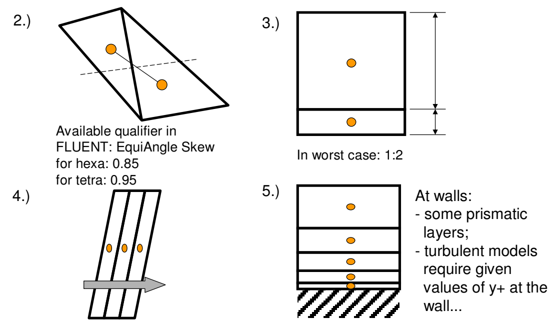
Meshing

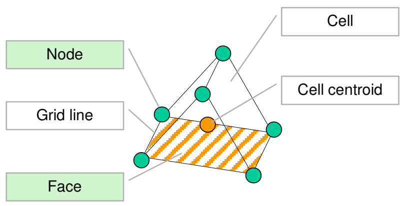
Dr. Gergely Kristóf
20-th April 2008.

Mesh quality

1.) Refine the mesh in those regions where you expect large gradients. Gradient based refinement is also available in FLUENT...



Elements of the numerical mesh



Content of the mesh file:
-Node positions
-Faces (referring to the order number of nodes)

Field variables are cell centroids in FLUENT system
Grid lines are straight sections.

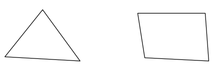
Edge meshing

- Size specification
- Refinement (with or without meshing)
- Copying the edge mesh
- Changing edge direction
- Soft link, hard link

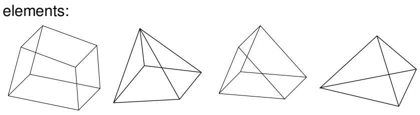
Geometrical models allowed in FLUENT:
- 2D (with slab symmetry)
- 2D axisymmetric
- 2D axisymmetric with swirl
- 3D

Element types in Gambit:

2D elements:



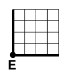
3D elements:



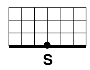
FLUENT can handle elements of arbitrary number of sides.
Problem: automatic generation of high quality mesh.

Face vertex types

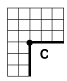
End (E): 0<szög<120



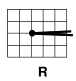
Side (S): 120<szög<216



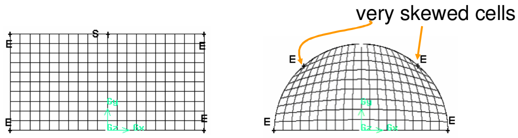
Corner (C): 216<szög<309



Reverse (R): 309<szög<360

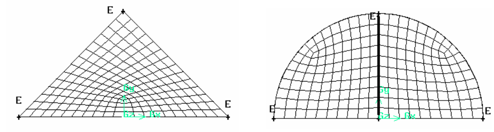


Hex – Map method



Vertex types allowed: **4*End+N*Side**
For periodic faces: **N*Side**
Oblong topology.
Internal mesh is controlled by edge meshes...

Quad - Tri-primitive method

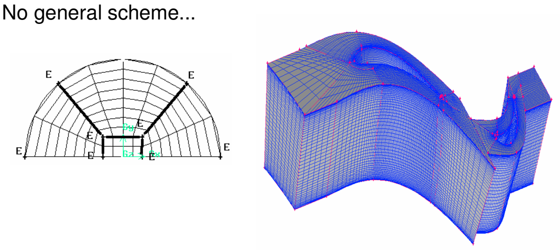


Vertex types allowed: **3*End+N*Side**
Meshes faces of triangular topology with quad cells.

Block structured mesh

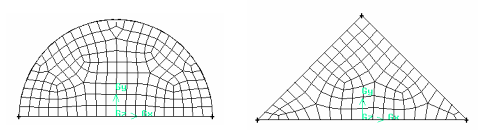
1. Decomposition of the domain into map-able blocks.
2. Block-wise mapping.

No general scheme...



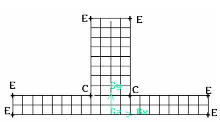
eg. turbine cascade

Quad – Pave method



- Face vertex types are ignored (excepting “T”);
- Fully automatic: size functions can be used for controlling the mesh size;
- Even number of intervals required on every edges;
- Usually creates asymmetric mesh on symmetric faces;
- Ensures higher quality close to the boundary.

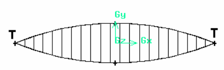
Quad – Submap method



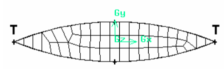
Vertex types allowed:
4*End+L*Side+M*(End+Corner)+N*(2*End+Reverse)
For periodic faces:
N*Side+M*(2*End+2*Corner)
Similar to Map method but it can leave out some blocks.

Some rarely used methods for quad meshing

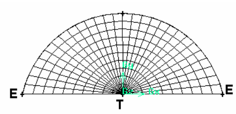
Quad - Tri Map



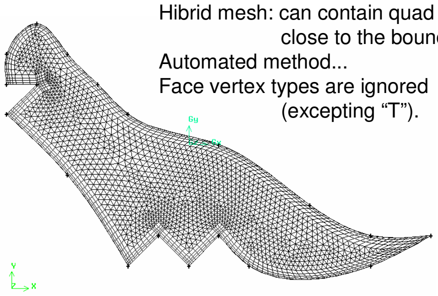
Quad - Tri Pave



Quad - Tri Wedge

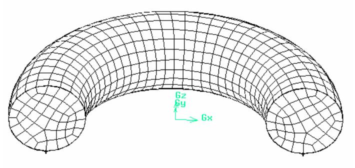


Tri – pave method



Hibrid mesh: can contain quad elements close to the boundary
Automated method...
Face vertex types are ignored (excepting "T").

Cooper method



Mesh of the source faces is pushed through the volume.
Side surfaces must be et least submapable.
It can generate prismatic (wedge) elements as well.

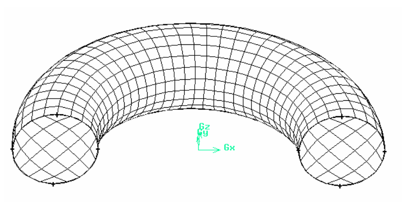
Creating a prismatic wall layer (boundary layer meshing)

- y+ in the center of the first cell must be smaller than 500;
- Sudden change in grid size is to be avoided;
- Sometimes 3 directional wall refinement is necessary (eg. for LES);
- Internal continuity;
- Face vertex types...

Some other method

- Hex – Submap
  Similar to quad-submap.
- Hex – Tet-primitive
  Similar to tri-primitive. Works on tetrahedral domain.
- Tet/hibrid – Tgrid
  Similar to tri-pave method.
  It can use prismatic elements close to the boundary (boundary layer mesh)
  Resolution can be controlled by size functions.

Hex-map method (3D)

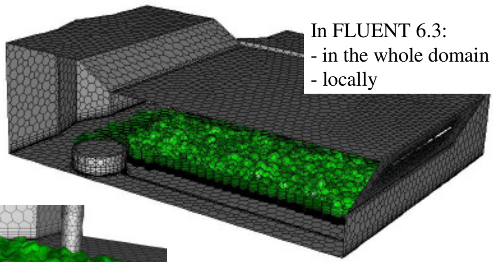


Similar to the quad-map (2D) method
Contour surfaces must be mapped. It requires 8 End vertices.
Complex volumes must be decomposed...

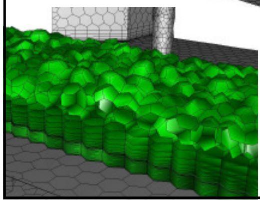
Size functions

- Source, attachement...
- On the basis of curvature or smallest gap...
- Preferably only one simple source object per size function (and only one size function per volume)

Dual (Polihedral) mesh

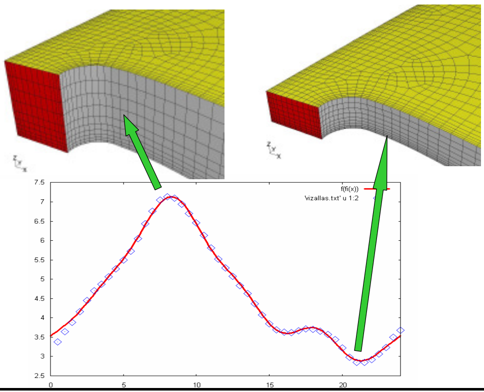


In FLUENT 6.3:
- in the whole domain
- locally

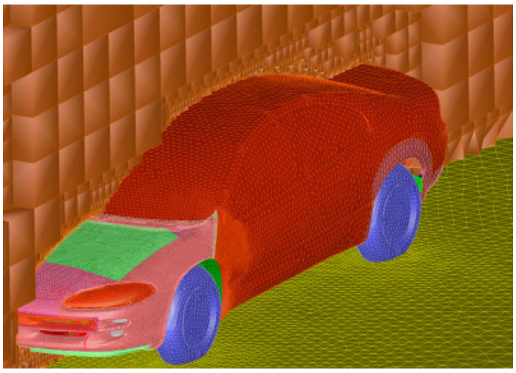


- Some 6 fold reduction of the number of cells
- Decreases accuracy (but not that much)
- Improves mesh quality (skewnes)
- About the same computing time/iteration.
- Accelerates the convergence (2 or 3 times).

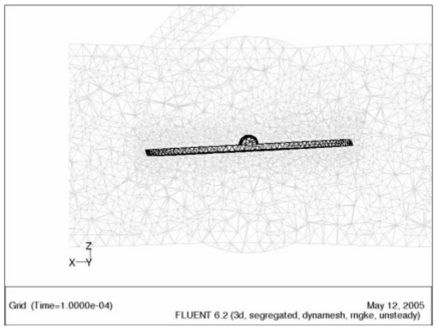
Deforming meshes #1



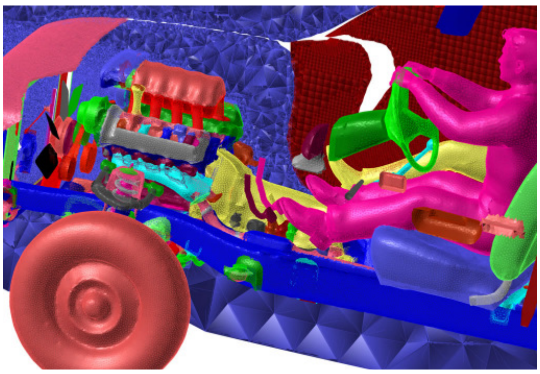
Hex-core



Deforming meshes #2



Tgrid – Surface wrapper



Airfioil database

http://www.ae.uiuc.edu/m-selig/ads/coord_database.html