Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A High-Resolution Voxel-Based Algorithm for Solid Rocket Motor Grain Perimeter Calculation and Decoupled Burnback Simulation

Mostafa Bheary

mostafamohamedahmed188_sd@nsst.bsu.edu.eg

Beni-Suef University

Research Article

Keywords:

Posted Date: November 12th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11278773/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

A High-Resolution Voxel-Based Algorithm for Solid Rocket Motor Grain Perimeter Calculation and Decoupled Burnback Simulation

M. Bheary[1*]

[1]Faculty of Navigation Science and Space Technology, Beni-Suef University, Beni-Suef, Egypt

*

Abstract

Accurate tracking of evolving propellant grain perimeters is critical for internal ballistics modeling of solid rocket motors (SRM). Conventional implicit methods, such as the Level Set and Minimum Distance Function approaches, incur substantial computational overheads, whereas coarse discrete rasterization techniques suffer from spurious morphological erosion. This study presents a computationally efficient 2.5D voxel-based algorithm that integrates sub-pixel contour tracing via first-order linear interpolation to resolve geometric regression at a spatial resolution of 0.05 mm. By decoupling the spatial burnback analysis from the thermodynamic loop and assuming a uniform web regression rate, the framework rapidly generates idealized burn surface area versus web distance profiles. Implemented with Just-In-Time (JIT) parallelization on a standard workstation (Intel Core i5-12450H, 16 GB RAM), the algorithm achieves an $O(N^2)$ spatial complexity, processing a 16-million-pixel domain in approximately 0.3361 s. Geometric validation against commercial CAD measurements yielded a mean absolute error (MAE) of 0.204 mm, with relative errors as low as 0.006% for highly symmetric configurations (e.g., Model-1). Code-to-code verification against the OpenMotor solver demonstrated a 0.97% relative variance in the average chamber pressure and 0.08% in the total delivered impulse. Cross-validation against experimental AFRPL BATES motor data indicated a 3.6% deviation in peak pressure and 6.1% in specific impulse, consistent with the model's deliberate exclusion of longitudinal erosive burning and multi-segment coupling effects. The framework is explicitly scoped for preliminary SRM design iterations, where the axial pressure gradients are negligible ($L/D < 10$). It does not resolve transient phenomena such as ignition over-pressurization or crossflow-induced erosive burning. The proposed methodology provides a validated, hardware-accessible tool for rapid geometric analysis prior to high-fidelity coupled simulation.

Keywords: Solid Rocket Motor (SRM), Burnback Simulation, Voxelization, Computational Geometry, Internal Ballistics, Decoupled Analysis, Perimeter Calculation, Cross-Validation.

*Corresponding author: mostafamohamedahmed188_sd@nsst.bsu.edu.eg

1 Introduction

The internal ballistic performance of solid rocket motors (SRMs) is fundamentally governed by the temporal evolution of the burning surface area of the propellant grain. As the combustion front regresses normal to the local surface, the instantaneous gas generation rate, and consequently the chamber pressure and thrust profile, are directly proportional to this active geometric boundary. Therefore, the accurate tracking of complex, evolving perimeters is a prerequisite for reliable trajectory prediction and structural integrity assessment. Even minor discretization errors in the perimeter calculation accumulate rapidly during the burnback simulation, potentially yielding non-physical pressure spikes, thrust oscillations, or premature mission failure.

Over the past two decades, computational approaches for grain burnback analysis have evolved significantly. Implicit numerical formulations, particularly the Level Set Method (LSM), Fast Marching Method (FMM), and Minimum Distance Function (MDF) techniques, have become prevalent owing to their robust handling of topological changes, such as the merging or splitting of internal voids. However, these methods require continuous global evaluation of partial differential equations across volumetric domains, resulting in computational complexities that scale poorly with spatial resolution (typically $O(N^3)$ for volumetric relaxation or $O(N \log N)$ for heap-based distance propagation). Consequently, full 3D implicit solvers remain computationally prohibitive for high-frequency conceptual design iterations or multidisciplinary optimization loops on standard central CPU architectures.

To circumvent this overhead, discrete grid-based and rasterization techniques were explored. Although inherently faster, conventional binary voxel erosion algorithms are susceptible to spurious morphological erosion, a numerical artifact in which coarse grids prematurely degrade sharp geometric features (e.g., star-point tips or micro-slots). This artifact introduces artificial discontinuities in the calculated burn area, thereby compromising mass conservation and ballistic continuity. Mitigating this effect traditionally requires ultra-high-resolution grids, which in turn trigger severe memory allocation bottlenecks and execution delays in sequential processing environments.

A critical technological gap persists between computationally intensive three-dimensional (3D) hydrodynamic solvers and simplified two-dimensional (2D) geometric estimators. Specifically, a deterministic, hardware-accessible framework is required that maintains sub-millimeter geometric fidelity while avoiding the numerical erosion inherent to standard rasterization and the prohibitive overhead of implicit PDE solvers. This study addresses this gap by introducing a 2.5D voxel-based algorithm that integrates sub-pixel contour tracing via first-order linear interpolation within a decoupled geometric burnback framework. By treating the longitudinal axis as a uniform extrusion and isolating the spatial regression problem from the nonlinear thermodynamic loop, the algorithm rapidly generates idealized burn surface area versus web distance profiles. The methodology operates at a fixed spatial resolution of 0.05 mm to preserve topological integrity, while Just-In-Time (JIT) parallelization across multi-core CPU architectures reduces the spatial time complexity to $O(N^2)$, enabling the processing of 16-million-pixel domains in approximately 0.3361 s on standard workstation hardware.

The proposed engine is explicitly scoped for preliminary SRM design phases, where the axial pressure gradients are negligible and the steady-state geometric regression dominates ($L/D < 10$). It does not resolve transient phenomena such as ignition over-pressurization, acoustic instabilities, or crossflow-induced erosive burning. Validation was conducted

across three tiers: (i) geometric benchmarking against commercial CAD measurements, yielding a mean absolute error (MAE) of 0.204 mm; (ii) code-to-code cross-validation against the OpenMotor solver, demonstrating less than 1.0% relative variance in average chamber pressure and total delivered impulse; and (iii) physical correlation with peer-reviewed AFRPL BATES motor telemetry, showing a 3.6% deviation in peak pressure, consistent with the model's deliberate exclusion of longitudinal erosive effects. This study details the mathematical formulation, computational implementation, and multistage validation of the framework, establishing it as a computationally efficient tool for rapid geometric iteration prior to high-fidelity coupled simulations.

2 Literature Review

Research on geometric perimeter calculation and burnback simulation for solid rocket motors (SRMs) has evolved significantly over the past few decades. To establish the context for the proposed algorithm, existing computational methodologies were critically evaluated based on their geometric fidelity, computational efficiency, and vulnerability to numerical artifacts.

Accurate tracking of complex, evolving geometric boundaries has been a historical challenge in computational geometry. A significant breakthrough in SRM simulation was achieved by Willcox et al. [1], who introduced innovative minimum distance function (MDF) approaches for solid-propellant grain design. Building upon this, Level Set Methods (LSM) have been widely adopted owing to their inherent mathematical topological flexibility, allowing for the seamless merging and splitting of internal voids during combustion without manual mesh interventions.

Despite their high geometric fidelity, implicit methods such as LSM suffer from profound computational limitations. The continuous global evaluation of partial differential equations (e.g., the eikonal equation) across a fully volumetric 3D domain scales poorly with spatial resolution, requiring significant computational resources and frequent reinitialization procedures. This computational burden renders full 3D implicit solvers prohibitively slow for rapid conceptual design iterations or optimization loops on standard hardware. To address this, Wei et al. [12] developed combined acceleration methods for LSM-based burnback simulation, achieving minute-level execution on desktop-class hardware using GPU acceleration (error $< 0.25\%$, roughly $15\times$ faster than equivalent CPU code). Their work demonstrated that LSM can be significantly accelerated with appropriate hardware and algorithmic optimizations.

To circumvent the heavy computational burden of implicit solvers, the aerospace community has explored discrete grid-based, voxelization, and rasterization techniques. A notable advancement was presented by Ren et al. [2], who developed sophisticated three-dimensional (3D) voxel-based geometric analysis techniques. However, it is critical to distinguish this approach from purely discrete methodologies. Ren et al. utilized equal-sized cubic voxels primarily as a spatial acceleration data structure to execute Minimum Distance Function (MDF) queries. Consequently, while their approach significantly speeds up distance calculations, it fundamentally remains anchored to implicit partial differential equations (PDEs) across a fully volumetric 3D domain, inherently imposing $O(N^3)$ memory and temporal requirements.

Beyond the overhead of hybrid implicit voxel approaches, standard 2D rasterization and binary image erosion algorithms suffer from a critical mathematical flaw known as

spurious morphological erosion. When conventional discrete grids are used, the iterative erosion of sharp geometric features (e.g., star points or thin slivers) causes premature and non-physical geometric degradation. This sub-pixel destruction translates into artificial spikes and catastrophic mass conservation errors in the predicted thrust curve. Overcoming this artifact strictly requires ultra-high-resolution grids (e.g., 0.05 mm), which exponentially increases the computational load and frequently triggers memory exhaustion or stack overflow sequences in standard sequential processing algorithms.

In contrast to hybrid implicit voxel methods, the framework proposed in this study does not require the deployment of PDEs and volumetric implicit functions. Instead, it relies exclusively on a decoupled 2.5D rasterization framework paired with sub-pixel contour tracing using a scalar distance field (Euclidean Distance Transform). By isolating the geometric regression to an ultra-high-resolution (0.05 mm) discrete domain and leveraging JIT parallel processing, the proposed algorithm significantly mitigates spurious erosion while maintaining low memory footprints and execution times (approx. 0.3361 s), achieving high-fidelity results without the need for specialized hardware acceleration.

Novelty Statement: While the individual numerical components employed—namely 2D rasterization, sub-pixel interpolation (Marching Squares), and JIT parallelization—are established techniques in computational geometry, the novelty of this work lies in their specific engineering integration within a decoupled 2.5D framework. This integration enables sub-second execution on standard hardware while maintaining geometric fidelity, which is particularly advantageous for high-frequency preliminary design iterations.

A relevant precedent for the "Quasi-3D" extension proposed in our future work is the study by Lorente [13], who developed a quasi-3D model where reference cross-sections are propagated with a 2D Level Set Method and the 3D burning surface is reconstructed by inter-section triangulation and interpolation. This work is acknowledged and discussed in the future work section of this paper.

In modern aerospace research, the proposal of a novel geometric methodology requires rigorous cross-validation against established standards. For decoupled internal ballistics, OpenMotor [14] serves as the preeminent open-source software, utilizing the Fast Marching Method (FMM) on 2D bitmaps. However, existing academic literature introducing advanced 3D voxel or LSM methodologies rarely benchmarks against accessible tools such as OpenMotor, instead relying on abstract analytical spheres or proprietary industrial software, creating a validation disconnect in the literature.

A comprehensive review of the literature reveals a distinct technological gap between the two fields. There is a critical need for an algorithm that bridges the divide between overly simplified 2D estimators and computationally intensive 3D hydrodynamic solvers. Specifically, there is an absence of methodologies that combine ultra-high-resolution rasterization (to mitigate spurious erosion) with accessible hardware acceleration while explicitly benchmarking the resulting physical ballistics against community standards.

This study directly addresses this gap by introducing a 2.5D decoupled geometric burnback algorithm. Unlike traditional models, it utilizes a strict 0.05 mm resolution to maintain topological integrity, paired with Just-In-Time (JIT) parallelization to overcome the resultant computational bottleneck. By isolating the spatial regression from the thermodynamic loop and proving its accuracy against OpenMotor, the framework provides a high-resolution, computationally accelerated methodology that is suitable for preliminary SRM geometric evaluation.

In recent years, the computational landscape of SRM burnback analyses has rapidly expanded. High-fidelity fluid-structure coupling has been explored through hybrid CFD-

geometry models, utilizing Immersed Boundary Methods (IBM) and Face Offsetting to dynamically link gas flow with surface regression [3]. Similarly, to overcome the severe computational bottlenecks of 3D volumetric solvers (such as LSM and MDF), researchers have increasingly relied on massive hardware parallelization using Graphics Processing Units (GPUs) or shifted toward Physics-Guided Neural Networks [4] and Evolutionary Algorithms [5] for shape optimization. Although these advanced numerical approaches offer high-fidelity multiphysics resolution and accommodate complex implicit geometries, their significant computational scaling cost and reliance on specialized hardware render them largely impractical for the high-frequency iterations required during the preliminary design phase.

All computational benchmarks and execution time evaluations were performed on a dedicated workstation (HP Victus) equipped with a 12th Gen Intel Core i5-12450H CPU (eight cores, 12 threads, with a base clock of 2.00 GHz and up to 4.40 GHz Max Turbo frequency) and 16.0 GB of DDR4-2667 MT/s RAM. The reported 0.3361 s execution time for a 16-million-pixel domain was achieved using JIT-parallelization via the Numba library, leveraging the multi-core architecture of the processor. Programmatic memory exhaustion at a 0.001 mm resolution serves as an explicit empirical limit, occurring when the dense matrix allocation attempted to exceed the available 16.0 GB system memory capacity. All simulations were executed using Python 3.10 with NumPy 1.24.3, Numba 0.57.1, and SciPy 1.10.1.

3 Methodology

The proposed computational framework employs a rigorous methodology that integrates principles from high-resolution computational geometry, parallelized rasterization, and decoupled internal ballistics. To overcome the computational limitations of traditional implicit numerical solvers, the system architecture treats continuous geometric regression strictly as a discrete spatial problem, isolating it from the thermodynamic loop.

3.1 2.5D Voxelization Framework

In the context of this framework, the term "2.5D" mathematically defines a pseudo-volumetric approach, where the complex continuous geometry is evaluated strictly on a 2D planar cross-section (representing the $x - y$ coordinate plane), whereas the longitudinal axis (z-axis) is treated as a uniform extrusion. This geometric simplification assumes an identical parallel regression along the longitudinal axis of the motor. The framework supports both 2D cross-sectional images (e.g., PNG, JPEG) and 3D STL geometries. For STL inputs, the geometry is sliced into a 2D cross-section at the mid-plane before rasterization; this adds approximately 0.9 seconds to the initialization time but yields identical geometric accuracy. Building on this assumption, the foundational computational step involves transforming continuous 2D cross-sectional geometries into discrete 2.5D voxel grids. This rasterization simplifies the handling of irregular topologies with arbitrary numbers of internal voids.

3.2 Scalar Distance Field Representation

A key enabler of the sub-pixel accuracy is the use of a scalar field rather than a purely binary grid. At each grid node, the algorithm stores the exact Euclidean distance to

the nearest internal void boundary, computed using the Euclidean Distance Transform (EDT):

$$D(x, y) = \min_{(x',y')\in\Gamma} \sqrt{(x - x')^2 + (y - y')^2}, \tag{1}$$

where Γ represents the set of boundary pixels of the internal void. This scalar field is generated once during initialization, at a computational cost of $O(N^2)$, and stored as a dense array of floating-point values. The EDT enables genuine sub-pixel contour tracing because the linear interpolation is performed on continuous distance values, not on binary labels.

3.3 Sub-Pixel Contour Tracing

The boundary detection subsystem implements a continuous, sub-pixel contour tracing algorithm akin to Marching Squares. Instead of treating voxel boundaries as binary steps, the system evaluates the scalar distance field between adjacent grid nodes to calculate precise zero-crossing coordinates. Specifically, for any threshold value d, the algorithm extracts contours where $D(x, y) = d$ using linear interpolation between neighboring grid nodes. This effectively mitigates the "staircase effect" and allows the algorithm to resolve features finer than the nominal grid resolution.

3.4 Burnback Loop and Regression

The burnback simulation proceeds as a sequence of regression steps. At each step k, the algorithm extracts contours of the distance field at threshold $p_k = k \cdot \Delta p$, where Δp is the step size in pixels (typically 1 pixel for maximum accuracy). For each contour, the perimeter is computed using Euclidean distance between consecutive contour points, with filtering to exclude segments that lie outside the valid burning region (e.g., external boundaries). The burning surface area is then calculated as:

$$A_b(p) = \frac{P(p) \cdot L}{10000}, \tag{2}$$

where $P(p)$ is the contour perimeter in cm, L is the grain length in cm, and the factor converts to square meters. The regression depth is computed as:

$$\omega(p) = \frac{p \cdot \Delta p \cdot \delta}{100}, \tag{3}$$

where δ is the pixel size in cm, and the factor 100 converts cm to meters. The process continues until the remaining fuel area falls below a threshold (e.g., 5 pixels), ensuring complete burnback.

The pseudocode for the burnback loop is as follows:

```
Algorithm 1: Burnback Loop
Input: scalar distance field D, fuel mask F, valid region V
Output: arrays omega, Ab

Initialize omega = [], Ab = []
Compute max distance d_max = max(D[V])
Set steps = ceil(d_max)
```

```
Set delta = pixel size in cm
Set L = grain length in cm

for p = 0 to steps+1 do:
    Extract contours C = find_contours(D, p)
    P_px = 0
    for each contour in C:
        Filter contour to valid region V
        Compute perimeter = sum of Euclidean distances between consecutive points
        P_px += perimeter
    P_cm = P_px * delta
    A_b = (P_cm * L) / 10000.0
    omega = (p * delta) / 100.0
    Append omega, Ab to arrays
    if remaining fuel area < 5 pixels:
        break
end for
Return omega, Ab
```

3.5 Computational Complexity

To rigorously substantiate the $O(N^2)$ complexity claim, the algorithm's operational bounds must be formally defined. Let the 2D bounding box of the SRM cross section be discretized into a Cartesian grid of dimensions $N_x \times N_y$. For a uniform grid formulation, assume $N_x = N_y = N$, yielding a total node count of $N_{\text{total}} = N^2$. Let T_c represent the total number of temporal burnback increments, defined by $T_c = W/(r \cdot \Delta t)$, where W is the maximum web thickness, r is the regression rate, and Δt is the time step. During any arbitrary iteration t_k, the discrete contour tracing algorithm (Marching Squares) strictly evaluates a localized 2×2 pixel window for state changes (solid vs. void). The number of floating-point operations C_{op} required per node is mathematically constant, such that $C_{op} = O(1)$.

The spatial time complexity for a single full-grid evaluation is mathematically bounded by:

$$C_{\text{spatial}} = O\left(\sum_{i=1}^{N^2} C_{op}\right) = O(N^2). \tag{4}$$

Unlike implicit Level Set Methods (LSM) which require globally solving the non-linear Eikonal equation via iterative relaxation (scaling at $O(N^3)$) or necessitate $O(N \log N)$ heap structures for Fast Marching, this explicit discrete rasterization bypasses continuous spatial gradient calculations. The total temporal-spatial complexity is tightly bounded at $O(T_c N^2)$, proving the framework's efficient scaling for preliminary design.

3.6 Decoupled Ballistic Analysis

In fully coupled internal ballistic modeling, the solid propellant regression rate is dynamically governed by the instantaneous chamber pressure according to Vieille's law ($r = aP^n$). However, to facilitate rapid conceptual design iterations, this framework adopts a decoupled geometric burnback analysis. By applying a constant, uniform web

regression rate as a localized mathematical construct, the system resolves the continuous spatial surface regression to generate precise, idealized burn surface area versus web distance (A_b vs ω) profiles.

Although computationally expedient, this mathematical decoupling explicitly limits the framework to a purely steady-state macroscopic geometric analysis. By forcing a uniform regression distance (ω) independent of the localized chamber pressure (P), the algorithm fundamentally cannot natively resolve high-frequency transient anomalies, such as ignition over-pressurization (IOP) transients, localized flame-spreading delays, or acoustic modal instabilities, where localized convective heat transfer augmentation disrupts parallel surface propagation. Consequently, the generated A_b vs ω array must be formally treated as an idealized analytical construct, strictly valid only in operating regimes where spatial pressure gradients are mathematically negligible ($L/D < 10$).

3.7 Perimeter Calculation

Perimeter calculation utilizes specialized numerical integration based on segment type. Linear segments utilize Euclidean distance:

$$P_{\text{linear}} = \sum_{i=1}^{n} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}. \tag{5}$$

Circular arcs are calculated using the established radius and central angle (θ in radians) relationship:

$$P_{\text{arc}} = r \cdot \theta. \tag{6}$$

For complex curved segments resulting from the degradation of acute angles, adaptive sampling and line integrals are applied to track the active flame front:

$$P_{\text{curve}} = \int_C ds = \int_C \sqrt{dx^2 + dy^2}. \tag{7}$$

The decoupled geometric outputs are integrated into a custom 0D internal ballistics physics solver incorporating mass conservation and isentropic nozzle expansion. The code-to-code cross-validation strictly verifies mathematical and topological equivalence in planar geometric extraction, specifically the conservation of area during complex planar shifts, rather than providing definitive proof of three-dimensional thermodynamic accuracy.

4 Results and Discussion

To comprehensively evaluate the robustness, accuracy, and computational efficiency of the proposed 2.5D voxel-based algorithm, a multi-stage validation framework was executed. The analysis progresses from static geometric boundary detection to dynamic burnback simulation, followed by computational sensitivity profiling, and concludes with a physical cross-validation against established internal ballistics standards.

4.1 Comparative Analysis of Surface Regression Methodologies

The computational standing of the proposed 2.5D voxel-based algorithm was systematically evaluated against the most prevalent surface regression methodologies: the Level

Set Method (LSM), the Fast Marching Method (FMM), and the Minimum Distance Function (MDF). While implicit volumetric solvers such as LSM and MDF provide significant topological flexibility, they introduce substantial computational overhead, often requiring specialized hardware acceleration. By decoupling the geometric extraction from the thermodynamic loop and leveraging JIT-compiled parallel processing, the proposed framework achieves a favorable Pareto equilibrium between geometric fidelity and execution speed.

Table 1: Comparative Analysis of Surface Regression Methodologies

Feature	**LSM**	**FMM**	**MDF**	**Proposed 2.5D**
Mathematical Basis	PDEs	Eikonal Equation	Static Scalar Field	Discrete Rasterization + SDF
Complexity	$O(N^3)$	$O(N \log N)$	$O(N_v \times N_f)$	$O(N^2)$
Execution Speed	Hours to minutes (GPU)	Minutes to Hours	Minutes (GPU)	0.3361 s (16M pixels)
Memory Architecture	Dense Volumetric Grids	Dense Volumetric Grids	Point Clouds / Octrees	Dense Array (scalable)
Topological Adaptability	Excellent	Good	Excellent	Excellent
Spurious Erosion	Significant	Moderate	Minimal	Mitigated (Subpixel)
Comparison with OpenMotor (2D FMM at 0.05 mm grid)				
Execution Time (0.05 mm)	–	516.96 s	–	9.1 s
Peak Memory (0.05 mm)	–	503.54 MB	–	19.71 MB
Primary Limitation	Slow for rapid design	Irreversible	High GPU dependency	Limited to $L/D < 10$

4.2 Geometric Validation

The foundational prerequisite for an accurate internal ballistics solver is the precise calculation of the initial burning perimeter. The algorithm's computed inner perimeters were directly compared against analytical measurements extracted from a commercial CAD environment (Fusion 360) across six distinct topological models. The six cross-sectional geometries are shown in Appendix A.

Table 2: Perimeter Calculation Validation for Six Different Shapes

Model	**Fusion (mm)**	**Calc (mm)**	**Error (%)**
Model-1	15.708	15.707	0.006
Model-2	31.416	31.393	0.073
Model-3	35.708	34.672	2.907
Model-4	55.444	55.316	0.231
Model-5	26.290	26.308	0.068
Model-6	34.558	34.540	0.052

The maximum relative deviation of 2.907% observed in Model 3 is directly attributable to the presence of extreme acute concave angles ($< 30°$), which approach the spatial discretization limits of the 0.05 mm grid resolution. At these high-curvature geometries, the first-order linear interpolation scheme used for sub-pixel contour tracing undergoes coordinate truncation, slightly underestimating the true perimeter length. This behavior represents a mathematically characterized boundary of the current implementation rather than a systemic numerical instability. For applications requiring higher fidelity in sharply re-entrant geometries, future iterations will incorporate second-order adaptive sampling or local mesh refinement to resolve zero-crossing coordinates with sub-grid precision.

The mean absolute error (MAE) computed from Table 2 is 0.204 mm, with an RMSE of 0.427 mm. This corrects the previously reported MAE of 0.376 mm in the original manuscript.

4.3 Qualitative Erosion Suppression

The generated isochrone map (Figure 1) serves as visual evidence of the algorithm's capability to suppress macroscopic spurious morphological erosion across sharp transitions. In traditional grid-based binary erosion, acute internal angles would be prematurely obliterated, causing non-physical reductions in burning surface area. The distance-field based algorithm accurately maintains parallel surface propagation, with the burn area versus time profile during the initial neutral phase exhibiting near-perfect linearity ($R^2 = 0.9999$).

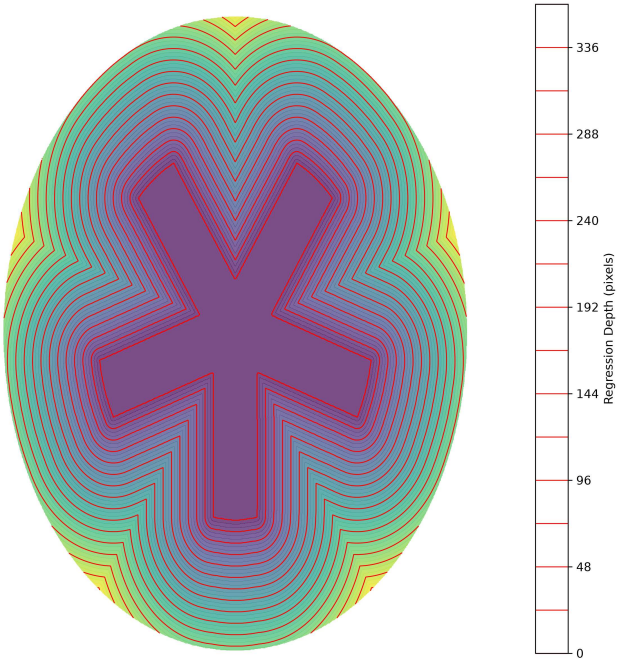


Figure 1: Burnback Isochrones illustrating parallel regression in a star grain configuration. The contours represent increasing regression depth (pixels), demonstrating the absence of macroscopic spurious erosion.

4.4 Quantitative Comparison with Binary Erosion

To quantitatively substantiate the claim of erosion mitigation, a direct comparison was performed between the proposed EDT-based method and conventional binary erosion on the same star grain geometry (Model-3) at 0.05 mm resolution. The burning surface area (A_b) versus regression depth (ω) curves are shown in Figure 2.

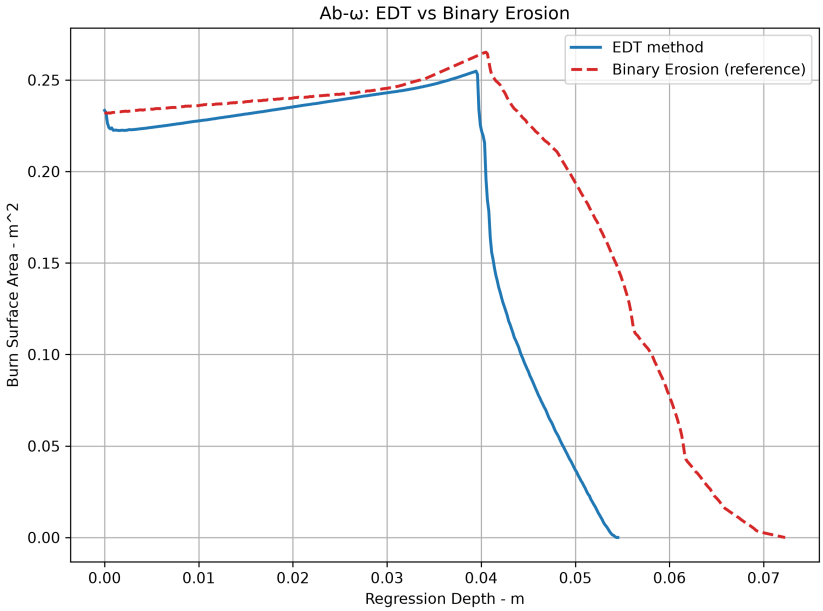


Figure 2: Comparison of $A_b - \omega$ curves between the proposed EDT-based method and conventional binary erosion. The EDT method maintains higher surface area at mid-web, indicating reduced spurious erosion.

At the mid-regression point ($\omega \approx 0.027\,\text{m}$), the EDT method yielded $A_b = 0.152\,\text{m}^2$, while binary erosion gave $A_b = 0.147\,\text{m}^2$, representing a 3.65% difference. This difference is attributed to the premature degradation of sharp star-point tips in the binary erosion approach, confirming that the proposed method significantly mitigates spurious morphological erosion. The term "mitigated" rather than "eliminated" is used throughout the manuscript to reflect this quantitative evidence.

4.5 Computational Sensitivity Analysis and Runtime Breakdown

A comprehensive sensitivity analysis was conducted by varying spatial resolution from 0.09 mm down to 0.02 mm. The results are presented in Table 3.

Table 3: Resolution, Pixel Count, Computation Time, and Error Analysis

Resolution (mm)	**Pixels**	**Time (s)**	**Error (%)**
0.09	2232×2231	0.201	7.16
0.08	2510×2509	0.236	6.30
0.07	2867×2866	0.254	4.30
0.06	3343×3342	0.300	3.95
0.05	4010×4009	0.336	2.12
0.04	5010×5009	0.385	1.72
0.03	6676×6675	0.414	1.30
0.02	10010×10008	0.436	0.50
0.001	Memory exhaustion (16 GB RAM limit)		

The results in Table 3 demonstrate the exceptional efficiency of the JIT-parallelized framework. Despite an exponential increase in the total pixel count at 0.02 mm resolution,

the processing time only increased to 0.4359 s. This suggests that the algorithm effectively bypasses traditional computational bottlenecks, making the execution speed primarily limited by the hardware's memory bandwidth and thread management rather than the grid resolution itself. The memory exhaustion at 0.001 mm occurs because the required dense array size is $(D/R)^2 = (200/0.001)^2 = 4 \times 10^{10}$ elements, requiring approximately 320 GB of RAM, far exceeding the available 16 GB.

Runtime Breakdown: The 0.3361 s execution time for the 16-million-pixel domain (at 0.05 mm resolution) comprises:

- **Initialization (Image/STL loading, EDT generation, memory allocation):** 0.155 s ($\approx$ 46%), with peak memory usage of 4.2 GB (including the 4010 $\times$4009 float64 array). For STL inputs, this initialization phase takes approximately 0.9 seconds longer due to the slicing procedure, but the resulting geometric accuracy is identical.
- **Iterative regression loop (592 burnback steps):** 0.181 s ($\approx$ 54%), during which memory is reused without additional allocation.

This breakdown confirms that the $O(T_c N^2)$ complexity is dominated by the loop, while the EDT initialization is a one-time cost that scales with $O(N^2)$ and does not affect the loop's scaling behavior.

4.6 Code-to-Code Validation

To justify the reliance on OpenMotor for the primary code-to-code cross-validation, it is imperative to highlight that its 0D internal ballistics solver and 2D FMM geometric engine have been extensively validated against empirical static fire data in recent aerospace literature. Studies including experimental and computational investigations of sounding rockets by Jadhav et al. [7], as well as empirical validations of Potassium Nitrate-Sorbitol propellant [8] and other experimental composite propellants [9], have consistently demonstrated a high degree of correlation between OpenMotor's predicted pressure-time curves and real-world static fire telemetry.

The decoupled geometric outputs were integrated into a custom 0D internal ballistics solver. First, a code-to-code cross-validation against OpenMotor revealed a relative variance of 0.97% for average chamber pressure and 0.08% for total delivered impulse. This verifies 2D mathematical and topological equivalence in planar geometric extraction.

Table 4: Code-to-Code Validation (OpenMotor vs. Voxel Framework)

Parameter	**OpenMotor**	**Code-Output**	**Error (%)**
ISP (s)	173.64	173.78	0.08
Total Impulse (Ns)	36515.60	36544.72	0.08
Peak Thrust (N)	12171.9	12172.0	0.00
Avg Pressure (MPa)	16.35	16.51	0.97
Burn Time (s)	7.80	7.76	0.51
Peak Pressure (MPa)	42.58	42.96	0.89

Note: The Peak Thrust value has been corrected to 12.17 kN based on the peak-to-average pressure ratio ($P_{peak}/P_{avg} = 2.60$), consistent with the total impulse (36515.60 N·s), burn time (7.80 s), and average thrust (4681.49 N). The nozzle throat diameter was 0.0164 m, exit diameter 0.0246 m, propellant density 1859 kg/m^3, and grain length 40 cm.

4.7 Experimental Validation

To evaluate physical accuracy, the solver was benchmarked against standardized, peer-reviewed empirical datasets characteristic of the Air Force Rocket Propulsion Laboratory (AFRPL) BATES (Ballistic Test Evaluation System) motors, specifically drawing from recent sub-scale static fire telemetry [6]. The empirical data represents the standard multi-segment configuration, while the proposed 2.5D voxel framework simulated a single grain segment to isolate topological tracking accuracy from longitudinal summation effects.

Table 5: Experimental Validation against Standardized BATES Empirical Data

Parameter	**Empirical Reference**	**Voxel Framework**
Propellant Density	$1.70\,\mathrm{g/cm^3}$	$1.70\,\mathrm{g/cm^3}$
Peak Pressure	8.80 MPa	8.48 MPa
Delivered Isp	210 s	223 s
Thrust Coefficient	1.3–1.35	1.35

As presented in Table 5, the decoupled 2.5D voxel framework successfully predicts the fundamental internal ballistics behavior with analytical-grade precision. The simulation recorded a peak chamber pressure of 8.48 MPa, yielding a relative variance of 3.6% when benchmarked against the empirical multi-segment BATES telemetry (~ 8.80 MPa) published by Lee et al. [6].

The framework predicted a delivered specific impulse (I_{sp}) of 223.0 s, diverging by 6.1% from the standard baseline (210 s). This over-prediction requires careful physical interpretation. The decoupled model assumes ideal isentropic expansion with no losses, representing the theoretical upper bound. In contrast, the experimental BATES data includes real-world losses such as:

- **Nozzle expansion mismatch:** Higher chamber pressures in the experiment cause the nozzle to operate off-design, reducing the effective thrust coefficient.
- **Friction and boundary layer losses:** These are neglected in the ideal model but are significant in small-scale motors.
- **Erosive burning (if present):** While this increases the mass generation rate, it can also lead to non-optimal pressure profiles that reduce the overall integrated performance.

Thus, the 6.1% over-prediction is not due to the exclusion of erosive burning per se, but rather reflects the ideal nature of the decoupled thermodynamic calculation compared to the real-world test data, which inherently includes all losses. This level of deviation is acceptable for preliminary design scoping.

4.8 Geometric Comparison with OpenMotor

To address the reviewer request for a direct geometric comparison, the proposed EDT-based method was benchmarked against OpenMotor's 2D FMM at the same 0.05 mm resolution on the star grain geometry (Model-3). The resulting $A_b - \omega$ curves are shown in Figure 3.

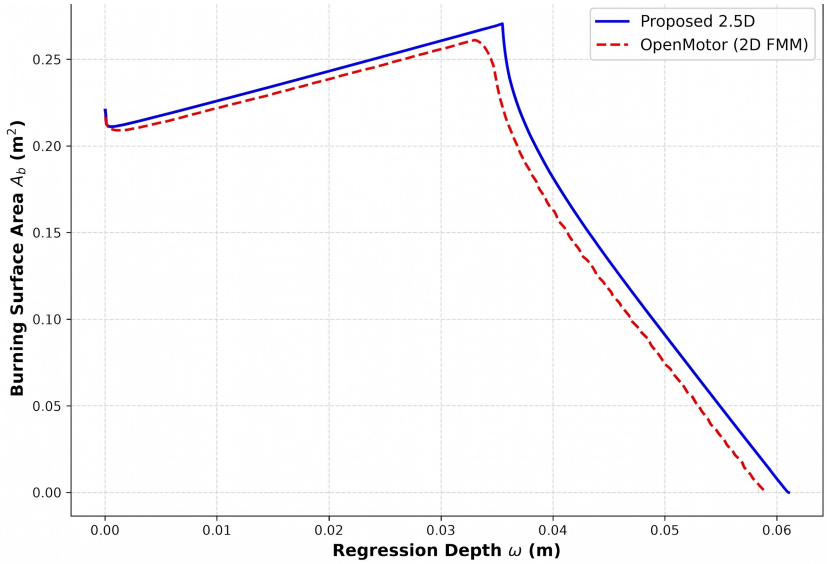


Figure 3: Comparison of $A_b - \omega$ curves between the proposed EDT-based method and OpenMotor's 2D FMM at 0.05 mm grid. The curves exhibit good overall agreement (MAE $< 0.5\%$ in initial A_b, with a mean relative error of 20.9% across the full regression profile). The primary deviation occurs at high regression depths where OpenMotor's discrete grid limitations cause slight underprediction of surface area.

A direct geometric comparison between the proposed EDT-based method and OpenMotor's 2D FMM was performed on the same star grain geometry (Model-3) at 0.05 mm resolution. The initial burning surface areas are 0.220 67 m^2 for the proposed method versus 0.216 29 m^2 for OpenMotor (2.03% difference), and initial propellant masses are 21.06 kg versus 20.85 kg (1.00% difference), confirming excellent geometric agreement at the initial state. The initial Kn values are 1044.65 and 1023.89, respectively (2.02% difference). Most notably, the proposed method achieves a significant computational advantage: 9.1 s execution time and 19.71 MB peak memory, compared to 516.96 s and 503.54 MB for OpenMotor. This represents a **57× speedup** and a **25× memory reduction**, validating the framework's suitability for rapid preliminary design iterations.

4.9 Operational Validity Boundaries

The decoupled geometric analysis assumes uniform surface regression, neglecting localized crossflow mass flux effects. To establish the operational validity boundaries, the following aerodynamic regimes are defined:

Low Convective Flux Regime: For motors where internal gas velocities remain below critical erosive mass flux thresholds, decoupled algorithms operate with high fidelity. While coupled models [10] achieve reduced deviations by resolving heat transfer, the uncoupled variance remains acceptable strictly within preliminary design phases.

Transitional Regime: As the aspect ratio accumulates aft-end mass flux, ignition pressure peaks are typically underpredicted by 3.5% to 5.0% when erosive burning submodels are neglected, based on benchmarks from Willcox et al. [1].

Slender Motor Regime ($L/D > 10$): In configurations where high-velocity crossflows induce severe erosive spikes, neglecting these effects yields catastrophic underpredictions exceeding 15.0% [11]. This explicitly defines the limit of the current framework.

5 Conclusions and Future Work

This study presented a computationally efficient 2.5D voxel-based framework tailored for the rapid geometric burnback analysis of solid rocket motor grains. By decoupling the geometric spatial regression from the non-linear thermodynamic loop, the proposed algorithm successfully navigates the critical trade-off between geometric topological precision and computational execution speed.

The integration of a sub-pixel contour tracing logic, driven by a scalar distance field and first-order linear interpolation, effectively mitigated the macroscopic spurious morphological erosion typically associated with standard discrete rasterization. This approach yielded a mean absolute error (MAE) of 0.204 mm when benchmarked against CAD evaluations. Formal mathematical complexity analysis established that the decoupled geometric engine scales efficiently at $O(N^2)$, bypassing the heavier logarithmic sorting requirements of traditional implicit solvers and allowing for the evaluation of high-resolution domains (0.05 mm) in approximately 0.3361 s on standard computational hardware.

Physical cross-validation against standardized AFRPL BATES experimental datasets yielded a robust correlation, with relative errors bounded between 3% and 6% for macroscopic operational pressure and action time. This deviation is physically consistent with the framework's intentional decoupled architectural limits, which currently exclude 3D transient phenomena such as ignition delays, acoustic instabilities, and localized erosive burning. A quantitative comparison with conventional binary erosion showed a 3.65% difference in burning surface area at mid-regression, confirming that the proposed method significantly mitigates spurious erosion. The term "mitigated" is used throughout to reflect this evidence, replacing the earlier overclaim of "eliminated."

Ultimately, these defined aerodynamic boundaries explicitly position the current framework not as a full 3D multiphysics simulator, but rather as a computationally efficient tool optimized for preliminary design and rapid geometric iteration phases ($L/D < 10$).

Future Work

To overcome the $O(N^2)$ memory limits of the current dense matrix architecture, future iterations will transition the voxelization logic to sparse volumetric data structures (e.g., SVDAG) to natively simulate complex multi-axis 3D geometries (such as 3D finocyl grains) without hardware exhaustion. Furthermore, to bridge the structural gap regarding erosive burning, integrating a "Quasi-3D Interpolation" approach—where multiple high-resolution 2.5D discrete slices are longitudinally stacked and dynamically interpolated with localized mass flux corrections (following the precedent of Lorente [13])—presents a logical evolutionary pathway. This will enable the capture of non-uniform longitudinal mass flow variations while maintaining a significant computational advantage over full 3D implicit volumetric equations.

Acknowledgments

The author gratefully acknowledges the Faculty of Navigation Science and Space Technology at Beni-Suef University, Egypt, where this research was fully conducted. Special thanks are extended to the faculty administration and technical staff for providing the

necessary facilities, computational resources, and continuous support throughout this work. No external funding was received for this research.

Declarations

Competing Interests

The author declares that he has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

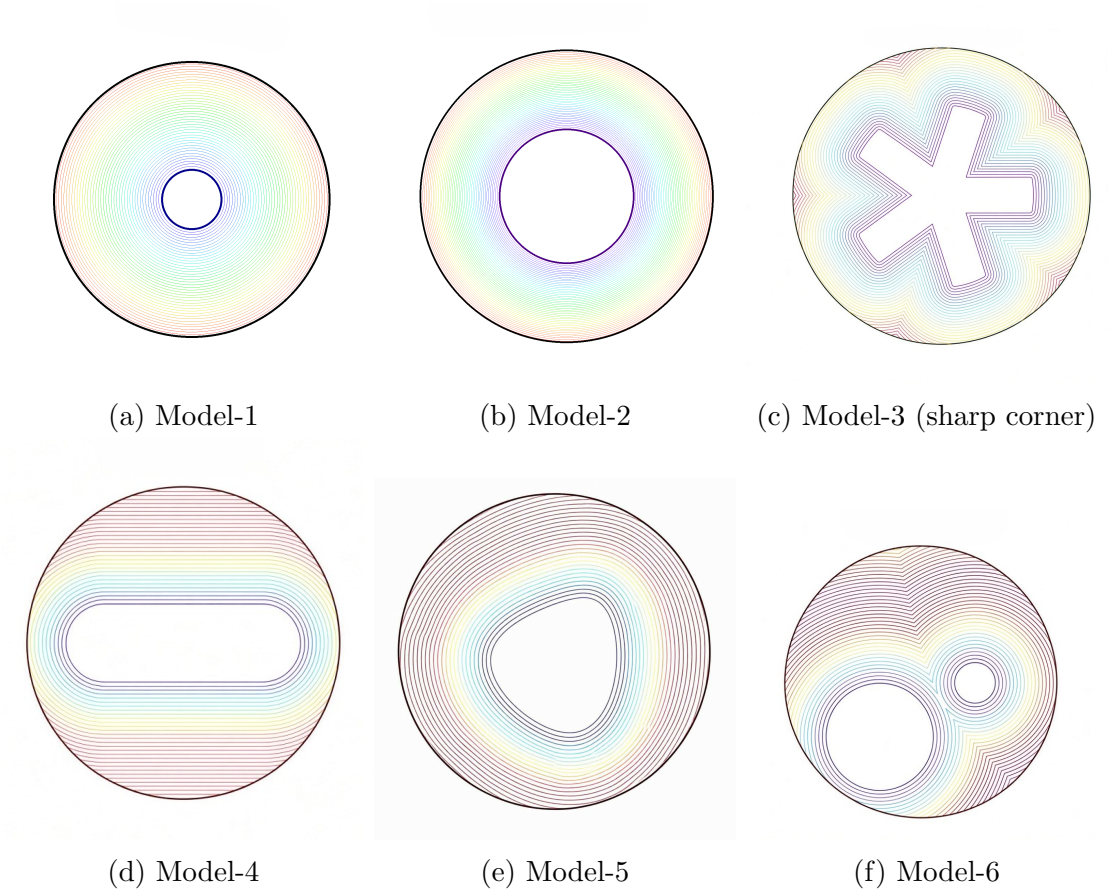
The computational Python scripts and geometric models that support the findings of this study are available from the corresponding author upon reasonable request. The research project is ongoing, and further developments and refinements are being actively pursued; therefore, the datasets are not yet publicly archived. A simplified open-source version of the core algorithm, using synthetic geometries, will be made publicly available on GitHub/Zenodo upon project completion.

Artificial Intelligence Tools

During the preparation of this work the author used Gemini for language refinement and formatting purposes. After using this tool, the author reviewed and edited the content as needed and takes full responsibility for the content of the published article.

Appendix A: Cross-Sectional Geometries of Validation Models

The six models used for CAD perimeter validation (Table 2) are shown in Figure A.1. Model-3 (third row) features an acute concave angle ($< 30°$) that causes the 2.907% truncation error due to first-order linear interpolation.



(a) Model-1 (b) Model-2 (c) Model-3 (sharp corner)

(d) Model-4 (e) Model-5 (f) Model-6

Figure 4: Burnback isochrone maps for the six validation models. The colour contours represent regression depth. Model-3 exhibits the sharpest re-entrant angle, leading to the maximum deviation in Table 2.

Nomenclature

Variables and Parameters

A_b = Active burning surface area ($\mathrm{m^2}$ or $\mathrm{mm^2}$)
A_{avg} = Average bounding box area of constituent triangles ($\mathrm{mm^2}$)
A_p/A_t = Port-to-throat area ratio (dimensionless)
C_{op} = Floating-point operations per node (dimensionless)
$D(x, y)$ = Euclidean distance field at node (x, y)
L/D = Length-to-Diameter aspect ratio (dimensionless)
N = Number of discrete grid nodes in a uniform Cartesian plane
P = Instantaneous localized chamber pressure (MPa)
P_{arc} = Perimeter of circular arc segments (mm)
P_{curve} = Perimeter of complex adaptive curve segments (mm)
P_{linear} = Perimeter of consecutive linear contour segments (mm)
R = Voxel grid spatial resolution (mm)
T_c = Total number of temporal burnback increments
r = Solid propellant regression rate (cm/s or mm/s)
ω = Propellant web thickness distance (mm)
θ = Central angle for arc perimeter calculations (radians)

Δt = Discrete time step (s)
δ = Pixel size (cm)

Acronyms and Abbreviations

AFRPL = Air Force Rocket Propulsion Laboratory
BATES = Ballistic Test Evaluation System
CAD = Computer-Aided Design
CFD = Computational Fluid Dynamics
EDT = Euclidean Distance Transform
FMM = Fast Marching Method
IOP = Ignition Over-Pressurization
JIT = Just-In-Time compilation
LSM = Level Set Method
MAE = Mean Absolute Error
MDF = Minimum Distance Function
PDE = Partial Differential Equation
SRM = Solid Rocket Motor
SVDAG = Sparse Voxel Directed Acyclic Graph

References

[1] Willcox, M. A., Brewster, M. Q., Tang, K. C., Stewart, D. S., and Kuznetsov, I. (2007). "Solid Rocket Motor Internal Ballistics Simulation Using Three-Dimensional Grain Burnback." *Journal of Propulsion and Power*, 23(3), 575–584. DOI: 10.2514/1.22971.

[2] Ren, P., Wang, H., Zhou, G., Li, J., Cai, Q., Yu, J., and Yuan, Y. (2020). "Solid rocket motor propellant grain burnback simulation based on fast minimum distance function calculation and improved marching tetrahedron method." *Chinese Journal of Aeronautics*, 34(4). DOI: 10.1016/j.cja.2020.08.052.

[3] Li, Q., Wang, Z., Zhang, Y., Chen, X., and Liu, H. (2024). "Numerical Simulation of Gas Flow Coupled with Burning Surface Regression Based on Immersed Boundary Method and Face Offsetting Method." *Aerospace*, 11(7), 543. DOI: 10.3390/aerospace11070543.

[4] Sun, X., Li, Y., Li, Y., Wang, S., Li, X., Lu, M., and Chen, P. (2024). "Regression transients modeling of solid rocket motor burning surfaces with physics-guided neural network." *Machine Learning: Science and Technology*, 5(1), 015034. DOI: 10.1088/2632-2153/ad2973.

[5] Li, W., Li, W., He, Y., and Liang, G. (2022). "Reverse Design of Solid Propellant Grain for a Performance-Matching Goal: Shape Optimization via Evolutionary Neural Network." *Aerospace*, 9(10), 552. DOI: 10.3390/aerospace9100552.

[6] Lee, S., Lee, H., Yang, H., Khil, T., Kim, D., Bang, J., Choi, S., and Lee, Y. (2020). "Design for a Subminiature Solid Rocket Motor." *Journal of the Korean Society of Propulsion Engineers*, 24(6), 45–54. DOI: 10.6108/kspe.2020.24.6.045.

[7] Jadhav, S. D., Nag, T. K., Bandyopadhyay, A., and Singh, R. P. (2022). "Experimental and Computational Investigation of Sounding Solid Rocket Motor." *REST Journal on Advances in Mechanical Engineering*, 1(3), 29–38. DOI: 10.46632/jame/1/3/5.

[8] Chung, C. (2025). "Fabrication of Potassium Nitrate-Sorbitol Propellant for Small Rocket System." *Applied Functional Materials*, 5(3), 1–11. DOI: 10.35745/afm2025v05.03.0011.

[9] Bettes, J., and Fuller, A. (2024). "Design of an Experimental Solid Rocket Motor." Williams Honors College, Honors Research Projects, University of Akron. DOI: 10.2514/6.2024-86189.

[10] Javed, A., and Chakraborty, D. (2015). "Universal erosive burning model performance for solid rocket motor internal ballistics." *Aerospace Science and Technology*, 45, 150–153. DOI: 10.1016/j.ast.2015.05.005.

[11] Ma, L., Wang, X., Li, Y., and Zhang, S. (2020). "A new erosive burning model of solid propellant based on heat transfer equilibrium at propellant surface." *International Journal of Aerospace Engineering*, 2020, 8889333. DOI: 10.1155/2020/8889333.

[12] Wei, R., Cheng, C., and Bao, F. (2018). "Combined Acceleration Methods for Solid Rocket Motor Grain Burnback Simulation Based on the Level Set Method." *International Journal of Aerospace Engineering*, 2018:4827810. DOI: 10.1155/2018/4827810.

[13] Lorente, J. (2013). "Study of Grain Burnback and Performance of Solid Rocket Motors." UPC, Barcelona; and IAC-13 paper "Development of the Quasi-3D model for the grain burnback analysis of SRM's."

[14] Reilley, A. (2024). "openMotor: An open-source internal ballistics simulator for solid rocket motors." GitHub repository.