Towards a High-Fidelity Multiphase Solver with Application to Space Debris Aerothermal Ablation Modeling

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

With the ever increasing annual launch rates and future in-orbit collisions, the sustainability of space activities is threatened by the exponential growth in the number of space debris orbiting the Earth. To face up to this worrying and unprecedented situation, the current design configurations will have to be adapted in order to maximize the destruction of re-entering decommissioned spacecraft parts. To sustain the implementation of these new technologies, grouped under the Design for Demise terminology, the development of reliable numerical methods able to simulate the complex multi-physical material response of ablating debris is essential. The first elements necessary for the establishment of such a software, namely the adaptation of the discontinuous Galerkin Argo solver to general equations of state and the use of a novel sharp interface method, are addressed in this work. The verification of the implementation of these two elements are conducted through comparisons with exact solutions and the SU2 code. The results obtained show good agreements and pave the way for the development of the next building blocks of this numerical tool.

I. Nomenclature

CFD	=	Computational Fluid Dynamics
D4D	=	Design for Demise
DGM	=	Discontinous Galerkin Method
EOS	=	Equation of State
FEM	=	Finite Elements Method
FVM	=	Finite Volume Method
HMF	=	Hierarchical Moment Fitting
IPM	=	Interior Penalty Method
QP	=	Quadrature Points
TPS	=	Thermal Protection Systems
VdW	=	Van der Waals
XDGM	=	eXtended Discontinuous Galerkin Method

*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.

II. Introduction

More than 60 years of space flight activities have generated an approximated number of 750,000 non-functional man-made objects larger than 1 cm in Earth orbit. These objects, including fragments of decommissioned satellites or rocket bodies, are identified as space debris. In a 'business as usual' scenario, the number of debris will continue to raise due to the intensification of satellite demands and the increasing launch rates, as a consequence the probability for catastrophic collisions between debris and operational satellites or human space flights will also grow. This evolution will negatively affect some orbital regions increasing dramatically the risk for operations. In this context, space debris mitigation guidelines have been endorsed by the United Nations General Assembly in 2007 [1] in order to ensure long-term sustainability of space activities. To support the compliance of future missions with these guidelines, the Design for Demise (D4D) technology [2] has a key role to play.

This new paradigm for design, initiated by the European Space Agency (ESA), was established for the intentional design, assembly, integration and testing of spacecraft components so that a spacecraft will break-up and be completely destroyed before reaching the ground once it re-enters the Earth's atmosphere after the completion of its mission. New safety requirements impose spacecraft manufacturers and mission controllers to ensure a 1-in-10 000 probability threshold for the casualty risk of a single uncontrolled reentry. To comply with this severe regulation, it is essential to understand the degradation mechanism and to be able to model and reproduce the physico-chemical phenomena occuring during the atmospheric reentry of an object.

When a debris reenters the Earth's atmosphere from the low Earth orbit (LEO), which is by far the most congested one, it is subjected to extreme temperatures and heat fluxes due to its hypersonic velocity of about 8 km/s. This intense aerothermodynamic heating comes from the hot plasma air surrounding the body that has come through a strong shock wave and the viscous dissipation of the high-kinetic-energy flow in the boundary layer adjacent to the body surface. At this point, it is important to distinguish between composite materials on the one hand, and metallic and glassy materials on the other hand, which are the two main types of materials found in space debris and classified as critical for demisabity [3]. For composite-made debris, the ablation process is similar to the one encountered for Thermal Protection Systems (TPS), which has been extensively studied in the past decades and remains an active research field. The degradation of metallic and glassy materials, which shares a lot of similarities with the degradation of meteors, is a more recent research field which has gotten increased interest with the implementation of space debris mitigation projects *. The peculiarity of this second class of materials is the interaction of the gas, liquid and solid phases during ablation, hence the name of three phases materials. Exposed to high-enthalpy flow, metallic debris undergoes melting with the subsequent apparition of a flowing liquid molten layer over the solid surface. This liquid layer is progressively removed by intense evaporation and shear forces exerted by the surrounding gas, while the surface of the solid recedes as the melting front progresses. This results in the aerothermal ablation of the debris, sketched in Fig. 1.

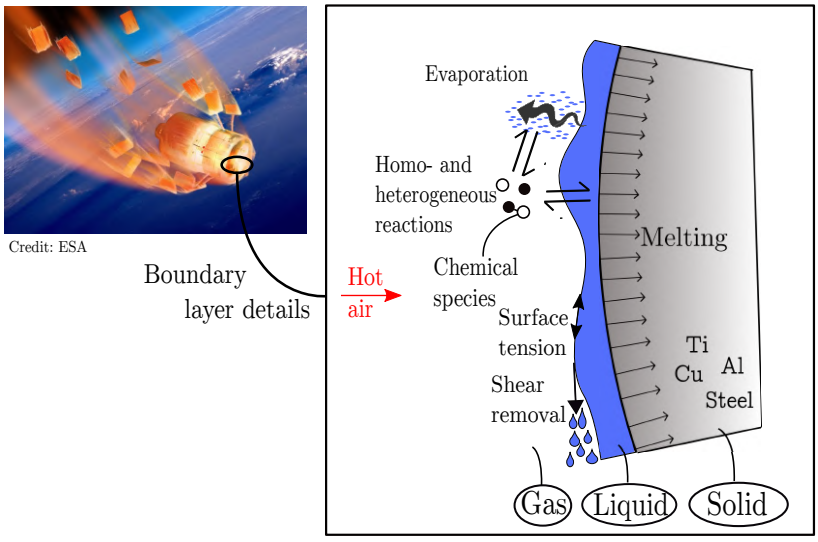


Fig. 1 Physico-chemical phenomena occuring in the vicinity of a metallic debris during its ablation

*The use of glassy materials for TPS was considered before the composite materials and they were therefore studied as early as in the late 50's [4] but it is only with the recent problematic of space debris that they are investigated again.

There exists several numerical tools to predict the survivability of space debris during reentry and assess the prospective risk on ground, including the NASA's ORSAT tool or the ESA's DRAMA tool. Due to the complexity of spacecraft geometries and the variety of flow regimes along the reentry trajectory, these multi-disciplinary tools adopt engineering methods based on simplifying assumptions. The Spacecraft Aerothermal Model tool (SAM) was recently developed to investigate the consequences of simplifying aerothermal assumptions in terms of the effect they have on spacecraft breakup predictions and inferred casualty risk. The conclusions drawn from this tool are clear: the aerothermal heating provided by the simplified models implemented in reentry tools diverge significantly from the results obtained with classical Computational Fluid Dynamics (CFD) codes, which results in unacceptable uncertainties on the estimation of the survavibility of space debris [5]. These models must therefore be improved and complemented with advanced physical models, high-fidelity CFD simulations and experimental campaigns to reach the desired level of reliability and sustain the development of the D4D technology.
As pointed out in [6], there are still technological gaps to establish a highly accurate CFD tool able to simulate the detailed aerothermal ablation process. Indeed, most of the current methods do not consider the presence of the liquid layer between the high-enthalpy gas and the solid surface of the debris, which has important effects on the gas-surface interaction and the heterogeneous chemical reactions. The literature concerning the modeling and the simulation of space debris made of three phase materials is in fact scarce. The scientific community studying the atmospheric entry of meteorites provides most of scientific papers for melting materials [7, 8]. When the presence of the molten layer is taken into account, as in [9], the presence of several species is disregarded and the numerical techniques for the treatment of the moving gas-liquid interface do not allow to reach high-order of accuracy in the interface region, which could influence the prediction of the debris destruction.

This paper is the first publication about the development of a new high-fidelity numerical tool for the simulation of the aerothermal ablation of three phases materials. The implementation of this tool will be conducted in the Argo software developed by Cenaero which relies on the discontinuous Galerkin method (DGM) [10], benefiting from several modules available in this multi-physics platform. From a numerical point of view, the situation depicted in Fig. 1 is particularly challenging as it involves the resolution of multi-species equations separated by moving interfaces with phase transformations. Different numerical ingredients are needed to deal with this problem and two of them are addressed in the present work. The first one is the extension of Argo to flow obeying general non-ideal equation of states (EOS). The governing equations considered are presented in Sec. III while the implementation details are given in Sec. IV.B and the appendix. The second element of the numerical tool covered in this work concerns the verification of a sharp interface method which has just been implemented in the platform [11] and which is used in conjunction with general EOS. The underlying theory of this novel method is described in Sec. IV.C, preceded by an introduction to the DGM in Sec. IV.A. These two numerical ingredients are verified on some simple test cases in Sec. V. Before the conclusion, the perspectives for future work are given by presenting an overview of the main building blocks of the numerical tool.

III. Generalized flow equations

In this work, single species equations with constant viscosity, conductivity and heat capacities, with no turbulence model and no source term are considered. The compressible Navier-Stokes equations that are solved by the solver can therefore be written, by splitting the convective and the diffusive parts, as

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

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

$$\mathbf{u} = \begin{pmatrix} \rho \\ \rho \mathbf{v} \\ \rho E \end{pmatrix}, \quad \mathbf{F}^c(\mathbf{u}) = \begin{pmatrix} \rho \mathbf{v} \\ \rho \mathbf{v}\mathbf{v} + p\mathbf{I} \\ \rho \mathbf{v} H \end{pmatrix}, \quad \mathbf{F}^d(\mathbf{u}, \nabla \mathbf{u}) = \begin{pmatrix} 0 \\ \boldsymbol{\tau} \\ \boldsymbol{\tau} \cdot \mathbf{v} - \mathbf{q} \end{pmatrix}, \tag{2}$$

in which the density is noted ρ, the velocity vector $\mathbf{v}$ and the static pressure p. The total energy E and the total enthalpy H per unit mass both include thermal and kinetic contributions

$$E = e + \frac{\mathbf{v} \cdot \mathbf{v}}{2}, \quad H = h + \frac{\mathbf{v} \cdot \mathbf{v}}{2} = e + \frac{p}{\rho} + \frac{\mathbf{v} \cdot \mathbf{v}}{2}, \tag{3}$$

with the specific internal energy e and the specific enthalpy h. The constitutive equations for the shear stress tensor $\boldsymbol{\tau}$ and the heat flux $\mathbf{q}$ are written as

$$\boldsymbol{\tau} = \mu \left[\nabla \mathbf{v} + (\nabla \mathbf{v})^T - \frac{2}{3} \nabla \cdot \mathbf{v} \mathbf{I} \right] , \quad \mathbf{q} = -\lambda \nabla T , \tag{4}$$

following the assumption of a Newtonian fluid and the Fourier law, with the dynamic viscosity μ, the thermal conductivity λ and the temperature T.
During an atmospheric reentry, the gas flow around a body goes through different regimes, from the rarefied to the continuous. In case of melting, the liquid layer can always be assumed as a continuous medium. In this work, since the Navier-Stokes equations are considered for all the phases, the continuous regime at low altitude is targeted. It is indeed during this regime that the most critical components of a spacecraft will start to significantly ablate and possibly be desintegrated. For meteoroids that travel at much higher speeds, the rarefied regime is important to study too [12].

A. General analytical equation of state

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 incomplete EOS that unequivocally determine the state of the fluid, which are a volumetric, i.e. $p = p(\rho, T)$, and a caloric, i.e. $e = e(\rho, T)$, EOS [13]. These two relations can be expressed, for example, under a general analytical form, known as the Mie-Grüneisen form:

$$\begin{cases} p(\rho, T) = p_{\text{ref}}(\rho) + \rho \, \Gamma(\rho) \, e_{\text{th}}(\rho, T) , \\ e(\rho, T) = e_{\text{ref}}(\rho) + e_{\text{th}}(\rho, T) , \end{cases} \tag{5}$$

where the Grüneisen coefficient is given by

$$\Gamma(\rho) = \frac{1}{\rho} \left(\frac{\partial p}{\partial e} \right)_{\rho=\text{const}} , \tag{6}$$

and the thermal part of the internal energy by

$$e_{\text{th}}(\rho, T) = e_0 + \int_{T_0}^{T} C_v(\rho, T') \, \mathrm{d}T' , \tag{7}$$

with a reference state $\bullet_0$ and the specific heat at constant volume C_v. p_{ref} and e_{ref} 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 [14]. 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 [15]. In particular, it has been used to model the behavior of compressible liquid metals [9, 16], which makes it a good candidate to model the condensed phase (i.e. the liquid and the solid) of a metallic ablating space debris. Moreover, it can be generalized to multicomponent equations [14] that will be considered in future work (see Sec. VI). That is why this general EOS is chosen as a basis to derive more specific EOS. In this work, the solver is verified with the perfect gas and the van der Waals EOS, whose definitions are recalled in Sec. VII.C. It would be rather straightforward to use it with other EOS based on the general Eqs. (5) by providing an expression for p_{ref}, e_{ref} and e_0, as well as some thermodynamic derivatives listed hereafter in Sec. III.B.

B. Thermodynamic derivatives required by the numerical scheme

As it is explained in Sec. IV.B, the extension of the solver to general EOS requires, in addition to the different terms in Eqs. (5), the expression of the following thermodynamic derivatives:

$$\begin{aligned}
\left(\frac{\partial p}{\partial \rho}\right)_{e=\text{const}}(\rho, e) &= (e - e_{\text{ref}})\left(\Gamma + \frac{\mathrm{d}\Gamma}{\mathrm{d}\rho}\rho\right) + \frac{\mathrm{d}p_{\text{ref}}}{\mathrm{d}\rho} - \Gamma\rho\frac{\mathrm{d}e_{\text{ref}}}{\mathrm{d}\rho}\,, \\
\left(\frac{\partial p}{\partial e}\right)_{\rho=\text{const}}(\rho, e) &= \Gamma\rho\,, \\
\left(\frac{\partial e}{\partial \rho}\right)_{T=\text{const}}(\rho, T) &= \frac{\mathrm{d}e_{\text{ref}}}{\mathrm{d}\rho} + \frac{\mathrm{d}e_T}{\mathrm{d}\rho}\,, \\
\left(\frac{\partial e}{\partial T}\right)_{\rho=\text{const}}(\rho, T) &= \frac{\mathrm{d}e_T}{\mathrm{d}T}\,.
\end{aligned} \tag{8}$$

The speed of sound, which is a quantity that needs to be computed during the resolution of the equations, is obtained from the partial derivatives of the pressure as

$$c^2 = \left(\frac{\partial p}{\partial \rho}\right)_e + \frac{p}{\rho^2}\left(\frac{\partial p}{\partial e}\right)_\rho\,. \tag{9}$$

The identification of these four partial derivatives to define any thermodynamic model allows to exploit the high level of abstraction of the Argo platform, written in C++.

IV. Numerical methods

The increase of fidelity of the models elaborated to reproduce the numerous phenomena contributing to the ablation of a reentering space debris are inevitably accompanied by an increase of the complexity of the numerical methods designed to solve them. In this section, the high-order discretization method adopted by Argo to approximate the solution of the system of non-linear partial differential equations (1) is first introduced. Then, the extension of the existing solver to more general and complex thermodynamic models is discussed. Finally, a state-of-the-art numerical technique to handle discontinous solutions is described.

A. The discontinuous Galerkin method (DGM)

As the result of their flexibility, robustness and efficiency, low-order schemes (i.e. limited to second-order accuracy in space) form the basis of all commercially available fluid flow solvers. The extension of these schemes, which mostly rely on the Finite Volume Method (FVM), to higher-order of convergence compromises their favorable properties. Recently, different unstructured high-order methods to which the DGM belongs, have emerged. These methods offer the possibility of combining the geometrical flexibility of FVM or Finite Element Method (FEM) with the high precision of finite difference or spectral methods. For the same number of degrees of freedom in the numerical approximation, high-order methods present superior accuracy compared to low-order ones. In many industrial flow problems, for which ever more stringent accuracy requirements are imposed, the computational cost associated to high-order methods become much less restrictive than with low-order methods, that would need too extensive computational resources [17]. The study of the ablation of a space debris during reentry and the associated deployment of the D4D technology can be seen as one situation for which the use of high-order methods should be favoured. The high stiffness of the thermophysical models considered for the metallic molten layer combined with the presence of several reacting species leads to localized and rapidly varying flow structures in the vicinity of the ablating surface. Added to this are the jumps in properties at the gas-liquid interface, which will be treated by a dedicated method described in Sec. IV.C. To capture these intricate flow features, a numerical tool providing high-resolution is necessary [10].

The DGM can be seen as a mix between the FEM and FVM methods, from which it takes the advantages. In this work, it is applied to discretize spatially Eq. (1) whose exact solution $\mathbf{u}$ belongs to an infinite dimensional space $\mathcal{V}$. Like in any FEM, the approximated solution $\mathbf{u}^h$ is searched into a finite dimensional subspace $\mathcal{V}^h \subset \mathcal{V}$. To construct $\mathcal{V}^h$, the computational domain is discretized into a finite number of elements: $\Omega \approx \cup\, \Omega_m$ $(m = 1, ..., M)$. The global solution is then represented as the sum of the local solutions on each element. The solutions on each element are expressed themselves as a linear combination of the linearly independent local shape functions $\phi_{m,n}$, forming a base for $\mathcal{V}^h$, and

the unknown nodal degrees of freedom $\mathbf{U}_{m,n}(t) \triangleq \mathbf{u}^h_m(\mathbf{x}_{m,n}, t)$ [18]:

$$\mathbf{u}(\mathbf{x}, t) \approx \mathbf{u}^h(\mathbf{x}, t) = \sum_{m=1}^{M} \mathbf{u}^h_m(\mathbf{x}, t), \tag{10a}$$

$$\text{with:} \quad \mathbf{u}^h_m(\mathbf{x}, t) = \sum_{n=1}^{N_m} \phi_{m,n}(\mathbf{x})\mathbf{U}_{m,n}(t), \quad \mathbf{x} \in \Omega_m \,, \tag{10b}$$

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, with no restriction on continuity between elements. Therefore, $\mathcal{V}^h = \oplus_{m=1}^{M}\{\phi_{m,n}\}_{n=1}^{N_m}$ is a broken functional space and the numerical solution is discontinuous between elements, i.e. the interface nodes between two elements are duplicated. The unknowns degrees of freedom $\mathbf{U}_{m,n}$ are found by imposing the orthogonality between the residual of Eq. (1), computed using the approximate solution $\mathbf{u}^h$, and any function in the test space $\mathcal{V}^h$:

$$\int_\Omega \mathbf{v}^h \mathcal{L}(\mathbf{u}^h)\,\mathrm{d}\Omega = 0\,, \forall \mathbf{v}^h \in \mathcal{V}^h \quad \text{where} \qquad \mathcal{L}(\mathbf{u}^h) = \frac{\partial \mathbf{u}^h}{\partial t} + \nabla \cdot \mathbf{F}^c(\mathbf{u}^h) - \nabla \cdot \mathbf{F}^d((\mathbf{u}^h), \nabla \mathbf{u}^h)\,. \tag{11}$$

In the Galerkin formulation, these test functions are written as a linear combination between their degrees of freedom $\mathbf{V}$ and the shape functions

$$\mathbf{v}^h(\mathbf{x}) = \sum_{m=1}^{M}\sum_{n=1}^{N_m} \phi_{m,n}(\mathbf{x})\mathbf{V}_{m,n}\,. \tag{12}$$

Using elementwise decomposition and integration by parts, Eq. (11) can be expanded as

$$\begin{aligned}
&\sum_{\Omega_m \in \Omega} \int_{\Omega_m} \mathbf{v}^h_m \cdot \frac{\partial \mathbf{u}^h_m}{\partial t}\,\mathrm{d}V \\
&- \sum_{\Omega_m \in \Omega} \left[\int_{\Omega_m} \nabla \mathbf{v}^h_m : \mathbf{F}^c(\mathbf{u}^h_m)\,\mathrm{d}V - \oint_{\partial\Omega_m} (\mathbf{v}^h_m \otimes \mathbf{n}) : \mathbf{F}^c(\mathbf{u}^h_m)\,\mathrm{d}S \right] \\
&+ \sum_{\Omega_m \in \Omega} \left[\int_{\Omega_m} \nabla \mathbf{v}^h_m : \mathbf{F}^d(\mathbf{u}^h_m, \nabla \mathbf{u}^h_m)\,\mathrm{d}V - \oint_{\partial\Omega_m} (\mathbf{v}^h_m \otimes \mathbf{n}) : \mathbf{F}^d(\mathbf{u}^h_m, \nabla \mathbf{u}^h_m)\,\mathrm{d}S \right] = 0\,,\ \forall \mathbf{v}^h_m \in \mathcal{V}^h\,,
\end{aligned} \tag{13}$$

with the normal to the face element $\mathbf{n}$. Due to the lack of continuity of the solution between the elements, the solution uniqueness is ensured by defining suitable interface numerical flux functions $\boldsymbol{\gamma}$ in order to provide the appropriate boundary conditions linking each element to its neighbors. Converting the sum on the element faces $\partial\Omega_m$ by the sum on all of the element interfaces I_i, the variational formulation becomes [10]

$$\begin{aligned}
&\sum_{\Omega_m \in \Omega} \int_{\Omega_m} \mathbf{v}^h_m \cdot \frac{\partial \mathbf{u}^h_m}{\partial t}\,\mathrm{d}V \\
&- \sum_{\Omega_m \in \Omega} \int_{\Omega_m} \nabla \mathbf{v}^h_m : \mathbf{F}^c(\mathbf{u}^h_m)\,\mathrm{d}V + \sum_{I_i \in I} \int_{I_i} \boldsymbol{\gamma}^c\,\mathrm{d}S \\
&+ \sum_{\Omega_m \in \Omega} \int_{\Omega_m} \nabla \mathbf{v}^h_m : \mathbf{F}^d(\mathbf{u}^h_m, \nabla \mathbf{u}^h_m)\,\mathrm{d}V - \sum_{I_i \in I} \int_{I_i} \boldsymbol{\gamma}^d\,\mathrm{d}S = 0\,,\ \forall \mathbf{v}^h_m \in \mathcal{V}^h\,.
\end{aligned} \tag{14}$$

The interface flux functions for the convective part $\boldsymbol{\gamma}^c$ and the diffusive part $\boldsymbol{\gamma}^d$ are significantly different and are described separately in the two following sections. They will be chosen such that the resulting method is consistent and conservative.

1. Convective interface flux function

Since the continuity of the solution is not ensured between elements, the DG discretization involves the solution of a Riemann problem at the boundary of each elements, as in the FVM . The characteristic directions associated to

the Jacobian matrix of the convective flux are used to build the flux functions, by the resolution of the exact Riemann problem or its linearized approximation. The interface flux term is then expressed as

$$\boldsymbol{\gamma}^c(\mathbf{u}^{h,+}, \mathbf{u}^{h,-}, \mathbf{v}^{h,+}, \mathbf{v}^{h,-}, \mathbf{n}) = \Big([\mathbf{v}^h] \cdot \mathbf{n}\Big)\, \mathcal{H}(\mathbf{u}^{h,+}, \mathbf{u}^{h,-}, \mathbf{n})\,, \tag{15}$$

where the jump operator is defined as $[\mathbf{a}] = \mathbf{a}^-\mathbf{n}^- + \mathbf{a}^+\mathbf{n}^+$. The + and − superscripts designate discontinuous quantities on either side of an interface, depending on the element from which the interface is approached. The numerical flux function $\mathcal{H}$ can be chosen as any of the available Riemann solvers in the literature. In this work, the SLAU scheme and the HLLC scheme were considered and extended to general EOS.

2. Diffusive interface flux function

In contrast to the discretization of the convective terms, many alternatives exist for the discretization of the diffusive terms [19]. In Argo, an interior penalty method (IPM) has been chosen to ensure a stable diffusive term. To understand this method, it is convenient to consider the variational formulation Eq. (14) written on a single element Ω_m. The interface flux functions can then be interpreted as imposing weak internal Dirichlet-type boundary conditions on the element boundaries [10]. The approach of Nitsche for imposing weak Dirichlet boundary conditions for elliptic problems (the diffusive part of the Navier-Stokes equations has indeed an elliptic character) is followed [19]. It consists in adding a term to the weak form penalizing the difference between the exact Dirichlet boundary condition and the actual value on the boundary, instead of incorporating the boundary conditions directly into the finite element space, which would lead to stability issues. In addition to this penalization term, another term is present into the weak form to ensure the consistency of the method. Nitsche's approach can be extended to impose the connectivity between elements, leading to the so-called IPM. By writing the diffusive flux as $\mathbf{F}^d = \mathbf{D}(\mathbf{u}^h)\nabla\mathbf{u}^h$, with $\mathbf{D}$ the Jacobian of the diffusive fluxes with respect to the solution gradients, the IPM reads:

$$\boldsymbol{\gamma}^d(\mathbf{u}^{h,+}, \mathbf{u}^{h,-}, \mathbf{v}^{h,+}, \mathbf{v}^{h,-}, \mathbf{n}) = [\mathbf{v}^h] : \langle \mathbf{D}\nabla\mathbf{u}^h \rangle + \theta[\mathbf{u}^h] : \langle \mathbf{D}\nabla\mathbf{v}^h \rangle + \alpha[\mathbf{v}^h] : [\mathbf{u}^h], \tag{16}$$

where the average operator is defined as $\langle a \rangle = 0.5 \cdot (a^- + a^+)$. The first term in this expression arises directly from the partial integration of the diffusive term, the second term determines if the form is symmetric ($\theta = 1$), non-symmetric ($\theta = -1$) or incomplete ($\theta = 0$), and the last term is the penalization term with the parameter α governing the stability of the method.

B. Adaptation of implicit DGM to general EOS

The purpose of building a numerical scheme handling general EOS is to be able to switch easily from a perfect gas EOS for the gas phase to a more complex EOS for the condensed phase. While it is common to describe the gas phase with the compressible Navier-Stokes equations closed by the perfect gas law, the liquid is usually assumed to be governed by the incompressible equations in order to avoid having to resolve the propagation of the fast pressure waves. With the incompressibility assumption, the density is not related to the pressure through an EOS. In the case of reentry conditions however, the temperature gradients inside the liquid layer can be really high, which necessitates to take into account compressibility effects inside this phase to correctly describe its thermal dilatation. The diffusion of chemical species inside the molten layer could be important to take into account as well since it influences the thermophysical properties and the heterogeneous reactions at the gas-liquid interface. To study these effects, the resolution of compressible multi-species equations is targeted, not only in the gas phase, but also inside the liquid phase, which reinforces even more the need for general EOS suitable for describing the behavior of the liquid phase. That is why the results presented in this work mainly concern the adaptation of the existing compressible solver for perfect gas of Argo to more general EOS under the form of Eqs. (5).

1. Convective and diffusive fluxes

The hydrodynamics is coupled with the thermodynamics through the use of the EOS that determines the pressure in the convective flux and the temperature (under the form of a gradient) in the diffusive flux, as function of two independent variables, namely the density and the temperature itself in the case of Eqs. (5). The density is directly available as it is part of the set of conservative variables $\mathbf{u} = (\rho, \rho v_x, \rho v_y, \rho E)^T$ for which the equations are solved (in two dimensions here). Obtaining the temperature on the other hand requires the resolution of the non-linear equation

$$T \;:\; e^*(\mathbf{u}) - e(\rho, T) = 0\,, \quad \text{with} \qquad e^*(\mathbf{u}) = \frac{\rho E}{\rho} - \frac{1}{2}\frac{(\rho v_x)^2 + (\rho v_y)^2}{\rho^2} = \frac{u_4}{u_1} - \frac{1}{2}\frac{u_2^2 + u_3^2}{u_1^2}\,, \tag{17}$$

where the notation e^* is used to distinguish the internal energy obtained from the conservative variables from the one given by the thermodynamic model e. When filling the convective flux vector, the pressure is thus readily obtained via $p = p(\rho, T)$ that depends on the thermodynamic model considered.
For the diffusive flux, the computation of the spatial temperature gradient is not straightforward. Indeed, contrary to the gradients of the conservative variables which are directly available in the DGM, obtaining the gradient of the temperature is a bit more involved as the EOS comes into play. For a general EOS where the internal energy depends both on density and temperature, the spatial gradient of the temperature can be obtained as [20]

$$\nabla T = \frac{1}{\left(\frac{\partial e}{\partial T}\right)_\rho}\left[\nabla e^* - \left(\frac{\partial e}{\partial \rho}\right)_T \nabla \rho\right], \quad \text{where} \qquad \nabla e^* = \frac{1}{\rho}\left[\nabla u_4 - \left(\frac{u_4}{u_1} - \frac{u_2^2+u_3^2}{u_1^2}\right)\nabla \rho - \nabla u_2 \frac{u_2}{u_1} - \nabla u_3 \frac{u_3}{u_1}\right], \quad (18)$$

in which the two partial derivatives of the internal energy listed in Eqs. (8) appear.
The use of a general EOS has also an influence on the construction of the interface flux functions $\boldsymbol{\gamma}$. For the convective part, the Riemann solver needs to be adapted. The extension of the SLAU scheme is straightforward as it only requires the computation of the pressure and the speed of sound according to the EOS considered. The formulation presented in [21] has been implemented for this scheme. For the HLLC Riemann solver, the Roe average needs to be generalized. The approach in [22] has been followed.
For the diffusive part, the component of the jacobian $\mathbf{D}$ of the diffusive flux with respect to the gradient of the conservative variables associated to the heat flux is extended to a general EOS.

2. Implicit time integration and jacobians

Whether it is to run steady or unsteady simulations, the selection of implicit time stepping schemes allows to alleviate the constraints on the time step associated to explicit schemes. These are particularly restrictive for the high temperature reentry flows around a space debris involving interfacial phenomena and compressible condensed phase, which makes the equations very stiff. In Argo, the standard scheme used is the multistep Backward Differentiation Formula at second order (BDF2). Applied to Eq. (14), the resulting algebraic system for the unknown nodal degrees of freedom $\mathbf{U}$ at the time step $t+2$ is

$$\mathbf{M}\frac{\left(3/2\mathbf{U}^{t+2} - 2\mathbf{U}^{t+1} + 1/2\mathbf{U}^{t}\right)}{\Delta t} = \mathcal{R}^{CD}(\mathbf{U}^{t+2}), \quad \text{with} \quad \mathcal{R}^{CD} = CV - CI - DV + DI + DT - DP, \quad (19)$$

where $\mathbf{M}$ is the mass matrix, CV is the convective volume term, CI the convective interface term, DV the diffusive volume term, DI the direct diffusive interface term, DT the symmetric term and DP the penalty term. The resolution of the non-linear system (19) at each time step is done by a Newton-Raphson (NR) algorithm, which requires in turn the resolution of a linear system at each iteration. A direct Gauss solver as well as iterative methods are implemented in Argo to solve this system. The linearization brought by the NR method involves the computation of the Jacobian matrices with respect to the conservative variables for each term of the right-hand side $\mathcal{R}^{CD}$. The analytical Jacobian of the convective flux has been generalized to generic EOS and its derivation is explained in Sec. VII.B. The Jacobian of the diffusive flux that is needed to express the Jacobians of the DV and DI terms and the one of the DT term are computed by finite differences for the moment but their analytical derivation for a general EOS should be feasible, based on the general expressions given in [23]. The Jacobian of the numerical flux function $\mathcal{H}$ for the CI term is also computed by finite differences but it could be expressed analytically as well, for certain Riemann solvers, taking inspiration from [24]. Having exact analytical Jacobians can improve significantly the performance of the solver.
What is important to point out here is that the analytical expressions of these different Jacobians, although quite complex, only involves the four thermodynamic derivatives of Eqs. (8), which provides a fully general adaptation of the implicit scheme to generic analytical EOS.

C. The extended DGM (XDGM)

In the context of multiphase flows, the discretization of the governing equations is made particularly difficult by the fact that the jump in the physical properties between the different phases must be resolved, with in addition the possible presence of phase transformation (like the evaporation of the molten layer in the case of space debris ablation) or surface tension phenomena. Two main approaches in the numerical treatment of interfaces separating different phases can be identified: the diffuse interface and the sharp interface approaches. In the first one, the discontinuities across the

interface are regularized over several grid cells, which facilitates the implementation. This simplification of the problem is at the expense of the preservation of the order of convergence of the method around the interface.
Since the phenomena taking place at the the gas-liquid interface drive the physics of the aerothermal ablation process, it is essential to preserve high accuracy in this region. The second and more complex approach preserves the order of the method by modifying the numerical scheme to capture the jumps across the interface, without any smoothing. This is realized by coupling at the interface the field equations valid in the bulk of each phase through the resolution of some jump conditions.
The extended discontinuous Galerkin method (XDGM), first proposed in [25] and further developed in [26] or [27], has been implemented in Argo by Schrooyen et al. and first results were presented in [11, 28]. The present work takes advantage of this novel method by coupling it to the generalized flow equations of Sec. III.

1. Representation of the interface

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:

$$\frac{\partial \varphi}{\partial t} + \nabla \cdot (\varphi \mathbf{v}_\Gamma) = \varphi \nabla \cdot (\mathbf{v}_\Gamma), \tag{20}$$

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 $\mathbf{v}_\Gamma$ is the velocity of the interface.
One of the most interesting feature of the XDGM is that the mesh is kept fixed and does not need to conform with the interface. Since the remeshing process can be time-consuming, especially if the interface deforms a lot like in the targeted application, this property is a real asset. Also for this kind of problems, the position of the interface inside the computational domain cannot be predicted prior to the simulation. It is therefore not possible to refine the mesh in the regions where one expects the interface to evolve, although it is often necessary to do so when using diffuse interface methods.

2. Extension of the finite element space inside the cut elements

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. The principle of the method is to use two distinct polynomial representations of the solution on the same element cut by an interface and to couple them through conditions across the interface in order to obtain a single discontinuous representation (for the solution itself or its first derivative). This concept is illustrated in Fig. 2.
The number of degrees of freedom on each element cut by the interface is doubled in order to provide separate degrees of freedom for each phase. The two representations are therefore defined on the entire element but the integrals in Eq. (14) are only performed over one part or the other of the cut element, depending which set of equations (e.g. for the gas or the liquid) is considered. These integrals have to be performed over the volume $\Omega_{m,i}$, over the element edges $\partial\Omega_{m,i}$ (the subscript i indicating the part of the element associated to the i-th phase) and over the interface Γ_m. The contribution of each phase on the cut element is then summed up to obtain the residual of the numerical scheme over this element. Note that the treatment of the other elements remains identical to the DGM. The contribution of a single cut element filled with two different phases to the global weak form can be written as

$$\begin{aligned}
\sum_{i=1}^{2} \Bigg[& \int_{\Omega_{m,i}} \mathbf{v}^h_{m,i} \cdot \frac{\partial \mathbf{u}^h_{m,i}}{\partial t} \, \mathrm{d}V \\
& \underbrace{- \int_{\Omega_{m,i}} \nabla \mathbf{v}^h_{m,i} : \mathbf{F}^c(\mathbf{u}^h_{m,i}) \, \mathrm{d}V}_{CV_{m,i}} + \underbrace{\int_{\partial\Omega_{m,i}} \boldsymbol{\gamma}^c(\mathbf{u}^{h,+}_{m,i}, \mathbf{u}^{h,-}_{m,i}) \, \mathrm{d}S}_{CS_{m,i}} + \underbrace{\int_{\Gamma_m} \boldsymbol{\gamma}^c(\mathbf{u}^h_{m,i}, \mathbf{u}^h_{m,j}) \, \mathrm{d}S}_{C\Gamma_{m,i}} \\
& \underbrace{+ \int_{\Omega_{m,i}} \nabla \mathbf{v}^h_{m,i} : \mathbf{F}^d(\mathbf{u}^h_{m,i}, \nabla \mathbf{u}^h_{m,i}) \, \mathrm{d}V}_{DV_{m,i}} \underbrace{- \int_{\partial\Omega_{m,i}} \boldsymbol{\gamma}^d(\mathbf{u}^{h,+}_{m,i}, \mathbf{u}^{h,-}_{m,i}) \, \mathrm{d}S}_{DS_{m,i}} \underbrace{- \int_{\Gamma_m} \boldsymbol{\gamma}^d(\mathbf{u}^h_{m,i}, \mathbf{u}^h_{m,j}) \, \mathrm{d}S}_{D\Gamma_{m,i}} \Bigg], \quad j = \{2, 1\},
\end{aligned} \tag{21}$$

where the interfacial fluxes integrated over the interface Γ_m couple the two solutions and lead to a single discontinuous representation of the solution inside the element. At this point, it is important to highlight the difference between

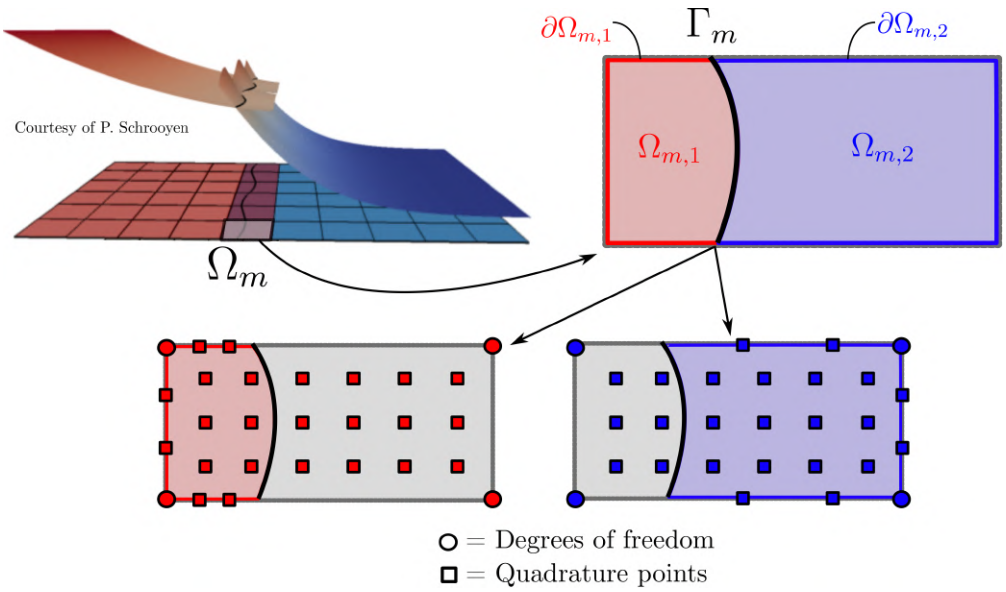


Fig. 2 Principle of the XDGM. The degrees of freedom are doubled on the cut elements in order to capture exactly the discontinuity at the phase interface Γ_m by coupling across it the two representations via the computation of fluxes. The residual is assembled by computing the integrals of the weak form inside the volume, on the element boundaries and along the interface, on each side of the interface.

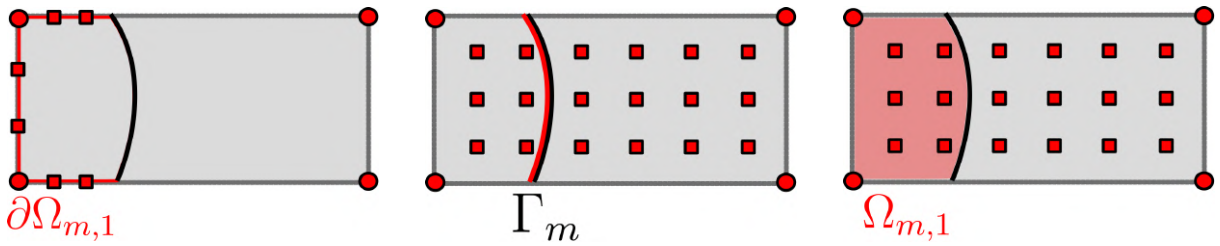
the way the interface flux functions are computed. For the terms CS and DS, it is computed in the same way as it is done in the DGM, i.e. by first collocating the solution to the quadrature points on the element boundaries and then by expressing $\boldsymbol{\gamma}^c(\mathbf{u}^{h,+}_{m,i}, \mathbf{u}^{h,-}_{m,i})$ and $\boldsymbol{\gamma}^d(\mathbf{u}^{h,+}_{m,i}, \mathbf{u}^{h,-}_{m,i})$ in terms of the solution in the element considered ($\mathbf{u}^{h,+}_{m,i}$) and the solution on the boundaries of the adjacent elements ($\mathbf{u}^{h,-}_{m,i}$). The only difference is that this could be realized on a segment of an element boundary if this boundary is cut by the interface. The computation of the terms $C\Gamma$ and $D\Gamma$ use instead the solution collocated to the quadrature points inside the element volume. The flux $\boldsymbol{\gamma}^c(\mathbf{u}^h_{m,i}, \mathbf{u}^h_{m,j})$ and $\boldsymbol{\gamma}^d(\mathbf{u}^h_{m,i}, \mathbf{u}^h_{m,j})$ are therefore obtained with the solution related to the phase being considered ($\mathbf{u}^h_{m,i}$) and the solution of the other phase ($\mathbf{u}^h_{m,j}$). These explanations will probably become more clear in the next section.
It has to be noted that this approach can also be applied to impose immersed boundary conditions on a body surface. In this case, the equations of a single phase are solved. In the element cut by the zero-iso contour of the level-set, which represents the solid surface, the degrees of freedom are still doubled. But to the ones associated to the solid part of the element, the boundary conditions are directly assigned such that these boundary conditions are imposed on the equations being solved. The third test case, presented in Sec. V.C, compares this approach with the standard imposition of boundary conditions.

3. Highly accurate spatial integration

The capacity of the XDGM of reaching high-order accuracy is directly linked to the accuracy with which the integrals in Eq. (21) can be estimated. Contrary to standard elements on which standard quadrature rules can be applied to evaluate the integrals, for a cut element the integrals need to be computed over surfaces which are only known implicitly and that can have arbitrary shapes. A special procedure for numerical integration is therefore requested. In this case, a hierarchical moment-fitting (HMF) method is used to evaluate in a highly accurate way any integrals on complex geometries [29]. It can be divided in three steps, illustrated by Fig. 3:

(a) The intersection points of the level-set with the edges of the cut element are first found to identify the segments composing $\partial\Omega_{m,i}$. In two dimensions, these intersections define inevitably 1-D segments that can be mapped on a parametric 1-D element such that a standard Gaussian quadrature rule can be applied to compute exactly the integrals $\int_{\partial\Omega_{m,i}} \bullet \, \mathrm{d}S$ in Eq. (21);

(b) The second step consists in the evaluation of the integral along the interface $\int_{\Gamma_m} g \, \mathrm{d}S$ of a certain function g which is not known explicitly. The idea of the method is to create a quadrature rule based on a set of explicitly



(a) Integration on the segments with classical quadrature rule **(b) Integration along the interface with QP inside the volume** **(c) Integration on the cut volume with QP inside the volume**

Fig. 3 Hierarchy of the moment-fitting method to compute integrals on cut elements

known functions $\{f_j\}_{j=1,\dots,M}$ that will be used after to evaluate the integral of g. These functions are typically the set of shape functions $\boldsymbol{\phi}_m$ on each element. In particular, one can define a divergence-free basis $\{\mathbf{f}_k\}_{k=1,\dots,K}$ associated to the space of the functions considered, and apply Gauss' theorem to obtain

$$\int_{\Gamma_m} \mathbf{f}_k \cdot \mathbf{n}_{\Gamma_m} \, \mathrm{d}A = -\int_{\partial\Omega_{m,i}} \mathbf{f}_k \cdot \mathbf{n} \, \mathrm{d}A \, . \tag{22}$$

The right-hand side term of Eq. (22) can be computed easily since the boundaries $\partial\Omega_{m,i}$ are known explicitly. The goal then is to find the weights w_n of the quadrature rule for the left-hand side term

$$\int_{\Gamma_m} \mathbf{f}_k \cdot \mathbf{n}_{\Gamma_m} \, \mathrm{d}A = \sum_{n=1}^{N} w_n \mathbf{f}_k(\mathbf{x}_n) \cdot \mathbf{n}_{\Gamma_m}(\mathbf{x}_n) \, . \tag{23}$$

If the position of the quadrature nodes $\mathbf{x}_n$ ($n = 1, \dots, N$) is appropriately predefined, the combination of Eq. (22) and Eq. (23) gives a linear system

$$\begin{bmatrix} \mathbf{f}_1(\mathbf{x}_1) \cdot \mathbf{n}_{\Gamma_m}(\mathbf{x}_1) & \dots & \mathbf{f}_1(\mathbf{x}_N) \cdot \mathbf{n}_{\Gamma_m}(\mathbf{x}_N) \\ \vdots & \ddots & \vdots \\ \mathbf{f}_K(\mathbf{x}_1) \cdot \mathbf{n}_{\Gamma_m}(\mathbf{x}_1) & \dots & \mathbf{f}_K(\mathbf{x}_N) \cdot \mathbf{n}_{\Gamma_m}(\mathbf{x}_N) \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_N \end{bmatrix} = \begin{bmatrix} -\int_{\partial\Omega_{m,i}} \mathbf{f}_1 \cdot \mathbf{n} \, \mathrm{d}A \\ \vdots \\ -\int_{\partial\Omega_{m,i}} \mathbf{f}_K \cdot \mathbf{n} \, \mathrm{d}A \end{bmatrix} , \tag{24}$$

that can be solved to obtain the weights of the quadrature rule. These weights are then used to evaluate $\int_{\Gamma_m} g \, \mathrm{d}S$, or in the present case, the terms $\int_{\Gamma_m} \boldsymbol{\gamma}^c \, \mathrm{d}S$ and $\int_{\Gamma_m} \boldsymbol{\gamma}^d \, \mathrm{d}S$ in Eq. (21). Note that it is important to ensure that the number of quadrature points N exceeds the number of moments M, i.e. that $N \geq \beta M$ holds, where β is a safety factor. This condition implies that the linear system Eq. (24) is under-determined and has to be solved in a least-squared sense. The condition $N \geq \beta M$ avoids the introduction of an additional error in the solution of this system;

(c) By defining now $\mathbf{f}_j$ such that $\nabla \cdot \mathbf{f}_j = f_j$ and by applying Gauss' theorem, the volume integrals over the basis functions f_j can be decomposed as

$$\int_{\Omega_{m,i}} f_j \, \mathrm{d}V = \int_{\partial\Omega_{m,i}} \mathbf{f}_j \cdot \mathbf{n} \, \mathrm{d}A + \int_{\Gamma_m} \mathbf{f}_j \cdot \mathbf{n}_{\Gamma_m} \, \mathrm{d}A \, , \tag{25}$$

where the first right-hand side term is directly obtained by classical quadrature rule and the second right-hand side term can be determined as explained in the second step. This leads again to a linear system whose solutions are the weights to integrate the terms $\int_{\Omega_{m,i}} \bullet \, \mathrm{d}V$ in Eq. (21).

Some cut elements can become problematic when the ratio between the surface occupied by one phase in the element is much smaller than the surface occupied by the other phase. Such elements are said to have a small volume fraction and could render the HMF method impractical due to the extreme time-step restriction they impose. A cell-agglomeration strategy, discussed in [27], is included in the method to avoid this problem. At each time step, a list of elements having a volume fraction smaller than a user-defined threshold is elaborated. Then, these elements are merged with other

neighbouring elements having the largest volume fraction to create new agglomerated elements. The computations are finally performed on these new elements. This procedure can decrease locally a bit the accuracy but it is essential to assure the convergence of the method.

V. Verification test cases

A. Riemann test cases

The one-dimensional Riemann problem gives an exact solution to the complex non-linear Euler equations and it is therefore very common to consider it in order to test the accuracy of computational fluid codes. Initially, two fluids at rest having different thermodynamic states are separated by a diaphragm. At time $t = 0$, the sudden breakdown of the diaphragm generates a high-speed flow, which propagates in the tube along with some characteristic waves that can be of different nature. Here, the well-known shock-tube problem is treated which consists initially in a high-pressure fluid in the left part of the tube and a low-pressure one in the right part. In this configuration, the fluid at high pressure expands through an expansion wave and pushes the fluid of the right part. The compression of the low-pressure fluid generates a shock wave propagating to the right. Finally, the expanded fluid is separated from the compressed fluid by a contact discontinuity. This situation is solved for the perfect gas and the van der Waals EOS and is compared to the exact solution that can be computed as detailed in Sec. VII.A. The initial conditions listed in Table 1 are taken from [30], except for the parameters of the van der Waals EOS which have been chosen arbitrarily such that differences with the case of the perfect gas EOS are noticeable. Information on the domain and the final time of the simulation are also given in Table 1. The SLAU and HLLC approximate Riemann solvers were selected in this work. The order of the polynomial

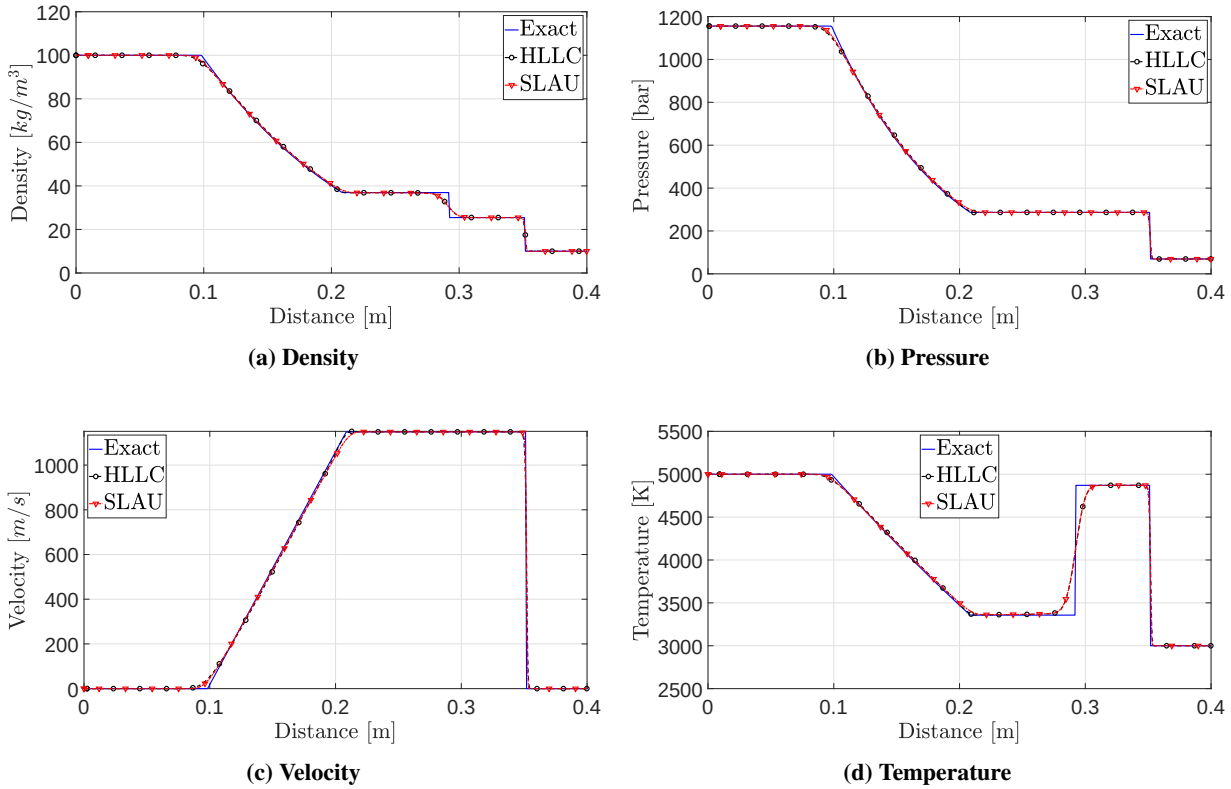
Table 1 Parameters selected for the Riemann test case

($\rho\,[kg/m^3]$, $v\,[m/s]$, $T\,[K]$)	
Left state	Right state
(100, 0, 5000)	(10, 0, 3000)

Parameters	Values
C_v	$577.8\,[J/kg.K]$
R	$231.11\,[J/kg.K]$
VdW EOS parameters	
A	0.5 (SI)
B	0.005 (SI)

Parameters	Values
# elements	1000
Length	$0.4\,[m]$
Diaphragm position	$0.2\,[m]$
Final time	PG: $80 \cdot 10^{-6}\,[s]$
	VdW: $60 \cdot 10^{-6}\,[s]$

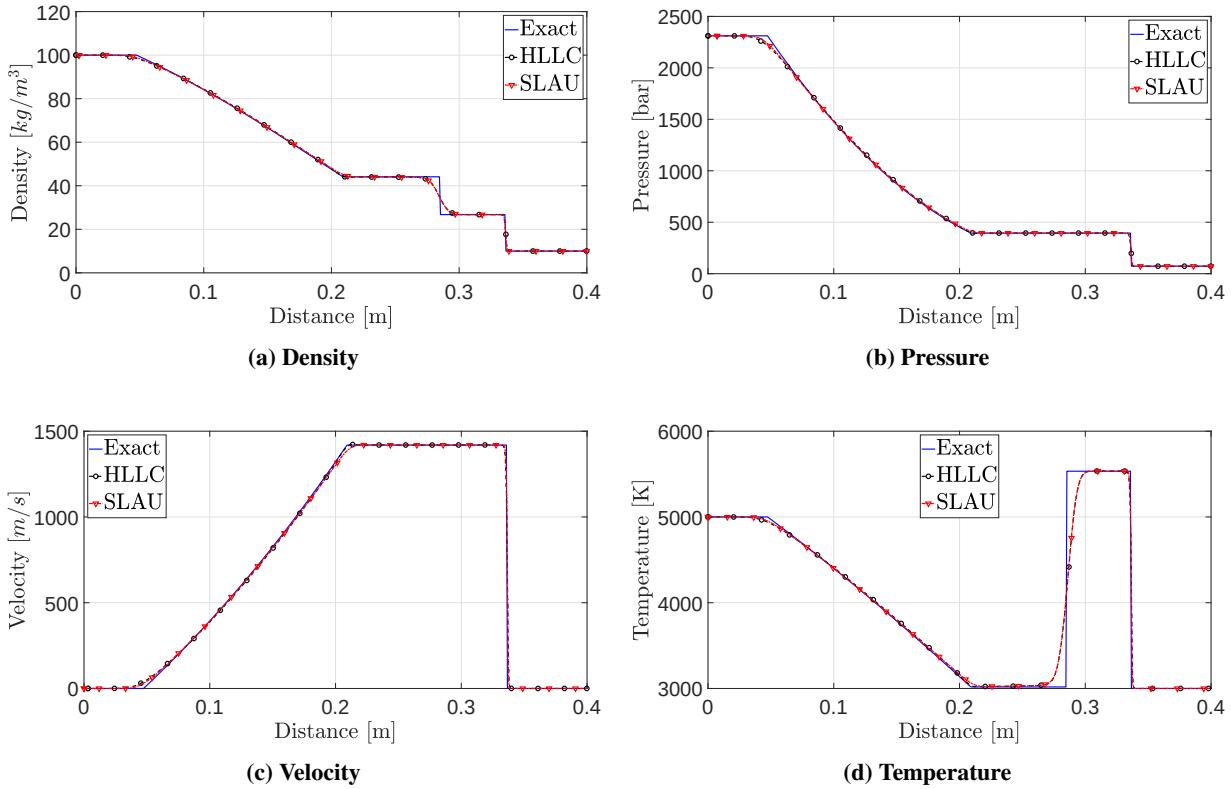
interpolation of the DGM solver inside each element is kept equal to 0, which corresponds to a classical first-order FVM scheme. If higher orders are considered, some spurious oscillations appear near the discontinuities due to a lack of numerical diffusion. Artificial viscosity can be used to suppress them but it was not investigated within this work.
The different graphs in Fig. 4 and Fig. 5 of the density, pressure, velocity and temperature show good agreement between the numerical and the exact solution, for both Riemann schemes, independently of the EOS selected. This inviscid test case proves therefore the validity of the adaptation of the SLAU and HLLC schemes to general EOS, the new analytical expression for the Jacobian of the convective flux and the computation of the pressure via an external generic routine.



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

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

Fig. 4 Comparison between the exact solution and the numerical solutions obtained with SLAU and HLLC schemes at interpolation order $p = 0$ for the Riemann test case with perfect gas law



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

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

Fig. 5 Comparison between the exact solution and the numerical solutions obtained with SLAU and HLLC schemes at interpolation order $p = 0$ for the Riemann test case with van der Waals EOS

B. Comparison with SU2 on a cylinder test case

In order to include viscous effects, the classical situation of a flow around a fixed cylinder in a channel is studied. The Reynolds number is chosen such that the flow is steady. The SU2 open-source CFD software, which was recently extended to work with non-ideal EOS such as the van der Waals EOS [24], is used to compare the results obtained with Argo, with the perfect gas EOS first and then with the van der Waals EOS. Some characteristics of these two codes are listed in Table 2. The boundary conditions, the thermodynamic parameters and the values of the dimensionless numbers are presented in Table 3.

The comparison of the meshes used with one code or the other in Fig. 6 highlights the important difference in term of refinement level. This simple test case already brings out the benefit of using high-order methods such as the DGM. For a number of degrees of freedom which is lower, Argo is able to produce results with the same level of accuracy than the one of SU2, with a shorter CPU time on this configuration.

In Fig. 7, a qualitative comparison between the two codes on the iso-contours of the Mach number is shown. Whether for the perfect gas case or the VdW EOS case, there is a good matching between the two solvers. It is difficult to attribute to the visible discrepancies between the results any precise cause as many parameters differ between the two codes. But the significant differences between the two meshes are most probably the main source of discrepancies.

A quantitative comparison on the values of the drag coefficients is also presented in Table 4. Even if the errors are small, the trend in the evolution of the drag coefficient values for the perfect gas law and the VdW EOS are different between the two codes. Again, it is difficult to provide a rigorous explanation about this observation. The use of curved elements with Argo to represent more accurately the cylinder surface might be one of the main reasons.

From the analyses made just above, it can be advanced that this test case establishes the ability of the new Argo feature to deal with non-ideal EOS.

Table 2 Characteristics of the CFD codes

	Argo	SU2
Spatial discretization	DGM	FVM
Order of convergence	5 (4^{th} interpolation order)	2 (MUSCL reconstruction)
Riemann solver	SLAU	Roe
Time discretization	BDF2	Euler implicit

Table 3 Boundary conditions, EOS parameters and dimensionless numbers for the cylinder test case

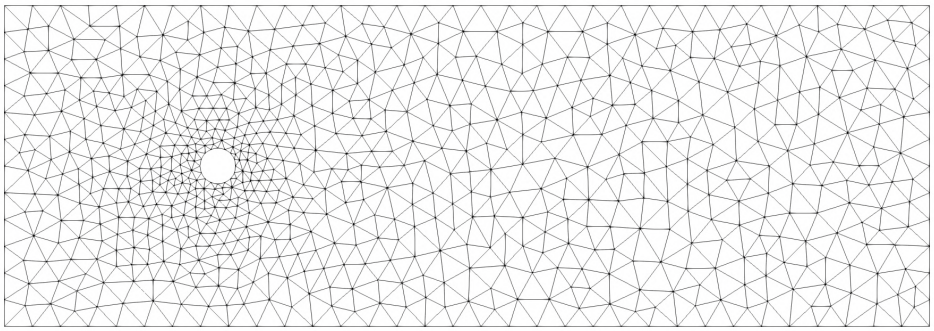
Boundary	Variables	Values
Inlet	Velocity	$30\,[m/s]$
	Temperature	$293.15\,[K]$
Cylinder	Velocity	$0\,[m/s]$
	Temperature	$293.15\,[K]$
Outlet	Pressure	$101320\,[Pa]$

Parameters	Values
C_v	$746.3174\,[J/kg.K]$
R	$288.1979\,[J/kg.K]$
VdW EOS parameters	
A	5.0 (SI)
B	0.5 (SI)

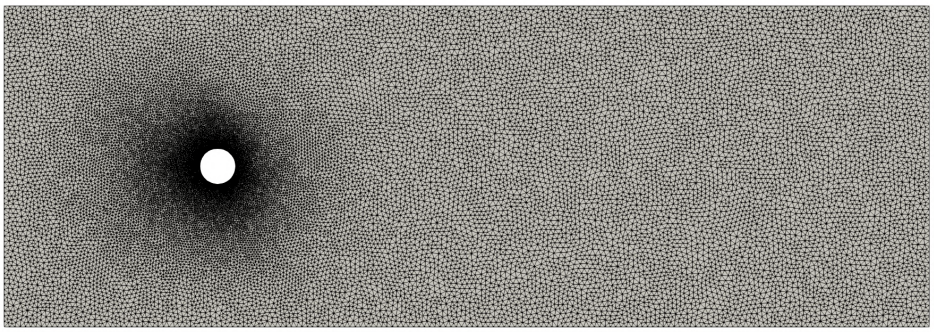
Dimensionless numbers	Values
$Re = \frac{\rho v D}{\mu}$	$20\,[-]$
$Pr = \frac{\mu C_p}{k}$	$1\,[-]$

Table 4 Comparison between the drag coefficients C_D obtained with Argo and SU2. The relative errors are expressed with respect to the SU2 results.

	Argo		SU2		Relative error	
	PG	VdW	PG	VdW	PG	VdW
C_D	2.45202	2.44363	2.46391	2.46465	0.56%	0.85%

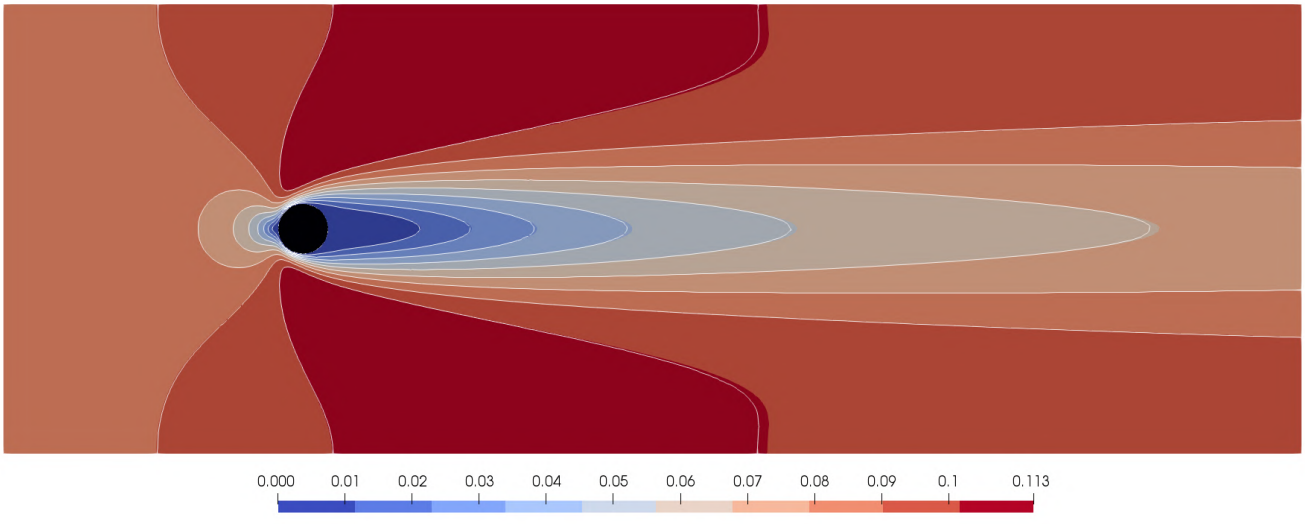


(a) ARGO, 5^{th} order of convergence ($p = 4$), 21570 degrees of freedom

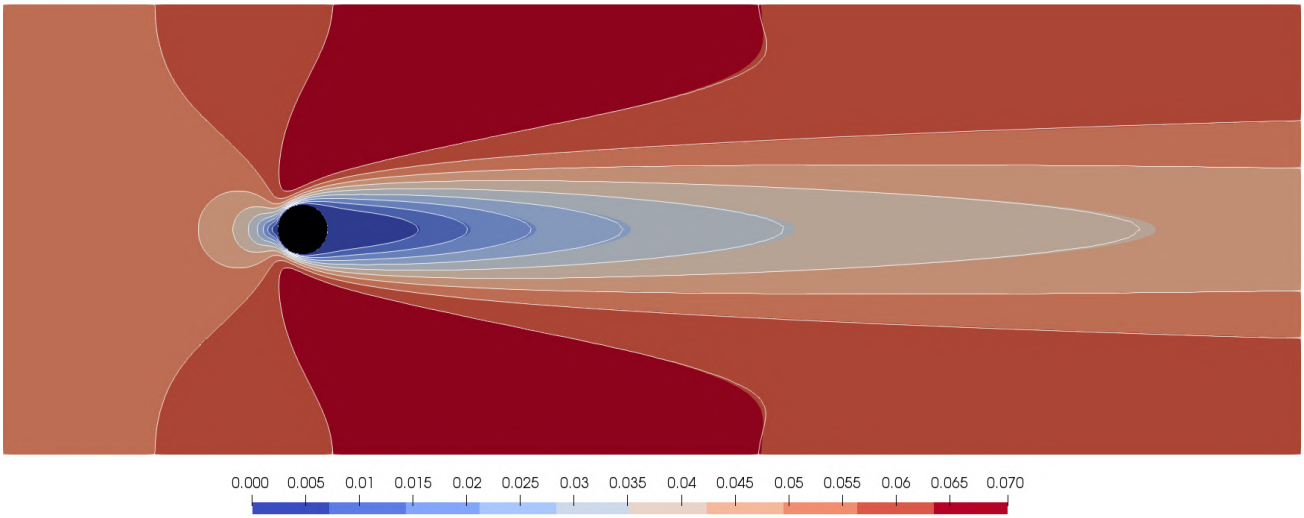


(b) SU2, 2^{nd} order of convergence, 46034 degrees of freedom

Fig. 6 Meshes used for the simulation

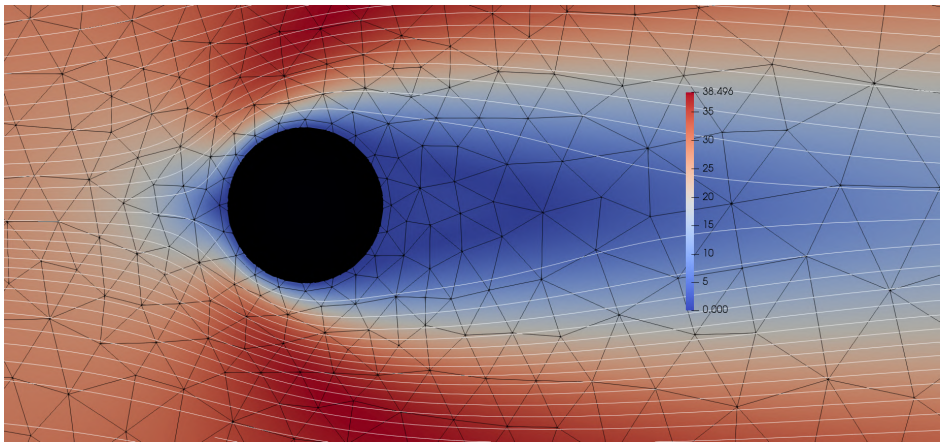


(a) Perfect gas EOS

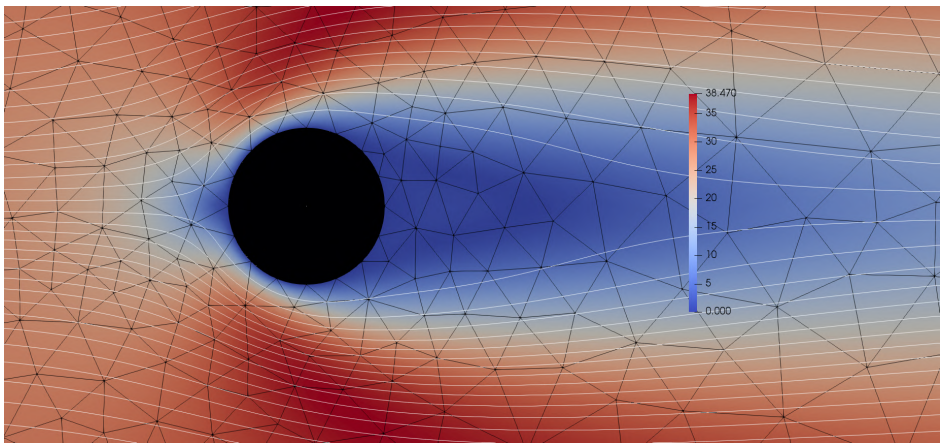


(b) Van der Waals EOS

Fig. 7 Mach number distribution around the cylinder: SU2 (white iso-contours) and ARGO (filled contours)



(a) XDGM



(b) Conforming mesh

Fig. 8 Velocity distribution and streamlines around the cylinder with VdW EOS, and the underlying mesh

C. The cylinder test case with XDGM

This last test case uses the same configuration and boundary conditions as the second test case. In Fig. 8, the distinction between a mesh conforming with the boundary of the cylinder and an immersed boundary cutting several elements is emphasized. For the present test case in which the interface has a simple shape and is fixed, there is no real benefit of using the XDGM. But when the simulation of ablating space debris that involves complex interface dynamics will be tackled, the availability of the XDGM will reveal necessary.

It must be mentioned that only one set of equations outside of the cylinder is solved in this test case. The XDGM is used to impose accurately a wall boundary condition on the cylinder surface but nothing is computed inside it. In future work, multiphase situations will be treated and the XDGM will serve as a sharp interface method coupling the gas phase with the condensed phase through the resolution of jump conditions at the interface.

In Fig. 9, a perfect matching between the pressure iso-contours obtained with a conforming mesh and an immersed interface can be observed. Moreover, the values of the drag coefficients listed in Table 5 appear to be very close by using both approaches. This validates the coupling of the new general EOS module with the XDGM inside Argo.

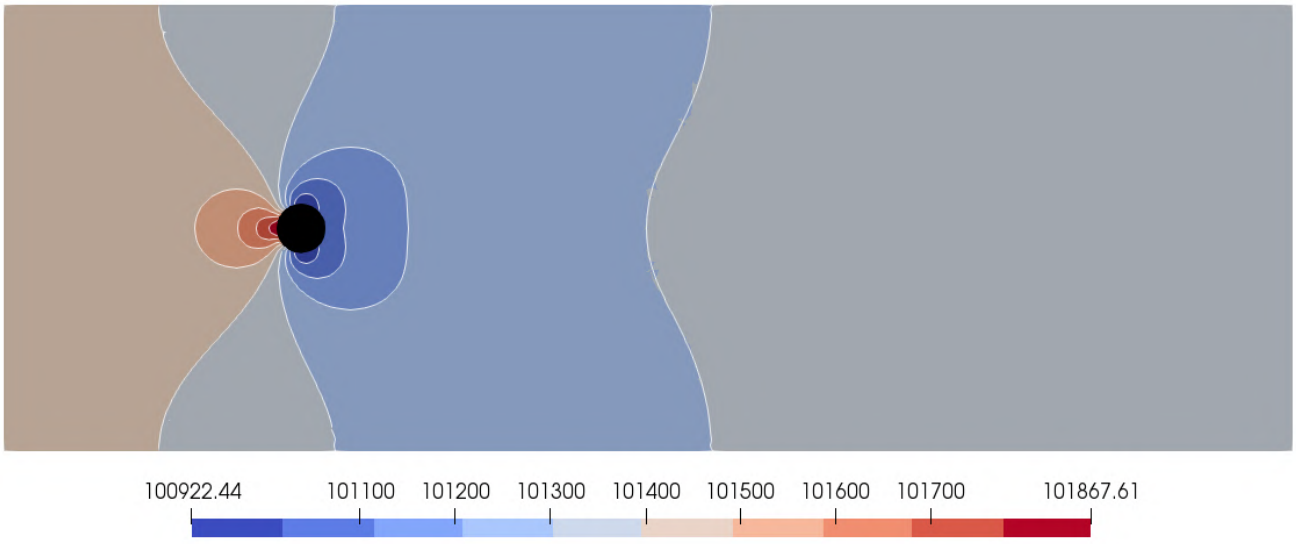


Fig. 9 Pressure distribution around the cylinder with VdW EOS: conforming mesh (white iso-contours) and XDGM (filled contours)

Table 5 Comparison between the drag coefficients C_D obtained with Argo and the VdW EOS, by using a conforming mesh and the XDGM to represent the cylinder. The relative error is expressed with respect to the results with a conforming mesh.

	Argo conforming VdW	Argo immersed VdW	Relative error
C_D	2.44363	2.444526	0.037%

VI. Perspectives

As stated previously, the theory and the results presented in this work are only about two elements necessary for the elaboration of a new module inside the Argo platform that will include other. To put into perspective what has been achieved and what remains to be done, here follows an overview, supported by Fig. 10, of the main building blocks of

the numerical tool to be developed in order to simulate in an unified way the aerothermal ablation of a metallic space debris.

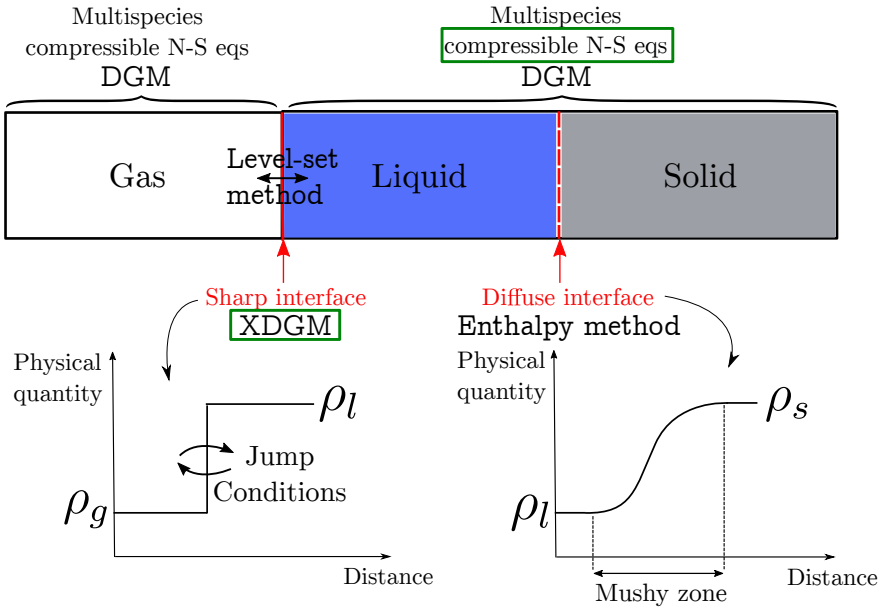


Fig. 10 Schematic representation of the three-phase problem to be studied and the associated main building blocks of the numerical tool. The two elements covered in this paper are surrounded by green boxes.

Governing equations for the three phases Simulate the transport of different species inside the gas and the condensed phase (i.e. liquid + solid) to take into account high-temperature effects is the first element of the tool. For the gas phase, a module solving the multi-species Navier-Stokes equations has been incorporated in Argo to study the ablation of TPS [31]. The thermophysical properties were obtained by the external library Mutation++ [32]. This contribution will be exploited to implement compressible multi-species equations for the condensed phase. One of the main challenges will be the accurate determination over a wide range of temperatures of the thermophysical properties for liquid and solid metallic alloys, for which there is no model comparable to those used for the gas. It is notably foreseen to couple the solver with an external library that would provide high accuracy EOS. This task will benefit from the adaptation of the solver to general EOS conducted in the present work.

Treatment of the liquid-solid interface and the melting For multicomponent materials like metallic alloys undergoing melting, the interfacial region is made of a mixture of solid and liquid components. This region, called the mushy zone, has a finite thickness and its modelization requires therefore a diffuse interface method. The popular approach is to reformulate the problem in such a way that the jump conditions between the two phases are implicitly incorporated in a new form of equations, which applies over the entire liquid-solid domain. A single set of equations is therefore solved to calculate field variables for the solid, liquid and mushy region. The enthalpy method, which meet these criteria, is available in Argo with incompressible equations [33]. It is planned to extend this method to compressible multi-species equations.

Treatment of the gas-liquid interface Contrary to the solid-liquid interface, the gas-liquid interface can be considered as a sharp discontinuity from a macroscopic point of view. The XDGM, developed in [11], will be used to couple the equations of the gas and the condensed phases. This will necessitate the resolution of jump conditions between the two set of equations, accounting for evaporation and surface tension at the interface. The use of this highly accurate sharp interface method in this work to represent an immersed solid boundary and its coupling to a general EOS model are the first steps towards this objective.

VII. Conclusions

The existing capabilities of the multi-physics Argo platform relying on the DGM were extended to the resolution of conservation laws obeying more general EOS than the perfect gas law. The implementation was verified against the analytical solution of a benchmark case as well as by performing code-to-code comparisons. This new functionality was then coupled to an available sharp interface method, called the XDGM. The results demonstrated the ability of this method to represent very accurately an immersed solid boundary. The results presented in this paper are the first steps in the development of an unified numerical tool for producing high-fidelity simulations of the aerothermal ablation of metallic re-entering space debris.

Appendix

A. Exact Riemann solver for general EOS

The exact Riemann solver for general EOS has been built based on [30] and [34]. One considers the Euler equations governed by a general EOS under the form $p = p(\rho, T)$ with a piecewise constant initial condition at time $t = 0$ with a discontinuity located at $x = x_0$:

$$\mathbf{U} = \begin{cases} \mathbf{U}_L & \text{if} \quad x < x_0 \,, \\ \mathbf{U}_R & \text{if} \quad x > x_0 \,, \end{cases} \tag{26}$$

where the set of unknowns is chosen as $\mathbf{U} = (\rho,\, u,\, p)$. This situation of hyperbolic systems of equations with piecewise constant initial condition is referred as a Riemann problem. The solution at $t > 0$ is made of piecewise constant states separated by curves in the plane (x, t), called the characteristics. Along these characteristics, certain quantities, called the invariants, are constant. Depending on the nature of the eigenvalues of the system, different flow features might appear: rarefaction waves, shocks or contact discontinuities. The first step is to determine the intermediate solutions in each region delimited by these special features and then to compute the solution in all the domain for a given time.

1. Determination of the intermediate states

The intermediate states one would like to determine at the left and at the right of the contact discontinuity are denoted by $\bullet_L^*$ and $\bullet_R^*$ respectively. The solver is based on the properties that the pressure and the velocity are continuous across a contact discontinuity. The density in the left and right intermediate states is then computed by solving the following non-linear system:

$$\begin{cases} p_L^*(\rho_L^*) - p_R^*(\rho_R^*) = 0 \,, \\ u_L^*(\rho_L^*) - u_R^*(\rho_R^*) = 0 \,, \end{cases} \tag{27}$$

with:

$$p_L^* = \begin{cases} p_{shock}(\mathbf{U}_L, \rho_L^*) & \text{if} \quad \rho_L^* > \rho_L \,, \\ p_{rar}(\mathbf{U}_L, \rho_L^*) & \text{if} \quad \rho_L^* < \rho_L \,, \end{cases} \qquad p_R^* = \begin{cases} p_{shock}(\mathbf{U}_R, \rho_R^*) & \text{if} \quad \rho_R^* > \rho_R \,, \\ p_{rar}(\mathbf{U}_R, \rho_R^*) & \text{if} \quad \rho_R^* < \rho_R \,, \end{cases} \tag{28}$$

$$u_L^* = \begin{cases} u_{shock}(\mathbf{U}_L, \rho_L^*) & \text{if} \quad \rho_L^* > \rho_L \,, \\ u_{rar}(\mathbf{U}_L, \rho_L^*) & \text{if} \quad \rho_L^* < \rho_L \,, \end{cases} \qquad u_R^* = \begin{cases} u_{shock}(\mathbf{U}_R, \rho_R^*) & \text{if} \quad \rho_R^* > \rho_R \,, \\ u_{rar}(\mathbf{U}_R, \rho_R^*) & \text{if} \quad \rho_R^* < \rho_R \,, \end{cases} \tag{29}$$

where the rar subscript indicates a solution computed through a rarefaction wave while $shock$ is for a solution across a shock wave.

Rarefaction wave Starting from the expression of the Riemann invariant for a rarefaction wave, the velocity across the fan can be computed, from a known left or right state, as

$$u_{rar}(\rho) = \begin{cases} u_L - \int_{\rho_L}^{\rho} c(\rho')/\rho' \, \mathrm{d}\rho' \,, \\ u_R + \int_{\rho_R}^{\rho} c(\rho')/\rho' \, \mathrm{d}\rho' \,, \end{cases} \tag{30}$$

The challenge here stems from the evaluation of the Riemann invariant integral which has to be performed for constant entropy. To do so, the speed of sound c is expressed in the general form

$$c^2 = \frac{-C_p}{C_v} \left(\frac{\partial T}{\partial \rho}\right)_p \Bigg/ \left(\frac{\partial T}{\partial p}\right)_\rho \,, \quad \text{with} \quad C_p = C_v - \frac{T}{\rho^2} \left(\frac{\partial \rho}{\partial T}\right)_p \Bigg/ \left(\frac{\partial T_{is}}{\partial p}\right)_\rho \,, \tag{31}$$

in which the temperature T_{is} is computed along an isentrope from

$$\frac{C_v}{T_{is}}\mathrm{d}T = \frac{1}{\rho^2}\frac{\mathrm{d}\rho}{(\partial T/\partial p)_\rho} \,. \tag{32}$$

The pressure is expressed directly by the EOS with this isentropic temperature:

$$p_{rar}(\rho) = p(\rho, T_{is}) \,. \tag{33}$$

Shock The pressure and the velocity across a shock from a left or right known state are determined, as function of the density, by rearranging the Rankine-Hugoniot jump relations:

$$\frac{1}{2}\left(p_{shock} - p_{L,R}\right)\left(1/\rho_{L,R} - 1/\rho\right) + e(p_{shock}, \rho) - e(p_{L,R}, \rho_{L,R}) = 0 \,, \tag{34}$$

where e is the internal energy and

$$u_{shock}(\rho) = \begin{cases} u_L - \sqrt{-(p_{shock}(\rho) - p_L)(1/\rho - 1/\rho_L)} \,, \\ u_R + \sqrt{-(p_{shock}(\rho) - p_R)(1/\rho - 1/\rho_R)} \,. \end{cases} \tag{35}$$

2. Computation of the global solution

From the intermediate states, it is possible to represent the solution in all the domain for any time. To illustrate the procedure, the example of the shock-tube problem treated in this work is taken. In this situation, a rarefaction wave propagates to the left, a shock wave to the right and a contact discontinuity lies in between the two. The different regions are computed as follows:

- On the left of the rarefaction wave: $x - x_0 < t \cdot V_{head} \,, \quad V_{head} = u_L - c_L(\rho_L, p_L)$

$$\mathbf{U} = \mathbf{U}_L \tag{36}$$

- Inside the rarefaction wave: $t \cdot V_{head} < x - x_0 < t \cdot V_{tail} \,, \quad V_{tail} = u_L^* - c_L^*(\rho_L^*, p_L^*)$

$$\rho \,:\, u_{rar}(\rho) - u(\rho) = 0 \,, \quad \text{with} \quad u(\rho) = \frac{x - x_0}{t} + c(\rho, T_{is}) \tag{37}$$

$$T = T_{is}(\rho) \tag{38}$$

$$p = p(\rho, T) \tag{39}$$

- Between the rarefaction and the contact: $t \cdot V_{tail} < x - x_0 < t \cdot V_{contact} \,, \quad V_{contact} = u_L^* = u_R^*$

$$\mathbf{U} = \mathbf{U}_L^* \tag{40}$$

- Between the contact and the shock: $t \cdot V_{contact} < x - x_0 < t \cdot V_{shock} \,, \quad V_{shock} = u_R - \frac{1}{\rho_R}\frac{u_R - u_R^*}{1/\rho_R - 1/\rho_R^*}$

$$\mathbf{U} = \mathbf{U}_R^* \tag{41}$$

- In the right of the shock: $x - x_0 > t \cdot V_{shock}$

$$\mathbf{U} = \mathbf{U}_R \tag{42}$$

B. Construction of the Jacobian of the convective flux

The vector of conservative variables in two dimensions is rewritten here for convenience:

$$\mathbf{u} = \left(\rho,\, \rho v_x,\, \rho v_y,\, \rho E\right)^T = (u_1,\, u_2,\, u_3,\, u_4)^T \,. \tag{43}$$

In the following, the steps to obtain the expression of the Jacobian of the convective flux in the x-direction are explained, based on [35]. The determination of the Jacobian for the y-direction follows exactly the same procedure. The x-component of the convective flux and its associated Jacobian matrix are written respectively as:

$$\begin{aligned} \mathbf{F}_x^c(\mathbf{u}) &= \left(u_2,\ \frac{u_2^2}{u_1} + p,\ \frac{u_2 u_3}{u_1},\ \frac{u_2 u_4}{u_1} + \frac{u_2}{u_1} p \right) , \\ \mathbf{J}_x^c &= \frac{\partial \mathbf{F}_x^c(\mathbf{u})}{\partial \mathbf{u}} . \end{aligned} \tag{44}$$

Since the convective flux F^c depends on the pressure p and the Jacobian matrix is built from the derivatives of F^c with respect to the conservative variables $\mathbf{u}$, it is convenient to introduce a relation which expresses the pressure as a function of the conservative variable vector, p^*, as:

$$\begin{aligned} p(\rho, e) &= p\left(\rho,\ \frac{\rho E}{\rho} - \frac{1}{2} \frac{(\rho v_x + \rho v_y)^2}{\rho^2} \right) = p\left(u_1,\ \frac{u_4}{u_1} - \frac{1}{2} \frac{(u_2 + u_3)^2}{u_1^2} \right) , \\ &\triangleq p^*(\mathbf{u}) . \end{aligned} \tag{45}$$

The different terms of the matrix $\mathbf{J}_x^c$ can then be computed as follows:

$$\begin{aligned} J_x^c(1,1) &= \tfrac{\partial}{\partial u_1}(u_2) = 0 & J_x^c(2,1) &= \tfrac{\partial}{\partial u_1}\left(\tfrac{u_2^2}{u_1} + p \right) = -v_x^2 + \tfrac{\partial p^*}{\partial \rho} \\ J_x^c(1,2) &= \tfrac{\partial}{\partial u_2}(u_2) = 1 & J_x^c(2,2) &= \tfrac{\partial}{\partial u_2}\left(\tfrac{u_2^2}{u_1} + p \right) = 2v_x + \tfrac{\partial p^*}{\partial (\rho v_x)} \\ J_x^c(1,3) &= \tfrac{\partial}{\partial u_3}(u_2) = 0 & J_x^c(2,3) &= \tfrac{\partial}{\partial u_3}\left(\tfrac{u_2^2}{u_1} + p \right) = \tfrac{\partial p^*}{\partial (\rho v_y)} \\ J_x^c(1,4) &= \tfrac{\partial}{\partial u_4}(u_2) = 0 & J_x^c(2,4) &= \tfrac{\partial}{\partial u_4}\left(\tfrac{u_2^2}{u_1} + p \right) = \tfrac{\partial p^*}{\partial (\rho E)} \end{aligned}$$

$$\begin{aligned} J_x^c(3,1) &= \tfrac{\partial}{\partial u_1}\left(\tfrac{u_2 u_3}{u_1} \right) = -v_x v_y & J_x^c(4,1) &= \tfrac{\partial}{\partial u_1}\left(\tfrac{u_2 u_4}{u_1} + \tfrac{u_2}{u_1} p \right) = -v_x H + v_x \tfrac{\partial p^*}{\partial \rho} \\ J_x^c(3,2) &= \tfrac{\partial}{\partial u_2}\left(\tfrac{u_2 u_3}{u_1} \right) = v_y & J_x^c(4,2) &= \tfrac{\partial}{\partial u_2}\left(\tfrac{u_2 u_4}{u_1} + \tfrac{u_2}{u_1} p \right) = H + v_x \tfrac{\partial p^*}{\partial (\rho v_x)} \\ J_x^c(3,3) &= \tfrac{\partial}{\partial u_3}\left(\tfrac{u_2 u_3}{u_1} \right) = v_x & J_x^c(4,3) &= \tfrac{\partial}{\partial u_3}\left(\tfrac{u_2 u_4}{u_1} + \tfrac{u_2}{u_1} p \right) = v_x \tfrac{\partial p^*}{\partial (\rho v_y)} \\ J_x^c(3,4) &= \tfrac{\partial}{\partial u_4}\left(\tfrac{u_2 u_3}{u_1} \right) = 0 & J_x^c(4,4) &= \tfrac{\partial}{\partial u_4}\left(\tfrac{u_2 u_4}{u_1} + \tfrac{u_2}{u_1} p \right) = v_x + v_x \tfrac{\partial p^*}{\partial (\rho E)} \end{aligned}$$

It can be noticed that all the terms of the convective Jacobian can be obtained with the set of conservative variables and the derivatives of the function $p^*(\mathbf{u})$ with respect to the conservative variables:

$$\begin{aligned} \frac{\partial p^*}{\partial \rho} &= \left(\frac{\partial p}{\partial \rho} \right)_e - \left(\frac{\partial p}{\partial e} \right)_\rho \frac{1}{\rho} \left(e - \frac{1}{2} \mathbf{v} \cdot \mathbf{v} \right) , \\ \frac{\partial p^*}{\partial (\rho v_x)} &= -\frac{v_x}{\rho} \left(\frac{\partial p}{\partial e} \right)_\rho , \\ \frac{\partial p^*}{\partial (\rho v_y)} &= -\frac{v_y}{\rho} \left(\frac{\partial p}{\partial e} \right)_\rho , \\ \frac{\partial p^*}{\partial (\rho E)} &= \frac{1}{\rho} \left(\frac{\partial p}{\partial e} \right)_\rho . \end{aligned} \tag{46}$$

Two of the four derivatives of the thermodynamic model of Eqs. (8), namely $\left(\frac{\partial p}{\partial \rho} \right)_e$ and $\left(\frac{\partial p}{\partial e} \right)_\rho$, are needed to build the Jacobian of the convective flux for general EOS.

C. Thermodynamic models

The two thermodynamic models considered for the test case in this work are the perfect gas law and the van der Waals EOS. Their definition are recalled here, based on the general Mie-Grüneisen form of Eq. (5).

1. Perfect gas law

The parameters of the Mie-Grüneisen form are given by

$$\begin{aligned} e_{ref} &= 0\,, \\ p_{ref} &= 0\,, \\ \Gamma &= \gamma - 1\,, \\ e_0 = 0 &\Rightarrow e_T = C_v\,T\,, \end{aligned} \tag{47}$$

where γ is the ratio of specific heats. The resulting volumetric and caloric equations of state are given by:

$$\begin{cases} p &= \rho(\gamma-1)e_T\,, \\ e &= e_T = C_v\,T\,. \end{cases} \tag{48}$$

All the derivatives of these parameters with respect to the density being equal to zero and C_v being constant, the thermodynamic derivatives of Eqs. (8) are:

$$\begin{aligned} \left(\frac{\partial p}{\partial \rho}\right)_{e=\text{const}} &= (\gamma-1)e\,, \\ \left(\frac{\partial p}{\partial e}\right)_{\rho=\text{const}} &= (\gamma-1)\rho\,, \\ \left(\frac{\partial e}{\partial \rho}\right)_{T=\text{const}} &= 0\,, \\ \left(\frac{\partial e}{\partial T}\right)_{\rho=\text{const}} &= C_v\,. \end{aligned} \tag{49}$$

2. Van der Waals EOS

The parameters of the Mie-Grüneisen form are given by

$$\begin{aligned} e_{ref} &= -A\rho\,, \\ p_{ref} &= -A\rho^2\,, \\ \Gamma &= \frac{\gamma-1}{1-B\rho}\,, \\ e_0 = 0 &\Rightarrow e_T = C_v\,T\,, \end{aligned} \tag{50}$$

The resulting equations of state are given by:

$$\begin{cases} p &= \frac{RT}{1/\rho - B} - A\rho^2\,, \\ e &= e_T = C_v\,T - A\rho\,, \end{cases} \tag{51}$$

where $R = (\gamma-1)C_v$. With the expressions of the derivatives of Γ, p_{ref} and e_{ref} with respect to the density, the thermodynamic derivatives needed by the numerical method are:

$$\begin{aligned} \left(\frac{\partial p}{\partial \rho}\right)_{e=\text{const}} &= \frac{e + 2\,A\,B - \rho^2\,A\,B}{\rho(1-\rho\,B)}\left(\frac{\partial p}{\partial e}\right)_{\rho=\text{const}} - 2\,A\,B\,\rho\,, \\ \left(\frac{\partial p}{\partial e}\right)_{\rho=\text{const}} &= \frac{\rho(\gamma-1)}{1-B\,\rho}\,, \\ \left(\frac{\partial e}{\partial \rho}\right)_{T=\text{const}} &= -A\,, \\ \left(\frac{\partial e}{\partial T}\right)_{\rho=\text{const}} &= C_v\,. \end{aligned} \tag{52}$$

Acknowledgments

The first author is supported by a UCLouvain-FSR (Fonds Spécial de Recherche) fellowship granted by the French Community of Belgium. The first author is also grateful to the von Karman Institute for Fluid Dynamics for the nine months fellowship provided during the Research Master program during which this work was started.

References

[1] United Nations, O. f. O. S. A., "Space Debris Mitigation Guidelines of the Committee on the Peaceful Uses of Outer Space," , Vienna, 2010. http://www.unoosa.org/pdf/publications/st_space_49E.pdf.

[2] Grassi, L., Bianchi, S., Destefanis, R., Kanzler, R., Bassaler, P., and Heinrich, S., "Design for demise techniques for medium/Large LEO satellites reentry," *European Space Agency - Publications*, 7th European conference on space debris, 2017. Darmstadt, Germany.

[3] "European Space Material Demisability Database (ESTIMATE)," http://fluidgravity.co.uk/fgewebsite/8ec2ea98-6387-4dbf-ad2b-23ce02a1d3ab/index.php, 2019. Accessed: 2019-03-08.

[4] Bethe, H., and Adams, M., "A theory for the ablation of glassy materials," *Journal of the aerospace sciences*, Vol. 26, 1959.

[5] Merrifield, J., Beck, J., Markelov, G., Leyland, P., and Molina, R., "Aerothermal Heating Methodology in the Spacecraft Aerothermal Model (SAM)," *Space Safety is No Accident*, edited by T. Sgobba and I. Rongier, The 7th IAASS Conference, Springer, Cham, Switzerland, 2005, pp. 463–468.

[6] Fujimoto, K., Tani, H., Negishi, H., Saito, Y., Iizuka, N., Okita, K., and Kato, A., "Uncertainty Quantification for Destructive Re-Entry Risk Analysis: JAXA Perspective," *Stardust Final Conference*, edited by M. Vasile, E. Minisci, L. Summerer, and P. McGinty, Astrophysics and Space Science Proceedings 52, Springer, Cham, Switzerland, 2018, pp. 283–300.

[7] Dias, B., Turchi, A., and Magin, T., "Stagnation-Line Simulations of Meteor Ablation," *45th AIAA Thermophysics Conference*, 2015.

[8] Chen, Y.-K., "Thermal ablation modeling for silicate materials," *54th AIAA Aerospace Sciences Meeting*, 2016, p. 1514.

[9] Peluchon, S., "Approximation numérique et modélisation de l'ablation liquide," Ph.D. thesis, Université de Bordeaux, 2017.

[10] 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.

[11] 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 (Sweden), 2017.

[12] Bariselli, F., Boccelli, S., Magin, T., Frezzotti, A., and Hubin, A., "Aerothermodynamic modelling of meteor entry flows in therarefied regime," *2018 Joint Thermophysics and Heat Transfer Conference*, 2018.

[13] 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.

[14] 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.

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

[16] Latige, M., "Simulation numérique de l'ablation liquide," Ph.D. thesis, Université de Bordeaux, 2013.

[17] Wang, Z., "High-order methods for the Euler and Navier–Stokes equations on unstructured grids," *Progress in Aerospace Sciences*, Vol. 43, 2007, p. 1–41.

[18] Hesthaven, J., and Warburton, T., *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*, Springer, 2008.

[19] Arnold, D., Brezzi, F., Cockburn, B., and Marini, L., "Unified analysis of discontinuous Galerkin methods for elliptic problems," *SIAM Journal Numerical Analysis*, Vol. 39, No. 5, 2001, p. 1749–1779.

[20] 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.

[21] Shima, E., and Kitamura, K., "On New Simple Low-Dissipation Scheme of AUSM-Family for All Speeds," *47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition*, 2009.

[22] Hu, X., Adams, N., and Iaccarino, G., "On the HLLC Riemann solver for interface interaction in compressible multi-fluid flow," *Journal of Computational Physics*, Vol. 228, 2009, p. 6572–6589.

[23] Yang, X., Cheng, J., Wang, C., Lua, H., and Si, J., "A Fast, Implicit Discontinuous Galerkin Method Based on Analytical Jacobians for the Compressible Navier-Stokes Equations," *54th AIAA Aerospace Sciences Meeting*, 2016.

[24] Rinaldi, E., Pecnik, R., and Colonna, P., "Exact Jacobians for implicit Navier–Stokes simulations of equilibrium real gas flows," *Journal of Computational Physics*, Vol. 270, 2014, pp. 459–477.

[25] Kummer, F., and Oberlack, M., "An extension of the discontinuous Galerkin method for the singular Poisson equation," *SIAM Journal on Scientific Computing*, Vol. 35, No. 2, 2013, pp. A603–A622.

[26] 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.

[27] 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, No. 1, 2017, pp. 3–30.

[28] 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.

[29] Müller, B., Kummer, F., and Oberlack, M., "Highly accurate surface and volume integration on implicit domains by means of moment-fitting," *International Journal for Numerical Methods in Engineering*, Vol. 96, No. 8, 2013, pp. 1–19.

[30] Saurel, R., Larini, M., and Loraud, J.-C., "Exact and Approximate Riemann Solvers for Real Gases," *Journal of Computational Physics*, Vol. 112, No. 1, 1994, pp. 126–137.

[31] Schooyen, P., "Numerical Simulation of Aerothermal Flows Through Ablative Thermal Protection Systems," Ph.D. thesis, Université catholique de Louvain, 2015.

[32] Scoggins, J., and Magin, T., "Development of Mutation++ : MUlticomponent Thermodynamics And Transport properties for IONized gases library in C++," *11th AIAA/ASME Joint Thermophysics and Heat Transfer Conference*, 2014.

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

[34] Quartapelle, L., Castelletti, L., Guardone, A., and Quaranta, G., "Solution of the Riemann problem of classical gasdynamics," *Journal of Computational Physics*, Vol. 190, 2003, pp. 118–140.

[35] Colonna, P., and Rebay, S., "Numerical simulation of dense gas flows on unstructured grids with an implicit high resolution upwind Euler solver," *International Journal for Numerical Methods in Fluids*, Vol. 46, 2004, p. 735–765.