Available online at www.sciencedirect.com

ScienceDirect

Procedia Engineering 163 (2016) 5 – 19

Procedia Engineering

www.elsevier.com/locate/procedia

25th International Meshing Roundtable (IMR25)

All hexahedral boundary layers generation

Loïc Maréchal[a,*]

[a]INRIA, 1 rue Honoré d'Estienne d'Orves, 91120 Palaiseau, France

Abstract

Boundary layers generation has recently become one of the most sought-after features in mesh generation. But behind this single feature hides a lot of different capacities that range from the most basic, like extruding a layer from a surface, to imprinting, cross-imprinting and multi-normals capabilities that are much more challenging to develop. This paper aims at implementing all those features using hexahedral elements only and is based on a previously developed automated hexahedral mesh generator based on the octree method. Both points, octree and hexahedra, implying some constraints that this paper will address in a very pragmatic way. Indeed, we view automatic meshing from the industrial user's point of view: *no mesh, no run*, consequently this work's philosophy has been always to favor automaticity and genericity over element's quality and geometric approximation.

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

Keywords: octree, hexahedral mesh, boundary layers, imprinting, cross-imprinting

1. Boundary-layers: what it is all about

Generating boundary layers (BL hereafter) consists in growing layers of stretched elements from a surface mesh.

In the early years of mesh generation, and up to a time not so distant, that was all BL where about. But when this simple feature became widely available and used in numerical simulations, a truckload of requests came from the users like multi-normals, normal-blending, imprinting, cross-imprinting, blending layers, and so on.

This section aims at listing and shortly describing all the desired features that should be implemented in an industrial grade automated mesher.

Source control. We want to grow layers from some specific surface references or patches and may want to specify a different set of layer parameters on each of those references or groups of references.

Layers parameters. For each source reference we would like to specify the number of layers, their size and geometrical growth factor.

Ending mode. Layers may not cover the whole surface and may stop at some surface patch limits. We would like to specify whether the layers should vanish with wedge shaped elements or should *imprints* the surface it stops at. This

* Corresponding author. Tel.: +33-177-578-005
E-mail address: loic.marechal@inria.fr

Peer-review under responsibility of the organizing committee of IMR 25
doi:10.1016/j.proeng.2016.11.007

parameter could be set by the user or automatically derived from the dihedral angle between the two surface quads that belong to the different domains. In case of wide angle, a wedge ending is preferred and when the angle is sharper than 90 degrees, imprinting is preferred, in both cases for the sake of elements quality.

Cross-imprinting. Two sets of layers growing from neighbor surface patches, may fuse or intersect each other, creating a *cross-imprinting* pattern.

Normal blending. Layers should grow orthogonally to the surface, but in some concavities, it may create distorted or invalid elements. This problem may be circumvented by slightly bending the normal vectors, skewing the elements, to preserve their quality.

Blending layers. The last layer of a set is connected directly to the volume mesh and may be so thin that a strong volume shock could happen, leading to imprecise numerical results. A set of extra layers, whose thickness should grow from the final *physical* layer up to the background element size, may be added to mitigate the size transition. Hence the name *blending* layers.

Multi-normals. When layers have to be grown from a very sharp convex angle, they tend to generate very poorly shaped elements. A nice way to deal with that is to round-up the sharp angle to preserve the mesh quality.

In this paper we will try to address each point as best as we can given the strong constraints coming from the octree structure, which suffers from its inherent isotropy and axis alignment, and the hexahedral elements that are much more rigid, topologically and geometrically than tetrahedra.

We know beforehand that we will not be able to reach the tetrahedral BL capacities that are much higher, thanks to the nimble nature of simplicial elements over hexahedra. The greatest capacities that are difficult, if not impossible, to implement with octree and hexahedra, are anisotropic meshing as is seen in [12], the ability to push the layers very far away from the boundary as seen in [9] and far stretching cross-imprinting as done in [11].

However, some interesting work has been done on hexahedral boundary layers generation using sheet insertion [7], smoothing and extrusion [10] and even non conformal meshes based on octree [8].

The philosophy behind this work is always to favor the automatic and generic aspects of the algorithm since it is intended to be used as a *black box* software component. So whatever the input surface is, and whatever the BL parameters the user requests, the method must always produce a valid result, be it different from what the user is expecting in some complex cases.

In other words, the constant parts of the equation are the automaticity and genericity and the variables are the elements quality and the geometric accuracy.

2. Octree based hexahedral meshing: a quick overview

The present work is based on *Hexotic*, an automated hexahedral-mesh generator based on the octree method that was presented at the IMR 2009 conference. Since the objective of this work is to focus on BL generation only, the reader is encouraged to read this previous paper for a deeper overview. For the sake of understanding, the mesher's basic principles are shortly described in this section. The process can be broken-down into the following steps:

1. refine an octree structure according to the input surface meshes local feature sizes,
2. derive a dual mesh from the octree,
3. color volume subdomains separated by boundary faces,
4. insert a buffer layer of elements near the surface,
5. project boundary nodes on a smoothed surface while displacing the other inner nodes for the sake of quality,
6. insert a buffer layer of elements near the sharp angles,
7. project the boundary nodes on the real surface with sharp features while smoothing the other nodes.

Such algorithm suffers from many drawbacks: it is isotropic, thus generating too many elements, and its geometric approximation is rather poor, especially near sharp angles. However, it is completely automated and can be integrated