	v_n	p	EOS	γ
Left fluid	1000.0	1.0	1.0	Perfect gas	1.4
Right fluid	1.0	1.0	1.0	Perfect gas	1.6



(a) Density **(b) Pressure**

Fig. 5 Comparison between the exact (solid) and the numerical (dotted) density field of the translating interface problem at $t = 0.2475\ s$ for 100 equidistant cells at order $p = 2$, with the initial solution (dashed).

B. Sod Riemann test case with pressure jump

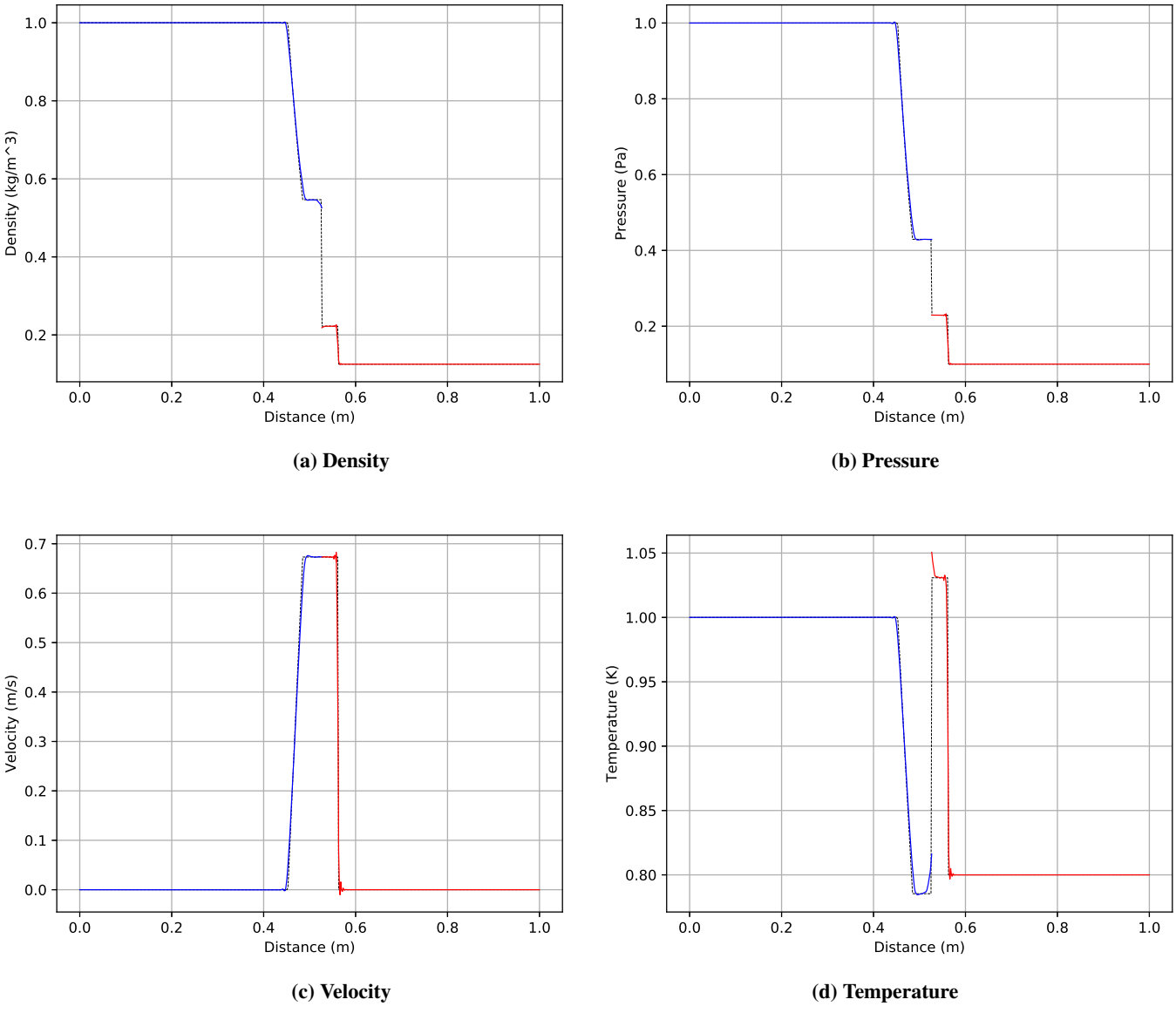
The original Sod test case is modified by imposing a pressure jump $\Delta p_\sigma = 0.2$ across the contact discontinuity, as it is done in [24]. In this situation, the apparition of the rarefaction fan on the left and especially the presence of the shock propagating to the right require the use of artificial viscosity to damp the oscillations near the strong gradients regions. At the beginning of the simulation, the shock starts to develop in the cells cut by the interface and it is therefore important to have a correct way of estimating the artificial viscosity.

The results in Fig. 6 are found to be in very good agreement with the exact solution. The jump in pressure at the contact discontinuity is perfectly imposed, as well as the continuity of the velocity, in accordance with the prescribed jumps. There are slight inclinations in the density profiles near the interface, that could be caused by the addition of the artificial

Table 2 Initial values for the two-phase Sod Riemann problem with pressure jump.

	ρ	v_n	p	EOS	γ
Left fluid	1.0	0.0	1.0	Perfect gas	1.4
Right fluid	0.125	0.0	0.1	Perfect gas	1.4

viscosity. This effect is similarly visible in the temperature plot. But it is fair to say that the contact discontinuity is still successfully kept sharp and that its movement is accurately predicted.



(a) Density **(b) Pressure**

(c) Velocity **(d) Temperature**

Fig. 6 Comparison between the exact (dashed) and the numerical (solid) solutions of the Sod test case with pressure jump at $t = 0.04\ s$ for 200 equidistant cells at order $p = 2$.

C. Gas-Liquid Riemann test case

The purpose of this test case is to demonstrate the performance of the method in a two-phase setting under challenging conditions and is inspired from the one proposed in [45]. It deals with a gas-liquid flow with initial ratio of pressure and density equal to 100. The gas is represented by a perfect gas EOS while the liquid obeys a stiffened gas EOS. The different parameters are listed in Table 3. In addition, a pressure jump between both phases is imposed:

$$\Delta p_\sigma = 10\ bar\ .$$

Table 3 Initial values for the gas-liquid Riemann problem

	ρ (kg/m^3)	v_n (m/s)	p (bar)	EOS	γ /	p_∞ (bar)
Gas	10	10	100	Perfect gas	1.4	/
Liquid	1000	0	1	Stiffened gas	5.5	4900

Fig. 7 shows that the important discontinuity in the density field across the phase interface is perfectly captured by the numerical scheme. The imposed pressure jump is respected while the normal velocity is continuous between both phases. The effect of the artificial viscosity is also illustrated on the velocity field where the important spurious oscillations are removed near the shock and rarefaction waves.

D. Isothermal vaporization test case

This third test case, taken from [46], is about the phase change of a system made of liquid and vapor phases of pure substance. The situation is isothermal and the phase transition is driven by dynamic effects. Classical one-dimensional phase change problems of this kind usually consider the phase transition to be driven by thermal effects instead, by assuming the interface temperature equal to the saturation temperature of the system and the mass flux given by the difference of energy conducted to and from the interface. Even if the release or absorption or latent heat can be taken into account in the present model Eqs. (17), the heat diffusion is not. Therefore it would have been complicated to compare the results with the available analytical solutions in the literature because the way of expressing the mass flux would have been different. With the present test case, the phase transition is triggered by a pressure difference between the liquid and the vapor and the mass flux model Eq. (12) is appropriate. More precisely, we consider the vaporization of the liquid due to a depressurization in the vapor phase imposed by a constant mass flux across the boundary. The other boundaries are solid walls with slip conditions. A sketch of the problem is shown in Fig. 8. As suggested in [46], the initial conditions are taken as the analytical solution of the problem, which implies that the speed of displacement of the interface is non-zero at the initial time and should remain constant during the simulation. These conditions are listed in Table 4 and corresponds to the properties of water.

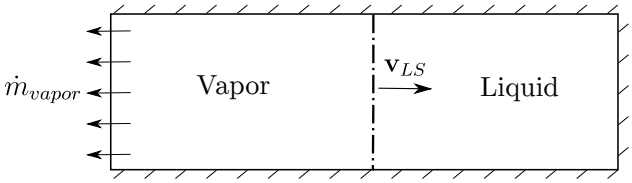


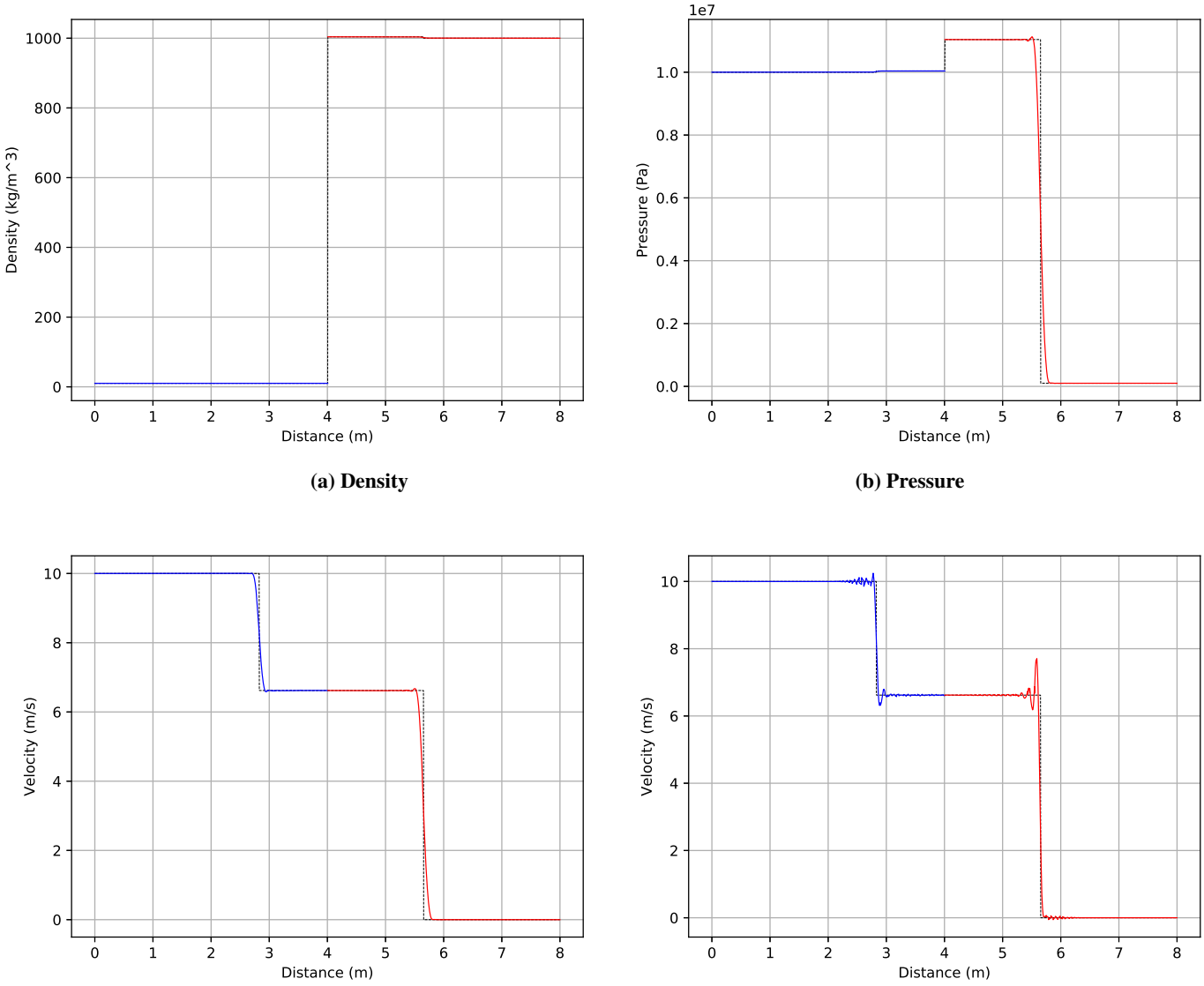
Fig. 8 Setup of the isothermal vaporization test case

	ρ (kg/m^3)	v_n (m/s)	p (bar)	T (K)	EOS	γ /	p_∞ (bar)
Vapor	0.947891	-105.49720	0.96045	373.15	Perfect gas	1.4	/
Liquid	1000.00880	0	1.06609	373.15	Stiffened gas	4.4	6000

$$\dot{m}_{vapor} = 100\ kg/(m^2.s)$$

Table 4 Initial values for the vaporization test case

It can be observed in Fig. 9 that the initial solution corresponding to the analytical one is perfectly conserved with time and that the position of the interface after $0.247\ s$ matches exactly the position predicted by the analytical solution. These results show that the HLLP solver succeeds in preserving the discontinuity at the interface in presence of mass transfer and provides an accurate value of the interface velocity for the evolution of the level-set.



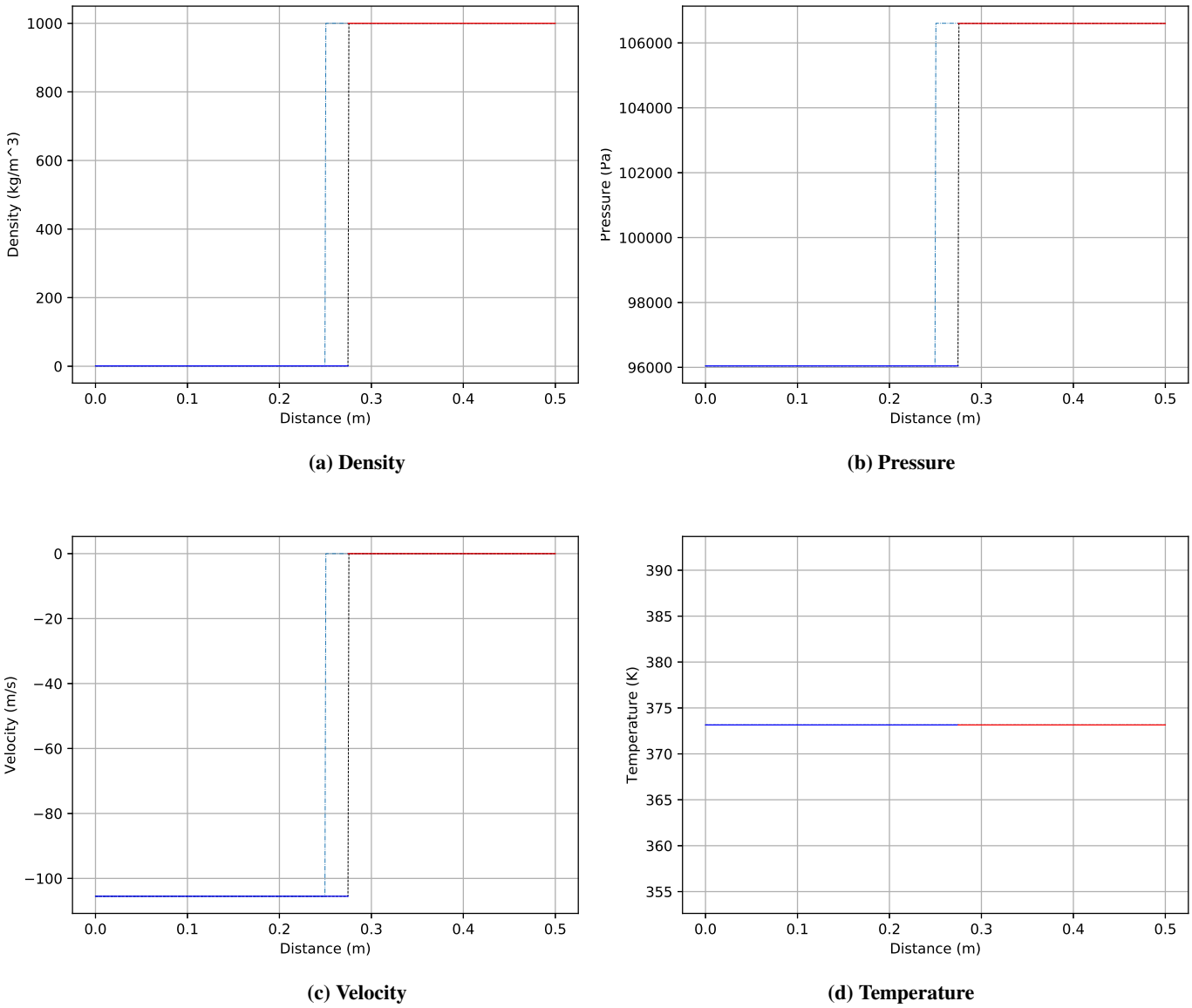
(a) Density **(b) Pressure**

(c) Velocity with artificial viscosity **(d) Velocity without artificial viscosity**

Fig. 7 Comparison between the exact (dashed) and the numerical (solid) solutions of the gas-liquid test case with pressure jump at $t = 0.001\ s$ for 200 equidistant cells at order $p = 2$.

E. Static liquid droplet immersed in a gas with pressure jump

This last two-dimensional test case concerns a liquid droplet immersed in a gas at rest (with no phase transition). The properties for the liquid correspond to water while the one for the gas to air. The droplet is initialized at a slightly higher pressure than the surrounding gas, which is balanced by a prescribed pressure jump at the interface. The setup is illustrated in Fig. 10. The jump in pressure is imposed by the linearized multiphase HLLC Riemann solver. It can be observed in Fig. 11 that after 0.5 s, corresponding to 500 iterations, the initial conditions are perfectly preserved and the droplet remains static as it should. This shows the accuracy with which the various integrals in the weak form of the XDG are performed and how important jumps can be enforced on coarse meshes. This is also a first verification of the ability of the proposed methodology to handle multi-dimensional situations with arbitrary interface geometries.



(a) Density **(b) Pressure**

(c) Velocity **(d) Temperature**

Fig. 9 Comparison between the exact (dashed) and the numerical (solid) solutions of the isothermal vaporization test case at $t = 0.247\ s$ for 200 equidistant cells at order $p = 2$, with the initial solution (dash-dotted).

VI. Conclusions and perspectives

In this paper, a high-order sharp interface method for the simulation of inviscid two-phase compressible flows has been presented. The numerical tool relies on an extended discontinuous Galerkin method (XDGM) taking advantage of the appealing properties of the DGM and allowing to preserve the high-order convergence in the vicinity of a material interface. The discontinuities at the interface are sharply captured by extending the DG space on the cut elements. The coupling between both phases is realized through multiphase Riemann solvers prescribing the physical jump conditions, with the possibility of including phase transition effects. The integrals on the cut elements in the variational discretisation are evaluated with high-order accurate quadrature schemes. The position of the interface is tracked by solving an advection equation for a level-set function, discretized by a classical DG scheme. Finally, an artificial viscosity shock-capturing method has been added to remove spurious oscillations in flow regions with high gradients of flow quantities due to compressible effects.

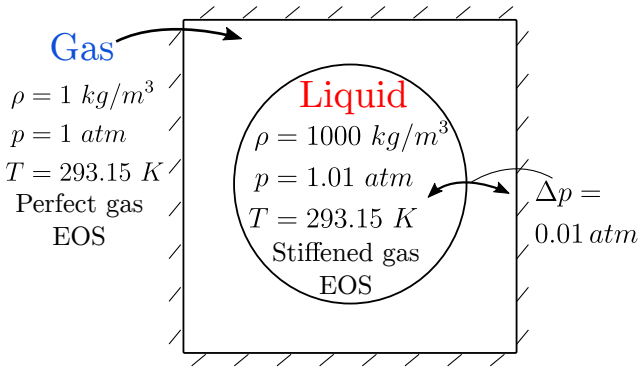
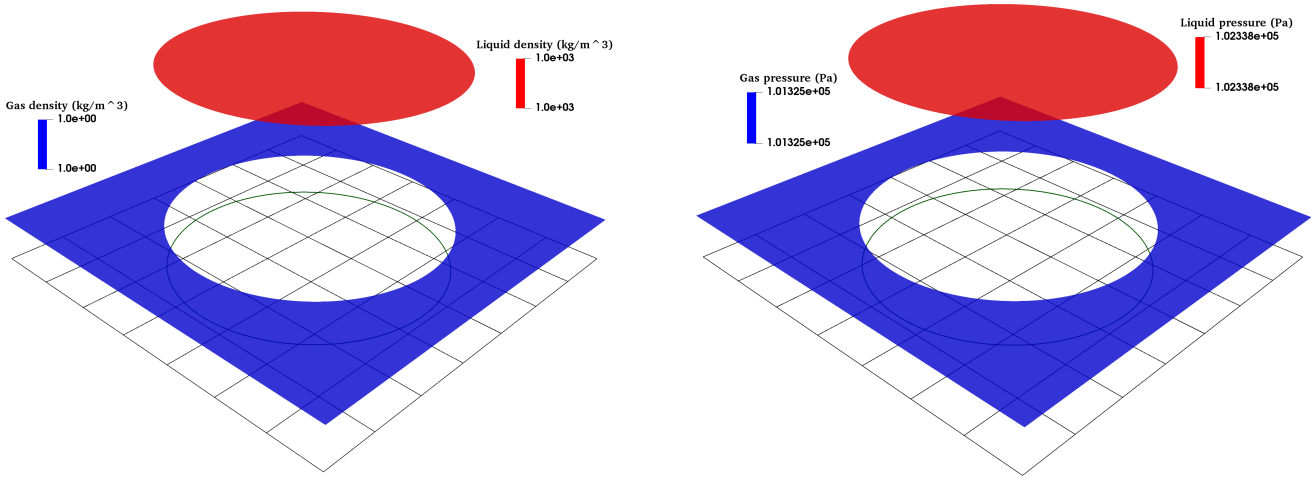


Fig. 10 Setup of the static liquid droplet immersed in a gas



(a) Density **(b) Pressure**

Fig. 11 Elevated solutions for the static droplet test case at $t = 0.5\,s$ on a 7×7 Cartesian grid at order $p = 2$, with the underlying mesh and interface.

Although the results presented in this work are restricted to simple configurations, they have shown the capacity of the current approach to handle multiphase flows with important jumps in physical properties and complex interfacial phenomena such as pressure jumps or phase transition processes. The next steps towards a solver able to simulate realistic gas-liquid flows are described below:

- verify the methods against higher dimensional test cases: the first thing to do will of course be to consider more complex test cases. The code is fully ready to treat two-dimensional problems and first promising results were already obtained, like for the droplet test case. However, the robustness of the method needs to be improved as its convergence seems quite sensitive to some numerical input parameters;
- implement other multiphase Riemann solvers: the two approximate Riemann solvers implemented so far are the fastest but also the ones involving the more important simplifications in terms of the physical phenomena taken into account at the interface. The exact multiphase Riemann solvers of [24] (without phase transition) and [32] (with phase transition), as well as the HLLCP approximate solver with distinct phase and material interfaces of [25] will be compared with the current solvers;
- take into account viscous effects: the viscous terms in the governing equations can have a dominant effect on the interface dynamics, especially between a gas and a liquid. To incorporate them into the compatibility jump conditions, the approach followed by [47] in a FV framework will be explored, i.e. using the solution of a Riemann solver to determine the inviscid fluxes and then to satisfy the conditions on the viscous stresses and heat fluxes.

Acknowledgments

The first author is supported by the Fonds de la Recherche Scientifique (FNRS) of Belgium, under a FRIA grant.

References

[1] Yang, S., Habchi, C., Yi, P., and Lugo, R., "Towards a multicomponent real-fluid fully compressible two-phase flow model," 14th Triennial International Conference on Liquid Atomization and Spray Systems, 2018. Chicago, USA.

[2] Cagnogne, J., Hillewaert, K., and Poletz, N., "A Discontinuous Galerkin Method For Multiphysics Welding Simulations," *Key Engineering Materials*, Vol. 611-612, 2014, pp. 1319–1326.

[3] Tamsaout, T., and Amara, E.-H., "Numerical investigation of hydrodynamics behaviour of melt layer during laser cutting of steel," *Applied Physics A*, Vol. 117, No. 1, 2014, pp. 383–388.

[4] Dias, B., Turchi, A., Stern, E. C., and Magin, T. E., "A model for meteoroid ablation including melting and vaporization," *Icarus*, 2020, p. 113710.

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

[6] Baer, M., and Nunziato, J., "A two-phase mixture theory for the deflagration-to-detonation transition (DDT) in reactive granular materials," *International journal of multiphase flow*, Vol. 12, No. 6, 1986, pp. 861–889.

[7] Abgrall, R., and Karni, S., "Computations of compressible multifluids," *Journal of computational physics*, Vol. 169, No. 2, 2001, pp. 594–623.

[8] Donea, J., Huerta, A., Ponthot, J.-P., and Rodríguez-Ferran, A., "Arbitrary Lagrangian-Eulerian Methods," *Encyclopedia of Computational Mechanics Second Edition*, 2017, pp. 1–23.

[9] Rohde, C., and Zeiler, C., "A relaxation Riemann solver for compressible two-phase flow with phase transition and surface tension," *Applied Numerical Mathematics*, 2014.

[10] Corrigan, A., Kercher, A. D., and Kessler, D. A., "A moving discontinuous Galerkin finite element method for flows with interfaces," *International Journal for Numerical Methods in Fluids*, Vol. 89, No. 9, 2019, pp. 362–406.

[11] Merkle, C., and Rohde, C., "The sharp-interface approach for fluids with phase change: Riemann problems and ghost fluid techniques," *ESAIM: Mathematical Modelling and Numerical Analysis*, Vol. 41, No. 6, 2007, pp. 1089–1123.

[12] Nourgaliev, R. R., Liou, M.-S., and Theofanous, T. G., "Numerical prediction of interfacial instabilities: Sharp interface method (SIM)," *Journal of Computational Physics*, Vol. 227, No. 8, 2008, pp. 3940–3970.

[13] Bai, X., and Deng, X., "A sharp interface method for compressible multi-phase flows based on the cut cell and ghost fluid methods," *Advances in Applied Mathematics and Mechanics*, Vol. 9, No. 5, 2017, pp. 1052–1075.

[14] Sauerland, H., "An XFEM based sharp interface approach for two-phase and free-surface flows," Ph.D. thesis, RWTH Aachen University, 2013.

[15] Saye, R., "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*, Vol. 344, 2017, pp. 647–682.

[16] Kummer, F., "Extended discontinuous Galerkin methods for two-phase flows: the spatial discretization," *International Journal for Computational Methods in Engineering*, Vol. 109, 2017, pp. 259–289.

[17] Gürkan, C., "Extended hybridizable discontinuous Galerkin method," Ph.D. thesis, Universitat Politècnica de Catalunya, 2018.

[18] Fechter, S., and Munz, C.-D., "A discontinuous Galerkin-based sharp-interface method to simulate three-dimensional compressible two-phase flow," *International Journal for Numerical Methods in Fluids*, Vol. 78, No. 7, 2015, pp. 413–435.

[19] Kummer, F., "The BoSSS Discontinuous Galerkin solver for incompressible fluid dynamics and an extension to singular equations." Ph.D. thesis, Technische Universität Darmstadt, 2012.

[20] Müller, B., Krämer-Eis, S., Kummer, F., and Oberlack, M., "A high-order Discontinuous Galerkin method for compressible flows with immersed boundaries," *Internal Journal For Numerical Methods in Engineering*, Vol. 110, 2017, pp. 3–30.

[21] Hillewaert, K., "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.

[22] Schrooyen, P., Cagnogne, J.-S., Poletz, N., and Hillewaert, K., "A High-order Extended Discontinuous Galerkin Method for Moving Immersed Interface Problem," *XDMS Conference*, Umea, 2017.

[23] Schrooyen, P., Arbaoui, L., Cagnone, J.-S., Poletz, N., and Hillewaert, K., "A High-order Extended Discontinuous Galerkin Method to Treat Hydrodynamics Problems," *ECCM-ECFD Conference*, Glasgow (UK), 2018.

[24] Fechter, S., Jaegle, F., and Schleper, V., "Exact and approximate Riemann solvers at phase boundaries," *Computers & Fluids*, Vol. 75, 2013, pp. 112–126.

[25] Fechter, S., Munz, C.-D., Rohde, C., and Zeiler, C., "Approximate Riemann solver for compressible liquid vapor flow with phase transition and surface tension," *Computers & Fluids*, Vol. 169, 2018, pp. 169–185.

[26] Vitale, S., Pini, M., Colonna, P., Gori, G., Guardone, A., Economon, T., Palacios, F., and Alonso, J., "Extension of the SU2 Open Source CFD code to the simulation of turbulent flows of fluids modelled with complex thermophysical laws," *22nd AIAA Computational Fluid Dynamics Conference*, 2015.

[27] Shyue, K.-M., "A Fluid-Mixture Type Algorithm for Compressible Multicomponent Flow with Mie–Gruneisen Equation of State," *Journal of Computational Physics*, Vol. 171, 2001, p. 678–707.

[28] Menikoff, R., and Plohr, B., "The Riemann problem for fluid flow of real materials," *Reviews of Modern Physics*, Vol. 61, 1989.

[29] Henneaux, D., Schrooyen, P., Dias, B., Turchi, A., Chatelain, P., and Magin, T., "Towards a High-Fidelity Multiphase Solver with Application to Space Debris Aerothermal Ablation Modeling," *AIAA Aviation 2019 Forum*, 2019, p. 2876.

[30] Zeiler, C., "Liquid vapor phase transitions: modeling, Riemann solvers and computation," Ph.D. thesis, University of Stuttgart, 2016.

[31] Esmaeeli, A., and Tryggvason, G., "Computations of film boiling. Part I: numerical method," *International journal of heat and mass transfer*, Vol. 47, No. 25, 2004, pp. 5451–5461.

[32] Fechter, S., "A Sharp Interface Method for Compressible Liquid-Vapor Flow with Phase Transition and Surface Tension," *Journal of Computational Physics*, 2017.

[33] Bond, M., and Struchtrup, H., "Mean evaporation and condensation coefficients based on energy dependent condensation probability," *Physical Review E*, Vol. 70, No. 6, 2004, p. 061605.

[34] Saye, R., "High-order quadrature methods for implicitly defined surfaces and volumes in hyperrectangles," *SIAM Journal on Scientific Computing*, Vol. 37, No. 2, 2015, pp. A993–A1019.

[35] Müller, B., Kummer, F., and Oberlack, M., "Highly accurate surface and volume integration on implicit domains by means of moment-fitting," *Internal Journal For Numerical Methods in Engineering*, Vol. 96, 2013, pp. 512–528.

[36] Toro, E. F., Spruce, M., and Speares, W., "Restoration of the contact surface in the HLL-Riemann solver," *Shock waves*, Vol. 4, No. 1, 1994, pp. 25–34.

[37] Deuflhard, P., *Newton methods for nonlinear problems: affine invariance and adaptive algorithms*, Vol. 35, Springer Science & Business Media, 2011.

[38] Marchandise, E., and Remacle, J.-F., "A stabilized finite element method using a discontinuous level set approach for solving two phase incompressible flows," *Journal of computational physics*, Vol. 219, No. 2, 2006, pp. 780–800.

[39] Marchandise, E., "Simulation of three-dimensional two-phase flows : coupling of a stabilized finite element method with a discontinuous level set approach," Ph.D. thesis, Universite catholique de Louvain, 2006.

[40] Kummer, F., Müller, B., and Utz, T., "Time integration for extended discontinuous Galerkin methods with moving domains," *International Journal for Numerical Methods in Engineering*, Vol. 113, No. 5, 2018, pp. 767–788.

[41] Persson, P.-O., and Peraire, J., "Sub-cell shock capturing for discontinuous Galerkin methods," *44th AIAA Aerospace Sciences Meeting and Exhibit*, 2006, p. 112.

[42] Barter, G. E., and Darmofal, D. L., "Shock capturing with PDE-based artificial viscosity for DGFEM: Part I. Formulation," *Journal of Computational Physics*, Vol. 229, No. 5, 2010, pp. 1810–1827.

[43] Geisenhofer, M., Kummer, F., and Müller, B., "A discontinuous Galerkin immersed boundary solver for compressible flows: Adaptive local time stepping for artificial viscosity–based shock-capturing on cut cells," *International Journal for Numerical Methods in Fluids*, Vol. 91, No. 9, 2019, pp. 448–472.

[44] Kreeft, J. J., and Koren, B., "A new formulation of Kapila's five-equation model for compressible two-fluid flow, and its numerical treatment," *Journal of Computational Physics*, Vol. 229, No. 18, 2010, pp. 6220–6242.

[45] Rouy, S., Helluy, P., and Kokh, S., "Test-case number 13: Shock tubes(PA)," *Multiphase Science and Technology*, Vol. 16, No. 1-3, 2004.

[46] Jamet, D., "Test-case number 7-B: Isothermal vaporization due to piston aspiration (PA)," *Multiphase Science and Technology*, Vol. 16, No. 1-3, 2004.

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