26th International Meshing Roundtable, IMR26, 18-21 September 2017, Barcelona, Spain

High-fidelity, dynamic CAD model for propagating surfaces and moving meshes

Bastien Andrieu\textsuperscript{a},*, Denis Gueyffier\textsuperscript{a}

\textsuperscript{a}ONERA, 29 avenue de la Division Leclerc, 92322 Châtillon, France

Abstract

We present a set of methods designed for tracking with high-order accuracy the propagation of surfaces in three dimensions with arbitrary geometric and topological complexity. Targeted applications are engineering and scientific problems for which a CAD (computer-aided design) model of the geometry is available, i.e. surface patches with high-order parameterizations as well as topological connectivity. Our method uses and maintains such a representation throughout the propagation. Validity of the dynamic boundary-representation model is ensured by handling geometric singularities. Examples are presented to demonstrate the accuracy of the pseudo-spectral method used for tracking each surface patch. A strategy for deriving a dynamic surface mesh is finally proposed for applications such as Finite Element/Volume computations with body-fitted volume grids in domains with deforming boundaries.

\textcopyright 2017 The Authors. Published by Elsevier Ltd.

Peer-review under responsibility of the scientific committee of the 26th International Meshing Roundtable.

Keywords: Surface propagation; B-Rep; Evolving Geometry; Surface mesh; Deforming mesh

1. Introduction

This paper presents a set of novel numerical methods for very accurately tracking fronts and for dynamically following deforming surface meshes resulting from complex coupled three-dimensional problems. The method relies on a boundary-representation model of the initial front, which can be generated by any computer-aided design (CAD) system, and maintains this representation – high-order surface parameterizations and topological connectivity – throughout the entire deformation. Reconstruction and tracking of the deforming surface are highly accurate for piecewise smooth geometries, as well as computationally efficient. Such precision has not been attained before with any front tracking/capturing method when addressing non-smooth surfaces. This method is particularly well suited for very precisely tracking surface deformations in the presence of geometric singularities (ridges, caustics, cusps, etc), which are either existing in the CAD model representing the initial surface or during the deformation process.

Practical problems that this approach tackles originate from complex, highly coupled physics. Such problems can involve deformation of the computational domain, usually discretized into volume meshes. Deformations of the computational domain may result from multi-physics problems such as combustion, fluid-structure interactions,
electromagnetic wave diffraction, multi-material simulations, icing, dendrite growth, to list a few. The work presented in this paper is based on an approach initially developed for canonical surface propagation problems, and extends it to address more general, complex applications.

**Prior work.** Techniques for dealing with propagating surfaces are generally divided into two broad categories. Eulerian methods, which capture the surface using an implicit representation (e.g. level-set [1] or volume-of-fluid [2] methods) are robust regarding changes in topology. These methods have been widely and successfully used for surfaces deforming under curvature flows (involving surface tension), but still require very fine grids to achieve decent accuracy in flows where geometric singularities develop. On the other hand, Lagrangian methods track explicitly the surface by deforming a mesh [3], potentially providing greater accuracy. However, these methods also require specific treatment in presence of singularities such as self-intersections.

Most Lagrangian tracking methods use a piecewise linear approximation of the surface [4]. In this work, we propose a high-order method based on spectral discretization in space. This approach relies on expanding a parameterization of the surface into a basis of global, smooth functions. Such a method – originally designed for surfaces in Willmore flows – has been used to simulate deforming and interacting vesicles in creeping flows with spectral accuracy [5]. Spherical harmonics as basis functions are suitable for surfaces with spherical topology [5] whereas trigonometric polynomials are ideal for periodic parameterizations [6]. The method described in this paper is suited for arbitrary geometric flows. We thus choose Chebyshev polynomials as they allow us to parameterize surfaces of arbitrary topology with greater flexibility. Our approach is also inspired by prior work on the *Fourier Continuation* method [7], which has been used to derive high-order local parameterizations of complex, yet non-deforming surfaces.

The remainder of this paper is organized as follows. Section 2 describes an algorithm for propagating smooth parametric surfaces. Extension of this method to more general surfaces is discussed in Section 3. Section 4 presents a technique for constructing a dynamic mesh of the propagating surface. Finally, examples are given to illustrate our method.

### 2. Pseudo-spectral method for the Lagrangian tracking of a parametric surface

In the present section we state some key properties of Chebyshev polynomials before outlining an algorithm for tracking propagating surfaces. Further details on Chebyshev polynomials and their use in spectral methods can be found in references [8,9].

#### 2.1. Chebyshev polynomials

Chebyshev polynomials are a family of orthogonal polynomials that are widely used in the field of numerical analysis, due to their excellent properties in the approximation of other functions.

**Definition.** The *n*-th Chebyshev polynomial (of the first kind) $T_n$ is a polynomial of degree $n$ defined by

$$T_n (\cos \theta) = \cos n\theta.$$

For $x \in [-1, 1]$, $T_0(x) = 1$, $T_1(x) = x$ and from the trigonometric identity $\cos n\theta + \cos(n-2)\theta = 2 \cos \theta \cos(n-1)\theta$ the following recurrence relation can be derived

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), \quad n \geq 2.$$  \hspace{1cm} (1)

Relation (1) allows efficient evaluation of Chebyshev polynomials, e.g. via Clenshaw’s algorithm [10].

On the interval $[-1, 1]$, $T_n$ reaches its extremal values at the $n + 1$ *Chebyshev-Gauss-Lobatto* (CGL) points

$$\hat{x}_k = \cos \frac{k\pi}{n}, \quad k = 0, \ldots, n.$$
Chebyshev expansion. A function \( f : [-1, 1] \to \mathbb{R} \) can be approximated by its truncated Chebyshev series
\[
    f_N(x) = \sum_{n=0}^{N} c_n T_n(x) \approx f(x),
\]
where Chebyshev expansion coefficients \( \{c_n\} \) can be computed given the values of \( f \) at CGL points [8]
\[
    c_n = \frac{2}{N} \sum_{k=0}^{N} f(\hat{x}_k) T_n(\hat{x}_k) = \frac{2}{N} \sum_{k=0}^{N} f(\hat{x}_k) \cos \frac{nk\pi}{N}.
\]
Besides, we have
\[
    f_N(\hat{x}_k) = \sum_{n=0}^{N} c_n \cos \frac{nk\pi}{N}.
\]
Equations (3) and (4) are Discrete Cosine Transforms (DCT-I). Both direct and inverse transforms can be performed by means of a Fast Cosine Transform (FCT) algorithm, thus providing a very efficient method for the construction and evaluation of Chebyshev expansions on the Chebyshev-Gauss-Lobatto grid.

Differentiation. The first derivative of \( f \) can also be approximated by its truncated Chebyshev series
\[
    f'_N(x) = \sum_{n=0}^{N} c^{(1)}_n T'_n(x) = \sum_{n=0}^{N} c_n T'_n(x) \approx f'(x),
\]
which coefficients \( \{c^{(1)}_n\} \) can be obtained via another recurrence relation [11]
\[
    c^{(1)}_n = \begin{cases} 
        0 & \text{if } n \geq N \\
        c^{(1)}_{n+2} + K_n(n+1)c_{n+1} & \text{if } 0 \leq n < N
    \end{cases}
\]
with \( K_N = 1 \) and \( K_n = 2 \) for \( n < N \).

Chebyshev expansion coefficients of higher-order derivatives of \( f \) are computed by using relation (5) recursively. As noticed by Wengle and Seinfeld [12], this recursive algorithm may suffer from ill-conditioning, i.e. errors in the smallest coefficients \( c^{(1)}_n \) are magnified such that the accuracy of even the largest coefficients is decreased. This problem can be solved by setting to zero all the coefficients smaller than a given threshold, chosen in this work to be about an order of magnitude above machine precision.

Fig. 1: Error in approximating the analytic function \( f : x \mapsto e^{3 \sin x} \) and its first two derivatives by their truncated Chebyshev expansions. The black, dotted curve shows the exponential rate of convergence.

Fig. 2: Exponential decay of the Chebyshev coefficients of the analytic function \( f : x \mapsto e^{3 \sin x} \) and its first two derivatives.

1 the double prime in the sum indicates that both first and last terms are halved.
Spectral accuracy. If \( f \) has \( m \) continuous derivatives then truncation error \( |f_N(x) - f(x)| \) decreases like \( N^{1-m} \) for all \( x \in [-1, 1] \) as \( N \to \infty \). In other words, the order of convergence depends on the smoothness of \( f \) [13]. In particular, if \( f \in C^\infty([-1, 1]) \), \( f_N \) converges uniformly towards \( f \) faster than any power of \( N \) for \( N \) sufficiently large.

Fig. 1 shows the exponential decay of approximation error for the smooth yet oscillatory function \( f(x) = e^{3 \sin 3x} \) and its first two derivatives (for which equation (5) is used to derive the Chebyshev expansion coefficients). Additionally, the accuracy of the spectral approximation can easily be estimated by checking the absolute values of Chebyshev coefficients, which decrease exponentially fast as well for infinitely differentiable functions, see Fig. 2.

2.2. Lagrangian tracking of a smooth parametric surface

Surface propagation problem. Let \( \Sigma \) be a smooth, orientable surface. The time evolution of a point \( x \in \Sigma \) is given by

\[
\frac{\partial x(t)}{\partial t} = V(x(t), t) N(x(t)),
\]

(6)

where \( N(x) \) is the direction normal to \( \Sigma \) at \( x \). For now, we assume that \( \Sigma \) remains regular, i.e. that \( N \) is defined for all \( x \in \Sigma \) and its orientation is consistent (we will address the case where \( \Sigma \) may become singular in Section 3). Velocity \( V \) is dictated by the phenomenon that governs surface propagation and may depend on geometric properties of \( \Sigma \).

Pseudo-spectral tracking method. To solve this surface propagation problem we devise the following pseudo-spectral method. We assume \( \Sigma \) can be parameterized by a tensor-product bivariate Chebyshev series

\[
\sigma(u, v) = \sum_{m=0}^{M} \sum_{n=0}^{N} e_{m,n} T_m(u) T_n(v),
\]

with \(-1 \leq u, v \leq 1\). This assumption holds only for open, 0-genus surfaces with one boundary. However we will see in Section 3 how our method extends to surfaces of arbitrary topology.

We introduce the set of collocation points

\[
\hat{x}_{m,n} = \sigma(\hat{u}_m, \hat{v}_n), \quad m = 0, \ldots, M, \quad n = 0, \ldots, N,
\]

where points \( \{(\hat{u}_m, \hat{v}_n)\} \) define a Chebyshev-Gauss-Lobatto tensor-product grid in the parametric domain of \( \Sigma \)

\[
\hat{u}_m = \cos \frac{m\pi}{M}, \quad \hat{v}_n = \cos \frac{n\pi}{N}.
\]

2 each Cartesian component of the position vector \( \sigma(u, v) \) is actually expanded as one real-valued series.
Coefficients \( \{c_{m,n}\} \) can thus be derived from \( \{\hat{x}_{m,n}\} \) via a (two-dimensional) Discrete Cosine Transform, and vice versa (as described in Section 2.1).

We solve equation (6) by tracking the points \( \{\hat{x}_{m,n}\} \) in a Lagrangian way. Using the parametric definition for \( \Sigma \), the unit normal vector is obtained by taking the normalized cross product of two tangent vectors

\[
N = \frac{\sigma_u \times \sigma_v}{\|\sigma_u \times \sigma_v\|},
\]

where \( \sigma_u = \frac{\partial \sigma}{\partial u} \) and \( \sigma_v = \frac{\partial \sigma}{\partial v} \) (see Fig. 3).

We use explicit time stepping for integration of equation (6). At each time step,

- Chebyshev expansion coefficients \( \{c_{m,n}\} \) for the position vector \( \sigma \) are computed via the (direct) Fast Cosine Transform of the position of collocation points \( \{\hat{x}_{m,n}\} \);
- Chebyshev expansion coefficients for the spatial derivatives \( \sigma_u \) and \( \sigma_v \) are then computed using relation (5);
- Tangent vectors are evaluated at collocation points via an inverse FCT of those coefficients;
- Unit normal vectors are finally computed using relation (7).

This spatial discretization yields spectral accuracy, i.e. a very good approximation of \( \Sigma \) can be obtained using few degrees of freedom, provided it is sufficiently smooth. However, in the presence of sharp features or geometric singularities, spectral approximations converge poorly due to Gibbs oscillations [7]. In practice, geometries involved in surface propagation problems are quite complex and not globally smooth. Furthermore, even surfaces that are initially smooth may develop naturally geometric singularities as they propagate (e.g. in Eikonal flows). In the following section we establish a framework to address this issue.

3. Tracking of complex, piecewise-smooth surfaces

The work presented in this paper focuses on surface propagation problems in industrial or scientific applications for which a CAD model of the geometry is available. The method presented in the following section takes this particular definition as a starting point and extends it to evolving geometries.

3.1. Boundary Representation

In most CAD systems, an object is defined by a Boundary Representation (B-rep) [14]. The boundary of a solid is composed of a collection of surface patches. These patches are delimited and connected by edges, which are portions of curves lying on adjacent patches. Edges are themselves delimited and connected by vertices.

Therefore, a B-rep model contains two types of information. First, topological information gives adjacency relationships and links between the entities (vertices, edges and patches). Second, geometric definitions of these entities are also provided (points, curves and surfaces).

In the following subsections, we give details on how to extend our pseudo-spectral tracking method to more complex surfaces, defined as B-rep models of solids.

3.2. Propagation of a set of surface patches

Each patch of a B-rep model is geometrically defined by a smooth parametric surface, typically over a (possibly trimmed) square parametric domain, and thus meets the requirements of the tracking algorithm presented in Section 2.2. Therefore an accurate approximation of the evolving surface is obtained by propagating each patch independently. This approximation is free of Gibbs oscillations because all the geometric singularities present in the initial surface are located at patch boundaries (edges of the B-rep model).

The algorithm for propagating multi-patch surfaces is thus a straightforward extension of the single-patch version. However, topological connections still need to be updated in order to define a valid B-rep model of the modified geometry.
3.3. Regeneration of a valid Boundary Representation

In order to reconstruct a valid B-rep model, geometric singularities must be dealt with. When two patches adjacent to a sharp edge propagate, they either separate (in convex regions) or intersect (in concave regions). We handle both situations with specific treatment, according to the phenomenon governing surface propagation. For wavefront-like propagation, Huygens’ principle states that each point on the wavefront behaves as a source of secondary waves that propagate in all directions, so the new wavefront is the envelope of these secondary waves. In this work, we focus on applications where the front represents the boundary of a solid, so the so-called entropy condition stating that “once a particle is burnt it stays burnt” [1] must be satisfied. This condition basically disallows self-intersections of the front.

Convex singularities. Patches incident to a sharp edge in a convex region of the surface separate and leave a gap as they propagate individually because points along the edge follow multiple directions. This gap must be filled in order to ensure a valid B-rep model. New patches are then created from these singular edges and inserted in the B-rep model with appropriate topological links to form a valid surface. These new patches are then propagated in the same way as other ones. The following technique is used to construct the new patches.

Each point along a propagating edge possesses two velocity vectors with same magnitude but possibly different directions $N_{\text{left}}$ and $N_{\text{right}}$ (cf. equation 6). A convex, sharp edge propagate into a new patch which requires a parameterization. The $u$-direction of such a new patch is chosen to be the direction tangential to the edge (Fig. 4(a)). Therefore, the new patch is initially degenerate in the $v$-direction, i.e. iso-$u$ curves are reduced to single points. However, the propagation direction varies along the $v$-direction from $N_{\text{right}}$ to $N_{\text{left}}$, so, as soon as it propagates, this new patch is no longer degenerate (Fig. 4(b)). A similar technique is used to generate new patches from vertices incident to several sharp edges, which possess a distinct velocity vector for each incident patch.

Concave singularities. In concave regions of the surface, adjacent patches interfere when propagating. Intersection curves must then be computed and their traces in the parametric domain of each patch are used to delineate and trim off invalid regions (Fig. 5). These curves provide a geometric definition for the new edges of the B-rep model, as well as the topological information required to connect the patches into a single surface.

The polynomials used in our method to parameterize surface patches having potentially high degrees, this step reveals challenging both in terms of robustness and efficiency, which are crucial as intersections are to be computed at each step of propagation.

Subdivision methods that reduce the problem to the intersection of piecewise linear approximations [15] become computationally expensive when high accuracy is required and may fail to capture the correct topology of intersection curves in presence of small features. Marching methods [16] are efficient at generating sequences of connected points.
Fig. 5: (a) Intersecting patches adjacent to a sharp feature. (b) Trace of the intersection curves in the parametric domain of the yellow, cylindrical patch. (c) 3D view of the trimmed patch and its incident B-rep edges and vertices.

along branches of the intersection curve but may suffer problems such as jumping from one branch to another when marching with large steps.

In this work, we use a method combining adaptive subdivision and marching. The loop detection criterion presented in [17] is used to find all the connected components of the intersection curve, and provides a marching algorithm safe from jumping components. Interrogation of tight oriented bounding volumes in the subdivision phase allows to discard pairs of subdivided surfaces that do not intersect. This way, intersection curves are computed efficiently and with high accuracy.

The presented method allows to propagate a surface with high-order accuracy. It takes as a starting point a collection of connected surface patches (analogous to the boundary representation of a solid) and maintains this representation while ensuring valid topological links as the geometry evolves.

Nevertheless, most engineering applications require a surface mesh, typically triangular, piecewise linear. In the following section we describe a method for deriving a dynamic mesh which conforms to the propagating surface.

4. Derivation of a dynamic surface mesh

The propagating surface generally represents the boundary of the computational domain for simulations using Finite Element or Finite Volume methods. Body-fitted volume meshes with an Arbitrary Lagrangian Eulerian formalism are often used in situations where the geometry evolves. Computational codes used in such situations usually do not support either addition/deletion of mesh elements or changes in mesh connectivity during the calculation.

In this work, we thus consider mesh smoothing techniques, which are based solely on vertex relocation and therefore ensure preservation of the mesh connectivity. Two different strategies are considered.

4.1. Local submeshes based on patch segmentation

A first approach consists in dividing the surface mesh into submeshes, each one of them conforming to a single patch. This makes both mesh generation and mesh smoothing easier by taking advantage of the parametric definition of surface patches. Two-dimensional techniques can be used, working on the \((u, v)\) coordinates of vertices in the 2D parametric domain. Vertices of the mesh are then guaranteed to remain exactly on the surface when mapped to the 3D Euclidean space. Parameterizations of general surfaces cannot be isometric and induce metric distortions. All operations, while being performed in the parametric domain, must therefore be driven by the resulting geometry in Euclidean space.

An initial mesh is generated using this indirect approach. Each initial submesh being generated separately, the discretization of their boundaries must be computed first in order to ensure conformity between adjacent submeshes. The trimming contour of a patch is defined by a closed sequence of B-rep edges. A discretization of these edges is thus constructed by distributing points uniformly along their supporting curves in 3D space. The number of points is
chosen in order to conform to a user-specified mesh size. Parametric coordinates of these points are also generated and form the trace of the curves in each adjacent patch’s parametric domain. The interior of the submeshes is then generated taking into account the intrinsic metric of the parameterizations [18].

At each step of the surface propagation, the mesh is updated following three stages.

Discretization of B-rep edges. First, the discretization of the B-rep edges must be regenerated, as for initial mesh generation. However, to enforce mesh connectivity preservation, the number of mesh vertices must remain constant over time for each B-rep edge.

Conformation to the new parametric domain. As the surface propagates, the parametric domain of a given patch evolves as well. The 2D submesh associated with this patch must then conform to the moving boundaries of its parametric domain. The parametric coordinates of boundary vertices are determined by the discretization of edges previously regenerated. The interior vertices are then relocated in order to fit inside the new parametric domain.

For this step, a Delaunay Graph method [19] is used. At time step $k$, the constrained Delaunay triangulation $T^k$ of the boundary points $\{b^k_j\}$ of the submesh is constructed (Fig. 6(b)). Each interior point $p^k_i$ of the submesh is assigned the triangle $t^k_i$ of $T^k$ in which it lies, as well as its barycentric coordinates $(\lambda^k_j)_{j=1,\ldots,3}$ in this triangle, so that

$$p^k_i = \sum_{j=1}^3 \lambda^k_j b^k_{v^k_j},$$

where $\{v^k_j, j = 1, \ldots, 3\}$ are the indices of the vertices of $t^k_i$.

The position of this point at time step $k+1$ is then obtained by interpolating the new positions of the vertices of $t^k_i$, using the same barycentric coordinates (Fig. 6(c))

$$p^{k+1}_i = \sum_{j=1}^3 \lambda^k_j b^{k+1}_{v^k_j}.$$ 

This step ensures the submesh conforms to the new boundary of the parametric domain and prevents inversion of elements. However, bad quality elements may still be produced because neither the actual geometry in 3D space nor mesh connectivity are taken into account. An additional smoothing step is thus performed in order to optimize mesh quality.

Mesh smoothing. Numerous techniques have been developed for smoothing 2D meshes, and can be extended to surface meshes lying on parametric patches. Laplacian smoothing, which consists in relocating each interior vertex at a weighted average of its neighbors, is the most commonly used method. It is straightforward and efficient but may cause distorted or even inverted elements near a concave boundary. Besides, it tends to produce isotropic meshes in 2D parameter space, which, mapped to the surface in 3D space become anisotropic due to metric distortions intrinsic to the parameterization.

Physics-based smoothing methods are essentially based on spring analogy. In this approach, the mesh is viewed as a system of springs connecting vertices [20]. This approach remains simple and computationally efficient but may fail to avoid inverted elements. Optimization-based methods are the most effective ones, but are more expensive and difficult to implement. This approach consists in relocating vertices in order to maximize a given quality metric. In [21], the minimum angle in the triangulation is chosen as the objective function, whereas the method presented in [22] seeks to minimize a distortion energy measuring the deviation of an actual element from an ideal one. Several other quality metrics can be found in reference [23]. Optimization-based methods have also been extended successfully to meshes supported by parametric surfaces [24]. However, to our knowledge, there has been no application to dynamic CAD surfaces.

In this work, an approach combining linear spring-based and optimization-based smoothing has been adopted. Spring analogy is used first, then an optimization step is performed to improve only elements with quality below a given threshold. This allows to keep a moderate computational cost while ensuring a good quality mesh.
A modified version of the “segment spring method” presented in [20] is used. It consists in applying Hooke’s law to vertex displacements, and seeking an equilibrium state. The force applied on vertex $i$ by the $n_i$ springs connected to it reads

$$ F_i = \sum_{j=1}^{n_i} \alpha_{i,j} (\delta_j - \delta_i), $$

where $\delta_i$ is the displacement of vertex $i$ in parameter space, and $\alpha_{i,j}$ is the stiffness of the spring connecting vertices $i$ and $j$, defined by

$$ \alpha_{i,j} = \frac{\phi_i + \phi_j}{2\ell_{i,j}}, $$

(8)

where $\ell_{i,j}$ is the Euclidean distance between vertices $i$ and $j$, and $\phi_i$ is an approximation of the inverse geodesic distance from vertex $i$ to the nearest boundary of the submesh. This additional factor allows to stiffen up edges close to moving boundaries, preventing excessive mesh distortions in these regions. All quantities involved in equation (8) are measured in 3D space so the equilibrium state corresponds to a regular mesh in physical space.

Then, an optimization procedure similar to the one presented in [25] is performed. The objective function to maximize is the triangle quality metric

$$ Q = \psi \frac{4 \sqrt{3} A}{\sum_{i=1}^{3} \ell_i^2}, $$

(9)

where $A$ is the area of the 3D-space triangle and $\ell_i$ the lengths of its edges. $\psi$ is the sign of the area of the (2D) parameter-space triangle, therefore this quality metric discriminates inverted elements. Unlike in [25], here the quality metric is optimized with respect to the parametric coordinates $(u, v)$.

**Limitations.** While facilitating mesh generation and smoothing by conforming to the segmentation into patches, this approach is subject to severe limitations. First, so-called “badly meshable regions” (e.g. taper or high aspect ratio patches) might be present in the initial geometry or develop as the surface propagates. In these regions, bad quality mesh elements are likely to be generated [26]. In addition, following this strategy, global mesh connectivity cannot be preserved unless strict topological consistency is maintained, i.e. the number of topological entities and their connections remain unchanged. In some situations, this might not be feasible even for minor deformations, intermittent remeshing is then necessary.
4.2. Global surface mesh

B-rep patches are essentially a means to provide a compact definition for the geometry, and generally bear no physical meaning, so the mesh should ideally not be aware of patch boundaries, except for feature edges. Therefore, a better suited approach would be to adapt a global surface mesh without considering the segmentation of the surface into patches.

Indirect mesh generation and smoothing techniques can still be used, however a global parameterization of the surface must be constructed. While such methods have been successfully implemented for mesh generation [26], they rely on piecewise linear approximations and the initial, high-order representation is lost.

On the other hand, direct methods can be used, i.e. mesh generation and smoothing are performed directly in 3D space. Whereas direct techniques have been developed for generating meshes over composite parametric surfaces with elements potentially lying on multiple patches [27], mesh smoothing for such configurations is fairly complex.

Surface mesh smoothing in 3D is much more challenging than its 2D counterpart, due to additional geometric constraints. Classical techniques such as Laplacian smoothing tend to smear out the sharp features and move mesh vertices off of the actual surface (defined by the set of patches). Feature- and volume-preserving smoothing methods for surface meshes without CAD definition have been developed [28]. The use of our underlying dynamic B-rep model may allow to derive simpler versions of such techniques, and is the subject of ongoing work.

5. Numerical examples

In this section, the presented method is illustrated with two examples of surface propagation problems.

5.1. Sphere in reversal vortex flow

In this first example, the accuracy of our tracking method is assessed by considering a classic yet challenging example of surface propagation: a sphere of radius \( R = 0.15 \) centered at \( c = (0.5, 0.75, 0.5) \) is placed in a strong, non-uniform vortical flow defined by the velocity field

\[
V(x, y, z) = (\sin^2(\pi x)(\sin(2\pi z) - \sin(2\pi y)), \sin^2(\pi y)(\sin(2\pi x) - \sin(2\pi z)), \sin^2(\pi z)(\sin(2\pi y) - \sin(2\pi x))).
\]

The flow is rendered periodic by multiplying velocity by \( \cos(\pi t/T) \), where \( T = 4 \). As shown by the snapshots on Fig. 7(a),(b),(c), the sphere undergoes severe distortions, without developing geometric singularities. Time-integration is performed using the fourth-order Runge-Kutta scheme.

Convergence of the spectral approximation is studied for different levels of discretization, at times \( t = 0 \) and \( t = T \), when the deforming surface is supposed to be identical to the initial sphere. Influence of the time step is also studied. Error, defined as

\[
E(x) = \frac{|R - ||x - c||_2|}{R},
\]

is evaluated on a uniform sample of 50 × 50 points for each patch. Results are reported on Fig. 7(d).

As the velocity field does not depend on geometric differential quantities of the surface, the contribution of spatial discretization to approximation error is essentially observable at \( t = 0 \). The black, dotted curve on Fig. 7(d) demonstrates superlinear convergence is achieved for increasing levels of discretization. Additional error is caused by time discretization, which, in this case, is \( O(\Delta t^4) \).

5.2. Burning solid propellant grain

In this second example, the burning surface of a solid rocket propellant grain is considered. The geometry, essentially a cone with five fins, contains many sharp features. The B-rep model (one fifth of the actual solid) is constituted of about 30 patches, which are represented in different colors in Fig. 8. Gray, transparent patches represent inert walls that are not propagated. The rest of the surface propagates under uniform velocity.

A dynamic surface mesh with fixed connectivity is derived using the approach described in Section 4.1 (Fig. 9(a),(b),(c). The mesh is designed to be uniform, isotropic at an intermediate state of the surface propagation.
Fig. 7: (a),(b),(c) Snapshots of the sphere in reversal vortex flow at times $t = 0, T/4, T/2$, respectively. (d) Error at $t = T$ for increasing levels of both spatial- and time-discretization.

Time evolution of the worst element quality is plotted on Fig. 9d. Both smoothing procedures (spring-based only and combined spring- and optimization-based) significantly improve quality of the most distorted elements. However, the green curve in Fig. 9(d) plateaus below the quality threshold triggering optimization, set here to 0.7.

Actually, very few triangles (less than 0.1% for most of the deformation) do not reach this threshold. These triangles have their three vertices fixed on submesh boundaries so their quality cannot be improved. This is a clear limitation of the strategy described in Section 4.1. Investigation of the second strategy is expected to yield further improvements. Eventually, comparisons with industrial standards such as the MESQUITE Mesh Quality Improvement Toolkit [29] will be performed.

In ongoing work, this dynamic surface mesh will be used as the boundary of a deforming, body-fitted volume mesh for complex CFD simulation of the internal flow inside a solid rocket booster with burning propellant grain, as in [6]. Mesh quality is crucial for good convergence of the CFD computation, as distorted element may lead to ill-conditioned matrices. As the volume mesh deforms also with fixed connectivity, its quality degrades quickly up to a critical point where remeshing both the surface and the volume becomes necessary. Prevention of this critical point as well as remeshing methodologies will be investigated in future work.

6. Conclusion

We have presented a framework to propagate surfaces defined as boundary-representation models. The CAD definition of the surface is maintained and updated during propagation. This framework relies on several key components. First, a Chebyshev pseudo-spectral method developed for the Lagrangian tracking of parametric surfaces yields high-order accuracy for smooth geometries. Second, techniques for dealing with geometric singularities allow to address
more complex configurations by resolving intersection and separation of surface patches. In addition, a strategy for deriving a dynamic mesh conforming to the propagating surface is proposed.

The presented method provides accurate, dynamic representations of propagating surfaces for problems central to many engineering and scientific applications. This algorithm can be integrated into deformable volume mesh methodologies and coupled with Finite Elements/Volume solvers for the simulation of complex, multi-physics problems with evolving three-dimensional geometries, such as combustion, ice build-up or fluid-structure interaction.
Furthermore, its ability to preserve surface mesh connectivity during propagation makes our method suitable for block-structured volume mesh generation as an advancing-front method. High-order surface and volume meshes could also be generated, by taking advantage of the high-degree polynomials used for patch parameterization. Finally, flexible, dynamic B-rep models generated by our algorithms could be used for CAD-based methods such as design optimization.

References


