ELSEVIER

Computer-Aided Design 37 (2005) 173–187

COMPUTER-AIDED DESIGN

www.elsevier.com/locate/cad

Three-dimensional anisotropic geometric metrics based on local domain curvature and thickness

Ko-Foa Tchon[a,*], Mohammed Khachan[b], François Guibault[b], Ricardo Camarero[a]

[a]Department of Mechanical Engineering, École Polytechnique de Montréal, C.P. 6079, Succ. Centre-ville, Montreal, Que., Canada H3C3A7

[b]Department of Computer Engineering, École Polytechnique de Montréal, C.P. 6079, Succ. Centre-ville, Montreal, Que., Canada H3C3A7

Received 3 December 2003; received in revised form 8 March 2004; accepted 7 May 2004

Abstract

A three-dimensional anisotropic Riemannian metric is constructed from a triangulated CAD model to control its spatial discretization for numerical analysis. In addition to the usual curvature criterion, the present geometric metric is also based on the local thickness of the modeled domain. This local thickness is extracted from the domain skeleton while local curvature is deduced from the model triangulated boundaries. A Cartesian background octree is used as the support medium for this metric and skeletonization takes advantage of this structure through an octree extension of a digital medial axis transform. For this purpose, the octree has to be refined according to not only boundary curvature but also a local separation criterion from digital topology theory. The resulting metric can be used to geometrically adapt any mesh type as long as metric-based adaptation tools are available. To illustrate such an application, geometric adaptation of overlay meshes used in grid-based methods for unstructured hexahedral mesh generation is presented. However, beyond mesh generation, the present metric may also be useful as a shape analysis tool and such a possibility could be explored in future developments.

Keywords: Riemannian metric; Skeleton; Octree; Digital topology; Mesh adaptation

1. Introduction

Preparing CAD models for numerical analysis usually involves repairing dirty geometries and removing unwanted features to make meshing possible. The accuracy of analysis tools such as the finite element and finite volume methods is strongly dependent on the quality of this mesh. Control of the size, stretching and orientation of the mesh elements is thus crucial. User experience can guide the generation of the mesh to manually adapt it to the problem at hand. Higher vertex densities can be requested in expected high load regions or boundary layers, for example. Such an a priori approach is, however, tedious and very approximate. Automatic methods based on a posteriori error estimators have received extensive attention over the years and proved the effectiveness of solution-based mesh adaptation. See Ref. [1] and the references cited therein, among others. The object-oriented remeshing toolkit (OORT) developed at the Center for Research on Computation and its Applications (CERCA), and now at École Polytechnique de Montréal, implements such methods [2]. However, when generating an initial mesh from a CAD model, no solution is yet available. Alternative methods based on the domain geometry must then be used and can be setup as an additional preprocessing step of the CAD model before meshing and numerical analysis.

The numerous unstructured mesh generation methods presented in the literature propose many different geometric adaptation approaches. However, like their solution-based counterparts, these algorithms always need to first map the characteristics of the target mesh elements at every point of the domain. Early advancing front methods relied on user specified sample points manually triangulated to form a coarse simplicial background mesh [3]. Target mesh properties were then computed at any point of the domain by locating the host background element and linearly interpolating the sample vertex values. Automated alternatives have then been developed using unconstrained Delaunay triangulations of the vertices of pre-meshed domain boundaries [4]. The target mesh spacing was then

* Corresponding author. Tel.: +1-514-340-4711x4582; fax: +1-514-340-5917.

E-mail address: ko-foa.tchon@polymtl.ca (K.-F. Tchon).

0010-4485//$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cad.2004.05.007

interpolated in the domain from boundary specified parameters. Furthermore, the discretization of the boundary itself can be automated using curvature, angle and proximity criteria. See Ref. [5] for example. However, such so-called empty Delaunay meshes are very coarse and may result in unwanted abrupt variations of the target mesh properties. To alleviate this side effect, an alternative interpolation scheme based on natural neighbors has been proposed [6]. Even smoother maps can be generated by diffusing target mesh parameters in uniform Cartesian background grids using point and line sources and a Poisson equation [7]. The resulting mesh gradation is very smooth and the uniform structure of the background grid facilitates host location for target parameter interpolation. A uniform grid cannot, however, capture very complex target maps with extreme length scale variations. Quadtrees, in two dimensions, and octrees, in three dimensions, are better suited for such maps because they enable local refinement while retaining an implicit recursive structure facilitating host location. The use of quadtrees and octrees for unstructured simplicial mesh generation has been pioneered two decades ago [8] and a review can be found in Ref. [9]. These methods recursively divide the domain bounding box until the boundary features are adequately resolved and store the result in a tree structure. Allowing only a difference of one refinement level between neighboring cells results in smooth gradation. To generate a valid mesh, the tree cells are then usually split into simplicial elements and the boundary is recovered. However, since the size distribution of the terminal cells is well adapted to the domain geometry by construction, the final tree structure can also be used almost directly as a target map for other meshing algorithms such as the advancing front method [10]. Quadtrees and octrees can also be used solely as support media for more elaborate sizing functions. Their refinement is then not directly based on the domain geometry but rather on the adequate capture of the sizing function gradients [11].

The above list of geometry-based mesh sizing control strategies is far from exhaustive and their combination would give infinite possibilities. Two main ideas emerge however. First of all, target mesh specifications may take many forms but storage as a Riemannian metric in a background mesh seems the most flexible approach. Such a control map can be constructed during the preparation of the CAD model and is decoupled from both the adaptation algorithm as well as the target mesh type, i.e. structured, unstructured or hybrid for example. This approach is also potentially compatible with solution-based adaptation algorithms. The second common idea is that geometric adaptation should be based on the local curvature of the domain. Curvature-based sizing is commonly used for curvilinear and surface meshes and has a solid theoretical foundation [12]. It is, however, insufficient to simply diffuse such a sizing throughout a three-dimensional domain. An additional adaptation criterion based on the local thickness of the domain must be introduced to take into account regions with flat and narrow gaps for example. Designing such a criterion is not trivial. Most previous attempts use heuristics based on proximity between boundary vertices, segments and facets, and strongly depend on the boundary mesh itself. That is why, as Quadros et al. [13], the present work proposes a more consistent approach based on skeletons to compute the local domain thickness. More precisely, digital topology theory is used to extract an approximate domain skeleton from a Cartesian background octree. To resolve possible small gaps in the domain, this octree is refined according to not only boundary curvature but also a topological separation criterion. Furthermore, to enable anisotropic adaptation, the octree is only used as a support for a Riemannian metric extracted from the domain boundary curvature tensor and the local thickness information embedded in the skeleton. The resulting algorithm has been implemented in a package called *GeoMetric* and applied to overlay mesh adaptation for grid-based unstructured hexahedral mesh generation methods.

2. Background octree generation

2.1. Domain definition

The required input for the Cartesian background octree generation is a domain geometry definition. This definition must allow boundary intersection and inside-outside tests for the octree cells, as well as closest point and local curvature interrogation. For the present project, triangulated boundary representations generated by CAD systems, as STereo Lithography (STL) files for example, were used. Since such triangulations simply serve as a support for geometric information, they do not have to be of high quality. They can be too fine but should not be too coarse or essential details will be lost. Ultimately, the user decides the level of details to be taken into account. Triangulations can also be *dirty*, i.e. not watertight. Dirt size should, however, be inferior to the size of the neighboring cells to make it invisible to the octree. One way to insure that is to make dirt size inferior to the size of the smallest possible cell. To accelerate intersection tests, the triangles are stored in an alternating digital tree (ADT) [14] and, to improve accuracy and robustness, adaptive precision arithmetic is used [15]. Furthermore, simulation of simplicity (SoS) copes with degenerate intersection configurations such as barely touching entities [16]. Finally, curvature information can be given by the user along with the triangle vertices or it can be estimated directly from the triangulation [17]. Fig. 1 shows the triangulation of an intricate mechanical part, i.e. a water jacket, that will be used throughout the present paper to illustrate the different steps of the algorithm.

2.2. Octree refinement

The Cartesian background octree is generated through recursive non-conformal refinement of the domain bounding box. To resolve the significant features of the geometry,