Automatic Block Decomposition of Parametrically Changing Volumes

R. Taghavi[1]

1 Simulation Works Inc., 401 The Empire Building, 360 North Robert Street, Saint Paul, MN 55101 - United States
e-mail: taghavi@siw.com

Résumé — **Décomposition automatique par blocs de volumes variables paramétrisés** — Une méthode est ici proposée pour la décomposition automatique de volumes instationnaires tels que ceux rencontrés dans les moteurs en calculs de structure en élément finis ou de mécanique de fluides. Des exemples d'application de cette méthode à la génération de maillages en hexahèdre sont présentés.

Mots-clés : maillage.

Abstract — *Automatic Block Decomposition of Parametrically Changing Volumes* — *A method is introduced for the automatic decomposition of time-varying volumes such as those encountered in engine FEA and CFD. Examples of the application of this method to all-hexahedral mesh generation are also presented.*

Keywords: meshing.

INTRODUCTION

As finite-element analysis (FEA) and computational fluid dynamics (CFD) become standard components of mechanical design, grid generation emerges as a serious bottleneck. In most cases, automatic tetrahedral (tet) meshing offers a viable solution. However, hexahedral (hex) meshes are still in demand due to a host of reasons that are not always technical in nature:

– certain applications do not accept tets;
– certain FEA and CFD applications cannot properly and accurately use tets;
– there is sometime a lack of symmetry inherent in solutions obtained with tets;
– tetrahedral (tet) meshes may require up to an order of magnitude more elements than equivalent hexahedral (hex) meshes, especially for boundary layers or high aspect ratio parts;
– FEA and CFD applications exact a higher quality from tets than hexes;
– tet meshes are harder to view and repair. For instance, the vertices of a one million-node tet mesh, scaled to an XGA screen, will hardly leave any pixels for the cursor, let alone node labels.

1 MOVING MESHES

FEA and CFD analysis for changing volumes such as combustion chambers, introduce additional mesh requirements. Since most CFD applications require greater mesh densities than users can afford, it is desirable that during geometry change, mesh adaptation be achieved mostly through stretching while maintaining element connectivity constant for as long as possible. In this fashion, we can keep the number of rezonings to a minimum, and minimize truncation errors. We will call this a robust block decomposition. In manual mesh generation, a smart choice of the initial block decomposition is a robust one which enables the user to maintain the same blocking throughout the geometry change, limiting rezoning to the addition or removal of a few element layers here and there.

In Delauney or advancing front mesh generation methods, no block decomposition is used, and consequently, there is no guaranty that a small parametric change in the geometry will not initiate a complete rezoning.

2 BLOCK DECOMPOSITION

The problem of automatic block decomposition is approached as the search for the coarsest possible all-hex mesh of a closed volume. We are also looking for a "robust" decomposition, so that it remains unchanged if the geometry is modified through "small" homeomorphic transformations, or more generally, any "small" transformation that maintains the connexity of the geometry.

Oil & Gas Science and Technology – Rev. IFP, Vol. 54 (1999), No. 2

3 AUTOMATIC SURFACE PARTITION USING FUZZY REASONING

Given an arbitrary shaped closed triangular surface mesh *S*, we assume that we already have a structured block decomposition of its internal volume. Considering the collection *B* of quadrilateral elements constituting the outer surface of the block decomposition, and assuming that every vertex of *B* is a vertex of *S*, each quad Q_i of *B* can be mapped into a collection of neighboring triangles A_i of *S* (Fig. 1).
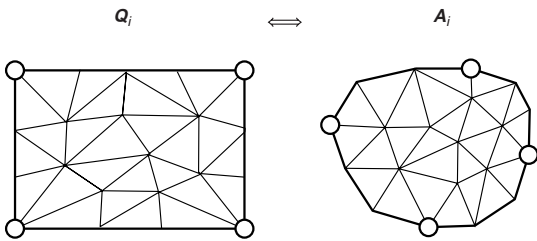

Figure 1

Mapping of a block face into a triangle set.

In this fashion, $\{Q_i\}$ with $i = [l, n]$, defines a partition of *S*:

$$S = \bigcup_{i=0}^{n} A_i \text{ and } \forall(i, j) A_i \bigcap A_j = \varnothing$$

As a result, we have $\sum_{i=0}^{n} \text{card} (A_i) = N$

Where:

A_i is the set of triangles included in Q_i

N is the total number of triangles

S is the triangulated surface the interior of which we want to decompose

n is the total number of triangles in *S*.

Conversely, we argue that any partition $\{A_i\}$ of *S* is the outer quad mesh of a structured block decomposition of the interior of *S*, if it satisfies the following non-exhaustive set of rules:

– the partition contains at least 6 sets ($6 \le n$);

– there is an even number of sets in the partition ($n = 0$ (2));

– two opposed directions may not coexist at a vertex of *B*;

– no folding should occur at any vertex of *B*;

– …

For a given surface partition to be the outer surface of a structured block decomposition, it is necessary that the above rules be satisfied. Starting with an initial partition $\{A_i\}^0$, we check for the above rules, and if any one is violated, we perturb partition $\{A_i\}^0$ to obtain partition $\{A_i\}^1$, and so on, using fuzzy logic reasoning, until all the rules are adhered to for $\{A_i\}^{last}$. This surface partition represents the outer surface of a realizable block decomposition.

4 APPLICATION TO ALL-HEXAHEDRAL MESH GENERATION

Once the block decomposition is at hand, mesh generation simply consists in refining each block down to the user-requested level. Figure 2 shows three simple meshes obtained from the same block decomposition at three levels of resolution.

5 EXAMPLES

This methodology has been successfully implemented in the Kubrix automatic mesh generator. Figure 3 shows several examples of engine components meshed using this method.
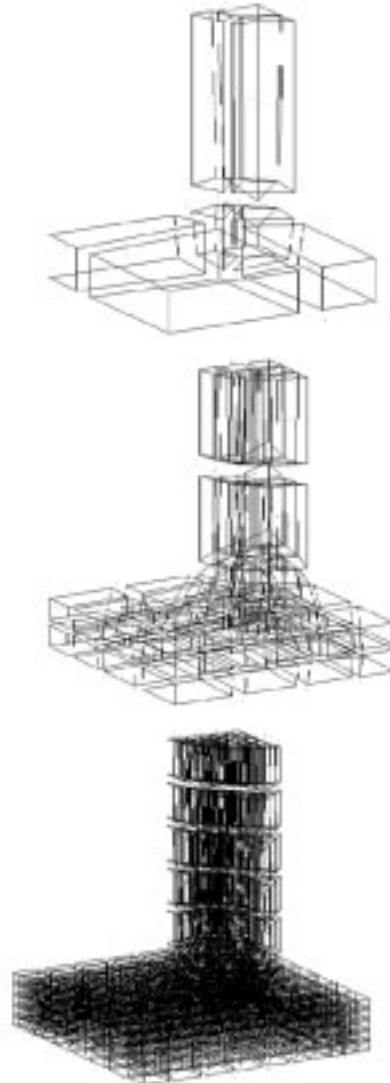

Figure 2

Meshing of a block-decomposed part with a resolution of, from left to right, 1, 2, and 5.
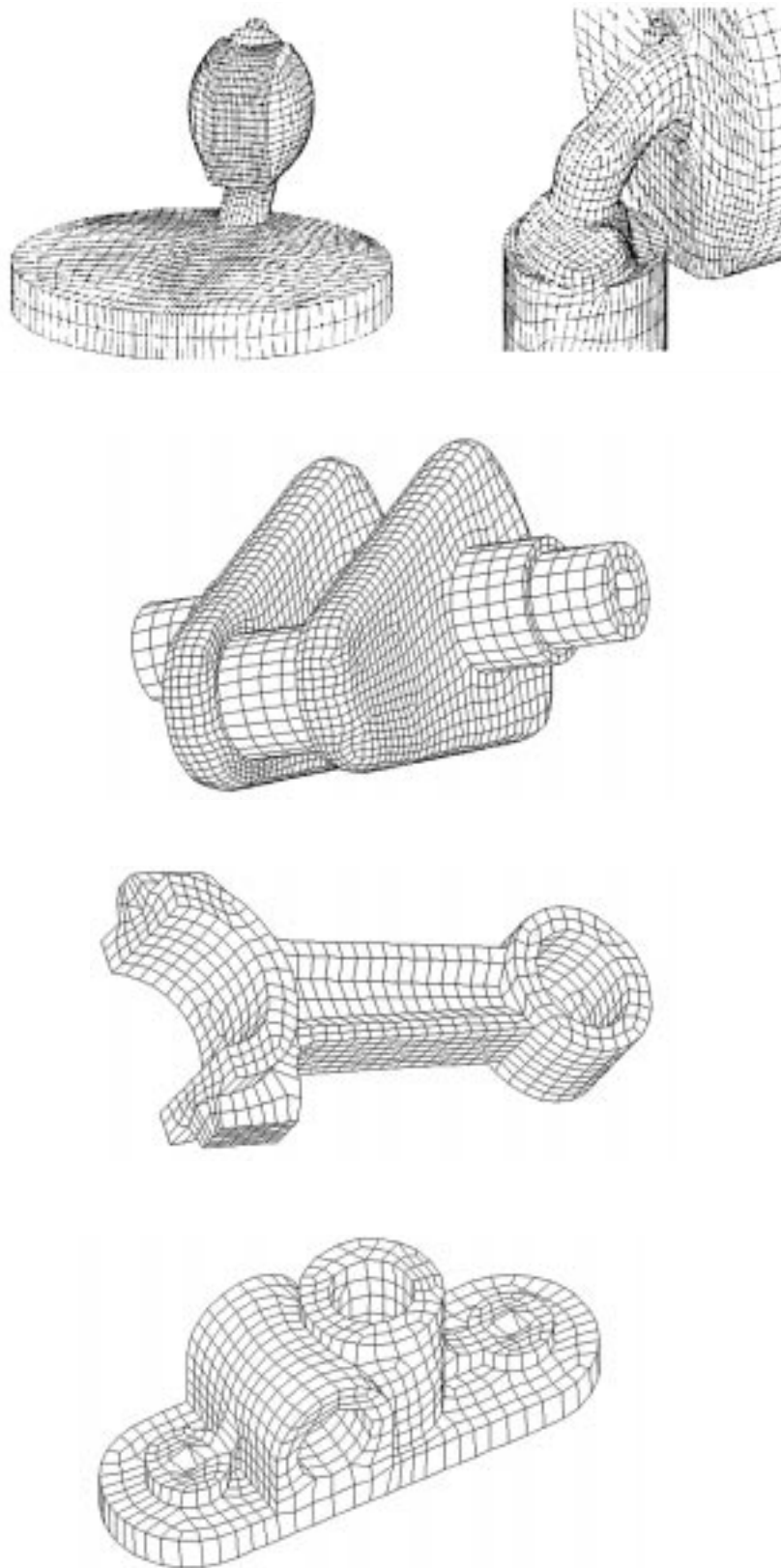
Figure 3

Examples of meshes using fuzzy-logic block decomposition.

CONCLUSION

Fuzzy-logic reasoning is used to obtain a partitioning of a closed triangular surface mesh into quads. The applied rules guarantee that the resulting quad mesh represents the surface of a structured volume block decomposition. This method is not only applicable to volumes as obtained with solid modelers, but also to multi-material volumes bound by non-manifold surfaces representing the boundary of a collection of juxtaposed solids separated by non-repeated surfaces.