

# MULTIPLE APPROACHES TO FRAME FIELD CORRECTION FOR CAD MODELS

Maxence Reberol<sup>1\*</sup>

Alexandre Chemin<sup>2</sup>

Jean-François Remacle<sup>3</sup>

<sup>1</sup>*Université catholique de Louvain, Avenue Georges Lemaitre 4, 1348 Louvain-la-Neuve, Belgium*

## ABSTRACT

Three-dimensional frame fields computed on CAD models often contain singular curves that are not compatible with hexahedral meshing. In this paper, we show how CAD feature curves can induce non meshable 3-5 singular curves and we study four different approaches that aims at correcting the frame field topology. All approaches consist in modifying the frame field computation, the two first ones consisting in applying internal constraints and the two last ones consisting in modifying the boundary conditions. Approaches based on internal constraints are shown not to be very reliable because of their interactions with other singularities. On the other hand, boundary condition modifications are more promising as their impact is very localized. We eventually recommend the 3-5 singular curve *boundary snapping* strategy, which is simple to implement and allows to generate topologically correct frame fields.

**Keywords:** frame field, hexahedral meshing, block decomposition

## 1. INTRODUCTION

In the last decade, frame field based approaches (§2) have become popular and promising for hexahedral and hex-dominant meshing. However, state-of-art frame fields are not guaranteed to have the right topology for full hexahedral meshing [1, 2, 3], especially when the CAD model contains feature curves on the boundary (§3).

In §3.3, we show that the 3-5 singular curves frame field singularity graph issue, i.e. singular curve where the hexahedral valence smoothly transitions from three (index  $+1/4$ ) on one extremity to five (index  $-1/4$ ) on the other one, is due to the presence of concave and curved feature curves on the model. This 3-5 singular curve is the most common problem that arises in 3D frame field topologies.

Based on these observations, we investigate four approaches to automatically correct the frame field topology (§4). We present the reasoning behind each approach as well as their pros and cons (frame fields corrected successfully, failures, and limitations). Two

of the approaches rely on extrusion of objects inside the frame field: (i) the extrusion of boundary feature curves (§4.1) and (ii) the extrusion of singular curve extremities (§4.2). Extrusion based methods work on simple cases but fail when the extrusion process is perturbed by other singularities of the frame field.

Another possibility for correcting frame fields is to remove the concave feature curves, by transforming them into fillets (§4.3). In other words, irregular 3-5 transitions are much less present in smooth models. In the smooth case, frame fields are generally compatible with hex meshing, but the difficulty is to map the singularity graph onto the non-smooth original geometry. Instead, and it is what we consider to be our best approach, we propose to remove the 3-5 singular curves by snapping them on the boundary and adapting the frame field boundary conditions (§4.4).

The presence of 3-5 singular curves in frame fields is one of the main issues that prevents frame-field based hexahedral meshing and automatic blocking. By removing them, our snapping correction extends the range of application of frame-field based meshing technology to a larger set of CAD models, where previous

---

\*maxence.reberol@uclouvain.be

techniques would fail because of incorrect frame field topology. Nevertheless, this approach is limited to 3-5 singular curves close to the boundary, which are the most frequent in CAD, and cannot handle arbitrary cases of 3-5 singular curves.

## 2. RELATED WORK

The standard frame field based meshing approach consists in building a smooth and boundary-aligned frame field (§2.1) and using it as a guide to build a hex-dominant [4, 5, 6, 7, 8] or a fully hexahedral mesh [9, 10, 11, 3, 12, 13]. When the frame field singularity graph does not match a feasible hexahedral mesh topology, mixed-integer parameterization approaches fail and hex-dominant ones produce lot of tetrahedra. Some heuristic-based attempts to correct the frame fields have been tried (§2.2), but the problem remains largely unsolved.

### 2.1 Frame field design

In dimension two, crosses are objects made of two orthogonal directions, invariant by the four rotations of respectively 0, 90, 180 and 270 degrees. Cross fields are usually represented by 2D vector fields of the form  $\mathbf{f}(\mathbf{x}) = (\cos(4\theta(\mathbf{x})), \sin(4\theta(\mathbf{x})))$  [14]. To get a smooth vector field in a 2D domain  $\Omega$ , the natural way is to minimize the Dirichlet energy  $\int_{\Omega} \|\nabla \mathbf{f}\|^2$  under Dirichlet boundary conditions. One issue with this approach is that the gradient of the frame field tends to infinity at singularities in the continuous setting. This is either ignored, as the energy stay finite after discretization, or addressed by using a scaling scalar field [15], or more recently by turning to the Ginzburg-Landau theory [16, 17]. In practice, the frame fields obtained with these different approaches are similar, with singularities whose indices are compatible with quadrilateral meshing.

In dimension three, frames are made of three orthogonal directions and are invariant by the twenty-four rotations of the cube. For the generation of boundary aligned smooth frame field, the ideas are similar to the 2D case: find a convenient representation of the frame field and minimize the Dirichlet energy. However, finding a unique and continuous representation is more tricky. The current best candidates are spherical harmonics [18] and fourth-order tensors [19]. In both cases, the spaces are of dimension nine and the frames live on a manifold of dimension three. The Dirichlet energy is minimized while staying on the frame manifold, which in practice is done either by optimizing Euler angles associated to frames [20] or iteratively with a non-linear solver and by using recurrent projections on the frame manifold [19]. The resulting frame fields exhibit singularities made of internal curves, that is

usually called the singularity graph. As in dimension two, the norm of the gradient tends to infinity at the singular curves, which can be seen as extrusions of the singular nodes of a boundary cross field.

Contrary to the 2D case, frame field singularity graphs do not always correspond to feasible hexahedral mesh connectivity. This issue has been extensively described in recent articles [1, 2, 3]. To our knowledge, there exist no frame field generation approach that has any kind of guarantee to provide a singular graph that is compatible with hex meshing.

### 2.2 Frame field correction

A simple 3D model exhibiting a fundamentally invalid singularity graph is the *notch model* [1, 2] (Figure 1), which is the boolean difference between a box and a cylinder. The singularity graph produced by all existing frame field methods is made of a single curve whose hexahedral valence is three on one extremity and five on the second one, that will be called a *3-5 singular curve* in the rest of the paper. In a hexahedral mesh, it is not possible to have an interior vertex adjacent to only one valence three edge and one valence five edge (other adjacent edges being regular, *i.e.* valence four). The only valid vertex configurations are the ones which are topologically equivalent to sphere triangulations, as described in [3]. When a singularity graph contains a different configuration, *e.g.* a 3-5 singular curve, we say it is invalid or not hex-meshable.

Given a valid singularity graph, there exist methods that allow to compute a smooth frame field, *e.g.* [3, 21]. Thus, correcting a frame field can be achieved by generating a valid singularity graph, *i.e.* the set of irregular edges of the associated block decomposition. But generating a valid singularity graph from scratch remains a totally open problem. An alternative approach is to start from an initial frame field, possibly non-meshable, and modify it in order to make it hex-meshable.

In [2], a valid frame field for the *notch model* is built by either manually extruding the concave feature curve inside the model (producing an internal surface) or by manually adding a fillet to the feature curve. In the present paper, we propose two approaches (§4.1, §4.3) that aim at automatizing these manual interventions.

To avoid the block decomposition degeneracies caused by a 3-5 singular curve, Zheng and his co-authors [12] propose to replace it by two singular curves, one with a valence of three and one with a valence of five, whose geometries are determined by tracing streamlines starting at the extremities of the 3-5 curve. One of our correction approach (§4.2) is based on this strategy.

It is also worth mentioning that there can be local defects in the extracted singularity graph, *e.g.* the *zig-zag* issue. They can be corrected with local operations, as detailed in [22]. We are not interested in these issues in this paper as they are artifacts of discretization choices, and it is possible to avoid them by using a frame per vertex instead of one per tetrahedron.

### 3. BOUNDARY-ALIGNED SMOOTH FRAME FIELD

In this section, we first describe the energy minimization formulation of the boundary-aligned smooth frame field problem (§3.1), which is common to most of the recent 3D frame field solvers. In this paper, the frame field is discretized on a tetrahedral mesh  $\mathcal{T}$ , by defining one frame per vertex (§3.2). More precisely, we use a continuous piecewise linear approximation of nine frame coefficients, which fully determine a 4-th order tensor representation of the frames [19]. The following section (§3.3) is a qualitative analysis of the associated frame field singularity graphs and of how boundary conditions can lead to configurations which do not correspond to hexahedral meshes.

As we do not enter into the details, readers unfamiliar with frame fields should refer to existing articles that explain extensively the theory and implementation of such fields, *e.g.* [18, 20, 23, 19].

#### 3.1 Continuous frame field formulation

The goal is to compute a frame field as smooth as possible and that is aligned with the boundaries/features of the model. The natural approach is to translate these requirements into a Dirichlet energy minimization problem. Minimizing the Dirichlet energy (1) ensures the smoothness in the domain and the Dirichlet boundary conditions (2) enforce the alignment with the model boundary. Formally, the frame field  $\mathbf{f}$  is the solution to the problem:

$$\mathbf{f} = \operatorname{argmin}_{\mathbf{f}(\mathbf{x}) \in \mathcal{O}} \int_{\Omega} \|\nabla \mathbf{f}\|^2 \quad (1)$$

subject to boundary conditions:

$$\begin{cases} \mathbf{f}(\mathbf{x}) = \mathbf{g}(\mathbf{x}) & \text{for } \mathbf{x} \in \partial\Omega_d \\ \mathbf{f}(\mathbf{x}) \perp \mathbf{n}(\mathbf{x}) & \text{for } \mathbf{x} \in \partial\Omega_s \end{cases} \quad (2)$$

with  $\mathcal{O}$  the space of frames,  $\partial\Omega_s$  a smooth subset of the model boundary  $\partial\Omega$ , where we want the frames to be tangent to the boundary,  $\mathbf{n}$  the boundary normal, and  $\partial\Omega_d$  another subset of  $\partial\Omega$  where we impose the three directions of the frames. Usually,  $\partial\Omega_d$  corresponds to the feature curves of the model, also called hard-edges

or ridges. The symbol  $\perp$  means that one of the three frame directions is parallel to a given vector (tangency constraint).

Before going further, one needs to choose a representation of the frames. Frames live in a space, denoted  $\mathcal{O}$ , which is the quotient space of the space of rotations  $SO(3)$  with the octahedral group  $O$  [23]. Unfortunately, there is no simple representation of these objects. To date, two representations have been proposed: spherical harmonics of degree four [18] and fourth-order tensors [19]. In both cases, frames live on manifolds of dimension three immersed in  $\mathbb{R}^9$ . There is an isomorphism between the two representations, so they are essentially equivalent. The continuity and uniqueness of these representations is a necessary condition to use directly their nine coefficients to compute distances and gradients, which is analogous to using the Euclidean distance on the unit circle instead of the circle arc length.

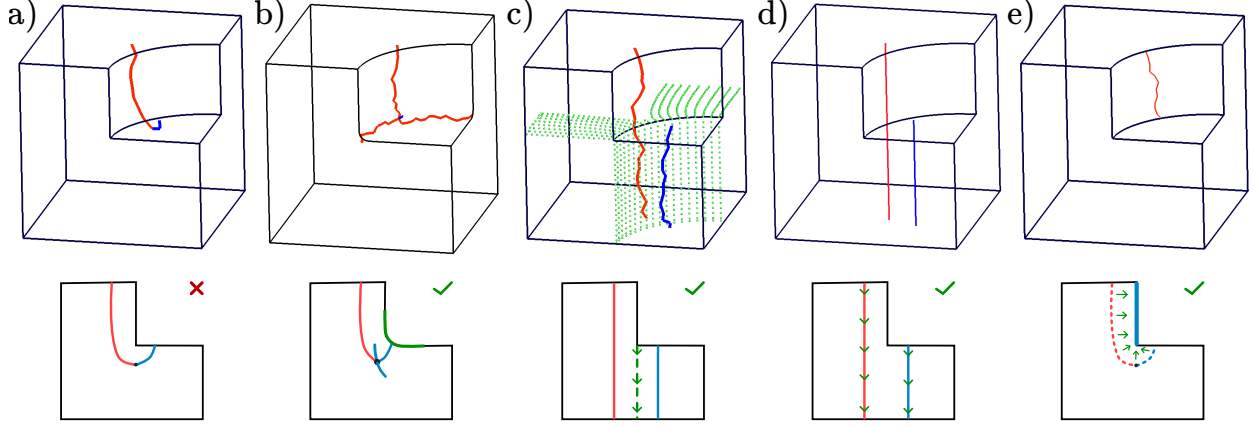
In the continuous setting, the Dirichlet energy (1) blows up because of the presence of singular curves and this singular behavior is the cause of many issues for the discretization.

#### 3.2 Discretization of the frame field problem

To find a numerical solution to the frame field problem introduced previously (§3.1), one needs to choose an approximation space for the frame field and a numerical scheme to solve the nonlinear problem (1).

We discretize the frame field on a tetrahedral mesh  $\mathcal{T}$  and use a continuous piecewise linear approximation of the frame coefficients, which can be coefficients of either the spherical harmonics representation or of the 4-th order tensor one. The discretized frame field  $\mathbf{f}_h$  is now entirely defined by its coefficients  $\mathbf{f}_i \in \mathbb{R}^9$  at each vertex  $v_i \in \mathcal{V}$  of the tetrahedral mesh. It is worth noting that inside a tetrahedron, the linearly interpolated coefficients do not correspond to a frame, but the closest frame can be obtained by projection on the frame manifold  $\mathcal{O}$ . Compared to the *one frame per tetrahedron* discretization, the piecewise linear approximation is more efficient: less unknowns for the same mesh, better accuracy [20], possibility to use linear finite elements.

If each  $\mathbf{f}_i$  is a frame, then the singularities, which correspond to infinite gradients in the continuous formulation, cannot be represented at vertices. With this approach, one possibility is to detect singularities by looking at axis permutations along edge loops. The smallest loops to look at are the internal faces of the tetrahedral mesh, which are simply triangles. The singularity graph is then made of connected chains of tetrahedra. In practice, it forces the singularity graph



**Figure 1:** a) Natural frame field singularity graph, made of one non-meshable 3-5 singular curve. b) When replacing the concave feature curve by a fillet, the new singularity graph is made of three valence-five and one valence-three singularity curves, which are compatible with hexahedral meshing. c) Via the streamlines (green), the feature curve is extruded inside the volume and the resulting singularity graph is made of two singular curves (one valence five and one valence three). d) Tracing streamlines from boundary singular nodes generates two valid singular curves. e) The initial invalid 3-5 singular curve of a) is snapped to the boundary surface, creating a new valence three feature curve. *Bottom row:* schematic view of the corrections, in the diagonal cut view.

to be locally very distorted as it must be contained inside tetrahedra, whose facets are randomly oriented in the mesh.

Another way to deal with singularities is to allow the coefficients  $\mathbf{f}_i$  to represent objects which are not frames. This is analogous to letting 2D crosses tend to zero at singularities instead of staying unit vectors. The advantage is that the singularities are smoother, because they are less affected by the tetrahedral mesh, but they are also more diffuse and there is no longer a clear localization of the singularities.

Frame field solvers usually work in two stages : initialization via Laplacian smoothing of the frame coefficients, without enforcing the frame constraint  $\mathbf{f} \in \mathcal{O}$ , followed by a smoothing of the frames, where the frames  $\mathbf{f}_i$  must lie on  $\mathcal{O}$  or stay close to it. The process usually converges to a local minimum which is not far from the initialization [20].

The frame coefficients can be forced to stay on the frame manifold  $\mathcal{O}$  by either recurring projections [19] or by optimizing the Euler angles of the associated rotations [20].

In any case, the Dirichlet energy associated to the frame field tends to infinity with mesh refinement, but stays finite because of the discretization. Consequently, these approaches only work on uniform meshes. On non-uniform meshes, the singularities move to areas of coarse elements as the same singularity graph topology can be represented while costing less energy.

We use 4th order tensors to represent the frames, and

we allow the coefficients  $\mathbf{f}_i$  to deviate from  $\mathcal{O}$  at singularities. This particular choice is not important for the rest of the paper: the correction techniques we study can be applied to all energy-minimizing frame field solvers.

### 3.3 Frame field singularity graph behavior

Boundary aligned frame fields produced by energy minimizing methods are interesting because they exhibit singularities that form a graph, which is topologically similar to the singular edges of a hexahedral mesh. Unfortunately, as described in the literature (§2.2), this graph is not always topologically equivalent to a valid hexahedral mesh.

As singular curves cost a lot of energy in (1), an amount tending to infinity with mesh refinement, the energy-based formulation of the frame field problem is asymptotically looking for solutions that minimize the length of the singular curves.

In CAD models, there are many feature curves on the boundary where we impose the frames, via the Dirichlet boundary conditions (2). These feature curves split the boundary in multiple patches, which become independent if we consider the associated surface cross field problems. This boundary *splitting* leads to the apparition of singular nodes on the surface, which are necessary to have coherent cross fields that respect the Poincaré-Hopf theorem. But from the point of view of the volume frame field, the boundary singular nodes must be the extremities of singular curves, as there are no isolated singular nodes in a 3D frame field.

A simple example exhibiting this behavior is a box with a circular arc imprinted on one face, see Figure 2.a.. This example can be seen as a simplification of the *notch* model. In this example, the two patches on top must have singular nodes to accommodate their boundary conditions, but the frame field solver, which is minimizing the Dirichlet energy, do not propagate these singularities inside the model as this would cost a lot of energy, but merge them as soon as possible in the volume. Models with such configurations are, for example, the ones containing boolean operations involving spheres or cylinders that do not go through the whole model, which are common in CAD modelling.

Our interpretation is that the energy-minimization formulation (§3.1) is similar to the Laplace equation  $-\Delta \mathbf{f} = 0$ , even with the additional constraint  $\mathbf{f}(\mathbf{x}) \in \mathcal{O}$ . Hence, this is essentially a smoothing kernel that act locally when possible. Constraints from the boundary that cost energy (non-zero frame gradient) do not propagate far in the volume.

On the other hand, hexahedral mesh topological constraints (chords, sheets) propagate on arbitrary long distances. Thus, hexahedral meshes associated to frame fields are often different from the ones that a user would produce by manually building a block decomposition.

An ideal answer to this issue would be a new frame field problem formulation that allows better propagation of boundary constraints, but unfortunately none has been successfully developed up to now.

Another important observation is that even if the frame field have a wrong topology, one of the frame directions is not affected (the vertical ones in the *notch* and *box with arc* examples). At the transition from valence three to valence five in the 3-5 singular curve, the frame field stable direction along the singularity is no longer tangent to the curve. Exploiting this stable direction is the basis of the *feature curve extrusion* (§4.1) and *singularity extrusion* (§4.2) correction approaches. It also worth noting that [2] tried to penalize the incoherency between the frame field stable direction and the singular curve tangents, in order to iteratively correct the frame field, but it leads to numerical instability and this approach was eventually unsuccessful.

## 4. FRAME FIELD CORRECTION TECHNIQUES

We present and discuss various strategies to automatically correct frame fields, for them to be suitable for full hexahedral meshing. We focus on removing the 3-5 singular curves.

Feature curve extrusion (§4.1) and transformation of feature curves into fillets (§4.3) are automation of the manual corrections applied to the *notch model* in [2]. Singularity extrusion (§4.2) has been introduced by [12] in the context of dual surface construction. Snapping of 3-5 singular curves to the boundary (§4.4) has not been used previously, to our knowledge. The four corrections are illustrated side-by-side on Figure 1 and Figure 3.

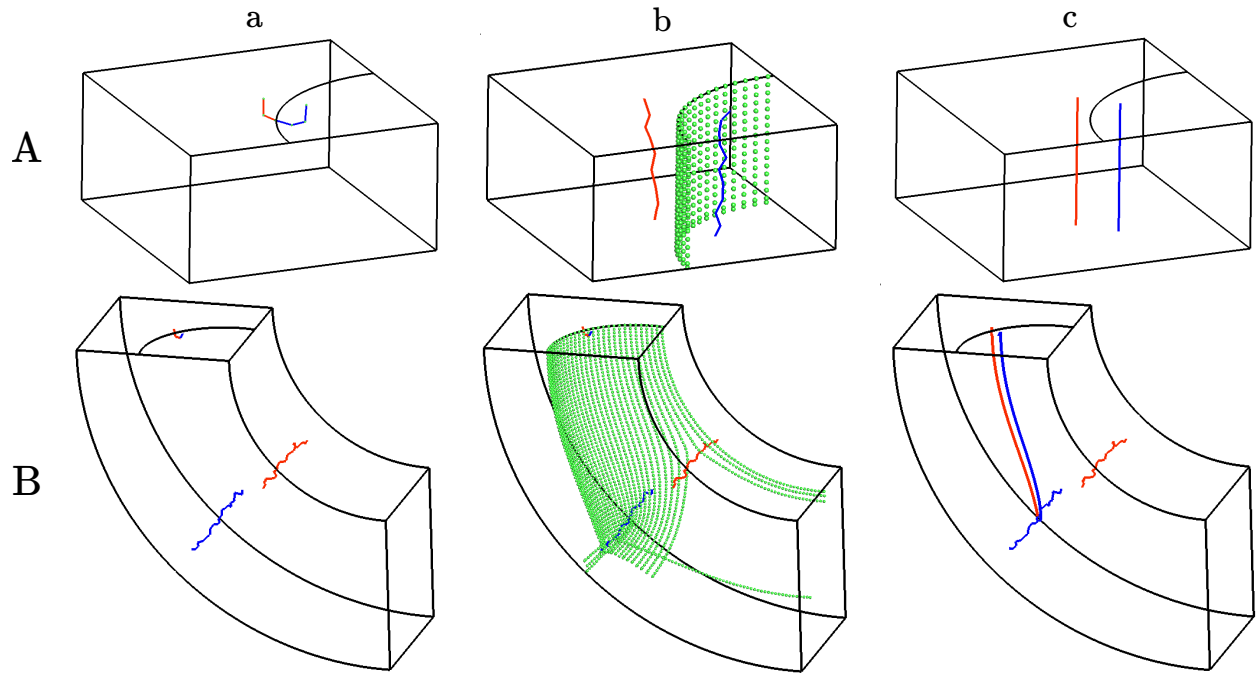
We show failure cases for the extrusion approaches (§4.1, §4.2), where the extruded objects interact badly with other singular curves (Figure 2). The transformation of feature curves to fillet (§4.3) is difficult to apply in practice because it requires to map the new singularity graph back to the initial geometry, which we did not automate. The snapping of 3-5 singular curves (§4.4) seems the most promising approach in our opinion, but it leads to hexahedra with invalid geometry (zero jacobian at some corners) and a post-processing (insertion of sheets) is required to have hexahedral meshes suitable for numerical simulation.

### 4.1 Feature curve extrusion

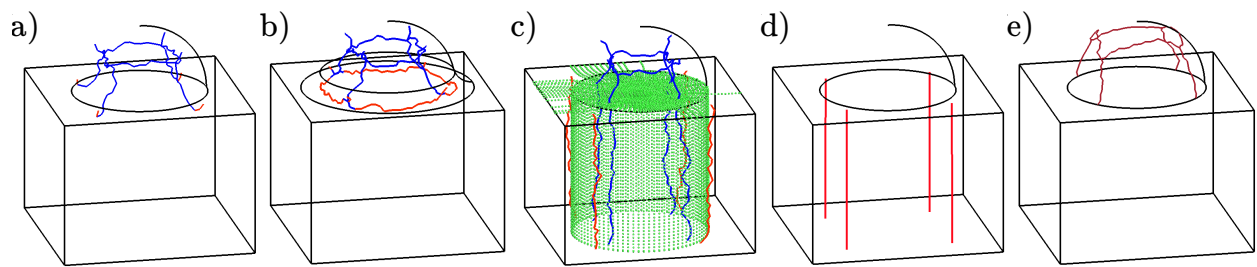
To recap the previous frame field observations (§3.3): a concave feature curve can cause 3-5 singular curves, but when it happens there is a stable direction in the frame field which is not affected. We propose to use the stable direction of the frame field to extrude the concave feature curves inside the model. Once the new internal surface (extrusion of the curve) is generated, we add internal constraints to the frame field formulation and compute a new one. While this approach works for some models, it does not when the extrusion process (streamline tracing) is perturbed by other singularities.

**Tracing streamlines from feature curves** The simplest approach would be to propagate the feature curves straight into the model, following the directions of the initial boundary normals. It would work for simple *blocky* models, but not when the boundaries are curved, because the new internal surfaces would not follow the curvature of the model boundaries.

Instead, we extrude the concave feature curves by using certain directions in a initial frame field. In practice, we achieve this extrusion by tracing streamlines, that start from the feature curves and end on the first reached boundaries. In our case, the tracing is done by using a standard fourth order Runge-Kutta explicit scheme, very similar to the one used with vector fields. The only difference is that at each new position, we extract the frame direction which is closest to the previous direction. The reader can refer to Algorithm 1 for a more detailed description of the streamline tracing



**Figure 2:** *Model A:* box with imprinted circle arc, simplified version of the *notch model*. *Model B:* same model but the box has curvature, which causes two additional singular lines. *Col. a:* singularities of the initial frame field. *Col. b:* feature curve extrusion, works for A but fails for B because of interaction with other singularities, which split the streamline trajectories. *Col. c:* singularity extrusion, works for A but fails for B because the streamlines hit another singularity.



**Figure 3:** Half-sphere on top of a box. a) Natural frame field singularity graph, containing four non-meshable 3-5 singular curves. b) When replacing the concave feature curve by a fillet, the new singularity graph contains four additional singular curves of valence five, making the graph compatible with hexahedral meshing. c) With internal constraints (green), the feature curve is extruded inside the volume and the resulting singularity graph is valid. d) Tracing streamlines from boundary singular nodes replace the four 3-5 singularities by four singularities of valence five, but four valence three singular lines are missing. These ones should be traced from the internal singular nodes. e) The four 3-5 singular curves are snapped to the boundary, forcing snapping of the other singularities.

---

**ALGORITHM 1:** Streamline tracing in frame field

---

**Input:** Initial position  $\mathbf{p}_0$   
Initial direction  $\mathbf{v}_0$   
Step length  $h$  for the explicit scheme  
**Output:** Streamline  $\mathcal{S}$  described by an ordered  
list of points  $(\mathbf{p}_l)_l$  and their associated  
directions  $(\mathbf{v}_l)_l$

```

 $\mathbf{p} = \mathbf{p}_0$ 
 $\mathbf{f}(\mathbf{p}) = \text{interpolate\_frame\_field\_at}(\mathbf{p})$ 
 $\mathbf{v} = \text{closest\_direction}(\mathbf{v}_0, \mathbf{f}(\mathbf{p}))$ 
repeat
  append  $(\mathbf{p}, \mathbf{v})$  to  $\mathcal{S}$ 
   $\mathbf{p} = \mathbf{p} + \frac{h}{6}(\mathbf{v}_1 + 2\mathbf{v}_2 + 2\mathbf{v}_3 + \mathbf{v}_4)$  with
   $\mathbf{v}_1 = \text{closest\_direction}(\mathbf{v}, \mathbf{f}(\mathbf{p}))$ 
   $\mathbf{v}_2 = \text{closest\_direction}(\mathbf{v}_1, \mathbf{f}(\mathbf{p} + \frac{h}{2}\mathbf{v}_1))$ 
   $\mathbf{v}_3 = \text{closest\_direction}(\mathbf{v}_2, \mathbf{f}(\mathbf{p} + \frac{h}{2}\mathbf{v}_2))$ 
   $\mathbf{v}_4 = \text{closest\_direction}(\mathbf{v}_3, \mathbf{f}(\mathbf{p} + h\mathbf{v}_3))$ 
   $\mathbf{v} = \mathbf{v}_4$ 
until  $\mathbf{p}$  outside tetrahedral mesh  $\mathcal{T}$ ;

```

---

process.

To determine the initial directions for the propagation, we look at the appropriate hexahedral valence associated to the feature curve. It is usually one or three, and occasionally two for user-inserted curves on smooth surfaces, such as the circular arc on the Figure 2, or valence four for model with cuts whose dihedral angles are close to 360 degrees. For valence one, there is no need to propagate the curves as the frame field directions are already imposed by the surface normals at both sides of the curve. For valence two, three and four, we respectively extend the curve in one, two and three directions.

Successful applications of the streamline tracing process can be seen on Figures 1.c., 2.A.b and 3.c., where the extruded curves are shown with green points.

**Internal constraints for frame field** To compute a new frame field that respects the extruded curve, we add internal constraints in the boundary conditions (2) of the frame field formulation (§3.1). The new constraints (3) are tangential conditions, forcing one direction of the frames, similar to the boundary conditions on the smooth parts of the boundary.

Consider a streamline  $\mathcal{S} = \{(\mathbf{p}_k, \mathbf{v}_k)\}_k$ , made of  $(\mathbf{p}_k)_k$  the points,  $(\mathbf{v}_k)_k$  the associated directions, started from the edge  $e$ , of tangential direction  $\mathbf{t}_e$ . Then the associated internal conditions are:

$$\forall k, \quad \mathbf{f}(\mathbf{p}_k) \perp (\mathbf{t}_e \times \mathbf{v}_k) \quad (3)$$

To apply these internal conditions precisely, we com-

pute a new tetrahedral mesh containing the points  $(\mathbf{p}_k)_k$  associated to all the streamlines traced from the concave feature curves.

In our examples, the singularity graphs of the new frame fields are displayed on Figures 1.c., 2.A.b and 3.c.. They correspond to valid hexahedral meshes, similar to the ones a user would generate by manually building the block decompositions.

**Failure cases and limitations** Extruding a feature curve inside a frame field is equivalent to tracing a sheet. If the sheet encounters another singularity orthogonally, then it is sheared in multiple parts. When the multiple parts can stay inside a future valid sheet of the hexahedral mesh, as for the horizontal sheet in Figure 1.c., then our approach still works. But if the sheet is sheared into parts that are sent in different arbitrary directions, then the new internal surface makes no sense and should not be used to constrain the frame field, as it would definitely produce an incoherent frame field. This failure case is illustrated on Figure 2.B.b..

The feature curve extrusion is also not applicable to models where the frame field contains limit cycles, such as the Nautilus example in Figure 1 of [2]. In such cases, which are rare, the streamlines will spiral indefinitely without reaching any boundary. They can be detected by monitoring the streamline lengths.

Regarding the model on Figure 2.B., other frame field solvers could produce a frame field without the pair of valid singularities that accommodates the curvature. In that case, the feature curve extrusion correction would work. However, it is still possible to add features to the model, *e.g.* boolean difference with a cylinder, that would introduce singularities to perturb the extrusion of the feature curve.

The initial 3-5 singular curve is a local solution of the frame field solver to the boundary constraints. By extruding the feature curve using the stable directions, we are trying to generate global constraints, as they typically propagate through the whole model. We think this approach cannot work in the general case, as there is no reason that the feature curve conserves its shape during the extrusion process in the frame field.

## 4.2 Boundary singular node extrusion

Another possible extrusion approach is to propagate the boundary singular nodes via the frame field stable directions, instead of propagating the concave feature curves. This approach has been implemented in [12] to separate the contour of dual sheets, in the context of block decomposition. The additional step in our case is to compute a new frame field that respect these

forced singularities.

The hope with this approach, compared to the feature curve extrusion (§4.1), is that node extrusion produce curves that would not interact with other singularities, as they are much smaller than surfaces.

To extrude the singular node extremities of 3-5 singular curves, we use the same streamline tracing method (Algorithm 1) that was used for curve extrusion (§4.1). For each extremity, we trace a streamline in the direction of the stable direction, until it reached a boundary.

The streamline vertices are added in a new tetrahedral mesh and specific internal conditions are applied on them: forcing them be singular frames with a unique imposed direction. Another alternative could be to use frame field generation methods with fixed singularity graph [3, 21], but it would require imposing an entire singularity graph and we prefer to use a more flexible frame field solver that still has the possibility to displace the previous valid singularities, and potentially to spawn new ones.

Successful applications are shown on Figures 1.d., 2.A.c and 3.d.. They correspond to the same valid hexahedral meshes as the ones generated by the previous approach (§4.1). On Figure 3.d., the four internal valence three singularities are missing because we did not implement the tracing from an extremity which is an internal node, but it would work.

**Limitations** This approach suffers from the same failure case than the feature curve extrusion, which is shown on Figure 2.B.c.. Both streamlines hit an existing singularity during the tracing process, leading to an incoherent singularity graph.

The singularity extrusion correction technique may have a better success rate than the feature curve extrusion on specific models (less risk of collision), but it suffers from the same fundamental flaw (interaction with other singularities) and does not work in the general case.

### 4.3 Transforming concave feature curves into fillets

In the frame field analysis section (§3.3), we explained that non hex-meshable 3-5 singular curves are caused by feature curves, at least in the cases we studied. We can try to remove all hard-edges and compute frame fields on smooth models, assuming that energy-minimizing frame fields on smooth models are less likely to contain 3-5 singular curves.

Even if smooth models can contain invalid 3-5 singular curves, *e.g.* the *rockerarm* in Figure 10 of [3], they are

much less widespread than with CAD models in our experience. We think it is worth trying to exploit the smoothing of feature curves to correct frame fields. A feature curve (made of hard-edges) can be replaced by a smooth transition between the two surface patches adjacent to the curve. In terms of CAD modeling, this is equivalent to placing a fillet on the curve, as illustrated on Figure 1.b., Figure 3.b. and Figure 5.b..

Smoothing a convex feature curve (hex-valence of one) induces an internal singular curve of valence three in the frame field and smoothing a concave feature curve (hex-valence of three) induces an internal singular curve of valence five. If the feature curve that is replaced by a fillet was responsible for a 3-5 singular curve, the newly introduced singular curves join the invalid curve at its transition between valence three and five, creating a hex-meshable singular node. In the case of the *notch* model (Figure 1.a. and b.), the new singular node connects one valence-three and three valence-five singular curves after the insertion of the fillet. The associated block decomposition is shown on Figure 4.a.. Another application is illustrated on Figure 3.b., where the frame field of the smoothed model contains four additional valence-five singular curves, making it suitable for full hexahedral meshing.

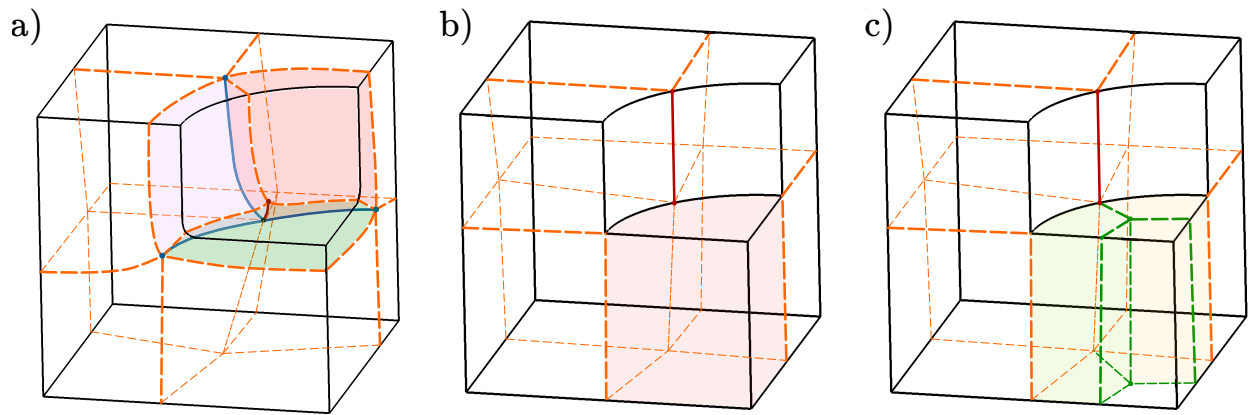
From the point of view of the block decomposition, the *fillet* correction introduces a new hexahedral layer (the colored blocks on Figure 4.a.). This layer was partially present in the initial frame field, which contained the 3-5 singular curve, but it was pinched and it did not correspond to topological blocks (six quadrangular faces). The smoothing of the hard-edges allows to recover valid blocks.

Another example of singularity graph obtained after transforming the feature curves into fillet is illustrated on Figure 5. This model is a volume version of the *box with arc* model, but with the arc replaced by a groove. As with the *notch* model, the smoothing of the feature curves introduces new singular curves that connect with the 3-5 singular curves, making all the singular nodes hex-meshable.

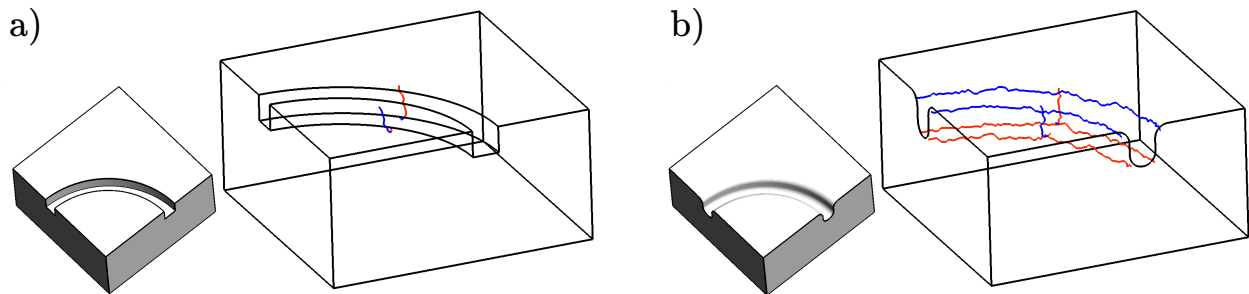
This approach is interesting because it is a local approach, which implies local modifications of the frame field, well in accordance with the spirit of the energy-minimizing frame field formulation.

**Limitations** Unfortunately, it is not straightforward to implement it in an automatic way. Once a hex-meshable frame field, or equivalently its singularity graph, is computed on the model with fillets, it must be brought back to the initial model with hard-edges. Before trying to develop an automatic technique for building such mapping, it is interesting to do it manually and observe the result.





**Figure 4:** Block decomposition of the *notch* model after correction. a) Feature curve transformed to a fillet. The block decomposition is valid. b) 3-5 singular curve snapped to boundary. The block decomposition is topologically valid but the colored block has a flat corner. c) Refinement of the block with a flat corner. The decomposition is topologically and geometrically valid.



**Figure 5:** a) Box with a circular rectangular groove on the top. The singularity graph is made of two 3-5 singular curves. b) The feature curves of the rectangular groove are transformed into fillets, making the geometry smooth. This change introduces singular curves along the groove, that connect with the previously invalid 3-5 singular curves. The new singularity graph, which contains four internal nodes, is compatible with hex meshing.

By carefully looking at the block decomposition on the smoothed *notch* model on Figure 4.a., we can see that the valence-three boundary singular node has been positioned at the center of the fillet, where the curvature is maximal. In the initial model, this node has to be mapped on the center the concave feature curve. This implies that the topological block in green will have two adjacent edges on this curve, forming a flat angle. This geometry is not valid from a numerical analysis point of view, as the jacobian of the hexahedra would be zero on this corner.

Instead of pursuing this approach, which would require lot of engineering, we focus on a similar but simpler one: the *boundary snapping* of 3-5 singular curves (§4.4).

#### 4.4 Boundary snapping of 3-5 singular curves

The previous *fillet* approach (§4.3) corrects the frame field singularity graph by adding singular curves in order to make all junctions hex-meshable. Another way to deal with the 3-5 singular curves is to make them disappear, instead of enriching them. By observing the block decomposition associated to the *fillet* correction on Figure 4.a., we can see that there is a layer of blocks close to the fillet (the colored ones). If we remove this layer, we get a new block decomposition where the singular curve is on the boundary, see Figure 4.b.. We propose to mimic this process by directly snapping 3-5 singular curves to the boundary, skipping the *fillet* correction (Figure 4.b.). A drawback is the resulting block decomposition has blocks with zero jacobian at some corners, but it was already the case with the *fillet* correction, and the final geometry can be improved by refining some blocks in a post-processing phase (Figure 4.c.).

**Singularity snapping** Our snapping strategy is simple. For each 3-5 singular curve, we snap both extremities. If an extremity is a boundary singular node, it is snapped to the closest feature curve, if it is an internal singular node, it is snapped to the closest point of the boundary surface. Other singularities (initially valid) may have only one of their extremity snapped, if so the other is also snapped. This process is applied iteratively until all necessary singularity extremities have been snapped.

Once the extremities are snapped, the geometry of the curve on the boundary must be recomputed. A simple approach is to take the shortest path between both extremities on the boundary triangle edges, *e.g.* Figure 1.e. and Figure 3.e., but a more accurate one is to build a new triangular mesh of the boundary that includes smooth curves joining the snapped extremities.

**Corrected frame field** To generate a frame field that respect the snapped curves, the boundary conditions (2) of the frame field formulation (§3.1) must be adapted. Close to the snapped curves, which can be seen as new feature curves, the corrected frame field should not be aligned with the boundary normals. From the hexahedral mesh point of view, we would like to have edges of valence one or of valence three on parts of the surface boundary that are smooth, *e.g.* locally flat.

On the snapped curves, instead of imposing tangency to the boundary normal, we impose frames (Dirichlet boundary conditions) that are tangent to the curve and 45-degrees rotated from the boundary normal (along the curve axis).

Feature curves that received a snapped extremity are split and new boundary conditions are obtained by linearly interpolating the frames at both extremities of the split curves.

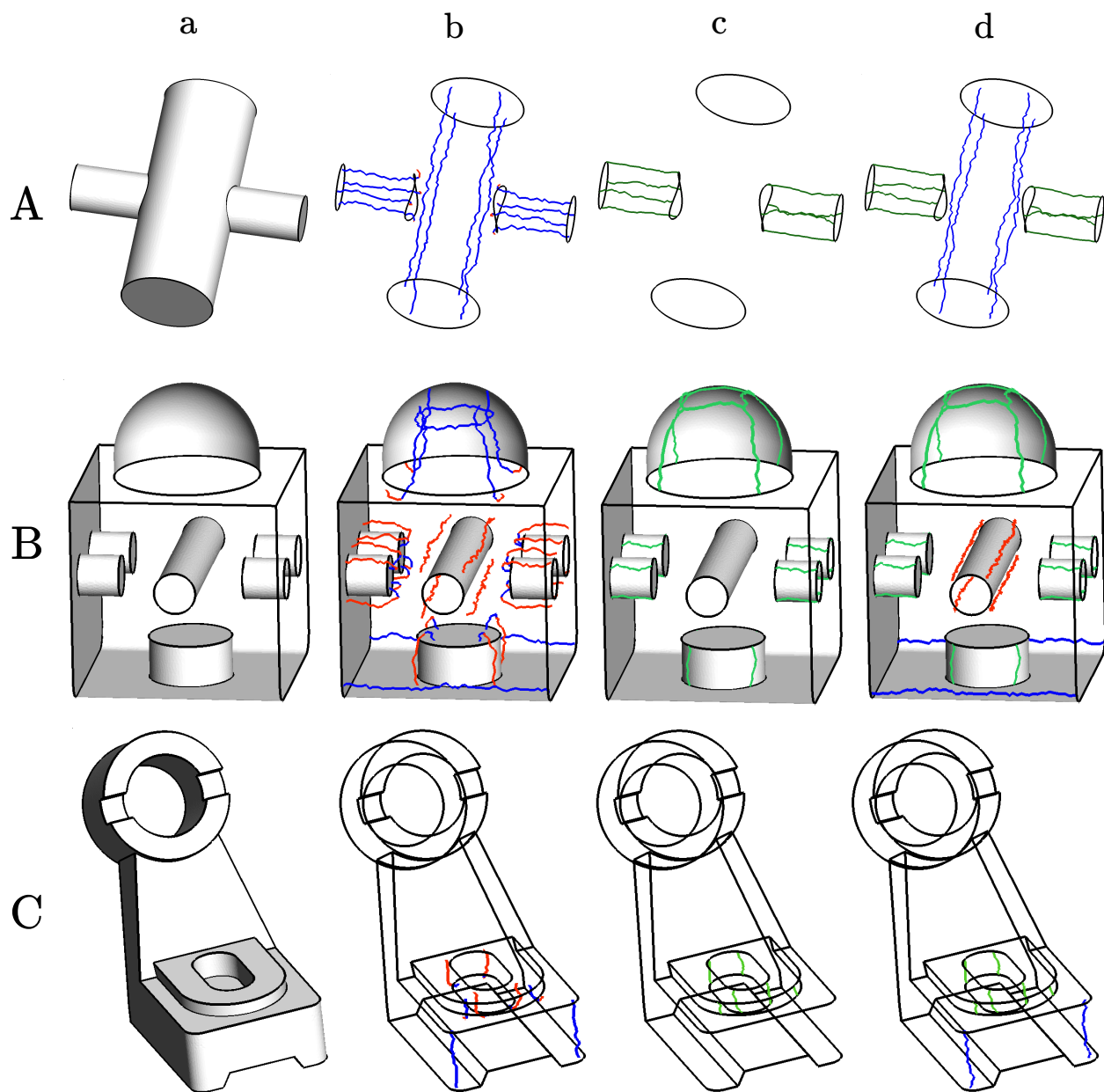
To avoid incompatible boundary conditions, we also remove the boundary alignment constraints on the vertices close to the snapped curves. These frames become free, allowing a smooth transition from the frames of the snapped curves to the frames on the rest of the model boundary.

While the new frame field is no longer boundary aligned everywhere, the affected areas remain localized and the resulting frame field is still similar to the initial one, minus the 3-5 singular curves that have been snapped.

**Applications** The singularity snapping correction is applied successfully on the *notch* model, see Figure 1.e., on the union of a cube and a half-sphere, see Figure 3.e. and on three more complicated CAD models, see Figure 6.

On the boxes with imprinted arcs (models A and B on Figure 2), the 3-5 singular curve would be reduced to a single node on the feature curve and the frame field would be as if there were no imprinted curves.

**Geometry and block refinement** One drawback of this approach is that it produces hexahedral blocks with an invalid geometry (zero jacobian at some corners), *e.g.* the bottom right block on Figure 4.b.. We propose to refine the affected blocks in a post-processing step, as shown on Figure 4.c.. For more complex cases, the post-processing refinement can follow a template-based strategy, such as [25]. To preserve the topology of a hexahedral mesh, the refinement must be propagated to adjacent blocks when they share a refined quadrangular face. This is equivalent to sheet insertion.



**Figure 6:** *Model A:* model built from the boolean union of two cylinders. *Model B:* model built from boolean operations between cylinders and a sphere, with two fillets on the bottom. *Model C:* model with various CAD features, from [24]. *Col. a:* initial models. *Col. b:* singularity graphs of the initial frame fields, which contain many 3-5 singular curves. *Col. c:* snapping of the 3-5 singular curves, the snapped curves are shown in dark green. *Col. d:* valid singularity graphs of the corrected frame fields, whose boundary conditions have been changed according to the snapped curves (in green).

**Limitations** This approach is only applicable when the 3-5 singular curves are close to the model boundary, as they are snapped on it. When dealing with CAD models, this is often the case because the invalid singularities are mostly caused by curved surface patches, *e.g.* from a boolean operation with a cylinder.

That being said, there are 3-5 singular curves that lie far inside the volume and whose cause is global, *e.g.* the non-meshable singularity graph of the *rockerarm* model shown in the Figure 10 of [3]. Our snapping technique does not handle such case.

## 5. CONCLUSIONS

To deal with non-meshable 3-5 singular curves, which are induced by CAD feature curves, we studied four heuristic-based frame field correction strategies.

The *feature curve extrusion* (§4.1) and the *boundary singular node extrusion* (§4.2) techniques try to change globally the frame field topology in the same way a human user would naively proceed. This approach does not work when there are interactions with singularities produced by other features of the model.

A more local and promising approach is to remove the feature curves that induce the 3-5 singular curves, by transforming them into fillets (§4.3). It allows additional singular curves that make the frame field singularity graph valid for hexahedral meshing. However, this approach is quite impractical because the mapping of the corrected frame field back to the initial geometry (without fillets) is not straightforward.

The last approach we explored is to remove the 3-5 singular curves by snapping them on the model boundary (§4.4). The resulting block decomposition associated to the corrected frame field is similar to the one obtained by using fillets, minus boundary layers. This technique, which is also local, is quite efficient and simpler to implement than the *fillet* correction. However, as the frame field is no longer aligned with the boundary everywhere, some blocks of the associated decomposition may have flat corners. Fortunately, these geometric defects can be removed via a post-processing block refinement procedure.

Short of having a better frame field formulation that do not produce 3-5 singular curves, we recommend using the *singular curve snapping* correction (§4.4), when applicable.

**Acknowledgement** This research is supported by the European Research Council (project HEX-TREME, ERC-2015-AdG-694020).

## References

- [1] Ray N., Sokolov D. “On smooth 3d frame field design.” *arXiv preprint arXiv:1507.03351*, 2015
- [2] Viertel R., Staten M.L., Ledoux F. “Analysis of Non-Meshable Automatically Generated Frame fields.” Tech. rep., Sandia National Laboratories (SNL-NM), Albuquerque, NM (United States), 2016
- [3] Liu H., Zhang P., Chien E., Solomon J., Bommès D. “Singularity-constrained Octahedral Fields for Hexahedral Meshing.” *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, 93:1–93:17, Jul. 2018
- [4] Baudouin T., Remacle J.F., Marchandise E., Henrotte F., Geuzaine C. “A frontal approach to hex-dominant mesh generation.” *Advanced Modeling and Simulation in Engineering Sciences*, vol. 1, no. 1, 8, 2014
- [5] Bernard P.E., Remacle J.F., Kowalski N., Geuzaine C. “Frame field smoothness-based approach for hex-dominant meshing.” *Computer-Aided Design*, vol. 72, 78–86, Mar. 2016
- [6] Gao X., Jakob W., Tarini M., Panozzo D. “Robust hex-dominant mesh generation using field-guided polyhedral agglomeration.” *Transactions on Graphics (TOG)*, vol. 36, no. 4, 1–13, Jul. 2017
- [7] Sokolov D., Ray N., Untereiner L., Lévy B. “Hexahedral-Dominant Meshing.” *Transactions on Graphics (TOG)*, vol. 35, no. 5, 1–23, Jun. 2016
- [8] Ray N., Sokolov D., Reberol M., Ledoux F., Lévy B. “Hex-dominant meshing: mind the gap!” *Computer-Aided Design*, vol. 102, 94–103, 2018
- [9] Nieser M., Reitebuch U., Polthier K. “CubeCover- Parameterization of 3D Volumes.” *Computer Graphics Forum*, vol. 30, no. 5, 1397–1406, Aug. 2011
- [10] Li Y., Liu Y., Xu W., Wang W., Guo B. “All-hex meshing using singularity-restricted field.” *Transactions on Graphics (TOG)*, vol. 31, no. 6, 1, Nov. 2012
- [11] Lyon M., Bommès D., Kobbelt L. “HexEx: robust hexahedral mesh extraction.” *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, 123, 2016
- [12] Zheng Z., Wang R., Gao S., Liao Y., Ding M. “Dual Surface Based Approach to Block Decomposition of Solid Models.” 2018

- [13] Livesu M., Pietroni N., Puppo E., Sheffer A., Cignoni P. “Loopy Cuts: Surface-Field Aware Block Decomposition for Hex-Meshing.” *arXiv preprint arXiv:1903.10754*, 2019
- [14] Palacios J., Zhang E. “Rotational symmetry field design on surfaces.” *Transactions on Graphics (TOG)*, vol. 26, no. 99, 55, Jul. 2007
- [15] Knöppel F., Crane K., Pinkall U., Schröder P. “Globally optimal direction fields.” *ACM Transactions on Graphics (ToG)*, vol. 32, no. 4, 59, 2013
- [16] Beaufort P.A., Lambrechts J., Henrotte F., Geuzaine C., Remacle J.F. “Computing cross fields A PDE approach based on the Ginzburg-Landau theory.” *Procedia engineering*, vol. 203, 219–231, 2017
- [17] Viertel R., Osting B. “An Approach to Quad Meshing Based on Harmonic Cross-Valued Maps and the Ginzburg–Landau Theory.” *SIAM Journal on Scientific Computing*, vol. 41, no. 1, A452–A479, 2019
- [18] Huang J., Tong Y., Wei H., Bao H. “Boundary aligned smooth 3D cross-frame field.” *Proceedings of the 2011 SIGGRAPH Asia Conference on - SA '11*. ACM Press, 2011
- [19] Chemin A., Henrotte F., Remacle J.F., Van Schaftingen J. “Representing three-dimensional cross fields using 4th order tensors.” 2018
- [20] Ray N., Sokolov D., Lévy B. “Practical 3D frame field generation.” *Transactions on Graphics (TOG)*, vol. 35, no. 6, 1–9, Nov. 2016
- [21] Corman E., Crane K. “Symmetric Moving Frames.” *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, 87:1–87:16, Jul. 2019
- [22] Jiang T., Huang J., Wang Y., Tong Y., Bao H. “Frame field singularity correction for automatic hexahedralization.” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 8, 1189–1199, 2014
- [23] Solomon J., Vaxman A., Bommers D. “Boundary element octahedral fields in volumes.” *ACM Transactions on Graphics (TOG)*, vol. 36, no. 3, 28, 2017
- [24] Ledoux F. “Paving the Path Towards Automatic Hexahedral Mesh Generation.” invited talk at the 27th International Meshing Roundtable, 2018
- [25] Schneiders R. “Refining Quadrilateral and Hexahedral Element Meshes.” *5th International Conference on Grid Generation in Computational Field Simulations*, pp. 679–688. CRC Press, 1996