24th International Meshing Roundtable (IMR24)

Common themes in multi-block structured quad/hex mesh generation

Cecil G Armstrong[*a], Harold J Fogg[b], Christopher M Tierney[a] and Trevor T Robinson[a]

[a]School of Mechanical and Aerospace Engineering, Queens University, Ashby Building, Stranmill Road Belfast BT9 5AH, UK
[b]Siemens Industry Software Ltd, Francis House, 112 Hills Road, Cambridge CB2 1PH, UK

Abstract

The automatic generation of structured multi-block quadrilateral (quad) and hexahedral (hex) meshes has been researched for many years without definitive success. The core problem in quad / hex mesh generation is the placement of mesh singularities to give the desired mesh orientation and distribution [1]. It is argued herein that existing approaches (medial axis, paving / plastering, cross / frame fields) are actually alternative views of the same concept. Using the information provided by the different approaches provides additional insight into the problem.

Keywords: quad meshing; hex meshing; medial axis; paving; plastering; cross fields; frame fields

1. Introduction

This paper summarizes and draws insights from existing approaches to multi-block structured quad / hex mesh generation. A structured mesh is one in which all interior nodes and elements have identical connectivity. Conversely, unstructured meshes have nodes and elements with arbitrary connectivity. One essential difference between an analysis featuring structured and unstructured meshes is the data structure they are stored within. Some advantages of structured meshes are:

---

[*] Corresponding author. Tel.: +44 28 9097 4124; fax: +44 28 9097 4148.
   E-mail address: c.armstrong@qub.ac.uk

- Fully-unstructured meshes have system matrices in Finite Element Modelling (FEM) which are sparse but with an irregular distribution of non-zero entries. On the other hand fully-structured grids lead to banded non-zero entries. This is useful because by storing each band as a vector in a contiguous block of memory, optimized routines such as SPARSE BANDED BLAS techniques [2] can be employed that minimize indirect memory accessing and therefore exhibit much faster execution times.
- They are eminently suitable for multigrid acceleration methods, which are more or less obligatory for the efficient solution of Computational Fluid Dynamics (CFD) simulations [3].
- Structured grids are highly suitable for efficiently resolving anisotropic features of the solution field such as boundary layers or shock waves, provided that the grid is appropriately orientated [4].

When seeking structured meshes, it is typically associated with the goal of creating grids of quad / hex elements where every interior node is connected to four / eight elements respectively. Quad / hex elements are often the preferred element choice due to the following advantages:

- For large-deformation and non-linear elasto-plastic structural analyses they do not suffer from non-physical stiffness due to `mesh-locking' to the same extent as equivalent tet meshes [5] and are therefore more accurate. Mesh-locking is present where the degrees of freedom in an element are less than the incompressibility constraints imposed on the element.
- In dynamic structural mechanics analyses with explicit solvers, the minimum time step is controlled by the acoustic wave propagation rate in the material. For numerical stability it is essential that the calculation can progress across an element without skipping nodes. This forces a correlation between the maximum allowable time step and the characteristic length of mesh elements. Stability is also dependent on element shape quality in terms of skewness. Hence, the smallest element with the poorest quality limits the time step for the whole analysis, with negative consequences on the solution time and computational expense. Much larger time steps are feasible with hex meshes than with tetrahedral (tet) meshes with similar degrees of freedom and are therefore used if at all possible.
- Smaller discretization errors are reported in CFD analyses using quad and hex meshes than using triangular (tri) and tet meshes, even with isotropic elements [6].
- Their layered structure facilitates anisotropic stretching in the row or column directions without degrading the numerical quality of the mesh. In broad terms, this is because the numerical quality of elements is dependent on their corner angles. Corners do not deviate from pi/2 for a square stretched along one of its side directions, whereas the corner angles of a triangle, no matter how it is orientated, become adversely large or small when the triangle undergoes large amounts of stretching.

However, fully structured hex meshes have some substantial disadvantages:

- They are notoriously difficult to generate. While tri and tet generation technology is fully mature with numerous existing 'push button' algorithms capable of quickly generating meshes adhering to size targets on arbitrary geometries, the development of equivalent quad and hex meshing algorithms has proved to be much more challenging. So much so that automatic boundary conforming all-hex mesh generation has been dubbed the `holy grail' in the meshing community [7].
- They are inflexible. Local modifications are difficult to implement and necessitate careful changes to the mesh topology over large scales. Hence, they are ill-suited to mesh adaptation methodologies as the containment of mesh refinement to specific regions is difficult.

Since boundary conformity is an important requirement of the mesh, a regular grid topology is too restrictive in most cases. Wrapping a grid around a complex geometry can be a tough problem which would typically produce very distorted elements. Therefore it is essential that some degree of unstructured organization is introduced to the mesh. This leads to the concept of multi-block structured meshes, or sometimes just block-structured meshes, where the domain is divided into sub-regions, called blocks, each of which is occupied by a structured grid. However, the unstructured nature can be introduced between adjacent blocks which are joined at nodes / edges of irregular connectivity, referred to as singularities. An effective multi-block topology maximises the proportion of the mesh that is structured and allows the mesh to align with the main anisotropic features of the solution field. In principle, they can be generated on any arbitrary domain and provide a much needed compromise between the simplicity of structured grids and the flexibility and generality of unstructured meshes. The macro-unstructured-micro-structured hierarchal layout of a multi-block structured mesh also naturally lends itself to parallelisation where the execution of

calculations on local groups of blocks are performed on parallel processors, each enjoying the numerical efficiencies of the structured grids. Even though the creation of multi-block meshes is simpler than fully structured hex mesh generation, the automated identification and creation of the blocks is non-trivial and relies on defining where to place the mesh singularities and determining how these singularities interact when propagating inwards from the boundary. This paper aims to summarise the existing approaches to solving this problem and highlight some of the key similarities between them. In this paper the approaches considered are limited to those that attempt to solve this problem by working inwards from the boundary. Therefore methods such as octree-based decomposition are not considered.

2. Quad Meshes in 2D

Many of the key features of multi-block structured mesh generation are common to both quad and hex meshes, therefore quad meshes are described first for ease of explanation. The key features of semi-structured quadrilateral meshes are the mesh 'singularities', where the mesh grid is irregular. These are the nodes where more or less than the regular number (4 in the interior of a quad mesh) of elements meet, Fig. 1 shows isolated singularities embedded in regular grids of 2D elements. Singularities can also be placed on the boundary of the region, Fig. 2 and two or more simple singularities can be combined into a single degenerate singularity if they are close together, Fig. 3.
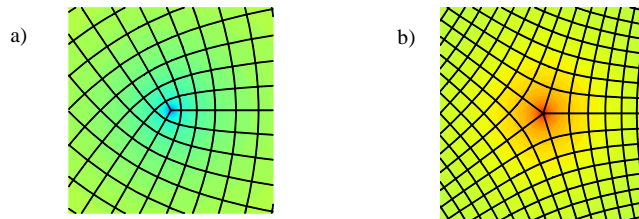

Fig. 1. (a) a negative singularity; (b) a positive singularity. The scalar field shown is a harmonic function representing a continuum description of the quad mesh. It relates to the exponential of isotropic element sizes.
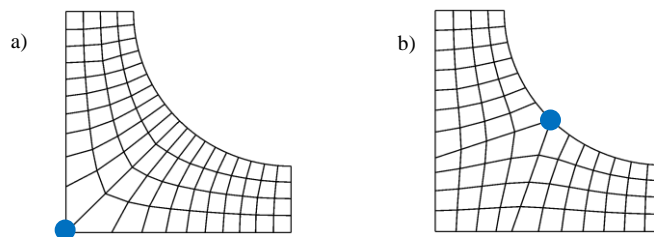

Fig. 2. Singularities on the boundary of a 2D region. The number of elements incident on the boundary is different than that implied by the geometry.

The minimum number of singularities to allow a multi-block mesh of a simply connected polygonal boundary can be found by counting the number and type of the corners in the boundary of the domain, though the placement of these singularities will make a huge difference to mesh quality. If the resulting block topology does not provide adequate control of mesh density then extra control can be provided by adding a pair of negative and positive mesh singularities, Fig. 4.
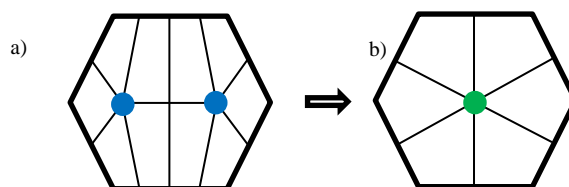

Fig. 3. Two 5-valent singularities (a) combined to give a single 6-valent singularity (b) in 2D

2.1. Placing mesh singularities in 2D

The industry standard method of creating multi-block meshes is to manually create a block topology using tools such as ICEM and then associate block edges and vertices with the appropriate entities of the real geometry. Whilst this can be effective in creating high quality meshes, and the meshing process can be used to suppress features below the scale of interest, it is a time-consuming process requiring a great deal of knowledge and expertise from the person creating the meshes.
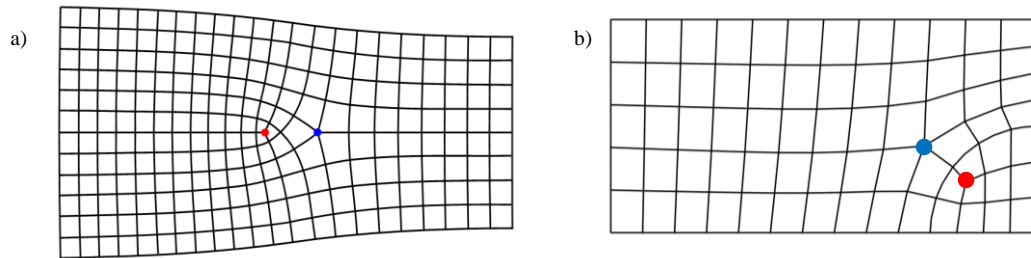

Fig. 4. Controlling mesh density with an extra singularity pair.

The medial axis (MA) method was used by Tam and Armstrong [8] to partition the region of interest into small sub-regions of simple shape, each of which contained at most one singularity. It employed a geometric property of shape called the medial axis, which is a skeleton or dimensionally reduced version of the shape. A characteristic of this approach was that singularities were placed as far as possible from the boundary, and that a good quality mesh was produced since the partitions subdividing the domain were between parts of the boundary in geometric proximity. An example of the resulting decompositions is shown in Fig. 5 a), where the partition is shown in yellow. The sub-regions containing singularities were meshed with a technique called midpoint subdivision [5][6], which divided the region into structured blocks around the singularity: a triangle is decomposed into three blocks, a pentagon into five etc. Fig. 5 a) was decomposed into two 5-sided prisms, each of which has a single positive singularity running through the wall thickness of the model. Using integer programming to control edge division numbers allows the different meshes in Fig. 5 a) and Fig. 5 b) to be generated from the same decomposition. Midpoint subdivision of 2D and 3D primitives has been available in Abaqus/CAE for some years.

a) 2D medial axis mesh in Abaqus/CAE
b) 2D medial axis mesh in Abaqus/CAE with different target mesh sizing
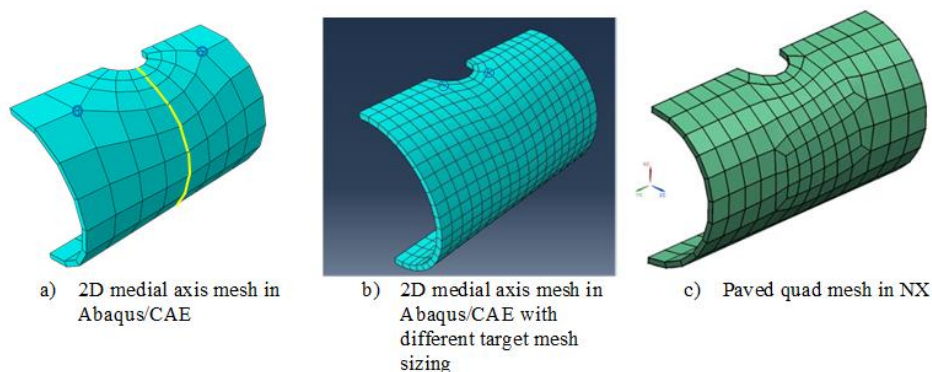c) Paved quad mesh in NX

Fig. 5. 2D extruded meshes. Note that medial axis meshes a) and b) have only two positive singularities, which are sufficient to mesh the 6-sided extrusion

The medial axis method can be viewed as a less restrictive form of the paving method [11]. In paving, quad elements are added to an advancing front from the boundary. Since paving only has a view of the local mesh geometry and topology, paving algorithms will generate many more mesh singularities than desired, with no control over where they are placed, Fig. 5 c).

The medial axis is where layers of elements advancing from the boundary collide. Since, unlike paving, the algorithm doesn't have to worry about the detailed mesh topology where the fronts collide (the MA is just used to identify regions which are opposite or adjacent to each other) the number of mesh singularities can be minimized. Note that the medial axis runs either parallel to the mesh flow or diagonally through it, Fig. 6.
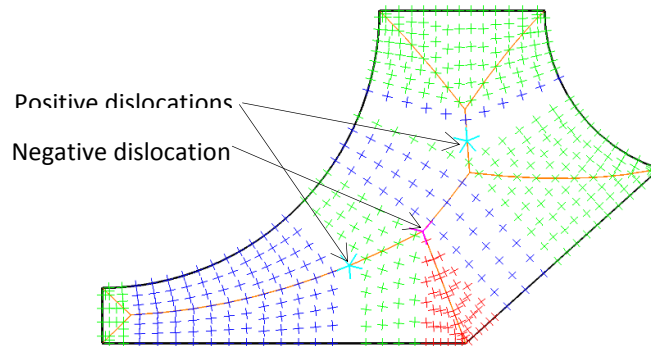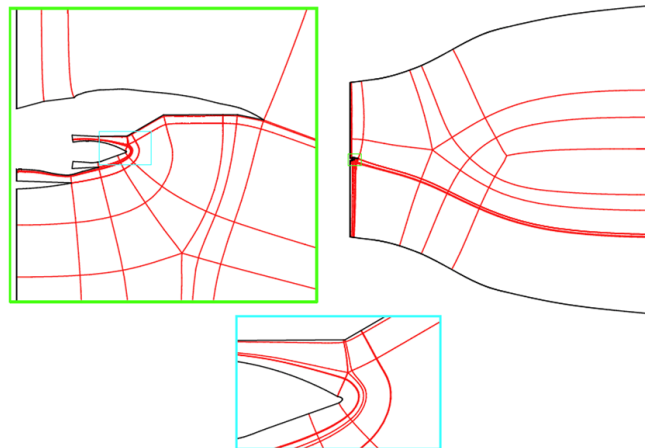
Positive dislocations

Negative dislocation

Fig. 6 Cross-field, medial axis and singularity placement based on where MA changes from parallel to mesh flow to diagonally through it

[1] Rolls-Royce test geometry, used with permission.

Fig. 7. Multiblock decomposition from [23]. Note the very different scales of the features in the model.

A method was presented in [12] for solving the mesh singularities and block topology of surfaces by a method that has similarities to the paving algorithm. The method proceeds in the same way by working inwards from boundaries but with the fundamental difference that only the element directionality represented by a 'cross field' is determined, instead of the complete mesh connectivity. By this method the cross field solution will vary continuously up to the medial axis where the advancing fronts collide and singularities of the cross-field, and hence the related quad mesh, potentially occur. Since this approach is based on a relatively simple algorithm for propagation of distance and orientation fields through a tri mesh of the domain, it has a number of advantages including: small features can be embedded within large domains; a more natural treatment of concavities is obtained, Fig. 8, where the mesh edges radiate naturally from the concavity rather than being constrained to follow lines joining the touching points of the inscribed circles associated with the medial axis, and the decomposition is no more difficult when Virtual Topology is present. A medial axis based decomposition method that implicitly uses a cross field was described [13] that allows for merging of nearby singularities and achieves effective treatment of concavities. Various other methods have also been used for generating cross fields on surfaces. Kowalski et al. [14] solve a diffusion problem to give an initial approximation of a unit vector representation and subsequently find a near-fit cross-field by iterative optimization. Methods that have been used on curved surfaces are principal curvature

direction approximation [15], [16], solving orthogonal Laplacian eigenfunctions of the surface [17] and constructing particular standing waves on the surface [18].
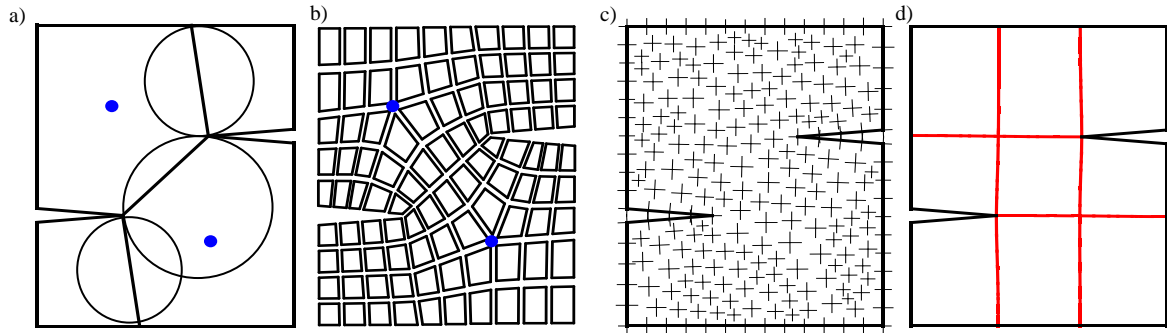

Fig. 8. Treatment of concavities using a) medial axis touching points and (right) cross fields.

Once the basic topology of the blocks (or the mesh-able primitives in the Tam and Armstrong case) has been found, the division numbers on the region edges need to be adjusted so that the numbers of elements on the edges are compatible. Using midpoint-subdivision primitives allows more freedom in the division number constraints than is available with a fixed block topology, Fig. 5 (a) and (b). The necessary constraints on edge division numbers can be enforced using standard routines for 'Integer Programming' (IP).

Note that using midpoint subdivision and IP allows adjustment of mesh size and orientation within a limited range without re-doing the singularity calculation. Fig. 5 (a) and (b) have the same decomposition into two 5-sided prisms: in (a) the singularities are connected by element edges but in (b), after a change in local target element size, the two singularities are not connected and they have been moved by the IP to optimize the element distribution. Of course the variation in target element size may eventually become so large that the singularity is attempting to move out of the primitive, Fig. 2, which could be interpreted as a signal that additional singularity pairs are needed. Using IP on an existing decomposition minimizes changes in mesh topology, which is important in optimization scenarios.

Fogg et al. [12], have shown that cross field methods for multi-block decomposition can be sensitive to both target element size and shape. It should therefore be possible to a) utilize solution error analysis to derive a mesh metric size and shape distribution based on the results of an initial analysis and then b) utilize that local size and shape information to produce a quad mesh and multi-block distribution that respects those metrics to adaptively refine the mesh geometry and topology to capture the solution to a prescribed error tolerance.

3. Hex meshes in 3D

The conventional approach to multi-blocking hex meshing is to manually define a block topology. Vertices, edges and faces of the block topology are then associated with real geometry. In ICEM a given element of the block topology can be associated with several geometric faces or edges, effectively accomplishing a virtual topology merging of these faces or edges. Standard block topology configurations like C- or O-type meshes around an interior hole can be used as macro elements in the block definition. An extensive set of block topology recipes is available in various packages, though it appears that these recipes usually have to be hand coded.

In 3D, mesh singularities form lines of element edges where more or less than 4 element edges meet. Excluding degenerate cases, positive and negative mesh line singularities can interact in only a very small number of ways, Fig. 9:

- Isolated mesh line singularities can travel through the domain, entering and exiting on the surface of the mesh. 2D extruded or swept meshes are an example of this class, where positive and negative mesh line singularities run along the sweep direction and are shown by the dashed red and blue lines in Fig. 9 b) and c) respectively. The sweep direction in thin sheet regions (where the lateral dimensions are large compared to the thickness) is between the top and bottom surface of the thin sheet. In long slender regions (where the

length of the region is large compared to the cross-section dimensions) the sweep direction is in the axial direction of the region. The mesh line singularities follow these sweep directions between source and target faces. In multi-sweep meshing, different combinations of line singularities enter and leave on different faces of the domain

- Isolated singularities can form a continuous loop, such as in a revolved 2D mesh
- Positive and negative singularities can combine. The rightmost two primitives in Fig. 9 show the simplest possible configurations, though additional degenerate combinations are possible. These were called the 'prime primitives' in Price et al. [19]
    - o Three positive and one negative mesh line singularities can combine. This corresponds to the midpoint subdivision algorithm applied to a 'cube-with-a-corner-chipped-off', Fig. 9 d).
    - o Four negative mesh line singularities can combine. This corresponds to the midpoint subdivision algorithm applied to a tetrahedral shape, Fig. 9 e). Note that this case can actually be further decomposed into two type c) negative singularities which pass each other in the interior of the tet volume, so it could be regarded as a degenerate combination.
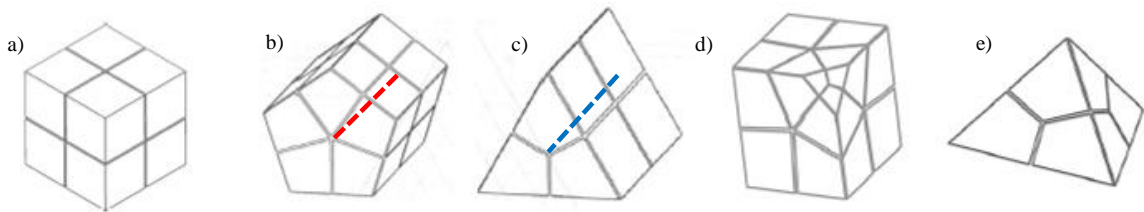

Fig. 9. Hex mesh singularities in 3D: a) Regular mesh; b) +ve singularity; c) -ve singularity; d) 3 +ve and 1 -ve singularities combine; e) 4 -ve singularities combine.

3.1. Placing mesh singularities in 3D

Plastering [20] is the 3D equivalent to paving, but the difficulties in resolving an appropriate mesh topology when the advancing element fronts collide are much more severe. This issue has been addressed through the concept of unconstrained plastering [21]. Also, in some variants of the algorithm the interior is filled with tetrahedral elements once a boundary layer of a few elements has been created.

The 3D equivalent of the medial axis method was described in Price et al [19] and Price and Armstrong [22]. The approach was to:

- mesh the interior of the medial surfaces with a 2D quad mesh and extrude in the thickness direction (green elements in Fig. 10 d))
- mesh the cross section in the vicinity of the medial edges and extrude along the axis of the medial edges (white elements in Fig. 10 c))
- mesh the vicinity of the medial vertices using midpoint subdivision (yellow elements in Fig. 10 b))
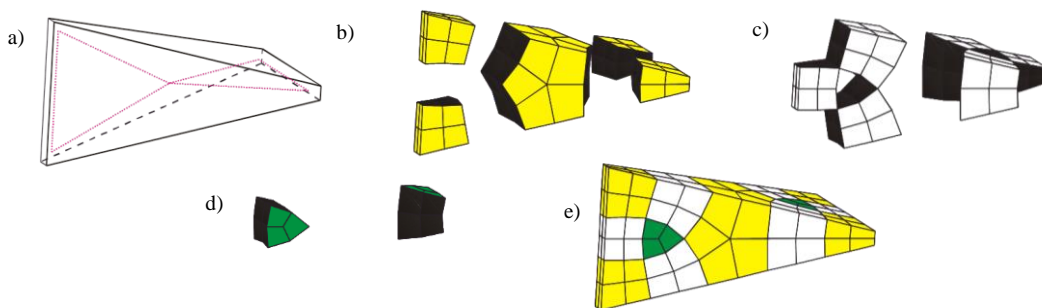
Fig. 10. Hex meshing using the medial axis in 3D: a) Medial surface representation; b) mid-point subdivision mesh; c) mesh extruded along medial edge; d) mesh extruded through thickness; e) assembled mesh

Whilst this approach can produce high quality meshes a robust commercial implementation has not appeared, probably because:

- a robust implementation of the 3D medial axis has required a substantial amount of effort
- the basic algorithm was vulnerable to small features on the medial axis / surface, even if there were no small features on the defining geometry
- the treatment of concavities, especially when two or more concavities are in proximity, was not ideal
- curvature and finite contact (where the between the sphere used to trace the medial axis and the boundary at the touching point have the same curvature and the contact may extend over a finite area) were only addressed in a superficial manner
- vertices with more than three attached edges were only addressed in a superficial manner
- problems such as edges with varying dihedral angles (where the optimum number of incident hex elements changes along the length of the edge) were not addressed at all

The first two of these has been the subject of a considerable development effort by TranscenData. Significant insight into the remaining problems has been developed by Fogg [23], but no generic solutions have been yet been demonstrated.

4. Incremental approaches to hex mesh blocking

Given that auto-hex meshing has been a long-standing problem, incremental approaches to geometry decomposition and meshing have been explored so that even in the absence of a completely automated solution useful reductions in the amount of manual effort can be accomplished. Given the small number of mesh singularity types identified above, it should be possible to partition the design geometry into sub-domains with a simple singularity configuration so that existing meshing algorithms can be applied.

4.1. Geometric reasoning

Geometric reasoning implies using geometric measures as *a priori* estimates of the type of mesh required. QUB focused first on identifying thin sheets of material corresponding to medial axis surfaces [24], to which shell meshes or extruded quad meshes could be applied. This was then extended to finding long-slender features by Makem et al [25], Fig. 11. The technique used by Makem searched for regions of high aspect ratio in the solid after the thin sheet regions had been removed. Since it didn't employ the 3D medial object, features such as sliver surfaces in the MO did not matter. The remaining regions corresponding to the sub-regions around medial vertices and small features in the MO were meshed with tets.
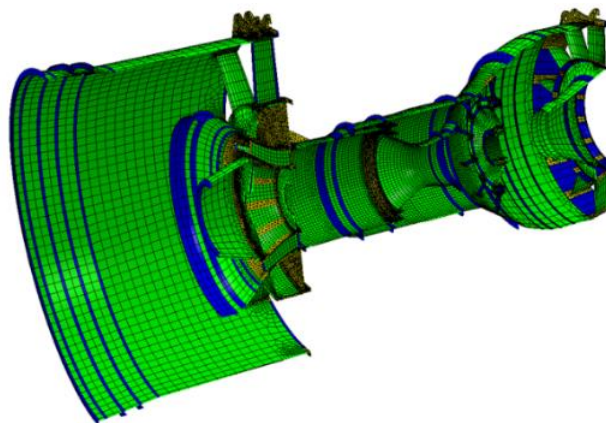

Fig. 11. Gas turbine casing model with thin sheet regions in green, long-slender regions in blue, residual complex areas in yellow. Result obtained within the CRESCENDO project, http://www.crescendo-fp7.eu/

4.1.1. Revolved regions

Axisymmetric regions such as flanges or complete cylinders can be decomposed with the addition of a single split, which can then act as both source and target of a swept mesh. The problem of finding axisymmetric regions in a general solid has been explored by the CAD community.

4.1.2. Many-to-many sweeps

Normally swept meshes are arranged to run from a source to target face. Some commercial implementations have many-to-one sweeps, where a series of faces are swept by different distances to a common target face, thus imprinting each source face mesh on the target face. A number of authors have described a many-to-many-sweep e.g. [26], where the total sweep is partitioned into a series of many-to-one sweeps so that quite complex sub-regions can be handled. The recognition of such mesh-able features is based largely on the topology of the region (for example the side faces of the sweep must be 4-sided), so that good quality elements are not guaranteed. However a substantial reduction in the amount of user effort on manual partitioning can be achieved. Some of the insights provided by Fogg [23] on subdivision and singularity tracing may be relevant in adding some geometric reasoning to improve the level of automation.

5. New fundamental approaches to hex meshing

5.1. Identifying surface singularities

Based on the theory of Bunin [27], Fogg [23] showed that the number of positive and negative singularities entering or exiting from a given surface must be

$$\sum_{vertices} Q \frac{\theta_c}{2\pi} + \sum_{edges} \int \kappa_g \, ds + \iint_{face} K \, dA + (n_+ - n_-)\frac{\pi}{2} = 0$$

where n_+ and n_- are the number of simple positive and negative singularities, Q is a source term scaled by the deviation of the corner angle from the number of elements at the corner times $\pi / 2$, κ_g is the geodesic curvature of the edges and K is the Gaussian curvature of the surface. The addition of the first three terms in the above equation always gives a multiple of $\pi/2$ and the integers n_+ and n_- must be such that the equation is satisfied. It follows that the minimum number of singularities is set by the geometry of the surface, specifically the corner angles, the geodesic curvature of the edges and total Gaussian curvature. In the minimum case either n_+ or n_- equals zero and the other balances the total flux in or out of the surface. Sometimes extra positive-negative singularities pairs are desirable to reduce the distortion or the mesh, to get a closer match to target element size or even to ensure valid mesh topologies. Fig 10 c) is a topological block, but the extra singularity pairs on opposite faces improve the element size distribution.

The geodesic curvature of a curve on a surface is [28]

$$\kappa_g = \boldsymbol{k}.(\boldsymbol{n} \times \boldsymbol{t})$$

where $\boldsymbol{t} = \frac{dx}{ds} = \frac{dx}{dt}\big/\frac{ds}{dt} = \frac{dx}{dt}\big/\left|\frac{dx}{dt}\right|$ is the tangent vector of the curve, $\boldsymbol{k} = \frac{dt}{ds} = \frac{dt}{dt}\big/\frac{ds}{dt}$ is its curvature vector and $\boldsymbol{n}$ is the normal to the surface. Note that the curvature vector can be found from the parametric curve geometry as

$$\boldsymbol{k}(t) = \frac{\boldsymbol{x}''(\boldsymbol{x}'\boldsymbol{x}') - \boldsymbol{x}'(\boldsymbol{x}'\boldsymbol{x}'')}{(\boldsymbol{x}'.\boldsymbol{x}'')^2}$$

where $\boldsymbol{x}'$ and $\boldsymbol{x}''$ are the first and second derivatives of the curve with respect to a parameter *t*.
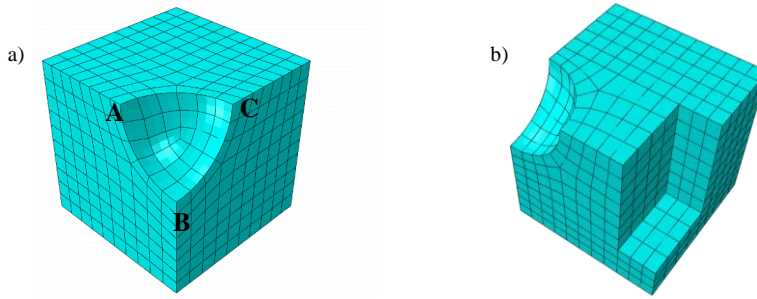
Fig. 12. Primitives meshed by midpoint subdivision (a) and (b) a primitive with the same pattern of singularities

For the spherical surface ABC of the object on the left in Fig. 12, all of the corner angles are $\pi / 2$ so the contribution of the first term in Bunin's equation is zero. For each of the edges AB, BC and CA, the curvature vector is of magnitude $1/r$ pointing towards the centre of the spherical surface, in the same direction as the surface normal. Both are perpendicular to the edge tangent. Therefore, the geodesic curvature of the edge is

$$\kappa_g = \boldsymbol{k}.(\boldsymbol{n} \times \boldsymbol{t}) = 0$$

The Gaussian curvature of the curved surface is

$$K = \kappa_1 \kappa_2 = \frac{1}{r^2}$$

where κ_1, κ_2 are the principal curvatures of the surface and r is the radius of curvature of the spherical surface. Given that the surface area of a sphere is $4\pi r^2$,

$$0 + 0 + \frac{1}{8}.\frac{1}{r^2}.4\pi r^2 + (n_+ - n_-)\frac{\pi}{2} = 0$$

implying that

$$(n_+ - n_-) = -1$$

which can be satisfied with one negative singularity. Clearly, for surface ABC a negative singularity in the middle of the face is appropriate.

For the 5-sided face adjacent to edge BC there is again no contribution from the vertices as they have corner angles of $\pi / 2$ with a target of 1 element per corner. The surface normal is perpendicular to the curvature vector and the edge tangent. Given that the edge tangent is oriented so that it is traversed anticlockwise with respect to the surface normal, the geodesic curvature of edge BC on this face is $\kappa_g = -1 / r$ so

$$\int \kappa_g ds = \int -\frac{1}{r} ds = -\frac{\pi}{2}$$

For this planar face the Gaussian curvature of the surface is zero, so the equation can be satisfied with one positive singularity. Therefore, there is one positive singularity entering each of the three 5-sided faces and one negative singularity entering the 3-sided face. When these singular edges meet in the interior of the object the result is one of the three 'prime primitives' identified in Price et al. [22].

The same analysis of the object on the right in Fig. 12 will come to the same conclusion – the extra concave and convex corners have the correct number of incident elements and do not therefore contribute to the mesh flux balance. Thus the midpoint subdivision algorithm can be trivially extended to mesh the primitive on the right in Fig. 12, given the appropriate division number constraints.

In 2D, for a medial axis with finite or curvature contact, this analysis also illustrates how to calculate whether and what type of singularities are needed. For the circular domain of Fig. 13, the medial axis is a single point whose medial disc is in contact with the whole boundary. Starting at any arbitrary point,

$$\int_0^{r\theta} \kappa_g \, ds = \int_0^{r\theta} -\frac{1}{r} \, ds = -\pi/2$$

when $\theta = \frac{\pi}{2}$, implying that a negative singularity is required every $\frac{\pi}{2}$ radians or that 4 negative singularities are required when traversing the complete circle.
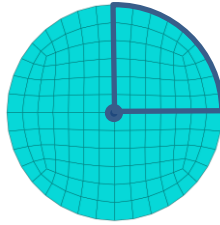

Fig. 13. One negative singularity per quadrant, based on integrating geodesic curvature round boundary associated with MA (single point at centre)

5.2. 3D frame fields for hex mesh generation

The extension of 2D cross field mesh generation to 3D introduces substantial new complexities. Some of these, like the propagation of fields from the boundary to the interior have been explored, but others, like the tracing of singular edges through the domain, require substantial new research.

5.2.1. Frame fields

The 3D equivalent of the cross field is a frame field, [29][30], a set of 3 mutually perpendicular unit vectors corresponding to the orientation of an infinitesimal hex element. Points on vertices and edges of the 3D mesh are completely defined, points on the interior of faces initially have one degree of freedom to rotate about the surface normal, and points in the interior are initially unconstrained. The problem is then to assign the remaining degrees of freedom in a background tet mesh, so that the local variation in the frame orientation can be analyzed to infer the presence or absence of a singularity, which can then be traced through the domain to define an appropriate network of singular block and mesh edges.

Various methods have been used for generating cross-frame fields in 3D. Nieser et al. [31] proposed a method to find a volume parametrization for hex meshing using cross-frame fields. This relies on a manually constructed meta-mesh which is effectively manually managing the occurrences of mesh singularities. Attempts to automate the cross-frame field generation [29], [30] lack the guarantee of a valid frame-field and associated line singularity networks.

Even measuring the relative rotation from one frame to another is not trivial, since there are 24 equivalent frames which could be used to represent a given hex element [32]. In the simplest case, the frames at the 4 vertices of a tetrahedron should correspond to the singularity patterns shown in Fig. 9, but there are other cases where for example a positive / negative singularity pair enter a given region but appear never to leave as they have combined to eliminate each other. This is feasible for a frame field but not a hex mesh.

6. Discussion

The cross / frame field, medial axis and paving / plastering approaches are different types of advancing front approach to quad / hex meshing. The crux of the problem is to determine how the different fronts interact as they advance from the boundary in orders to arrive at an optimum solution containing a valid and effective arrangement of unstructured singularities. Approaches such as the medial axis and cross/frame field aim to position singularities based on information about the proximity and relative orientations of the bits of the model boundary that they are close to, compared to paving / plastering methods which generate elements while advancing from the boundary. Therefore, the final shape and topology of the mesh, which relies on the placement of the singularities, is often compromised using paving / plastering since local considerations of the advancing mesh topology often results in many more mesh singularities than desired. In 2D, the cross field representation of a mesh is very flexible and can be smoothed to allow pairs of positive negative singularities that might occur in proximity to each other in a paving

method to cancel each other and therefore give a simpler block structured mesh topology. Boundary alignment consideration provides greater control over the placement of singularities.

Note that, as in Fig. 5, decomposition into primitives mesh-able by midpoint subdivision allows singularities to move relative to each other to create variations in mesh size and shape. This is much cheaper than re-computing a new block topology and also minimizes changes in mesh topology under changes in target element size or geometric perturbation. Both of these characteristics are desirable for optimization studies. When singularities move to primitive boundary, a new primitive topology will be required.

The fully detailed 3D Medial Object (MO) can lack robustness for complex models and is sensitive to small boundary and medial axis / surface features. However, Fogg [12] has shown how small features can be embedded within a the 2D block decomposition, allowing for greater control over mesh sizing. An alternative to the medial axis approach for finding thin-sheet regions (Section 4.1) is to utilize mid-surface information to imprint opposing face pairs, top and bottom faces bounding a mid-surface, on one another to isolate the thin-sheet region [33].

Section 4 describes the incremental approach to hex meshing where the aim is to automate the decomposition process. This approach is limited to certain classes of geometries where the singularity placement is simple and known meshing strategies like mapping, sub-mapping and sweeping may be applied. Incremental approaches have merit in reducing the elapsed meshing time and may fit well within an automated meshing process if it could be guaranteed that the residual regions, which may be meshed using more expensive and sophisticated algorithms, do not become more difficult to mesh. The method for identifying surface singularities presented in Section 5.1 could be used as an initial step to direct the decomposition process and help assess the validity of the resulting sub-regions as to whether the decomposition performed was useful or not. It could also be used to ascertain which meshing approach to utilize. For example, two opposing surfaces with the same singularity structure might help identify sweepable regions in a model.

7. Conclusions

It was stated at the beginning of the paper that it is the placement of the singularities (either positive or negative) which is key to the creation of multi-block structured meshing. In 2D there are substantial similarities between advancing front, paving, medial axis and cross-field methods. The implementation details are hugely different, but the conceptual equivalences could help target an optimum solution.

In 3D there are many similarities between the plastering, generalized plastering, medial axis and frame field approaches. The challenges in using these technologies for hex mesh generation is much greater than for quad elements. Recognizing the conceptual equivalence between the different approaches should assist in the development of efficient and comprehensive solutions. The open issue in 3D is to determine the tracing of singularities through the domain to obtain well-shaped multi-block decompositions that are adjusted to suit target element size, shape and alignment. The eventual solution might be expected to exploit the knowledge of the equivalence between the advancing frame field and advancing layers of elements to quantify all possible interactions and to use this information to create a network of singularities which is globally valid. It would also be expected that such a solution would be extensible to the computation of a solution in a mesh metric field which captures the desired variation in target element size and anisotropy, since this has been shown to be possible in 2D [12], and to treat small features and concavities .

Acknowledgements

References

[1] D. Bommes, B. Lévy, N. Pietroni, E. Puppo, C. Silva, M. Tarini, and D. Zorin, "Quad-Mesh Generation and Processing : a survey," 2012.

[2] S. Turek, M. Altieri, and C. Becker, "Proposal for SPARSE BANDED BLAS techniques.," *University of Heidelberg Germany*. 1999.

[3] S. Turek, D. Goddeke, S. Buijssen, and H. Wobker, "Hardware-Oriented Multigrid Finite Element Solvers on GPU-Accelerated Clusters," in *Scientific Computing with Multicore and Accelerators*, crc pRESS, 2010, pp. 113–130.

[4] N. Qin and X. Liu, "Flow feature aligned grid adaptation," *Int. J. Numer. Methods Eng.*, vol. 67, no. 6, pp. 787–814, 2006.

[5] S. E. Benzley, E. Perry, K. Merkley, B. Clark, and G. Sjaardema, "A comparison of all hexagonal and all tetrahedral finite element meshes for elastic and elasto-plastic analysis," in *4th International Meshing Roundtable*, 1995, pp. 179–191.

[6] A. Probst, E. Mazlum, and R. Radespiel, "Investigation of computational uncertainties of airfoil flow phenomena close to trailing edges," in *RTO-Symp. Computational Uncertainty in Military Vehicle Design, RTO-MP- AVT-147*, 2007.

[7] T. Blacker, "Meeting the Challenge for Automated Conformal Hexahedral Meshing."

[8] T. K. H. Tam and C. G. Armstrong, "2D Finite Element Mesh Generation By Medial Axis Subdivision," *Adv. Eng. Softw. Work.*, vol. 13, pp. 313–324, 1991.

[9] T. S. Li, C. G. Armstrong, and R. M. McKeag, "Quad mesh generation for k-sided faces and hex mesh generation for trivalent polyhedra," *Finite Elem. Anal. Des.*, vol. 26, no. 4, pp. 279–301, Aug. 1997.

[10] T. S. Li, R. M. McKeag, and C. G. Armstrong, "Hexahedral meshing using midpoint subdivision and integer programming," *Comput. Methods Appl. Mech. Eng.*, vol. 124, no. 1–2, pp. 171–193, Jun. 1995.

[11] T. D. Blacker and M. B. Stephenson, "Paving: A new approach to automated quadrilateral mesh generation," *Int. J. Numer. Methods Eng.*, vol. 32, no. 4, pp. 811–847, 1991.

[12] H. J. Fogg, C. G. Armstrong, and T. T. Robinson, "Automatic generation of multiblock decompositions of surfaces," *Int. J. Numer. Methods Eng.*, vol. 101, no. 13, pp. 965–991, 2015.

[13] H. J. Fogg, C. G. Armstrong, and T. T. Robinson, "Enhanced medial-axis-based block-structured meshing in 2-D," *Comput. Des.*, 2015. doi:10.1016/j.cad.2015.07.001

[14] N. Kowalski, F. Ledoux, and P. Frey, "Automatic domain partitioning for quadrilateral meshing with line constraints," *Eng. Comput.*, pp. 1–17, 2014.

[15] P. Alliez, D. Cohen-Steiner, O. Devillers, B. Lévy, and M. Desbrun, "Anisotropic polygonal remeshing," *ACM Trans. Graph.*, vol. 22, no. 3, p. 485, 2003.

[16] F. Kälberer, M. Nieser, and K. Polthier, "QuadCover - Surface parameterization using branched coverings," *Comput. Graph. Forum*, vol. 26, no. 3, pp. 375–384, 2007.

[17] M. Zhang, J. Huang, X. Liu, and H. Bao, "A wave-based anisotropic quadrangulation method," *ACM Trans. Graph.*, vol. 29, no. 4, p. 1, 2010.

[18] J. Huang, M. Zhang, J. Ma, X. Liu, L. Kobbelt, and H. Bao, "Spectral quadrangulation with orientation and alignment control," *ACM Trans. Graph.*, vol. 27, no. 5, p. 1, 2008.

[19] M. A. Price, C. G. Armstrong, and M. A. Sabin, "Hexahedral mesh generation by medial surface subdivision: Part I. Solids with convex edges," *Int. J. Numer. Methods Eng.*, vol. 38, no. 19, pp. 3335–3359, 1995.

[20] T. Blacker and R. Meyers, "Seams and wedges in plastering: A 3-D hexahedral mesh generation algorithm," *Eng. Comput.*, vol. 9, no. 2, pp. 83–93, 1993.

[21] M. L. Staten, R. A. Kerr, S. J. Owen, T. D. Blacker, M. Stupazzini, and K. Shimada, "Unconstrained plastering—Hexahedral mesh generation via advancing-front geometry decomposition," *Int. J. Numer. Methods Eng.*, vol. 81, no. 2, pp. 135–171, 2010.

[22] M. A. Price and C. G. Armstrong, "Hexahedral mesh generation by medial surface subdivision: Part II. Solids with flat and concave edges," *Int. J. Numer. Methods Eng.*, vol. 40, no. 1, pp. 111–136, 1997.

[23] H. J. Fogg, "Automatic generation of multiblock decompositions for meshing," Queen's University Belfast, 2014.

[24] T. T. Robinson, C. G. Armstrong, and R. Fairey, "Automated mixed dimensional modelling from 2D and 3D CAD models," *Finite Elem. Anal. Des.*, vol. 47, no. 2, pp. 151–165, 2011.

[25] J. E. Makem, C. G. Armstrong, and T. T. Robinson, "Automatic decomposition and efficient semi-structured meshing of complex solids," *Eng. Comput.*, vol. 30, no. 3, pp. 345–361, Dec. 2012.

[26] D. White, S. Saigal, and S. Owen, "CCSweep: automatic decomposition of multi-sweep volumes," *Eng. Comput.*, vol. 20, no. 3, pp. 222–236, 2004.

[27] G. Bunin, "A continuum theory for unstructured mesh generation in two dimensions," *Comput. Aided Geom. Des.*, vol. 25, no. 1, pp. 14–40, Jan. 2008.

[28] M. Lipschultz, *Theory and problems of differential geometry*. New York: McGraw-Hill, 1969.

[29] Y. Li, Y. Liu, W. Xu, W. Wang, and B. Guo, "All-hex meshing using singularity-restricted field," *ACM Trans. Graph.*, vol. 31, no. 6, p. 1, 2012.

[30] J. Huang, Y. Tong, H. Wei, and H. Bao, "Boundary Aligned Smooth 3D Cross-frame Field," *ACM Trans. Graph.*, vol. 30, no. 6, pp. 143:1–143:8, Dec. 2011.

[31] M. Nieser, U. Reitebuch, and K. Polthier, "C UBE C OVER – Parameterization of 3D Volumes," vol. 30, no. 5, 2011.

[32] N. Kowalski, F. Ledoux, and P. Frey, "Block-structured Hexahedral Meshes for CAD Models Using 3D Frame Fields," *Procedia Eng.*, vol. 82, pp. 59–71, 2014.

[33] L. Sun, C. G. Armstrong, T. T. Robinson, and C. M. Tierney, "Automatic Thick Thin Decomposition of Complex Body for Hex Dominant Meshing," in *Paper presented at 4th Aircraft Structural Design Conference*, 2014.