HPC Computer Aided Engineering @ CINECA

Raffaele Ponzini Ph.D.

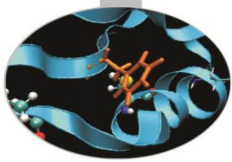
CINECA

SuperComputing Applications

and Innovation Department – SCAI

16-18 June 2014

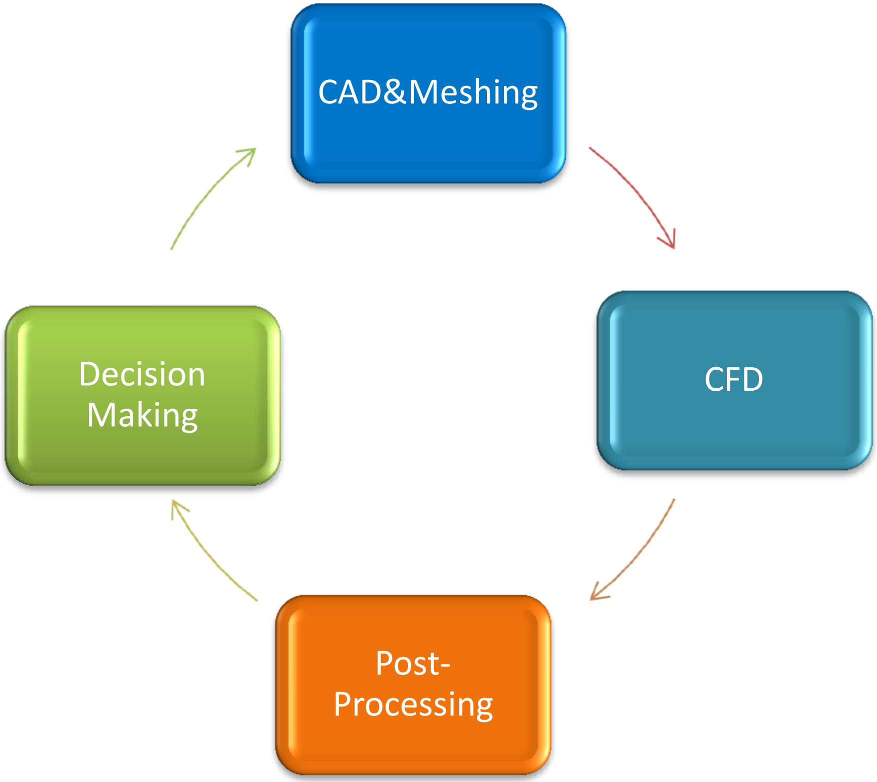
Segrate (MI), Italy



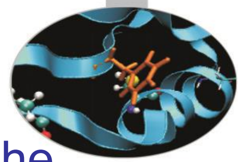
Outline

- Open-source CAD and Meshing tools
- Meshing main concepts
- Meshing tools for OpenFOAM

Overview



Geometry



- The starting point for all problems is a "geometry" that is in few words the shape of the problem to be analyzed or in other words the computational domain defined by the problem
- Geometry is managed through Computer Aided Design tools (CAD)
- Geometries can be created top-down or bottom-up.
- Top-down refers to an approach where the computational domain is created by performing logical operations on primitive shapes such as cylinders, bricks, and spheres.
- Bottom-up refers to an approach where one first creates vertices (points), connects those to form edges (lines), connects the edges to create faces, and combines the faces to create volumes.
- Advanced CAD tools have parametric description of the curves used to build the geometry
- CAD allows to define a virtualized description of the geometry components in terms of volumes, surfaces (faces), lines (edges) and points (vertices or nodes)

CAD tolerance and file format standards are still an issue

Mesh

- The computational mesh is the discretized counterpart of the geometry CAD
- The mesh cells are the atomic elements in which the physics of the flow is solved.
- Cells are grouped and labeled according to different properties of the numerical boundaries defined for the problem (boundaries or patches in OpenFOAM)
- The mesh has a crucial impact on the CFD modelling since the overall workflow is build on top of the mesh
- Mesh quality parameters evaluation is crucial to move forward in a CFD modelling workflow

Open-source CAD and Meshing tools

CAD
(geometry design; parametric)

Geometry import
(pre-processor environment; tolerances)

Geometry clean-up for meshing

Open-source CAD and Meshing tools

Mesh topology design

↓

Mesh generation

↓

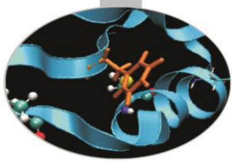
Mesh quality check

↓

CFD solver

This workflow can be iterated over and over in order to found a proper mesh for a selected CFD solver

Open-source CAD and Meshing tools



- Open source CAE resources CAELINUX: http://www.caelinux.com/CMS/
  - SALOME: http://www.salome-platform.org/
- freeCAD: http://freecadweb.org/

- Meshing:
  - GMSH: http://www.geuz.org/gmsh/
  - SALOME: http://www.salome-platform.org/

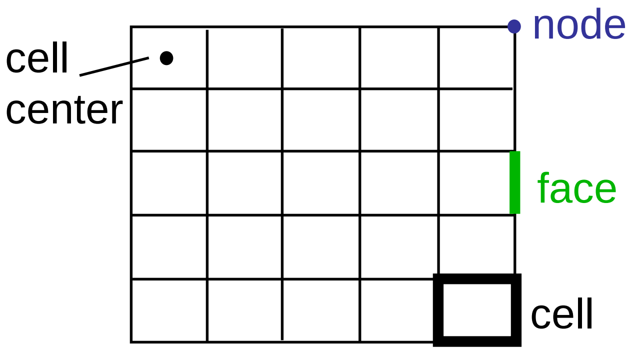
We are not interest in this course on dealing with CAD issues

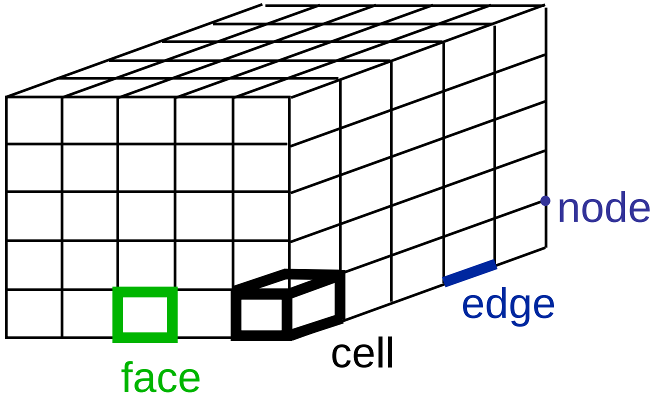
Mesh

There are several type of mesh that differs for the topology of the elements used to discretize the domain:

1. Single-block, structured grid
2. Multi-block, structured grid
3. Unstructured grid
4. Unstructured Tetrahedral grid
5. Hybrid grid

Terminology

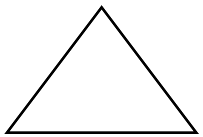




SCAI
SuperComputing Applications and Innovation

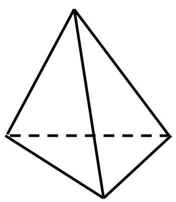
Mesh

- Many different cell/element and grid types are available. Choice depends on the problem and the solver capabilities.
- Cell or element types:

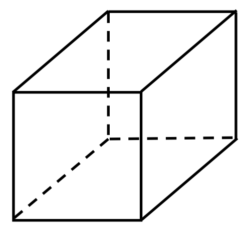


triangle

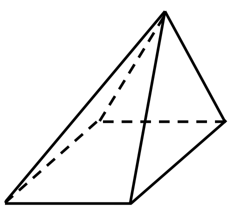
quadrilater



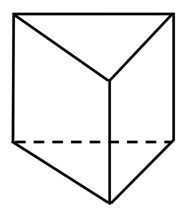
tetrahedron



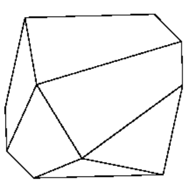
prism with quadrilateral base
(**hexahedron** or “**hex**”)



pyramid



prism with triangular base
(**wedge**)



arbitrary polyhedron

PRACE CINECA

Mesh

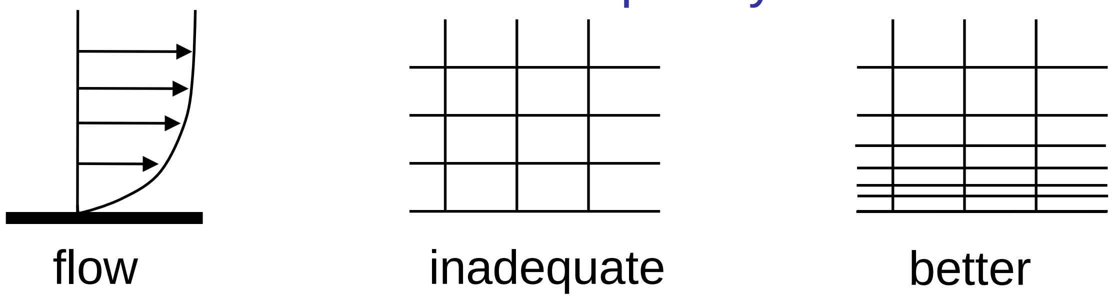
- Tri mesh: mesh consisting entirely of triangular elements.
- Quad mesh: consists entirely of quadrilateral elements.
- Hex mesh: consists entirely of hexahedral elements.
- Tet mesh: mesh with only tetrahedral elements.
- Hybrid mesh: mesh with one of the following:
  - Triangles and quadrilaterals in 2D.
  - Any combination of tetrahedra, prisms, pyramids in 3D.
  - Boundary layer mesh: prisms at walls
  - Hexcore: hexahedra in center and other cell types at walls.
- Polyhedral mesh: consists of arbitrary polyhedra.

Mesh quality

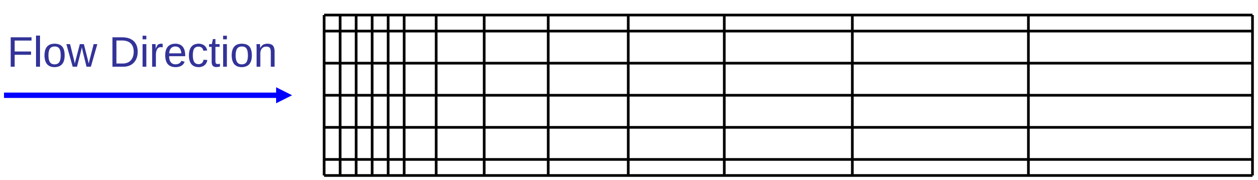
- For the same cell count, hexahedral meshes will give more accurate solutions, especially if the grid lines are aligned with the flow.
- The mesh density should be high enough to capture all relevant flow features.
- The mesh adjacent to the wall should be fine enough to resolve the boundary layer flow. In boundary layers, quad, hex, and prism/wedge cells are preferred over tri's, tets, or pyramids.
- Three main measures of quality (depends on pre-processor):
  - Skewness
  - Aspect ratio
  - Non orthogonality

Mesh design

- Pertinent flow features should be adequately resolved.

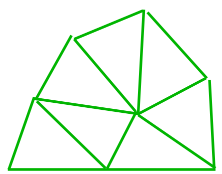


- Cell aspect ratio (width/height) should be near one where flow is multi-dimensional.
- Quad/hex cells can be stretched where flow is fully-developed and essentially one-dimensional.

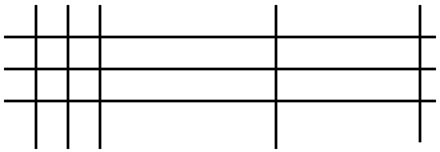


Mesh design

- Change in cell/element size should be gradual

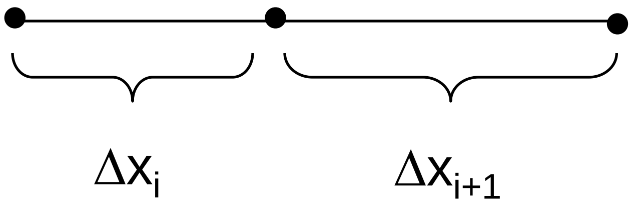


smooth change in cell size



sudden change in cell size

- Ideally, the maximum change in grid spacing should be <30%:



$$\frac{\Delta x_{i+1}}{\Delta x_i} \leq 1.3$$

Mesh design

- More cells can give higher accuracy (if well spent). The downside is increased memory and CPU time for mesh creation (complexity)
- To keep cell count down:
  - Use a non-uniform grid to cluster cells only where they are needed.
  - Use solution adaption to further refine only selected areas.
- Cell counts of the order:
  - 1E5 small/intermediate suitable for desktop pc and for debugging of solver setup
  - 1E6 regular day by day work
  - 1E7 expensive, suitable for HPC centers
  - 1E9 hardly manageable (my 2 cents experience with billion cells mesh: http://www.ansys.com/staticassets/ANSYS/staticassets/resourcelibrary/article/AA-V3-I2-Sailing-Past-a-Billion.pdf)
- Time mesh generation can be non-negligible

Main sources of errors

- Mesh too coarse
- High skewness
- Large jumps in volume between adjacent cells
- Large aspect ratios
- Non orthogonal cells
- Inappropriate boundary layer mesh

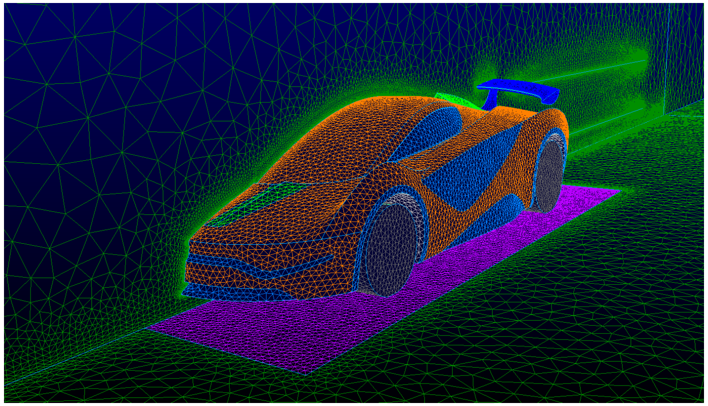
Meshing take home message

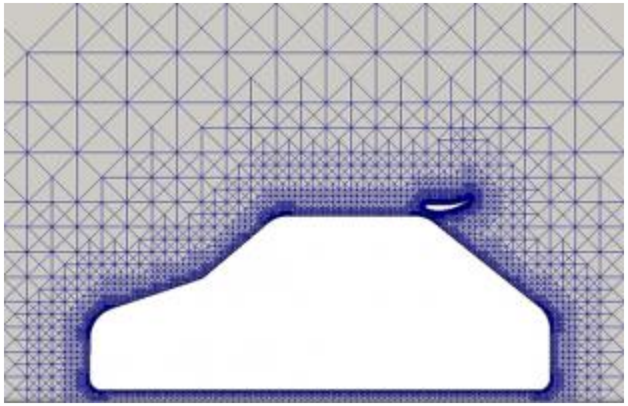
Appropriate choice of grid type depends on:

1. Geometric complexity (geometry is king)
2. Flow field patterns (BC are queens)
3. Cell and element types supported by solver (need to verify your mesh into the solver)

Meshing tools for openFoam

- Commercial: Pointwise (Pointwise Inc.)
- Open-source: snappyHexMesh (OpenCFD ltd.)





Pointwise

- Site: http://www.pointwise.com/
- General purpose pre-processor for any CFD code (commercial or open-source)
- Suitable to handle also very complex geometry
- GUI equipped
- User-friendly
- Serial
- Scriptable (glyph)
- Rich documentation (tutorials, webinar,…)
- Strongly oriented to automation
- Sophisticated visual mesh quality check (his own metrics)
- Support CAD import (most formats)
  and simple geometry creation

snappyHexMesh

- Natural pre-processor of openFOAM
- No GUI
- Hard to use with very complex geometry
- Quite slow learning curve
- Parallel
- Scriptable by definition
- Exhaustive mesh quality information
- Support stl CAD import and simple geometry creation
- Other geometry manipulation tools are available within the openFOAM toolbox

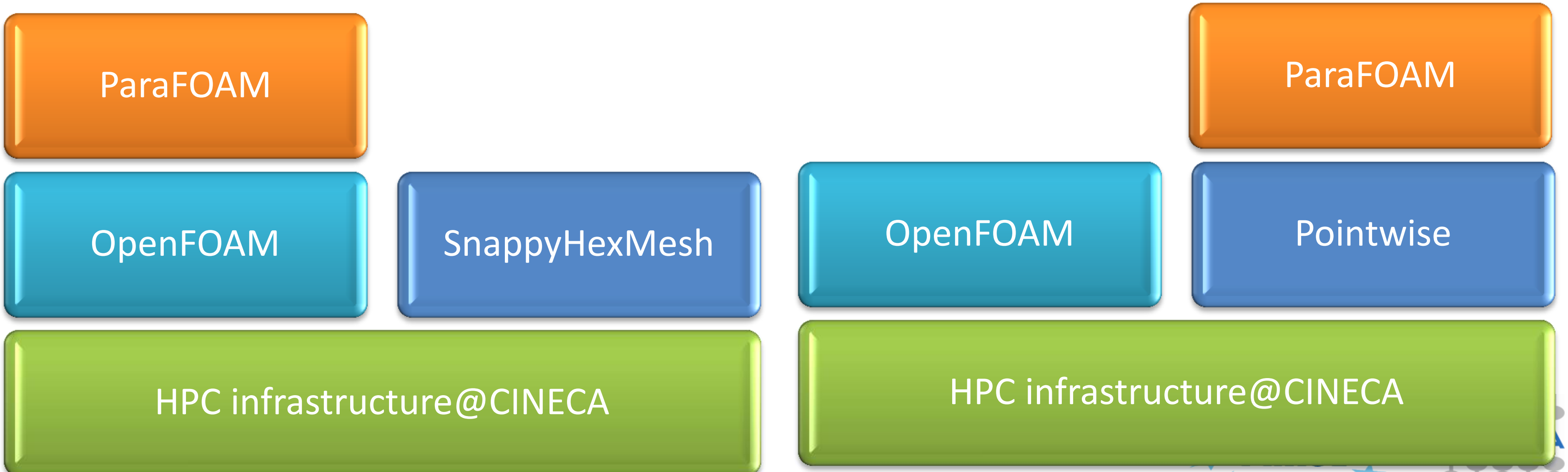
Pointwise vs snappyHexMesh

Two different philosophies:

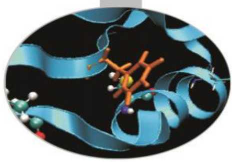
- snappyHexMesh:
  - from boundaries of the domain to object geometry
  - trying to satisfy quality criteria (user driven) during the mesh creation
- Pointwise:
  - from object geometry surface up to boundary domains
  - A-posteriori mesh quality check

snappyHexMesh or Pointwise

- Both tools are well suited in our experience to be used with success in HPC platforms for automated and productive workflow in industrial applications.
- Pro’s and con’s should be evaluated case by case
- License business model involved by Pointwise is not a limiting factor in that the cost is small (order of 2k euros) compared to usual CFD solver license cost (order of 10k -100k euros)



Paraview for mesh visualization



In the next tutorial you will learn how to use snappyHexMesh to build meshes suitable to perform CFD analysis in openFOAM.

In order to visualize the resulting mesh generated using snappyHexMesh you can use Paraview.