Finite Elements in Analysis and Design 136 (2017) 1–17

Contents lists available at ScienceDirect

Finite Elements in Analysis and Design

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

Yet another hexahedral dominant meshing algorithm: HexDom

B. Kaan Karamete [a,*], Romain Aubry [a], Eric L. Mestreau [a], Saikat Dey [b]

[a] Sotera Defense Solutions, 1501 Farm Credit Drive, McLean, VA 22102, USA
[b] US Naval Research Laboratory, Code 7130, 4555 Overlook Ave SW, Washington DC 20375, USA

ARTICLE INFO

keywords:
Hexahedral volume meshing
Cavity meshing
Tetrahedra
Boundary layers

ABSTRACT

In this paper, we describe a robust meshing algorithm for obtaining a mixed mesh with large number of hexahedral/prismatic elements grown over the domain boundary respecting the user imposed anisotropic metric where physics matter the most and in areas where it is required to have the least number of elements. The inner section away from the boundaries is filled with the terminal octants of a non-conformal octree. The remaining unmeshed portion of the domain within the hexahedral/prismatic faces is filled with narrow bands of tetrahedra. The novel idea of the meshing algorithm is the formation of the cavity as slim as possible between the exposed faces of the outer most boundary layers and the octant faces of the inner most terminal octants, in such a way that the length scales of the cavity mesh spacings would allow the frontal tetrahedral meshing algorithm robustly succeeding to fill the cavity respecting its boundary faces without recovery issues. The algorithm could be applied to non-cubical, arbitrary geometries that can also be non-manifold. Each domain region is meshed recursively and within which, the tetrahedral filling algorithm constructs as many manifold cavity shells as the problem constrains are imposed by the boundary layers and the mesh size settings. The final hexahedral dominant mesh is exported to a face-based finite volume format (OpenFoam) so that the non-manifold nature of the mesh is captured by flux based numerical solvers consistently and accurately.

1. Introduction

Accurate computation of flow fields around complex geometries has many practical applications and remains a challenge. Understanding of hydrodynamics of surface-ships and submarines often relies on numerical solution of the Reynolds-Averaged-Navier-Stokes (RANS) problem. The need to discretize the exterior volume in the immediate vicinity of the structure to resolve the viscous boundary-layer, as well as include large extents of the exterior volume including the free-surface, often result in excessively large numerical problems. Due to the ability to discretize large volumes with fewer elements when compared to tetrahedra, hexahedral meshing of the volume in such applications offers computational savings over a purely tetrahedral approach. For applications of interest here, flow fields generally impose greater gradients around the boundaries than anywhere else in the domain. Alternatively, the gradients do not change as much in the bulky interior regions of the problem domain. Therefore, fine and gradient compatible directional resolution in the mesh are needed to capture the solution gradients around the boundaries for accuracy, and fewer elements are desired to fill the space in the interior to save from thecomputational effort. For both accounts, the use of hexahedral/prismatic elements make sense more than the tetrahedra. Hence, based on these observations, our intent in this study is to come up with a fast and robust algorithm of generating hexahedra dominant (HexDom) meshes where it makes sense for the numerics of the computational physics simulation. In the remainder of this section, we explore the hexahedral meshing algorithms in the literature, attempt to point out their pros-cons and introduce our approach.

The use of the octrees in meshing has been an attractive option due to its definition of filling the space with hierarchically structured boxes without much algorithmic complexity [1–3]. The algorithms that make use of octrees in hexahedral meshing differ in the way how they treat and respect arbitrary geometric shape of the boundary enclosing the domain regions. In general, the boundaries can not be matched with the regular, orthogonal shape of the octants unless the geometry is a variant of or mappable to a cuboid.

Mapping option has been a favorable alternative due to its clear algorithmic formulation and efficiency in producing quality grids, that found enormous interest particularly in turbo-machinery, during much of the 1980–90s [4–7]. In this category, the domain is partitioned into

* Corresponding author.
E-mail address: kaan.karamete.ctr@nrl.navy.mil (B.K. Karamete).

http://dx.doi.org/10.1016/j.finel.2017.07.009
Received 21 March 2017; Received in revised form 24 July 2017; Accepted 25 July 2017
Available online XXX

blocks, homomorphic to cuboid, so that a mapping technique between the multi-block computational domain and the physical space could be applied. Mapped meshing problem was formulated akin to solving a set of elastic structural equations in elliptic or quasi hyperbolic PDE form to compute the coordinates of the hexahedral element vertices corresponding to the structured IJK equivalents defined in the computational block space [8,9]. This formulation required the domain to be partitioned into computational blocks and block faces/edges/vertices were to be associated with the true geometry. It was often preferred to use a facilitation step to compute the transfinite interpolation (TFI) followed by the elliptic smoother to eliminate possible fold-overs and establish orthogonality within some distance off the boundary layer walls [10]. Nevertheless, in general, the partitioning step remained to be the main bottleneck of the mapping option. The success of the semi-automatic partitioning alternative is debatable as the turn-around time to generate a quality hexahedral grid required a thorough training and a lot of experience in partitioning the complex geometric models and mapping the blocks. Though many software products [11,12] opted to use the semi-automatic approach in building the blocks, there were also studies advocated to use the automatic partitioning algorithms. Hence further research has been dedicated in the automation of the block partitioning schemes [13–15]. Some used the skeletal shape of the domain in partitioning [16]. The skeleton of the domain boundary is nothing but a dimensional reduction of its 3D volume to a surface where surface points are equidistant to the 3D surface, a.k.a., the medial axis (MA). The blocks were constructed by using the line and surface segments of the computed MA, thereby casting the problem of partitioning the domain into the problem of computing its MA [17]. This was somewhat problematic as well since the MA computation itself required the tetrahedralization of the domain. The stipulation is that numerically, the centers of the circumspheres of the generated tetrahedra converges to the true MA of the object [13]. Hence, the smoothness of MA surface relies on the coarseness level of the volume meshing with no or fewer interior vertices. There must be a required level of detail in the boundary surface mesh to prevent the ripples in the MA surface formation which remains to be the major numerical issues in this category [13,18]. In general, the state of the structured all hexahedral meshing technology has come to reach its limit with algorithms whose success depends on the effectiveness of its partitioning schemes and the accuracy of its elliptic solvers.

In this study, we use the octree based unstructured alternative by compromising on the all hexahedral mesh argument with hexahedral and prismatic element generation where it is desired the most as a better suited element type based on the general physics of the numerical problem. In the unstructured octree hexahedral dominant category of algorithms, there are two majorly distinct approaches in the way the octants match the curved boundaries. In the first approach, the set of octants closer to the boundary is detected and quadrilateral boundary octant faces are smoothed and projected over to the surface, implicitly creating the mixed type mesh on the surface [19–23]. This inflation and/or projection step may cause invalidity and quality issues, and requires heuristic complex algorithms to eliminate the invalid configurations that is extensively discussed by Schneider [1]. The ultimate remedy in tackling the mismatch and eliminate invalidity is the adjustment of the sampling size of the octree over the curved boundaries which often results in undesired mesh size sensitivity and unpredictable outcomes. As of late, the projection and snapping idea is furthered by embedding the boundary layers via subdivision of the first layer off the boundary (mother layers) with come restrictions on the first cell's height [24]. The second category consists of similar ideas akin to the iso-surface extraction from a field data. The conformal mesh topology at the intersections between the octants and the curved surface boundary, is resolved by the use of intersection element templates depending on the combinatorial look-up table according to the location of intersections within each octant [25,26]. These templates may result in highly distorted element formations and may not have all hexahedra but certainly more robust in its outcome in generating a valid mesh. However this approach also suffers from its sensitivity to the local octree sizing.

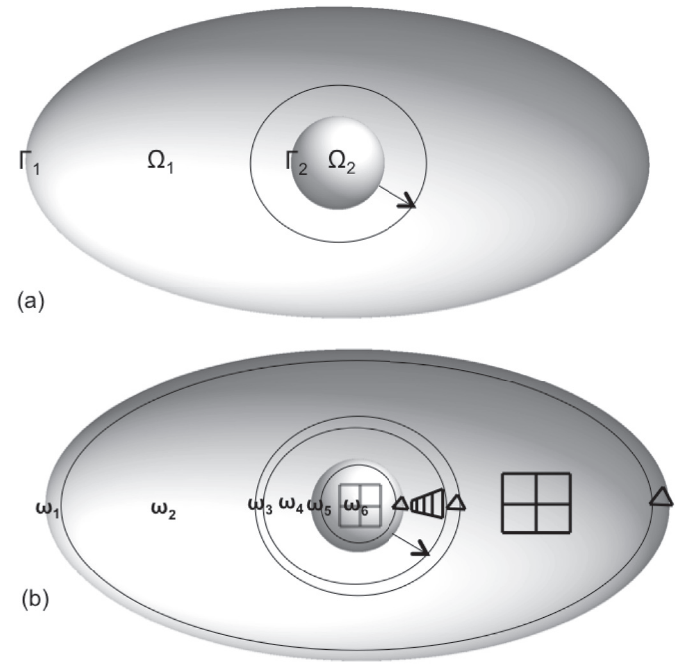

Fig. 1. The geometry and mesh domains of HexDom: (a) The geometry domains; Γ_1: The outer surface boundary of the region domain Ω_1, Ω_1:The outer region domain between the ellipse and the sphere, Γ_2: The inner surface boundary of the region domain Ω_2, Ω_2:The inner region domain for the sphere. The arrow shows the outer bound of the boundary layer thickness of Γ_2. (b) The mesh domains; ω_1: the cavity mesh between the surface mesh of Γ_1 and the outermost inner octants of Ω_1, ω_2: the octant mesh domain belonging to Ω_1, ω_3: the cavity mesh between the innermost octants of Ω_1 and the outermost layer of prismatic boundary layer mesh, ω_4: the prism mesh in the boundary layer thickness of Ω_1, ω_5: the cavity mesh between the surface mesh of Γ_2 and the outermost inner octants of Ω_2, ω_6: the octant mesh belonging to Ω_2.

As of late, there have been a series of improving ideas in the way the hex dominant meshing is conceptualized [27–29]. The central idea comes from the observation of the point distribution nature of the structured all-hex meshing. The regular distribution of mesh points guided by the enclosing boundaries, increases the chance of constructing hexahedral mesh topology. Mathematically proven 'frame fields' which is an extension to MAT concept is adopted to generate the regular point distributions [29]. The direct frontal approach or extrusion of the hexahedral elements over this point set has been studied. Alternatively, a tetrahedral frontal algorithm is proposed followed by the merging of tetrahedra into hexes using sophisticated set of templates [27]. Nevertheless, regardless of how the frontal algorithm is implemented, the hexahedral fronts inevitably collide and clash with each other not only in the domain but also on the surface boundary, leaving pockets of unmeshed cavities that may have dimensionally problematic or non-manifold topology and thus quite difficult to mesh with tetrahedra. Even though the general application to a wide range of complex boundaries has not been fully established yet, there were also reports of promising results using this approach lately [27].

Our idea is not too far off from the main point of the above technology except algorithmically it is much simpler and more robust in generating hexes and forming the cavities. The central idea is to put hexahedral elements where it is necessary for the accuracy and efficiency of the physical numerical simulation. Hence, the two basic goals of our algorithm is to generate hexes in the boundary layers and in the interior bulky sections non-conformally. Indeed, similar ideas have been exercised by Yerry and Shephard [2,3] in which the boundary layer generation process is considered as a post operation after the tetrahe-