3D mesh displacement strategy to simulate the thermal degradation of materials under atmospheric reentry conditions

Nicolas Perron, Nicolas Dellinger, Ysolde Prévereaud, M. Balat-Pichelin

3D Mesh Displacement Strategy to Simulate the Thermal Degradation of Materials under Atmospheric Reentry Conditions

Nicolas Perron[a], Nicolas Dellinger[a,*], Ysolde Prévereaud[a], Marianne Balat-Pichelin[b]

[a] *ONERA / DMPE, Université de Toulouse, F-31055 Toulouse - France*
[b] *PROMES-CNRS, Font-Romeu Odeillo, France*

Abstract

A three-dimensional strategy to compute mesh displacement following surface recession due to ablation is proposed and implemented in the finite volume material response code MoDeTheC. Due to the application to the thermal degradation of space debris during atmospheric reentry, the strategy developed is based onto a very general formulation that can deal with any mesh topology and object shape without preliminary identification of ablated surfaces. First, a new moving mesh method modifies the grid to take into account changes due to ablation. Subsequently, a shape preservation mesh balancing method redistributes the mesh vertices to maintain the grid quality. Finally, a smoothing algorithm is applied to prevent high frequency mesh oscillations. The new 3D mesh displacement strategy is verified on many 2D and 3D test cases to prove the capabilities of the method.

Keywords: ablation, moving mesh method, smoothing, space debris, atmospheric reentry

1. Introduction

Since the beginning of space exploration, the human activity in space has generated a significant number of space debris. During the last forty years, around 16 000 tons of space debris have experienced a terrestrial atmospheric reentry. Between 10 and 40 % of that mass may have reached the Earth surface, representing a potential threat to ground safety. The estimate of the total casualty area became a major issue for all space actors and specifically for CNES (*Centre National d'Études Spatiales* - French Space Agency) which is in charge of ensuring the right application of the French Space Operation Act by 2021 for French satellites-and-launchers operators and launch operations from French Guyana spaceport. According to the French law guidelines, the maximum casualty risk probability is set to 10^{-4}.

*Corresponding author
Email addresses: nicolas.perron@onera.fr (Nicolas Perron), nicolas.dellinger@onera.fr (Nicolas Dellinger), ysolde.prevereaud@onera.fr (Ysolde Prévereaud), marianne.balat@promes.cnrs.fr (Marianne Balat-Pichelin)

March 18, 2022

Although the atmospheric reentry of spacecrafts is studied since many decades, new challenges arise from space debris research. The main difference is linked to the nature of this kind of reentry: the trajectory is unknown and uncontrolled, the shapes are various and, unlike thermal protection systems, the different kind of materials encountered are not designed to withstand such conditions. For these reasons, materials might experience significant degradation.

The evaluation of ground risk is achieved using codes divided into two groups: Object-Oriented Codes and Spacecraft-Oriented Codes. The first group of codes separately computes each part of the debris (satellite for example) with important geometric simplifications, whereas the second one computes the whole debris. Most of the spatial agencies use an Object-Oriented Code : DAS (NASA) [1], ORSAT-J (JAXA) [2], DEBRISK (CNES) [3], DRAMA (ESA) [4] and/or Spacecraft-Oriented Code SCARAB (HTG - ESA) [5], PAMPERO (CNES) [6], ARES (ONERA) [7]. Even if these codes allow to evaluate the ground risk for a low computational cost, the simplified models used do not account for the in-depth material phenomena.

An accurate simulation of the material degradation is achieved with a material response code, which can then be coupled with aerothermal and flight mechanics simulations. A large number of ablative-material response code are developed and a summary of the models used is provided by Lachaud *et al.* [8]. Some of these codes use the Finite Volume Method (FVM) as PATO [9] but the majority uses the Finite Element Method (FEM) as SAMCEF Amaryllis [10] since structural solvers were historically developed with the FEM for material deformation purposes.

At ONERA, the atmospheric reentry prediction is computed with the in-house code ARES (Atmospheric Re-Entry Software) [7] which allows to couple four independent codes: AtMoS, FAST, MUSIC and MoDeTheC. AtMoS (Atmosphere Model Software) provides atmosphere data (pressure, temperature, density, composition) depending on the planet (Earth and Mars), the geographic position (altitude, longitude, latitude), the date (day, month, year) and specific phenomena (as luni-solar winds, magnetic disturbances). The aerothermodynamic modelling tool FAST (Fast Aerothermodynamic Solver for Trans-atmospheric vehicle) computes the 3D wall heat flux distribution as well as the skin friction and pressure distributions allowing to calculate the aerodynamic coefficients. These coefficients are used in the multi-objects 3DOF or 6DOF trajectory software MUSIC (MUlti SImulators in Combination).

MoDeTheC, from the French *Modélisation de la Dégradation Thermique des Composites*, is a material response code initially developed to simulate thermal degradation due to fire exposure [11]. This software computes heat and mass transfer based on Fourier' and Darcy's laws throughout anisotropic, porous composite materials. Chemical reactions (pyrolysis and oxidation) and ablation can also be taken into account. The equation formulation is based on the FVM on unstructured meshes. The FVM was chosen instead of the FEM because its implementation is more straightforward in order to handle gas flow within the material. Mesh displacement due to ablation is achieved through an Arbitrary Lagrangian-Eulerian formulation, limited to 2D geometries. A coupling is implemented between FAST and MoDeTheC with the resolution of an energy balance at the fluid/structure interface

providing the wall temperature and heat fluxes. Applications of the MoDeTheC solver address, for example, decomposition of carbon-reinforced composite materials in aeronautical fire scenarios and of composite materials, metals and alloys in atmospheric re-entry scenarios.

To decrease the error for ground risk estimation, material degradation computations have to be improved. Therefore, as previously mentioned, the 3D thermochemical behaviour of the material has to be fully computed, in particular the 3D mesh displacement due to ablation. This paper presents the general 3D mesh displacement strategy developed to simulate space debris thermal degradation during atmospheric reentry.

Firstly, the material response code MoDeTheC is briefly presented, as well as the numerical strategy of the solver. Then, the interface tracking and the mesh redistribution methods from the literature and potentially compatible with space debris application are introduced. The analysis of these methods show that some are not compatible with MoDeTheC and others are not directly applicable for the space debris requirements. Parts of these numerical methods are combined to define a new strategy for the mesh displacement and smoothing. The new strategy, divided into three main steps, is described with: 1- a mesh displacement method ; 2- a shape preservation mesh balancing method ; 3- a mesh smoothing method. Then, a verification campaign on theoretical test cases is exhibited to prove the accuracy of the mesh displacement strategy.

2. MoDeTheC material response code

MoDeTheC is a material response solver developed in first intention to compute the thermo-chemical features of a composite material exposed to fire. A composite material like CFRP (Carbon Fibers Reinforced Polymer) is a porous material initialy composed of fibers, matrix and gas. When the material temperature increases, the matrix reacts through a pyrolyse reaction and is transformed into a carbon residue (char) and a mixture of decomposition gases. With the increase in material porosity, the gas is allowed to be advected in the material and ejected at the material surface. After the pyrolysis, the material is decomposed with the presence of oxygen atoms and molecules through oxidative reactions.

The material modeling in MoDeTheC is based on a multi-species approach. A set of I species (solid or gas) including J gaseous species is chosen to define the composite material. The thermo-chemical decomposition of the material is defined with a set of M chemical reactions. Each degradation reaction is driven by an Arrhenius rate equation and is associated to an heat of reaction. The heat and mass change due to the chemical reactions are taken into account in the mass and energy conservation equations.

In the following, the subscript i refers to any species, the subscript j refers to a gaseous species and the subscript g refers to the gas mixture. When no subscript is applied, the variable refers to the material. The mass conservation equations are:

$$\begin{cases} \dfrac{\partial (\rho Y_i)}{\partial t} = \dot{\omega}_i & \text{for solid species} \quad (1) \\ \dfrac{\partial (\rho Y_j)}{\partial t} + \overrightarrow{\nabla} \cdot \left(\rho_g \dfrac{Y_j}{Y_g} \overrightarrow{v}_g \right) = \dot{\omega}_j & \text{for gaseous species} \quad (2) \end{cases}$$

where ρ is the density (in kg/m^3), Y the mass fraction, $\dot{\omega}$ the mass source term due to chemistry (in kg/(m$^3 \cdot$ s)) and $\overrightarrow{v}_g$ the average gas velocity (in m/s).

The energy conservation equation is:

$$\frac{\partial (\rho h)}{\partial t} + \overrightarrow{\nabla} \cdot (\rho_g h_g \overrightarrow{v}_g) = \overrightarrow{\nabla} \cdot \left(\overline{\overline{k}}^{\star} \overrightarrow{\nabla} T \right) + \dot{Q}_R \tag{3}$$

where h is the enthalpy (in J/kg), $\overline{\overline{k}}^{\star}$ the effective thermal conductivity tensor (in W/(m $\cdot$ K)) and $\dot{Q}_R$ the total heat release rate of the chemical reactions (in W/m^3).

In these equations, the average gas velocity $\overrightarrow{v}_g$ is calculated using Darcy's law:

$$\overrightarrow{v}_g = -\frac{\overline{\overline{K_P}}}{\mu_g} \overrightarrow{\nabla} p_g \tag{4}$$

where $\overline{\overline{K_P}}$ is the permeability second order tensor (in m^2), μ_g the dynamic viscosity of the gas mixture (in Pa $\cdot$ s) and p_g the average gas pressure computed under the assumption of the ideal gas mixture law (in Pa).

The governing equations described above are implemented in MoDeTheC using the finite-volume formulation in 2D and 3D for unstructured fixed meshes. The resolution of the system is done using a first order operator splitting. The thermal diffusive and reactive numerical subsystem is first computed with a time step Δt. Then, the advective subsystem is solved with a sub-iteration procedure based on the maximum value of the CFL (Courant Friedrichs Lewy) number.

In MoDeTheC, the material ablation is modelled at the boundaries using several models: imposed ablation velocity, energy balance based on latent heat of sublimation or chemical reactions (oxidation, nitridation for example). The ablation model returns a mass and an energy increment which are integrated in the mass and energy conservation equations. These data are integrated in the numerical model using the Arbitrary Lagrangian-Eulerian approach. The velocity of the ablated is boundary is finally computed as follows:

$$\overrightarrow{V}_{\text{ab},f} = -\frac{\dot{m}_{\text{ab},f}}{\rho A_f} \overrightarrow{n}_f \tag{5}$$

where $\dot{m}_{\text{ab},f}$ is the ablated mass of the boundary cell linked to f (in kg/s), A_f the face area (in m^2) and $\overrightarrow{n}_f$ the outward-pointing normal vector to f. However, the mesh deformation requires the displacement velocity for the mesh vertices. The face/vertex conversion is not trivial and is the purpose of section 4.3.

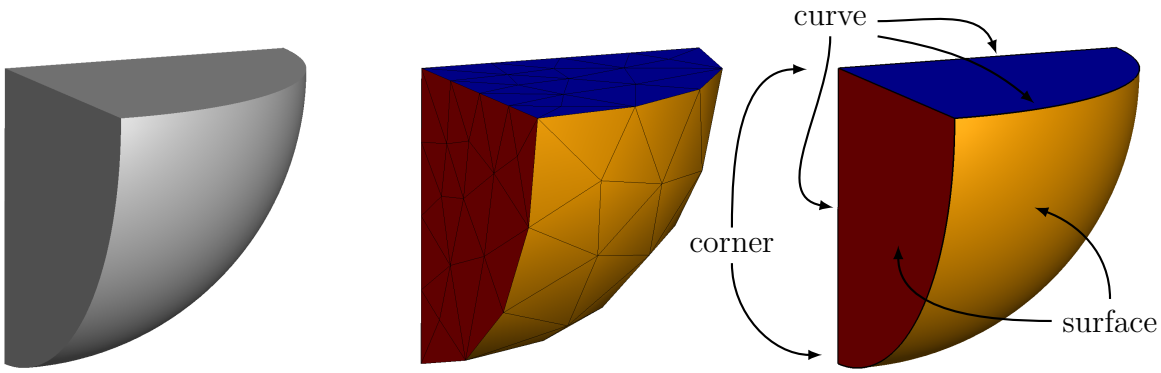
The boundary mesh deformation is followed by an adaption of the entire mesh which results in a displacement velocity $\overrightarrow{V}_{\text{mesh}}$ for all faces. The mesh displacement can be interpreted as the displacement of the conservative variables (ρY, ρh) in the opposite direction, *i.e.* with the velocity $-\overrightarrow{V}_{\text{mesh}}$. The advective motion of the mesh $\overrightarrow{\nabla} \cdot \left(u \overrightarrow{V}_{\text{mesh}} \right)$ is added in the heat and mass equations of the first operator Splitting.

3. Numerical methods for mesh displacement

The description of the MoDeTheC solver shows the numerical choices made to model ablation. The conservative equations are processed with the Arbitrary Lagrangian-Eulerian approach and the boundary is explicitly displaced. Thus, the overall computational domain needs to be adapted to follow the boundary deformation. The numerical methods from the literature for the interface tracking and the mesh adaptation are respectively described in sections 3.2 and 3.3 after the definition of the technical vocabulary in section 3.1.

3.1. Introduction of the technical vocabulary used

The technical vocabulary used to describe the mesh is defined in this section to avoid misinterpretation and is illustrated in Fig. 1, 2 and 3. These illustrations use a geometry representing an eighth of a sphere (Fig. 1a) modelled by a coarse mesh (Figs. 1b). The basic notions of a mesh (**vertex**, **edge** and **face**) are shown in Fig. 2.



(a) Geometry. (b) Mesh. (c) Surface, curve and corner.

Figure 1: Mesh vocabulary on an eight of a sphere geometry.

The comparison between the mesh (Fig. 1b) of the geometry and the geometry itself (Fig. 1a) shows that the mesh induces a loss of information about the geometry. Indeed, a gap between the boundary edges and the arcs exists (Fig. 2 and 3). Also, the surface faces of the mesh are not overlaid with the surface of the sphere. This difference between the mesh and the geometry occurs for every curved geometry because first order elements are used in the MoDeTheC solver. The use of higher-order meshes would be a way to alleviate this issue but beyond the scope of the present article.

To generalise, the term **arc** is used to name the space curve associated to an edge of the mesh, and a **surface patch** is used to name the geometrical surface associated to a face of a mesh. In case of planar surface, arcs and edges overlay as well as faces and surface patches.

On every mesh, the sharp edges and corners can be detected (cf. Section 4.2.1) to split the surface mesh. By following this procedure, the eighth of a sphere is split into 4 boundary zones: three flat surfaces and one curved surface. For a given boundary zone, each face is modelled by a surface patch. The union of these surface patches is called a **surface** (Fig.

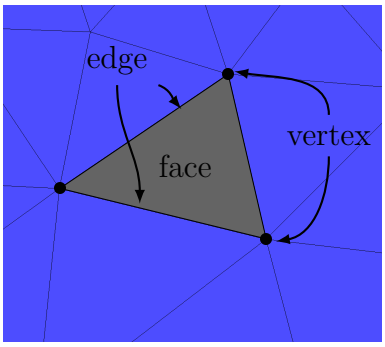


Figure 2: Mesh vocabulary : vertex, edge and face.

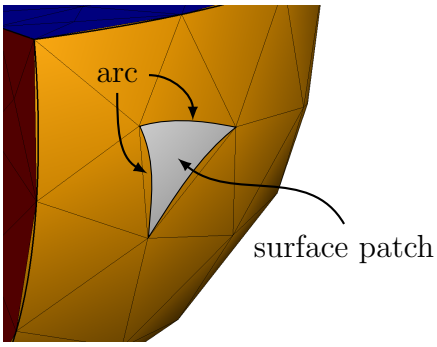


Figure 3: Geometry vocabulary : arc and surface patch.

1c). The intersection between two surfaces defines a **curve** and the intersection between two or more curves defines a **corner**.

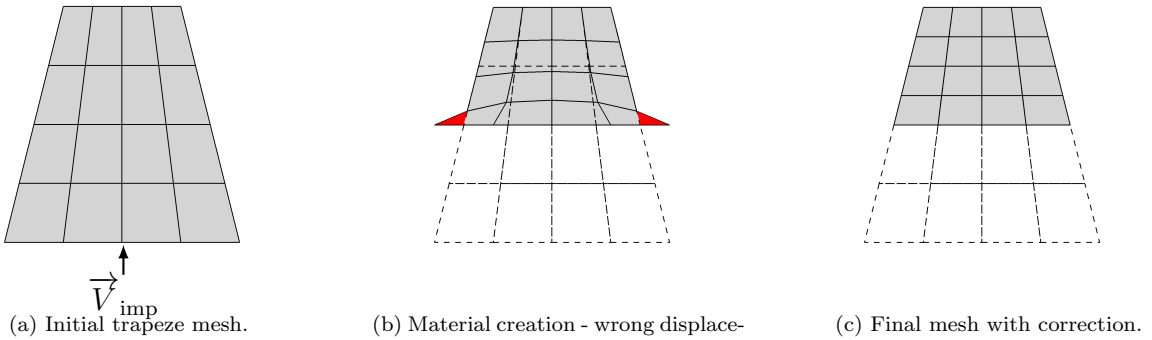
Finally, the authors wish to clarify the use of the word "smoothing" for the next sections of the paper. In many publications, mesh smoothing refers to the action of moving vertices to give a regular aspect to the mesh. In the same time, smoothing methods are also used to remove undesirable noise in the mesh. In this paper, a different term is associated with each method:

- mesh **balancing** refers to the action of redistributing the vertices to maintain the grid quality and avoid cell inversion;
- mesh **smoothing** refers to the work of avoiding high frequency oscillations (numerical noise) between boundary vertices.

3.2. Interface Tracking Methods

When an interface tracking method is used, the displacement of boundary faces, edges and vertices depends on the calculation of a displacement vector for each boundary vertex. With a structured grid, the easiest method is to move the boundary vertices along internal edges. Then, the internal vertices are proportionally distributed between the moved boundary and a fixed boundary [12, 13]. However, this method cannot be easily transposed to unstructured grids. Indeed, the internal edge along which the boundary vertex has to be displaced is not unique. Hence, a boundary displacement method is applied to the normal direction of the boundary faces, in order to be suitable for both structured and unstructured grids.

During the ablation process, moving the boundary faces in the normal direction can lead to the creation of areas where the material cannot be, *i.e.* irrelevant mass added to the system (Fig. 4). To prevent this pathological behaviour and to correctly displace structured and unstructured meshes, boundary conditions are generally defined as stationary, sliding or ablating as done by Droba [14]. Then, a treatment is applied to each vertex depending on the type of boundary surface around it.



(a) Initial trapeze mesh.

(b) Material creation - wrong displacement.

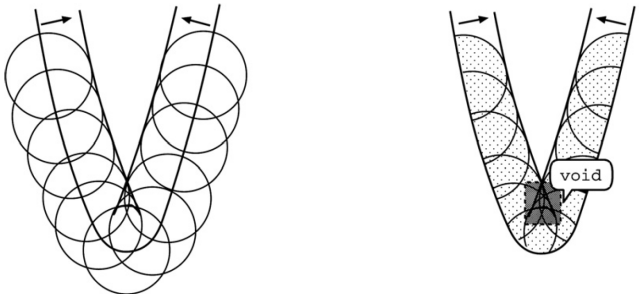
(c) Final mesh with correction.

Figure 4: Illustration of the displacement correction need.

In 2D, the displacement of a corner vertex is found using the intersection of the two displaced edges. In 3D, corner vertices have 3 different boundary surfaces around them. For each boundary surface around the 3D corner vertex, the best fit plane for the face displacement is found. Then, the new location of the 3D corner vertex is the intersection of the 3 planes. However, this strategy for the displacement of 3D corner vertices cannot be used for computations with important shape deformation. Indeed, when a corner vertex is smoothed, the intersection between the 3 planes around it is no longer defined. The strategy described by Droba [14] allows to deform complex geometries in 2D. However, the 3D strategy was not verified with test cases and the formulation seems inadequate for shape changes. Moreover, for space debris atmospheric reentry applications, boundary surfaces cannot be classified at the beginning of the calculation since every part of the debris is likely to be ablated due to both tumbling and fragmentation processes. Therefore, this method cannot be used without modifications.

Another method for the displacement of an explicit surface was developed by Jiao [15] and mainly used for icing and solid rocket motor propellant grain combustion computations. The displacement direction is firstly driven by the normal vector of each face. The displacement vector is corrected with a wave frontal motion to avoid issues in the singular points (Fig. 5b). Then, the optimal displacement distance is chosen to remove the exact ablated mass in the least square sense. The main advantage of this method compared to Droba [14] method is that singularities of the geometry are managed without a preliminary classification. However, the wavefrontal motion for the surface displacement never preserves the object shape. Jiao [15] gives the example of a cube with a constant displacement velocity applied on each boundary face. The wavefrontal motion smooth the geometry whereas the advective motion (equivalent to the strategy of Droba) preserves the cube shape. Depending on the computation objective, a choice between these two approaches is required with Jiao algorithm.

The first boundary displacement methods based on internal edge direction are not suitable for the unstructured meshes to be treated by MoDeTheC. The other methods only



(a) Envelope of sphere. (b) Envelope of balls.

Figure 5: Two-dimensional illustration of difference between envelopes of spheres and balls [15].

consider the boundary mesh and are thus compatible. However, one constraint is that MoDeTheC solver might be able to ablate any part of the geometry without preliminary identification which is not the case with the method proposed by Droba. Among the boundary displacement method of the literature, only the one presented by Jiao [15] could be used for space debris application. However, the procedure described to correct the motion is not able to maintain the shape of the object studied as illustrated in Fig. 4. This analysis motivates the development of a new strategy to deal with 3D surface deformation.

3.3. Mesh redistribution

The entire mesh needs to be redistributed to follow the boundary displacements with the ALE formulation, generally in a two-step procedure to conserve the object shape. First, the boundary vertices are redistributed and then the internal vertices. The redistribution techniques can be based on the resolution of Partial Differential Equations (PDE), on mechanical analogies (mass-spring system, linear elastic solid) or on topological changes (local or global remeshing). Several comparisons between these techniques can be found in [16, 17].

The Laplacian smoothing is a simple and efficient method. It was largely studied and many derivations were suggested. The main drawback of the Laplacian smoothing is the flattening of the geometry which reduces the geometry volume and creates zero-volume cells. For the boundary vertex redistribution, a first solution is to limit the vertices displacement to the neighboring faces [18]. Other solutions were developed to solve this problem based on Laplacian derivations (bilaplacian, Taubin smoothing, mean curvature flow) [19, 20]. Even if derived methods exhibit better solutions, surface flattening is still observed [20].

Another method to redistribute the boundary vertices is to balance them in the tangential plane of the surface (null space smoothing) [15]. After some iterations, the shape reaches a steady state and the smoothing does not modify the shape anymore. The main advantage of the null space smoothing method is that the shape is not smoothed by the process. The method only redistribute the vertices over it. However, because the shape is not smoothed, unphysical noise appears at the surface [21]. To prevent the high frequency oscillations of the mesh, a smoothing step to repair the boundary mesh is required.

4. Description of the Mesh Displacement Strategy developed

4.1. Strategy overview

A new strategy is suggested for managing 3D mesh displacements to describe the thermal degradation of materials along space debris atmospheric reentry. The main objective is to maintain the mesh quality during the ablation process. The strategy was built with the in-house material response code MoDeTheC constraints and for the space debris application. So, the main constraints are:

- to define a unique strategy for any kind of mesh type (structured, unstructured and hybrid),
- to be relevant to any kind of 3D debris shape,
- to automatically detect the moving surface without preliminary identification, since in case of space debris atmospheric reentry the surface recession can occur anywhere due to space debris tumbling.

The computation of ablation in a material response code based on an ALE formulation requires the implementation of adequate mesh displacement, mesh balancing and mesh smoothing methods. In Sections 3.2 and 3.3, some methods for mesh displacement, balancing and smoothing were presented and discussed.

The proposed mesh displacement strategy (Fig. 6) is performed in three steps:

1. The **Moving Surface Mesh Method** which modifies the surface grid to take into account changes due to ablation.
2. The **Shape-Preserving Mesh Balancing Method** which maintains the mesh quality.
3. The **Mesh Smoothing Method** which avoids undesirable mesh oscillations.

The main differences between the developed method and methods proposed in the literature are the formulation of the displacement based on geometrical considerations and the automatic sharp feature detection which enables space debris thermal degradation computations. For the mesh balancing method, the null space smoothing [15] shows excellent shape conservation despite the creation of undesirable noise. This method is thus used in combination with a Laplacian operator to control the vertex distribution. The undesirable noise is removed in the third step with a least square curve/surface smoothing. The three steps of this strategy are independent and can be used separately.

The inputs for this strategy are:

- the mesh with all connectivity between vertices, faces and edges;
- the displacement length of each boundary face.

4.2. Mesh processing

The mesh displacement strategy described in this article uses specific mesh data in addition to the vertex location and connectivities. The way to obtain these required global data, used several times in the mesh displacement strategy, is presented in this section.

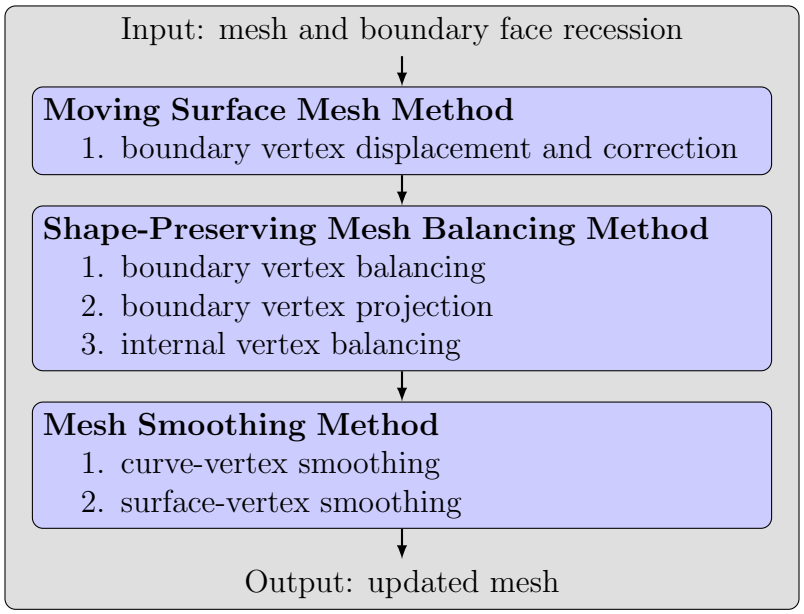
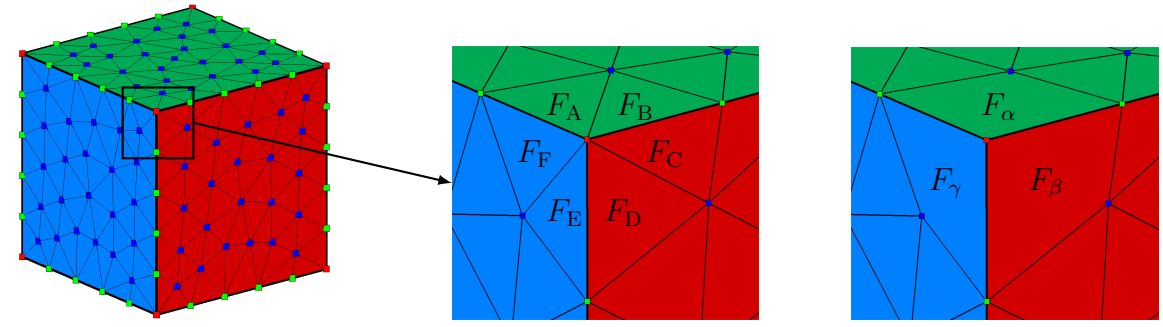


Figure 6: Mesh displacement strategy.

4.2.1. Sharp edges and corners detection

The sharp feature detection is done at the beginning of the calculation and after each mesh displacement. It is defined as a function of the angle between every couple of adjacent faces for each vertex. If this angle is greater than a user-defined limit angle θ_{lim}, the edge is considered as sharp. The default value in MoDeTheC is set to $\theta_{\text{lim}} = 45°$. The vertices are finally classified (Fig. 7a) between those having zero sharp edge ■, 2 sharp edges ■ and 3 ■. Object with vertices having 4 or more sharp edges are not presented in this article because they are not encountered in the application cases of interest. Nevertheless, the strategy described can handle such geometries with simple adjustments in the vertices correction procedure.

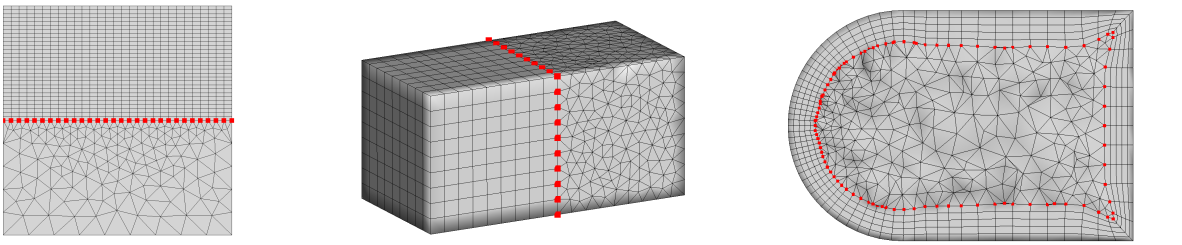


(a) Boundary vertex classification. (b) Mesh with initial faces. (c) Mesh with equivalent faces.

Figure 7: Sharp edge detection and displacement steps on cube mesh.

4.2.2. Hybrid interface detection

As already mentioned, MoDeTheC solver manages all standard elements for structured and unstructured meshes. The interface between a structured and an unstructured area is of particular interest for the mesh balancing because important displacements can occur leading to a loss in mesh quality. Thus, the interface is detected to apply a special formulation at these vertices. This interface detection is done as follows: if all the elements around a vertex are not of the same type and at least one element is of structured type, the vertex is declared to be located on the interface. The result of the interface detection is shown in Fig. 8 for 2D and 3D cases.



(a) Square (2D). (b) Rectangular cuboid (3D). (c) Internal view of a blunted cylinder (3D).

Figure 8: Illustration of the hybrid interface detection shown in red.

4.3. Moving Surface Mesh Method

The goal of the Moving Surface Mesh method is to convert the displacement length of boundary faces into vertex displacements. For this step, only the mesh on the boundaries is considered while the internal mesh remains fixed.

For the moving surface mesh method, the vertices are divided into two groups. For all vertices located on sharp edges (■ and ■ in Fig. 7), the *Average of smooth faces* and *Vertices correction procedure* steps described in sections 4.3.1 and 4.3.2 are applied. For all vertices outside the sharp edges (■ in Fig. 7), only the *Average of smooth faces* step described in Section 4.3.1 is applied.

4.3.1. Average of smooth faces

The displacement of the corner (■) vertex in Fig. 7b depends on the surface recession $\overrightarrow{V}^{i}_{\text{dep}}$ of its neighboring faces F_i and their respective normal vector $\overrightarrow{n}_i$ (for $i \in \{A, \ldots, F\}$). The first step is to consider adjacent faces in the same surface. This means that the displacement of three sets of faces ($\{F_A; F_B\}$, $\{F_C; F_D\}$ and $\{F_E; F_F\}$) is calculated (Fig. 7b).

The displacement due to faces connected with smooth edges is computed with an area-weighted average of each face recession velocity. As illustrated in Fig. 9, the displacement vector due to two smooth faces j and k is:

$$\overrightarrow{V}^{\alpha}_{\text{d}} = \frac{1}{\sum\limits_{i\in\{j,k\}} \mathscr{A}_i} \sum_{i\in\{j,k\}} \mathscr{A}_i \overrightarrow{V}^{i}_{\text{d}} \tag{6}$$

where $\mathscr{A}_i$ is the area of face F_i. $\overrightarrow{V}^{\alpha}_{\text{dep}}$ is the equivalent displacement vector of faces F_j and F_k. Eq. 6 has a general formulation which can be extended to any number of adjacent faces in a group of faces not sharing sharp edges. After this step, an equivalent mesh (Fig. 7c) is virtually obtained where the corner vertex is shared by faces (F_α, F_β and F_γ) only separated by sharp edges. For each equivalent face, the vertex displacement vector is thus known.

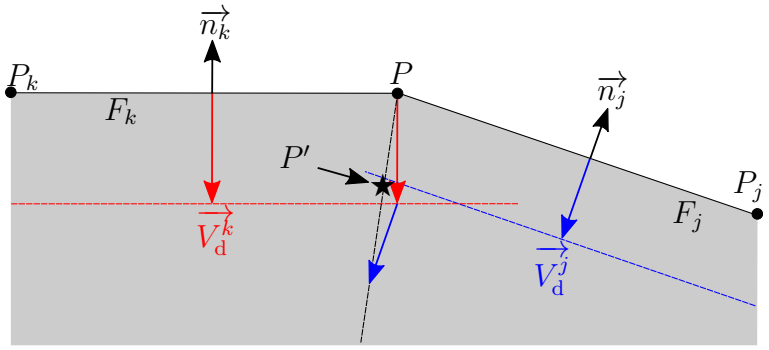


Figure 9: Displacement of smooth vertex computed with an area-weighted average of each face recession.

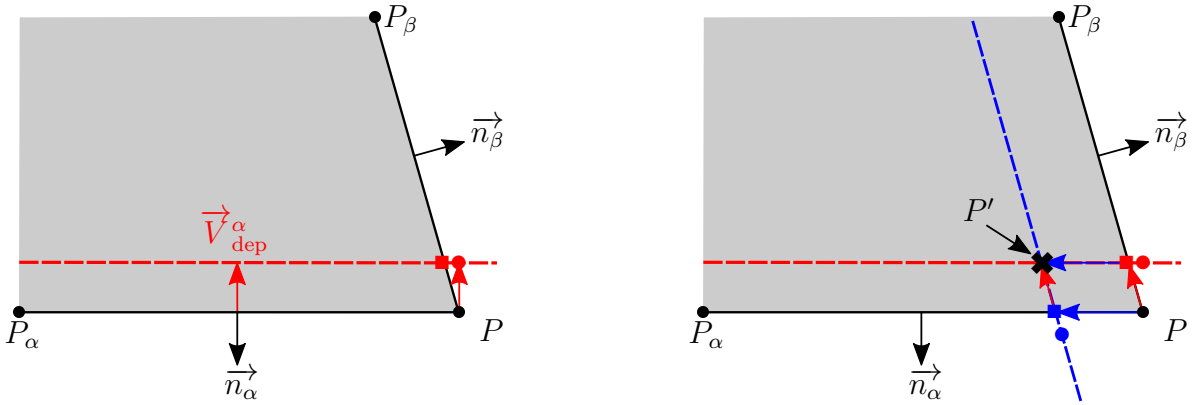
4.3.2. Vertices correction procedure

The purpose of the correction is to prevent vertices from getting out of the computational domain. A sharp vertex displacement without correction is illustrated with ● in Fig. 10a.

The correction moves the vertex to the closest boundary face or edge with the following procedure applied separately to each equivalent face. First, the complementary face or edge is found. In 2D, the complementary face is the only other face directly linked to the vertex. For corner vertices with 3 sharp edges in 3D, the complementary edge is the edge between the two other equivalent faces. For example, the complementary edge of F_α in Fig. 7c is the edge between F_β and F_γ. The intersection between the complementary face or edge and the normal plane to $\overrightarrow{V}^{\alpha}_{\text{dep}}$ gives a point P^{α}_{cor} (■ in Fig. 10a). The final position of the displaced vertex P' is obtained with the sum of the corrected displacement of each equivalent face (Fig. 10b).

For corner vertices with more than 3 sharp edges, a way to correct the displacement would be first to find the closest boundary face to the displaced vertex ●. The intersection between this face and the normal plane to $\overrightarrow{V}^{\alpha}_{\text{dep}}$ gives a line L^α. The corrected displacement ■ in Fig. 10a is found with the normal projection of ● on the line L^α.

After these steps, the grid is changed to take into account the mass removal and the updated material shape. The next step of the mesh displacement strategy is to balance and smooth the mesh while preserving the shape to avoid cell inversions and keep a good mesh refinement.



(a) Sharp vertex correction.

(b) Sharp vertex displacement.

Figure 10: Vertex correction procedure.

4.4. Shape-Preserving Mesh Balancing Method

The proposed balancing method is designed to maintain the geometry shape during the vertex redistribution procedure. The method is based on the sharp edge and vertex detection procedures described above and is divided into 4 steps:

1. curve-vertex balancing (■ vertices with ■ and ■ neighbours[1] in Fig. 7a) ;
2. surface-vertex balancing (■ vertices with ■, ■ and ■ neighbours in Fig. 7a) ;
3. boundary-vertex projection on arc or surface patch ;
4. internal vertex balancing.

Note that during this procedure, the corner vertices (■ in Fig. 7a) remain unchanged whereas all the other vertices are balanced with the moving surface mesh method.

The curve, surface and internal vertex balancing are done with a Laplacian formulation detailed below and satisfying two objectives:

– prevent any change of the initial grid refinement;
– process mesh distortion and large deformation.

4.4.1. Weight determination for the converged Laplacian operator

The Laplacian formulation uses converged weights calculated at the beginning of the computation. These weights allow not to move the relative position of vertices when it is not necessary (*i.e.* before important ablation occurs).

The calculation of these weights is done by solving N systems of N_{v}^i vertices where N is the number of vertices in the mesh and N_{v}^i is the number of vertices connected to the vertex i. Another solution is to solve one system of N vertices but this second option has an higher

[1] ■ vertices are not considered because they do not belong to the curves.

computational cost. A least square algorithm is implemented in MoDeTheC and gives the weight w'_{ij} of the vertex i linked to the vertex j ($w'_{ij} \neq w'_{ji}$).

4.4.2. Laplacian operator

The Laplacian operator $\overrightarrow{\mathscr{L}}$ modifies the position of each vertex at each time-step with a sub-iteration process up to $\overrightarrow{\mathscr{L}} = 0$:

$$\overrightarrow{P}_i^{n+1} = \overrightarrow{P}_i^{n} + \gamma \overrightarrow{\mathscr{L}}\left(P_i^n\right) \tag{7}$$

where $\gamma = 0.4$ is a damping factor and n the sub-iteration number. In this method, the Laplacian is divided into 2 sub-models $\overrightarrow{\mathscr{L}_1}$ and $\overrightarrow{\mathscr{L}_2}$ with $\overrightarrow{\mathscr{L}} = (1-\alpha)\overrightarrow{\mathscr{L}_1} + \alpha\overrightarrow{\mathscr{L}_2}$ where $\alpha \in [0,1]$ is a parameter to pass from a Laplacian formulation to an other and is defined as:

$$\alpha = \min\left(1\,,\ \frac{\left\|\overrightarrow{P_i P_{i_0}}\right\|_2}{L_{\text{lap}}}\right) \tag{8}$$

where P_{i_0} is the initial location of P_i at the beginning of the computation (*i.e.* $P_{i_0} \neq P_i^0$). The characteristic length for the Laplacian smoothing L_{lap} is defined by the user depending on the length and shape of the object. For the *2D Sinusoidal Slider* example shown in Fig. 20, a good choice is the half of an oscillation length *i.e.* $L_{\text{lap}} = 0.5\,\text{m}$.

The first Laplacian $\overrightarrow{\mathscr{L}_1}$ is the dominant term when the mesh is close to the initial mesh ($\alpha \to 0$):

$$\overrightarrow{\mathscr{L}_1}\left(P_i^n\right) = \frac{1}{\sum\limits_{j\in\mathscr{J}_i} w'_{ij}} \sum_{j\in\mathscr{J}_i} w'_{ij}\,\overrightarrow{P_i^n P_j^n} \tag{9}$$

where $\mathscr{J}_i$ is the set of vertices directly connected to P_i and w'_{ij} is the weight of the edge between P_{i_0} and P_{j_0}, determined with the procedure described in Section 4.4.1.

The second Laplacian $\overrightarrow{\mathscr{L}_2}$ homogenizes the mesh and prevents cell inversion when the mesh is no longer close to the initial one ($\alpha \to 1$):

$$\overrightarrow{\mathscr{L}_2}\left(P_i^n\right) = \frac{1}{\sum\limits_{j\in\mathscr{J}_i} w''_j} \sum_{j\in\mathscr{J}_i} w''_j\,\overrightarrow{P_i^n P_j^n} \tag{10}$$

where the vertex weight $w''_j = \left\|\overrightarrow{P_i^n P_j^n}\right\|$ is updated at each iteration for surface and internal balancing and at each sub-iteration for curve balancing.

Vertices on hybrid interfaces, detected thanks to the procedure described in Section 4.2.2, are always balanced with the Laplacian operator $\overrightarrow{\mathscr{L}_1}$. This choice allows to avoid important distortion of the mesh due to the difference in vertex density between structured and unstructured zones (as illustrated in Appendix A).

The main issue with the Laplacian balancing is that it tends to flatten the surface as previously mentioned. Even if this problem is solved with the following projection step on

the null space, the Laplacian balancing is modified to keep the mesh consistent during the sub-iteration process. Thus, the vertex displacement is constrained to the two adjacent sharp edges for the boundary curve balancing. Similarly, the vertex displacement is constrained to the adjacent faces for the boundary surface balancing.

4.4.3. Specific cell layer processing

As an option for the mesh balancing method, the user is free to activate a different algorithm for the prism, hexahedral and quadrilateral layers balancing. To do so, the user set the boundary surfaces from which layers are sought (*i.e.* active surface for layer research), for example Γ_1 and Γ_2 in Fig. 11. All vertices in these surfaces (■ in Fig. 11) are used as departure vertices for the algorithm. From the identified vertices, an automatic detection of prism and hexahedral (in 3D) as well as quadrilateral (in 2D) layers near the boundary is performed at the beginning of the computation as follows:

- if the vertex belongs to one surface (P_B) or to several surfaces (P_A) for which layer research is activated, the linked vertices are sought inside the domain;
- if the vertex belongs to several surfaces and at least one of these surfaces is not active to layer research (P_C), the linked vertices are sought along the boundary surfaces.

The algorithm looks for linked vertices up to an interface vertex (■) or up to another boundary. Let's remember that interface vertices are either detected with the procedure described in section 4.2.2 or user-defined.

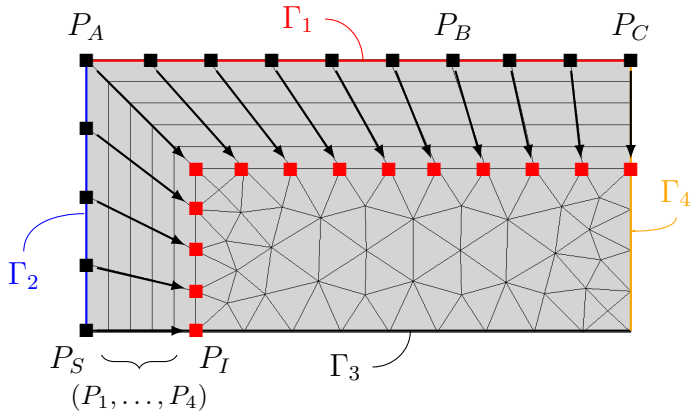


Figure 11: Example of automatic detection of cell layers between a boundary and an interface.

Prism, hexahedral and quadrilateral layers detected with this process are balanced with the specific cell layer method. To illustrate the method, five linked vertices represented in Fig. 12 are considered, where P_S is a boundary vertex and P_1, P_2 and P_3 are internal vertices. P_I can be located on another boundary surface or on an interface. The boundary vertex displacement due to ablation and balancing is written $\delta\overrightarrow{P}_S$.

The displacement of the layer vertices depends on the nature of P_I:

- when P_I is an interface vertex (an internal vertex located at the boundary between two kinds of mesh), the layer vertex displacements are equal to the displacement of

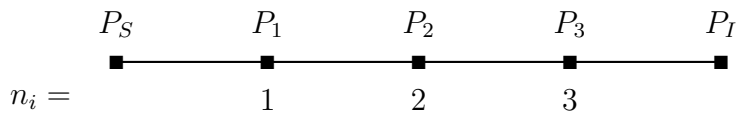


Figure 12: Cell layer balancing between a boundary vertex P_S and an interface or boundary vertex P_I.

the boundary vertex:

$$\delta\overrightarrow{P}_1 = \delta\overrightarrow{P}_2 = \delta\overrightarrow{P}_3 = \delta\overrightarrow{P}_I = \delta\overrightarrow{P}_S \tag{11}$$

- when P_I is on another boundary surface, P_I can also have a displacement $\delta\overrightarrow{P}_I$ with $\delta\overrightarrow{P}_I \neq \delta\overrightarrow{P}_S$. In this case, the relative location n_i of the layer vertices P_i between P_S and P_I is used to compute the displacement of layer vertices:

$$\delta\overrightarrow{P}_i = \frac{n_m + 1 - n_i}{n_m + 1}\,\delta\overrightarrow{P_S} + \frac{n_i}{n_m + 1}\,\delta\overrightarrow{P}_I \tag{12}$$

where n_m is the number of inner vertices of the layer ($n_m = 3$ in Fig. 12).

The cell layer displacement, when P_I is an interface vertex, can lead to a tangling of the mesh after important boundary vertex displacement. This issue occurs in the ablation simulation when the material is close to the demise. To manage this, the presented mesh displacement strategy will be associated to a remeshing algorithm in the future.

4.4.4. Vertex projection

The shape of the computed object is altered drastically by the application of the Laplacian operator during the computation. The balancing of the vertices belonging to curves and surfaces are thus followed by a projection of the boundary vertices onto the reconstructed mesh skin to prevent flattening process. This mesh skin is based on the vertex location after the moving surface mesh method and before the Laplacian operator. This step usually requires an arc equation for each boundary edge and a surface patch equation for each boundary face. In this strategy, these equations are reduced to the tangential space at the vertex. This tangential space is define as the orthogonal plane to the normal vector of the vertex. The normal vector $\overrightarrow{n}$ at a boundary vertex is defined as the average of the normal vector of the edges (2D) or face (3D) around the vertex. The arc and the surface patch are thus approximated with respectively a straight line and a plane.

Boundary vertices are projected onto the curve and surface tangential approximations according to the number of sharp edge linked to them:

- vertices with at least 3 sharp edges (■ in Fig. 7a) are not redistributed: they are not affected by this step;
- vertices with 2 sharp edges (■ in Fig. 7a) are redistributed along these 2 edges: they are projected up to the tangential line;
- vertices with 0 sharp edge (■ in Fig. 7a) are redistributed on their neighboring faces: they are projected up to the tangential plane.

Note that during this study, a second order reconstruction of arc and surface patch was implemented with Farouki *et al.* [22] methodology to take into account the curvature of a curved geometry. The ablation simulation carried out so far (some examples are shown in Section 5) have shown excellent shape preservation with both the tangential approximation of arcs and surface patches and the second order reconstruction. However, it was observed that the curved reconstruction does not improve significantly the results with an higher CPU cost.

4.5. Mesh Smoothing Method

At the beginning of the smoothing step, all the boundary vertices are considered because the noise appears during the projection step. The set of vertices to smooth is the reduced with the procedure described in Section 4.5.1. This vertex detection is mandatory because the smoothing algorithm described below (as any smoothing method) can alter the object shape if the procedure is applied at every time iteration on every boundary vertex of the mesh.

The smoothing method implemented in MoDeTheC is based on a polynomial interpolation of curves and surfaces obtained with a weighted least square method. For every vertex to smooth, a set of coefficients for Eq. 13 is computed.

$$\begin{cases} y(x) = \sum_i a_i x^i & \text{for curves} \\ z(x,y) = \sum_i \sum_j a_{ij} x^i y^j & \text{for surfaces} \end{cases} \tag{13}$$

First, the algorithm used to detect the vertices for the smoothing step is detailed. Then, the smoothing of curves in 2D and surfaces in 3D is described. Finally, the extension of the methodology to smooth curves in 3D is presented.

4.5.1. Detection of vertices for the smoothing step

The smoothing step of the mesh displacement algorithm is applied only on vertices where an oscillation is detected. This detection is done in two steps at each time step:

1. Vertex classification
   First, the normal vector $\overrightarrow{n}$ at a boundary vertex is defined as the average of the normal vector of the edges around the vertex. Then, for each boundary vertex, the scalar products between its normal vector $\overrightarrow{n}$ and the unit vectors of the edges around the vertex are calculated (Fig. 13). A smoothing tag i_l is set to every boundary vertex:
   (a) corner vertex: $i_l = -3$
   (b) when all scalar product around the vertex are positive: $i_l = 1$ (S_2 and S_4)
   (c) when all scalar product around the vertex are negative: $i_l = -1$ (S_1, S_3 and S_5)
   (d) when all scalar product around the vertex are null: $i_l = 2$
   (e) when, around the vertex, at least one scalar product is positive and at least one is negative: $i_l = -2$

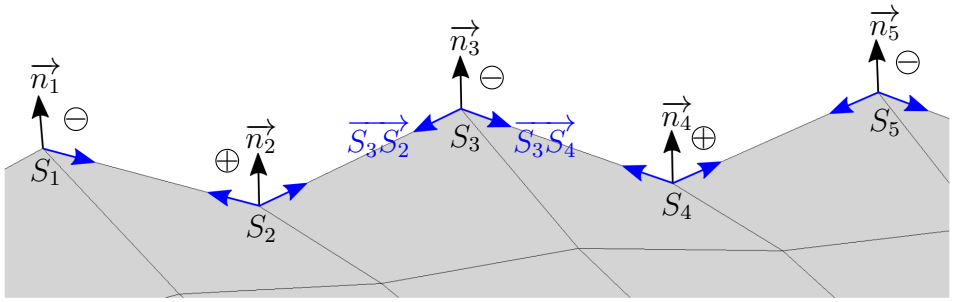


Figure 13: 2D schema of undesirable mesh oscillations where a smoothing is required.

2. Choice of vertices to smooth
   Vertices with a smoothing tag equal to -3 or 2 are not included in the set of vertex to smooth because corner vertices ($i_l = -3$) should not move during the balancing and smoothing steps and because vertices in a plane should not be smoothed. For the others, the smoothing tag of each vertex is compared to the smoothing tag of its neighbours. If at least one neighbour has a smoothing tag different from the current vertex smoothing tag, the vertex has to be smoothed.

4.5.2. 2D curve and 3D surface smoothing

The smoothing strategy illustrated in Fig. 14 is divided into 6 steps.

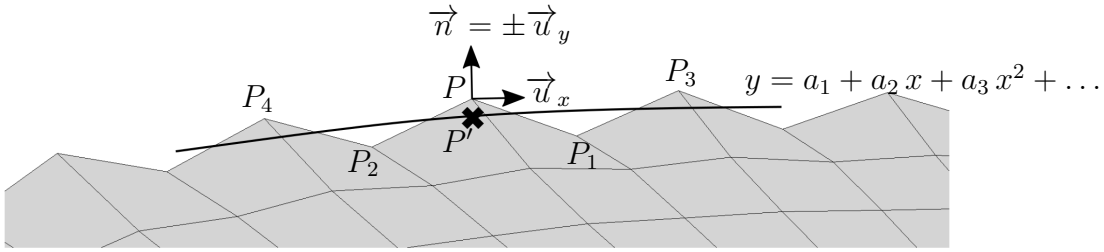


Figure 14: 2D schema of a mesh with undesirable noise where a smoothing step is required.

1. The smoothing of point P requires a set $\mathcal{N}$ of N_D vertices around P. The minimum number of neighbouring vertex N_D depends on the degree of the interpolation equation N_deg and on the dimension of the geometry.

$$\begin{cases} N_\mathrm{D} = N_\mathrm{deg} + 1 & \text{in 2D} \\ N_\mathrm{D} = N_\mathrm{deg}\,(N_\mathrm{deg} + 1)/2 & \text{in 3D} \end{cases} \tag{14}$$

2. The set of vectors $\overrightarrow{PP_i}$ is normalized by $\left(\left\|\overrightarrow{PP_1}\right\|_2 + \max_{i\in\mathcal{N}} \left\|\overrightarrow{PP_i}\right\|_2\right)$. This ensure that all distance $\left\|\overrightarrow{PP_i}\right\|_2$ are strictly lower than 1.

3. An orthonormal local basis is built so that $(P, \overrightarrow{u}_x, \overrightarrow{u}_y)$ is the plane normal to vector $\overrightarrow{n}$ through P and $\overrightarrow{u}_z$ is collinear to the normal vector $\overrightarrow{n}$ of the vertex P. In 2D, $\overrightarrow{u}_x$ is the normal vector to $\overrightarrow{n}$ and $\overrightarrow{u}_y = \pm\overrightarrow{n}$. The next steps are carried out in this basis.
4. The least square system to solve is written in matrix form with:

$$\text{In 2D:} \quad \mathrm{A} = \begin{bmatrix} 1 & 0 & 0 & \cdots \\ 1 & x_1 & x_1^2 & \cdots \\ 1 & x_2 & x_2^2 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \qquad \mathrm{B} = \begin{bmatrix} 0 \\ y_1 \\ y_2 \\ \vdots \end{bmatrix} \qquad \mathrm{X} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \end{bmatrix} \tag{15}$$

$$\text{In 3D:} \quad \mathrm{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 1 & x_1 & y_1 & x_1^2 & x_1\,y_1 & y_1^2 & \cdots \\ 1 & x_2 & y_2 & x_2^2 & x_2\,y_2 & y_2^2 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \qquad \mathrm{B} = \begin{bmatrix} 0 \\ z_1 \\ z_2 \\ \vdots \end{bmatrix} \qquad \mathrm{X} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \end{bmatrix} \tag{16}$$

The least square formulation is weighted to take into account the distance between the vertex P and its neighbours P_i. The weight function θ is defined on $[0, 1]$, strictly positive, decreasing and with $\theta(0) = 1$. In MoDeTheC, the weight function $\theta(x) = 2\,x^3 - 3\,x^2 + 1$ is used.
The vertex weight w_i is obtained using the variable x defined by the normalised distance between the vertex to smooth and the neighbour vertex. The weight matrix W is:

$$W = \begin{bmatrix} 1 & & & \\ & w_1 & & \\ & & w_2 & \\ & & & \ddots \end{bmatrix} \tag{17}$$

5. The resolution of the weighted least square problem gives the vector X for which the error is minimal [23]:

$$X = \left({}^t\!A\,W\,A\right)^{-1}\,{}^t\!A\,W\,B \tag{18}$$

6. The least square problem is here written in the local basis centered on the vertex P. The coordinate of P in this basis are thus $(0, 0, 0)$ (or $(0, 0)$ in 2D). In the local basis, the displacement of P to the smoothed vertex P' is given by the first coefficient a of the vector X. This displacement is finally formulated in the global basis of the geometry to move the vertex.

4.5.3. 3D curve smoothing

The previous formulation cannot smooth 3D space curves because a space curve is not described with one polynomial equation. Smoothing a 3D space curve uses the definition of this curve as being the intersection between two surfaces locally approximated by two planes. The strategy is then to reduce the smoothing of a space curve to the smoothing of

two 2D curves. For each surface, the space curve is projected to its approximate plane and the resulting 2D curve is smoothed.

In practice, the two first steps of the smoothing procedure for 2D curve described in Section 4.5.2 are identical in this case. Then, steps 3 to 6 are done twice: once for each normal vector of the vertex[2]. Once the smoothing of the vertex relative to each surface is done, both corrections are added to calculate the final correction of the smoothing step.

The impact of the smoothing step on the resulting geometry during an ablation computation is presented in 5.2.4.

5. Validation and demonstration test cases

The mesh displacement strategy has been validated on multiple geometries and test cases in 2D and 3D. In this section, the geometric validation is presented to verify the displacement of boundary nodes with an imposed ablation velocity. The mesh displacement on sharp edges and the shape preservation during the mesh balancing and smoothing are then validated. In addition, the relevance of the material response code MoDeTheC to deal with important shape variations is demonstrated in simulations assuming imposed heat flux at the boundaries. In this section, all meshes are generated with GMSH 3.0.2 [24] unless stated otherwise.

5.1. Validation of the mesh displacement method

The cube and truncated cone test cases are computed to validate the mesh displacement method without the influence of the projection and smoothing steps.

5.1.1. Cube

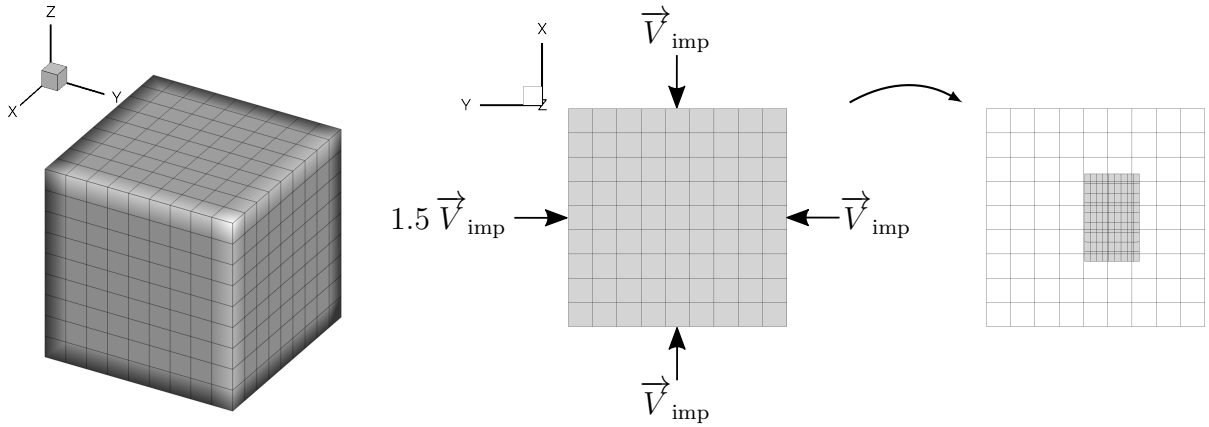
The first test case presented is a cube of 1 m side. The 3D cube mesh (Fig. 15a) is composed of 729 cells and 6 surfaces (as defined in Section 3.1).

Ablation velocities are imposed on each of the boundary surfaces of the cube during the 3 s of simulation with a time step $\delta t = 0.01\,\mathrm{s}$. Four of the planes have the same ablation velocity $\overrightarrow{V}_{\mathrm{imp}} = 0.1\,\mathrm{m/s}$, one plane has an higher ablation velocity $1.5\,\overrightarrow{V}_{\mathrm{imp}}$ and the last plane is fixed.

A 2D slice of the cube mesh at the beginning of the computation and at the end are presented in Fig. 15b and 15c. During this computation, the cube is transformed into a rectangular cuboid. The important results of this test case are:

- Right angles are kept.
- Plane surfaces are still planes after the ablation.
- The absolute displacement error δ_x regarding the plane location is $1.2 \times 10^{-8}\,\mathrm{m}$ for all ablated planes, where $\delta_x = |x_{\mathrm{th}} - x_{\mathrm{num}}|$ with x_{th} the theoretical location of the planes and x_{num} the numerical result.
- The same behaviour and results are obtained with structured and unstructured meshes.

[2]Curve vertices have two normal vectors: one for each surface.

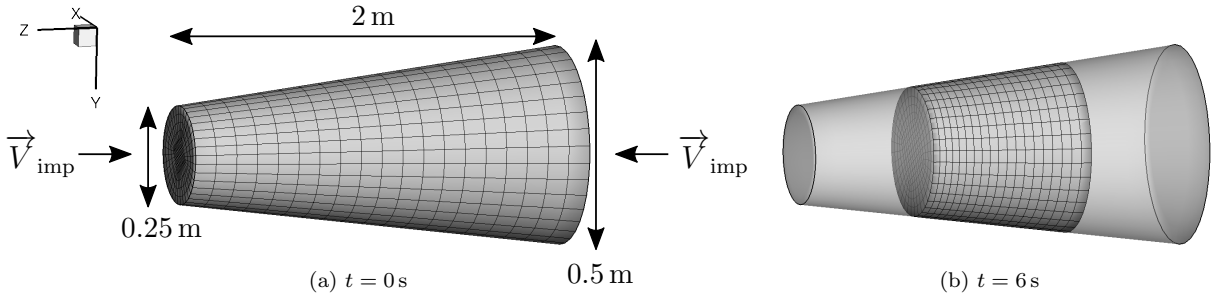


(a) 3D structured mesh. (b) Slice of the initial mesh. (c) Slice of the final mesh.

Figure 15: View of cube mesh before and after ablation.

5.1.2. Truncated cone

The node displacements on acute and obtuse angles are checked with a truncated cone geometry meshed with 6743 cells (Fig. 16a). An ablation velocity $\overrightarrow{V}_{\text{imp}} = 0.1\,\text{m/s}$ is imposed on the two plane surfaces.



(a) $t = 0\,\text{s}$ (b) $t = 6\,\text{s}$

Figure 16: View of the truncated cone geometry and mesh before and after ablation.

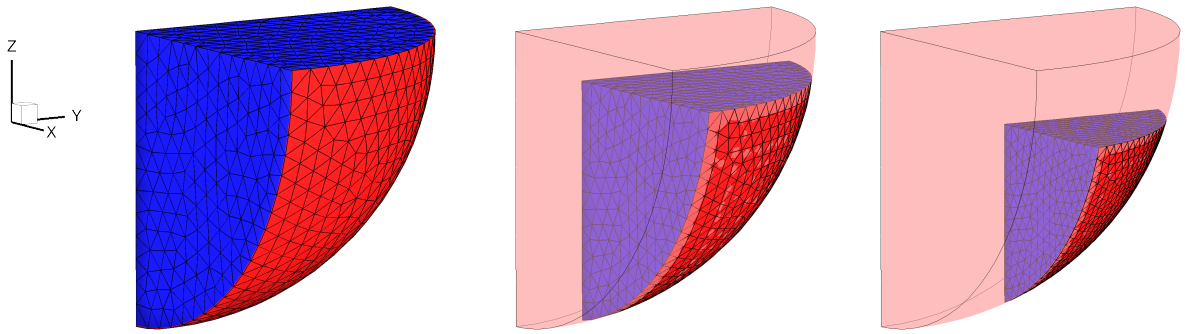
Fig. 16b compares the final and initial meshes and shapes after the 6 s simulation with a time step $\delta t = 0.05\,\text{s}$. The left and right disks are still planar and the angle between each disk and the cone boundary remains unchanged. This result proves that for acute angles, the mesh displacement does not create matter (as illustrated in Fig. 4).

5.2. Validation and demonstration of the boundary mesh balancing and smoothing methods

The balancing and smoothing method detailed respectively in Sections 4.4 and 4.5 are composed of the Laplacian operator, a projection on the tangential approximation of arcs and surface patches and a smoothing step. The next test cases show that this strategy succeeded in performing mesh displacement without flattening.

5.2.1. Eighth of a sphere

The ablation of an eighth of a sphere of 0.5 m radius is simulated with structured and unstructured meshes (Fig. 17a). This geometry is composed of 4 surfaces: 3 planes and 1 curved surface.



(a) $t = 0\,\mathrm{s}$ (b) $t = 8\,\mathrm{s}$ (c) $t = 15\,\mathrm{s}$

Figure 17: Ablation of an eighth of a sphere.

With this geometry, the first test consist in imposing during 15 s with a time step $\delta t = 0.01\,\mathrm{s}$ a constant ablation velocity $\overrightarrow{V}_{\mathrm{imp}} = 0.01\,\mathrm{m/s}$ on each planar boundary (blue part in Fig. 17). The vertices of the spherical surface (red part in Fig. 17) have to stay on the initial sphere shape during the shape preservation balancing and the smoothing procedures.

The solution at $t = 8\,\mathrm{s}$ and $t = 15\,\mathrm{s}$ (Fig. 17) show that the spherical shape of the red surface is not affected during the computation. Moreover, the distance between the vertices of the curved surface and the initial center of the sphere must remain constant and equal to 0.5 m. The minimal, maximal and average distance of the curved surface vertices $\left\|\overrightarrow{OP_i}\right\|$ are plotted as a function of time in Fig. 18. During the 15 s of computation, the error on the average distance is less than 0.1 % which allows to maintain the spherical shape of the object. The minimal and maximal distance reach respectively 0.4978 m and 0.5007 m. This proves that all vertices stay very close to the original sphere.

The second test case is performed with the same geometry, the same meshes and numerical parameters. The only difference is that the ablation velocity is not applied on the planar surfaces but on the curved surface (red in Fig. 17). The relative error δ_d between the theoretical and numerical distance $\left\|\overrightarrow{OP_i}\right\|$ is calculated and shown in Fig. 19 for the block structured mesh.

This second test case on the eighth of a sphere shows again excellent results about the mesh displacement method in MoDeTheC. The average error during the computation is less than 0.02 % and the minimal and maximal errors do not exceed 0.03 %.

5.2.2. 2D Sinusoidal Slider

At first, the *Sinusoidal Slider* test case of Droba [14] illustrated in Fig. 20 is used to validate the vertex displacement on concave and convex surfaces. A structured mesh

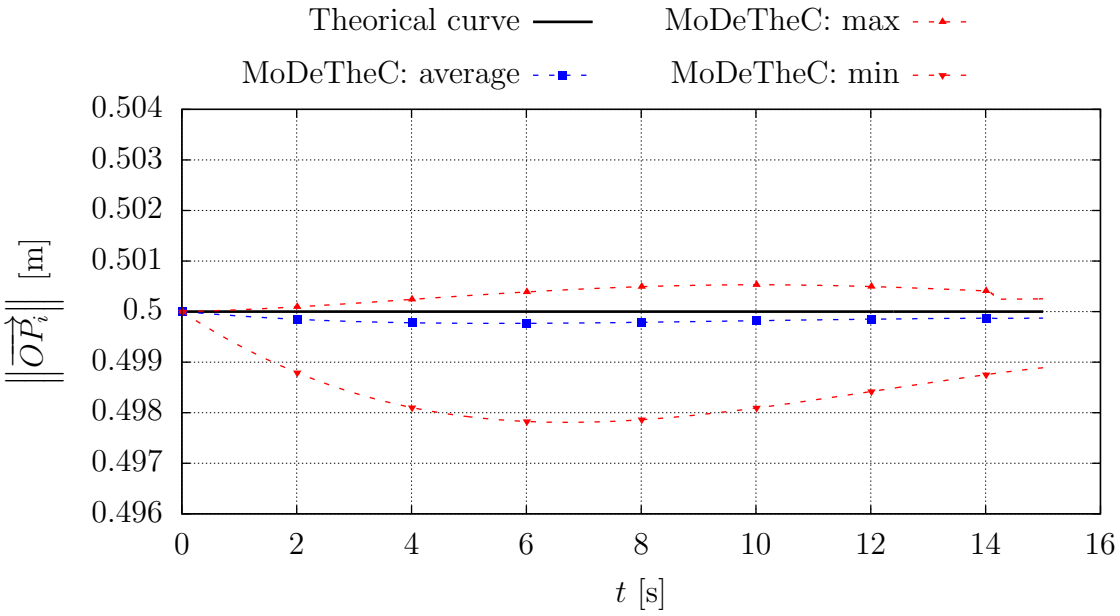


Figure 18: Minimal, maximal and average distance between the initial center of the sphere and the curved surface vertices - Imposed ablation velocity on planar boundaries - Unstructured mesh.

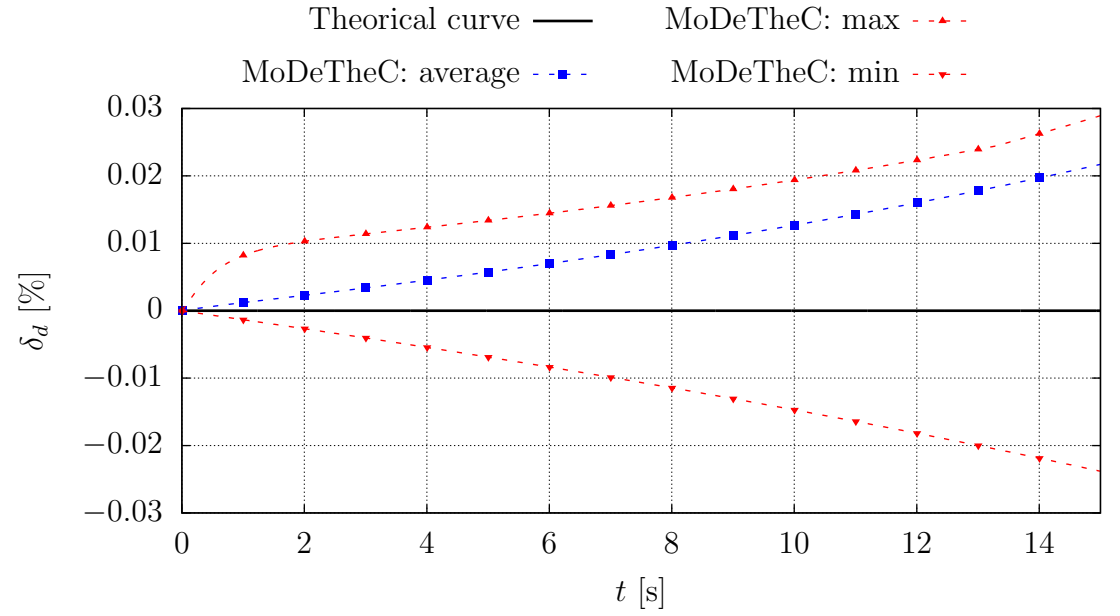


Figure 19: Relative error on the distance between the initial center of the sphere and the curved surface vertices - Imposed ablation velocity on curved boundary - Structured mesh.

composed of 2500 cells (Fig. 21a) is used.

A heat flux $\Phi = 20\,\mathrm{MW/m^2}$ is imposed on the boundary Γ_2 whereas the heat flux on other boundaries is set to zero. To compute the heat transfers and the surface ablation due

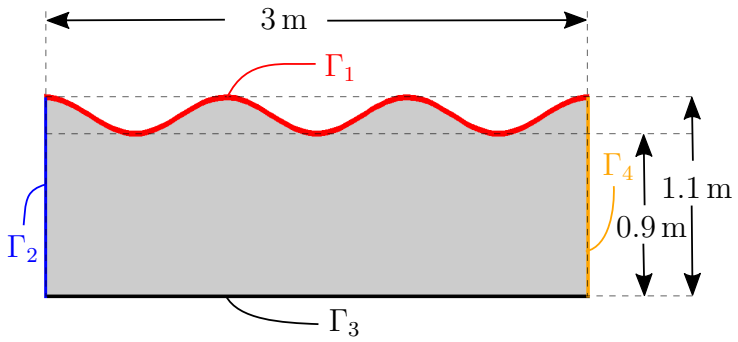
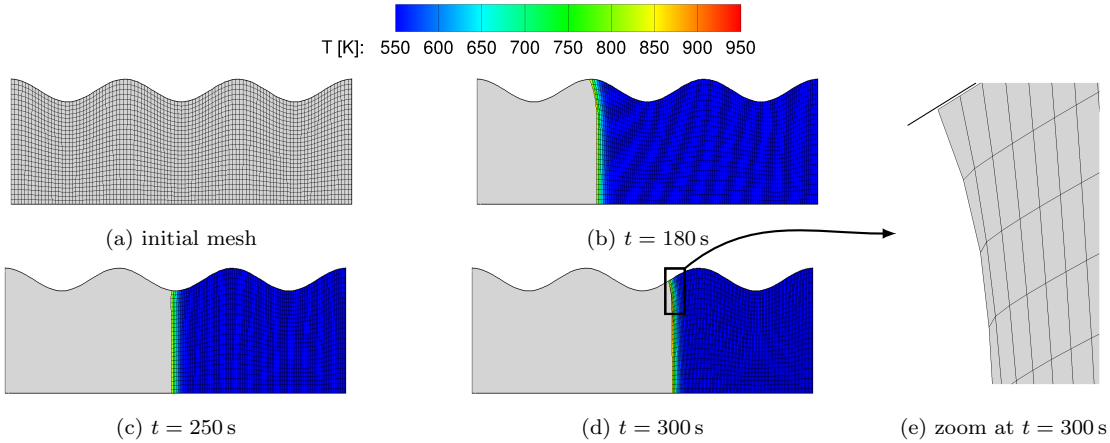


Figure 20: Schema of the *2D Sinusoidal Slider* [14].

to sublimation, the fictive material defined by Droba [14] is used. The ablation model for the sublimation is the latent enthalpy model defined by Mullenix [25]. This computation is run during 400 s with a time step $\delta t = 0.025\,\mathrm{s}$.

Density	$\rho = 2000\,\mathrm{kg/m^3}$
Heat capacity	$C_P = 1000\,\mathrm{J/(kg \cdot K)}$
Thermal conductivity	$\lambda = 0.4\,\mathrm{W/(m \cdot K)}$
Sublimation temperature	$T_{\mathrm{sub}} = 1000\,\mathrm{K}$
Sublimation enthalpy	$\Delta H_{\mathrm{sub}} = 205\,\mathrm{kJ/kg}$
Initial temperature	$T_{\mathrm{i}} = 500\,\mathrm{K}$

Table 1: Material properties for the *Sinusoidal Slider* test cases.



(a) initial mesh (b) $t = 180\,\mathrm{s}$ (c) $t = 250\,\mathrm{s}$ (d) $t = 300\,\mathrm{s}$ (e) zoom at $t = 300\,\mathrm{s}$

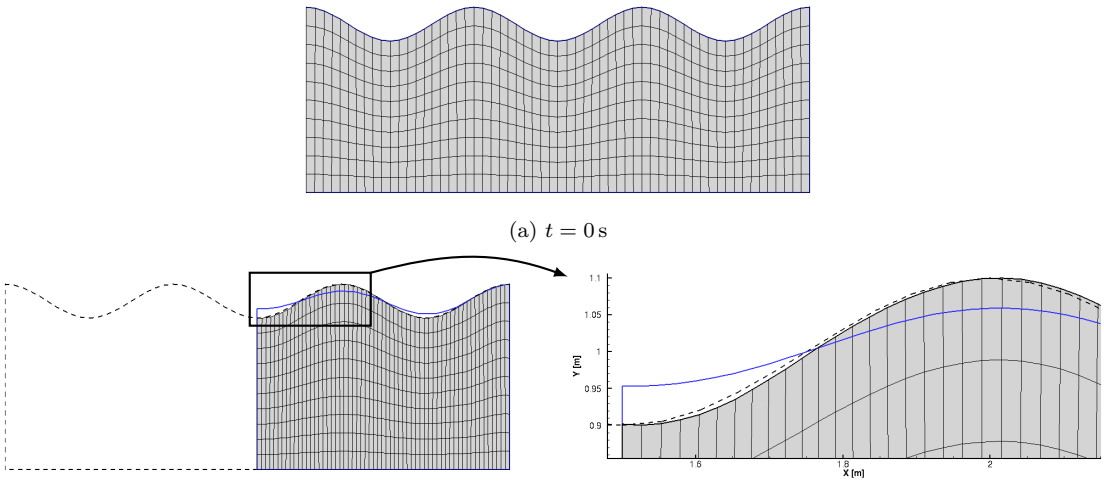
Figure 21: Ablation of the *2D Sinusoidal Slider* with an imposed heat flux.

The type of boundary condition used in MoDeTheC for ablation is not involved in the mesh displacement model. It only modifies the input data, *i.e.* the ablation rate of each face.

The results of the simulation are given in Fig. 21. The shape of the boundary curve Γ_1 is retained and the boundary curve Γ_3 remains straight. However, the boundary curve Γ_2 bends slightly at its junction with the boundary curve Γ_1 due to a difference in heat transfer in the material between the low part of the geometry (near Γ_3) and the high part (near Γ_1). For the same energy input at the boundary curve Γ_2, the lower part of the geometry always heats the same volume close to Γ_3. However, the upper part of the geometry heats either a larger or a smaller volume due to the oscillations of Γ_1. This results in a curvature on the top of Γ_1 while the bottom part remains flat. The combined effect of this curvature with the Laplacian smoothing of the internal vertices stops the computation after 310 s due to cell inversion (Fig. 21e).

The continuation of the computation can be achieved with multiple solutions. The first one is to reduce the time step. This will raise the final time but the error is only shifted. Such a solution also increases the computation cost. Another possibility is to change the balancing method of the internal vertices. The Laplacian smoothing is known to have such issues and might be replaced by an iso-parametric smoothing. The last solution is to re-mesh a part or the entire domain.

The *Sinusoidal Slider* geometry is also used to highlight the necessity of the projection step in the strategy described in Section 4.4.4. A coarse mesh composed of 600 cells (Fig. 22a) is used. For this test case, a constant ablation velocity $\overrightarrow{V}_{\text{imp}} = 0.01\,\text{m/s}$ is set on the boundary curve Γ_2 and no ablation is imposed on the others boundaries. The final time is set to 150 s and the time step is 0.05 s. For comparison purposes, this case is run with and without the projection step implemented in MoDeTheC.



(a) $t = 0\,\text{s}$

(b) $t = 150\,\text{s}$

(c) zoom at $t = 150\,\text{s}$

Figure 22: Ablation of the *2D Sinuosidal Slider* with an imposed ablation velocity: with (in black) and without (in blue) boundary vertex projection.

The result of the mesh displacement strategy is shown in Fig. 22. The initial shape of the *2D Sinuosidal Slider* is represented with dashed black line. The shape of the domain

with and without projection are also shown respectively in black line and in blue line. The result shows that the projection step is mandatory to maintain the sinusoidal shape of the boundary Γ_1 (Fig. 22b and 22c). From the point of view of an ablation computation, material is created around the bottom of the sinusoid and material is removed around the top of it. Finally, Fig. 22 proves the ability of the mesh displacement strategy to keep the object shape even when the mesh is not fine.

The accuracy of the shape-preserving mesh balancing method is validated with a mesh dependancy study shown in Fig. 23. The previous computation with an imposed ablation velocity is performed with 50, 100 and 200 edges on the boundary curve Γ_1. As a comparison the mesh shown in Fig. 22 has 60 edges on Γ_1.

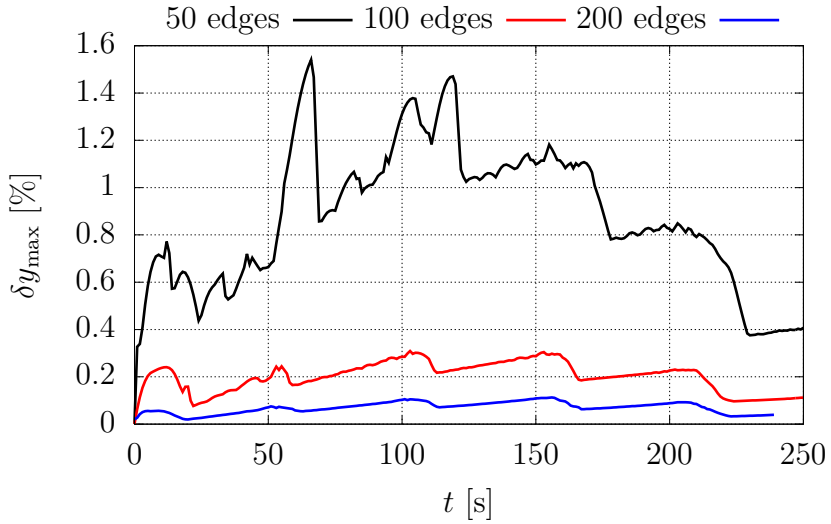
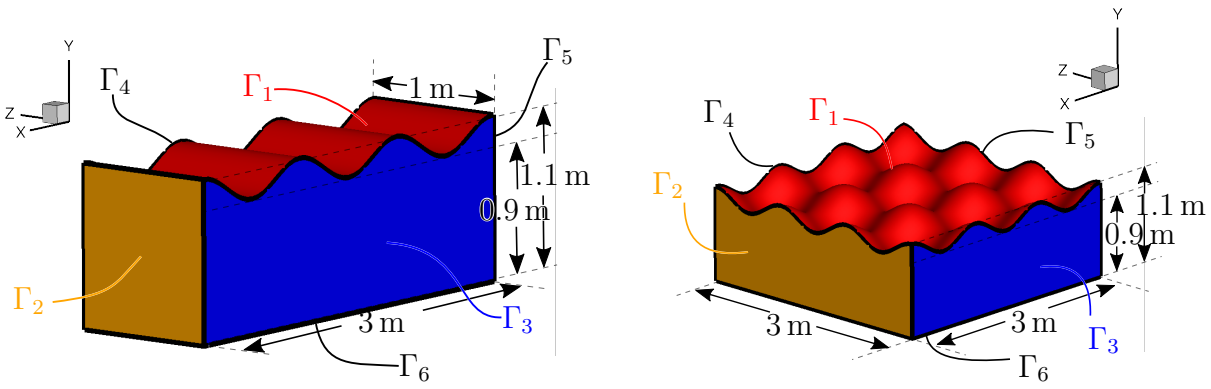


Figure 23: Maximal error for the 2D displacement of the boundary curve Γ_1 during the computation.

At each time step, the maximal error $\delta y_{\max}$ between the numerical and theoretical position of Γ_1 is plotted. Even with a coarse mesh, the error does not exceed 1.6 %. The maximal error is drastically reduced to 0.35 % when a medium mesh is used. The computation with the fine mesh, represented with the blue curve in Fig. 23, does not end at 250 s because of cell inversion.

5.2.3. 3D Sinusoidal Slider

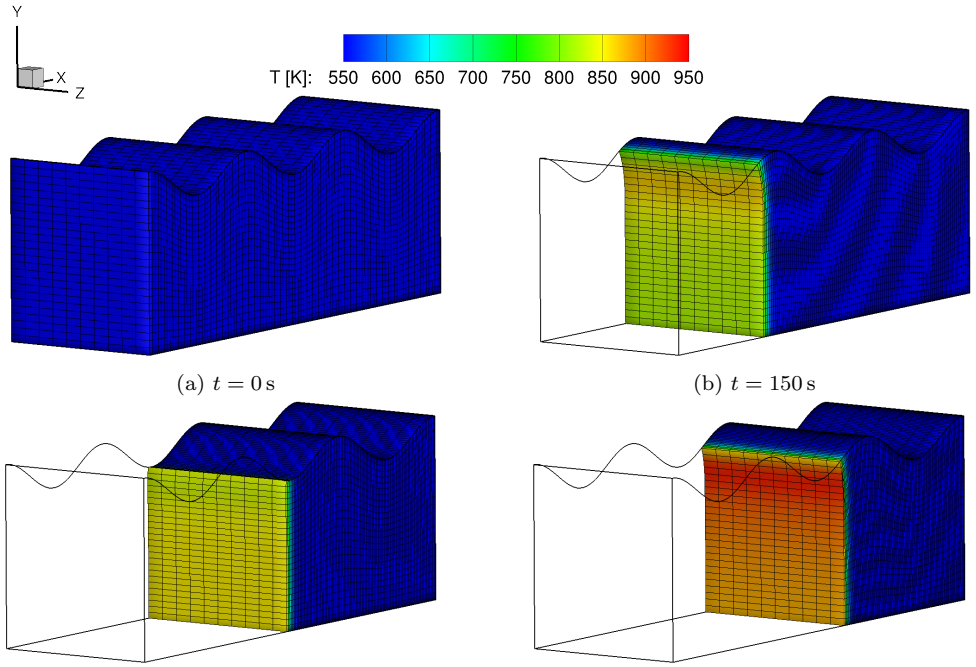
The *2D Sinusoidal Slider* test case proposed by Droba [14] is here extended to 3D geometries to prove the capacity of the mesh displacement strategy to maintain 3D space curves and surfaces. The *2D Sinusoidal Slider* is extended to 3D in two different ways. The first is to extrude the 2D mesh following the third dimension (Fig. 24a) and the second is to define a sinusoid in two space directions (Fig. 24b). These test cases are named *2D-E Sinusoidal Slider* and *3D Sinusoidal Slider* respectively. The *2D-E Sinusoidal Slider* is meshed with a structured mesh with a refinement equivalent to the fine mesh of the *2D Sinusoidal Slider* (Fig. 21a) whereas the *3D Sinusoidal Slider* refinement is equivalent to the coarse mesh of Fig. 22a. These meshes are respectively composed of 25 000 and 36 000 cells. The material properties are unchanged for these computations.



(a) *2D-E Sinusoidal Slider.* (b) *3D Sinusoidal Slider.*

Figure 24: Schema of the *3D Sinusoidal Slider* test cases.

The *2D-E Sinusoidal Slider* is computed during 400 s with a time step $\delta t = 0.025$ s and an imposed heat flux $\Phi = 20\,\mathrm{MW/m^2}$ on Γ_2. The five other boundary surfaces have no heat flux.



(a) $t = 0$ s (b) $t = 150$ s

(c) $t = 250$ s (d) $t = 300$ s

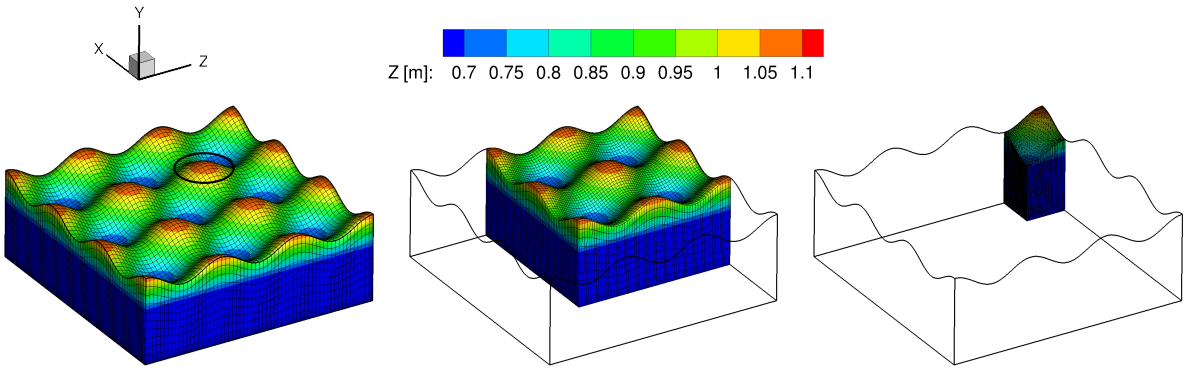
Figure 25: Ablation of the *2D-E Sinusoidal Slider* with an imposed heat flux.

The computation of the *2D-E Sinusoidal Slider* gives similar results to the 2D test case. The shape of the boundary surface Γ_1 is not modified during the ablation process. As for the 2D case, the boundary surface Γ_2 is curved and this curvature induces a cell inversion in the volume and at the boundaries Γ_3 and Γ_4 after 310 s. Moreover, the computation results agree with the expected results since the 2D-E displaced mesh is close to the extrusion of the *2D Sinusoidal Slider* result. The projection and smoothing steps maintain the shape of

the sinusoid while preventing the amplification of the numerical oscillations of the mesh.

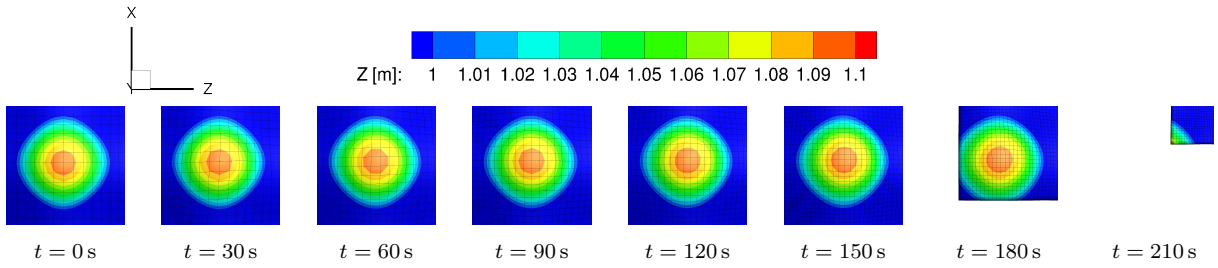
The *3D Sinusoidal Slider* test case is computed during 250 s with a time step $\delta t = 0.05\,\mathrm{s}$ and an imposed ablation velocity $\overrightarrow{V}_{\mathrm{imp}}$ on both Γ_2 and Γ_3 boundary surfaces in Fig. 24b. The mesh during the ablation process is shown with the contour of the Z coordinate (Fig. 26). The reduced set of contour values and the initial boundary curves of the domain illustrate the conservation of the sinusoidal surface. The height of the sinusoid is kept. Moreover, the boundary curves between Γ_1 and Γ_4 and between Γ_1 and Γ_5 strictly follow the initial curves.

The position of the sinusoid is also preserved during the computation even if a small displacement in the ablation direction is observed as illustrated in Fig. 27, where the time evolution of the Y coordinate contour at a specific location (black circle in Fig. 26a) of the sinusoid is shown. In this area, the Y coordinate error for the highest vertex is less than 0.22 % during the computation.



(a) $t = 0\,\mathrm{s}$ (b) $t = 100\,\mathrm{s}$ (c) $t = 250\,\mathrm{s}$

Figure 26: Ablation of the *3D Sinusoidal Slider* with imposed ablation velocity.



$t = 0\,\mathrm{s}$ $t = 30\,\mathrm{s}$ $t = 60\,\mathrm{s}$ $t = 90\,\mathrm{s}$ $t = 120\,\mathrm{s}$ $t = 150\,\mathrm{s}$ $t = 180\,\mathrm{s}$ $t = 210\,\mathrm{s}$

Figure 27: Zoom on a bump of figure 26 during the computation.

A equivalent quantitative study as for the for *2D Sinusoidal Slider* is performed with the 2D-E geometry. The mesh refinement is changed for the boundary surface Γ_1 in Fig. 24a. The maximal error shown in Fig. 28 is very similar to the results of Fig. 23. The coarse mesh reaches a maximal error of 1.6 % and the medium mesh error does not exceed 0.35 %. This proves the ability of the mesh displacement strategy to maintain the 3D shape when large ablation occurs.

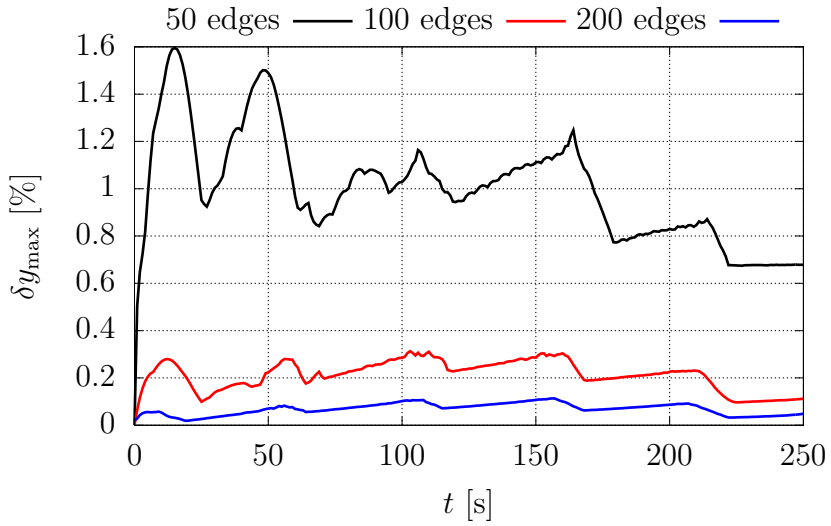


Figure 28: Maximal error for the 2D-E displacement of the boundary surface Γ_1 during the computation.

5.2.4. Doubly Receding Rectangle

The *Doubly Receding Rectangle* test case proposed by Droba [14] challenges the capacity of the mesh displacement method to smooth a geometry. The domain consists in a rectangle where a heat flux $\phi = 4\,\mathrm{MW/m^2}$ is set to the boundary curves Γ_1 and Γ_2 (Fig. 29). The heat flux of the boundary curves Γ_3 and Γ_4 is set to zero. The computation is run during 500 s with a time step $\delta t = 0.05\,\mathrm{s}$. For this computation, an hybrid mesh composed of 2374 cells is generated with 7 layers of quadrilateral close to Γ_1 and Γ_2 (Fig. 30a).

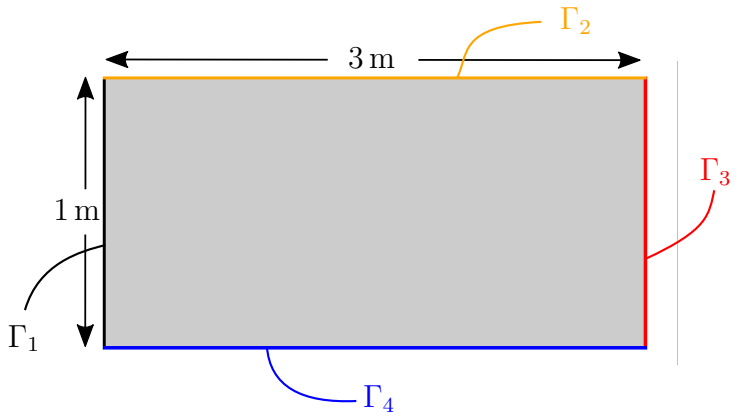
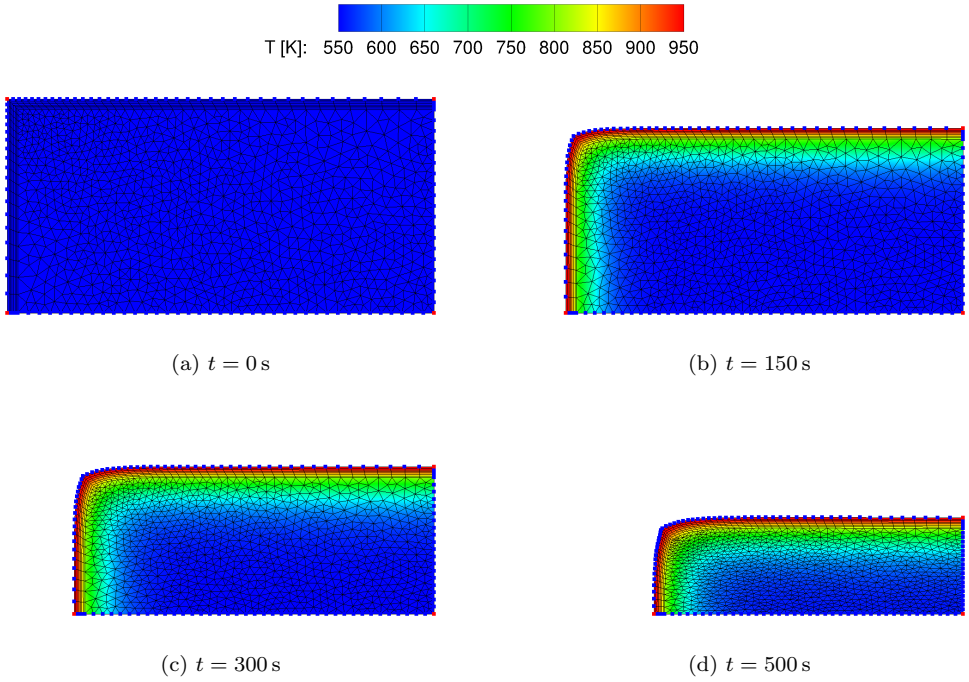


Figure 29: Schema of the *2D Doubly Receding Rectangle* [14].

The thermal response of the theoretical material defined in Table 1 is finally presented in Fig. 30 during the ablation computation. At 80 s, the upper left corner is smoothed because the energy received at the corner is higher due to the heat flux on both boundaries. Up to 500 s, the smoothed geometry is ablated and the size of the cell layers close to Γ_1 and Γ_2 are maintained.



(a) $t = 0\,\mathrm{s}$ (b) $t = 150\,\mathrm{s}$

(c) $t = 300\,\mathrm{s}$ (d) $t = 500\,\mathrm{s}$

Figure 30: View of the temperature field inside the material during the ablation of the *2D Doubly Receding* with imposed heat flux.

The *Doubly Receding Rectangle* test case is also used to highlight the impact of the smoothing step during an ablation computation. The same case is thus performed twice: firstly with the strategy of Section 4 (in black) and secondly by disabling the smoothing step of the procedure (in red in Fig. 31).

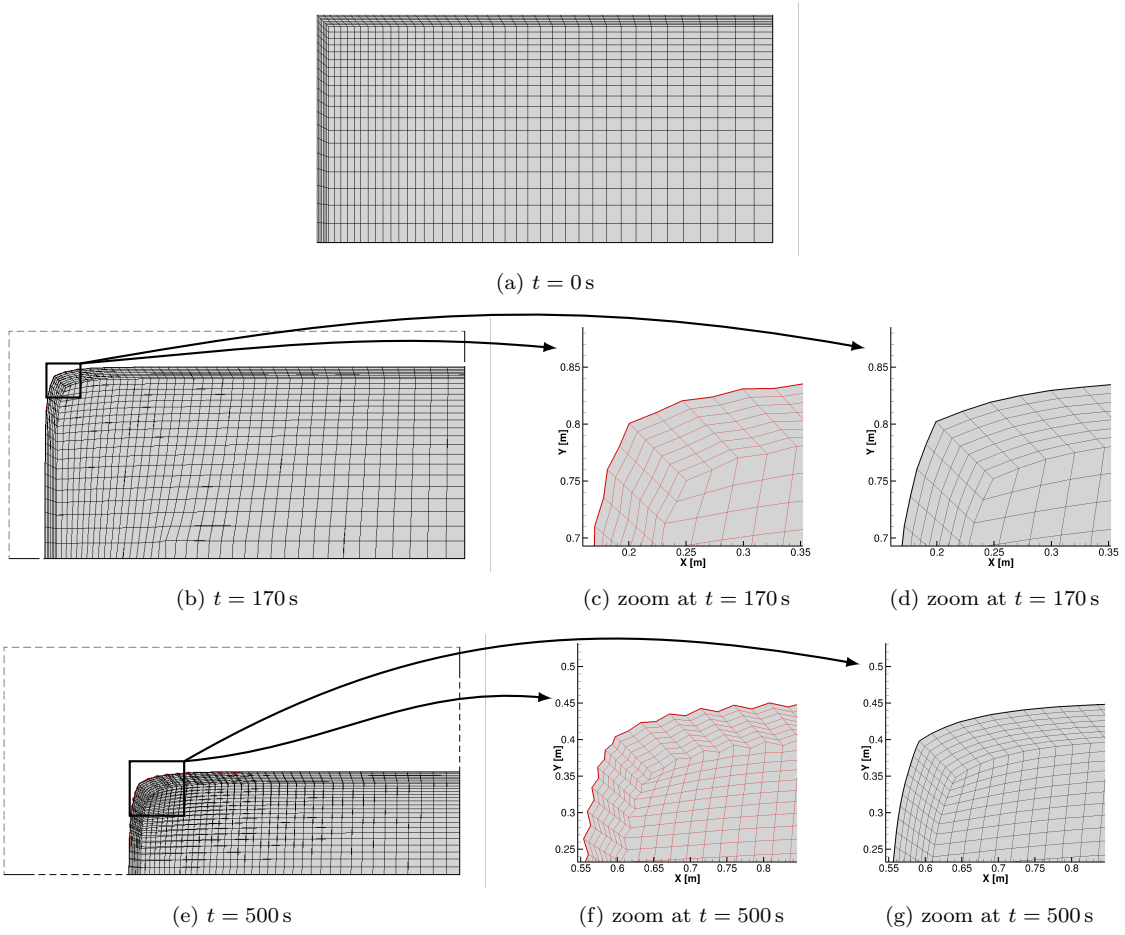
The computation is performed on a block structured mesh (Fig. 31a). The result of the computation at $t = 170\,\mathrm{s}$ and $t = 500\,\mathrm{s}$ is shown in Fig. 31b and 31e. The impact of the mesh smoothing step during the computation is seen with the close views.

At $t = 170\,\mathrm{s}$, both methods smooth the corner vertex between Γ_1 and Γ_2. However, the zooms at the upper left corner vertex show that an undesirable noise appears without the smoothing step (Fig. 31c). During the computation, this noise increases and the resulting mesh at $t = 500\,\mathrm{s}$ is shown in Fig. 31f with strong oscillations. With the smoothing step described in Section 4.5, this noise is removed during the computation without shape flattening (Fig. 31d and 31g).

5.2.5. 3D Receding Cuboid

The general formulation of the mesh displacement strategy described in this paper allows to smooth 2D and 3D geometries. The *Doubly Receding Rectangle* test case is here converted in 3D in two different ways. The first is an extrusion of the *Doubly Receding Rectangle* (Fig. 32a) and the second is a triple receding cuboid (Fig. 32b). These test cases are named *3D Doubly Receding* and *3D Triple Receding* respectively.

The *3D Doubly Receding* geometry is meshed (11 470 cells) with 3 structured blocks to

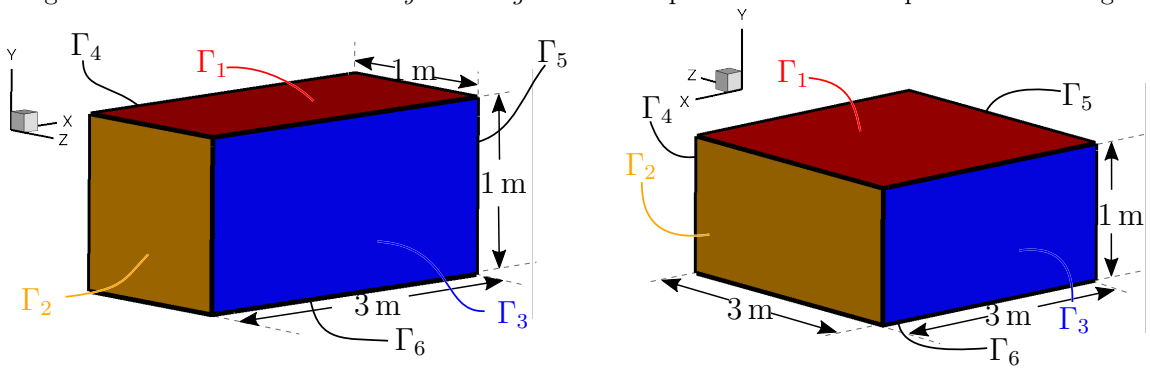


(a) $t = 0\,\mathrm{s}$

(b) $t = 170\,\mathrm{s}$ (c) zoom at $t = 170\,\mathrm{s}$ (d) zoom at $t = 170\,\mathrm{s}$

(e) $t = 500\,\mathrm{s}$ (f) zoom at $t = 500\,\mathrm{s}$ (g) zoom at $t = 500\,\mathrm{s}$

Figure 31: Ablation of the *Doubly Receding 2D* with imposed heat flux - Impact of smoothing step.

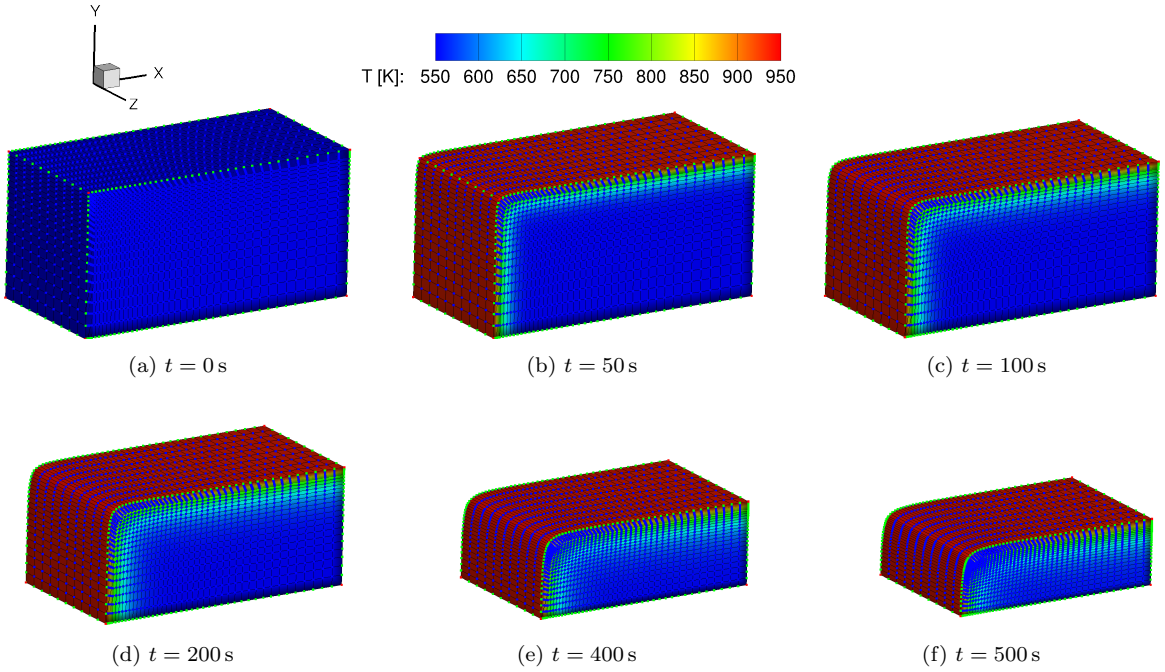


(a) Doubly receding. (b) Triple receding.

Figure 32: Schema of the *3D Receding Cuboid* test cases.

refine the mesh close to the boundary surfaces Γ_1 and Γ_2. The numerical parameters are the same as for the *2D Receding Rectangle*. So, an heat flux $\phi = 4\,\mathrm{MW/m^2}$ is set to the boundary surfaces Γ_1 and Γ_2 whereas the heat flux on the other boundaries is set to zero. During the computation, the boundary curve between Γ_1 and Γ_2 is smoothed (Fig. 33).

In the same time, the boundary surfaces Γ_3, Γ_4, Γ_5 and Γ_6 are maintained in their initial plane without any imposed sliding condition. Finally, the boundary curves and surfaces are smoothed and no undesirable noise appears.



(a) $t = 0\,\mathrm{s}$ (b) $t = 50\,\mathrm{s}$ (c) $t = 100\,\mathrm{s}$

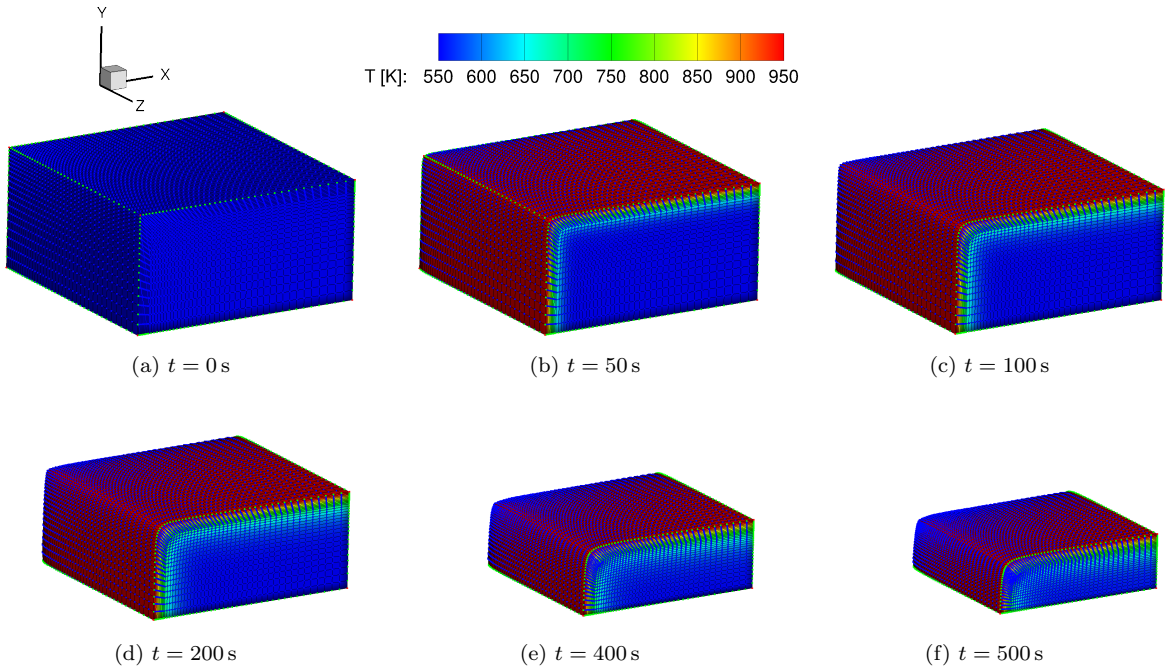
(d) $t = 200\,\mathrm{s}$ (e) $t = 400\,\mathrm{s}$ (f) $t = 500\,\mathrm{s}$

Figure 33: View of the temperature field inside the material during the ablation of the *Doubly Receding Cuboid* with imposed heat flux.

The *3D Triple Receding* test case challenges the ability of MoDeTheC to smooth corner vertex. To do so, the heat flux $\phi = 4\,\mathrm{MW/m^2}$ is set to three boundary surfaces: Γ_1, Γ_2 and Γ_4. The domain is meshed (55 926 cells) with structured blocks to refine the mesh close to the boundary surfaces where the heat flux is applied. The mesh displacement correctly smoothes the corner vertex and the boundary curves, as seen in Fig. 34.

6. Conclusion

A 3D mesh displacement strategy for the simulation of material degradation during atmospheric reentry of space debris is proposed. The strategy requires three independent steps, all based on a sharp edge and node detection. First, the mesh shape is modified to take into account ablation due to physico-chemical phenomena. The present method computes the node displacement such that any geometry can be managed. Then, the boundary and internal nodes are redistributed to maintain a good mesh quality. The combination of the Laplacian operator and a projection step on curves and surfaces enables the mesh to be smoothed at each time-step without any undesirable flattening. Finally, the mesh is smoothed to prevent high frequency oscillations of the boundary surfaces and curves. This 3D strategy is compliant with structured, unstructured and hybrid meshes. Moreover, for space debris application, any material surface can be ablated without having to predetermine or specific boundary conditions.



(a) $t = 0\,\mathrm{s}$ (b) $t = 50\,\mathrm{s}$ (c) $t = 100\,\mathrm{s}$

(d) $t = 200\,\mathrm{s}$ (e) $t = 400\,\mathrm{s}$ (f) $t = 500\,\mathrm{s}$

Figure 34: View of the temperature field inside the material during the ablation of the *Triple Receding Cuboid* with imposed heat flux.

This strategy is implemented in the finite volume material response code MoDeTheC which uses an ALE approach with an interface tracking method. However, neither the finite volume formulation nor the ALE method are prerequisite for this strategy. The general formulation of the node displacement, the mesh balancing and the mesh smoothing allows to use this strategy in different solvers. Moreover, the mesh displacement step can easily be coupled to another balancing and smoothing methods because all these steps are independent.

The verification of the strategy implemented in MoDeTheC was done on many 2D/3D different shapes and ablation configurations (imposed velocity or sublimation model). Some examples were shown to justify the choice of the 3D mesh displacement strategy and to prove its accurate behaviour.

Future work consists in the complete validation of MoDeTheC ablation models. Simulation with increasing complexity will be computed to take into account all the physical phenomena of the material degradation. These computations will be compared to other ablation codes and to experimental results. Moreover, the ablation of other geometries, representative of space debris, will be checked.

Finally, one objective of space debris material ablation is to compute the degradation of the entire object. To do so, the mesh displacement strategy described in this paper and based on the ALE formulation is not enough. So, a local and/or global remesh algorithm should be implemented in MoDeTheC in the future.

Acknowledgments

This work was supported by the ONERA (French Space Laboratory) and CNES (French Space Agency) through the co-funding of the PhD of N. Perron.

Appendix A. Interface and layer treatments for internal balancing

The mesh balancing used in MoDeTheC can compute large mesh deformation thanks to the use of the Laplacian operator $\overrightarrow{\mathscr{L}}$, to the special treatment imposed on interfaces and to the cell layer displacement described in Section 4.4.

The doubly receding rectangle test case presented in Section 5.2.4 is here used with the block structured mesh shown in Fig. A.35a. The balancing is done with the Laplacian operator $\overrightarrow{\mathscr{L}}$ and three options are compared:

- without interface and layer treatments (Fig. A.35b);
- with interface treatment and without layer treatment (Fig. A.35c);
- with interface and layer treatments (Fig. A.35d).

At $t = 100\,\mathrm{s}$, the displacement of internal and boundary vertices is large enough for the parameter α defined in Section 4.4.2 to be equal to 1. The mesh balancing is therefore applied with $\overrightarrow{\mathscr{L}}_2$ operator. The mesh distortion due to $\overrightarrow{\mathscr{L}}_2$ operator is seen in Fig. A.35b were the boundary layer refinement is lost. The balancing of the interface vertices circled in Fig. A.35a with the operator $\overrightarrow{\mathscr{L}}_1$ improves the mesh quality (Fig. A.35c). However, the size of the cell layers close to the upper left corner is increased due to the mesh curvature. This issue is solved with the cell layer treatment as shown in Fig. A.35d. Thus, with the interface and layer treatments, the quality of the mesh is maintained up to 500 s in the all domain, and particularly close the boundaries Γ_1 and Γ_2.

References

[1] J.-C. Liou, J. Opiela, A. Vavrin, B. Draeger, P. Anzmeador, C. Ostrom, Debris assessment software user's guide, Tech. rep., NASA, NASA/TP-2020-5002380 (2020).

[2] A. Kato, Y. Hyodo, K. Kishida, Toward the world common re-entry safety assessment procedure, Tech. rep., Japan Society for Aeronautical and Space Sciences, ISTS 2002-n-14 (2002).

[3] P. Omaly, J. Annaloro, DEBRISK v3: CNES tool evolutions for re-entry risk analysis, IAASS, 2019.

[4] I. Holbrough, J. Beck, V. Braun, S. Lemmens, Upgrading the re-entry risk component of esa's drama software, Vol. 8, 8th European Conference on Space Debris, ESA Space Debris Office, 2021.

[5] R. Kanzler, T. Lips, B. Fritsche, A. Breslau, P. Kärräng, M. Spel, A. Pagan, G. Herdrich, S. Lemmens, SCARAB 4 - Extension of the high-fidelity re-entry break-up simulation software based on new measurement types, Vol. 8, 8th European Conference on Space Debris, ESA Space Debris Office, 2021.

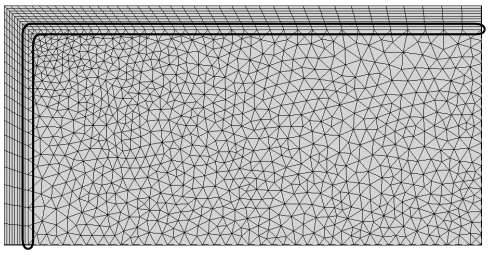
[6] J. Annaloro, P. Omaly, V. Rivola, M. Spel, Elaboration of a new spacecraft-oriented tool : PAMPERO, 8th European Symposium on Aerothermodynamics for Space Vehicles, 2015.

[7] Y. Prévereaud, Contribution à la modélisation de la rentrée atmospherique des débris spatiaux., PhD Thesis, Université de Toulouse, Toulouse, France (2014).

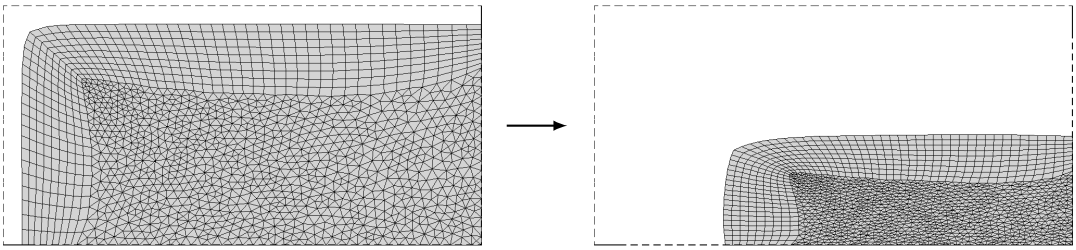
[8] J. Lachaud, T. Magin, I. Cozmuta, N. Mansour, A short review of ablative-material response models and simulation tools, 7th Aerothermodynamics Symposium, European Space Agency, Brugge, Belgium, 2011.

[9] J. Lachaud, N. N. Mansour, Porous-material analysis toolbox based on openfoam and applications, Journal of Thermophysics and Heat Transfer 28 (2) (2014) 191–202.

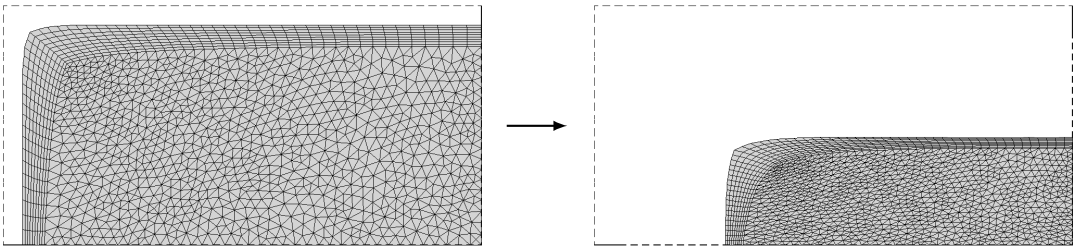
[10] SAMCEF Amaryllis, http://www.sigmeo.fr/samcef, Accessed 3 November 2020.
[11] V. Biasi, Modelisation thermique de la dégradation d'un matériau composite soumis au feu, PhD Thesis, Université de Toulouse, Toulouse, France (2014).
[12] J. Wiebenga, I. Boyd, Computation of multi-dimensional material response coupled to hypersonic flow, 43rd AIAA Thermophysics Conference, 2012. doi:10.2514/6.2012-2873.
[13] S. Peluchon, Approximation numérique et modélisation de l'ablation liquide, Ph.D. thesis, L'Université de Bordeaux (2017).
[14] J. Droba, Tangle-free Finite Element Mesh Motion for Ablation Problems, 46th AIAA Thermophysics Conference, 2016. doi:10.2514/6.2016-3386.
[15] X. Jiao, Face offsetting: A unified approach for explicit moving interfaces, Journal of Computational Physics 220 (2006). doi:10.1016/j.jcp.2006.05.021.
[16] T. Tang, Moving mesh methods for computational fluid dynamics, Contemp. Math. 383 (2005). doi:10.1090/conm/383/07162.
[17] W. Daldoul, Moving mesh for complex geometries, Research report, Ecole Nationale Supérieure des Mines de Paris (2017).
[18] R. Garimella, M. Shashkov, P. M. Knupp, Triangular and quadrilateral surface mesh quality optimization using local parametrization, Computer Methods in Applied Mechanics and Engineering 193 (2004) 913–928. doi:10.1016/j.cma.2003.08.004.
[19] A. Belyaev, Y. Ohtake, A comparison of mesh smoothing methods, Israel-Korea Bi-National Conference on Geometric Modeling and Computer Graphics, Tel Aviv University (2003).
[20] Y. Ohtake, A. Belyaev, I. Bogaevski, Mesh regularization and adaptive smoothing, Computer-Aided Design 33 (2001) 789–800. doi:10.1016/S0010-4485(01)00095-1.
[21] X. Tong, D. Thompson, Q. Arnoldus, E. Collins, E. Luke, Three-dimensional surface evolution and mesh deformation for aircraft icing applications, Journal of Aircraft 54 (3) (2017) 1047–1063.
[22] R. T. Farouki, N. Szafran, L. Biard, Construction of bézier surface patches with bézier curves as geodesic boundaries, Computer-Aided Design 41 (2009) 772–781.
[23] Shalizi, C., Lecture 24-25: Weighted and generalized least squares, https://www.stat.cmu.edu/ cshalizi/mreg/15/lectures/24/lecture-24--25.pdf (visited on 4/12/2020).
[24] C. Geuzaine, J.-F. Remacle, Gmsh: A three-dimensional finite element mesh generator with built-in pre-and post-processing facilities 79 (2008).
[25] N. Mullenix, A coupled gas dynamics and heat transfer method for simulating the laser ablation process of carbon nanotube production, Master's thesis (2005).



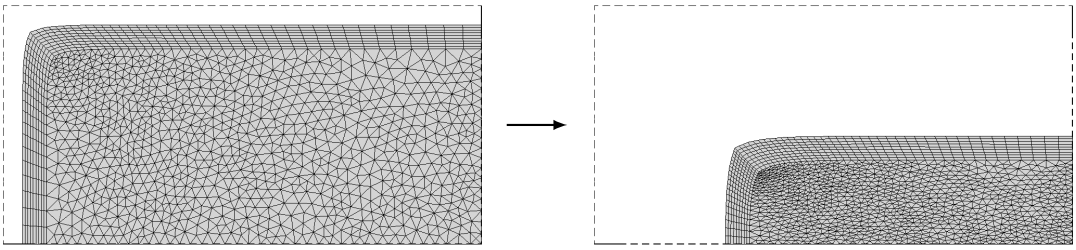
(a) Initial mesh.



(b) Without interface and layer treatments.



(c) With interface treatment and without layer treatment.



(d) With interface and layer treatments.

Figure A.35: Ablation of the *Doubly Receding 2D* with imposed heat flux - Impact of the interface and layer treatments on internal balancing.