ARTICLE IN PRESS

Computer-Aided Design ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

A template-based approach for parallel hexahedral two-refinement☆

Steven J. Owen [a,*], Ryan M. Shih [b], Corey D. Ernst [c]

[a] Sandia National Laboratory, Albuquerque, NM, USA
[b] University of California, Berkeley, CA, USA
[c] Elemental Technologies, American Fork, UT, USA

ARTICLE INFO

Keywords:
Hexahedron
Mesh refinement
2-refinement
Adaptivity
Parallel

ABSTRACT

We provide a template-based approach for generating locally refined all-hex meshes. We focus specifically on refinement of initially structured grids utilizing a 2-refinement approach where uniformly refined hexes are subdivided into eight child elements. The refinement algorithm consists of identifying marked nodes that are used as the basis for a set of four simple refinement templates. The target application for 2-refinement is a parallel grid-based all-hex meshing tool for high performance computing in a distributed environment. The result is a parallel consistent locally refined mesh requiring minimal communication and where minimum mesh quality is greater than scaled Jacobian 0.3 prior to smoothing.

Published by Elsevier Ltd.

1. Introduction

Massively parallel platforms, such as those deployed at the US Department of Energy Laboratories, have enabled computational simulation of enormous complexity. For applications requiring hexahedral elements, traditional methods of mesh generation can require significant user interaction which will not easily scale for these problems. Fully automatic, scalable and embedded meshing methods are an increasingly important requirement for these next-generation computing platforms. Mesh generation based on an overlay grid procedure is an ideal candidate for high performance computing, however to be effective it must provide for geometry-sensitive mesh size adaptation.

Overlay grid procedures for generating all-hex meshes [1–3] usually rely on some form of refinement strategy to capture small features. Most of these methods begin with a regular three-dimensional Cartesian grid that is adaptively refined based on various geometric criteria to form an octree subdivided mesh. A Boundary representation (B-rep) of the geometry of interest is then super-imposed on the octree mesh, where nodes are snapped to the geometry and elements falling outside of the B-Rep are discarded.

In order to maintain continuity between refined and unrefined elements in the mesh, transition patterns are normally imposed. These patterns can utilize either a 3-refinement or a 2-refinement methodology. For 3-refinement, each edge of the uniformly refined grid is divided into three segments. On a three-dimensional hex element, this results in a $3 \times 3 \times 3$ subdivision or 27 elements. 2-refinement, on the other hand, will split each edge in two resulting in a $2 \times 2 \times 2$ subdivision with 8 elements.

Most refinement operations can be thought of as introducing a pillow layer of hexes surrounding a column of hexes. Fig. 1 illustrates a 2D 3-refined mesh where the green elements were initially marked for uniform refinement. An example continuous pillow layer of quads (shown in red) is shown wrapping a single column of quads, noting that the same pattern is repeated throughout the mesh. Fig. 2 illustrates a 2-refined mesh with a similar pillow layer surrounding two columns of quads. In 3-refinement, we note that the pillow layer can be accomplished within a single quad layer, whereas 2-refinement requires at least a pair of quad columns to accomplish the pillow. This problem extends to 3D (shown in Fig. 3) where sheets of hexes must be introduced to accomplish the refinement. For 3-refinement, since the pillowed sheet of hexes can be effected within a single column of hexes, each refinement transition can be performed independently and within a single element, making its implementation relatively straightforward. In contrast, 2-refinement must determine a consistent pairing of hex layers to effect the refinement transitions, making its implementation more challenging.

The 3-refinement strategy can produce undesirable high mesh size gradients in transition regions. In spite of this, it remains the most popular form of refinement because of its ease of implementation. The 3-refinement pattern, illustrated in Fig. 1,

☆ This paper has been recommended for acceptance by Chennai Guest Editor.

* Corresponding author.
E-mail address: sjowen@sandia.gov (S.J. Owen).

http://dx.doi.org/10.1016/j.cad.2016.09.005
0010-4485/Published by Elsevier Ltd.

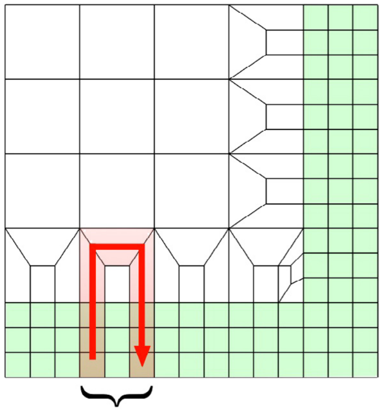

Fig. 1. Example 2D 3-refinement showing pillow loops accomplished within a single layer of elements. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

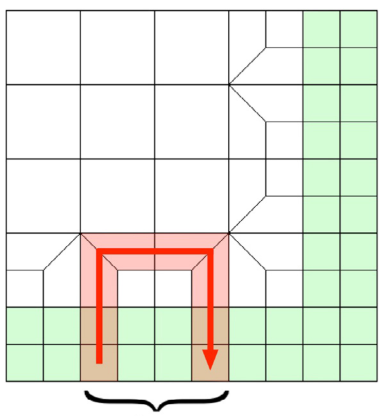

Fig. 2. Example 2D 2-refinement showing pillow loops requiring at least a pair of adjacent element layers.

has well-defined templates for transition elements that can be introduced based on a set of marked nodes. First introduced by Schneiders [4] the number and pattern of marked nodes on an element define the precise subdivision template to be used. This deterministic template-based approach to refinement, is relatively well-understood and easy to implement. Although beneficial for implementation, the change in element size and resulting mesh quality in the transition regions for 3-refinement can be problematic.

In contrast, 2-refinement, illustrated in Fig. 2, by most measures is a more desirable approach because of the smoother size transitions that can be effected. Its implementation, however, can be more difficult, particularly for parallel distributed domains. Complications in 2-refinement can arise when the pairing patterns to form transition elements from nearby refinement zones intersect with each other, or when the pairing of element layers must extend across processor boundaries. Several methods have been proposed which appear to present good results for serial applications, however implementation details for some of these methods are sparse and their application to distributed memory parallel is not addressed.

We address the need for hexahedral mesh refinement for distributed memory parallel environments. A refinement strategy that uses a deterministic algorithm that yields the same results regardless of the domain decomposition strategy is desirable.

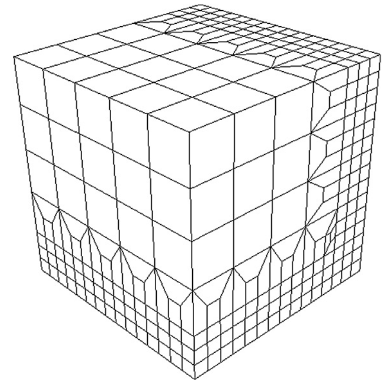

Fig. 3. Example 3D mesh using a 3-refinement procedure.

Because templates can provide local criteria for subdivision, they provide an attractive solution for parallel applications where very little inter-processor communication is necessary.

In this work we introduce a new deterministic 2-refinement strategy based on templates and marked nodes. We limit our scope to structured grids since they are used as the basis for our target overlay grid application [5]. We propose a method that provides the following characteristics:

1. One-to-eight uniform refinement.
2. Controlled mesh size gradients.
3. Minimum mesh quality within transition elements of 0.3 scaled Jacobian.
4. Parallel-consistent (same result regardless of decomposition).
5. Scalable to massively parallel environments.
6. Multiple levels of refinement.

2. Previous work

Most localized hex refinement strategies in the literature have largely been based on Schneiders' initial work on Octree meshing [4]. A detailed set of 3-refinement templates were developed and used to locally capture feature sizes. Schneiders later expands these procedures to include 2-refinement [6], introducing templates similar to those shown in Figs. 4 and 5. In this work, the concepts of directionally refined parallel layers is introduced, as well as a node marking strategy based on alternating nodes surrounding the refinement region to enforce pairing. We note that because of the selected overlapping strategy of the intersecting pillowing [7] operations, that transitions can quickly become complex, especially for concave regions reducing resulting mesh quality.

Harris [8] et al., and later expanded by Edgel [9] and Malone [10] also provide background and implementation of a directional parallel pillowing strategy for 2-refinement. These works do not address the use of templates, instead use pillow operations on arbitrarily selected adjacent hex layers based on expansion from an initial uniform refinement zone. Mesh quality in transition zones as well as generality and extension for concave regions also was problematic in these works. In contrast, in this work, we extend Harris' parallel pillowing strategy to use a global deterministic template-based approach that is applicable for any shaped region while maintaining mesh quality in excess of 0.3 scaled Jacobian in transition regions.

Other 2-refinement strategies include Ebeida et al. [11], Marechal [3] and Zhang et al. [12]. While each of these methods offers unique benefits, they each begin with a balanced octree decomposition of an initial uniform grid. They later define column groupings of four elements extended from the octree surrounding