Solution methods for the Incompressible Navier-Stokes Equations

- Discretization schemes for the Navier-Stokes equations
- Pressure-based approach
- Density-based approach
- Convergence acceleration
- Periodic Flows
- Unsteady Flows

Background (from ME469A or similar)

Navier-Stokes (NS) equations

Finite Volume (FV) discretization

Discretization of space derivatives (upwind, central, QUICK, etc.)

Pressure-velocity coupling issue

Pressure correction schemes (SIMPLE, SIMPLEC, PISO)

Multigrid methods

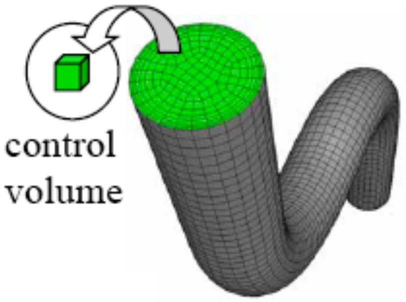
NS equations

Conservation laws:

Rate of change + advection + diffusion = source

$$\frac{d}{dt}\int_{\Omega} \rho d\Omega + \int_{S} \rho(\vec{V} \cdot \vec{n}_S)dS = 0$$

$$\frac{d}{dt}\int_{\Omega}(\rho\vec{V})d\Omega + \int_{S}(\rho\vec{V})(\vec{V} \cdot \vec{n}_S)dS + \int_{S}(\vec{\vec{\tau}} \cdot \vec{n})dS = \int_{S}(-p\vec{n})dS$$



Fluid region of pipe flow discretized into finite set of control volumes (mesh).

NS equations

Differential form:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0$$

$$\frac{\partial}{\partial t}(\rho \vec{v}) + \nabla \cdot (\rho \vec{v} \vec{v}) = -\nabla p + \nabla \cdot (\overline{\overline{\tau}})$$

$$\overline{\overline{\tau}} = \mu \left[(\nabla \vec{v} + \nabla \vec{v}^{\,\mathrm{T}}) - \frac{2}{3} \nabla \cdot \vec{v} I \right]$$

The advection term is non-linear
The mass and momentum equations are coupled (via the velocity)
The pressure appears only as a source term in the momentum equation
No evolution equation for the pressure
There are four equations and five unknowns (ρ, V, p)

NS equations

Compressible flows:

The mass conservation is a transport equation for density. With an additional energy equation p can be specified from a thermodynamic relation (ideal gas law)

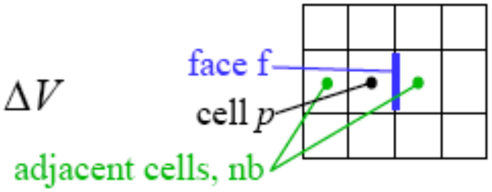
Incompressible flows:

Density variation are not linked to the pressure. The mass conservation is a constraint on the velocity field; this equation (combined with the momentum) can be used to derive an equation for the pressure

Finite Volume Method

Discretize the equations in conservation (integral) form

$$\frac{(\rho\phi_p)^{t+\Delta t} - (\rho\phi_p)^t}{\Delta t}\Delta V + \sum_{\text{faces}} \rho_f \, \phi_f \, V_f \, A_f = \sum_{\text{faces}} \Gamma_f \, (\nabla\phi)_{\perp,f} \, A_f + S_\phi \, \Delta V$$



Eventually this becomes…

$$a_p \, \phi_p + \sum_{nb} a_{nb} \, \phi_{nb} = b_p$$

Pressure-based solution of the NS equation

The continuity equation is combined with the momentum and the divergence-free constraint becomes an elliptic equation for the pressure

$$\frac{\partial}{\partial x_i}\left(\frac{\partial p}{\partial x_i}\right) = -\frac{\partial}{\partial x_i}\left[\frac{\partial\left(\rho u_i u_j\right)}{\partial x_j}\right]$$

To clarify the difficulties related to the treatment of the pressure, we will define EXPLICIT and IMPLICIT schemes to solve the NS equations:

It is assumed that space derivatives in the NS are already discretized:

$$\frac{\partial}{\partial x_i} \rightarrow \frac{\delta}{\delta x_i}$$

Explicit scheme for NS equations

Semi-discrete form of the NS

$$\frac{\partial\left(\rho u_i\right)}{\partial t}=-\frac{\delta\left(\rho u_i u_j\right)}{\delta x_j}+\frac{\delta \tau_{ij}}{\delta x_i}-\frac{\delta p}{\delta x_i}=H_i-\frac{\delta p}{\delta x_i}$$

Explicit time integration

$$\left(\rho u_i\right)^{n+1}-\left(\rho u_i\right)^n=\Delta t\left(H_i^n-\frac{\delta p}{\delta x_i}^n\right)$$

The n+1 velocity field is NOT divergence free

$$\frac{\delta\left(\rho u_i\right)^{n+1}}{\delta x_i} \neq 0$$

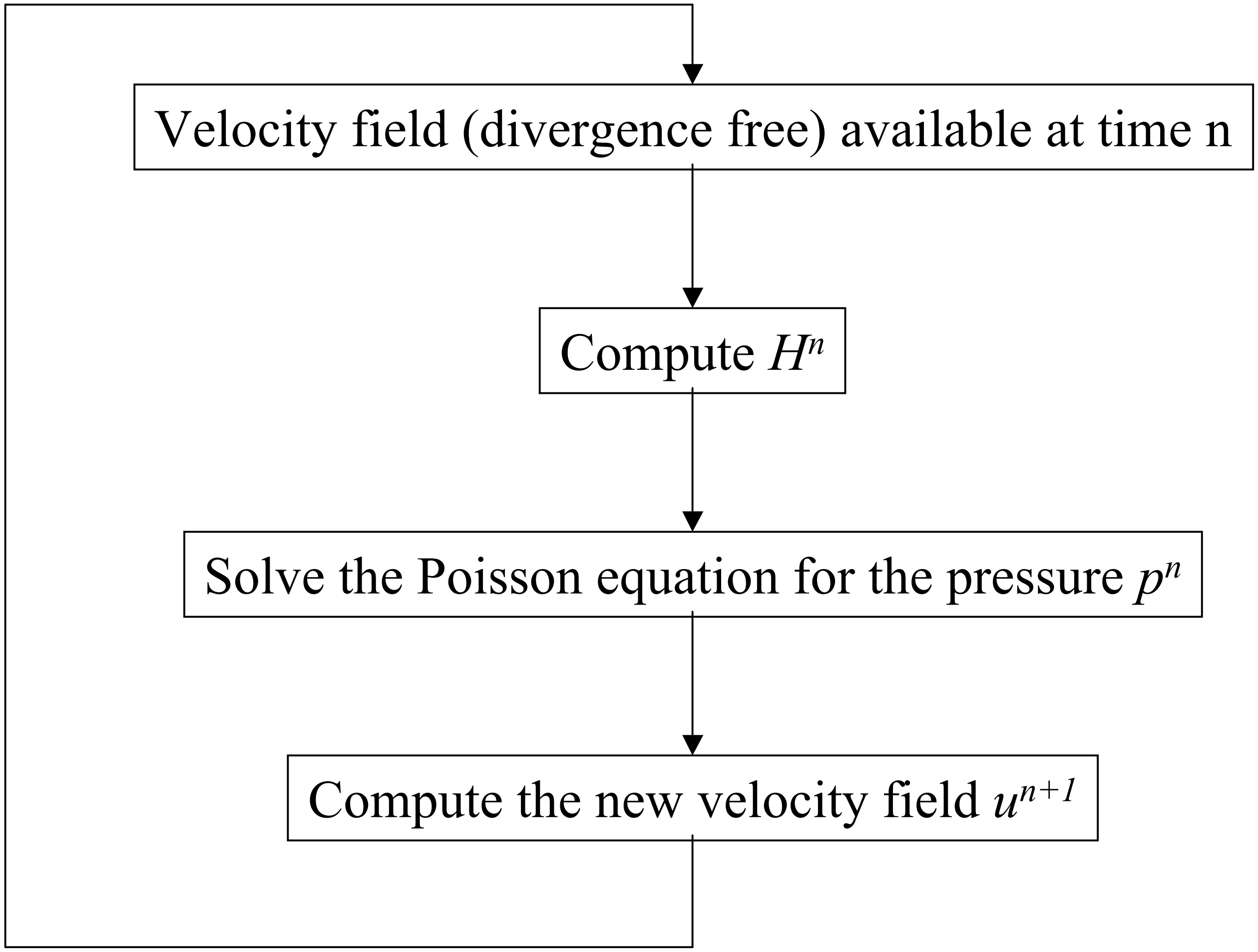
Take the divergence of the momentum

$$\frac{\delta}{\delta x_i}\left(\rho u_i\right)^{n+1}-\frac{\delta}{\delta x_i}\left(\rho u_i\right)^n=\Delta t \frac{\delta}{\delta x_i}\left(H_i^n-\frac{\delta p}{\delta x_i}^n\right)$$

Elliptic equation for the pressure

$$\frac{\delta}{\delta x_i}\left(\frac{\delta p^n}{\delta x_i}\right)=\frac{\delta}{\delta x_i} H_i^n$$

Explicit pressure-based scheme for NS equations



Velocity field (divergence free) available at time n

Compute H^n

Solve the Poisson equation for the pressure p^n

Compute the new velocity field u^{n+1}

Implicit scheme for NS equations

Semi-discrete form of the NS

$$\frac{\partial\left(\rho u_i\right)}{\partial t}=-\frac{\delta\left(\rho u_i u_j\right)}{\delta x_j}+\frac{\delta \tau_{ij}}{\delta x_i}-\frac{\delta p}{\delta x_i}=H_i-\frac{\delta p}{\delta x_i}$$

Implicit time integration

$$\left(\rho u_i\right)^{n+1}-\left(\rho u_i\right)^n=\Delta t\left(H_i^{n+1}-\frac{\delta p}{\delta x_i}^{n+1}\right)$$

Take the divergence of the momentum

$$\frac{\delta}{\delta x_i}\left(\frac{\delta p^{n+1}}{\delta x_i}\right)=\frac{\delta}{\delta x_i}H_i^{n+1}$$

The equations are coupled and non-linear

Navier-Stokes Equations

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \dfrac{\partial (\rho v_i)}{\partial x_i} = 0 \\ \dfrac{\partial \rho v_i}{\partial t} + \dfrac{\partial (\rho v_j v_i)}{\partial x_j} = \dfrac{\partial \tau_{ij}}{\partial x_j} - \dfrac{\partial p}{\partial x_i} \\ \dfrac{\partial (\rho E)}{\partial t} + \dfrac{\partial (\rho v_j E)}{\partial x_j} = \dfrac{\partial}{\partial x_j}\left(k\dfrac{\partial T}{\partial x_j}\right) + \dfrac{\partial}{\partial x_j}(\tau_{ij} v_i) \end{cases}$$

Conservation of mass

Conservation of momentum

Conservation of energy

Newtonian fluid $$\tau_{ij} = \mu\left[\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) - \frac{2}{3}\frac{\partial v_k}{\partial x_k}\right]$$

In 3D: 5 equations & 6 unknowns: *p*, ρ, v_i, *E(T)*

Need supplemental information: equation of state

Approximations

Although the Navier-Stokes equations are considered the appropriate conceptual model for fluid flows they contain 3 major approximations:

1. Continuum hypothesis
2. Form of the diffusive fluxes
3. Equation of state

Simplified conceptual models can be derived introducing additional assumptions: incompressible flow

$$\frac{\partial u_i}{\partial x_i} = 0$$

Conservation of mass (continuity)

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_j u_i}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\mu \frac{\partial u_i}{\partial x_j}\right) - \frac{1}{\rho}\frac{\partial p}{\partial x_i}$$

Conservation of momentum

Difficulties:

Non-linearity, coupling, role of the pressure

A Solution Approach

The momentum equation can be interpreted as a advection/diffusion equation for the velocity vector

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_j u_i}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\mu \frac{\partial u_i}{\partial x_j}\right) - \frac{1}{\rho}\frac{\partial p}{\partial x_i}$$

The mass conservation should be used to derive the pressure… taking the divergence of the momentum:

$$\frac{\partial}{\partial x_i}\left[\xcancel{\frac{\partial u_i}{\partial t}} + \frac{\partial u_j u_i}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\mu \frac{\partial u_i}{\partial x_j}\right) - \frac{1}{\rho}\frac{\partial p}{\partial x_i}\right]$$

A Poisson equation for the pressure is derived

$$\frac{\partial}{\partial x_i}\left(\frac{1}{\rho}\frac{\partial p}{\partial x_i}\right) = RHS$$

The Projection Method

$$\left\{\begin{aligned} &\frac{u_i^{n+1}-u_i^n}{\Delta t}+C_i^{n+1}=D_i^{n+1}-\frac{1}{\rho}\frac{\partial p}{\partial x_i}^{n+1} \\ &\frac{\partial}{\partial x_i}\left(\frac{1}{\rho}\frac{\partial p}{\partial x_i}^{n+1}\right)=RHS^{n+1}\end{aligned}\right.$$

Implicit, coupled and non-linear

Predicted velocity $\frac{u_i^*-u_i^n}{\Delta t}+C_i^m=D_i^m-\frac{1}{\rho}\frac{\partial p^n}{\partial x_i}$ but $\frac{\delta u^*}{\delta x_i}\neq 0$

assuming $p^{n+1}=p^n+p^*$ and taking the divergence

$$\frac{1}{\Delta t}\left(\frac{\delta u_i^*}{\delta x_i}-\cancel{\frac{\delta u_i^n}{\delta x_i}}\right)+\underbrace{\frac{\delta C_i^m}{\delta x_i}=\frac{\delta D_i^m}{\delta x_i}-\frac{\delta}{\delta x_i}\Bigg(\frac{1}{\rho}\frac{\delta p}{\delta x_i}^{n+1}}_{}-\frac{1}{\rho}\frac{\delta p^*}{\delta x_i}\Bigg)$$

this is what we would like to enforce

we obtain $\frac{\delta}{\delta x_i}\left(\frac{1}{\rho}\frac{\delta p^*}{\delta x_i}\right)=\frac{1}{\Delta t}\frac{\delta u_i^*}{\delta x_i}$

combining (corrector step) $\frac{u_i^{n+1}-u_i^*}{\Delta t}=-\frac{1}{\rho}\frac{\delta p^*}{\delta x_i}$

Alternative View of Projection

Reorganize the NS equations (Uzawa)
$$\begin{bmatrix} A & G \\ D & 0 \end{bmatrix}\begin{bmatrix} v^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} RHS \\ 0 \end{bmatrix}$$

LU decomposition
$$\begin{bmatrix} A & 0 \\ D & -DA^{-1}G \end{bmatrix}\begin{bmatrix} I & A^{-1}G \\ 0 & I \end{bmatrix}\begin{bmatrix} v^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} RHS \\ 0 \end{bmatrix}$$

Exact splitting
$$\left\{ \begin{array}{l} \begin{bmatrix} A & 0 \\ D & -DA^{-1}G \end{bmatrix}\begin{bmatrix} v^{*} \\ p^{*} \end{bmatrix} = \begin{bmatrix} RHS \\ 0 \end{bmatrix} \\ \\ \begin{bmatrix} I & A^{-1}G \\ 0 & I \end{bmatrix}\begin{bmatrix} v^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} v^{*} \\ p^{*} \end{bmatrix} \end{array} \right.$$

$$\begin{aligned} Av^{*} &= RHS \\ Dv^{*} - DA^{-1}Gp^{*} &= 0 \\ v^{n+1} + A^{-1}Gp^{n+1} &= v^{*} \\ p^{n+1} &= p^{*} \end{aligned}$$

Momentum eqs.

Pressure Poisson eq.

Velocity correction

Alternative View of Projection

Exact projection requires the inversion of the LHS of the momentum eq. thus is costly.

Approximate projection methods are constructed using two auxiliary matrices (time-scales)

$$\begin{aligned} Av^* &= RHS \\ Dv^* - DB_1Gp^{n+1} &= 0 \\ v^{n+1} + B_2Gp^{n+1} &= v^* \end{aligned}$$

Momentum eqs.

Pressure Poisson eq.

Velocity correction

The simplest (conventional) choice is

$$B_1 = B_2 \approx I\Delta t$$

What about steady state?

Solution of the steady-state NS equations is of primary importance

Steady vs. unsteady is another hypothesis that requires formalization…

Mom. Equations

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_j u_i}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\mu \frac{\partial u_i}{\partial x_j}\right) - \frac{1}{\rho}\frac{\partial p}{\partial x_i}$$

Reference Quantities

$$\tilde{t} = \frac{t}{T} \qquad \tilde{x_i} = \frac{x_i}{L} \qquad \tilde{u_i} = \frac{u_i}{U} \qquad \tilde{p} = \frac{p}{\rho U^2}$$

Non dimensional Eqn

$$St\frac{\partial u_i}{\partial t} + \frac{\partial u_j u_i}{\partial x_j} = \frac{1}{Re}\frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j}\right) - \frac{\partial p}{\partial x_i}$$

Reynolds and Strouhal #s

$$Re = \frac{UL}{\nu} \qquad St = \frac{L}{TU} = \frac{fL}{U}$$

Implicit scheme for steady NS equations

Compute an intermediate velocity field
(eqns are STILL non-linear)

$$a_P(u_i)^*_P = \sum_f a_f(u_i^* \cdot n_i)_f - \frac{1}{\rho}\frac{\delta p}{\delta x_i}^n$$

Define a velocity and a pressure correction

$$\begin{cases} u^{n+1} = u^* + u' \\ p^{n+1} = p^n + p' \end{cases}$$

Using the definition and combining

$$\begin{cases} a_P(u_i)^{n+1}_P = \sum_f a_f(u_i^{n+1} \cdot n_i)_f - \frac{1}{\rho}\frac{\delta p}{\delta x_i}^{n+1} \\ a_P(u_i)^*_P = \sum_f a_f(u_i^* \cdot n_i)_f - \frac{1}{\rho}\frac{\delta p}{\delta x_i}^n \end{cases}$$

Derive an equation for u’

$$a_P(u_i)'_P = \sum_f a_f[(u_i^{n+1} - u_i^*) \cdot n_i]_f - \frac{1}{\rho}\frac{\delta p}{\delta x_i}'$$

$$(u_i)'_P = (\tilde{u_i})' - \frac{1}{a_P}\frac{1}{\rho}\frac{\delta p}{\delta x_i}'$$

Implicit scheme for steady NS equations

Taking the divergence..

$$\frac{\delta}{\delta x_i}(u_i)_P^{n+1} = 0 = \frac{\delta}{\delta x_i}(u_i)_P^* + \frac{\delta}{\delta x_i}(u_i)_P'$$

$$0 = \frac{\delta}{\delta x_i}(u_i)_P^* + \frac{\delta}{\delta x_i}(\tilde{u_i})' - \frac{\delta}{\delta x_i}\left(\frac{1}{a_P}\frac{1}{\rho}\frac{\delta p'}{\delta x_i}\right)$$

We obtain a Poisson system for the pressure correction…

Solving it and computing a gradient:

$$(u_i)_P' = (\tilde{u_i})' - \frac{1}{a_P}\frac{1}{\rho}\frac{\delta p'}{\delta x_i}$$

So we can update

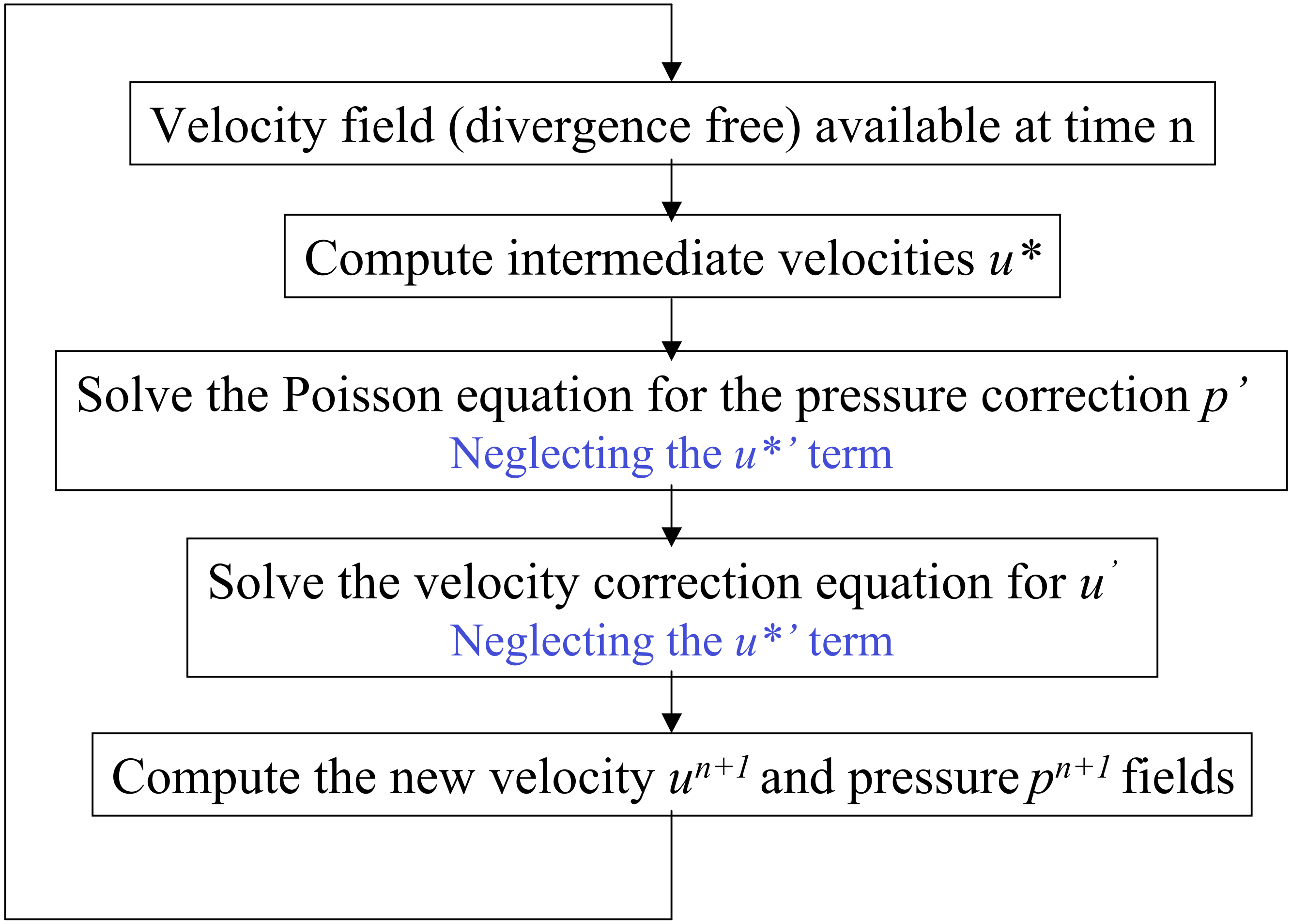
$$u^{n+1} = u^* + u'$$

And also the pressure at the next level

$$p^{n+1} = p^n + p'$$

Implicit pressure-based scheme for NS equations (SIMPLE)

SIMPLE: Semi-Implicit Method for Pressure-Linked Equations



Velocity field (divergence free) available at time n

Compute intermediate velocities u^*

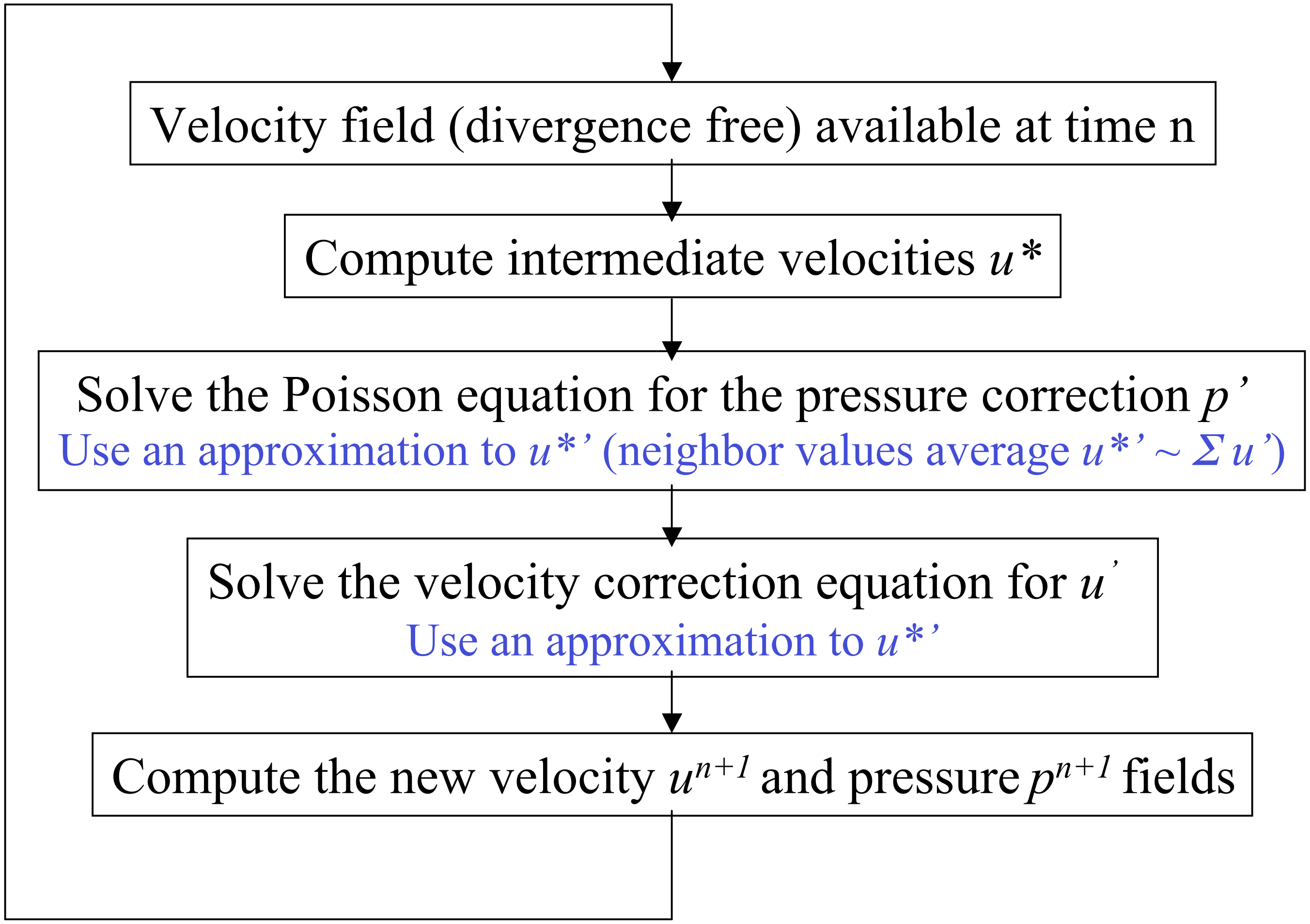
Solve the Poisson equation for the pressure correction p'
Neglecting the $u^{*'}$ term

Solve the velocity correction equation for u'
Neglecting the $u^{*'}$ term

Compute the new velocity u^{n+1} and pressure p^{n+1} fields

Implicit pressure-based scheme for NS equations (SIMPLEC)

SIMPLE: SIMPLE Corrected/Consistent



Velocity field (divergence free) available at time n

Compute intermediate velocities u^*

Solve the Poisson equation for the pressure correction p'

Use an approximation to $u^{*'}$ (neighbor values average $u^{*'} \sim \Sigma u'$)

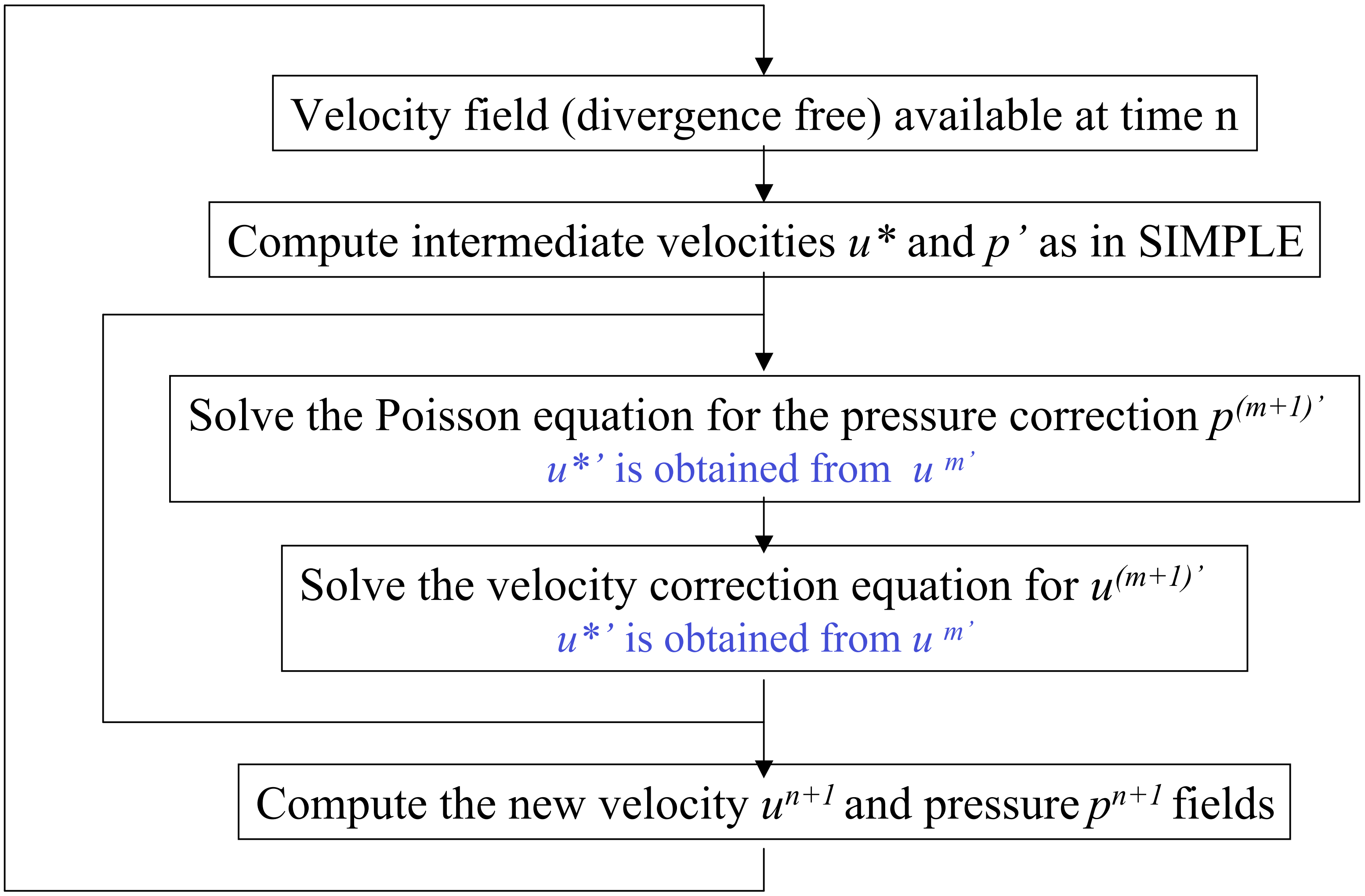
Solve the velocity correction equation for u'

Use an approximation to $u^{*'}$

Compute the new velocity u^{n+1} and pressure p^{n+1} fields

Implicit pressure-based scheme for NS equations (PISO)

PISO: Pressure Implicit with Splitting Operators



Velocity field (divergence free) available at time n

Compute intermediate velocities u^* and p' as in SIMPLE

Solve the Poisson equation for the pressure correction $p^{(m+1)'}$

$u^{*'}$ is obtained from $u^{m'}$

Solve the velocity correction equation for $u^{(m+1)'}$

$u^{*'}$ is obtained from $u^{m'}$

Compute the new velocity u^{n+1} and pressure p^{n+1} fields

SIMPLE, SIMPLEC & PISO - Comments

In SIMPLE under-relaxation is required due to the neglect of $u^{*\prime}$

$$u^{n+1} = u^* + \alpha_u u' \qquad\qquad p = p^n + \alpha_p p'$$

There is an optimal relationship $\alpha_p = 1 - \alpha_u$

SIMPLEC and PISO do not need under-relaxation

SIMPLEC/PISO allow faster convergence than SIMPLE

PISO is useful for irregular cells

Under-relaxation

Is used to increase stability (smoothing)

Variable under-relaxation

$$\phi = \phi_{\mathrm{old}} + \alpha \Delta \phi$$

Equation (implicit) under-relaxation

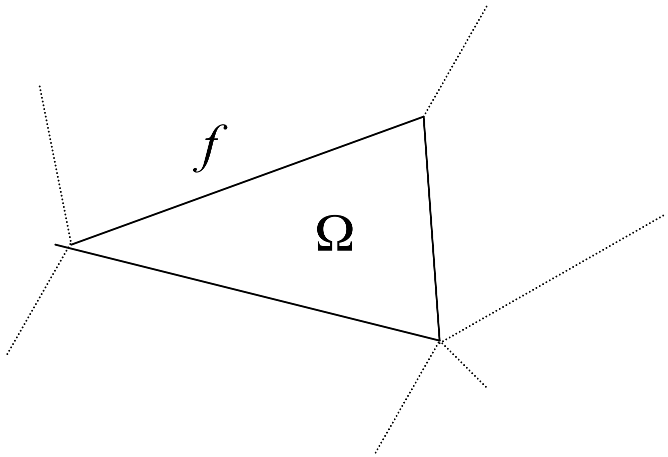
$$\frac{a_p \phi}{\alpha} = \sum_{nb} a_{nb} \phi_{nb} + b + \frac{1-\alpha}{\alpha} a_p \phi_{\mathrm{old}}$$

Segregated (pressure based) solver in FLUENT

FV discretization for mixed elements

$$\int_S \rho\phi(\vec{V}\cdot\vec{n}_S)dS = \int_S \Gamma_\phi(\vec{\nabla}\phi\cdot\vec{n})dS + \int_\Omega S_\phi d\Omega$$

$$\sum_f^{N_{faces}} \rho_f\phi_f(\vec{V}\cdot\vec{n})_f dS_f = \sum_f^{N_{faces}} \Gamma_\phi(\vec{\nabla}\phi\cdot\vec{n})_f dS_f + S_\phi\Omega$$



The quantities at the cell faces can be computed using several different schemes

Discretization of the equations

Options for the segregated solver in FLUENT

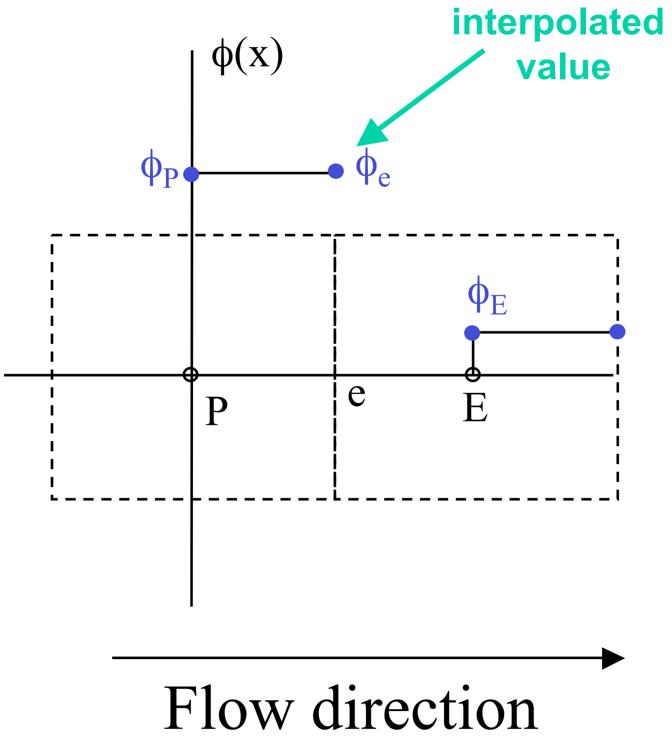
Discretization scheme for convective terms
  - 1st order upwind (UD)
  - 2nd order upwind (TVD)
  - 3rd order upwind (QUICK), only for quad and hex

Pressure interpolation scheme (pressure at the cell-faces)
  - linear (linear between cell neighbors)
  - second-order (similar to the TVD scheme for momentum)
  - PRESTO (mimicking the staggered-variable arrangement)

Pressure-velocity coupling
  - SIMPLE
  - SIMPLEC
  - PISO

Discretization of the convective terms

Determine the face value



1st Order Upwind

Depending on the flow direction ONLY

Very stable but dissipative

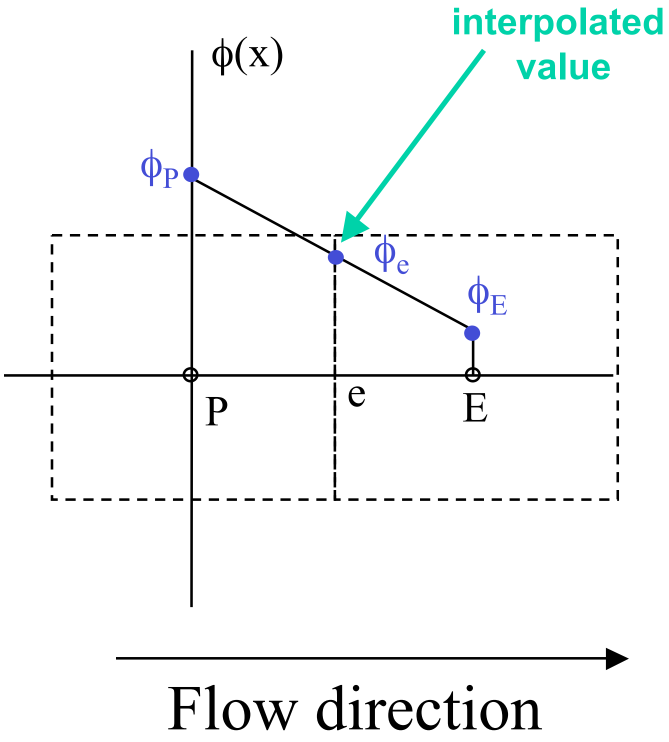
Discretization of the convective terms

Determine the face value

Central differencing (2nd order)

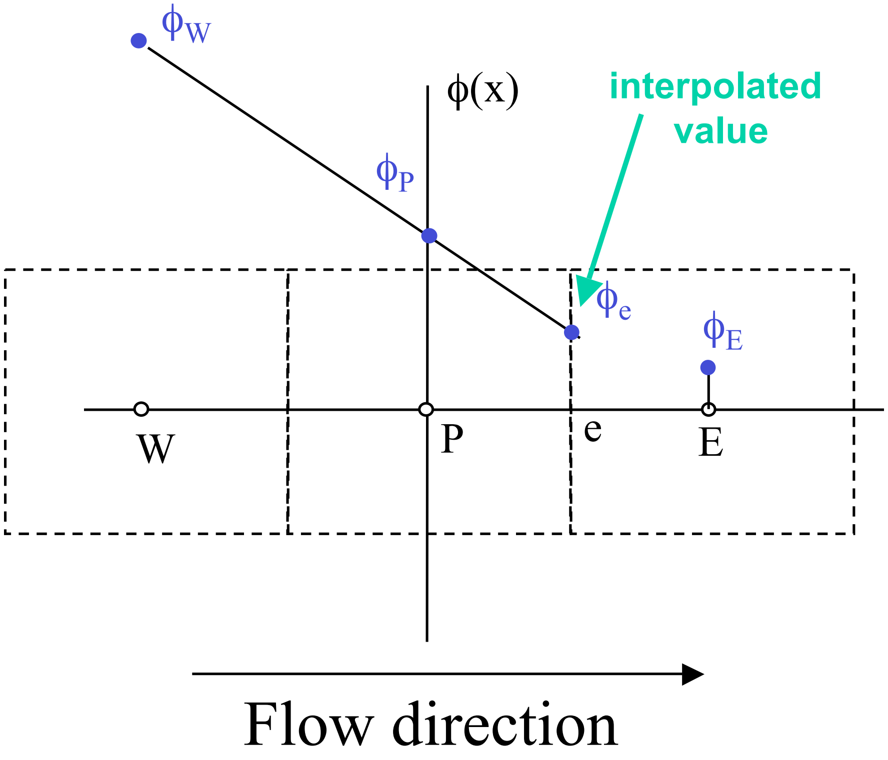
Symmetric. Not depending on the flow direction

Not dissipative but dispersive (odd derivatives)



Discretization of the convective terms

Determine the face value



2nd order upwind

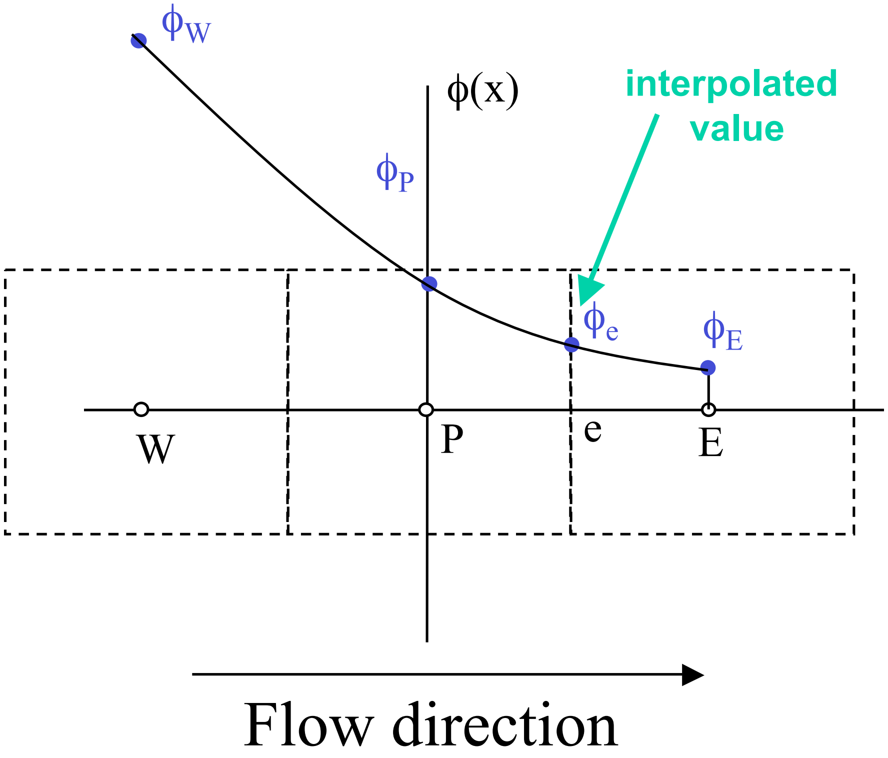
Depends on the flow direction

Less dissipative than 1st order but not bounded (extrema preserving)

Possibility of using limiters

Discretization of the convective terms

Determine the face value



Quick (Quadratic Upwind Interpolation for Convection Kinetics)

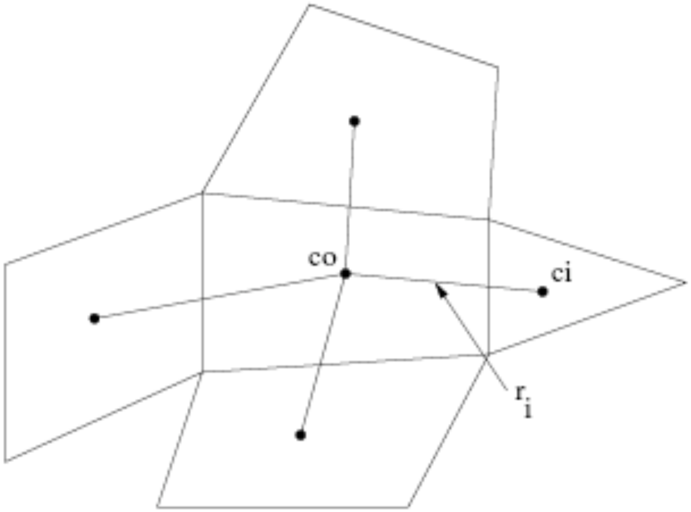
Formally 3rd order

Depends on the flow direction

As before it is not bounded

Evaluation of gradients

Gauss Gradient



$$(\nabla\phi)_{c0} \;=\; \frac{1}{\mathcal{V}}\sum_f \overline{\phi}_f \, \vec{A}_f$$

$$\overline{\phi}_f = \frac{\phi_{c0} + \phi_{c1}}{2}$$

Least Square Gradient

$$(\nabla\phi)_{c0} \cdot \Delta r_i = (\phi_{ci} - \phi_{c0})$$

$$[J](\nabla\phi)_{c0} = \Delta\phi$$

LS system

Solution of the equation

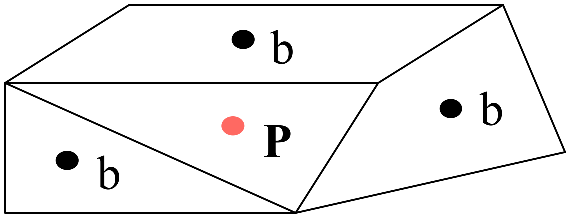
ϕ is one of the velocity component and the convective terms must be linearized:

$$\sum_{f}^{N_{faces}} \rho_f \phi_f (\vec{V} \cdot \vec{n})_f dS_f = \sum_{f}^{N_{faces}} \Gamma_\phi (\nabla \vec{\phi} \cdot \vec{n})_f dS_f + S_\phi \Omega$$

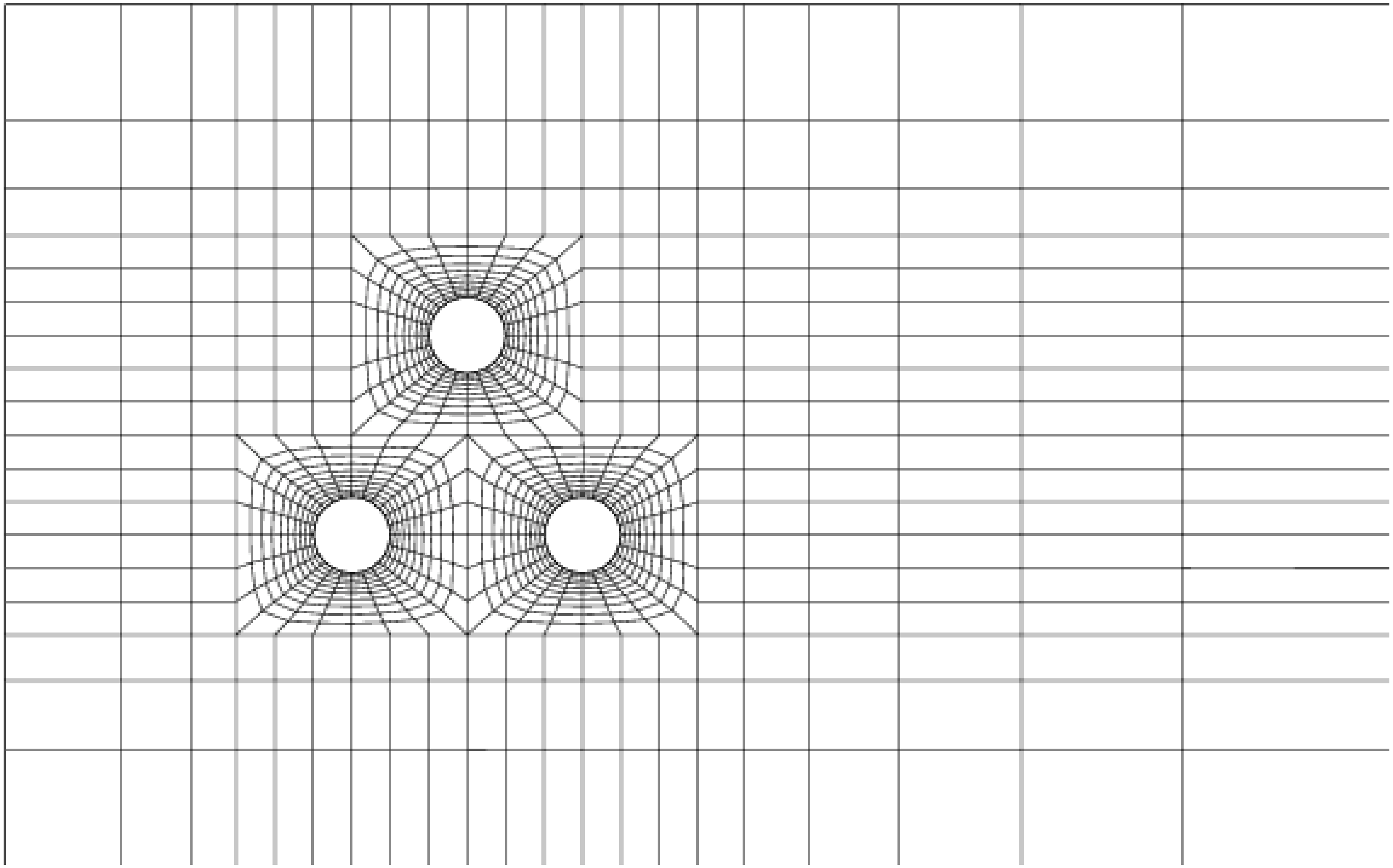
$$a_P \phi_P = \sum_b a_b \phi_b + RHS$$

This correspond to a sparse linear system for each velocity component

Fluent segregated solver uses:
Point Gauss-Seidel technique
Multigrid acceleration

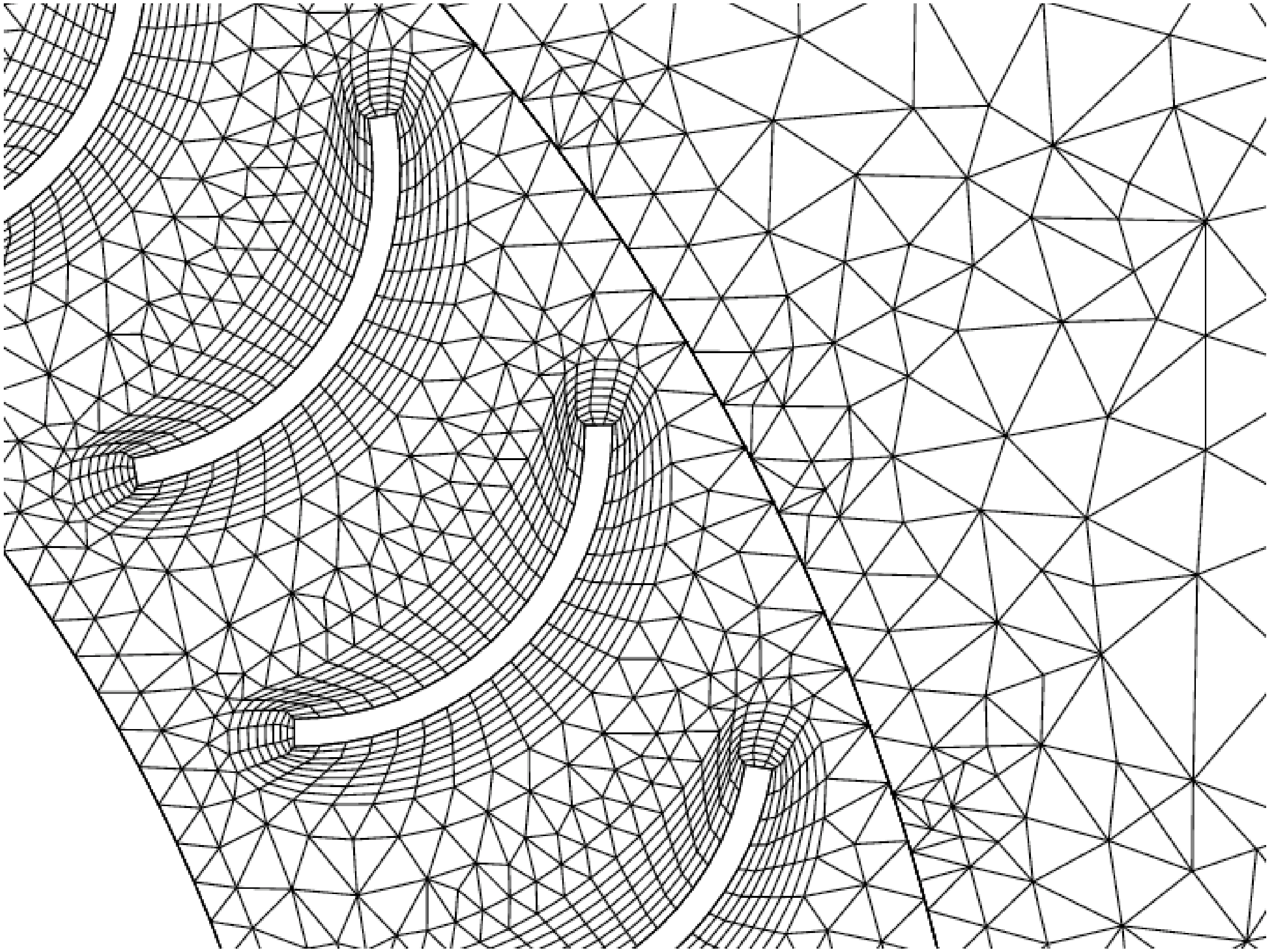


Grids



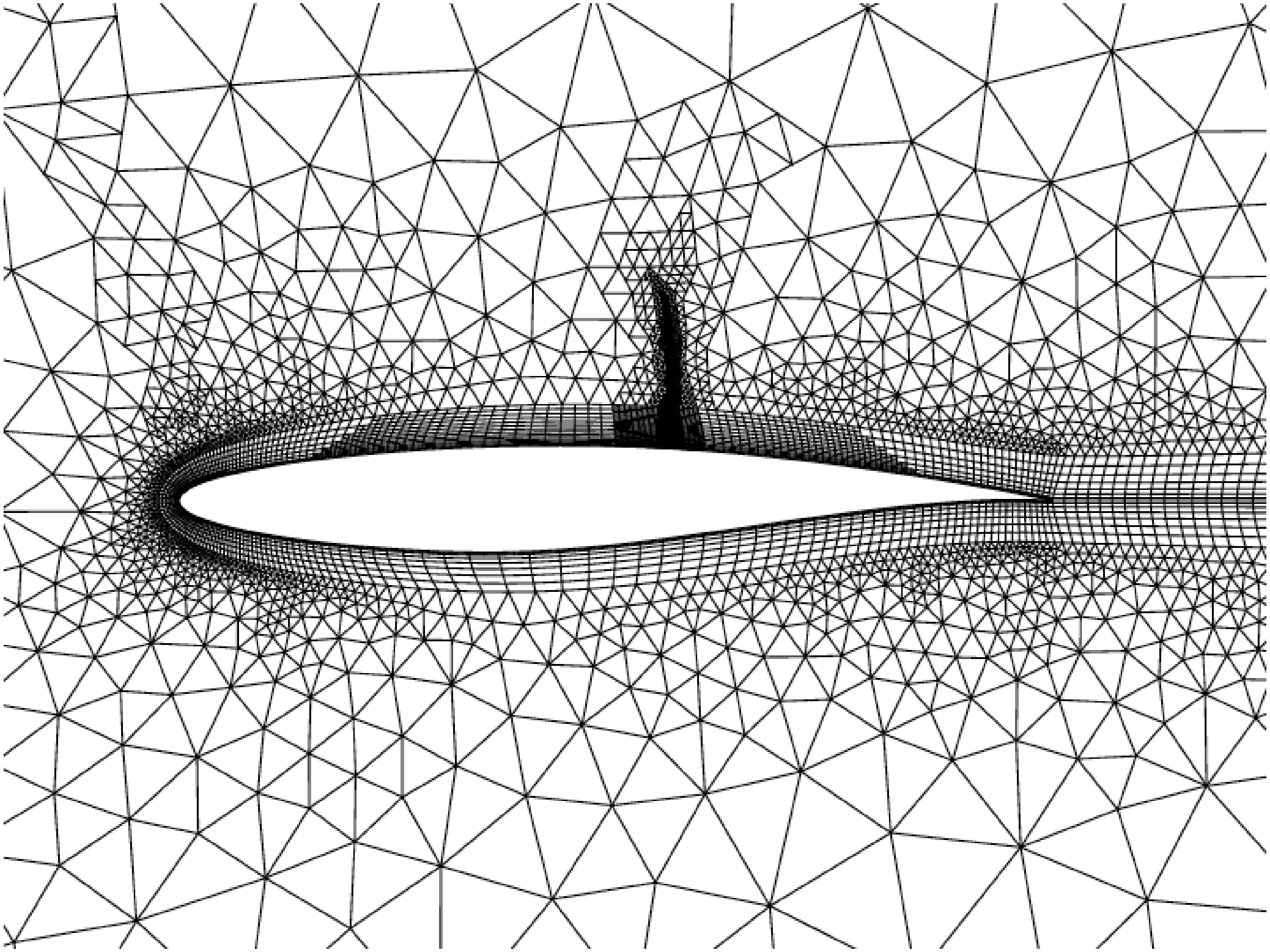
Multiblock structured - Gambit

Grids



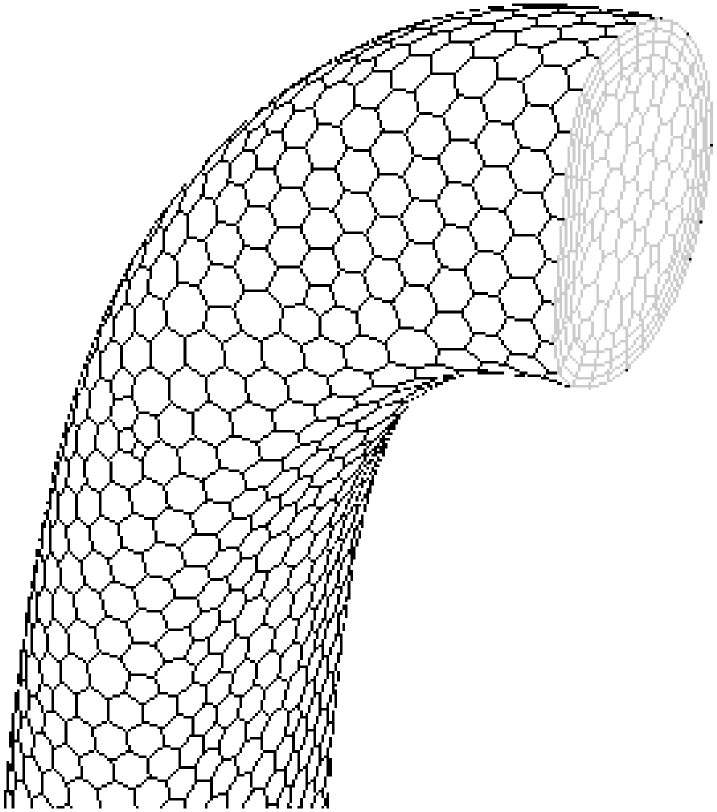
Hybrid non-conformal - Gambit

Grids



Hybrid adaptive - non-Gambit

Grids



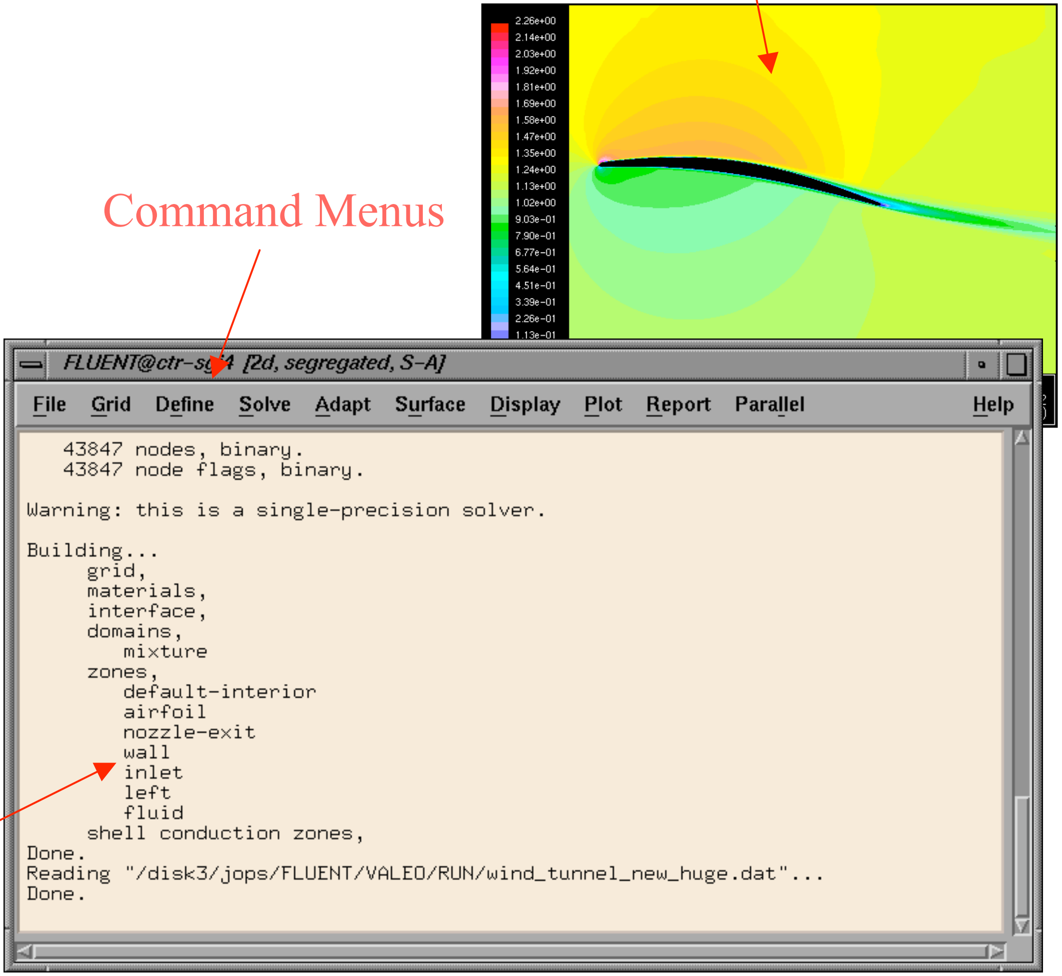
Polyhedral - non-Gambit

Set-up of problems with FLUENT

Read/Import the grid
Define the flow solver option
Define the fluid properties
Define the discretization scheme
Define the boundary condition
Define initial conditions
Define convergence monitors
Run the simulation
Analyze the results

Graphics Window

Command Menus

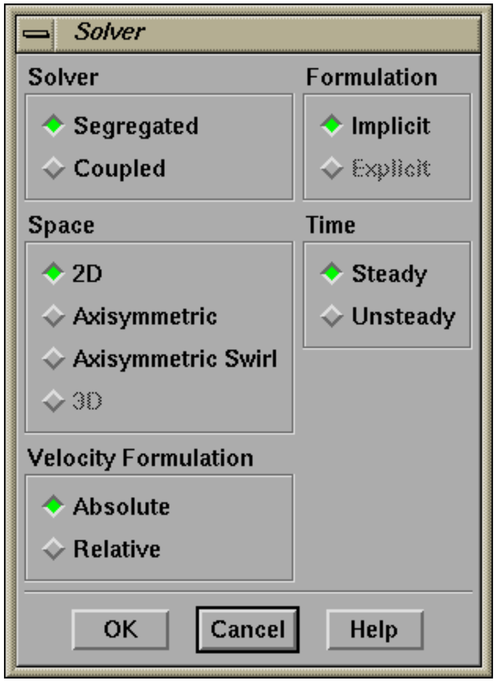


```
   43847 nodes, binary.
   43847 node flags, binary.

Warning: this is a single-precision solver.

Building...
     grid,
     materials,
     interface,
     domains,
        mixture
     zones,
        default-interior
        airfoil
        nozzle-exit
        wall
        inlet
        left
        fluid
     shell conduction zones,
Done.
Reading "/disk3/jops/FLUENT/VALEO/RUN/wind_tunnel_new_huge.dat"...
Done.
```

Text Window

Solver set-up

Define → Models → Solver



Solver
- ● Segregated
- ○ Coupled

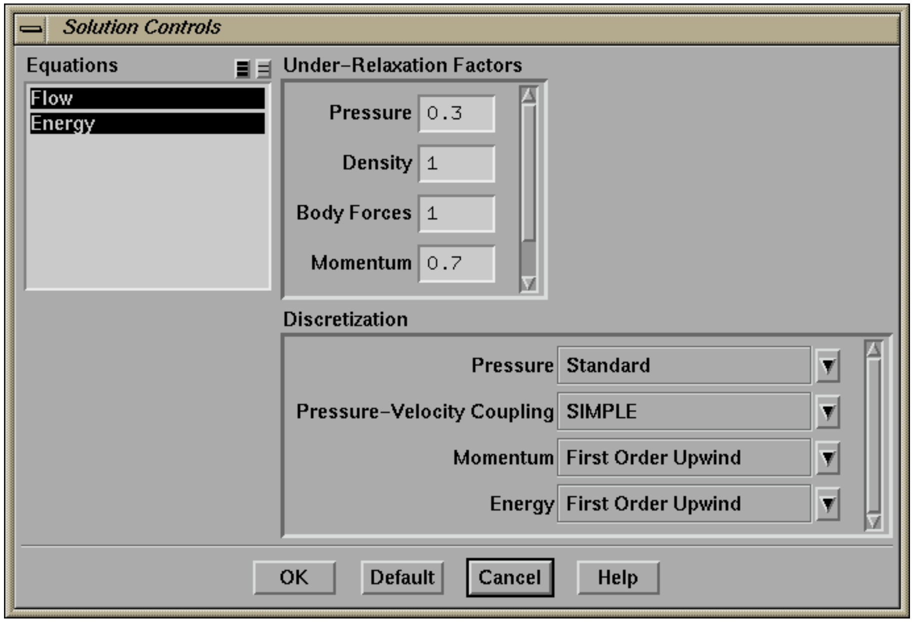
Formulation
- ● Implicit
- ○ Explicit

Space
- ● 2D
- ○ Axisymmetric
- ○ Axisymmetric Swirl
- ○ 3D

Time
- ● Steady
- ○ Unsteady

Velocity Formulation
- ● Absolute
- ○ Relative

Define → Controls → Solution



Equations
- Flow
- Energy

Under-Relaxation Factors

Pressure	0.3
Density	1
Body Forces	1
Momentum	0.7

Discretization

Pressure	Standard
Pressure-Velocity Coupling	SIMPLE
Momentum	First Order Upwind
Energy	First Order Upwind

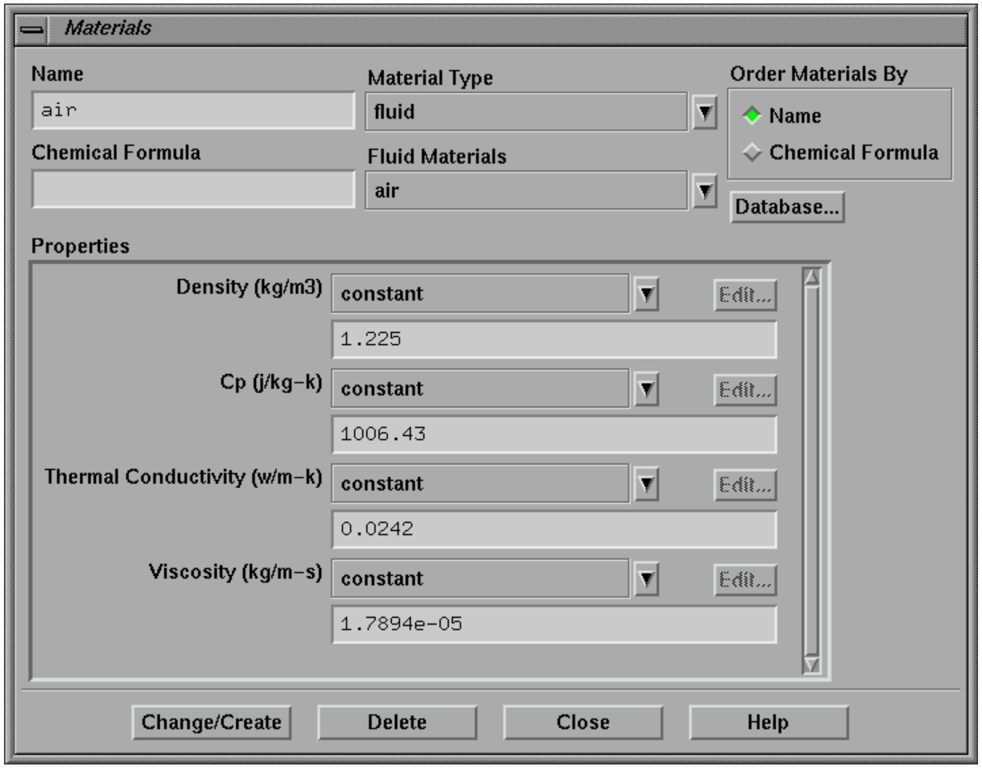
Example: text commands can be used (useful for batch execution)

```
define/models/solver segregated
define/models/steady
…
```

```
solve/set/discretization-scheme/mom 1
solve/set/under-relaxation/mom 0.7
…
```

Material properties

Define → Materials



Name: air

Material Type: fluid

Order Materials By: ● Name ○ Chemical Formula

Chemical Formula:

Fluid Materials: air

Properties

Density (kg/m3): constant — 1.225

Cp (j/kg-k): constant — 1006.43

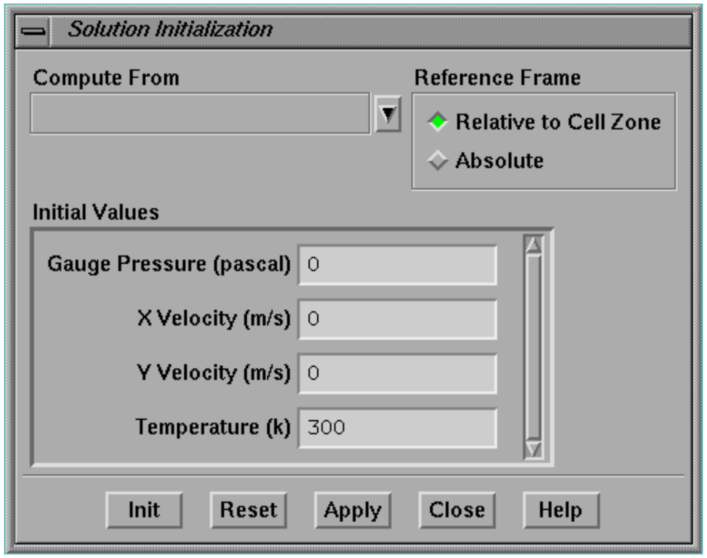
Thermal Conductivity (w/m-k): constant — 0.0242

Viscosity (kg/m-s): constant — 1.7894e-05

Quantities are ALWAYS dimensional

Initial and boundary conditions

Solve → Initialize → Initialize



Solution Initialization

Compute From

Reference Frame

● Relative to Cell Zone

○ Absolute

Initial Values

Gauge Pressure (pascal): 0

X Velocity (m/s): 0

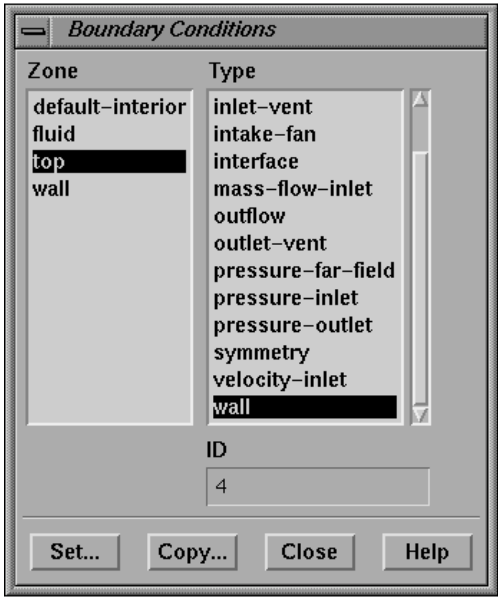
Y Velocity (m/s): 0

Temperature (k): 300

Only constant values can be specified

More flexibility is allowed via patching

Define → Boundary Conditions



Boundary Conditions

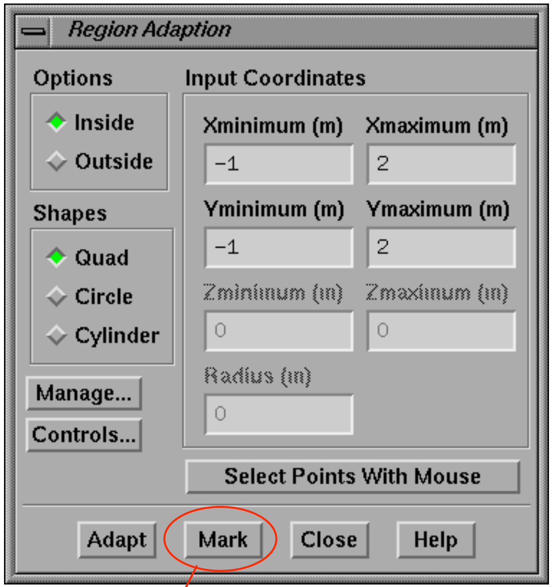
Zone	Type
default-interior	inlet-vent
fluid	intake-fan
top	interface
wall	mass-flow-inlet
	outflow
	outlet-vent
	pressure-far-field
	pressure-inlet
	pressure-outlet
	symmetry
	velocity-inlet
	wall

ID: 4

BCs will be discussed case-by-case

Initial conditions using patching

Adapt → Region → Mark



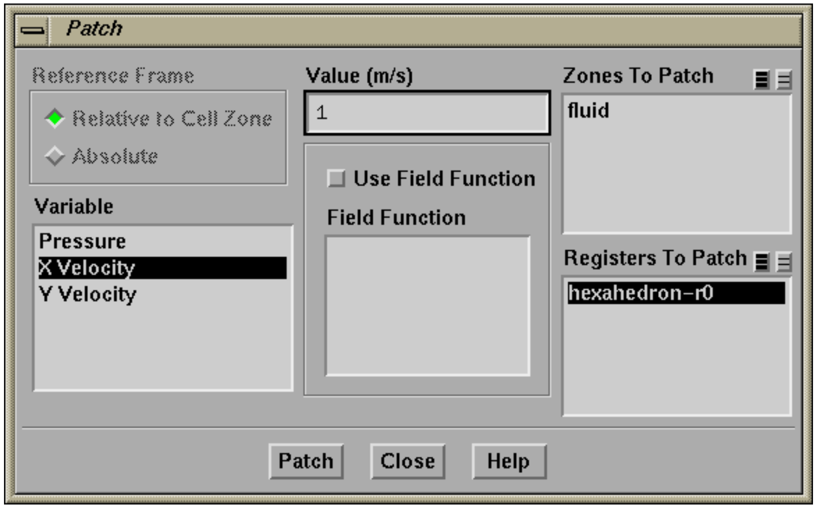
Options: ◆ Inside ◇ Outside

Shapes: ◆ Quad ◇ Circle ◇ Cylinder

Xminimum (m)	Xmaximum (m)
-1	2
Yminimum (m)	Ymaximum (m)
-1	2
Zminimum (m)	Zmaximum (m)
0	0
Radius (m)	
0	

Mark a certain region of the domain
(cells are stored in a register)

Solve → Initialize → Patch



Reference Frame: ◆ Relative to Cell Zone ◇ Absolute

Value (m/s): 1

☐ Use Field Function

Field Function:

Variable: Pressure, **X Velocity**, Y Velocity

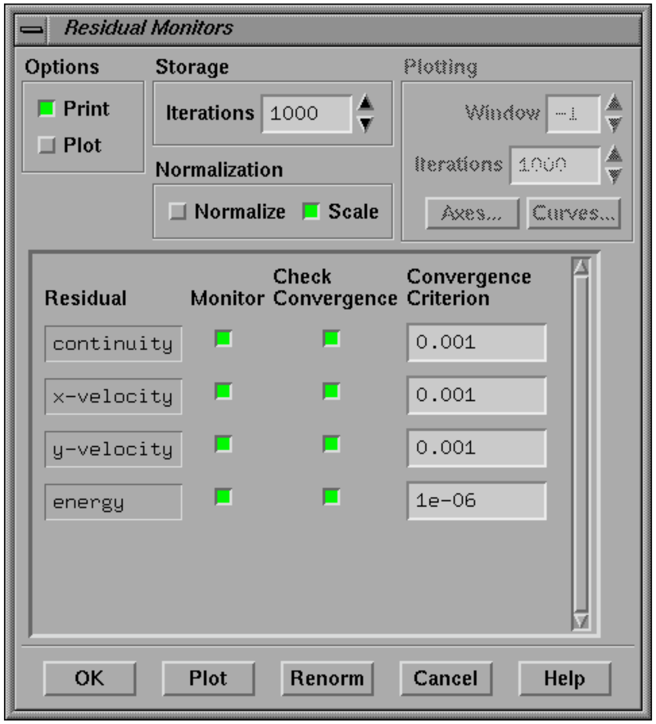
Zones To Patch: fluid

Registers To Patch: **hexahedron-r0**

Patch desired values for each variable
in the region (register) selected

Convergence monitors

Solve → Monitors → Residuals



Options: ☑ Print ☐ Plot

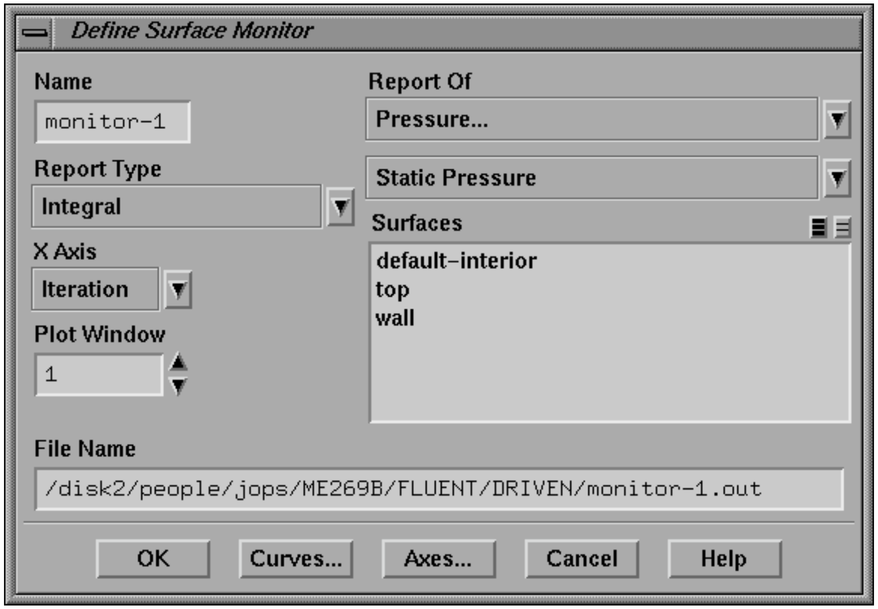
Storage: Iterations 1000

Normalization: ☐ Normalize ☑ Scale

Plotting: Window 1 Iterations 1000

Residual	Monitor	Check Convergence	Convergence Criterion
continuity	☑	☑	0.001
x-velocity	☑	☑	0.001
y-velocity	☑	☑	0.001
energy	☑	☑	1e-06

Solve → Monitors → Surface



Name: monitor-1

Report Of: Pressure...

Static Pressure

Report Type: Integral

X Axis: Iteration

Plot Window: 1

Surfaces:
default-interior
top
wall

File Name: /disk2/people/jops/ME269B/FLUENT/DRIVEN/monitor-1.out

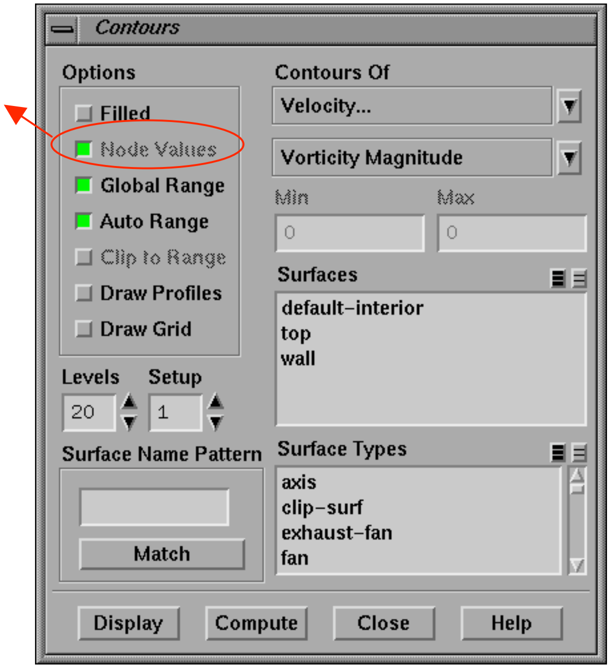
Convergence history of the equation residuals are stored together with the solution
User-defined monitors are NOT stored by default

Postprocessing

Display → Contours

Plot → XY Plot

Cell-centered data are Computed This switch interpolates the results on the cell-vertices



Contours

Options

☐ Filled
☑ Node Values
☑ Global Range
☑ Auto Range
☐ Clip to Range
☐ Draw Profiles
☐ Draw Grid

Contours Of: Velocity...
Vorticity Magnitude

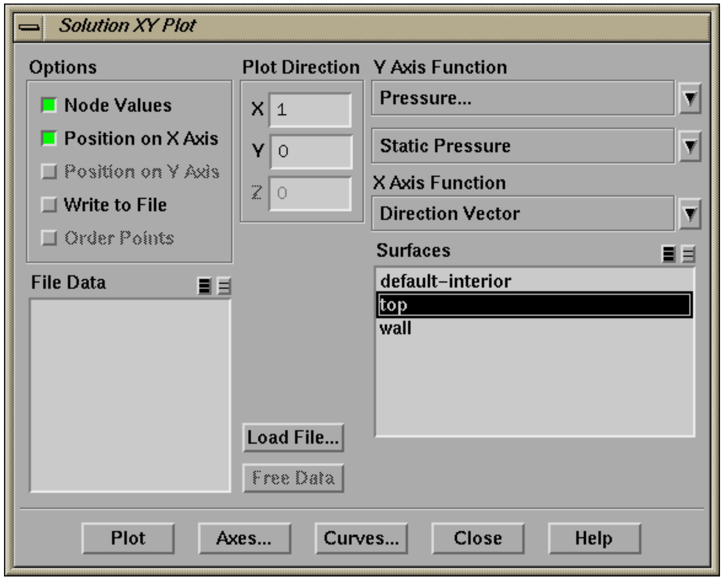
Min: 0
Max: 0

Levels: 20
Setup: 1

Surfaces:
default-interior
top
wall

Surface Name Pattern

Surface Types:
axis
clip-surf
exhaust-fan
fan



Solution XY Plot

Options

☑ Node Values
☑ Position on X Axis
☐ Position on Y Axis
☐ Write to File
☐ Order Points

Plot Direction
X: 1
Y: 0
Z: 0

Y Axis Function: Pressure...
Static Pressure

X Axis Function: Direction Vector

File Data

Surfaces:
default-interior
top
wall

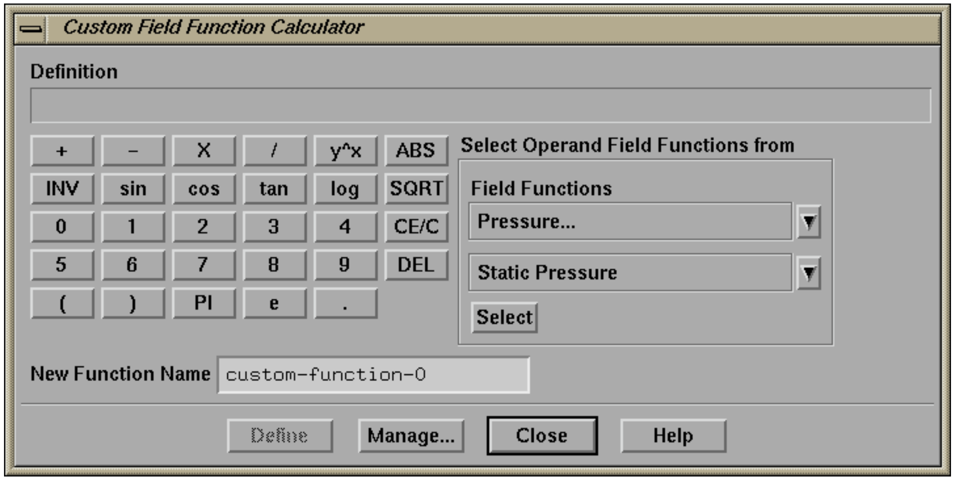
Detailed post-processing

Define additional quantities

Define plotting lines, planes and surfaces

Compute integral/averaged quantities

Define → Custom Field Function

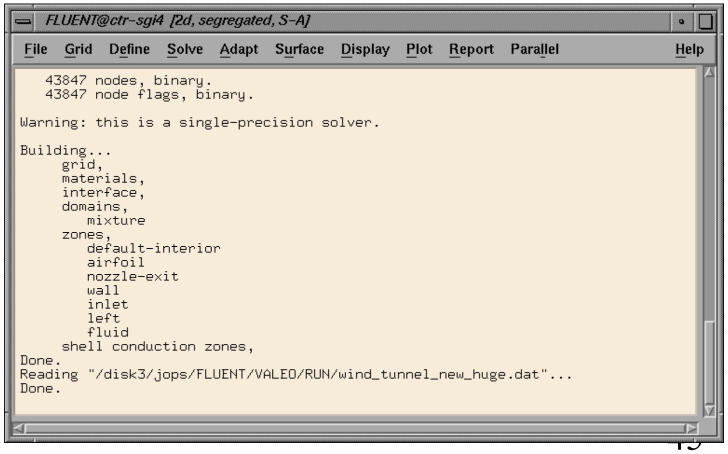


New Function Name: custom-function-0

Fluent GUI - Summary

File: I/O
Grid: Modify (translate/scale/etc.), Check
Define: Models (solver type/multiphase/etc.), Material (fluid properties),
Boundary conditions
Solve: Discretization, Initial Condition, Convergence Monitors
Adapt: Grid adaptation, Patch marking
Surface: Create zones (postprocessing/monitors)
Display: Postprocessing (View/Countors/Streamlines)
Plot: XY Plots, Residuals
Report: Summary, Integral
Parallel: Load Balancing, Monitors

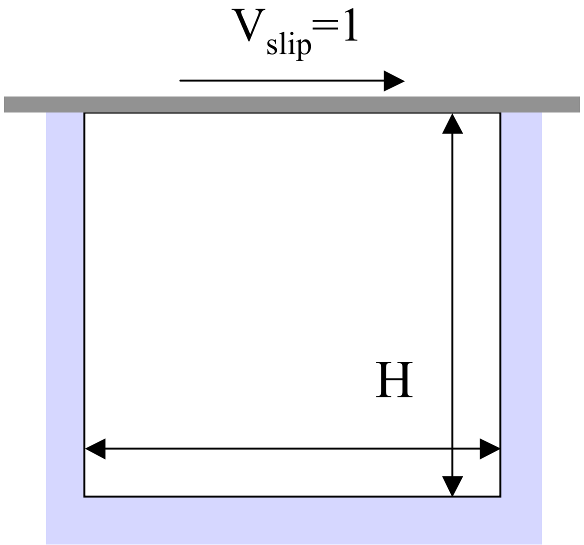
Typical simulation →



```
   43847 nodes, binary.
   43847 node flags, binary.

Warning: this is a single-precision solver.

Building...
     grid,
     materials,
     interface,
     domains,
        mixture
     zones,
        default-interior
        airfoil
        nozzle-exit
        wall
        inlet
        left
        fluid
     shell conduction zones,
Done.
Reading "/disk3/jops/FLUENT/VALEO/RUN/wind_tunnel_new_huge.dat"...
Done.
```

Example – Driven cavity

Classical test-case for incompressible flow solvers



Problem set-up

Material Properties:
$\rho = 1 kg/m^3$
$\mu = 0.001 kg/ms$

Reynolds number:
H = 1m, V_{slip}= 1m/s
$Re = \rho V_{slip} H/\mu = 1{,}000$

Boundary Conditions:
Slip wall ($u = V_{slip}$) on top
No-slip walls the others

Initial Conditions:
$u = v = p = 0$

Convergence Monitors:
Averaged pressure and friction on the no-slip walls

Solver Set-Up

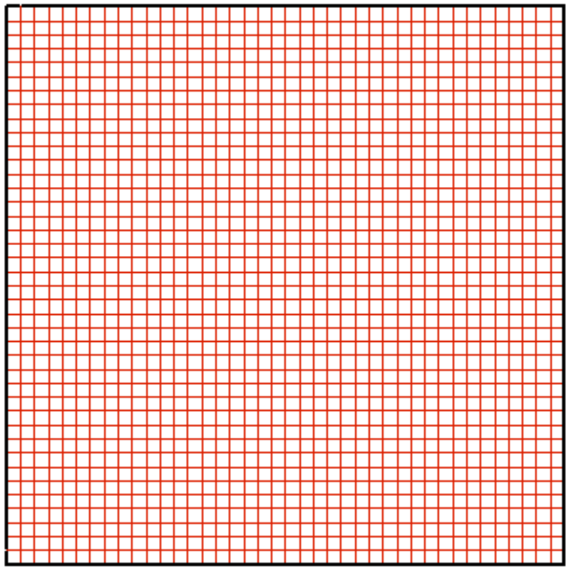
Segregated Solver

Discretization:
2nd order upwind
SIMPLE

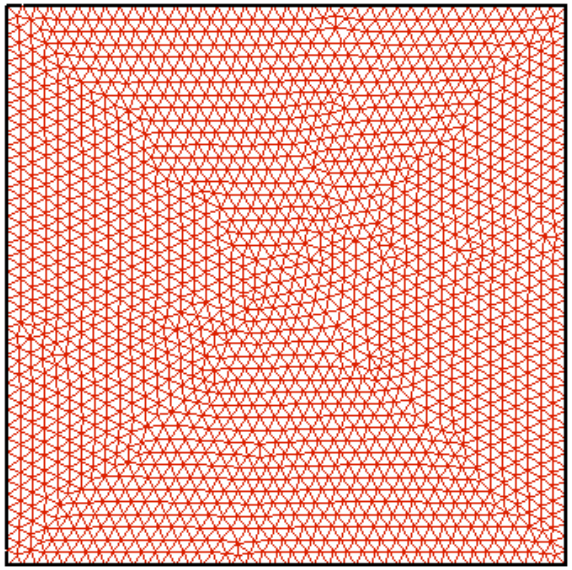
Multigrid
V-Cycle

Example – Driven cavity

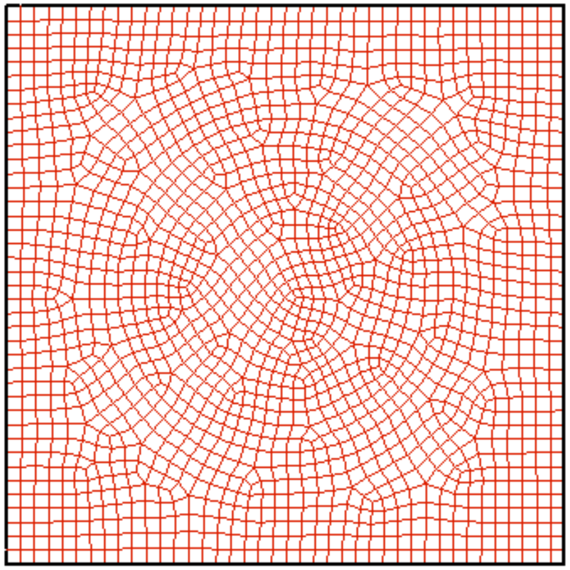
The effect of the meshing scheme



Quad-Mapping 1600 cells



Tri-Paving 3600 cells

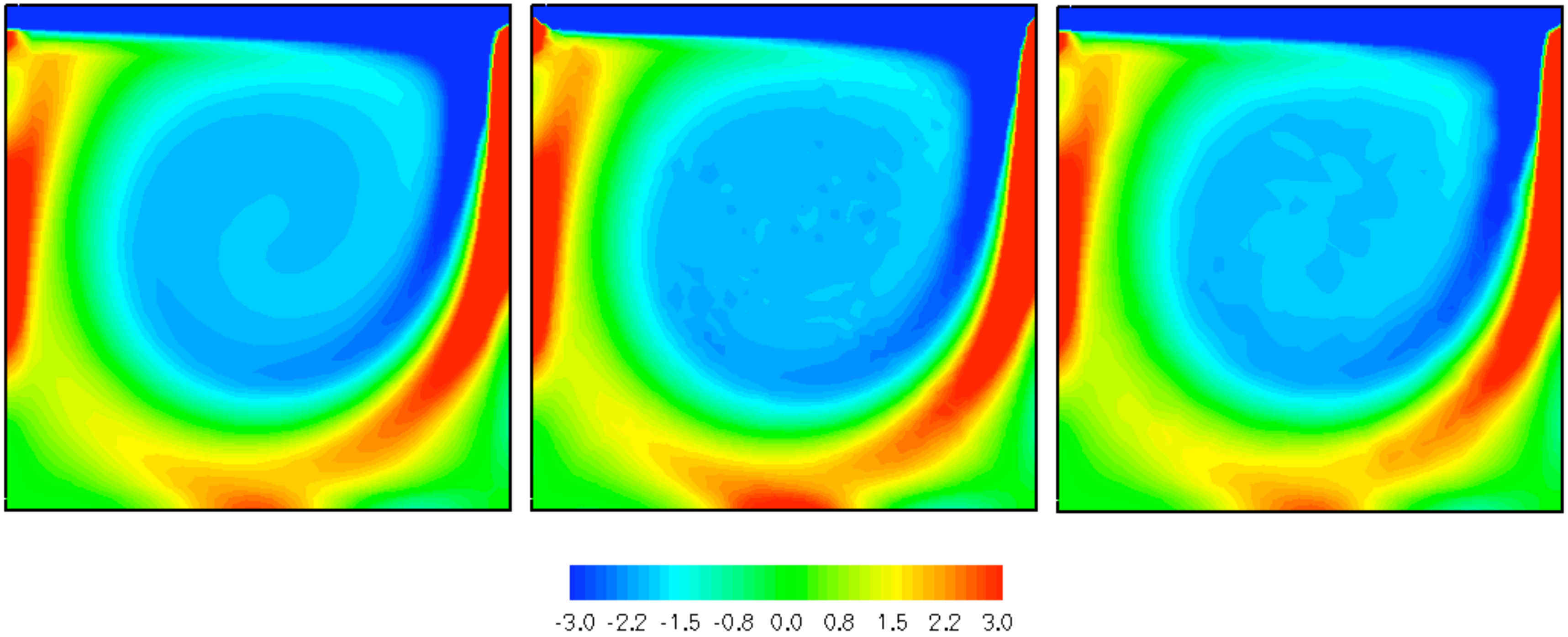


Quad-Paving 1650 cells

Edge size on the boundaries is the same

Example – Driven cavity

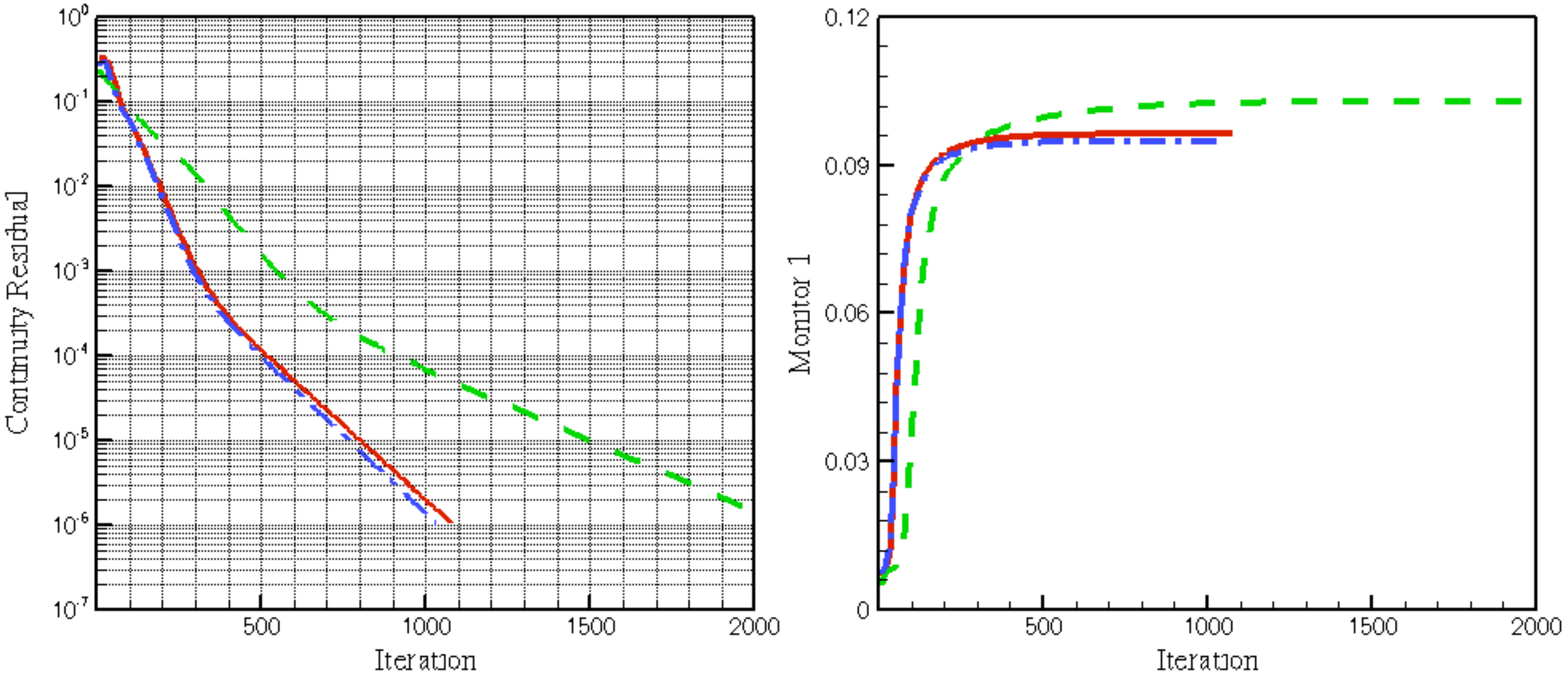
The effect of the meshing scheme – Vorticity Contours



Quad-Mapping 1600 cells Tri-Paving 3600 cells Quad-Paving 1650 cells

Example – Driven cavity

The effect of the meshing scheme – Convergence



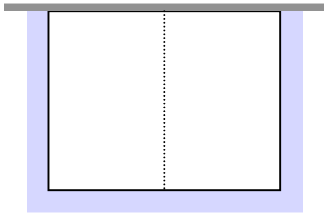
Quad-Mapping 1600 cells

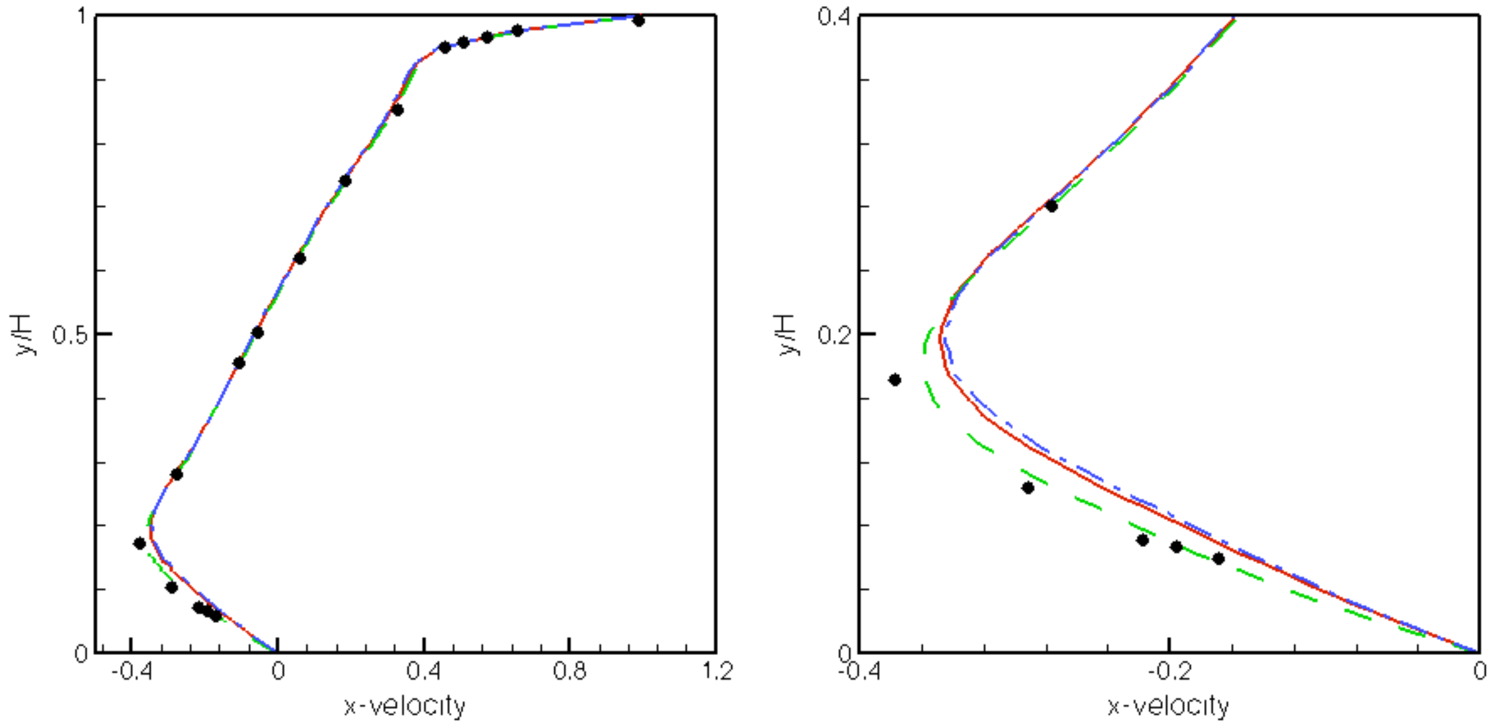
Tri-Paving 3600 cells

Quad-Paving 1650 cells

Example – Driven cavity

The effect of the meshing scheme
x-velocity component in the middle of the cavity



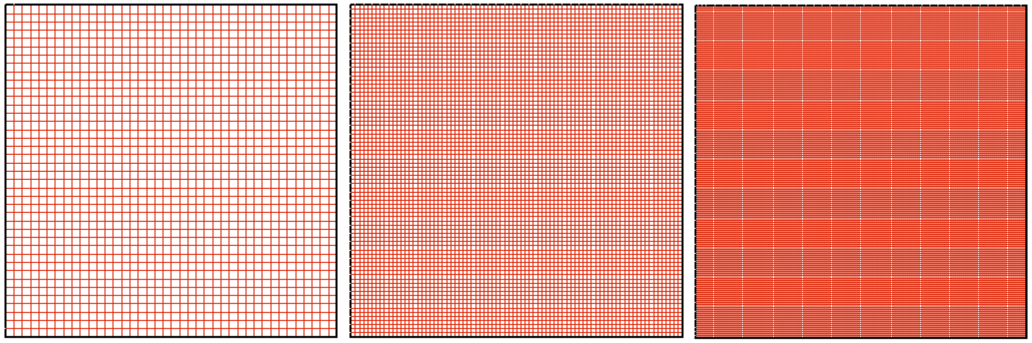


Quad-Mapping Tri-Paving Quad-Paving

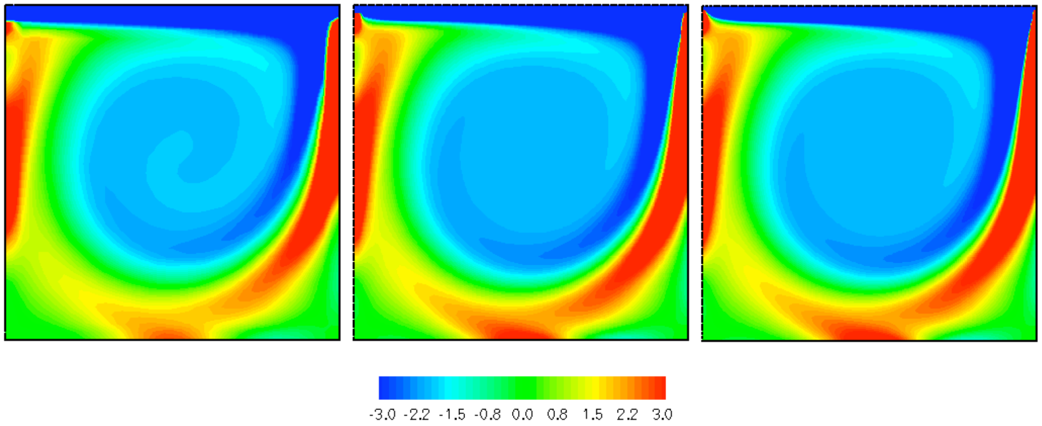
Symbols corresponds to
Ghia *et al.*, 1982

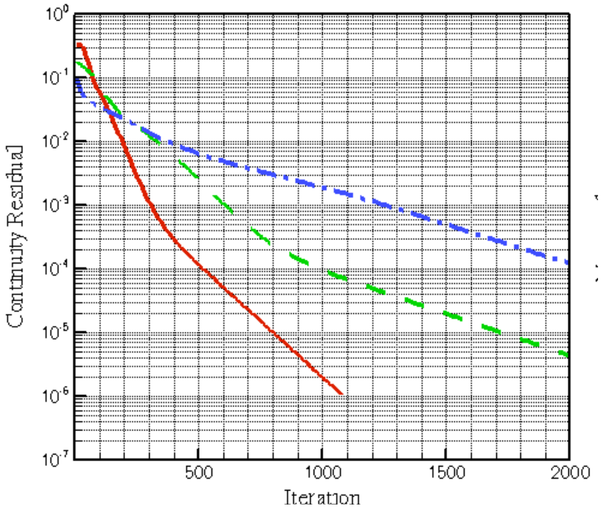
Example – Driven cavity

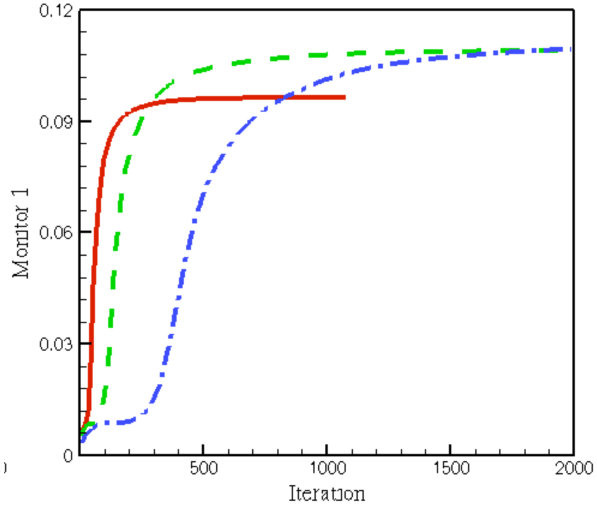
Grid Sensitivity – Quad Mapping Scheme



Vorticity Contours







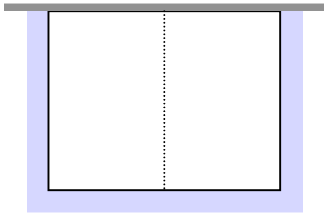
1600 cells ━━━

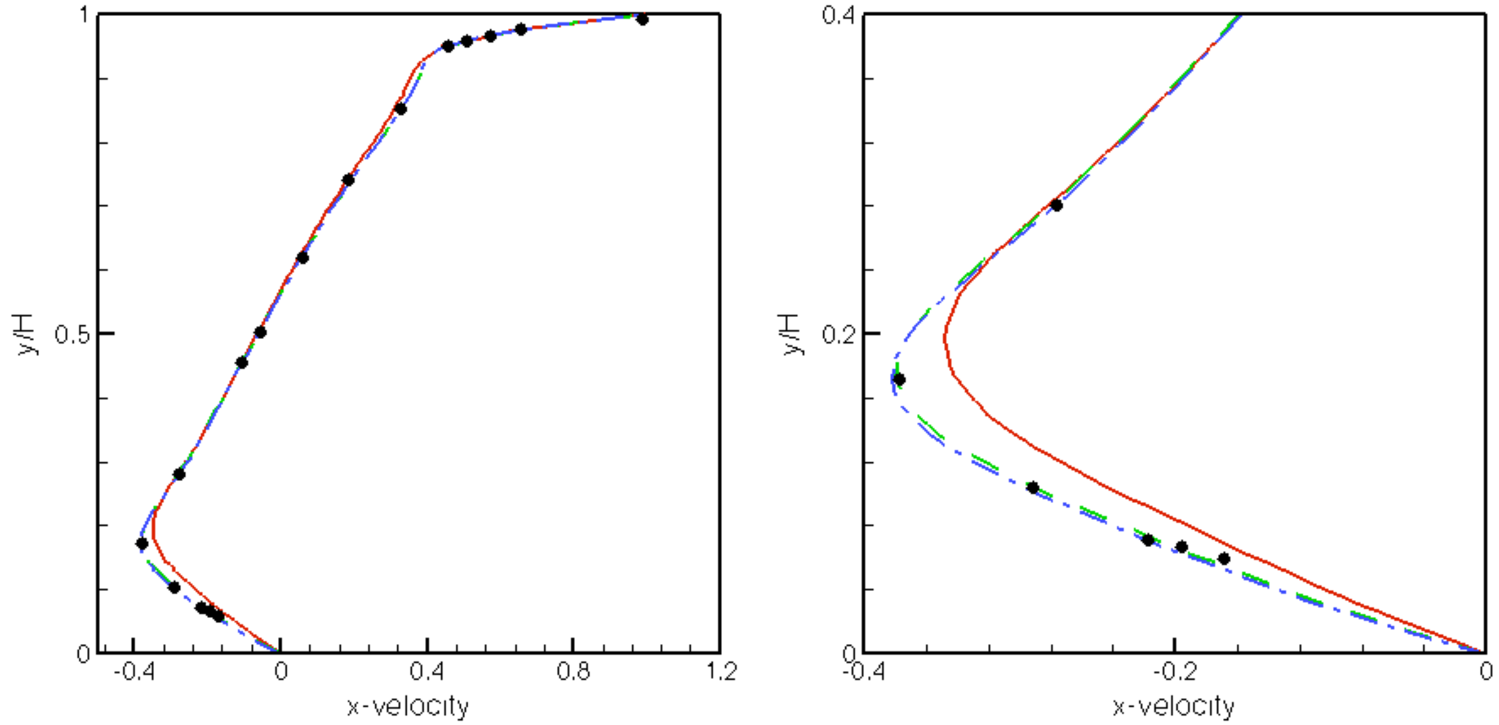
6400 cells ━ ━ ━

25600 cells ━ · ━

Example – Driven cavity

Grid Sensitivity – Quad Mapping Scheme
x-velocity component in the middle of the cavity





1600 cells

6400 cells

25600 cells

Symbols corresponds to Ghia *et al.,* 1982

How to verify the accuracy?

Define a reference solution (analytical or computed on a very fine grid)
Compute the solution on successively refined grids
Define the error as the deviation of the current solution from the reference
Compute error norms
Plot norms vs. grid size (the slope of the curve gives the order of accuracy)

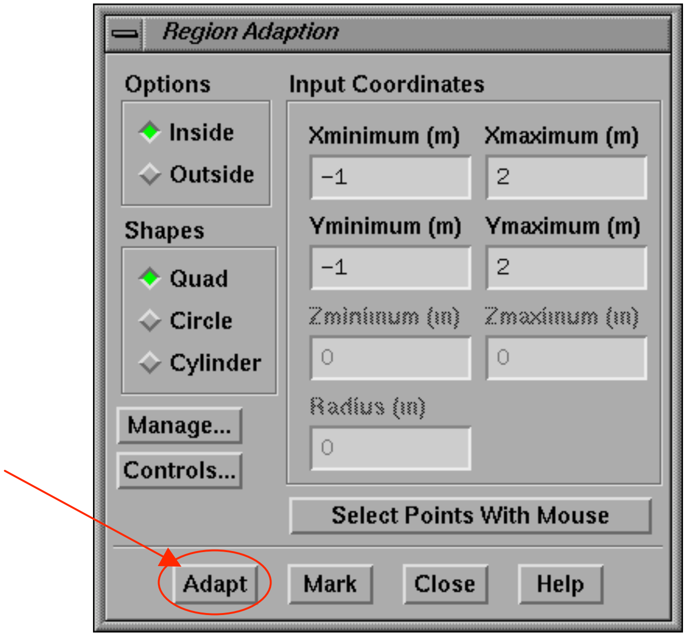
Problems with unstructured grids:

1) Generation of a suitable succession of grids
2) Definition of the grid size

Generation of successively refined grid

1) Modify grid dimensions in GAMBIT and regenerate the grid
2) Split all the cells in FLUENT

Adapt → Region → Adapt



Region Adaption

Options
● Inside
○ Outside

Shapes
● Quad
○ Circle
○ Cylinder

Input Coordinates

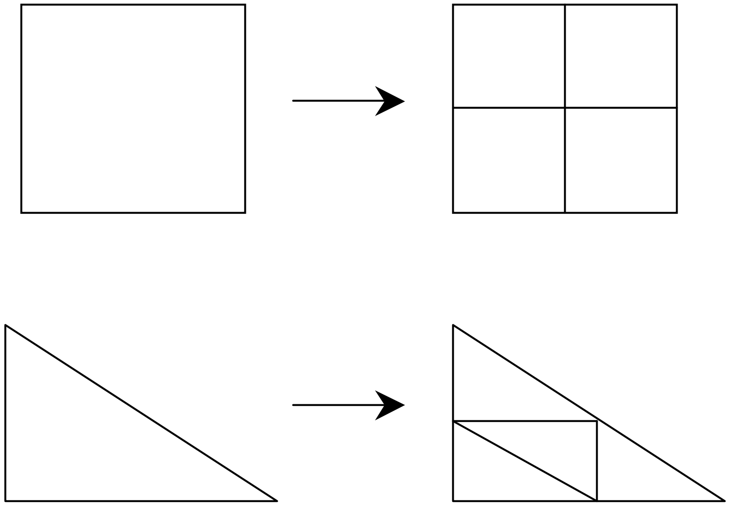
Xminimum (m)	Xmaximum (m)
-1	2

Yminimum (m)	Ymaximum (m)
-1	2

Zminimum (m)	Zmaximum (m)
0	0

Radius (m): 0

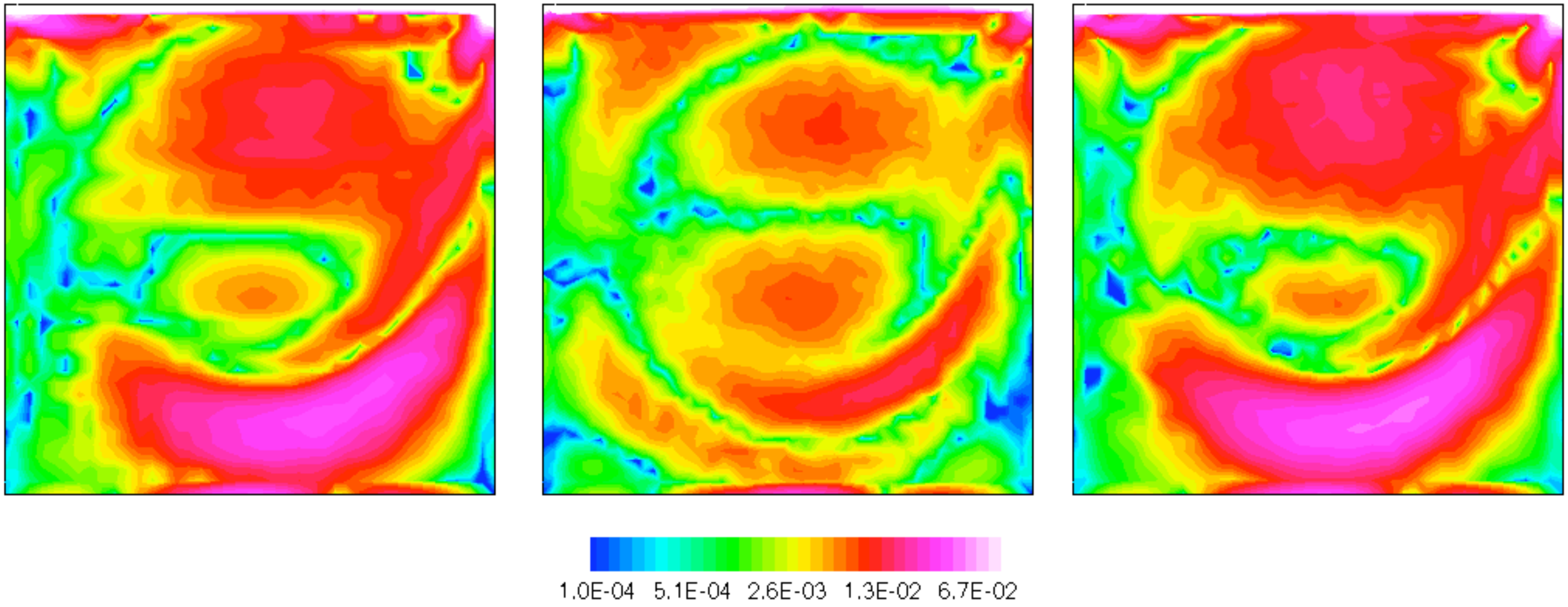
The region MUST contain the entire domain



Element shape & metric properties are preserved

Driven Cavity - Error evaluation

Reference solution computed on a 320x320 grid (~100,000 cells)
Reference solution interpolated on coarse mesh to evaluate local errors



Quad-Mapping Tri-Paving Quad-Paving

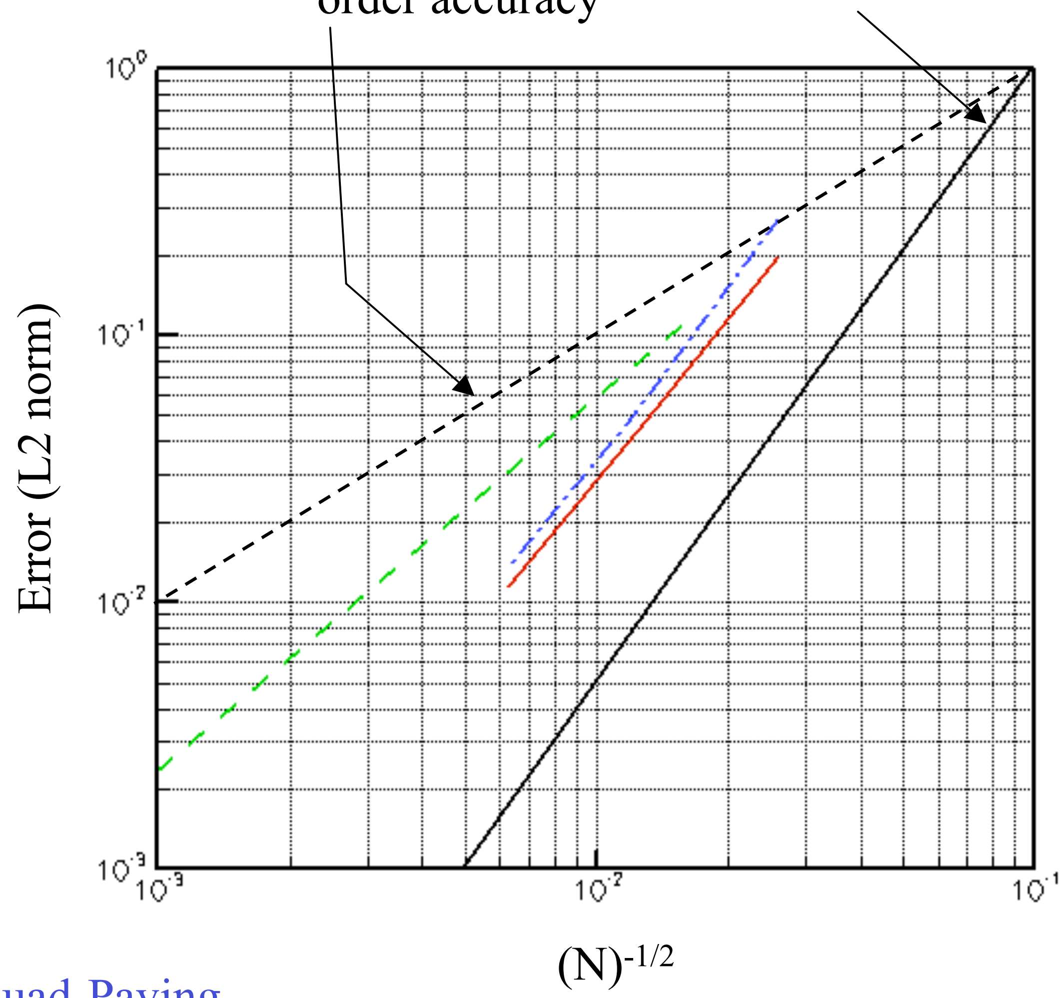
Note that the triangular grid has more than twice as many grid cells

Driven Cavity – Accuracy evaluation

Quad and Pave meshing schemes yield very similar accuracy (close to 2nd order)

Tri meshing scheme yields Slightly higher errors and lower accuracy

Note that the definition of Δx is questionable (a change will only translate the curves not change the slope)



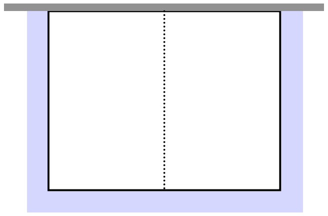
Nominal 1st order accuracy

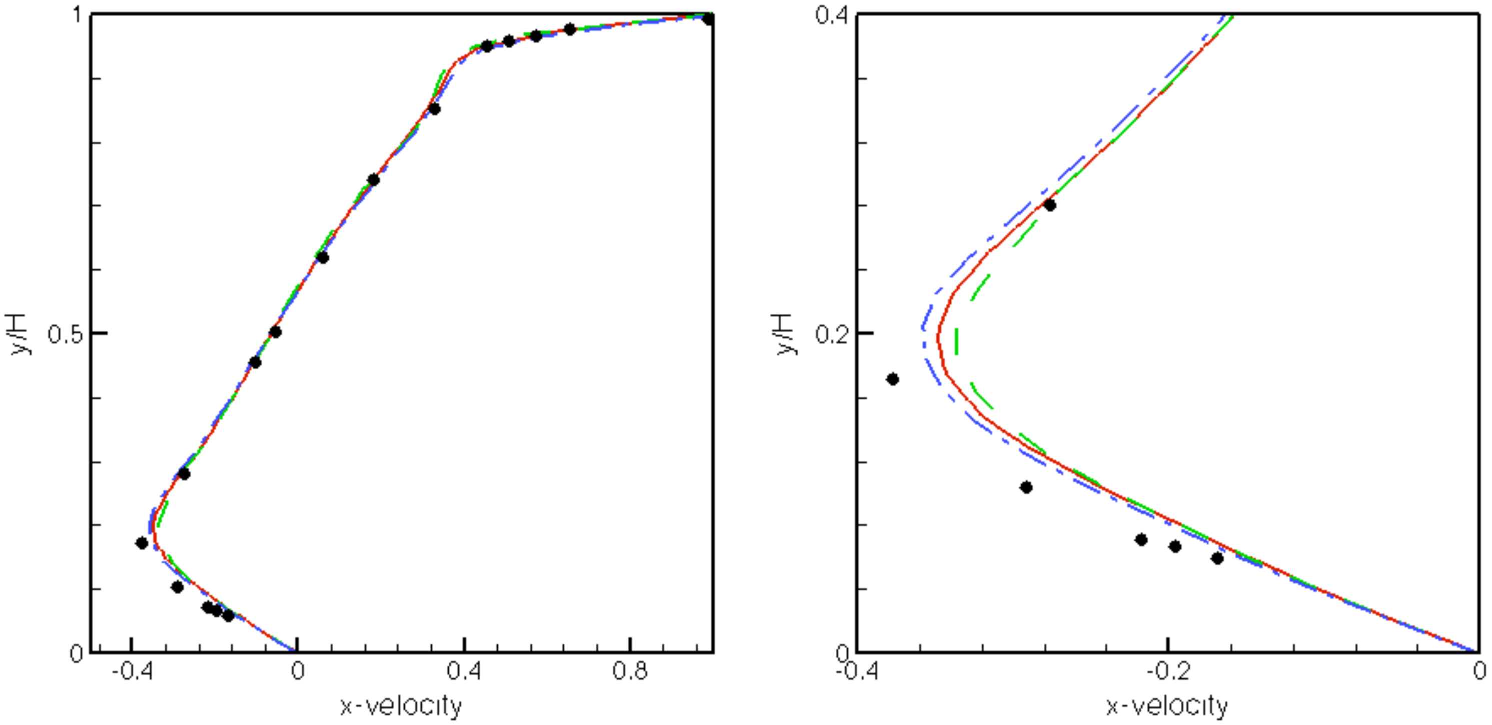
Nominal 2nd order accuracy

Quad-Mapping Tri-Paving Quad-Paving

Driven Cavity – Fluent vs. other CFD codes

Quad Mapping Scheme (1600 cells)
x-velocity component in the middle of the cavity





FLUENT

StarCD

NASA INS2D

Symbols corresponds to
Ghia *et al.*, 1982

Techniques for the incompressible NS equations

Pressure correction schemes

Artificial compressibility approach

Vorticity-streamfunction formulation

Density-based approach

Techniques for the incompressible NS equations

Vorticity-streamfunction approach

It is effectively a change-of-variables; introducing the streamfunction and the vorticity vector the continuity is automatically satisfied and the pressure disappears (if needed the solution of a Poisson-like equation is still required). *It is advantageous in 2D because it requires the solution of only two PDEs but the treatment of BCs is difficult. In addition in 3D the PDEs to be solved are six*

Artificial compressibility approach

A time-derivative (of pressure) is added to the continuity equation with the goal of transforming the incompressible NS into a hyperbolic system and then to apply schemes suitable for compressible flows. *The key is the presence of a user-parameter* β *(related to the artificial speed of sound) that determines the speed of convergence to steady state*

Density-based solvers for the NS equations

The equation are written in compressible form and, for low Mach numbers, the flow is effectively incompressible

$$\frac{\partial}{\partial t}\int_{\Omega}\rho d\Omega+\int_{S}\rho(\vec{V}\cdot\vec{n}_S)dS=0$$

$$\frac{\partial}{\partial t}\int_{\Omega}(\rho\vec{V})d\Omega+\int_{S}(\rho\vec{V})(\vec{V}\cdot\vec{n}_S)dS+\int_{S}(\vec{\vec{\tau}}\cdot\vec{n})dS+\int_{S}(p\vec{n})dS=0$$

$$\frac{\partial}{\partial t}\int_{\Omega}\rho E d\Omega+\int_{S}\rho\vec{V}E dS+\int_{S}(\vec{q}\cdot\vec{n})dS=0$$

The energy equation is added to link pressure and density through the equation of state

$$E=H-p/\rho \qquad H=h+|\mathbf{v}|^2/2$$

In compact (vector) form:

$$\frac{\partial}{\partial t}\int_{V}\mathbf{W}\,dV+\oint[\mathbf{F}-\mathbf{G}]\cdot d\mathbf{A}=\int_{V}\mathbf{H}\,dV$$

$$\mathbf{W}=\begin{Bmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ \rho E\end{Bmatrix},\ \mathbf{F}=\begin{Bmatrix}\rho\mathbf{v}\\ \rho\mathbf{v}u+p\hat{\mathbf{i}}\\ \rho\mathbf{v}v+p\hat{\mathbf{j}}\\ \rho\mathbf{v}w+p\hat{\mathbf{k}}\\ \rho\mathbf{v}E+p\mathbf{v}\end{Bmatrix},\ \mathbf{G}=\begin{Bmatrix}0\\ \tau_{xi}\\ \tau_{yi}\\ \tau_{zi}\\ \tau_{ij}v_j+\mathbf{q}\end{Bmatrix}$$

Density-based solvers for the NS equations

Stiffness occurs because of the disparity between fluid velocity and speed of sound (infinite in zero-Mach limit)

The equations are solved in terms of the primitive variables

$$\frac{\partial \mathbf{W}}{\partial \mathbf{Q}} \frac{\partial}{\partial t} \int_V \mathbf{Q}\, dV + \oint [\mathbf{F} - \mathbf{G}] \cdot d\mathbf{A} = \int_V \mathbf{H}\, dV$$

$$\vec{Q} = \left[p, \vec{V}, T\right]^T$$

Note that the continuity becomes (again) an evolution equation for the pressure

where

$$\frac{\partial \mathbf{W}}{\partial \mathbf{Q}} = \begin{bmatrix} \rho_p & 0 & 0 & 0 & \rho_T \\ \rho_p u & \rho & 0 & 0 & \rho_T u \\ \rho_p v & 0 & \rho & 0 & \rho_T v \\ \rho_p w & 0 & 0 & \rho & \rho_T w \\ \rho_p H - \delta & \rho u & \rho v & \rho w & \rho_T H + \rho C_p \end{bmatrix}$$

$$\rho_p = \left.\frac{\partial \rho}{\partial p}\right|_T, \quad \rho_T = \left.\frac{\partial \rho}{\partial T}\right|_p$$

$\delta = 1$ ideal gas

$\delta = 0$ incompressible fluid

Density-based solvers for the NS equations

The time derivative is modified (preconditioned) to force all the eigenvalues to be of the same order (similar to the artificial compressibility approach)

$$\Gamma \frac{\partial}{\partial t} \int_V \mathbf{Q}\, dV + \oint [\mathbf{F} - \mathbf{G}] \cdot d\mathbf{A} = \int_V \mathbf{H}\, dV$$

$$\Gamma = \begin{bmatrix} \Theta & 0 & 0 & 0 & \rho_T \\ \Theta u & \rho & 0 & 0 & \rho_T u \\ \Theta v & 0 & \rho & 0 & \rho_T u \\ \Theta w & 0 & 0 & \rho & \rho_T u \\ \Theta H - \delta & \rho u & \rho v & \rho w & \rho_T H + \rho C_p \end{bmatrix} \neq \frac{\partial \mathbf{W}}{\partial \mathbf{Q}} = \begin{bmatrix} \rho_p & 0 & 0 & 0 & \rho_T \\ \rho_p u & \rho & 0 & 0 & \rho_T u \\ \rho_p v & 0 & \rho & 0 & \rho_T v \\ \rho_p w & 0 & 0 & \rho & \rho_T w \\ \rho_p H - \delta & \rho u & \rho v & \rho w & \rho_T H + \rho C_p \end{bmatrix}$$

The eigenvalues of Γ are

$$u, \; u, \; u, \; u' + c', \; u' - c'$$

where

$$\begin{aligned} u &= \mathbf{v} \cdot \hat{n} \\ u' &= u\,(1 - \alpha) \\ c' &= \sqrt{\alpha^2 u^2 + U_r^2} \end{aligned}$$

$$\Theta = \left(\frac{1}{U_r^2} - \frac{\rho_T}{\rho C_p} \right)$$

$$\begin{aligned} \alpha &= \left(1 - \beta U_r^2\right)/2 \\ \beta &= \left(\rho_p + \frac{\rho_T}{\rho C_p} \right) \end{aligned}$$

Density-based solvers for the NS equations

Limiting cases

Compressible flows
(ideal gas):

$$U_r = c$$

$$\beta = (\gamma RT)^{-1} = 1/c^2$$

$$\Gamma = \frac{\partial \mathbf{W}}{\partial \mathbf{Q}}$$

Incompressible flows (ideal gas):

$$U_r \to 0$$

$$\alpha \to 1/2$$

All eigenvalues
are comparable

Incompressible fluids:

$$\beta = 0$$

FLUENT density-based solver

Explicit Scheme

Multistage Runge-Kutta scheme

$$\begin{aligned} \mathbf{Q}^0 &= \mathbf{Q}^n \\ \Delta\mathbf{Q}^i &= -\alpha_i \Delta t \Gamma^{-1} \mathbf{R}^{i-1} \\ \mathbf{Q}^{n+1} &= \mathbf{Q}^m \end{aligned}$$

$$\mathbf{R}^i = \sum^{N_{\text{faces}}} \left(\mathbf{F}(\mathbf{Q}^i) - \mathbf{G}(\mathbf{Q}^i) \right) \cdot \mathbf{A} - V\mathbf{H}$$

Residual Smoothing

$$\bar{R}_i = R_i + \epsilon \sum (\bar{R}_j - \bar{R}_i)$$

Multigrid acceleration

FLUENT density-based solver

Implicit Scheme

Euler (one-step) implicit with Newton-type linearization

$$\left[\mathrm{D} + \sum_{j}^{N_{\mathrm{faces}}} \mathrm{S}_{j,k}\right] \Delta \mathbf{Q}^{n+1} = -\mathbf{R}^n$$

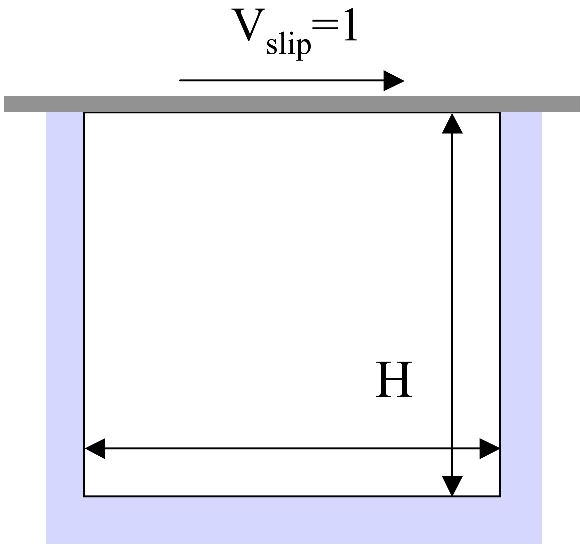
$$\begin{aligned} \mathrm{D} &= \frac{V}{\Delta t}\Gamma + \sum_{j}^{N_{\mathrm{faces}}} \mathrm{S}_{j,i} \\ \mathrm{S}_{j,k} &= \left(\frac{\partial \mathbf{F}_j}{\partial \mathbf{Q}_k} - \frac{\partial \mathbf{G}_j}{\partial \mathbf{Q}_k}\right) \end{aligned}$$

Point Gauss-Seidel iterations

Multigrid acceleration

Example – Driven cavity

Classical test-case for incompressible flow solvers



Problem set-up

Material Properties:
$\rho = 1\text{kg/m}^3$
$\mu = 0.001\text{kg/ms}$

Reynolds number:
$H = 1\text{m}, V_{slip} = 1\text{m/s}$
$Re = \rho V_{slip} H/\mu = 1{,}000$

Boundary Conditions:
Slip wall ($u = V_{slip}$) on top
No-slip walls the others

Initial Conditions:
$u = v = p = 0$

Convergence Monitors:
Averaged pressure and friction on the no-slip walls

Solver Set-Up

Coupled Solver

Discretization:
2nd order upwind
Implicit

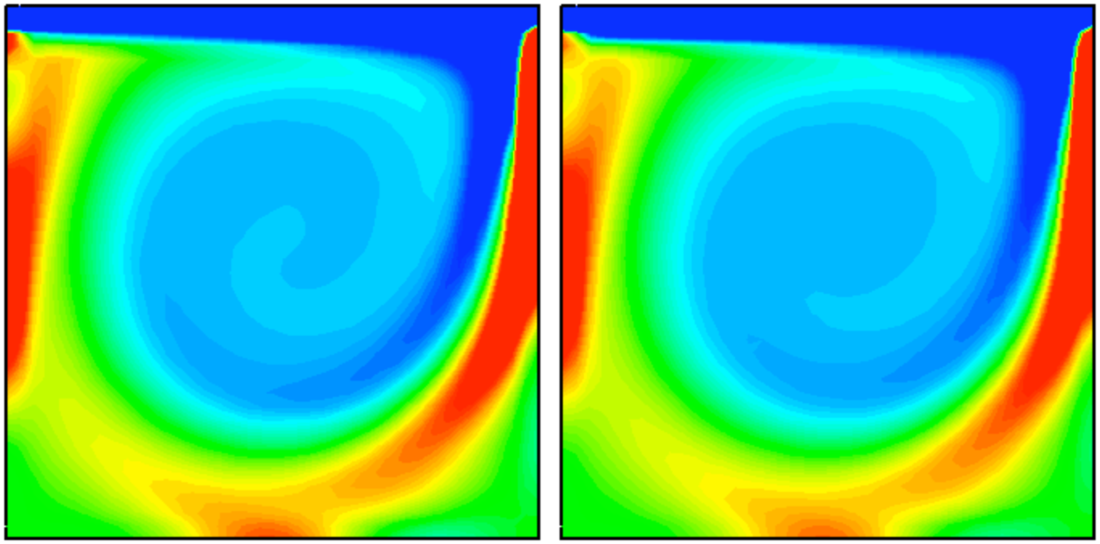
Multigrid
V-Cycle

Example – Driven cavity

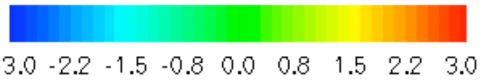
Effect of the solver - Quad mesh (1600 cells)

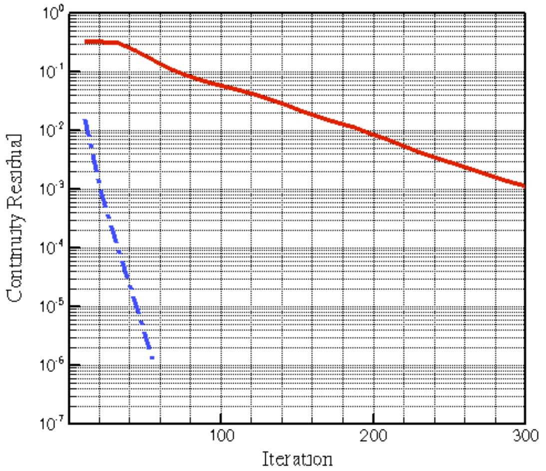
Segregated —

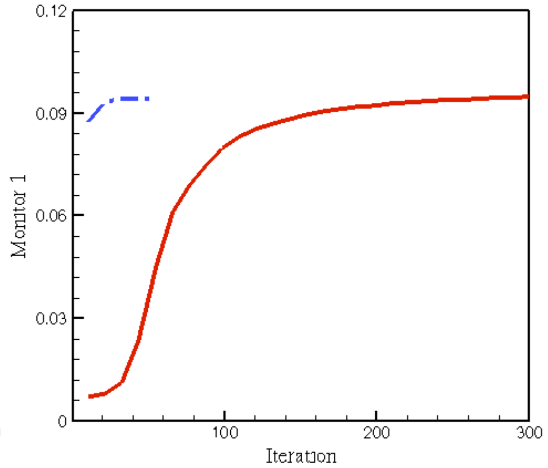
Coupled – · –



Vorticity Contours

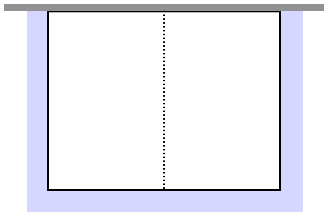


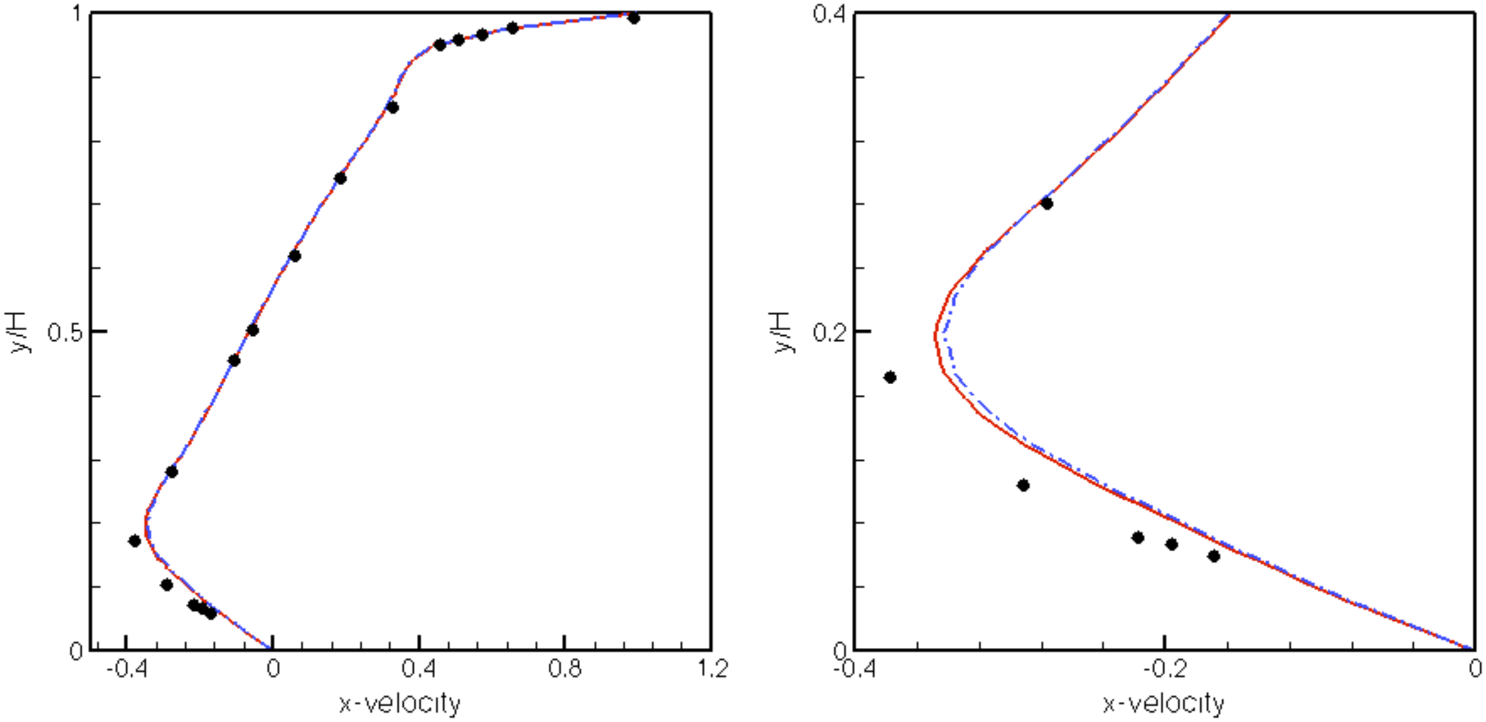




Example – Driven cavity

Effect of the solver - Quad mesh (1600 cells)
x-velocity component in the middle of the cavity





Segregated

Coupled

Symbols corresponds to Ghia *et al.*, 1982

Multigrid acceleration

Basic idea: the global error (low-frequency) on a fine grid appears as a local error (high-frequency) on coarse meshes.

Why it is important: linear system solver like Gauss-Seidel are effective in removing high-frequency errors but VERY slow for global errors. Note that, on structured, grid line-relaxation (or ADI-type) schemes can be used to improve the performance of Gauss-Seidel; on unstructured grid similar concepts are extremely difficult to implement.

Convergence Speed: number of iterations on the finest grid required to reach a given level of convergence is roughly independent on the number of grid nodes (multigrid convergence)

Two-grid scheme

1. α smoothings are performed on the fine grid to reduce the high-frequency components of the errors (pre-smoothing, αS)
2. the residual (error) is transferred to next coarser level (restriction, R)
3. γ iterations are performed on this grid level for the "correction" equation
4. the problem is transferred back to the fine grid (prolongation, P)
5. β smoothings are performed on the fine grid to remove the high-frequency errors introduced on the coarse mesh (post-smoothing, βS)

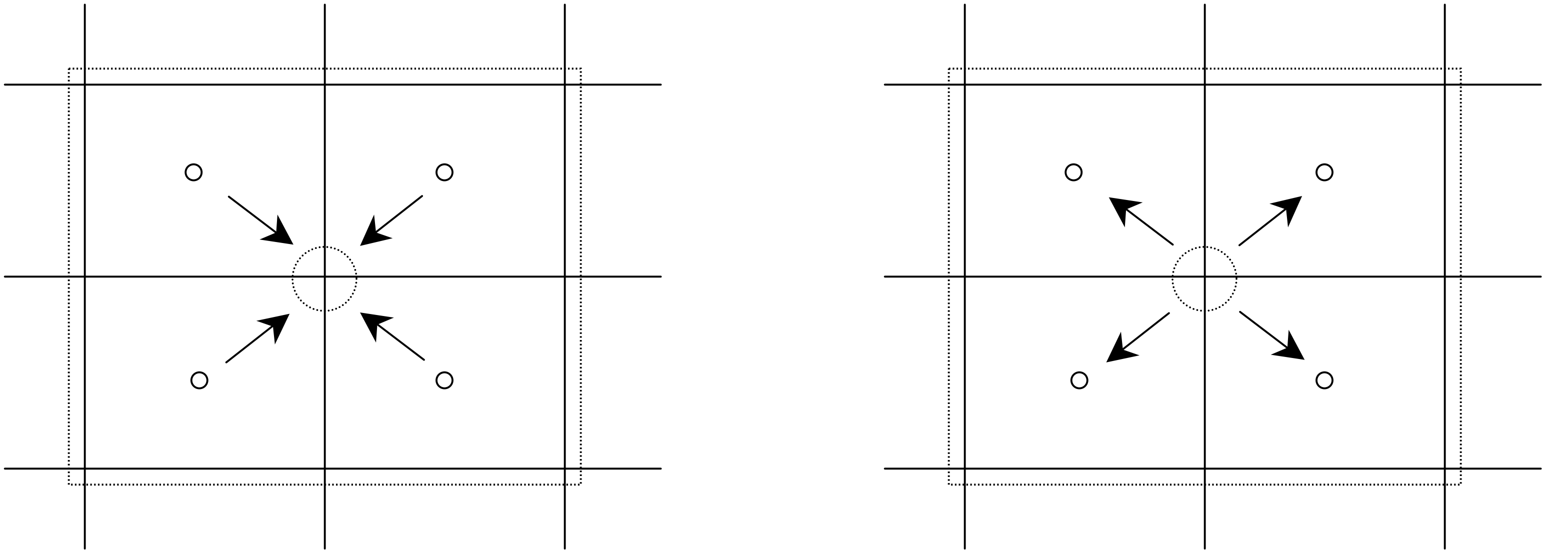
Parameters to be defined are α, β, γ

Multigrid Formalism

$\rho^h = b - A^h x^h$	After few sweeps at level h
$\rho^{2h} = R\rho^h$	Transfer (restrict) the residual
$A^{2h} e^{2h} = \rho^{2h}$	Modified system on the coarse grid
$e^h = P e^{2h}$	Transfer (prolong) the solution
$x = x^h + e^h$	Correct

Definition of the error and residual $Ax_e = b$ $\rho = b - Ax$

Restriction & Prolongation Operators



☐ Fine Level

⬚ Coarse Level

Algebraic Multigrid

The coarse levels are generated without the use of any discretization on coarse levels; *in fact no hierarchy of meshes is needed*

AMG is effectively a solver for linear systems and the restriction and prolongation operators might be viewed as means to modify (group or split) the coefficient matrix

Formally:

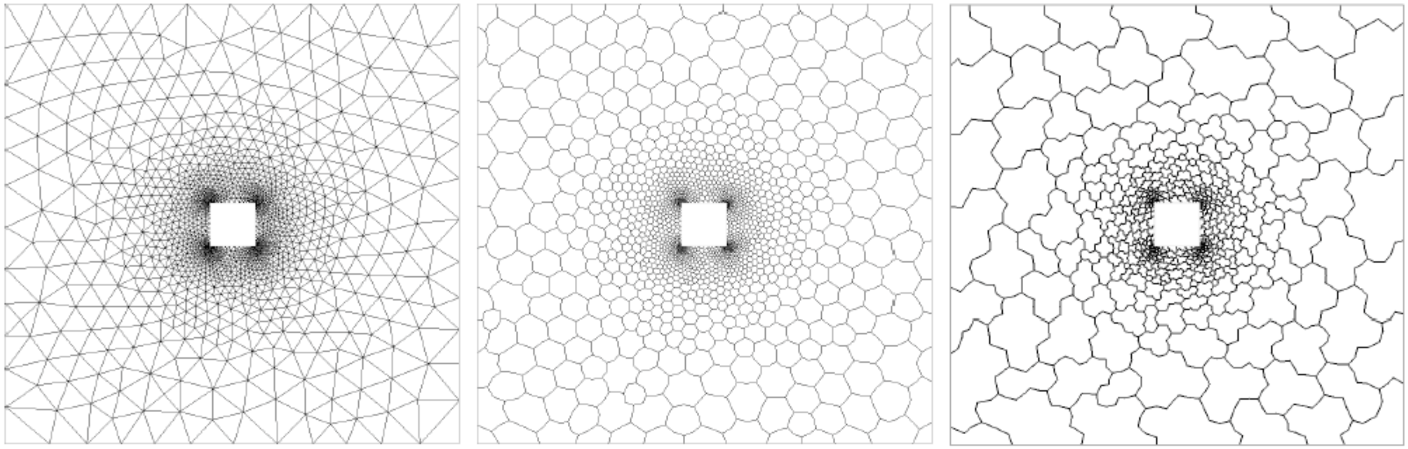
$$e^h = Pe^{2h}$$

$$A^h e^h = A^h P e^{2h} = \rho^h \qquad \left(RA^hP\right) e^{2h} = A^{2h} e^{2H} = R\rho^h$$

Geometric multigrid *should* perform better than AMG because non-linearity of the problem are retained on coarse levels (correction equation)

Multigrid for unstructured meshes

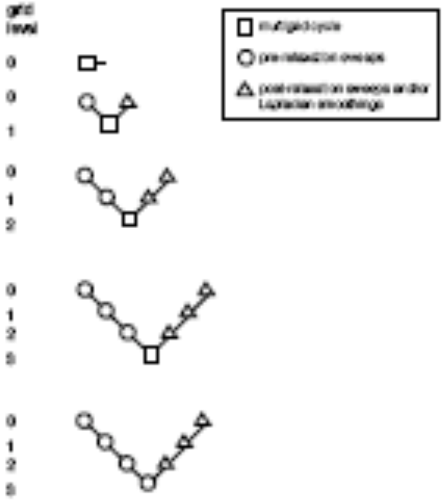
Aggregative Coarsening: fine grid cells are collected into a coarse grid element



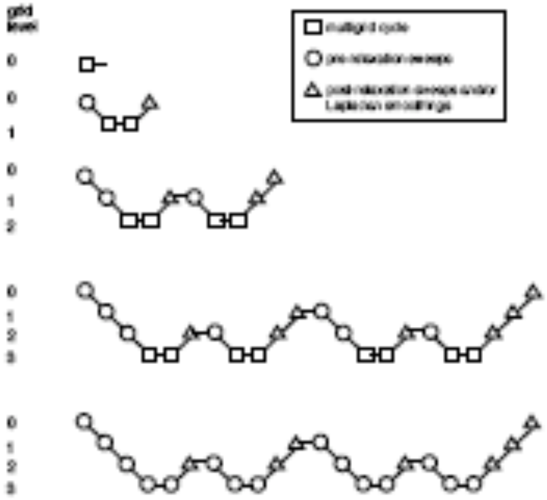
Selective Coarsening: few fine grid cells are retained on the coarser grids…

Multigrid in Fluent

Level Cycling: V, W and F (W+V)



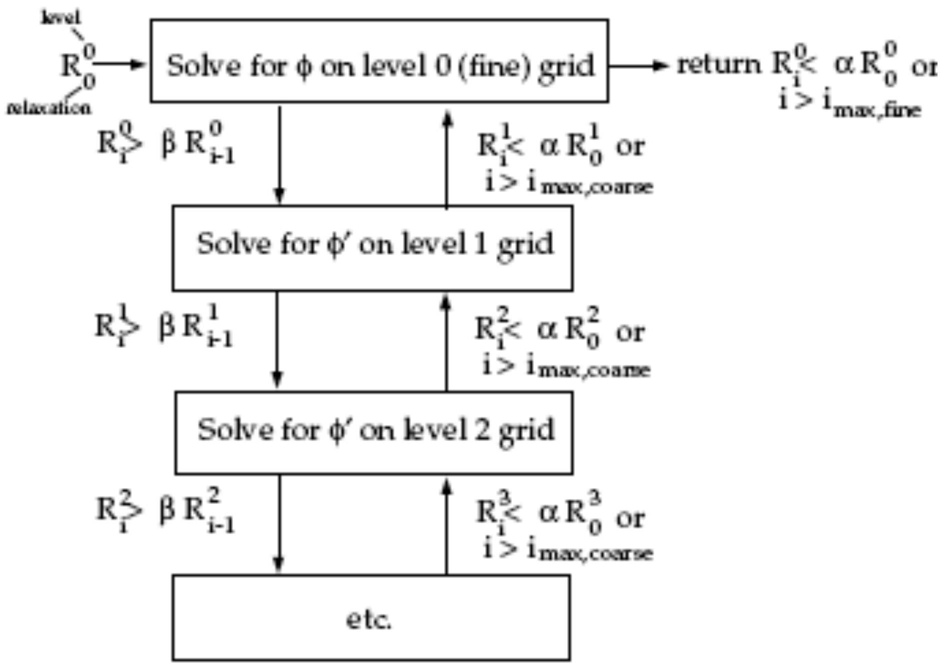
V-Cycle



W-Cycle

Multigrid in Fluent

Flexible Cycle



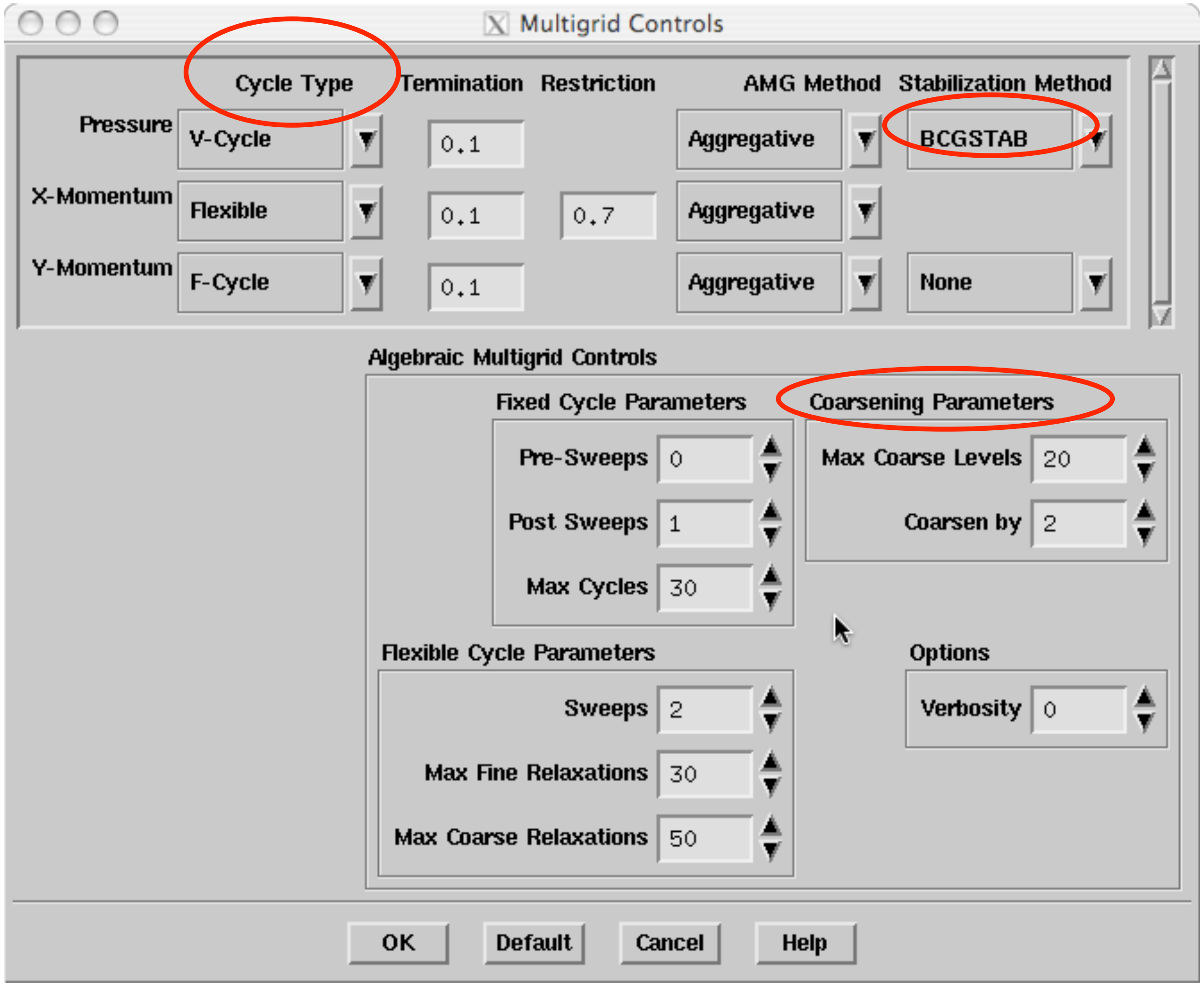
Restriction Criteria:

A coarser level is invoked as soon as the residual reduction rate is below a certain %

Termination Criteria:

The corrections are transferred to a finer level as soon as a certain residual level is reached

Multigrid in Fluent



	Cycle Type	Termination	Restriction	AMG Method	Stabilization Method
Pressure	V-Cycle	0.1		Aggregative	BCGSTAB
X-Momentum	Flexible	0.1	0.7	Aggregative	
Y-Momentum	F-Cycle	0.1		Aggregative	None

Algebraic Multigrid Controls

Fixed Cycle Parameters

Pre-Sweeps 0

Post Sweeps 1

Max Cycles 30

Coarsening Parameters

Max Coarse Levels 20

Coarsen by 2

Flexible Cycle Parameters

Sweeps 2

Max Fine Relaxations 30

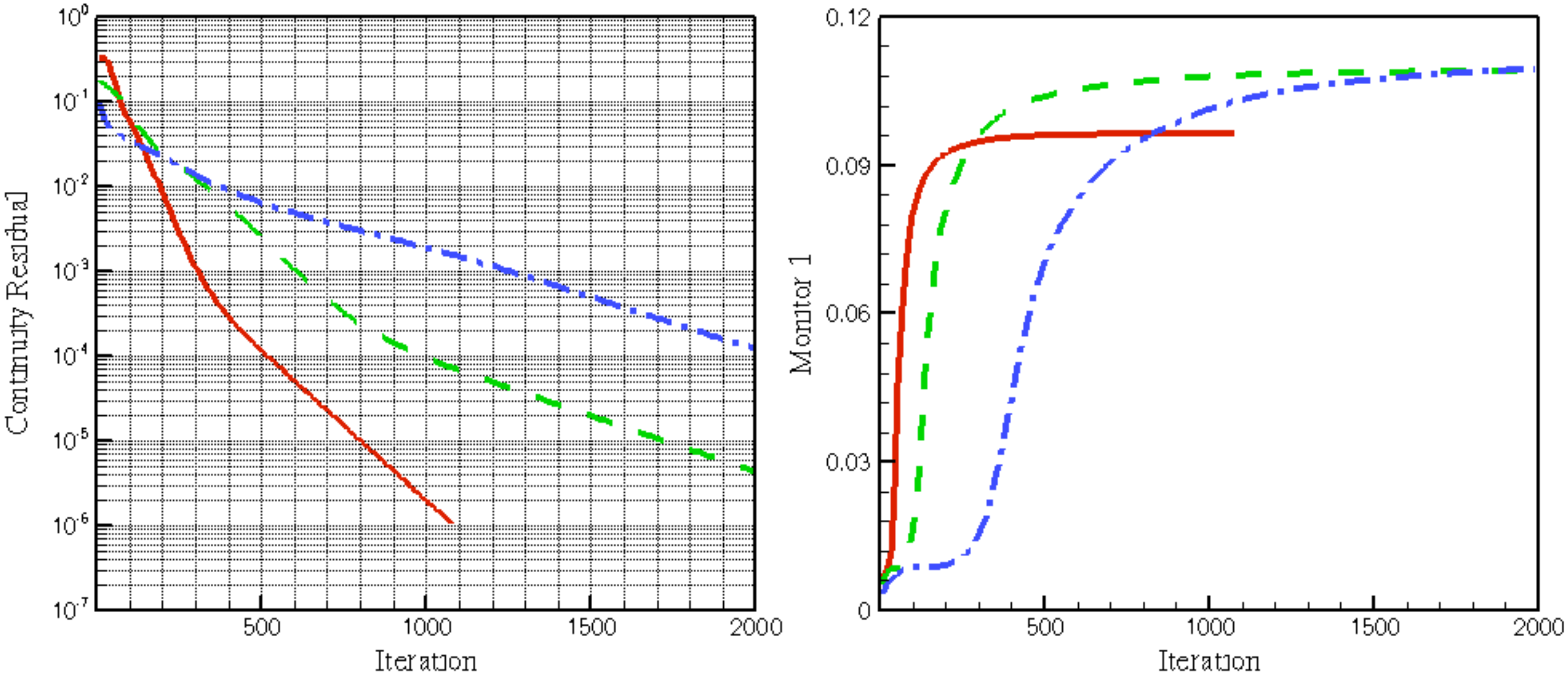
Max Coarse Relaxations 50

Options

Verbosity 0

Algebraic Multigrid Performance

Convergence for the segregated solver



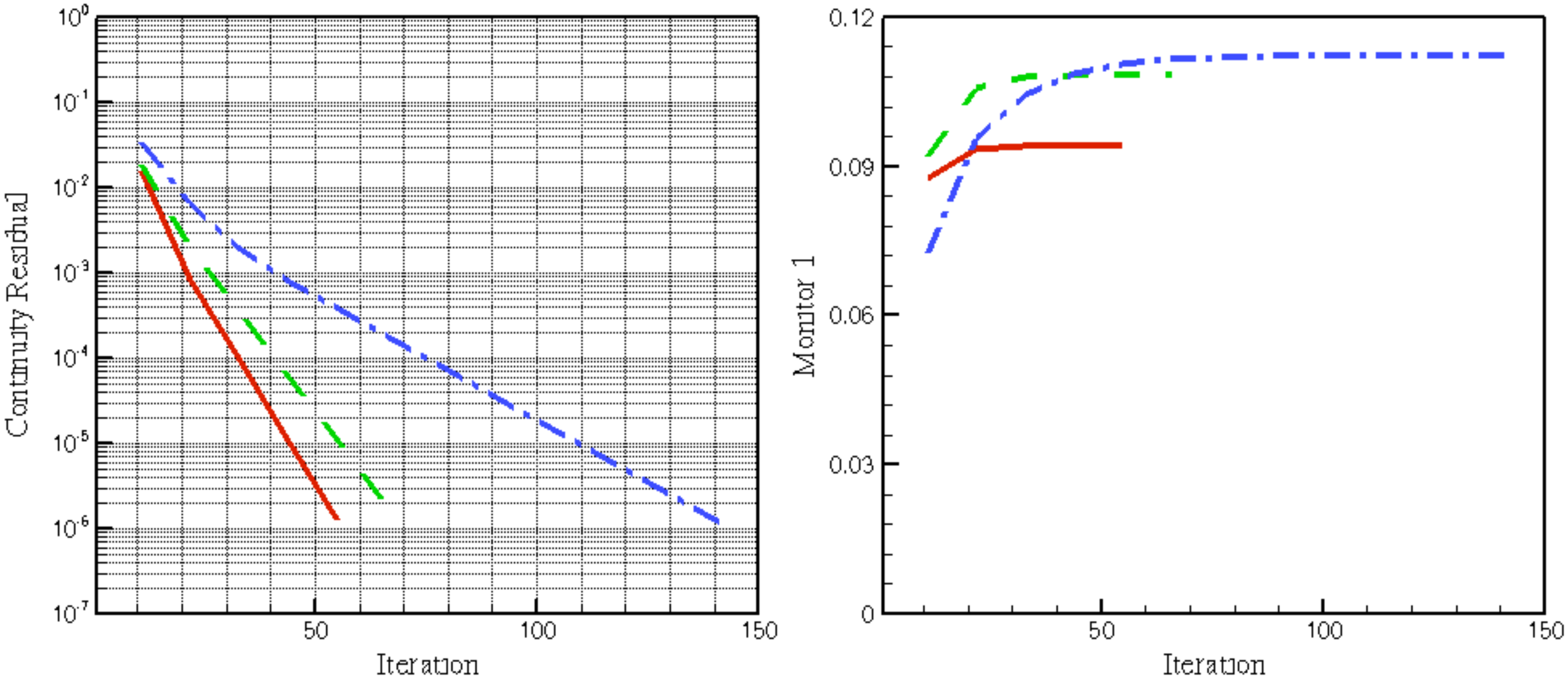
1600 cells ━━━

6400 cells ‒ ‒ ‒

25600 cells ‒ · ‒

Algebraic Multigrid Performance

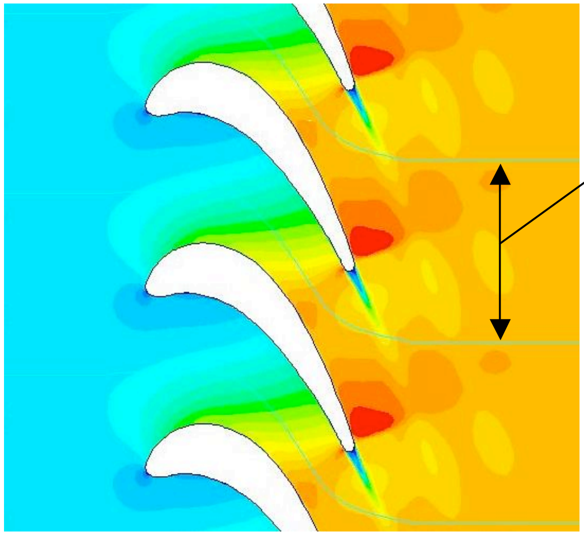
Convergence for the coupled solver

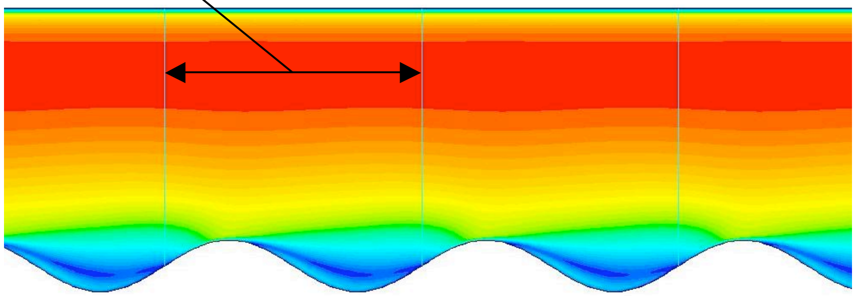


1600 cells — 6400 cells - - - 25600 cells -·-

Periodic Flows

Geometrical periodicity





Periodicity simply corresponds to matching conditions on the two boundaries

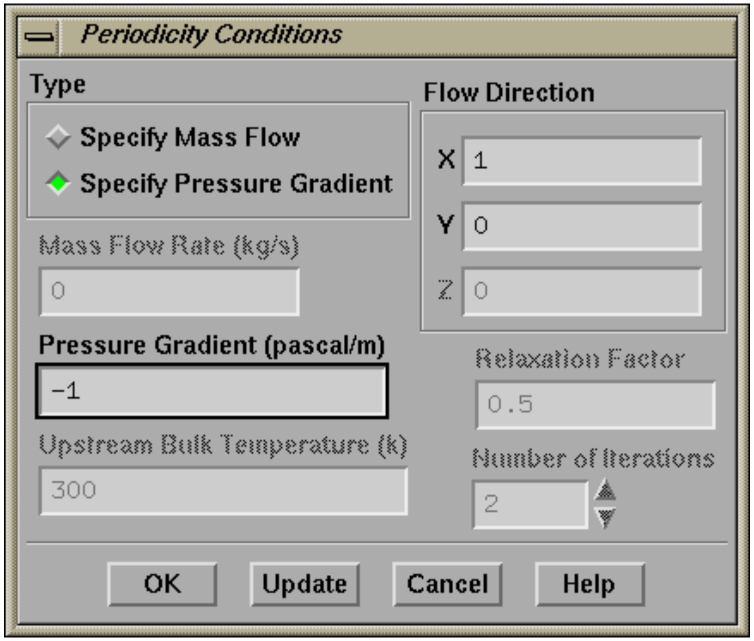
The velocity field is periodic BUT the pressure field is not. The pressure gradient drives the flow and is periodic. A pressure JUMP condition on the boundary must be specified

Periodic Flows – Set-Up

In the segregated solver periodicity can be imposed by fixing either the mass flow or the pressure drop

In the coupled solver periodicity is enforced by fixing the pressure drop

Define → Periodic Conditions



Periodicity Conditions

Type

○ Specify Mass Flow

● Specify Pressure Gradient

Mass Flow Rate (kg/s): 0

Pressure Gradient (pascal/m): -1

Upstream Bulk Temperature (k): 300

Flow Direction

X: 1

Y: 0

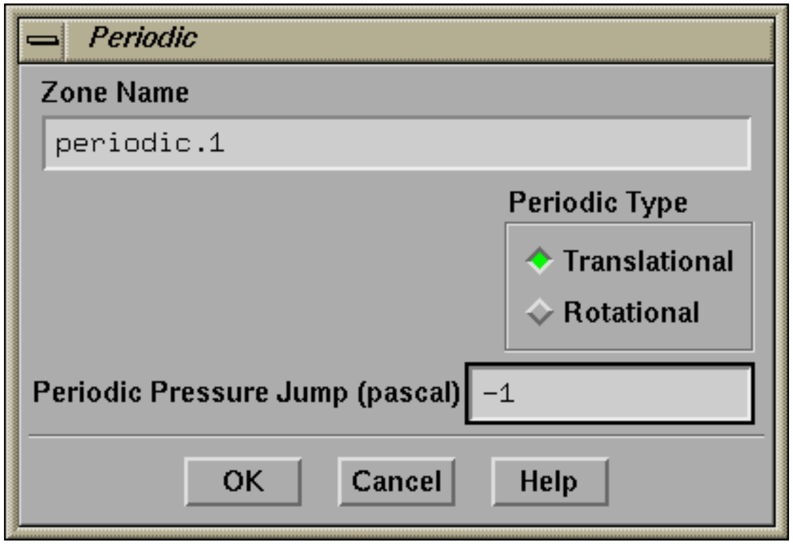
Z: 0

Relaxation Factor: 0.5

Number of Iterations: 2

Segregated solver

Define → Boundary Conditions



Periodic

Zone Name: periodic.1

Periodic Type

● Translational

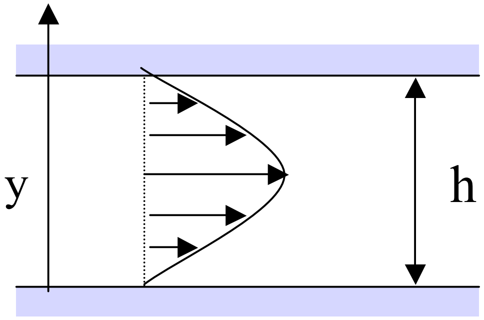
○ Rotational

Periodic Pressure Jump (pascal): -1

Coupled Solver

Periodic Flow Example – 2D channel

An analytical solution of the Navier-Stokes equations (Poiseuille flow) can be derived:



Solution in the form *u=u(y)*

The pressure drop balances the viscous drag on the walls

Navier-Stokes equations

$$\nu \frac{d^2 u}{dy^2} = \frac{1}{\rho}\frac{dp}{dx}$$

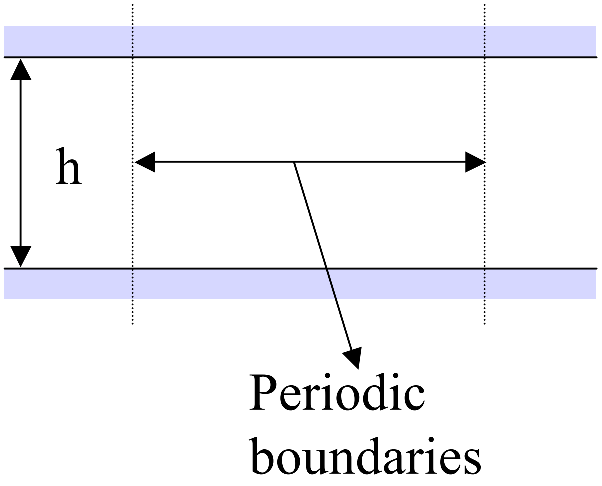
Velocity distribution in the channel

$$u = \frac{1}{2\rho\nu}\left(-\frac{dp}{dx}\right) y\,(h-y)$$

Averaged velocity

$$\overline{u} = \frac{h^2}{12\rho\nu}\left(-\frac{dp}{dx}\right)$$

Periodic Flow Example – 2D channel



Problem set-up

Material Properties:
$\rho = 1 kg/m^3$
$\mu = 0.1 kg/ms$

Reynolds number:
$h = 2m$, $V_{ave} = 1m/s$
$Re = \rho V_{slip} h/\mu = 20$

Boundary Conditions:
Periodicity $\Delta p=0.3$
No-slip top/bottom walls

Initial Conditions:
$u = 1$; $v = p = 0$

Exact solution:
$V_{ave} = 1$

Solver Set-Up

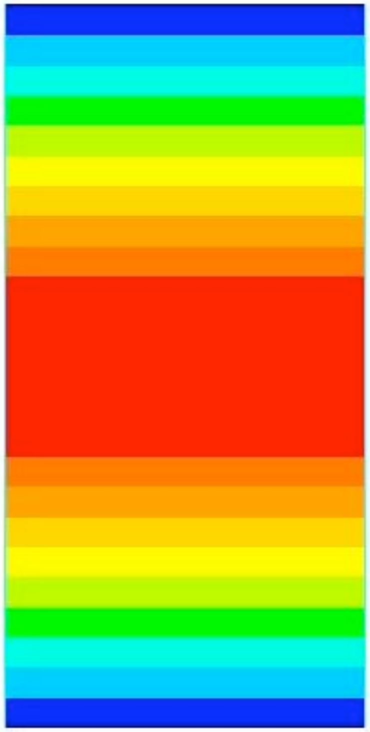
Coupled Solver

Discretization:
2nd order upwind
SIMPLE

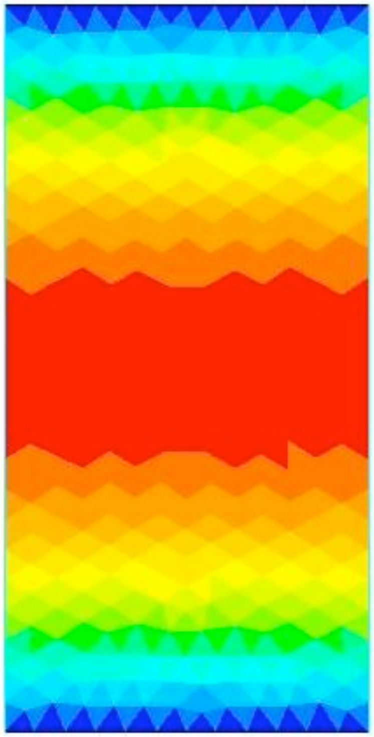
Multigrid
V-Cycle

Periodic Flow Example – 2D channel

x-velocity distribution in the channel



Quad-Mapping

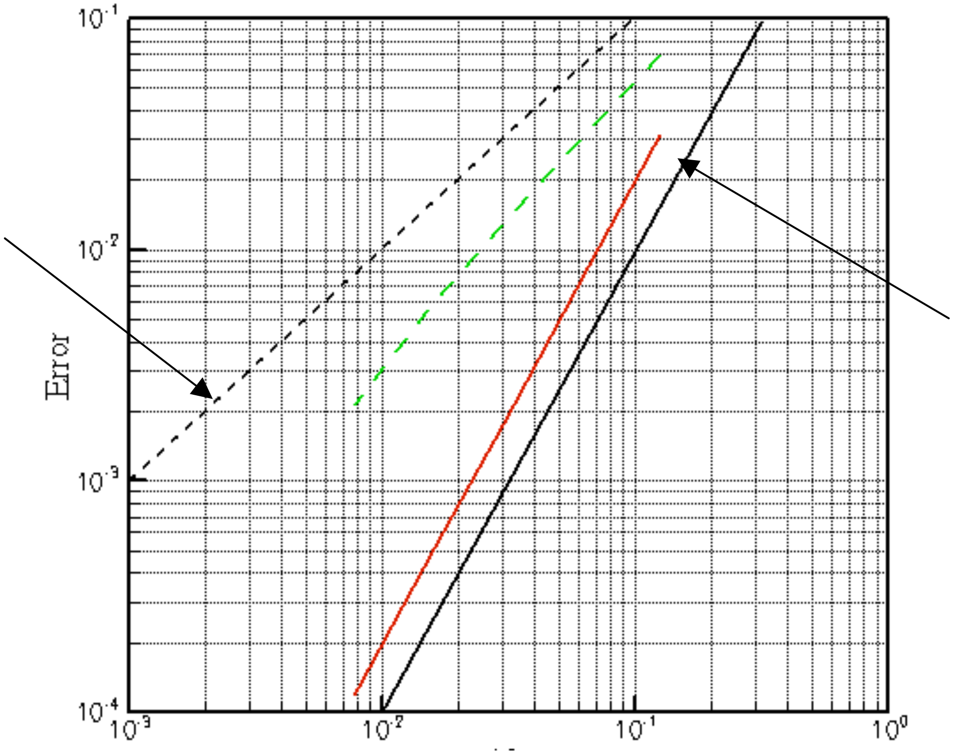


Tri-Paving

Cell-centered values showed (no interpolation)

Periodic Flow Example – 2D channel

The error in this case CAN be computed with reference to the exact solution
In this case the computed averaged velocity error is plotted



$(N)^{-1/2}$

Nominal 1st order accuracy

Nominal 2nd order accuracy

Quad-Mapping 🟥 Tri-Paving 🟩

This test-case is available on the class web site

Overview of commercial CFD codes

About 30 packages.

Three major general-purpose products (covering ~50% of the market):
FLUENT, StarCD, CFX

	Grid Type	**Pressure Based**	**Density Based**	**Multigrid**	**System Solver**	**Discretization**
FLUENT	Unstructured Mixed	SIMPLE SIMPLEC PISO	Coupled Implicit Preconditioned	Algebraic Geometric	Gauss-Seidel	UD/TVD QUICK
StarCD	Unstructured Mixed	SIMPLE SIMPISO PISO	-	-	Conjugate Gradient	UD/TVD QUICK CD
CFX	Unstructured Mixed	SIMPLE	-	Algebraic Coupled	ILU	UD/TVD QUICK CD