Fluent Solver Settings

Instructor: Dr. Ahmed Nagib Elmekawy

Introduction

Lecture Theme:

Fluent requires inputs (solver settings) which tell it how to calculate the solution. By introducing the concepts of accuracy, stability and convergence, the purpose of each setting can be understood. Emphasis will be placed on convergence, which is critical for the CFD simulation.

Learning Aims:

You will learn:

How to choose the solver and the discretization schemes
How to initialize the solution
How to monitor and judge solution convergence and accuracy

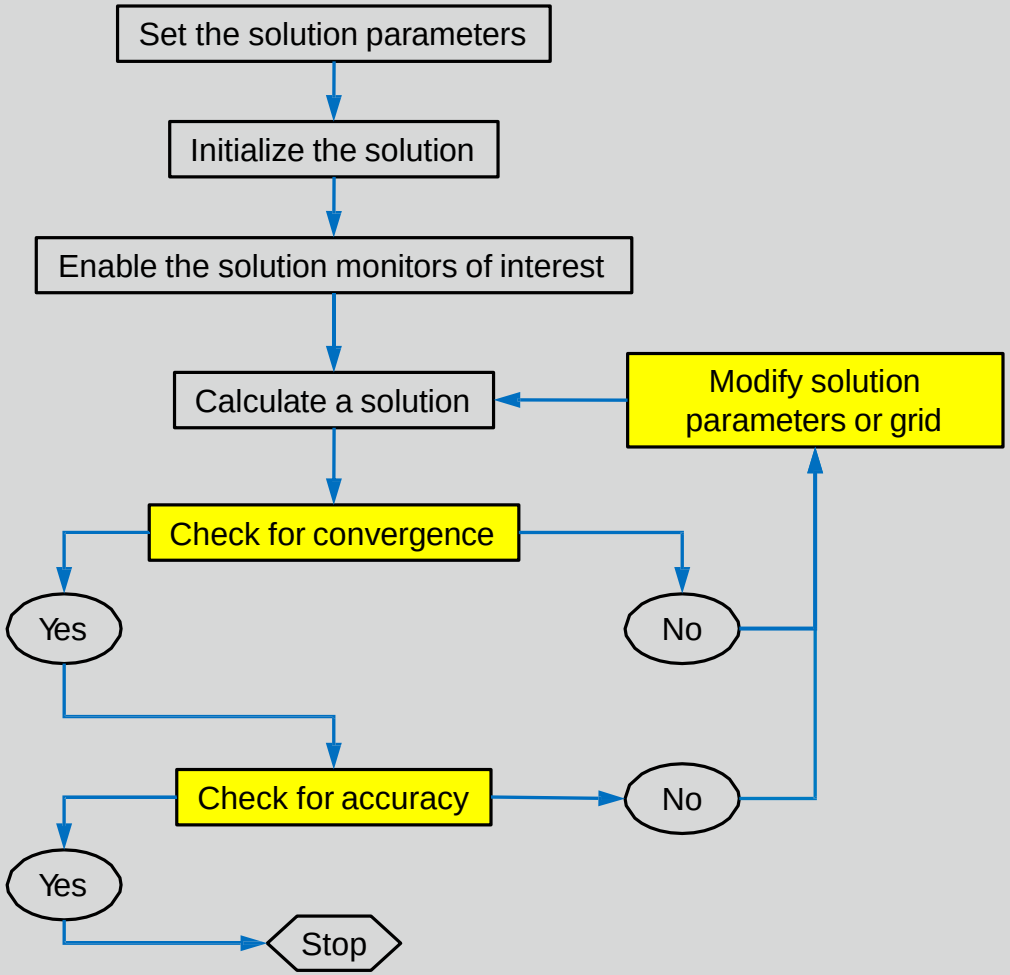
Learning Objectives:

You will be able to choose appropriate solver settings for your CFD simulation and be able to monitor and judge solution convergence

© 2012 ANSYS, Inc.

Solution Procedure Overview

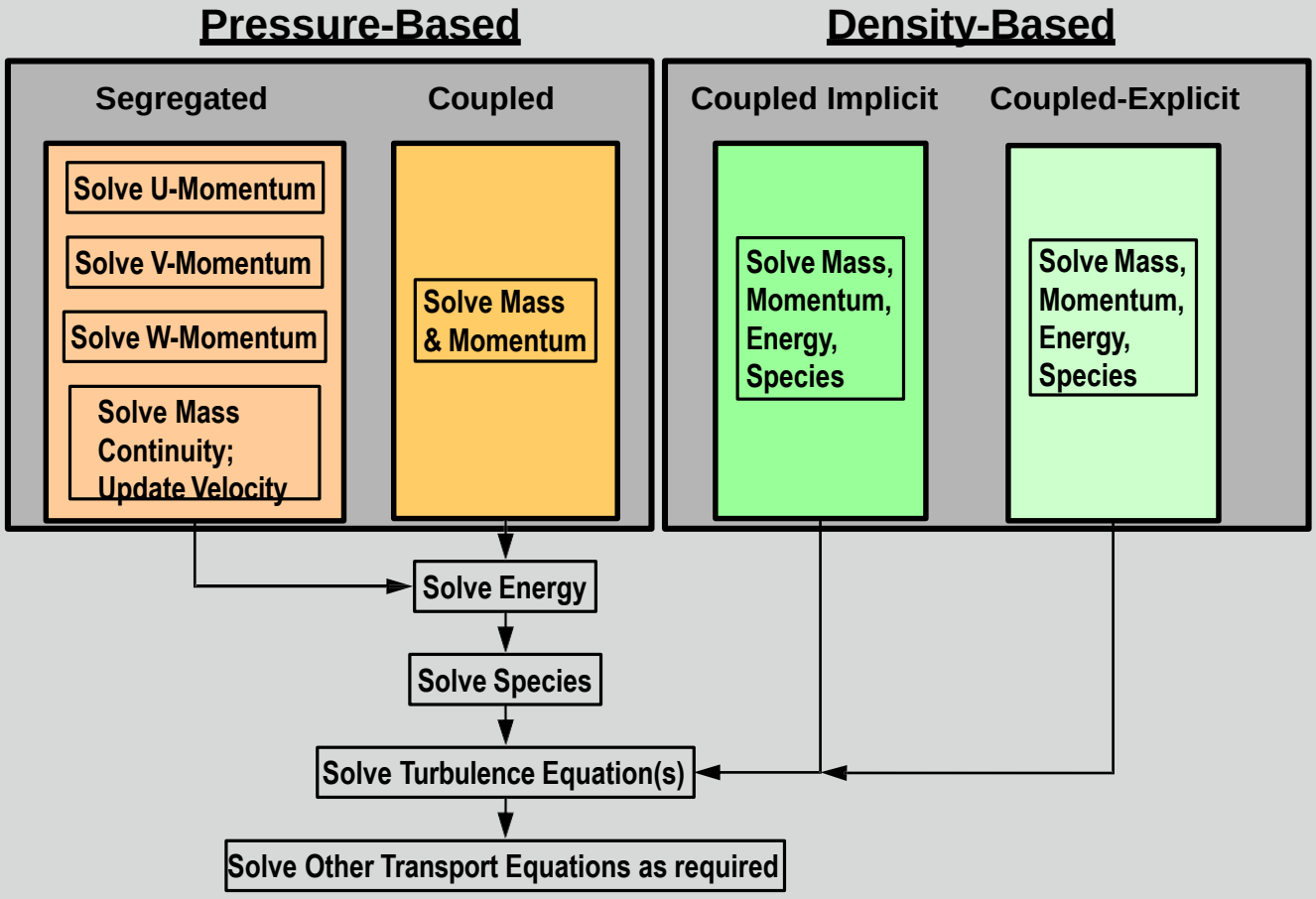
- **The sketch to the right shows the basic workflow for any simulation**
- **This lecture will look at all the items in the chart**
  - **Solution parameters**
    - Choosing the solver
    - Discretization schemes
  - **Initialization**
  - **Convergence**
    - Monitoring convergence
    - Stability
      - Setting Under-relaxation
      - Setting Courant number
      - Setting Pseudo-timestep
    - Accelerating convergence
  - **Accuracy**
    - Grid Independence
    - Adaption



© 2012 ANSYS, Inc.

Available Solvers

- **There are two kinds of solvers available in Fluent**
  - Pressure based
  - Density based



© 2012 ANSYS, Inc.

ANSYS

Fundamental Equations of Fluid Dynamics

- Continuity Equation

  which is based on the conservation of mass

- Momentum Equation

  which is based on the conservation of momentum

  (Newton's Second Law)

- Energy Equation

  which is based on the conservation of energy

  (First Law of Thermodynamics)

These equations can be derived by applying the conservation laws to a fluid passing through an infinitesimal, fixed control volume

ANSYS

Common Notations in Fluid Dynamics

density – ρ

dynamic viscosity – μ

kinematic viscosity – $\nu = \dfrac{\mu}{\rho}$

velocity – $V = (v_1, v_2, v_3)$ or (u, v, w)

pressure – p

gravitational acceleration – g

temporal coordinate – t

spatial coordinates – $X = (x_1, x_2, x_3)$ or (x, y, z)

nabla operator – $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})$

ANSYS

Continuity Equation

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

The first term represents the rate of increase of the density (per unit volume) in the control volume.

The second term represents the rate of mass flux (per unit volume) passing out of the surface surrounding the control volume.

A flow in which the density of each fluid element remains constant (i.e., $\frac{D\rho}{Dt} = 0$) is called incompressible. For simplicity, we will assume $\rho =$ Const everywhere.

The continuity equation of an incompressible flow is reduced to

$$\nabla \cdot V = 0$$

ANSYS

Momentum Equation

$$\frac{\partial}{\partial t}(\rho V) + \nabla \cdot (\rho V V) = \rho f + \nabla \cdot \Pi$$

Left-hand side:

The first term represents the rate of increase of the momentum (per unit volume) in the control volume.

The second term represents the rate of momentum lost (per unit volume) by convection through the control surface.

Right-hand side:

The first term represents the body force (such as gravity) per unit volume.

The second term represents the surface force per unit volume.

ANSYS

Momentum Equation

$$\frac{\partial}{\partial t}(\rho V) + \nabla \cdot (\rho V V) = \rho f + \nabla \cdot \Pi$$

where Π denotes the stress tensor of the fluid.

We will only consider a Newtonian fluid, i.e., the stress at any point is linearly dependent on the rate of deformation of the fluid. Then

$$\Pi_{ij} = -p\delta_{ij} + \mu \left[\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) - \frac{2}{3}\delta_{ij} \frac{\partial v_k}{\partial x_k} \right] \qquad (i, j, k = 1, 2, 3)$$

where δ_{ij} is the Kronecker delta function and where the Einstein summation convention is used.

For incompressible flow with constant viscosity, the equation is reduced to

$$\rho \frac{DV}{Dt} = \rho \frac{\partial V}{\partial t} + \rho V \cdot \nabla V = \rho f - \nabla p + \mu \nabla^2 V$$

ANSYS

Energy Equation

For an incompressible flow without external heat addition, the energy equation can be simplified as

$$\rho c_v \frac{DT}{Dt} = \kappa \nabla^2 T + \phi$$

where T is the temperature, c_v is the specific heat at a constant volume, κ is thermal conductivity coefficient, and ϕ is the dissipation function which represents the rate of mechanical energy lost due to viscosity.

Note that this equation can be decoupled from the continuity and momentum equations.

We will assume that the temperature changes are negligible so that it is not necessary to solve the energy equation.

ANSYS

Summary – Incompressible Navier-Stokes Equations

We assume ρ and μ are constant and there is no external force.

Momentum equation:

$$\rho\frac{\partial V}{\partial t} + \rho V \cdot \nabla V = -\nabla p + \mu\nabla^2 V$$

Continuity equation (also called the incompressibility condition)

$$\nabla \cdot V = 0$$

For two-dimensional (2D) problems, we have

$$\begin{aligned} \rho u_t + \rho u u_x + \rho w u_z &= -p_x + \mu(u_{xx} + u_{zz}) \\ \rho w_t + \rho u w_x + \rho w w_z &= -p_z + \mu(w_{xx} + w_{zz}) \\ u_x + w_z &= 0 \end{aligned}$$

where u and w are the horizontal and vertical velocity components, respectively.

ANSYS

Pressure Equation Approach

By taking the divergence of the momentum equations and using the incompressibility condition, we can obtain a Poisson equation for the pressure

$$\nabla^2 p = -\rho \nabla \cdot (V \cdot \nabla V)$$

In 2D case that gives

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial z^2} = 2\rho \left(u_x w_z - u_z w_x\right)$$

At each time step, we solve the pressure equation first, then perform time-marching for the momentum equations to find the velocity.

Disadvantages: high resolution is required to satisfy the incompressibility constraint; special treatment is needed for the pressure boundary conditions (Gresho 1991).

ANSYS

The SIMPLE Procedure

The SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) procedure is an improvement of the pressure equation approach by involving guess-and-correct operations:

$$p = p_0 + p', \qquad u = u_0 + u', \qquad w = w_0 + w'$$

The corrections to the pressure and velocity are related by

$$u' = -A\frac{\partial p'}{\partial x}, \qquad w' = -A\frac{\partial p'}{\partial z}, \qquad \text{with} \quad A = \frac{\Delta t}{\rho}$$

Step 1. Guess the pressure (p_0) at each grid point.

Step 2. Solve the momentum equations to find (u_0, w_0).

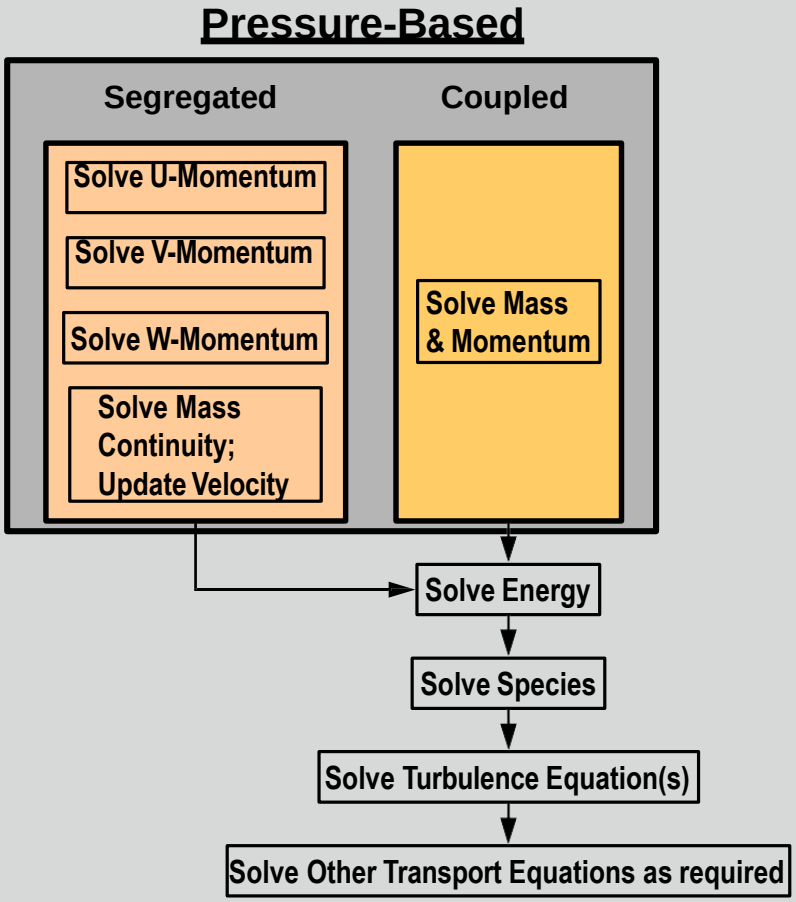
Step 3. Solve the equation $\nabla^2 p' = \frac{1}{A}(\nabla \cdot V_0)$ for p'.

Step 4. Correct the pressure and velocity by $p = p_0 + p'$ and

$$u = u_0 - \frac{A}{2\Delta x}(p'_{i+1,j} - p'_{i-1,j}), \quad w = w_0 - \frac{A}{2\Delta z}(p'_{i,j+1} - p'_{i,j-1})$$

Pressure-based Solver (PBS)

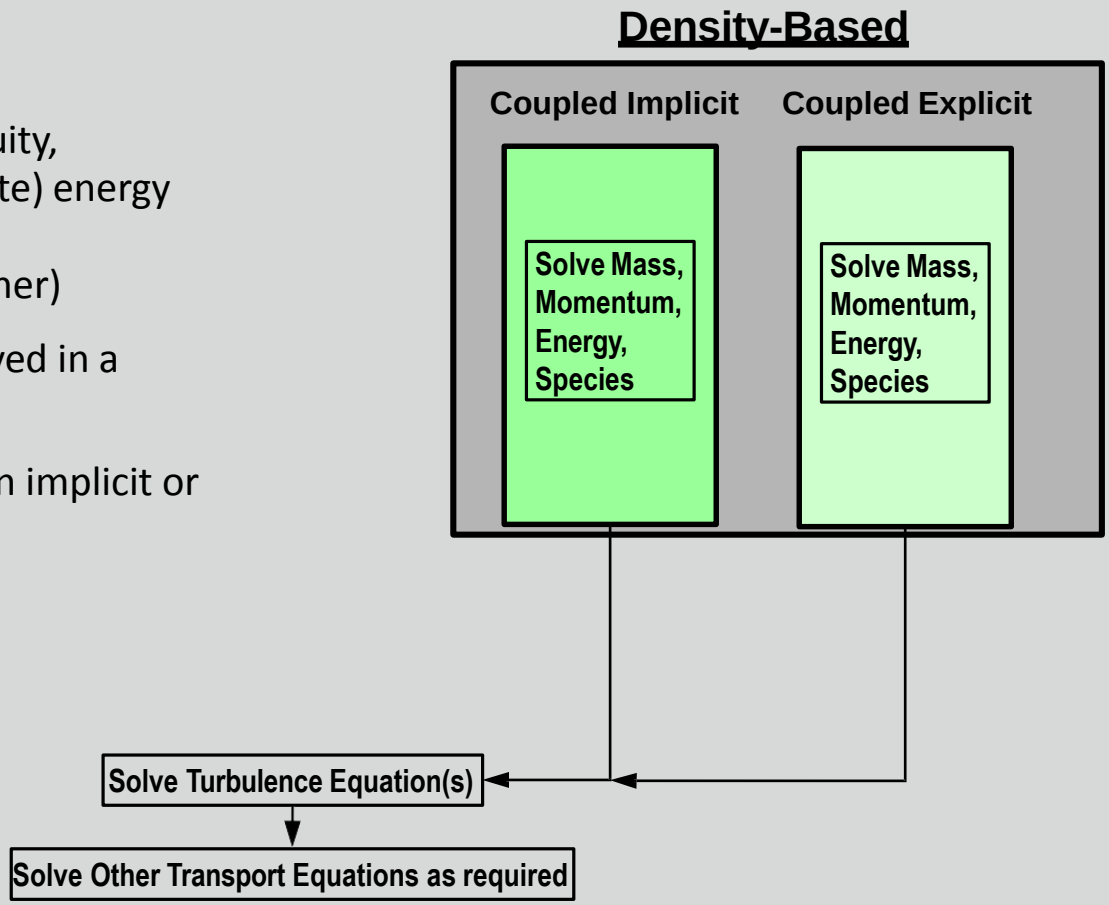
- **The pressure-based solvers**
  - Velocity field is obtained from the momentum equation
  - Mass conservation (continuity) is achieved by solving a pressure correction equation
    - Pressure-velocity coupling algorithms are derived by reformatting the continuity equation
    - The pressure equation is derived in such a way that the velocity field, corrected by the pressure, satisfies continuity
  - Energy equation (where appropriate) is solved sequentially
  - Additional scalar equations are also solved in a segregated (sequential) fashion



© 2012 ANSYS, Inc.

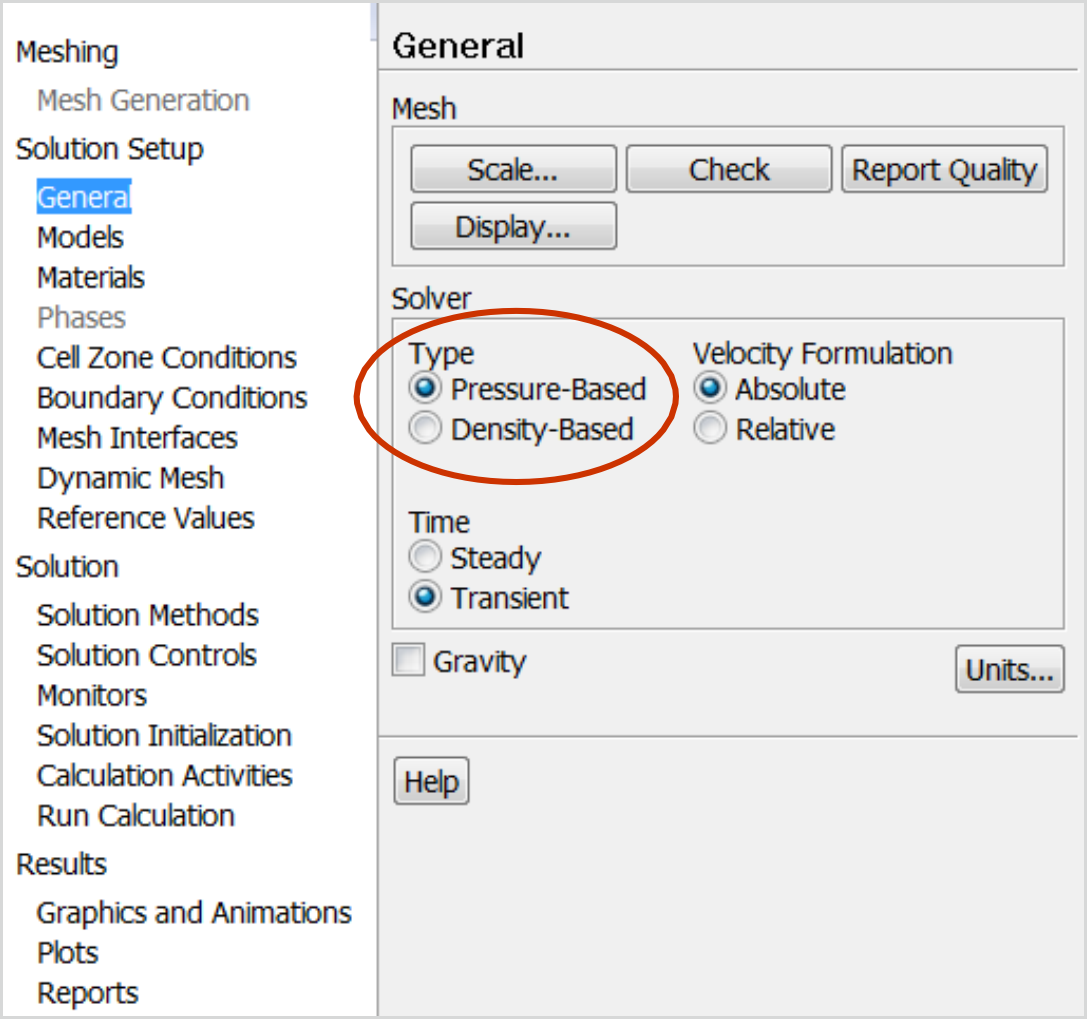
Density-based Solver (DBS)

- **Density-based Solver (DBS)**
  - The governing equations of continuity, momentum, and (where appropriate) energy and species transport are solved simultaneously (i.e., coupled together)
  - Additional scalar equations are solved in a segregated fashion
  - The density-based solver can be run implicit or explicit



© 2012 ANSYS, Inc.

Choosing a Solver

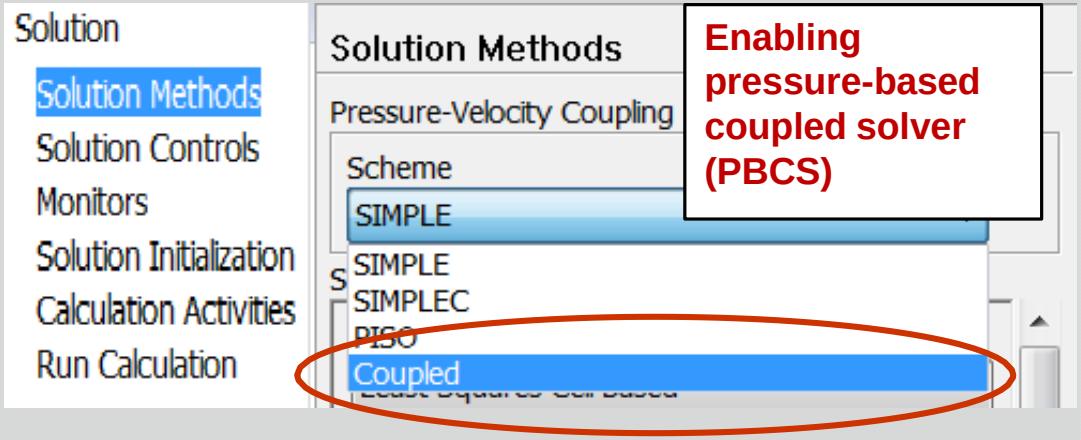


© 2012 ANSYS, Inc.

Choosing a Solver – Pressure Based

- **The pressure-based solver (segregated) is applicable for a wide range of flow regimes from low speed incompressible flow to high-speed compressible flow**
  - Requires less memory (storage) compared to coupled solvers
  - Allows flexibility in the solution procedure – damping of all equations separately
  - Examples: Good for the majority of day-to-day applications; for convergence issues switch to PBCS or DBCS

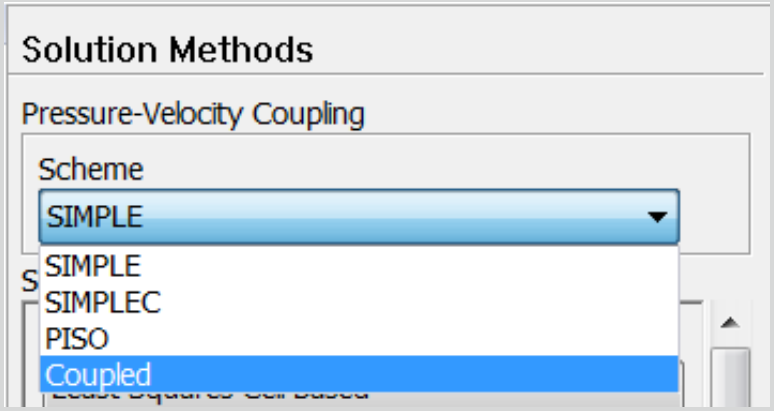
- **The pressure-based coupled solver is applicable for most flows, and yields superior performance to the standard (segregated) pressure-based solver**
  - Not available with NITA, periodic mass-flow, fixed-velocity option
  - Requires 1.5–2 times more memory than the segregated solver.
  - Examples: More demanding applications where pressure-velocity coupling rules convergence, e.g. high inertia or body forces



© 2012 ANSYS, Inc.

PBS - 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 correction when using the PBS**



- **Five algorithms are available in Fluent**
  - **Semi-Implicit Method for Pressure-Linked Equations (SIMPLE)**
    - The default scheme, robust (memory efficient)
  - **Coupled**
    - Enable the Pressure-based coupled Solver. (faster convergence than segregated)
  - **SIMPLE-Consistent (SIMPLEC)**
    - Allows faster convergence than SIMPLE for simple problems (allow high under-relaxation factors) (e.g., laminar flows with no physical models employed)
  - **Pressure-Implicit with Splitting of Operators (PISO)**
    - **Useful for unsteady flow** problems or for meshes containing cells with higher than average skewness
  - **Fractional Step Method (FSM) for unsteady flows only**
    - Used with the NITA scheme; similar characteristics as PISO (used in LES for example)

© 2012 ANSYS, Inc.

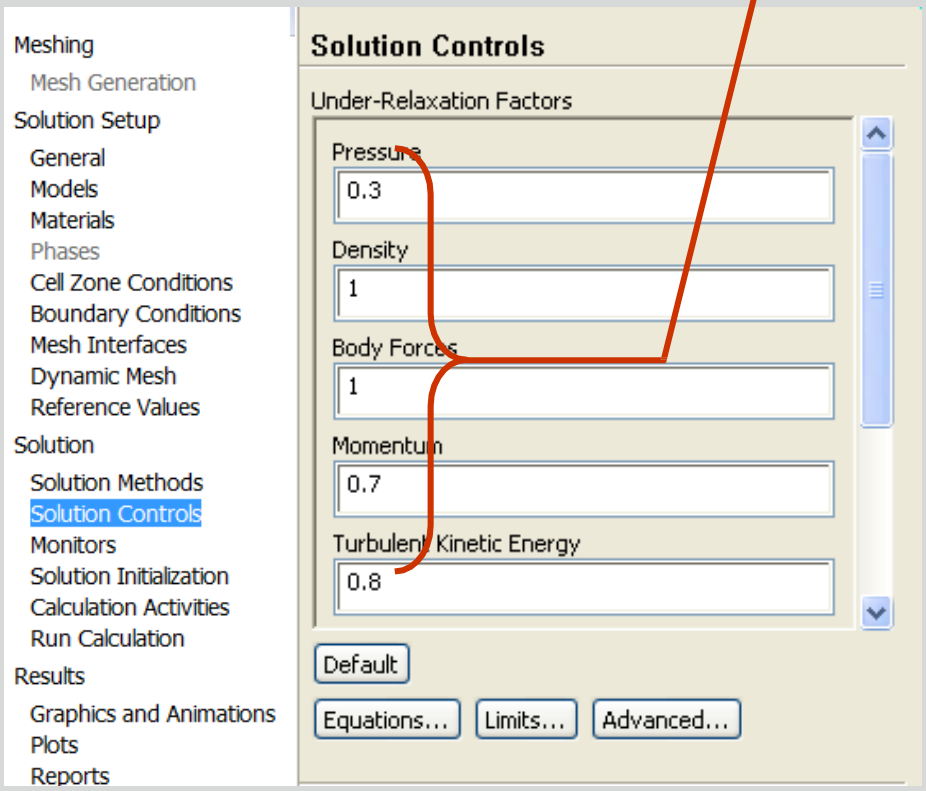
PBS Segregated Procedure - Under-Relaxation Factors

- **Implicit under-relaxation factors are used for SIMPLE, SIMPLEC, PISO**
  - **The under-relaxation factor, α, is included to stabilize the iterative process for the pressure-based solver**
  - **The final, converged solution is independent of the under-relaxation factor**
    - **Only the number of iterations required for convergence is dependent**

$$\phi_P = \phi_{P,\text{old}} + \alpha \Delta \phi_P$$

- **Default settings are suitable for a wide range of problems**
  - You can reduce the values when necessary
  - Appropriate settings are best learned from experience!

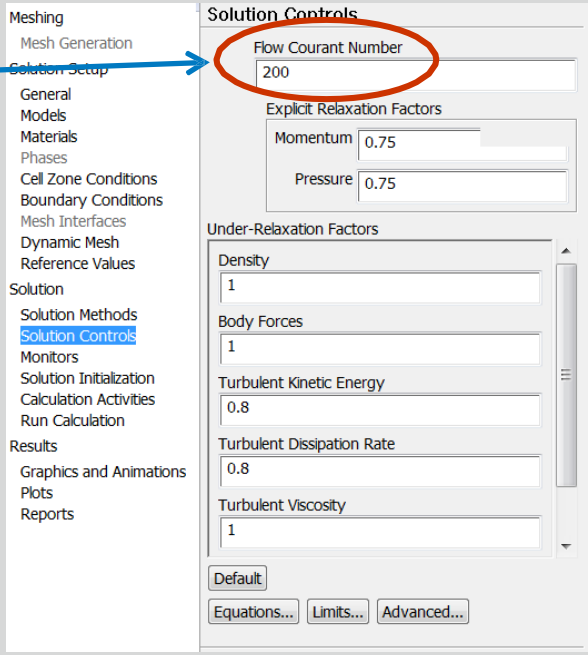
- Note : For the density-based solver, under-relaxation factors for equations outside the coupled set are modified as in the pressure-based solver

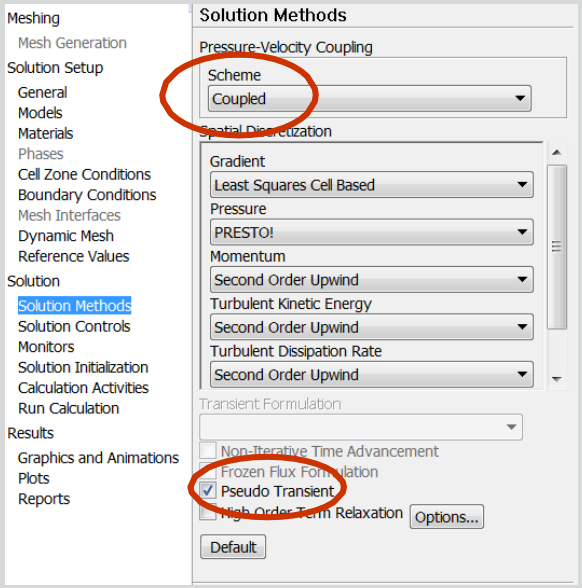


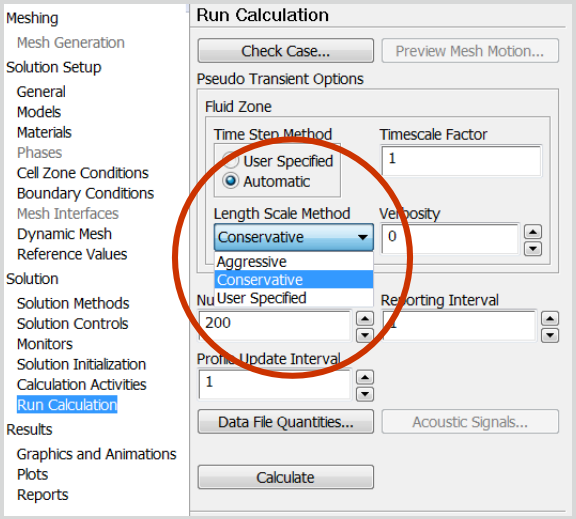
© 2012 ANSYS, Inc.

Pressure Based Coupled Solver

- 2 main options to control convergence:
  - Piloted by Courant number: default =200
    - can be reduced for more complex physics 10-50 (multiphase, combustion)
  - Pseudo-transient (similar to CFX solver)







- Pseudo time step is determined automatically from velocity and domain size.
- User-specified: Characteristic physical time is chosen

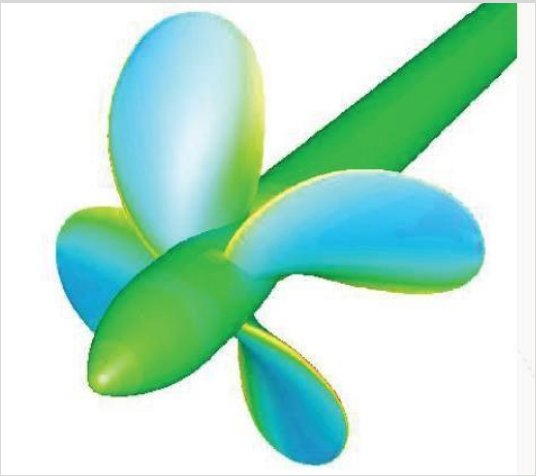
Pseudo-transient: Better convergence for meshes with large aspect ratio cells

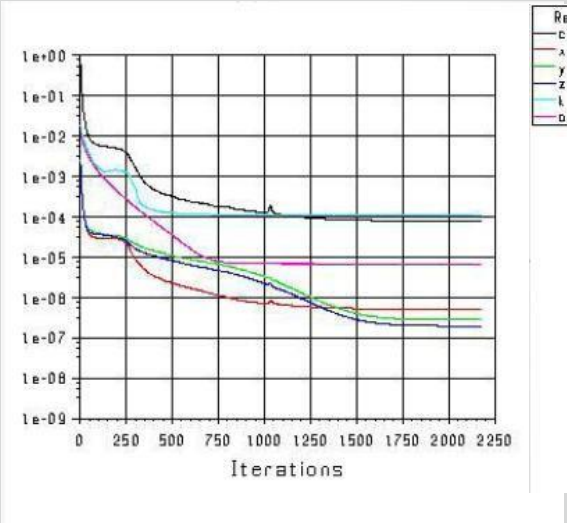
© 2012 ANSYS, Inc.

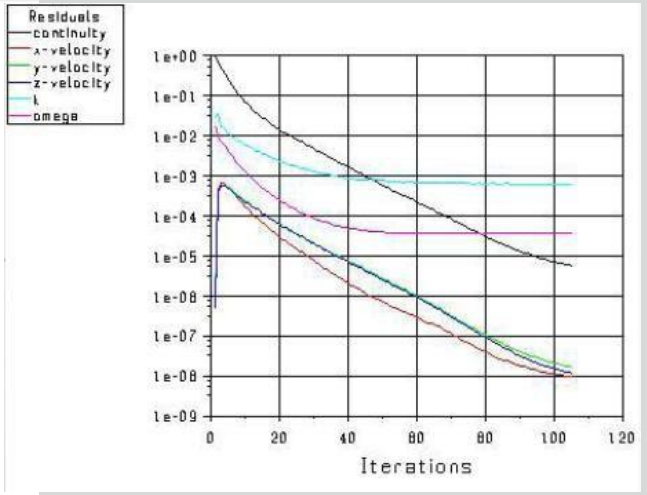
Pressure-Based Coupled Solver: Convergence

- **Pressure based coupled solver with default settings**

Rotating propeller 1500 rpm **SIMPLE: ~2250 iterations** **Coupled: ~120 iterations**





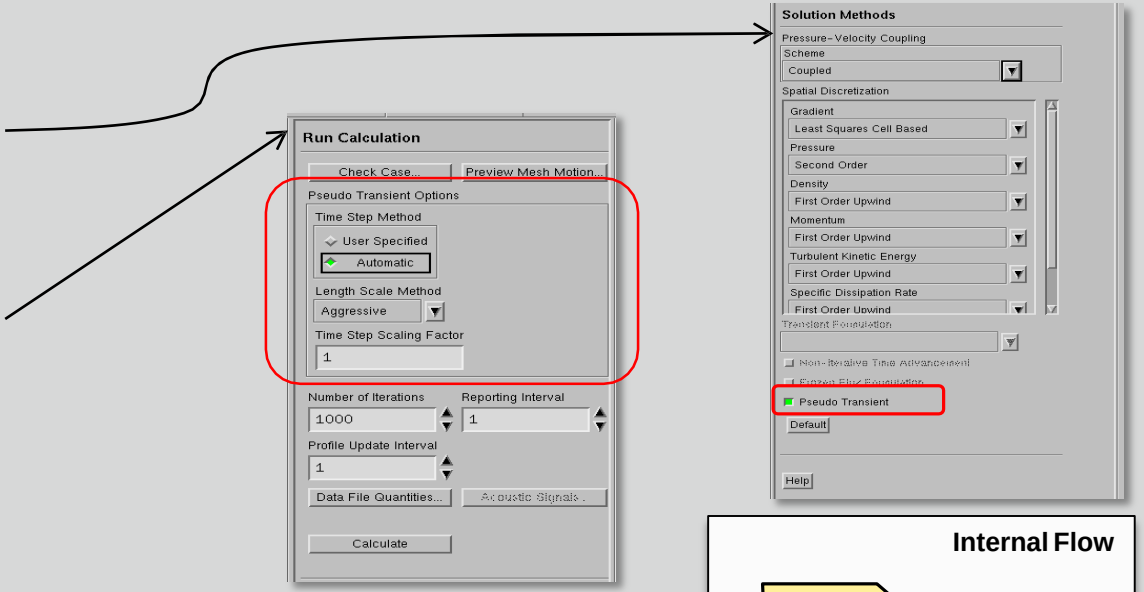


- **Approximately 2250 iterations of SIMPLE (default) in 3.5 hours**
- **Approximately 120 iterations of coupled 13 minutes**

© 2012 ANSYS, Inc.

Using the Pseudo-transient Solution Method

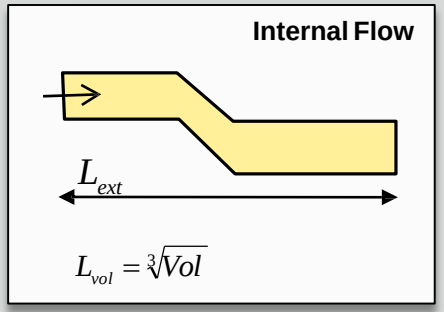
- **Solution Method panel**
  - Select Pseudo Transient
- **Run Calculation panel**
  - Select Time step method
    - Automatic (default)
    - User Specified

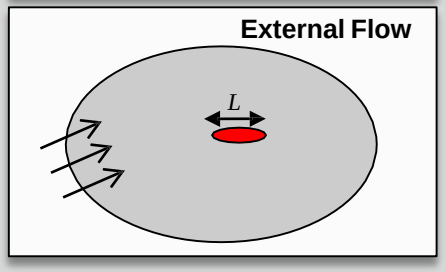


  - For Automatic
    - Select Length Scale Method (time=length/velocity)
      - Aggressive : $Max(L_{ext}, L_{Vol})$
      - Conservative : $Min(L_{ext}, L_{Vol})$
      - User Specified
      - Conservative setting is the default
    - Specify "Time Step Scaling factor": additional user control to scale automatic method

Internal Flow

External Flow





© 2012 ANSYS, Inc.

Choosing a Solver – Density Based

- **The density-based solver is applicable when there is a strong coupling, or interdependence, between density, energy, momentum, and/or species**

- **Density-based Coupled Implicit**
  - The implicit option is generally preferred over explicit since explicit has a very strict limit on time scale size (CFL constraint) as implicit does not have
  - Examples: **High speed compressible flow with combustion, hypersonic flows, shock interactions**

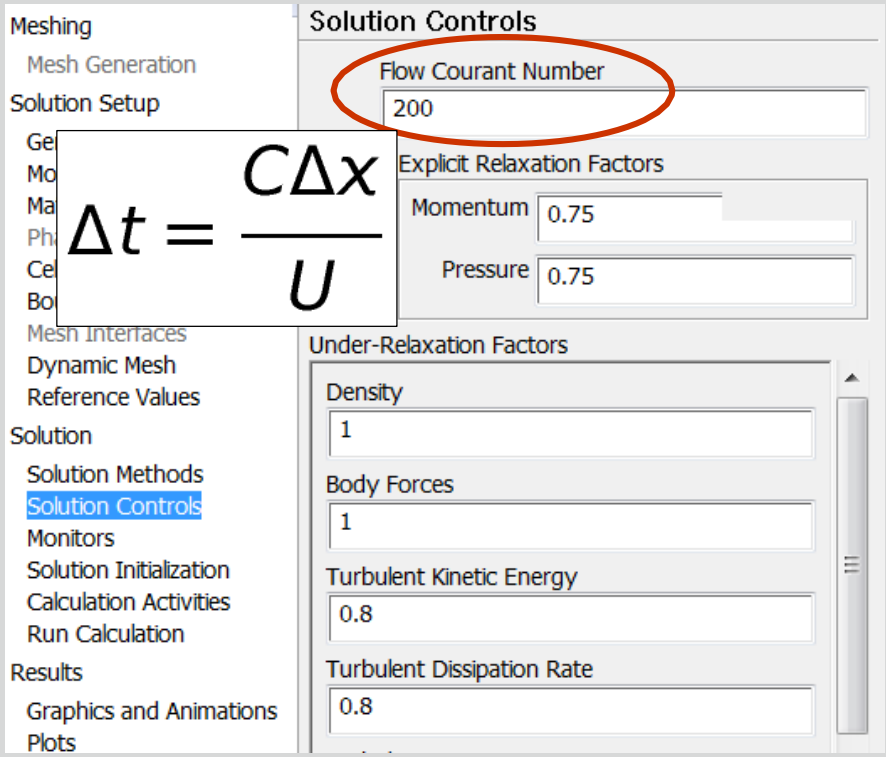
- **Density-based Coupled Explicit**
  - The explicit approach is used for cases where the characteristic time scale of the flow is on the same order as the acoustic time scale
  - Example: **propagation of high-Mach shock waves, shock tube problem**

© 2012 ANSYS, Inc.

DBS Iterative Procedure – Courant Number

- **A pseudo-transient term is included in the density-based solver even for steady state problems**
  - **The Courant number (CFL)** defines the time scale size
  - **The pseudo-transient option** is available for DBS as well as PBS.
- **For density-based explicit solver:**
  - Stability constraints impose a maximum limit on the Courant number (<2)
- **For density-based implicit solver:**
  - The Courant number is theoretically not limited by stability constraints
    - Default value is 5
      - (can be reduced for start up to 0.1-2)
      - Values of 100 – 1000 are common in external aero
  - Solution steering can be used to automatically adjust the Courant number as the solution iterates such that it has an optimal value at all stages of the calculation
    - See Workshop 04 “Fluid flow around the NACA0012 Airfoil”. Additional details also available in the Appendix

$$\Delta t = \frac{C\Delta x}{U}$$



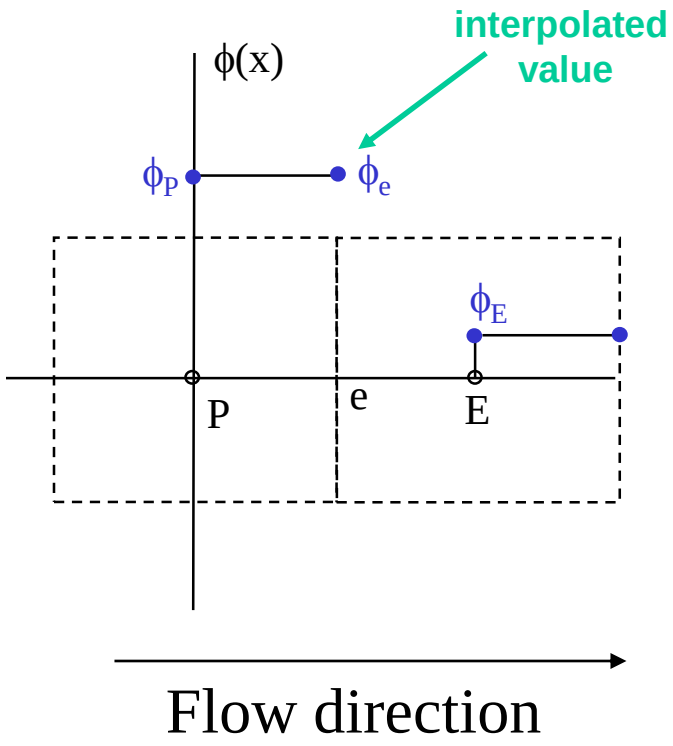
© 2012 ANSYS, Inc.

Numerical schemes: finding face values

- Face values of ϕ and $\partial\phi/\partial x$ are found by making assumptions about variation of ϕ between cell centers.
- Number of different schemes can be devised:
  - First-order upwind scheme.
  - Central differencing scheme.
  - Power-law scheme.
  - Second-order upwind scheme.
  - QUICK scheme.
- We will discuss these one by one.

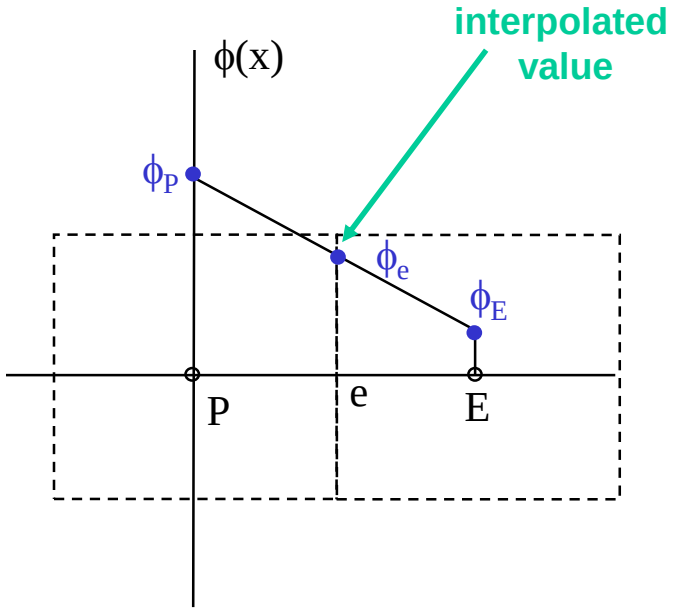
First order upwind scheme

- This is the simplest numerical scheme. It is the method that we used earlier in the discretization example.
- We assume that the value of ϕ at the face is the same as the cell centered value in the cell upstream of the face.
- The main advantages are that it is easy to implement and that it results in very stable calculations, but it also very diffusive. Gradients in the flow field tend to be smeared out, as we will show later.
- This is often the best scheme to start calculations with.



Central differencing scheme

- We determine the value of ϕ at the face by linear interpolation between the cell centered values.
- This is more accurate than the first order upwind scheme, but it leads to oscillations in the solution or divergence if the local Peclet number is larger than 2. The Peclet number is the ratio between convective and diffusive transport: $$Pe = \frac{\rho u L}{\mathbf{D}}$$
- It is common to then switch to first order upwind in cells where Pe>2. Such an approach is called a hybrid scheme.

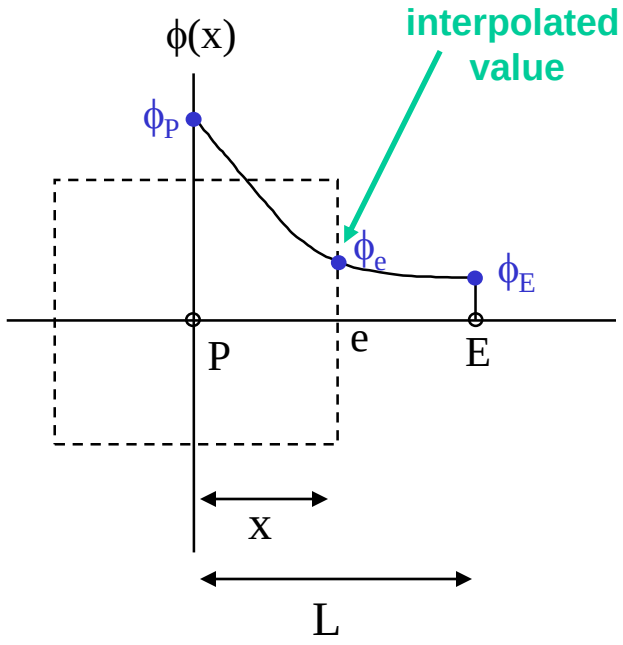


Power law scheme

- This is based on the analytical solution of the one-dimensional convection-diffusion equation.
- The face value is determined from an exponential profile through the cell values. The exponential profile is approximated by the following power law equation:

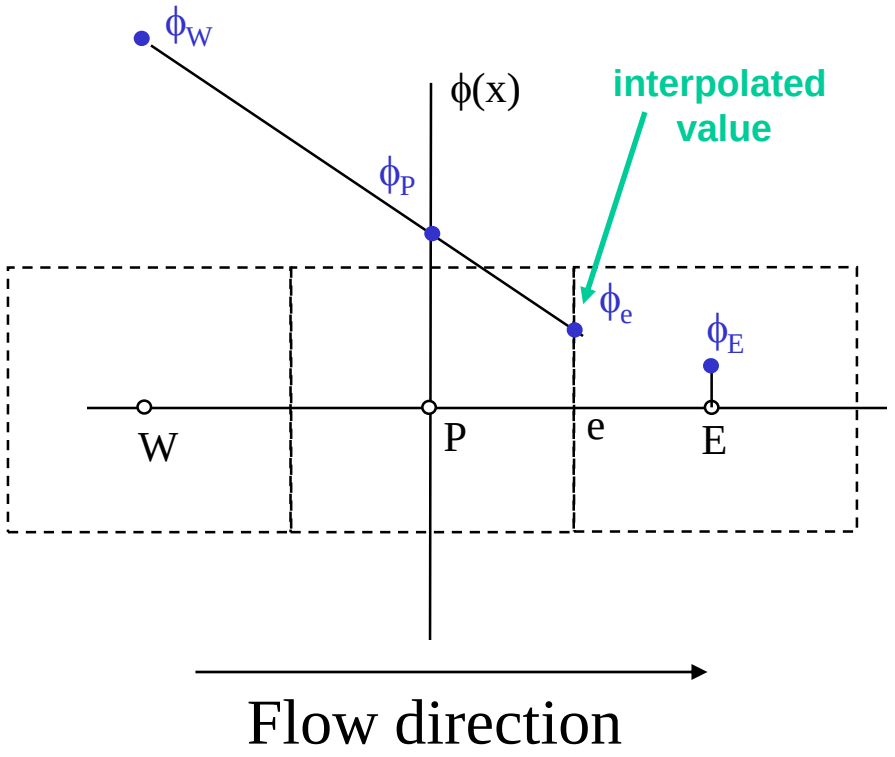
$$\phi_e = \phi_P - \frac{(1-0.1\mathrm{Pe})^5}{\mathrm{Pe}}(\phi_E - \phi_P)$$

- Pe is again the Peclet number.
- For Pe>10, diffusion is ignored and first order upwind is used.



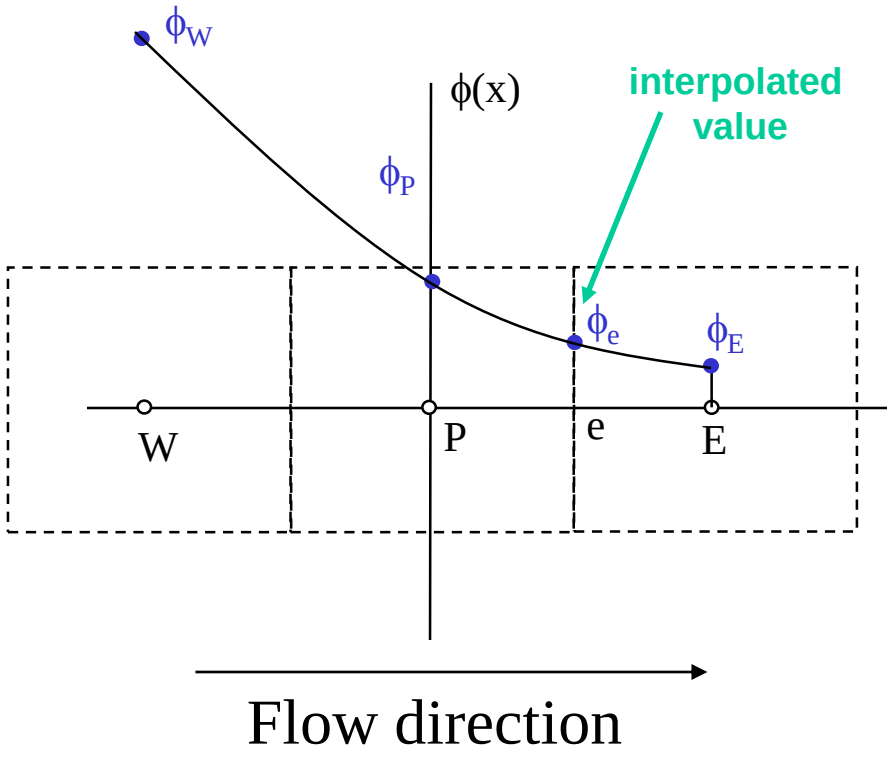
Second-order upwind scheme

- We determine the value of ϕ from the cell values in the two cells upstream of the face.
- This is more accurate than the first order upwind scheme, but in regions with strong gradients it can result in face values that are outside of the range of cell values. It is then necessary to apply limiters to the predicted face values.
- There are many different ways to implement this, but second-order upwind with limiters is one of the more popular numerical schemes because of its combination of accuracy and stability.



QUICK scheme

- QUICK stands for Quadratic Upwind Interpolation for Convective Kinetics.
- A quadratic curve is fitted through two upstream nodes and one downstream node.
- This is a very accurate scheme, but in regions with strong gradients, overshoots and undershoots can occur. This can lead to stability problems in the calculation.



Accuracy of numerical schemes

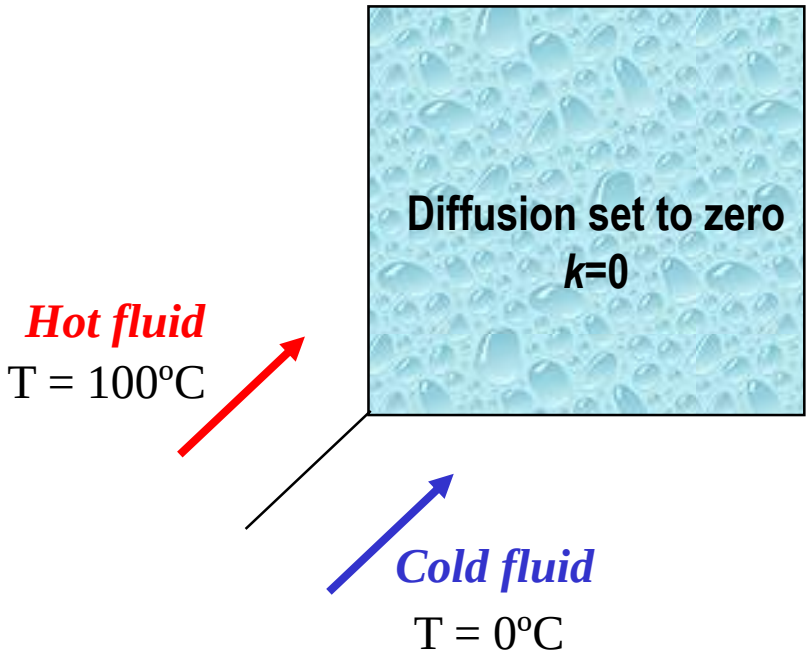
- Each of the previously discussed numerical schemes assumes some shape of the function ϕ. These functions can be approximated by Taylor series polynomials:

$$\phi(x_e) = \phi(x_P) + \frac{\phi'(x_P)}{1!}(x_e - x_P) + \frac{\phi''(x_P)}{2!}(x_e - x_P)^2 + ... + \frac{\phi^n(x_P)}{n!}(x_e - x_P)^n + ...$$

- The first order upwind scheme only uses the constant and ignores the first derivative and consecutive terms. This scheme is therefore considered first order accurate.
- For high Peclet numbers the power law scheme reduces to the first order upwind scheme, so it is also considered first order accurate.
- The central differencing scheme and second order upwind scheme do include the first order derivative, but ignore the second order derivative. These schemes are therefore considered second order accurate. QUICK does take the second order derivative into account, but ignores the third order derivative. This is then considered third order accurate.

Accuracy and false diffusion (1)

- False diffusion is numerically introduced diffusion and arises in convection dominated flows, i.e. high Pe number flows.
- Consider the problem below. If there is no false diffusion, the temperature will be exactly 100 ºC everywhere above the diagonal and exactly 0 ºC everywhere below the diagonal.
- False diffusion will occur due to the oblique flow direction and non-zero gradient of temperature in the direction normal to the flow.
- Grid refinement coupled with a higher-order interpolation scheme will minimize the false diffusion as shown on the next slide.

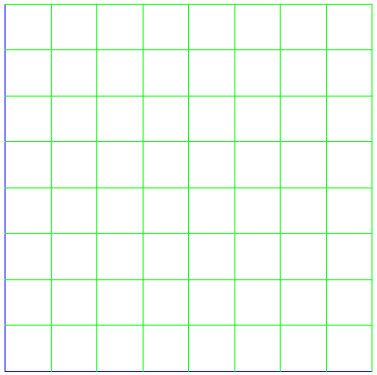


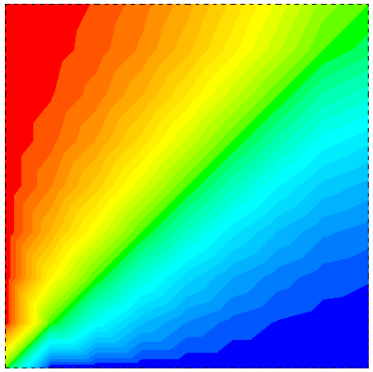
Accuracy and false diffusion (2)

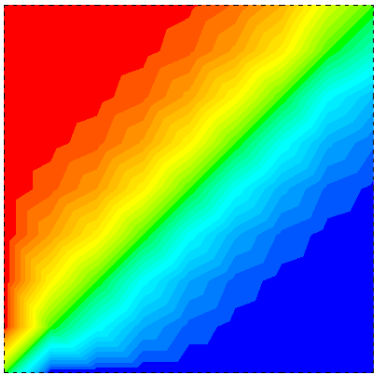
First-order Upwind

Second-order Upwind

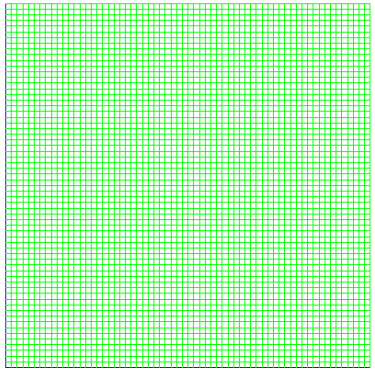
8 x 8

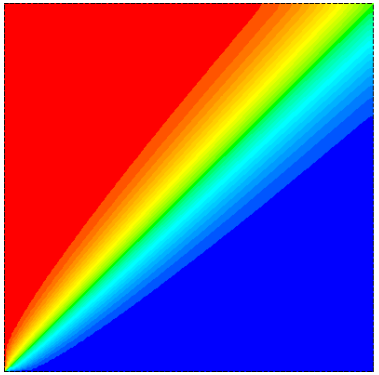


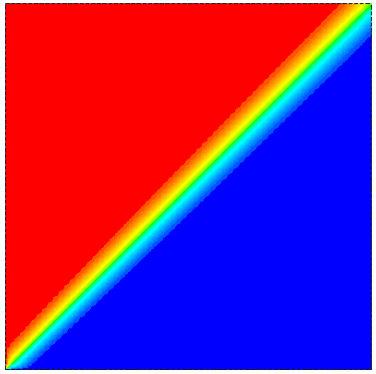




64 x 64







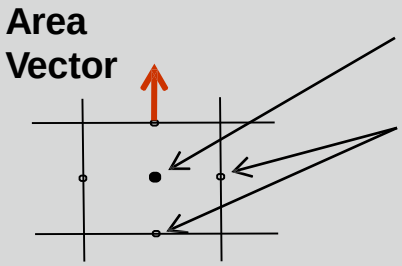
Solution accuracy

- Higher order schemes will be more accurate. They will also be less stable and will increase computational time.
- It is recommended to always start calculations with first order upwind and after 100 iterations or so to switch over to second order upwind.
- This provides a good combination of stability and accuracy.
- The central differencing scheme should only be used for transient calculations involving the large eddy simulation (LES) turbulence models in combination with grids that are fine enough that the Peclet number is always less than one.
- It is recommended to only use the power law or QUICK schemes if it is known that those are somehow especially suitable for the particular problem being studied.

PBS - Interpolation Methods for Pressure

- **Interpolation is required for calculating cell-face pressures in order to compute pressure gradient (Gauss method):**

$$\nabla P = \frac{\sum_{face} P_{face} \vec{A}_{face}}{Vol_{cell}}$$



Value (e.g. Pressure) computed here

But solver must estimate the value at each face in order to compute pressure gradient (Gauss method)

- **Standard** – The 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 instead)
- **PRESTO!** – Use for highly swirling flows, flows involving steep pressure gradients (porous media, fan model, etc.), or in strongly curved domains
- **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

© 2012 ANSYS, Inc.

Discretization (Interpolation Methods)

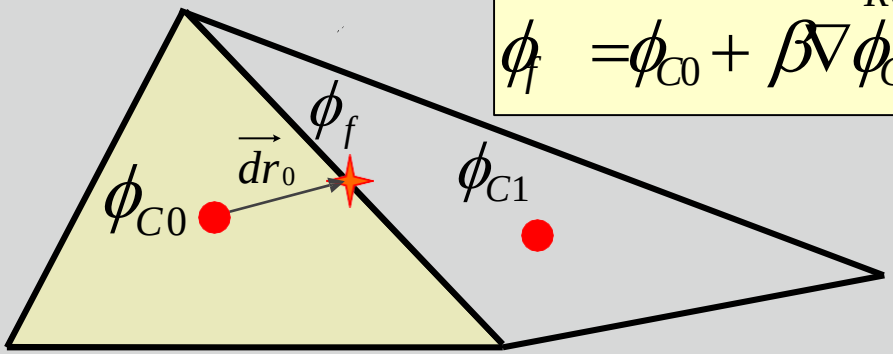
- **Field variables (stored at cell centers) must be interpolated to the faces of the control volumes**

$$V_{\text{cell}}\frac{\partial(\rho\phi)}{\partial t}+\sum_{f}^{N_{\text{faces}}}\rho_f\mathbf{V}_f\phi_f\cdot\mathbf{A}_f=\sum_{f}^{N_{\text{faces}}}\Gamma_\phi\nabla\phi_f\cdot\mathbf{A}_f+S_\phi V_{\text{cell}}$$

- **Interpolation schemes for the convection term:**
  - **First-Order Upwind** – Easiest to converge, only first-order accurate
  - **Power Law** – More accurate than first-order for flows when $Re_{cell} < 5$ (typ. low Re flows)
  - **Second-Order Upwind** – Uses larger stencils for 2nd order accuracy, essential with tri/tet mesh or when flow is not aligned with grid; convergence may be slower
  - **Monotone Upstream-Centered Schemes for Conservation Laws (MUSCL)** – Locally 3rd order convection discretization scheme for unstructured meshes; more accurate in predicting secondary flows, vortices, forces, etc.
  - **Quadratic Upwind Interpolation (QUICK)** – Applies to quad/hex and hybrid meshes, useful for rotating/swirling flows, 3rd-order accurate on uniform Quad mesh

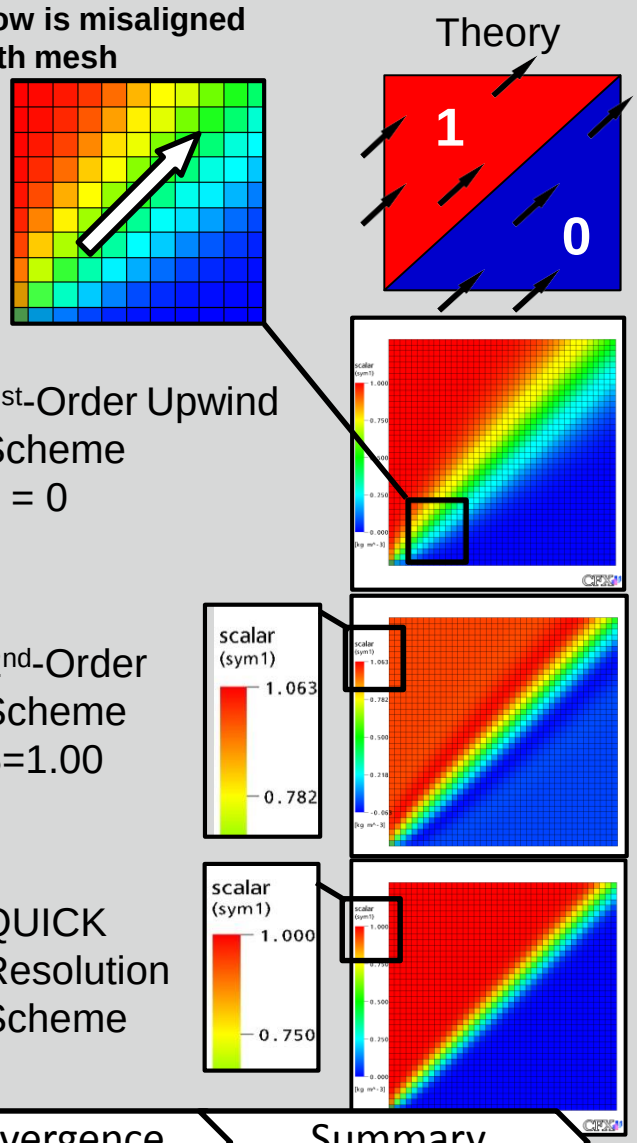
© 2012 ANSYS, Inc.

Effects of Discretization



$$\phi_f = \phi_{C0} + \beta \nabla \phi_{C0} \bullet dr_0 \longrightarrow^{RG}$$

- **If $\beta = 0$ we get the 1st-Order-*Upwind* convection scheme, i.e. no correction**
  - This is robust but only first order accurate
  - Sometimes useful for initial runs
- **If $\beta = 1$ we get the 2nd-Order-Upwind Scheme**
  - Additional Limiters must be added to guaranteed the solution to be bounded ($\phi_{C0} < \phi_f < \phi_{C1}$)
- **The *QUICK Resolution* scheme "maximizes" β throughout the flow domain while keeping the solution bounded**



© 2012 ANSYS, Inc.

Discretization (Interpolation Methods)

- **Interpolation schemes for the diffusive term:**

$$V_{\text{cell}}\frac{\partial(\rho\phi)}{\partial t}+\sum_{f}^{N_{\text{faces}}}\rho_f\mathbf{V}_f\phi_f\cdot\mathbf{A}_f=\sum_{f}^{N_{\text{faces}}}\Gamma_\phi\nabla\phi_f\cdot\mathbf{A}_f+S_\phi V_{\text{cell}}$$

  – Always central-differenced & 2nd order accuracy

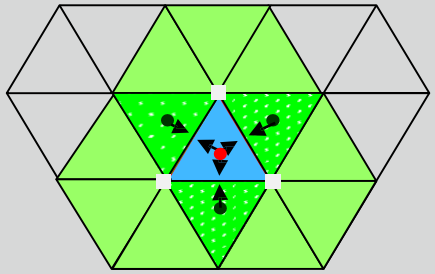
© 2012 ANSYS, Inc.

Interpolation Methods (Gradients)

- **Gradients of solution variables are required in order to evaluate diffusive fluxes, velocity derivatives, and for higher-order discretization schemes.**

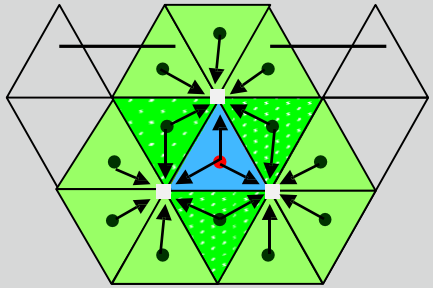
$$V_{\text{cell}}\frac{\partial(\rho\phi)}{\partial t}+\sum_{f}^{N_{\text{faces}}}\rho_f\mathbf{v}_f\phi_f\cdot\mathbf{A}_f=\sum_{f}^{N_{\text{faces}}}\Gamma_\phi\nabla\phi_f\cdot\mathbf{A}_f+S_\phi V_{\text{cell}}$$

$$\phi_f=\phi_{C0}+\beta\nabla\phi_{C0}^{RG}\cdot\overrightarrow{dr_0}$$



Cell based/ Least-Squares

- **The gradients of solution variables at cell centers can be determined using three approaches:**
  - Green-Gauss Cell-Based – Good, but solution may have false diffusion (smearing of the solution fields)
  - Green-Gauss Node-Based – More accurate; minimizes false diffusion; **(strongly recommended for tri/tet and hybrid meshes)**
  - Least-Squares Cell-Based – **The default method**. Less expensive to compute than Node-Based gradients. Slightly more expensive than Cell-Based gradients. However, exactly reconstruct linear field on highly skewed or distorted meshes. **(appropriate for any kind of meshes)**



Node-Based

© 2012 ANSYS, Inc.

Initialization

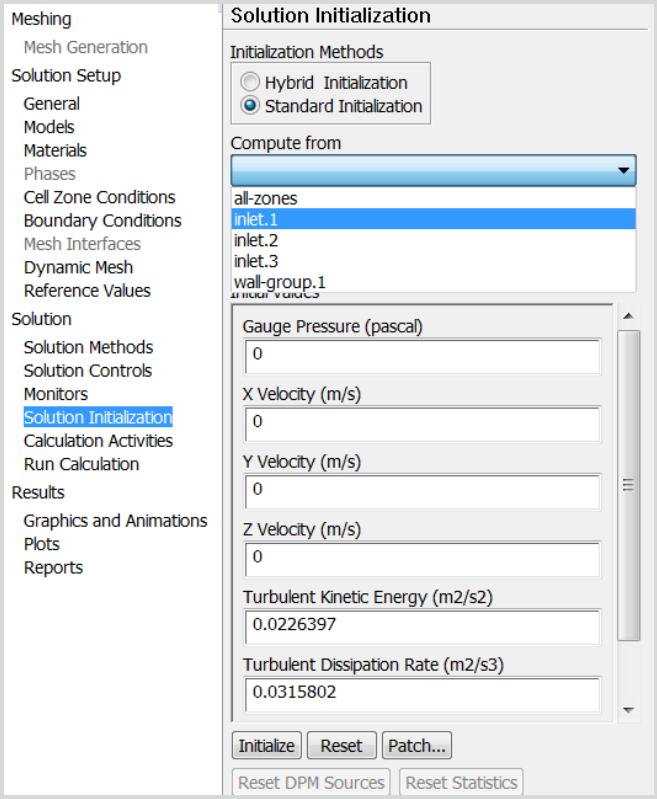
- **Fluent requires that all solution variables be initialized before starting iterations**
  - A realistic initial guess improves solution stability and accelerates convergence
  - In some cases a poor initial guess may cause the solver to fail during the first few iterations

- **5 ways to initialize the flow field**
  - Standard initialization
  - Patch values
  - Hybrid initialization (solves potential equation)
  - FMG initialization
  - Starting from a previous solution

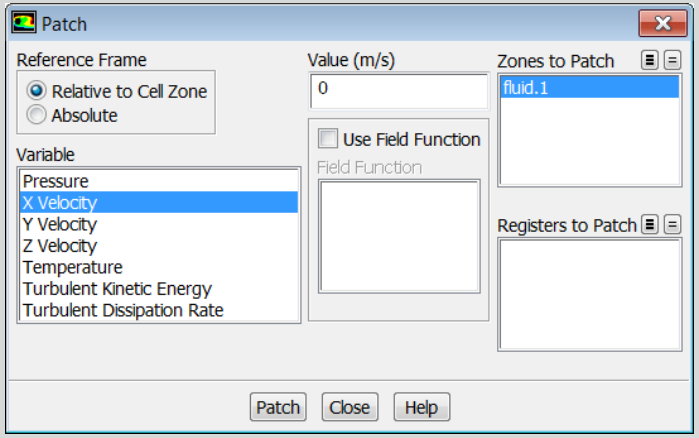
© 2012 ANSYS, Inc.

Standard Initialization and Patch Values

- **Standard Initialization**
  - Generally the user selects an inlet boundary under “Compute from” to automatically fill the initialization values with the values that are specified at the inlet boundary



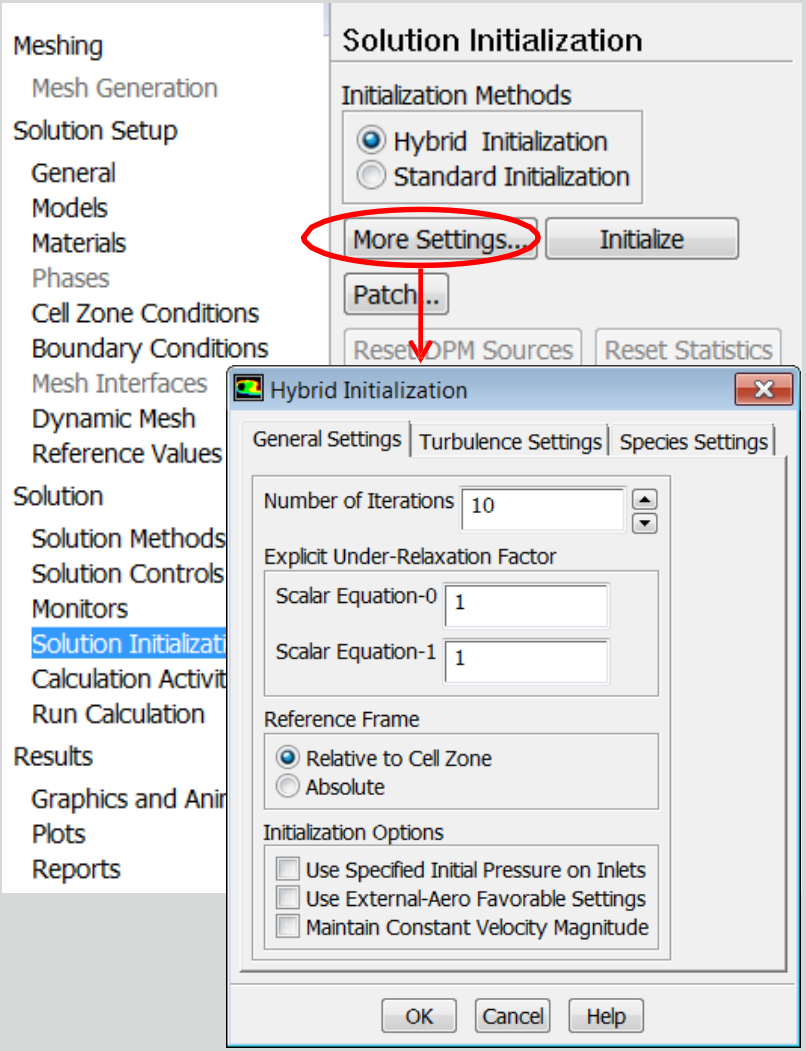
- **Patch values for individual variables in certain regions**
  - Free jet flows (high velocity for jet)
  - Combustion problems (high temperature region to initialize reaction)
  - Cell registers (created by marking the cells in the Adaption panel) can be used for patching values into various regions of the domain.
  - Multiphase flows (patch different phase volume fractions in one or more regions)



© 2012 ANSYS, Inc.

Hybrid Initialization

- The default initialization method
- This provides a quick approximation of the flow field, by a collection of methods.
- It solves Laplace's equation to determine the velocity and pressure fields.
- All other variables, such as temperature, turbulence, species mass fractions, volume fractions, etc., will be automatically patched based on domain averaged values or a particular interpolation method.



© 2012 ANSYS, Inc.

FMG Initialization

- **Full Multigrid (FMG) Initialization**
  - Can be used to create a better initialization of the flow field
    - FMG Initialization is useful for complex flow problems involving large pressure and velocity gradients on large meshes
  - FMG uses the Full Approximation Storage (FAS) Multigrid method to solve the flow problem on a sequence of coarser meshes
    - Euler equations are solved with first-order accuracy on the coarse-level meshes

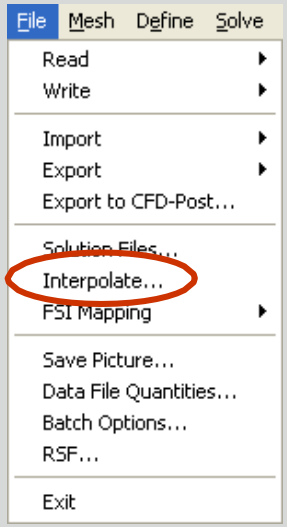
- **To enable FMG initialization, execute the TUI command**
  `/solve/init/fmg-initialization`

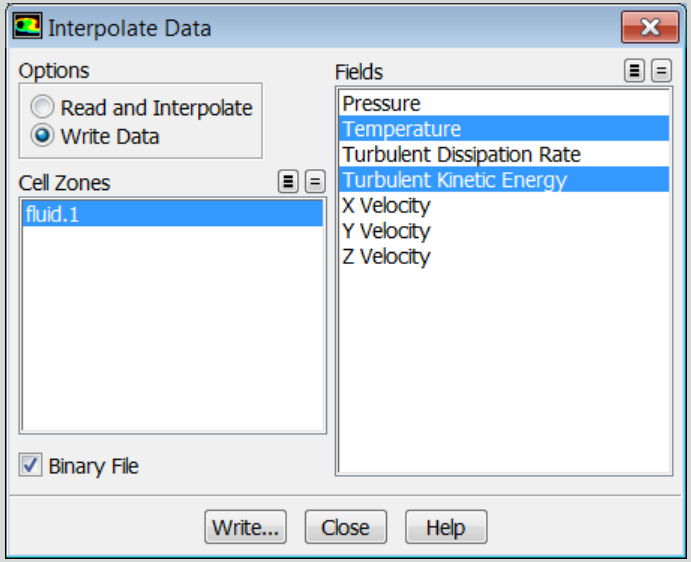
- **Settings can be accessed by the TUI command**
  `/solve/init/set-fmg-initialization`

© 2012 ANSYS, Inc.

Starting from a Previous Solution

- **A previously calculated solution can be used as an initial condition when changes are made to the case setup**
  - Use solution interpolation to initialize a run (especially useful for starting fine-mesh cases when coarse-mesh solutions are available)
  - Once the solution is initialized, additional iterations always use the current data set as the starting point





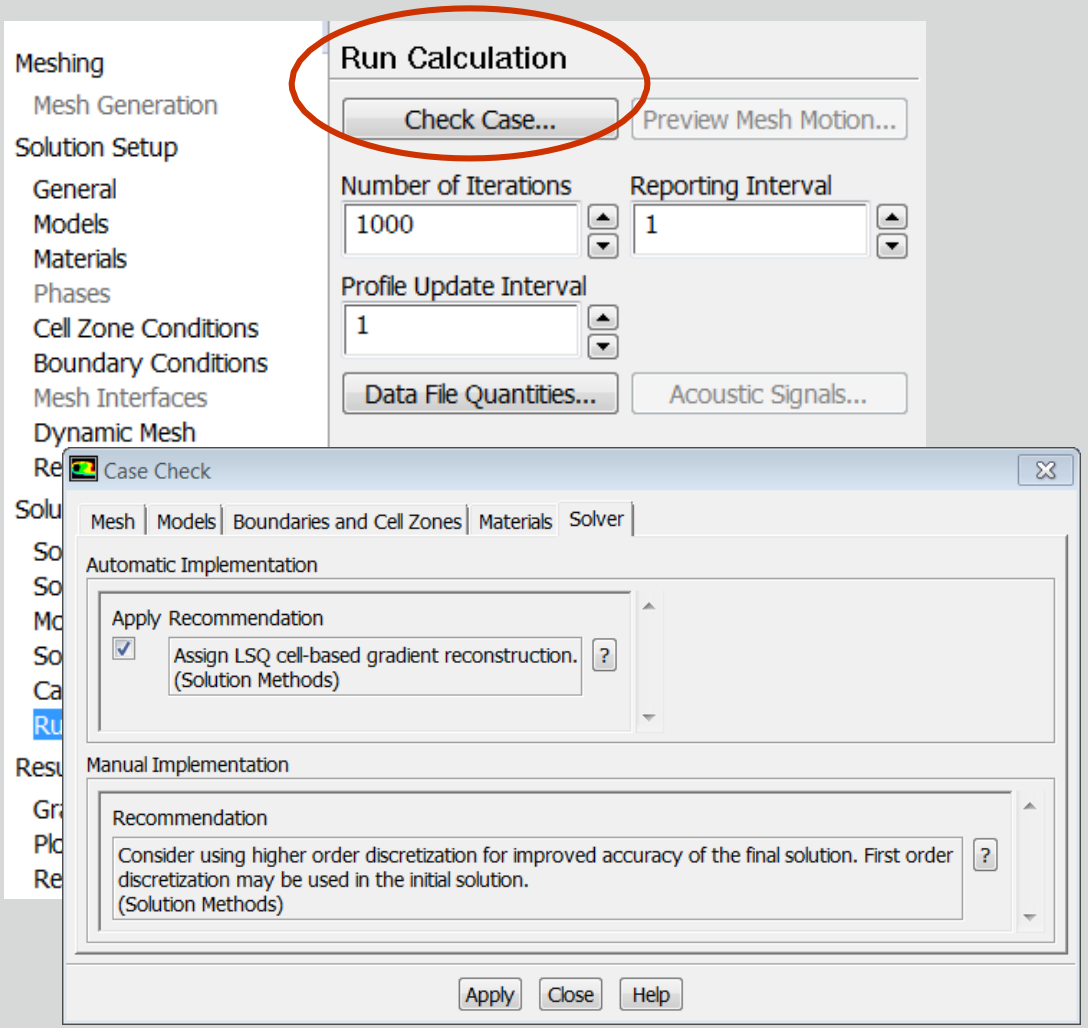
- **Sometimes solving a simplified version of the problem first will provide a good initial guess for the real problem:**

Actual Problem	Initial Condition
Heat Transfer	Isothermal
Natural convection	Low Rayleigh number
Combustion / reacting flow	Cold flow (no combustion)
Turbulence	Inviscid (Euler) solution

© 2012 ANSYS, Inc.

Case Check

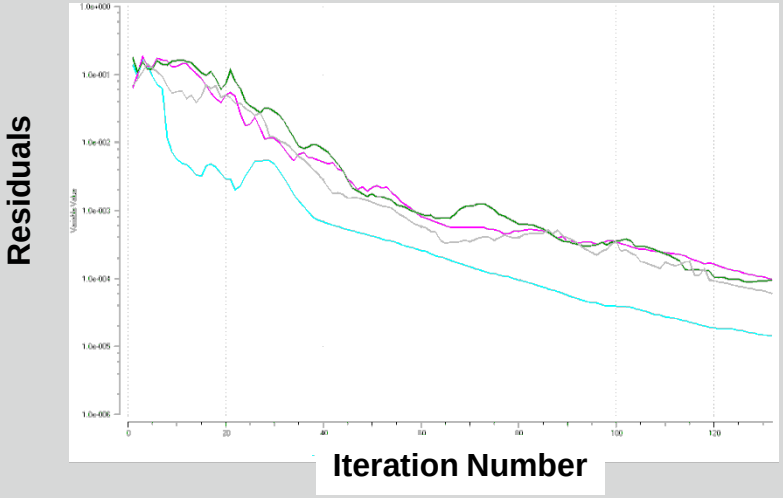
- **Case Check is a utility in Fluent which searches for common setup errors and inconsistencies**
  - Provides guidance in selecting case parameters and models
- **Tabbed sections contain recommendations which the user can apply or ignore**

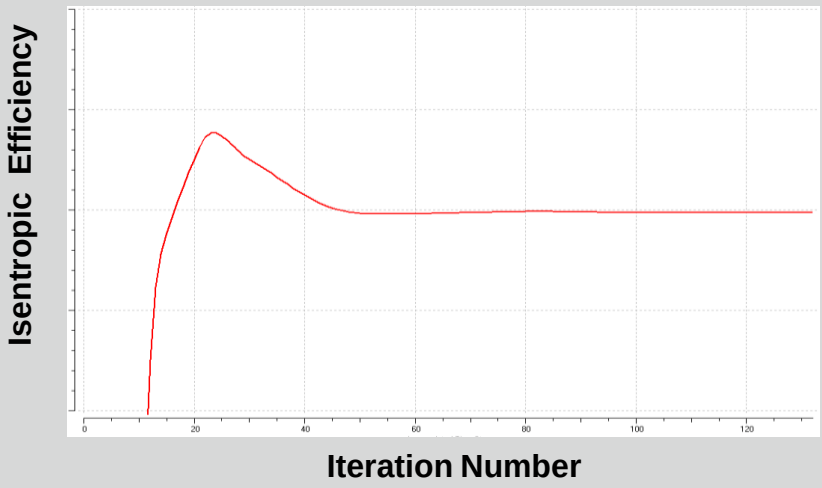


© 2012 ANSYS, Inc.

Convergence

- **The solver must perform enough iterations to achieve a converged solution**
- **At convergence, the following should be satisfied:**
  - All discrete conservation equations (momentum, energy, etc.) are obeyed in all cells to a specified tolerance (Residual).
    - The residual measures the imbalance of the current numerical solution and is related but NOT EQUAL to the numerical error
  - Overall mass, momentum, energy, and scalar balances are achieved
  - Target quantities reach constant values
    - Integral: e.g. Pressure drop
    - Local: e.g. Velocity at specified position





© 2012 ANSYS, Inc.

Convergence

- **Monitoring convergence using residual history:**
  - Generally, a decrease in residuals by three orders of magnitude can be a sign of convergence
  - Scaled energy residual should decrease to 10^{-6} (for the pressure-based solver)
  - Scaled species residual may need to decrease to 10^{-5} to achieve species balance

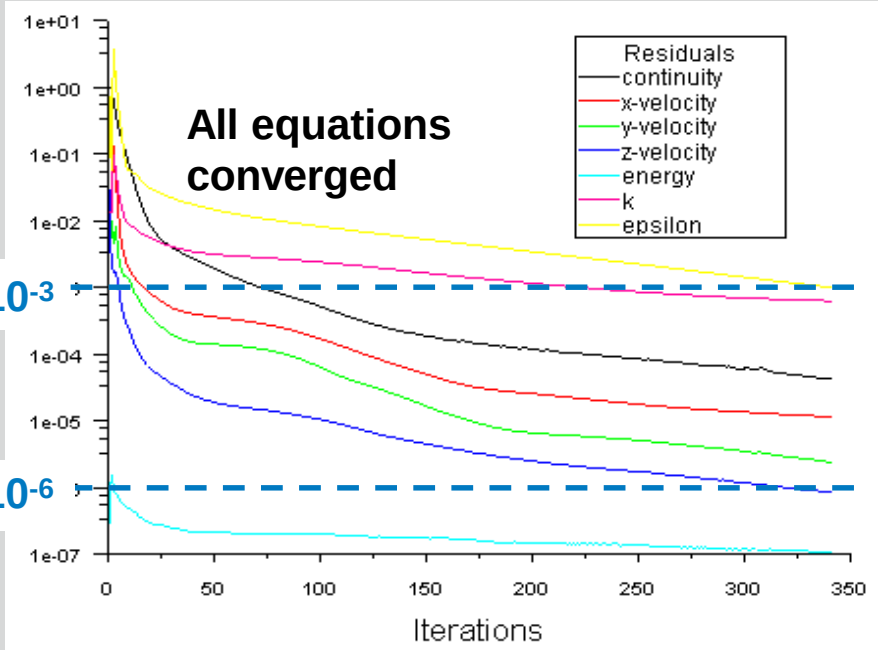
- **Best practice is to also monitor quantitative variables to decide convergence:**
  - Ensure that overall mass/heat/species conservation is satisfied
  - Monitor other relevant key variables/physical quantities for confirmation

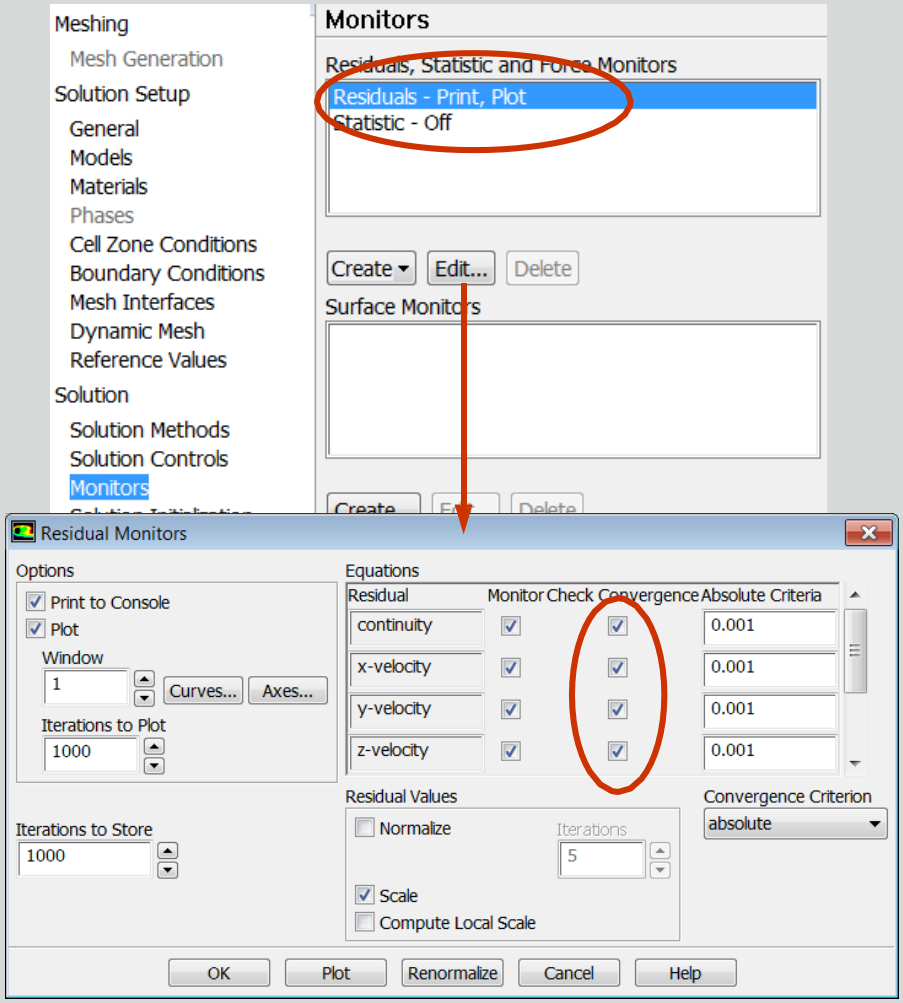
© 2012 ANSYS, Inc.

Convergence Monitors – Residuals

- **Residual plots show when the residual values have reached the specified tolerance**
  - It is possible to modify or disable the default checking criterion for convergence

    (to prevent the calculation from being stopped prematurely)

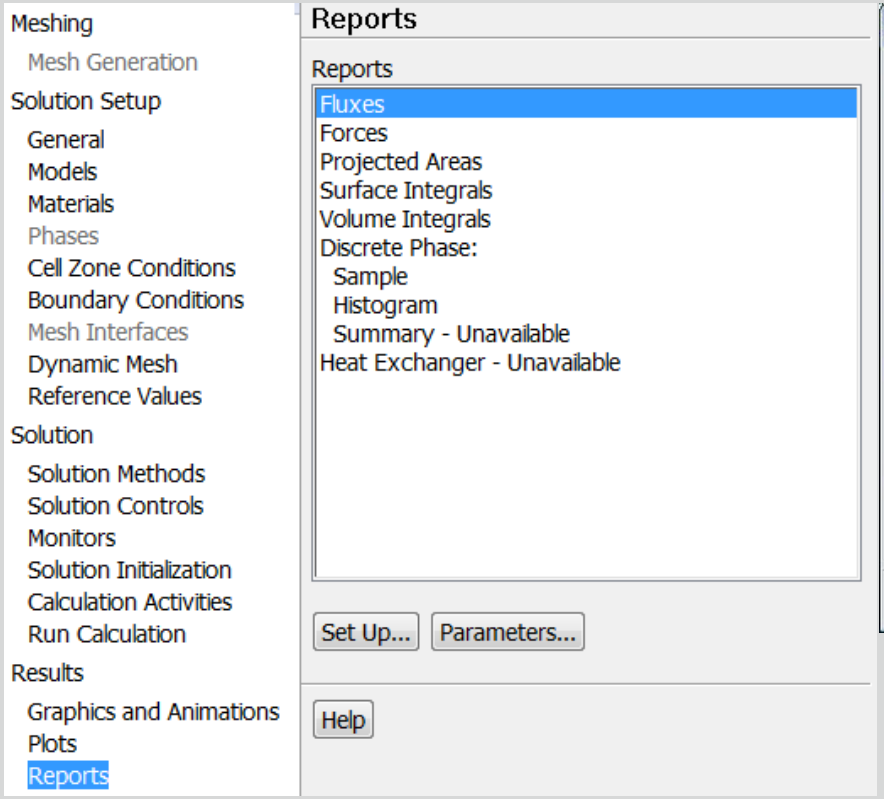


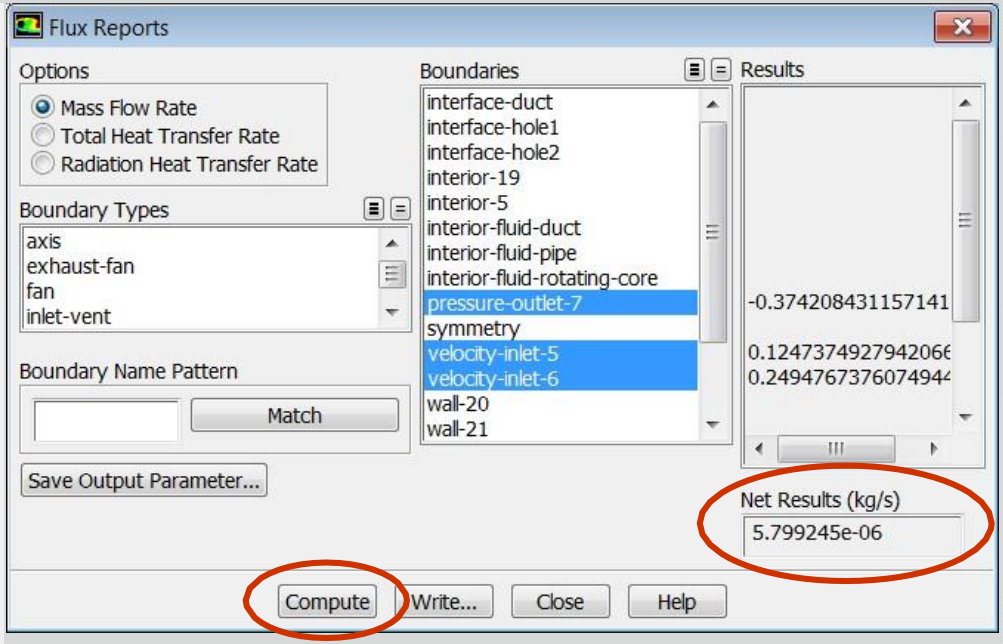


© 2012 ANSYS, Inc.

Checking Overall Flux Conservation

- **The net flux imbalance (shown in the GUI as Net Results) should be less than 1% of the smallest flux through the domain boundary**

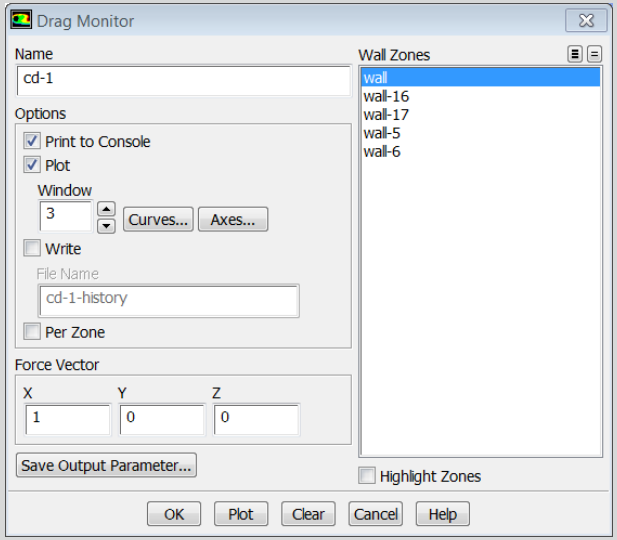


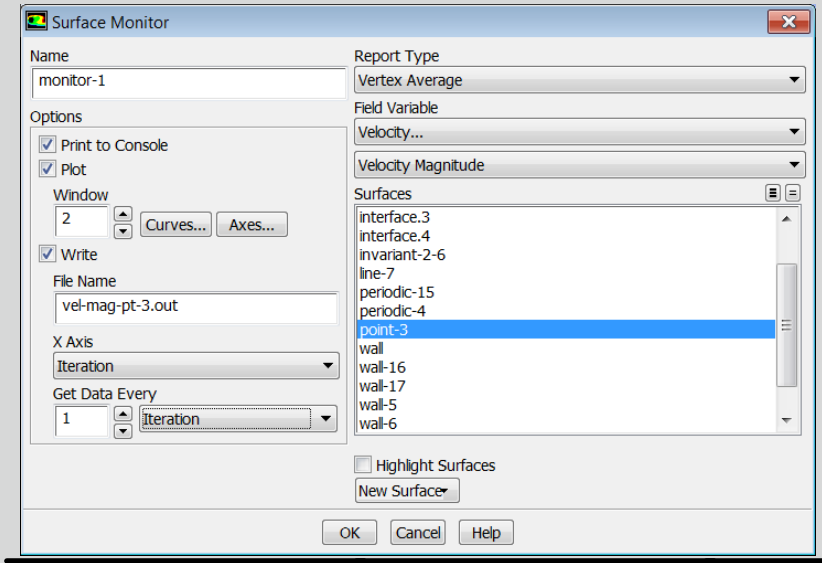


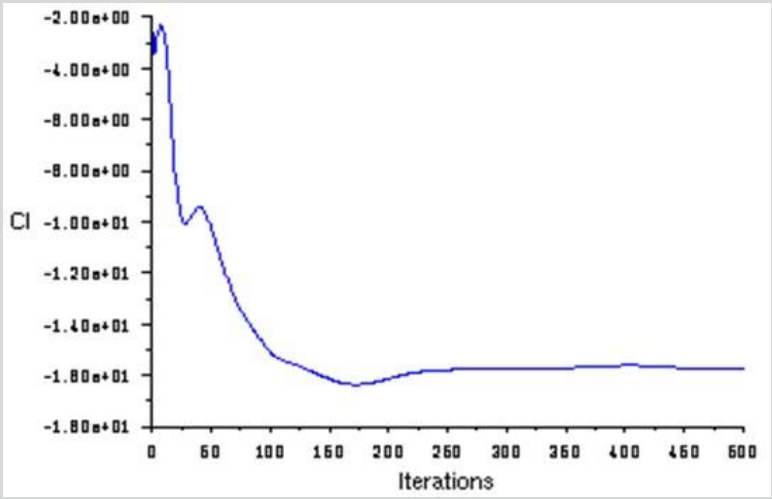
© 2012 ANSYS, Inc.

Convergence Monitors – Forces and Surfaces

- In addition to residuals, you can also monitor:
  - Lift, drag and moment coefficients
  - Relevant variables or functions (e.g. surface integrals) at a boundary or any defined surface
- These additional monitored quantities are important convergence indicators
- These monitors can also be used to determine when iterations stop (details in Appendix)



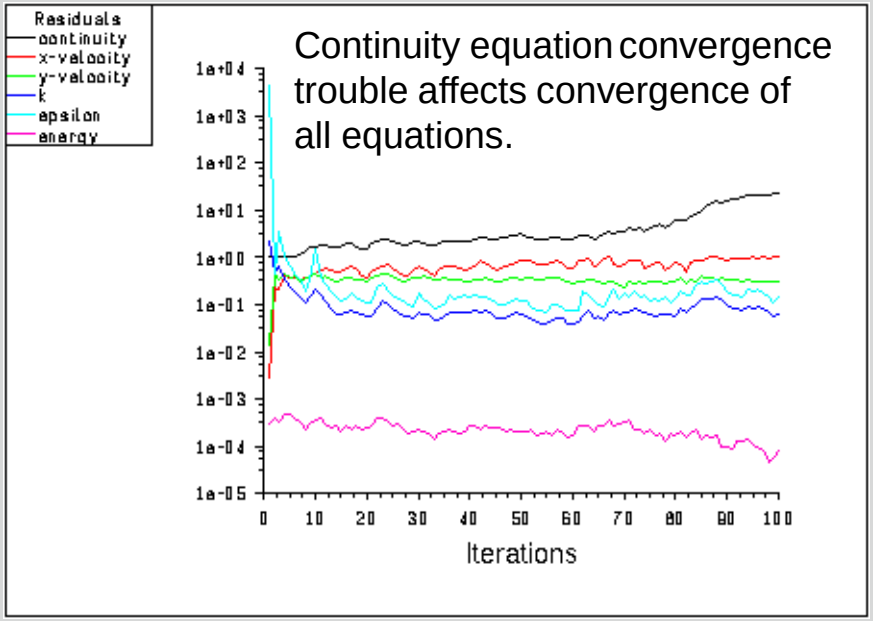




© 2012 ANSYS, Inc.

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 that the problem is well-posed
  - Compute an initial solution using a first-order discretization scheme
  - For the pressure-based solver, decrease underrelaxation factors for equations having convergence problems
  - For the density-based solver, reduce the Courant number
  - Remesh or refine cells which have large aspect ratio or large skewness.
    - Remember that you cannot improve cell skewness by using mesh adaption!



© 2012 ANSYS, Inc.

Accelerating Convergence

- **Convergence can be accelerated by:**
  - Supplying better initial conditions
    - Starting from a previous solution (using file/interpolation when necessary)
  - Gradually increasing under-relaxation factors or Courant number
    - Excessively high values can lead to solution instability and convergence problems
    - You should always save case and data files before continuing iterations
  - Starting with a good quality mesh with appropriate mesh resolution
    - The orthogonal quality reported in Mesh > Info > Quality should have a minimum value of .01 and an average value that is much higher

© 2012 ANSYS, Inc.

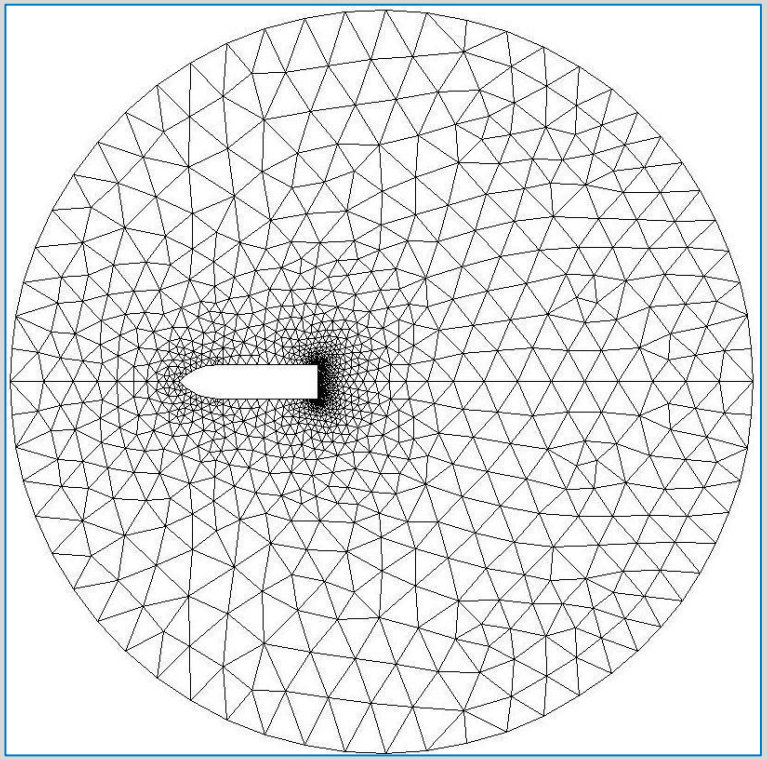
Convergence vs Accuracy

- **A converged solution is not necessarily an accurate solution**
  - Accuracy depends on :
    - Order of the discretization schemes (2nd order schemes are recommended)
    - Mesh resolution
    - Boundary Conditions
    - Model limitations
    - Geometry simplifications
    - Precision of the solver (2d/3d or 2ddp/3ddp)
    - …

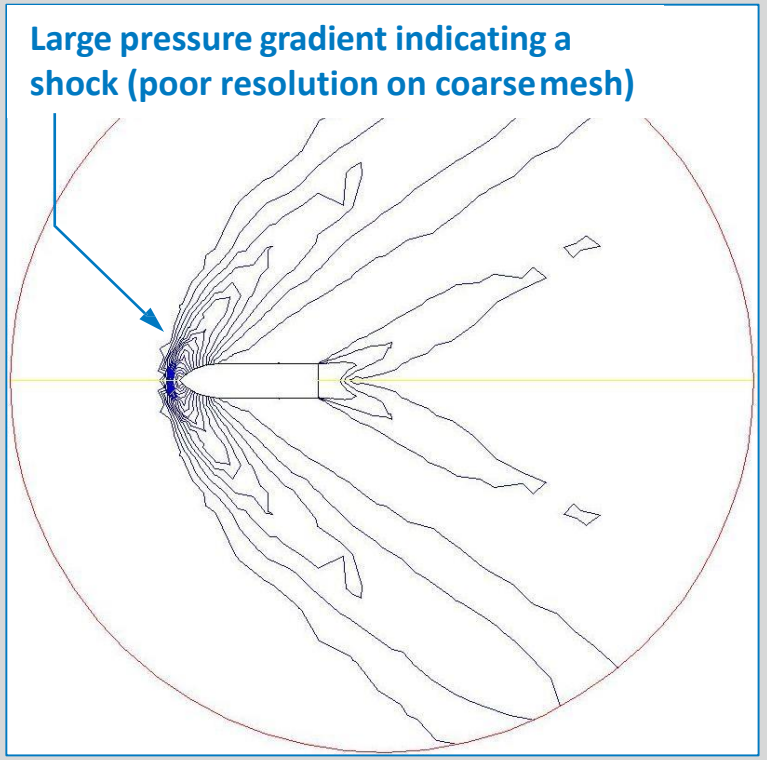
© 2012 ANSYS, Inc.

Adaption Example – 2D Planar Shell

- **Adapt grid in regions of large pressure gradient to better resolve the sudden pressure rise across the shock**



Initial Mesh



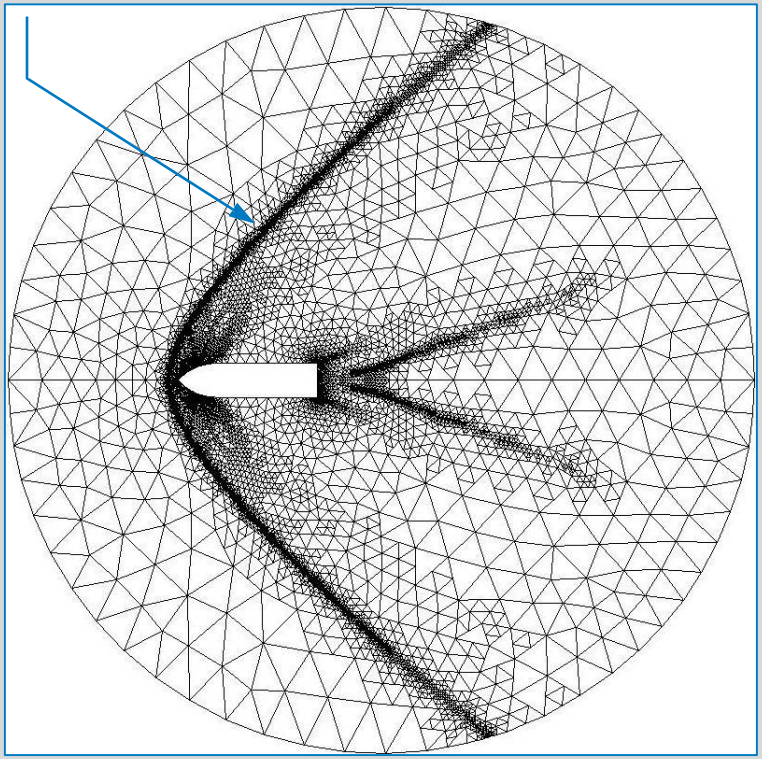
Pressure Contours on Initial Mesh

© 2012 ANSYS, Inc.

2D Planar Shell – Solution on Adapted Mesh

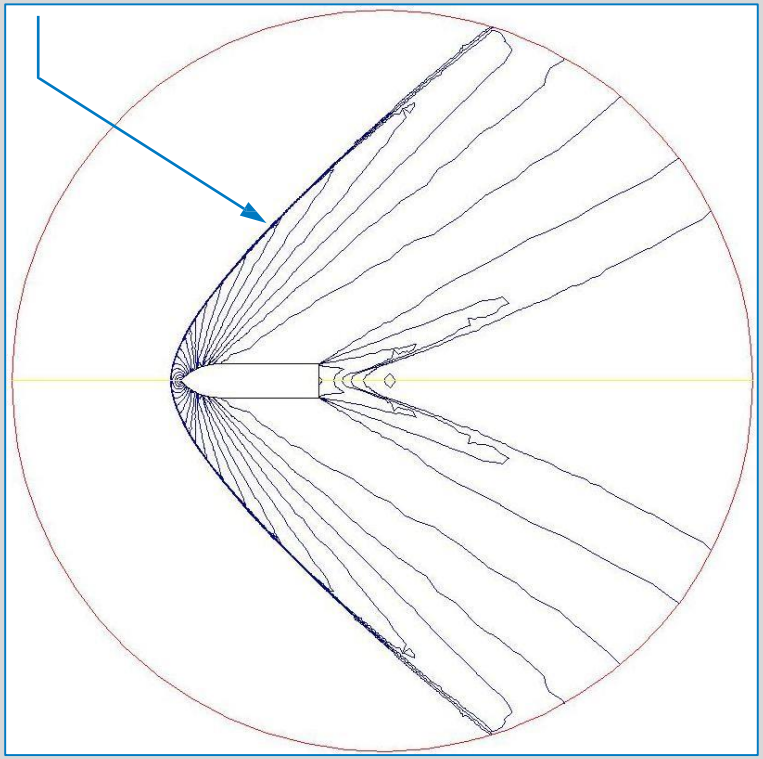
- **Solution-based mesh adaption allows better resolution of the bow shock and expansion wave.**

Adapted cells in locations of large pressure gradients



Adapted Mesh

Mesh adaption yields much-improved resolution of the bow shock.

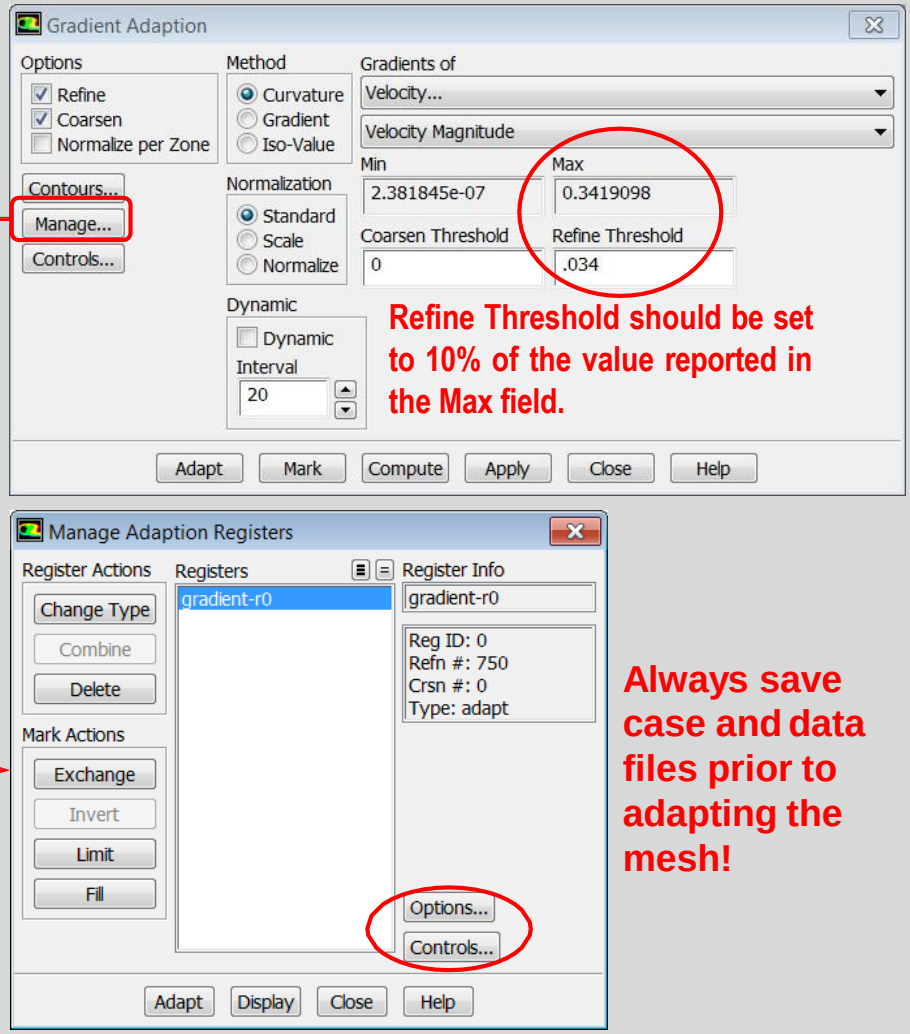


Pressure Contours on Adapted Mesh

© 2012 ANSYS, Inc.

Grid Adaption

- **Grid adaption adds more cells where needed to resolve the flow field without the pre-processor**
- **Adaption proceeds in three steps:**
  - Mark cells satisfying the adaption criteria and store them in a "register"
  - Display and modify the register
  - Click on Adapt to adapt the cells listed in the register
- **Registers can be defined based on:**
  - Gradients or iso-values of all variables
  - All cells on a boundary
  - All cells in a region with a defined shape
  - 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



© 2012 ANSYS, Inc.

Summary

- **Solution procedure for both the pressure-based and density-based solvers is identical**
  - Set adequate solver & solution parameters
  - Calculate until you get a converged solution
  - Obtain a second-order solution (recommended)
  - Refine the mesh and recalculate until a grid-independent solution is obtained

- **All solvers provide tools for judging and improving convergence and ensuring stability**

- **A converged solution is not necessarily accurate**
  - Solution accuracy depends on Mesh resolution, appropriateness of the physical models, boundary conditions, …

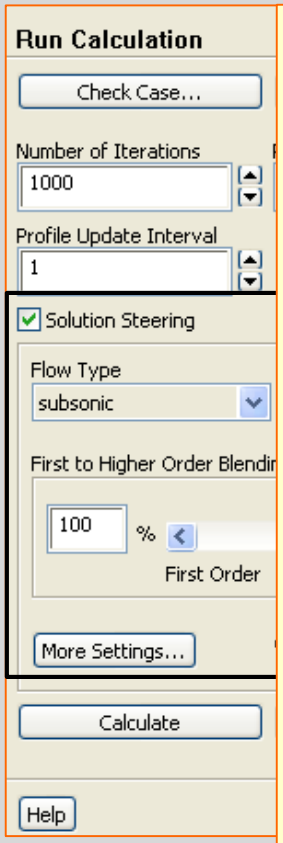
© 2012 ANSYS, Inc.

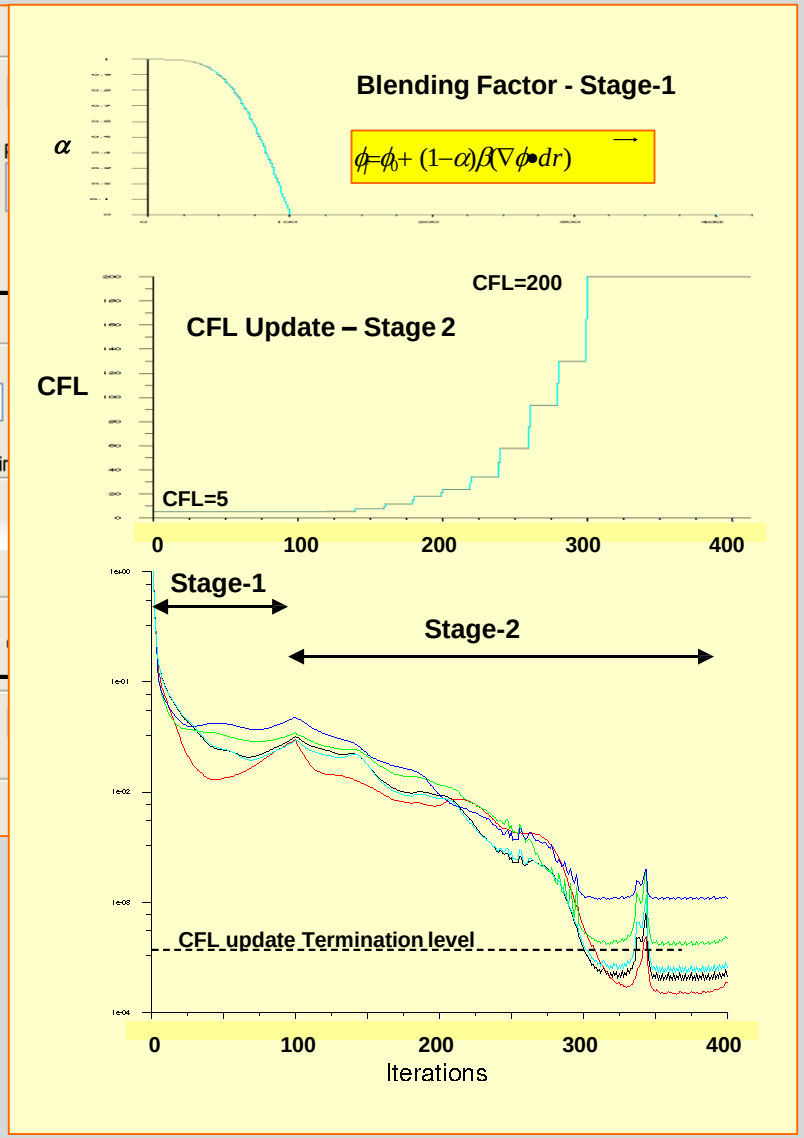
Appendix

DBS Iterative Procedure – Solution Steering

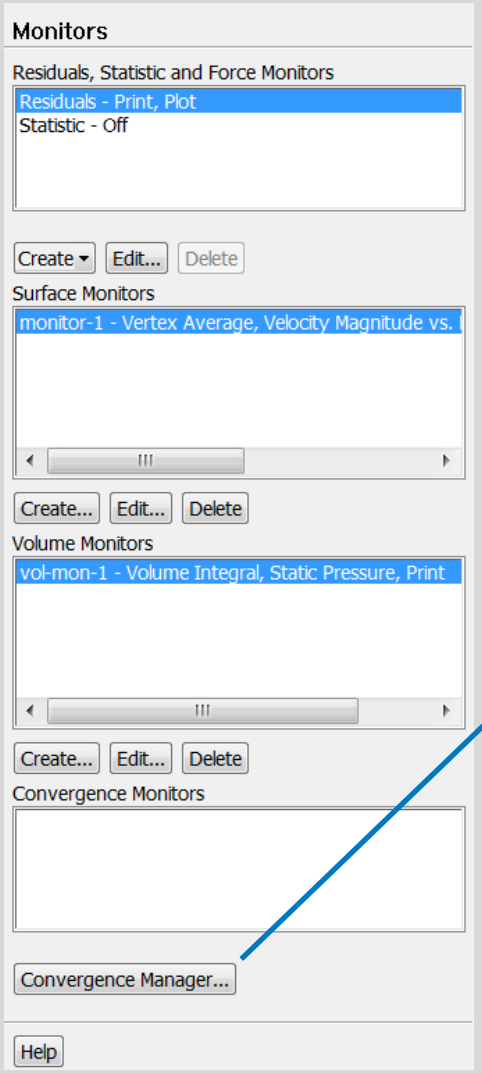
- **Objective:**
  - Provide an expert system that will navigate the solution from the difficult initial starting solution to a converged solution with minimum user intervention and tweaking
    - Particularly useful for high speed compressible flows

- **Activating Solution Steering:**
  - Set up your case the normal way
  - Perform initialization (i.e. typically from flow inlets)
  - Activate Solution Steering
  - Select type of flow that best characterizes the problem
    - incompressible, subsonic, transonic, supersonic , hypersonic flows
  - Start iterating

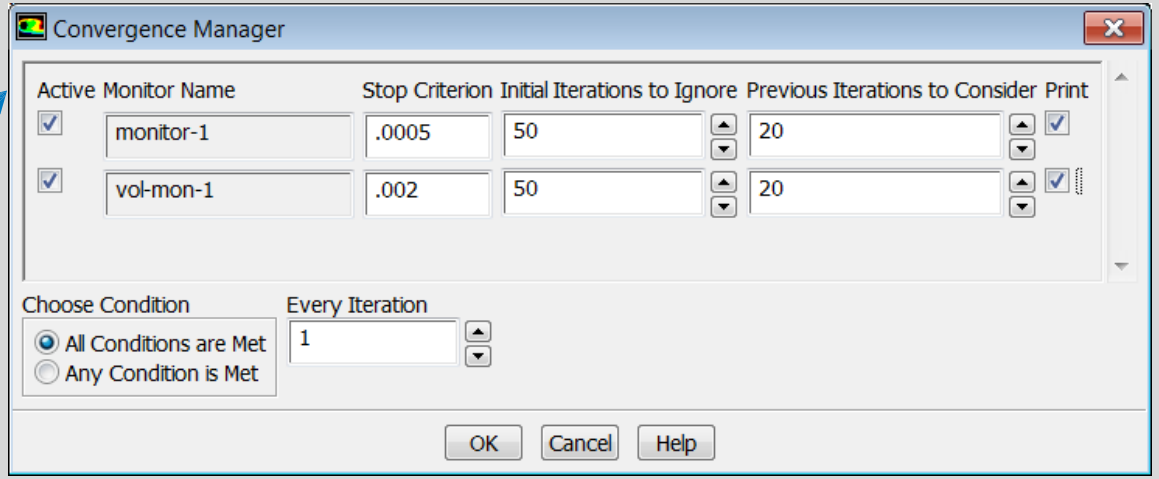




Monitors as Convergence Criteria



- Use monitors to determine solution convergence
  - Lift and drag monitors, for example
  - Use with or without residual monitors
- Stop criterion based on solution monitor 'residual', which is defined as
  - Res-m(1)=abs[m(n)-m(n-1)]/m(n)
  - where m(n) = monitor value at iteration number 'n'
- Solution will be stopped when 'residual' is less than the stop criterion over specified number of previous iterations

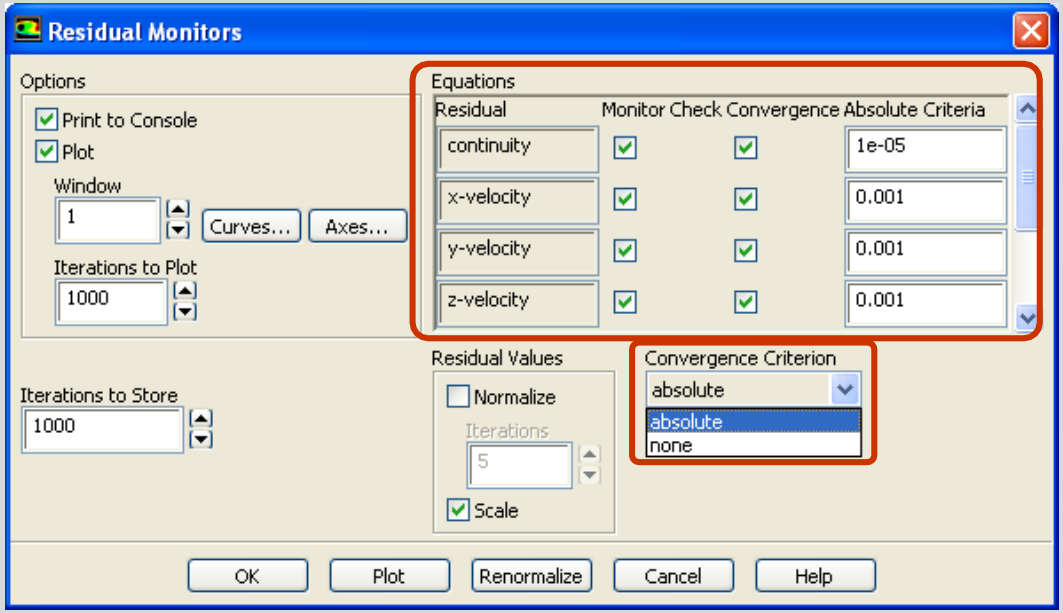


Tightening the Convergence Tolerance

- **If solution 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**

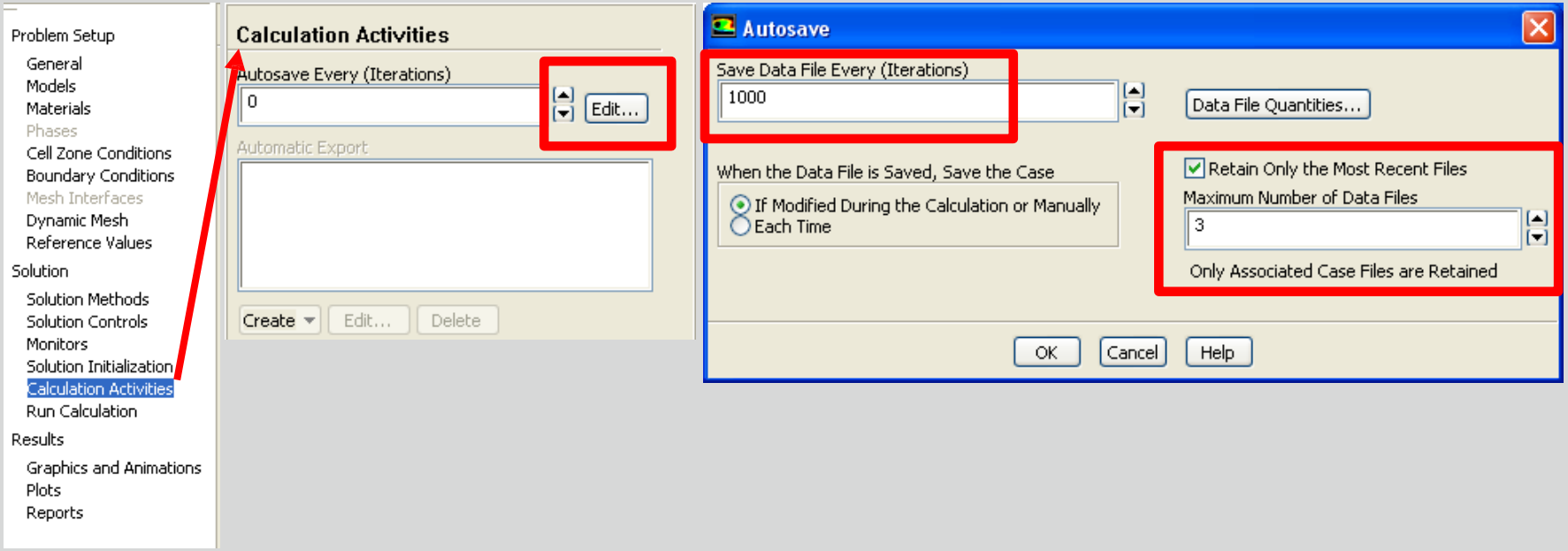
- **In this case, you need to:**
  - Reduce values of Convergence Criteria or disable Check Convergence in the Residual Monitors panel
  - Continue iterations until the solution converges

- **Selecting None under Convergence Criterion disables convergence checking for all equations**



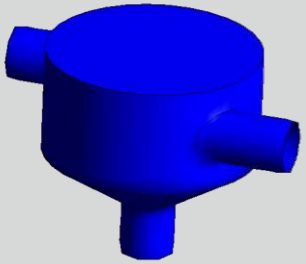
Autosave

- The solution can be saved at regular intervals during the calculation using the Autosave menu, which is accessed through the Calculation Activities menu panel
  - This can be helpful in situations where convergence difficulties are encountered, because it makes it possible to return to an earlier point in the calculation
  - Keeping a large number of Autosave files can use a large amount of disk space, therefore in most cases, it is recommended to use the option to retain only the most recent files

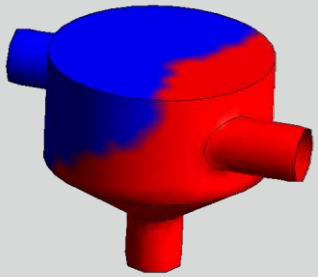


Running Simulations in Parallel

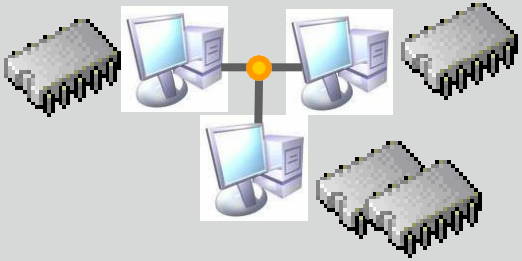
- Serial

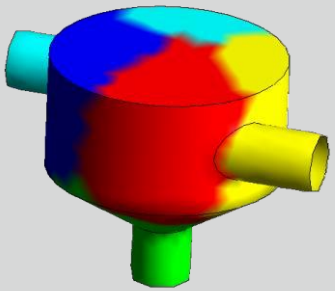


- Local Parallel
  - Shared Memory



- Distributed Parallel
  - Distributed Memory

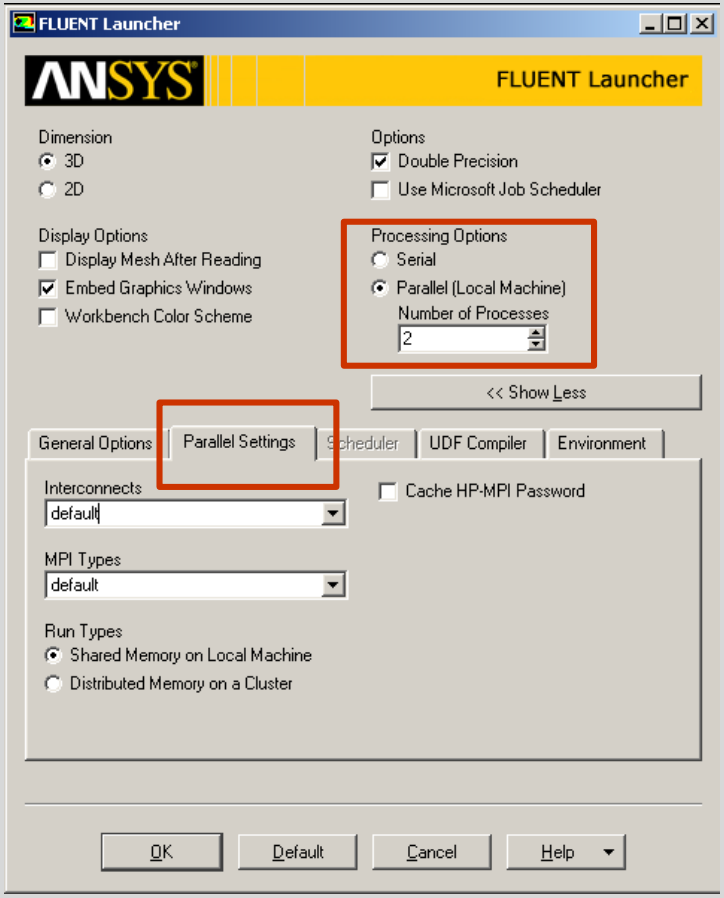




- Different communication methods are available (MPICH2, HP MPI, PVM)
  - See documentation "When To Use MPI or PVM" for more details, but HP MPI is recommended in most cases

Running Simulations in Parallel …

- **In the Fluent Launcher you can choose Parallel and set the Parameter**

- **If you choose Distributed Memory, you have to specify the names of the computers which you want to connect**
  - You can specify the names directly
  - You can specify a file which contains the names

- **For further information see Chapter 32 in User Guide**