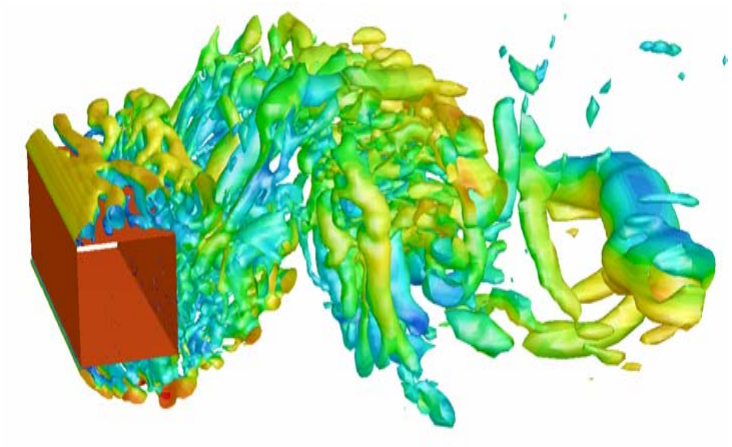
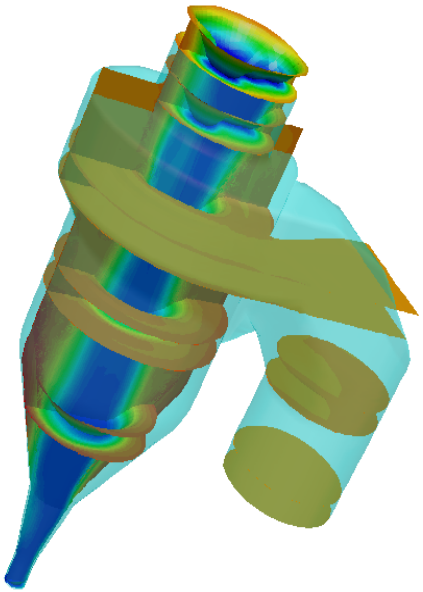
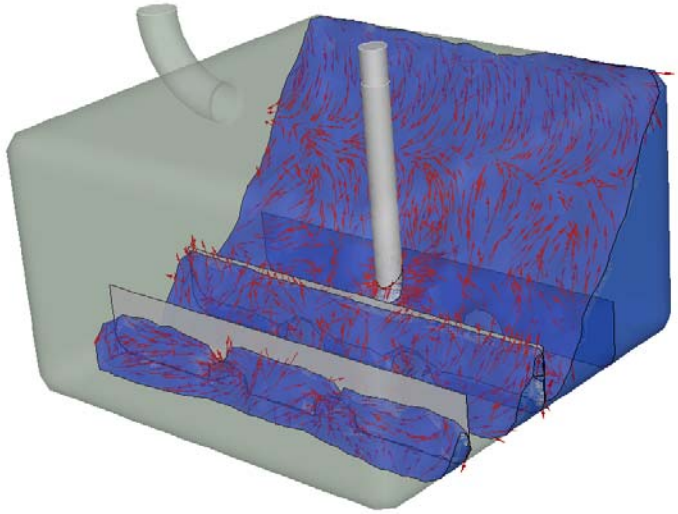
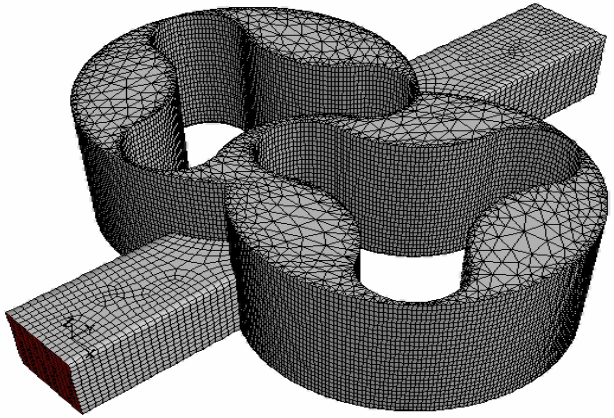
Fluent 6.2
Software Capabilities

Basic Training Course



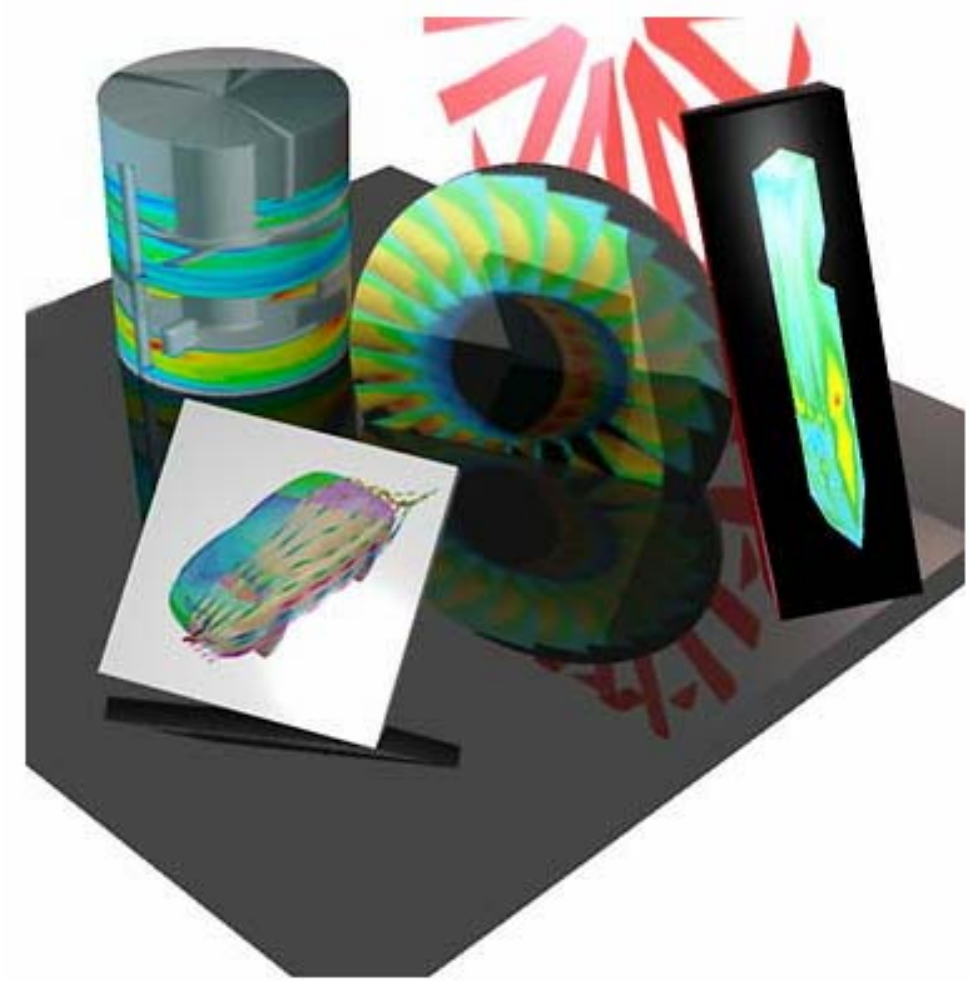






FLUENT®
The Right Answer in CFD™

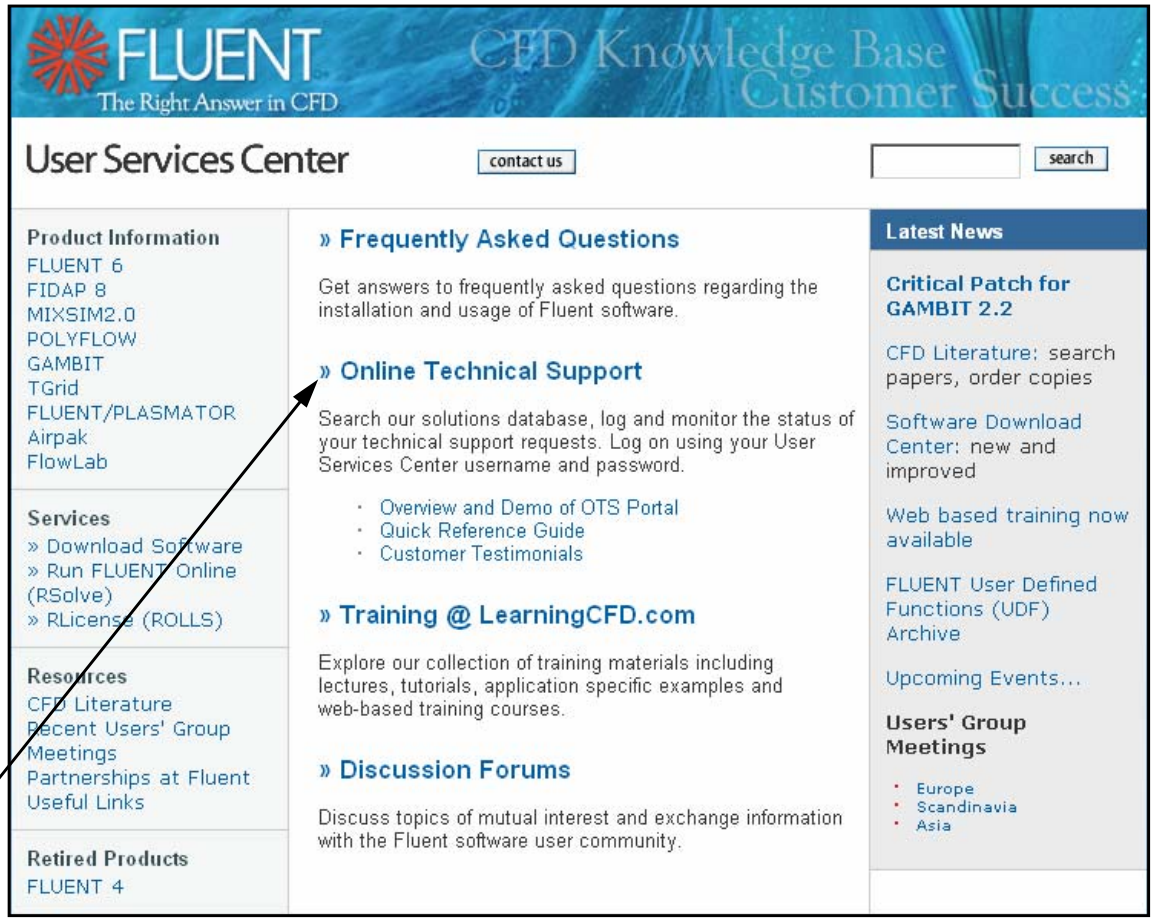
Table of Content



Fluent Inc.

Fluent User Services Center

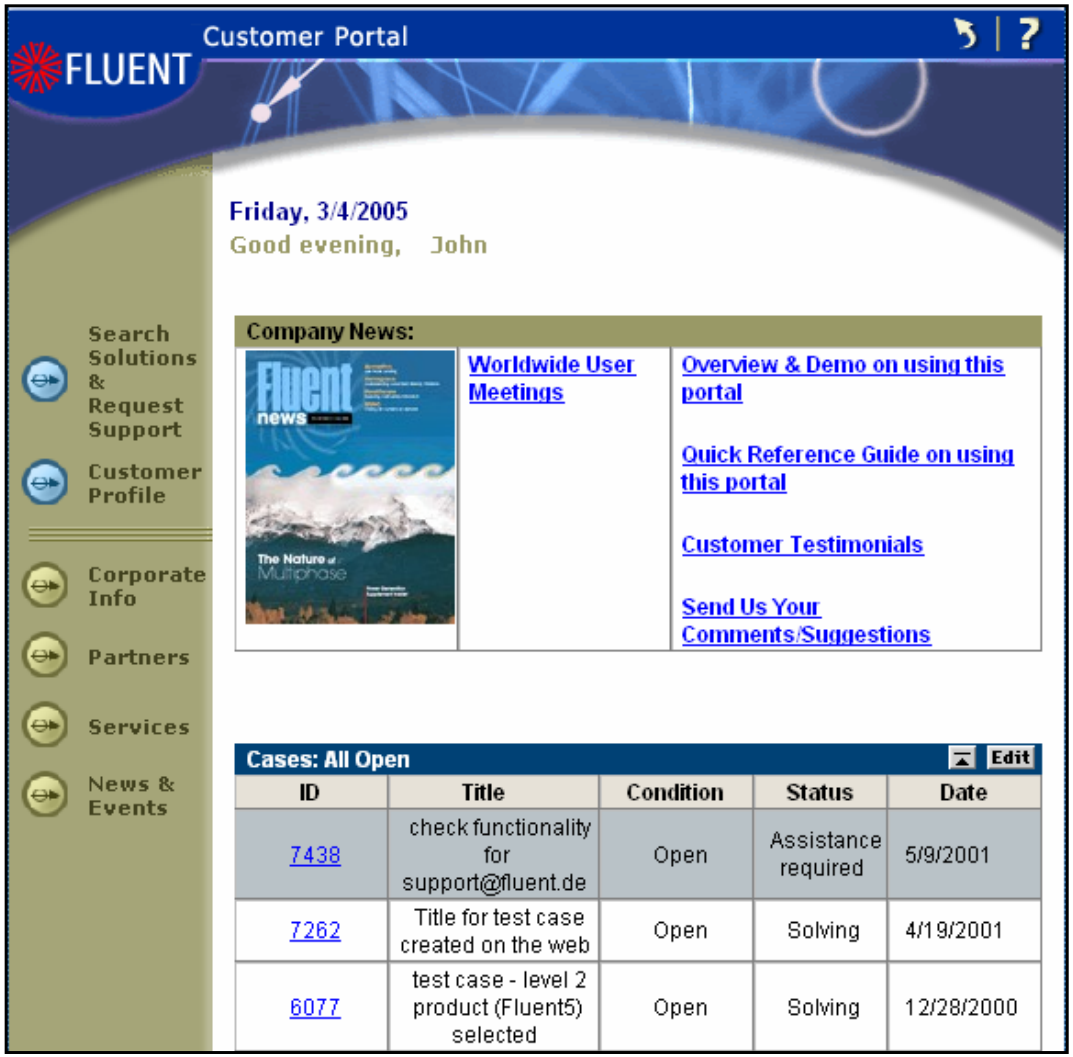
- **www.fluentusers.com**
  - Please register today!
  - Services Include
    - Release Information
    - Download Updates
    - Documentation
    - Supported Platforms
    - Defects/Workarounds
    - Presentations
    - Training
    - Online Technical Support
      - Quick Reference Guide
      - Overview and Demo via web based training



http://www.fluentusers.com/

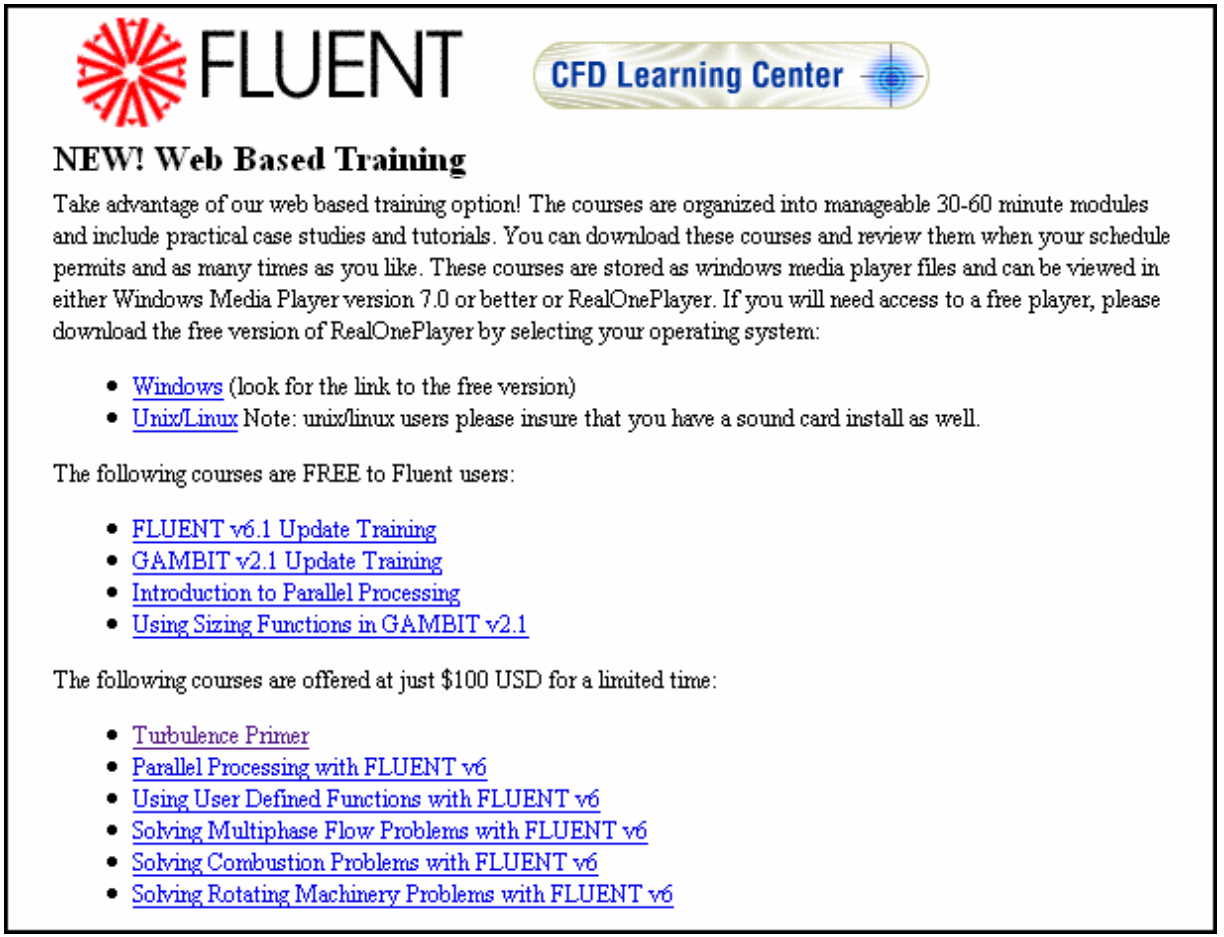
Online Technical Support

- Access from USC
  - Same Account as USC
  - Services Include
    - Find Solutions
    - Log a Support Request
    - Monitor Status
    - Automatic Updates
  - Web Based Training Module Available
- ftp files to/from support
  - ftp to ftp.fluent.com
  - log on as ftp and use email address for password
  - cd to incoming/***xxx*** or outgoing/***xxx*** and put files in binary mode.
    - ***xxx*** = support engineer initials



Web Based Trainings

- Web based training courses available from our LearningCFD.com web site.
- Some courses are free, others have a nominal charge.
- Modules can be downloaded to your computer and viewed as many times as you like
- More modules are planned and will be available in the near future


FLUENT CFD Learning Center

NEW! Web Based Training

Take advantage of our web based training option! The courses are organized into manageable 30-60 minute modules and include practical case studies and tutorials. You can download these courses and review them when your schedule permits and as many times as you like. These courses are stored as windows media player files and can be viewed in either Windows Media Player version 7.0 or better or RealOnePlayer. If you will need access to a free player, please download the free version of RealOnePlayer by selecting your operating system:

- Windows (look for the link to the free version)
- Unix/Linux Note: unix/linux users please insure that you have a sound card install as well.

The following courses are FREE to Fluent users:

- FLUENT v6.1 Update Training
- GAMBIT v2.1 Update Training
- Introduction to Parallel Processing
- Using Sizing Functions in GAMBIT v2.1

The following courses are offered at just $100 USD for a limited time:

- Turbulence Primer
- Parallel Processing with FLUENT v6
- Using User Defined Functions with FLUENT v6
- Solving Multiphase Flow Problems with FLUENT v6
- Solving Combustion Problems with FLUENT v6
- Solving Rotating Machinery Problems with FLUENT v6

http://learningcfd.com/login/online/index.htm

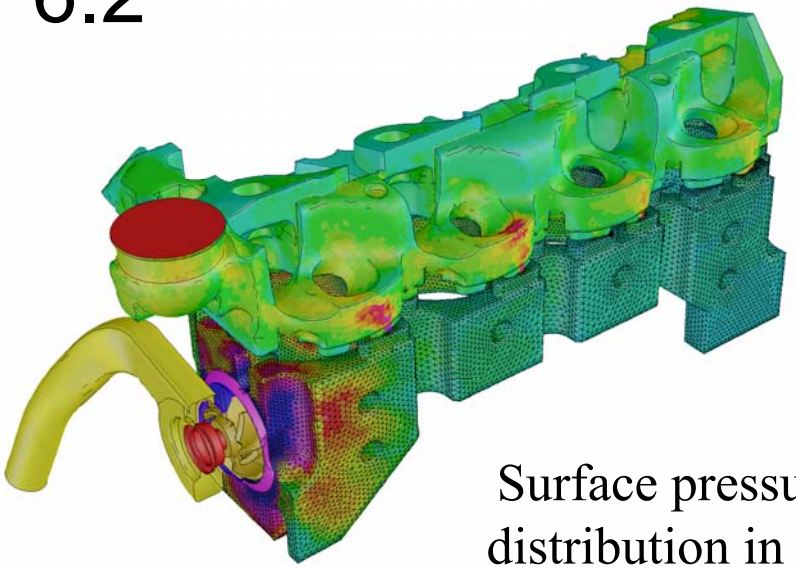
User Group Meetings

- Attend the annual UGM and:
  - meet with the users and staff of Fluent
  - attend short-courses
  - learn of other Fluent applications presented by users
  - provide input to future development of software

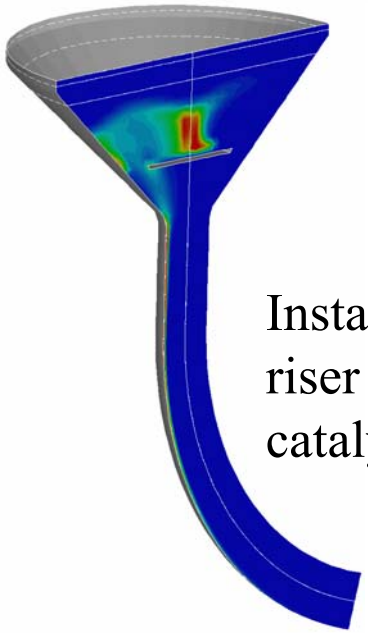
- Worldwide User Group Meetings:
  - USA (Dearborn, MI)
    - typ. Early-June
  - European Meetings
    - typ. Mid-September through early October
  - Asia-Pacific Meetings
    - typ. Mid-October through early November

FLUENT 6.2

- Fluent 6.2 applications:
  - External/internal automotive flows and in-cylinder flows
  - High speed aerodynamics
  - Rocket flows
  - Turbomachinery
  - Reactor Vessels
  - Cyclones
  - Bubble Columns
  - Mixing tanks
  - Fluidized Beds
  - Flow-induced noise prediction
  - Dynamic Mesh
  - *many more …*



Surface pressure distribution in an automotive engine cooling jacket.

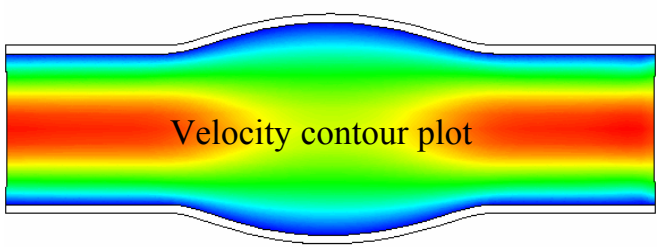


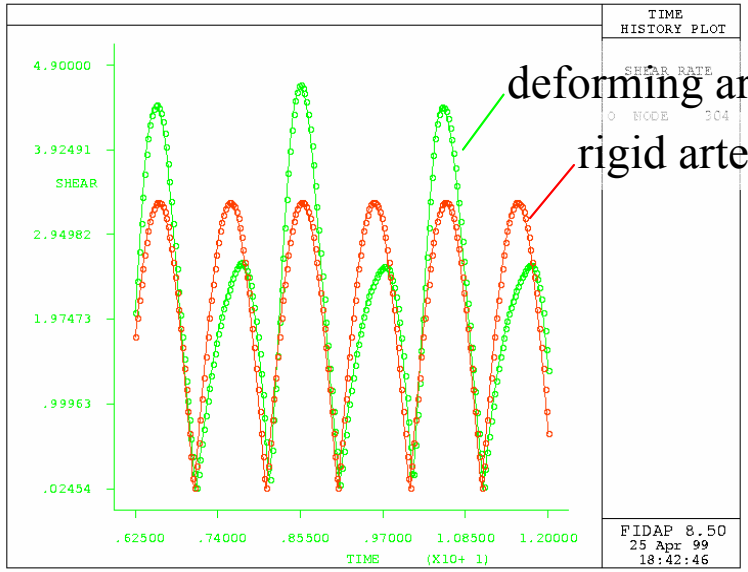
Instantaneous solids concentration in a riser indicating uniform distribution of catalyst at the riser head.

FIDAP

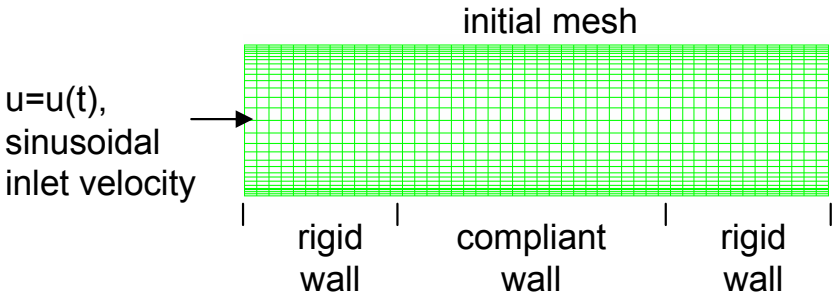
- FIDAP Applications:
  - Polymer processing: non-Newtonian flow in extrusion dies
  - Thin film coating flows
  - Biomedical: oxygenators, blood pumps, deforming arteries
  - Semiconductor crystal growth
  - Other metal, glass, and chemical processing problems

Example: Pulsatile flow in an artery with a compliant vein graft.





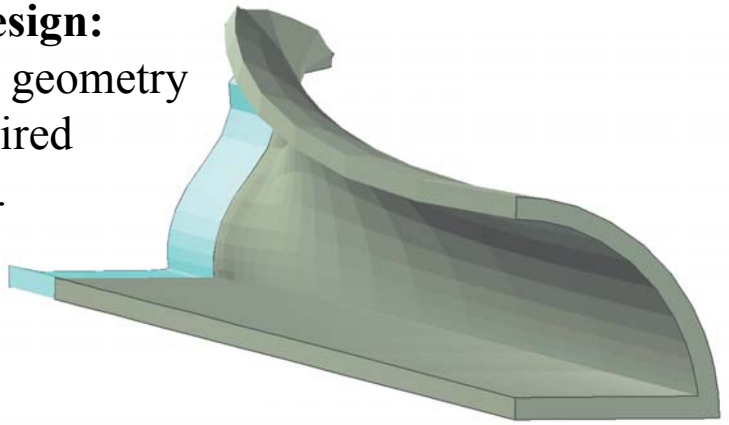
Time history plot of wall shear rate- Deformations cannot be neglected!



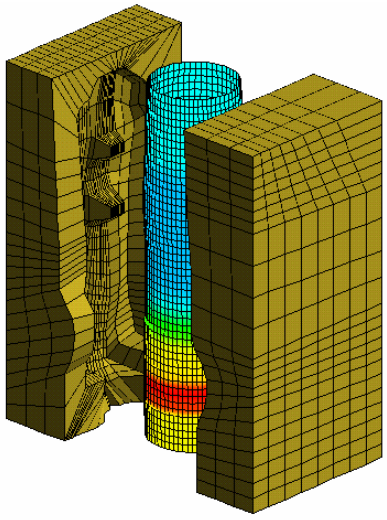
POLYFLOW

- FEM solver for laminar, viscous flows for complex rheologies and free surface
  - POLYFLOW Applications:
    - Extrusion, coextrusion, die design
    - Blow molding, thermoforming
    - Film casting, glass sheet forming/stretching, fiber drawing
    - Chemical reactions, foaming
    - Viscoelastic flows ("memory effects")

Inverse Die Design:
Determines die geometry based upon desired extruded shape.



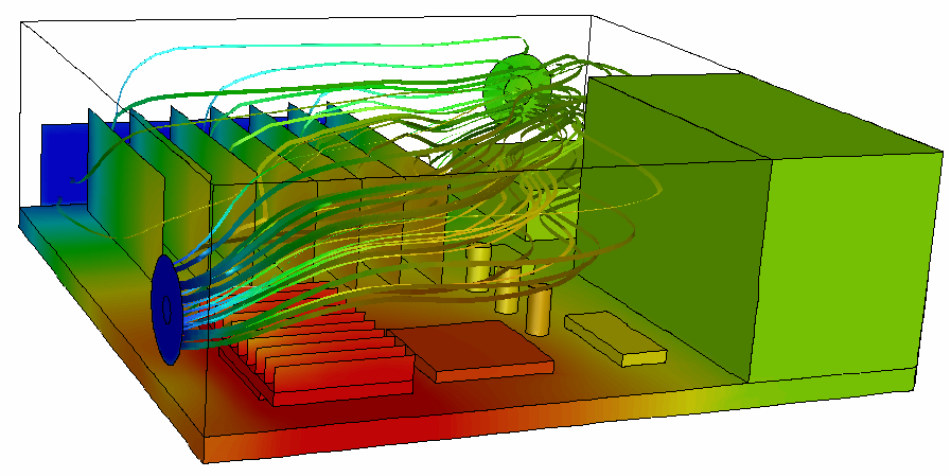
Requested part shape and calculated die lip shape for a rubber car door seal.



Blow molding simulation of a gas tank using the membrane element.

IcePak

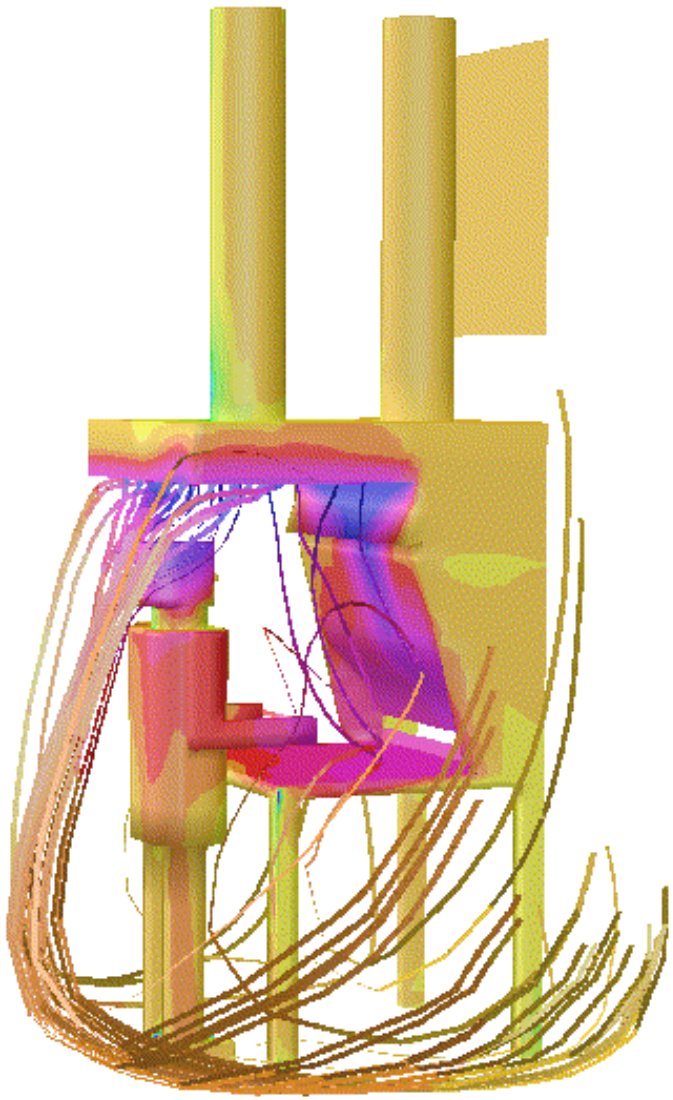
- IcePak is focused on electronics cooling design:
  - Cooling airflow, heat conduction, convection and radiation heat transfer
- The user interface and automatic meshing are tailored for applications such as:
  - Cabinet design
  - Fan placement
  - Board-level design
  - Heat sink evaluation



Flow pathlines and temperature distribution in a fan-cooled computer cabinet.

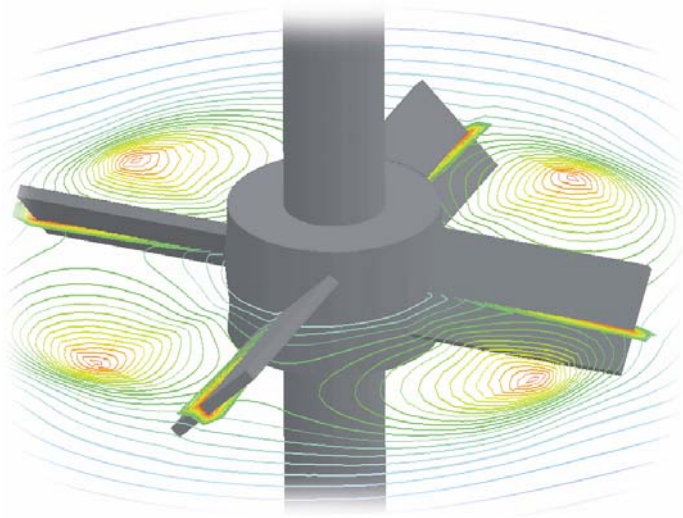
Airpak

- Simplifies the design and analysis of ventilation systems
- Accurate, quick, and easy-to-use design tool that empowers designers and professionals, without extensive backgrounds in computer applications, to utilize the powers of advanced CFD tools
- Optimize your designs or pinpoint problems based on accurate predictions of airflow patterns, thermal conditions, comfort conditions, and/or contamination control effectiveness



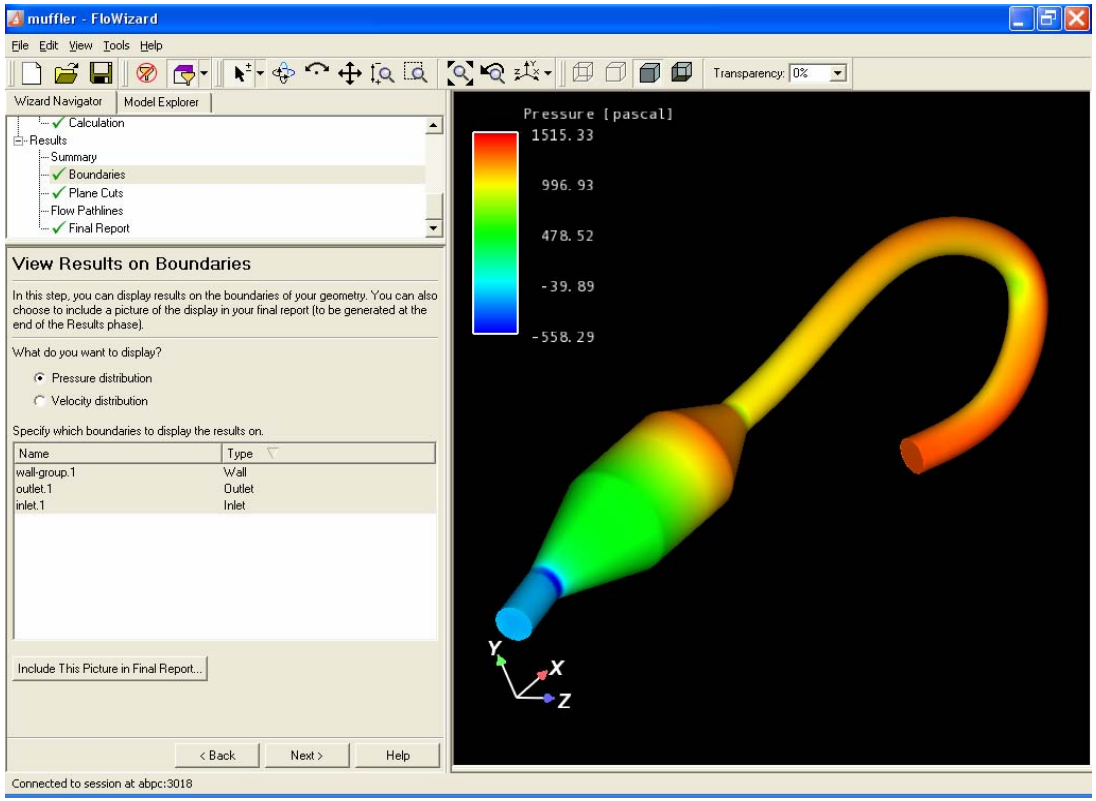
MixSim

- MixSim is a specialized user interface that allows quick and easy set-up of mixing tank simulations.
- The tank size, bottom shape, baffle configuration, number and type of impellers, etc. are specified directly.
- The mesh and complete problem definition are then automatically created.
- Other features include:
  - Impeller libraries from leading equipment manufacturers
  - Transient sliding mesh, steady-state multiple reference frame models
  - Non-Newtonian rheology



FloWizard

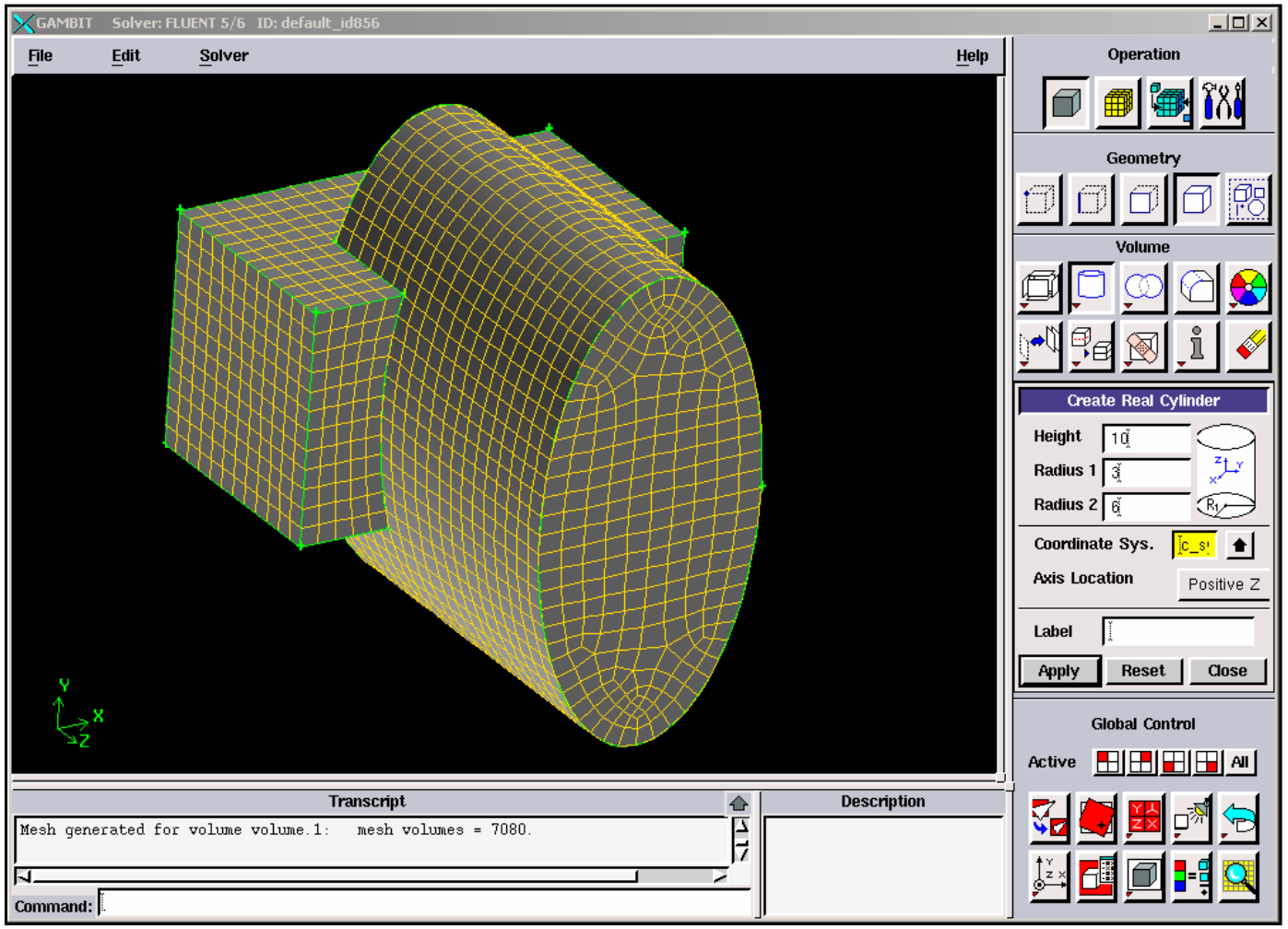
- Our first general purpose CFD product for non-specialists driven by FLUENT and Gambit .
- FloWizard's focus is on ease of use and automation:
  - It is a highly-automated, "first pass" simulation tool for use in basic flow and heat transfer calculations.
  - A Wizard-based interface guides the user through all the steps of a CFD analysis, from problem set-up to post-processing.
  - The user is insulated from specialized CFD parameters, such as discretization and turbulence model choices.



FloWizard

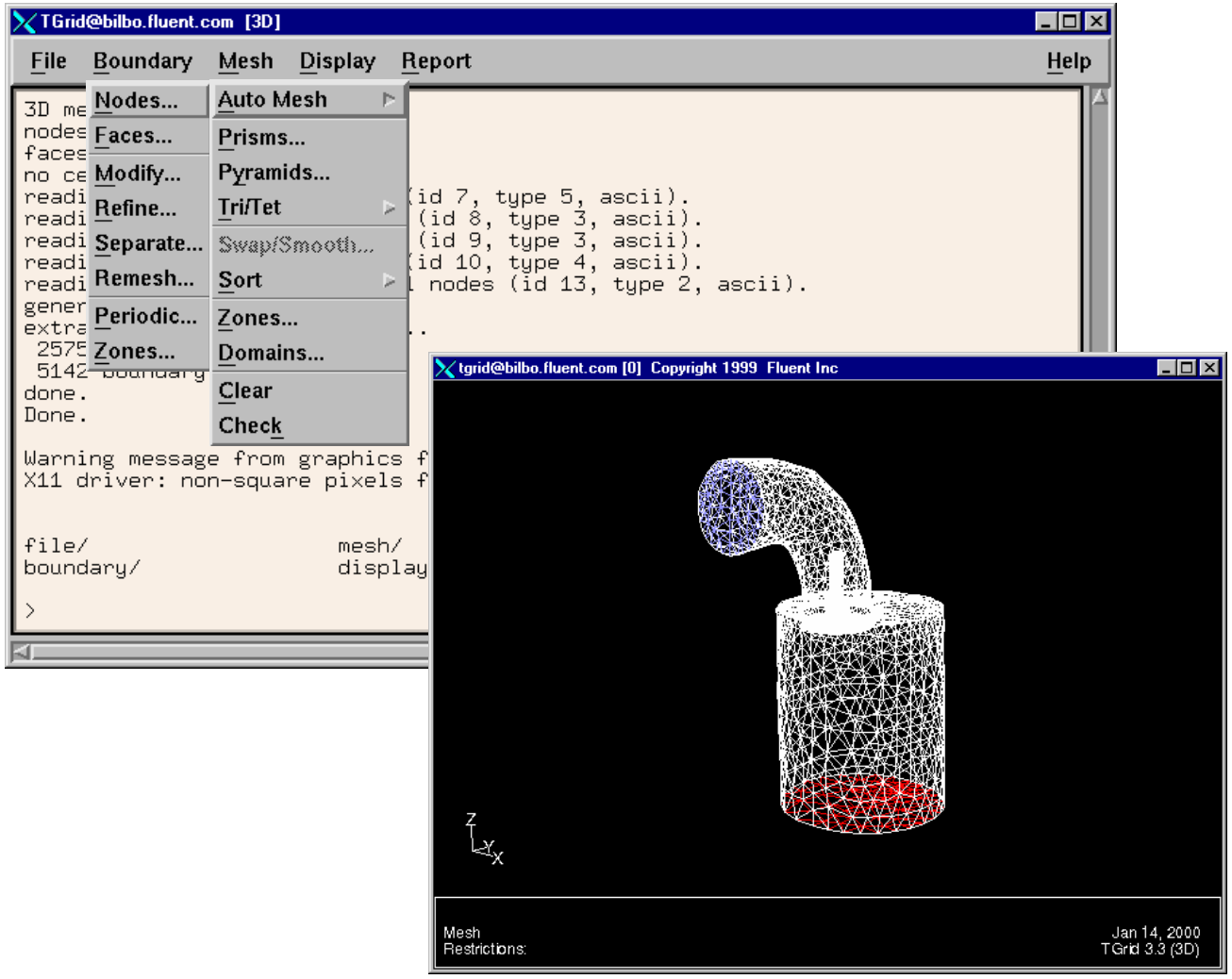
Pre-processor: Gambit

- A single, integrated pre-processor for CFD analysis.
  - Geometry creation
  - Mesh generation
  - Mesh quality examination
  - Boundary zone assignment



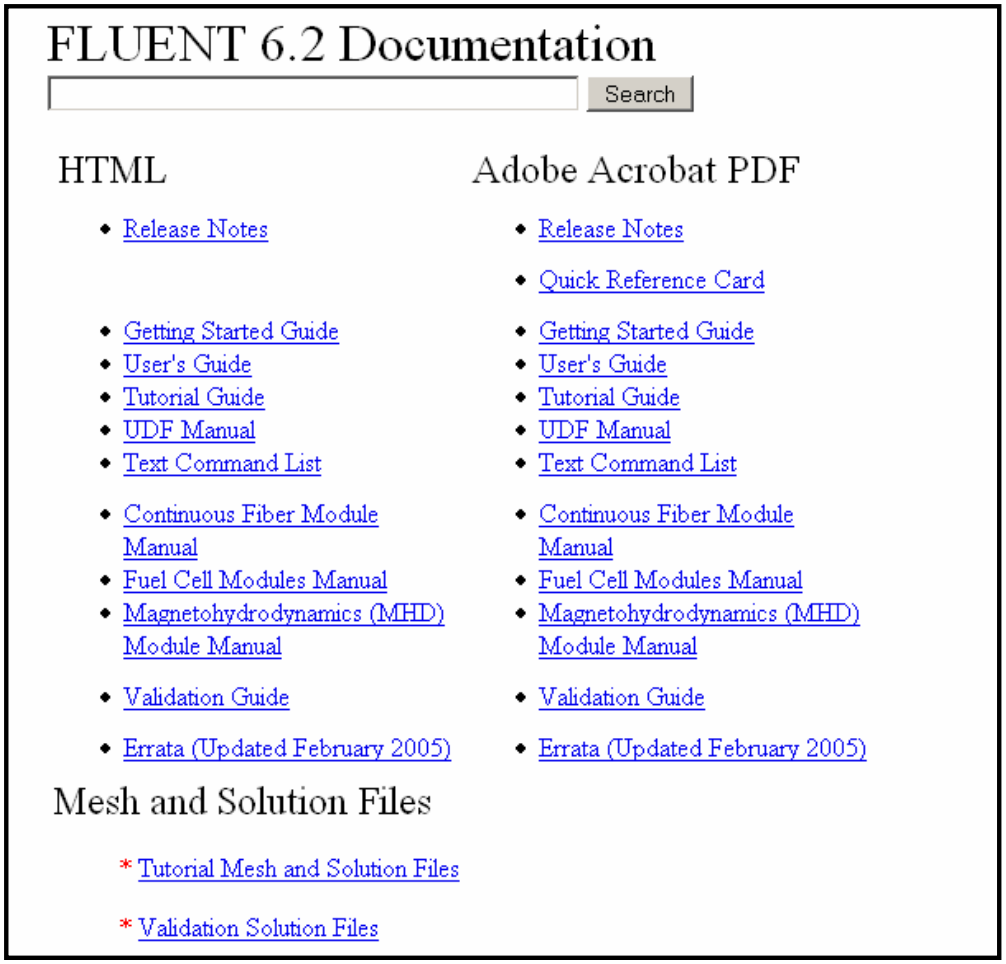
Pre-processor: TGrid

- A pre-processor for tet/hybrid mesh generation.
- Useful when starting with triangular surface mesh.



Documentation

- Documentation for all products available at www.fluentusers.com
- Separate CD for each product (e.g., FLUENT 6, TGrid, etc.) containing all the manuals for that product.
- Two formats available:
  - HTML
    - for general viewing, searching, limited printing
  - Adobe Acrobat PDF
    - for high quality printing of one or many pages



Fluent 6.2 Documentation Web Page at www.fluentusers.com

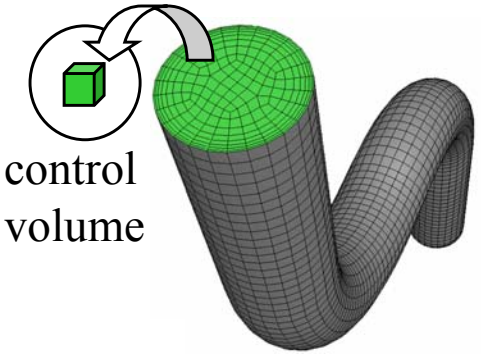
Introduction to CFD Analysis

What is CFD?

- Computational Fluid Dynamics (CFD) is the science of predicting fluid flow, heat and mass transfer, chemical reactions, and related phenomena by solving numerically the set of governing mathematical equations.
  - Conservation of mass, momentum, energy, species, ...
- The results of CFD analyses are relevant in:
  - conceptual studies of new designs
  - detailed product development
  - troubleshooting
  - redesign
- CFD analysis complements testing and experimentation.
  - Reduces the total effort required in the experiment design and data acquisition

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 is discretized into a finite set of control volumes (mesh).

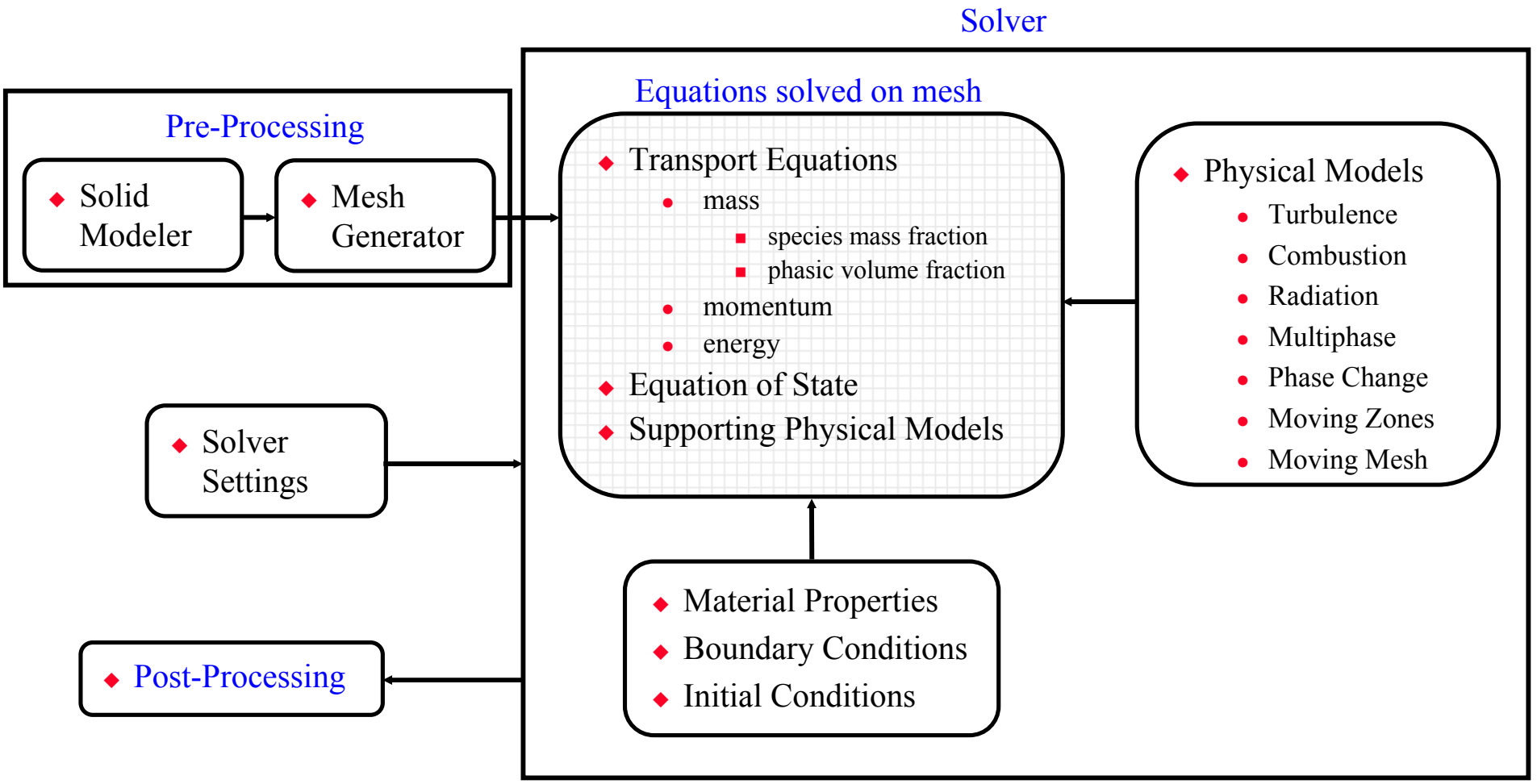
$$\underbrace{\frac{\partial}{\partial t}\int_V \rho\phi dV}_{\text{unsteady}} + \underbrace{\oint_A \rho\phi\mathbf{V}\cdot d\mathbf{A}}_{\text{convection}} = \underbrace{\oint_A \Gamma\nabla\phi\cdot d\mathbf{A}}_{\text{diffusion}} + \underbrace{\int_V S_\phi dV}_{\text{generation}}$$

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

Partial differential equations are discretized into a system of algebraic equations.

- *All* algebraic equations are then solved numerically to render the solution 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

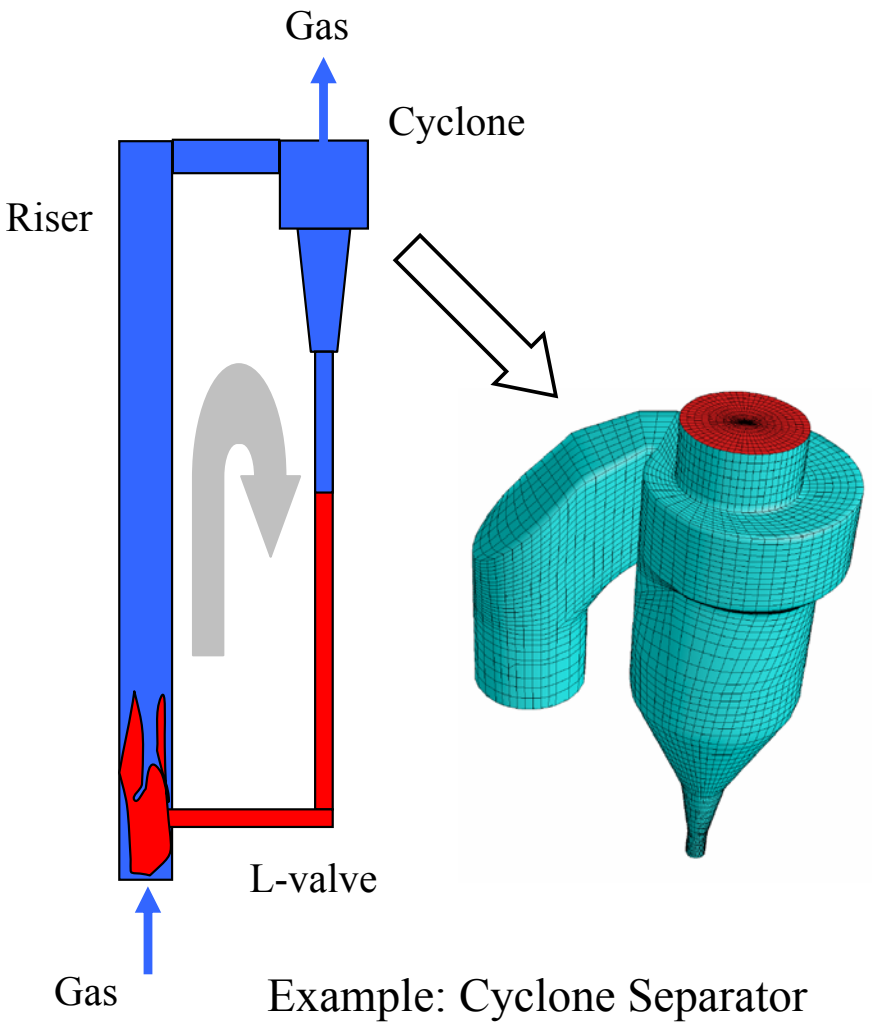
- Problem Identification and Pre-Processing
  1. Define your modeling goals.
  2. Identify the domain you will model.
  3. Design and create the grid.

- 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

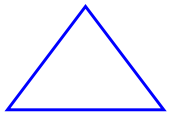
- How will you isolate a piece of the complete physical system?
- Where will the computational domain begin and end?
  - Do you have boundary condition information at these boundaries?
  - Can the boundary condition types accommodate that information?
  - Can you extend the domain to a point where reasonable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



Example: Cyclone Separator

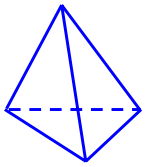
Design and Create the Grid

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

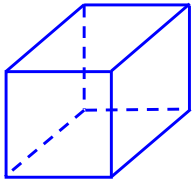


triangle

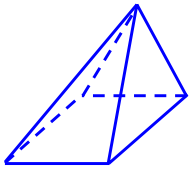
quadrilateral



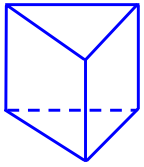
tetrahedron



hexahedron



pyramid

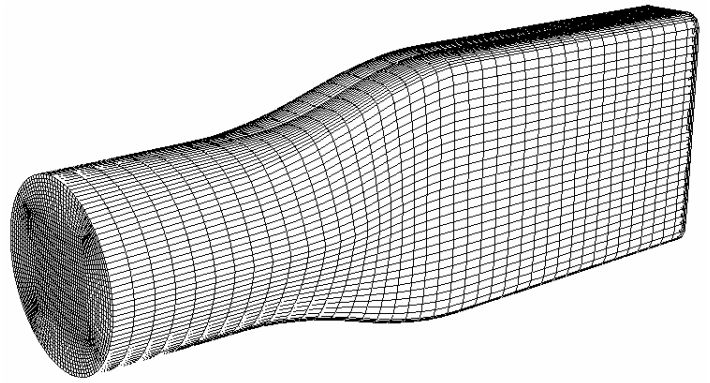


prism/wedge

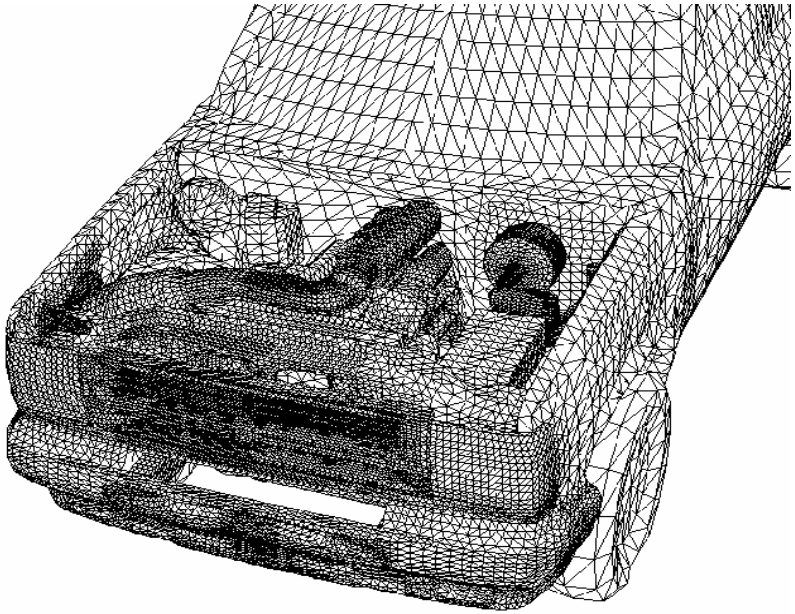
- Can you benefit from Mixsim, Icepak, or Airpak?
- Can you use a quad/hex grid or should you use a tri/tet grid or hybrid grid?
  - How complex is the geometry and flow?
  - Will you need a non-conformal interface?
- What degree of grid resolution is required in each region of the domain?
  - Is the 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 higher-quality solutions with fewer cells than a comparable tri/tet mesh.
  - Align the gridlines with the flow.

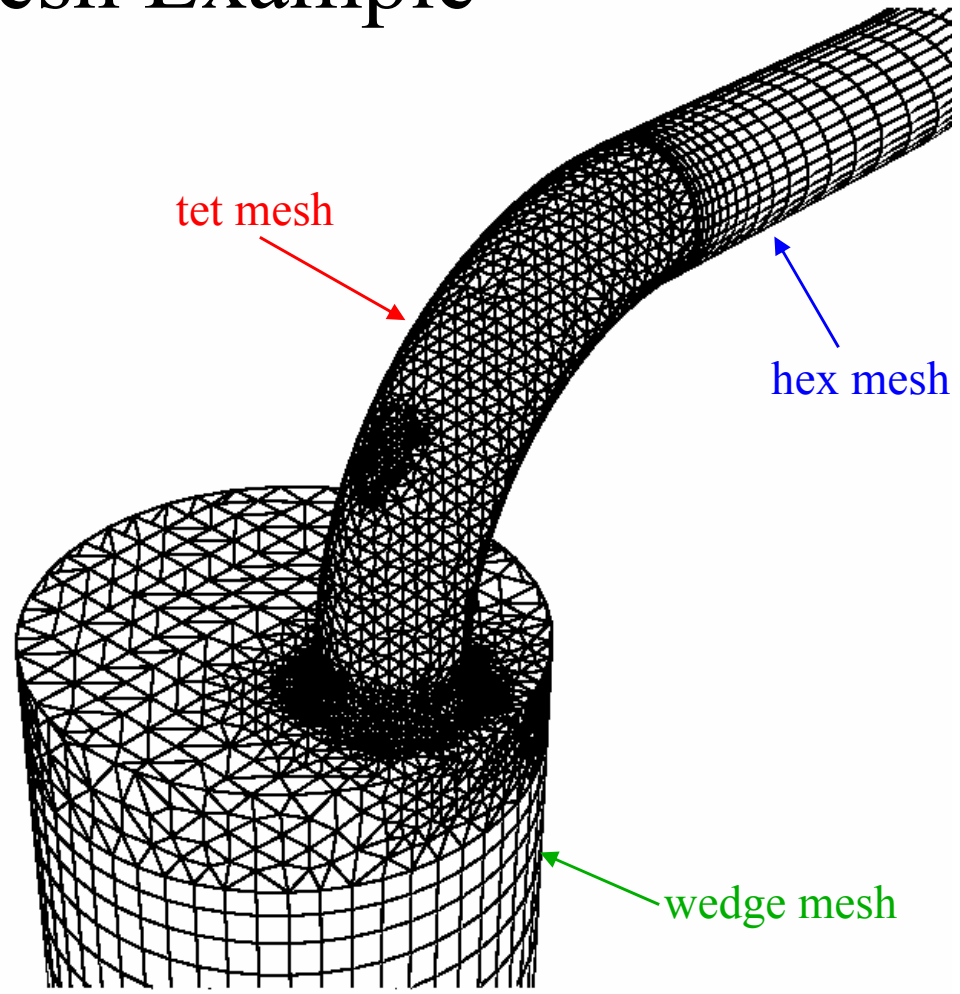


- For **complex** geometries, quad/hex meshes show no numerical advantage, and you can save meshing effort by using a tri/tet mesh.



Hybrid Mesh Example

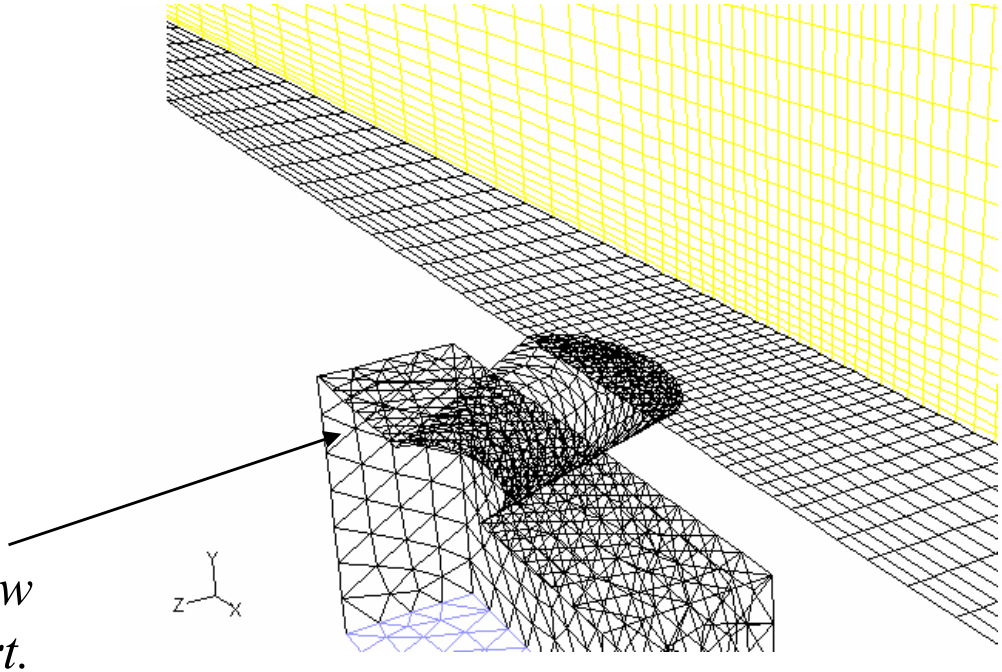
- Valve port grid
  - Specific regions can be meshed with different cell types.
  - Both efficiency and accuracy are enhanced relative to a hexahedral or tetrahedral mesh alone.
  - Tools for hybrid mesh generation are available in Gambit and TGrid.



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.

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

- 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

Examine the Results

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

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

- 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

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

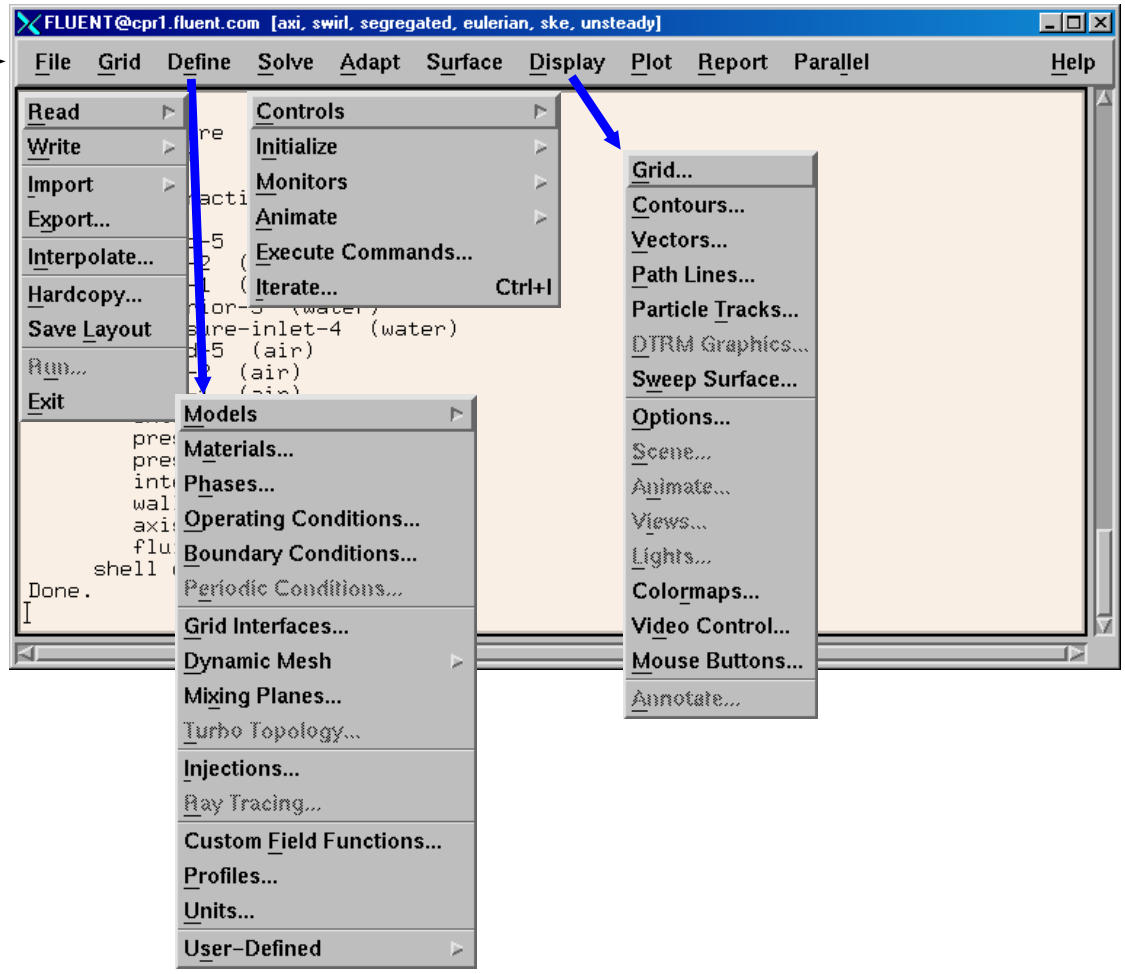
Navigating the PC at Fluent

- Login as fluent; password: fluent
- Directories
  - Session will start in D:\users\fluent
  - Change directory to and save work in D:\users\fluent\fluent
    - Or in the following areas:
      - Fluent tutorial mesh files are in D:\users\fluent\fluent\tut
      - Gambit tutorial mesh files are in D:\users\fluent\fluent\gambit\tut
- To start Fluent 6 / Gambit2:
  - **From startup menu: Programs → fluent inc → `fluent`**
  - **From command prompt: `fluent 2d or fluent 3d`**
- Your support engineer will save your work at the end of the week.
- !Note: It is recommended that you restart **`fluent`** for each tutorial to avoid mixing solver settings from different tutorials.

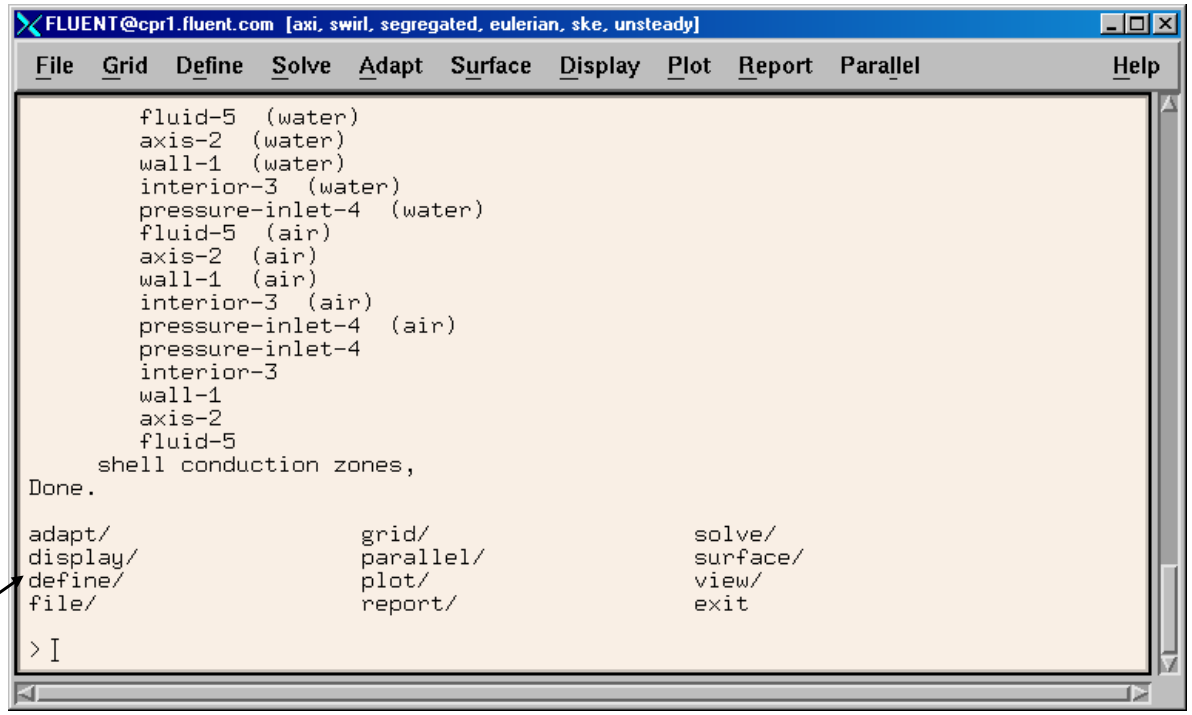
Solver Basics

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



Inputs to the Solver

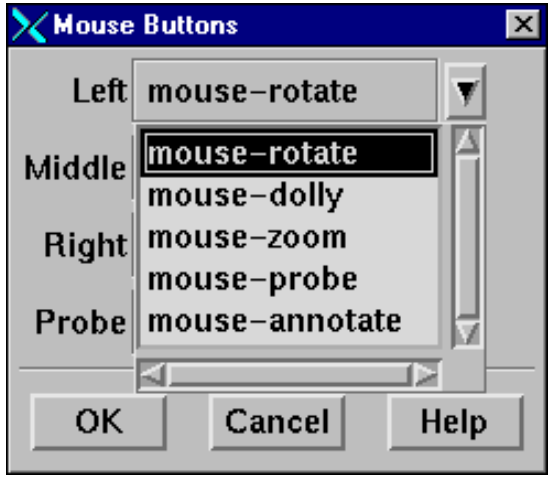


- GUI commands have a corresponding TUI command.
  - Advanced commands are only available through TUI.
  - 'Enter' displays command set at current level.
  - 'q' moves up one level.
- Journal/Transcript write capability.

Mouse Functionality

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

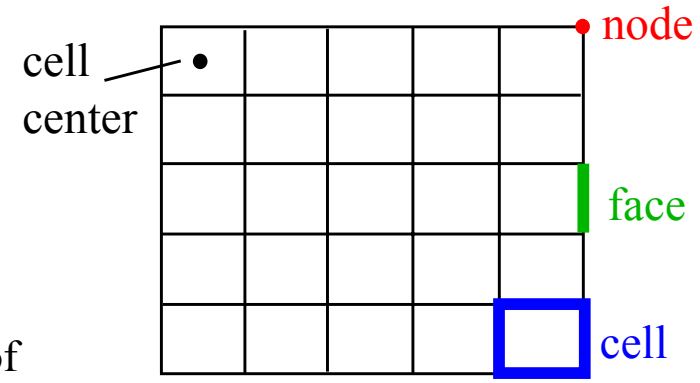
  Display → Mouse Buttons...



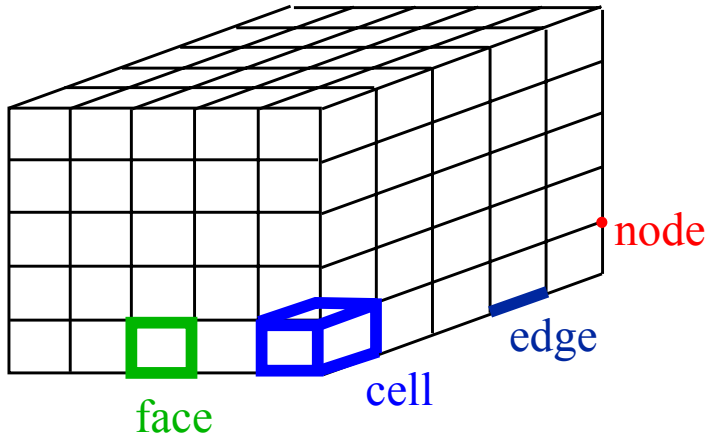
- Default Settings:
  - 2D Solver
    - Left button translates (dolly)
    - Middle button zooms
    - Right button selects/probes
  - 3D Solver
    - Left button rotates about 2-axes
    - Middle button zooms
      - Middle click on point in screen centers point in window
    - Right button selects/probes
- 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.

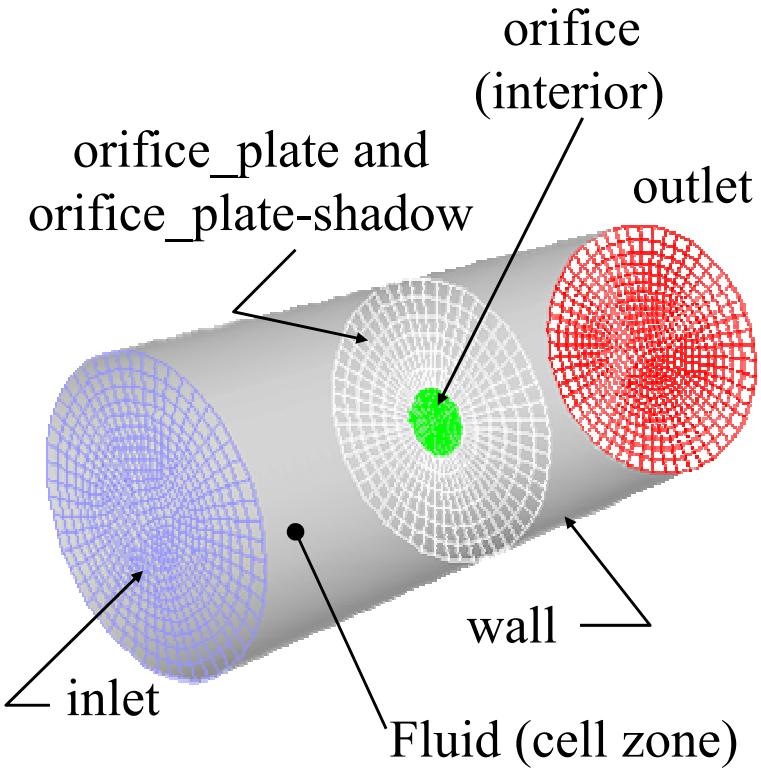


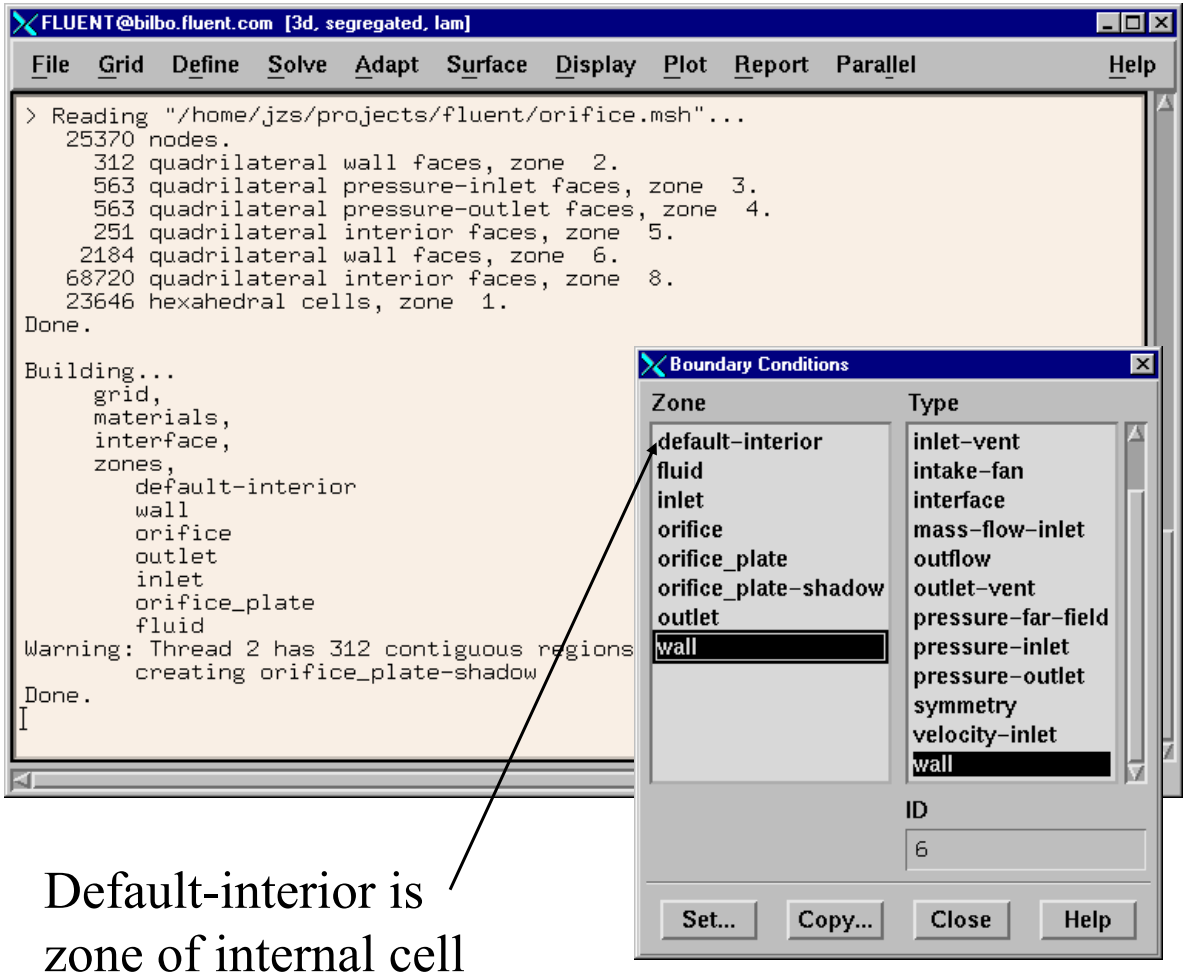
Simple 2D mesh



Simple 3D mesh

Reading Mesh: Zones



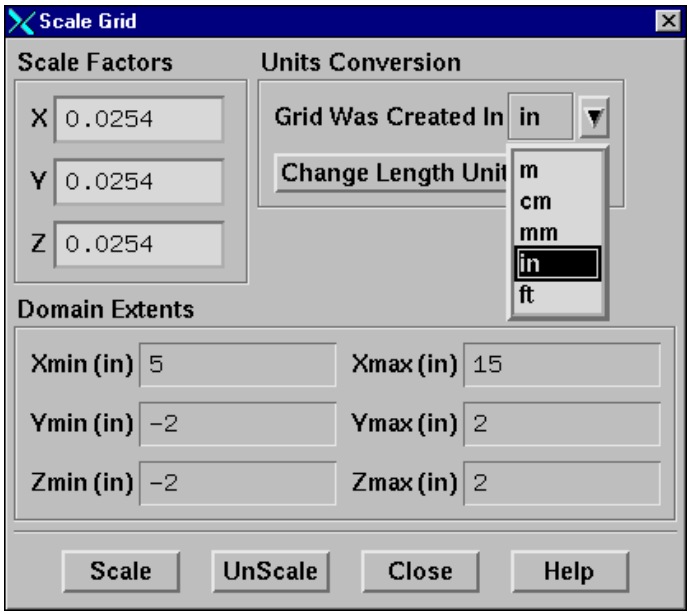


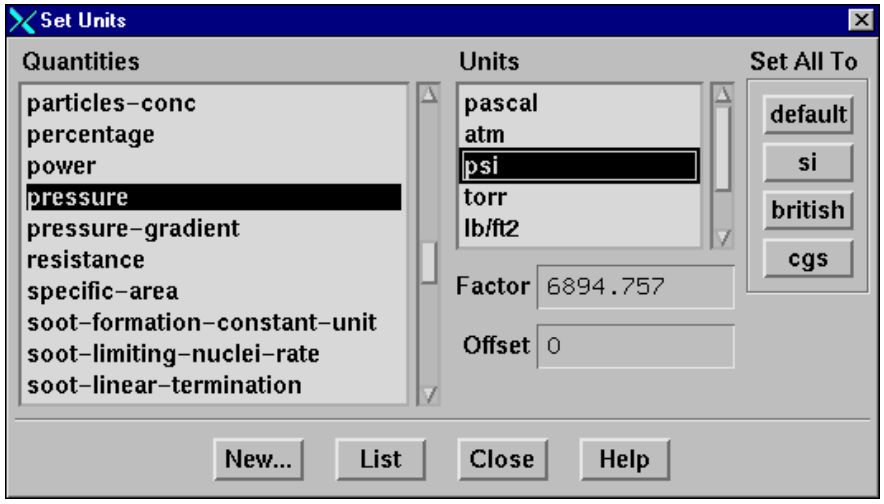
- Example: Face and cell zones associated with Pipe Flow through orifice plate.

Default-interior is zone of internal cell faces (not used).

Scaling Mesh and Units

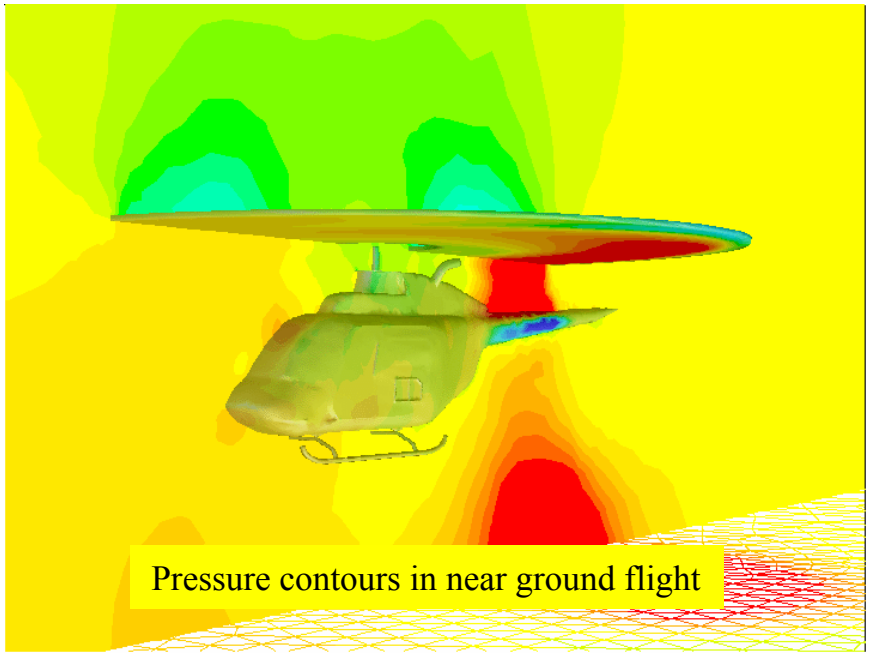
- All physical dimensions initially assumed to be in *meters*.
  - Scale grid accordingly.
- Other quantities can also be scaled independent of other units used.
  - Fluent defaults to SI units.

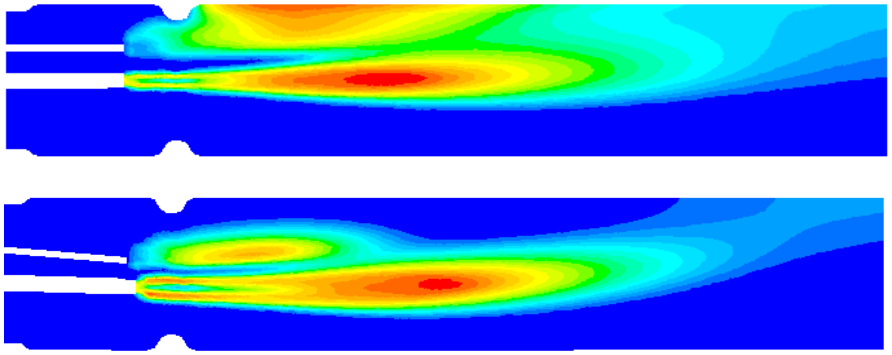




Models in Fluent 6 (1)

- Fluid Flow and Heat Transfer
  - Momentum, continuity, energy equations
  - Radiation models
- Turbulence
  - RANS-based models including Spalart-Allmaras k-ε, k-ω, and RSM
  - LES and DES
- Species Transport
- Volumetric Reactions
  - Arrhenius finite-rate chemistry
  - Turbulent fast chemistry
    - Eddy Dissipation, non-Premixed, premixed, partially premixed
  - Turbulent finite-rate chemistry
    - EDC, laminar flamelet, composition PDF transport
- Surface Reactions

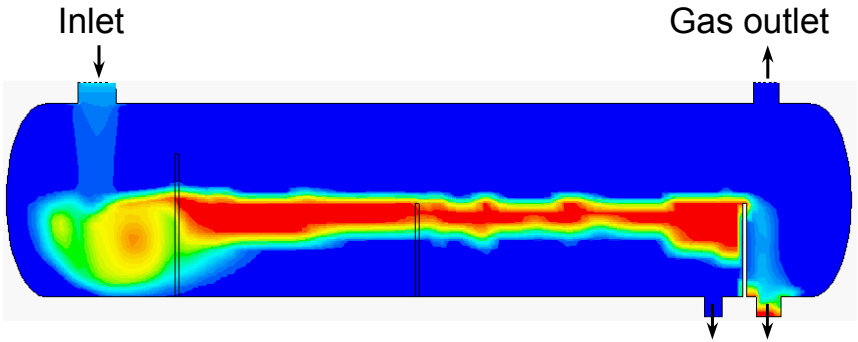




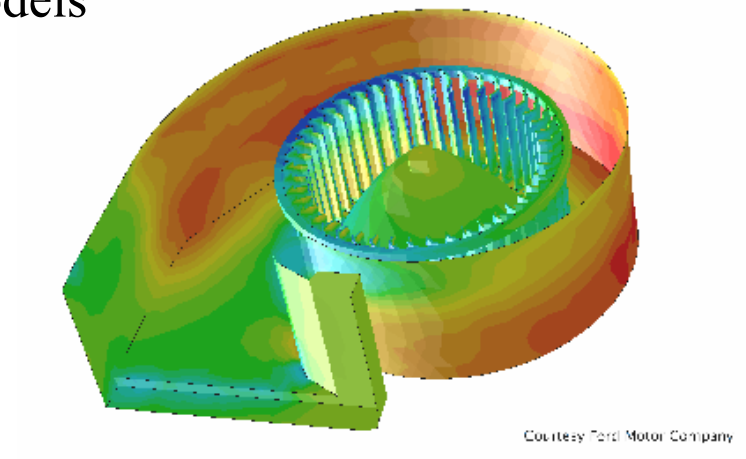
Temperature contours for kiln burner retrofitting.

Models in Fluent 6 (2)

- Multiphase Flows
  - Discrete Phase Model (DPM)
  - Volume of Fluid (VOF) model for immiscible fluids
  - Mixture Model
  - Eulerian-Eulerian and Eulerian-Granular Models
  - Liquid/Solid and Cavitation Phase Change Models
- Flows involving Moving Parts
  - Moving zones
    - Single/Multiple Rotating Reference Frames
    - Mixing Plane Model
    - Sliding Mesh Model
  - Moving and Deforming (dynamic) Mesh
- User-Defined Scalar Transport Equations



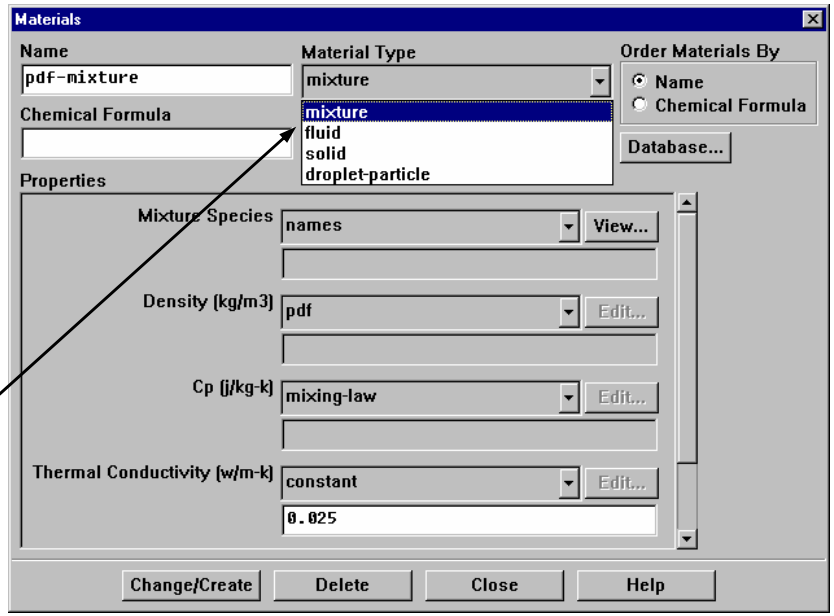
Contours of oil volume fraction in three phase separator.



Pressure contours for squirrel cage blower.

Material Types and Property Definition

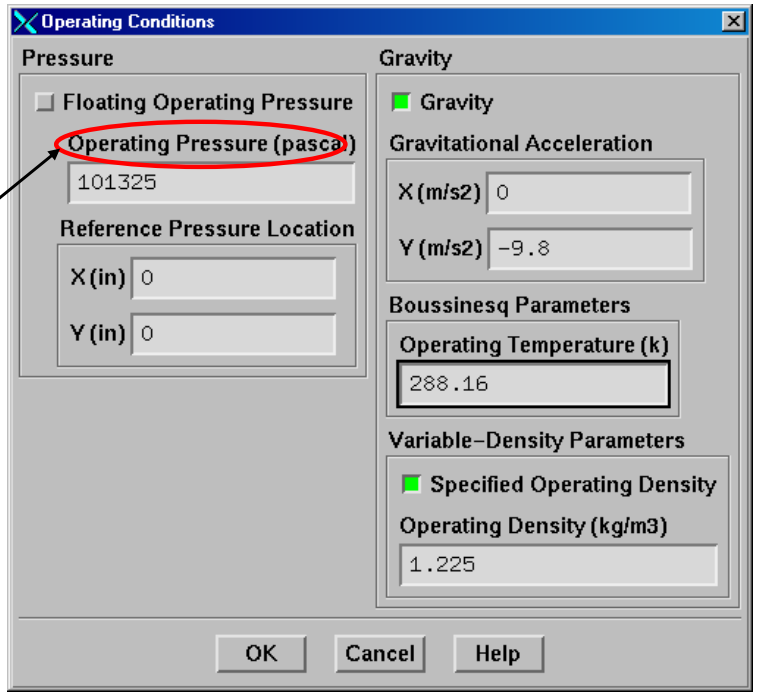
- Physical models may require inclusion of additional materials and dictates which properties need to be defined.
- Material *properties* defined in Materials Panel:
  - Single-Phase, Single Species Flows
    - Define fluid/solid properties
    - Real gas model (NIST's REFPROP 7.0 or user-defined C-function library)
  - 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.
  - Multiphase Flows (Single Species)
    - Define properties for all fluids and solids



Fluid Density

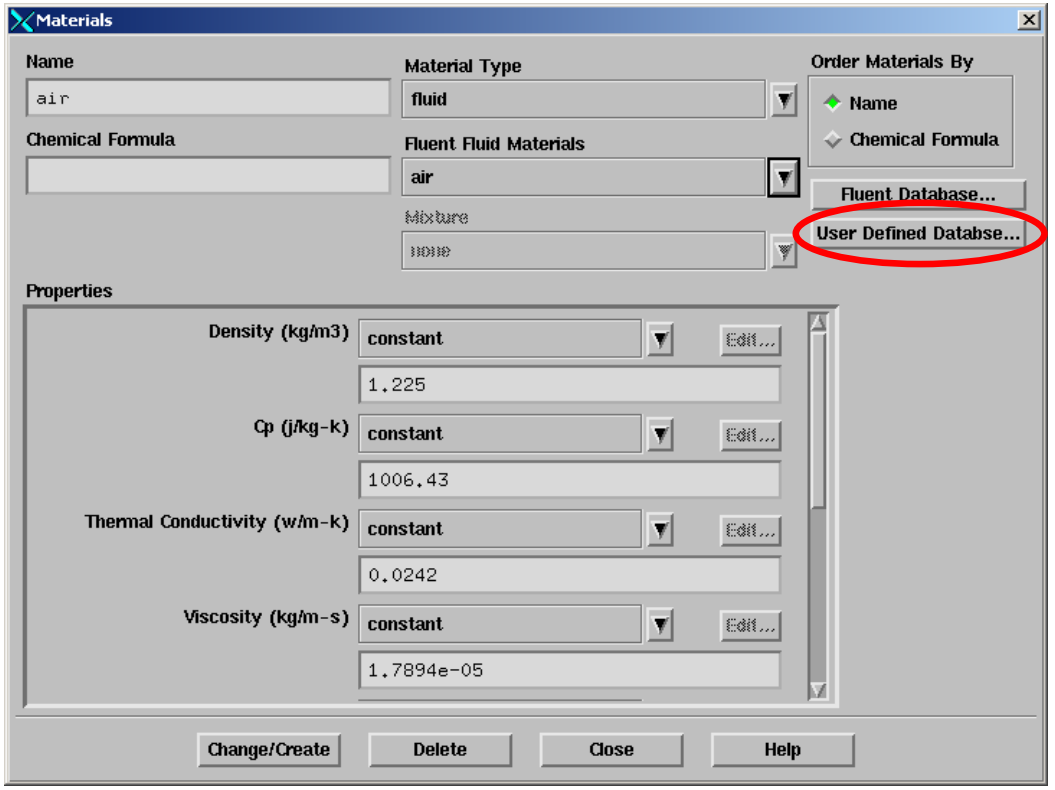
Define → Materials...

- For ρ = constant, incompressible flow:
  - Select constant for density
- For variable density, incompressible flows:
  - $\rho = p_{operating}/RT$
    - Use incompressible-ideal-gas for density
    - Set $p_{operating}$ close to the mean pressure in the problem
- For compressible flow:
  - $\rho = p_{absolute}/RT$
    - use ideal-gas for density
    - For low-Mach-number flows, set $p_{operating}$ close to mean pressure of the problem to avoid round-off errors
    - Use Floating Operating Pressure for unsteady flows with large, gradual changes in absolute pressure (segregated solver only).
- Density can also be defined as a function of temperature:
  - polynomial or piecewise-polynomial
  - Boussinesq model to be discussed in heat transfer lecture
- For compressible liquids, density variation is specified by a user-defined density function



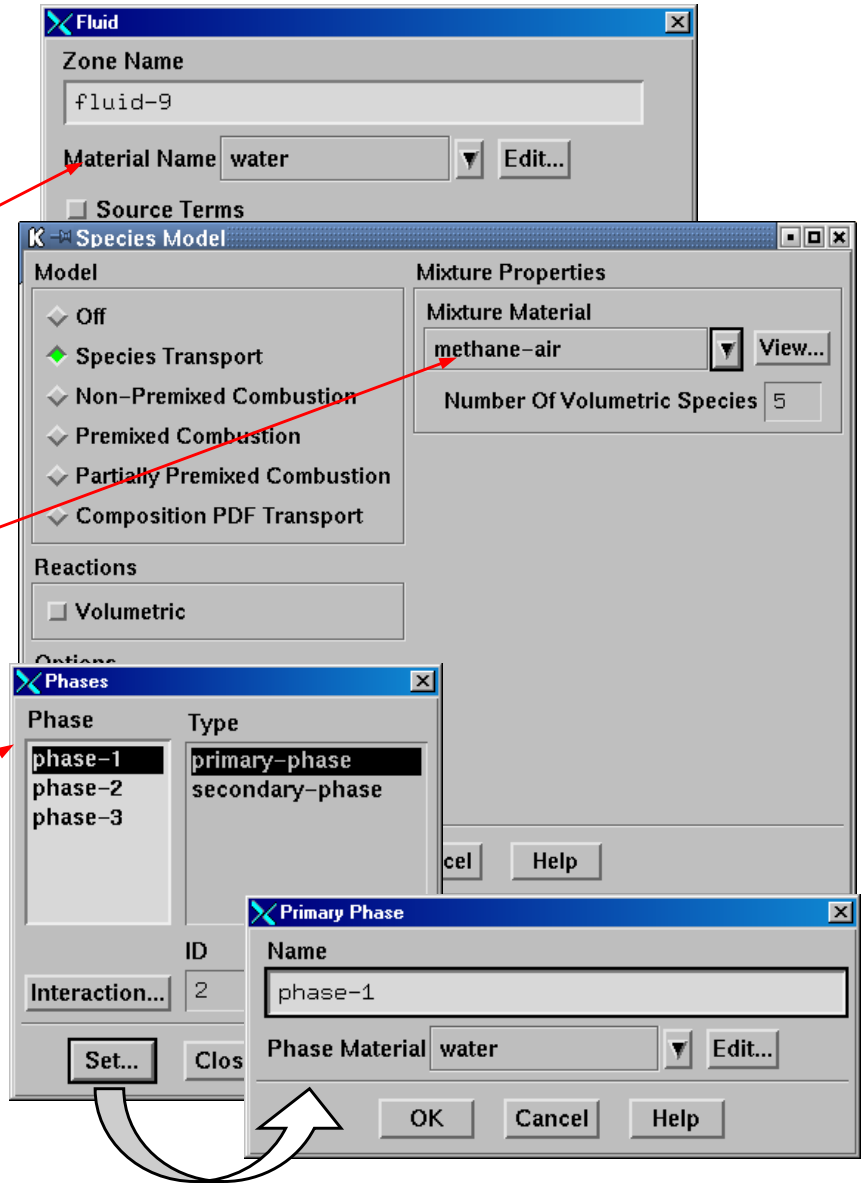
Custom Material Database

- Custom material database:
  - Create a new custom database of material properties and reaction mechanisms from materials in an existing case file for re-use in different cases
  - Custom databases can be created, accessed and modified from the standard materials panel in FLUENT 6.2

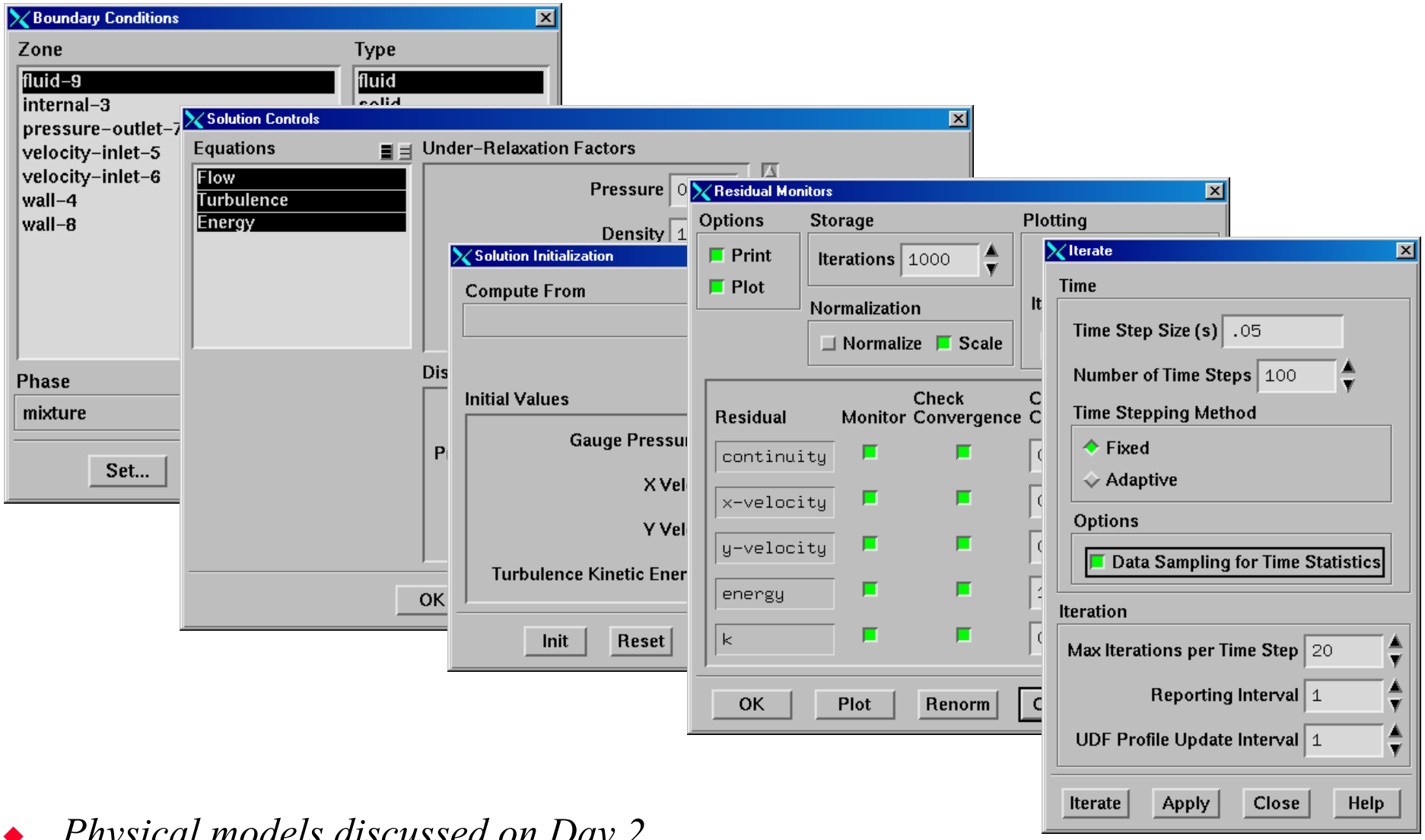


Material Assignment

- Materials are *assigned* to cell zone where assignment method depends upon models selected:
  - Single-Phase, Single Species Flows
    - Assign material to fluid zone(s) in Fluid Panel (within Define→B.C.)
  - Multiple Species (Single Phase) Flows
    - Assign mixture material to fluid zones in Define→Species Panel or in Pre-PDF.
    - All fluid zones consist of the 'mixture'
  - Multiphase (Single Species) Flows
    - Primary and secondary phases selected in Define →Phases Panel.
    - All fluid zones consist of the 'mixture'



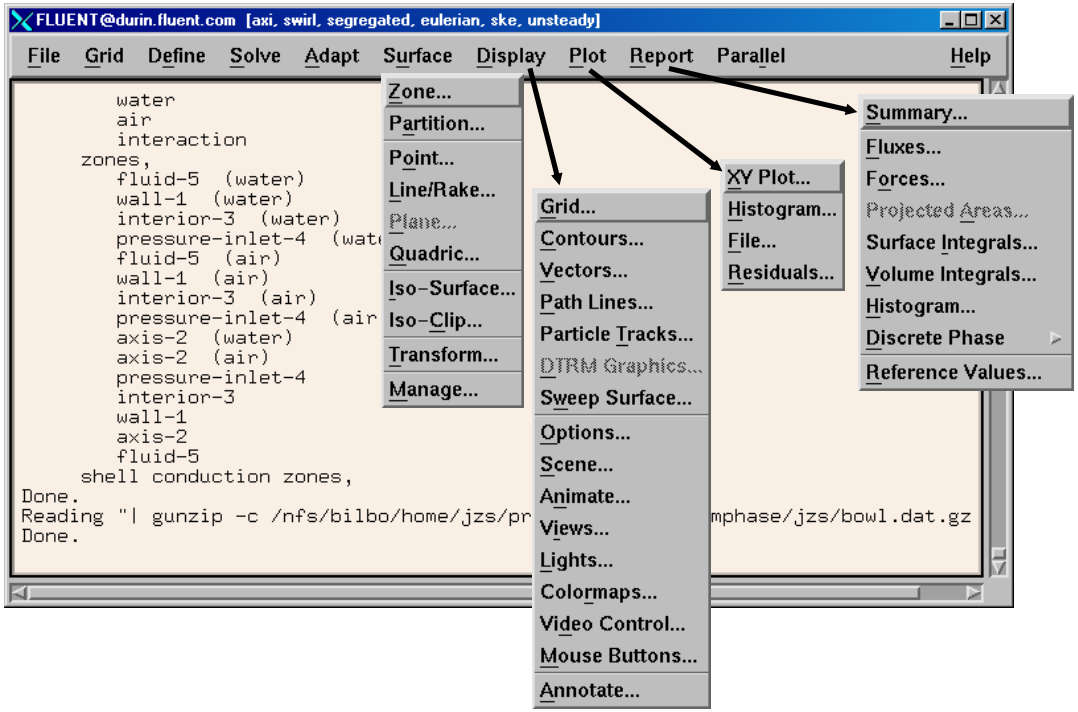
Solver Execution: Other Lectures...

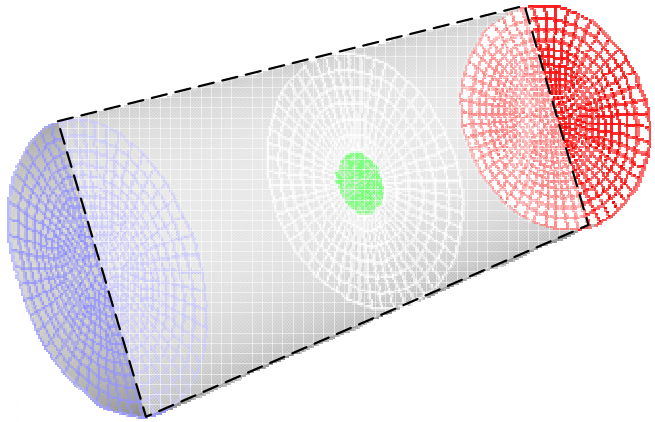


- *Physical models discussed on Day 2.*

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 by users

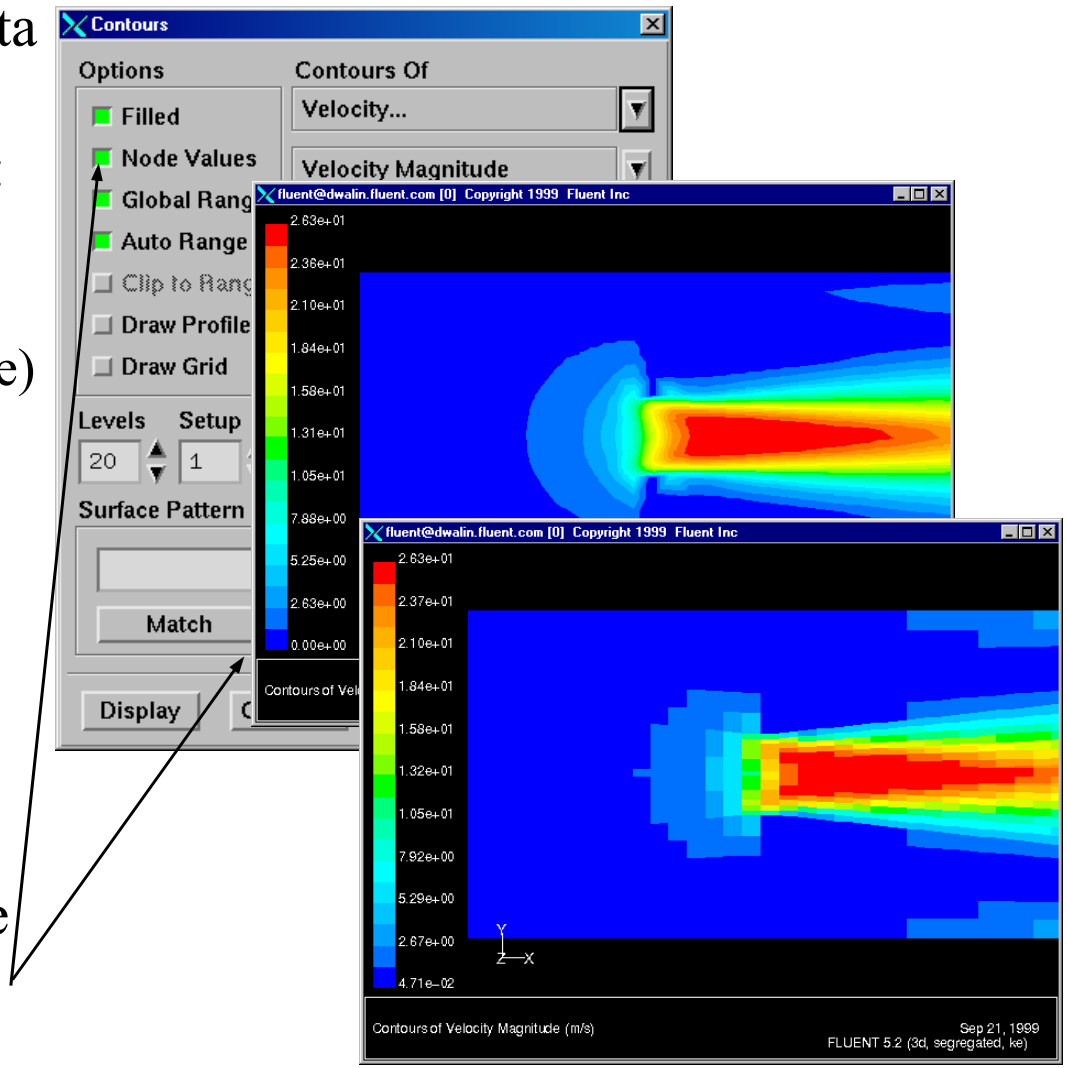




- Example: an Iso-Surface of a constant grid coordinate can be created for viewing data within a plane.

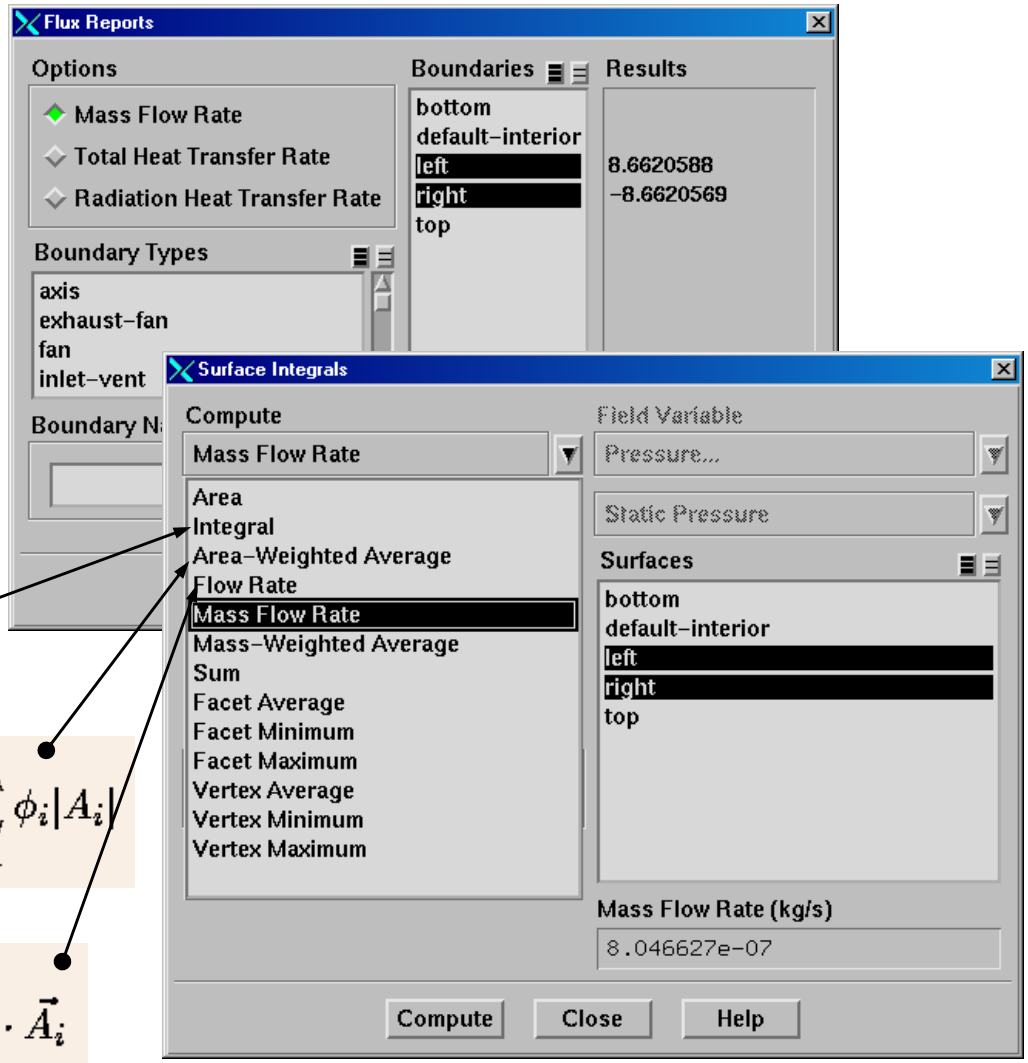
Post-Processing: Node Values

- 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



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

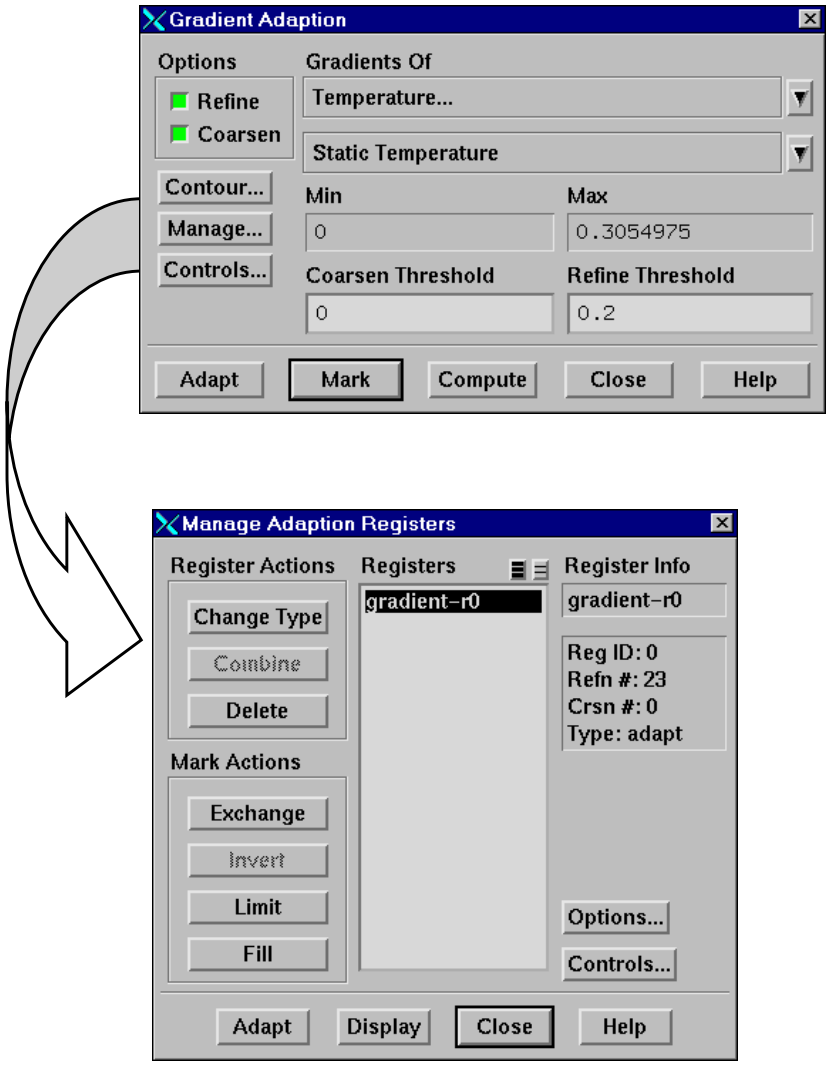
$$\frac{1}{A}\int \phi dA = \frac{1}{A}\sum_{i=1}^{n} \phi_i |A_i|$$

Examples:

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

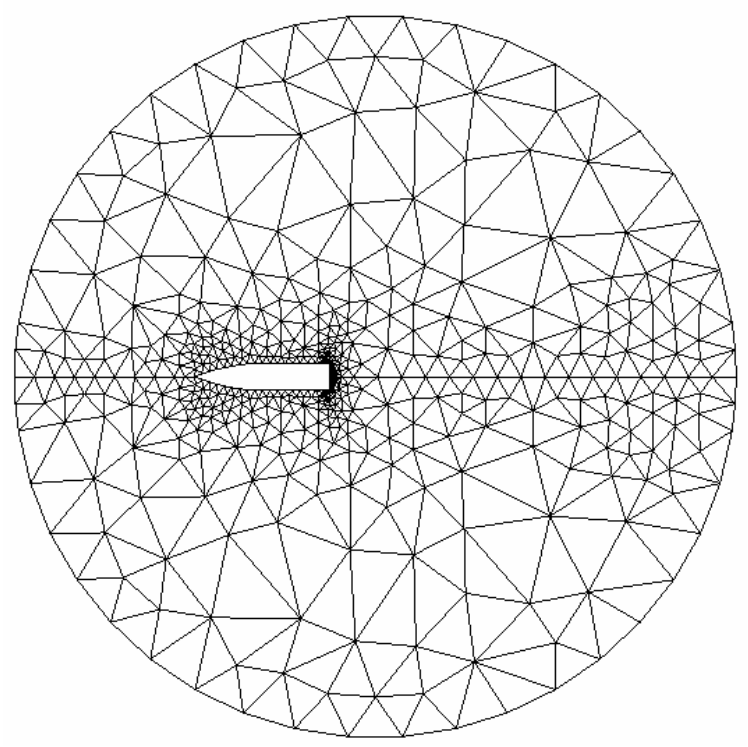
Solver Enhancements: Grid Adaption

- 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...

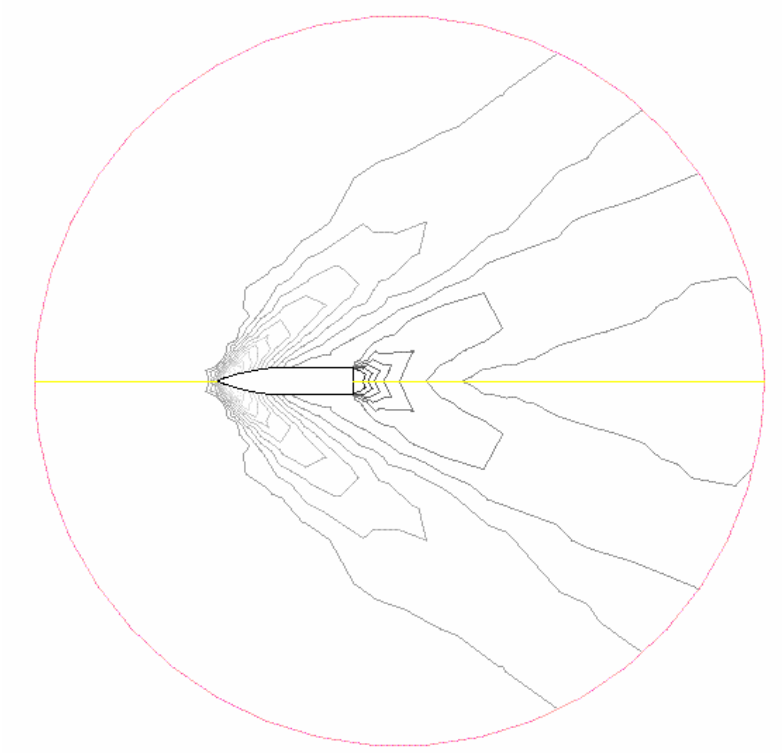


Adaption Example: 2D Planar Shell

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

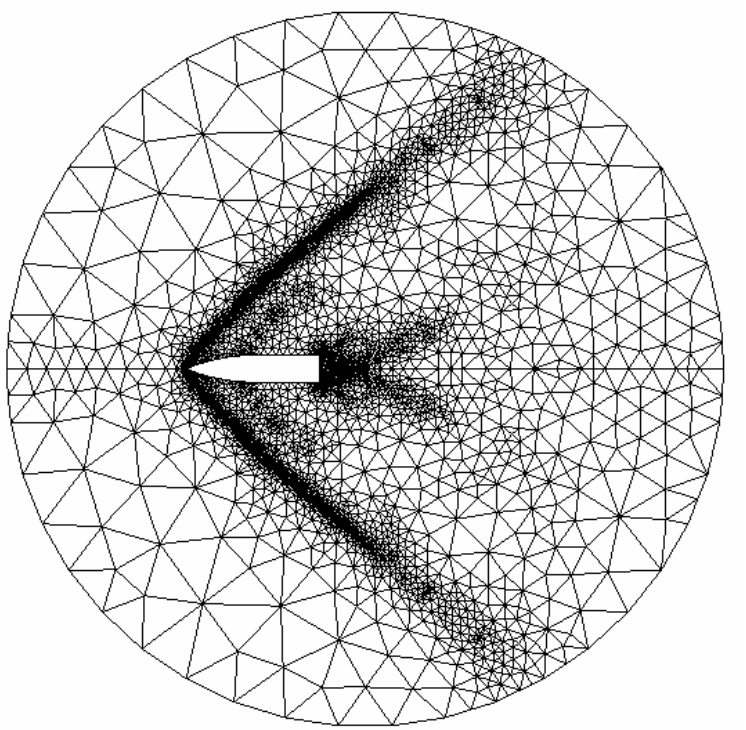


2D planar shell - initial grid

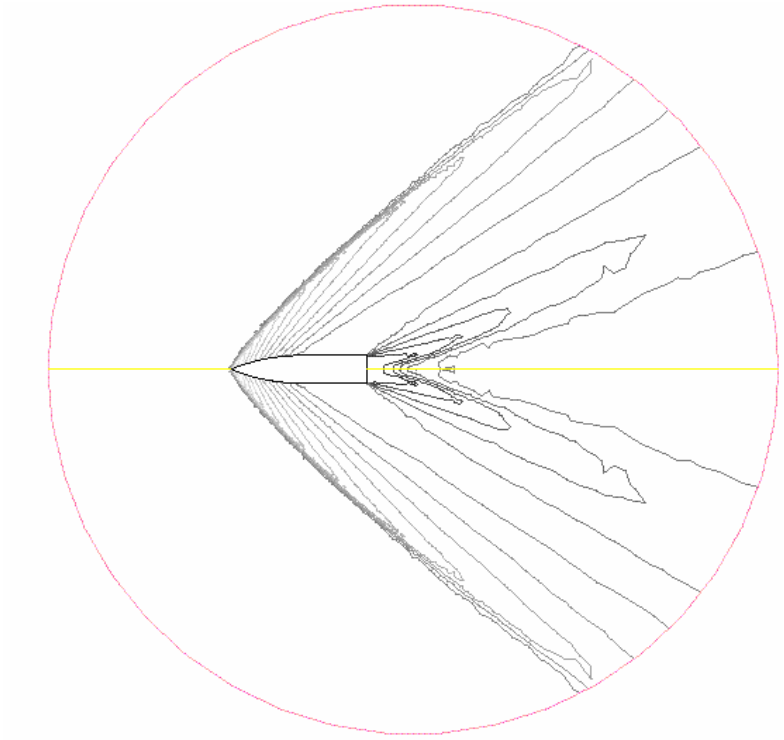


2D planar shell - contours of pressure initial grid

Adaption Example: Final Grid and Solution



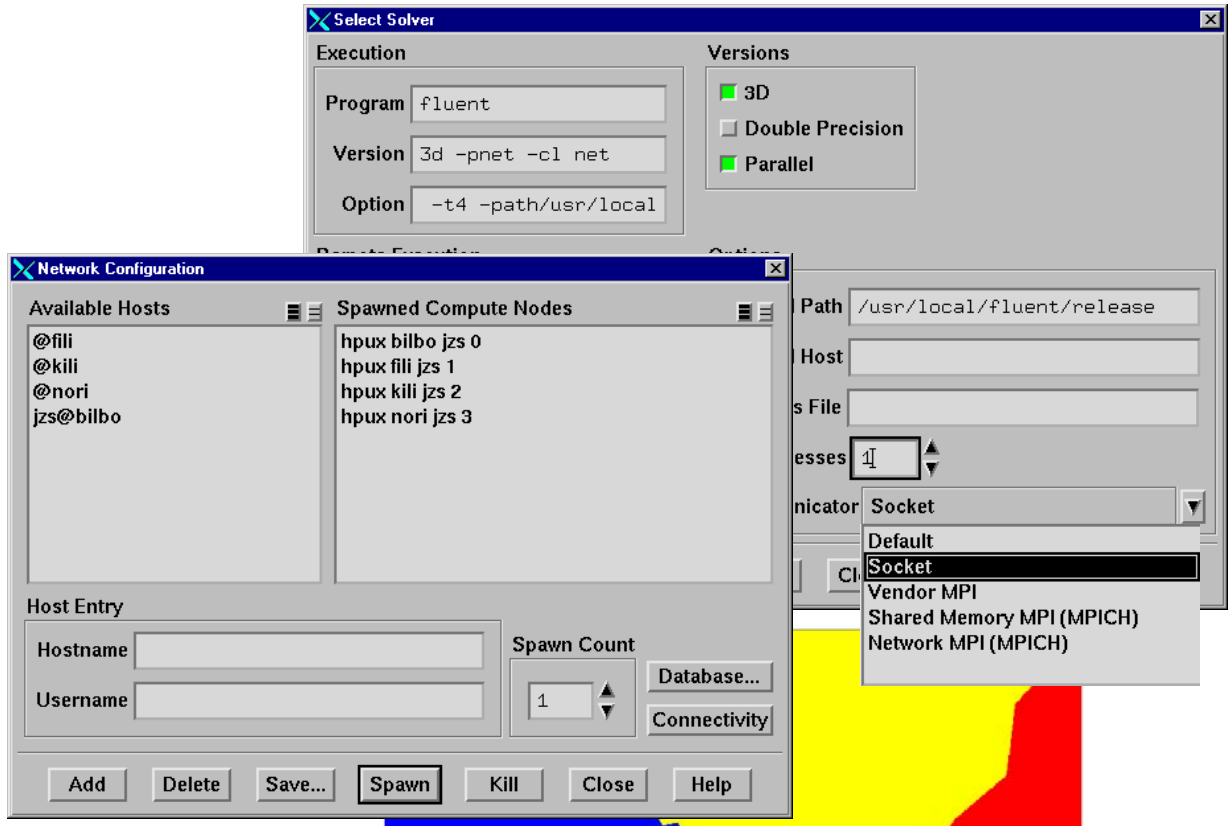
2D planar shell - final grid



2D planar shell - contours of pressure final grid

Solver Enhancements: Parallel Solver

- With 2 or more processes, Fluent can be run on multiple processors.
- Can run on a dedicated, multiprocessor machine, or a network of machines.
- Mesh can be partitioned manually or automatically.
- Some models not yet ported to parallel solver.
  - See release notes.



Partitioned grid for multi-element airfoil.

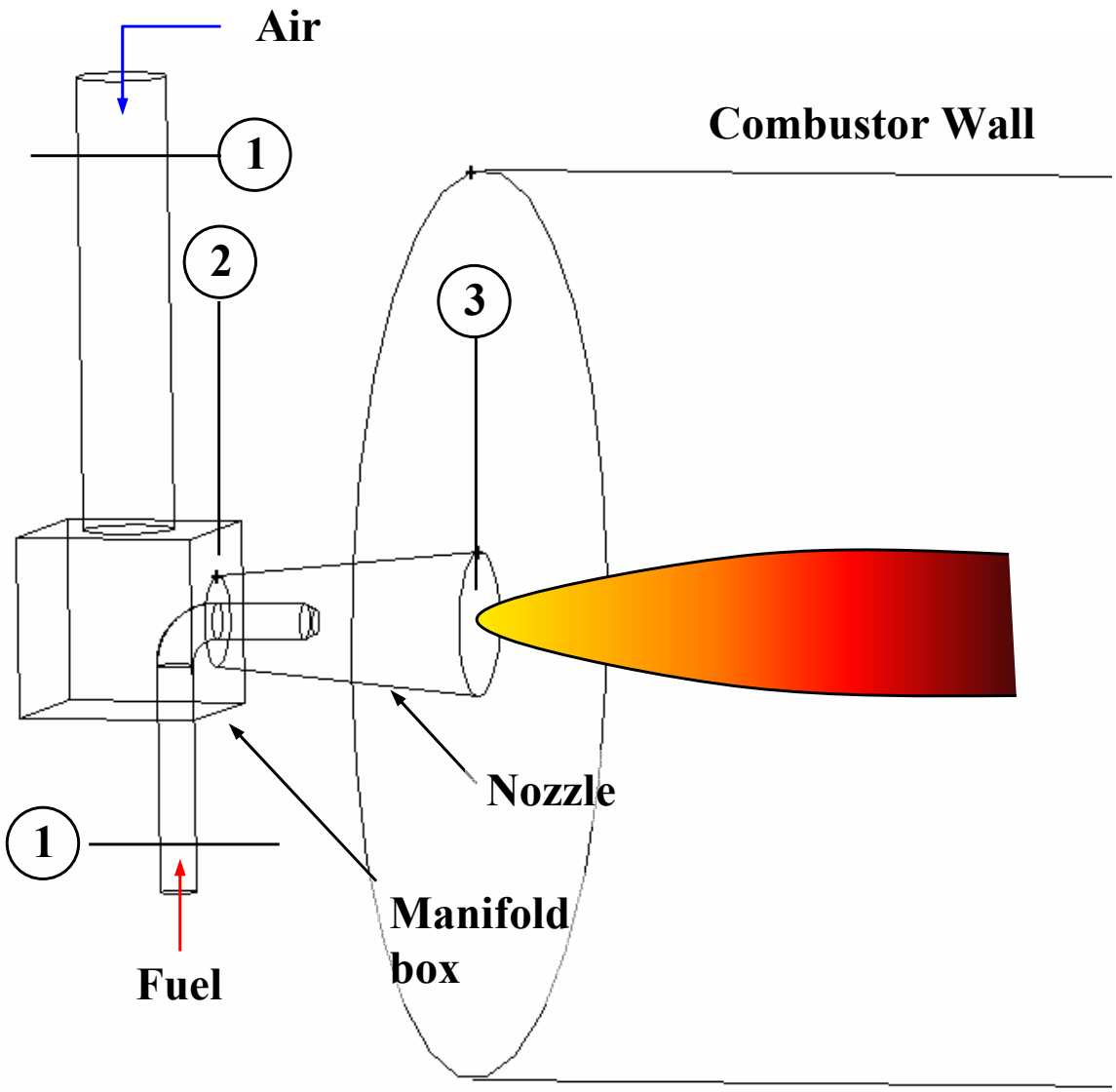
Boundary Conditions

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

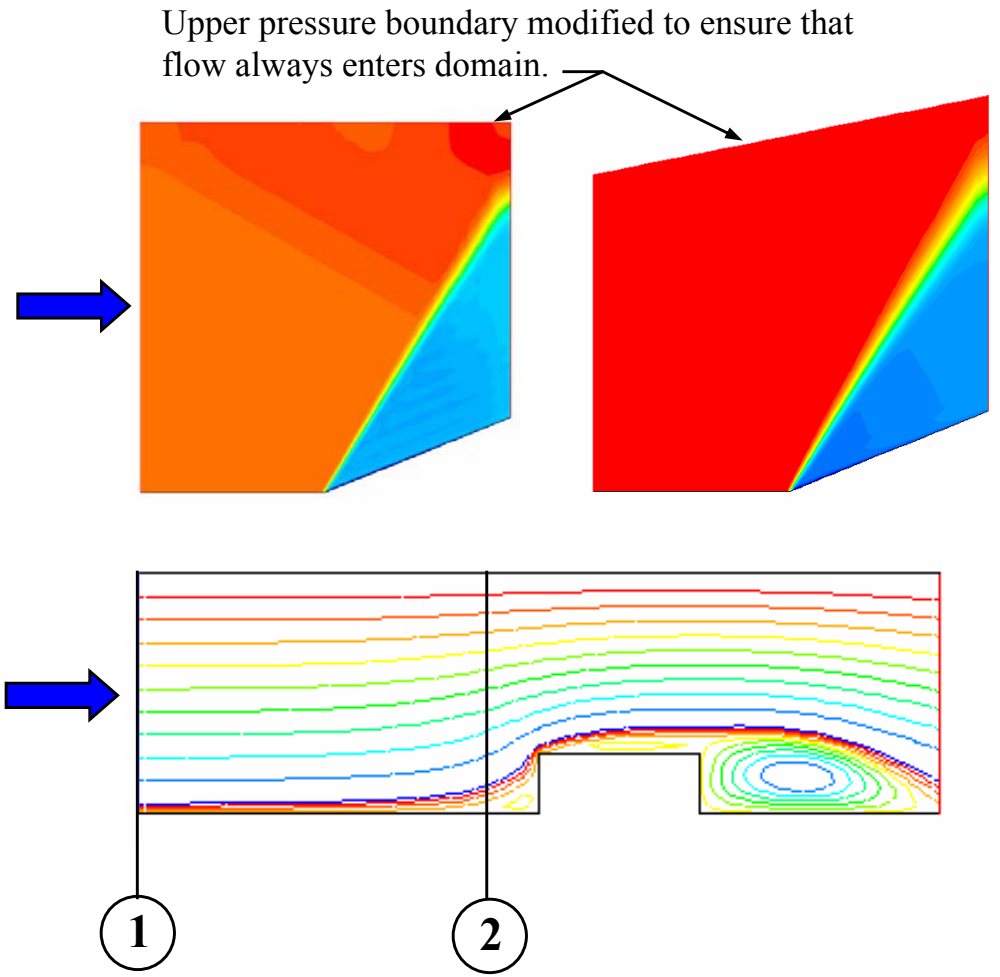
Locating Boundaries: Example

- 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



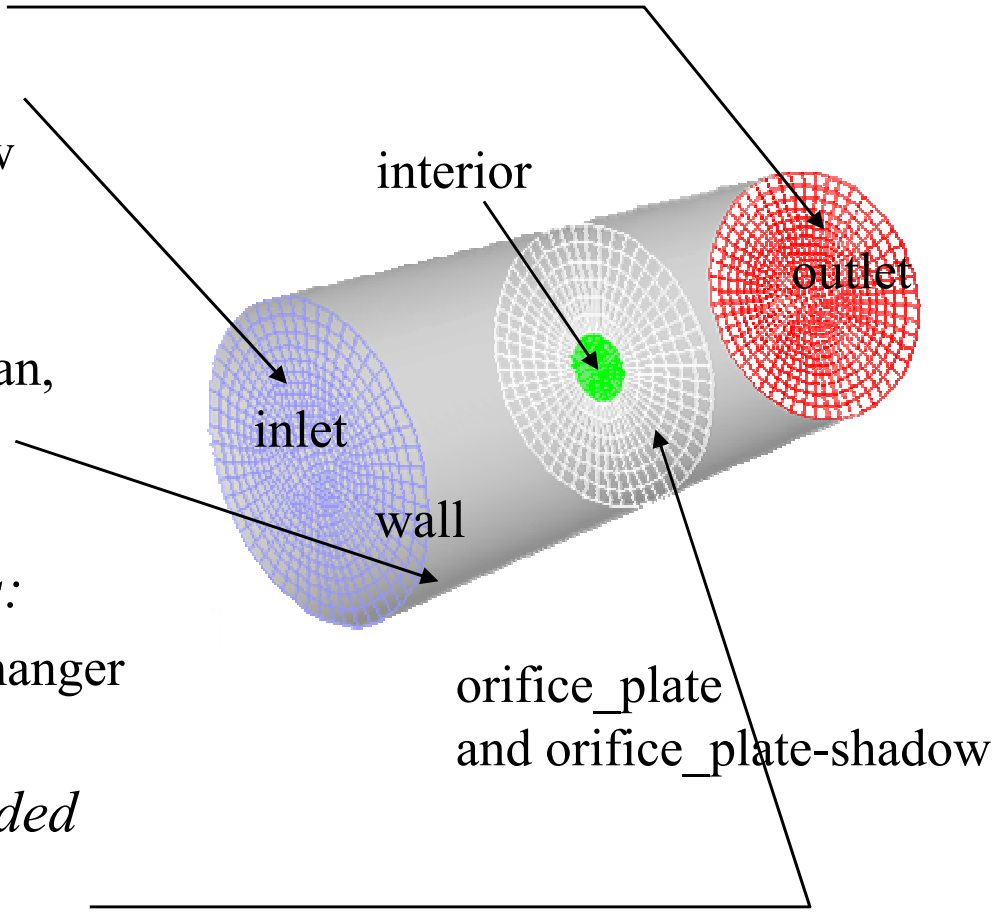
General Guidelines

- 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 the boundary.
    - Otherwise it would introduce error early in calculation.



Available Boundary Condition Types

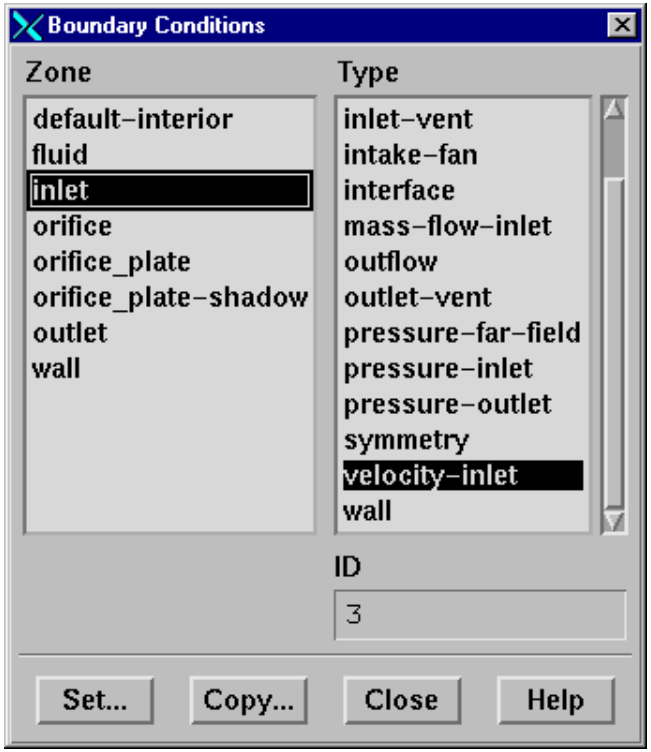
- Boundary Condition Types of *External Faces:*
  - **General**: Pressure inlet, pressure outlet
  - **Incompressible**: Velocity inlet, outflow
  - **Compressible flows**: Mass flow inlet, pressure far-field, mass flow outlet
  - **Special**: Inlet vent, outlet vent, intake fan, exhaust fan
  - **Other**: Wall, symmetry, axis, periodic
- Boundary Condition Types of *cell zones:*
  - Fluid, solid, porous media and heat exchanger models
- Boundary Condition Types of *double-sided internal faces*
  - Fan, interior, porous Jump, radiator, walls



Changing Boundary Condition Types

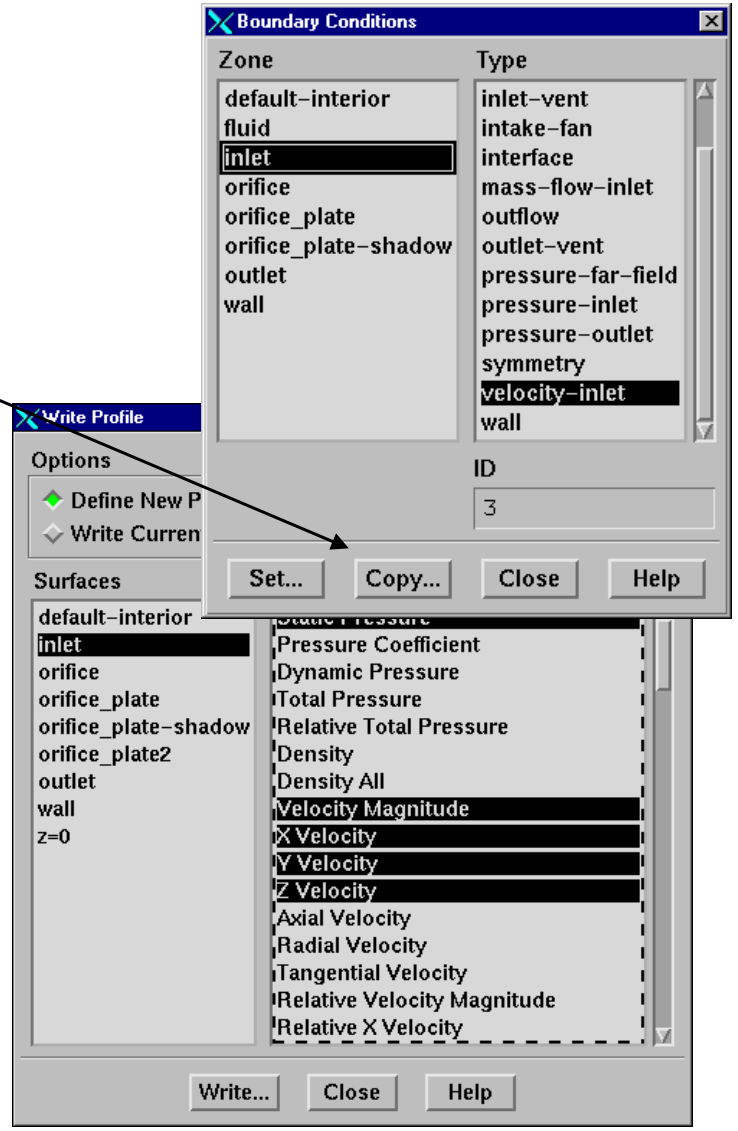
- Zones and zone types are initially defined in pre-processor.
- To change zone type for a particular zone:

  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.



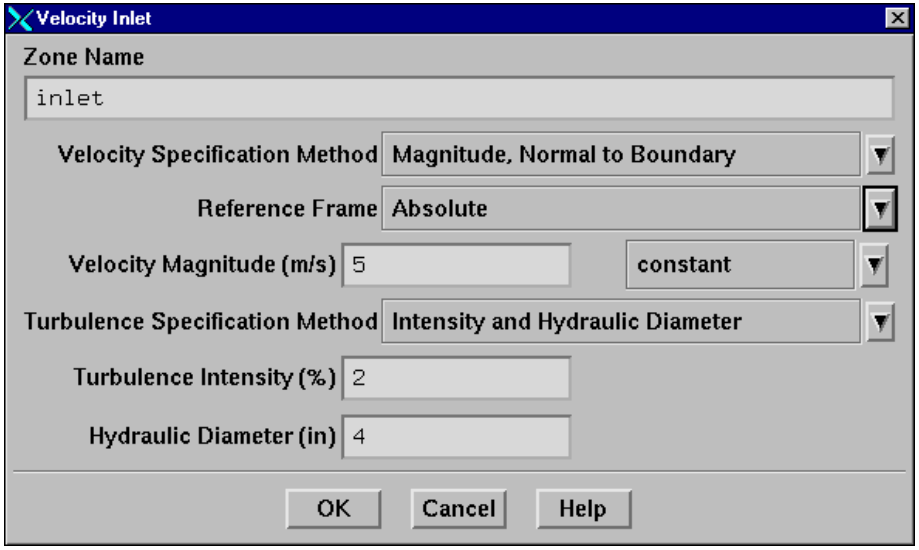
Setting Boundary Condition Data

- 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.



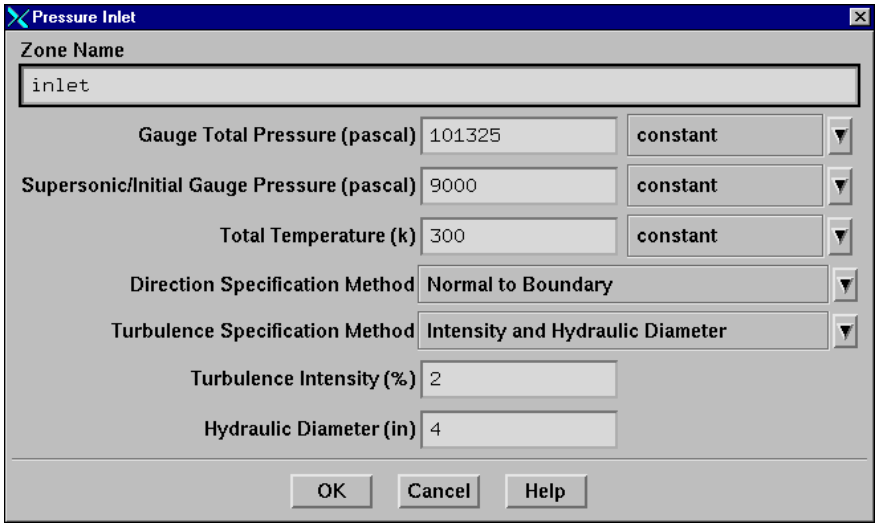
Velocity Inlet

- 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



From 1-D Compressible flow relationship:

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

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

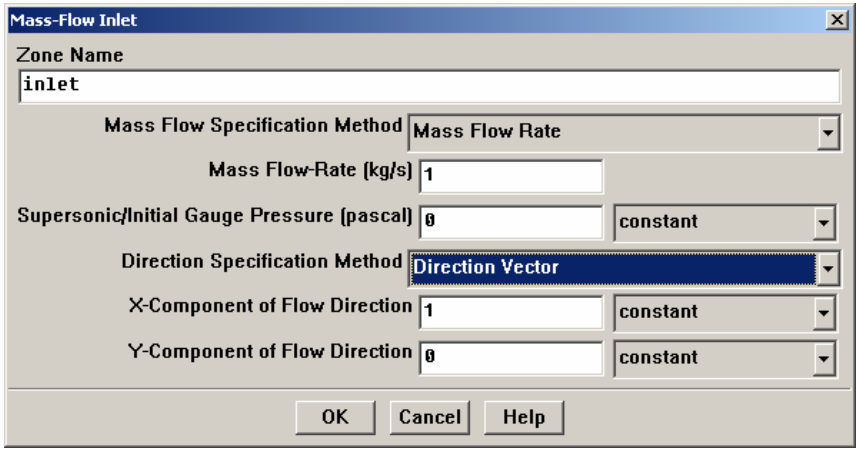
Incompressible flows: $p_{total} = p_{static} + \frac{1}{2}\rho v^2$

Pressure Inlet (2)

- Note: *Gauge* pressure inputs are required.
  - $p_{absolute} = p_{gauge} + p_{operating}$
  - Operating pressure level sometimes may affect solution accuracy (when pressure fluctuations are relatively small).
  - 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.

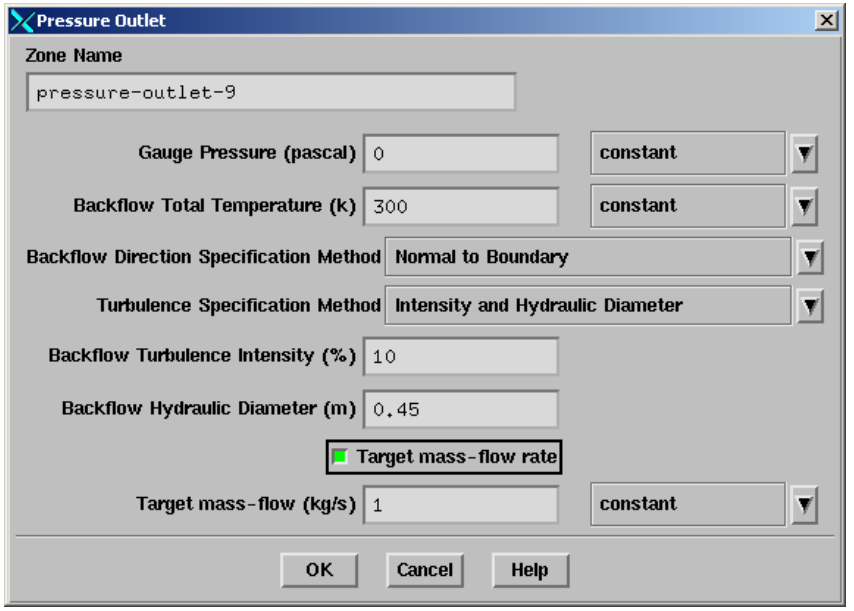
Mass Flow Inlet

- Specify:
  - (a) Mass Flow Rate or (b) Mass Flux
    - (a) implies uniform mass flux
    - (b) can be defined by profiles/UDF
  - 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
- Intended for compressible; can be used for incompressible flows.
  - Total pressure adjusts to accommodate mass flow inputs.
  - More difficult to converge than with pressure inlet.



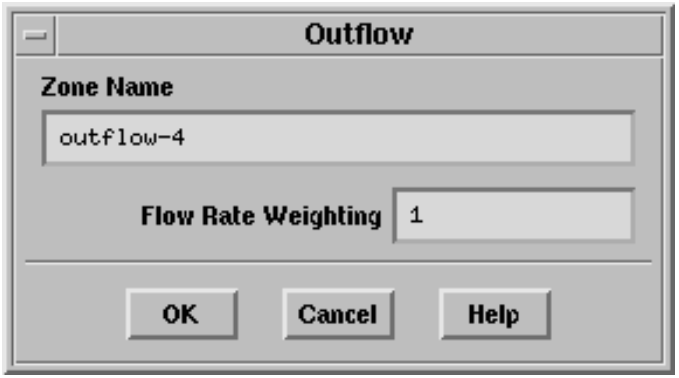
Pressure Outlet

- 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 can be normal to the boundary, set by direction vector or from neighboring cell .
  - Backflow boundary data must be set for all transport variables.
  - Convergence difficulties are reduced by providing realistic backflow quantities
- Suitable for compressible and incompressible flows
  - Specified pressure is ignored if flow is locally supersonic at the outlet
- Can be used as a "free" boundary in an external or unconfined flow
- For ideal gas (compressible) flow, non-reflecting outlet boundary conditions (NRBC) are available



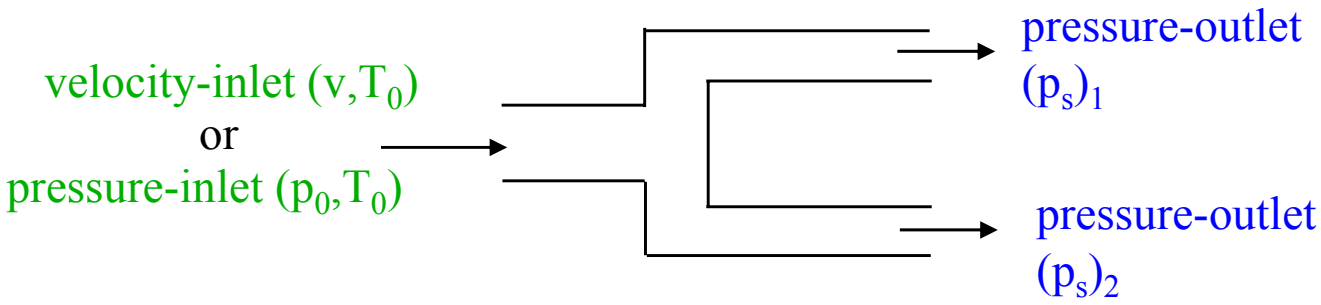
Outflow

- 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 the exit flow is "fully-developed."
- Intended for incompressible flows:
  - Cannot be used with a Pressure-Inlet boundary: must use Velocity-Inlet BC at the 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.

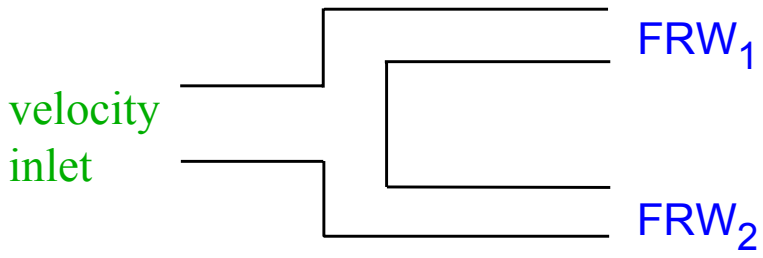


Modeling Multiple Exits

- 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.

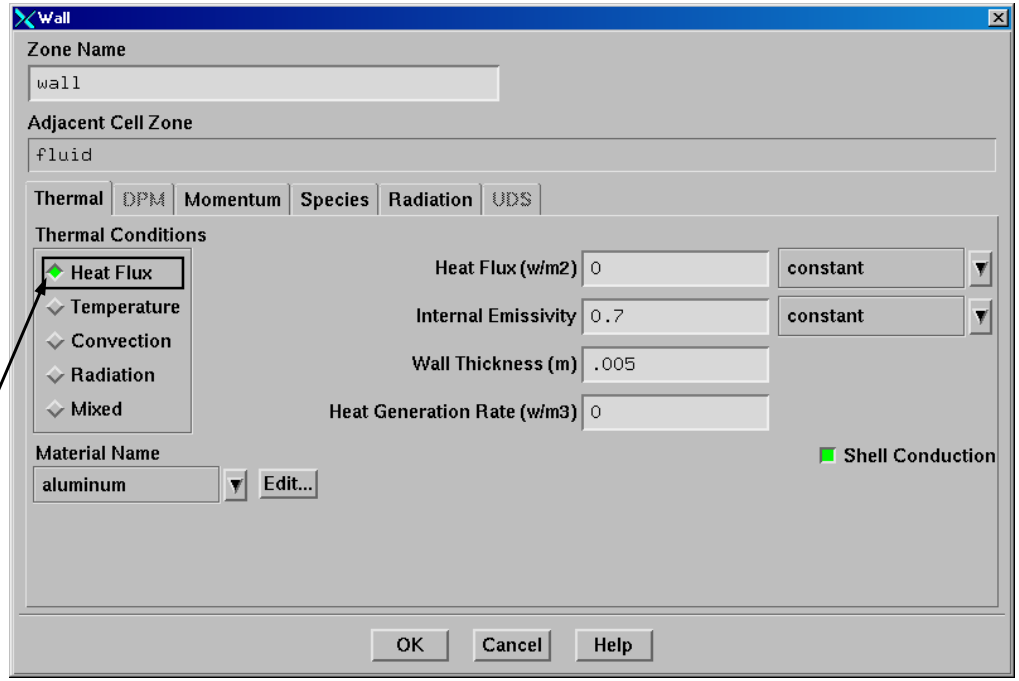


Other Inlet/Outlet Boundary Conditions

- Pressure Far Field
  - Available when density is calculated from the ideal gas law
  - Used to model free-stream compressible flow at infinity, with free-stream Mach number and static conditions specified.
- Target Mass Flow Rate (not available for the multiphase models) for Pressure Outlet
  - Specify mass flow rate for an outlet (constant or via UDF hook)
  - Options to choose iteration method in TUI
- Exhaust Fan/Outlet Vent
  - Model external exhaust fan/outlet vent with specified pressure jump/loss coefficient and ambient (discharge) pressure and temperature.
- Inlet Vent/Intake Fan
  - Model inlet vent/external intake fan with specified loss coefficient/ pressure jump, flow direction, and ambient (inlet) pressure and temperature
- Inlet boundary conditions for large-eddy/detached-eddy simulations are covered in the Turbulence Modeling lecture

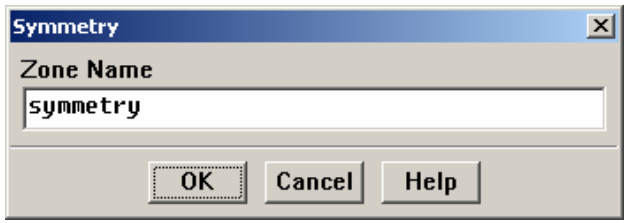
Wall Boundaries

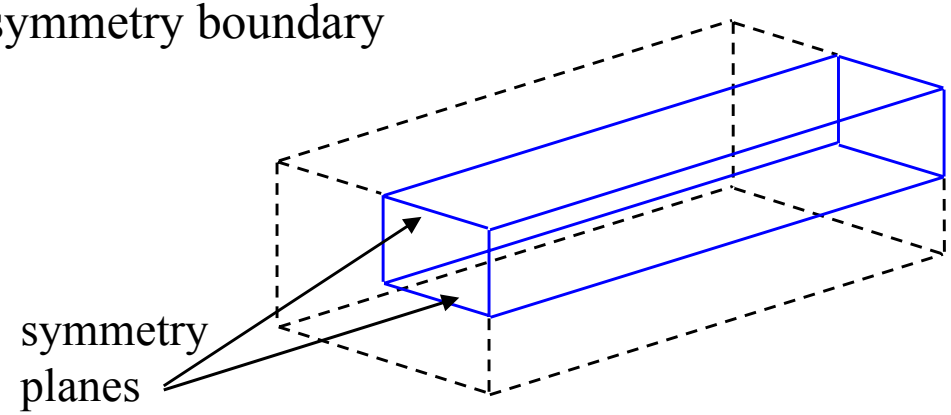
- Used to bound fluid and solid regions.
- In viscous flows, no-slip condition enforced at walls:
  - Tangential fluid velocity equal to wall velocity.
  - Normal velocity component = 0
  - Shear stress can also be specified.
- Thermal boundary conditions:
  - several types available
  - Wall material 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 boundaries.

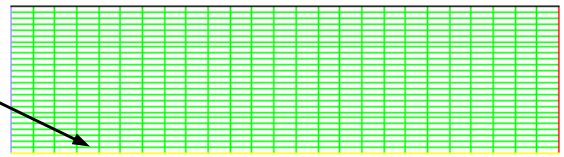


Symmetry and Axis Boundaries

- Symmetry Boundary
  - Used to reduce computational effort in problem.
  - No inputs required.
  - Flow field *and* geometry must be symmetric:
    - Zero normal velocity at symmetry plane
    - Zero normal gradients of all variables at symmetry plane
    - Must take care to correctly define symmetry boundary locations.
  - Can be used to model slip walls in viscous flow
- Axis Boundary
  - Used at centerline for axisymmetric problems.
  - No user inputs required.

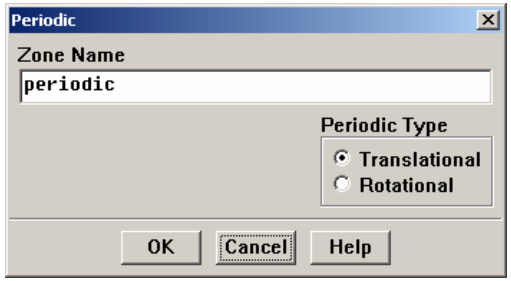


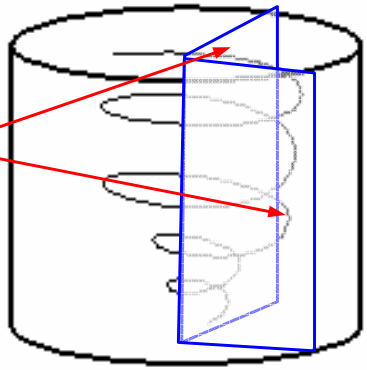




Periodic Boundaries

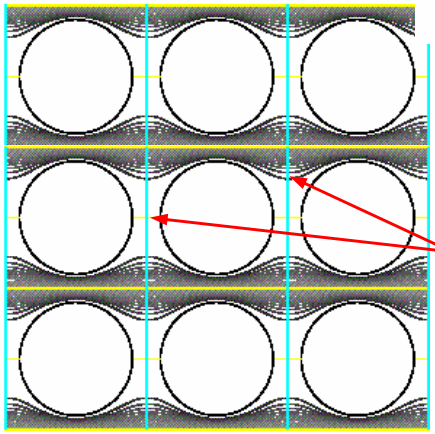
- Used to reduce computational effort in problem.
- Flow field and geometry *must* be either translationally or rotationally periodic.
- For rotationally periodic boundaries:
  - $\Delta p = 0$ across periodic planes.
  - Axis of rotation must be defined in fluid zone.
- For translationally periodic boundaries:
  - Δp can be finite across periodic planes.
    - Models fully developed conditions.
    - Specify either mean Δp per period or net mass flow rate.
    - Periodic boundaries defined in Gambit are translational.





Rotationally periodic planes

flow →

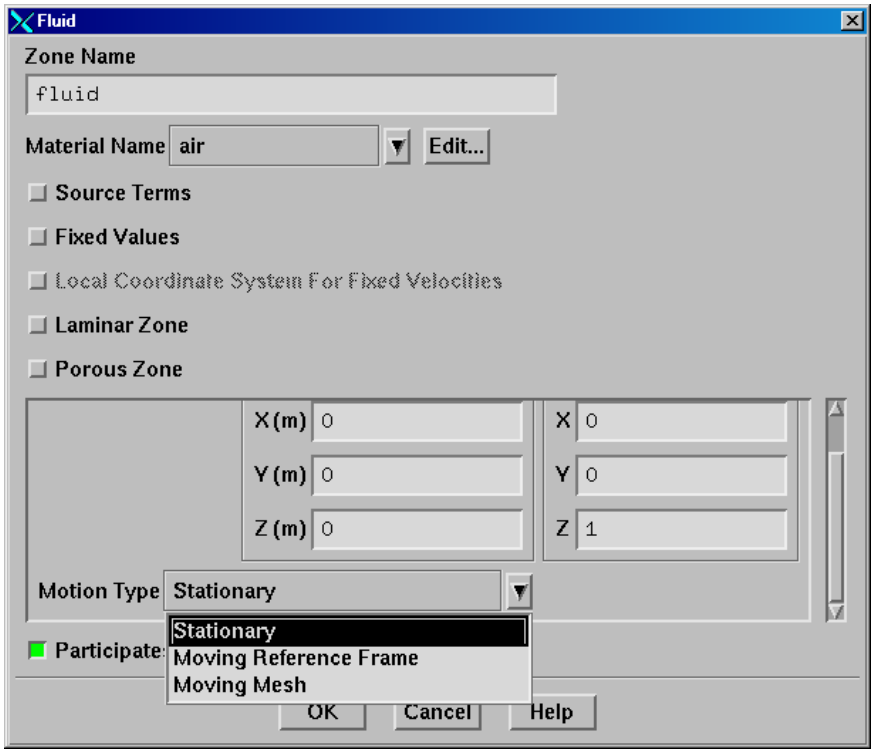


Translationally periodic planes

2D tube heat exchanger

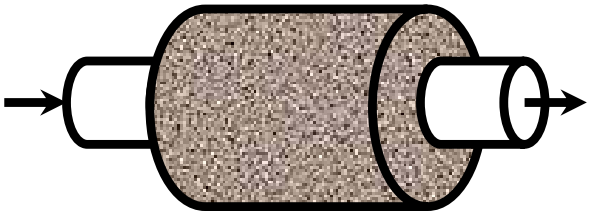
Cell Zones: Fluid

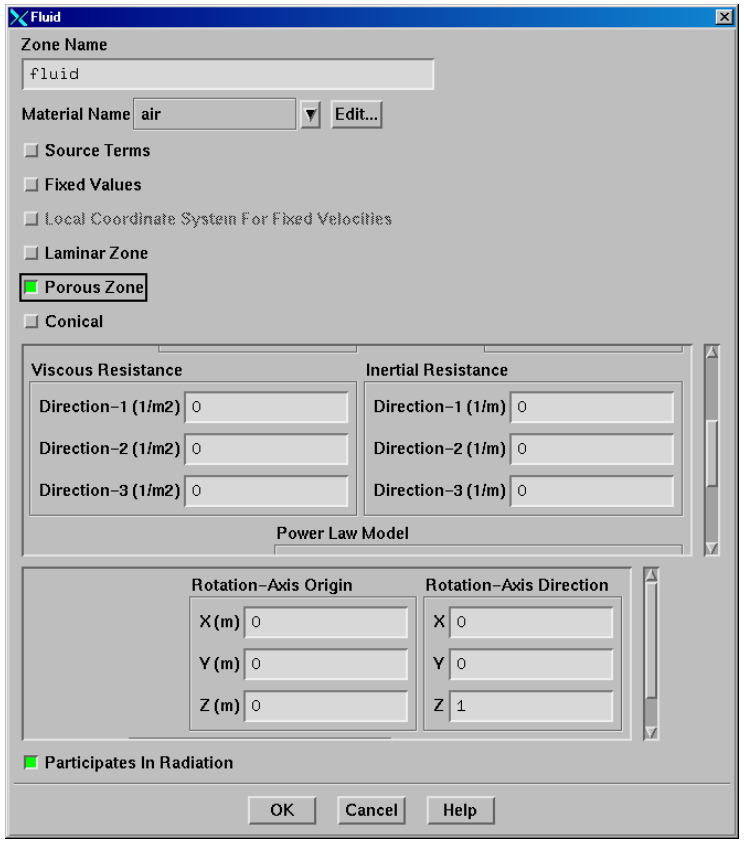
- A fluid cell zone is a group of cells for which all active equations are solved
- Fluid material selection is required
  - For multi-species or multiphase flows, the material is not shown, but fluid zone consists of the mixture or the phases
- Optional inputs allow setting of source terms:
  - mass, 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



Porous Media Conditions

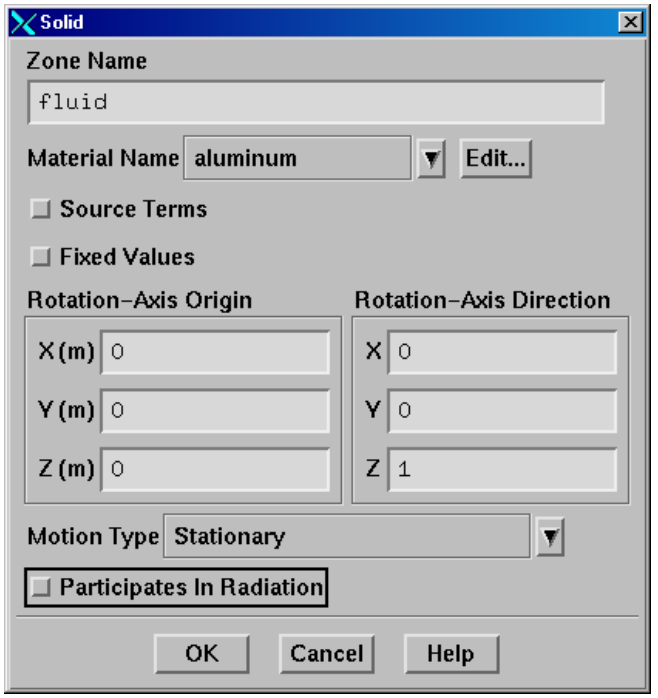
- Porous zone modeled as special type of fluid zone.
  - Enable Porous Zone option in Fluid panel.
  - Pressure loss in flow determined via user inputs of resistance coefficients to lumped parameter model
- Used to model flow through porous media and other "distributed" resistances, e.g.,
  - Packed beds
  - Filter papers
  - Perforated plates
  - Flow distributors
  - Tube banks





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 the material name defined in the materials (solid) panel
- 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 the cell faces only:
  - Thickness of these internal faces is zero
  - These internal faces provide means of introducing step changes in flow properties.
- Used to implement various physical models including:
  - Fans
  - Radiators
  - Porous-jump models
    - Preferable over porous media for its better convergence behavior.
  - Interior walls

Summary

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

Solver Settings

Outline

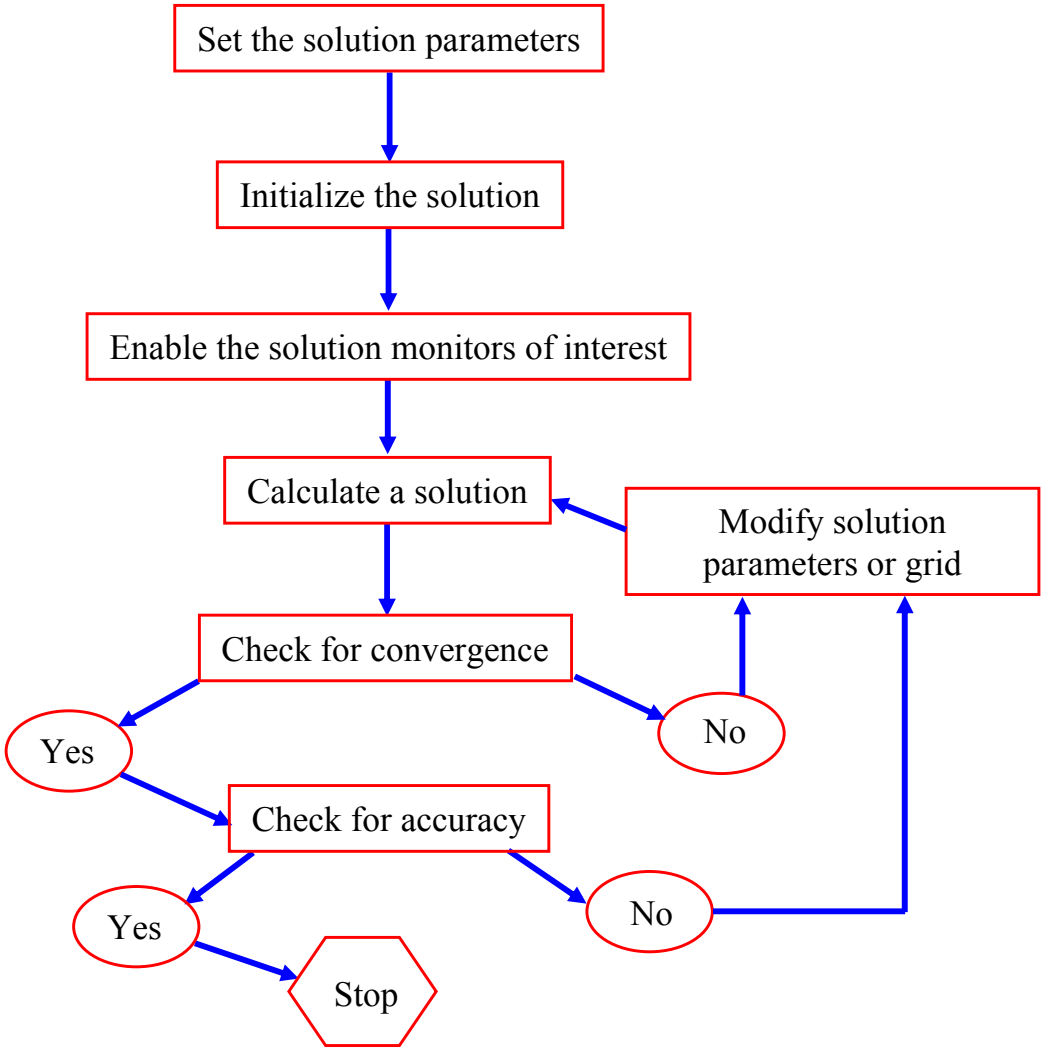
- Using the Solver
  - Setting Solver Parameters
  - Convergence
    - Definition
    - Monitoring
    - Stability
    - Accelerating Convergence
  - Accuracy
    - Grid Independence
    - Grid Adaption
  - Unsteady Flows Modeling
    - Unsteady-flow problem setup
    - Non-iterative Transient Advancement (NITA) schemes
    - Unsteady flow modeling options
  - Summary
  - Appendix

Outline

- **Using the Solver (solution procedure overview)**
  - **Setting Solver Parameters**
  - Convergence
    - Definition
    - Monitoring
    - Stability
    - Accelerating Convergence
  - Accuracy
    - Grid Independence
    - Grid Adaption
  - Unsteady Flows Modeling
    - Unsteady-flow problem setup
    - Non-iterative Transient Advancement (NITA) schemes
    - Unsteady flow modeling options
  - Summary
  - Appendix

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 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 solvers!
    - Improved pre-conditioning in Fluent v6.2 for the coupled-implicit solver enhances accuracy and robustness for low-Mach number flows
  - 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

Discretization (Interpolation Methods)

- Field variables (stored at cell centers) must be interpolated to the faces of the control volumes in the FVM:

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

- Interpolation schemes for the **convection** term:
  - First-Order Upwind Scheme
    - easiest to converge, only first-order accurate
  - Power Law Scheme
    - more accurate than first-order for flows when $Re_{cell} < 5$ (typ. low Re flows)
  - Second-Order Upwind Scheme
    - uses larger 'stencils' for 2nd order accuracy, essential with tri/tet mesh or when flow is not aligned with grid; convergence may be slower
  - MUSCL
    - Locally third-order convection discretization scheme for unstructured meshes
    - Based on blending of CD and SOU. More accurate in predicting secondary flows, vortices, forces, etc.
  - Quadratic Upwind Interpolation (QUICK)
    - applies to quad/hex and hybrid meshes (not applied to tri mesh), useful for rotating/swirling flows, 3rd-order accurate on uniform mesh

Interpolation Methods for Face Pressure

- Interpolation schemes for calculating cell-face pressures when using the segregated solver in FLUENT are available as follows:
  - Standard
    - default scheme; reduced accuracy for flows exhibiting large surface-normal pressure gradients near boundaries (but should not be used when steep pressure changes are present in the flow - PRESTO! scheme should be used)
  - Linear
    - use when other options result in convergence difficulties or unphysical behavior.
  - Second-Order
    - use for compressible flows; not to be used with porous media, jump, fans, etc. or VOF/Mixture multiphase models.
  - Body Force Weighted
    - use when body forces are large, e.g., high Ra natural convection or highly swirling flows.
  - PRESTO!
    - use for highly swirling flows, flows involving steep pressure gradients (porous media, fan model, etc.), or in strongly curved domains.

Pressure-Velocity Coupling

- Pressure-Velocity Coupling refers to the numerical algorithm which uses a combination of continuity and momentum equations to derive an equation for pressure (or *pressure correction*) when using the segregated solver
- Three algorithms available in FLUENT:
  - SIMPLE
    - default scheme, robust
  - SIMPLEC
    - Allows faster convergence for simple problems (e.g., laminar flows with no physical models employed).
  - PISO
    - useful for unsteady flow problems or for meshes containing cells with higher than average skewness

Initialization

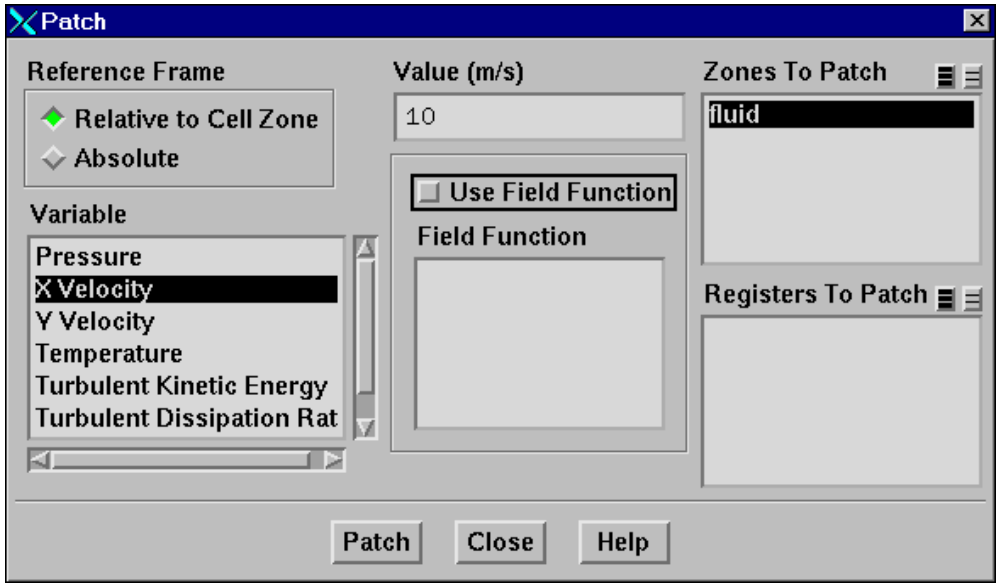
- Iterative procedure requires that all solution variables be initialized before calculating a solution.

  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)
  - Cell registers (created by marking the cells in the Adaption panel) can be used for "patching" different values in cell zones



Outline

- Using the Solver
  - Setting Solver Parameters
  - **Convergence**
    - **Definition**
    - **Monitoring**
    - **Stability**
    - **Accelerating Convergence**
  - Accuracy
    - Grid Independence
    - Grid Adaption
  - Unsteady Flows Modeling
    - Unsteady-flow problem setup
    - Non-iterative Transient Advancement (NITA) schemes
    - Unsteady flow modeling options
  - Summary
  - Appendix

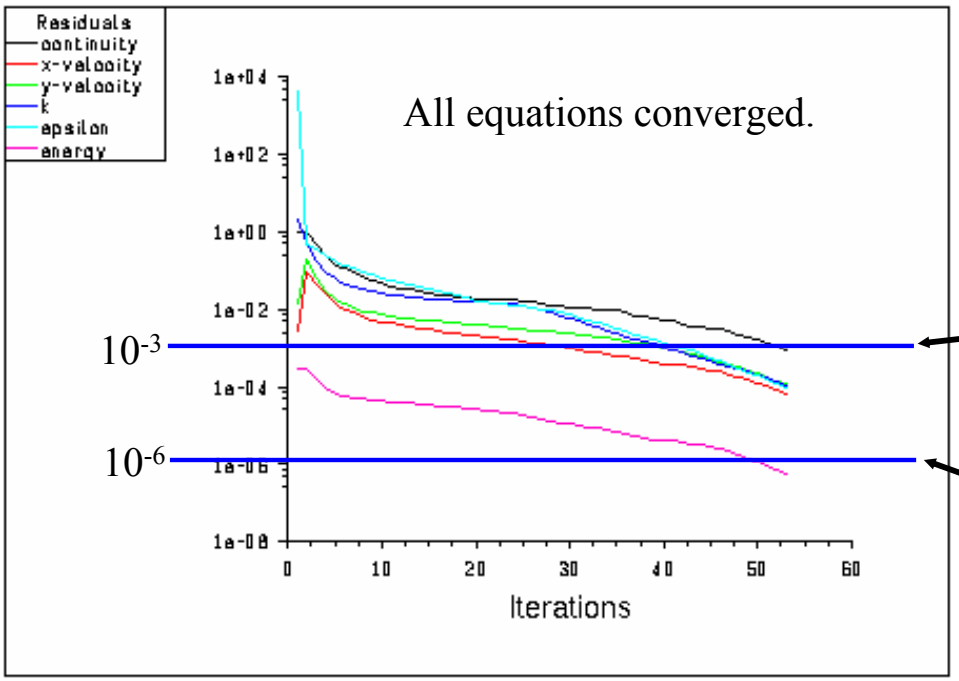
Convergence

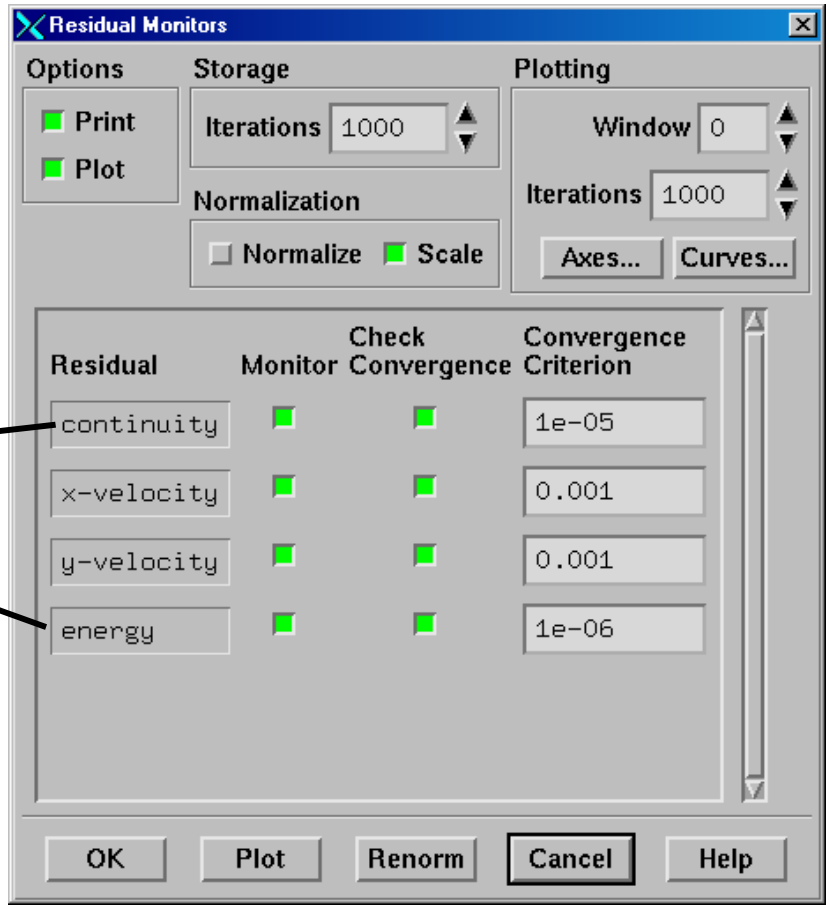
- At convergence:
  - All discrete conservation equations (momentum, energy, etc.) are obeyed in all cells *to a specified tolerance*.
  - Solution no longer changes with more iterations.
  - Overall mass, momentum, energy, and scalar balances are achieved.
- Monitoring convergence with residuals' history:
  - 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 relevant key variables/physical quantities for a confirmation.
  - Ensure that property conservation is satisfied.

Convergence Monitors: Residuals

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

  Solve → Monitors → Residual...



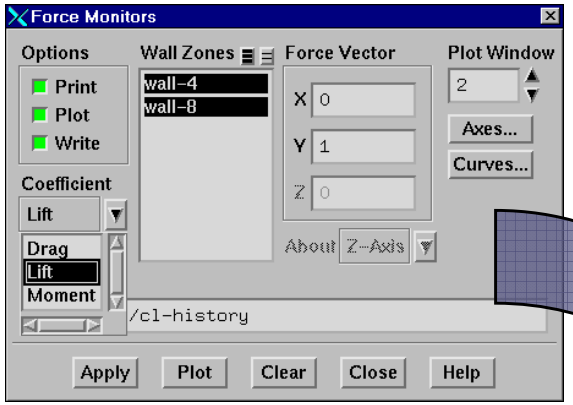


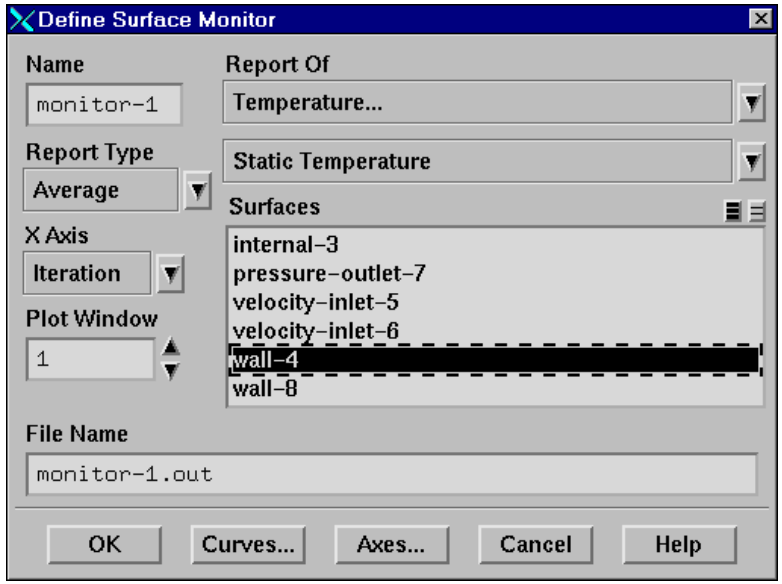
Convergence Monitors: Forces/Surfaces

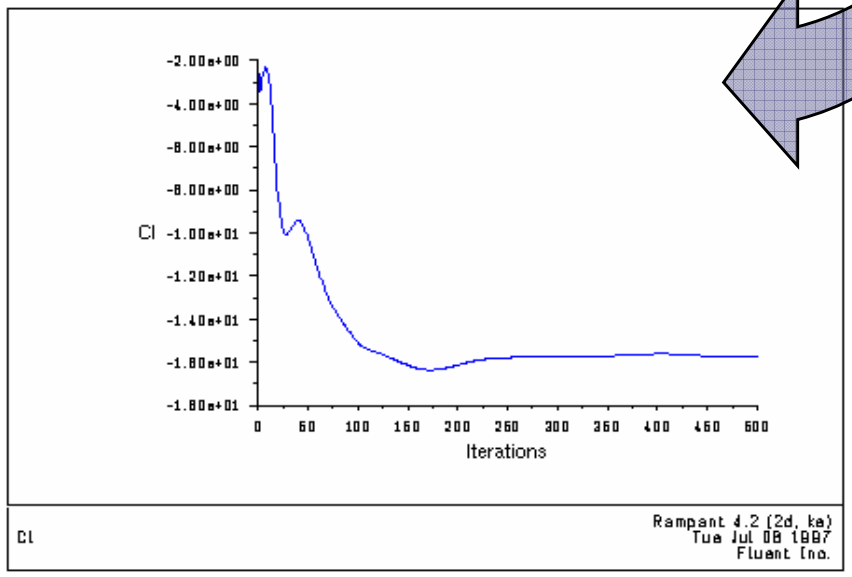
- In addition to residuals, you can also monitor:
  - Lift, drag, or moment

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

    Solve → Monitors → Surface...



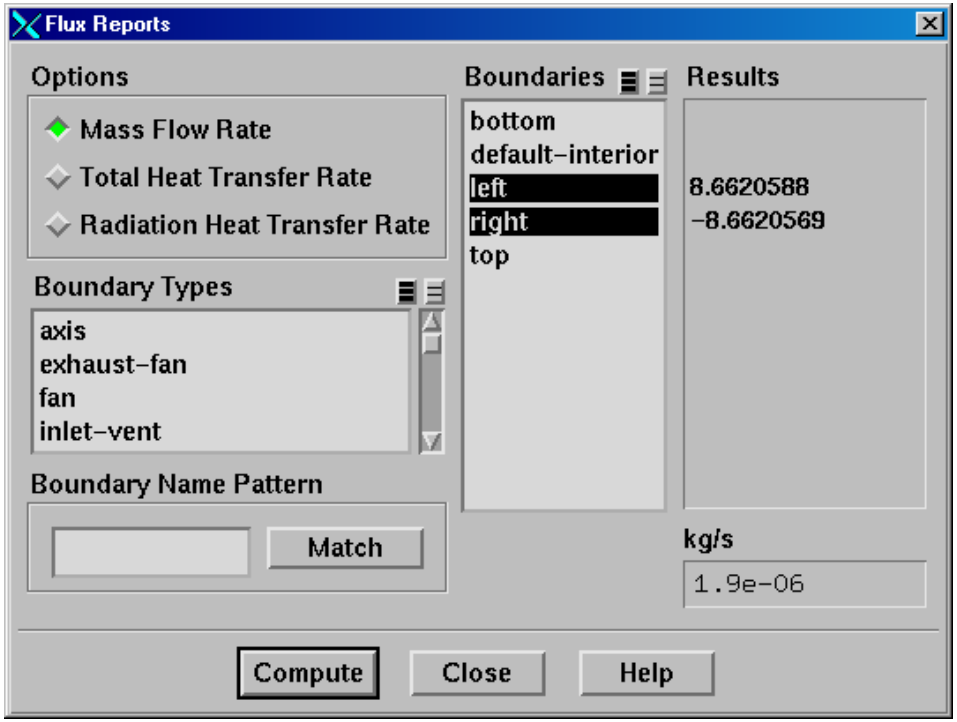




Checking for Property Conservation

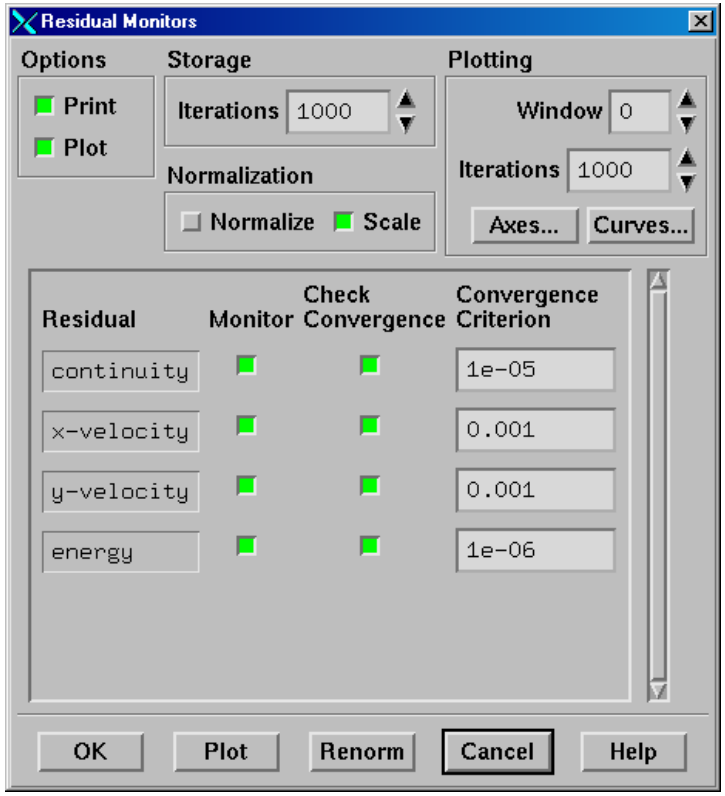
- In addition to monitoring residual and variable histories, you should also check for overall heat and mass balances.
  - At a minimum, the net imbalance should be less than 1% of smallest flux through domain boundary.

Report → Fluxes...



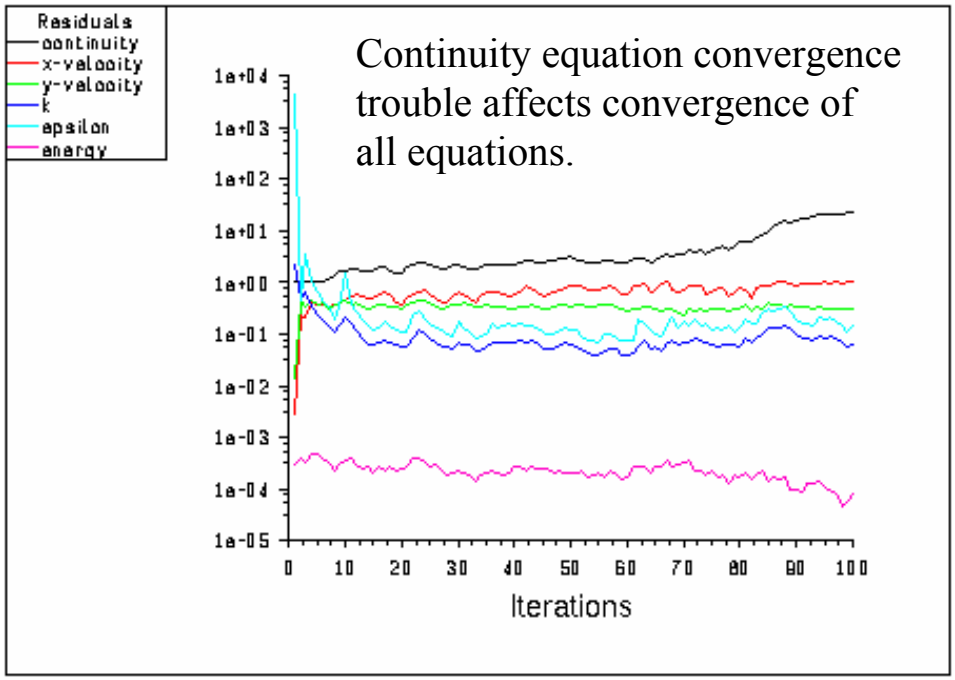
Tightening the Convergence Tolerance

- If your monitors indicate that the solution is converged, *but* the solution is still changing or has a large mass/heat imbalance, this clearly indicates the solution is not yet converged. You need to:
  - Tighten the Convergence Criterion or disable Check Convergence in "residual monitors" panel
  - Then iterate until solution converges



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 are very misleading!
- Troubleshooting:
  - Ensure the problem is well posed.
  - Compute an initial solution with a first-order discretization scheme.
  - Decrease under-relaxation for equations having convergence trouble (segregated solver).
  - Reduce *Courant* number (coupled solver).
  - Re-mesh or refine cells with high aspect ratio or highly skewed cells.

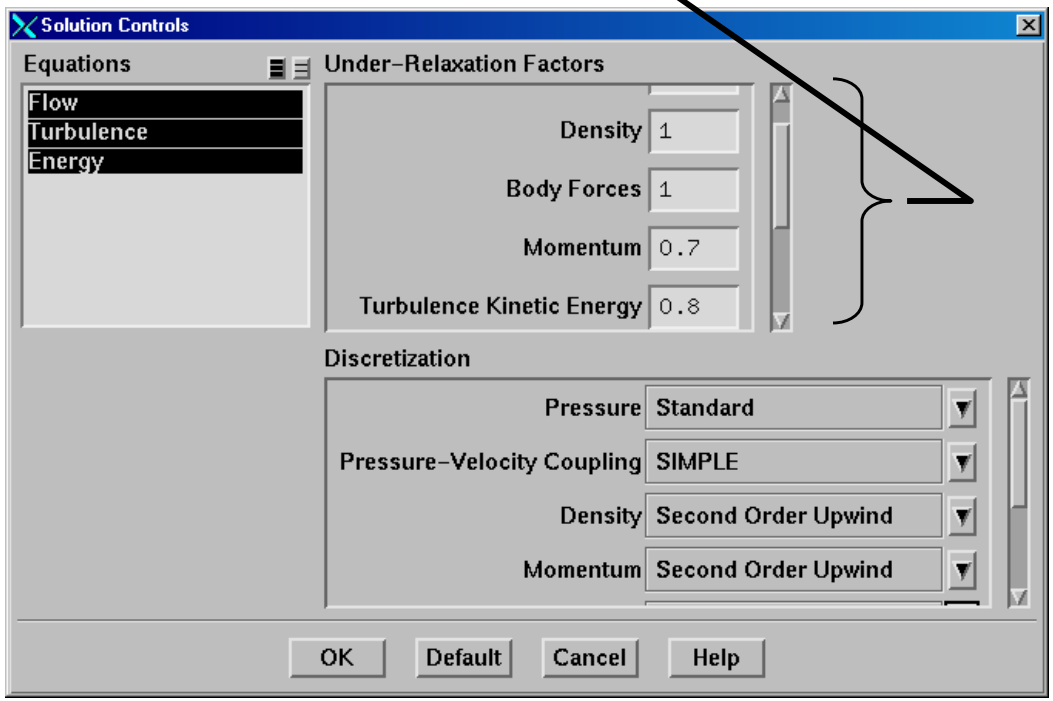


Modifying Under-relaxation Factors

- Under-relaxation factor, α, 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

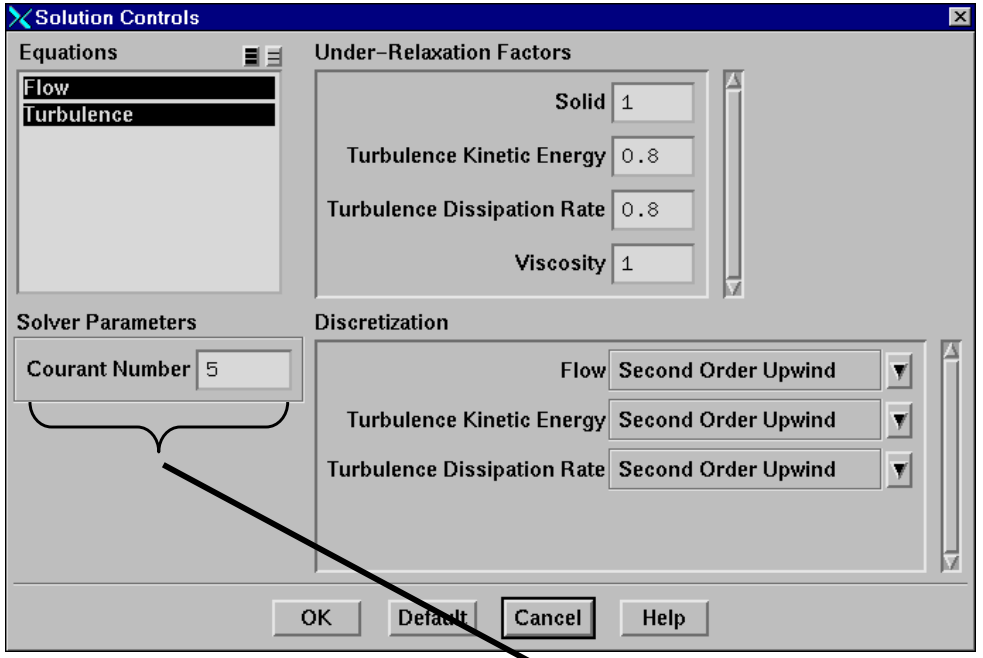
$$\phi_p = \phi_{p,old} + \alpha \Delta \phi_p$$



- 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

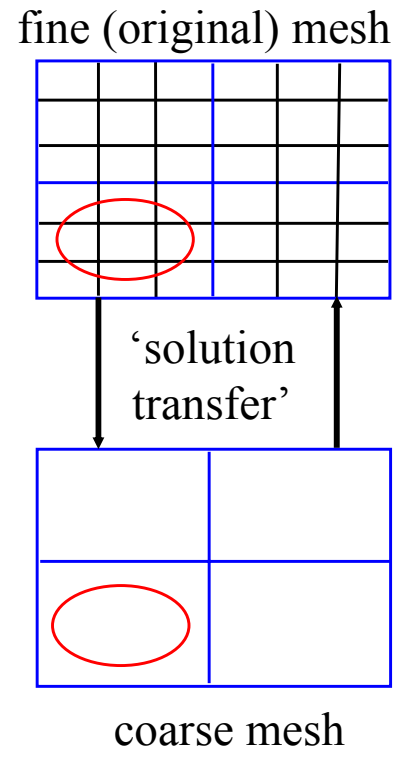
Starting from a Previous Solution

- Previous solution can be used as an initial condition when changes are made to problem definition.
  - Use File→Interpolate to initialize a run (especially useful for starting the fine-mesh cases when coarse-mesh solutions are available).
  - Once initialized, additional iterations always use *current data set* as the starting point.
  - Some suggestions on how to provide initial conditions for some actual problems:

Actual Problem	Initial Condition
flow with heat transfer	isothermal solution
natural convection	lower *Ra* solution
combustion	cold flow solution
turbulent flow	Euler solution

Multigrid Solver

- The Multigrid solver accelerates convergence by solving the discretized equations on multiple levels of mesh densities so that the "low-frequency" errors of the approximate solution can be efficiently eliminated
  - Influence of boundaries and far-away points are more easily transmitted to interior of coarse mesh than on fine mesh.
  - Coarse mesh defined from original mesh
    - Multiple coarse mesh 'levels' can be created.
      - Algebraic Multigrid (AMG): 'coarse mesh' emulated algebraically.
      - Full Approx. Storage Multigrid (FAS): 'cell coalescing' defines new grid.
        - a coupled-explicit solver option
    - Final solution is for original mesh
  - Multigrid solver operates automatically in the background
- Consult FLUENT User's Guide for additional options and technical details



Outline

- Using the Solver
  - Setting Solver Parameters
  - Convergence
    - Definition
    - Monitoring
    - Stability
    - Accelerating Convergence
  - **Accuracy**
    - **Grid Independence**
    - **Grid Adaption**
  - Unsteady Flows Modeling
    - Unsteady-flow problem setup
    - Non-iterative Transient Advancement (NITA) schemes
    - Unsteady flow modeling options
  - Summary
  - Appendix

Solution Accuracy

- A converged solution is not necessarily a correct one!
  - Always inspect and evaluate the solution by using available data, physical principles and so on.
  - Use the second-order upwind discretization scheme for final results.
  - Ensure that solution is grid-independent:
    - Use adaption to modify the grid or create additional meshes for the grid-independence study
- If flow features do not seem reasonable:
  - Reconsider physical models and boundary conditions
  - Examine mesh quality and possibly re-mesh the problem
  - Reconsider the choice of the boundaries' location (or the domain): inadequate choice of domain (especially the outlet boundary) can significantly impact solution accuracy

Mesh Quality and Solution Accuracy

- Numerical errors are associated with calculation of cell gradients and cell face interpolations.
- Ways to contain the numerical errors:
  - Use higher-order discretization schemes (second-order upwind, MUSCL)
  - Attempt to align grid with the flow to minimize the "false diffusion"
  - 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 in non-uniform meshes
      - 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 (but higher ratios are allowed in boundary layers)
      - 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 adaption.
        - If you know where large gradients are expected, you need to have fine grids in the original mesh for that region, e.g., boundary layers.
      - Adapt grid.
        - Data from original grid is automatically interpolated to finer grid.
        - FLUENT offers dynamic mesh adaption which automatically changes the mesh according to the criteria set by users
  - Continue calculation till convergence.
  - Compare results obtained w/different grids.
  - Repeat the procedure if necessary
- Different meshes on a single problem: Use TUI commands /file/write-bc and /file/read-bc to facilitate the set up of a new problem; better initialization can be obtained via interpolation from existing case/data by using
  - File → Interpolate...

Outline

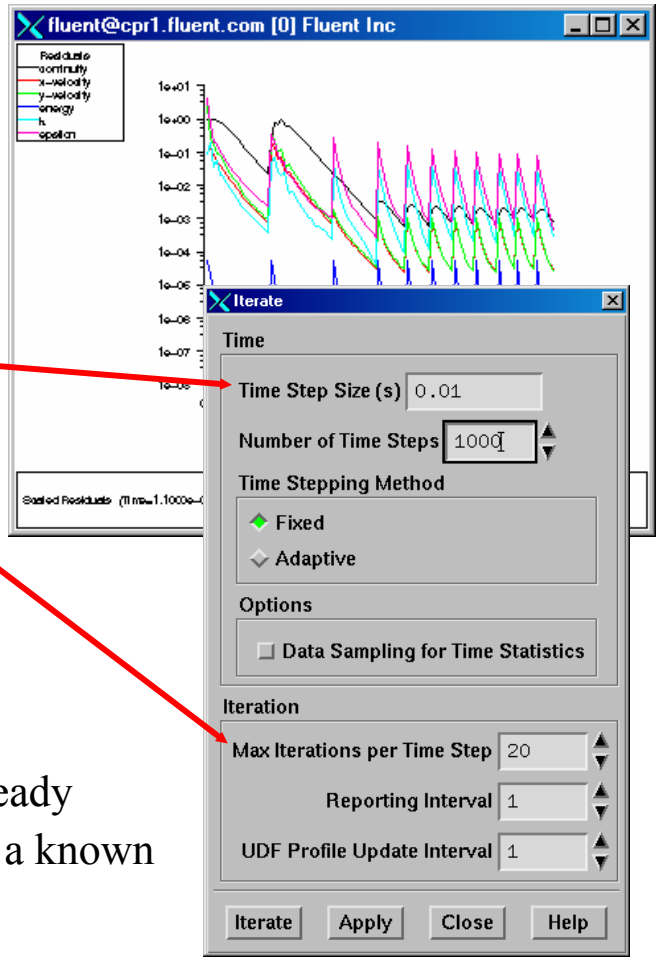
- Using the Solver
  - Setting Solver Parameters
  - Convergence
    - Definition
    - Monitoring
    - Stability
    - Accelerating Convergence
  - **Accuracy**
    - **Grid Independence**
    - **Grid Adaption**
  - **Unsteady Flows Modeling**
    - **Unsteady-flow problem setup**
    - **Non-iterative Transient Advancement (NITA) schemes**
    - **Unsteady flow modeling options**
  - Summary
  - Appendix

Unsteady Flow Modeling

- Transient solutions are available with both segregated and coupled solvers.
  - Solver iterates to convergence within each time level, then advances to the next (the Iterative Time Advancement (ITA) scheme)
  - Solution initialization defines initial condition and it must be realistic
- For segregated solver:
  - Time step size, Δt, is set in "Iterate" panel
    - Δt must be small enough to resolve time dependent features; make sure the convergence is reached within the "Max iterations per time step"
    - The time-step size's order of magnitude can be estimated as:

$$\Delta t \approx \frac{\text{typical cell size}}{\text{characteristic flow speed}}$$

    - Time-step size estimate can also be chosen so that the unsteady characteristics of the flow can be resolved (e.g., flow with a known period of fluctuations)
  - To iterate without advancing time step, use '0' time steps
  - PISO scheme may aid in accelerating convergence for unsteady flows



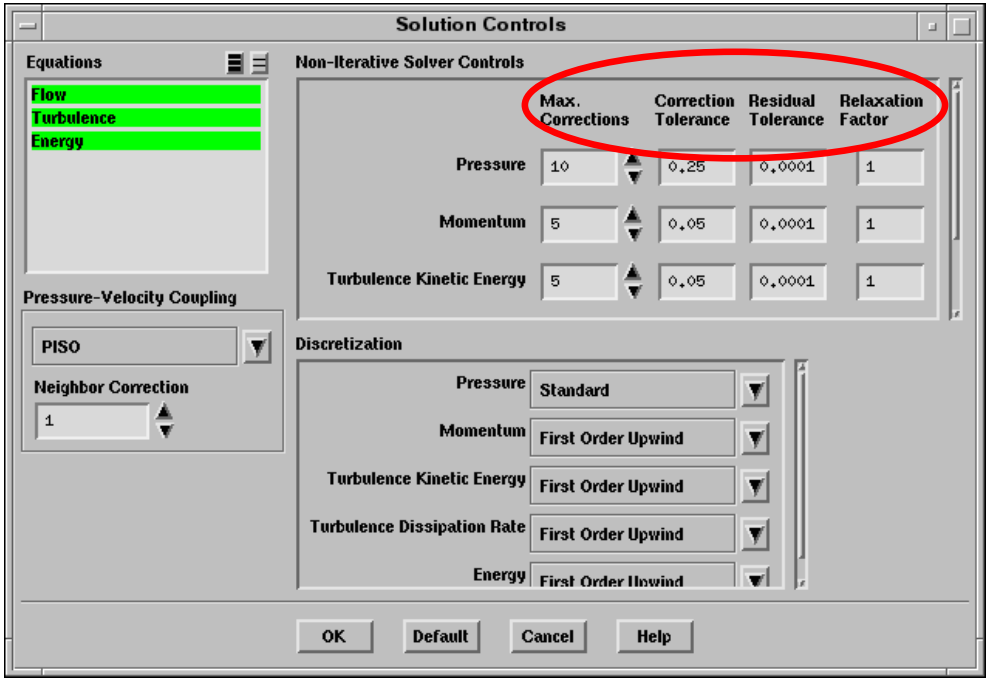
NITA Schemes for the Segregated Solver

Overall time-discretization error for 2nd-order scheme: $O(\Delta t^2)$ = Truncation error: $O(\Delta t^2)$ + Splitting error (due to eqn segregation): $O(\Delta t^n)$

- Non-iterative time advancement (NITA) schemes reduce the splitting error to $O(\Delta t^2)$ by using sub-iterations (not the more expensive *outer iterations to* eliminate the splitting errors used in ITA) per time step
- NITA runs about twice as fast as ITA scheme
- Two flavors of NITA schemes available in Fluent v6.2:
  - PISO (NITA/PISO)
    - Energy and turbulence equations are still loosely coupled
  - Fractional-step method (NITA/FSM)
    - About 20% cheaper than NITA/PISO on a per time-step basis
- NITA schemes have a wide range of applications for unsteady simulations: e.g., incompressible, compressible (subsonic, transonic), turbomachinery flows, etc.
- NITA schemes are not available for multiphase (except VOF), reacting flows, porous media, and fan models, etc. Consult Fluent User's Guide for additional details.

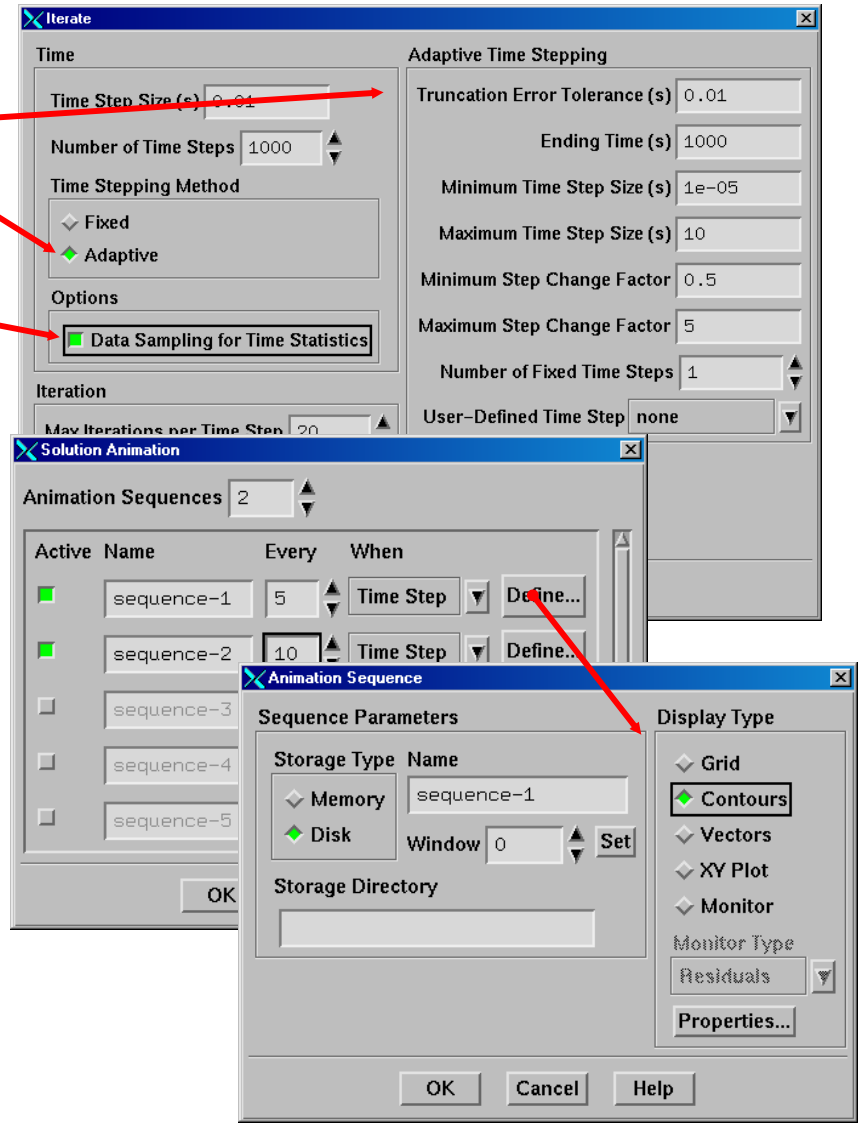
NITA Solution Control and Monitoring

- Sub-iterations are performed for discretized equations till the **Correction Tolerance** is met or the number of sub-iterations has reached the **Max Corrections**
- Algebraic multigrid (AMG) cycles are performed for each sub-iteration. AMG cycles terminate if the default AMG criterion is met or the **Residual Tolerance** is sastisfied for the last sub-iteration
- **Relaxation Factor** is used for solutions between each sub-iteration



Unsteady Flow Modeling Options

- Adaptive Time Stepping
  - Automatically adjusts time-step size based on local truncation error analysis
  - Customization possible via UDF
- Time-averaged statistics may be acquired.
  - Particularly useful for LES turbulence modeling
- If desirable, animations should be set up before iterating (for flow visualization)
- For the Coupled Solver, Courant number defines:
  - the global time-step size for coupled explicit solver
  - the pseudo time-step size for coupled implicit solver
    - Real time-step size must still be defined in the Iteration panel



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.

Appendix

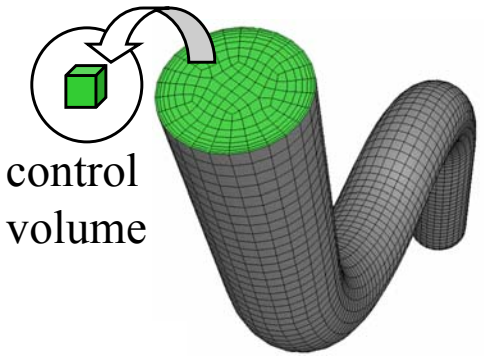
- Background
  - Finite Volume Method
  - Explicit vs. Implicit
  - Segregated vs. Coupled
  - Transient Solutions
  - Flow Diagrams of NITA and ITA Schemes

Background: Finite Volume Method - 1

- FLUENT solvers are based on the finite volume method.
  - Domain is discretized into a finite set of control volumes or cells.
- General transport equation for mass, momentum, energy, etc. is applied to *each* cell and discretized. For cell p,

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

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



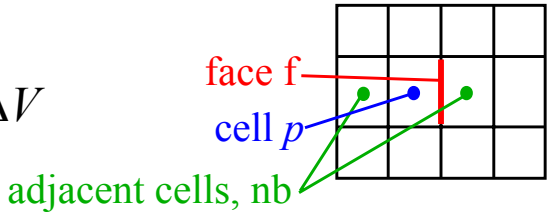
Fluid region of pipe flow discretized into finite set of control volumes (mesh).

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

Background: Finite Volume Method - 2

- Each transport equation is discretized into algebraic form. For cell p,

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



- Discretized equations require information at cell centers *and* faces.
  - Field data (material properties, velocities, etc.) are stored at cell centers.
  - Face values are interpolated in terms of local *and* adjacent cell values.
  - Discretization accuracy depends upon 'stencil' size.
- The discretized equation can be expressed simply as:

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

  - Equation is written out for every control volume in domain resulting in an *equation set*.

Background: Linearization

- Equation sets are solved iteratively.
  - Coefficients a_p and a_{nb} are typically functions of solution variables (nonlinear and coupled).

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

  - Coefficients are written to use values of solution variables from previous iteration.
    - Linearization: removing coefficients' dependencies on ϕ.
    - De-coupling: removing coefficients' dependencies on other solution variables.
  - Coefficients are updated with each outer iteration.
    - For a given inner iteration, coefficients are constant (frozen).
      - ϕ_p can either be solved explicitly or implicitly.

Background: Explicit vs. Implicit

- Assumptions are made about the knowledge of ϕ_{nb}:
  - Explicit linearization - unknown value in each cell computed from relations that include only *existing* values (ϕ_{nb} assumed known from previous iteration).
    - ϕ_p solved explicitly using Runge-Kutta scheme.
  - Implicit linearization - ϕ_p *and* ϕ_{nb} are assumed unknown and are solved using linear equation techniques.
    - Equations that are implicitly linearized tend to have less restrictive stability requirements.
    - The equation set is solved simultaneously using a second iterative loop (e.g., point Gauss-Seidel).

Background: Coupled vs. Segregated

- Segregated Solver
  - If the only unknowns in a given equation are assumed to be for a *single variable*, then the equation set can be solved without regard for the solution of other variables.
    - coefficients a_p and a_{nb} are scalars.

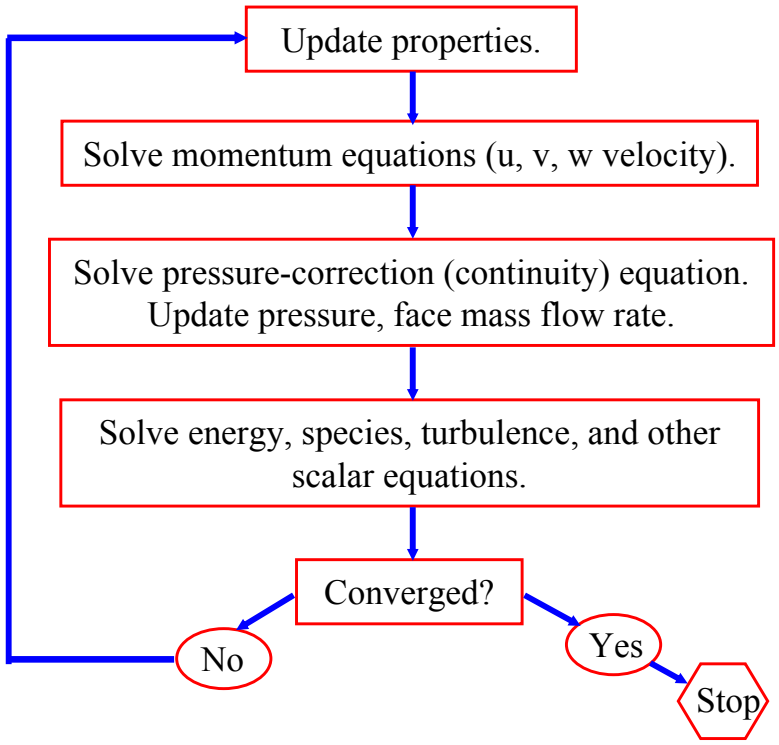
$$a_p \phi_p + \sum_{nb} a_{nb} \phi_{nb} = b_p$$

- Coupled Solver
  - If *more than one variable* is unknown in each equation, and each variable is defined by its own transport equation, then the equation set is coupled together.
    - coefficients a_p and a_{nb} are $N_{eq} x\ N_{eq}$ matrices
    - ϕ is a vector of the dependent variables, $\{p, u, v, w, T, Y\}^T$

Background: Segregated Solver

- In the segregated solver, each equation is solved separately.
- The continuity equation takes the form of a pressure correction equation as part of SIMPLE algorithm.
- Under-relaxation factors are included in the discretized equations.
  - Included to improve stability of iterative process.
  - Under-relaxation factor, α, in effect, limits change in variable from one iteration to next:

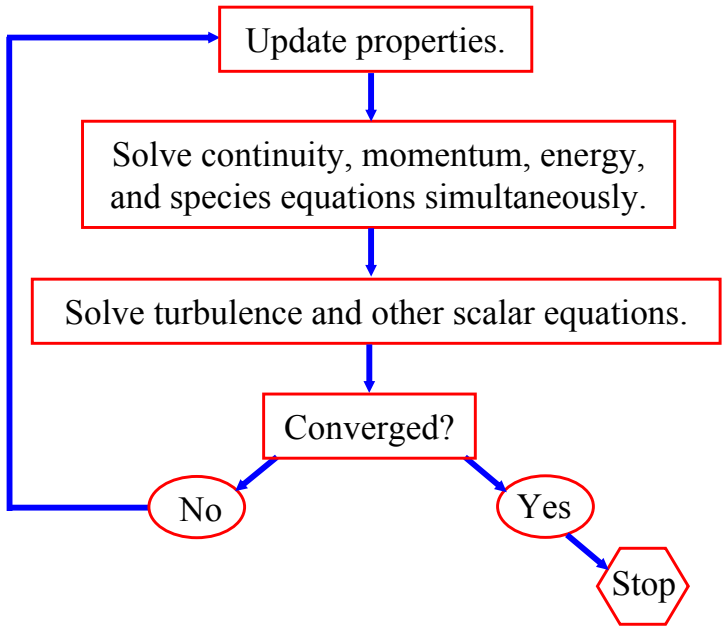
$$\phi_p = \phi_{p,old} + \alpha \Delta \phi_p$$



Background: Coupled Solver

- Continuity, momentum, energy, and species are solved simultaneously in the coupled solver.
- Equations are modified to resolve compressible *and* incompressible flow.
- Transient term is always included.
  - Steady-state solution is formed as time increases and transients tend to zero.
- For steady-state problem, 'time step' is defined by Courant number.
  - Stability issues limit maximum time step size for explicit solver but not for implicit solver.

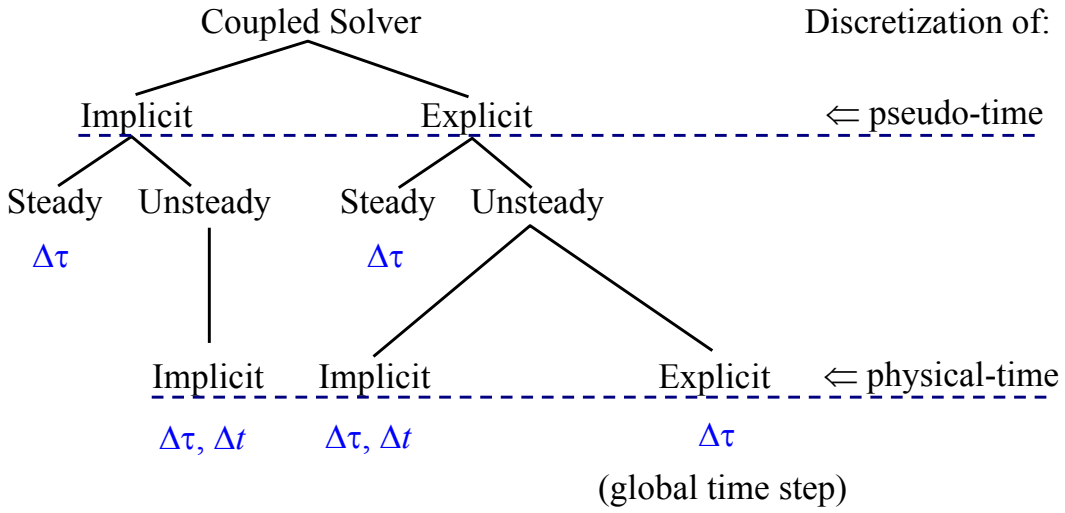
$$\Delta t = \frac{(\mathrm{CFL})\Delta x}{u} \quad \textit{where}$$

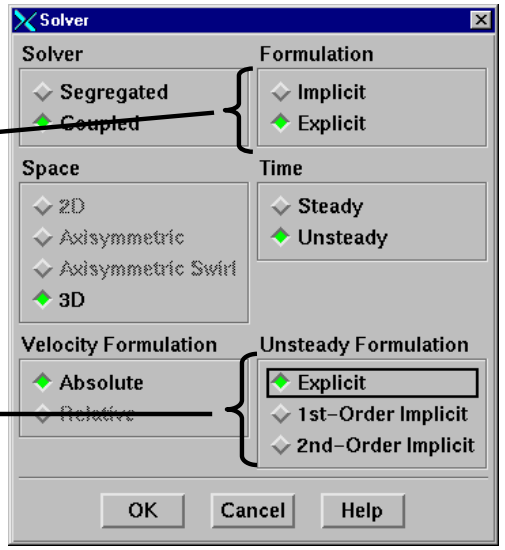


CFL = Courant-Friedrichs-Lewy-number
u = appropriate velocity scale
Δx = grid spacing

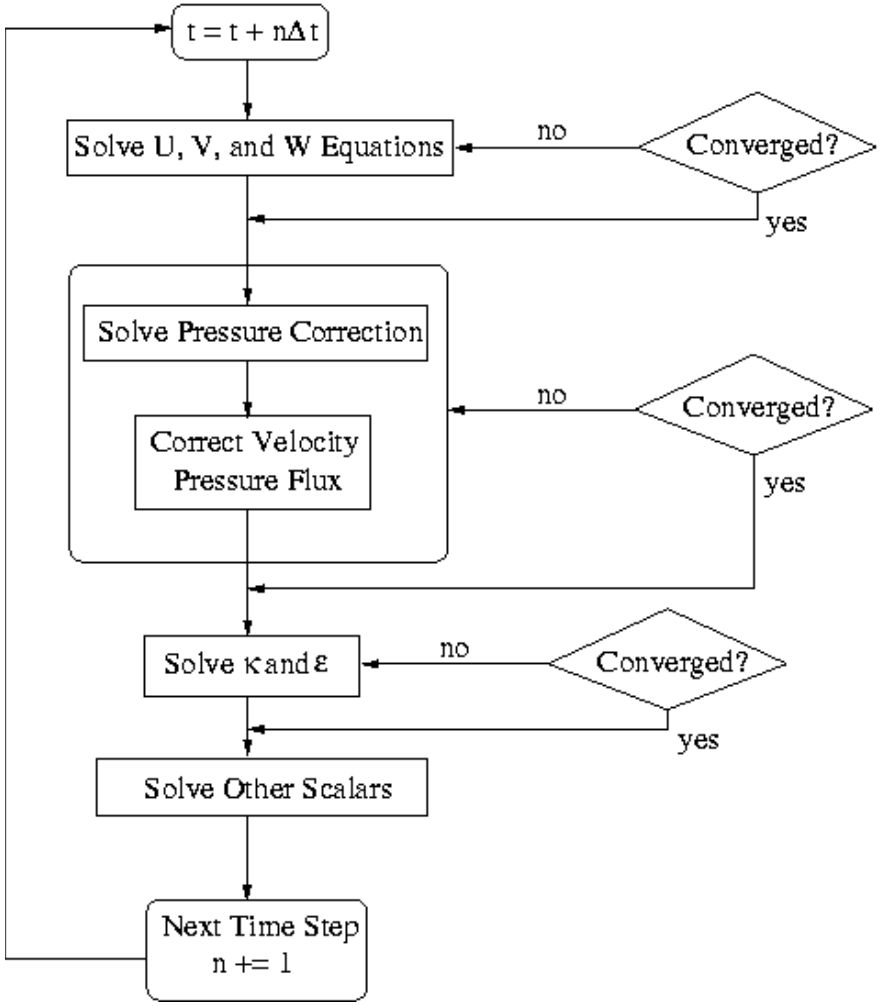
Background: Coupled/Transient Terms

- Coupled solver equations always contain a transient term.
- Equations solved using the unsteady coupled solver may contain *two* transient terms:
  - Pseudo-time term, $\Delta\tau$.
  - Physical-time term, Δt.
- Pseudo-time term is driven to near zero at each time step and for steady flows.
- Flow chart indicates which time step size inputs are required.
  - Courant number defines $\Delta\tau$
  - Inputs to Iterate panel define Δt.

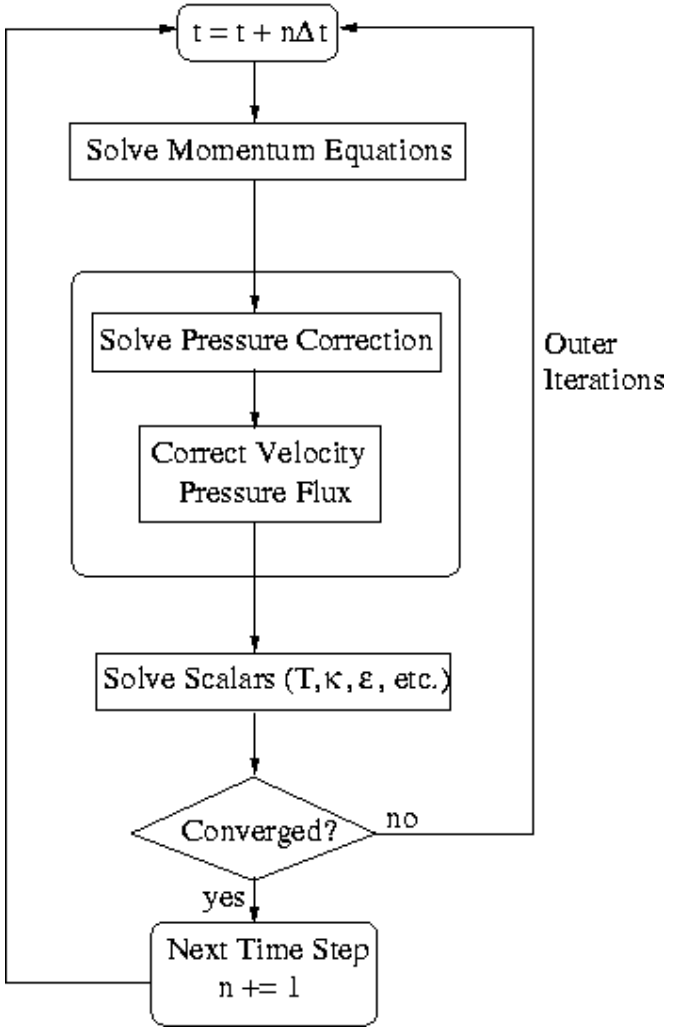




NITA versus ITA

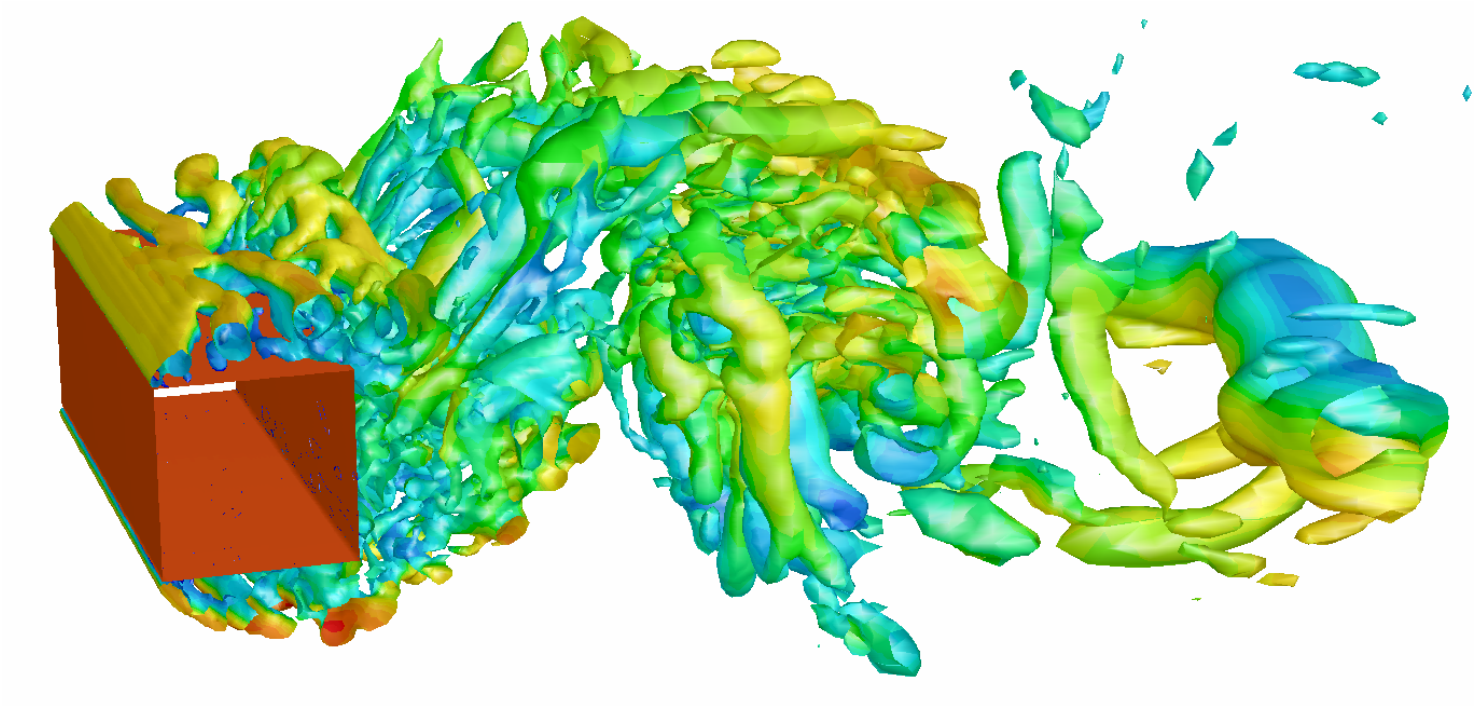


NITA scheme



ITA scheme

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 characterize turbulent eddies.
  - Enhanced mixing (matter, momentum, energy, etc.) results
- Fluid properties and velocity exhibit random variations
  - Statistical averaging results in accountable, turbulence related transport mechanisms.
  - This characteristic allows for *Turbulence Modeling*.
- Contains a wide range of turbulent eddy sizes (scales spectrum).
  - The size/velocity of large eddies is on the order of mean flow.
    - Large eddies derive energy from the mean flow
  - Energy is transferred from larger eddies to smaller eddies
    - In the smallest eddies, turbulent energy is converted to internal energy by viscous dissipation.

Is the Flow Turbulent?

External Flows

$Re_x \geq 5\times10^5$ along a surface

$Re_D \geq 20{,}000$ around an obstacle

Internal Flows

$Re_{D_h} \geq 2{,}300$

where $Re_L \equiv \dfrac{\rho U L}{\mu}$

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

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

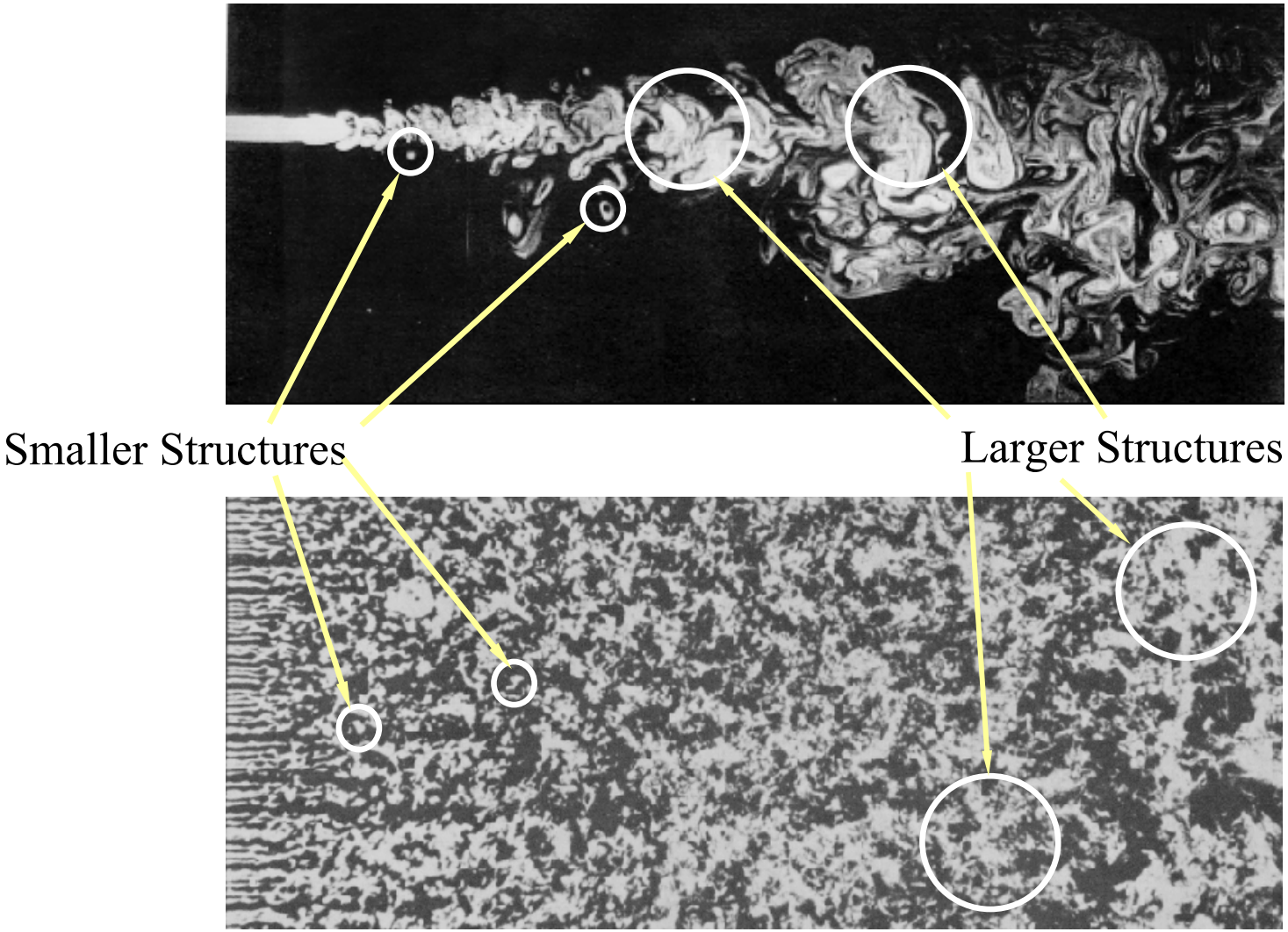
Natural Convection

$\mathrm{Ra} \geq 10^9\,\mathrm{Pr}$

where $\mathrm{Ra} = \dfrac{g\beta\Delta T L^3}{\alpha\nu}$

is the Rayleigh number

Two Examples of Turbulent Flow

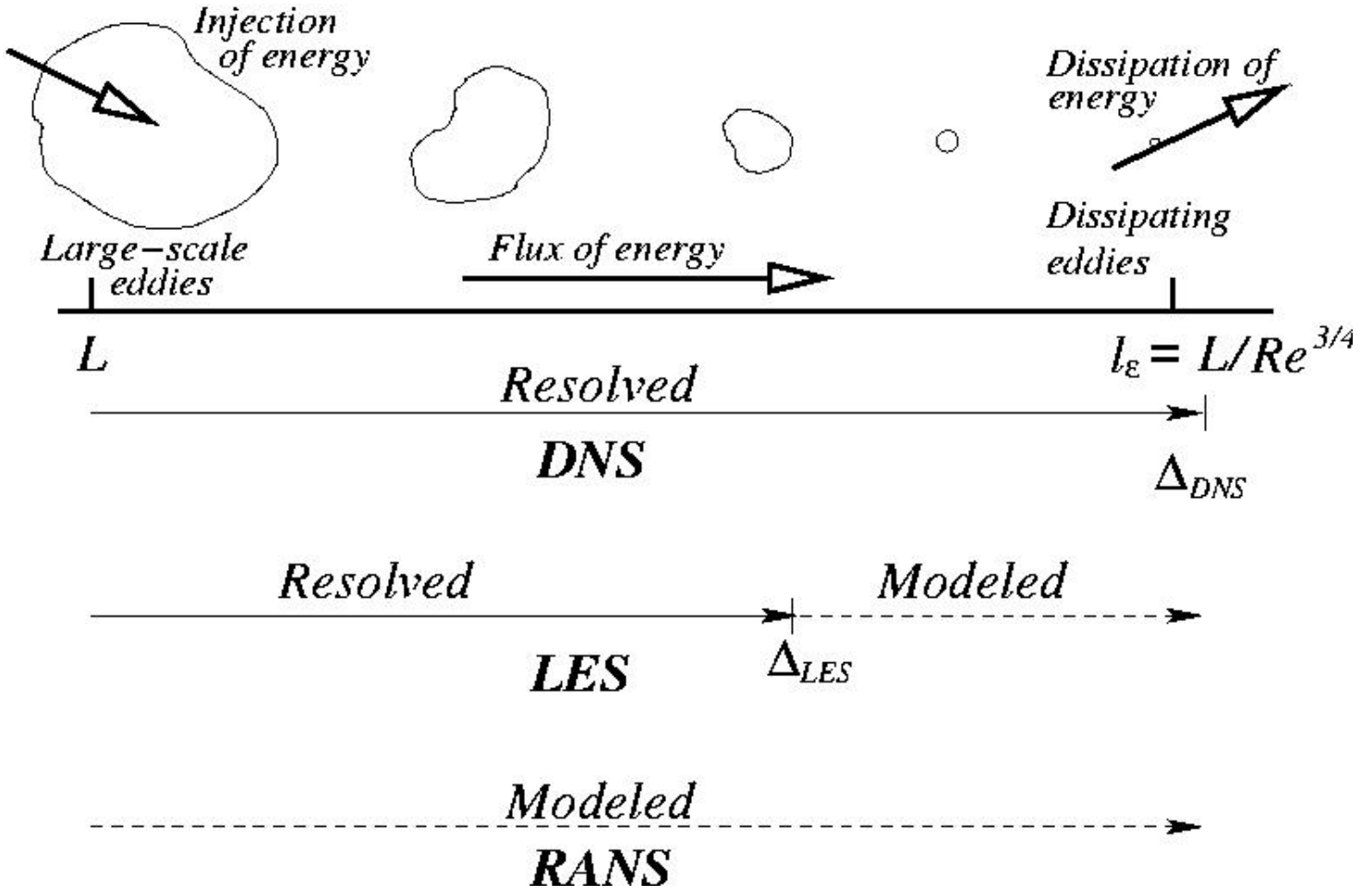


Overview of Computational Approaches

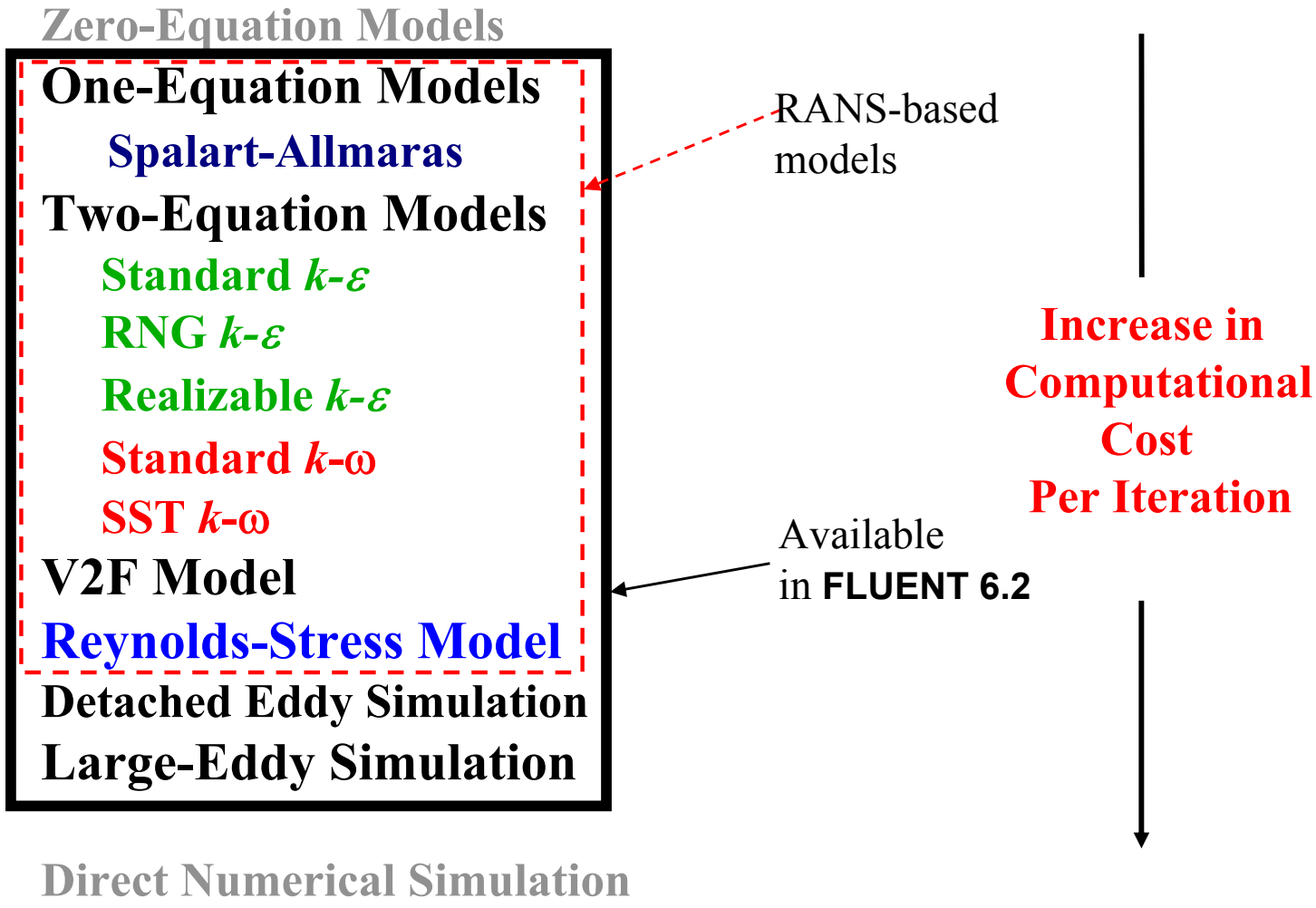
- Direct Numerical Simulation (DNS)
  - Theoretically all turbulent flows can be simulated by numerically solving the full Navier-Stokes equations.
  - Resolves the whole spectrum of scales. No modeling is required.
  - But the cost is too prohibitive! Not practical for industrial flows - DNS is not available in Fluent.
- Large Eddy Simulation (LES)
  - Solves the spatially averaged N-S equations. Large eddies are directly resolved, but eddies smaller than the mesh sizes are modeled.
  - Less expensive than DNS, but the amount of computational resources and efforts are still too large for most practical applications.
- Reynolds-Averaged Navier-Stokes (RANS) Equations Models
  - Solve ensemble-averaged Navier-Stokes equations
  - All turbulence scales are modeled in RANS.
  - The most widely used approach for calculating industrial flows.
- There is not yet a single turbulence model that can reliably predict *all* turbulent flows found in industrial applications with sufficient accuracy.

Turbulence Scales and Prediction Methods

energy cascade
(Richardson, 1922)



Turbulence Models in Fluent

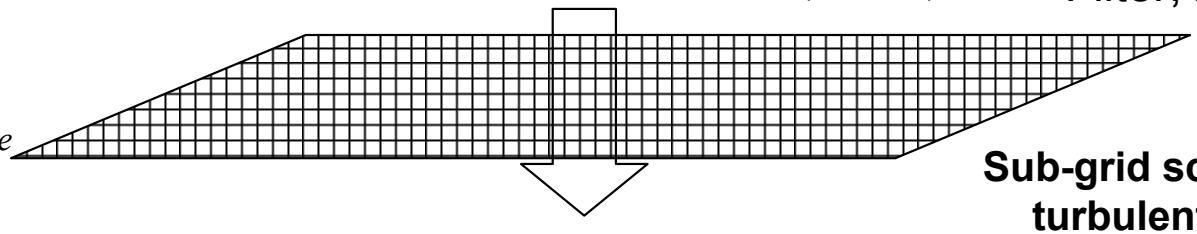


Large Eddy Simulation (LES)

N-S equation

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

$$\mathbf{u}(\mathbf{x},t) = \underbrace{\bar{\mathbf{u}}(\mathbf{x},t)}_{\textit{resolved scale}} + \underbrace{\mathbf{u}'(\mathbf{x},t)}_{\textit{subgrid scale}}$$



Sub-grid scale (SGS) turbulent stress

Filtered N-S equation

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial \bar{u}_i \bar{u}_j}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\nu \frac{\partial \bar{u}_i}{\partial x_j}\right) - \boxed{\frac{\partial \tau_{ij}}{\partial x_j}} \longrightarrow \tau_{ij} \equiv \overline{u_i u_j} - \bar{u}_i \bar{u}_j$$

- Spectrum of turbulent eddies in the Navier-Stokes equations is filtered:
  - The filter is a function of grid size
  - Eddies smaller than the grid size are removed and modeled by a sub-grid scale (SGS) model
  - Larger eddies are directly solved numerically by the filtered transient N-S equation

LES in FLUENT

- LES has been most successful for high-end applications where the RANS models fail to meet the needs. For example:
  - Combustion
  - Mixing
  - External Aerodynamics (flows around bluff bodies)
- Implementations in FLUENT:
  - Sub-grid scale (SGS) turbulent models:
    - Smagorinsky-Lilly model
    - WALE model
    - Dynamic Smagorinsky-Lilly model
    - Dynamic kinetic energy transport model
    - Detached eddy simulation (DES) model
- LES is applicable to all combustion models in FLUENT
- Basic statistical tools are available: Time averaged and root-mean-square (RMS) values of solution variables, built-in FFT
- Before running LES, one should consult guidelines in the "Best Practices For LES" (containing advice for mesh, SGS models, numerics, BC's, and more)

Detached Eddy Simulation (DES)

- Motivation
  - For high-Re wall bounded flows, LES becomes prohibitively expensive to resolve the near-wall region
  - Using RANS in near-wall regions would significantly mitigate the mesh resolution requirement
- RANS/LES hybrid model based on the Spalart-Allmaras turbulence model:

$$\frac{D\tilde{\nu}}{Dt} = C_{b1}\tilde{S}\tilde{\nu} - C_{w1}f_w\left(\frac{\tilde{\nu}}{\bar{d}}\right)^2 + \frac{1}{\sigma_{\tilde{\nu}}}\left[\frac{\partial}{\partial x_j}\left\{(\mu+\rho\tilde{\nu})\frac{\partial\tilde{\nu}}{\partial x_j}\right\}+\ldots\right]$$

$$\bar{d} = \min\left(d_w, C_{DES}\Delta\right)$$

  - One-equation SGS turbulence model
  - In equilibrium, it reduces to an algebraic model.
- DES is a practical alternative to LES for high-Reynolds number flows in external aerodynamic applications

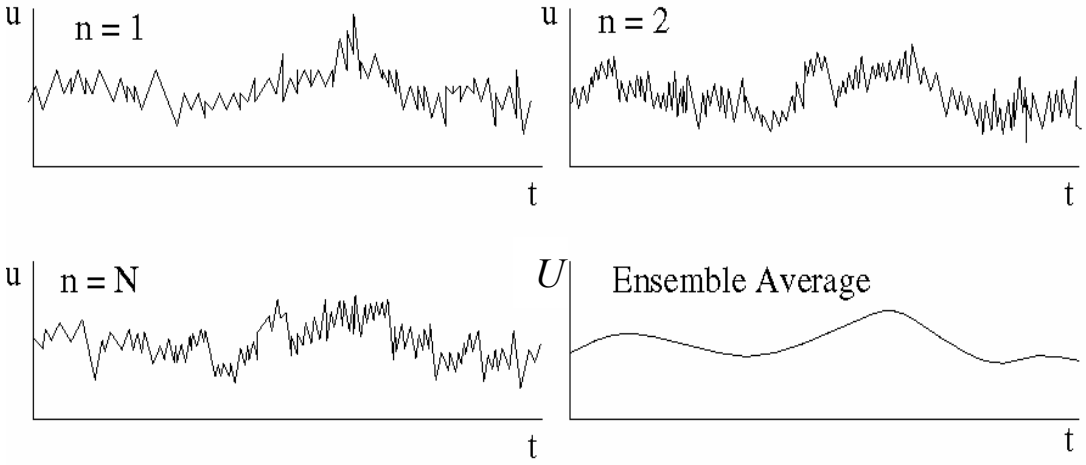
RANS Modeling: Ensemble-Averaging

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

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

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

Mean fluctuation



- The Reynolds-averaged momentum equations are as follows:

$$\rho\left(\frac{\partial U_i}{\partial t} + U_k\frac{\partial U_i}{\partial x_k}\right) = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\mu\frac{\partial U_i}{\partial x_j}\right) + \frac{\partial R_{ij}}{\partial x_j}$$

where $R_{ij} = -\rho\overline{u_i'u_j'}$ is called the Reynolds stresses. The Reynolds stresses are additional unknowns introduced by the averaging procedure, hence they must be modeled (related to the averaged flow quantities) in order to close the equations.

The Closure Problem

- The RANS models can be closed in one of the following ways:

(1) Eddy-Viscosity Models (EVM):

$$R_{ij} = -\rho\overline{u_i' u_j'} = \mu_t\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right) - \frac{2}{3}\mu_t \frac{\partial U_k}{\partial x_k}\delta_{ij} - \frac{2}{3}\rho k \delta_{ij}$$

Boussinesq hypothesis – Reynolds stresses are modeled using an eddy (or turbulent) viscosity μ_t. (The hypothesis is reasonable for simple turbulent shear flows: boundary layers, round jets, mixing layers, channel flows, etc.)

(2) Reynolds-Stress Models (RSM): solving transport equations for the individual Reynolds stresses:

- Modeling is still required for many terms in the transport equations.
- RSM is more advantageous in complex 3-D turbulent flows with large streamline curvature and swirl, but the model is more complex, computationally intensive, more difficult to converge than eddy-viscosity models.

Calculating μ_t for the Eddy-Viscosity Models

- Based on *dimensional analysis*, μ_t can be determined from a turbulence time scale (or velocity scale) and a length scale:
  - $k \equiv \overline{u_i' u_i'}/2$ is the turbulent kinetic energy [L^2/T^2]
  - $\varepsilon \equiv \nu \overline{\partial u_i'/\partial x_j (\partial u_i'/\partial x_j + \partial u_j'/\partial x_i)}$ is the turbulence dissipation rate [L^2/T^3]
  - $\omega \sim \varepsilon/k$ is the specific dissipation rate [1/T]
- μ_t is calculated differently under various turbulence models:
  - Spalart-Allmaras:
    - This one-equation model solves only one transport equation for a modified turbulent viscosity.

    $$\mu_t \equiv f(\tilde{\nu})$$
  - Standard *k*-ε, RNG *k*-ε, Realizable *k*-ε
    - These two-equation models solve transport equations for k and ε.

    $$\mu_t \equiv f\left(\frac{\rho k^2}{\varepsilon}\right)$$
  - Standard *k*-ω, SST *k*-ω
    - These two-equation models solve transport equations for *k* and ω.

    $$\mu_t \equiv f\left(\frac{\rho k}{\omega}\right)$$

RANS Models - Spalart-Allmaras Model

A low-cost model solving an equation for the modified eddy viscosity $\tilde{\nu}$

$$\frac{D\tilde{\nu}}{Dt} = G_\nu + \frac{1}{\sigma_{\tilde{\nu}}}\left[\frac{\partial}{\partial x_j}\left\{(\mu + \rho\tilde{\nu})\frac{\partial \tilde{\nu}}{\partial x_j}\right\} + C_{b2}\rho\left(\frac{\partial \tilde{\nu}}{\partial x_j}\right)^2\right] - Y_\nu + S_{\tilde{\nu}}$$

- Eddy-viscosity is obtained from

$$\mu_t = \rho\,\tilde{\nu}\,f_{v1}, \quad f_{v1} \equiv \frac{(\tilde{\nu}/\nu)^3}{(\tilde{\nu}/\nu)^3 + C_{v1}^3}$$

- The variation of $\tilde{\nu}$ very near the wall is easier to resolve than k and ε.
- Mainly intended for aerodynamic/turbo-machinery applications with mild separation, such as supersonic/transonic flows over airfoils, boundary-layer flows, etc.

RANS Models - Standard k-ε (SKE) Model

- Transport equations for k and ε:

$$\frac{D}{Dt}(\rho k) = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] + G_k - \rho\varepsilon$$

$$\frac{D}{Dt}(\rho \varepsilon) = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right] + C_{e1}\frac{\varepsilon}{k}G_k - \rho C_{\varepsilon 2}\frac{\varepsilon^2}{k}$$

where $C_\mu = 0.09, \quad C_{\varepsilon 1} = 1.44, \quad C_{\varepsilon 2} = 1.92, \quad \sigma_k = 1.0, \quad \sigma_\varepsilon = 1.3$

- The most widely-used engineering turbulence model for industrial applications
- Robust and reasonably accurate; it has many sub-models for compressibility, buoyancy, and combustion, etc.
- Performs poorly for flows with strong separation, large streamline curvature, and high pressure gradient.

RANS Models - Realizable k-ε (RKE) Model

- Realizable k-ε (RKE) model ensures "realizability" of the k-ε model, i.e.,
  - Positivity of normal stresses
  - Schwarz' inequality for Reynolds shear-stresses
- Good performance for flows with axisymmetric jets.

RANS Models - RNG k-ε Model

- Constants in the k-ε equations are derived using the Renormalization Group method.
- RNG's sub-models include:
  - Differential viscosity model to account for low-Re effects
  - Analytically derived algebraic formula for turbulent Prandtl/Schmidt number
  - Swirl modification
- Performs better than SKE for more complex shear flows, and flows with high strain rates, swirl, and separation.

RANS Models - k-ω Models

$$\mu_t = \alpha^* \rho \frac{k}{\omega}$$

$$\rho \frac{Dk}{Dt} = \tau_{ij} \frac{\partial U_i}{\partial x_j} - \rho \beta^* f_{\beta^*} k\omega + \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right]$$

$$\rho \frac{D\omega}{Dt} = \alpha \frac{\omega}{k} \tau_{ij} \frac{\partial U_i}{\partial x_j} - \rho \beta f_\beta \omega^2 + \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_\omega}\right)\frac{\partial \omega}{\partial x_j}\right]$$

specific dissipation rate: ω

$$\omega \approx \frac{\varepsilon}{k} \propto \frac{1}{\tau}$$

- Belongs to the general 2-equation EVM family. Fluent 6 supports the standard k-ω model by Wilcox (1998), and Menter's SST k-ω model (1994).
- k-ω models have gained popularity mainly because:
  - Can be integrated to the wall without using any damping functions
  - Accurate and robust for a wide range of boundary layer flows with pressure gradient
- Most widely adopted in the aerospace and turbo-machinery communities.
- Several sub-models/options of k-ω : compressibility effects, transitional flows and shear-flow corrections.

RANS-Models - Reynolds-Stress Model (RSM)

$$\frac{\partial}{\partial t}\left(\rho\overline{u_i'u_j'}\right)+\frac{\partial}{\partial x_k}\left(\rho U_k\overline{u_i'u_j'}\right)=P_{ij}+F_{ij}+D_{ij}^{T}+\Phi_{ij}-\varepsilon_{ij}$$

Stress-production

Rotation-production

Turbulent diffusion

Dissipation

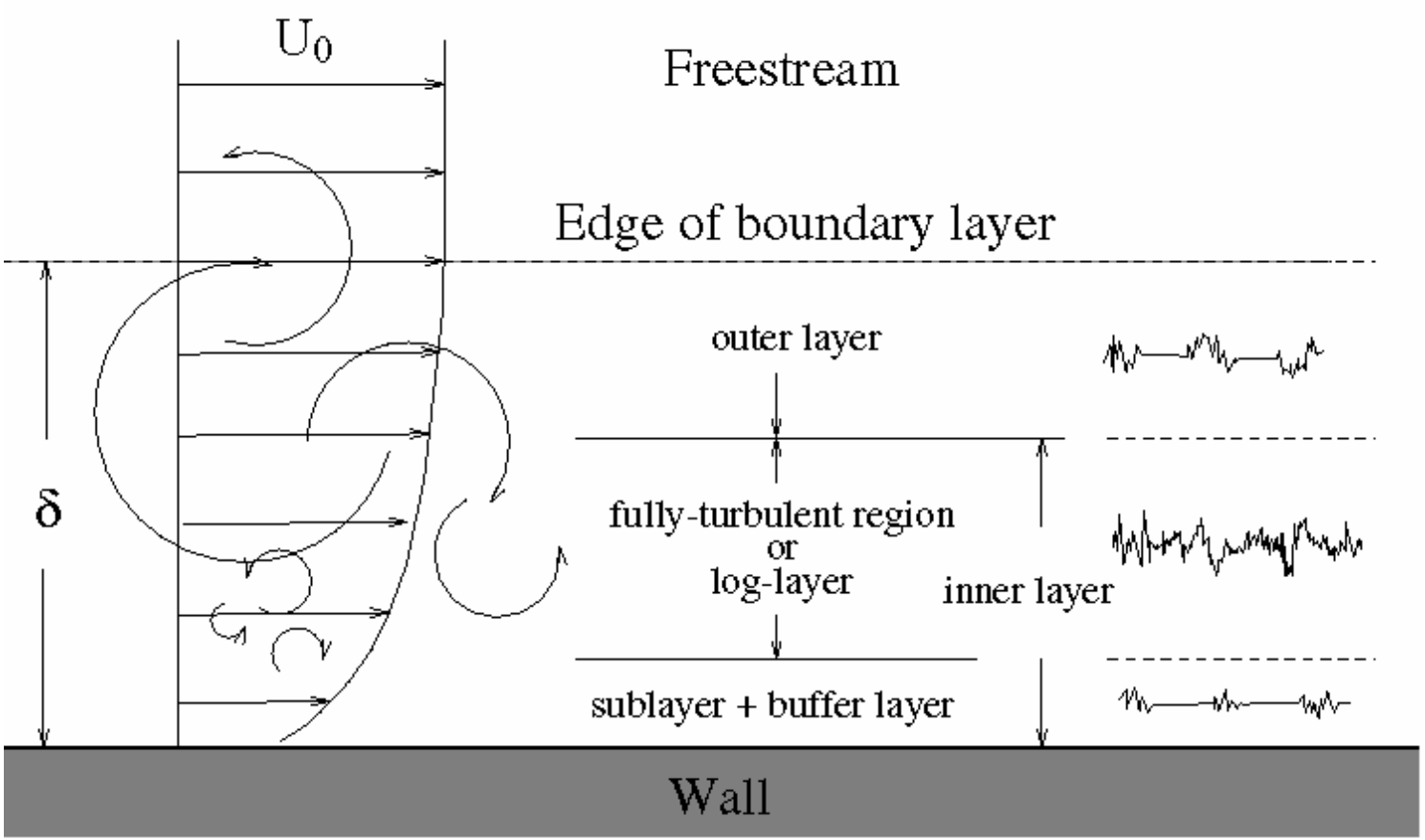
Pressure strain

Modeling required for these terms

- Attempts to address the deficiencies of the EVM.
- RSM is the most 'physically sound' model: anisotropy, history effects and transport of Reynolds stresses are directly accounted for.
- RSM requires substantially more modeling for the governing equations (the pressure-strain is most critical and difficult one among them).
- But RSM is more costly and difficult to converge than the 2-equation models.
- Most suitable for complex 3-D flows with strong streamline curvature, swirl and rotation.

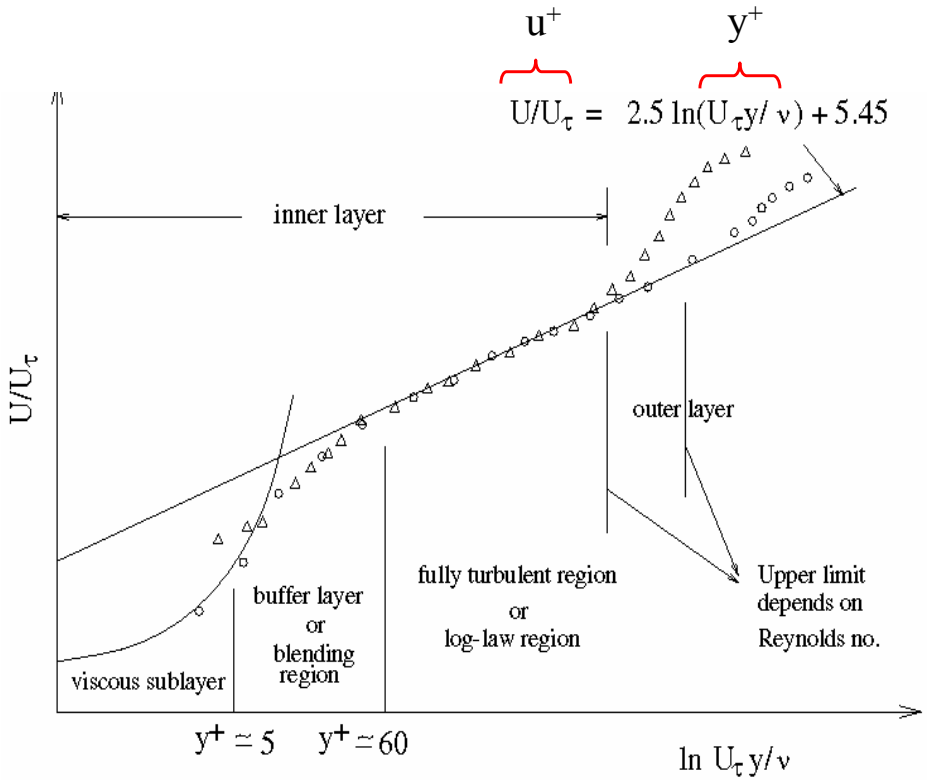
Near-Wall Treatments: The Structure of Near-Wall Flows

- The structure of turbulent boundary layers in the near-wall region:



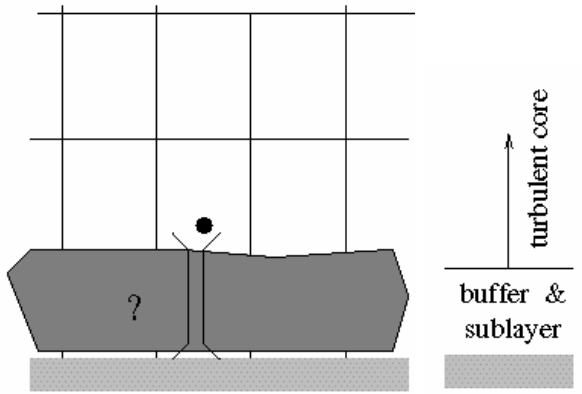
Wall Boundary Conditions

- Accurate near-wall modeling is important:
  - Successful prediction of frictional drag, pressure drop, separation, etc., depends on the fidelity of local wall shear predictions.
  - Near-wall modeling is used to supply boundary conditions for turbulent flows.
- Most k-ε and RSM turbulence models are not valid in the near-wall region:
  - Special near-wall treatment is required to provide proper BC's:
    - Standard wall functions
    - Non-Equilibrium wall functions
    - Enhanced wall treatment
- S-A, k-ω models are capable of resolving the steep near-wall profiles - provided the mesh is sufficiently fine.

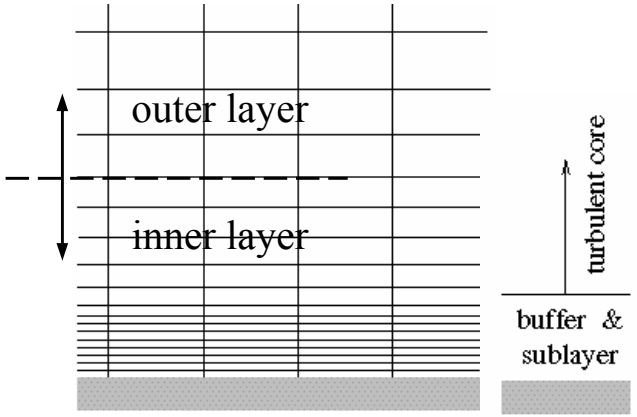


Near-Wall Modeling Options

- In general, wall functions are a *collection or set of laws* that serve as boundary conditions for momentum, energy, and species as well as for turbulence quantities.
- **Wall Function Options**
  - The Standard and Non-equilibrium Wall Functions (SWF and NWF) use the law of the wall to supply boundary conditions for turbulent flows.
    - The near-wall mesh can be relatively coarse.
    - For equilibrium boundary layers and full-developed flows where log-law is valid.



- **Enhanced Wall Treatment Option**
  - Combines the use of blended law-of-the wall and a two-layer zonal model.
    - Suitable for low-*Re* flows or flows with complex near-wall phenomena.
    - Turbulence models are modified for the inner layer.
    - Generally requires a fine near-wall mesh capable of resolving the viscous sub-layer (more than 10 cells within the inner layer)



Placement of The First Grid Point

- For standard or non-equilibrium wall functions, each wall-adjacent cell's centroid should be located within the log-law layer: $y_p^+ \approx 30 - 300$
- For the enhanced wall treatment (EWT), each wall-adjacent cell's centroid should be located within the viscous sublayer: $y_p^+ \approx 1$
  - EWT can automatically accommodate cells placed in the log-law layer.
- How to estimate the size of wall-adjacent cells before creating the grid:
  - $y_p^+ \equiv y_p u_\tau / \nu \;\Rightarrow\; y_p \equiv y_p^+ \nu / u_\tau$, $\quad u_\tau \equiv \sqrt{\tau_w / \rho} = U_e \sqrt{\overline{c}_f / 2}$
  - The skin friction coefficient can be estimated from empirical correlations:

$$\text{Flat Plate: } \frac{\overline{C_f}}{2} \approx 0.037\ \mathrm{Re}_L^{-1/5} \qquad \text{Duct: } \frac{\overline{C_f}}{2} \approx 0.039\ \mathrm{Re}_D^{-1/4}$$

- Use post-processing (e.g., xy-plot or contour plot) to double check the near-wall grid placement after the flow pattern has been established.

Near-Wall Modeling: Recommended Strategy

- Use SWF or NWF for most high *Re* applications (Re > 10^6) for which you cannot afford to resolve the viscous sub-layer.
  - There is little gain from resolving the viscous sub-layer. The choice of core turbulence model is more important.
  - Use NWF for mildly separating, reattaching, or impinging flows.
- You may consider using EWT if:
  - The characteristic *Re* is low or if near wall characteristics need to be resolved.
  - The same or similar cases which ran successfully previously with the two-layer zonal model (in Fluent v5).
  - The physics and near-wall mesh of the case is such that y^+ is likely to vary significantly over a wide portion of the wall region.
- *Try to make the mesh either coarse or fine enough to avoid placing the wall-adjacent cells in the buffer layer ($y^+ = 5 \sim 30$).*

Boundary Conditions at Inlet and Outlet

- When turbulent flow enters a domain at inlets or outlets (backflow), boundary conditions for k, ε, ω and/or $\overline{u_i u_j}$ must be specified, depending on which turbulence model has been selected
- Four methods for directly or indirectly specifying turbulence parameters:
  - *Explicitly* input k, ε, ω, or $\overline{u_i u_j}$
    - This is the only method that allows for profile definition.
    - See user's guide for the correct *scaling* relationships among them.
  - Turbulence intensity and length scale
    - Length scale is related to size of large eddies that contain most of energy.
      - For boundary layer flows: $l \approx 0.4\delta_{99}$
      - For flows downstream of grid: $l \approx$ opening size
  - Turbulence intensity and hydraulic diameter
    - Ideally suited for internal (duct and pipe) flows
  - Turbulence intensity and turbulent viscosity ratio
    - For external flows: $1 < \mu_t/\mu < 10$
- Turbulence intensity depends on upstream conditions: $u/U \approx \sqrt{2k/3}/U < 20\%$

Stochastic Inlet Velocity Boundary Condition

- It is often important to specify realistic turbulent inflow velocity BC for accurate prediction of the downstream flow:

$$u_i(\mathbf{x},t) = \underbrace{U_i(\mathbf{x})}_{time-averaged} + \underbrace{u_i'(\mathbf{x},t)}_{coherent+random}$$

- The random-number based stochastic inlet BC in FLUENT v6.1 is superseded by two new methods in v6.2.
  - Spectral synthesizer
    - Able to synthesize anisotropic, inhomogeneous turbulence from RANS results (k-ε, k-ω, and RSM fields)
  - Vortex method
    - Turbulence is mimicked using the velocity-field induced by many quasi-random point-vortices on a plane. It uses turbulence data (e.g., intensity, k-ε, k-ω) as inputs
- Can be used for RANS/LES zonal hybrid approach

GUI for Turbulence Models

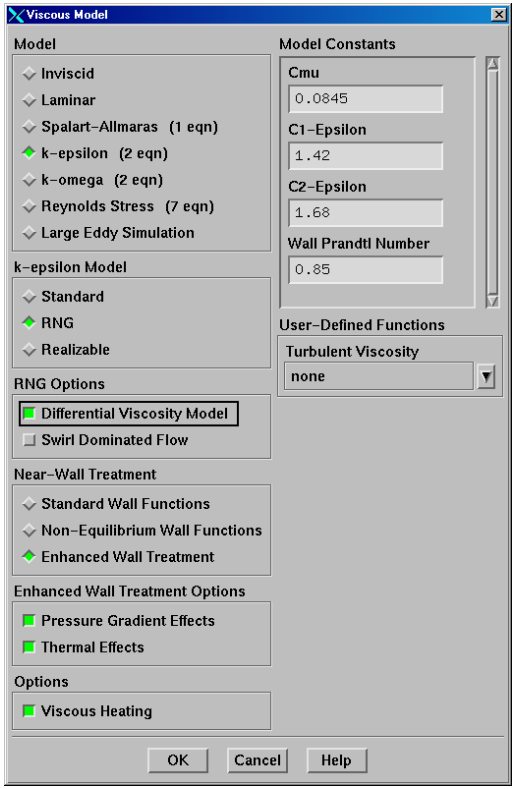
Define → Models → Viscous...

Inviscid, Laminar, or Turbulent

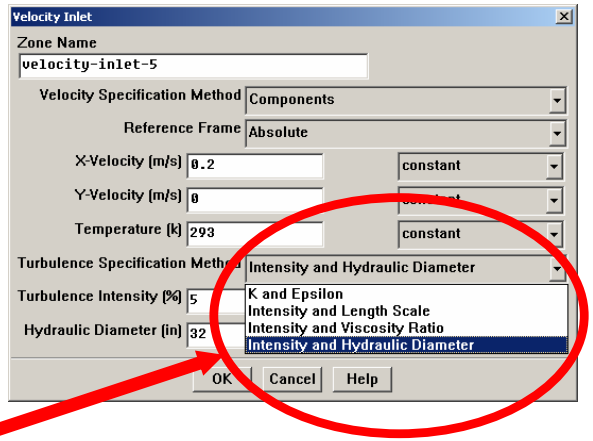
Turbulence Model Options

Near Wall Treatments

Additional Turbulence Options

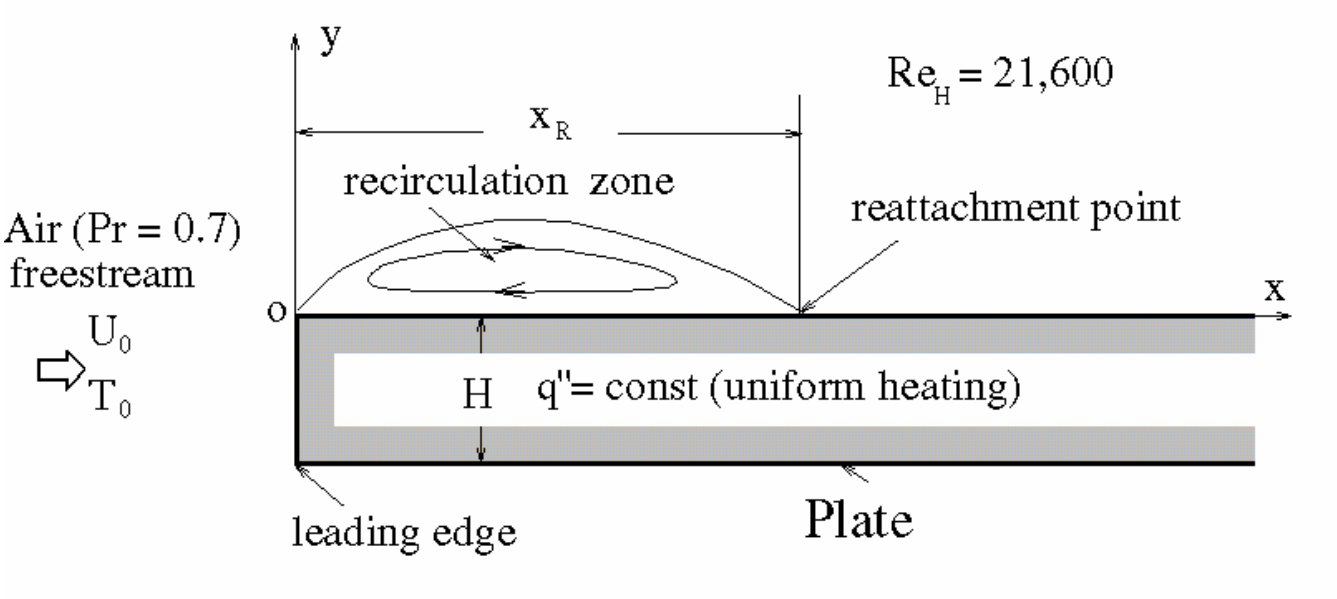


Define → Boundary Conditions ...



In the absence of available data, use more familiar quantities to specify boundary conditions at inlets and outlets

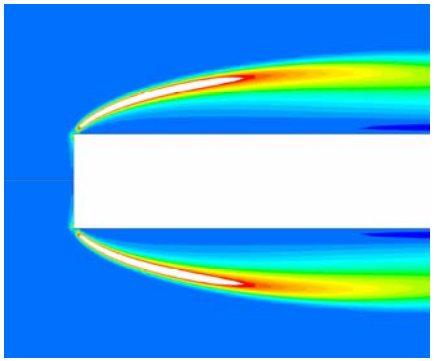
Example (1): Turbulent Flow Over a Blunt Plate

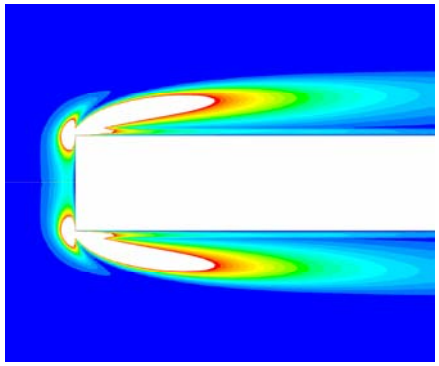


Reynolds-Stress model ("exact")

Standard k-ε model

Contour plots of turbulent kinetic energy (TKE)

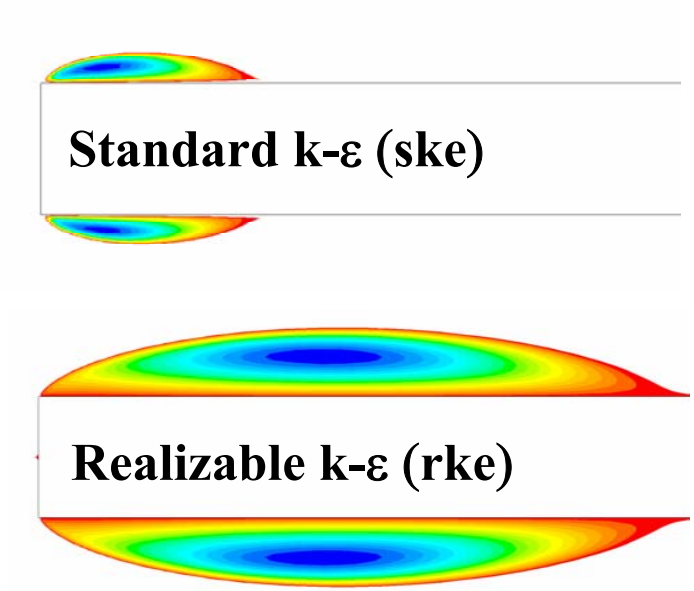




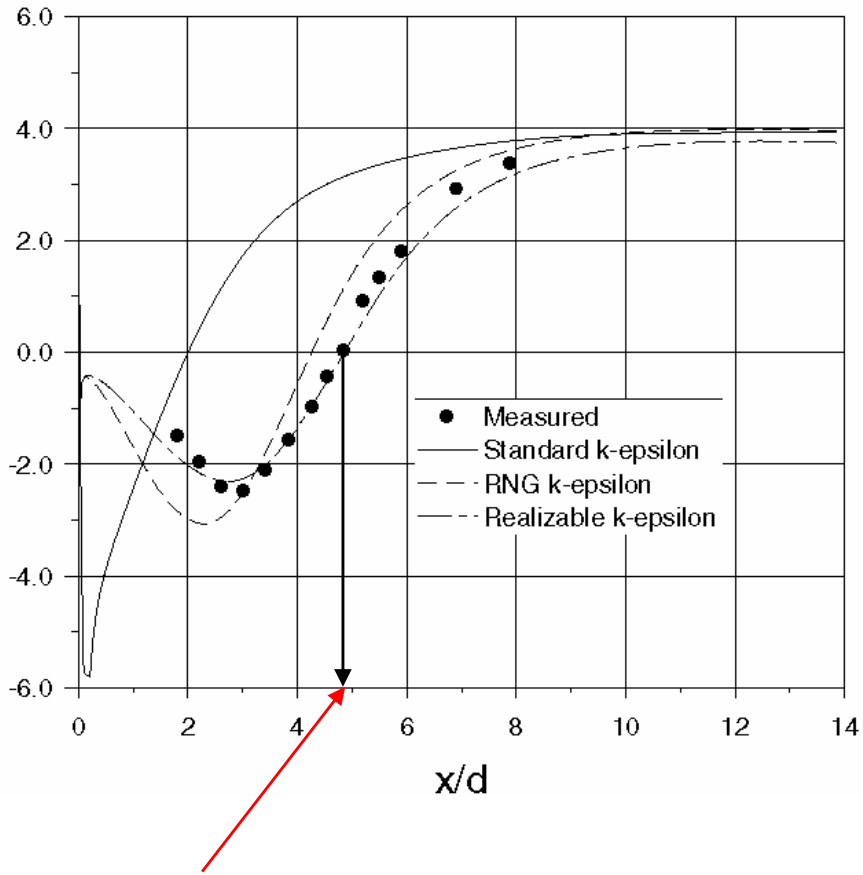
The Standard k-ε model is known to give spuriously large TKE on the font face of the plate

Example (1): Turbulent Flow over a Blunt Plate

Predicted separation bubble:



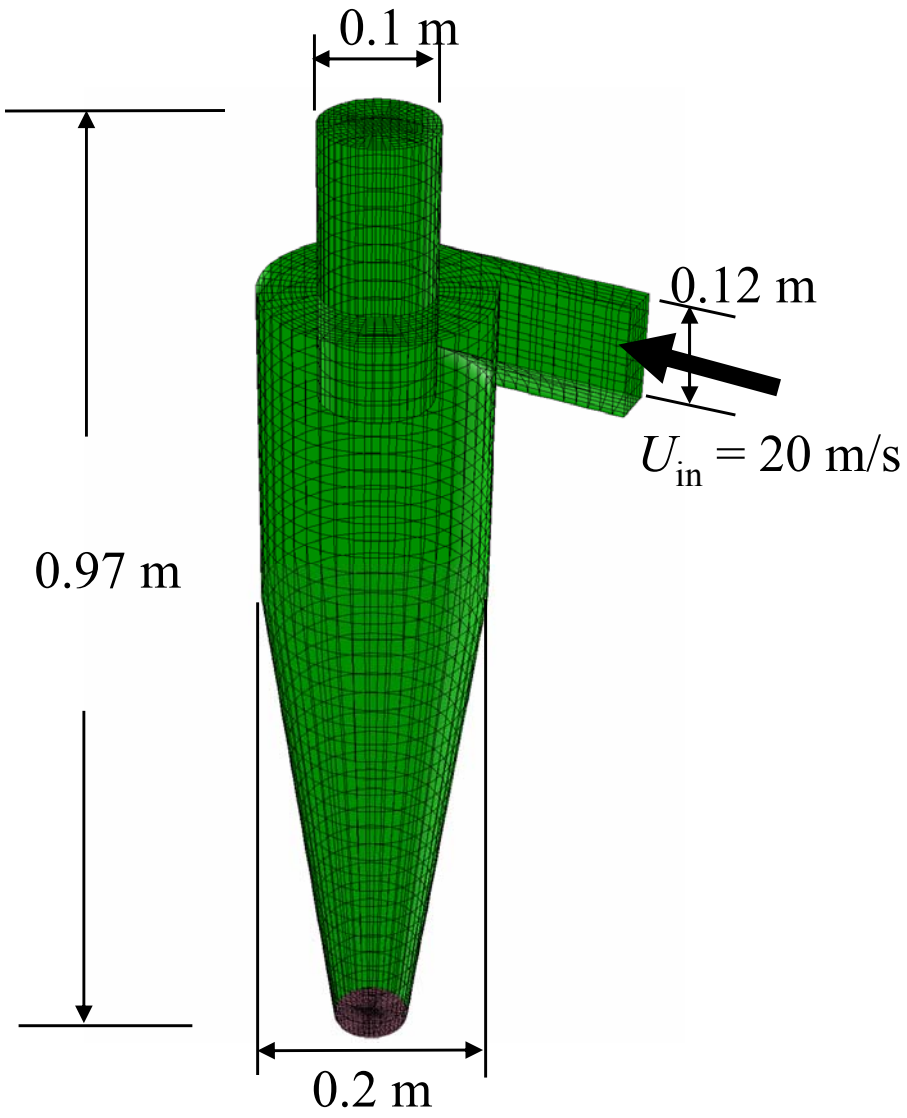
Skin Friction coefficient



ske model severely underpredicts the size of the separation bubble, while rke model predicts the size exactly.

Experimentally observed reattachment point is at x/d = 4.7

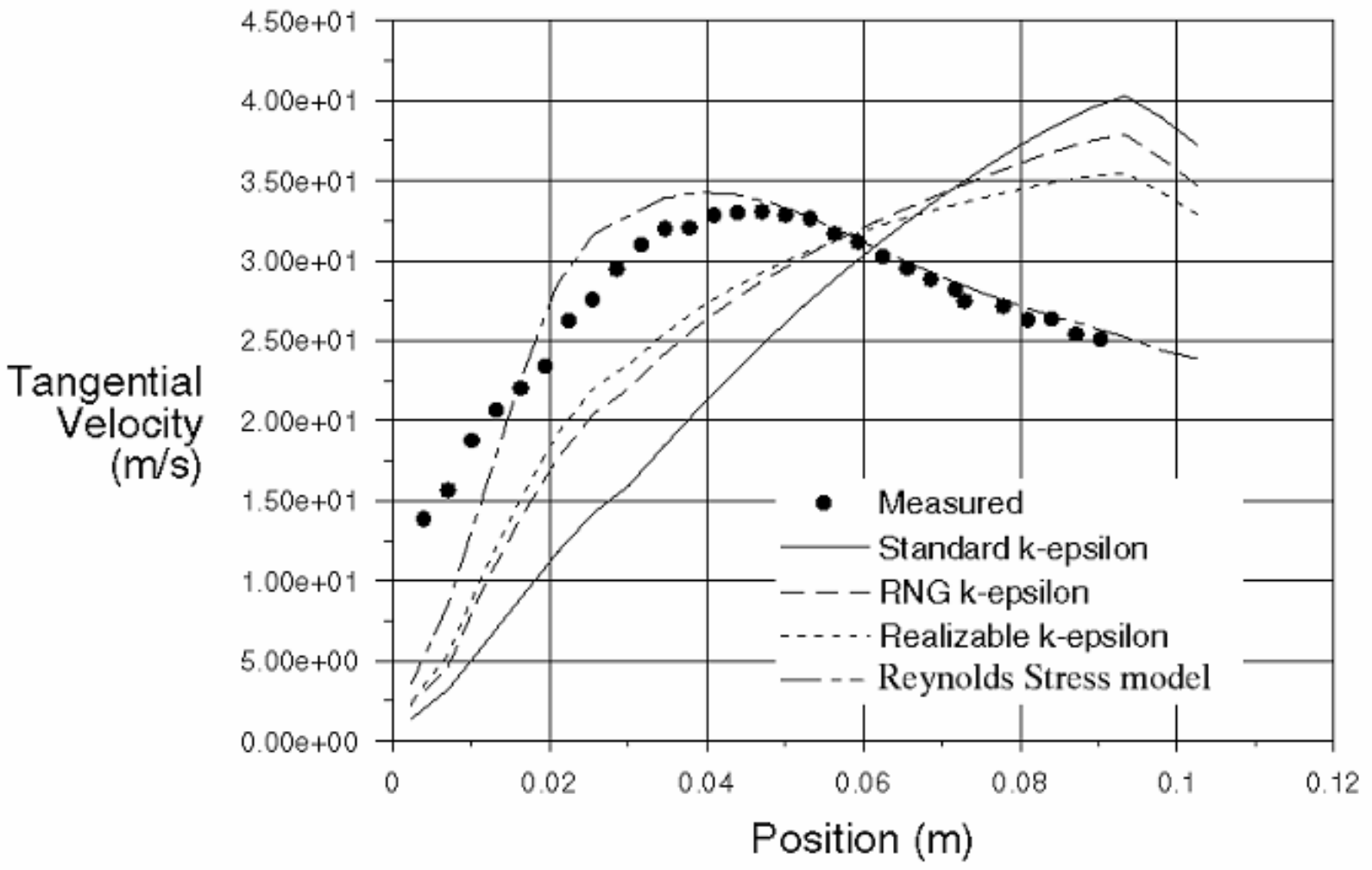
Example (2): Turbulent Flow in a Cyclone



- 40,000 cell hexahedral mesh
- High-order upwind scheme was used.
- Computed using SKE, RNG, RKE and RSM (second moment closure) models with the standard wall functions
- Represents highly swirling flows ($W_{max} = 1.8\ U_{in}$)

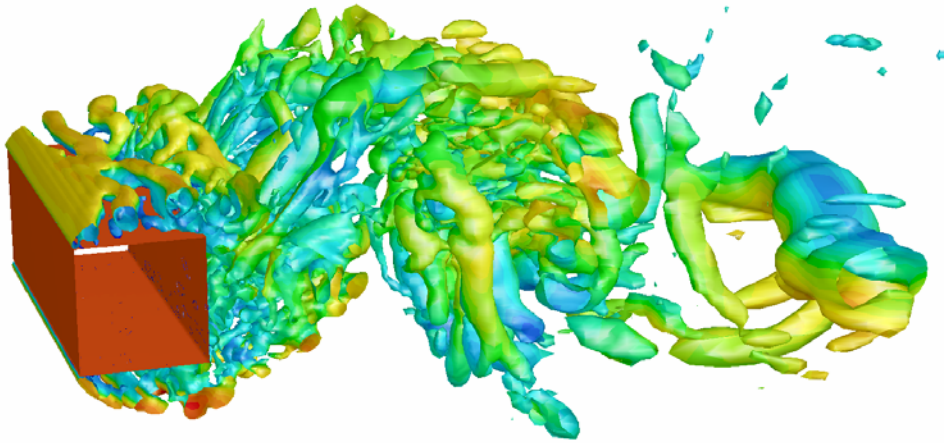
Example (2): Turbulent Flow in a Cyclone

- Tangential velocity profile predictions at 0.41 m below the vortex finder

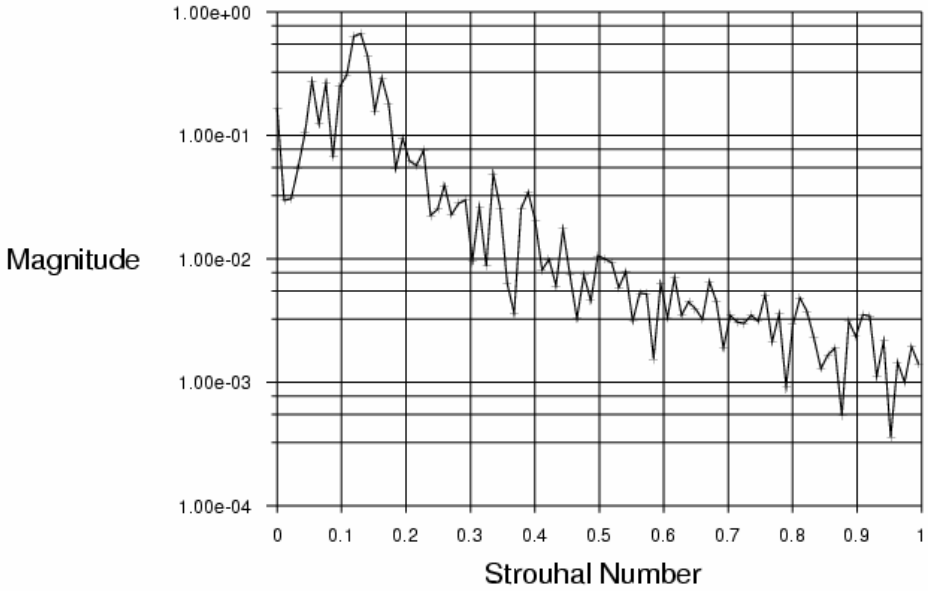


Example (3): LES of the Flow Past a Square Cylinder (Re_H = 22,000)

	C_D	St
Dynamic Smag.	2.28	0.130
Dynamic TKE	2.22	0.134
Exp.(Lyn *et al.*, 1992)	2.1 – 2.2	0.130

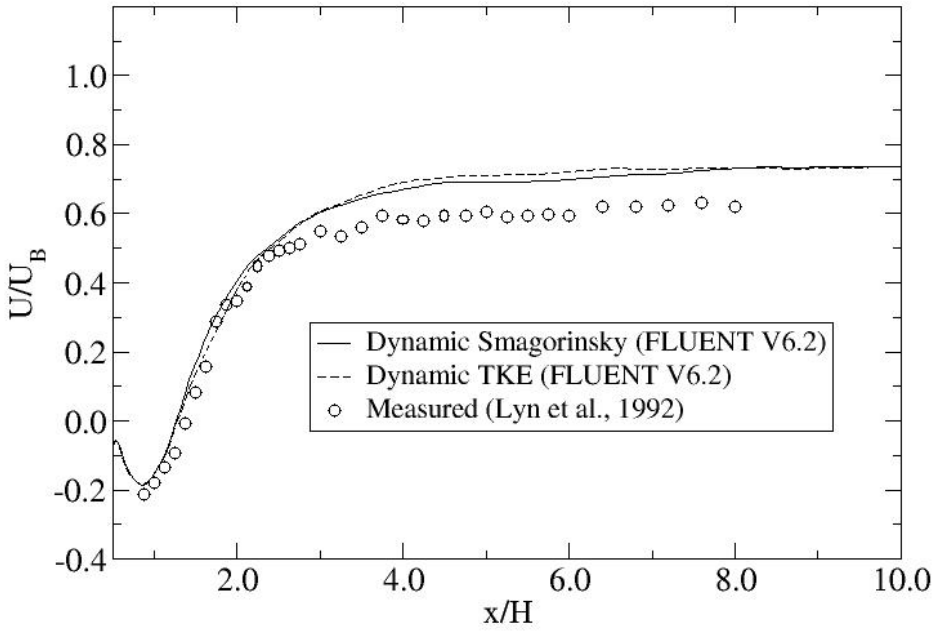


Iso-contours of instantaneous vorticity magnitude

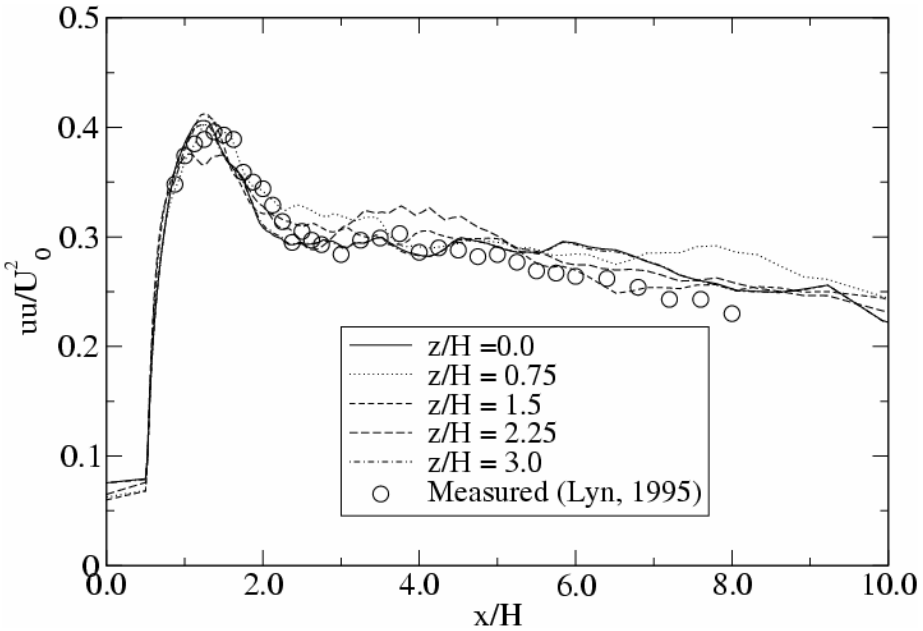


C_L spectrum

Example (3): LES of the Flow Past a Square Cylinder (Re_H = 22,000)



Streamwise mean velocity along the wake centerline



Streamwise normal stress along the wake centerline

Summary: Turbulence Modeling Guidelines

- Successful turbulence modeling requires engineering judgment of:
  - Flow physics
  - Computer resources available
  - Project requirements
    - Accuracy
    - Turnaround time
  - Near-wall treatments
- Modeling Procedure
  - Calculate characteristic *Re* and determine whether the flow is turbulent.
  - Estimate wall-adjacent cell centroid y^+ before generating the mesh.
  - Begin with SKE (standard k-ε) and change to RKE, RNG, SKO, SST or V2F if needed. Check the tables in the appendix as a starting guide.
  - Use RSM for highly swirling, 3-D, rotating flows.
  - Use wall functions for wall boundary conditions except for the low-*Re* flows and/or flows with complex near-wall physics.

Appendix

- Summary of RANS Turbulence Models: Description, Model Behavior and Usage
- More Details on Near-wall Modeling
- Turbulent Heat Transfer Modeling
- Additional Information on Menter's SST k-ω Model
- V2F Turbulence Model
- Initial Velocity Field for LES/DES

RANS Turbulence Model Descriptions

Model	Description:
Spalart-Allmaras	A single transport equation model solving directly for a modified turbulent viscosity. Designed specifically for aerospace applications involving wall-bounded flows on a fine, near-wall mesh. Fluent's implementation allows use of coarser meshes. •Option to include strain rate in k production term improves predictions of vortical flows.
Standard k-ε	The baseline two transport equation model solving for k and ε. This is the default k-ε model. Coefficients are empirically derived; valid for fully turbulent flows only. •Options to account for viscous heating, buoyancy, and compressibility are shared with other k-ε models.
RNG k-ε	A variant of the standard k-ε model. Equations and coefficients are analytically derived. Significant changes in the ε equation improves the ability to model highly strained flows. •Additional options aid in predicting swirling and low Re flows.
Realizable k-ε	A variant of the standard k-ε model. Its 'realizability' stems from changes that allow certain mathematical constraints to be obeyed which ultimately improves the performance of this model.
Standard k-ω	A two transport equation model solving for k and ω, the specific dissipation rate (ε/k) based on Wilcox (1998). This is the default k -ω model. Demonstrates superior performance for wall bounded and low-Re flows. Shows potential for predicting transition. •Options account for transitional, free shear, and compressible flows.
SST k-ω	A variant of the standard k-ω model. Combines the original Wilcox model (1988) for use near walls and standard k-ε model away from walls using a blending function. Also limits turbulent viscosity to guarantee that $\tau_t \sim k$. •The transition and shearing options borrowed from SKO. No compressibility option.
RSM	Reynolds stresses are solved directly with transport equations avoiding isotropic viscosity assumption of other models. Use for highly swirling flows. •Quadratic pressure-strain option improves performance for many basic shear flows.

RANS Turbulence Model Behavior and Usage

Model	Behavior and Usage
Spalart-Allmaras	Economical for large meshes. Performs poorly for 3D flows, free shear flows, flows with strong separation. Suitable for mildly complex (quasi-2D) external/internal flows and b.l. flows under pressure gradient (e.g. airfoils, wings, airplane fuselage, missiles, ship hulls).
Standard *k*-ε	Robust. Widely used despite the known limitations of the model. Performs poorly for complex flows involving severe ∇p, separation, strong stream line curvature. Suitable for initial iterations, initial screening of alternative designs, and parametric studies.
RNG *k*-ε	Suitable for complex shear flows involving rapid strain, moderate swirl, vortices, and locally transitional flows (e.g., b.l. separation, massive separation and vortex-shedding behind bluff bodies, stall in wide-angle diffusers, room ventilation)
Realizable *k*-ε	Offers largely the same benefits and has similar applications as RNG. Possibly more accurate and easier to converge than RNG.
Standard *k*-ω	Superior performance for wall-bounded b.l., free shear, and low Re flows. Suitable for complex boundary layer flows under adverse pressure gradient and separation (external aerodynamics and turbomachinery). Can be used for transitional flows (though tends to predict early transition). Separation is typically predicted to be excessive and early.
SST *k*-ω	Similar benefits as SKO. Dependency on wall distance makes this less suitable for free shear flows.
RSM	Physically the most sound RANS model. Avoids isotropic eddy viscosity assumption. More CPU time and memory required. Tougher to converge due to close coupling of equations. Suitable for complex 3D flows with strong streamline curvature, strong swirl/rotation (e.g. curved duct, rotating flow passages, swirl combustors with very large inlet swirl, cyclones).

Near-wall modeling (1):
Standard and Non-Equilibrium Wall Functions

- **Standard Wall Function**
  - Momentum boundary condition based on Launder-Spaulding law-of-the-wall:

$$U^* = y^* \quad \text{for} \quad y^* < y_v^*$$
$$U^* = \frac{1}{\kappa}\ln\left(Ey^*\right) \quad \text{for} \quad y^* > y_v^*$$

where $U^* \equiv \frac{U_P C_\mu^{1/4} k_P^{1/2}}{\tau_w/\rho}$ $\quad y^* \equiv \frac{\rho C_\mu^{1/4} k_P^{1/2} y_P}{\mu}$

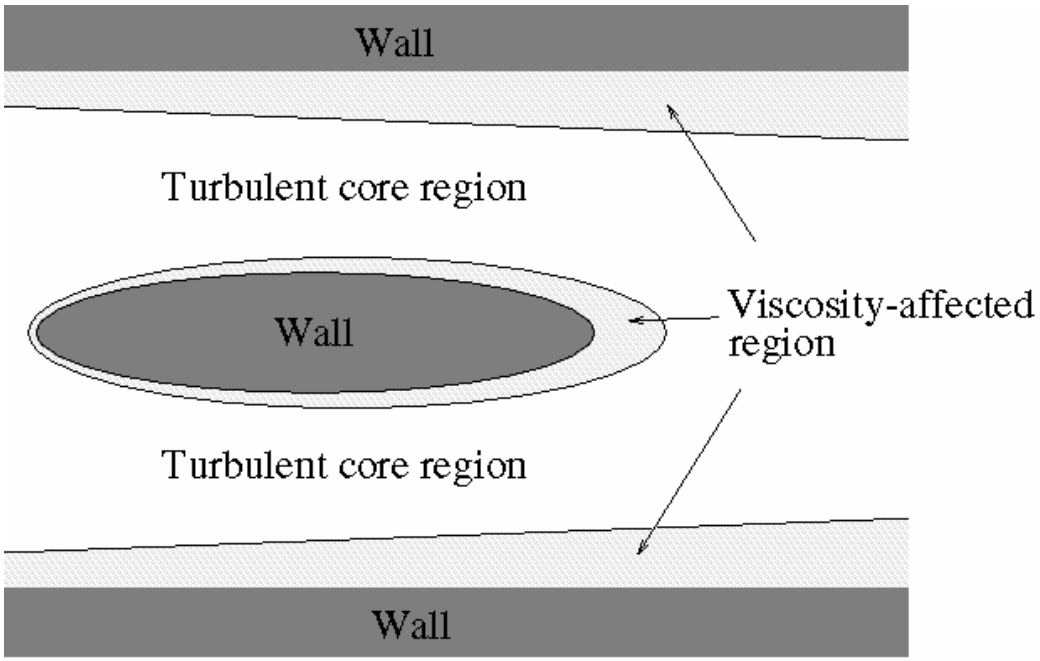
  - Similar 'wall laws' apply for energy and species.
  - Additional formulas account for k, ε, and $\rho\overline{u_i u_j}$.
  - Less reliable when flow departs from conditions assumed in their derivation.
    - Severe ∇p or highly non-equilibrium near-wall flows, high transpiration or body forces, low Re or highly 3D flows
- **Non-Equilibrium Wall Function**
  - SWF is modified to account for stronger ∇p and non-equilibrium flows.
    - Useful for mildly separating, reattaching, or impinging flows.
    - Less reliable for high transpiration or body forces, low Re or highly 3D flows.
- The Standard and Non-Equilibrium Wall functions are options for the k-ε and RSM turbulence models.

Near-wall modeling (2):
Enhanced Wall Treatment

- **Enhanced Wall Treatment**
  - *Enhanced wall functions*
    - Momentum boundary condition based on *blended* law-of-the-wall (Kader). $u^+ = e^{\Gamma} u^+_{lam} + e^{1/\Gamma} u^+_{turb}$
    - Similar blended 'wall laws' apply for energy, species, and ω.
    - Kader's form for blending allows for incorporation of additional physics.
      - Pressure gradient effects
      - Thermal (including compressibility) effects
  - *Two-layer zonal model*
    - A blended two-layer model is used to determine near-wall ε field.
      - Domain is divided into viscosity-affected (near-wall) region and turbulent core region.
        - Based on 'wall-distance' turbulent Reynolds number: $Re_y \equiv \rho\sqrt{k}\,y/\mu$
        - Zoning is dynamic and solution adaptive.
      - High *Re* turbulence model used in outer layer.
      - 'Simple' turbulence model used in inner layer.
    - Solutions for ε and μ_t in each region are blended, e.g., $\lambda_\varepsilon (\mu_t)_{\text{outer}} + (1-\lambda_\varepsilon)(\mu_t)_{\text{inner}}$
- The Enhanced Wall Treatment option is available for the *k*-ε and RSM turbulence models (EWT is the sole treatment for Spalart Allmaras and *k*-*ω* models).

Near Wall Modeling(3):
Two-Layer Zones

- The two regions are demarcated on a cell-by-cell basis:
  - $Re_y > 200$
    - turbulent core region
  - $Re_y < 200$
    - viscosity affected region
  - $Re_y = \rho k^{1/2} y/\mu$
  - y is shortest distance to nearest wall
  - zoning is dynamic and solution adaptive



Turbulent Heat Transfer

- The Reynolds averaging of the energy equation produces an additional term $\overline{u_i' t'}$
  - Analogous to the Reynolds stresses, this is the turbulent heat flux term. An isotropic turbulent diffusivity is assumed:

$$\overline{u_i' t'} = -\nu_T \frac{\partial T}{\partial x_i}$$

  - Turbulent diffusivity is usually related to eddy viscosity via a turbulent Prandtl number (modifiable by the users):

$$\mathrm{Pr_t} = \nu_t / \nu_T \approx 0.85 - 0.9$$

- Similar treatment is applicable to other turbulent scalar transport equations.

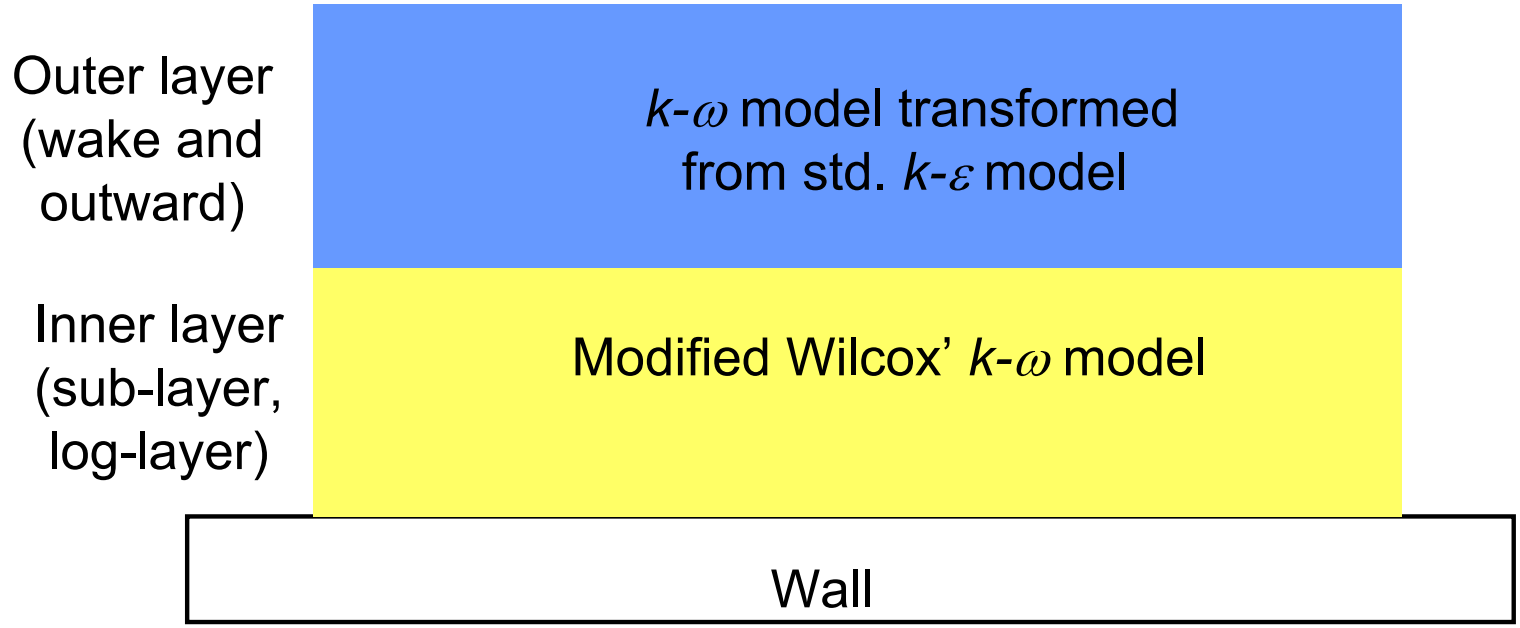
Menter's SST k-ω Model *Background*

Many people, including Menter (1994), have noted that:

- k-ω model has many good attributes and perform much better than k-ε models for *boundary layer flows*.
- Wilcox' original k-ω model is overly sensitive to the freestream value (BC) of ω, while k-ε model is not prone to such problem.
- Most two-equation models, including k-ε models, over-predict turbulent stresses in the wake (velocity-defect) region, which leads to poor performance of the models for boundary layers under adverse pressure gradient and separated flows.

Menter's SST k-ω Model *Main Components*

- The SST k-ω model consists of
  - Zonal (blended) k-ω / k-ε equations (to address item 1 and 2 in the previous slide)
  - Clipping of turbulent viscosity so that turbulent stress stay within what is dictated by the structural similarity constant. (Bradshaw, 1967) - addresses item 3 in the previous slide



Menter's SST *k*-ω Model *Blended k-ω Equations*

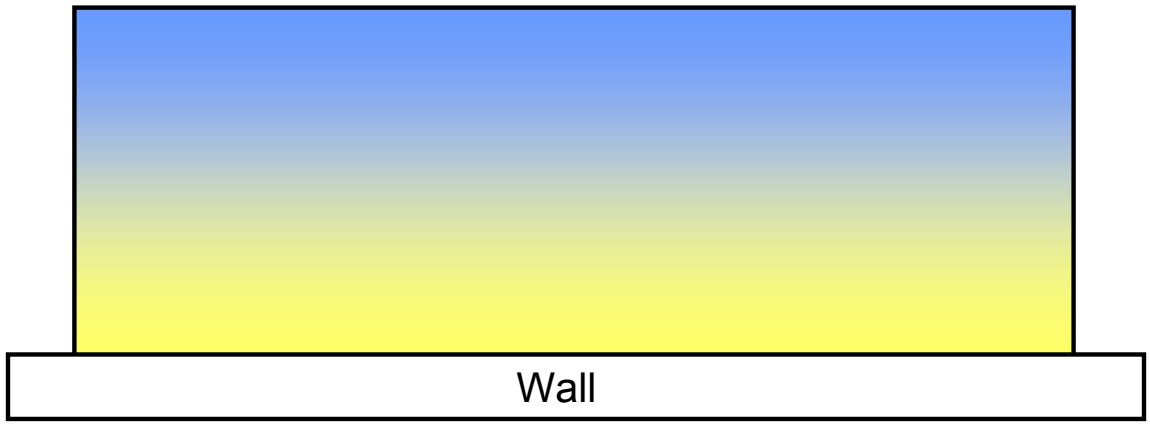
- The resulting blended equations are:

$$\rho\frac{Dk}{Dt}=\tau_{ij}\frac{\partial U_i}{\partial x_j}-\beta^* k\rho\omega+\frac{\partial}{\partial x_j}\left[\left(\mu+\frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right]$$

$$\rho\frac{D\omega}{Dt}=\frac{\gamma}{\nu_t}\tau_{ij}\frac{\partial U_i}{\partial x_j}-\beta\rho\omega^2+\frac{\partial}{\partial x_j}\left[\left(\mu+\frac{\mu_t}{\sigma_\omega}\right)\frac{\partial \omega}{\partial x_j}\right]$$

$$+2\rho(1-F_1)\sigma_{\omega 2}\frac{1}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j}$$

$$\phi=F_1\phi_1+(1-F_1)\phi_1\,,\qquad \phi=\beta\,,\sigma_k\,,\sigma_\omega\,,\gamma$$



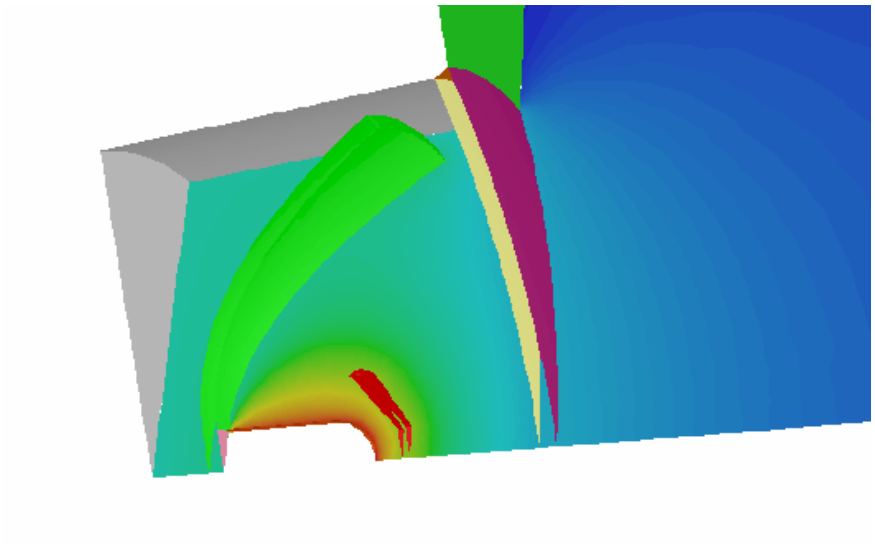
V2F Turbulence Model

- A model developed by Paul Durbin's group at Stanford:
  - Durbin suggests that the wall-normal fluctuations $\overline{v'^2}$ are responsible for the near-wall damping of the eddy viscosity
  - Requires two additional transport equations for $\overline{v'^2}$ and a relaxation function f to be solved together with k and ε
  - Eddy viscosity model is $\nu_t \sim \overline{v'^2}\, T$ instead of $\nu_t \sim k\, T$
- Promising results for many 3-D low-Re boundary-layer flows. For example, excellent predictions for heat transfer in jet impingement and separated flows, where k-ε models fail miserably.
- But it is a member of the EVM---same limitations still apply.
- V2F is an embedded add-on functionality in Fluent 6.x which requires a separate license from Cascade Technologies (www.turbulentflow.com)

Initial Velocity Field for LES/DES

- Initial condition for velocity field does not affect statistically stationary solutions
- However, starting LES with a realistic turbulent velocity field can substantially shorten the simulation time to get to statistically stationary state
- The spectral synthesizer can be used to superimpose turbulent velocity on top of the mean velocity field
  - Uses steady-state RANS (k-ε, k-ω, RSM, etc.) solutions as inputs to the spectral synthesizer
  - Accessible via a TUI command:
    `/solve/initialize/init-instantaneous-vel`

Heat Transfer Modeling



Headlamp modeled with
Discrete Ordinates
Radiation Model

Outline

- Introduction
- Conjugate Heat Transfer
- Natural Convection
- Radiation
- Periodic Heat Transfer

Introduction

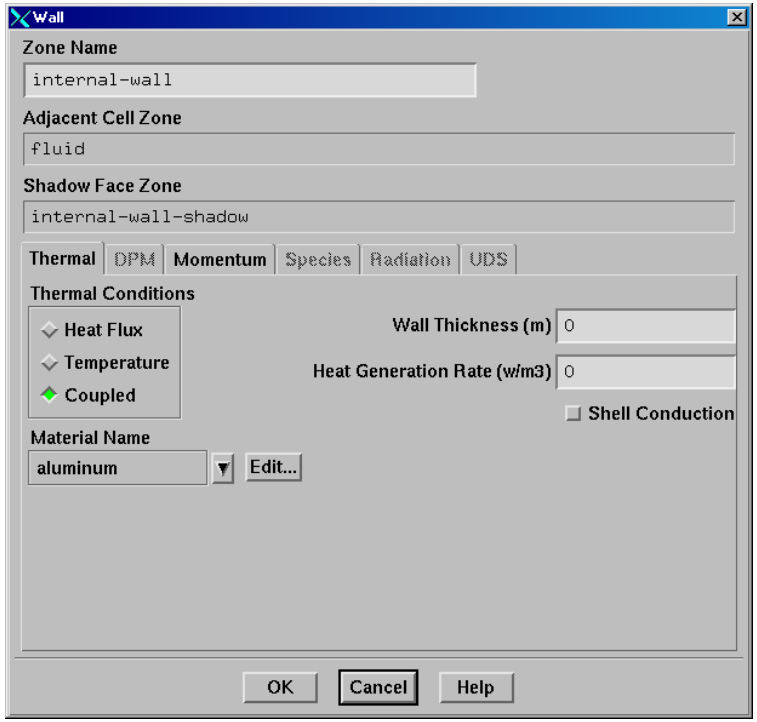
- Energy transport equation:

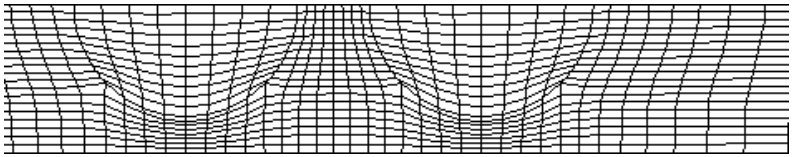
$$\frac{\partial}{\partial t}(\rho E)+\nabla\cdot\left(\vec{V}(\rho E+p)\right)=\nabla\cdot\left(k_{eff}\nabla T-\sum_j h_j J_j+(\bar{\bar{\tau}}_{eff}\cdot\vec{V})\right)+S_h$$

  - Energy source due to chemical reaction is included for reacting flows.
  - Energy source due to species diffusion included for multiple species flows.
    - Always included in coupled solver; can be disabled in segregated solver.
  - Energy source due to viscous heating:
    - Describes thermal energy created by viscous shear in the flow.
      - Important when shear stress in fluid is large (e.g., lubrication) and/or in high-velocity, compressible flows.
    - Often negligible
      - not included by default for segregated solver; always included for coupled solver.
  - In solid regions, simple conduction equation solved.
    - Convective term can also be included for moving solids.

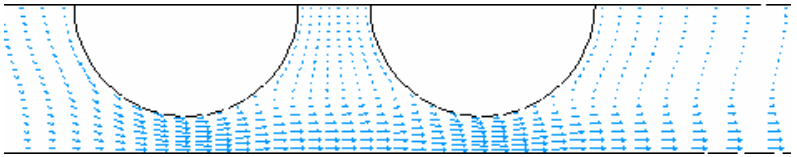
Conjugate Heat Transfer

- Ability to compute conduction of heat through solids, coupled with convective heat transfer in fluid.
- Coupled Boundary Condition:
  - available to wall zone that separates two cell zones.

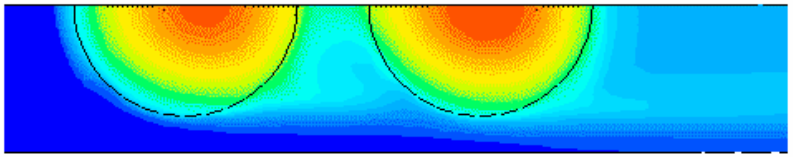




Grid



Velocity vectors

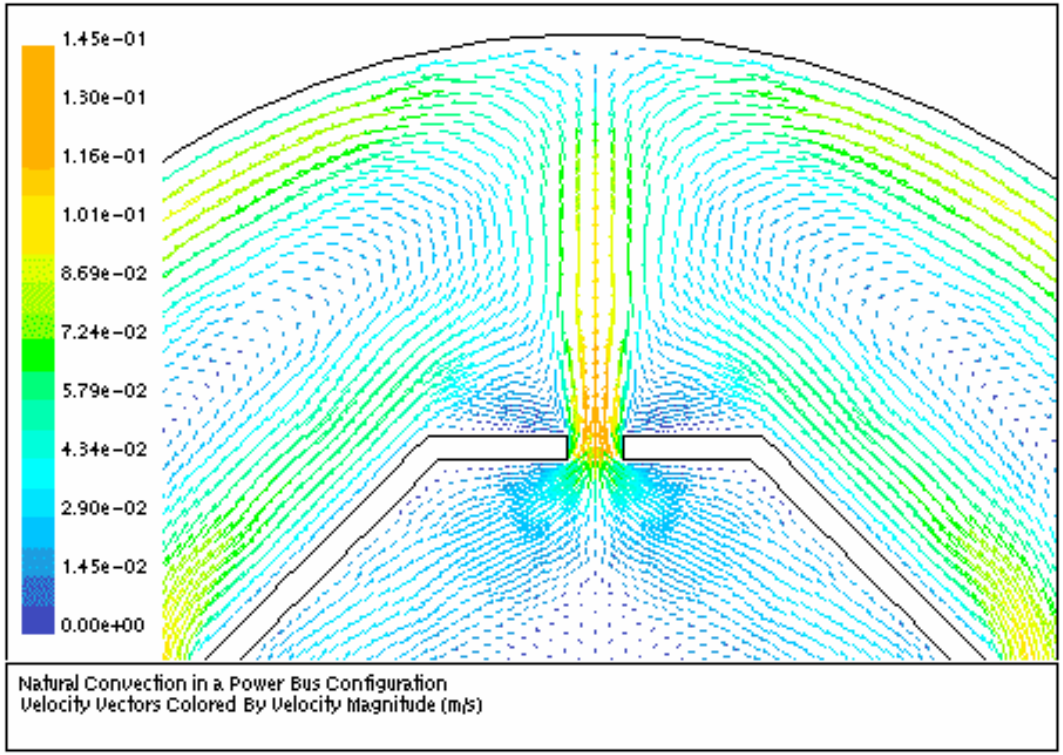


Temperature contours

Example: Cooling flow over fuel rods

Natural Convection - Introduction

- Natural convection occurs when heat is added to fluid and fluid density varies with temperature.
- Flow is induced by force of gravity acting on density variation.
- When gravity term is included, pressure gradient and body force term in the momentum equation are re-written as:



$$-\frac{\partial p}{\partial x}+\rho g \Rightarrow -\frac{\partial p'}{\partial x}+(\rho-\rho_o)g$$

where $p' = p - \rho_o g x$

- This format avoids potential roundoff error when gravitational body force term is included.

Natural Convection – the Boussinesq Model

- Boussinesq model assumes the fluid density is uniform
  - *Except* for the body force term in the momentum equation along the direction of gravity, we have:

$$(\rho - \rho_0)g = -\rho_0 \beta (T - T_0)g$$

  - Valid when density variations are small (i.e., small variations in T).
- It provides faster convergence for many natural-convection flows than by using fluid density as function of temperature.
  - Constant density assumptions reduces non-linearity.
  - Suitable when density variations are small.
  - Cannot be used together with species transport or reacting flows.
- Natural convection problems inside closed domains:
  - For steady-state solver, Boussinesq model must be used.
    - The constant density, ρ_o, properly specifies the mass of the domain.
  - For unsteady solver, Boussinesq model or ideal-gas law can be used.
    - Initial conditions define mass in the domain.

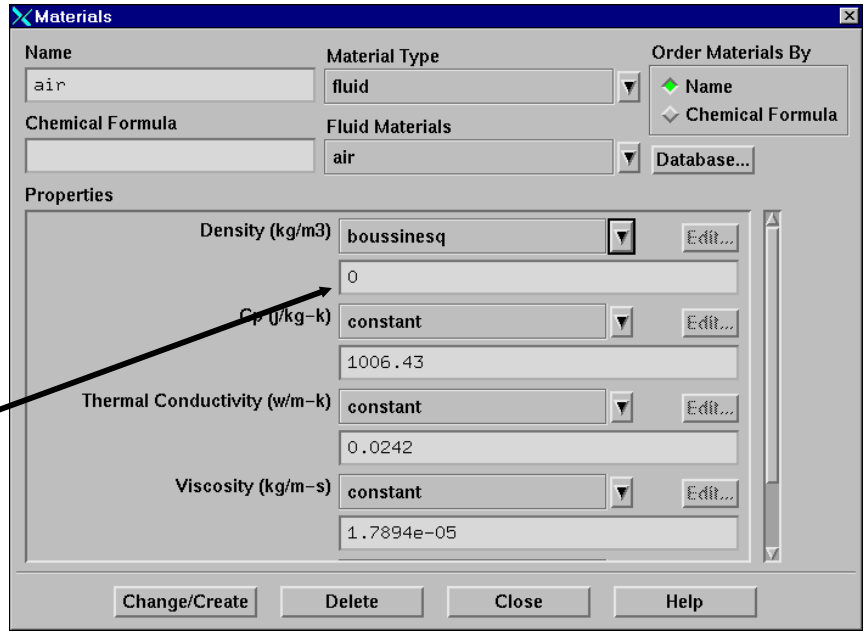
User Inputs for Natural Convection

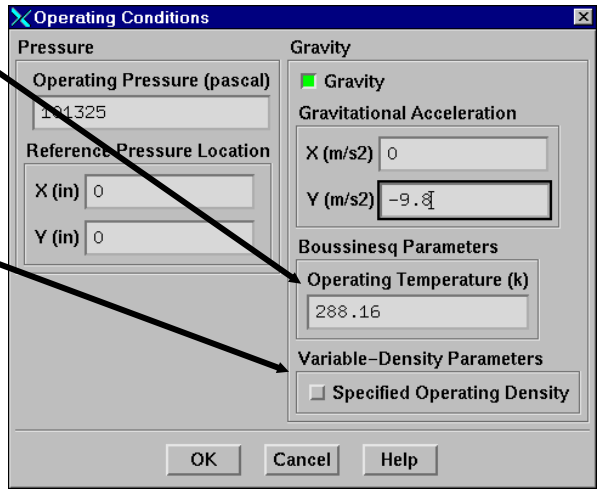
1. Set gravitational acceleration.

   Define → Operating Conditions...

2. Define density model.
   - If using Boussinesq model:
     - Select Boussinesq as the Density method and assign constant value, ρ_o.

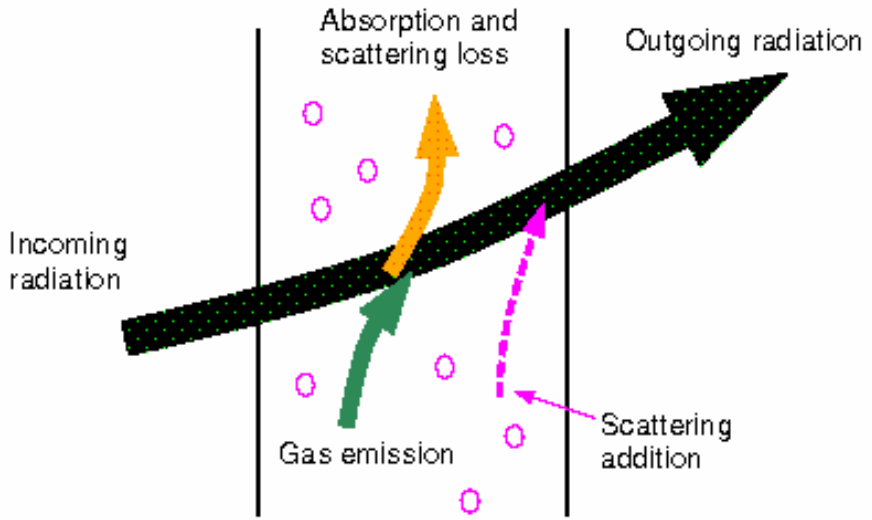
       Define → Materials...
     - Set Thermal Expansion Coefficient, β.
     - Set Operating Temperature, T_o.
   - If using temperature dependent model, (e.g., ideal gas or polynomial):
     - Specify Operating Density or,
     - Allow Fluent to calculate ρ_o from a cell average (default, every iteration).





Radiation

- Radiation effects should be accounted for when $Q_{rad} = \sigma(T_{\max}^4 - T_{\min}^4)$ is of equal or greater magnitude than that of convective and conductive heat transfer rates.
- To account for radiation, radiative intensity transport equations (RTEs) are solved.
  - Local absorption by fluid and at boundaries links RTEs with energy equation.
- Radiation intensity, $I(\mathbf{r},s)$, is directionally and spatially dependent.
- Intensity, $I(\mathbf{r},s)$, along any direction can be modified by:
  - Local absorption
  - Out-scattering (scattering away from the direction)
  - Local emission
  - In-scattering (scattering into the direction)
- Five radiation models are provided:
  - Discrete Ordinates Model (DOM)
  - Discrete Transfer Radiation Model (DTRM)
  - P-1 Radiation Model
  - Rosseland Model
  - Surface-to-Surface (S2S)



Discrete Ordinates Model

- The radiative transfer equation is solved for a discrete number of finite solid angles, s_i:

$$\frac{\partial I_{si}}{\partial x_i} + \underbrace{(a+\sigma_s)I(r,s)}_{\text{absorption}} = \underbrace{an^2\frac{\sigma T^4}{\pi}}_{\text{emission}} + \underbrace{\frac{\sigma_s}{4\pi}\int_0^{4\pi} I(r,s')\Phi(s\cdot s')d\Omega'}_{\text{scattering}}$$

- Advantages:
  - Conservative method leads to heat balance for coarse discretization.
    - Accuracy can be increased by using a finer discretization.
  - Most comprehensive radiation model:
    - Accounts for scattering, semi-transparent media, specular surfaces, and wavelength-dependent transmission using banded-gray option.
- Limitations:
  - Solving a problem with a large number of ordinates is CPU-intensive.

Discrete Transfer Radiation Model (DTRM)

- Main assumption: radiation leaving surface element in a specific range of solid angles can be approximated by a single ray.
- Uses ray-tracing technique to integrate radiant intensity along each ray:

$$\frac{dI}{ds} + aI = \frac{a\sigma T^4}{\pi}$$

- Advantages:
  - Relatively simple model.
  - Can increase accuracy by increasing number of rays.
  - Applies to wide range of optical thicknesses.
- Limitations:
  - Assumes all surfaces are diffuse.
  - Effect of scattering not included.
  - Solving a problem with a large number of rays is CPU-intensive.

P-1 Model

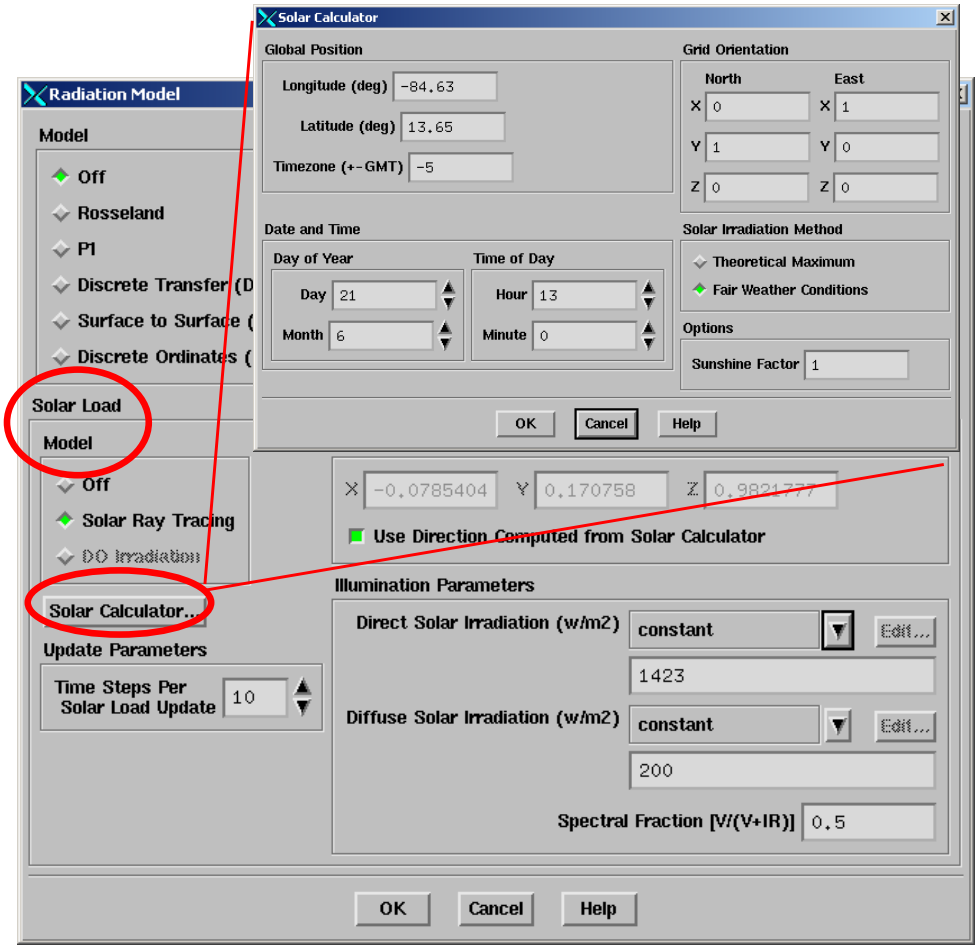
- Main assumption: Directional dependence in RTE is integrated out, resulting in a diffusion equation for incident radiation.
- Advantages:
  - Radiative transfer equation easy to solve with little CPU demand.
  - Includes effect of scattering.
    - Effects of particles, droplets, and soot can be included.
  - Works reasonably well for combustion applications where optical thickness is large.
- Limitations:
  - Assumes all surfaces are diffuse.
  - May result in loss of accuracy, depending on complexity of geometry, if optical thickness is small.
  - Tends to overpredict radiative fluxes from localized heat sources or sinks.

Surface-to-Surface Radiation Model

- The S2S radiation model can be used for modeling enclosure radiative transfer without participating media.
  - e.g., spacecraft heat rejection system, solar collector systems, radiative space heaters, and automotive underhood cooling
  - View-factor based model
  - Non-participating media is assumed.
- Limitations:
  - The S2S model assumes that all surfaces are diffuse.
  - The implementation assumes gray radiation.
  - Storage and memory requirements increase *very* rapidly as the number of surface faces increases.
    - Memory requirements can be reduced by using clusters of surface faces.
      - Clustering does not work with sliding meshes or hanging nodes.
  - Not to be used with periodic or symmetry boundary conditions.
  - Cannot be used for models with multiple enclosures geometry.

Solar Load Model

- Solar load model
  - Ray tracing algorithm for solar radiant energy transport: Compatible with all radiation models
  - Available with parallel solver (but ray tracing algorithm is not parallelized)
  - 3D only
- Specifications
  - Sun direction vector
  - Solar intensity (direct, diffuse)
  - Solar calculator for calculating direction and direct intensity using theoretical maximum or "fair weather conditions"
  - Transient cases
    - When direction vector is specified with solar calculator, sun direction vector will change accordingly in transient simulation
    - Specify "time steps per solar load update"

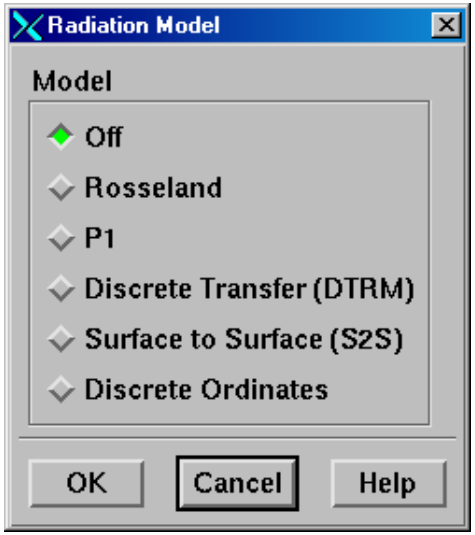


Choosing a Radiation Model

- For certain problems, one radiation model may be more appropriate in general.

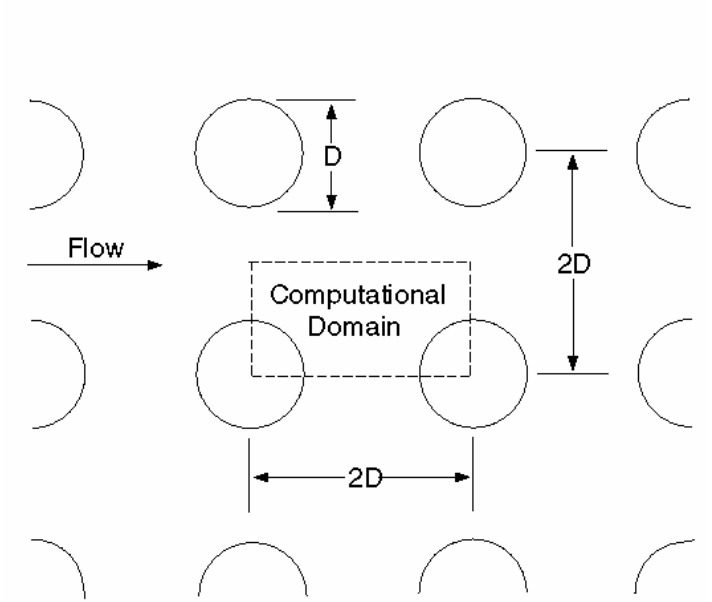
  Define → Models → Radiation...

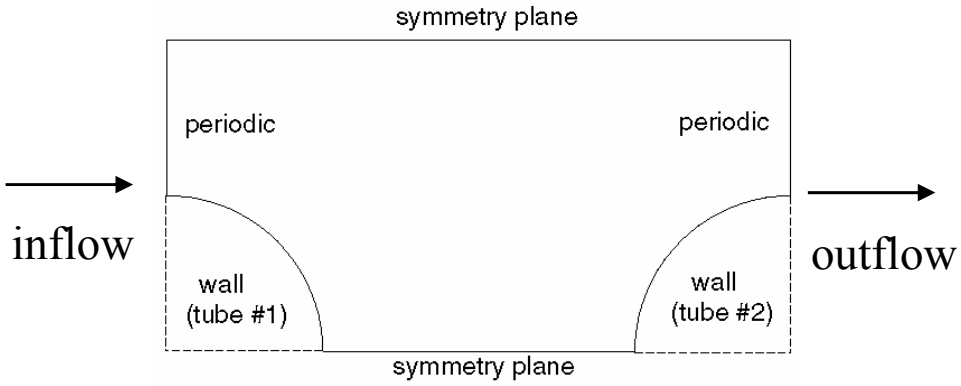
  - Computational effort: P-1 gives reasonable accuracy with less effort.
  - Accuracy: DTRM and DOM more accurate.
  - Optical thickness: DTRM/DOM for optically thin media (optical thickness << 1); P-1 better for optically thick media.
  - Scattering: P-1 and DOM account for scattering.
  - Particulate effects: P-1 and DOM account for radiation exchange between gas and particulates.
  - Localized heat sources: DTRM/DOM with sufficiently large number of rays/ordinates is more appropriate.



Periodic Heat Transfer (1)

- Also known as streamwise-periodic or fully-developed flow.
- Used when flow and heat transfer patterns are repeated, e.g.,
  - Compact heat exchangers
  - Flow across tube banks
- Geometry and boundary conditions repeat in the streamwise direction.





Outflow at one periodic boundary is inflow at the other

Periodic Heat Transfer (2)

- Temperature (and pressure) vary in the streamwise direction.
- Scaled temperature (and periodic pressure) is same at periodic boundaries.
- For fixed wall temperature problems, scaled temperature defined as:

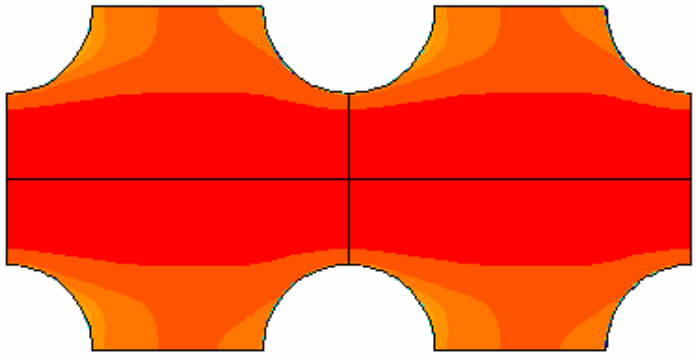
$$\theta = \frac{T - T_{wall}}{T_b - T_{wall}}$$

T_b = suitably defined bulk temperature

- Can also model flows with specified wall heat flux.

Periodic Heat Transfer (3)

- Periodic heat transfer is subject to the following constraints:
  - Either constant temperature or fixed flux bounds.
  - Conducting regions cannot straddle periodic plane.
  - Thermodynamic and transport properties cannot be functions of temperature.
  - Viscous heating and volumetric heat sources cannot be used with constant wall temperature boundary conditions.



Contours of Scaled Temperature

Summary

- Heat transfer modeling is available in all Fluent solvers.
- After activating heat transfer, you must provide:
  - Thermal conditions at walls and flow boundaries
  - Fluid properties for energy equation
- Available heat transfer modeling options include:
  - Species diffusion heat source
  - Combustion heat source
  - Conjugate heat transfer
  - Natural convection
  - Radiation
  - Periodic heat transfer

User Defined Functions

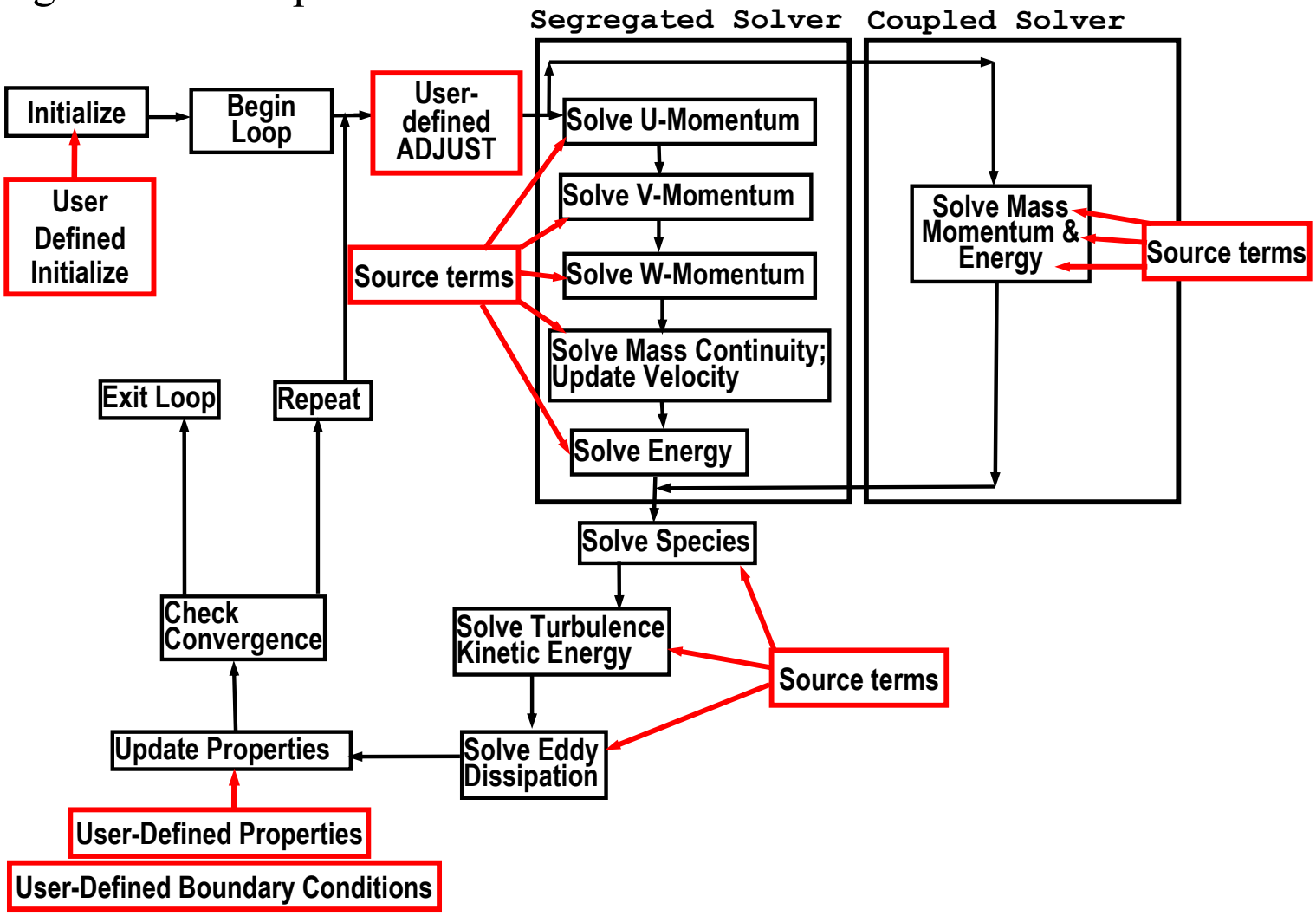
Introduction

- What is a User Defined Function?
  - A UDF is a routine (programmed by the user) written in **C** which can be dynamically linked with the solver.
    - Standard **C** functions
      - e.g., trigonometric, exponential, control blocks, do-loops, file i/o, etc.
    - Pre-Defined Macros
      - Allows access to field variable, material property, and cell geometry data.
- Why build UDF's?
  - Standard interface cannot be programmed to anticipate all needs.
    - Customization of boundary conditions, source terms, reaction rates, material properties, etc.
    - Adjust functions (once per iteration)
    - Execute on Demand functions
    - Solution Initialization

User Access to Fluent Solver

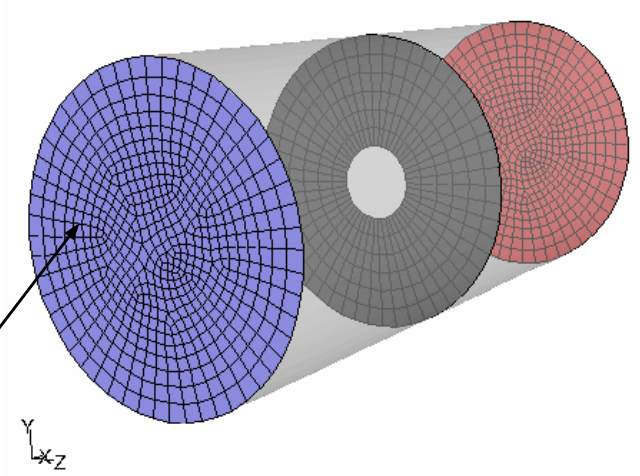
- FLUENT is designed to allow users to access the solver at some strategic instances during the solution process:

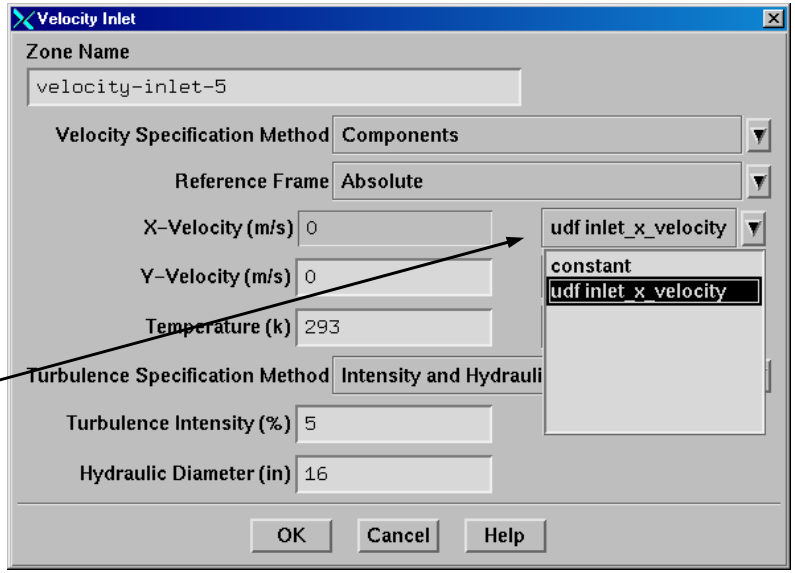
Flow Diagram of FLUENT Solvers



UDF Basics

- UDF's assigns values (e.g., boundary data, source terms) to individual cells and cell faces in fluid and boundary zones.
  - In a UDF, zones are referred to as threads.
  - A looping macro is used to access individual cells belonging to a thread.
    - e.g., a face-loop macro visits 563 faces on face zone 3 (velocity-inlet).
      - Position of each face is available to calculate and assign spatially varying properties.
    - Thread and variable references are automatically passed to UDF when assigned to boundary in GUI.





- Values returned to the solver by UDFs **must be in SI units.**

Using UDFs in the Solvers

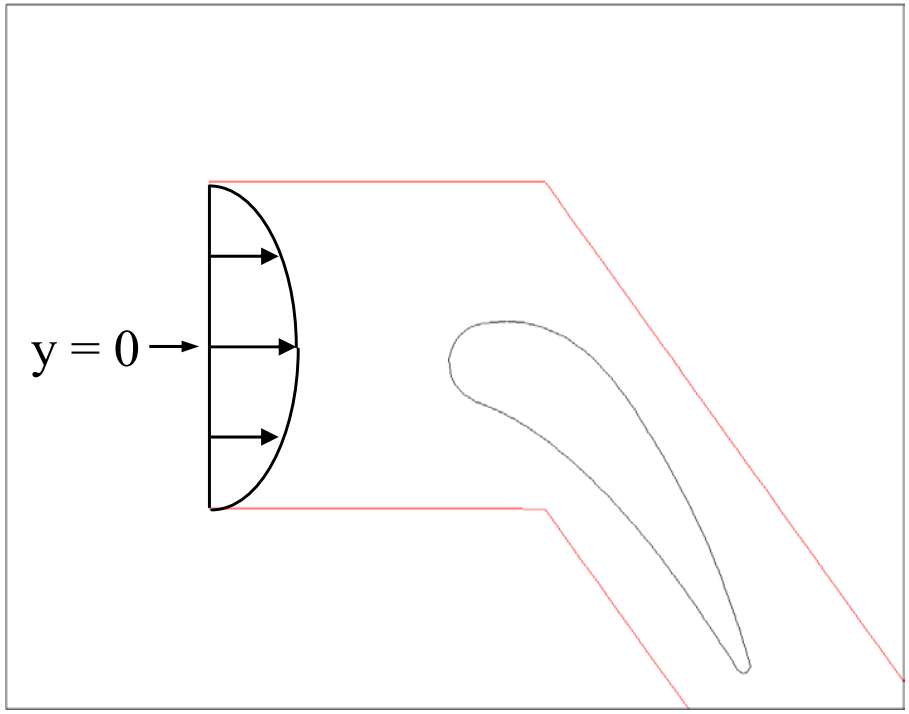
- The basic steps for using UDFs in FLUENT are as follows:

STEP 1: Create a file containing the UDF source code
STEP 2: Start the solver and read in your case/data files
STEP 3: Interpret or Compile the UDF
STEP 4: Assign the UDF to the appropriate variable and zone in BC panel.
STEP 5: Set the UDF update frequency in the Iterate panel
STEP 6: Run the calculation

Example: Non-Uniform Inlet Velocity

- A non-uniform inlet velocity is to be imposed on the 2D turbine vane shown below. The x-velocity variation is to be specified as:

$$u(y) = 20\,[\,1 - (y/0.0745)^2\,]$$



Step 1: Source Code

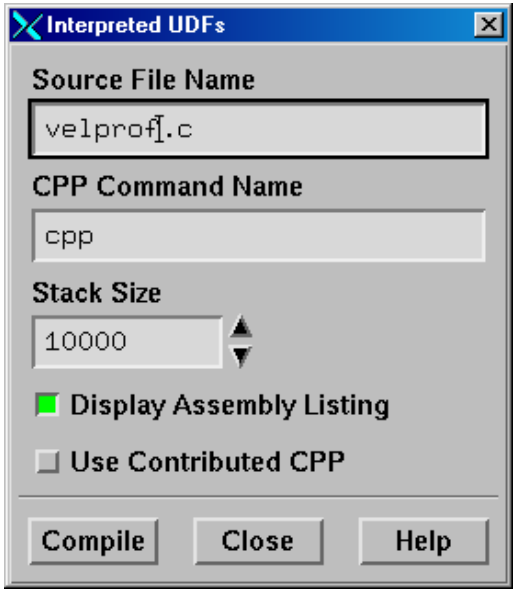
- The `DEFINE_PROFILE` macro allows the function `inlet_x_velocity` to be defined.
  - All UDFs begin with a `DEFINE_` macro.
  - `inlet_x_velocity` will be identifiable in solver GUI.
- `thread` and `nv` are *dynamic* references: *input* to the UDF to identify the zone and variable being defined, respectively.
- The macro `begin_f_loop` loops over all faces, `f`, on `thread`.

```
#include "udf.h"

DEFINE_PROFILE(inlet_x_velocity, thread, nv)
{
   float x[3]; /* this will hold the position
                  vector*/
   float y;
   face_t f;

   begin_f_loop(f, thread)
   {
     F_CENTROID(x,f,thread);
     y = x[1];
     F_PROFILE(f, thread, nv) =
                          20.*(1.- y*y/(.0745*.0745));
   }
   end_f_loop(f, thread)
 }
```

- The `F_CENTROID` macro assigns cell position vector to `x[]`.
- The `F_PROFILE` macro applies the velocity component to face `f`.

Step 3: Interpreting the UDF

- The UDF is saved as velprof.c
- Define → User Defined Functions → Interpreted…

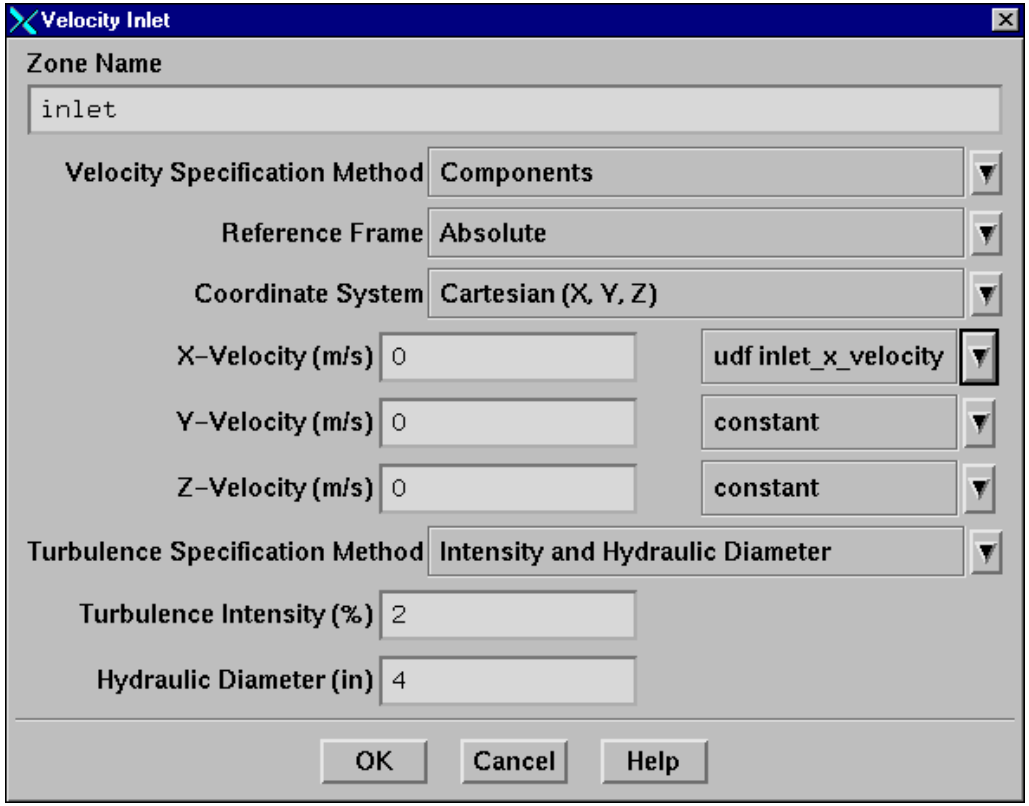


- Click Compile
- The assembly language code will scroll past window. A snapshot is shown in the right.

```
velocity_profile:
                    .local.pointer
    thread (r0)
                    .local.int
    position (r1)
     0              .local.end
     0              save
.local.int f (r6)
     8              push.int 0
    10              save
                    .local.int.
.
.
.
.L1:
132             restore
  133               restore
  134               ret.v
```

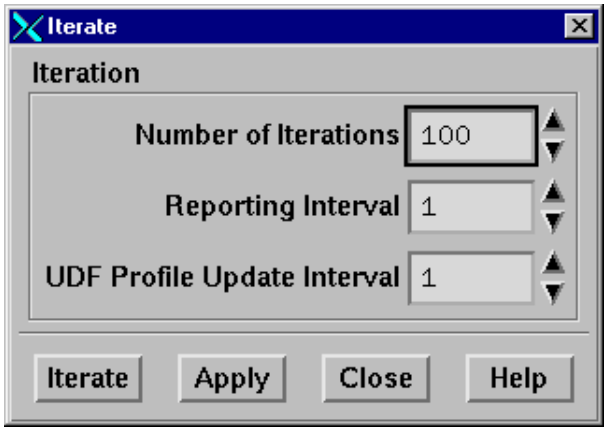
Step 4: Activating the UDF

- Access the boundary condition panel.
- Switch from constant to the UDF function in the X-Velocity dropdown list.



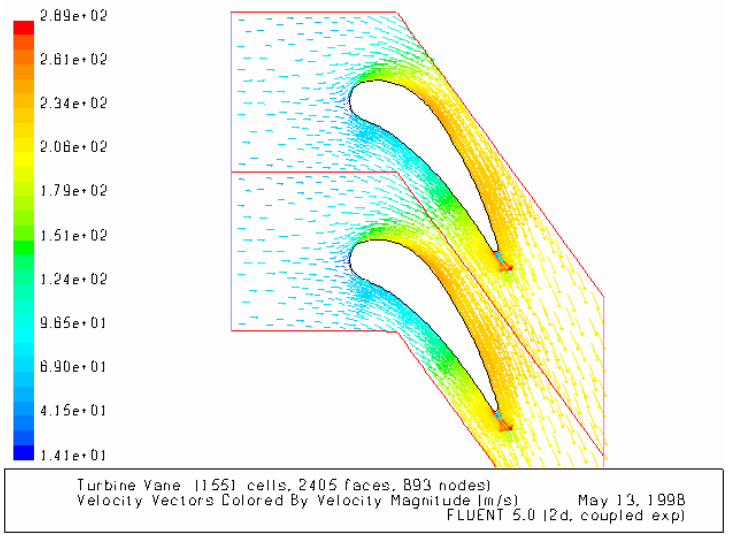
Step 5 and 6: Run the Calculation

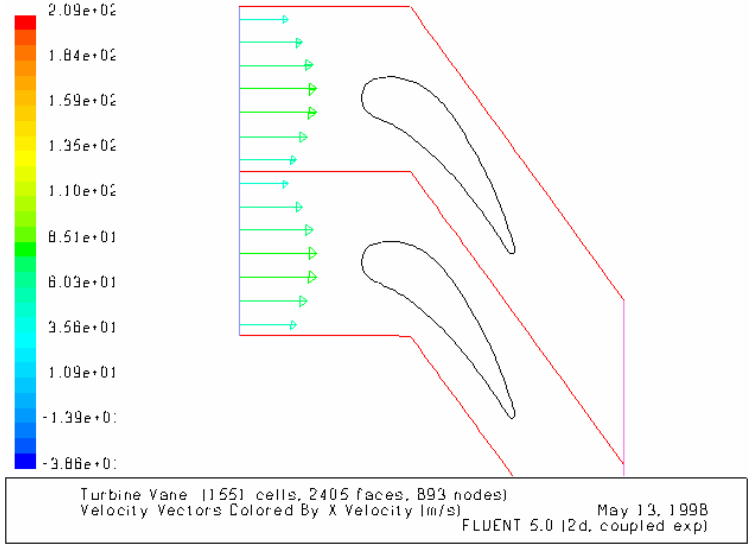
- You can change the UDF Profile update Interval in the Iterate panel (default value is 1).
- Run the calculation as usual.



Numerical Solution of the Example

- The figure at right shows velocity field throughout turbine blade passage.
- The bottom figure shows the velocity vectors at the inlet. Notice the imposed parabolic profile.





Macros

- Macros are pre-defined (Fluent) functions:
  - **`DEFINE_`** macros allows definitions of UDF functionality.
  - Variable access macros allow access to field variables and cell information.
  - Utility macros provide looping capabilities, thread identification, vector and numerous other functions.
- Macros are defined in *header* files.
  - The udf.h header file must be included in your source code.
    - #include “udf.h”
  - The header files must be accessible in your path.
    - Typically stored in Fluent.Inc/src/ directory.
- A list of often used macros is provided in the UDF User’s Guide.
  - Help → More Documentation

DEFINE Macros

- Any UDF you write must begin with a **DEFINE** macro:
  - 18 'general purpose' macros and 13 DPM and multiphase related macros (not listed):

DEFINE_ADJUST(name,domain); general purpose UDF called every iteration
DEFINE_INIT(name,domain); UDF used to initialize field variables
DEFINE_ON_DEMAND(name); defines an 'execute-on-demand' function
DEFINE_RW_FILE(name,fp); customize reads/writes to case/data files
DEFINE_PROFILE(name,thread,index); defines boundary profiles
DEFINE_SOURCE(name,cell,thread,dS,index); defines source terms
DEFINE_HEAT_FLUX(name,face,thread,c0,t0,cid,cir); defines heat flux
DEFINE_PROPERTY(name,cell,thread); defines material properties
DEFINE_DIFFUSIVITY(name,cell,thread,index); defines UDS and species diffusivities
DEFINE_UDS_FLUX(name,face,thread,index); defines UDS flux terms
DEFINE_UDS_UNSTEADY(name,cell,thread,index,apu,su); defines UDS transient terms
DEFINE_SR_RATE(name,face,thread,r,mw,yi,rr); defines surface reaction rates
DEFINE_VR_RATE(name,cell,thread,r,mw,yi,rr,rr_t); defines vol. reaction rates
DEFINE_SCAT_PHASE_FUNC(name,cell,face); defines scattering phase function for DOM
DEFINE_DELTAT(name,domain); defines variable time step size for unsteady problems
DEFINE_TURBULENT_VISCOSITY(name,cell,thread); defines procedure for calculating turbulent viscosity
DEFINE_TURB_PREMIX_SOURCE(name,cell,thread,turbflamespeed,source); defines turb. flame speed
DEFINE_NOX_RATE(name,cell,thread,nox); defines NOx production and destruction rates

Some Thread and Looping Utility Macros

`cell_t c;` defines a cell
`face_t f;` defines a face
`Thread *t;` pointer to a thread
`Domain *d;` pointer to collection of all threads

} Specialized variable types used for referencing.

`thread_loop_c(t, d){}` loop that steps through all cell threads in domain
`thread_loop_f(t, d){}` loop that steps through all face threads in domain
`begin_c_loop(c, t){} end_c_loop(c, t)` loop that steps through all cells in a thread
`begin_f_loop(f, t){} end_f_loop(f, t)` loop that steps through all faces in a thread
`c_face_loop(c, t, n){}` loop that steps through all faces of a cell

`Thread *tf = Lookup_Thread(domain, ID);` return thread pointer of integer ID of zone
`THREAD_ID(tf);` returns zone integer ID of thread pointer

Code enclosed in `{}` is executed in loop.

Geometry and Time Macros

`C_NNODES(c, t);` returns nodes/cell
`C_NFACES(c, t);` returns faces/cell
`F_NNODES(f, t);` returns nodes/face
`C_CENTROID(x, c, t);` returns coordinates of cell centroid in array `x[]`
`F_CENTROID(x, f, t);` returns coordinates of face centroid in array `x[]`
`F_AREA(A, f, t);` returns area vector in array `A[]`
`C_VOLUME(c, t);` returns cell volume
`C_VOLUME_2D(c, t);` returns cell volume for axisymmetric domain

`real flow_time();` returns actual time
`int time_step;` returns time step number
`RP_Get_Real("physical-time-step");` returns time step size

Cell Field Variable Macros

`C_R(c,t);` density
`C_P(c,t);` pressure
`C_U(c,t);` u-velocity
`C_V(c,t);` v-velocity
`C_W(c,t);` w-velocity
`C_T(c,t);` temperature
`C_H(c,t);` enthalpy
`C_K(c,t);` turbulent KE
`C_D(c,t);` turbulent dissipation rate
`C_O(c,t);` specific dissipation of tke
`C_YI(c,t,i);` species mass fraction
`C_UDSI(c,t,i);` UDS scalars
`C_UDMI(c,t,i);` UDM scalars

`C_DUDX(c,t);` velocity derivative
`C_DUDY(c,t);` velocity derivative
`C_DUDZ(c,t);` velocity derivative

`C_DVDX(c,t);` velocity derivative
`C_DVDY(c,t);` velocity derivative
`C_DVDZ(c,t);` velocity derivative
`C_DWDX(c,t);` velocity derivative
`C_DWDY(c,t);` velocity derivative
`C_DWDZ(c,t);` velocity derivative

`C_MU_L(c,t);` laminar viscosity
`C_MU_T(c,t);` turbulent viscosity
`C_MU_EFF(c,t);` effective viscosity
`C_K_L(c,t);` laminar thermal conductivity
`C_K_T(c,t);` turbulent thermal conductivity
`C_K_EFF(c,t);` effective thermal conductivity
`C_CP(c,t);` specific heat
`C_RGAS(c,t);` gas constant
`C_DIFF_L(c,t);` laminar species diffusivity
`C_DIFF_EFF(c,t,i);` effective species diffusivity

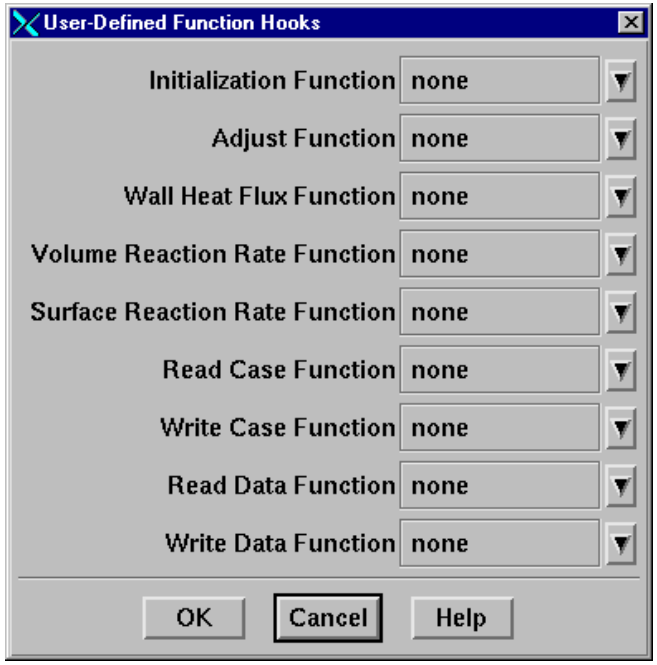
Face Field Variable Macros

- Face field variables are only available when using the segregated solver and generally, only at exterior boundaries.

`F_R(f,t);` density
`F_P(f,t);` pressure
`F_U(f,t);` u-velocity
`F_V(f,t);` v-velocity
`F_W(f,t);` w-velocity
`F_T(f,t);` temperature
`F_H(f,t);` enthalpy
`F_K(f,t);` turbulent KE
`F_D(f,t);` tke dissipation
`F_O(f,t);` specific dissipation of tke
`F_YI(f,t,i);` species mass fraction
`F_UDSI(f,t,i);` UDS scalars
`F_UDMI(f,t,i);` UDM scalars
`F_FLUX(f,t);` mass flux across face f, defined out of domain at boundaries.

Other UDF Applications

- In addition to defining boundary values, source terms, and material properties, UDFs can be used for:
  - Initialization
    - Executes once per initialization.
  - Adjust
    - Executes every iteration.
  - Wall Heat Flux
    - defines fluid-side diffusive and radiative wall heat fluxes in terms of heat transfer coefficients
    - applies to all walls
  - User Defined Surface and Volumetric Reactions
  - Read-Write to/from case and data files
    - Read order and Write order must be same.
  - Execute-on-Demand capability
    - Not accessible during solve

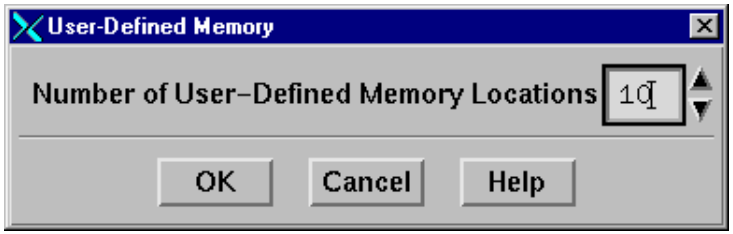


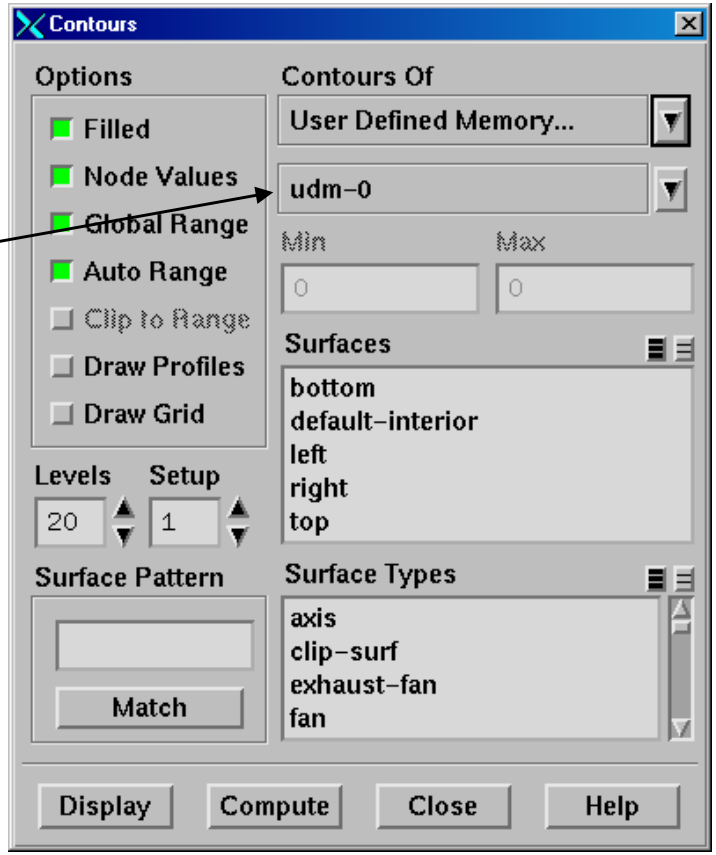
User Defined Memory

- User-allocated memory

  Define → User-Defined → Memory...

  - Up to 500 field variables can be defined.
  - Can be accessed by UDFs:
    - `C_UDMI(cell,thread,index);`
    - `F_UDMI(face,thread,index);`
  - Can be accessed for post-processing.
  - Information is stored in data file.

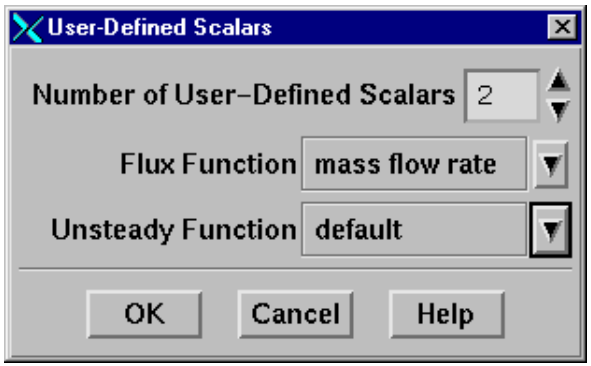




User Defined Scalars

- FLUENT can solve (up to 50) generic transport equations for User Defined Scalars, ϕ_k:

$$\frac{\partial \phi_k}{\partial t} + \frac{\partial}{\partial x_i}\left(F_i \phi_k - \Gamma_k \frac{\partial \phi_k}{\partial x_i} \right) = S_{\phi_k} \qquad k = 1, \ldots, N_{scalars}$$

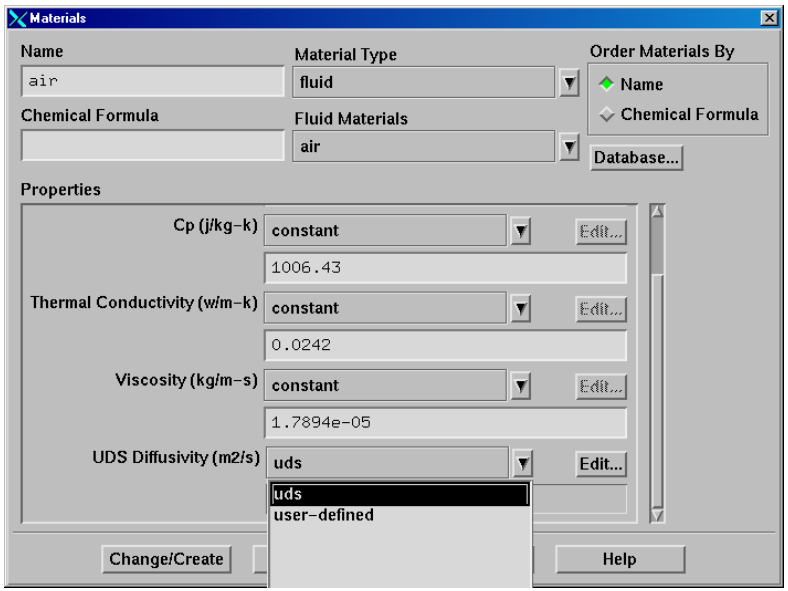


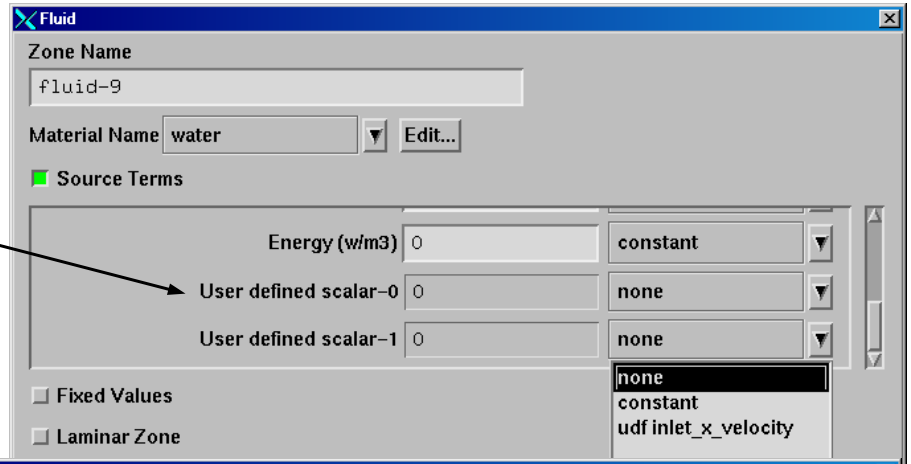
- User specifies:

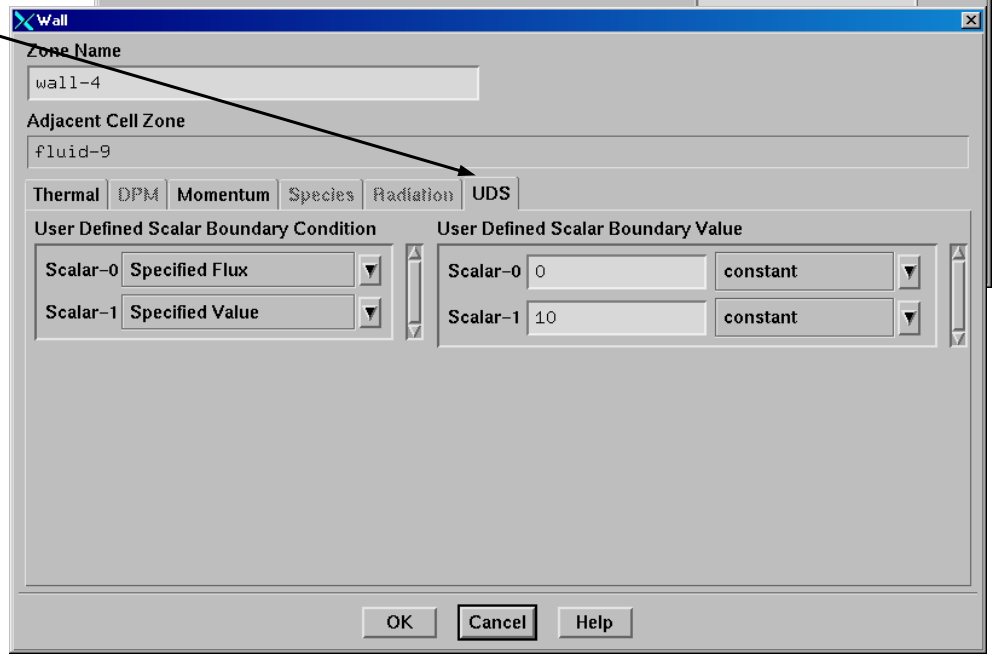
  Define → Models → User-Defined Scalars…
  - Number of User-Defined Scalars
  - Flux Function, F
    - `DEFINE_UDS_FLUX(name,face,thread,index)`
    - `DEFINE_UDS_UNSTEADY(name,cell,thread,index,apu,su)`
    - `if` statements needed to associate multiple flux and transient functions with each UDS.
- Example
  - Can be used to determine magnetic and/or electric field in a fluid zone.

User Defined Scalars (2)

- User must also specify:
  - Source terms, S_ϕ
  - Diffusivity, Γ_ϕ
    - `if` statements needed to define UDF diffusivities for each UDS.
  - Boundary Conditions for each UDS.
    - Specified Flux or Specified Value.
- Define as constant or with UDF.







Interpreted vs. Compiled UDF's

- Functions can either be read in and *interpreted* at run time (as in the example) or *compiled* and grouped into a shared library that is linked with the standard FLUENT executable.
- Interpreted vs. compiled code
  - Interpreter -
    - Interpreter is a large program that sits in the computer's memory.
    - Executes code on a "line by line" basis instantaneously
    - Advantages - does not need a separate compiler
    - Disadvantage - slow and takes up space in memory
  - Compiler (refer to User's Guide for instructions)-
    - Code is translated "once" into machine language (object modules).
    - Efficient way to run UDF's. Uses Makefiles
    - Creates "shared libraries" which are linked with the rest of the solver
    - Overcomes interpreter limitations e.g. mixed mode arithmetic, structure references etc.

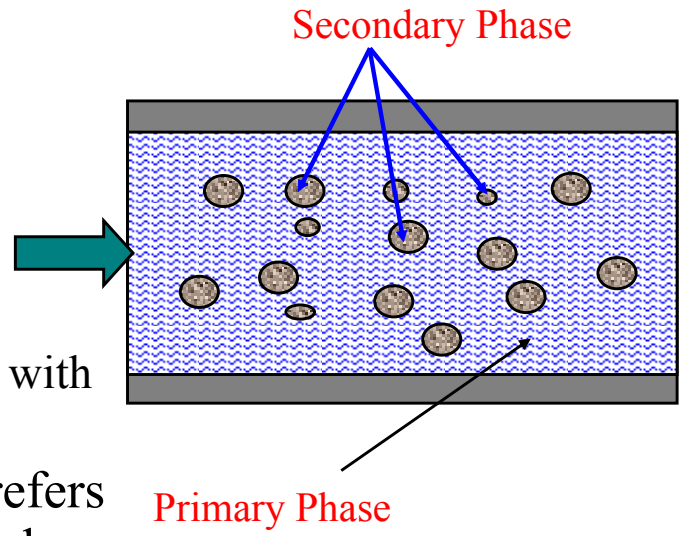
Supporting UDF's

- Because UDF's can be very complicated, Fluent Inc. does not assume responsibility for the accuracy or stability of solutions obtained using UDFs that are user-generated.
  - Support will be limited to guidance related to communication between a UDF and the FLUENT solver.
  - Other aspects of the UDF development process that include conceptual function design, implementation (writing C code), compilation and debugging of C source code, execution of the UDF, and function design verification will remain the responsibility of the UDF author.
    - Consulting option

Modeling Multiphase Flows

Introduction

- A phase is a class of matter with a definable boundary and a particular dynamic response to the surrounding flow/potential field.
- Phases are generally identified by solid, liquid or gaseous states of matter but can also refer to other forms:
  - Materials with different chemical properties but in the same state or phase (*i.e.* liquid-liquid, such as, oil-water)
- The fluid system is defined by a primary and multiple secondary phases.
  - One of the phases is considered continuous (primary)
  - The others (secondary) are considered to be dispersed within the continuous phase.
  - There may be several secondary phase denoting particles with different sizes
- In contrast, multi-component flow (species transport) refers to flow that can be characterized by a single velocity and temperature field for all the species.



Choosing a Multiphase Model

In order to select the appropriate model, users must know *a priori* the characteristics of the flow in terms of the following:

- The flow regime
- Volume loading
- Turbulent or Laminar flow
- Stokes number

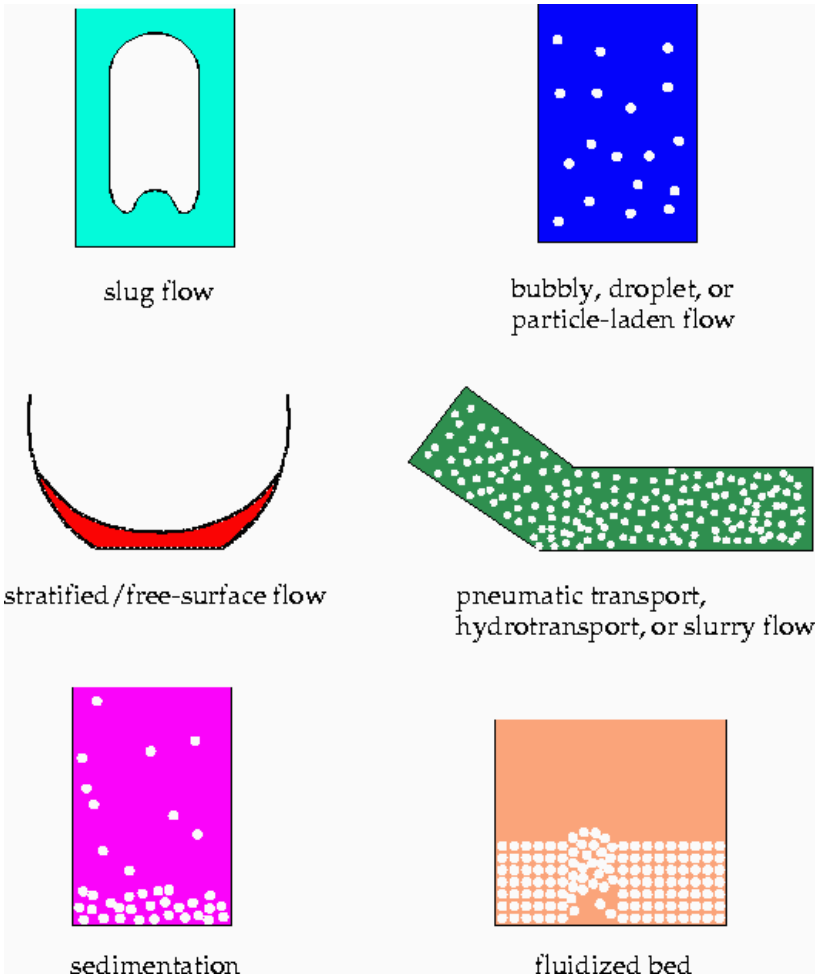
Flow Regimes

- Multiphase Flow Regimes

gas-liquid liquid-liquid
- **Bubbly flow:** Discrete gaseous bubbles in a continuous fluid. E.g.: Absorbers, evaporators, sparging devices.
- **Droplet flow:** Discrete fluid droplets in a continuous gas. E.g.: Atomizers, combustors.
- **Slug flow:** Large bubbles in a continuous liquid.
- **Stratified/free-surface flow:** Immiscible fluids separated by a clearly-defined interface. E.g.: Free surface flows.

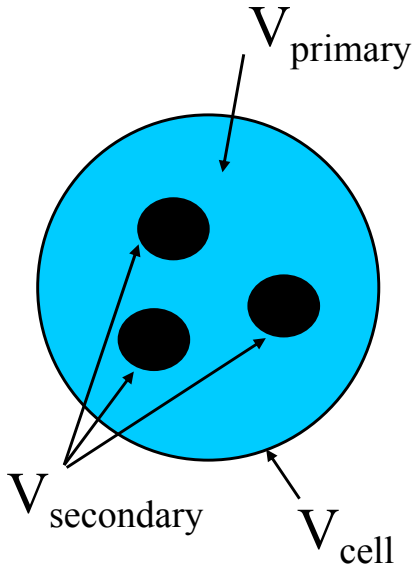
gas-solid
- **Particle-laden flow:** Discrete solid particles in a continuous gas. E.g.: cyclone separators, air classifiers, dust collectors, and dust-laden environmental flows.
- **Fluidized Beds:** fluidized bed reactors.

liquid-solid
- **Slurry Flow:** Particle flow in liquids, solids suspension, sedimentation, and hydro-transport.



Volume and Particulate loading

- Volume loading: Dilute or Dense:
  - Refers to the volume fraction of secondary phase(s)
    - Volume fraction = $\frac{\text{Volume of the phase in a cell/domain}}{\text{Volume of the cell/domain}}$
  - For dilute loading (< 10%) average particle-particle distance is about 2 times particle diameter so particle-particle interactions can be neglected.



Turbulence modeling in Multiphase Flows

- Turbulence modeling with multiphase flows can be challenging. Presently, single-phase models like k-ε or RSM are used to model turbulence in primary phase. Turbulence equations may contain additional terms to account for turbulence modification by secondary phase.
- If phases are separated and their density ratio is of order 1 or if particle volume fraction is low (<10%), then single phase model can be used for “mixture” of phases
- In other cases either single phase models are still used or “particle presence modified” models are used

Stokes Number

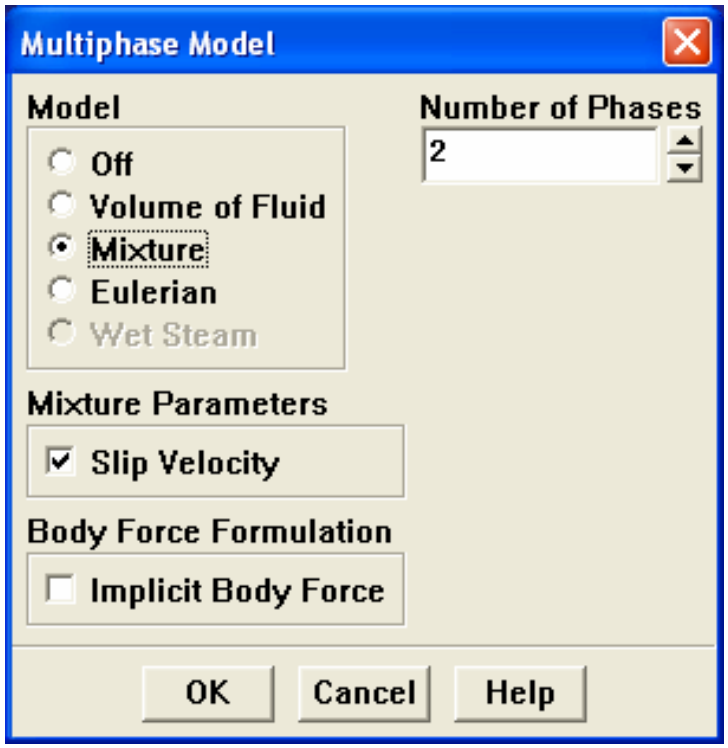
- For system with intermediate particulate loading, the Stokes number provides a guidance for choosing the most appropriate model:
  - Stokes Number, St , is the ratio of particle (the dispersed phase) relaxation time (τ_d) to the problem's characteristic time scale (t_c):
    - $St = \tau_d / t_c$
    - Where $\tau_d = \rho_d d_d^2 / 18\mu_c$ and $t_c = D / U$ (D and U are characteristic length and velocity scales of the problem)
  - When $St << 1$
    - particles will follow the flow closely (in equilibrium).
  - When $St > 1$
    - Particles will move independently of the flow field.

Phases as Mixtures of Species

- In all Fluent's multiphase models any phase can be either single material or a mixture of species
- Material definition of phase mixtures is the same as in single-phase flows
- It is possible to model heterogeneous reactions, i.e., reactions where reactants and products belong to different phases. This means that heterogeneous reactions will lead to interfacial mass exchange

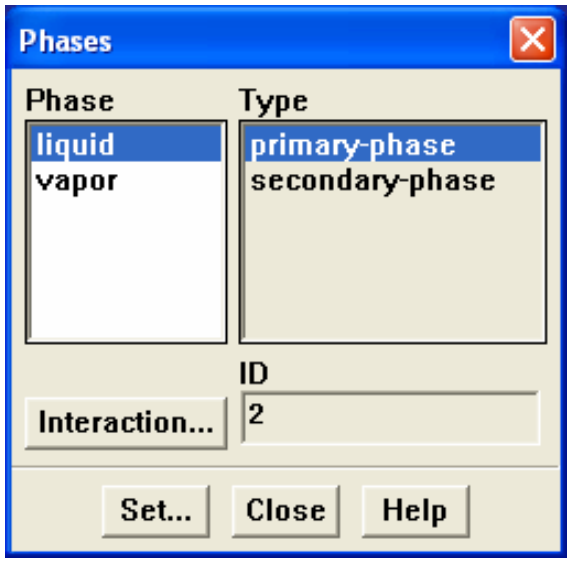
Multiphase Models in Fluent

Define → Models → Multiphase…

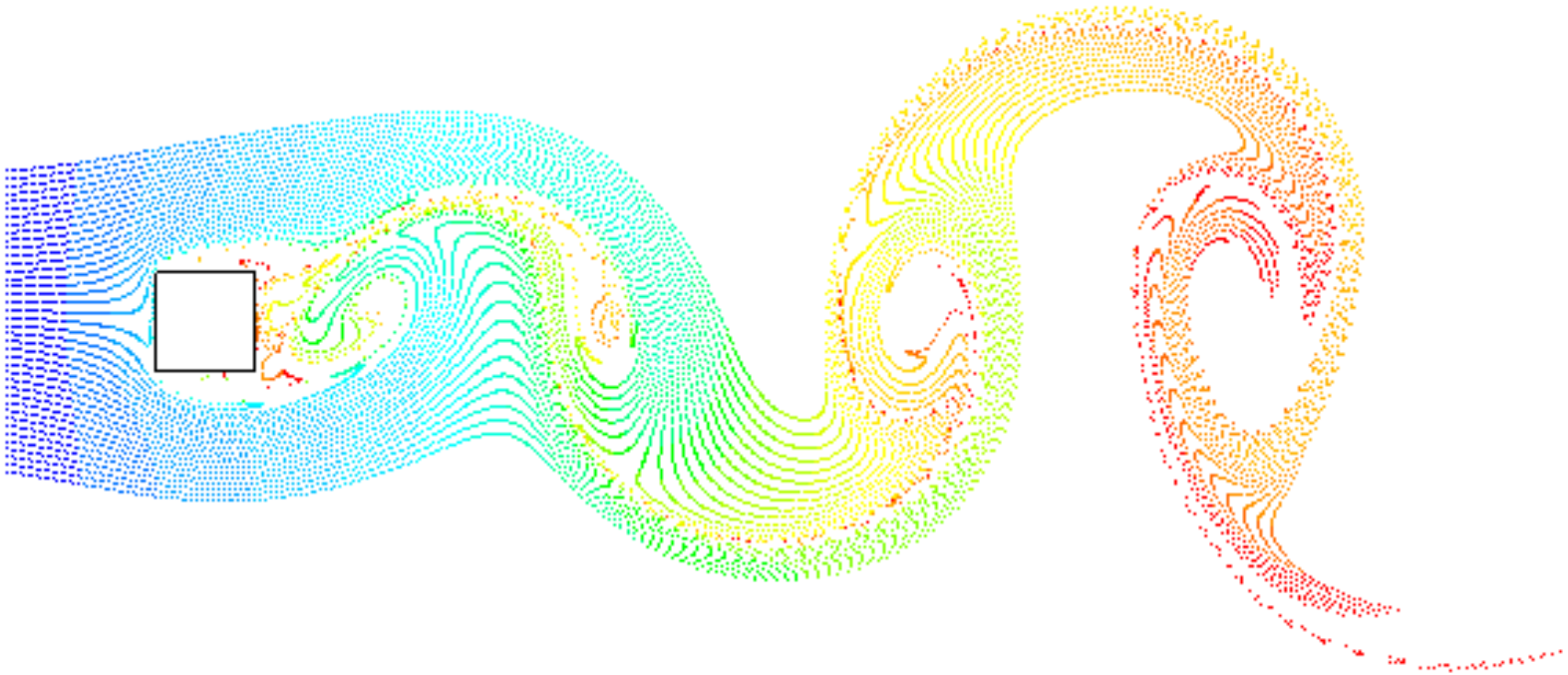


- Discrete Phase Model (DPM)
- Mixture Model
- Eulerian Multiphase Flow Model
- Volume of Fluid Model (VOF)

Define → Phases



Discrete Phase Model



Discrete Phase Model (DPM)

- Trajectories of particles/droplets/bubbles are computed in a Lagrangian frame.
  - Particles can exchange heat, mass, and momentum with the continuous gas phase.
  - Each trajectory represents a *group* of particles of the same initial properties.
  - Particle-Particle interaction is neglected.
  - Turbulent dispersion can be modeled with stochastic tracking or a "particle cloud" model.
- Numerous sub-modeling capabilities:
  - Heating/cooling of the discrete phase
  - Vaporization and boiling of liquid droplets
  - Volatile evolution and char combustion for combusting particles
  - Droplet breakup and coalescence using Spray Models
  - Erosion/Accretion

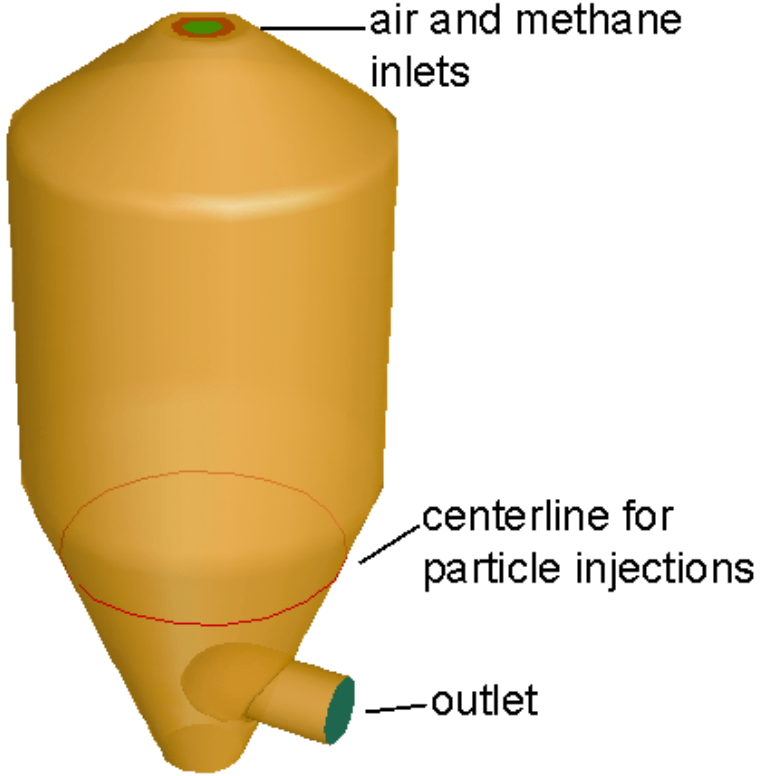
Applicability of DPM

- Flow regime: Bubbly flow, droplet flow, particle-laden flow.
- Volume loading: Must be Dilute (volume fraction < 12%).
- Particulate Loading: Low to moderate.
- Turbulence modeling: Weak to strong coupling between phases.
- Stokes Number: All ranges of stokes number.

Application examples: Cyclones, spray dryers, particle separation and classification, aerosol dispersion, liquid fuel and coal combustion. etc.

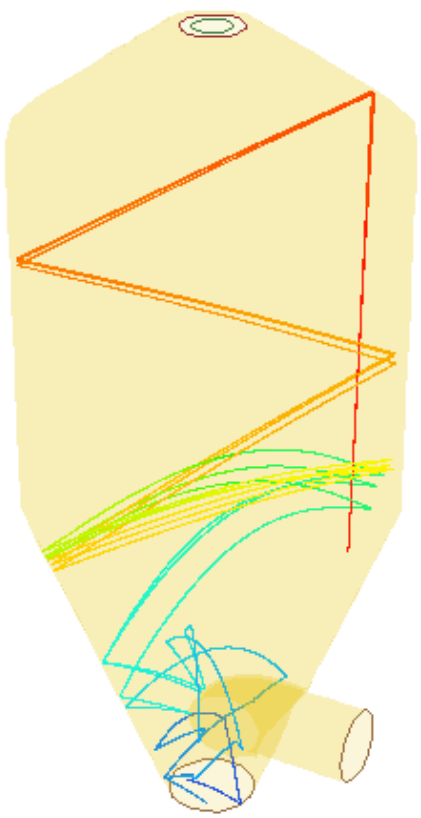
Example: Spray Dryer Simulation

- Spray drying involves the transformation of a liquid spray into dry powder in a heated chamber. The flow, heat and mass transfer are simulated by DPM capability in FLUENT.
- CFD simulation plays a very important role in optimizing the various parameters for the spray dryer.

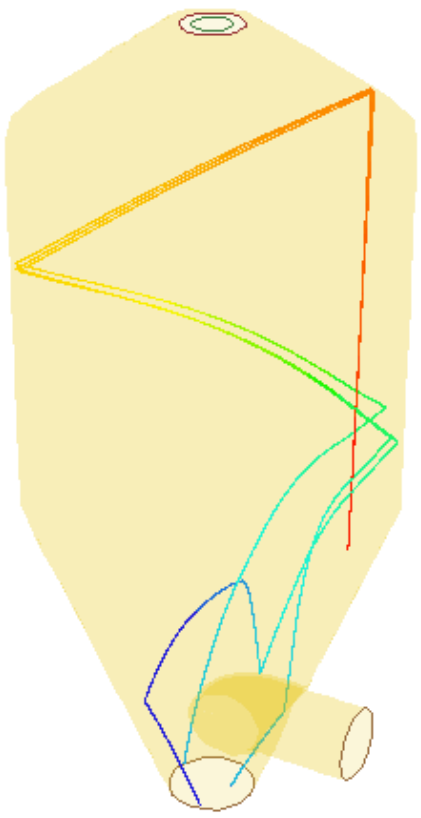


Path lines indicating the gas flow field

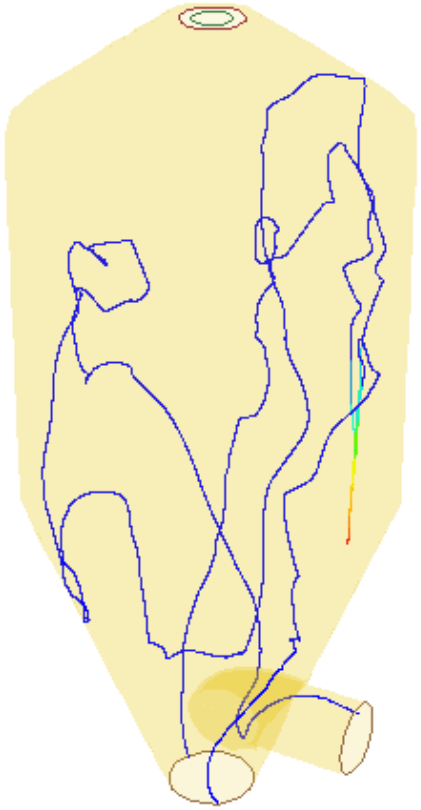
Spray Dryer Simulation (2)



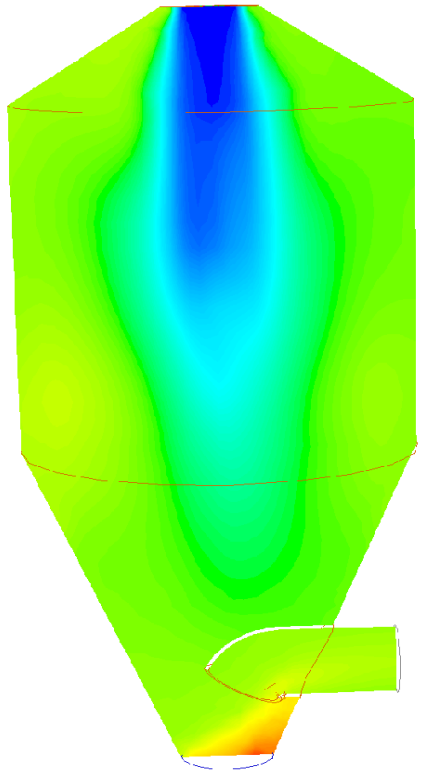
Initial particle
Diameter: 2 mm



1.1 mm



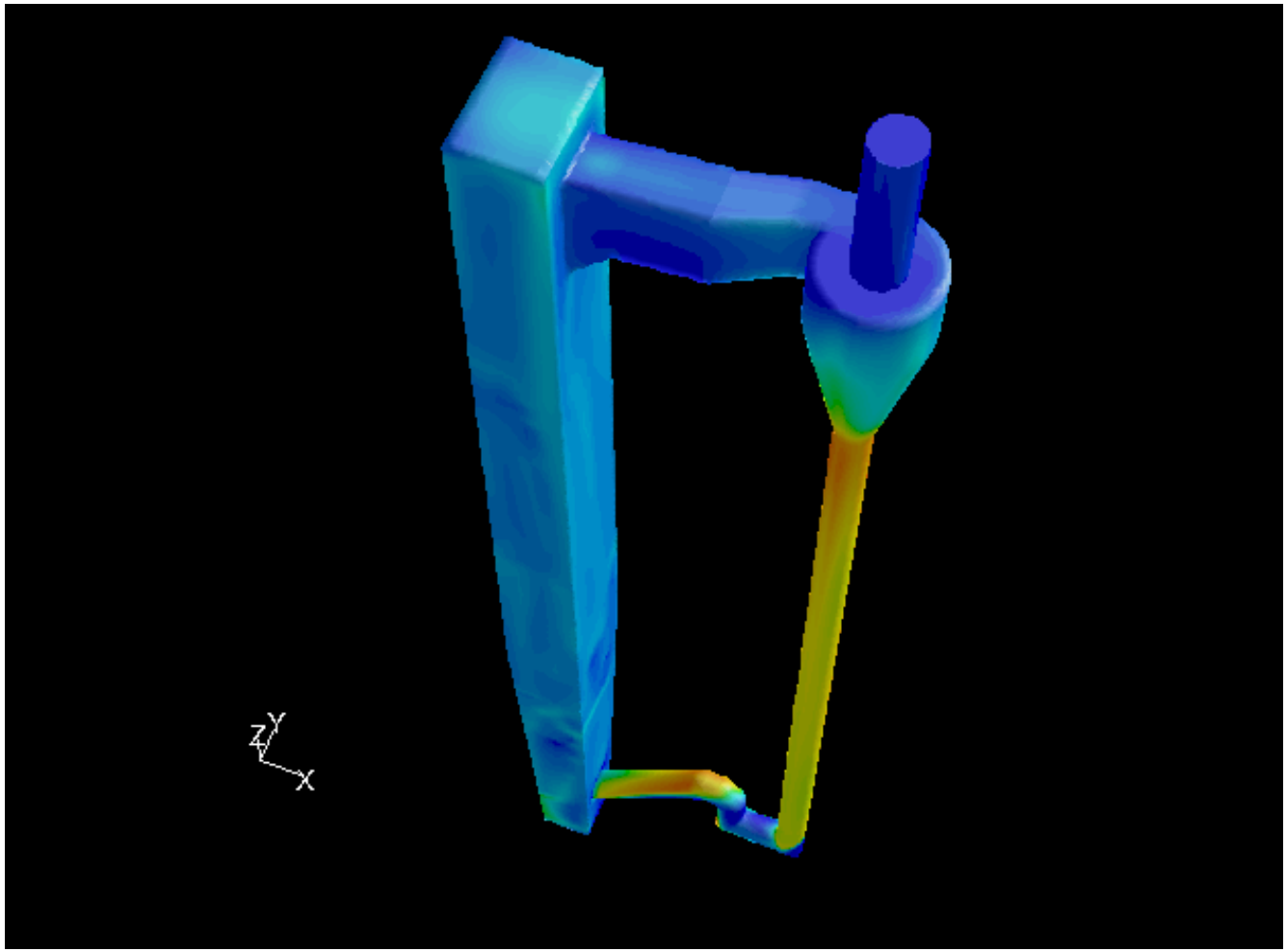
0.2 mm



Contours of the evaporated water

Stochastic particle for particles of different initial diameters

The Eulerian Multiphase Model



The Eulerian Multiphase Model

- Description
  - The most sophisticated and general purpose multiphase model.
    - Used to model multiple interpenetrating phases (gas, liquid, and/or solid).
  - Solves continuity, momentum and energy equations for each phase.
    - Volume fractions characterize equation set for each phase.
    - Coupling among phases achieved through inter-phase exchange terms.
      - Several models available to define inter-phase exchange coefficients.
        - Typically drag coefficient models described in terms of local *Reynolds number*.
      - Strong coupling makes this model more difficult to use than Mixture Model.
    - Heat, mass transfer between phases including heterogeneous reactions is possible
  - Uses a single pressure field for all phases.

The Granular option in the Eulerian Model

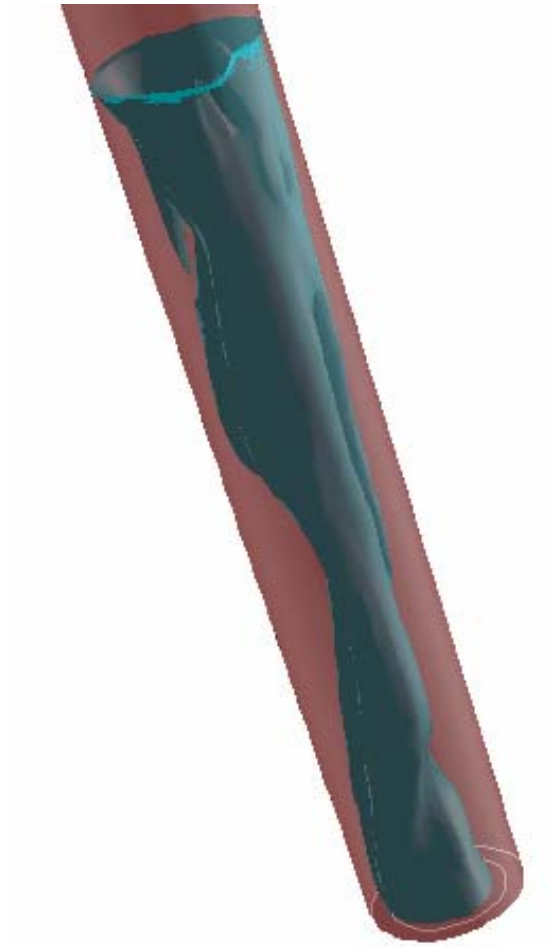
- For fluid-granular flows, each granular phase is treated as a distinct interpenetrating granular 'fluid'.
- In addition to primary fluid phase, secondary granular phase momentum equations are solved.
  - Solid-phase stresses and properties derived using analogy of thermal motion of molecules.
    - Intensity of particle velocity fluctuations determines the stresses, viscosity, and pressure of the solid phase.
    - Kinetic energy associated with the particle velocity fluctuations represented by a "pseudo-thermal" or granular temperature.
    - Inelasticity of the granular phase is taken into account

Applicability of Eulerian model

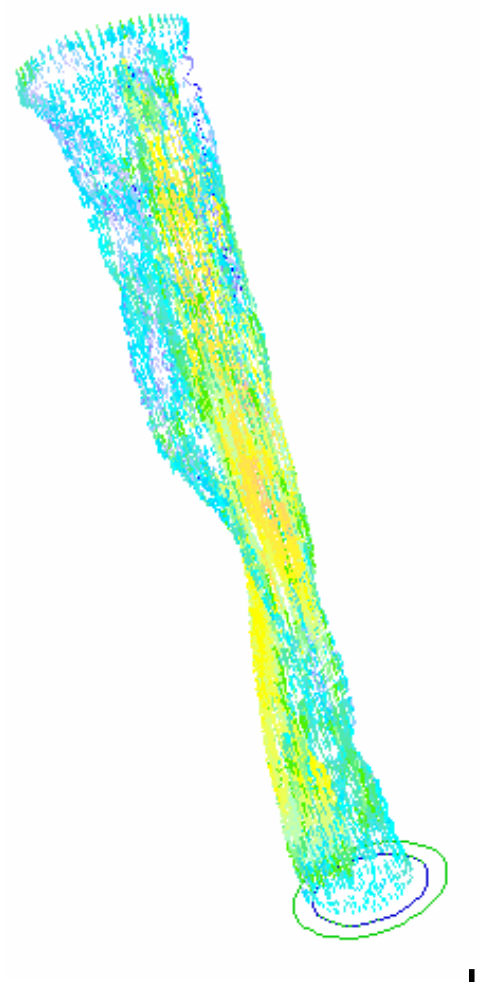
- Flow regime: Bubbly flow, droplet flow, slurry flow, fluidized beds, particle-laden flow.
- Volume loading: Dilute to dense.
- Particulate Loading: Low to high.
- Turbulence modeling: Weak to strong coupling between phases.
- Stokes Number: All ranges of Stokes number.

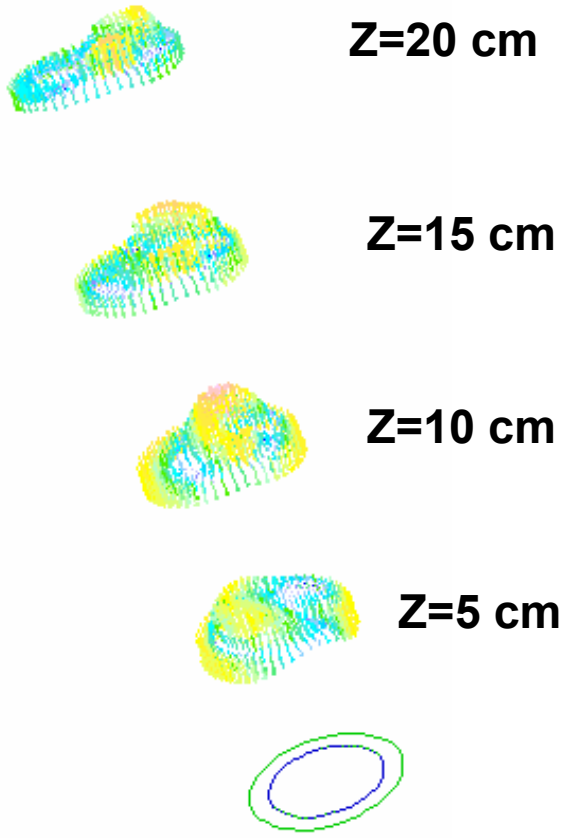
Application examples: High particle loading flows, slurry flows, sedimentation, hydro-transport, fluidized beds, risers, packed bed reactors.

Example: 3D Bubble Column



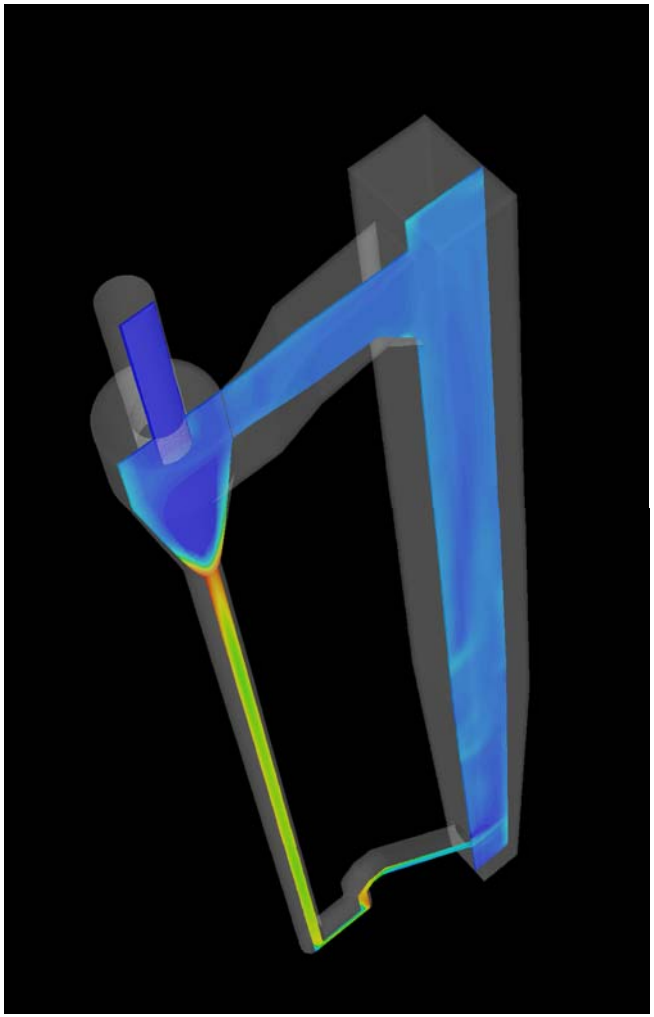
Gas
Volume Fraction = 0.175

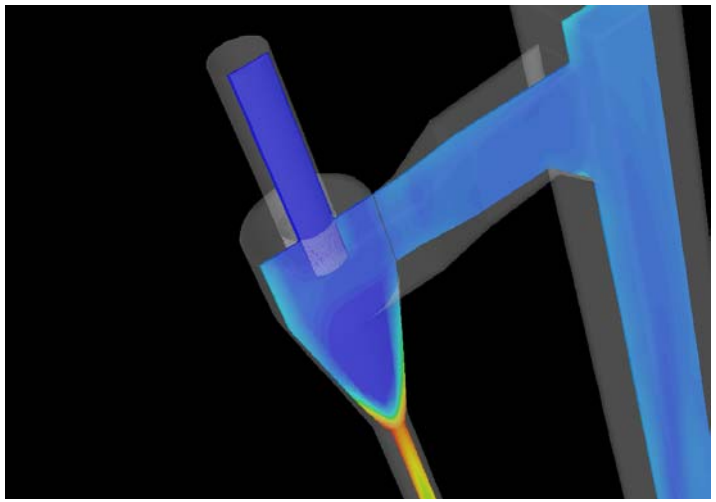


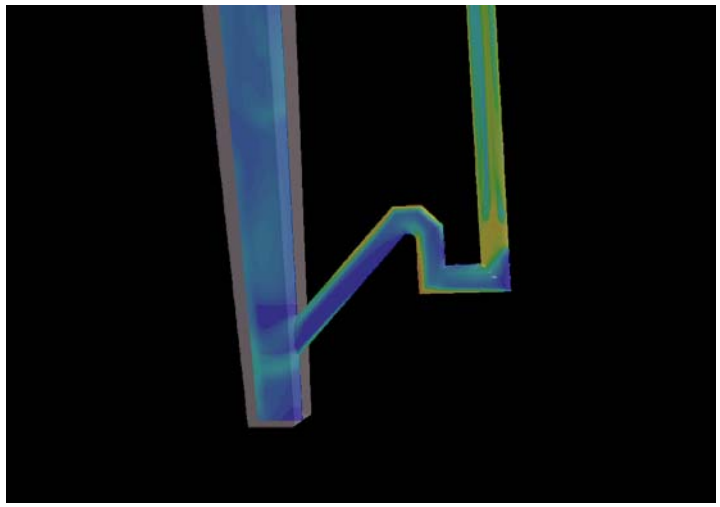


Liquid Velocity Vectors

Circulating Fluidized Bed

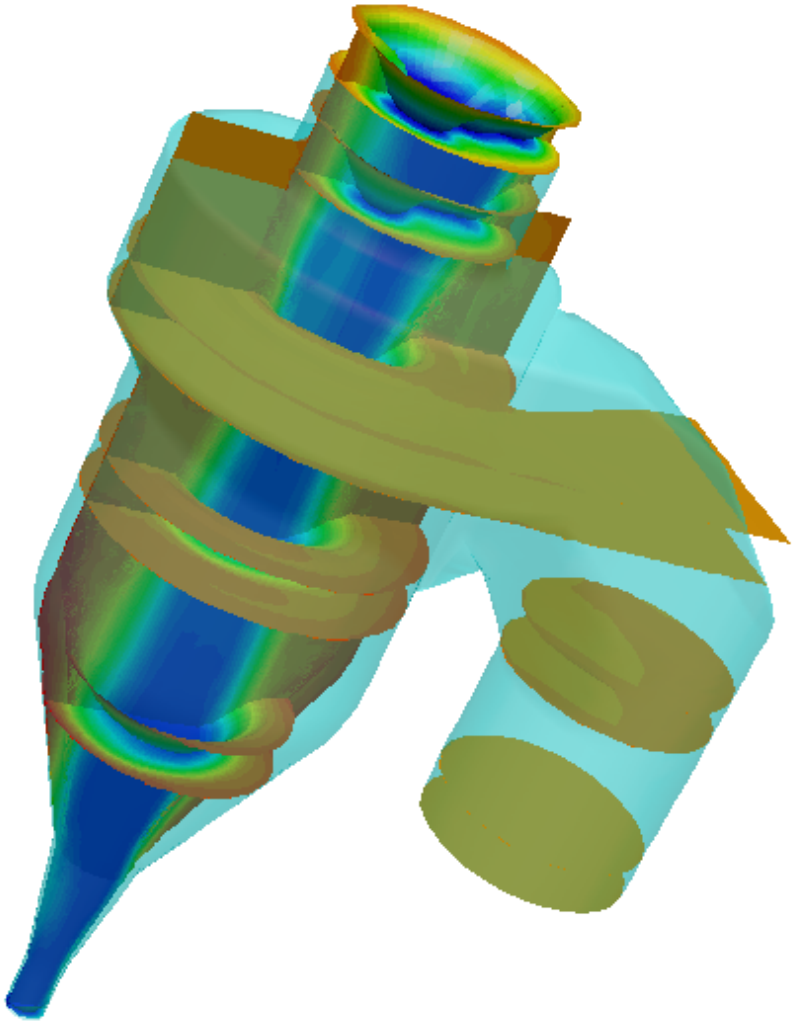






Contours of Solid Volume Fraction

The Mixture Model



Courtesy of
Fuller Company

The Mixture Model

- Description
  - Mixture model is simplified Eulerian approach for modeling N-phase flows.
  - The simplification is based on main assumption that Stokes number is small, i.e., magnitude and direction of particle velocity vector is close to that of primary fluid
  - Solves the mixture momentum equation (for mass-averaged mixture velocity) and prescribes relative velocities to describe the dispersed phases.
    - Inter-phase exchange terms depend on relative (slip) velocities which are algebraically determined based on main assumption that St<<1. This means that phase separation can me modeled with Mixture model.
    - Turbulence and Energy equations are also solved for the mixture, if required.
    - Only one of the phases may be defined as compressible.
  - Solves the transport equation of volume fraction for each secondary phase.
  - Mass exchange is possible including heterogeneous reactions
  - A submodel for cavitation is available, see the Appendix for more detail.

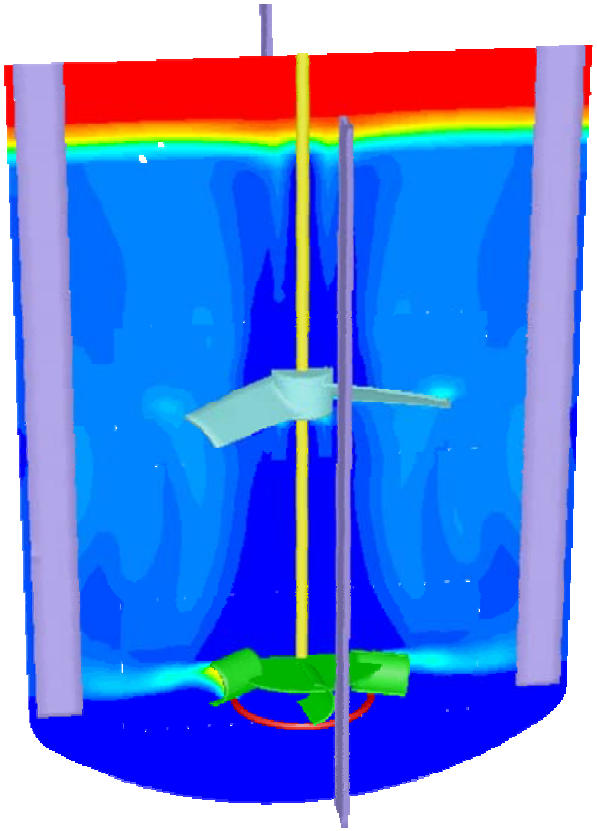
Applicability of Mixture model

- Flow regime: Bubbly flow, droplet flow, slurry flow.
- Volume loading: Dilute to moderately dense.
- Particulate Loading: Low to moderate.
- Turbulence modeling: Weak coupling between phases.
- Stokes Number: Stokes Number $<<$ 1.

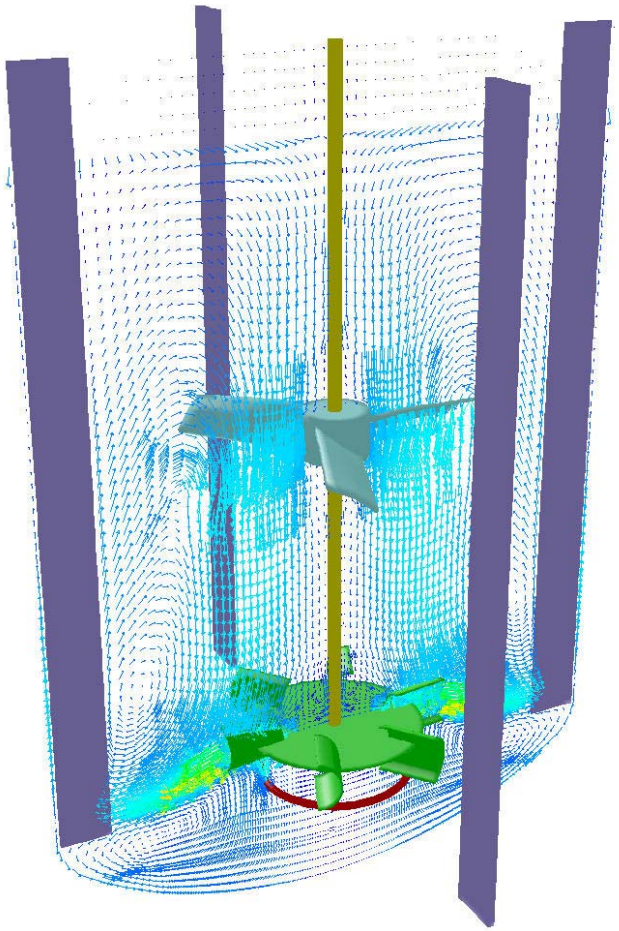
Application examples: Hydrocyclones, bubble column reactors, solid suspensions, gas sparging.

Example: Gas Sparging in Mixing Tanks

- The sparging of nitrogen gas into a stirred tank is simulated by the mixture multiphase model. The rotating impeller is simulated by using multiple-reference-frame (MRF) model.
- FLUENT simulation provided a good prediction on the gas-holdup of the agitation system.

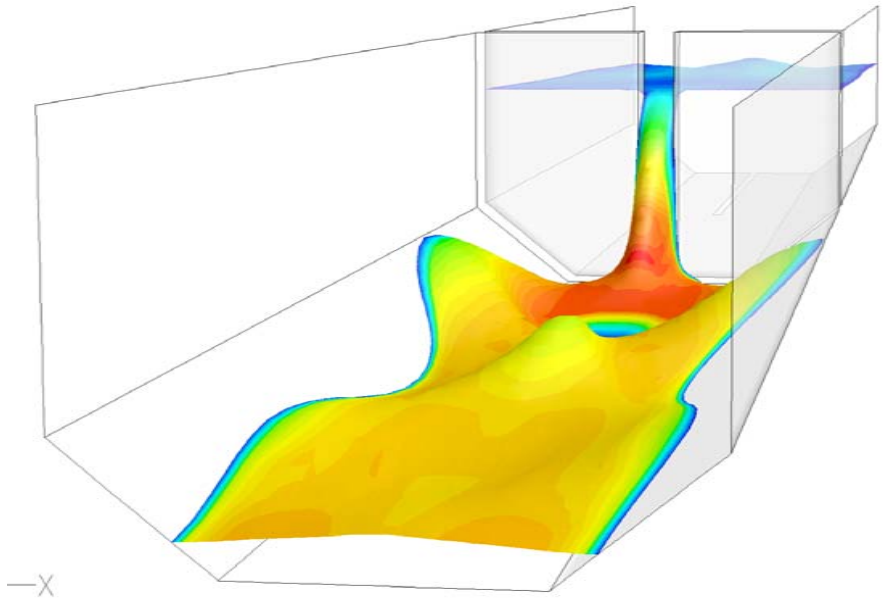


Contours of volume fraction of gas (t = 15s)



Velocity vectors of water on a center plane

The Volume of Fluid Model (VOF)



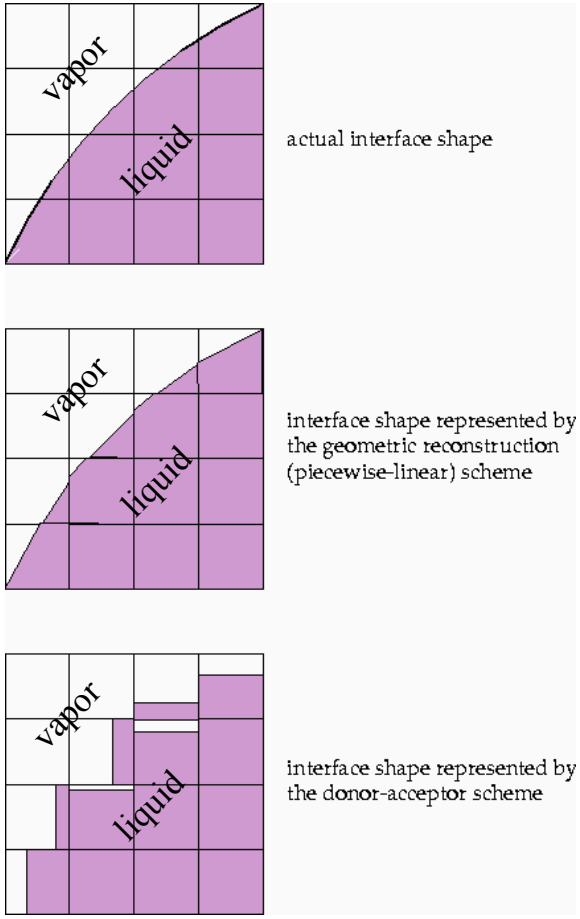
The Volume of Fluid Model (VOF)

- Description
  - The VOF model is designed to track the position of the interface between two or more immiscible fluids.
  - A single momentum equation is solved and the resulting velocity field is shared by all phases.
  - Turbulence and energy equations are also shared by all phases, if required.
  - One of the phases may be defined as compressible.
  - Surface tension and wall adhesion effects can be taken into account.
  - Solves transport equation for volume fraction of each secondary phase. Any phase can be a mixture of species. Heterogeneous reactions are possible.

Interface Interpolation Schemes

- The standard interpolation schemes used in Fluent are used to obtain the face fluxes whenever a cell is completely filled with one phase.
- Four interpolation schemes available for evaluating face fluxes when interface is in cell.
  - Geometric Reconstruction
    - Default scheme, must be used unsteady, most accurate
  - Donor-Acceptor
    - Must be used unsteady, best scheme if mesh contains highly skewed hex elements.
  - Euler-Explicit
    - Must be used unsteady, use if skewed cells exist in a tet/hybrid mesh.
  - Implicit
    - Can be used steady or unsteady.

decreasing accuracy



Applicability of VOF model

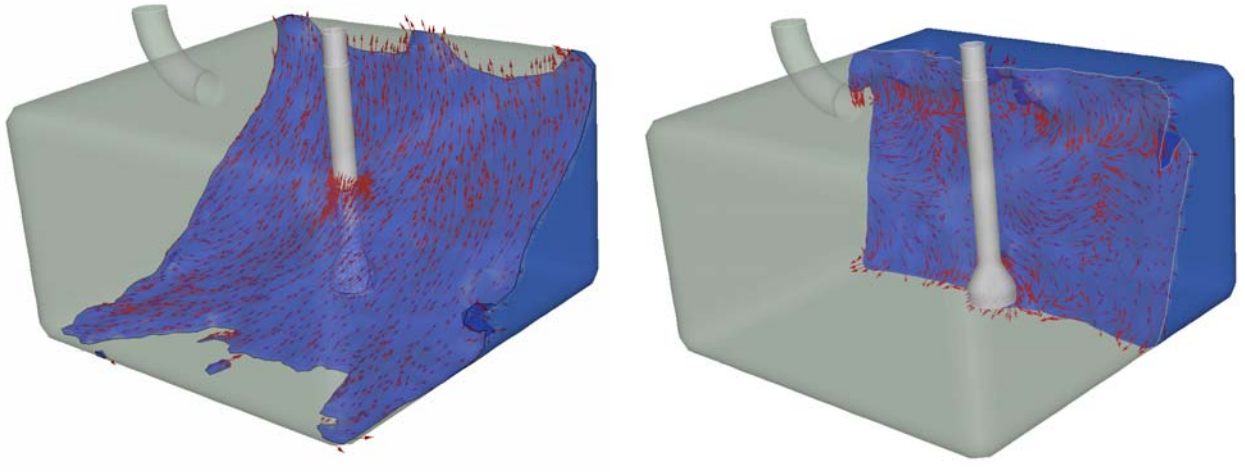
- Flow regime: Slug flow, stratified/free-surface flow.
- Volume loading: Dilute to dense.
- Particulate Loading: Low to high.
- Turbulence modeling: Weak to moderate coupling between phases.
- Stokes Number: All ranges of Stokes number.

Application examples:Large slug flows, filling, off-shore oil tank sloshing, boiling, coating.

Example: Automotive Fuel Tank Sloshing

Sloshing (free surface movement) of liquid in an automotive fuel tank under various accelerating conditions is simulated by the VOF model in FLUENT.

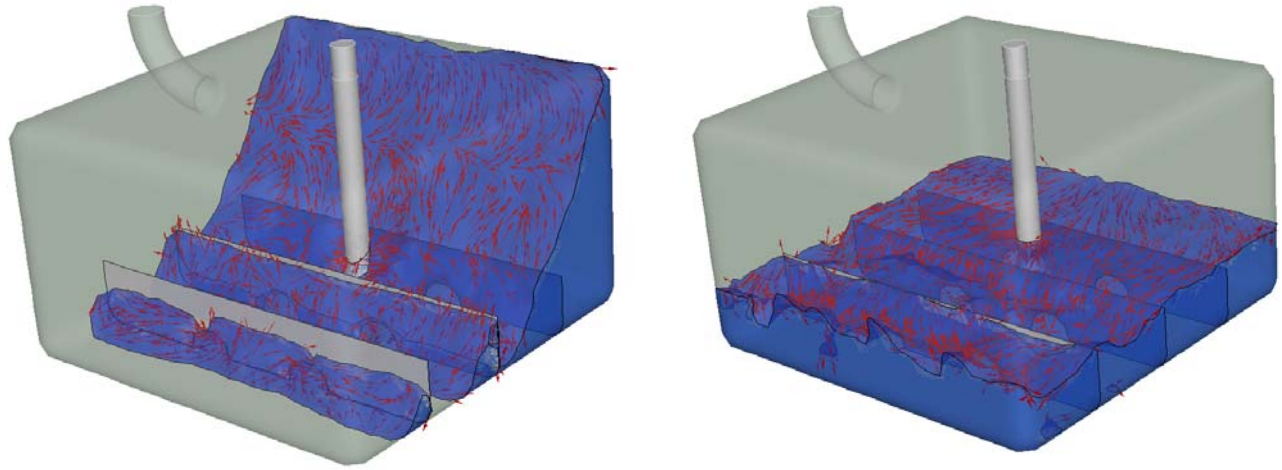
◆CSimulation shows the tank with internal baffles (at bottom) will keep the fuel intake orifice to be fully submerged at all times, while the intake orifice is out of the fuel at certain times for the tank without internal baffles (top).



t=1.05 s

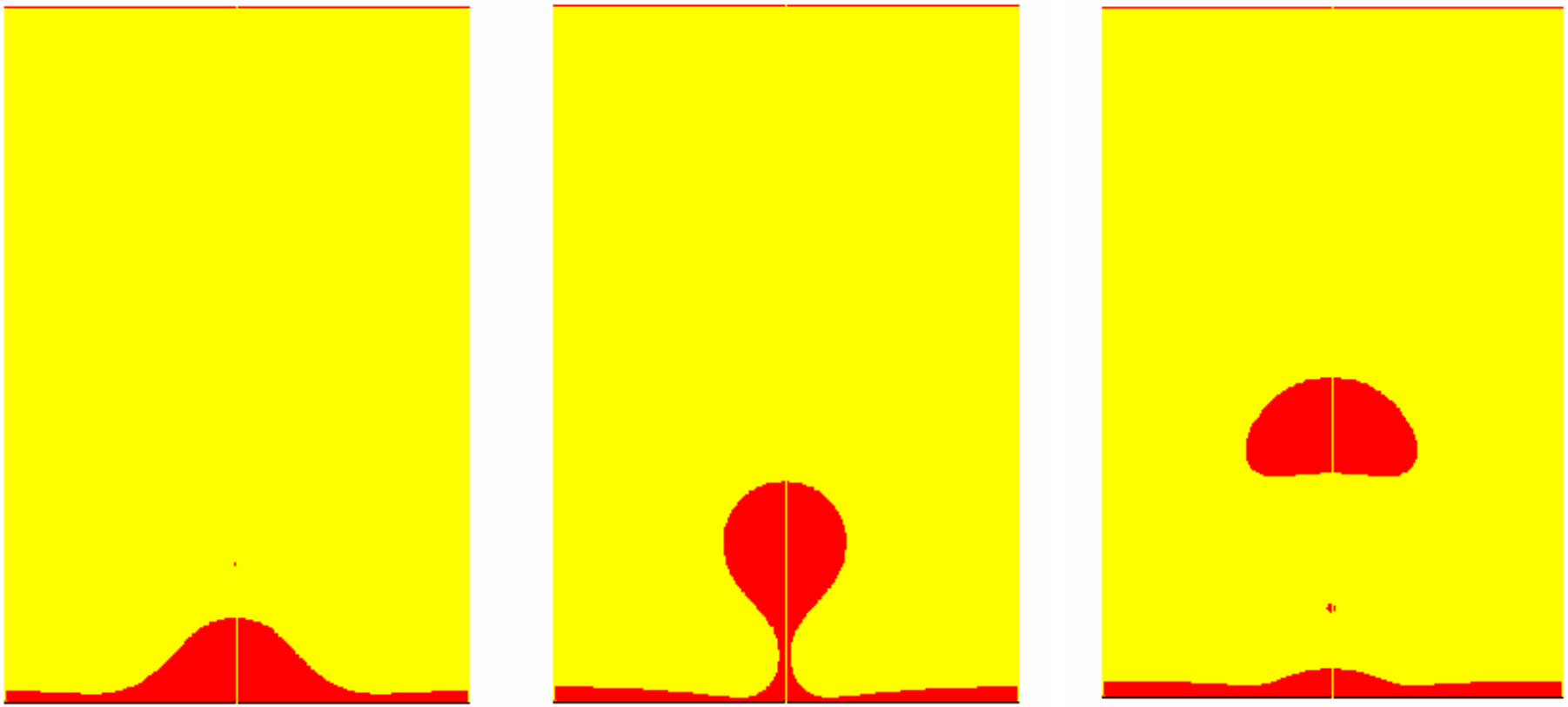
Fuel tank without baffles

t=2.05 s



Fuel tank with baffles

Example: Horizontal Film Boiling Simulation by the VOF Model



Plots showing the rise of bubbles during the film boiling process
(the contours of vapor volume fraction are shown in red)

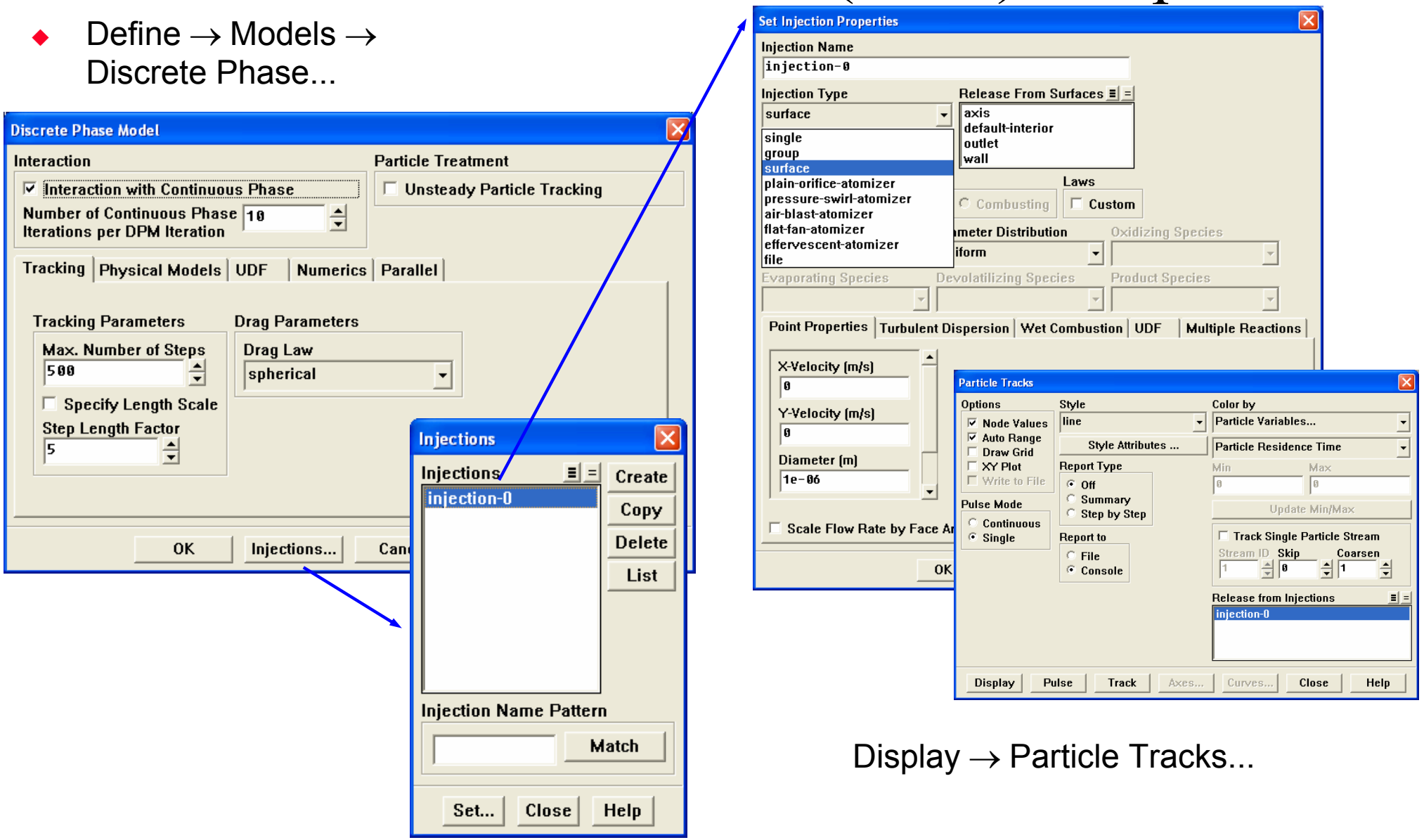
Summary

- Choose an appropriate model for your application based on: Flow regime, volume loading, particulate loading, turbulent or laminar, and Stokes number.
  - For free surface and stratified flows, choose VOF.
  - For High particle loading flows use Eulerian-Granular Model.
  - For Low-to-Moderate particle loadings consider Stokes number:
    - For $St > 1$, Mixture model is not applicable.
      - Choose DPM or Eulerian.
    - For $St \leq 1$, all models are applicable
      - Choose least CPU demanding model based on other requirements.
- Strong coupling among phase equations solve better with lowered under-relaxation factors.
- Users should understand the limitations and applicability of each model.

Appendix

Discrete Phase Model (DPM) Setup

- Define → Models → Discrete Phase...

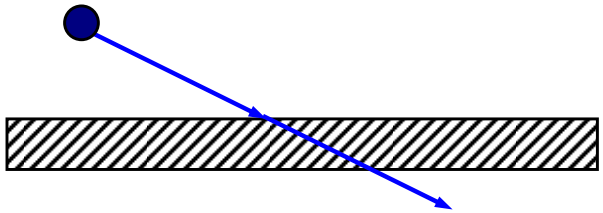


Display → Particle Tracks...

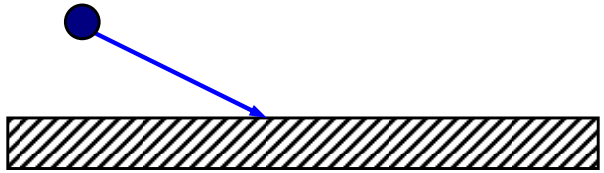
DPM Boundary Conditions

- Boundary Conditions for DPM must be set.

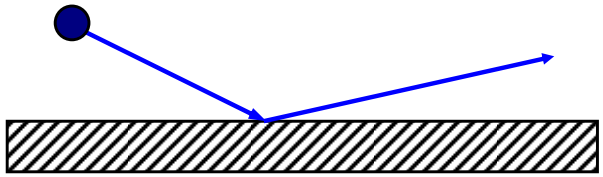
Escape

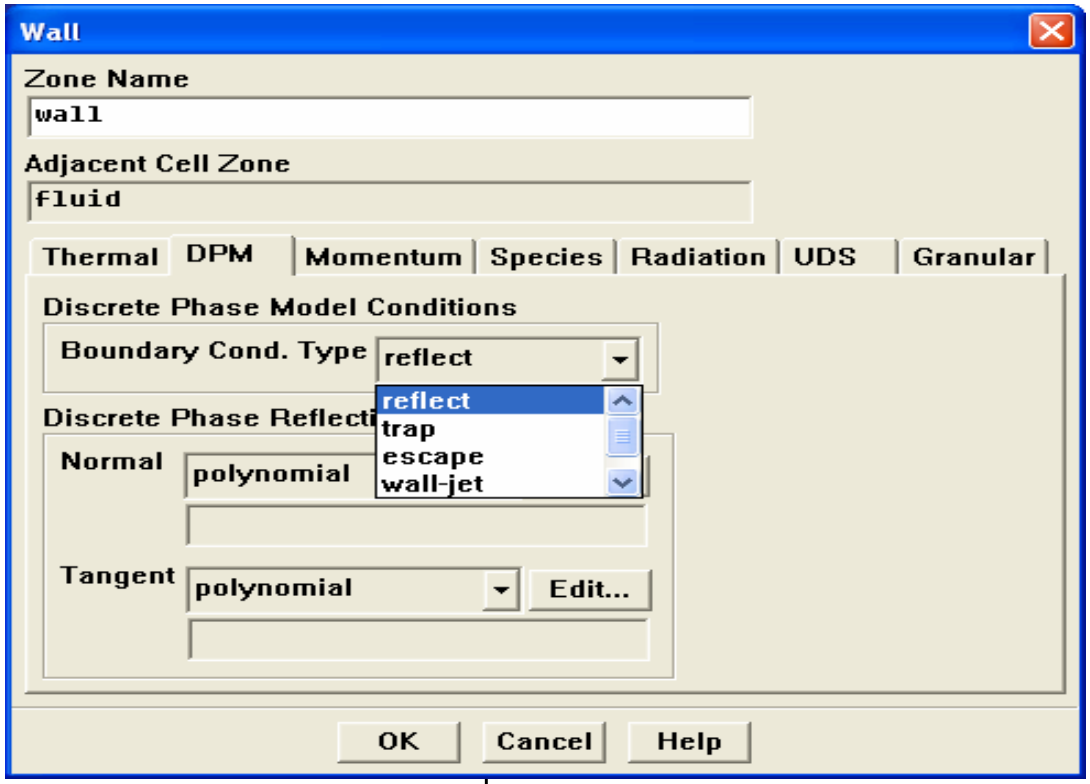


Trap

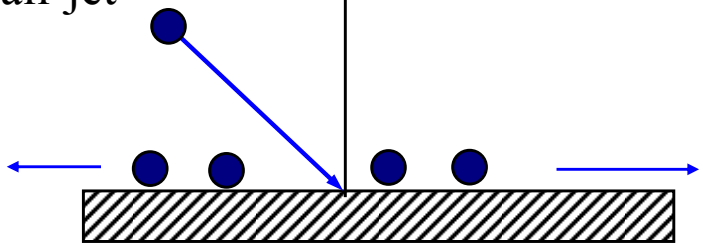


Reflect





Wall-jet



Mixture Model Equations

- Solves one equation for continuity of the mixture

$$\frac{\partial \rho_m}{\partial t} + \nabla \cdot (\rho_m \vec{u}_m) = \dot{m}$$

- Solves for the transport of volume fraction of each secondary phase

$$\frac{\partial}{\partial t}(\alpha_k \rho_k) + \nabla \cdot (\alpha_k \rho_k \vec{u}_m) = -\nabla \cdot (\alpha_k \rho_k \vec{u}_k^r)$$

"drift velocity"

$$\vec{u}_k^r = \vec{u}_k - \vec{u}_m$$

- Solves one equation for the momentum of the mixture

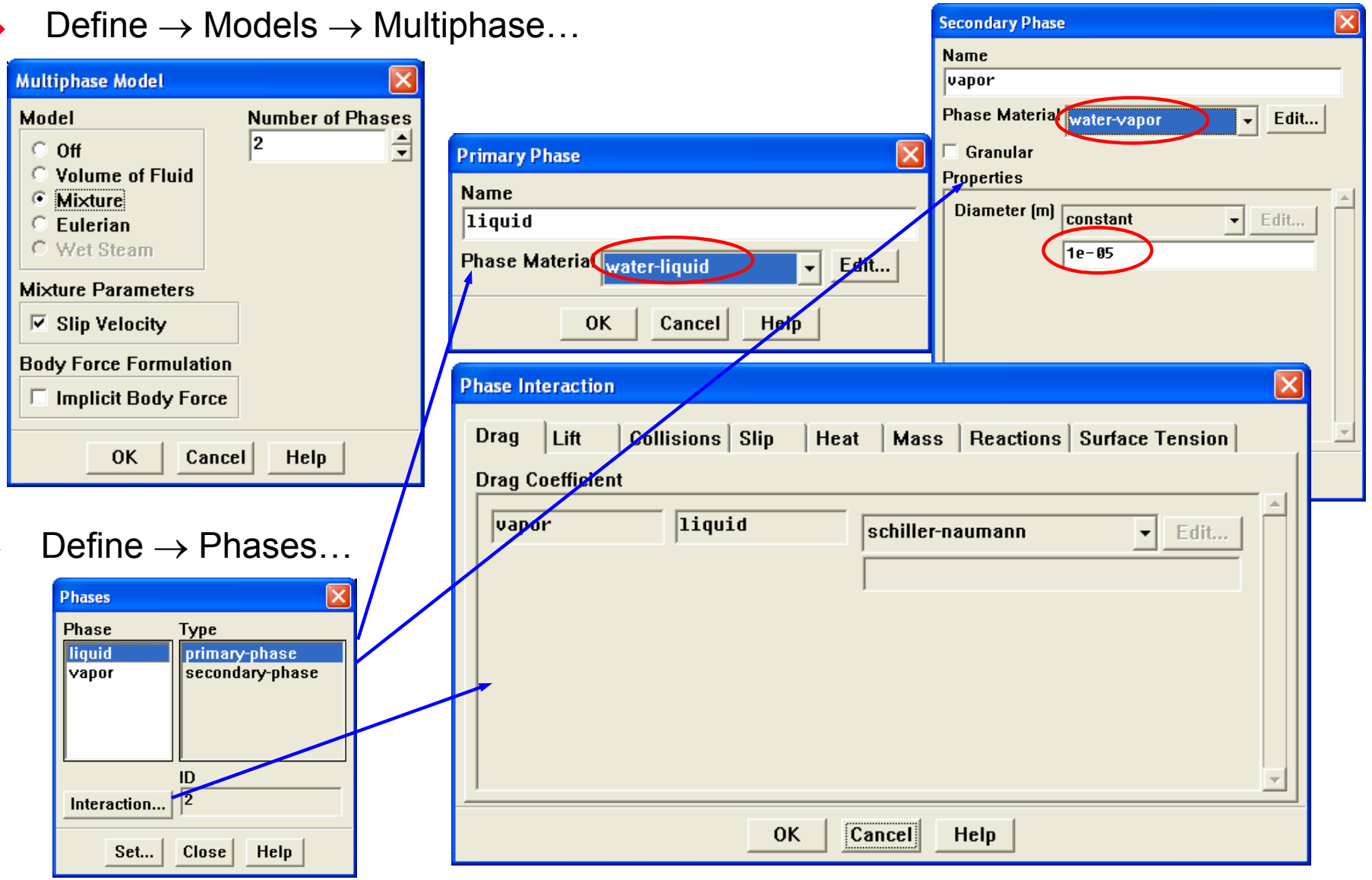
$$\frac{\partial}{\partial t}(\rho \vec{u}_m) + \nabla \cdot (\rho_m \vec{u}_m \vec{u}_m) = -\nabla p + \nabla \cdot \left[\mu_m (\nabla \vec{u}_m + \nabla \vec{u}_m^T)\right] + \rho_m \vec{g} + \vec{F} + \nabla \cdot \left(\sum_{k=1}^{n} \alpha_k \rho_k \vec{u}_k^r \vec{u}_k^r \right)$$

- Where mixture properties are defined as:

$$\rho_m = \sum_{k=1}^{n} \alpha_k \rho_k \qquad \mu_m = \sum_{k=1}^{n} \alpha_k \mu_k \qquad \vec{u}_m = \frac{\sum_{k=1}^{n} \alpha_k \rho_k \vec{u}_k}{\rho_m}$$

Mixture Model Setup (1)

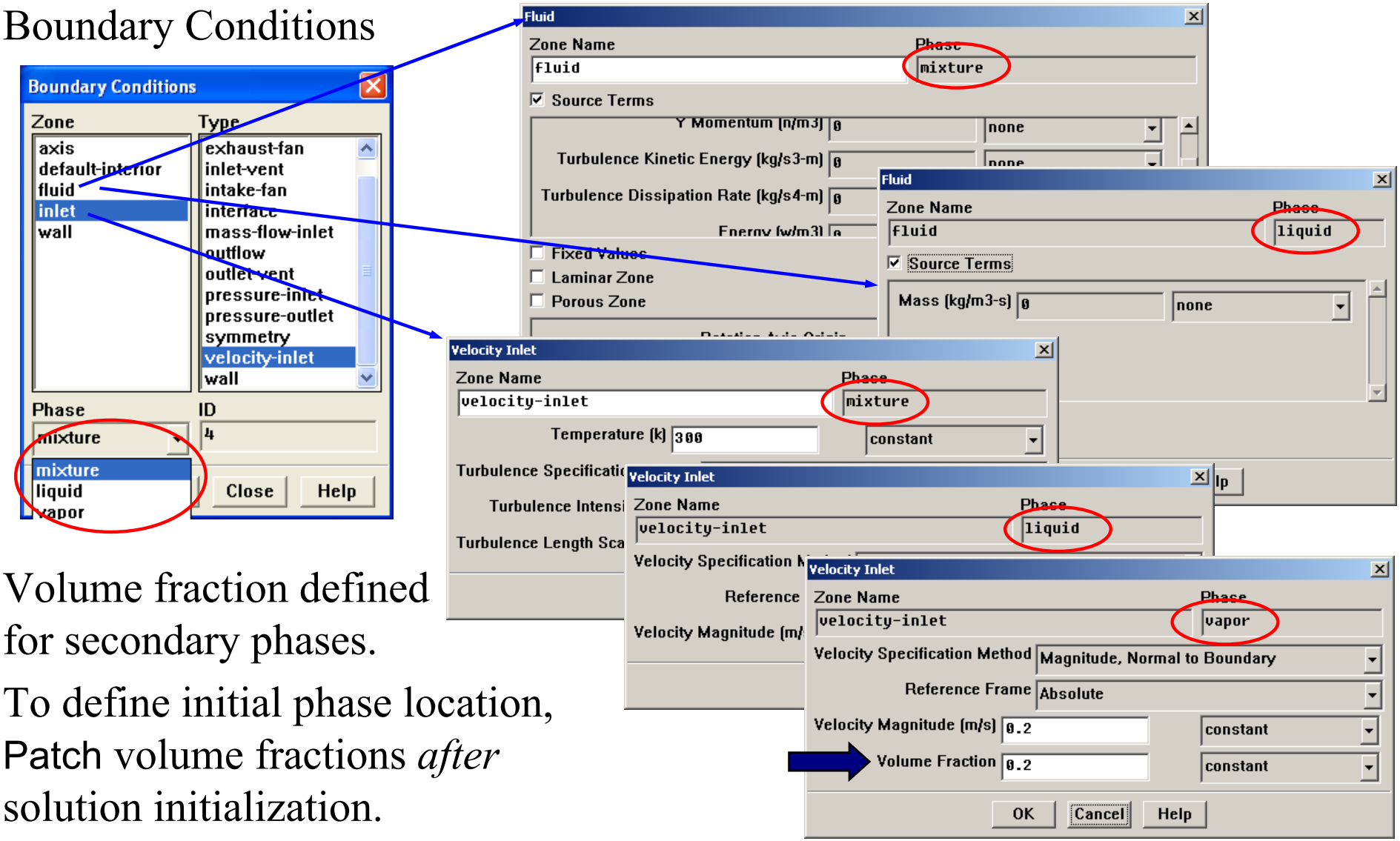
- Define → Models → Multiphase…



- Define → Phases…

Mixture Model Setup (2)

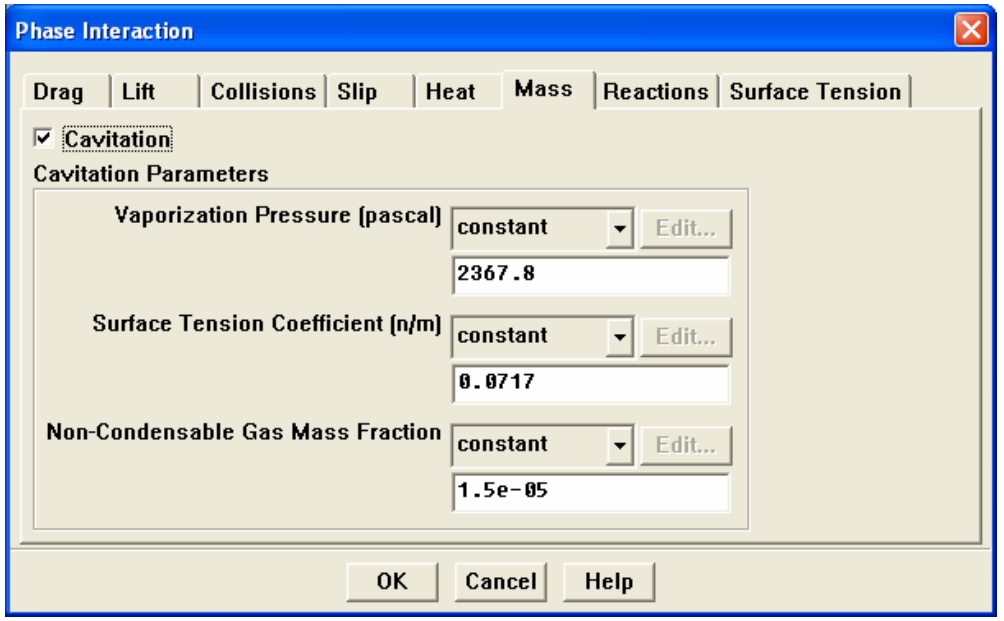
- Boundary Conditions



- Volume fraction defined for secondary phases.
- To define initial phase location, Patch volume fractions *after* solution initialization.

Cavitation Submodel

- Description
  - The Cavitation model models the formation of bubbles when liquid's local pressure becomes less than the vapor pressure.
  - The effect of non-condensable gases is included.
  - The mass conservation equation for vapor bubble
  - phase includes vapor generation and condensation terms which depend of sign of difference between local pressure and vapor saturation pressure corrected for non-condensable gas presence
  - Used with the Mixture model (but incompatible with VOF).
  - Tutorial is available for learning the in-depth setup procedures.



Eulerian Multiphase Model Equations

- Continuity:

Volume fraction for the q^{th} phase

$$\frac{\partial}{\partial t}\alpha_q\rho_q + \nabla\cdot\left(\alpha_q\rho_q\vec{u}_q\right) = \sum_{p=1}^{n}\dot{m}_{pq}$$

- Momentum for q^{th} phase:

$$\underbrace{\frac{\partial}{\partial t}\left(\alpha_q\rho_q\vec{u}_q\right)}_{\text{transient}} + \underbrace{\nabla\cdot\left(\alpha_q\rho_q\vec{u}_q\vec{u}_q\right)}_{\text{convective}} = \underbrace{-\alpha_q\nabla p}_{\text{pressure}} + \underbrace{\alpha_q\rho_q\vec{g}}_{\text{body}} + \underbrace{\nabla\cdot\bar{\bar{\tau}}_q}_{\text{shear}} + \underbrace{\sum_{p=1}^{n}\left(\vec{R}_{pq} + \dot{m}_{pq}\vec{u}_q\right)}_{\substack{\text{interphase forces exchange} \\ \text{interphase mass exchange}}} + \underbrace{\alpha_q\rho_q\left(\vec{F}_q + \vec{F}_{lift,q} + \vec{F}_{vm,q}\right)}_{\text{external, lift, and virtual mass forces}}$$

Solids pressure term is included for granular model.

- The inter-phase exchange forces are expressed as: $\vec{R}_{pq} = K_{pq}\left(\vec{u}_p - \vec{u}_q\right)$

  In general: $\vec{F}_{pq} = -\vec{F}_{qp}$

- Energy equation for q^{th} phase can be shown similarly.

fluid-fluid exchange coefficient

Eulerian Multiphase Model Equations

- Multiphase species transport for species *i* belonging to mixture of q^{th} phase:

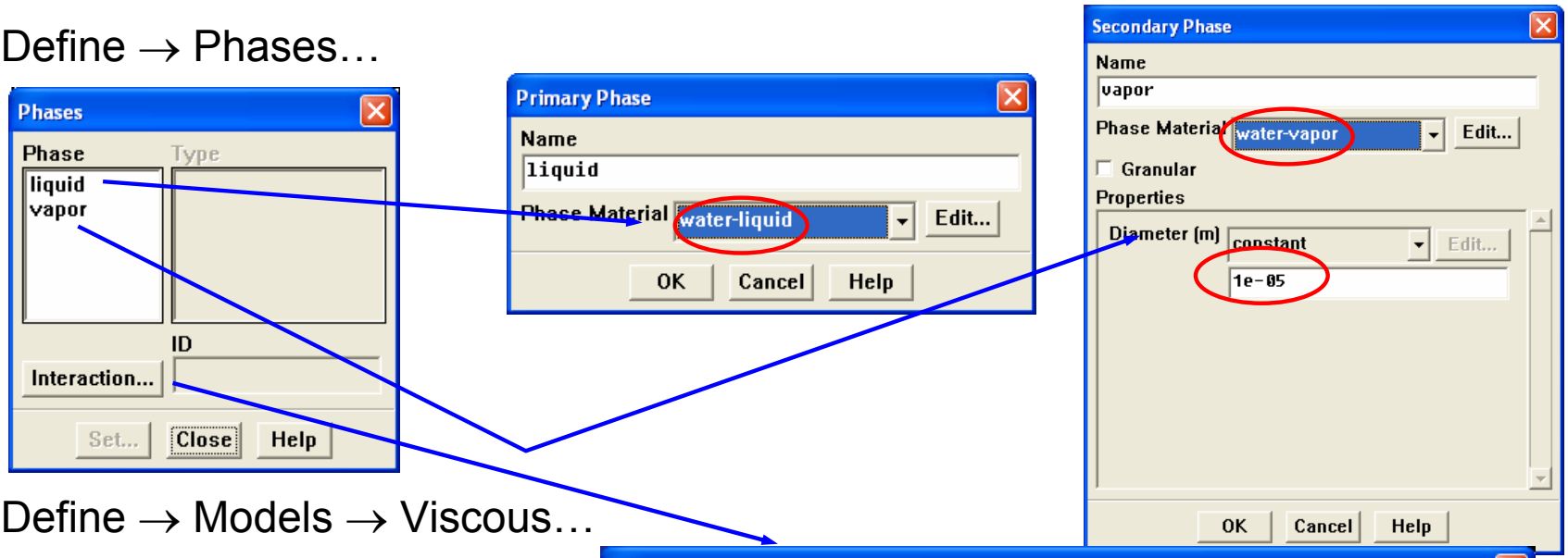
Mass fraction of species *i* in q^{th} phase

$$\underbrace{\frac{\partial}{\partial t}\left(\alpha^q \rho^q Y_i^q\right)}_{\text{transient}} + \underbrace{\nabla \cdot \left(\alpha^q \rho^q \vec{u}^q Y_i^q\right)}_{\text{convective}} = \underbrace{-\nabla \cdot \alpha^q \vec{J}_i^q}_{\text{diffusion}} + \underbrace{\alpha^q R_i^q}_{\text{homogeneous reaction}} + \underbrace{\alpha^q S_i^q}_{\text{homogeneous production}} + \underbrace{\sum_{p=1}^{n}\left(\dot{m}_{p^i q^j} - \dot{m}_{q^j p^i}\right)}_{\text{heterogeneous reaction}}$$

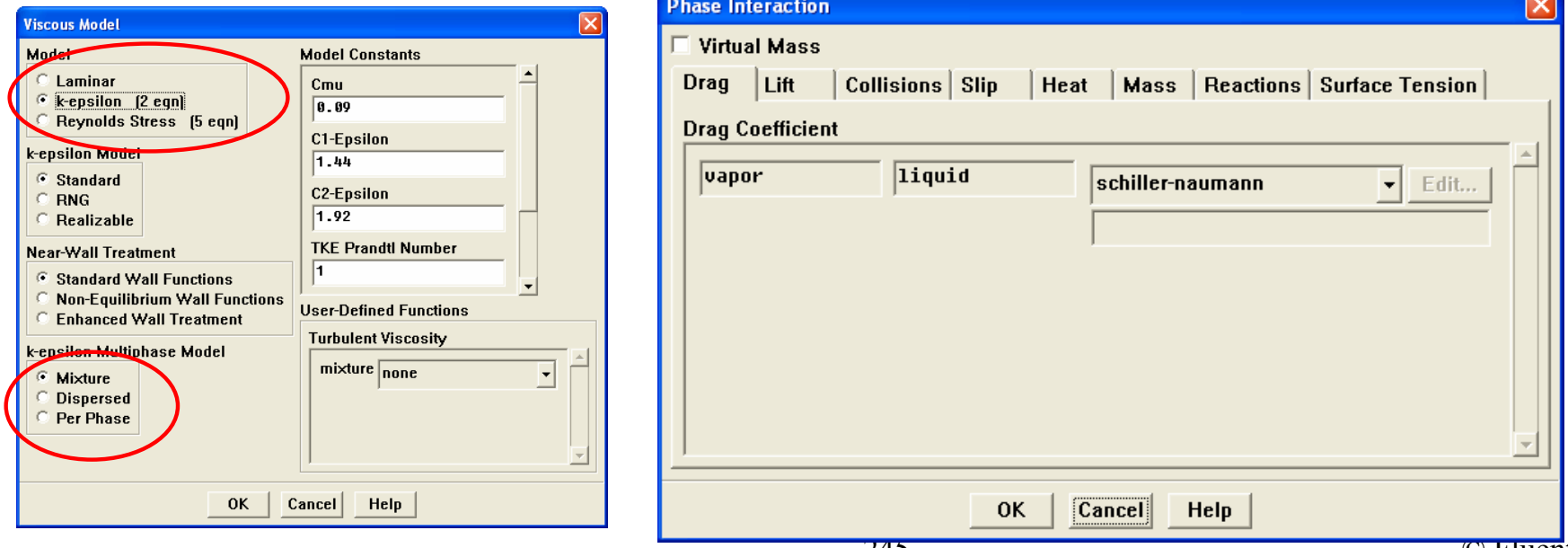
- Homogeneous and heterogeneous reactions are setup the same as in single phase
- The same species may belong to different phases without any relation between themselves

Eulerian Model Setup (1)

- Define → Phases…

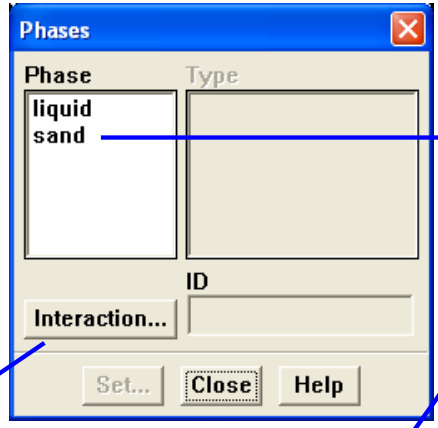


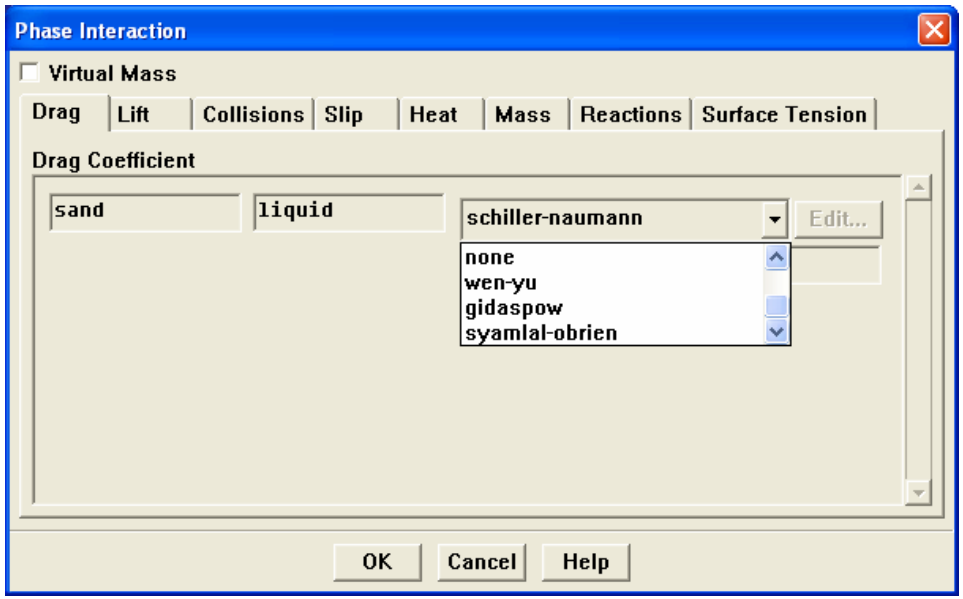
- Define → Models → Viscous…

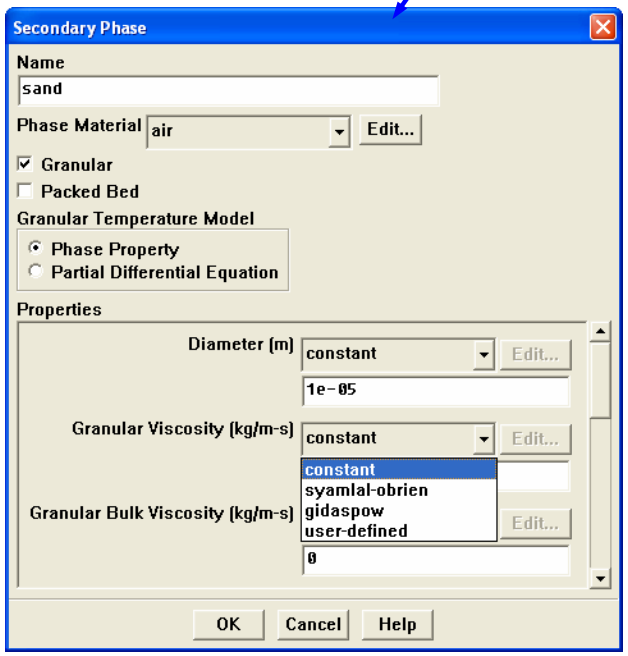


Eulerian-Granular Model Setup

- Granular option must be enabled when defining the secondary phases.
- Granular properties require definition.
- Phase interaction models appropriate for granular flows must be selected.

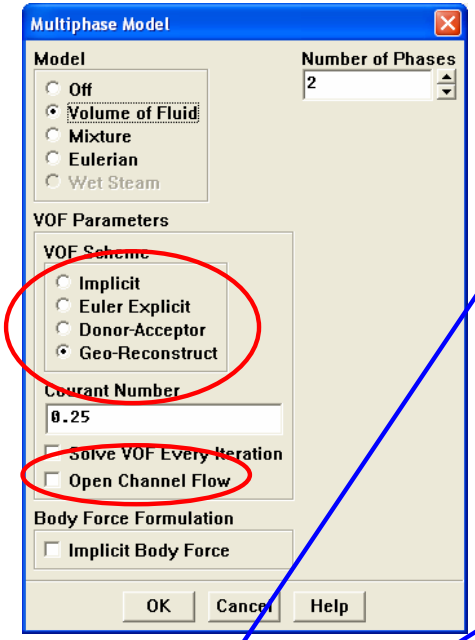


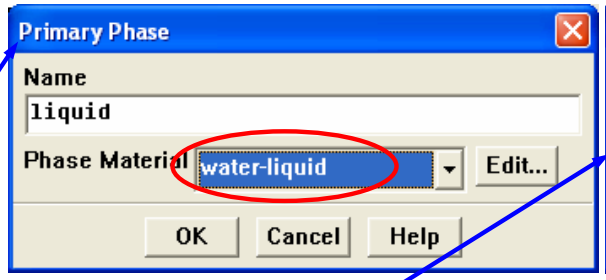


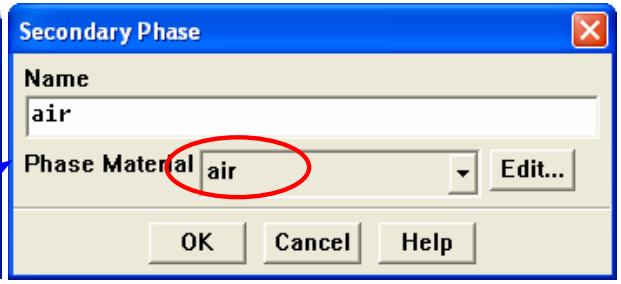


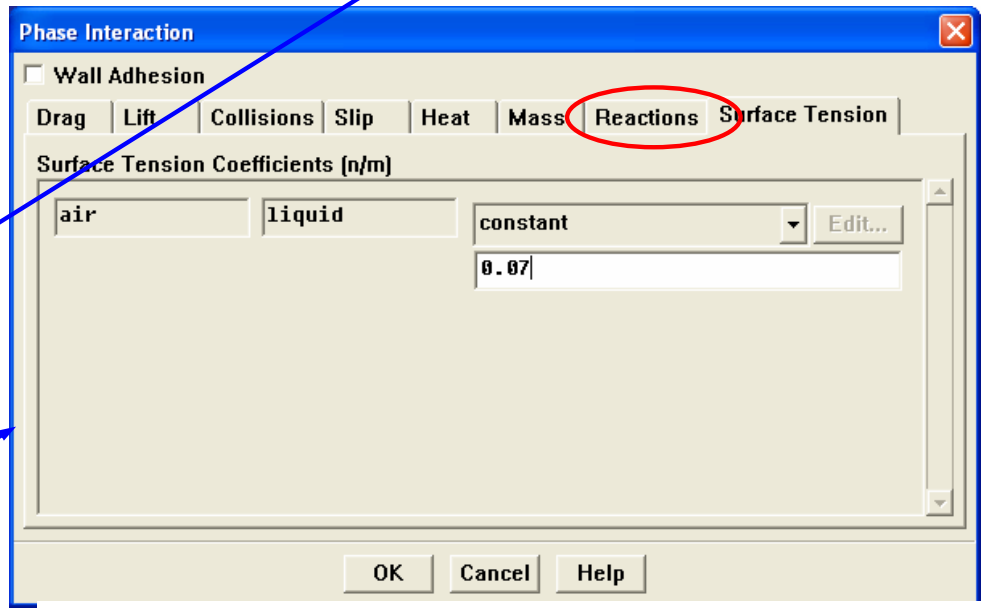
VOF Model Setup

- Define → Models → Multiphase…

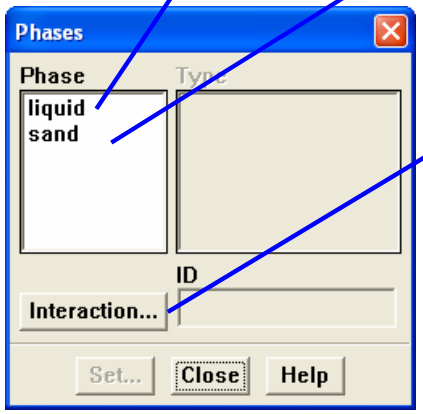








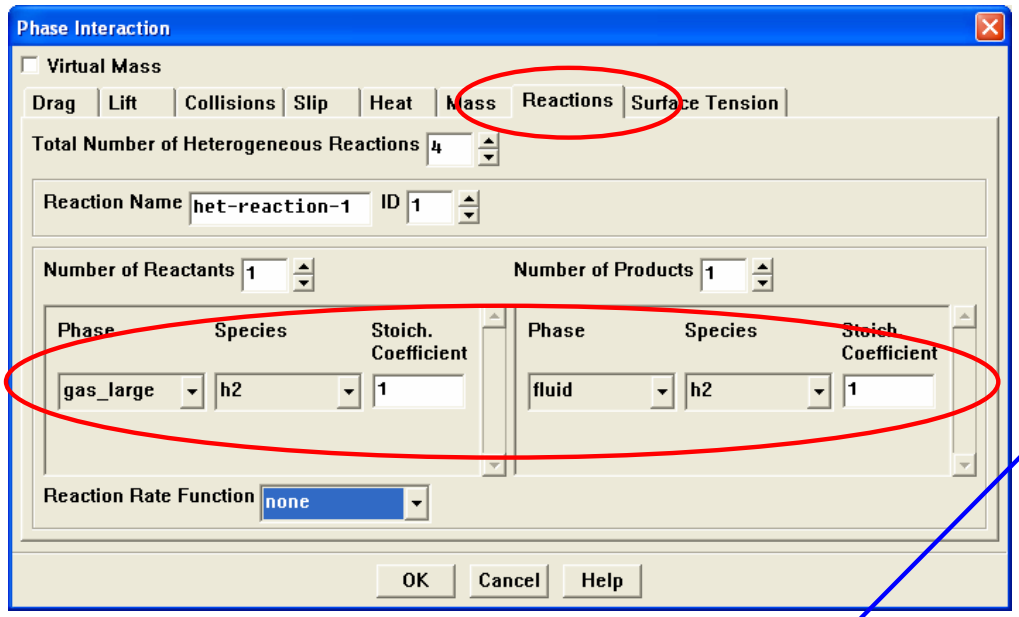
- Define → Phases…

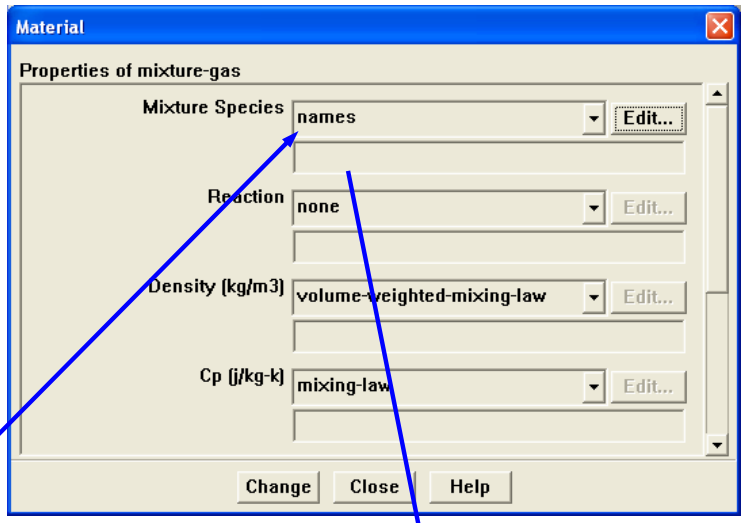


- Define → Operating Conditions…
  - Operating Density should be set to that of lightest phase with body forces enabled.

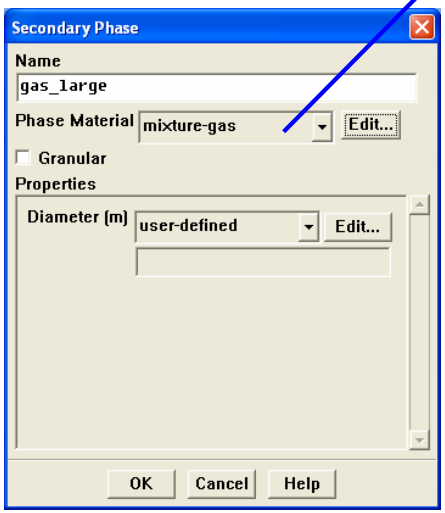
Heterogeneous reaction setup

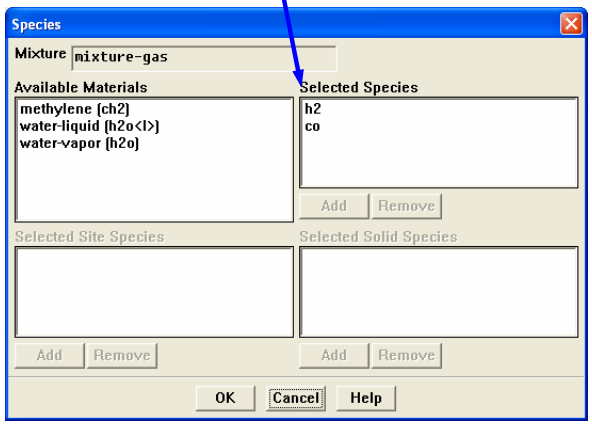
- Define → Phases → Interaction…





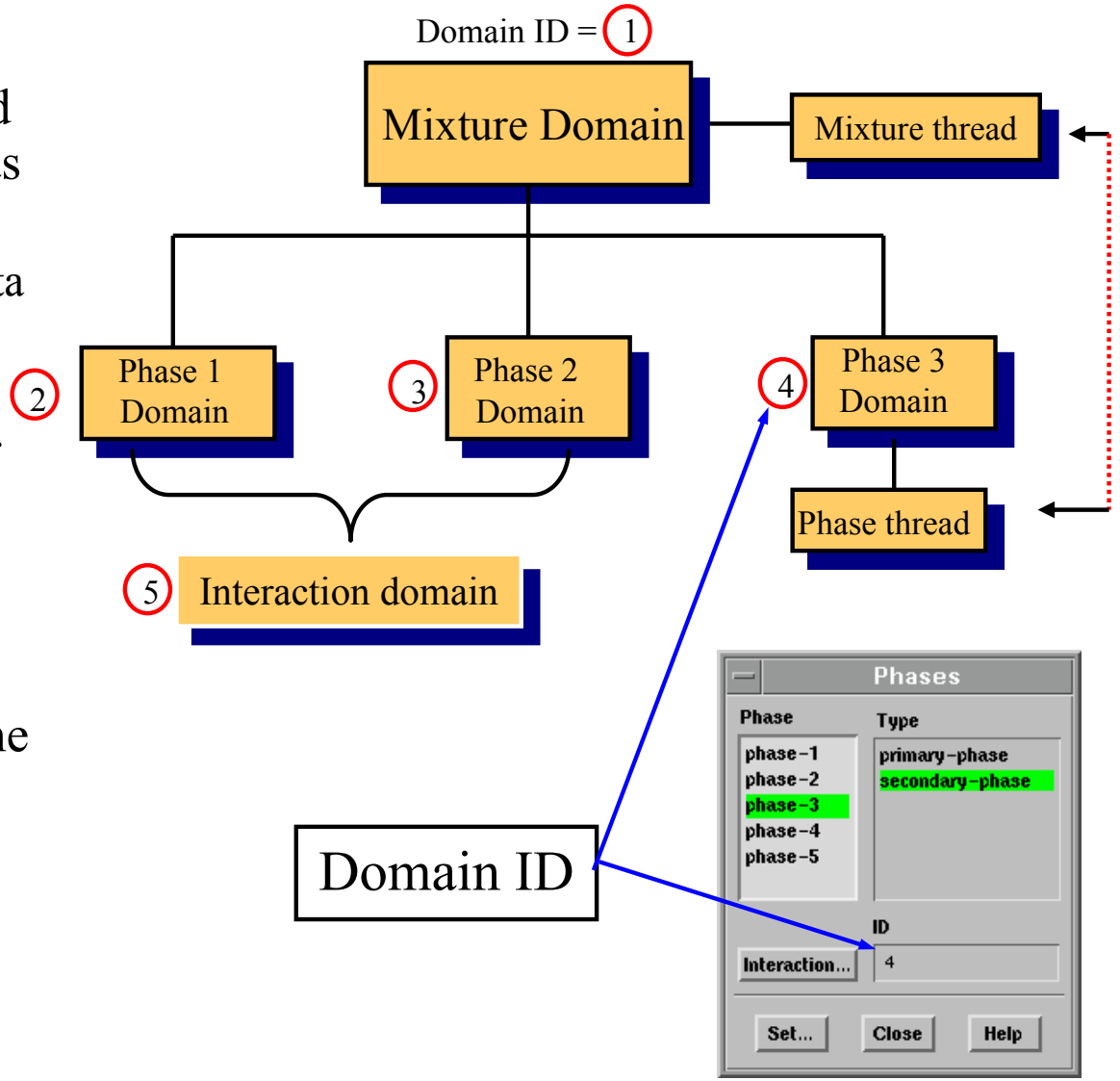
- Define → Phases…





UDFs for Multiphase Applications

- When a multiphase model is enabled, storage for properties and variables is set aside for mixture as well as for individual phases.
  - Additional thread and domain data structures required.
- In general the type of `DEFINE` macro determines which thread or domain (mixture or phase) gets passed to your UDF.
- *e.g.* `C_R(cell,thread)` will return the *mixture* density if `thread` is the *mixture* thread or the *phase* densities if it is the *phase* thread.
- Numerous macros exist for data retrieval.

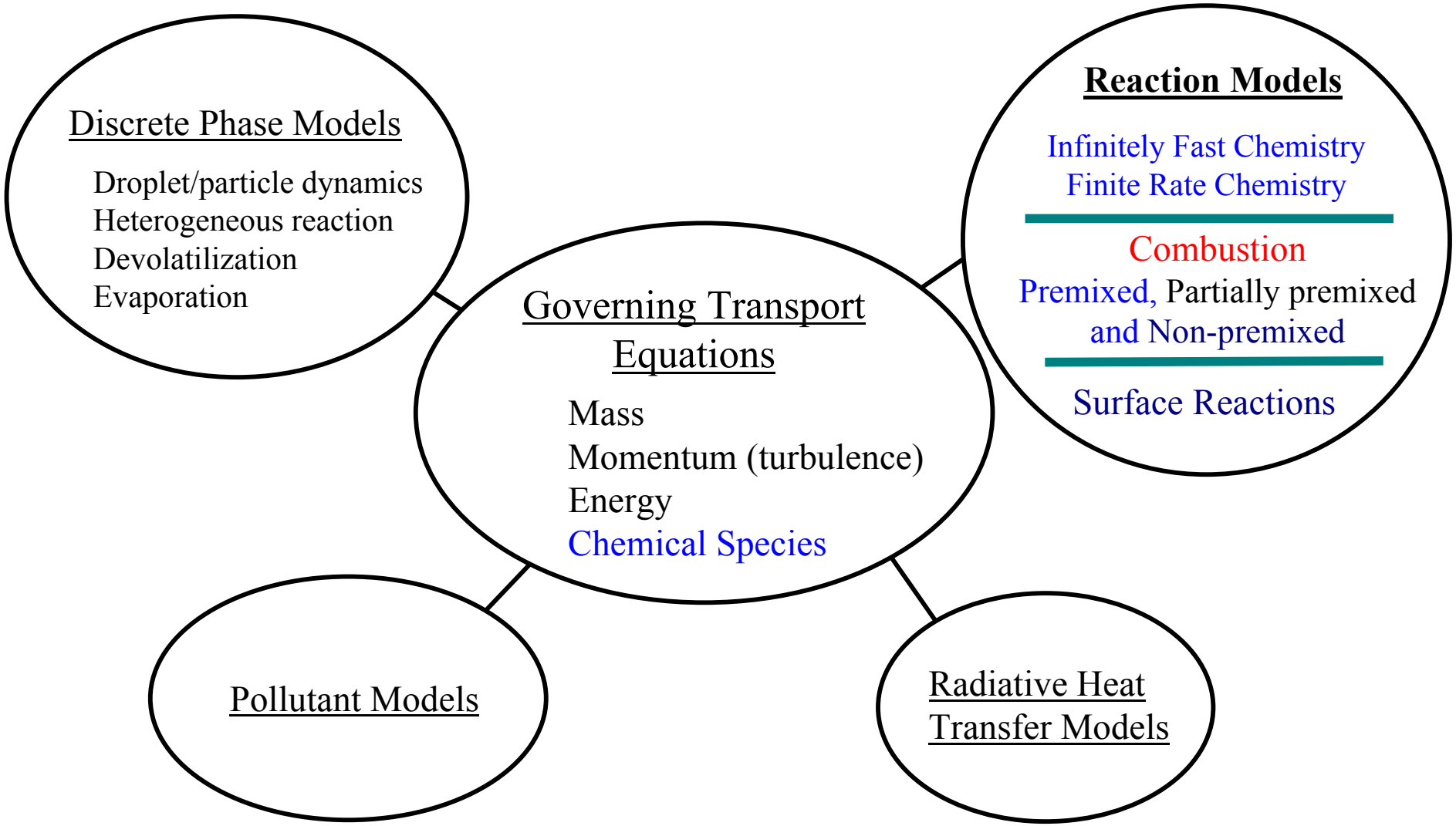


Reacting-Flow Models in FLUENT

Outline

- **Aspects of Reaction Modeling and Fundamental Concepts**
- Reaction Models in FLUENT
  - Overview
  - Fast Chemistry Models: eddy dissipation, premixed, partially premixed, non-premixed
  - Moderately non-equilibrium model: Laminar flamelet for non-premixed flames
  - Finite-rate Chemistry Models: Laminar finite-rate, eddy dissipation concept (EDC), composition PDF
- ISAT Algorithm, DPM and Pollutant Formation Models
- Surface Reactions
- Examples and Case Studies

Aspects of Reaction Modeling



Scale Analysis

- Reynolds number

$$\mathrm{Re} = \left(\frac{\rho U L}{\mu} \right) \sim \frac{\text{inertial force}}{\text{viscous force}}$$

ρ, U, L, μ are characteristic (e.g. inlet) density, velocity, length and dynamic viscosity, respectively

Turbulent reacting flow models are valid at high Reynolds numbers

- Damköhler Number

$$\mathrm{Da} = \frac{\text{characteristic flow time}}{\text{characteristic chemical time}}$$

When chemical reaction rates are fast in comparison with fluid mixing rates, Da >> 1. Conversely, when reaction rates are slow relative to fluid mixing rates, Da << 1.

Chemical Kinetics

- The *k-th* species mass fraction transport equation is:

$$\frac{\partial}{\partial t}(\rho Y_k) + \frac{\partial}{\partial x_i}(\rho u_i Y_k) = \frac{\partial}{\partial x_i}\left(\rho D_k \frac{\partial Y_k}{\partial x_i}\right) + R_k$$

- Nomenclature: chemical species, denoted S_k, react as:

$$\sum_{k=1}^{N} \nu'_k S_k \rightarrow \sum_{k=1}^{N} \nu''_k S_k$$

Example: $CH_4 + 2O_2 \rightarrow CO_2 + 2H_2O$

$$S_1 = CH_4 \quad S_2 = O_2 \quad S_3 = CO_2 \quad S_4 = H_2O$$
$$\nu'_1 = 1 \quad \nu'_2 = 2 \quad \nu'_3 = 0 \quad \nu'_4 = 0$$
$$\nu''_1 = 0 \quad \nu''_2 = 0 \quad \nu''_3 = 1 \quad \nu''_4 = 2$$

Chemical Kinetics (cont'd)

- The calculated reaction rate is proportional to the products of the reactant concentrations raised to the power of their respective stoichiometric coefficients.
- *K-th* species reaction rate (for a single reaction):

$$R_k = M_k (\nu''_k - \nu'_k) \left(A T^{\beta} e^{-\frac{E}{RT}} \right) \prod_{j=1}^{N} C_j^{\nu'^*_k}$$

A = pre-exponential factor
C_j = molar concentration = $\rho Y_j / M_j$
M_k = molecular weight of species k
E = activation energy
R = universal gas constant = 8314 J / (kgmol K)
β = temperature exponent

- Note that for global reactions, $\nu'^*_k \neq \nu'_k$, and may be non-integer

Reynolds (Time) Averaged Species Equation

$$\left\{\frac{\partial}{\partial t}\left(\overline{\rho}\,\overline{Y}_k\right)+\right\}\frac{\partial}{\partial x_i}\left(\overline{\rho}\,\overline{u}_i\,\overline{Y}_k\right)+\frac{\partial}{\partial x_i}\left(\overline{\rho u''_i\,Y''_k}\right)=\frac{\partial}{\partial x_i}\left(\overline{\rho D_k}\frac{\partial \overline{Y}_k}{\partial x_i}\right)+\overline{R}_k$$

{unsteady term} (zero for steady flows)

convection by mean velocity

convection by turbulent velocity fluctuations

molecular diffusion

mean chemical source term

- Y_k, D_k, R_k are the *k-th* species mass fraction, diffusion coefficient and chemical source term, respectively
- Turbulent flux term modeled by mean gradient diffusion as $\overline{\rho u''_i\,Y''_k} = \mu_t/Sc_t \cdot \partial \overline{Y}_k/\partial x_i$, which is consistent in the k-ε context
- Gas-phase combustion modeling focuses on $\overline{R}_k$
  - Arguably more difficult to model than the Reynolds stresses (in turbulence modeling)

Difficulties of Modeling Reactive Flows

- Solving *directly* the species transport and the Arrhenius rate equations for a reacting flow problem is difficult and impractical:
  - **Turbulence**
    - Most industrial flows are turbulent.
    - DNS of non-reacting and reacting turbulent flows is not possible because of the wide range of time and length scales.
  - **Chemistry**
    - Realistic chemical mechanisms cannot be described by a single reaction equation.
      - Tens of species, hundreds of reactions
      - Known in detail for only a limited number of fuels
    - Stiff kinetics (wide range of reacting time scales)
  - **Turbulence-Chemistry Interaction**
    - The sensitivity of reaction rates to local changes is complicated by enhanced mixing of turbulent flows.

Modeling Chemistry

- Simplifications in the chemistry
  - Resort to global chemical mechanisms
    - Example: complex methane combustion mechanism
    - Species: **CH4 CH3 H O2 HO2 H2 O OH H2O CH2O HCO CO CO2 CH3O N2 H2O2 N**
      - 46-step reactions can be approximated by 2-step global reactions:

…CH3+O2=CH2O+OH, CH3+O=CH2O+H, CH3+OH=CH2O+H2, HO2+CO=CO2+OH, H2+O2=2OH, OH+H2=H2O+H, H+O2=OH+O O+H2=OH+H… } **CH4+O2 =CO+H2O**
2CO+O2=2CO2

  - Assume infinitely fast chemistry
    - Reaction rates are controlled by turbulent mixing rates (mixing limited) which can be estimated from turbulence properties
    - Avoid modeling complex reaction rates and assume chemical equilibrium
- Stiff chemistry
  - Use very small time steps to solve problems (impractical).
  - Use the **stiff solver** that enables larger time steps to be used.
    - Still computationally expensive

Turbulence-Chemistry Interaction

- Arrhenius reaction-rate terms are highly nonlinear

$$R_k = AT^{\beta} \prod_j C_j^{\nu_j} \exp\left(-E/RT\right)$$

- Cannot neglect the effects of turbulence fluctuations on chemical production rates:

$$\overline{R}_k \neq R_k(\overline{T})$$

Gas-Phase Reacting Flows: Configurations

- *Non-Premixed* combustion
  - Fuel and oxidizer enter the reaction zone in separate streams
  - Can be simplified to a mixing problem (under certain restrictions)
  - Turbulent eddies distort the laminar flame shape and enhance mixing
  - Examples: pulverized coal furnaces, diesel engine, pool fires, …
- *Premixed* combustion
  - Fuel and oxidizer are already mixed at the molecular level prior to ignition, a very thin flame front propagates into the unburnt regions
  - Rate of propagation (the laminar flame speed) depends on the internal flame structure and turbulence eddies, hence it is much more difficult to model than the non-premixed combustion
  - Examples: lean-premixed gas turbine combustors, internal combustion engines, …
- *Partially premixed* combustion
  - Reacting systems with both non-premixed and premixed fuel/oxidizer streams
  - Example: lean premixed combustors with diffusion pilot flames and/or cooling air jets, …

Outline

- Aspects of Reaction Modeling and Fundamental Concepts
- **Reaction Models in FLUENT**
  - **Overview**
  - Fast Chemistry Models: eddy dissipation, premixed, partially premixed, non-premixed
  - Moderately non-equilibrium model: Laminar flamelet for non-premixed flames
  - Finite-rate Chemistry Models: Laminar finite-rate, eddy dissipation concept (EDC), composition PDF
- ISAT Algorithm, DPM and Pollutant Formation Models
- Surface Reactions
- Examples and Case Studies

Reaction Models in Fluent

<table>
<tr><th>Flow config. / Chemistry</th><th>Premixed Combustion</th><th>Non-Premixed Combustion</th><th>Partially Premixed Combustion</th></tr>
<tr><td rowspan="2">Infinitely Fast Chemistry</td><td>Premixed Combustion Model
(Reaction Progress Variable)*</td><td>Non-Premixed Equilibrium Model
(Mixture Fraction + PrePDF)</td><td>Partially Premixed Model
(Reaction Progress Variable + Mixture Fraction)</td></tr>
<tr><td colspan="3">Eddy Dissipation Model</td></tr>
<tr><td rowspan="2">Finite-Rate Chemistry</td><td></td><td>Non-Premixed Laminar Flamelet Model</td><td></td></tr>
<tr><td colspan="3">Laminar Finite-Rate Model
Eddy-Dissipation Concept (EDC) Model
Composition PDF transport Model</td></tr>
</table>

* Rate classification not truly applicable since species mass fraction is not determined.

Homogeneous (Gas-Phase) Combustion Models

- **Eddy-dissipation**
  - Use 1 or 2-step global (heat-release) reaction mechanisms
  - The chemical reaction rate is the turbulent mixing rate
    - Finite-Rate option is a switch for the premixed flames
- **Non-premixed**
  - Use mixture fraction f and assumed PDF, instead of solving the equations of species transport and reacting rates for equilibrium chemistry
  - Laminar Flamelets for moderate non-equilibrium chemistry
- **Premixed**
  - Reduce chemistry to reaction progress variable c
  - Turbulent Flame Speed Closure: Zimont model
- **Partially premixed**
  - Combine non-premixed and premixed models
  - Assumptions in both models apply
- **Eddy-dissipation concept (EDC)**
  - A turbulence-chemistry interaction model for turbulent flames with detailed chemistry
- **Composition PDF transport**
  - More rigorous treatment of turbulence-chemistry interaction than EDC model, but it is substantially more expensive

Outline

- Aspects of Reaction Modeling and Fundamental Concepts
- Reaction Models in FLUENT
  - Overview
  - **Fast Chemistry Models: eddy dissipation, premixed, partially premixed, non-premixed**
  - Moderately non-equilibrium model: Laminar flamelet for non-premixed flames
  - Finite-rate Chemistry Models: Laminar finite-rate, eddy dissipation concept (EDC), composition PDF
- ISAT Algorithm, DPM and Pollutant Formation Models
- Surface Reactions
- Examples and Case Studies

Eddy Dissipation Model (EDM)

- A model for estimating the mean reaction rate (of species k), R_k
- Reaction rate is defined using the eddy-breakup model:
  - Assumes that chemical reactions occur much faster than turbulence can mix reactants into the flame (Da >> 1)
  - Combustion is completely controlled by turbulent mixing rates which are proportional to the large-eddy lifetime scale, k/ε
- Chemical reaction is approximated by the *global* (1 or 2 step) mechanism
- Reynolds (time) averaged species mass fraction equations for (N-1) species are solved
- Finite-Rate/Eddy Dissipation Option: the smaller of the Arrhenius and eddy-dissipation rates is used as the reaction rate

Eddy Dissipation Model (cont'd)

- Applicability:

  Flow Regime: Turbulent flow (high Re)

  Chemistry: Fast Chemistry (Da >> 1)

  Flow configuration : Premixed, non-premixed, partially premixed

- Applications
  - Widely used for chemical reacting flows at high Da and Re
  - Example: BERL combustor, IFRF coal combustion
- Limitations
  - Unreliable when flow (mixing) and kinetic time scales are comparable (Da ~1)
  - Does not predict kinetically-controlled intermediate species and dissociation effects
  - Cannot realistically model phenomena which depend on detailed kinetics such as ignition, extinction and low-Da flows

Premixed Combustion Model

- In premixed system combustion within the domain occurs as a thin flame front that propagates into the unburnt region of reactants
  - Flame front propagates at a speed dictated by laminar flame speed and local turbulence eddies
  - A reaction progress variable c is solved to predict the position of the flame front (Zimont model)
    - c is defined such that $c = 0$ for unburnt mixture and $c = 1$ for burnt mixture.
    - For non-adiabatic flows, temperature is determined from local enthalpy which is calculated from enthalpy transport equation.
- Applicability:
  - Flow regime: turbulent
  - Chemistry: infinitely fast chemistry
- Application
  - Lean-premixed gas turbine combustor

Partially Premixed Combustion Model

- Reaction progress variable and mixture fraction approach are combined
  - Solves transport equations for reaction progress variable c, mixture fraction f, and its variance.
  - The premixed reaction-progress variable, c, determines the position of the flame front.
    - Behind the flame front (c = 1), the mixture is burnt and the equilibrium or laminar flamelet mixture fraction solution is used.
    - Ahead of the flame front (c = 0), the species mass fractions, temperature, and density are calculated from the mixed but unburnt mixture fraction.
    - Within the flame (0 < c <1), a linear combination of the unburnt and burnt mixtures is used.
- Applicability of the Premixed combustion model
  - Flow regime: turbulent
  - Chemistry: Equilibrium or moderate non-equilibrium (Flamelet)
  - Flow configuration : partially premixed
- Application
  - Systems with both premixed and non-premixed streams

Non-Premixed Equilibrium Model

- For infinitely fast chemistry in turbulent flows, reaction proceeds to chemical equilibrium instantaneously when the fuel and oxidizer mix (mixing limited).
- The species and enthalpy transport equations collapse into a single conservation equation for mixture fraction f, under the following assumptions
  - Species diffusion coefficients are equal
  - Lewis number (Le) = 1
  - Moderate Mach number
- Mixture fraction can be directly related to species mass fraction, mixture density, and mixture temperature
  - These relations are known for instantaneous (not time-averaged) data
- A probability density function (PDF) is used to relate instantaneous data to time-averaged data (turbulent reacting flow simulations predict *time-averaged* properties)
  - PDF is assumed to be a beta function (which can describe a wide range of PDF shapes)
  - Requires the variance of f which is based on local turbulence parameters

Non-Premixed Equilibrium Model (cont'd)

- Transport equations for mixture fraction and mixture fraction variance are solved (instead of the individual species equations), reaction is simplified as a mixing problem
- Applicability:

Flow regime: Turbulent flow

Chemistry: Fast chemistry (assuming equilibrium condition)

Flow configuration: Non-premixed

- Applications
  - Model of choice if assumptions (see the previous slide) are valid for gas-phase combustion problems
  - Accurate tracking of intermediate species concentrations and dissociation effects *without* requiring the knowledge of detailed chemical kinetic rates
  - Turbulence-chemistry interactions are accounted for rigorously via PDFs
- Limitations
  - Unreliable when mixing and kinetic time scales are comparable (Da ~1)
  - Cannot realistically model phenomena which depend on detailed kinetics such as ignition, extinction and low-Da flows

Outline

- Aspects of Reaction Modeling and Fundamental Concepts
- Reaction Models in FLUENT
  - Overview
  - Fast Chemistry Models: eddy dissipation, premixed, partially premixed, non-premixed
  - **Moderately non-equilibrium model: Laminar flamelet for non-premixed flames**
  - Finite-rate Chemistry Models: Laminar finite-rate, eddy dissipation concept (EDC), composition PDF
- ISAT Algorithm, DPM and Pollutant Formation Models
- Surface Reactions
- Examples and Case Studies

Laminar Flamelet Model For Non-premixed Flames

- Extension of mixture fraction/PDF model to problems in where moderate chemical non-equilibrium effects are important
- Turbulent diffusion flame is modeled as an ensemble of stretched laminar flamlets (opposed flow diffusion flames)
  - Temperature, density and species (for adiabatic) specified by two parameters, the mixture fraction and scalar dissipation rate χ (flame strain rate).
- Applicability:

  Flow regime: Turbulent flow

  Chemistry: Moderate non-equilibrium due to aerodynamic straining, *not* suitable for slow chemistry

  Flow configuration: Non-premixed
- Applications
  - Predictions of lift-off and blow-off phenomena in jet flames

Outline

- Aspects of Reaction Modeling and Fundamental Concepts
- **Reaction Models in FLUENT**
  - Overview
  - Fast Chemistry Models: eddy dissipation, premixed, partially premixed, non-premixed
  - Moderately non-equilibrium model: Laminar flamelet for non-premixed flames
  - **Finite-rate Chemistry Models: Laminar finite-rate, eddy dissipation concept (EDC), composition PDF**
- ISAT Algorithm, DPM and Pollutant Formation Models
- Surface Reactions
- Examples and Case Studies

Detailed Chemistry Models

- For reactions described by detailed kinetic mechanisms, the chemistry models are numerically stiff : tens of species, hundreds of reactions with large spread in reaction time scales
- CHEMKIN-format reaction mechanisms and thermal properties can be imported directly
- Available Models:
  - Laminar finite rate
  - EDC Model
  - PDF transport
  - KINetics model
- Detailed kinetics are used to model:
  - Flame ignition and extinction
  - Pollutants (NOx, CO, UHCs)
  - Slow (non-equilibrium) chemistry
- CPU intensive, ISAT (In-situ Adaptive Tabulation) algorithm can be used for acceleration (Laminar, EDC, PDF transport models)

Laminar Finite-Rate Model

- Stiff chemistry solver (available for both coupled and segregated solver)
- Applicability of the finite-rate model
  - Flow regime: Laminar flow
  - Chemistry: Finite-rate chemistry
  - Flow configuration : Premixed, non-premixed, partially premixed
- Deficiencies
  - CPU intensive: ISAT (In-situ Adaptive Tabulation) algorithm for acceleration
  - Unsuitable for turbulent combustion problems
- Applications
  - Laminar diffusion flame, laminar premixed flames in gas furnaces, etc.

Eddy Dissipation Concept (EDC) Model

- EDC model is an extension of the eddy-dissipation model to account for finite-rate chemistry and detailed mechanisms in turbulent flows.
- EDC is a turbulence-chemistry interaction model and species transport equation is solved for each species
- Applicability of the EDC model
  - Flow Regime: Turbulent flow
  - Chemistry: Finite-rate chemistry
  - Flow configuration : Premixed, non-premixed, partially premixed
- Deficiencies
  - CPU intensive. ISAT (In-situ Adaptive Tabulation) algorithm can be used for acceleration
- Applications
  - Predictions of finite-rate phenomena in turbulent reacting flows, for example, slow CO burnout or nitrogen oxides formation

Composition PDF Transport model

- The PDF (probability density function) transport model is used to incorporate finite-rate chemistry in turbulent flames (and is more rigorous than the EDC model)
- The PDF represents the fraction of time that the fluid spends at each state
- The mean reaction rate can be calculated from the PDF as:

$$\overline{\dot{w}_k} = \int_0^\infty \int_0^1 \cdots \int_0^1 \dot{w}_k \, P \, dY_N \cdots dY_1 \, dT$$

- PDF transport equation has very high dimensionality and cannot be solved by finite-volume method. Monte Carlo Method is used in solving the PDF transport equation.
- Applicability of the PDF transport model
  - Flow Regime: Turbulent flow
  - Chemistry: Finite-rate chemistry
  - Flow configuration : Premixed, non-premixed, partially premixed
- Deficiency
  - Very CPU intensive (ISAT algorithm can be employed to accelerate the computation)
- Applications
  - Predictions of finite-rate phenomena in turbulent reacting/combusting flows, like CO, NOx

Outline

- Aspects of Reaction Modeling and Fundamental Concepts
- Reaction Models in FLUENT
  - Overview
  - Fast Chemistry Models: eddy dissipation, premixed, partially premixed, non-premixed
  - Moderately non-equilibrium model: Laminar flamelet for non-premixed flames
  - Finite-rate Chemistry Models: Laminar finite-rate, eddy dissipation concept (EDC), composition PDF
- **ISAT Algorithm, DPM and Pollutant Formation Models**
- Surface Reactions
- Examples and Case Studies

In-Situ Adaptive Tabulation (ISAT) Algorithm

- Motivations
  - In EDC, Composition PDF, and laminar finite rate models, the stiff reaction system (a set of ODE's) is solved for each cell (or for each particle) in every iteration.
  - Table look-up method is effective, but the traditional *prior* tabulation approach is not feasible for a system with a large number of species; moreover, it is inefficient since only a small part of the composition space is accessed during a simulation.
- In-Situ Adaptive Tabulation
  - Table is constructed in situ - as the reactive flow calculation is performed, so that only the accessed region of the composition space is tabulated.
  - Tabulation error is controlled adaptively by the user specified tolerance.
- Advantage
  - Avoid the repeating work in numerical integration, and reduce cpu time significantly.
    - Initial iterations are slower as table is being built, but accelerate as more retrieval operations take place.
- Application
  - ISAT can be used with PDF-transport, EDC, and laminar finite-rate models

Discrete Phase Model (DPM)

- Description
  - Trajectories of particles/droplets/bubbles are computed in a Lagrangian frame.
  - Particles can exchange heat, mass, and momentum with continuous gas phase.
  - Each trajectory represents a group of particles of the same initial properties.
  - Particle-Particle interaction is neglected.
  - Discrete phase *volume* fraction must < 10% (*mass* loading is not limited).

$$\frac{d\vec{u}_p}{dt} = f_{\text{drag}}(\vec{u} - \vec{u}_p) + \vec{g}(\rho_p - \rho)/\rho_p + \vec{F}/\rho_p$$

- Numerous sub-modeling capabilities:
  - Heating/cooling of the discrete phase
  - Vaporization and boiling of liquid droplets
  - Volatile evolution and char combustion for combusting particles
  - Droplet breakup and coalescence using Spray Models
  - Erosion/Accretion
- Numerous Applications
  - Particle separation and classification, spray drying, aerosol dispersion, bubble sparging of liquids, liquid fuel and coal combustion.

NO_x Models

- NO_x consists of mostly nitric oxide (NO) which is harmful to the environment.
- Three mechanisms included in FLUENT for NO_x production:
  - Thermal NO_x - Zeldovich mechanism (oxidation of atmospheric N)
    - Most significant at high temperatures
  - Prompt NO_x - empirical mechanisms by De Soete, Williams, etc.
    - Contribution is in general small
    - Significant at fuel rich zones
  - Fuel NO_x - Empirical mechanisms by De Soete, Williams, etc.
    - Predominant in coal flames where fuel-bound nitrogen is high and temperature is generally low.
- NO_x reburn chemistry
  - NO can be reduced in fuel rich zones by reaction with hydrocarbons.
- SNCR models (Ammonia injection)
- These models provide qualitative trends of NO_x formation.

Soot Formation Models

- Two soot formation models are available:
  - One-step model (Khan and Greeves)
    - Single transport equation for soot mass fraction
  - Two-Step model (Tesner)
    - Transport equations for radical nuclei and soot mass fraction concentrations
- Soot formation modeled by empirical rate constants

$$\overline{R}_{formation} = C\ p_f \Phi^n e^{-E/RT}$$

where C, p_f and Φ are a model constant, fuel partial pressure and equivalence ratio, respectively

- Soot combustion (destruction) modeled by Magnussen model
- Soot affects the radiation absorption
  - Enable Soot-Radiation option in the Soot panel

Outline

- Aspects of Reaction Modeling and Fundamental Concepts
- Reaction Models in FLUENT
  - Overview
  - Fast Chemistry Models: eddy dissipation, premixed, partially premixed, non-premixed
  - Moderately non-equilibrium model: Laminar flamelet for non-premixed flames
  - Finite-rate Chemistry Models: Laminar finite-rate, eddy dissipation concept (EDC), composition PDF
- ISAT Algorithm, DPM and Pollutant Formation Models
- **Surface Reactions**
- Examples and Case Studies

Surface Reactions

- Molecules from the gaseous phase are first adsorbed onto the surface with available "sites." The surface-adsorbed molecules would then react with some other surface species.
- Site balance equation is solved for every surface-adsorbed (or "site") species.
- Detailed surface reaction mechanism can be considered (any number of reaction steps and any number of gas-phases or/and site species).
- Surface chemistry mechanism in Surface CHEMKIN format can be imported into Fluent
- The surface-adsorbed species are treated as distinct from the same species in the gaseous phase.
- Surface reaction can occur at a wall or in porous media.
- Different surface reaction mechanisms can be specified on different surfaces.
- Applications
  - Catalytic reactions, CVD, etc.

Outline

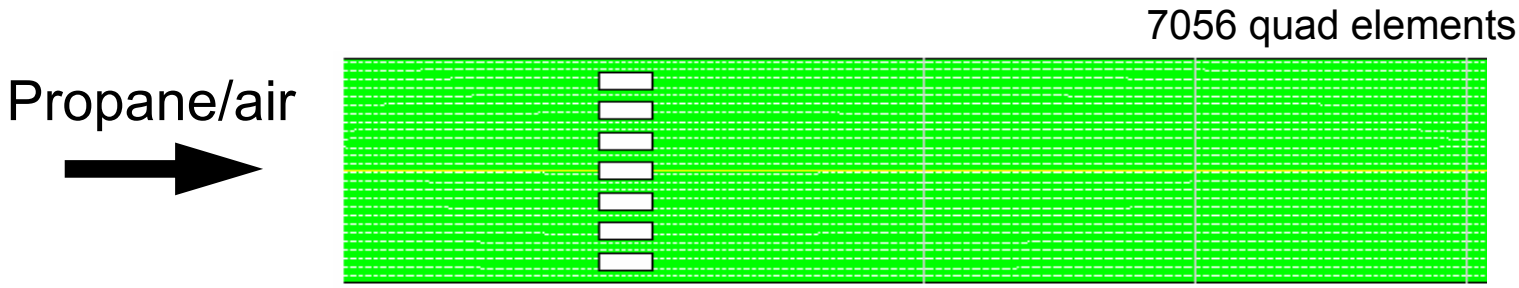
- Aspects of Reaction Modeling and Fundamental Concepts
- Reaction Models in FLUENT
  - Overview
  - Fast Chemistry Models: eddy dissipation, premixed, partially premixed, non-premixed
  - Moderately non-equilibrium model: Laminar flamelet for non-premixed flames
  - Finite-rate Chemistry Models: Laminar finite-rate, eddy dissipation concept (EDC), composition PDF
- ISAT Algorithm, DPM and Pollutant Formation Models
- Surface Reactions
- **Examples and Case Studies**

Examples and Case Studies

- Anderson Premixed Flame
  - Laminar finite rate
  - EDC
  - PDF transport
- IFRF MMF5 flame
  - Coal combustion
  - Eddy dissipation model
- GE LM-1600 Gas Turbine Combustor
  - Flamelet Model

Case Study: Lean Premixed Flame

- Lean premixed pre-vaporized (LPP) propane/air flame
  - Reduction of the emissions of combustion pollutants (CO and NOx)
  - Measurement of CO and NOx emissions*
- Axisymmetric flame tube
  - Geometry: d = 10.25 cm, L = 42.92 cm
  - Propane/air mixture: v = 25 m/s, T = 800 K
  - Flame holder: 51% open area



*Anderson, D. N., NASA Lewis Research Center, NASA-TM-X-71592, March 1975

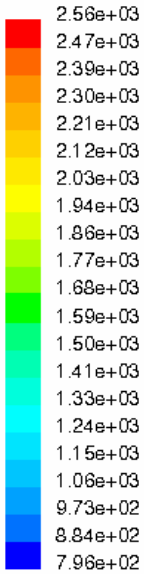
Lean Premixed Flame (cont'd)

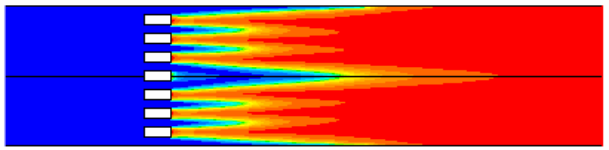
- Weak turbulence-chemistry interaction
  - Perfectly premixed
  - Thin flame zone
  - Chemical reactions occur right behind the flame front
- Propane detailed chemical mechanism
  - Kundu mechanism with NO and N_2O formation*
    - 17 species and 23 step reactions
- Finite-rate chemistry combustion models
  - Laminar finite-rate model
    - Fully ignoring turbulence-chemistry interactions
  - EDC model
    - Might lead to under-prediction of NOx formation: The reaction occurs quickly and NOx has less time to accumulate.
  - Composition PDF transport
    - Accurately capturing the turbulence-chemistry interaction

*Kundu, K. P., et al., NASA Lewis Research Center, AIAA paper 98-3986, July 1998

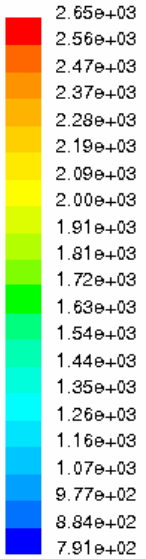
Lean Premixed Flame (cont'd)

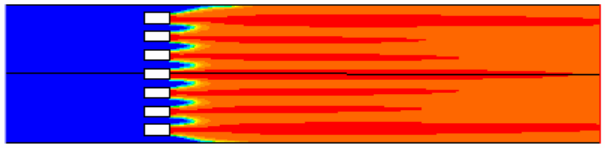
Temperature Contours



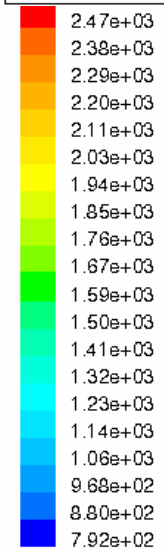


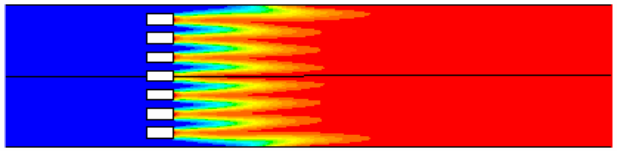
EDC model





Laminar finite-rate Model

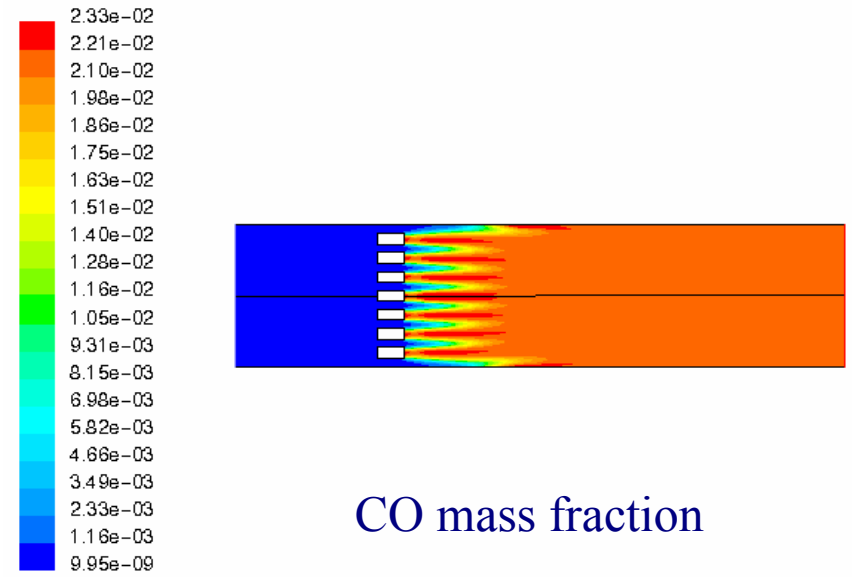




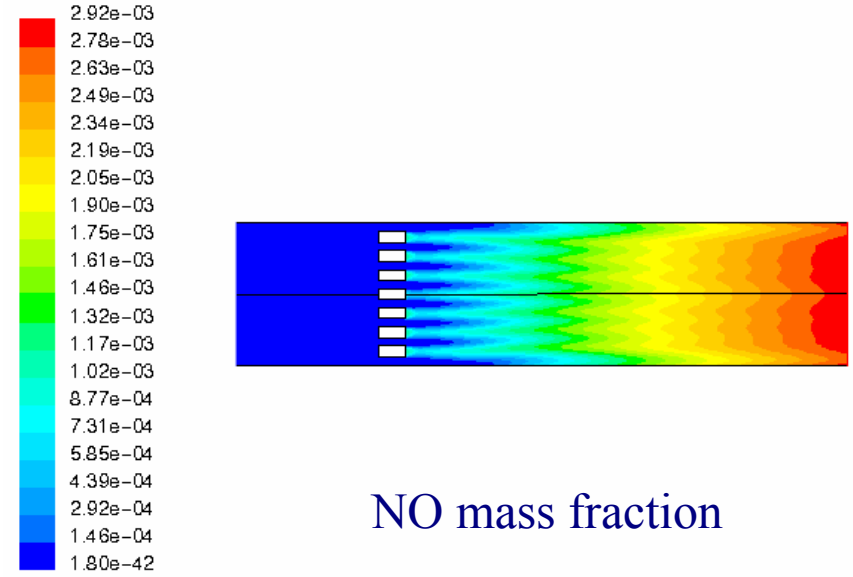
PDF model

Lean Premixed Flame (cont'd)

Mass fractions of CO and NO from the PDF transport model



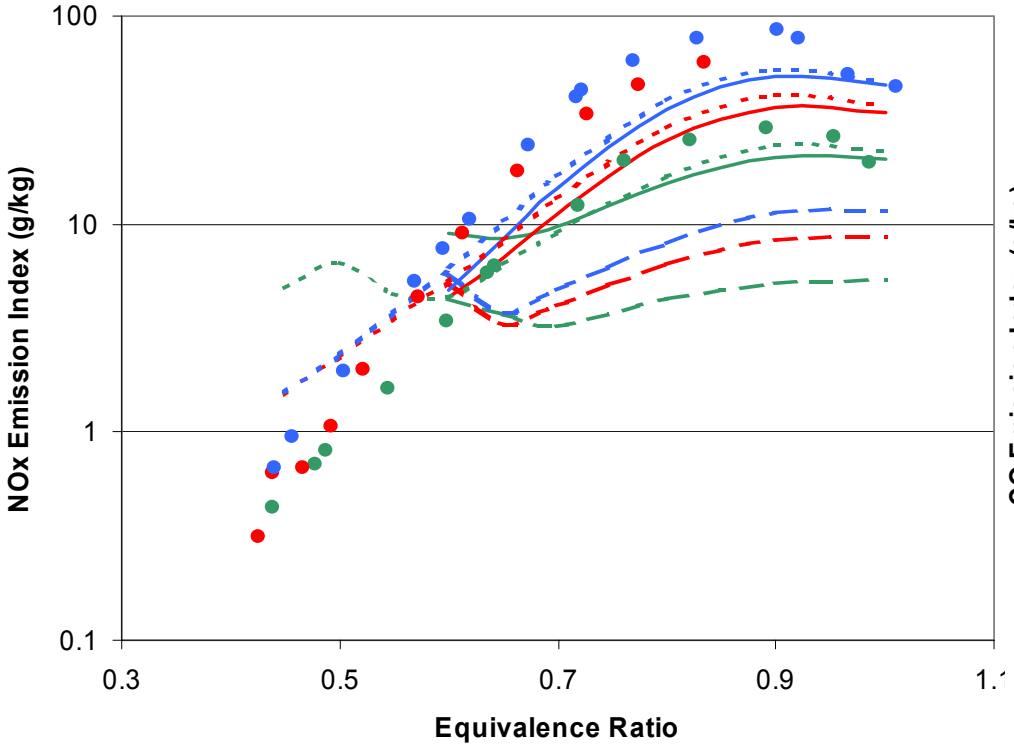
CO mass fraction

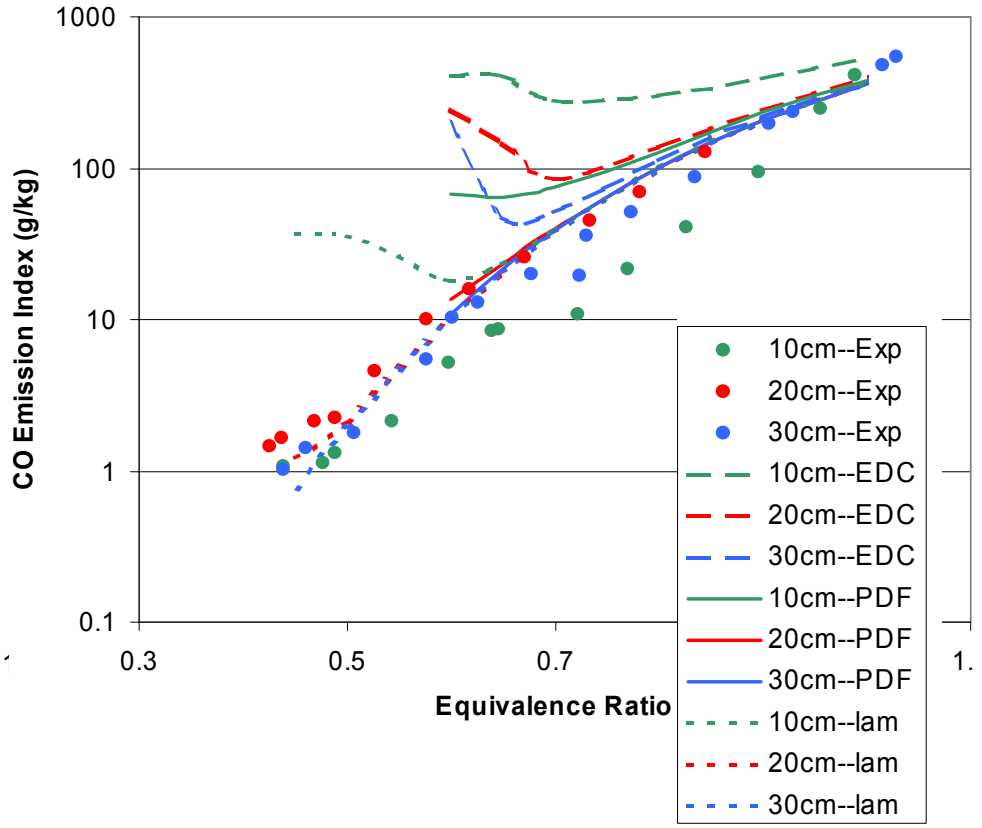


NO mass fraction

Lean Premixed Flame (cont'd)

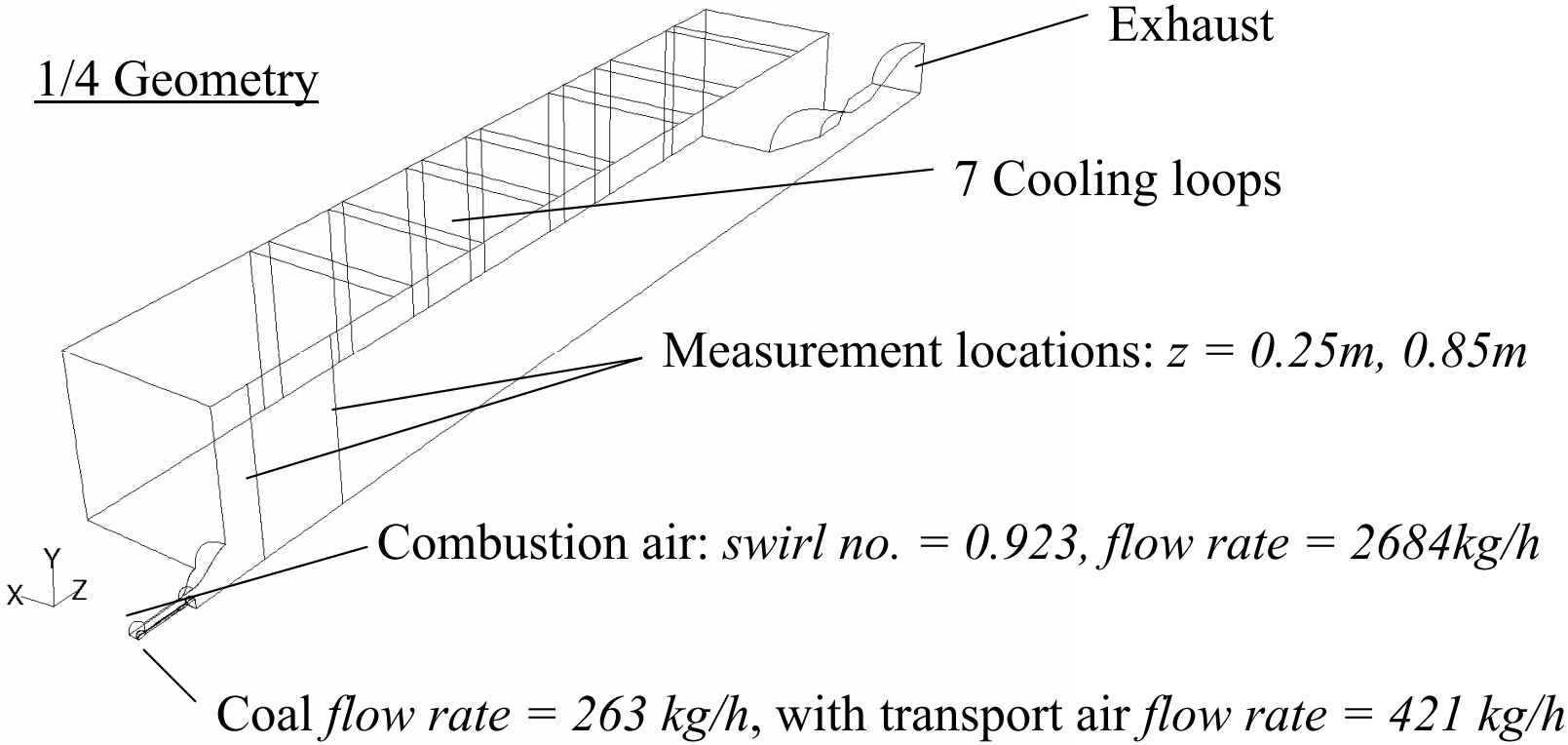
Emission Index = $\dot{m}_{\text{Pollutant}} / \dot{m}_{\text{fuel}}$





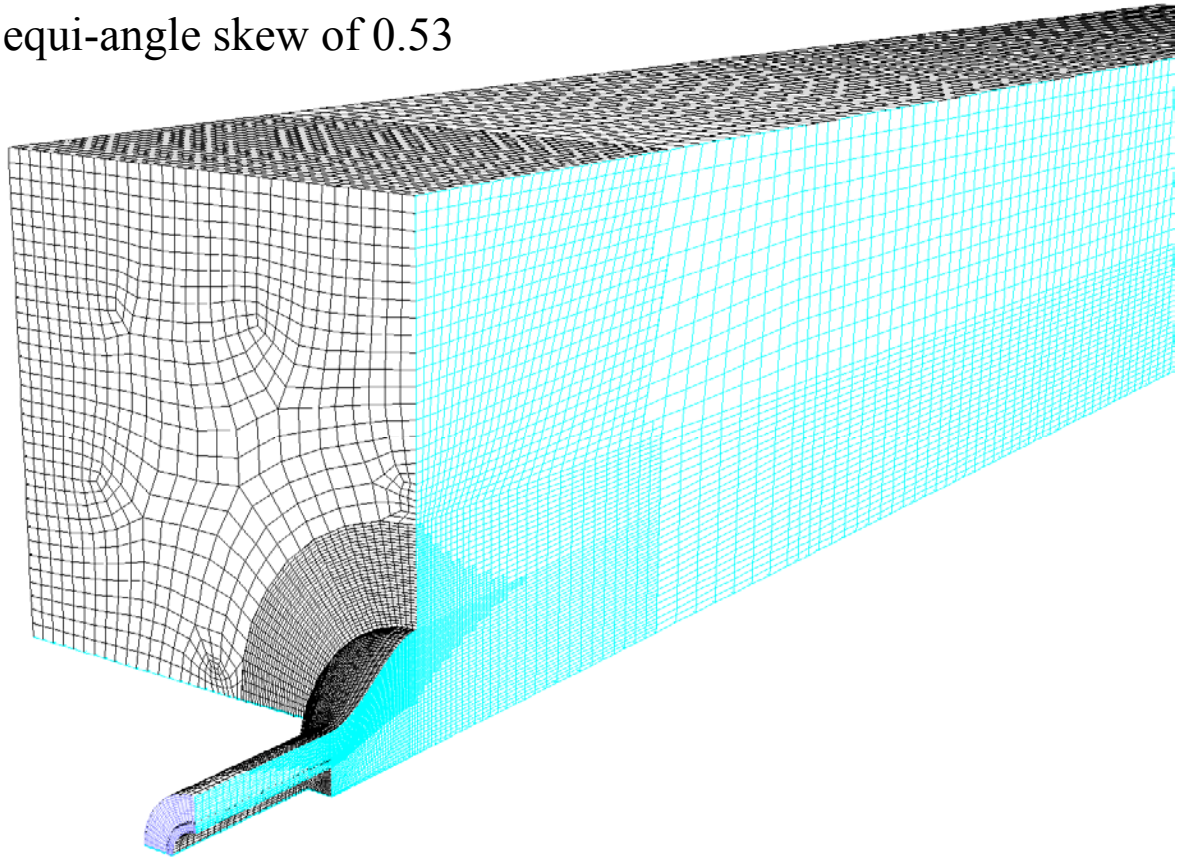
IFRF Swirling Pulverized Coal Flame

- IFRF industrial scale furnace
- Built on simulation by Peters and Weber (1997), *Mathematical Modeling of a 2.4 MW Swirling, Pulverized Coal Flame*, Combustion Science and Technology, **122**, 131 [Ref. 1]



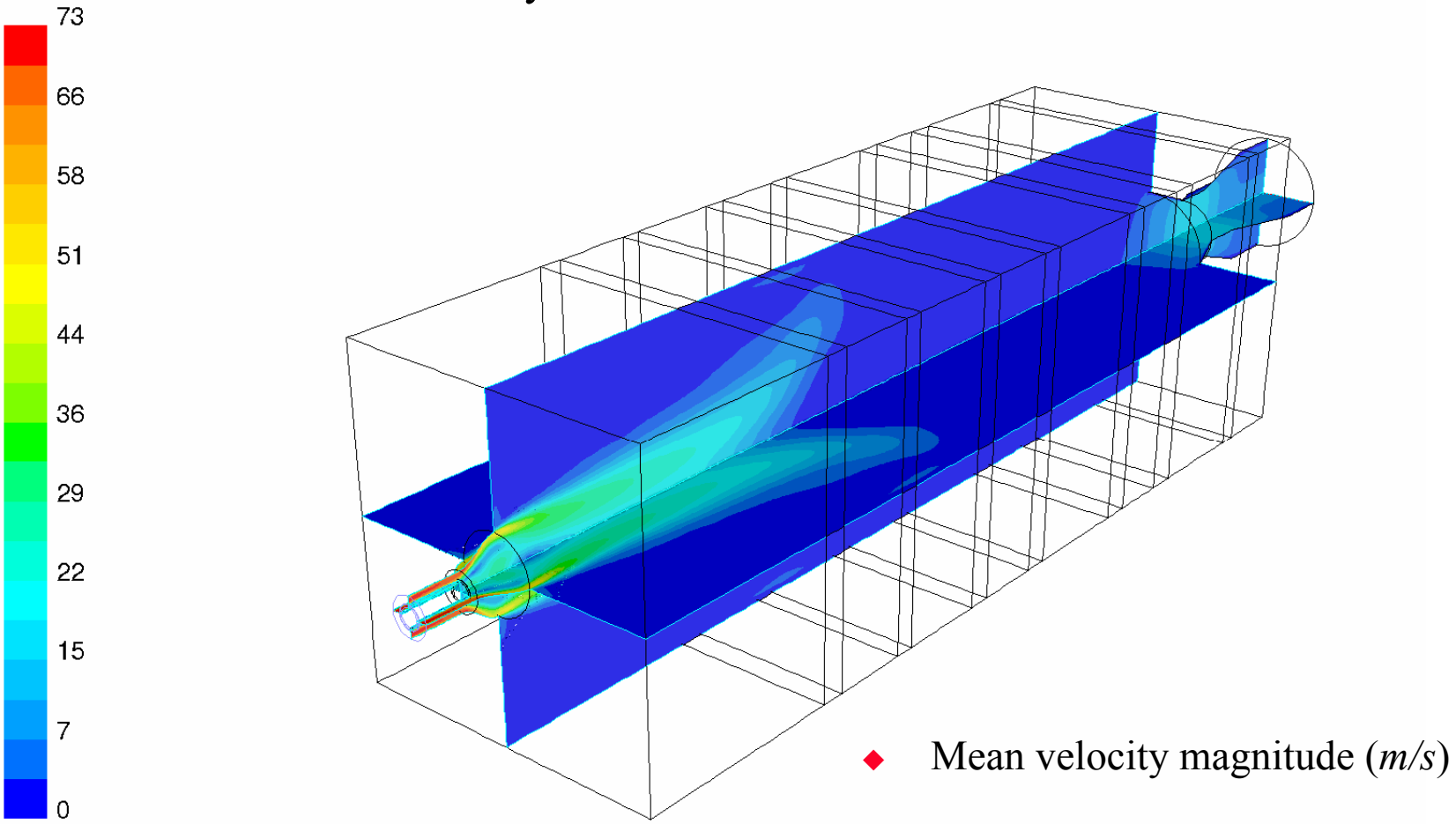
Grid

- 3D, one quarter geometry model due to periodicity
- Unstructured hexahedral mesh
  - 70k cells before adaption
  - 260k cells after region adaption near inlet
  - Maximum equi-angle skew of 0.53



Turbulence Modeling

- Standard k–ε turbulence model, standard wall functions
  - Not substantially sensitive to the turbulence model



- Mean velocity magnitude (*m/s*)

Gas Phase Combustion Modeling

- Eddy Dissipation (Magnussen) model [parameters from Ref. 1]
  - Two step reaction

$$VOL + 2.46O_2 \longrightarrow 2.17CO + 0.633CO_2 + 2.118H_2O + 0.071N_2$$

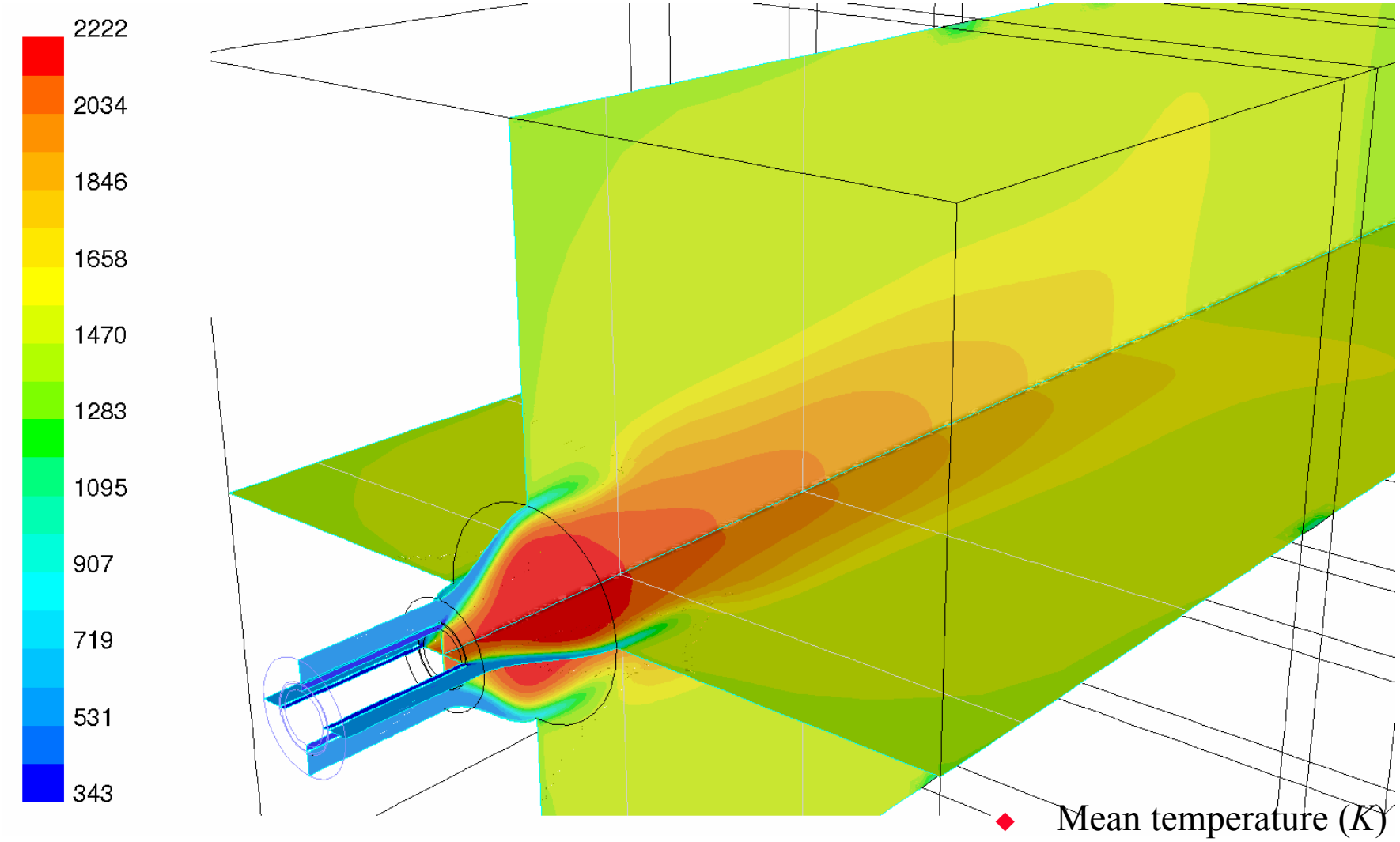
$$CO + 0.5O_2 \longrightarrow CO_2$$

  - Model constants $A = 0.6$, $B = 10^{20}$ (standard $A = 4$, $B=0.5$)
  - Adjusted specific heat's *(SI units)*

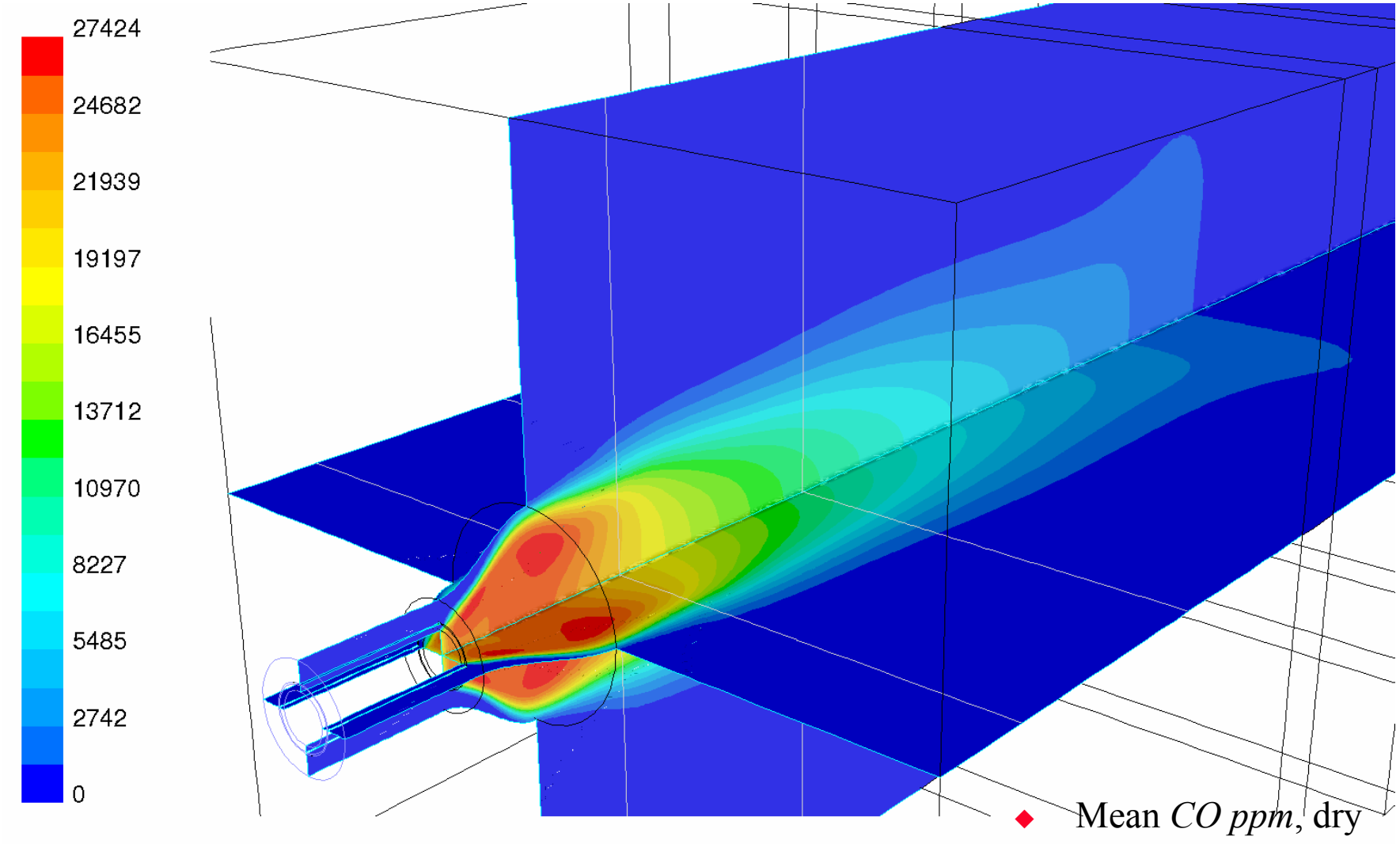
$$c_p(T) = \sum_i c_i \cdot T^i$$

Species	c_0	c_1	c_2	c_3	c_4	c_5	c_6
N_2	1.027e3	2.162e-2	1.486e-4	-4.484e-8			
CH_4	2.005e3	-6.814e-1	7.086e-3	-4.714e-6	8.513e-10		
CO	1.047e3	-1.568e-1	5.399e-4	-3.011e-7	5.050e-11		
H_2	1.415e4	1.737e-1	6.900e-4				
CO_2	5.354e2	1.279	-5.468e-4	-2.382e-7	1.892e-10		
H_2O	1.938e3	-1.181	3.644e-3	-2.863e-6	7.596e-10		
O_2	8.763e2	1.228e-1	5.583e-4	-1.202e-6	1.147e-9	-5.124e-13	8.566e-17

Gas Phase Combustion Modeling



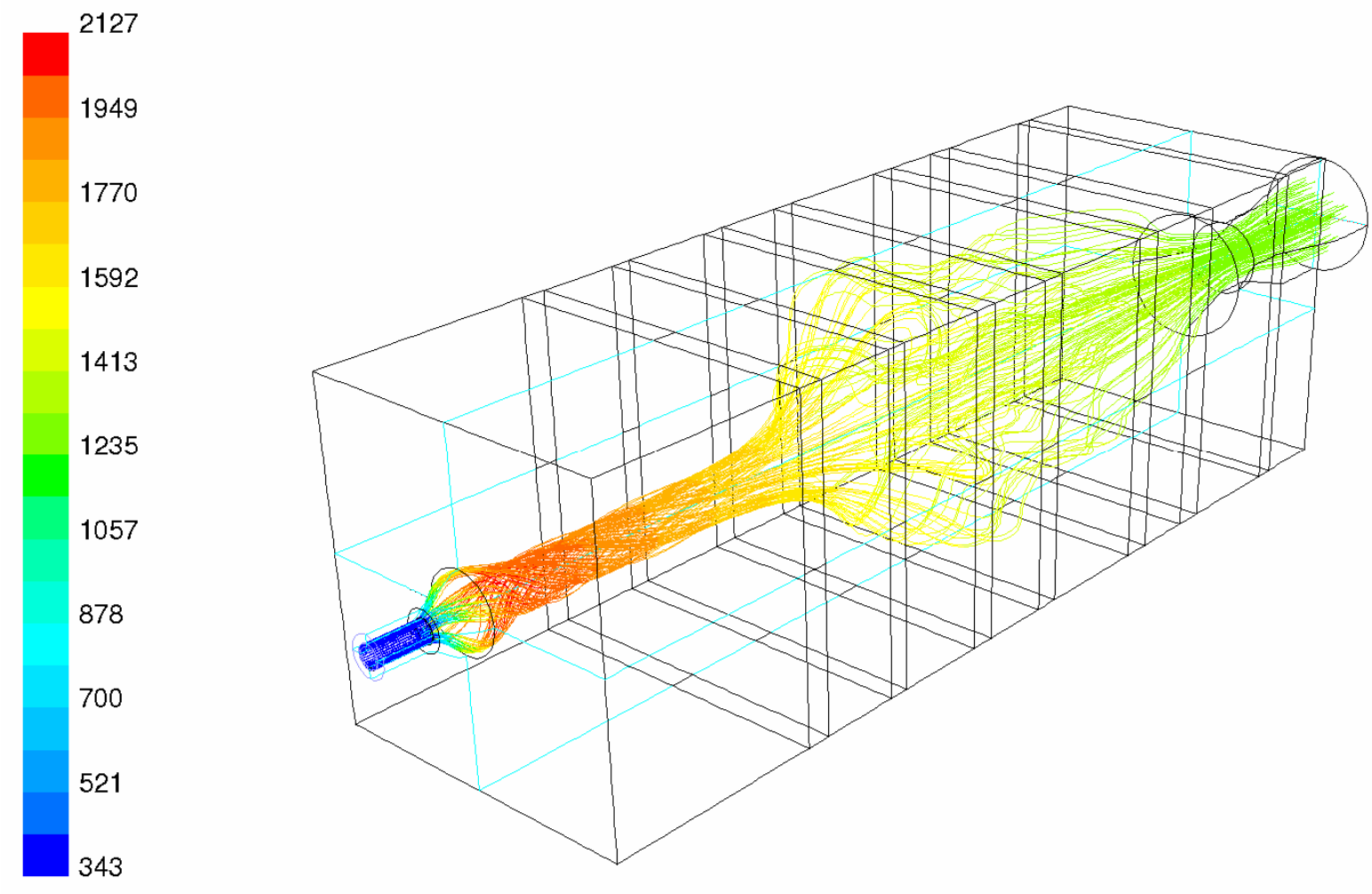
Gas Phase Combustion Modeling



Discrete Phase Modeling

- Gottelborn hvBb coal
  - Proximate (weight %, dry), *volatiles 55.0, fixed carbon 36.7, ash 8.3*
  - Ultimate (weight %, daf) *C 80.36, H 5.08, N 1.45, S 0.94, O 12.17*
  - Lower Calorific Value (LCV, *MJ/kg* daf) *volatiles 32.3, char 32.9*
- Rosin-Rammler size distribution
  - Smallest $1\mu m$, largest $300\mu m$, mean $45\mu m$, spread *1.36*
- Single rate devolatization model,
  - $A=2*10^5\ s^{-1}$, $E=7.4*10^7\ J/kmol$
- Kinetics/diffusion-limited surface (char) combustion
  - O_2 diffusion rate const = $5*10^{-12}\ kg/m^2sPa$, $A=6.7\ kg/m^2sPa^{0.5}$, $E=1.14*10^8\ J/kmol$
- Discrete Random Walk (DRW) model
  - 21600 tracks per DPM iteration, 10 particle sizes
  - 25 gas phase iterations per DPM iteration

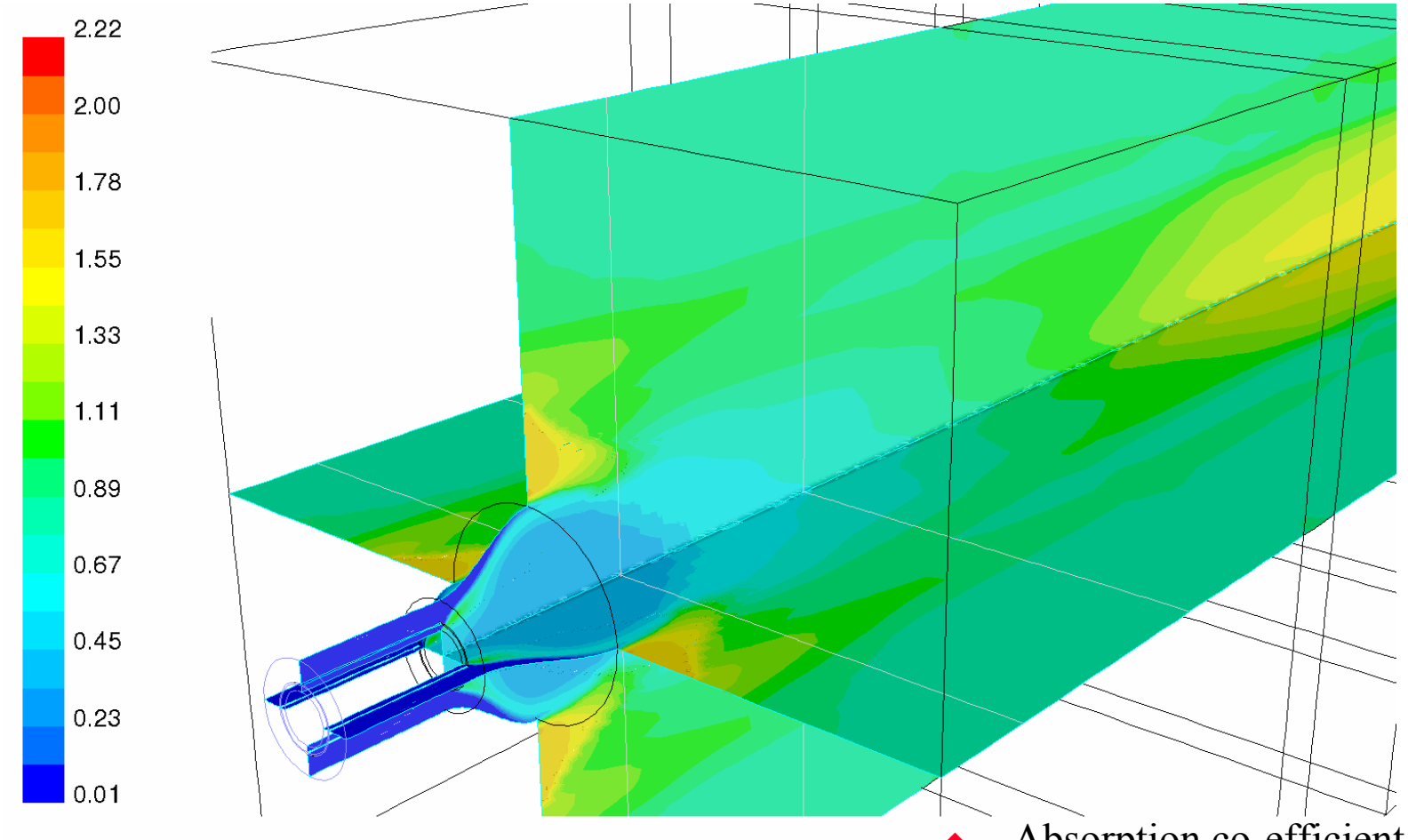
Discrete Phase Modeling



- Tracks of *1 μm* particles, colored by particle temperature (*K*)

Radiation Modeling

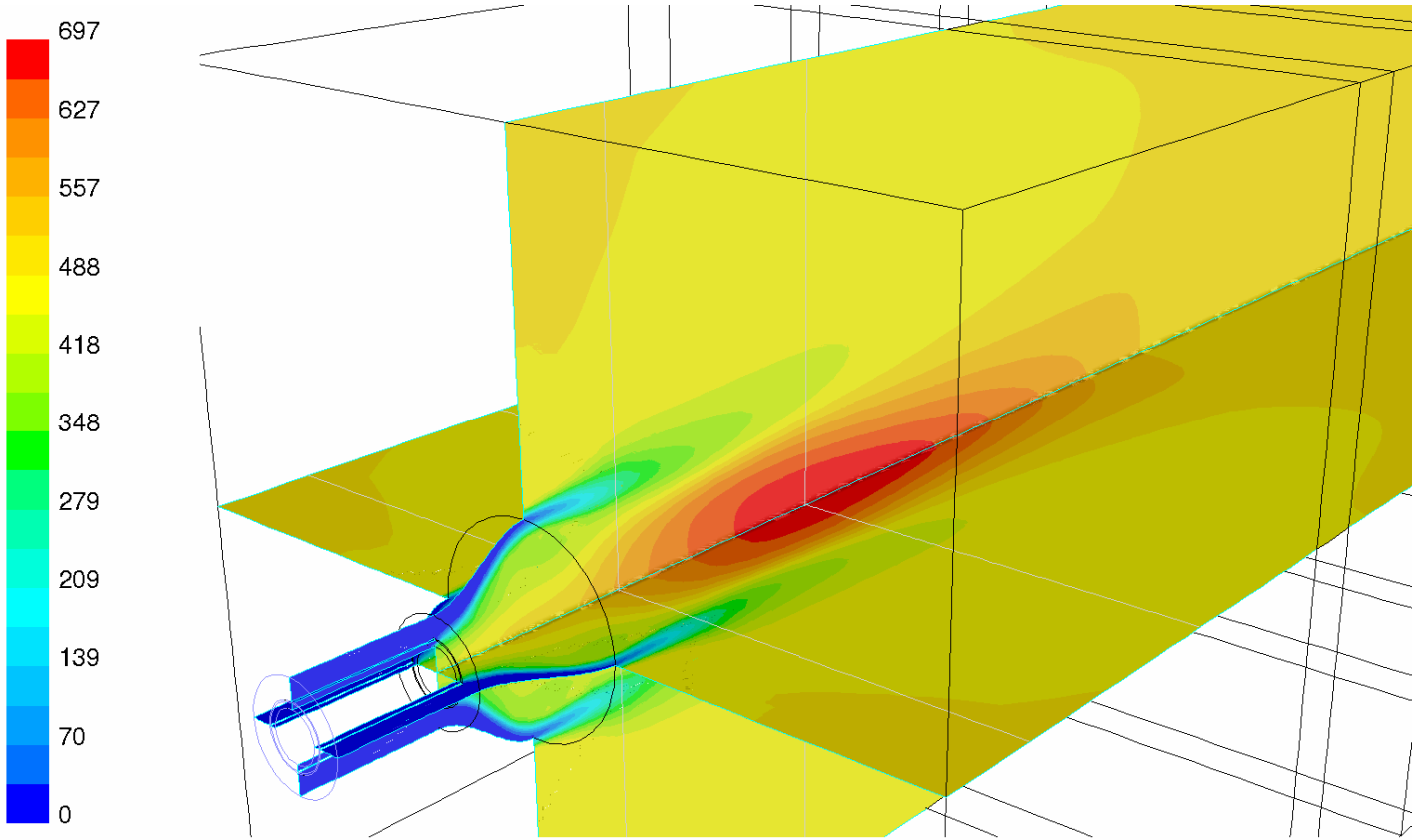
- *P-1* radiation model (optical thickness > 1)
  - *WSGGM* for absorption co-efficient



- Absorption co-efficient (m^{-1})

NOx Modeling

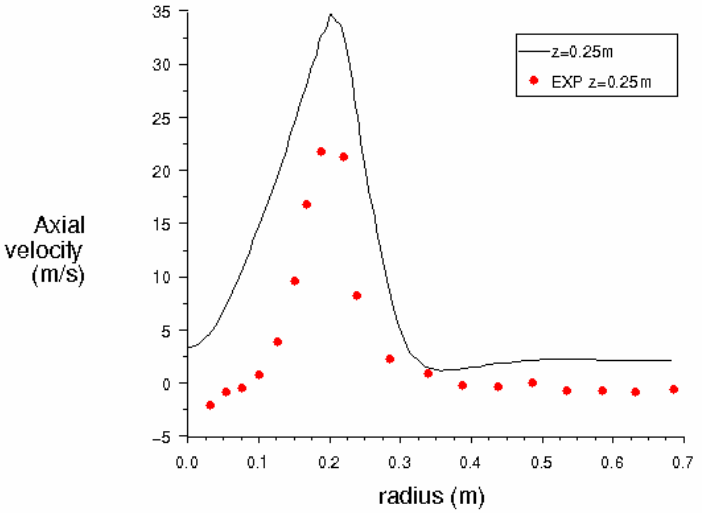
- Thermal and fuel *NOx*
- *[O]* from partial equilibrium assumption
- Post-processed: assumed shape β pdf

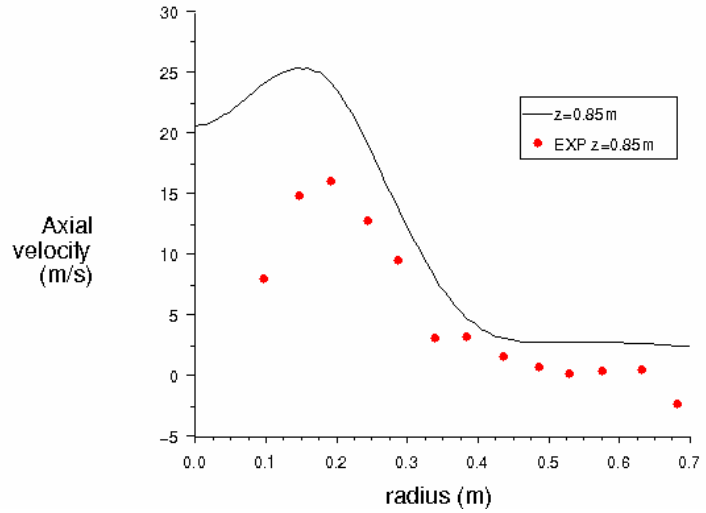


- Mean *NO ppm*, dry

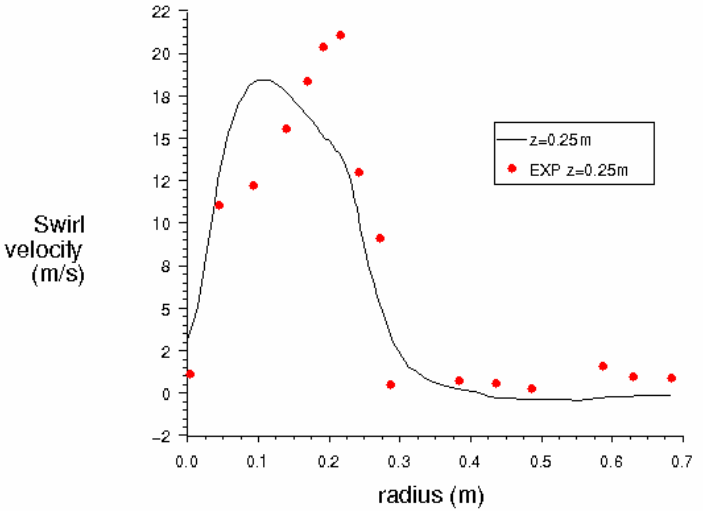
Results Velocity Field

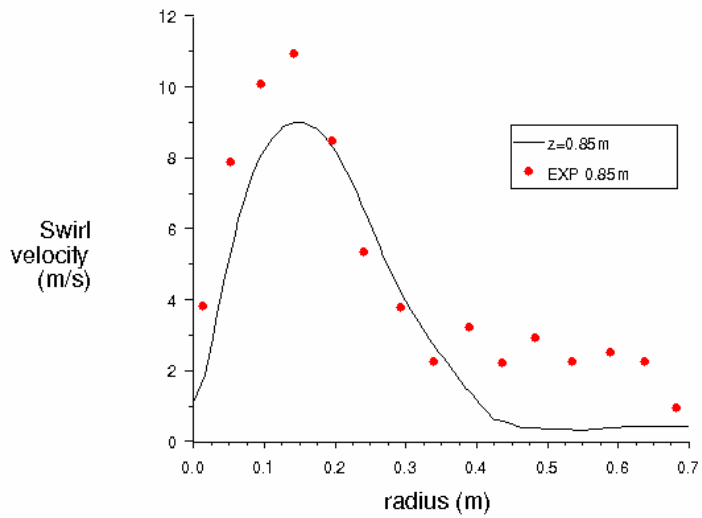
- Mean axial velocity (m/s)





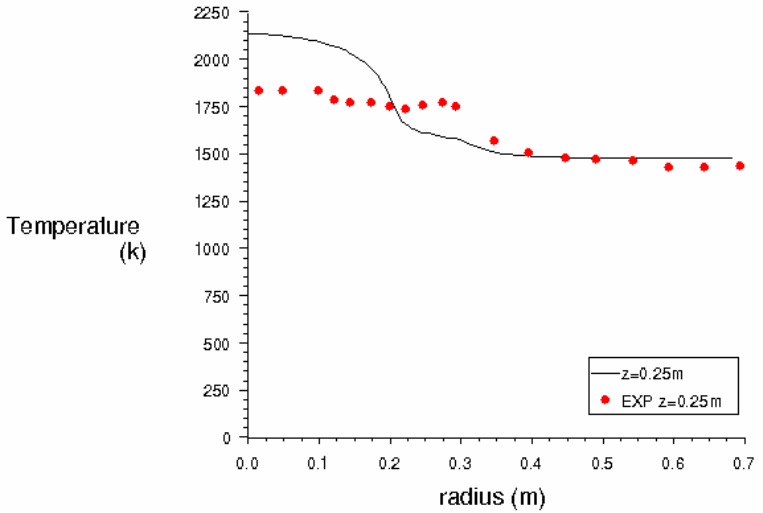
- Mean swirl velocity (m/s)

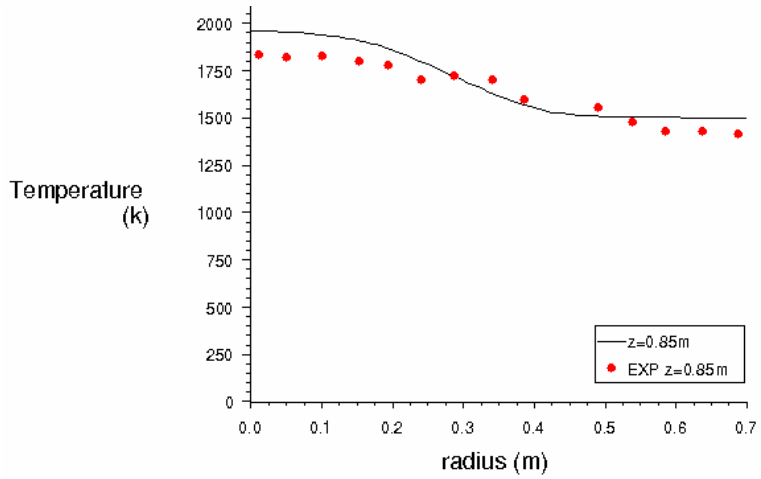




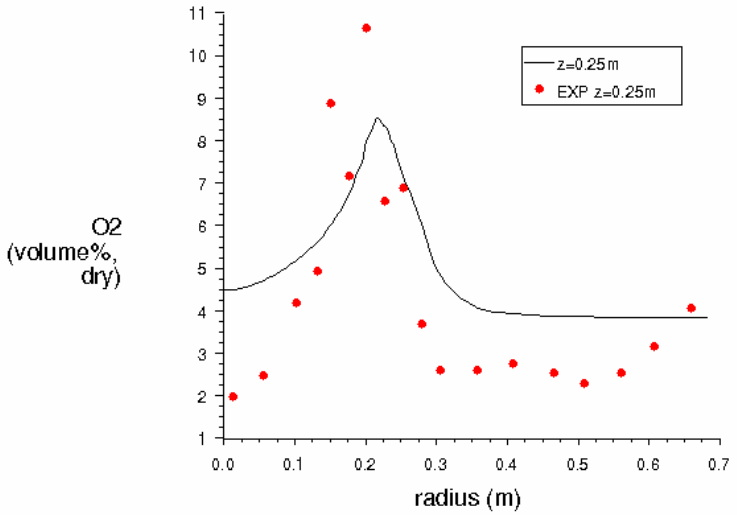
Results Temperature/Species Field

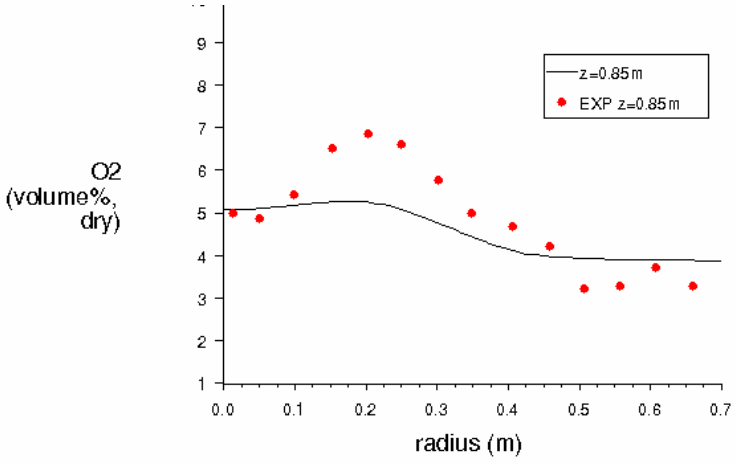
- Mean temperature (K)





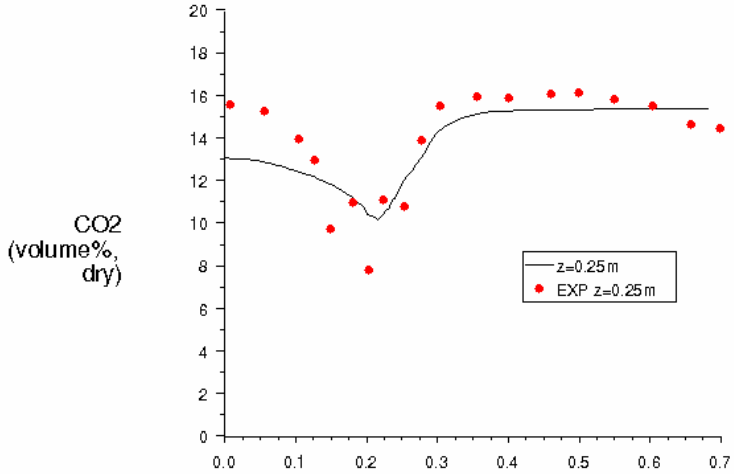
- Mean O_2 (volume %, dry)

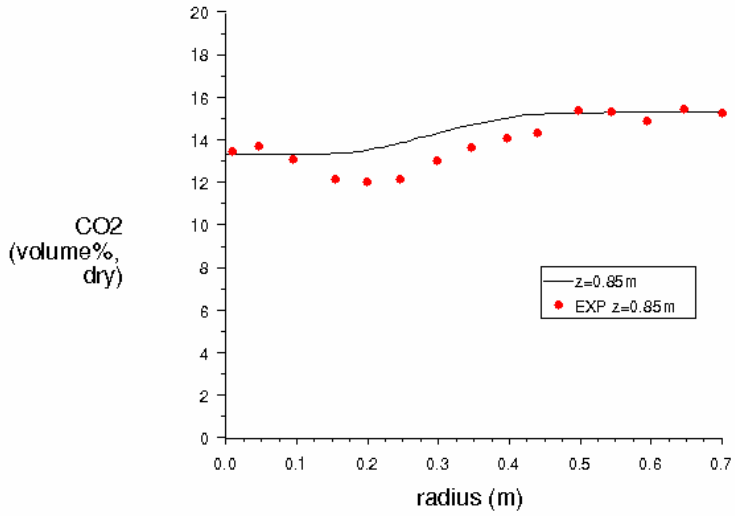




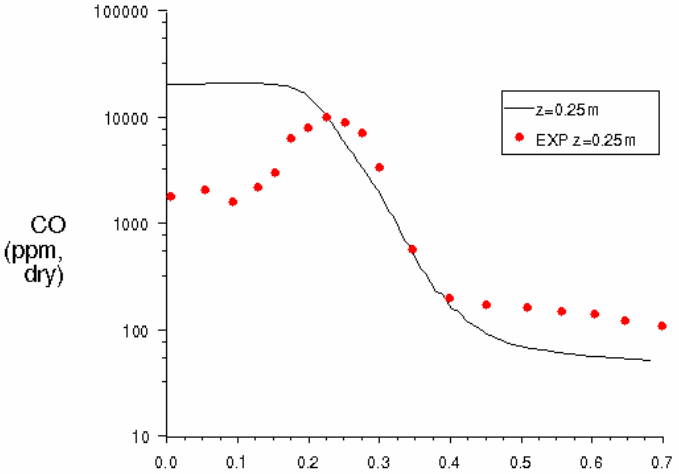
Results Species Field

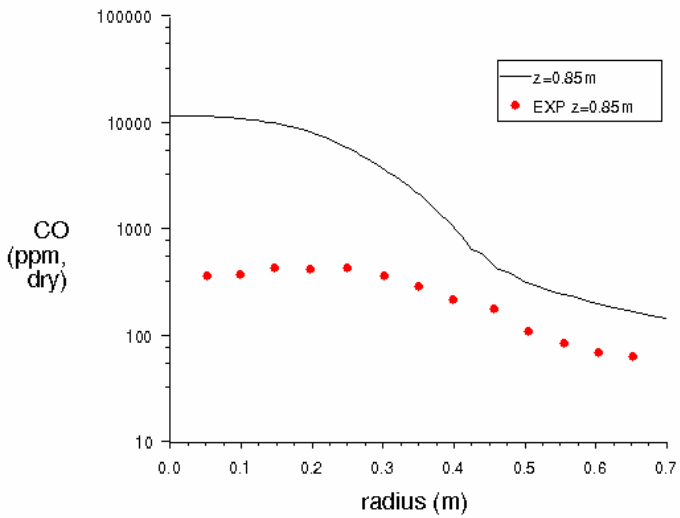
- Mean CO_2 (volume %, dry)





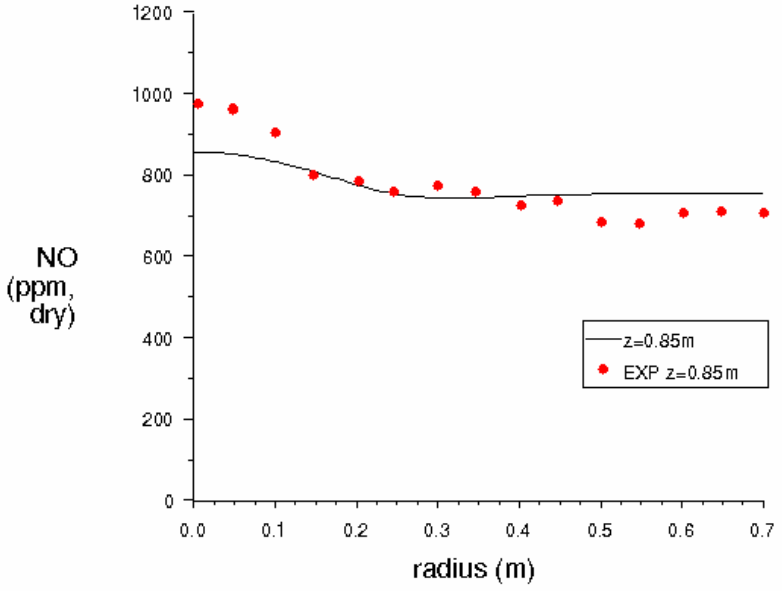
- Mean CO (ppm, dry)

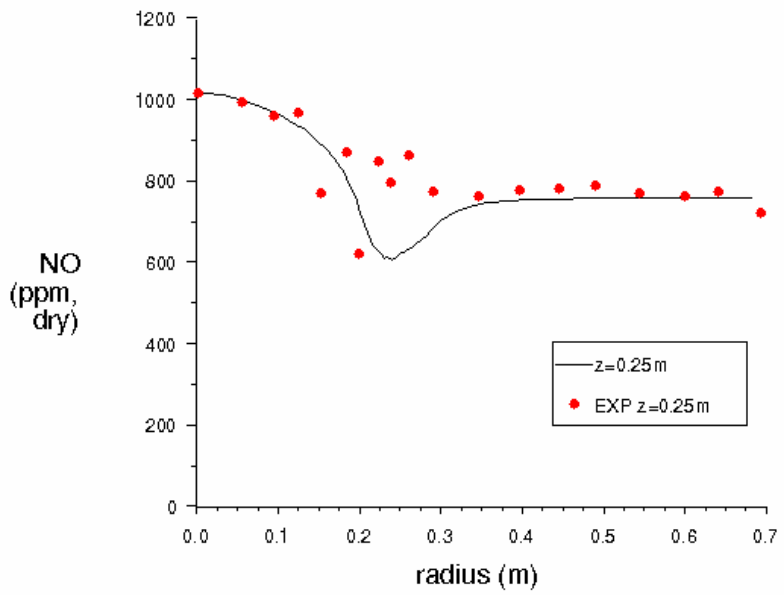




Results NOx Field

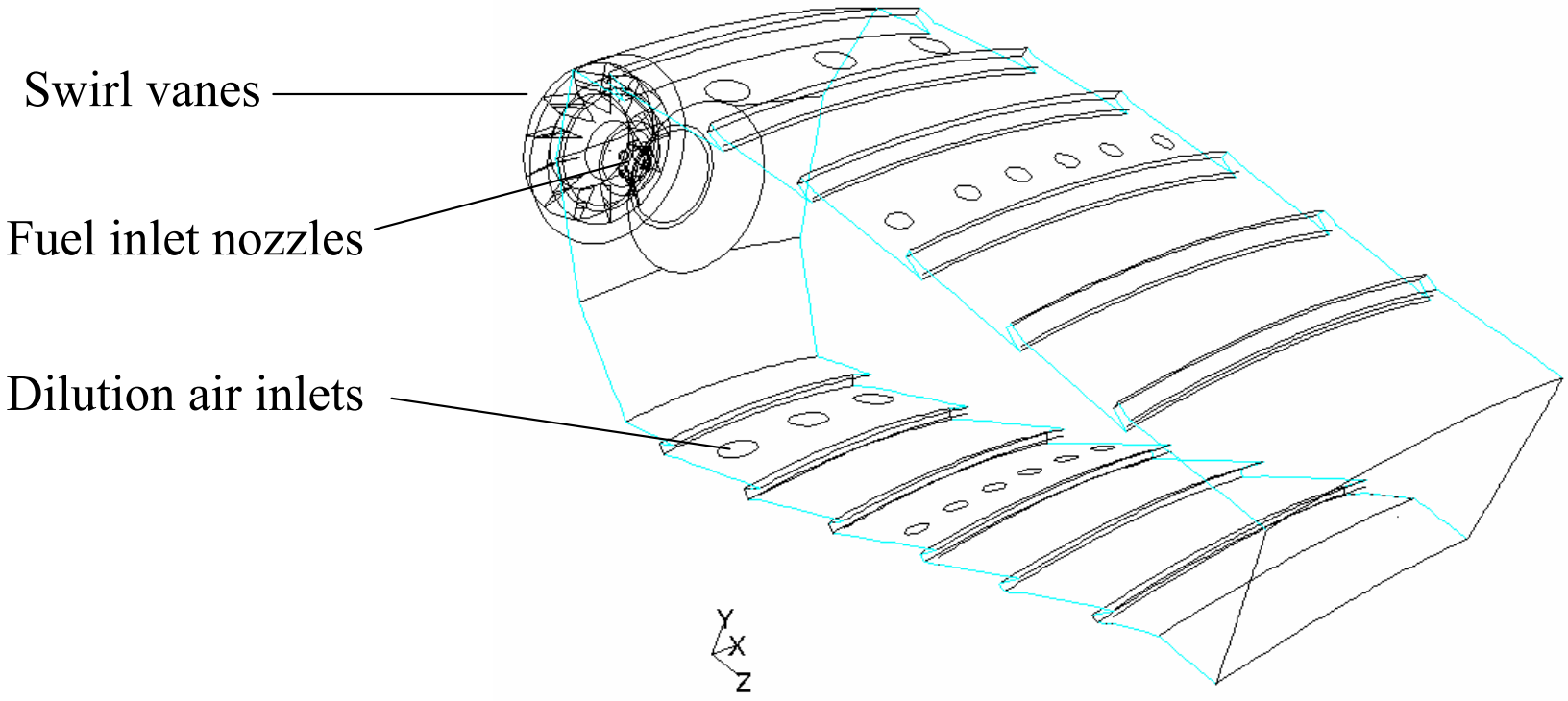
- Mean NO (ppm, dry)





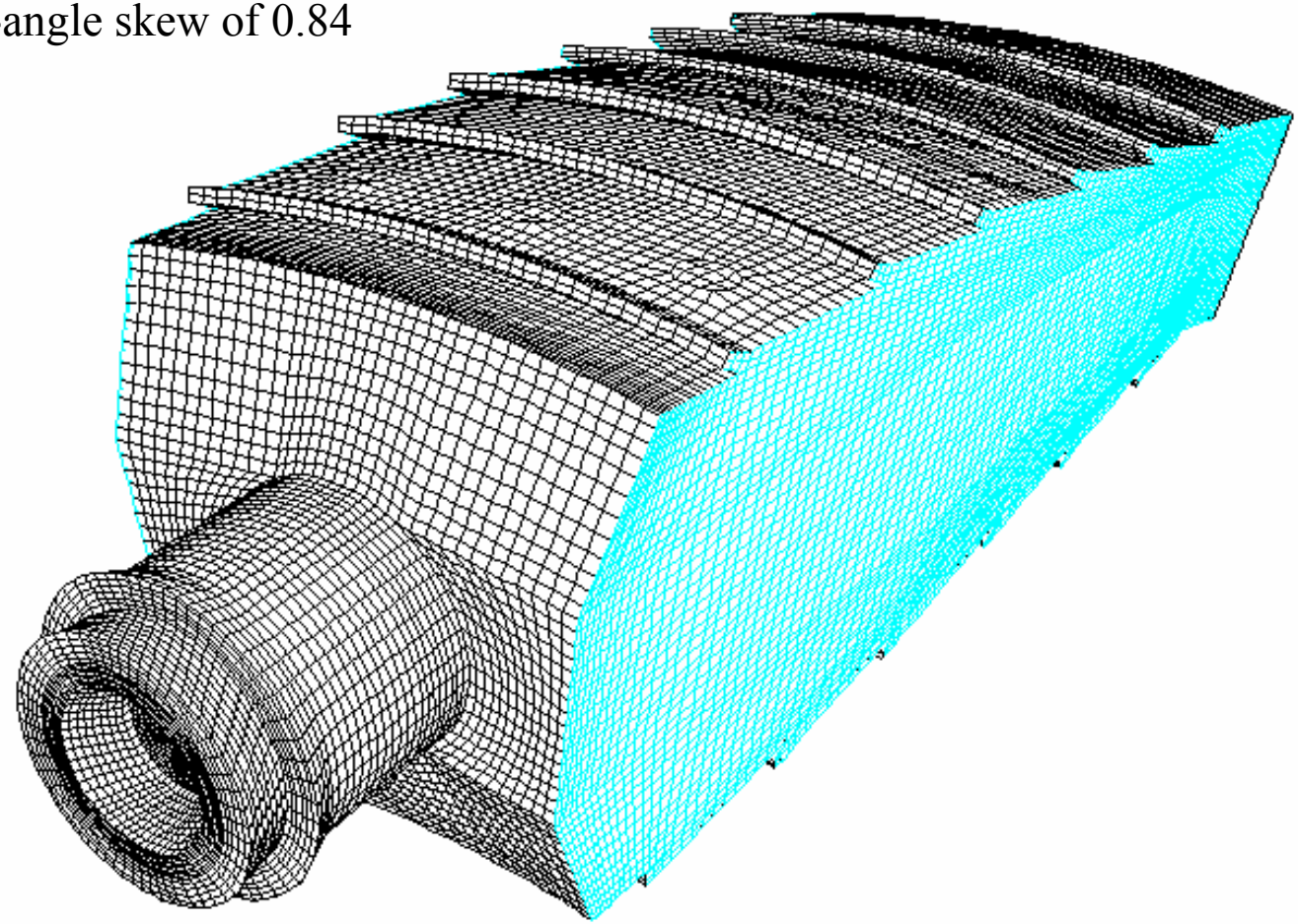
GE LM-1600 Gas Turbine Combustor

- Courtesy of *Nova Research and Technology Corp.*, Calgary, Canada
- Non-premixed, natural gas
- 12.8 *MW*, 19:1 pressure ratio (full load)
- Annular combustion chamber, 18 nozzles



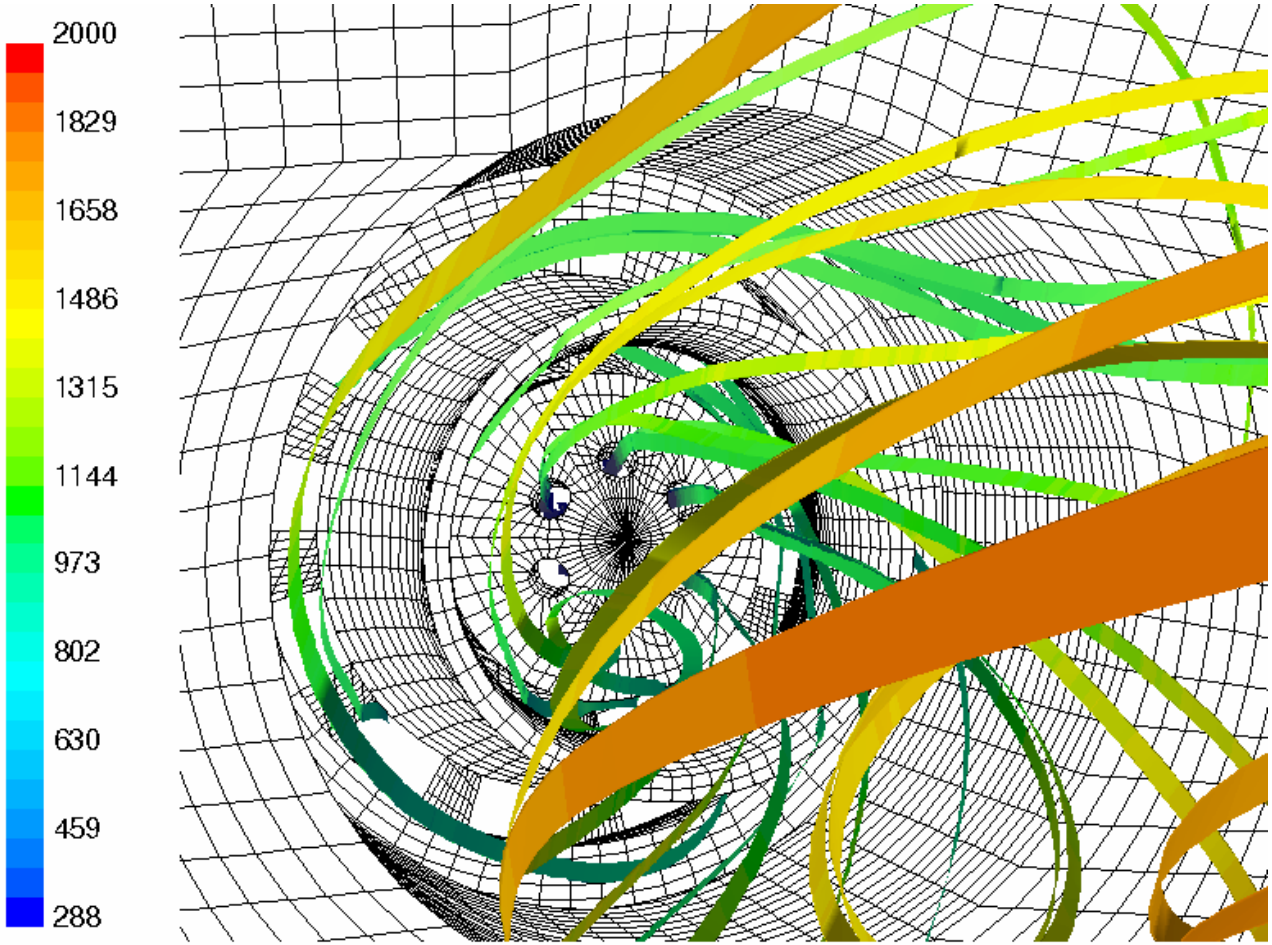
Grid

- 3D, 1/18*th* geometry model due to periodicity
- Multi-block hexahedral mesh
  - Maximum equi-angle skew of 0.84
  - 286k cells



Turbulence Modeling

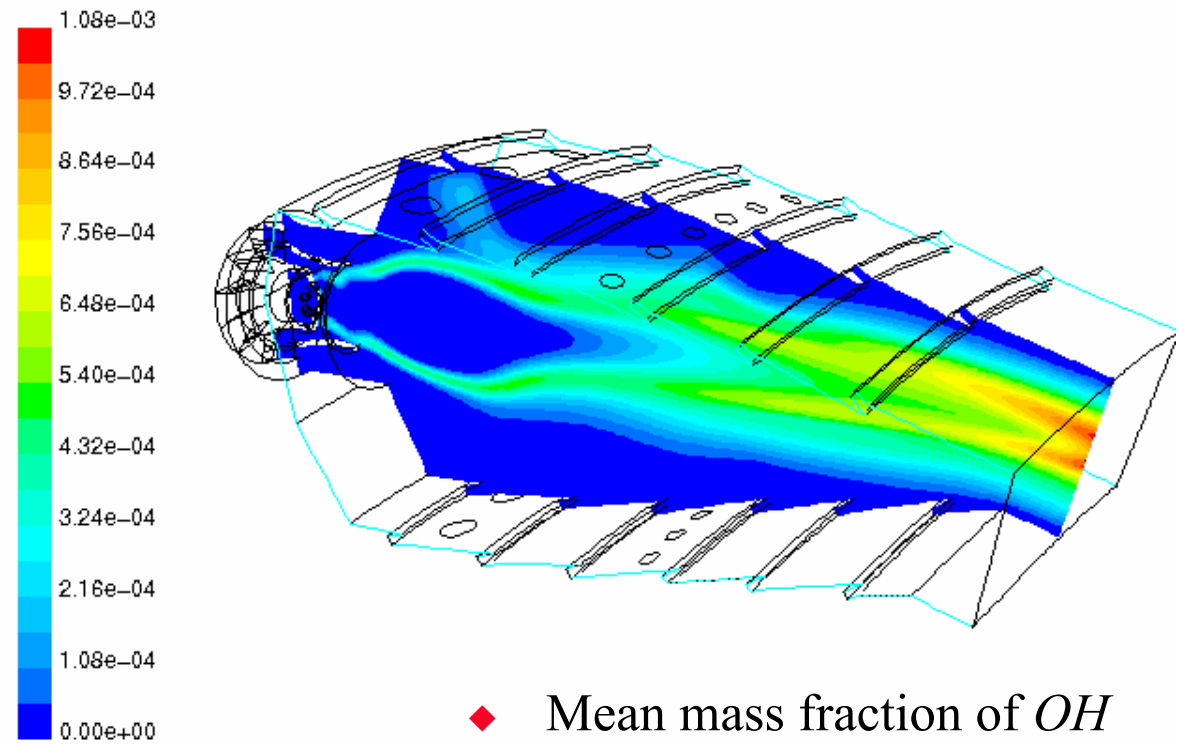
- Standard $k-\varepsilon$ turbulence model



- Path ribbons colored by temperature (K)

Gas Phase Combustion Modeling

- Laminar Flamelet model
  - 22 species, 104 reactions reduced GRI-MECH 1.22 mechanism
    A. Kazakov and M. Frenklach, *http://www.me.berkeley.edu/drm*
  - Flamelets solved in mixture fraction space
  - Differential diffusion (*Le* effects) included



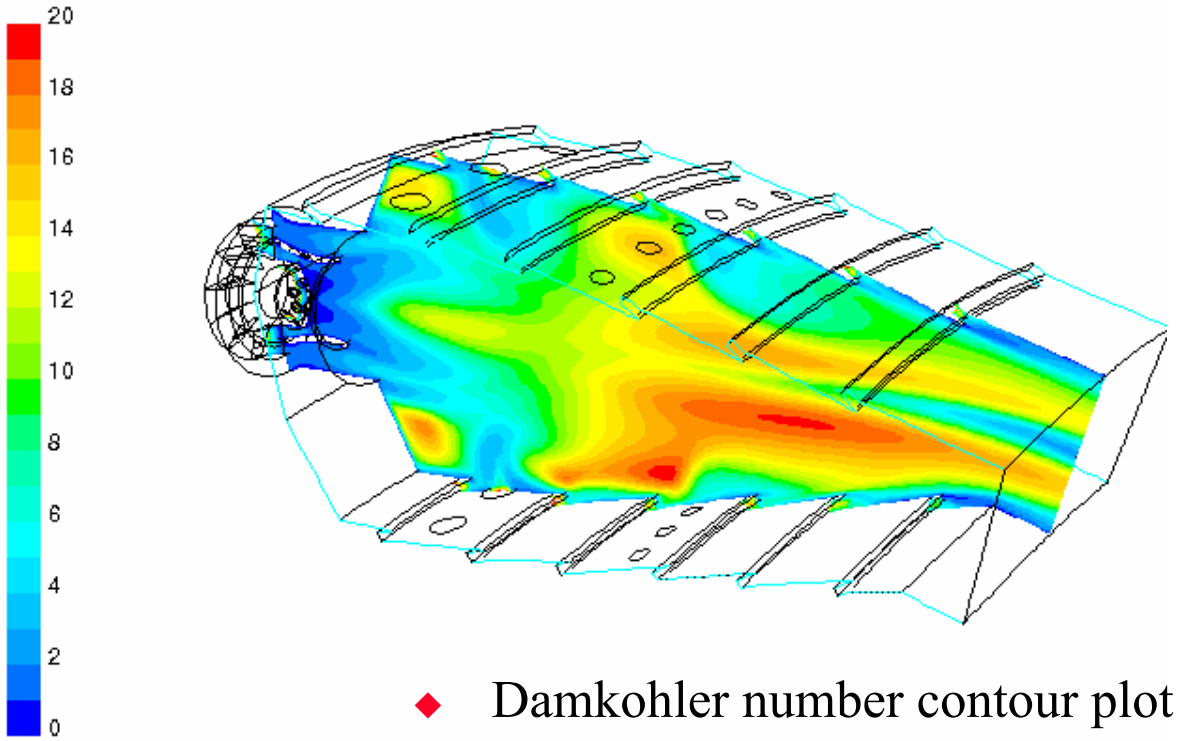
- Mean mass fraction of *OH*

Gas Phase Combustion Modeling

- Deviation from chemical equilibrium measured by Damkohler no.

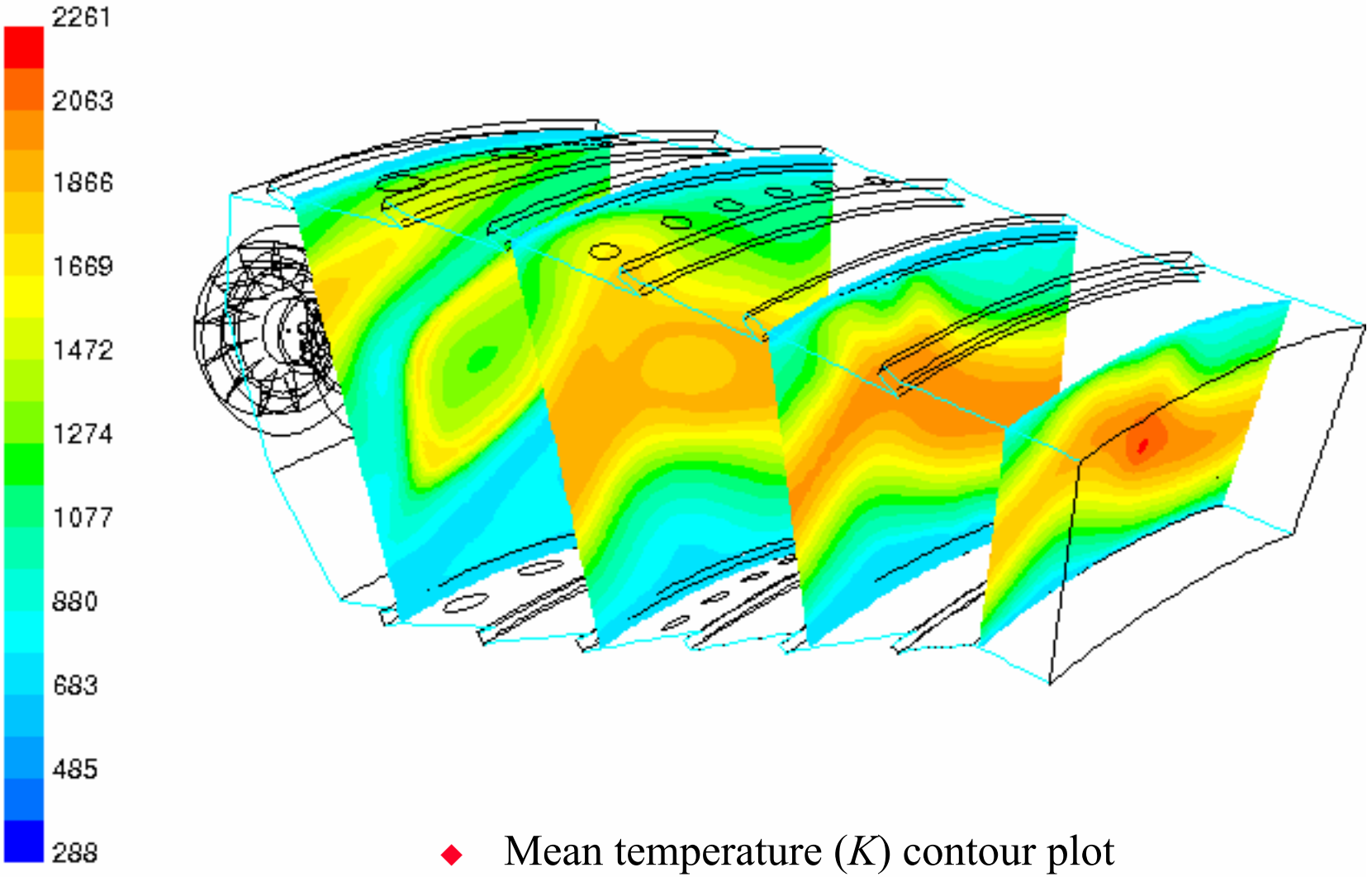
$$Da = \frac{turbulent\ time\ scale}{chemical\ time\ scale} = \frac{\tilde{k} / \tilde{\varepsilon}}{a_q^{-1}}$$

  - a_q is the laminar flamelet extinction strain rate = 11700 s^{-1}



- Damkohler number contour plot

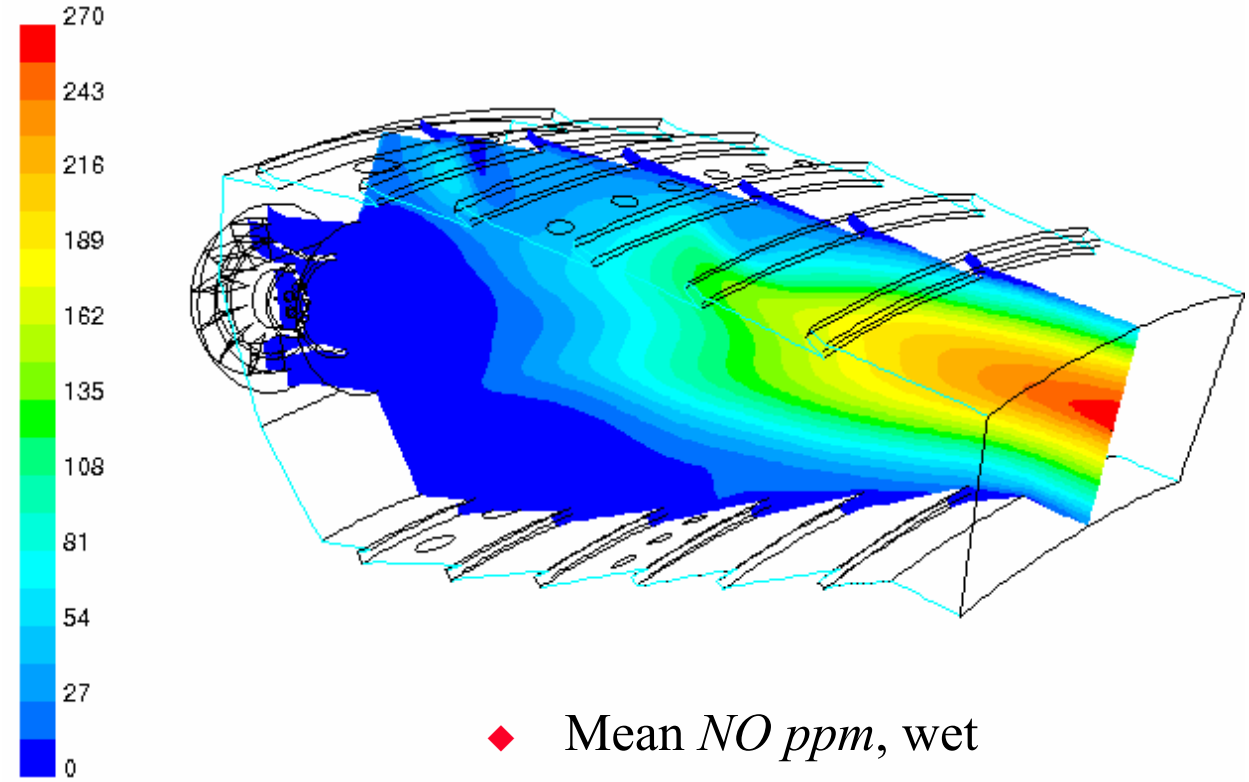
Gas Phase Combustion Modeling



- Mean temperature (*K*) contour plot

NOx Modeling

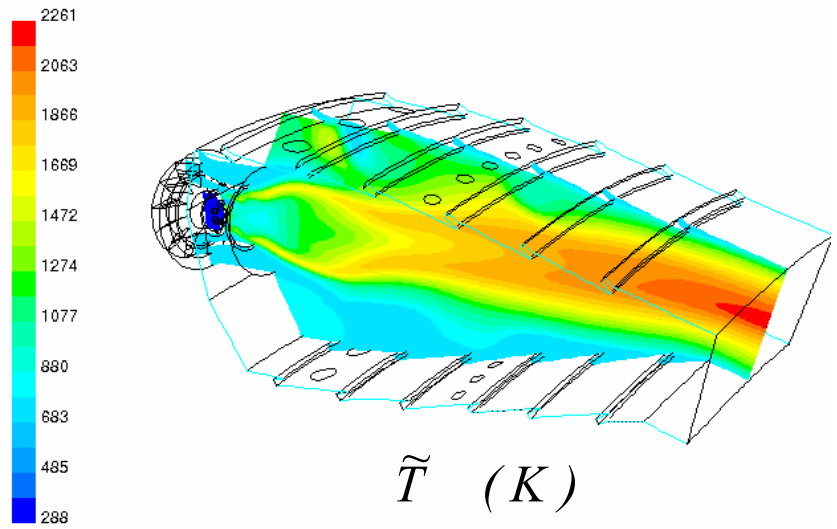
- Thermal and prompt *NOx:*
  - Zeldovich thermal *NO* dominant: $\frac{d[NO]}{dt} = 2k[O][N_2], \quad k = Ae^{-E/RT}$
- Species and temperature from Laminar Flamelet model
- Post-processed: assumed shape β pdf



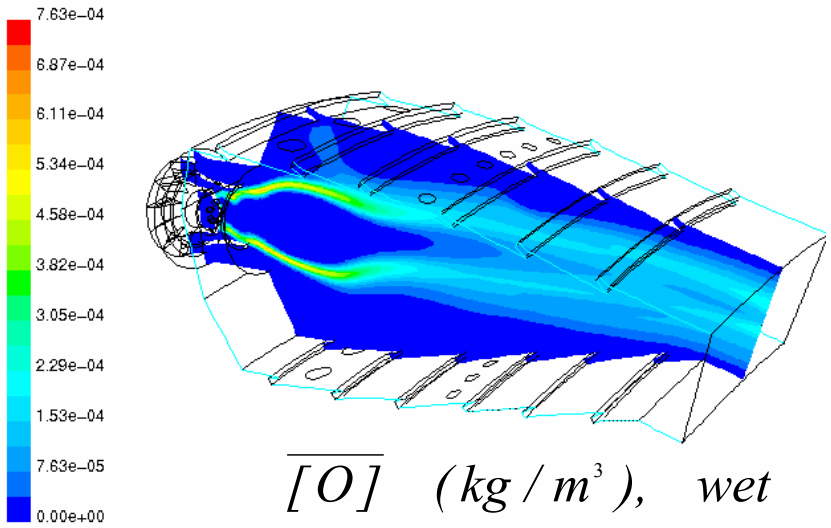
- Mean *NO ppm*, wet

NOx Modeling

- Laminar Flamelet model

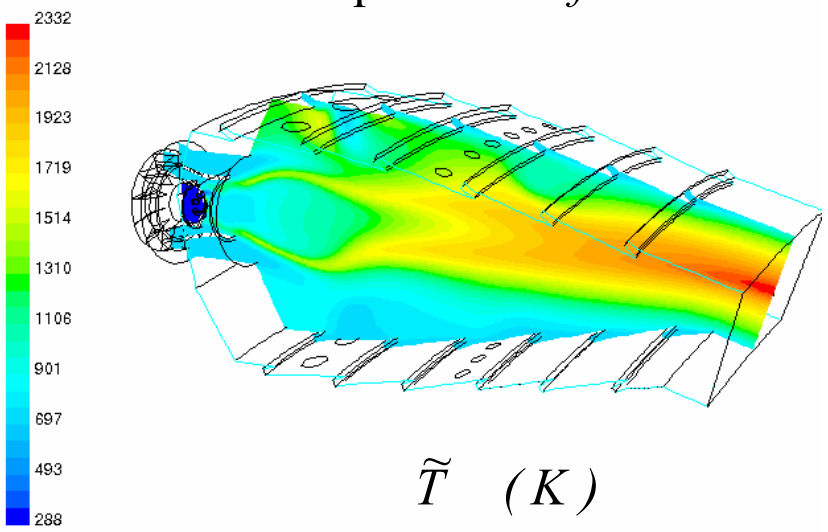


$\tilde{T}$ (K)

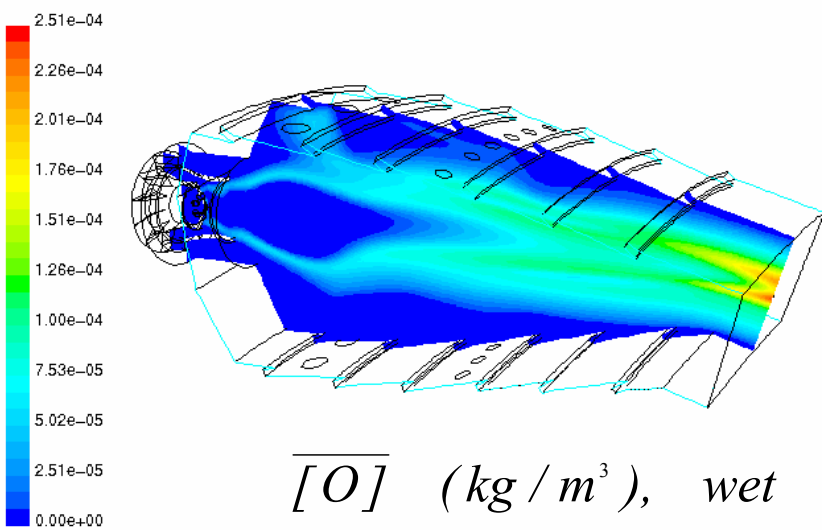


$\overline{[O]}$ (kg/m^3), wet

- Equilibrium *f* model



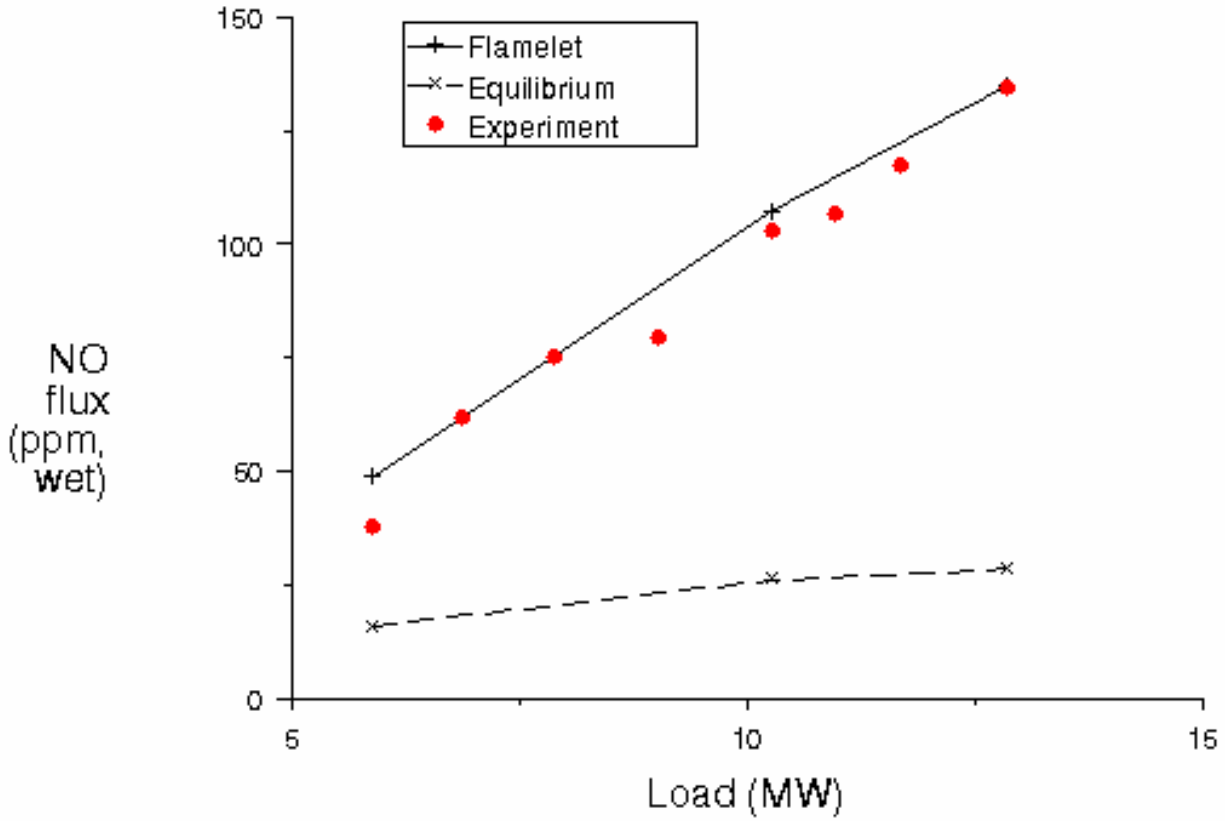
$\tilde{T}$ (K)



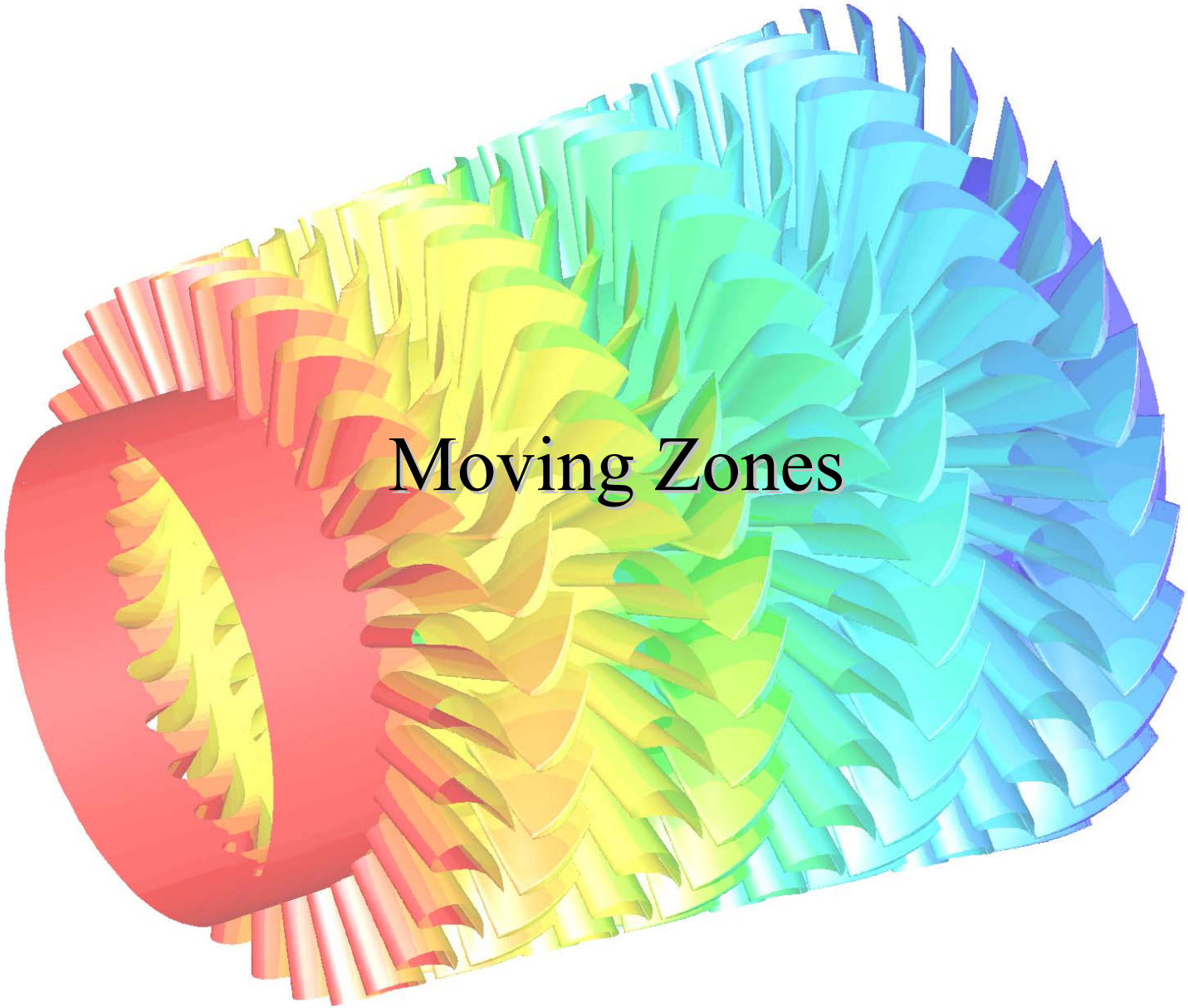
$\overline{[O]}$ (kg/m^3), wet

NOx Modeling

- Plot of *NO* flux exiting outlet vs. combustor load



$$NO\ exit\ \ flux = 10^6 \frac{\int \rho X_{NO} \vec{V} \cdot d\vec{A}}{\int \rho \vec{V} \cdot d\vec{A}}$$



Moving Zones

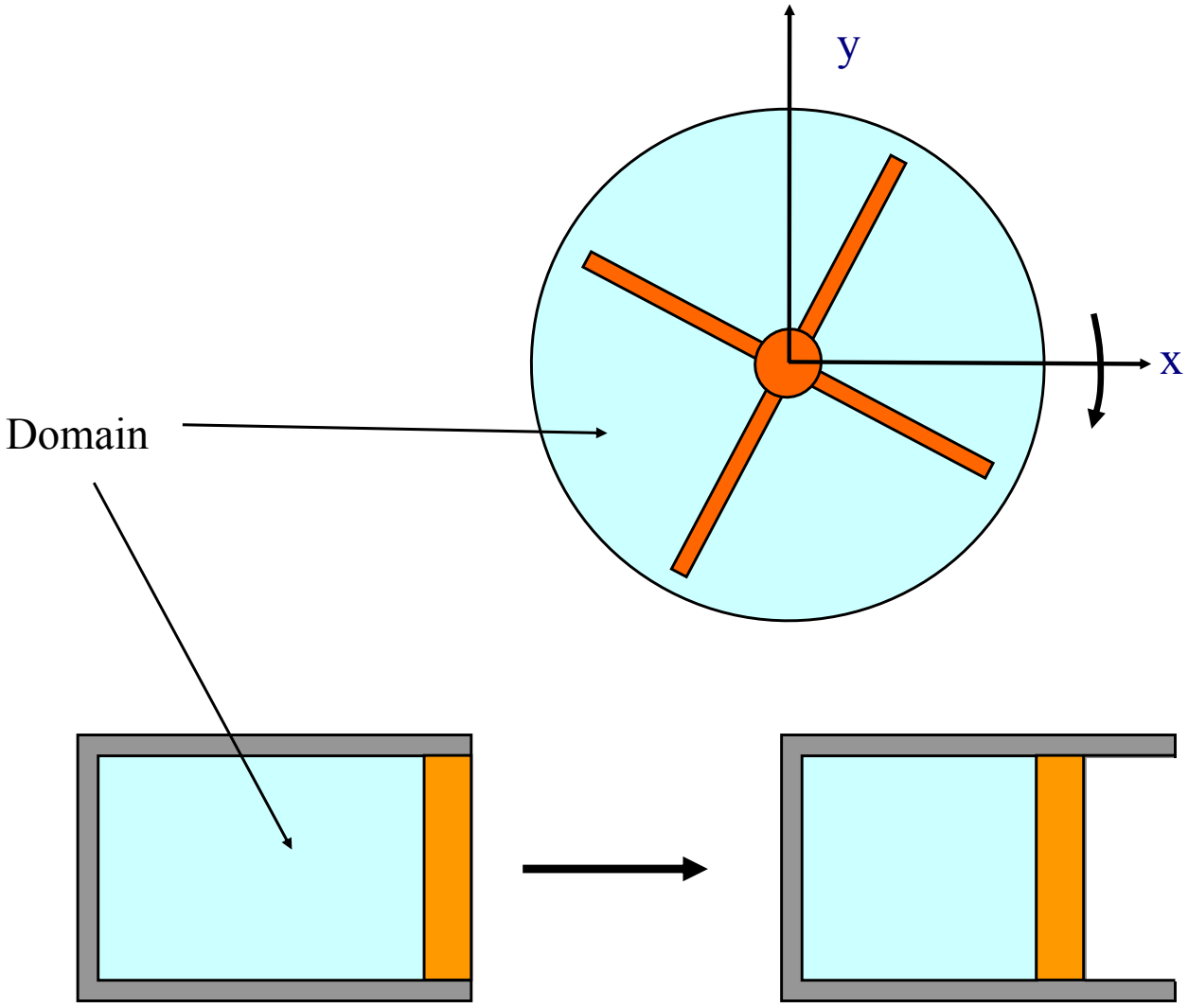
Outline

- **Introduction and Overview of Modeling Approaches**
- Single-Reference Frame (SRF) Model
- Multiple Zones and Multiple-Reference Frame Model (MRF)
- Mixing Plane Model (MPM)
- Sliding Mesh Model (SMM)
- Dynamic Mesh (DM) Model
- Summary
- Appendix

Introduction

- Many engineering problems involve flows through domains which contain **translating** or **rotating** components
  - Examples – **Translational motion**:
    - Train moving in a tunnel, longitudinal sloshing of fluid in a tank, etc.
  - Examples – **Rotational motion**:
    - Flow though propellers, axial turbine blades, radial pump impellers, etc.
- There are two basic modeling approaches for moving domains:
  - If the domain does not change shape as it moves (rigid motion), we can solve the equations of fluid flow in a **moving reference frame**.
    - Additional acceleration terms added to the momentum equations
    - Solutions become **steady** with respect to the moving reference frame
    - Can couple with stationary domains through interfaces
  - If the domain changes shape (deforms) as it moves, we can solve the equations using dynamic mesh (DM) techniques
    - Domain position and shape are functions of time
    - Solutions are inherently **unsteady**

Moving Reference Frame versus Dynamic Mesh



Moving Reference Frame – Domain moves with rotating coordinate system

Dynamic Mesh – Domain changes shape as a function of time

Overview of Modeling Approaches

- Single Reference Frame (SRF)
  - Entire computational domain is referred to a moving reference frame
- Multiple Reference Frame (MRF)
  - Selected regions of the domain are referred to moving reference frames
  - Interaction effects are **ignored** → steady-state
- Mixing Plane (MPM)
  - Influence of neighboring regions accounted for through use of a mixing plane model at rotating/stationary domain interfaces
  - Circumferential non-uniformities in the flow are **ignored** → steady-state
- Sliding Mesh (SMM)
  - Motion of specific regions accounted for by a mesh motion algorithm
  - Flow variables interpolated across a sliding interface
  - Unsteady problem - can capture all interaction effects with complete fidelity, but more computationally expensive than SRF, MRF, or MPM
- Dynamic Mesh (DM)
  - Like sliding mesh, except that domains are allowed to move and deform with time
  - Mesh deformation accounted for using spring analogy, remeshing, and mesh extrusion techniques

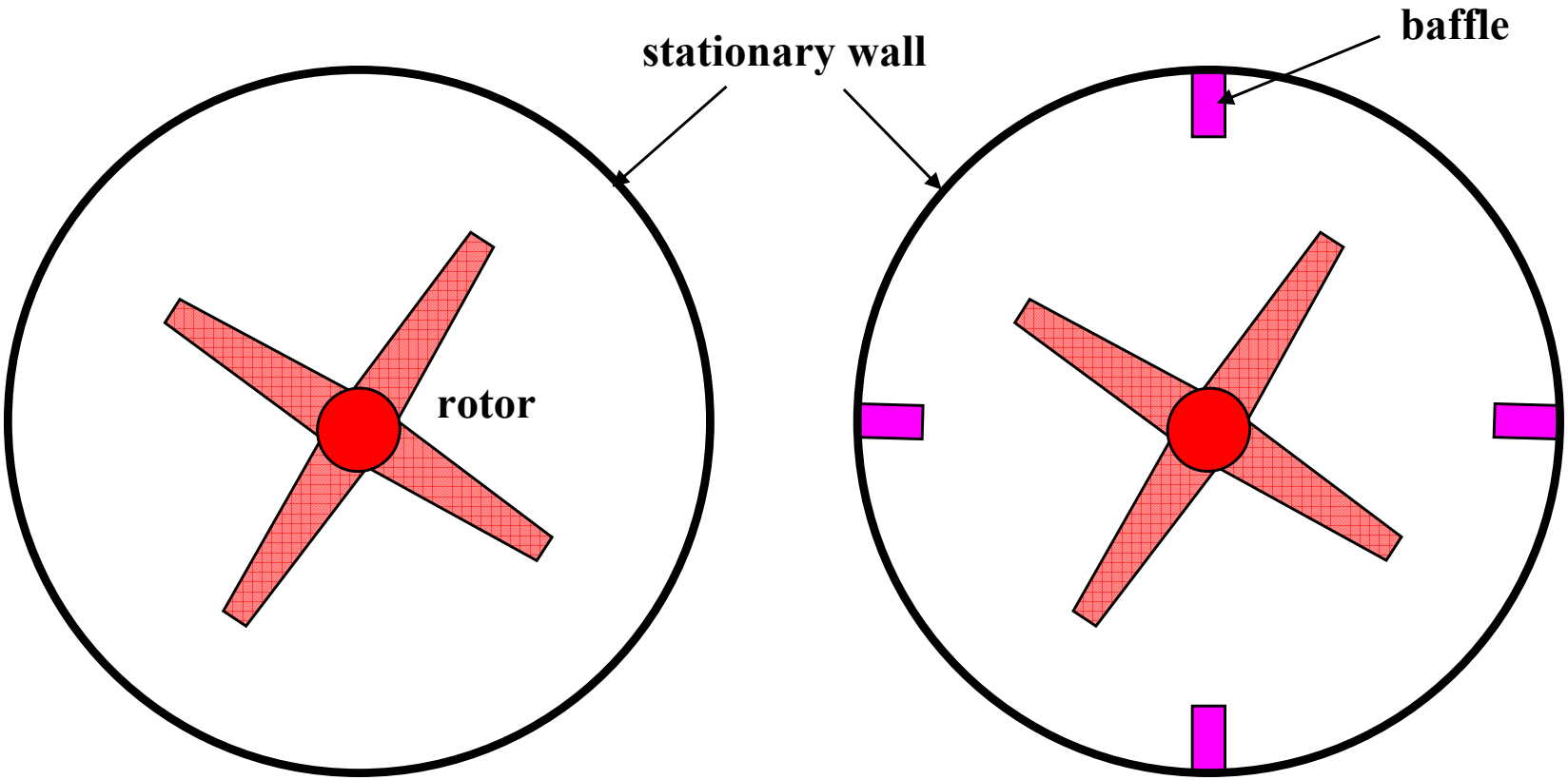
Outline

- Introduction and Overview of Modeling Approaches
- **Single-Reference Frame (SRF) Model**
- Multiple Zones and Multiple-Reference Frame Model (MRF)
- Mixing Plane Model (MPM)
- Sliding Mesh Model (SMM)
- Dynamic Mesh (DM) Model
- Summary
- Appendix

Introduction to SRF Modeling

- SRF assumes a **single fluid domain** which rotates with a constant speed with respect to a specified axis.
- Why use a rotating reference frame?
  - Flow field which is **unsteady** when referred to a stationary frame becomes **steady** in the rotating frame
  - Steady-state problems are easier to solve...
    - simpler BCs
    - low computational cost
    - easier to post-process and analyze
- Boundaries zones must conform to the following requirements:
  - Boundaries which **move with the fluid domain** may assume **any shape**
  - Boundaries **which are stationary** (with respect to the fixed frame) **must be surfaces of revolution**
- Can employ rotationally-periodic boundaries for efficiency (reduced domain size)

Stationary Walls in SRF Models



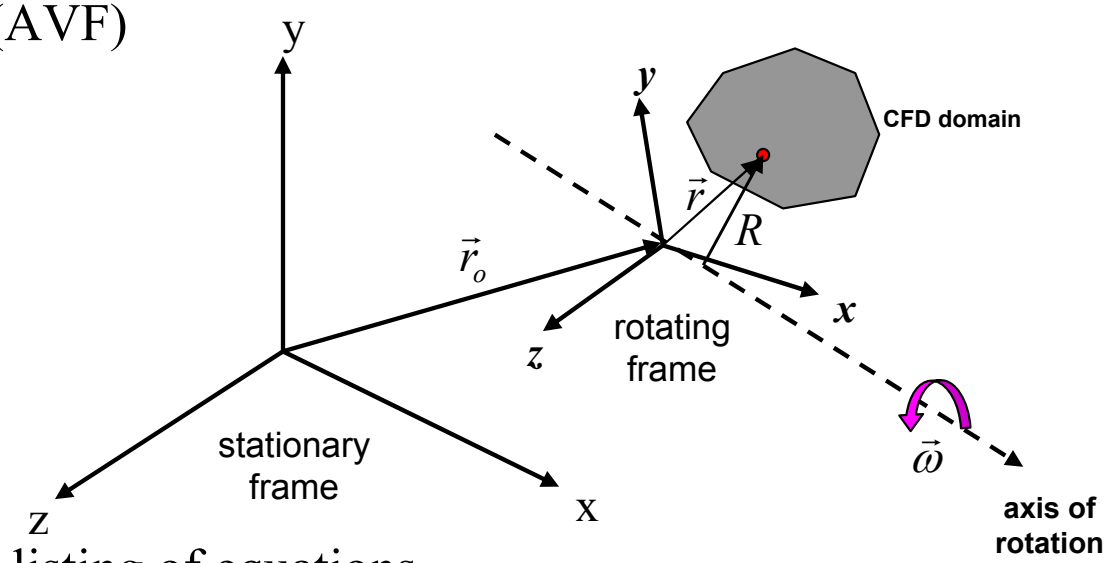
Correct

Wrong!

Wall with baffles ***not*** a surface of revolution!

N-S Equations: Rotating Reference Frames

- Equations can be solved in absolute or rotating (relative) reference frame.
  - Relative Velocity Formulation (RVF)
    - Obtained by transforming the stationary frame N-S equations to a rotating reference frame
    - Uses the *relative velocity* and *relative total internal energy* as the dependent variables
  - Absolute Velocity Formulation (AVF)
    - Derived from the relative velocity formulation
    - Uses the *absolute velocity* and *absolute total internal energy* as the dependent variables
  - Rotational source terms appear in momentum equations.
  - Refer to Appendix for a detailed listing of equations



The Velocity Triangle

- The relationship between the absolute and relative velocities is given by

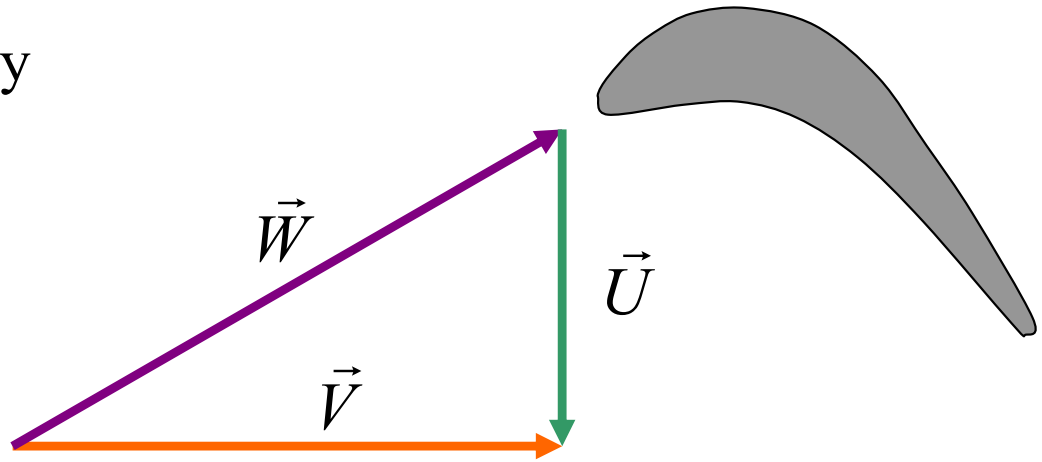
$$\vec{V} = \vec{W} + \vec{U}$$

$$\vec{U} \equiv \vec{\omega} \times \vec{r}$$

- In turbomachinery, this relationship can be illustrated using the laws of vector addition. This is known as the Velocity Triangle

$\vec{V}$ = Absolute Velocity

$\vec{W}$ = Relative Velocity



Comparison of Formulations

- Relative Velocity Formulation: x-momentum equation

$$\frac{\partial \rho w_x}{\partial t} + \nabla \cdot \rho \vec{W} w_x = -\frac{\partial p}{\partial x} + \nabla \cdot \vec{\tau}_{vrx} - \rho(2\vec{\omega} \times \vec{W} + \vec{\omega} \times \vec{\omega} \times \vec{r}) \cdot \hat{i}$$

Coriolis acceleration Centripetal acceleration

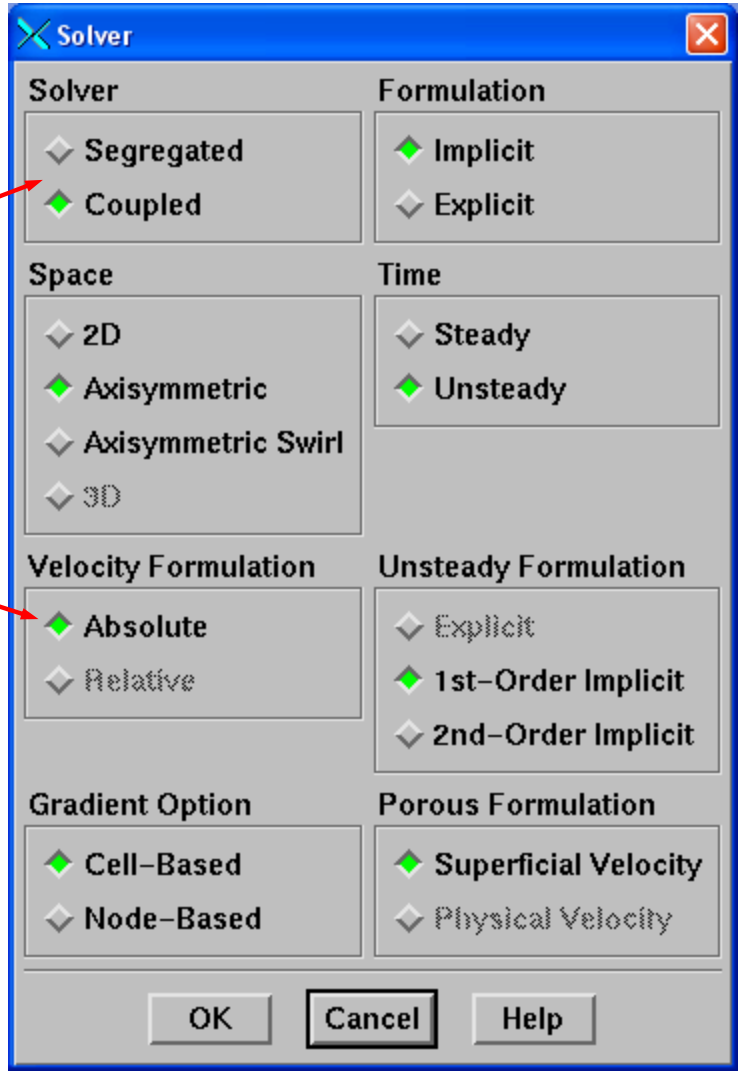
- Absolute Velocity Formulation: x-momentum equation

$$\frac{\partial \rho v_x}{\partial t} + \nabla \cdot \rho \vec{W} v_x = -\frac{\partial p}{\partial x} + \nabla \cdot \vec{\tau}_{vx} - \rho(\vec{\omega} \times \vec{V}) \cdot \hat{i}$$

Coriolis + Centripetal accelerations

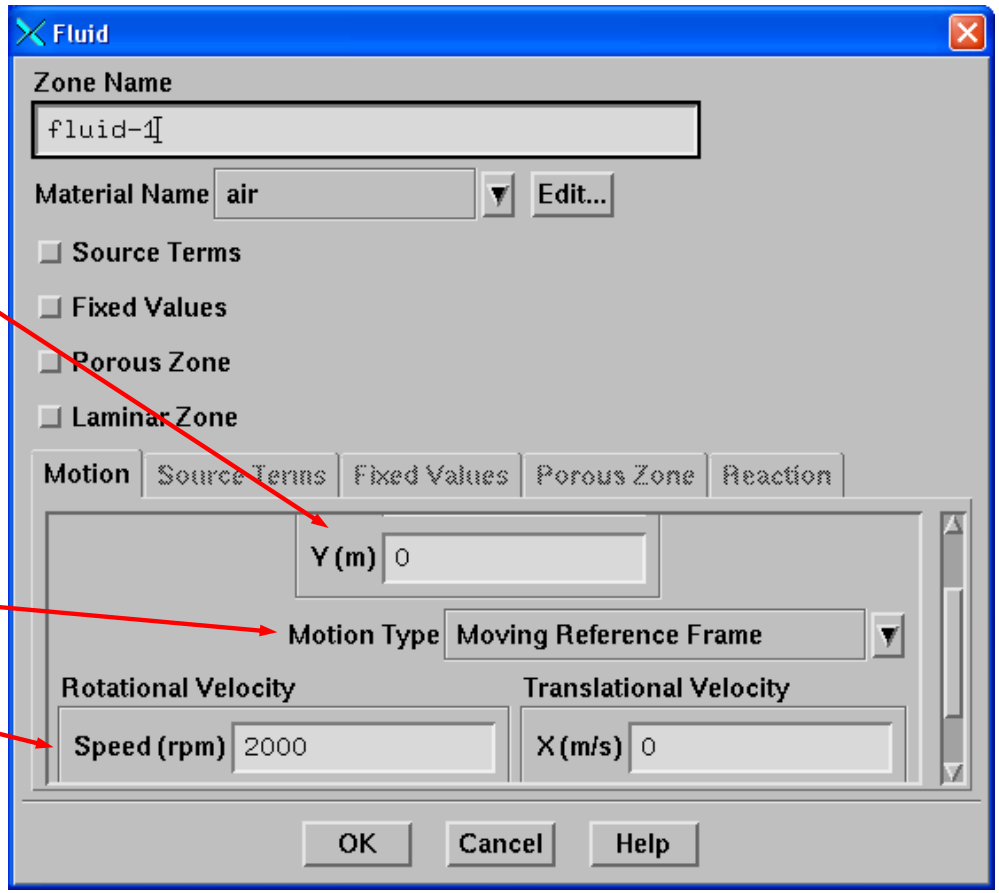
SRF Set-up: Solver

- Same considerations for general flow field modeling apply to SRF solver choice
  - Segregated – Incompressible, low-speed compressible flows
  - Coupled – High-speed compressible flows
- Velocity Formulation recommendations
  - Use absolute velocity formulation (AVF) when inflow comes from a stationary domain
  - Use relative velocity formulation (RVF) with closed domains (all surfaces are moving) or if inflow comes from a rotating domain
    - NOTE: RVF is only available in the segregated solver
  - In many cases, either can be used successfully



SRF Set-up: Fluid BCs

- Use fluid BC panel to define rotational axis origin and direction vector for rotating reference frame
  - Direction vectors should be unit vectors but Fluent will normalize them if they aren't
- Select Moving Reference Frame as the Motion Type for SRF
- Enter Rotational Velocity
  - Rotation direction defined by right-hand rule
  - Negative speed implies rotation in opposite direction



SRF Set-up: Inlet/Outlet Boundaries

- Velocity Inlets:
  - Absolute or relative velocities may be defined regardless of formulation.
- Pressure Inlet:
  - Definition of total pressure depends on velocity formulation:

$$p_{t,abs} = p + \frac{1}{2}\rho W^2 \quad \textbf{incompressible, RVF}$$

$$p_{t,rel} = p + \frac{1}{2}\rho V^2 \quad \textbf{incompressible, AVF}$$

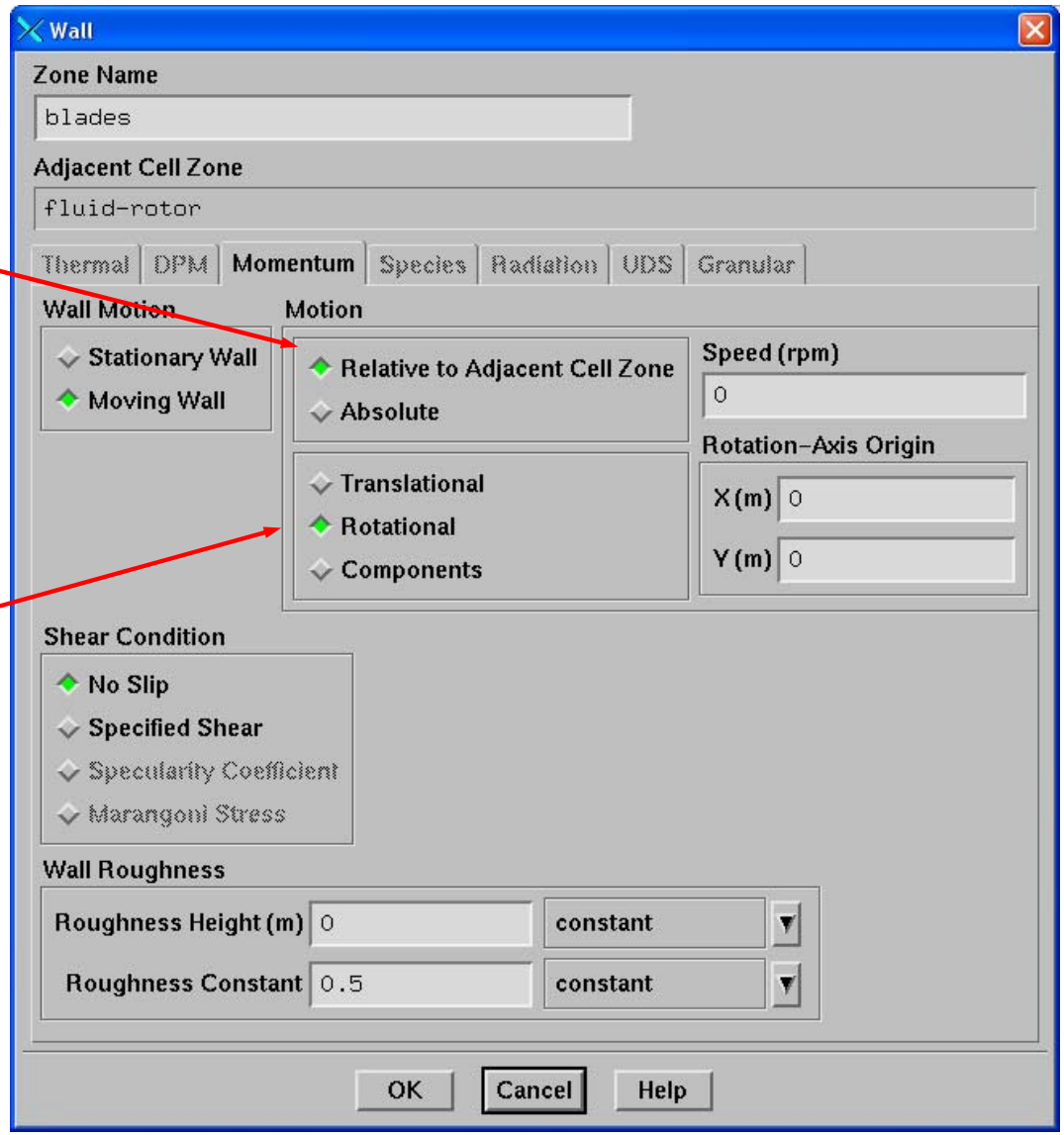
- Pressure Outlet:
  - For axial flow problems with swirl at outlet, radial equilibrium assumption option can be applied such that:
    - Specified pressure is hub pressure

$$\left.\frac{\partial p}{\partial R}\right|_{outlet} = \rho \frac{V_\theta^2}{R}$$

- Other BCs for SRF problems
  - Non-reflecting BCs
  - Target mass flow outlet

Wall BCs

- For moving reference frames, you can specify the wall motion in either the absolute or relative frames
- Recommended specification of wall BCs for all moving reference frame problems…
  - **For <u>stationary surfaces</u>** (in the absolute frame) use **zero Rotational speed, Absolute**
  - **For <u>moving surfaces</u>**, use **zero Rotational speed, Relative to Adjacent Cell Zone**



Solution Strategies for SRF Problems

- High degree of coupling between momentum equations when rotational terms are large can make convergence difficult.
- **Strategies**
  - Ensure that the mesh is sufficiently refined to resolve large gradients in pressure and swirl velocity.
  - Use reasonable values for initial conditions
    - New in Fluent 6.2: FMG initialization – provides Euler solution as initial condition using the coupled-explicit solver (good for rotating machinery problems)
  - Begin the calculations using a low rotational speed, increasing the rotational speed gradually in order to reach the final desired operating condition.
  - Begin calculations using first order discretization and switch to second order.
  - For the Segregated solver
    - Use the PRESTO! Or Body-Force Weighted schemes which are well-suited for the steep pressure gradients involved in rotating flows.
    - Reduce the under-relaxation factors for the velocities, perhaps to 0.3-0.5.

Outline

- Introduction and Overview of Modeling Approaches
- Single-Reference Frame (SRF) Model
- **Multiple Zones and Multiple-Reference Frame Model (MRF)**
- Mixing Plane Model (MPM)
- Sliding Mesh Model (SMM)
- Dynamic Mesh (DM) Model
- Summary
- Appendix

Multiple Zone Modeling

- Many rotating machinery problems involve stationary components which **cannot** be described by surfaces of revolution (SRF **not valid**).
- Systems like these can be solved by dividing the domain into **multiple fluid zones** – some zones will be rotating, others stationary.
- For multiple zone models, we can apply one of the following approaches:
  - Multiple reference frame model (MRF)
    - Simplified interface treatment - rotational interaction between reference frames is not accounted for.
  - Mixing plane model (MPM)
    - Interaction between reference frames are approximated through circumferential averaging at fluid zone interfaces (mixing planes).
  - Sliding mesh model (SMM)
    - Accurately models the relative motion between moving and stationary zones at the expense of more CPU time (inherently **unsteady**).

Introduction to the MRF Model

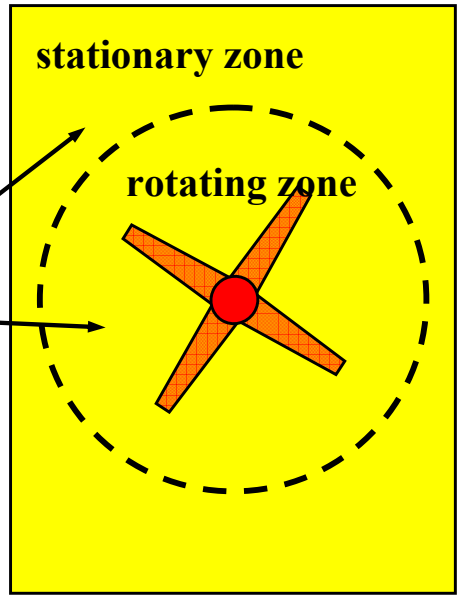
- The domain is subdivided into stationary and rotating fluid zones.
  - More than one rotating zone is permitted.
  - Zones can rotate at different speeds.
- Governing equations are solved in each fluid zone.
  - SRF equations used in rotating zones.
  - At the interfaces between the rotating and stationary zones, appropriate transformations of the velocity vector and velocity gradients are performed to compute fluxes of mass, momentum, energy, and other scalars.
  - Flow is assumed to be steady in each zone (clearly an approximation).
- MRF ignores the relative motions of the zones with respect to each other.
  - Does not account for fluid dynamic interaction between stationary and rotating components.
  - For this reason MRF is often referred to as the “frozen rotor” approach.
- Ideally, the flow at the MRF interfaces should be relatively uniform or “mixed out.”

Implications of the MRF Model

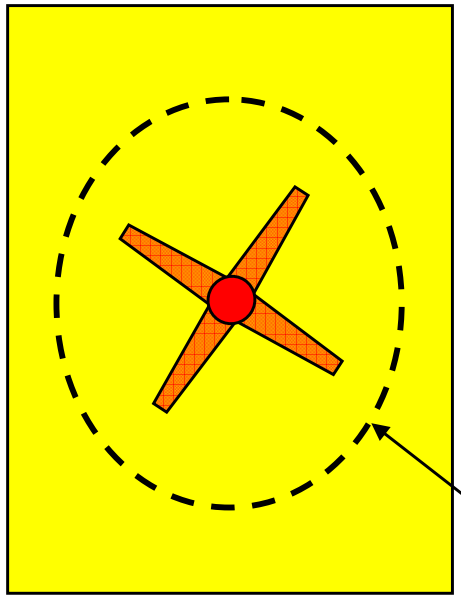
- Walls which are contained **within the rotating fluid zone interfaces** are assumed to be moving with the fluid zones and may assume any shape.
- The interface between a rotating zone and the adjacent stationary zone **must be a surface of revolution** with respect to the axis of rotation of the rotating zone.

Consider a mixing impeller inside a rectangular vessel:

- Problem can be described with two reference frames or zones.



Correct

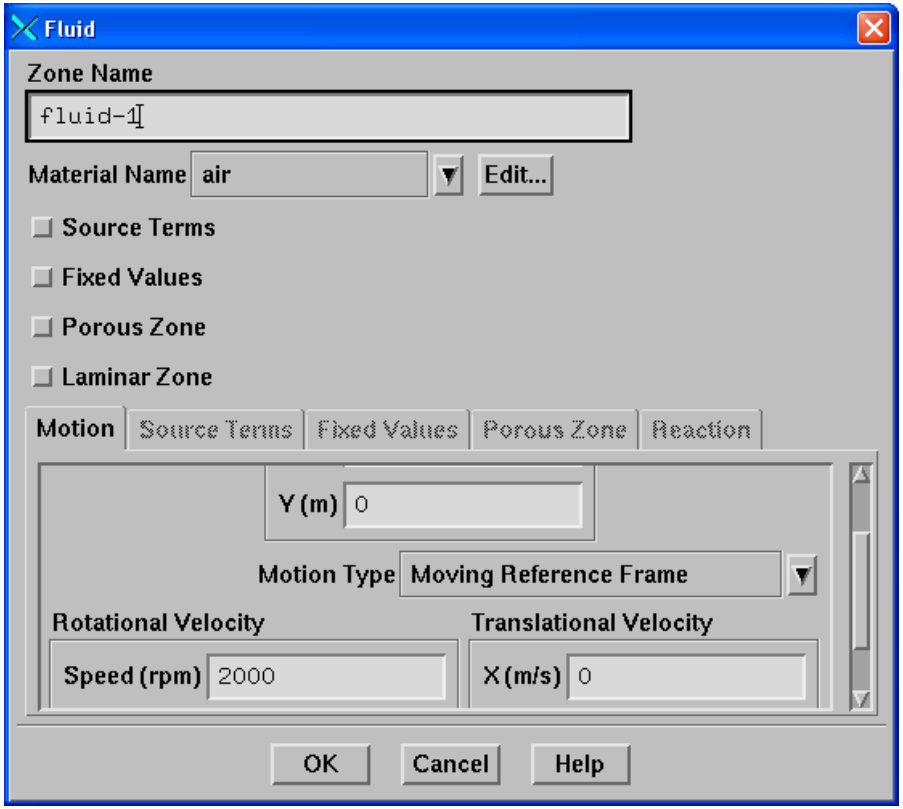


Wrong!

Interface is not a surface or revolution

MRF Set-Up

- Generate mesh with appropriate stationary and rotating fluid zones
  - Interfaces can be conformal or non-conformal
  - Non-Conformal Interfaces
    - Provides flexibility to switch to Sliding Mesh Model (SMM) easily.
    - Requires a Grid Interface to be defined.
- For each rotating fluid zone (Fluid BC), select Moving Reference Frame as the Motion Type and enter the rotational speed.
  - Identical to SRF except multiple zones can be defined.
- Set up for other BCs and Solver settings same as SRF model.



Outline

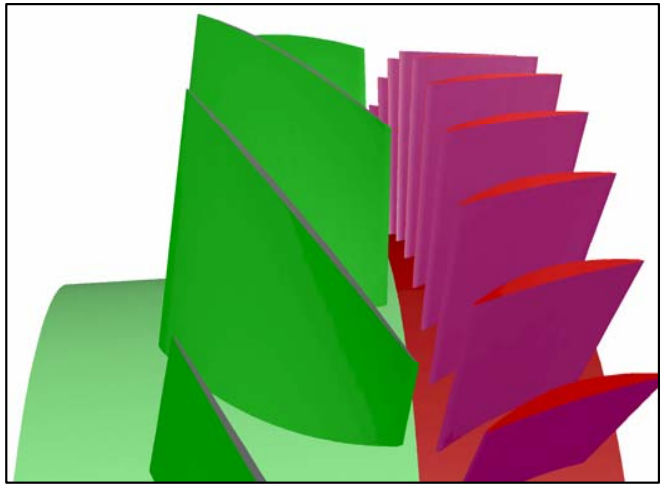
- Introduction and Overview of Modeling Approaches
- Single-Reference Frame (SRF) Model
- Multiple Zones and Multiple-Reference Frame Model (MRF)
- **Mixing Plane Model (MPM)**
- Sliding Mesh Model (SMM)
- Dynamic Mesh (DM) Model
- Summary
- Appendix

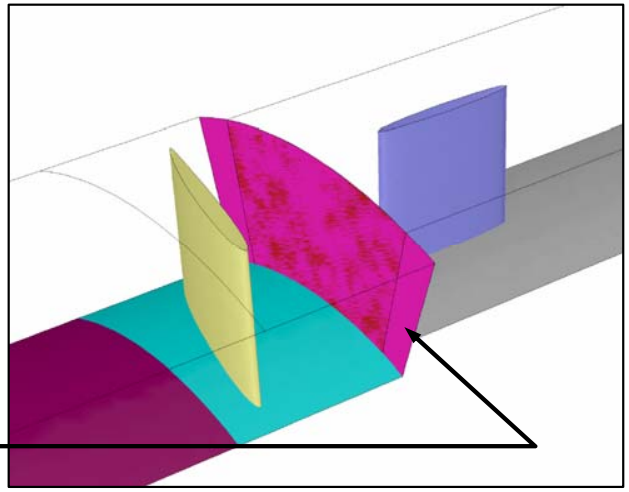
Introduction to Mixing Plane Model (MPM)

- The MPM was originally implemented to accommodate rotor/stator and impeller/vane flows in axial and centrifugal turbomachines.
  - Can also be applied to a more general class of problems.
- Typically, the domain is divided into rotating and stationary zones that correspond to the rotors and stators.
  - Multiple rotor/stator 'stages' are allowed.
- Governing equations are solved in each domain.
  - Flow is assumed steady in each domain.
  - The interfaces between the domains are called the mixing planes.
  - Circumferentially averaged profiles of flow variables are computed at the mixing planes.
  - The profiles are used as boundary conditions to the adjacent domains.
  - As the solution converges, the mixing plane boundary conditions will adjust to the prevailing flow conditions.

MPM for Turbomachinery Problems

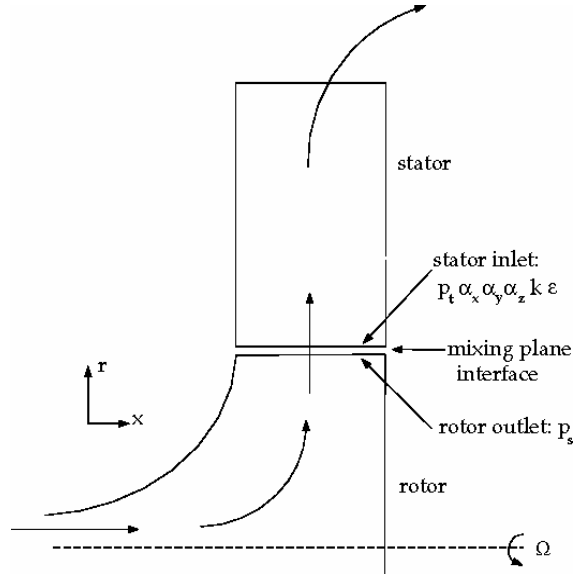
- For multistage turbomachinery problems,
  - The stage boundary conditions are often known (e.g. inlet total pressure and temperature and stage outlet static pressure) but not the **inter-stage conditions.**
  - Blade counts will generally **not be the same** from one row to the next.
- MRF could be used only if we have equal periodic angles for each row.
- The MPM requires only **a single blade passage per blade row** regardless of the number of blades.
  - This is accomplished by mixing out (averaging) the circumferential non-uniformities in the flow at the inter-stage (mixing plane) interface.



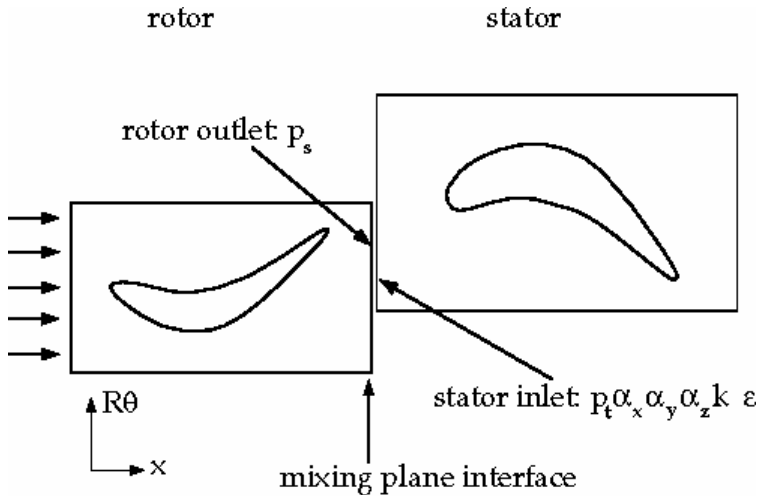


MPM Mixing Planes

- A mixing plane is an interface that consists of the *outlet* of an upstream domain and the *inlet* to the adjacent downstream domain.
  - The inlet/outlet boundaries must be assigned BC types in one of the following combinations:
    - Pressure-Inlet / Pressure-Outlet
    - Velocity-Inlet / Pressure-Outlet
    - Mass-Flow-Inlet / Pressure-Outlet
- The MPM has been implemented for both axial and radial turbomachinery blade rows.
  - For axial machines, radial profiles are used.
  - For radial (centrifugal) machines, axial profiles are used.



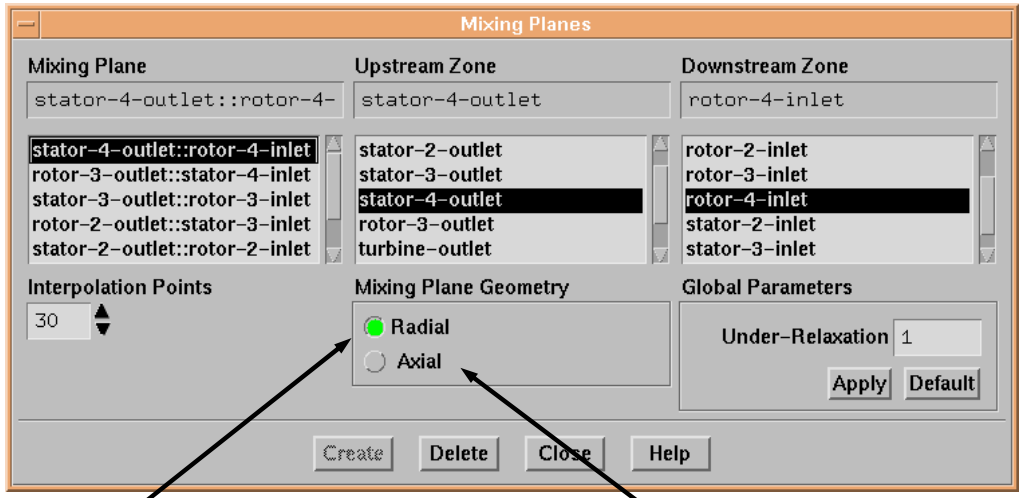
radial machines



axial machines

MPM Setup

- Set fluid zones as Moving Reference Frames and define zone velocities.
- Assign appropriate BC types to inlet-outlet boundary pairs.
- Select upstream and downstream zones which will comprise mixing plane pair.
- Set the number of points for profile interpolation.
  - Should be about the same axial/radial resolution as the mesh.
- Mixing Plane Geometry determines method of circumferential averaging.
  - Choose Radial for axial flow machines.
  - Choose Axial for radial flow machines.
- Mixing plane controls
  - **Under-Relaxation** - Profile changes are underrelaxed using factor between 0 and 1



$$\overline{\phi}\big|_z(r) = \frac{1}{\Delta\theta_p}\int_{\Delta\theta_p}\phi(r,\theta)d\theta$$

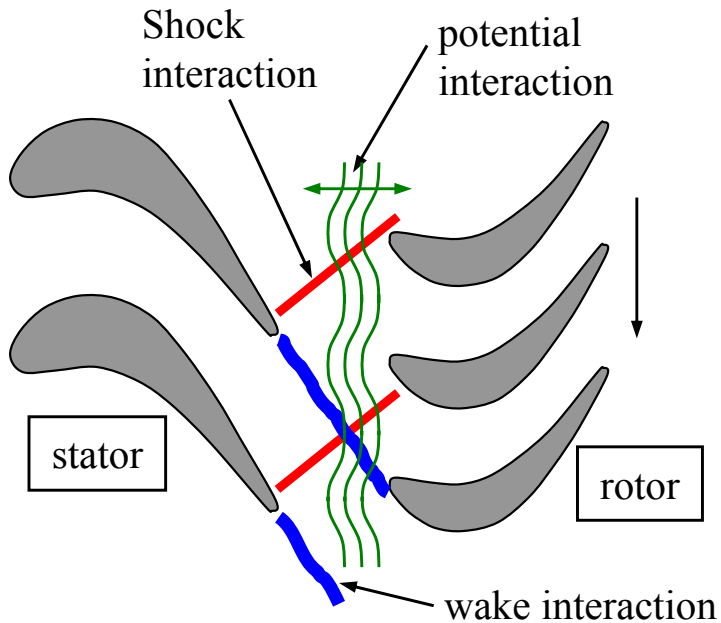
$$\overline{\phi}\big|_r(z) = \frac{1}{\Delta\theta_p}\int_{\Delta\theta_p}\phi(z,\theta)d\theta$$

Outline

- Introduction and Overview of Modeling Approaches
- Single-Reference Frame (SRF) Model
- Multiple Zones and Multiple-Reference Frame Model (MRF)
- Mixing Plane Model (MPM)
- **Sliding Mesh Model (SMM)**
- Dynamic Mesh (DM) Model
- Summary
- Appendix

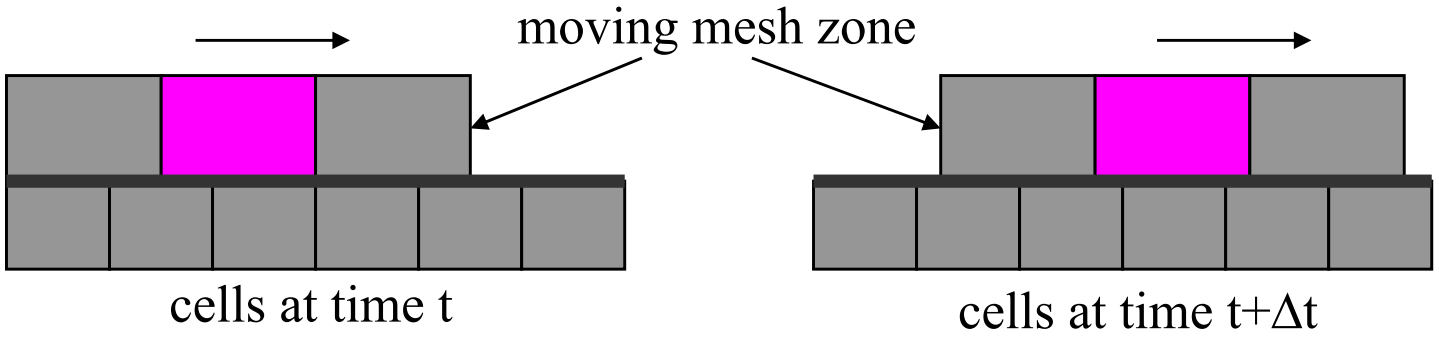
Introduction to Sliding Mesh Model (SMM)

- The relative motion of stationary and rotating components in a turbo-machine will give rise to **unsteady interactions**. These interactions are generally classified as follows:
  - **Potential interactions** (pressure wave interactions)
  - **Wake interactions**
  - **Shock interactions**
- Both MRF and MPM neglect unsteady interaction entirely and thus are limited to flows where these effects are weak.
- If unsteady interaction can not be neglected, we can employ the Sliding Mesh Model to account for the relative motions of the stationary and rotating components.



Implications of the SMM

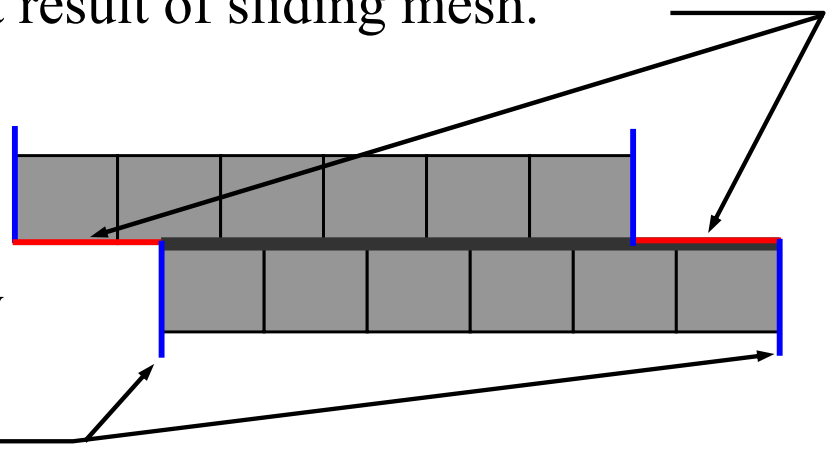
- Like the MRF model, the domain is divided into moving and stationary subdomains.
- Unlike MRF, the mesh in each subdomain moves relative to one another, and thus the mathematical problem is inherently **unsteady.**

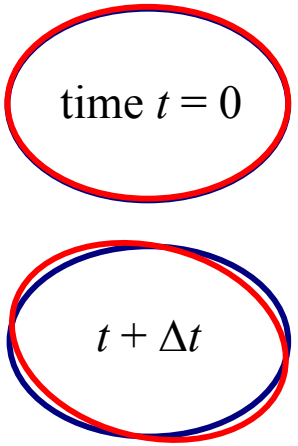


- The governing equations are solved in the inertial reference frame for absolute quantities.
  - For each time step, the meshes *are moved* and the fluxes at the sliding interfaces are recomputed.
  - Relative velocity formulation does not apply.

Sliding Interfaces

- Sliding interfaces must follow the same rules as MRF problems and must be defined as non-conformal:
  - The interface between a rotating subdomain and the adjacent stationary/rotating subdomain **must be a surface of revolution** with respect to the axis of rotation of the rotating subdomain.
    - Many failures of sliding mesh models can be traced to interface geometries which are not surfaces of revolution!
  - Any translation of the interface cannot be normal to itself.
- Zones are exposed as a result of sliding mesh.
  - Can either be:
    - Periodic
    - Walls
  - If periodic, boundary zones must also be periodic.

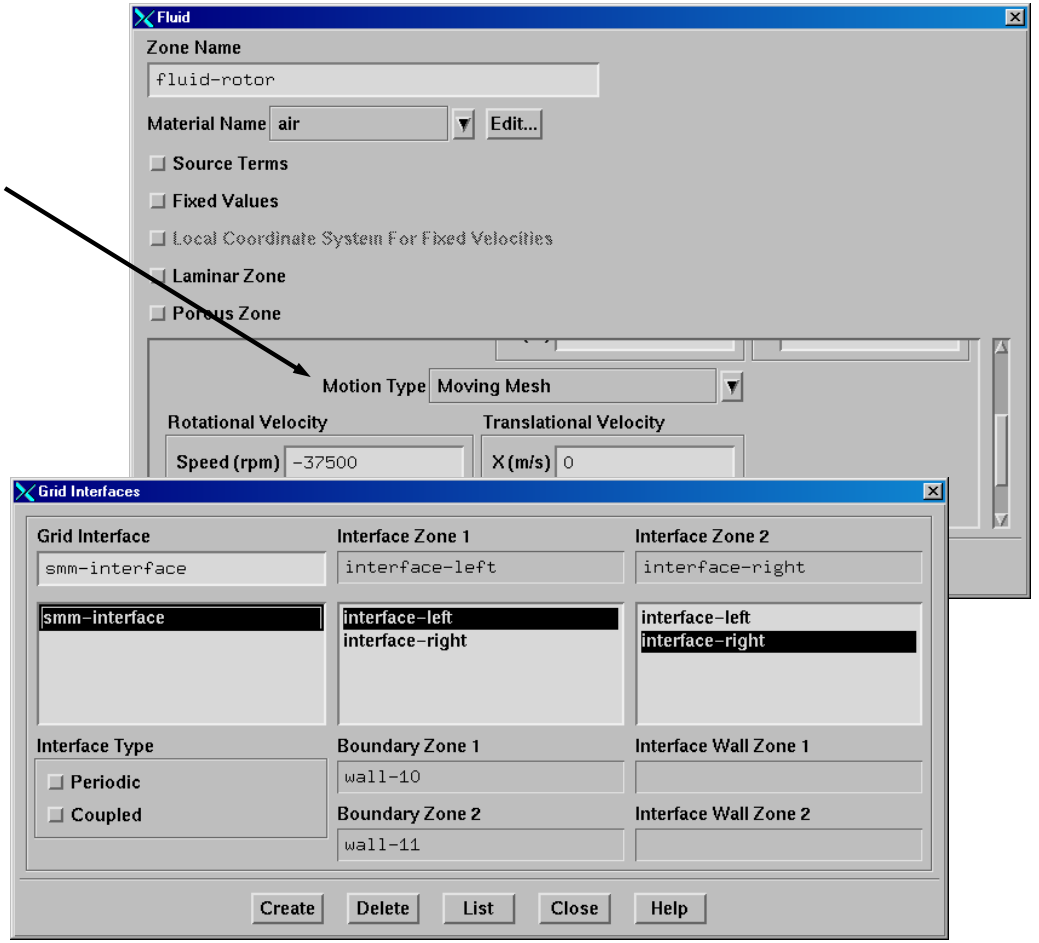




Elliptic interface is not a surface of revolution.

SMM Setup

- Enable unsteady solver.
- Define overlapping zones as Interface
- For moving zones, select Moving Mesh as Motion Type in Fluid BC panel.
  - By default, velocity of walls are zero relative to the adjacent mesh's motion.
- For each interface zone pair, create a non-conformal interface
  - Enable Periodic option if sliding/rotating motion is periodic.
  - Enable Coupled for conjugate heat transfer.
- Other BCs are same as SRF, MRF models



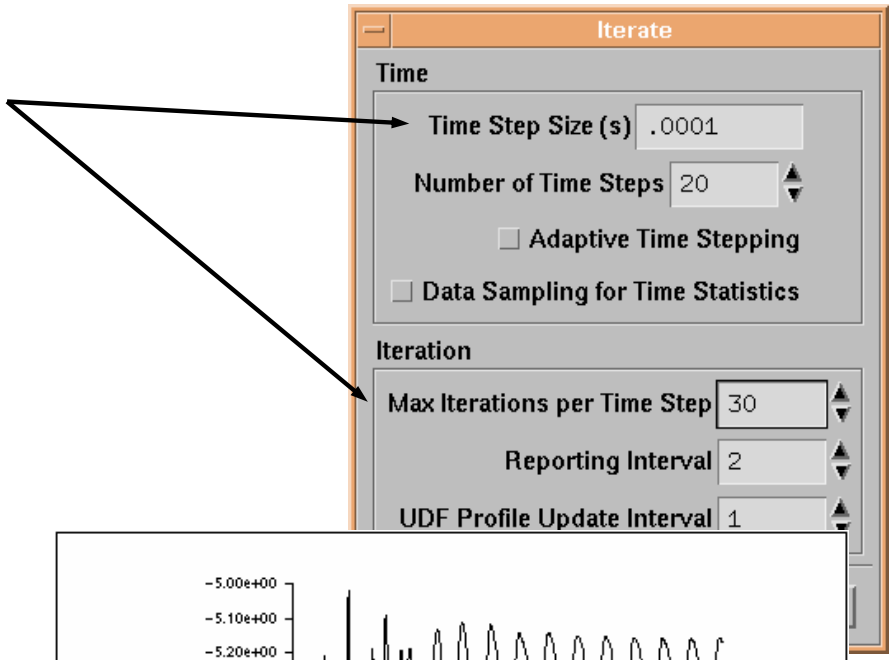
Solving SMM Problems

- Choose appropriate Time Step Size and Max Iterations Per Time step to ensure good convergence with each time step.
  - Time Step Size should be no larger than the time it takes for a moving cell to advance past a stationary point:

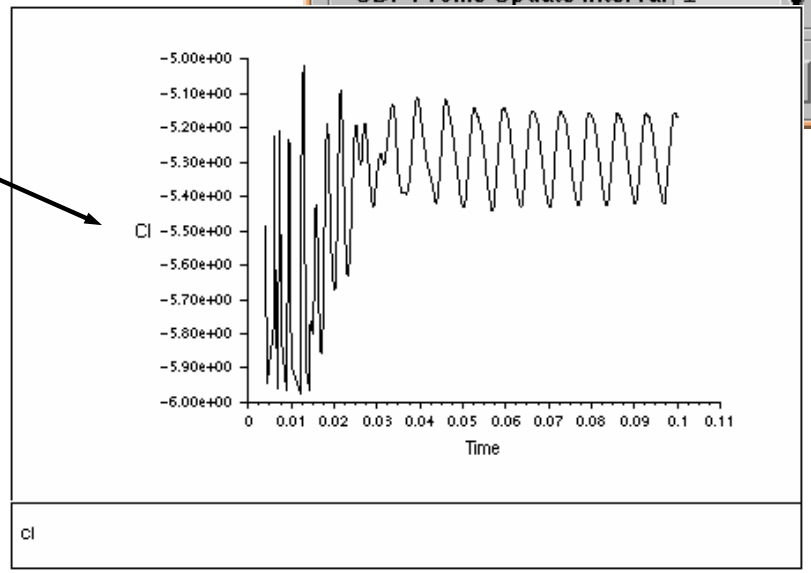
$$\Delta t \approx \frac{\Delta s}{\omega R}$$

Δs = average cell size

ωR = translational speed



- Advance the solution until the flow becomes **time-periodic** (pressures, velocities, etc., oscillate with a repeating time variation).
  - Usually requires several revolutions of the grid.
  - Good initial conditions can reduce the time needed to achieve time-periodicity

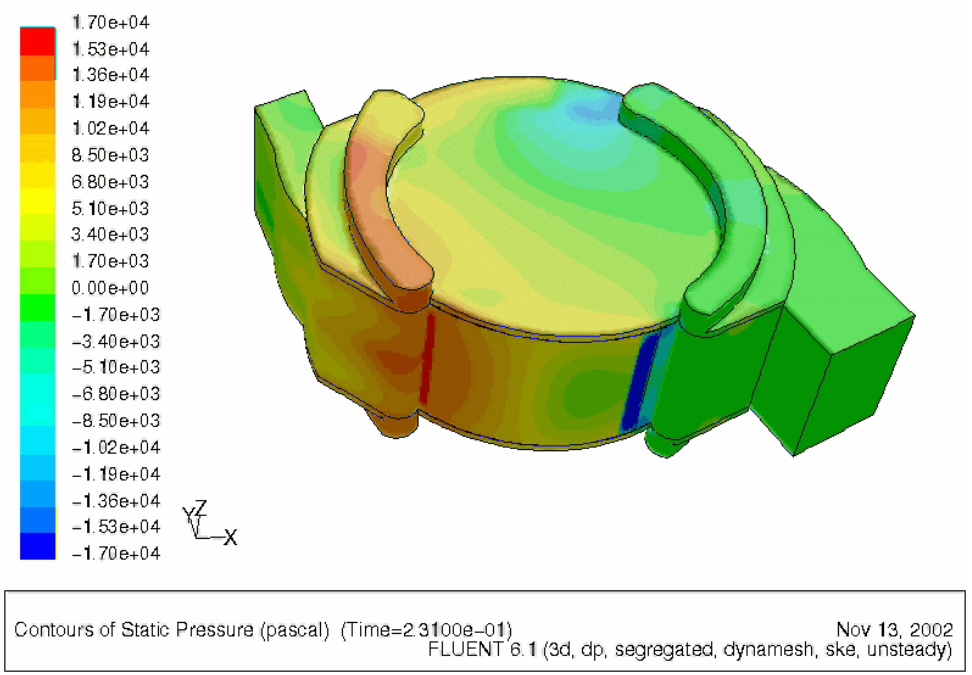


Outline

- Introduction and Overview of Modeling Approaches
- Single-Reference Frame (SRF) Model
- Multiple Zones and Multiple-Reference Frame Model (MRF)
- Mixing Plane Model (MPM)
- Sliding Mesh Model (SMM)
- **Dynamic Mesh (DM) Model**
- Summary
- Appendix

What is the Dynamic Mesh (DM) Model?

- A method by which the solver (FLUENT) can be instructed to move boundaries and/or objects, and to adjust the mesh accordingly
- Examples:
  - Automotive piston moving inside a cylinder
  - A flap moving on an airplane wing
  - A valve opening and closing
  - An artery expanding and contracting



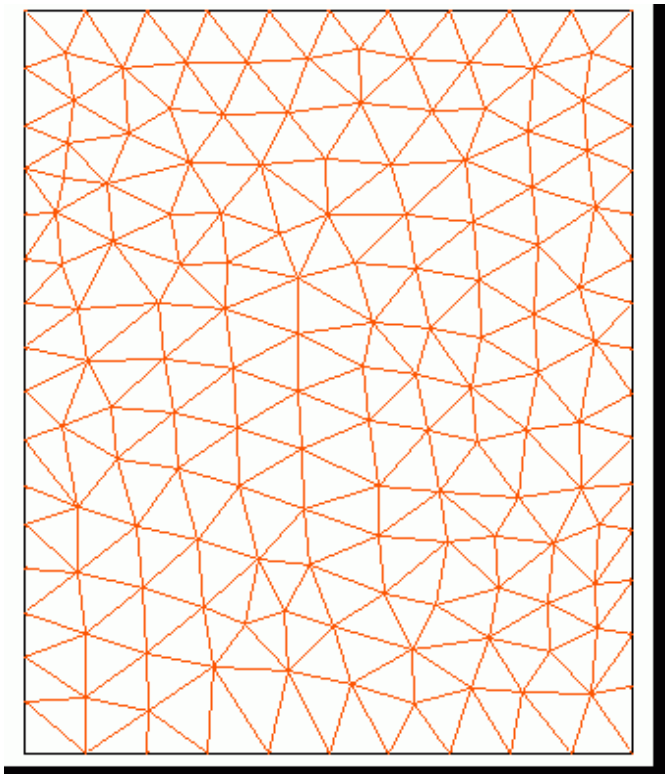
Volumetric fuel pump

Dynamic Mesh (DM) Model: Features

- Internal node positions are automatically calculated based on user specified boundary/object motion, cell type, and meshing schemes
  - Spring analogy (smoothing)
  - Local remeshing
  - Layering
  - 2.5 D
  - User defined mesh motion
- Boundaries/Objects motion can be moved based on:
  - In-cylinder motion (RPM, stroke length, crank angle, …)
  - Prescribed motion via profiles or UDF
  - Coupled motion based on hydrodynamic forces from the flow solution, via FLUENT's 6 DOF model.
- Different mesh motion schemes may be used for different zones. Connectivity between adjacent zones may be non-conformal.

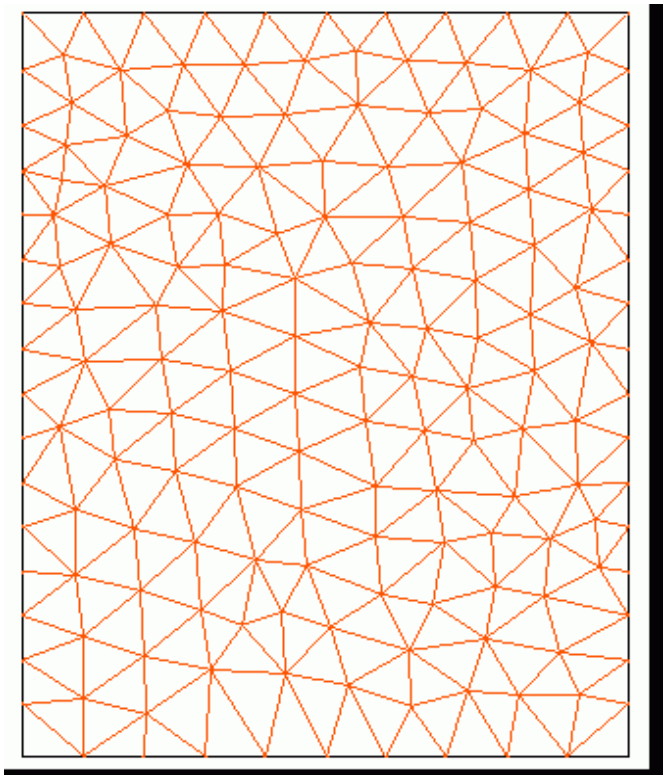
Spring Analogy (Spring Smoothing)

- The nodes move as if connected via springs, or as if they were part of a sponge;
- Connectivity remains unchanged;
- Limited to relatively small deformations when used as a stand-alone meshing scheme;
- Available for tri and tet meshes;
- May be used with quad, hex and wedge mesh element types, but that requires a special command;



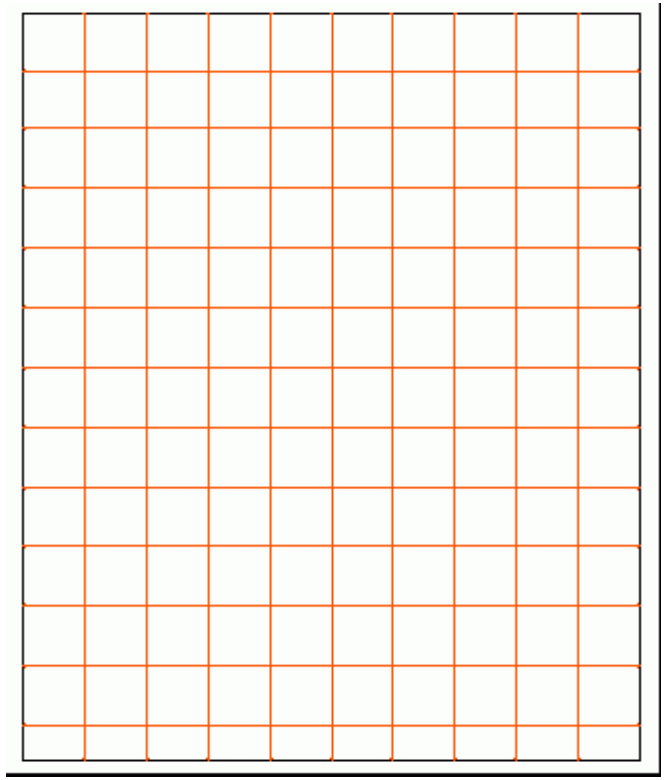
Local Remeshing

- As user-specified skewness and size limits are exceeded, local nodes and cells are added or deleted;
- As cells are added or deleted, connectivity changes;
- Available only for tri and tet mesh elements;
- The animation also shows smoothing (which one typically uses together with remeshing).

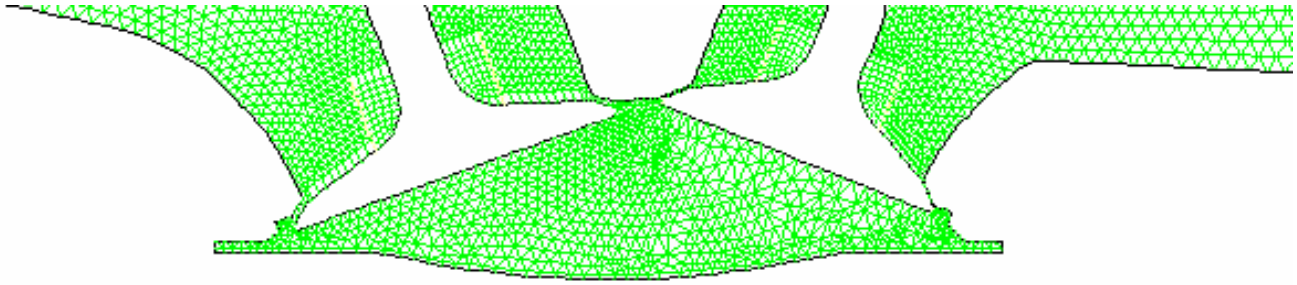


Layering

- Cells are added or deleted as the zone grows and shrinks;
- As cells are added or deleted, connectivity changes;
- Available for quad, hex and wedge mesh elements.



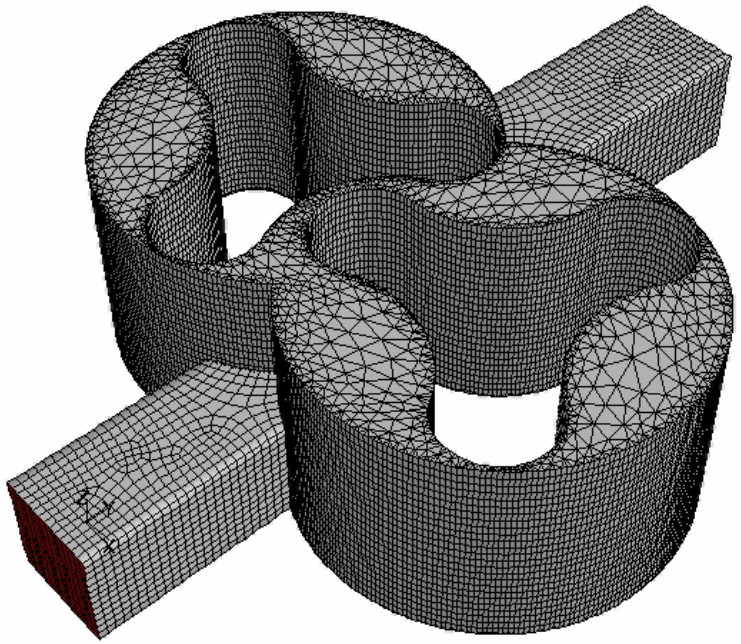
Combination of Approaches



- Initial mesh needs proper decomposition;
- Layering:
  - Valve travel region;
  - Lower cylinder region.
- Remeshing:
  - Upper cylinder region.
- Non-conformal interface between zones.

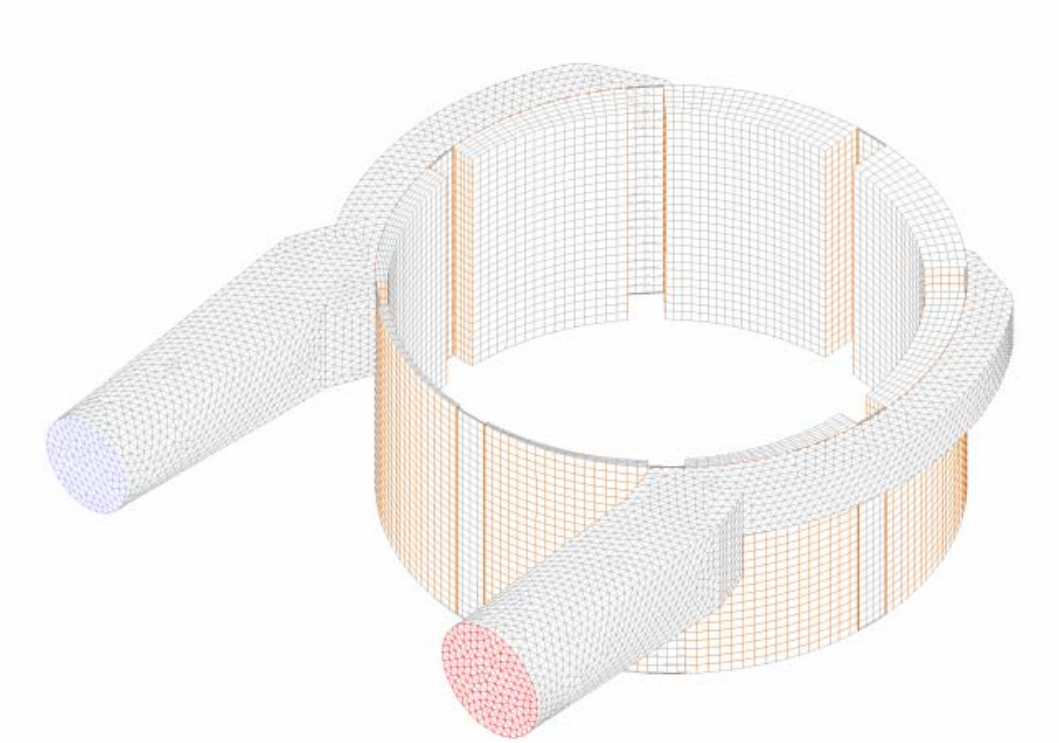
2.5 D

- The 2.5D mesh essentially is a 2D triangular mesh which is extruded along the normal axis of the specific dynamic zone that you are interested in modeling.
  - Rigid body motion is applied to the moving boundary zones
  - Triangular extrusion surface is assigned to a deforming zone with remeshing and smoothing enabled.
  - The opposite side of the triangular mesh is extruded and assigned to be a deforming zone as well, with only smoothing enabled.



User Defined Mesh Motion

- Mesh is defined by the user through a udf
- No connectivity change is allowed if using user defined function to move the mesh
- Useful applications include:
  - Vane pumps
  - Gerotor pumps
  - Bearing
  - Rotary compressors

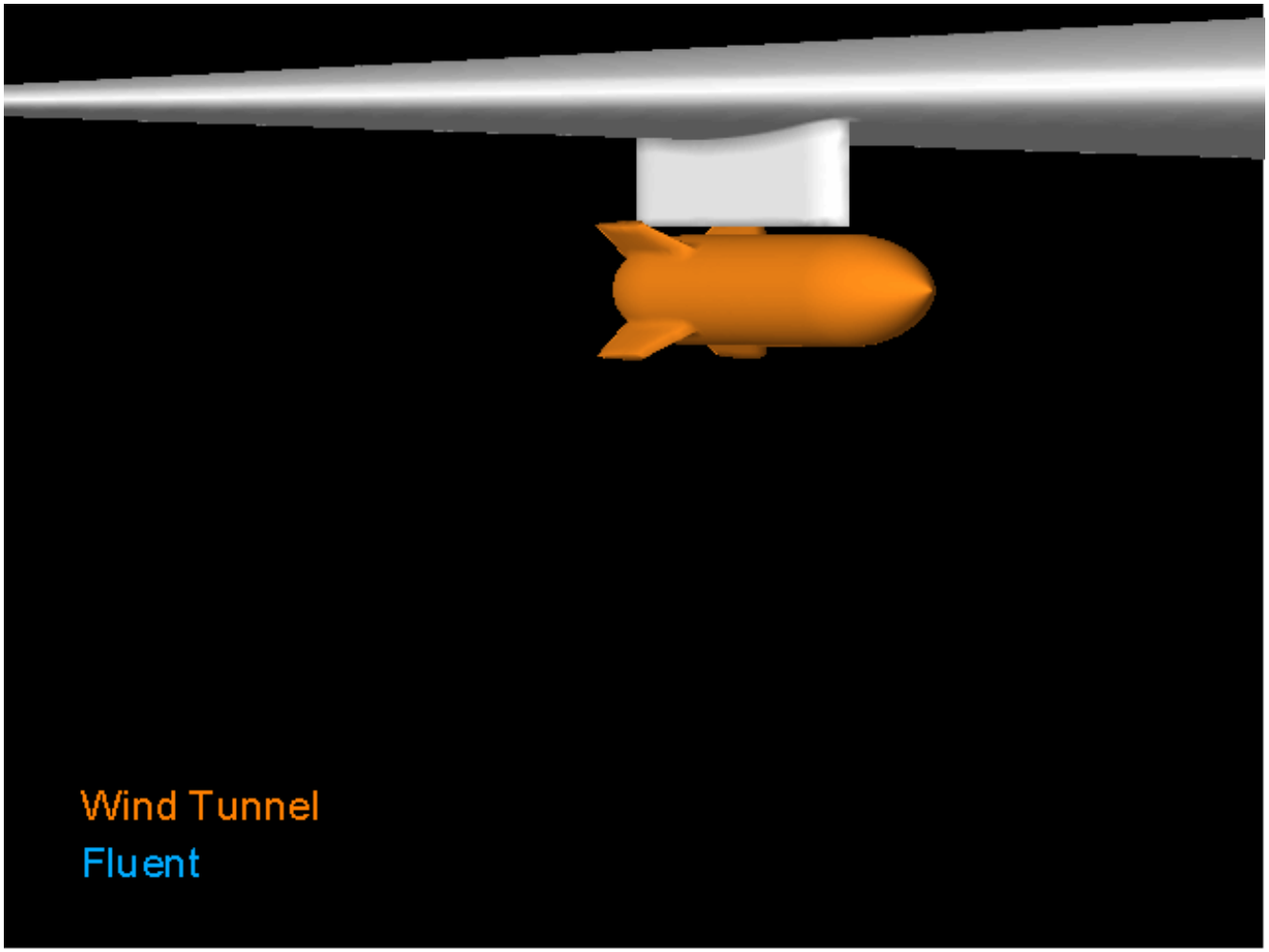


6 DOF Coupled Motion

- Objects move as a result of aerodynamic forces and moments acting together with other forces, such as the gravity force, thrust forces, or ejector forces (i.e., forces used to initially push objects away from an airplane or rocket, to avoid collisions)
- In such cases, the motion and the flow field are thus coupled, and we call this coupled motion
- Fluent provides a UDF (user-defined function) that computes the trajectory of an object based on the aerodynamic forces/moments, gravitational force, and ejector forces. This is often called a 6-DOF (degree-of-freedom) solver , and we refer to it as the 6-DOF UDF;
- The 6-DOF UDF is fully parallelized.

6 DOF Coupled Motion (cont'd)

- Store dropped from a delta wing (NACA 64A010) at Mach 1.2;
- Ejector forces dominate for a short time;
- All-tet mesh;
- Smoothing; remeshing with size function;
- Fluent results agree very well with wind tunnel results!



Outline

- Introduction and Overview of Modeling Approaches
- Single-Reference Frame (SRF) Model
- Multiple Zones and Multiple-Reference Frame (MRF) Model
- Mixing Plane Model (MPM)
- Sliding Mesh Model (SMM)
- Dynamic (Moving and Deforming) Mesh Model
- **Summary**
- **Appendix**

Summary

- Five different approaches may be used to model flows over moving parts.
  - Single (Rotating) Reference Frame Model
  - Multiple Reference Frame Model
  - Mixing Plane Model
  - Sliding Mesh Model
  - Dynamic Mesh Model
- Enabling these models, involves in part, changing the stationary fluid zones to either Moving Reference Frame or Moving Mesh.
  - Moving Mesh models must use unsteady solver
- Lecture focused on rotating components, though, translational motion can also be addressed.
  - Restrictions on geometry of interface still applies.
  - SMM can be used, for example, to study flow between two trains passing each other.

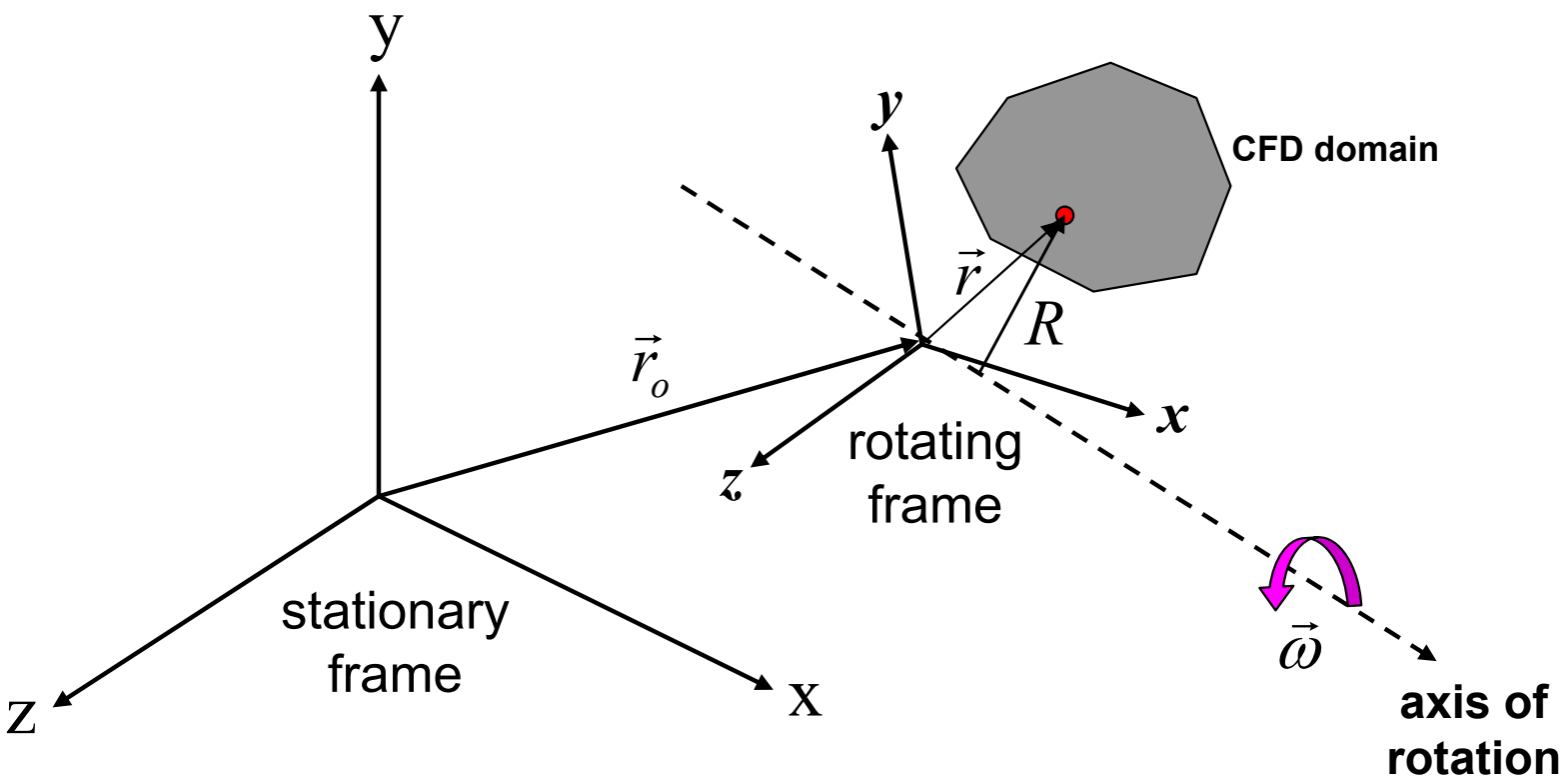
Appendix

- Navier-Stokes equations for moving reference frames
  - Relative Velocity Formulation
  - Absolution Velocity Formulation
- Navier-Stokes equations for sliding mesh problems

N-S Equations: Rotating Reference Frame

- Two different formulations are used in Fluent
  - Relative Velocity Formulation (RVF)
    - Obtained by transforming the stationary frame N-S equations to a rotating reference frame
    - Uses the **relative velocity** as the dependent variable in the momentum equations
    - Uses the **relative total internal energy** as the dependent variable in the energy equation
  - Absolute Velocity Formulation (AVF)
    - Derived from the relative velocity formulation
    - Uses the **absolute velocity** as the dependent variable in the momentum equations
    - Uses the **absolute total internal energy** as the dependent variable in the energy equation
  - NOTE: RVF and AVF are **equivalent forms** of the N-S equations!
    - Identical solutions should be obtained from either formulation with equivalent boundary conditions

Reference Frames



Note: **R** is perpendicular to axis of rotation

Assumptions and Definitions

- Assumptions
  - No translation ($d\vec{r}_o / dt = 0$)
  - Steady rotation (ω = constant) about specified axis
    - axis passes through origin of rotating frame
  - Ignore body forces due to gravity and other effects (for the equations shown)
  - Ignore energy sources (for the equations shown)
- Definitions
  - Absolute velocity ($\vec{V}$) - Fluid velocity with respect to the stationary (absolute) reference frame
  - Relative velocity ($\vec{W}$) - Fluid velocity with respect to the rotating reference frame
- 3-D compressible, laminar forms of the equations presented in the following slides (other forms are similar)

Relative Velocity Formulation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \vec{W} = 0 \qquad \text{(continuity)}$$

$$\frac{\partial \rho w_x}{\partial t} + \nabla \cdot \rho \vec{W} w_x = -\frac{\partial p}{\partial x} + \nabla \cdot \vec{\tau}_{vrx} + B_x \qquad \text{(x momentum)}$$

$$\frac{\partial \rho w_y}{\partial t} + \nabla \cdot \rho \vec{W} w_y = -\frac{\partial p}{\partial y} + \nabla \cdot \vec{\tau}_{vry} + B_y \qquad \text{(y momentum)}$$

$$\frac{\partial \rho w_z}{\partial t} + \nabla \cdot \rho \vec{W} w_z = -\frac{\partial p}{\partial z} + \nabla \cdot \vec{\tau}_{vrz} + B_z \qquad \text{(z momentum)}$$

$$\frac{\partial \rho e_{tr}}{\partial t} + \nabla \cdot \rho \vec{W} \left(e_{tr} + \frac{p}{\rho} \right) = \nabla \cdot \left(\vec{\tau}_{vrx} w_x + \vec{\tau}_{vry} w_y + \vec{\tau}_{vrz} w_z - \vec{q} \right) \qquad \text{(energy)}$$

Relative Velocity Formulation (2)

$$\vec{W} = w_x \hat{i} + w_y \hat{j} + w_z \hat{k}$$ (relative velocity vector)

$$e_{tr} = e + \frac{1}{2}\left(W^2 - \omega^2 R^2\right)$$ (relative total internal energy)

$$\vec{q} = -\kappa \nabla T$$ (Fourier's Law)

$$\left.\begin{aligned}
\vec{\tau}_{vrx} &= \mu\left[\frac{\partial \vec{W}}{\partial x} + \nabla w_x - \frac{2}{3}\left(\nabla \cdot \vec{W}\right)\hat{i}\right] \\
\vec{\tau}_{vry} &= \mu\left[\frac{\partial \vec{W}}{\partial y} + \nabla w_y - \frac{2}{3}\left(\nabla \cdot \vec{W}\right)\hat{j}\right] \\
\vec{\tau}_{vrz} &= \mu\left[\frac{\partial \vec{W}}{\partial z} + \nabla w_z - \frac{2}{3}\left(\nabla \cdot \vec{W}\right)\hat{k}\right]
\end{aligned}\right\}$$ (viscous terms)

Relative Velocity Formulation (3)

- Acceleration terms due to rotating reference frame

$$\vec{B} = B_x\hat{i} + B_y\hat{j} + B_z\hat{k}$$
$$= -\rho\left(2\vec{\omega}\times\vec{W} + \vec{\omega}\times\vec{\omega}\times\vec{r}\right)$$

Coriolis acceleration

centripetal acceleration

Absolute Velocity Formulation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \vec{W} = 0 \qquad \text{(continuity)}$$

$$\frac{\partial \rho v_x}{\partial t} + \nabla \cdot \rho \vec{W} v_x = -\frac{\partial p}{\partial x} + \nabla \cdot \vec{\tau}_{vx} + B_x \qquad \text{(x momentum)}$$

$$\frac{\partial \rho v_y}{\partial t} + \nabla \cdot \rho \vec{W} v_y = -\frac{\partial p}{\partial y} + \nabla \cdot \vec{\tau}_{vy} + B_y \qquad \text{(y momentum)}$$

$$\frac{\partial \rho v_z}{\partial t} + \nabla \cdot \rho \vec{W} v_z = -\frac{\partial p}{\partial z} + \nabla \cdot \vec{\tau}_{vz} + B_z \qquad \text{(z momentum)}$$

$$\frac{\partial \rho e_t}{\partial t} + \nabla \cdot \rho \vec{W} \left(e_t + \frac{p}{\rho} \right) = \nabla \cdot \left(\vec{\tau}_{vx} v_x + \vec{\tau}_{vy} v_y + \vec{\tau}_{vz} v_z - \vec{q} \right) \qquad \text{(energy)}$$

Absolute Velocity Formulation (2)

$$\vec{V} = v_x \hat{i} + v_y \hat{j} + v_z \hat{k}$$ (absolute velocity vector)

$$e_t = e + \frac{1}{2} V^2$$ (total internal energy)

$$\vec{q} = -\kappa \nabla T$$ (Fourier's Law)

$$\vec{\tau}_{vx} = \mu \left[\frac{\partial \vec{V}}{\partial x} + \nabla v_x - \frac{2}{3} \left(\nabla \cdot \vec{V} \right) \hat{i} \right]$$

$$\vec{\tau}_{vy} = \mu \left[\frac{\partial \vec{V}}{\partial y} + \nabla v_y - \frac{2}{3} \left(\nabla \cdot \vec{V} \right) \hat{j} \right]$$

$$\vec{\tau}_{vz} = \mu \left[\frac{\partial \vec{V}}{\partial z} + \nabla v_z - \frac{2}{3} \left(\nabla \cdot \vec{V} \right) \hat{k} \right]$$

(viscous terms)

Absolute Velocity Formulation (3)

- Acceleration term due to rotating reference frame

$$\vec{B} = B_x\hat{i} + B_y\hat{j} + B_z\hat{k}$$
$$= -\rho\vec{\omega}\times\vec{V}$$

Acceleration reduces to single term involving rotational speed and absolute velocity

Velocity Formulation Recommendations

- Use AVF when inflow comes from a stationary domain
  - Absolute total pressure or absolute velocities are usually known in this case
  - Example: Flow in a ducted fan system, where inlet is a stationary duct
- Use RVF with closed domains (all surfaces are moving) or if inflow comes from a rotating domain
  - Relative total pressure or relative velocities are usually known in this case
  - Example: Swirling flow in a disk cavity
- As noted previously, RVF and AVF are equivalent, and therefore either can be used successfully for most problems
  - Discrepancies on the same mesh can occur if the magnitude of absolute velocity gradients are very different than magnitude of the relative velocity gradients
  - Differences between solutions should disappear with suitable mesh refinement

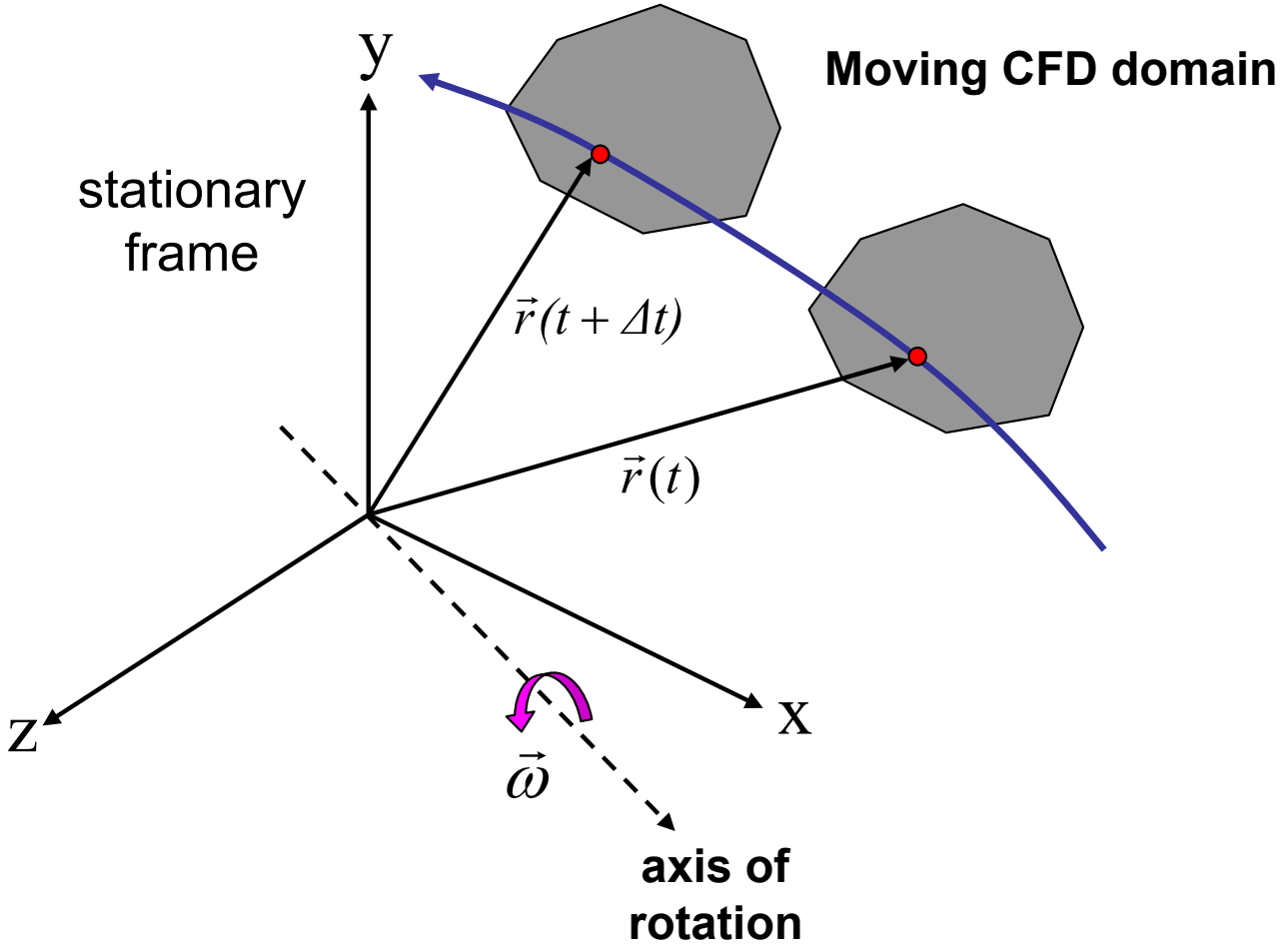
N-S Equations: Moving Mesh Form

- The sliding mesh (aka moving mesh) formulation assumes that the computational domain moves relative to the stationary frame
  - No reference frame is attached to the computational domain
- The motion of any point in the domain is given by a time rate of change of the position vector ($\dot{\vec{r}}$)
  - $\dot{\vec{r}}$ is also known as the grid speed
  - For rigid body rotation at constant speed

$$\dot{\vec{r}} = \vec{\omega} \times \vec{r} = \vec{U}$$

- Equations will be presented in integral form

Moving Mesh Illustration



N-S Equations: Moving Mesh (1)

$$\frac{d}{dt}\int_V \rho dV + \oint_S \rho(\vec{V}-\vec{U})\cdot = 0 \qquad \text{(continuity)}$$

$$\frac{d}{dt}\int_V \rho v_x dV + \oint_S \left[\rho\left(\vec{V}-\vec{U}\right)v_x + p\hat{i}\right]\cdot d\vec{S} = \oint_S \vec{\tau}_{vx}\cdot d\vec{S} \qquad \text{(x momentum)}$$

$$\frac{d}{dt}\int_V \rho v_y dV + \oint_S \left[\rho\left(\vec{V}-\vec{U}\right)v_y + p\hat{j}\right]\cdot d\vec{S} = \oint_S \vec{\tau}_{vy}\cdot d\vec{S} \qquad \text{(y momentum)}$$

$$\frac{d}{dt}\int_V \rho v_z dV + \oint_S \left[\rho\left(\vec{V}-\vec{U}\right)v_z + p\hat{k}\right] = \oint_S \vec{\tau}_{vz}\cdot d\vec{S} \qquad \text{(z momentum)}$$

$$\frac{d}{dt}\int_V \rho e_t + \oint_S \rho\left(\vec{V}-\vec{U}\right)\left(e_t + \frac{p}{\rho}\right)\cdot d\vec{S} = \oint_S \left(\vec{\tau}_{vx}v_x + \vec{\tau}_{vy}v_y + \vec{\tau}_{vz}v_z - \vec{q}\right)\cdot d\vec{S} \qquad \text{(energy)}$$

N-S Equations: Moving Mesh (2)

- In the foregoing equations, V and $\vec{S}$ are the volume and boundary surface of the control volume, respectively
  - V remains constant since the mesh is not deforming
  - $\vec{S} = \vec{S}(t)$ since the area vectors are changing orientation as control volume moves
- The time derivative (d / dt) represents differentiation with respect to time following the moving domain
- The convecting velocity is again the relative velocity
- All spatial derivatives computed relative to the stationary frame