

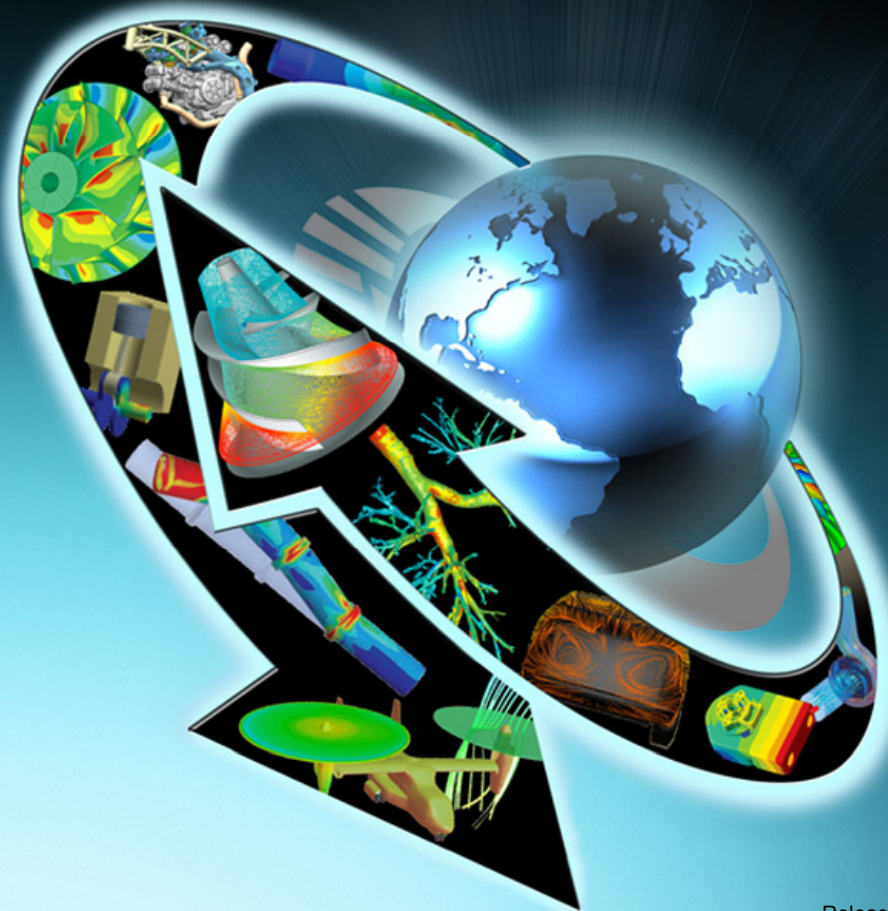


Customer Training Material

Lecture 5

Solver Settings

Introduction to ANSYS FLUENT

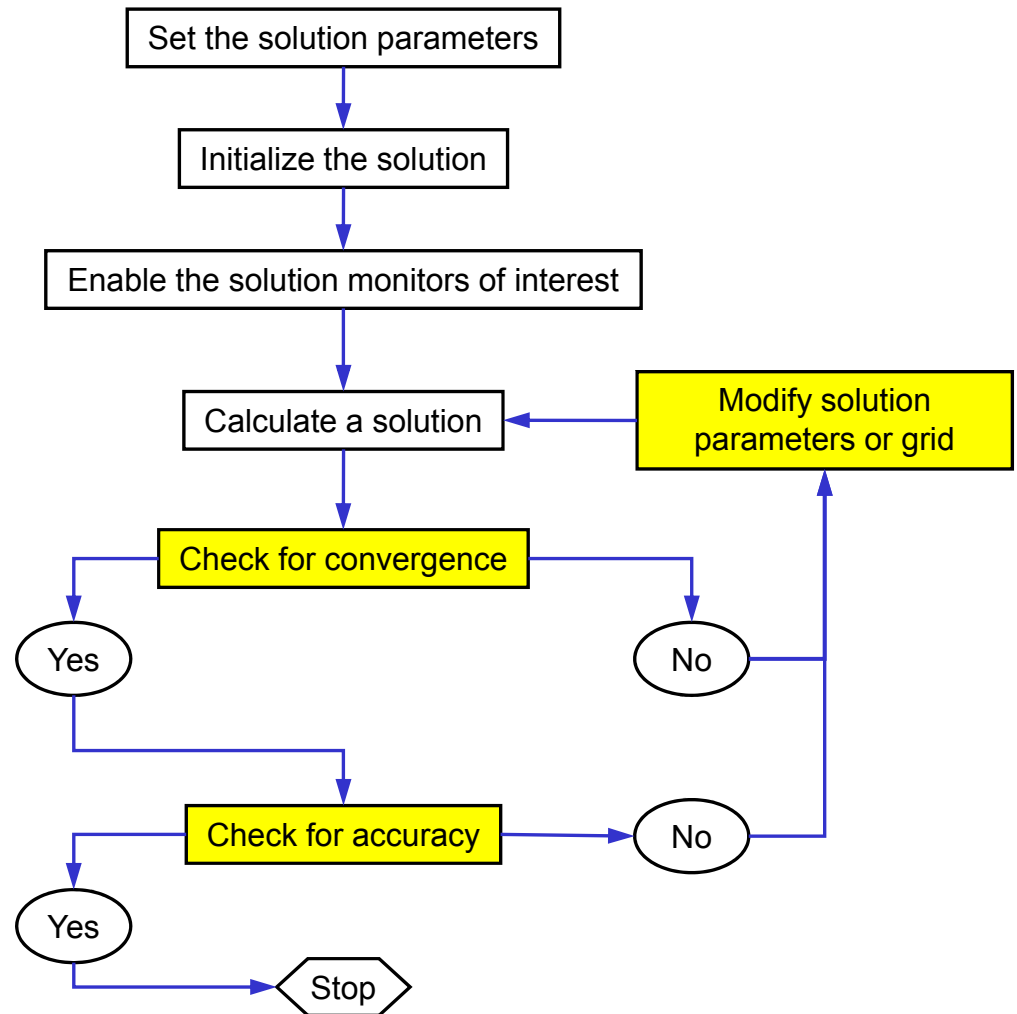


Solver Settings - Introduction

- So far we have looked at how to setup a basic flow simulation in FLUENT.
- However you should not assume that just because you have 'an answer' that that is the 'correct answer'.
- By modifying the solver settings you can improve both:
 - The rate of convergence of the simulation
 - The accuracy of the computed result.

Solution Procedure Overview

- The sketch opposite shows the basic workflow for any simulation.
- This lecture will look at:
 - the choice of solver
 - discretisation schemes
 - checking convergence
 - assessing accuracy



-
- ```

graph TD
 subgraph "Pressure-Based (segregated)"
 direction TB
 A1[Solve U-Momentum]
 A2[Solve V-Momentum]
 A3[Solve W-Momentum]
 A4[Solve Mass Continuity;
Update Velocity]
 end

 subgraph "Pressure-Based (coupled)"
 direction TB
 B1[Solve Mass & Momentum]
 end

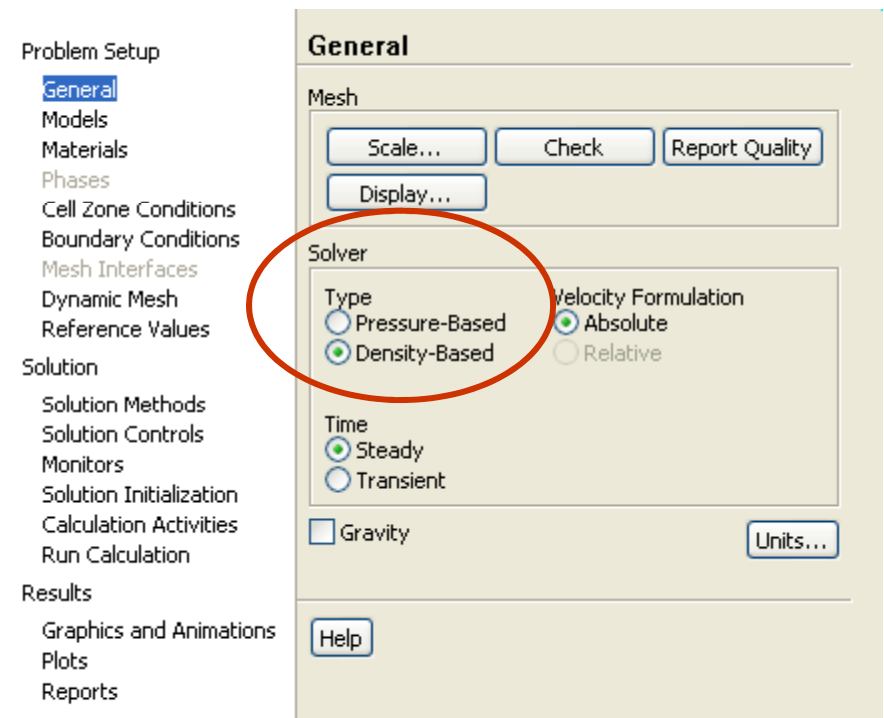
 subgraph "Density-Based (coupled)"
 direction TB
 C1[Solve Mass, Momentum,
Energy, Species]
 end

 A4 --> D1[Solve Energy]
 B1 --> D1
 C1 --> D3[Solve Turbulence Equation(s)]

 D1 --> D2[Solve Species]
 D2 --> D3
 D3 --> D4[Solve Other Transport Equations as required]

```
- The flowchart illustrates the solution sequence for three different CFD approaches:
- Pressure-Based (segregated):** This approach involves solving the momentum equations (U, V, W) and the mass continuity equation sequentially. The mass continuity equation is used to update the velocity.
  - Pressure-Based (coupled):** This approach involves solving the mass and momentum equations simultaneously.
  - Density-Based (coupled):** This approach involves solving the mass, momentum, energy, and species equations simultaneously.
- The solution sequence is as follows:
- Solve U-Momentum, V-Momentum, and W-Momentum (segregated approach).
  - Solve Mass Continuity; Update Velocity (segregated approach).
  - Solve Mass & Momentum (coupled approach).
  - Solve Mass, Momentum, Energy, Species (coupled approach).
  - Solve Energy (segregated approach).
  - Solve Species (segregated approach).
  - Solve Turbulence Equation(s) (segregated approach).
  - Solve Other Transport Equations as required (segregated approach).

- **Density-Based Coupled Solver**
  - Equations for continuity, momentum, energy and species (if required) are solved in vector form.
  - Pressure is obtained through an equation of state.
  - Additional scalar equations are solved in a segregated fashion.
- The DBCS can be run either explicit or implicit.
  - Implicit – Uses a point-implicit Gauss-Seidel / symmetric block Gauss-Seidel / ILU method to solve for variables.
  - Explicit: uses a multi-step Runge-Kutta explicit time integration method



# Choosing a Solver

- The **pressure-based solver** is applicable for a wide range of flow regimes from low speed incompressible flow to high-speed compressible flow.
  - Requires less memory (storage).
  - Allows flexibility in the solution procedure.
- The **pressure-based coupled solver** (PBCS) is applicable for most single phase flows, and yields superior performance to the standard pressure-based solver.
  - Not available for multiphase (Eulerian), periodic mass-flow and NITA cases.
  - Requires 1.5–2 times more memory than the segregated solver.
- The **density-based coupled solver** (DBCS) is applicable when there is a strong coupling, or interdependence, between density, energy, momentum, and/or species.
  - Examples: High speed compressible flow with combustion, hypersonic flows, shock interactions.
  - The implicit option is generally preferred over explicit since it has a very strict limit on time step size
  - 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. (e.g. propagation of high-Ma shock waves).

# Solver Settings

## Discretisation

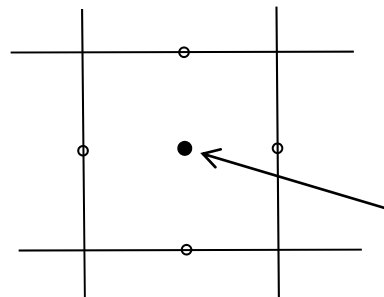


- In FLUENT, solver variables are stored at the centre of the grid cells (control volumes).

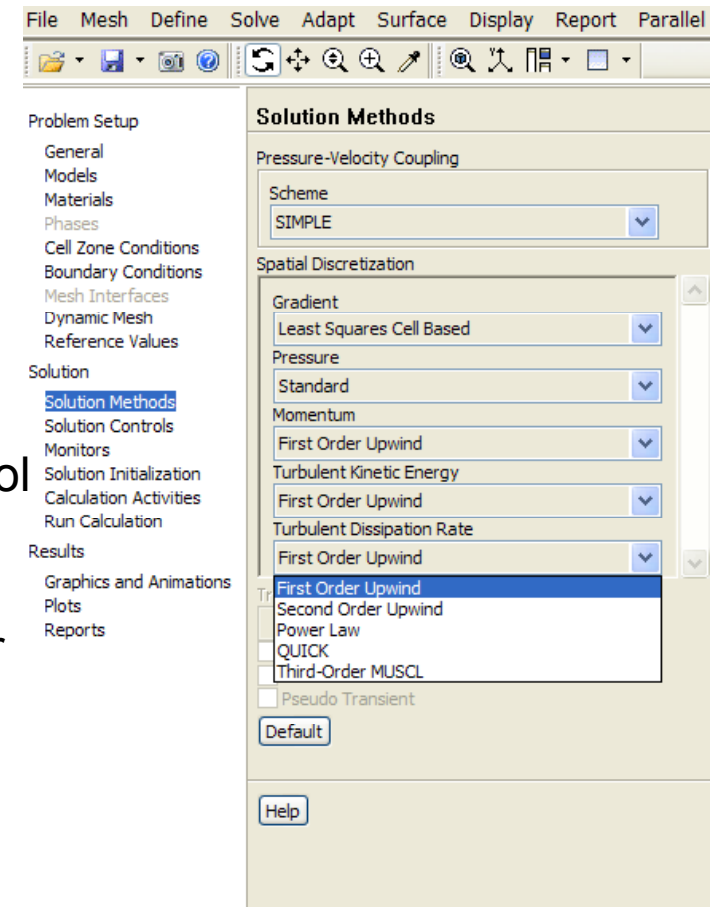
- Recall the general form of the transport equations shown in an earlier lecture:

$$\frac{\partial}{\partial t} \int_V \rho \phi dV + \oint_A \rho \phi \mathbf{V} \cdot d\mathbf{A} = \oint_A \Gamma_\phi \nabla \phi \cdot d\mathbf{A} + \int_V S_\phi dV$$

- To implement these equations, we need to know the values of this quantity  $\phi$  at the faces of the control volumes, and the gradients of  $\phi$  for the grid cell.
- The following slides show the options available for interpolating the cell-centre data, and these are set as 'Solution Methods'

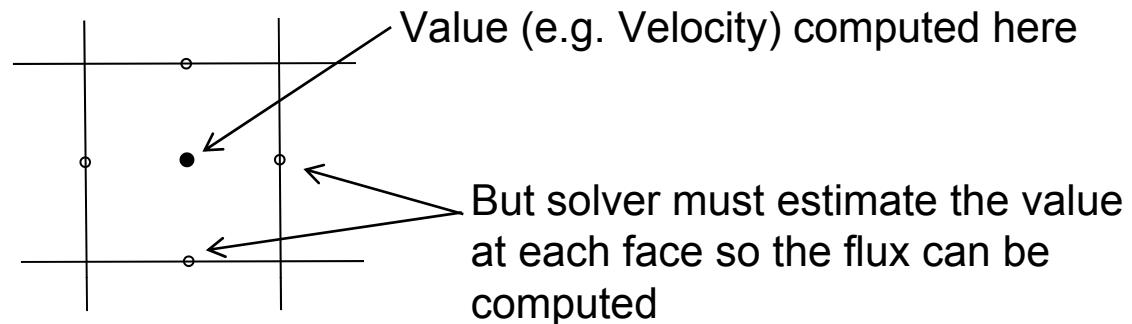


Value (e.g. Velocity) computed here



# Discretisation (Interpolation Methods)

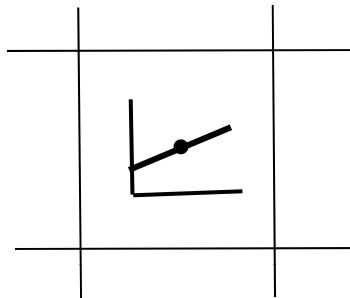
- Field variables (stored at cell centers) must be interpolated to the faces of the control volumes.



- 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 discretisation 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 mesh.

# Interpolation Methods (Gradients)

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



- The gradients of solution variables at cell centers can be determined using three approaches:
  - Green-Gauss Cell-Based** – Least computationally intensive. Solution may have false diffusion.
  - Green-Gauss Node-Based** – More accurate/computationally intensive; minimizes false diffusion; recommended for unstructured meshes.
  - Least-Squares Cell-Based** – Default method; has the same accuracy and properties as Node-based Gradients and is less computationally intensive.

# Interpolation Methods for Pressure

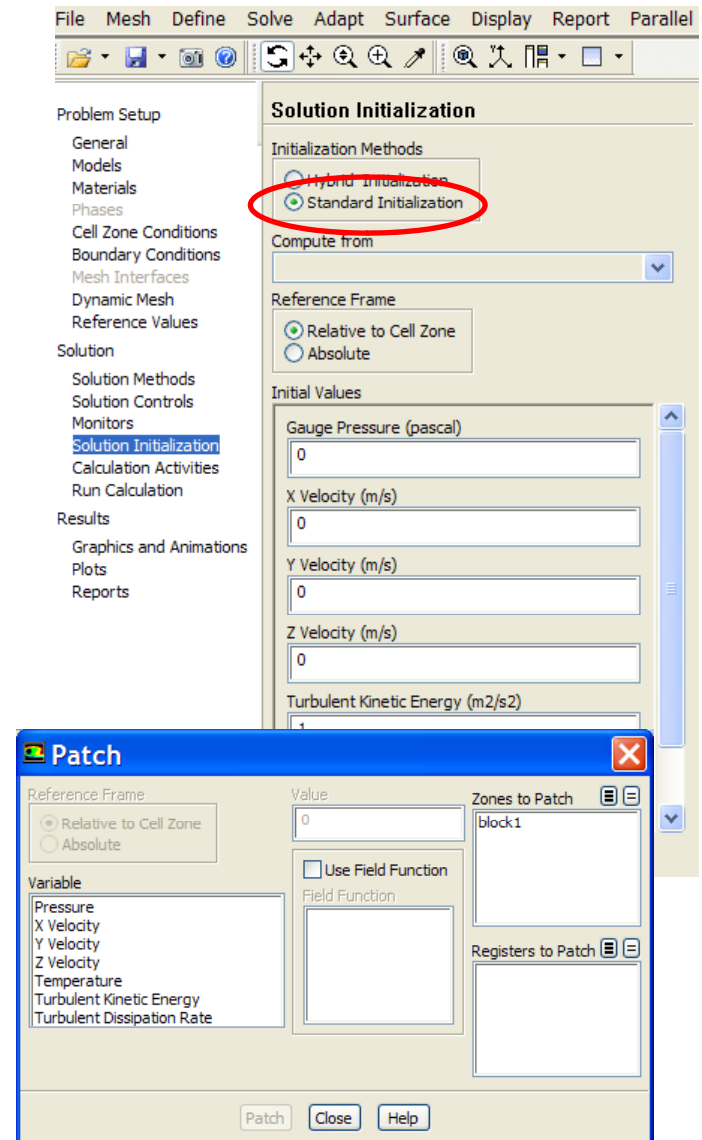
- Interpolation schemes for calculating cell-face pressures when using the pressure-based solver in FLUENT are available as follows:
  - **Standard** – The default scheme; reduced accuracy for flows exhibiting large surface-normal pressure gradients near boundaries (but should not be used when steep 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

# 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 pressure-based solver.
- Five algorithms are available in FLUENT.
  - Semi-Implicit Method for Pressure-Linked Equations (**SIMPLE**)
    - The default scheme, robust
  - SIMPLE-Consistent (**SIMPLEC**)
    - Allows faster convergence for simple problems (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.
    - Used with the NITA scheme; similar characteristics as PISO.
  - **Coupled** (this is how the pressure-based coupled solver, described previously, is enabled)

# Standard Initialization

- The solver works in an iterative manner.
- Therefore before the very first iteration, a value must exist for every quantity in every grid cell.
- Setting this value is called 'Initialization'
- **The more realistic the value, the better (quicker) convergence will be.**
- It is possible to mark some cells (using the adaption tools, or having different cell zones) and patch specific values into certain cells.
  - Free jet flows (high velocity for jet)
  - Combustion problems (high temperature region to initialize reaction)

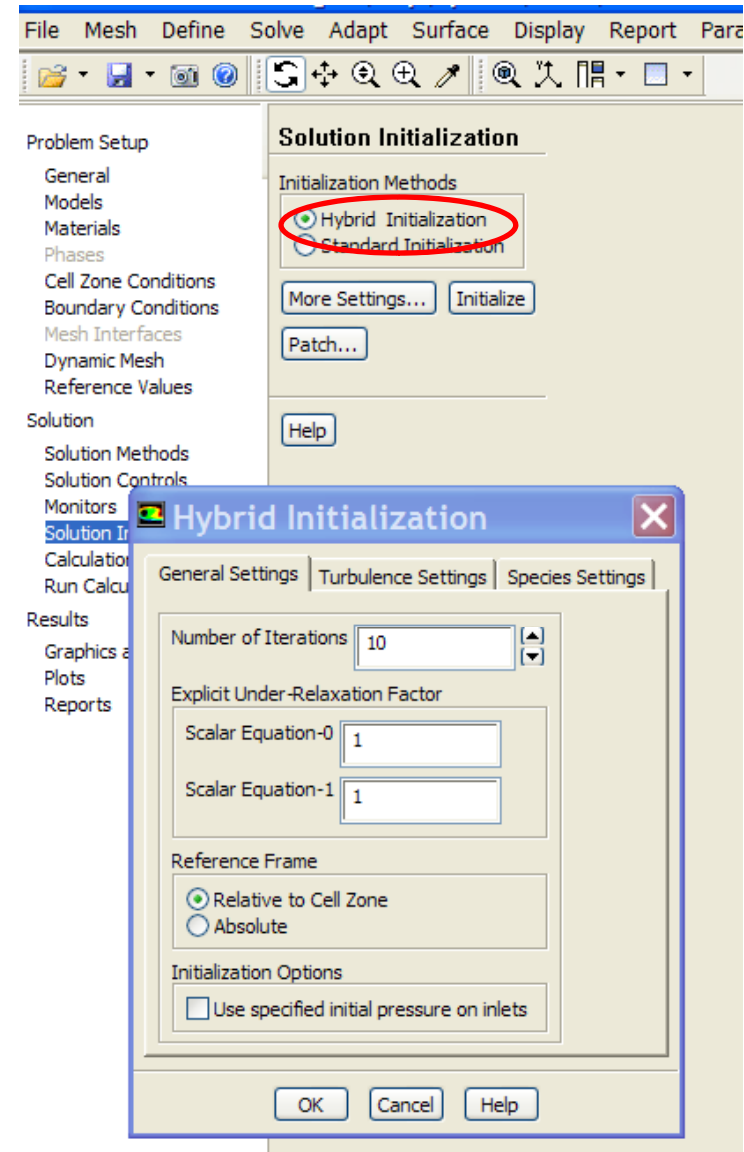


# FMG Initialization

- Full MultiGrid (FMG) Initialization solves the flow problem on a sequence of coarser meshes, before transferring the solution onto the actual mesh
- FMG 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.
  - Euler equations are solved with first-order accuracy on the coarse-level meshes.
  - It can be used with both pressure and density based solvers, but only in steady mode.
- To enable FMG initialization:
  - Pressure-based solver: `/solve/init/fmg-initialization`
  - Density-based solver: Enabled in the GUI if using density-based solver.

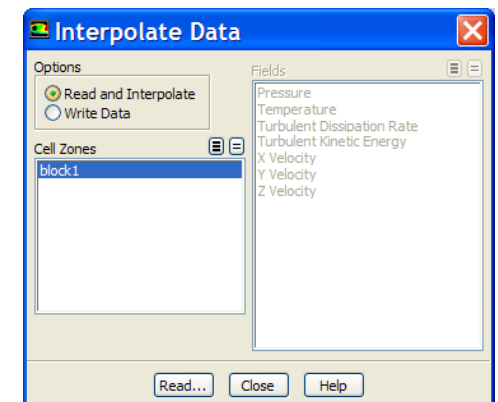
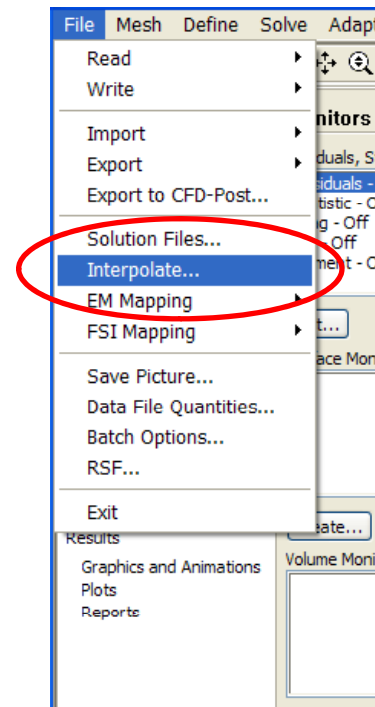
# Hybrid Initialization

- A new feature at R13 is **Hybrid Initialization**.
- This can be selected in the GUI from the Initialization menu.
- 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 fractions, volume fractions, etc., will be automatically patched based on domain averaged values or a particular interpolation method.

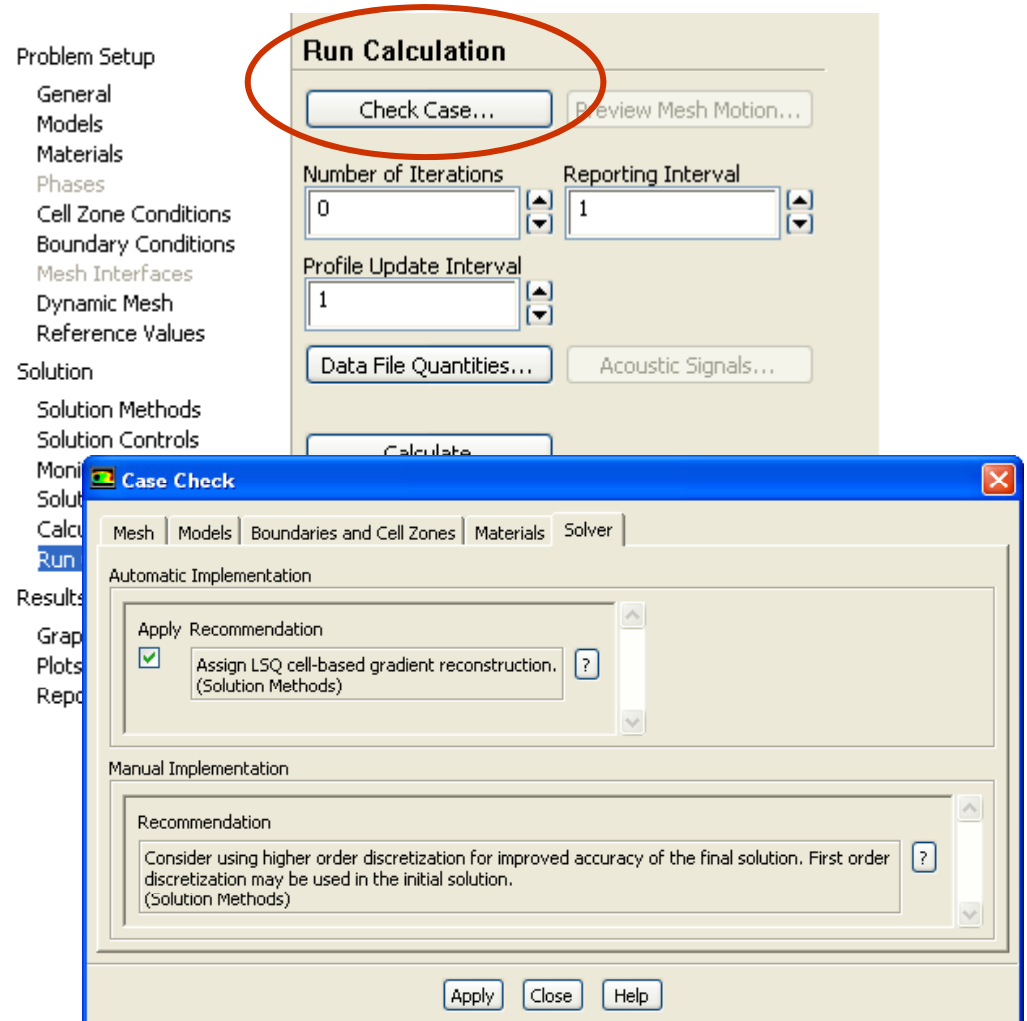


# Starting from a Previous Solution

- Convergence rates are dependent on how good the starting point is.
- Therefore if you already have a similar result from another simulation, you can save time by interpolating that result into the new simulation.
- First use the 'Write' option on the old model to save the required values to disk.
- Then use the 'Read and Interpolate' option on the new model.



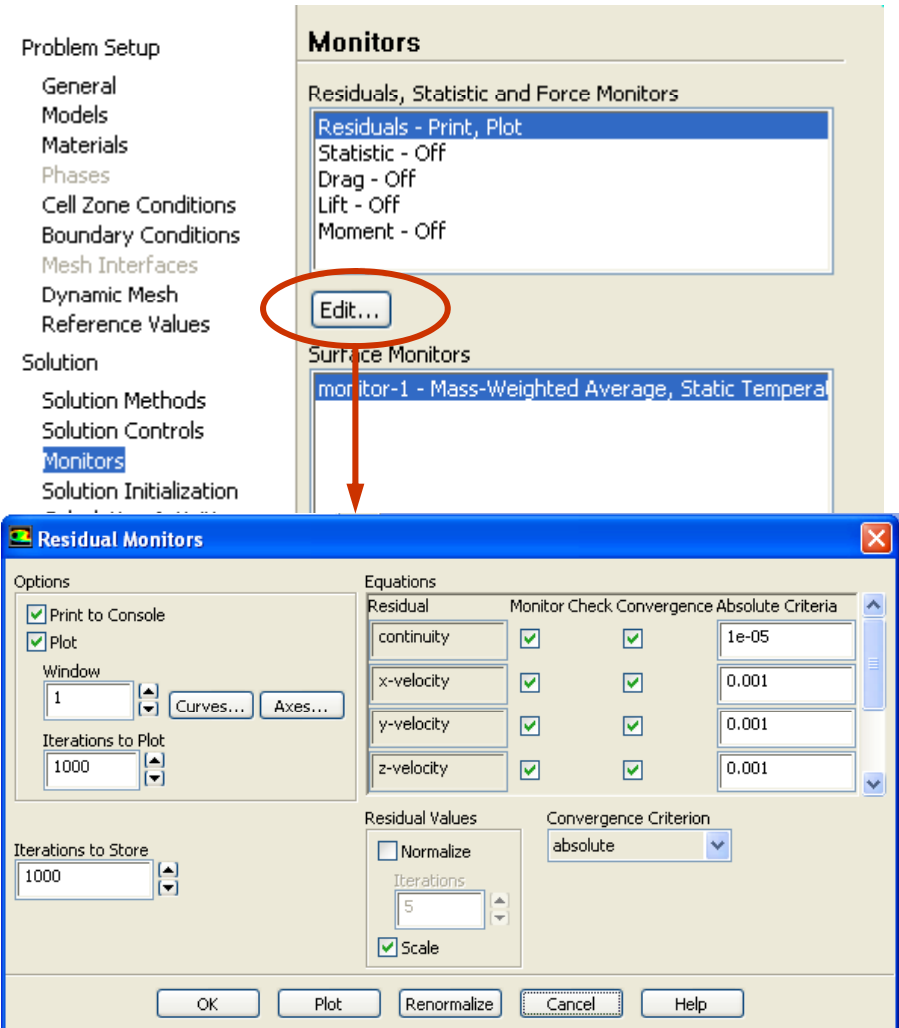
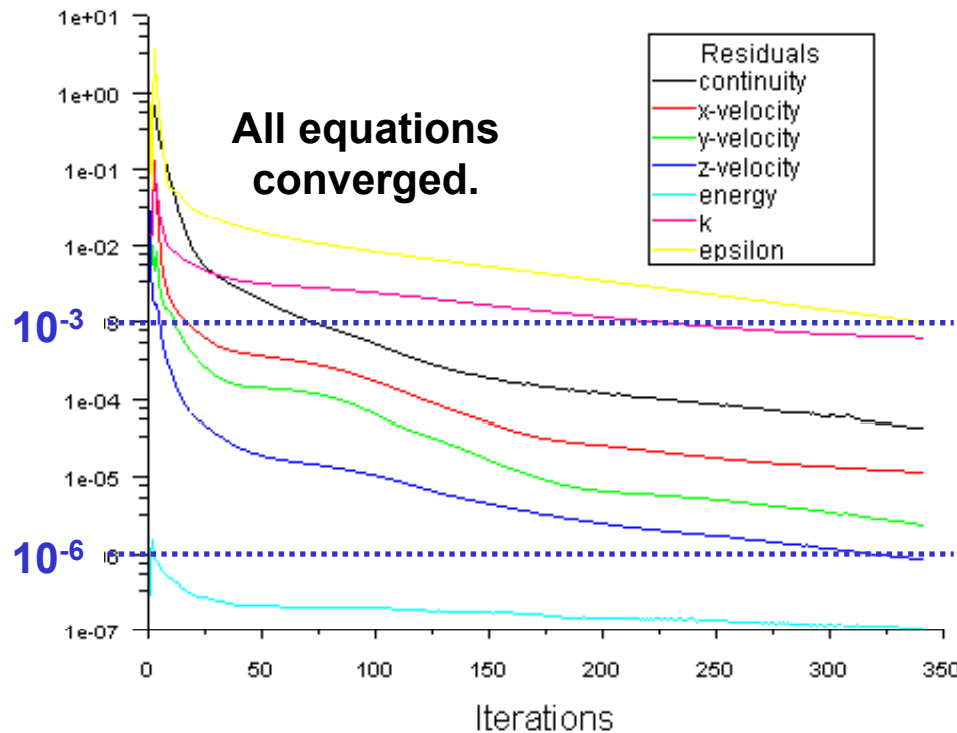
- **Case Check** is a utility in FLUENT which searches for common setup errors and inconsistencies.
  - Provides guidance in selecting case parameters and models.
- Tabbed sections contain recommendations which the user can optionally apply or ignore.



- The solver should be given sufficient iterations such that the problem is **converged**
- At convergence, the following should be satisfied:
  - The solution no longer changes with subsequent iterations.
  - Overall mass, momentum, energy, and scalar balances are achieved.
  - All equations (momentum, energy, etc.) are obeyed in all cells **to a specified tolerance**
- Monitoring convergence using residual history:
  - Generally, a decrease in residuals by **three orders of magnitude** indicates at least qualitative convergence. At this point, the major flow features should be established.
  - 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.

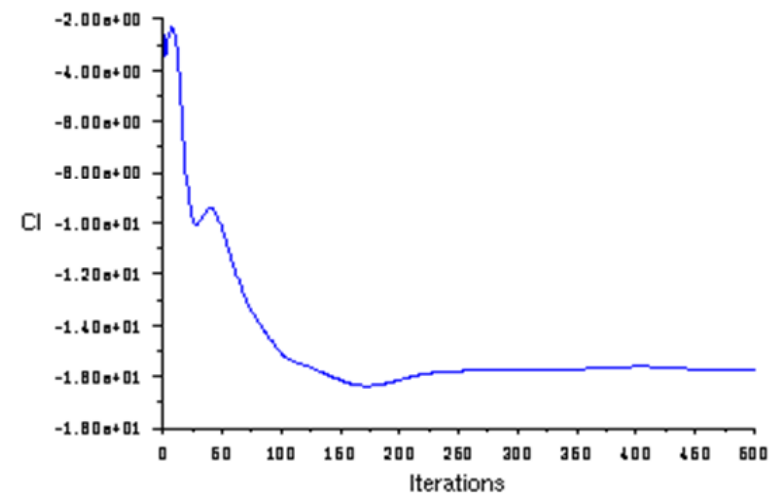
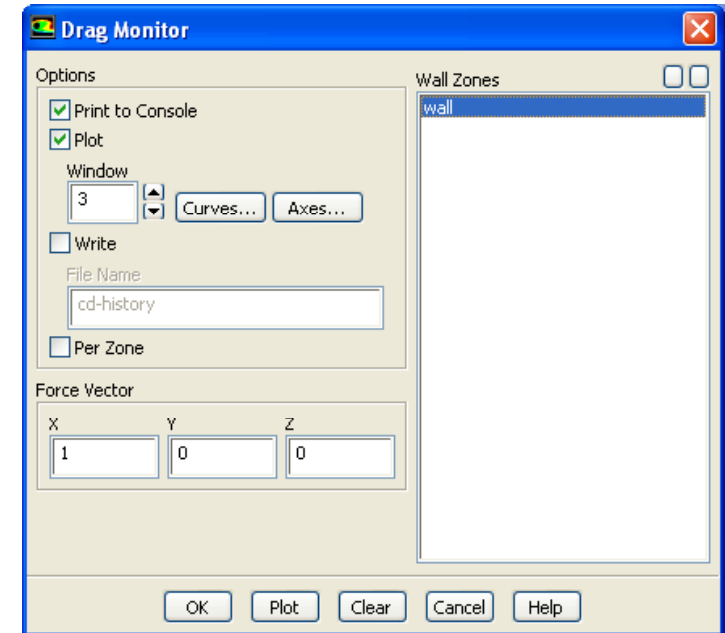
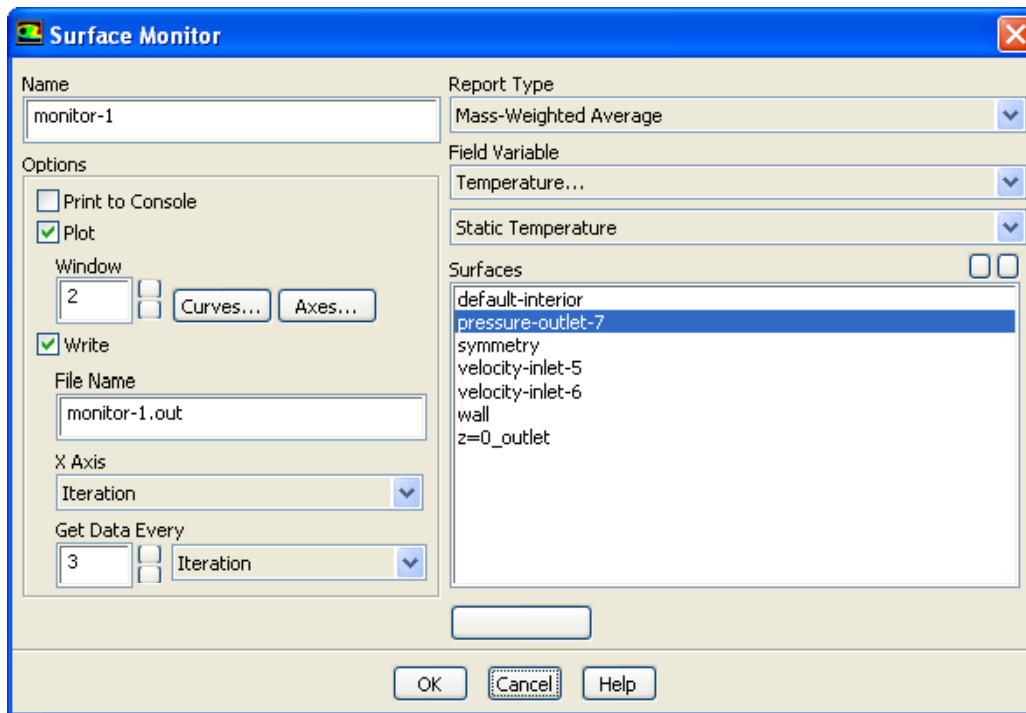
## Convergence Monitors – Residuals

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



# Convergence Monitors – Forces and Surfaces

- If there is a particular value you are interested in (lift coefficient, average surface temperature etc), it is useful to plot how that value is converging.



# Checking Overall Flux Conservation

- Another important metric to assess whether the model is converged is to check the overall heat and mass balance.
- The net flux imbalance (shown in the GUI as Net Results) should be less than 1% of the smallest flux through the domain boundary

The screenshot displays the ANSYS Solver Settings interface. On the left, the 'Problem Setup' tree shows 'Reports' selected under 'Results'. The 'Reports' panel on the right lists various report types, with 'Fluxes' highlighted. Below this panel are buttons for 'Set Up...', 'Parameters...', and 'Help'.

The 'Flux Reports' dialog box is open, showing the following settings:

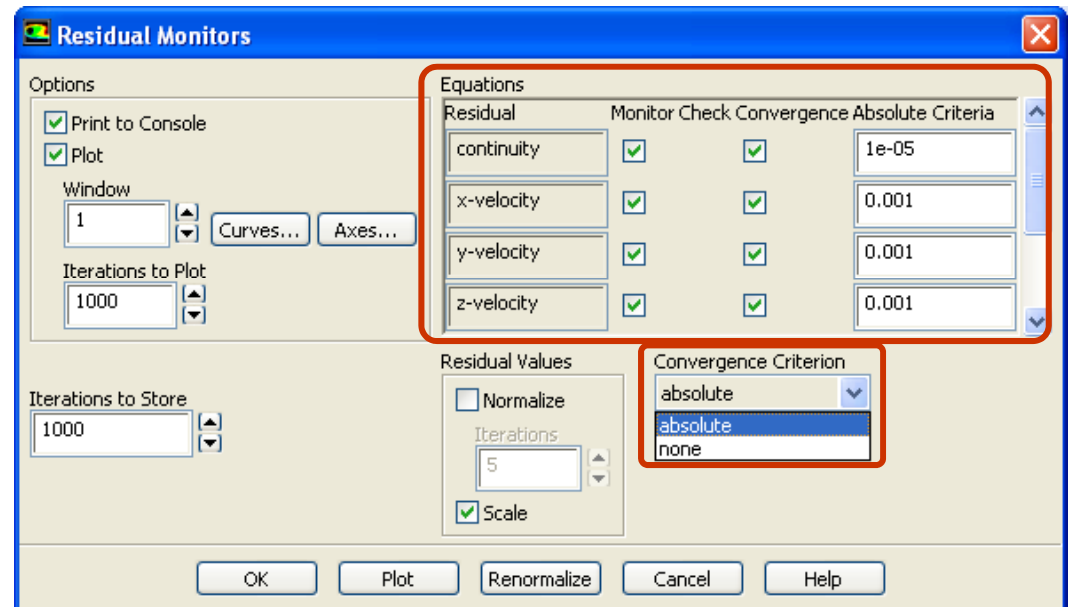
- Options:** Mass Flow Rate (selected), Total Heat Transfer Rate, Radiation Heat Transfer Rate.
- Boundary Types:** axis, exhaust-fan, fan, inlet-vent.
- Boundary Name Pattern:** (empty field) and Match button.
- Save Output Parameter...** button.
- Boundaries:** default-interior, pressure-outlet-7, symmetry, velocity-inlet-5, velocity-inlet-6, wall.
- Results:**

| Boundary          | Value      |
|-------------------|------------|
| default-interior  | -1.9195043 |
| pressure-outlet-7 | 1.6175208  |
| symmetry          | 0.30180666 |
| velocity-inlet-5  |            |
| velocity-inlet-6  |            |
| wall              |            |
- Net Results (kg/s):** -0.0001768172 (circled in red).
- Buttons:** Compute (circled in red), Write..., Close, Help.

# 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 Criterion or disable Check Convergence in the Residual Monitors panel.
  - Continue iterations until the solution converges.

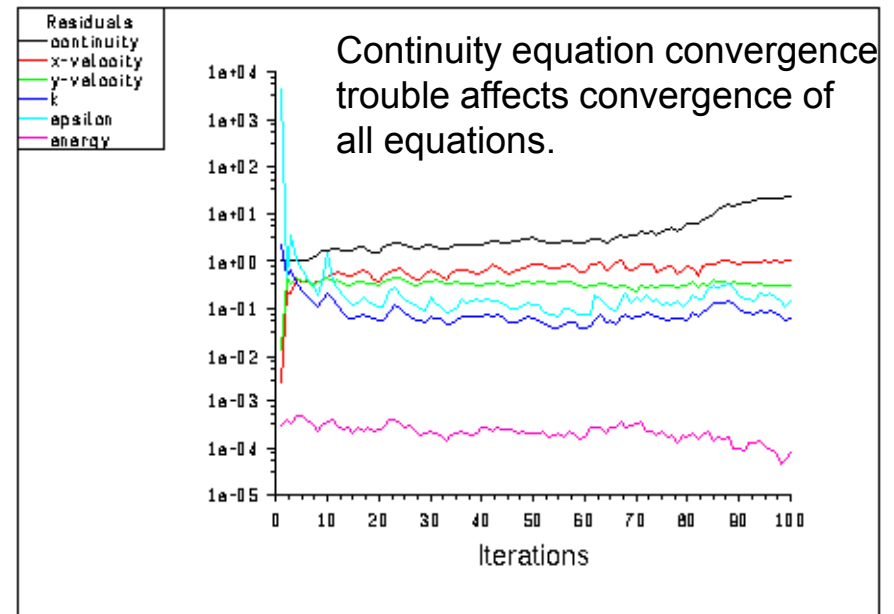
- Selecting **None** under Convergence Criterion disables convergence checking for all equations.



# Convergence Difficulties

- Sometimes running for further iterations is not the answer:
  - Either the solution is diverging (aka “blowing up”)
  - Or the residuals are ‘stuck’ with a large imbalance still remaining.

- Troubleshooting
  - Ensure that the problem is well-posed.
  - Compute an initial solution using a first-order discretisation scheme.
  - Alter the under-relaxation or Courant numbers (see following slides)



- Check the mesh quality. It can only take one very skewed grid cell to prevent the entire solution converging [This is why you should ALWAYS check the mesh quality before spending time with the solver]

# Modifying Under-Relaxation Factors

- Under-relaxation factor,  $\alpha$ , is included to stabilize the iterative process for the pressure-based solver.
- Use default under-relaxation factors to start a calculation.
- If value is too high, the model will be unstable, and may fail to converge
- If value is much too low, it will take longer (more iterations) to converge.
  - Default settings are suitable for a wide range of problems, you can reduce the values when necessary.
  - Appropriate settings are best learned from experience!

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

### Problem Setup

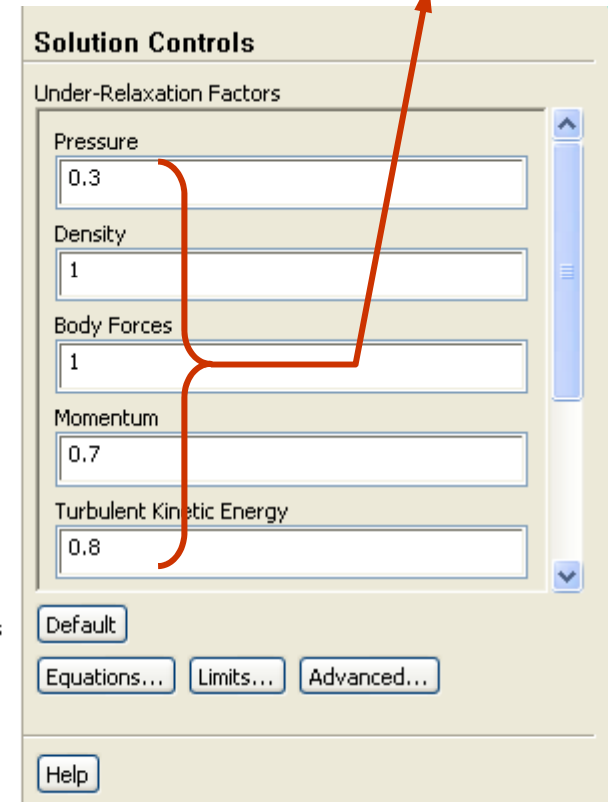
General  
Models  
Materials  
Phases  
Cell Zone Conditions  
Boundary Conditions  
Mesh Interfaces  
Dynamic Mesh  
Reference Values

### Solution

Solution Methods  
**Solution Controls**  
Monitors  
Solution Initialization  
Calculation Activities  
Run Calculation

### Results

Graphics and Animations  
Plots  
Reports



**Solution Controls**

Under-Relaxation Factors

Pressure: 0.3

Density: 1

Body Forces: 1

Momentum: 0.7

Turbulent Kinetic Energy: 0.8

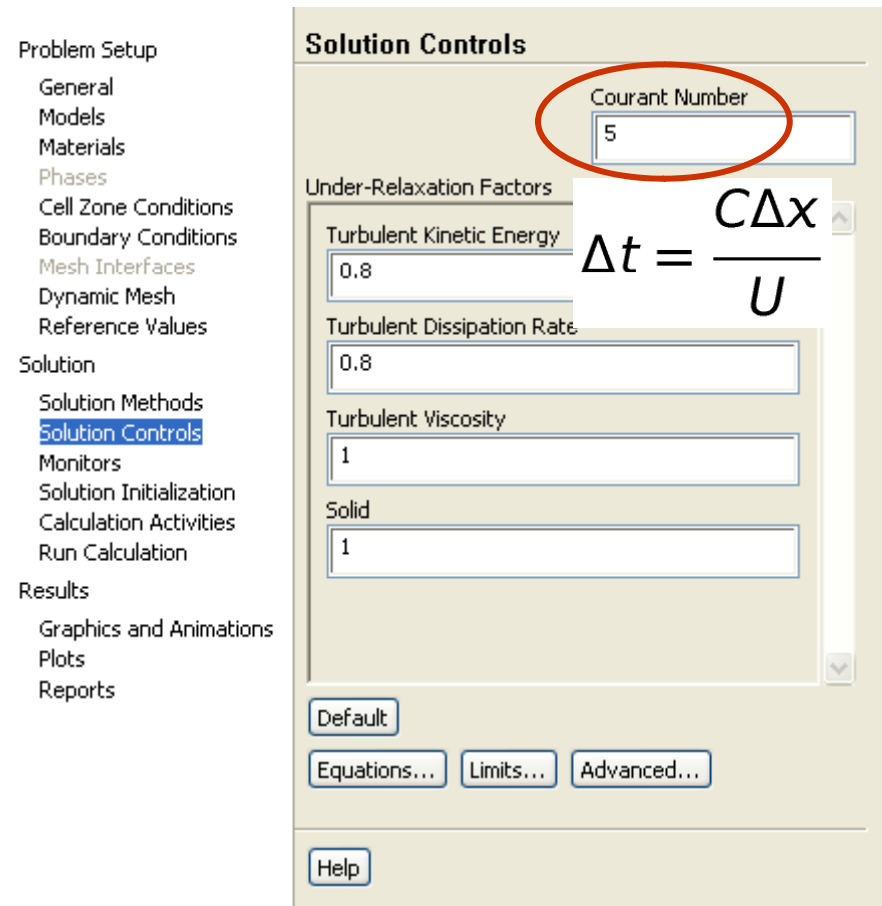
Default

Equations... Limits... Advanced...

Help

# Modifying the Courant Number

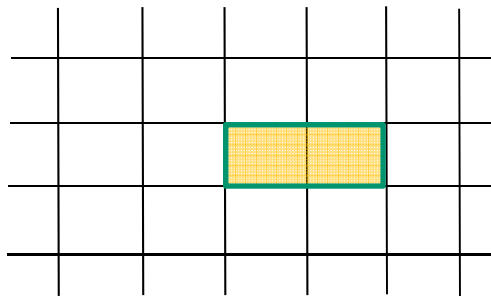
- The **Courant number** is the main control for stability when using the **coupled solvers**.
- A transient term is included in the density-based solver even for steady state problems.
  - The Courant number defines the time step size.
- For density-based explicit solver:
  - Stability constraints impose a maximum limit on the Courant number.
    - Cannot be greater than 2 (default value is 1).
- For density-based implicit solver:
  - The Courant number is not limited by stability constraints.
    - Default value is 5.



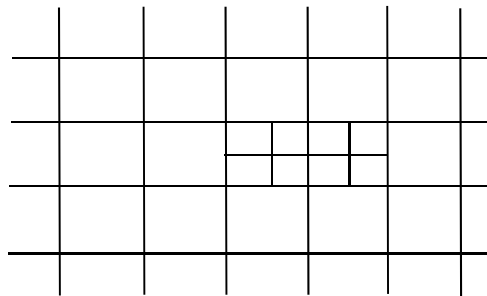
- **Remember, a converged solution is not necessarily a correct one!**
  - Always inspect and evaluate the solution by using available data, physical principles and so on.
  - Use the second-order upwind discretisation scheme for final results.
  - Ensure that solution is grid-independent (see next slide)
- If flow features **do not** seem reasonable:
  - Reconsider physical models and boundary conditions
  - Examine mesh quality and possibly remesh 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

- It is important to verify that the mesh used was **fit-for-purpose**.
  - Even if the grid metrics like skewness are showing the mesh is of a good quality, there may still be too few grid cells to properly resolve the flow.
- To trust a result, **it must be grid-independent**. In other words, if the mesh is refined further, the solution does not change.
- Typically you will perform this test once for your class of problem.
- Either:
  - Go back to the meshing tool and modify the settings to give a finer mesh.
  - Or use the 'Adaption' tools in FLUENT to refine the mesh you already have.  
*Make sure you save the model first*
  - Run on your simulation (remember you can start from your past result) and assess whether the grid refinement has changed the result.

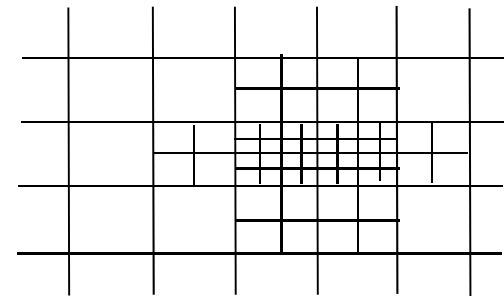
- Although a mesh must be refined where flow features change rapidly, sometimes the location cannot be determined during the initial meshing.
- Examples include:
  - Refinement around shock waves
  - Refinement around free-surface boundaries.
- FLUENT can refine localised regions of the mesh, selected by:
  - Regions where parameters (eg density) change rapidly
  - Regions selected manually by the user drawing a bounding-box.



Original mesh, with  
highlighted cells to be  
adapted



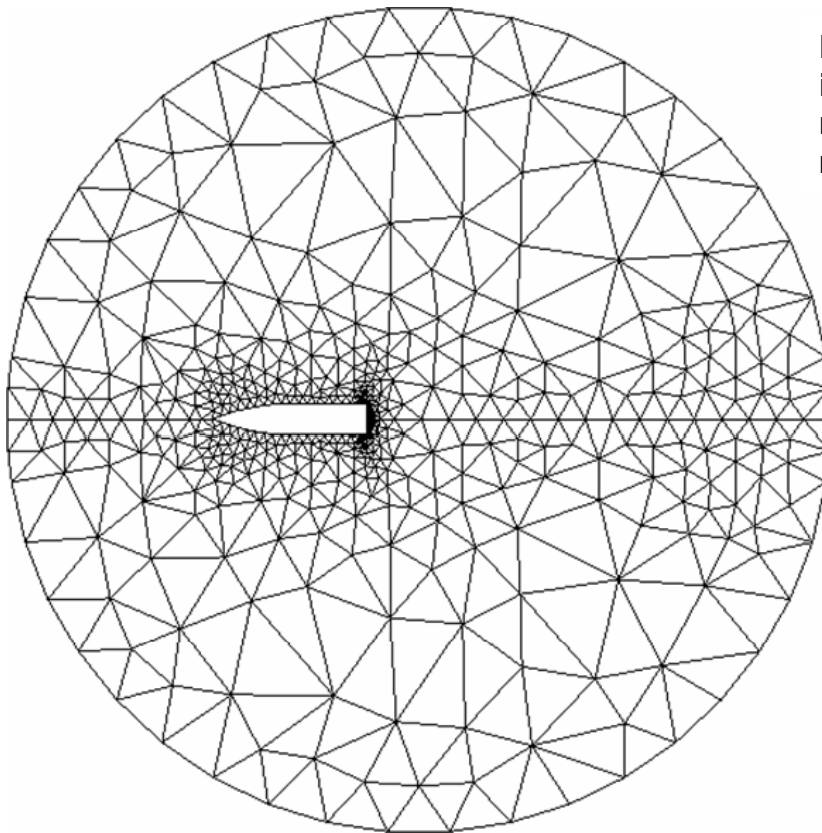
Result from 1 level of  
adaption on marked  
cells



Result from 2 levels of  
adaption on originally  
marked cells

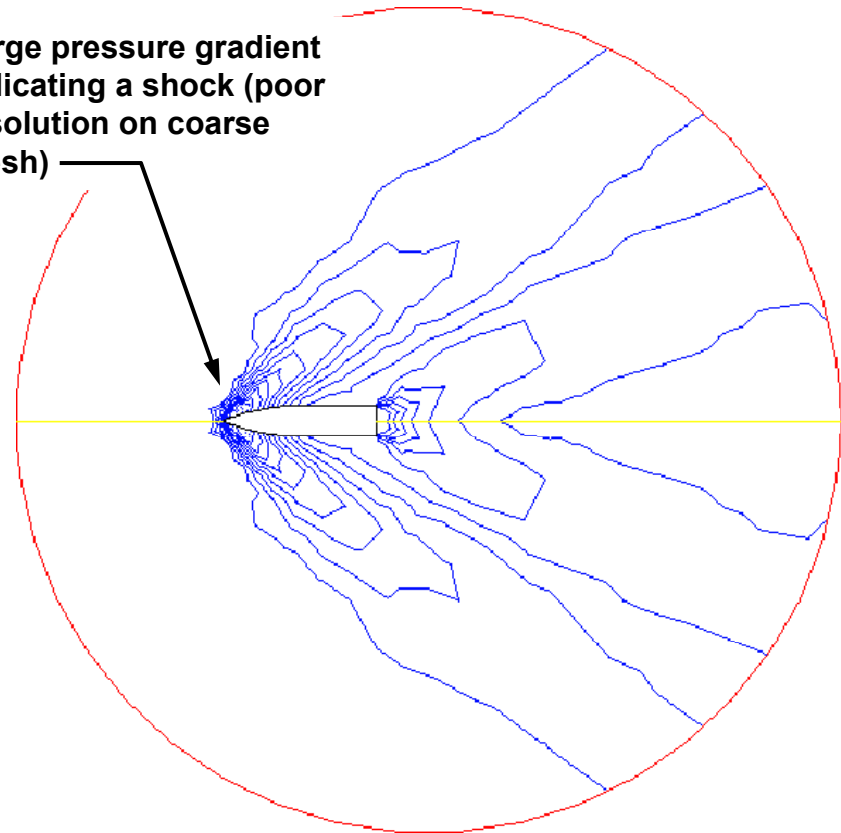
## Mesh Adaption – Supersonic Projectile [1]

- Example: The location of the shock wave is not known when the mesh is first created



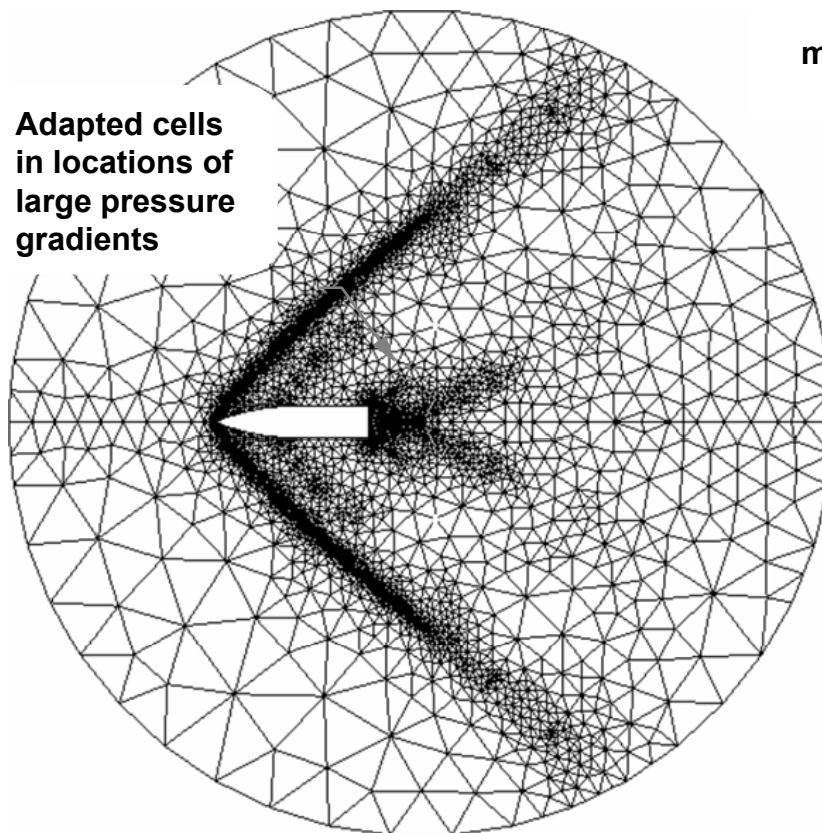
**Initial Mesh (Generated by Preprocessor)**

Large pressure gradient  
indicating a shock (poor  
resolution on coarse  
mesh)

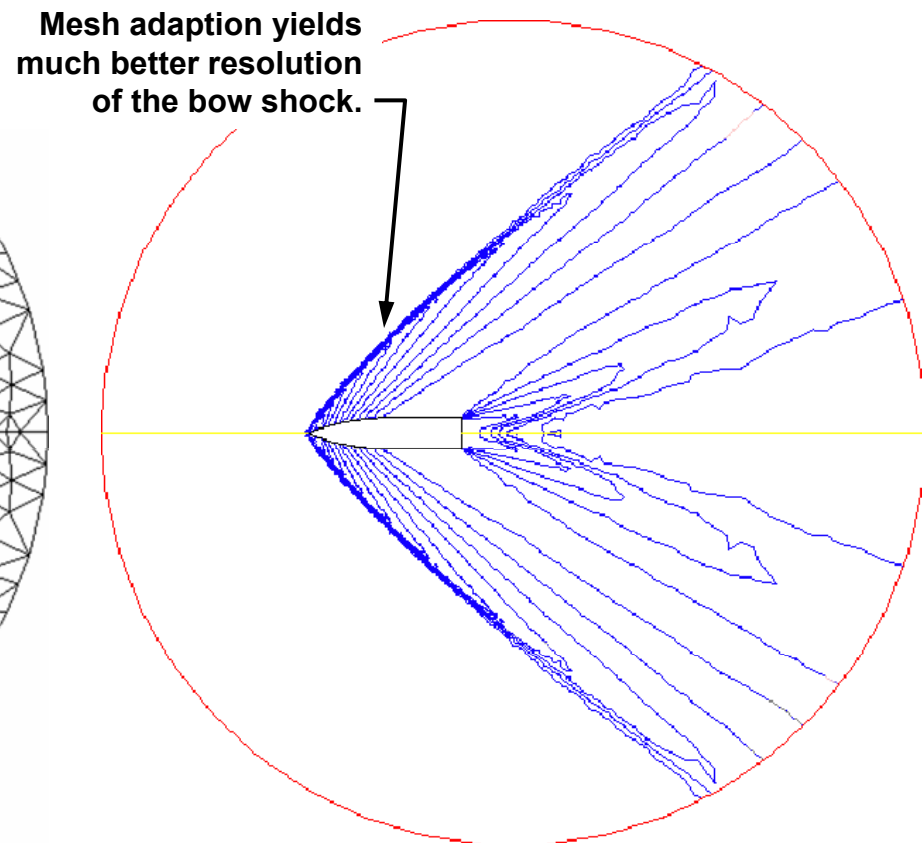


**Pressure Contours on Initial Mesh**

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



**Adapted Mesh (Multiple Adaptions Based on Gradients of Pressure)**

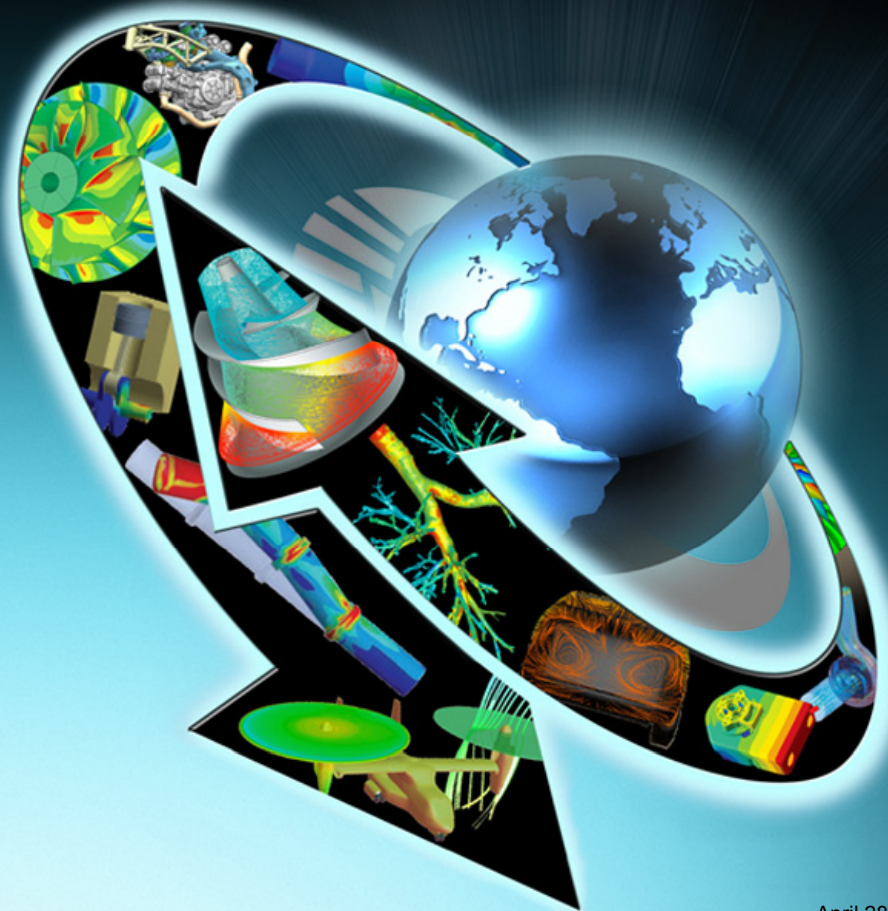


**Pressure Contours on Adapted Mesh**

- Make sure your final results are computed with the optimal numerical schemes (*the FLUENT defaults aim to give a stable solution, not necessarily the most accurate one*).
- All solvers provide tools for judging and improving convergence and ensuring stability.
- All solvers provide tools for checking and improving accuracy.
- Solution procedure for both the pressure-based and density-based solvers is identical.
  - Calculate until you get a converged solution
  - Obtain a second-order solution (recommended)
  - Refine the mesh and recalculate to verify grid-independence of the result
- Solution accuracy will depend on the appropriateness of the physical models that you choose and the boundary conditions that you specify.



## Appendix : Additional notes



- **Mesh adaption** refers to refinement and/or coarsening cells where needed to resolve the flow field without returning to the preprocessor.
  - Mark cells satisfying the adaption criteria and store them in a “register.”
  - Display and modify the register if desired.
  - Click Adapt to adapt the cells listed in the register.
- Registers can be defined based on:
  - Gradients or isovalues 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

