Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 00 (2016) 000–000

SAND2016-7754C

Procedia Engineering

www.elsevier.com/locate/procedia

25th International Meshing Roundtable

Analysis of Non-Meshable Automatically Generated Frame fields

Ryan Viertel[a,b], Matt Staten[a], Franck Ledoux[c]

[a] *Sandia National Laboratories, 1515 Eubank SE , Albuquerque, NM 87123, USA*
[b] *University of Utah, 201 Presidents Cir, Salt Lake City, UT 84112, USA*
[c] *Commissariat Energie Atomique C.E.A, Chemin du Ru, 91680 Bruyères-le-Châtel, France*

Abstract

Recent methods for frame field generation in two and three dimensions are reviewed. Frame fields generated automatically in 2D and 3D are analyzed with respect to quad and hex mesh generation. Problems are identified with automatically generated frame fields that prevent mesh generation via current methods. Specifically, there exist geometries that contain limit cycles and cannot be parameterized or decomposed by separatrices of the frame field. In 3D, singularity lines occur that minimize the field curvature but do not align with the frame field. These types of singularities make it impossible to create a mesh that both follows the frame field, and simultaneously respects the singularity as an irregular node in the mesh. Specific examples are presented that illustrate these problems. For each example, streamlines are used to help visualize properties of the frame fields, problems are analyzed, and options to potentially mitigate such problems are discussed.

Peer-review under responsibility of the organizing committee of IMR 25.

Keywords: frame field; hex mesh;

1. Introduction

The study of frame fields is a new and promising area of research in the quest to automate hex mesh generation. Their use has led to robust and efficient algorithms for generating high quality quad meshes in 2D that satisfy many desired properties [1]. A major goal in the research of frame fields now is to extend these automatic mesh generation methods to 3D. Significant progress has been made on various fronts ([2–5]), and hex meshes have automatically been generated on several geometries. Despite this progress there are still some problems with automatically generated frame fields that cannot be addressed by current methods. These problems need to be solved in order to reach the goal of robust automatic hex mesh generation. In this note, specific examples of problems that need to be addressed are presented along with a discussion of possible options to mitigate these issues.

* Corresponding author. Tel.: +1-801-669-4887.
E-mail address: rvierte@sandia.gov

1877-7058
Peer-review under responsibility of the organizing committee of IMR 25.

2. Background

A frame defines a preferred orientation to place a hex or quad element in a mesh of a given geometry. Since most of our discussion will be in 3D, frames will be defined as follows:

Definition 2.1. A *frame* is a set of right handed coordinates (U, V, W) where U, V, and W are all unit vectors. Two frames s and t are equivalent if there is a rotation corresponding to the cubical symmetry group $\mathcal{G}$ that transforms s to t. A *discrete frame field* over a mesh is a frame defined at each node of the mesh. Frame values within regions of the mesh are given by a linear interpolation of the frames defined at the nodes incident to the region.

The core idea behind automatic frame field generation is to smoothly propagate the preferred orientation that is naturally defined on the boundary onto the interior of a domain. For example, in 2D the natural orientation of a frame can be defined at any point of a C^1 continuous boundary curve by the pair of unitary normal and tangent vectors at that point. In 3D, frames are fully constrained on geometric curves and vertices, and constrained such that one of the coordinate vectors points in the normal direction for each frame defined on a boundary surface. In each automatic frame field generation method [1,2,5], these constraints are enforced and a field energy which corresponds to the difference between orientation of frames is minimized.

Minimizing the field energy leads to a smooth transition of frames over most of the geometry, however in order to satisfy all of the boundary constraints singularities appear in the frame field. Conceptually these singularities correspond to simply connected edge sets in a hex mesh with irregular valence. Mathematically, these singularities occur where the rotational transition between frames is not the identity.

Definition 2.2. A *matching*, Π_{st} between two frames s and t is the best transformation between them in the following sense:

$$\Pi_{st} := \operatorname*{argmin}_{P \in \mathcal{G}} \|[U_s|V_s|W_s] - [U_t|V_t|W_t]P^T\|_F$$

where $P \in \mathcal{G}$ is a mapping from the cubical symmetry group and $\|\cdot\|_F$ is the Frobenius norm.

Definition 2.3. Given a tet mesh over a geometric domain and a discrete frame field with frames defined at each node of the mesh, the *type* of a tet face, f, is a is given by

$$\text{type}(f, n_0) = \Pi_{n_2 n_0} \circ \Pi_{n_1 n_2} \circ \Pi_{n_0 n_1}$$

Where n_i are the frames at the nodes incident to f. If the type of a face is not the identity mapping, then the face is said to be *singular*.

We will use streamlines to visualize some of the issues with automatically generated frame fields.

Definition 2.4. Given a continuous frame field defined over a domain Ω, let $\gamma(s)$ be the parameterization of a C^1 continuous curve not defined at any singularity. Then γ is a streamline of the frame field if

$$\forall s \in [0, 1], \partial\gamma(s) \times x(\gamma(s)) = 0$$

where $x \in (U, V, W)$, the coordinate vectors of the frame defined at $\gamma(s)$.

2.1. Previous work

Various methods have been proposed to extract quad meshes from frame fields in 2D [1,6]. Of note is the 2013 paper by Kowalski et. al, in which the authors proposed a method to automatically generate frame fields on 2D domains and then extract a mesh by first decomposing the domain into 4 sided regions using the singularity graph of the frame field. Others have taken the more direct approach of parameterizing the domain with the help of a guiding frame field [6]. Automatic quad mesh generation techniques in 2D using frame fields are robust and can mesh a large set of geometries.

The extension of these methods to 3D has proven to be difficult, but significant progress has been made in recent years. In 2011 Nieser et. al. [7] described the CubeCover method (an extension of [6]). This work describes

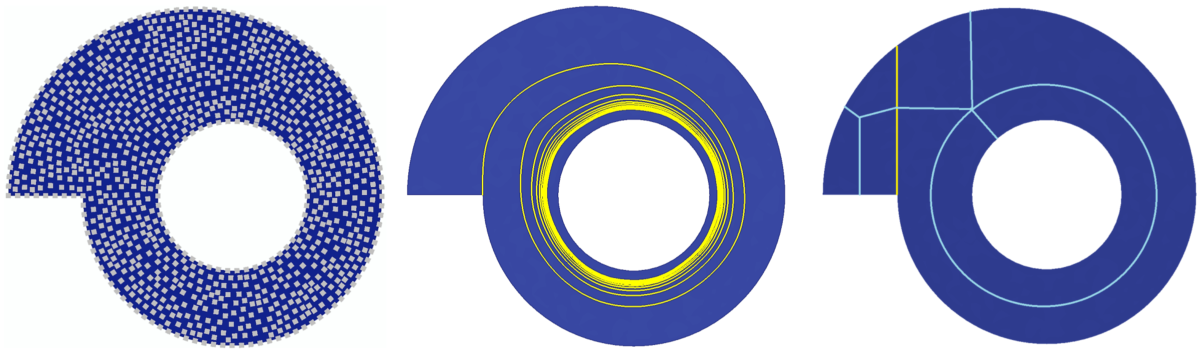

Fig. 1. (a) Nautilus with frame field. Each box represents the preferred orientation of a quad element at that point. (b) A streamline originating from the concave corner. This illustrates the limit cycle of the field. (c) A block decomposition of the geometry that results from the automatically generated frame field after the geometry is cut from the concave corner directly up (yellow line).

a method to parameterize geometries to generate a hex mesh and laid a foundational framework on frame fields in three dimensions. In particular this work established the necessary topological condition that in order to get a valid parameterization of a geometry, singularities of the frame field must have a type that is a rotation about a single coordinate axis. Singularities with types that are rotations about more than one axis are called compound singularities. At the time that the CubeCover method was developed, there were no methods for automatically generating frame fields and the algorithm required that the user manually define a frame field on the geometry. Since then, Huang et. al. [2] and Ray et. al. [5] have developed methods to automatically generate frame fields in 3D. Jiang et. al. [4] and Li et. al. [3] then combined automatic frame field generation with the CubeCover method to automate hex mesh generation. In order to use automatically generated frame fields to parameterize a geometric domain, both authors found it is necessary to perform operations to validate the types of singularities occurring in the geometry, either by removing compound singularities or fixing issues due to the discretization of the field. After validating the singularity graph both authors were successful at automatically generating high quality hex meshes on various geometries. Despite these successes, there are still examples of automatically generated frame fields that cannot be used to parameterize a domain. These problems can occur even when each singularity in the geometry has a valid type.

3. Problematic examples

Even if a frame field is boundary aligned and the type of every singularity is a rotation about a single coordinate axis, automatic methods [1,2,5] for frame field generation lead to examples where the frame field does not correspond to a valid mesh of the geometry. The examples discussed in this research note can be classified as one of two types. Either the geometry induces a limit cycle in the frame field, or singularities do not align with the frame field.

Example 3.1. The nautilus shown in fig. 1 illustrates the first potential problem with automatically generated frame fields. In this 2D example, the frames are aligned with the boundary, and the energy minimization has led to a smooth frame field with no singularities (fig. 1(a)), however no proper parameterization of the domain can be obtained because a limit cycle in the frame field leads to a degeneracy. This is visualized using streamlines of the frame field in fig. 1(b).

In a parameterization of the nautilus using this frame field, grid lines in the angular direction would all converge onto the same curve, leading to an infinite number of elements with arbitrarily small volume. Further, the block decomposition method proposed in [1] would not work since the streamline originating from the concavity also would approach this limit cycle.

One way to mitigate this problem is to introduce an additional constraint to the geometry. Rather than simply constraining the frame field to align with the boundary, we can "cut" the geometry and require that the frames align to the boundary of the cut as well. Cutting the nautilus from the concavity directly up results in a frame field that induces a block decomposition equivalent to fig. 1(c). Such a process of cutting geometries where limit cycles occur

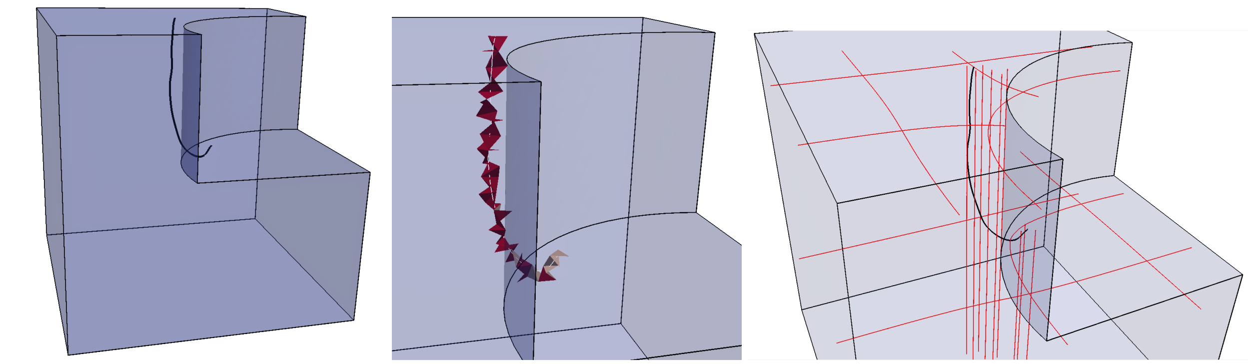

Fig. 2. (a) The "notch" - a cube with a quarter cylinder cut out of the corner. The thick black curve represents the singularity graph of the automatically generated frame field. (b) A close up of the singularity with type classification. Tan faces represent faces where the singularity type is a positive rotation with respect to the local coordinates. Red represents a negative rotation. Positive rotations correspond to 3 valent singularities in the hex mesh while negative rotations correspond to 5 valent singularities. (c) Streamlines (in red) show the directionality of the frame field. You can see that the singularity shown here is not aligned to the frame field.

could potentially be left to the user, however for robust automatic mesh generation more research needs to be done to automatically determine how to make cuts that will change the problematic topology.

Example 3.2. Fig. 2 shows an example of another simple geometry, a cube with a quarter cylinder cut out of the corner, where a frame field generated by a state of the art method cannot be used to generate a hex mesh. This example is noted in [5], but we analyze it in more detail here. The white line in fig. 2(a) is a bezier curve fit to the set of points where the singularities occur on each tetrahedral face. This curve gives us a representation of the singularity graph of the geometry. The frames on the top surface of the geometry induce a 5 valent singularity, while at frames on the lower surface induce a 3 valent singularity. In a hex mesh, it is clearly not possible for a singularity to change valence. In fact, it is mathematically impossible in a hex mesh for 3 and 5 valent singularities to meet at a junction without other singularity lines also meeting there [7,8]. Indeed, by classifying each tetrahedral face as a 3 or a 5, depending on whether the singularity type is a rotation in the positive or negative direction about the local coordinate frame, we can see that this singularity graph does not yield a clear classification of the singularity (fig. 2(b)). That is, the frame field in this example is a singularity restricted field as defined in [3] - every singular face has a valid type - however the singularity graph is still not valid since the direction of rotation with respect to the local coordinates changes from face to face.

Another key observation about this frame field is that the singularity graph is not aligned to the direction of the frame field. Singularity alignment with the frame field is a necessary condition for generating a hex mesh from a frame field that is not explicitly mentioned in any previous works. In a hex mesh, singularity lines are by necessity aligned with the mesh because they are made up of edges of the mesh. Thus in order to generate a hex mesh that follows the directions of the frame field, whose singularity graph corresponds to singularities in the hex mesh, it is also necessary that the singularity graph of the frame field be aligned to the frames. In fig. 2(c) streamlines are used to visualize the direction of the frame field. In this particular case all frames are aligned with the z-axis, and it is clear that the singularity does not stay in line with the frames.

Intuitively, the singularity graph of the notch would be two separate lines, a 5 valent singularity propagating from the top surface to the bottom, and a 3 valent singularity propagating from the lower surface to the bottom (fig. 3(a)). This does not occur with the automatically generated frame field because the geometric constraint that induces the two singularities (i.e. the curved surface from the cylinder cutout) only exists at the top of the geometry. The field energy is minimized by the two singularities coalescing. One way to mitigate this problem is to blend the curve along the concavity where this geometric constraint disappears. Doing so changes the topology of the geometry and yields a different singularity graph which corresponds to a valid hex mesh (fig. 3(b)). Blending curves is a local operation which can get rid of some problematic singularities. This is an ideal solution when viable, however design specific considerations may require sharp corners. In such situations, it may be possible to devise a strategy for propagating

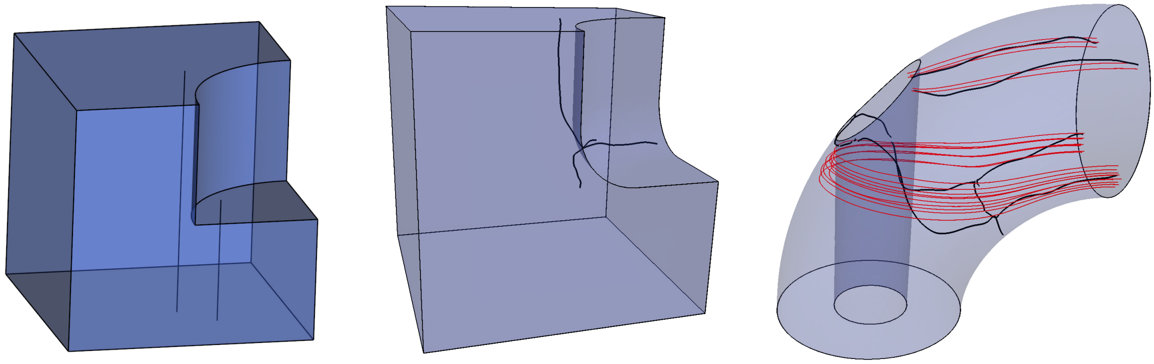

Fig. 3. (a) The singularity graph of the intuitive mesh on the "notch" geometry. (b) Blending the curve along the concavity changes the topology and results in a valid singularity graph that corresponds to a hex mesh. (c) A geometry with singularities that do not occur near a concavity and are still not in line with the frame field. Note that the singularities at the top of the geometry are inline with the streamlines, but the singularities at the bottom begin in line with the streamlines and then diverge.

geometric constraints that end at a concavity throughout the geometry by following the frame field. This however is a global problem, and it is not clear how to handle situations where propagating multiple constraints leads to an intersection of those constraints, or how to propagate along the frame field when the directions diverge because of a singularity. Furthermore, this problem is not specific to constraints that end in concavities. Singularities that are not aligned with the frame field can occur in various situations (fig. 3(c)).

Another potential strategy we tried is to add a constraint to the minimization problem when solving for the frame field. Such a constraint could appear in the objective function as a penalty when singularities are not aligned with the frame field. We attempted to do this in an a. posteriori fashion as follows:

1. Compute the frame field using the method described in [5].
2. Find the singularity line, and assign each node attached to tet faces that the singularity line passes through a desired direction by projecting the node to the singularity line and calculating the tangent at that point.
3. During the field smoothing step add in the new constraints for the nodes near a singularity. This will potentially introduce new singularities into the field. The hope being that these new singularities change the topology of the frame field to a valid one, since all frames should be in line with singularities.
4. Iterate steps 2-3 for a fixed number of steps or until convergence to a frame field where frames are aligned to the singularities.

When constraining the nodes near singularities in step 3 exactly in the direction of the singularity line, this process produces new singularities at each step and the frame field becomes chaotic. If instead we constrain the frames to a direction between their previous orientation and the direction of the singularity, we see essentially two behaviors. If the frame moves significantly to the direction of the singularity line, we see the same chaotic behavior as before. If the frame stays mostly in the orientation that it previously was, then the new constraints do little to change the frame field and the process appears to be almost identical to the smoothing step without the new constraint. Thus if the constraint is too strong, the process becomes chaotic. If it is too weak, it can't change the topology of the singularity graph, and our results are no different than if we didn't include the constraint. This approach seems ineffective and singularity alignment to the frame field remains an open problem.

4. Conclusions

Frame fields are a promising new area of research because of their potential to simplify the current process of producing a high quality hex mesh. Significant progress has been made in recent years. Methods have been developed to automatically generate smooth frame fields with minimal singularities, and to parameterize and hex mesh volumes whose frame fields have valid singularity graphs. In order for these methods to work robustly for a large and industrial geometries, issues such as those presented in this research note need to be addressed. More research needs to be done to classify singularity graphs and frame fields that can be addressed by current methods, and to find conditions or methods that will produce frame fields with the desired properties.

Acknowledgments

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energys National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND2016-XXXX C.

References

[1] N. Kowalski, F. Ledoux, P. Frey, A PDE Based Approach to Multidomain Partitioning and Quadrilateral Meshing, in: X. Jiao, J.-C. Weill (Eds.), Proceedings of the 21st International Meshing Roundtable, Springer Berlin Heidelberg, 2013, pp. 137–154. URL: https://www.ljll.math.upmc.fr/frey/publications/imr21.pdf, dOI: 10.1007/978-3-642-33573-0_9.
[2] J. Huang, Y. Tong, H. Wei, H. Bao, Boundary Aligned Smooth 3d Cross-frame Field, in: Proceedings of the 2011 SIGGRAPH Asia Conference, SA '11, ACM, New York, NY, USA, 2011, pp. 143:1–143:8. URL: http://doi.acm.org/10.1145/2024156.2024177. doi:10.1145/2024156.2024177.
[3] Y. Li, Y. Liu, W. Xu, W. Wang, B. Guo, All-hex Meshing Using Singularity-restricted Field, ACM Trans. Graph. 31 (2012) 177:1–177:11.
[4] T. Jiang, J. Huang, Y. Wang, Y. Tong, H. Bao, Frame Field Singularity Correction for Automatic Hexahedralization, IEEE Transactions on Visualization and Computer Graphics 20 (2014) 1189–1199.
[5] N. Ray, D. Sokolov, On Smooth 3d Frame Field Design, arXiv:1507.03351 [cs] (2015). ArXiv: 1507.03351.
[6] F. Klberer, M. Nieser, K. Polthier, QuadCover - Surface Parameterization using Branched Coverings, Computer Graphics Forum 26 (2007) 375–384.
[7] M. Nieser, U. Reitebuch, K. Polthier, CubeCover Parameterization of 3d Volumes, Computer Graphics Forum 30 (2011) 1397–1406.
[8] R. Bowen, S. Fisk, Generations of triangulations of the sphere, Mathematics of Computation 21 (1967) 250–252.