Hybrid Viscous Layer Insertion in a Tetrahedral Mesh

Guillaume Vialaneix[1,2], Yvan Fournier[1], and Tamy Boubekeur[2,3]

[1] EDF R&D
[2] Telecom ParisTech
[3] CNRS - LTCI

Abstract. Computational Fluid Dynamics (CFD) methods for viscous computations often require high quality computational meshes. The viscous layer area needs to be meshed at a high resolution, leading to highly anisotropic meshes. The methodology developed in this article addresses this issue, enabling the creation of a viscous layer in a tetrahedral (or hybrid polyhedral) mesh, so that the inner meshing is preserved while the viscous layer area is correctly meshed. It uses a surfacic clustering process, followed by the combination of the deletion of a layer of cells and the generation of new ones through the projection of the remaining surface faces onto the original surface.

1 Introduction

The highly anisotropic nature of meshes adapted to viscous-layers studies led to various approaches of hybrid meshes, using tetrahedra, prisms, and sometimes hexahedra. We can first cite the advancing layer techniques, such as proposed by Kallinderis et al. [Kal95, Kha95], Sharov and Nakahashi [Sha98], or Dyedov [DEJ$^+$09]. The process is based upon a triangulated surface, which is advancing toward the interior of the volume using prisms. Once the viscous layer area is meshed in such a way, the rest of the volume is meshed using classic isotropic triangulation techniques.

Ito et al. [IN02] proposed a technique to insert a prismatic layer into a tetrahedral mesh, using pyramids for conformity. The main drawback of all those techniques is illustrated in fig. 1: the offset, as defined above, is quite high near the feature angle of the mesh. Karman [Kar07] proposed a method to solve this problem, by inserting hexahedra at features. However, they use pillowing (contraction of the mesh) in order to create their layer, and by so, the whole mesh is modified. The approach of Yamakawa and Shimada [Yam11], creating full-hexahedra meshes, also end with poor quality regarding the offset criterion.

We use here three quality criteria that we tried to optimize with our algorithm. Considering two cells of a mesh, A and B their centers, I the intersection of the $[AB]$ segment and the common face between the cells, which center is F, and $\mathbf{n}$ its normal. M is the middle of $[AB]$. We define three quality criteria of the interface between the cells, defined in table 1.

Table 1. Definition of quality criteria

Criterion	Definition	Optimal value	Corresponding error
offset	$\|\mathbf{IF}\|$	0	tangent to the face
weighting	$\frac{min(\|\mathbf{AI}\|, \|\mathbf{BI}\|}{\|\mathbf{AB}\|}$	$\frac{1}{2}$	normal to the face
non-orthogonality	$(\mathbf{AB}, \mathbf{n})$	0	angular, on the face

2 Our Approach

Our method is composed of three main phases : first, we detect the different surfacic clusters of the mesh, then we delete a layer of cells, faces and vertices, and finally we create a layer of prisms and hexahedra to fill the empty space. The first step will help to determine where to insert hexahedra in addition to prisms in order to obtain a good-quality mesh. In the deletion step, we remove one layer of cells, and all the faces and vertices whose only contact are with those cells. We conclude by presenting the projection of the remaining faces and vertices onto the surface of the mesh, creating prisms and hexahedra.

2.1 Cluster Surface

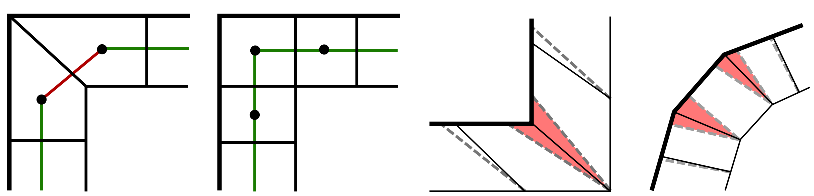

Fig. 1. From left to right: the main drawback of existing techniques (the offset value of the red edge), our solution, a concave corner and some *smooth* edges, areas where hexahedra (dotted lines) are not to be used (light red), and one must prefer prisms (solid lines).

The figure 1 illustrates the compromise we try to achieve through our process: the necessity to create hexahedral cells on sharp features, but they are only needed on sharp features, or the output quality will not be good either. In order to achieve this, we want to group surface faces into clusters, each one representing a geometrical entity of the mesh surface. Hexahedra will only be inserted at the junction between two clusters.

In the initialization step, each face defines its own cluster. Then, we iteratively consider every pair of faces that have different cluster numbers, and if the angle between their normals is inferior to an user-defined threshold t_α, we merge the two clusters. The algorithm goes on until convergence. This algorithm has the advantage of being fully controlled by only one parameter : the merging threshold t_α, set to 25°as a rule of thumb. In order to strengthened our process, we apply a normal enhancement filtering to the mesh normals: the final normal $\mathbf{n}_f$ is given by $\mathbf{n}_f = \mathbf{n} - \lambda \times \mathbf{n}_s$, with $0 \leq \lambda \leq 1$, $\mathbf{n}$ the initial value, $\mathbf{n}_s$ a smoothed version.

2.2 Projections

We then erase a full layer of boundary cells, and by doing so, end up with a *subsurface*, the surface of the resulting mesh, after the deletion step. We need now to project this subsurface onto the original surface of the mesh. We need to determine on how many clusters we have to project each subsurface vertex, and how many of them are forming a concave zone, in order to determine how many projection each vertex will have. Each vertex of the subsurface will inherit the clusters numbers of its former surfacic neighbors.

Let us first consider the case of the surface faces forming only one cluster (a sphere, for example). Then, each vertex of the subsurface only needs to be projected once on the surface, and from the triangles on the subsurface, prisms are extruded to the surface. In order to achieve a good matching of the surface, we use here the Hermite Point-Set Surface algorithm [AA09]. We search for the three nearest-neighbors on the surface that lie on the same cluster, and we then are able to project the subsurface point onto the surface.

In order to ensure the quality of the output mesh according to the criteria defined above, and more specifically the offset, we need to create hexahedra on the sharp features of the mesh. Every time that a vertex needs to be projected on two different clusters, which do not form a concavity, we impose a third projection on the intersection of those clusters. Similarly, we will add four projections to a vertex that already needs to be projected on three clusters, as depicted on fig.2. If some clusters are forming a concavity, we simply compute one projection, on the intersection of the clusters.

Those projections can generate a geometry intersecting itself, specifically around concavities. We introduce here a relaxation step of the projecting vectors, then of the resulting vertices, followed by another step of projection on the surface.

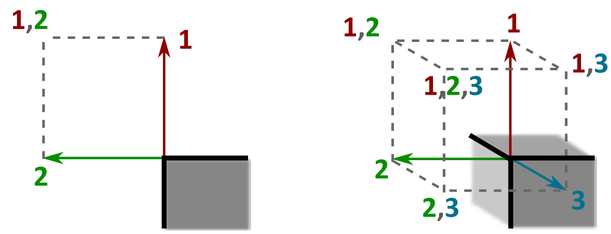

Fig. 2. Left : template for the projection of one point on two different clusters. Right : projection of one point on three clusters.

2.3 Viscous Layer

Now that the outer layer of the mesh is made of prisms and hexahedra, it is quite straightforward to create a viscous layer (made of several layers of cells) in the desired areas of the mesh. To each cluster can be associated an integer n that tells us how many cells have to be created between the subsurface and the surface, and a real factor α, that encodes the variation of the cells size toward the surface. For each edge linking a surface vertex p_s to a subsurface one p_{ss}, we insert $n-1$ points $p_1, \ldots, p_{n-1}$ such that $p_k = \lambda_k p_{ss} + (1-\lambda_k)p_s$, with $\lambda_k = 1/\alpha^k$.

3 Results and Discussion

The whole process described in this paper was based on the free CFD software *Code_Saturne* (EDF R&D). Figure 3 presents an example of the meshes we were able to generate. The choice of the α factor in the viscous layer creation enable the control over the weighting quality criteria, while the insertion of hexahedron in the middle of prisms at sharp features maintain a low offset in the mesh. The very geometric nature of prisms and hexahedron ensures that the third criterion, non-orthogonality stays at low levels. On a cubic mesh (no concavities, only planes or sharp edges), of about 15k cells, we observe a significant improvement of our criteria, as maximum values dropped off: $9,02$ to $0,71$ for the offset, $9,04$ to $0,86$ for the weighting, and the non-orthogonality went from $88,8°$to $60,0°$.

Our method enables us to create good-quality meshes according to those three criteria, while preserving the whole inner meshing. However, there is still work to be done on the robustness of the process, mainly due to the irregularity of the subsurface. Future work may also include the parallelization of the process, in a distributed way.

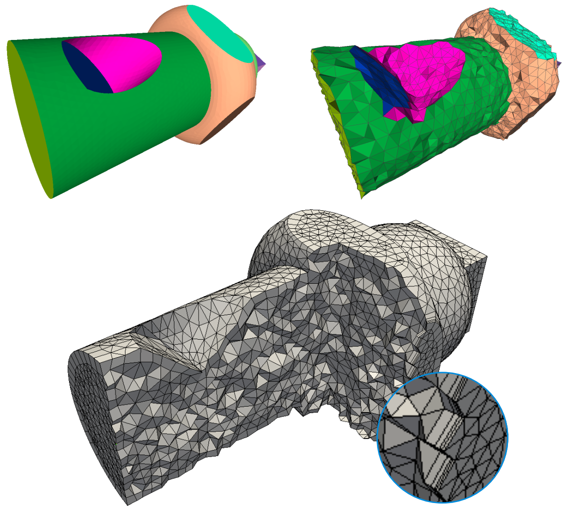

Fig. 3. The main steps of our method. Top left: surface faces clustering. Top right: subsurface faces. Bottom: final result, only one cluster meshed in viscous layer.

References

[AA09] Marc Alexa and Anders Adamson. Interpolatory point set surfacesconvexity and hermite data. *ACM Trans. Graph.*, 28(2):20:1–20:10, May 2009.

[DEJ+09] Volodymyr Dyedov, Daniel Einstein, Xiangmin Jiao, Andrew Kuprat, James Carson, and Facundo del Pin. Variational generation of prismatic boundary-layer meshes for biomedical computing. *International Journal for Numerical Methods in Engineering*, 79(8):907–945, 2009.

[IN02] Yasushi Ito and Kazuhiro Nakahashi. Unstructured mesh generation for viscous flow computations. In *Proceedings of the 11th International Meshing Roundtable*, pages 367–377, 2002.

[Kal95] Kallinderis, Y. and Khawaja, A. and McMorris, H. Hybrid Prismatic/Tetrahedral Grid Generation for Complex 3-D Geometries. *AIAA-95-0211*, 1995.

[Kar07] S. L. Karman. Unstructured viscous layer insertion using linear-elastic smoothing. *AIAA Journal*, 45(1):168–190, 2007.

[Kha95] Khawaja, A. and McMorris, H. and Kallinderis, Y. Hybrid Grids for Viscous Flows Around Complex 3-D Geometries Including Multiple Bodies. *AIAA-95-1685*, 1995.

[Sha98] Sharov, D. and Nakahashi, K. Hybrid Prismatic/Tetrahedral Grid generation for Viscous Flow Applications. *AIAA Journal*, 36(2):157–162, 1998.

[Yam11] Yamakawa, Soji and Shimada, Kenji. Automatic All-Hex Mesh Generation of Thin-Walled Solids via a Conformal Pyramid-Less Hex, Prisms and Tet Mixed Mesh. *20th International Meshing Roundtable*, 2011.