

Alma Mater Studiorum Università di Bologna  
Archivio istituzionale della ricerca

Numerical Evaluation of the Effects of Inclusions on Solid Rocket Motor Performance

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

*Published Version:*

Fabrizio Ponti, S.M. (2020). Numerical Evaluation of the Effects of Inclusions on Solid Rocket Motor Performance. AIAA JOURNAL, 58(9), 4028-4036 [10.2514/1.J058735].

*Availability:*

This version is available at: <https://hdl.handle.net/11585/769367> since: 2024-09-05

*Published:*

DOI: <http://doi.org/10.2514/1.J058735>

*Terms of use:*

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>).  
When citing, please refer to the published version.

(Article begins on next page)

# Numerical Evaluation of the Effects of Inclusions on Solid Rocket Motor Performance

F. Ponti\* and S. Mini†

University of Bologna, FC 47121 Forlì, Italy

and

A. Annovazzi‡

Avio S.p.A., 00034 Colleferro, Italy

<https://doi.org/10.2514/1.J058735>

This work describes the application of original computer graphics methods to examine the effect of inclusions within the solid rocket motor thrust chamber. The adoptance of a dynamic three-dimensional triangular mesh with a self-intersection removal algorithm, as the core of the burning surface regression module, allows the aforementioned target to be reached. Many phenomena not measurable in an analytical closed form, like different-shaped voids within the grain, can be investigated. Indeed, both local heterogeneities and surface imperfections could generate unexpected variations in the combustion chamber pressure pattern and early thermal protections exposure. The aforementioned techniques are developed and tested on a ballistic simulator developed using computer graphics methods to improve mesh handling. Simulation results are carried out and discussed.

## Nomenclature

$a$	=	burning rate experimental factor, $m^{1-n} \cdot s^{2n-1} \cdot kg^{-n}$
$F_{anis_i}$	=	anisotropy factor of burning surface mesh $i$ th node
$n$	=	burning rate experimental exponent
$p_{0_i}$	=	combustion chamber pressure of burning surface mesh $i$ th node, Pa
$r_{b_i}$	=	burning rate of burning surface mesh $i$ th node, m/s
$t$	=	time, s
$t_0$	=	reference time, s
$\Delta S_i$	=	displacement of burning surface mesh $i$ th node, m
$\Delta t$	=	simulation time step, s

## I. Introduction

IN THE design and development of a new solid propellant rocket motor (SRM), the use of numerical tools able to simulate, predict, and reconstruct [1] the behavior of a given motor in all its operative conditions is crucial: more accurate performance predictions would reduce the need of expensive testing, thus decreasing the cost of development for each SRM [2]. The main factor influencing the combustion chamber pressure, and hence the thrust, is the time evolution of the burning surface and the surface regression rate. Indeed, in the real case, a lot of factors affect the propellant combustion [3–6], resulting both in propellant rate variation from the theoretical one and earlier protections exposure than expected. As an example, the occurrence of inclusions within the grain affects the time instant at which the burning surface reaches the SRM case, increasing the time exposure of the thermal protections to a high temperature, as will be mentioned in the following. Inclusions are usually generated during the propellant casting process because of air that remains trapped within the grain due to propellant's high viscosity and high density. These properties seize the air inside the propellant, which

causes the porosity in grain. When the air inclusions are exposed to combustion, the flame propagation through them is faster, thus causing a significative burning surface progression increase, and therefore a sooner exposure of the case thermal protections to the high-temperature gases produced by combustion.

Since thermal protections are designed to withstand a certain amount of thermal stresses for a theoretic time interval, if they are not thick enough to absorb all the heat coming from the combustion chamber, they are no longer able to insulate the case itself [7]. The final consequence could even be the launcher failure. Hence, the capability to estimate the aforementioned phenomena is of great importance in order to ensure the proper functioning of the launcher itself.

A variety of computer graphics procedures linked to solid rocket internal ballistic simulators have been proposed in the literature to numerically compute combustion chamber thermodynamic parameter variation with the thrust–time profile as the main goal. One example is represented by SPP (which is a solid propellant rocket motor performance computer program) [8]. This simulation software shows three available approaches (two-dimensional, axisymmetric, and three-dimensional): all of them based on Boolean geometry methods. It consists of a combination or intersection of primitive solids (cones, spheres, prisms, cylinders, tori) used to recreate the most common rocket propellant configurations (finocyl design, tapered star design, etc.). The same approach is used in the PIBAL (which is a propulsion and internal ballistic software) environment, which was developed by Société Nationale des Poudres et des Explosifs Propulsion [9]. It includes several grain macrofunctions: each related to a specific two-dimensional (2-D) or three-dimensional (3-D) grain shape (axisymmetric multi-fin motor, wagon wheel, etc.). The aforementioned technique, although it is effective in terms of computing surface and volume shapes, is unable to deal with inclusions and defects that can generate self-intersections during surface regression.

On the other hand, a recent technique to track the interface evolution is represented by the family of level-set methods, which was first introduced by Osher and Sethian [10]. According to the classical approach [11], the interface (namely a 2-D/3-D surface) is computed as the zero contour of a higher-dimensional function (i.e., the level-set function). From a geometrical point of view, the interface is computed as the intersection between a higher-dimensional function (hyperbolic surface in Fig. 1) and a plane identifying the zero-contour section.

The level-set function is usually defined as the Euclidian distance to the interface [11]. If a velocity field  $\mathbf{v}(x, y, z)$  is introduced, then the level-set function is deformed through a time variation and a spatial variation, depending on the scalar product between  $\mathbf{v}$  and the spatial gradient of the level-set function itself. The resulting initial

Presented as Paper 2019-3959 at the Propulsion and Energy Forum and Exposition, Indianapolis, IN, August 19–22, 2019; received 6 September 2019; revision received 17 April 2020; accepted for publication 17 April 2020; published online 18 May 2020. Copyright © 2020 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved. All requests for copying and permission to reprint should be submitted to CCC at [www.copyright.com](http://www.copyright.com); employ the eISSN 1533-385X to initiate your request. See also AIAA Rights and Permissions [www.aiaa.org/randp](http://www.aiaa.org/randp).

\*Full Professor, Department DIN, via Fontanelle 40; [fabrizio.ponti@unibo.it](mailto:fabrizio.ponti@unibo.it)

†Ph.D. Candidate, Department DIN, via Fontanelle 40; [stefano.mini3@unibo.it](mailto:stefano.mini3@unibo.it)

‡Senior Engineer, AVIO Space Propulsion Design Department, Rome; [adriano.annovazzi@avio.com](mailto:adriano.annovazzi@avio.com)

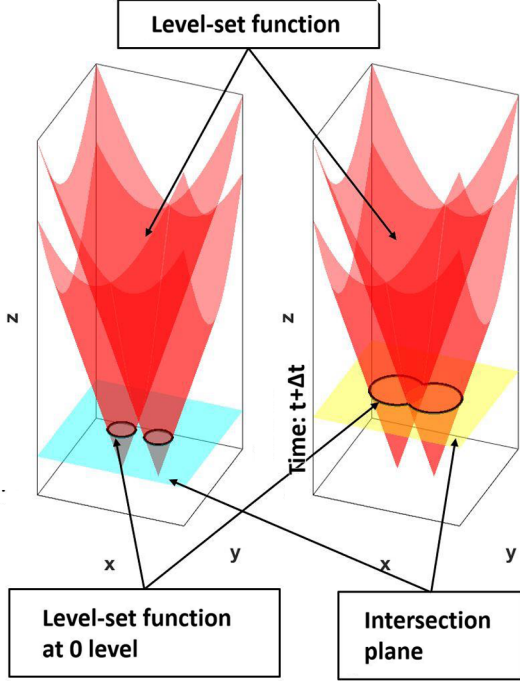


Fig. 1 Level-set method.

value partial differential equation for the evolution of the level-set function is close to a Hamilton–Jacobi equation [10], and it is solved according to the hyperbolic conservation laws. Practically, level-set method implementation is equivalent to the numerical integration of level-set function partial derivatives in time and space. The main advantage of that method is an easy evaluation of surface curvature, and normal and topology evolution. However, the section-by-section description used by the level-set approach makes the velocity field have some form of prevalent orientation.

Some improvements of the level-set method have been addressed by Willcox et al. [12,13] by using a minimum distance function (MDF) approach in the Rocballist software. The minimum distance to the initial surface can be used to represent the detailed shape of the burning surface as it evolves in time: indeed, the MDF sign indicates whether the grid point is on the solid or gas side of the interface. In contrast to the level-set method, there is no hyperbolic partial differential equation to be solved. Despite this, the method is efficient in terms of computational time; the main disadvantages are the inapplicability to radial and azimuthal burn rate variations (with respect to axial variation) and the function sign error due to the presence of sharp-angled cones pointing toward the chamber volume: in practice, sharp-angled cones could appear as inclusions within the grain itself. The use of a dynamic 3-D triangular mesh as the core of the burning surface regression module in the ROBOOST (which is a rocket boost simulation tool) performance estimation code allows us to reach the aforementioned target with little penalty of accuracy due to the finite mesh discretization into triangular elements. A self-intersection removal procedure, based on triangle–triangle intersection, can be used to preserve mesh coherence [14]: each triangular mesh element is meant to be a pure geometrical representation to let the triangle vertices (nodes of the mesh) translate in the domain, simulating in that way the burning surface regression process.

The numerical procedure described in this paper and integrated in the ballistic simulator ROBOOST [15] is meant to overcome all those limitations, proposing a new effective approach in order to give more detailed estimations of SRM burning surface regression, dealing with inclusions/nonconventional grain geometry configurations. Finally, the burning surface regression of a multiperforated SRM is considered in order to ensure the effectiveness of the self-intersections removal procedure because the multiperforated configuration, in its burning surface evolution, generates challenging self-intersections.

## II. Code Overview

### A. ROBOOST Code Structure

ROBOOST is a ballistic simulation software developed in the MATLAB environment at the Laboratory of Propulsion and Mechanics of the University of Bologna in collaboration with AVIO S.p.A. Figure 2 shows the modular structures of the code: each module is assigned to a specific function.

The general approach is based on the burning surface regression, which is modeled as a 3-D triangular mesh (denoted as the grain regression module in Fig. 2). The same discretization procedure is used to obtain the numerical domain of the cavities/inclusions within the grain. The above mentioned strategy enables ROBOOST to handle generic geometric shapes (in terms of inclusions and propellant configurations): indeed, it can accept Initial Graphics Exchange Specification format files as input.

Mesh node regression is obtained by adding a displacement along the normal direction of each node [15], for which the amount is expressed by Eq. (1):

$$\Delta S_i = r_{b_i} \cdot \Delta t \quad (1)$$

$$r_{b_i} = F_{anis_i} \cdot a \cdot p_{0_i}^n \quad (2)$$

Because each node is managed independently from the others, point-by-point surface motion can be defined by introducing grain anisotropies as a burn rate variation table. This table can be included in the anisotropies computation module (Fig. 2), for which the output is given in terms of a normalized factor  $F_{anis_i}$  [Eq. (2)]. That feature allows a more “physical” description of the surface regression process, especially in providing burn rate heterogeneities to the simulator. During simulations, since mesh triangular elements are indirectly deformed and compressed by displacements, a remeshing module is needed in order to preserve mesh resolution and coherence (the mesh should be two-manifold coherent) [14]. For instance, the increase of the triangular element dimension, during burning surface evolution, can produce overlapped triangles that must be removed to obtain a correct mesh evolution. The remeshing module includes procedures that preserve the domain length of single edges or triangle areas as well, maintaining them in the user-defined range by collapsing anomalous elements.

The second fundamental block ROBOOST consists of is the ballistics module: it consists of a one-dimensional (1-D) unsteady

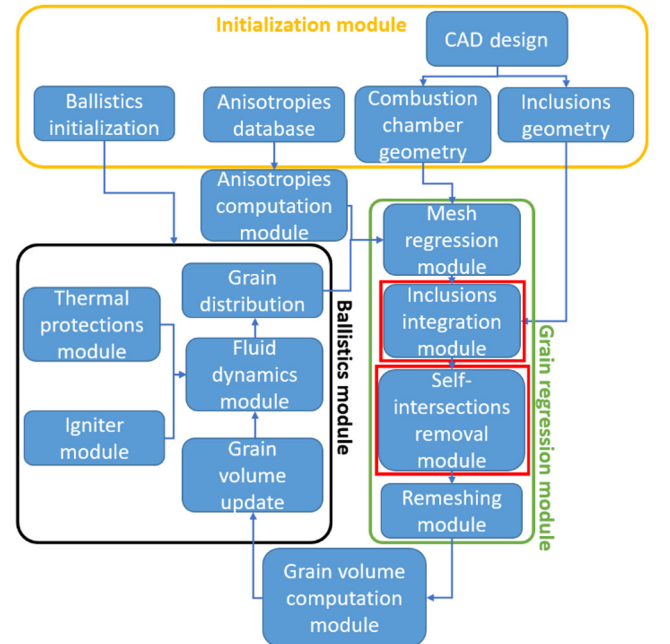


Fig. 2 ROBOOST modules layout.

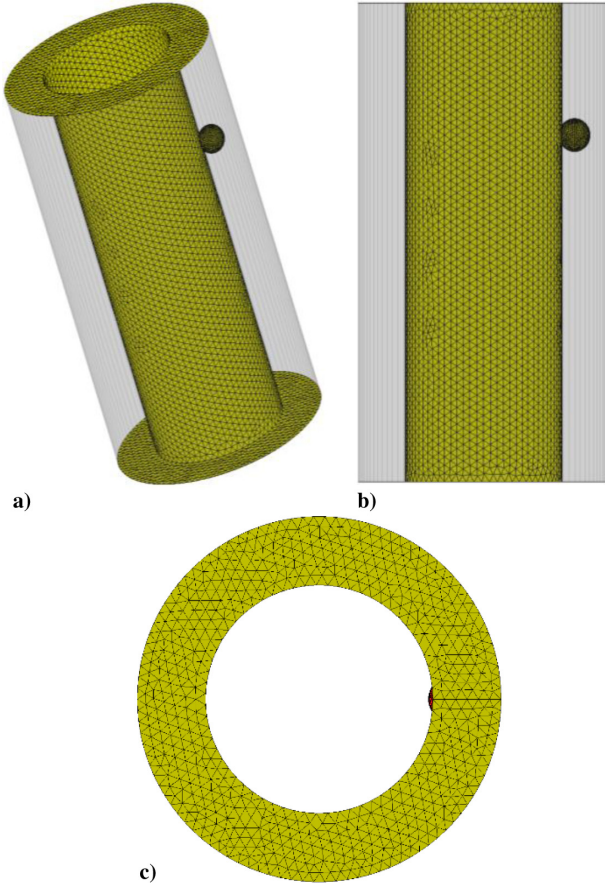


Fig. 3 Self-intersections generation.

fluid dynamics solved along the thrust chamber symmetry axis [14–16]. Thus, velocity and pressure fields of the flows are obtained with a low computational effort and a little penalty in time and space. The grain regression module is linked to the ballistics module in the following way: at a certain time interval  $\Delta t = t - t_0$ , first, the regression module estimates the amount of burned propellant subtracting propellant volume at time  $t$  from propellant volume at time  $t_0$ . Then, the ballistics module estimates the thermodynamic parameters variation within the combustion chamber; the new burning rate is computed, starting from the combustion chamber pressure by means of Eq. (2). Finally, the new burning rate estimation is used to compute

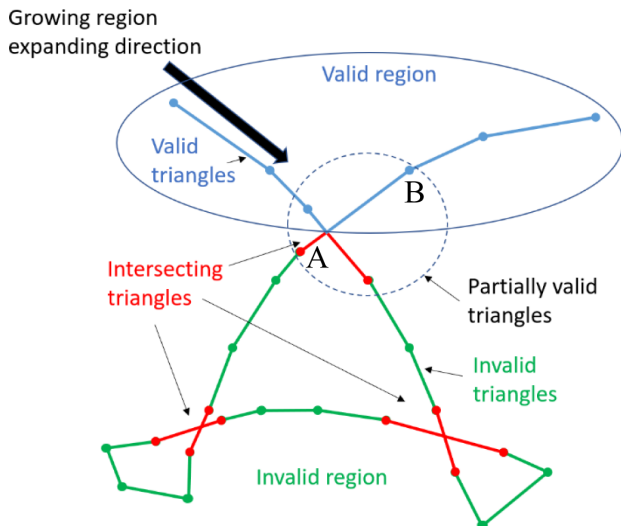


Fig. 4 Valid/invalid region.

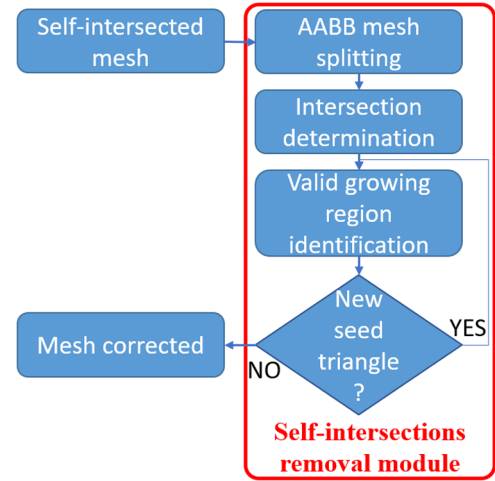


Fig. 5 Self-intersections algorithm flowchart.

a new value of burned propellant in the grain regression module: If the burned grain volume computed at time  $t_0$  differs from burned grain volume computed at time  $t$  less than a fixed tolerance, the code moves to the next time iteration; otherwise, the aforementioned iterative procedure is repeated until convergence [14]. The overall process continues for each simulation time instant until the burnout time is achieved. The ballistics module, as shown in Fig. 2, includes two other models: the thermal protections module and the igniter module. The first estimates the thermal protections ablation phenomenon occurring close to the case and the nozzle throat section erosion, whereas the second evaluates the initial ignition transient phase, before the SRM is completely ignited, which is characterized by igniter behavior.

#### B. Self-Intersections Removal Algorithm

When a cavity is crossed by the burning surface, the regression process can cause the generation of self-intersections, implying a severe error in the propellant burning surface and volume evaluation. These terms highly affect the burned propellant mass per unit time: an erroneous thrust estimation occurs as the final simulation result.

Figure 3 shows the occurrence of self-intersections: the yellow internal cylindrical surface, during surface regression (Figs. 3b and 3c), expands across the spherical cavity.

The consequence is the creation of the red overlapping region (Fig. 3c), which describes the combustion surface in an unrealistic

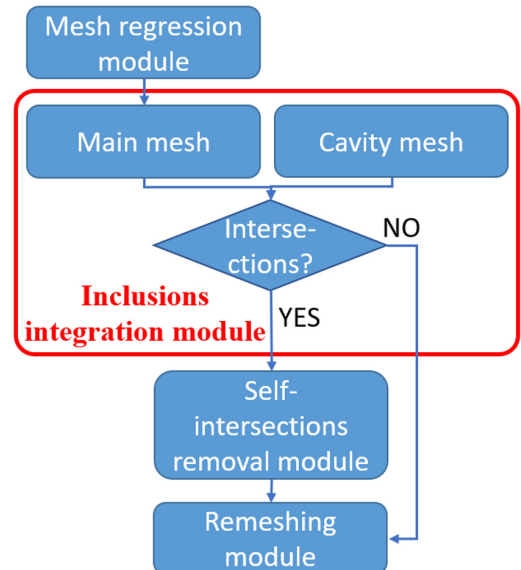


Fig. 6 Cavities' integration flowchart.



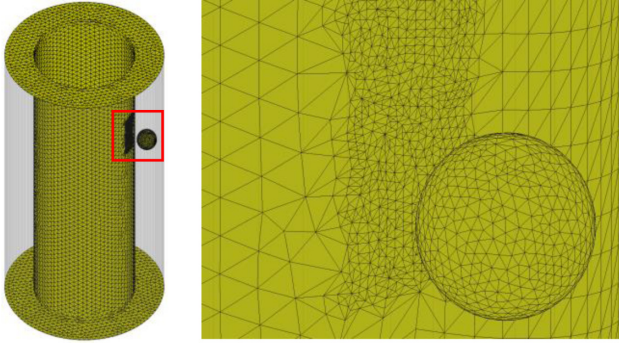


Fig. 7 Local mesh refinement before including the cavity.

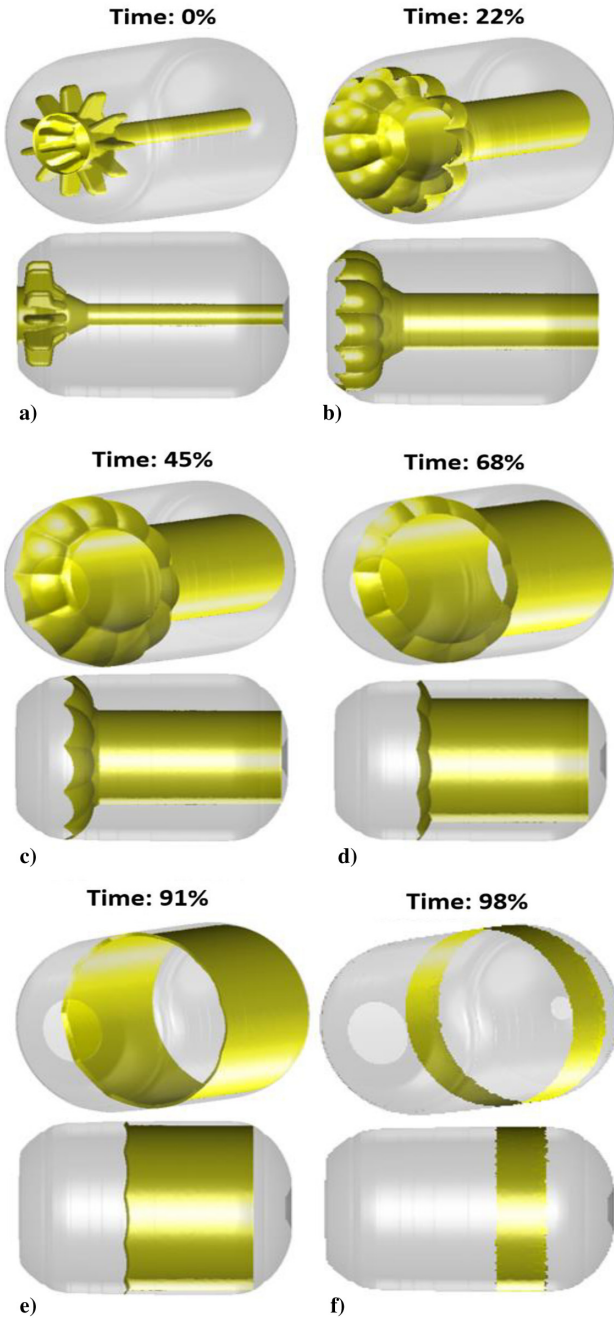


Fig. 8 Burning surface regression in Z9.

way. An efficient self-intersection removal algorithm has been developed and integrated in ROBOOST in order to avoid the aforementioned issue. The basic approach was depicted in detail in Refs. [17,18]: the triangular mesh is divided into three sets of triangles (Fig. 4) [17]:

- 1) The valid triangles are triangles entirely contained in the valid region, which is defined as all triangles that represent the physical grain surface.
- 2) Partially valid triangle are triangles that lie on the boundary between the valid and invalid regions.
- 3) Invalid triangles are triangles within the invalid region, which should be removed entirely.

The algorithm is aimed at identifying the mesh valid region, and it mainly consists of four steps (Fig. 5).

First, the mesh is split into several subsets using an axis-aligned bounding box (AABB) partitioning procedure. The advantage is a reduced computational time required to analyze all mesh triangular elements: only triangles belonging to different subsets are compared together in order to find triangle–triangle intersections. Moreover, triangles subsets share their boundary regions with each other: overlapped buckets ensure a complete intersection detection. Mesh triangle self-intersections are obtained by means of the Moller algorithm [18] based on the intersections of the planes over which the two triangles lie and verifying that the intersection points belong to the triangle domain. Later, a seed triangle (a valid triangle to initiate the whole process) is necessary in order to start the “valid growing region identification” phase. Let  $\Psi$  be the set of vertices on the convex hull of the triangular mesh; any triangle having three vertices in  $\Psi$  can serve as the seed triangle. Triangles in the same region of the seed triangle are identified

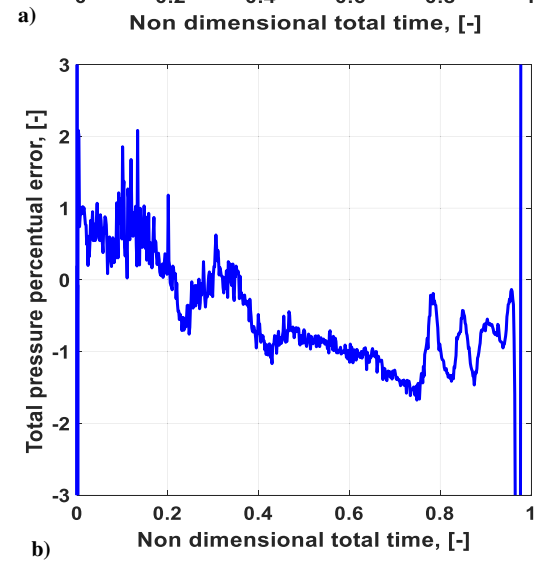
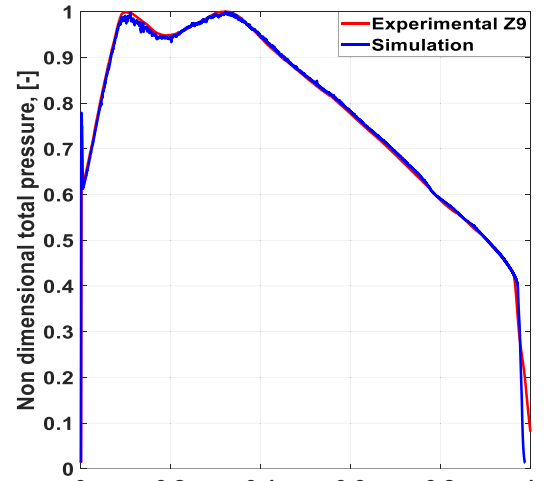


Fig. 9 Combustion chamber pressure comparison.

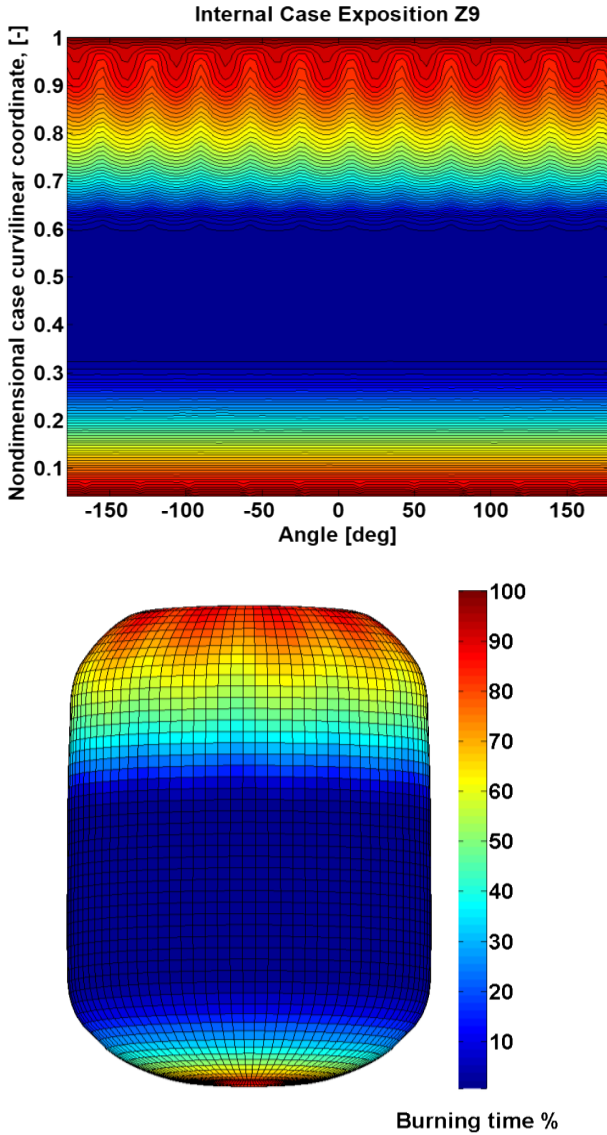


Fig. 10 Z9 thermal protection exposition map.

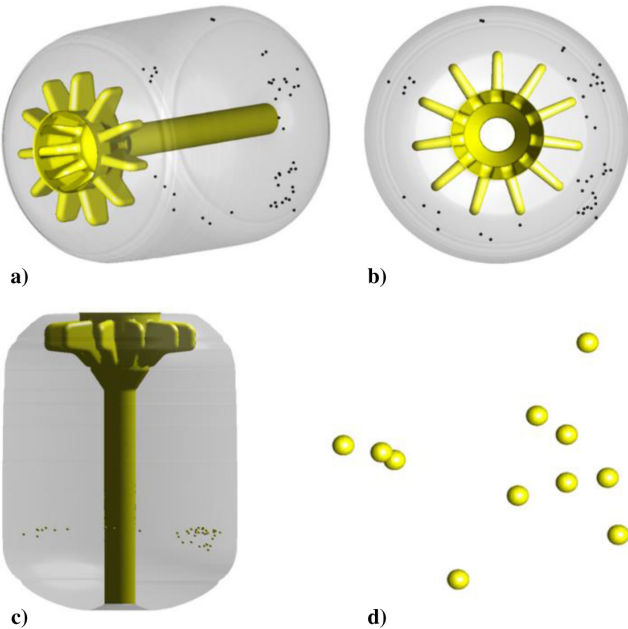


Fig. 11 Z9 with inclusions.

as valid triangles. Now, the valid growing region expands over the whole mesh until invalid triangles are detected. Triangles at the valid region border, sharing at least one edge with the invalid regions, are marked as partially valid triangles. A fundamental part of the explained procedure is a criterion that lets the valid growing region process cross over the self-intersections and move in the right direction to the counterpart triangle (the partially valid triangle at the opposite side of an intersection with respect to the side from which the valid growing region is expanding; line AB in Fig. 4): a compatibility criterion between a triangle normal and its counterpart normal is used. Sometimes, a new seed triangle could be found, especially when a portion of the valid region was surrounded with triangle–triangle intersection points: in this case, the valid growing region identification procedure is

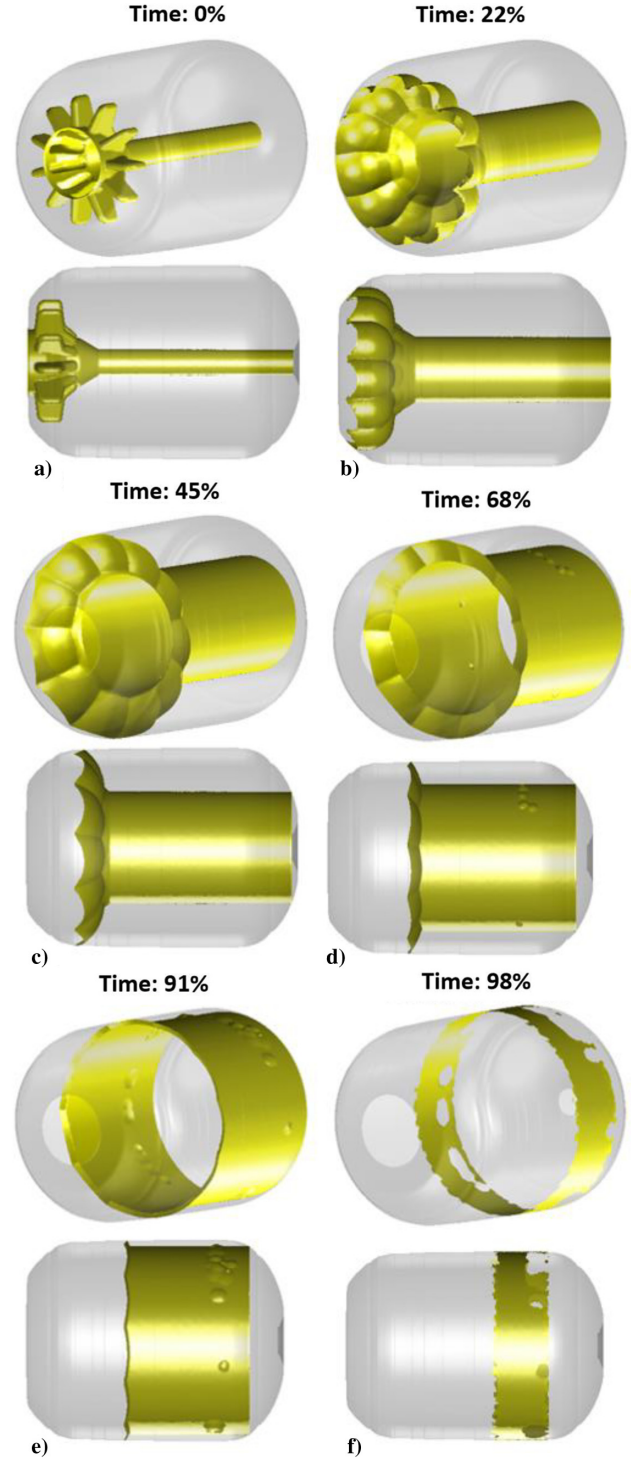


Fig. 12 Burning surface regression in Z9 with inclusions.

run again until new seed triangles are no longer found. The final result is the corrected mesh.

### C. Cavities Integration Procedure

Let the main mesh be the burning surface without any kind of inclusions at a certain temporal iteration of the whole grain regression module process. The cavities' integration algorithm (Fig. 6) essentially consists of verifying if intersections occurs between the main mesh and each cavity mesh.

If there is at least one intersection point, meaning that the propellant has a common region with the cavity volume, then the cavity is included using the self-intersection removal module, which was explained earlier in this paper. On the contrary, if no intersection is found, the main mesh remains unaltered: the burning surface has not approached the cavity surface yet. The previous procedure runs for each cavity: indeed, ROBOOST can handle a generic number of cavities with generic shapes. Furthermore, in order to facilitate cavity integration in the burning surface mesh, a local mesh refinement procedure is performed before cavity sticking (Fig. 7); indeed, intersection points between the cavity and the burning surface mesh are more accurately identified if triangles are of the same size.

## III. Results

As a first step, the ROBOOST simulation software has been applied to the third stage of a Vega launcher, namely, Zefiro 9 (Z9).

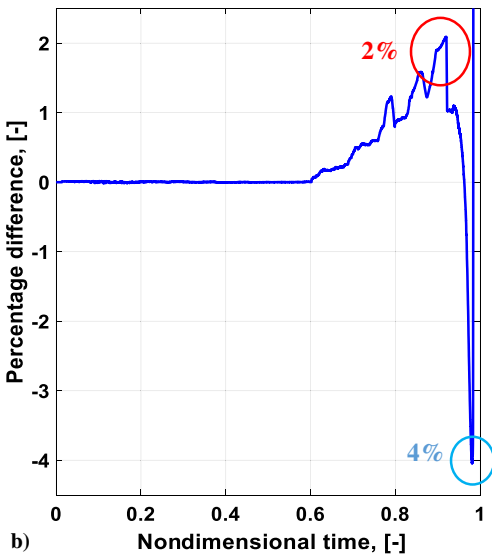
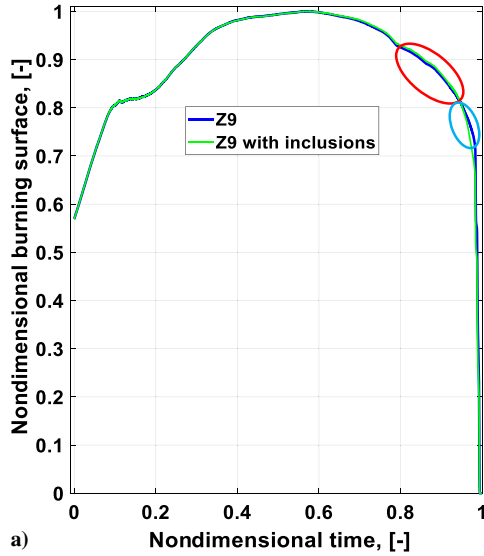


Fig. 13 Burning surface comparison.

Vega is designed to launch small payloads: up to 1500 kg satellites for scientific and Earth observation missions in low Earth orbits. It consists of four stages: three of them are solid propellant based; and the fourth, coupled with the payload fairing, is essentially a liquid propellant engine. Z9 is 3.5 m tall, has a diameter of slightly less than 2 m, weighs 11,500 kg, and burns 10,500 kg of high-density Aluminum (Al), Hydroxyl-terminated-polybutadiene (HTPB), and Ammonium Perchlorate (AP) (Al-HTPB-AP) composite propellant. Its central bore has a circular section in the fore and central parts, as well as a finocyl-shaped configuration in the rear part near the nozzle inlet. The motor solid geometry has been generated using SolidWorks CAD software, and its conversion to triangle surface mesh has been completed with an open-source software that is a free 3-D finite element mesh generator, namely, Gmsh [19]. A variable resolution in the range of 5–30 mm is established for the burning surface: indeed, the smallest triangles are located close to the mesh high curvature regions; instead, the largest ones are centered in low curvature zones. The propellant burning surface regression, obtained with ROBOOST software, is shown in Fig. 8.

A comparison between the simulation's and the experimental motor internal pressures is proposed (Fig. 9a). The red experimental profile of the Z9 firing is assumed as a reference curve.

Contributions provided by the igniter [20], nozzle erosion [15], and the ablation of thermal protections [15] are considered in the aforementioned simulation. The thermal protection contribution is obtained using a semiempirical formulation developed by starting from models explained in Refs. [21,22]. Moreover, an empirical

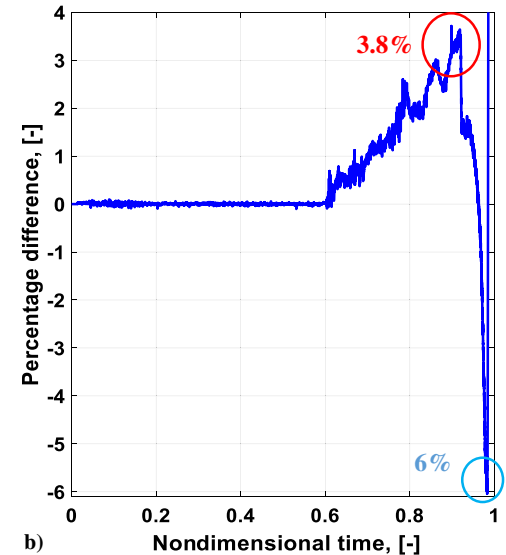
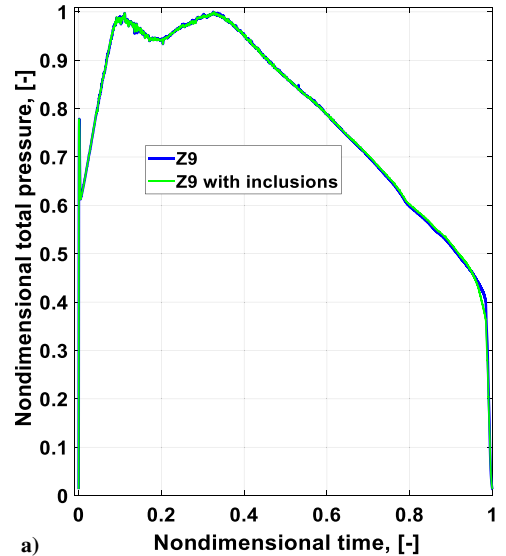


Fig. 14 Combustion chamber pressure comparing.



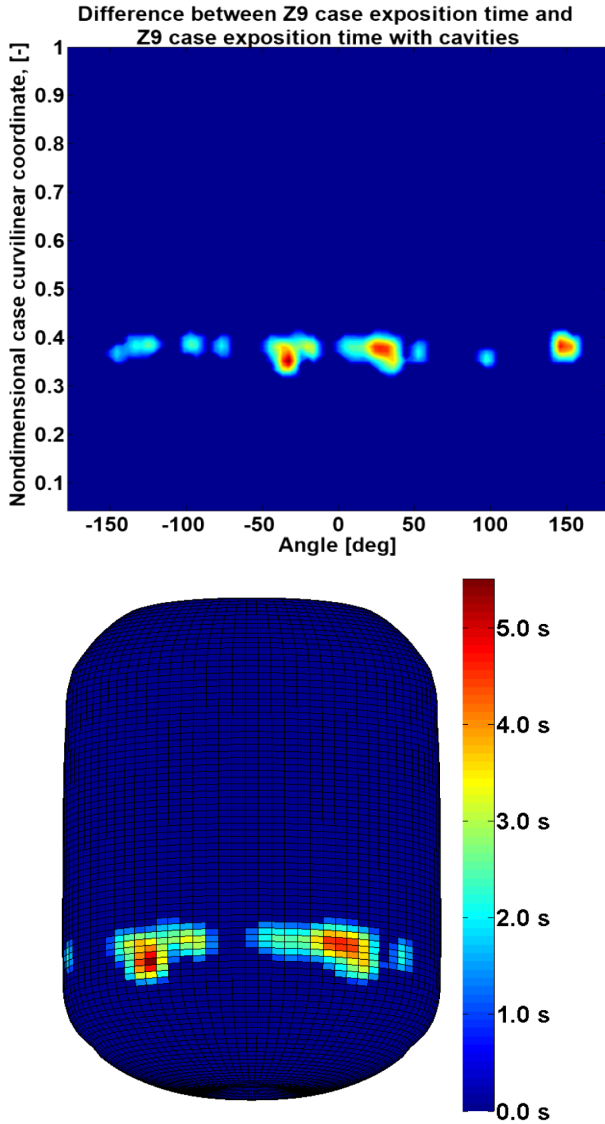


Fig. 15 Thermal protection exposition comparison.

web-dependent hump function [23], acquired from a real firing test, is included in the burn rate computation in order to give a pressure estimation closer to reality. The hump function essentially involves the influence that grain heterogeneities, due to propellant casting process, have on the combustion chamber pressure. The total pressure relative percentage error computed with respect to the experimental data (Fig. 9b) belongs to a variation range of  $\pm 2\%$ , which can be considered satisfactory. Figure 10 shows the thermal protection exposition time along the SRM case.

The color bar represents a color-varying scale linked to the case exposition time percentage (time has been normalized with respect to burnout time). Indeed, the red-orange zones are the first ones to be exposed to the combustion chamber heat; on the other hand, blue regions correspond to lower thermally stressed regions due to an exposition time close to zero. All previously mentioned considerations make the internal case exposition map an effective tool to preliminary design thermal protection thickness for a SRM case.

After validating the code, the following simulation shows the impact of inclusions within the propellant on the Z9 combustion chamber ballistics. The same mesh used in the validation process has also been employed in this simulation. Moreover, a certain spatial distribution of air inclusions is assumed (Figs. 11a–11c). Forty-six spherical-shaped cavities with a diameter of 20 mm are considered in the simulation (Fig. 11d).

Figure 12 shows the burning surface regression results. From a qualitative point of view, it is possible to highlight that, when cavities

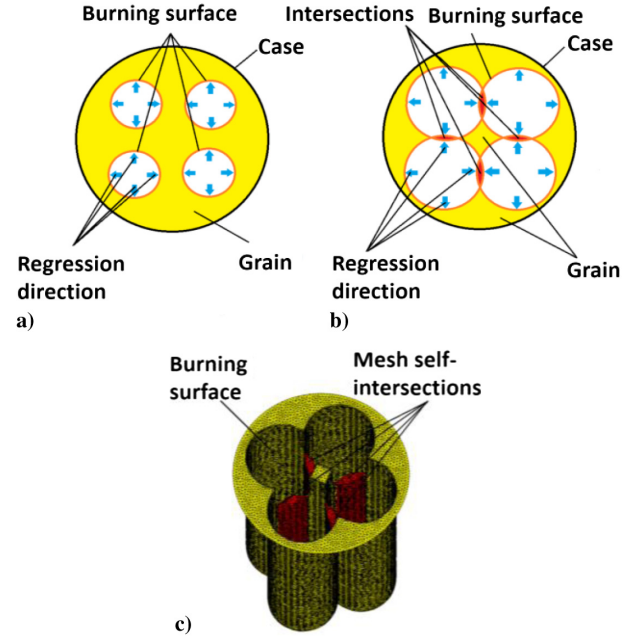


Fig. 16 Self-intersections generation in multiperforated grain.

are included (Figs. 12d–12f), the surface evolution changes with respect to the standard Z9 configuration (Figs. 8d–8f).

The percentual difference (Figs. 13b and 14b) is computed as the relative error with respect to the standard Z9 configuration shown before.

When inclusions are introduced, an increment of the burning surface occurs with a peak of 2% with respect to the standard Z9 simulation (Fig. 13b), implying a combustion chamber pressure increase of 3.8% (Fig. 14b).

However, because of the cavities' surface evolution during the regression phenomenon (Fig. 12e), the burning surface approaches the case earlier; thus, the burning surface decreases faster in time. At 98% of the burnout time, the burning surface of Z9 with cavities is 4% lower than Z9 without them, and the pressure is 6% lower. Hence, the pressure variation due to the simulated cavities' distribution is quite bounded. On the other hand, case exposition shows a more critical behavior.

Figure 15 shows the difference between the Z9 exposition time with cavities and the previous Z9 exposition time map (Fig. 10). The color bar is linked to how much earlier the Z9 case with cavities is exposed to the combustion chamber heat.

A peak of 5 s earlier is reached along the region where most of the cavities are located: this could give the possibility to perform a theoretical check on the capability of the insulation layer to withstand thermal stresses.

Finally, in order to test the reliability and robustness of the self-intersection removal algorithm in challenging conditions, a SRM test geometry was designed (Figs. 16a and 17a). The chosen test geometry is about 1 m long, with a maximum case diameter of 0.8 m. Its main characteristics are related to its central bore: it consists of four tubular shapes, each with an average diameter of 0.1 m and a depth of 0.4 m. As in previous examples, the motor's solid geometry has been generated using SolidWorks CAD software and the mesh with Gmsh software [19]. The generated mesh consists of 32,500 triangles defined with 16,400 mesh nodes with a variable resolution in the range of 1–10 mm, and it is established for the burning surface. This simulation represents an example of the effectiveness and robustness of the self-intersection removal algorithm used: indeed, this geometry reveals many self-intersecting triangles located close to the intersection region of the four tubular structures (red regions in Fig. 16c).



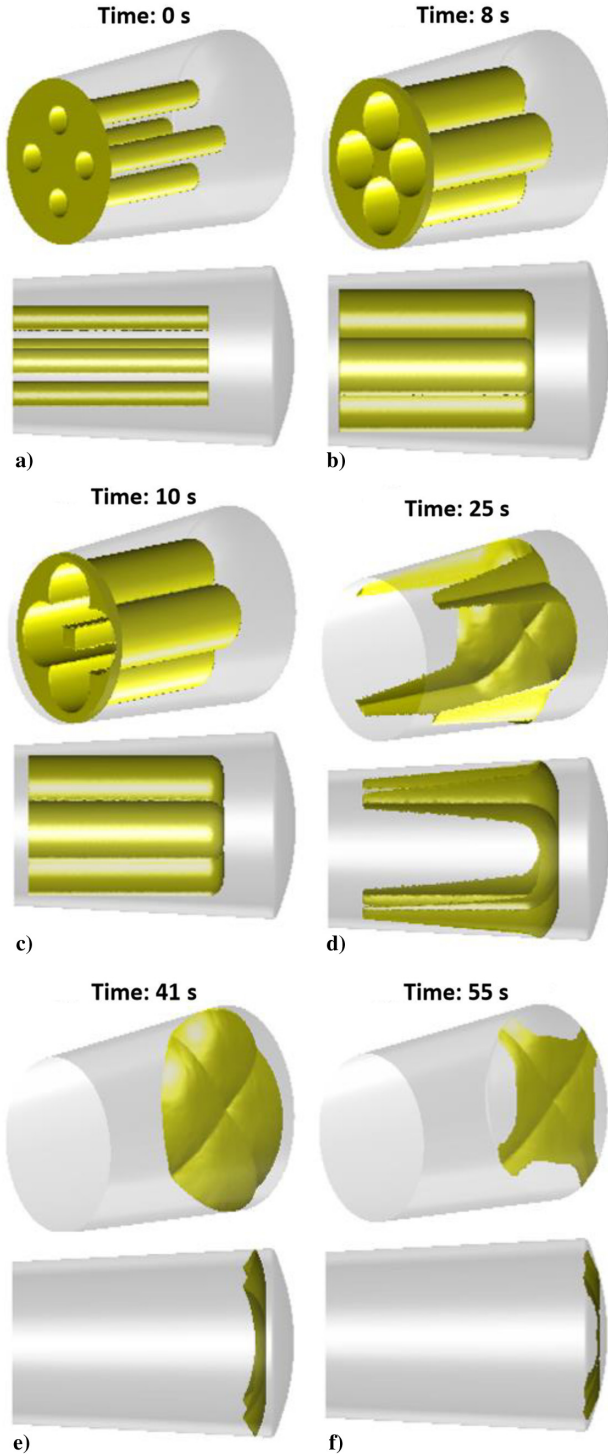


Fig. 17 Burning surface regression in a multiperforated SRM.

Figure 17 shows the spatial evolution of the burning surface in its time regression process. More precisely, all self-intersections are removed with the aforementioned algorithm in the time range between Figs. 17b and 17c.

That step is needed in order to correctly estimate the geometrical parameter's time evolution (Fig. 18). In fact, the self-intersection's occurrence results in an erroneous burning surface increment (Fig. 18a) due to the arising of the red intersecting regions (Fig. 16c), which lead to a wrong volume-time slope increment as well (Fig. 18b).

Furthermore, another interesting result is the internal case exposition map evaluated at the end of combustion (Fig. 19).

The blue zone can be linked to an end-burn combustion type, which takes place during the last time instants (Figs. 17e and 17f).

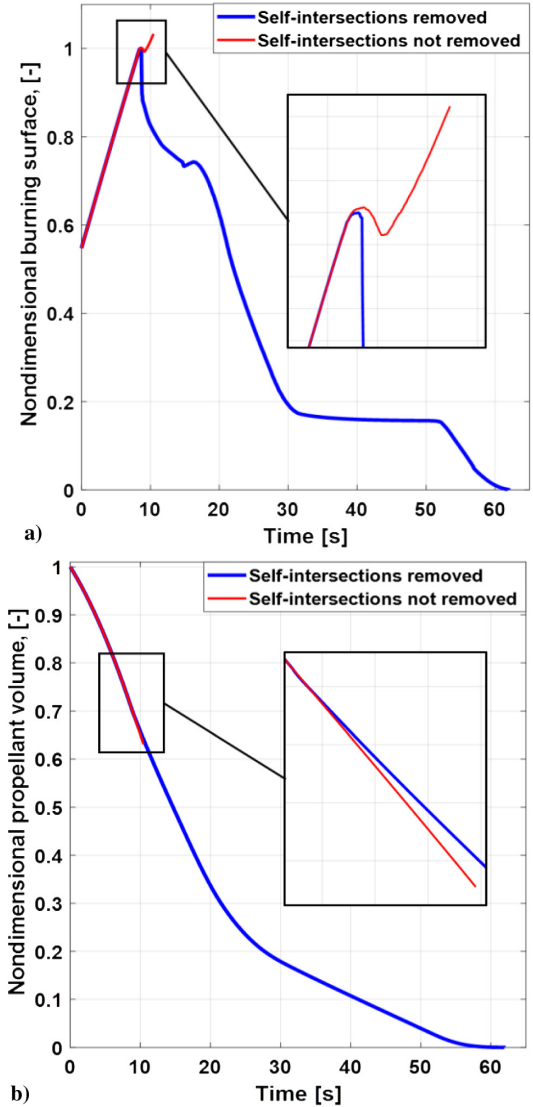


Fig. 18 Burning surface and propellant volume time evolution.

Four orange-to-yellow regions can be identified near the 2-D map horizontal centerline: these are related to the propellant tubular structures intersecting each other (Figs. 17c and 17d), which approach the case earlier than the other part of the burning surface.

All simulations previously discussed have been carried out with the ROBOOST software installed on a calculator with the following features: 16 GB of RAM on an Intel Core i7-7<sup>th</sup> generation CPU machine with 3.10 GHz and a NVIDIA Quadro M1200 graphic card. The two Z9 simulations (with and without cavities) lasted 2 h; instead, the multiperforated propellant geometry lasted 4 h due to the larger number of self-intersections to fix.

#### IV. Conclusions

An original computer graphics procedure has been developed and integrated in ROBOOST. By means of a self-intersection removal algorithm, the code can produce estimations regarding both pressure variations due to grain cavities and the case time exposure of a generically shaped solid rocket motor. The simulator provided with the aforementioned algorithm has been validated with the Z9 SRM, comparing simulated results with experimental data. A quite appreciable error in the range of  $\pm 2\%$  is obtained. The validated approach proved to be effective in evaluating the impact of a cavities' distribution on combustion chamber pressure and thermal protection exposure time variations.

Future developments will be focused on evaluating the grain debonding phenomenon.

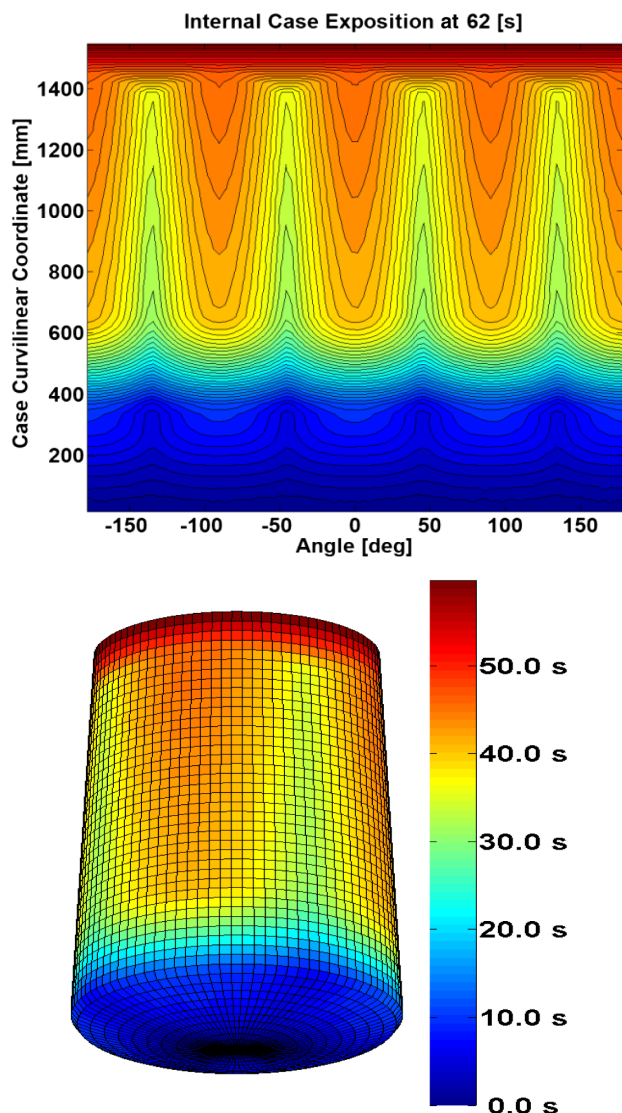


Fig. 19 Multiperforated SRM thermal protection exposition.

### Acknowledgments

The authors would like to thank the AVIO S.p.A. Space Division for supplying us with the necessary data, as well as the Laboratory of Propulsion and Mechanics of the University of Bologna.

### References

- [1] Kallmeyer, T., and Sayer, L. H., "Differences Between Actual and Predicted Pressure-Time Histories of Solid Rocket Motors," AIAA Paper 1982-1094, June 1982.  
<https://doi.org/10.2514/6.1982-1094>
- [2] Sutton, G. P., and Biblarz, O., "Rocket Propulsion Elements," *Rocket Testing*, 7th ed., Wiley, New York, 2010, pp. 711–726.
- [3] Hessler, R. O., and Glick, R. L., "Consistent Definitions for Burning-Rate Measurements in Solid Rocket Motors," *Combustion Explosion and Shock Waves*, Vol. 36, No. 1, 2000, pp. 83–106.  
<https://doi.org/10.1007/BF02701517>
- [4] Majdalani, J., and Van Moorhem, W. K., "Improved Time-Dependent Flowfield Solution for Solid Rocket Motors," *AIAA Journal*, Vol. 36, No. 2, 1998, pp. 241–248.  
<https://doi.org/10.2514/2.7507>
- [5] Sotter, G., and Swithenbank, J., "Vortices in Solid Propellant Rocket Motors," *AIAA Journal*, Vol. 1, No. 7, 1963, pp. 1682–1684.  
<https://doi.org/10.2514/3.1891>
- [6] Kordig, J. W., and Fuller, G. H., "Correlation of Nozzle Submergence Losses in Solid Rocket Motors," *AIAA Journal*, Vol. 5, No. 1, 1967, pp. 175–177.  
<https://doi.org/10.2514/3.3933>
- [7] Hill, P. G., and Peterson, C. R., "Mechanics and Thermodynamics of Propulsion," *Solid-Propellant Combustion Chambers*, 2nd ed., Addison Wesley, New York, 1992, pp. 598–600.
- [8] Dunn, S. S., and Coats, D. E., "3-D Grain Design and Ballistic Analysis Using the SPP97 Code," AIAA Paper 1997-3340, July 1997.  
<https://doi.org/10.2514/6.1997-3340>
- [9] Dauch, F., and Ribéreau, D., "A Software for SRM Grain Design and Internal Ballistics Evaluation, PIBAL (Propulsion and Internal Ballistics)," AIAA Paper 2002-4299, July 2002.  
<https://doi.org/10.2514/6.2002-4299>
- [10] Osher, S., and Sethian, J. A., "Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations," *Journal of Computational Physics*, Vol. 79, No. 1, 1988, pp. 12–49.  
[https://doi.org/10.1016/0021-9991\(88\)90002-2](https://doi.org/10.1016/0021-9991(88)90002-2)
- [11] Gibou, F., Fedkiw, R., and Osher, S., "A Review of Level-Set Methods and Some Recent Applications," *Journal of Computational Physics*, Vol. 353, No. 15, 2018, pp. 82–109.  
<https://doi.org/10.1016/j.jcp.2017.10.006>
- [12] Willcox, M. A., Brewster, M. Q., Tang, K. C., Stewart, D. S., and Kuznetsov, I., "Solid Rocket Motor Internal Ballistics Simulation Using Three-Dimensional Grain Burnback," *Journal of Propulsion and Power*, Vol. 23, No. 3, 2007, pp. 575–584.  
<https://doi.org/10.2514/1.22971>
- [13] Willcox, M. A., Brewster, M. Q., Tang, K. C., and Stewart, D. S., "Solid Propellant Grain Design and Burnback Simulation Using a Minimum Distance Function," *Journal of Propulsion and Power*, Vol. 23, No. 2, 2007, pp. 465–475.  
<https://doi.org/10.2514/1.22937>
- [14] Bertacin, R., Ponti, F., Corti, E., Fedele, D., and Annovazzi, A., "Numerical Simulation of the Zefiro 9 Performance Using a New Dynamic SRM Ballistic Simulator," AIAA Paper 2013-4174, July 2013.  
<https://doi.org/10.2514/6.2013-4174>
- [15] Bertacin, R., Ponti, F., and Annovazzi, A., "A New Three-Dimensional Ballistic Model for Solid Rocket Motor Non-Homogeneous Combustion," AIAA Paper 2012-3974, Aug. 2012.  
<https://doi.org/10.2514/6.2012-3974>
- [16] Maggi, F., De Luca, L. T., Bandera, A., Subith, V. S., and Annovazzi, A., "Burning Rate Measurements on Small-Scale Solid Rocket Motors," *Defense Science Journal*, Vol. 56, No. 3, 2006, pp. 353–367.  
<https://doi.org/10.14429/dsj>
- [17] Jung, W., Shin, H., and Choi, B., "Self-Intersection Removal in Triangular Mesh Offsetting," *Computer-Aided Design and Applications*, Vol. 1, No. 1, 2004, pp. 477–484.  
<https://doi.org/10.1080/16864360.2004.10738290>
- [18] Moller, T., "A Fast Triangle-Triangle Intersection Test," *Journal of Graphic Tools*, Vol. 2, No. 2, 1997, pp. 25–30.  
<https://doi.org/10.1080/10867651.1997.10487472>
- [19] Geuzaine, C., and Remacle, J. F., "Gmsh: A Three-Dimensional Finite Element Mesh Generator with Built-In Pre- and Post-Processing Facilities," *International Journal for Numerical Methods in Engineering*, Vol. 79, No. 11, 2009, pp. 1309–1331.  
<https://doi.org/10.1002/nme.v79:11>
- [20] Ponti, F., Souhair, N., Mini, S., and Annovazzi, A., "0D Unsteady—1-D Quasi-Stationary Internal Ballistic Coupling for ROBOOST Simulation Tool," AIAA Paper 2019-4140, Aug. 2019.  
<https://doi.org/10.2514/6.2019-4140>
- [21] Schoner, R. J., "User's Manual Aerotherm Charring Material Thermal Response and Ablation Program," U.S. Air Force Rocket Propulsion Lab., AFRPL-TR-70-92, Wright-Patterson AFB, OH, April 1970.
- [22] Moyer, C. B., and Rindal, R. A., "An Analysis of the Coupled Chemically Reacting Boundary Layer and Charring Ablator. Part 2- Finite Difference Solution for the In-Depth Response of Charring Materials Considering Surface Chemical and Energy Balances," NASA CR-1061, June 1968.
- [23] Viganò, D., Annovazzi, A., and Maggi, F., "Monte Carlo Uncertainty Quantification Using Quasi-1-D SRM Ballistic Model," *International Journal of Aerospace Engineering*, Vol. 2016, April 2016, Paper 3765796.  
<https://doi.org/10.1155/2016/3765796>