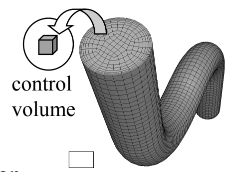
Introduction to CFD Analysis

What is CFD?

- Computational Fluid Dynamics (CFD) is the science of predicting fluid flow, heat transfer, mass transfer, chemical reactions, and related phenomena by solving mathematical equations that represent physical laws, using a numerical process.
  - Conservation of mass, momentum, energy, species, ...
- The result of CFD analyses is relevant engineering data:
  - conceptual studies of new designs
  - detailed product development
  - troubleshooting
  - redesign
- CFD analysis complements testing and experimentation.
  - Reduces the total effort required in the laboratory.

How does CFD work?

- FLUENT solvers are based on the finite volume method.
  - Domain is discretized into a finite set of control volumes or cells.
  - General conservation (transport) equation for mass, momentum, energy, etc.,



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

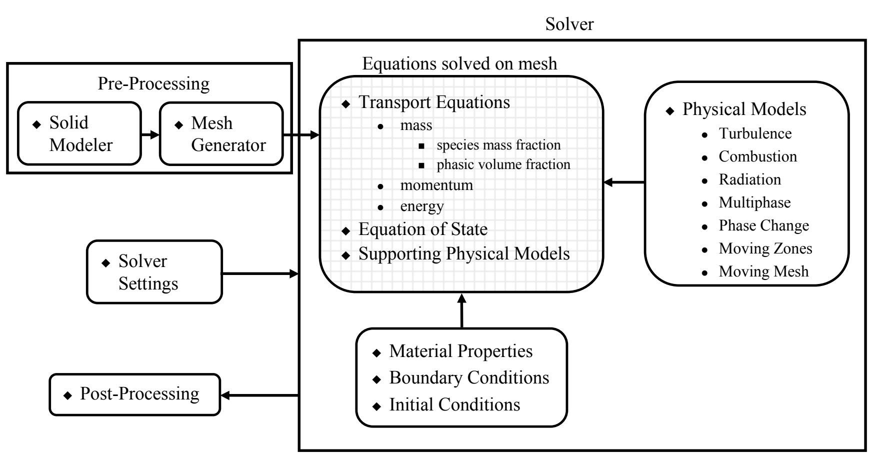
$$\underbrace{\frac{\partial}{\partial t}\int_V \rho\phi\, dV}_{\text{unsteady}} + \underbrace{\oint_A \rho\phi\mathbf{V}\cdot d\mathbf{A}}_{\text{convection}} = \underbrace{\oint_A \Gamma\nabla\phi\cdot d\mathbf{A}}_{\text{diffusion}} + \underbrace{\int_V S_\phi\, dV}_{\text{generation}}$$

Eqn.	ϕ
continuity	**1**
x-mom.	**u**
y-mom.	**v**
energy	**h**

are discretized into algebraic equations.

  - *All* equations are solved to render flow field.

CFD Modeling Overview



CFD Analysis: Basic Steps

- Problem Identification and Pre-Processing
  1. Define your modeling goals.
  2. Identify the domain you will model.
  3. Design and create the grid.
- Solver Execution
  4. Set up the numerical model.
  5. Compute and monitor the solution.
- Post-Processing
  6. Examine the results.
  7. Consider revisions to the model.

Define Your Modeling Goals

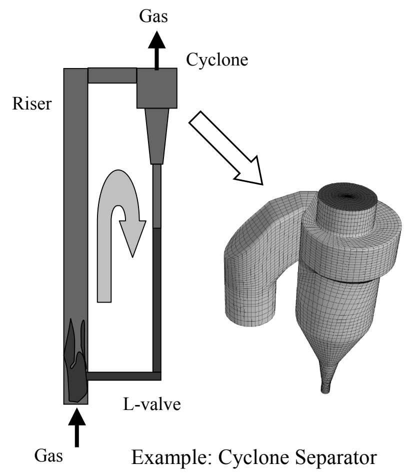
- Problem Identification and Pre-Processing
  1. Define your modeling goals.
  2. Identify the domain you will model.
  3. Design and create the grid.

- What results are you looking for, and how will they be used?
  - What are your modeling options?
    - What physical models will need to be included in your analysis?
    - What simplifying assumptions do you *have* to make?
    - What simplifying assumptions *can* you make?
    - Do you require a unique modeling capability?
      - User-defined functions (written in C) in FLUENT 6
      - User-defined subroutines (written in FORTRAN) in FLUENT 4.5
- What degree of accuracy is required?
- How quickly do you need the results?

Identify the Domain You Will Model

- Problem Identification and Pre-Processing
  1. Define your modeling goals.
  2. Identify the domain you will model.
  3. Design and create the grid

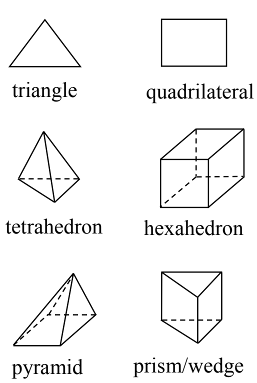
- How will you isolate a piece of the complete physical system?
- Where will the computational domain begin and end?
  - Do you have boundary condition information at these boundaries?
  - Can the boundary condition types accommodate that information?
  - Can you extend the domain to a point where reasonable data exists?
- Can the problem be simplified to 2D?



Example: Cyclone Separator

Design and Create the Grid

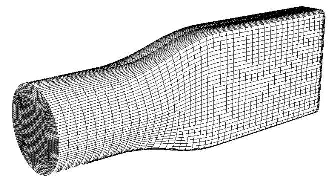
- Problem Identification and Pre-Processing
  1. Define your modeling goals.
  2. Identify the domain you will model.
  3. Design and create the grid.



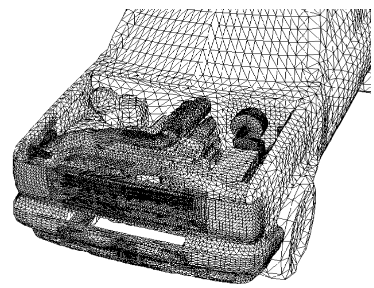
- Can you benefit from Mixsim, Icepak, or Airpak?
- Can you use a quad/hex grid or should you use a tri/tet grid or hybrid grid?
  - How complex is the geometry and flow?
  - Will you need a non-conformal interface?
- What degree of grid resolution is required in each region of the domain?
  - Is the resolution sufficient for the geometry?
  - Can you predict regions with high gradients?
  - Will you use adaption to add resolution?
- Do you have sufficient computer memory?
  - How many cells are required?
  - How many models will be used?

Tri/Tet vs. Quad/Hex Meshes

- For **simple** geometries, quad/hex meshes can provide high-quality solutions with fewer cells than a comparable tri/tet mesh.
  - Align the gridlines with the flow.

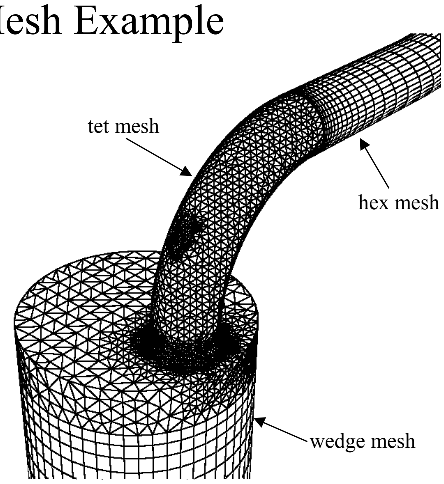


- For **complex** geometries, quad/hex meshes show no numerical advantage, and you can save meshing effort by using a tri/tet mesh.



Hybrid Mesh Example

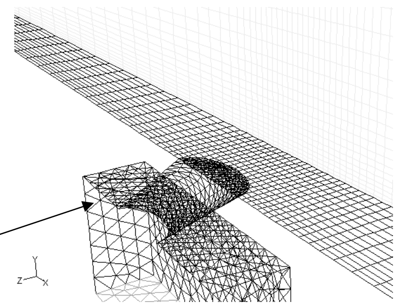
- Valve port grid
  - Specific regions can be meshed with different cell types.
  - Both efficiency and accuracy are enhanced relative to a hexahedral or tetrahedral mesh alone.
  - Tools for hybrid mesh generation are available in Gambit and TGrid.



Hybrid mesh for an
IC engine valve port

Non-Conformal Mesh Example

- **Nonconformal mesh:** mesh in which grid nodes do not match up along an interface.
  - Useful for 'parts-swapping' for design study, etc.
  - Helpful for meshing complex geometries.
- Example:
  - 3D Film Cooling Problem
    - Coolant is injected into a duct from a plenum
      - Plenum is meshed with tetrahedral cells.
      - Duct is meshed with hexahedral cells.



Plenum part can be replaced with new geometry with reduced meshing effort.

Set Up the Numerical Model

- Solver Execution
  4. Set up the numerical model.
  5. Compute and monitor the solution.

Solving initially in 2D will provide valuable experience with the models and solver settings for your problem in a short amount of time.

- For a given problem, you will need to:
  - Select appropriate physical models.
    - Turbulence, combustion, multiphase, etc.
  - Define material properties.
    - Fluid
    - Solid
    - Mixture
  - Prescribe operating conditions.
  - Prescribe boundary conditions at all boundary zones.
  - Provide an initial solution.
  - Set up solver controls.
  - Set up convergence monitors.

Compute the Solution

- Solver Execution
  4. Set up the numerical model.
  5. Compute and monitor the solution.

- The discretized conservation equations are solved *iteratively*.
  - A number of iterations are usually required to reach a converged solution.
- Convergence is reached when:
  - Changes in solution variables from one iteration to the next are negligible.
    - Residuals provide a mechanism to help monitor this trend.
  - Overall property conservation is achieved.
- The accuracy of a converged solution is dependent upon:
  - Appropriateness and accuracy of physical models.
  - Grid resolution and independence
  - Problem setup

A converged and grid-independent solution on a well-posed problem will provide useful engineering results!

Examine the Results

- Post-Processing
  6. Examine the results.
  7. Consider revisions to the model.

- Examine the results to review solution and extract useful data.
  - Visualization Tools can be used to answer such questions as:
    - What is the overall flow pattern?
    - Is there separation?
    - Where do shocks, shear layers, etc. form?
    - Are key flow features being resolved?
  - Numerical Reporting Tools can be used to calculate quantitative results:
    - Forces and Moments
    - Average heat transfer coefficients
    - Surface and Volume integrated quantities
    - Flux Balances

Examine results to ensure property conservation and correct physical behavior. High residuals may be attributable to only a few cells of poor quality.

Consider Revisions to the Model

- Post-Processing
  6. Examine the results.
  7. Consider revisions to the model.

- Are physical models appropriate?
  - Is flow turbulent?
  - Is flow unsteady?
  - Are there compressibility effects?
  - Are there 3D effects?
- Are boundary conditions correct?
  - Is the computational domain large enough?
  - Are boundary conditions appropriate?
  - Are boundary values reasonable?
- Is grid adequate?
  - Can grid be adapted to improve results?
  - Does solution change significantly with adaption, or is the solution grid independent?
  - Does boundary resolution need to be improved?

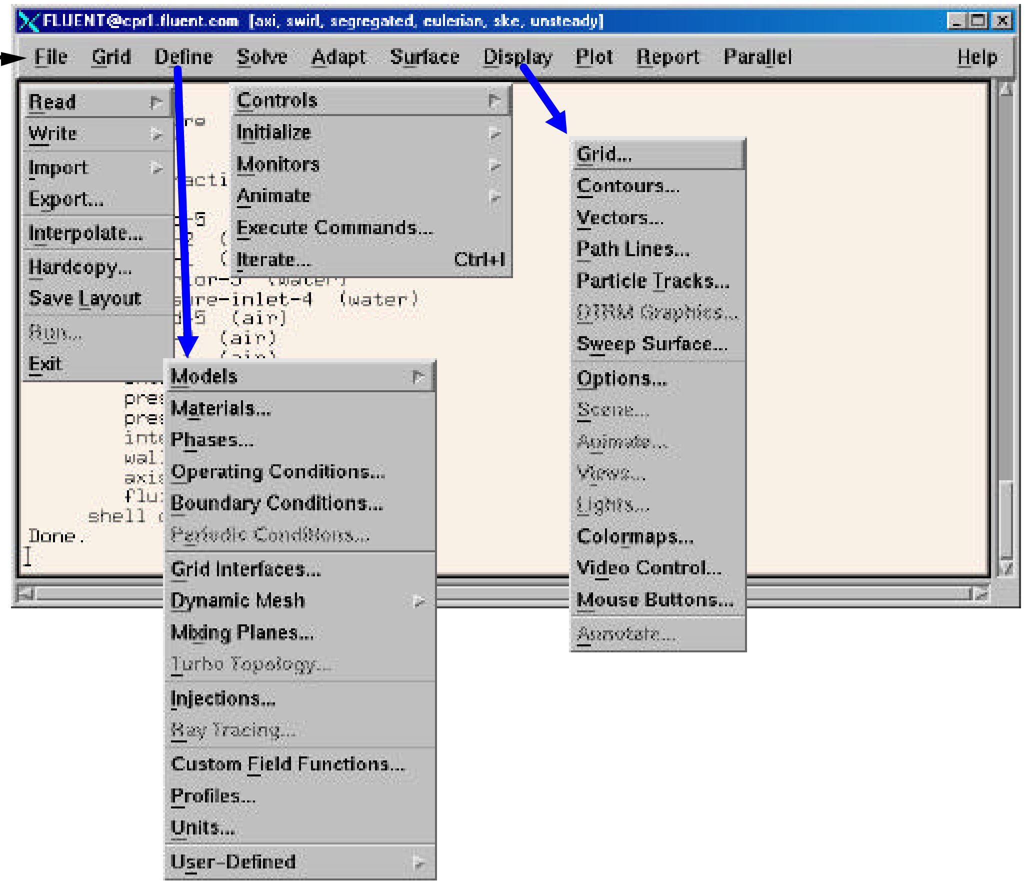
FLUENT DEMO

- **Startup Gambit (Pre-processing)**
  - **load database**
  - **define boundary zones**
  - **export mesh**
- **Startup Fluent (Solver Execution)**
  - **GUI**
  - **Problem Setup**
  - **Solve**
- **Post-Processing**
- **Online Documentation**

Solver Basics

Solver Execution

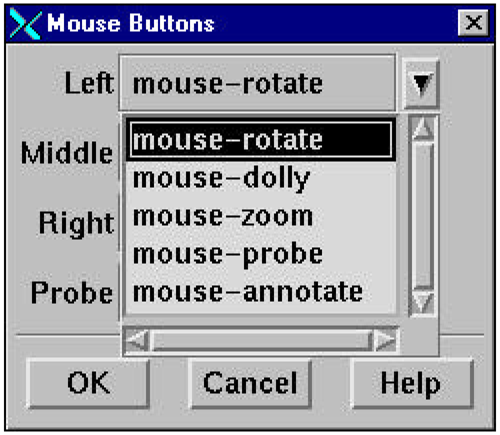
- Solver Execution:
  - Menu is laid out such that order of operation is generally left to right.
    - Import and scale mesh file.
    - Select physical models.
    - Define material properties.
    - Prescribe operating conditions.
    - Prescribe boundary conditions.
    - Provide an initial solution.
    - Set solver controls.
    - Set up convergence monitors.
    - Compute and monitor solution.
  - Post-Processing
    - Feedback into Solver
    - Engineering Analysis



Mouse Functionality

- Mouse button functionality depends on solver and can be configured in the solver.

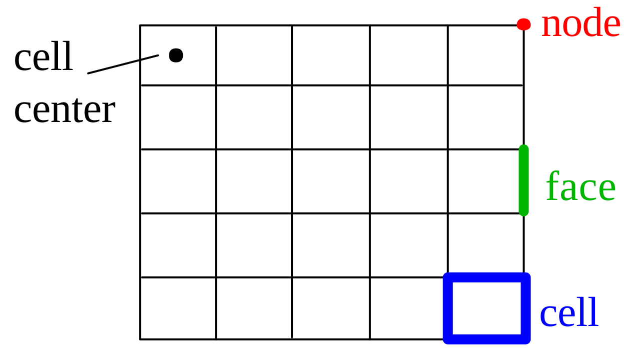
  Display → Mouse Buttons...



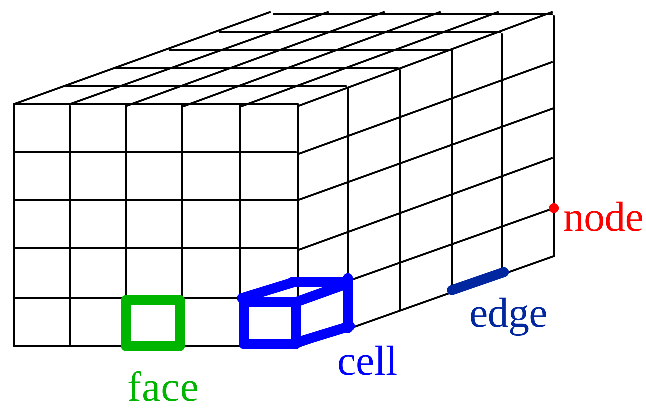
- Default Settings:
  - 2D Solver
    - Left button translates (dolly)
    - Middle button zooms
    - Right button selects/probes
  - 3D Solver
    - Left button rotates about 2-axes
    - Middle button zooms
      - Middle click on point in screen centers point in window
    - Right button selects/probes
- Retrieve detailed flow field information at point with Probe enabled.
  - Right click on grid display.

Reading Mesh: Mesh Components

- Components are defined in preprocessor
  - **Cell** = control volume into which domain is broken up
    - computational domain is defined by mesh that represents the fluid and solid regions of interest.
  - **Face** = boundary of a cell
  - **Edge** = boundary of a face
  - **Node** = grid point
  - **Zone** = grouping of nodes, faces, and/or cells
    - Boundary data assigned to face zones.
    - Material data and source terms assigned to cell zones.



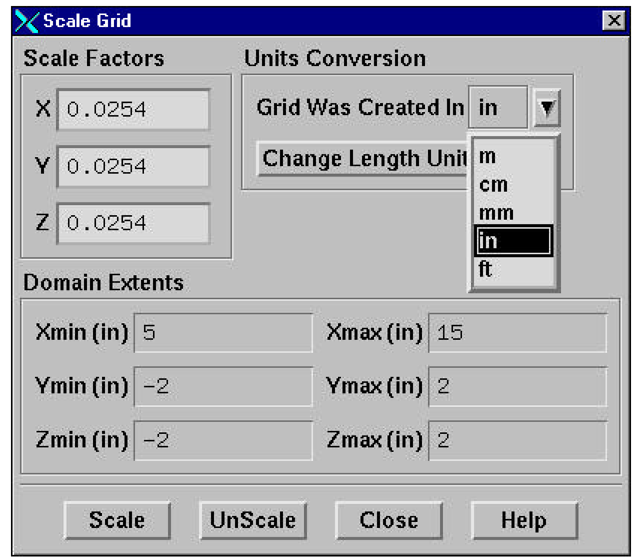
Simple 2D mesh

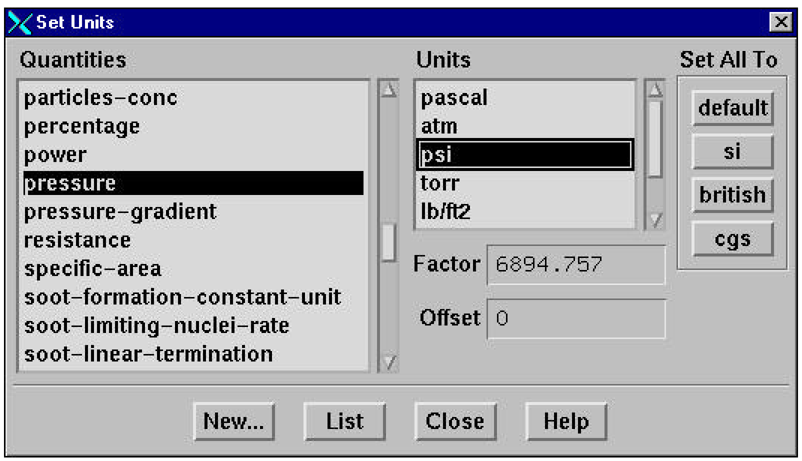


Simple 3D mesh

Scaling Mesh and Units

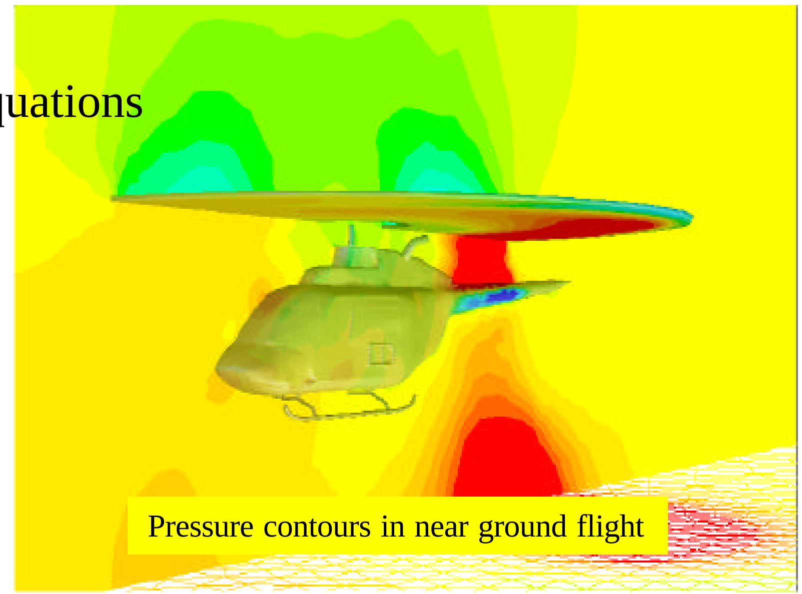
- All physical dimensions initially assumed to be in *meters.*
  - Scale grid accordingly.
- Other quantities can also be scaled independent of other units used.
  - Fluent defaults to SI units.

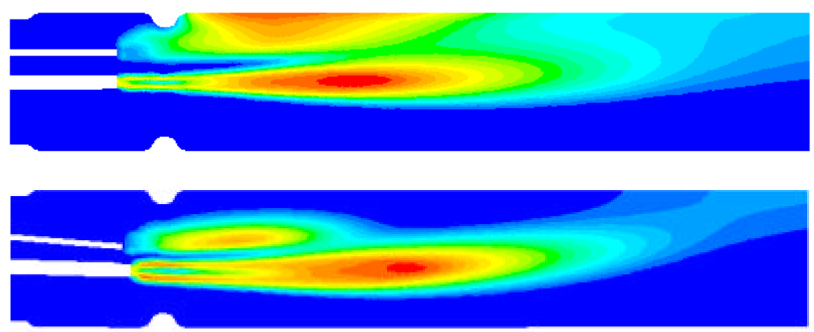




Models in Fluent 6 (1)

- Fluid Flow and Heat Transfer
  - Momentum, Continuity, and Energy Equations
  - Radiation Models
- Turbulence
  - RANS based models including k-ε, k-ω, and RSM.
  - LES
- Species Transport
  - Arrhenius Rate Chemistry
  - Turbulent Fast Chemistry
    - Eddy Dissipation, Non-Premixed, Premixed, Partially premixed
  - Turbulent Finite Rate Chemistry
    - EDC, laminar flamelet
  - Surface Reactions

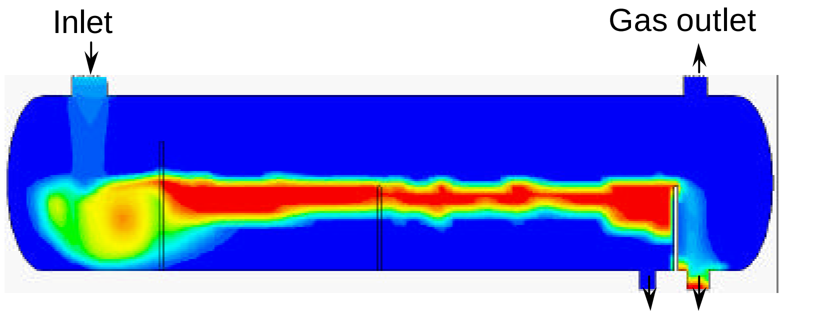




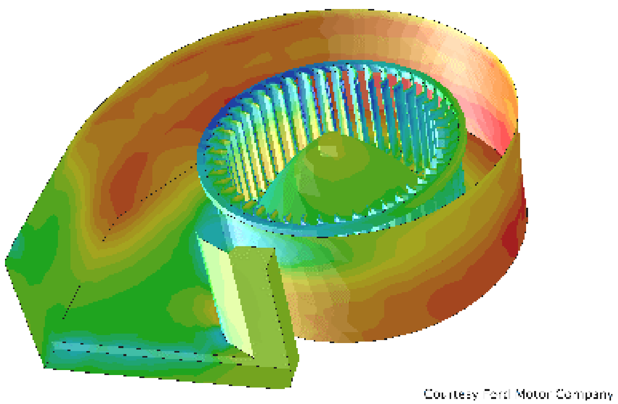
Temperature contours for kiln burner retrofitting.

Models in Fluent 6 (2)

- Multiple Phase Flows
  - Discrete Phase Model
  - VOF modeling of immiscible fluids
  - Mixture Model
  - Eulerian-Eulerian and Eulerian-Granular (heat transfer in Fluent 4.5 only)
  - Liquid/Solid and Cavitation Phase Change Models
- Flows involving Moving Parts
  - Moving zones
    - Rotating/Multiple Reference Frame
    - Mixing Plane
    - Sliding Mesh Model
  - Deforming Mesh (limited capability)
    - Special license needed, exception: Fluent 4.5
- User-Defined Scalar Transport



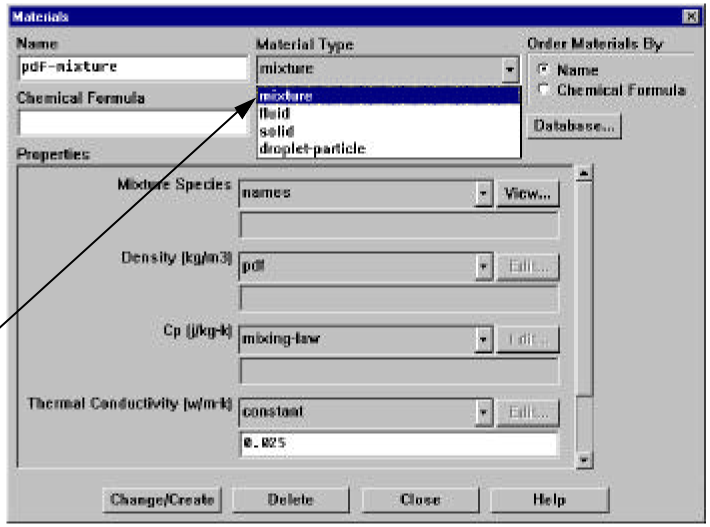
Contours of oil volume fraction in three phase separator.



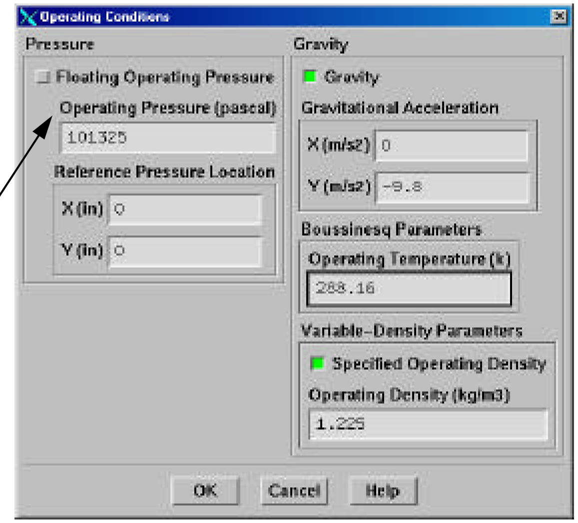
Pressure contours for squirrel cage blower.

Material Types and Property Definition

- Physical models may require inclusion of additional materials and dictates which properties need to be defined.
- Material *properties* defined in Materials Panel.
  - Single-Phase, Single Species Flows
    - Define fluid/solid properties
    - Real gas model (NIST’s REFPROP)
  - Multiple Species (Single Phase) Flows
    - *Mixture Material* concept employed
      - Mixture properties (composition dependent) defined separately from constituent’s properties.
      - Constituent properties must be defined.
    - *PDF Mixture Material* concept
      - PDF lookup table used for mixture properties.
        - Transport properties for mixture defined separately.
      - Constituent properties extracted from database.
  - Multiple Phase Flows (Single Species)
    - Define properties for all fluids and solids.

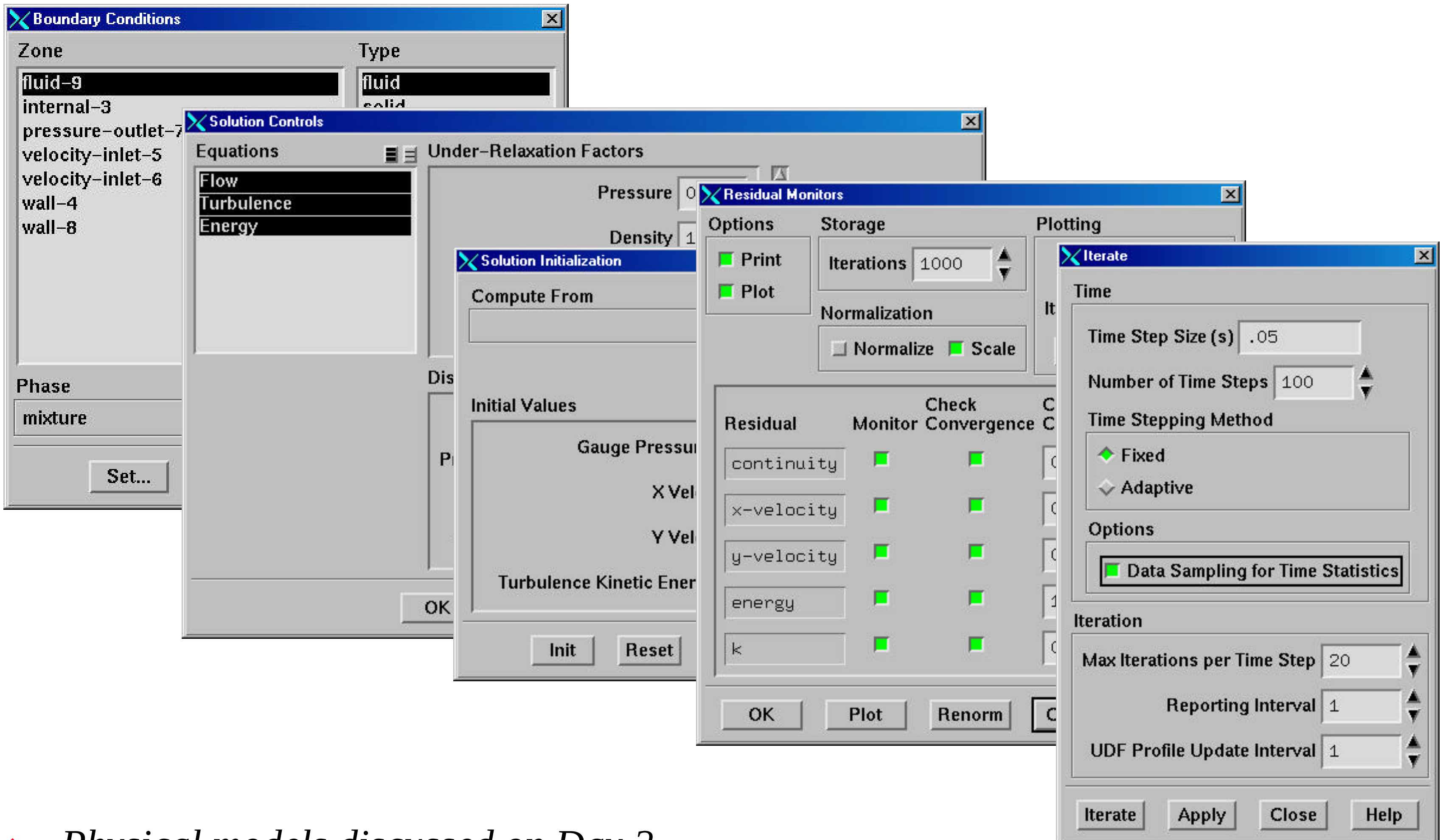


Fluid Density



- For ρ = constant, incompressible flow:
  - Select constant in Define → Materials...
- For incompressible flow:
  - $\rho = p_{operating}/RT$
    - Use incompressible-ideal-gas
    - Set $p_{operating}$ close to mean pressure in problem.
- For compressible flow use ideal-gas:
  - $\rho = p_{absolute}/RT$
    - For low Mach number flows, set $p_{operating}$ close to mean pressure in problem to avoid round-off errors.
    - Use Floating Operating Pressure for unsteady flows with large, gradual changes in absolute pressure (seg. only).
- Density can also be defined as a function of Temperature
  - polynomial or piecewise-polynomial
  - boussinesq model discussed in heat transfer lecture.
- Density can also be defined using UDF- *not to be function of pressure!*

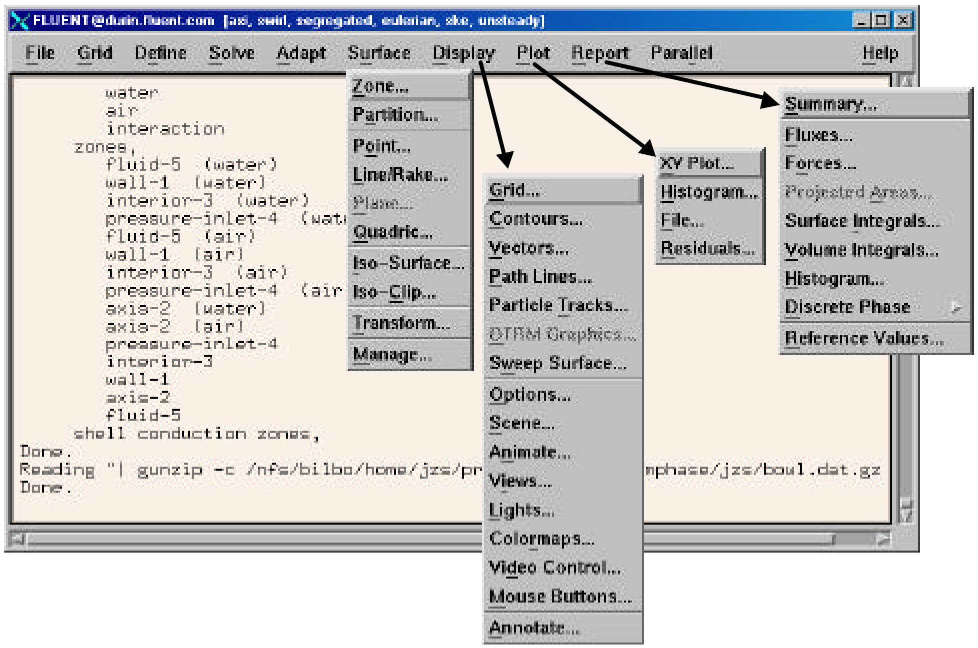
Solver Execution: Other Lectures...

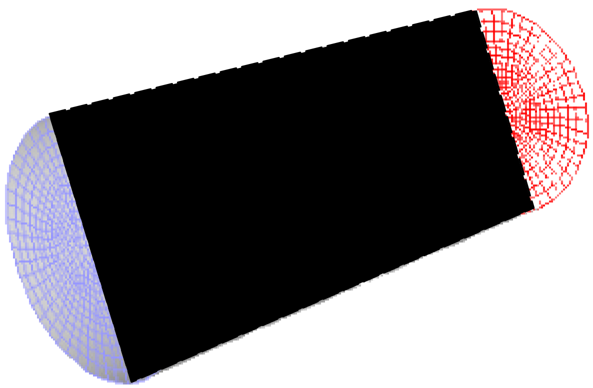


- *Physical models discussed on Day 2.*

Post-Processing

- Many post-processing tools are available.
- Post-Processing functions typically operate on surfaces.
  - Surfaces are automatically created from zones.
  - Additional surfaces can be created.

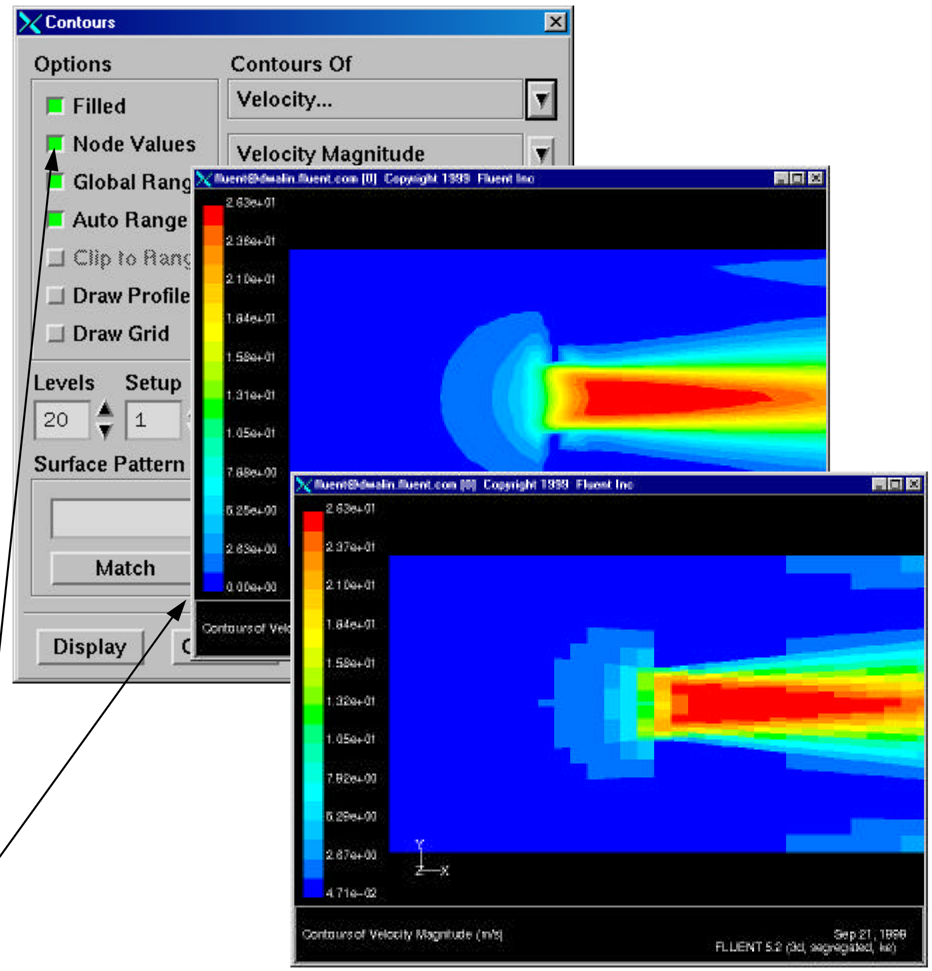




- Example: an Iso-Surface of constant grid coordinate can be created for viewing data within a plane.

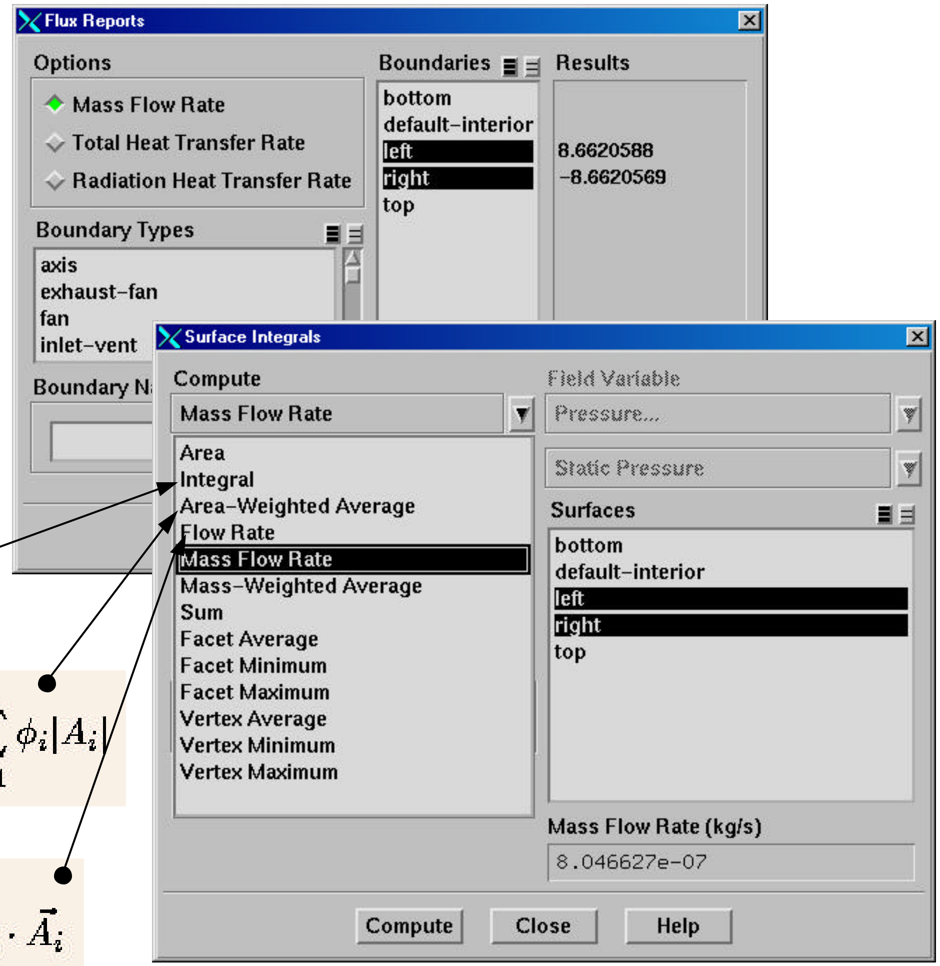
Post-Processing: Node Values

- Fluent calculates field variable data at cell centers.
- Node values of the grid are either:
  - calculated as the average of neighboring cell data, or,
  - defined explicitly (when available) with boundary condition data.
- Node values on surfaces are interpolated from grid node data.
- data files store:
  - data at cell centers
  - node value data for primitive variables at boundary nodes.
- Enable Node Values to interpolate field data to nodes.



Reports

- Flux Reports
  - Net flux is calculated.
  - Total Heat Transfer Rate includes radiation.
- Surface Integrals
  - slightly less accurate on user-generated surfaces due to interpolation error.
- Volume Integrals



$$\int \phi \, dA = \sum_{i=1}^{n} \phi_i |A_i|$$

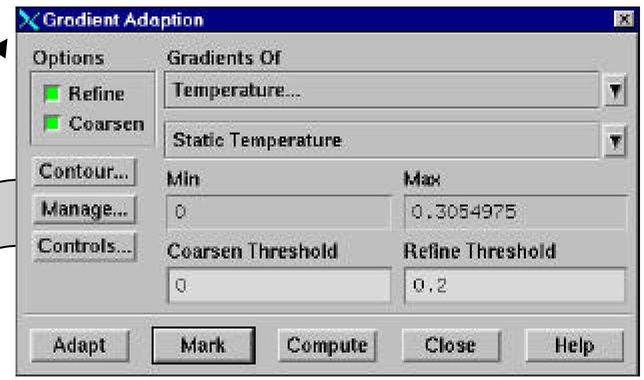
$$\frac{1}{A} \int \phi \, dA = \frac{1}{A} \sum_{i=1}^{n} \phi_i |A_i|$$

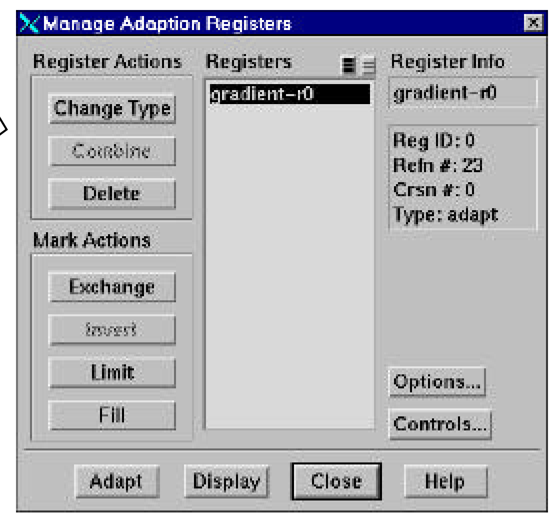
Examples:

$$\int \phi \rho \vec{V} \cdot d\vec{A} = \sum_{i=1}^{n} \phi_i \rho_i \vec{V}_i \cdot \vec{A}_i$$

Solver Enhancements: Grid Adaption

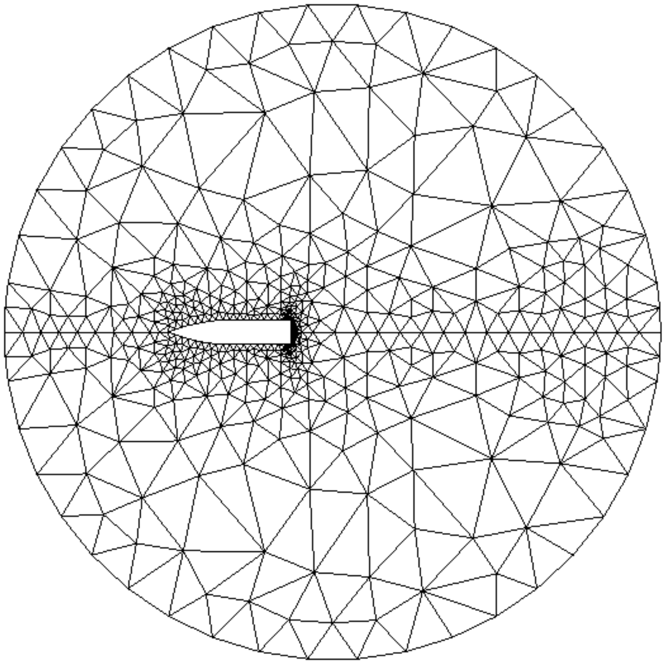
- Grid adaption adds more cells where needed to resolve the flow field *without pre-processor*.
- Fluent adapts on cells listed in register.
  - Registers can be defined based on:
    - Gradients of flow or user-defined variables
    - Iso-values of flow or user-defined variables
    - All cells on a boundary
    - All cells in a region
    - Cell volumes or volume changes
    - y^+ in cells adjacent to walls
  - To assist adaption process, you can:
    - Combine adaption registers
    - Draw contours of adaption function
    - Display cells marked for adaption
    - Limit adaption based on cell size and number of cells: Controls...



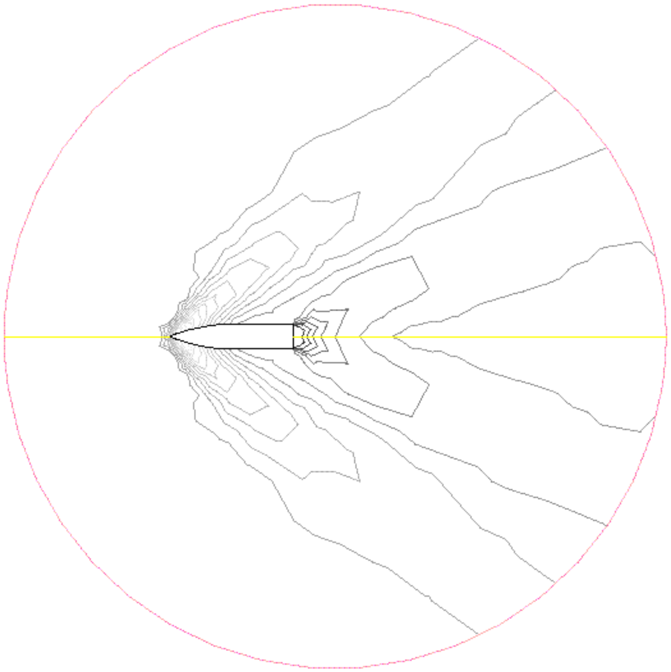


Adaption Example: 2D Planar Shell

- Adapt grid in regions of high pressure gradient to better resolve pressure jump across the shock.

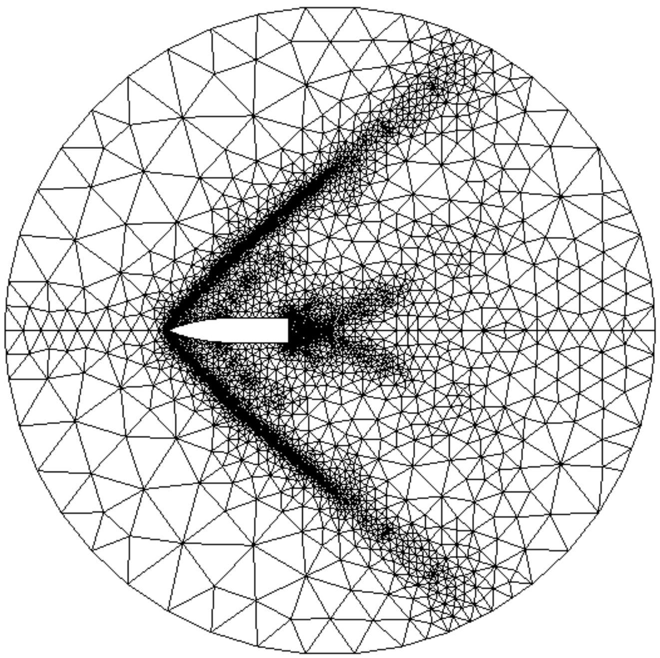


2D planar shell - initial grid

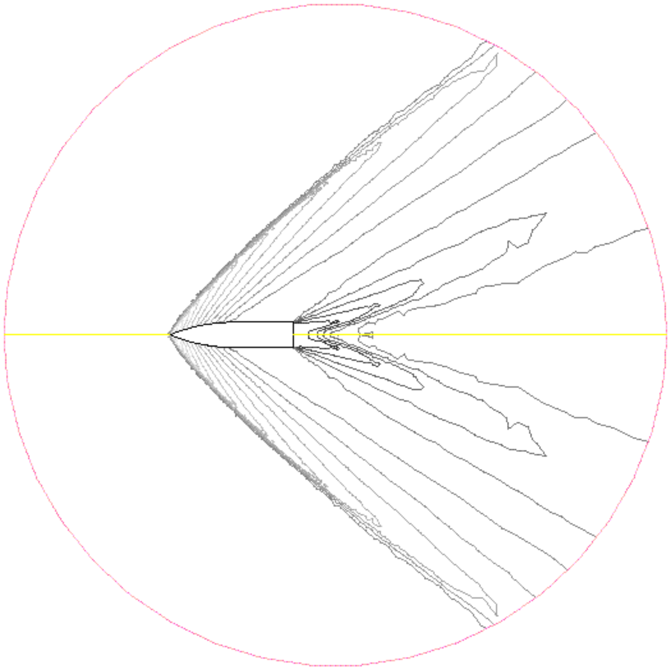


2D planar shell - contours of pressure initial grid

Adaption Example: Final Grid and Solution



2D planar shell - final grid



2D planar shell - contours of pressure
final grid

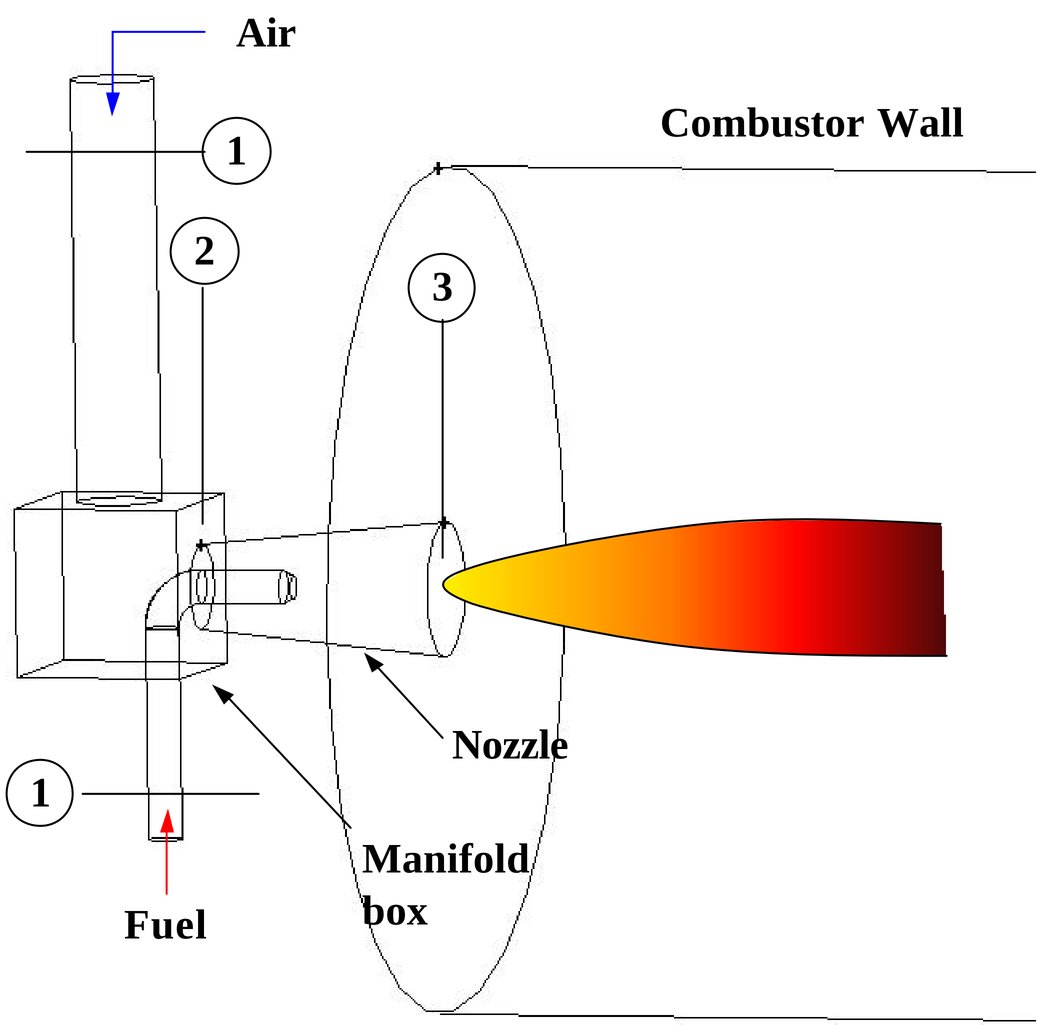
Boundary Conditions

Defining Boundary Conditions

- To define a problem that results in a *unique* solution, you must specify information on the dependent (flow) variables at the domain boundaries.
  - Specifying fluxes of mass, momentum, energy, etc. into domain.
- Defining boundary conditions involves:
  - identifying the location of the boundaries (e.g., inlets, walls, symmetry)
  - supplying information at the boundaries
- The data required at a boundary depends upon the boundary condition *type* and the physical models employed.
- You must be aware of the information that is required of the boundary condition and locate the boundaries where the information on the flow variables *are known or can be reasonably approximated*.
  - Poorly defined boundary conditions can have a significant impact on your solution.

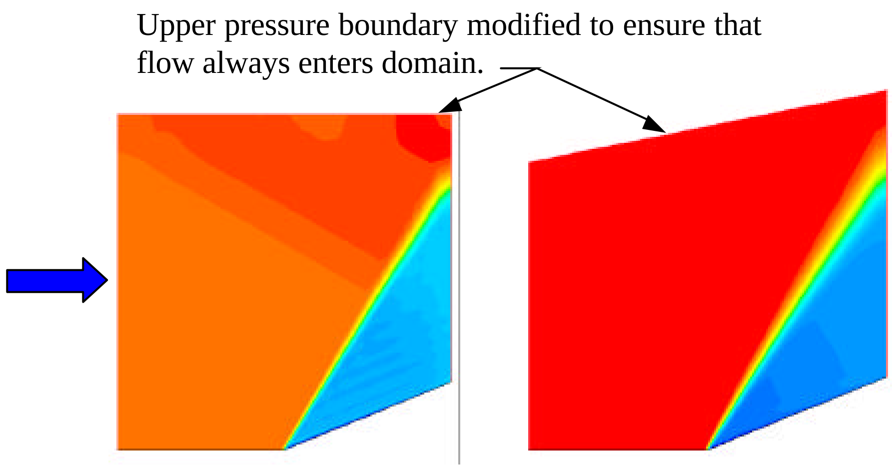
Locating Boundaries: Example

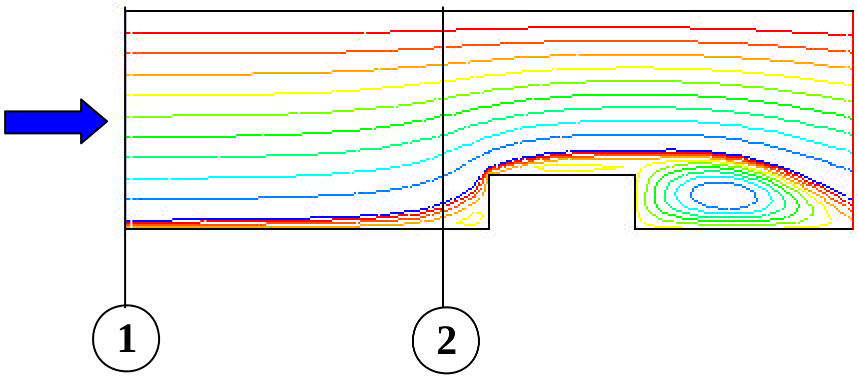
- Three possible approaches in locating inlet boundaries:
  - 1. Upstream of manifold
    - Can use uniform profile
    - Properly accounts for mixing
    - Non-premixed reaction models
    - Requires more cells
  - 2. Nozzle inlet plane
    - Non-premixed reaction models
    - Requires accurate profile data
  - 3. Nozzle outlet plane
    - Premixed reaction model
    - Requires accurate profile



General Guidelines

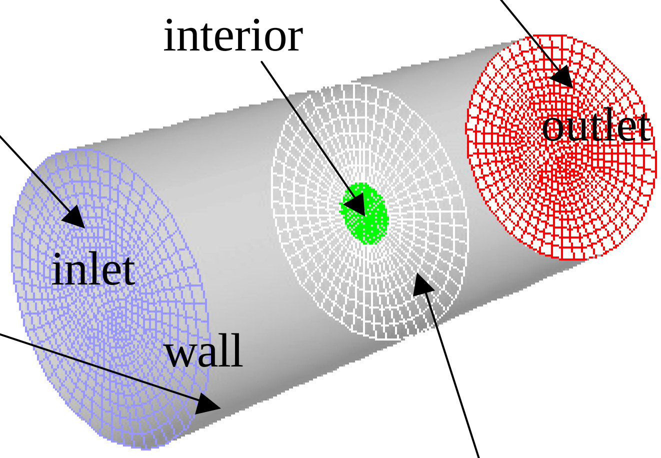
- General guidelines:
  - If possible, select boundary location and shape such that flow either goes in or out.
    - Not necessary, but will typically observe better convergence.
  - Should not observe large gradients in direction normal to boundary.
    - Indicates incorrect set-up.
  - Minimize grid skewness near boundary.
    - Introduces error early in calculation.





Available Boundary Condition Types

- Boundary Condition Types of *External Faces*
  - **General**: Pressure inlet, Pressure outlet
  - **Incompressible**: Velocity inlet, Outflow
  - **Compressible flows**: Mass flow inlet, Pressure far-field
  - **Special**: Inlet vent, outlet vent, intake fan, exhaust fan
  - **Other**: Wall, Symmetry, Periodic, Axis
- Boundary Condition Types of *Cell 'Boundaries'*
  - Fluid and Solid
- Boundary Condition Types of *Double-Sided Face 'Boundaries'*
  - Fan, Interior, Porous Jump, Radiator, Walls

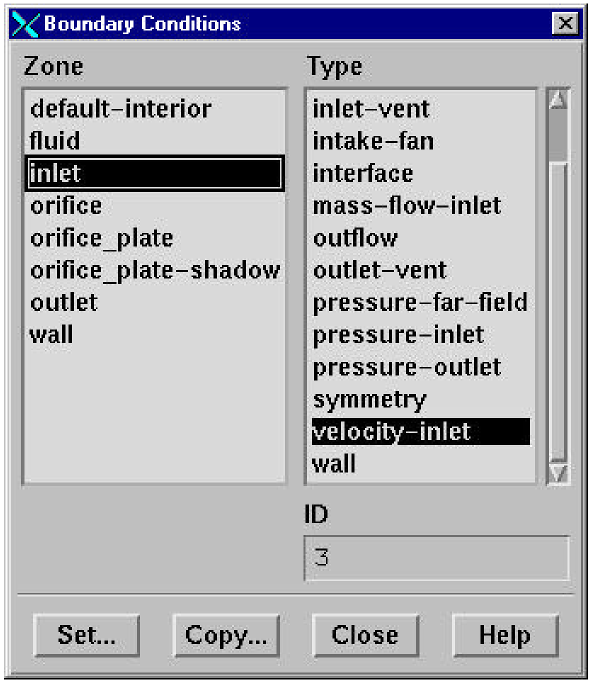


Changing Boundary Condition Types

- Zones and zone types are initially defined in pre-processor.
- To change zone type for a particular zone:

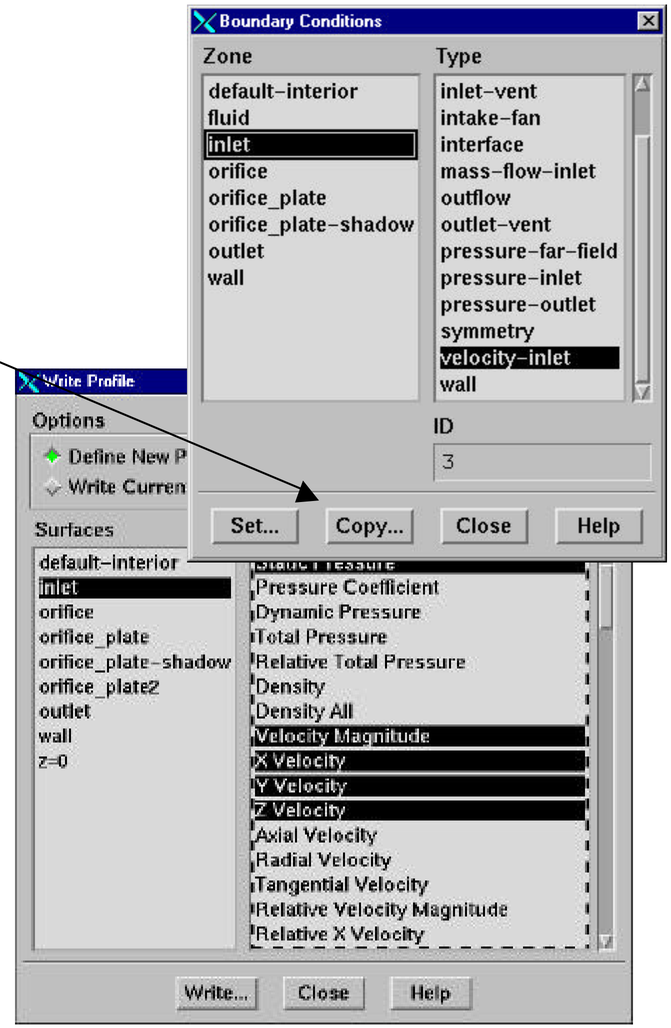
  Define → Boundary Conditions...

  - Choose the zone in Zone list.
    - Can also select boundary zone using right mouse button in Display Grid window.
  - Select new zone type in Type list.



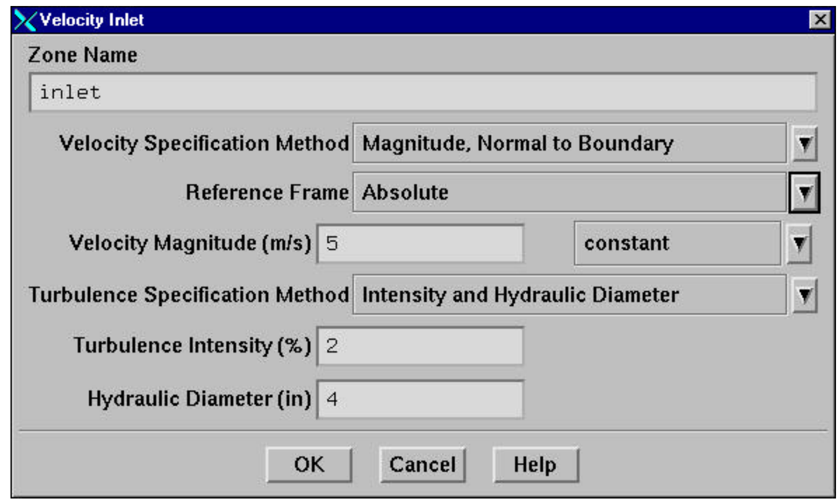
Setting Boundary Condition Data

- Explicitly assign data in BC panels.
  - To set boundary conditions for particular zone:
    - Choose the zone in Zone list.
    - Click Set... button
  - Boundary condition data can be copied from one zone to another.
- Boundary condition data can be stored and retrieved from file.
  - file → write-bc and file → read-bc
- Boundary conditions can also be defined by UDFs and Profiles.
- Profiles can be generated by:
  - Writing a profile from another CFD simulation
  - Creating an appropriately formatted text file with boundary condition data.



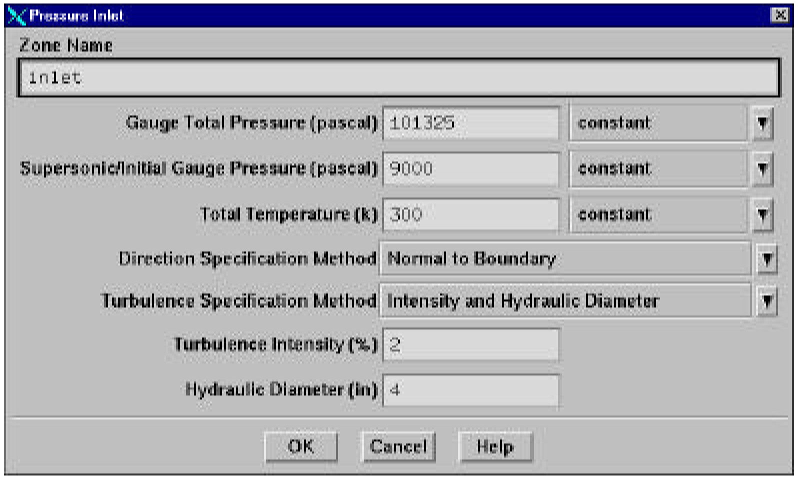
Velocity Inlet

- Specify Velocity by:
  - Magnitude, Normal to Boundary
  - Components
  - Magnitude and Direction
- Velocity profile is uniform by default
- Intended for incompressible flows.
  - Static pressure adjusts to accommodate prescribed velocity distribution.
  - Total (stagnation) properties of flow also varies.
  - Using in compressible flows can lead to non-physical results.
- Can be used as an outlet by specifying negative velocity.
  - You must ensure that mass conservation is satisfied if multiple inlets are used.



Pressure Inlet (1)

- Specify:
  - Total *Gauge* Pressure
    - Defines energy to drive flow.
    - Doubles as back pressure (static gauge) for cases where back flow occurs.
      - Direction of back flow determined from interior solution.
  - Static *Gauge* Pressure
    - Static pressure where flow is locally supersonic; ignored if subsonic
    - Will be used if flow field is initialized from this boundary.
  - Total Temperature
    - Used as static temperature for incompressible flow.
  - Inlet Flow Direction



Compressible flows:

$$p_{total,abs} = p_{static,abs}\left(1+\frac{k-1}{2}M^2\right)^{k/(k-1)}$$

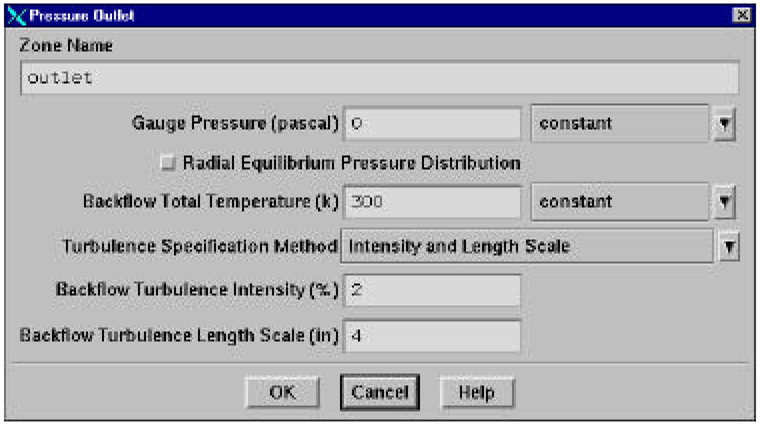
$$T_{total} = T_{static}\left(1+\frac{k-1}{2}M^2\right)$$

Incompressible flows: $p_{total} = p_{static} + \frac{1}{2} r v^2$

Pressure Inlet (2)

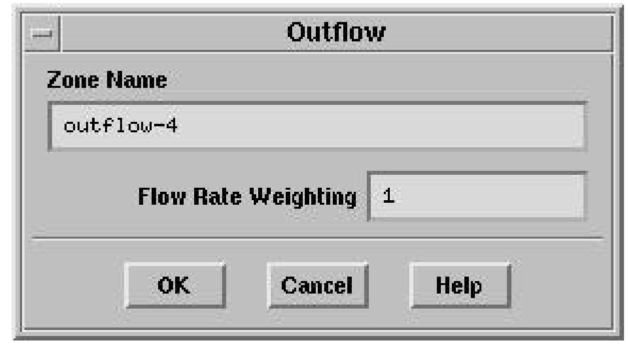
- Note: *Gauge* pressure inputs are required.
  - $p_{absolute} = p_{gauge} + p_{operating}$
  - Operating pressure input is set under: Define → Operating Conditions
- Suitable for compressible and incompressible flows.
  - Pressure inlet boundary is treated as loss-free transition from stagnation to inlet conditions.
  - Fluent calculates static pressure and velocity at inlet
  - Mass flux through boundary varies depending on interior solution and specified flow direction.
- Can be used as a "free" boundary in an external or unconfined flow.

Pressure Outlet



- Specify static *gauge* pressure
  - Interpreted as static pressure of environment into which flow exhausts.
  - Radial equilibrium pressure distribution option available.
  - Doubles as inlet pressure (*total gauge*) for cases where backflow occurs.
- Backflow
  - Can occur at pressure outlet during iterations or as part of final solution.
  - Backflow direction is assumed to be *normal* to the boundary.
  - Backflow boundary data must be set for all transport variables.
  - Convergence difficulties minimized by realistic values for backflow quantities.
- Suitable for compressible and incompressible flows
  - Pressure is ignored if flow is locally supersonic.
- Can be used as a “free” boundary in an external or unconfined flow.

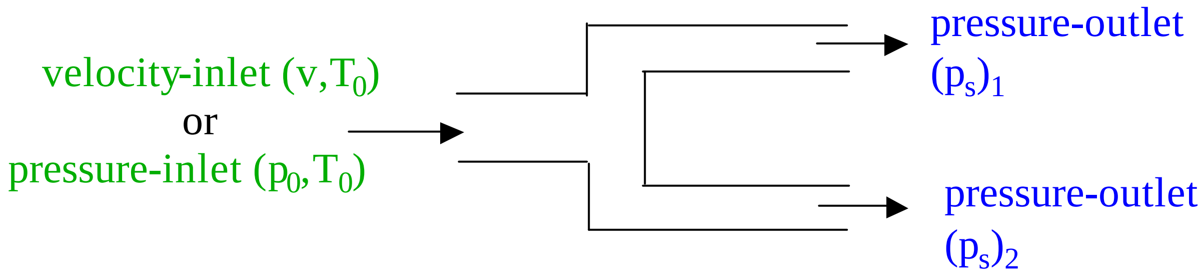
Outflow



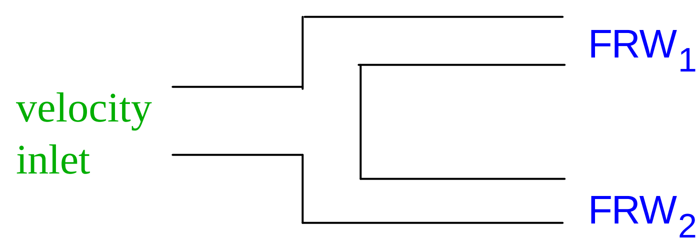
- No pressure *or* velocity information is required.
  - Data at exit plane is extrapolated from interior.
  - Mass balance correction is applied at boundary.
- Flow exiting Outflow boundary exhibits zero normal diffusive flux for all flow variables.
  - Appropriate where exit flow is close to fully developed condition.
- Intended for incompressible flows.
  - Cannot be used with a Pressure Inlet; must use velocity inlet.
    - Combination does not uniquely set pressure gradient over whole domain.
  - Cannot be used for unsteady flows with variable density.
- Poor rate of convergence when back flow occurs during iteration.
  - Cannot be used if back flow is expected in final solution.

Modeling Multiple Exits

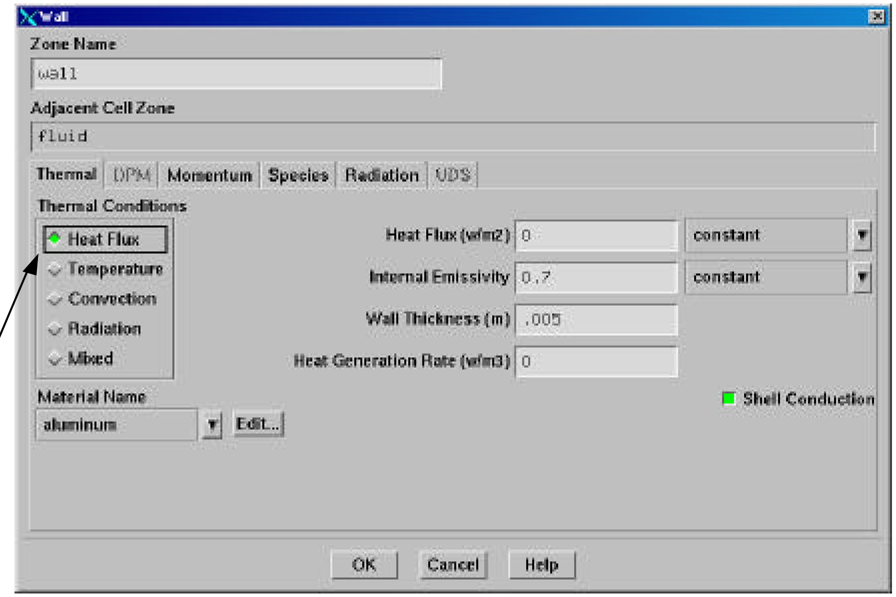
- Flows with multiple exits can be modeled using Pressure Outlet or Outflow boundaries.
  - Pressure Outlets



  - Outflow:
    - Mass flow rate fraction determined from Flow Rate Weighting by:
      - $m_i = FRW_i / \Sigma FRW_i$ where $0 < FRW < 1$.
      - FRW set to 1 by default implying equal flow rates
    - static pressure varies among exits to accommodate flow distribution.



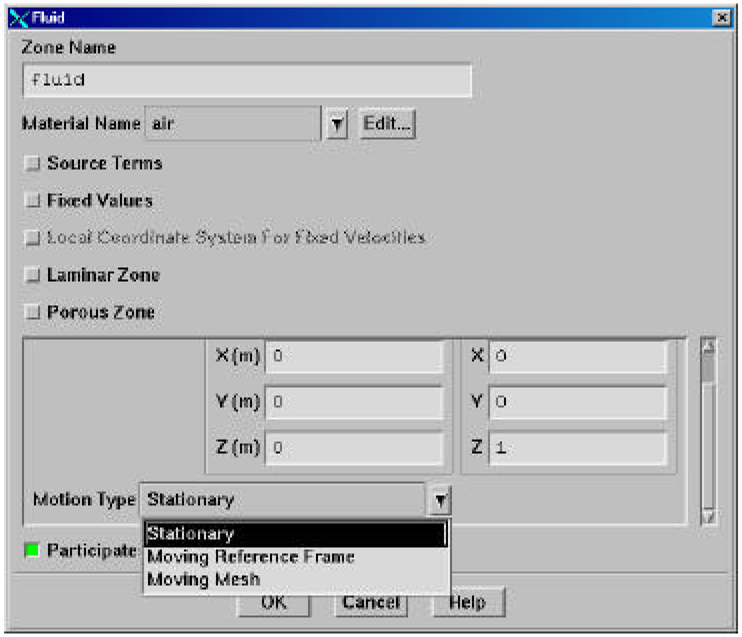
Wall Boundaries



- Used to bound fluid and solid regions.
- In viscous flows, no-slip condition enforced at walls:
  - Tangential fluid velocity equal to wall velocity.
  - Normal velocity component = 0
  - Shear stress can also be specified.
- Thermal boundary conditions:
  - several types available
  - Wall material and thickness can be defined for 1-D or shell conduction heat transfer calculations.
- Wall roughness can be defined for turbulent flows.
  - Wall shear stress and heat transfer based on local flow field.
- Translational or rotational velocity can be assigned to wall.

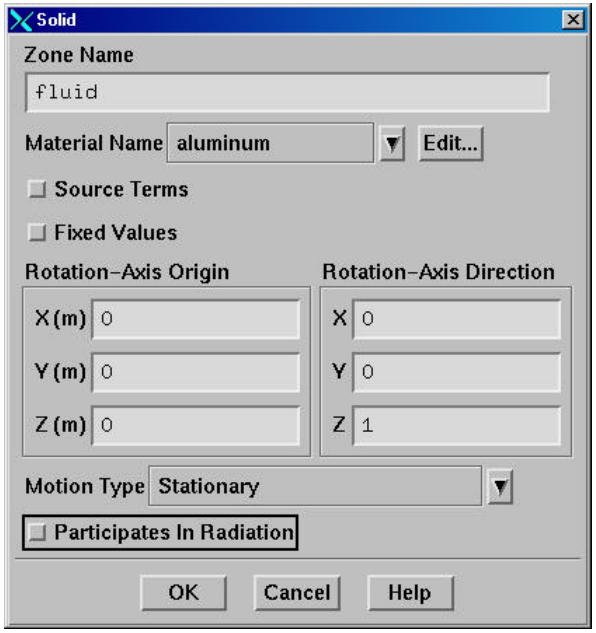
Cell Zones: Fluid

- Fluid zone = group of cells for which all active equations are solved.
- Fluid material input required.
  - Single species, phase.
- Optional inputs allow setting of source terms:
  - mass, momentum, energy, etc.
- Define fluid zone as laminar flow region if modeling transitional flow.
- Can define zone as porous media.
- Define axis of rotation for rotationally periodic flows.
- Can define motion for fluid zone.



Cell Zones: Solid

- "Solid" zone = group of cells for which only heat conduction problem solved.
  - No flow equations solved
  - Material being treated as solid may actually be fluid, but it is assumed that no convection takes place.
- Only required input is material type
  - So appropriate material properties used.
- Optional inputs allow you to set volumetric heat generation rate (heat source).
- Need to specify rotation axis if rotationally periodic boundaries adjacent to solid zone.
- Can define motion for solid zone



Internal Face Boundaries

- Defined on cell faces
  - Do not have finite thickness
  - Provide means of introducing step change in flow properties.
- Used to implement physical models representing:
  - Fans
  - Radiators
  - Porous jump
    - Preferable over porous media- exhibits better convergence behavior.
  - Interior wall

Summary

- Zones are used to assign boundary conditions.
- Wide range of boundary conditions permit flow to enter and exit solution domain.
- Wall boundary conditions used to bound fluid and solid regions.
- Repeating boundaries used to reduce computational effort.
- Internal cell zones used to specify fluid, solid, and porous regions.
- Internal face boundaries provide way to introduce step change in flow properties.

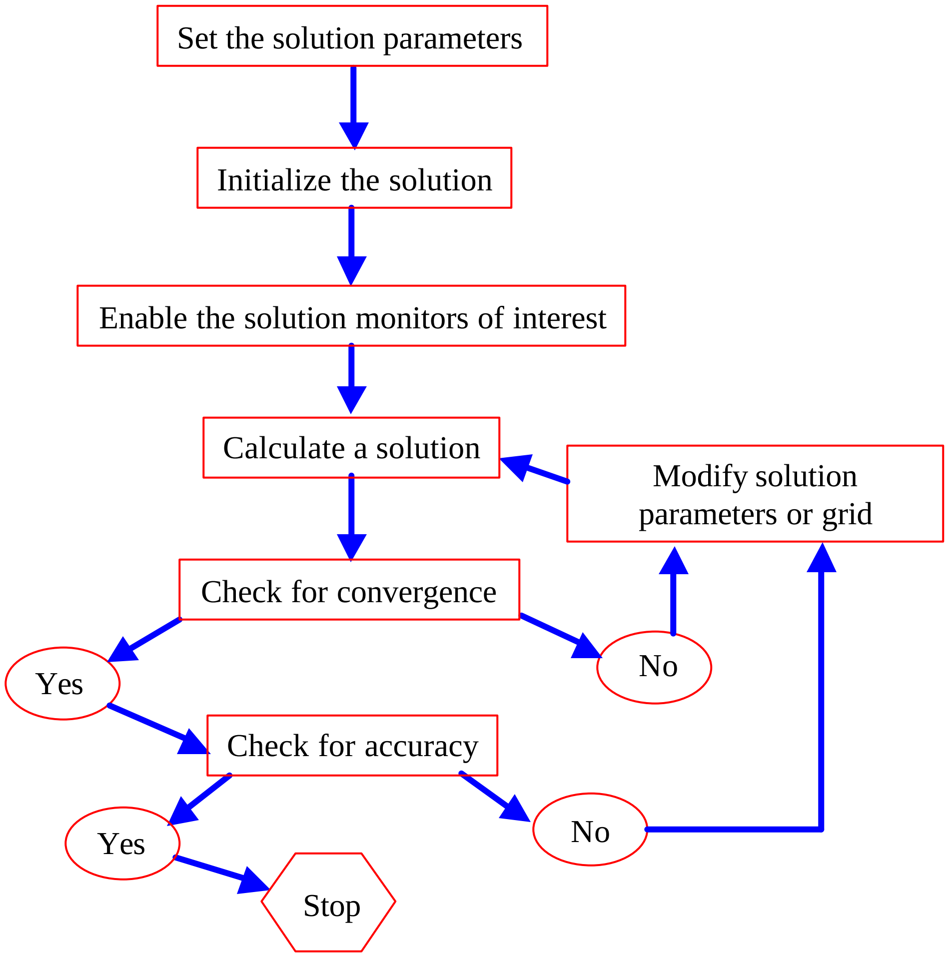
Solver Settings

Outline

- Using the Solver
  - Setting Solver Parameters
  - Convergence
    - Definition
    - Monitoring
    - Stability
    - Accelerating Convergence
  - Accuracy
    - Grid Independence
    - Adaption
- Appendix: Background
  - Finite Volume Method
  - Explicit vs. Implicit
  - Segregated vs. Coupled
  - Transient Solutions

Solution Procedure Overview

- Solution Parameters
  - Choosing the Solver
  - Discretization Schemes
- Initialization
- Convergence
  - Monitoring Convergence
  - Stability
    - Setting Under-relaxation
    - Setting Courant number
  - Accelerating Convergence
- Accuracy
  - Grid Independence
  - Adaption



Choosing a Solver

- Choices are Coupled-Implicit, Coupled-Explicit, or Segregated (Implicit)
- The Coupled solvers are recommended if a strong inter-dependence exists between density, energy, momentum, and/or species.
  - e.g., high speed compressible flow or finite-rate reaction modeled flows.
  - In general, the Coupled-Implicit solver is recommended over the coupled-explicit solver.
    - Time required: Implicit solver runs roughly twice as fast.
    - Memory required: Implicit solver requires roughly twice as much memory as coupled-explicit *or* segregated-implicit solvers!
  - The Coupled-Explicit solver should only be used for unsteady flows when the characteristic time scale of problem is on same order as that of the acoustics.
    - e.g., tracking transient shock wave
- The Segregated (implicit) solver is preferred in all other cases.
  - Lower memory requirements than coupled-implicit solver.
  - Segregated approach provides flexibility in solution procedure.

Initialization

- Iterative procedure requires that all solution variables be initialized before calculating a solution.

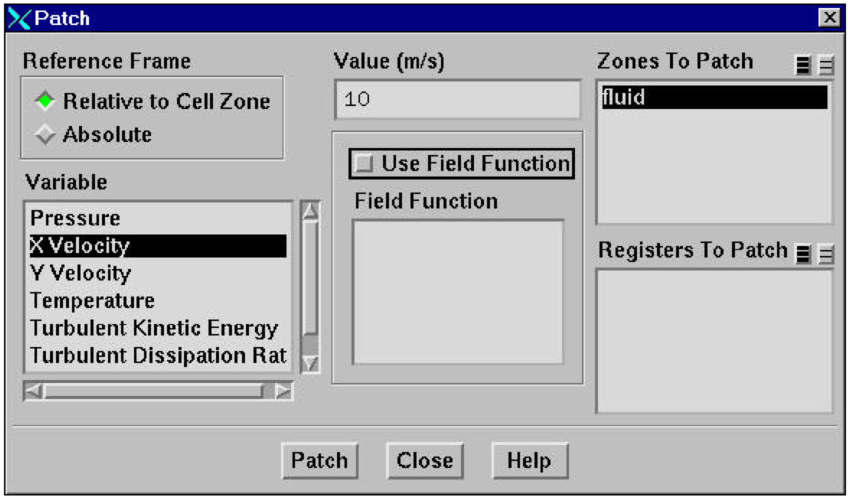
  Solve → Initialize → Initialize...

  - Realistic 'guesses' improves solution stability and accelerates convergence.
  - In some cases, **correct** initial guess is required:
    - Example: high temperature region to initiate chemical reaction.

- "Patch" values for individual variables in certain regions.

  Solve → Initialize → Patch...

  - Free jet flows
    (patch high velocity for jet)
  - Combustion problems
    (patch high temperature for ignition)



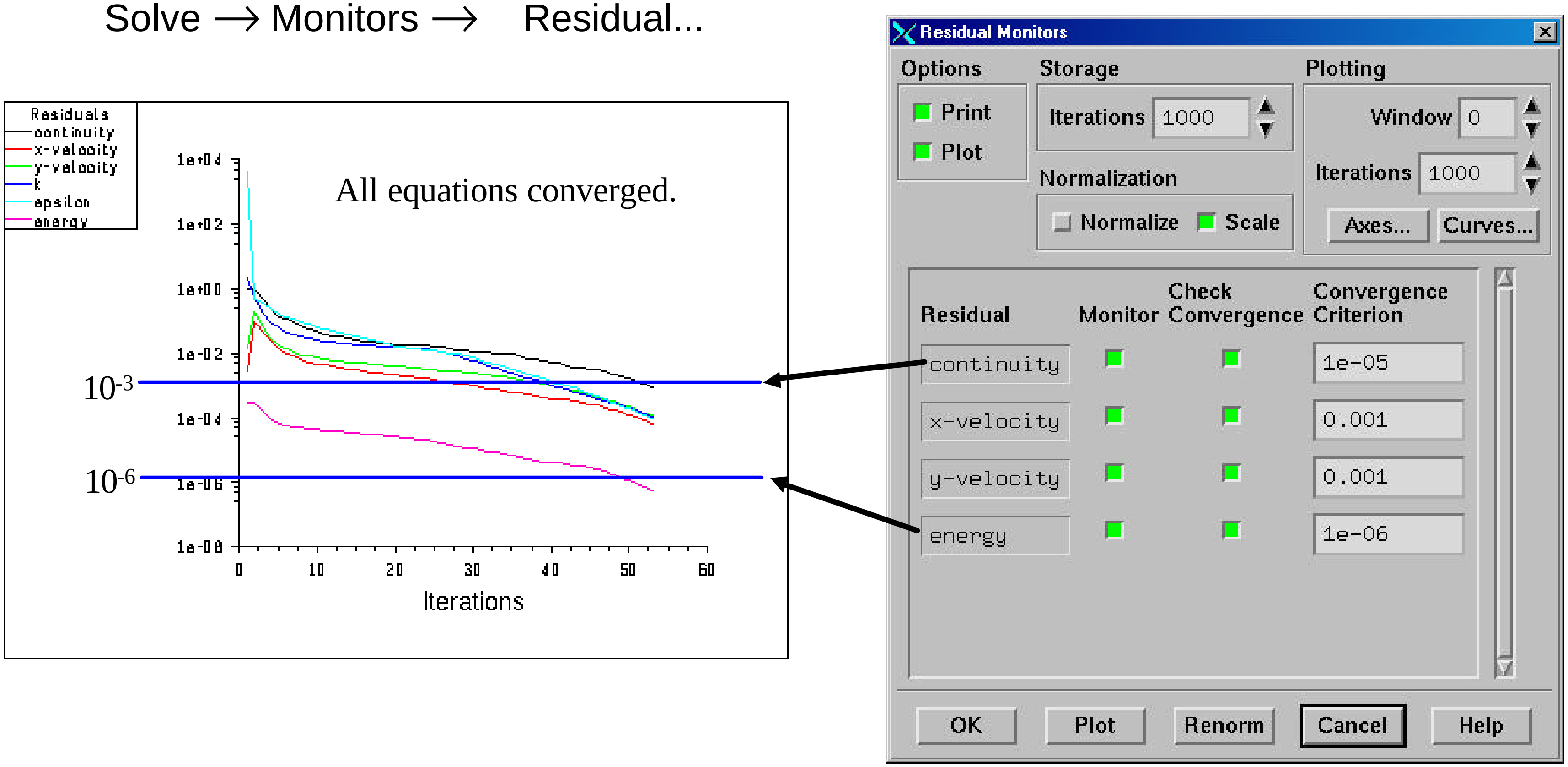
Convergence

- At convergence:
  - All discrete conservation equations (momentum, energy, etc.) are obeyed in all cells *to a specified tolerance*.
  - Solution no longer changes with more iterations.
  - Overall mass, momentum, energy, and scalar balances are obtained.
- Monitoring convergence with residuals:
  - Generally, a decrease in residuals by 3 orders of magnitude indicates at least qualitative convergence.
    - Major flow features established.
  - Scaled energy residual must decrease to 10^{-6} for segregated solver.
  - Scaled species residual may need to decrease to 10^{-5} to achieve species balance.
- Monitoring quantitative convergence:
  - Monitor other variables for changes.
  - Ensure that property conservation is satisfied.

Convergence Monitors: Residuals

- Residual plots show when the residual values have reached the specified tolerance.

Solve ⟶ Monitors ⟶ Residual...

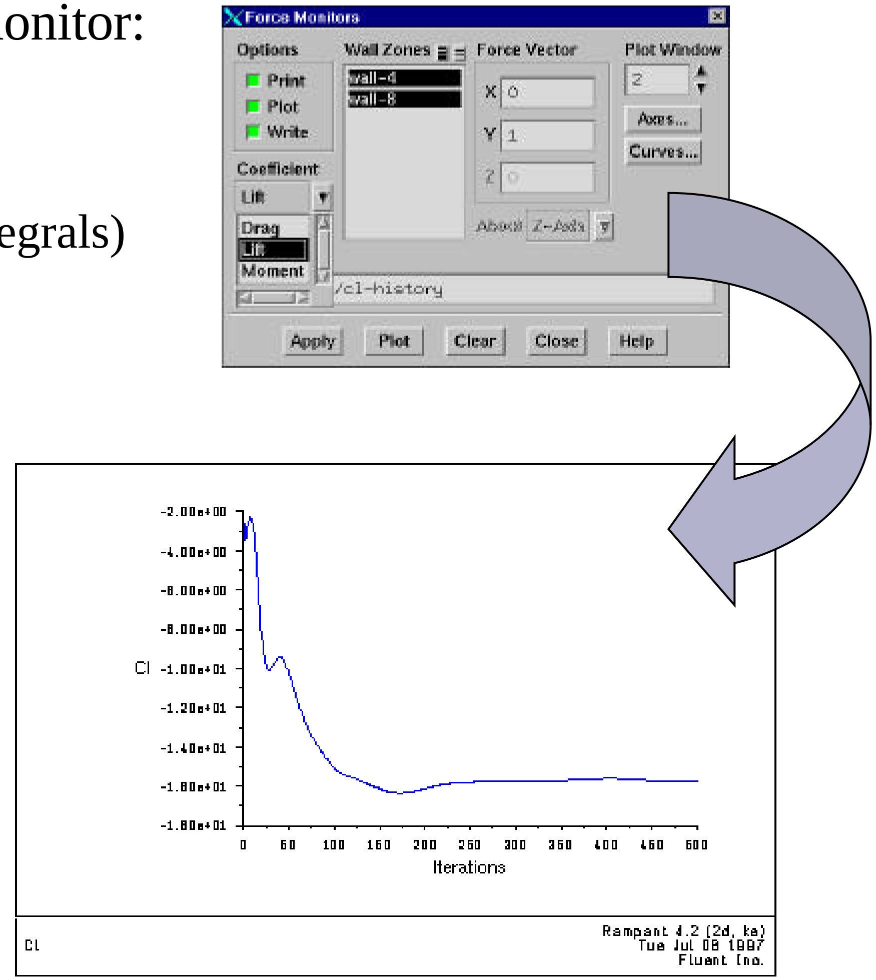


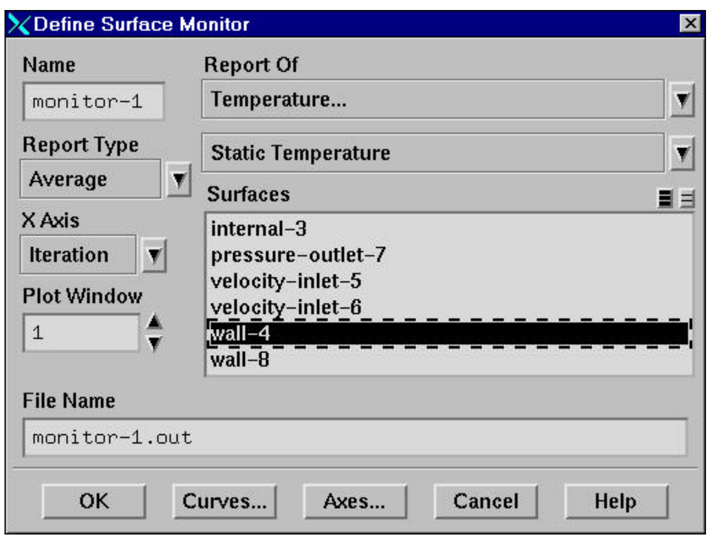
Convergence Monitors: Forces/Surfaces

- In addition to residuals, you can also monitor:
  - Lift, drag, or moment

    Solve → Monitors → Force...
  - Variables or functions (e.g., surface integrals) at a boundary or any defined surface:

    Solve → Monitors → Surface...

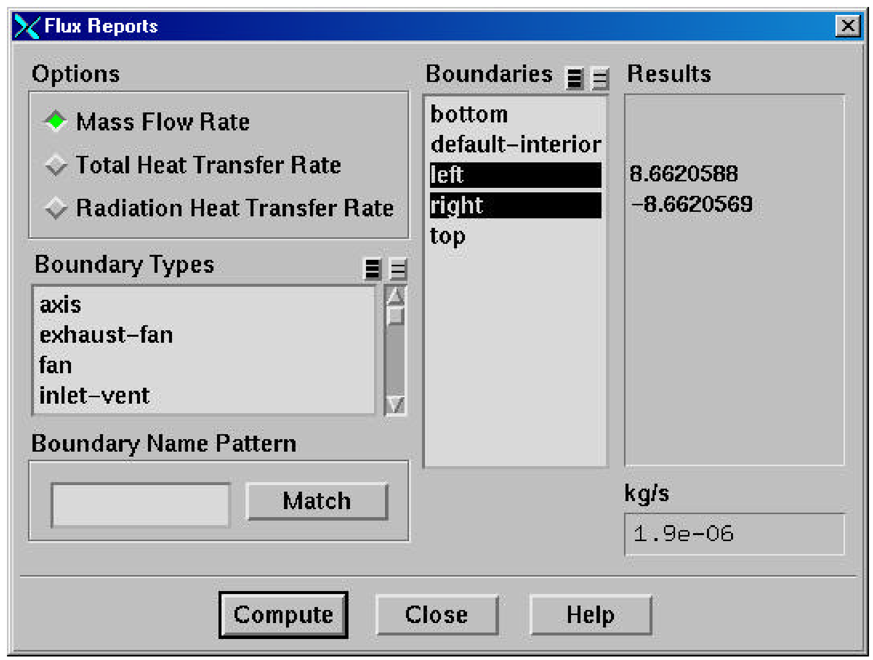




Checking for Property Conservation

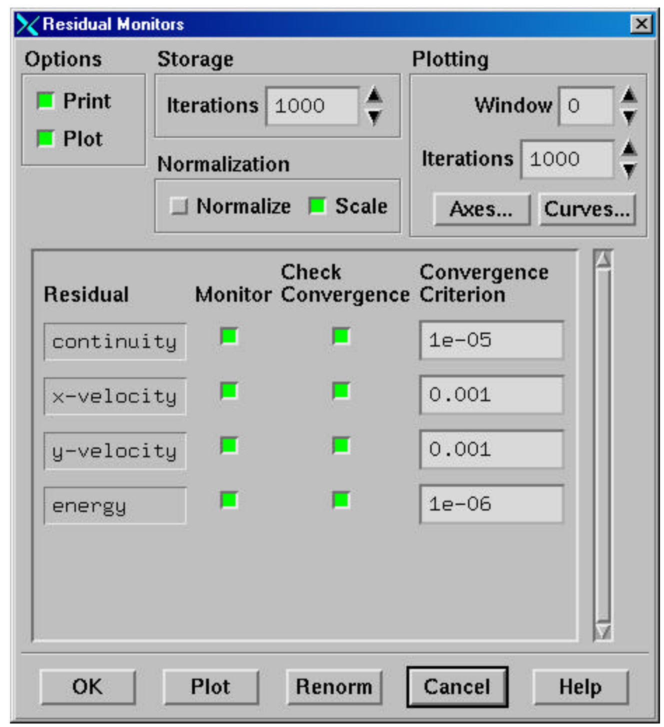
- In addition to monitoring residual and variable histories, you should also check for overall heat and mass balances.
  - At a minimum, the net imbalance should be less than 1% of smallest flux through domain boundary.

Report → Fluxes...



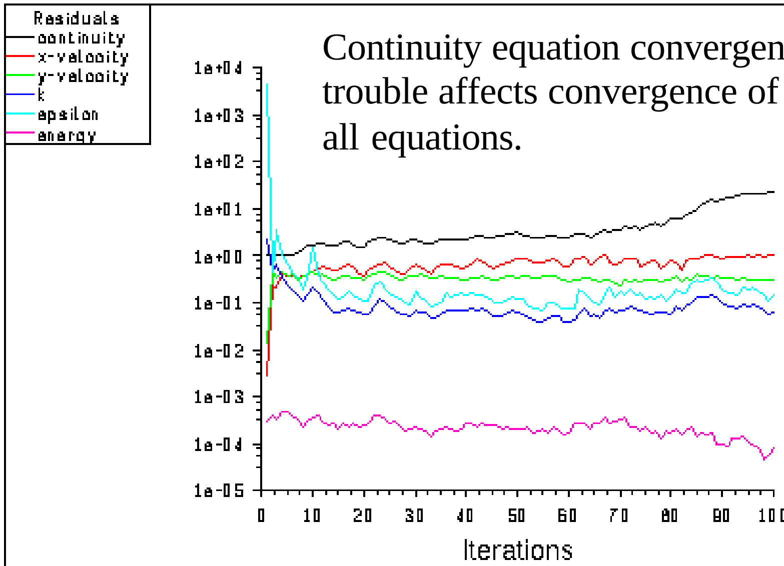
Decreasing the Convergence Tolerance

- If your monitors indicate that the solution is converged, but the solution is still changing or has a large mass/heat imbalance:
  - Reduce Convergence Criterion or disable Check Convergence.
  - Then calculate until solution converges to the new tolerance.



Convergence Difficulties

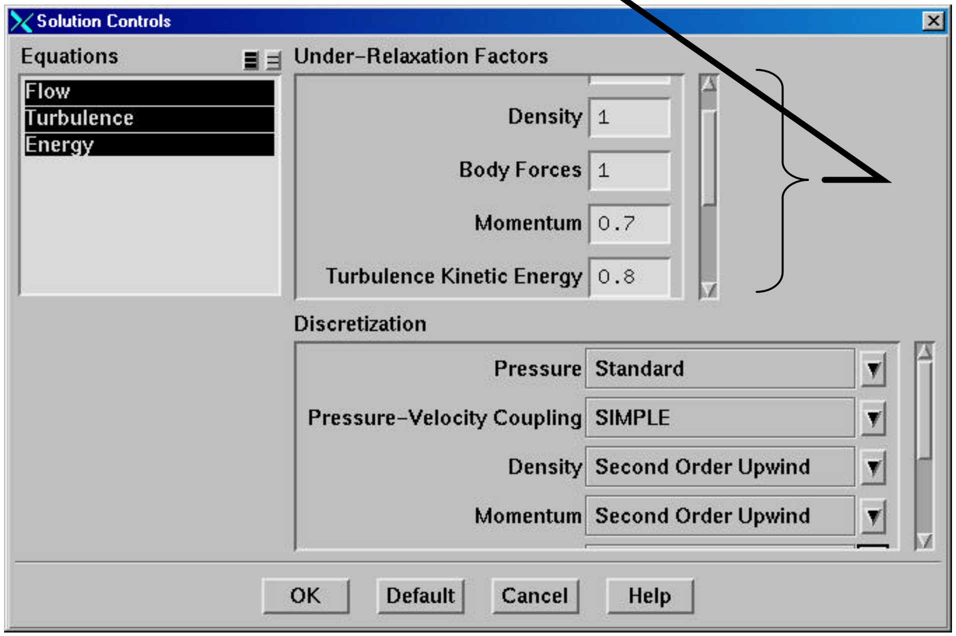
- Numerical instabilities can arise with an ill-posed problem, poor quality mesh, and/or inappropriate solver settings.
  - Exhibited as increasing (diverging) or “stuck” residuals.
  - Diverging residuals imply increasing imbalance in conservation equations.
  - Unconverged results can be misleading!
- Troubleshooting:
  - Ensure problem is well posed.
  - Compute an initial solution with a first-order discretization scheme.
  - Decrease under-relaxation for equations having convergence trouble (segregated).
  - Reduce Courant number (coupled).
  - Re-mesh or refine grid with high aspect ratio or highly skewed cells.



Continuity equation convergence trouble affects convergence of all equations.

Modifying Under-relaxation Factors

$$f_p = f_{p,old} + a \Delta f_p$$

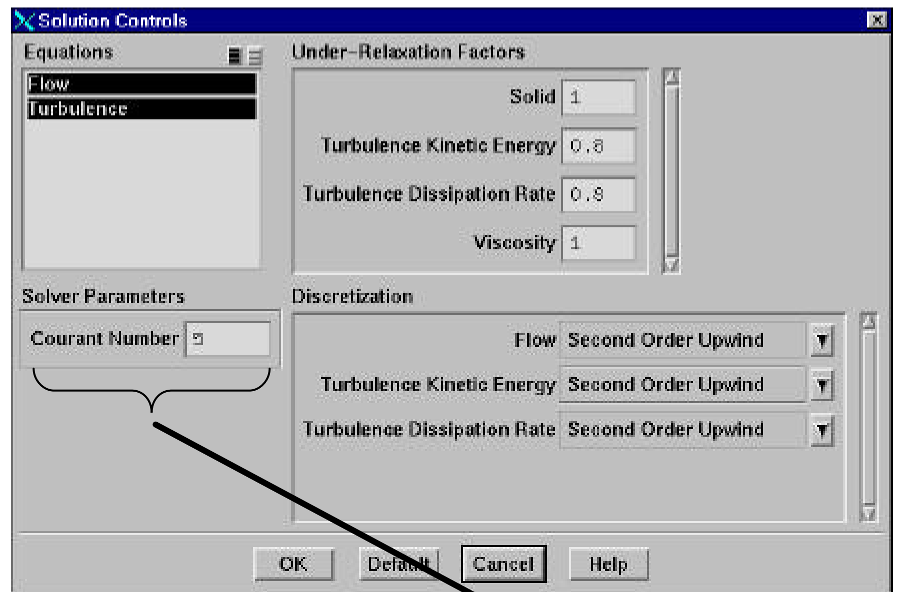


- Under-relaxation factor, *a*, is included to stabilize the iterative process for the segregated solver.
- Use default under-relaxation factors to start a calculation.

  Solve → Controls → Solution...
- Decreasing under-relaxation for *momentum* often aids convergence.
  - Default settings are aggressive but suitable for wide range of problems.
  - ‘Appropriate’ settings best learned from experience.
- For coupled solvers, under-relaxation factors for equations *outside* coupled set are modified as in segregated solver.

Modifying the Courant Number

- Courant number defines a 'time step' size for steady-state problems.
  - A transient term is included in the coupled solver even for steady state problems.
- For coupled-explicit solver:
  - Stability constraints impose a maximum limit on Courant number.
    - Cannot be greater than 2.
      - Default value is 1.
    - Reduce Courant number when having difficulty converging.
- For coupled-implicit solver:
  - Courant number is not limited by stability constraints.
    - Default is set to 5.



$$\Delta t = \frac{(\mathrm{CFL})\Delta x}{u}$$

Accelerating Convergence

- Convergence can be accelerated by:
  - Supplying good initial conditions
    - Starting from a previous solution.
  - Increasing under-relaxation factors or Courant number
    - Excessively high values can lead to instabilities.
    - Recommend saving case and data files before continuing iterations.
  - Controlling multigrid solver settings.
    - Default settings define robust Multigrid solver and typically do not need to be changed.

Accuracy

- A converged solution is not necessarily an accurate one.
  - Solve using 2nd order discretization.
  - Ensure that solution is grid-independent.
    - Use adaption to modify grid.
- If flow features do not seem reasonable:
  - Reconsider physical models and boundary conditions.
  - Examine grid and re-mesh.

Mesh Quality and Solution Accuracy

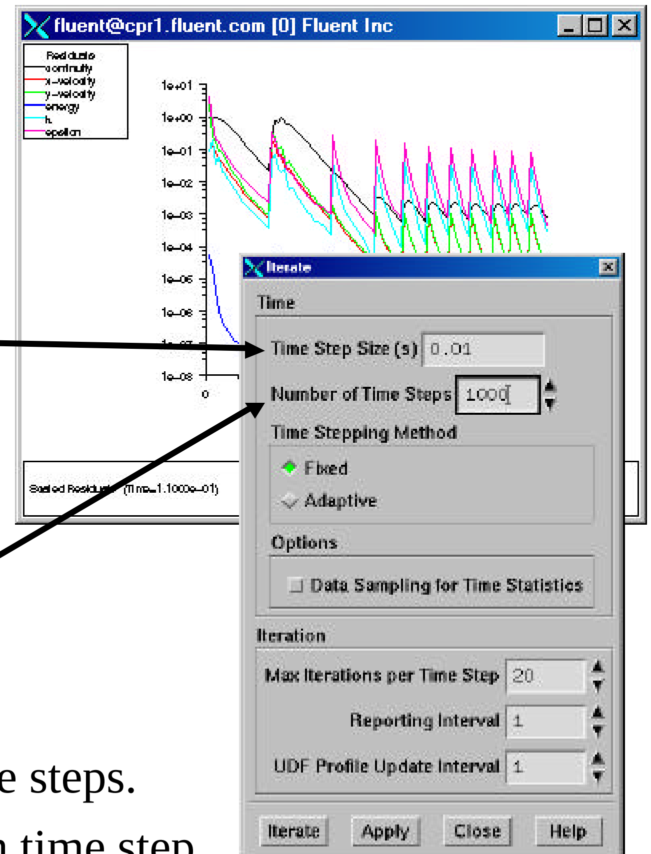
- Numerical errors are associated with calculation of cell gradients and cell face interpolations.
- These errors can be contained:
  - Use higher order discretization schemes.
  - Attempt to align grid with flow.
  - Refine the mesh.
    - Sufficient mesh density is necessary to resolve salient features of flow.
      - Interpolation errors decrease with decreasing cell size.
    - Minimize variations in cell size.
      - Truncation error is minimized in a uniform mesh.
      - Fluent provides capability to adapt mesh based on cell size variation.
    - Minimize cell skewness and aspect ratio.
      - In general, avoid aspect ratios higher than 5:1 (higher ratios allowed in b.l.).
      - Optimal quad/hex cells have bounded angles of 90 degrees
      - Optimal tri/tet cells are equilateral.

Determining Grid Independence

- When solution no longer changes with further grid refinement, you have a “grid-independent” solution.
- Procedure:
  - Obtain new grid:
    - Adapt
      - Save original mesh before adapting.
        - If you know where large gradients are expected, concentrate the original grid in that region, e.g., boundary layer.
      - Adapt grid.
        - Data from original grid is automatically interpolated to finer grid.
    - file → write-bc and file → read-bc facilitates set up of new problem
    - file → reread-grid and File → Interpolate...
  - Continue calculation to convergence.
  - Compare results obtained w/different grids.
  - Repeat procedure if necessary.

Unsteady Flow Problems

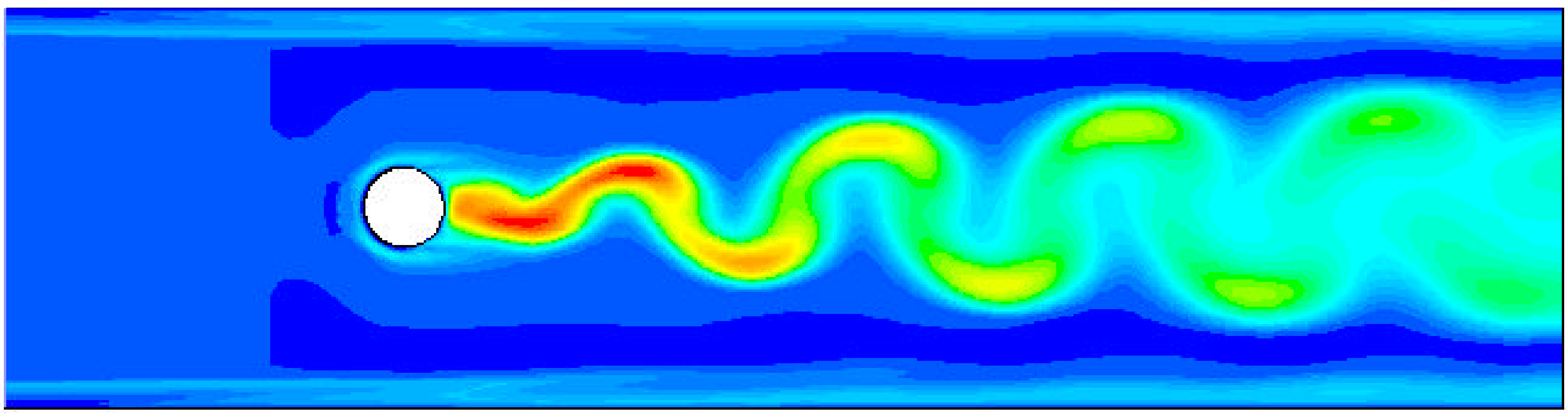
- Transient solutions are possible with both segregated and coupled solvers.
  - Solver iterates to convergence at each time level, then advances automatically.
  - Solution Initialization defines initial condition and must be realistic.
- For segregated solver:
  - Time step size, Δt, is input in Iterate panel.
    - Δt must be small enough to resolve time dependent features and to ensure convergence within 20 iterations.
    - May need to start solution with small Δt.
  - Number of time steps, N, is also required.
    - N*Δt = total simulated time.
  - To iterate without advancing time step, use '0' time steps.
  - PISO may aid in accelerating convergence for each time step.



Summary

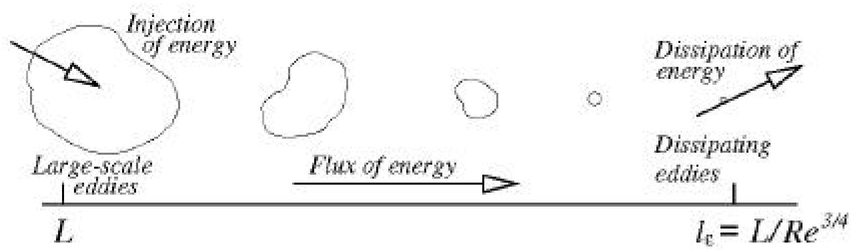
- Solution procedure for the segregated and coupled solvers is the same:
  - Calculate until you get a converged solution.
  - Obtain second-order solution (recommended).
  - Refine grid and recalculate until grid-independent solution is obtained.
- All solvers provide tools for judging and improving convergence and ensuring stability.
- All solvers provide tools for checking and improving accuracy.
- Solution accuracy will depend on the appropriateness of the physical models that you choose and the boundary conditions that you specify.

Modeling Turbulent Flows



What is Turbulence?

- Unsteady, irregular (aperiodic) motion in which transported quantities (mass, momentum, scalar species) fluctuate in time and space
  - Identifiable swirling patterns characterizes turbulent eddies.
  - Enhanced mixing (matter, momentum, energy, etc.) results
- Fluid properties exhibit random variations
  - Statistical averaging results in accountable, turbulence related transport mechanisms.
  - This characteristic allows for *Turbulence Modeling*.
- Wide range in size of turbulent eddies (scales spectrum).
  - Size/velocity of large eddies on order of mean flow.
    - derive energy from mean flow



Is the Flow Turbulent?

External Flows

$Re_x \geq 5 \times 10^5$ along a surface

$Re_D \geq 20,000$ around an obstacle

Internal Flows

$Re_{D_h} \geq 2,300$

where $Re_L \equiv \dfrac{r\,UL}{m}$

$L = x, D, D_h,$ etc.

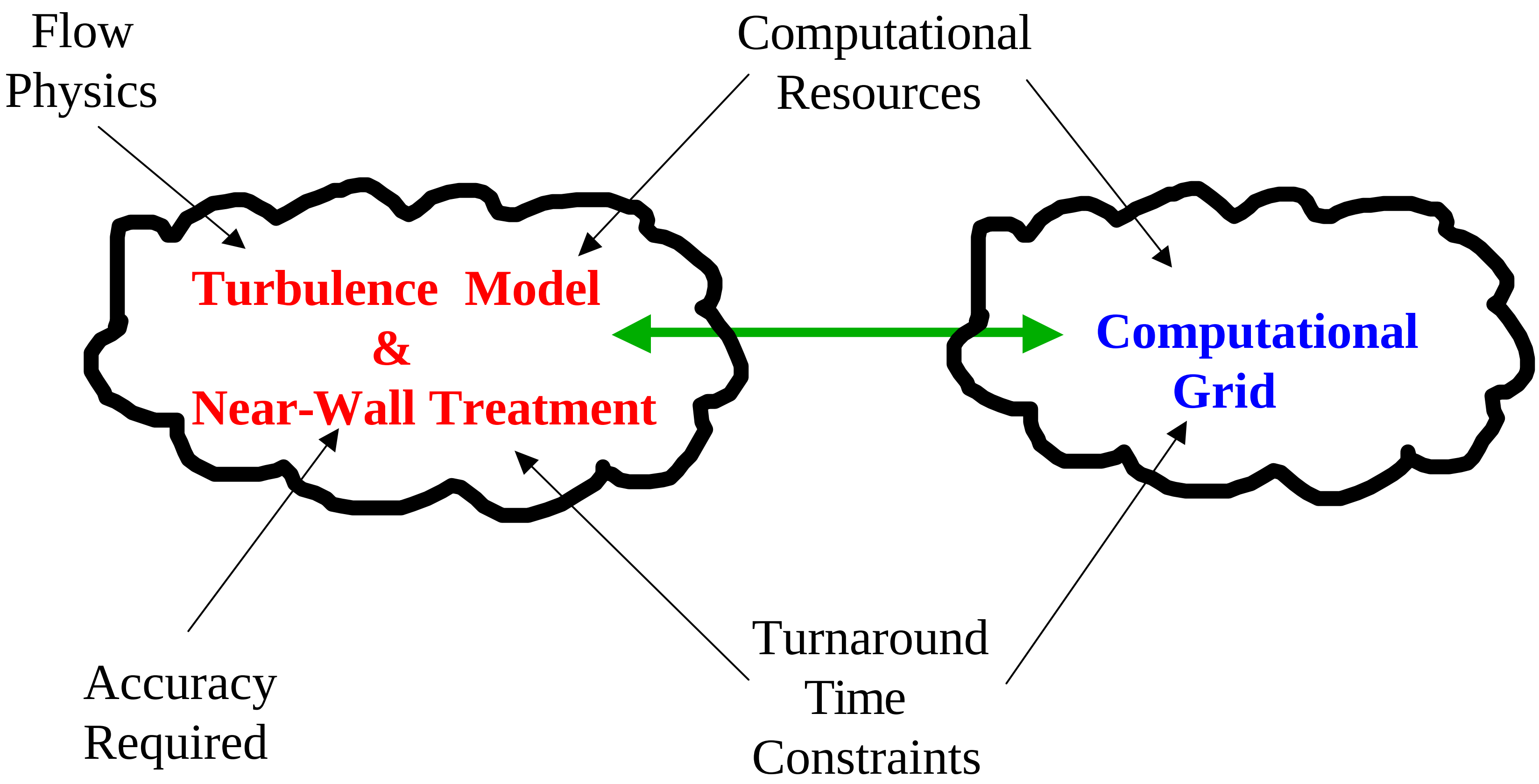
Other factors such as free-stream turbulence, surface conditions, and disturbances may cause earlier transition to turbulent flow.

Natural Convection

$Ra \geq 10^8 - 10^{10}$

where $Ra \equiv \dfrac{gb\Delta TL^3 r}{ma}$

Choices to be Made



Modeling Turbulence

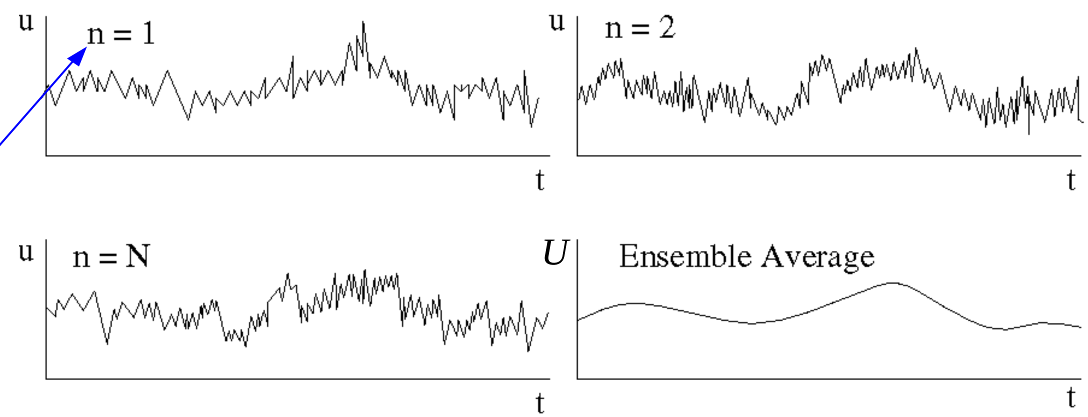
- Direct numerical simulation (DNS) is the solution of the time-dependent Navier-Stokes equations *without* recourse to modeling.
  - Mesh must be fine enough to resolve smallest eddies, yet sufficiently large to encompass complete model.
  - Solution is inherently unsteady to capture convecting eddies.
  - *DNS is only practical for simple low-Re flows*.
- The need to resolve the full spectrum of scales is not necessary for most engineering applications.
  - Mean flow properties are generally sufficient.
  - Most turbulence models resolve the mean flow.
- Many different turbulence models are available and used.
  - There is no single, universally reliable engineering turbulence model for wide class of flows.
  - Certain models contain more physics that may be better capable of predicting more complex flows including separation, swirl, etc.

Modeling Approaches

- 'Mean' flow can be determined by solving a set of modified equations.
- Two modeling approaches:
  - (1) Governing equations are *ensemble or time averaged* (RANS-based models).
    - Transport equations for mean flow quantities are solved.
    - All scales of turbulence are modeled.
    - If mean flow is unsteady, Δt is set by global unsteadiness.
  - (2) Governing equations are *spatially averaged* (LES).
    - Transport equations for 'resolvable scales.'
    - Resolves larger eddies; models smaller ones.
    - Inherently unsteady, Δt set by small eddies.
    - Resulting models requires more CPU time/memory and is not practical for the majority of engineering applications.
- Both approaches requires modeling of the scales that are averaged out.

RANS Modeling - Ensemble Averaging

- Imagine how velocity, temperature, pressure, etc. might vary in a turbulent flow field downstream of a valve that has been slightly perturbed:

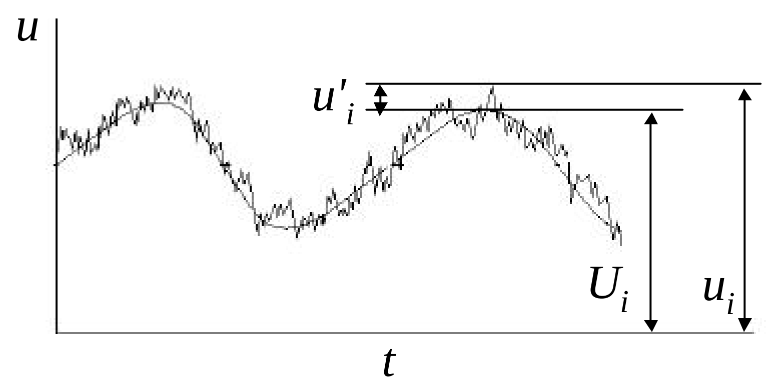


n identifies the 'sample' ID

- Ensemble averaging may be used to extract the mean flow properties from the instantaneous properties.

$$U_i(\vec{x},t) = \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} u_i^{(n)}(\vec{x},t)$$

$$u_i(\vec{x},t) = U_i(\vec{x},t) + u_i'(\vec{x},t)$$



Turbulence Models in Fluent

Zero-Equation Models

One-Equation Models

Spalart-Allmaras

Two-Equation Models

Standard *k*-*e*

RNG *k*-*e*

Realizable *k*-*e*

Standard *k*-w

SST *k*-w

Reynolds-Stress Model

Large-Eddy Simulation

Direct Numerical Simulation

RANS-based models

Available in FLUENT 6

Increase Computational Cost Per Iteration

Large Eddy Simulation (LES)

- Motivation:
  - Large eddies:
    - Mainly responsible for transport of momentum, energy, and other scalars, directly affecting the mean fields.
    - Anisotropic, subjected to history effects, and flow-dependent, i.e., strongly dependent on flow configuration, boundary conditions, and flow parameters.
  - *Small eddies* tend to be more isotropic, less flow-dependent, and hence more amenable to modeling.
- Approach:
  - LES resolves large eddies and models only small eddies.
  - Equations are similar in form to RANS equations
    - Dependent variables are now spatially averaged instead of time averaged.
- Large computational effort
  - Number of grid points, $N_{LES} \propto \mathrm{Re}_{u_t}^2$
  - Unsteady calculation

Summary: Turbulence Modeling Guidelines

- Successful turbulence modeling requires engineering judgement of:
  - Flow physics
  - Computer resources available
  - Project requirements
    - Accuracy
    - Turnaround time
  - Turbulence models & near-wall treatments that are available
- Modeling Procedure
  - Calculate characteristic *Re* and determine if Turbulence needs modeling.
  - Estimate wall-adjacent cell centroid y^+ first before generating mesh.
  - Begin with SKE (standard *k-e*) and change to RNG, RKE, SKO, or SST if needed.
  - Use RSM for highly swirling flows.
  - Use wall functions unless low-*Re* flow and/or complex near-wall physics are present.