Algorithms for Quadrilateral and Hexahedral Mesh Generation

Robert Schneiders
MAGMA Gießereitechnologie GmbH
Kackertstr. 11
52072 Aachen
Germany
Email: R.Schneiders@magmasoft.de

Abstract

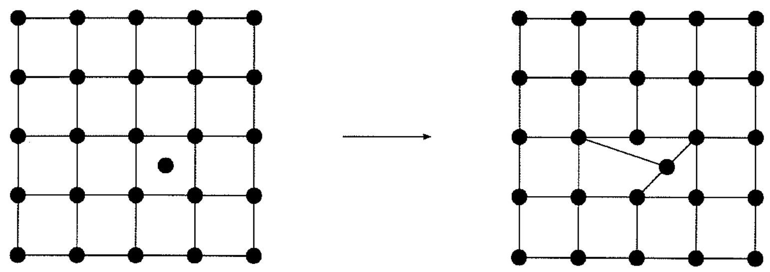
This lecture reviews the state of the art in quadrilateral and hexahedral mesh generation. Three lines of development – block decomposition, superposition and the dual method – are described. The refinement problem is discussed, and methods for octree-based meshing are presented.

1 Introduction

Quadrilateral and hexahedral elements have been proved to be useful for finite element and finite volume methods, and for some applications they are preferred to triangles or tetrahedra. Therefore quadrilateral and hexahedral mesh generation has become a topic of intense research.

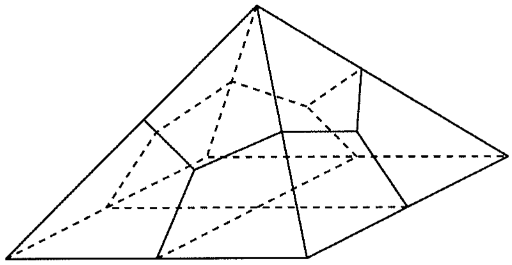
It turned out that especially hexahedral mesh generation is a very difficult task. A hexahedral element mesh is a very "stiff" structure from a geometrical point of view, a fact that is illustrated by the following observation: Consider a structured grid and a new node that must be inserted by using local modifications (fig. 1). While this can be done – not in a very elegant way – in 2D, it is impossible in 3D! Thus, one cannot generate a hexahedral element mesh by point insertion methods, a technique which has been used successfully for the generation of tetrahedral element meshes (Delaunay-type algorithms).

Figure 1: Inserting a point into a structured quadrilateral element mesh



Many algorithms for the generation of tetrahedral element meshes are advancing front methods, where a volume is meshed starting from a discretization of its surface and building the volume mesh layer by layer. It is very difficult to use this idea for hex meshing, even for very simple structures! Fig. 2 shows a pyramid whose basic square has been split into four and whose triangles have been split into three quadrilateral faces each. It has been shown that a hexahedral element mesh exists whose surface matches the given surface mesh exactly [Mitchell 1996], but all known solutions [Carbonera] have degenerated or zero-volume elements.

Figure 2: Surface mesh for a pyramid



The failure of point-insertion and advancing-front type algorithms severely limits the number of approaches to deal with the hex meshing problem. Most algorithms can be classified either as block-decomposition, superposition or dual methods, which will be presented in section 2, section 3 and section 4.

Adaptive mesh generation is more difficult than for triangular and tetrahedral meshes. Fig. 3 shows an example: The mesh in fig. 3a has been derived from a structured quadrilateral mesh by recursively splitting elements. In order to get rid of the hanging nodes, neighboring elements are split to create a conformal transition to the coarse part of the mesh (fig. 3b). This probleem is equivalent to the generation of quadrilateral/hexahedral meshes from a quadtree/octree structure.

Figure 3: Quadrilateral mesh refinement

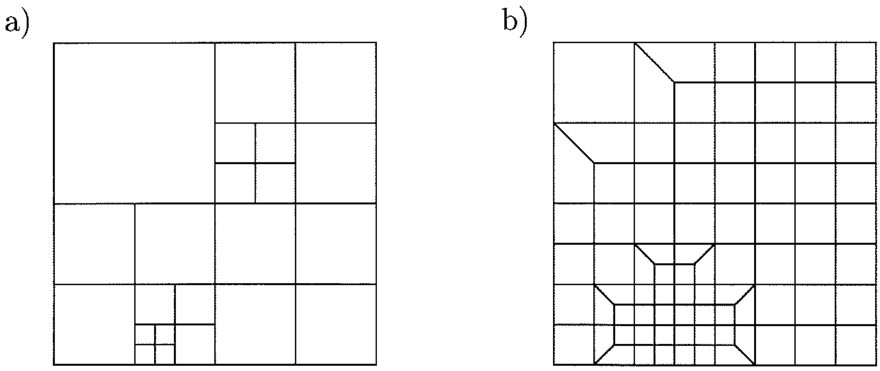


Fig. 4 shows a simple example for the 3D case (motivated by octree-based meshing) where the fine region seems to be connected to the coarse in a reasonable manner. Unfortunately, the solution is not valid!

This can be concluded from the following relation between the number of elements H, the number of internal faces F_i and the number of boundary faces F_b of a hexahedral mesh:

$$6 \cdot H = 2 \cdot F_i + F_b \tag{1}$$