Unstructured and Semi-Structured Hexahedral Mesh Generation Methods

J. Sarrate¹, E. Ruiz-Gironés¹, and X. Roca²

 ¹ Laboratori de Càlcul Numèric (LaCàN), Departament de Matemàtica Aplicada III, Universitat Politècnica de Catalunya, Barcelona, Spain
² Aerospace Computational Design Laboratory (ACDL), Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, USA.

Abstract

Discretization techniques such the finite element method, the finite volume method or the discontinuous Galerkin method are the most used simulation techniques in applied sciences and technology. These methods rely on a spatial discretization adapted to the geometry and to the prescribed distribution of element size. Several fast and robust algorithms have been developed to generate triangular and tetrahedral meshes. In these methods local connectivity modifications are a crucial step. Nevertheless, in hexahedral meshes the connectivity modifications propagate through the mesh. In this sense, hexahedral meshes are more constrained and therefore, more difficult to generate. However, in many applications such as boundary layers in computational fluid dynamics or composite material in structural analysis hexahedral meshes are preferred. In this work we present a survey of developed methods for generating structured and unstructured hexahedral meshes.

Keywords: mesh generation, quadrilateral mesh, hexahedral mesh, unstructured mesh.

1 Introduction

Spatial discretizations, represented by a mesh, have been related with computational methods since the appearance of the first electronic computers. Later, the evolution of computer graphics capabilities has induced to use meshes in new areas of application such as medical imaging, scientific visualization and 3D animation. However, automatic mesh generation was first prompted by many applications of computational methods in applied science and engineering. Specifically, the generation of a mesh is a pre-requisite for the application of several numerical techniques including the finite difference method (FDM), the finite element method (FEM), and the finite volume method (FVM).

The application of the above numerical techniques to 3D simulations lead to meshes composed by polyhedral elements. In these applications, the most common types are the tetrahedral (four triangular faces) and hexahedral (six quadrilateral faces) elements. There exist several fast and robust implementations of tetrahedral meshes. The classical three approaches to generate tetrahedral meshes are: the advancing front method [1–3], the Delaunay approach [4, 5] and the octree based methods [6, 7]. Additional information on these methods can be found in [8–10]. In these methods, local connectivity modifications are a crucial step. However, in hexahedral meshes, the connectivity modifications might propagate through the mesh. Therefore, hexahedral meshes are more constrained and only a limited type of geometries can be automatically meshed with high-quality hexahedral elements. Moreover, there exists no automatic hexahedral meshe that generates high-quality meshes on an arbitrary geometry. That is, hexahedral mesh generation is still considered as an open problem.

In industrial applications, the semi-automatic process for obtaining a hexahedral mesh, is often the most time-consuming task of the whole analysis. Nevertheless, hexahedral elements are preferred in a wide range of applications, see [11]:

- (i) In elastic and elasto-plastic analysis, eigenvalues of stiffness matrix for hexahedra are smaller than those for tetrahedra. Specifically, the tetrahedron is too stiff and locks in bending tests [12].
- (ii) In structural analysis, empirical studies have shown that hexahedral elements provide more accuracy than tetrahedral elements for the same computational cost [13, 14]. To obtain similar accuracy a tetrahedral mesh usually requires between four and ten times more elements than a hexahedral mesh.
- (iii) In Navier-Stokes computations, elements with high aspect ratio are required at boundary layers. Stretched hexahedra perform better than stretched tetrahedra capturing the anisotropy of the flow field over such viscous regions.
- (iv) In structural dynamics simulations of composite materials, elements strictly aligned with material features are required. Hexahedral elements reproduce better than tetrahedra the anisotropic properties of composite materials.

These advantages hold when comparing trilinear hexahedra versus linear tetrahedra but not in the case of their high-order versions, see details in [15]. Quadratic tetrahedra require less computational effort than quadratic hexahedra to obtain similar accuracy [13, 14]. However, hexahedra can also be suitable for high-order applications of the FEM. For instance, the spectral element method (SEM) is an accurate and efficient technique that explodes the characteristics of hexahedral elements [16]. Efficiency is achieved by using the tensor product of Gauss-Lobatto points to determine the location of both interpolation and integration points. This results in diagonal mass matrices and the adequacy to parallel implementations.

For these reasons, special attention is focused on developing automatic algorithms to generate hexahedral meshes. However, the utilization of hexahedra in industrial application is often hampered by the conversion of the computer aided design (CAD) model into a mesh adapted to the details of the geometry and to the prescribed distribution of the element size. Generating a mesh of the model can represent the ninety per cent of the analysis time [17]. In this process the most time-consuming tasks are:

- (i) Geometry healing. Usually, the CAD model that defines the geometry cannot be directly used in a mesh generation algorithm. For instance, the model may contain duplicated entities that corrupt the topology tree. In addition, meshing techniques usually require watertight models. That is, the geometry should not contain gaps in the boundary. In this step, it is usual that some parts of the geometry may have to be reconstructed from scratch.
- (ii) *Geometry de-featuring*. This step is related to the previous one and is often performed at the same time. The initial model usually contains small details that do not influence on the final analysis, but can increase the computational cost of the numerical simulation. For instance, small protrusions or extrusions, screw holes or extremely realistic representations of the geometry. The user has to manually remove them from the initial model. Hence, it is a time-consuming step that has to be performed by trained personnel that have to decide which features are relevant to the numerical analysis.
- (iii) Geometry decomposition. Depending on the mesh generation algorithm to be applied, the geometry has to be manually decomposed. This is especially true when the user needs hexahedral meshes, because there is no hexahedral mesher that can handle an arbitrary geometry. Typically, the decomposition step is not automatically performed and, for complex geometries, it can also be a difficult and time-expensive operation.
- (iv) Mesh generation. The user assigns the mesh generation algorithm to be applied and the prescribed element size. Note that incompatible element sizes may induce bad quality meshes, in the sense of element shape and accuracy of the initial geometry approximation. Moreover, it may impede the mesher to generate a mesh at all. In this case, the user has to manually detect the areas where an invalid element size has been prescribed and correct it.



Figure 1: (a) A hexahedral chord. (b) A hexahedral sheet.

Several works have analyzed the geometrical and topological properties of hexahedral meshes and their dual counterpart [11, 18–21]. These properties show that a hexahedral mesh cannot be considered as a *local* object, but as a *global* object, since a local modification in the mesh connectivity may propagate through the whole mesh. These properties can be summarized as follow:

- (i) *Opposing faces (dual chord)*. We can think of a hexahedra as three pairs of opposed faces. Since each face has an opposing face, a hexahedral mesh can be viewed as a collection of hexahedral chords, see Figure 1(a).
- (ii) Continuous layers (dual sheet). A hexahedra contains three sets of logically parallel edges. We can select all the logical parallel edges of a hexahedral mesh to construct a sheet of hexahedra. Thus, a hexahedral mesh can be viewed as a collection of hexahedral sheets that intersect each other, see Figure 1(b).
- (iii) *Element quality.* The angles between faces have to be less than π in order to obtain a positive element Jacobian. Moreover, the ideal mesh should contain hexahedra whose angles should be similar to $\pi/2$.

The first property implies that each dual chord that starts at the boundary also has to terminate at the boundary. Thus, the number of quadrilaterals in the boundary of a hexahedral mesh is an even number. More specifically, it is needed an even number of quadrilaterals at the boundary to generate a hexahedral mesh of the volume interior, see [18]. In addition, the first two properties imply that hexahedral meshes are more constrained than tetrahedral ones. For example, local connectivity modifications are straightforward in tetrahedral meshes. However, the local connectivity modifications of hexahedral meshes are propagated through dual chords and dual sheets. For this reason, local connectivity modifications of hexahedral meshes are difficult. Moreover, the same issue happens when refining or de-refining a hexahedral mesh. That is, the modifications introduced by the algorithm are propagated through the geometry. Finally, we highlight that the third property implies that high-quality hexahedral meshes are preferred to capture boundary layers.

During the last decades, several general-purpose algorithms for fully automatic hexahedral mesh generation have been proposed. These algorithms are described and

	Geometry Decomposition	Inside-Outside	Outside-Inside
Indirect			Combining Tris/Tets
methods			Q Morph/H Morph
	Methods for specific geometries	Grid-based	Primal Methods
	Submapping	Receding Front	Plastering
	Multisweeping/multiaxis sweeping		Uncons. plastering
Direct methods	General purpose methods Medial axis / surfaces Feature-based Local Dual Contribution Cross-Field based methods		Dual Methods Whisker Weaving

Table 1: Classification of quadrilateral and hexahedral mesh generation methods.

classified in the surveys by Owen [22], Blacker [11], Tautges [23], Baker [24], Shimada [25], and Shepherd [26]. A more detailed presentation of some quadrilateral and hexahedral mesh generation methods can be found in [8, 10]. However, none of the existent algorithms is robust, automatic and generates high-quality meshes for an arbitrary geometry. Therefore, special attention has been focused on existing algorithms that decompose the entire geometry into several simpler pieces. These smaller volumes can be easily meshed by well-known methods that exhibit an outstanding performance in these simpler volumes.

In this paper we present an overview of the current methods to generate quadrilateral and hexahedral meshes. It is worth to notice that they can be classified according to several criteria since they share several properties. For instance, the unconstrained plastering [27–29] can be considered as an unconstrained method or as a method that generates the hexahedral mesh advancing from the boundaries towards the inner part of the geometry.

Before classifying hexahedral mesh generation algorithms we consider *core methods*, see Section 2. Core methods are specific-purpose methods that deal with simple and non-general geometry configurations. Therefore, they are widely used to mesh the sub-volumes obtained by other methods that decompose the geometry into simpler pieces.

We first classify hexahedral mesh generation methods according to two main categories, see Table 1. On one hand, we consider *indirect methods* or methods that generate hexahedral meshes from a tetrahedral one, see Section 3. On the other hand, we consider *direct methods* or methods that directly generate the hexahedral mesh, see Section 4. In addition, hexahedral meshing algorithms can also be classified according to how the mesh is generated. In this sense we consider: methods that decompose the geometry in smaller pieces (automatically or based on human interaction), methods that first create the inner part of the mesh and then adapt it to the boundaries of the domain, i.e. *inside-outside* methods, and those that generate the hexahedral mesh advancing from the boundary towards the inner part of the geometry, i.e. *outside-inside* methods.

2 Core methods

Hexahedral mesh generation techniques have been improved in the last decade. Several algorithms have been devised in order to generate hexahedral meshes for any arbitrary geometry. However, a general and fully automatic hexahedral mesh generation algorithm is still an unreachable goal. That is, further research is still needed in order to work out a general purpose algorithm that, given any volume, generates high quality hexahedral elements at low cost (both in cpu and in user interaction time). However, specific algorithms have been developed in order to mesh specific geometries delivering high-quality meshes in a very efficient manner. In addition, they are used to mesh the small pieces or blocks obtained by decomposition methods. They can be classified in three groups: 1. Methods based on primitives (Section 2.1); 2. methods based on mappings (Section 2.2); and 3. the sweep method (Section 2.3).

2.1 Methods based on primitives

Primitive hexahedral meshers [30] are fast and easy to implement algorithms. While they are not applicable to a great number of geometries, they constitute the basis for more complex methods that are able to mesh more complicated geometries. The basic idea behind these methods is to identify simple geometrical shapes and mesh them with a predetermined template. A tetrahedra divided in four hexahedra is an example of a meshing template that can be applied to mesh, for instance, an octant of an sphere. Other templates include spheres, cylinders or boxes. The applicability of the algorithm is quite limited, but the element quality is usually good. In addition, it is worth to notice that several meshing environments include these kind of templates.

2.2 Methods based on mappings

This methods are designed to mesh domains that are topologically equivalent to a quadrilateral (2D) or hexahedra (3D) using a mapping between a computational domain and the physical domain. The mapping can be defined explicitly (algebraic methods or explicit conformal methods) or implicitly (PDE-based methods).

2.2.1 Algebraic mapping methods

The *transfinite interpolation method* or TFI [31] is the most used algebraic method to generate structured meshes on generalized boxes. In these methods an explicit expression is used to map a predetermined mesh on the computational domain onto the physical domain, see Figure 2. The mesh quality of the final mesh is directly related to the resemblance of the geometry with a box. The success of this method is based on the following five properties: 1. it is efficient from the computational point of view (both in terms of memory foot-print and computational time), 2. it is a Boolean sum of univariate interpolations, 3. it can accommodate several interpolation functions



Figure 2: Transfinite interpolation (TFI) method.

(liner, cubic, Hermite or Bezier/Bernstein polynomials), 4. it provides explicit control over grid point location; and 5. it is suited to incorporate grid spacing control. Other algebraic methods have been developed that further extend the potencial and the applicability of this explicit methods [32, 33]. A detailed presentation of algebraic and transfinite interpolation can be found in [8, 10, 34].

2.2.2 Explicit Conformal mappings

One of the most used techniques to generate structured meshes for outer domains was through the use of explicit conformal mappings. That is, a mapping that locally preserves the angles. Early on, the numerical methods to solve a PDE were based on finite differences techniques. Thus, orthogonal grids were desired in order to achieve better accuracy. The main idea is to solve the PDE in a computational domain discretized with an orthogonal grid. Then, the PDE is expressed in the coordinates of the computational domain. Since conformal mappings preserve the angles, the transformed PDE does not contain additional cross-derivative terms. However, conformal mappings had to be obtained for each different geometry, and its usage was abandoned in front of the elliptic-based methods. Some examples of conformal mappings are the Sells' circle mapping [35] to mesh the exterior domain of a two-dimensional airfoil, and the axisymmetric mapping of South and Jameson, [36].

2.2.3 PDE-based methods

PDE-based methods have been widely used for many decades to generate structured quadrilateral and hexahedral meshes [37, 38]. In these methods the mapping between a computational domain and the physical domain is defined implicitly by a PDE equation. Elliptic equations are the most used option for inner (bounded) domains. Control functions have been added to ensure orthogonality and element shape (stretched elements for boundary layers). Although elliptic methods are more expensive from the computational point of view than algebraic methods, they are preferred in several applications due to the smoothness of the obtained grids. For outer (unbounded) domains hyperbolic equations are also used in a large range of simulations.

erate the mesh by propagating an initial configuration. Although only partial control on the location of the mesh nodes in the outer boundary is possible, these methods provide control on orthogonality and element area. Hyperbolic methods are faster than elliptic ones. However, their major disadvantage is that singularities in the initial data propagate through the mesh. Parabolic methods combines, in some sense, the advantages of elliptic and parabolic methods. They maintain some advantages of the elliptic methods (smoothness, orthogonality, shape of the elements, specification of the exact nodal location on the outer boundary) and some other from hyperbolic methods (computational efficiency). A detailed presentation on PDE-based methods can be found in [8, 34, 39].

2.3 Sweeping

Most of the commercial CAD packages allow to model volumes by extruding, or sweeping, a surface along a delimited axis. These one-to-one sweep volumes are defined by a *source surface*, a *target surface* and a series of *linking-sides* (see Figure 3(a)). In order to ensure that a given geometry is one-to-one sweepable, the following conditions must be satisfied:

- (i) The source and target surfaces must be topologically equivalent (they must have the same number of holes and logical sides). However, they may have different areas and curvatures.
- (ii) The linking-sides must be mappable, or equivalently, defined by four logical sides.
- (iii) The sweep volume has one source surface and one target surface.
- (iv) The sweep volume must be defined by only one axis.

A detailed presentation on constraints which must be met for a volume to be sweepable, in a generic sense, are presented in [40]. Based on the definition of an extrusion geometry, the traditional procedure to generate an all-hexahedral mesh by sweeping consists of the following four steps:

- (i) Generation of a quadrilateral mesh over the source surface (structured or not).
- (ii) Projection of the source mesh onto the target surface.
- (iii) Generation of a structured quadrilateral mesh over the linking-sides.
- (iv) Generation of the inner layers of nodes and elements.

Several quadrilateral surface mesh generation algorithms can be used in the first step [41–46]. The gridding of the linking-sides involved in the third step can be generated using any standard structured quadrilateral surface mesh generator [8,47]. Hence,



Figure 3: (a) Generic one-to-one sweepable volume; (b) Available data for generating the inner layers of nodes: • boundary nodes of the cap surfaces, and \circ boundary nodes of the inner layers.

the two main issues to be dealt with by any sweep algorithm are the second and fourth step. In both steps, the meshes to be generated (i.e. the mesh over the target surface and the inner layer meshes) must be topologically equivalent to the source surface mesh.

In order to project the mesh on the source surface to the target surface, two different strategies are devised:

- (i) Least-squares approximation of an affine mapping. An affine mapping that converts the nodes from the source face to the target face computed. The mapping is generated by means of a least-squares approximation of an affine mapping, see [48, 49]. Then in order to improve the location of inner nodes, a boundary error method is introduced in [50] and then described in [51].
- (ii) Faceted projection. This method uses a triangulation to project the nodes from the source to the target faces, see [52]. First, the method generates a triangulation of the source and target faces using the same topological mesh. Thus, each triangle on the source face has a corresponding one in the target face. Then, the nodes on the source face are transferred to the target face using the triangulations via barycentric coordinates and elevation information. One special case of faceted projection is defined in [53], where the mapping between the source and target face is an harmonic function.

Projection methods based on facets [52] and least-squares approximations [48, 50, 51,54,55] are also used to compute the position of the inner nodes in step (iv). According to [56] projection algorithms based on a least-squares approximation of an affine mapping are the fastest option to project nodes. In [53], it is proposed a method based



Figure 4: Decomposition of: (a) a triangle into three quadrilaterals; and (b) a tetrahedron into four hexahedra.

on deformation cages. A computational domain is constructed in which the sweeping procedure is performed. Then, using a cage deformation method, the inner nodes are mapped from the computational domain to the physical domain.

3 Indirect methods

Indirect methods rely on a previous triangular or tetrahedral discretization of the domain. Then, this initial mesh is transformed in a quadrilateral or hexahedral mesh. Two approaches have been developed to perform this transformation. On the one hand, there are methods that decompose each triangle or tetrahedron into three quadrilaterals or four hexahedra, respectively (see Figure 4). On the other hand, there are methods that combine several tetrahedral elements to create a single hexahedron.

Decomposition methods, although easy to implement, have not been used in industry due to the low quality of the generated hehahedra. Therefore, special attention has been focused on methods that combine existing triangles and tetrahedra.

Several methods have been proposed to combine triangular elements in order to generate quadrilateral meshes. For instance, [57] is based on an heuristic combination of triangles resulting in a quad-dominant mesh. This technology was improved by considering an advancing-front approach during the triangle combination [58–60]. These methods start from the boundary of the domain and advance towards the inner part of the geometry until the fronts collide and no triangles are left. In this sense, they can also be classified as outside-inside methods. Reference [61] proposes a local method based on edge classification to transform triangular surface meshes into quadrilateral ones. On the contrary, in reference [62] a graph-based method to to combine triangles into quadrilateral elements using the Blossom algorithm. Later, in references [63, 64] the method is modified to incorporate an alignment of the vertices of the triangular

mesh to obtain oriented quads in the final mesh. A similar approach also base on graph theory is also presented in [65], where an open source implementation of an indirect method is detailed.

The advacing-front approach presented in the Q-morph method [60] were extended to three dimensions in the H-morph algorithm [66]. This algorithm generates hexahedral-dominant meshes for arbitrary volumes. That is, the mesh contains both tetrahedral and hexahedral elements. The H-Morph method starts with an initial tetrahedra. It uses an advancing front technique where the initial front consists of a set of prescribed quadrilateral surface facets. Fronts are processed and new hexahedra are generated. The procedure continues until no tetrahedra remain within the volume, or tetrahedra remain which cannot be transformed or combined into valid hexahedral elements. It is important to point out that Q-morph is a boundary sensitive method, since the mesh is generated in an outside-to-inside advancing-front manner, and can be applied to any type of geometries. However, its application in industry has been hampered by its failure to generate full hexahedral meshes in some cases.

4 Direct methods

Several approaches have been used to directly generate a full hexahedral mesh. On the one hand, there are methods that generate the final mesh by decomposing the geometry in sub-volumes that can be meshed using one of the core methods, see Section 4.1. On the other hand, several methods try to generate the hexahedral mesh by creating and inserting elements in the geometry. This creation and insertion process can be started in the inner part of the geometry and adjusted to reproduce the boundaries of the domain using an *inside-outside* approach, see Section 4.2, or can be started at the boundaries of the domain and then advanced inwards the geometry using an *outside-inside* approach, see Section 4.3.

4.1 Geometry decomposition

These methods rely on idea that a high-quality hexahedral mesh can be generated if the geometry is decomposed into simpler volumes. Some algorithms perform this decomposition taking into account specific properties of the geometry to be meshed, see Section 4.1.1. Therefore, its application is restricted to certain geometry configurations. On the contrary, other methods are designed to obtain a decomposition of an arbitrary geometry into simple blocks, see Section 4.1.2. Finally, it is worth to mention that there are several attempts to combine different hexahedral meshes into a single, conformal mesh, see [67–69]. In this way, the user can generate an hexahedral mesh on each part and then match them all to create a conformal mesh of the whole domain.

4.1.1 Methods for specific geometries

Submapping

One of the most important techniques to generate structured hexahedral meshes is the submapping algorithm [70, 71]. This method relies on a geometric decomposition of the domain into patches logically equivalent to an hexahedron. Then, each patch is meshed separately using a standard structured mesh generation algorithm such as the transfinite interpolation method. The mesh compatibility between patches is previously imposed by solving an integer linear problem (ILP). The quality of the obtained mesh is governed by the objective function of the integer linear problem.

The submapping method has been extensively used to mesh simply connected *blocky* geometries. Therefore, in [72–74], the authors develop special algorithms in order to automatically discretize surfaces and volumes with holes. Inner holes of the geometries are connected to the boundary in order to automatically generate a simply-connected geometry. Thus, the standard submapping algorithm can be applied. In the case of volumes with through holes, the solution of the integer linear problem may not guarantee the mesh compatibility, because there are missing equations. To overcome this drawback, the missing equations are generated using a graph representation of the geometry.

Reference [75] proposes a submapping technique that avoids the decomposition of the geometry. Once the boundary mesh is generated, the position of the inner volume nodes is computed by solving a linear problem.

Multisweeping and multiaxis sweeping

Sweeping algorithms have been successfully applied to extrusion geometries for many years. These algorithms can be applied in extrusion volumes defined by one source and one target surface, see [48, 49, 52, 53, 55]. For this reason, they are called oneto-one sweeping methods. However, real CAD extrusion models are not composed by a single one-to-one volume. Therefore, special attention has been focused on algorithms that automatically decompose any extrusion geometry into these simpler pieces. Many-to-one algorithms [76] can handle extrusion geometries with several source surfaces but a single target surface, see Figure 6(b). These algorithms decompose the initial volume into one-to-one sub-volumes and then discretize each one separately. In the last years, several algorithms have been proposed to mesh many-to-many extrusion geometries with multiple source and target surfaces, see [50, 51, 77–79]. Figure 6(c) presents an extrusion geometry with two source surfaces and two target surfaces. The many-to-many methods split the original geometry into many-toone sub-volumes. Each sub-volume is further decomposed into one-to-one barrels. The decomposition process is achieved by projecting target surfaces to corresponding source surfaces. In addition, an imprinting process between source and target surfaces is performed to determine the decomposition. Multi-sweeping algorithms have been improved in order to increase the robustness of the imprinting process and the



Figure 5: Examples of submapping meshes. Quadrilateral submapping mesh over an engine of a DLR F6 aircraft: (a) general view, and (b) detailed view. Hexahedral submapping mesh for a gear: (c) general view, and (d) detailed view.



Figure 6: Simple extrusion geometries with its source surfaces (light gray) and target surfaces (dark gray). (a) One-to-one extrusion geometry. (b) Many-to-one extrusion geometry. (c) Many-to-many extrusion geometry. (d) Multi-axis sweep geometry.



Figure 7: Many-to-many sweep meshes: (a) Mechanical piece; (b) gear; (c) linking-rod.

location of inner nodes created during the decomposition. These aspects are of the major importance when the geometry contains high-curved surfaces, and non-planar or twisted sweep levels. Finally, in [80], a multi-axis sweeping algorithm that can handle geometries with several sweeping axis is presented, see Figure 6(d).

Other Methods

Specific methods have been devised for a specific kind of geometries. For instance, tubular structures appears in many industrial and medical applications. In references [81,82] two different strategies for this kind of geometries are presented.

4.1.2 General purpose methods

Medial axis/surface

The medial axis transform has been widely used in mesh generation to decompose the initial geometry [83–87]. The medial axis is defined as the locus of the centers

of spheres that are tangent to the boundary of the geometry in two or more points, where all such spheres are contained inside the volume. While the algorithms based on the medial axis can potentially decompose any kind of volumes, the truth is that the computation of the medial axis is a difficult task that is not fully solved. The medial axis is heavily sensitive to boundary modifications. In addition, it may become degenerated, since it may be locally defined by curves or points. For instance, the medial axis of a sphere is a point.

Reference [88] proposes a new methodology to generate all-hexahedral meshes on arbitrary geometries by using the medial axis. The main idea consists on generating a quadrilateral mesh on the medial surface and then *sweep* the quadrilaterals from the interior to the boundary. However, problems may arise when the medial axis is degenerated, since there is no one-to-one mapping from the medial axis to the boundary. The medial axis is also used to analyze the geometry and suggest manual decompositions of the geometry into sweepable sub-volumes, see [89].

Feature-based methods

Feature-based decomposition algorithms try to decompose the initial geometry into sub-domains by using its features. Typically, these features consist on paths of edges. Then, using these paths of edges, virtual surfaces are constructed in order to decompose the geometry. The process is iterated until each of the sub-domains can be meshed. See [90–93] for additional details.

Other kind of decomposition methods try to compute a poly-cube of the initial model and then, decompose the poly-cube in order to generate an hexahedral mesh. Finally, the mesh of the poly-cube is mapped to the initial geometry. The main problem of this approach is the construction of the poly-cube that approximates the initial geometry, since it has a great impact on the quality of the final mesh. Several techniques have been proposed in [94–96].

The fun-sheet matching method, see [97], obtains a block decomposition of the geometry by means of hexahedral sheets extraction. First, a tetrahedral mesh is generated and the is converted into an hexahedral mesh by partitioning each tetrahedron into four hexahedra. Additional layers of hexahedral elements are placed at the boundary to ensure that the boundary is properly captured. Then, the algorithm iteratively removes hexahedral sheets until a block decomposition is computed.

Local Dual Contributions

The objective of this method [98,99] is to obtain a block decomposition of a geometry. Therefore, the final mesh will be obtained by applying, for instance, transfinite interpolation and ensuring mesh compatibility through block faces. The first characteristic of this method is that to generate the block decomposition, it builds a geometrical and topological description of its dual. Therefore, it can also be classified as a dual method. The second characteristic is that it builds this dual representation without a prescribed

quadrilateral surface mesh. Thus, it can also be classified as an unconstrained method.

The method starts from a coarse tetrahedral discretization of the domain. Then, each mesh tetrahedron is split into four hexahedra, see [100]. From this tetrahedral mesh, an initial dual description of the block decomposition is build. However, this dual description does not reproduce the features of the boundary in the mesh. To overcome this drawback, the method applies a hierarchical scheme and a set of matching rules that explicitly insert descriptions of dual surfaces and handle their intersection. That is, it generates a dual of the block mesh with intersections of the proper multiplicity, without gaps and that respects the boundary features of the domain. Then this dual description is *dualized* to obtain a topological decomposition of the domain in blocks (ultra-coarse quadrilaterals or hexahedra). Thus, the Local Dual Contribution algorithm can be decomposed in three steps: *i*) the generation of a coarse reference mesh composed by tetrahedral elements; *ii*) the insertion of dual surfaces by addition of local dual contributions; and *iii*) the dual regions are ized to obtained the final block mesh.

Cross-field based methods

A new family of methods to generate quadrilateral meshes on surfaces have been developed by the computer graphics community [101–106]. These methods initially sought a globally smooth parameterization of the surface that does not require any previous partition of the geometry. These parameterizations are derived from a directionally field (also called cross-field). For instance, this directionally fields can be aligned with the principal directions of curvature . These parameterizations provide well shaped quadrilateral regions that are almost structured (most of the inner nodes have valence four). Thus, a high-quality quadrilateral mesh can be generated. However, cross-fields based on main curvatures may present discontinuities and oscillations. Therefore, the cross-field is processed in order to obtain a smooth representation.

In reference [107], the cross-field is used to compute an automatic partitioning of an arbitrary planar geometry. Similar to sub-mapping method, this method generates a geometry decomposition into four-sided regions with curvilinear edges. However, it can deal with any arbitrary geometry (in contrast to block geometries) and are based on solving a PDE (in contrast to solving a linear integer problem). It is important to point out that in [107] the cross-field is computed by imposing that it has to be aligned with the boundary of the domain (to generate high-quality meshes on the boundary) and is computed in the interior by solving a PDE. Cross-fields are also used in Reference [108] to compute the decomposition of planar geometries. However in this case the cross-field is propagated towards the interior of the domain using a fast marching algorithm [109]. It is worth to notice that Reference [108] extends the method to non-planar surfaces.

Cross-field based methods have been extended to generate hexahedral meshes for arbitrary geometries. The *CubeCover* method [110] extends to volumetric geometries

the ideas introduced in [103] for surfaces. In this case the cross-field is prescribed on the boundary (aligned with the boundary) and extended to the inner part. The final hexehedral mesh is generated from this cross-field. Similar approaches are used in [111–113], although the procedure to compute the inner cross-field is different. Although cross-field based methods are a promising alternative for hexahedral mesh generation, they still exhibit some limitations. On one hand, they do not provide a theoretical guarantees that they can always generate a valid all-hex mesh. Nevertheless, they can generate full hexahedral meshes in some complex geometries. On the other hand, further research is needed in order to deal with non-constant element size. Finally, it is important to point out that cross-field based methods are unconstrained methods since they mesh the geometry from an unmeshed boundary.

4.2 Inside-Outside

Grid-based methods

The grid-based methods are one of the most successful approaches to hexahedral meshing [114, 115]. The main reason is that the approach is highly automated and that it can be applied to any geometry. The mesh is generated from the inside to the outside of the volume. First, a grid mesh is generated in the inner part of the geometry. The initial mesh can be generated using a Cartesian grid or using an octree to adapt the inner mesh to the prescribed element size. Then, this mesh is projected or adapted in order to capture the boundary of the domain. Since the mesh is generated in an inside-outside manner, high-quality elements are generated in the interior. However, the element quality near the boundary may be lower. In addition, the final mesh may not correctly capture the geometric features present in the model.

References [116, 117] describe an algorithm to extract adaptive quadrilateral and hexahedral meshes directly from volumetric data. The main advantage of this method is that it is not necessary to construct a geometric model from the volumetric data. Moreover, in [118], a new grid-based method to generate hex-meshes for multi-material domains, is presented.

Reference [119] proposes an algorithm to embed topological features in a hexahedral mesh in order to correctly capture the geometrical features present in the volume. The topological features of the geometry are included in a Cartesian mesh without the need to enrich the initial mesh. In this way, the grid-based methods can correctly mesh geometries that present sharp features. Reference [120] presents a method to correctly capture the boundary of assembly models using grid-based methods. Finally, it is important to point out that special efforts have been focused on the development of new templates to adapt the inner mesh to the geometry boundaries [121, 122]

Receding Front

Two of the most used techniques to generate unstructured hexahedral meshes are the advancing front methods and the grid-based techniques. On the one hand, the grid-based methods are robust and fully automatic. In addition, these methods generate high-quality elements in the inner part of the mesh. These advantages are possible because the mesh is generated using an inside-outside approach. However, the grid-based methods generate low quality hexahedra near the boundary and the final mesh depends on the spatial orientation of the domain. These drawbacks appear because the inner mesh does not have layers of hexahedra that progressively adapt the inner mesh to the boundary shape of the domain.

On the other hand, the advancing front methods, see Section 4.3, generate highquality meshes near the boundary that do not depend on the orientation of the object. This is possible because the elements are generated layer by layer following the shape of the boundary surface. However, the advancing front methods are less robust and automatic than the grid-based ones. As the fronts advance into the volume, they collide between them and the intersections have to be computed. In addition, the fronts define complex voids in the inner parts of the domain. Those voids are as difficult to mesh as the original geometry and, for this reason, they can not be automatically meshed.

The receding front method combines the advantages of the advancing front methods and the grid-based techniques, while avoiding its disadvantages. The method pre-computes a set of mesh fronts by combining the solution of two Eikonal equations. Then, the mesh is generated from the inside to the outside using an advancing front technique that uses the pre-computed fronts as a guide in order to avoid front collisions. The method has been applied to generate hexahedral meshes for exterior domains, see Figures 8(a) and 8(b). However, further research is needed to improve its robustness and to check its capabilities for arbitrary geometries.

The idea of advancing a front from the inner part of the geometry has been also used in [123] where the fronts are guided by a harmonic field. Similarly to the receding front method, further research is needed to extend it to more complex geometries with variable features near the boundary.

4.3 Outside-Inside

Advancing front techniques are the framework for the outside-inside approach to generate hexahedral meshes. These fronts are advanced according to geometrical properties, *primal methods*, or according to topological properties, *dual methods*.

4.3.1 Primal methods

Advancing front techniques for quadrilateral meshing was first introduced by the *paving method* in [41, 42]. Later, further improvements [44] and extensions to surface meshing [43] were proposed. The input of the method is a pre-meshed boundary.



Figure 8: Meshes generated using the receding front method for: (a) the exterior domain of a five-pointed star and; (b) the exterior domain of a plane.

Fronts are determined from this discretized boundary and are propagated (outsideinside approach) until they collide or small voids are left. Note that in this process the basic geometric operations are: node location, element creation and front management (seaming of elements, wedge and tuck insertion). In this case, special templates are used to obtain the final quadrilateral mesh. The paving method produces high-quality meshes with quadrilateral elements well-aligned with the geometry boundary.

The *plastering method* [124–126] is the natural extension of the paving method to generate hexahedral meshes for any arbitrary three-dimensional geometry. It starts from a quadrilateral surface mesh of the geometry and advances inwards the solid. As the fronts advance elements are created. Finally, the fronts collide or generate inner unmeshed voids. The plastering method generates high quality meshes at the boundary, however it fails to mesh these inner voids and therefore, hybrid meshes may be obtained as a result. In some way, it propagates the quadrilateral boundary mesh to the inner part of the geometry and yields to an over-constraint boundary mesh for the inner voids. To overcome this issue, the unconstrained plastering [27-29] was introduced. This method starts from an unmeshed boundary. Therefore, there is not a boundary mesh topology that has to be respected. Moreover, in the unconstrained plastering layers advance inward the geometry delaying the definition of the elements until they are completely defined by the intersection of the fronts. Unfortunately, hybrid meshes are also obtained in the case of complex geometries and therefore, further research is needed in order to achieve a full unstructured hexahedral mesh generator.

4.3.2 Dual methods

Dual Methods are based on the concept of the dual mesh. In Reference [19], the authors introduce the Spatial Twist Continuum (STC) to represent the dual of an all-

hexahedral mesh. The whisker weaving algorithm uses the STC to construct a hexahedral mesh, see [127]. The algorithm begins with a quadrilateral mesh of the boundary and then constructs hexahedral element connectivity advancing into the solid. Finally, when the dual mesh is constructed, the nodes are positioned in the physical space. However, since the algorithm only takes into account the topology of the mesh, and not the geometrical features, low-quality elements can be generated. Several extensions of the whisker weaving algorithm have been presented in the literature to overcome its original drawbacks and to improve the quality of the final mesh. In [128], the authors introduce different *weaving patterns* to improve the mesh quality. In addition, the new patterns are also able to modify the boundary quadrilateral mesh to eliminate intersecting dual curves. Moreover, in [129], the geometric features of the model are also taken into account, which further improves the mesh quality. However, for an arbitrary geometry, it is not yet possible to obtain a mesh without inverted elements. Other approaches of dual methods are the ones proposed in [130, 131]. These algorithms perform a decomposition of the geometry in the dual space. Therefore, they can also be considered as decomposition methods and included in Section 4.1. Although they are able to generate a dual mesh, they do not generate the primal mesh. Recently, the dual method presented in [130] has been extended to more complex geometries in [132].

5 Concluding Remarks

In this work we have presented a detailed survey of hexahedral mesh generation algorithms. Although there exist several attempts to develop a high-quality hexahedral mesh generation algorithm for arbitrary geometries, none of them completely fulfills all the requirements. For instance, the grid-based methods are able to generate hexahedral meshes for a wide range of geometries (from mechanical to bio-mediacal applications). However, they fail to deliver adequate discretizations for computational fluid dynamics or solid mechanics applications, where high-quality elements have to be aligned with the boundaries. For this kind of applications, the industry and commercial packages rely on assembly models to mesh each part separately while maintaining the compatibility between them. For large models, this approach still requires large and intensive human iteration. To overcome this drawback automatic decomposition methods are currently investigated. One of the most promising alternative are the cross-field based methods. These methods compute a decomposition of the geometry by solving a PDE and then, each part is meshed separately using a mapping algorithm. Although the final mesh is not currently adapted to a prescribed size function, the elements can be coarsened or refined using existing algorithms in order to obtain the desired element size. Although encouraging results have been obtained during the last years, several aspects of the method have to be revised and improved. For instance, a deeper analysis is needed in order to find a sufficient condition to obtain an all-hexahedral structure or decomposition.

Assembly models will be intensively used in industry in the near future. Therefore,

there will be an increasing demand of methods that allow generating high-quality meshes for these models. Since assembly models can be meshed in several steps, it is important to analyze the topological constraints that allow obtaining conformal hexahedral meshes at the interface between two previously meshed sub-volumes.

Finally, it is worth to notice that meshing algorithms are closely related to the computational methods that use these meshes. Therefore, new trends in computational mechanics community will pose new challenges and will open new possibilities. For instance, during the last decade high-order methods have attracted interest in the FEM community. The standard strategy to generate high-order meshes is an *a posteriori* approach. That is, first, a linear mesh is generated according to the requirements of the numerical simulation, and then it is curved to match the boundary geometry. In this process inverted element may appear that will preclude the use of the high-order mesh in a subsequent analysis. Thus, robust untangling and smoothing techniques will be needed to improve these meshes. Moreover, new approaches can be devised that improve the performance of current high-order mesh generation technology.

Similarly, iso-geometric analysis suggests new applications and opens frontiers to our community. The close relationship between CAD representation and volume decomposition becomes stronger in this scenario. In particular, automatic and robust volume parameterization and volume decomposition algorithms are needed.

Acknowledgements

This research was partially supported by CONACYT-SENER ("Fondo Sectorial CONA-CYT SENER HIDROCARBUROS", grand contract 163723).

References

- R. Lohner, K. Morgan, and J. Peraire. Finite element methods for high speed flows. In *Proceedings of the 7th Computational Fluid Dynamics Conference*, 1985.
- [2] J. Peraire, J. Peiro, L. Formaggia, K. Morgan, and Zienkiewicz O. Finite element Euler computations in three dimensions. *International Journal for Numerical Methods in Engineering*, 26:2135–2159, 1988.
- [3] R. Lohner and P. Parikh. Generation of three-dimensional unstructured grids by the advancing-front method. *International Journal for Numerical Methods in Fluids*, 8:1135–1149, 1988.
- [4] T. J. Baker. Three dimensional mesh generation by triangulation of arbitrary point sets. In *Proceedings of the 8th Computational Fluid Dynamics Conference*, 1987.

- [5] P. L. George, F. Hecht, and E. Saltel. Constraint of the boundary and automatic mesh generation. *Numerical Grid Generation in Computational Fluid Mechanics*, 1988.
- [6] M. Shephard and M. Georges. Automatic three-dimensional mesh generation by the finite octree technique. *International Journal for Numerical Methods in Engineering*, 32:709–749, 1991.
- [7] M. Yerry and M. Shephard. Automatic three-dimensional mesh generation by the modified-octree technique. *International Journal for Numerical Methods in Engineering*, 20:1965–1990, 1984.
- [8] J. F. Thompson, B. K. Soni, and N. P. Weatherill. *Handbook of grid generation*. CRC Press, 1999.
- [9] P. L. George and H. Borouchaki. *Delaunay triangulation and meshing: application to finite elements*. Hermes Paris, 1998.
- [10] P.J. Frey and P.L. George. Mesh Generation: Application to Finite Elements. Wiley, 2008.
- [11] T. D. Blacker. Automated conformal hexahedral meshing constraints, challenges and opportunities. *Engineering with Computers*, 17(3):201–210, 2001.
- [12] S. E. Benzley, E. Perry, K. Merkley, B. Clark, and G. Sjaardema. A comparison of all-hexahedral and all-tetrahedral finite element meshes for elastic and elastoplastic analysis. In *Proceedings of the 4th International Meshing Roundtable*, 1995.
- [13] A. O. Cifuentes and A. Kalbag. A performance study of tetrahedral and hexahedral elements in 3-d finite element structural analysis. *Finite Elements in Analysis and Design*, 12:313–318, 1992.
- [14] V. Weingarten. The controversy over hex or tet meshing. *Machine design*, 66(8):74–76, 1994.
- [15] A. Huerta, A. Angeloski, X. Roca, and J. Peraire. Efficiency of high-order elements for continuous and discontinuous Galerkin methods. *Int. J. Numer. Methods Eng.*, 96(9):529–560, 2013.
- [16] A. T. Patera. A spectral element method for fluid dynamics-laminar flow in a channel expansion. *Journal of Computational Physics*, 54(3):468–488, 1984.
- [17] M. Halpern. Industrial requirements and practices in finite element meshing: a survey of trends. In *6th International Meshing Roundtable*, pages 399–411, 1997.

- [18] S. A. Mitchell. A characterization of the quadrilateral meshes of a surface which admit a compatible hexahedral mesh of the enclosed volume. In Proceedings of the 13th Annual Symposium on Theoretical Aspects of Computer Science, 1996.
- [19] P. Murdoch, S. E. Benzley, T. D. Blacker, and S. A. Mitchell. The spatial twist continuum: A connectivity based method for representing all-hexahedral finite element meshes. *Finite Elements in Analysis and Design*, 28(2):137–149, 1997.
- [20] T. J. Tautges and S. E. Knoop. Topology modification of hexahedral meshes using atomic dual-based operations. In *Proceedings 12th International Meshing Roundtable*, pages 415–423, 2003.
- [21] F. Ledoux and J. Shepherd. Topological and geometrical properties of hexahedral meshes. *Engineering with Computers*, 26(4):419–432, 2010.
- [22] S. J. Owen. A survey of unstructured mesh generation. In *Proceedings of the 7th International Meshing Roundtable*, pages 239–267, 1998.
- [23] T. J. Tautges. The generation of hexahedral meshes for assembly geometry: survey and progress. *International Journal for Numerical Methods in Engineering*, 50(12):2617–2642, 2001.
- [24] T. J. Baker. Mesh generation: art or science? Progress in Aerospace Sciences, 41(1):29–63, 2005.
- [25] K. Shimada. Current trends and issues in automatic mesh generation. *Computer-Aided Design and Applications*, 3(6):741–750, 2006.
- [26] J. F. Shepherd. *Topologic and geometric constraint-based hexahedral mesh generation*. PhD thesis, The University of Utah, 2007.
- [27] M.L. Staten, S. J. Owen, and T. D. Blacker. Unconstrained paving & plastering: A new idea for all hexahedral mesh generation. In *Proceedings of the 14th International Meshing Roundtable*, 2005.
- [28] M.L. Staten, R. A. Kerr, S. J. Owen, and T. D. Blacker. Unconstrained paving and plastering: Progress update. In *Proceedings of the 15th International Meshing Roundtable*, pages 469–486. Springer, 2006.
- [29] M.L. Staten, R. A. Kerr, S. J. Owen, T. D. Blacker, M. Stupazzini, and K. Shimada. Unconstrained plastering – Hexahedral mesh generation via advancingfront geometry decomposition. *International Journal for Numerical Methods in Engineering*, 81(5):135–171, 2010.
- [30] M. Stephenson and T. D. Blacker. Using Conjoint Meshing Primitives to Generate Quadrilateral and Hexahedral Elements in Irregular Regions. Technical report, ASME G0502B, 1989.

- [31] W. Gordon and A. Hall. Construction of curvilinear coordinate systems and applications to mesh generation. *International Journal for Numerical Methods in Engineering*, 1973.
- [32] W. A. Cook and W. R. Oakes. Mapping methods for generating threedimensional meshes. *Computers In Mechanical Engineering*, CIME Research Supplement:67–72, 1982.
- [33] T. Shih, R. T. Bailey, H. L. Nguyen, and R. J Roelke. Algebraic grid generation for complex geometries. *International journal for numerical methods in fluids*, 13(1):1–31, 1991.
- [34] P. M. Knupp and S. Steinberg. *Fundamentals of grid generation*, volume 1. CRC press Boca Raton, 1993.
- [35] C. C. L. Sells. Plane subcritical flow past a lifting aerofoil. Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences, 308(1494):377–401, 1969.
- [36] J. South and A. Jameson. Relaxation solutions for inviscid axisymmetric transonic flow over blunt or pointed objects. In AIAA Computational Fluid Dynamics Conference, 1973.
- [37] A. M. Winslow. Numerical solution of the quasilinear poisson equation in a nonuniform triangle mesh. *Journal of computational physics*, 1(2):149–172, 1966.
- [38] J. F. Thompson, F. C. Thames, and C. W. Mastin. Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies. *Journal of Computational Physics*, 15(3):299–319, 1974.
- [39] J. F. Thompson, Z. Warsi, and C. W. Mastin. *Numerical grid generation: foundations and applications*, volume 38. North-holland Amsterdam, 1985.
- [40] D.R. White and T. J. Tautges. Automatic scheme selection for toolkit hex meshing. *International Journal for Numerical Methods in Engineering*, 49(1-2):127–144, 2000.
- [41] T. D. Blacker and M. B. Stephenson. Paving: A new approach to automated quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering*, 32(4):811–847, 1991.
- [42] T. D. Blacker, M. B. Stephenson, and S. A. Canann. Analysis automation with paving: a new quadrilateral meshing technique. *Advances in engineering soft*ware and workstations, 13(5):332–337, 1991.

- [43] R. J. Cass, S.E. Benzley, R. J.Meyers, and T.D. Blacker. Generalized 3-d paving: An automated quadrilateral surface mesh generation algorithm. *International Journal for Numerical Methods in Engineering*, 39(9):1475–1489, 1996.
- [44] D. R. White and P. Kinney. Redesign of the paving algorithm: Robustness enhancements through element by element meshing. In *Proceedings 6th International Meshing Roundtable*, pages 323–335, 1997.
- [45] J. Sarrate and A. Huerta. Efficient unstructured quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering*, 49:1327–1350, 2000.
- [46] J. Sarrate and A. Huerta. Automatic mesh generation of nonstructured quadrilateral meshes over curved surfaces in \mathbb{R}^3 . In *Proceedings of the 3rd European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS*, Barcelona, Spain, 2000.
- [47] R. Haber, M. S. Shephard, J. F. Abel, R. H. Gallagher, and D. P. Greenberg. A general two-dimensional, graphical finite-element preprocessor utilizing discrete transfinite mappings. *International Journal for Numerical Methods in Engineering*, 17(7):1015–1044, 1981.
- [48] P. M. Knupp. Next-generation sweep tool: A method for generation all-hex meshes on two-and-one-half dimensional geometries. In *Proceedings of the 7th International Meshing Roundtable*, 1998.
- [49] X. Roca, J. Sarrate, and A. Huerta. Mesh projection between parametric surfaces. *Communications in Numerical Methods in Engineering*, 22:591–603, 2006.
- [50] T. D. Blacker. The cooper tool. In *Proceedings of the 5th International Meshing Roundtable*, 1996.
- [51] D. R. White, S. Saigal, and S. J. Owen. CCSweep: automatic decomposition of multi-sweep volumes. *Engineering with Computers*, 20:222–236, 2004.
- [52] M. L. Staten, S. A. Canann, and S. J. Owen. BMSweep: locating interior nodes during sweeping. *Engineering with Computers*, 15:212–218, 1999.
- [53] S. Cai and T. J. Tautges. Robust one-to-one sweeping with harmonic s-t mappings and cages. In *Proceedings of the 22th International Meshing Roundtable*, 2013.
- [54] X. Roca and J. Sarrate. An automatic and general least-squares projection procedure for sweep meshing. *Engineering with Computers*, 26:391–406, 2010.

- [55] X. Roca and J. Sarrate. Least-squares approximation of affine mappings for sweep mesh generation: functional analysis and applications. *Engineering with Computers*, 29:1–15, 2013.
- [56] M. A. Scott, M. N. Earp, S. E. Benzley, and M. B. Stephenson. Adaptive sweeping techniques. In *Proceedings of the 14th International Meshing Roundtable*, pages 417–432. Springer, 2005.
- [57] S. H. Lo. Generating quadrilateral elements on plane and over curved surfaces. *Computers & structures*, 31(3):421–426, 1989.
- [58] C. K. Lee and S. H. Lo. A new scheme for the generation of a graded quadrilateral mesh. *Computers & structures*, 52(5):847–857, 1994.
- [59] E. Rank, M. Schweingruber, and M. Sommer. Adaptive mesh generation and transformation of triangular to quadrilateral meshes. *Communications in Numerical Methods in Engineering*, 9(2):121–129, 1993.
- [60] S. J. Owen, M.L. Staten, S. A. Canann, and S. Saigal. Q-morph: an indirect approach to advancing front quad meshing. *International Journal for Numerical Methods in Engineering*, 44(9):1317–1340, 1999.
- [61] M. S. Ebeida, K. Karamete, E. Mestreau, and S. Dey. Q-tran: A new approach to transform triangular meshes into quadrilateral meshes locally. In *Proceedings of the 19th International Meshing Roundtable*, pages 23–34. Springer, 2010.
- [62] J. F. Remacle, J. Lambrechts, B. Seny, E. L. Marchandise, A. Johnen, and C. Geuzainet. Blossom-quad: A non-uniform quadrilateral mesh generator using a minimum-cost perfect-matching algorithm. *International Journal for Numerical Methods in Engineering*, 89(9):1102–1119, 2012.
- [63] T. C. Baudouin, J. F. Remacle, E. L. Marchandise, and J. Lambrechts. l_p lloyd's energy minimization for quadrilateral surface mesh generation. In *Proceedings* of the 20th International Meshing Roundtable, pages 473–487. Springer, 2012.
- [64] T. C. Baudouin, J. F. Remacle, E. L. Marchandise, J. Lambrechts, and F. Henrotte. Lloyd's energy minimization in the l_p norm for quadrilateral surface mesh generation. *Engineering with Computers*, 30(1):97–110, 2014.
- [65] C. S. Verma and T. J. Tautges. Jaal: Engineering a high quality all-quadrilateral mesh generator. In *Proceedings of the 20th International Meshing Roundtable*, pages 511–530. Springer, 2012.
- [66] S. J. Owen and S. Saigal. H-morph: an indirect approach to advancing front hex meshing. *International Journal for Numerical Methods in Engineering*, 49:289–312, 2000.

- [67] M.L. Staten, J.F. Shepherd, and K. Shimada. Mesh matching–creating conforming interfaces between hexahedral meshes. In *Proceedings of the 18th International Meshing Roundtable*, pages 467–484, 2008.
- [68] M. L. Staten, J. F. Shepherd, F. Ledoux, and K. Shimada. Hexahedral mesh matching: Converting non-conforming hexahedral-to-hexahedral interfaces into conforming interfaces. *International journal for numerical methods in engineering*, 82(12):1475–1509, 2010.
- [69] J. Chen, H. Zhu, S. Gao, and H. Wu. An improved hexahedral mesh matching algorithm. In *Proceedings of the 22nd International Meshing Roundtable*, pages 183–201. Springer, 2014.
- [70] D. R. White. Automatic quadrilateral and hexahedral meshing of pseudocartesian geometries using virtual subdivision. Master's thesis, Brigham Young University, 1996.
- [71] M. Whiteley, D. White, S. E. Benzley, and T. Blacker. Two and three quarter dimensional meshing facilitators. *Engineering with Computers*, 12:144–154, 1996.
- [72] E. Ruiz-Gironés and J. Sarrate. Discretización de superficies mútiplemenete conexas mediante submapping. *Revista Internacional de Métodos Numéricos* para Cálculo y Diseño en Ingeniería, 2007.
- [73] E. Ruiz-Gironés and J. Sarrate. Generation of structured meshes in multiply connected surfaces using submapping. *Advances in Engineering Software*, 41:379–387, 2010.
- [74] E. Ruiz-Gironés and J. Sarrate. Generation of structured hexahedral meshes in volumes with holes. *Finite Elements in Analysis and Design*, 46:792–804, 2010.
- [75] R. Chen and P. Xi. A digraph-based hexahedral meshing method for coupled quasi-polycubes. *Computer Methods in Applied Mechanics and Engineering*, 268:18–39, 2014.
- [76] M. A. Scott, S. E. Benzley, and S. J. Owen. Improved many-to-one sweeping. *International Journal for Numerical Methods in Engineering*, 65:332–348, 2006.
- [77] M. Lai, S. Benzley, and D. White. Automated hexahedral mesh generation by generalized multiple source to multiple target sweeping. *International Journal for Numerical Methods in Engineering*, 49:261–275, 2000.
- [78] J. Shepherd, S. E. Benzley, and S. Mitchell. Interval assignment for volumes with holes. *International Journal for Numerical Methods in Engineering*, 49:277–288, 2000.

- [79] E. Ruiz-Gironés, X. Roca, and J. Sarrate. Using a computational domain and a three-stage node location procedure for multi-sweeping algorithms. *Advances* in Engineering Software, 42(9):700–713, 2011.
- [80] K. Miyoshi and T. Blacker. Hexahedral mesh generation using multi-axis cooper algorithm. In *Proceedings of the 9th International Meshing Roundtable*, 2000.
- [81] G. Xiong, S. Musuvathy, and T. Fang. Automated structured all-quadrilateral and hexahedral meshing of tubular surfaces. In *Proceedings of the 21st International Meshing Roundtable*, pages 103–120. Springer, 2013.
- [82] E Marchandise, C Geuzaine, and JF Remacle. Cardiovascular and lung mesh generation based on centerlines. *International journal for numerical methods in biomedical engineering*, 29(6):665–682, 2013.
- [83] C. G. Armstrong, D. J. Robinson, R. M. McKeag, T. S. Li, S. J. Bridgett, R. J. Donaghy, and C. A. McGleenan. Medials for meshing and more. In *Proceedings of the 4th International Meshing Roundtable*, 1995.
- [84] M.A. Price, C.G. Armstrong, and M.A. Sabin. Hexahedral mesh generation by medial surface subdivision: Part I. Solids with convex edges. *International Journal for Numerical Methods in Engineering*, 38:3335–3359, 1995.
- [85] M.A. Price and C.G. Armstrong. Hexahedral mesh generation by medial surface subdivision: Part II. solids with flat and concave edges. *International Journal for Numerical Methods in Engineering*, 40(1):111–136, 1997.
- [86] A. Sheffer, M. Etzion, A. Rappoport, and M. Bercovier. Hexahedral mesh generation using the embedded Voronoi graph. *Engineering with Computers*, 15(3):248–262, 1999.
- [87] A. Sheffer and M. Bercovier. Hexahedral meshing of non-linear volumes using Voronoi faces and edges. *International Journal for Numerical Methods in Engineering*, 49(1-2):329–351, 2000.
- [88] W. R. Quadros. Laytracks3d: Mesh generator for general assembly models using medial axis transform. In *Research note presented at the 22nd International Meshing Roundtable*, 2013.
- [89] J. H. Lu, I. Song, W. R. Quadros, and K. Shimada. Geometric reasoning in sketch-based volumetric decomposition framework for hexahedral meshing. In *Proceedings of the 21st International Meshing Roundtable*, pages 297–314, 2013.
- [90] H. Sakurai. Volume decomposition and feature recognition: Part 1—polyhedral objects. *Computer-Aided Design*, 27(11):833–843, 1995.

- [91] H. Sakurai and P. Dave. Volume decomposition and feature recognition, part ii: curved objects. *Computer-Aided Design*, 28(6):519–537, 1996.
- [92] R. Sonthi, G. Kunjur, and R. Gadh. Shape feature determination usiang the curvature region representation. In *Proceedings of the fourth ACM symposium* on Solid modeling and applications, pages 285–296. ACM, 1997.
- [93] Y. Lu, R. Gadh, and T. J. Tautges. Feature based hex meshing methodology: feature recognition and volume decomposition. *Computer-Aided Design*, 33(3):221–232, 2001.
- [94] Y. He, H. Wang, C. Fu, and H. Qin. A divide-and-conquer approach for automatic polycube map construction. *Computers & Graphics*, 33(3):369–380, 2009.
- [95] J. Gregson, A. Sheffer, and E. Zhang. All-hex mesh generation via volumetric polycube deformation. In *Computer graphics forum*, volume 30, pages 1407– 1416. Wiley Online Library, 2011.
- [96] M. Livesu, N. Vining, A. Sheffer, J. Gregson, and R. Scateni. Polycut: monotone graph-cuts for polycube base-complex construction. ACM Transactions on Graphics (TOG), 32(6):171, 2013.
- [97] N. Kowalski, F. Ledoux, M. L. Staten, and S. J. Owen. Fun sheet matching automatic generation of block-structured hexahedral mesh using fundamental sheets. In *10th usnccm*, 2009.
- [98] X. Roca and J. Sarrate. Local dual contributions on simplices: a tool for block meshing. In *Proceedings of the 17th International Meshing Roundtable*, pages 513–531. Springer, 2008.
- [99] X. Roca and J. Sarrate. Local dual contributions: Representing dual surfaces for block meshing. *International Journal for Numerical Methods in Engineering*, 83:709–740, 2010.
- [100] G. F. Carey. Hexing the tet. Communications in Numerical Methods in Engineering, 18(3):223–227, 2002.
- [101] G. Bunin. A continuum theory for unstructured mesh generation in two dimensions. *Computer Aided Geometric Design*, 25(1):14–40, 2008.
- [102] N. Ray, W. C. Li, B. Lévy, A. Sheffer, and P. Alliez. Periodic global parameterization. *ACM Transactions on Graphics*, 2006.
- [103] F. Kälberer, M. Nieser, and K. Polthier. Quadcover-surface parameterization using branched coverings. In *Computer Graphics Forum*, volume 26, pages 375–384. Wiley Online Library, 2007.

- [104] J. Palacios and E. Zhang. Rotational symmetry field design on surfaces. In *ACM Transactions on Graphics (TOG)*, volume 26, page 55. ACM, 2007.
- [105] N. Ray, B. Vallet, L. Alonso, and B. Lévy. Geometry-aware direction field processing. ACM Transactions on Graphics (TOG), 29(1):1, 2009.
- [106] D. Bommes, H. Zimmer, and L. Kobbelt. Mixed-integer quadrangulation. In *ACM Transactions on Graphics (TOG)*, volume 28, page 77. ACM, 2009.
- [107] N. Kowalski, F. Ledoux, and P. J. Frey. A pde based approach to multidomain partitioning and quadrilateral meshing. In *Proceedings of the 21st International Meshing Roundtable*, pages 137–154. Springer, 2013.
- [108] H. Fogg, C. Armstrong, and T. Robinson. Multi-block decomposition using cross-fields. In VI International Conference on Adaptive Modeling and Simulation, ADMOS, 2013.
- [109] J. A. Sethian. Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science, volume 3. Cambridge university press, 1999.
- [110] M. Nieser, U. Reitebuch, and K. Polthier. Cubecover–parameterization of 3d volumes. In *Computer Graphics Forum*, volume 30, pages 1397–1406. Wiley Online Library, 2011.
- [111] J. Huang, Y. Tong, H. Wei, and H. Bao. Boundary aligned smooth 3d crossframe field. In ACM Transactions on Graphics (TOG), volume 30, page 143. ACM, 2011.
- [112] J. Huang, T. Jiang, Y. Wang, Y. Tong, and H. Bao. Automatic frame field guided hexahedral mesh generation. Technical report, Technical Report MSU-CSE-12-9, Department of Computer Science, Michigan State University, East Lansing, Michigan, 2012.
- [113] Y. Li, Y. Liu, W. Xu, W. Wang, and B. Guo. All-hex meshing using singularityrestricted field. *ACM Transactions on Graphics (TOG)*, 31(6):177, 2012.
- [114] R Schneiders and R Bünten. Automatic generation of hexahedral finite element meshes. *Computer Aided Geometric Design*, 12(7):693–707, 1995.
- [115] R. Schneiders, R. Schindler, and F. Weiler. Octree-based generation of hexahedral element meshes. In *Proceedings of the 5th International Meshing Roundtable*, 1996.
- [116] Y Zhang, C. L. Bajaj, and BS Sohn. 3D finite element meshing from imaging data. *Computer Methods in Applied Mechanics and Engineering*, 194(48-49):5083–5106, 2005.

- [117] Y Zhang and C. L. Bajaj. Adaptive and quality quadrilateral/hexahedral meshing from volumetric data. *Computer Methods in Applied Mechanics and Engineering*, 195(9-12):942–960, 2006.
- [118] Y. Zhang, T. J. Hughes, and C. L. Bajaj. An automatic 3d mesh generation method for domains with multiple materials. *Computer methods in applied mechanics and engineering*, 199(5):405–415, 2010.
- [119] S. J. Owen and J.F. Shepherd. Embedding Features in a Cartesian Grid. In *Proceedings of the 18th International Meshing Roundtable*, 2009.
- [120] J. Qian and Y. Zhang. Automatic unstructured all-hexahedral mesh generation from b-reps for non-manifold cad assemblies. *Engineering with Computers*, 28(4):345–359, 2012.
- [121] Y. Ito, A. M. Shih, and B. K. Soni. Octree-based reasonable-quality hexahedral mesh generation using a new set of refinement templates. *International Journal for Numerical Methods in Engineering*, 77(13):1809–1833, 2009.
- [122] J. Qian, Y. Zhang, W. Wang, A. C. Lewis, and M. A. Qidwai A. B. Geltmacher. Quality improvement of non-manifold hexahedral meshes for critical feature determination of microstructure materials. *International journal for numerical methods in engineering*, 82(11):1406–1423, 2010.
- [123] M. Li and R. Tong. All-hexahedral mesh generation via inside-out advancing front based on harmonic fields. *The Visual Computer*, 28(6-8):839–847, 2012.
- [124] S. A. Canann. Plastering: A new approach to automated 3-d hexahedral mesh generation. *American Institute of Aeronautics and Astronics*, 1992.
- [125] S. A. Canann. Plastering and Optismoothing: New Approaches to Automated, 3-D Hexahedral Mesh Generation and Mesh Smoothing. PhD thesis, Brigham Young University. Department of Civil Engineering., 1991.
- [126] T. D. Blacker and R. J. Meyers. Seams and wedges in Plastering: a 3-D hexahedral mesh generation algorithm. *Engineering with Computers*, 9(2):83–93, 1993.
- [127] T. J. Tautges, T. Blacker, and S. A. Mitchell. The whisker weaving algorithm: a connectivity-based method for constructing all-hexahedral finite element meshes. *International Journal for Numerical Methods in Engineering*, 39:3327–3349, 1998.
- [128] N. T. Folwell and S. A. Mitchell. Reliable whisker weaving via curve contraction. *Engineering with computers*, 15(3):292–302, 1999.
- [129] F. Ledoux and J.C. Weill. An extension of the reliable Whisker weaving algorithm. In *Proceedings of the 4th International Meshing Roundtable*, 2008.

- [130] M. Müller-Hannemann. Hexahedral mesh generation by successive dual cycle elimination. *Engineering with Computers*, 15(3):269–279, 1999.
- [131] N. Calvo and S. Idelsohn. All-hexahedral element meshing: Generation of the dual mesh by recurrent subdivision. *Computer Methods in Applied Mechanics and Engineering*, 182:371–378, 2000.
- [132] M. Kremer, D. Bommes, I. Lim, and L. Kobbelt. Advanced automatic hexahedral mesh generation from surface quad meshes. In *Proceedings of the 22nd International Meshing Roundtable*, pages 147–164. Springer, 2014.