



**Politecnico
di Torino**

ScuDo

Scuola di Dottorato - Doctoral School
WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation

Doctoral Program in Aerospace Engineering (38th cycle)

Numerical Simulation and Inverse Design of Solid Rocket Motors with Non Uniform Ballistic Properties

By

Giovanni Polizzi

Supervisor(s):

Prof. D., Pastrone

Prof. A., Ferrero

Prof. F., Masseni

Doctoral Examination Committee:

Prof. D. Bianchi, Referee, La Sapienza

Prof. C. Paravan, Referee, Politecnico di Milano

Politecnico di Torino

2026

Declaration

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Giovanni Polizzi
2026

* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

I would like to dedicate this thesis to my family

Acknowledgements

As I reach the end of this journey, I would like to acknowledge those who contributed to this work and supported me throughout these years.

I am deeply grateful to Prof. Dario Pastrone, Prof. Andrea Ferrero, and Prof. Filippo Masseni for their guidance, patience, and trust throughout my doctoral studies. Working alongside them has been a pleasure, and their valuable advice and support were essential during the most critical phases of this research.

I would like to thank my fellow PhD students for sharing the challenges of this journey. The daily exchange with you made this path far more rewarding.

A special thought goes to my family, for their unconditional support and for believing in me even in moments of doubt.

Finally, I am grateful to my long-time friends for being there through it all, whether near or far.

To all of you, thank you.

Abstract

The design of Solid Rocket Motor (SRM) grains has traditionally been constrained by manufacturing limitations and by the assumption of spatially uniform ballistic properties or discrete variations of the composition, as in dual-thrust grains. Recent advances in additive manufacturing technologies have introduced new opportunities for the realization of complex three-dimensional grain geometries and locally tailored material characteristics, enabling novel thrust modulation leverage. In addition to increased design flexibility, additive manufacturing may also offer potential advantages at process level, such as reductions in manufacturing complexity and cost, improved repeatability, and a mitigation of risks related to the handling of toxic or hazardous propellant additives. These developments require advanced numerical tools capable of accurately simulating grain regression and internal ballistics in the presence of non-uniform burning behavior. This thesis presents a numerical simulation and inverse design framework for the analysis and optimization of solid rocket motors with complex grain geometries and spatially varying burning rates. The proposed approach is based on the Level Set Methods to implicitly represent the evolving burning surface during grain regression, providing a flexible and robust description of arbitrary three-dimensional configurations. The instantaneous geometry representation is coupled with a zero-dimensional chamber formulation to predict pressure–time histories under non-uniform ballistic conditions. Different strategies for distance field initialization and surface evolution are investigated. To enable efficient inverse design and reduce computational costs, surrogate modeling techniques are introduced to approximate the motor pressure response as a function of geometric and ballistic parameters. These surrogates are integrated within stochastic optimization algorithms, allowing the exploration of large design spaces that would otherwise be computationally prohibitive. A linear viscoelastic mechanical model is included to validate the structural performance of the optimized grain and to compare classical and non-uniform grain configurations. The proposed methodology

is validated through a set of test cases, ranging from canonical grain configurations to motors characterized by spatially varying burning rates. The framework is further applied to real life application, demonstrating that different combinations of grain geometry and ballistic distributions can achieve comparable performance levels when appropriate design flexibility is introduced. Overall, the results highlight the potential of combining additive manufacturing technologies with advanced numerical simulation and inverse design techniques to expand the performance envelope of solid rocket motors. By enabling controlled thrust shaping and performance tailoring through geometric and material design, the proposed framework provides a solid foundation for future developments in flexible, high-performance solid propulsion systems.

Contents

List of Figures	ix
List of Tables	xv
List of Symbols and Acronyms	xvi
1 Introduction	1
1.1 Overview on Solid Rocket Motors	1
1.2 Potential impact of Additive Manufacturing for SRM	5
1.3 Overview of the present work	7
2 Simulation Tools	10
2.1 Level Set Method	11
2.1.1 Initialization via SDF computation	16
2.1.2 Initialization via Heat Method	18
2.1.3 Finite Difference on Uniform Grids	29
2.1.4 Reintialization procedure	39
2.1.5 Finite Volume on Octree grids	40
2.2 0D Chamber Model	47
2.3 Throat Erosion Model	50
2.4 Linear Viscoelatic Model	51

3	Initial Testing	55
3.1	Area computation error	56
3.2	Surface transport test, star shaped grain	59
3.3	Test on experimental firing tests	64
3.4	Test with non-uniform ballistic	66
4	SRM Design	72
4.1	Design approach and modifications	73
4.2	Surrogate modelling	77
4.3	Optimization approach	81
4.4	Test application	83
4.4.1	Polynomial distribution - Finocyl geometry	83
4.4.2	RBF distribution - Harmonic and Finocyl geometry	101
4.5	Surrogate applicability	112
5	Application	114
5.1	Zefiro 23-like geometry	114
6	Conclusions	125
	References	128
	Appendix A	137

List of Figures

1.1	Space Launch System close up at Kennedy Space Center. SLS Solid Rocket Booster derived from Space Shuttle Boosters are one of the prominent and latest application of SRM in the space launcher field (Credit to NASA/Sam Lott)	3
1.2	Simplified perspective three-quarter section view of a typical solid propellant rocket motor with the propellant grain (orange), nozzle (yellow), igniter (red) and case with insulation layer (blue).	5
1.3	Example from literature of materials study on new additive manufacturing techniques for energetic materials	7
2.1	Example representation of 2D domain with a zero level interface that divides the domain	12
2.2	Schematic representation of 2D domain with implicit function and its zero level intersection. Here the contour merging is visible	12
2.3	Visualization of finocyl geometry inner cavity triangulation from which SDF is computed and burning surface implicitly reconstructed	18
2.4	Visualization of intermediate step of heat method. Figure from [1] .	21
2.5	Visualization of intermediate step of signed heat method. M is a generic manifold, in this work simply an $\mathbb{R}^2$ or $\mathbb{R}^3$ domain	23
2.6	u_0 source diffused for a circle	24
2.7	Distance field computed with Signed Heat Method (circle)	24
2.8	u_0 source with pseudo normals for Signed Heat Method on a complex 2D shape	25

2.9	Distance field computation for a complex 2D shape	25
2.10	Gradient visualization detail for a complex 2D shape	26
2.11	u_0 source diffused for a sphere	26
2.12	Distance field computed with Signed Heat Method (sphere)	27
2.13	u_0 source with pseudo normals for Signed Heat Method on a 3D finocyl shape	27
2.14	Distance field computation for a 3D finocyl shape	28
2.15	Sphere zero level initialized with Signed Heat Method on an adaptive octree grid	29
2.16	Finocyl zero level initialized with Signed Heat Method on an adap- tive octree grid	29
2.17	Three dimensional stencil used to compute discrete operators on cartesian grids	30
2.18	Sparsity pattern of the Laplacian matrix L implementing the 7-point stencil on a uniform Cartesian grid. The matrix has value 6 on the main diagonal (black) and -1 on the six off-diagonals corresponding to the neighbors in x , y , and z directions.	31
2.19	Sparsity patterns of directional derivative matrices on a uniform Cartesian grid. Each matrix implements centered finite differences along one coordinate direction.	32
2.20	Representation of hierarchical octree structure and Z-ordering. Every active level has a different color. Note the ordering reading the tree from left to right.	41
2.21	2D scheme of diamond stencil	42
2.22	3D scheme of diamond stencil	43
2.23	Visualization of quadtree 2D grid adaptation. On octree grids the adaptation is performed on eight cell.	47
3.1	Visualization of test sphere for cartesian grid	56
3.2	Visualization of test sphere for octree grid	57

3.3	Area computation maximum error for sphere, computed with TVDRK-ENO at second order	57
3.4	Area computation maximum error for sphere, computed with Finite Volume on adaptive octree grids	58
3.5	Burnback simulation of star-shaped grain after 1 second	60
3.6	Mean error over normal transport for different schemes varying grid dimension	61
3.7	Area computation mean error for star shaped grain burnback, computed with TVDRK-ENO at second order	62
3.8	Area computation mean error for star shaped grain burnback, computed with TVDRK-WENO at fifth order in space and third order on time	62
3.9	Pressure evaluation of the star shaped test case	64
3.10	Eight fins inhibited front-end finocyl geometry used for testing. The figure shows the inner cavity of the grain at $t = 0$	65
3.11	Results of burnback simulation of finocyl geometries. Comparison with experimental and numerical references, for both front burning cases	66
3.12	Radial distribution of pre-exponential coefficient a as a function of b . Coefficient $a_0 = 7.25e - 05$ and $a_c = 4e - 05$	67
3.13	Visualization of ballistic properties distribution during burnback of front burning finocyl	68
3.14	Modified pressure time performance for non-uniform finocyl grain	69
3.15	Pressure Time for cylindrical shape modified for neutral burning	70
3.16	Pressure time of a cylindrical shape modified for dual thrust, between 100 bar and 50 bar	70
4.1	Solid rocket motor design flow	73
4.2	Design flow proposed	77
4.3	General Geometry and Grain Geometric Parameters	85

4.4	Example configurations for testing geometry	85
4.5	Free Length Dataset	88
4.6	Fixed Length Dataset	88
4.7	Loss Function over training epochs	89
4.8	Neural Network Prediction on some test samples	90
4.9	Global Residual Plot for Neural Network	90
4.10	ABFC Prediction on some test samples	90
4.11	Global Residual Plot for ABFC	91
4.12	POD-RBF Prediction on some test samples	91
4.13	Global Residual Plot for POD-RBF	91
4.14	Target curves vs free length dataset	93
4.15	Target curves vs fixed length dataset	93
4.16	Optimum visualization for target curves	95
4.17	Target B2 optimal configuration, grain volume and casing	100
4.18	Temperature distribution inside the grain and casing at the end of temperature cycle	100
4.19	Von Mises stress inside the grain at the end of pressurization ramp .	101
4.20	CAD visualization of 3D grain example, $M = 5$	103
4.21	Example of port configuration	104
4.22	Ballistic distribution example. Values obtained using parameters in- side the RBF-Port design envelope. Scalar field refers to a parameter field	104
4.23	Dataset for RBF-harmonic	106
4.24	Dataset for RBF-Fin	107
4.25	Optimum visualization for target curves RBF	108
4.26	RBF-HAR optimum B5 geometry of grain and casing	110
4.27	RBF-HAR optimum B5 temperature distribution in grain and casing	111

4.28	RBF-HAR optimum B5 stress distribution in grain	111
5.1	Geometry comparison. Different grain geometries can achieve comparable performance levels by suitably altering the ballistic distribution.	116
5.2	Zefiro23-like target pressure–time curve (black) compared against the RBF-HAR dataset	116
5.3	Zefiro23-like target pressure–time curve (black) compared against the RBF-FIN dataset	117
5.4	Zefiro23-like target pressure–time curve (black) compared against the RBF-CYL dataset	117
5.5	Optima on Zefiro23-like target. Exploiting ballistic distribution leads to convergent result for different geometries	120
5.6	Zefiro23-like geometry utilized for structural comparison. Note that the section plane is tilted to align with fin axis and show the cavity. .	121
5.7	RBF-HAR optimum geometry utilized for structural comparison . .	121
5.8	Zefiro geometry temperature distribution	122
5.9	RBF-HAR optimum geometry temperature distribution	122
5.10	Zefiro geometry stress distribution	124
5.11	RBF-HAR optimum geometry stress distribution	124
A.1	Visualization of Finocyl geometry with polynomial ballistic distribution free length case, comparison for random samples	137
A.2	Pressure time visualization from dataset for respective Finocyl configuration, free length case	138
A.3	Visualization of Finocyl geometry with polynomial ballistic distribution fixed length case, comparison for random samples	138
A.4	Pressure time visualization from dataset for respective Finocyl configuration, fixed length case	139
A.5	Visualization of Finocyl geometry with polynomial ballistic distribution RBF case, comparison for random samples	139

A.6	Pressure time visualization from dataset for respective Finocyl configuration, RBF case	140
A.7	Visualization of Harmonic geometry with polynomial ballistic distribution RBF case, comparison for random samples	140
A.8	Pressure time visualization from dataset for respective Harmonic configuration, RBF case	141
A.9	Geometries of optima obtained to match Zefiro23 pressure time curve, Fin and Harmonic baseline configurations	141

List of Tables

4.1	Parameters Table Free Length	86
4.2	Parameters Table Fixed Length	86
4.3	Median RMSE on test dataset	92
4.4	Requirements computed for test curves	94
4.5	Comparison of Performance. Application on two configuration types for 3000 samples and 4096 samples databases (* if constraints violation)	96
4.6	Mechanical and thermal properties of materials	99
4.7	Variation of solid propellant's specific heat with temperature	99
4.8	Prony series coefficients of HTPB-based propellant	99
4.9	WLF function coefficients of HTPB-based propellant	100
4.10	Parameters Table RBF-Port	105
4.11	Comparison of Performance for RBF Harmonic and Fin. (* indicates constraint violation)	109
5.1	Requirements computed for Zefiro23-like pressure curve. Note that A_t is now a design parameter	116
5.2	Parameters Table RBF-Cylinder	119
5.3	Comparison of Performance on Zefiro23-like Optimization	119

List of Symbols and Acronyms

α	Numerical diffusion coefficient / CFL coefficient / Thermal expansion coefficient
β	Coefficient in pressure integration scheme
δ	Dirac delta function
Δ	Laplacian operator
Δt	Time step
Δx	Spatial step in x direction
ε	Grain shape parameter / smoothing bandwidth / deformation
ε	Nozzle expansion ratio
γ	Specific heat ratio
Γ	Critical mass flow function
λ	Reduced time constant
Λ	Integrated ballistic parameter
ω	Weight in WENO scheme
Ω	Domain
ϕ	Implicit level set function
Φ	Generic scalar field
ρ	Density
ρ_b	Propellant density
ρ_c	Chamber gas density
σ	RBF shape parameter
τ	Time variable in convolution integral
θ	Characteristic angle / angular coordinate
A	Area / coefficient in harmonic function
A_b	Burning area
A_t	Throat area

A_{th}	Throat area
B	Coefficient in harmonic function
C_1, C_2	WLF shifting coefficients
C_F	Thrust coefficient
D	Grain diameter
D^n	Divided difference of order n
$E(t)$	Relaxation modulus
E_∞	Asymptotic relaxation modulus
F	Thrust
G	Godunov function
I_{sp}	Specific impulse
I_{tot}	Total impulse
K_t	Heat kernel
L	Grain length / Laplacian matrix
M	Mach number / number of harmonics
N	Number of grid points / number of basis functions
P	Polynomial / perimeter
P_2	Polynomial coefficients vector
Q	Polynomial interpolation
R	Radial coordinate / pressure derivative term
R_g	Specific gas constant
R_{th}	Throat radius
S	Surface / smoothness indicator / smoothed sign function
T	Temperature
T_0	Reference temperature
T_c	Chamber temperature
V	Volume / velocity
V_c	Chamber volume
X	Design parameter vector
$\dot{m}$	Mass flow rate
$\dot{m}_b$	Burning mass flow rate
$\dot{m}_{th}$	Throat mass flow rate
$\dot{s}$	Throat erosion rate
a	Pre-exponential coefficient in burning rate law
a_0	Pre-exponential coefficient at motor axis

a_c	Pre-exponential coefficient at casing
a_T	Time-temperature shift factor
b	Exponent in ballistic distribution
c	Coefficient
c^*	Characteristic velocity
c_0	Reference characteristic velocity coefficient
d	Distance
dr	Radial distance increment
dt	Time step
f	Function
g	Constraint function
h	Grid spacing
i, j, k	Grid indices
l	Characteristic length
m	Pressure exponent for c^*
n	Pressure exponent in burning rate law
p	Pressure / polynomial order
p_c	Chamber pressure
p_e	Exit pressure
p_{amb}	Ambient pressure
p_{eq}	Equilibrium pressure
r	Burning rate / radial coordinate
r_c	RBF center radial coordinate
t	Time
t_b	Burning time
u	Heat diffusion variable
v	Velocity component / vertex value
w	Weight
w_f	Web fraction
x, y, z	Cartesian coordinates
y_{split}	Axial coordinate where function splits
$\nabla \cdot$	Divergence operator
∇	Gradient operator
∂	Partial derivative
$(\cdot)_0$	Initial value

$(\cdot)_b$	Burning / propellant
$(\cdot)_c$	Chamber
$(\cdot)_f$	Face
$(\cdot)_g$	Guessed / computed value
$(\cdot)_i$	Index i / cell i
$(\cdot)_n$	n-th port
$(\cdot)_{eq}$	Equilibrium value
$(\cdot)_{req}$	Required value
$(\cdot)_{th}$	Throat
$(\cdot)_{x,y,z}$	x,y,z-component / derivative
$(\cdot)^*$	Optimal value
$(\cdot)^+$	Right value
$(\cdot)^-$	Left value
$(\cdot)^n$	Time integration level n
2D	Two-Dimensional
3D	Three-Dimensional
ABFC	Adaptive Basis Function Construction
AP	Ammonium Perchlorate
CAD	Computer-Aided Design
CFL	Courant-Friedrichs-Lewy condition
DB	Double Based propellant
DDFV	Discrete Duality Finite Volume
ENO	Essentially Non-Oscillatory
FEM	Finite Element Method
FV	Finite Volume
HJ	Hamilton-Jacobi
HMX	Cyclotetramethylene-tetranitramine
HRE	Hybrid Rocket Engine
HTPB	Hydroxyl-Terminated Polybutadiene
LRE	Liquid Rocket Engine
LS	Level Set
LSM	Level Set Method
MEOP	Maximum Expected Operating Pressure
NC	Nitrocellulose
NG	Nitroglycerin

NN	Neural Network
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
POD	Proper Orthogonal Decomposition
PSO	Particle Swarm Optimization
RBF	Radial Basis Function
RDX	Research Department eXplosive, cyclotrimethylene trinitramine
RMSE	Root Mean Square Error
ROM	Reduced Order Model
SDF	Signed Distance Function
SRB	Solid Rocket Booster
SRM	Solid Rocket Motor
STL	STereoLithography file format
TVD-RK	Total Variation Diminishing Runge-Kutta
WENO	Weighted Essentially Non-Oscillatory
WLF	Williams-Landel-Ferry

Chapter 1

Introduction

1.1 Overview on Solid Rocket Motors

Since the late 19th century, humanity has started to learn to exploit the energy stored in the chemical bonds to propel man-made objects through the atmosphere and above. Chemical propelled rockets convert this energy, obtained by the exothermic combustion reaction between oxidizer and fuel molecules, into kinetic energy, through an expansion in a propulsive nozzle. The high velocity exhaust gases generate thrust via action-reaction principle, allowing the rocket to accelerate. The earliest applications of solid propulsion can be traced back to ancient China, where firework-like devices and rudimentary military rockets exploited the same basic physical principles. Nevertheless, only the advent of modern chemistry, industrial manufacturing and the development of fundamental rocket theory by prominent figures like Tsiolkovsky, Oberth and Goddard, enabled the foundations of modern rocketry applications. Subsequent developments in propellant chemistry, structural materials, and motor configuration, particularly during the mid-20th century with the wartime development and the space race, enabled the rise of powerful and reliable rockets, which remain essential components in modern launch vehicles and defense systems. Propellants can be stored both in liquid and solid phase. Accordingly, chemical rockets can be classified in Liquid Rocket Engines (LREs), which utilize both oxidizer and fuel in a liquid state, Solid Rocket Motors (SRMs) which stores the propellant components in a solid compound, while Hybrid Rocket Engines (HREs) exploit a combination of liquid oxidizer and a solid fuel grain.

SRMs store propellant in a solid cartridge called grain, inside the motor thrust chamber, within the rocket case. Once the reaction is ignited thanks to an external energy source, the inner surface of the grain is burned and it regresses at a stable rate, producing hot gases in the process. These gases build up chamber internal pressure, and mass flow is accelerated through the grain port, i.e. the inner section of the grain, up to the motor nozzle exit section. Grain burning proceeds until all propellant inside the case is depleted. SRMs are inherently simple, since they do not require all auxiliary systems for fluid control like LREs and HREs. They also offer unmatched storage features and are ready-to-use for an extended time period, exceeding 20 years in some applications. They can provide high thrust levels and have the best impulse to volume ratio, thanks to higher density of the solid propellant, which limits the amount of volume required for the same required total impulse. For these features, SRMs are the unchallenged choice for defense application when fast response, long storage, reliability and compactness are crucial. In this field, SRMs are utilized in tactical and strategic missiles, anti-aircraft systems, anti-tank rockets and multiple-launch rocket systems. At the same time, thanks to their high thrust level, SRMs have been widely used both as boosters to enhance lift capability of space launchers first stages and as first stages for smaller launchers. Notable applications include the Space Shuttle Solid Rocket Boosters, Space Launch System solid rocket boosters (depicted in Figure 1.1), Ariane 5 P230, Vega P80 and P120C motors, as well as smaller launchers like Pegasus and Minotaur. Moreover, they have been used in other space related specific application, such as Orbital Transfer Vehicles, even if this application is now uncommon in commercial spaceflight. SRMs have further been used to provide gas generator function, for starting liquid engines or tank pressurization. Gas generator function is also provided in other civilian applications, such as airbag inflation and fire extinguisher devices. On the other hand, SRMs provide a lower specific impulse compared to LREs. Although throttling is sometimes feasible using a movable pintle to adjust the nozzle throat area, SRMs are generally less flexible than LREs in delivering variable thrust. Their thrust output depends primarily on internal ballistics and grain geometry, making it largely fixed. Since combustion continues until the propellant is exhausted, SRMs cannot be reignited at will. In some applications, multiple grains may allow for staged or sequential burns with separate ignitions. However, these burns are predefined and permit only a limited number of ignition events.



Fig. 1.1 Space Launch System close up at Kennedy Space Center. SLS Solid Rocket Booster derived from Space Shuttle Boosters are one of the prominent and latest application of SRM in the space launcher field (Credit to NASA/Sam Lott)

The predefined thrust output of the motor is linked mainly to the composition of the propellant, which is the main factor which influences propellant burning rate, i.e.

the velocity at which the burning surface moves, and grain geometry. Solid propellant can be classified mainly in two categories: double based (DB) propellant, where the cartridge is composed by an homogeneous mixture of oxidizer and fuel agents, and composite propellants, where an heterogeneous mix is created with a polymeric matrix which contains particles of oxidizers. Common choice for DB propellants are nitrocellulose (NC), a solid ingredient that absorbs liquid nitroglycerin (NG), plus minor percentages of additives, while for composite propellants hydroxyl-terminated polybutadiene (HTPB) is used as polymeric matrix, with dispersed ammonium perchlorate crystals (AP) as oxidizer load. Both DB and composite propellants can be enhanced in terms of burning rate with energetic materials such as crystalline nitramines (HMX or RDX) and metallic powders (usually aluminum), among others. Additional chemical agents could be required in the composition to enhance mechanical properties or realize manufacturing process. The exact formulation of the solid propellant is usually tailored for a specific application, varying in additives and components percentage. As stated before grain internal geometry plays a crucial role in determining SRM thrust profile. The inner shape of the grain defines burning area and consequently hot gas mass flow produced, which in turns defines chamber pressure. The pressure level varies propellant burning rate, hence influences surface movement and progressive values of burning surface. The continuous mutual influence between grain inner geometry and chamber pressure allows for a dynamic equilibrium which grants sufficient mass flow in the chamber to sustain required thrust level, for whole motor operation.

The sensitivity of SRM performance to both propellant formulation and internal grain geometry imposes strict requirements on manufacturing and quality control. SRMs manufacturing process is a complex task, not only for the safety hazards posed by the manipulation of potentially explosive materials, but also for the quality requirements that the final product has to meet. Any defect, on the surface or inside the grain, can cause a local variation of burning rate. A sudden increase in burning rate could lead to detonation with catastrophic consequences. For these reasons, modern composite based propellants solid grains are meticulously crafted with a mixing-casting-curing process. All the components are mixed in order to obtain a uniform composition in the pot and then poured inside the motor case. The specific three dimensional geometry of the grain is casted utilizing mandrels, which are then removed before applying a thermal cycle to the grain, in order to cure the

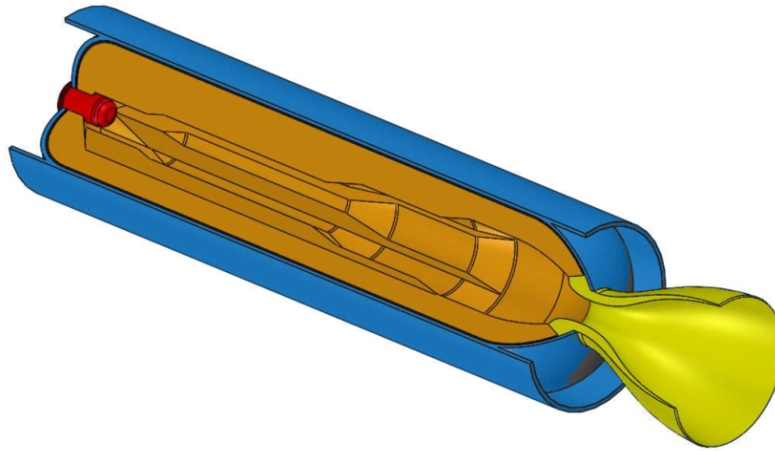


Fig. 1.2 Simplified perspective three-quarter section view of a typical solid propellant rocket motor with the propellant grain (orange), nozzle (yellow), igniter (red) and case with insulation layer (blue).

polymeric matrix and obtain the final solid product. A schematic of a SRM assembly is presented in Figure 1.2.

1.2 Potential impact of Additive Manufacturing for SRM

The traditional manufacturing process and chemicals, whose fundamental aspects are presented in [2–6], presents several limitations. First, the aforementioned additives used to enable the process are toxic compounds. For example, HTPB is cured using multifunctional isocyanates, a class of compounds that includes agents classified as potential human carcinogens and shown to cause cancer in animal studies [7, 8]. Moreover, the thermal curing cycle is time-consuming and requires large, costly furnace facilities. Lastly, grain casting through mandrels imposes limitation in attainable geometries, since final grain shape has to be compatible with mandrel extraction. Collapsible mandrels can be utilized to mitigate this limitation, but at increased cost and complexity. In general, these limitations in casting the shape of the motor reduce the design flexibility and, consequently, the performance envelope of the rocket. To overcome these constraints, research efforts are focusing on innovative additive manufacturing techniques for solid propellant grain manufacturing.

Numerous additive-manufacturing techniques and applications have already reached a mature level of technological readiness, covering a wide spectrum of processes and materials, from polymeric compounds to metal alloys [9–11]. Interest in applying these technologies to energetic materials, and specifically to the manufacture of solid rocket motor grains, is therefore increasing. An overview of these developments is provided in [12, 13], while several applications are reported in [14–23] and in NDIA proceedings for defense-related studies [24, 25]. Among the proposed solutions, an innovative approach based on additive manufacturing and UV-light curing has been introduced for the production of composite propellants [26]. The method eliminates the need for a mandrel and relies on photocurable polymeric binders, which are less toxic and more compatible with 3D-printing processes. This deposition–curing process, in which the grain is built through successive extrusion and irradiation steps, has been explored in several studies [27–31] and represents a promising alternative to conventional manufacturing routes, potentially reducing production timelines by avoiding long thermal curing cycles. Previous investigations [27, 28] compared different UV-curable resins and identified Polybutadiene Diacrylate (PBDA) as a suitable binder for additive manufacturing. The resulting formulations exhibited mechanical and ballistic properties comparable to traditional PB-based composite propellants and showed adequate sensitivity to UV irradiation even at high solid loading, with Ammonium Sulfate (AS) contents up to 90 percent in weight. AS was adopted as an inert surrogate for Ammonium Perchlorate (AP) due to similar physical properties during mechanical testing. Pot-life and extrusion–curing tests were also conducted [29, 30], highlighting the need for further tuning to account for the non-Newtonian behavior of the material during extrusion. Overall, these studies demonstrated the potential of the proposed process, whose material properties have been selected as a reference for the characterization and development of numerical simulation models, while ongoing efforts focus on increasing the Technology Readiness Level through improvements in deposition-system design and grain-design strategies. Nonetheless, this additive manufacturing technology has the potential to improve grain production by enabling customized shapes that are otherwise unachievable via conventional casting. Another key feature of the method is the possibility to locally control the composition of the solid propellant, potentially varying mixing composition prior to local deposition and curing. This feature could enable the production of grains with non-uniform ballistic properties, altering local burning rate through a change in propellant formulation. Another mean of local

ballistic properties variation could be the production of hollow grains with porous texture. This type of geometry can be feasible with additive manufacturing processes and, even if this approach could partially reduce volumetric efficiency of the motor, it can be used to effectively control the regression rate. Some example of testing and experimental activities on UV-curing and additive manufacturing are presented in Figure 1.3

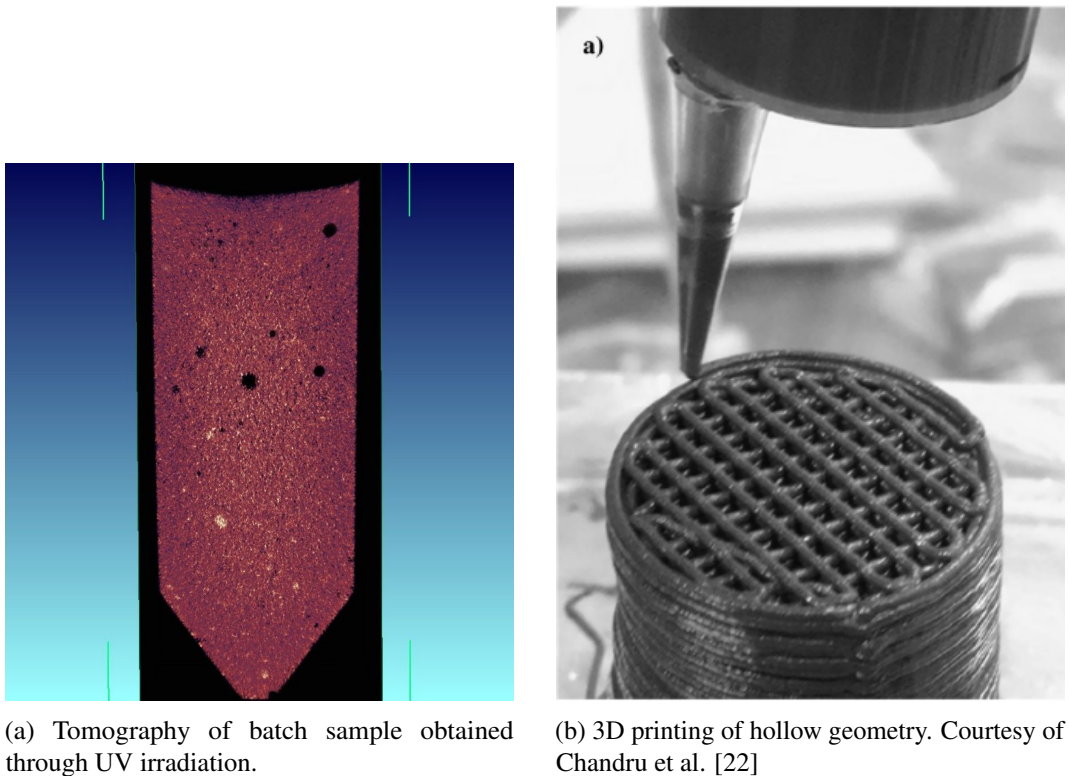


Fig. 1.3 Example from literature of materials study on new additive manufacturing techniques for energetic materials

1.3 Overview of the present work

The rationale behind this work stems from the findings and appealing features of these additive manufacturing concepts. In particular, the aim is to study, explore and understand how non-uniform ballistic properties and constraint-free geometries can be exploited to influence SRM performances. In order to achieve this goal, simulation tools have been developed to evaluate pressure-time response for non-uniform grains

and assess their performances. These tools have been used to support inverse design optimization, i.e. to identify the required configuration to achieve a specific pressure-time history. In particular, Level Set (LS) Methods have been employed to evaluate burning surface geometrical features, and LS time integration was employed to evaluate surface movement under a certain normal velocity distribution. Different implementations have been tested, both on structured cartesian grids and on adaptive linear octree grids. This geometric module has been coupled with a 0D chamber lumped-parameter model, which utilizes geometrical features computed through level set integration to evaluate pressure evolution inside the motor. This predictive tool is the foundation of the inverse design optimization routine, which implements a modified version of the standard design workflow for SRM, accounting for non-uniform properties. This routine employs stochastic optimization algorithms, which are enabled in their numerous evaluations by the use of surrogate models. The latter, trained on data computed through LS evaluations, are able to rapidly predict pressure-time outcomes and accelerate the performance computation for each individual. To present the results obtained in pursuing this objective, this thesis is organized in six chapters. After this introductory section, Chapter 2 is entirely devoted to the description of the numerical methods employed and developed during the research. First, the lumped-parameter model of the chamber is presented, followed by a comprehensive description of Level Set Methods and the coupling between the two models utilized to evaluate pressure evolution in the grain as a consequence of burning development. The next two sections address the problem of geometry initialization, presenting different methods tested for distance field computation. Moreover, numerical schemes for the discrete solution of the LS equation used to compute surface motion are presented in detail in two dedicated sections, for different types of tested grids. The chapter also includes the definition of the strategy employed to define non-uniform regression rates and to control the velocity field which describes surface movement. Additional models for throat erosion and for the mechanical assessment of the grain are presented to conclude this chapter. All these models are implemented for pressure computation and to support the design optimization tool. The next Chapter 3 contains several applications to test the performance of the pressure computation tool. First, pressure computation is performed on a star-shaped grain, for which analytical computation can provide a validation benchmark for the model. A comparison between results obtained with different area computation methods is also presented here. After that the influence

of variable regression rates is explored, using a cylindrical geometry modified to produce a dual thrust pressure profile. The chapter is closed with an application based on experimental curves obtained for small solid grains. In Chapter 4 the inverse design optimization strategy is presented, including the modifications required when dealing with variable burning rates, followed by the description of surrogate models used to accelerate pressure evaluation, which are here described, tested and compared. The following section contains the definition of the optimization problem and the description of the algorithms employed to identify the optimal geometrical-ballistic configuration to meet a desired pressure-time requirement. Several tests are then carried out to assess the tool, using different frameworks in terms of geometries and ballistic distributions. The chapter concludes with some considerations on advantages and limitations of surrogate models application. Chapter 5 is devoted to present some real life civilian applications in the space launch sector, along with some considerations on the improvements that greater design flexibility can offer. Finally, Chapter 6 concludes the work by summarizing the main findings and presenting final considerations on the study and its possible developments in the future.

Chapter 2

Simulation Tools

This section contains an overview on the models and numerical methods employed to evaluate pressure-time response of the burning grain. The aim is to perform burnback simulations accounting for non-uniform ballistic properties of the propellant. Pressure inside the rocket motor grain can be described through a dynamic equilibrium of mass flow produced during propellant burning and mass flow expelled through the nozzle. The mass flow produced by the burning can be computed as $\dot{m}_b = \rho_b A_b r$, where $\dot{m}_b$ is the mass flow, ρ_b is the propellant density, A_b is the burning area and r is the burning rate, i.e. the velocity at which the surface moves as it is consumed during combustion process. This burning rate depends from the pressure inside the motor and, for several propellants, can be approximated with the law $r = ap_c^n$, where p_c is the chamber pressure, and coefficient a and n are empirical values linked to propellant formulation. This relation is affected also by grain temperature, which can affect motor performance, but this work is focused on exploring potentially new thrust profiles in reference operating conditions and the impact of grain temperature is left to future studies. The dynamics of the chamber is derived from the mass conservation equation and this requires the instantaneous evaluation of the grain geometry. These evaluations affect the mass flow balance, hence instantaneous pressure level. The latter varies burning rate and, consequently, surface motion and new geometrical features of the grain. This mutual influence between motor pressure and grain geometry requires a proper coupling between chamber lumped-parameter model and geometry reconstruction. The chosen approach is to evaluate a series of equilibrium states. After computing the equilibrium pressure, geometry is evaluated after computing its motion caused by corresponding burning rate. This

approach is valid if the fluidics phenomena are faster than the surface movement, a valid hypothesis for slow burning propellant like solid one. Details are presented in the corresponding sections. Even if this model relies on simplified description of chamber fluid-dynamics, it gives reliable and fast results, with an acceptable fidelity, especially for a design tool which requires faster evaluations for the main performance driver of the motor, the pressure. In particular, no non-ideal correction factors are included in the present formulation: c^* -efficiency, CF-efficiency, hump correction and scale factor effects are neglected. This choice is consistent with the scope of the work, which is focused on the implementation and geometric management of non-uniform ballistic property distributions rather than calibrating absolute performance predictions. Integration of such correction factors, which would require experimental or high-fidelity firing data for proper calibration, is identified as a future development. To reconstruct the burning surface and evaluate its motion, different methods can be employed [32–34]. Among them, Level Set Methods reconstruct the surface through an implicit definition [35–37]. The desired interface is defined as the zero level of this function. After initialization of the implicit function, obtained through the computation of the Signed Distance Field (SDF), the movement of the surface can be evaluated through the solution of a Partial Differential Equation (PDE), under certain velocity conditions. Different strategies for SDF initialization and PDE numerical solution are presented in the chapter in the correspondent sections. Finally, models for nozzle throat erosion and structural assessment are presented. The former are based on simplified proportional erosion evaluation, while the latter are based on Linear Viscoelastic theory, a common choice for solid propellants. All these models are the foundation for performance evaluation and design of solid grains.

2.1 Level Set Method

Level set methods are utilized to dynamically reconstruct interfaces. These methods rely on the definition of an implicit function $\phi(\vec{x}, t)$ on a defined domain and the interface is defined as the zero level of the implicit function, hence where $\phi = 0$. In a two dimensional domain the boundary can be visualized as the zero level contour while in three dimensional domain it forms a closed surface.

This approach offers the advantages of easily extend the surface description to different dimensional settings and to naturally manage changes in topological

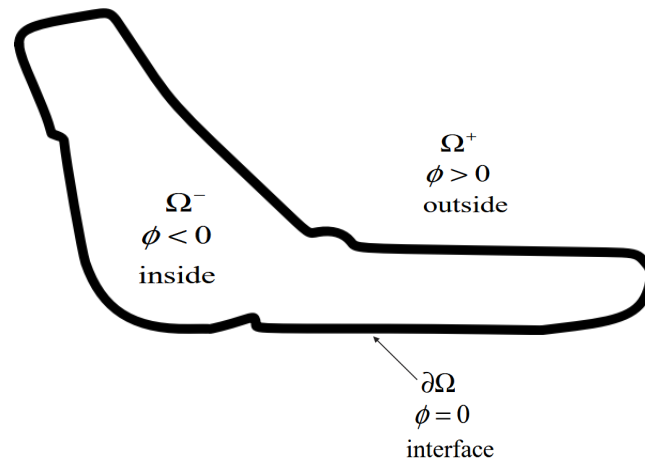


Fig. 2.1 Example representation of 2D domain with a zero level interface that divides the domain

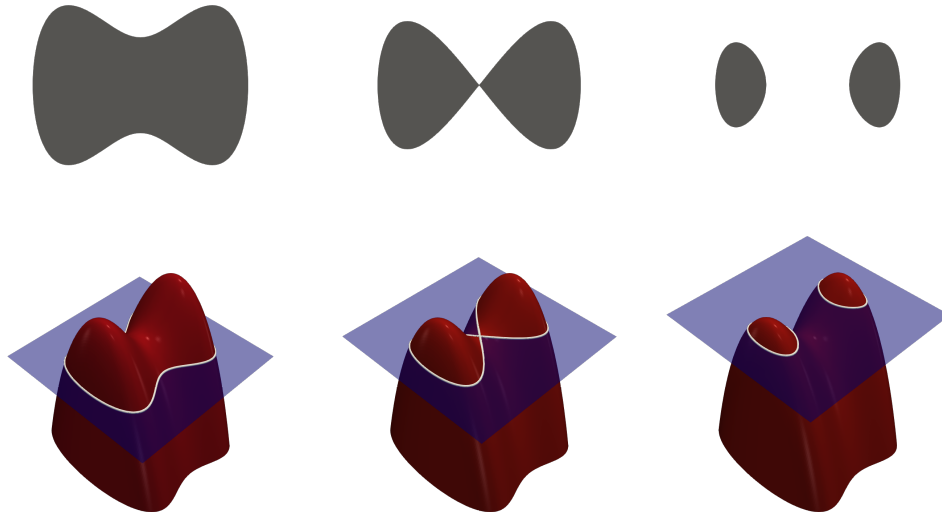


Fig. 2.2 Schematic representation of 2D domain with implicit function and its zero level intersection. Here the contour merging is visible

features of the interface, like merging or separation of the front. It is widely used in different fields, from problems faced in studying waves propagation, like shock-

wave, deflagration and detonation or burning phenomena, to image processing and computer vision, like image restoration or surface reconstruction and to track the interface between two different phases in multiphase simulations. The problem of tracking burning surface of the grain can be addressed to this topic. Instantaneous burning surface is described through the zero level of the ϕ function, which identifies the boundary between the solid grain and the motor chamber, hence the surface that burns at a given time.

The goal of the method is to track the motion of the surface under a velocity field defined in the domain. The motion is evaluated by solving the so called *Level Set Equation* [35]

$$\frac{\partial \phi}{\partial t} + \vec{V} \cdot \nabla \phi = 0 \quad (2.1)$$

The equation in this form is a particular case of an *Hamilton-Jacobi* (HJ) equation, a hyperbolic PDE, and describes in an *Eulerian* fashion the surface evolution, since it is captured by the time evolution of function ϕ . The velocity field can be defined according to the problem under study and in this case, which faces the movement of the surface along its normal direction, it can be defined as follows

$$\vec{V}(\vec{x}, t) = r(\vec{x}, t) \vec{n} = a(\vec{x}) p(t)^n \vec{n} \quad (2.2)$$

linking surface motion to burning rate properties of the grain, exploiting aforementioned relations. Normally, this advection problem can be solved with a general external velocity field, but in this particular case the field vector is oriented along local normal direction, defined by the unit vector normal to the surface pointed towards external region, i.e. the zone of the domain where $\phi > 0$. Thus, the surface burns normal to itself and advances until it meets motor casing. Once the velocity field has been established, the implicit function ϕ must be initialized to meet the desired shape at its zero value. The most suitable choice for implicit function is the signed distance from the interface, which is suitable since it has a zero value on the interface, and can be exploited to define inner and outer regions of the space, assigning a positive or negative value according to a certain convention. Moreover, it allows to simplify level set equation since it retains the valuable property of a unitary gradient norm, i.e. $|\nabla \phi| = 1$. This intrinsic property of distance functions allows to simplify computation for implicit function, for example for the definition

of the normal unit vector $\vec{n} = \frac{\nabla\phi}{|\nabla\phi|}$, which is now exactly defined by the gradient field, without necessary rescaling. However, the initial evaluation of the Signed Distance Field (SDF) can be tricky. In fact, for simple geometries, like spheres or cylinders, this definition can be performed analytically, using correspondent implicit equations of primitive geometries. One can also achieve more complex shapes employing boolean operations to perform union and intersection of such geometries. However, this approach is limited and complex features, like fins in aft-finocyl solid rocket grains, are more challenging to evaluate analytically. Hence, for such complex shapes, it is easier to evaluate distance field and initialize the interface shape by a discrete computation of the signed distance function. This discrete computation can be performed in different ways, employing for example Fast Marching Methods [36], where an exact computed narrow band can be propagated with unitary velocity to retrieve distances in the rest of the domain, or by using diffusive-type approaches or directly computing Euclidean distance on the computational grid. The methods evaluated in this work are presented in the corresponding sections of this chapter.

After the shape initialization, one can describe the geometry of the interface and evaluate its geometrical features. Evaluating correctly the burning area of the grain shape reconstructed by the level set method is fundamental to compute pressure in the motor, as described in Section 2.1. To perform geometrical computation, is possible to exploit the following relation of surface integrals

$$\int_{\partial\Omega} g(\vec{x}) dS = \int_{\Omega} g(\vec{x}) \delta(\phi(\vec{x})) |\nabla\phi(\vec{x})| dV \quad (2.3)$$

Where $g(\vec{x})$ is a generic function and $\delta(\phi(\vec{x}))$ is the Dirac delta function of the variable ϕ . Selecting a constant unitary function for $g(\vec{x})$ and in the case of a signed distance field with $|\nabla\phi| = 1$, the integral reduces only to the integration of the Dirac delta on the domain volume and allows to evaluate the area of $\delta\Omega$. To allow numerical computation of the volume integral, delta function is smoothed on a narrow band near the surface as follows

$$\delta_{\varepsilon}(\phi) = \begin{cases} \frac{1}{2\varepsilon} \left(1 + \cos\left(\frac{\pi\phi}{\varepsilon}\right) \right) & |\phi| \leq \varepsilon \\ 0 & |\phi| > \varepsilon \end{cases} \quad (2.4)$$

where ε is the thickness of the narrow band, and is usually chosen as a small multiplier of the grid characteristic dimension. As $\varepsilon \rightarrow 0$ the function tends to the Dirac delta and the numerical estimation of the integral reaches the measure of the $\partial\Omega$ boundary, hence the area of the burning surface in our case. Another mean of computation of the burning surface is the direct summation of the area of the triangles which form the discrete triangulation of the surface. In fact, the 3D shape can be reconstructed through interpolation of the discrete volume data of ϕ by a triangulation algorithm, reconstructing the zero isosurface. Using this discrete data is possible to compute areas of the single triangles, and sum to compute the whole burning surface. In Chapter 3, some comparisons between the two strategies are presented. Once initialization has been performed and velocity field has been computed, it is possible to solve the Level Set Equation to evaluate surface evolution. To numerically solve the Level Set Equation, different numerical methods can be employed, selecting the most suitable one according to the nature of the equation, which can vary depending on the velocity field imposed, and considering the type of computational grid employed. In this work, two different discrete spatial reconstruction have been employed: a structured cartesian grid and an adaptive octree grid, which is based on a cartesian hierarchical data structure in which each cell can be divided in eight subcells to perform adaptive mesh refinement. This grid structure is managed thanks to bitpit/PABLO software [38], while uniform cartesian structured grid is managed in the environment created by Mitchell [39]. For a normal motion like the one evaluated in the burnback analysis of solid grains, the equation has an hyperbolic nature and upwind computation of spatial derivatives can be utilized to evaluate advection of the surface through finite differencing on cartesian grids, for both first and second order accuracy. Spatial reconstruction accuracy can be increased through Essentially Non Oscillatory (ENO) or Weighted ENO (WENO) schemes. To compute Level Set Equation solution on adaptive octree grids, conservative form is more suitable and Finite Volume methods can be employed. Time evolution computation can be carried out by using Eulerian explicit integration or higher order Runge-Kutta schemes. During LS transport, numerical error can accumulate and the implicit function can lose its SDF properties, i.e. it no longer represents the true distance. Moreover, accumulation zones where LS stops evolving, such as the motor casing in this specific application where the velocity is imposed as zero, or minima where the function flattens to $\nabla\phi = 0$ and LS becomes stuck without updates, for example regions at equal distance from two different points on the surface, can cause incorrect

updates to the implicit function. To preserve the correct values of ϕ during time integration, one can use reinitialization procedures. During a reinitialization step, the implicit function ϕ is corrected to retrieve true distance in the new configuration computed after surface movement. There are different methods to correct ϕ , in this work the reinitialization equation is employed

$$\phi_t + \mathbf{sign}(\phi_0)(|\nabla\phi| - 1) = 0 \quad (2.5)$$

The time integration of this equation in the pseudo-time $\hat{t}$ allows to update the implicit function in the domain until $|\nabla\phi| = 1$ is restored up to a certain tolerance. For well behaved functions this approach is sufficient to retrieve the SDF, however sharp gradients can cause consistent variation of ϕ and can produce unwanted movement of the surface. To prevent these movements, a smeared version of the sign function can be utilized, especially in the nearby of the surface, to avoid excessive movement. To further improve the reinitialization, additional strategies can be employed, such as area preserving conditions or subcell fix method. The latter, employed for the reinitialization, is presented in the section dedicated to numerical implementation of Level Set. Each time step of the LS integration is coupled with pressure integrator results, to update velocity field with the new pressure information at each time step. Details on numerical methods evaluated in this work are presented in next sections.

2.1.1 Initialization via SDF computation

To define the Signed Distance Field, different strategies can be employed [40–42]. To initialize the SDF, the first strategy here employed is a direct Eulerian distance computation on the grid nodes, exploiting information from geometry data, employing techniques in [43, 41]. The shape of the burning grain is designed in CAD software and reconstructed through a triangulation. The triangulation data are used to define the position of the geometry inside the computational grid and distance is computed for every node in the grid. To correctly evaluate distance data for every grid point, it is important to evaluate the nearest feature in the triangulation, hence, if the smallest distance from the specific node of the computational grid is from a triangulation node, edge or triangle. In fact, dealing with complex shapes, it is not sufficient to evaluate point-to-point distance to assess the correct information for distance field initialization. The distance evaluation must be carried out with the

projection of the grid node onto the nearest feature, after its evaluation, and then the computed value must be signed according to the chosen sign convention.

To achieve this computation, the following steps are carried out. Firstly, for every grid node, the distance from every triangulation node is computed. The minimum point-to-point distance among the computed ones, is slightly increased to define a bounding sphere for the specific grid point evaluated. All triangulation nodes inside the bounding sphere are marked and each triangle containing a marked node is selected as a possible candidate, since it can contain the nearest feature to the grid point. For each marked triangle, distance is computed from vertices, edges and from surface, using node positioning to define the local normal of each feature and computing the projection of the grid node on each of them. The smallest distance is selected and, among all marked triangle distances, the smallest is defined as SDF value in the specific grid point. To assign a proper sign to this value, one can use the aforementioned normal to define a convention. Since normals are considered positive if pointing outwards the surface, a positive dot product with the position vector of the grid point gives a positive sign for an outside point, while a negative sign for an inside point. This approach works if one can properly define a unit normal vector for the nearest feature. This is straightforward for triangle surfaces, but is less obvious for vertices and edges, for which a proper pseudo-normal must be defined. This pseudo-normal can be defined, following the work by [44], with a proper combination of the normals of the insisting edges weighted with the angle formed between two adjacent edges and the selected node of the triangulation. For a single edge, local normal is a symmetric combination, i.e. with equal weights, of the normal of the two surfaces of the triangles which share the node. With this crafted information on surface orientation in the nodes, all distance information in the grid can be properly signed, and burning surface can be implicitly reconstructed through LSM. An overview of the SDF initialization algorithm is resumed in Algorithm 1.

This method is very accurate in reconstructing the surface and can easily deal with complex shapes, but it can be expansive since it requires a lot of distance computation and it is very sensitive to the dimension of the problem. However, since initialization heavily influences the precision of the geometry reconstruction and every error is carried along successive LS equation integration, it is acceptable to perform costly computation in this phase to increase the precision of successive ones.

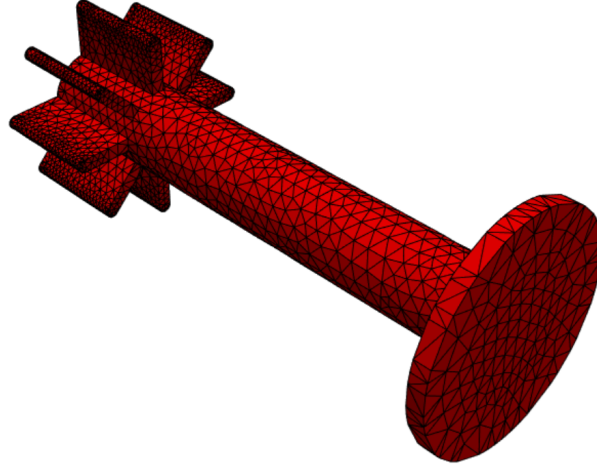


Fig. 2.3 Visualization of finocyl geometry inner cavity triangulation from which SDF is computed and burning surface implicitly reconstructed

Algorithm 1 Direct Distance Computation

Step	Description
1	Select grid point
2	Compute non-squared Euclidean distance from each triangulation point
3	Define the sphere bound and mark candidate triangles
4	For each triangle in the bound T . Evaluate vertices distance Evaluate edges distances Evaluate surface distance Select minimum and compute pseudo-normals if needed
5	Perform dot product and sign the distance
6	Repeat for every grid point

2.1.2 Initialization via Heat Method

The other approach chosen to initialize distance field is the so called Heat Method or Signed Heat Method. This approach stems from findings published in [45]. The method is based on the consideration that, under specific assumption, a diffusive equation can be solved to retrieve the distance from a defined subset, in this case

the burning surface in a three dimensional domain. In particular, the method is based on the idea that, considering an infinitely small time, heat irradiates from a source following the shortest path and one can use the information on the irradiating directions to reconstruct the shortest paths and evaluate the distances in the domain, from that heat source. From a mathematical point of view, heat diffusion is governed by the unsteady heat equation

$$\partial_t u - \Delta u = 0, \quad u(0, x) = u_0(x), \quad (2.6)$$

whose solution admits a representation formula in terms of the fundamental solution or heat kernel $K_t(x, y)$,

$$u(t, x) = \int_{\Omega} K_t(x, y) u_0(y) dy, \quad (2.7)$$

which allows to evaluate the diffused scalar $u(t, x)$ at a point x after a diffusion time t from an initial distribution u_0 , and in particular from a point source located at y [46]. In Euclidean space the kernel has explicit Gaussian form

$$K_t(x, y) = (4\pi t)^{-n/2} \exp\left(-\frac{\phi(x, y)^2}{4t}\right), \quad (2.8)$$

where $\phi(x, y)$ denotes the Euclidean distance between the two points. The Gaussian form means that diffusion from a point source produces a normal distribution in space whose variance grows linearly in time. Taking the logarithm gives

$$\log K_t(x, y) = -\frac{\phi(x, y)^2}{4t} - \frac{n}{2} \log(4\pi t), \quad (2.9)$$

and since $t \log t \rightarrow 0$ as $t \rightarrow 0^+$, multiplying by $-4t$ removes the prefactor contribution and isolates the squared distance in the limit. This asymptotic link between infinitesimal-time diffusion and metric distance is precisely formalized by the so called *Varadhan formula*

$$\phi(x, y) = \lim_{t \rightarrow 0} \sqrt{-4t \log K_t(x, y)} \quad (2.10)$$

This implementation is strongly dependent on exact heat kernel evaluation and can be heavily influenced by errors in it. For this reason, its application for distance

computation has been limited. Crane et al. [1, 47] proposed, exploiting the aforementioned relation, the Heat Method which evaluates distance exploiting information on heat gradient from a source, since, for small time interval, the latter is parallel to true distance. In fact, for short diffusion time the heat field generated by a source set satisfies the

$$u(t, x) = A(t) \exp\left(-\frac{\phi(x)^2}{4t}\right), \quad (2.11)$$

where $\phi(x)$ denotes the distance from the source and A is a slowly varying factor. Taking the gradient gives

$$\nabla u \approx u \left(\nabla \log A - \frac{1}{4t} \nabla(\phi^2) \right) = u \left(\nabla \log A - \frac{\phi}{2t} \nabla \phi \right), \quad (2.12)$$

so that, as $t \rightarrow 0^+$, the dominant term is proportional to $\nabla \phi$ and therefore ∇u is asymptotically parallel to the true distance gradient. The method therefore discards magnitude information and keeps only directional data by defining the normalized vector field

$$X = -\frac{\nabla u}{\|\nabla u\|}, \quad (2.13)$$

which approximates the unit distance gradient field. The main steps for the method are resumed in Algorithm 2. Firstly, the heat source is defined following the studied shape and heat diffusion is computed for a small time t . After computing heat diffusion, u field is retrieved and its normalized gradient field is evaluated. Finally, distance is retrieved through a Poisson problem obtained by least-squares integration of the vector field, namely

$$\Delta \phi = \nabla \cdot X, \quad (2.14)$$

whose solution gives the distance function up to an additive constant. The method has the advantage to retrieve distance information only from the gradient ∇u without magnitude information, hence it is more flexible with respect to direct usage of Varadhan relation or different algorithms. Moreover, the method can be applied provided the existence of gradient, divergence and laplacian operator, and it is not linked to data type and can be extended to any curved domain for which these operators can be generalized. For these reasons, the Heat Method can be employed to compute distance fields on cartesian domains as well as geodesic distances from sources defined on curved manifolds, starting from a variety of different data type from which the source can be reconstructed.

Algorithm 2 The Heat Method [1]**Step** **Description**

- | | |
|-----|---|
| I | Integrate the heat equation $\dot{u} = \Delta u$ for a fixed short time t . |
| II | Evaluate the normalized vector field $X = -\nabla u / \nabla u $. |
| III | Solve the Poisson equation $\Delta \phi = \nabla \cdot X$. |

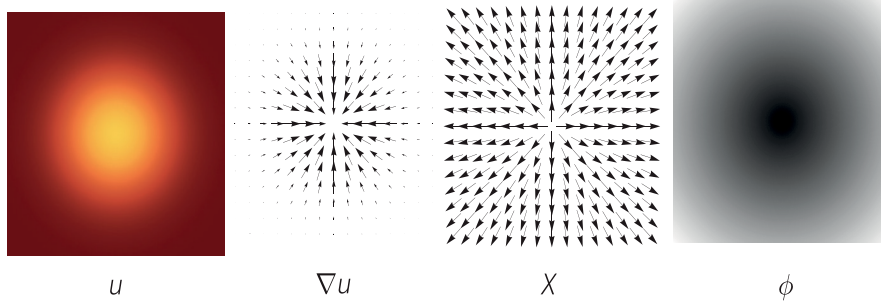


Fig. 2.4 Visualization of intermediate step of heat method. Figure from [1]

The first step of the method requires the time stepping to compute u field after diffusion. To numerically compute the time step, a single backward Euler step is performed as follows

$$\frac{\partial u}{\partial t} = \Delta u \Rightarrow (\mathbb{I} - t\Delta)u_t = u_0 \quad (2.15)$$

where u_t and u_0 are the values of the variable at the end of the time step and at the start time respectively, and $\mathbb{I}$ is the identity matrix. The initial condition u_0 is defined assigning an high value in the grid points which are near the analyzed shape, while the rest of the domain is set to zero. Heat diffuses from this u_0 in the domain. To perform this diffusion step, Laplacian computation is required. Second spatial derivative must be computed. For a structured cartesian grid one can employ finite differences, utilizing second order accurate centered schemes. For a three dimensional grid, this computation requires a 7-point stencil, described in Section 2.1.3. This computational stencil allows to build the laplacian matrix L , and utilize it to compute the time integration step. A proper boundary condition must be provided to perform the Laplacian evaluation. Literature shows that homogeneous Neumann conditions with zero flux on the border can be employed to evaluate this diffusion step, provided a sufficient distance from the surface to the boundary. After computing u_t , the method requires the computation of its gradient and again on the same grid

one can employ centered schemes to evaluate partial derivatives along the three dimensions of the cartesian domain. Applying correspondent directional derivative matrix obtained through these schemes to u allows to compute the gradient and evaluate flipped normalized field X . The final step requires solving the Poisson problem by computing divergence, which is evaluated with the same numerical schemes for gradient computation. Using the same matrices produced for previous steps, it is immediate to compute the final step, solving for ϕ and evaluating the distance in the domain, with boundary conditions coherent to the first step. Since the solution ϕ is defined up to an arbitrary constant value, a proper shift of the solution is applied to meet zero level at source, hence the mean value of the solution in the source grid points is used as shift factor.

The heat method allows to compute the distance without a sign, hence a proper method to define the sign and divide the domain between inner and outer part is required. An extension of the method, called Signed Heat Method [48, 49], which performs the same operations of the basic method, but solves a vector diffusion problem, is resumed in the Algorithm 3. A visualization of the intermediate steps of the method is provided in Figure 2.5. On cartesian grid, this problem reduces simply to the computation of the heat diffusion component by component, and the normalized field X is not computed through the gradient computation, but is formed by the various components of the vector diffusion problem. By solving this problem, information on normal direction is retained through the integration and can be used to perform dot product and evaluate the sign of the distance at a certain point in the domain. Of course, this information must be provided during the definition of initial condition, when instead of defining a scalar u_0 , the local normal vector rescaled with u_0 magnitude is defined. This requires additional computations and the definition of normal local can be obtained through pseudo-normal definition already exploited for direct distance computation. Some testing has been carried out on two dimensional grids, on different shapes, starting from a simple circular shape (Figure 2.6 and Figure 2.7) to more complex geometries (Figure 2.8, Figure 2.9 and Figure 2.10). The three dimensional extension follows the same logic, starting from spherical geometries (Figure 2.11 and Figure 2.12) to complex finocyl shapes (Figure 2.13 and Figure 2.14). The computation follows the same steps as the previously described algorithms, changing the derivative operators evaluation in order to deal with vector components. Results show a good performance of the method for 2D shapes, even for sign evaluation. However, results for 3D case are more problematic, since the

initialization requires a lot of discrete source points to reconstruct complex shapes. For simple geometries, like spheres, the computation is fast and accurate with a small number of points, while the test on complex grain shapes is not satisfactory.

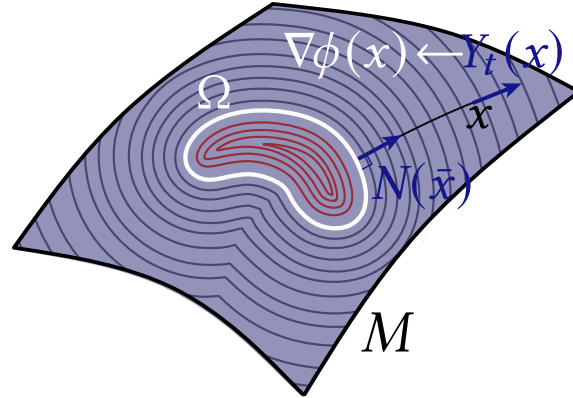


Fig. 2.5 Visualization of intermediate step of signed heat method. M is a generic manifold, in this work simply an $\mathbb{R}^2$ or $\mathbb{R}^3$ domain

Algorithm 3 Signed Heat Method [48]

Step	Description
(1)	Solve the vector diffusion equation $\frac{d}{dt} X_t = \Delta^\nabla X_t$
(2)	Evaluate the vector field $Y_t = -X_t / \ X_t\ $.
(3)	Solve the Poisson equation $\Delta \phi = \nabla \cdot Y_t$

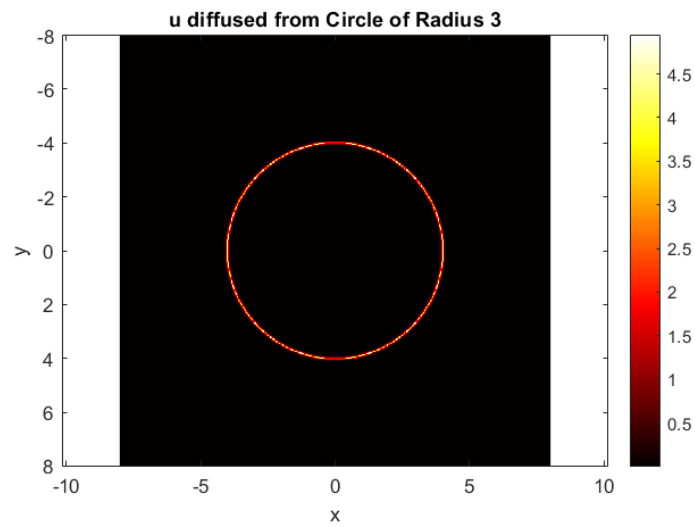


Fig. 2.6 u_0 source diffused for a circle

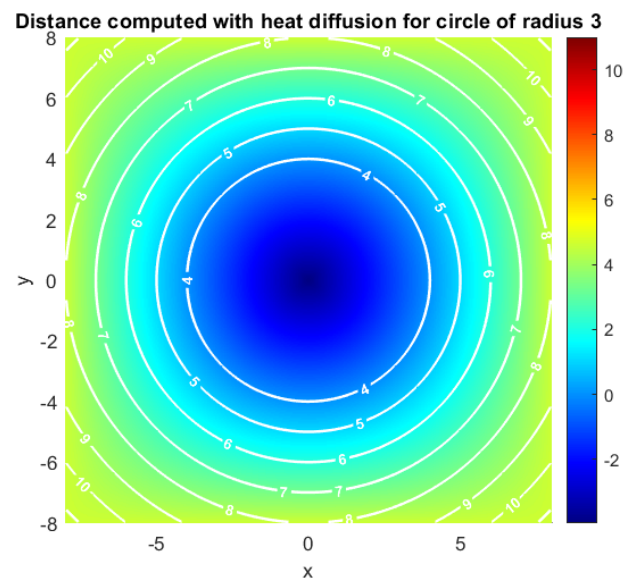


Fig. 2.7 Distance field computed with Signed Heat Method (circle)

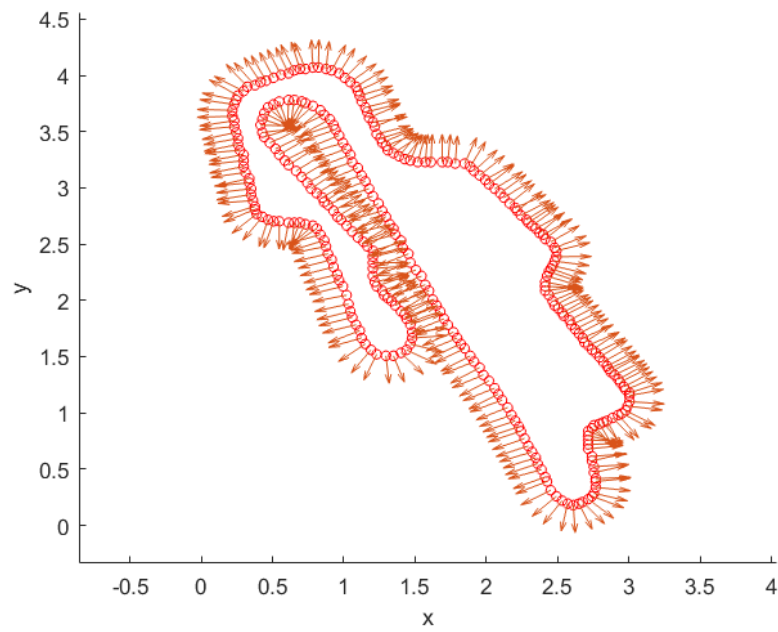


Fig. 2.8 u_0 source with pseudo normals for Signed Heat Method on a complex 2D shape

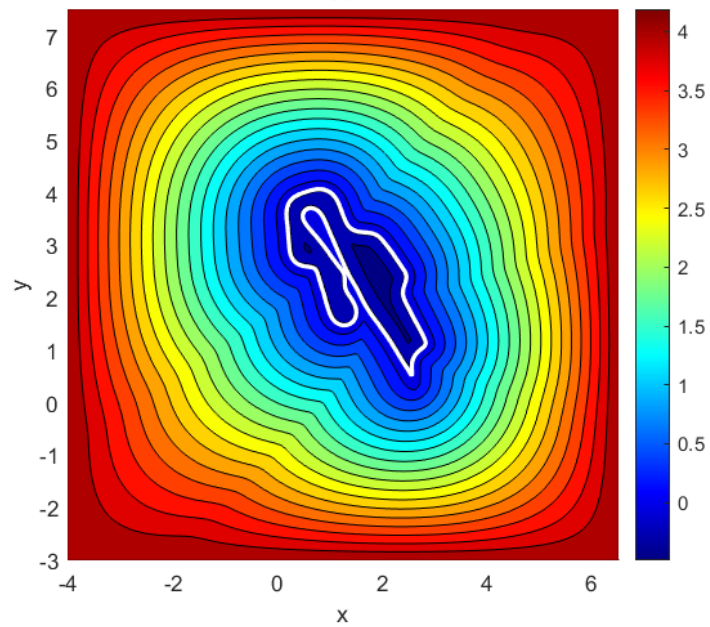


Fig. 2.9 Distance field computation for a complex 2D shape

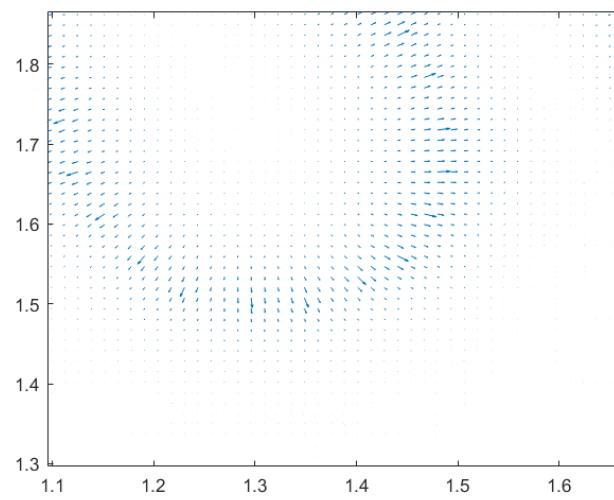


Fig. 2.10 Gradient visualization detail for a complex 2D shape

3D Source Visualization

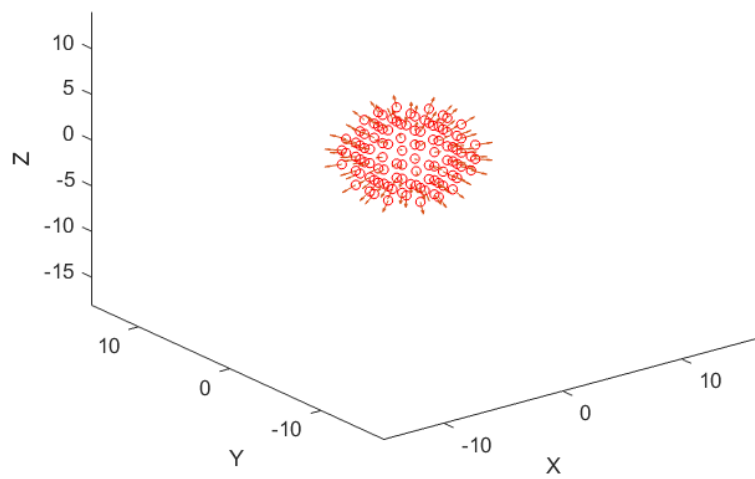


Fig. 2.11 u_0 source diffused for a sphere

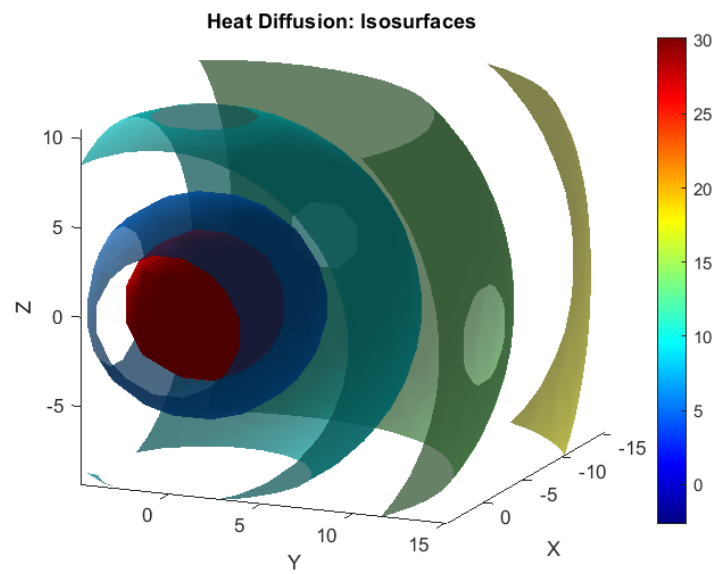


Fig. 2.12 Distance field computed with Signed Heat Method (sphere)

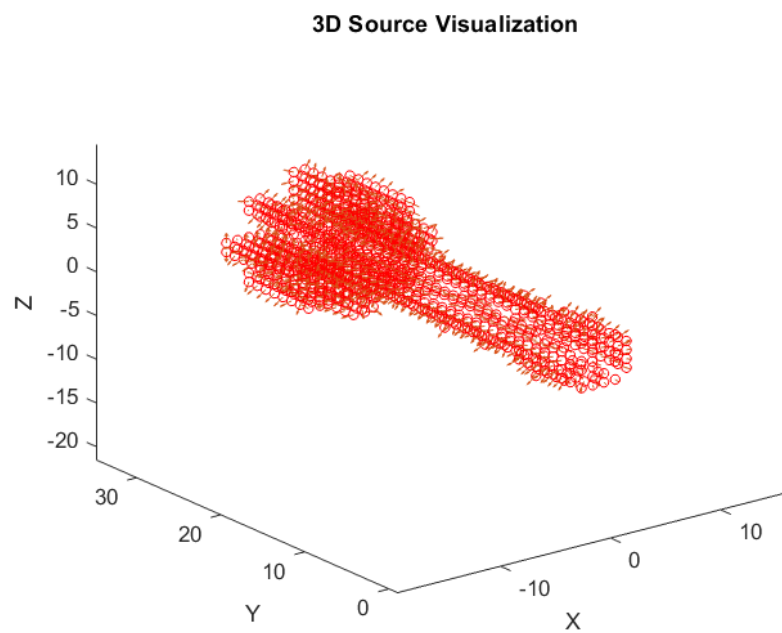


Fig. 2.13 u_0 source with pseudo normals for Signed Heat Method on a 3D finocyl shape

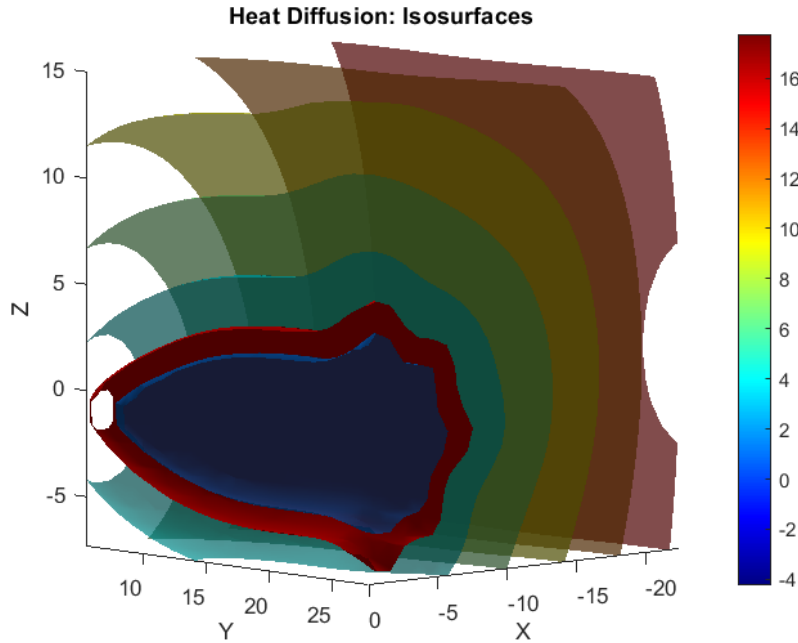


Fig. 2.14 Distance field computation for a 3D finocyl shape

After testing on cartesian grids, the method is tested on adaptive octree grids. In this framework, initialization of heat source is performed using information on exact distance on a narrow band and sign is evaluated by normal dot product computation. Sign is propagated in the domain with a propagation algorithm, exploiting the sign change in the narrow band to divide the domain. After initialization, heat method is applied to retrieve distance field. When dealing with adaptive octree grids, level jump in grid resolution requires particular attention. To easily deal with local refinement, Finite Volume schemes are more suitable than Finite Differences to evaluate discrete derivative operators. Further details on the discrete computation using FV is presented in Section 2.1.5. The method performs well on adaptive grids, and the local refinement allows to obtain higher accuracy in describing complex shapes with respect to cartesian grids. However, higher refinements levels exhibit some instabilities and oscillation in divergence computation, limiting the advantages of local refinement and reducing the favor towards heat methods with respect to direct computation of the SDF.

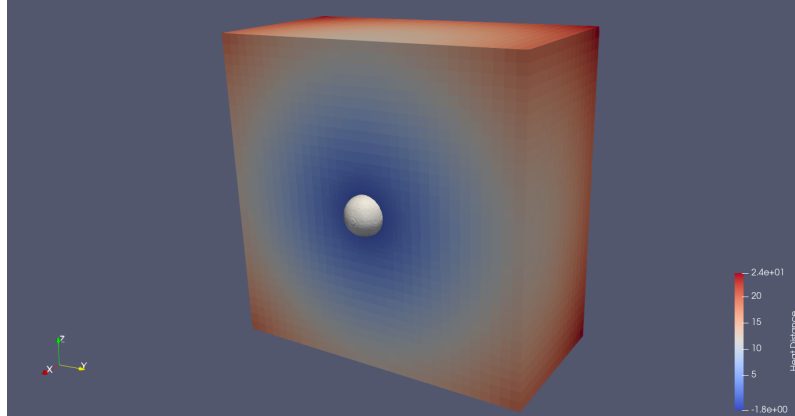


Fig. 2.15 Sphere zero level initialized with Signed Heat Method on an adaptive octree grid

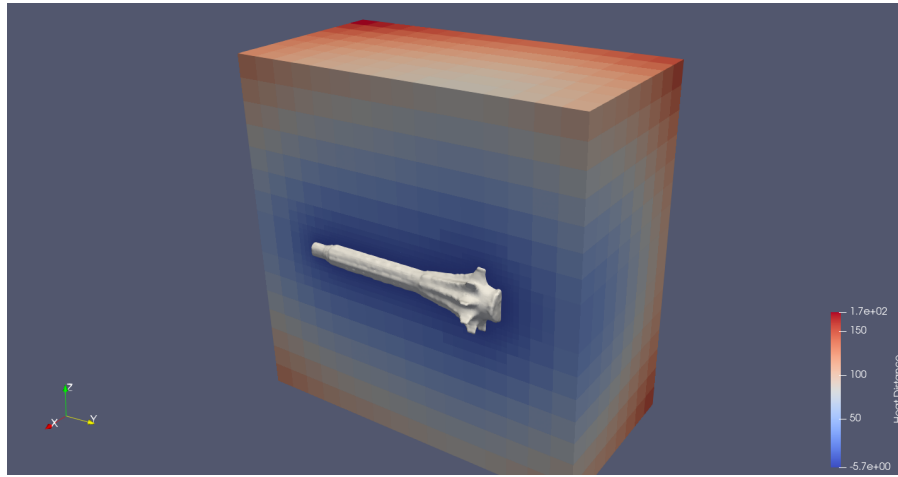


Fig. 2.16 Finocyl zero level initialized with Signed Heat Method on an adaptive octree grid

2.1.3 Finite Difference on Uniform Grids

This subsection contains the description of numerical schemes employed to initialize the implicit function and solve the level set equation on a uniform cartesian grid. As previously described, diffusion-like problems can be employed to initialize distance fields. In order to do that, discrete derivative operators must be defined in the domain and a natural choice in a uniform cartesian grid setting is to utilize finite differencing. Finite difference schemes on uniform Cartesian grids provide a simple and well-established framework for the discretization of elliptic and parabolic operators, and represent a standard choice in the numerical treatment of diffusion-type problems

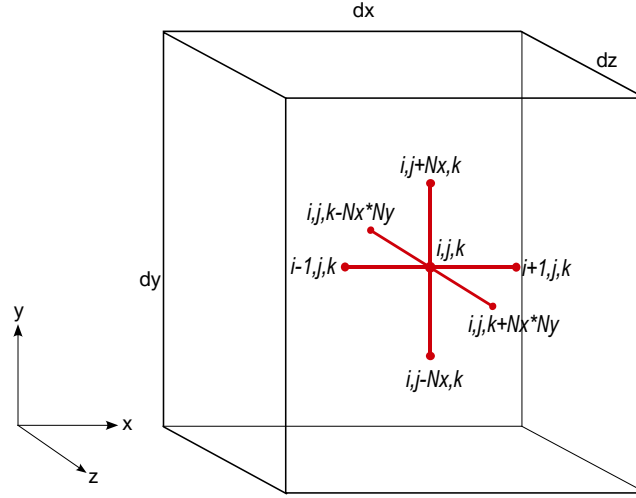


Fig. 2.17 Three dimensional stencil used to compute discrete operators on cartesian grids

[50, 51]. Using classical centered schemes one can easily compute first and second order derivatives. Employing a 7-point stencil, as the one pictured in Figure 2.17 for a three dimensional case, the Laplacian at a generic grid point, for a N^3 uniform grid, can be written as

$$\Delta_h u_{i,j,k} = \frac{1}{h^2} \left(u_{i+1,j,k} + u_{i-1,j,k} + u_{i,j+1,k} + u_{i,j-1,k} + u_{i,j,k+1} + u_{i,j,k-1} - 6u_{i,j,k} \right),$$

$$i, j, k = 1, 2, \dots, N$$
(2.16)

Employing the discrete laplacian definition, it is possible to assemble the laplacian matrix L , with a classical multidagonal sparse pattern and the following structure represented in 2.18. With this definition, it is possible to evaluate the Laplacian operator and perform the first step of the Heat Method algorithm following Equation 2.15. To complete initialization through the Heat Method, gradient components are computed using the same finite difference scheme and assembling partial derivative matrices to evaluate gradient components everywhere in the domain. Basically, these matrices are constructed extracting co-diagonals from laplacian matrix L . The very same matrices can be used to compute divergence of a vector field, by applying them consistently to vector field components and summing. For a uniform grid with $N_x = N_y = N_z = 3$ we obtain the structures represented in Figure 2.19

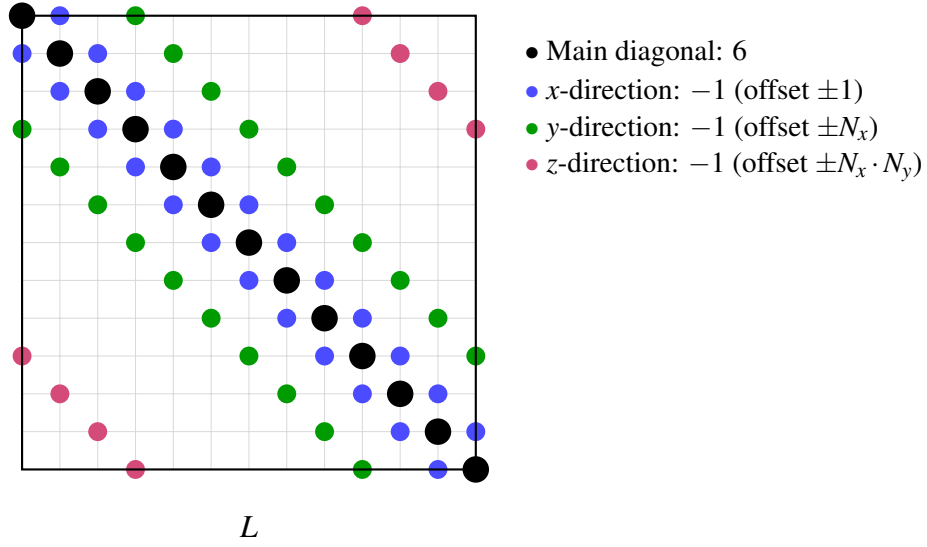


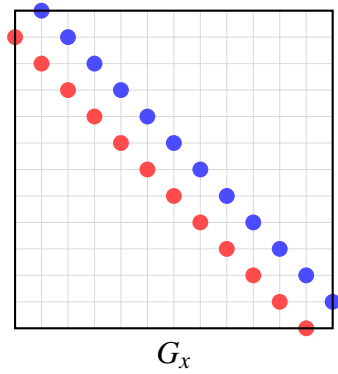
Fig. 2.18 Sparsity pattern of the Laplacian matrix L implementing the 7-point stencil on a uniform Cartesian grid. The matrix has value 6 on the main diagonal (black) and -1 on the six off-diagonals corresponding to the neighbors in x , y , and z directions.

With these matrices it is possible to define discrete operators, operating on scalar u and vector $\mathbf{v}$ on N degree of freedom, as follows

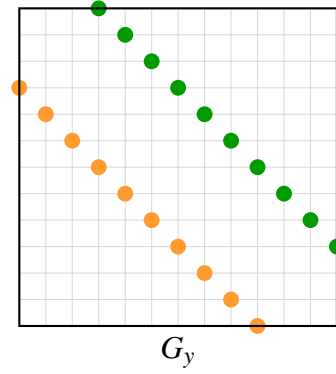
$$\nabla_h u = \begin{bmatrix} G_x \\ G_y \\ G_z \end{bmatrix} u = \begin{bmatrix} G_x u \\ G_y u \\ G_z u \end{bmatrix} \in \mathbb{R}^{3N}, \quad u \in \mathbb{R}^N$$

$$\nabla_h \cdot \mathbf{v} = \begin{bmatrix} G_x & G_y & G_z \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = G_x v_x + G_y v_y + G_z v_z \in \mathbb{R}^N, \quad \mathbf{v} \in \mathbb{R}^{3N} \quad (2.17)$$

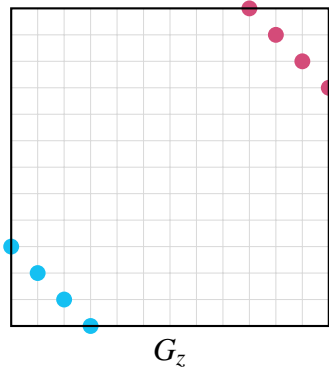
and perform all the necessary operations to complete the Heat Method Initialization. Besides the initialization, this discretization can be employed in approximating the spatial derivative terms during resolution of LS equation. However, this type of discretization is more suitable for diffusive type velocity field for example in fronts whose propagation is affected by local mean curvature, while in advection type equation like 2.1, an upwinding scheme is desirable. In the case of a pure normal advection of the front defined by zero level of implicit function ϕ , velocity



(a) G_x matrix structure with non-zero elements on the first and second codiagonals (offset ± 1). Blue: upper diagonal, Red: lower diagonal.



(b) G_y matrix structure with non-zero elements on diagonals offset by $\pm N_x$ (shown here with $N_x = 3$). Green: upper band, Orange: lower band.



(c) G_z matrix structure with non-zero elements on diagonals offset by $\pm N_x \cdot N_y$ (shown here with $N_x \cdot N_y = 4$). Purple: upper band, Cyan: lower band.

Fig. 2.19 Sparsity patterns of directional derivative matrices on a uniform Cartesian grid. Each matrix implements centered finite differences along one coordinate direction.

field is defined as $\vec{V} = a\vec{n}$. If one is dealing with SDF, the unit gradient property of can be employed and the LS equation simply states $\phi_t = -a$, which means that in a discrete step of length Δt the front moves of a quantity $a\Delta t$, and the points of the domain at this distance from the original zero level at the starting time t_0 would be the new position of the interface after Δt time. This simple case requires that unit gradient property of SDF is valid from initial definition of ϕ function throughout time integration, and this is not always verified both for discrete computation errors in initialization phase and during numerical solution of the LS equation, and for velocity fields which are non-uniform both in space and time. The SDF property of the implicit function can be restored through a reinitialization procedure, that will be described later in the section, but also in this case it is convenient to solve the equation with methods that don't require such simplifications. Even if more generic methods can be more expensive, they have far better accuracy and are more flexible and robust. Most importantly, they naturally handle non-uniform velocity fields, which are central to this work.

The simplest upwind scheme that can be used to compute the spatial derivative term of LS equation is constructed as

$$\phi_{x_j}^{\pm} = \left(\frac{\partial \phi}{\partial x_j} \right)_{\text{up}} = \begin{cases} \frac{\phi(\mathbf{x}) - \phi(\mathbf{x} - \Delta x_j \mathbf{e}_j)}{\Delta x_j}, & v_j > 0 \quad (+) \\ \frac{\phi(\mathbf{x} + \Delta x_j \mathbf{e}_j) - \phi(\mathbf{x})}{\Delta x_j}, & v_j < 0 \quad (-) \end{cases} \quad (2.18)$$

where j is the generic direction in $\mathbb{R}^3$, $\mathbf{e}_j$ is the correspondent unit vector, Δx_j is the cell dimension, hence the spatial step of the differencing and finally v_j is the component of the velocity vector expressed in the basis $\mathbf{e}$. The upwinding direction along a specific component depends on the sign of the velocity component in that direction. This well known method allows to reconstruct in a simple way the gradient of ϕ and it can be coupled with a forward Euler time integration scheme as follows. Only a component is here reported, but the extension to higher dimension is straightforward. For the i -th grid point

$$\phi_i^{n+1} = \phi_i^n - \Delta t v_i^n \left(\frac{\partial \phi}{\partial x} \right)_{\text{up},i}^n = \phi_i^n - \Delta t v_i^n \begin{cases} \frac{\phi_i^n - \phi_{i-1}^n}{\Delta x}, & v_i^n > 0 \\ \frac{\phi_{i+1}^n - \phi_i^n}{\Delta x}, & v_i^n < 0 \end{cases} \quad (2.19)$$

This widely used scheme provides a convergent reconstruction of the solution, provided its stability is granted through the satisfaction of the *CFL* condition, here reported for a three dimensional setting

$$\Delta t \left(\frac{\max\{|\vec{V}|\}}{\min\{\Delta x, \Delta y, \Delta z\}} \right) = \alpha \leq 1 \quad (2.20)$$

To improve the first order accuracy of the classical upwind scheme, one can employ different strategies. While centered schemes like the one previously described provide better accuracy, they can't provide stability and are not suited for pure advective settings. To improve accuracy of the reconstruction while keeping upwinding properties, Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory schemes are employed to reconstruct oriented derivatives $\phi_{x,y,z}^\pm$ [35, 52–54]. ENO schemes were firstly employed to reconstruct numerical fluxes by constructing the smoothest polynomial interpolation and were after extended in [55] to solution of Hamilton-Jacobi equations. Accurate one-sided derivatives are computed by adaptively selecting the smoothest stencil for polynomial interpolation, by the use of divided differences.

Starting from nodal values $\phi_i = \phi(x_i)$ on a uniform grid, the divided differences are defined recursively. Zeroth-order differences coincide with the nodal values:

$$D_i^0 \phi = \phi_i. \quad (2.21)$$

First, second, and third divided differences are defined as

$$D_{i+1/2}^1 \phi = \frac{D_{i+1}^0 \phi - D_i^0 \phi}{\Delta x}, \quad (2.22)$$

$$D_i^2 \phi = \frac{D_{i+1/2}^1 \phi - D_{i-1/2}^1 \phi}{2 \Delta x}, \quad D_{i+1/2}^3 \phi = \frac{D_{i+1}^2 \phi - D_i^2 \phi}{3 \Delta x} \quad (2.23)$$

These differences quantify local smoothness and guide the selection of the interpolation stencil. The derivative is approximated by differentiating a local interpolating polynomial

$$\phi(x) \approx Q_0(x) + Q_1(x) + Q_2(x) + Q_3(x), \quad \phi_x(x_i) \approx Q'_1(x_i) + Q'_2(x_i) + Q'_3(x_i). \quad (2.24)$$

The first-order term recovers the standard upwind derivative previously described:

$$Q'_1(x_i) = D_{k+1/2}^1 \phi, \quad (2.25)$$

with $k = i - 1$ for ϕ_x^- and $k = i$ for ϕ_x^+ . Higher-order corrections are obtained by comparing the magnitude of successive divided differences and choosing the smoothest contribution, hence the side with smallest absolute value for divided difference. For example, for the second order:

$$c = \begin{cases} D_k^2 \phi & \text{if } |D_k^2 \phi| \leq |D_{k+1}^2 \phi|, \\ D_{k+1}^2 \phi, & \text{otherwise} \end{cases} \quad Q'_2(x_i) = c(2(i-k) - 1) \Delta x. \quad (2.26)$$

Similarly, after defining $k^* = k - 1$ if $D_k^2 \phi$ is selected and $k^* = k$ otherwise, the third-order term is defined as

$$\tilde{c} = \begin{cases} D_{k^*+1/2}^3 \phi, & \text{if } |D_{k^*+1/2}^3 \phi| \leq |D_{k^*+3/2}^3 \phi| \\ D_{k^*+3/2}^3 \phi, & \text{otherwise} \end{cases} \quad (2.27)$$

$$Q'_3(x_i) = \tilde{c}[3(i-k^*)^2 - 6(i-k^*) + 2](\Delta x)^2$$

Finally, the ENO derivative is given by

$$\phi_x(x_i) \approx Q'_1(x_i) + Q'_2(x_i) + Q'_3(x_i), \quad (2.28)$$

where upwinding is preserved by the first term that recalls classic scheme, while additional terms implemented through adaptive stencil enables high-order accuracy in smooth regions while avoiding oscillations near steep gradients or discontinuities, up to third order accuracy with this approximation. The same reconstruction is applied to compute derivatives along other axes $\phi_{y,z}$ in every point on the grid, and can consequently be used in a time integration scheme to evaluate the gradient of the

implicit function. WENO schemes arise from this implementation. In particular, the adaptive selection of the third order ENO polynomial interpolation allows to define three different polynomial, by combining the possible choices for higher order terms. For example, the left derivative along x direction can be expressed as

$$\begin{aligned}\phi_1^x &= \frac{1}{3}d_1 - \frac{7}{6}d_2 + \frac{11}{6}d_3 \\ \phi_2^x &= -\frac{1}{6}d_2 + \frac{5}{6}d_3 + \frac{1}{3}d_4 \\ \phi_3^x &= \frac{1}{3}d_3 + \frac{5}{6}d_4 - \frac{1}{6}d_5\end{aligned}\tag{2.29}$$

where $d_1 = D^- \phi_{i-2}$, $d_2 = D^- \phi_{i-1}$, $d_3 = D^- \phi_i$, $d_4 = D^- \phi_{i+1}$, $d_5 = D^- \phi_{i+2}$, with $D^- \phi$ as the left upwind derivative at i -th grid point as defined in Equation 2.18.

The choice of the least oscillating polynomial among the three is considered to be an overkill for regions of the space where data are already sufficiently smooth. For this reason, a convex combination of these polynomials, through an adaptive weighting of them, is employed:

$$\phi_x = \omega_1 \phi_1^x + \omega_2 \phi_2^x + \omega_3 \phi_3^x, \quad \omega_1 + \omega_2 + \omega_3 = 1, \quad 0 \leq \omega_k \leq 1. \tag{2.30}$$

The weights are computed based on smoothness indicators of each candidate stencil, which can be evaluated as

$$S_1 = \frac{13}{12}(v_1 - 2v_2 + v_3)^2 + \frac{1}{4}(v_1 - 4v_2 + 3v_3)^2, \tag{2.31}$$

$$S_2 = \frac{13}{12}(v_2 - 2v_3 + v_4)^2 + \frac{1}{4}(v_2 - v_4)^2, \tag{2.32}$$

$$S_3 = \frac{13}{12}(v_3 - 2v_4 + v_5)^2 + \frac{1}{4}(3v_3 - 4v_4 + v_5)^2. \tag{2.33}$$

The unnormalized weights α_k are defined as

$$\alpha_1 = \frac{0.1}{(S_1 + \varepsilon)^2}, \quad \alpha_2 = \frac{0.6}{(S_2 + \varepsilon)^2}, \quad \alpha_3 = \frac{0.3}{(S_3 + \varepsilon)^2}, \tag{2.34}$$

where $\varepsilon = 10^{-6} \max\{v_1^2, v_2^2, v_3^2, v_4^2, v_5^2\} + 10^{-99}$ prevents division by zero.

Finally, the normalized adaptive weights are obtained by

$$\omega_k = \frac{\alpha_k}{\alpha_1 + \alpha_2 + \alpha_3}, \quad k = 1, 2, 3. \quad (2.35)$$

This construction ensures that in smooth regions the convex combination achieves optimal fifth-order accuracy, while in non-smooth regions the weights automatically reduce the influence of stencils crossing steep gradients or discontinuities, recovering the original ENO behavior. The very same construction can be replicated for right derivative [35].

Higher accuracy can also be obtained for time discretizations. Even though the numerical solution is generally more sensitive to spatial discretizations, higher order time schemes are useful to provide stability when integrating ENO/WENO schemes. Total Variation Diminishing Runge-Kutta (TVD-RK) methods have been proposed to perform such discretizations [35, 56]. These methods rely on a convex combination of simple explicit Euler time steps, in order to obtain a better accuracy while preserving the variation diminishing property of the explicit Euler scheme under CFL condition. This connection is evident since first and second order TVD-RK methods are identical to Euler and Heun schemes respectively. The latter is shown below

$$\begin{aligned} \text{Step 1: } & \frac{\phi^{n+1} - \phi^n}{\Delta t} + V^n \cdot \nabla \phi^n = 0 \\ \text{Step 2: } & \frac{\phi^{n+2} - \phi^{n+1}}{\Delta t} + V^{n+1} \cdot \nabla \phi^{n+1} = 0 \\ \text{Final Average: } & \phi^{n+1} = \frac{1}{2}\phi^n + \frac{1}{2}\phi^{n+2} \end{aligned} \quad (2.36)$$

while the third order TVD-RK scheme uses the following combination of time steps to perform the reconstruction

$$\begin{aligned}
\text{Step 1: } & \frac{\phi^{n+1} - \phi^n}{\Delta t} + V^n \cdot \nabla \phi^n = 0 \\
\text{Step 2: } & \frac{\phi^{n+2} - \phi^{n+1}}{\Delta t} + V^{n+1} \cdot \nabla \phi^{n+1} = 0 \\
\text{Average Step: } & \phi^{n+\frac{1}{2}} = \frac{3}{4}\phi^n + \frac{1}{4}\phi^{n+2} \\
\text{Step 4: } & \frac{\phi^{n+\frac{3}{2}} - \phi^{n+\frac{1}{2}}}{\Delta t} + V^{n+\frac{1}{2}} \cdot \nabla \phi^{n+\frac{1}{2}} = 0 \\
\text{Final Average: } & \phi^{n+1} = \frac{1}{3}\phi^n + \frac{2}{3}\phi^{n+\frac{3}{2}}
\end{aligned} \tag{2.37}$$

Although higher order TVD-RK schemes can be constructed, since practical implementation show little improvement beyond this level of accuracy with a strongly increased computational cost, further increased by the necessity of coupling the chamber lumped parameter model in the specific proposed application, no further implementation is applied in the scope of this work. After selecting the spatial derivatives computational scheme, the numerical solution of the LS problem in the case of a pure advection movement, requires selecting the appropriate directional contribution for each derivative. This is performed through the Godunov method [57], which determines, one dimension at a time, the correct upwind direction based on the characteristic information of the equation. In points where the characteristics converge, such as sonic locations or where multiple fronts meet, the method selects the derivative corresponding to the dominant propagation direction, ensuring a non-oscillatory and physically consistent evolution. In contrast, where characteristics diverge, the local contribution to the flux may be null, since no unique upwind direction can be identified. The selection of the directional derivative in each coordinate direction follows the classical expression attributed to Rouy and Tourin [58], here reported for x direction

$$\phi_x^2 \approx \begin{cases} \max \left(\max(\phi_x^-, 0)^2, \min(\phi_x^+, 0)^2 \right), & v > 0, \\ \max \left(\min(\phi_x^-, 0)^2, \max(\phi_x^+, 0)^2 \right), & v < 0, \end{cases} \tag{2.38}$$

where ϕ_x^- and ϕ_x^+ are the left and right derivative approximations in the considered spatial direction, and v denotes the local characteristic speed of the equation. Artificial viscosity methods provide an alternative mechanism for enforcing the

entropy condition through a small artificial viscosity additional contribution. While straightforward to implement, these introduce a degree of smoothing that reduces the accuracy near sharp gradients, and are therefore less suited in situations where preservation of the interface shape is critical.

2.1.4 Reintialization procedure

To perform reinitialization as described in the introduction of the chapter, the subcell fix method is employed [59]. Starting from equation 2.5 an Euler time integration scheme can be used for each i -th node

$$\phi_i^{n+1} = \phi_i^n - \Delta t \mathbf{sign}(\phi_i^0) G(\phi)_i \quad (2.39)$$

where $G(\phi) = ||\nabla\phi|| - 1$. The computation of this term can be carried out using Godunov approach, selecting the characteristic direction by finding the greatest magnitude of lateral derivatives

$$G(\phi)_i = \begin{cases} \max(|a_+|, |b_-|) - 1 & \text{if } \phi_i^0 > 0 \\ \max(|a_-|, |b_+|) - 1 & \text{if } \phi_i^0 < 0 \end{cases} \quad (2.40)$$

where, for a single direction, a and b are upwind left and right derivatives

$$\begin{aligned} a &\equiv D_x^- \phi_i = \frac{\phi_i - \phi_{i-1}}{\Delta x} \\ b &\equiv D_x^+ \phi_i = \frac{\phi_{i+1} - \phi_i}{\Delta x} \end{aligned} \quad (2.41)$$

The suffix $\pm$ define maximum and minimum with respect to zero. The definition is analogous to 2.38 Selecting the right derivative for every dimension of the problem, each component of the vector $G(\phi)$ can be computed. Using the smoothed approximation of the sign function S , defined as

$$S(\phi) = \frac{\phi}{\sqrt{\phi^2 + \Delta x^2}} \quad (2.42)$$

using the mean dimension of the grid cell Δx , and the correction factor D

$$D = \frac{\phi^0}{\|\nabla \phi^0\|} \quad (2.43)$$

the subcell fix evaluation for reinitialization is written as

$$\phi_i^{n+1} = \phi_i^n - \Delta t \begin{cases} S(\phi_i^0)|\phi_i^n| - D_i^0 & \text{if } x_i \in \Gamma_{h_{\min}}^0 \\ S(\phi_i^0)G(\phi)_i^n & \text{otherwise} \end{cases} \quad (2.44)$$

where $\Gamma_{h_{\min}}^0$ is a narrow band near the interface at the start of reinitialization. The sign function can also be employed instead of its smoothed version S . This update allows to perform reinitialization reducing surface movement and dealing with accumulation or zero gradient points. The latter can be treated by a robust evaluation of D , adding a small amount at denominator to avoid zero division. The redistancing iteration continue until convergence is reached, by meeting a certain threshold in variation between two consecutive steps or by reaching maximum number of iteration. To maintain a well behaved implicit function without performing excessive redistancing, the procedure is performed for a fixed number of times during level set integration.

2.1.5 Finite Volume on Octree grids

To perform the LS initialization and numerical solution on octree structures, it is easier to rely on Finite Volumes schemes [60], which allows to easily manage level jump for non uniform grids. The employed grid structure utilizes linear balanced octree adaptive mesh refinement, hence grid is locally refined following a certain criterion, with the constraint to have at most one level jump at each interface and indexing is built upon Morton's Z-ordering [61] with no information retained about the lower levels of the tree. Some examples of numerical problems solved with FV and adaptive octree grids are provided in [62, 63]. A 2D representation is presented in Figure 2.20 to visualize the general structure of the grid data and the ordering.

From the representation is easy to see how difficult can be to define a computational stencil to build finite difference schemes as the ones utilized on uniform grids, since the adaptive structure requires stencil adaptation. Finite Volumes schemes on

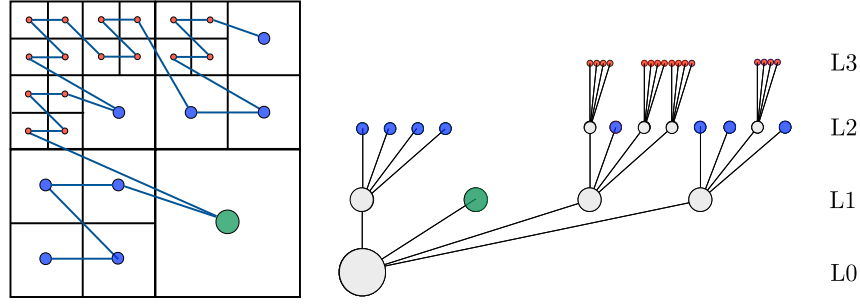


Fig. 2.20 Representation of hierarchical octree structure and Z-ordering. Every active level has a different color. Note the ordering reading the tree from left to right.

the other hand, allows to adapt to the changing morphology easily. To apply FV, heat method and the level set equation must be expressed in an integral form. Starting from Heat Method, the first step requires the computation of a small time step of an unsteady diffusion from an heat source, as described in 2.1. The computation of discrete Laplacian on octree grids is carried out, for the variable u as follows

$$\Delta u = \nabla \cdot (\nabla u)$$

$$\int_{\Omega} \Delta u dV = \int_{\partial\Omega} (\nabla u) \cdot \mathbf{n} dS \approx \sum_{f \in \partial\Omega} (\nabla u)_f \cdot \mathbf{n}_f |S_f| \quad (2.45)$$

The mid-point approximation is evaluated over each cell volume Ω by summing the contribution of all faces of the cell external surface $f \in \partial\Omega$. The symbol $\mathbf{n}_f$ denotes the outward normal to the face, $|S_f|$ is the face area, and $(\nabla u)_f$ is the gradient at the face. This yields the i -th cell-centered Laplacian

$$(\Delta u)_i \approx \frac{1}{|\Omega_i|} \sum_{f \in \partial\Omega_i} (\nabla u)_f \cdot \mathbf{n}_f |S_f| \quad (2.46)$$

Once the grid structures is defined, the computation of the Laplacian requires to evaluate the gradient at the interface. On an interface between to cells with the same level, one can compute gradient by evaluating directional derivative through centered finite difference, using information from the cells which share the interface. In this case, the computed quantity is already the normal component of the gradient. However, a locally refined grid has level jumps, where interface is subdivided. A proper way to compute gradient at interface level jump is obtained utilizing a Discrete Duality Finite Volume (DDFV) scheme, which are employed to reconstruct gradients

on non uniform grids by creating subgrid cells where one can evaluate quantities required to evaluate the gradient. In particular, for balanced octree grids a diamond stencil can be utilized, following the methods developed in [64, 65] and presented for quadtree/octree application in [66]. A visualization of the stencil is presented in Figure 2.21 for two dimensional domains

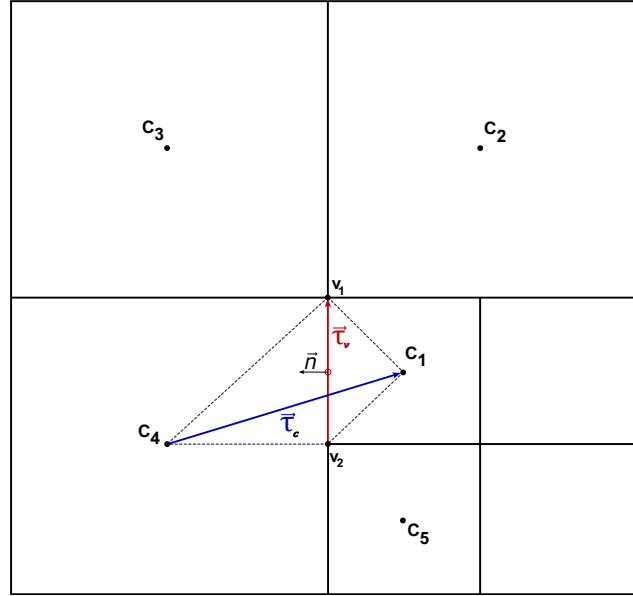


Fig. 2.21 2D scheme of diamond stencil

The stencil utilizes information from cell centers along the center connecting direction, evaluating gradient component in this direction, and information from the vertices along direction tangential to the interface itself. These components are projected along the normal to evaluate all the contributions and compute the gradient flux. While the data in the cell centers are available, the data in the vertex must be computed exploiting the information of the cells which share the evaluated vertex. Different methods can be used to combine these data, for example vertex value can be retrieved by an average of the data from the interested cells weighted by the distance from the vertex. After the construction of the substencil and the evaluation of vertices data, the value of the projection of the gradient along interface normal direction can be obtained starting from the computation of its projections along $\vec{\tau}_c$ and $\vec{\tau}_v$, i.e. the unit vectors defining the center connection direction and the direction tangential to the interface

$$\begin{aligned}\nabla\Phi \cdot \vec{\tau}_c &= \frac{\Phi_{c_1} - \Phi_{c_4}}{\Delta c} \\ \nabla\Phi \cdot \vec{\tau}_v &= \frac{\Phi_{v_1} - \Phi_{v_2}}{\Delta l}\end{aligned}\tag{2.47}$$

where Δc represents the distance between cell centers. The gradient can be decomposed as

$$\nabla\Phi = \nabla\Phi_1 \tilde{\mathbf{e}}_1 + \nabla\Phi_2 \tilde{\mathbf{e}}_2 = \nabla\Phi_1 \tilde{\mathbf{n}} + (\nabla\Phi \cdot \vec{\tau}_v) \tilde{\mathbf{e}}_2\tag{2.48}$$

where $\tilde{\mathbf{e}}_i$ are the unit vectors in the coordinate directions, formed by the normal $\tilde{\mathbf{n}}$ and tangential directions on the interface. Hence, choosing $\tilde{\mathbf{e}}_1 = \tilde{\mathbf{n}}$ yields to

$$\nabla\Phi \cdot \mathbf{n} = \nabla\Phi_1\tag{2.49}$$

Combining Eqs. (2.47) and (2.48), in particular by multiplying (2.48) by $\vec{\tau}_c$, the normal gradient at face j across a level transition is:

$$(\nabla\Phi)_j \cdot \mathbf{n}_j = \frac{1}{\vec{\tau}_c \cdot \mathbf{n}_j} \left(\frac{\Phi_{c_1} - \Phi_{c_4}}{\Delta c} - \frac{\Phi_{v_1} - \Phi_{v_2}}{\Delta l} \vec{\tau}_v \cdot \vec{\tau}_c \right)\tag{2.50}$$

The extension to three dimension follows the same concepts, but now the direction of contribution are three, with the two tangential directions on the interface and the center connecting direction, as shown in Figure 2.22. In a similar framework as

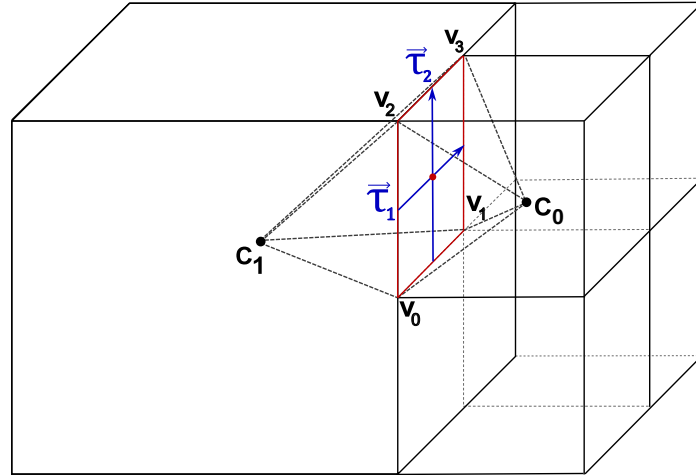


Fig. 2.22 3D scheme of diamond stencil

2D case, the gradient flux can be computed as

$$\begin{aligned}
\nabla\Phi \cdot \vec{\tau}_c &= \frac{\Phi_{c_0} - \Phi_{c_1}}{\Delta c} \\
\nabla\Phi \cdot \vec{\tau}_1 &= \frac{1}{2} \left(\frac{\Phi_{v_3} - \Phi_{v_1}}{\Delta l} + \frac{\Phi_{v_2} - \Phi_{v_0}}{\Delta l} \right) \\
\nabla\Phi \cdot \vec{\tau}_2 &= \frac{1}{2} \left(\frac{\Phi_{v_3} - \Phi_{v_2}}{\Delta l} + \frac{\Phi_{v_1} - \Phi_{v_0}}{\Delta l} \right)
\end{aligned} \tag{2.51}$$

where Δc is the distance between cell centers, Δl is the face size, $\vec{\tau}_c$ is the centers connecting direction, $\vec{\tau}_1, \vec{\tau}_2$ are tangential directions. In this case directional derivative along tangential direction are computed edge-to-edge, thus vertex evaluations are combined through an arithmetic average. Following the same approach as in the two-dimensional case, the normal gradient at face j across a level transition becomes

$$\begin{aligned}
(\nabla\Phi)_j \cdot \vec{n}_j &= \frac{1}{\vec{\tau}_c \cdot \vec{n}_j} \left[\frac{\Phi_{c_0} - \Phi_{c_1}}{\Delta c} + \right. \\
&\quad \left. - \frac{\vec{\tau}_1 \cdot \vec{\tau}_c}{2\Delta l} (\Phi_{v_3} - \Phi_{v_1} + \Phi_{v_2} - \Phi_{v_0}) \right. \\
&\quad \left. - \frac{\vec{\tau}_2 \cdot \vec{\tau}_c}{2\Delta l} (\Phi_{v_3} - \Phi_{v_2} + \Phi_{v_1} - \Phi_{v_0}) \right]
\end{aligned} \tag{2.52}$$

With this approach one can compute a second order approximation for gradient and approximate the conservative form of the diffusion equation. The solution is computed through the definition of the discrete laplacian matrix by entering the weight, for each cell, of the cell contributing to diamond and vertices stencils, used to accumulate gradient and evaluate cell divergence. For the specific problem of heat method first step, the matrix is multiplied by the time step Δt and summed to identity matrix. After the definition of initial source u_0 and the definition of boundary conditions as described in Heat Method section, one can solve the correspondent linear system to solve the implicit time step and evaluate u_t . The method follows with the second step, which requires the computation of the normalized gradient field. To compute normalization, one has to retrieve the information on gradient by newly computed u_t values at cell centers. There are different ways to reconstruct information for the gradient also in the cell center, however since the following step of the heat method requires the computation of the $\nabla \cdot \nabla u_t$, one can compute the interface value using the same diamond stencil applied before and use them to

compute fluxes and divergence approximation with the same mid-point rule applied before. With this computation, the right end side of the Poisson problem is ready and the third step of the heat method can be carried out using the same discrete laplacian matrix assembled before. As described before, Poisson problem can be solved up to a constant value, which is defined imposing a zero value of ϕ on the interface. The constraint is realized by penalizing correspondent GDL in the laplacian matrix by adding a very high value on the main diagonal to force the solution to the desired value.

To solve Level Set Equation through finite volume, one has to rewrite equation 2.1 in conservative form

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\vec{V} \phi) = \phi \nabla \cdot \vec{V} \quad (2.53)$$

and then apply integration on the domain

$$\frac{\partial}{\partial t} \int_{\Omega} \phi dV + \int_{\Omega} \nabla \cdot (\phi \vec{V}) dV = \int_{\Omega} \phi \nabla \cdot \vec{V} dV \quad (2.54)$$

In this form one can proceed to the finite volume numerical discretizations of the equations, which can be evaluated applying the same approach described before

$$\begin{aligned} \frac{\partial}{\partial t} \int_{\Omega} \phi dV &\approx \frac{d}{dt} (\bar{\phi}_i V_i) \\ \int_{\Omega} \nabla \cdot (\phi \vec{V}) dV &= \int_{\partial \Omega} \phi \vec{V} \cdot \vec{n} dS \approx \sum_{f \in \partial \Omega} (\phi \vec{V})_f \cdot \vec{n}_f |S_f| \\ \int_{\Omega} \phi \nabla \cdot \vec{V} dV &\approx \bar{\phi}_i \int_{\partial \Omega} \vec{V} \cdot \vec{n} dS \approx \bar{\phi}_i \sum_{f \in \partial \Omega} \vec{V}_f \cdot \vec{n}_f |S_f| \end{aligned} \quad (2.55)$$

where $\bar{\phi}_i$ is the mean value of implicit function on i -th cell. At this point, the computation requires the evaluation of both velocity fluxes $\vec{V}_f \cdot \vec{n}_f$ and conservative quantity $(\phi \vec{V})_f \cdot \vec{n}_f$ at cell interfaces. For the latter, a Local Lax-Friedrichs scheme, as the one described in Section 1.1, can be employed. Using this diffusive flux allows to introduce artificial viscosity and pick the right entropy condition when dealing with sonic point during normal advection of the surface

$$(\phi \vec{V})_f \cdot \vec{n}_f = \frac{1}{2} (\phi^+ + \phi^-) V_{n,f} - \frac{\alpha}{2} |V_{n,f}| (\phi^+ - \phi^-) \quad (2.56)$$

Where $V_{n,f}$ is the normal component of the velocity at the interface, α is the numerical diffusion coefficient of the scheme, and $\phi^\pm$ are the reconstructions of the scalar function on the left and right sides of the interface. The latter can be computed through different methodologies, for instance polynomial reconstruction. A second order polynomial reconstruction can be evaluated, starting from the i -th cell value, at the point $\vec{x}$, as

$$P^{R2}(\vec{x}) = \phi_i + (\vec{x} - \vec{x}_i)^T (\nabla \phi)_i \quad (2.57)$$

where the cell centered gradient $(\nabla \phi)_i$ can be evaluated through a quadratic least-square approximation, using the compact stencil of the i -th cell, formed by the neighboring cells. The source term requires computation of the velocity on the interface center, and this computation can be performed by an average of interface vertices values, the latter computed through the same weighted average on the sub-stencil formed by neighbors cells. After the computation of integral terms, the last step is the time integration of ϕ on the cells, performed by second order RK scheme.

$$\begin{aligned} k_1 &= -\frac{1}{V_i} \sum_f (\phi \vec{V})_f^n \cdot \vec{n}_f A_f \\ \bar{\phi}_i^{n+1/2} &= \bar{\phi}_i^n + \frac{\Delta t}{2} k_1 \\ k_2 &= -\frac{1}{V_i} \sum_f (\phi \vec{V})_f^{n+1/2} \cdot \vec{n}_f A_f \\ \bar{\phi}_i^{n+1} &= \bar{\phi}_i^n + \Delta t k_2 \end{aligned} \quad (2.58)$$

The surface transport requires to maintain local grid refinement through adaptive mesh refinement. Octree structure is redefined to perform refinement at new surface position, with an update of ϕ to evaluate its new values in the refined cell centers. This update is performed by extrapolating parent values when mesh is refined or coarsened. In the first case, eight new cells are created starting from parent cells, and values are updated by using the P^{R2} reconstruction defined in Equation 2.57, using the distance from the parent center to the new created cell center, for each of the eight cells created during the refinement. On the other hand, when eight cells are merged to coarse the grid in a single one, the new value at the cell center is simply computed as an average of the child cells which revert to the parent. Figure 2.23 resumes the procedure.

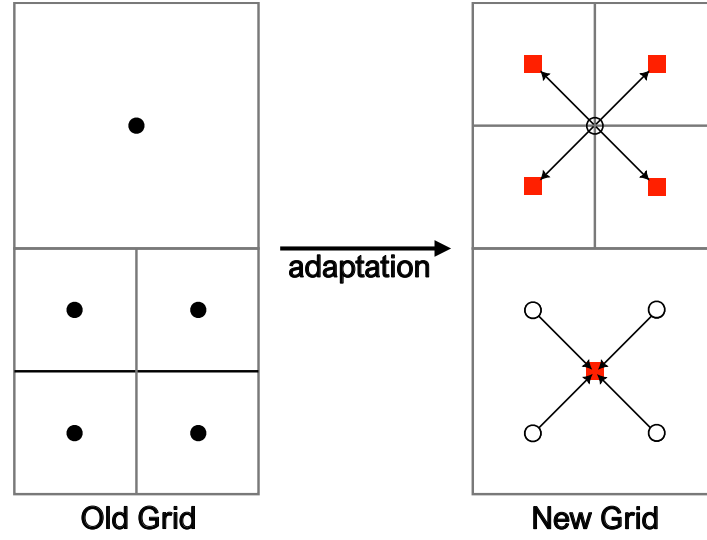


Fig. 2.23 Visualization of quadtree 2D grid adaptation. On octree grids the adaptation is performed on eight cell.

To perform redistancing on octree grids, the very same approach presented for cartesian structured grid is employed, using subcell fix method. The main difference in this case is the computation of lateral derivatives to evaluate during characteristic selection. To perform the derivative computation, the same discrete approach as the one used to compute gradient at cell interface is employed. Reinitialization is then performed for a fixed number of time to preserve distance field.

2.2 0D Chamber Model

Here the 0D chamber model, applied in [67], is reported. The model relies on mass conservation in combustion chamber and pressure is considered as only time-dependent. Pressure evolution is a consequence of mass balance between flow of produced gas $\dot{m}_b$ thanks to combustion processes and mass flow expelled through nozzle $\dot{m}_{th}$. A complete description of mass conservation and chamber equilibrium can be found in [2].

We recall the equation for burning mass flow rate in the case of uniform burning rate r

$$\dot{m}_b = \rho_b A_b r \quad (2.59)$$

The mass balance derives from conservation equation and can be written as

$$\dot{m}_b = \rho_b A_b r = \dot{m}_{th} + \frac{d(\rho_c V_c)}{dt} = \frac{p_c A_t}{c^*} + \frac{d(\rho_c V_c)}{dt} \quad (2.60)$$

If an ideal gas mixture is considered then the time derivative in Eq. 2.60 can be rewritten as

$$\frac{d(\rho_c V_c)}{dt} = V_c \frac{d\rho_c}{dt} + \rho_c \frac{dV_c}{dt} = \frac{V_c}{R_g T_c} \frac{dp_c}{dt} + \rho_c A_b r \quad (2.61)$$

Hence

$$r \rho_b A_b = \frac{p_c A_t}{c^*} + \frac{V_c}{R_g T_c} \frac{dp_c}{dt} + \rho_c A_b r \quad (2.62)$$

Considering the ideal rocket model the characteristic velocity can be evaluated as $c^* = \frac{\sqrt{R_g T_c}}{\Gamma}$. The complete definition of Γ is later reported in Equation 4.2. As a result, considering that $\rho_b \gg \rho_c$, the pressure derivative can be evaluated as

$$\frac{dp_c}{dt} = \frac{(c^* \Gamma)^2}{V_c} (\rho_b A_b r - \frac{p_c A_t}{c^*}) = R(p_c, Geometry) \quad (2.63)$$

This ordinary differential equation (ODE) can be integrated provided that burning area A_b and chamber volume V_c , are available. These variables are computed through the LS integration. From Eq. 2.63, the time derivative can be approximated implicitly as

$$\frac{dp_c}{dt} \approx \frac{p_c^{k+1} - p_c^k}{\Delta t} = R(p^{k+1}, Geometry) \quad (2.64)$$

The second term in the expansion can be expanded through a Taylor approximation, computing the Jacobian $\frac{\partial R}{\partial p_c}$, whose expression will be reported in the following. Here geometrical features are considered frozen at previous time step k . In a similar way, pressure is considered frozen during geometry integration through level set. The time step Δt is the same between LS and pressure integration. With these assumptions it is possible to write:

$$\frac{p_c^{k+1} - p_c^k}{\Delta t} = R(p^k, Geometry) + \frac{\partial R}{\partial p_c}(p_c^{k+1} - p_c^k) \quad (2.65)$$

Solving for p^{k+1} leads to

$$\begin{aligned} p_c^{k+1} &= p_c^k + \frac{R(p_c^k, Geometry)}{\beta} \\ \beta &= \frac{1}{\Delta t} - \frac{\partial R}{\partial p_c} \end{aligned} \quad (2.66)$$

Recalling the burning rate law $r = ap_c^n$. With this model, burning rate augmentation due to erosive burning, related to high gas speeds in the chamber, is neglected. While some correlation can be employed to link erosive burning to motor characteristic length and aspect ratio, an axially-resolved model should be employed to predict this effect with enough accuracy [68]. For solid grains which are particularly prone to erosive burning, i.e. those characterized by a low port-to-throat area ratio leading to high Mach numbers within the port, the employed model could underestimate the burning rate. Nonetheless, proposed model maintains sufficient degree of fidelity, reducing computational cost in order to exploit it during the evaluation of multiple configurations. Since r can potentially vary on the surface due to the non-uniform grain properties, the computation of the burned gas mass flow rate requires integration over the burning surface $\partial\Omega$. This variation is controlled through the variation of the coefficient a in the domain, while coefficient n is assumed constant. In general a combination of variable field could be assumed, but since limited data are available for the propellant, this simplification allows to reduce numerical complexity in data handling, but a variable field for n can be easily considered using the same approach as the one used for a . The zero level set, i.e. the burning surface, is reconstructed through a surface triangulation at every time step (Figure 2.3). Since the burning rate is defined on every point of the grid, it is possible to assign the value of the nearest point of the grid to the correspondent triangle barycenter of the surface triangulation. As a result, the burning mass flow rate is computed as

$$\dot{m}_b = \rho_b p_c^n \int_{\partial\Omega} a dA \approx \rho_b p_c^n \sum_{i=1}^N A_{b_i} a_i \quad (2.67)$$

where N is the number of triangles of the isosurface and propellant density and gas pressure are assumed to be uniform. This computation allows to evaluate the term R and its derivative with respect to pressure:

$$\begin{aligned} \frac{\partial R}{\partial p_c} &= \frac{(c_0 \Gamma)^2}{V_c} [(2m+n)p_c^{2m+n+1} \Lambda - \frac{A_t}{c_0} (m+1) p_c^m] \\ \Lambda &= \rho_b \sum_{i=1}^N A_{b_i} a_i \end{aligned} \quad (2.68)$$

Through this model is possible to compute pressure-time evolution as a consequence of grain burning.

2.3 Throat Erosion Model

The burnback analysis which arises from the coupling of LS surface evaluation and OD chamber model, allows to evaluate grain performance. The latter is quite affected from throat area A_t , which is the main driver of the discharge from motor chamber. Since the variation of this parameter can heavily alter the pressure evolution, it's useful to introduce a model to evaluate its modification during motor operation. In fact, to preserve nozzle throat from high gas temperatures, usually carbon-carbon inserts are employed. This protective material preserves the metal structure of the nozzle consuming itself with the interaction of hot gases produced by solid propellant combustion. As described in [69–71], ablation of the nozzle throat is mainly linked to chemical oxidation of carbon by oxidizing species in the hot gas mixture, in particular H_2O , OH and CO_2 are the main species to cause carbon insert oxidation at the typical temperature of $T = 3000 - 3500K$ reached in SRM hot gas exhaust [70], while oxidation from other species is not favored in this typical range of temperature or their concentration is simply too small to produce significant impact. A smaller contribution to ablation is produced by kinetic effects but, in this range of temperature, the chemical contribution is the highest factor to the consumption of the insert. This phenomenon allows to preserve the nozzle, but has an impact on the throat section dimension. In fact, as the insert is consumed, A_t grows. To evaluate this effect, a simplified model is here proposed. It is based on Bartz proportional

law, where the consumption of the insert is linked to a reference test case [72]. In particular, the proportionality is evaluated as

$$\dot{s} = \dot{s}_{\text{ref}} \left(\frac{p_c}{p_{c,\text{ref}}} \right)^{0.8} \left(\frac{R_{th,\text{ref}}}{R_{th}} \right)^{0.2} \quad (2.69)$$

where $\dot{s}$ is the throat erosion rate, hence the velocity at which the insert surface moves. $\dot{s}_{\text{ref}}$ is the reference erosion rate, p_c is the chamber pressure, $p_{c,\text{ref}}$ is the reference chamber pressure, R_{th} is the throat radius, and $R_{th,\text{ref}}$ is the reference throat radius. While this model is simple, it is important to correctly evaluate a proper reference for the analyzed insert, otherwise the results can be misleading. In the hypothesis of a carbon-carbon insert, working with HTPB-AP solid propellants, reference data have been selected from [70]. In particular, the reference values are $\dot{s}_{\text{ref}} = 3.942e - 4 \text{ m/s}$ and $R_{th,\text{ref}} = 2.75e - 2 \text{ m}$, for a $p_{c,\text{ref}} = 69e05 \text{ Pa}$ obtained by numerical computations tailored to match BATES motor experimental data [73]. The use of Bartz law allows to rapidly evaluate throat erosion rate and compute the evolution of throat area during motor operation by integrating this computation with the burnback solver. At every time step the new throat area is computed to properly evaluate chamber equilibrium and pressure evolution. After the new pressure has been computed, A_t is updated and surface is moved according to the chamber pressure.

2.4 Linear Viscoelastic Model

In this section, the numerical model used to perform structural analysis of solid rocket grain is presented [74–77]. The model employed requires to account for the viscoelastic properties of rubber like material. In fact, the particle loaded polybutadiene binder changes its elastic modulus in response to load and temperature history. A proper structural assessment of solid grain requires to demonstrate that the grain can withstand operational load after some defined load history, linked to the storage condition which the motor may face prior to its usage. The nonlinear viscoelastic behavior of the solid propellant produces a softening of the material, hence a reduction in its elastic modulus, when the material faces a temperature increase or an external load. A proper structural model has to face this material behavior. A common choice used for this purpose is to employ linear viscoelastic theory [78] to characterize the

material for a successive Finite Element analysis. This theory requires that materials properties are homogeneous and that superposition principle is valid. With this assumptions, the stress can be evaluated using the following convolution integral

$$\sigma(t) = \int_0^t E(t - \tau) \frac{d\varepsilon}{d\tau} d\tau \quad (2.70)$$

where $E(t)$ is the *relaxation modulus*, hence the time dependent coefficient that allows to evaluate stress in response of a certain deformation. The integral sum up all the contribution to the stress field by the summation of successive deformation, accounting for the variation in the response of the material. E_t is characterized by uniaxial constant strain tensile tests. To evaluate the effect of the temperature on the relaxation modulus a common choice is to consider the material as thermorheologically simple, hence temperature effect can be accounted for with a shifting in time response variation of the material. In other words, the effect of the temperature is equivalent to a longer, or shorter, application of a deformation in terms of material relaxation. With this approach, the convolution integral is modified as follows

$$\sigma(\xi) = \int_0^\xi E(\xi - \xi') \frac{d\varepsilon}{d\xi'} d\xi' \quad (2.71)$$

where, for an isothermal loading, $\xi = t/a_T$ is the reduced time and a_T is a time-temperature shift factor, which is a function of temperature. To evaluate $E(\xi)$ different tensile test are conducted at different temperature level. A global master curve for relaxation modulus is constructed by shifting each evaluation at a specific temperature by the a_T factor, that can be computed with the Williams-Landel-Ferry (WLF) function as follows

$$\log a_T = \frac{-C_1(T - T_0)}{C_2 + T - T_0} \quad (2.72)$$

where T_0 is the reference temperature, i.e. the temperature where $a_T = 1$ and C_1, C_2 are the shifting coefficient of the material. The master curve can be approximated through a Prony series, in order to allow for numerical implementation and coupling with FEM solvers. Prony approximation can be constructed as follows

$$E(\xi) = E_\infty + \sum_{i=1}^N E_i \exp\left(-\frac{\xi}{\lambda_i}\right) \quad (2.73)$$

and can be evaluated from N discrete evaluation for the same number of reduced time $\lambda_i = (t/a_T)_i$ and with asymptotic relaxation value E_∞ . The exponential fit for relaxation data allows to have a continuous representation for the viscoelastic behavior of the material. With a proper definition of the material coefficient for the model, the use of Prony approximation with WLF shift is widely accepted to characterize linear viscoelasticity and run FEM simulations. The latter have been performed on Ansys Mechanical commercial software, employed to reproduce the thermomechanical response of the propellant material under representative conditions. The geometry of the system, consisting of a cylindrical propellant grain enclosed within a metallic case, naturally lends itself to an axisymmetric numerical representation. Thermal insulation has been neglected in the definition of the geometry, for the sake of simplicity during a preliminary design phase, since the biggest constraint to grain movement is the highly resistant metal of the casing. This assumptions reduce the computational cost of the analysis without affecting the accuracy of the stress and strain evaluation. The metallic casing is modeled as a linear elastic body, while the propellant domain is assigned the Prony-based linear viscoelastic constitutive behavior just described. The interface between the two components is treated as perfectly bonded, in accordance with the manufacturing process and the effective adhesion between propellant and case. The analysis is formulated as a transient thermostructural problem, since for solid rocket motors the structural integrity during service is strongly influenced by the thermal history experienced after curing. Once the grain completes polymerization, it is effectively stress-free at a curing reference temperature. Subsequent cooling toward the operational thermal environment induces non-negligible residual stresses due to the constrained thermal contraction of the viscoelastic matrix. Over time, these stresses evolve and accumulate under the superimposed effects of temperature-dependent material relaxation of viscoelastic solids. Such effects cannot be reproduced through steady-state analysis and therefore require a time-dependent formulation consistent with the reduced-time framework introduced earlier. Several operational loading conditions may affect the grain. The main one are pressurization internal loads, thermal induced loads, dynamic loads such as vibrations and inertial loads and operation related loads, like aerodynamic heating during fly or transport. In this preliminary design phase only the internal pressurization and pre-operational thermal loads are considered. This simplification allows focusing on the dominant contribution to the maximum stress in the propellant, while maintaining a computationally tractable model and isolating the influence of

the viscoelastic response on structural margins. Moreover, since operational loadings requires a deeper knowledge in motor operation and usage, this considerations exceed the scope of this work. Default solver parameters and convergence tolerances for transient viscoelastic simulations are adopted, as they are sufficient to ensure numerical stability and physically meaningful evolution of the stress field across the simulated time window. Post-processing of the numerical results concentrates on the evaluation of the maximum von Mises stress within the propellant domain during the entire thermal and mechanical cycle. This output provides an indication of the most critical regions inside the grain and allows assessment of the material capability to withstand the operational conditions once viscoelastic relaxation, strain accumulation, and residual stress development are taken into account. Different failure criteria can be utilized to validate the selected geometry, an overall description can be found in [74]. In this work the effective stress is taken into account to evaluate the complex triaxial stress field which arises due to complex shapes of the grains, and compared with ultimate compressive stress for the material, eventually corrected with knockout factors. In this work, this numerical structural assessment can give an initial indication whether non-uniform properties can be exploited to obtain solid motors with non conventional geometries, which can experience less stress accumulation compared to standard geometries with the same performances.

Chapter 3

Initial Testing

In this section, initial testing of the software is presented. Initially, testing is carried on area computation. Some basic shapes are generated and evaluated through analytic initialization and surface computation with level set method and both area computation method presented. The different schemes described in the previous chapter are compared to assess the accuracy achieved for area computation, with different grid resolution, to verify the consistency of the schemes. After that, more complex shapes are evaluated. The initial assessment started with the evaluation of a sphere inflation under normal movement, to evaluate for a simple test case the area computation and the performance of different numerical methods and with both finite difference computation on cartesian grid and finite volume on adaptive octree grids. Later, a star shaped grain is used as reference for its non trivial shape and the availability of an analytic formulation to compute burning surface evolution. Here the level set numerical solution is evaluated on a significant geometry, but without the use of non-uniform burning rates. Further evaluations are carried out on finocyl geometries, in particular a confrontation with experimental firing test data for small solid grain. Finally, the evaluation of the ability of the tool to evaluate non-uniform burning rates is carried out by applying the variable ballistic properties to the same finocyl geometry previously tested and by identifying the ballistic configuration which alters the pressure response of a cylindrical grain to produce a neutral burning and a dual-thrust solid grain.

3.1 Area computation error

Initial tests have been carried out on a simple sphere to evaluate the accuracy of area computation for a normal transport problem. Area evolution is computed throughout sphere inflation with both triangulation and smoothed Dirac delta integration for finite differences on Cartesian grids, while only the second method is employed to evaluate area for normal movement computed with finite volume on octree grids. For the latter, dynamic refinement is performed to follow the surface as it moves outward, maintaining the local refinement level. For both cases, reinitialization is performed at every time step until a threshold on unitary gradient value is achieved, in order to obtain accurate results, especially for Dirac area computation. Simulations are performed for a fixed time length for finite difference with a unit sphere radius, while a fixed number of time steps is defined for finite volume computation with a unitary sphere radius of two. Figures 3.1 and 3.2 show the initial conditions for both cases. Results obtained for this simple shape show similar performance for both tested methods, and they are summarized in Figures 3.3 and 3.4

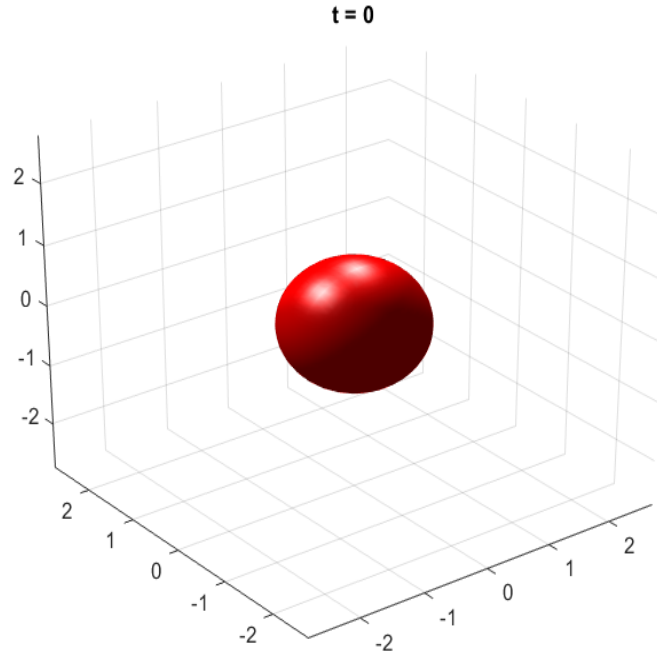


Fig. 3.1 Visualization of test sphere for cartesian grid

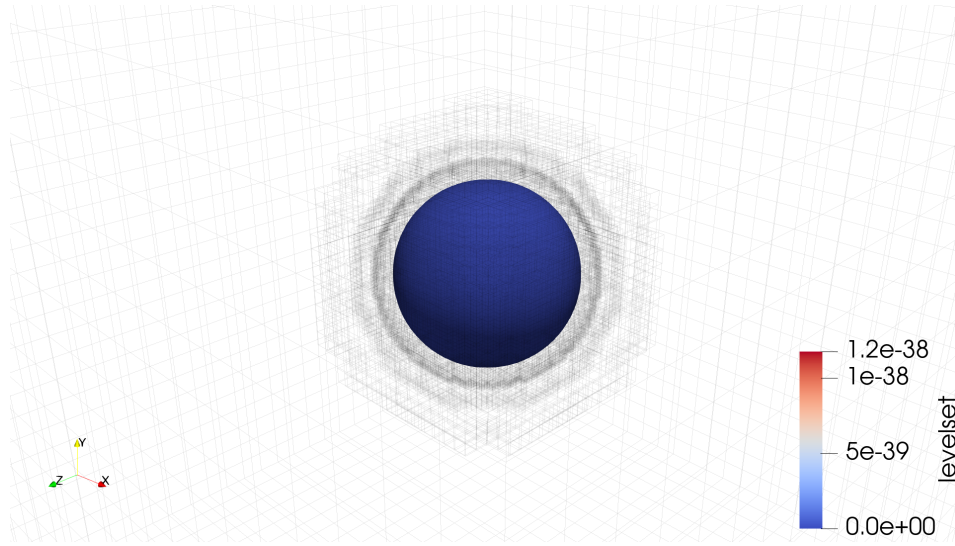


Fig. 3.2 Visualization of test sphere for octree grid

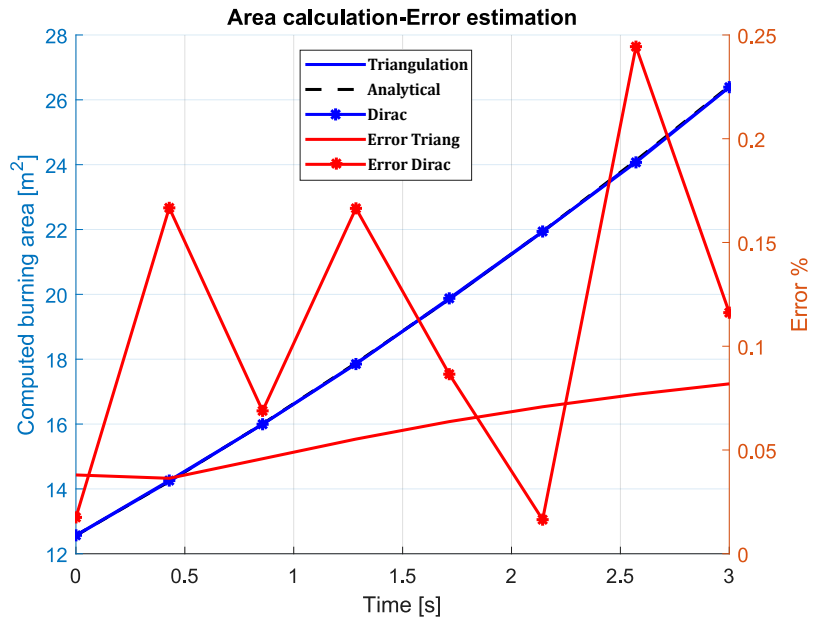


Fig. 3.3 Area computation maximum error for sphere, computed with TVDRK-ENO at second order

For the finite difference case, both triangle summation and Dirac delta integration have been used to compute burning area, while for finite volumes only the latter has been employed. In every tested configuration, the percentage error in area computation, i.e. the absolute difference between numerical and analytical evaluation of the area divided by the latter, remains very small, with the exception of the first steps

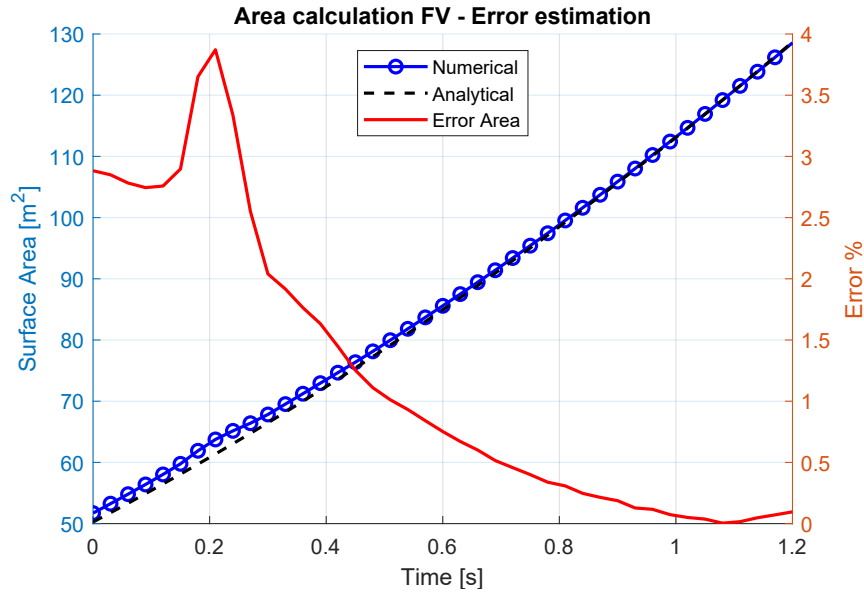


Fig. 3.4 Area computation maximum error for sphere, computed with Finite Volume on adaptive octree grids

of the finite volume computation, where the maximum value of area error reaches almost four percent. While this transient peak is likely linked to local grid refinement behavior during topological changes rather than representing the steady-state error level, its impact must be critically assessed since, given the nonlinear dependence of chamber pressure on burning surface, surface errors are amplified according to the proportionality $p_c \propto A_b^{1/(1-n)}$. Using the specific value of $n = 0.32$ adopted in this preliminary evaluation, a 4% peak surface error propagates to a 6% error in chamber pressure. However, for higher values of n typical of composite propellants ($n \approx 0.5 - 0.6$), the amplification factor could increase significantly, potentially pushing the pressure error toward 10%. Consequently, when operating with highly pressure-sensitive formulations, estimating a maximum admissible surface error as a function of n is crucial, and finer grid refinements must be employed to strictly contain the propagated error. Following these initial, independent verification tests of both finite difference schemes on uniform Cartesian grids and FV transport on adaptive octree grids, the finite difference approach was selected to carry out all successive computations. This choice was driven by the need for a robust trade-off between accuracy, computational stability, and software integration during the subsequent optimization phase. Even though a uniform Cartesian grid increases the computational cost to resolve fine geometric features compared to local adaptive re-

finement, it avoids the complexities and transient noise associated with managing the dynamic refinement near the zero-level interface. Moreover, the software framework utilized for the finite difference implementation allows for a more straightforward coupling with CAD geometry creation, as well as with the optimization and surrogate modeling libraries later employed. Because the finite difference implementation provides sufficient and stable accuracy with a reasonable computational effort, it was chosen as the primary method for the subsequent evaluations. Further work on FV and octree grids can be foreseen to improve computational efficiency for high-fidelity evaluations, provided proper exploitation of adaptive mesh refinement. The selected numerical approximation was then tested on grain geometries and coupled with the 0D chamber model to perform the burnback simulations.

3.2 Surface transport test, star shaped grain

First tests carried out on significant geometries faced the analysis of a six points star shaped grain. This complex geometry, quite used in solid rocket design, allows to test LS performance on complex shapes with particular features, in particular the star points which disappear during propellant consumption that allow to have a neutral burning. It is generated by an analytical formulation, with the implicit formulation of a cylinder intersected by hyperplanes. A visualization of the obtained inner cavity geometry is presented in Figure 3.5.

Moreover, the burning surface evolution for a star-shaped grain, can be evaluated through an analytical formulation, providing a proper parametrization, using the following relations from [79]

$$S_{\text{adim}} = \begin{cases} 2n \left[\frac{\sin(\varepsilon\pi/n)}{\sin\theta} + \frac{y}{l} \left(\frac{\pi}{2} + \frac{\pi}{n} - \theta - \cot\theta \right) + (1 - \varepsilon)\frac{\pi}{n} \right] & \text{if } \frac{y}{l} \leq \frac{\sin(\varepsilon\pi/n)}{\cos\theta} \\ 2n \left[\frac{y}{l} \left(\frac{\pi}{n} + \arcsin\left(\frac{l}{y} \sin(\varepsilon\pi/n)\right) \right) + (1 - \varepsilon)\frac{\pi}{n} \right] & \text{otherwise} \end{cases}$$

$$A = P \cdot L = S_{\text{adim}} \cdot l \cdot L \quad (3.1)$$

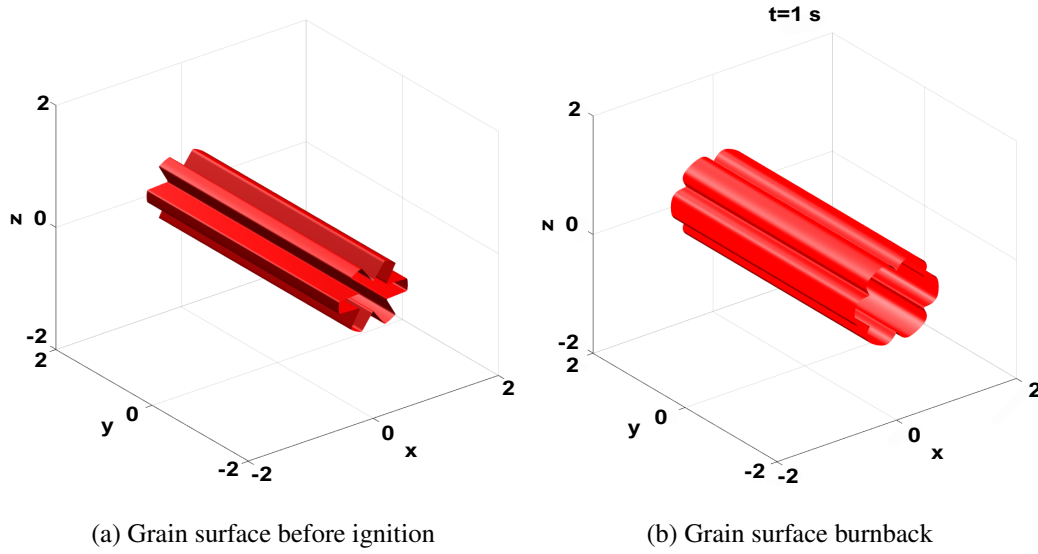


Fig. 3.5 Burnback simulation of star-shaped grain after 1 second

where n represents the number of star points, ε is the grain shape parameter (star point fraction), θ is star point opening angle, y indicates the radial burn depth, l is the radius of the grain, L represents the axial length of the grain, S_{adim} is the dimensionless perimeter, P is the burning perimeter, and A is the burning area. Through this definition one can evaluate area computation error in the same way presented in the previous section. The computation is performed for normal transport of the star shaped grain at constant burning velocity. Different grid dimension and numerical schemes have been evaluated to compare the results for the computation of a complex shape. An overall comparison is presented in Figure 3.6, while results for second and fifth order schemes are presented in Figure 3.7 and 3.8. When area is computed with triangulation, maximum error is present at the start of the computation, when dealing with the reconstruction of sharp features like star points, which can cause sub-sampling errors if the grid is too coarse near these features. However, the error is confined under 2.5 percent and it reduces when points disappear after burning. The area is almost neutral as expected and the burning becomes progressive when fins are consumed, until propellant depletion. For area computed with smoothed Dirac delta volume integral, initial computation is more accurate but the error increases up to five percent, even with reinitialization. The test shows how sharp feature requires fine grids to avoid sub-sampling, but at the same time that the error remains confined and limited. The comparison between different numerical schemes shows how, for refined grids, the reduction in mean

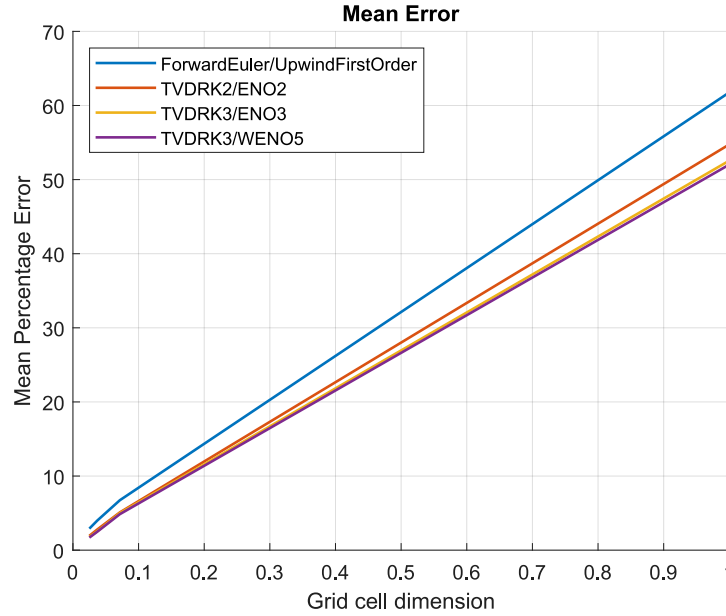


Fig. 3.6 Mean error over normal transport for different schemes varying grid dimension

error is consistent switching from first to second order, while higher orders show diminishing returns in investing higher computational effort, since error reduction in area computation is negligible. For this reason, second order TVDRK-ENO schemes are selected for later computations, since they give the best trade off value between accuracy and computational time, provided a grid which is sufficiently refined to solve the sharpest features of the geometry. Following area estimation, a rescaled star geometry, sharing the same topology and ballistic coefficients, is used to verify the coupling with 0D chamber model. The pressure integration is performed after each area update of the LS integration. The ignition transient is neglected and the first pressure value, which starts the area computation defining initial burning rate r , is computed by evaluating equilibrium pressure inside the chamber with the initial burning area of the star. From this equilibrium pressure, surface burns and change its shape and burned mass flow, altering chamber equilibrium and evolving pressure in the chamber. The burning is stopped at a certain casing radius r_{casing} , after which burning velocity is null, and surface movement stops. This causes a coalescence at the radius, which in turns generates a jump in the implicit function ϕ . The discontinuity can cause errors in Dirac area computation, since the smoothing at the interface can't deal with such a jump, and even frequent reinitialization don't seem to reduce the errors. The computation is carried out with direct integration of

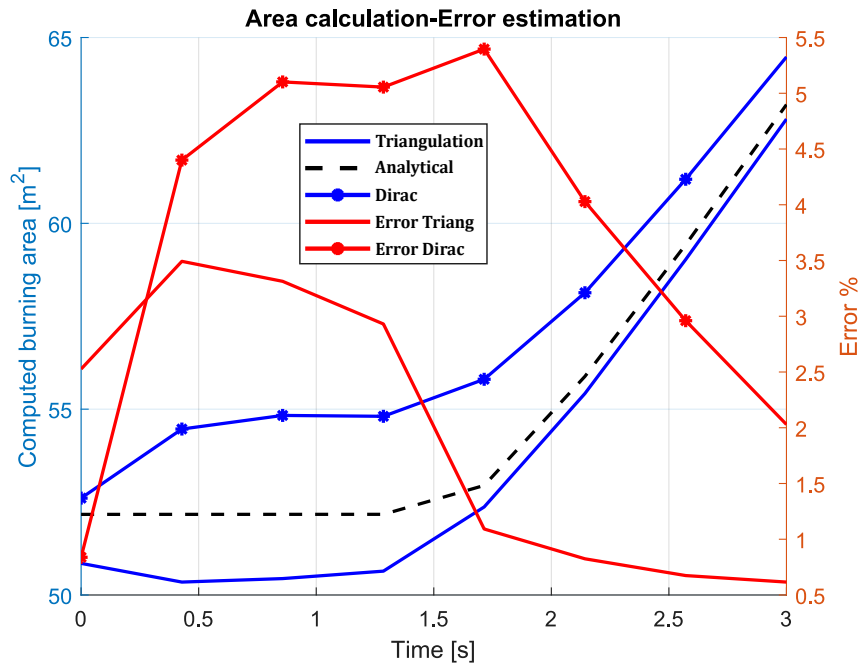


Fig. 3.7 Area computation mean error for star shaped grain burnback, computed with TVDRK-ENO at second order

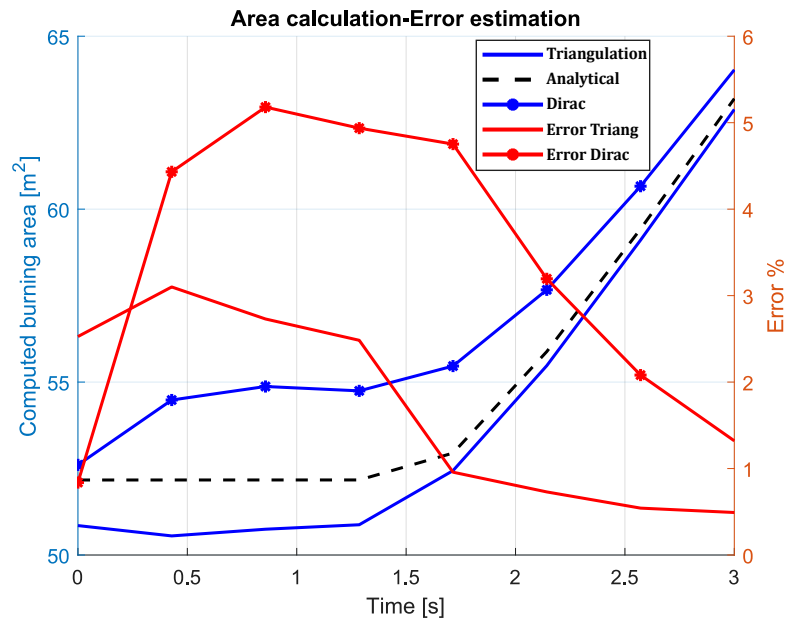


Fig. 3.8 Area computation mean error for star shaped grain burnback, computed with TVDRK-WENO at fifth order in space and third order on time

surface triangulation, and the arrest condition at the grain end is imposed by turning

off the summation of triangles whose barycenter lies within a specified threshold from the casing. A constant threshold allows the definition of a tubular grain casing, while an axis-dependent threshold can be employed to modify the casing shape along the grain length. Although some testing has been performed on dome-shaped end sections, the results presented here focus on tubular grains. This choice simplifies the analysis, while the implemented routine remains general and is capable of handling alternative casing shapes. Similarly, throat area erosion and its computation through the proposed model (Section 2.1) are neglected in the present analysis due to the lack of reliable reference data. Nonetheless, the implemented routine is general and capable of handling throat erosion effects, and this simplification does not affect the generality of the procedure. The results obtained are shown in Figure 3.9. As stated before, the pressure simulation was performed on a geometrically rescaled configuration with an extended length and an increased casing threshold, hence a greater web thickness, introduced to achieve a more realistic absolute burning time and operating pressure level for a typical SRM, while retaining the same ballistic coefficients and the same grain topology. This accounts for the time scale difference with respect to Figs. 3.7 and 3.8. The behavior of the pressure-time curve reflects, as expected, the area behavior and shows the initial neutral burning followed by an increase in pressure when the star tips are consumed and the burning area starts to increase. This trend continues until the front reaches the motor casing, resulting in a strong reduction in chamber pressure until burn-out during the tail-off phase. Some small oscillations can be seen in the neutral phase, due to area reconstruction errors which generate pressure fluctuations. If the computed area is not perfectly neutral, the equilibrium pressure varies, but the stabilizing effect of having a lower than one n exponent for the $r = ap^n$ relation gives a pressure which oscillates around the expected value. It is worth noting that the area computation performed on the same star grain topology using the same numerical tools, reported in Figs. 3.7 and 3.8, does not exhibit oscillatory behavior. This confirms that the oscillations visible in the pressure curve originate from the sensitivity of the quasi-steady pressure balance to small perturbations in the reconstructed burning surface area, rather than from numerical instability in the 0D integration itself. When the area effectively changes during the progressive and regressive phases, this effect is no longer visible.

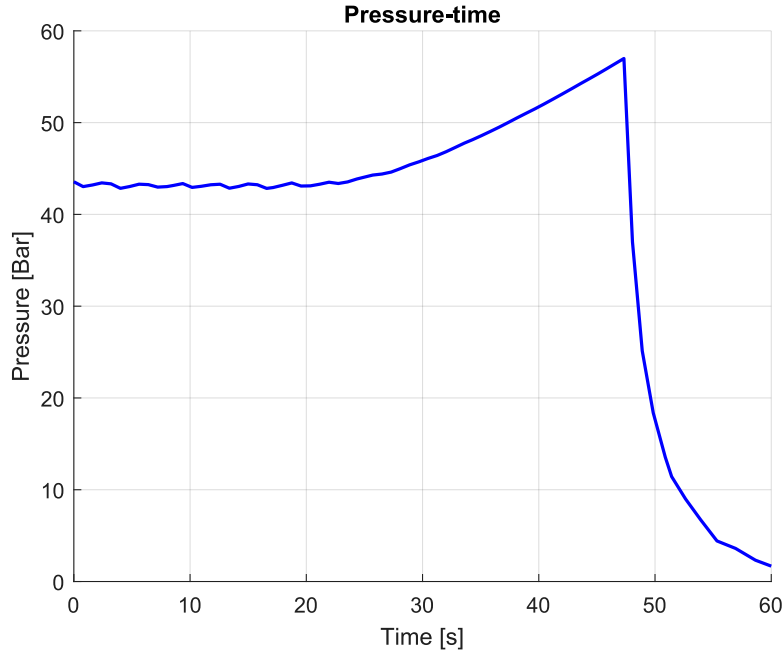


Fig. 3.9 Pressure evaluation of the star shaped test case

3.3 Test on experimental firing tests

After initial testing, an initial application of the burnback simulation tool is performed on experimental data. The reference pressure-time curves and geometries, taken from [80], describe the ballistics of a finocyl grain. Further numerical evaluations for the same geometries, proposed in [81], are also considered for comparison. In this case, the complex shape of the grain requires defining the geometry using direct SDF computation. The inner cavity burning area is reconstructed in CAD software, triangulated, and exported to the burnback environment where the implicit function ϕ is defined. Among the configurations studied, an 8-fin configuration has been selected as reference. As reported in the original experimental study [80], the firing tests experienced an imperfect inhibition of the grain's front face, causing the physical setup to behave as a front-burning configuration rather than a fully inhibited one. To rigorously validate the framework against both the intended theoretical design and the actual experimental outcome, two separate simulations were conducted. The inhibited simulation assumes perfect front-face inhibition and aligns with the numerical reference [81], which modeled the nominal theoretical design. The front-burning simulation accounts for the additional burning surface exposed by the

imperfect inhibition and matches the experimental firing data from [80]. Rather than representing an inconsistency between the two references, this comparison demonstrates the solver's capability to accurately capture pressure evolution under different geometric boundary conditions. Figure 2.3 presented in SDF computation section shows the front-end burning finocyl shape analyzed, while Figure 3.10 shows the zero level of the implicit function after ϕ initialization, for the inhibited finocyl geometry. Figure 3.11 shows results of burnback simulation compared with experimental data and numerical references, showing good accordance for the two considered instances. Some differences arise with the comparison with experimental data during tail-off phase. This can be explained with a different behavior during ignition and flame spreading phase, which is neglected in the simulation. The pressure build up in reality is slower and the grain chamber reaches the same pressure level as the one computed at equilibrium by the burnback solver later. Hence, the simulation evaluates a faster first part of the burning at higher pressure level, which reflects in a faster depletion of the propellant and an anticipated tail-off. The greater the ignition and flame spreading transient, the greater the expected difference in tail-off phase. It's important to consider that methods to accelerate ignition transient are usually applied in real motor operation, and this effect is greatly reduced in these cases.

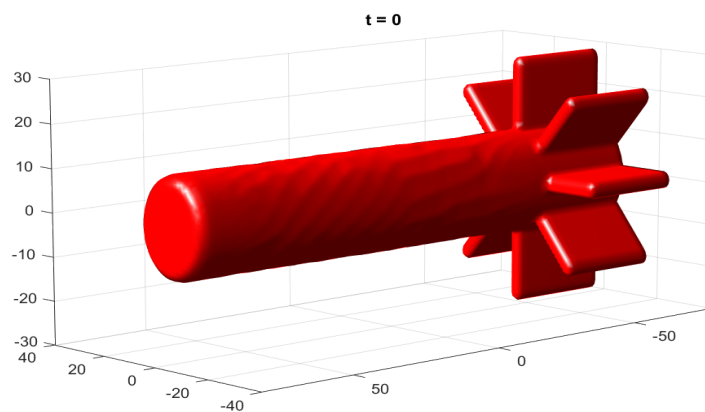


Fig. 3.10 Eight fins inhibited front-end finocyl geometry used for testing. The figure shows the inner cavity of the grain at $t = 0$

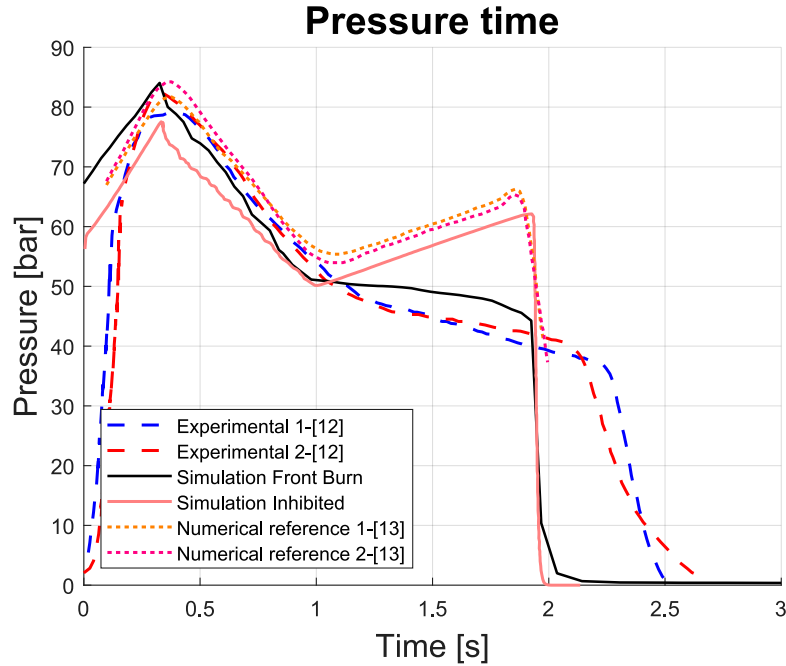


Fig. 3.11 Results of burnback simulation of finocyl geometries. Comparison with experimental and numerical references, for both front burning cases

3.4 Test with non-uniform ballistic

Following the pressure integration tests, the work focused on assessing the numerical tool's ability to simulate grains with nonuniform properties. This is useful to exploit the potential of innovative manufacturing processes, such as additive manufacturing, which allow for controlled variations in propellant composition.

Nonuniform ballistic properties were modeled by assuming a spatially varying coefficient a in the burning rate equation. The pressure exponent n was assumed constant as a modeling simplification to limit the design space dimensionality, with a value chosen lower than one to ensure combustion stability. In the initial methodology, a is described as a function of the radial coordinate R , the distance from the motor axis. Defining R_c as the absolute radial position of the casing measured from the grain main axis, the following polynomial law was selected to govern the distribution

$$a(R) = \frac{a_0 - a_c}{R_c^b} (R_c - R)^b + a_c \quad (3.2)$$

Where a_0 and a_c are fixed values of the pre-exponential coefficient at motor axis and motor casing respectively, and b is the exponent of the ballistic law. Altering these three parameters allows to effectively control the variation of a along the radius, while the distribution is constant along motor axis. In this specific test case, the ballistic property variation is therefore strictly one-dimensional in the radial direction. The framework is capable of handling multi-dimensional distributions, as demonstrated in subsequent case studies. The resulting distribution is plotted with respect to the absolute radial coordinate R for various exponent b values in Figure 3.12, using fixed $a_c = 4e - 05$ and $a_0 = 7.25e - 05$. These coefficients were tuned to induce significant variations in ballistic properties, while maintaining plausible data according to literature [80]

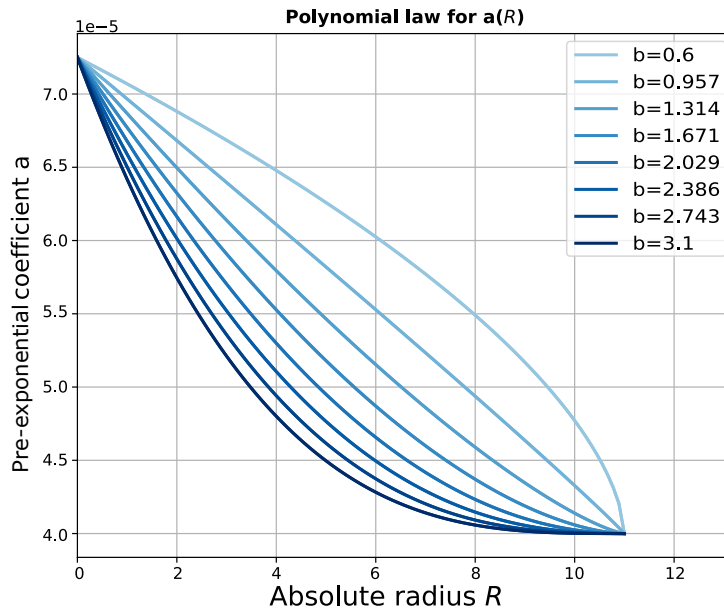


Fig. 3.12 Radial distribution of pre-exponential coefficient a as a function of b . Coefficient $a_0 = 7.25e - 05$ and $a_c = 4e - 05$.

This distribution model was applied to the finocyl geometry previously studied, as shown in Figure 3.13, with results detailed in Figure 3.14. Reducing the initial reaction rate, i.e. a_0 value, decreases the mass produced during burning, so the combustion chamber pressure is lower compared to the uniform case. Also, the lower chamber pressure combined with a slowing burning rate due to the ballistic property variation leads to a prolonged motor operation time. These findings show the tool

can handle nonuniform ballistic properties and highlight how the performance of a fixed geometry can be significantly altered using additive manufacturing techniques

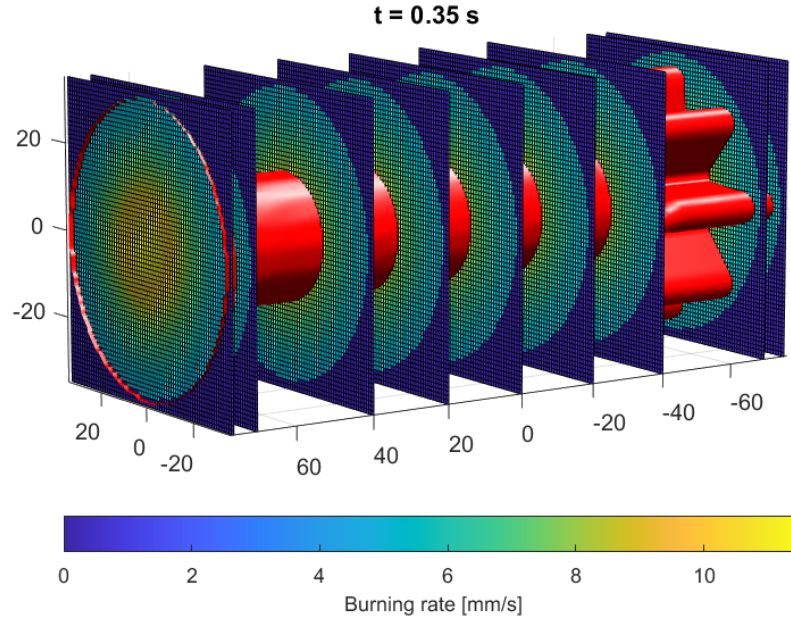


Fig. 3.13 Visualization of ballistic properties distribution during burnback of front burning finocyl

After these preliminary tests, other evaluations have been carried out. A primary goal was to achieve a cylindrical grain with neutral burning behavior. To satisfy the neutral burning condition, one has to meet the null condition for the chamber pressure time derivative, hence

$$\frac{dp_c}{dt} = \frac{(c^*\Gamma)^2}{V_c} \left(\rho_b A_b r - \frac{p_c A_t}{c^*} \right) = 0 \quad (3.3)$$

This implies that the burning rate must vary to exactly compensate for the growing burning area. Here, R represents again the absolute radial coordinate. By rearranging the equation and using the established relation for pressure derivative, the required distribution for a is:

$$a(R) = \frac{A_t}{c_0} \frac{p_c^{1-m-n}}{\rho_b 2\pi L} \frac{1}{R} \quad (3.4)$$

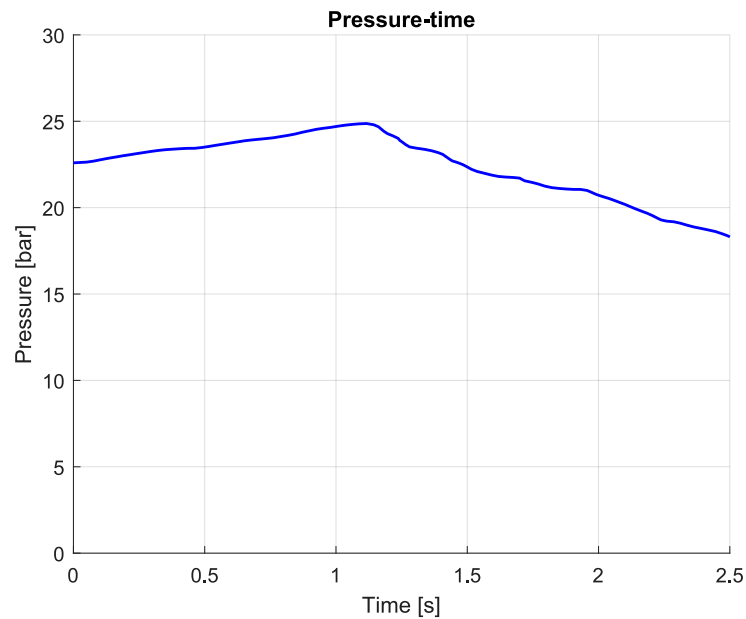


Fig. 3.14 Modified pressure time performance for non-uniform finocyl grain

A hyperbolic distribution of a is therefore necessary to obtain neutral burning at a specifically chosen chamber pressure for a cylindrical grain. Unlike standard cases where equilibrium pressure results from a given geometry and propellant properties, this inverse problem treats the chamber pressure as a design constraint to determine the required property distribution. The results in Figure 3.15 confirm the overall desired behavior is achieved. However, numerical errors in computing the burning area induce oscillations in the pressure history. These oscillations are amplified because the analytical a distribution provides neutral behavior for the exact theoretical burning area, not for the numerical area affected by numerical error. The errors force the solution away from desired pressure level, but the integration is later stabilized, in a similar manner as described for star shaped grain test case in 3.2. Specifically, a small numerical error in the discrete area evaluation perturbs the chamber pressure, which in turn alters the local burning rate. Since the hyperbolic $a(R)$ distribution is tailored to the exact theoretical area, any spatial deviation of the burning front triggers a restoring pressure effect, generating the observed oscillations. Future developments should address this sensitivity through tighter time-stepping or improved geometrical accuracy in the area computation, to reduce the entity of the oscillations before their stabilization.

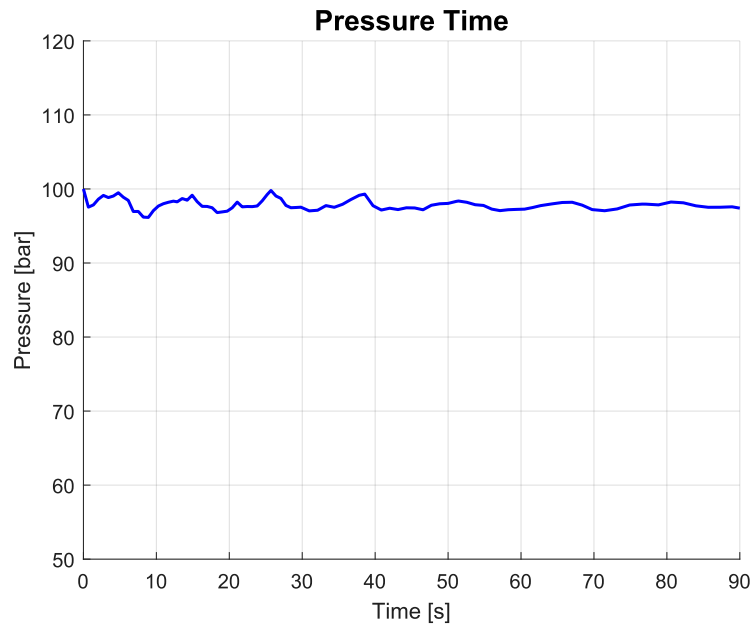


Fig. 3.15 Pressure Time for cylindrical shape modified for neutral burning

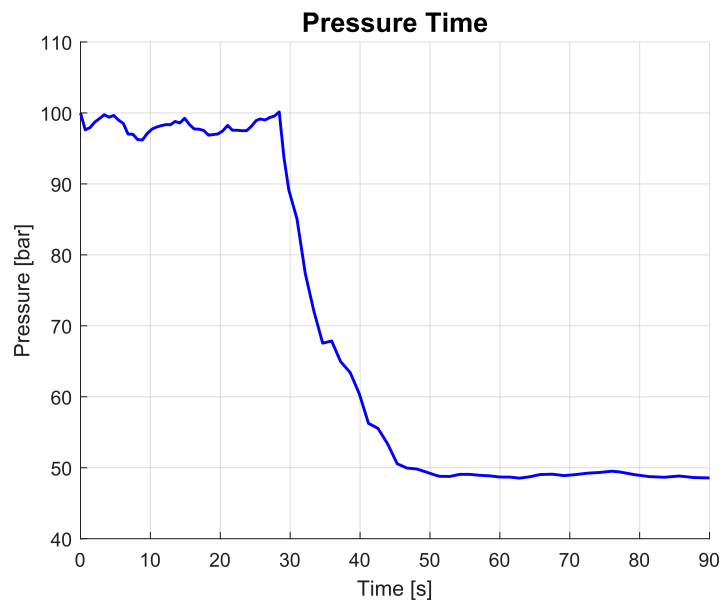


Fig. 3.16 Pressure time of a cylindrical shape modified for dual thrust, between 100 bar and 50 bar

A final test designed a dual-thrust cylindrical grain. This involved defining piece-wise hyperbolic functions for the coefficient a , effectively dividing the grain behavior into two sections separated by a specific radial coordinate, R_{shift} , where the pressure shift occurs. The analytical form remains consistent with the neutral burning case, but the target constant pressure changes between the two sections

$$a(R) = \begin{cases} \frac{A_t}{c_0} \frac{p_{c1}^{1-m-n}}{\rho_b 2\pi L} \frac{1}{R} & \text{for } R < R_{\text{shift}} \\ \frac{A_t}{c_0} \frac{p_{c2}^{1-m-n}}{\rho_b 2\pi L} \frac{1}{R} & \text{for } R \geq R_{\text{shift}} \end{cases} \quad (3.5)$$

The results in Figure 3.16 show limitations similar to the neutral burning case regarding numerical oscillations, especially during the first phase before the shift. Despite the oscillations, whose amplitude requires further reduction to achieve the precision needed for quantitative design, the results demonstrate the core concept of a spatially tuning of the ballistic distribution used to command macroscopic variations in motor performance, purely through material property gradients. It is worth noting that these test cases represent a fundamentally different design paradigm from uniform ballistic configurations. By spatially tuning the $a(R)$ distribution, the pressure evolution is actively decoupled from the geometric surface progression. This decoupling is precisely the additional design degree of freedom that the present framework is built to exploit, enabling preliminary performance analysis and design of pressure-time profiles that are no longer constrained by geometric evolution alone, significantly expanding the available design space. With these obtained results, in the next chapters an automated design procedure is presented to design solid propellant grains, leveraging the non-uniform distribution as a new design parameter, in order to enhance design flexibility for this type of rocket engines, surpassing limitation of traditional design and manufacturing approach.

Chapter 4

SRM Design

Having established the LS-based burnback simulation capability and validated it against analytical and experimental benchmarks, attention now turns to exploiting this tool within an inverse design framework. The problem of designing solid rocket motors requires a careful approach to define a geometrical configuration able to achieve desired performances while complying with numerous ballistic, propellant, and envelope constraints. Several works in the literature have addressed inverse design and optimization of solid propellant grains using different strategies[75, 82–88]. However, these approaches generally assume uniform ballistic properties within the grain and do not explicitly account for spatially varying reaction rates as part of the design process. Even though motor performance is mainly determined by the propellant's geometrical and ballistic features, identifying suitable combinations to meet requirements can be challenging due to manufacturing constraints that severely limit attainable grain geometries. In this chapter, a general overview of traditional solid rocket motor design is presented, followed by modifications introduced by accounting for the enhanced flexibility achievable with additive manufacturing concepts. The overall procedure is supported by the creation of surrogate models, built to speed up burnback computation and performance evaluation to support a stochastic optimization framework, the latter described in the corresponding section. The chapter also contains applications on inverse design problems, where the optimization framework is applied, starting from the desired pressure-time curve, to identify feasible and possibly optimal combinations of geometry and non-uniform ballistic properties distribution that meets the requirements while complying with the constraints of the design envelope.

4.1 Design approach and modifications

This section describes the solid rocket motor design approach selected to account for additive manufacturing novelties. It stems from traditional approach to solid rocket motor design, presented in [89]. This description covers the main steps required to define grain geometry to meet performance requirements and various constraints imposed by mission, environment, and other design fields. The overall procedure is summarized in Figure 4.1 .

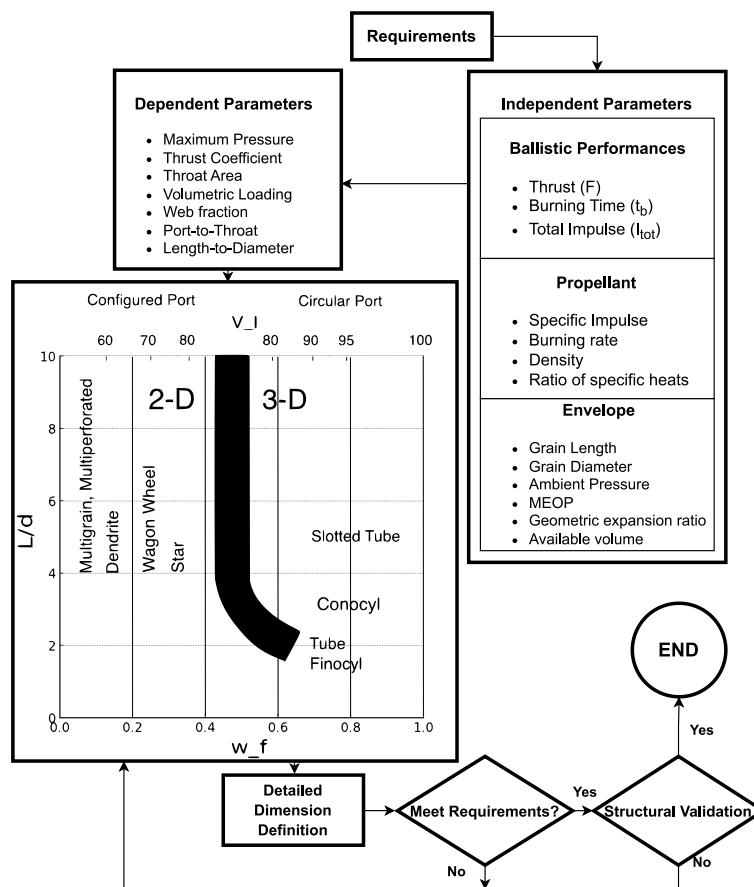


Fig. 4.1 Solid rocket motor design flow

The design flow starts from the definition of motor requirements, which are derived from a set of independent parameters, i.e., parameters imposed by mission, selected propellant, and envelope constraints. Ballistic performance is defined to meet mission requirements for the motor. Thrust, in terms of maximum delivered

propulsive force, burn time, and total impulse delivered by the rocket, are the main performance figures required to achieve mission objectives. The design flow requires selecting the propellant and, consequently, defining its main characteristics: specific impulse, burning rate, density, and specific heat ratio γ . Finally, additional requirements are defined from the operating conditions of the motor. Dimensional constraints including grain length, diameter, and available volume are imposed by the motor's interface with other systems, as well as nozzle expansion ratio. Maximum Expected Operating Pressure (MEOP) is determined by pressure curve requirements and grain casing design and structural behavior, while ambient pressure is determined by motor operating conditions and environmental design point. Once these parameters are extracted from requirements, the designer can compute dependent parameters using relationships among independent parameters, further constraining the grain design. Based on these constraints, an initial selection of the baseline geometry can be performed using geometry-dependent parameters such as web fraction w_f , length-to-diameter ratio L/D , and volumetric loading fraction v_L , using selection maps. This pre-selection relies on previous expertise and designer experience and helps to guide proper selection of a compatible geometry before proceeding with detailed dimensional definition, avoiding the exploration of numerous potentially unfeasible baseline configurations. Once the geometry is pre-selected, the designer must compute grain dimensions to meet all parameters and potentially iterate through the selection-dimensioning phase until compliance is achieved. After achieving compliance, structural assessment is performed to validate the selected geometry, with possible iteration if constraints are violated. These iterative phases also allow searching for optimal solutions among feasible designs to maximize performance metrics, for example by identifying the minimum-stress configuration.

Starting from the traditional approach outlined above, the design methodology has been modified to accommodate non-uniform propellant grains enabled by additive manufacturing. The main modification involves treating propellant characteristics as design parameters rather than fixed constraints. Additionally, the computation of geometric requirements traditionally used to select baseline grain shape from design tables is suppressed. In the proposed approach, the baseline geometry is defined a priori, and performance requirements are met by modifying the preselected configuration and exploiting the non-uniform ballistic characteristics of the propellant, eliminating the need for geometry selection. Target performances are computed starting from the desired pressure-time curve, with envelope constraints

added subsequently. For this analysis, propellant temperature sensitivity is neglected, while propellant density is assumed constant. Density variations due to composition or infill density changes are neglected due to limited available data, though this could be incorporated into the level set solver's fluid dynamics module once accurate correlations from deposition technology studies become available. Given the expected limited variability when ballistic is altered varying propellant composition, this approximation should not significantly affect results. Similar considerations apply to the specific heat ratio of combustion products, which is also assumed constant. Once the desired pressure curve is selected, the MEOP is evaluated as the maximum pressure level, while required burning time $t_{b,req}$ is computed as the time at which pressure reaches ten percent of MEOP. These quantities are directly computed from the pressure curve. Additional parameters must be defined, including maximum required thrust F at MEOP, grain diameter D , grain length L , pressure expansion ratio p_e/p_c , and ambient pressure P_{amb} to define the limits of the design space. To explore different designs thoroughly, several strategies are implemented with different combinations of baseline geometry and burning rate distributions. Moreover, a key distinction among strategies concerns throat area A_t . Some cases treat it as a dependent parameter computed from requirements, following traditional approach under the hypothesis of ideal rocket, while others include it as a design variable to expand the feasible design space. While in the standard design flow other geometrical constraints for grain configuration allows selecting suitable geometries to meet thrust requirement, the definition of an a-priori baseline geometry and a pre-defined parametric space could limit the feasibility of a specific thrust level, since burning area A_b could be insufficient to grant enough mass flow to maintain desired pressure for A_t values computed to obtain such thrust. Designer must be aware of this and select an adequate parameter space to include configurations which are compatible with desired pressure and thrust. Varying A_t as a design parameter allows to overcome this limitation and to increase design flexibility. In the cases where the throat area is kept fixed, A_t is not treated as an optimization variable but as a design parameter defined *a priori*. Since the main objective is related to the pressure–time curve computation, a nominal value of A_t is first estimated from the required thrust level and MEOP as

$$A_t = \frac{F}{p_c C_F} \quad (4.1)$$

where C_F is the thrust coefficient evaluated under the hypothesis of an ideal rocket nozzle with isentropic flow. For the assigned specific heat ratio γ and pressure ratio p_e/p_c , the critical mass flow function Γ and the geometric expansion ratio ε are obtained from

$$\Gamma = \frac{\sqrt{\gamma}}{\left(1 + \frac{\gamma-1}{2}\right)^{\frac{1+\gamma}{2(\gamma-1)}}} \quad (4.2)$$

$$\varepsilon = \frac{\Gamma}{\sqrt{\frac{2\gamma}{\gamma-1} \left[\left(\frac{p_e}{p_c}\right)^{\frac{2}{\gamma}} - \left(\frac{p_e}{p_c}\right)^{\frac{\gamma+1}{\gamma}} \right]}} \quad (4.3)$$

and the thrust coefficient is computed as

$$C_F = \Gamma \sqrt{2 \frac{\gamma}{\gamma-1} \left(1 - \left(\frac{p_e}{p_c}\right)^{\frac{\gamma-1}{\gamma}}\right)} + \varepsilon \left(\frac{p_e}{p_c} - \frac{p_{amb}}{p_c}\right) \quad (4.4)$$

This estimate is used to define a consistent nominal operating point for the subsequent exploration of the design space. A limited number of level-set simulations is then performed to verify the feasibility of the selected configuration, in terms of sustained chamber pressure, complete grain consumption, and compatibility of the resulting pressure level with the prescribed operating conditions. If the preliminary simulations highlight infeasible or physically inconsistent behavior, the value of A_t is adjusted and the screening procedure is repeated before generating the full dataset.

The selection of the configuration in the design space is performed through a stochastic optimization algorithm. The evaluation of individuals can be performed directly with LS burnback computation, however, to speed up the numerous evaluation required to perform stochastic optimization, the former is surrogated through the utilization of meta-models. Different surrogate models have been evaluated, trained on LS burnback data, to guess pressure time response for specific ballistic-geometric grain configurations. The proposed approach is resumed in Figure 4.2. The design starts from the definition of baseline geometry and ballistic distribution, which are described by a set of parameters that defines the design space, whose limits are defined utilizing mission and envelope parameters, as well as material selection for ballistic parameters. The design space is sampled through a sampling plan X that is produced using a Latin hypercube design optimized for maximum space-filling

properties, following [90]. The configurations in the sample plan are generated and their burnback is computed through LS solver and obtained pressure-time curves are used to train surrogate models. Design target can be defined from desired performance curve and employed to define the optimization problem, along with additional constraints if desired. The next chapters describe the surrogate strategies and optimization algorithms employed.

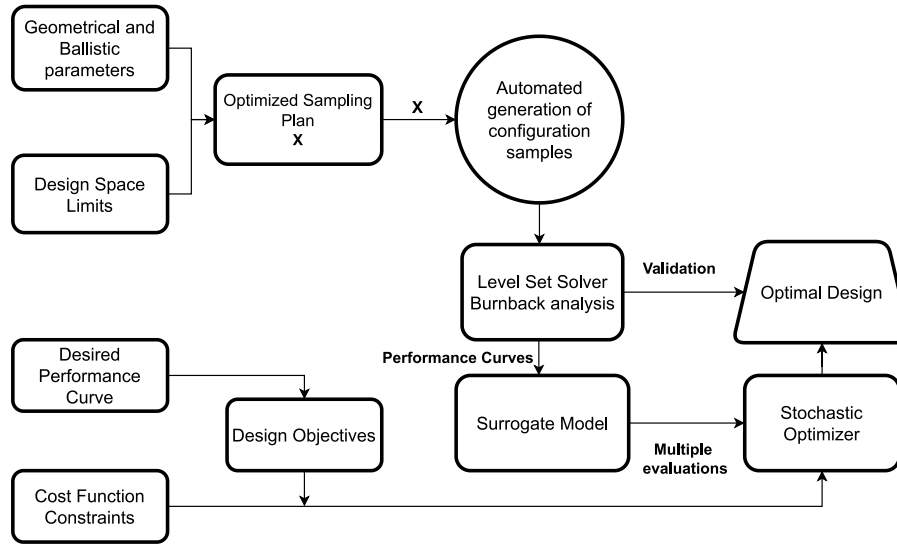


Fig. 4.2 Design flow proposed

4.2 Surrogate modelling

This section contains the methods developed to create, from a certain number of LS evaluations, a model able to predict the pressure-time response of a certain geometric-ballistic configuration, described with a predefined set of parameters, and fast enough to compute a very high number of configuration in stochastic optimizer application. Firstly, the dataset produced from design space sampling is randomly split to save a certain number of sample for test purposes, in order to evaluate the performances of the model against unseen data. Training the model to guess a pressure-time curve requires a tailored approach. In particular, the chosen approach is to train the model to guess a specific pressure value at a specific burning time. The parameter vector is augmented with a normalized time value, and the augmented vector is coupled with a pressure label for the specific configuration from the dataset. A single curve, linked

to a specific configuration, is sampled in 20 time evaluations, and for each time evaluation the correspondent pressure level is used as data label. With this approach, models are trained to evaluate, for a single configuration, the pressure value in every time snap. From every pressure-time guess, the response curve computed by the surrogate is then reconstructed. The performances of the surrogate are computed by evaluating the Root Mean Square Error (RMSE) on validation set and global absolute error on the same set, i.e. the difference between guessed and real value of pressure, for a specific time snapshot, for all configurations in the set. Different methodologies are applied to surrogate the response of the grain burnback, relying on Neural Network, Adaptive Basis Functions and Proper Orthogonal Decomposition. The three methods are compared through the aforementioned RMSE mean value over the time steps, as well as Global Residual values. The latter are the absolute difference, at each time step, between real value and the value guessed by the model. Computing it for every sample in the test set allows to evaluate mean value of the residual at each time step and to construct a mean residual line. The closer it is to zero, the denser the evaluation of the surrogate are around the actual pressure value and the better the performance of the model.

The first used methodology relies on a deep learning network, similar to the approach shown in [84, 91]. The network is built and trained using the TensorFlow library [92]. Among all data, 10 percent is allocated for the creation of a test set to evaluate the network's performances, while another 10 percent is used as a validation set to tune the model's hyperparameters, such as regularization terms in every layer and adaptive parameters of stochastic gradient descent optimizer. The remaining configurations constitute the training set for the network. As previously described, the network is trained to predict the entire pressure-time curve by evaluating specific snapshots of the training curves, sampled from LS Solver pressure integration routine. The neural network adopted in this work is a fully connected feed-forward architecture. The network outputs a single scalar value corresponding to the predicted chamber pressure at the specified time snapshot. The model is composed of eight hidden layers, with a gradually decreasing number of neurons in order to progressively compress the information content and enforce hierarchical feature extraction. The number of neurons per layer is set to (125, 125, 115, 115, 105, 105, 90, 80) respectively. All hidden layers employ the Exponential Linear Unit (ELU) activation function, which was selected to ensure smooth nonlinear behavior and stable gradient propagation in a deep regression setting. Weight initialization is performed using

a Gaussian random normal distribution, while L2 regularization is applied to all hidden layers to mitigate overfitting. A stronger regularization is imposed on the deeper layers, reflecting the increasing abstraction level of the learned features. The output layer uses a linear activation function, consistent with the regression nature of the problem. Training is performed using the AdamW optimizer [93], and the model parameters are tuned based on validation loss monitoring. The depth of the network may appear excessive with respect to the dimensionality of the input space. However, preliminary investigations showed that shallower architectures exhibited limited expressive capability in capturing the highly nonlinear coupling between geometric parameters and the pressure evolution, particularly in correspondence of peak pressure and late burn phases. Increasing depth proved more effective than increasing width in improving approximation accuracy, while maintaining acceptable computational cost.

The second methodology relies on the interpolation via Adaptive Basis Function Construction (ABFC). This approach involves the approximation of a general function, in this case, the pressure value at a defined time snapshot, via a sum of polynomial functions weighted with a scalar value

$$F(X) = \sum_{i=1}^k a_i f_i(X) \quad (4.5)$$

The function depends on the vector of parameters X , hence on geometric-ballistic properties of the examined configuration. The single polynomial basis function $f_i(X)$ is computed as follows

$$f_i(X) = \prod_{j=1}^d x_j^{r_{ij}} \quad \text{with} \quad i = 1, 2, \dots, k \quad (4.6)$$

The exponent r_{ij} and the coefficients a_i are defined through an iterative process aimed at minimizing the mean squared error between the approximation $F(X)$ and the known pressure value at a given time snapshot. Through this adaptive procedure, the structure of the polynomial basis is progressively constructed, allowing the number of terms and their effective polynomial order to be tailored to the specific problem while avoiding unnecessary basis addition and excessive model complexity. The number of basis functions k used to approximate the target function is a user-

defined meta-parameter which can be tuned to alter the approximation, as well as the polynomial order p of the basis function (i.e. the maximum value of r_{ij}). In this application, these values are fixed and not further optimized. Greater details on ABFC implementation can be found in [94]. Once the ABFC approximation is constructed for every time snap, the pressure-time curve is reconstructed, guessing with the surrogate model the pressure value for a certain parameters configuration at the desired time. Also, for this surrogate approach, RMSE is computed on the test set, the latter created by splitting 10 percent of the overall database. Global Residual plots are obtained and shown in Figure 4.11.

The third applied methodology relies on Reduced Order Modeling, exploiting Proper Orthogonal Decomposition (POD) to reduce problem dimensionality [95]. Starting from the original design space POD techniques are applied to reduce the parameter space. Interpolation is applied in the reduced space to create a Reduced Order Model (ROM) and to approximate the pressure-time curve. The ROM is created using the EZyRB library [96], a flexible tool which allows ROMs to be created starting from high-level snapshots of the solution. In this application, the snapshots are the time evaluation of the pressure for each configuration. The initial database from which ROM is computed is formed by a parameters matrix, which contains all the geometric-ballistic parameters of a single configuration, and its correspondent snapshots, i.e. all the pressure evaluations for that single configuration. From this database, POD is computed by the solver using Truncated Singular Value Decomposition, where the reduced basis is constructed by retaining a fixed number of POD modes corresponding to the largest singular values. Successively, the Radial Basis Function is applied as an interpolation method to predict the modal coefficients for the new reduced set of parameters. Again, RMSE is computed on the test set. Global Residual plots are obtained and shown in Figure 4.13. Comparison between sparser and denser datasets shows, as expected, a better behavior of the latter case when all other surrogate models parameters are fixed. The POD-ROM approach treats time differently with respect to previous methodologies, since in this case snapshots are produced with a sampling of the pressure-time response but no additional parameters are added to design space.

The surrogate modeling strategies introduced in this section define the methodological framework adopted for the prediction of the pressure-time response. Their practical implementation, together with the setup of the application tests and the

selection of representative baseline geometries, is addressed in the following section, where the behavior of the different approaches is analyzed and discussed.

4.3 Optimization approach

After the selection of a suitable surrogate model, it is possible to start the design optimization phase. Global design of solid rocket motors requires a careful approach in order to define a geometrical configuration able to achieve desired performances while complying with numerous ballistic, propellant, and envelope constraints, as described in the introduction to this chapter. After defining all requirements from the target curve, the optimization problem can be formulated to identify the optimal geometrical–ballistic configuration satisfying the imposed constraints. The problem is defined as follows:

$$\begin{aligned} \text{find } \min_X \quad & f(X) = \sqrt{\frac{1}{N} \sum_{i=1}^N (p_{g,i} - p_{req,i})^2} \\ \text{subject to} \quad & g_1(X) = |I_{tot,g} - I_{tot,req}| - 0.1 I_{tot,req} \leq 0, \\ & g_2(X) = |t_{b,g} - t_{b,req}| - 0.1 t_{b,req} \leq 0 \end{aligned}$$

The optimization problem aims to determine the parameter vector X that minimizes the Root Mean Square Error (RMSE) between the predicted and target pressure values over the selected time snapshots, while enforcing constraints on total impulse $I_{tot,g}$ and burning time $t_{b,g}$ within a tolerance of 10% of their required values. This choice reflects the fact that the reference performance for the design problem is the temporal evolution of chamber pressure itself, which represents the primary design target, while integral quantities are considered as secondary indicators. Focusing the optimization on the pressure–time history allows the identification of configurations that best reproduce the desired ballistic behavior over the entire burning phase, rather than solutions that satisfy global performance metrics but exhibit significantly different transient responses. The additional constraints on total impulse and burning time are introduced to enforce global ballistic consistency and to avoid convergence toward solutions that may exhibit low RMSE values while being physically or operationally unacceptable. In particular, localized discrepancies between the predicted and target pressure profiles could lead to limited RMSE values while producing configurations with excessive deviations in total impulse or burning duration. The

imposed constraints ensure that candidate solutions remain within acceptable bounds in terms of integral performance parameters, even when an accurate matching of the target pressure curve is not achievable. Within this formulation, the optimization problem may therefore assume the character of a feasibility problem rather than a pure curve-matching task. In cases where the prescribed design space does not allow an exact reproduction of the target pressure–time evolution, the optimization process is still capable of identifying approximate solutions that satisfy the imposed constraints on total impulse and burning time. Such solutions highlight limitations related to the selected parameterization or design space definition, rather than shortcomings of the optimization algorithm itself, and remain of interest from a preliminary design perspective. The optimization is formulated as a mono-objective problem, with the objective function defined as the Root Mean Square Error between the predicted and target pressure–time profiles, computed over uniformly discretized time snapshots. This choice reflects the primary objective of the design process, which is to reproduce as accurately as possible the temporal evolution of the chamber pressure rather than matching isolated performance indicators.

The optimization problem is solved using the Pymoo framework, which provides a unified and flexible environment for defining constrained optimization problems and implementing population-based stochastic algorithms. The design variables are treated as continuous parameters and bounded consistently with the domain used for surrogate model training, ensuring that all candidate solutions remain within the region of validity of the surrogate approximation. A Particle Swarm Optimization (PSO) algorithm is adopted in this work [97]. The choice of PSO is motivated by its suitability for continuous, non-convex optimization problems characterized by a moderate number of design variables. Preliminary tests were conducted comparing PSO with alternative evolutionary strategies available within the same framework, including Genetic Algorithms. These tests indicated comparable solution quality, while PSO exhibited faster and more stable convergence behavior for the present problem setup. For this reason, PSO was selected for the optimization campaigns presented in this work. The swarm is initialized with a fixed population size of 100 individuals and evolved for a prescribed number of 15 generations, which are selected based on preliminary convergence assessments to balance exploration of the design space and computational cost. No additional termination criteria are imposed. Standard PSO parameters provided by the pymoo implementation are employed. Constraint handling is performed directly within the optimization loop

through inequality constraints defined on global ballistic quantities, ensuring physical admissibility of the candidate solutions throughout the search process. The pressure–time response required for the evaluation of each individual is computed using the previously introduced surrogate models, enabling a large number of objective function evaluations and making the use of a population-based stochastic optimizer computationally affordable.

4.4 Test application

After the definition of surrogate models and optimization problem, some applications are here reported to evaluate the overall approach. The basic setting of the evaluations of individual configuration is the same Level Set computation performed in previous chapters. A first study is conducted on finocyl geometries, with different design space in terms of variability of geometrical and ballistic parameters, the latter varied with a polynomial law. This configuration is used to evaluate surrogate performances, to assess the effect of sampling density and design space dimensionality and to explore the flexibility of a specific baseline and the effect of design space limits on the latter. A second test is performed on a different harmonic baseline geometry with a greater variability with respect to finocyl, and a ballistic distribution controlled with a Radial Basis Function (RBF) approximation.

4.4.1 Polynomial distribution - Finocyl geometry

The first step of the design optimization problem concerned the definition of the parameter space. A specific finocyl geometry with non-uniform ballistic properties and a fixed number of fins has been selected as a baseline configuration for optimization. This choice is motivated by the fact that this non-trivial geometry is widely used and well studied, hence, it offers a valid comparison to assess the influence of non-uniform ballistic properties of the propellant. In general, the overall optimization procedure can be carried out on uniform grain also, but in the scope of this work the importance of ballistic variability is prominent and, consequentially, surrogate models and optimizer have been conceived to manage non-uniform distribution. Nonetheless, uniform distribution can be obtained from defined distribution altering parameters accordingly (e.g. in polynomial distribution of Equation 3.2

selecting $a_0 = a_c$). The grain geometry is given by the dimensions of the main sections, e.g., diameter and length, while the fin section is described by the height and the thickness of the fin. The non-uniform ballistic properties are defined by the two coefficients a_c and b of the polynomial function previously used in the initial evaluation of non-uniform properties, reported in Equation 3.2, with constant $a_0 = 7.25e - 5$ [67][80]. The variation of burning rate is obtained varying the a coefficient of Saint-Robert law, following the desired radial function. The parameter range selection ensures that the burning rate values remain within the typical range for solid propellants. Moreover, the total variation is limited to a maximum doubling of the burning rate within the grain, i.e. $a_{max} = 2a_{min}$. This constraint is based on the fact that such a variation can be achieved with reasonable modifications to the propellant composition [98]. While further exploiting additive manufacturing through infill density variation could lead to even greater deviations in burning rate values [22], in the scope of this work, a more limited variation is considered to have a first assessment of the influence of non-uniform ballistic properties distributions on grain design. A first approach describes the individual configuration using a total of eleven parameters, nine geometric parameters, and two ballistic distribution parameters, with no specific limitation on grain overall length, hence with a *free-length* approach. A second *fixed-length* approach has been developed to enforce the maximum length requirement for the grain and to reduce the dimensionality of the problem. Fixing the overall length, as well as the length percentage of each section, reduces the design parameters to six, four for geometry, and two for ballistic distribution. The baseline geometry schematics are presented in Figure 4.3 and in Figure 4.4a, varying within the limits reported in Table 4.1 for the free-length, eleven parameter case. In addition to the design parameters, the parametric tables also contain fixed parameters that are used to run the LS simulation. These parameters are defined in the requirement definition phase (see Section 4.1) and are transferred to the LS solver before starting the evaluation of samples and the creation of data set curves. Among them, A_t is defined following the procedure described in the previous section, using a target value for F of 6000 N at a reference MEOP value of 90 and 58 bar for the two cases, for some sample configurations in the parameter space. The fixed length case is reported in Table 4.2, where the lengths are related to the required length of the grain. The fin section of the grain, which contains a fixed number of eight fins, is the aft portion of the geometry, the one corresponding to L5 section in Figure 4.3. Figure 4.4b shows a 3D visualization of possible configurations. Note that the geometries

presented here represent the inner cavity of the grain, whose surface moves in a direction normal to itself as the burning proceeds.

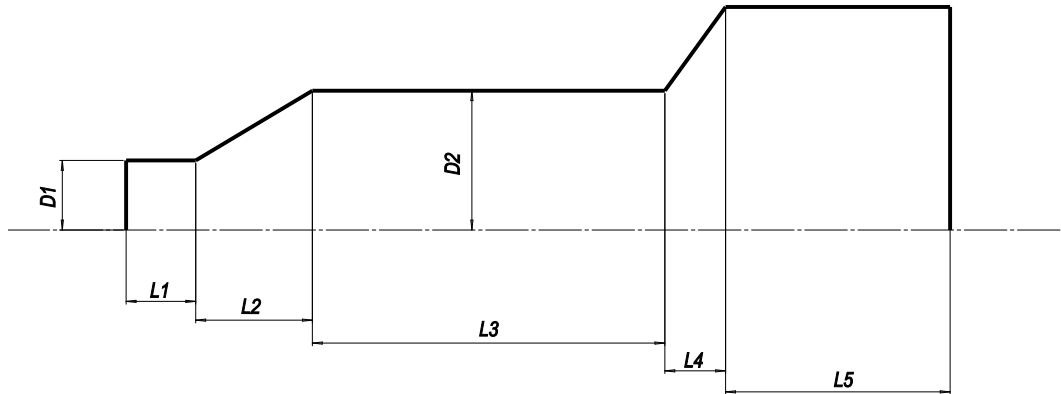
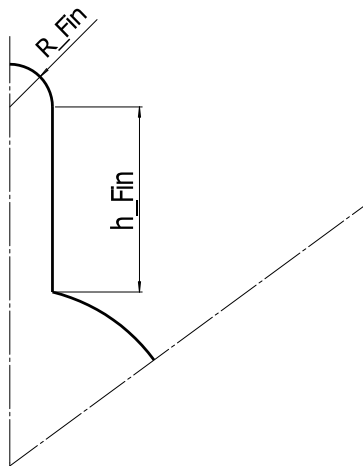
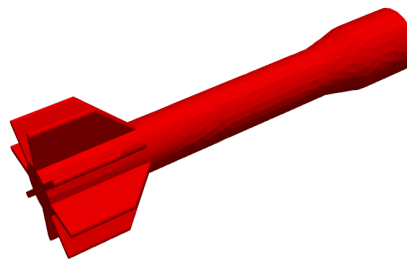


Fig. 4.3 General Geometry and Grain Geometric Parameters



(a) Fin parameters



(b) Burning surface reconstruction example

Fig. 4.4 Example configurations for testing geometry

	Parameters	Values and limits
Geometric	D1	[3,8] <i>cm</i>
	L1	[4,8] <i>cm</i>
	L2	[4,8] <i>cm</i>
	D2	[3,8] <i>cm</i>
	L3	[10,20] <i>cm</i>
	L4	[3,5] <i>cm</i>
	L5	[3,8] <i>cm</i>
	Fin Height	[3,5] <i>cm</i>
	Fin Radius	[0.2,0.4] <i>cm</i>
Fixed/Ballistic	Casing Radius	11 <i>cm</i>
	Throat Area	6.75 <i>cm</i> ²
	Propellant Density	1740 <i>kg/m</i> ³
	Ballistic a_c	[3e-05,5e-05]
	Ballistic b	[0.6,3.1]

Table 4.1 Parameters Table Free Length

	Parameters	Values and limits
Geometric	D1	[3,8] <i>cm</i>
	L1	0.2 L <i>cm</i>
	L2	0.2 L <i>cm</i>
	D2	[3,8] <i>cm</i>
	L3	0.2 L <i>cm</i>
	L4	0.2 L <i>cm</i>
	L5	0.2 L <i>cm</i>
	Fin Height	[3,5] <i>cm</i>
	Fin Radius	[0.2,0.4] <i>cm</i>
Fixed/Ballistic	Casing Radius	11 <i>cm</i>
	Throat Area	6.75 <i>cm</i> ²
	Propellant Density	1740 <i>kg/m</i> ³
	Grain Length, L	30 <i>cm</i>
	Ballistic a_c	[3e-05,5e-05]
	Ballistic b	[0.6,3.1]

Table 4.2 Parameters Table Fixed Length

As previously described, this configuration is the one used to test the selected surrogate models. For both *free* and *fixed* approaches, datasets are created with a certain number of level set evaluations. After the definition of the parameter space and its limits, a Latin Hypercube sampling plan is produced, as described in [90]. The plan is optimized to maximize its parameters space filling properties. Once the sample population is defined, each grain configuration is evaluated using the LS solver. To evaluate each sample, CAD geometry and surface reconstruction via STL files are required, as described in [67].

To produce the corresponding files for every configuration defined by the sampling plan, an automated parametric routine is developed to generate the necessary files from different software tools and run all burnback simulations obtaining, for each configuration, a pressure-time curve. These curves are computed using fixed parameters chosen in design requirements definition phase, such as A_t and ρ_b . Two different sampling plans are used. For the free-length space parameter definition, a pool of 3000 samples is produced, while the fixed length case relies on 4096 evaluation to produce pressure-time curves dataset, in order to obtain a denser sampling of the parameters space. The pressure response is evaluated over a 20 second burn period. The overall simulation time has been chosen to allow most of the configurations to complete the burn, as a priori knowledge of the total burning time is not available. Moreover, the selected simulation time must be greater than the required burning time defined in the design phase. The burning time is then normalized with respect to the maximum burning period to allow proper rescaling for successive surrogate training. Figure 4.5 and Figure 4.6 show the pressure curves obtained for all samples in the free-length parameter case, as well as the fixed length pressure-time set.

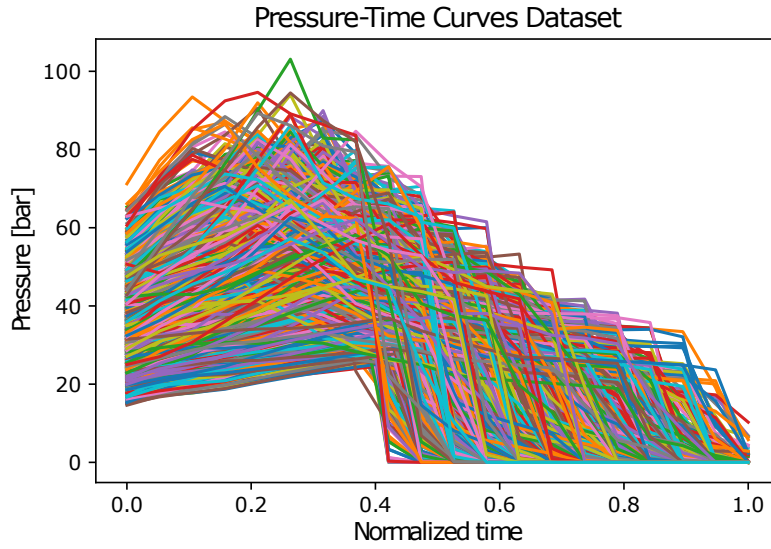


Fig. 4.5 Free Length Dataset

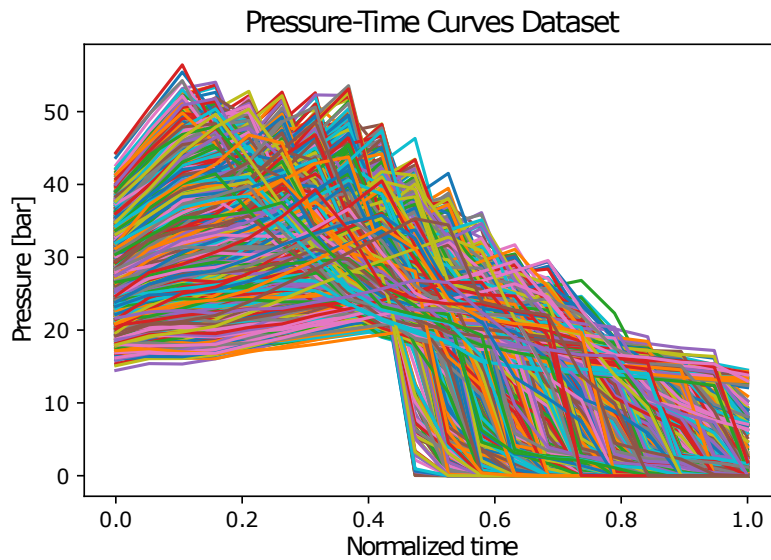


Fig. 4.6 Fixed Length Dataset

By using these pressure-time curves it is possible to train surrogate models as described in the corresponding section. For the first selected approach, the Neural Network approximation, training performance are presented in Figure 4.7, where loss function on training and validation set are presented. Results show that loss decreases for both sets through the epochs, and no overfitting is present. Figure 4.8 shows some

random sample prediction of the surrogate in the validation set, for both free and fixed length cases. The mean RMSE over test set is computed to evaluate the model. Performances on whole test set are reported in Figure 4.9, which shows a scatter plot of residuals computed on every test configuration, i.e. the pressure difference at every time snap, for all the curves in the test set. The mean value of the error at each step allows constructing the mean test error curve, reported over the residual scatter plot. Global residual plots are reported also for ABFC and POD-RBF approaches, in Figure 4.11 and Figure 4.13 respectively, as well as the prediction of the two models on some random test samples in Figure 4.10 and Figure 4.12. Comparison between sparser and denser datasets shows, as expected, a better behavior of the latter case when all other surrogate models parameters are fixed. The obtained results showed better performances for the Deep Neural Network approach, which is the selected method with the lowest RMSE value (see Table 4.3) and a better residual distribution in the time simulation period, with acceptable results even for sparser training datasets. On the contrary, other methods employed didn't show acceptable results and their performance are poor compared to NN approach. For this reason, the latter is selected to support the evaluation phase in the stochastic optimization part of the design procedure. While further evaluation of additional metamodels could enhance approximation accuracy or reveal potentially better methods, such an analysis falls outside the scope of this work. In the context of this study, the selection is thus based on the most suitable tested approach for the current framework.

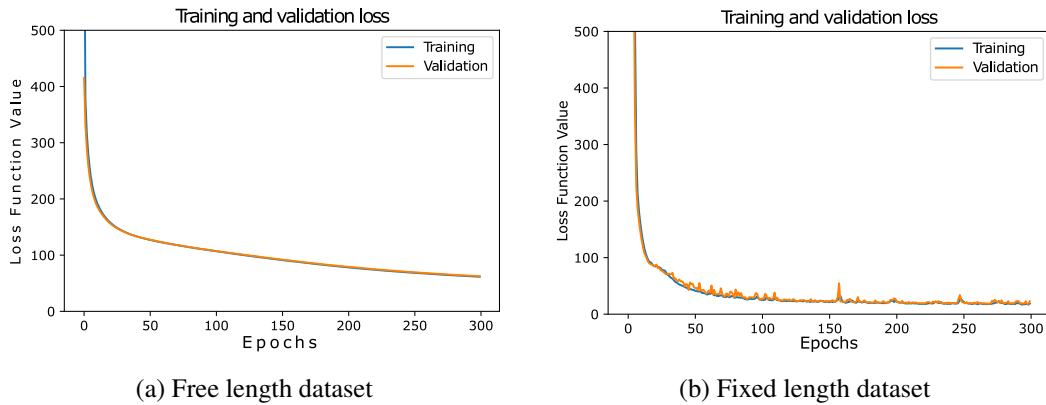


Fig. 4.7 Loss Function over training epochs

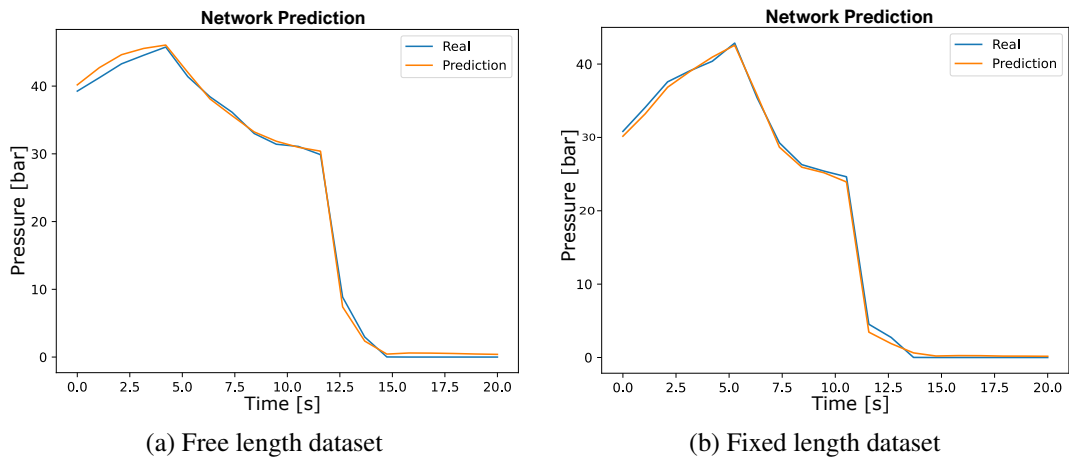


Fig. 4.8 Neural Network Prediction on some test samples

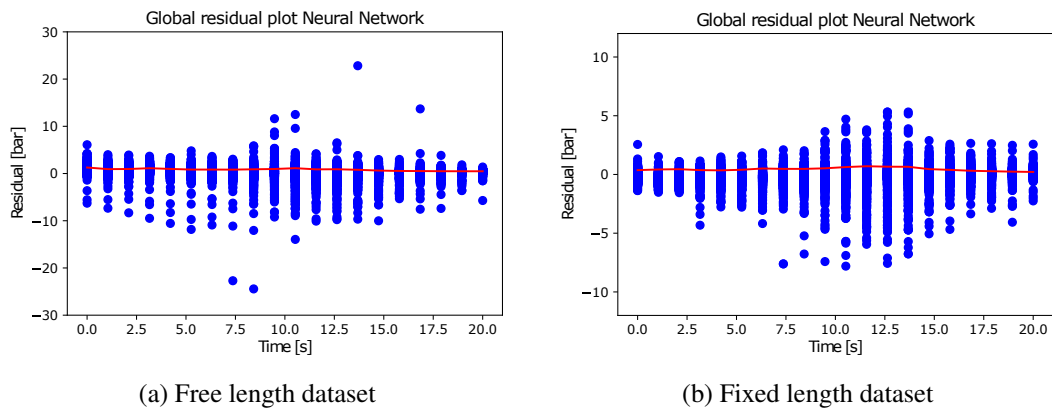


Fig. 4.9 Global Residual Plot for Neural Network

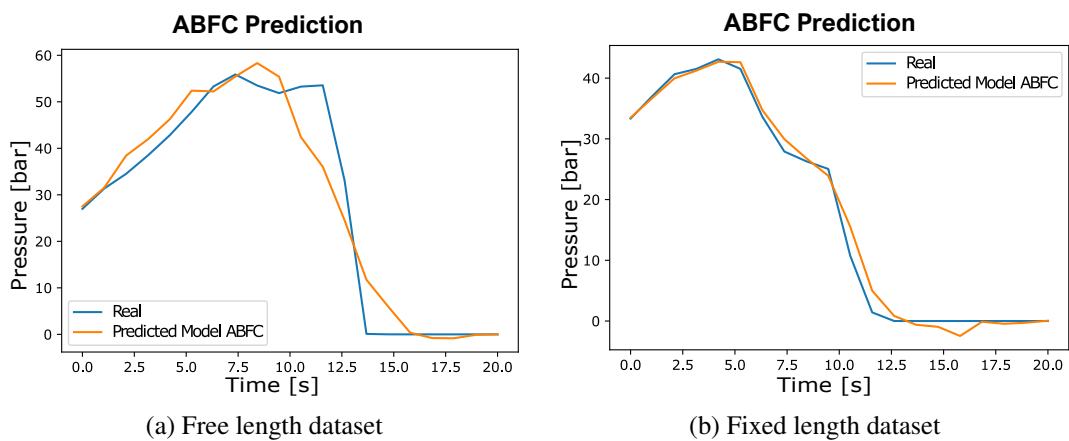


Fig. 4.10 ABFC Prediction on some test samples

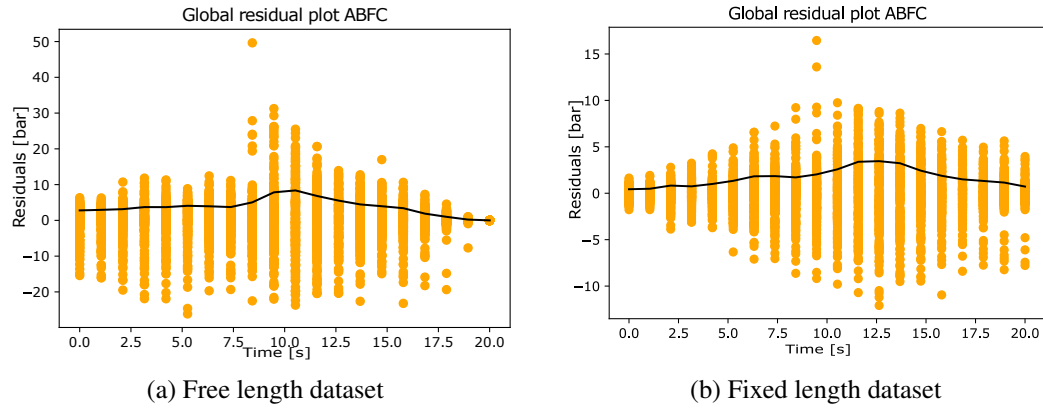


Fig. 4.11 Global Residual Plot for ABFC

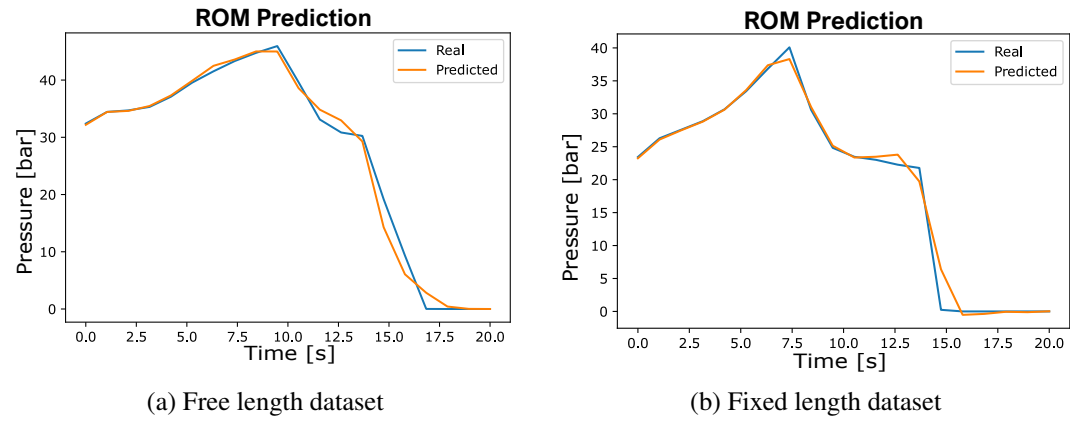


Fig. 4.12 POD-RBF Prediction on some test samples

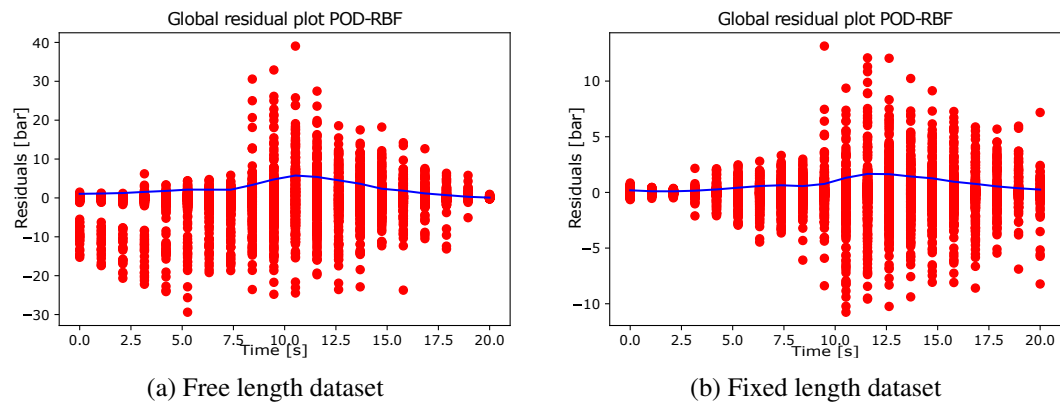


Fig. 4.13 Global Residual Plot for POD-RBF

Model	RMSE-Free Length	RMSE-Fixed Length
Neural Network	0.0636	0.0399
ABFC	5.2181	2.2299
POD-RBF	3.8951	1.3095

Table 4.3 Median RMSE on test dataset

Once the surrogate model is selected and the optimization routine is defined, the latter is tested to assess its ability to identify the optimal configuration with respect to specific pressure–time target curves. The selection of the target curves was carried out with specific criteria: the initial targets were designed to mimic a boost-sustain mission, exhibiting a shape similar to the training curves, with a main peak occurring within the first half of the burning time (before $0.5t_b$) and followed by an approximately constant or slightly decreasing profile after the initial boost phase. Target named Booster 1 to 4 (B1-B4) are created in this manner. In a second step, these curves were slightly modified by altering the peak position and its amplitude, while preserving the general trend of a pronounced maximum followed by a more or less gradual decay until burnout. Booster 5 and 6 (B5-B6) are selected in this fashion. Finally, as a more realistic application case, a pressure–time curve optimized for the first stage of a launcher was employed as the target for the optimization routine, following the work in [99]. Booster 7 and 8 (B7-B8) are obtained from reference, with two different maximum operating pressure levels required. This approach is chosen to verify firstly the optimizer performance on familiar data and then evaluate its performances in evaluating curves outside the training dataset. Test curve set is visualized in Figure 4.14 and Figure 4.15, while computed requirements for target curves are summarized in Table 4.4. Ambient pressure from the working environment considered in this application is $P_{amb} = 1$ atm. A constant specific heat ratio of $\gamma = 1.26$ is adopted as a generic methodological baseline for the numerical framework. While typical values for metalized composite solid propellants fall within the 1.15 – 1.25 range, the chosen value acts primarily as a thermodynamic scaling factor. Because the proposed inverse design methodology aims to match the non-dimensional shape of the pressure-time profile, this assumption affects the absolute magnitude of the evaluated performance parameters but does

not compromise the validity, convergence, or the geometric shaping capabilities of the optimization algorithm. Table 4.5 summarizes the optimum vectors and the performance of the resulting solutions.

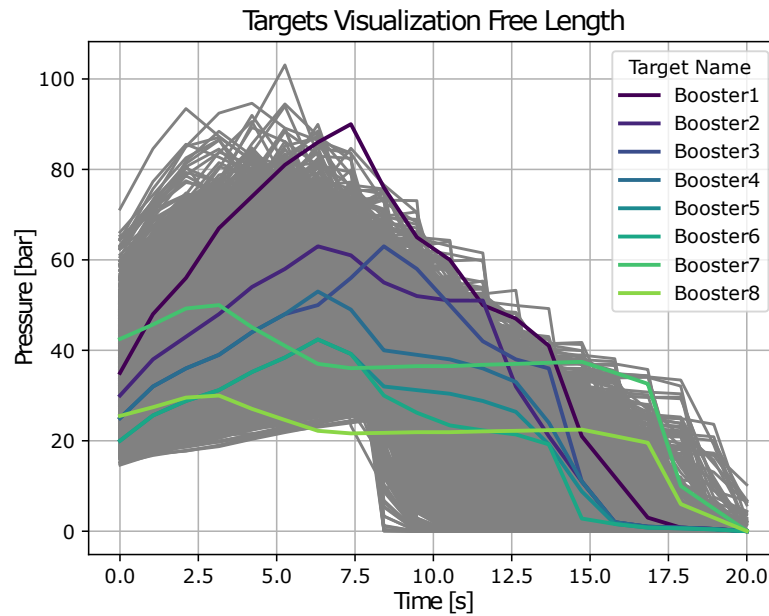


Fig. 4.14 Target curves vs free length dataset

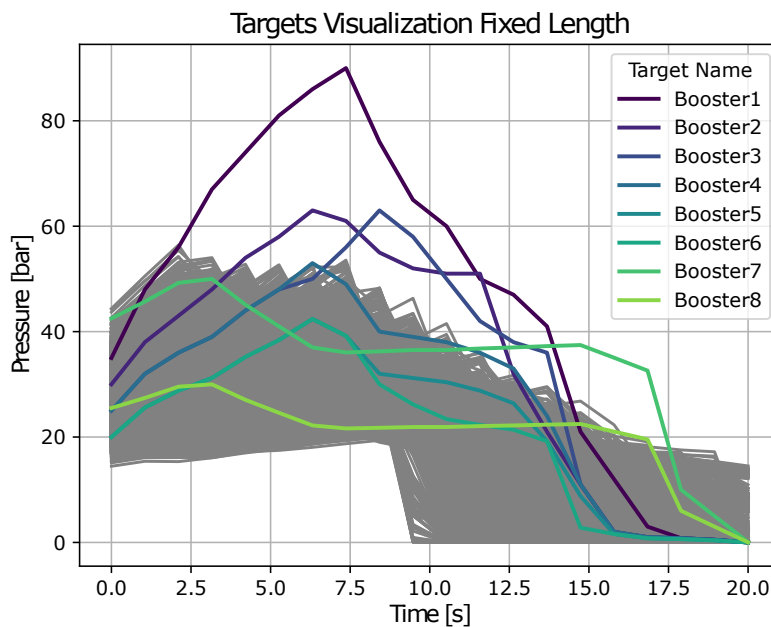


Fig. 4.15 Target curves vs fixed length dataset

Target	I_{tot} , kNs	t_b , s	A_t , cm ²	$\frac{p_e}{p_c}$	D , m	L , m	L_m , m	P_{max} , bar
B1	96.98	16.84	6.75	0.01	0.22	0.3	0.49	90.0
B2	68.66	15.78	6.75	0.01	0.22	0.3	0.49	63.0
B3	64.09	15.79	6.75	0.01	0.22	0.3	0.49	63.0
B4	54.22	15.79	6.75	0.01	0.22	0.3	0.49	53.0
B5	41.19	15.79	6.75	0.01	0.22	0.3	0.49	42.4
B6	37.76	14.73	6.75	0.01	0.22	0.3	0.49	42.4
B7	67.20	17.89	6.75	0.01	0.22	0.3	0.49	50.0
B8	34.74	17.89	6.75	0.01	0.22	0.3	0.49	30.0

Table 4.4 Requirements computed for test curves

The results obtained show how, for the *free-length* dataset, optimizer is well-behaved for the first group of curves, achieving optima close to the target (see Figure 4.16, Target 2-3-5), while the optimum is quite different from the target for curves that deviates from the dataset, possibly resulting in feasible solutions but with a poor match to the target curve (see Figure 4.16, Target 4-6-7-8). On the contrary, results for *fixed-length* dataset are not equally satisfactory. These results show the ability of the optimizer routine to explore the space of the design parameters and to find a feasible configuration to meet pressure requirements, provided the target curve remains sufficiently similar to the training dataset. Conversely, when the target curves deviate substantially from the original dataset, the evaluated optima are less satisfactory, and this can be explained with some considerations. In particular, if the required curve is too dissimilar from the dataset, optimizer cannot find an acceptable solution inside the design envelope. The main reason for this behavior should be attributed to the limitation of the design space. Since the design space is sampled for database creation including its limits, a target far from this dataset could easily be produced by a configuration which is outside the defined limits. This limitation is particularly evident for the fixed-length database, where the limited envelope of the dataset does not allow the optimizer to find an optimal solution. The attempt to search for optimal solution for target that vastly exceed design envelope, leads

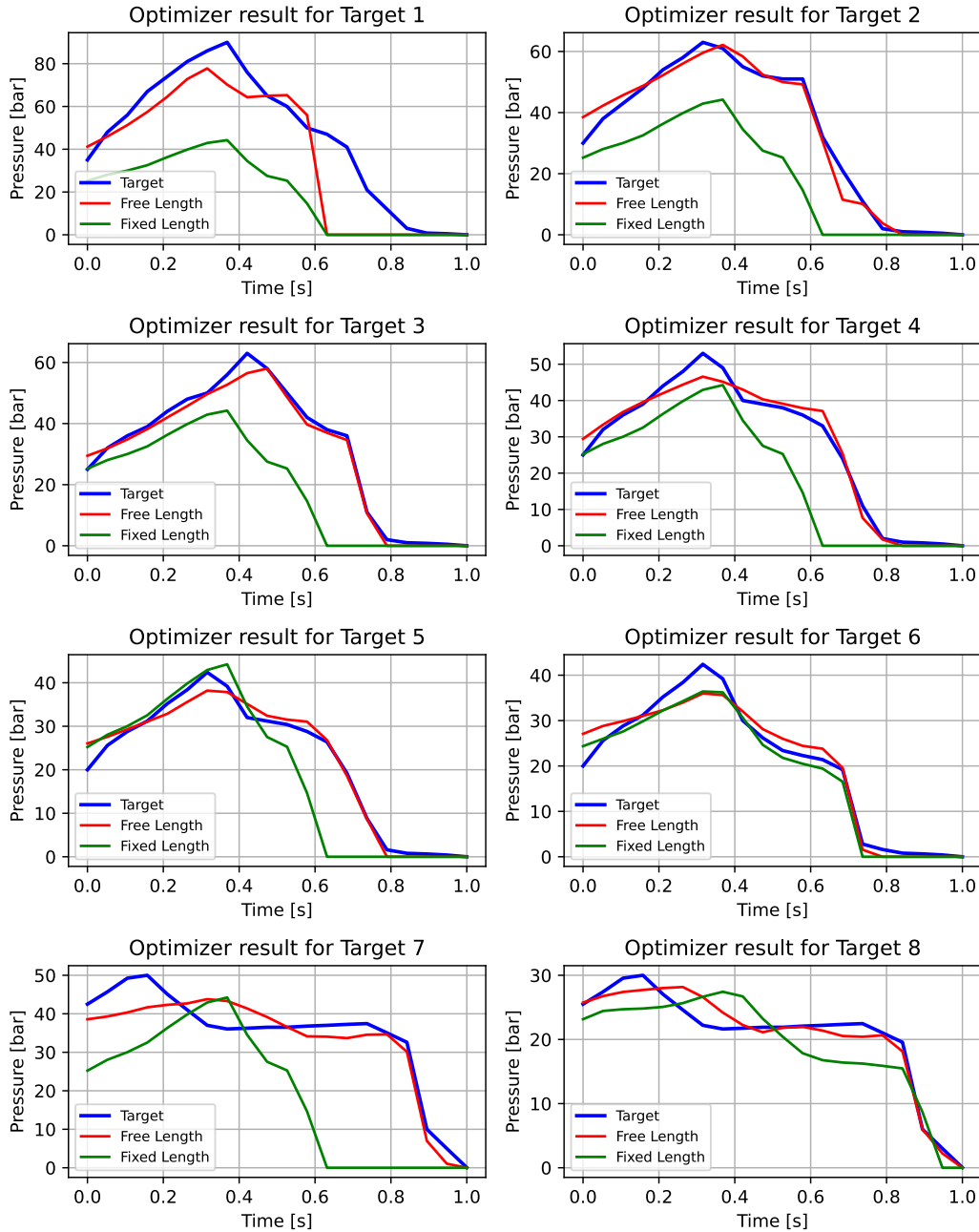


Fig. 4.16 Optimum visualization for target curves

the optimizer to a common solution at the edge of design space. The solution also violates the imposed constraints. This means that the selected target is unfeasible, since it's impossible to find a solution within the design space which satisfies imposed constraints. This happens for all the target curves outside the envelope (see Figure

4.16, Target 1-2-3-4-5-7). As stated above, this parameter space limitation also influences the free-length case. In this case, even if curves are inside the dataset envelope and feasible configurations are found, the overall shape of the curve is not attainable with the defined parameters, and this lack in flexibility produces slightly different pressure-time shapes for computed optima.

Booster	Database 3000 Free Length	Database 4096 Fixed Length
B1	Performances: $I_{sp} = 254.65 \text{ s}$; $t_b = 12.63^* \text{ s}$; $I_{tot} = 78033.68^* \text{ Ns}$	Performances: $I_{sp} = 198.95 \text{ s}$; $t_b = 12.63^* \text{ s}$; $I_{tot} = 36253.28^* \text{ Ns}$
B2	Performances: $I_{sp} = 238.05 \text{ s}$; $t_b = 15.78 \text{ s}$; $I_{tot} = 67779.71 \text{ Ns}$	Performances: $I_{sp} = 198.95 \text{ s}$; $t_b = 12.63^* \text{ s}$; $I_{tot} = 36253.28^* \text{ Ns}$
B3	Performances: $I_{sp} = 237.80 \text{ s}$; $t_b = 15.79 \text{ s}$; $I_{tot} = 61572.43 \text{ Ns}$	Performances: $I_{sp} = 198.95 \text{ s}$; $t_b = 12.63^* \text{ s}$; $I_{tot} = 36253.28^* \text{ Ns}$
B4	Performances: $I_{sp} = 225.06 \text{ s}$; $t_b = 15.79 \text{ s}$; $I_{tot} = 53510.23 \text{ Ns}$	Performances: $I_{sp} = 198.95 \text{ s}$; $t_b = 12.63^* \text{ s}$; $I_{tot} = 36253.28^* \text{ Ns}$
B5	Performances: $I_{sp} = 220.75 \text{ s}$; $t_b = 15.79 \text{ s}$; $I_{tot} = 41453.50 \text{ Ns}$	Performances: $I_{sp} = 198.95 \text{ s}$; $t_b = 12.63^* \text{ s}$; $I_{tot} = 36253.28^* \text{ Ns}$
B6	Performances: $I_{sp} = 220.91 \text{ s}$; $t_b = 14.73 \text{ s}$; $I_{tot} = 37416.69 \text{ Ns}$	Performances: $I_{sp} = 185.94 \text{ s}$; $t_b = 14.73 \text{ s}$; $I_{tot} = 34716.67 \text{ Ns}$
B7	Performances: $I_{sp} = 228.96 \text{ s}$; $t_b = 18.94 \text{ s}$; $I_{tot} = 63964.19 \text{ Ns}$	Performances: $I_{sp} = 198.95 \text{ s}$; $t_b = 12.63^* \text{ s}$; $I_{tot} = 36253.28^* \text{ Ns}$
B8	Performances: $I_{sp} = 202.51 \text{ s}$; $t_b = 18.94 \text{ s}$; $I_{tot} = 34602.36 \text{ Ns}$	Performances: $I_{sp} = 172.69 \text{ s}$; $t_b = 18.94 \text{ s}$; $I_{tot} = 31676.27 \text{ Ns}$

Table 4.5 Comparison of Performance. Application on two configuration types for 3000 samples and 4096 samples databases (* if constraints violation)

However, even with a sparser initial dataset, the increased envelope and the greater variability of configurations make it possible to obtain better results in finding

a wider class of pressure-time curves with respect to a more limited design space. Hence, to find the optimum configuration for meeting that specific target, it may be necessary to extend the parameter space bounds to increase the variability of the pressure response. It could be possible to compensate for limited geometric flexibility by increasing variability of ballistic properties or changing the baseline configuration entirely. Otherwise, we must accept that the evaluated optimum, although potentially far from the desired curve, represents the nearest feasible configuration within the defined design space provided that it complies, at least, with optimization constraints for I_{tot} and t_b .

To assess the validity of the optimum found, the design procedure requires a mechanical stress evaluation of the selected configuration in order to verify whether it can withstand the prescribed pressure level during operation. The structural assessment is performed using the model presented in Section 2.4. As previously described, the analysis adopts a linear viscoelastic model for the propellant grain and computes the stresses that develop in the grain after ignition and pressure build-up, following a temperature cycle imposed on the motor. The overall geometry is simplified in accordance with a pre-design evaluation, and it is formed by the grain cylindrical volume and the metal casing, neglecting thermal insulation layers, nozzle insertions and every other component which is not directly available in an early design phase. The main parameters used to calibrate the model are reported in Tables 4.8 and 4.9, following [75]. Material properties for both the casing metal and the propellant are also taken from the same reference, and reported in Table 4.6. In the context of the work, only one target has been evaluated in this part to assess the structural models. In particular, target B2 optimum configuration found by the optimizer is evaluated, and its geometry is presented in Figure 4.17. Mechanical boundary conditions are introduced to obtain a well-posed structural problem within the simplified cylindrical geometry adopted in this work. The external surface of the casing and the lateral faces of the model are constrained in order to prevent rigid-body motions and out-of-plane deformations that, in the real motor, would be limited by the presence of the nozzle hardware and end closures, which are not included at this preliminary design stage. Radial constraints are also imposed on the cylindrical casing surface. Although the casing is free to deform radially under internal pressurization in an operational motor, this assumption is intentionally introduced here to simplify the structural assessment and to isolate the stress state induced in the propellant by internal pressure loading. The resulting

configuration does not aim at reproducing the exact casing deformation, but provides a stable and conservative framework for the preliminary evaluation of stresses in the grain. As described in the aforementioned section, the analysis requires the evaluation of the stress built up inside the grain after internal pressurization, for the grain that faced a certain thermal cycle and accumulated thermal stress before ignition, and changed its mechanical characteristics in accordance with the linear viscoelastic relaxation curve approximated with the correspondent prony series. The thermal cycle is here simulated with a hot grain, with a starting temperature of 58 °C, cooled through convection in cold environment, for a sufficient time to reach operating temperature inside the grain. The latter, as well as the cooling time and starting temperature, are specific for the manufacturing process, storage condition, selected mission and several other factors which in the context of this work are neglected. Since no a-priori knowledge of cooling time is provided, a fixed time of 6000 s is chosen to grant, provided the dimensions of the grains evaluated in this phase, a sufficient cooling with a reasonable temperature and temperature distribution in the grain, the latter presented in Figure 4.18. Even without achieving uniform temperature distribution, the solver can manage this type of distribution and designer can tune the cooling profile to best reflect the real motor operating condition. Internal pressurization is applied through a ramp of 20 s up to the equilibrium chamber pressure at ignition. This loading strategy is adopted to ensure numerical stability and convergence of the nonlinear transient solver. Transient mechanical analysis of the whole time-frame is performed to evaluate tensions induced by differential thermal shrinkage during cooling with additional stresses which arises from the ignition and internal pressurization. The stress field is evaluated at the end of the pressurization ramp, which is taken as representative of the most critical condition associated with grain ignition. This choice allows assessment of combined stress concentrations arising from the combined effects of residual thermal stresses and the initial pressure loading. Although compressive loading is dominant, the complex grain geometry and the applied constraints generate a non-negligible deviatoric stress component, resulting in a fully triaxial stress state for which the use of an equivalent stress measure such as von Mises is appropriate. A safety margin is evaluated by comparing the peak equivalent stress with the allowable compressive strength, adjusted through correction factors to account for multi-axiality and temperature- or aging-related effects, following the criteria reported in [74]. Within these assumptions, the structural analysis provides a preliminary but

physically consistent validation of the optimized configuration, sufficient to support the design choices discussed in this work. Figure 4.19 shows stress distribution inside the grain. As expected, maximum stress concentration is localized in sharp features like fin grains and section diameter changes. Maximum equivalent stress σ_c is compared to allowable stress through the following margin of safety criterion on stress $MS = \frac{K_f \sigma_a}{DF \sigma_c} - 1$ [74, 75], where global knockout factor K_f is taken from [75], while allowable stress σ_a is taken from experimental data on HTPB-based composite propellant, from [100]. A safety factor $DF = 1.5$ is also applied to compute a margin of safety of $MS = 1.19$, which for the case in exam is greater than one, resulting in an acceptable optimum for the pre-design structural assessment.

Properties	Maraging steel	HTPB-based propellant
E , MPa	190,000	1.09 (E_∞)
ν	0.3	0.499
ρ , kg/m ³	8000	1700
α , mm/mm·°C	10.1×10^{-6}	9×10^{-5}
k , W/m·K	25.3	0.60
c_p , J/kg·K	813	see Table 4.7

Table 4.6 Mechanical and thermal properties of materials

c_p , J/kg·K	Temperature, K
895	233
525	273
380	313
270	343

Table 4.7 Variation of solid propellant's specific heat with temperature

i	E_i , MPa	λ_i
1	0.5	0.0052
2	0.21	0.0578
3	0.105	0.4059
4	0.075	3.5383
5	0.054	37.6457
6	0.051	240.954

Table 4.8 Prony series coefficients of HTPB-based propellant

This analysis is reported to verify the consistency and applicability of the proposed mechanical model within a preliminary design framework. The designer can

C_1	C_2	$T, ^\circ\text{C}$
210.2	4542.4	35

Table 4.9 WLF function coefficients of HTPB-based propellant

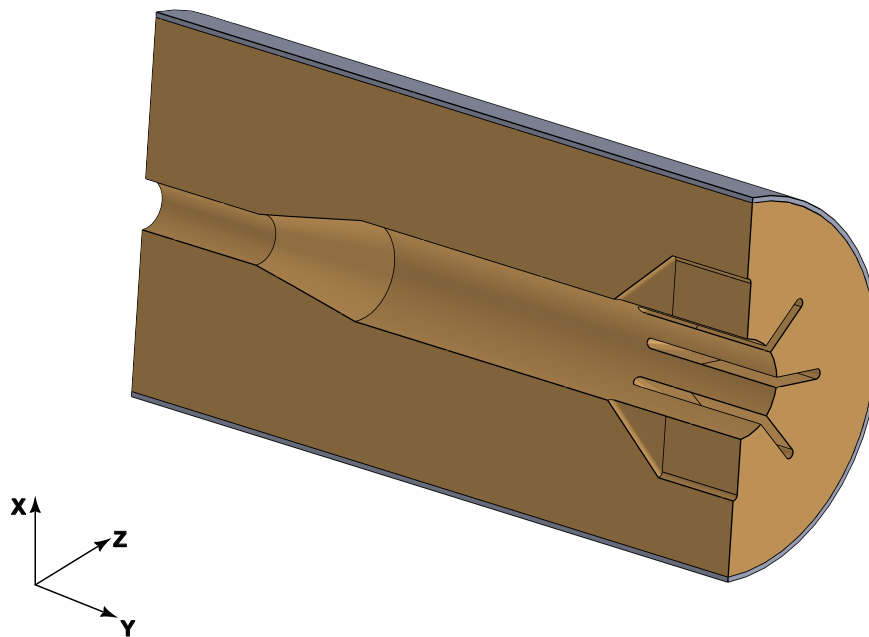


Fig. 4.17 Target B2 optimal configuration, grain volume and casing

M: FIXED ENDS Optimum #2 FIN-POLY

Temperature

Type: Temperature

Unit: $^\circ\text{C}$

Time: 6000 s

Max: 4,7163

Min: -25,536

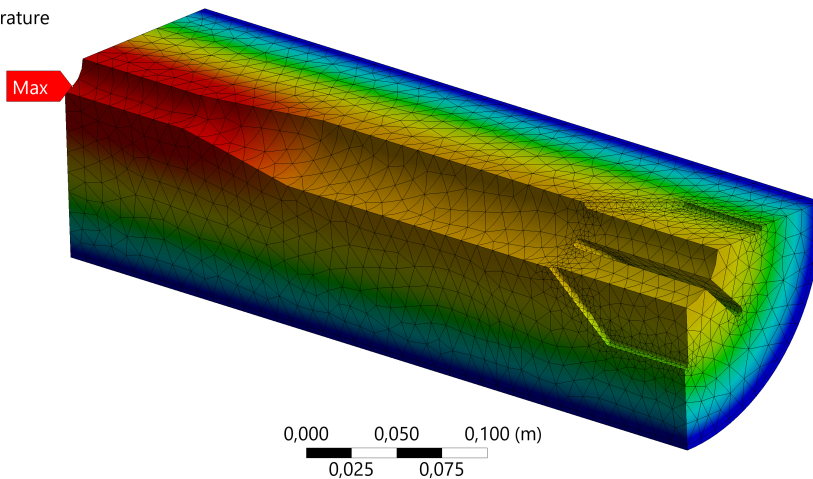
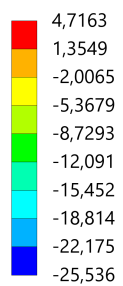


Fig. 4.18 Temperature distribution inside the grain and casing at the end of temperature cycle

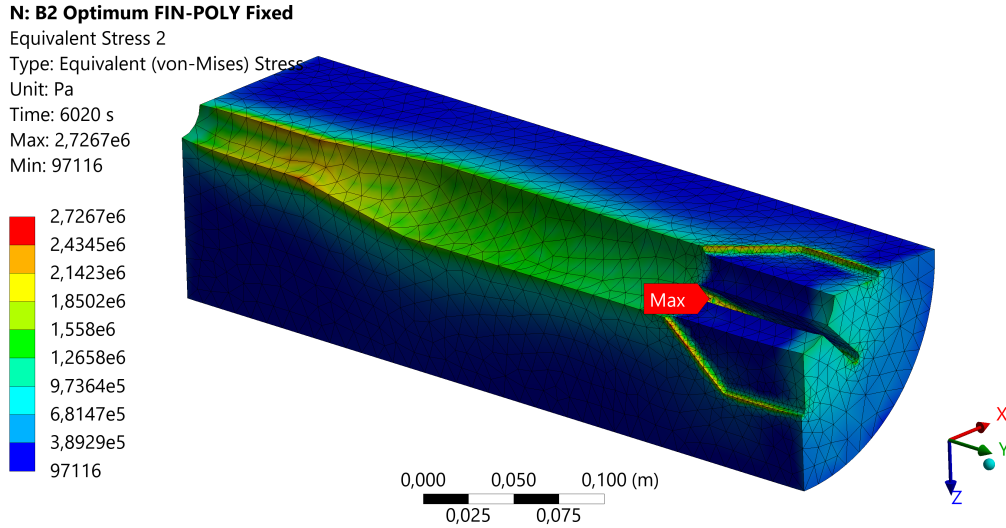


Fig. 4.19 Von Mises stress inside the grain at the end of pressurization ramp

tune all the parameters to best reflect the studied materials and working conditions the motor will face during operation. The same analysis can also be used to compare optimum configurations obtained with different geometries and can be potentially integrated into the design flow as an optimization criterion.

4.4.2 RBF distribution - Harmonic and Finocyl geometry

With the aim to improve the flexibility of the design tool, different strategies have been implemented. First, the throat area A_t was included in the design parameters. Additionally, a new ballistic distribution was selected to enhance the variability of the burning rate inside the grain. Finally, a new geometry definition has been implemented to obtain different geometries with respect to baseline finocyl geometries. The introduction of A_t as a parameter allows for important modification in the pressure-time response, since this value extensively affects chamber discharge and nozzle expansion, and also guarantees a greater envelope of attainable thrust F , once MEOP is defined. The new distribution of the coefficient a is based on a Gaussian Radial Basis Function (RBF) [101]. This reconstruction allows to easily create ballistic distributions, fixing its centers and weights, matching desired variations

inside the grain. The implemented RBF is a function of the radial coordinate r and is defined as follows

$$\begin{aligned} f_r(r) &= \sum_{i=1}^N w_i \varphi_i(r) \\ \varphi_i(r) &= e^{-(\sigma|r-r_c|)^2} \end{aligned} \quad (4.7)$$

Where N is the number of basis functions used, $r_{c,i}$ is the radial coordinate of i -th center, w_i is the i -th weight of the distribution and σ is the shape parameter of the function. To implement variation along motor axis, this distribution is multiplied by a piecewise polynomial of the axial coordinate y , defined as follows

$$f_y(y) = \begin{cases} f_y^- = \sum_{j=0}^P c_j y^j & y \leq y_{split} \\ f_y^+ = f_y^-(y_{split}) - m(y - y_{split}) & y > y_{split} \end{cases} \quad (4.8)$$

Where P is the polynomial order of the axis distribution before split, y_{split} is the axial coordinate where the function shifts to a linear behavior, m is the slope of the linear function. The final distribution $f(r,y) = f_r f_y$ allows for a greater variation and greater control of the distribution by designer, since a greater number of parameters are involved in the definition of ballistic properties, with a maximum of $2N + P + 3$ parameters available for design and optimization.

The other modification to test application is a new geometry definition, based on the creation of the geometry of the inner cavity of the grain through the definition of port profile, for a certain number of grain sections. The profile of the port is described by an harmonic function of angular radial coordinate θ and is defined in the first quadrant, given the number of harmonics M , as follows

$$p(\theta) = \sum_{n=0}^M A_n \cos(4n\theta) + B_n \sin(n\theta) \quad 0 \leq \theta \leq \frac{\pi}{2} \quad (4.9)$$

The values are then mirrored in the other quadrants to enforce the axial symmetry of the grain. The profile of the port is computed for every desired section, and axial position is assigned to every port dividing the total length L of the grain by the desired number of ports. The 3D envelope of the grain is reconstructed by interpolating the

profiles, exploiting loft function in CAD environment. This choice allows to obtain a wide range of geometries, varying from a simple cylindrical bore to complex star-type shapes (see Figure 4.20 and Figure 4.21). However, it is more difficult to interpret how the choice of parameters affects the overall shape. If k is the number of ports selected to describe the geometry, this approach adds a maximum of kM parameters to the design space, plus the fixed length of the grain. Therefore, the total number of parameters that the designer could use in theory is $2N + 2kM + P + 4$, accounting for geometry and ballistic variability. The designer must define the number of parameters used by defining the values of N, k, M, P and eventually fixing the values of additional parameters, such as y_{split}, m, σ, L . Moreover, it's also possible to fix some of the terms used in RBF definition or port definition, not accounting them as design parameters. In this work, a limited number of parameters has been taken into account.

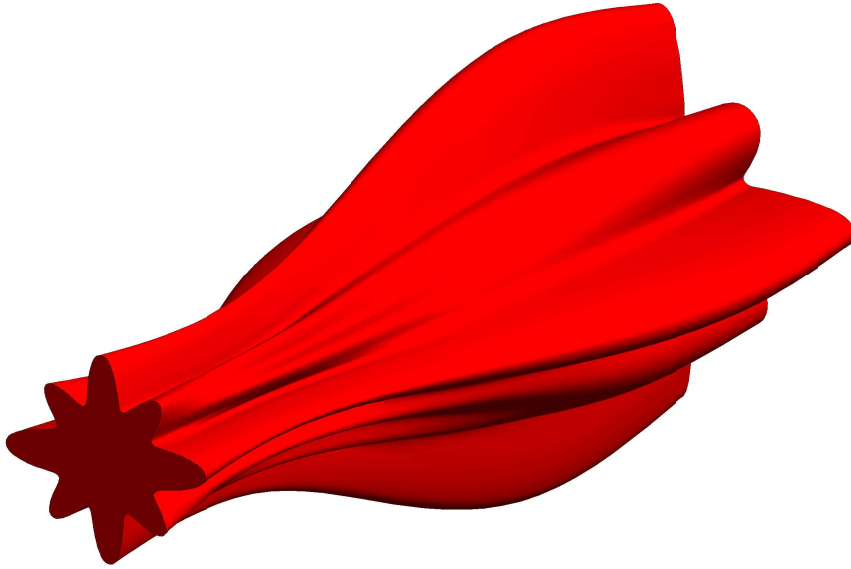


Fig. 4.20 CAD visualization of 3D grain example, $M = 5$

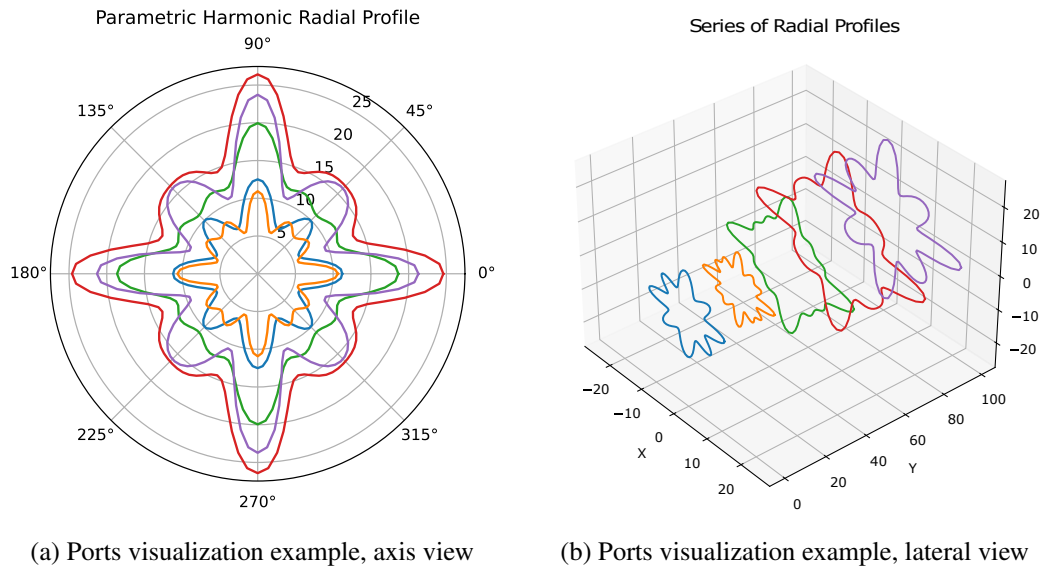


Fig. 4.21 Example of port configuration

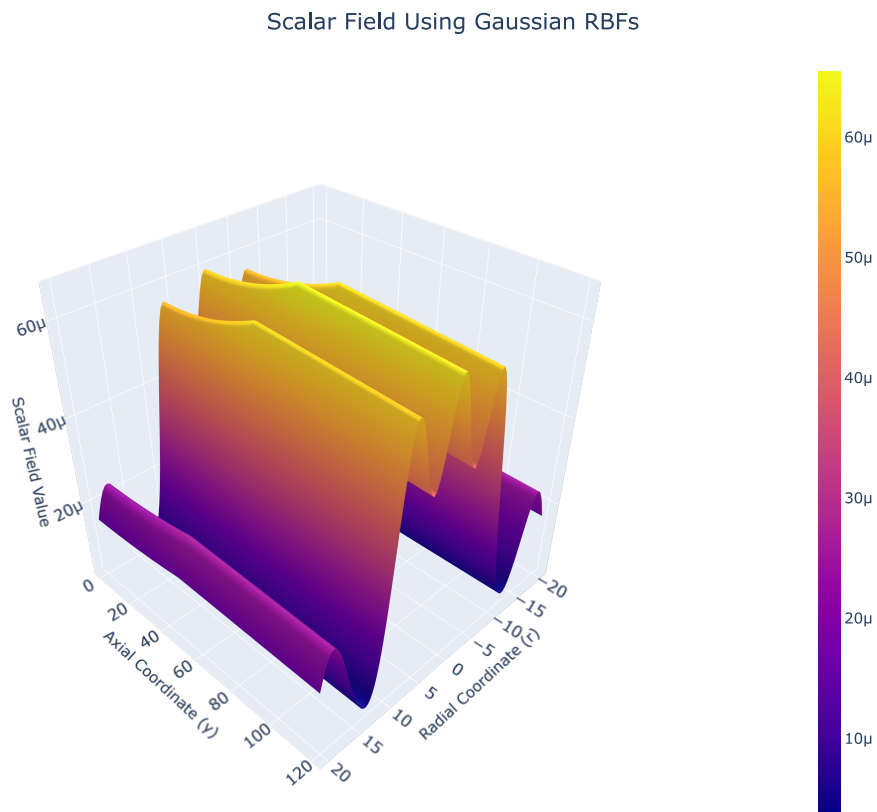


Fig. 4.22 Ballistic distribution example. Values obtained using parameters inside the RBF-Port design envelope. Scalar field refers to a parameter field

Parameters		Values and limits
Geometric ($k = 5$)	A_1 (One for each k -th port)	[3,10]
	A_2	0.5
	A_3	0.5
	B_1 (One for all k ports)	[1,3]
	B_2	0.5
	B_3	0.2
	y_{split}	50 <i>cm</i>
	m	6e-04
	Casing Radius	15 <i>cm</i>
Fixed/Ballistic	Throat Area	[12,28] <i>cm</i> ²
	Grain Length, L	40 <i>cm</i>
	w_1	[4e-05,12.5e-05]
	w_2	[3e-05,8.5e-05]
	w_3	[2e-05,4.5e-05]
	$r_{c,1}$	[1,2.5] <i>cm</i>
	$r_{c,2}$	[3,8] <i>cm</i>
	$r_{c,3}$	[8.5,10] <i>cm</i>
	σ	[2,7.5]
	c_1	[0.2,0.58]
	c_2	-9.25e-05
	c_3	1.25e-05

Table 4.10 Parameters Table RBF-Port

Once the new design has been selected, a new application on previous test curves is run to evaluate to which extent a broader variation of ballistic properties can be exploited. In particular, the RBF configuration implemented to evaluate the design process has $N = 3, P = 3$ but, as stated before, only the first P-term is considered as a design parameter, while the other two are fixed. The same approach is used for the $k = 5$ ports definition, where of the $M = 3$ terms of the sum, only the first is used as design parameter. Moreover, only the A term of the expansion varies between the ports, for a total of five design parameters, while B term remains the same for all ports. With these assumptions, this type of configuration is described by 15 parameters. Table 4.10 resumes all the design parameters and the values which define the configuration, where grain length L and casing radius r_{casing} are increased from the previous application to ensure sufficient burning time. Figure 4.22 shows

an example of Gaussian RBF obtained with described setup. RBF is applied to both finocyl geometry and harmonic-port geometry. For the finocyl-RBF application the geometric parameters are the same used for Fixed Length case (see Table 4.2). The finocyl geometry is described by four geometric design parameters, hence in this case we have a total of 13 parameters to define the configuration. The same approach used previously is now applied to these new configurations. New datasets, formed by 3000 individuals, are computed through LS evaluation of grain burnback and pressure computation. These datasets are used to train surrogate models to guess the pressure response of a specific configuration. The latter are applied to evaluate individuals in PSO algorithm. Figure 4.23 and Figure 4.24 show datasets envelope and targets for the new configurations, while Figure 4.25 and Table 4.11 resume the optimization results on the target set. Requirements on A_t are now not enforced, however now desired thrust F is imposed to have an initial guess of throat area to obtain such level of thrust and which varies for each configuration. For these reason, also required I_{tot} varies in this new application.

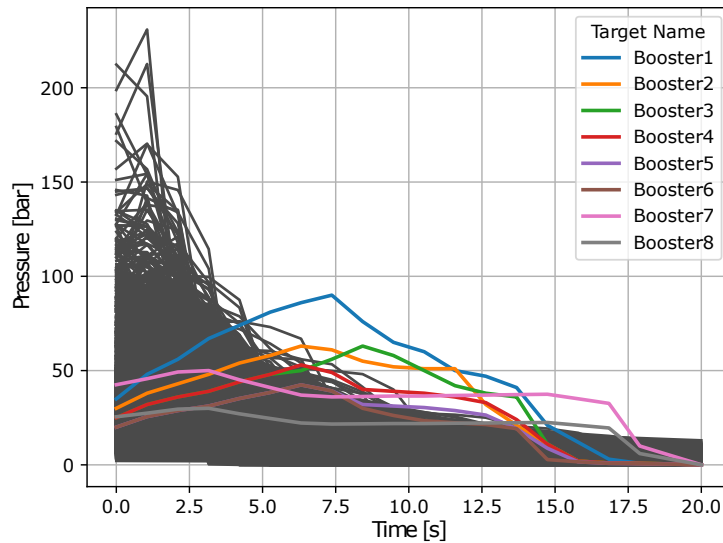


Fig. 4.23 Dataset for RBF-harmonic

The first key consideration is the effect of ballistic property distribution on the dataset envelope. Comparing Figure 4.23, 4.24 and Figure 4.14, Figure 4.15, it is evident that the same finocyl baseline geometry produces significantly different pressure-time envelopes due to greater variation in the burning rate. Obtained results deviate considerably from the desired pressure curve, particularly for the initial targets, which were designed to replicate the behavior of previous datasets. For

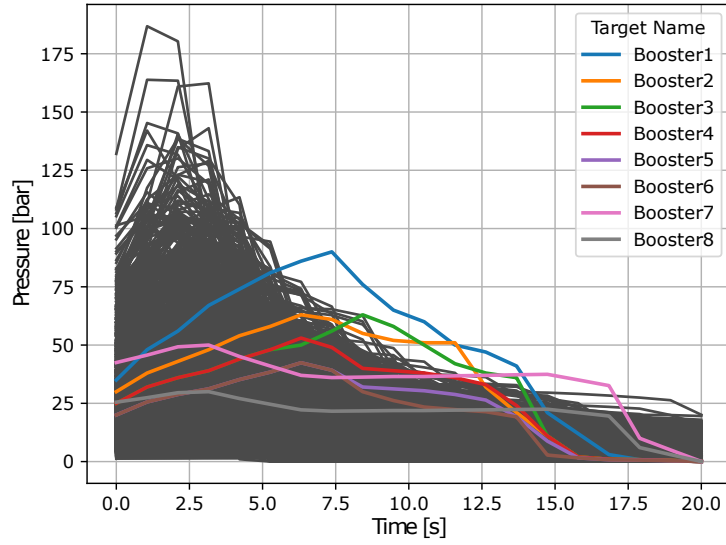


Fig. 4.24 Dataset for RBF-Fin

the last two targets, although the desired pressure curve is not fully achieved, the results show an improvement compared to the previous approach, especially in the fin configuration, due to added variability introduced by the Radial Basis Function distribution. However, this increased variability appears to reduce the optimal values for I_{sp} . For these targets, the two different geometries yield similar optimal solutions, highlighting the significant influence of ballistic distribution. Further considerations must be made regarding constraint violations. Many of the optimal solutions, despite deviating from the desired pressure curve, do not violate constraints on I_{tot} and t_b . This can be attributed to the use of throat area A_t as a design parameter, which directly affects thrust computation. Since this parameter is now variable, the resulting thrust magnitude can change, allowing the delivered I_{tot} to meet the required value even if the pressure curve does not align with the target. In this new configuration, imposing I_{tot} as a constraint alone is insufficient to ensure similarity in pressure performance. Therefore, additional constraints must be introduced, such as enforcing a required thrust curve $F(t)$ instead of pressure curve or imposing a maximum tolerance on the root mean square error (RMSE) of the pressure curve. Although the results are not entirely satisfactory, it is crucial to emphasize the primary advantage of this new approach: the flexibility provided by the novel configuration framework. This design environment allows designers to introduce complexity simply by adjusting the number of parameters governing the geometry and RBF distribution, thereby expanding the design space. However, this increased flexibility comes at the cost

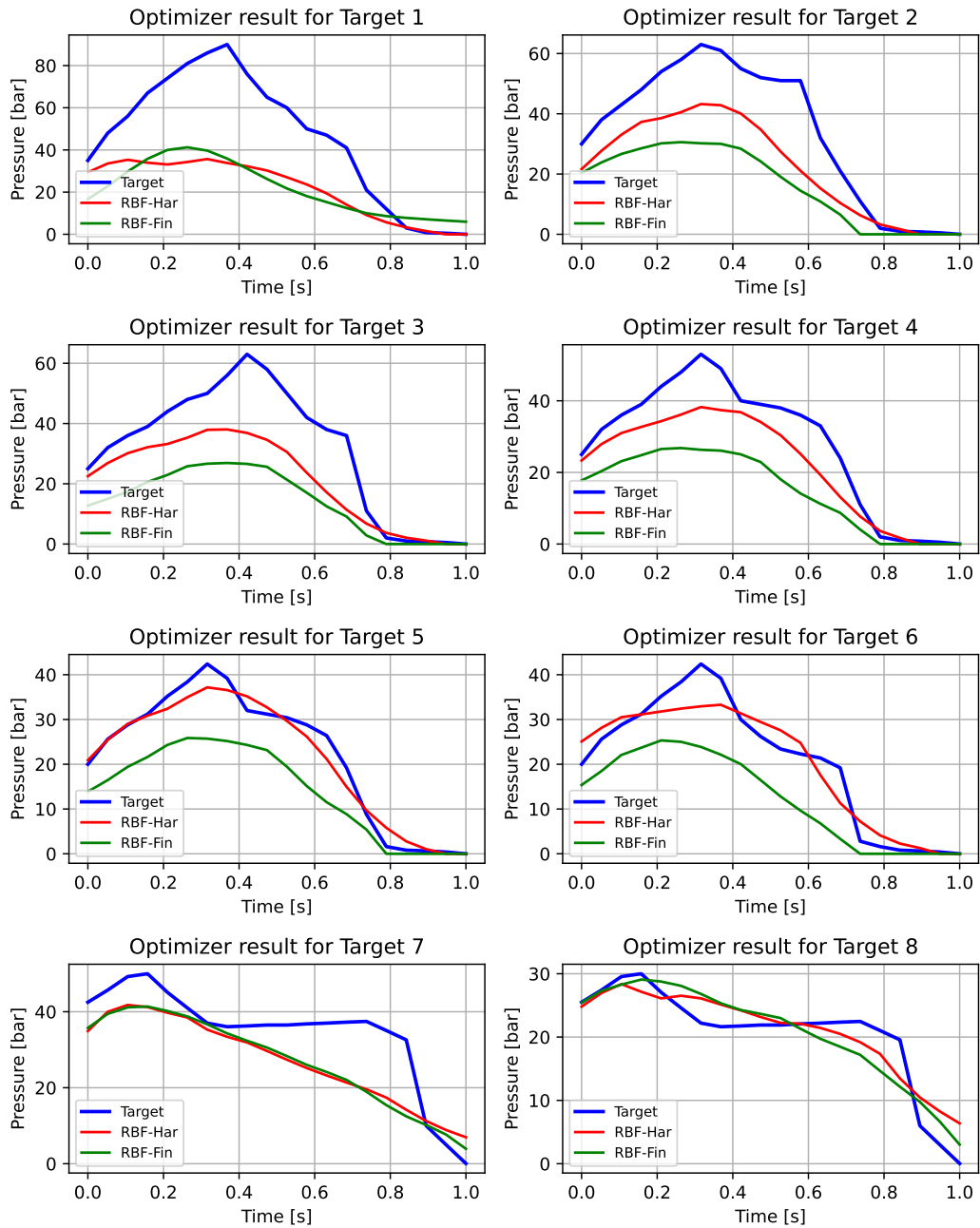


Fig. 4.25 Optimum visualization for target curves RBF

Booster	RBF-Harmonic	RBF-Fin
B1	Performances: $I_{sp} = 171.01 \text{ s}; \quad t_b = 17.89 \text{ s};$ $I_{tot} = 67658.30 \text{ Ns}$	Performances: $I_{sp} = 141.06 \text{ s}; \quad t_b = 20.00 \text{ s};$ $I_{tot} = 65937.25 \text{ Ns}$
B2	Performances: $I_{sp} = 174.66 \text{ s}; \quad t_b = 16.84 \text{ s};$ $I_{tot} = 71965.33^* \text{ Ns}$	Performances: $I_{sp} = 176.35 \text{ s}; \quad t_b = 14.74 \text{ s};$ $I_{tot} = 79868.90^* \text{ Ns}$
B3	Performances: $I_{sp} = 166.80 \text{ s}; \quad t_b = 16.84 \text{ s};$ $I_{tot} = 65965.10 \text{ Ns}$	Performances: $I_{sp} = 161.47 \text{ s}; \quad t_b = 14.74 \text{ s};$ $I_{tot} = 74942.46 \text{ Ns}$
B4	Performances: $I_{sp} = 171.55 \text{ s}; \quad t_b = 16.84 \text{ s};$ $I_{tot} = 69125.67^* \text{ Ns}$	Performances: $I_{sp} = 167.13 \text{ s}; \quad t_b = 15.79 \text{ s};$ $I_{tot} = 75741.14^* \text{ Ns}$
B5	Performances: $I_{sp} = 169.91 \text{ s}; \quad t_b = 16.84 \text{ s};$ $I_{tot} = 66680.28^* \text{ Ns}$	Performances: $I_{sp} = 163.07 \text{ s}; \quad t_b = 15.79 \text{ s};$ $I_{tot} = 75514.79^* \text{ Ns}$
B6	Performances: $I_{sp} = 168.93 \text{ s}; \quad t_b = 16.84 \text{ s};$ $I_{tot} = 61158.74^* \text{ Ns}$	Performances: $I_{sp} = 161.79 \text{ s}; \quad t_b = 14.74 \text{ s};$ $I_{tot} = 75572.99^* \text{ Ns}$
B7	Performances: $I_{sp} = 199.55 \text{ s}; \quad t_b = 20.00 \text{ s};$ $I_{tot} = 87374.70 \text{ Ns}$	Performances: $I_{sp} = 188.23 \text{ s}; \quad t_b = 20.00 \text{ s};$ $I_{tot} = 87162.99 \text{ Ns}$
B8	Performances: $I_{sp} = 160.85 \text{ s}; \quad t_b = 20.00 \text{ s};$ $I_{tot} = 68605.84 \text{ Ns}$	Performances: $I_{sp} = 172.93 \text{ s}; \quad t_b = 20.00 \text{ s};$ $I_{tot} = 76901.13 \text{ Ns}$

Table 4.11 Comparison of Performance for RBF Harmonic and Fin. (* indicates constraint violation)

of higher computational requirements and necessitates careful consideration of the applicability of surrogate modeling (see Section 4.5).

Also in this second case a structural assessment is performed to evaluate stress distribution in the geometry, employing the same model, materials, thermal cycle, loading and constraints of the previous finocyl test case. The selected target for the analysis is target B5, whose geometry is presented in Figure 4.26.

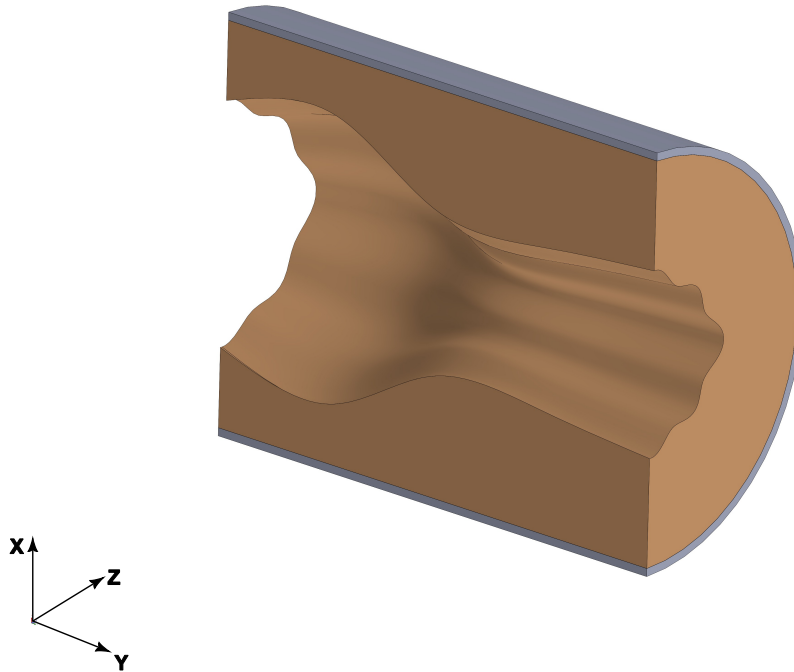


Fig. 4.26 RBF-HAR optimum B5 geometry of grain and casing

Figure 4.27 shows temperature distribution in the grain and casing. While the imposed thermal cycle is identical to the previous case, the resulting temperature distribution differs slightly due to the different geometry. This choice is motivated by the necessity to define the same operating condition to effectively compare different solutions. For the same reason, the pressure level is intentionally set to the ignition pressure of target B2, in order to compare the two geometries under identical and conservative loading conditions. Under these conditions, thermal induced stress combines with internal pressurization to generate the stress field presented in Figure 4.28. The same operating condition and cycle loading on this smoother geometry results in a lower peak stress concentration with respect to the previous finocyl geometry. Applying the same failure criterion on this computation gives a sufficient margin of safety since the computed equivalent stress remains lower than the one obtained in the previously accepted finocyl case. is lower than previous acceptable case. Moreover, since pressure loading is higher than required pressure level, the

computation is conservative and the structural assessment supports the feasibility of the found optimum for B5 within the adopted modeling assumptions.

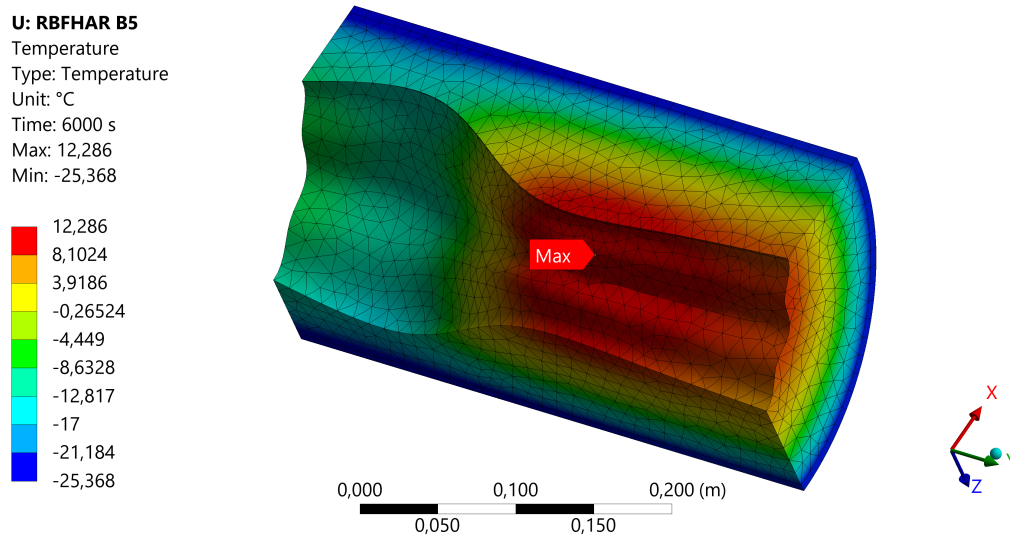


Fig. 4.27 RBF-HAR optimum B5 temperature distribution in grain and casing

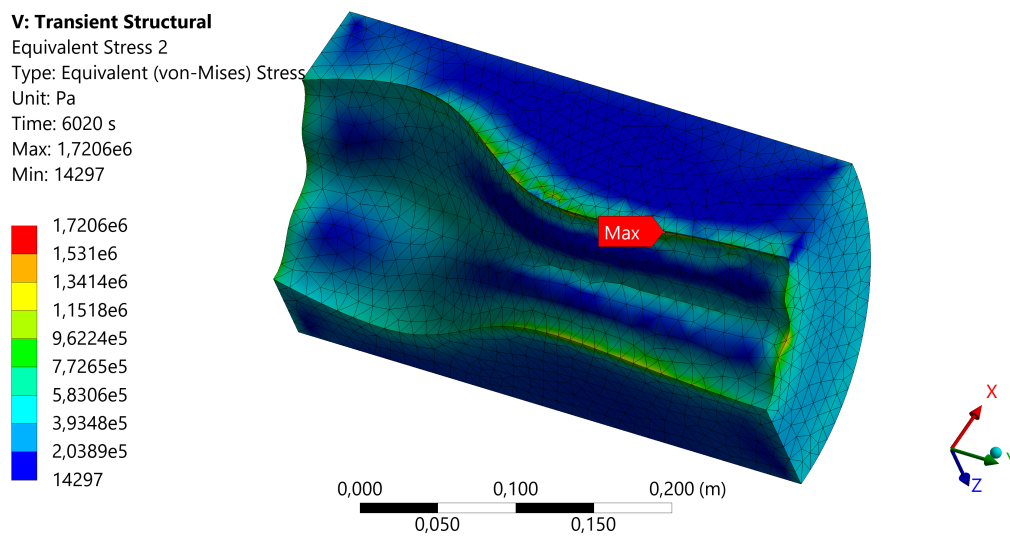


Fig. 4.28 RBF-HAR optimum B5 stress distribution in grain

4.5 Surrogate applicability

The suitability of surrogate modeling for optimal search requires careful evaluation. In the performed test, datasets were generated using at least 3000 LS evaluations, after which PSO was executed with a total of 1500 evaluations. Under these conditions, it is feasible to directly employ level-set evaluations within the PSO procedure, even with a number of evaluations exceeding those used in the optimization run of the test. This conclusion, however, is valid only for the specific setup considered and cannot be regarded as generally applicable. A key aspect to be examined is the number of design variables, as it directly affects the number of sample evaluations needed to properly represent the design space. A design space of limited dimensionality, such as that adopted in the RBF-Cylindrical configuration, can be adequately described using a smaller number of LS evaluations. When the number of LS evaluations required in the offline phase is lower than the number of PSO evaluations, the adoption of surrogate modeling represents a viable option. By defining t_{LS} as the time needed for a single LS evaluation, t_S as the time required for a surrogate evaluation, N_{OPT} as the number of PSO evaluations, and N_T as the number of evaluations used for training, the condition of equal total computational cost between direct and surrogate-based approaches can be expressed as

$$t_{LS}N_{OPT} = t_{LS}N_T + t_S N_T \quad (4.10)$$

Since $t_S \ll t_{LS}$, the cost associated with the online phase is negligible and, as previously noted, surrogate models are advantageous when $N_T < N_{OPT}$. It is worth noting that this assumption becomes increasingly justified as the fidelity of the Level Set simulation grows. Higher-resolution grids or higher-order integration schemes significantly increase t_{LS} while leaving t_S essentially unchanged, further widening the computational gap between direct and surrogate evaluation and reinforcing the practical advantage of the surrogate based approach in high-fidelity design contexts. A limitation, however, lies in the difficulty of estimating in advance the number of evaluations required for both the training and optimization phases. In addition, the manner in which the design tool is employed must be considered. Because the computational burden of surrogate models is largely concentrated in the offline phase and the online cost is negligible, this approach is *configuration-specific* and must be incurred only when a new design configuration is analyzed. Conversely, direct

evaluations are *target-specific*, as they are independent of the configuration but must be repeated for each optimized target. By introducing N_E as the number of evaluated targets, the cost balance can be reformulated as

$$N_E t_{LS} N_{OPT} = t_{LS} N_T + N_E t_S N_T \quad \Rightarrow \quad N_E = \frac{N_T}{N_{OPT}} \quad (4.11)$$

Equivalently, the condition $N_T \leq N_E N_{OPT}$ ensures a reduction in computational cost when using surrogate models. To provide an explicit quantitative breakdown for the specific configurations evaluated in this work, the offline dataset generation required $N_T = 3000$ evaluations, while the direct optimization run required $N_{OPT} = 1500$ evaluations. Substituting these specific values yields a break-even point of $N_E = 2$. Consequently, for a single optimization target ($N_E = 1$), the direct Level Set approach is computationally preferable, requiring half the total evaluations. However, if the designer needs to explore three or more different target pressure profiles ($N_E > 2$) using the same geometric baseline, the surrogate modeling approach immediately provides a clear computational advantage. Therefore, in situations requiring multiple optimization cycles across different targets, such as during early design stages where target specifications may change, N_E can be sufficiently large to justify surrogate training. It should also be noted that modifications to constraints, resulting in changes to the bounds of design parameters, necessitate retraining and may reduce the advantages of surrogate modeling even for large values of N_E . Furthermore, instead of generating datasets through LS evaluations, existing data obtained from high-fidelity burnback simulations or firing tests can be employed. When such data are available, the offline computational cost is effectively eliminated, making surrogate modeling particularly advantageous. Finally, owing to the flexibility of Pymoo, direct evaluations have been integrated into the optimization phase, allowing the optimization framework to operate without the use of meta-models. Depending on the specific design problem and application, the designer can therefore select the most appropriate evaluation strategy.

Chapter 5

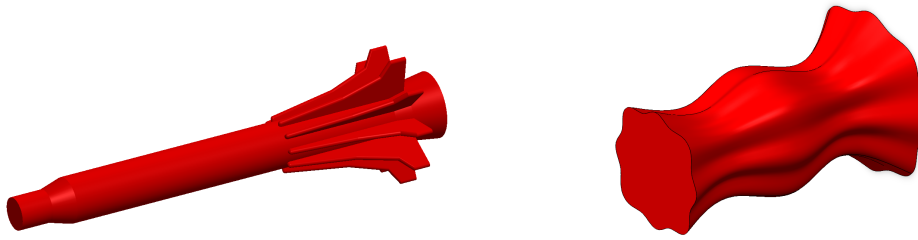
Application

After the selection and testing of surrogate models and optimization strategies, a real life case study application is analyzed to better evaluate the influence of non-uniform ballistic properties and to which extent they can be leveraged. The chapter contains the evaluation of a rescaled and normalized version of Vega's second stage Zefiro23 solid rocket grain geometry, exploring all the different strategies previously proposed to find a new geometric-ballistic configuration with comparable performance. The different geometries are also analyzed to evaluate their structural performances and compared to assess advantages in using different configuration for the mission.

5.1 Zefiro 23-like geometry

The case study is conducted on a Zefiro 23-like solid rocket grain. The grain geometry is inspired by the macroscopic architectural features of the actual Zefiro 23 motor, based on available literature, but has been spatially rescaled and geometrically simplified to ensure consistency with the design envelope of the surrogate training dataset. The motor's pressure-time curve shape is selected as the target for optimization, with the curve being sampled from reference [102] at 20 time snapshots. The reference values reported are non-dimensional and have been rescaled by requiring a MEOP of 35 bar and a maximum burning time of 20 seconds. These values are consistent with the imposed design envelope defining the minimum and maximum allowable grain dimensions, i.e. this level of pressure is attainable for the required burning time by configurations in the training dataset. Since the objective of the optimization is to

achieve a given pressure-time curve shape, the chosen scaling does not affect the validity of the approach. The resulting configuration therefore represents a normalized surrogate scenario rather than a direct representation of the actual Zefiro23 ballistic performance, and will be referred to as the *Zefiro23-like* geometry throughout this section. The curve obtained with these hypotheses leads to requirements reported in Table 5.1, with the same ambient pressure and specific heat values used in the test applications. To achieve this type of curve, Zefiro23-like grain features a specific and relatively complex finocyl shape (as shown in Figure 5.1). In this part of the study, the goal is to obtain similar performances using a simplified geometry by leveraging the ballistic distribution, since in the application on target set it was evident the great extent to which RBF can alter grain performances. In particular, finocyl-RBF and harmonic-RBF configuration will run to find optima configurations. Furthermore, a deeper level of simplification is achieved by coupling the RBF distribution with a cylindrical geometry, extending the geometric simplification, in a similar manner to dual-boost cylindrical grain evaluated during LS solver non-uniform ballistic development in Section 2.3. The cylindrical-RBF configuration is described only by RBF values, with the same design choices made for previous cases. Here, the inner radius of the cylinder is the second center value of the distribution, while the first center is located in the motor axis and the third is placed at casing radius. Hence, these two values are fixed, while other design parameters are the three weights of the distribution and σ . Moreover, no variation is considered along the motor axis. Due to the significant geometric simplification, the number of design parameters is substantially reduced, resulting in a total of six design parameters, including the throat area A_t . This new configuration is summarized in Table 5.2.



(a) Zefiro23-like finocyl geometry, adapted from [102]

(b) Port geometry example obtained within the selected envelope

Fig. 5.1 Geometry comparison. Different grain geometries can achieve comparable performance levels by suitably altering the ballistic distribution.

Target	I_{tot} , kNs	t_b , s	$*A_t$, cm ²	$\frac{p_e}{p_c}$	D , m	L , m	L_m , m	P_{max} , bar
Zefiro23-like	79.59	20.0	13.49	0.01	0.3	0.4	0.4	33.0

Table 5.1 Requirements computed for Zefiro23-like pressure curve. Note that A_t is now a design parameter

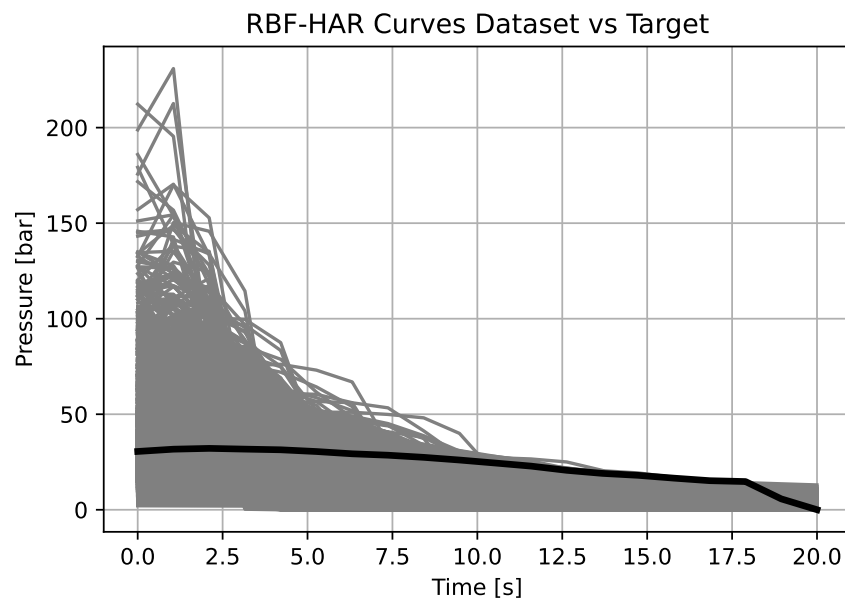


Fig. 5.2 Zefiro23-like target pressure–time curve (black) compared against the RBF-HAR dataset

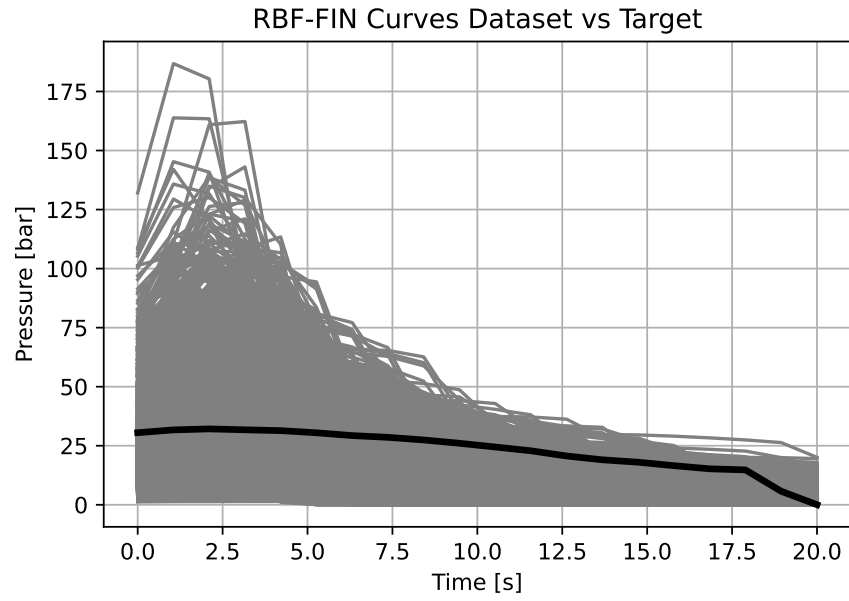


Fig. 5.3 Zefiro23-like target pressure–time curve (black) compared against the RBF-FIN dataset

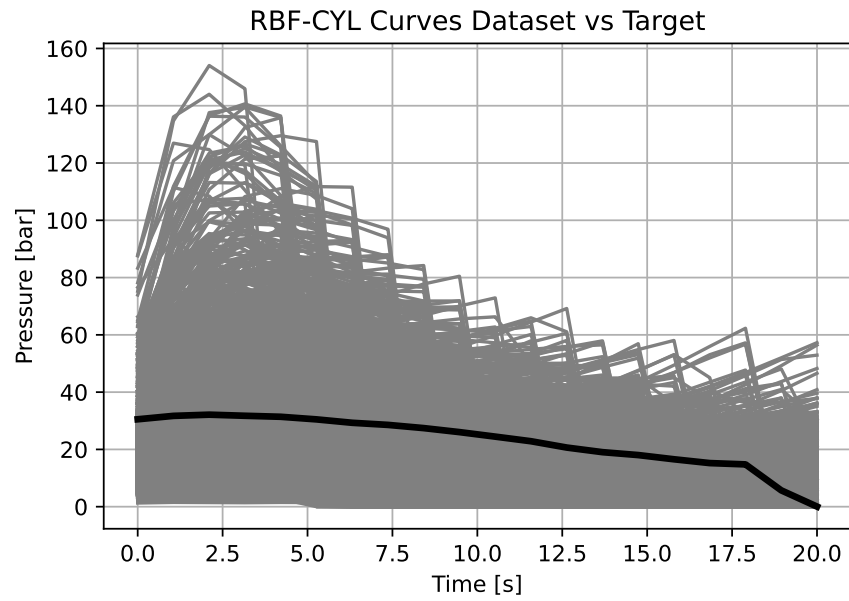


Fig. 5.4 Zefiro23-like target pressure–time curve (black) compared against the RBF-CYL dataset

As in previous cases, a dataset of 3,000 samples was generated for the RBF-cylinder configuration and used to train the surrogate model. Figures 5.2, 5.3,

5.4 present the selected target alongside the previously computed datasets and the new cylindrical-RBF dataset. Figure 5.5 illustrates the optimization results for the Zefiro23-like pressure-time curve across the three design approaches, while Table 5.3 resumes optimum vectors and computed performances. In this case, the selected target is well approximated by the optimal solutions, with minor deviations in the tail-off phase, while all three approaches satisfy the design constraints. As previously observed, the computed specific impulse values (170–177 s) are significantly lower than those expected for a full-scale Zefiro 23-class motor operating in vacuum (285 s). This discrepancy is a direct and physical consequence of the specific boundary conditions of this test case, since the performance calculation was performed at sea-level ambient pressure ($P_{amb} = 1\text{ atm}$) on a geometrically rescaled configuration with a reduced MEOP of 35 bar. Operating a nozzle at significantly reduced chamber pressure while discharging into a 1 atm environment induces severe overexpansion, heavily penalizing the pressure effect on C_F and consequently depressing the computed Isp. This boundary condition was not representative of actual operational conditions and was retained consistently across all configurations to ensure a fair comparative evaluation. The primary objective of this case study is to demonstrate the inverse design capability of the framework, specifically its ability to manipulate geometric and ballistic distributions to match a target pressure-time profile across diverse configurations. This objective is successfully achieved. The absolute performance metrics in Table 5.1 reflect the scaled, off-design scenario and should not be interpreted as representative of full-scale Zefiro23 vacuum performance. Evaluation under vacuum conditions and optimized nozzle expansion, by altering the nozzle and environmental requirements of the optimization previously described, is identified as a natural next step, but falls outside the scope of the present work.

	Parameters	Values and limits
Geometric	r_{inner}	$r_{c,2}$
	Casing Radius	15 <i>cm</i>
	Throat Area	[12,28] <i>cm</i> ²
	Grain Length, L	40 <i>cm</i>
Fixed/Ballistic	w_1	[4e-05,12.5e-05]
	w_2	[3e-05,8.5e-05]
	w_3	[2e-05,4.5e-05]
	$r_{c,1}$	0 <i>cm</i>
	$r_{c,2}$	[3,8] <i>cm</i>
	$r_{c,3}$	15 <i>cm</i>
	σ	[2,7.5]

Table 5.2 Parameters Table RBF-Cylinder

Configuration	Performances for Zefiro23-like optimization
RBF-Harmonic	Performances: $I_{sp} = 177.39 \text{ s}; \quad t_b = 20.0 \text{ s}; \quad I_{tot} = 72901.31 \text{ Ns}$
RBF-Fin	Performances: $I_{sp} = 170.70 \text{ s}; \quad t_b = 20.0 \text{ s}; \quad I_{tot} = 78713.21 \text{ Ns}$
RBF-Cylinder	Performances: $I_{sp} = 176.64 \text{ s}; \quad t_b = 20 \text{ s}; \quad I_{tot} = 81450.25 \text{ Ns}$

Table 5.3 Comparison of Performance on Zefiro23-like Optimization

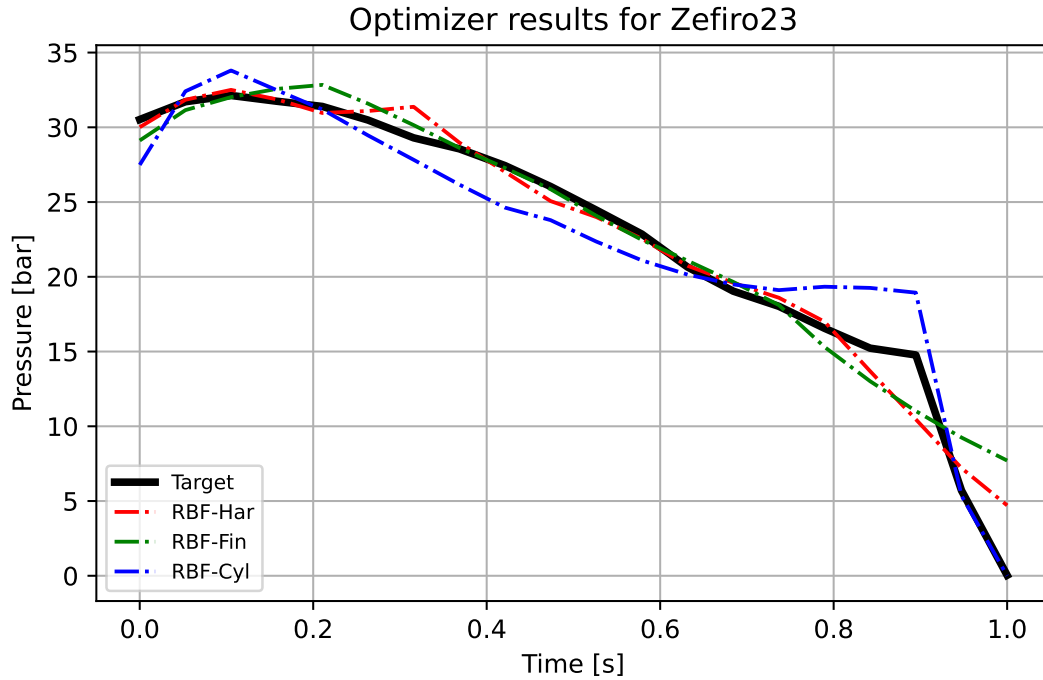


Fig. 5.5 Optima on Zefiro23-like target. Exploiting ballistic distribution leads to convergent result for different geometries

The possibility to alter the ballistic properties to change the shape of the grain while maintaining similar performance allows to employ grain shapes with lower stress concentrations with respect to a classical finocyl shape. The desired pressure-time curve could be achieved by employing variable ballistic instead of particular geometrical features, and a simplified geometry could result in a lower maximum stress inside the grain. To evaluate this aspect, the obtained optima for RBF-HAR case, the smoothest geometry among the proposed, is compared to original Zefiro23-like shape applying the structural model utilized in this work (2.4). The structural comparison follows the same hypotheses and uses the same parameters of the analysis conducted for finocyl evaluation in 4.4.1, under the assumptions of employing the same materials for grain and casing. The same constraints and contacts, introduced to account, in a simplified model, for the restraining effect of motor hardware not explicitly included at this stage, and the same thermal cycle and pressure ramping internal loading are utilized. This setup allows attributing the observed differences in stress concentration primarily to geometric effects. Figure 5.6 and Figure 5.7 show the geometrical configuration of the grain and casing for Zefiro23-like and Optimum, respectively. Since the same assumptions are made to compute the two

geometries, their comparison is meaningful within the adopted modeling assumptions and highlights the differences in the stress state arising from the two configurations.

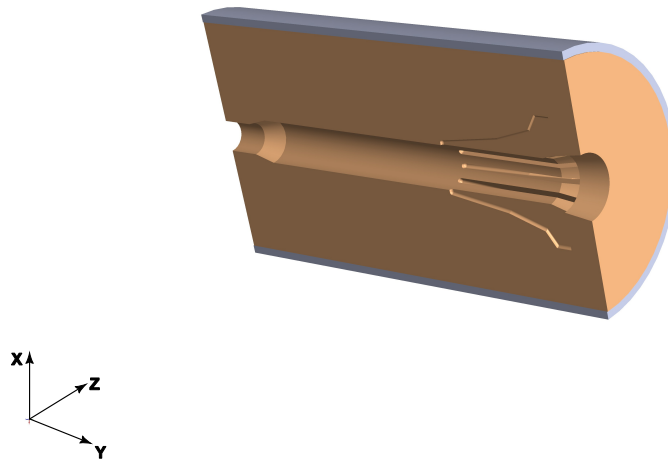


Fig. 5.6 Zefiro23-like geometry utilized for structural comparison. Note that the section plane is tilted to align with fin axis and show the cavity.

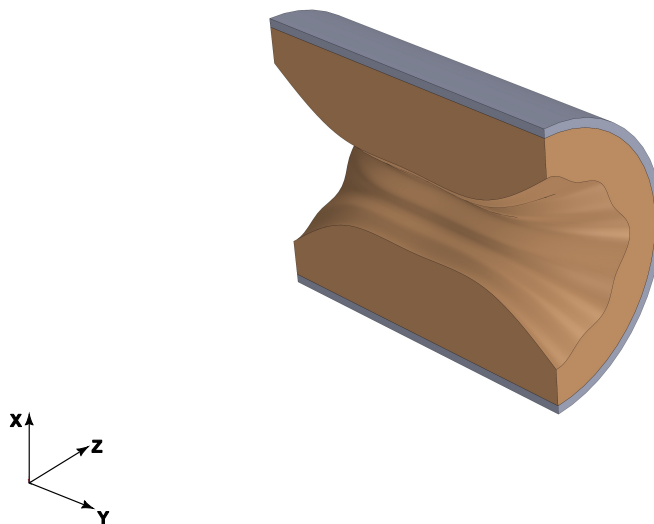


Fig. 5.7 RBF-HAR optimum geometry utilized for structural comparison

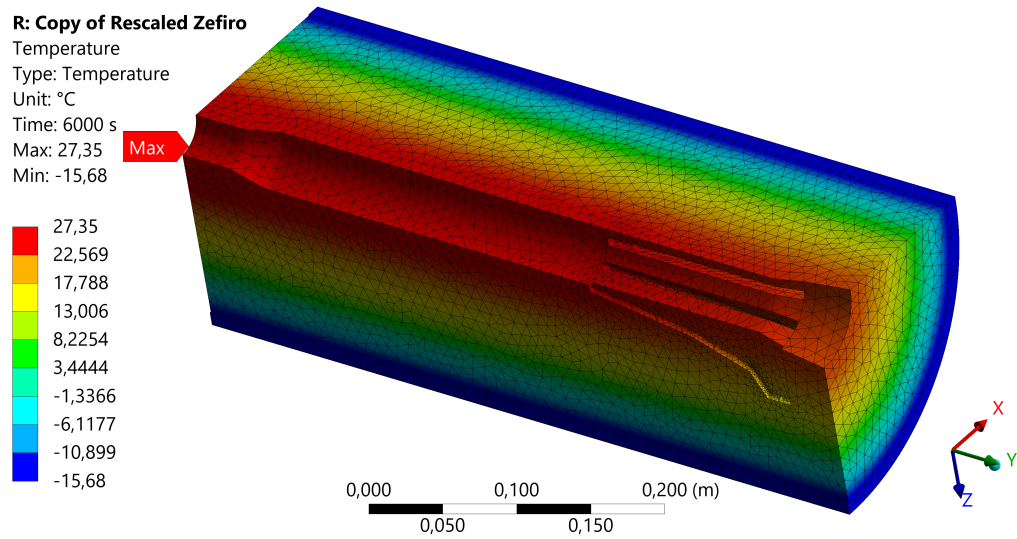


Fig. 5.8 Zefiro geometry temperature distribution

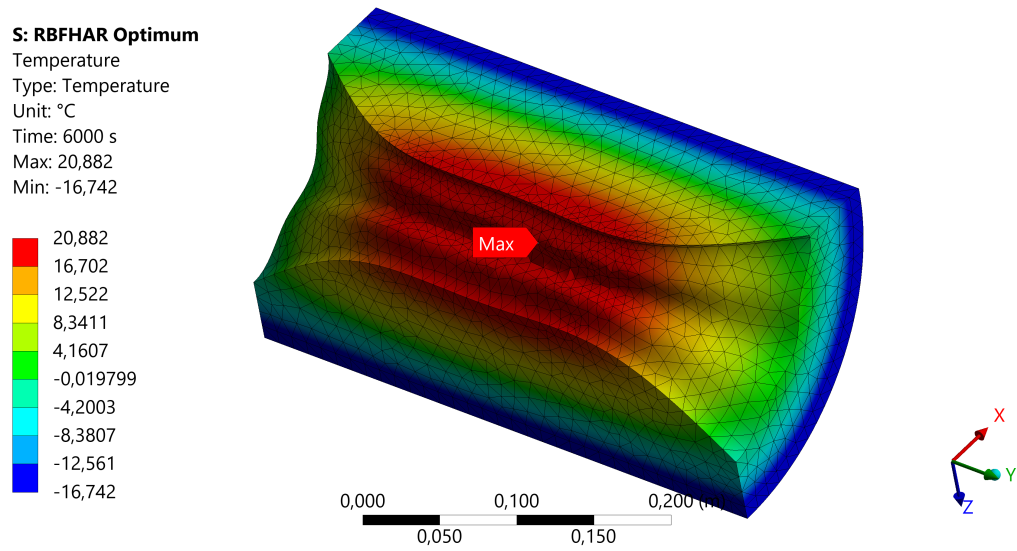


Fig. 5.9 RBF-HAR optimum geometry temperature distribution

Temperature profile inside the grain is reported in Figure 5.8 and Figure 5.9, while equivalent stress distribution is presented in Figure 5.10 and Figure 5.11. Also in this case, to allow for a proper comparison, harmonic port optimum and Zefiro grain are subject to the same thermal cycle already used in previous computation. For the specific dimensions of the grains, this thermal cycle produces higher temperature in the grain with respect to previous cases. However, this choice is linked to operational conditions, which are selected in the design problem definition, and in this context can be and, within the scope of this work, it is treated as a prescribed operating condition for the comparative assessment. Nonetheless, applying a coherent state to the two grains allows for a proper comparison between the geometries. The obtained results show that equivalent maximum stress computed in the harmonic port grain at the end of pressurization, is almost two times lower than the one computed in the reference finocyl shape. This reduction is linked to the absence of local features which concentrate stresses, and can constitute a possible advantage of this grain type, provided similar performance is achieved thanks to non-uniform ballistic, which is shown to be effective in enabling alternative grain designs of solid rockets, while meeting the same target performance. However, to provide a fully comprehensive evaluation of the optimized configuration, the volumetric loading fraction of the harmonic geometry must also be considered. A quantitative evaluation has been carried out for both configurations to assess their spatial efficiency within the defined casing. The reference Zefiro23-like geometry exhibits a highly packed volume, typical of optimized traditional designs, yielding a volumetric loading fraction of approximately 96%. In contrast, the optimized RBF-HAR geometry, while successfully matching the target pressure-time profile and reducing stress concentrations, results in a lower volumetric loading fraction of approximately 76%. This reduction in volumetric efficiency highlights an inherent trade-off of the current mathematical parameterization: while the methodology provides high flexibility in shaping the macroscopic ballistic curve and smoothing local features, it tends to generate wider port volumes, which would negatively impact the structural coefficient of a physical motor. Recognizing this limitation, future developments of the optimization framework should include the volumetric loading fraction not just as an a posteriori evaluation metric, but as an active constraint or a secondary objective function. This could be practically implemented by appropriately constraining the lower bounds of the port profile parametrization, forcing the harmonic functions to explore volumetrically denser configurations. Such an approach would ensure the

generation of geometries that are simultaneously ballistically accurate, structurally smooth, and volumetrically efficient.

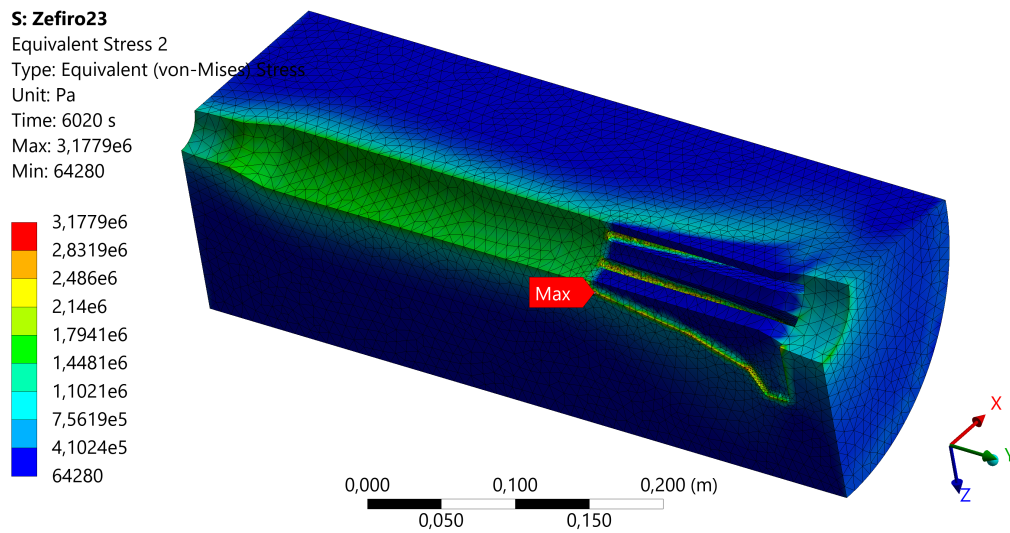


Fig. 5.10 Zefiro geometry stress distribution

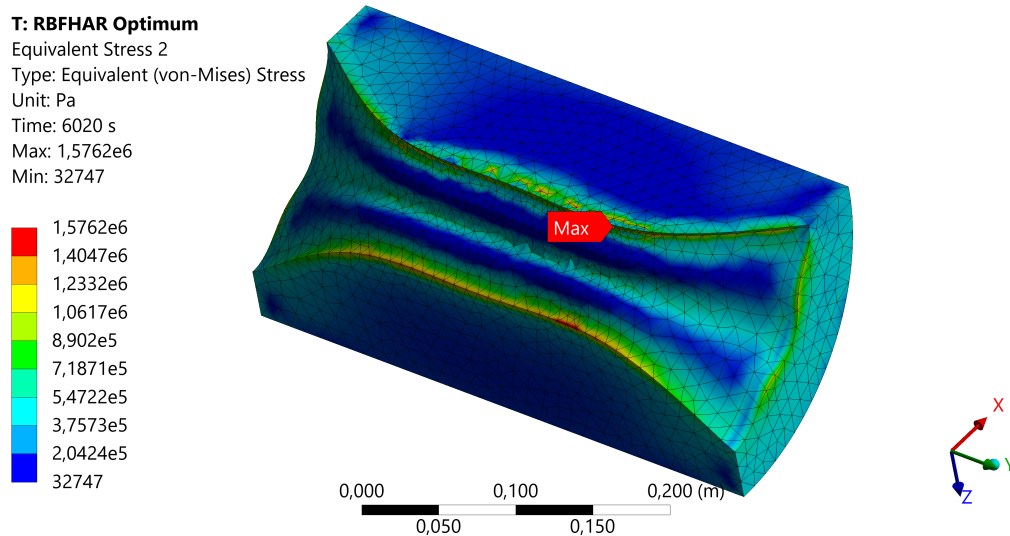


Fig. 5.11 RBF-HAR optimum geometry stress distribution

Chapter 6

Conclusions

This dissertation faced the development of numerical methods and inverse design strategies in the analysis and design of solid rocket motor (SRM) grains, with particular emphasis on configurations enabled by additive manufacturing technologies. The motivation of the work stemmed from the growing interest in non-conventional grain geometries and non-uniform ballistic properties that can be enabled by innovative additive manufacturing techniques for energetic materials, which challenge traditional design assumptions and require new simulation and optimization tools. A comprehensive numerical framework has been developed to simulate SRM internal ballistics while accounting for complex three-dimensional grain geometries and spatially varying burning rates. The proposed approach relies on Level Set Methods to implicitly represent the evolving burning surface and to accurately compute its geometrical features throughout the burnback process. The geometric module has been coupled with a zero-dimensional chamber model, enabling the evaluation of pressure–time histories under non-uniform ballistic conditions with reasonable trade-off between computing effort and accuracy. Different strategies for distance field initialization and surface transport were investigated and compared, highlighting the advantages and limitations of structured Cartesian grids and adaptive octree-based finite volume discretizations.

Beyond non-uniform burnback simulation, the work introduced an inverse design framework aimed at identifying optimal grain configurations capable of meeting prescribed pressure–time requirements. To make this process computationally viable, surrogate models based on deep neural network, adaptive basis functions and reduced-

order techniques were developed and assessed. These models proved effective in approximating the pressure response of complex configurations and enabled the use of stochastic optimization algorithms for the exploration of large design spaces. The structural assessment of the grain, based on a linear viscoelastic characterization of the solid grain, has been introduced as validation of the optimum design on top of the design cycle. The methodology has been validated through a series of test cases of increasing complexity, including canonical geometries, configurations with spatially varying burning rates, and applications based on experimental data. The application to realistic case study, such as the Zefiro 23 motor, demonstrated that different grain geometries and ballistic distributions can achieve comparable performance levels when appropriate design flexibility is introduced. These results confirm that non-uniform ballistic properties, when properly exploited, represent a powerful additional degree of freedom in SRM design. Structural model has been employed to compare alternative geometries available to meet the same target. The obtained results shows the advantage in using smoother geometries which can mimic the behavior of more complex shapes by leveraging ballistic non-uniformity.

From a broader perspective, the results of this work suggest that the combination of additive manufacturing technologies with advanced numerical design tools has the potential to significantly expand the SRM performance envelope. The capability to locally tailor grain properties and geometry opens new possibilities for thrust shaping, performance optimization, and structural load management that are difficult or impossible to achieve with conventional manufacturing techniques. However, despite these promising outcomes, some limitations remain. The chamber flow model adopted in this work is intentionally simplified to favor computational efficiency, and more detailed fluid-dynamic descriptions could be considered to capture secondary effects, especially for harmonic port grains where 0D model reduces its accuracy. Additionally, the thermo-mechanical analyses has been limited to simplified geometrical models. Moreover, further experimental validation is required to fully assess the feasibility of the proposed configurations in terms of ballistic properties distribution. Additional studies on material characterization, properties and manufacturing process should give additional information to tune ballistic distributions and set material properties of the propellant. Nonetheless, the overall simulation environment can be easily tuned to account for new information provided by additional experimentation.

Future developments may include the integration of higher-fidelity fluid and structural models, the study of new baseline geometries and distribution, and the

experimental validation of additively manufactured grains with spatially varying properties. Furthermore, the proposed inverse design methodology could be extended to multi-objective optimization problems, accounting simultaneously for performance, structural integrity, and manufacturability constraints in the optimization routine.

In conclusion, this work contributes a unified numerical and optimization framework for the analysis and design of advanced solid rocket motor grains. By bridging modern manufacturing concepts with physics-based simulation and data-driven design tools, it lays the groundwork for future developments in flexible and high-performance solid propulsion systems.

References

- [1] Keenan Crane, Clarisse Weischedel, and Max Wardetzky. Geodesics in heat: A new approach to computing distance based on heat flow. *ACM Transactions on Graphics (TOG)*, 32(5):1–11, 2013.
- [2] George P. Sutton and Oscar Biblarz. *Rocket Propulsion Elements*. John Wiley & Sons, Hoboken, 9th edition, 2016.
- [3] Stephen D. Heister, William E. Anderson, Timothée L. Pourpoint, and R. Joseph Cassady. *Rocket propulsion*, volume 47. Cambridge University Press, 2019.
- [4] Herve Austruy, Marco Biagioni, and Yves Pelipenko. Improvement in propellant and process for Ariane 5 boosters. In *34th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, page 3847, 1998.
- [5] Jean Thépénier and Gilles Fonblanc. Advanced technologies available for future solid propellant grains. *Acta Astronautica*, 48(5-12):245–255, 2001.
- [6] Kinga Lysien, Agnieszka Stolarczyk, and Tomasz Jarosz. Solid propellant formulations: A review of recent progress and utilized components. *Materials*, 14(21):6657, 2021.
- [7] World Health Organization Anon, International Agency for Research on Cancer. IARC monographs on the identification of carcinogenic hazards to humans, 2022. [Online; accessed 18/10/2022].
- [8] Claudia Bolognesi, Xaver Baur, Boleslaw Marczyński, Hannu Norppa, Ovnair Sepai, and Gabriele Sabbioni. Carcinogenic risk of toluene diisocyanate and 4, 4'-methylenediphenyl diisocyanate: Epidemiological and experimental evidence. *Critical Reviews in Toxicology*, 31(6):737–772, 2001.
- [9] Kumar Kanishka and Bappa Acherjee. Revolutionizing manufacturing: A comprehensive overview of additive manufacturing processes, materials, developments, and challenges. *Journal of Manufacturing Processes*, 107:574–619, 2023.
- [10] Longfei Zhou, Jenna Miller, Jeremiah Vezza, Maksim Mayster, Muhammad Raffay, Quentin Justice, Zainab Al Tamimi, Gavyn Hansotte, Lavanya Devi

- Sunkara, and Jessica Bernat. Additive manufacturing: a comprehensive review. *Sensors*, 24(9):2668, 2024.
- [11] Joel C. Najmon, Sajjad Raeisi, and Andres Tovar. Review of additive manufacturing technologies and applications in the aerospace industry. *Additive manufacturing for the aerospace industry*, pages 7–31, 2019.
- [12] Nikita V. Muravyev, Konstantin A. Monogarov, Uwe Schaller, Igor V. Fomenkov, and Alla N. Pivkina. Progress in additive manufacturing of energetic materials: Creating the reactive microstructures with high potential of applications. *Propellants, Explosives, Pyrotechnics*, 44(8):941–969, 2019.
- [13] Kai Miao, Min Tang, Zichen Wang, Shixiong Song, Jiawei Shi, Yang Ran, Shuang Gao, Heng Xue, Dichen Li, and Quanbin Ren. Additive manufacturing of solid propellants: Exploring the frontier of solid propulsion systems. *Additive Manufacturing Frontiers*, page 200263, 2025.
- [14] Michiel H. Straathof, Chris A. van Driel, Joost N.J. van Lingen, Bastiaan L.J. Ingenhuth, A. Tessa ten Cate, and Hessel H. Maalderink. Development of propellant compositions for vat photopolymerization additive manufacturing. *Propellants, Explosives, Pyrotechnics*, 45(1):36–52, 2020.
- [15] Xu Zhou, Yaofeng Mao, Dawei Zheng, Lin Zhong, Ruihao Wang, Bing Gao, and Dunju Wang. 3D printing of RDX-based aluminized high explosives with gradient structure, significantly altering the critical dimensions. *Journal of Materials Science*, 56(15):9171–9182, 2021.
- [16] Levi Kirby, H. S. Udaykumar, and Xuan Song. Pressure-assisted binder jetting for additive manufacturing of mock energetic composites. *Propellants, explosives, pyrotechnics*, 49(1):e202300175, 2024.
- [17] M. S. McClain, A. Afriat, B. J. Montano, S. Ray, J. Rhoads, I. E. Gunduz, and S. F. Son. Investigation of additively manufactured layered composite solid propellant. In *AIAA Scitech 2020 Forum*, 2020. AIAA Paper 2020-1427.
- [18] M. S. McClain, A. Afriat, B. J. Montano, J. F. Rhoads, I. E. Gunduz, and S. F. Son. Dynamic combustion of functionally graded additively manufactured composite solid propellant. *Journal of Propulsion and Power*, 37(5):725–732, 2021.
- [19] Yongjin Chen, Shuhong Ba, and Hui Ren. Additive manufacturing of a special-shaped energetic grain and its performance. *Micromachines*, 12(12):1509, 2021.
- [20] Nihan Chen, Chunlin He, and Siping Pang. Additive manufacturing of energetic materials: Tailoring energetic performance via printing. *Journal of Materials Science & Technology*, 127:29–47, 2022.

- [21] Uwe Scheithauer, Eric Schwarzer, Hans-Jürgen Richter, and Tassilo Moritz. Thermoplastic 3D printing—an additive manufacturing method for producing dense ceramics. *International Journal of Applied Ceramic Technology*, 12(1):26–31, 2015.
- [22] R. A. Chandru, N. Balasubramanian, C. Oommen, and B. N. Raghunandan. Additive manufacturing of solid rocket propellant grains. *Journal of Propulsion and Power*, 34(4):1090–1093, 2018.
- [23] M. Li, W. Yang, M. Xu, R. Hu, and L. Zheng. Study of photocurable energetic resin based propellants fabricated by 3D printing. *Materials & Design*, 207, Sept. 2021.
- [24] J. Wejsa. Ardec report. Presentation at the NDIA Armament Systems Forum.
- [25] R. Tillinghast and J. Zunino. Additive manufacturing methods, techniques, procedures, and applications. enabling technologies for military applications. Presentation at the NDIA Armament Systems Forum.
- [26] D. Pastrone, M. Sangermano, S. Garino, and F. Maggi. Composite propellant manufacturing process based on deposition and light-activated polymerization for solid rocket motors, October 20 2020. WO2020212785A1.
- [27] Simone Garino, Paola Antonaci, Dario Pastrone, Marco Sangermano, and Filippo Maggi. Photo-polymerization for additive manufacturing of composite solid propellants. *Acta Astronautica*, 182:58–65, 2021.
- [28] Andrea Galavotti, Camilla Noè, Giovanni Polizzi, Paola Antonaci, Filippo Maggi, Filippo Masseni, and Dario Pastrone. Solid rocket propellant photo-polymerization with an in-house LED-UV prototype. *Polymers*, 15(7):1633, 2023.
- [29] Alessandra Zumbo, Leonardo Stumpo, Paola Antonaci, Andrea Ferrero, Filippo Masseni, Giovanni Polizzi, Giacomo Tetti, and Dario Pastrone. Rheological and mechanical characterization of 3D-printable solid propellant slurry. *Polymers*, 16(5):576, 2024.
- [30] Filippo Masseni, Alessandra Zumbo, Leonardo Stumpo, Giovanni Polizzi, Giacomo Tetti, Tommaso Lusetti, Camilla Noé, Paola Antonaci, Andrea Ferrero, Dario Pastrone, and Filippo Maggi. Advancements in photocurable solid propellants for additive manufacturing. In *AIAA SCITECH 2025 Forum*, 2025. AIAA Paper 2025-2330.
- [31] M. S. McClain, I. E. Gunduz, and S. F. Son. Additive manufacturing of ammonium perchlorate composite propellant with high solids loadings. *Proceedings of the Combustion Institute*, 37(3):3135–3142, 2019.
- [32] James M. Hyman. Numerical methods for tracking interfaces. *Physica D: Nonlinear Phenomena*, 12(1-3):396–407, 1984.

- [33] James A. Sethian. Curvature and the evolution of fronts. *Communications in Mathematical Physics*, 101(4):487–499, 1985.
- [34] Cyril W. Hirt and Billy D. Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of computational physics*, 39(1):201–225, 1981.
- [35] Stanley Osher and Ronald Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Applied Mathematical Sciences. Springer New York, New York, 2003.
- [36] James A. Sethian et al. *Level set methods and fast marching methods*, volume 98. Cambridge Cambridge UP, 1999.
- [37] Frederic Gibou, Ronald Fedkiw, and Stanley Osher. A review of level-set methods and some recent applications. *Journal of Computational Physics*, 353:82–109, 2018.
- [38] bitpit: Open source library for scientific High Performance Computing.
- [39] Ian M. Mitchell. The flexible, extensible and efficient toolbox of level set methods. *Journal of Scientific Computing*, 35:300–329, 2008.
- [40] Mark W. Jones, J. Andreas Bærentzen, and Milos Sramek. 3D distance fields: A survey of techniques and applications. *IEEE Transactions on visualization and Computer Graphics*, 12(4):581–599, 2006.
- [41] J. Andreas Bærentzen. Robust generation of signed distance fields from triangle meshes. In *Fourth International Workshop on Volume Graphics, 2005.*, pages 167–239. IEEE, 2005.
- [42] Towaki Takikawa, Andrew Glassner, and Morgan McGuire. A dataset and explorer for 3D signed distance functions. *Journal of Computer Graphics Techniques*, 11(2), 2022.
- [43] Mark W. Jones. 3D distance from a point to a triangle. *Department of Computer Science, University of Wales Swansea Technical Report CSR-5*, 5, 1995.
- [44] Andreas Bærentzen and Henrik Aanaes. Generating signed distance fields from triangle meshes. 01 2002.
- [45] Sathamangalam R. Srinivasa Varadhan. On the behavior of the fundamental solution of the heat equation with variable coefficients. *Communications on Pure and Applied Mathematics*, 20(2):431–455, 1967.
- [46] Lawrence C Evans. *Partial differential equations*, volume 19. American mathematical society, 2022.
- [47] Keenan Crane, Clarisse Weischedel, and Max Wardetzky. The heat method for distance computation. *Communications of the ACM*, 60(11):90–99, 2017.

- [48] Nicole Feng and Keenan Crane. A heat method for generalized signed distance. *ACM Transactions on Graphics (TOG)*, 43(4):1–19, 2024.
- [49] Nicholas Sharp, Yousuf Soliman, and Keenan Crane. The vector heat method. *ACM Transactions on Graphics (TOG)*, 38(3):1–19, 2019.
- [50] Randall J. LeVeque. *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*. SIAM, 2007.
- [51] Alfio Quarteroni, Riccardo Sacco, and Fausto Saleri. *Numerical mathematics*, volume 37. Springer Science & Business Media, 2006.
- [52] Guang-Shan Jiang and Chi-Wang Shu. Efficient implementation of weighted ENO schemes. *Journal of computational physics*, 126(1):202–228, 1996.
- [53] Ami Harten, Bjorn Engquist, Stanley Osher, and Sukumar R. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes, III. *Journal of computational physics*, 131(1):3–47, 1997.
- [54] Xu-Dong Liu, Stanley Osher, and Tony Chan. Weighted essentially non-oscillatory schemes. *Journal of computational physics*, 115(1):200–212, 1994.
- [55] Stanley Osher and Chi-Wang Shu. High-order essentially nonoscillatory schemes for Hamilton–Jacobi equations. *SIAM Journal on numerical analysis*, 28(4):907–922, 1991.
- [56] Chi-Wang Shu and Stanley Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of computational physics*, 77(2):439–471, 1988.
- [57] S.K. Godunov. Finite difference methods for the computation of discontinuous solutions of the equations of fluid dynamics. *Mat. Sb.*, 47:271–306, 1959.
- [58] Elisabeth Rouy and Agnes Tourin. A viscosity solutions approach to shape-from-shading. *SIAM Journal on Numerical Analysis*, 29(3):867–884, 1992.
- [59] Giovanni Russo and Peter Smereka. A remark on computing distance functions. *Journal of computational physics*, 163(1):51–67, 2000.
- [60] Randall J. LeVeque. *Finite volume methods for hyperbolic problems*, volume 31. Cambridge university press, 2002.
- [61] Guy M. Morton. *A computer oriented geodetic data base and a new technique in file sequencing*. International Business Machines Company, 1966.
- [62] Michel Bergmann, Antoine Fondanèche, and Angelo Iollo. An eulerian finite-volume approach of fluid-structure interaction problems on quadtree meshes. *Journal of Computational Physics*, 471:111647, 2022.

- [63] Antoine Fondaneche. *Interaction fluide-structure dans un dispositif vasculaire actif*. PhD thesis, Université de Bordeaux, 2021.
- [64] William John Coirier. *An adaptively-refined, Cartesian, cell-based scheme for the Euler and Navier-Stokes equations*. PhD thesis, University of Michigan, 1994.
- [65] Yves Coudière, Jean-Paul Vila, and Philippe Villedieu. Convergence rate of a finite volume scheme for a two dimensional convection-diffusion problem. *ESAIM: Mathematical Modelling and Numerical Analysis*, 33(3):493–516, 1999.
- [66] Claire Taymans. *Solving incompressible Navier-Stokes equations on Octree grids: towards application to wind turbine blade modelling*. PhD thesis, Université de Bordeaux, 2018.
- [67] Giovanni Polizzi, Andrea Ferrero, Filippo Masseni, and Dario Pastrone. A numerical method for burnback analysis of UV-cured solid rocket propellant grains. In *AIAA SCITECH 2024 Forum*, page 0635, 2024. AIAA Paper 2024-0635.
- [68] Michael A. Willcox, M. Quin Brewster, Kung-Chyun Tang, D. Scott Stewart, and Igor Kuznetsov. Solid rocket motor internal ballistics simulation using three-dimensional grain burnback. *Journal of Propulsion and Power*, 23(3):575–584, 2007.
- [69] V. Borie, J. Brulard, and G. Lengelle. Aerothermochemical analysis of carbon-carbon nozzle regression in solid-propellant rocket motors. *Journal of Propulsion and Power*, 5(6):665–673, 1989.
- [70] Daniele Bianchi, Francesco Nasuti, and Emanuele Martelli. Coupled analysis of flow and surface ablation in carbon-carbon rocket nozzles. *Journal of Spacecraft and Rockets*, 46(3):492–500, 2009.
- [71] Daniele Bianchi and Agostino Neri. Numerical simulation of chemical erosion in Vega solid-rocket-motor nozzles. *Journal of Propulsion and Power*, 34(2):482–498, 2018.
- [72] R.A. Ellis. Solid rocket motor nozzles – NASA space vehicle design criteria (chemical propulsion). *NASA SP-8115*, 1975.
- [73] R. Geisler and C. Beckman. The history of the BATES motors at the air force rocket propulsion laboratory. In *34th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, page 3981, 1998.
- [74] A. Whitehouse, R. Couturier, P. Donguy, H. Besser, and L. Meyer. Structural assessment of solid propellant grains. Technical Report AGARD-AR-350, Advisory Group for Aerospace Research and Development (AGARD), Neuilly-sur-Seine, France, 1997.

- [75] Ceyhun Tola and Melike Nikbay. Solid rocket motor propellant optimization with coupled internal ballistic–structural interaction approach. *Journal of Spacecraft and Rockets*, 55(4):936–947, 2018.
- [76] James S. Noel. *Solid propellant grain structural integrity analysis*. National Aeronautics and Space Administration, 1973.
- [77] J. Edmund Fitzgerald and William L. Hufferd. Handbook for the engineering structural analysis of solid propellants. Technical report, University of Utah, 1971.
- [78] David Russell Bland. *The theory of linear viscoelasticity*. Courier Dover Publications, 2016.
- [79] Marcel Barrère, André Jaumotte, Baudouin Fraeijs de Veubeke, and Jean Vandenkerckhove. *Rocket Propulsion*. Elsevier, Amsterdam, 1960.
- [80] G. Püskülcü and A. Ulas. 3-D grain burnback analysis of solid propellant rocket motors: Part 1–ballistic motor tests. *Aerospace Science and Technology*, 12(8):579–584, 2008.
- [81] G. Püskülcü and A. Ulas. 3-D grain burnback analysis of solid propellant rocket motors: Part 2–modeling and simulations. *Aerospace Science and Technology*, 12(8):585–591, 2008.
- [82] Ali Kamran and Liang Guozhu. An integrated approach for optimization of solid rocket motor. *Aerospace Science and Technology*, 17(1):50–64, 2012.
- [83] Wu Zeping, Wang Donghui, Zhang Weihua, Patrick Okolo. N, and Fei Yang. Solid-rocket-motor performance-matching design framework. *Journal of Spacecraft and Rockets*, 54(3):698–707, 2017.
- [84] Tyler M. Sheils. *Tapered Grain Geometry and Statistical Learning for Solid Rocket Motor Simulation*. PhD thesis, Auburn University, 2023.
- [85] Georg Poppe, Patrick Koch, Ognjan Bozic, and Daniel Lancelle. Optimization of finocyl grain geometries of solid rocket boosters. In *European Conference for Aerospace Sciences (ISSN: 1270-9638)*, 2015.
- [86] Kurt Maute and Subhayan De. Shape and material optimization of problems with dynamically evolving interfaces applied to solid rocket motors. *Structural and Multidisciplinary Optimization*, 65(8):218, 2022.
- [87] Saeed Mesgari, Mehrdad Bazazzadeh, and Alireza Mostofizadeh. Finocyl grain design using the genetic algorithm in combination with adaptive basis function construction. *International Journal of Aerospace Engineering*, 2019(1):3060173, 2019.

- [88] Wang Dong-Hui, Fei Yang, Hu Fan, and Zhang Wei-Hua. An integrated framework for solid rocket motor grain design optimization. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 228(7):1156–1170, 2014.
- [89] W. T. Brooks. Solid propellant grain design and internal ballistics. NASA Special Publication NASA-SP-8076, National Aeronautics and Space Administration, Washington, D.C., 1972.
- [90] Alexander Forrester, Andras Sobester, and Andy Keane. *Engineering Design via Surrogate Modelling: a Practical Guide*. John Wiley & Sons, Hoboken, 2008.
- [91] Hyung Suk Lee, Soon Wook Kwon, and Joon Sang Lee. Neural networks for the burn back performance of solid propellant grains. *Aerospace Science and Technology*, 137:108283, 2023.
- [92] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
- [93] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- [94] Gints Jekabsons and Y. Zhang. Adaptive basis function construction: an approach for adaptive building of sparse polynomial regression models. *Machine Learning*, 1(10):127–155, 2010.
- [95] Anindya Chatterjee. An introduction to the proper orthogonal decomposition. *Current science*, pages 808–817, 2000.
- [96] Nicola Demo, Marco Tezzele, and Gianluigi Rozza. EZyRB: Easy reduced basis method. *Journal of Open Source Software*, 3(24):661, 2018.
- [97] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, volume 4, pages 1942–1948. iee, 1995.
- [98] Ahmet Göçmez, Cevat Erişken, Ülkü Yilmazer, Fikret Pekel, and Saim Özkar. Mechanical and burning properties of highly loaded composite propellants. *Journal of Applied Polymer Science*, 67(8):1457–1464, 1998.

- [99] Lorenzo Federici, Alessandro Zavoli, Guido Colasurdo, Lucandrea Mancini, and Agostino Neri. Integrated optimization of first-stage SRM and ascent trajectory of multistage launch vehicles. *Journal of Spacecraft and Rockets*, 58(3):786–797, 2021.
- [100] Xiangdong Chen, Jianwei Lai, Xin-long Chang, Youhong Zhang, Lei Zhang, and Chao Wang. Compressive mechanical properties of HTPB propellant at low temperatures and high strain rates. *Results in physics*, 7:4079–4084, 2017.
- [101] Martin Dietrich Buhmann. Radial basis functions. *Acta numerica*, 9:1–38, 2000.
- [102] Enrico Cavallini. *Modeling and numerical simulation of solid rocket motors internal ballistics*. PhD thesis, Università degli Studi di Roma" La Sapienza", 2010.

Appendix A

In this appendix some visualization of geometries and respective pressure time curves are presented to better visualize databases utilized to train surrogate models. Here four random samples are extracted from database and shown. Moreover, optimized geometries for Zefiro23 application are presented for Finocyl and Harmonic test case, to better show the difference in geometries and the influence of the ballistic properties in altering pressure response of the grain.

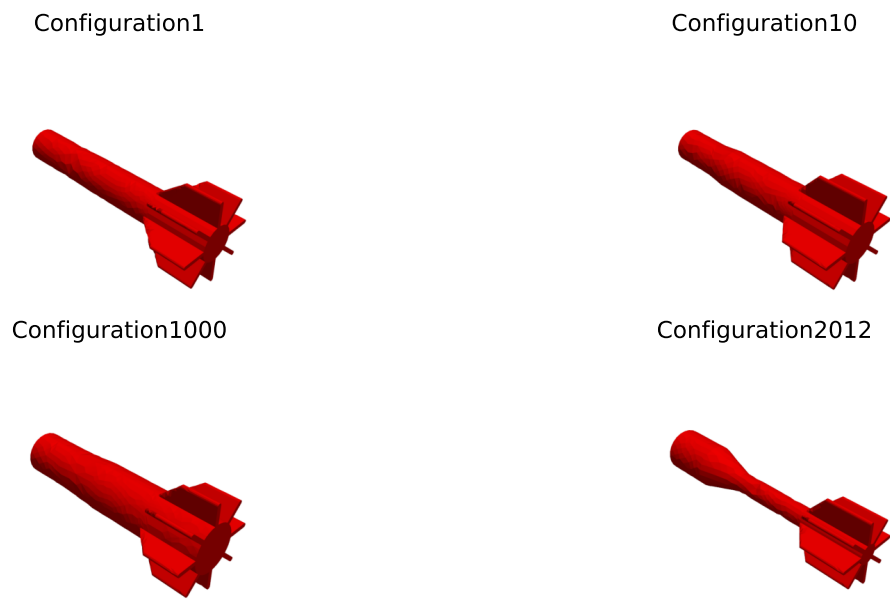


Fig. A.1 Visualization of Finocyl geometry with polynomial ballistic distribution free length case, comparison for random samples

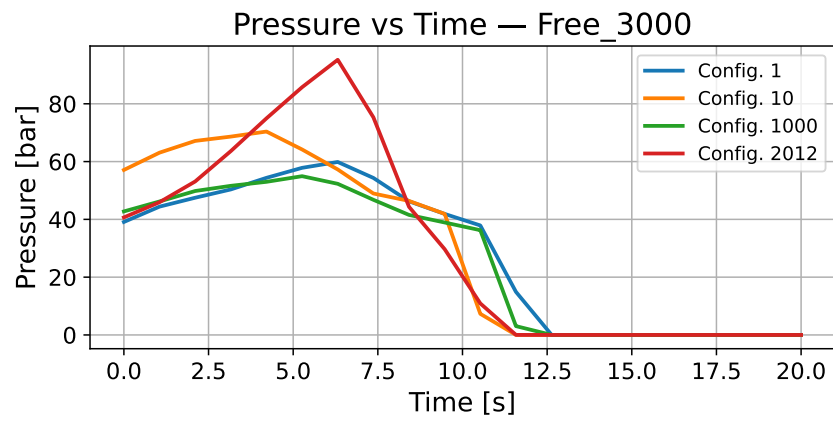


Fig. A.2 Pressure time visualization from dataset for respective Finocyl configuration, free length case

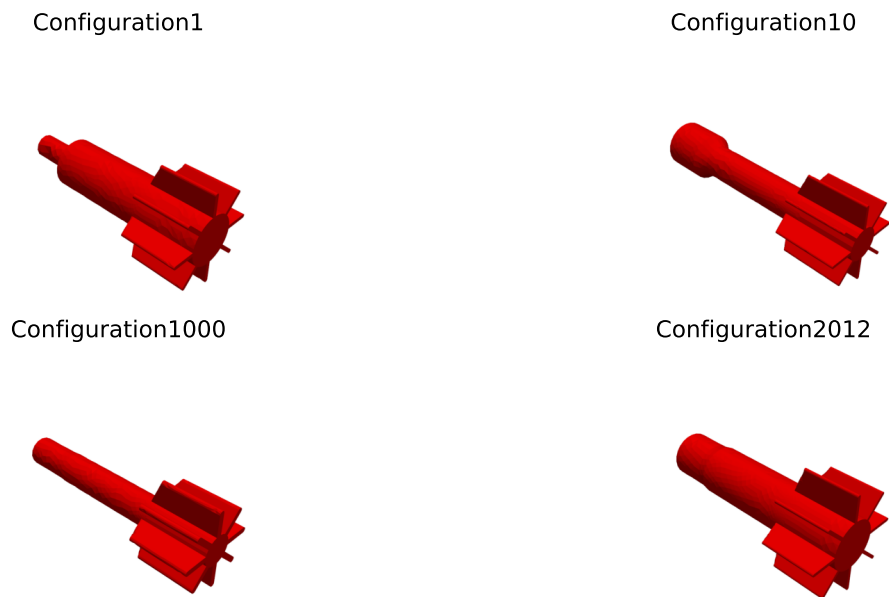


Fig. A.3 Visualization of Finocyl geometry with polynomial ballistic distribution fixed length case, comparison for random samples

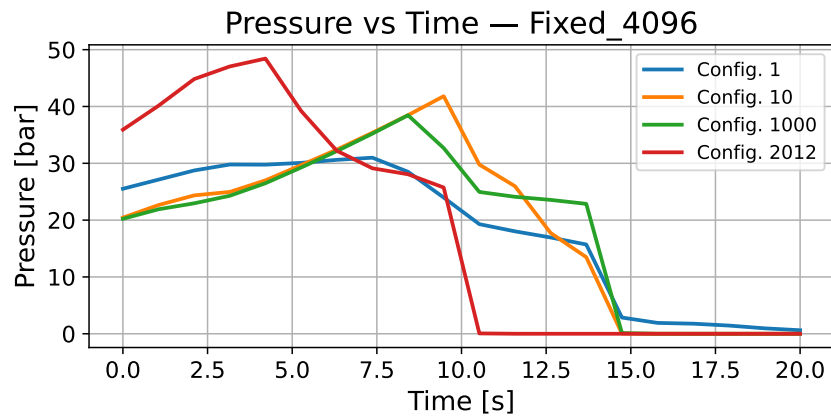


Fig. A.4 Pressure time visualization from dataset for respective Finocyl configuration, fixed length case

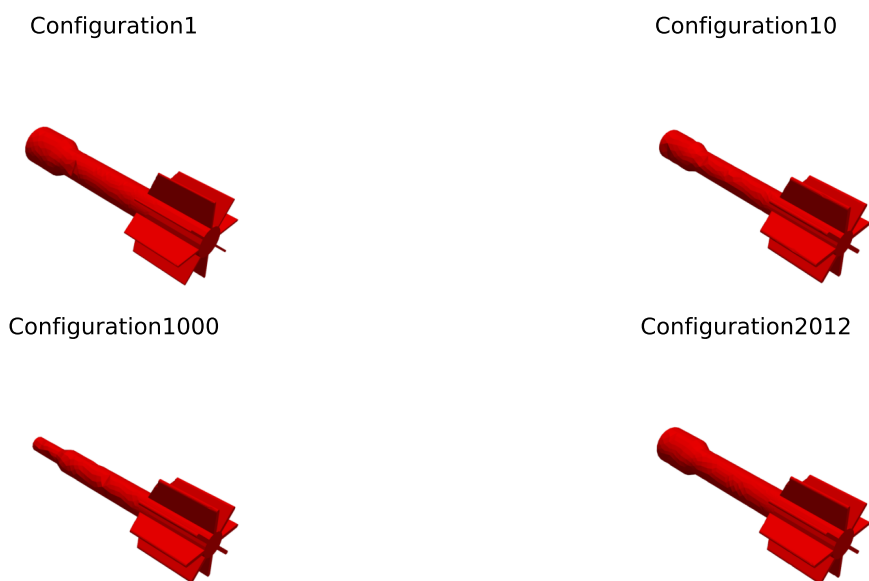


Fig. A.5 Visualization of Finocyl geometry with polynomial ballistic distribution RBF case, comparison for random samples

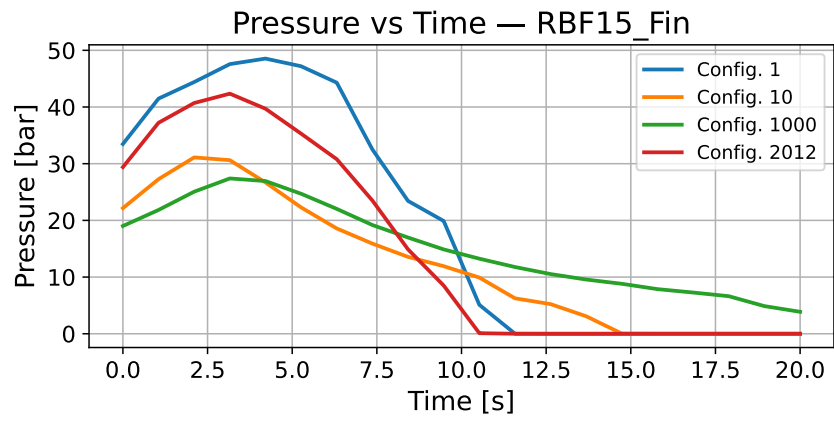


Fig. A.6 Pressure time visualization from dataset for respective Finocyl configuration, RBF case

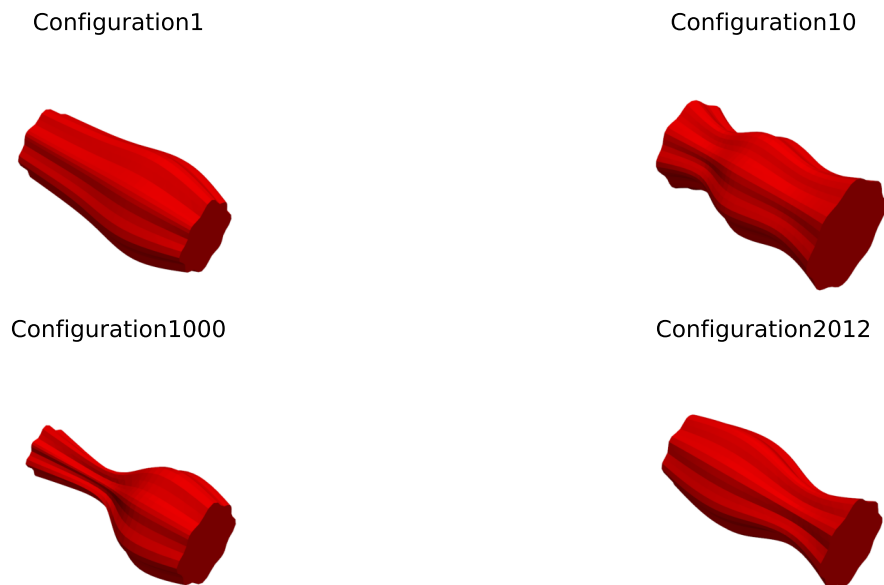


Fig. A.7 Visualization of Harmonic geometry with polynomial ballistic distribution RBF case, comparison for random samples

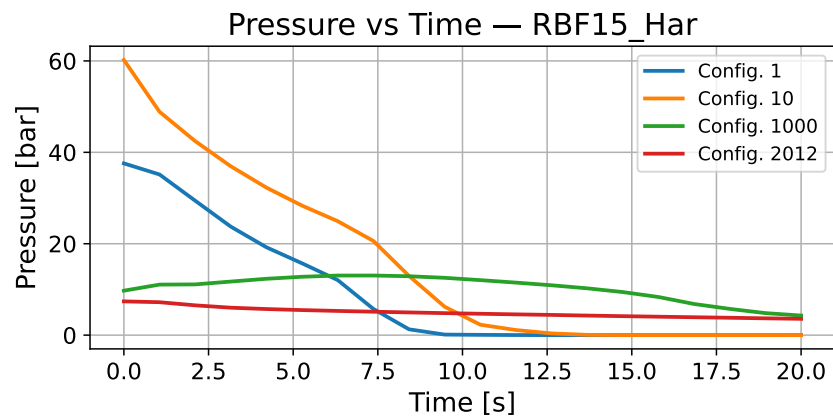


Fig. A.8 Pressure time visualization from dataset for respective Harmonic configuration, RBF case

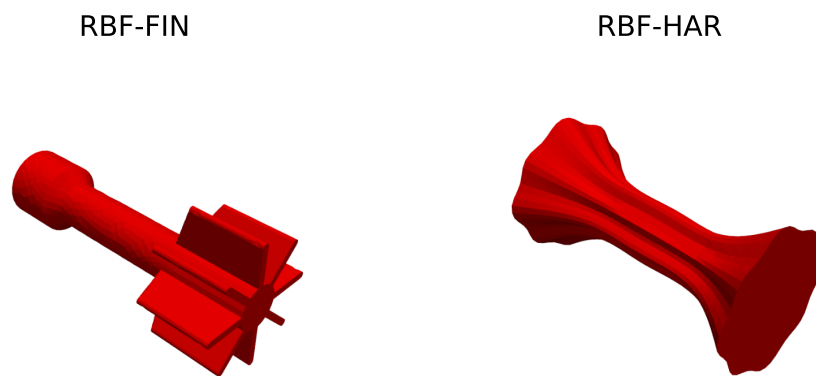


Fig. A.9 Geometries of optima obtained to match Zefiro23 pressure time curve, Fin and Harmonic baseline configurations