# Geometry modeling and Meshing

## Numerical Flow Simulation

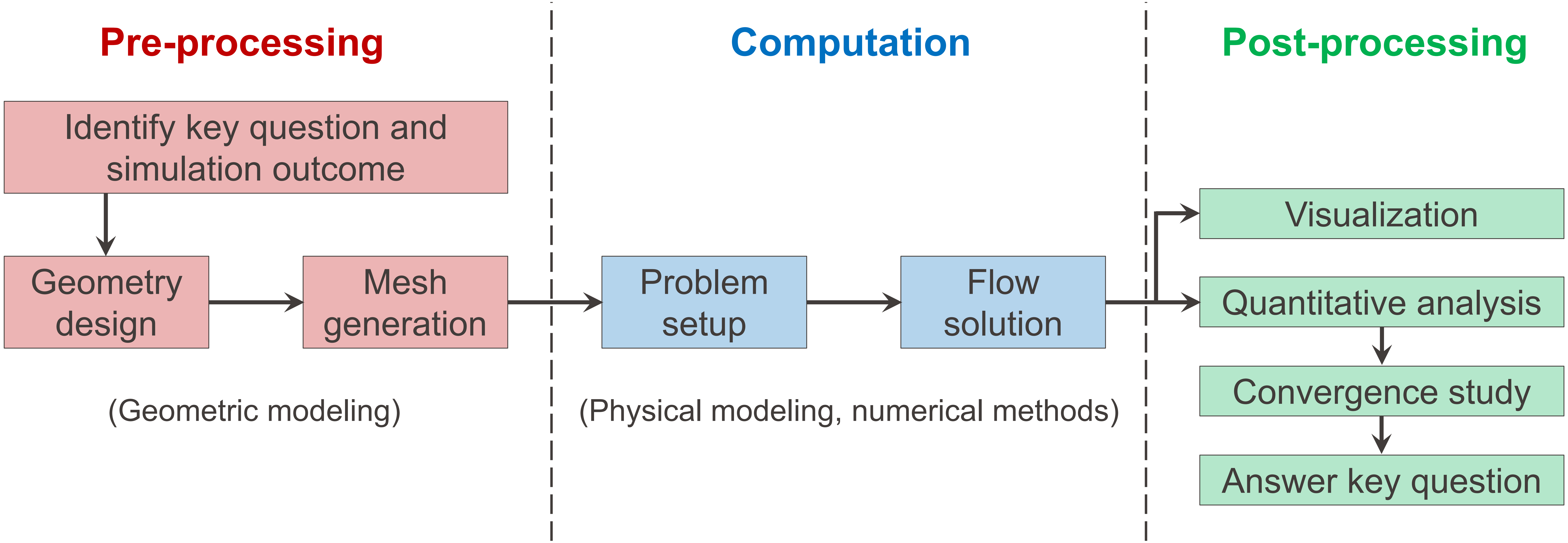
Edouard Boujo

Fall 2025

École
polytechnique
fédérale
de Lausanne

# Numerical simulation workflow

- Reminder:



**Pre-processing**

**Computation**

**Post-processing**

Identify key question and simulation outcome

Geometry design

Mesh generation

(Geometric modeling)

Problem setup

Flow solution

(Physical modeling, numerical methods)

Visualization

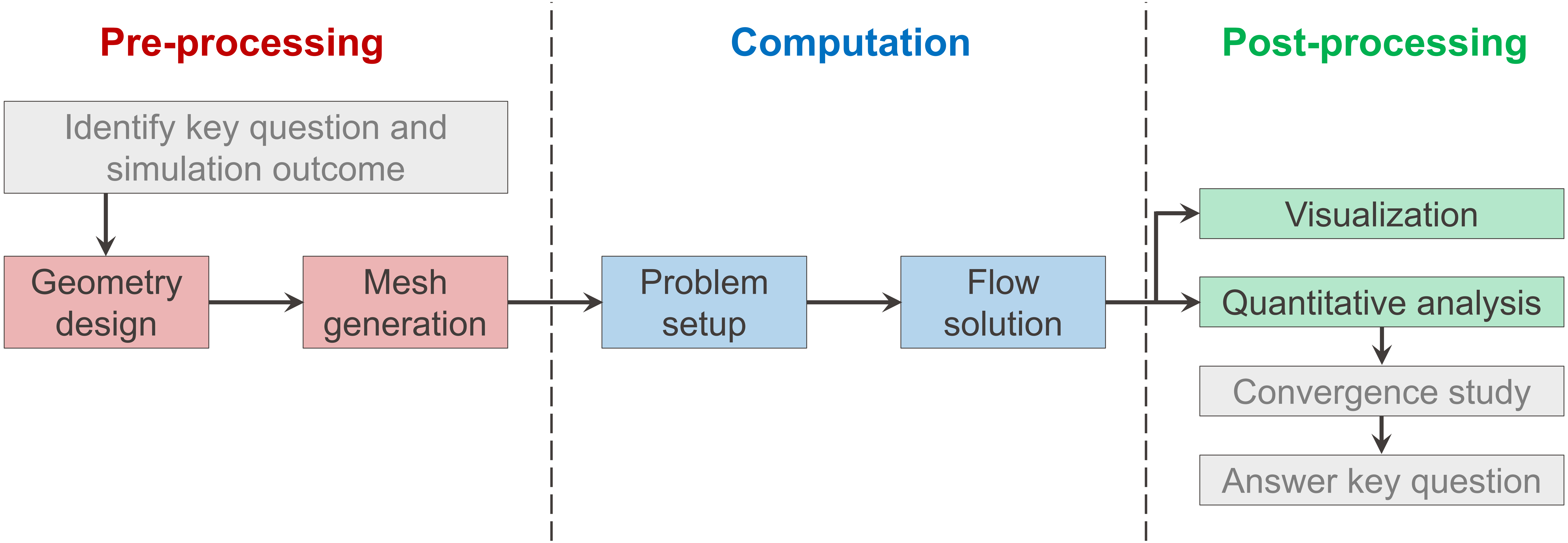
Quantitative analysis

Convergence study

Answer key question

# Numerical simulation workflow

- Some of the software available in Ansys:



**Pre-processing**

**Computation**

**Post-processing**

Identify key question and simulation outcome

Geometry design

Mesh generation

Problem setup

Flow solution

Visualization

Quantitative analysis

Convergence study

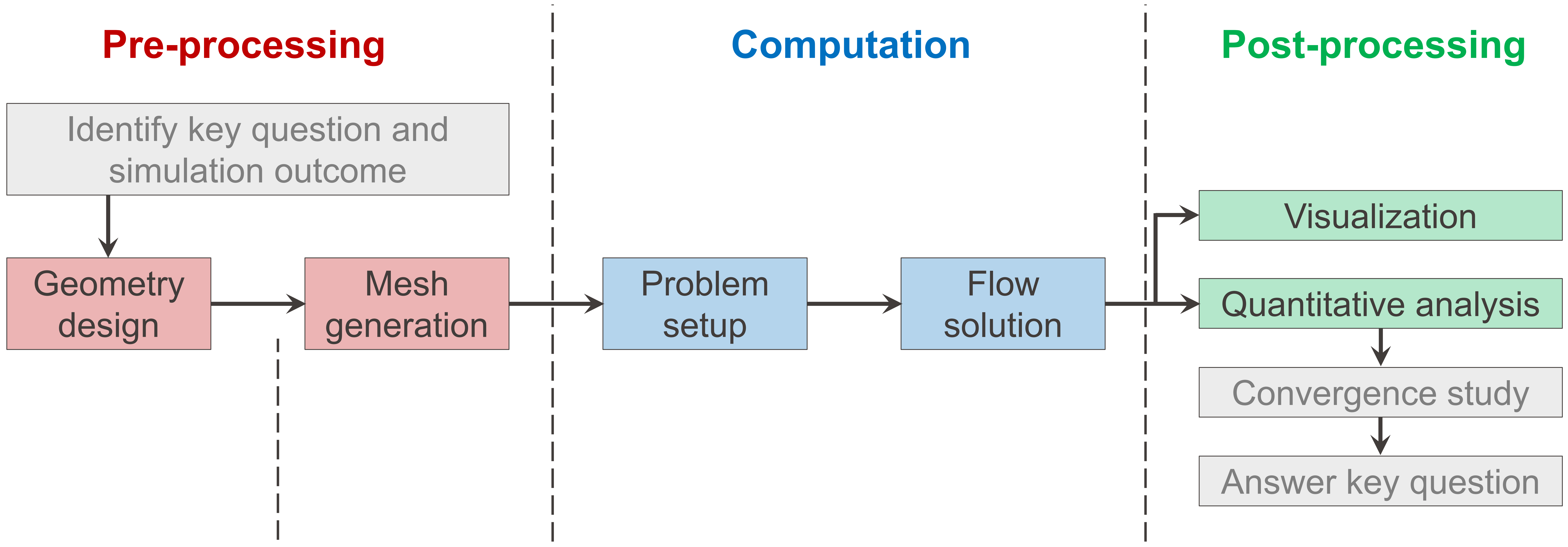
Answer key question

- Design Modeler (soon obsolete)
- SpaceClaim*
- External soft

(*) Available both in Workbench and stand-alone

# Numerical simulation workflow

- Some of the software available in Ansys:



**Pre-processing**

**Computation**

**Post-processing**

Identify key question and simulation outcome

Geometry design

Mesh generation

Problem setup

Flow solution

Visualization

Quantitative analysis

Convergence study

Answer key question

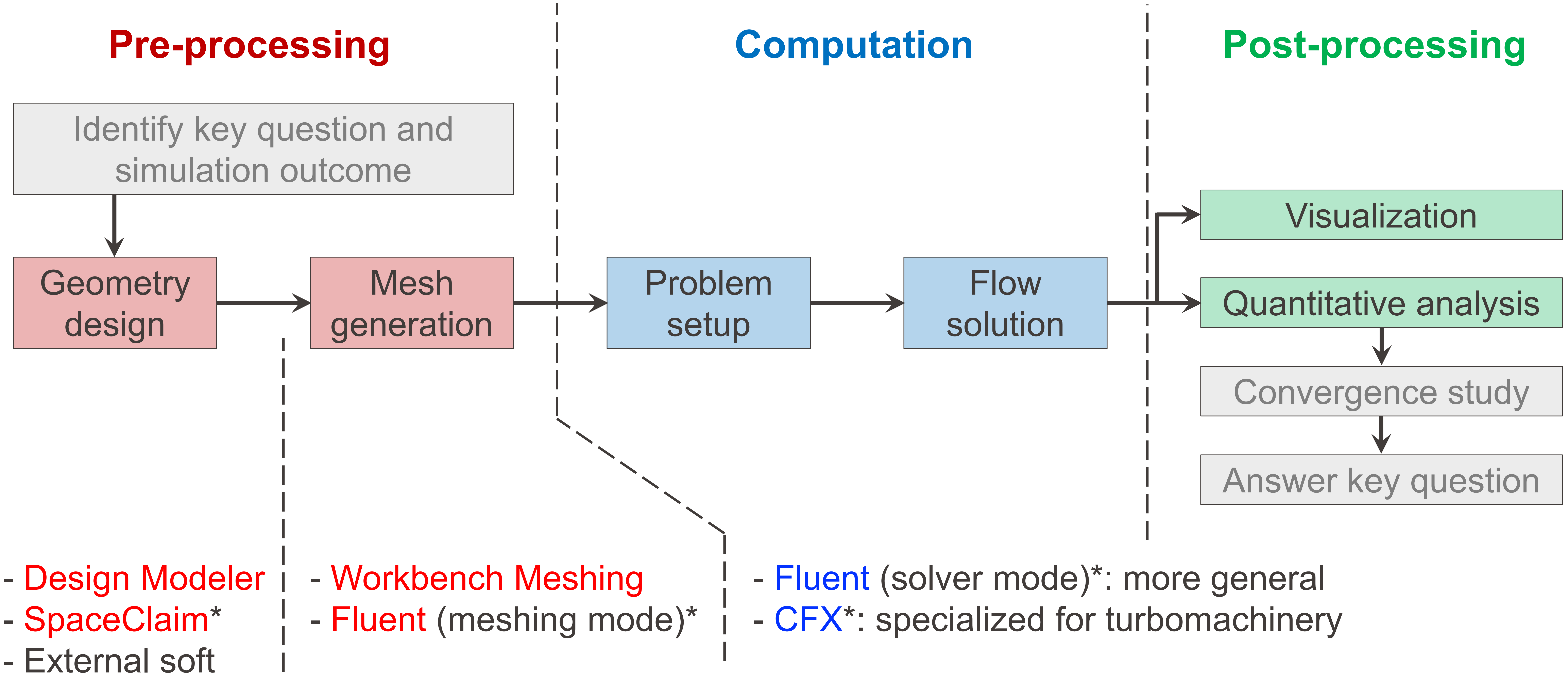
- Design Modeler
- SpaceClaim*
- External soft

- Workbench Meshing: more control, but steeper learning curve. Both 2D and 3D.
- Fluent (meshing mode)*: easier to learn, but less control. Only 3D

(*) Available both in Workbench and stand-alone

# Numerical simulation workflow

- Some of the software available in Ansys:



**Pre-processing**

**Computation**

**Post-processing**

Identify key question and simulation outcome → Geometry design → Mesh generation → Problem setup → Flow solution → Visualization / Quantitative analysis → Convergence study → Answer key question

- Design Modeler
- SpaceClaim*
- External soft

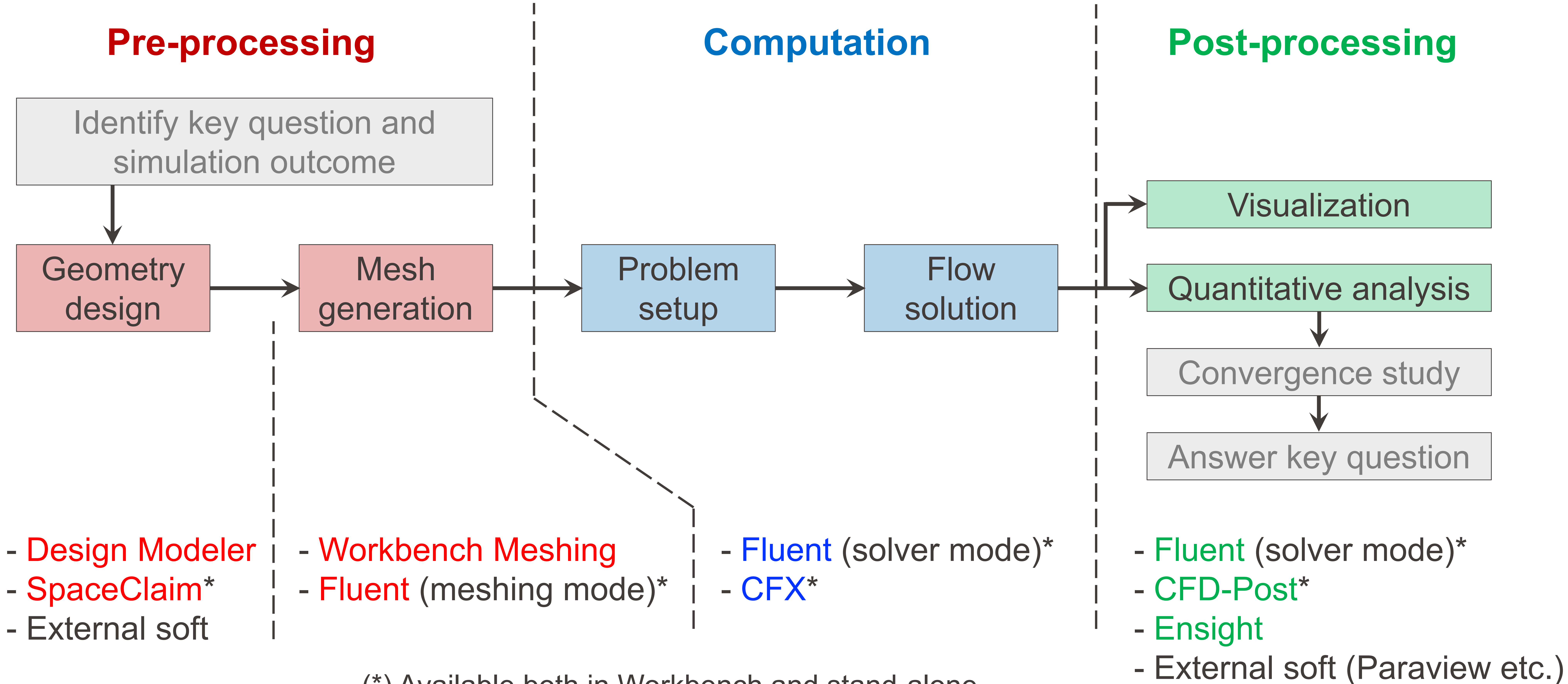
- Workbench Meshing
- Fluent (meshing mode)*

- Fluent (solver mode)*: more general
- CFX*: specialized for turbomachinery

(*) Available both in Workbench and stand-alone

# Numerical simulation workflow

- Some of the software available in Ansys:



**Pre-processing**

**Computation**

**Post-processing**

Identify key question and simulation outcome

Geometry design

Mesh generation

Problem setup

Flow solution

Visualization

Quantitative analysis

Convergence study

Answer key question

- Design Modeler
- SpaceClaim*
- External soft

- Workbench Meshing
- Fluent (meshing mode)*

- Fluent (solver mode)*
- CFX*

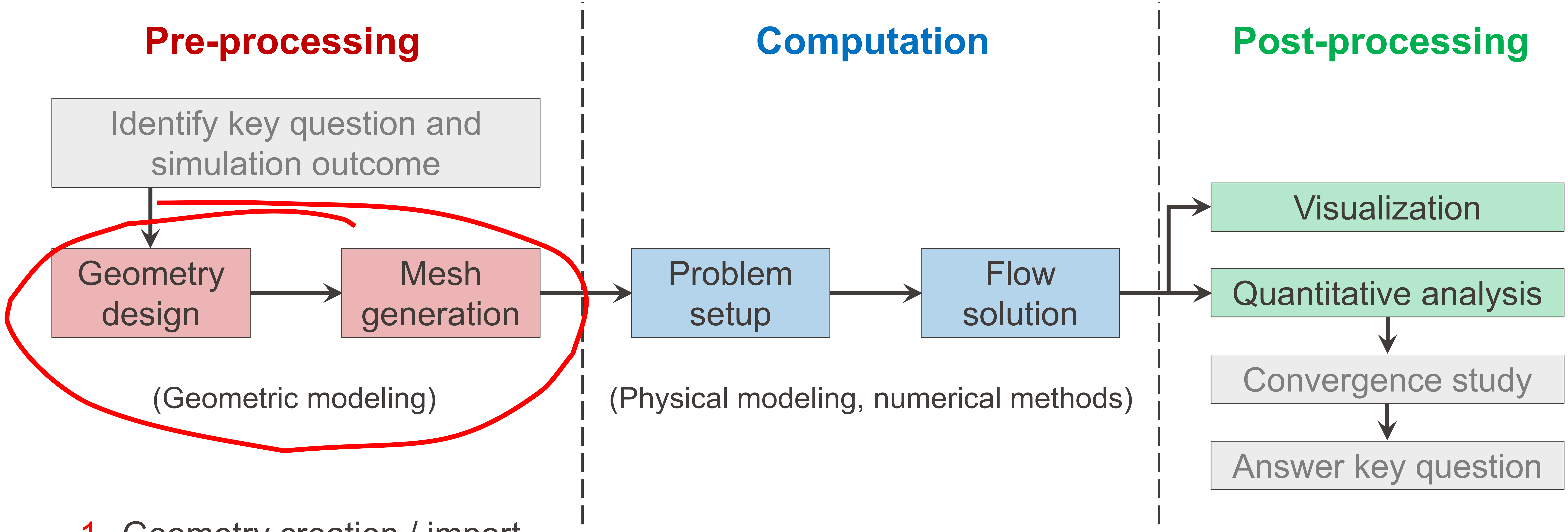
- Fluent (solver mode)*
- CFD-Post*
- Ensight
- External soft (Paraview etc.)

(*) Available both in Workbench and stand-alone

Numerical Flow Simulation

# Numerical simulation workflow

- This week:

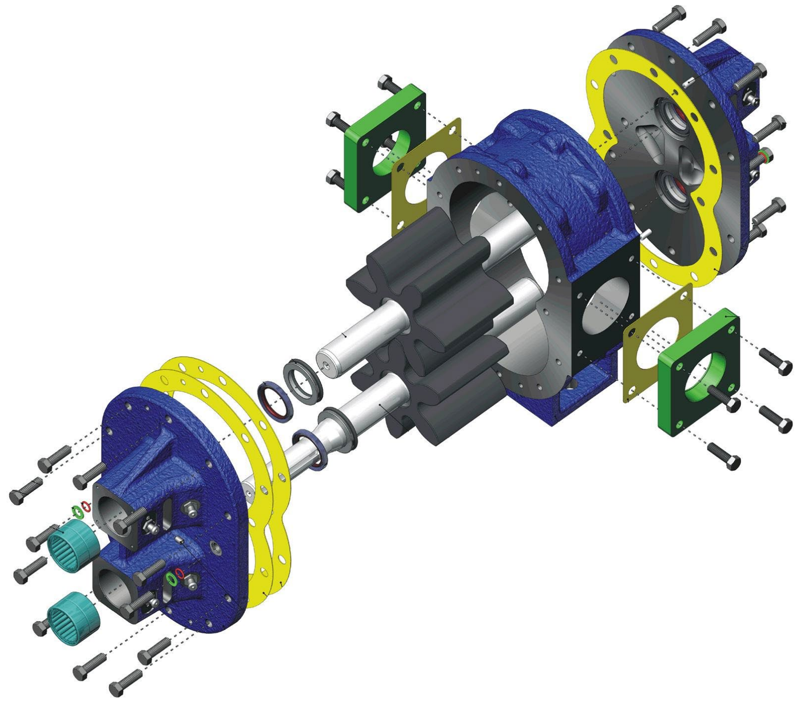


Numerical Flow Simulation

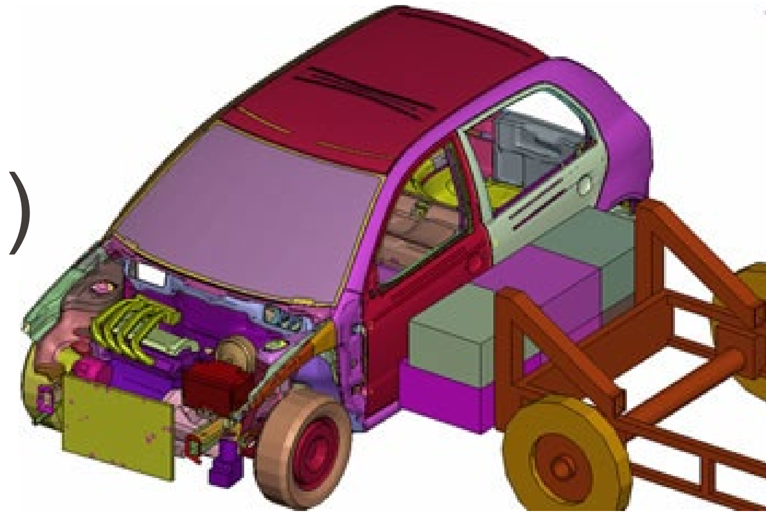
1. Geometry creation / import
2. Geometry clean-up (if needed)
3. Mesh creation / import
4. Mesh quality control + improvement

# Geometry design

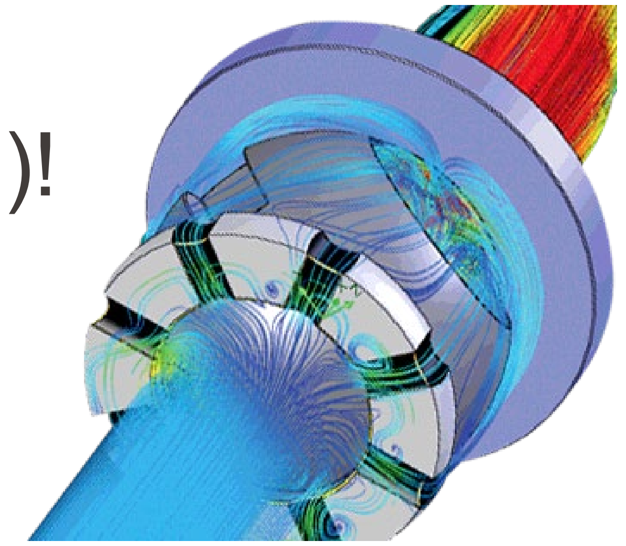
- Different processes for different needs:
  - Computer-Aided Manufacturing (CAM)
    - Interest: **manufacturing**, **interaction** between objects
    - Conception: generally complex objects / parts / assemblies



  - “Finite Element” Analysis (FEA)
    - Interest: simulation domain for **structural analysis** (deformation, stress…)
    - Conception: boundary surfaces / volumes (not necessarily closed)



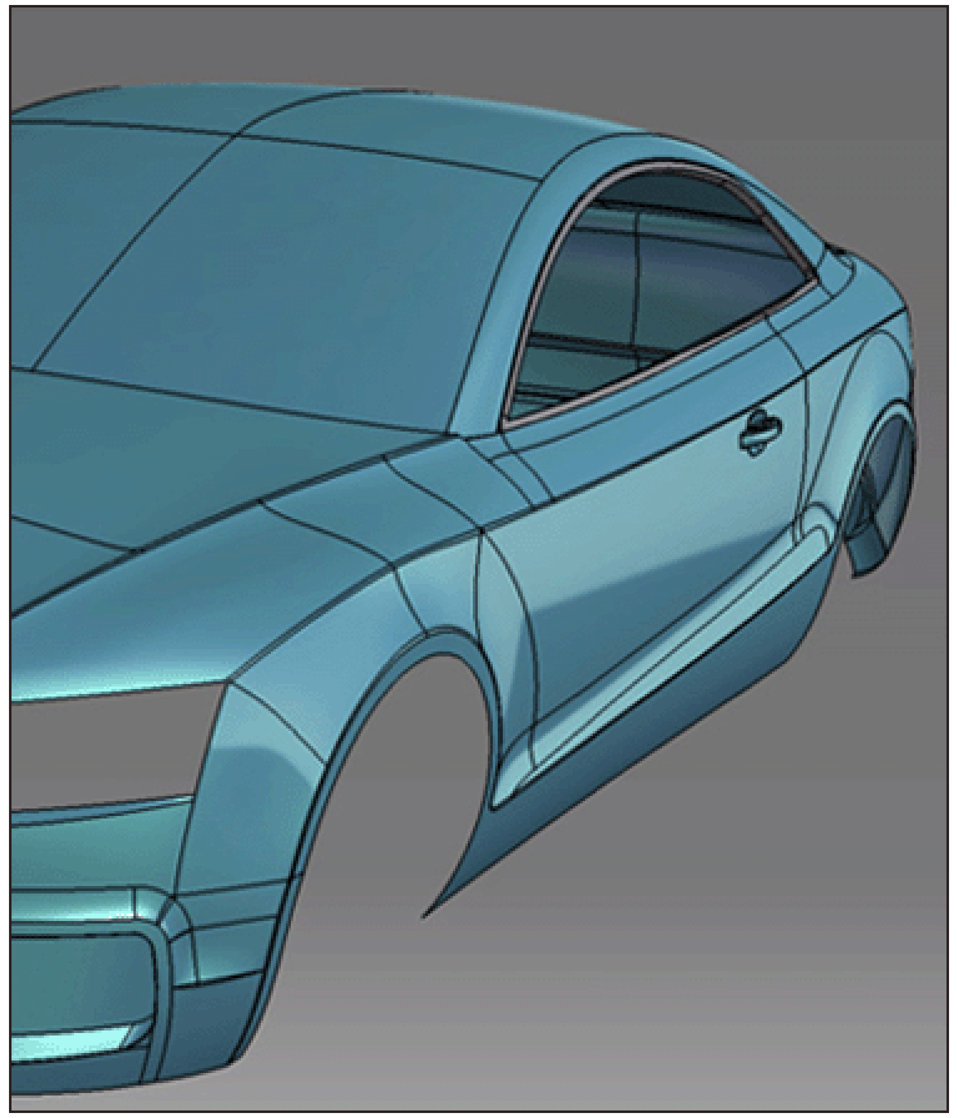
  - Computational Fluid Dynamics (CFD)
    - Interest: simulation domain for **flow analysis**
    - Conception: volume occupied by the fluid. Must be closed (“watertight”)!



  - Others: computer graphic design (for publishing, advertising or education), etc.

# Geometry import/export

- Various ways to exchange geometry:
  - “Geometric modeling kernels”:
    - Parasolid [.x_t, .x_b] (used in Abaqus, Workbench, SolidWorks, Solid Edge, ANSYS, Comsol…)
    - ACIS [.sat] (used in Gambit / Trelis, AutoCAD, Cadkey, TurboCad…)
    - Convergence Geometric Modeler (used in CATIA)
    - …
  - Industry standard file formats:
    - STEP [.stp] ("Standard for the Exchange of Product model data”, ISO 10303)
    - IGES [.iges, .igs] (“Initial Graphics Exchange Specification”)
    - …



# Geometry clean-up

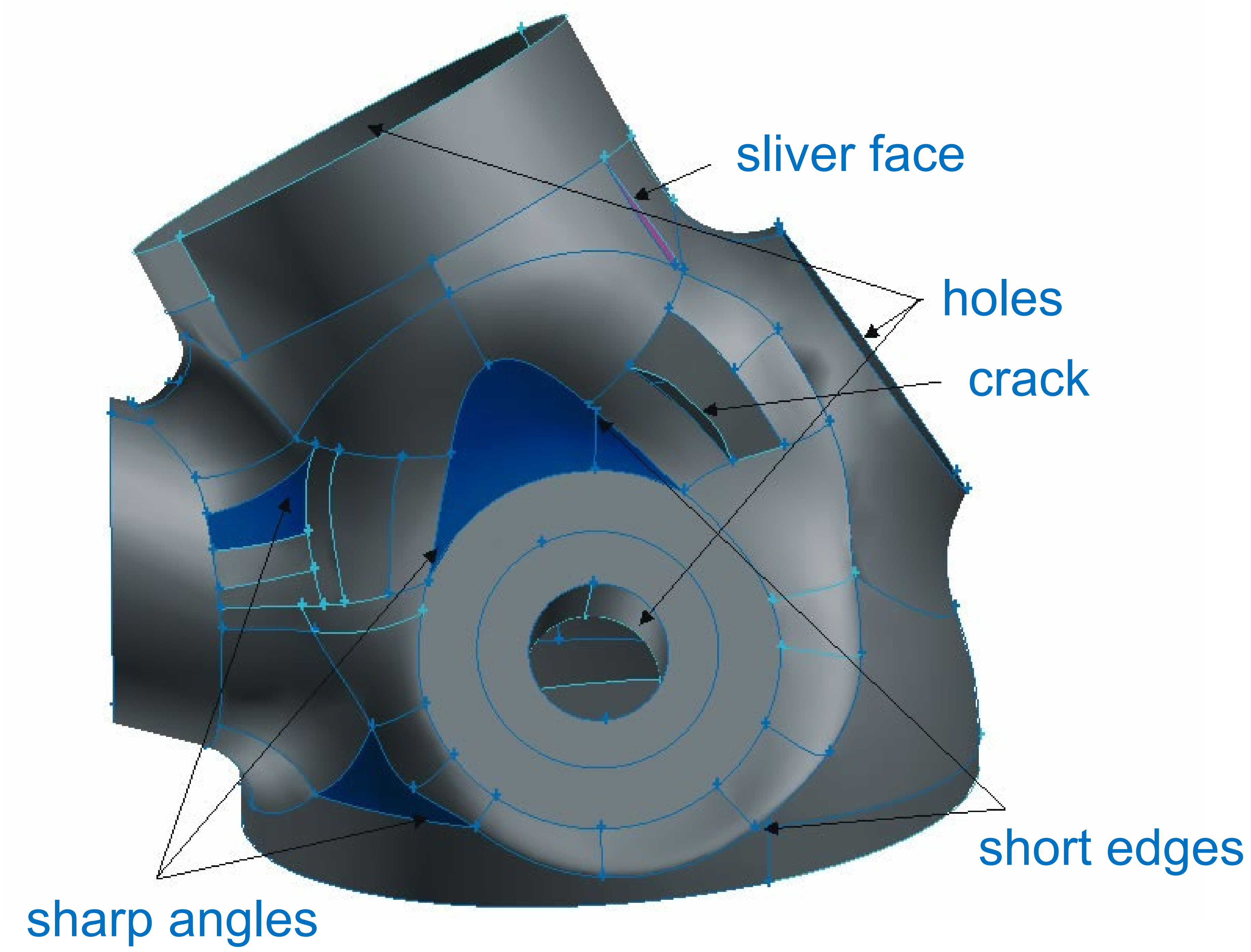
- Many problems may arise at import, especially if original CAD geometry was designed for a different purpose:
  - Volumes not closed (not watertight).
  - Small features that are difficult to mesh but do not influence significantly the flow.
  - Small surfaces that must be merged to simplify meshing (because each independent surface in the geometry will be meshed independently).
  - Translation between different CAD systems may result in corrupt / incomplete geometry / topology.

- Clean-up:
  - Identify problems: holes, sliver faces / cracks (too small to be meshed), sharp angles / short edges (give poor mesh), etc.
  - Automated “healing”: geometry simplification, stitching (connect edges or faces), geometry building.

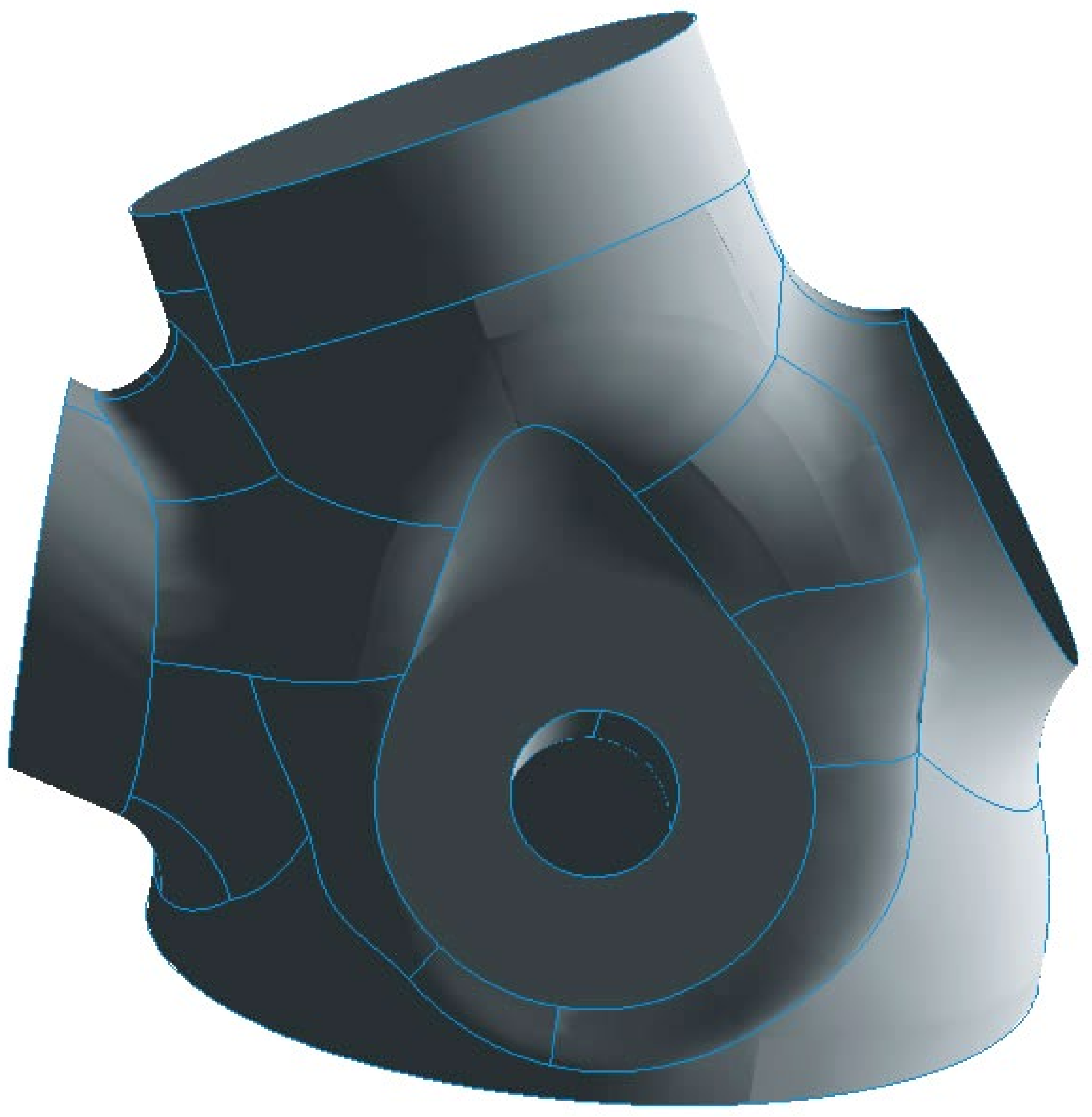
# Geometry clean-up

- Example

Imported geometry



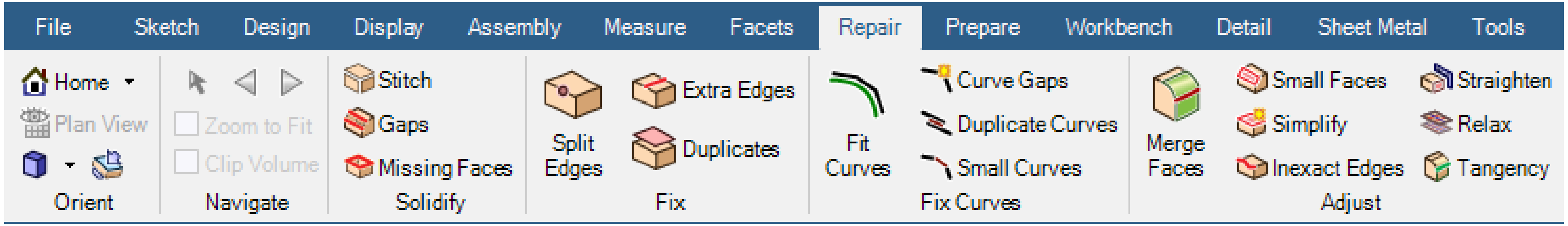
Geometry after “healing”



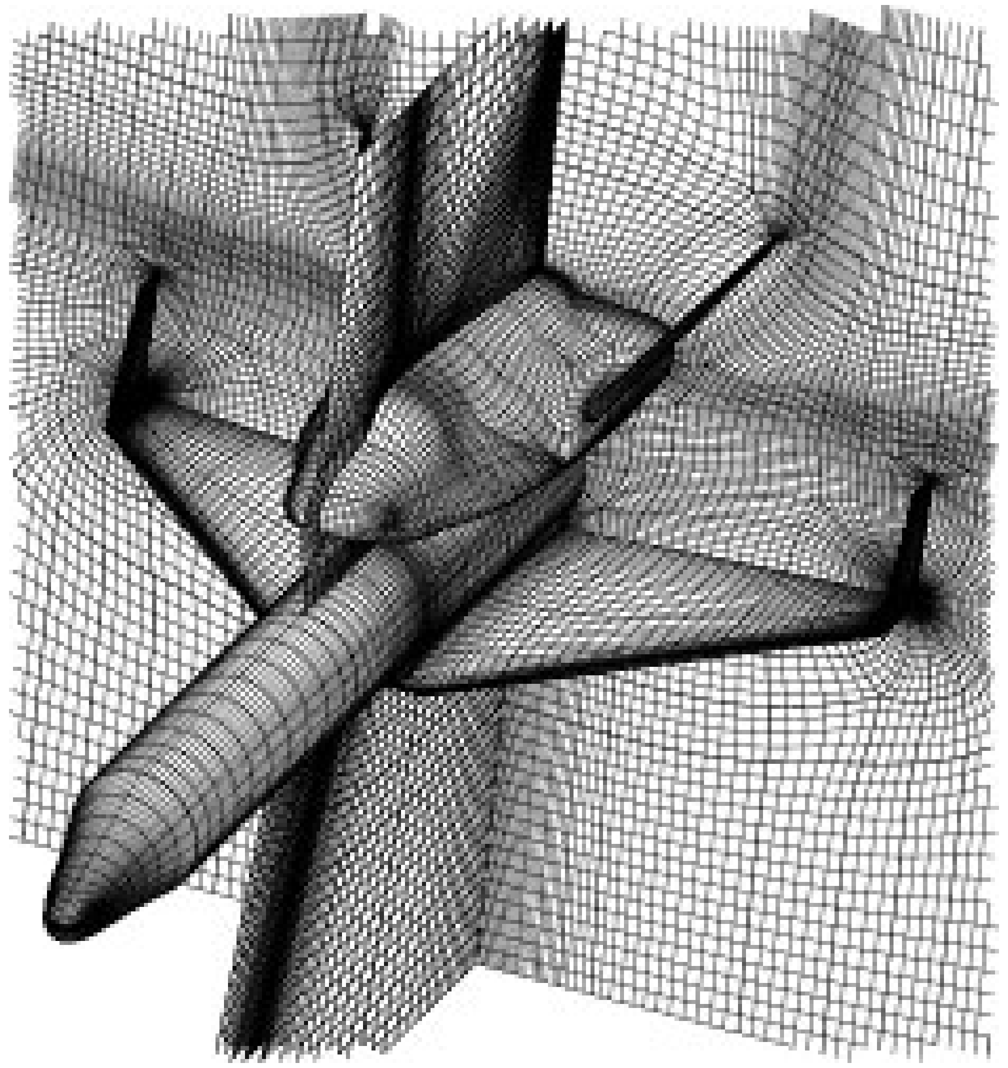
Numerical Flow Simulation

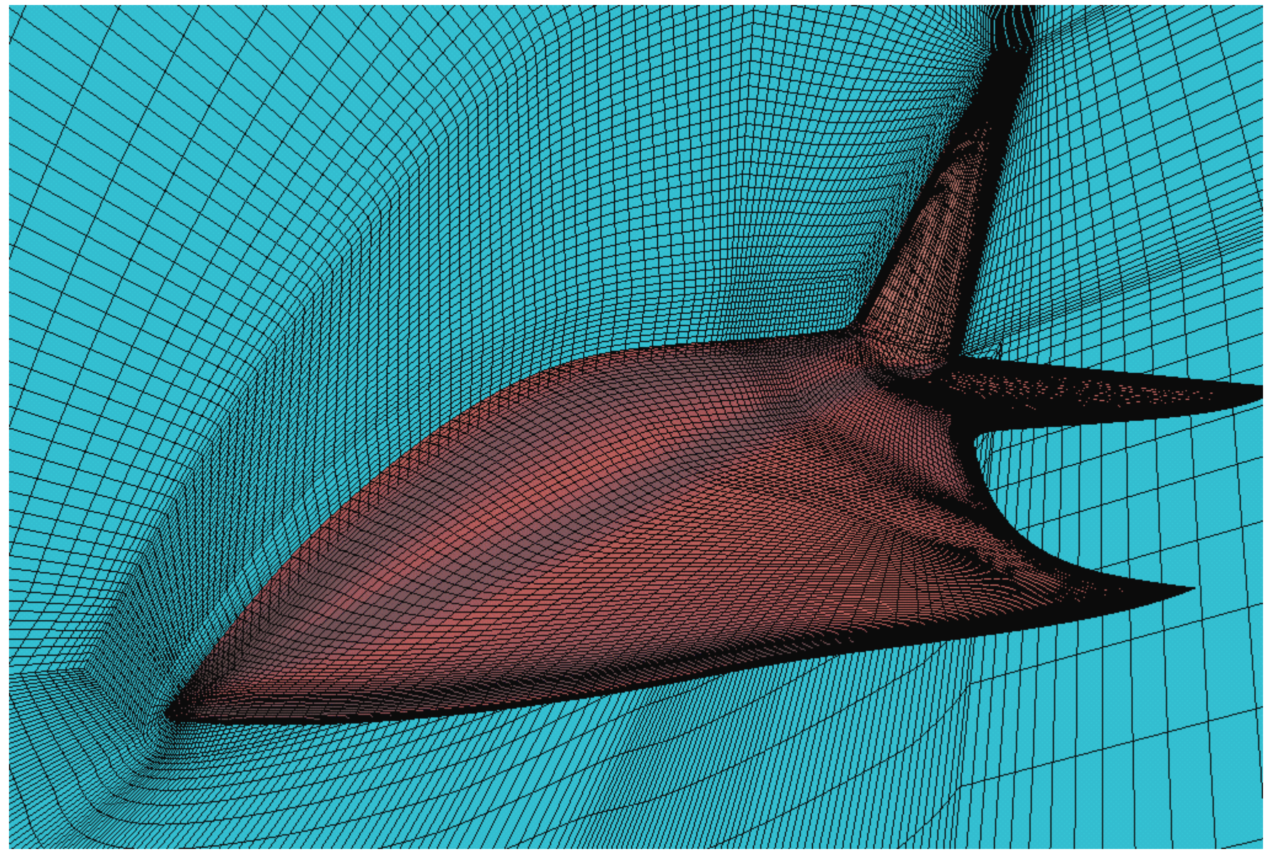
# Geometry clean-up

- In SpaceClaim: Repair tab

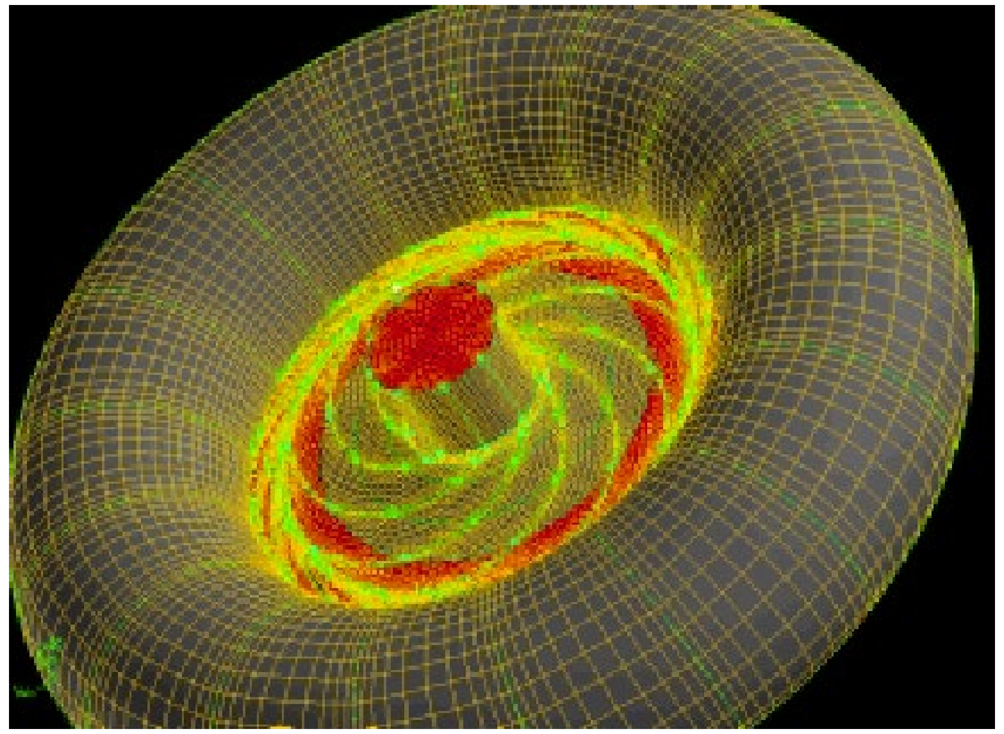


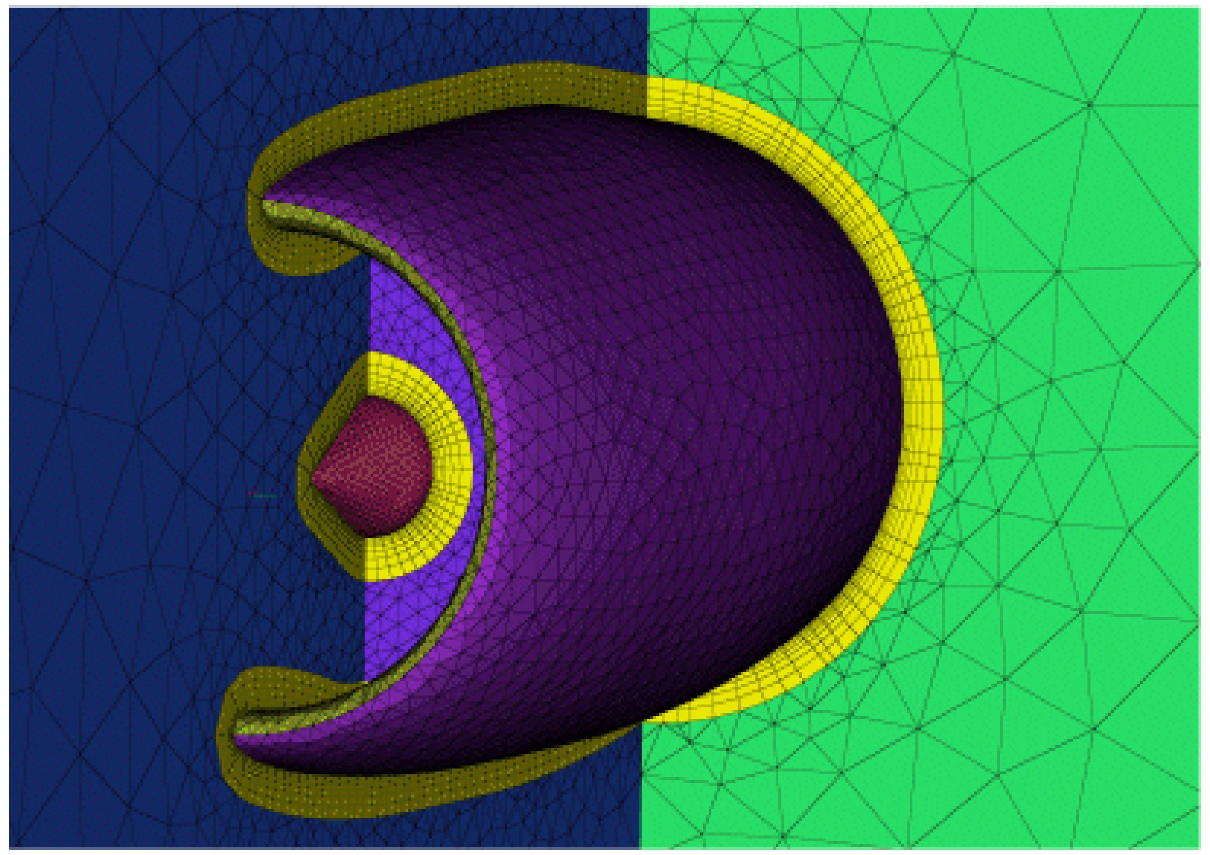
# Wide variety of meshes

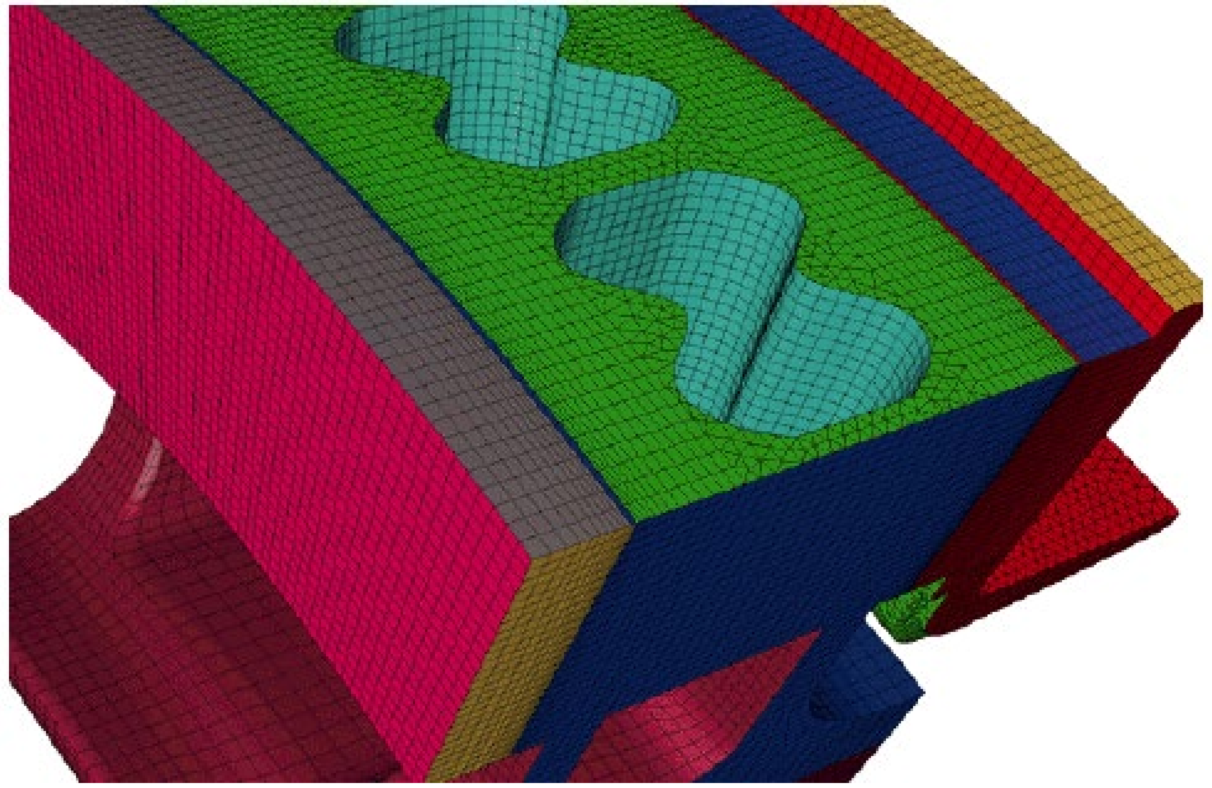




Numerical Flow Simulation



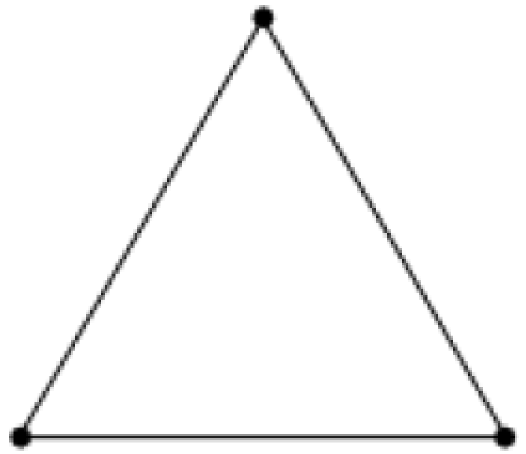




# Mesh elements

- Common types of mesh elements (= cells = control volumes):

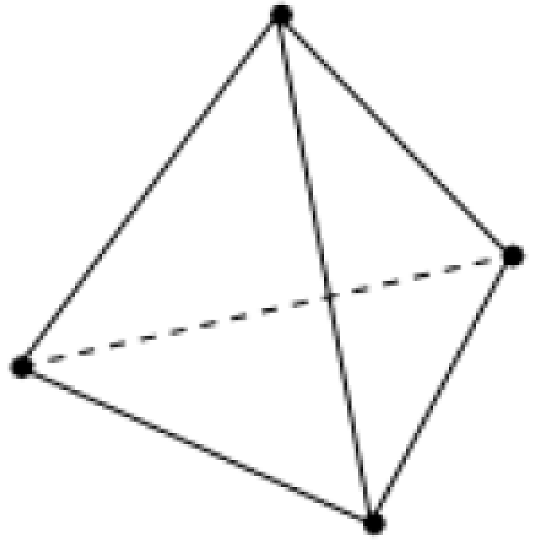
**2D elements**



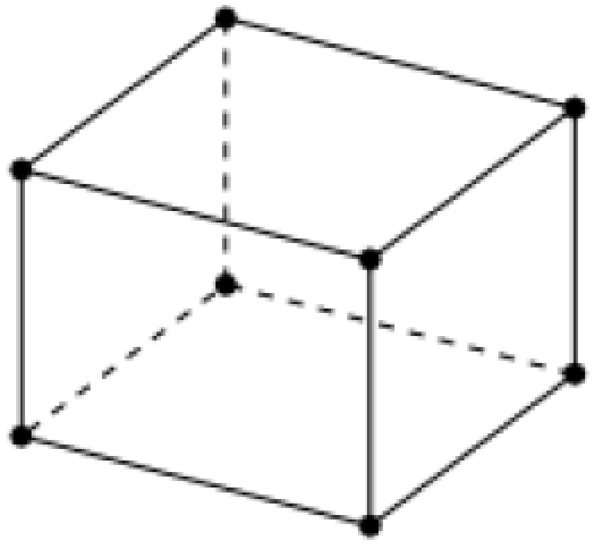
**Triangle**
3 nodes/edges
(not necessarily equilateral…)

**Quadrilateral**
4 nodes/edges
(not necessarily orthogonal…)

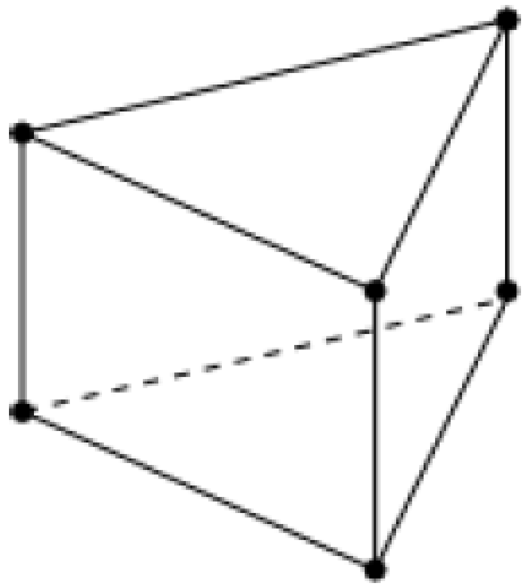
**3D elements**



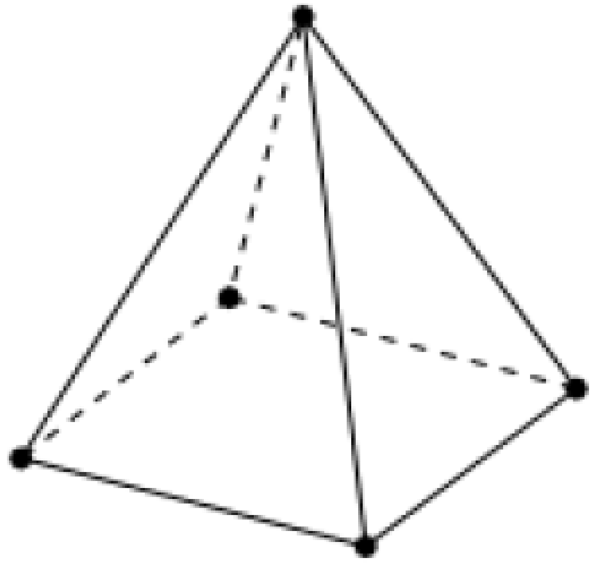
**Tetrahedron**
4 nodes, 4 faces



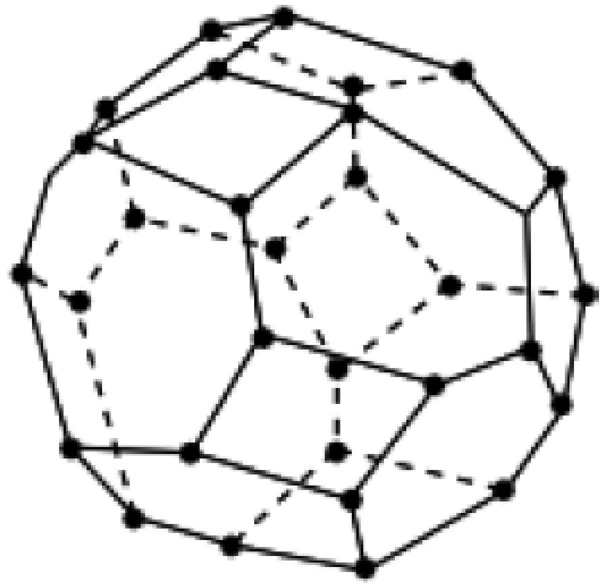
**Hexahedron**
8 nodes, 6 faces



**Wedge/Prism**
6 nodes, 5 faces



**Pyramid**
5 nodes, 5 faces

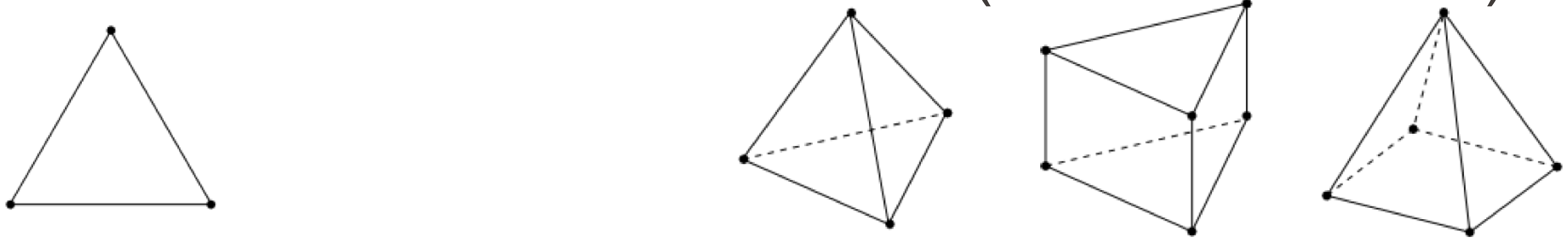


**Polyhedron**

# Mesh elements

- Each type of element has different properties regarding the numerical approximation of gradients and fluxes.
- In general, quads / hex are more accurate (under some conditions) and can help reduce the number of elements (larger aspect ratio).

- Triangles, tetras, prisms etc. make it easier to mesh complex geometries and can help reduce the number of elements (local refinement).

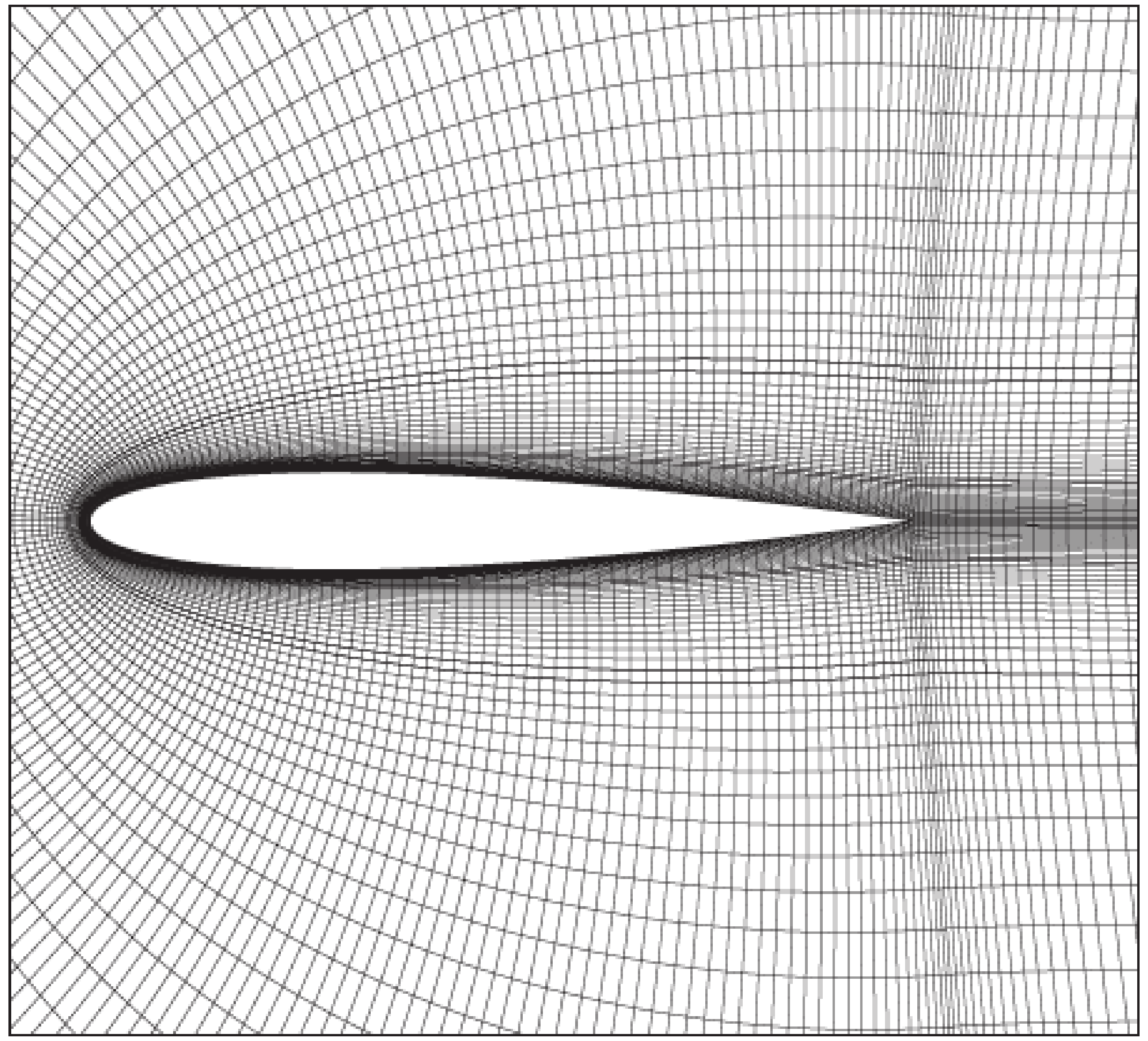


- In the end, choice up to the user, as long as mesh size and quality acceptable.

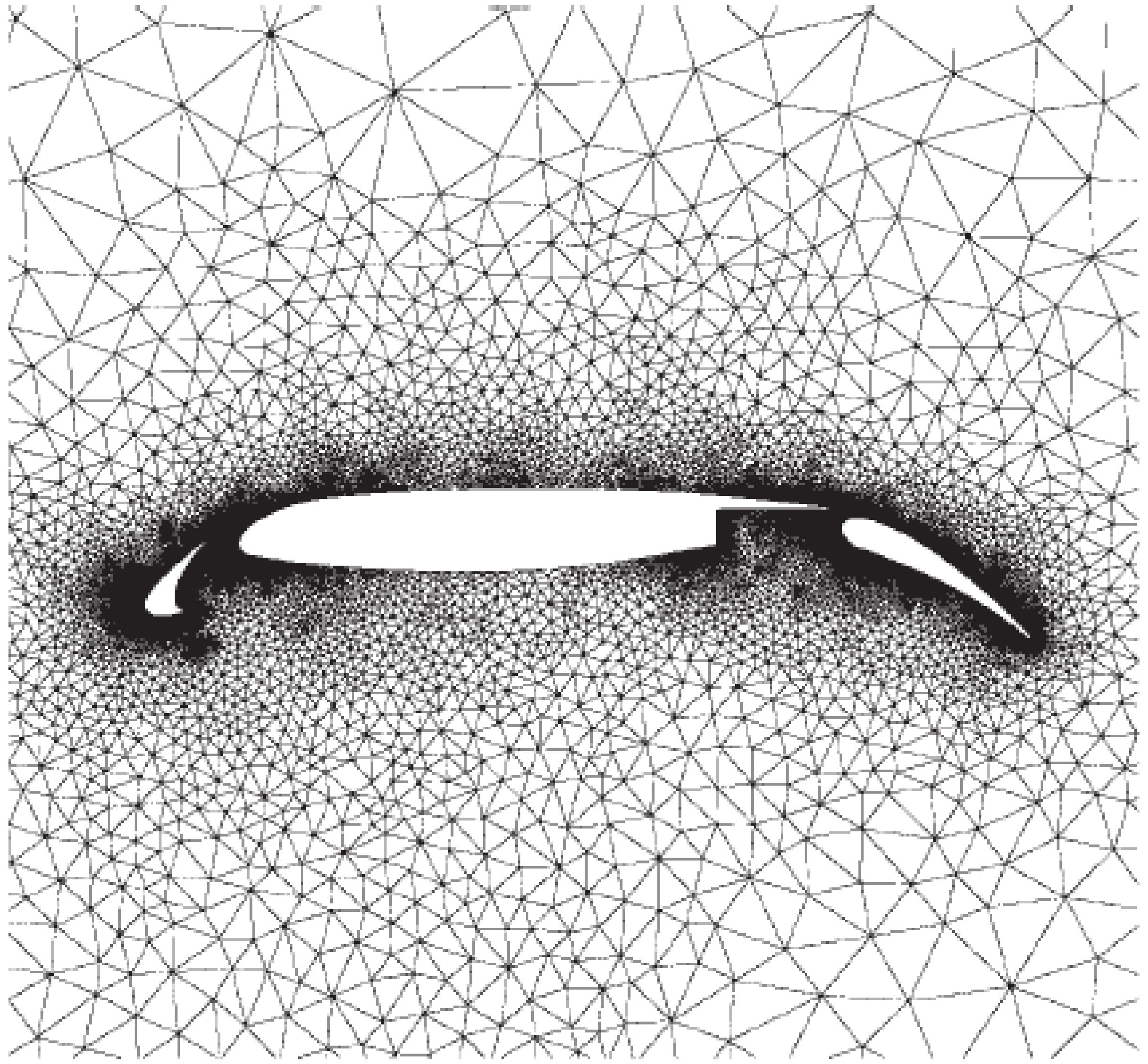
# Mesh types

- Two basic types:

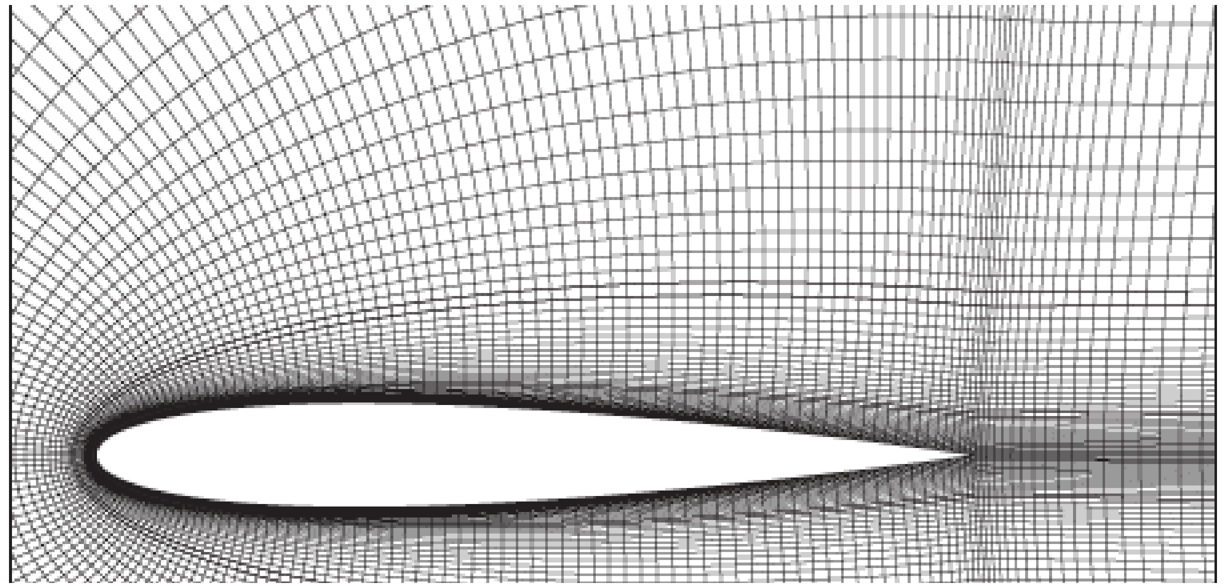
1. Structured

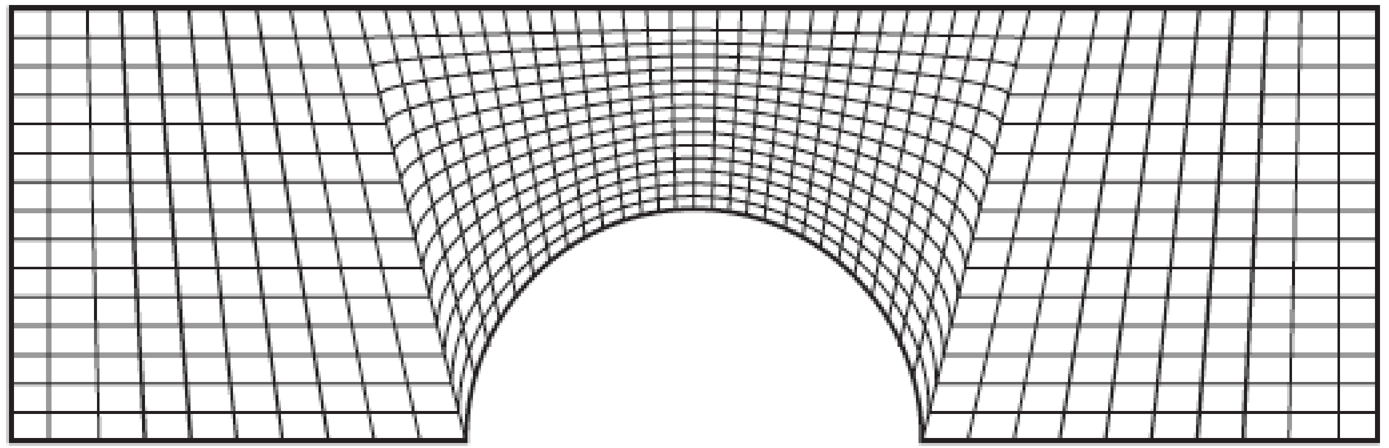


2. Unstructured



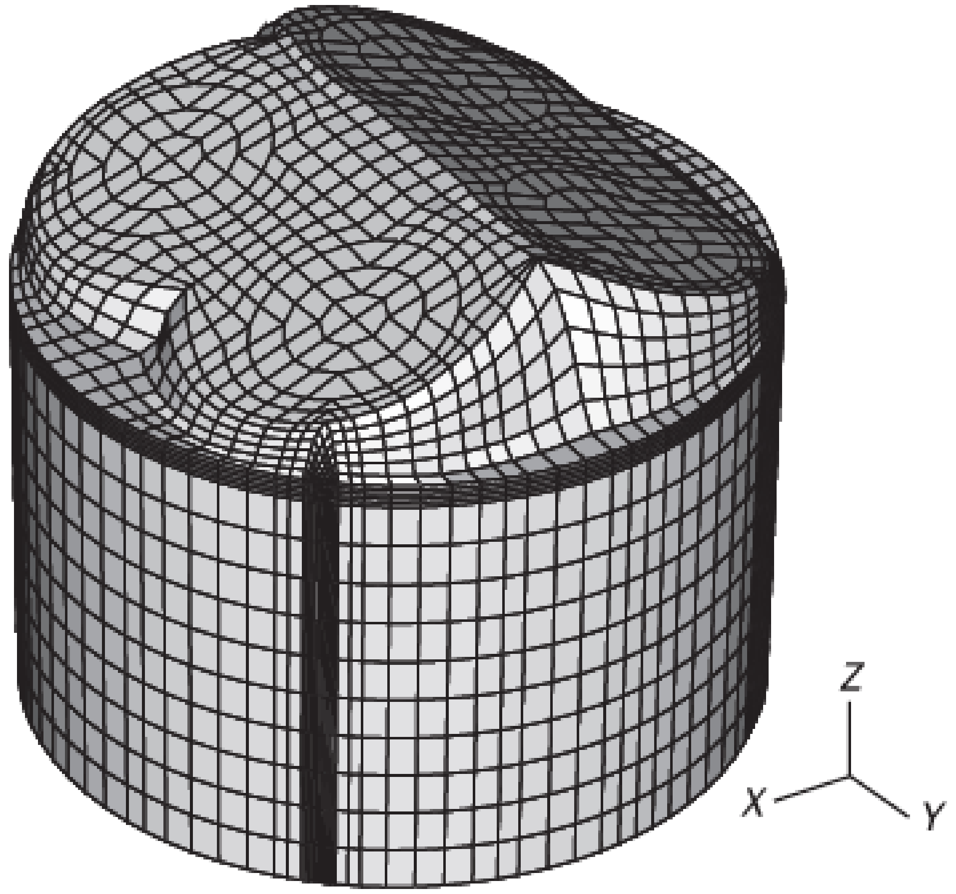
# Mesh types



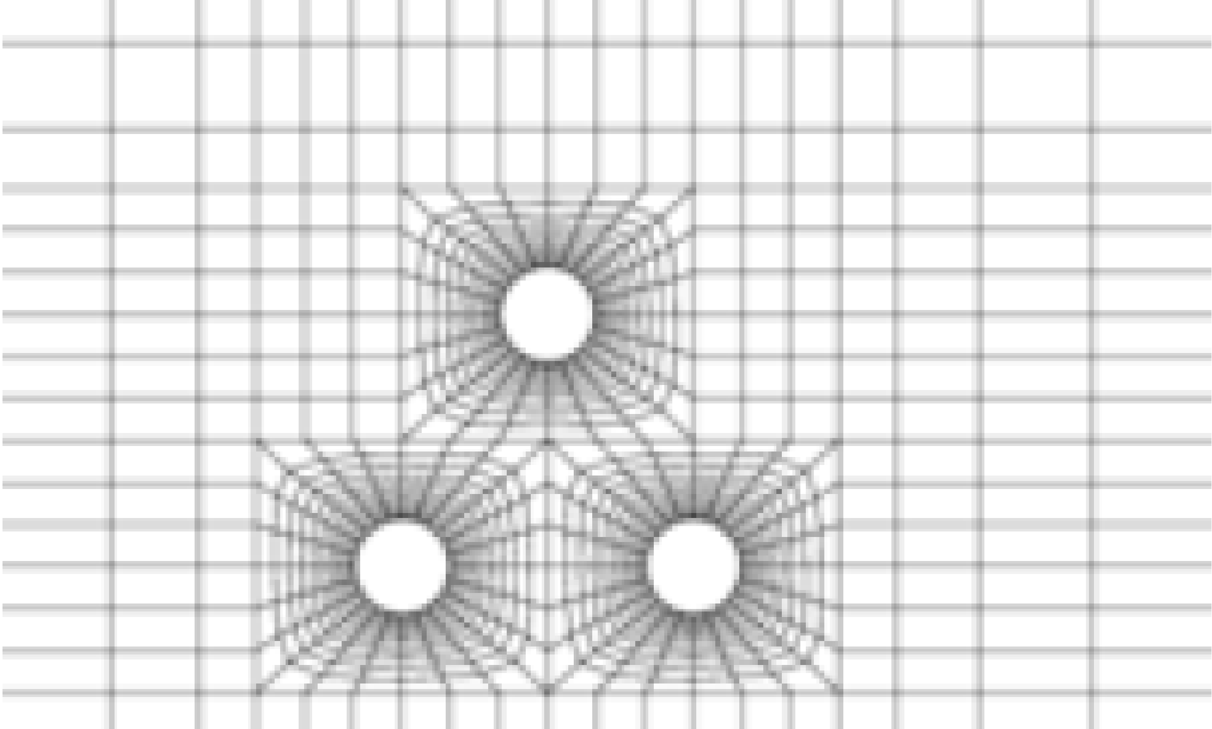


1. Structured mesh

   - Sets of lines where members of the same set don’t cross, and members of different sets cross only once → position of any CV uniquely defined by two or three indices (in 2D or 3D). Fixed number of neighbors.
   - Simple neighbor connectivity, structured matrix, efficient solution algorithms.
   - Generally quad / hex. (Orthogonal or not.)
   - Difficult/impossible to use with complex geometries.
   - Difficult to refine locally. (Inefficient use of resources.)

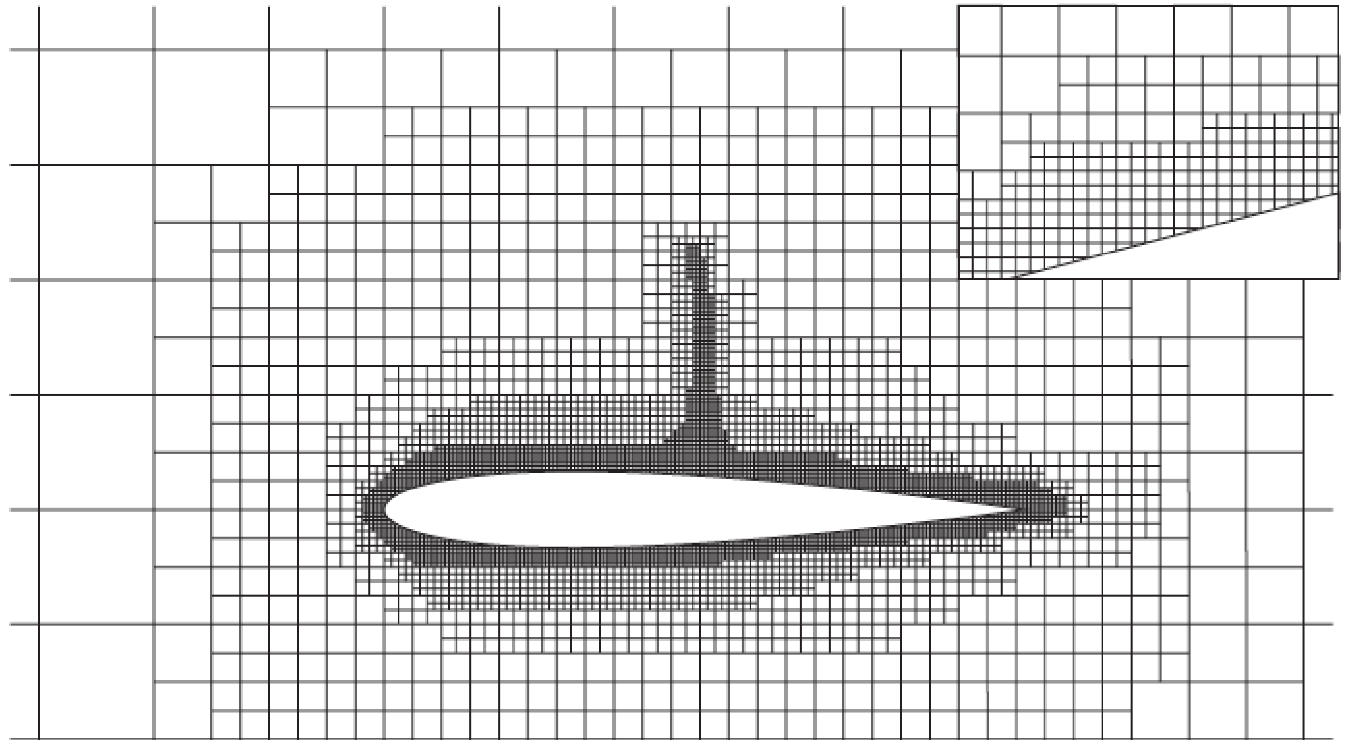


# Mesh types

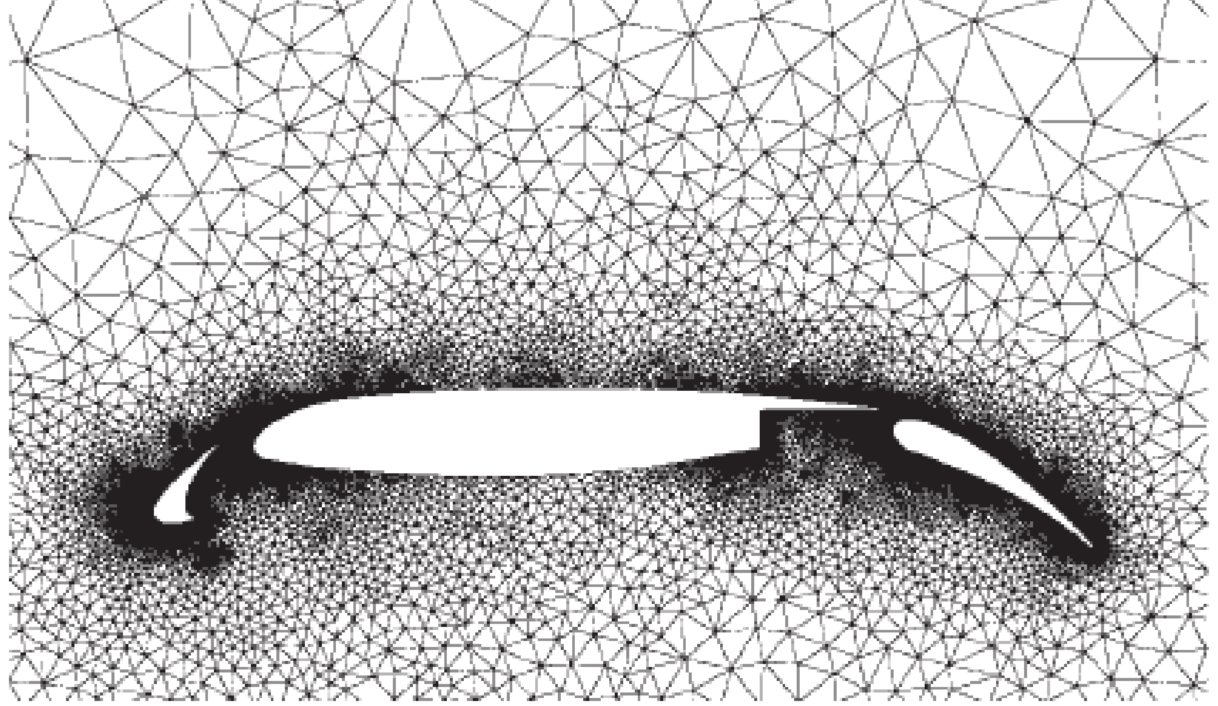


1. Block-structured mesh

   - Domain divided into subdomains (blocks), each with a structured grid.
   - More flexible, can mesh more complex geometries.
   - Easier local refinement (block-wise).
   - Special case: adaptive mesh. Recursive refinement.

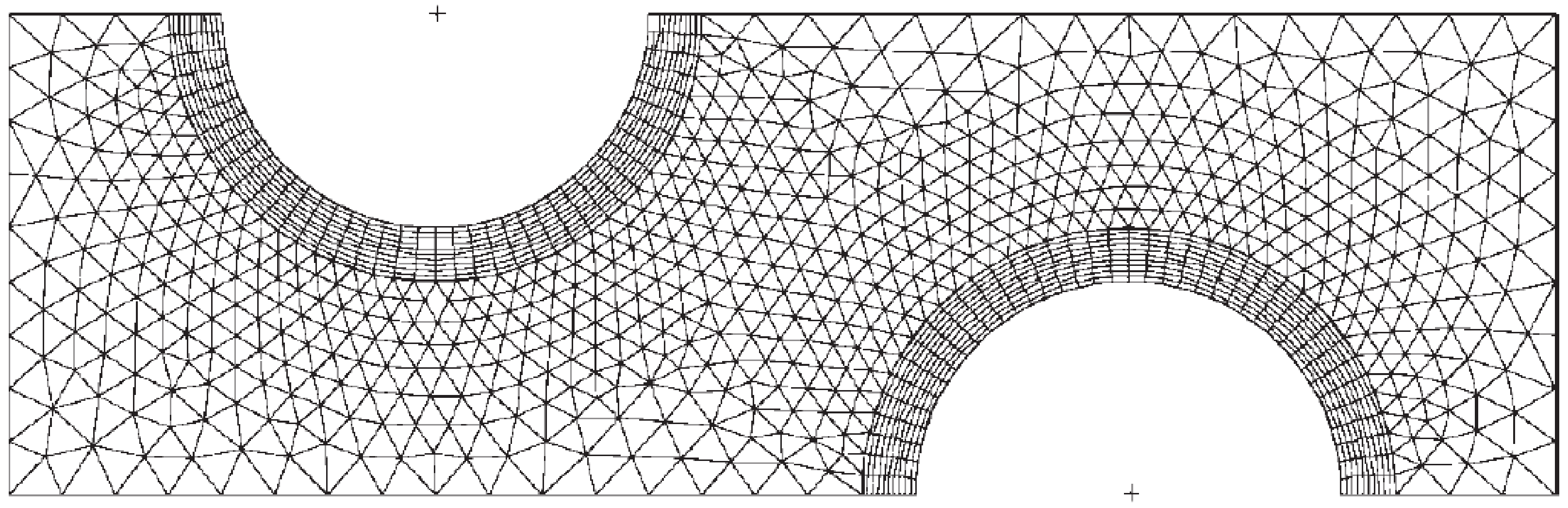


# Mesh types



## 2. Unstructured mesh

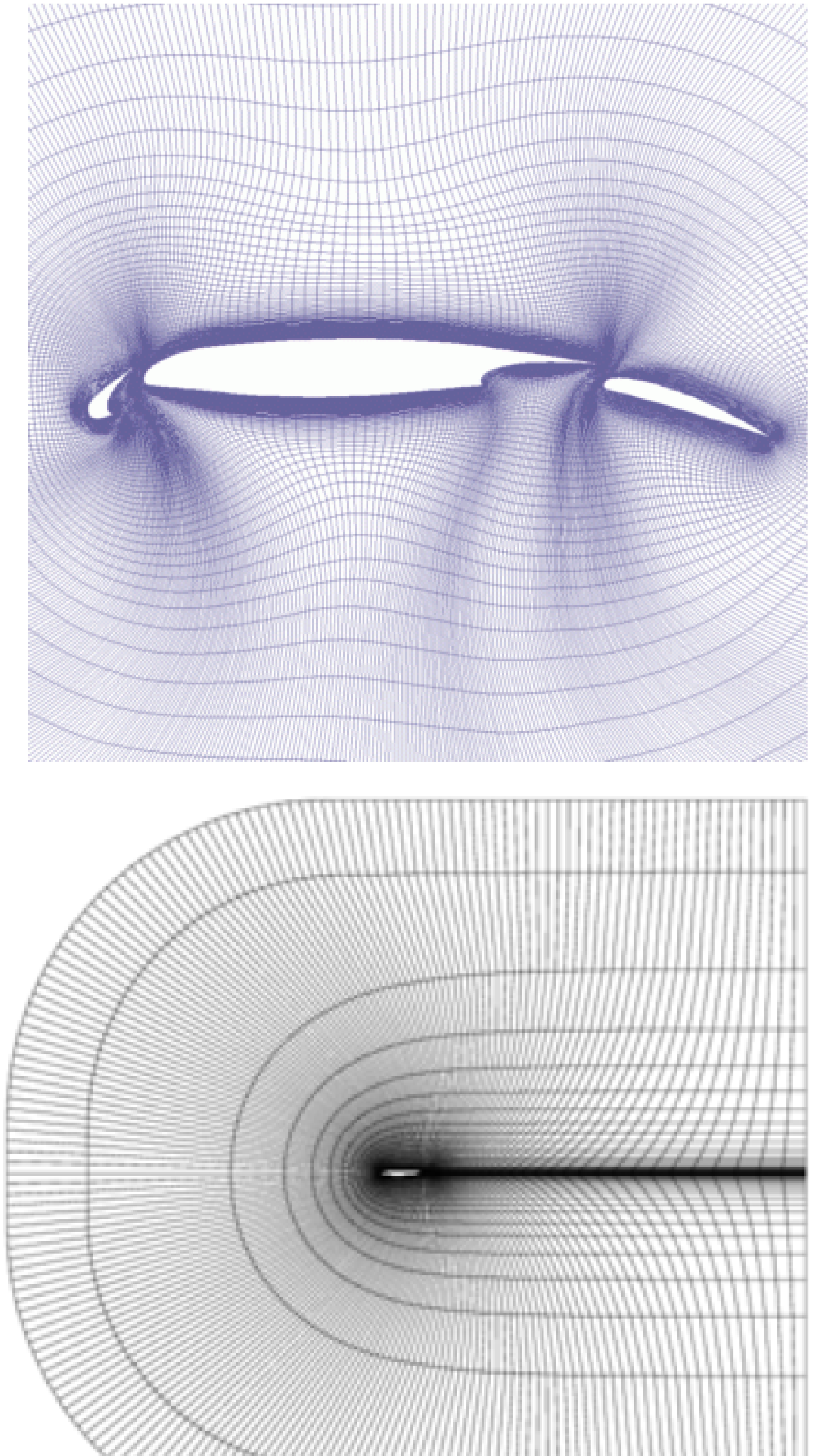
- No structure. Arbitrary element shape, connectivity and number of neighbors.
- Generally tri / tetra.
- Can also mix different shapes (hybrid mesh) to achieve better resolution and alignment where needed (with quad / hex) and still use resources efficiently. Often used for boundary layers.



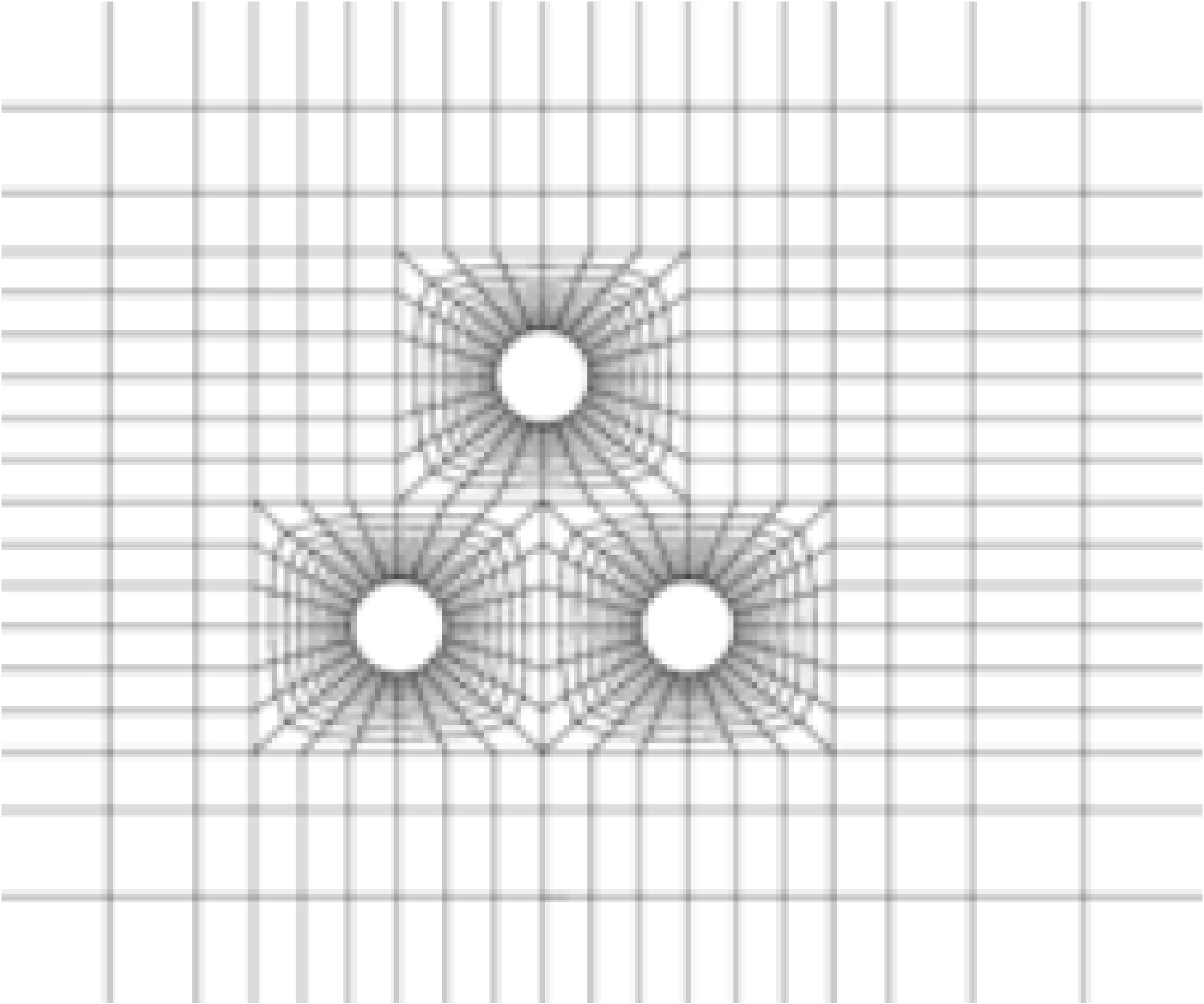
- Can mesh very complex geometries.
- Very good control of local refinement (including solution-based semi-automatic refinement).

# Some 2D examples

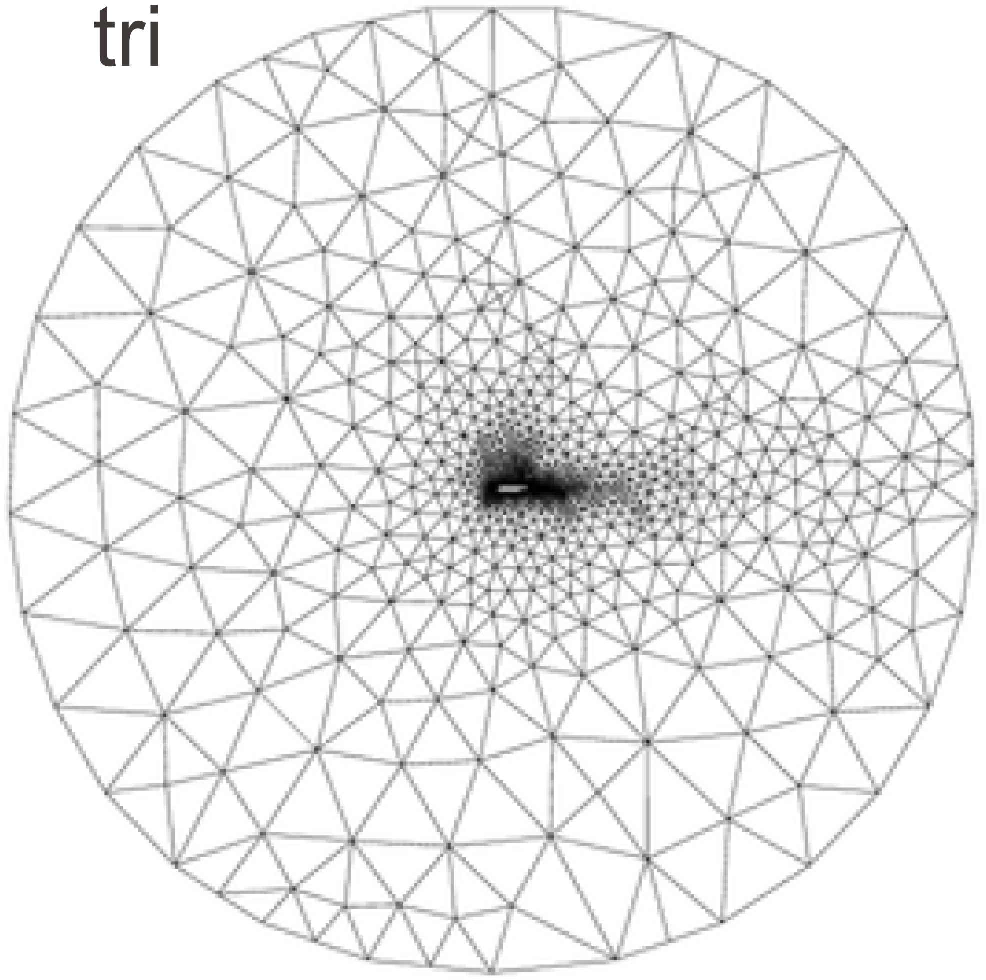
Structured, quad



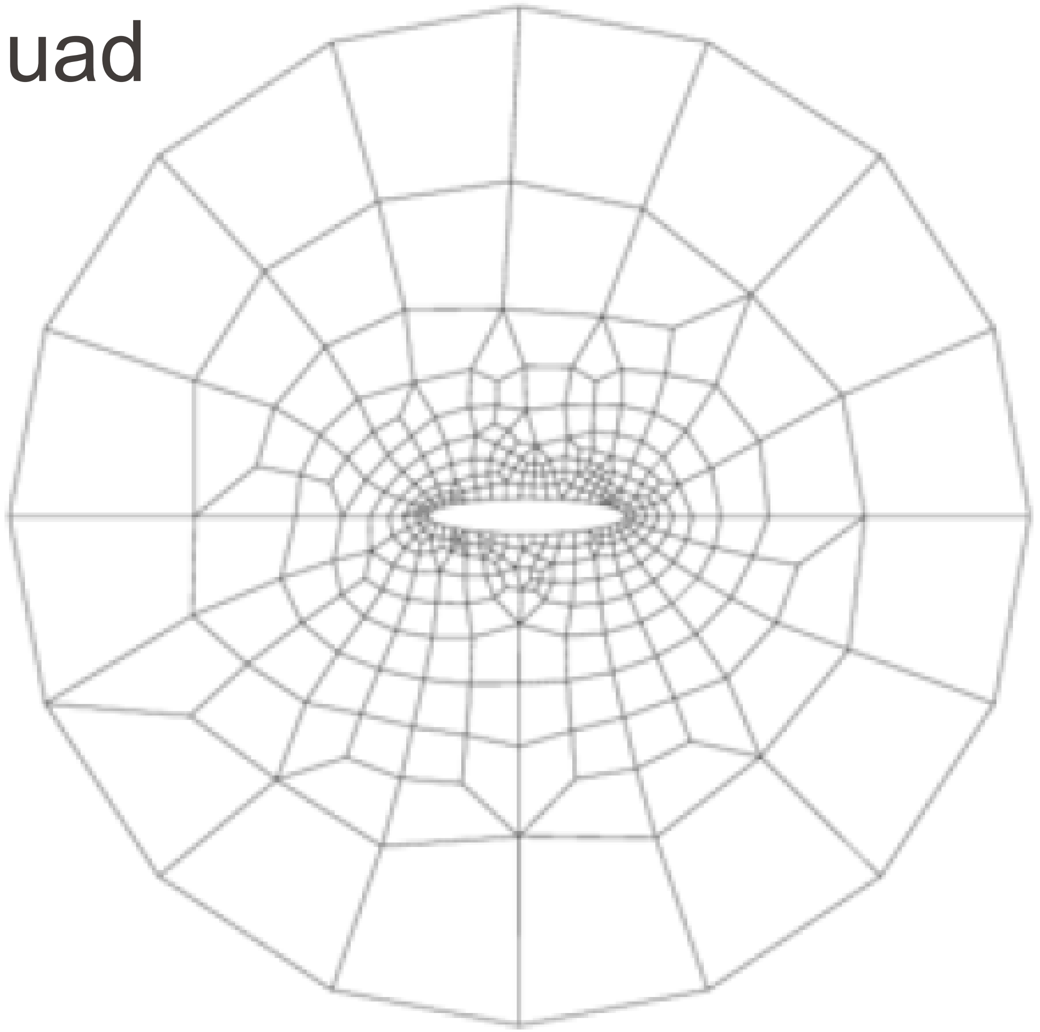
Block-structured, quad



Unstructured, tri



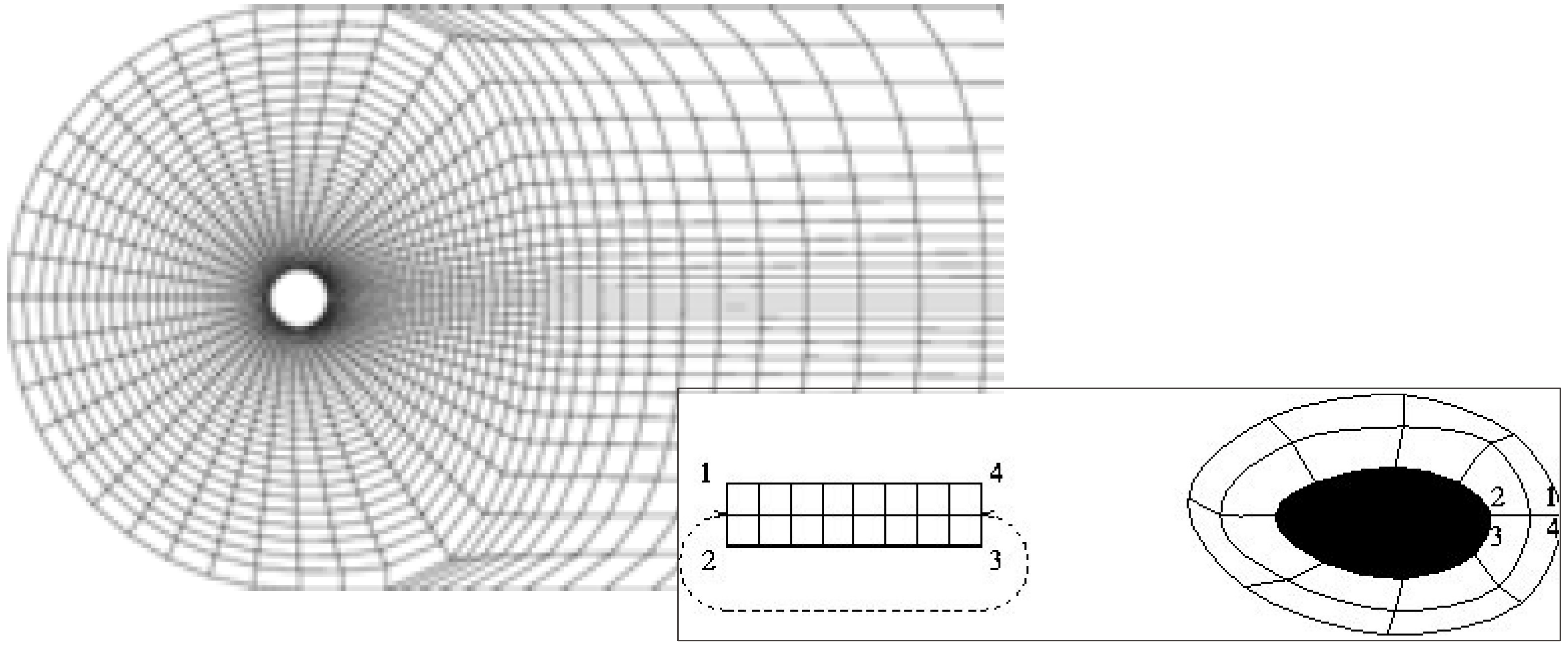
Unstructured, quad



Numerical Flow Simulation

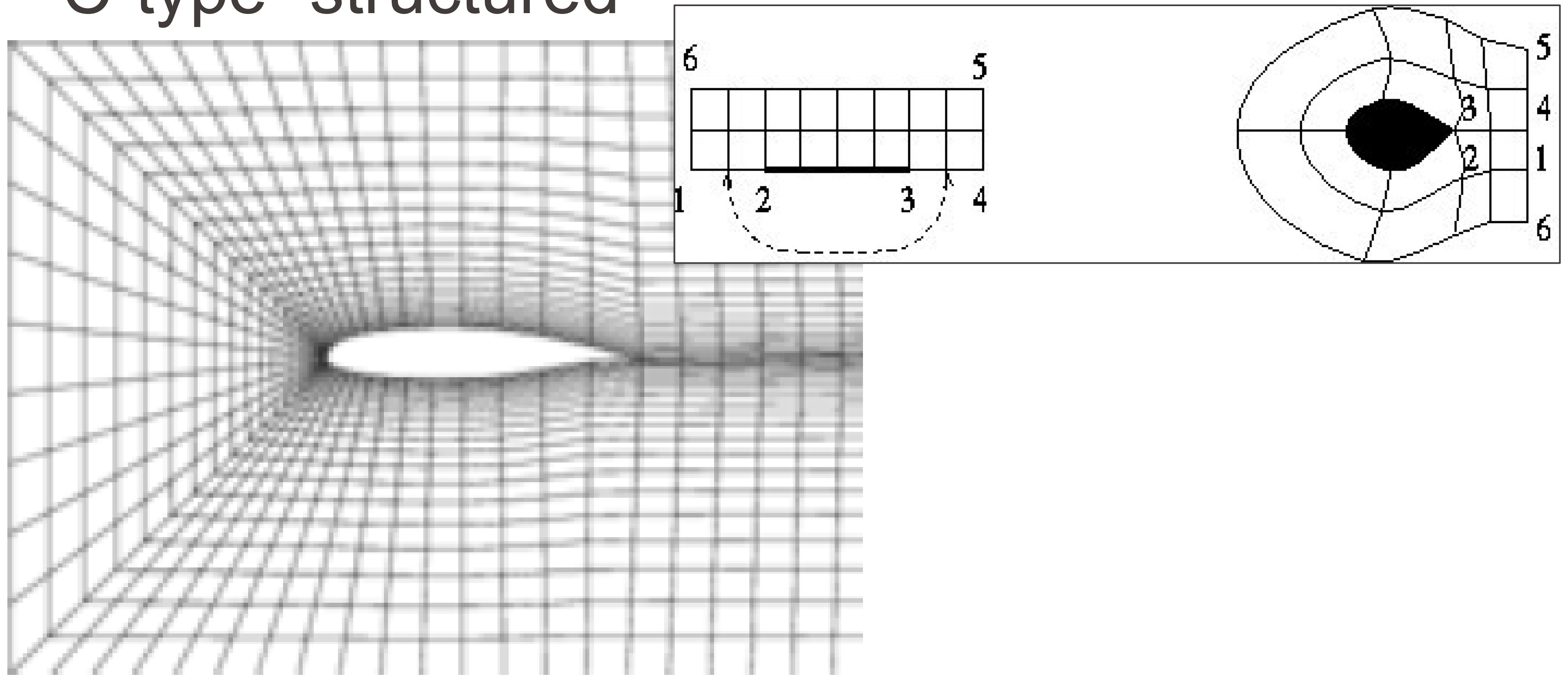
# Some 2D examples

"O-type" structured

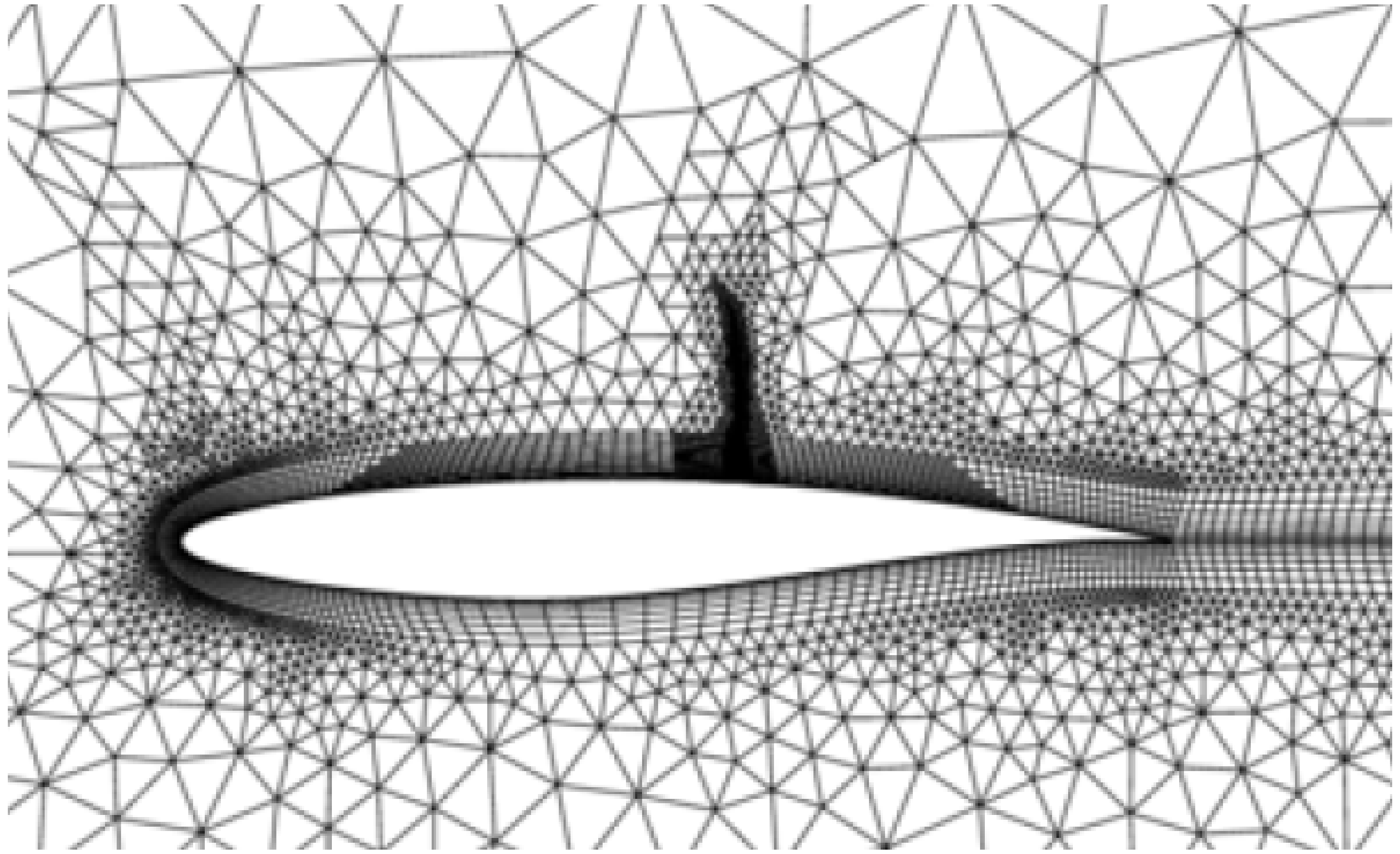


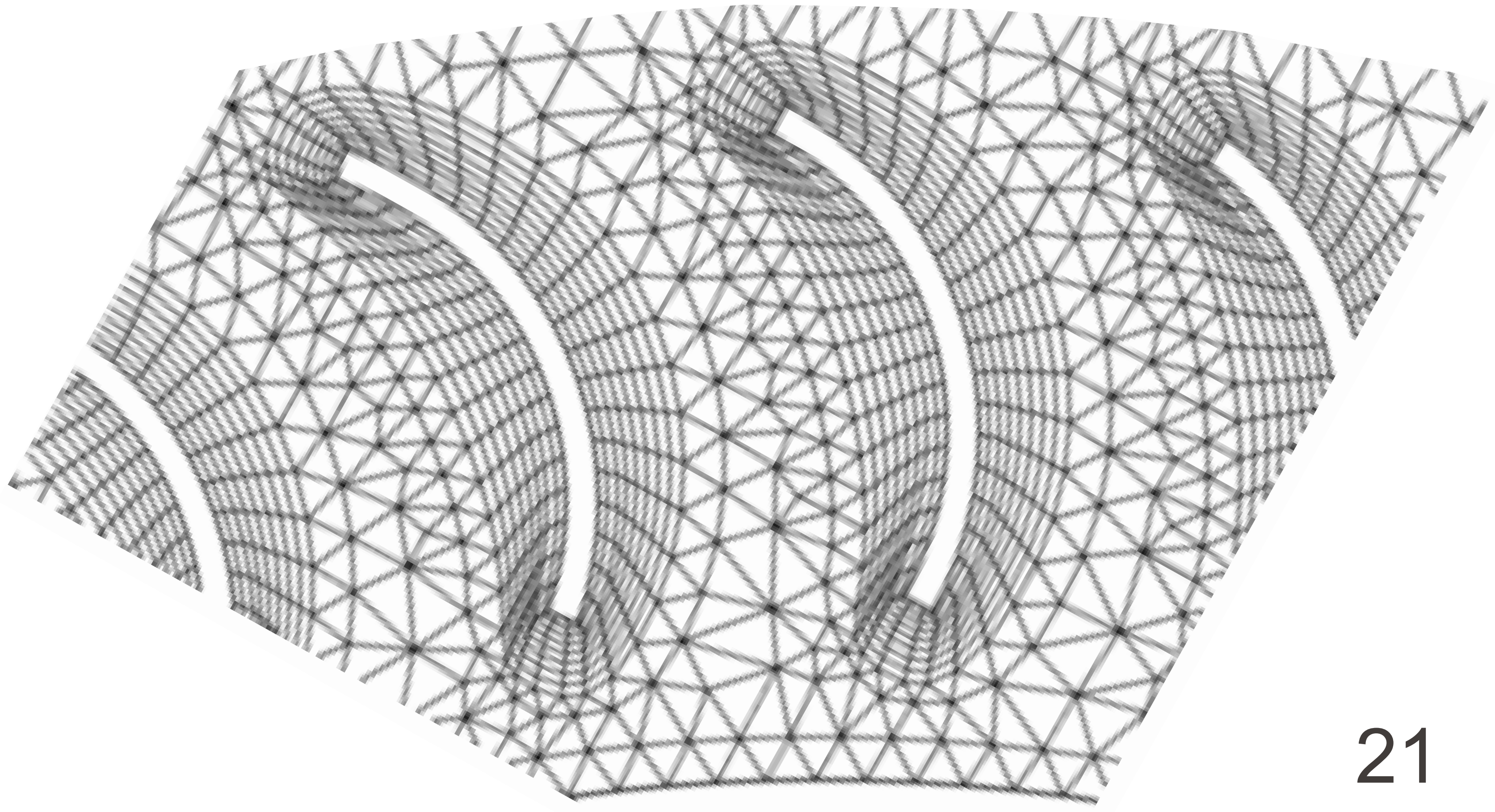
Different topologies
(different mappings
from Cartesian mesh)

"C-type" structured



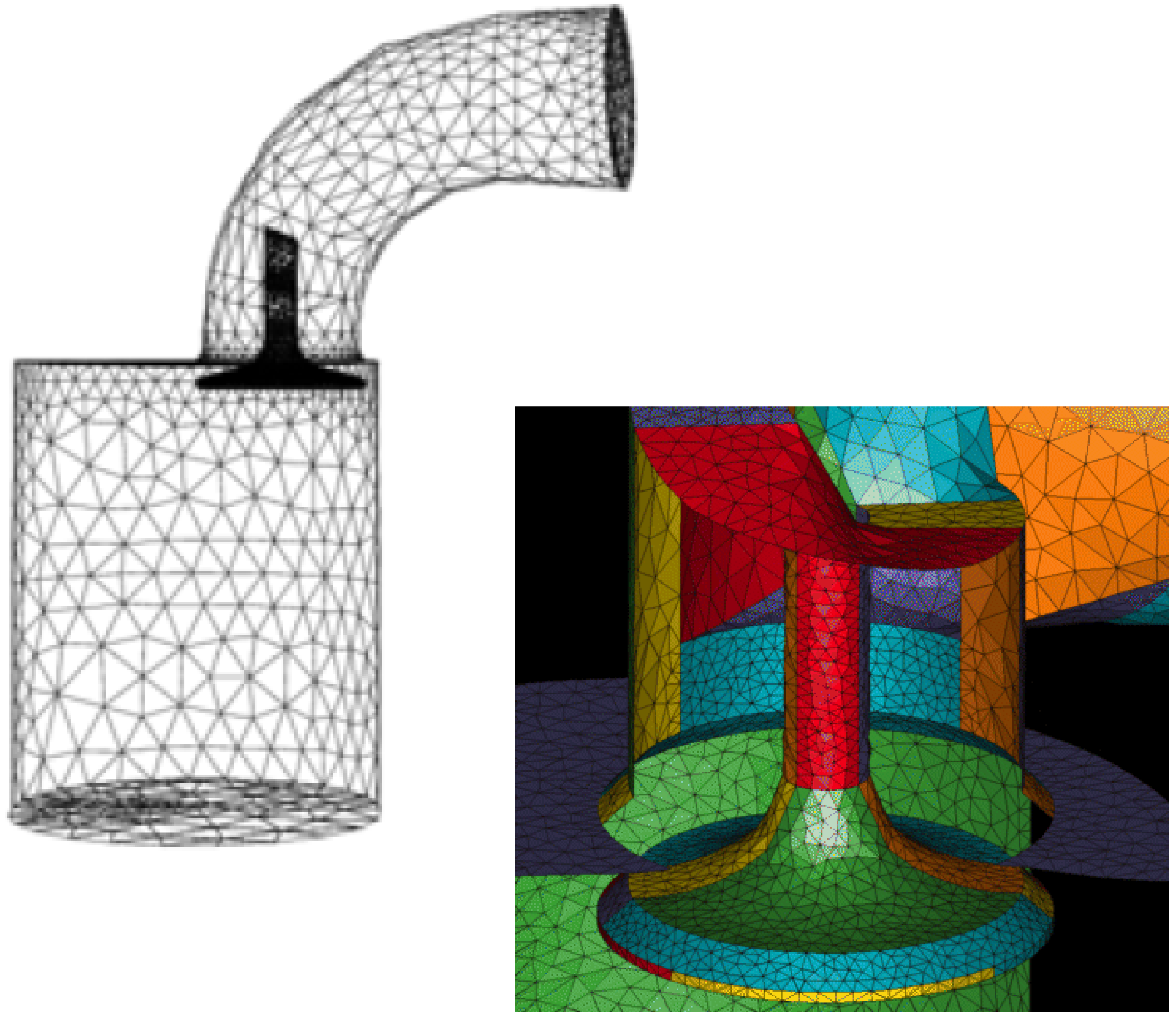
Hybrid tri/quad



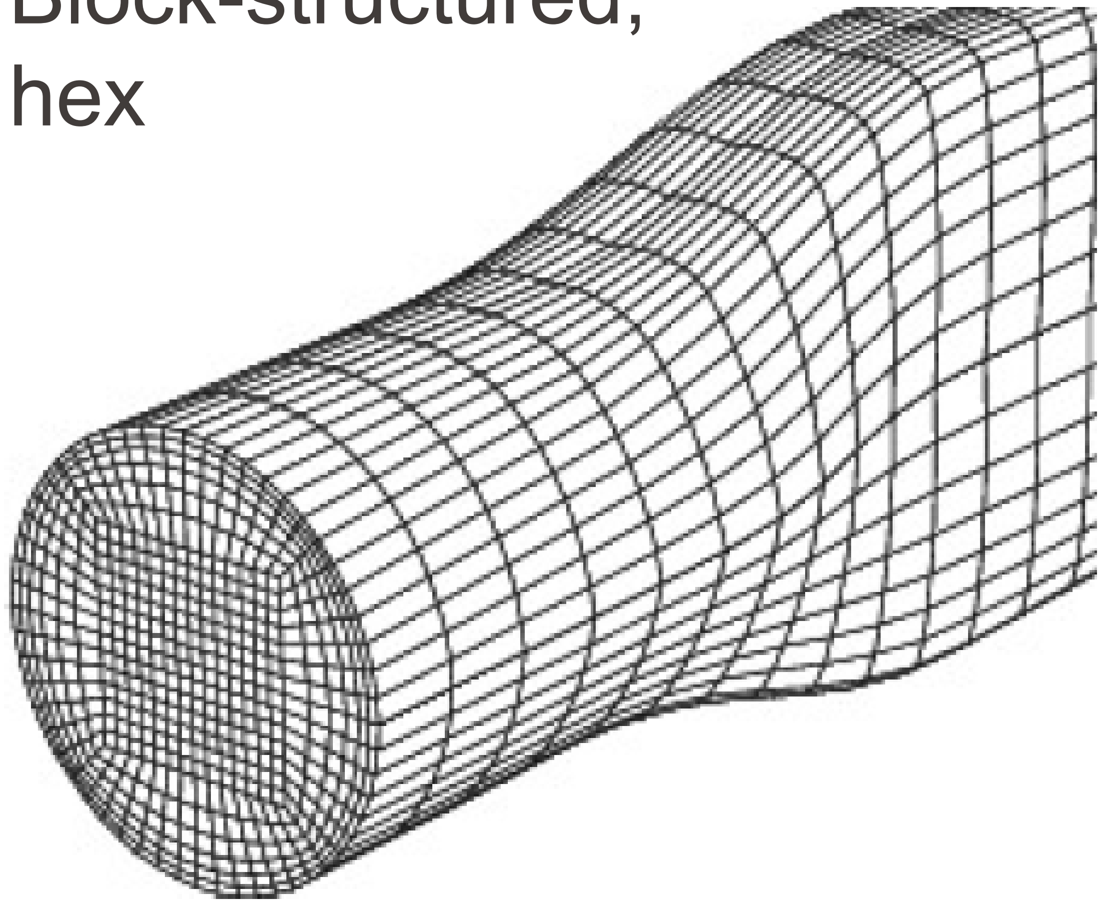


# Some 3D examples

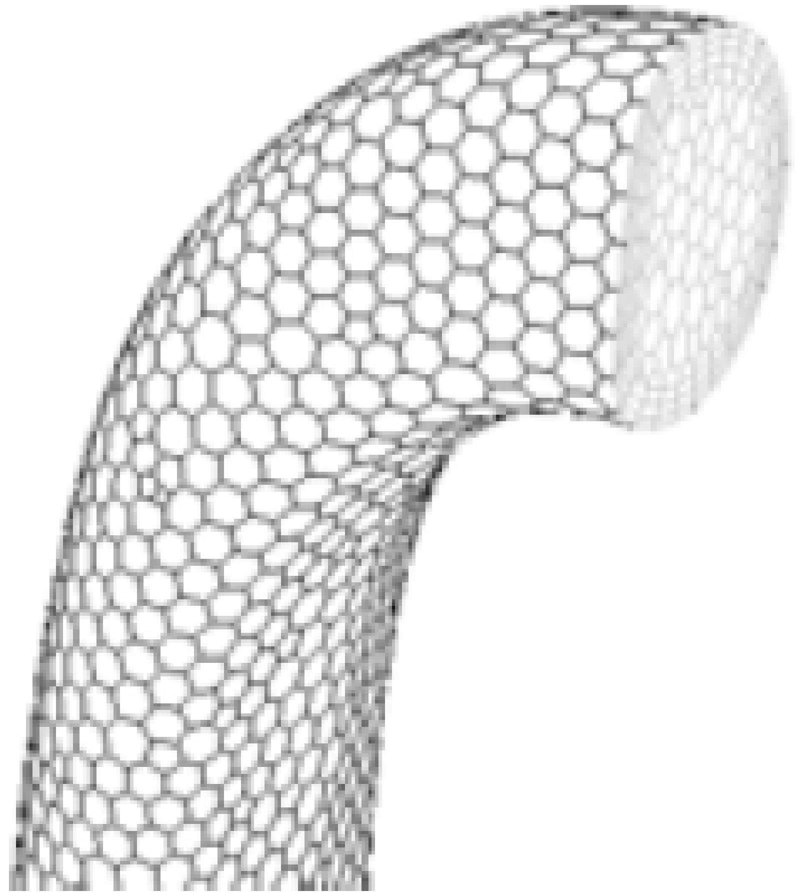
Unstructured, tetra



Block-structured,
hex



Polyhedral



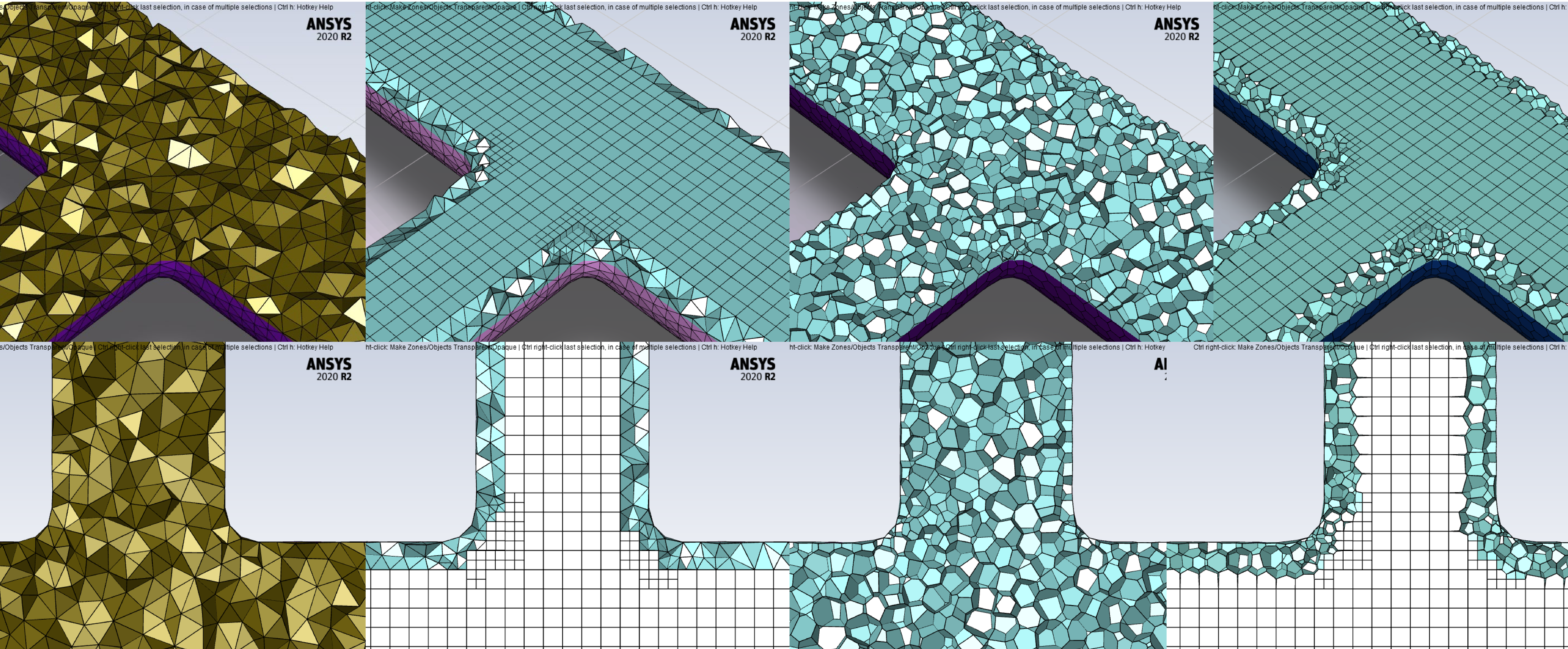
# Some 3D examples

Unstructured,
tetrahedra

Hybrid, tetra/hexa
("hexcore")

Unstructured,
polyhedra

Hybrid, poly/hexa
("poly-hexcore")



# What is a good mesh?

- Importance of:
    - Elements size: trade-off between accuracy and computational cost (time, memory).
    - Elements distribution: not all regions need the same level of details. For ex., large velocity gradients in shear layers (wake, jet, separation), boundary layers, shocks.
    - Elements shape: direct and strong impact on the solution accuracy.

- A “good mesh” should:
    - Be as coarse as possible, but as fine as necessary.
    - Use physically/numerically suitable element shapes (may differ in different regions).

- No written theory about mesh generation, no universally accepted definition of a good mesh. Rather a set of good standard practices.
- Meshing requires experience.

# What is a good mesh?

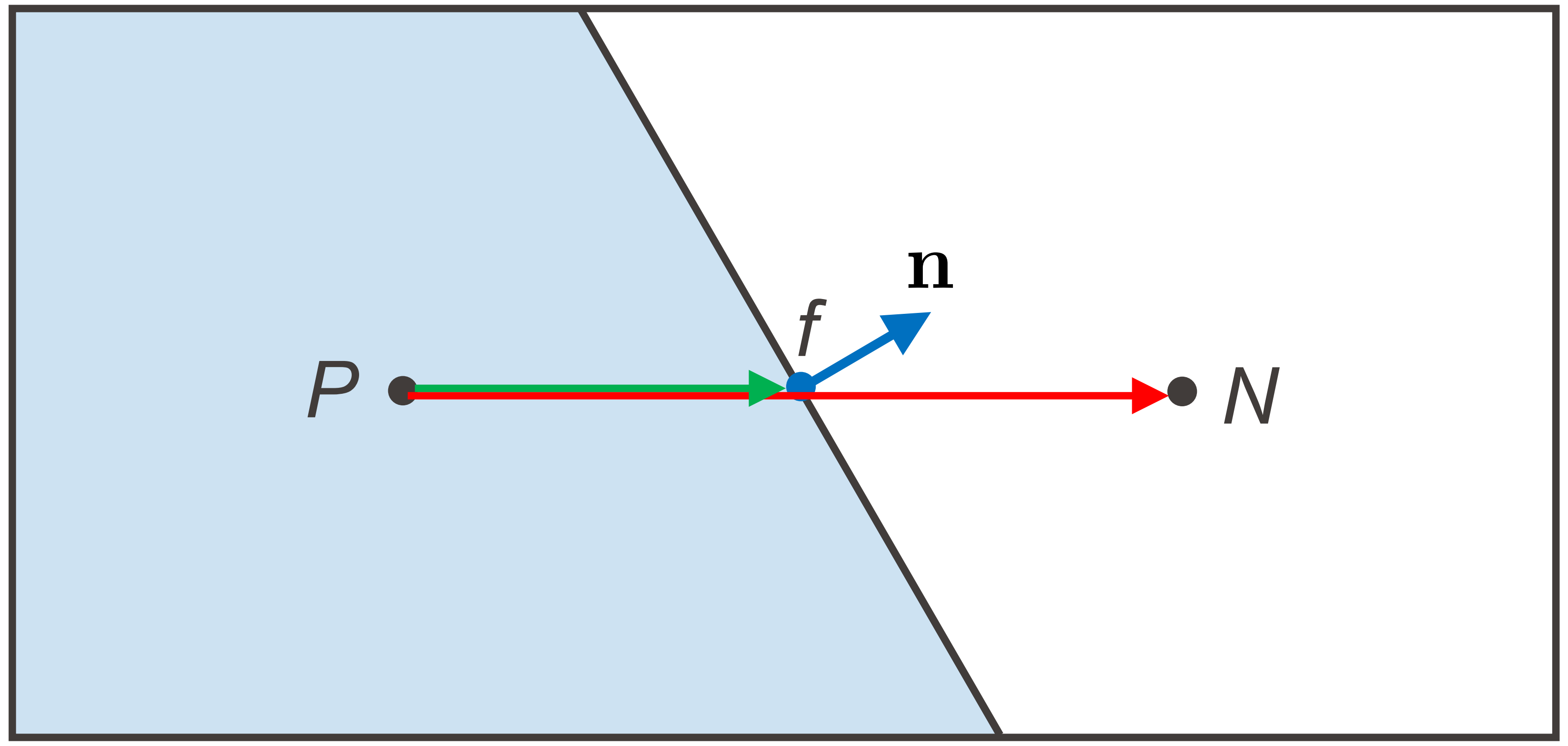
- **Qualitative** (subjective) rule of thumb: elements shape and distribution should be pleasing to the eye. (Strong element distortion and spatial variation to be avoided.)

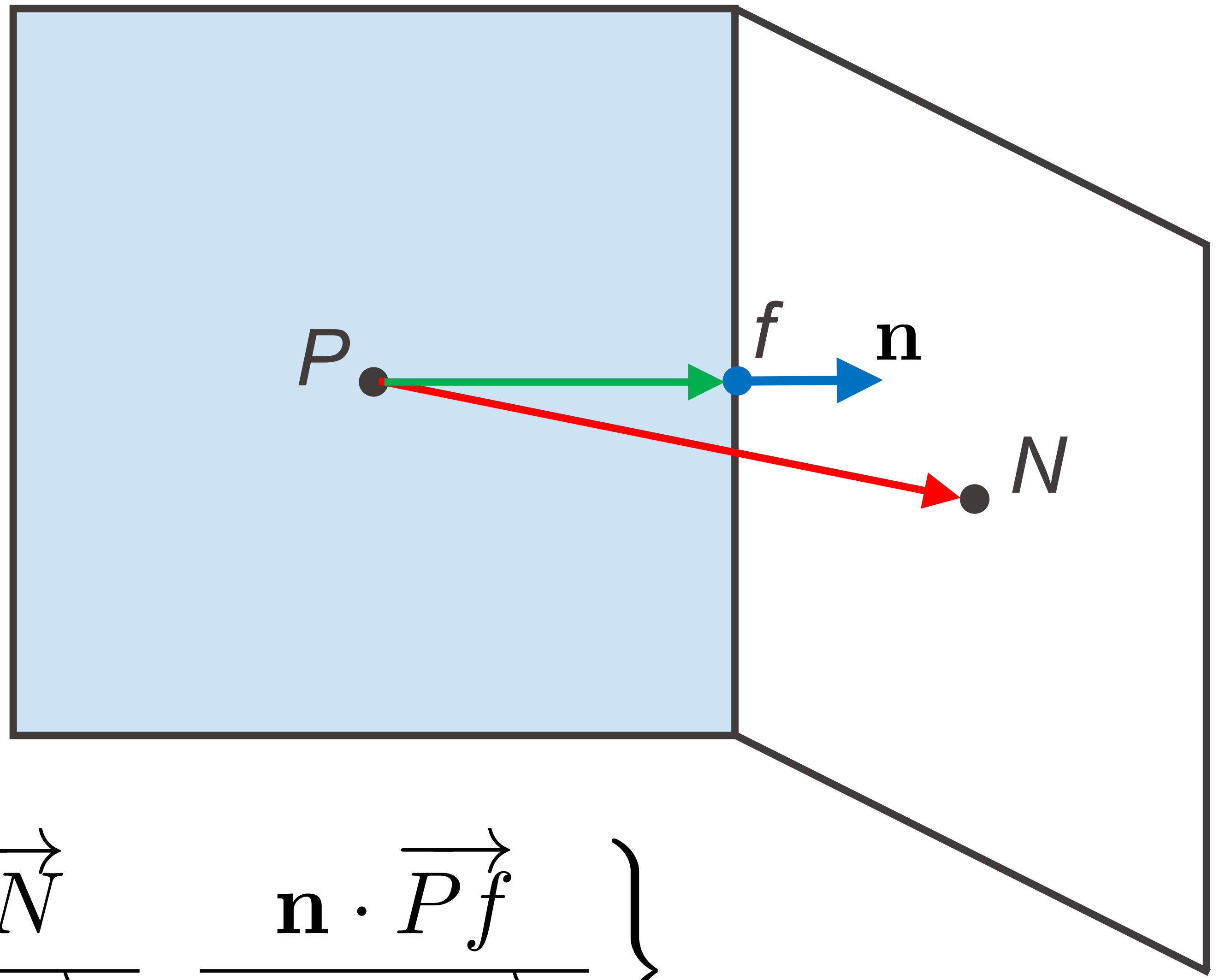
- **Quantitative** criteria: mesh **quality metrics**. Significant role in accuracy and stability of the computation. For example:
  - Size
  - Orthogonality
  - Skewness
  - Aspect ratio
  - Smoothness
  - …

- Should always evaluate these metrics to assess the mesh quality.
  (If not to aim for an exceptionally good mesh, at least to avoid a bad mesh.)

# Mesh quality metric #1: orthogonality

- For each face, alignment between face normal vector **n** and vector *PN* (from node to neighbor node) or vector *Pf* (from node to face center).





- Fluent: for each element, evaluate

$$\min_{\text{faces}}\left\{\frac{\mathbf{n}\cdot\overrightarrow{PN}}{||\mathbf{n}||\,||\overrightarrow{PN}||},\frac{\mathbf{n}\cdot\overrightarrow{Pf}}{||\mathbf{n}||\,||\overrightarrow{Pf}||}\right\}$$

  (0: worst, 1: best)

- Non-orthogonality affects the convective and diffusive terms, and adds numerical diffusion.

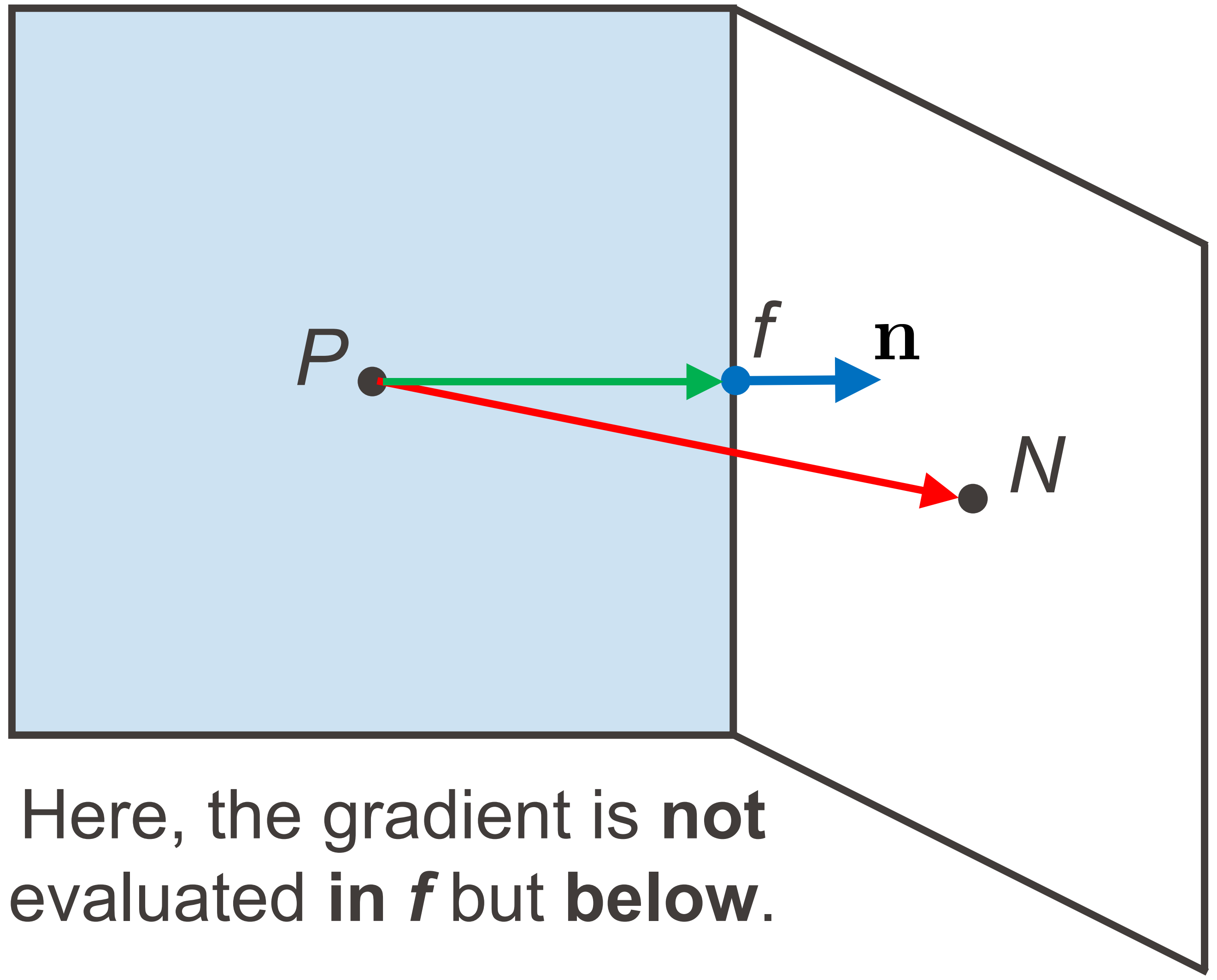
# Mesh quality metric #1: orthogonality

- For instance, the diffusive terms can be discretized with central differencing as:

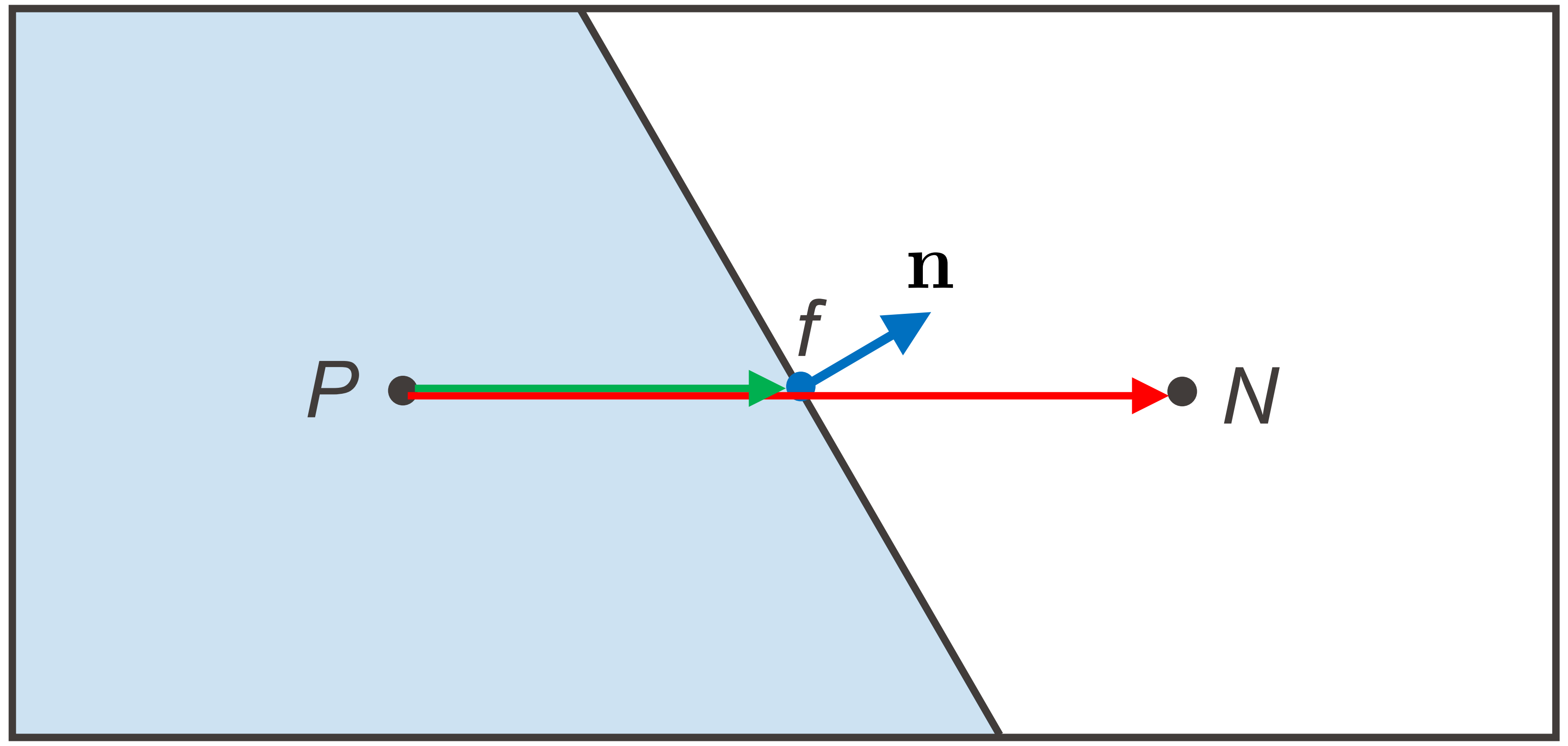
$$\int_{A_i} grad(\phi)\cdot\mathbf{n}\,dA \approx grad(\phi)|_f\cdot\mathbf{n}\,A_i \approx \frac{\phi_N-\phi_P}{||\overrightarrow{PN}||}A_i$$

2nd-order midpoint rule: accurately represented by this CD scheme only if *PN* intersects face at *f*

This CD scheme accurately represents normal gradient only if **n** aligned with *PN*



Here, the gradient is **not** evaluated **in *f*** but **below**.



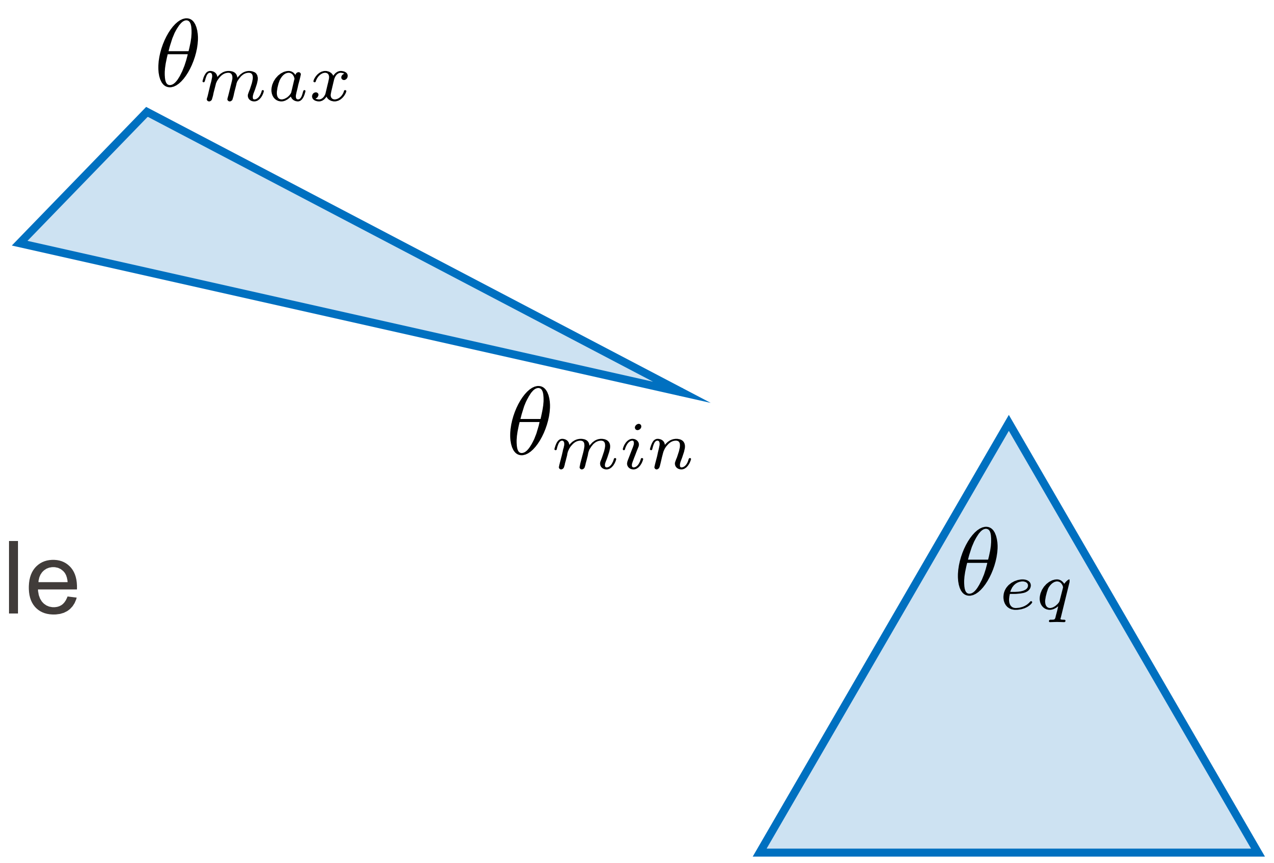
Here, the evaluated gradient is **not the normal gradient**.

# Mesh quality metric #2: skewness

- Deviation from equilateral element. For example:

$$\max\left\{\frac{\theta_{max}-\theta_{eq}}{180^\circ-\theta_{eq}},\frac{\theta_{eq}-\theta_{min}}{\theta_{eq}}\right\}$$

where $\theta_{eq}$ is the characteristic equilateral angle
(90° for quad / hex, 60° for tri / tetra)

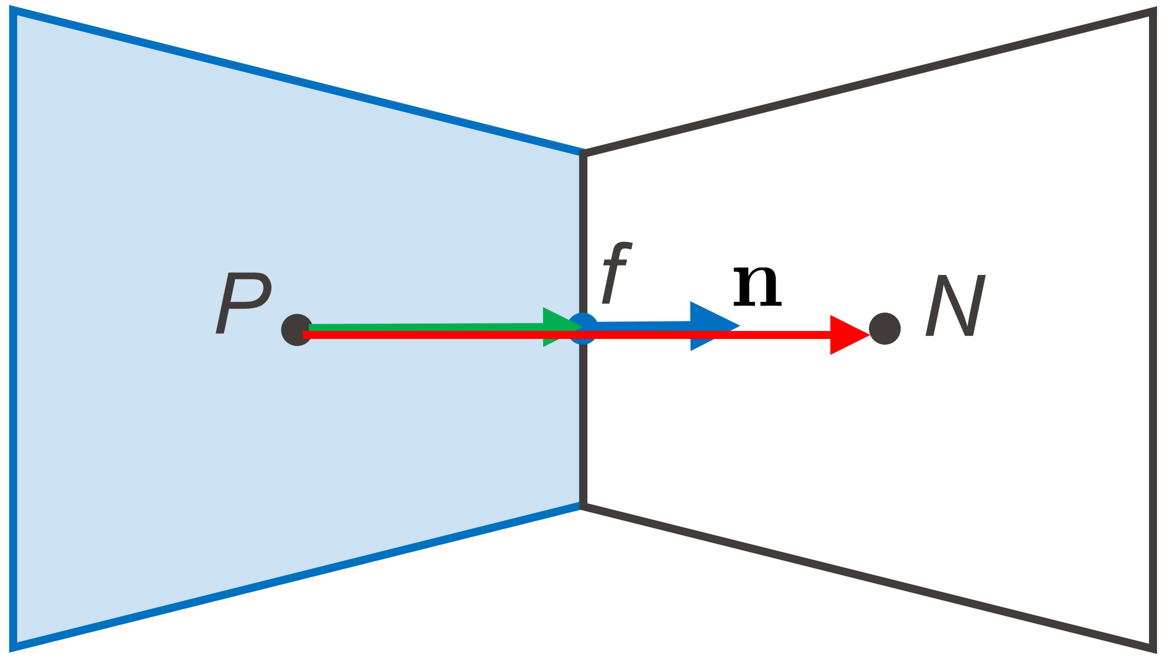


(0: best, 1: worst)

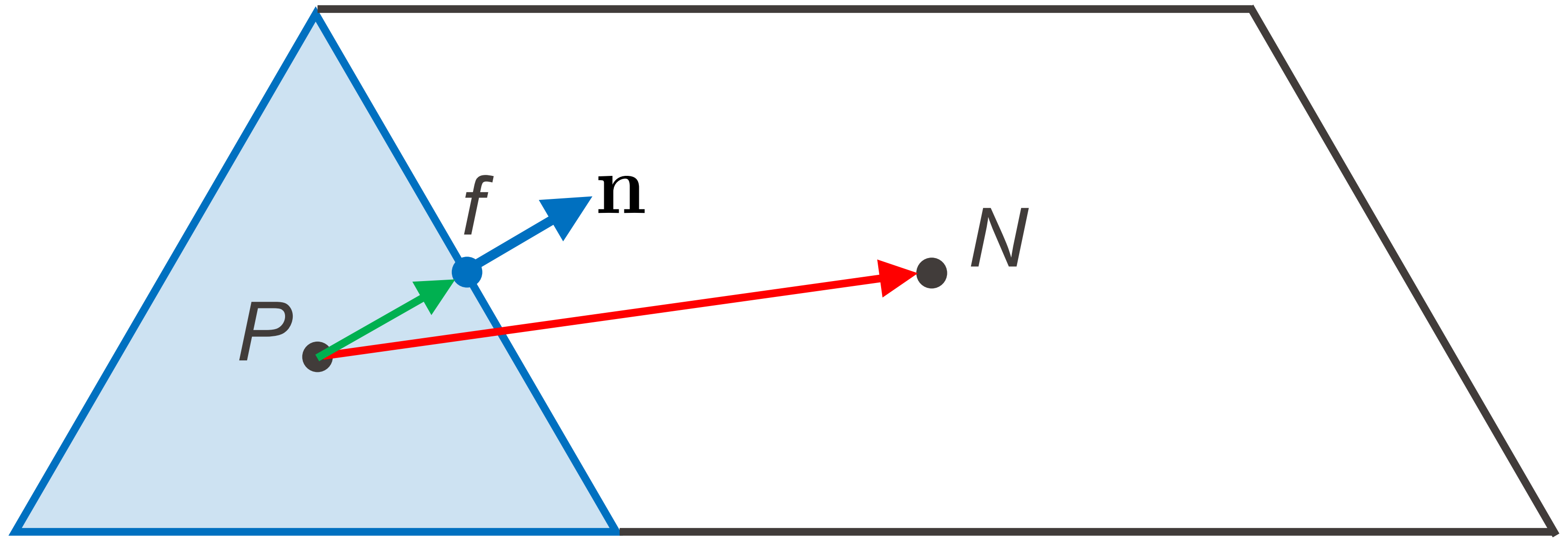
- Large skewness can lead to convergence difficulties.

# Mesh quality metrics

- In general, orthogonality and skewness are independent.



Good orthogonality,
poor skewness.

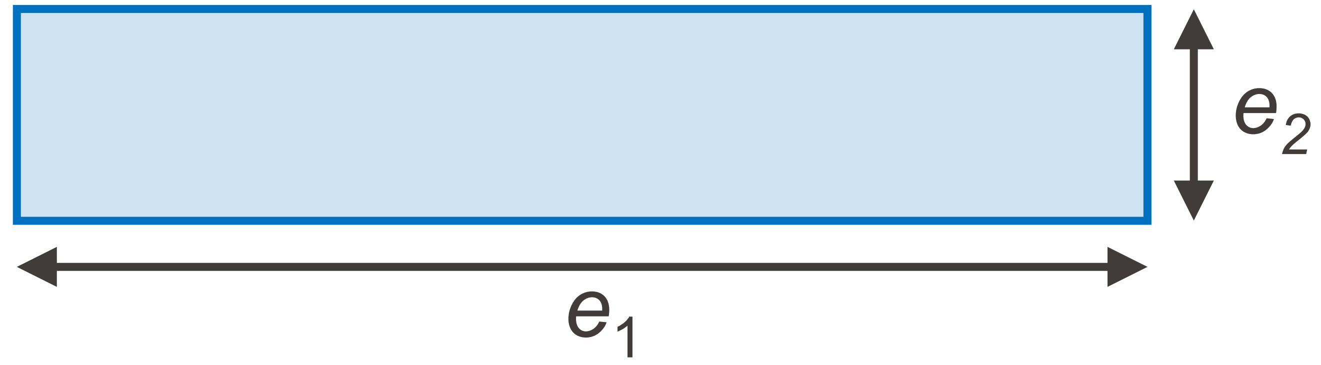


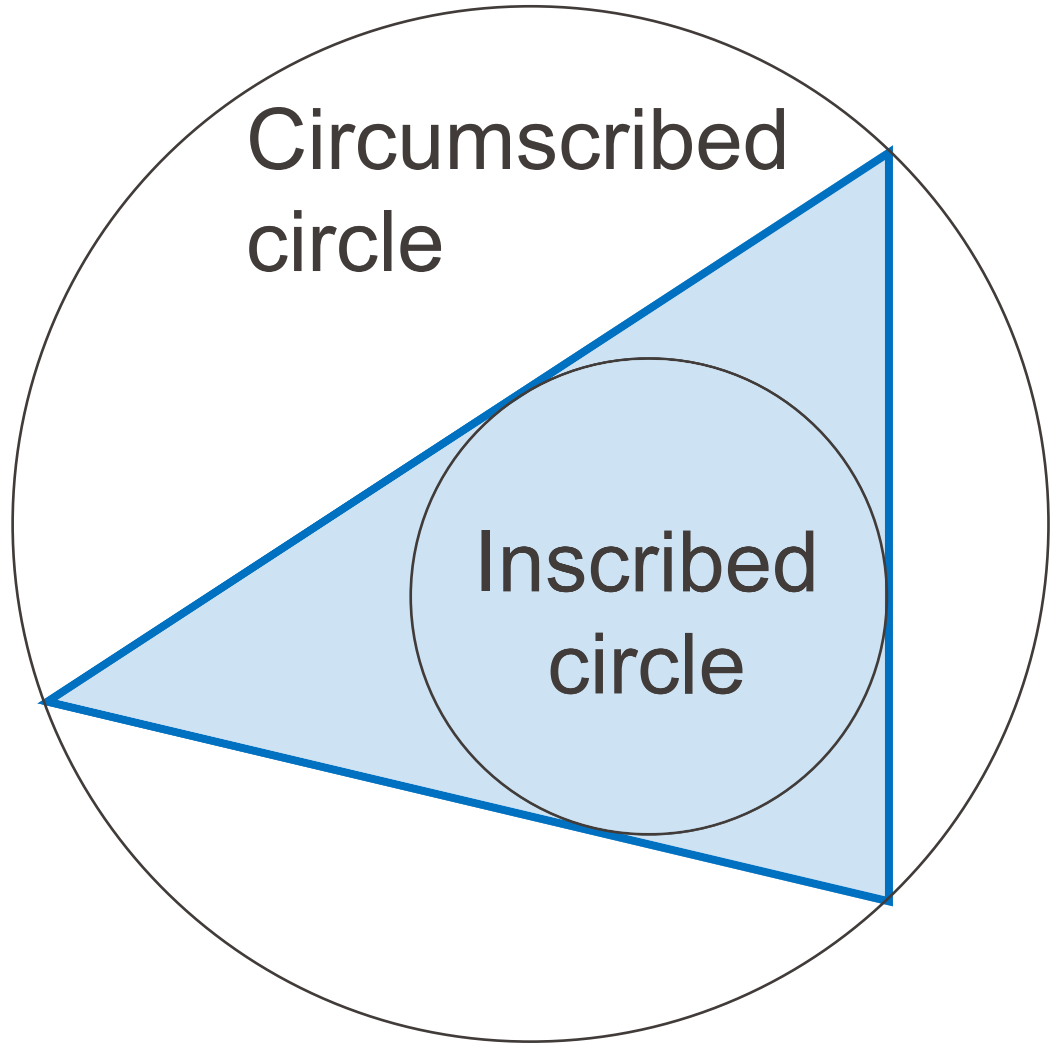
Good skewness,
poor orthogonality.

# Mesh quality metric #3: aspect ratio (AR)

- Measure of element stretching.
  - Quad / hex: ratio of longest to shortest side $\frac{\max(e_i)}{\min(e_i)}$
  - Tri / tetra: ratio of circumscribed/inscribed circles radii $\propto \frac{R}{r}$

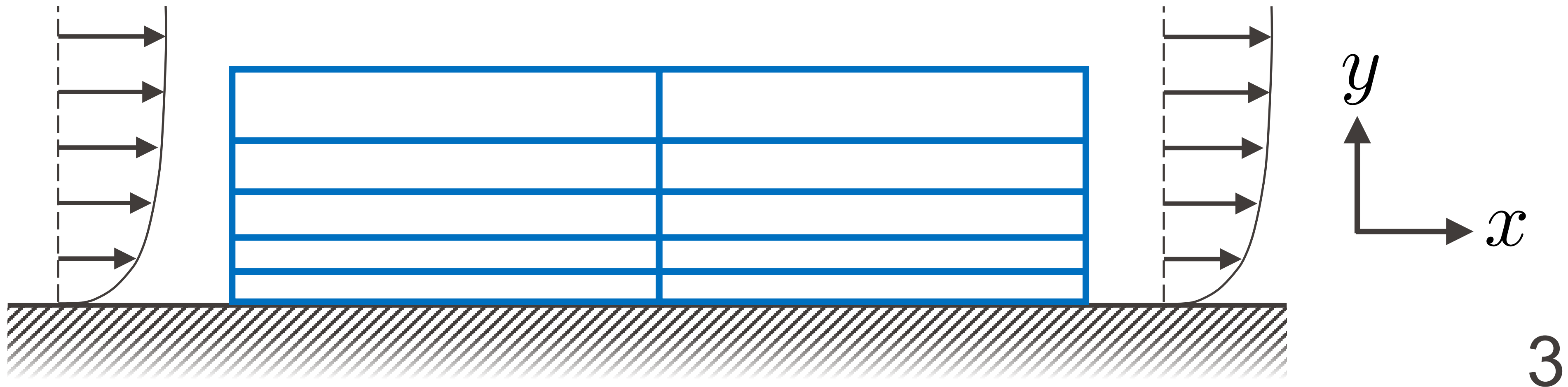
  (>>1: poor, 1: best)





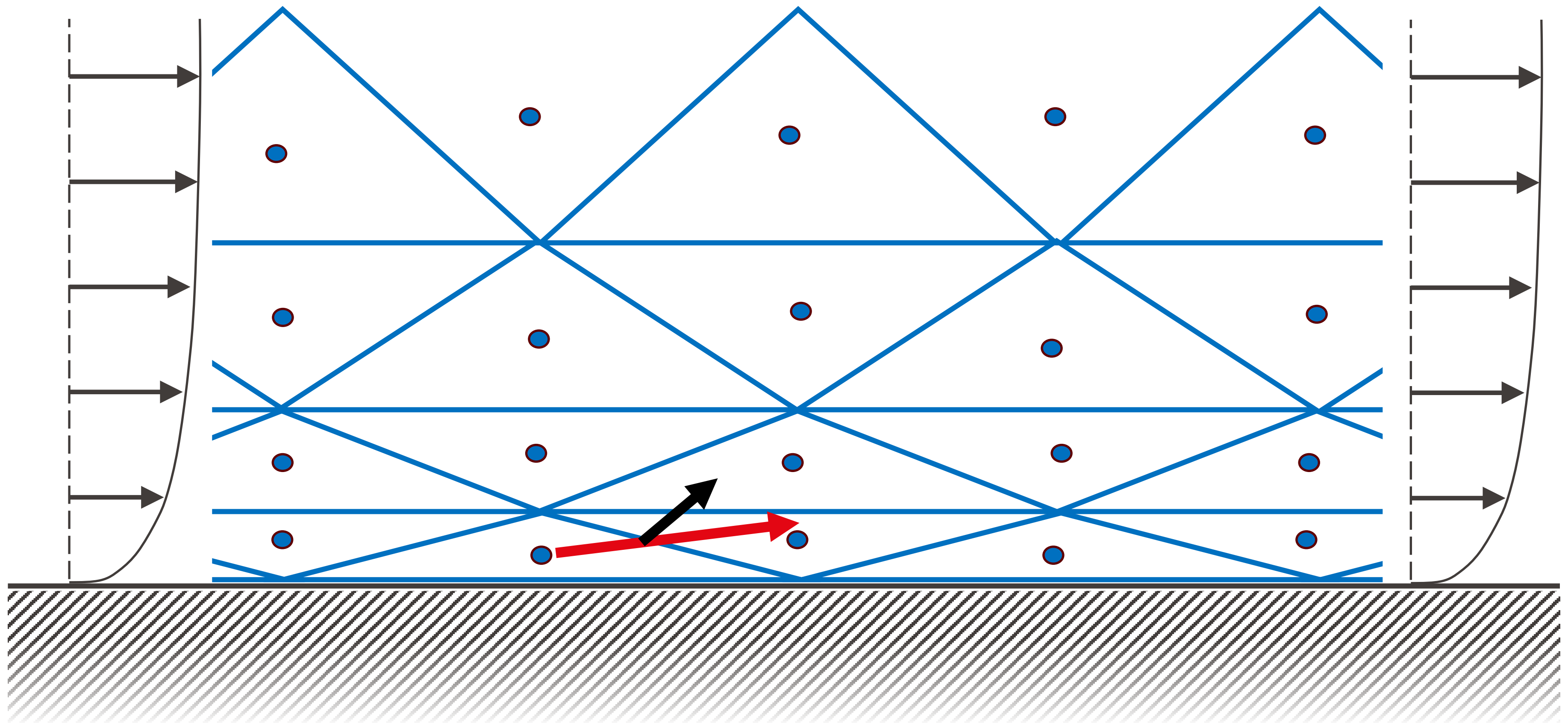
- Large AR smear gradients.
- But: large AR ok if small gradient in the long direction
  → good to use stretched quad / hex / prism elements aligned with the flow direction for highly anisotropic flows (e.g. boundary layers).

$$\phi_f = \phi_P + \underbrace{\frac{\partial \phi}{\partial x}}_{\text{Small tangential derivative}} \Delta x + \frac{\partial \phi}{\partial y} \underbrace{\Delta y}_{\text{Small normal thickness}} + O(\Delta^2)$$



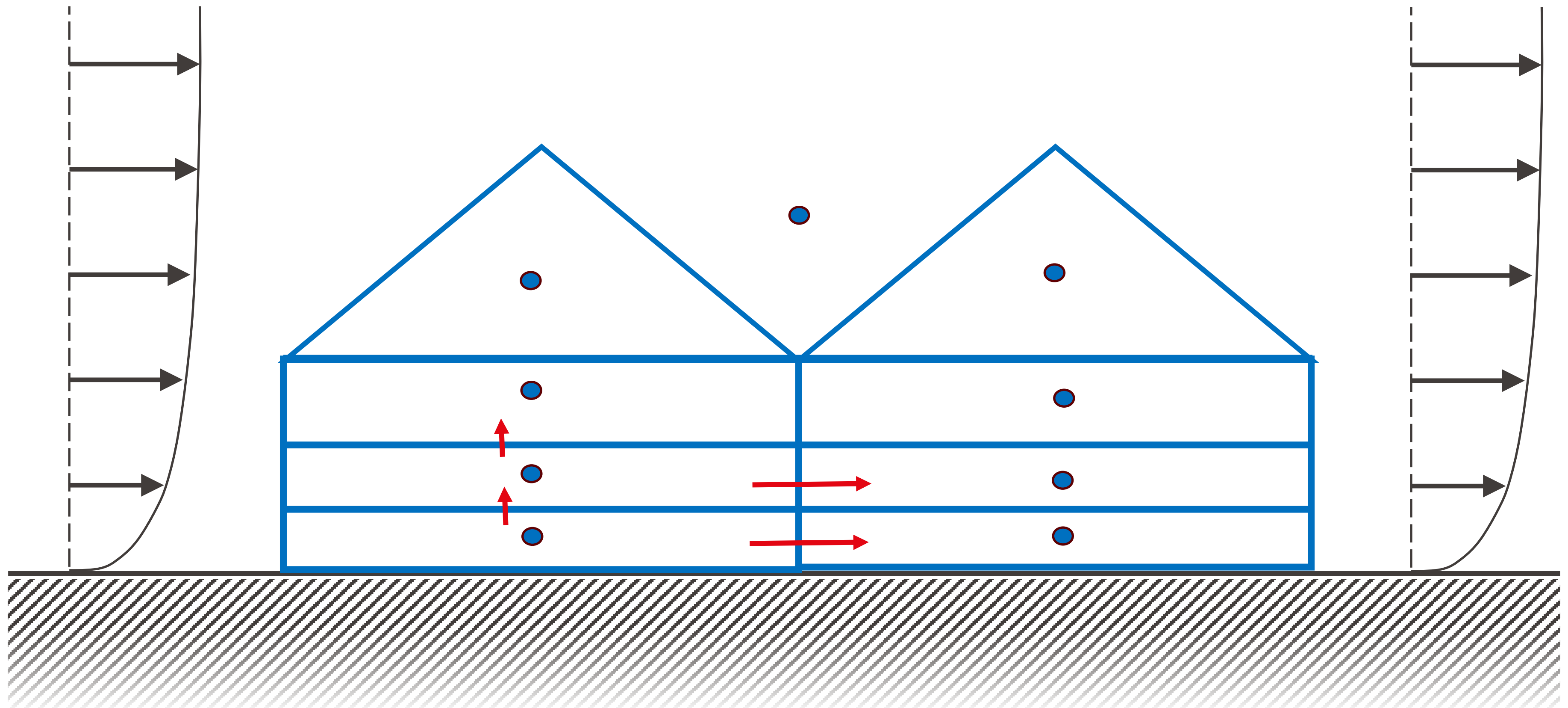
# Prism Layers near walls for unstructured mesh

- Tetrahedral mesh near the walls (still nicely arranged)



Refinement in *y*-direction exacerbates the non-orthogonality, leading to numerical diffusion in a region with high gradients.

- Prism layers (structured hexahedral mesh) for the Boundary Layer even with tetrahedra for the core

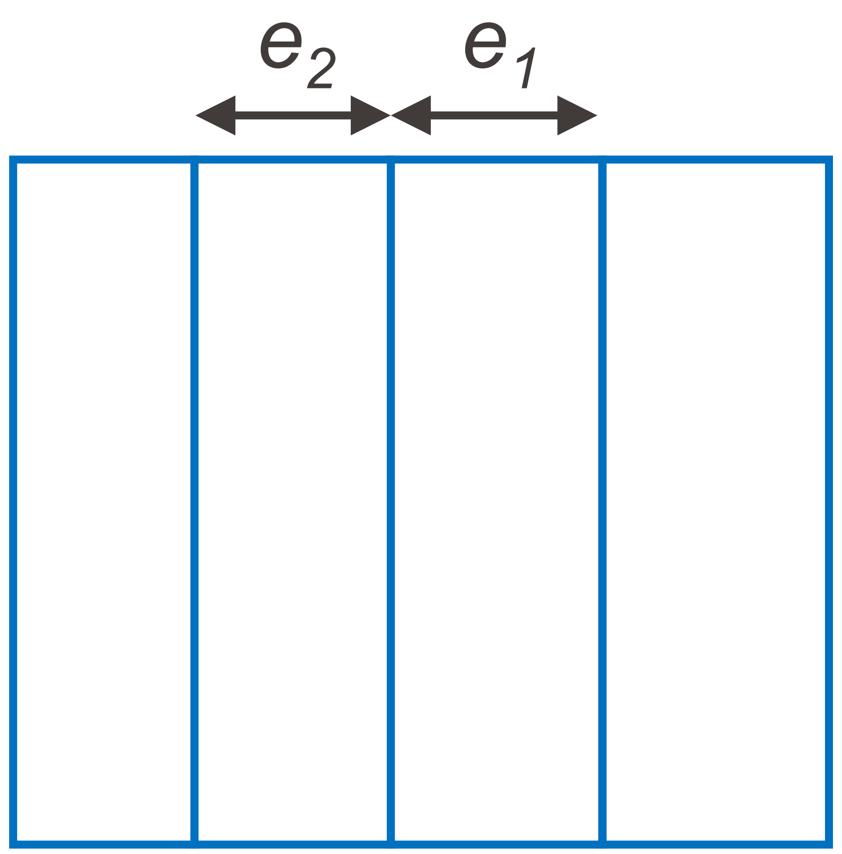


Hexahedral mesh ensures low numerical diffusion near the walls, a must for unstructured mesh.

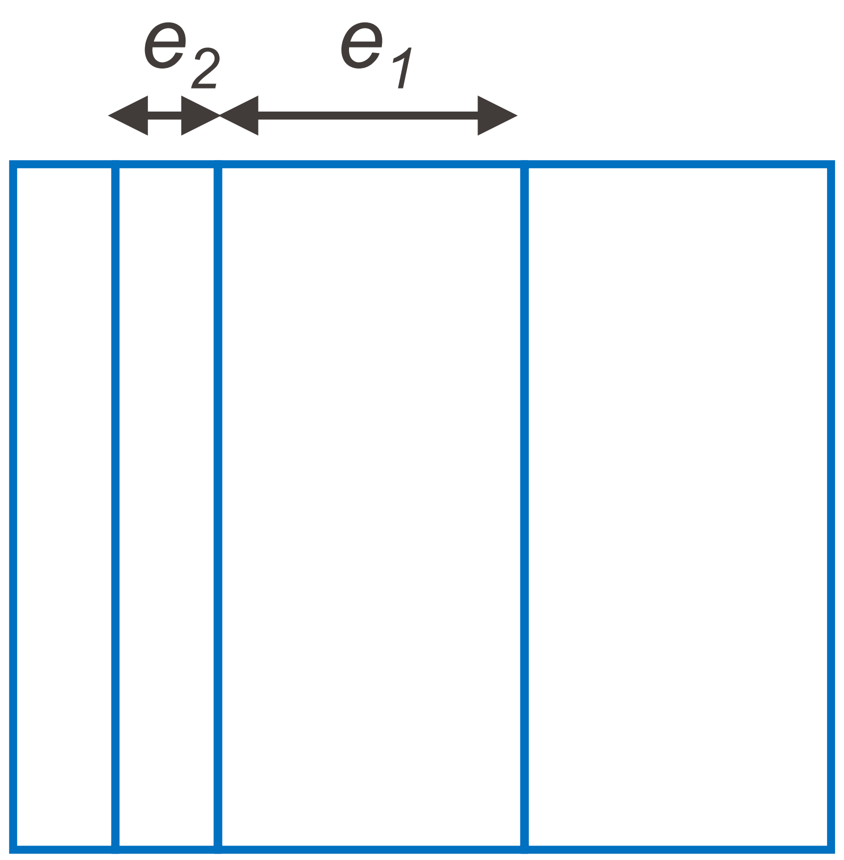
# Mesh quality metric #4: smoothness

- Or “expansion rate”, “growth factor”, “uniformity”…
- Measure of the size variation between neighbor elements. For example: $\frac{e_1}{e_2}$.

(>>1: poor, 1: best, should typically aim for values < 1.2-1.3)



Smooth variation



Rapid variation

# Mesh quality metric #4: smoothness

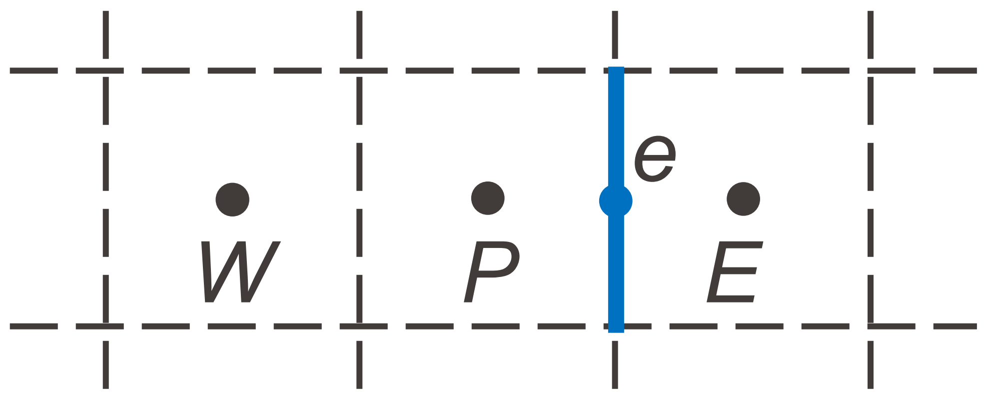
- Or “expansion rate”, “growth factor”, “uniformity”…
- Measure of the size variation between neighbor elements. For example: $\frac{e_1}{e_2}$ .

(>>1: poor, 1: best, should typically aim for values < 1.2-1.3)

- Rapid variations lead to larger truncation errors. For example, with central differencing:

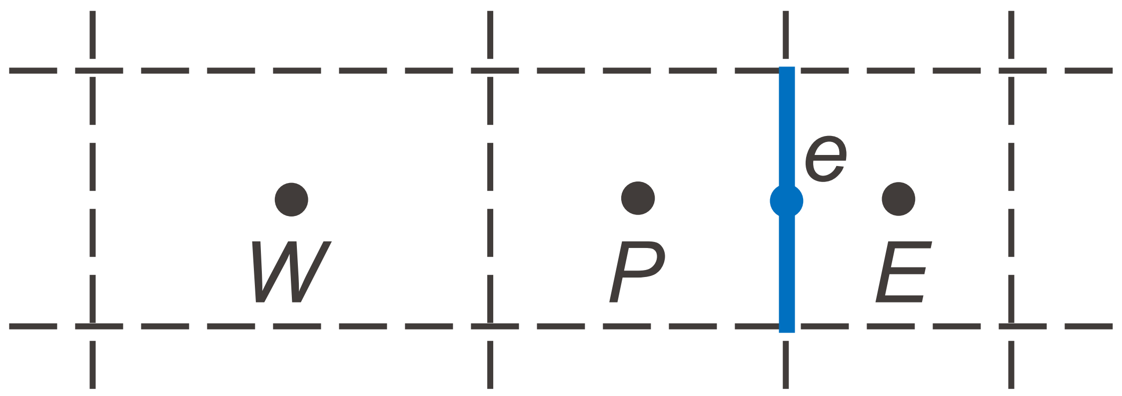
$$\left(\frac{\partial\phi}{\partial x}\right)_e \simeq \frac{\phi_E-\phi_P}{x_E-x_P}+\frac{(x_e-x_E)^2-(x_e-x_P)^2}{2(x_E-x_P)}\left(\frac{\partial^2\phi}{\partial x^2}\right)_e+O(\delta x^2)$$

If the grid is uniform, $|x_e\text{-}x_E|=|x_e\text{-}x_P|$, 2nd-order scheme:



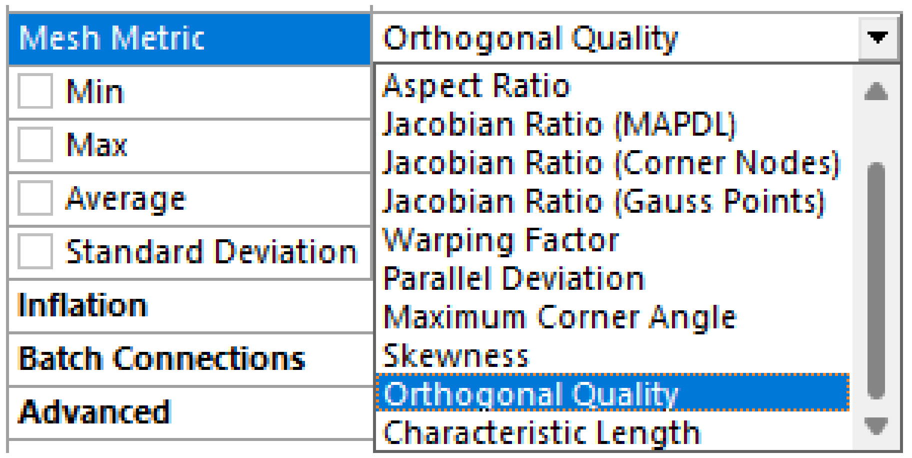
$$\left(\frac{\partial\phi}{\partial x}\right)_e \simeq \frac{\phi_E-\phi_P}{x_E-x_P}+O(\delta x^2)$$

Otherwise, effectively 1st-order scheme, with larger error as $|x_e\text{-}x_E|/|x_e\text{-}x_P|$ gets different from 1:



# Mesh quality metrics

- **In Workbench meshing:**
  - For each metric, can compute the min, max, average, and plot a distribution histogram (Details of Mesh > Quality > Mesh metric).



Mesh Metric: Orthogonal Quality

☐ Min

☐ Max

☐ Average

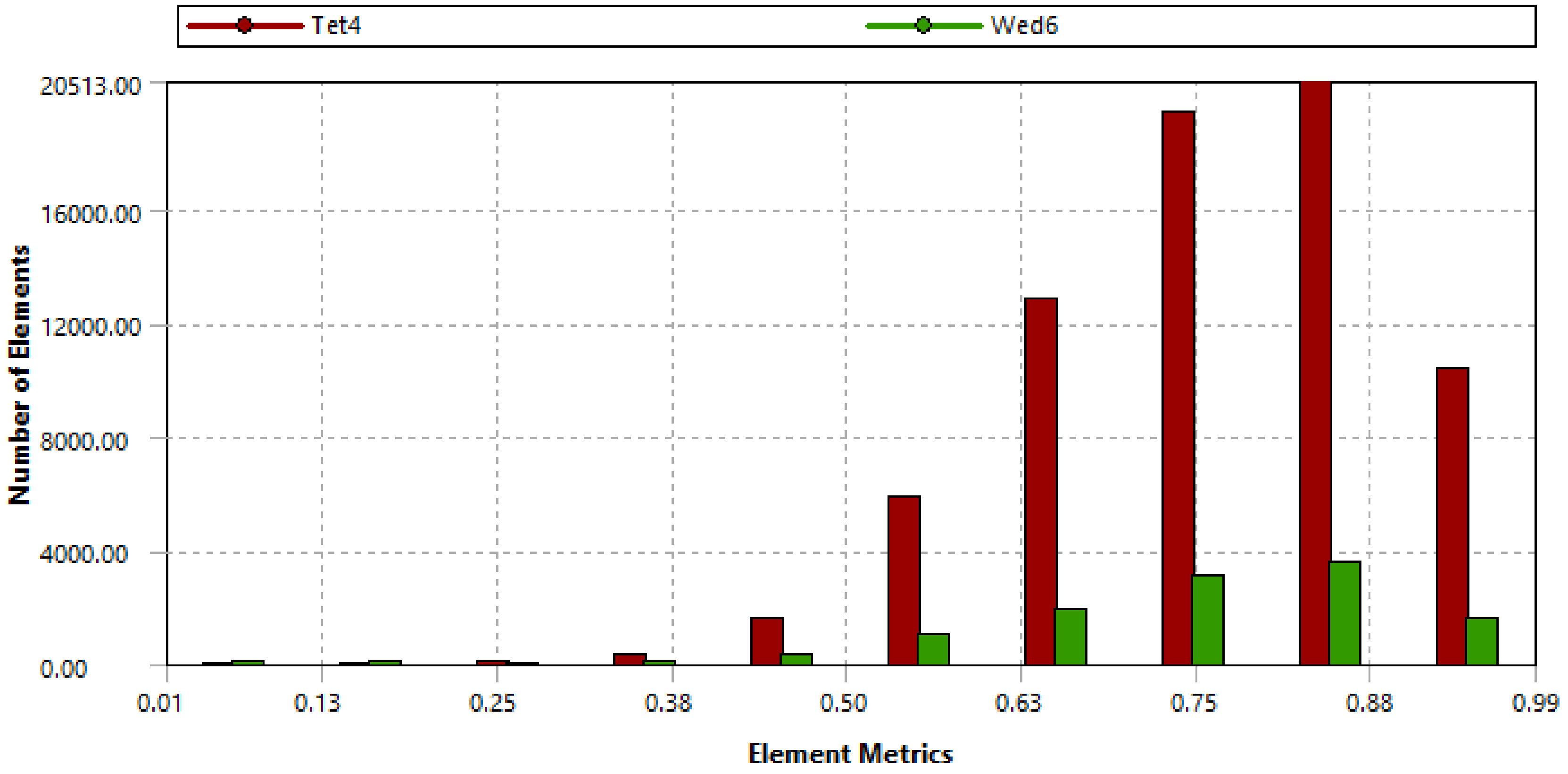
☐ Standard Deviation

Inflation

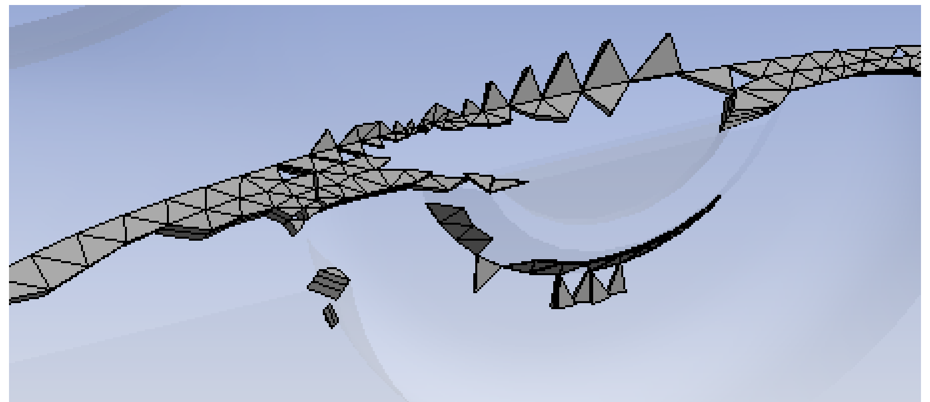
Batch Connections

Advanced

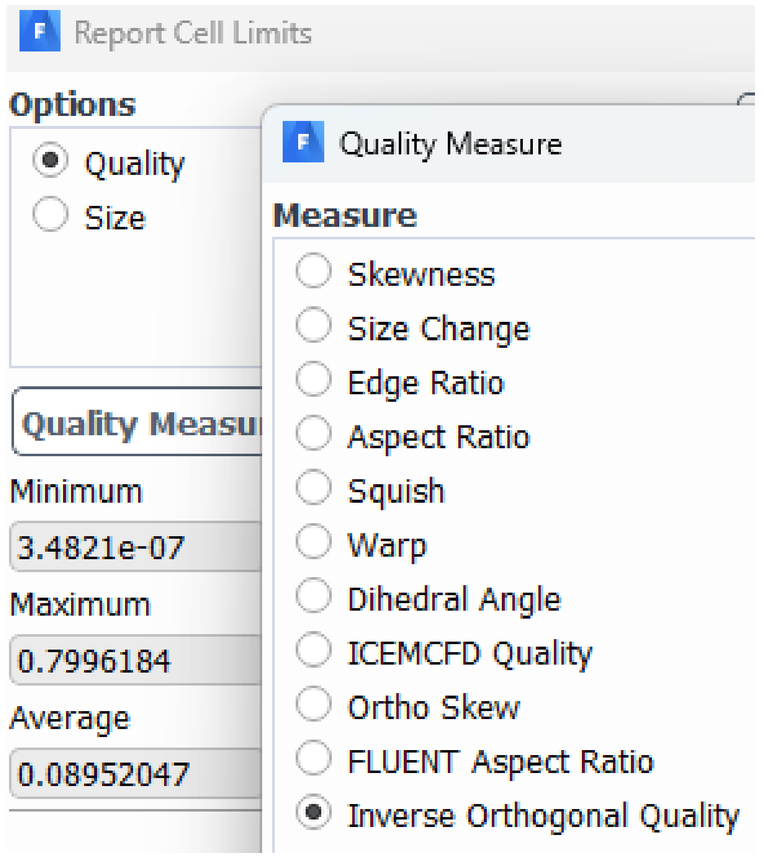
Aspect Ratio
Jacobian Ratio (MAPDL)
Jacobian Ratio (Corner Nodes)
Jacobian Ratio (Gauss Points)
Warping Factor
Parallel Deviation
Maximum Corner Angle
Skewness
Orthogonal Quality
Characteristic Length



  - Can display faces and cells falling in each quality range:



- **In Fluent (meshing mode):**
  - For each metric, can compute the min, max, average (Report > Face / Cell Limits), and plot a distribution histogram (Display > Plot > Face / Cell Distribution).



Report Cell Limits

Options

● Quality

○ Size

Quality Measure

Minimum: 3.4821e-07

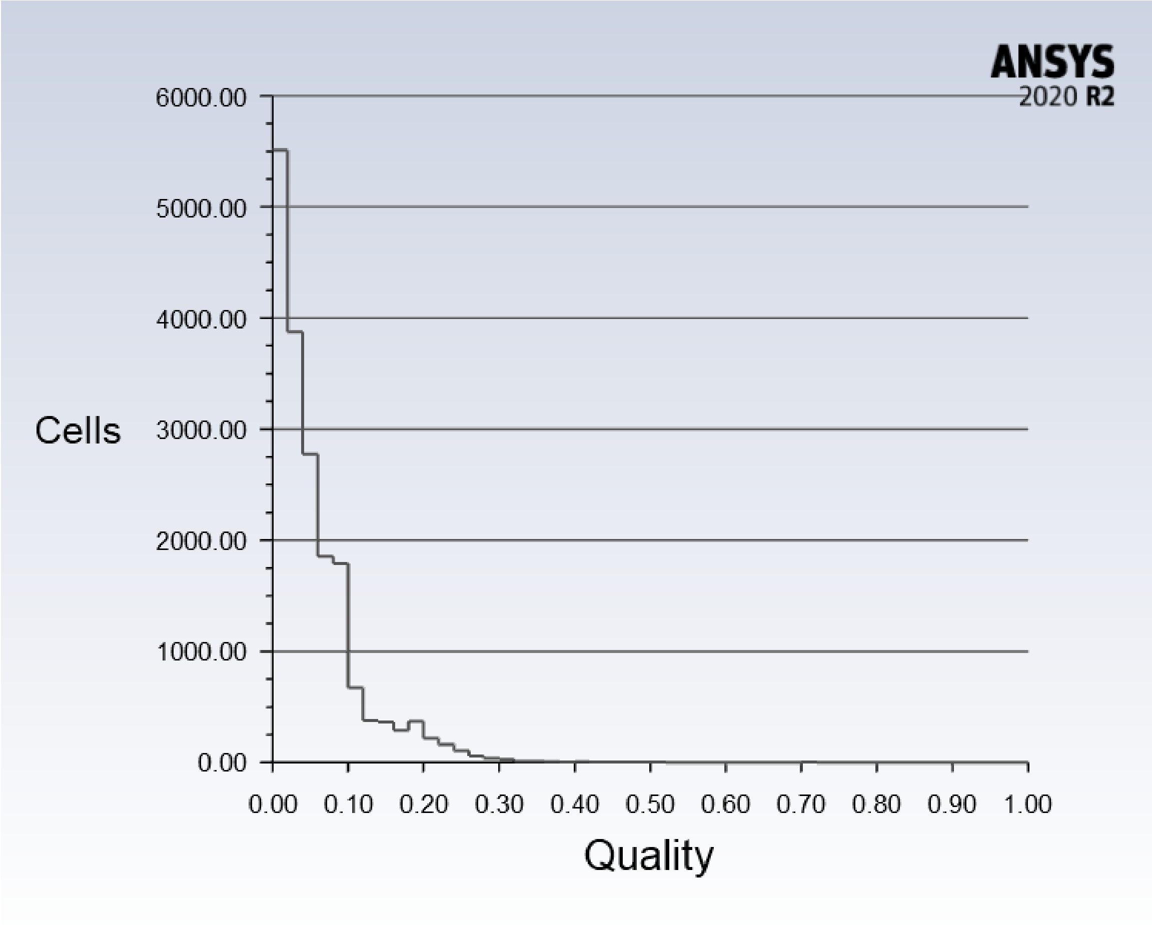
Maximum: 0.7996184

Average: 0.08952047

Quality Measure

Measure

○ Skewness
○ Size Change
○ Edge Ratio
○ Aspect Ratio
○ Squish
○ Warp
○ Dihedral Angle
○ ICEMCFD Quality
○ Ortho Skew
○ FLUENT Aspect Ratio
● Inverse Orthogonal Quality



  - Cannot display faces / cells in each quality range.

# Mesh adaption

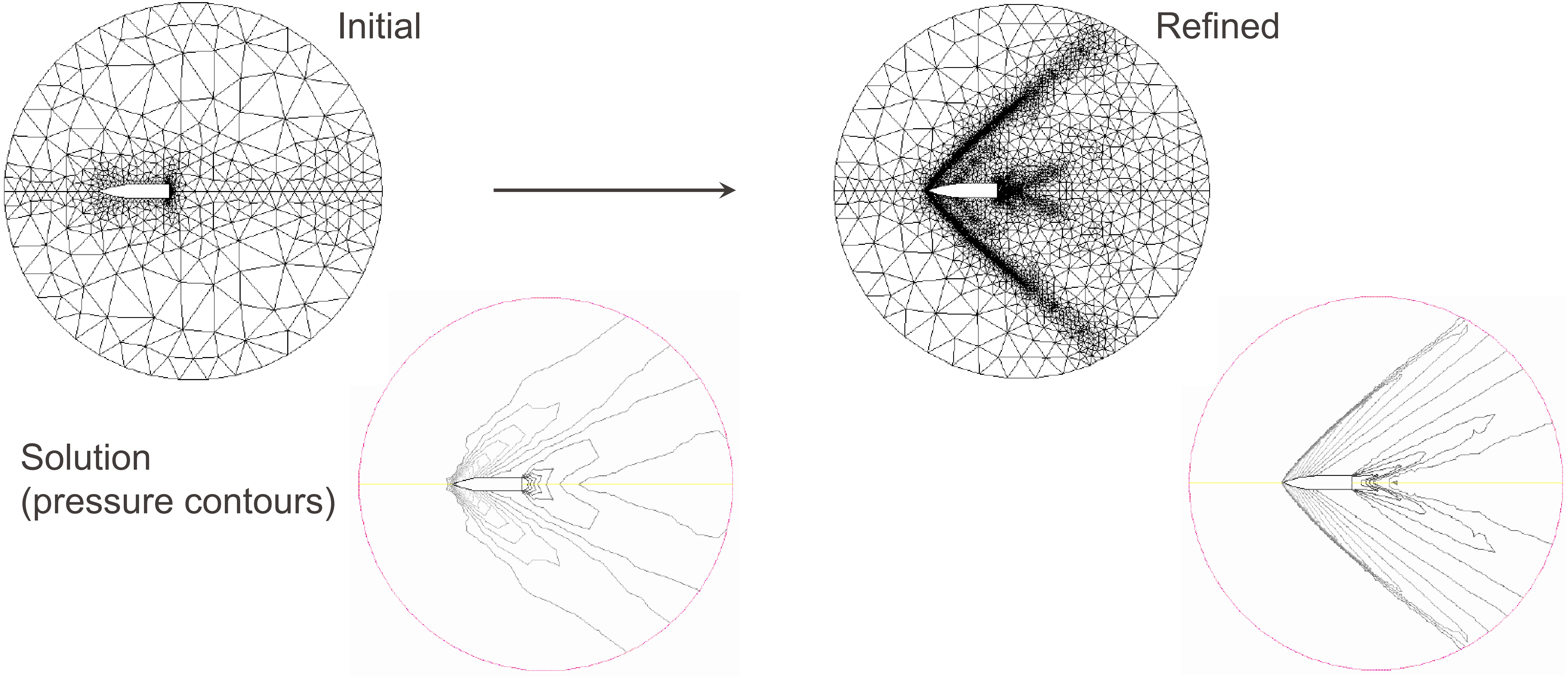
- Can improve the mesh by refining/coarsening based on the geometry and/or the flow solution.

- Useful when don't know a priori in which regions a fine mesh is needed, or when theses regions vary over time.

- For ex., Fluent provides mesh adaption based on different criteria:
    - Gradient (velocity for shear layers, pressure for shocks…)
    - Iso-value (high-velocity jets, low-pressure wakes, reaction rate for combustion…)
    - Region
    - Element size or size change
    - Wall $y^+$ or $y^*$ (turbulent flows)
    - Volume fraction (multiphase flows with Volume of Fluid method)

# Mesh adaption: example

- 2D supersonic flow around a projectile, 5 refinement cycles



Initial

Refined

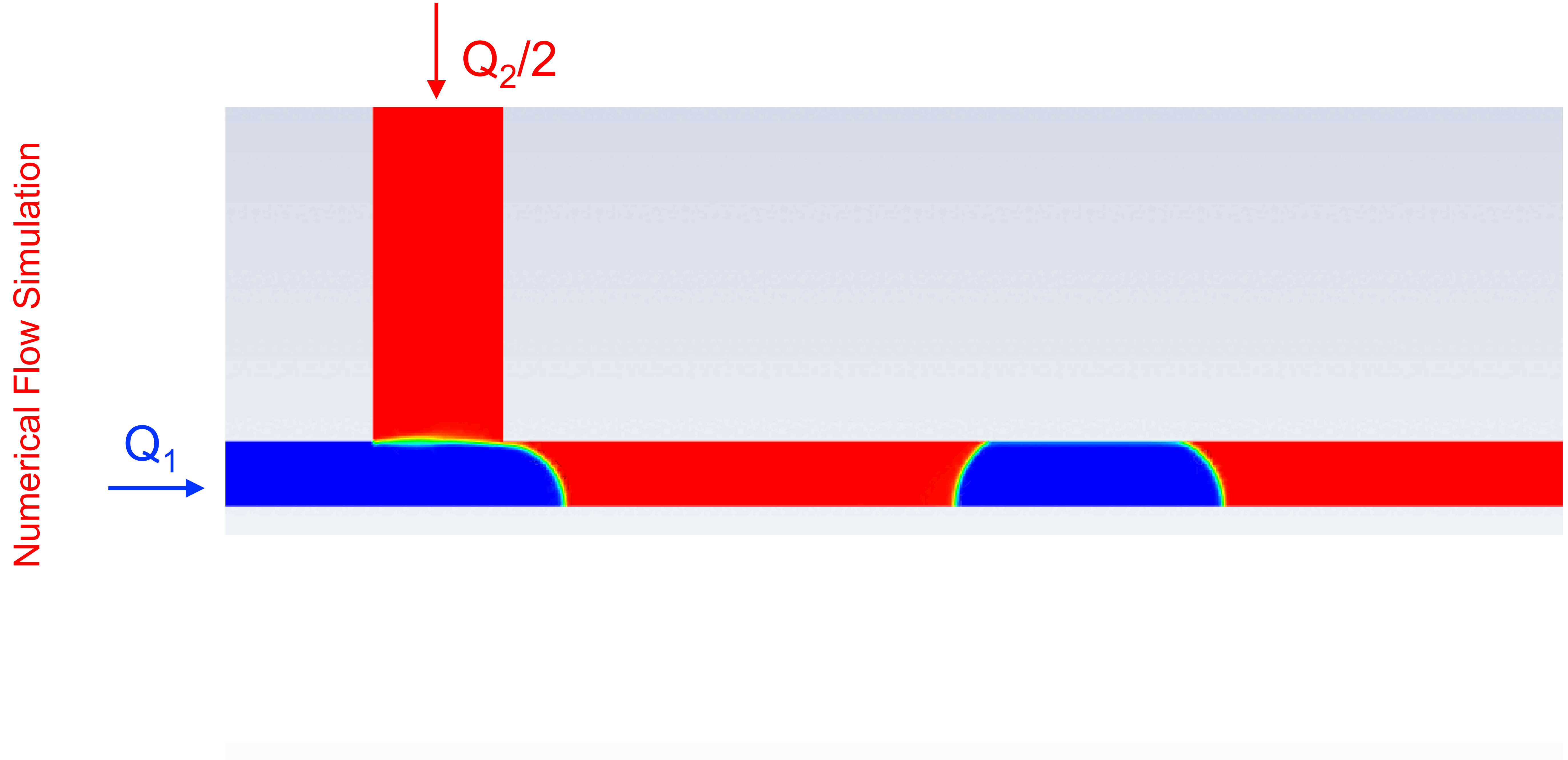
Mesh

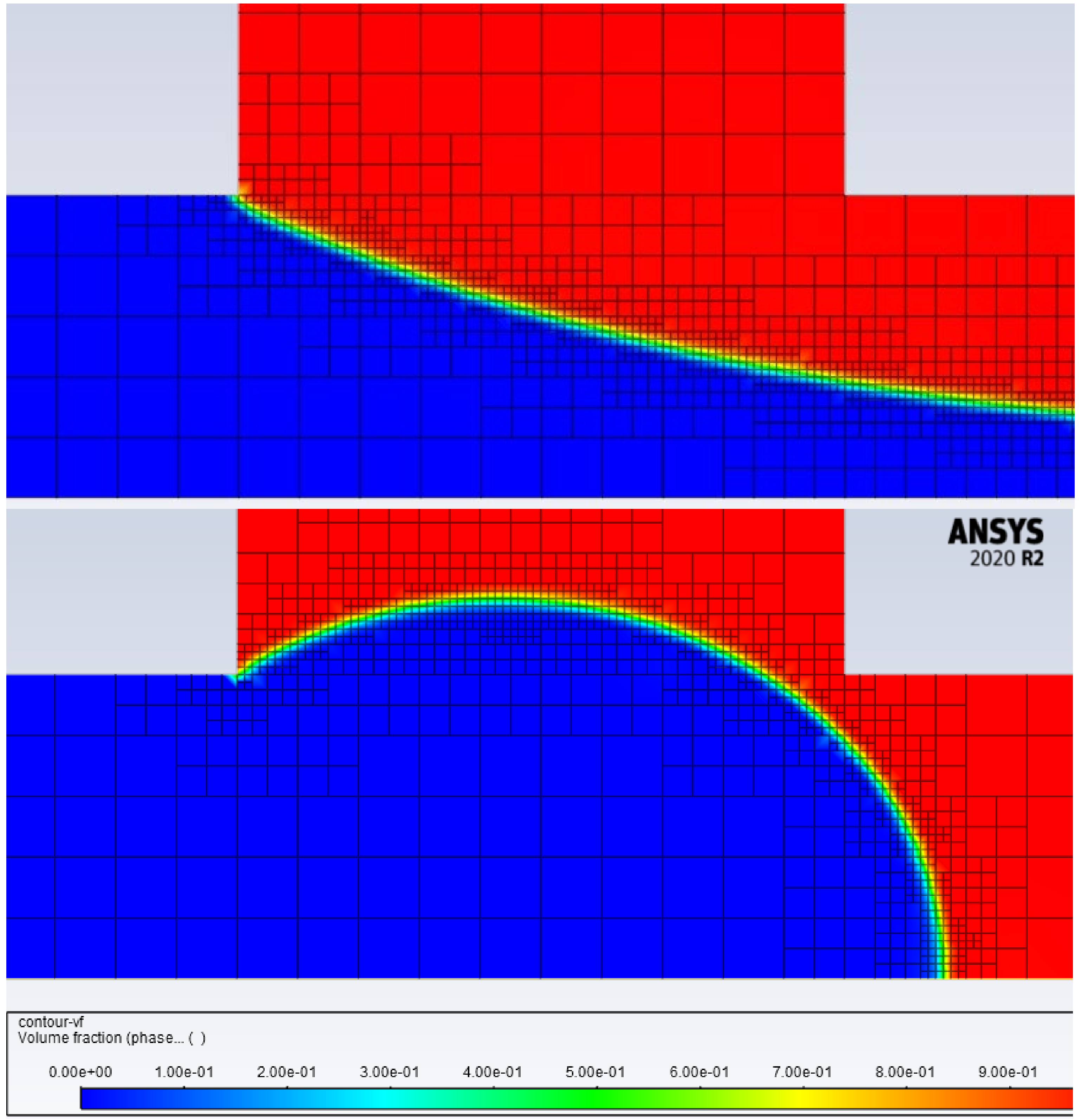
Solution
(pressure contours)

- May need to re-adapt for each flow condition.

# Mesh adaption: example

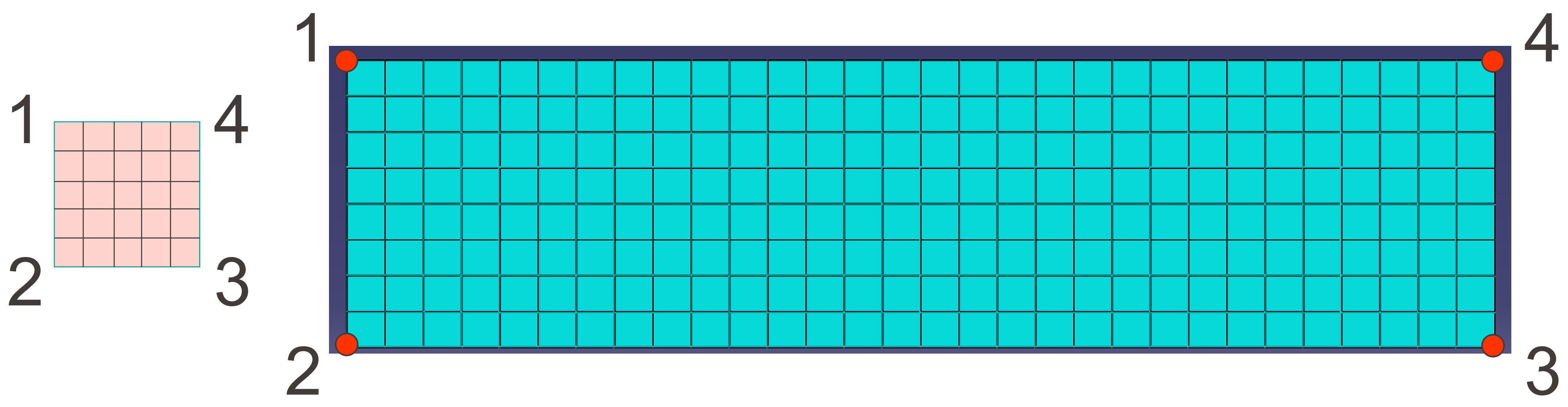
- “Flow-focusing” microfluidic device for droplet production at very low *Re*. Example of flow with an interface (2 different phases like liquid and gas, or 2 immiscible liquids like water and oil).
- Time-dependent refinement/coarsening based on volume fraction.

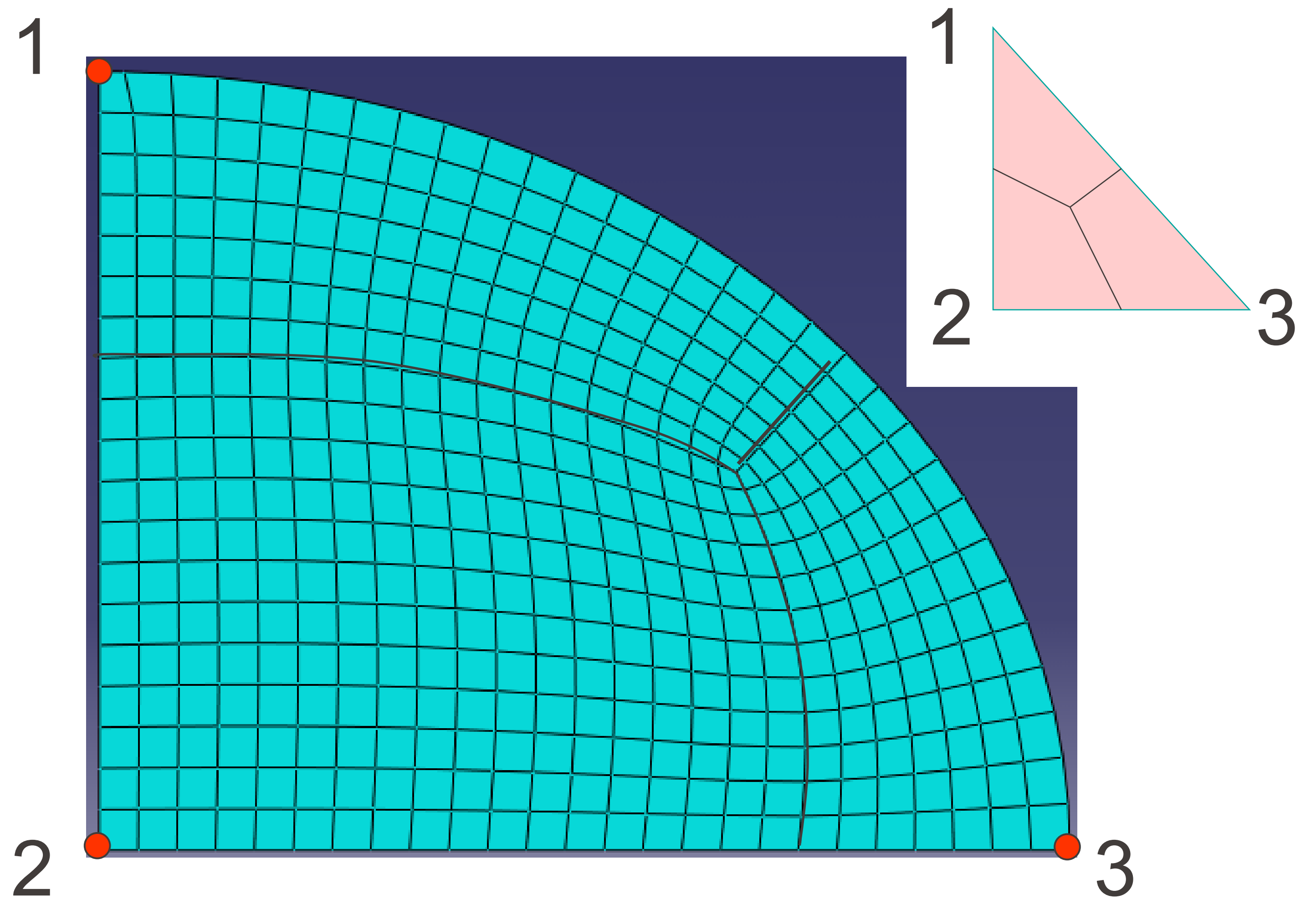


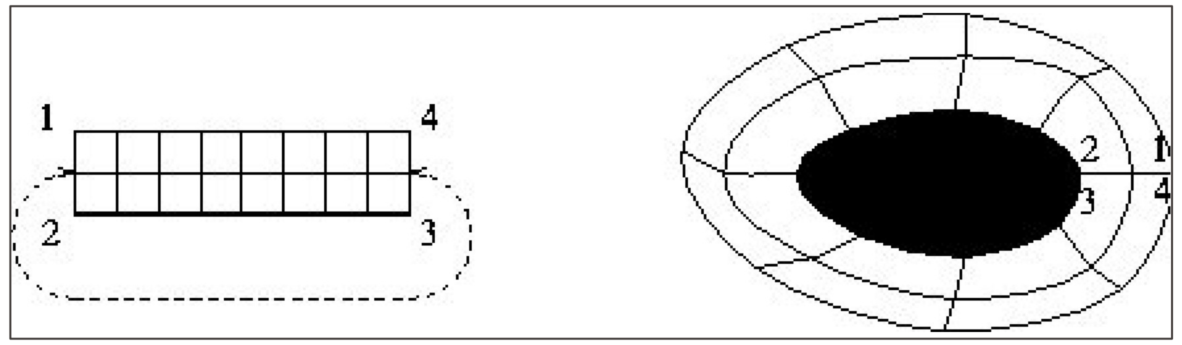


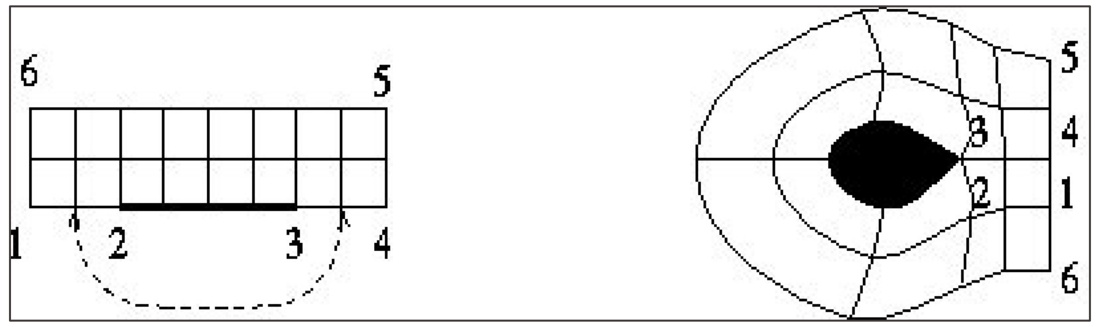
# Many different meshing algorithms

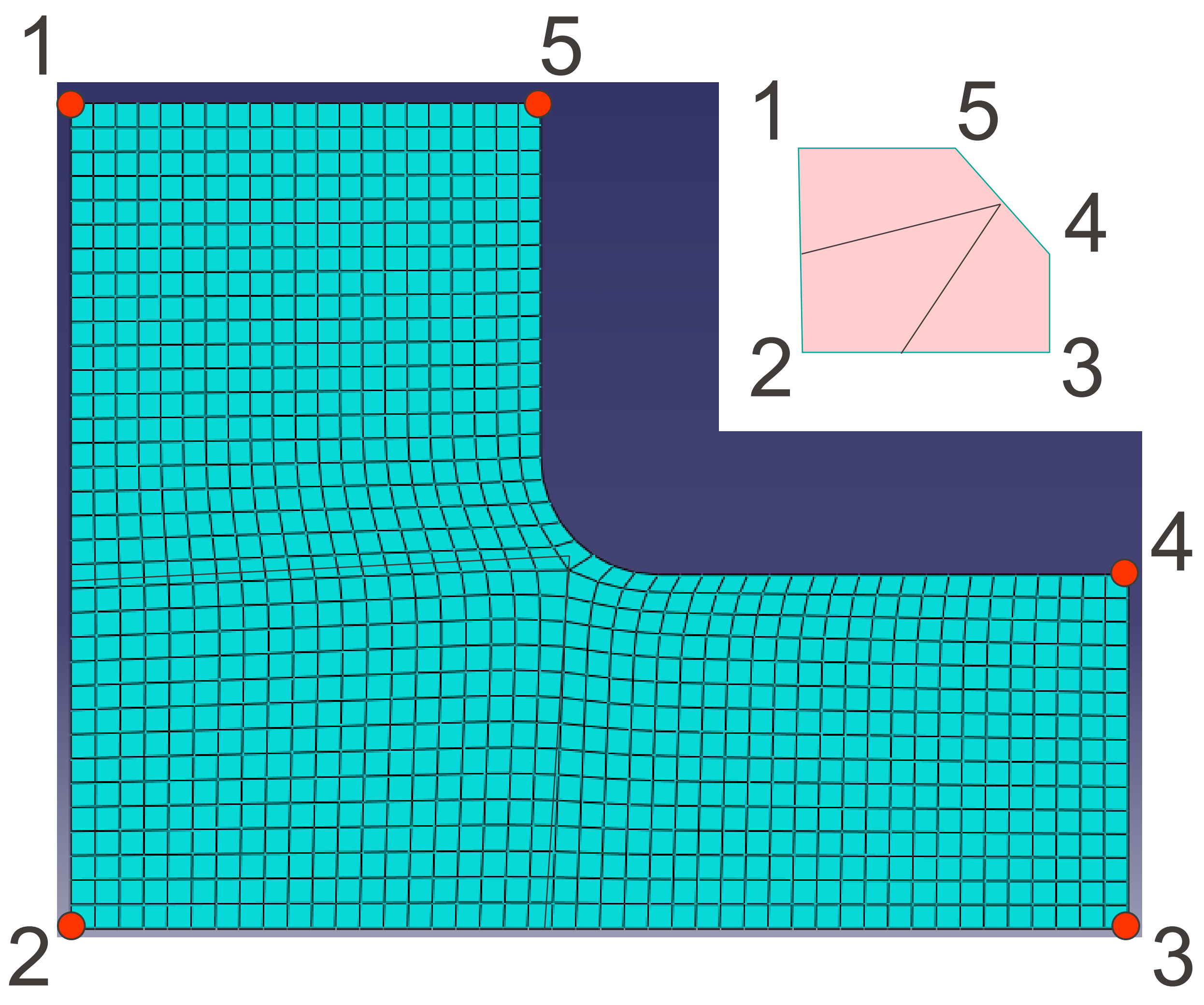
- 2D structured (quad) mesh: map/deform a simple polygon onto the actual surface.









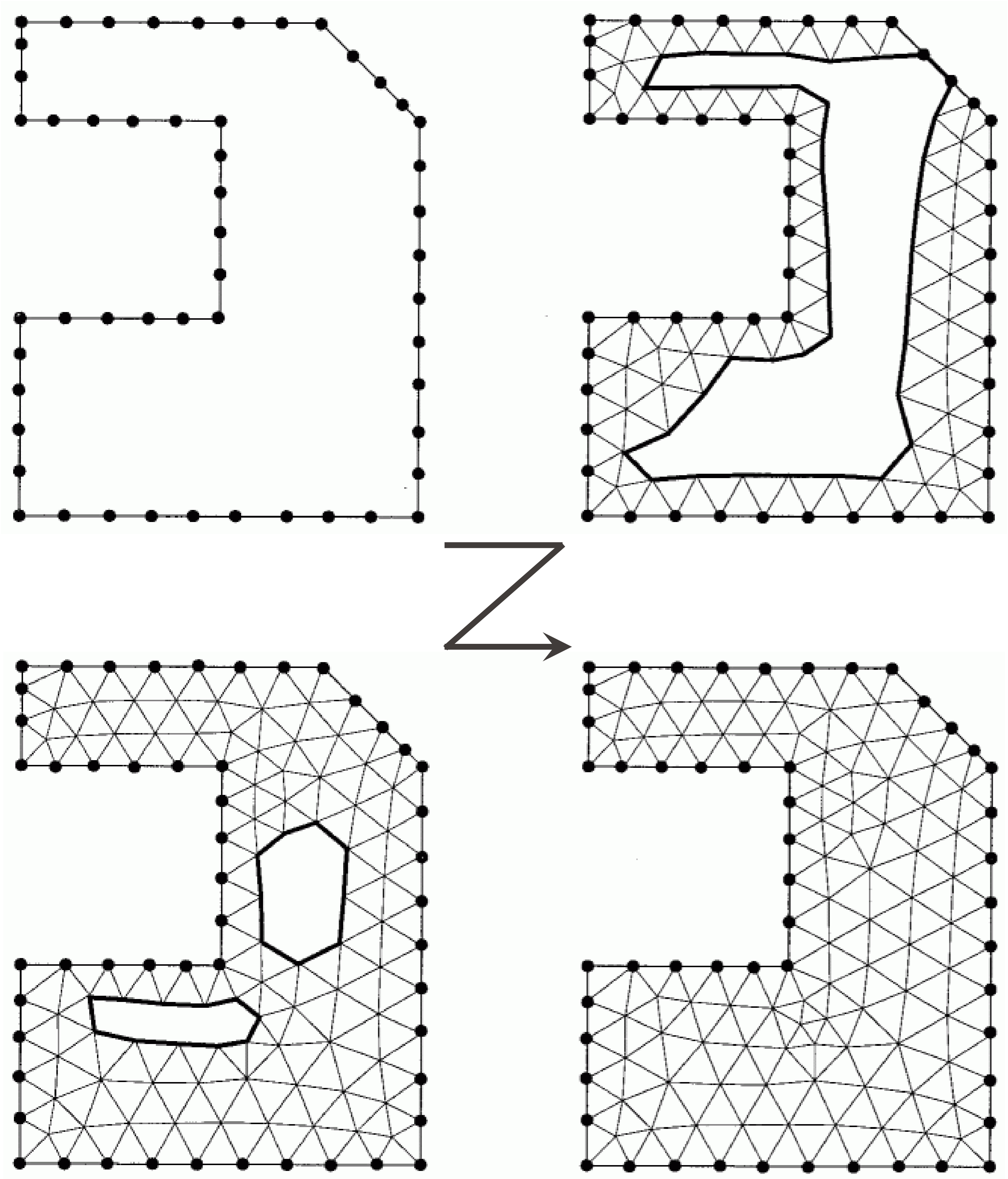


Numerical Flow Simulation

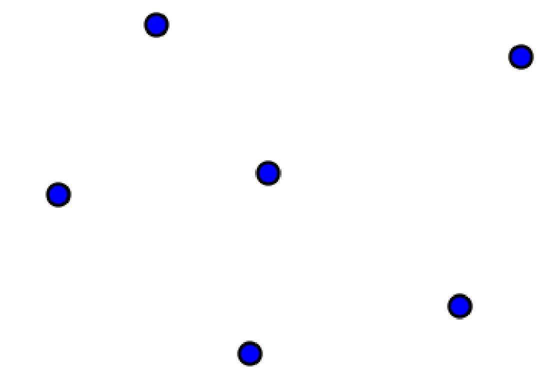
# Many different meshing algorithms

- 2D unstructured (tri) mesh:

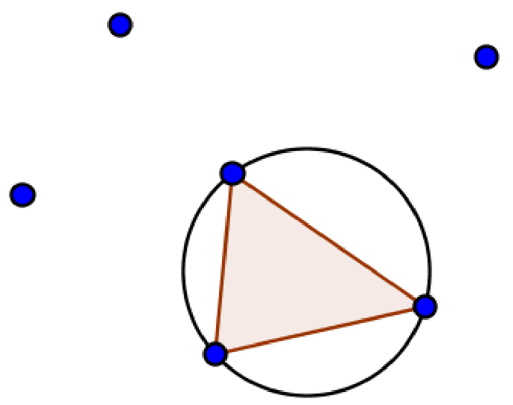
Advancing front



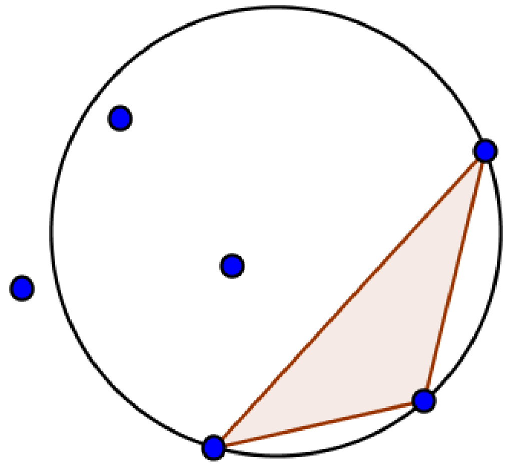
Delaunay



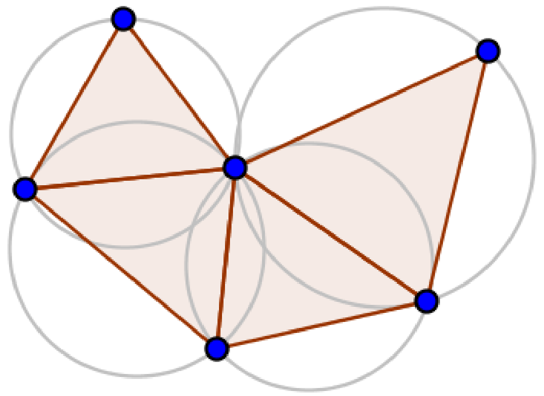
A collection of points



A Delaunay triangle



A non-Delaunay triangle

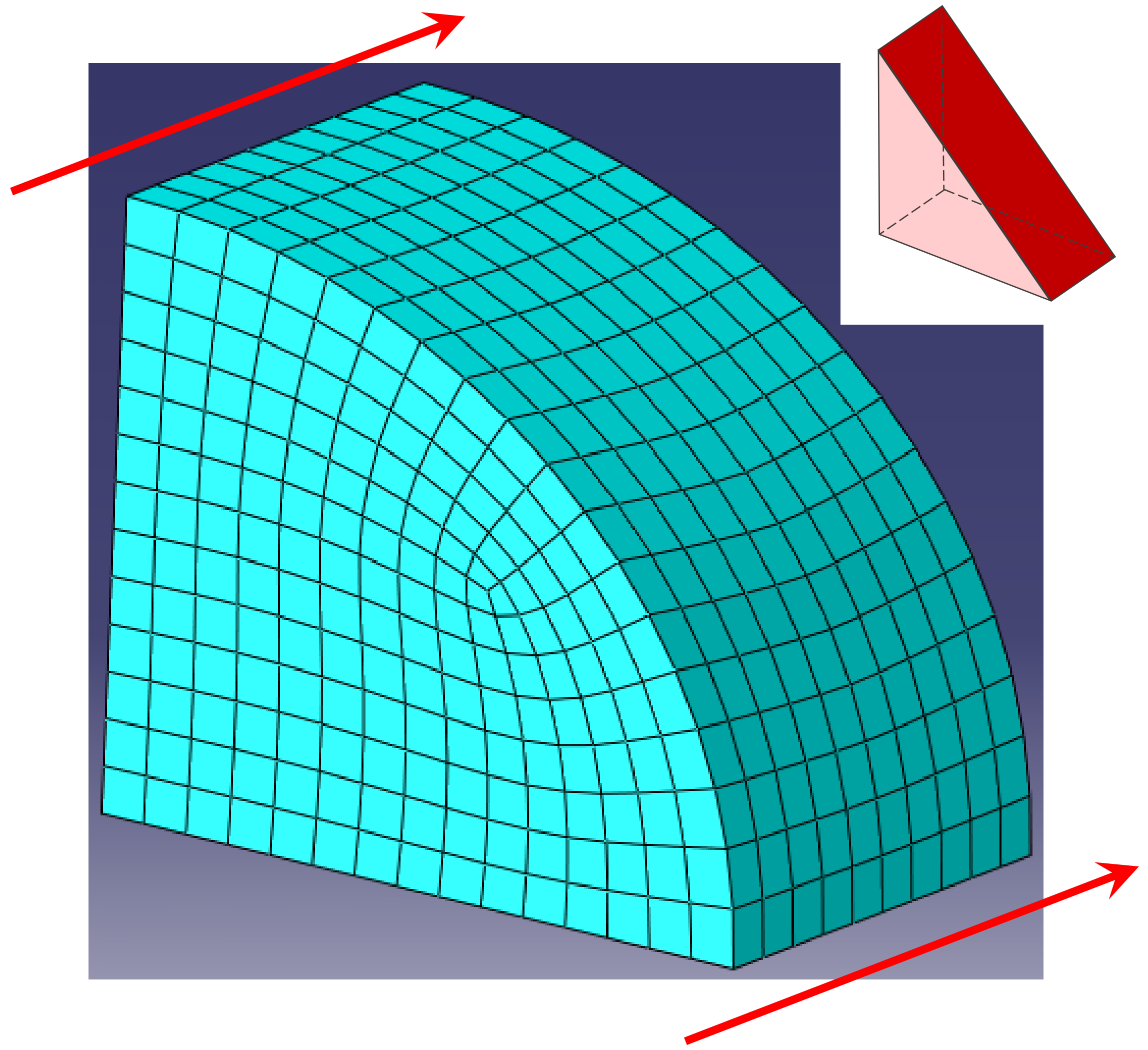


A Delaunay triangulation

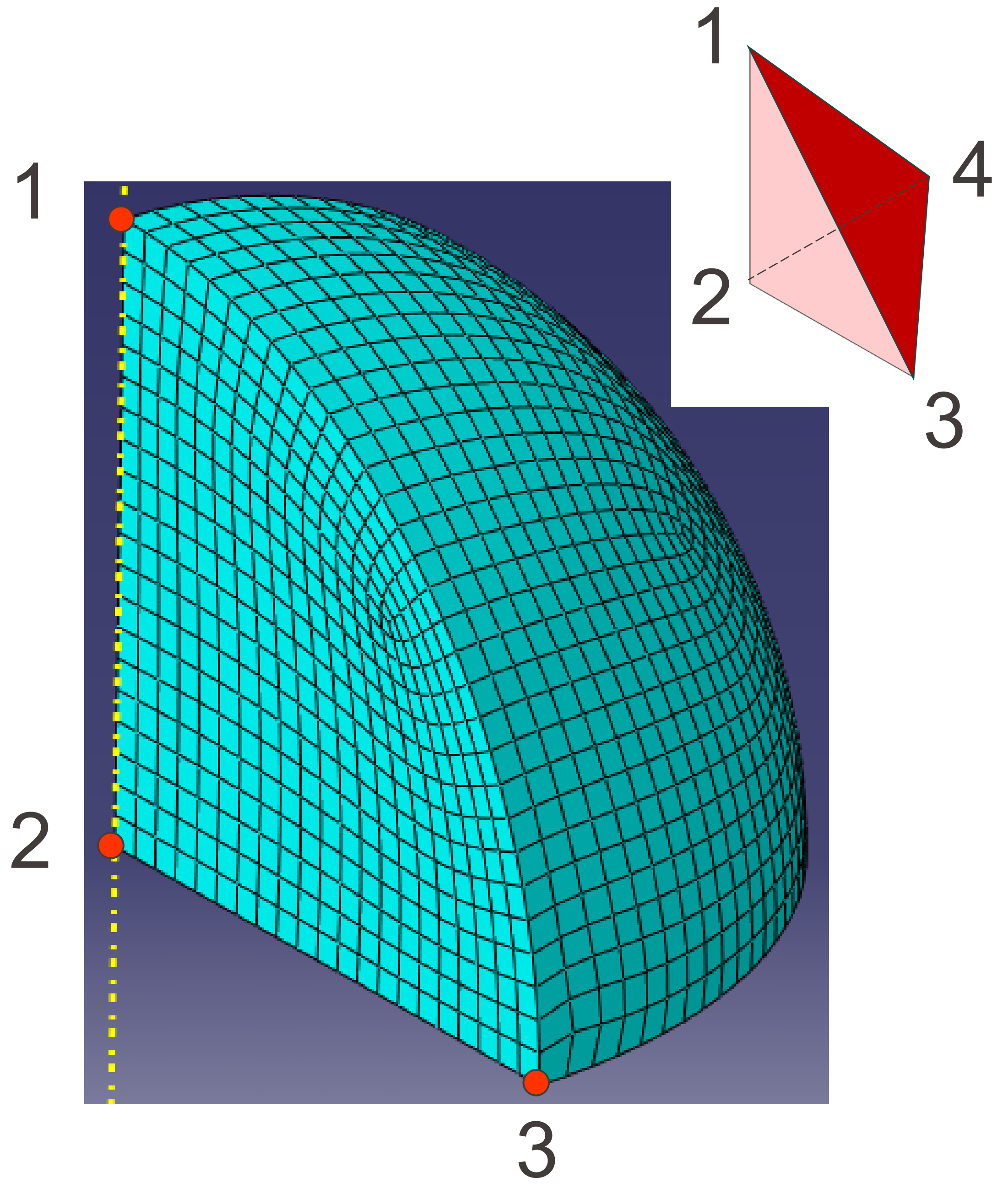
# Many different meshing algorithms

- 3D structured mesh (hex):

Extrude a structured 2D (quad) mesh.

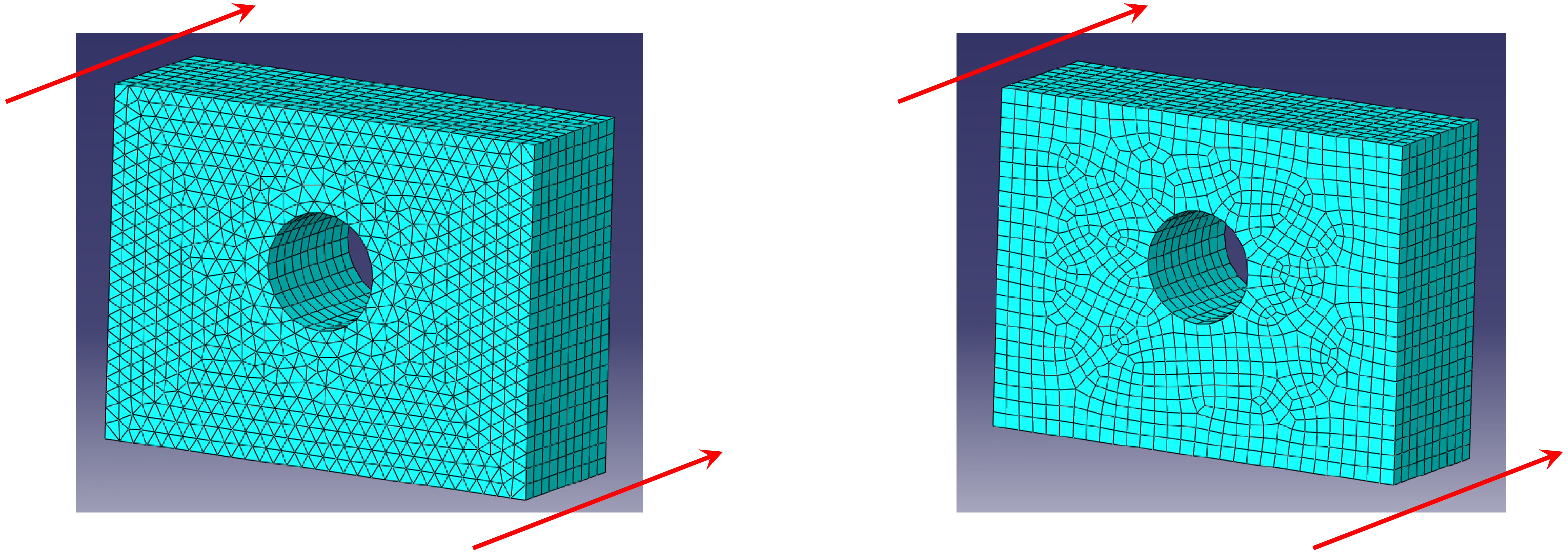


Map/deform a simple polyhedron onto the actual volume



# Many different meshing algorithms

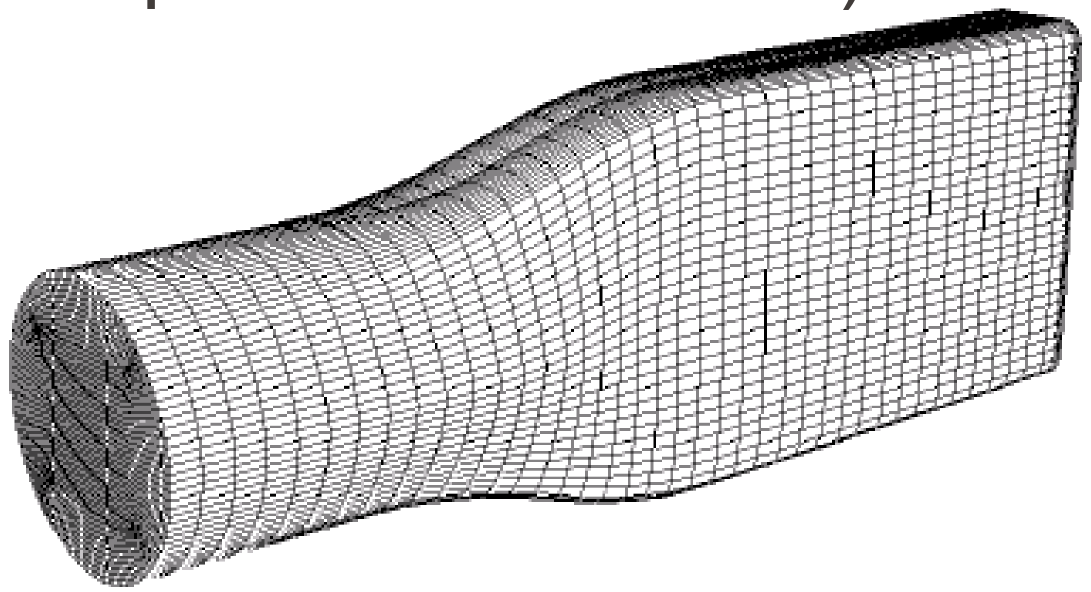
- 3D swept mesh: extrude unstructured 2D mesh (tri→prism / quad→hex)



- 3D unstructured mesh: advancing front / Delaunay
- Other algorithms: partitioning method, grid method, paving method…

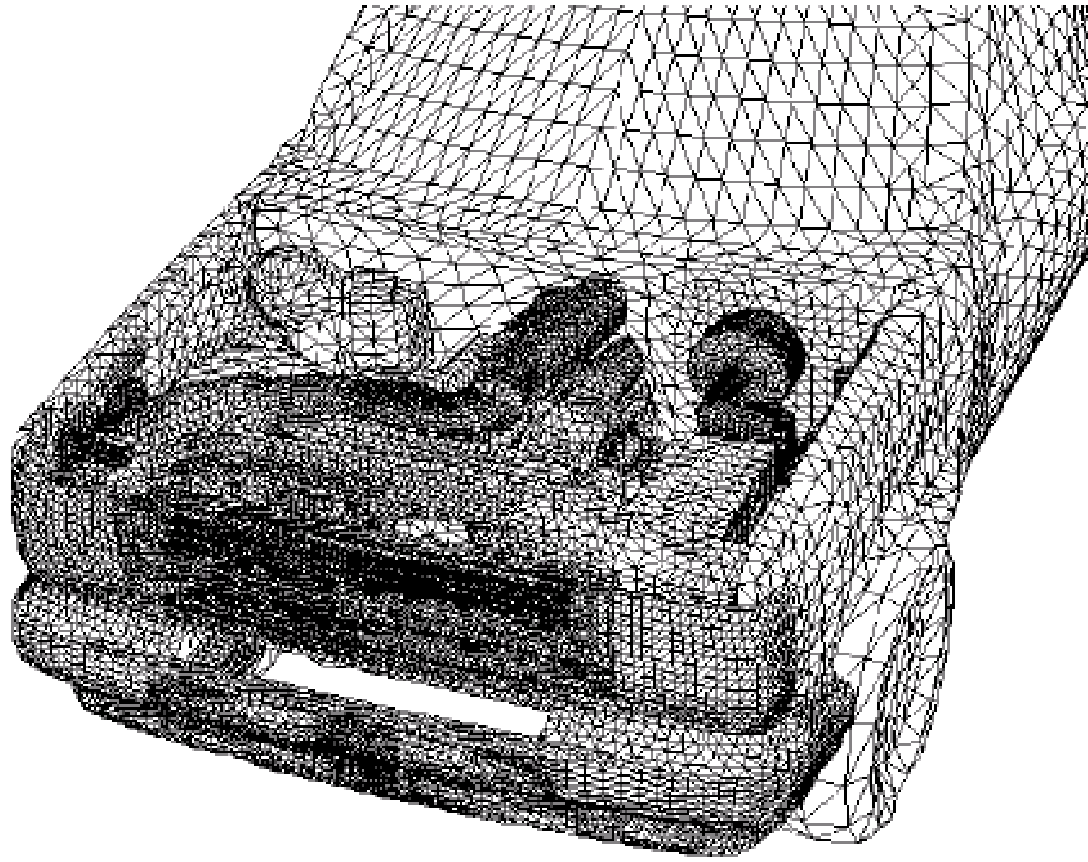
# Some guidelines: choice of mesh guided by...

- Geometry
  - Some elements better adapted to some simple geometries (e.g. quad / prism for channels).
  - Automatic unstructured meshing easier for complex geometries.



- Flow characteristics: large gradients require special attention.

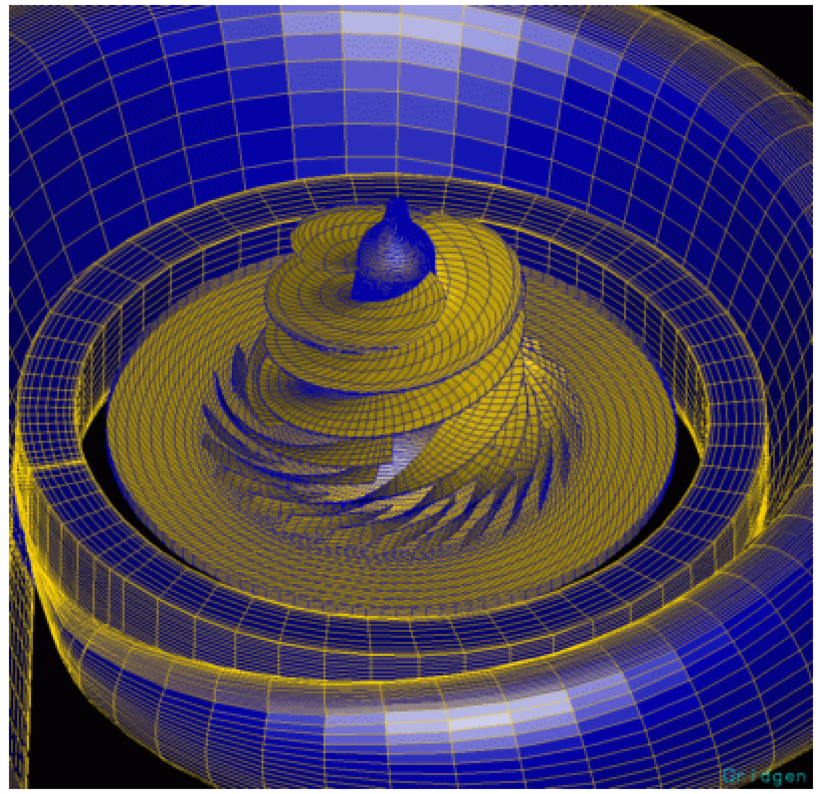
- Meshing time
  - Mesh generation extremely time consuming for complex geometries.
  - Automatic unstructured mesh generation faster.

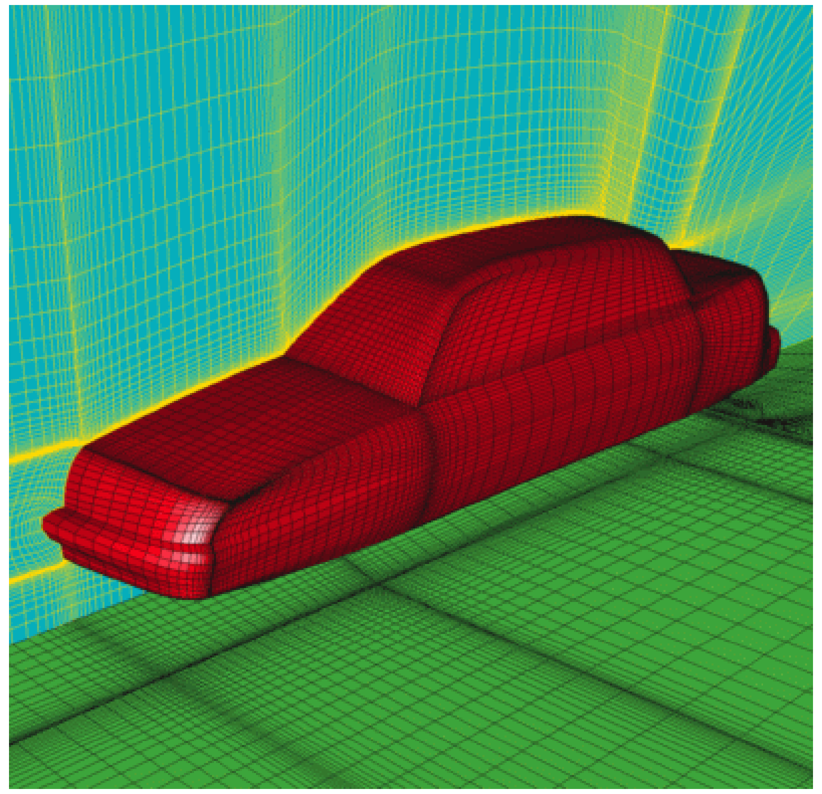


- Computational resources
  - Simple geometries / boundary layers: quad / hex may use fewer cells.
  - Complex geometries / different length scales: tri / tetra may be better.

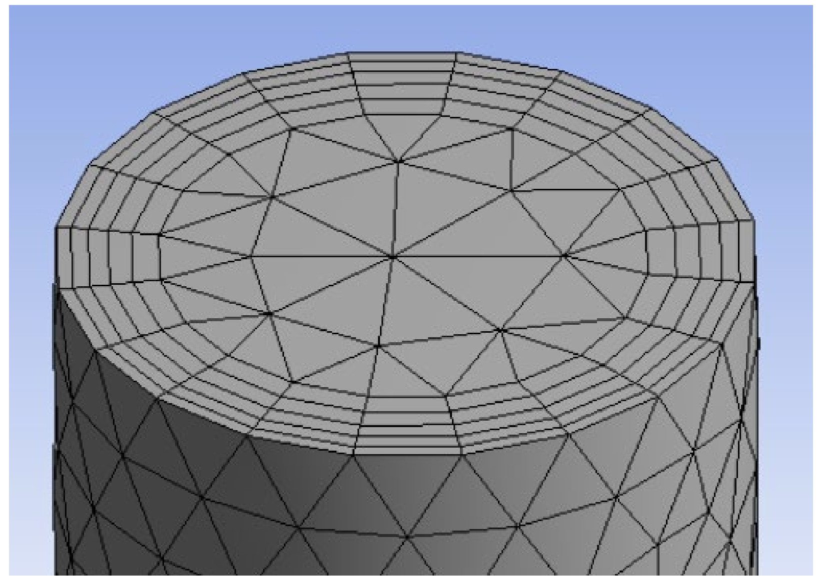
- Solution accuracy
  - Less numerical diffusion when flow aligned with the mesh.
  - For simple flows, quad / hex are preferred. For complex flows, no preferred element.

# Some guidelines for mesh generation





- Minimize mesh complexity
    - Use structured mesh when appropriate.
    - Use quad / hex elements when possible.
    - Use tri / tetra elements for complex geometries.
- Optimize number of mesh cells
    - Don’t use too many / too few elements. Refine where needed, coarsen where possible.
    - Use quad / hex when possible (e.g. boundary layers, long pipes).
- Maximize solution accuracy
    - Concentrate elements in critical regions (boundary layers, wakes, jets, shocks).
    - Align quad / hex elements with flow direction.
    - Avoid poor quality elements.
    - Maximize mesh continuity: rapid size variations should be avoided (e.g. edge of boundary layers). Sometimes poor transition with automatic mesh generation / adaption.



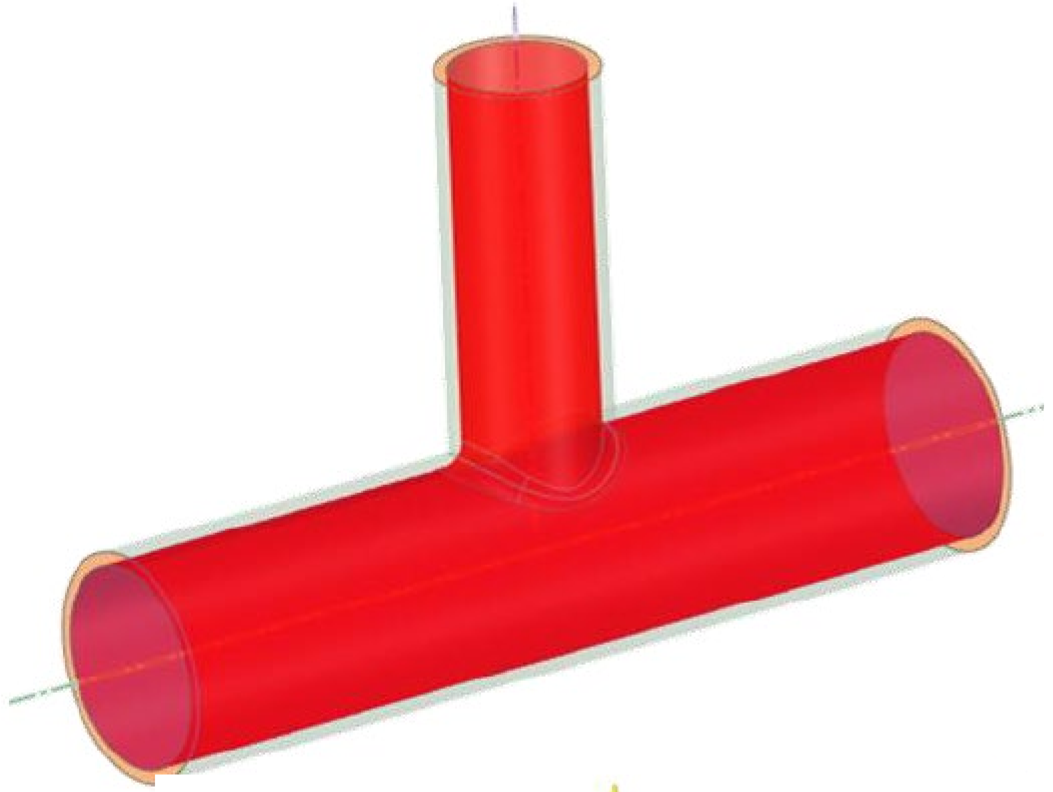
# Summary

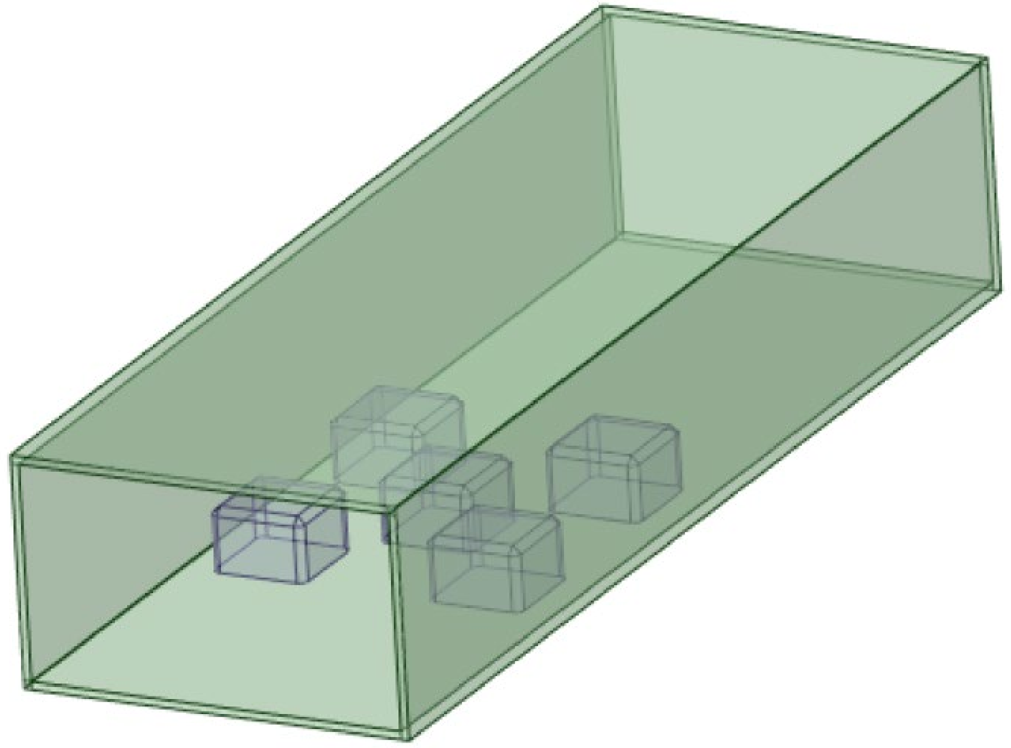
- Geometry modeling:
  - Define a geometry suitable for CFD before meshing.
  - Clean geometry of unwanted features.
  - Make sure the geometry is closed.

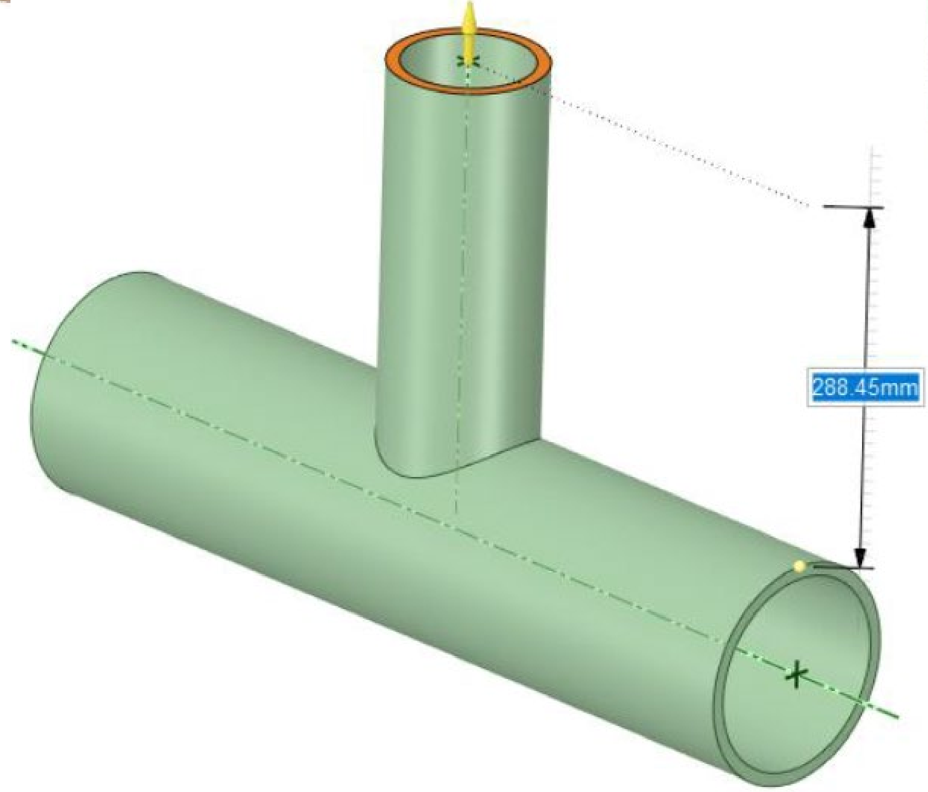
- Meshing:
  - Many different mesh elements and mesh types.
  - Trade-off between resources (setup time, computation time, memory) and accuracy.
  - Convergence and accuracy depend on mesh size / quality.
  - Strive for mesh quality. Refine in regions of large gradients.

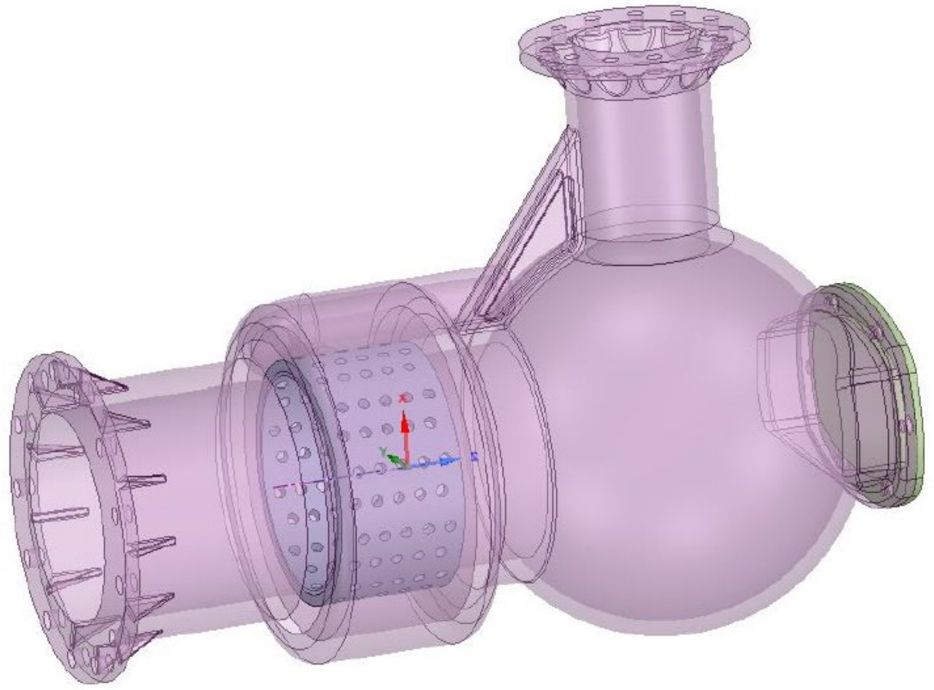
# Tutorials: geometry modeling with SpaceClaim

1. Extracting a fluid volume
2. Repairing an existing geometry; using a symmetry plane to split
3. Creating a geometry from scratch
4. Named selections; bodies of influence







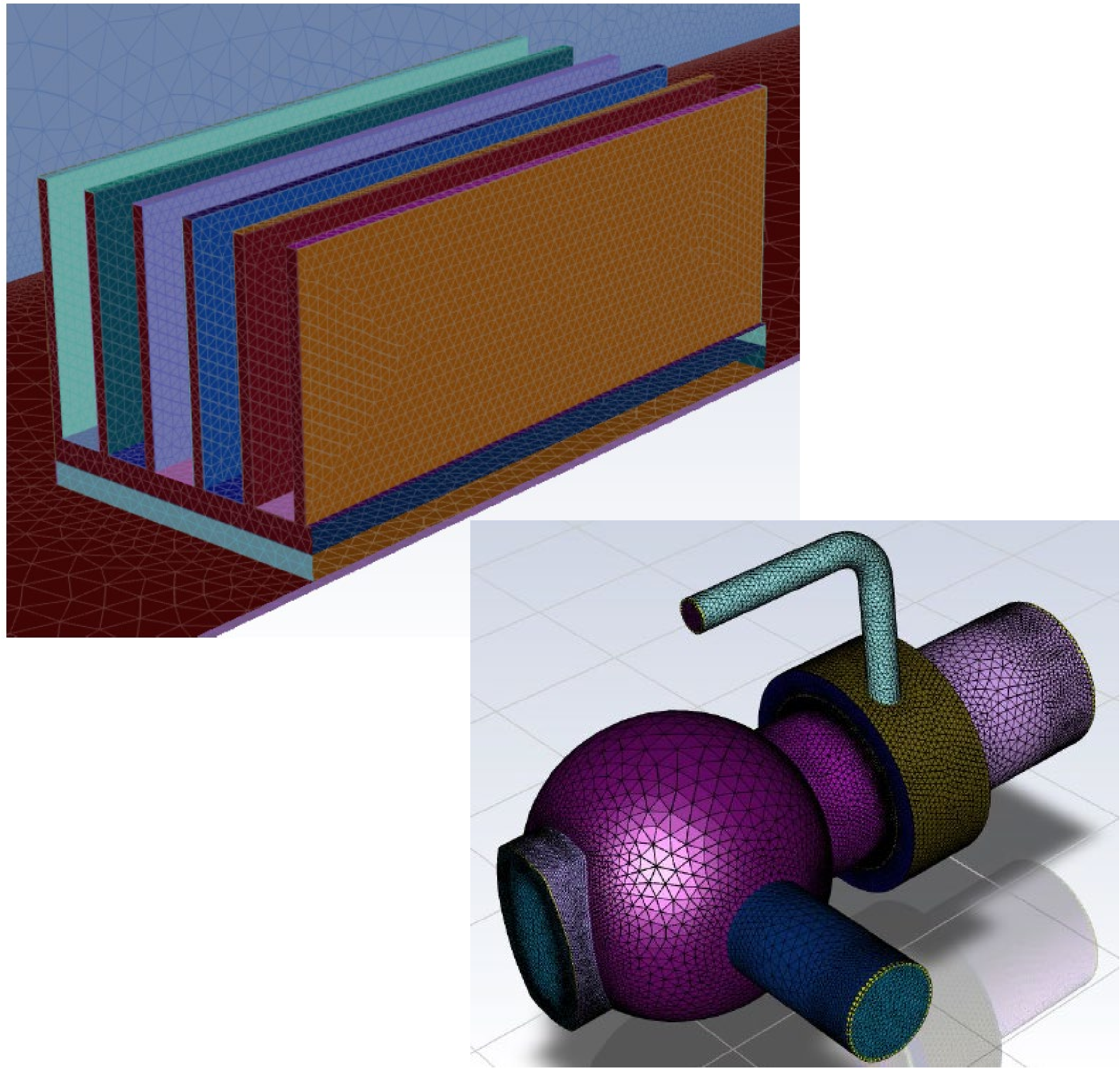


- Consider how you would perform these examples using alternative CAD software that you may be familiar with (e.g. Catia, Solidworks).

# Tutorials: meshing with Fluent meshing mode

1. Overview of the “Watertight Geometry” workflow (mesh generation of a heat sink)

2. Overview of the “Watertight Geometry” workflow (mesh generation of a mixer)



# Tutorials: meshing with Workbench Meshing

1. Meshing basics
   (mesh generation in a T-junction)

2. Meshing methods
   (review of different meshing methods)

3. Global mesh controls
   (sizing and inflation)

4. Local mesh controls
   (hybrid mesh on a multi-body part)