Basics of computational fluid dynamics

EGME 520 – Fall 2018

Salvador Mayoral, Ph.D.

Department of Mechanical Engineering
California State University, Fullerton

November 29, 2018

Outline

- Finite element analysis vs. computational fluid dynamics
- Motivation for computational mechanics
- Meshing – Domain discretization
- Grid dependency

Finite element analysis vs. computational fluid dynamics

What are they?

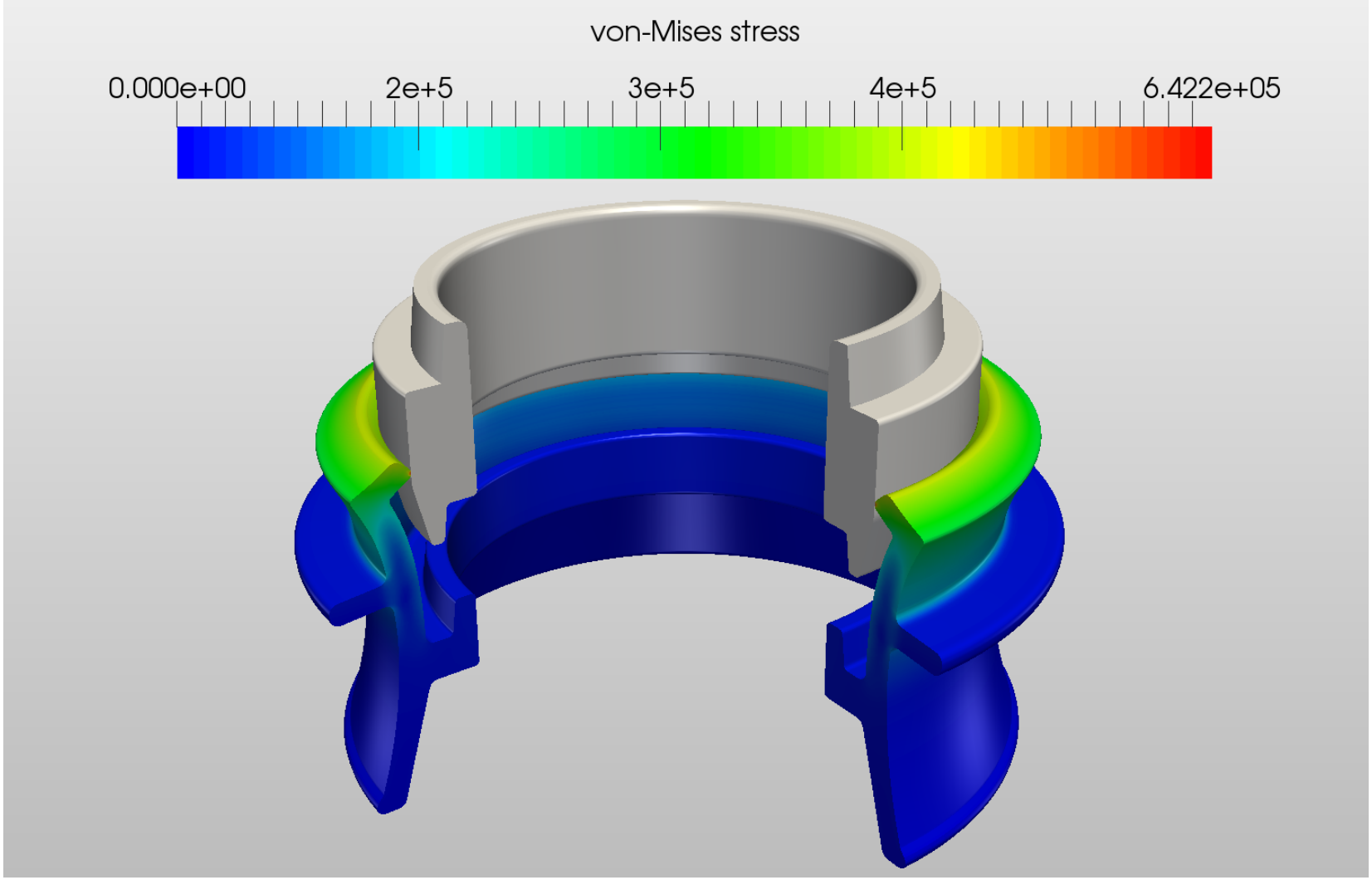
Typically – FEA is for structural applications and CFD for fluid dynamics applications... this is not completely accurate.

Finite element analysis (FEA) or finite element method (FEM)
A *specific numerical technique* that solves a continuous problem stated in the form of a PDE

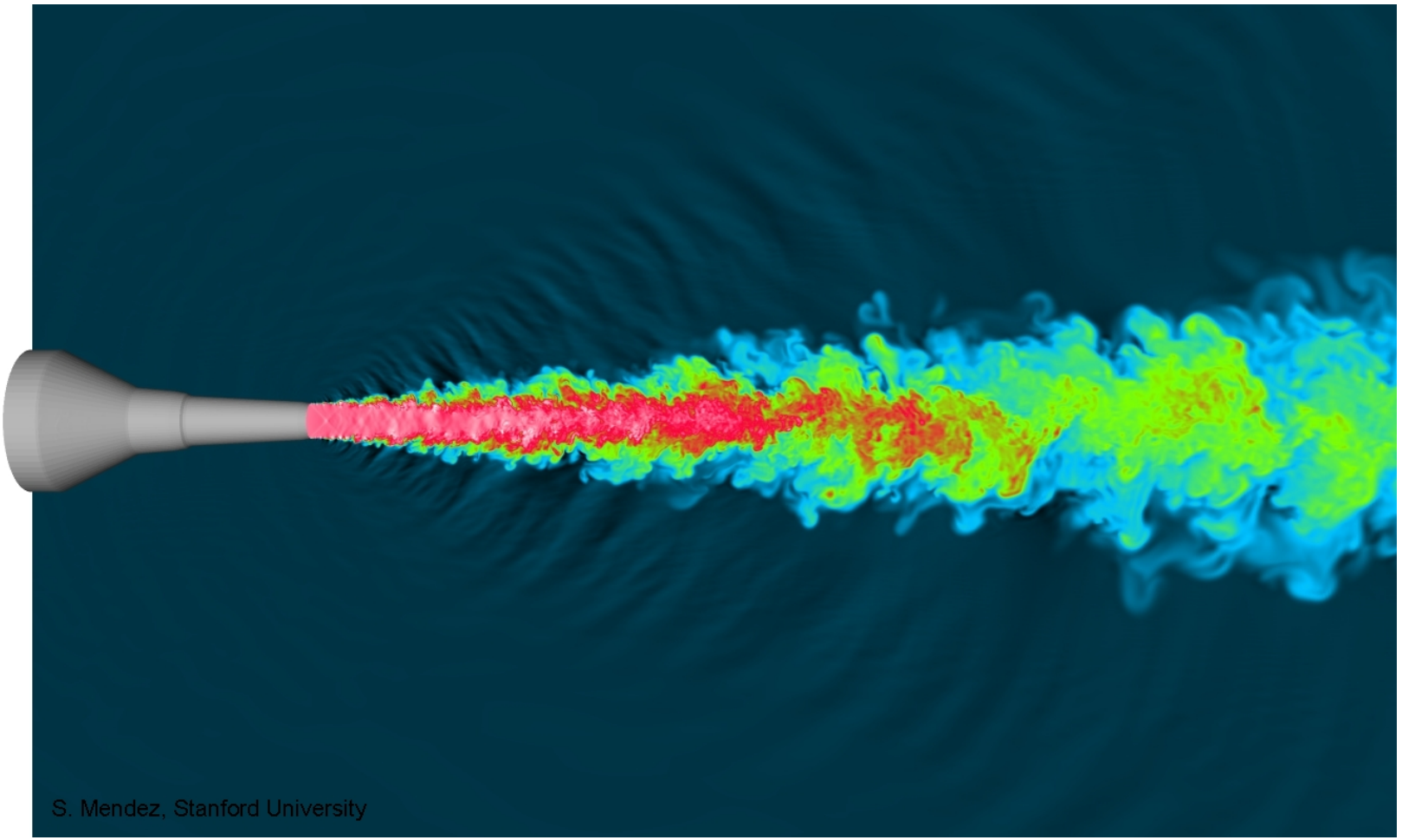
Computational fluid dynamics (CFD)
The use of the numerical techniques to solve fluid dynamics problems.

- Utilizes several approaches such as Finite Difference Methods, Finite Element Methods, Finite Volume Methods, and so on.
- Most CFD solutions utilize finite volume methods, some use FEM
- Formulations
  - Reynolds Averaged Navier-Stokes, RANS
  - Large Eddy Simulation, LES
  - Direct Numerical Simulation, DNS
- RANS is the least accurate but most commonly used.
  - $k-\epsilon$, $k-\omega$, shear-stress transport (SST), and Spalart-Allmaras

FEM simulation done on Rubber seal



LES simulation of a jet



CFD - Colorful fluid dynamics!

Motivation for computational mechanics

Computational mechanics, simulations

Implementation of numerical analysis to solve problems in mechanics for which a quantitative theory already exists.

- *Mechanics* – solid mechanics, fluid dynamics, heat transfer, electromagnetics, vibrations, and acoustics.
- *Mathematics* – partial differential equations, linear algebra and numerical methods
  - Finite element
  - Finite difference
  - Finite volume
  - Boundary element
- *Computer science* – programming, algorithms, and parallel computing
  - Fortran
  - C++, recently has gotten more popular, *e.g.* OpenFOAM
  - MATLAB
  - Python

Experiments vs. simulations

Simulations gives an insight into flow patterns that are difficult, expensive or impossible to study using experimental techniques

Experiments	**Simulations**
Quantitative *description* of phenomena using measurements	Quantitative *prediction* of phenomena using numerical methods
One quantity at a time	All desired quantities
Limited number of points and time instants	High resolution in space and time
Laboratory-scale model	Actual scale
Limited range of operating conditions	Virtually any realistic operating conditions

Note: Simulations do not completely replace experimentation, they reduce the amount of experimentation and overall cost.

Experiments vs. simulations

Experiments	**Simulations**
Expensive	Cheap(er)
Slow	Fast(er)
Sequential	Parallel
Single-purpose	Multiple-purpose
Equipment and personnel are difficult to transport	Simulation software is portable, easy to use and modify

Results from simulations are never 100% reliable

- Input data may involve too much guessing or imprecision
- Mathematical model of the problem at hand may be inadequate
- Accuracy of the results is limited by the available computing power
- Error sources: modeling, discretization, iteration, implementation

Simulation process

1. Problem statement information about the flow
2. Mathematical model, governing equations with initial and boundary conditions
3. Mesh generation nodes/cells, time instants
4. Space discretization coupled ODE/DAE systems
5. Time discretization algebraic system $\mathbf{Ax} = \mathbf{b}$
6. Iterative solver discrete function values
7. Simulation runs parameters, stopping criteria
8. Post-processing visualization, analysis of data
9. Verification model validation / adjustment

Mathematical model

1. Identify the acting forces
2. Choose a suitable flow model (laminar, inviscid, compressible, etc.) and reference frame.
3. Define the *computational domain* in which to solve the problem.
   - For solid mechanics it is the body of interest
   - For fluids, it is the flow field around the body of interest
4. Formulate the governing equations
5. Simplify the governing equations to reduce the computational effort:
   - Symmetries and predominant directions (1D/2D)
   - Neglect the terms which have little or no influence on the results
   - Add constituitive relations and specify initial/boundary conditions.

Discretization process

Governing equations are transformed into a set of algebraic equations

1. Mesh generation – computational domain is decomposed into a finite set of cells/elements
   - Structured or unstructured, meshes
   - CAD tools (geometry) + grid generators (mesh)
   - Mesh size number of elements or nodes
2. Space discretization (approximation of spatial derivatives)
   - Finite differences/volumes/elements
   - High- vs. low-order approximations
3. Time discretization (approximation of temporal derivatives)
   - Explicit vs. implicit schemes, stability constraints
   - Local time-stepping, adaptive time step control

Iterative solution strategy

The coupled nonlinear algebraic equations are solved iteratively

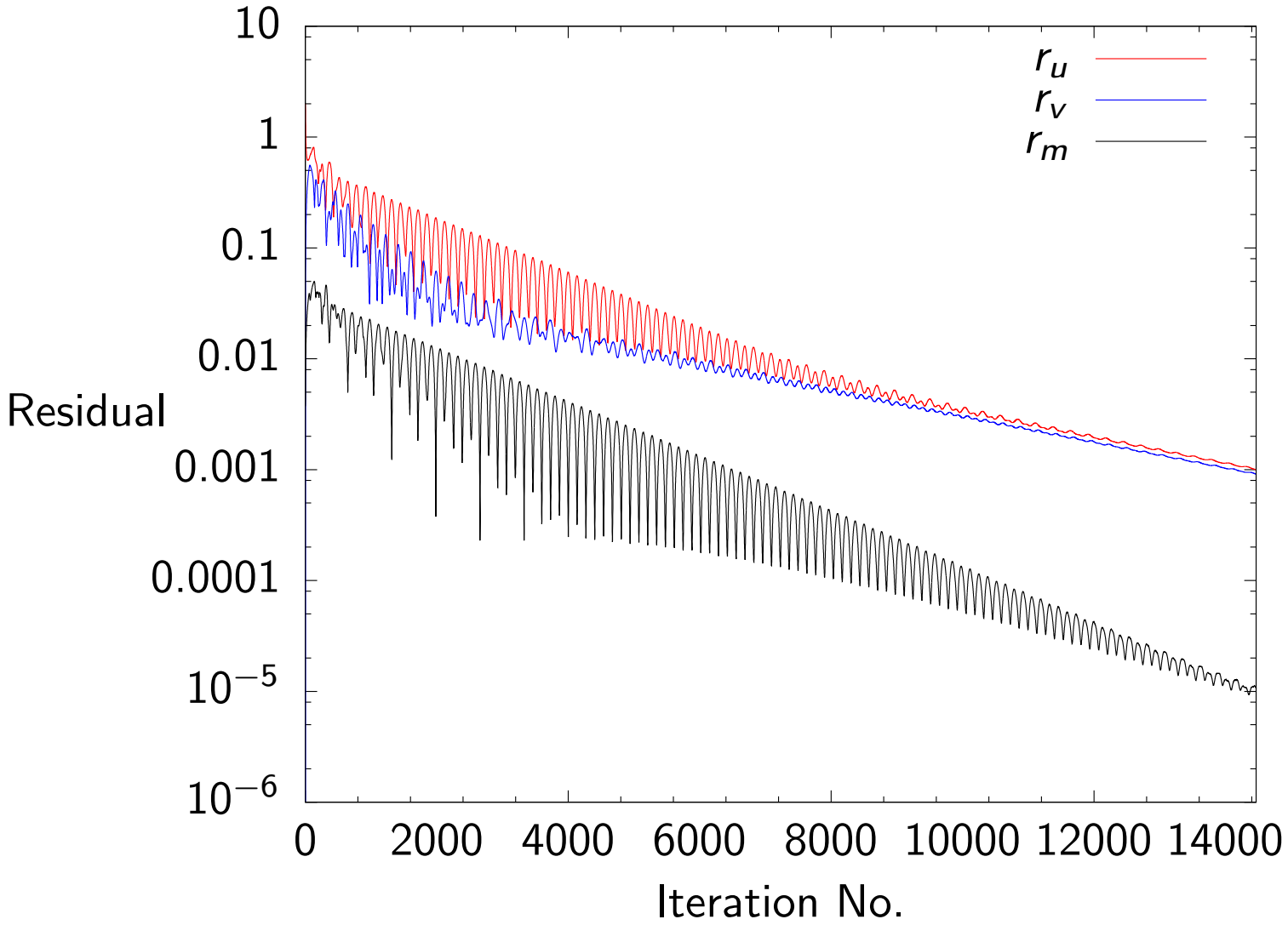
$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

- Judicious guess and check
- *Residual* - difference in value of a quantity between two iterations.
  - The residual will never be exactly zero.
  - The lower the residual, the more numerically accurate the solution.
  - The lower the residual, the less the results will change

- *Convergence criteria:* the point where we deem the solution *converged* is defined by the judgment of the user.
  - Residual Values are less than a predetermined value
  - Solution imbalances – solution satisfies the governing equation within 1% error.
  - Quantities of interest – values from post processing don't change.

Note: The larger the number of nodes the larger the matrix **A**, more time

Example of residual plots



Meshing – Domain discretization

Meshing

Body is divided into an equivalent system of many smaller units

- Domain is discretized into a finite set of control volumes or elements.
- The discretized domain is called the *grid* or mesh.
- Governing equations are discretized into algebraic equations.
- Solution is obtained at discreet points (*nodes*), solution is interpolated in between nodes

The grid has a significant impact on:

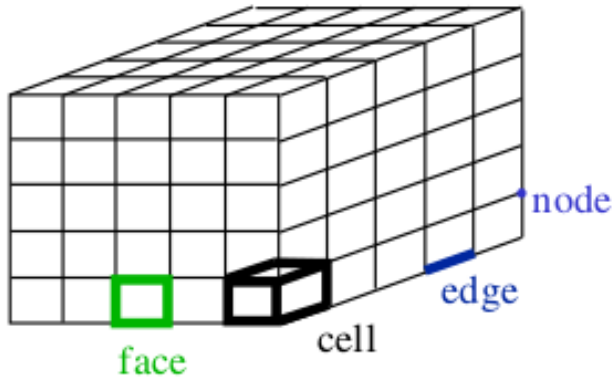
- Rate of convergence (or even lack of convergence).
- Solution accuracy.
- CPU time required.

Importance of mesh quality for good solutions

- Grid density
- Skewness
- Tetrahedral vs. hexagonal
- Boundary layer mesh for turbulence modeling

Meshing terminology

- *Cell* – control volume into which domain is broken up.
- *Node* – grid point.
- *Cell center* – center of a cell.
- *Edge* – boundary of a face.
- *Face* – boundary of a cell.
- *Zone* – grouping of nodes, faces, and cells
  - Wall boundary zone.
  - Fluid cell zone.
- *Domain* – group of node, face and cell zones.



Typical cell shapes

Many different cell/element and grid types are available.

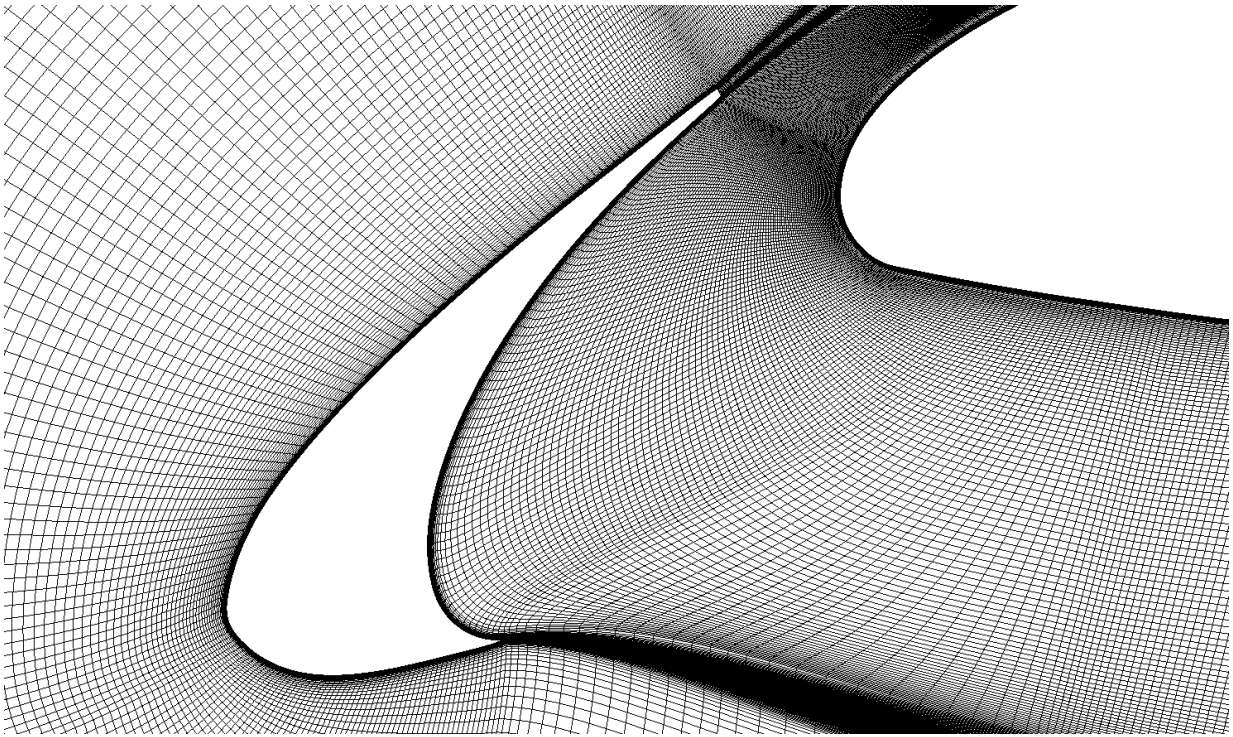
- Choice depends on the problem and the solver capabilities.
- What degree of grid resolution is required in each region of the domain?
- Will you use adaption to add resolution?
- Do you have sufficient computer memory?

Quadrilateral (2D) or hexahedron (3D)

Structured grid - quadrilateral or hexahedron

Simple geometries, provide high-quality solutions with fewer cells

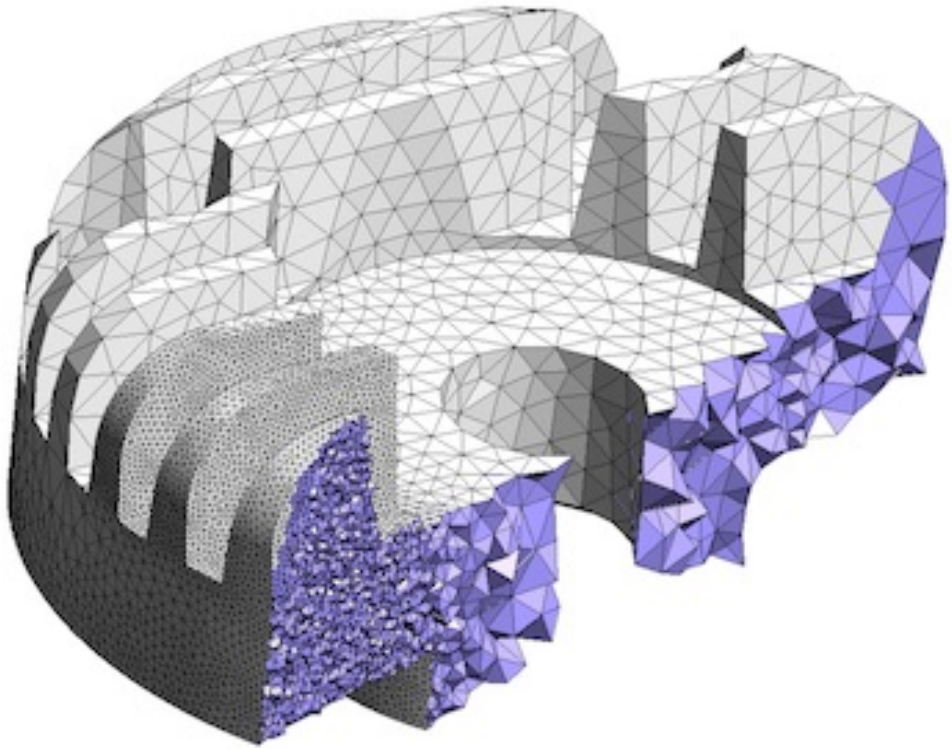
- Single-block, structured grid.
- i, j, k indexing to locate neighboring cells.
- Grid lines must pass all through domain.
- Can't be used for very complicated geometries.



Unstructured grid - triangle or tetrahedron

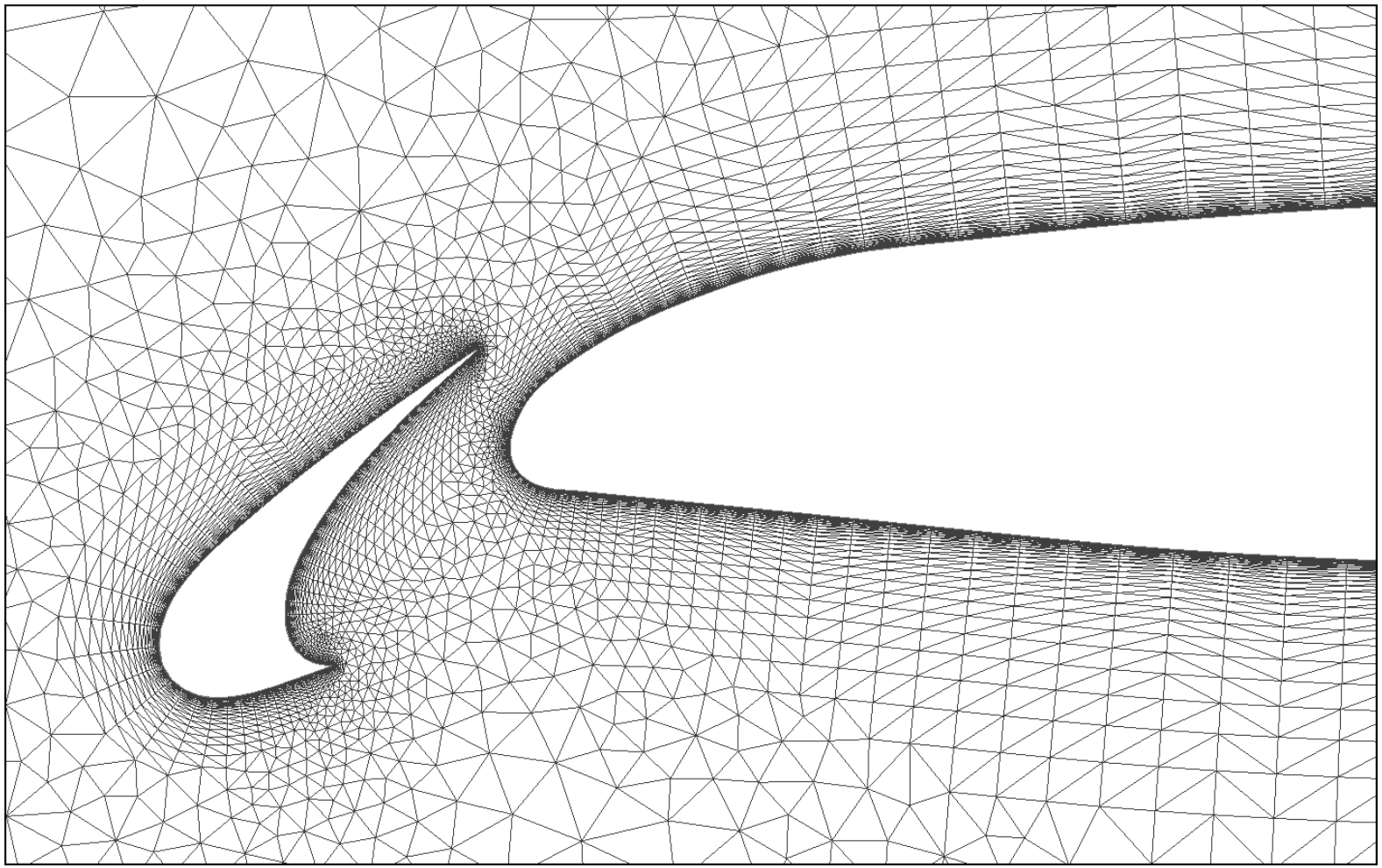
Complex geometries, meshes show no numerical advantage, and you can save meshing effort.

- The cells are arranged in an arbitrary fashion.
- No i, j, k grid index and constraints on cell layout.
- Requires some memory and CPU overhead for referencing.



Hybrid meshes

Use the most appropriate cell type in any combination.



Mesh quality

A poor quality grid will cause inaccurate solutions and/or slow convergence.

1. Skewness
2. Smoothness, change in size
3. Aspect ratio

Some general observations

- The mesh density should be high enough to capture all relevant flow features.
- For the same cell count, hexahedral meshes will give more accurate solutions, especially if the grid lines are aligned with the flow.
- The mesh adjacent to the wall should be fine enough to resolve the boundary layer flow
- In boundary layers, quadrilateral and hexahedron cells are preferred

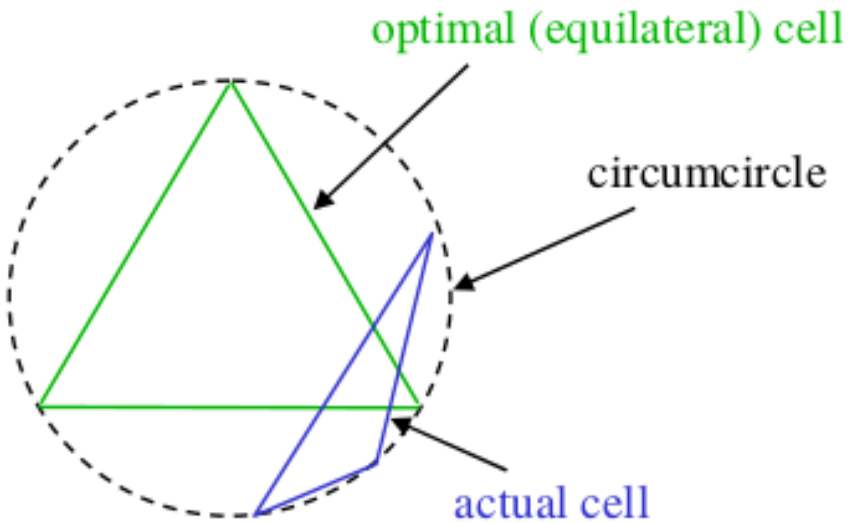
Skewness

Two methods for determining skewness:

1. Equilateral volume

$$\text{Skewness} = \frac{\text{optimal cell size} - \text{cell size}}{\text{optimal cell size}}$$

- Applies only to triangles and tetrahedrons.
- Default method for triangles and tetrahedrons



Skewness

Two methods for determining skewness:

2. Deviation from a normalized equilateral angle:

$$\text{Skewness} = \max\left(\frac{\theta_{\text{max}} - 90^\circ}{90^\circ}, \frac{90^\circ - \theta_{\text{min}}}{90^\circ}\right)$$

- Applies to all cell and face shapes.
- Always used for prisms and pyramids.

Equiangle skewness

Common measure of quality is based on equiangle skew.

$$\text{Equiangle skewness} = \max\left(\frac{\theta_{\max} - \theta_e}{180^\circ - \theta_e}, \frac{\theta_e - \theta_{\min}}{180^\circ - \theta_e}\right)$$

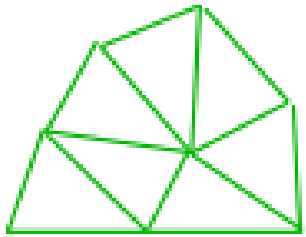
- $\theta_{\max}$ – largest angle in face or cell.
- $\theta_{\min}$ – smallest angle in face or cell.
- θ_e – angle for equiangular face or cell *e.g.*, 60° for triangle, 90° for square.

Minimize equiangle skew

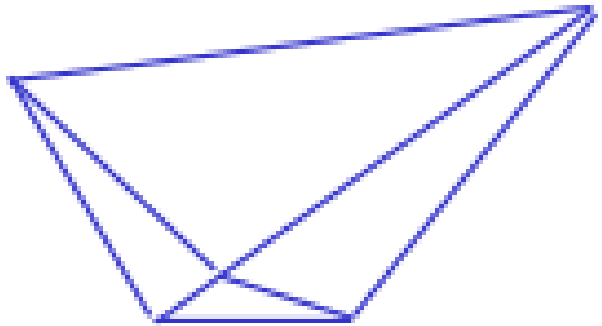
- Hex and quad cells: skewness should not exceed 0.85
- Triangles: skewness should not exceed 0.85
- Tetrahedrons: skewness should not exceed 0.9

Smoothness

Change in size should be gradual (smooth).

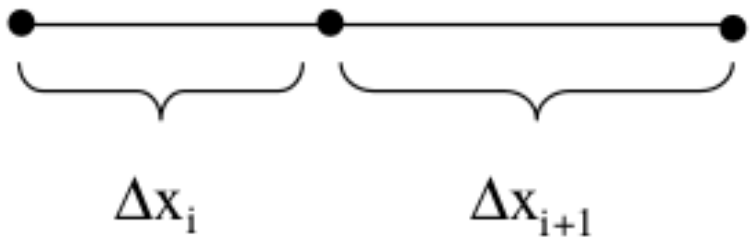


Smooth change in cell size



Large jump in cell size

Ideally, the maximum change in grid spacing should be $< 20\%$

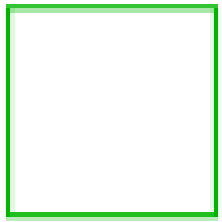


such that

$$\frac{\Delta x_{i+1}}{\Delta x_i} \leq 1.2$$

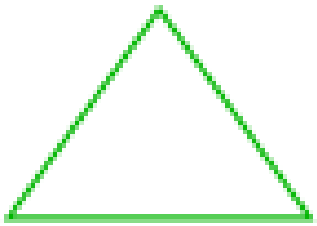
Aspect ratio

Ratio of longest edge length to shortest edge length.

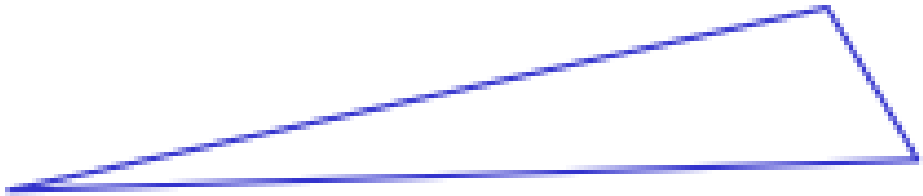


aspect ratio = 1

high-aspect-ratio quadrilateral



aspect ratio = 1



high-aspect-ratio triangle

Aspect ratio equal to 1 is ideal.

Grid design guidelines: total cell count

Sources of error

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

More cells can give higher accuracy.

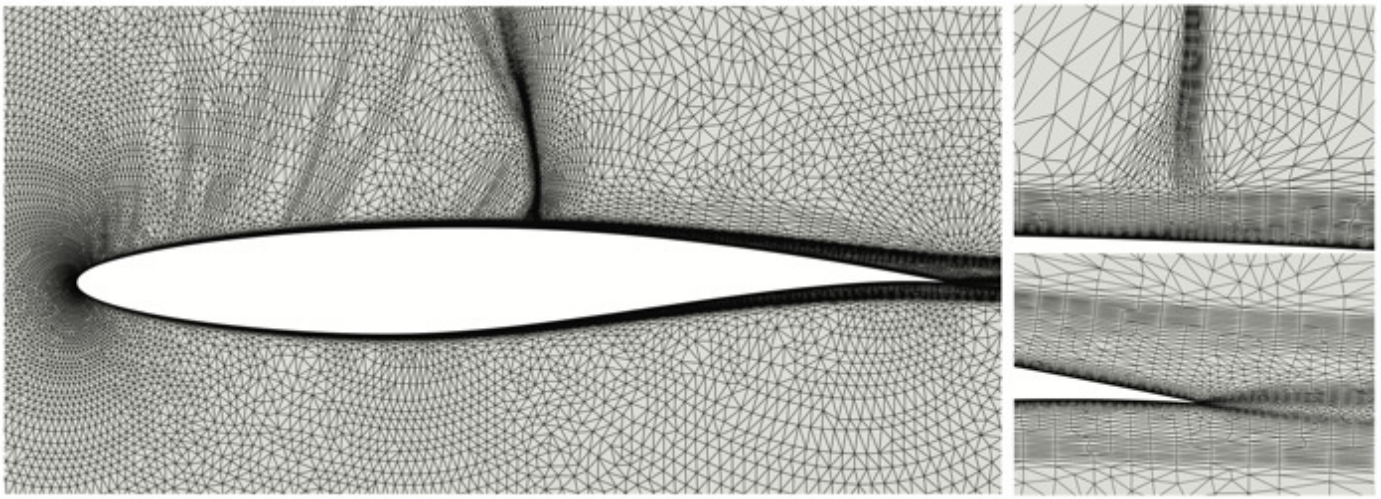
- Increased memory and CPU time.
- To keep cell count down:
  - Use a non-uniform grid to cluster cells only where they are needed.
  - Use solution adaption to further refine only selected areas.

Mesh adaption

- Add more cells where needed to resolve the flow field.
- Gradients of flow or user-defined variables.
- All cells on a boundary.

Mesh adaptation

Mesh for transonic flow over an airfoil

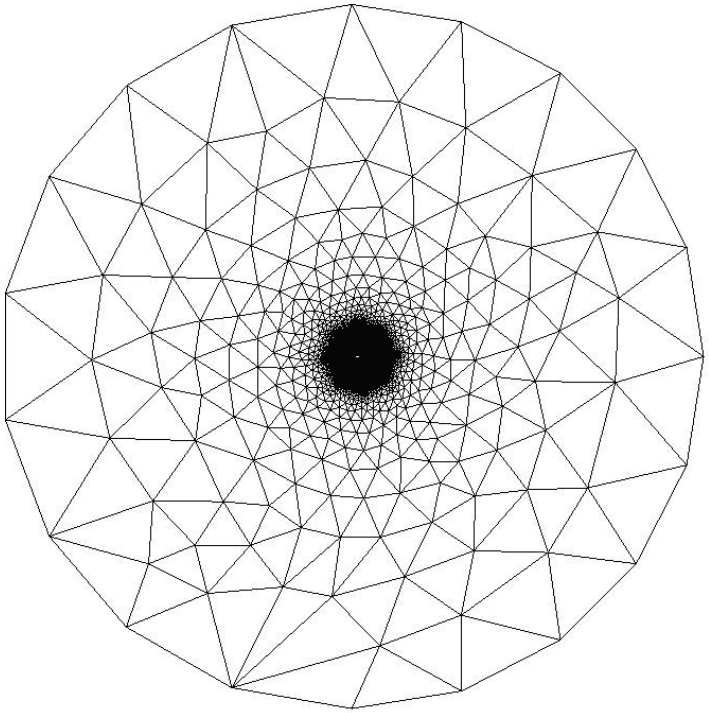


- Grip points are concentrated near the body for the boundary layer
- Points are concentrated near the shock location
- *Idea* - Increase the grid density were variable gradients are high

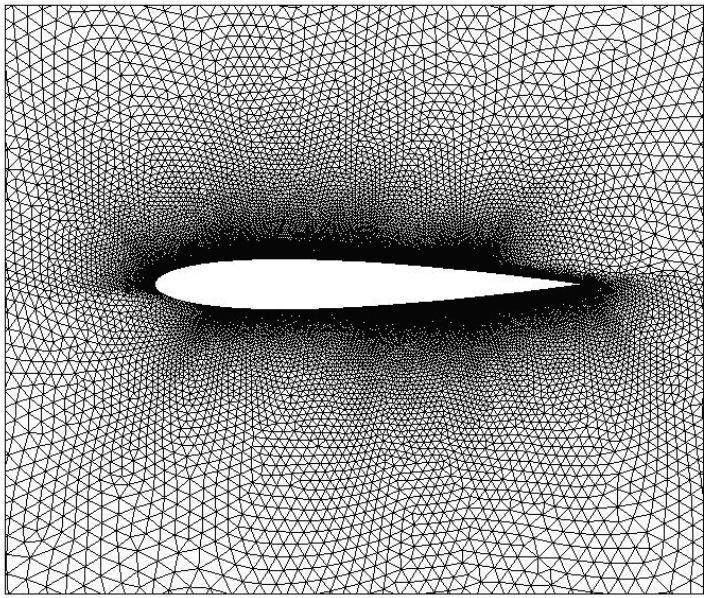
$$u \simeq \frac{\Delta x}{\Delta t}$$

Defining the flow domain for external flows

Flow domain over an airfoil



Full domain



Mesh near the airfoil

Flow domain is typically extended to a radial distance approximately 10 times the largest length scale.

Grid dependence

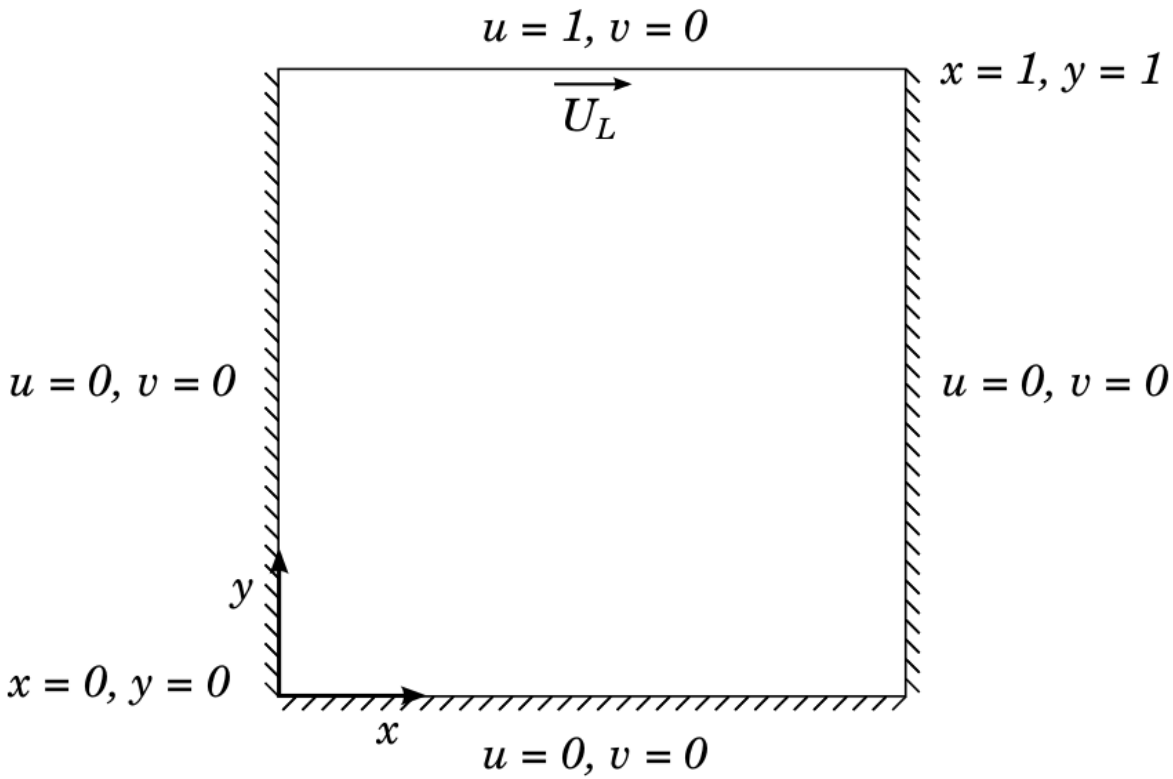
Grid dependence

Just because a simulation has converged, *e.g.* residual values are roughly $10^{-4} - 10^{-5}$, does not mean the solution is correct!

Grid independent study

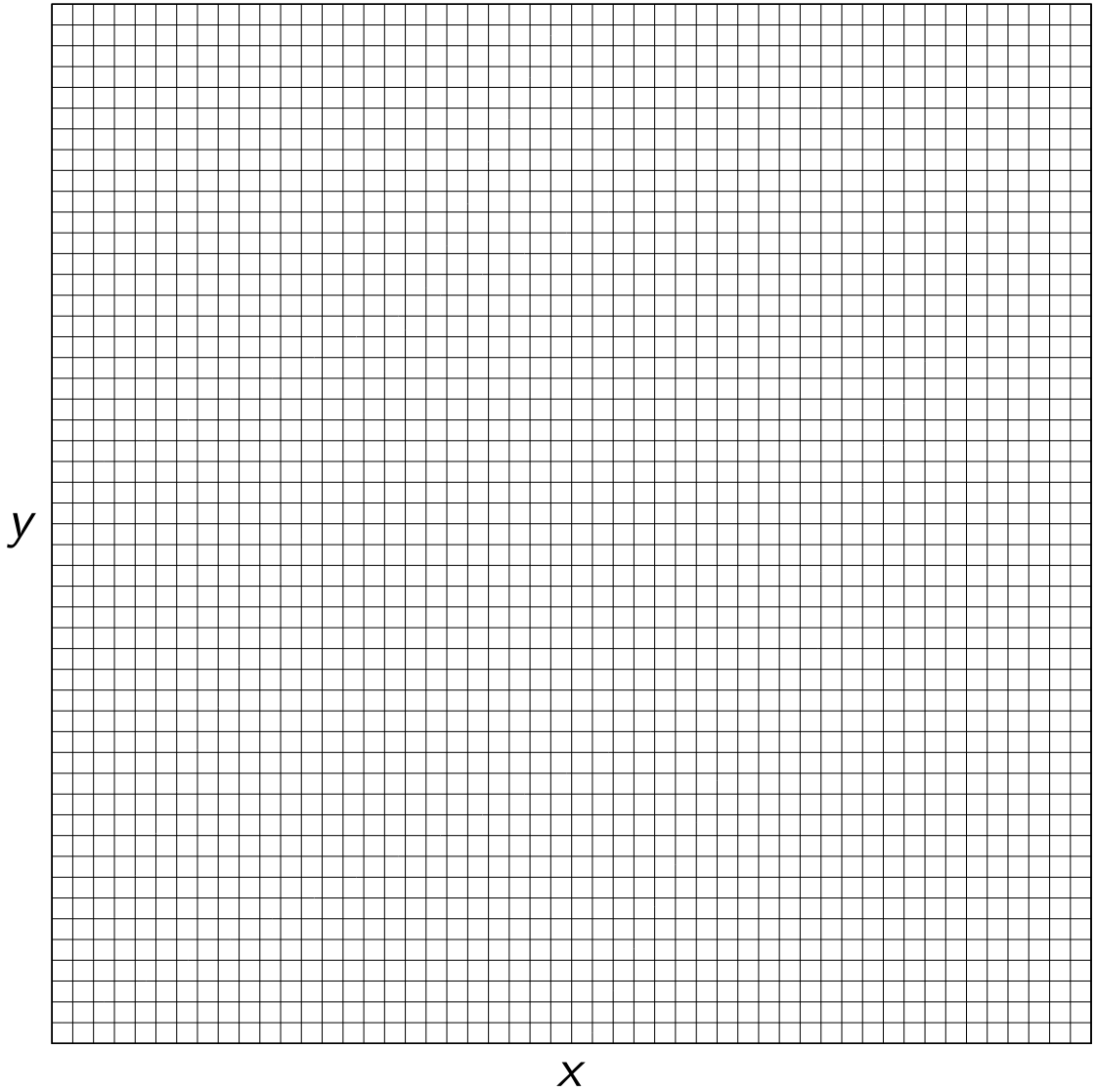
- Verify that the solution is also independent of the mesh resolution.
- Grid density is high enough where all flow features are being resolved
- Rerun the simulation for successively finer grids until the solution no longer changes.
- Not checking this is a common cause of erroneous results in simulation, especially in CFD
- Should at least be carried out once for each type of problem that you deal with so that the next time a similar problem arises, you can apply the same mesh sizing.
- The best way to check for a mesh independent solution is to plot a graph of the resultant monitor value vs the number of cells

Example - Lid driven cavity flow

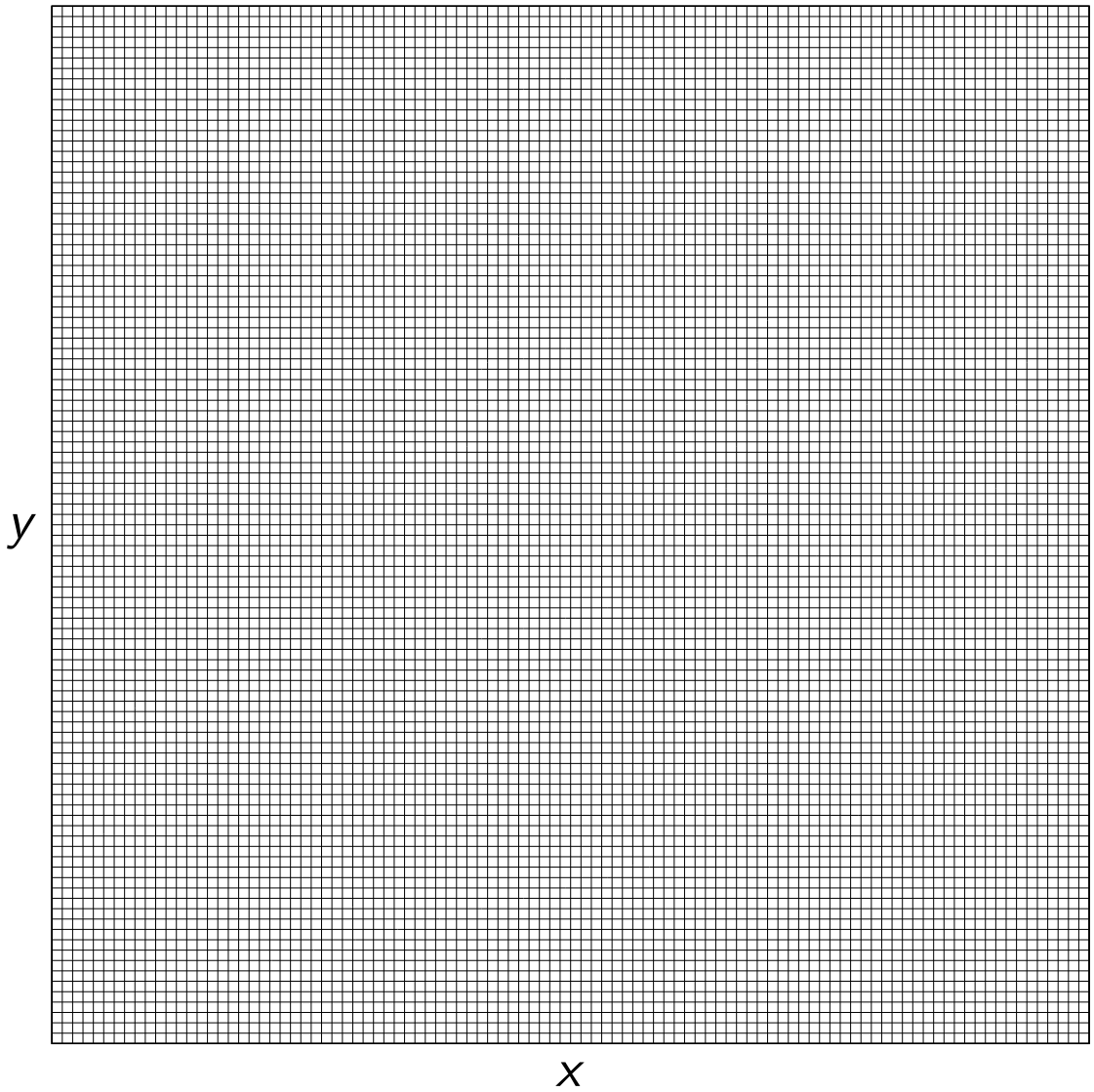


- Classic CFD problem
- Look at the centerline velocity profiles
- Decompose domain into 51×51, 101×101, 201×201, 401×401, and 601×601 nodes

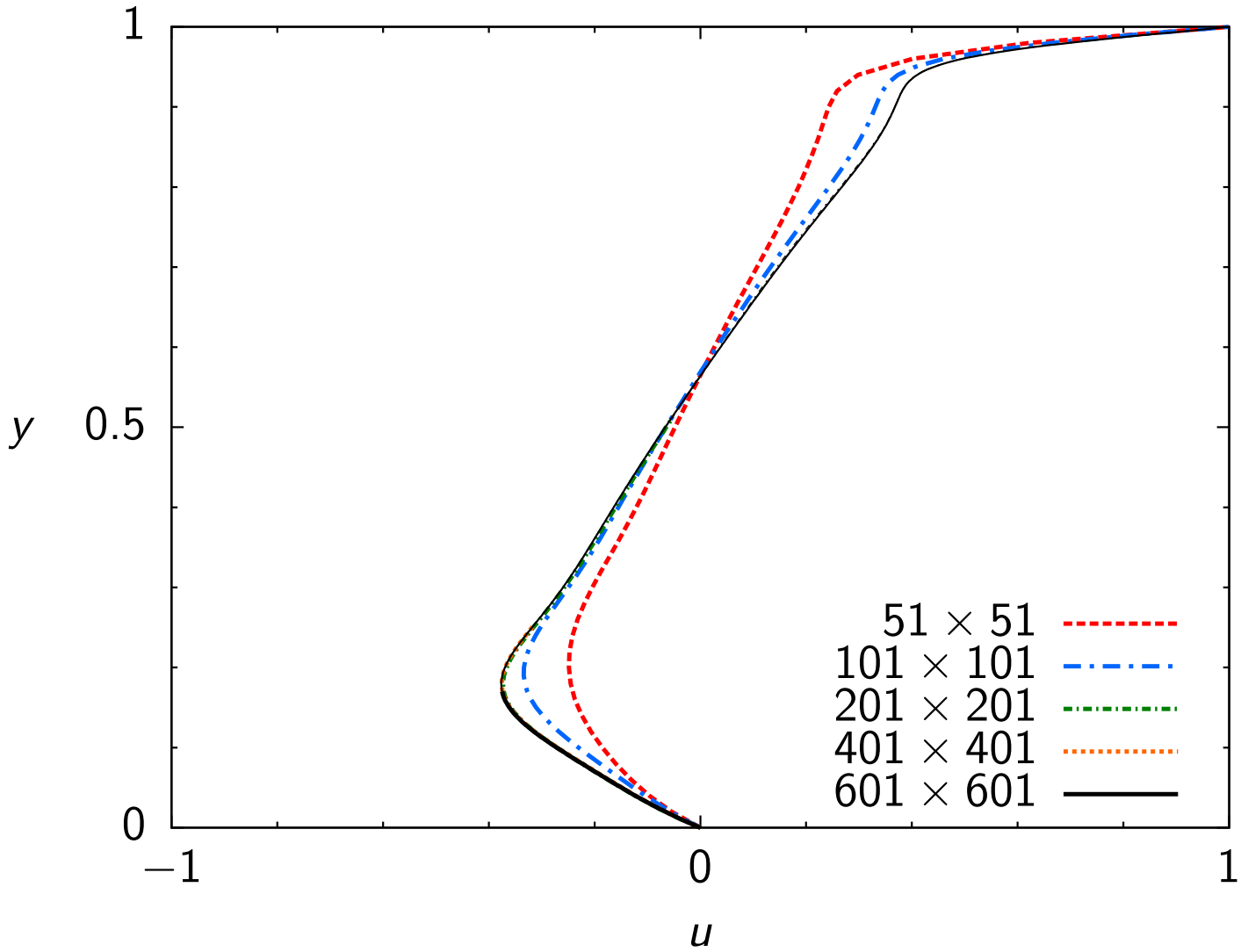
Lid driven cavity flow, 51×51 mesh



Lid driven cavity flow, 101×101 mesh

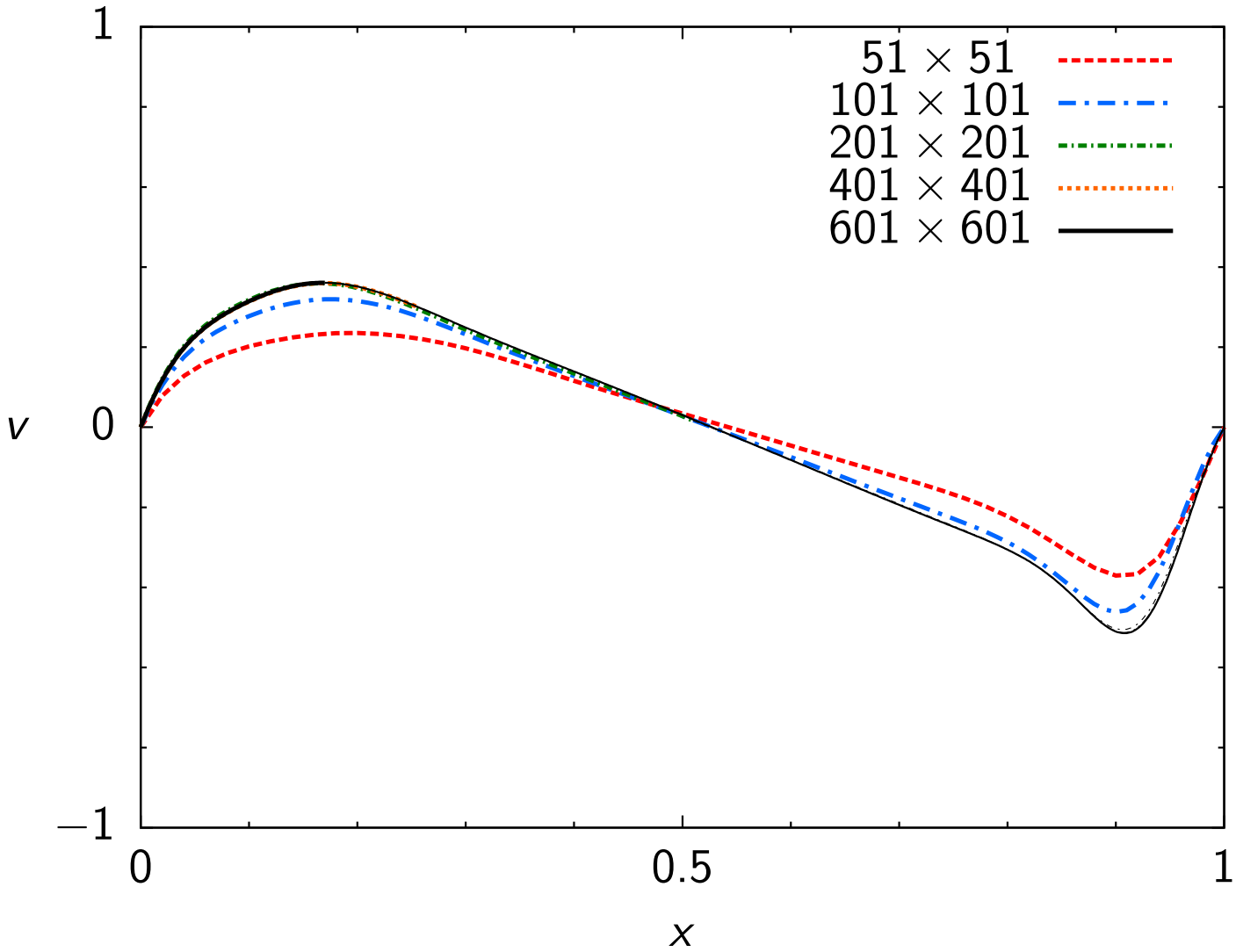


Grid dependency study, u centerline velocity profile



Note: 201×201 is good enough

Grid dependency study, v centerline velocity profile



Note: 201×201 is good enough

Questions?