ECCOMAS Congress 2016
VII European Congress on Computational Methods in Applied Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (eds.)
Crete Island, Greece, 5–10 June 2016

# HIGH-ORDER TRACKING METHOD FOR THE SIMULATION OF BURNING FRONTS IN 3D

**Denis Gueyffier and Bastien Andrieu**

ONERA – The French Aerospace Lab
Palaiseau, France
e-mail: {denis.gueyffier, bastien.andrieu}@onera.fr

**Keywords:** front-tracking method, Eikonal flow, spectral method.

**Abstract.** *We present a novel set of techniques designed for very accurately tracking a propagating front of arbitrarily complex 3*D *geometry. The method relies on a segmentation into smooth patches, allowing for the use of spectral methods while avoiding Gibbs phenomenon. High-order accuracy is thus obtained using few degrees of freedom. We show how this method robustly and accurately tackles difficult geometrical problems arising when a surface evolves under an Eikonal flow, e.g. formation of singularities and large variation in time of the surface area. In particular, interactions between patches are handled by using a topological representation of the front and taking appropriate actions in order to preserve a physically valid front.*

# 1 INTRODUCTION

Simulation of fluid flow in presence of moving interfaces or boundaries is a difficult task, yet of critical importance in domains such as solid rocket propulsion. A solid rocket motor is basically composed of a casing in which a solid propellant – the grain – is inserted. Once ignited, the solid propellant converts to high-temperature, high-pressure gases that are expelled through a nozzle, producing thrust. As the grain burns radially outward, the volume of the chamber increases in time. On the other hand, the propellant's burning velocity is linked to local properties of the internal flow inside the combustion chamber. Accurately predicting the time evolution of the burning grain surface is thus of utmost importance, as the propellant's geometry is the only means to control rocket thrust.

This paper presents recent advances in the development of a high-order method for tracking this burning surface.

# 2 BURNING FRONT PROBLEM

When burning, the propellant regresses following the local normal direction to the grain surface. The burning velocity is given by empirical laws based upon the propellant composition and local variables of the internal flow such as static pressure or velocity.

Several approaches exist when dealing with the representation of a dynamic interface, and can be divided into two main categories. The so-called front-*capturing* methods rely on an implicit description of the front (e.g. level-set methods [4]). While they are well suited to address interfaces of complex topology, very fine grids are generally needed to obtain decent accuracy. On the other hand, front-*tracking* methods are based on an explicit representation and thus need to deform a mesh. These explicit methods offer a potentially more accurate description of the front yet are in general more complex to implement.

The method presented in this paper falls into the second category. The time-dependent grain surface is assumed to be represented parametrically as follows

$$\begin{aligned} \gamma : \mathbb{R}^2 \times \mathbb{R}^+ &\to \mathbb{R}^3 \\ (u, v, t) &\mapsto \mathbf{x} = (x, y, z) \end{aligned} \tag{1}$$

The position vector $\mathbf{x}$ is then solution to the following partial differential equation (PDE)

$$\frac{\partial \mathbf{x}}{\partial t} = V(\mathbf{x})\mathbf{n}(\mathbf{x}) \, , \tag{2}$$

$V$ being the grain burning velocity and $\mathbf{n}$ the unit outward normal vector, obtained using the cross-product of two tangent vectors.

$$\mathbf{n} = \frac{\mathbf{x}_u \times \mathbf{x}_v}{\|\mathbf{x}_u \times \mathbf{x}_v\|} \, , \tag{3}$$

where subscript $u$ (resp. $v$) denotes differentiation with respect to the $u$ parameter (resp. $v$).

The front can be represented implicitly by a level-set function $T$ such that surfaces $\mathbf{x} = \gamma(t)$ are solution of equation

$$T(\mathbf{x}) = t \, . \tag{4}$$

Following [4], equation 4 can be restated as a form of the well-known Eikonal equation, which in our case reads

$$\|\nabla T\|^2 = \frac{1}{V^2} \, . \tag{5}$$

The Eikonal equation is relevant to many front-propagation problems. This equation is a hyperbolic PDE whose solutions often exhibit singularities, also called *caustics*. Dealing with Eikonal flows is particularly difficult as the propagating front undergoes dramatic distortions.

In [1], a novel method has been presented for the simulation of the internal flow inside the chamber of a solid rocket booster, coupled with the computation of the burning grain regression in a 2D configuration. This approach makes use of differentiation in Fourier space to track the burning front very accurately, while handling the singularities (caustics) which develop naturally in time. A dynamic surface mesh of the front is then constructed. A volume mesh of the chamber is deformed to follow the propagation of the front. The method is also designed to preserve the connectivity of both the surface and volume meshes, as most CFD codes do not handle either addition or deletion of mesh elements during the calculation. It is finally demonstrated that the method yields excellent accuracy even with very few degrees of freedom.

In the following sections, we present an extended version of this novel approach, designed to address more complex, fully three-dimensional geometries. This paper focuses essentially on the tracking of the burning front. The improvement of volume mesh deformation techniques will be addressed in future work.

## 3 HIGH-ORDER FRONT-TRACKING METHOD

### 3.1 Motivation

Most grain geometries used in solid rocket boosters are not globally smooth, therefore the direct extension of the previous method yields poor results, as truncated Fourier series of non-smooth or non-periodic functions are polluted by the well-known Gibbs phenomenon (fig. 1 left). A solution to this problem has been proposed by Bruno et al. [2]. Their idea is to decompose a piecewise smooth surface into patches. A smooth, periodic extension of each patch is then computed, yielding a high-order local parameterization for each one of them. This strategy allows to circumvent the Gibbs phenomenon and therefore to recover the good properties intrinsic to spectral methods such as exponential convergence with the number of degrees of freedom.

An outline of our method is given in the following sections, including the dynamic implementation of Bruno's technique in our front-propagation algorithm.

### 3.2 Segmentation into patches

The first stage consists in extracting smooth, quadrilateral patches from the piecewise smooth initial grain surface. Such decomposition can easily be derived from a CAD file, as solids are already defined as unions of smooth faces (Bézier surfaces, NURBS, ...) in most standard CAD file formats. Smooth patches can also be extracted from a surface mesh, given an appropriate feature-detection algorithm [5].

### 3.3 Front-propagation algorithm

On each patch, computation of normal vectors and other high-order geometric quantities is performed via differentiation in Fourier space. Unlike in [1], Fourier coefficients are not obtained with a Fast Fourier Transform (FFT) algorithm but result from the "Fourier continuation" method presented in [2] and briefly described in the paragraph below.

### *Fourier continuation*

The principle of this technique is the following: let $f$ be a smooth, non-periodic function $f : [0,1] \to \mathbb{R}$, sampled at $n$ data points. We seek a trigonometric polynomial $\hat{f}$, truncated to $M \leq n$ modes and with period $T > 1$ that matches $f$ on $[0,1]$ in the least square sense.

$$\hat{f}(x_j) = \sum_{k=1}^{M} a_m e^{\frac{i2\pi}{T} m_k x_j} \approx f(x_j) \qquad (1 \leq j \leq n)\,, \tag{6}$$

with $m_k = k - (\lfloor M/2 \rfloor + 1)$, $(1 \leq k \leq M)$. This is equivalent to solving a least-square minimization problem.

$$\min_{\mathbf{a}} \|A\mathbf{a} - \mathbf{f}\|^2\;, \tag{7}$$

with $\mathbf{a} = (a_1, \ldots, a_M)^T$, $\mathbf{f} = (f(x_1), \ldots, f(x_n))^T$ and $A_{j,k} = e^{\frac{i2\pi}{T} m_k x_j}$.
Solving problem 7 yields the $M$ first coefficients of a trigonometric polynomial which fits $f$ on $[0,1]$ up to machine precision (see figure 1) and is a periodic continuation of $f$ on a larger domain $[0,T]$. This polynomial can be used to interpolate $f$ between data points or even to approximate its derivatives. The $p$-th derivative of $\hat{f}$ reads

$$\hat{f}^{(p)}(x) = \sum_{k=1}^{M} \left(\frac{i2\pi}{T} m_k\right)^p a_m e^{\frac{i2\pi}{T} m_k x} \approx f^{(p)}(x)\,. \tag{8}$$

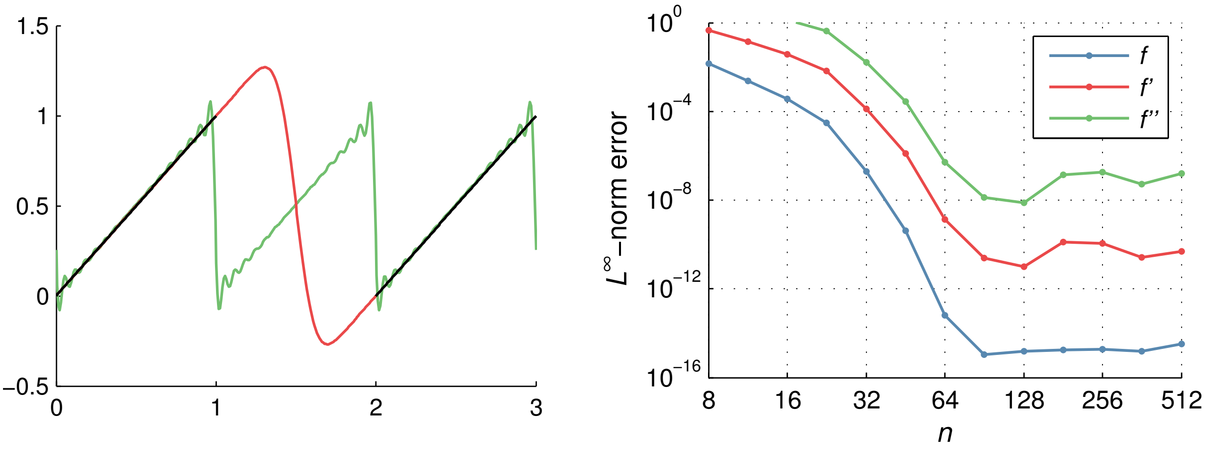


Figure 1: Left: the truncated Fourier series (green) of a non-periodic function (e.g. $f(x) = x$, in black) is polluted by oscillations. However, the periodic extension (red) computed by the Fourier continuation method matches $f$ on its definition domain. Right: maximal approximation error on $f$ and its first two derivatives for various degrees of freedom $n$.

This technique can easily be extended to higher dimensions, and is used to construct a high-order parameterization of each patch $\Phi : (u,v) \mapsto \mathbf{x} = (x,y,z)$, where $\Phi$ is a trigonometric polynomial.

### Propagation of the burning front

Time integration of equation 2 is performed using simple forward Euler time stepping, the normal direction being given by the normalized cross products of the two tangent vectors $\mathbf{x}_u$ and $\mathbf{x}_v$.

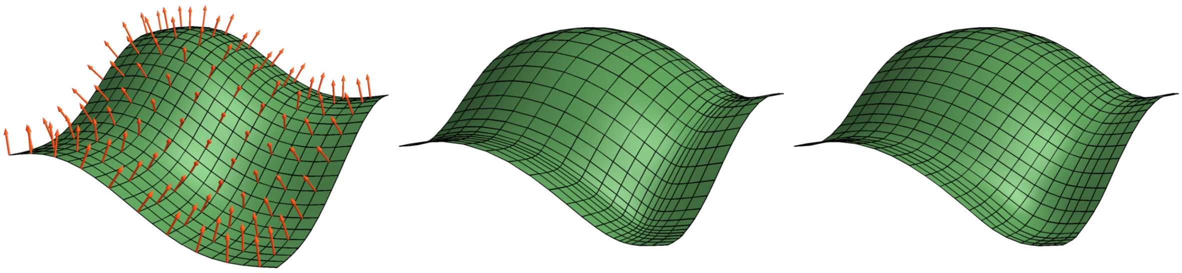


Figure 2: Left: initial surface patch. Center: same patch after an offset along its local normal directions, the surface mesh is distorted. Right: same patch after a few steps of regularization.

### *Surface mesh regularization*

As the surface is transported along its local normal direction, vertices are drawn apart in convex regions, whereas they are clustered together in concave areas, leading to large distortions of the surface mesh. These distortions can be the source of aliasing errors and numerical instability. To prevent this, we add a tangential motion to the vertices. The shape of the surface is then preserved, provided that the boundaries are properly constrained. This tangential motion is governed by an ellipitic operator analogous to a manifold-projected Laplacian that prevents clustering of vertices and leads to a better distribution of vertices along the surface (fig. 2).

In practice, several small steps of tangential regularization are performed for each step of propagation along the normal direction.

### *Topological representation*

Each patch is treated separately by the front propagation algorithm. However, when reconstructing a single mesh for the entire surface, one must handle interactions between patches (stitching, removal of invalid intersections, . . . ). Toward this end, the patches can be connected into a graph-like structure (*layout*) which stores only topological information. This *layout* is constituted of two types of elements, *nodes* and *segments*:

- a *node* corresponds either to a vertex shared by several patches or to a patch corner, and is linked to the local parametric coordinates $(u, v)$ of the vertex in each of its indicent patches ;
- a *segment* is a pair of connected *nodes*, and can be seen as a portion of a boundary between two patches.

These elements are quite similar to those used for topological representation in several CAD file formats. An example of such a layout is given in fig. 3 for a simple geometry.

## 3.4 Singularities

The initial geometry might contain slope discontinuities such as sharp ridges or creases. These geometric singularities are necessarily located at patch boundaries. When patches on both sides of such a singularity are propagated individually, they either leave a gap or intersect. In both situations, the surface mesh will become invalid if no special action is taken. In the following two paragraphs, we describe how we handle singularities, thanks to the layout introduced previously.

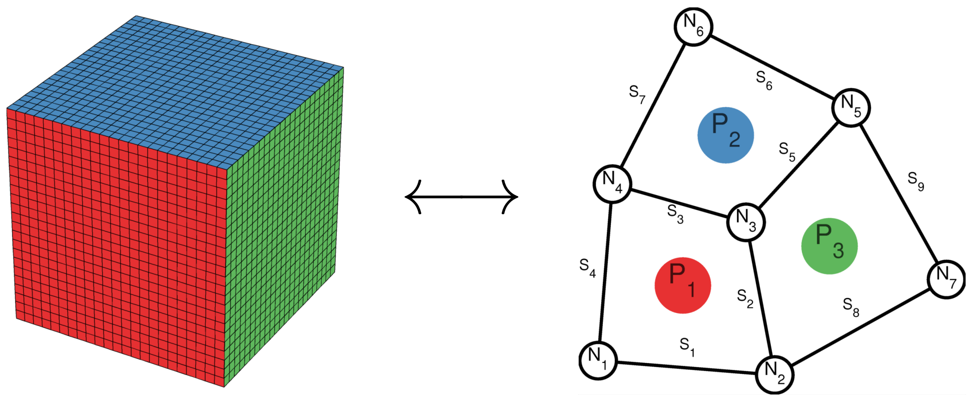


Figure 3: Exemple of a piecewise smooth surface (three faces of a cube, left) and its corresponding layout (right). Each patch is represented in a different color.

### *Ridges*

Patches incident to a ridge separate when propagating as vertices along the ridge follow multiple directions (fig. 4). The non-physical gap thus formed must be filled. Such regions are easily identified thanks to the topological layout. New patches can then be generated from propagating ridges and inserted between the patches that are no longer adjacent. The geometrical construction of these new patches follows the entropy-satisfying Huygens' principle, which is also at the base of the level-set methods [4]. Once constructed, the new patches are treated like the usual ones using the front-propagation algorithm.

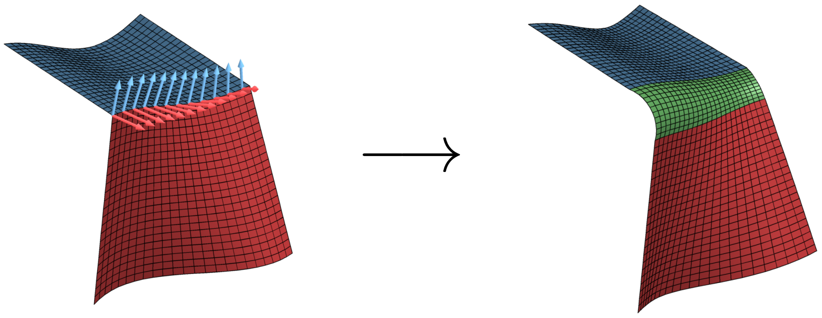


Figure 4: The vertices along a ridge have different velocities in the two incident patches, forming a gap when they propagate. A new patch (green) is created to fill the gap, respecting Huygens' principle.

### *Creases*

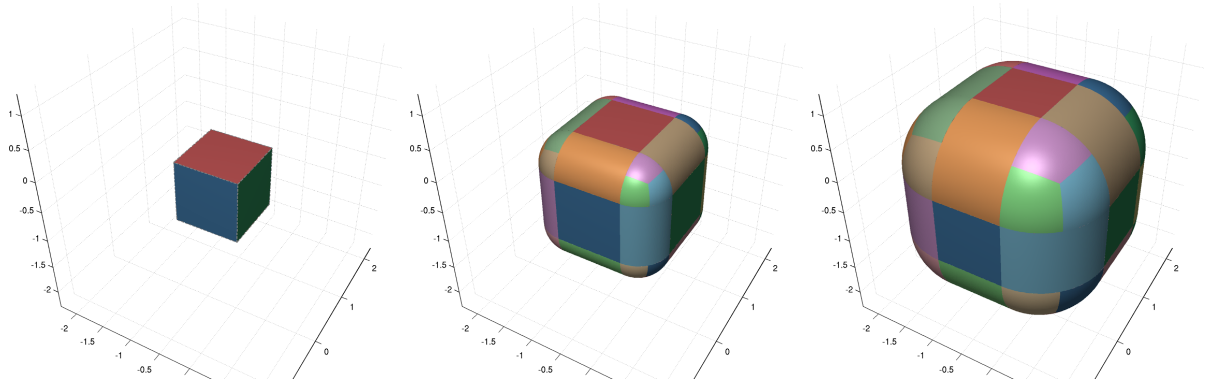
Two patches incident to the same crease intersect when being propagated, forming caustics. Only the exterior portion of each patch remains valid. Intersection curves are therefore computed in order to delineate invalid parts in each patch, which are then trimmed off.

Despite the effort in regularizing the surface mesh, additional caustics are bound to develop in finite time in smooth concave regions due to the very nature of the Eikonal flow. Patches in which this occurs are no longer smooth and thus become prone to the Gibbs phenomenon. If no action is taken, oscillations develop at the vicinity of the caustic and quickly propagate to the whole patch, leading to numerical instability. However, formation of such caustics can be anticipated, as curvature grows indefinitely where new creases are being formed. Thus, we can handle this issue by splitting patches on the verge of becoming singular, yielding two new smooth patches that are incident to a new crease. This splitting procedure is triggered when local maximal curvature exceeds a threshold value (some fraction of the normal displacement in one time step).

### 3.5 Preservation of the mesh connectivity

The method described in the previous section is thus able to propagate a piecewise smooth front by treating smooth patches separately and handling interactions between them in order to preserve the topology of the whole surface. However, a single mesh of the front is not yet available. On top of that, the connectivity of the union of all the patches is not preserved, mostly due to addition of new patches. However, this can be solved by constructing a global parameterization of the front based upon the layout described at the end of section 3.3. This structure can indeed be used to gather each patch's local parameterization into a set of charts, forming a global mapping from parameter space to the surface.

Once a global parameterization is available, a unique connectivity can be mapped to the front at each time step. The result is a dynamic surface mesh that keeps the same connectivity over time.



*offset distance* $= 0$ *offset distance* $= 0.5$ *offset distance* $= 1$

Figure 5: Three stages of the expansion of a cube computed by our method. Each patch is represented in a different color.

## 4 NUMERICAL RESULTS

Experiments have been run on simple geometries for which analytic results are known. The test exposed in the present section considers the expansion of a unit cube under uniform speed. The convergence of our method is studied using different levels of refinement of the surface mesh: a coarse one and a finer one (with twice as many vertices).

As the cube expands, its surface can be seen as the level-set of a distance function from the initial cube. Mean error on distance is computed in both configuration for an offset distance ranging from 0 to 1 (fig. 6 left).

The area of each patch can also be computed by integrating the area element $d\mathcal{A}$ given by the first fundamental form.

$$d\mathcal{A} = \sqrt{EG - F^2} dudv \, , \tag{9}$$

where $E = \mathbf{x}_u \cdot \mathbf{x}_u$, $F = \mathbf{x}_u \cdot \mathbf{x}_v$, and $G = \mathbf{x}_v \cdot \mathbf{x}_v$ are computed by spectral differentiation, as described in section 3.3. Total area of the expanding cube is then compared to an analytic solution.

$$\mathcal{A}(d) = 8 + 6\pi d + 4\pi d^2 \, , \tag{10}$$

where $d$ denotes the offset distance from the initial unit cube. (Each edge of the cube propagates as a quarter of a cylinder, and each vertex as an eighth of a sphere, yielding formula 10).

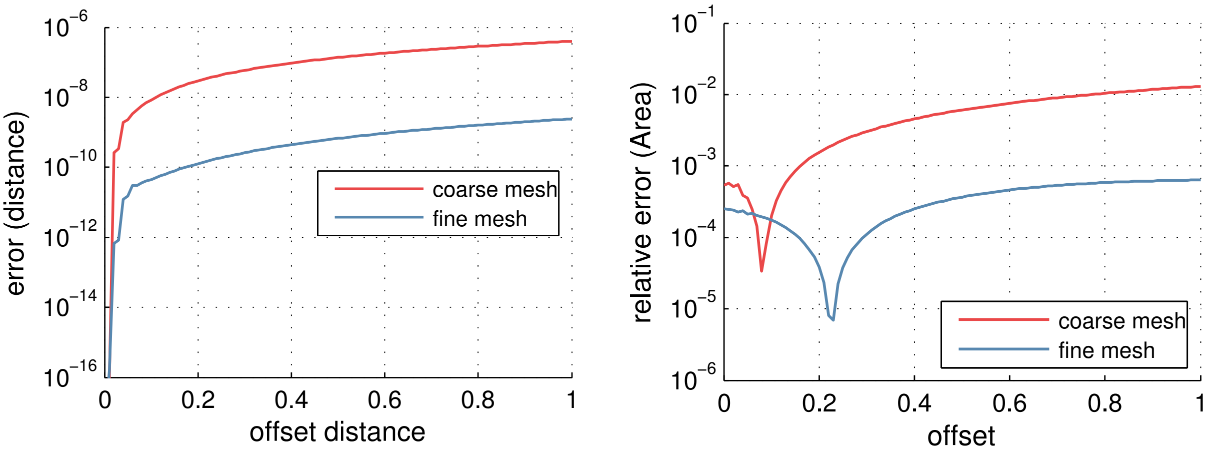


Figure 6: Left: error on distance from the initial cube. Right: relative error on total area.

## 5 CONCLUSION

We have presented a novel spectrally accurate tracking method for burning fronts. Numerical experiments have demonstrated the validity of the method on simple test cases and further tests of increasing complexity and realism will be performed in future studies. Eventually, this algorithm will be integrated into deformable volume mesh methodologies and coupled to a Navier-Stokes solver for the simulation of the internal fluid flow in a solid rocket motor with burning propellant grain.

## REFERENCES

[1] D. Gueyffier, F. X. Roux, Y. Fabignon, G. Chaineray, N. Lupoglazoff, F. Vuillot, J. Hijlkema, and F. Alauzet, Accurate Computation of Grain Burning Coupled with Flow Simulation in Rocket Chamber, *Journal of Propulsion and Power*, **31:6**, 1761–1776, 2015.

[2] O. P. Bruno, Y. Han, and M. M. Pohlman, Accurate, high-order representation of complex three-dimensional surfaces via Fourier continuation analysis, *Journal of Computational Physics*, **227**, 1094–1125, 2007.

[3] S. O. Unverdi, and G. Tryggvason, A Front Tracking Method for Viscous Incompressible Flows, *Journal of Computational Physics*, **100**, 25–37, 1992.

[4] J. A. Sethian, Level Set Methods and Fast Marching Methods, *Cambridge University Press*, 1999.

[5] A. Myles, N. Pietroni, D. Zorin, Robust Field-aligned Global Parametrization, *ACM Transactions on Graphics*, **33:4**, 135:1–135:14, 2014.