Chapter 22. Using the Solver

This chapter describes the FLUENT solver and how to use it. Details about the solver algorithms used by FLUENT are provided in Sections 22.1–22.5. Section 22.6 provides an overview of how to use the solver, and the remaining sections provide detailed instructions.

- Section 22.17: Animating the Solution
- Section 22.18: Executing Commands During the Calculation
- Section 22.19: Convergence and Stability

22.1 Overview of Numerical Schemes

FLUENT allows you to choose either of two numerical methods:

- segregated solver
- coupled solver

Using either method, FLUENT will solve the governing integral equations for the conservation of mass and momentum, and (when appropriate) for energy and other scalars such as turbulence and chemical species. In both cases a control-volume-based technique is used that consists of:

- Division of the domain into discrete control volumes using a computational grid.
- Integration of the governing equations on the individual control volumes to construct algebraic equations for the discrete dependent variables ("unknowns") such as velocities, pressure, temperature, and conserved scalars.
- Linearization of the discretized equations and solution of the resultant linear equation system to yield updated values of the dependent variables.

The two numerical methods employ a similar discretization process (finite-volume), but the approach used to linearize and solve the discretized equations is different.

The general solution methods are described first in Sections 22.1.1 and 22.1.2, followed by a discussion of the linearization methods in Section 22.1.3.

22.1.1 Segregated Solution Method

The segregated solver is the solution algorithm previously used by FLUENT 4. Using this approach, the governing equations are solved sequentially (i.e., segregated from one another). Because the governing equations are non-linear (and coupled), several iterations of the solution loop must be performed before a converged solution is obtained. Each iteration consists of the steps illustrated in Figure 22.1.1 and outlined below:

1. Fluid properties are updated, based on the current solution. (If the calculation has just begun, the fluid properties will be updated based on the initialized solution.)
2. The u, v, and w momentum equations are each solved in turn using current values for pressure and face mass fluxes, in order to update the velocity field.
3. Since the velocities obtained in Step 2 may not satisfy the continuity equation locally, a "Poisson-type" equation for the pressure correction is derived from the continuity equation and the linearized momentum equations. This pressure correction equation is then solved to obtain the necessary corrections to the pressure and velocity fields and the face mass fluxes such that continuity is satisfied.
4. Where appropriate, equations for scalars such as turbulence, energy, species, and radiation are solved using the previously updated values of the other variables.
5. When interphase coupling is to be included, the source terms in the appropriate continuous phase equations may be updated with a discrete phase trajectory calculation.
6. A check for convergence of the equation set is made.

These steps are continued until the convergence criteria are met.

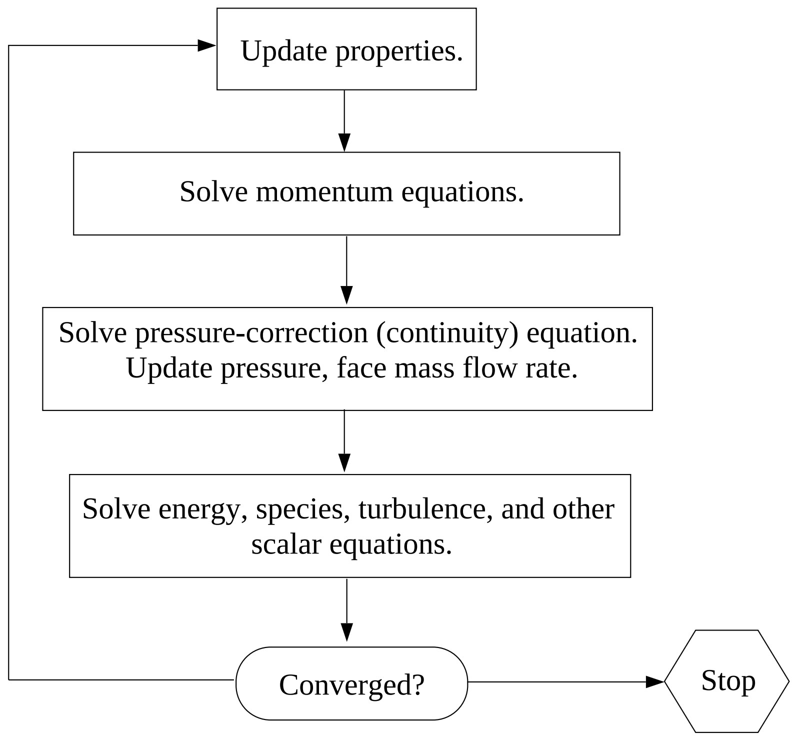

Update properties.
Solve momentum equations.
Solve pressure-correction (continuity) equation.
Update pressure, face mass flow rate.
Solve energy, species, turbulence, and other scalar equations.
Converged?
Stop

Figure 22.1.1: Overview of the Segregated Solution Method

22.1.2 Coupled Solution Method

The coupled solver solves the governing equations of continuity, momentum, and (where appropriate) energy and species transport simultaneously (i.e., coupled together). Governing equations for additional scalars will be solved sequentially (i.e., segregated from one another and from the coupled set) using the procedure described for the segregated solver in Section 22.1.1. Because the governing equations are non-linear (and coupled), several iterations of the solution loop must be performed before a converged solution is obtained. Each iteration consists of the steps illustrated in Figure 22.1.2 and outlined below:

1. Fluid properties are updated, based on the current solution. (If the calculation has just begun, the fluid properties will be updated based on the initialized solution.)
2. The continuity, momentum, and (where appropriate) energy and species equations are solved simultaneously.
3. Where appropriate, equations for scalars such as turbulence and radiation are solved using the previously updated values of the other variables.
4. When interphase coupling is to be included, the source terms in the appropriate continuous phase equations may be updated with a discrete phase trajectory calculation.
5. A check for convergence of the equation set is made.

These steps are continued until the convergence criteria are met.

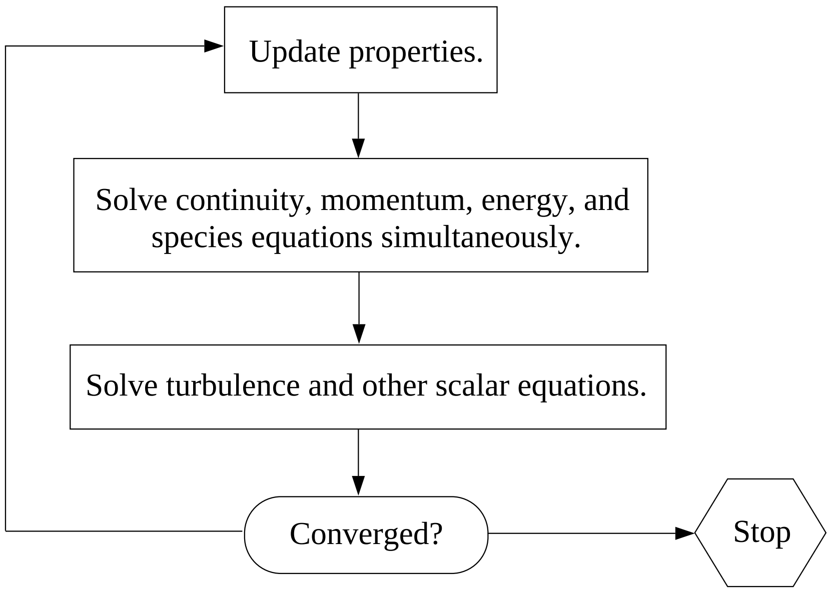

Update properties.
Solve continuity, momentum, energy, and species equations simultaneously.
Solve turbulence and other scalar equations.
Converged?
Stop

Figure 22.1.2: Overview of the Coupled Solution Method

22.1.3 Linearization: Implicit vs. Explicit

In both the segregated and coupled solution methods the discrete, non-linear governing equations are linearized to produce a system of equations for the dependent variables in every computational cell. The resultant linear system is then solved to yield an updated flow-field solution.

The manner in which the governing equations are linearized may take an "implicit" or "explicit" form with respect to the dependent variable (or set of variables) of interest. By implicit or explicit we mean the following:

- implicit: For a given variable, the unknown value in each cell is computed using a relation that includes both existing and unknown values from neighboring cells. Therefore each unknown will appear in more than one equation in the system, and these equations must be solved simultaneously to give the unknown quantities.
- explicit: For a given variable, the unknown value in each cell is computed using a relation that includes only existing values. Therefore each unknown will appear in only one equation in the system and the equations for the unknown value in each cell can be solved one at a time to give the unknown quantities.

In the segregated solution method each discrete governing equation is linearized implicitly with respect to that equation's dependent variable. This will result in a system of linear equations with one equation for each cell in the domain. Because there is only one equation per cell, this is sometimes called a "scalar" system of equations. A point implicit (Gauss-Seidel) linear equation solver is used in conjunction with an algebraic multigrid (AMG) method to solve the resultant scalar system of equations for the dependent variable in each cell. For example, the x-momentum equation is linearized to produce a system of equations in which u velocity is the unknown. Simultaneous solution of this equation system (using the scalar AMG solver) yields an updated u-velocity field.

In summary, the segregated approach solves for a single variable field (e.g., p) by considering all cells at the same time. It then solves for the next variable field by again considering all cells at the same time, and so on. There is no explicit option for the segregated solver.

In the coupled solution method you have a choice of using either an implicit or explicit linearization of the governing equations. This choice applies only to the coupled set of governing equations. Governing equations for additional scalars that are solved segregated from the coupled set, such as for turbulence, radiation, etc., are linearized and solved implicitly using the same procedures as in the segregated solution method. Regardless of whether you choose the implicit or explicit scheme, the solution procedure shown in Figure 22.1.2 is followed.

If you choose the implicit option of the coupled solver, each equation in the coupled set of governing equations is linearized implicitly with respect to all dependent variables in the set. This will result in a system of linear equations with N equations for each cell in the domain, where N is the number of coupled equations in the set. Because there are N equations per cell, this is sometimes called a "block" system of equations. A point implicit (block Gauss-Seidel) linear equation solver is used in conjunction with an algebraic multigrid (AMG) method to solve the resultant block system of equations for all N dependent variables in each cell. For example, linearization of the coupled continuity, x-, y-, z-momentum, and energy equation set will produce a system of equations in which p, u, v, w, and T are the unknowns. Simultaneous solution of this equation system (using the block AMG solver) yields at once updated pressure, u-, v-, w-velocity, and temperature fields.

In summary, the coupled implicit approach solves for all variables (p, u, v, w, T) in all cells at the same time.

If you choose the explicit option of the coupled solver, each equation in the coupled set of governing equations is linearized explicitly. As in the implicit option, this too will result in a system of equations with N equations for each cell in the domain. And likewise, all dependent variables in the set will be updated at once. However, this system of equations is explicit in the unknown dependent variables. For example, the x-momentum equation is written such that the updated x velocity is a function of existing values of the field variables. Because of this, a linear equation solver is not needed. Instead, the solution is updated using a multi-stage (Runge-Kutta) solver. Here you have the additional option of employing a full approximation storage (FAS) multigrid scheme to accelerate the multi-stage solver.

In summary, the coupled explicit approach solves for all variables (p, u, v, w, T) one cell at a time.

Note that the FAS multigrid is an optional component of the explicit approach, while the AMG is a required element in both the segregated and coupled implicit approaches.

22.2 Discretization

FLUENT uses a control-volume-based technique to convert the governing equations to algebraic equations that can be solved numerically. This control volume technique consists of integrating the governing equations about each control volume, yielding discrete equations that conserve each quantity on a control-volume basis.

Discretization of the governing equations can be illustrated most easily by considering the steady-state conservation equation for transport of a scalar quantity ϕ. This is demonstrated by the following equation written in integral form for an arbitrary control volume V as follows:

$$\oint \rho\phi\,\vec{v}\cdot d\vec{A} = \oint \Gamma_\phi\,\nabla\phi\cdot d\vec{A} + \int_V S_\phi\,dV \qquad (22.2\text{-}1)$$

where

ρ	=	density
$\vec{v}$	=	velocity vector ($= u\,\hat{\imath} + v\,\hat{\jmath}$ in 2D)
$\vec{A}$	=	surface area vector
Γ_ϕ	=	diffusion coefficient for ϕ
$\nabla\phi$	=	gradient of ϕ ($= \partial\phi/\partial x)\,\hat{\imath} + (\partial\phi/\partial y)\,\hat{\jmath}$ in 2D)
S_ϕ	=	source of ϕ per unit volume

Equation 22.2-1 is applied to each control volume, or cell, in the computational domain. The two-dimensional, triangular cell shown in Figure 22.2.1 is an example of such a control volume. Discretization of Equation 22.2-1 on a given cell yields

$$\sum_{f}^{N_{\text{faces}}} \rho_f \vec{v}_f \phi_f \cdot \vec{A}_f = \sum_{f}^{N_{\text{faces}}} \Gamma_\phi \, (\nabla \phi)_n \cdot \vec{A}_f + S_\phi V \qquad (22.2\text{-}2)$$

where

N_{faces}	=	number of faces enclosing cell
ϕ_f	=	value of ϕ convected through face f
$\rho_f \vec{v}_f \cdot \vec{A}_f$	=	mass flux through the face
$\vec{A}_f$	=	area of face f, $\|A\|$ $(= \|A_x \hat{\imath} + A_y \hat{\jmath}\|$ in 2D)
$(\nabla \phi)_n$	=	magnitude of $\nabla \phi$ normal to face f
V	=	cell volume

The equations solved by FLUENT take the same general form as the one given above and apply readily to multi-dimensional, unstructured meshes composed of arbitrary polyhedra.

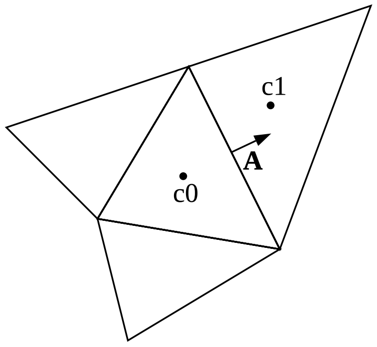


Figure 22.2.1: Control Volume Used to Illustrate Discretization of a Scalar Transport Equation

FLUENT stores discrete values of the scalar ϕ at the cell centers ($c0$ and $c1$ in Figure 22.2.1). However, face values ϕ_f are required for the convection terms in Equation 22.2-2 and must be interpolated from the cell center values. This is accomplished using an upwind scheme.

Upwinding means that the face value ϕ_f is derived from quantities in the cell upstream, or "upwind," relative to the direction of the normal velocity v_n in Equation 22.2-2. FLUENT allows you to choose from several upwind schemes: first-order upwind, second-order upwind, power law, and QUICK. These schemes are described in Sections 22.2.1–22.2.4.

The diffusion terms in Equation 22.2-2 are central-differenced and are always second-order accurate.

22.2.1 First-Order Upwind Scheme

When first-order accuracy is desired, quantities at cell faces are determined by assuming that the cell-center values of any field variable represent a cell-average value and hold throughout the entire cell; the face quantities are identical to the cell quantities. Thus when first-order upwinding is selected, the face value ϕ_f is set equal to the cell-center value of ϕ in the upstream cell

22.2.2 Power-Law Scheme

The power-law discretization scheme interpolates the face value of a variable, ϕ, using the exact solution to a one-dimensional convection-diffusion equation

$$\frac{\partial}{\partial x}(\rho u \phi) = \frac{\partial}{\partial x}\Gamma\frac{\partial \phi}{\partial x} \tag{22.2-3}$$

where Γ and ρu are constant across the interval ∂x. Equation 22.2-3 can be integrated to yield the following solution describing how ϕ varies with x:

$$\frac{\phi(x) - \phi_0}{\phi_L - \phi_0} = \frac{\exp(\mathrm{Pe}\frac{x}{L}) - 1}{\exp(\mathrm{Pe}) - 1} \tag{22.2-4}$$

where

$$\begin{aligned}\phi_0 &= \phi|_{x=0}\\ \phi_L &= \phi|_{x=L}\end{aligned}$$

and Pe is the Peclet number:

$$\mathrm{Pe} = \frac{\rho u L}{\Gamma} \tag{22.2-5}$$

The variation of $\phi(x)$ between $x = 0$ and $x = L$ is depicted in Figure 22.2.2 for a range of values of the Peclet number. Figure 22.2.2 shows that for large Pe, the value of ϕ at $x = L/2$ is approximately equal to the upstream value. This implies that when the flow is dominated by convection, interpolation can be accomplished by simply letting the face value of a variable be set equal to its "upwind" or upstream value. This is the standard first-order scheme for FLUENT.

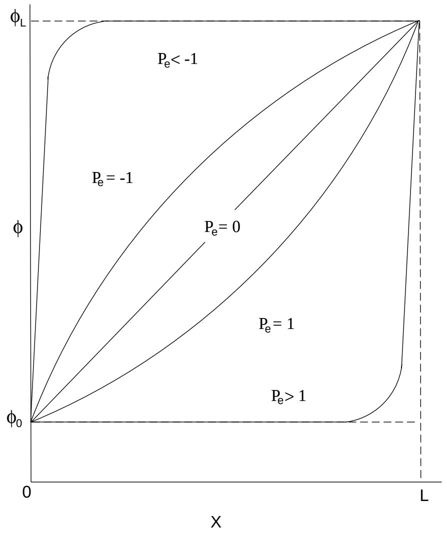


Figure 22.2.2: Variation of a Variable ϕ Between $x = 0$ and $x = L$ (Equation 22.2-3)

If the power-law scheme is selected, FLUENT uses Equation 22.2-4 in an equivalent "power law" format [172], as its interpolation scheme.

As discussed in Section 22.2.1, Figure 22.2.2 shows that for large Pe, the value of ϕ at $x = L/2$ is approximately equal to the upstream value. When Pe=0 (no flow, or pure diffusion), Figure 22.2.2 shows that ϕ may be interpolated using a simple linear average between the values at $x = 0$ and $x = L$. When the Peclet number has an intermediate value, the interpolated value for ϕ at $x = L/2$ must be derived by applying the "power law" equivalent of Equation 22.2-4.

22.2.3 Second-Order Upwind Scheme

When second-order accuracy is desired, quantities at cell faces are computed using a multidimensional linear reconstruction approach [7]. In this approach, higher-order accuracy is achieved at cell faces through a Taylor series expansion of the cell-centered solution about the cell centroid. Thus when second-order upwinding is selected, the face value ϕ_f is computed using the following expression:

$$\phi_f = \phi + \nabla\phi \cdot \Delta\vec{s} \tag{22.2-6}$$

where ϕ and $\nabla\phi$ are the cell-centered value and its gradient in the upstream cell, and $\Delta\vec{s}$ is the displacement vector from the upstream cell centroid to the face centroid. This formulation requires the determination of the gradient $\nabla\phi$ in each cell. This gradient is computed using the divergence theorem, which in discrete form is written as

$$\nabla\phi = \frac{1}{V}\sum_{f}^{N_{\rm faces}} \tilde{\phi}_f \, \vec{A} \tag{22.2-7}$$

Here the face values $\tilde{\phi}_f$ are computed by averaging ϕ from the two cells adjacent to the face. Finally, the gradient $\nabla\phi$ is limited so that no new maxima or minima are introduced.

22.2.4 QUICK Scheme

For quadrilateral and hexahedral meshes, where unique upstream and downstream faces and cells can be identified, FLUENT also provides the QUICK scheme for computing a higher-order value of the convected variable ϕ at a face. QUICK-type schemes [133] are based on a weighted average of second-order-upwind and central interpolations of the variable. For the face e in Figure 22.2.3, if the flow is from left to right, such a value can be written as

$$\phi_e = \theta \left[\frac{S_d}{S_c + S_d} \phi_P + \frac{S_c}{S_c + S_d} \phi_E \right] + (1 - \theta) \left[\frac{S_u + 2S_c}{S_u + S_c} \phi_P - \frac{S_c}{S_u + S_c} \phi_W \right] \tag{22.2-8}$$

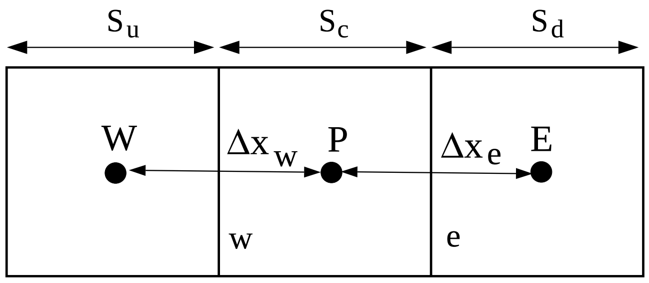


Figure 22.2.3: One-Dimensional Control Volume

$\theta = 1$ in the above equation results in a central second-order interpolation while $\theta = 0$ yields a second-order upwind value. The traditional QUICK scheme is obtained by setting $\theta = 1/8$. The implementation in FLUENT uses a variable, solution-dependent value of θ, chosen so as to avoid introducing new solution extrema.

The QUICK scheme will typically be more accurate on structured grids aligned with the flow direction. Note that FLUENT allows the use of the QUICK scheme for unstructured or hybrid grids as well; in such cases the usual second-order upwind discretization scheme (described in Section 22.2.3) will be used at the faces of non-hexahedral (or non-quadrilateral, in 2D) cells. The second-order upwind scheme will also be used at partition boundaries when the parallel solver is used.

22.2.5 Central-Differencing Scheme

A second-order-accurate central-differencing discretization scheme is available for the momentum equations when you are using the LES turbulence model. This scheme provides improved accuracy for LES calculations.

The central-differencing scheme calculates the face value for a variable (ϕ_f) as follows:

$$\phi_{f,\mathrm{CD}} = \frac{1}{2}(\phi_0 + \phi_1) + \frac{1}{2}(\nabla\phi_{r,0} \cdot \vec{r}_0 + \nabla\phi_{r,1} \cdot \vec{r}_1) \tag{22.2-9}$$

where the indices 0 and 1 refer to the cells that share face f, $\nabla\phi_{r,0}$ and $\nabla\phi_{r,1}$ are the reconstructed gradients at cells 0 and 1, respectively, and $\vec{r}$ is the vector directed from the cell centroid toward the face centroid.

It is well known that central-differencing schemes can produce unbounded solutions and non-physical wiggles, which can lead to stability problems for the numerical procedure. These stability problems can often be avoided if a deferred approach is used for the central-differencing scheme. In this approach, the face value is calculated as follows:

$$\phi_f = \underbrace{\phi_{f,\mathrm{UP}}}_{\text{implicit part}} + \underbrace{(\phi_{f,\mathrm{CD}} - \phi_{f,\mathrm{UP}})}_{\text{explicit part}} \tag{22.2-10}$$

where UP stands for upwind. As indicated, the upwind part is treated implicitly while the difference between the central-difference and upwind values is treated explicitly. Provided that the numerical solution converges, this approach leads to pure second-order differencing.

22.2.6 Linearized Form of the Discrete Equation

The discretized scalar transport equation (Equation 22.2-2) contains the unknown scalar variable ϕ at the cell center as well as the unknown values in surrounding neighbor cells. This equation will, in general, be non-linear with respect to these variables. A linearized form of Equation 22.2-2 can be written as

$$a_P\,\phi = \sum_{\mathrm{nb}} a_{\mathrm{nb}}\phi_{\mathrm{nb}} + b \tag{22.2-11}$$

where the subscript nb refers to neighbor cells, and a_P and a_{nb} are the linearized coefficients for ϕ and ϕ_{nb}.

The number of neighbors for each cell depends on the grid topology, but will typically equal the number of faces enclosing the cell (boundary cells being the exception).

Similar equations can be written for each cell in the grid. This results in a set of algebraic equations with a sparse coefficient matrix. For scalar equations, FLUENT solves this linear system using a point implicit (Gauss-Seidel) linear equation solver in conjunction with an algebraic multigrid (AMG) method which is described in Section 22.5.3.

22.2.7 Under-Relaxation

Because of the nonlinearity of the equation set being solved by FLUENT, it is necessary to control the change of ϕ. This is typically achieved by under-relaxation, which reduces the change of ϕ produced during each iteration. In a simple form, the new value of the variable ϕ within a cell depends upon the old value, $\phi_{\rm old}$, the computed change in ϕ, $\Delta\phi$, and the under-relaxation factor, α, as follows:

$$\phi = \phi_{\rm old} + \alpha\Delta\phi \tag{22.2-12}$$

22.2.8 Temporal Discretization

For transient simulations, the governing equations must be discretized in both space and time. The spatial discretization for the time-dependent equations is identical to the steady-state case. Temporal discretization involves the integration of every term in the differential equations over a time step Δt. The integration of the transient terms is straightforward, as shown below.

A generic expression for the time evolution of a variable ϕ is given by

$$\frac{\partial\phi}{\partial t} = F(\phi) \tag{22.2-13}$$

where the function F incorporates any spatial discretization. If the time derivative is discretized using backward differences, the first-order accurate temporal discretization is given by

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} = F(\phi) \tag{22.2-14}$$

and the second-order discretization is given by

$$\frac{3\phi^{n+1} - 4\phi^n + \phi^{n-1}}{2\Delta t} = F(\phi) \tag{22.2-15}$$

where

ϕ	=	a scalar quantity
$n+1$	=	value at the next time level, $t + \Delta t$
n	=	value at the current time level, t
$n-1$	=	value at the previous time level, $t - \Delta t$

Once the time derivative has been discretized, a choice remains for evaluating $F(\phi)$: in particular, which time level values of ϕ should be used in evaluating F?

Implicit Time Integration

One method is to evaluate $F(\phi)$ at the future time level:

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} = F(\phi^{n+1}) \tag{22.2-16}$$

This is referred to as "implicit" integration since ϕ^{n+1} in a given cell is related to ϕ^{n+1} in neighboring cells through $F(\phi^{n+1})$:

$$\phi^{n+1} = \phi^n + \Delta t F(\phi^{n+1}) \tag{22.2-17}$$

This implicit equation can be solved iteratively by initializing ϕ^i to ϕ^n and iterating the equation

$$\phi^i = \phi^n + \Delta t F(\phi^i) \tag{22.2-18}$$

for the first-order implicit formulation, or

$$\phi^i = 4/3\phi^n - 1/3\phi^{n-1} + 2/3\Delta t F(\phi^i) \qquad (22.2\text{-}19)$$

for the second-order implicit formulation, until ϕ^i stops changing (i.e., converges). At that point, ϕ^{n+1} is set to ϕ^i.

The advantage of the fully implicit scheme is that it is unconditionally stable with respect to time step size.

Explicit Time Integration

A second method is available when the coupled explicit solver is used. This method evaluates $F(\phi)$ at the current time level:

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} = F(\phi^n) \qquad (22.2\text{-}20)$$

and is referred to as "explicit" integration since ϕ^{n+1} can be expressed explicitly in terms of the existing solution values, ϕ^n:

$$\phi^{n+1} = \phi^n + \Delta t F(\phi^n) \qquad (22.2\text{-}21)$$

(This method is also referred to as "global time stepping".)

Here, the time step Δt is restricted to the stability limit of the underlying solver (i.e., a time step corresponding to a Courant number of approximately 1). In order to be time-accurate, all cells in the domain must use the same time step. For stability, this time step must be the minimum of all the local time steps in the domain.

The use of explicit time stepping is fairly restrictive. It is used primarily to capture the transient behavior of moving waves, such as shocks, because it is more accurate and less expensive than the implicit time stepping methods in such cases. You *cannot* use explicit time stepping in the following cases:

- Calculations with the segregated or coupled implicit solver. The explicit time stepping formulation is available only with the coupled explicit solver.

- Incompressible flow. Explicit time stepping cannot be used to compute time-accurate incompressible flows (i.e., gas laws other than ideal gas). Incompressible solutions must be iterated to convergence within each time step.
- Convergence acceleration. FAS multigrid and residual smoothing cannot be used with explicit time stepping because they destroy the time accuracy of the underlying solver.

22.3 The Segregated Solver

In this section, special practices related to the discretization of the momentum and continuity equations and their solution by means of the segregated solver are addressed. These practices are most easily described by considering the steady-state continuity and momentum equations in integral form:

$$\oint \rho\, \vec{v} \cdot d\vec{A} = 0 \tag{22.3-1}$$

$$\oint \rho\vec{v}\, \vec{v} \cdot d\vec{A} = -\oint p\boldsymbol{I} \cdot d\vec{A} + \oint \overline{\overline{\tau}} \cdot d\vec{A} + \int_V \vec{F}\, dV \tag{22.3-2}$$

where $\boldsymbol{I}$ is the identity matrix, $\overline{\overline{\tau}}$ is the stress tensor, and $\vec{F}$ is the force vector.

22.3.1 Discretization of the Momentum Equation

The discretization scheme described in Section 22.2 for a scalar transport equation is also used to discretize the momentum equations. For example, the x-momentum equation can be obtained by setting $\phi = u$:

$$a_P\, u = \sum_{\rm nb} a_{\rm nb}\, u_{\rm nb} + \sum p_f \mathrm{A} \cdot \hat{\imath} + S \tag{22.3-3}$$

If the pressure field and face mass fluxes were known, Equation 22.3-3 could be solved in the manner outlined in Section 22.2, and a velocity field obtained. However, the pressure field and face mass fluxes are not known a priori and must be obtained as a part of the solution. There are important issues with respect to the storage of pressure and the discretization of the pressure gradient term; these are addressed next.

FLUENT uses a co-located scheme, whereby pressure and velocity are both stored at cell centers. However, Equation 22.3-3 requires the value of the pressure at the face between cells $c0$ and $c1$, shown in Figure 22.2.1. Therefore, an interpolation scheme is required to compute the face values of pressure from the cell values.

Pressure Interpolation Schemes

The default scheme in FLUENT interpolates the pressure values at the faces using momentum equation coefficients [192]. This procedure works well as long as the pressure variation between cell centers is smooth. When there are jumps or large gradients in the momentum source terms between control volumes, the pressure profile has a high gradient at the cell face, and cannot be interpolated using this scheme. If this scheme is used, the discrepancy shows up in overshoots/undershoots of cell velocity.

Flows for which the standard pressure interpolation scheme will have trouble include flows with large body forces, such as in strongly swirling flows, in high-Rayleigh-number natural convection and the like. In such cases, it is necessary to pack the mesh in regions of high gradient to resolve the pressure variation adequately.

Another source of error is that FLUENT assumes that the normal pressure gradient at the wall is zero. This is valid for boundary layers, but not in the presence of body forces or curvature. Again, the failure to correctly account for the wall pressure gradient is manifested in velocity vectors pointing in/out of walls.

Several alternate methods are available for cases in which the standard pressure interpolation scheme is not valid:

- The linear scheme computes the face pressure as the average of the pressure values in the adjacent cells.
- The second-order scheme reconstructs the face pressure in the manner used for second-order accurate convection terms (see Section 22.2.3). This scheme may provide some improvement over the standard and linear schemes, but it may have some trouble if it is used at the start of a calculation and/or with a bad mesh. The second-order scheme is not applicable for flows with discontinuous pressure gradients imposed by the presence of a porous medium in the domain or the use of the VOF or mixture model for multiphase flow.
- The body-force-weighted scheme computes the face pressure by assuming that the normal gradient of the difference between pressure

and body forces is constant. This works well if the body forces are known a priori in the momentum equations (e.g., buoyancy and axisymmetric swirl calculations).

- The PRESTO! (PREssure STaggering Option) scheme uses the discrete continuity balance for a "staggered" control volume about the face to compute the "staggered" (i.e., face) pressure. This procedure is similar in spirit to the staggered-grid schemes used with structured meshes [172]. Note that for triangular and tetrahedral meshes, comparable accuracy is obtained using a similar algorithm.

See Section 22.7.3 for recommendations on when to use these alternate schemes.

22.3.2 Discretization of the Continuity Equation

Equation 22.3-1 may be integrated over the control volume in Figure 22.2.1 to yield the following discrete equation

$$\sum_{f}^{N_{\text{faces}}} J_f A_f = 0 \tag{22.3-4}$$

where J_f is the mass flux through face f, ρv_n.

As described in Section 22.1, the momentum and continuity equations are solved sequentially. In this sequential procedure, the continuity equation is used as an equation for pressure. However, pressure does not appear explicitly in Equation 22.3-4 for incompressible flows, since density is not directly related to pressure. The SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) family of algorithms [172] is used for introducing pressure into the continuity equation. This procedure is outlined in Section 22.3.3.

In order to proceed further, it is necessary to relate the face values of velocity, $\vec{v}_n$, to the stored values of velocity at the cell centers. Linear interpolation of cell-centered velocities to the face results in unphysical checker-boarding of pressure. FLUENT uses a procedure similar to that outlined by Rhie and Chow [192] to prevent checkerboarding. The face value of velocity is not averaged linearly; instead, momentum-weighted averaging, using weighting factors based on the a_P coefficient from equation 22.3-3, is performed. Using this procedure, the face flux, J_f, may be written as

$$J_f = \hat{J}_f + d_f \left(p_{c0} - p_{c1}\right) \tag{22.3-5}$$

where p_{c0} and p_{c1} are the pressures within the two cells on either side of the face, and $\hat{J}_f$ contains the influence of velocities in these cells (see Figure 22.2.1). The term d_f is a function of $\bar{a}_P$, the average of the momentum equation a_P coefficients for the cells on either side of face f.

Density Interpolation Schemes

For compressible flow calculations (i.e., calculations that use the ideal gas law for density), FLUENT applies upwind interpolation of density at cell faces. (For incompressible flows, FLUENT uses arithmetic averaging.) Three interpolation schemes are available for the density upwinding at cell faces: first-order upwind (default), second-order-upwind, and QUICK.

The first-order upwind scheme (based on [109]) sets the density at the cell face to be the upstream cell-center value. This scheme provides stability for the discretization of the pressure-correction equation, and gives good results for most classes of flows. The first-order scheme is the default scheme for compressible flows.

The second-order upwind scheme provides stability for supersonic flows and captures shocks better than the first-order upwind scheme. The QUICK scheme for density is similar to the QUICK scheme used for other variables. See Section 22.2.4 for details.

! The second-order upwind and QUICK schemes for density are not available for compressible multiphase calculations; the first-order upwind scheme is used for the compressible phase, and arithmetic averaging is used for the incompressible phases.

See Section 22.7.4 for recommendations on choosing an appropriate density interpolation scheme for your compressible flow.

22.3.3 Pressure-Velocity Coupling

Pressure-velocity coupling is achieved by using Equation 22.3-5 to derive an equation for pressure from the discrete continuity equation (Equation 22.3-4) FLUENT provides the option to choose among three pressure-velocity coupling algorithms: SIMPLE, SIMPLEC, and PISO. See Section 22.8 for instructions on how to select these algorithms.

SIMPLE

The SIMPLE algorithm uses a relationship between velocity and pressure corrections to enforce mass conservation and to obtain the pressure field.

If the momentum equation is solved with a guessed pressure field p^*, the resulting face flux, J_f^*, computed from Equation 22.3-5

$$J_f^* = \hat{J}_f^* + d_f\,(p_{c0}^* - p_{c1}^*) \tag{22.3-6}$$

does not satisfy the continuity equation. Consequently, a correction J_f' is added to the face flux J_f^* so that the corrected face flux, J_f

$$J_f = J_f^* + J_f' \tag{22.3-7}$$

satisfies the continuity equation. The SIMPLE algorithm postulates that J_f' be written as

$$J_f' = d_f\,(p_{c0}' - p_{c1}') \tag{22.3-8}$$

where p' is the cell pressure correction.

The SIMPLE algorithm substitutes the flux correction equations (Equations 22.3-7 and 22.3-8) into the discrete continuity equation (Equation 22.3-4) to obtain a discrete equation for the pressure correction p' in the cell:

$$a_P\,p' = \sum_{\rm nb} a_{\rm nb}\,p_{\rm nb}' + b \tag{22.3-9}$$

where the source term b is the net flow rate into the cell:

$$b = \sum_{f}^{N_{\text{faces}}} J_f^* A_f \tag{22.3-10}$$

The pressure-correction equation (Equation 22.3-9) may be solved using the algebraic multigrid (AMG) method described in Section 22.5.3. Once a solution is obtained, the cell pressure and the face flux are corrected using

$$p = p^* + \alpha_p \, p' \tag{22.3-11}$$

$$J_f = J_f^* + d_f \, (p'_{c0} - p'_{c1}) \tag{22.3-12}$$

Here α_p is the under-relaxation factor for pressure (see Section 22.2.7 for information about under-relaxation). The corrected face flux, J_f, satisfies the discrete continuity equation identically during each iteration.

SIMPLEC

A number of variants of the basic SIMPLE algorithm are available in the literature. In addition to SIMPLE, FLUENT offers the SIMPLEC (SIMPLE-Consistent) algorithm [243]. SIMPLE is the default, but many problems will benefit from the use of SIMPLEC, as described in Section 22.8.1.

The SIMPLEC procedure is similar to the SIMPLE procedure outlined above. The only difference lies in the expression used for the face flux correction, J'_f. As in SIMPLE, the correction equation may be written as

$$J_f = J_f^* + d_f \, (p'_{c0} - p'_{c1}) \tag{22.3-13}$$

However, the coefficient d_f is redefined as a function of $(\overline{a_P - \sum_{\text{nb}} a_{\text{nb}}})$ The use of this modified correction equation has been shown to accelerate convergence in problems where pressure-velocity coupling is the main deterrent to obtaining a solution.

PISO

The Pressure-Implicit with Splitting of Operators (PISO) pressure-velocity coupling scheme, part of the SIMPLE family of algorithms, is based on the higher degree of the approximate relation between the corrections for pressure and velocity. One of the limitations of the SIMPLE and SIMPLEC algorithms is that new velocities and corresponding fluxes do not satisfy the momentum balance after the pressure-correction equation is solved. As a result, the calculation must be repeated until the balance is satisfied. To improve the efficiency of this calculation, the PISO algorithm performs two additional corrections: neighbor correction and skewness correction.

Neighbor Correction

The main idea of the PISO algorithm is to move the repeated calculations required by SIMPLE and SIMPLEC inside the solution stage of the pressure-correction equation [97]. After one or more additional PISO loops, the corrected velocities satisfy the continuity and momentum equations more closely. This iterative process is called a momentum correction or "neighbor correction". The PISO algorithm takes a little more CPU time per solver iteration, but it can dramatically decrease the number of iterations required for convergence, especially for transient problems.

Skewness Correction

For meshes with some degree of skewness, the approximate relationship between the correction of mass flux at the cell face and the difference of the pressure corrections at the adjacent cells is very rough. Since the components of the pressure-correction gradient along the cell faces are not known in advance, an iterative process similar to the PISO neighbor correction described above is desirable [64]. After the initial solution of the pressure-correction equation, the pressure-correction gradient is recalculated and used to update the mass flux corrections. This process, which is referred to as "skewness correction", significantly reduces convergence difficulties associated with highly distorted meshes. The PISO skewness correction allows FLUENT to obtain a solution on a highly skewed mesh in approximately the same number of iterations as required for a more orthogonal mesh.

Special Treatment for Strong Body Forces in Multiphase Flows

When large body forces (e.g., gravity or surface tension forces) exist in multiphase flows, the body force and pressure gradient terms in the momentum equation are almost in equilibrium, with the contributions of convective and viscous terms small in comparison. Segregated algorithms converge poorly unless partial equilibrium of pressure gradient and body forces is taken into account. FLUENT provides an optional "implicit body force" treatment that can account for this effect, making the solution more robust.

The basic procedure involves augmenting the correction equation for the face flow rate, Equation 22.3-12, with an additional term involving corrections to the body force. This results in extra body force correction terms in Equation 22.3-10, and allows the flow to achieve a realistic pressure field very early in the iterative process.

This option is available only for multiphase calculations, but it is turned off by default. Section 20.6.11 includes instructions for turning on the implicit body force treatment.

In addition, FLUENT allows you to control the change in the body forces through the use of an under-relaxation factor for body forces.

22.3.4 Steady-State and Time-Dependent Calculations

The governing equations for the segregated solver do not contain any time-dependent terms if you are performing a steady-state calculation.

For time-accurate calculations, an implicit time stepping scheme is used. See Section 22.2.8 for details.

22.4 The Coupled Solver

The coupled solver in FLUENT solves the governing equations of continuity, momentum, and (where appropriate) energy and species transport simultaneously as a set, or vector, of equations. Governing equations for additional scalars will be solved sequentially (i.e., segregated from one another and from the coupled set).

22.4.1 Governing Equations in Vector Form

The system of governing equations for a single-component fluid, written to describe the mean flow properties, is cast in integral, Cartesian form for an arbitrary control volume V with differential surface area $d\boldsymbol{A}$ as follows:

$$\frac{\partial}{\partial t}\int_V \boldsymbol{W}\, dV + \oint [\boldsymbol{F} - \boldsymbol{G}] \cdot d\boldsymbol{A} = \int_V \boldsymbol{H}\, dV \tag{22.4-1}$$

where the vectors $\boldsymbol{W}, \boldsymbol{F}$, and $\boldsymbol{G}$ are defined as

$$\boldsymbol{W} = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{Bmatrix}, \quad \boldsymbol{F} = \begin{Bmatrix} \rho \boldsymbol{v} \\ \rho \boldsymbol{v} u + p\hat{\boldsymbol{\imath}} \\ \rho \boldsymbol{v} v + p\hat{\boldsymbol{\jmath}} \\ \rho \boldsymbol{v} w + p\hat{\boldsymbol{k}} \\ \rho \boldsymbol{v} E + p\boldsymbol{v} \end{Bmatrix}, \quad \boldsymbol{G} = \begin{Bmatrix} 0 \\ \boldsymbol{\tau}_{xi} \\ \boldsymbol{\tau}_{yi} \\ \boldsymbol{\tau}_{zi} \\ \boldsymbol{\tau}_{ij} v_j + \boldsymbol{q} \end{Bmatrix}$$

and the vector $\boldsymbol{H}$ contains source terms such as body forces and energy sources.

Here ρ, $\boldsymbol{v}$, E, and p are the density, velocity, total energy per unit mass, and pressure of the fluid, respectively. $\boldsymbol{\tau}$ is the viscous stress tensor, and $\boldsymbol{q}$ is the heat flux.

Total energy E is related to the total enthalpy H by

$$E = H - p/\rho \tag{22.4-2}$$

where

$$H = h + |\boldsymbol{v}|^2/2 \tag{22.4-3}$$

The Navier-Stokes equations as expressed in Equation 22.4-1 become (numerically) very stiff at low Mach number due to the disparity between the fluid velocity $\boldsymbol{v}$ and the acoustic speed c (speed of sound). This is also true for incompressible flows, regardless of the fluid velocity, because acoustic waves travel infinitely fast in an incompressible fluid (speed of sound is infinite). The numerical stiffness of the equations under these conditions results in poor convergence rates. This difficulty is overcome in FLUENT's coupled solver by employing a technique called (time-derivative) preconditioning [261].

22.4.2 Preconditioning

Time-derivative preconditioning modifies the time-derivative term in Equation 22.4-1 by pre-multiplying it with a preconditioning matrix. This has the effect of re-scaling the acoustic speed (eigenvalue) of the system of equations being solved in order to alleviate the numerical stiffness encountered in low Mach number and incompressible flow.

Derivation of the preconditioning matrix begins by transforming the dependent variable in Equation 22.4-1 from conserved quantities $\boldsymbol{W}$ to primitive variables $\boldsymbol{Q}$ using the chain-rule as follows:

$$\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{Q}} \frac{\partial}{\partial t} \int_V \boldsymbol{Q}\, dV + \oint [\boldsymbol{F} - \boldsymbol{G}] \cdot d\boldsymbol{A} = \int_V \boldsymbol{H}\, dV \tag{22.4-4}$$

where $\boldsymbol{Q}$ is the vector $\{p, u, v, w, T\}^T$ and the Jacobian $\partial \boldsymbol{W}/\partial \boldsymbol{Q}$ is given by

$$\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{Q}} = \begin{bmatrix} \rho_p & 0 & 0 & 0 & \rho_T \\ \rho_p u & \rho & 0 & 0 & \rho_T u \\ \rho_p v & 0 & \rho & 0 & \rho_T v \\ \rho_p w & 0 & 0 & \rho & \rho_T w \\ \rho_p H - \delta & \rho u & \rho v & \rho w & \rho_T H + \rho C_p \end{bmatrix} \tag{22.4-5}$$

where

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

and $\delta = 1$ for an ideal gas and $\delta = 0$ for an incompressible fluid.

The choice of primitive variables $\boldsymbol{Q}$ as dependent variables is desirable for several reasons. First, it is a natural choice when solving incompressible flows. Second, when we use second-order accuracy we need to reconstruct $\boldsymbol{Q}$ rather than $\boldsymbol{W}$ in order to obtain more accurate velocity and temperature gradients in viscous fluxes, and pressure gradients in inviscid fluxes. And finally, the choice of pressure as a dependent variable allows the propagation of acoustic waves in the system to be singled out [245].

We precondition the system by replacing the Jacobian matrix $\partial\boldsymbol{W}/\partial\boldsymbol{Q}$ (Equation 22.4-5) with the preconditioning matrix Γ so that the preconditioned system in conservation form becomes

$$\Gamma\frac{\partial}{\partial t}\int_V \boldsymbol{Q}\, dV + \oint [\boldsymbol{F} - \boldsymbol{G}] \cdot d\boldsymbol{A} = \int_V \boldsymbol{H}\, dV \qquad (22.4\text{-}6)$$

where

$$\Gamma = \begin{bmatrix} \Theta & 0 & 0 & 0 & \rho_T \\ \Theta u & \rho & 0 & 0 & \rho_T u \\ \Theta v & 0 & \rho & 0 & \rho_T u \\ \Theta w & 0 & 0 & \rho & \rho_T u \\ \Theta H - \delta & \rho u & \rho v & \rho w & \rho_T H + \rho C_p \end{bmatrix} \qquad (22.4\text{-}7)$$

The parameter Θ is given by

$$\Theta = \left(\frac{1}{U_r^2} - \frac{\rho_T}{\rho C_p}\right) \qquad (22.4\text{-}8)$$

The reference velocity U_r appearing in Equation 22.4-8 is chosen locally such that the eigenvalues of the system remain well conditioned with respect to the convective and diffusive time scales [261].

The resultant eigenvalues of the preconditioned system (Equation 22.4-6) are given by

$$u,\ u,\ u,\ u'+c',\ u'-c' \tag{22.4-9}$$

where

$$\begin{aligned}
u &= \boldsymbol{v}\cdot\hat{n}\\
u' &= u\,(1-\alpha)\\
c' &= \sqrt{\alpha^2u^2+U_r^2}\\
\alpha &= \left(1-\beta U_r^2\right)/2\\
\beta &= \left(\rho_p+\frac{\rho_T}{\rho C_p}\right)
\end{aligned}$$

For an ideal gas, $\beta=(\gamma RT)^{-1}=1/c^2$. Thus, when $U_r=c$ (at sonic speeds and above), $\alpha=0$ and the eigenvalues of the preconditioned system take their traditional form, $u\pm c$. At low speed, however, as $U_r\rightarrow 0$, $\alpha\rightarrow 1/2$ and all eigenvalues become of the same order as u. For constant-density flows, $\beta=0$ and $\alpha=1/2$ regardless of the values of U_r. As long as the reference velocity is of the same order as the local velocity, all eigenvalues remain of the order u. Thus, the eigenvalues of the preconditioned system remain well conditioned at all speeds.

Note that the non-preconditioned Navier-Stokes equations are recovered exactly from Equation 22.4-6 by setting $1/U_r^2$ to ρ_p, the derivative of density with respect to pressure. In this case Γ reduces exactly to the Jacobian $\partial\boldsymbol{W}/\partial\boldsymbol{Q}$.

Although Equation 22.4-6 is conservative in the steady state, it is not, in a strict sense, conservative for time-dependent flows. This is not a problem, however, since the preconditioning has already destroyed the time accuracy of the equations and we will not employ them in this form for unsteady calculations.

Flux-Difference Splitting

The inviscid flux vector $\boldsymbol{F}$ appearing in Equation 22.4-6 is evaluated by a standard upwind, flux-difference splitting [194]. This approach acknowledges that the flux vector $\boldsymbol{F}$ contains characteristic information propagating through the domain with speed and direction according to the eigenvalues of the system. By splitting $\boldsymbol{F}$ into parts, where each part contains information traveling in a particular direction (i.e., characteristic information), and upwind differencing the split fluxes in a manner consistent with their corresponding eigenvalues, we obtain the following expression for the discrete flux at each face:

$$\boldsymbol{F} = \frac{1}{2}\left(\boldsymbol{F}_R + \boldsymbol{F}_L\right) - \frac{1}{2}\Gamma\,|\hat{\mathrm{A}}|\,\delta\boldsymbol{Q} \qquad (22.4\text{-}10)$$

Here $\delta\boldsymbol{Q}$ is the spatial difference $\boldsymbol{Q}_R - \boldsymbol{Q}_L$. The fluxes $\boldsymbol{F}_R = \boldsymbol{F}\left(\boldsymbol{Q}_R\right)$ and $\boldsymbol{F}_L = \boldsymbol{F}\left(\boldsymbol{Q}_L\right)$ are computed using the (reconstructed) solution vectors $\boldsymbol{Q}_R$ and $\boldsymbol{Q}_L$ on the "right" and "left" side of the face. The matrix $|\hat{\mathrm{A}}|$ is defined by

$$|\hat{\mathrm{A}}| = M\,|\Lambda|\,M^{-1} \qquad (22.4\text{-}11)$$

where Λ is the diagonal matrix of eigenvalues and M is the modal matrix that diagonalizes $\Gamma^{-1}\mathrm{A}$, where A is the inviscid flux Jacobian $\partial\boldsymbol{F}/\partial\boldsymbol{Q}$.

For the non-preconditioned system (and an ideal gas) Equation 22.4-10 reduces to Roe's flux-difference splitting [194] when Roe-averaged values are used to evaluate $\Gamma\,|\hat{\mathrm{A}}|$. At present, arithmetic averaging of states $\boldsymbol{Q}_R$ and $\boldsymbol{Q}_L$ is used.

In its current form, Equation 22.4-10 can be viewed as a second-order central difference plus an added matrix dissipation. The added matrix dissipation term is not only responsible for producing an upwinding of the convected variables, and of pressure and flux velocity in supersonic flow, but it also provides the pressure-velocity coupling required for stability and efficient convergence of low-speed and incompressible flows.

22.4.3 Time Marching for Steady-State Flows

The coupled set of governing equations (Equation 22.4-6) in FLUENT is discretized in time for both steady and unsteady calculations. In the steady case, it is assumed that time marching proceeds until a steady-state solution is reached. Temporal discretization of the coupled equations is accomplished by either an implicit or an explicit time-marching scheme. These two algorithms are described below.

Explicit Scheme

In the explicit scheme a multi-stage, time-stepping algorithm [101] is used to discretize the time derivative in Equation 22.4-6. The solution is advanced from iteration n to iteration $n+1$ with an m-stage Runge-Kutta scheme, given by

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

where $\Delta \boldsymbol{Q}^i \equiv \boldsymbol{Q}^i - \boldsymbol{Q}^n$ and $i = 1, 2, \ldots, m$ is the stage counter for the m-stage scheme. α_i is the multi-stage coefficient for the i^{th} stage. The residual $\boldsymbol{R}^i$ is computed from the intermediate solution $\boldsymbol{Q}^i$ and, for Equation 22.4-6, is given by

$$\mathbf{R}^i = \sum^{N_{\text{faces}}} \Big(\boldsymbol{F}(\boldsymbol{Q}^i) - \boldsymbol{G}(\boldsymbol{Q}^i) \Big) \cdot \boldsymbol{A} - V \boldsymbol{H} \tag{22.4-12}$$

The time step Δt is computed from the CFL (Courant-Friedrichs-Lewy) condition

$$\Delta t = \frac{\text{CFL} \Delta x}{\lambda_{\max}} \tag{22.4-13}$$

where $\lambda_{\max}$ is the maximum of the local eigenvalues defined by Equation 22.4-9.

The convergence rate of the explicit scheme can be accelerated through use of the full-approximation storage (FAS) multigrid method described in Section 22.5.4.

Implicit Residual Smoothing

The maximum time step can be further increased by increasing the support of the scheme through implicit averaging of the residuals with their neighbors. The residuals are filtered through a Laplacian smoothing operator:

$$\bar{R}_i = R_i + \epsilon \sum (\bar{R}_j - \bar{R}_i) \tag{22.4-14}$$

This equation can be solved with the following Jacobi iteration:

$$\bar{R}_i^m = \frac{R_i + \epsilon \sum \bar{R}_j^{m-1}}{1 + \epsilon \sum 1} \tag{22.4-15}$$

Two Jacobi iterations are usually sufficient to allow doubling the time step with a value of $\epsilon = 0.5$.

Implicit Scheme

In the implicit scheme, an Euler implicit discretization in time of the governing equations (Equation 22.4-6) is combined with a Newton-type linearization of the fluxes to produce the following linearized system in delta form [259]:

$$\left[\mathrm{D} + \sum_j^{N_{\rm faces}} \mathrm{S}_{j,k} \right] \Delta \boldsymbol{Q}^{n+1} = -\boldsymbol{R}^n \tag{22.4-16}$$

The center and off-diagonal coefficient matrices, D and $\mathrm{S}_{j,k}$ are given by

$$\mathrm{D} \quad = \quad \frac{V}{\Delta t} \Gamma + \sum_j^{N_{\rm faces}} \mathrm{S}_{j,i} \tag{22.4-17}$$

$$\mathrm{S}_{j,k} = \left(\frac{\partial \boldsymbol{F}_j}{\partial \boldsymbol{Q}_k} - \frac{\partial \boldsymbol{G}_j}{\partial \boldsymbol{Q}_k}\right) \qquad (22.4\text{-}18)$$

and the residual vector $\boldsymbol{R}^n$ and time step Δt are defined as in Equation 22.4-12 and Equation 22.4-13, respectively.

Equation 22.4-16 is solved using a point Gauss-Seidel scheme in conjunction with an algebraic multigrid (AMG) method (see Section 22.5.3) adapted for coupled sets of equations.

22.4.4 Temporal Discretization for Unsteady Flows

For time-accurate calculations, explicit and implicit time-stepping schemes are available. (The implicit approach is also referred to as "dual time stepping".)

Explicit Time Stepping

In the explicit time stepping approach, the explicit scheme described above is employed, using the same time step in each cell of the domain, and with preconditioning disabled.

Dual Time Stepping

To provide for efficient, time-accurate solution of the preconditioned equations, we employ a dual time-stepping, multi-stage scheme. Here we introduce a preconditioned pseudo-time-derivative term into Equation 22.4-1 as follows:

$$\frac{\partial}{\partial t}\int_V \boldsymbol{W}\, dV + \Gamma\frac{\partial}{\partial \tau}\int_V \boldsymbol{Q}\, dV + \oint [\boldsymbol{F} - \boldsymbol{G}] \cdot d\boldsymbol{A} = \int_V \boldsymbol{H}\, dV \qquad (22.4\text{-}19)$$

where t denotes physical-time and τ is a pseudo-time used in the time-marching procedure. Note that as $\tau \to \infty$, the second term on the LHS of Equation 22.4-19 vanishes and Equation 22.4-1 is recovered.

The time-dependent term in Equation 22.4-19 is discretized in an implicit fashion by means of either a first- or second-order accurate, backward difference in time. This is written in semi-discrete form as follows:

$$\begin{aligned}\left[\frac{\Gamma}{\Delta\tau}+\frac{\epsilon_0}{\Delta t}\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{Q}}\right]\Delta\boldsymbol{Q}^{k+1} \quad &+ \quad \frac{1}{V}\oint[\boldsymbol{F}-\boldsymbol{G}]\cdot d\boldsymbol{A} \\ = \boldsymbol{H} \quad &+ \quad \frac{1}{\Delta t}\left(\epsilon_0\boldsymbol{W}^k-\epsilon_1\boldsymbol{W}^n+\epsilon_2\boldsymbol{W}^{n-1}\right)\end{aligned}$$

where $\{\epsilon_0=\epsilon_1=1/2, \epsilon_2=0\}$ gives first-order time accuracy, and $\{\epsilon_0 = 3/2, \epsilon_1=2, \epsilon_2=1/2\}$ gives second-order. k is the inner iteration counter and n represents any given physical-time level.

The pseudo-time-derivative is driven to zero at each physical time level by a series of inner iterations using either the implicit or explicit time-marching algorithm. Throughout the (inner) iterations in pseudo-time, $\boldsymbol{W}^n$ and $\boldsymbol{W}^{n-1}$ are held constant and $\boldsymbol{W}^k$ is computed from $\boldsymbol{Q}^k$. As $\tau\rightarrow\infty$, the solution at the next physical time level $\boldsymbol{W}^{n+1}$ is given by $\boldsymbol{W}(\boldsymbol{Q}^k)$.

Note that the physical time step Δt is limited only by the level of desired temporal accuracy. The pseudo-time-step $\Delta\tau$ is determined by the CFL condition of the (implicit or explicit) time-marching scheme.

22.5 Multigrid Method

The FLUENT solver contains two forms of multigrid: algebraic (AMG) and full-approximation storage (FAS). As discussed in Section 22.1.3, AMG is an essential component of both the segregated and coupled implicit solvers, while FAS is an important, but optional, component of the coupled explicit solver. (Note that when the coupled explicit solver is used, since the scalar equations (e.g., turbulence) are solved using the segregated approach, AMG will also be used.)

This section describes the mathematical basis of the multigrid approach. Common aspects of AMG and FAS are presented first, followed by separate sections that provide details unique to each method. Information about user inputs and controls for the multigrid solver is provided in Sections 22.19.3 and 22.19.4.

22.5.1 Approach

FLUENT uses a multigrid scheme to accelerate the convergence of the solver by computing corrections on a series of coarse grid levels. The use of this multigrid scheme can greatly reduce the number of iterations and the CPU time required to obtain a converged solution, particularly when your model contains a large number of control volumes.

The Need for Multigrid

Implicit solution of the linearized equations on unstructured meshes is complicated by the fact that there is no equivalent of the line-iterative methods that are commonly used on structured grids. Since direct matrix inversion is out of the question for realistic problems and "whole-field" solvers that rely on conjugate-gradient (CG) methods have robustness problems associated with them, the methods of choice are point implicit solvers like Gauss-Seidel. Although the Gauss-Seidel scheme rapidly removes local (high-frequency) errors in the solution, global (low-frequency) errors are reduced at a rate inversely related to the grid size. Thus, for a large number of nodes, the solver "stalls" and the residual reduction rate becomes prohibitively low.

The multi-stage scheme used in the coupled explicit solver can efficiently remove local (high-frequency) errors as well. That is, the effect of the solution in one cell is communicated to adjacent cells relatively quickly. However, the scheme is less effective at reducing global (low-frequency) errors—errors which exist over a large number of control volumes. Thus, global corrections to the solution across a large number of control volumes occur slowly, over many iterations. This implies that performance of the multi-stage scheme will deteriorate as the number of control volumes increases.

Multigrid techniques allow global error to be addressed by using a sequence of successively coarser meshes. This method is based upon the principle that global (low-frequency) error existing on a fine mesh can be represented on a coarse mesh where it again becomes accessible as local (high-frequency) error: because there are fewer coarse cells overall, the global corrections can be communicated more quickly between adjacent cells. Since computations can be performed at exponentially decaying expense in both CPU time and memory storage on coarser meshes, there is the potential for very efficient elimination of global error. The fine-grid relaxation scheme or "smoother", in this case either the point-implicit Gauss-Seidel or the explicit multi-stage scheme, is not required to be particularly effective at reducing global error and can be tuned for efficient reduction of local error.

The Basic Concept in Multigrid

Consider the set of discretized linear (or linearized) equations given by

$$A\,\phi_e + b = 0 \tag{22.5-1}$$

where ϕ_e is the exact solution. Before the solution has converged there will be a defect d associated with the approximate solution ϕ:

$$A\,\phi + b = d \tag{22.5-2}$$

We seek a correction ψ to ϕ such that the exact solution is given by

$$\phi_e = \phi + \psi \tag{22.5-3}$$

Substituting Equation 22.5-3 into Equation 22.5-1 gives

$$\begin{aligned} A\,(\phi + \psi) + b &= 0 & (22.5\text{-}4) \\ A\,\psi + (A\,\phi + b) &= 0 & (22.5\text{-}5) \end{aligned}$$

Now using Equations 22.5-2 and 22.5-5 we obtain

$$A\,\psi + d = 0 \tag{22.5-6}$$

which is an equation for the correction in terms of the original fine level operator A and the defect d. Assuming the local (high-frequency) errors have been sufficiently damped by the relaxation scheme on the fine level, the correction ψ will be smooth and therefore more effectively solved on the next coarser level.

Restriction and Prolongation

Solving for corrections on the coarse level requires transferring the defect down from the fine level (restriction), computing corrections, and then transferring the corrections back up from the coarse level (prolongation). We can write the equations for coarse level corrections ψ^H as

$$A^H \psi^H + R\, d = 0 \qquad (22.5\text{-}7)$$

where A^H is the coarse level operator and R the restriction operator responsible for transferring the fine level defect down to the coarse level. Solution of Equation 22.5-7 is followed by an update of the fine level solution given by

$$\phi^{\rm new} = \phi + P\, \psi^H \qquad (22.5\text{-}8)$$

where P is the prolongation operator used to transfer the coarse level corrections up to the fine level.

Unstructured Multigrid

The primary difficulty with using multigrid on unstructured grids is the creation and use of the coarse grid hierarchy. On a structured grid, the coarse grids can be formed simply by removing every other grid line from the fine grids and the prolongation and restriction operators are simple to formulate (e.g., injection and bilinear interpolation).

The difficulties of applying multigrid on unstructured grids are overcome in separate fashion by each of the two multigrid methods used in FLUENT. While the basic principles discussed so far and the cycling strategy described in Section 22.5.2 are the same, the techniques for construction of restriction, prolongation, and coarse grid operators are different, as discussed in Section 22.5.3 and Section 22.5.4 for the AMG and FAS methods, respectively.

22.5.2 Multigrid Cycles

A multigrid cycle can be defined as a recursive procedure that is applied at each grid level as it moves through the grid hierarchy. Four types of multigrid cycles are available in FLUENT: the V, W, F, and flexible ("flex") cycles. The V and W cycles are available in both AMG and FAS, while the F and flexible cycles are restricted to the AMG method only. (The W and flex AMG cycles are not available for solving the coupled equation set due to the amount of computation required.)

The V and W Cycles

Figures 22.5.1 and 22.5.2 show the V and W multigrid cycles (defined below). In each figure, the multigrid cycle is represented by a square, and then expanded recursively to show the individual steps that are performed within the cycle. The individual steps are represented by a circle, one or more squares, and a triangle, connected by lines: circle-square-triangle for a V cycle, or circle-square-square-triangle for a W cycle. The squares in this group expand again, into circle-square-triangle or circle-square-square-triangle, and so on. You may want to follow along in the figures as you read the steps below.

For the V and W cycles, the traversal of the hierarchy is governed by three parameters, β_1, β_2, and β_3, as follows:

1. First, iterations are performed on the current grid level to reduce the high-frequency components of the error (local error). For AMG, one iteration consists of one forward and one backward Gauss-Seidel sweep. For FAS, one iteration consists of one pass of the multi-stage scheme (described in Section 22.4.3).

   These iterations are referred to as pre-relaxation sweeps because they are performed before moving to the next coarser grid level. The number of pre-relaxation sweeps is specified by β_1.

   In Figures 22.5.1 and 22.5.2 this step is represented by a circle and marks the start of a multigrid cycle. The high-wave-number components of error should be reduced until the remaining error is expressible on the next coarser mesh without significant aliasing.

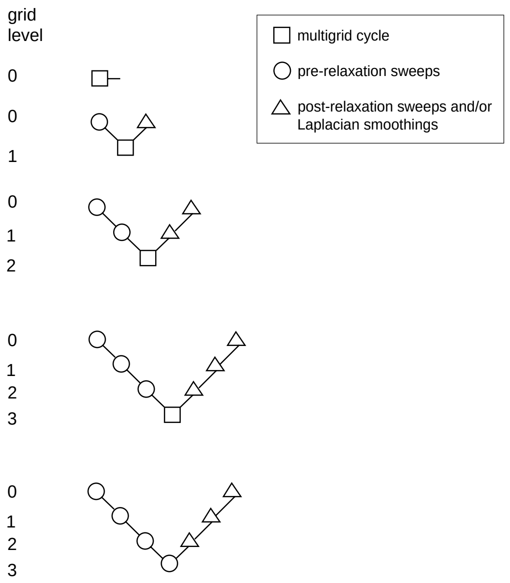


Figure 22.5.1: V-Cycle Multigrid

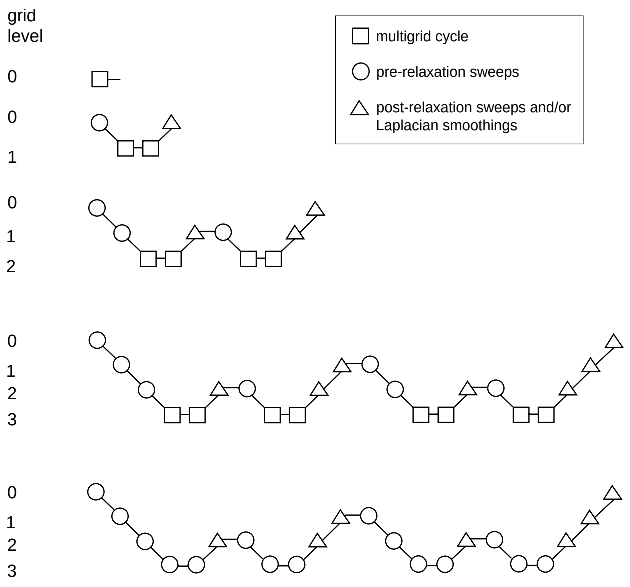


Figure 22.5.2: W-Cycle Multigrid

If this is the coarsest grid level, then the multigrid cycle on this level is complete. (In Figures 22.5.1 and 22.5.2 there are 3 coarse grid levels, so the square representing the multigrid cycle on level 3 is equivalent to a circle, as shown in the final diagram in each figure.)

! In the AMG method, the default value of β_1 is zero (i.e., no pre-relaxation sweeps are performed).

2. Next, the problem is "restricted" to the next coarser grid level using Equation 22.5-7.

   In Figures 22.5.1 and 22.5.2, the restriction from a finer grid level to a coarser grid level is designated by a downward-sloping line.

3. The error on the coarse grid is reduced by performing a specified number (β_2) of multigrid cycles (represented in Figures 22.5.1 and 22.5.2 as squares). Commonly, for fixed multigrid strategies β_2 is either 1 or 2, corresponding to V-cycle and W-cycle multigrid, respectively.

4. Next, the cumulative correction computed on the coarse grid is "interpolated" back to the fine grid using Equation 22.5-8 and added to the fine grid solution. In the FAS method, the corrections are additionally smoothed during this step using the Laplacian smoothing operator discussed in Section 22.4.3.

   In Figures 22.5.1 and 22.5.2 the prolongation is represented by an upward-sloping line.

   The high-frequency error now present at the fine grid level is due to the prolongation procedure used to transfer the correction.

5. In the final step, iterations are performed on the fine grid to remove the high-frequency error introduced on the coarse grid by the multigrid cycles. These iterations are referred to as post-relaxation sweeps because they are performed after returning from the next coarser grid level. The number of post-relaxation sweeps is specified by β_3.

   In Figures 22.5.1 and 22.5.2, this relaxation procedure is represented by a single triangle.

For AMG, the default value of β_3 is 1. Since the default value for β_1 is 0 (i.e., pre-relaxation sweeps are not performed), this procedure is roughly equivalent to using the solution from the coarse level as the initial guess for the solution at the fine level. For FAS, the default value of β_3 is zero (i.e., post-relaxation sweeps are not performed); post-relaxation sweeps are never performed at the end of the cycle for the finest grid level, regardless of the value of β_3. This is because for FAS, post-relaxation sweeps at the fine level are equivalent to pre-relaxation sweeps during the next cycle.

22.5.3 Algebraic Multigrid (AMG)

This algorithm is referred to as an "algebraic" multigrid scheme because, as we shall see, the coarse level equations are generated without the use of any geometry or re-discretization on the coarse levels; a feature that makes AMG particularly attractive for use on unstructured meshes. The advantage being that no coarse grids have to be constructed or stored, and no fluxes or source terms need be evaluated on the coarse levels. This approach is in contrast with FAS (sometimes called "geometric") multigrid in which a hierarchy of meshes is required and the discretized equations are evaluated on every level. In theory, the advantage of FAS over AMG is that the former should perform better for non-linear problems since non-linearities in the system are carried down to the coarse levels through the re-discretization; when using AMG, once the system is linearized, non-linearities are not "felt" by the solver until the fine level operator is next updated.

AMG Restriction and Prolongation Operators

The restriction and prolongation operators used here are based on the additive correction (AC) strategy described for structured grids by Hutchinson and Raithby [96]. Inter-level transfer is accomplished by piecewise constant interpolation and prolongation. The defect in any coarse level cell is given by the sum of those from the fine level cells it contains, while fine level corrections are obtained by injection of coarse level values. In this manner the prolongation operator is given by the transpose of the restriction operator

$$P = R^T \tag{22.5-9}$$

The restriction operator is defined by a coarsening or "grouping" of fine level cells into coarse level ones. In this process each fine level cell is grouped with one or more of its "strongest" neighbors, with a preference given to currently ungrouped neighbors. The algorithm attempts to collect cells into groups of fixed size, typically two or four, but any number can be specified. In the context of grouping, strongest refers to the neighbor j of the current cell i for which the coefficient A_{ij} is largest. For sets of coupled equations A_{ij} is a block matrix and the measure of its magnitude is simply taken to be the magnitude of its first element. In addition, the set of coupled equations for a given cell are treated together and not divided amongst different coarse cells. This results in the same coarsening for each equation in the system.

AMG Coarse Level Operator

The coarse level operator A^H is constructed using a Galerkin approach. Here we require that the defect associated with the corrected fine level solution must vanish when transferred back to the coarse level. Therefore we may write

$$R\, d^{\mathrm{new}} = 0 \tag{22.5-10}$$

Upon substituting Equations 22.5-2 and 22.5-8 for d^{new} and ϕ^{new} we have

$$\begin{aligned} R\left[A\,\phi^{\mathrm{new}} + b\right] &= 0 \\ R\left[A\left(\phi + P\,\psi^H\right) + b\right] &= 0 \end{aligned} \tag{22.5-11}$$

Now rearranging and using Equation 22.5-2 once again gives

$$\begin{aligned} R\,A\,P\,\psi^H + R\left(A\,\phi + b\right) &= 0 \\ R\,A\,P\,\psi^H + R\,d &= 0 \end{aligned} \tag{22.5-12}$$

Comparison of Equation 22.5-12 with Equation 22.5-7 leads to the following expression for the coarse level operator:

$$A^H = R\,A\,P \tag{22.5-13}$$

The construction of coarse level operators thus reduces to a summation of diagonal and corresponding off-diagonal blocks for all fine level cells within a group to form the diagonal block of that group's coarse cell.

The F Cycle

The multigrid F cycle is essentially a combination of the V and W cycles described in Section 22.5.2.

Recall that the multigrid cycle is a recursive procedure. The procedure is expanded to the next coarsest grid level by performing a single multigrid cycle on the current level. Referring to Figures 22.5.1 and 22.5.2, this means replacing the square on the current level (representing a single cycle) with the procedure shown for the 0-1 level cycle (the second diagram in each figure). We see that a V cycle consists of:

pre sweep → restrict → V cycle → prolongate → post sweep

and a W cycle:

pre sweep → restrict → W cycle → W cycle → prolongate → post sweep

An F cycle is formed by a W cycle followed by a V cycle:

pre sweep → restrict → W cycle → V cycle → prolongate → post sweep

As expected, the F cycle requires more computation than the V cycle, but less than the W cycle. However, its convergence properties turn out to be better than the V cycle and roughly equivalent to the W cycle. The F cycle is the default AMG cycle type for the coupled equation set.

The Flexible Cycle

For the flexible cycle, the calculation and use of coarse grid corrections is controlled in the multigrid procedure by the logic illustrated in Figure 22.5.3. This logic ensures that coarser grid calculations are invoked when the rate of residual reduction on the current grid level is too slow. In addition, the multigrid controls dictate when the iterative solution of the correction on the current coarse grid level is sufficiently converged and should thus be applied to the solution on the next finer grid. These two decisions are controlled by the parameters α and β shown in Figure 22.5.3, as described in detail below. Note that the logic of the multigrid procedure is such that grid levels may be visited repeatedly during a single global iteration on an equation. For a set of 4 multigrid levels, referred to as 0, 1, 2, and 3, the flex-cycle multigrid procedure for solving a given transport equation might consist of visiting grid levels as 0-1-2-3-2-3-2-1-0-1-2-1-0, for example.

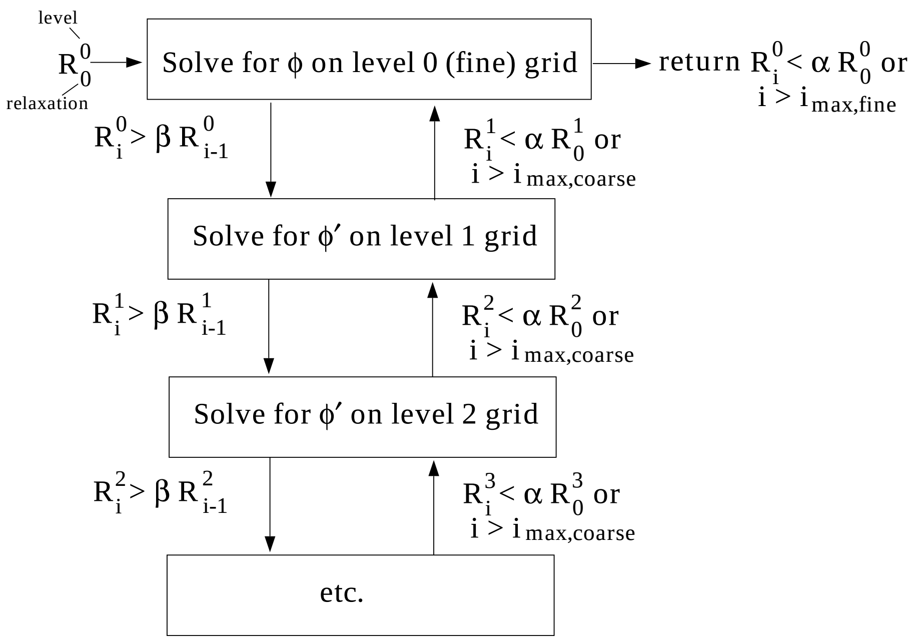

level
R_0^0
relaxation
Solve for ϕ on level 0 (fine) grid
return $R_i^0 < \alpha R_0^0$ or $i > i_{max,fine}$
$R_i^0 > \beta R_{i-1}^0$
$R_i^1 < \alpha R_0^1$ or $i > i_{max,coarse}$
Solve for ϕ' on level 1 grid
$R_i^1 > \beta R_{i-1}^1$
$R_i^2 < \alpha R_0^2$ or $i > i_{max,coarse}$
Solve for ϕ' on level 2 grid
$R_i^2 > \beta R_{i-1}^2$
$R_i^3 < \alpha R_0^3$ or $i > i_{max,coarse}$
etc.

Figure 22.5.3: Logic Controlling the Flex Multigrid Cycle

The main difference between the flexible cycle and the V and W cycles is that the satisfaction of the residual reduction tolerance and termination criterion determine when and how often each level is visited in the flexible cycle, whereas in the V and W cycles the traversal pattern is explicitly defined.

The Residual Reduction Rate Criteria

The multigrid procedure invokes calculations on the next coarser grid level when the error reduction rate on the current level is insufficient, as defined by

$$R_i > \beta R_{i-1} \tag{22.5-14}$$

Here R_i is the absolute sum of residuals (defect) computed on the current grid level after the ith relaxation on this level. The above equation states that if the residual present in the iterative solution after i relaxations is greater than some fraction, β (between 0 and 1), of the residual present after the $(i-1)$th relaxation, the next coarser grid level should be visited. Thus β is referred to as the residual reduction tolerance, and determines when to "give up" on the iterative solution at the current grid level and move to solving the correction equations on the next coarser grid. The value of β controls the frequency with which coarser grid levels are visited. The default value is 0.7. A larger value will result in less frequent visits, and a smaller value will result in more frequent visits.

The Termination Criteria

Provided that the residual reduction rate is sufficiently rapid, the correction equations will be converged on the current grid level and the result applied to the solution field on the next finer grid level.

The correction equations on the current grid level are considered sufficiently converged when the error in the correction solution is reduced to some fraction, α (between 0 and 1), of the original error on this grid level:

$$R_i < \alpha R_0 \tag{22.5-15}$$

Here, R_i is the residual on the current grid level after the ith iteration on this level, and R_0 is the residual that was initially obtained on this grid level at the current global iteration. The parameter α, referred to as the termination criterion, has a default value of 0.1. Note that the above equation is also used to terminate calculations on the lowest (finest) grid level during the multigrid procedure. Thus, relaxations are continued on each grid level (including the finest grid level) until the criterion of this equation is obeyed (or until a maximum number of relaxations has been completed, in the case that the specified criterion is never achieved).

22.5.4 Full-Approximation Storage (FAS) Multigrid

FLUENT's approach to forming the multigrid grid hierarchy for FAS is simply to coalesce groups of cells on the finer grid to form coarse grid cells. Coarse grid cells are created by agglomerating the cells surrounding a node, as shown in Figure 22.5.4. Depending on the grid topology, this can result in cells with irregular shapes and variable numbers of faces. The grid levels are, however, simple to construct and are embedded, resulting in simple prolongation and relaxation operators.

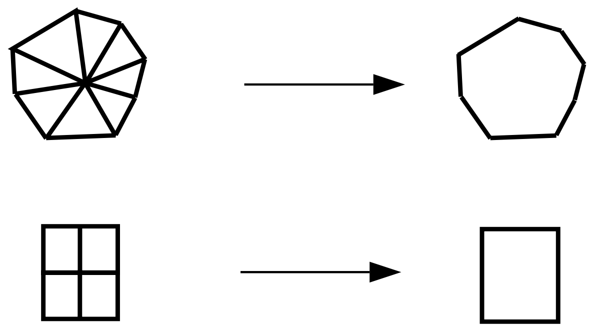


Figure 22.5.4: Node Agglomeration to Form Coarse Grid Cells

It is interesting to note that although the coarse grid cells look very irregular, the discretization cannot "see" the jaggedness in the cell faces.

The discretization uses only the area projections of the cell faces and therefore each group of “jagged” cell faces separating two irregularly-shaped cells is equivalent to a single straight line (in 2D) connecting the endpoints of the jagged segment. (In 3D, the area projections form an irregular, but continuous, geometrical shape.) This optimization decreases the memory requirement and the computation time.

FAS Restriction and Prolongation Operators

FAS requires restriction of both the fine grid solution ϕ and its residual (defect) d. The restriction operator R used to transfer the solution to the next coarser grid level is formed using a full-approximation scheme [26]. That is, the solution for a coarse cell is obtained by taking the volume average of the solution values in the embedded fine grid cells. Residuals for the coarse grid cell are obtained by summing the residuals in the embedded fine grid cells.

The prolongation operator P used to transfer corrections up to the fine level is constructed to simply set the fine grid correction to the associated coarse grid value.

The coarse grid corrections ψ^H, which are brought up from the coarse level and applied to the fine level solution, are computed from the difference between the solution calculated on the coarse level ϕ^H and the initial solution restricted down to the coarse level $R\phi$. Thus correction of the fine level solution becomes

$$\phi^{\text{new}} = \phi + P\left(\phi^H - R\phi\right) \qquad (22.5\text{-}16)$$

FAS Coarse Level Operator

The FAS coarse grid operator A^H is simply that which results from a re-discretization of the governing equations on the coarse level mesh. Since the discretized equations presented in Sections 22.2 and 22.4 place no restrictions on the number of faces that make up a cell, there is no problem in performing this re-discretization on the coarse grids composed of irregularly shaped cells.

There is some loss of accuracy when the finite-volume scheme is used on the irregular coarse grid cells, but the accuracy of the multigrid solution is determined solely by the finest grid and is therefore not affected by the coarse grid discretization.

In order to preserve accuracy of the fine grid solution, the coarse level equations are modified to include source terms [100] which insure that corrections computed on the coarse grid ϕ^H will be zero if the residuals on the fine grid d^h are zero as well. Thus, the coarse grid equations are formulated as

$$A^H \phi^H + d^H = d^H(R\phi) - Rd^h \tag{22.5-17}$$

Here d^H is the coarse grid residual computed from the current coarse grid solution ϕ^H, and $d^H(R\phi)$ is the coarse grid residual computed from the restricted fine level solution $R\phi$. Initially, these two terms will be the same (because initially we have $\phi^H = R\phi$) and cancel from the equation, leaving

$$A^H \phi^H = -Rd^h \tag{22.5-18}$$

So there will be no coarse level correction when the fine grid residual d^h is zero.

22.6 Overview of How to Use the Solver

Once you have defined your model and specified which solver you want to use (see Section 1.6), you are ready to run the solver. The following steps outline a general procedure you can follow:

1. Choose the discretization scheme and, for the segregated solver, the pressure interpolation scheme (see Section 22.7).
2. (segregated solver only) Select the pressure-velocity coupling method (see Section 22.8).
3. Set the under-relaxation factors (see Section 22.9).
4. (coupled explicit solver only) Turn on FAS multigrid (see Section 22.11).
5. Make any additional modifications to the solver settings that are suggested in the chapters or sections that describe the models you are using.
6. Initialize the solution (see Section 22.13).
7. Enable the appropriate solution monitors (see Section 22.16).
8. Start calculating (see Section 22.14 for steady-state calculations, or Section 22.15 for time-dependent calculations).
9. If you have convergence trouble, try one of the methods discussed in Section 22.19.

The default settings for the first three items listed above are suitable for most problems and need not be changed. The following sections outline how these and other solution parameters can be changed, and when you may wish to change them.

22.7 Choosing the Discretization Scheme

FLUENT allows you to choose the discretization scheme for the convection terms of each governing equation. (Second-order accuracy is automatically used for the viscous terms.) When the segregated solver is used, all equations are, by default, solved using the first-order upwind discretization for convection. When one of the coupled solvers is used, the flow equations are solved using the second-order scheme by default, and the other equations use the first-order scheme by default. See Section 22.2 for a complete description of the discretization schemes available in FLUENT.

In addition, when you use the segregated solver, you can specify the pressure interpolation scheme. See Section 22.3.1 for a description of the pressure interpolation schemes available in FLUENT.

22.7.1 First Order vs. Second Order

When the flow is aligned with the grid (e.g., laminar flow in a rectangular duct modeled with a quadrilateral or hexahedral grid) the first-order upwind discretization may be acceptable. When the flow is not aligned with the grid (i.e., when it crosses the grid lines obliquely), however, first-order convective discretization increases the numerical discretization error (numerical diffusion). For triangular and tetrahedral grids, since the flow is never aligned with the grid, you will generally obtain more accurate results by using the second-order discretization. For quad/hex grids, you will also obtain better results using the second-order discretization, especially for complex flows.

In summary, while the first-order discretization generally yields better convergence than the second-order scheme, it generally will yield less accurate results, especially on tri/tet grids. See Section 22.19 for information about controlling convergence.

For most cases, you will be able to use the second-order scheme from the start of the calculation. In some cases, however, you may need to start with the first-order scheme and then switch to the second-order scheme after a few iterations. For example, if you are running a high-Mach-number flow calculation that has an initial solution much different than the expected final solution, you will usually need to perform a few iterations with the first-order scheme and then turn on the second-order scheme and continue the calculation to convergence.

For a simple flow that is aligned with the grid (e.g., laminar flow in a rectangular duct modeled with a quadrilateral or hexahedral grid), the numerical diffusion will be naturally low, so you can generally use the first-order scheme instead of the second-order scheme without any significant loss of accuracy.

Finally, if you run into convergence difficulties with the second-order scheme, you should try the first-order scheme instead.

22.7.2 Other Discretization Schemes

The QUICK discretization scheme may provide better accuracy than the second-order scheme for rotating or swirling flows solved on quadrilateral or hexahedral meshes. In general, however, the second-order scheme is sufficient and the QUICK scheme will not provide significant improvements in accuracy.

A power law scheme is also available, but it will generally yield the same accuracy as the first-order scheme.

The central-differencing scheme is available only when you are using the LES turbulence model, and it should be used only when the mesh spacing is fine enough so that the magnitude of the local Peclet number (Equation 22.2-5) is less than 1.

22.7.3 Choosing the Pressure Interpolation Scheme

As discussed in Section 22.3.1, a number of pressure interpolation schemes are available when the segregated solver is used in FLUENT. For most cases the "standard" scheme is acceptable, but some types of models may benefit from one of the other schemes:

- For problems involving large body forces, the body-force-weighted scheme is recommended.

- For flows with high swirl numbers, high-Rayleigh-number natural convection, high-speed rotating flows, flows involving porous media, and flows in strongly curved domains, use the PRESTO! scheme.
- For compressible flows, the second-order scheme is recommended.
- Use the second-order scheme for improved accuracy when one of the other schemes is not applicable.

! The second-order scheme cannot be used with porous media or with the VOF or mixture model for multiphase flow.

Note that you will not specify the pressure interpolation scheme if you are using the Eulerian multiphase model. FLUENT will use the solution method described in Section 20.4.8 for Eulerian multiphase calculations.

22.7.4 Choosing the Density Interpolation Scheme

As discussed in Section 22.3.2, three density interpolation schemes are available when the segregated solver is used to solve a single-phase compressible flow.

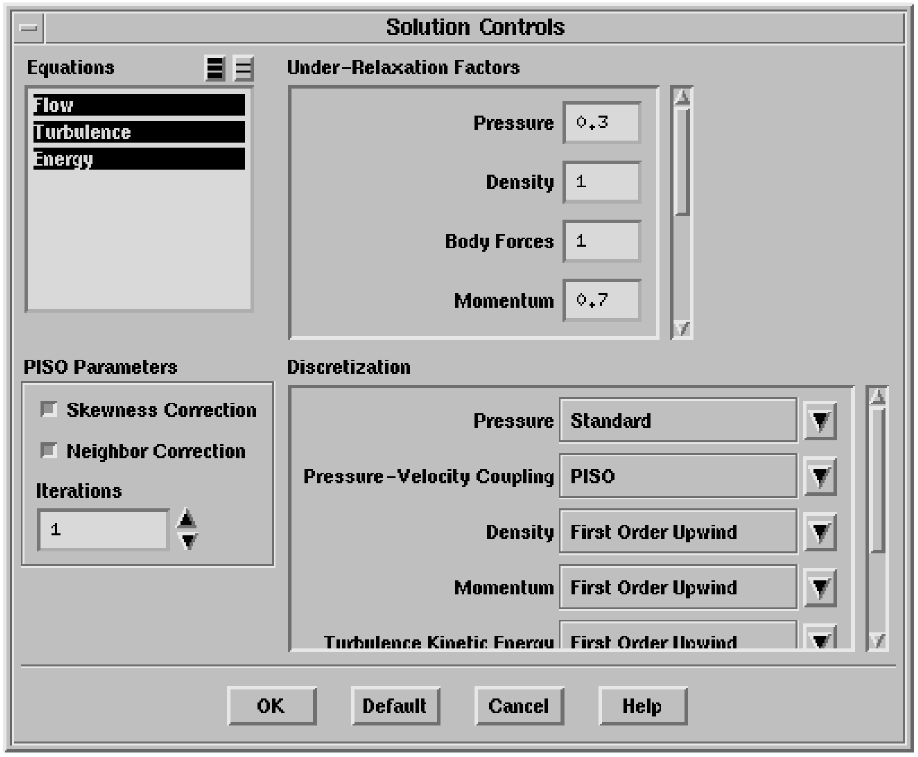
The first-order upwind scheme (the default) provides stability for the discretization of the pressure-correction equation, and gives good results for most classes of flows. If you are calculating a compressible flow with shocks, the first-order upwind scheme may tend to smooth the shocks; you should use the second-order-upwind or QUICK scheme for such flows. For compressible flows with shocks, using the QUICK scheme for all variables, including density, is highly recommended for quadrilateral, hexahedral, or hybrid meshes.

! Note that you will not be able to specify the density interpolation scheme for a compressible multiphase calculation; the first-order upwind scheme will be used for the compressible phase.

22.7.5 User Inputs

You can specify the discretization scheme and, for the segregated solver, the pressure interpolation scheme in the Solution Controls panel (Figure 22.7.1).

Solve ⟶ Controls ⟶Solution...



Equations
Flow
Turbulence
Energy

Under-Relaxation Factors

Pressure	0.3
Density	1
Body Forces	1
Momentum	0.7

PISO Parameters
Skewness Correction
Neighbor Correction
Iterations 1

Discretization

Pressure	Standard
Pressure-Velocity Coupling	PISO
Density	First Order Upwind
Momentum	First Order Upwind
Turbulence Kinetic Energy	First Order Upwind

Figure 22.7.1: The Solution Controls Panel for the Segregated Solver

For each scalar equation listed under Discretization (Momentum, Energy, Turbulence Kinetic Energy, etc. for the segregated solver or Turbulence Kinetic Energy, Turbulence Dissipation Rate, etc. for the coupled solvers) you can choose First Order Upwind, Second Order Upwind, Power Law, QUICK, or (if you are using the LES turbulence model) Central Differencing in the adjacent drop-down list. For the coupled solvers, you can choose either First Order Upwind or Second Order Upwind for the Flow equations (which include momentum and energy). Note that the panel shown in Figure 22.7.1 is for the segregated solver.

If you are using the segregated solver, select the pressure interpolation scheme under Discretization in the drop-down list next to Pressure. You can choose Standard, Linear, Second Order, Body Force Weighted, or PRESTO!. If you are using the segregated solver and your flow is compressible (i.e., you are using the ideal gas law for density), select the density interpolation scheme under Discretization in the drop-down list next to Density. You can choose First Order Upwind, Second Order Upwind, or QUICK. (Note that Density will not appear for incompressible or multiphase flows.)

If you change the settings for Discretization, but you then want to return to FLUENT's default settings, you can click on the Default button. FLUENT will change the settings to the defaults and the Default button will become the Reset button. To get your settings back again, you can click on the Reset button.

22.8 Choosing the Pressure-Velocity Coupling Method

FLUENT provides three methods for pressure-velocity coupling in the segregated solver: SIMPLE, SIMPLEC, and PISO. Steady-state calculations will generally use SIMPLE or SIMPLEC, while PISO is recommended for transient calculations. PISO may also be useful for steady-state and transient calculations on highly skewed meshes.

! Pressure-velocity coupling is relevant only for the segregated solver; you will not specify it for the coupled solvers.

22.8.1 SIMPLE vs. SIMPLEC

In FLUENT, both the standard SIMPLE algorithm and the SIMPLEC (SIMPLE-Consistent) algorithm are available. SIMPLE is the default, but many problems will benefit from the use of SIMPLEC, particularly because of the increased under-relaxation that can be applied, as described below.

For relatively uncomplicated problems (laminar flows with no additional models activated) in which convergence is limited by the pressure-velocity coupling, you can often obtain a converged solution more quickly using SIMPLEC. With SIMPLEC, the pressure-correction under-relaxation factor is generally set to 1.0, which aids in convergence speed-up. In some problems, however, increasing the pressure-correction under-relaxation to 1.0 can lead to instability. For such cases, you will need to use a more conservative under-relaxation value or use the SIMPLE algorithm. For complicated flows involving turbulence and/or additional physical models, SIMPLEC will improve convergence only if it is being limited by the pressure-velocity coupling. Often it will be one of the additional modeling parameters that limits convergence; in this case, SIMPLE and SIMPLEC will give similar convergence rates.

22.8.2 PISO

The PISO algorithm (see Section 22.3.3) with neighbor correction is highly recommended for all transient flow calculations, especially when you want to use a large time step. (For problems that use the LES turbulence model, which usually requires small time steps, using PISO may result in increased computational expense, so SIMPLE or SIMPLEC should be considered instead.) PISO can maintain a stable calculation with a larger time step and an under-relaxation factor of 1.0 for both momentum and pressure. For steady-state problems, PISO with neighbor correction does not provide any noticeable advantage over SIMPLE or SIMPLEC with optimal under-relaxation factors.

PISO with skewness correction is recommended for both steady-state and transient calculations on meshes with a high degree of distortion.

When you use PISO neighbor correction, under-relaxation factors of 1.0 or near 1.0 are recommended for all equations. If you use just the PISO skewness correction for highly-distorted meshes, set the under-relaxation factors for momentum and pressure so that they sum to 1 (e.g., 0.3 for pressure and 0.7 for momentum). If you use both PISO methods, follow the under-relaxation recommendations for PISO neighbor correction, above.

22.8.3 User Inputs

You can specify the pressure-velocity coupling method in the Solution Controls panel (Figure 22.7.1).

$\boxed{\text{Solve}}\longrightarrow\boxed{\text{Controls}}\longrightarrow$Solution...

Choose SIMPLE, SIMPLEC, or PISO in the Pressure-Velocity Coupling drop-down list under Discretization.

If you choose PISO, the panel will expand to show the PISO Parameters. By default, both Skewness Correction and Neighbor Correction are on. If you want to use just neighbor correction or just skewness correction, you can turn off the appropriate option. The default number of Iterations is set to 1; you should not need to change this value.

22.9 Setting Under-Relaxation Factors

As discussed in Section 22.2.7, the segregated solver uses under-relaxation to control the update of computed variables at each iteration. This means that all equations solved using the segregated solver, *including the non-coupled equations solved by the coupled solvers* (turbulence and other scalars, as discussed in Section 22.1.2), will have under-relaxation factors associated with them.

In FLUENT, the default under-relaxation parameters for all variables are set to values that are near optimal for the largest possible number of cases. These values are suitable for many problems, but for some particularly nonlinear problems (e.g., some turbulent flows or high-Rayleigh-number natural-convection problems) it is prudent to reduce the under-relaxation factors initially.

It is good practice to begin a calculation using the default under-relaxation factors. If the residuals continue to increase after the first 4 or 5 iterations, you should reduce the under-relaxation factors.

Occasionally, you may make changes in the under-relaxation factors and resume your calculation, only to find that the residuals begin to increase. This often results from increasing the under-relaxation factors too much. A cautious approach is to save a data file before making any changes to the under-relaxation factors, and to give the solution algorithm a few iterations to adjust to the new parameters. Typically, an increase in the under-relaxation factors brings about a slight increase in the residuals, but these increases usually disappear as the solution progresses. If the residuals jump by a few orders of magnitude, you should consider halting the calculation and returning to the last good data file saved.

Note that viscosity and density are under-relaxed from iteration to iteration. Also, if the enthalpy equation is solved directly instead of the temperature equation (i.e., for non-premixed combustion calculations), the update of temperature based on enthalpy will be under-relaxed. To see the default under-relaxation factors, you can click on the Default button in the Solution Controls panel.

For most flows, the default under-relaxation factors do not usually require modification. If unstable or divergent behavior is observed, however, you need to reduce the under-relaxation factors for pressure, momentum, k, and ϵ from their default values to about 0.2, 0.5, 0.5, and 0.5. (It is usually not necessary to reduce the pressure under-relaxation for SIMPLEC.) In problems where density is strongly coupled with temperature, as in very-high-Rayleigh-number natural- or mixed-convection flows, it is wise to also under-relax the temperature equation and/or density (i.e., use an under-relaxation factor less than 1.0). Conversely, when temperature is not coupled with the momentum equations (or when it is weakly coupled), as in flows with constant density, the under-relaxation factor for temperature can be set to 1.0.

For other scalar equations (e.g., swirl, species, mixture fraction and variance) the default under-relaxation may be too aggressive for some problems, especially at the start of the calculation. You may wish to reduce the factors to 0.8 to facilitate convergence.

User Inputs

You can modify the under-relaxation factors in the Solution Controls panel (Figure 22.7.1).

$\boxed{\text{Solve}} \longrightarrow \boxed{\text{Controls}} \longrightarrow$ Solution...

You can set the under-relaxation factor for each equation in the field next to its name under Under-Relaxation Factors.

! If you are using the segregated solver, all equations will have an associated under-relaxation factor. If you are using one of the coupled solvers, only those equations that are solved sequentially (see Section 22.1.2) will have under-relaxation factors.

If you change under-relaxation factors, but you then want to return to FLUENT's default settings, you can click on the Default button. FLUENT will change the factors to the default values and the Default button will become the Reset button. To get your settings back again, you can click on the Reset button.

22.10 Changing the Courant Number

For FLUENT's coupled solvers, the main control over the time-stepping scheme is the Courant number (CFL). The time step is proportional to the CFL, as defined in Equation 22.4-13.

Linear stability theory determines a range of permissible values for the CFL (i.e., the range of values for which a given numerical scheme will remain stable). When you specify a permissible CFL value, FLUENT will compute an appropriate time step using Equation 22.4-13. In general, taking larger time steps leads to faster convergence, so it is advantageous to set the CFL as large as possible (within the permissible range).

The stability limits of the coupled implicit and explicit solvers are significantly different. The explicit solver has a more limited range and requires lower CFL settings than does the coupled implicit solver. Appropriate choices of CFL for the two solvers are discussed below.

22.10.1 Courant Numbers for the Coupled Explicit Solver

Linear stability analysis shows that the maximum allowable CFL for the multi-stage scheme used in the coupled explicit solver will depend on the number of stages used and how often the dissipation and viscous terms are updated (see Section 22.19.5). But in general, you can assume that the multi-stage scheme is stable for Courant numbers up to 2.5. This stability limit is often lower in practice because of nonlinearities in the governing equations.

The default CFL for the coupled explicit solver is 1.0, but you may be able to increase it for some 2D problems. You should generally not use a value higher than 2.0.

If your solution is diverging, i.e., if residuals are rising very rapidly, and your problem is properly set up and initialized, this is usually a good sign that the Courant number needs to be lowered. Depending on the severity of the startup conditions, you may need to decrease the CFL to a value as low as 0.1 to 0.5 to get started. Once the startup transients are reduced you can start increasing the Courant number again.

22.10.2 Courant Numbers for the Coupled Implicit Solver

Linear stability theory shows that the point Gauss-Seidel scheme used in the coupled implicit solver is unconditionally stable. However, as with the explicit solver, nonlinearities in the governing equations will often limit stability.

The default CFL for the coupled implicit solver is 5.0. It is often possible to increase the CFL to 10, 20, 100, or even higher, depending on the complexity of your problem. You may find that a lower CFL is required during startup (when changes in the solution are highly nonlinear), but it can be increased as the solution progresses.

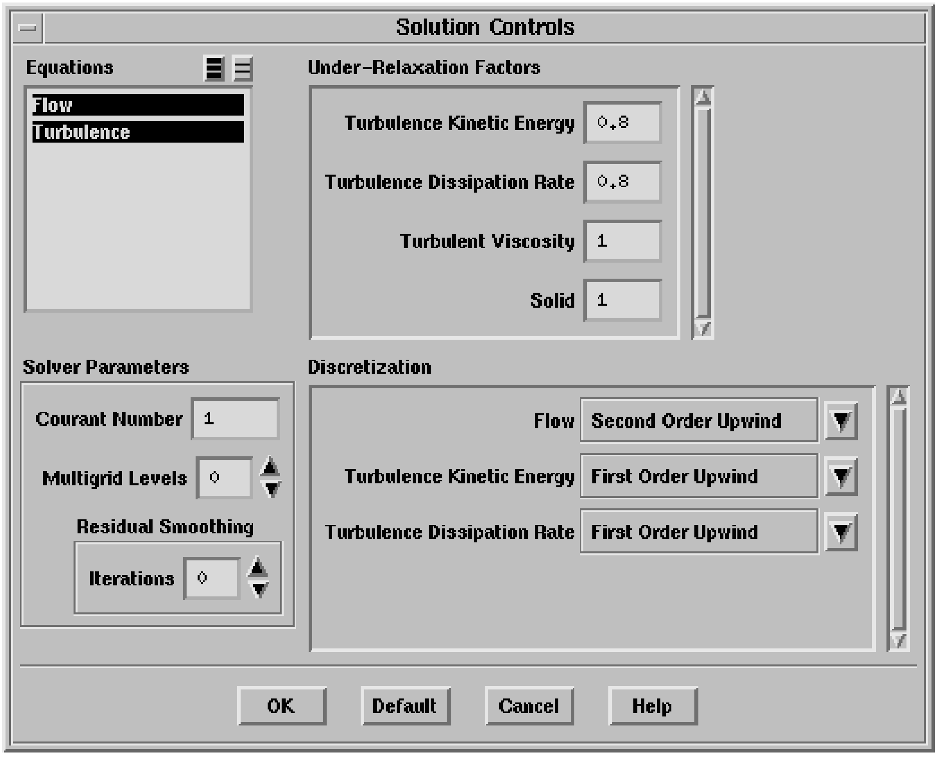
The coupled AMG solver has the capability to detect divergence of the multigrid cycles within a given iteration. If this happens, it will automatically reduce the CFL and perform the iteration again, and a message will be printed to the screen. Five attempts are made to complete the iteration successfully. Upon successful completion of the current iteration the CFL is returned to its original value and the iteration procedure proceeds as required.

22.10.3 User Inputs

The Courant number is set in the Solution Controls panel (Figure 22.10.1).

$\boxed{\text{Solve}} \longrightarrow \boxed{\text{Controls}} \longrightarrow$ Solution...

Enter the value for Courant Number under Solver Parameters. (Note that the panel shown in Figure 22.10.1 is for the coupled explicit solver. For the coupled implicit solver, the Courant Number is the only item that will appear under Solver Parameters.)



Turbulence Kinetic Energy	0.8
Turbulence Dissipation Rate	0.8
Turbulent Viscosity	1
Solid	1

Courant Number	1
Multigrid Levels	0
Iterations	0

Flow	Second Order Upwind
Turbulence Kinetic Energy	First Order Upwind
Turbulence Dissipation Rate	First Order Upwind

Figure 22.10.1: The Solution Controls Panel for the Coupled Explicit Solver

When you select the coupled explicit solver in the Solver panel, FLUENT will automatically set the Courant Number to 1; when you select the coupled implicit solver, the Courant Number will be changed to 5 automatically.

22.11 Turning On FAS Multigrid

As discussed in Section 22.5, FAS multigrid is an optional component of the coupled explicit solver. (AMG multigrid is always on, by default.) Since nearly all coupled explicit calculations will benefit from the use of the FAS multigrid convergence accelerator, you should generally set a non-zero number of coarse grid levels before beginning the calculation. For most problems, this will be the only FAS multigrid parameter you will need to set. Should you encounter convergence difficulties, consider applying one of the methods discussed in Section 22.19.4.

! Note that you cannot use FAS multigrid with explicit time stepping (described in Section 22.2.8) because the coarse grid corrections will destroy the time accuracy of the fine grid solution.

22.11.1 Setting Coarse Grid Levels

As discussed in Section 22.5.4, FAS multigrid solves on successively coarser grids and then transfers corrections to the solution back up to the original fine grid, thus increasing the propagation speed of the solution and speeding convergence. The most basic way you can control the multigrid solver is by specifying the number of coarse grid levels to be used.

As explained in Section 22.5.4, the coarse grid levels are formed by agglomerating a group of adjacent “fine” cells into a single “coarse” cell. The optimal number of grid levels is therefore problem-dependent. For most problems, you can start out with 4 or 5 levels. For large 3D problems, you may want to add more levels (although memory restrictions may prevent you from using more levels, since each coarse grid level requires additional memory). If you believe that multigrid is causing convergence trouble, you can decrease the number of levels.

If FLUENT reaches a coarse grid with one cell before creating as many levels as you requested, it will simply stop there. That is, if you request 5 levels, and level 4 has only 1 cell, FLUENT will create only 4 levels, since levels 4 and 5 would be the same.

To specify the number of grid levels you want, set the number of Multigrid Levels in the Solution Controls panel (Figure 22.10.1) under Solver

Parameters.

$\boxed{\text{Solve}}\longrightarrow\boxed{\text{Controls}}\longrightarrow$Solution...

You can also set the Max Coarse Levels under FAS Multigrid Controls in the Multigrid Controls panel.

$\boxed{\text{Solve}}\longrightarrow\boxed{\text{Controls}}\longrightarrow$Multigrid...

Changing the number of coarse grid levels in one panel will automatically update the number shown in the other.

Coarse grid levels are created when you first begin iterating. If you want to check how many cells are in each level, request one iteration and then use the Grid/Info/Size menu item (described in Section 5.6.1) to list the size of each grid level. If you are satisfied, you can continue the calculation; if not, you can change the number of coarse grid levels and check again.

For most problems, you will not need to modify any additional multigrid parameters once you have settled on an appropriate number of coarse grid levels. You can simply continue your calculation until convergence.

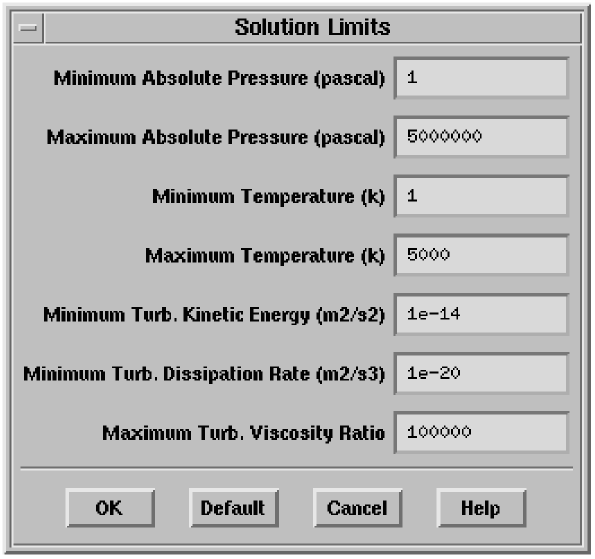
22.12 Setting Solution Limits

In order to keep the solution stable under extreme conditions, FLUENT provides limits that keep the solution within an acceptable range. You can control these limits with the Solution Limits panel (Figure 22.12.1).

$\boxed{\text{Solve}}\longrightarrow\boxed{\text{Controls}}\longrightarrow$Limits...

FLUENT applies limiting values for pressure, temperature, and turbulence quantities. The purpose of these limits is to keep the absolute pressure or the temperature from becoming 0, negative, or excessively large during the calculation, and to keep the turbulence quantities from becoming excessive. FLUENT also puts a limit on the rate of reduction of temperature to prevent it from becoming 0 or negative.

! Typically, you will not need to change the default solution limits. If pressure, temperature, or turbulence quantities are being reset to the limiting value repeatedly (as indicated by the appropriate warning messages in



Minimum Absolute Pressure (pascal)	1
Maximum Absolute Pressure (pascal)	5000000
Minimum Temperature (k)	1
Maximum Temperature (k)	5000
Minimum Turb. Kinetic Energy (m2/s2)	1e-14
Minimum Turb. Dissipation Rate (m2/s3)	1e-20
Maximum Turb. Viscosity Ratio	100000

Figure 22.12.1: The Solution Limits Panel

the console window), you should check the dimensions, boundary conditions, and properties to be sure that the problem is set up correctly and try to determine why the variable in question is getting so close to zero or so large. You can use the "marking" feature (used to mark cells for adaption) to identify which cells have a value equal to the limit. (Use the Iso-Value Adaption panel, as described in Section 23.5.) In very rare cases, you may need to change the solution limits, but only do so if you are sure that you understand the reason for the solver's unusual behavior. (For example, you may know that the temperature in your domain will exceed 5000 K. Be sure that any temperature-dependent properties are appropriately defined for high temperatures if you increase the maximum temperature limit.)

Limiting the Values of Solution Variables

The limiting minimum and maximum values for absolute pressure are shown in the Minimum and Maximum Absolute Pressure fields. If the FLUENT calculation predicts a value less than the Minimum Absolute Pressure or greater than the Maximum Absolute Pressure, the corresponding limiting value will be used instead. Similarly, the Minimum and Maximum Temperature are limiting values for energy calculations.

The Maximum Turb. Viscosity Ratio and the Minimum Turb. Kinetic Energy are limiting values for turbulent calculations. If the calculation predicts a k value less than the Minimum Turb. Kinetic Energy, the limiting value will be used instead. For the viscosity ratio limit, FLUENT uses the limiting maximum value of turbulent viscosity ($C_\mu k^2/\epsilon$) in the flow field relative to the laminar viscosity. If the ratio calculated by FLUENT exceeds the limiting value, the ratio is set to the limiting value by limiting ϵ to the necessary value.

Limiting the Reduction Rate for Temperature

In FLUENT's coupled solvers, the rate of reduction of temperature is controlled by the Positivity Rate Limit. The default value of 0.2, for example, means that temperature is not allowed to decrease by more than 20% of its previous value from one iteration to the next. If the temperature change exceeds this limit, the time step in that cell is reduced to bring the change back into range and a "time step reduced" warning is printed. (This reduced time step will be used for the solution of all variables in the cell, not just for temperature.) Rapid reduction of temperature is an indication that the temperature may become negative. Repeated "time step reduced" warnings should alert you that something is wrong in your problem setup. (If the warning messages stop appearing, the calculation may have "recovered" from the time-step reduction.)

Resetting Solution Limits

If you change and save the value of one of the solution limits, but you then want to return to the default limits set by FLUENT, you can reopen the Solution Limits panel and click on the Default button. FLUENT will change the values to the defaults and the Default button will become the Reset button. To get your values back again, you can click on the Reset button.

22.13 Initializing the Solution

Before starting your CFD simulation, you must provide FLUENT with an initial "guess" for the solution flow field. In many cases, you must take extra care to provide an initial solution that will allow the desired final solution to be attained. A real-life supersonic wind tunnel, for example, will not "start" if the back pressure is simply lowered to its operating value; the flow will choke at the tunnel throat and will not transition to supersonic. The same holds true for a numerical simulation: the flow must be initialized to a supersonic flow or it will simply choke and remain subsonic.

There are two methods for initializing the solution:

- Initialize the entire flow field (in all cells).
- Patch values or functions for selected flow variables in selected cell zones or "registers" of cells. (Registers are created with the same functions that are used to mark cells for adaption.)

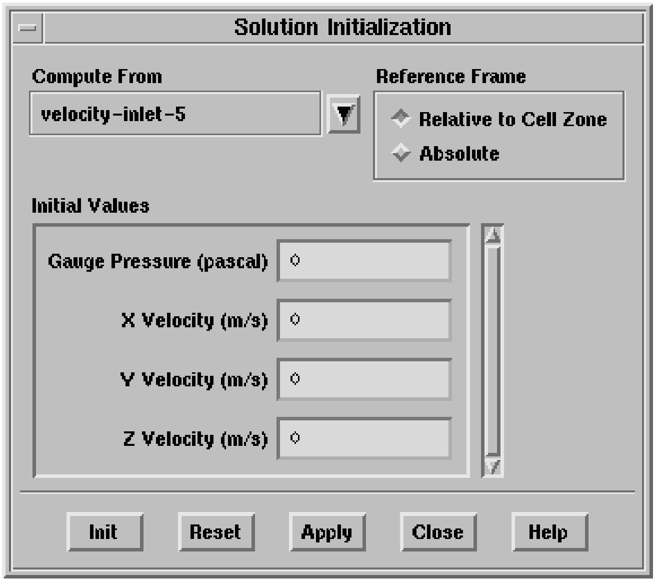
! Before patching initial values in selected cells, you must first initialize the entire flow field. You can then patch the new values over the initialized values for selected variables.

22.13.1 Initializing the Entire Flow Field

Before you start your calculations *or* patch initial values for selected variables in selected cells (Section 22.13.2) you must initialize the flow field in the entire domain. The Solution Initialization panel (Figure 22.13.1) allows you to set initial values for the flow variables and initialize the solution using these values.

$\boxed{\text{Solve}}\longrightarrow\boxed{\text{Initialize}}\longrightarrow$Initialize...

You can compute the values from information in a specified zone, enter them manually, or have the solver compute average values based on all zones. You can also indicate whether the specified values for velocities are absolute or relative to the velocity in each cell zone. The steps for initialization are as follows:



Compute From: velocity-inlet-5

Reference Frame: [✓ Relative to Cell Zone] Absolute

Initial Values

Gauge Pressure (pascal)	0
X Velocity (m/s)	0
Y Velocity (m/s)	0
Z Velocity (m/s)	0

Figure 22.13.1: The Solution Initialization Panel

1. Set the initial values:

   - To initialize the flow field using the values set for a particular zone, select the zone name in the Compute From drop-down list. All values under the Initial Values heading will automatically be computed and updated based on the conditions defined at the selected zone.
   - To initialize the flow field using computed average values, select all-zones in the Compute From drop-down list. FLUENT will compute and update the Initial Values based on the conditions defined at all boundary zones.
   - If you wish to change one or more of the values, you can enter new values manually in the fields next to the appropriate variables. If you prefer to enter all values manually, you can do so without selecting a zone in the Compute From list.

2. If your problem involves moving reference frames or sliding meshes, indicate whether the initial velocities are absolute velocities or velocities relative to the motion of each cell zone by selecting Absolute or Relative to Cell Zone under Reference Frame. (If no zone motion occurs in the problem, the two options are equivalent.) The default reference frame for velocity initialization in FLUENT is relative. If the solution in most of your domain is rotating, using the relative option may be better than using the absolute option.

3. Once you are satisfied with the Initial Values displayed in the panel, you can click on the Init button to initialize the flow field. If solution data already exist (i.e., if you have already performed some calculations or initialized the solution), you must confirm that it is OK to overwrite those data.

Saving and Resetting Initial Values

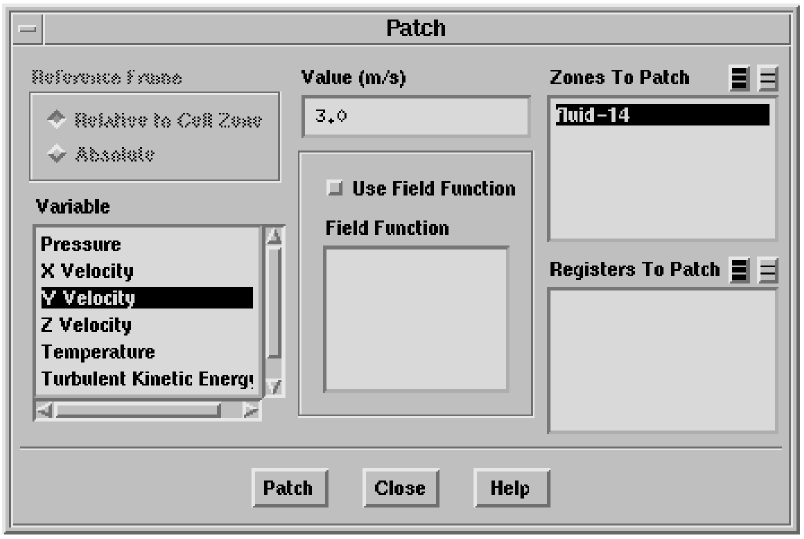
When you initialize the solution by clicking on Init, the initial values will also be saved; should you need to reinitialize the solution later, you will find the correct values in the panel when you reopen it. If you wish to define initial values now, but you are not yet ready to initialize the solution, you can follow the instructions above for setting the values and then click Apply instead of Init. This will save the currently displayed values without initializing the solution. You can return to the panel later on and perform the initialization.

If you accidentally select the wrong zone from the Compute From list or manually set a value incorrectly, you can use the Reset button to reset all fields to their "saved" values. Values are saved each time the panel is opened, before Compute From is executed, and after Init or Apply is executed.

22.13.2 Patching Values in Selected Cells

Once you have initialized (or calculated) the entire flow field, you may patch different values for particular variables into different cells. If you have multiple fluid zones, for example, you may want to patch a different temperature in each one. You can also choose to patch a custom field function (defined using the Custom Field Function Calculator panel) instead of a constant value. If you are patching velocities, you can indicate whether the specified values are absolute velocities or velocities relative to the cell zone's velocity. All patching operations are performed with the Patch panel (Figure 22.13.2).

$\boxed{\text{Solve}}\longrightarrow\boxed{\text{Initialize}}\longrightarrow$Patch...



Value (m/s): 3.0

Use Field Function

Field Function

Variable: Pressure, X Velocity, Y Velocity (selected), Z Velocity, Temperature, Turbulent Kinetic Energy

Zones To Patch: fluid-14 (selected)

Registers To Patch

Figure 22.13.2: The Patch Panel

1. Select the variable to be patched in the Variable list.
2. In the Zones To Patch and/or Registers To Patch lists, choose the zone(s) and/or register(s) for which you want to patch a value for the selected variable.

3. If you wish to patch a constant value, simply enter that value in the Value field. If you want to patch a previously-defined field function, turn on the Use Field Function option and select the appropriate function in the Field Function list.

4. If you selected a velocity in the Variable list, and your problem involves moving reference frames or sliding meshes, indicate whether the patched velocities are absolute velocities or velocities relative to the motion of each cell zone by selecting Absolute or Relative to Cell Zone under Reference Frame. (If no zone motion occurs in the problem, the two options are equivalent.) The default reference frame for velocity patching in FLUENT is relative. If the solution in most of your domain is rotating, using the relative option may be better than using the absolute option.

5. Click on the Patch button to update the flow-field data. (Note that patching will have no effect on the iteration or time-step count.)

Using Registers

The ability to patch values in cell registers gives you the flexibility to patch different values within a single cell zone. For example, you may want to patch a certain value for temperature only in fluid cells with a particular range of concentrations for one species. You can create a cell register (basically a list of cells) using the functions that are used to mark cells for adaption. These functions allow you to mark cells based on physical location, cell volume, gradient or isovalue of a particular variable, and other parameters. See Chapter 23 for information about marking cells for adaption. Section 23.9 provides information about manipulating different registers to create new ones. Once you have created a register, you can patch values in it as described above.

Using Field Functions

By defining your own field function using the Custom Field Function Calculator panel, you can patch a non-constant value in selected cells. For example, you may want to patch varying species mass fractions throughout a fluid region. To use this feature, simply create the function as described in Section 27.5, and then perform the function-patching operation in the Patch panel, as described above.

Using Patching Later in the Solution Process

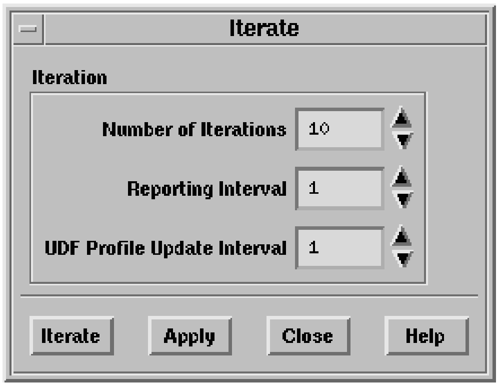
Since patching affects only the variables for which you choose to change the value, leaving the rest of the flow field intact, you can use it later in the solution process without losing calculated data. (Initialization, on the other hand, resets all data to the initial values.) For example, you might want to start a combustion calculation from a cold-flow solution. You can simply read in (or calculate) the cold-flow data, patch a high temperature in the appropriate cells, and continue the calculation.

Patching can also be useful when you are solving a problem using a step-by-step technique, as described in Section 22.19.2.

22.14 Performing Steady-State Calculations

For steady-state calculations, you will request the start of the solution process using the Iterate panel (Figure 22.14.1).

$\boxed{\text{Solve}}\longrightarrow$Iterate...



Number of Iterations: 10
Reporting Interval: 1
UDF Profile Update Interval: 1

Figure 22.14.1: The Iterate Panel

In this panel, you will supply the number of additional iterations to be performed in the Number of Iterations field. (For unsteady calculation inputs, see Section 22.15.1.) If no calculations have been performed yet, FLUENT will begin calculations starting at iteration 1, using the initial solution. If you are starting from current solution data, FLUENT will begin at the last iteration performed, using the current solution data as its starting point.

By default, FLUENT will update the convergence monitors (described in Section 22.16) after each iteration. If you increase the Reporting Interval from the default of 1 you can get reports less frequently. For example, if you set the Reporting Interval to 2, the monitors will print or plot reports at every other iteration. Note that the Reporting Interval also specifies how often FLUENT should check if the solution is converged. For example, if your solution converges after 40 iterations, but your Reporting Interval is set to 50, FLUENT will continue the calculation for an extra 10 iterations before checking for (and finding) convergence.

When you click on the Iterate button, FLUENT will begin to calculate. During iteration, a Working dialog is displayed. Clicking on the Cancel button or typing `<Control-C>` in the FLUENT console window will interrupt the iteration, as soon as it is safe to stop. (See below for more details.)

Updating UDF Profiles

If you have used a user-defined function (UDF) to define any boundary conditions, properties, etc., you can control the frequency with which the function is updated by modifying the value of the UDF Profile Update Interval. If UDF Profile Update Interval is set to n, the function will be updated after every n iterations.

By default, the UDF Profile Update Interval is set to 1. You might want to increase this value if your profile computation is expensive. See the separate UDF Manual for details about creating and using UDFs.

Interrupting Iterations

As mentioned above, you can interrupt the calculation by clicking on the Cancel button in the Working dialog box that appears while the solver is calculating. In addition, on most, but not all, computer systems you will be able to interrupt calculations using a control sequence, usually `<Control-C>`. This allows you to stop the calculation process before proceeding with the remainder of the requested iterations.

Resetting Data

After you have performed some iterations, if you decide to start over again from the first iteration (e.g., after making some changes to the problem setup), you can reinitialize the solution using the Solution Initialization panel, as described in Section 22.13.1.

22.15 Performing Time-Dependent Calculations

FLUENT can solve the conservation equations in time-dependent form, to simulate a wide variety of time-dependent phenomena, such as

- vortex shedding and other time-periodic phenomena
- compressible filling and emptying problems
- transient heat conduction
- transient chemical mixing and reactions

Figures 22.15.1 and 22.15.2 illustrate the time-dependent vortex shedding flow pattern in the wake of a cylinder.

Activating time dependence is sometimes useful when attempting to solve steady-state problems which tend toward instability (e.g., natural convection problems in which the Rayleigh number is close to the transition region). It is possible in many cases to reach a steady-state solution by integrating the time-dependent equations.

See Section 22.2.8 for details about temporal discretization.

22.15.1 User Inputs for Time-Dependent Problems

To solve a time-dependent problem, you will follow the procedure outlined below:

1. Enable the Unsteady option in the Solver panel (Figure 22.15.3), and specify the desired Unsteady Formulation.

   $\boxed{\text{Define}}\longrightarrow\boxed{\text{Models}}\longrightarrow$Solver...

   The 1st-Order Implicit formulation is sufficient for most problems. If you need improved accuracy, you can use the 2nd-Order Implicit formulation instead. The Explicit formulation (available only if the coupled explicit solver is selected under Solver and Formulation at the top of the panel) is used primarily to capture the transient behavior of moving waves, such as shocks. See Section 22.2.8 for details.

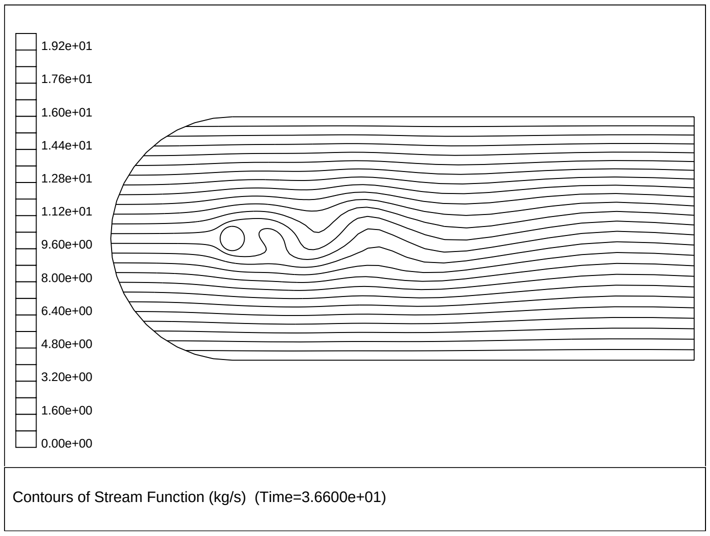


Figure 22.15.1: Time-Dependent Calculation of Vortex Shedding (t=3.66 sec)

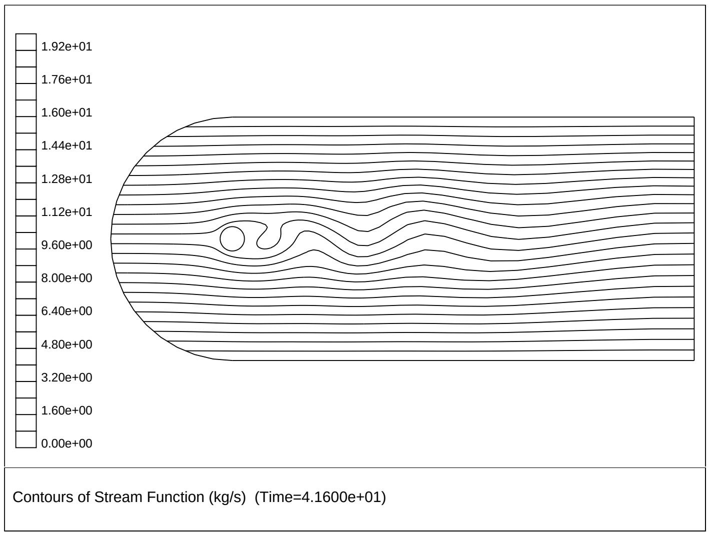
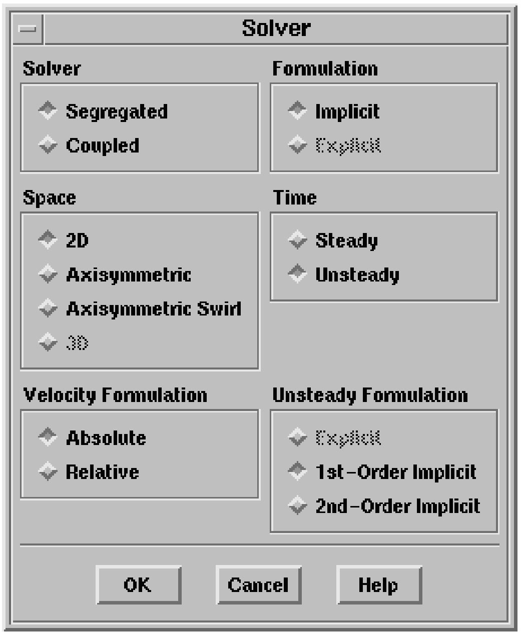


Figure 22.15.2: Time-Dependent Calculation of Vortex Shedding (t=41.6 sec)



Solver: [✓ Segregated] Coupled

Formulation: [✓ Implicit] Explicit

Space: [✓ 2D] Axisymmetric Axisymmetric Swirl 3D

Time: Steady [✓ Unsteady]

Velocity Formulation: [✓ Absolute] Relative

Unsteady Formulation: Explicit [✓ 1st-Order Implicit] 2nd-Order Implicit

Figure 22.15.3: The Solver Panel for an Unsteady Calculation

2. Define all relevant models and boundary conditions. Note that any boundary conditions specified using user-defined functions can be made to vary in time. See the separate UDF Manual for details.

3. If you are using the segregated solver, choose PISO as the Pressure-Velocity Coupling scheme under Discretization in the Solution Controls panel.

   $\boxed{\text{Solve}}\longrightarrow\boxed{\text{Controls}}\longrightarrow$Solution...

   In general, you will not need to modify the PISO Parameters from their default values. See Section 22.8.2 for more information about using the PISO algorithm.

! If you are using the LES turbulence model with small time steps, the PISO scheme may be too computationally expensive. It is therefore recommended that you use SIMPLE or SIMPLEC instead of PISO.

4. (optional) If you are using the explicit unsteady formulation *or* if you are using the adaptive time stepping method (described below and in Section 22.15.2) it is recommended that you enable the printing of the current time (for the explicit unsteady formulation) or the current time step size (for the adaptive time stepping method) at each iteration, using the Statistic Monitors panel.

   $\boxed{\text{Solve}} \longrightarrow \boxed{\text{Monitors}} \longrightarrow$ Statistic...

   Select time (for the current time) or delta_time (for the current time step size) in the Statistics list and turn on the Print option. When FLUENT prints the residuals to the console window at each iteration, it will include a column with the current time or the current time step size.

5. (optional) Use the Force Monitors panel or the Surface Monitors panel to monitor (and/or save to a file) time-varying force coefficient values or the average, mass average, integral, or flux of a field variable or function on a surface as it changes with time. See Section 22.16 for details.

6. Set the initial conditions (at time $t = 0$) using the Solution Initialization panel.

   $\boxed{\text{Solve}} \longrightarrow \boxed{\text{Initialize}} \longrightarrow$ Initialize...

   You can also read in a steady-state data file to set the initial conditions.

   $\boxed{\text{File}} \longrightarrow \boxed{\text{Read}} \longrightarrow$ Data...

7. Use the automatic saving feature to specify the file name and frequency with which case and data files should be saved during the solution process.

   $\boxed{\text{File}} \longrightarrow \boxed{\text{Write}} \longrightarrow$ Autosave...

   See Section 3.3.4 for details about the use of this feature.

   You may also want to request automatic execution of other commands using the Execute Commands panel. See Section 22.18 for details.

8. (optional) If you want to create a graphical animation of the solution over time, you can use the Solution Animation panel to set up the graphical displays that you want to use in the animation. See Section 22.17 for details.

9. (optional) If you want FLUENT to gather data for time statistics (i.e., time-averaged and root-mean-square values for solution variables) during the calculation, follow these steps:

   (a) Turn on the Data Sampling for Time Statistics option in the Iterate panel.

   $\boxed{\text{Solve}}\longrightarrow$Iterate...

   Enabling this option will allow you to display and report both the mean and the root-mean-square (RMS) values, as described in Section 22.15.3.

   ! Note that gathering data for time statistics is not meaningful inside a moving cell zone (i.e., a sliding zone in a sliding mesh problem).

   (b) Initialize the flow statistics.

   $\boxed{\texttt{solve}}\longrightarrow\boxed{\texttt{initialize}}\longrightarrow$`init-flow-statistics`

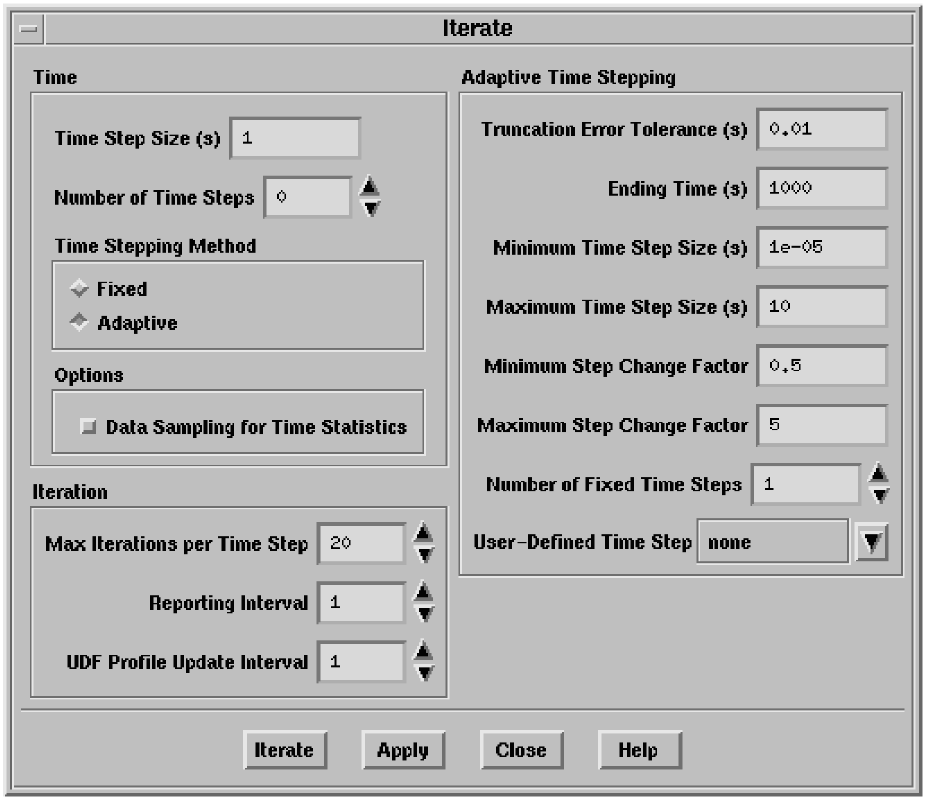
   Note that you can also use the `init-flow-statistics` text command to reset the flow statistics after you have gathered some data for time statistics. If you perform, say, 10 time steps with the Data Sampling for Time Statistics option enabled, check the results, and then continue the calculation for 10 more time steps, the time statistics will include the data gathered in the first 10 time steps unless you reinitialize the flow statistics.

10. Specify time-dependent solution parameters and start the calculation, as described below for the implicit and explicit unsteady formulations:

    - If you have chosen the 1st-Order or 2nd-Order Implicit formulation, the procedure is as follows:

      (a) Set the time-dependent solution parameters in the Iterate panel (Figure 22.15.4).

$\boxed{\text{Solve}}\longrightarrow$Iterate...



Time

Time Step Size (s) 1

Number of Time Steps 0

Time Stepping Method

Fixed [✓ Adaptive]

Options

Data Sampling for Time Statistics

Iteration

Max Iterations per Time Step 20

Reporting Interval 1

UDF Profile Update Interval 1

Adaptive Time Stepping

Truncation Error Tolerance (s) 0.01

Ending Time (s) 1000

Minimum Time Step Size (s) 1e-05

Maximum Time Step Size (s) 10

Minimum Step Change Factor 0.5

Maximum Step Change Factor 5

Number of Fixed Time Steps 1

User-Defined Time Step none

Figure 22.15.4: The Iterate Panel for Implicit Unsteady Calculations

Solution parameters for the implicit unsteady formulations are as follows:

– Max Iterations per Time Step: When FLUENT solves the time-dependent equations using the implicit formulation, iteration is necessary at each time step. This parameter sets a maximum for the number of iterations per time step. If the convergence criteria are met before this number of iterations is performed, the solution will advance to the next time step.

– Time Step Size: The time step size is the magnitude of Δt. Since the FLUENT formulation is fully implicit, there is no stability criterion that needs to be met in determining Δt. However, to model transient phenomena properly, it is necessary to set Δt at least one order of magnitude smaller than the smallest time constant in the system being modeled. A good way to judge the choice of Δt is to observe the number of iterations FLUENT needs to converge at each time step. The ideal number of iterations per time step is 10–20. If FLUENT needs substantially more, the time step is too large. If FLUENT needs only a few iterations per time step, Δt may be increased. Frequently a time-dependent problem has a very fast "startup" transient that decays rapidly. It is thus often wise to choose a conservatively small Δt for the first 5–10 time steps. Δt may then be gradually increased as the calculation proceeds.

For time-periodic calculations, you should choose the time step based on the time scale of the periodicity. For a rotor/stator model, for example, you might want 20 time steps between each blade passing. For vortex shedding, you might want 20 steps per period.

By default, the size of the time step is fixed (as indicated by the selection of Fixed under Time Stepping Method). To have FLUENT modify the size of the time step as the calculation proceeds, select Adaptive and specify the parameters under Adaptive Time Stepping in the expanded Iterate panel. See Section 22.15.2 for details.

With the Adaptive time stepping method, the value you specify for the Time Step Size will be the initial size of the time step. As the calculation proceeds, the Time Step Size shown in the Iterate panel will be the size of the *current* time step.

(b) Specify the desired Number of Time Steps in the Iterate panel and click Iterate.

As it calculates a solution, FLUENT will print the current time at the end of each time step.

- If you have chosen the Explicit unsteady formulation, you will follow a different procedure:

  (a) Use the default settings for the Solver Parameters in the Solution Controls panel.

  $\boxed{\text{Solve}} \longrightarrow \boxed{\text{Controls}} \longrightarrow$ Solution...

  If you have modified the Solver Parameters, you can click on the Default button to retrieve the default settings.

  (b) Specify the desired Number of Iterations and click Iterate.

  $\boxed{\text{Solve}} \longrightarrow$ Iterate...

  Remember that when the explicit unsteady formulation is used, each iteration is a time step. When FLUENT prints the residuals to the console window, it will include a column with the current time (if you requested this in step 4, above).

11. Save the final data file (and case file, if you have modified it) so that you can continue the unsteady calculation later, if desired.

    $\boxed{\text{File}} \longrightarrow \boxed{\text{Write}} \longrightarrow$ Data...

Additional Inputs

The procedures for setting the reporting interval, updating UDF profiles, interrupting iterations, and resetting data are the same as those for steady-state calculations. See Section 22.14 for details.

! If you are using a user-defined function in your time-dependent calculation, note that, in addition to being updated after every n iterations (where n is the value of the UDF Profile Update Interval), the function will also be updated at the first iteration of each time step.

22.15.2 Adaptive Time Stepping

As mentioned in Section 22.15.1, it is possible to have the size of the time step change as the calculation proceeds, rather than specifying a fixed size for the entire calculation. This section provides a brief description of the algorithm that FLUENT uses to compute the time step size, as well as an explanation of each of the parameters that you can set to control the adaptive time stepping.

! Adaptive time stepping is available only with the segregated and coupled implicit solvers; it cannot be used with the coupled explicit solver. In addition, it cannot be used with the VOF or discrete phase model.

The Adaptive Time Stepping Algorithm

The automatic determination of the time step size is based on the estimation of the truncation error associated with the time integration scheme (i.e., first-order implicit or second-order implicit). If the truncation error is smaller than a specified tolerance, the size of the time step is increased; if the truncation error is greater, the time step size is decreased.

An estimation of the truncation error can be obtained by using a predictor-corrector type of algorithm [82] in association with the time integration scheme. At each time step, a predicted solution can be obtained using a computationally inexpensive explicit method (forward Euler for the first-order unsteady formulation, Adams-Bashford for the second-order unsteady formulation). This predicted solution is used as an initial condition for the time step, and the correction is computed using the non-linear iterations associated with the implicit (segregated or coupled) algorithm. The norm of the difference between the predicted and corrected solutions is used as a measure of the truncation error. By comparing the truncation error with the desired level of accuracy (i.e., the truncation error tolerance), FLUENT is able to adjust the time step size by increasing it or decreasing it.

Specifying Parameters for Adaptive Time Stepping

The parameters that control the adaptive time stepping appear in the Iterate panel, as described in Section 22.15.1. These parameters are as follows:

Truncation Error Tolerance specifies the threshold value to which the computed truncation error is compared. Increasing this value will lead to an increase in the size of the time step and a reduction in the accuracy of the solution. Decreasing it will lead to a reduction in the size of the time step and an increase in the solution accuracy, although the calculation will require more computational time. For most cases, the default value of 0.01 is acceptable.

Ending Time specifies an ending time for the calculation. Since the ending time cannot be determined by multiplying the number of time steps by a fixed time step size, you need to specify it explicitly.

Minimum/Maximum Time Step Size specify the upper and lower limits for the size of the time step. If the time step becomes very small, the computational expense may be too high; if the time step becomes very large, the solution accuracy may not be acceptable to you. You can set the limits that are appropriate for your simulation.

Minimum/Maximum Step Change Factor limit the degree to which the time step size can change at each time step. Limiting the change results in a smoother calculation of the time step size, especially when high-frequency noise is present in the solution. If the time step change factor, f, is computed as the ratio between the specified truncation error tolerance and the computed truncation error, the size of time step Δt_n is computed as follows:

- If $1 < f < f_{\rm max}$, Δt_n is increased to meet the desired tolerance.
- If $1 < f_{\rm max} < f$, Δt_n is increased, but its maximum possible value is $f_{\rm max} \Delta t_{n-1}$.
- If $f_{\rm min} < f < 1$, Δt_n is unchanged.
- If $f < f_{\rm min} < 1$, Δt_n is decreased.

Number of Fixed Time Steps specifies the number of fixed-size time steps that should be performed before the size of the time step starts to change. The size of the fixed time step is the value specified for Time Step Size in the Iterate panel.

It is a good idea to perform a few fixed-size time steps before switching to the adaptive time stepping. Sometimes spurious discretization errors can be associated with an impulsive start in time. These errors are dissipated during the first few time steps, but they can adversely affect the adaptive time stepping and result in extremely small time steps at the beginning of the calculation.

Specifying a User-Defined Time Stepping Method

If you want to use your own adaptive time stepping method, instead of the method described above, you can create a user-defined function for your method and select it in the User-Defined Time Step drop-down list. The other inputs under Adaptive Time Stepping will not be used when you select a user-defined function.

See the separate UDF Manual for details about creating and using user-defined functions.

22.15.3 Postprocessing for Time-Dependent Problems

The postprocessing of time-dependent data is similar to that for steady-state data, with all graphical and alphanumeric commands available. You can read a data file that was saved at any point in the calculation (by you or with the autosave option) to restore the data at any of the time levels that were saved.

$\boxed{\text{File}} \longrightarrow \boxed{\text{Read}} \longrightarrow$ Data...

FLUENT will label any subsequent graphical or alphanumeric output with the time value of the current data set.

If you save data from the force or surface monitors to files (see step 5 in Section 22.15.1), you can read these files back in and plot them to see a time history of the monitored quantity. Figure 22.15.5 shows a sample plot generated in this way.

If you enabled the Data Sampling for Time Statistics option in the Iterate panel, FLUENT will compute the time average (mean) and root-mean-

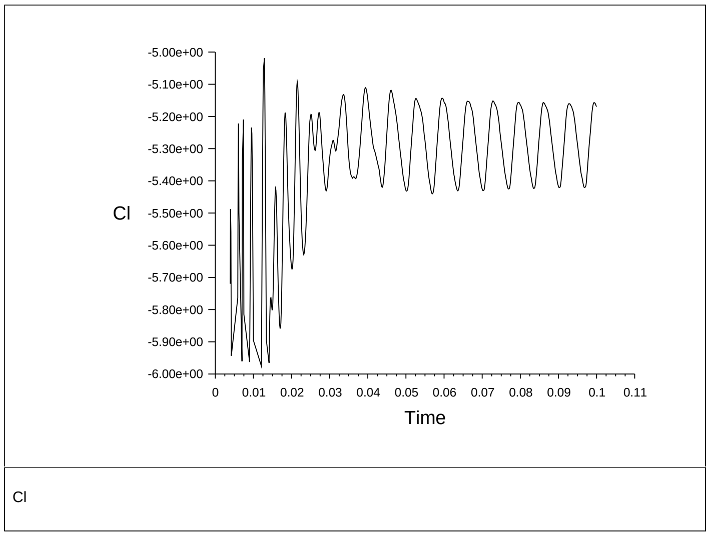


Figure 22.15.5: Lift Coefficient Plot for a Time-Periodic Solution

squares of the instantaneous values sampled during the calculation. The mean and root-mean-square (RMS) values for all solution variables will be available in the Unsteady Statistics... category of the variable selection drop-down list that appears in postprocessing panels.

22.16 Monitoring Solution Convergence

During the solution process you can monitor the convergence dynamically by checking residuals, statistics, force values, surface integrals, and volume integrals. You can print reports of or display plots of lift, drag, and moment coefficients, surface integrations, and residuals for the solution variables. For unsteady flows, you can also monitor elapsed time. Each of these monitoring features is described below.

22.16.1 Monitoring Residuals

At the end of each solver iteration, the residual sum for each of the conserved variables is computed and stored, thus recording the convergence history. This history is also saved in the data file. The residual sum is defined below.

On a computer with infinite precision, these residuals will go to zero as the solution converges. On an actual computer, the residuals decay to some small value ("round-off") and then stop changing ("level out"). For "single precision" computations (the default for workstations and most computers), residuals can drop as many as six orders of magnitude before hitting round-off. Double precision residuals can drop up to twelve orders of magnitude. Guidelines for judging convergence can be found in Section 22.19.1.

Definition of Residuals for the Segregated Solver

After discretization, the conservation equation for a general variable ϕ at a cell P can be written as

$$a_P \phi_P = \sum_{\rm nb} a_{\rm nb} \phi_{\rm nb} + b \tag{22.16-1}$$

Here a_P is the center coefficient, $a_{\rm nb}$ are the influence coefficients for the neighboring cells, and b is the contribution of the constant part of the source term S_c in $S = S_c + S_P \phi$ and of the boundary conditions. In Equation 22.16-1,

$$a_P = \sum_{\text{nb}} a_{\text{nb}} - S_P \qquad (22.16\text{-}2)$$

The residual R^ϕ computed by FLUENT's segregated solver is the imbalance in Equation 22.16-1 summed over all the computational cells P. This is referred to as the "unscaled" residual. It may be written as

$$R^\phi = \sum_{\text{cells } P} \left| \sum_{\text{nb}} a_{\text{nb}}\phi_{\text{nb}} + b - a_P\phi_P \right| \qquad (22.16\text{-}3)$$

In general, it is difficult to judge convergence by examining the residuals defined by Equation 22.16-3 since no scaling is employed. This is especially true in enclosed flows such as natural convection in a room where there is no inlet flow rate of ϕ with which to compare the residual. FLUENT scales the residual using a scaling factor representative of the flow rate of ϕ through the domain. This "scaled" residual is defined as

$$R^\phi = \frac{\sum_{\text{cells } P} \left| \sum_{\text{nb}} a_{\text{nb}}\phi_{\text{nb}} + b - a_P\phi_P \right|}{\sum_{\text{cells } P} |a_P\phi_P|} \qquad (22.16\text{-}4)$$

For the momentum equations the denominator term $a_P\phi_P$ is replaced by a_Pv_P, where v_P is the magnitude of the velocity at cell P.

The scaled residual is a more appropriate indicator of convergence for most problems, as discussed in Section 22.19.1. This residual is the default displayed by FLUENT. Note that this definition of residual is also used by Fluent Inc.'s structured grid solver FLUENT 4.

For the continuity equation, the unscaled residual for the segregated solver is defined as

$$R^c = \sum_{\text{cells } P} |\text{rate of mass creation in cell P}| \qquad (22.16\text{-}5)$$

The segregated solver's scaled residual for the continuity equation is defined as

$$\frac{R^c_{\text{iteration } N}}{R^c_{\text{iteration } 5}} \tag{22.16-6}$$

The denominator is the largest absolute value of the continuity residual in the first five iterations.

The scaled residuals described above are useful indicators of solution convergence. Guidelines for their use are given in Section 22.19.1. It is sometimes useful to determine how much a residual has decreased during calculations as an additional measure of convergence. For this purpose, FLUENT allows you to normalize the residual (either scaled or unscaled) by dividing by the maximum residual value after M iterations, where M is set by you in the Residual Monitors panel in the Iterations field under Normalization.

$$\bar{R}^\phi = \frac{R^\phi_{\text{iteration } N}}{R^\phi_{\text{iteration } M}} \tag{22.16-7}$$

Normalization in this manner ensures that the initial residuals for all equations are of $O(1)$ and is sometimes useful in judging overall convergence.

By default, $M = 5$. You can also specify the normalization factor (the denominator in Equation 22.16-7) manually in the Residual Monitors panel.

Definition of Residuals for the Coupled Solvers

A residual for the coupled solvers is simply the time rate of change of the conserved variable ($\boldsymbol{W}$). The RMS residual is the square root of the average of the squares of the residuals in each cell of the domain:

$$R(\boldsymbol{W}) = \sqrt{\sum \left(\frac{\partial \boldsymbol{W}}{\partial t}\right)^2} \tag{22.16-8}$$

Equation 22.16-8 is the unscaled residual sum reported for all the coupled equations solved by FLUENT's coupled solvers.

! The residuals for the equations that are solved sequentially by the coupled solvers (turbulence and other scalars, as discussed in Section 22.1.2) are the same as those described above for the segregated solver.

In general, it is difficult to judge convergence by examining the residuals defined by Equation 22.16-8 since no scaling is employed. This is especially true in enclosed flows such as natural convection in a room where there is no inlet flow rate of ϕ with which to compare the residual. FLUENT scales the residual using a scaling factor representative of the flow rate of ϕ through the domain. This "scaled" residual is defined as

$$\frac{R(\boldsymbol{W})_{\text{iteration } N}}{R(\boldsymbol{W})_{\text{iteration } 5}} \tag{22.16-9}$$

The denominator is the largest absolute value of the residual in the first five iterations.

The scaled residuals described above are useful indicators of solution convergence. Guidelines for their use are given in Section 22.19.1. It is sometimes useful to determine how much a residual has decreased during calculations as an additional measure of convergence. For this purpose, FLUENT allows you to normalize the residual (either scaled or unscaled) by dividing by the maximum residual value after M iterations, where M is set by you in the Residual Monitors panel in the Iterations field under Normalization.

Normalization of the residual sum is accomplished by dividing by the maximum residual value after M iterations, where M is set by you in the Residual Monitors panel in the Iterations field under Normalization:

$$\bar{R}(\boldsymbol{W}) = \frac{R(\boldsymbol{W})_{\text{iteration } N}}{R(\boldsymbol{W})_{\text{iteration } M}} \tag{22.16-10}$$

Normalization in this manner ensures that the initial residuals for all equations are of $O(1)$ and is sometimes useful in judging overall convergence.

By default, $M = 5$, making the normalized residual equivalent to the scaled residual. You can also specify the normalization factor (the denominator in Equation 22.16-10) manually in the Residual Monitors panel.

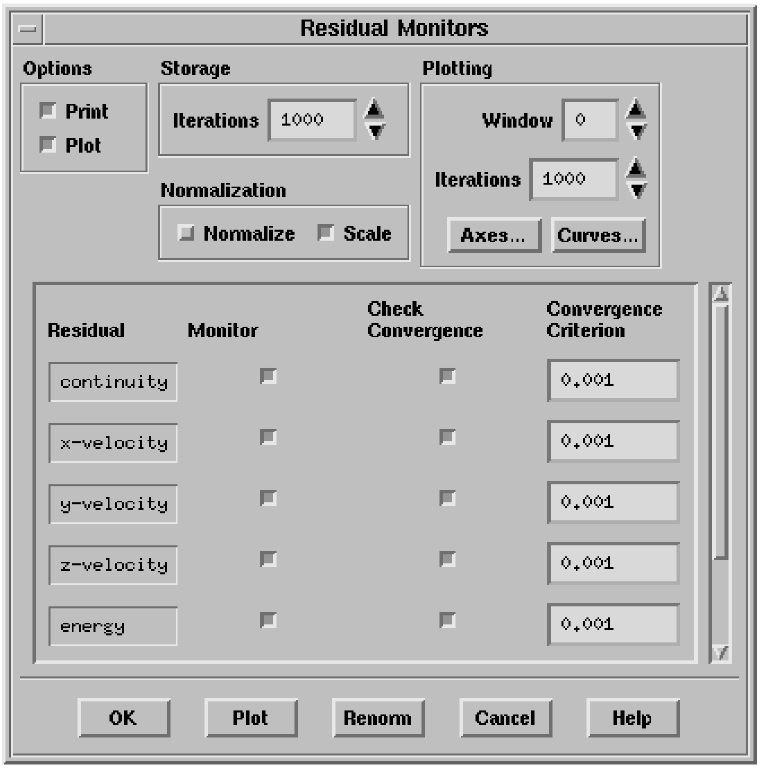
Overview of Using the Residual Monitors *Panel*

All inputs controlling the monitoring of residuals are entered using the Residual Monitors panel (Figure 22.16.1).

$\boxed{\text{Solve}}\longrightarrow\boxed{\text{Monitors}}\longrightarrow$Residual...

or

$\boxed{\text{Plot}}\longrightarrow$Residuals...



Options: ✓ Print, ✓ Plot

Storage: Iterations 1000

Plotting: Window 0, Iterations 1000

Normalization: Normalize, ✓ Scale

Residual	Monitor	Check Convergence	Convergence Criterion
continuity	✓	✓	0.001
x-velocity	✓	✓	0.001
y-velocity	✓	✓	0.001
z-velocity	✓	✓	0.001
energy	✓	✓	0.001

Figure 22.16.1: The Residual Monitors Panel

In general, you will only need to enable residual plotting and modify the convergence criteria using this panel. Additional controls are available for disabling monitoring of particular residuals, and modifying normalization and plot parameters.

Printing and Plotting Residuals

By default, residual values for all relevant variables are printed in the text (console) window after each iteration. If you wish to disable this printout, turn off Print under Options. To enable the plotting of residuals after each iteration, turn on Plot under Options. Residuals will be plotted in the graphics window (with the window ID set in the Window field) during the calculation.

If you wish to display a plot of the current residual history, simply click on the Plot push button.

Modifying Convergence Criteria

In addition to plotting and printing residual values during the calculation, FLUENT will also check for convergence. If convergence is being monitored, the solution will stop automatically when each variable meets its specified convergence criterion. Convergence checks can be performed only for variables for which you are monitoring residuals (i.e., variables for which the Monitor option is enabled).

You can choose whether or not you want to check the convergence for each variable by turning on or off the Check Convergence option for it in the Residual Monitors panel. To modify the convergence criterion for a particular variable, enter a new value in the corresponding Convergence Criterion field.

Plot Parameters

If you choose to plot the residual values (either interactively during the solution or using the Plot button after calculations are complete), there are several display parameters you can modify.

In the Window field under Plotting, you can specify the ID of the graphics window in which the plot will be drawn. When FLUENT is iterating, the active graphics window is temporarily set to this window to update the residual plot, and then returned to its previous value. Thus, the residual plot can be maintained in a separate window that does not interfere with other graphical postprocessing.

By changing the Iterations entry under Plotting, you can modify the number of residual history points to be displayed in the plot. If you specify n points, FLUENT will display the last n history points. Since the y axis is scaled by the minimum and maximum values of all points in the plot, you can "zoom in" on the end of the residual history by setting Iterations to a value smaller than the number of iterations performed. If, for example, the residuals jumped early in the calculation when you turned on turbulence, that peak broadens the overall range in residual values, making the smaller fluctuations later on almost indistinguishable. By setting the value of Iterations so that the plot does not include that early peak, your y-axis range is better suited to the values that you are interested in seeing.

You can also modify the attributes of the plot axes and the residual curves. Click on the Axes... or Curves... button to open the Axes panel or Curves panel. See Sections 25.8.8 and 25.8.9 for details.

Disabling Monitoring

If your problem requires the solution of many equations (e.g., turbulence quantities and multiple species), a plot that includes all residuals may be difficult to read. In such cases, you may choose to monitor only a subset of the residuals, perhaps those that affect convergence the most. You can indicate whether or not you want to monitor residuals for each variable by enabling or disabling the relevant check box in the Monitor list of the Residual Monitors panel.

Controlling Normalization

By default, scaling of residuals (see Equations 22.16-4 and 22.16-9) is enabled and the default convergence criterion is 10^{-6} for energy and P-1 equations and 10^{-3} for all other equations. Residual normalization (i.e., dividing the residuals by the largest value during the first few iterations) is also available but disabled by default.

Normalization can be used with both scaled and unscaled residuals. Note that if normalization is enabled, the convergence criterion may need to be adjusted appropriately. See Section 22.19.1 for information about judging convergence based on the different types of residual reports. (Both the raw residuals and scaling factors are stored in the data file, so you can switch between scaled and unscaled residuals.) To report unscaled residuals, simply turn off the Scale option under Normalization.

! If you switch from scaled to unscaled residuals (or vice versa) and you are normalizing the residuals (as described below), you must click on the Renorm button to recompute the normalization factors.

If you wish to normalize the residuals (see Equation 22.16-7 or 22.16-10), turn on the Normalize option under Normalize. The Normalization Factor column will be added to the panel at this time. FLUENT will normalize the printed or plotted residual for each variable by the value indicated as the Normalization Factor for that variable. The default Normalization Factor is the maximum residual value after the first 5 iterations. To use the maximum residual value after a different number of iterations (i.e., specify a different value for M in Equation 22.16-7 or 22.16-10), you can modify the Iterations entry under Normalization.

In some cases, the maximum residual may occur sometime after the iteration specified in the Iterations field. If this should occur, you can click on the Renorm button to set the normalization factors for all variables to the maximum values in the residual histories. Subsequent plots and printed reports will use the new normalization factor.

You can also specify the normalization factor (the denominator in Equation 22.16-7 or 22.16-10) explicitly. To modify the normalization factor for a particular variable, enter a new value in the corresponding Normalization Factor field in the Residual Monitors panel.

If you wish to report unnormalized, unscaled residuals (Equation 22.16-3 or 22.16-8), turn off the Normalize and Scale options under Normalization in the Residual Monitors panel. Note that unnormalized, unscaled residuals are stored in the data file regardless of whether the reported residuals are normalized or scaled.

Storing Residual History Points

Residual histories for each variable are automatically saved in the data file, regardless of whether they are being monitored. You can control the number of history points to be stored by changing the Iterations entry under Storage. By default, up to 1000 points will be stored. If more than 1000 iterations are performed (i.e., the limit is reached), every other point will be discarded—leaving 500 history points—and the next 500 points will be stored. When the total hits 1000 again, every other point will again be discarded, etc. If you are performing a large number of iterations, you will lose a great deal of residual history information about the beginning of the calculation. In such cases, you should increase the Iterations value to a more appropriate value. Of course, the larger this number is, the more memory you will need, the longer the plotting will take, and the more disk space you will need to store the data file.

Postprocessing Residual Values

If you are having solution convergence difficulties, it is often useful to plot the residual value fields (e.g., using contour plots) to determine where the high residual values are located. When you use one of the coupled solvers, the residual values for all solution variables are available in the Residuals... category in the postprocessing panels. (If you read case and data files into FLUENT, you will need to perform at least one iteration before the residual values are available for postprocessing.) For the segregated solver, however, only the mass imbalance in each cell is available by default.

If you want to plot residual value fields for a segregated solver calculation, you will need to do the following:

1. Read in the case and data files of interest (if they are not already in the current session).

2. Use the `expert` command in the `solve/set/` text menu to enable the saving of residual values.

   $\boxed{\texttt{solve}} \longrightarrow \boxed{\texttt{set}} \longrightarrow$`expert`

Among other questions, FLUENT will ask if you want to save cell residuals for postprocessing. Enter `yes` or `y`, and keep the default settings for all of the other questions (by pressing the `<RETURN>` key).

3. Perform at least one iteration.

The solution variables for which residual values are available will appear in the Residuals... category in the postprocessing panels. Note that residual values are *not* available for the radiative transport equations solved by the discrete ordinates radiation model.

22.16.2 Monitoring Statistics

If you are solving a fully-developed periodic flow, you may want to monitor the pressure gradient or the bulk temperature ratio, as discussed in Section 8.3.

If you are solving an unsteady flow (especially if you are using the explicit time stepping option), you may want to monitor the "time" that has elapsed during the calculation. The physical time of the flow field starts at zero when you initialize the flow. (See Section 22.15 for details about modeling unsteady flows.)

If you are using the adaptive time stepping method described in Section 22.15.2, you may want to monitor the size of the time step, Δt.

You can use the Statistic Monitors panel (Figure 22.16.2) to print or plot these quantities during the calculation.

$\boxed{\text{Solve}} \longrightarrow \boxed{\text{Monitors}} \longrightarrow$ Statistic...

The procedure for setting up this monitor is listed below:

1. Indicate the type of report you want by turning on the Print option for a printout or the Plot option for a plot. You can enable these options simultaneously.

2. Select the appropriate quantity in the Statistics list.

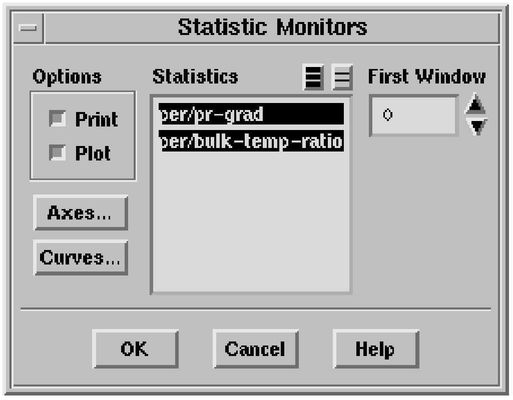


Figure 22.16.2: The Statistic Monitors Panel

3. If you are plotting the quantities, you can set any of the plotting options discussed below.

Plot Parameters

If you choose to plot the statistics, there are several display parameters you can modify.

In the First Window field, you can specify the ID of the graphics window in which the plot will be drawn (or in which the first plot will be drawn, if you are plotting more than one quantity.) When FLUENT is iterating, the active graphics window is temporarily set to this window to update the plot, and then returned to its previous value. Thus, the statistics plot can be maintained in a separate window that does not interfere with other graphical postprocessing. Note that additional quantities that you have selected in the Statistics list will be plotted in windows with incrementally higher IDs.

You can also modify the attributes of the plot axes and curves. Click on the Axes... or Curves... button to open the Axes panel or Curves panel. See Sections 25.8.8 and 25.8.9 for details.

22.16.3 Monitoring Forces and Moments

At the end of each solver iteration, the lift, drag, and/or moment coefficient can be computed and stored to create a convergence history. You can print and plot this convergence data, and also save it to an external file. The external file is written in the FLUENT XY plot file format described in Section 25.8.5. Monitoring forces can be useful when you are calculating external aerodynamics, for example, and you are especially interested in the forces. Sometimes the forces converge before the residuals have dropped three orders of magnitude, so you can save time by stopping the calculation earlier than you would if you were monitoring only residuals. (You should be sure to check the mass flow rate and heat transfer rate as well, using the Flux Reports panel described in Section 26.2, to ensure that the mass and energy are being suitably conserved.)

! The force and moment coefficients use the reference values described in Section 26.8. Specifically, the force coefficients use the reference area, density, and velocity, and the moment coefficients use the reference area, density, velocity and length.

! Only the processed force coefficient data is saved. If you decide to change any of the parameters controlling the force monitoring, such as the reference values, force vector, moment center, moment axis, or wall zones, you may see a discontinuity in the data: the previous data is not updated. Usually, if you have made changes you will want to delete the previous force coefficient data before continuing to iterate.

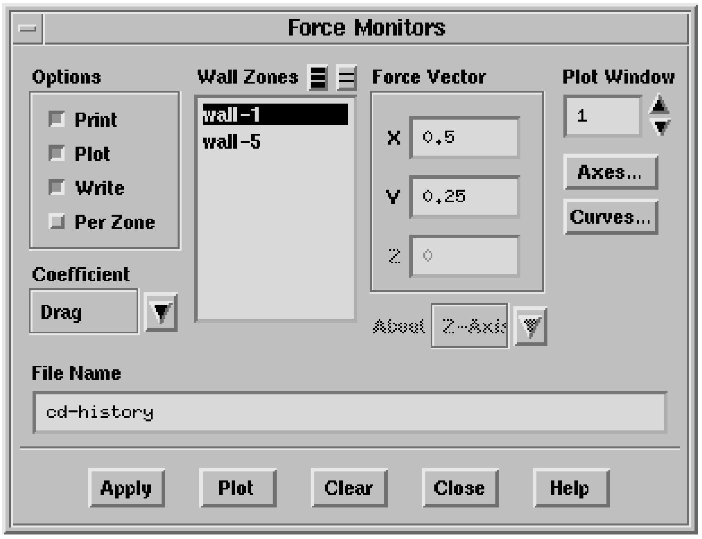
Monitoring Forces in Unsteady Flow Calculations

If you are calculating unsteady flow, the specified force reports will be updated after each time step, rather than after each iteration. All other features of the force monitor and the related setup procedures are unchanged.

Overview of Using the Force Monitors *Panel*

You can use the Force Monitors panel (Figure 22.16.3) to print, plot, and save the convergence history of the drag, lift, and moment coefficients on specified wall zones.

$\boxed{\text{Solve}}\longrightarrow\boxed{\text{Monitors}}\longrightarrow$Force...



Options: [✓ Print] [✓ Plot] [✓ Write] Per Zone
Wall Zones: [✓ wall-1] wall-5
Force Vector: X 0.5, Y 0.25, Z 0
Plot Window: 1
Coefficient: Drag
About: Z-Axis
File Name: cd-history

Figure 22.16.3: The Force Monitors Panel

In this panel you will indicate the types of reports that you want (printouts, plots, or files), and specify which coefficients and wall zones are of interest. Additional information will be entered for each coefficient that is monitored. You can also modify the plot parameters.

! Remember to click Apply after making the desired modifications to the setup for each coefficient report.

Specifying the Force Coefficient Report

For each coefficient that you choose to monitor, you will set all appropriate parameters in the Force Monitors panel and click Apply. You can monitor one, two, or all three of the coefficients (drag, lift, and moment vector component) while iterating. When you select the desired coefficient, the current (or default) panel settings are shown for that coefficient. Clicking the Apply button will save the current panel settings for the selected coefficient.

The procedure for specifying a force coefficient report is listed below:

1. Indicate the type of report you want (printout, plot, or file), as described below.
2. If you want to monitor the force or moment on individual walls in a single printout, plot, or file, turn on the Per Zone option. See below for details.
3. Choose the coefficient of interest by selecting Drag, Lift, or Moment in the Coefficient drop-down list.
4. In the Wall Zones list, select the wall zone(s) on which the selected coefficient is to be computed. If you are monitoring more than one coefficient, the selected wall zones are often the same for each coefficient. If you want, however, you can have each coefficient computed on a different set of zones.
5. Depending on the coefficient selected, do one of the following:
   - If you are monitoring the drag or lift coefficient, enter the X, Y, and Z components of the Force Vector along which the forces will be computed. The Force Vector heading will appear only if you have selected Drag or Lift in the Coefficient drop-down list. By default, drag is computed in the x direction and lift in the y direction.
   - If you are monitoring the moment coefficient, enter the Cartesian coordinates (X, Y, and Z) of the Moment Center about which moments will be computed. The Moment Center heading will appear only if you have selected Moment in the Coefficient drop-down list. The default moment center is (0,0,0). You will also need to specify which component of the moment vector you wish to monitor. Presently, you can monitor only one component of the moment vector at a time. Select X-Axis, Y-Axis, or Z-Axis in the About drop-down list. (This list is active only if you have selected Moment in the Coefficient drop-down list.) For two dimensional flows, only the moment vector about the z-coordinate axis exists.

6. Click Apply and repeat the process for additional coefficients, if desired.

Printing, Plotting, and Saving Force Coefficient Histories

There are three methods available for reporting the selected force coefficients. To print the coefficient value(s) in the text (console) window after each iteration, turn on the Print option under Options in the Force Monitors panel. To plot the coefficient in the graphics window indicated in Plot Window, turn on the Plot option. If you want to save the values to a file, turn on the Write option and specify the File Name. You can enable any combination of these options simultaneously.

! If you choose *not* to save the force coefficient data in a file, this information will be lost when you exit the current FLUENT session.

If you wish to display a plot of the current force-coefficient history, simply click on the Plot button.

Plot Parameters

If you choose to plot the force coefficients (either interactively during the solution or using the Plot button after calculations are complete), there are several display parameters you can modify.

In the Plot Window field, you can specify the ID of the graphics window in which the plot for each force coefficient will be drawn. When FLUENT is iterating, the active graphics window is temporarily set to this window to update the plot, and then returned to its previous value. Thus, the force-coefficient plots can be maintained in separate windows that do not interfere with other graphical postprocessing.

You can also modify the attributes of the plot axes and the coefficient curves. The same attributes apply to all force-monitor plots. Click on the Axes... or Curves... button to open the Axes panel or Curves panel. See Sections 25.8.8 and 25.8.9 for details.

Monitoring Forces and Moments on Individual Walls

By default, FLUENT will compute and monitor the total force or moment for all of the selected walls combined together. If you have selected multiple walls and you want to monitor the force or moment on each wall separately, you can turn on the Per Zone option in the Force Monitors panel. The specified force vector or moment axis will apply to all selected walls.

If the monitor results are printed to the console (using the Print option), the force or moment on each wall will be printed in a separate column. If the results are plotted (using the Plot option), a separate curve for each wall will be drawn in the specified plot window. If the results are written to a file (using the Write option), the file will be in a tab-separated column format based on the XY plot file format described in Section 25.8.5.

Discarding the Force-Monitoring Data

Should you decide that the data gathered by the force monitor are not useful (e.g., if you are restarting the calculation or you changed one of the reference values), you can discard the accumulated data by clicking on the Clear button. The Clear button will delete all monitoring data for the coefficient selected in the Coefficient drop-down list, including the associated history file (with the name in the File Name field). When you use the Clear button, you will need to confirm the data discard in a Question dialog box. Only the force-monitoring data is removed using this operation; the solution data is not affected.

22.16.4 Monitoring Surface Integrals

At the end of each solver iteration or time step, the average, mass average, integral, flow rate, or other integral report of a field variable or function can be monitored on a surface. You can print and plot these convergence data, and also save them in an external file. The external file is written in the FLUENT XY plot file format described in Section 25.8.5. The report types available are the same as those in the Surface Integrals panel, as described in Section 26.5.

Monitoring surface integrals can be used to check for both iteration convergence and grid independence. For example, you can monitor the average value of a certain variable on a surface. When this value stops changing, you can stop iterating. You can then adapt the grid and reconverge the solution. The solution can be considered grid-independent when the average value on the surface stops changing between adaptions.

Overview of Defining Surface Monitors

You can use the Surface Monitors panel (Figure 22.16.4) to create surface monitors and indicate whether and when each one's history is to be printed, plotted, or saved. The Define Surface Monitor panel (Figure 22.16.5), opened from the Surface Monitors panel, allows you to define what each monitor tracks (i.e., the average, integral, flow rate, mass average, or other integral report of a field variable or function on one or more surfaces).

Defining Surface Monitors

You will begin the surface monitor definition procedure in the Surface Monitors panel (Figure 22.16.4).

$\boxed{\text{Solve}} \longrightarrow \boxed{\text{Monitors}} \longrightarrow$ Surface...

The procedure is as follows:

1. Increase the Surface Monitors value to the number of surface monitors you wish to specify. As this value is increased, additional

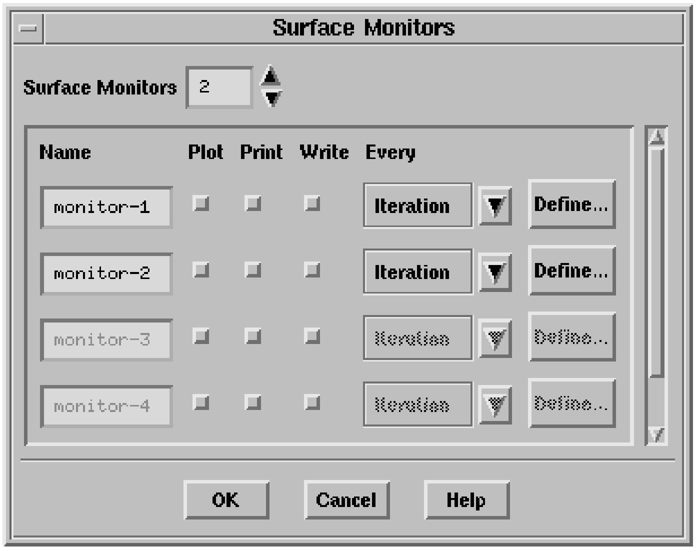


Figure 22.16.4: The Surface Monitors Panel

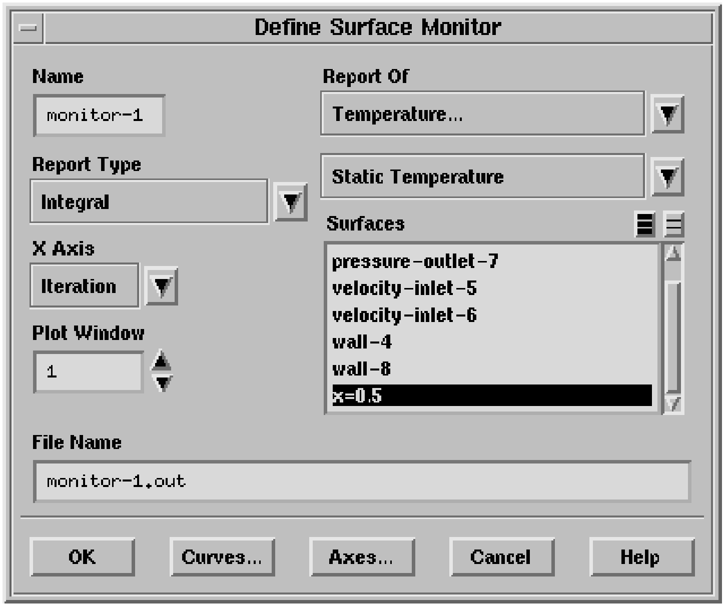
monitor entries in the panel will become editable. For each monitor, you will perform the following steps.

2. Enter a name for the monitor under the Name heading, and use the Plot, Print, and Write check buttons to indicate the report(s) you want (plot, printout, or file), as described below.

3. Indicate whether you want to update the monitor every Iteration or every Time Step by selecting the appropriate item in the drop-down list below Every. Time Step is a valid choice only if you are calculating unsteady flow. If you specify every Iteration, and the Reporting Interval in the Iterate panel is greater than 1, the monitor will be updated at every reporting interval instead of at each iteration (e.g., for a reporting interval of 2, the monitor will be updated after every other iteration). If you specify every Time Step, the reporting interval will have no effect; the monitor will always be updated after each time step.

4. Click on the Define... button to open the Define Surface Monitor

panel (Figure 22.16.5). Since this is a modal panel, the solver will not allow you to do anything else until you perform steps 5–10, below.



Name: monitor-1
Report Of: Temperature...
Static Temperature
Report Type: Integral
X Axis: Iteration
Plot Window: 1
Surfaces: pressure-outlet-7, velocity-inlet-5, velocity-inlet-6, wall-4, wall-8, x=0.5
File Name: monitor-1.out

Figure 22.16.5: The Define Surface Monitor Panel

5. In the Define Surface Monitor panel, choose the integration method for the surface monitor by selecting Integral, Area-Weighted Average, Flow Rate, Mass Flow Rate, Mass-Weighted Average, Sum, Facet Average, Facet Minimum, Facet Maximum, Vertex Average, Vertex Minimum, or Vertex Maximum in the Report Type drop-down list. These methods are described in Section 26.5.

6. In the Surfaces list, choose the surface or surfaces on which you wish to integrate.

7. Specify the variable or function to be integrated in the Report Of drop-down list. First select the desired category in the upper drop-down list. You can then select one of the related quantities in the

lower list. (See Chapter 27 for an explanation of the variables in the list.)

8. If you are plotting the data or writing them to a file, specify the parameter to be used as the x-axis value (the y-axis value corresponds to the monitored data). In the X Axis drop-down list, select Iteration, Time Step, or Flow Time as the x-axis function against which monitored data will be plotted or written. Time Step and Flow Time are valid choices only if you are calculating unsteady flow. If you choose Time Step, the x axis of the plot will indicate the time step, and if you choose Flow Time, it will indicate the elapsed time.

9. If you are plotting the monitored data, specify the ID of the graphics window in which the plot will be drawn in the Plot Window field. When FLUENT is iterating, the active graphics window is temporarily set to this window to update the plot, and then returned to its previous value. Thus, each surface-monitor plot can be maintained in a separate window that does not interfere with other graphical postprocessing.

10. If you are writing the monitored data to a file, specify the File Name.

11. Remember to click OK in the Surface Monitors panel after you finish defining all surface monitors.

Printing, Plotting, and Saving Surface Integration Histories

There are three methods available for reporting the selected surface integration. To print the surface integration in the text (console) window after each iteration, turn on the Print option in the Surface Monitors panel. To plot the integrated values in the graphics window indicated in Plot Window (in the Define Surface Monitor panel), turn on the Plot option in the Surface Monitors panel. If you want to save the values to a file, turn on the Write option in the Surface Monitors panel and specify the File Name in the Define Surface Monitor panel. You can enable any combination of these options simultaneously.

! If you choose *not* to save the surface integration data in a file, this information will be lost when you exit the current FLUENT session.

Plot Parameters

You can modify the attributes of the plot axes and curves used for each surface-monitor plot. Click on the Axes... or Curves... button in the Define Surface Monitor panel for the appropriate monitor to open the Axes panel or Curves panel for that surface-monitor plot. See Sections 25.8.8 and 25.8.9 for details.

22.16.5 Monitoring Volume Integrals

At the end of each solver iteration or time step, the volume or the sum, volume integral, volume average, mass integral, or mass average of a field variable or function can be monitored in one or more cell zones. You can print and plot these convergence data, and also save them in an external file. The external file is written in the FLUENT XY plot file format described in Section 25.8.5. The report types available are the same as those in the Volume Integrals panel, as described in Section 26.6.

Monitoring volume integrals can be used to check for both iteration convergence and grid independence. For example, you can monitor the average value of a certain variable in a particular cell zone. When this value stops changing, you can stop iterating. You can then adapt the grid and reconverge the solution. The solution can be considered grid-independent when the average value in the cell zone stops changing between adaptions.

Overview of Defining Volume Monitors

You can use the Volume Monitors panel (Figure 22.16.6) to create volume monitors and indicate whether and when each one's history is to be printed, plotted, or saved. The Define Volume Monitor panel (Figure 22.16.7), opened from the Volume Monitors panel, allows you to define what each monitor tracks (i.e., the volume or the sum, integral, or average of a field variable or function in one or more cell zones).

Defining Volume Monitors

You will begin the volume monitor definition procedure in the Volume Monitors panel (Figure 22.16.6).

$\boxed{\text{Solve}} \longrightarrow \boxed{\text{Monitors}} \longrightarrow$ Volume...

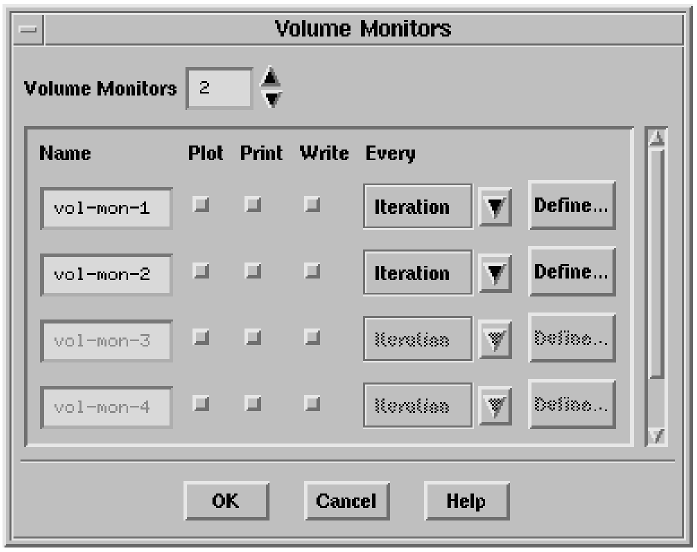


Figure 22.16.6: The Volume Monitors Panel

The procedure is as follows:

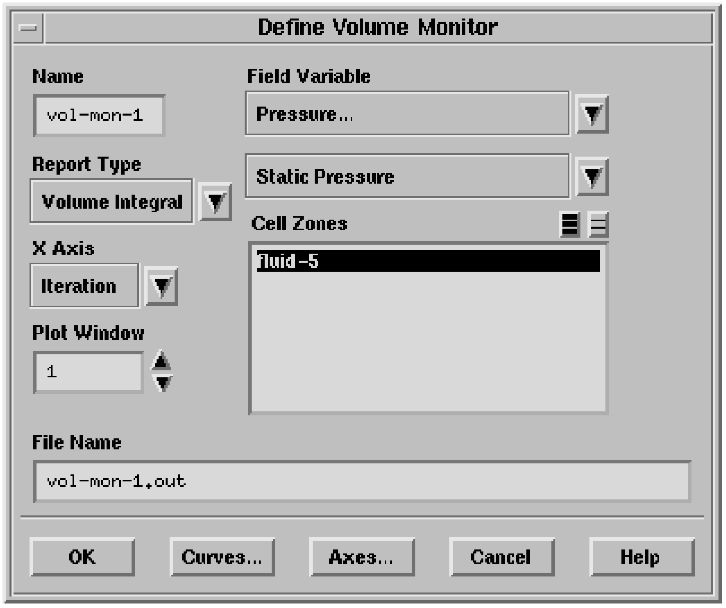
1. Increase the Volume Monitors value to the number of volume monitors you wish to specify. As this value is increased, additional monitor entries in the panel will become editable. For each monitor, you will perform the following steps.

2. Enter a name for the monitor under the Name heading, and use the Plot, Print, and Write check buttons to indicate the report(s) you want (plot, printout, or file), as described below.

3. Indicate whether you want to update the monitor every Iteration or every Time Step by selecting the appropriate item in the drop-

down list below Every. Time Step is a valid choice only if you are calculating unsteady flow. If you specify every Iteration, and the Reporting Interval in the Iterate panel is greater than 1, the monitor will be updated at every reporting interval instead of at each iteration (e.g., for a reporting interval of 2, the monitor will be updated after every other iteration). If you specify every Time Step, the reporting interval will have no effect; the monitor will always be updated after each time step.

4. Click on the Define... button to open the Define Volume Monitor panel (Figure 22.16.7). Since this is a modal panel, the solver will not allow you to do anything else until you perform steps 5–10, below.


Define Volume Monitor

Name: vol-mon-1
Field Variable: Pressure...
Static Pressure
Report Type: Volume Integral
X Axis: Iteration
Plot Window: 1
Cell Zones: fluid-5
File Name: vol-mon-1.out

Figure 22.16.7: The Define Volume Monitor Panel

5. In the Define Volume Monitor panel, choose the integration method for the volume monitor by selecting Volume, Sum, Volume Integral, Volume-Average, Mass Integral, or Mass-Average in the Report Type

drop-down list. These methods are described in Section 26.6.

6. In the Cell Zones list, choose the cell zone(s) on which you wish to integrate.

7. Specify the variable or function to be integrated in the Report Of drop-down list. First select the desired category in the upper drop-down list. You can then select one of the related quantities in the lower list. (See Chapter 27 for an explanation of the variables in the list.)

8. If you are plotting the data or writing them to a file, specify the parameter to be used as the x-axis value (the y-axis value corresponds to the monitored data). In the X Axis drop-down list, select Iteration, Time Step, or Flow Time as the x-axis function against which monitored data will be plotted or written. Time Step and Flow Time are valid choices only if you are calculating unsteady flow. If you choose Time Step, the x axis of the plot will indicate the time step, and if you choose Flow Time, it will indicate the elapsed time.

9. If you are plotting the monitored data, specify the ID of the graphics window in which the plot will be drawn in the Plot Window field. When FLUENT is iterating, the active graphics window is temporarily set to this window to update the plot, and then returned to its previous value. Thus, each volume-monitor plot can be maintained in a separate window that does not interfere with other graphical postprocessing.

10. If you are writing the monitored data to a file, specify the File Name.

11. Remember to click OK in the Volume Monitors panel after you finish defining all volume monitors.

Printing, Plotting, and Saving Volume Integration Histories

There are three methods available for reporting the selected volume integration. To print the volume integration in the text (console) window after each iteration, turn on the Print option in the Volume Monitors panel. To plot the integrated values in the graphics window indicated in Plot Window (in the Define Volume Monitor panel), turn on the Plot option in the Volume Monitors panel. If you want to save the values to a file, turn on the Write option in the Volume Monitors panel and specify the File Name in the Define Volume Monitor panel. You can enable any combination of these options simultaneously.

! If you choose *not* to save the volume integration data in a file, this information will be lost when you exit the current FLUENT session.

Plot Parameters

You can modify the attributes of the plot axes and curves used for each volume-monitor plot. Click on the Axes... or Curves... button in the Define Volume Monitor panel for the appropriate monitor to open the Axes panel or Curves panel for that volume-monitor plot. See Sections 25.8.8 and 25.8.9 for details.

22.17 Animating the Solution

During the calculation, you can have FLUENT create an animation of contours, vectors, XY plots, monitor plots (residual, statistic, force, surface, or volume), or the mesh (useful primarily for moving mesh simulations). Before you begin the calculation, you will specify and display the variables and types of plots you want to animate, and how often you want plots to be saved. At the specified intervals, FLUENT will display the requested plots, and store each one. When the calculation is complete, you can play back the animation sequence, modify the view (for grid, contour, and vector plots), if desired, and save the animation to a series of hardcopy files or an MPEG file

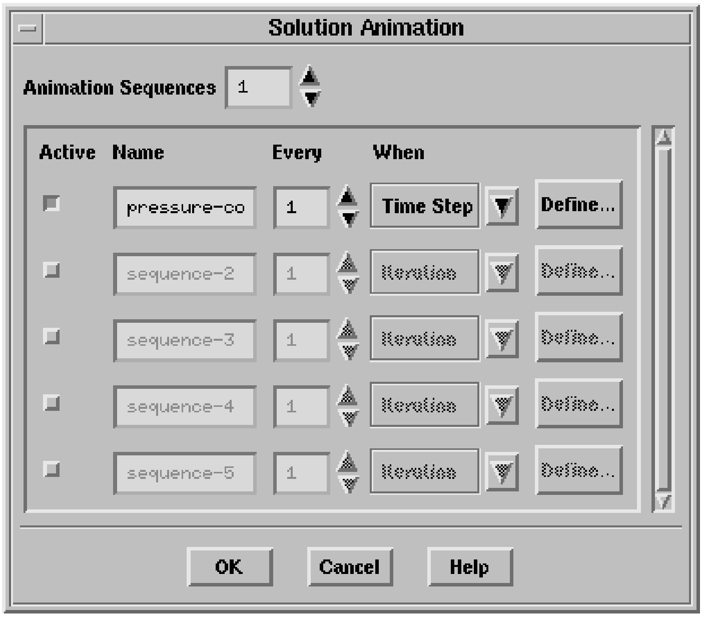
Instructions for defining a solution animation sequence are provided in Section 22.17.1. Sections 22.17.2 and 22.17.3 describe how to play back and save the animation sequences you have created, and Section 22.17.4 describes how to read a previously-saved animation sequence into FLUENT.

22.17.1 Defining an Animation Sequence

You can use the Solution Animation panel (Figure 22.17.1) to create an animation sequence and indicate how often each frame of the sequence should be created. The Animation Sequence panel (Figure 22.17.2), opened from the Solution Animation panel, allows you to define what each sequence displays (e.g., contours or vectors of a particular variable), where it is displayed, and how each frame is stored.

You will begin the animation sequence definition in the Solution Animation panel (Figure 22.17.1).

$\boxed{\text{Solve}} \longrightarrow \boxed{\text{Animate}} \longrightarrow$ Define...



Animation Sequences 1

Active	Name	Every	When	
✓	pressure-co	1	Time Step	Define...
	sequence-2	1	Iteration	Define...
	sequence-3	1	Iteration	Define...
	sequence-4	1	Iteration	Define...
	sequence-5	1	Iteration	Define...

Figure 22.17.1: The Solution Animation Panel

The procedure is as follows:

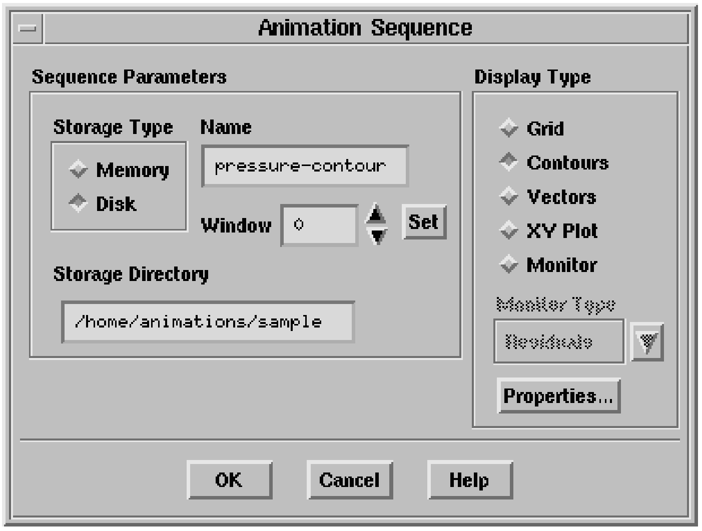
1. Increase the Animation Sequences value to the number of animation sequences you wish to specify. As this value is increased,

additional sequence entries in the panel will become editable. For each sequence, you will perform the following steps.

2. Enter a name for the sequence under the Name heading. This name will be used to identify the sequence in the Playback panel, where you can play back the animation sequences that you have defined or read in. This name will also be used as the prefix for the file names if you save the sequence frames to disk.

3. Indicate how often you want to create a new frame in the sequence by setting the interval under Every and selecting Iteration or Time Step in the drop-down list below When. (Time Step is a valid choice only if you are calculating unsteady flow.) For example, to create a frame every 10 time steps, you would enter `10` under Every and select Time Step under When.

4. Click on the Define... button to open the Animation Sequence panel (Figure 22.17.2).



Storage Type: Memory, [✓ Disk]

Name: pressure-contour

Window: 0

Storage Directory: /home/animations/sample

Display Type: [✓ Contours]

Figure 22.17.2: The Animation Sequence Panel

5. Define the Sequence Parameters in the Animation Sequence panel.

   (a) Specify whether you want FLUENT to save the animation sequence frames in memory or on your computer's hard drive by selecting Memory or Disk under Storage Type.

   ! Note that a FLUENT metafile is created for each frame in the animation sequence. These files contain information about the entire scene, not just the view that is displayed in the plot. As a result, they can be quite large. By default, the files will be stored to disk. If you do not want to use up disk space to store them, you can instead choose to store them in memory. Storing them in memory will, however, reduce the amount of memory available to the solver. Note that the playback of a sequence stored in memory will be faster than one stored to disk.

   (b) If you selected Disk under Storage Type, specify the directory where you want to store the files in the Storage Directory field. (This can be a relative or absolute path.)

   (c) Specify the ID of the graphics window where you want the plot to be displayed in the Window field, and click Set. (The specified window will open, if it is not already open.)

   When FLUENT is iterating, the active graphics window is set to this window to update the plot. If you want to maintain each animation in a separate window, specify a different Window ID for each.

6. Define the display properties for the sequence.

   (a) Under Display Type in the Animation Sequence panel, choose the type of display you want to animate by selecting Grid, Contours, Vectors, XY Plot, or Monitor. If you choose Monitor, you can select any of the available types of monitor plots in the Monitor Type drop-down list: Residuals, Force, Statistics, Surface, or Volume.

   The first time that you select Contours, Vectors, or XY Plot, or one of the monitor types if you select Monitor, FLUENT will open the corresponding panel (e.g., the Contours panel or the

Vectors panel) so you can modify the settings and generate the display. To make subsequent modifications to the display settings for any of the display types, click the Properties... button to open the panel for the selected Display Type.

(b) Define the display in the panel for the selected Display Type (e.g., the Contours or Solution XY Plot panel), and click Display or Plot.

! You must click Display or Plot to initialize the scene to be repeated during the calculation.

See below for guidelines on defining display properties for grid, contour, and vector displays.

7. Remember to click OK in the Solution Animation panel after you finish defining all animation sequences.

Note that, when you click OK in the Animation Sequence panel for a sequence, the Active button for that sequence in the Solution Animation panel will be turned on automatically. You can choose to use a subset of the sequences you have defined by turning off the Active button for those that you currently do not wish to use.

Guidelines for Defining an Animation Sequence

If you are defining an animation sequence containing grid, contour, or vector displays, note the following when you are defining the display:

- If you want to include lighting effects in the animation frames, be sure to define the lights before you begin the calculation. See Section 25.2.6 for information about adding lights to the display.
- If you want to maintain a constant range of colors in a contour or vector display, you can specify a range explicitly by turning off the Auto Range option in the Contours or Vectors panel. See Section 25.1.2 or 25.1.3 for details.
- Scene manipulations that are specified using the Scene Description panel will *not* be included in the animation sequence frames. View

modifications such as mirroring across a symmetry plan *will* be included.

22.17.2 Playing an Animation Sequence

Once you have defined a sequence (as described in Section 22.17.1) and performed a calculation, or read in a previously created animation sequence (as described in Section 22.17.4), you can play back the sequence using the Playback panel (Figure 22.17.3).

Solve ⟶ Animate ⟶Playback...

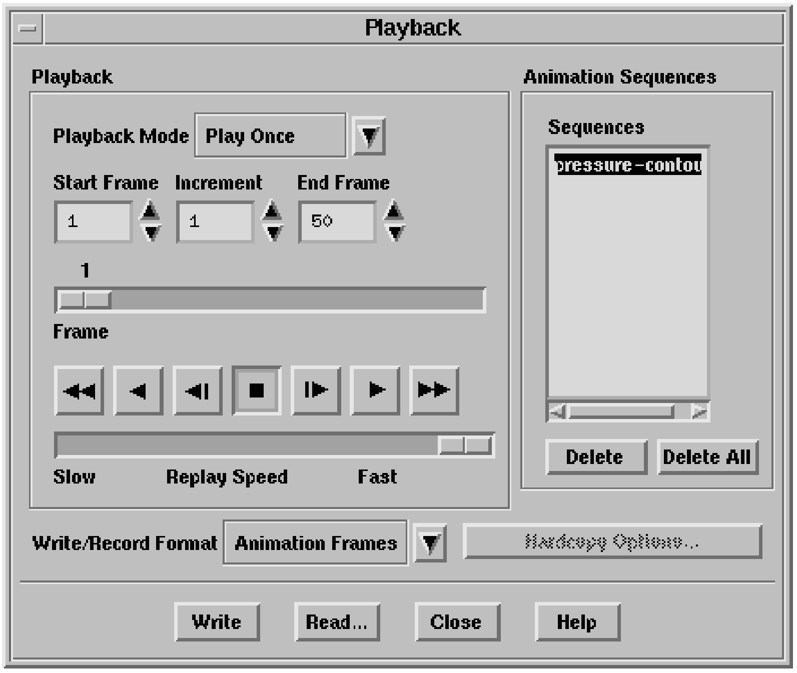


Figure 22.17.3: The Playback Panel

Under Animation Sequences in the Playback panel, select the sequence you want to play in the Sequences list. To play the animation once through from start to finish, click on the "play" button under the Playback heading. (The buttons function in a way similar to those on a standard video cassette player. "Play" is the second button from the right—a single triangle pointing to the right.) To play the animation backwards once, click on the "play reverse" button (the second from the left—a single triangle point to the left). As the animation plays, the Frame scale shows the number of the frame that is currently displayed, as well as its relative position in the entire animation. If, instead of playing the complete animation sequence, you want to jump to a particular frame, move the Frame slider bar to the desired frame number, and the frame corresponding to the new frame number will be displayed in the graphics window.

! For smoother animations, turn on the Double Buffering option in the Display Options panel (see Section 25.2.7). This will reduce screen flicker during graphics updates.

Additional options for playing back animations are described below.

Modifying the View

If you want to replay the animation sequence with a different view of the scene, you can use your mouse to modify (e.g., translate, rotate, zoom) it in the graphics window where the animation is displayed. Note that any changes you make to the view for an animation sequence will be lost when you select a new sequence (or reselect the current sequence) in the Sequences list.

Modifying the Playback Speed

Different computers will play the animation sequence at different speeds, depending on the complexity of the scene and the type of hardware used for graphics. You may want to slow down the playback speed for optimal viewing. Move the Replay Speed slider bar to the left to reduce the playback speed (and to the right to increase it).

Playing Back an Excerpt

You may sometimes want to play only one portion of a long animation sequence. To do this, you can modify the Start Frame and the End Frame under the Playback heading. For example, if your animation contains 50 frames, but you want to play only frames 20 to 35, you can set Start Frame to 20 and End Frame to 35. When you play the animation, it will start at frame 20 and finish at frame 35.

"Fast-Forwarding" the Animation

You can "fast-forward" or "fast-reverse" the animation by skipping some of the frames during playback. To fast-forward the animation, you will need to set the Increment and click on the fast-forward button (the last button on the right—two triangles pointing to the right). If, for example, your Start Frame is 1, your End Frame is 15, and your Increment is 2, when you click on the fast-forward button, the animation will show frames 1, 3, 5, 7, 9, 11, 13 and 15. Clicking on the fast-reverse button (the first button on the left—two triangles pointing to the left) will show frames 15, 13, 11,...1.

Continuous Animation

If you want the playback of the animation to repeat continuously, there are two options available. To continuously play the animation from beginning to end (or from end to beginning, if you use one of the reverse play buttons), select Auto Repeat in the Playback Mode drop-down list. To play the animation back and forth continuously, reversing the playback direction each time, select Auto Reverse in the Playback Mode drop-down list.

To turn off the continuous playback, select Play Once in the Playback Mode list. This is the default setting.

Stopping the Animation

To stop the animation during playback, click on the "stop" button (the square in the middle of the playback control buttons). If your animation contains very complicated scenes, there may be a slight delay before the animation stops.

Advancing the Animation Frame by Frame

To advance the animation manually frame by frame, use the third button from the right (a vertical bar with a triangle pointing to the right). Each time you click on this button, the next frame will be displayed in the graphics window. To reverse the animation frame by frame, use the third button from the left (a left-pointing triangle with a vertical bar). Frame-by-frame playback allows you to freeze the animation at points that are of particular interest.

Deleting an Animation Sequence

If you want to remove one of the sequences that you have created or read in, select it in the Sequences list and click on the Delete button. If you want to delete all sequences, click on the Delete All button.

! Note that if you delete a sequence that has not yet been saved to disk (i.e., if you selected Memory under Storage Type in the Animation Sequence panel), it will be removed from memory permanently. If you want to keep any animation sequences that are stored only in memory, you should be sure to save them (as described in Section 22.17.3) before you delete them from the Sequences list or exit FLUENT.

22.17.3 Saving an Animation Sequence

Once you have created an animation sequence, you can save it in any of the following formats:

- Solution animation file containing the FLUENT metafiles
- Hardcopy files, each containing a frame of the animation sequence
- MPEG file containing each frame of the animation sequence

Note that, if you are saving hardcopy files or an MPEG file, you can modify the view (e.g., translate, rotate, zoom) in the graphics window where the animation is displayed, and save the modified view instead of the original view.

Solution Animation File

If you selected Disk under Storage Type in the Animation Sequence panel, then FLUENT will save the solution animation file for you automatically. It will be saved in the specified Storage Directory, and its name will be the Name you specified for the sequence, with a `.cxa` extension (e.g., `pressure-contour.cxa`). In addition to the `.cxa` file, FLUENT will also save a metafile with a `.hmf` extension for each frame (e.g., `pressure-contour_2.hmf`). The `.cxa` file contains a list of the associated `.hmf` files, and tells FLUENT the order in which to display them.

If you selected Memory under Storage Type, then the solution animation file (`.cxa`) and the associated metafiles (`.hmf`) will be lost when you exit from FLUENT, unless you save them as described below.

You can save the animation sequence to a file that can be read back into FLUENT (see Section 22.17.4) when you want to replay the animation. As noted in Section 22.17.4, the solution animation file can be used for playback in FLUENT independent of the case and data files that were used to generate it.

To save a solution animation file (and the associated metafiles), select Animation Frames in the Write/Record Format drop-down list in the Playback panel, and click on the Write button. FLUENT will save a `.cxa` file, as well as a `.hmf` file for each frame of the animation sequence. The filename for the `.cxa` file will be the specified sequence Name (e.g., `pressure-contour.cxa`), and the filenames for the metafiles will consist of the specified sequence Name followed by a frame number (e.g., `pressure-contour_2.hmf`). All of the files (`.cxa` and `.hmf`) will be saved in the current working directory.

Hardcopy File

You can also generate a hardcopy file for each frame in the animation sequence. This feature allows you to save your sequence frames to hardcopy files used by an external animation program such as ImageMagick. As noted above, you can modify the view in the graphics window before you save the hardcopy files.

To save the animation as a series of hardcopy files, follow these steps:

1. Select Hardcopy Frames in the Write/Record Format drop-down list in the Playback panel.

2. If necessary, click on the Hardcopy Options... button to open the Graphics Hardcopy panel and set the appropriate parameters for saving the hardcopy files. (If you are saving hardcopy files for use with ImageMagick, for example, you may want to select the window dump format. See Section 3.12.1 for details.) Click Apply in the Graphics Hardcopy panel to save your modified settings.

! Do not click on the Save... button in the Graphics Hardcopy panel. You will save the hardcopy files from the Playback panel in the next step.

3. In the Playback panel, click on the Write button. FLUENT will replay the animation, saving each frame to a separate file. The filenames will consist of the specified sequence Name followed by a frame number (e.g., `pressure-contour_2.ps`), and they will all be saved in the current working directory.

MPEG File

It is also possible to save all of the frames of the animation sequence in an MPEG file, which can be viewed using an MPEG decoder such as mpeg_play. Saving the entire animation to an MPEG file will require less disk space than storing individual window dump files (using the hardcopy method), but the MPEG file will yield lower-quality images.

As noted above, you can modify the view in the graphics window before you save the MPEG file.

To save the animation to an MPEG file, follow these steps:

1. Select MPEG in the Write/Record Format drop-down list in the Playback panel.

2. Click on the Write button.

   FLUENT will replay the animation and save each frame to a separate scratch file, and then it will combine all the files into a single

MPEG file. The name of the MPEG file will be the specified sequence Name with a `.mpg` extension (e.g., `pressure-contour.mpg`), and it will be saved in the current working directory.

22.17.4 Reading an Animation Sequence

If you have saved an animation sequence to a solution animation file (as described in Section 22.17.3), you can read that file back in at a later time (or in a different session) and play the animation. Note that you can read a solution animation file into any FLUENT session; you do not need to read in the corresponding case and data files. In fact, you do not need to read in any case and data files at all before you read a solution animation file into FLUENT.

To read a solution animation file, click on the Read... button in the Playback panel. In the resulting Select File dialog box, specify the name of the file to be read.

22.18 Executing Commands During the Calculation

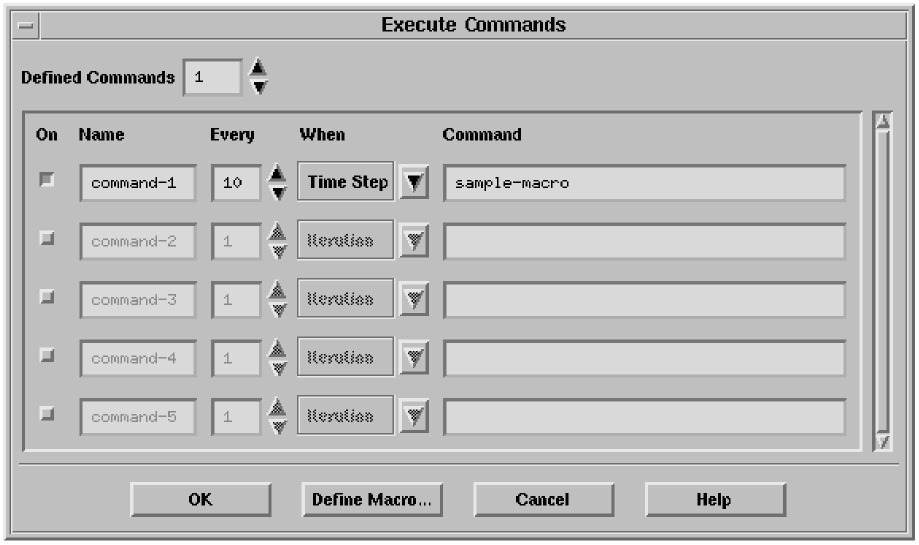
As described in Sections 22.16 and 22.17, respectively, you can report and monitor various quantities (e.g., residuals, force coefficients) and create animations of the solution while the solver is performing calculations. FLUENT also includes a feature that allows you to define your own command(s) to be executed during the calculation at specified intervals. For example, you can ask FLUENT to export data to a third-party package such as FIELDVIEW after every time step. You will specify a series of text commands or use the GUI to define the steps to be performed.

! If you want to save case or data files at intervals during the calculation, you must use the Autosave Case/Data panel (opened with the File/Write/Autosave... menu item). See Section 3.3.4 for details.

22.18.1 Specifying the Commands to be Executed

You will indicate the command(s) that you want the solver to execute at specified intervals during the calculation using the Execute Commands panel (Figure 22.18.1).

Solve ⟶ Execute Commands...



Defined Commands 1

On	Name	Every	When	Command
	command-1	10	Time Step	sample-macro
	command-2	1	Iteration	
	command-3	1	Iteration	
	command-4	1	Iteration	
	command-5	1	Iteration	

Figure 22.18.1: The Execute Commands Panel

The procedure is as follows:

1. Increase the Defined Commands value to the number of commands you wish to specify. As this value is increased, additional command entries will become editable. For each command, you will perform the following steps.
2. Turn on the On check button next to the command if you want it to be executed during the calculation. You may define multiple commands and choose to use only a subset of them by turning off the check button for those that you do not wish to use.
3. Enter a name for the command under the Name heading.
4. Indicate how often you want the command to be executed by setting the interval under Every and selecting Iteration or Time Step in

the drop-down list below When. (Time Step is a valid choice only if you are calculating unsteady flow.) For example, to execute the command every 10 iterations, you would enter `10` under Every and select Iteration under When.

! If you specify an interval in iterations, be sure to keep the Reporting Interval in the Iterate panel at its default value of 1.

5. Define the command by entering a series of text commands in the Command field, or by entering the name of a command macro you have defined (or will define) as described in Section 22.18.2.

! If the command to be executed involves saving a file, see Section 22.18.3 for important information.

22.18.2 Defining Macros

Macros that you define for automatic execution during the calculation can also be used interactively by you during the problem setup or post-processing. For example, if you define a macro that performs a certain type of adaption after each iteration, you can also use the macro to perform this adaption interactively.

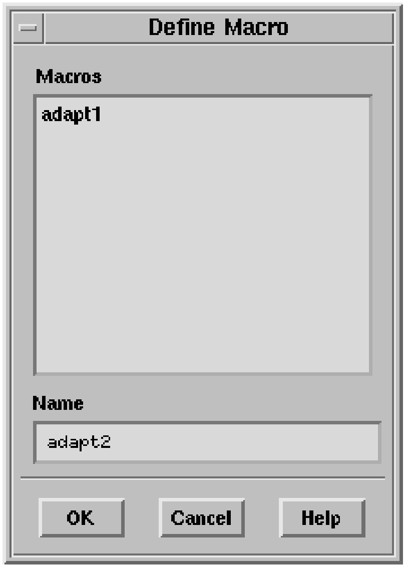
Definition of a macro is accomplished as follows:

1. In the Execute Commands panel, click on the Define Macro... button to open the Define Macro panel (Figure 22.18.2). Since this is a "modal" panel, the solver will not allow you to do anything else until you perform step 2, below.

2. In the Define Macro panel, specify a Name for the macro (e.g., `adapt1`) and click on OK. (The Define Macro... button in the Execute Commands panel will become the End Macro button.)

3. Perform the steps that you want the macro to perform. For example, if you want the macro to perform gradient adaption, open the Gradient Adaption panel, specify the appropriate adaption function and parameters, and click on Adapt to perform the adaption.

! If the command to be executed involves saving a file, see Section 22.18.3 for important information.



adapt1

adapt2

Figure 22.18.2: The Define Macro Panel

4. When you have completed the steps you wish the macro to perform, click on the End Macro button in the Execute Commands panel.

As noted above, once you have defined a macro for execution during the calculation, you can use it at any time. If you defined the macro called `adapt1` to adapt based on pressure gradient, you can simply type `adapt1` in the console (text) window to perform this adaption. This macro is independent of any text menus, so you need not move to a different text menu to use it. Macros can be saved to and read from files. To save all macros that are currently defined, use the `file/write-macros` text command. To read all the macros in a macro file, use the `file/read-macros` text command.

! A macro, like a journal file, is a simple record/playback function. It will therefore know nothing about the state in which it was recorded or the state in which it is being played back. You must be careful that all surfaces, variables, etc. that are used by the macro have been properly defined when you (or FLUENT) invoke the macro.

22.18.3 Saving Files During the Calculation

If the command to be executed during the calculation involves saving a file, you must include a special character in the file name when you enter it in the Select File dialog box so that the solver will know to assign a new name to each file it saves. You can number the files by iteration number or by time step. (These special characters for numbering files are also useful when you are saving window dumps for use in animations, as described in Section 3.12.1.) See Section 3.1.7 for details about these special characters for filenames.

! If you want to save case or data files at intervals during the calculation, you must use the Autosave Case/Data panel (opened with the File/Write/Autosave... menu item). See Section 3.3.4 for details.

22.19 Convergence and Stability

Convergence can be hindered by a number of factors. Large numbers of computational cells, overly conservative under-relaxation factors, and complex flow physics are often the main causes. Sometimes it is difficult to know whether you have a converged solution. In the following sections, some of the numerical controls and modeling techniques that can be exercised to enhance convergence and maintain stability are examined.

- Section 22.19.1: Judging Convergence
- Section 22.19.2: Step-by-Step Solution Processes
- Section 22.19.3: Modifying Algebraic Multigrid Parameters
- Section 22.19.4: Modifying FAS Multigrid Parameters
- Section 22.19.5: Modifying Multi-Stage Time-Stepping Parameters

You should also refer to Sections 22.7 and 22.8 for information about how the choice of discretization scheme or (for the segregated solver) pressure-velocity coupling scheme can affect convergence. Manipulation of under-relaxation parameters and multigrid settings to enhance convergence is discussed in Sections 22.9 and 22.19.3.

22.19.1 Judging Convergence

There are no universal metrics for judging convergence. Residual definitions that are useful for one class of problem are sometimes misleading for other classes of problems. Therefore it is a good idea to judge convergence not only by examining residual levels, but also by monitoring relevant integrated quantities such as drag or heat transfer coefficient.

For most problems, the default convergence criterion in FLUENT is sufficient. This criterion requires that the scaled residuals defined by Equation 22.16-4 or 22.16-9 decrease to 10^{-3} for all equations except the energy and P-1 equations, for which the criterion is 10^{-6}.

Sometimes, however, this criterion may not be appropriate. Typical situations are listed below.

- If you make a good initial guess of the flow field, the initial continuity residual may be very small leading to a large scaled residual for the continuity equation. In such a situation it is useful to examine the unscaled residual and compare it with an appropriate scale, such as the mass flow rate at the inlet.
- For some equations, such as for turbulence quantities, a poor initial guess may result in high scale factors. In such cases, scaled residuals will start low, increase as non-linear sources build up, and eventually decrease. It is therefore good practice to judge convergence not just from the value of the residual itself, but from its behavior. You should ensure that the residual continues to decrease (or remain low) for several iterations (say 50 or more) before concluding that the solution has converged.

Another popular approach to judging convergence is to require that the unscaled residuals drop by three orders of magnitude. FLUENT provides residual normalization for this purpose, as discussed in Sections 22.16.1 and 22.16.1. In this approach the convergence criterion is that the normalized unscaled residuals should drop to 10^{-3}. However, this requirement may not be appropriate in many cases:

- If you have provided a very good initial guess, the residuals may not drop three orders of magnitude. In a nearly-isothermal flow, for example, energy residuals may not drop three orders if the initial guess of temperature is very close to the final solution.

- If the governing equation contains non-linear source terms which are zero at the beginning of the calculation and build up slowly during computation, the residuals may not drop three orders of magnitude. In the case of natural convection in an enclosure, for example, initial momentum residuals may be very close to zero because the initial uniform temperature guess does not generate buoyancy. In such a case, the initial nearly-zero residual is not a good scale for the residual.

- If the variable of interest is nearly zero everywhere, the residuals may not drop three orders of magnitude. In fully-developed flow in a pipe, for example, the cross-sectional velocities are zero. If these velocities have been initialized to zero, initial (and final) residuals are both close to zero, and a three-order drop cannot be expected.

In such cases, it is wise to monitor integrated quantities, such as drag or overall heat transfer coefficient, before concluding that the solution has converged. It may also be useful to examine the un-normalized unscaled residual, and determine if the residual is small compared to some appropriate scale. Alternatively, the scaled residual defined by Equation 22.16-4 or 22.16-9 (the default) may be considered.

Conversely, it is possible that if the initial guess is very bad, the initial residuals are so large that a three-order drop in residual does not guarantee convergence. This is specially true for k and ϵ equations where good initial guesses are difficult. Here again it is useful to examine overall integrated quantities that you are particularly interested in. If the solution is unconverged, you may drop the convergence tolerance, as described in Section 22.16.1.

22.19.2 Step-by-Step Solution Processes

One important technique for speeding convergence for complex problems is to tackle the problem one step at a time. When modeling a problem with heat transfer, you can begin with the calculation of the isothermal flow. To solve turbulent flow, you might start with the calculation of laminar flow. When modeling a reacting flow, you can begin by computing a partially converged solution to the non-reacting flow, possibly including the species mixing. When modeling a discrete phase, such as fuel evaporating from droplets, it is a good idea to solve the gas-phase flow field first. Such solutions generally serve as a good starting point for the calculation of the more complex problems. These step-by-step techniques involve using the Solution Controls panel to turn equations on and off.

Selecting a Subset of the Solution Equations

FLUENT automatically solves each equation that is turned on using the Models family of panels. If you specify in the Viscous Model panel that the flow is turbulent, equations for conservation of turbulence quantities are turned on. If you specify in the Energy panel that FLUENT should enable energy, the energy equation is activated. Convergence can be sped up by focusing the computational effort on the equations of primary importance. The Equations list in the Solution Controls panel allows you to turn individual equations on or off temporarily.

$\boxed{\text{Solve}} \longrightarrow \boxed{\text{Controls}} \longrightarrow$Solution...

A typical example is the computation of a flow with heat transfer. Initially, you will define the full problem scope, including the thermal boundary conditions and temperature-dependent flow properties. Following the problem setup, you will use the Solution Controls panel to temporarily turn off the energy equation. You can then compute an isothermal flow field, remembering to set a reasonable initial value for the temperature of the fluid.

! This is possible only for the segregated solver; the coupled solvers solve the energy equation together with the flow equations in a coupled manner, so you cannot turn off the energy equation as described above.

When the isothermal flow is reasonably well converged, you can turn the energy equation back on. You can actually turn off the momentum and continuity equations while the initial energy field is being computed.

When the energy field begins to converge well, you can turn the momentum and continuity equations back on so that the flow pattern can adjust to the new temperature field. The temperature will couple back into the flow solution by its impact on fluid properties such as density and viscosity. The temperature field will have no effect on the flow field if the fluid properties (e.g., density, viscosity) do not vary with temperature. In such cases, you can compute the energy field without turning the flow equations back on again.

! If you have specified temperature-dependent flow properties, you should be sure that a realistic value has been set for temperature throughout the domain before disabling calculation of the energy equation. If an unrealistic temperature value is used, the flow properties dependent on temperature will also be unrealistic, and the flow field will be adversely affected. Instructions for initializing the temperature field or patching a temperature field onto an existing solution are provided in Section 22.13.

Turning Reactions On and Off

To solve a species mixing problem prior to solving a reacting flow, you should set up the problem including all of the reaction information, and save the complete case file. To turn off the reaction so that only the species mixing problem can be solved, you can use the Species Model panel to turn off the Volumetric Reactions.

$\boxed{\text{Define}} \longrightarrow \boxed{\text{Models}} \longrightarrow$ Species...

Once the species mixing problem has partially converged, you can return to the Species Model panel and turn the Volumetric Reactions option on again. You can then resume the calculation starting from the partially converged data.

For combustion problems you may want to patch a hot temperature in the vicinity of the anticipated reactions before you restart the calculation. See Section 22.13.2 for information about patching an initial value for a flow variable.

22.19.3 Modifying Algebraic Multigrid Parameters

The default algebraic multigrid settings are appropriate for nearly all problems, but in rare cases you may need to make minor adjustments. This section describes how to analyze the multigrid solver's performance to determine which parameters should be modified. It also provides examples of suggested settings for particular types of problems and explains how to set the multigrid parameters.

Analyzing the Algebraic Multigrid Solver

As mentioned earlier, in most cases the multigrid solver will not require any special attention from you. If, however, you have convergence difficulties or you want to minimize the overall solution time by using more aggressive settings, you can monitor the multigrid solver and modify the parameters to improve its performance. (The instructions below assume that you have already begun calculations, since there is no need to monitor the solver if you do not fit into one of the two categories above.)

To determine whether your convergence difficulties can be alleviated by modifying the multigrid settings, you will check if the requested residual reduction is obtained on each grid level. To minimize solution time, you will check to see if switching to a more powerful cycle will result in overall reduction of work by the solver.

Monitoring the Algebraic Multigrid Solver

The steps for monitoring the solver are as follows:

1. Set multigrid Verbosity to 1 or 2 in the Multigrid Controls panel.

   $\boxed{\text{Solve}}\longrightarrow\boxed{\text{Controls}}\longrightarrow$Multigrid...

2. Request a single iteration using the Iterate panel.

   $\boxed{\text{Solve}}\longrightarrow$Iterate...

If you set the verbosity to 2, the information printed in the console (text) window for each equation will include the following:

- equation name
- equation tolerance (computed by the solver using a normalization of the source vector)
- residual value after each fixed multigrid cycle or fine relaxation for the flex cycle
- number of equations in each multigrid level, with the zeroth level being the original (finest-level) system of equations

Note that the residual printed at cycle or relaxation 0 is the initial residual before any multigrid cycles are performed.

When verbosity is set to 1, only the equation name, tolerance, and residuals are printed.

A portion of a sample printout is shown below:

```
pressure correction equation:
 tol.  1.2668e-05
    0  2.5336e+00
    1  4.9778e-01
    2  2.5863e-01
    3  1.9387e-01

  multigrid levels:
     0    918
     1    426
     2    205
     3     97
     4     45
     5     21
     6     10
     7      4
```

Interpreting the Algebraic Multigrid Report

By default, the flex cycle is used for all equations except pressure correction, which uses a V cycle. Typically, for a flex cycle only a few (5–10) relaxations will be performed at the finest level and no coarse levels will be visited. In some cases one or two coarse levels may be visited. If the maximum number of fine level relaxations is not sufficient, you may want to increase the maximum number (as described in Section 22.19.3) or switch to a V cycle (as described in Section 22.19.3).

For pressure correction, a V cycle is used by default. If the maximum number of cycles (30 by default) is not sufficient, you can switch to a W cycle (using the Multigrid Controls panel, as described in Section 22.19.3). Note that for the parallel solver, efficiency may deteriorate with a W cycle. If you are using the parallel solver, you can try increasing the maximum number of cycles by increasing the value of Max Cycles in the Multigrid Controls panel under Fixed Cycle Parameters.

$\boxed{\text{Solve}} \longrightarrow \boxed{\text{Controls}} \longrightarrow$ Multigrid...

Changing the Maximum Number of Relaxations

To change the maximum number of relaxations, increase or decrease the value of Max Fine Relaxations or Max Coarse Relaxations in the Multigrid Controls panel (Figure 22.19.1) under Flexible Cycle Parameters.

$\boxed{\text{Solve}} \longrightarrow \boxed{\text{Controls}} \longrightarrow$ Multigrid...

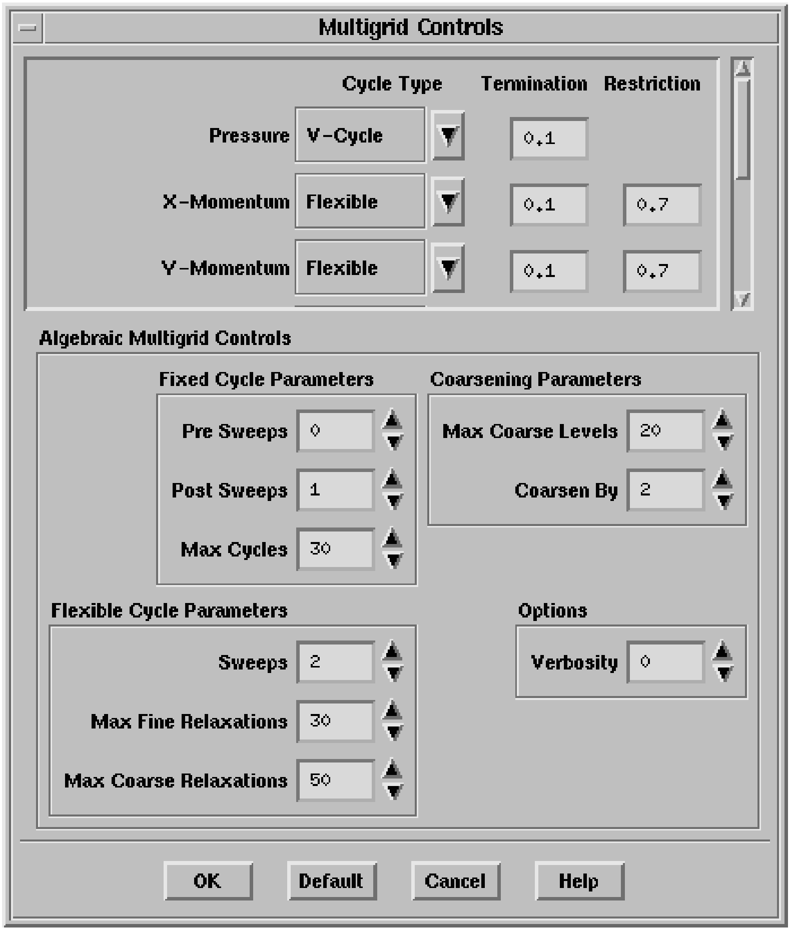
Specifying the Multigrid Cycle Type

By default, the V cycle is used for the pressure equation and the flex cycle is used for all other equations. (See Section 22.5.2 for a description of these cycles.) To change the cycle type for an equation, you will use the top portion of the Multigrid Controls panel (Figure 22.19.1).

For each equation, you can choose Flexible, V-Cycle, W-Cycle, or F-Cycle in the adjacent drop-down list.

Setting the Termination and Residual Reduction Parameters

When you use the flex cycle for an equation, you can control the multigrid performance by modifying the Termination and/or Restriction criteria for that equation at the top of the Multigrid Controls panel (Figure 22.19.1).


Multigrid Controls

	Cycle Type	Termination	Restriction
Pressure	V-Cycle	0.1	
X-Momentum	Flexible	0.1	0.7
Y-Momentum	Flexible	0.1	0.7

Algebraic Multigrid Controls

Fixed Cycle Parameters

Pre Sweeps	0
Post Sweeps	1
Max Cycles	30

Coarsening Parameters

Max Coarse Levels	20
Coarsen By	2

Flexible Cycle Parameters

Sweeps	2
Max Fine Relaxations	30
Max Coarse Relaxations	50

Options

Verbosity	0

Figure 22.19.1: The Multigrid Controls Panel

$\boxed{\text{Solve}}\longrightarrow\boxed{\text{Controls}}\longrightarrow$Multigrid...

The Restriction criterion is the residual reduction tolerance, β in Equation 22.5-14. This parameter dictates when a coarser grid level must be visited (due to insufficient improvement in the solution on the current level). With a larger value of β, coarse levels will be visited less often (and vice versa). The Termination criterion, α in Equation 22.5-15, governs when the solver should return to a finer grid level (i.e., when the residuals have improved sufficiently on the current level).

For the V, W, or F cycle, the Termination criterion determines whether or not another cycle should be performed on the finest (original) level. If the current residual on the finest level does not satisfy Equation 22.5-15, and the maximum number of cycles has not been performed, FLUENT will perform another multigrid cycle. (The Restriction parameter is not used by the V, W, and F cycles.)

Additional Algebraic Multigrid Parameters

There are several additional parameters that control the algebraic multigrid solver, but there will usually be no need to modify them. These additional parameters are all contained in the Multigrid Controls panel (Figure 22.19.1).

$\boxed{\text{Solve}}\longrightarrow\boxed{\text{Controls}}\longrightarrow$Multigrid...

Coarsening Parameters

For all multigrid cycle types, you can control the maximum number of coarse levels (Max Coarse Levels under Coarsening Parameters) that will be built by the multigrid solver. Sets of coarser simultaneous equations are built until the maximum number of levels has been created, or the coarsest level has only 3 equations. Each level has about half as many unknowns as the previous level, so coarsening until there are only a few cells left will require about as much total coarse-level coefficient storage as was required on the fine mesh. Reducing the maximum coarse levels will reduce the memory requirements, but may require more iterations to achieve a converged solution. Setting Max Coarse Levels to 0 turns off the algebraic multigrid solver.

Another coarsening parameter you can control is the increase in coarseness on successive levels. The Coarsen By parameter specifies the number of fine grid cells that will be grouped together to create a coarse grid cell. The algorithm groups each cell with its closest neighbor, then groups the cell and its closest neighbor with the neighbor's closest neighbor, continuing until the desired coarsening is achieved. Typical values are in the range from 2 to 10, with the default value of 2 giving the best performance, but also the greatest memory use. You should not adjust this parameter unless you need to reduce the memory required to run a problem.

Fixed Cycle Parameters

For the fixed (V, W, and F) multigrid cycles, you can control the number of pre- and post-relaxations (β_1 and β_3 in Section 22.5.2). Pre Sweeps sets the number of relaxations to perform before moving to a coarser level. Post Sweeps sets the number to be performed after coarser level corrections have been applied. Normally one post-relaxation is performed and no pre-relaxations are done (i.e., $\beta_3 = 1$ and $\beta_1 = 0$), but in rare cases, you may need to increase the value of β_1 to 1 or 2.

Returning to the Default Multigrid Parameters

If you change the multigrid parameters, but you then want to return to FLUENT's default settings, you can click on the Default button in the Multigrid Controls panel. FLUENT will change all settings to the defaults, and the Default button will become the Reset button. To get your settings back again, you can click on the Reset button.

22.19.4 Setting FAS Multigrid Parameters

For most calculations, you will not need to modify any FAS multigrid parameters once you have set the number of coarse grid levels. If, however, you encounter convergence difficulties, you may consider the following suggested procedures.

! Recall that FAS multigrid is used only by the coupled explicit solver.

Combating Convergence Trouble

Some problems may approach convergence steadily at first, but then the residuals will level off and the solution will "get stuck." In some cases (e.g., long thin ducts), this convergence trouble may be due to multigrid's slow propagation of pressure information through the domain. In such cases, you should turn off multigrid by setting Multigrid Levels to 0 in the Solution Controls panel (under Solver Parameters).

$\boxed{\text{Solve}}\longrightarrow\boxed{\text{Controls}}\longrightarrow$Solution...

"Industrial-Strength" FAS Multigrid

In some cases, you may find that your problem is converging, but at an extremely slow rate. Such problems can often benefit from a more aggressive form of multigrid, which will speed up the propagation of the solution corrections. For such problems, you can try the "industrial-strength" multigrid settings.

! These settings are very aggressive and assume that the solution information passed through the multigrid levels is somewhat accurate. For this reason, you should only attempt the procedure described here after you have performed enough iterations that the solution is off to a good start. Using "industrial-strength" multigrid too early in the calculation process—when the solution is far from correct—will not help convergence and may cause the calculation to become unstable, as very incorrect values are propagated quickly to the original grid. Note also that while these multigrid settings will usually reduce the total number of iterations required to reach convergence, they will greatly increase the computation time for each multigrid cycle. Thus the solver will be performing fewer but longer iterations.

The strategy employed is as follows:

- Increase the number of iterations performed on each grid level before proceeding to a coarser level
- Increase the number of iterations performed on each grid level after returning from a coarser level

- Allow full correction transfer from one level to the next finer level, instead of transferring reduced values of the corrections
- Do not smooth the interpolated corrections when they are transferred from a coarser grid to a finer grid

You can set all of the parameters for this strategy under FAS Multigrid Controls in the Multigrid Controls panel (Figure 22.19.2) and then continue the calculation.

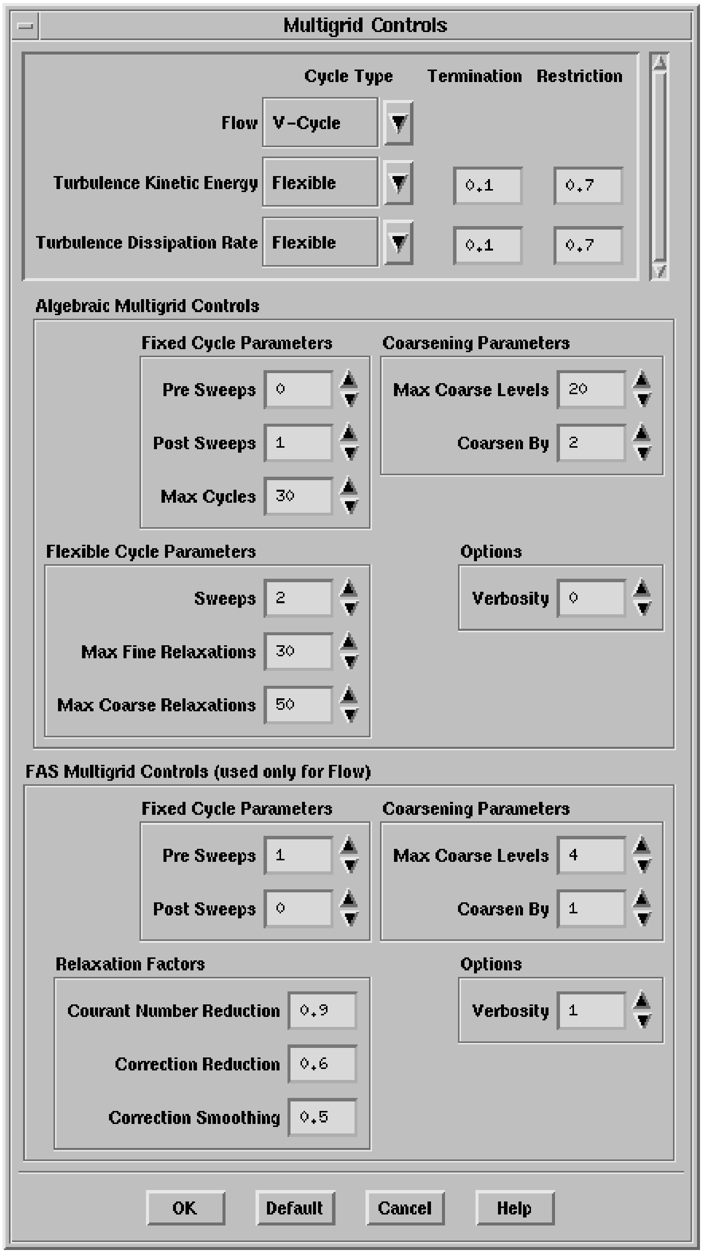
$\boxed{\text{Solve}} \longrightarrow \boxed{\text{Controls}} \longrightarrow$ Multigrid...

Increasing the number of iterations performed on each grid level before proceeding to a coarser level (the value of β_1 described in Section 22.5.2) will improve the solution passed from each finer grid level to the next coarser grid level. Try increasing the value of Pre Sweeps (under FAS Multigrid Controls, *not* under Algebraic Multigrid Controls) to 10.

Increasing the number of iterations performed on each level after returning from a coarser level will improve the corrections passed from each coarser grid level to the next finer grid level. Errors introduced on the coarser grid levels can therefore be reduced before they are passed further up the grid hierarchy to the original grid. Try increasing the value of Post Sweeps (under FAS Multigrid Controls, *not* under Algebraic Multigrid Controls) to 10.

By default, the full values of the multigrid corrections are not transferred from a coarser grid to a finer grid; only 60% of the value is transferred. This prevents large errors from transferring quickly up to the original grid and causing the calculation to become unstable. It also prevents a "good" solution from propagating quickly to the original grid, however. By increasing the Correction Reduction to 1, you can transfer the full values from coarser to finer grid levels, speeding the propagation of the solution and, usually, the convergence as well.

When the corrections on a coarse grid are passed back to the next finer grid level, the values are, by default, interpolated and then smoothed. Disabling the smoothing so that the actual value in a coarse grid cell is assigned to the fine grid cells that comprise it can also aid convergence.



	Cycle Type	Termination	Restriction
Flow	V-Cycle		
Turbulence Kinetic Energy	Flexible	0.1	0.7
Turbulence Dissipation Rate	Flexible	0.1	0.7

Algebraic Multigrid Controls

Fixed Cycle Parameters

Pre Sweeps	0
Post Sweeps	1
Max Cycles	30

Coarsening Parameters

Max Coarse Levels	20
Coarsen By	2

Flexible Cycle Parameters

Sweeps	2
Max Fine Relaxations	30
Max Coarse Relaxations	50

Options

Verbosity	0

FAS Multigrid Controls (used only for Flow)

Fixed Cycle Parameters

Pre Sweeps	1
Post Sweeps	0

Coarsening Parameters

Max Coarse Levels	4
Coarsen By	1

Relaxation Factors

Courant Number Reduction	0.9
Correction Reduction	0.6
Correction Smoothing	0.5

Options

Verbosity	1

Figure 22.19.2: The Multigrid Controls Panel

To turn off smoothing, set Correction Smoothing to 0. Large discontinuities between cells will be smoothed out implicitly as a result of the additional Post Sweeps performed.

Additional Multigrid Parameters

There are several additional parameters that control the multigrid solver, but there will usually be no need to modify them. These additional parameters are all contained in the Multigrid Controls panel.

$\boxed{\text{Solve}} \longrightarrow \boxed{\text{Controls}} \longrightarrow$ Multigrid...

Specifying the Multigrid Cycle Type

By default, the V cycle is used for the flow equations. (See Section 22.5.2 for a description of the available cycles.) To change to a W cycle, you can select it in the drop-down list next to Flow at the top of the Multigrid Controls panel (Figure 22.19.2).

Reducing the Time Step for Coarse Grid Levels

The Courant Number Reduction (at the bottom of the Multigrid Controls panel) sets the factor by which to reduce the Courant number for coarse grid levels (i.e., every level except the finest). Some reduction of time step (such as the default 0.9) is typically required because the stability limit cannot be determined as precisely on the irregularly shaped coarser grid cells.

22.19.5 Modifying Multi-Stage Time-Stepping Parameters

The most common parameter you will change to control the multi-stage time-stepping scheme is the Courant number. Instructions for modifying the Courant number are presented in Section 22.10, and procedures for modifying other less commonly changed parameters are provided in this section.

Using Residual Smoothing to Increase the Courant Number

Implicit residual smoothing (or averaging) is a technique that can be used to reduce the time step restriction of the solver, thereby allowing the Courant number to be increased. The implicit smoothing is implemented with an iterative Jacobi method, as described in Section 22.4.3. You can control residual smoothing in the Solution Controls panel.

$\boxed{\text{Solve}} \longrightarrow \boxed{\text{Controls}} \longrightarrow$Solution...

By default, the number of Iterations for Residual Smoothing is set to zero, indicating that residual smoothing is disabled. If you increase the Iterations counter to 1 or more, you can enter the Smoothing Factor. A smoothing factor of 0.5 with 2 passes of the Jacobi smoother is usually adequate to allow the Courant number to be doubled.

Changing the Multi-Stage Scheme

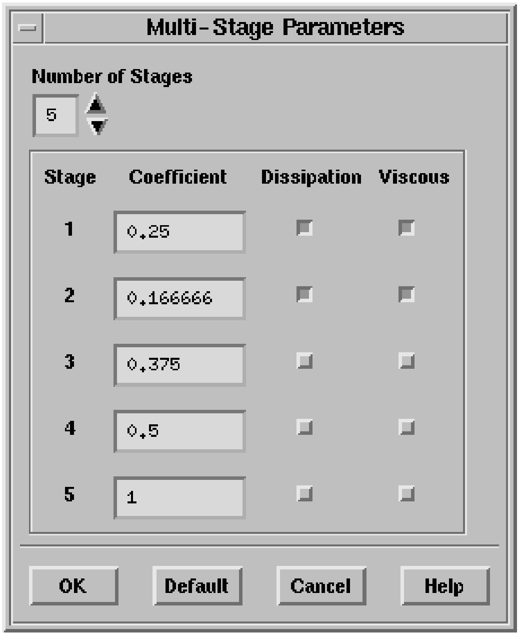
It is possible to make several changes to the multi-stage time-stepping scheme itself. You can change the number of stages and set a new multi-stage coefficient for each stage. You can also control whether or not dissipation and viscous stresses are updated at each stage. These changes are made in the Multi-Stage Parameters panel (Figure 22.19.3).

$\boxed{\text{Solve}} \longrightarrow \boxed{\text{Controls}} \longrightarrow$Multi-Stage...

! You should not attempt to make changes to FLUENT's multi-stage scheme unless you are very familiar with multi-stage schemes and are interested in trying a different scheme found in the literature.

Changing the Coefficients and Number of Stages

By default, the FLUENT multi-stage scheme uses 5 stages with coefficients of 0.25, 0.166666, 0.375, 0.5, and 1.0 for the first through fifth stages, respectively. You can decrease the number of stages using the arrow buttons for Number of Stages in the Multi-Stage Parameters panel. (If you want to increase the number of stages, you will need to use the text-interface command `solve/set/multi-stage`.) For each stage, you can modify the Coefficient. Coefficients must be greater than 0 and less than 1. The final stage should always have a coefficient of 1.



Number of Stages: 5

Stage	Coefficient	Dissipation	Viscous
1	0.25	✓	✓
2	0.166666	✓	✓
3	0.375	☐	☐
4	0.5	☐	☐
5	1	☐	☐

Figure 22.19.3: The Multi-Stage Parameters Panel

Controlling Updates to Dissipation and Viscous Stresses

For each stage, you can indicate whether or not artificial dissipation and viscous stresses are evaluated. If a Dissipation box is selected for a particular stage, artificial dissipation will be updated on that stage. If not selected, artificial dissipation will remain “frozen” at the value of the previous stage. If a Viscous box is selected for a particular stage, viscous stresses will be updated on that stage. If not selected, viscous stresses will remain “frozen” at the value of the previous stage. Viscous stresses should always be computed on the first stage, and successive evaluations will increase the “robustness” of the solution process, but will also increase the expense (i.e., increase the CPU time per iteration). For steady problems, the final solution is independent of the stages on which viscous stresses are updated.

Resetting the Multi-Stage Parameters

If you change the multi-stage parameters, but you then want to return to the default scheme set by FLUENT, you can click on the Default button in the Multi-Stage Parameters panel. FLUENT will change the values to the defaults and the Default button will become the Reset button. To get your values back again, you can click on the Reset button.