Dislocation Meshing – A Credible Solution to Automatic Hexahedral Meshing

Mr Jon Aas
CT Innovations Ltd, UK

Abstract

The behaviour of quadrilateral and hexahedral meshes is controlled by patterns that have been hidden to end-users and software vendors in the CAD, FEA and CFD communities for decades. A study of their behaviour has uncovered a number of patterns. The mathemization of these have resulted in a mathematical representation of a super-set of meshes far beyond any known contemporary meshing algorithm. An ad hoc implementation has been replaced by a new mathematical theory for representation of shapes in software using Cellular Non-Manifold Boundary-Representation Solid Models (B-rep Solid Models), creating a mathematical continuum from Parameterized CAD to FEA results evaluation and back.

The paper details the patterns and the mathematical representation of them in effective algorithm implementations. The potential in this discovery, inventions and innovations leads to practical solutions to problems that are holding the analysis community back.

Pictures printed on paper are a totally inadequate way of representing the material covered in this paper. The presentation will put this right, allowing users to see video sequences and the 3D models in an FEA pre-processor.

1. Introduction

At the **NAFEMS World Congress '97** in Stuttgart, 9-11 April 1997 Dr Bruce E. MacNeal and Dr Richard H. MacNeal gave a presentation called **"Future Issues – a Code Developer's Perspective"** where they stated: *"The "holy grail" of automatic hexahedral meshing still eludes us. Automatic meshing technology will have a long growth phase."*, /1/.

A lot of research and development efforts, resources and time have been spent on the search for the Holy Grail – Automatic Hexahedral Meshing – before and after the NAFEMS congress in 1997. So far, the software developers and the analysis community as a whole know a lot about what doesn't work.
The alternative approaches are exhausted and newcomers to the problem are encouraged to turn the stack of failed approaches over and start again, repeating the mistakes of others with the same outcome as before.

Nothing significantly has happened in this domain for a decade or more, and everybody seems resigned to live with "second best" on a permanent basis. No one has actually asked the question **"Somehow, there must be a better way, right?"**

Dislocation Meshing is a research effort completely detached from any other effort in the search for the Holy Grail. It is based on sound scientific practice of understanding the problem at hand before a solution is proposed.

The objective of this research has been clear from the start:
- to discover the patterns that control the behaviour of quadrilateral and hexahedral meshes.
- to invent the mathematical formulation that can take a B-rep Solid Model as input and deliver a hexahedral mesh as the result.
- to discover how the process can be controlled using high level Volume Mesh Controls to specify what goes on inside the cluster of cells.

A long time was spent on understanding the behaviour of quadrilateral and hexahedral meshes until the patterns that control their behaviour emerged.

The mathemization of these patterns resulted in a system of equations that describe the flow of quadrilateral elements in the interior of 2D regions and the flow of hexahedral elements in the interior of 3D solid models. These systems of equations can be hand-created or computer-generated each time, requiring substantial manual and computational effort. Each set of underdetermined equation systems has many solutions, and technology to evaluate the meshes and analysis results coming from them must be developed.

2. The separation of geometry and topology

Dislocation Meshing is a topological approach to quadrilateral and hexahedral meshing. The approach establishes the connectivity of the elements within the boundary of the shape and subsequently maps the connected element mesh into a geometric space. This is a "connectivity first – coordinates second" approach enabling a user to stretch and compress the geometry without losing the connectivity of the mesh once it is created.

The approach described here expects the computer systems to hold a cluster of B-rep Solid Models as a logical unit. The technical term for these is Cellular Non-Manifold Boundary-Representation Solid Models, /2/, /3/. a mouthful any day of the year. For readability of the paper, the short form 'B-rep Solid Model' is used whenever the cellular non-manifold version is described.

3. The patterns that control the behaviour of quadrilateral and hexahedral meshes

The patterns that control the behaviour of quadrilateral and hexahedral meshes are detailed in the following list. Examples are used to visualise the patterns, while accompanying text explains the consequence of the discovery of them. Design decisions are discussed to create an implementable approach that can be expressed in mathematical terms.

Pattern #1: The behaviour of quadrilateral meshes are controlled by irregular vertices in the interior of the meshes

The topology of the boundary of a quadrilateral mesh is called a face and consists of the vertices and edges that make up the external and internal edge loops. Any number of geometric forms may share the same topological face definition.

A quadrilateral mesh may be created in a 2D plane or be embedded in a single or double curved surface standing in 3D space. The number of vertices in the boundary determines how the quadrilateral mesh can flow in the interior. There is a mathematical relationship between the number of vertices in the boundary and the minimum number of irregular vertices needed to create a well-formed quadrilateral mesh. Irregular vertices in a quadrilateral mesh are of two kinds:
- three-way irregular vertices where three elements meet
- five-way irregular vertices where five elements meet
Four-way vertices where four elements meet are regular.

The vertices in the boundary of a face will all have two edges meeting and are here called $\mathbf{V_2}$. For a face with no internal loops, the equation is the number of irregular vertices expressed as:

$$\mathbf{D = V_2 - 4} \qquad (1)$$

Where
D is the number of irregular vertices in the face
$\mathbf{V_2}$ is the number of vertices in the boundary of the face

Examples are shown in figure 1 where a triangle, quadrilateral, pentagon, hexagon and heptagon are shown with their respective coarse quadrilateral carpets.

- a triangular face has a D= -1, that is it needs a three-way irregular vertex, called a negative irregular vertex, to form a quadrilateral mesh.
- a quadrilateral face has a D= 0, that is it needs no irregular vertices to form a quadrilateral mesh

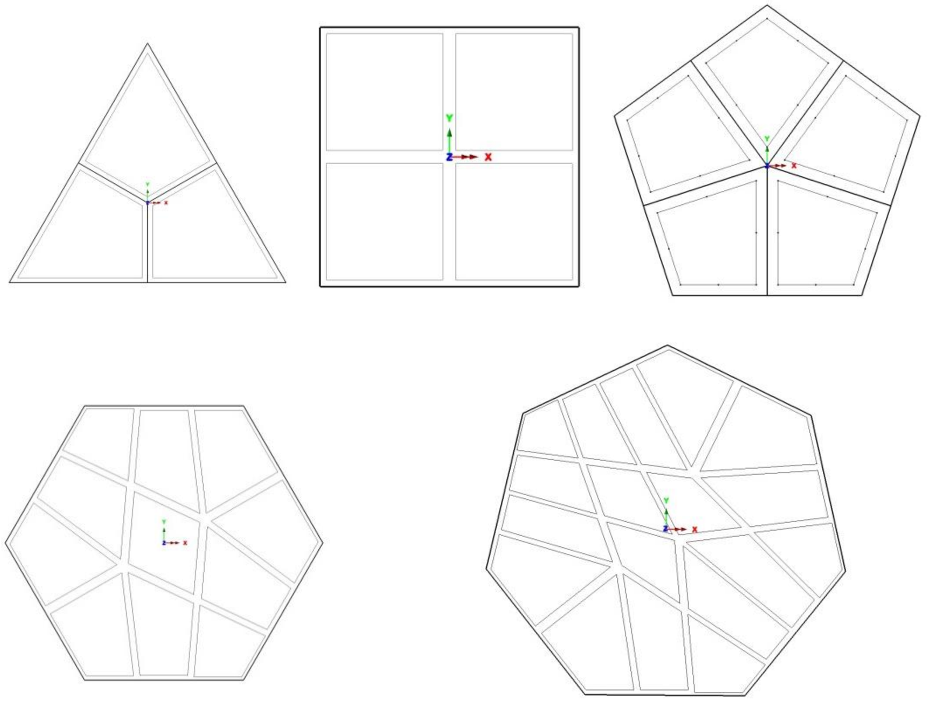


Figure 1: Faces with no internal loops, their coarse quadrilateral carpets and the number of irregular vertices shown

- a pentagonal face has a D= + 1, that is it needs a five-way irregular vertex, called a positive irregular vertex, to form a quadrilateral mesh
- a hexagonal face has a D= + 2, that is it needs two positive irregular vertices to form a quadrilateral mesh. There are 3 alternative structures
- a heptagonal face has a D = + 3, that is it needs three positive irregular vertices to form a quadrilateral mesh. There are 14 alternative structures.

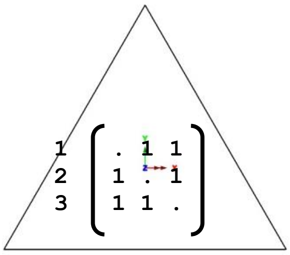
The irregular vertices can have any position in the faces controlled by the widths of the flow lines in the boundary and the internal flow lines, appearing in the hexagon for the first time. The three faces: the triangle, the quadrilateral and the pentagon form the 2D Kernels, from where any other tessellation can be built by combining these in all sorts of ways.

The relationships between the edges and the flowlines crossing them can be expressed as an equation system in this format:

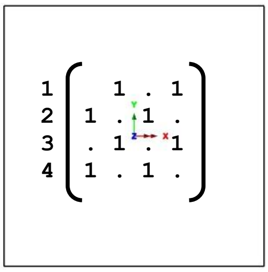
$$\mathbf{Aw} - \mathbf{I}\,\mathbf{L} = \mathbf{0} \tag{2}$$

Where

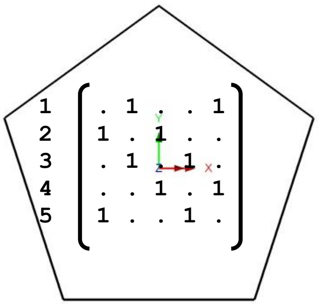
A is an edge-edge opposite matrix for the face
w is the width of a flow line parallel to an edge
I is the identity matrix
L is the length in number of elements for the edge, i.e., its division number



$$\begin{array}{cc} & 3 \\ \begin{array}{c}1\\2\\3\end{array} & \begin{bmatrix}1&.&1\\1&1&.\\.&1&1\end{bmatrix}\end{array} W - \begin{array}{c} 3 \\ \begin{bmatrix}1&.&.\\.&1&.\\.&.&1\end{bmatrix}\end{array} L = 0$$



$$\begin{array}{cc} & 4 \\ \begin{array}{c}1\\2\\3\\4\end{array} & \begin{bmatrix}.&1&.&1\\1&.&1&.\\.&1&.&1\\1&.&1&.\end{bmatrix}\end{array} W - \begin{array}{c} 4 \\ \begin{bmatrix}1&.&.&.\\.&1&.&.\\.&.&1&.\\.&.&.&1\end{bmatrix}\end{array} L = 0$$



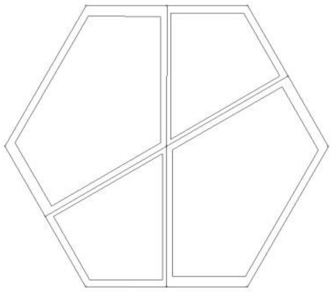
$$\begin{array}{cc} & 5 \\ \begin{array}{c}1\\2\\3\\4\\5\end{array} & \begin{bmatrix}.&1&.&.&1\\1&.&1&.&.\\.&1&.&1&.\\.&.&1&.&1\\1&.&.&1&.\end{bmatrix}\end{array} W - \begin{array}{c} 5 \\ \begin{bmatrix}1&.&.&.&.\\.&1&.&.&.\\.&.&1&.&.\\.&.&.&1&.\\.&.&.&.&1\end{bmatrix}\end{array} L = 0$$

Figure 2: The E_E matrix and the corresponding underdetermined equation system for the triangle, quadrilateral and pentagon, the 2D Kernels

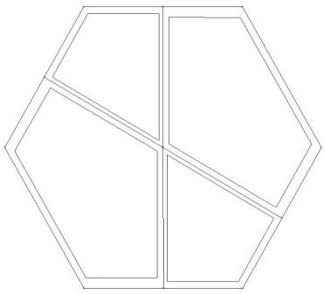
The equation systems shown in Figure 2 are an underdetermined system which requires half of the unknowns to be specified so the equation system can be solved. This format is chosen so it is possible to describe a mesh in terms of its flow line widths only, its division numbers only or a combination of the two.

The triangle, quadrilateral and pentagon have one or none irregular vertex as a minimum and finding a solution to the equation system is straight forward. For the hexagon and any polygon with more than five vertices in the boundary there is more than one equation system. The hexagon has two positive irregular vertices that can be positioned relative to each other in three different ways, here represented as pentagons:

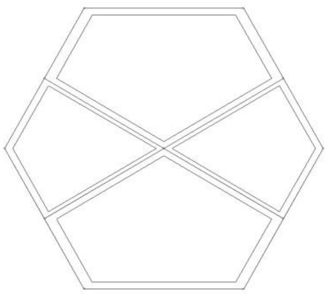
The hexagons in figure 3 use three alternative subdivisions into quadrilaterals and pentagons to separate the positive irregular vertices, each subdivision using two out of a set of three internal edges cutting across the face. This can be expressed in a shared equation system by including all the internal flow lines in the equation system.



$$\begin{array}{c} \\ 1\\2\\3\\4\\5\\6 \end{array}\begin{array}{c} \qquad\qquad 6 \qquad\qquad\qquad 2 \qquad \\ \left[\begin{array}{cccccc|ccc} . & 1 & . & . & . & 1 & 1 & . & . \\ 1 & . & 1 & . & . & . & . & 1 & . \\ . & 1 & . & 1 & . & . & . & . & . \\ . & . & 1 & . & 1 & . & 1 & . & . \\ . & . & . & 1 & . & 1 & . & 1 & . \\ 1 & . & . & . & 1 & . & . & . & . \end{array}\right] \end{array} W = L$$



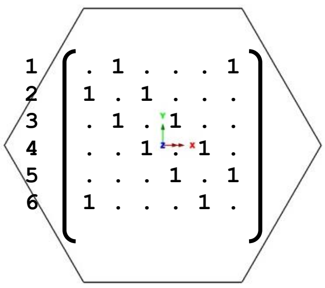
$$\begin{array}{c} \\ 1\\2\\3\\4\\5\\6 \end{array}\begin{array}{c} \qquad\qquad 6 \qquad\qquad\qquad 2 \qquad \\ \left[\begin{array}{cccccc|ccc} . & 1 & . & . & . & 1 & 1 & . & . \\ 1 & . & 1 & . & . & . & . & . & . \\ . & 1 & . & 1 & . & . & . & . & 1 \\ . & . & 1 & . & 1 & . & 1 & . & . \\ . & . & . & 1 & . & 1 & . & . & . \\ 1 & . & . & . & 1 & . & . & . & 1 \end{array}\right] \end{array} W = L$$



$$\begin{array}{c} \\ 1\\2\\3\\4\\5\\6 \end{array}\begin{array}{c} \qquad\qquad 6 \qquad\qquad\qquad\qquad 2 \\ \left[\begin{array}{cccccc|ccc} . & 1 & . & . & . & 1 & . & . & . \\ 1 & . & 1 & . & . & . & . & 1 & . \\ . & 1 & . & 1 & . & . & . & . & 1 \\ . & . & 1 & . & 1 & . & . & . & . \\ . & . & . & 1 & . & 1 & . & 1 & . \\ 1 & . & . & . & 1 & . & . & . & 1 \end{array}\right] \end{array} W = L$$

Figure 3: The three cases for the hexagon where the face can be subdivided into a carpet of quadrilaterals and pentagons in three different ways

As the three alternative subdivisions share the set of internal edges, the three individual equation systems for a hexagon can be compacted to:



$$\begin{array}{c} 1\\2\\3\\4\\5\\6 \end{array}\left[\begin{array}{cccccc} . & 1 & . & . & . & 1 \\ 1 & . & 1 & . & . & . \\ . & 1 & . & 1 & . & . \\ . & . & 1 & . & 1 & . \\ . & . & . & 1 & . & 1 \\ 1 & . & . & . & 1 & . \end{array}\right]$$

$$\begin{array}{c} \\ 1\\2\\3\\4\\5\\6 \end{array}\begin{array}{c} \qquad 6 \qquad\qquad\qquad 3 \qquad \\ \left[\begin{array}{cccccc|ccc} . & 1 & . & . & . & 1 & 1 & . & . \\ 1 & . & 1 & . & . & . & . & 1 & . \\ . & 1 & . & 1 & . & . & . & . & 1 \\ . & . & 1 & . & 1 & . & 1 & . & . \\ . & . & . & 1 & . & 1 & . & 1 & . \\ 1 & . & . & . & 1 & . & . & . & 1 \end{array}\right] \end{array} W - \begin{array}{c} 6 \\ \left[\begin{array}{cccccc} 1 & . & . & . & . & . \\ . & 1 & . & . & . & . \\ . & . & 1 & . & . & . \\ . & . & . & 1 & . & . \\ . & . & . & . & 1 & . \\ . & . & . & . & . & 1 \end{array}\right] \end{array} L = 0$$

$$\begin{array}{c} 1\\2\\3 \end{array}\left[\begin{array}{ccc} 1 & 1 & . \\ 1 & . & 1 \\ . & 1 & 1 \end{array}\right]$$

Figure 4: The set of underdetermined equation systems for the hexagon

The additional matrix below the internal flow lines in figure 4 keeps a track on which flow lines belong in a group, hence it is called a group matrix (G). The number of unknowns is now extended to 15, so more of them must be specified for the equation system to have a solution. Only two of the internal flow lines are used in each relative positioning of the irregular vertices in the hexagon. The number of equations is 6, the number of groups is 3 and the number of unknowns in each group is 6+2+6=14.

The solutions are found by specifying enough of the external and internal widths for the flow lines, or two flow lines widths and all the division numbers, or … You get the drift.

For meshing of a hexagon it would be natural to specify all the flow line widths and solve the equation system with respect to the division numbers. That means the user controls the positioning of the positive irregular vertices in the face and lets the algorithm find the corresponding division numbers.

The traditional meshing technique for a hexagon where all division numbers are given as input requires further two unknowns to be specified. The traditional solution uses fixed patterns where the positions of the irregular vertices are built in, i.e., both internal flow line widths are fixed. There are three orientations of the pattern, often ignored when using fixed patterns.

With this approach, all possible cases can be considered and all solutions are represented in one compact underdetermined equation system. A solution to the equation system will be a vector of flow line widths from where the division numbers are computed. Together they give the absolute positioning of the irregular vertices and uniquely define a particular mesh for the hexagon. All possible solutions for a hexagon can be described uniquely this way.
The same principle is extended to the other faces with more vertices in the boundary, /4/.

This set is a combination of all the Edge-Edge opposite matrices for a face with no internal loops.

This form is chosen because it can represent all the ways a face with no internal loops can be subdivided into a set of 2D Kernels, and ultimately a Coarse Quadrilateral Carpet. The cases that are included so far are based on equation (1), establishing a mathematical relationship between the face topology and the number of irregular vertices it must have to create a Coarse Quadrilateral Carpet. Their presence is a Law of Nature.
This set of irregular vertices is called the **Fundamental Set**.

The general format for the set of underdetermined equation systems for single faces becomes:

$$\begin{array}{c}\left[\mathbf{A_e} \mid \mathbf{A_i}\right] \\ \left[\mathbf{G}\right]\end{array}\begin{bmatrix}\mathbf{W_e}\\ \mathbf{W_i}\end{bmatrix} - \left[\mathbf{I}\right]\left[\mathbf{L}\right] = \mathbf{0}$$

Where

A_e is the edge-edge opposite matrix for the external edges in the face
A_i is the edge opposite matrix for the internal edges in the face
W_e is the vector for widths for the external edges
W_i is the vector for the widths for the internal edges
G is the group matrix indicating which internal edges form a group
I is the unity matrix (only ‘1’s on the diagonal and ‘0’s elsewhere)
L is the vector of division numbers for all external edges

Figure 5: The general form of the underdetermined equation system for single faces

It is clear that a quadrilateral mesh can have additional irregular vertices in it, and these appear in groups of two, a negative and a positive irregular vertex. These are independent of the topological characteristics of the face and can appear in any numbers in any position in the quadrilateral carpet. This set of irregular pairs of vertices is called the **Auxiliary Set**.

Pattern #2: The behaviour of hexahedral meshes are controlled by irregular edges running through the meshes

The key characteristics of hexahedral meshes are the presence of irregular edges throughout their interior. On the outside of a hexahedral mesh there are irregular nodes (three, five or more elements meeting at an irregular node, four meeting at a regular node) used as the start of irregular edges which stretch into the volume and emerge at another irregular node somewhere on another external face.

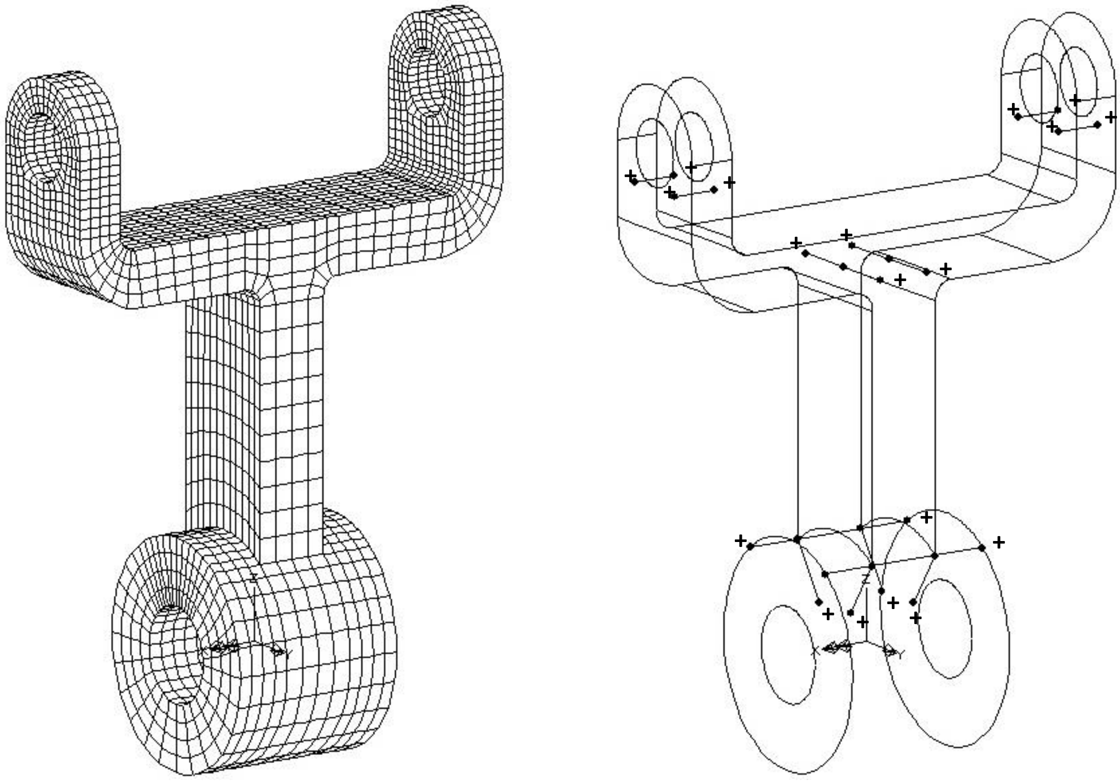


Figure 6: A hexahedral mesh and the outline with irregular edge networks indicated

The model in figure 6 has only single irregular edges as the shape is dominated by 2.5D parts. This is the case for a large number of shapes. The irregular edges are all five-way irregular edges, hence the plus signs.

In more chunky 3D models, the irregular edge networks start at the exterior of the shape at an irregular vertex and go into the volume to interact with other irregular edges and come out again at a different external vertex.

The polyhedron called a Dodecahedron (0-0-12) is used to show alternative irregular edge networks, all using the single positive irregular vertices in the boundary, one for each of the pentagons, i.e., the Fundamental Set, 12 in all.

The different irregular edge networks are created by cutting through the shape with planar cuts, each cut creating further subdivision of the cells into smaller cells. From two cuts onwards, all cells are 3D Kernels. The Figures 7 and 8 show the Coarse Hexahedral Cluster in each case and half the model is included to show how the irregular edges are gradually separated in a hexagonal cut.

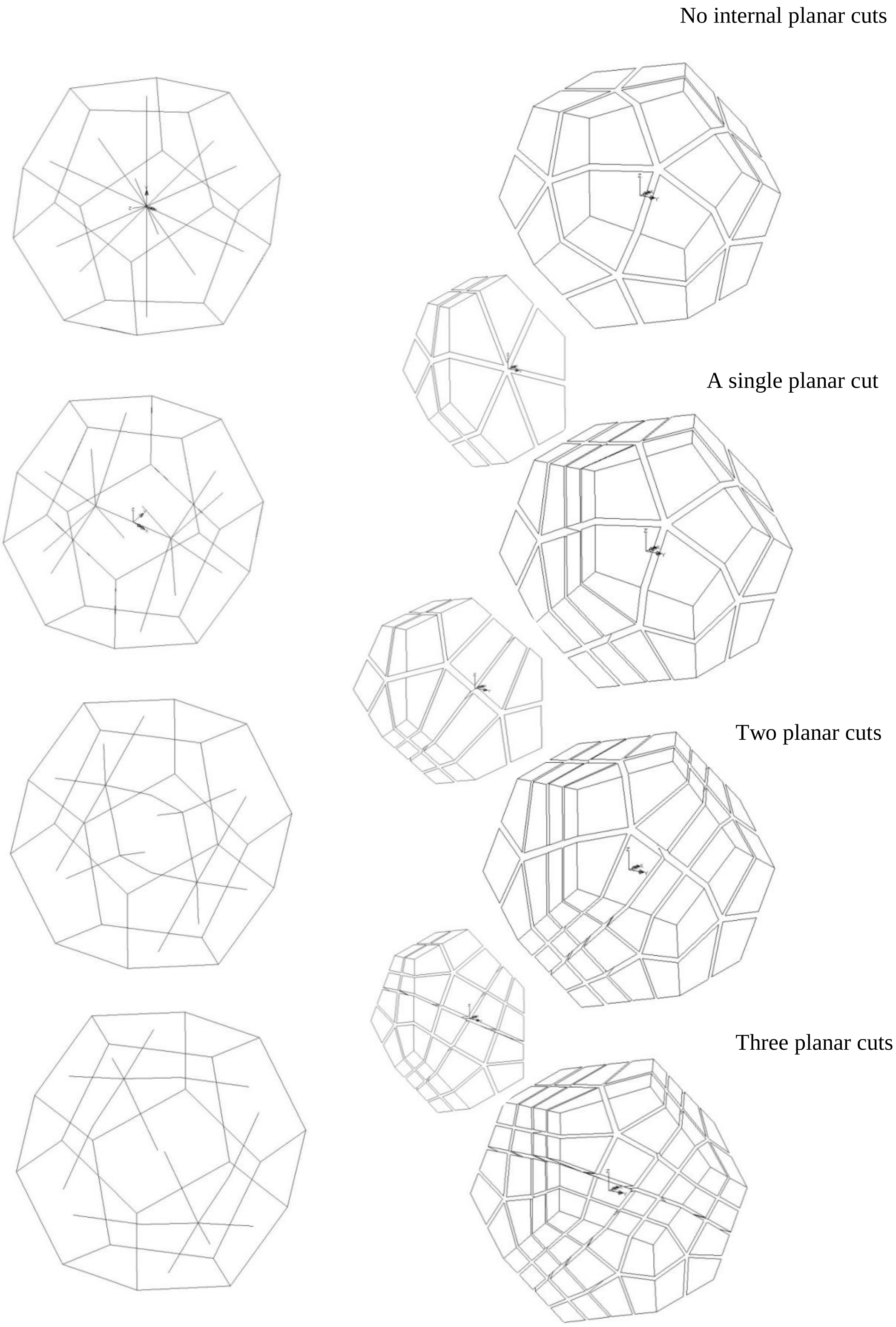
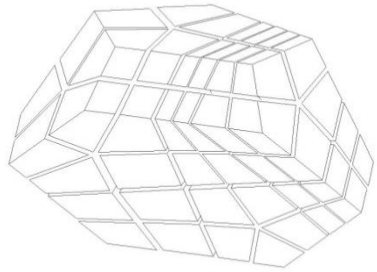


Figure 7: A Dodecahedron with progressive separation between the irregular edge networks



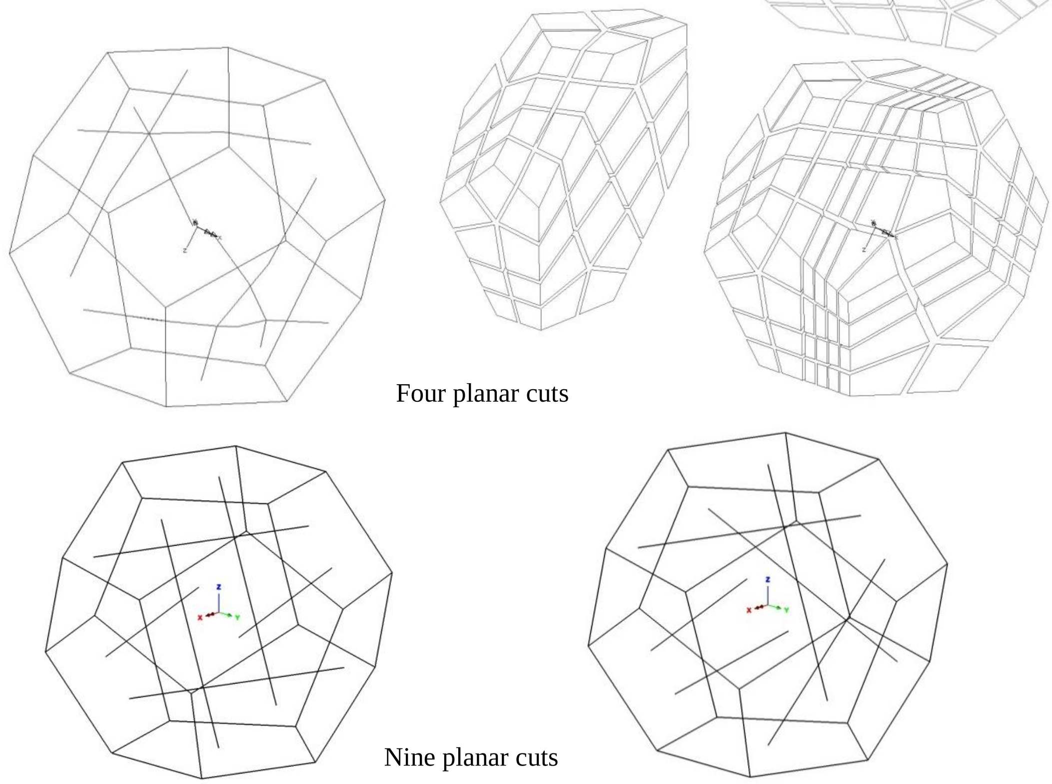


Figure 8: A Dodecahedron with further progressive separation between the irregular edge networks

- the first has no separation, i.e., this is the network for Mid-Point Sub Division (MPSD)
- the second has one planar cut separating the network into two internal vertices with one connection between them
- the third has two planar cuts separating the networks into four internal vertices with a loop connecting them
- the fourth has three planar cuts through it creating two internal vertices with no connection between them
- the fifth has four planar cuts through it creating three internal vertices, two are closely connected
- the sixth and seventh have nine planar cuts through them, separating the irregular edges completely, leaving six independent edges through the volume.

For this topology, nine cuts will create full separation. There are two alternative sets of irregular edge networks with full separation, using two alternative set of planar cuts.

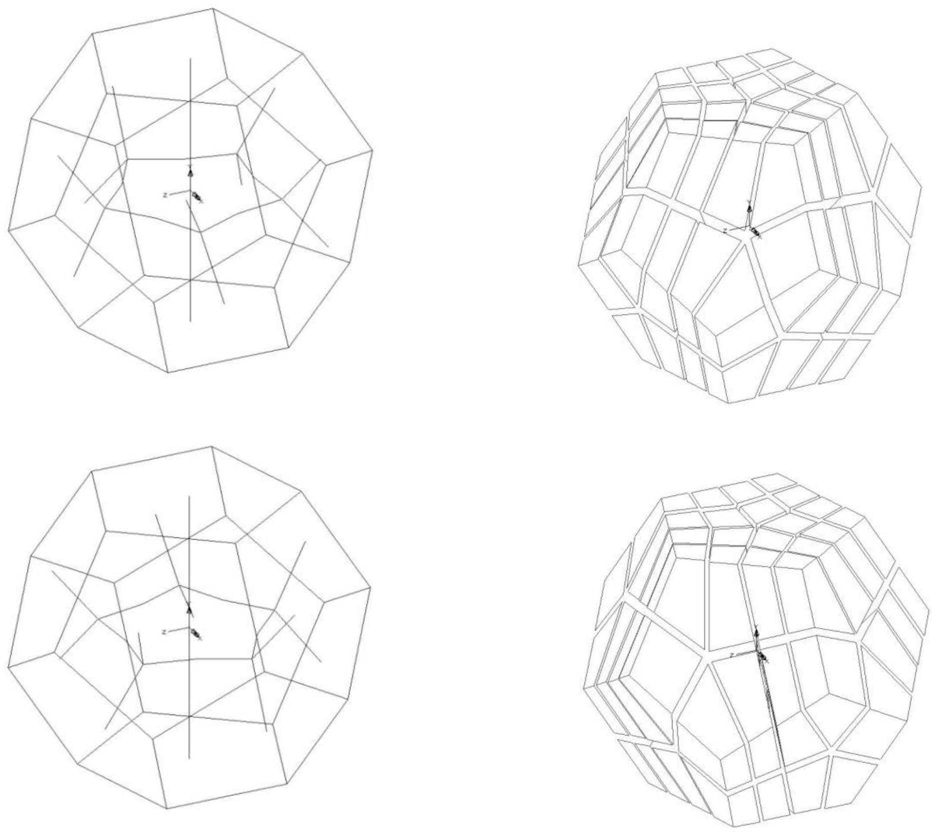


Figure 9: The Dodecahedron is also a Pentagonal Truncated Trapezohedron with two alternative irregular edge networks

The (0-0-12) topology has additionally two irregular edge networks created by connecting the top and bottom pentagons with a single irregular edge as shown in figure 9. This divides the cell into five equal sub-cells that have an irregular vertex each connected to each other in a loop. There are two cases, one left and one right. The other Truncated Trapezohedra show the same structure. Pictures 7, 8 and 9 include some of the alternative irregular edge networks for the Dodecahedron (also used as a Pentagonal Truncated Trapezohedron). There are hundreds of them, determined by the number of flow sheets cutting through the shape, any one of nine, any two of nine, etc. All represent a particular structure for a class of hexahedral meshes for a particular face as the bottom face. Due to symmetry in the Dodecahedron, all these irregular edge networks can use any of the 12 faces as the bottom face. Each of them can be derived from the F_F matrix.

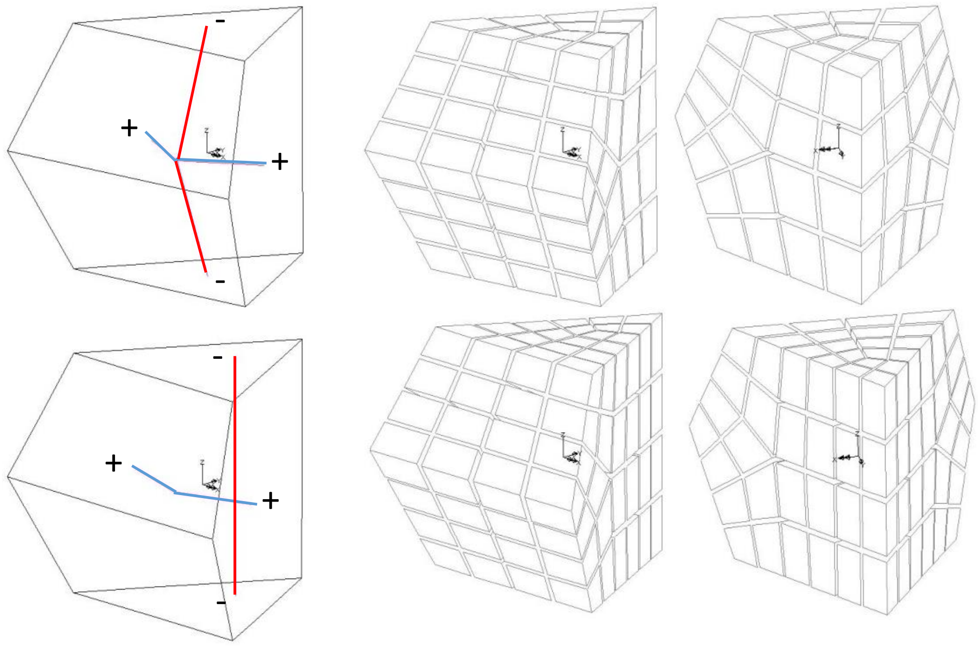


Figure 10: The NoName shape from the 3D Kernels (2-2-2) with both negative and positive irregular edges.

Pattern #3: There are two kinds of irregular edges in a hexahedral mesh:
The pattern discovered for 2D quadrilateral meshes extends to 3D hexahedral meshes, there are two different irregular edges in a hexahedral mesh:
- three-way irregular edges where three element edges meet (red)
- five-way irregular edges where five element edges meet (blue or black)

The four-way edges are regular as they have four elements sharing the edge. Figure 10 shows the only two configurations for the irregular edges in (2-2-2), the first is the configuration where all edges meet in the middle (MPSD) and the second is the full separation case. The respective meshes are shown from two directions. All flow sheets have a width of 2.

Figure 11 shows two of the full separations for the Quadrilateral Truncated Trapezohedron (0-2-8) with their respective meshes and analysis results from a linear static analysis. Here, all irregular edges have five elements meeting.

The irregular edges with more than five elements meeting follow the same pattern as for irregular vertices in faces. Edges where six or more elements meet are combinations of positive irregular edges,
- two positive irregular edges combine to a six-way irregular edge,
- three positive irregular edges combine to a seven-way irregular edge and so on.

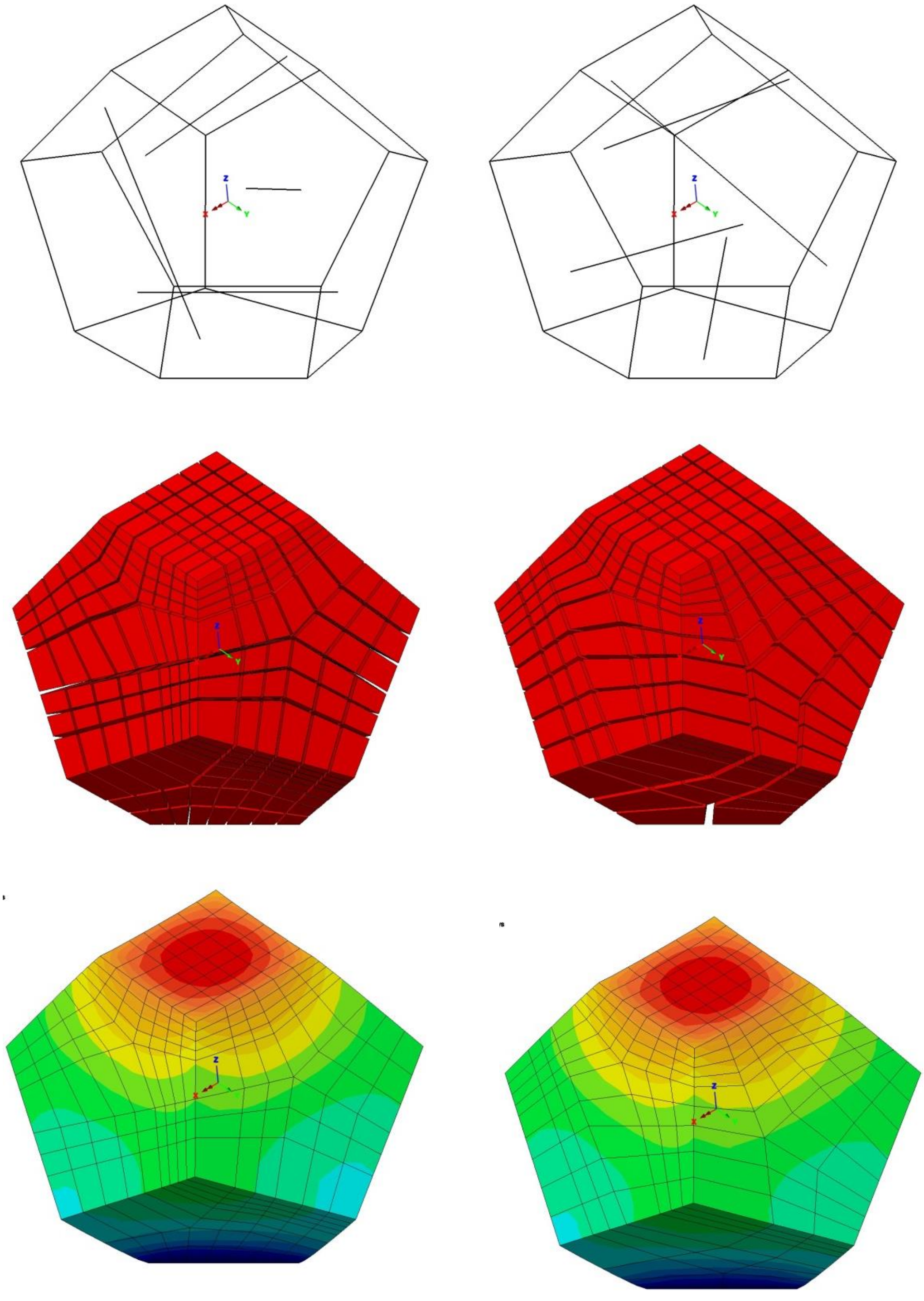


Figure 11: The Quadrilateral Truncated Trapezohedron (0-2-8) with full separation of their four positive irregular edges

Pattern #4: There are two kinds of irregular edge networks.

The first set is derived from the number of irregular vertices in the faces bounding the cell. This set of irregular edges must be present in the mesh as a Law of Nature, they cannot be avoided. This set of irregular edges is called the **Fundamental Set**.

In a triangular face, a negative irregular edge will start at the negative irregular vertex, go into the mesh and interact with other irregular edges inside the mesh, both negative and positive. For a quadrilateral, there are no irregular vertices, hence no irregular edge to connect up. For a pentagonal face, a positive irregular edge will start at a positive irregular vertex, go into the mesh and interact with other irregular edges inside the mesh, both negative and positive and come out again somewhere.

The irregular edges are positioned in the cells by controlling the width of each of the external and internal faces. In this way the mesh flow is determined by controlling the interior structure of a hexahedral mesh.

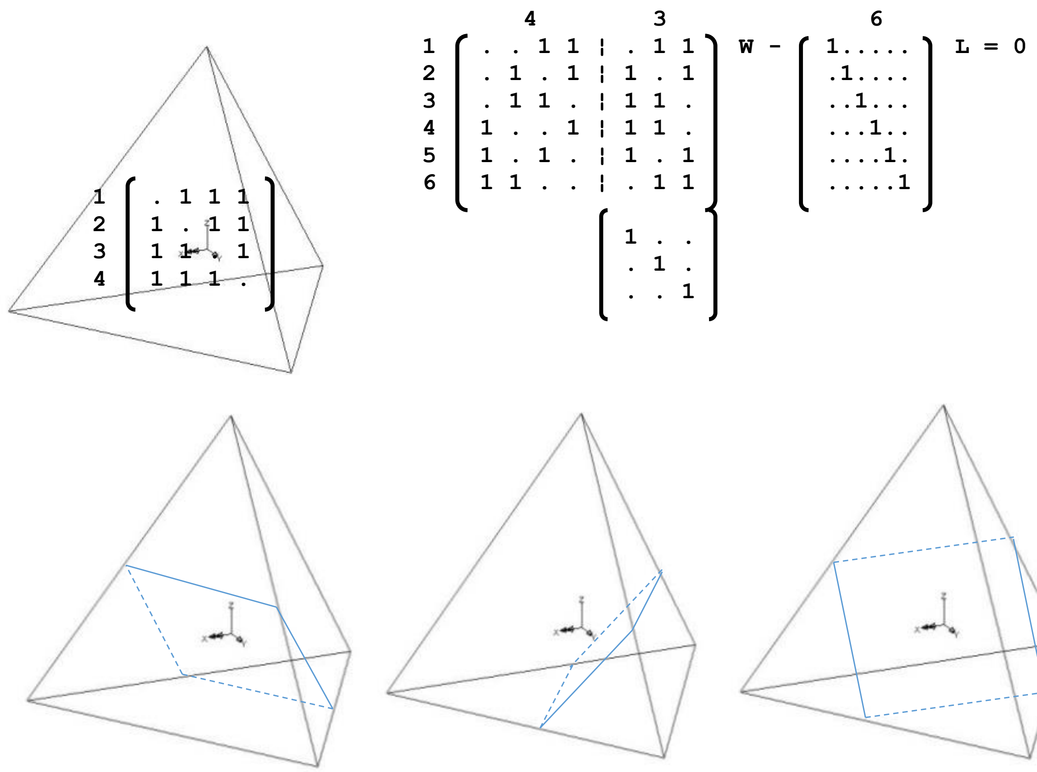


Figure 12: The tetrahedron (4-0-0), an F_F matrix representing it and the corresponding underdetermined equation system and the three internal flow sheets

The irregular edges can form any number of irregular edge networks and the positioning of the irregular vertices and edges in these networks are controlled by the width of external and internal faces. The widths are part of the Volume Mesh Controls for positioning irregular edges and the interior vertices.
The calculation of which internal faces are needed to create all the alternative separations that the irregular edge can take uses the F_F matrix as the input and produces a set of underdetermined equation systems as output. This follows the same path as for quadrilateral carpets in 2D. An example illustrates this.

The Tetrahedron is represented by an F_F matrix and the corresponding compact underdetermined equation system. The G-matrix represent the three possible ways the tetrahedron can be subdivided into two prisms, each with their separate negative irregular edge. This is similar to the hexagon in 2D.

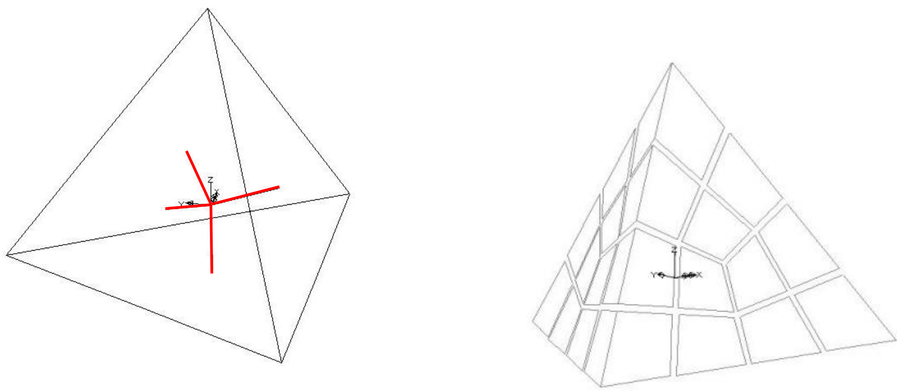


Figure 13: The irregular edge network for the Tetrahedron (4-0-0) with no separation and an MPSD mesh where all the external flow sheets have a width = 2 and the absent internal flow sheets widths are all 0

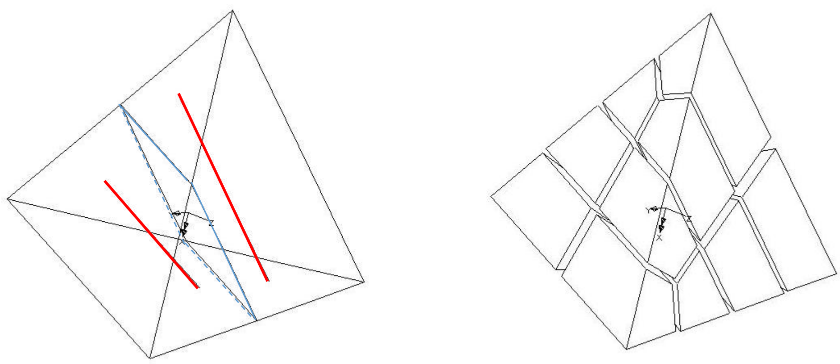


Figure 14: The irregular edge network for the Tetrahedron (4-0-0) with an internal flow sheet where all the external flow sheets have a width = 1 and the internal flow sheet a width = 2.

The same method is used for all polyhedra, to transform the F_F matrix into a set of underdetermined equation systems that can be solved by specifying the widths of the flow sheets, the division numbers or a combination of these, /4/.

The second set of irregular edge networks are created by combining a negative and positive irregular edge as a pair. These are independent of the characteristics of the cell exterior and any number of them can be added to the hexahedral mesh as long as they form structures that yield hexahedral meshes.

These pairs are called the **Auxiliary Set**.

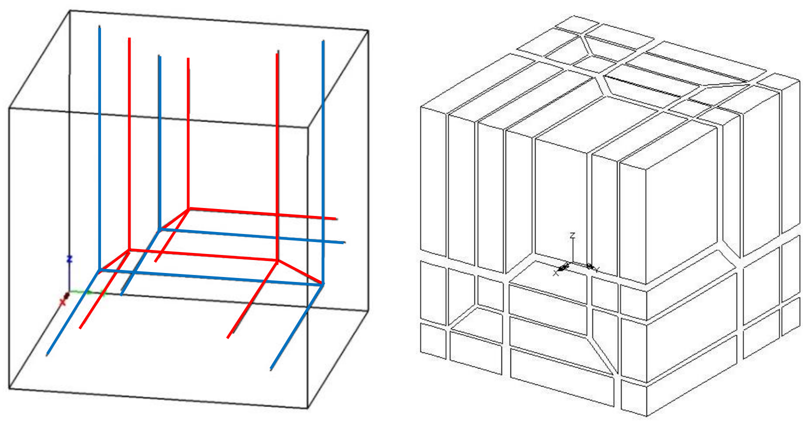


Figure 15: An example of pairs of irregular edge networks in a cube (0-6-0) together with its Coarse Hexahedral Cluster

There are a vast number of alternative structures where pairs of negative and positive irregular edges interact with other pairs to form a hexahedral mesh. Any number of these structures can be added to the structure created using the Fundamental Set. Their composition, relative and absolute positioning are defined by additional Volume Mesh Controls.

4. Solving the set of underdetermined equation systems in 3D

The Meshing process consist of two stages:
- to create a Coarse Hexahedral Cluster, by specifying “in or out”, 1 or 0, for all the internal flow sheets.
- to create a hexahedral mesh in the Coarse Hexahedral Cluster by specifying the widths as a positive integer.

The group definitions determine which internal flow sheets should be considered, if given the value ‘1’ it will be present and given the value ‘0’ it will be absent within the group. When using ‘0’, all groups containing this internal flowsheet are excluded.

The general form of the set of underdetermined equation systems in 3D is given as follows:

$$\underbrace{\left[\mathbf{A_e} \mid \mathbf{A_i}\right]}_{\left[\mathbf{G}\right]}\begin{bmatrix}\mathbf{w_e}\\ \mathbf{w_i}\end{bmatrix} - \left[\mathbf{I}\right]\left[\mathbf{L}\right] = \mathbf{0}$$

Where

Ae is the edge-face opposite matrix for the external faces in the polyhedron
Ai is the edge to internal face opposite matrix for the internal faces in the polyhedron
We is the vector for widths for the external faces
Wi is the vector for the widths for the internal faces
G is the group matrix indicating which internal faces form a group
I is the unity matrix (only '1's on the diagonal and '0's elsewhere)
L is the vector of division numbers for all external edges

Figure 16: The general form of the set of underdetermined equation systems for a polyhedron

All the external flow sheet must be present, but may have a width = 0. This will position the positive irregular edge in the boundary of the polyhedron, either in the face, along an edge or at the internal vertex in a vertex of the polyhedron. There are restrictions to what combination of external width=0 that can be used. Negative irregular edges cannot be positioned in the boundary of the shape.

In the same way as for the 2D underdetermined equation systems, enough unknowns have to be specified to create a square equation system which can then be solved. A sequence of alternative meshes can be created by using a range of values for the fixed unknowns, controlling both the presence or absence for internal flow sheets and the values for the widths for all flow sheets in the mesh. A large number of alternative meshes can be created and compared. Alternative strategies for solving the set of underdetermined equation systems are described in more detail in /4/.

A design decision: The Auxiliary Set cannot be controlled by the equation system created for the Fundamental Set. Hence, the Auxiliary Set is kept out of the first tessellation process from cell cluster to Coarse Hexahedral Cluster and introduced as part of the hex-in-hex meshing process instead. Meshing then becomes a choice of regular and irregular hex-in-hex meshing, i.e., an additional set of Volume Mesh Controls is used to position the irregular edge networks in the Auxiliary Set within a Coarse Hexahedral Cluster.

5. The mathemization of these patterns

The patterns that control the behaviour of quadrilateral meshes can be extended into 3D. Hexahedral meshes have the same topological characteristics and follow the same patterns as the quadrilateral meshes. The underdetermined equation system in 2D can be used in 3D, now the unknowns are still edges, but their lengths are derived from external and internal faces called flow sheets. The equation systems are a group of E_F opposite adjacency matrices and their solution is expressed as a vector of the widths of each of the internal and external flow sheets.

$$\mathbf{E_E} \quad \Longrightarrow \quad \mathbf{Aw - I\,L = 0} \quad \Longleftarrow \quad \mathbf{F_F}$$

2D shape **3D shape**

Figure 17: The same mathematical representation can be used in both 2D and 3D

Dislocation Meshing is unique among meshing technologies in the fact that the 2D solution can be extended directly into 3D.

So far, the set of underdetermined equation systems are developed for a single polyhedron at a time. Real objects are made up of several cells and the adjacency matrices must represent the assemblies of the individual polyhedra, the topological representation of the cells. In the same way the set of underdetermined equation systems for the individual cell must be assembled into a shared set of underdetermined equation systems for the whole object.

The linear algebra for these operations follow the same path as every reader is familiar with: the building of a global stiffness matrix from a set of cells and their individual stiffness matrices. The stiffness matrix assembly process uses a set of cells that are defined as elements: triangular or quadrilateral in 2D and tetrahedral, hexahedral or pentahedral in 3D and the relationship between their local node numbers and the global numbering system. This is a particular case of shape assembly; the same principle can be followed for assembly of any combination of different topologies, for both the combined F_F matrix and the combined set of underdetermined equation systems.

The end result is a global representation of external and internal flow sheets that separate the irregular edge networks in all possible ways.

Dislocation Meshing can create thousands of meshes for a given engineering object:
- firstly, thousands of classes of irregular edge networks that use external and internal flow sheets to tessellate the shape into a cluster of 3D Kernels and eventually a Coarse Hexahedral Cluster. This gives the relative positioning of the irregular edge networks.
- secondly, each of the Coarse Hexahedral Clusters can be given a range of widths for each of the external and internal flow sheets. Each combination results in a specific mesh using regular hex-in-hex meshing algorithms
- thirdly, any number of pairs of irregular edge networks from the Auxiliary Set can be added to the Coarse Hexahedral Cluster in an irregular hex-in-hex meshing algorithm.

All of them can be described uniquely by the set of underdetermined equation systems and the solutions vectors to these equations. Here is a method that can describe a super-set of solutions that are all known to exist.

This enables a user to create any number of different meshes, combining the two sets of irregular edge networks. The automatic hexahedral meshing process is no longer a question of **"Can you hex mesh this shape, Y or N?"**, but instead **"Which mesh should I choose?"**

Unfortunately, the introduction of so many possibilities creates a new problem: How to find the "most suitable" mesh for a given engineering object? and "Why spend so much algorithmic effort to find the full set of irregular edge networks when 99.9% will be discarded anyway?"

Experience will establish good algorithms to reduce the amount of options created in the first place. Geometric characteristics can be used at an early stage to identify where the irregular edge networks must be placed for a well-formed mesh:
- a majority of the irregular edges must be placed on the boundary of the shape to achieve well-formed meshes. Identifying "which goes where" early will immediately reduce the number of internal flow sheet groups and the size of the equation systems.
- geometric characteristics on "chunkiness" can be used to position the irregular vertices in the volume at a point giving the best distribution of element shape quality scores across the adjacent elements. The outcome will be a list of flow line widths directly, then used as input to reduce the number of specified unknowns so the equation system can be solved.

So far, we have described the discovery of the patterns in the behaviour of quadrilateral and hexahedral meshes and the invention of algorithms to embed these patterns, leading to a consistent and complete representation of Coarse Hexahedral Clusters from a well-defined set of shapes, the 3D Kernels. This is called the Meshing process.

To successively create a cluster of polyhedra that eventually consist of 3D Kernels, a new mathematical notation has been developed blending together B-rep solid modelling and Graph Theory. The background for this is described in the following.

6. Euler's Polyhedral Formula - scalar representation of shapes

Leonard Euler /5/ was a polymath in the 18th century contributing significantly to a number of areas in mathematics. Among structural engineers he is known for the formulas for buckling capacity of columns /6/, for solid modellers he is known for the Euler's Polyhedron Formula /7/ that dates back to 1750:

$$\mathbf{v - e + f = 2} \tag{3}$$

Where

v is the number of vertices in a closed polyhedron

e is the number of edges in a polyhedron

f is the number of faces in a polyhedron

The formula relates the three components of the boundary of a topological structure, the vertices, edges and faces for a two-manifold topology. Any closed single B-rep Solid Model conforms to the formula.

However, it is a scalar equation, counting how many of each, irrespective of their type or relationship to other components of the shape topology.

7. Expansion of Euler's Polyhedral Formula – vector representation of shapes

Tessellation of solid models requires knowledge of what kind of face that is in the boundary: triangle, quadrilateral, pentagon and beyond and how many faces meet at a vertex, i.e., its valency. There will be two faces meeting at each edge, i.e., at this point we are dealing with two-manifolds.

The expanded Euler's Polyhedral Formula becomes:

$$\mathbf{V_3 + V_4 + V_5 + V_6 + \ldots - E + F_3 + F_4 + F_5 + F_6 + \ldots = 2} \tag{4}$$

Where

$\mathbf{V_i}$ is a vertex with i number of edges meeting at it (i-valent vertex)

E is an edge with two faces meeting at it (2-valent edge)

$\mathbf{F_i}$ is a face with i number of edges in the boundary (i-valent face)

The Fundamental Set of irregular edges is always present in any hexahedral mesh. Well-formed hexahedral meshes have the necessary number of irregular edges and no more. The minimum is defined by the 2D Kernels, triangles, quadrilaterals and pentagons forming the boundary of polyhedra with 3-valent vertices only. The general form in equation (4) can therefore be specialised.

The topological characteristics of these shapes are:
- 3-valent vertices
- 2-valent edges
- 3-, 4- and 5- valent faces

The subset defined by introducing these definitions in equation (4) is represented as

$$\mathbf{V_3 - E + F_3 + F_4 + F_5 = 2} \qquad (5)$$

We can now express the vertices and edges as a function of faces this way:
The number of $\mathbf{V_3}$ is three times the number of vertices in all the faces:

$$\mathbf{3\,V_3 = 3F_3 + 4F_4 + 5F_5} \qquad (6)$$

The number of edges **E** is twice the number of edges in all the faces:

$$\mathbf{2\,E = 3F_3 + 4F_4 + 5F_5} \qquad (7)$$

By combining equations (6) and (7) in equation (5), we get:

$$\mathbf{3\,F_3 + 2\,F_4 + 1\,F_5 = 12} \qquad (8)$$

Equation (8) describes all polyhedra with 3-valent vertices, 2-valent edges and bound by only triangles, quadrilaterals and pentahedra. There are 19 solutions to this equation, but only 11 are actual polyhedra, the proof is included in /4/. The solutions are shown in table 1. These 11 polyhedra are called the 3D Kernels and are shown graphically in figure 18.

The members of the 3D Kernels can be described as a vector of faces of different type, using the number of triangles, quadrilaterals and pentagons as a sequence. The short form chosen here uses these numbers with a hyphen between them, like (4-0-0) for a tetrahedron, (0-6-0) for a cube and (0-0-12) for a Dodecahedron, see figure 18.

#	T	Q	P	V	E	F	Exists	Name
1	4	0	0	4	6	4	x	Tetrahedron
2	3	1	1	5	9	6	-	
3	3	0	3	6	12	8	-	
4	2	3	0	5	9	6	x	Triangular Prism
5	2	2	2	6	12	8	x	NoName
6	2	1	4	7	15	10	-	
7	2	0	6	8	18	12	x	Triangular Truncated Trapezohedron
8	1	4	1	6	12	8	-	
9	1	3	3	7	15	10	x	Cube with a corner cut off
10	1	2	5	8	18	12	-	
11	1	1	7	9	21	14	-	
12	1	0	9	10	24	16	-	
13	0	6	0	6	12	8	x	Cube, Hexahedron
14	0	5	2	7	15	10	x	Pentagonal Prism
15	0	4	4	8	18	12	x	Cube with two edges cut off
16	0	3	6	9	21	14	x	Cube with three edges cut off
17	0	2	8	10	24	16	x	Quadrilateral Truncated Trapezohedron
18	0	1	10	11	27	18	-	
19	0	0	12	12	30	20	x	Pentagonal Truncated Trapezohedron

Where
the sequence number for the solutions
T number of Triangles in the boundary
Q number of Quadrilaterals in the boundary
P number of Pentagons in the boundary
V, E and **F** as before
Exists an x is placed in the row for each existing polyhedron
Name is the established technical term for the polyhedron in question, or an explanation of what it is in terms of v, e and f.

Table 1: The 19 solutions to equation (8) with the actual 11 high-lighted

The 3D Kernels include polyhedra that are recognised from elsewhere:
- three of the five classic Platonic Solids are present, the tetrahedron, the hexahedron and the dodecahedron;
- three Prisms, the triangular, the quadrilateral and pentagon prisms
- three Truncated Trapezohedra, the triangular, the quadrilateral and the pentagonal.

In addition, there are polyhedra that have no other affiliation

All of these have their respective F_F matrices and set of underdetermined equation systems, some with substantial G-matrices, /4/.

Counting vertices, edges and faces of different valency will give the same results as the scalar Euler's Polyhedral Formula.

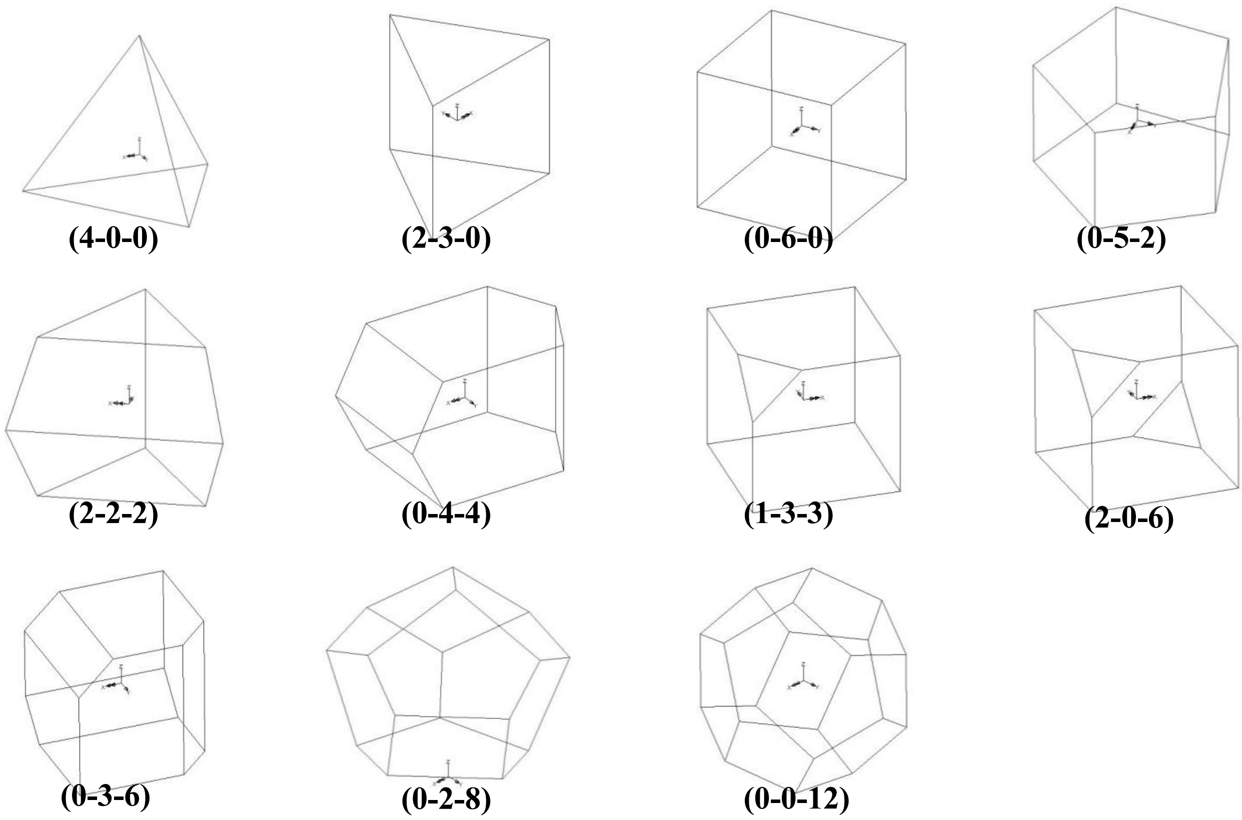


Figure 18: The 11 polyhedra forming the 3D Kernels.

8. Expansion of Euler's Polyhedral Formula – matrix representation of shapes

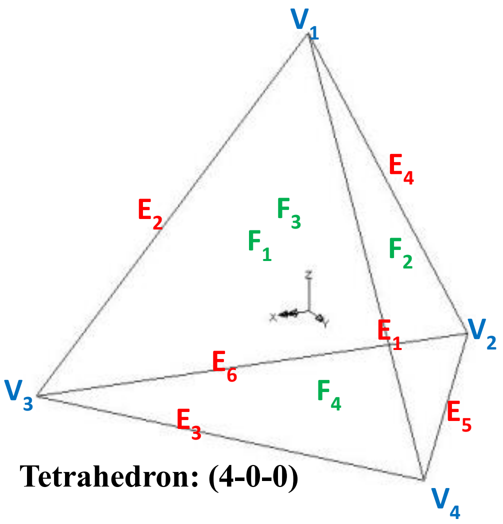
However, we need a better representation of a polyhedron so we can keep track of the connectivity in the polyhedron irrespectively of the faces in its boundary:
- the connectivity in a vertex, what type of faces is connected to the vertex
- the connectivity across the edges, what type of face is connected to the edge on either side.
- the connectivity between faces, what type of face is adjacent across a particular edge.

For this, we need to expand Euler's Polyhedral Formula further.

The relationships between the vertices, edges and faces in a polyhedron can be expressed as matrices that define the adjacency between them, for example the relationships between faces, i.e., the F_F matrix. These matrices are known as adjacency matrices. The adjacency matrices can be derived from the vector version of the Euler's Polyhedral Formula, /4 /.

The added complexity of this expansion is that the sequencing of the instances in the boundary comes into play. Vertices, edges and faces must have a position in a sequence so they can get their own row and column.

The relevant adjacency matrices for a Tetrahedron (4-0-0) will be used to make an important point.



Tetrahedron: (4-0-0)

$$F_F = \begin{bmatrix} . & 1 & 1 & 1 \\ 1 & . & 1 & 1 \\ 1 & 1 & . & 1 \\ 1 & 1 & 1 & . \end{bmatrix} \qquad E_F = \begin{bmatrix} 1 & 1 & . & . \\ 1 & . & 1 & . \\ 1 & . & . & 1 \\ . & 1 & 1 & . \\ . & 1 & . & 1 \\ . & . & 1 & 1 \end{bmatrix}$$

$$V_E = \begin{bmatrix} 1 & 1 & . & 1 & . & . \\ . & . & . & 1 & 1 & 1 \\ . & 1 & 1 & . & . & 1 \\ 1 & . & 1 & . & 1 & . \end{bmatrix} \qquad F_V = \begin{bmatrix} 1 & 1 & 1 & . \\ . & 1 & 1 & 1 \\ 1 & . & 1 & 1 \\ 1 & 1 & . & 1 \end{bmatrix}$$

Figure 19: A Tetrahedron with a naming sequence and four of the corresponding incidence matrices

The adjacency matrices shown in Figure 19 can all be extracted from the F_F matrix or created by interrogating the data structure holding the shape. However, these adjacency matrices relate to each other in an algorithmic way. The following relationships exist:

$$\mathbf{F_D = F_V * V_E \qquad \text{where} \quad F_D = F_E{}^A + 2{*}F_E{}^O} \tag{9}$$

where

$F_E{}^A$ is a Face-Edge Adjacent matrix

$2{*}F_E{}^O$ is a Face-Edge Opposite matrix times two

Further by extracting the $\mathbf{F_E{}^A}$ matrix:

$$\mathbf{F_F + D = F_E{}^A * (F_E{}^A)^T} \tag{10}$$

Where

D is a diagonal matrix with the diagonal values equal to the number of edges in the respective face

$\mathbf{(F_E{}^A)^T}$ is the Face-Edge adjacency matrix transposed

or

$$\mathbf{F_F = F_E{}^A * (F_E{}^A)^T - D} \tag{11}$$

These relationships between the adjacency matrices also applies to a cluster of cells, and equations (9), (10) and (11), allow the development of algorithms to create the Face-Face adjacency matrix when the Face-Vertex and Vertex-Edge matrices are known.

The algorithm described above is simple and easy to implement in an FEA pre-processor to alleviate the manual effort needed to create “meshable cells” by hand. The algorithm works on clusters of cells as in Cellular Non-Manifold B-rep Solid Models.

Both the vector expansion and matrix expansion of Euler’s Polyhedral Formula contains the scalar information of a polyhedron. This can be verified by counting the number of each entity instance as before.

9. A mathematical continuum

The adjacency matrix algorithm is also the core algorithm in the process of splitting a B-rep Solid Model into smaller and simpler polyhedra, examples of planar cut operators are shown in Figures 7, 8 and 9.

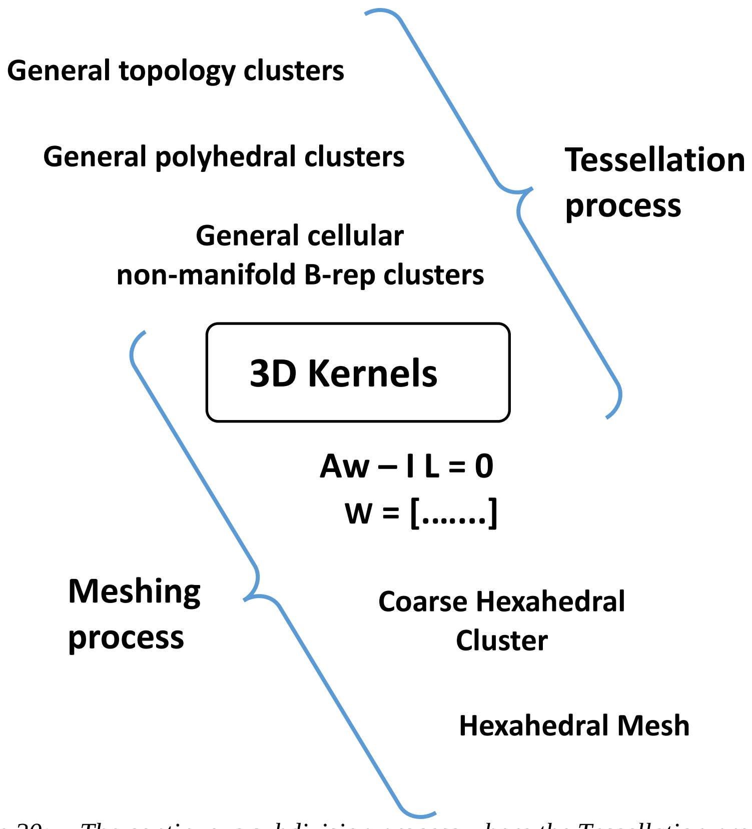


Figure 20: The continuous subdivision process where the Tessellation process meets the Meshing process at the 3D Kernels.

The Topology Operators available to tessellate a shape can start from the original B-rep Solid Model and extract the topology information into the mathematical representation used to hold the adjacency relationships for the cluster of cells. A planned sequence of Tessellation Operators will reduce the initial topological complexity of the shape to a cluster of less complex topologies that eventually consists of the 3D Kernels. There are a number of different operators that can contribute to a tessellation into 3D Kernels and beyond, a Coarse Hexahedral Cluster, /8/.
This is where the Tessellation process meets the Meshing process, creating a two-part continuous process of the subdivision process from the B-rep Solid Model to the Coarse Hexahedral Cluster as shown in figure 20.

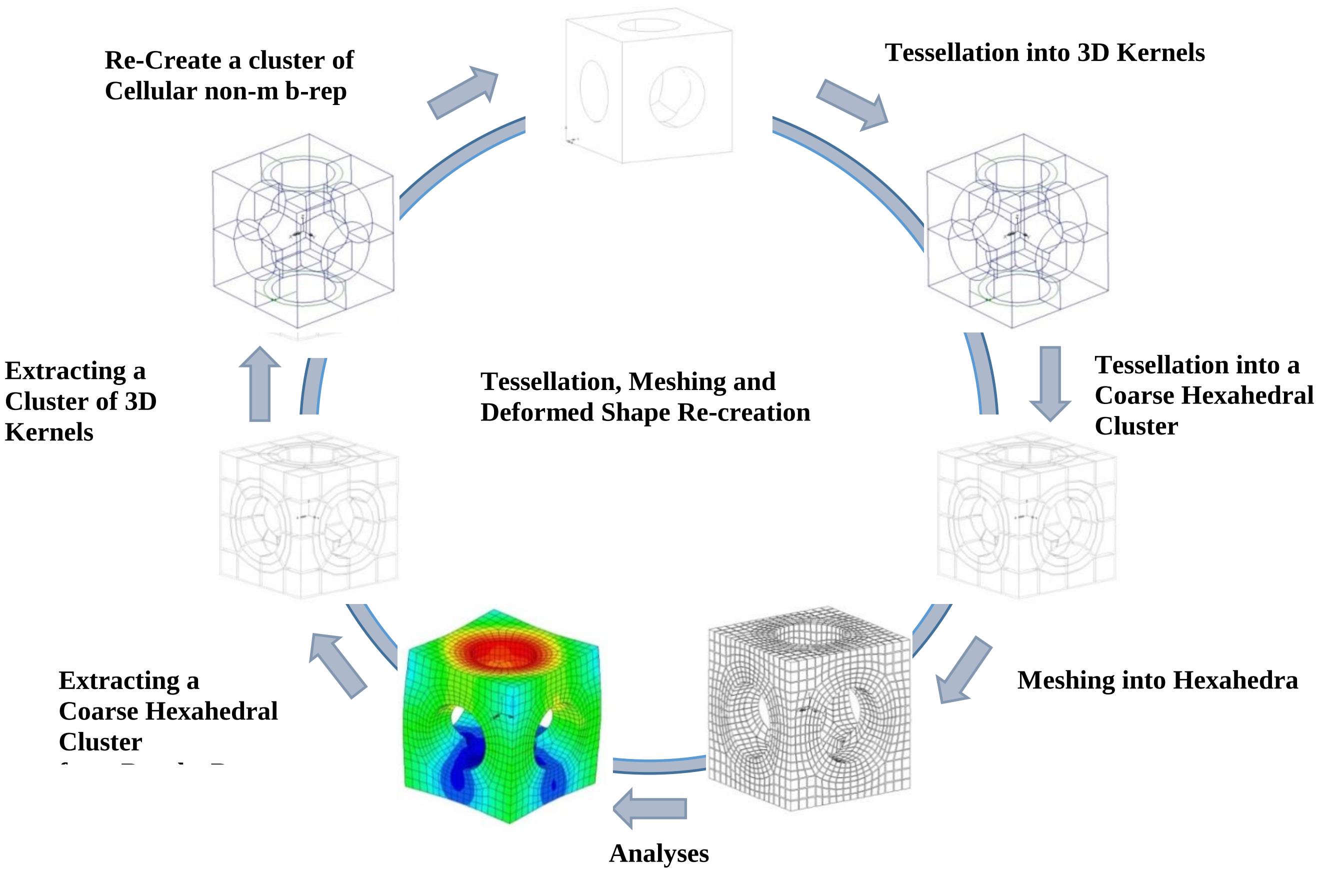


Figure 21: The Tessellate and Merge Operators together form a loop

The Tessellation and Meshing processes create a well-defined set of topologies, resulting in a finite element mesh with well-defined element flows. The process can be reversed, i.e., a set of Merge Operators can identify the structure in a finite element mesh and separate the Fundamental Set and Auxiliary Set of irregular edge networks to create a Coarse Hexahedral Cluster and the 3D Kernels. Other Merge Operators can extract the larger topologies from a cluster of smaller ones and eventually end up with a B-rep Solid Model again, /10/. This is shown in principle in Figure 21.

10. Discussion

The discovery of the significance of the irregular vertices in polygons and irregular edge networks in polyhedra opens up a new way of specifying quadrilateral and hexahedral meshes. A user can define the Volume Mesh Controls to position the irregular edges and internal vertices in the volume directly, and the external quadrilateral carpet and edge division numbers are only algorithm outputs.

Dislocation Meshing is a super-set meshing technology, spanning out a subset of a large group of possible hexahedral meshes in a cluster of cells using both the Fundamental and the Auxiliary Sets of irregular edge networks. Within the complete space of alternatives, Dislocation Meshing is designed to focus on:
- the subset that uses irregular edge networks defined by the 3D Kernels only.
- the necessary number of irregular edges defined by the Fundamental Set.

As a consequence, a large number of alternative irregular edge networks are excluded; mainly those with lower quality mesh flows and poorly shaped elements, often created by excessive use of the Auxiliary Set. In the remaining subset Dislocation Meshing can create a large number of hexahedral meshes focusing on well-formed mesh flows and well-formed elements, /4/.

Other efforts in the domain of automatic hexahedral meshing can be measured by their effectiveness in creating any subset of these known solutions. Where these methods cannot create hexahedral meshes consistently (or not at all), the theory underpinning Dislocation Meshing can be used to explain why, /9/.

11. Conclusion

Dislocation Meshing is founded on well-established mathematical theories and expands these to a natural mathematical language for representation and manipulation of surface and solid models using Boundary-Representation Solid Model and Graph Theory notations. Dislocation Meshing is one of many applications of a general mathematical representation of shape, an application independent theory that is surprisingly versatile. /10/.

The resulting algorithms for Dislocation Meshing can be implemented in any computer environment that uses B-rep Solid Model notation. Hence, it can fill the hole for hexahedral meshing in existing FEA pre-processors. And even better: Tessellation and Meshing can become solid modelling operations working directly on parameterised solid models, eliminating data transfer all together. Any irregular edge network and its corresponding Coarse Hexahedral Cluster can be used unchanged in a sequence of mesh-refinements based on h-convergence for a verification study as part of a Verification and Validation process.

The Tessellation and Meshing strategies described here require High Performance Computing (HPC) support to be a practical tool. The speed by which sequences of FEA solver input files can be created requires effective HPC implementation of the solvers for the subsequent analyses. The harvesting of the results from large search spaces requires relational databases and HPC graphics.

Dislocation Meshing is the missing link between the past and the future:
- it fits in existing software architectures and can revive forgotten technologies like p- and hp-versions of FEA for accuracy and sub-structuring for reduced clock wall-time when solving large equation systems;
- it is an enabling technology to fully exploit the speed of emerging computer technology, like Pica FLOPS machines, /11/, Exa FLOPS machines, /12/ and whatever comes next, /13/.
What do you plan to do today so you are ready when Exa FLOPS machines arrive?

12. References

There is an unending list of possible references for the topics covered in this paper. Instead of selecting an arbitrary reference in a long list of possibilities, here is a modern way of directing readers to authoritative sources for further reading: to use the Internet, specifically Wikipedia which gives a clear and concise presentation of most topics with a list of references for further reading.

/1/ MacNeal, Dr Bruce E and MacNeal, Dr Richard H.: (1997) “Future Issues – A Code Developer’s Perspective”, Proceedings of NAFEMS World Congress ’97, Volume1, NAFEMS, pages 84-94.

/2/ About Boundary-Representation Solid Models: https://en.wikipedia.org/wiki/Boundary_representation

/3/ About Solid Modelling: https://en.m.wikipedia.org/wiki/Solid_modeling

/4/ Aas, Jon (2019) “Dislocation Meshing – Theory Basis”, CT Innovations Ltd, internal document.

/5/ About Leonard Euler: https://en.wikipedia.org/wiki/Leonhard_Euler

/6/ About Euler’s buckling capacity of columns: https://en.wikipedia.org/wiki/Euler%27s_critical_load

/7/ About Euler’s Polyhedral Formula: https://plus.maths.org/content/eulers-polyhedron-formula

/8/ Aas, Jon (2019) “Dislocation Meshing – Implementation strategies”, CT Innovations Ltd, internal document.

/9/ Aas, Jon (2019) “Dislocation Meshing – Comparison with other Automatic Hexahedral Meshing Technologies”, CT Innovations Ltd, internal document.

/10/ Aas, Jon (2019) “Topology Operator Theory – What it can be used for”, CT Innovations Ltd, internal document.

/11/ about Pica FLOPS computing: https://www.top500.org/news/summit-up-and-running-at-oak-ridge-claims-first-exascale-application/

/12/ about Exa FLOPS computing: https://www.nextbigfuture.com/2017/07/update-on-the-race-to-the-exaflop-supercomputer.html

/13/ about FLOPS: https://en.wikipedia.org/wiki/FLOPS