# CFD EXPERTS

Simulate the Future

WWW.CFDEXPERTS.NET

Ansys
2021 R2
Engineering What's Ahead.



# Fluent Beta Features Manual

Ansys
ANSYS, Inc.
Southpointe
2600 Ansys Drive
Canonsburg, PA 15317
ansysinfo@ansys.com
http://www.ansys.com
(T) 724-746-3304
(F) 724-514-9494

Release 2021 R2
July 2021

Published in the U.S.A.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1: Introduction

This document contains information about Ansys Fluent 2021 R2 beta features, which provide options for modeling and reporting that are outside of the normal scope of Ansys Fluent. These features are not always accessible through the standard menus and dialog boxes, and will require the following text user interface (TUI) command to enable them:

`define → beta-feature-access`

**Note:**

Please note that if you enable beta features in this case, use any beta features, and then disable beta features, the beta features you put into use may still be active, even though the text and graphical interfaces for these features may no longer be visible. It is therefore recommended that you save a separate copy of the case before any beta feature is activated. This will allow you to return to working on the case with only released features if you desire.

**Important:**

Note that beta features have not been fully tested and validated. Ansys, Inc. makes no commitment to resolve defects reported against these prototype features. However, your feedback will help us improve the overall quality of the product.

**Note:**

Beta features are not subject to our Class 3 error reporting system. In addition, we will not guarantee that the input files using this beta feature will run successfully when the feature is finally released so you may, therefore, need to modify the input files.

Included in the information about the beta features are references to related chapters and sections in the Ansys Fluent 2021 R2 Getting Started Guide, User's Guide, Theory Guide, UDF Manual, and Advanced Add-On Modules.

# Chapter 2: Files, Import and Export

## 2.1. Mapping Data for Fluid-Structure Interaction (FSI) Applications

When writing files for a finite element analysis (FEA) as part of a fluid-structure interaction problem in standalone Fluent, you can ensure that the forces mapped to the FEA mesh are conserved by performing the following steps:

1. Enable the beta feature access (as described in Introduction (p. 1)).

2. Read an FEA mesh, using either the **Read** button of the **Volume FSI Mapping** or **Surface FSI Mapping** dialog box, or the `file/fsi/read-fsi-mesh` text command.

3. Enable the conservation of the mapped forces by using the following text command:

   `file` → `fsi` → `conserve-force?`

# Chapter 3: Meshes

## 3.1. Smoothing Registers

If the surface between a marked and an unmarked region is wrinkled, you can use the `/adapt/smooth-register` TUI command to smooth it.

```
> /adapt/smooth-register
marking register id/name  [] 0

  118586 cells marked after smoothing step 1
  119718 cells marked after smoothing step 2
```

The smoothing is accomplished by marking additional cells at the boundaries of the specified marked register. The resulting collection of cells (i.e. the cells in the original register and the newly marked cells) are added to a new register. The original register is preserved. You can thus see which cells have been added to the marked region using register operations.

## 3.2. Meshing Mode Access

For 3D serial processing, you have the ability to switch from the solution mode of Fluent to the meshing mode at any point, even when a mesh or case file is in memory. By enabling beta feature access (Introduction (p. 1)), the following text command will always be available in the console, and can be used as described in the chapter on switching between meshing and solution modes in the Fluent Getting Started Guide:

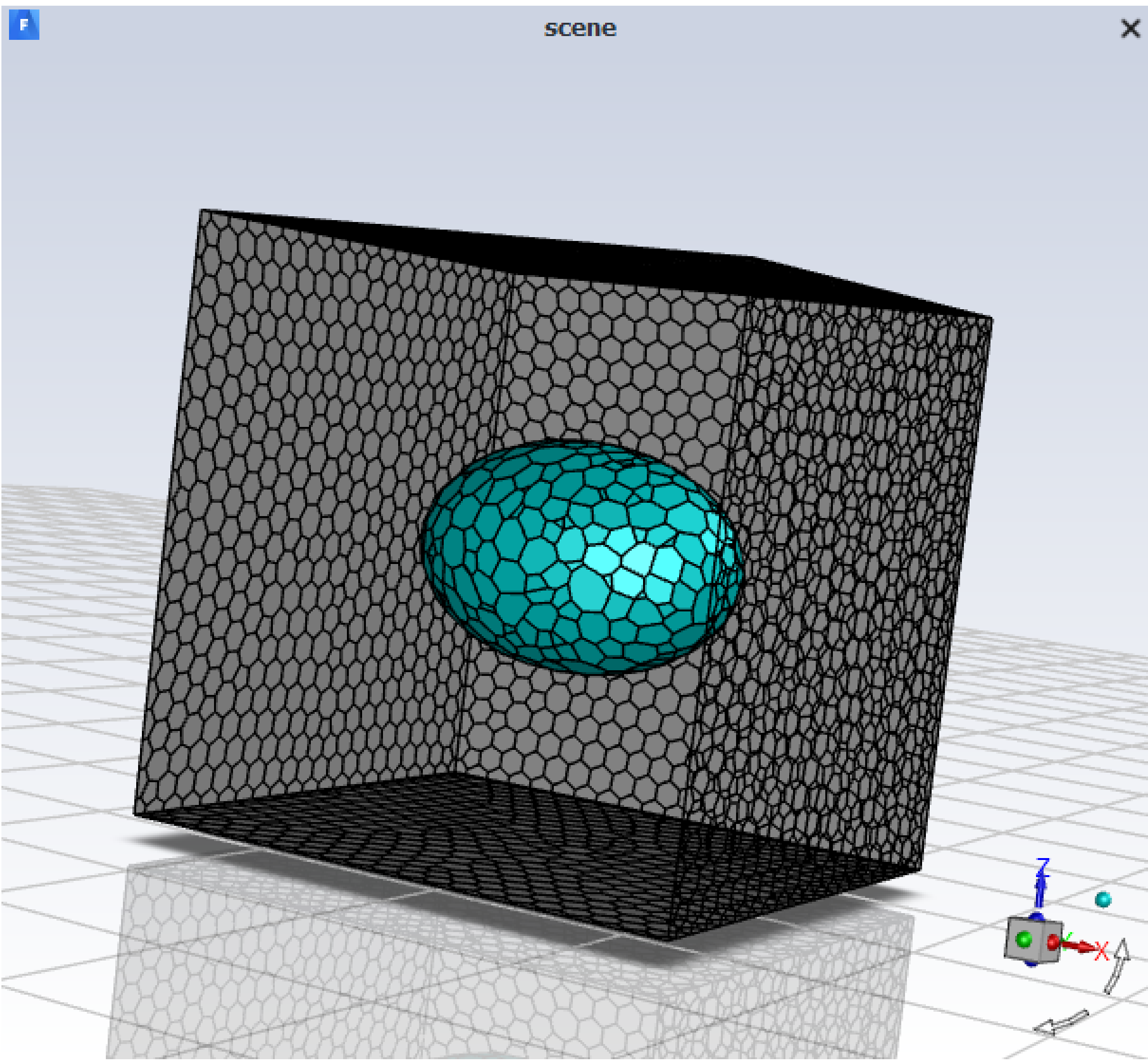
```
switch-to-meshing-mode
```

# Chapter 4: Surfaces

## 4.1. Ellipsoid Surface

With beta features enabled (as described in Introduction (p. 1)), you can create ellipsoid surfaces for displaying data.

**Figure 4.1: Example Ellipsoid Surface on a Model**




To create an ellipsoid surface:

1. Enable beta features by entering the `define/beta-feature-access/yes ok` text command in the Console.

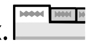
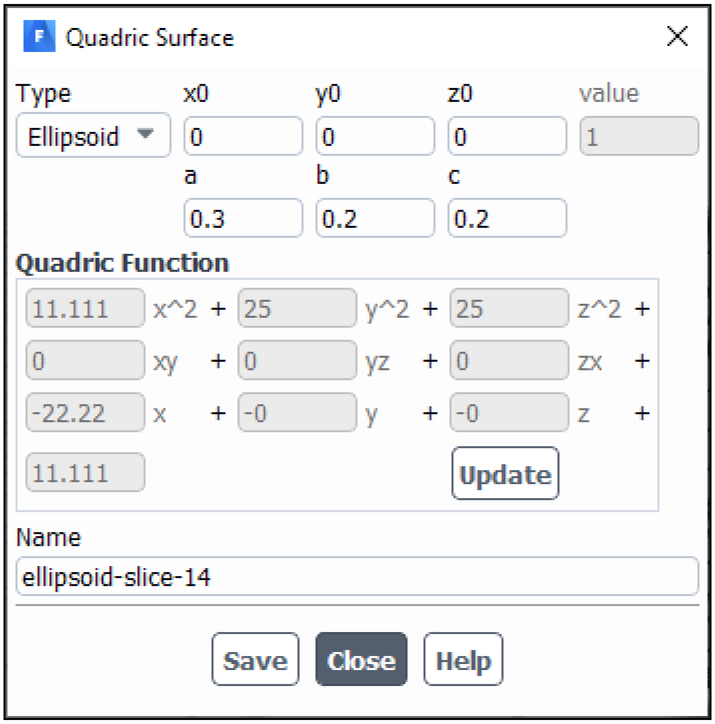
2. Open the **Quadric Surface** dialog box.  **Domain** → **Surfaces** → **Create** → **Quadric...**

**Figure 4.2: The Quadric Surface Dialog Box**




3. Select **Ellipsoid** from the **Type** drop-down list.

4. Provide values for the coordinates **x0**, **y0**, **z0** and the coefficients **a**, **b**, and **c**. These coordinates and coefficients are used in the ellipsoid surface equation: $\frac{x_0{}^2}{a^2}+\frac{y_0{}^2}{b^2}+\frac{z_0{}^2}{c^2}=1$.

5. Click **Create** to create the defined ellipsoid surface.

# Chapter 5: Boundary Conditions

In this chapter you will find descriptions of beta functionality for setting up boundary conditions.



## 5.1. Pressure Far-field Type

With beta features enabled (as described in Introduction (p. 1)), Fluent's density-based solver offers an alternative implementation of the pressure far-field boundary condition. This alternative option employs a Riemann-solver to evaluate the numerical-flux at the far-field boundary, thus providing the same level of accuracy and robustness as the interior faces. This treatment is a global application of the local tangency-correction used to regain robustness on boundary faces tangent to the imposed flow direction (Tangency Correction in the *Fluent User's Guide*). It is also advantageous for calculations where boundary-layers or wakes extend beyond the boundary of the computational domain. This flux-based pressure far-field can be selected with the following text command:

```
/define/boundary-condition/bc-settings/pressure-far-field/type?
```

## 5.2. Reference Temperature from a Boundary

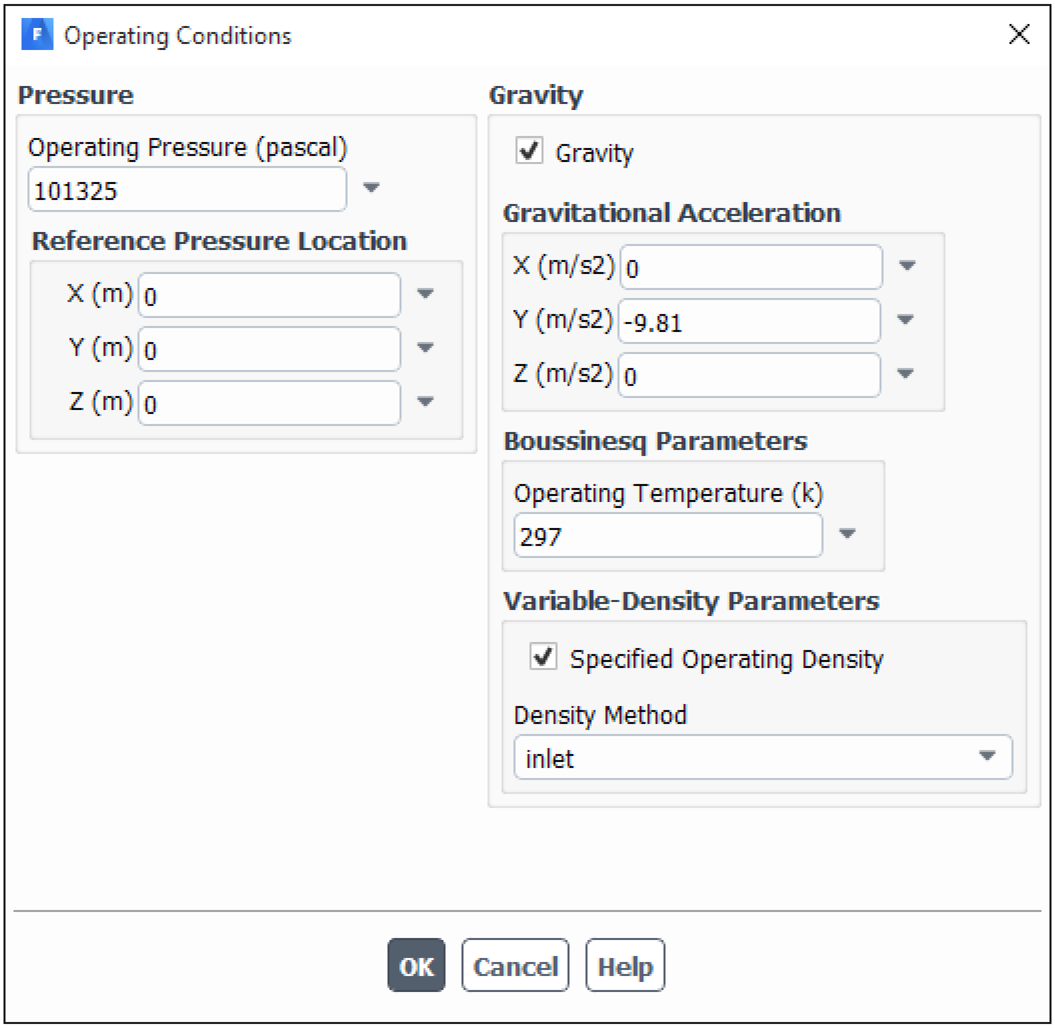
When any fluid material inside the domain is an **incompressible-ideal-gas** or **ideal-gas**, the option of specifying the **Density Method** will appear as a drop-down list in the **Operating Conditions** dialog box. Select one of the inlet boundaries (velocity inlet, mass-flow-inlet, pressure-inlet) for the calculation of the operating density. The temperature specified in the temperature tab of an inlet boundary dialog box will be used to calculate the operating density. If no boundary type is an **Inlet**, then Ansys Fluent will calculate the reference density using the default method.

**Important:**

- This option can be used only when you specify the temperature and/or species concentration on the boundary as **constant**.
- This option will not be available if the boundary has a profile or UDF for temperature.
- This option is only available for the pressure-based solver.

Specifying the inlet boundary for the calculation of reference density helps in predicting quiescent flows.

**Figure 5.1: The Operating Conditions Dialog Box**




After enabling beta feature access (Introduction (p. 1)), you can use the following text command:

`define → operating-conditions → use-inlet-temperature-for-operating-density`

Enter the `Zone-id/name [()]`.

# 5.3. Wall Roughness Settings for Turbulence Models

After enabling beta feature access (Introduction (p. 1)), the **Wall Roughness** settings in the **Momentum** tab of the **Wall** dialog box become available for some additional turbulence model combinations:

1. $\varepsilon$-equation models with enhanced wall treatment (EWT) or the Menter-Lechner (ML) near-wall treatment.

   The wall roughness model must work in wall function mode, which requires fully rough flows ($K_s^+$ > 90). In order to avoid a singularity in the downward shift of the velocity profiles for large roughness heights and low values of $y^+$, Fluent redefines the roughness height based on the mesh refinement (see section in the Fluent User's Guide,"Wall Roughness Effects in Turbulent Wall-Bounded Flows") for these near-wall treatments:

$$K_s^+ = \min(K_s^+, y^+)$$

The mesh requirement for rough walls in this case is $y^+ > K_s^+$, so that a reduction of $K_s^+$ for fine meshes (small $y^+$ values) is avoided.

The following turbulence models based on the $\varepsilon$-equation can be combined with rough walls and EWT or ML:

- all $k$-$\varepsilon$ models (that is, standard, RNG, and realizable)
- Reynolds stress model with the **Linear Pressure-Strain** option selected
- Detached eddy simulation (DES) model with the **Realizable k-epsilon** option selected

2. Reynolds stress models based on the $\omega$-equation

   Reynolds stress models based on the $\omega$-equation are using the same rough wall model as the two-equation $\omega$-based models like BSL or SST. However, no special calibration and only limited testing has been performed in combination with Reynolds stress models.

   In order to avoid a singularity in the downward shift of the velocity profiles for large roughness heights and low values of $y^+$, the wall is virtually shifted to 50% of the roughness element height. For more information about wall roughness for $\omega$-based models, see the section in the Fluent User's Guide,"Wall Roughness Effects in Turbulent Wall-Bounded Flows".

The following Reynolds stress models based on the $\omega$-equation can be combined with rough walls:

- **Stress-Omega**
- **Stress-BSL**

## 5.4. Minimal Pressure Reflection Boundary

The minimal pressure reflection (MPR) approach is based on the treatment for boundary conditions proposed by Azab and Mustafa [1] (p. 13). In Ansys Fluent, the MPR method has been modified using AI techniques in order to minimize pressure-wave reflections from the boundary, but not eliminate them completely. The amount of reflectiveness is carefully optimized in order not to contaminate the solution.

The MPR approach uses a linear system of equations consisting of Euler equations in full nonlinear multidimensional form, the MPR condition, and boundary conditions. The pressure variation at the MPR boundary can be expressed in terms of variations of conservative variables for any gas law:

$$\begin{cases} \Delta P = \Delta Q_1 \dfrac{\partial P}{\partial Q_1} + \Delta Q_2 \dfrac{\partial P}{\partial Q_2} + \Delta Q_3 \dfrac{\partial P}{\partial Q_3} + \Delta Q_4 \dfrac{\partial P}{\partial Q_4} + \Delta Q_5 \dfrac{\partial P}{\partial Q_5} \\ \Delta P = MPR_{var} \\ MPR_{var} = \Delta Q_1 \dfrac{\partial P}{\partial Q_1} + \Delta Q_2 \dfrac{\partial P}{\partial Q_2} + \Delta Q_3 \dfrac{\partial P}{\partial Q_3} + \Delta Q_4 \dfrac{\partial P}{\partial Q_4} + \Delta Q_5 \dfrac{\partial P}{\partial Q_5} \end{cases} \tag{5.1}$$

where

$$\begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \\ Q_5 \end{bmatrix} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho e_{tt} \end{bmatrix}$$

where $\rho$ is density, $u$, $v$, and $w$ are $X$, $Y$, and $Z$ velocity components, respectively, and $e_{tt}$ is total energy.

The term $MPR_{var}$ is used in the MPR condition to minimize the pressure wave reflection from the boundary were it is applied.

Note the following limitations when using the MPR treatment:

- The MPR treatment is available only for the density based solver.
- The material of the working fluid should be compressible.
- Currently the MPR treatment is available for the following boundary conditions:
  - pressure inlet
  - pressure outlet
  - velocity inlet

The procedure for assigning the MPR condition to selected boundary faces is as follows:

1. Enable beta features access as described in Introduction (p. 1).
2. Enter the following text command in the console:

   ```
   solve/set/advanced/non-reflecting-boundary-treatment
   ```

   The text submenu of the available boundary types is printed in the console:
   - `pressure inlet`
   - `pressure outlet`
   - `velocity inlet`
3. Select the boundary types where the MPR treatment will be applied during a simulation, and at the prompt, enable or disable the MPR treatment. For example:

   ```
   solve/set/advanced/non-reflecting-boundary-treatment pressure-inlet

   Enable minimal pressure-reflection treatment? [no] yes
   ```
4. For boundaries where you want to use the MPR treatment, enable the non reflecting option for the acoustic wave model either in appropriate boundary condition dialog boxes or in the text user interface (TUI).

# Bibliography

[1] M. Azab and M. I. Mustafa. "Numerical solution of inviscid transonic flow using hybrid finite volume-finite difference solution technique on unstructured grid". *Aerospace Sciences and Aviation Technology*. 14. 1–9. 2011.

# Chapter 6: Physical Properties

In this chapter you will find descriptions of beta functionality for defining physical properties of materials.



## 6.1. NIST Real Gas Model

### 6.1.1. Overview

In previous releases of Fluent, the NIST real gas model was available only through text commands and using special commands outside the general set up for material properties. This limitation has been removed as a beta feature, and the NIST real gas model can now be accessed in the same way as other material property options (through the **Create/Edit Materials** dialog box).

The NIST model itself remains unchanged with identical behavior as previous versions, so only the methods for activating the model have changed. The legacy text commands remain available for backwards compatibility but cannot be used simultaneously with the new methods.

### 6.1.2. Enhancements/Limitations

The new implementation overcomes several limitations of the legacy NIST real gas implementation and provides the following enhancements:

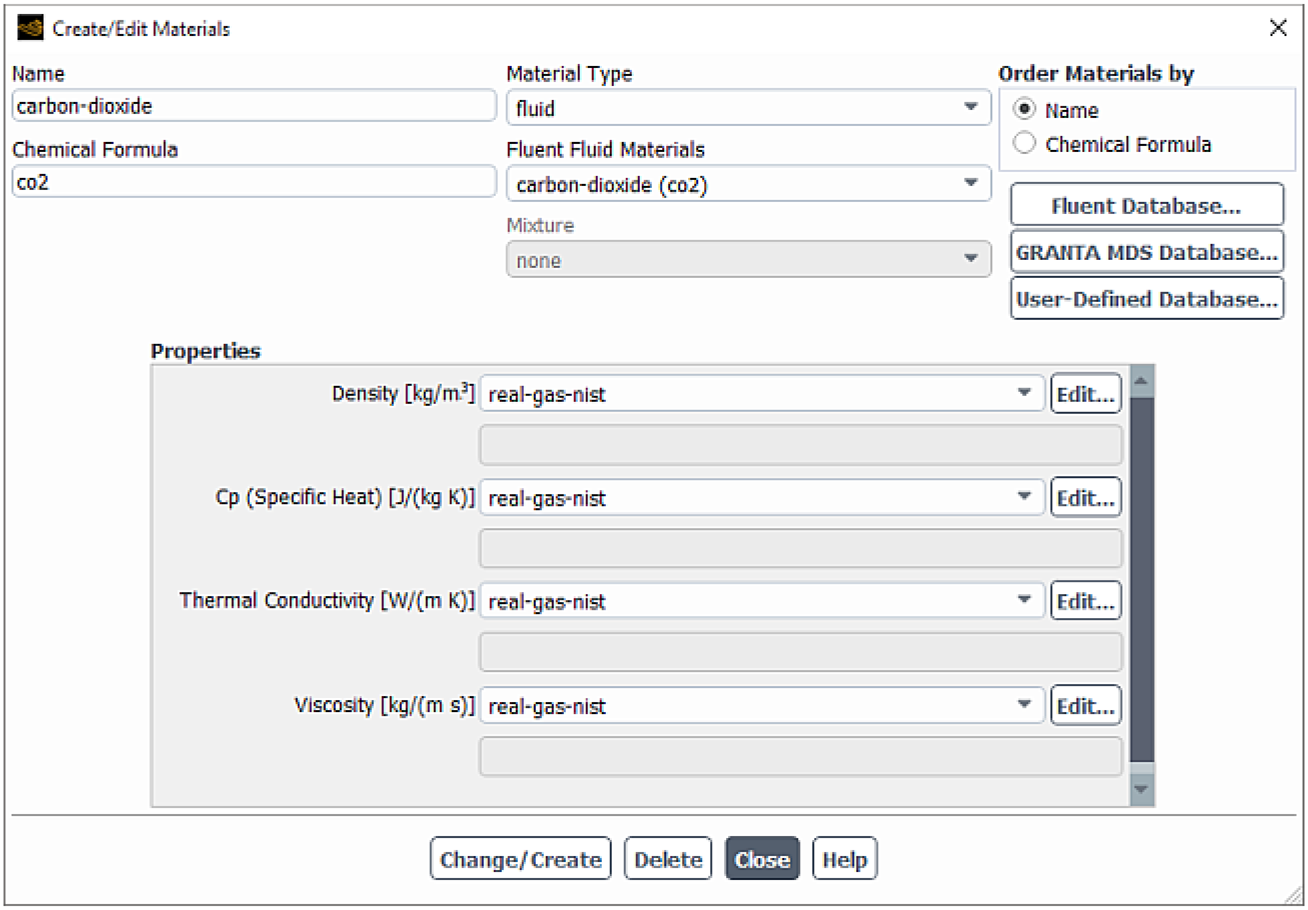
- NIST model can be enabled from **Create/Edit Materials** dialog box.
- Information on selected NIST data can be displayed in **NIST Fluid Data** dialog box.
- Materials with NIST model may have user-defined names.
- Materials with NIST model may be edited/modified.
- Multispecies model can enabled from **Species Model** panel.
- Multiple different NIST materials may be defined in the library. However, only a single NIST data set may be active in a simulation.
- Tabulation is not yet supported in the new workflow for the NIST model.

Except for the above, the limitations described in Limitations of the NIST Real Gas Models in the *Fluent User's Guide* remain unchanged.

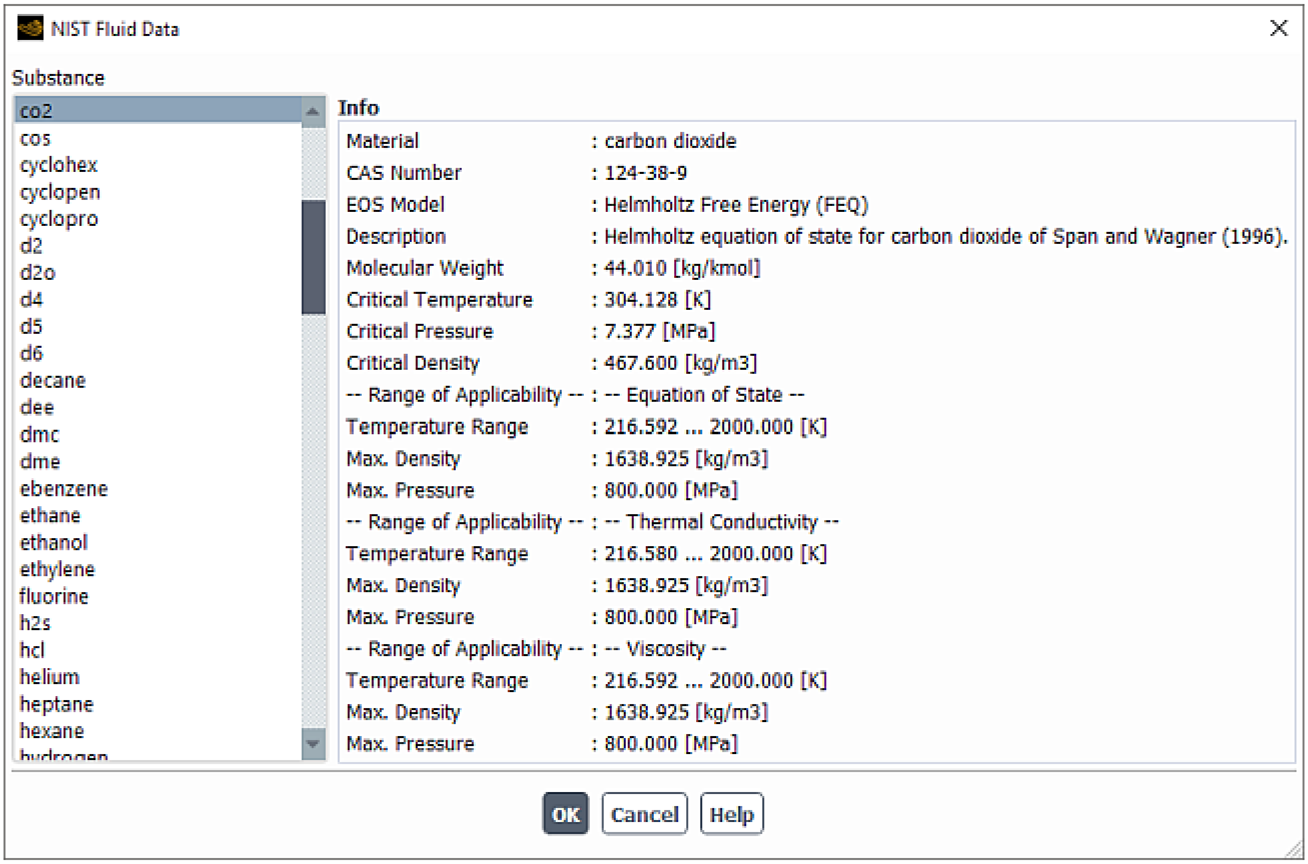
## 6.1.3. Using NIST Real Gas Material Properties

For single-component flow, you can enable the NIST real gas model by selecting **real-gas-nist** for **Density** on the **Create/Edit Materials** dialog box.

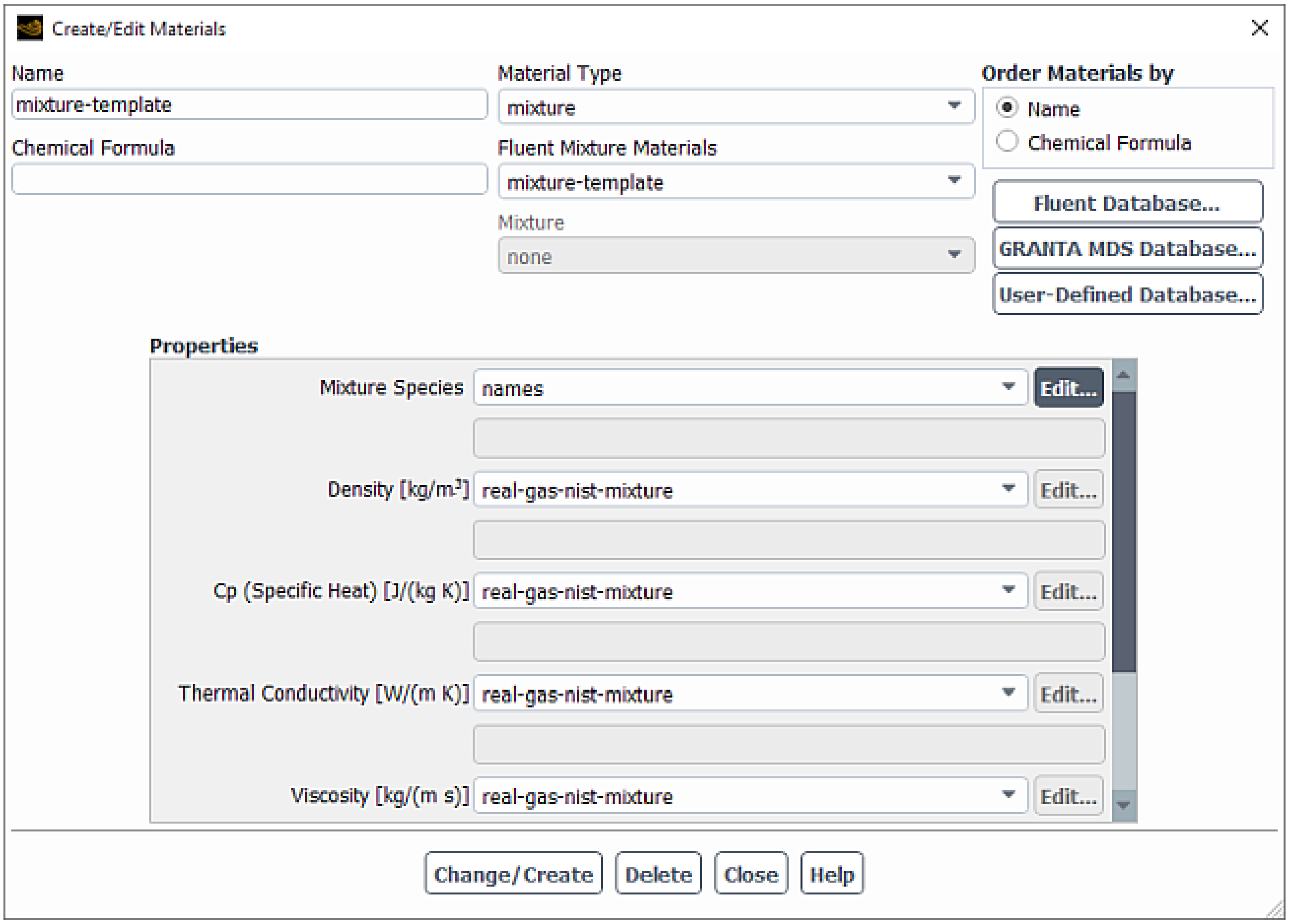
**Physics** → **Models** → **Materials** → **Create/Edit...**




When you select **Edit...** for **Density**, you will be prompted to select the appropriate NIST data from the **NIST Fluid Data** dialog box.




For multi-component flow, after enabling the **Species Transport** model , select **real-gas-nist-mixture** for **Density** in the mixture material definition.




Then, for each species, select the desired NIST data as you would for single-component flow.

# 6.2. Real Gas Property (RGP) Table Files

## 6.2.1. Creating RGP Files

You can create your own RGP file. For information about the RGP file contents and the file format, refer to the following sections:

- Real Gas Property (RGP) File Contents in the Ansys CFX—Solver Modeling Guide
- Real Gas Property (RGP) File Format in the Ansys CFX—Solver Modeling Guide

Real Gas Property (RGP) files that have been used in Ansys CFX can also be used in Ansys Fluent.

You can create RGP table files from the NIST REFPROP material properties using the RGP Generator command line tool. The RGP Generator produces RGP files for the liquid and vapor states of a pure substance. The liquid and vapor phases can be saved either individually in separate files or in two separate RGP data sections in the same single file. The generate tables include:

- Liquid superheat table

The superheat table points are spaced adaptively within the specified error tolerance. The liquid properties as a function of temperature and pressure will be available in your simulation once you read the liquid superheat tables into your session.

- Metastable gas states (incorporated in the superheat table section by default)

  The metastable states are extrapolated as far as possible. Beyond that, the liquid tables are filled with the vapor data, and vice versa.

  Generation of the metastable gas states can be turned off.

- Saturation table information (within the specified error tolerance)

  Two saturation table sections are included: one is discretized based on the liquid saturation data and the other based on the vapor saturation data. Typically, the two discretization data are not the same.

The commands can be executed in the System Coupling command console (Python), which is launched by running the following command:

- Windows:

  ```
  %AWP_ROOT212%\SystemCoupling\bin\systemcoupling.bat
  ```

- Linux:

  ```
  $AWP_ROOT212/SystemCoupling/bin/systemcoupling
  ```

Below is a typical example of commands for generating an RGP file:

```
import pyExt.RefProp as arp
fluidsPath = arp.getFluidsPath()
mat = arp.RefPropLib()

fluidList = ['fluids/water.fld']
mat.setup(fluidsPath, fluidList)

interpError = 1.0e-3
Tmin = 300.0
Tmax = 700.0
Pmin = 100000.0
Pmax = 25000000.0

# Generate both the liquid and vapor components, independent settings possible
h2ollow = arp.RGPLiquid("H2OL", Tmin, Tmax, Pmin, Pmax, interpError)
h2ovlow = arp.RGPVapor("H2OV", Tmin, Tmax, Pmin, Pmax, interpError)
arp.generateRGPfile("H2O-low.rgp", h2ollow, h2ovlow)

# Generate both the liquid and vapor components in the RGP file in one go
h2omed = arp.RGPSettings("H2O", Tmin, Tmax, Pmin, Pmax, 1.0e-4, arp.ADAPT_AUTO_TP)
arp.generateRGPfile("H2O-med.rgp", h2omed)

# Generate just the vapor component in the RGP file, for entire range
h2ovapor = arp.RGPVapor("H2OV", interpError)
arp.generateRGPfile("H2O-vapor.rgp", h2ovapor)
```

# Chapter 7: Heat Transfer

This chapter contains information relating to heat transfer features implemented as beta features in Ansys Fluent 2021 R2.



## 7.1. Shell Conduction with Non-Conformal Interfaces

After enabling beta feature access, as described in Introduction (p. 1), you can create shells on non-conformal interfaces (NCI).

The following is a list a known limitations:

- Reading a `.cas` file with saved shells on NCI into Fluent Meshing may cause it to crash.
- Switching from Solution Mode to Meshing Mode may crash if the case has shells on NCI.
- A shell with NCI may produce an incorrect solution if the secondary component of the temperature gradient (the gradient in the direction along the face separating the two cells) is included in the shell zone. In order to exclude this component, set the boolean rpvar `temperature/shell-secondary-gradient?` to `#f`.
- Shell NCI is supported for the default one-to-one auto pairing mesh interface method but not the non-default many-to-many interface.

## 7.2. Improving Convergence for Two-Way Thermal Coupling Simulations

Two-way thermal coupling cases may experience slow convergence when the near wall temperature value used in the heat transfer coefficient calculation is close to the wall temperature and is not a good representation of the free stream temperature. This typically occurs when the first node of the mesh is well within the boundary layer.

Convergence can be improved for such cases by allowing Ansys Fluent to send System Coupling a heat transfer coefficient value based on a constant, user-specified reference temperature. Note that this option results in the heat transfer coefficient having the same definition as the field variable **Surface Heat Transfer Coef.**; for details, see the Fluent User's Guide.

---

**Note:**

Basing the heat transfer coefficient value on a constant reference temperature may not be suitable when there is a strong variation in the free stream temperature along the coupled wall.

---

To improve the rate of convergence for your two-way thermal coupling case, perform the following steps:

1. Set up your coupled simulation with Fluent in Workbench, using a System Coupling component.

2. In Fluent, enable beta feature access, as described in Introduction (p. 1).

3. Specify that the heat transfer coefficient is based on a reference temperature by using the following text command:

   `define` → `models` → `system-coupling-settings` → `use-tref-in-htc-calculation?`

4. Define the reference temperature in the **Temperature** field of the **Reference Values** task page. It is recommended that you use a value that is close to the fluid free stream temperature.

   **Setting Up Physics** → **Solver** → **Reference Values...**

# Chapter 8: Heat Exchangers

## 8.1. Alternate Formulation for the Dual Cell Heat Exchanger

It is a well known fact that the dual cell model depends on the resolution of the core meshes. If the core mesh is very coarse, then accuracy is severely affected. Make sure you first enable beta feature access, as described in Introduction (p. 1), then activate the alternate formulation for heat transfer using the following text command:

`define` → `models` → `heat-exchanger` → `dual-cell-model` → `alternate-formulation?`

The results obtained using the alternate formulation is mesh independent and gives a reliable solution even on very coarse meshes. Please note that the default formulation and alternate formulation results are comparable on a sufficiently fine core mesh. Also the alternate formulation should not be used for non-matching core meshes.

For background information about the dual cell heat exchanger, see the section on using the dual cell heat exchangers in the Fluent User's Guide).

# Chapter 9: Turbulence

This chapter contains information relating to turbulence models implemented as beta features in Ansys Fluent 2021 R2.



## 9.1. Estimation of the LES Resolution Quality

Two output fields have been added to help you in the estimation of resolution quality in a scale-resolving simulation of a turbulent flow. **LES Resolution Estimate**, in the **Turbulence...** category, is intended to be visualized in a preliminary RANS simulation, which will later be continued using a scale-resolving model. The value of this field is equal to the ratio of the local turbulence length scale, obtained in the RANS simulation, to the local mesh cell size. Mesh regions, where this value is equal to or higher than 5, may be considered as sufficiently refined for a scale-resolving simulation. This preliminary estimation of mesh resolution quality relies on the assumption that the RANS solution correctly represents the time-averaged properties of a considered turbulent flow.

Another output field, **LES Resolution Quality**, in the **Unsteady Statistics...** category, becomes available when time statistics are computed during a scale-resolving simulation. This field characterizes the achieved resolution quality as the resolved portion of the turbulence kinetic energy:

$$q=k_r/\left(k_r+k_m\right) \tag{9.1}$$

where the resolved kinetic energy is directly computed from the mean square velocity fluctuations

$$k_r=0.5\left(\overline{u'u'}+\overline{v'v'}+\overline{w'w'}\right) \tag{9.2}$$

and the time-averaged modeled subgrid-scale kinetic energy is evaluated based on the subgrid-scale eddy viscosity $\mu_t$, density $\rho$, and strainrate magnitude $S$:

$$k_m=\overline{\frac{1}{0.3}\frac{\mu_t}{\rho}S} \tag{9.3}$$

A **LES Resolution Quality** that is 0.8 or higher normally indicates a sufficiently resolved turbulent flow solution.

In the current beta feature status, these two output fields are not yet available with all models. For example, the first field, **LES Resolution Estimate**, is not available with the Spalart-Allmaras model.

## 9.2. Combination of SDES/SBES model with (k,ε)-turbulence models

The hybrid RANS-LES models **Shielded Detached Eddy Simulation** (SDES) and **Stress-Blended Eddy Simulation** (SBES), which have both an improved shielding function compared to DDES / IDDES, are a released functionality only for some turbulence models based on the ω-equation and the Realizable k-ε turbulence model with scalable wall functions or enhanced wall treatment. The following combinations with k-ε turbulence models are available as beta functionality:

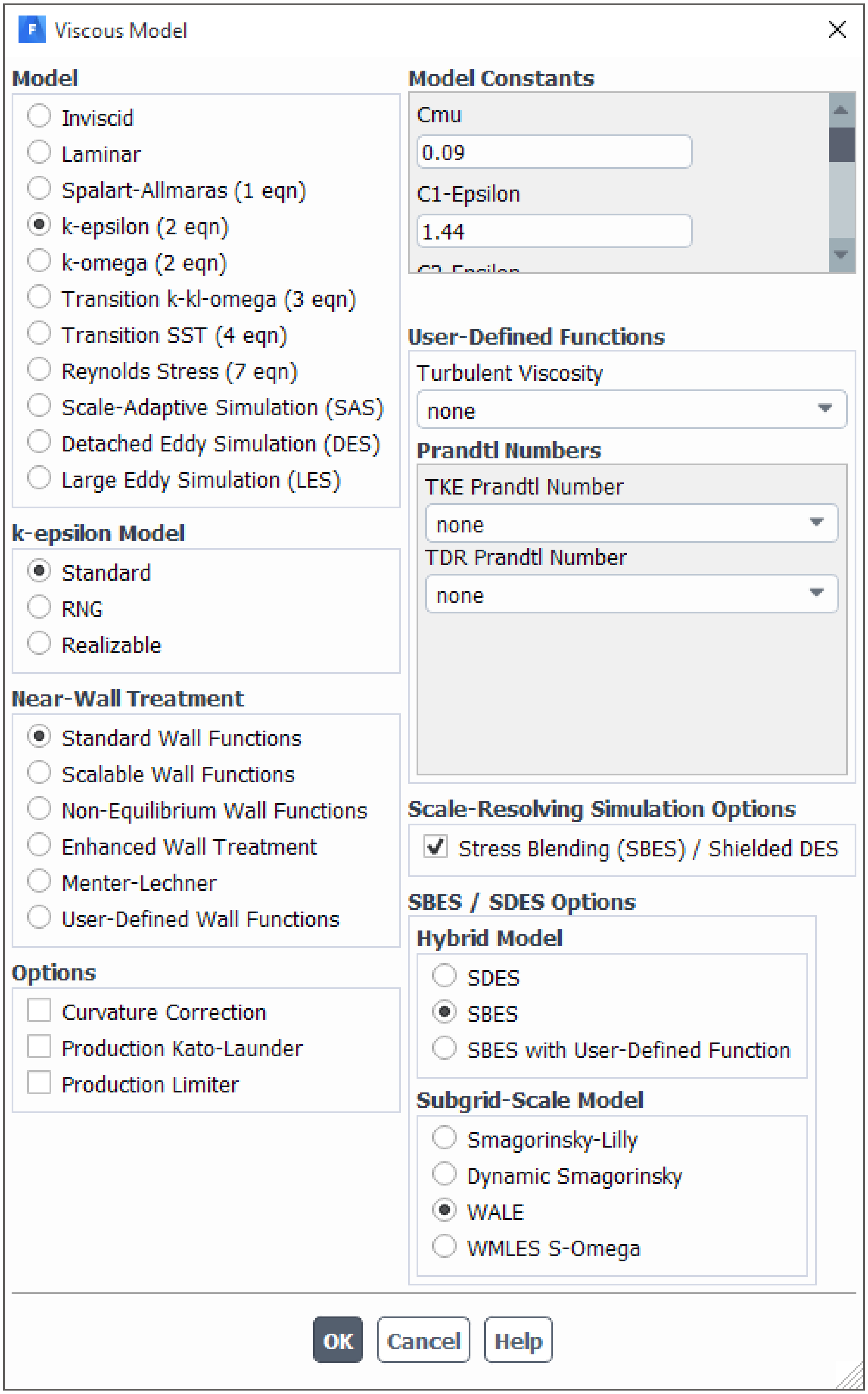
- Standard k-ε model with all near wall treatments
- RNG k-ε model with all near wall treatments
- Realizable k-ε model with all near wall treatments other than scalable wall functions or enhanced wall treatment

### 9.2.1. Accessing the SDES/SBES model with (k,ε)-turbulence models

Both models are available for transient cases with (k,ε)-turbulence models after enabling beta feature access, as described in Introduction (p. 1).

They can be accessed in the **Viscous Model** dialog box in the **Scale-Resolving Simulation Options** group box.

**Figure 9.1: The Viscous Model Dialog Box with SBES Enabled**




The corresponding TUI command to enable the SBES/SDES model is:

```
/define/models/viscous/turbulence-expert/turb-add-sbes-sdes? yes
```

The **Hybrid Model** option can be specified using:

```
/define/models/viscous/turbulence-expert/sbes-sdes-hybrid-model
```

For SBES in addition a **Subgrid-Scale Model** can be chosen by:

```
/define/models/viscous/turbulence-expert/sbes-sgs-option
```

### 9.2.2. Known Limitations

Although all near-wall treatments are allowed, the testing has been restricted to **Scalable Wall Functions** and **Enhanced Wall Treatment**.

## 9.3. Near-Wall Treatment for the Porous Media Interface

Two models are available in Ansys Fluent for the treatment of porous media:

- The (full) porous media model
- The ‘porous jump’ condition

The porous media model is applied in a cell zone. Several input parameters can be specified to determine the pressure loss in the flow. The ‘porous jump’ condition is a 1D simplification of the porous media model and is applied to a face zone.

In the porous medium the standard conservation equations for turbulence quantities are solved by default. The turbulence inside the porous medium is treated as though the solid medium has no effect on the turbulence production and dissipation rates. A detailed description of these models can be found in the Ansys Fluent User’s Guide.

An enhancement to the ‘porous jump’ condition is now available as a beta feature, allowing you to enable a turbulent wall treatment at the fluid side of the interface. When you assume, for example, turbulent fluid flow over a porous medium, then the porous medium has an effect on the fluid similar to a wall at the interface depending on its porosity. This beta option has been introduced to include the effects of the porous material on the fluid side of such an interface. An Example (p. 29) shows the combination of the beta ‘porous jump’ option with a porous media zone.

The turbulent near-wall treatment on the fluid side of a fluid/porous media interface has mainly two components:

- In the momentum equation, the wall shear stress is introduced at the interface.
- In the transport equation for the specific dissipation rate $\omega$ or dissipation rate $\varepsilon$ wall values are prescribed.

Furthermore, the diffusion term is set to zero at the interface, and the near-wall distance is updated to obtain a proper wall treatment.

The near-wall treatment which is used at the interface behaves identical to that of a solid wall, also with respect to y+.

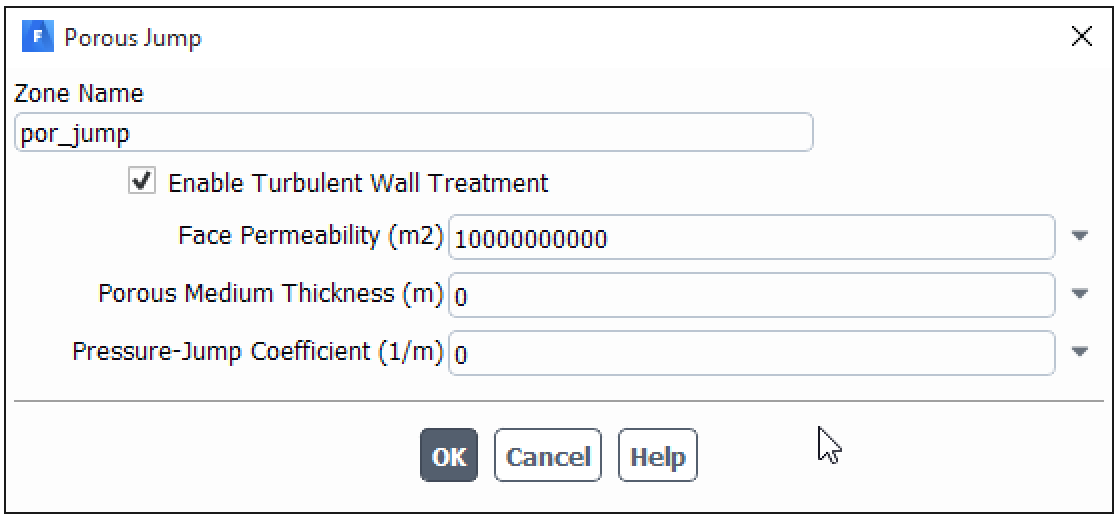
### 9.3.1. Accessing the Turbulent Wall Treatment Option

Near-wall treatment for the porous media interface is available after enabling beta feature access, as described in Introduction (p. 1).

The **Enable turbulent wall treatment** option in the **Porous Jump** dialog box (see Figure 9.2: The Porous Jump Dialog Box (p. 29)) is used to enable or disable a turbulent near-wall treatment on the fluid side of a 'porous jump' interface. In the **Viscous Model** dialog box, different near-wall treatments can be selected depending on the turbulence model. The selected treatment is then applied both for solid walls and at the fluid side of the interface.

The **Enable turbulent wall treatment** option can be used together with the **k-epsilon** and **k-omega** turbulence models, **SST** and **Transition SST**, **Reynolds Stress** models, **Detached Eddy Simulation (DES)** and **Scale-Adaptive Simulation (SAS)**.

**Figure 9.2: The Porous Jump Dialog Box**




### 9.3.2. Example

As outlined in the introduction, the purpose of the **Enable turbulent wall treatment** option is to include the effects of the porous material on the turbulent flow, for example at an interface between a fluid and a porous medium. Figure 9.3: Setup of Two Channel Flows Separated by Wall / Porous Jump Interface; Color Denotes Contours of the Streamwise Velocity Component. (p. 30) shows the setup of two channel flows, which are separated by a wall in the first part and by a 'porous jump' interface in the larger second part. The Reynolds number based on the centerline velocity and channel half height in each channel is $5e^{+5}$.

The lower zone in the second part is defined as a cell zone where the porous media model has been activated. A viscous loss term has been specified with a small viscous resistance of $14000m^{-2}$ in the streamwise direction. The interface between the lower porous media zone and the upper pure fluid zone has been defined as 'porous jump' with a medium thickness of zero. Therefore, no pressure jump occurs across the interface. It is only used to enable the turbulent wall treatment at the fluid side of the interface. Two simulations have been performed with and without the turbulent wall treatment at the interface in order to investigate the influence.

Inlet boundary condition profiles are prescribed for the velocity and the turbulence quantities which have been obtained from a 1D periodic channel flow. The outlet is specified as 'pressure outlet'. The grid is refined both to the walls (y+ ~ 2) and to the interface. The SST turbulence model has been used for both simulations.

**Figure 9.3: Setup of Two Channel Flows Separated by Wall / Porous Jump Interface; Color Denotes Contours of the Streamwise Velocity Component.**

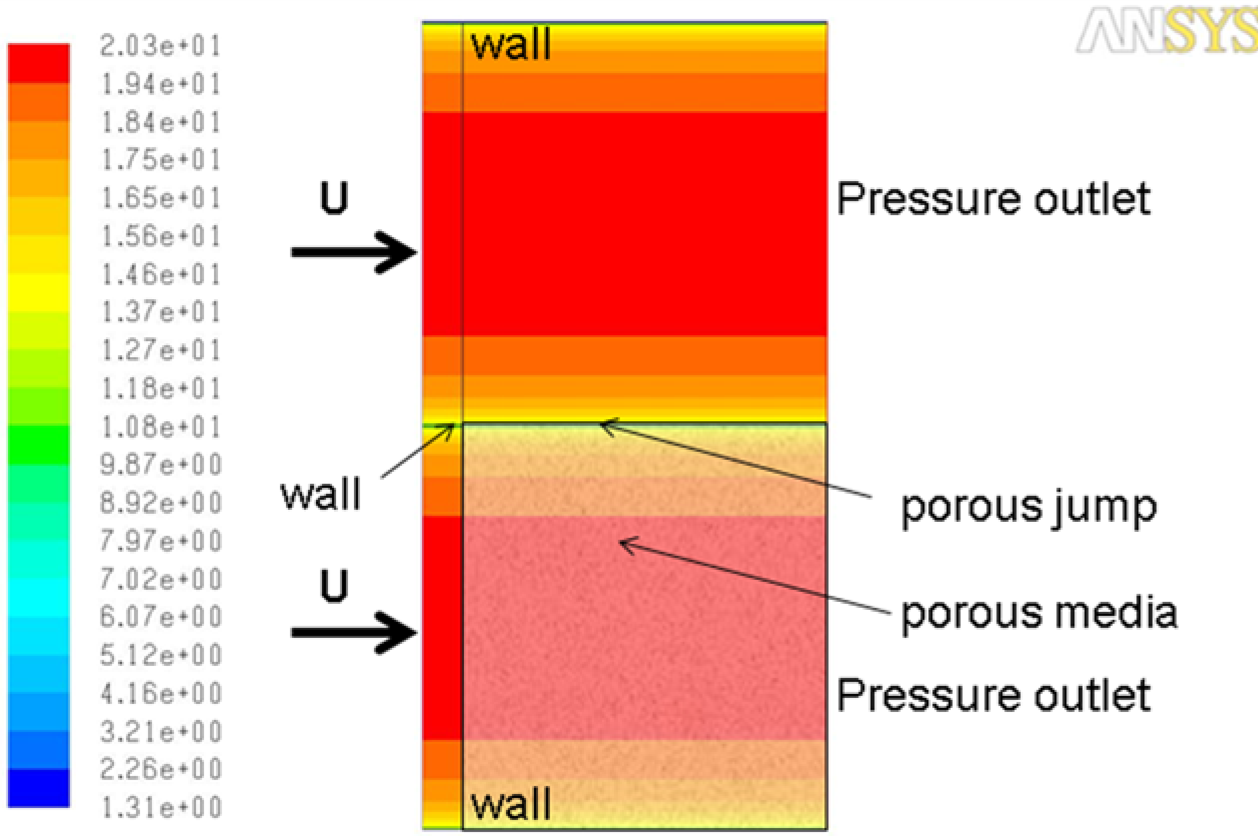



Figure 9.4: Profiles of the Streamwise Velocity Component Near the Outlet at Position x = 0.9 m Without (top) and With (bottom) Near-Wall Treatment at the Interface. Red Denotes the Pure Fluid Side, Black Represents the Side of the Porous Media (p. 31) shows profiles of the streamwise velocity component at position x = 0.9 m near the outlet without (top) and with (bottom) near-wall treatment at the interface. The profiles in the pure fluid zone are colored in red. The profiles in the zone where the porous media model is activated are colored in black.

The inclusion of the wall shear stress in the momentum equation has reduced the streamwise velocity at the interface on the fluid side. In the figures cell center, values have been plotted to allow a direct comparison with the effect caused by the walls. The effect of the new treatment at the interface is similar to the influence of the upper wall.

The profiles of the turbulent viscosity ratio are shown in Figure 9.5: Profiles of the Turbulent Viscosity Ratio Near the Outlet at Position x=0.9m Without (top) and With (bottom) Near-Wall Treatment at the Interface. Red Denotes the Pure Fluid Side, Black Represents the Side of the Porous Media (p. 32). In this simulation, a fine grid has been used which allows you to resolve the viscous sublayer. The eddy viscosity ratio is nearly zero close to the walls. The interface shows the same effect due to the additional source term in the $\omega$-equation when the near-wall treatment has been enabled and reduces the turbulent viscosity ratio.

**Figure 9.4: Profiles of the Streamwise Velocity Component Near the Outlet at Position x = 0.9 m Without (top) and With (bottom) Near-Wall Treatment at the Interface. Red Denotes the Pure Fluid Side, Black Represents the Side of the Porous Media**

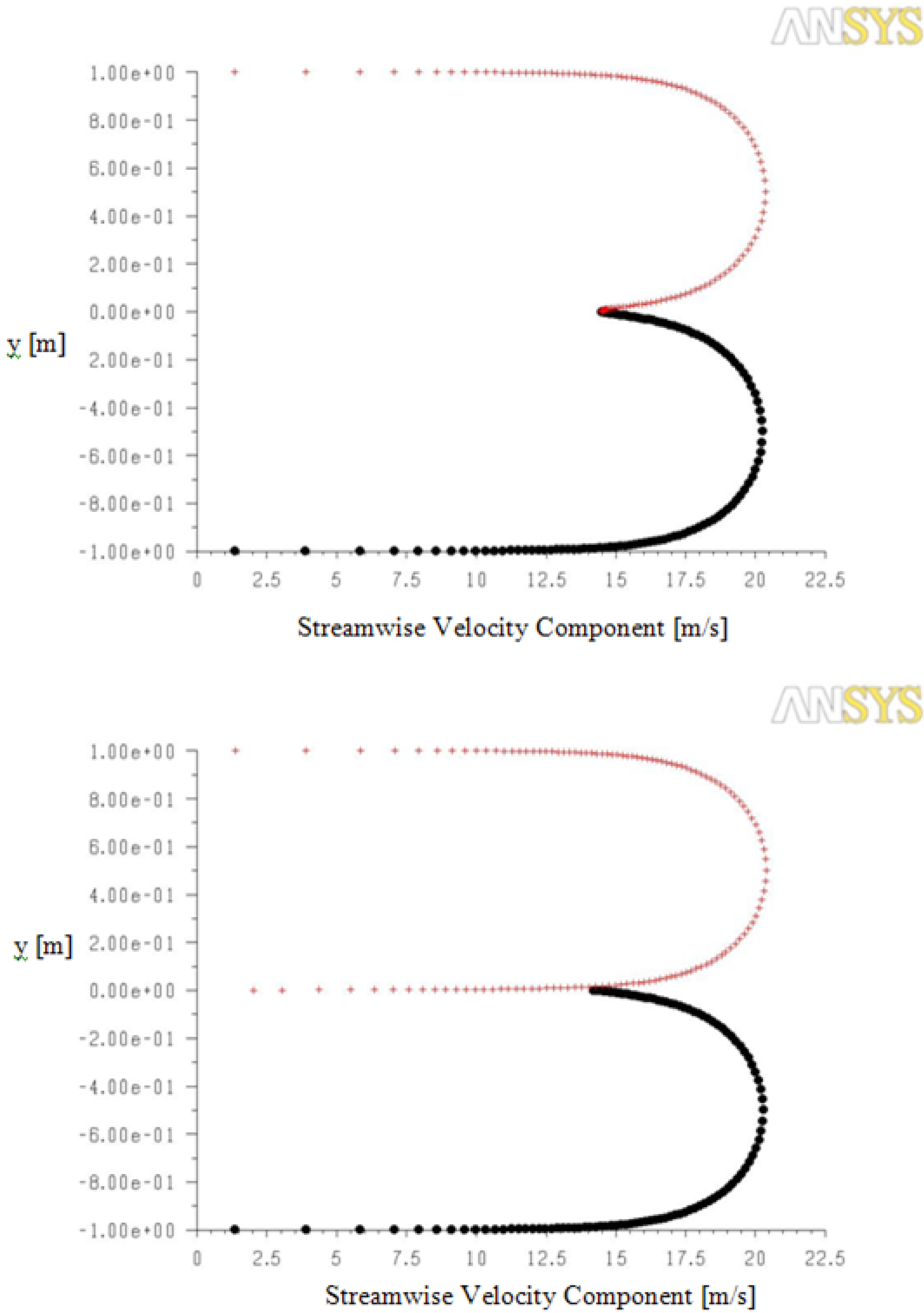
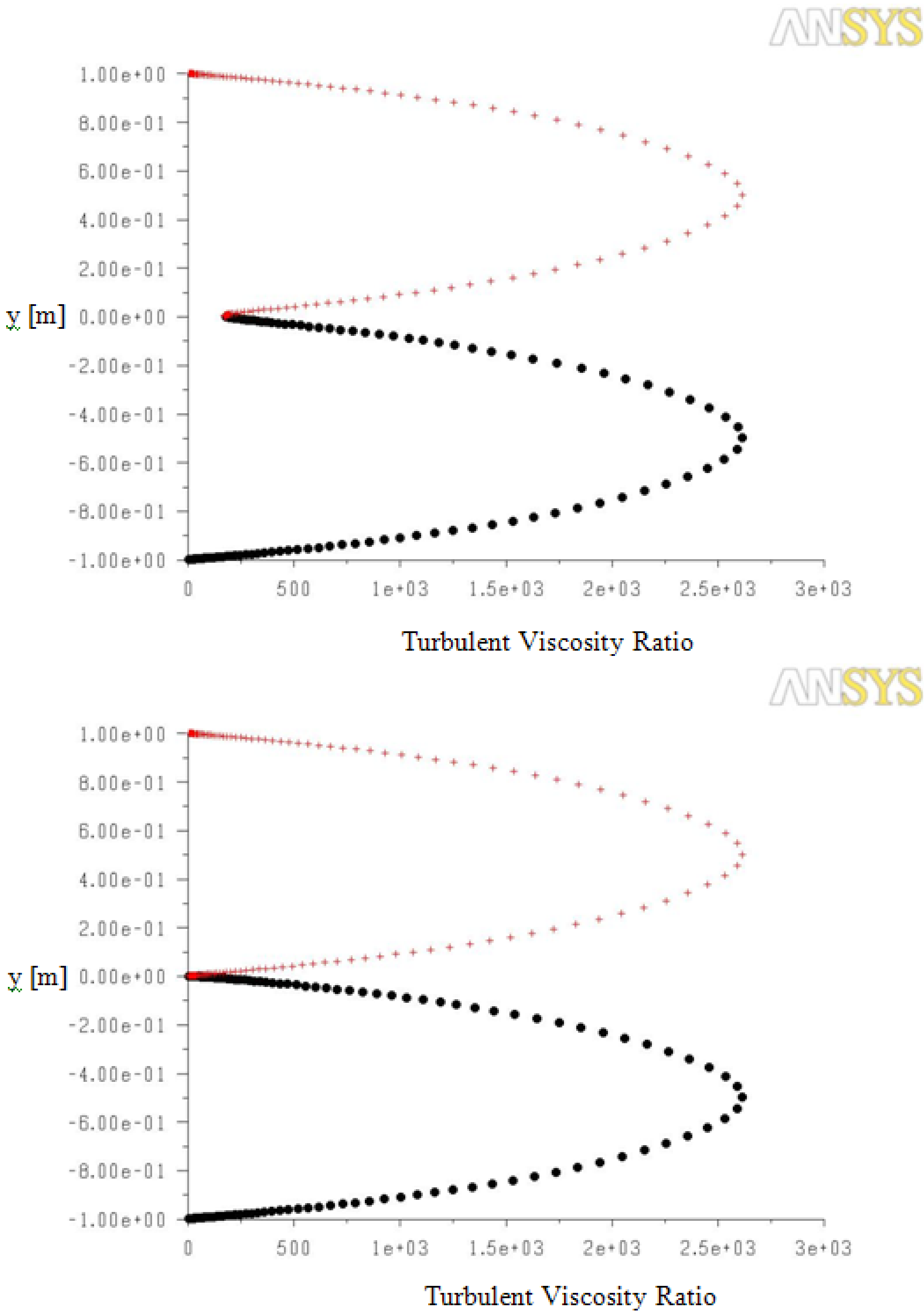


**Figure 9.5: Profiles of the Turbulent Viscosity Ratio Near the Outlet at Position x=0.9m Without (top) and With (bottom) Near-Wall Treatment at the Interface. Red Denotes the Pure Fluid Side, Black Represents the Side of the Porous Media**



## 9.4. Explicit Algebraic Reynolds Stress Model

Explicit Algebraic Reynolds Stress Models (EARSM) represent an extension of the standard two-equation models. They are derived from the Reynolds stress transport equations and give a nonlinear relation between the Reynolds stresses and the mean strain-rate and vorticity tensors. Due to the higher order terms, many flow phenomena are included in the model without the need to solve transport equations for individual Reynolds stresses. The WJ-BSL-EARSM allows an extension of the baseline (BSL) $k-\omega$ turbulence model to capture the following flow effects:

- Anisotropy of Reynolds stresses
- Secondary flows

The BSL model blends the robust and accurate formulation of the $k$-$\omega$ model in the near-wall region with the freestream independence of the $k$-$\varepsilon$ model in the far field. The BSL model was developed by Menter [3] (p. 37) and is described in the section on baseline k-omega model in the Fluent Theory Guide).

The implementation of the WJ-BSL-EARSM in Ansys Fluent is based on the explicit algebraic Reynolds stress model of Wallin and Johansson [4] (p. 37). Differences from the original formulation by Wallin and Johansson are explained in the following text.

With EARSM, the Reynolds stresses are computed from the anisotropy tensor according to its definition:

$$\overline{u_i u_j}=k\left(a_{ij}+\frac{2}{3}\delta_{ij}\right)$$

where the anisotropy tensor $a_{ij}$ is searched as a solution of the following implicit algebraic matrix equation:

$$N\mathbf{a}=-A_1\mathbf{S}+\left(\mathbf{a\Omega}-\mathbf{\Omega a}\right)-A_2\left(\mathbf{aS}-\mathbf{Sa}-\frac{2}{3}tr\left\{\mathbf{aS}\right\}\right),\text{ with}\quad N=A_3+A_4\left(P_k/\varepsilon\right) \tag{9.4}$$

The coefficients $A_i$ in this matrix equation depend on the $C_i$-coefficients of the pressure-strain term in the underlying Reynolds stress transport model. Their values are selected here as $A_1$=1.245, $A_2$=0, $A_3$=1.8, $A_4$=2.25.

The values of $A_2$, $A_3$, and $A_4$ are the same as those used in the original model by Wallin and Johansson [4] (p. 37). As for the value of $A_1$, it is increased from 1.2 to 1.245 in the course of calibrating EARSM for its use together with the BSL $k-\omega$ model.

$S_{ij}$ and $\Omega_{ij}$ denote the non-dimensional strain-rate and vorticity tensors, respectively. They are defined as:

$$S_{ij}=\frac{1}{2}\tau\left(\frac{\partial U_i}{\partial x_j}+\frac{\partial U_j}{\partial x_i}\right) \tag{9.5}$$

$$\Omega_{ij}=\frac{1}{2}\tau\left(\frac{\partial U_i}{\partial x_j}-\frac{\partial U_j}{\partial x_i}\right) \tag{9.6}$$

where the time-scale $\tau$ is given by:

$$\tau=k/\varepsilon=1/\left(C_\mu\omega\right),\quad C_\mu=0.09 \tag{9.7}$$

In order to arrive at an explicit solution of the Equation 9.4 (p. 33), the anisotropy tensor is expressed as a polynomial based on the strain rate and the vorticity tensors as follows:

$$a_{ij}=\beta_1 S_{ij}+\beta_3\left(\Omega_{ik}\Omega_{kj}-\tfrac{1}{3}II_\Omega\delta_{ij}\right)+\beta_4\left(S_{ik}\Omega_{kj}-\Omega_{ik}S_{kj}\right)$$
$$+\beta_6\left(S_{ik}\Omega_{kl}\Omega_{lj}+\Omega_{ik}\Omega_{kl}S_{lj}-\tfrac{2}{3}IV\delta_{ij}-II_\Omega S_{ij}\right) \tag{9.8}$$

The $\beta$-coefficients are evaluated to:

$$\beta_1=-N/Q$$

$$\beta_3=-12\cdot IV/\left(N\cdot Q\cdot\left(2N^2-II_\Omega\right)\right)$$

$$\beta_4=-1/Q$$

$$\beta_6=-6\cdot N/\left(Q\cdot\left(2N^2-II_\Omega\right)\right)$$

where the denominator Q is:

$$Q=\left(N^2-2II_\Omega\right)/A_1$$

The invariants, which appear in the formulation of the anisotropy tensor and the coefficients, are defined by:

$$II_S=S_{kl}S_{lk}$$

$$II_\Omega=\Omega_{kl}\Omega_{lk}$$

$$IV=S_{kl}\Omega_{lm}\Omega_{mk}$$

The model representation of the anisotropy tensor Equation 9.8 (p. 34) and its coefficients $\beta_i$ follows the original model by Wallin and Johansson [4] (p. 37) with two differences. First, the fourth order tensor polynomial contribution (the $\beta_9\left(\mathbf{\Omega S\Omega}^2-\mathbf{\Omega}^2\mathbf{S\Omega}\right)$ term) is neglected in Equation 9.8 (p. 34). Second, the tensor basis is slightly changed for convenience according to Apsley and Leschziner [1] (p. 37). Although the tensor basis is changed, the model remains algebraically equivalent to the original model of Wallin and Johansson. The latter change results in correspondingly changed expressions for the coefficients $\beta_i$.

In three-dimensional flows, the equation to be solved for the function $N$ is of sixth order and no explicit solution can be derived, whereas in two-dimensional mean flows the function $N$ can be derived from a cubic equation, an analytic solution of which is recommended by Wallin and Johansson [4] (p. 37) also for three-dimensional cases:

$$N=\begin{cases} A_3/3+\left(P_1+\sqrt{P_2}\right)^{1/3}+\mathrm{sign}\left(P_1-\sqrt{P_2}\right)\left|P_1-\sqrt{P_2}\right|^{1/3} & \text{for } P_2\geq 0 \\ A_3/3+2\left(P_1^2-P_2\right)^{1/6}\cos\left(\frac{1}{3}\arccos\left(\frac{P_1}{\sqrt{P_1^2-P_2}}\right)\right) & \text{for } P_2<0 \end{cases} \tag{9.9}$$

where

$$P_1=\left(\frac{A_3^2}{27}+\frac{A_1A_4}{6}II_S-\frac{2}{3}II_\Omega\right)\cdot A_3$$

$$P_2=P_1^2-\left(\frac{A_3^2}{9}+\frac{A_1A_4}{3}II_S+\frac{2}{3}II_\Omega\right)^3$$

In the original model by Wallin and Johansson [4] (p. 37), the diffusion terms in the transport equations for $k$ and $\omega$ were calculated using the effective eddy viscosity, $\mu_t^{eff}=\left(C_\mu^{eff}/C_\mu\right)\cdot k/\omega$, of EARSM, where $C_\mu^{eff}=-\beta_1/2$. The EARSM model, implemented in Ansys Fluent, uses the standard eddy viscosity $\mu_t=k/\omega$ for the diffusion terms. This model change helps to avoid the problems with the asymptotic behavior at the boundary layer edge, which were reported by Hellsten [2] (p. 37).

For the underlying BSL $k-\omega$ model, the standard coefficients are used.

### 9.4.1. Accessing the WJ-BSL-EARSM Model

The EARSM model is available after enabling beta feature access, as described in Introduction (p. 1).

---

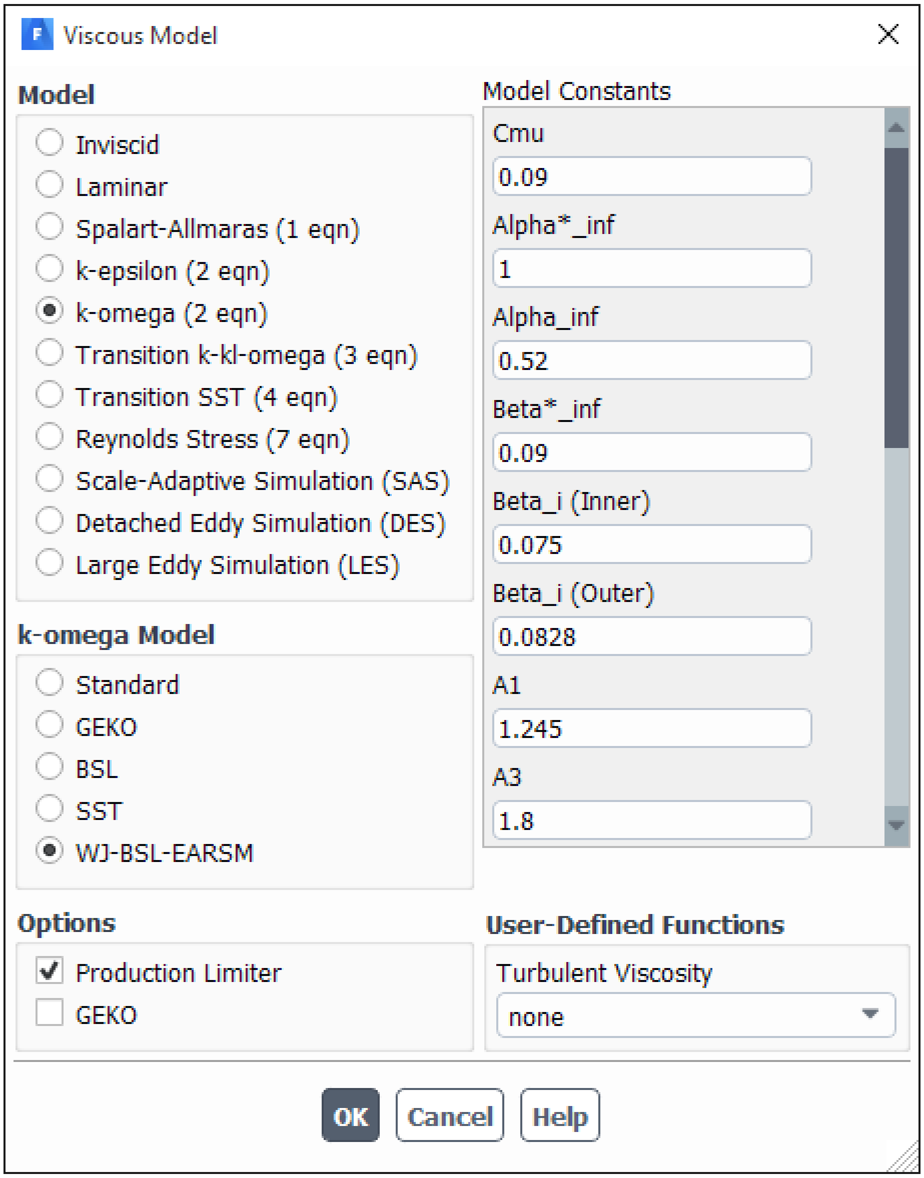
**Note:**

It is only available for 3D cases, and is not supported with the gap model.

---

The model is available in the interface when the **k-omega** model is selected. The **WJ-BSL-EARSM** option becomes available under **k-omega Model**, as shown in Figure 9.6: The Viscous Model Dialog Box (p. 36).

**Figure 9.6: The Viscous Model Dialog Box**




You can also enable the **WJ-BSL-EARSM** via the tui command: `define/models/viscous/kw-wj-bsl-earsm?`.

**Note:**

The GEKO can be enabled in combination with the WJ-BSL-EARSM by selecting **GEKO** under **Options**.

## 9.4.2. Applying Scale-Adaptive Simulation (SAS) with WJ-BSL-EARSM

Scale-Adaptive Simulation (SAS) can now be combined with **WJ-BSL-EARSM** in the same manner as other omega-based URANS turbulence models, as described in the section on setting up scale-adaptive simulation in the Fluent User's Guide.

For additional information about SAS, see the section on scale-adaptive simulation in the Fluent User's Guide.

### 9.4.3. WJ-BSL-EARSM Variables in Expressions

The following field variables are available for expressions:

| Description | Variable | Section | Type |
|---|---|---|---|
| Machine Learning WJ EARSM Beta1 | MachineLearningWJEARSMBeta1 | Turbulence... | Scalar |
| Machine Learning WJ EARSM Beta3 | MachineLearningWJEARSMBeta3 | Turbulence... | Scalar |
| Machine Learning WJ EARSM Beta4 | MachineLearningWJEARSMBeta4 | Turbulence... | Scalar |
| Machine Learning WJ EARSM Beta6 | MachineLearningWJEARSMBeta6 | Turbulence... | Scalar |

## 9.5. References

### Bibliography

[1] D.D. Apsley and M.A. Leschziner. "A new low-reynolds-number nonlinear two-equation turbulence model for complex flows". *International Journal of Heat and Fluid Flow*. 19. 209–222. 1998.

[2] A. Hellsten. "New advanced $k$−$\omega$ turbulence model for high-lift aerodynamics". *AIAA Paper 2004-1120*. Reno, Nevada. 2004.

[3] F.R. Menter. "Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications". *AIAA Journal*. 32(8). 1598–1605. August 1994.

[4] S. Wallin and A. Johansson. "A complete explicit algebraic Reynolds stress model for incompressible and compressible flows". *Journal of Fluid Mechanics*. 403. 89–132. 2000.

[5] Bell, J. H. & Mehta, R. D., (1990): "Development of a two-stream mixing layer from tripped and untripped boundary layers", AIAA J., vol. 28, no.12, pp. 2034-2042..

[6] Bradbury, L.J.S., (1965), "The Structure of a Self-Preserving Plane Jet", Journal of Fluid Mechanics Vol. 23, pp.31-64. .

[7] Driver, D.M., (1991): "Reynolds Shear Stress Measurements in a Separated Boundary Layer Flow", AIAA 22nd Fluid Dynamics, Plasma Dynamics and Laser Conference, AIAA-91-1787. .

[8] Launder B. E. and Sharma B. I.. (1974): "Application of the Energy-Dissipation Model of Turbulence to the Calculation of Flow near a Spinning Disc". Letters in Heat and Mass Transfer, 1:131–138..

[9] Vogel, J.C. & Eaton, J.K., (1985): "Combined Heat Transfer and Fluid Dynamic Measurements Downstream of a Backward-Facing Step", Vol. 107, Journal of Heat Transfer, pp. 922 – 929..

[10] Wygnanski, I. and Fiedler, H.E., (1968), "Some Measurements in the Self-Preserving Jet", Boeing Scientific Research Labs, Document D1-82-0712. .

[11] Michael L. Shur, Philippe R. Spalart, Michael K. Strelets, Andrey K. Travin; Synthetic Turbulence Generators for RANS-LES Interfaces in Zonal Simulations of Aerodynamic and Aeroacoustic Problems; Flow Turbulence Combust (2014) 93: 63. .

[12] Prandtl, L. (1952), Uber die ausgebildete Turbulenz, ZAMM, 5, 136–139..

[13] Spalart, P. R. "Strategies for Turbulence Modelling and Simulations," International Journal of Heat and Fluid Flow,Vol. 21, No. 3, June 2000, pp. 252–263..

# Chapter 10: Combustion

This chapter contains information relating to the combustion models implemented as beta features in Ansys Fluent 2021 R2.



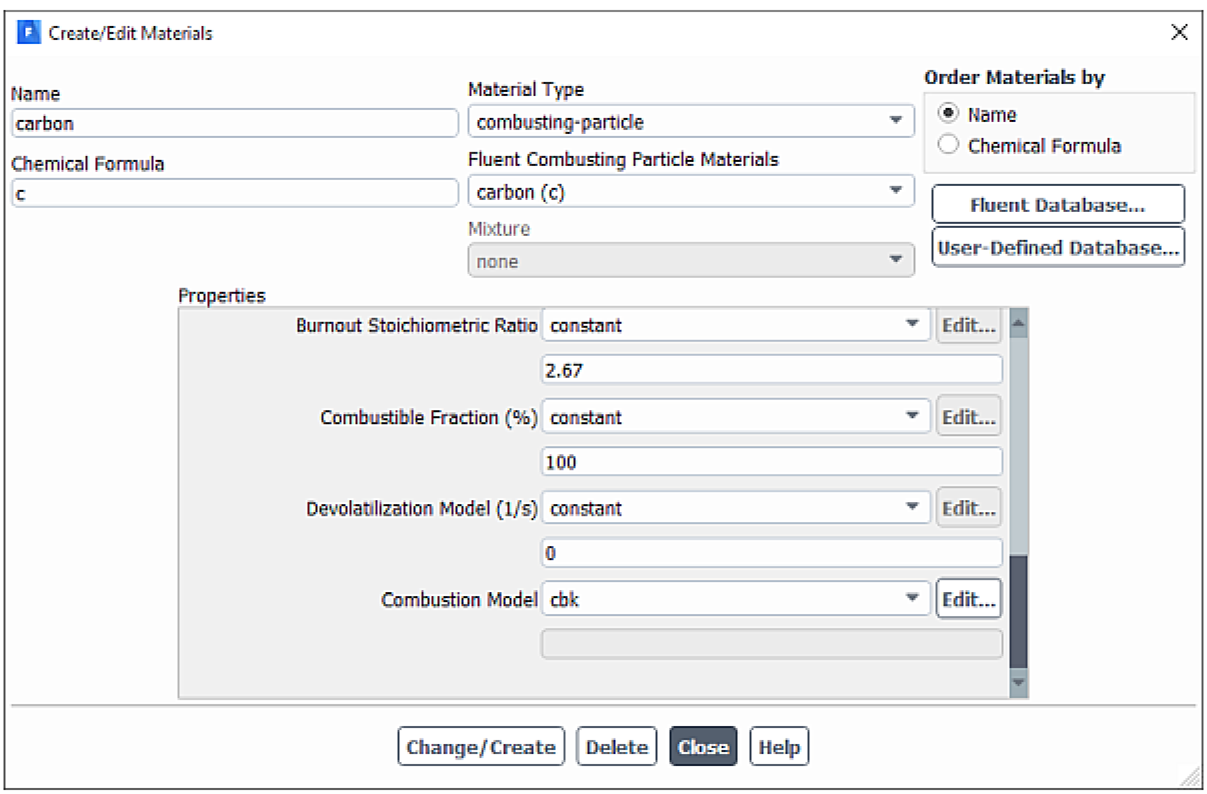
## 10.1. Char Burnout Kinetics (CBK) Model

The Char Burnout Kinetics (CBK) model describes char oxidation under conditions relevant to pulverized coal combustion processes. It includes effects of thermal annealing and ash inhibition on the char combustion. The form of this model that is available in Ansys Fluent only applies to atmospheric conditions and has no statistical kinetics.

The CBK model can be defined as a material property in the **Create/Edit Materials** dialog box (Figure 10.1: The Create/Edit Materials Dialog Box with the CBK Model Selected (p. 39)) for problems in which you have defined discrete-phase injections.

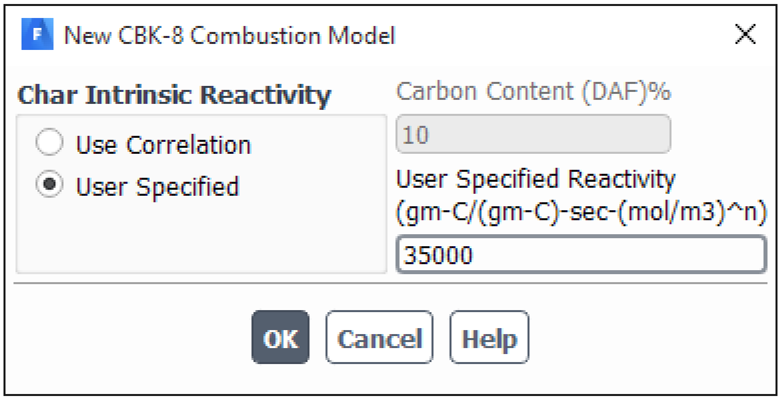
**Physics** → **Materials** → **Create/Edit...**

**Figure 10.1: The Create/Edit Materials Dialog Box with the CBK Model Selected**



Once you have selected **combusting-particle** from the **Material Type** drop-down list, you can select **cbk** from the **Combustion Model** drop-down list in the **Properties** group box. The **New CBK-8 Combustion Model** dialog will open, allowing you to set the char reactivity parameters (see Figure 10.2: The New CBK-8 Combustion Model Dialog Box (p. 40)). The default values are acceptable.

**Figure 10.2: The New CBK-8 Combustion Model Dialog Box**




### 10.1.1. References

1. FORTRAN program CBK8. Brown University, Providence, 1998.

2. R. Hurt, J. K. Sun, and M. Lunden. A Kinetic Model of Carbon Burnout in Pulverized Coal Combustion. Combustion and Flame, 113:181-197, 1998.

3. R. E. Mitchell, R. Hurt, L. L. Baxter, and D. R. Hardesty. Compilation of Sandia Coal Char Combustion Data and Kinetic Analyses: Milestone Report. Technical Report SAND92-8208, Sandia National Laboratory, Livermore, CA, 1992.

4. J. K. Sun and R. Hurt. Mechanics of Extinction and Near-Extinction in Pulverized Solid Fuel Combustion. Proceedings of the Combustion Institutes, 28:2205-2213, 2000.

# Chapter 11: Pollutants

The following models have been deprecated and are no longer supported in Ansys Fluent:

- The Coal Derived Soot model
- The Atomic Balance for Sulfur model
- The Mercury Pollutant Formation model.

# Chapter 12: Acoustics

This chapter contains information relating to acoustics models implemented as beta features in Ansys Fluent 2021 R2.



## 12.1. Modal Analysis

Ansys Fluent implements a finite-volume method for computing the resonance frequencies and natural acoustic modes for any enclosure. Sound wave propagation, reflection, diffraction, and convection are taken into account. The formulation requires an input of the mean flow obtained from a steady-state solution, together with prescribed boundary conditions. The method involves solving the Linearized Navier-Stokes Equations with the iterative Implicitly Restarted Arnoldi method to find the eigenvalues (frequencies) and eigenvectors (mode shapes of the pressure and velocity fluctuations).

### 12.1.1. Limitations

The following limitations apply to the modal analysis model currently implemented in Ansys Fluent:

- The modal analysis model is applicable to a steady-state, compressible ideal-gas solution.
- The modal analysis model is available for the double precision, serial Ansys Fluent version.
- The modal analysis model treats all domain boundaries as sound reflecting boundaries. Fluid particles can enter and leave the domain but acoustic waves are reflected.

### 12.1.2. Modal Analysis Theory

The system of 3D Linearized Navier-Stokes equations, in a Cartesian coordinate system are:

$$\frac{\partial \rho'}{\partial t} + \rho' \frac{\partial u_{j0}}{\partial x_j} + \rho_0 \frac{\partial u_j'}{\partial x_j} + u_j' \frac{\partial \rho_0}{\partial x_j} + u_{j0} \frac{\partial \rho'}{\partial x_j} = 0 \tag{12.1}$$

$$\begin{gathered} \frac{\partial u_i'}{\partial t} + u_{j0} \frac{\partial u_i'}{\partial x_j} + u_j' \frac{\partial u_{i0}}{\partial x_j} + \frac{1}{\rho_0} \frac{\partial p'}{\partial x_i} - \frac{\rho'}{\rho_0^2} \frac{\partial p_0}{\partial x_i} = \\ \frac{1}{\rho_0} \left( \frac{\partial \tau_{i1}'}{\partial x_1} + \frac{\partial \tau_{i2}'}{\partial x_2} + \frac{\partial \tau_{i3}'}{\partial x_3} \right) - \frac{\rho'}{\rho_0^2} \left( \frac{\partial \tau_{i1}^0}{\partial x_1} + \frac{\partial \tau_{i2}^0}{\partial x_2} + \frac{\partial \tau_{i3}^0}{\partial x_3} \right) \end{gathered} \tag{12.2}$$

$$\begin{gathered} \frac{\partial p'}{\partial t} + u_{j0} \frac{\partial p'}{\partial x_j} + u_j' \frac{\partial p_0}{\partial x_j} + \gamma p_0 \frac{\partial u_j'}{\partial x_j} + \gamma p' \frac{\partial u_{j0}}{\partial x_j} = \\ (\gamma - 1) \left[ \frac{\partial \left( u_{i0} \tau_{ij}' \right)}{\partial x_j} + \frac{\partial \left( u_i' \tau_{ij}^0 \right)}{\partial x_j} + \lambda \frac{\partial T'}{\partial x_j^2} - u_{io} \frac{\partial \tau_{ij}'}{\partial x_j} - u_i' \frac{\partial \tau_{ij}^0}{\partial x_j} \right] \end{gathered} \tag{12.3}$$

where $\tau_{ij}$ is the molecular stress tensor

$$\tau_{ij}=2\mu\left\{-\frac{\delta_{ij}}{3}\frac{\partial u_k}{\partial x_k}+\frac{1}{2}\left(\frac{\partial u_j}{\partial x_i}+\frac{\partial u_i}{\partial x_i}\right)\right\} \tag{12.4}$$

and the subscript 0 represents the mean values, and superscript $'$ represents the acoustic fluctuation about this mean (for example, $p=p_0+p'$). Note that a thermally perfect ideal gas is assumed. $\delta_{ij}$ is the Kronecker delta function, $\mu$ is the molecular viscosity, $\lambda$ is the conductive heat diffusivity coefficient, $c_v$ and $c_p$ are the specific heats at constant volume and constant pressure, respectively.

The mean flow conditions are first solved from a steady-state solution of the Reynolds-Averaged Navier-Stokes equations. The fluctuating quantities are assumed harmonic functions in time, so that $W'(t;x)=exp(-iwt)W'(x)$, where $W'=(p'u'v'w'T')$. Substituting this into the linearized Navier-Stokes gives

$$(i\omega)*\frac{\partial Q}{\partial W}*W'(x)+\frac{\partial Res(W_0)}{\partial W}*W'(x)=0 \tag{12.5}$$

The Jacobians $\frac{\partial Q}{\partial W}$ and $\frac{\partial Res(W_0)}{\partial W}$ are the direct linearizations of the discretized flow equations.

With $\gamma=-i\omega$, this system of equations can be written as:

$$[A]X=\lambda X \tag{12.6}$$

where $[A]=\frac{\partial Res(W_0)}{\partial W}$, $[A]=[B]^{-1}[A]$, $[B]=\frac{\partial Q}{\partial W}$ and $X=W'(x)$, $X$ is the solution of the eigenvalues problem. $W'$ is the eigenvector for the complex eigenvalue $\lambda=-i\omega$. The proper acoustic response of boundaries is included in the modal analysis model [1].

Equation 12.6 (p. 44) is an eigen-system and the iterative Implicitly Restarted Arnoldi method is use to compute a small set of eigenvalues and eigenvectors in a limited range of interest. The Implicitly Restarted Arnoldi method uses the ARPACK package (http://www.caam.rice.edu/software/ARPACK/), which is a collection of Fortran 77 subroutines designed to solve large-scale eigenvalue problems.

### 12.1.3. Using the Modal Analysis Model

Make sure you first enable beta feature access, as described in Introduction (p. 1). The procedure for computing the resonance frequencies and acoustic modes using the modal analysis model in Ansys Fluent is as follows:

1. Calculate a converged steady-state compressible RANS solution.

2. Enable the **Modal Analysis** acoustics model.

   **Setup** → **Models** → **Acoustics** **Edit...**

3. Set the associated **Model Constants** in the **Acoustics Model** dialog box.

4. Compute the resonance frequencies and acoustic modes by clicking the **Solve** button.

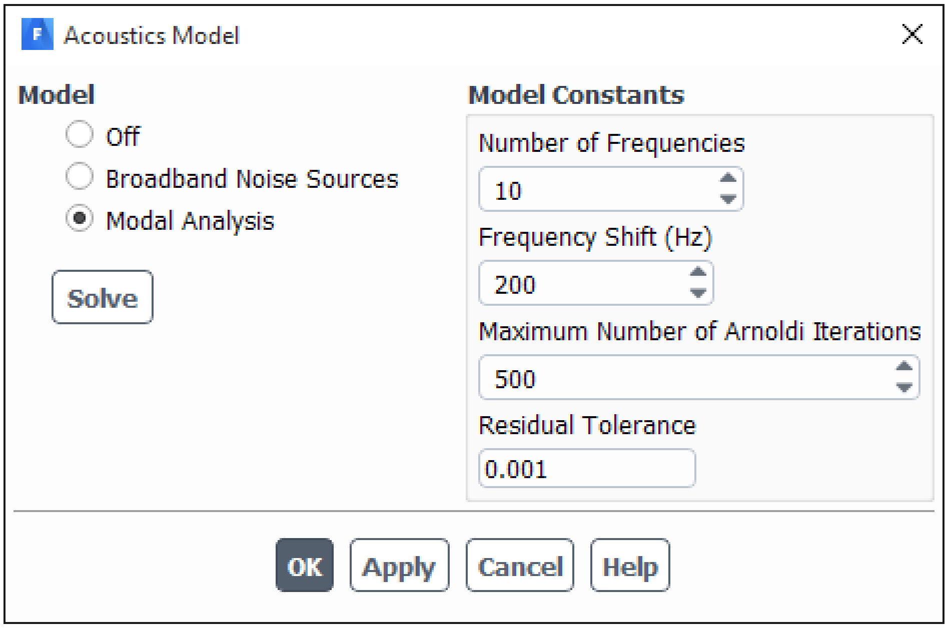
5. Postprocess the acoustic modes.

   **Postprocessing** → **Graphics** → **Contours** → **Edit...**

### 12.1.4. Setting Model Parameters

Under **Model Constants** in the **Acoustics Model** dialog box (Figure 12.1: The Acoustics Model Dialog Box (p. 45)), specify the relevant acoustic parameters used by the model.

**Figure 12.1: The Acoustics Model Dialog Box**




**Number of Frequencies**

is the requested maximum number of eigenvalues (natural acoustic frequencies). The default of 30.

**Frequency Shift**

is the frequency around which the eigenvalues will be solved. When the frequency shift is zero, the Arnoldi algorithm computes the eigenvalues around the smallest magnitude (SM). The default is 200Hz.

**Maximum Number of Arnoldi Iterations**

: the Implicitly Restarted Arnoldi Method is terminated after this many iterations if not converged. The default is 500.

**Residual Tolerance**

is the convergence criterion for the Arnoldi algorithm. The default is 0.001.

### 12.1.5. Postprocessing of the Modal Analysis Model

Postprocessing of the acoustic modes is accomplished by selecting the **Acoustics** category in the postprocessing dialog boxes:

- Acoustic Pressure Mode n

where n ranges from 1 to 10.

#### 12.1.5.1. References

1. Caraeni M., Devaki R. K., Aroni M., Oswald M., Srikanth KVSS, Efficient Acoustic Modal Analysis for Industrial CFD, AIAA-2009-1332, 2009

# Chapter 13: Discrete Phase

This chapter contains information relating to the pollution models implemented as beta features in Ansys Fluent 2021 R2.



## 13.1. Extended Collision Stencil

The extended collision stencil uses particles in neighboring cells to calculate the collision probability. The current model only uses particles in the current cell to calculate the collision frequency, where in this case particles in the closest N cells to the current particle location (where N is set by the user) are also allowed to collide.

The model should be used when the mesh is very fine and the spray is represented by relatively few particles per cell. Using a larger sample volume improves the statistics of the collision calculation. Note that the cost for calculating collision depends on the square of n, where n is the number of particles in the sampled volume, so increasing the number of cells beyond 3 or 4 can increase the computation time significantly.

To use this model (after enabling beta feature access, described in Introduction (p. 1) ), enter the following text command:

`define → models → dpm → spray-model → droplet-collision?` and enter the following responses to the commands:

```
Spray collision model [yes]
Spray collision model type [0-O'Rourke, 1-Stencil, 2-NTC]  [0] 1
Spray collision stencil size  [1]
 Spray collision event type [0-All, 1-Eff.Diam, 2-Impinging]  [0]
```

## 13.2. Volume Injections

When beta features and unsteady particle tracking are enabled, then for volume injections, you can enable the **Use Volume Fraction Instead of Flow Rate** option and specify **Volume Fraction** (rather than **Total Flow Rate** or **Total Mass**) in the **Set Injection Properties** dialog box (**Point Properties** tab). During simulation, Ansys Fluent will introduce parcels into the specified zones or bounding shape until either the specified **Volume Fraction** or **Injection Limit per Cell** is reached.

In the context of DEM, the initial particle positions are adjusted to ensure that no particle overlap exists in order to prevent un-physical behavior. You can specify either the total flow rate or the total mass of particles injected and you can specify either the total number of parcels or the number of parcels per CFD cell. Note that for injections using DEM, the initial volume fraction is limited to 30%-40% depending on size distribution. Standard DPM injections do not have an initial volume fraction limitation.

**Tip:**

If you are performing a DEM simulation and require an initial volume fraction higher than the 30%-40% guideline (for example, to initialize a packed bed), you can specify a larger volume for the initial seeding and run a preliminary calculation to allow the particles to settle.

## 13.3. Discrete Element Method with van der Waals Forces

After enabling beta features as described in Introduction (p. 1), you can account for van der Waals forces when using the discrete element method (DEM). Including van der Waals forces in DEM simulations is most important for particle diameters smaller than 40 to 100 micron.

Based on the studies of London [3] (p. 50) and Hamaker [1] (p. 49), the van der Waals interaction potential for two spherical particles, $U_{vdw}$, can be calculated by [2] (p. 49):

$$U_{vdw} = -\frac{H}{6}\left[\frac{2r_1r_2}{h^2+2h(r_1+r_2)} + \frac{2r_1r_2}{h^2+2h(r_1+r_2)+4r_1r_2} + \ln\left(\frac{h^2+2h(r_1+r_2)}{h^2+2h(r_1+r_2)+4r_1r_2}\right)\right] \tag{13.1}$$

where:

$H$ = Hamaker constant.

$h$ = surface-to-surface distance between the two interacting particles.

$r_1$ and $r_2$ = radii of the two interacting particles. For particle-wall interactions, $r_2$ is considered to be infinity.

$H$ can be calculated by:

$$H=\sqrt{H_1H_2}$$

where $H_1$ and $H_2$ are the Hamaker constants of the materials of the two particles, respectively.

The van der Waals force acting on a particle located at $\vec{P_1}$ by another particle located at $\vec{P_2}$ is calculated by:

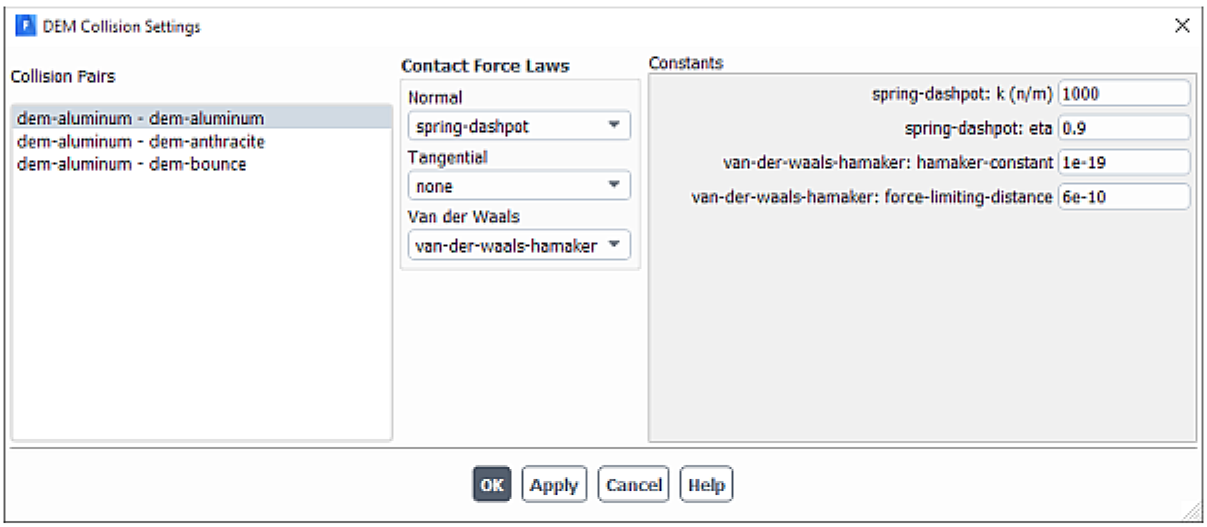
$$\vec{F}_{vdw} = -\nabla U_{vdw} \tag{13.2}$$

This is a force of attraction with the magnitude of the partial derivative of $U_{vdw}$ with respect to $h$ that points from the center of the first particle to the center of the second particle, $\left(\vec{P_2} - \vec{P_1}\right) / \|\vec{P_2} - \vec{P_1}\|$. For particle-wall interaction, the van der Waals force points in the direction of the shortest distance between the particle and the wall.

To avoid a singularity problem in Equation 13.2 (p. 49) when the particle surfaces are in contact (that is, $h$=0), the van der Waals force of attraction is clipped, that is it stops increasing, as the particle separation drops below a user-specified value, `force-limiting-distance`.

To model van der Waals forces during DEM particle-particle or particle-wall interactions:

1. In the **DEM Collision Settings** dialog box (that opens by clicking **Set...** in the **DEM Collisions** dialog box), from the **Van Der Waals** drop-down list, select `van-der-waals-hamaker`.




2. Specify the following force law parameters:

   - `hamaker-constant`: $H$ in Equation 13.1 (p. 48). Default value: 1e-19 J.
   - `force-limiting-distance`: is used to define an upper limit for the van der Waals force. Default value: 6e-10 m.

## Bibliography

[1] H. C. Hamaker. "The London – van der Waals attraction between spherical particles". *In: Physica IV*. 10. 1058–1072. 1937.

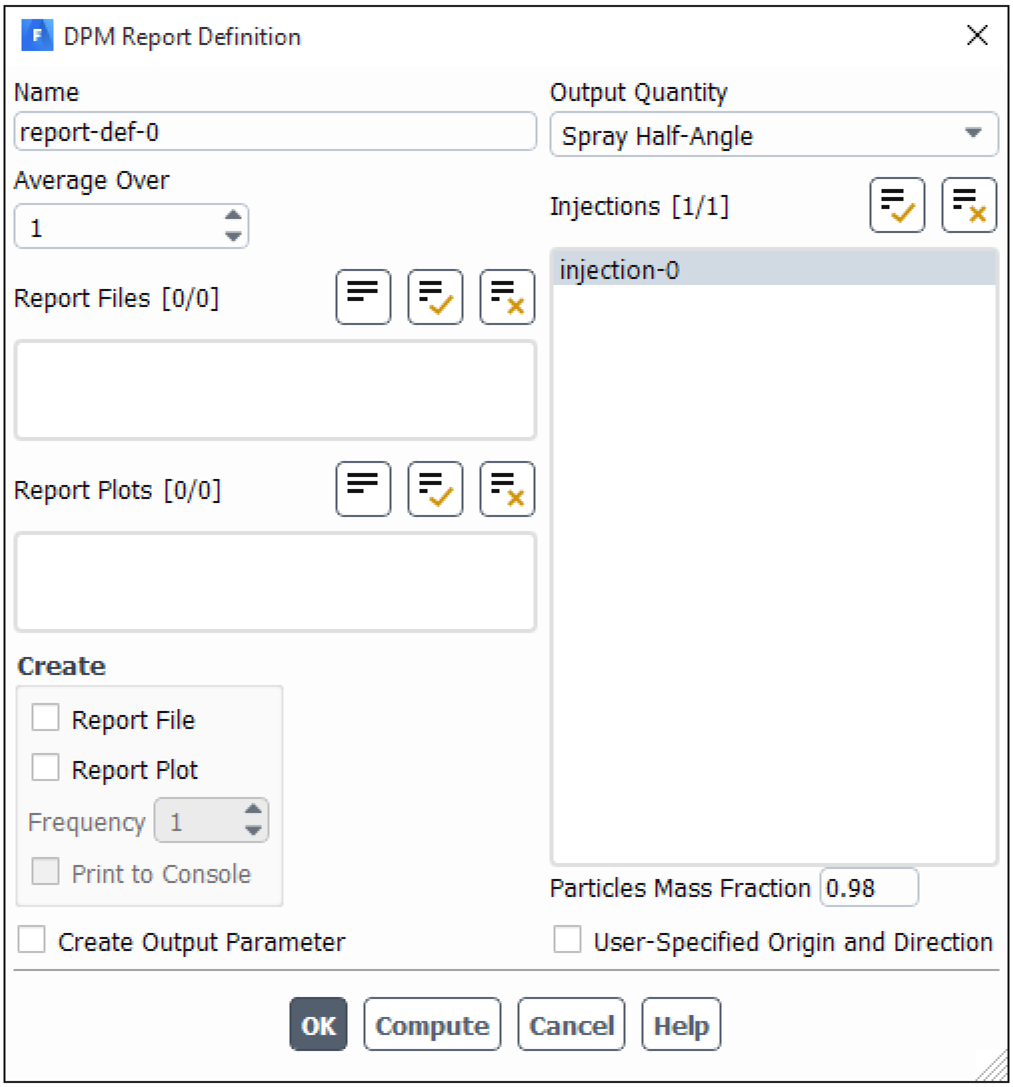
[2] C.-J. Lin, W.-C. J. Wei, T. Iwai, C.-W. Hong, and P. Greil. "Discrete Element Method (DEM) Simulation and Processing of Mo/AL2O3 Granules in Fluidizing Bed". *Proc. Nat. Sci. Counc. RAOC(A)*. 24. 5. 394-404. 2000.

[3] F. London. "The general theory of molecular forces". *Trans. Faraday Soc.*. 33. 8-26. 1937.

## 13.4. DPM Report—Spray Half-Angle

With beta features enabled (as described in Introduction (p. 1)), injections defined, and **Unsteady Particle Tracking** enabled in the **Discrete Phase Model** dialog box, you can create **DPM Report Definitions** with **Spray Half-Angle** as the **Output Type**.

**Solution** → **Reports** → **Definitions** → **New** → **DPM Report** → **Spray Half-Angle...**




Spray Half-Angle reports are similar to the Spray Penetration Length reports except that the spray half-angle is calculated as the opening half-angle (between the axis and surface of a cone) that contains the specified mass fraction of all the particles belonging to the selected injection(s).

To setup the DPM Report Definition dialog box:

1. Enter a name for the report definition under **Name**.

2. (optional) To have Fluent calculate a running average for the **DPM Report Definition** you can enter a positive integer greater than 1 (the default) for **Average Over**.

   *Specifying a number greater than 1 means that Fluent will print, plot, and write the running average value of the selected variable instead of the current value of the same variable.*

   *The value reported is averaged over the last N iterations/time steps, where N is your specified **Average Over** value. When the iteration number is lower than N, Fluent calculates the average of the available variable values.*

3. If you want the report definition written to a file or plotted, either select an existing report file or plot from the **Report Files** or **Report Plots** lists, or create a new report file or plot by enabling **Report File** and/or **Report Plot** in the **Create** group box.

4. If you want the report definition available as an output parameter, enable **Create Output Parameter**.

5. Select the injection(s) that will be used for calculating the spray half-angle from the **Injections** list.

   - Single injection selected: Fluent uses the specified injection location or user-specified coordinates.
   - Multiple injections selected: Fluent averages the coordinates of the injections. If Fluent cannot average the coordinates a message will be printed in the console—in this case you should manually specify the injection origin and axis coordinates.

6. Specify the **Particles Mass Fraction**.

   *This is the injection/particle mass fraction expected within the cone specified by the origin, direction, and the half-angle calculated by this report.*

7. (optional) Enable **User-Specified Origin and Direction** to manually specify the origin and direction for the selected injection(s).

8. Click **OK** to create the DPM Report Definition.

## 13.5. Particle Tracking Within the Eulerian Multiphase Framework

When using the DPM in conjunction with the Eulerian multiphase model, you can track particles in an arbitrary Eulerian phase or in the mixture of all phases. The particles are one-way coupled with the continuous phase(s), so that only forces from the fluid phase are considered when computing the particle equation of motion, while the source terms arising from the dispersed phase are not considered in the fluid flow equation.

When particles are tracked in a single Eulerian phase, only forces from that phase are taken into account. In case of flow separation, the particles are not accelerated in regions within which the phase of interaction is not present. Body forces, such as gravitation, do not depend on the phase of interaction.

When particles are tracked in the mixture of all phases, the force contribution from each phase to the particle acceleration is weighted by the volume fraction of that phase.

### 13.5.1. Limitations

- The feature is available only for **Massless** and **Inert** particle types.

- For particle tracking in all phases, you cannot specify the drag law per injection-phase pair. The drag law you select for the injection will be applied to all injection-phase interactions.
- Within the DDPM framework, injections assigned to a DPM phase can interact only with the primary phase.
- Particle turbulent dispersion is calculated from turbulent velocity fluctuations of the primary phase only. If the phase of interaction is not set to either a primary phase or all-phases, turbulent dispersion of particles is not accounted for.
- Wall film particles interact only with the primary phase.
- The DPM cloud model is not compatible with this feature.
- The following forces are not supported:
  - Forces arising in rotating frames of reference
  - Thermophoretic force
  - Brownian motion
- Particle rotation is supported only for particle tracking in a single Eulerian phase.

---

**Important:**

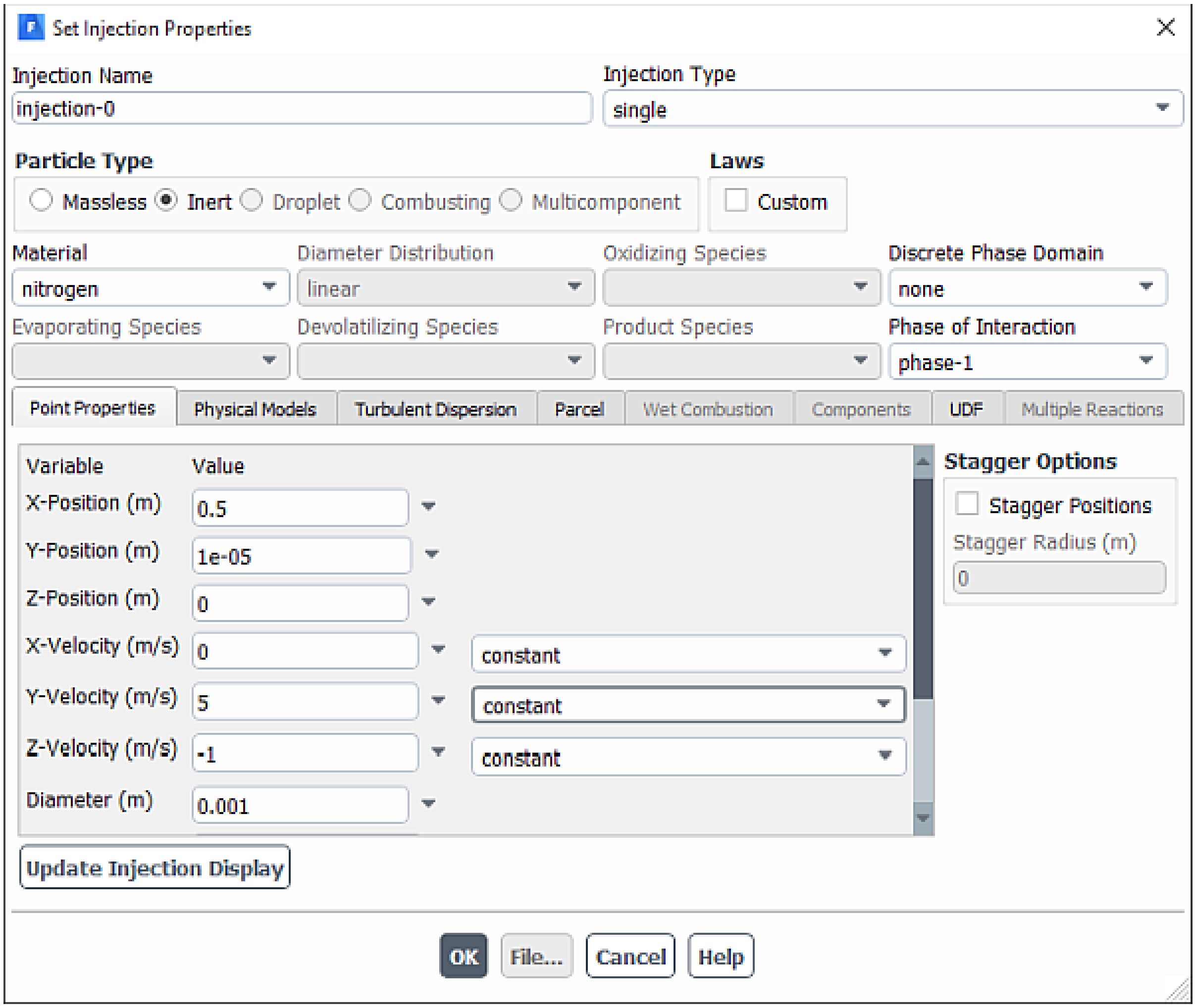
The particle integration using the **trapezoidal** numerical scheme is under development and is not yet suitable for use. Instead use the Eulerian **implicit** scheme.

---

## 13.5.2. Using Particle Tracking with the Eulerian Multiphase Model

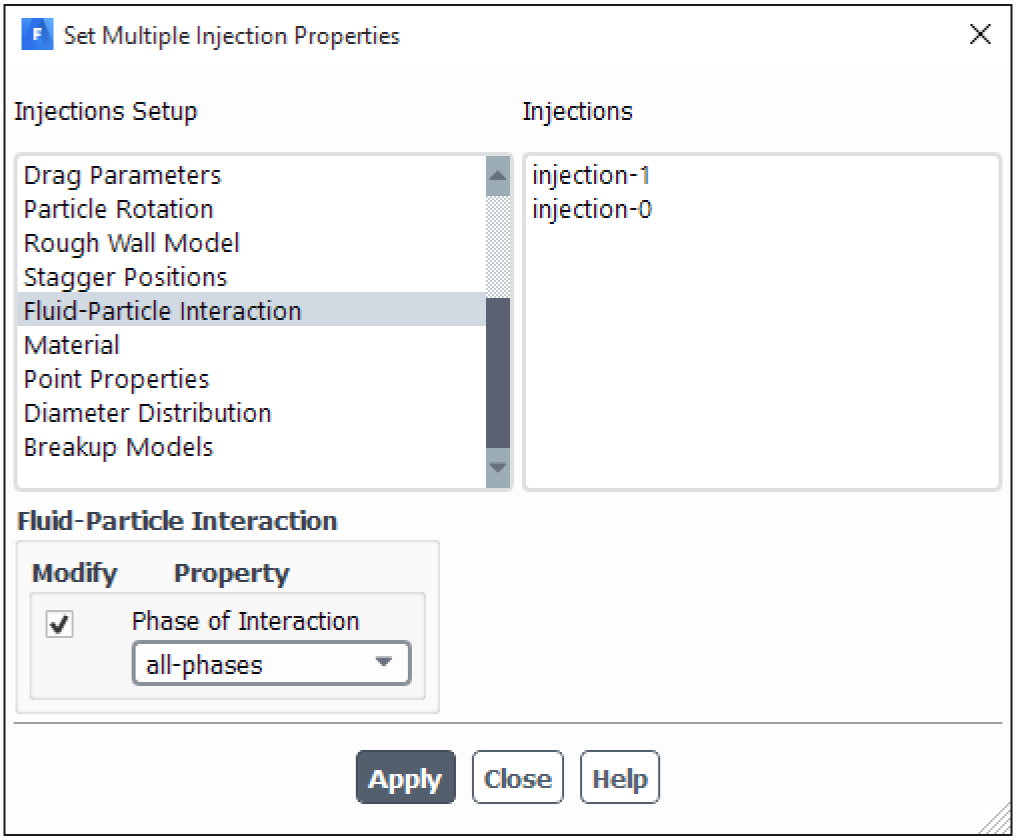
1. Enable the beta feature access (as described in Introduction (p. 1)).

   In the **Set Injection Properties** dialog box, the **Phase of Interaction** drop-down-list appears below the **Discrete Phase Domain** drop-down list.




2. From the **Phase of Interaction** drop-down-list, select the phase in which particles will be tracked. If you want to track particles in all phases, select **all-phases**. The default phase of interaction is the primary phase.

3. If you want to change the phase of interaction for multiple injections, select **Fluid-Particle Interaction** from the **Injection Setup** selection list in the **Set Multiple Injection Properties** dialog box, and then select the appropriate phase from the **Phase of Interaction** drop-down list.




You can also set the phase of interaction in a multiphase flow using the following text command:

```
define/injections/properties/set/interaction/fluid-particle-interac-
tion/phase-of-interaction
```

Note that this command is available only when you select the injection(s) using `define/injections/properties/set/pick-injections-to-set`.

## 13.6. Using the Non-Iterative Time-Advancement (NITA) Solver with the DPM model

After enabling the beta feature access (Introduction (p. 1)), you can use the non-iterative time-advancement (NITA) scheme for your DPM transient flow calculations in order to increase the speed and efficiency of your simulation. The NITA solver is recommended for problems that use small time steps (the Courant number is below unity everywhere in the domain). For background information about the non-iterative time-advancement scheme, see the Fluent Theory Guide.

Details about using the NITA option are described in the section on setting solution controls for the non-iterative solver in the Fluent User's Guide. When using NITA, you should keep the time steps sufficiently small so that the Courant number remains below unity everywhere.

Note that when the **Non-Iterative Time-Advancement** option is selected in the **Solution Methods** task page, the **Update DPM Sources Every Flow** Iteration option becomes unavailable in the **Discrete Phase Model** dialog box.

## 13.7. Force transferred to System Coupling from a Wall Boundary

For simulations that involve the DPM or DEM models, when **Floating Operating Pressure** is enabled, the pressure force that Ansys Fluent returns to System Coupling will include the particle contribution due to particle-wall interactions:

$$F = (p + p_{raise}) - p_{ref} + p_{particles} \tag{13.3}$$

where $p_{particles}$ is the DPM wall normal pressure due to particles interacting with the wall. For more information, see section Force transferred to System Coupling from a Wall Boundary in the Ansys Fluent User's Guide.

## 13.8. Using High-Resolution Tracking with Ansys Fluent Models

When beta features access is enabled:

- You can use the high resolution method with the temporal interpolation of flow solution variables to the DPM particle position
- You can use the `define/models/dpm/numerics/high-resolution-tracking/enable-transient-variable-interpolation?` text command if you want the flow variables to be interpolated both temporally and spatially to the current particle position and time as the particle trajectory is integrated from the previous flow solver timestep to the current flow time. This text command option is available only for transient cases with unsteady particle tracking. This option is not compatible with the interpolation of flow gradients (the `define/models/dpm/numerics/high-resolution-tracking/interpolate-flow-solution-gradients` text command option). The interpolation of flow gradients should be disabled prior to activating transient variable interpolation. If both options are enabled, Ansys Fluent will automatically disable transient variable interpolation.

See section Tracking Parameters and Options for the Discrete Phase Model in the Ansys Fluent User's Guide for information about the high-resolution tracking method.

## 13.9. Blocking Effect

### 13.9.1. Theory

In standard Lagrangian-Eulerian predictions, particles are assumed to be mass points and, therefore, do not displace the carrier phase. The volume fraction of the carrier phase is assumed to be unity throughout the entire domain, regardless of the actual particle concentration. Physically, the assumption of particles sharing the same physical space with the carrier phases only holds true if the volume fraction of the Lagrangian phase is negligible. In many technically relevant simulations, the volume fraction of the local particle phase may not be small, and the blocking effect of the particulate phase on the carrier phase may need to be taken into account.

The blocking effect impact requires the flow around the particles to be incorporated into transport equations. Any transported variable $\varphi$ belonging to a carrier phase is represented by:

$$\frac{\delta}{\delta t}(\rho\varphi) + \nabla(\rho U \varphi) = \nabla(\rho \Gamma \nabla \varphi) + S \tag{13.4}$$

where

$\rho$ = carrier phase density

$\Gamma$ = kinematic combined (laminar + turbulent) diffusivity

$U$ = velocity of the carrier phase

$S$ = optional source per unit volume

The blocking effect changes the transport equation to:

$$\frac{\delta}{\delta t}\left(\rho\alpha_f\varphi\right)+\ \nabla\left(\rho\alpha_f U\varphi\right)=\ \nabla.\left(\alpha_f\rho\Gamma\nabla\varphi\right)+\alpha_f S \tag{13.5}$$

where $\alpha_f$ is the carrier phase volume fraction.

The carrier phase volume fraction $\alpha_f$ is computed from the particle phase volume fraction $\alpha_p$ as:

$$\alpha_f=1-\min\left(\alpha_p,\ \alpha_{p,max}\right) \tag{13.6}$$

where $\alpha_{p,max}$ is a user-specified value. This value is clipped between 0 and 1.

The DPM blocking model modifies the single-phase equations to incorporate the effect of blocking due to the appearance of the carrier phase volume fraction in the transient, convection, diffusion, and source terms of the corresponding transport equation. Note that particle source terms in the carrier phase are still computed using the "mass-point" assumption, that is, no scaling of the particle source terms is performed based on the volume fraction of the carrier phase.

## 13.9.2. Using the Blocking Effect

Once beta features are enabled, (Introduction (p. 1)), you can access the blocking effect option via the Text User Interface (TUI).

To enable the blocking effect of particles on the carrier phase, use the following text command:

```
define/models/dpm/interaction> enable-flow-blocking-by-particles?
Enable inclusion of DPM volume fraction in continuous flow? [no] yes
```

Once the blocking effect is enabled, you can set the maximum particle volume fraction $\alpha_{p,max}$ used to compute the carrier phase volume fraction $\alpha_f$ in Equation 13.6 (p. 56) by issuing the following text command:

```
define/models/dpm/interaction> max-vf-allowed-for-blocking
Maximum DPM volume fraction used in continuous flow (0 1.) [0.95]
```

For high accuracy solution, using the default value of 0.95 is recommended.

This feature does not require **Interaction with Continuous Phase** to be enabled. That is, the blocking effect can also be used in simulations solving one-way coupled particles. The availability of the DPM volume fraction is automatically triggered by the Ansys Fluent solver.

The blocking effect model has the following limitations:

- It is limited to single-phase (one carrier phase) flows.
- The density-based solver is not supported.

The impact of the blocking effect will be strongest in simulations where large particle volume fractions are present in the computational domain, or where the blockage due to the particulate phase may significantly alter the flow structure of the carrier phase. Typical examples are simulations of a particle spray, where the mesh close to the injection nozzle is refined in such a way that the injection nozzle cross-section is discretized by multiple cells, or jet in cross-flow applications, where the blockage effect of the compact liquid jet may enhance the vortex structure in the wake of the jet.

## 13.10. Stochastic Kuhnke Model

With beta features enabled (as described in Introduction (p. 1)), the Stochastic Kuhnke Model becomes available as an additional option for the Stanton-Rutland, Kuhnke, and User-Defined models.

### 13.10.1. Theory

The Stochastic Kuhnke model [1] (p. 61), [2] (p. 61) is derived from the Kuhnke model and has the following features:

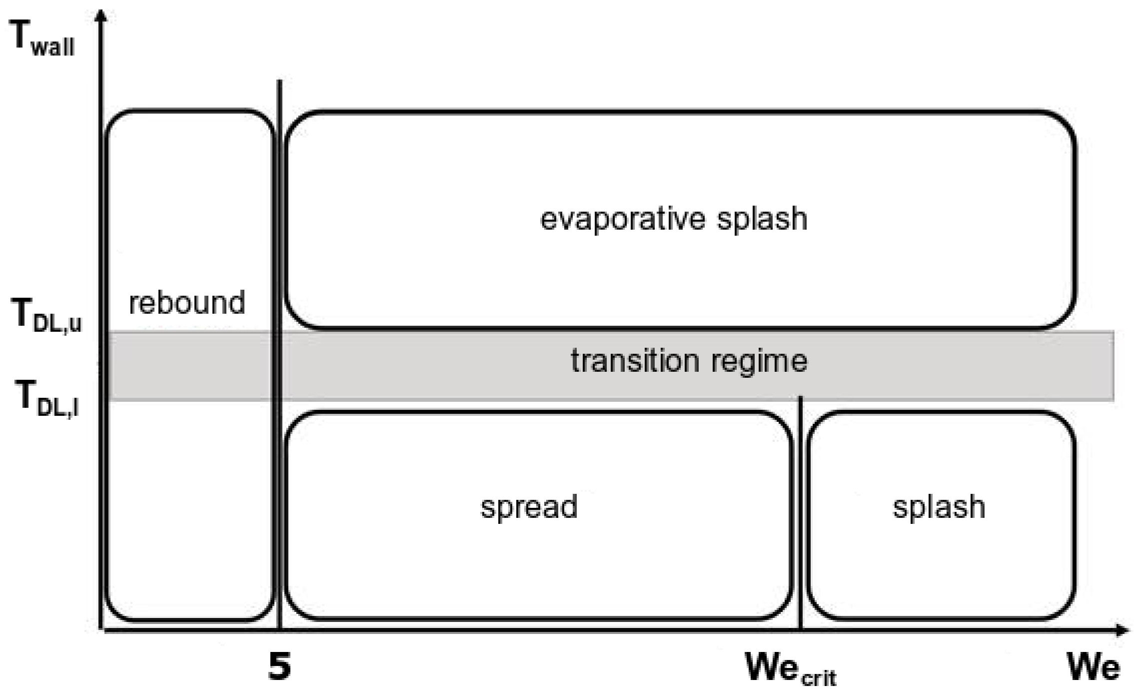
- Uses different regime transition criteria that have been tuned for SCR modeling applications
- Includes stochastic effects into the critical temperature transition process
- Introduces the evaporative splash regime with the "partial evaporation" concept, where a fraction of the impinging droplets is assumed to completely vaporize under certain conditions

The model is compatible with both Lagrangian and Eulerian Wall Film models.

#### Regime Definition

The regime map of the Stochastic Kuhnke model is shown in Figure 13.1: The Regime Map of the Stochastic Kuhnke Model (p. 58).

**Figure 13.1: The Regime Map of the Stochastic Kuhnke Model**




The model considers all relevant impingement phenomena by classifying them into the following four core regimes based on critical temperature and Weber number transition criteria:

- rebound
- spread
- splash
- evaporative splash

The spread and splash regimes lead to formation of a wall film. In addition, the transition regime is introduced to avoid abrupt changes during the film deposition process.

The regime transition criteria are defined as follows. A critical temperature range is defined between the upper and lower deposition limits $T_{DL,u}$ and $T_{DL,l}$:

$$T_{DL,u} = T_{sat} + \Delta T_{DL} \tag{13.7}$$

$$T_{DL,l} = T_{DL,u} - \Delta T_{lim} \tag{13.8}$$

where

$T_{sat}$ = liquid film saturation temperature [K]

$\Delta T_{DL}$ = upper deposition limit offset [K]

$\Delta T_{lim}$ = deposition range temperature difference [K]

The Weber number criterion is defined differently for wet and dry walls:

- Critical Weber number for splashing on a wet wall:

$$We_{crit,w} = A_w La^{-0.18} \tag{13.9}$$

- Critical Weber number for splashing on a dry wall:

$$We_{crit,D} = A \cdot La^{-0.18} \tag{13.10}$$

In Equation 13.9 (p. 58) and Equation 13.10 (p. 59):

$La$ = Laplace number

$A_w$ = Laplace number constant

$A$ = function of wall roughness $r_z$ interpolated from Table 13.1: Parameter A as a Function of Wall Roughness (p. 59)

**Table 13.1: Parameter *A* as a Function of Wall Roughness**

| $r_z$ (microns) | $A$ |
|---|---|
| 0.05 | 5264 |
| 0.14 | 4534 |
| 0.84 | 2634 |
| 3.1 | 2056 |
| 12 | 1322 |

The Laplace number is expressed as:

$$La = \rho\sigma d / \mu^2 \tag{13.11}$$

where

$\rho$ = droplet density

$\sigma$ = droplet surface tension

$d$ = droplet diameter

$\mu$ = droplet viscosity

Finally, the Stochastic Kuhnke model assigns the appropriate regime for each impingement event according to the droplet Weber number (see the definition of the Weber number in section "The Kuhnke Model" in the Ansys Fluent Theory Guide) and the wall conditions. These regimes are:

- Rebound

  Regime boundaries: $We < 5$ and $\begin{cases} T_{wall} < T_{DL,l} \text{ and Wet wall} \\ or \\ T_{wall} > T_{DL,l} \end{cases}$

  The impinging drop is emitted without change in size. The velocity components of the rebounding particle are calculated in accordance with section "Rebound" for the Kuhnke model in the Ansys Fluent Theory Guide.

- Spread

  Regime boundaries: $T_{wall} < T_{DL,l}$ and $\begin{cases} 5 < \mathrm{We} < \mathrm{We}_{SW} \text{ and Wet wall} \\ or \\ \mathrm{We} < \mathrm{We}_{SD} \text{ and Dry wall} \end{cases}$

The impinging drop is added to the wall film.

- Splash

  Regime boundaries: $T_{wall} < T_{DL,l}$ and $\begin{cases} \mathrm{We} > \mathrm{We}_{SW} \text{ and Wet wall} \\ or \\ \mathrm{We} > \mathrm{We}_{SD} \text{ and Dry wall} \end{cases}$

  Part of the impinging droplet mass is deposited to form a film, and the rest of the droplet is atomized. The splashed mass fraction and the splashed droplet variables are computed according to section "Splashing" for the Kuhnke model in the Ansys Fluent Theory Guide.

- Evaporative Splash

  Regime boundaries: $T_{wall} > T_{DL,u}$ and $We > 5$

  A small fraction of the impinging liquid spray is assumed to vaporize immediately upon impact, while the rest is completely atomized emitting smaller sized droplets. The vaporization events occurring in this regime are controlled by random number sampling. Similarly to the splash regime, the splashed droplet variables are computed as described in section "Splashing" for the Kuhnke model in the Ansys Fluent Theory Guide.

- Transition Regime

  Regime boundaries: $T_{DL,l} < T_{wall} < T_{DL,u}$ and $We > 5$

  Under these conditions, any of the regimes (spread, splash, or evaporative splash) may be assigned to the impinging drop. This is decided based on comparison of a random number with the deposition ratio $dr$ computed from Equation 13.12 (p. 60) in combination with the droplet Weber number and the wet or dry wall criterion.

$$dr = \frac{T_{wall} - T_{DL,l}}{T_{DL,u} - T_{DL,l}} \tag{13.12}$$

## 13.10.2. Using the Stochastic Kuhnke Model

To use the stochastic Kuhnke impingement/splashing model, follow these steps:

1. Enable beta feature access as described in Introduction (p. 1).
2. In the **Wall** dialog box, **DPM** tab, select **wall-film** for **Boundary Cond. Type** (**Discrete Phase Model Conditions** group).
3. From the **Impingement/Splashing Model** drop-down list, select the **stochastic kuhnke** option.
4. Specify the **Number of Splashed Drops**.
5. In the **Critical Temperature** group box, specify the following parameters:
   - **Upper Deposition Limit Offset**: is $\Delta T_{DL}$ in Equation 13.7 (p. 58)
   - **Deposition Delta T**: is $\Delta T_{lim}$ in Equation 13.8 (p. 58)

- **Laplace Number Constant**: is $A_w$ in Equation 13.9 (p. 58)
- **Partial Evaporation Ratio**: is the mass fraction of the impinging liquid spray that vaporizes immediately upon impact when the droplet is in the evaporative splash regime

6. Specify the **DPM Wall Roughness Parameter** $r_z$. This parameter will be used to interpolate the constant $A$ from Table 13.1: Parameter $A$ as a Function of Wall Roughness (p. 59), which will be subsequently used in Equation 13.10 (p. 59) to calculate critical Weber number for splashing on a dry wall.

## Bibliography

[1] H. Smith, M. Zöchbauer, and T. Lauer. "Advanced Spray Impingement Modelling for an Improved Prediction Accuracy of the Ammonia Homogenisation in SCR Systems". *"SAE Technical Paper 2015-01-1054"*. doi:10.4271/2015-01-1054. 2015.

[2] C. Bai and A. Gosman. "Development of Methodology for Spray Impingement Simulation". *"SAE Technical Paper 950283"*. 1995.

# Chapter 14: Multiphase Modeling

This chapter contains information relating to the multiphase models implemented as beta features in Ansys Fluent 2021 R2.



## 14.1. Interfacial Viscous Dissipation Method

### 14.1.1. Theory

The Interfacial Viscous Dissipation method introduces the artificial viscous damping term in the momentum equation. The term is active near the fluid-fluid interface and biased towards the lighter phase. The method is available with the VOF model only.

In this method, the interfacial viscosity is represented as:

$$\mu_{a,ij} = D_{ij} \max\left(\mu_i,\ \mu_j\right) \tag{14.1}$$

where

$\mu_{a,ij}$ = artificial interfacial viscosity between phases $i$ and $j$

$\mu_i$, $\mu_j$ = viscosities of phases $i$ and $j$

$D_{ij}$ = dissipation intensity

The method uses the dissipation intensity to control the effectiveness of the viscous damping treatment.

Although the method can improve solution stability for cases suffering from large spurious currents in the interfacial cells, the artificial nature of the numerical treatment may adversely affect the interfacial shear stress and the effects of other physical phenomena in certain applications. Consequently, this method is only suitable for cases where the lighter phase is of not much interest.

### 14.1.2. Using the Interfacial Viscous Dissipation Method

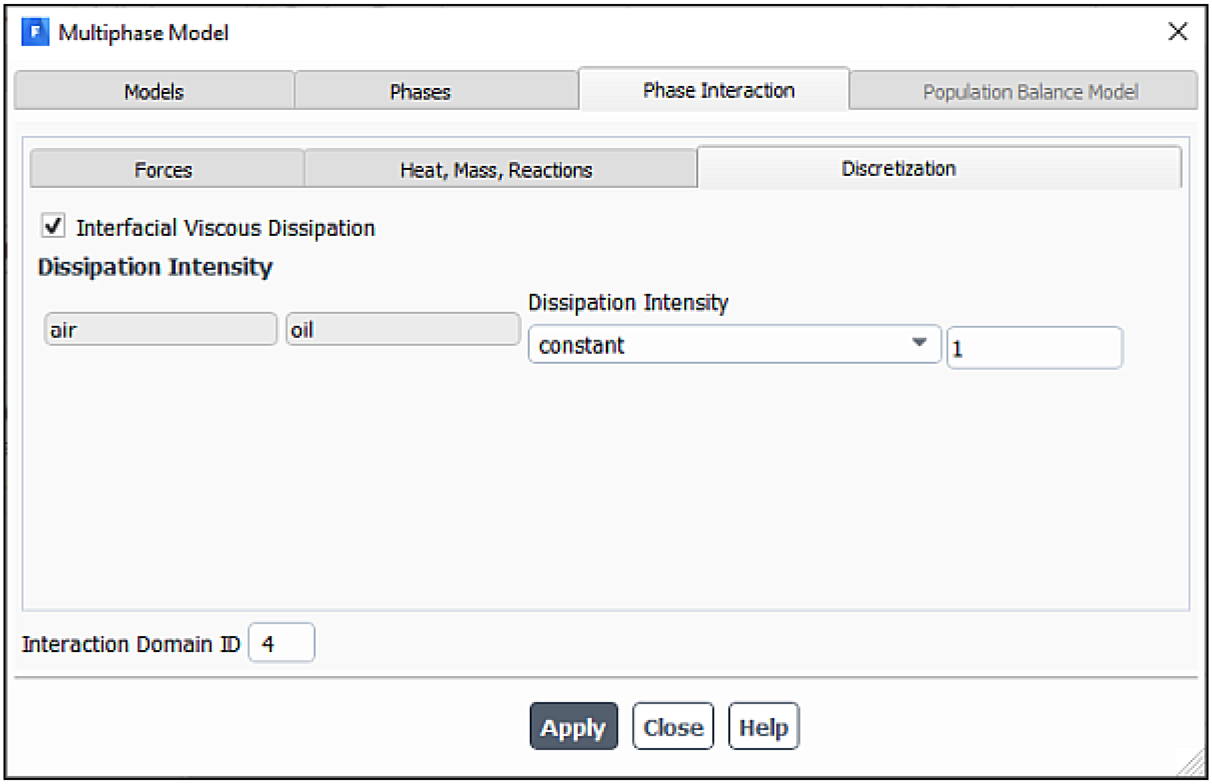
To use the Interfacial Viscous Dissipation method, follow the steps below:

1. Make sure your first enable beta feature access as described in Introduction (p. 1).

2. Set the Interfacial Viscous Dissipation method.

   **Physics** → **Models** → **Multiphase...**

   a. In the **Phase Interaction** > **Discretization** tab of the **Multiphase Model** dialog box, select **Interfacial Viscous Dissipation**.




   b. Enter a value for **Dissipation Intensity**.

      Note that setting the dissipation intensity above the default value of 1 may have an adverse effect on solution stability in some cases. In general, using values higher than 10 is not recommended.

### 14.1.3. Postprocessing

When the solution is complete, you can generate graphical plots or alphanumerical reports of artificial interfacial viscosity after enabling the following text command: `solve/set/advanced/retain-temporary-solver-mem`.

The **Artificial Interfacial Viscosity** variable becomes available in the field variable drop-down list under the **Phases...** category in the postprocessing dialog boxes.

## 14.2. Evaporation-Condensation Binary Mass Transfer Mechanism

The saturation data lookup table for a binary mixture can be used for modeling mass and heat transfer in the multiphase flash calculation. The procedural steps for incorporating the NIST binary saturation data into a multiphase case are listed below.

1. Create a binary mixture saturation table for a single-phase case as described in Using the NIST Real Gas Models in the Ansys Fluent User's Guide.

   Note that the binary saturation table contains only sets of saturation temperature (bubble and dew points) vs. saturation pressure data for different mixture compositions.

2. Exit from the current Ansys Fluent session.

   ---

   **Note:**

   Creating the NIST binary mixture saturation table for a single-phase flow case and then using this table with the multi-phase model enabled within the same Ansys Fluent session is not recommended.

   ---

3. Start a new Ansys Fluent session.

4. Enable beta feature access as described in Introduction (p. 1).

5. Enable the **Species Transport** model (if not already enabled).

6. For liquid and vapor phases, create mixture materials with exactly the same species as in the NIST binary table.

   Note that in the Ansys Fluent materials database, there is a limited number of materials that are available in both liquid and vapor phases. If the material you are creating does not exist in the Ansys Fluent database in liquid and vapor forms, you must create new materials with appropriate properties for both liquid and vapor mixtures.

7. Enable the Eulerian multiphase model.

8. In the **Multiphase Model** dialog box, select **Evaporation-Condensation** (**Eulerian Parameters** group box) and ensure that the **Lee Model** (**Evaporation-Condensation Model Options** group box) is selected.

9. In the **Primary Phase** dialog box, from the **Phase Material** drop-down list, select the liquid mixture material that you have created.

10. In the **Secondary Phase** dialog box, from the **Phase Material** drop-down list, select the vapor mixture material that you have created.

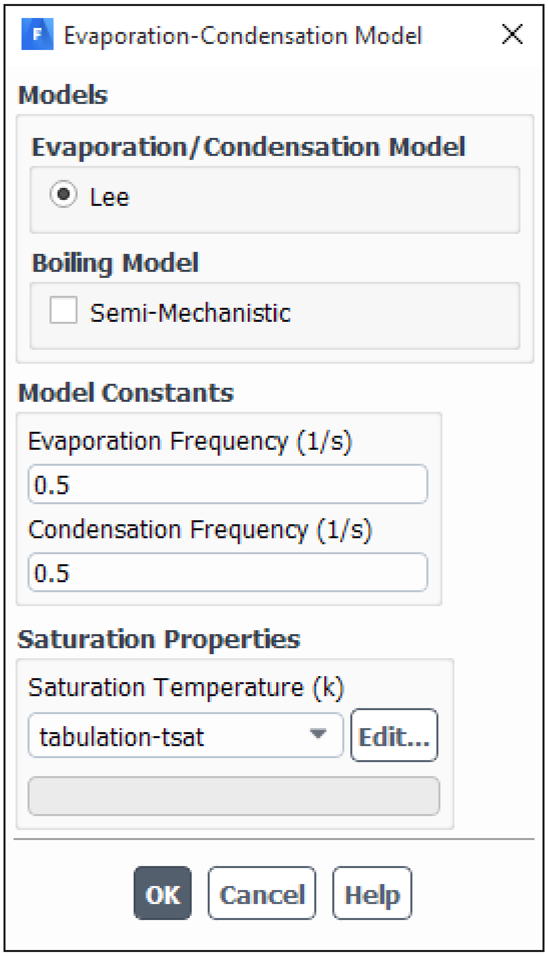
11. In the **Phase Interaction** dialog box, in the **Mass** tab, define the evaporation-condensation mass transfer mechanism.

    1. Set **Number of Mass Transfer Mechanisms** to 1.

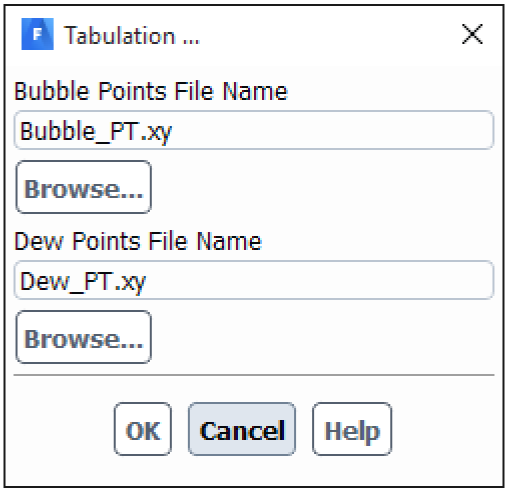
2. Select the liquid mixture material from the **From Phase** drop-down list and the vapor mixture material from the**To Phase** drop-down list.

3. From the **Mechanism** drop-down list, select **evaporation-condensation**.

4. In the **Evaporation-Condensation Model** dialog box that opens, from the **Saturation Temperature** drop-down list, select **tabulation-tsat**.




5. In the **Tabulation...** dialog box that opens, enter the **Bubble Points File Name** and **Dew Points File Name**. You can use the **Browse** button to navigate to a directory/file if you wish.




The saturation data (bubble and dew points) that have been read into Ansys Fluent will apply to the mass and heat transfer models during simulation.

**Note:**

Note that using this beta feature may pose additional numerical challenges.

## 14.3. Using the NITA Scheme with the Mixture Multiphase Model

After enabling the beta feature access (as described in Introduction (p. 1)), you can use the non-iterative time-advancement (NITA) scheme in Mixture multiphase transient flow simulations. The NITA solver significantly reduces computational time and CPU resources. For background information about the non-iterative time-advancement scheme, see the Fluent Theory Guide.

Details about using the NITA option are described in the section on setting solution controls for the non-iterative solver in the Fluent User's Guide.

You can also use hybrid NITA expert options as described in section on non-iterative solver expert options in the Fluent User's Guide. The hybrid NITA expert method 1 (`compressible flow, heat/mass transfer`) is recommended for most mixture multiphase cases.

In case of convergence issues, you can increase the PISO neighbor correction up to 4.

If you are solving for slip velocities, you can specify the explicit relaxation factor for **Slip Velocity** in the **Solution Controls** task page.

## 14.4. Large Eddy Simulation (LES) Model for Eulerian Multiphase

### 14.4.1. Theory

In the LES approach, large eddies are resolved directly, while small eddies are modeled. Therefore, modeling approximations are required only for part of the turbulent spectrum that are less likely to be influenced by the flow and boundary conditions. If the mesh and time step allow a large proportion of eddies to be resolved, the LES model produce better resolution of transient, recirculation and vortex structures than RANS models. However, the challenge of a multitude of length and time scales present in turbulent flows is intensified due to the additional length and time scales caused by the secondary phases.

Within the Euler-Euler approach, the effects of the phase interfaces are averaged, and semi-empirical formulations are used to account for phase interactions. These models are generally developed for a certain type of multiphase flow regimes and, therefore, suffer from inaccurate prediction of flow-regime transitions, particularly when the flow geometry changes. The use of LES combined with data obtained from DNS and experiments has the potential of developing multiphase flow strategies that rely on underlying microscale physics and may accurately predict the flow-regime transition [1] (p. 70).

Numerous applications of the Euler-Euler based LES to dispersed multiphase flows are reported in literature [2] (p. 70), [3] (p. 70). [4] (p. 70), and [5] (p. 70). In existing interaction models, drag force depends on a power of slip velocity between phases, which, if averaged, produces extra turbulent correlations that are not directly modeled, but indirectly accounted for through other terms. Since the resolved velocities are used within LES, the ignored correlations are less significant and, therefore, LES has an impact on model constants used in the interaction models. Better interaction models are very much so a research topic at present [1] (p. 70), and the publications sited above more or less use the same interaction models as are used with RANS models.

The transport equations are filtered in LES. For CFD analyses, the most appropriate filter is the cell control volume. The filtered equations are the same as ensemble-averaged equations with the following differences:

- the variables represent resolved values rather than ensemble-averaged RANS values
- the turbulent correlations are the subgrid correlations rather than RANS correlations

In Ansys Fluent, the following methods are available for modeling the subgrid correlations:

- Mixture LES multiphase model

  This model uses mixture properties and velocities. It is applicable to multiphase flows when phases separate as in stratified or nearly stratified multiphase flows, and when the density ratio between phases is close to 1.

- Dispersed LES multiphase model

  This model Is applicable when there is clearly one carrier phase and the concentrations of the secondary phases are dilute. The turbulence correlations within the carrier phase are modeled, and the dispersed phase kinematic viscosity is assumed to be equal to the carrier phase kinematic viscosity or ignored altogether.

The LES subgrid models available for the Eulerian multiphase model are:

- Smagorinsky-Lilly model
- Dynamic Smagorinsky-Lilly model
- Wall adapting local eddy-viscosity (WALE) model
- Algebraic wall-modeled LES model (WMLES)

- Dynamic kinetic energy subgrid-scale model

The above models are described in section Subgrid-Scale Models in the Ansys Fluent Theory Guide.

The first four models use algebraic relations to obtain the subgrid correlations from resolved dependent gradients (such as velocity). Gradients are obtained:

- from mixture variables for the Mixture LES multiphase model

- from the carrier phase variables (such as velocity gradients) for the Dispersed LES multiphase model

For the dynamic kinetic energy subgrid-scale model, the transport equation is the same as its single-phase version described in section Dynamic Kinetic Energy Subgrid-Scale Model in the Ansys Fluent Theory Guide. However, like the RANS transport equation for the turbulent kinetic energy, it contains an extra source term for modeling the interaction between the dispersed phases and the continuous phase. These models are the same as their counterparts in RANS models described in section k-$\varepsilon$ Dispersed Turbulence Model in the Ansys Fluent Theory Guide.

The turbulent dispersion forces and turbulent interaction models used with LES are the same as those available for RANS models. However, unlike the RANS models that uses averages for all turbulence spectrum, here all turbulent parameters are based on values obtained for subgrid scales. Therefore, the impact of any errors resulting from these models is expected to be less significant for LES than for RANS. For more information about these models, see section Turbulent Dispersion Force and section Turbulence Interaction Models in the Ansys Fluent Theory Guide.

## 14.4.2. Usage

After enabling beta features access (Introduction (p. 1)), the **Large Eddy Simulation (LES)** option becomes available in the **Viscous Model** dialog box (**Model** group box) for Eulerian multiphase cases.

---

**Note:**

The Large Eddy Simulation (LES) model is not compatible with the Population Balance model.

---

Once you select **Large Eddy Simulation (LES)**, you can choose from the following LES subgrid-scale models:

- **Smagorinsky-Lilly**
- **WALE**
- **WMLES**
- **WMLES S-Omega**
- **Kinetic-Energy Transport**

For modeling the subgrid correlations, you can select from the following models in the **LES Multiphase Model** group box:

- **Mixture**
- **Dispersed**

See Theory (p. 67) for details about these models.

## Bibliography

[1] R.O. Fox. "Large-Eddy Simulation Tools for Multiphase Flows". *Annu. Rev. Fluid Mech.*. 44. 47–76. 2012.

[2] N.G Dean, T. Solberg, B. H. Hjertager. "Large Eddy Simulation of the Gas-Liquid Flow in a Square Cross-Sectioned Bubble Column". *UChemical Engineering Science*. 56. 6341–6349. 2001.

[3] D. Zhang, N.G Dean, J.A.M Kuipers. "Numerical Simulation of Dynamic Flow Behaviour in a Bubbly Column: A Study of Closure for Turbulence and Interface Forces". *Chemical Engineering Science*. 61. 7593–7608. 2006.

[4] M.T Dhotre, B. Niceno, B.L. Smith, M. Simiano. "Large-Eddy Simulation of the Large Scale Bubble Plume". *Chemical Engineering Science*. 64. 2692–2704. 2009.

[5] M.V. Tabib, P. Schwarz. "Quantifying Sub-Grid Scale Turbulent Dispersion Force and its Effect Using One-Equation SGS Large Eddy Simulation Model in Gas-Liquid and Liquid-Liquid System". *Chemical Engineering Science*. 66. 3071–3086. 2011. February 2016.

# 14.5. Expert Options for the QMOM

After enabling beta features as described in Introduction (p. 1), you can access expert options for the quadrature method of moments (QMOM). These options appear in the text user interface (TUI) under the following text menu:

```
define/models/multiphase/population-balance/expert/qmom
```

The available options are:

**`inversion-algorithm`**

Enables the use of the advanced moment inversion algorithm [1] (p. 71) for the solution of population balance equations. You can select from the following options:

- `0`: the product-difference algorithm. See section “The Quadrature Method of Moments (QMOM)” in theAnsys Fluent Theory Guide for details.
- `1`: the Wheeler algorithm.

**`realizable-moments?`**

Ensures the reailzability of the moments of the PBE.

The product-difference algorithm is used to compute the weights and abscissas from the moments of the particle size distribution (see section “Numerical Method” in the Ansys Fluent Theory Guide for details). Typically, positive values of the weights and abscissa ensure that the distribution function also remains positive. The positive moments are called "realizable". To avoid non-physical values of the corresponding weights and abscissa, the moments should always remain realizable.

**`print-realizable-moment-warning?`**

When enabled, prints diagnostics for the realizable moments during the QMOM solution.

## Bibliography

[1] A. Passalacqua, F. Laurent, E. Madadi-Kandjani, J.C. Heylmun, R.O. Fox. "An open-source quadrature-based population balance solver for OpenFOAM". *Chemical Engineering Science*. 176. 306–318. 2018.

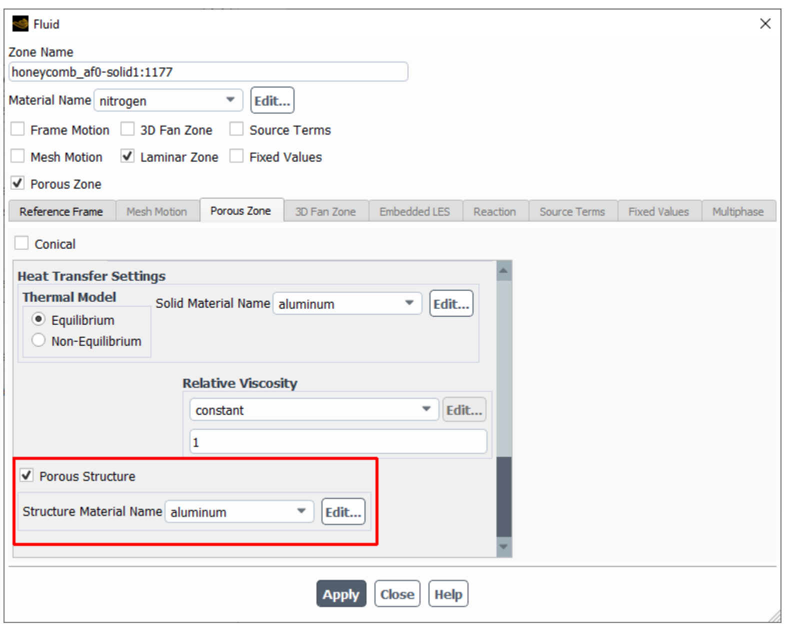
# Chapter 15: The Structural Model for Intrinsic Fluid-Structure Interaction (FSI)

This chapter contains information relating to the structural model for intrinsic fluid-structure interaction (FSI) implemented as beta features in Ansys Fluent 2021 R2.

## 15.1. Porous Structure Modeling

The structural model allows you to calculate deformation of porous structures. To set up such a simulation, enable beta features access (as described in Introduction (p. 1)). Once beta features are enabled, for every porous zone that you intend to consider as a porous structure, select the **Porous Structure** option within the **Porous Zone** tab for the corresponding porous zone(s).

**Figure 15.1: The Porous Structure Option When Defining Porous Zones**




Once the **Porous Structure** option is enabled, select the **Structure Material Name** to be used within the porous structure. All porous zones that are declared as porous structures will be added to the overall list of zones on which the structural model is intended to be applied. All porous structures must be enabled up-front (similarly to how solid zones are available up-front) before the general structural model is enabled. This will allow Fluent to determine whether the case set up contains any incompatibilities relevant for the structural model.

The porous structure feature can also be enabled using either of the text user interface (TUI) commands:

```
/define/boundary-conditions/fluid
```

```
/define/boundary-conditions/fluid/set/fluid
```

Refer to the following sections for details about the theoretical basis of the porous structure model:



### 15.1.1. Constitutive Equations and Finite Element Discretization

The constitutive equations that are used by the structural model for solid zones are equally applicable to porous structures as well, however, care is required when defining and selecting structural materials within the porous structure. Such material must take into account the level of porosity (or any other relevant properties) that may be present within the porous material.

### 15.1.2. Boundary Conditions

As with solid zones, the boundary conditions for porous structures are available at free walls as well as at coupled walls between the fluid and solid zones. Additionally, structural boundary conditions can be applied to many other zones that may surround the porous structure, including, velocity-inlet, pressure-inlet, pressure-outlet, mass-flow inlet, mass-flow outlet, outflow, and pressure-far-field. For boundaries between fluid and porous-structure zones, the structural boundary conditions can also be set to a porous jump.

# Chapter 16: Reduced Order Models (ROMs)

## 16.1. Manual Production of ROM Files from Stand-Alone Fluent

You can generate the files required for producing a reduced order model (ROM) outside of the Ansys Workbench environment.

Producing a ROM using **3D ROM** in Ansys Workbench requires that you have the following:

- An Ansys Fluent case file with the **Reduced Order Model** defined
- A ROM mesh file (`rommsh`)
- Snapshot files for each of the solved input parameters

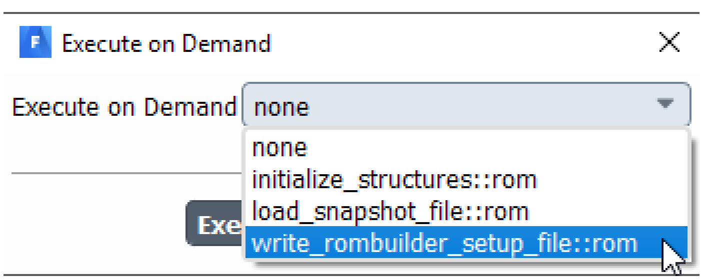
To generate these files, use the following steps:

1. Create a Fluent case file that has the **Reduced Order Model** defined (*Refer to Defining a ROM in the Fluent User's Guide for additional information*).

   **Setup** → **Models** **Reduced Order Model** → **Edit...**

2. Use the **Execute on Demand** dialog box to tell Fluent to generate the setup files required for creating the mesh and snapshot files.

   **User-Defined** → **User Defined** → **Execute on Demand...**

   a. Select `write_rombuilder_setup_file::rom` from the drop-down and click **Execute**.

   b. Close the **Execute on Demand** dialog box.

3. Enter the `(%rom-write-mesh-file "mesh.rommsh")` command into the console and press **Enter**.

   *This command creates the `mesh.rommsh` file in the Fluent working directory. This mesh is used to display the ROM.*

4. Review your input parameters to ensure they are setup as you intend. This includes setting the input parameters to the desired initial value.

   **Parameters & Customization** → **Parameters** → **Input Parameters**

5. Initialize the solution and click **OK** to dismiss the **Question** dialog box that may appear. *It is OK to clear the data that you generated before beginning the ROM process. You can always write case and data files before initializing the solution, if you have concerns.*

   **Solution** → **Initialization** → **Initialize**

6. Begin the calculation.

   **Solution** → **Run Calculation** → **Calculate**

7. Once the calculation is complete, enter the `(%rom-write-snapshot-file "snp_1.romsnp")` command into the Fluent console and press **Enter**, to generate the snapshot file for these input parameter values.

8. Generate snapshot files for the remaining design points that you intend to explore by following the preceding four steps (specify input parameters, initialize, calculate, create snapshot file).

9. Save the case file with the defined ROM. (**File** → **Write** → **Case**).

Once you have generated the ROM snapshot files and ROM mesh file, you are ready to produce the ROM.

## Generating a 3D ROM from Manually Produced Snapshot and Mesh Files

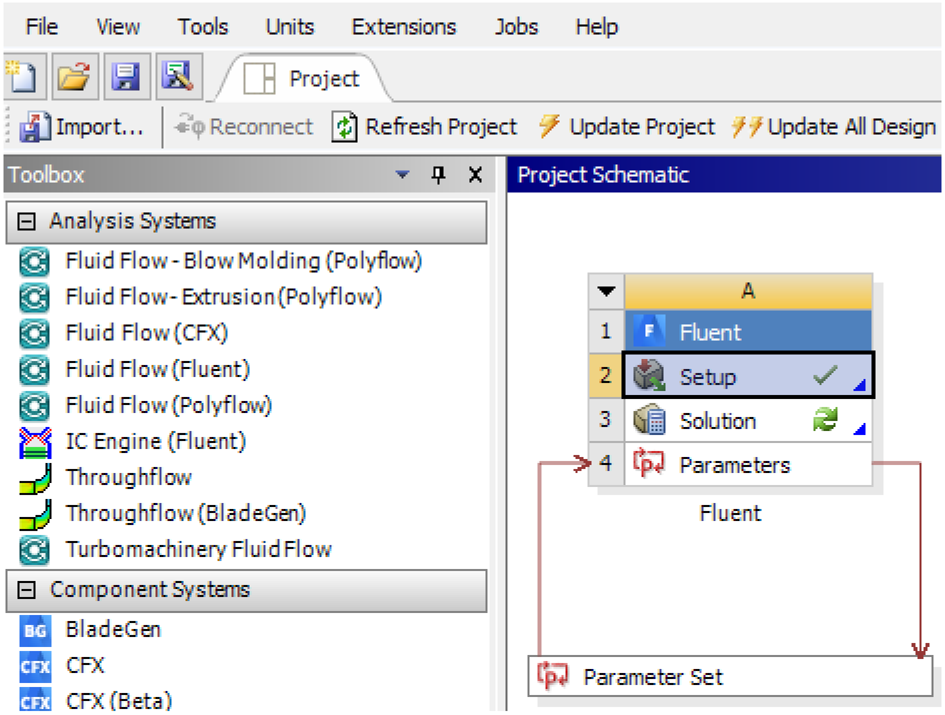
1. Look in your Ansys installation directory to locate `Rom.MeshProcessing.exe` (<installation directory>\ANSYS inc\<version>\Addins\ROM\bin\Win64 (or Linux64)), then navigate to this location from the Windows or Linux command line.

2. Run the following command on the Windows or Linux command line:

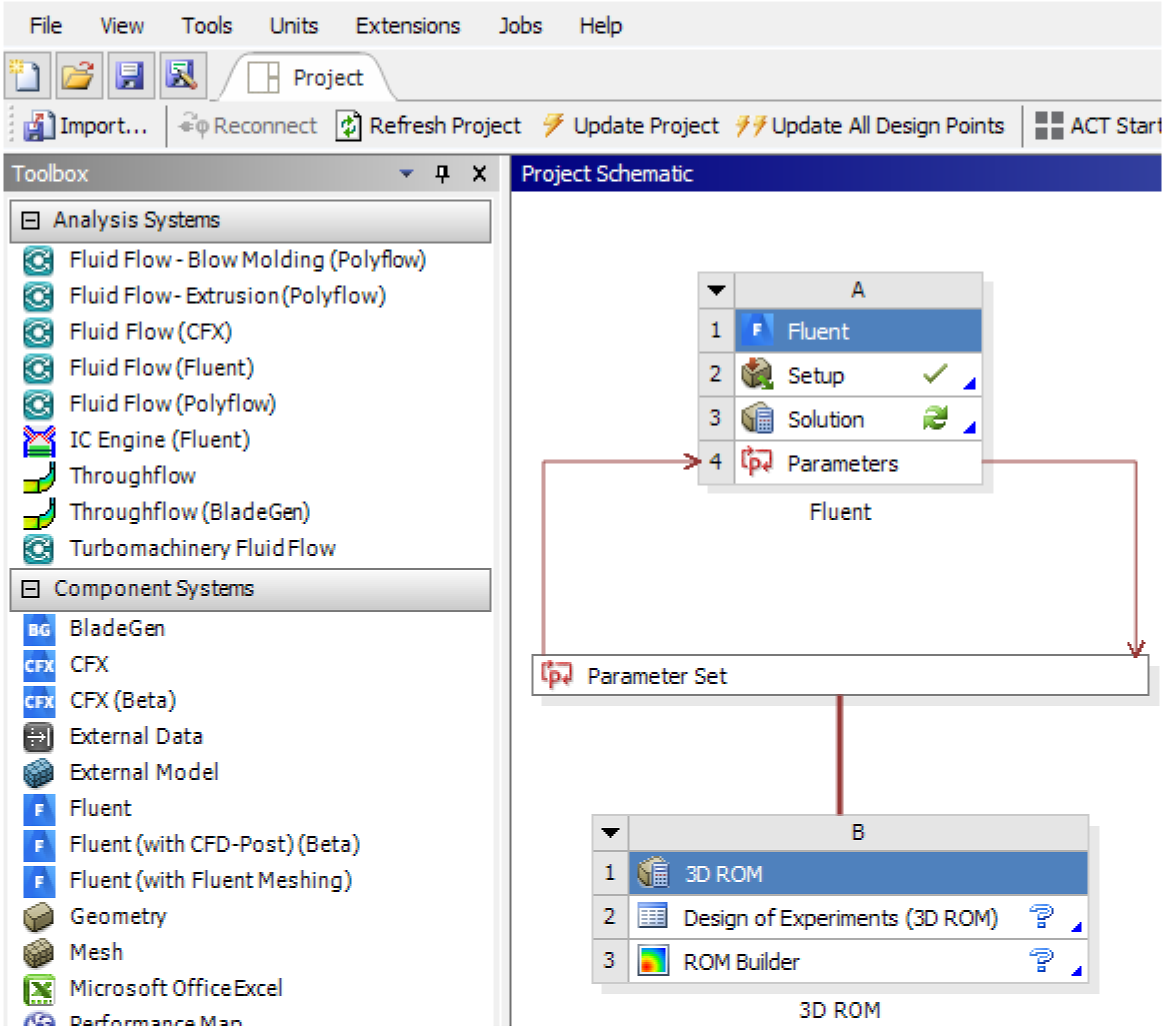
   ```
   Rom.MeshProcessing.exe <path to Fluent working directory>\mesh.rommsh
   <path to Fluent working directory>\mesh_processed.rommsh
   ```

   *This command creates the* `mesh_processed.rommsh` *mesh file in the Fluent working directory.*

3. Start Ansys Workbench, open a Fluent component system, and load the case file you saved from stand-alone Fluent (with ROM defined).




4. Double-click **3D ROM** under **Design Exploration** to connect a 3D ROM system to the Parameter Set bar of the Fluent component system.



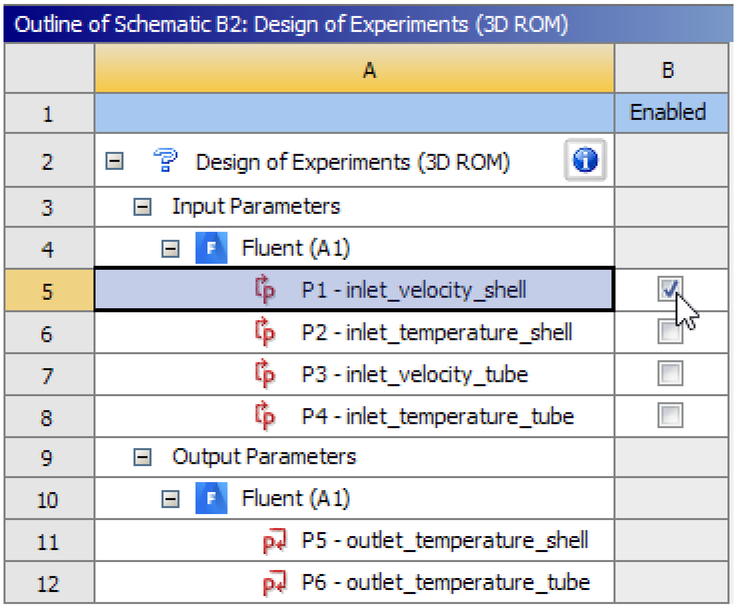

5. Transfer the ROM settings of the Fluent system to the 3D ROM system by opening Fluent, then closing it again. (Double-click the Fluent **Solution** cell, then close Fluent once it opens). *Alternatively, you can right-click the **Solution** cell and click **Update**, however this will take longer, as Fluent is re-solving for the specified design points.*

6. Double-click **Design of Experiments (3D ROM)** in the **Project Schematic**, to open the **Design of Experiments (3D ROM)** tab.

7. Enable **Show Advanced Options** for **Design Exploration** in the **Options** dialog box (**Tools** → **Options...**).

8. Define the parametric space in the **Design of Expriments (3D ROM)** Outline view.

    a. Select the design point parameters that you want included in the ROM by enabling/disabling them in the Outline view (column **B**).




b. Specify bounding values (such as minimum and maximum) in the Properties view for the enabled parameters.

c. Click **Preview** in the toolbar to generate the design points without updating them.

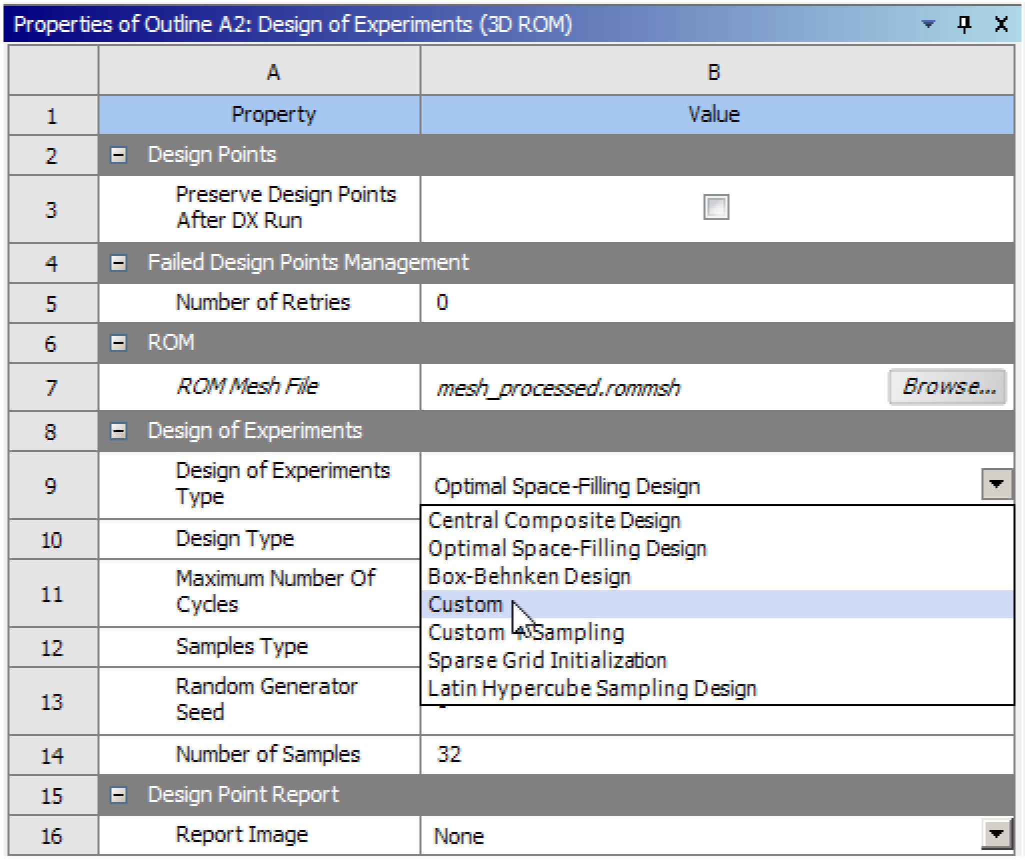
9. Load the processed ROM mesh.

   a. Select **Design of Experiments (3D ROM)** in the Outline view to display its properties.

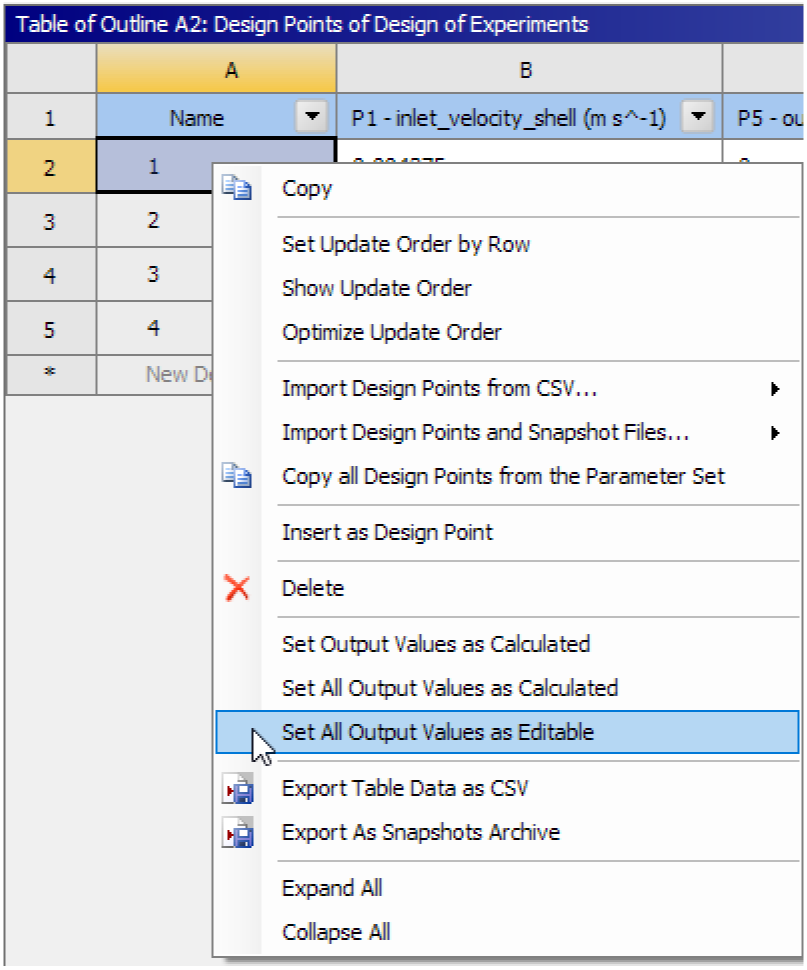
   b. Click **Browse...** in the **ROM Mesh File** cell and select `mesh_processed.rommsh`.

   c. Click **Copy and Proceed** when prompted.

10. Select **Custom** from the **Design of Experiments Type** drop-down.



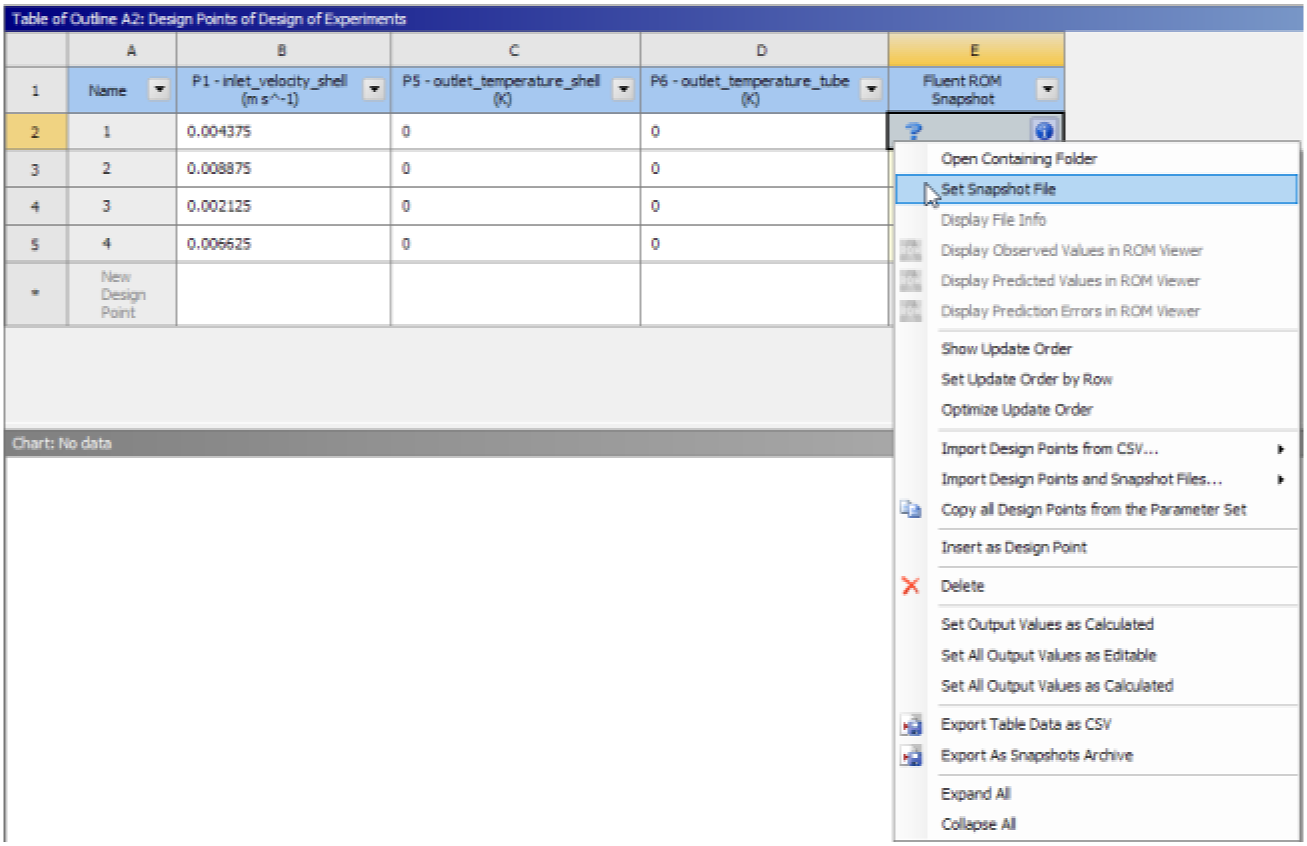

11. Right-click in the **Table** pane and select **Set All Output Values as Editable**.




12. Enter placeholder values for the cells of columns **P5** and **P6** (0 is an appropriate placeholder value).

13. Manually set the snapshot file for each design point:

    a. Right-click the **Fluent ROM Snapshot** cell for the first design point, browse to `snp_1.romsnp` and click **Copy and Proceed**.



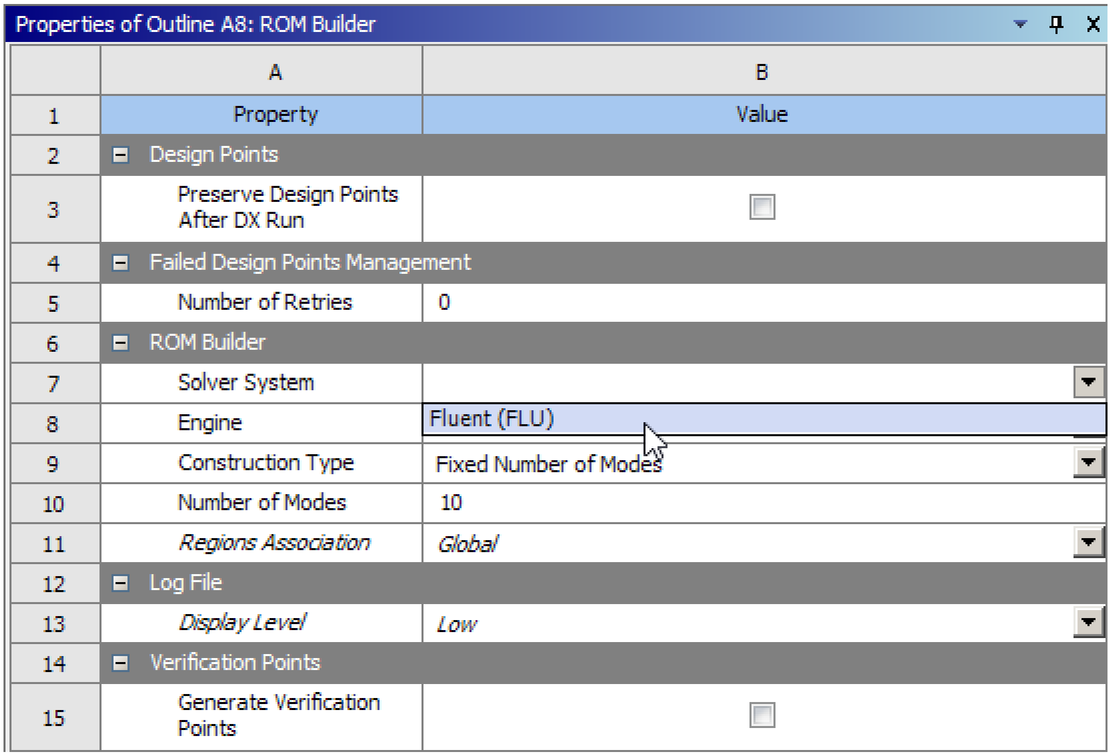

b. Repeat this process for all of the design points.

14. Save the Ansys Workbench project and update the **Design of Experiments (3D ROM)** cell (by right-clicking it and selecting **Update**).

    *This operation is fast because all of the outputs are already calculated.*

15. Click the **ROM Builder** cell to display its properties.

16. Select **Fluent (FLU)** as the **Solver System** (under **ROM Builder**).




17. Update the **ROM Builder** cell to create the ROM.

18. You can now open the ROM in the ROM Viewer to review and analyze the results.

For additional information on using 3D ROMs as well as an example, refer to *the DesignXplorer Beta Features Manual*.

# Chapter 17: Solver

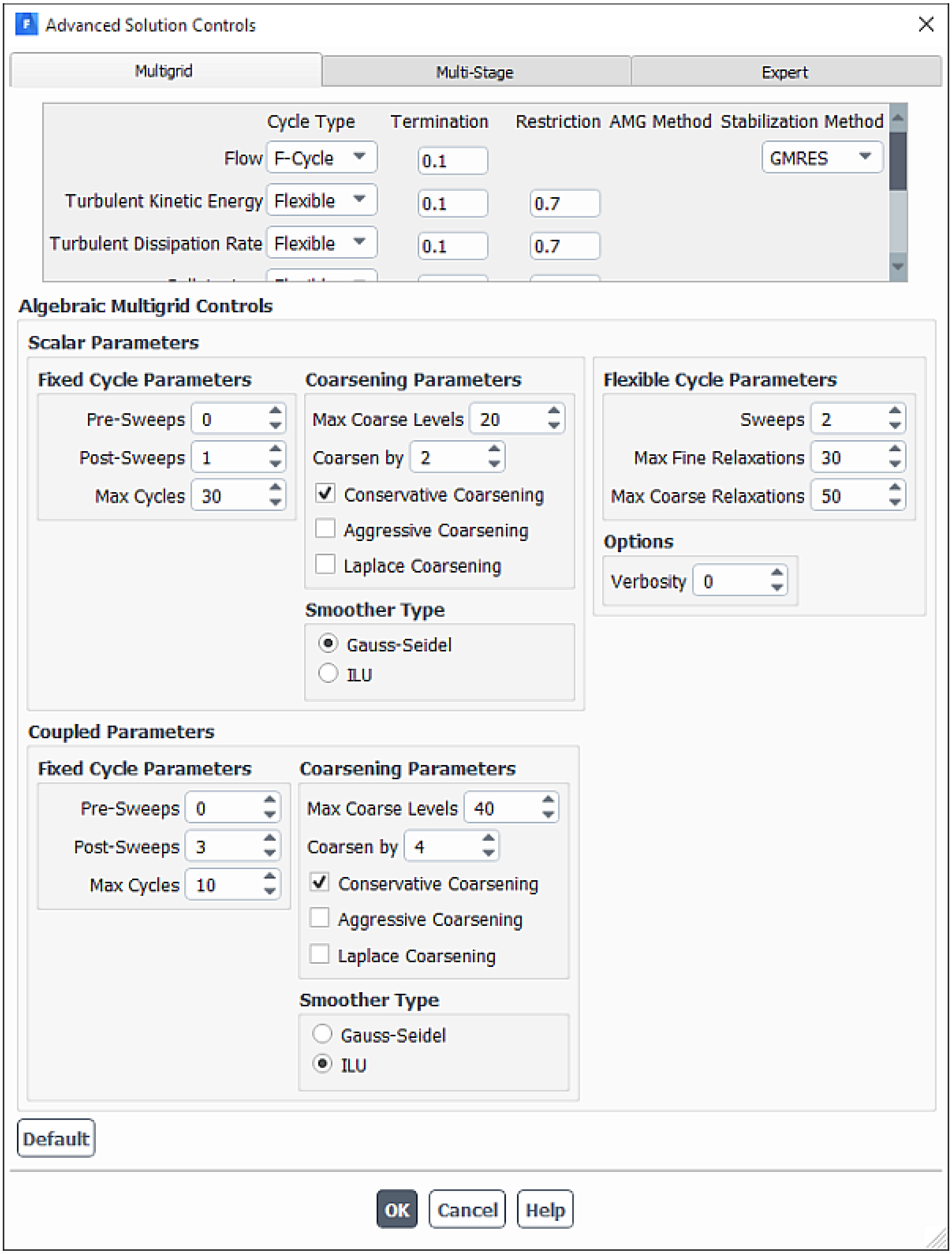
This chapter contains information about the beta features related to the following topics:



## 17.1. Stabilization Methods for the Density-Based Solver

When using the implicit density-based solver, you can choose another stabilization method rather than the AMG solver if you are using a fixed-type cycle (F, V, or W cycle) for the flow correction. If you enable beta features access (as described in Introduction (p. 1)), then you can select one of the following for **Flow** in the **Multigrid** tab of the **Advanced Solution Controls** dialog box: the bi-conjugate gradient stabilized method (**BCGSTAB**), the recursive projection method (**RPM**), or the generalized minimal residual method (**GMRES**). For further details on these stabilization methods, see Spatial Discretization in the User's Guide .

**Figure 17.1: The Multigrid Tab in the Advanced Solution Controls Dialog Box**



## 17.2. Reduced Rank Extrapolation (RRE) Method

The Reduced Rank Extrapolation method (RRE) is a technique used to accelerate the convergence of numerical methods involving nonlinear iterative solution algorithms. A Finite Volume simulation of the Navier-Stokes flow equations, as performed in Ansys Fluent, is such a method. Benefits of the application of RRE include better convergence rates, removal of residual stalling, and improved coupling between equations among different numerical models. The algorithm is independent of the type of flow solver and equally applicable to explicit, implicit, and pressure and density based algorithms. The drawback is an increased memory consumption.

The RRE method is representative of the set of so called Krylov type methods. Such methods explore the idea originally proposed by Krylov [3] (p. 97) that the solution of the linear system $Ax=b$ lies within the space defined as

$$K_k(A,b)=span\{b,Ab,AAb,...,A^{k-1}b\} \tag{17.1}$$

which is also known as the Krylov subspace.

The RRE method obtains a vector $x_n$ as a linear combination of orthogonal basis vectors of $K_n$, which minimizes the Euclidian norm of the residual vector $r_n=Ax_n-b$. This is usually a better approximation of the solution at the given iteration. A linear least squares procedure is applied in order to solve the minimization problem. Due to the memory requirements imposed by the need to keep solution vectors from previous iterations in memory, only a limited subset of the entire Krylov space $K_k$ is stored. The method operates on that subset once the user specified size $l$ (i.e. number of nonlinear flow solver iterations) is reached. Then a new subspace is populated with the solution vectors from the following $l$ iterations and subsequently the RRE procedure is restarted. It was found that in certain cases it might be beneficial to not store the solution vector at each iteration, but to skip a certain number of iterations. This is because of the fact that two subsequent solution vectors are almost linearly dependent and do not provide much new information for the Krylov subspace. The number of skipped iterations is also a user specifiable value.

Ansys Fluent's RRE method operates simultaneously on a predefined set of main flow variables, which are stored into a shared solution vector $x_n$ and uses the regular flow solver as a preconditioner to generate the products $A^k x$.

It can be shown that when applied to a linear problem, the RRE method is equivalent to the GMRES method derived by Y. Saad and M. H. Schultz [6] (p. 97). Details of the method as well as investigations of its behavior when applied to the Navier-Stokes problems can be found in [7] (p. 97) and [2] (p. 97).

**Note:**

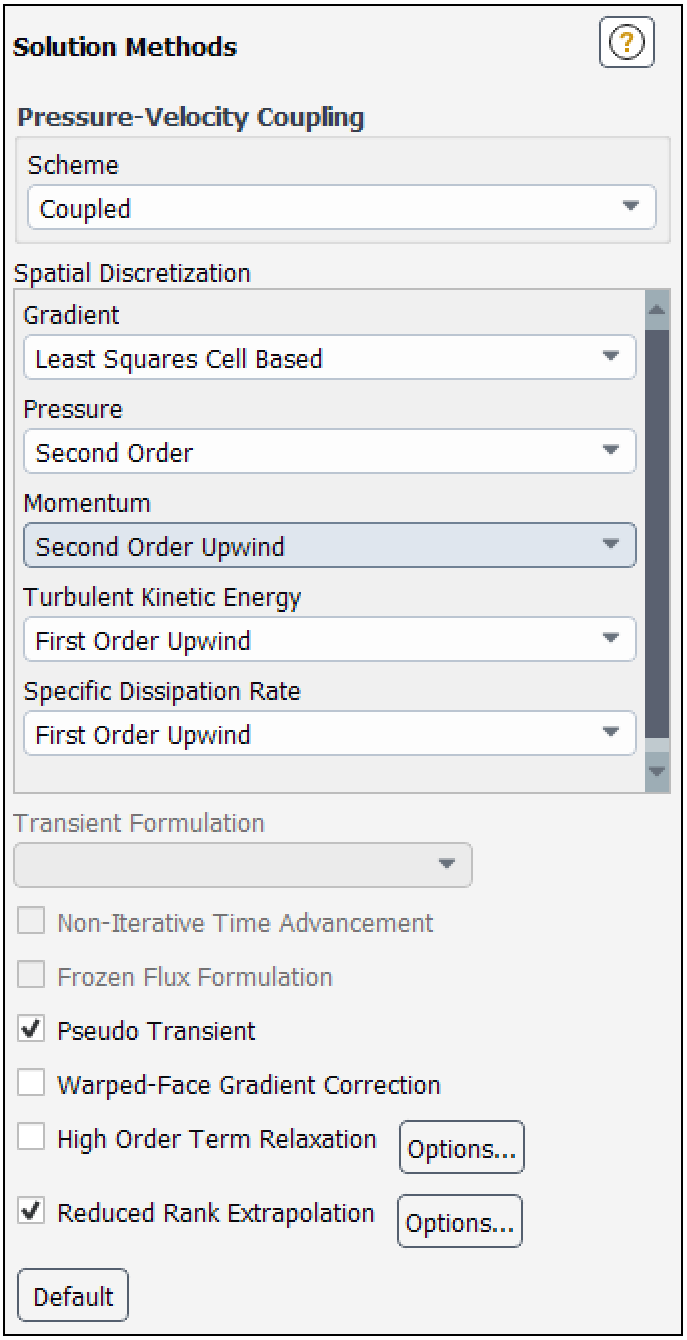
The Krylov subspace data is not written into the data file for reasons of keeping the file size small and the i/o times limited. A certain jump of residual values if a precomputed data is read into a new Ansys Fluent session is expected.

To use the **Reduced Rank Extrapolation** option, follow the steps outlined below:

1. Make sure you enable access to the beta features (Introduction (p. 1)).

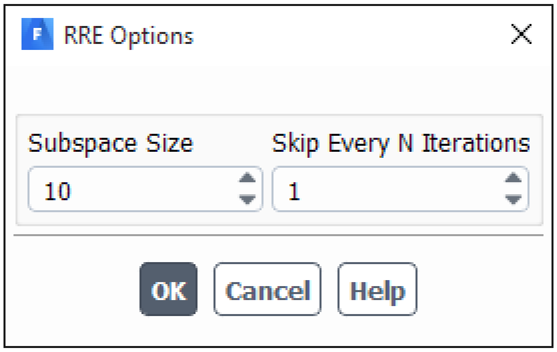
2. In the **Solution Methods** task page, enable the **Reduced Rank Extrapolation** option.

**Figure 17.2: The Solution Methods Task Page**




3. Click the **Options...** button to specify the RRE settings in the **RRE Options** dialog box.

**Figure 17.3: The RRE Options Dialog Box**




a. Specify the size of the subspace in the **RRE Options** dialog box.

b. Specify the number of iterations to skip while building it.

---

**Note:**

In certain cases, it may be beneficial to increase the size of the Krylov subspace to 25 and store each 10th vector.

---

4. Run the solution.

## 17.3. Executing Commands at a User-specified Iteration or Time Step

You can make modifications at a particular point during a simulation with the `Execute command only once?` command. This command gives you a choice to execute commands:

- at the beginning of a particular iteration, or
- at the end of a particular time step.

The purpose of this option is to give you greater control over when modifications are applied; changes are made at a specified iteration or time step even if you halt the run and write the files. The `Execute command only once?` command is available through the text user interface (TUI) console.

To activate this feature, enable beta feature access (Introduction (p. 1)) and set the scheme session variable `execute-command-at?` to `true`:

```
> (set! execute-command-at? #t)
```

### 17.3.1. Executing a Command at a Particular Iteration

The following example demonstrates how to execute a command at the beginning of a given iteration:

```
/solve/execute-commands> add-edit
Name of the command [command-1] command-1
Adding command-1
Execute command only once? [no] yes
```

```
Options: "iteration"
When ["iteration"] "iteration"
iteration no. [1] 5
Command [""] "display close-window 1"
```

**Note:**

This is available for both transient and steady-state cases.

### 17.3.2. Executing a Command at a Particular Time Step

The following example demonstrates how to execute a command at the end of a particular time step:

```
/solve/execute-commands> add-edit
Name of the command [command-1] command-2
Adding command-2
Execute command only once? [no] yes


Options: "iteration" "time-step"
When ["iteration"] "time-step"
time-step no. [1] 5
Command [""] "display close-window 1"
```

**Note:**

This is available only for transient cases.

## 17.4. Alternative Rhie-Chow Flux With Moving Or Dynamic Meshes

In single phase unsteady simulations that use sliding, moving, or dynamic meshes, an alternative Rhie-Chow flux treatment can be enabled that includes an additional transient term in the formulation. This is expected to yield improved results in these cases. To enable the alternative Rhie-Chow formulation, enable beta features and use the following TUI command:

```
> /solve/set/moving-mesh-numerics
Alternative Rhie-Chow flux treatment for unsteady sliding, moving, and/or dynamic mesh? [no] yes
```

## 17.5. Automatic Solver Defaults Based on Setup

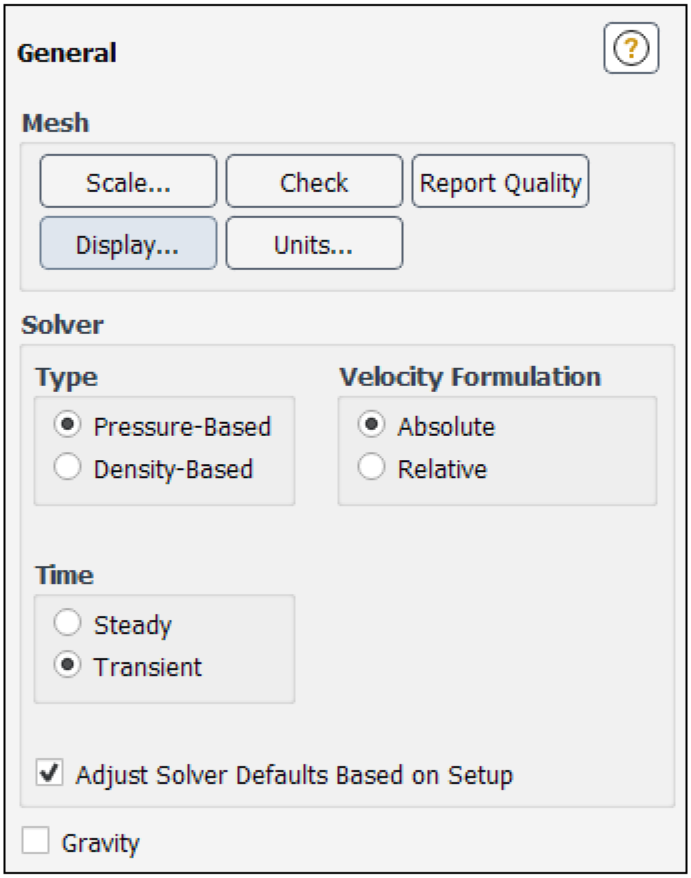
As a beta feature, you can have Ansys Fluent automatically adjust various solver parameters to recommended settings based on certain characteristics of the problem you are trying to solve. For example, the recommended settings often vary depending on whether the problem is steady or unsteady, compressible or incompressible, and so on. To enable the automatic solver defaults, perform the following steps:

1. Enable beta features using the following TUI command:

```
> define beta-feature-access
Enable beta features? [no] yes

It is recommended that you save your case/data files before enabling beta features.
This will assist in reverting to released functionality if needed.
```

```
OK to proceed? [cancel] OK

Beta features enabled.
```

2. Enable **Adjust Solver Defaults Based on Setup** in the **General** task page.




Once enabled, Fluent will automatically adjust solver settings to recommended values based on pre-defined contexts that correspond to various classes of problem. As settings are adjusted, Fluent will print information messages to the text console alerting you to the changes it has made. You can override the automatically chosen settings through the GUI or TUI as you normally would. Your changes to the context are persistent.

## 17.6. Roe Flux-Difference Splitting Scheme in the Pressure-Based Solver

When beta features are enabled, the Roe flux-difference splitting scheme can be used with the pressure-based solvers for compressible single-phase flow, both steady-state and transient. Compared with the default Rhie-Chow flux method (described in the section on discretization of the continuity equation in the Fluent Theory Guide), the Roe scheme shows much less dissipation in areas of strong pressure and velocity gradients in supersonic flows and produces less smearing of shocks. In can improve feature resolution in such areas at the expense of reduced numerical stability compared with the Rhie-Chow method.

### 17.6.1. Roe Flux-Difference Splitting Theory

The Roe flux-difference splitting scheme implements upwinding of the inviscid fluxes based on the direction of the characteristics of the flow equations [5] (p. 97). In the pressure-based solver framework, the matrix form of the preconditioned Roe flux (Equation 17.6 (p. 92)) is converted to scalar form with:

$$\mathbf{F}_f = u_f \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho H \end{pmatrix}_{up} + p_f \begin{pmatrix} 0 \\ n_x \\ n_y \\ n_z \\ 0 \end{pmatrix} \tag{17.2}$$

The fluxes contain convective and pressure contributions. The convective part is computed based on the face-normal velocity, $u_f$, and the upwinded flow variables. The pressure part is computed from the face pressure, $p_f$, projected on the appropriate coordinate directions. The face-normal velocity and pressure are expressed as Mach-weighted averages of the adjacent cell values [1] (p. 97):

$$\begin{aligned} u_f &= \lambda_0\left(M^*\right)u_L + \lambda_1\left(M^*\right)u_R - \lambda_p\left(M^*\right)\left(p_R - p_L\right) \\ p_f &= \beta_0\left(M^*\right)p_L + \beta_1\left(M^*\right)p_R - \beta_u\left(M^*\right)\left(u_R - u_L\right) \end{aligned} \tag{17.3}$$

where $\lambda_0$,$\lambda_1$, $\lambda_p$ and $\beta_0$, $\beta_1$, $\beta_u$ depend on the preconditioned face Mach number, $M^*$.

In the low-Mach-number limit, $M^* \rightarrow 0.44$ and the face velocity and pressure reduce to an approximately arithmetic average of the left and right cell values, plus dissipation:

$$\begin{aligned} u_f &= 0.72u_L + 0.28u_R - \frac{\Delta p}{2\rho_{up}U_{ref}} \\ p_f &= 0.72p_L + 0.28p_R - \tfrac{1}{2}\rho U_{ref}\Delta u \end{aligned} \tag{17.4}$$

For supersonic flow, the face velocity and pressure (Equation 17.3 (p. 92)) reduce to pure upwinding:

$$u_f = \begin{cases} u_L & M^* > 1 \\ u_R & M^* < -1 \end{cases} \quad ; \quad p_f = \begin{cases} p_L & M^* > 1 \\ p_R & M^* < -1 \end{cases} \tag{17.5}$$

The reduction to pure upwinding in the case of supersonic flow is in contrast with the Rhie-Chow method, in which the extra dissipation terms associated with $\Delta p$ and $\Delta u$ remain [4] (p. 97).

---

**Note:**

This equation is taken from the section on preconditioning in the solver chapter of the Fluent Theory Guide

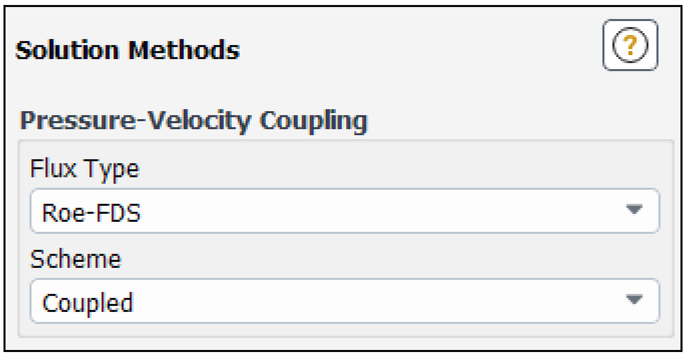
$$\Gamma \frac{\partial}{\partial t}\int_V Q dV + \oint [F - G] \cdot dA = \int_V H dV \tag{17.6}$$

---

### 17.6.2. Using the Roe Flux-Difference Splitting Scheme in the Pressure-Based Solver

The Roe flux method is available in the pressure-based solver for compressible single-phase flow. It is available for both steady-state and transient simulations using any of the pressure-velocity coupling schemes. Compared with the default Rhie-Chow flux method (described in the section on discretization of the continuity equation in the Fluent Theory Guide), the Roe scheme shows much less dissipation in areas of strong pressure and velocity gradients in supersonic flows and produces less smearing of shocks. In can improve feature resolution in such areas at the expense of reduced numerical stability compared with the Rhie-Chow method.

After enabling beta features (Introduction (p. 1)), the **Roe-FDS** method is available under **Flux Type** in the **Solution Methods** task page.




#### Recommendations

As a result of the reduced dissipation of the **Roe-FDS** flux method, divergence may occur during the solution. The following strategies may help alleviate divergence when using **Roe-FDS**.

- Start the simulation with the **Rhie-Chow** flux method and switch to the **Roe-FDS** method.
- Start with first order discretization of pressure, velocity, temperature, and density and then switch to higher-order discretization methods.
- Reduce the under-relaxation values, in particular for temperature.
- For steady-state flows, consider using the pseudo-transient method which has been observed to give better convergence. If divergence is observed with pseudo-transient, try reducing the time scale factor for the energy equation in the **Expert** tab of the **Advanced Solution Controls** dialog box (see the section on setting solution controls for the pseudo transient method in the Fluent User's Guide.

## 17.7. Improved Second Order Transient Formulations

The following beta features are available when you have selected a second-order **Transient Formulation** in the **Solution Methods** task page:

- You can enable the variable time step size formulation (as described in the Fluent Theory Guide and the Fluent User's Guide) for dynamic meshes. Note that this combination is only available with the pressure-based solver, and that dynamic meshes are not compatible with the **Bounded Second Order Implicit** formulation.

- You can enable a variant of that time formulation that provides better solution stability and boundedness. Note that these variants are available for both **Second Order Implicit** and **Bounded Second Order Implicit** formulations, and have the following limitations:
  - They are not available with the density-based solver, only the pressure-based solver.
  - They are only applicable to the implicit volume fraction discretization schemes for multiphase, and not to the explicit volume fraction schemes.
  - They do not support the Singhal et al. cavitation model.

  Note that if one of these variants is enabled for a case that uses a species model (not including **Species Transport**), you can also specify whether or not variables related to the combustion model are treated as bounded.

To enable any of these beta features, select the pressure-based solver, select a second-order transient formulation, enable beta feature access (Introduction (p. 1)), and then respond appropriately to the prompts in the following text command:

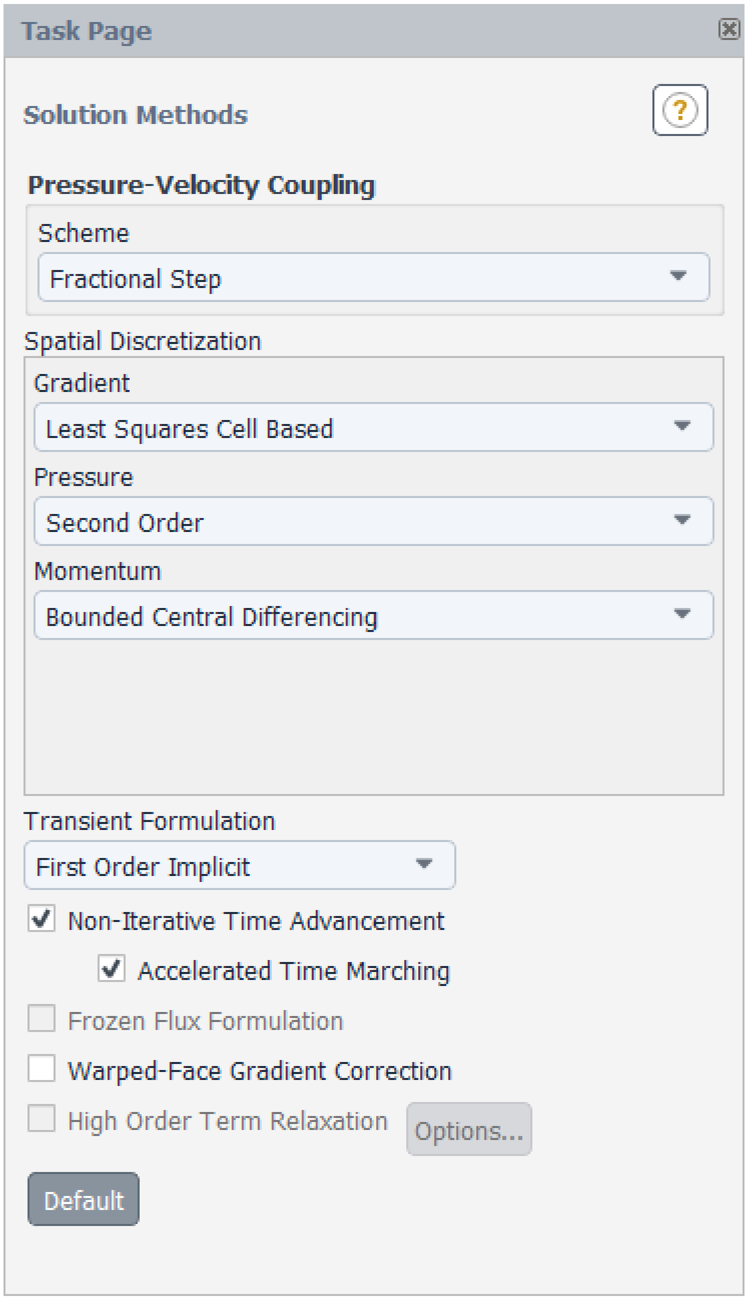
`solve` → `set` → `second-order-time-options`

## 17.8. Accelerated Time Marching with the Non-Iterative Solver

When using the Non-Iterative Time Advancement (NITA) scheme, it is possible to use the accelerated time marching option for cases in which the Large Eddy Simulation (LES) turbulence model is not enabled (even though this option is intended to be used with LES). To do so, perform the following steps:

1. Enable beta features access (as described in Introduction (p. 1)).
2. Enable the **Non-Iterative Time Advancement** option in the **Solution Methods** task page, and then enable the **Accelerated Time Marching** option immediately below it.

**Figure 17.4: The Solution Methods Task Page**




For further details about the **Accelerated Time Marching** option, see Setting Solution Controls for the Non-Iterative Solver in the Fluent User's Guide.

## 17.9. Hybrid NITA with Single-Phase Flows

When you want to use the Non-Iterative Time Advancement (NITA) scheme with a single-phase flow, hybrid NITA expert options are available that allow you to easily define a number of settings that are suitable for certain applications. These settings are designed to provide faster solver performance and better robustness without sacrificing accuracy. Depending on your case, the hybrid NITA expert settings may optimize NITA expert controls, explicit relaxation factors, and outer iterations.

To use the hybrid NITA expert options for a single-phase flow, perform the following steps:

1. Enable beta features access (as described in Introduction (p. 1)).
2. Enter the following text command:

   solve → set → nita-expert-controls → hybrid-nita-settings

   You will then be prompted to select from the following options:

   - 0: This ensures no hybrid NITA expert options are used.
   - 1: This enables hybrid NITA settings that are recommended for flows with a constant density, such as LES simulations.
   - 2: This enables hybrid NITA settings that are recommended for flows with a variable density, such as combustion simulations.

   If your selection results in a change to the settings, the changes will be printed in the console.

## 17.10. Equation Ordering for Multiphase Flows

For transient simulations that use the pressure-based solver, it is possible to specify that the model equations are solved in an order that is optimized for volumetric expansion when using the following multiphase models:

- the Volume of Fluid (VOF) model
- the Mixture model

This ordering can improve the speed of convergence for flows in which the density is strongly dependent on thermal effects, chemical composition, and so on (such as combustion simulations).

To use this optimized ordering in such circumstances, select the transient, pressure-based solver, select the appropriate multiphase model, enable beta features access (as described in Introduction (p. 1)), and enter the following text command:

```
> solve/set/equation-ordering

Equation Order [standard]>--> optimized-for-volumetric-expansion
```

## 17.11. Enhanced Poor Mesh Numerics

For cases that use the density-based implicit solver, you can improve the solver robustness on meshes of poor quality by issuing the following poor mesh numerics text command:

solve/set/poor-mesh-numerics/enhanced-pmn? **yes**

This option allows you to obtain a solution otherwise not possible in some cases by applying local corrections and adding numerical stability measures on the poor-quality cells. Note, however, that if the mesh quality did not hinder the stability of the density-based solver, using this text command option will not affect the solution convergence.

By default, the orthogonality-quality threshold below which a cell is considered ill-shaped is set to 0.2, the CFL limit is set to 1, and a 1$^{st}$ order scheme is selected. You can adjust the cell quality threshold by using the `solve/set/poor-mesh-numerics/set-quality-threshold` text command. The order of the poor-mesh-numerics scheme should not be changed.

## 17.12. References

### Bibliography

[1] V. A. Ivanov and B. P. Makarov. Improving Shock Capturing in Pressure-Based Solvers. *22$^{nd}$ Annual Conference of the CFD Society of Canada*. Toronto, Canada: June 1–4, 2014.

[2] A. Jemcov and J. P. Maruszewski. Nonlinear Flow Solver Acceleration by Reduced Rank Extrapolation. *AIAA journal 2008-609*.

[3] A.N. Krylov. On the numerical solution of the equation by which in technical questions frequencies of small oscillations of material systems are determined. *Izvestija AN SSSR (News of Academy of Sciences of the USSR)*. Otdel. mat. i estest. nauk, VII, Nr.4, (in Russian). 491–539. 1931.

[4] S. R. Mathur and J. Y. Murthy. A Pressure Based Method for Unstructured Meshes. *Numerical Heat Transfer*. 31. 196–216. 1997.

[5] P. L. Roe. Approximate Riemann Solvers, Parameter Vectors and Difference Schemes. *Journal of Computational Physics*. 43. 357–372. 1982.

[6] Y. Saad and M.H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput*. 7. 856–869. 1986. doi:10.1137/0907058.

[7] A. Sidi. Efficient Implementation of Minimal Polynomial and Reduced Rank Extrapolation Methods. *NASA Technical Memorandum 103240, ICOMP-90-20*.

# Chapter 18: Adaption

## 18.1. Anisotropic Adaption Based on the PUMA Method

PUMA-based anisotropic adaption is available through the `mesh/adapt/set/anisotropic-adaption?` text command (for details, see Refining and Coarsening in the User's Guide); if you would like to use the graphic user interface (GUI) to set it up, as well as use predefined criteria to easily set up the adaption controls with criteria that is commonly used for boundary layers, perform the following steps:

1. Read a 3D mesh that contains prismatic cells in the regions where you want to apply anisotropic adaption. Prismatic cells are hexahedral cells or wedge or polyhedral cells that have the same number of nodes on the top and bottom faces relative to the splitting direction.

2. Enable beta features access (as described in Introduction (p. 1)).

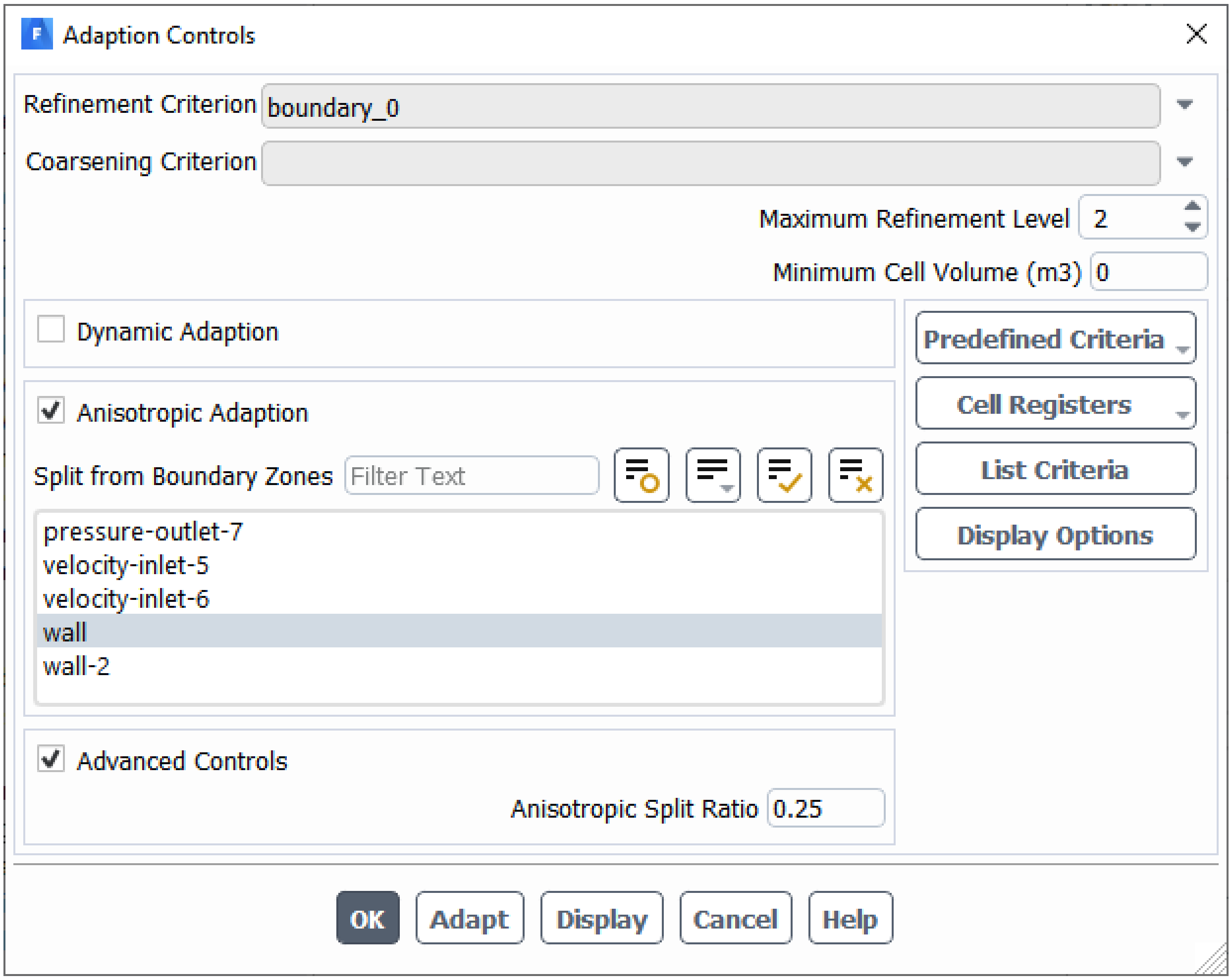
3. Ensure that the PUMA method is selected by using the following text command:

   `mesh` → `adapt` → `set` → `method`

4. Open the **Adaption Controls** dialog box.

   **Domain** → **Adapt** → **Refine / Coarsen...**

**Figure 18.1: The Adaption Controls Dialog Box**




5. You can make a selection from the **Predefined Criteria** drop-down list to easily set up the adaption controls with criteria that is commonly used for boundary layers (either for cells within a certain distance of a boundary zone or based on their $y^+$ or $y^*$ values). For further details, see Predefined Criteria for Boundary Layer Adaption (p. 101).

6. If you do not want to use **Predefined Criteria**, you can select a suitable cell register from the **Refinement Criterion** drop-down list; typically you would select a boundary or Yplus/Ystar register. Then enable the **Anisotropic Adaption** option, and make selections from the **Split from Boundary Zones** list to indicate the direction of splitting: the cells will only be split along the directions that are perpendicular to the selected zones.

   You can control exactly where the cells are split relative to the selections in the **Split from Boundary Zones** list by enabling the **Advanced Controls** option and entering a value for the **Anisotropic Split Ratio** field, which is defined as follows:

$$\text{split ratio}=\frac{\text{height (from the base face) of the splitting point on the edge}}{\text{original height of the edge}}$$

The default value for this ratio is 0.5.

---

**Note:**

The **Minimum Cell Orthogonal Quality** setting is not available as part of the **Advanced Controls** when the **Anisotropic Adaption** option is enabled.

---

7. Click **OK** to save the adaption control settings or click **Adapt** to adapt the mesh.

### 18.1.1. Predefined Criteria for Boundary Layer Adaption

When applying PUMA-based anisotropic adaption to a boundary layer, you can use the **Boundary Layer...** selections in the **Predefined Criteria** drop-down list to increase the ease of setup of the **Adaption Controls** dialog box. With these selections, Fluent will automatically generate the necessary cell registers for refinement, and will define the settings in the **Adaption Controls** dialog box accordingly.

To use predefined criteria for adapting a boundary layer, perform the following steps:

1. Read a 3D mesh that contains prismatic cells in the regions where you want to apply anisotropic adaption. Prismatic cells are hexahedral cells or wedge or polyhedral cells that have the same number of nodes on the top and bottom faces relative to the splitting direction.

2. Enable beta features access (as described in Introduction (p. 1)).

3. If you are going to adapt cells based on their $y^+$ or $y^*$ values, select a turbulence model in the **Viscous Model** dialog box, and then run the calculation to generate data.

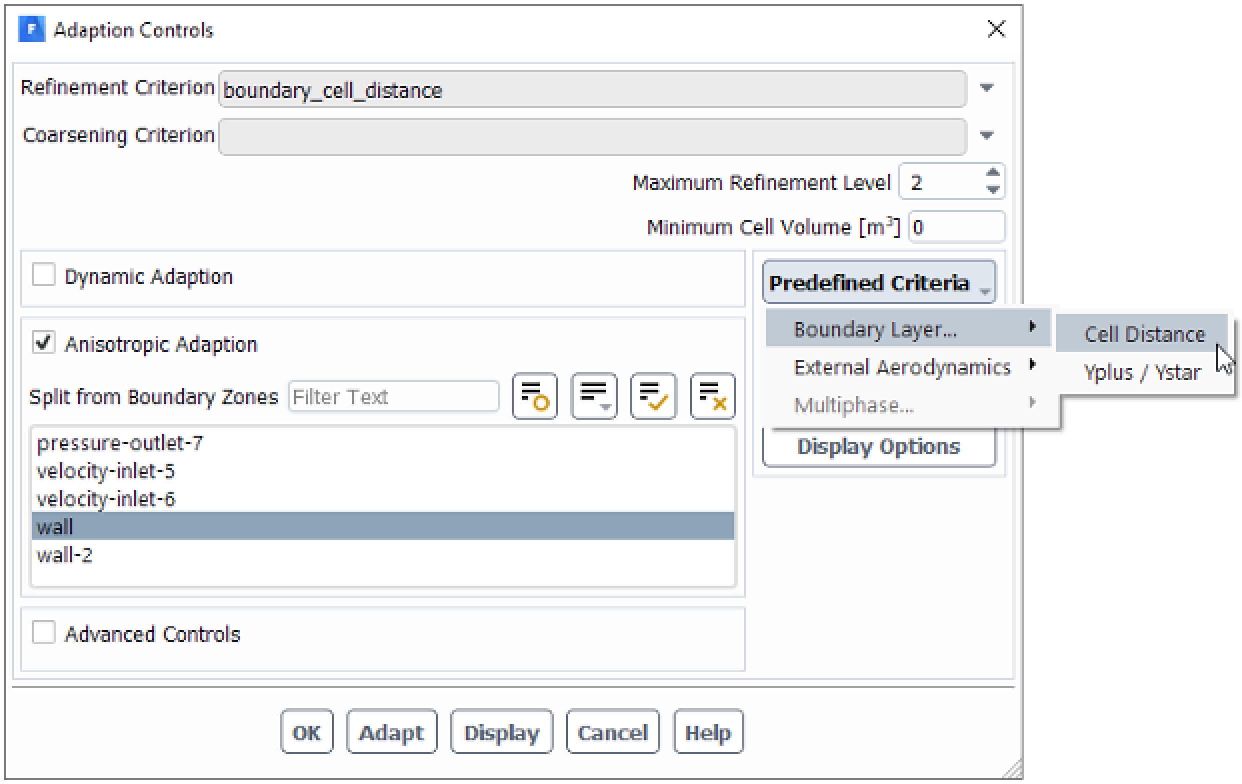
4. Ensure that the PUMA method is selected by using the following text command:

   `mesh → adapt → set → method`

5. Open the **Adaption Controls** dialog box.

   **Domain → Adapt → Refine / Coarsen...**

**Figure 18.2: Selecting Boundary Layer Predefined Criteria in the Adaption Controls Dialog Box**

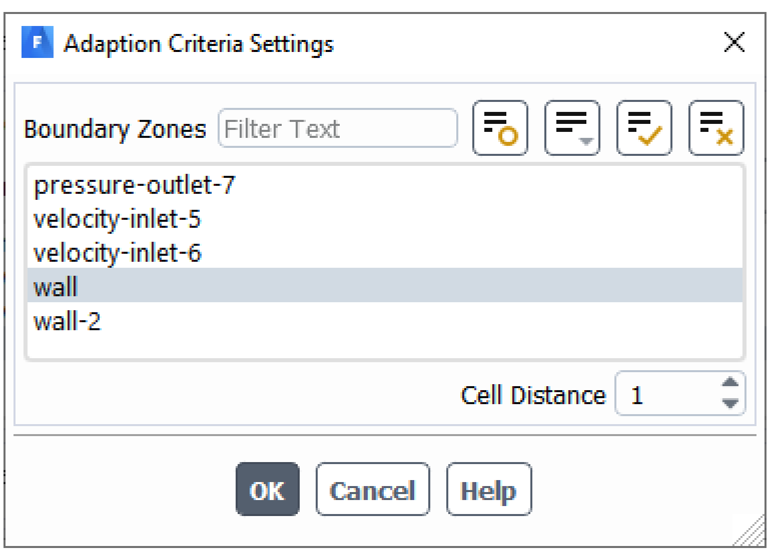



a. Select **Boundary Layer...** from the **Predefined Criteria** drop-down list, and then select one of the following from the menu that opens:

- **Cell Distance**: This sets up adaption based on a cell's proximity to one or more boundaries.

  When selected, the **Adaption Criteria Settings** dialog box opens to complete the setup. You must select the **Boundary Zones** where you want adaption, and define the **Cell Distance** to indicate the number of cell layers you want to adapt.

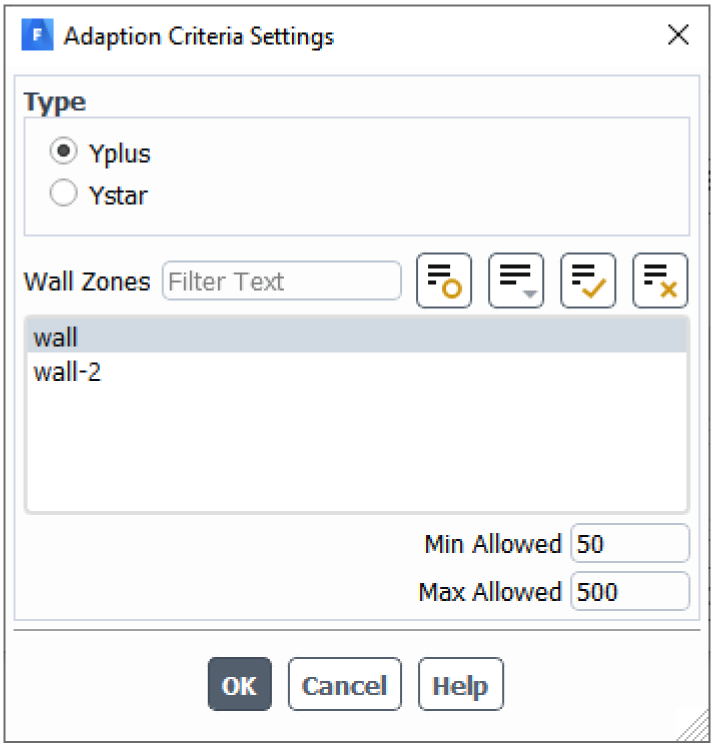
  **Figure 18.3: The Adaption Criteria Settings Dialog Box for the Cell Distance Criterion**



- **Yplus / Ystar**: This sets up adaption that will refine cells on specified viscous wall zones that have $y^+$ or $y^*$ values within a specified range.

  When selected, the **Adaption Criteria Settings** dialog box opens to complete the setup. You must select whether the adaption is based on **Yplus** or **Ystar** values. Then select the **Wall Zones** where you want adaption, and define the range by entering values for **Min Allowed** and **Max Allowed**.

  **Figure 18.4: The Adaption Criteria Settings Dialog Box for the Yplus / Ystar Criterion**


b. By default, the **Anisotropic Adaption** option will be enabled in the **Adaption Controls** dialog box, so that the cells are refined anisotropically where possible; such cells will only be split along the directions that are perpendicular to the zones selected from the **Split from Boundary Zones** list. Note that you can control exactly where the cells are split by enabling the **Advanced Controls** option and entering a value for the **Anisotropic Split Ratio** field, as described in Anisotropic Adaption Based on the PUMA Method (p. 99).

c. A cell register is created as part of the predefined criteria. Note its name in the **Refinement Criterion** field or in the console, and review it in the **Boundary Register** or **Yplus/Ystar Register** dialog box to verify that it is appropriate. For example:

   **Solution → Cell Registers → boundary_cell_distance** **Edit...**

   Note that the register dialog boxes provide additional functionality that is not available in the **Adaption Criteria Settings** dialog box:

   - The **Boundary Register** dialog box allows you to switch to the **Normal Distance** or **Volume Distance** method when the **Cell Distance** method is not preferred for your case.

- The **Yplus/Ystar Register** dialog box allows you to compute the minimum and maximum values of $y^+$ or $y^*$, which can help you determine what values to enter for **Min Allowed** and **Max Allowed**.

## 18.2. Predefined Criteria for Aerodynamics with the Pressure-Based Solver

When using the pressure-based solver, you can use predefined criteria to set up aerodynamic simulations in which you want to adapt the mesh to properly account for shocks. With this predefined criteria, Fluent will automatically generate the necessary cell registers for refinement and coarsening, based on a shock wave identification parameter $\hat{s}$:

$$\hat{s}=e^{c_1(s-1)^2} \tag{18.1}$$

where

$$s=\left(\frac{1}{a}\right)\left[\frac{\vec{u}\cdot\nabla p}{\|\nabla p\|+c_2}\right] \tag{18.2}$$

and

$a$ is the speed of sound for the fluid

$\nabla p$ is the gradient of fluid pressure

$\vec{u}$ is the fluid velocity

$c_1$ is the exponential filter constant

$c_2$ is the linear noise reduction constant

To use predefined criteria for adapting a mesh with a shocks, perform the following steps:

1. Select the **Pressure-Based** solver in the **General** task page.

2. Set up your external aerodynamic simulation, and initialize and/or run the calculation to generate data.

3. Enable beta features access (as described in Introduction (p. 1)).

4. Open the **Manual Mesh Adaption** or **Automatic Mesh Adaption** dialog box.

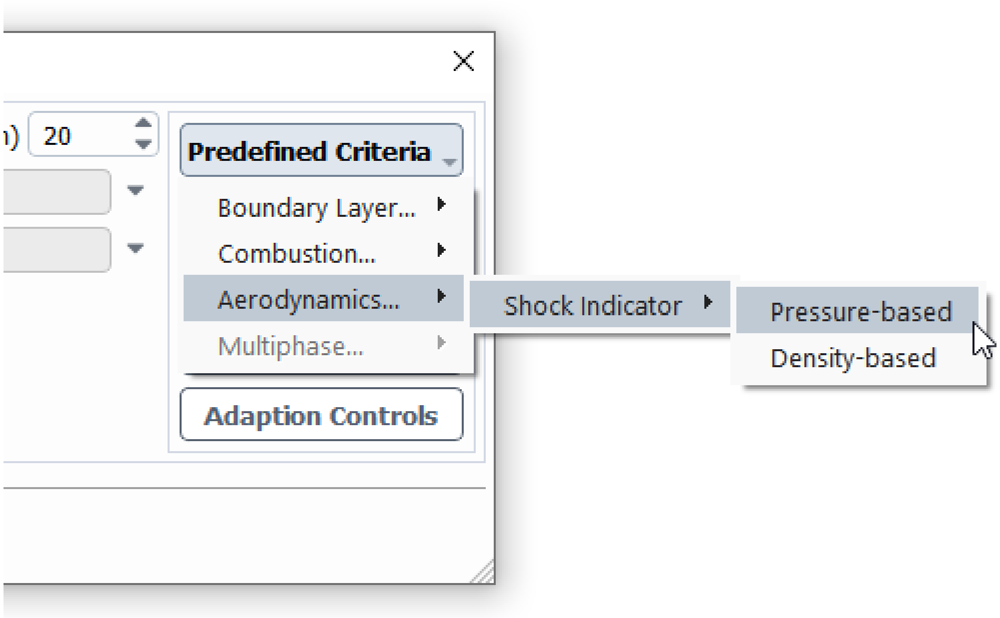
   **Domain → Adapt → Manual...**

   or

   **Domain → Adapt → Automatic...**

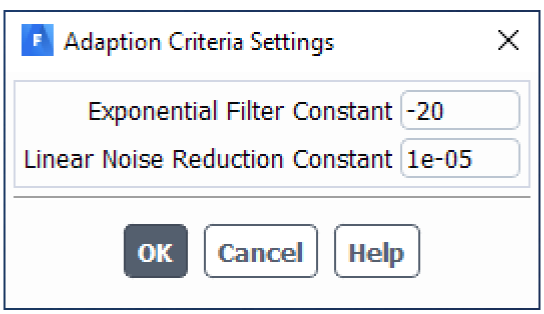
   a. Select **Aerodynamics...** from the **Predefined Criteria** drop-down list, and then select **Shocks / Pressure-based** from the menu that opens.

**Figure 18.5: The Predefined Criteria Drop-Down List**




b. Complete the setup in the **Adaption Criteria Settings** dialog box that opens. **Exponential Filter Constant** corresponds to $c_1$ in Equation 18.1 (p. 104) and can be used to vary the amount of space captured around the shock by the criterion; **Linear Noise Reduction Constant** corresponds to $c_2$ in Equation 18.2 (p. 104) and can be used to limit the capturing to shock areas, while excluding spurious results due to numerical noise.

**Figure 18.6: The Adaption Criteria Settings Dialog Box for the Cell Distance Criterion**

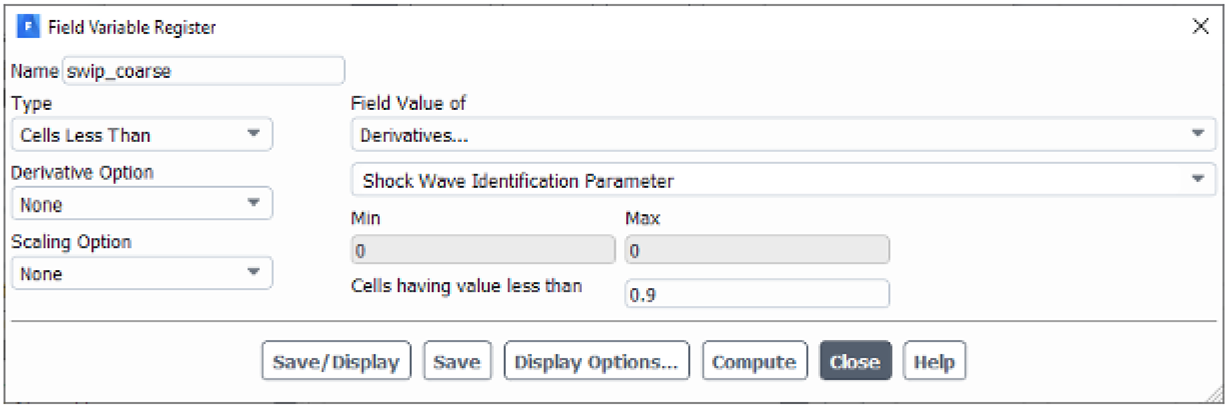



c. For an automatic adaption criteria, by default the cells will be refined at a **Frequency** of `20` iterations or time steps. You can adjust this setting as necessary.

d. Cell registers are created as part of the predefined criteria. Note their names in the **Refinement Criterion** and **Coarsening Criterion** fields or in the console, and review them in the **Field Variable Register** dialog box to verify that they are appropriate. For example:

**Solution → Cell Registers → swip_coarse Edit...**

**Figure 18.7: The Field Variable Register Dialog Box for the Coarsening Criterion**




5. When postprocessing, note that a **Shock Wave Identification Parameter** field variable is available in the **Derivatives...** category, which corresponds to $\hat{s}$ in Equation 18.1 (p. 104).

# Chapter 19: Graphics, Postprocessing, and Reporting

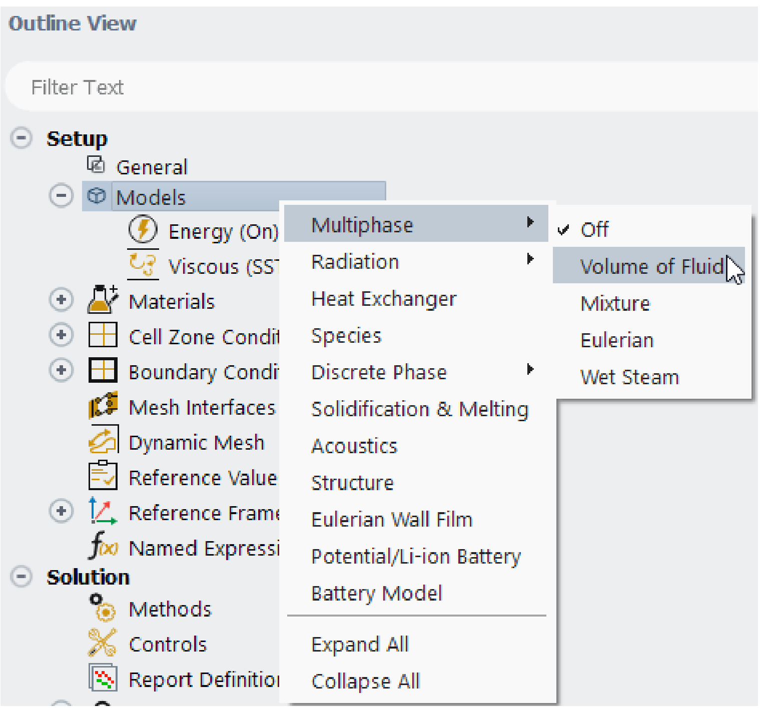
## 19.1. Hide Duplicate Nodes in Mesh Display

With beta features enabled (as described in Introduction (p. 1)), you can disable the display of duplicate mesh nodes by setting the `display/set/duplicate-node-display?` text command to `yes`.

## 19.2. Model Tree Matches Case Settings

With beta features enabled (as described in Introduction (p. 1)), the **Models** branch in the **Outline View** tree will only contain models that are enabled for this case. To enable models using the **Outline View**, right-click the **Models** branch and enable the desired model, as shown in Figure 19.1: Enable the Volume of Fluid Multiphase Model using the Models Branch Context Menu (p. 107).

**Figure 19.1: Enable the Volume of Fluid Multiphase Model using the Models Branch Context Menu**




---

**Note:**

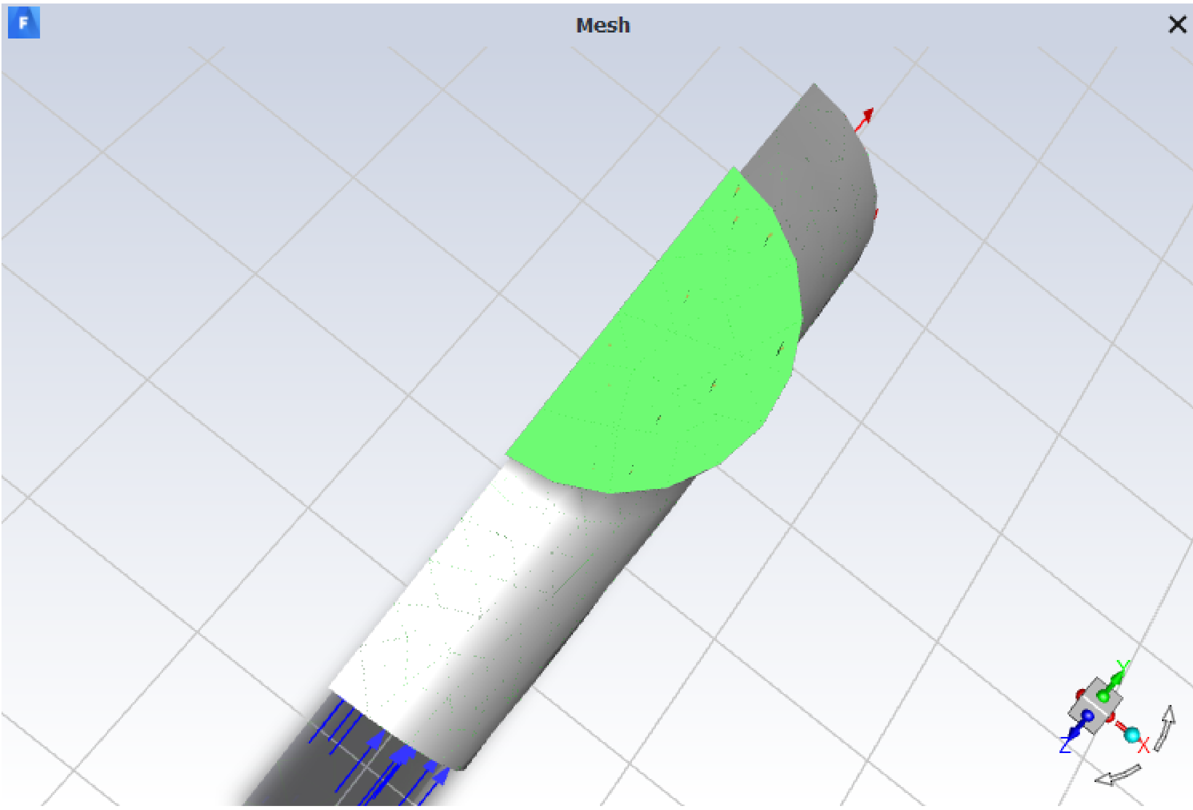
If you enable a model and then disable it, it will remain in the tree.

---

## 19.3. Make the View Normal to the Selected Surface

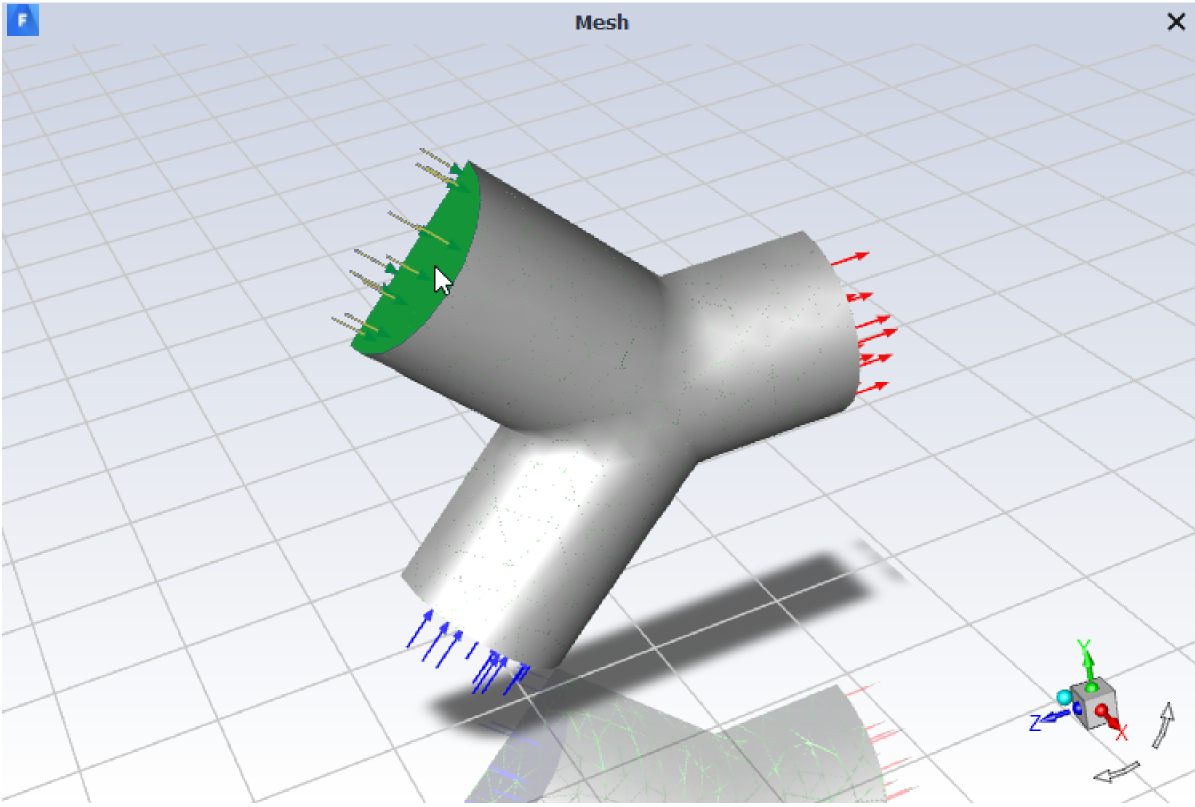
With beta features enabled (as described in Introduction (p. 1)), you can change the graphics window view to make it normal to a selected surface, as shown in Figure 19.2: View Set as Normal to the Selected Surface (green) (p. 108).

**Figure 19.2: View Set as Normal to the Selected Surface (green)**




To make the view normal to the selected surface:

1. Select a surface in the graphics window (left mouse button click with the default mouse mapping).




2. Click **Set Camera Normal to the Selected Surface** () in the toolbar to update the graphics window view.

## 19.4. Force, Drag, Lift, and Moment Report Definitions Using Reference Frames

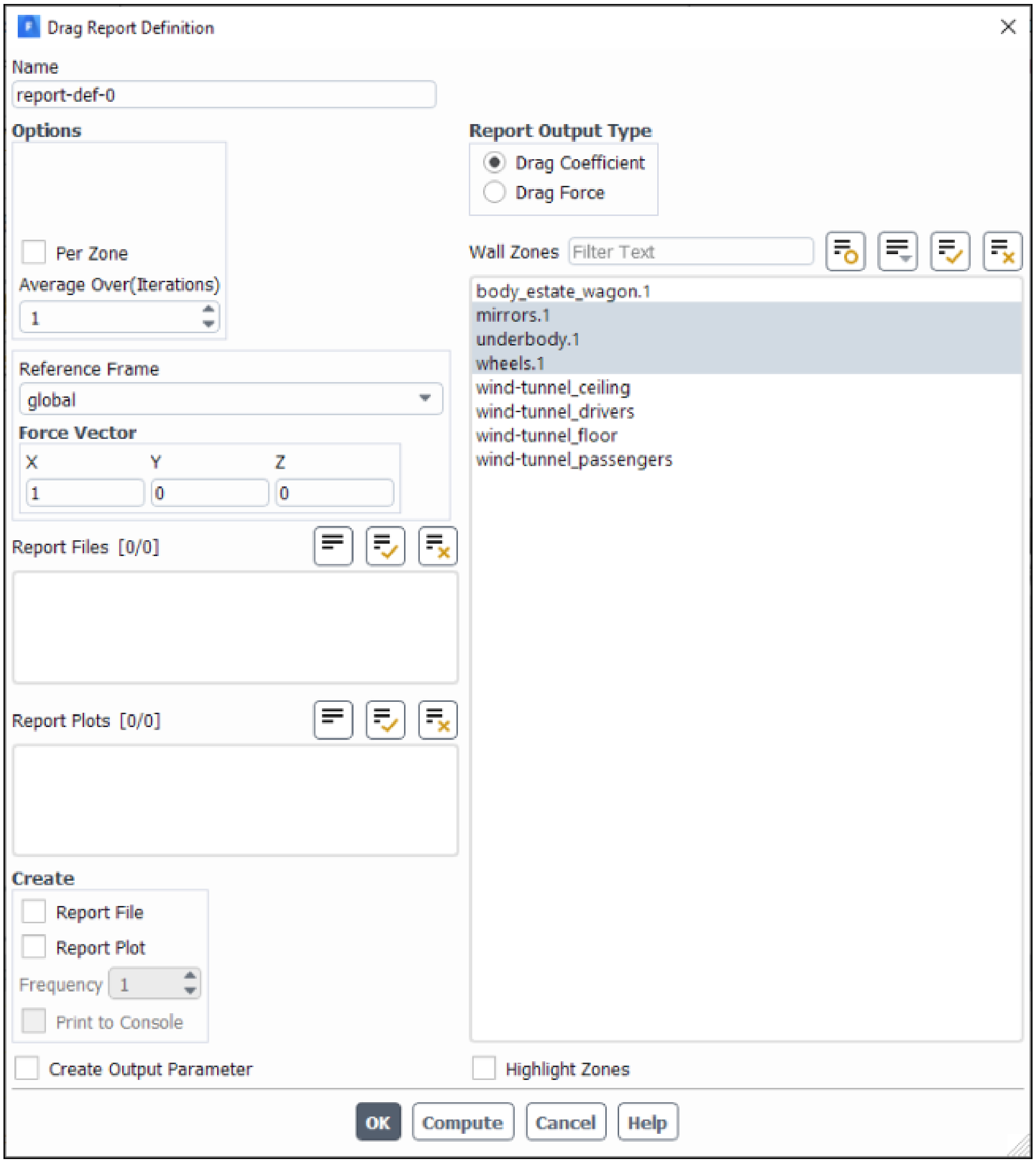
With beta features enabled (as described in Introduction (p. 1)), you can specify the frame of reference for force, drag, lift, and moment report definition computations.

To specify the reference frame for a report definition:

1. Define the reference frame that you want to use for this report, as described in Reference Frames in the *Fluent User's Guide*.

2. Open the appropriate report definition dialog box (**Drag** | **Lift** | **Moment** | **Force**) by right-clicking **Report Definitions** in the tree and selecting **New**>**Force Report**.

**Solution** → **Report Definitions** → **New** → **Force Report** → **Drag...** | **Lift...** | **Moment...** | **Force...**




3. (Optional) Enable **Per Zone** if you want the report definition computed on a per-zone basis.

4. Select the desired reference frame from the **Reference Frame** drop-down list.

5. Provide the components for the **Force Vector** (**Drag**, **Lift**, **Force** only) or the **Moment Center** and **Moment Axis** (**Moment** only).

6. Setup the rest of the report definition as you normally would and click **OK** to create the report definition.

   *Refer to Creating Report Definitions in the Fluent User's Guide for additional information on report definitions.*

## 19.5. Interactive Plots

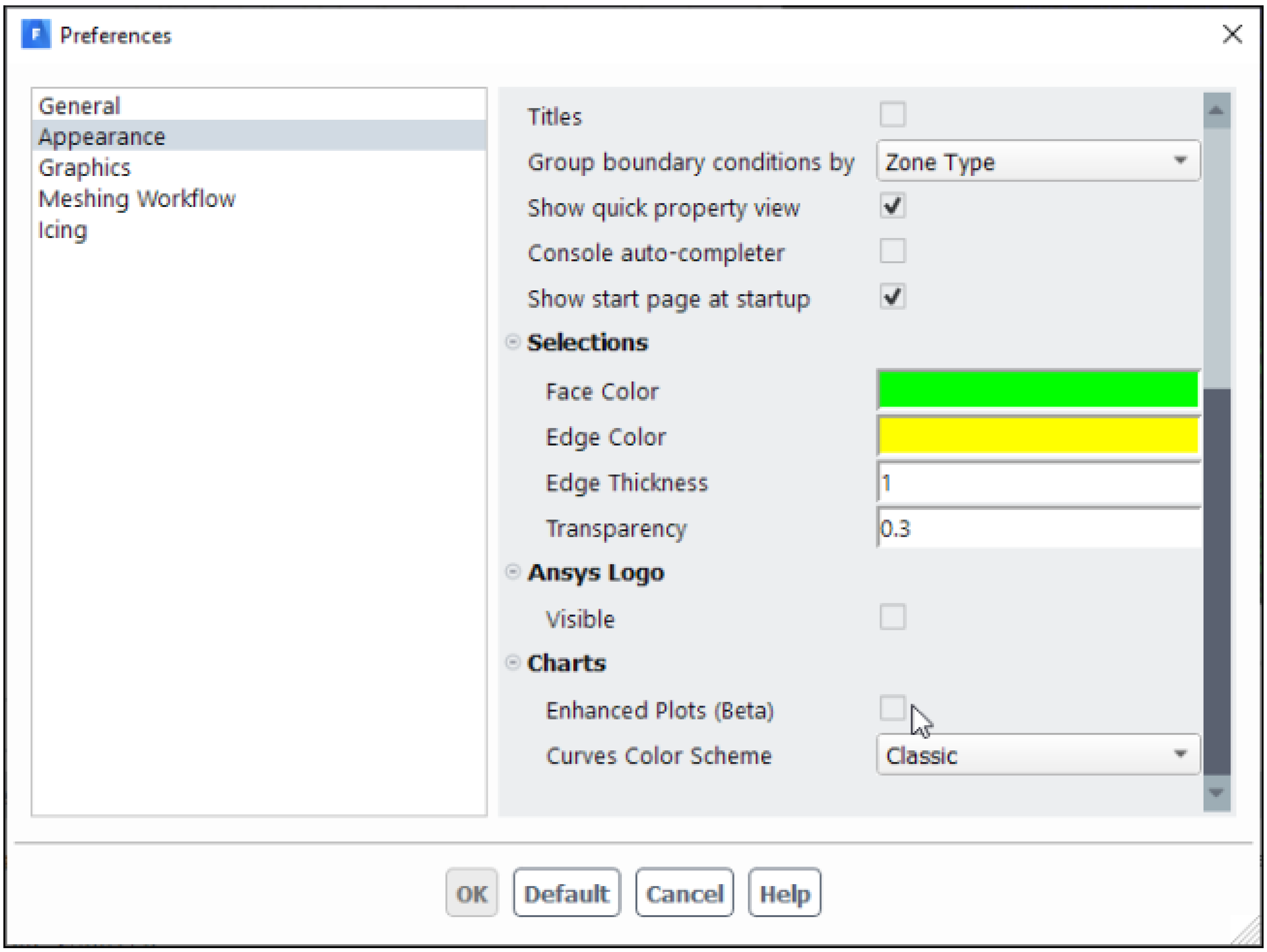
You can enable enhanced interactive plots as a beta feature in preferences. These plots/charts allow you to zoom in/out, enable/disable curves, quickly query curve values at various points, and so on.

To access enhanced plots:

1. Enable **Enhanced Plots (Beta)** in preferences.

   **File → Preferences...**

   **Figure 19.3: Enabling Enhanced Plots**




Enabling enhanced plots provides the following interactions for 2D plots in the graphics window:

- Enable/disable curves by clicking the associated colored square in the legend.
- See values at any location along a plotted curve by hovering over the point of interest.
- Zoom in/out on the plot by using a box zoom with the left mouse button (drag from upper left to lower-right to zoom in and the opposite motion to zoom out).

- Fit the plot to the size of the graphics window by clicking the Fit to Window button ( ) or by **Ctrl + A**.
- Edit the properties of a curve by double-clicking it in the plot window. Here you can edit properties like line thickness, marker style and so on. This can be done while the solver is iterating.
- Resize and scroll through the legend when the number of entries exceeds the size of the plot, allowing you to review all of the entries.
- Edit axis properties by double-clicking the axis.
- Control the color and fonts of the plot title, axis title, axes, and legend.
- Specify a threshold for the total number of plot points, after which, the plot can be reverted to the standard plot behavior to preserve performance.

---

**Note:**

- You can change the legend location using the **Legend Alignment** drop-down in **Preferences**.
- You can control the default colors for new plots via the **Curves Color Scheme** drop-down list. The updated colors will only appear for new plots, so you will have to re-plot any existing plots to see the new color scheme applied.

---

### Limitations

- Annotations are not compatible with enhanced plots.
- Solution animations of 2D enhanced plots are not compatible with the HSF image file format.

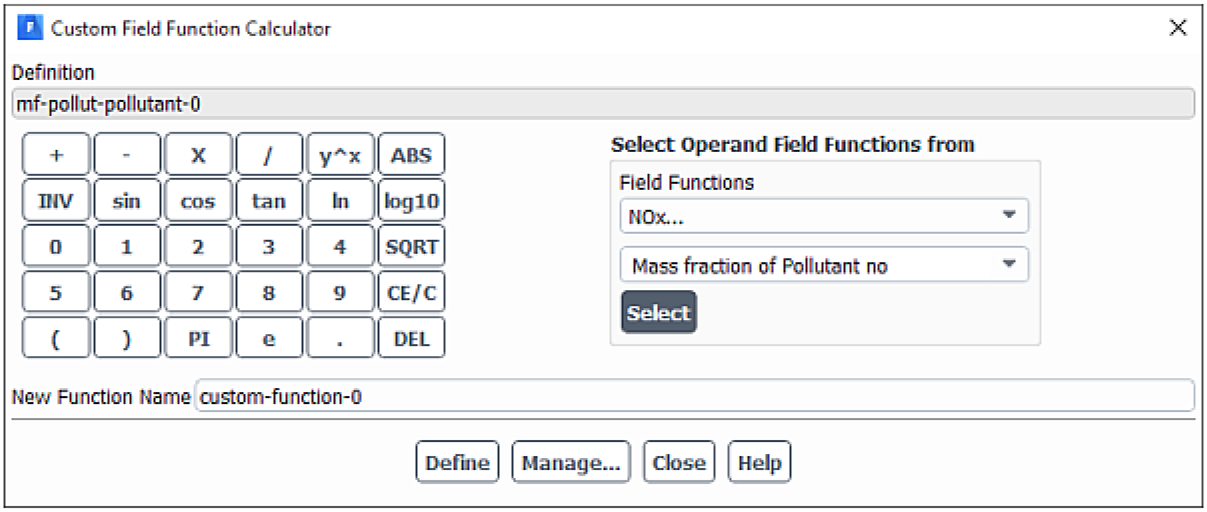
## 19.6. Postprocessing Unsteady Statistics Using Custom Field Functions

You can postprocess unsteady statistics of any variable in your Ansys Fluent simulation, by performing the following steps:

1. Set up your transient problem.
2. Create a custom field function for the each of the variables for which you want to postprocess unsteady statistics, using the **Custom Field Function Calculator** (Figure 19.4: The Custom Field Function Calculator Dialog Box (p. 113)). For detailed instructions, see the chapter on creating custom field functions in the Fluent User's Guide.

   **User-Defined → Field Functions → Custom...**

**Figure 19.4: The Custom Field Function Calculator Dialog Box**




### Important:

The maximum number of custom field functions that can be calculated and postprocessed for unsteady statistics is 50.

3. Enable the beta feature access (as described in Introduction (p. 1)).

4. Enable data sampling for the unsteady calculation.

   **Solution** → **Run Calculation** → **Data Sampling for Time Statistics**

5. Enable unsteady statistics for custom field functions by using the following text command:

   `solve → set → unsteady-statistics-cff`

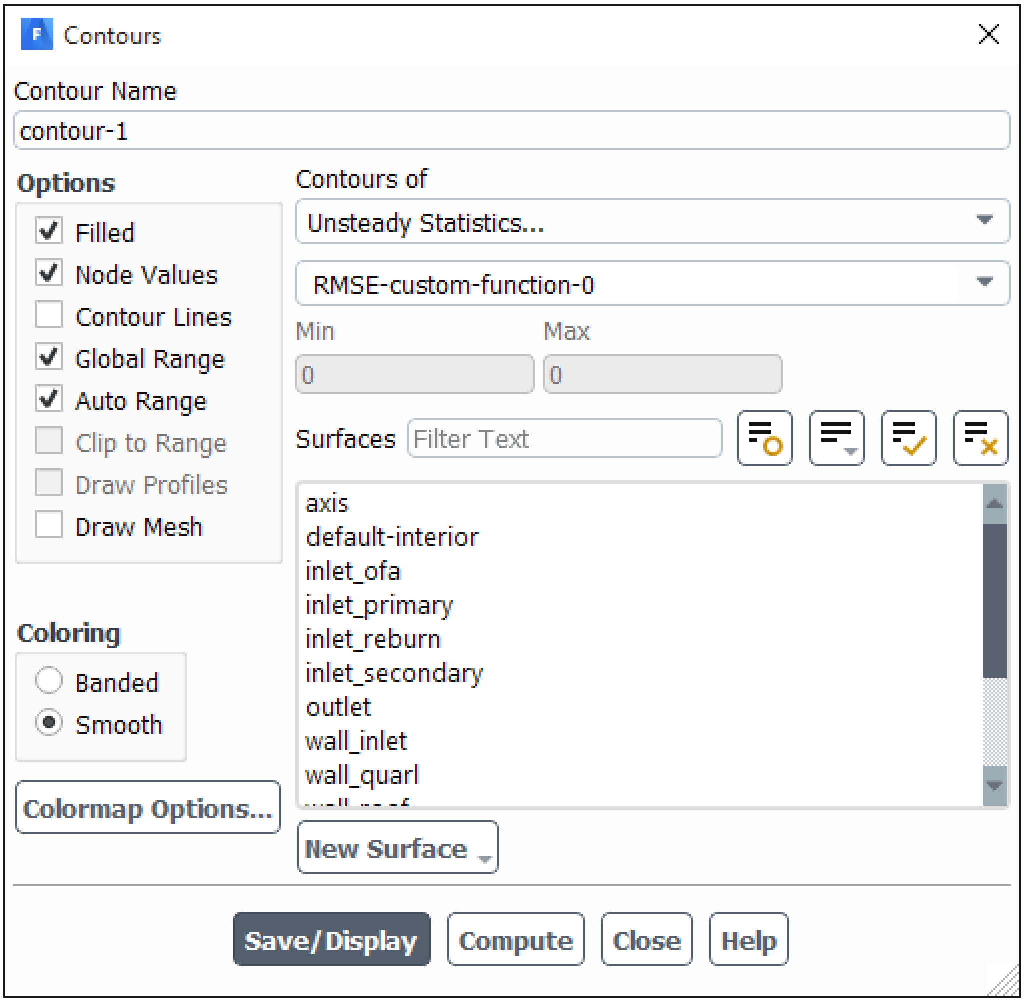
   You will be prompted to enter the frequency at which the unsteady statistics will be sampled, as well as to specify the custom field functions you want to postprocess.

6. Run the calculation.

   **Solution** → **Run Calculation** → **Calculate**

7. When the calculation is complete, the unsteady statistics for your custom field functions will be available for postprocessing as field variables. The root-mean-square of the function will be named **RMS-***function_name*, and the mean value of the function will be named **Mean-***function_name*, where *function_name* is the name of the custom field function you defined in step 2. For example, in the **Contours** dialog box, you could select **Unsteady Statistics...** and **RMS-uns-custom-functon-0** for the **Contours of** drop-down lists (see Figure 19.5: The Contours Dialog Box (p. 114)).

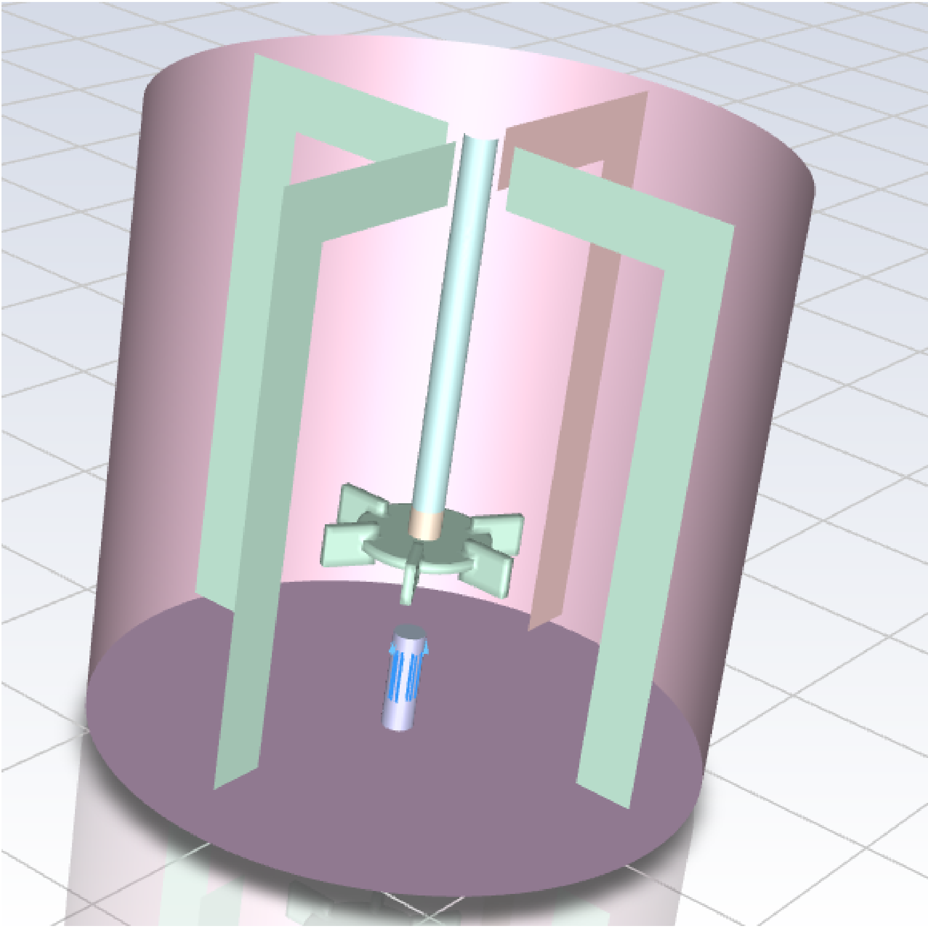
**Figure 19.5: The Contours Dialog Box**




## 19.7. Modern Pastel Colors for Mesh Display

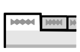
With beta features enabled (as described in Introduction (p. 1)), you can choose to display the mesh using a modern pastel color scheme that is similar to the colors used in Ansys SpaceClaim. Figure 19.6: Pastel Colors (p. 115) shows an example of pastel colors applied to a simple mixing tank.

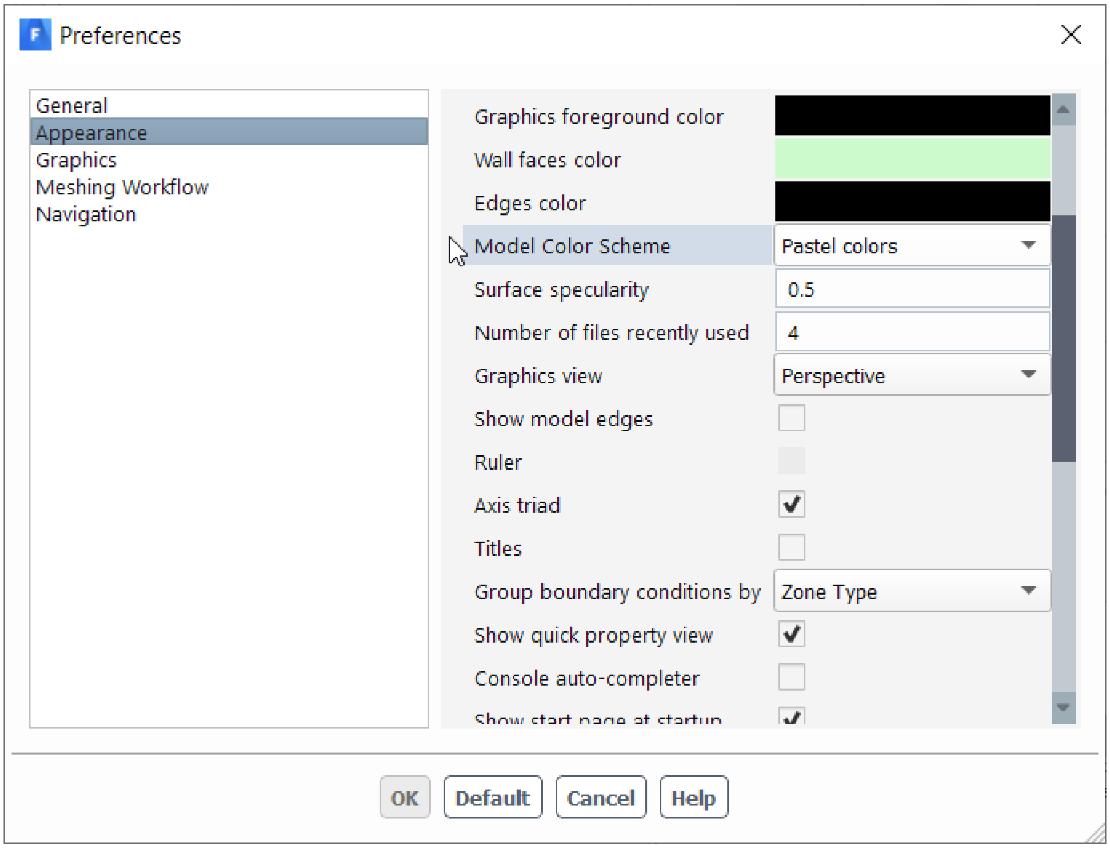
**Figure 19.6: Pastel Colors**



To use modern pastel colors:

1. Open **Preferences**.

    **File** → **Preferences...**




2. Select **Pastel Colors** from the **Model Color Scheme** drop-down list (in the **Appearance** branch).

3. Display your model using the **Mesh Display** dialog box.

 **Domain → Display...**

# Chapter 20: Turbomachinery

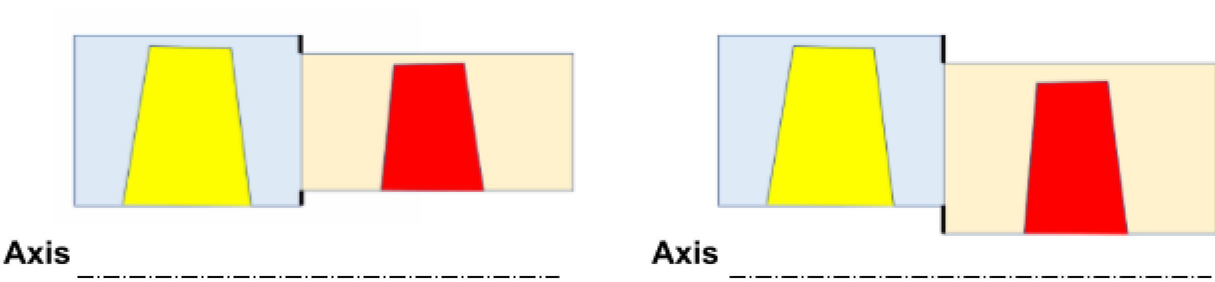
This chapter contains information relating to turbomachinery modeling implemented as beta features in Ansys Fluent 2021 R2.



## 20.1. General Turbo Interface with Lip Feature

In many turbomachines, the hub or the shroud walls from one blade row to another may not be at the same radial height (axial machine) or axial position (in radial machines). This topological discrepancy results in what sometimes is termed a lip feature. The lip feature can be an actual flow blockage such as walls or can be bleed or injection.

**Figure 20.1: Two possible configuration of lip feature formation in an axial turbomachine**




To access this feature, enable beta feature access (Introduction (p. 1)).

To model this feature (if it exists in the geometry) you must select the **Non-Overlapping Wall** option when creating a General Turbo Interface (GTI). For a **Pitch-Scale** or **No Pitch-Scale** interface, this is all that is needed to model a frozen rotor or Transient Rotor/Stator case with the lip feature.

The non-overlapping part of the interface forms walls by default. These walls are listed with wall boundary conditions. If the lip feature happens to represent a region of injection or bleed then you can change the boundary to an inflow or outflow boundary.

If the non-overlapping wall option is to be used with the mixing-plane model then ensure that the default intersector-base formulation is used. The formulation is set using the TUI command: `/define/turb-model/general-turbo-interfaces-settings/mixing-plane-model-settings/formulation`.

# Chapter 21: Parallel Processing

This chapter contains information related to parallel processing beta features in Ansys Fluent 2021 R2.



## 21.1. Multidomain Architecture for Conjugate Heat Transfer

For conjugate heat transfer (CHT) problems, particularly transient simulations and/or those with combustion, the dominant time-scales in the fluid and solid zones are often quite different. In most transient cases, it is desirable to use a larger time step / run less iterations for solid zones compared to the fluid zones, in order to increase the speed at which the solid heat transfer reaches steady-state without compromising the solution accuracy of the fluid flow. This can be done by using multidomain architecture: it allows you to easily separate the fluid and solid calculations and run them in unique Fluent sessions (which may each use specified numbers of processes and may be run on separate machines), so that different time scales can be used for each, and only the energy equations will be solved for the solid zones. The temperature and heat flux data is exchanged between the sessions throughout the simulations. Multidomain architecture for CHT is an alternative to specifying a solid time step size for transient flows or specifying fluid / solid time scales for pseudo-transient flows, and by the nature of the methodology and the control it affords you, it should result in faster calculations than these other methods.

---

**Note:**

The ability to use multidomain architecture with unique Fluent sessions for fluid and solid calculations is supported for Linux only.

---

To use the multidomain architecture for CHT with multiple sessions, perform the following steps:

1. Launch Fluent, using a number of processes equal to the total amount you would like for the fluid and solid calculations.

   When submitting multidomain jobs to a queue, it may be helpful to include the `-mda=<x>` command line option, where <x> is the number of processes used for the "helper" session that will handle the solid calculations. This allows the queue resources to be allocated appropriately. For example, if you enter `fluent 3d -t5 -mda=2`, then 2 processes will be used for the solid calculations and the remaining processes (5-2=3) will be used for session that opens and handles the fluid calculations. Note that the total core count (`-t<n>`) must be supported by your license, and must be greater than the number specified for the helper session. The resources will be configured between the server and helper session even if you include the `-cnf=<h>` command line option, if it is required to run the calculations on specific machines.

If you are using Fluent Launcher or later decide to change processor allocation, note that you can edit the number of processes used for the helper session from within Fluent using a text command, as described in a later step.

2. Set up a case file with fluid and solid cell zones.

**Note:**

The solid and fluid zones can have a non-conformal interface, but they must not move relative to each other (as in a sliding mesh).

3. Enable beta features access (as described in Introduction (p. 1)).

4. Enable loosely coupled conjugate heat transfer by using the following text command:

   `parallel` → `multidomain` → `conjugate-heat-transfer` → `enable?`

5. Enable the use of a secondary (helper) session for conjugate heat transfer co-simulation by using the following text command:

   `parallel` → `multidomain` → `conjugate-heat-transfer` → `set` → `session-mode`

6. The fluid calculations will be handled by the Fluent session that is currently open, while the solid calculations will be handled by the helper session. Set up the coupling between these sessions by using the following text command:

   `parallel` → `multidomain` → `conjugate-heat-transfer` → `set` → `coupling`

   For steady-state simulations, you need only specify how many fluid iterations you want to run for each solid iteration. For transient simulations, you must specify the coupling method:

   - Enter `0` to use the same time step size for fluids and solids, and then enter the number of time steps per coupling between fluids and solids
   - Enter `1` to use smaller time steps for fluids (and couple at every solid time step), and then enter the solid-to-fluid time step size ratio; a value of 10 or higher is recommended for the latter.

7. If you did not launch Fluent with the `-mda=<x>` command line option, set up the helper session that will handle the solid calculations by using the following text command:

   `parallel` → `multidomain` → `conjugate-heat-transfer` → `set` → `helper-session`

   You will be prompted to enter the number of processes to be used by the helper session; this will be in addition to the number of processes used in the current session, and by default (when the `-mda=<x>` command line option was not used) it will be set to 4, so you must ensure that the total core count is supported by your license. You will also be given the option to run it on specific machine(s): enter `""` to use the current machine; or list (in quotes) the machines / hosts file, similar to how this is done when using the `-cnf=<h>` command line option (for example: `"machine1,machine1,machine2"`).

8. Run the calculations by using one of the following text commands:

- For steady-state simulations:

  `parallel` → `multidomain` → `solve` → `iterate`

  You will be prompted to enter the number of iterations.

- For transient simulations:

  `parallel` → `multidomain` → `solve` → `dual-time-iterate`

  You will be prompted to enter the number of fluid time steps, along with the number of iterations per time step.

Your settings may not be followed exactly; for example, if you enter a number of fluid time steps that is not a multiple of the solid-to-fluid time step size ratio, the number of time steps will be increased to an appropriate value to ensure the proper data exchange between the sessions.

Note that the helper session will run in the background, and will automatically close when the calculations are complete and all of the data has been exchanged. A transcript named `_temp_helper.trn` will be saved in your working directory for this helper session, along with case and data files for the initial state of the helper session and a journal file for the actions taken on these files (all of which begin with the prefix `mda_`).

## 21.2. HPC-X Message Passing Interface

The use of the HPC-X message passing interface (MPI) for parallel simulations on Linux is supported as a beta feature with Mellanox OpenFabrics Enterprise Distribution (MLNX_OFED) versions 4.3-1 to 4.4-2. To use this MPI, include the following command line option when launching Fluent: `-mpi=hpcx`.

## 21.3. Intel 2019 Message Passing Interface

The use of the Intel 2019 update 8 message passing interface (MPI) for parallel simulations is supported as a beta feature for the following:

- Linux with shared memory on a local machine or with distributed memory on a cluster
- Windows with distributed memory on a cluster (note that Intel 2019 update 8 is already supported as a full feature with shared memory on a local machine)

To use this MPI, include the following command line option when launching Fluent: `-mpi=intel2019`.

This MPI can resolve the following issue for the default MPI (Intel MPI) that was noted in the Ansys, Inc. Known Issues and Limitations for Release 2020 R2: on Windows or mixed Windows / Linux, the Intel 2019 MPI allows you to dynamically spawn additional processes when switching from meshing mode to solution mode if you specified the use of fewer meshing processes than solver processes (for example, `fluent 3d -meshing -tm3 -t5 -mpi=intel2019`). Note that this MPI does not allow you to perform such dynamic spawning using the `/parallel/spawn-solver-process` text command in meshing mode. (236195 / 257722)

---

**Note:**

To use the Intel 2019 MPI when running Fluent on Windows with distributed memory on a cluster, you must have the MPI installed on all of the relevant machines. You can download the MPI installer from the following website and install it with the default options: https://release211.s3.amazonaws.com/w_mpi_p_2019.8.254.exe?AWSAccessKeyId=AKIAJ6ZOVHCFGGNWYUKA&Expires=1616344322&Signature=F0LhimZvgwe5xxuUQwPBXzfPQi4%3D

---

# Chapter 22: Adjoint Solver

This chapter contains information about the beta features relating to the adjoint solver that are available in Ansys Fluent 2021 R2.



## 22.1. The Algebraic Transition Model with the Adjoint Solver

The generalized $k$-$\omega$ (GEKO) turbulence model with the Algebraic Transition Model is supported as a beta feature by the adjoint solver when the adjoint turbulence model is enabled and the algebraic transition model is enabled in the **Viscous Model** dialog box. For such cases, the following field variables are available for postprocessing (in the **Sensitivities...** category):

- **Sensitivity to Algebraic Intermittency**

  This field variable is the sensitivity of the observable with respect to the intermittency $\gamma$.

- **Sensitivity to Critical Reynolds Number**

  This field variable is the sensitivity of the observable with respect to the critical Reynolds number $\mathrm{Re}_{\theta C}$.

## 22.2. Periodic Morphing

Periodic morphing in the theta direction for cylindrical region conditions is supported as a beta feature by the adjoint solver. Periodic morphing in the theta direction is made available in the **Region Conditions** tab of the **Design Tool** only after the following settings have been applied:

- **Cylindrical** is selected from the **Region Geometry** drop-down list in the **Region** tab
- **Direct Interpolation** or **Radial Basis Function** is selected in the **Morphing Method** group box of the **Design Change** tab

**Figure 22.1: Design Tool Dialog Box with Periodic Morphing Option**


Morphing is enforced as periodic, repeated by the desired number of repeats in the morphing region, by entering the desired number of repeats under **Periodic** in the **In Theta** group box. It is recommended that the morphing region be `360` degrees in the theta direction so that no continuity condition needs to be enforced. Note that periodic morphing is distinct from periodic boundary conditions and periodic repeats. For periodic morphing, when the desired number of repeats is `0` or `1`, no repeats will be applied.

# Chapter 23: Fluent in Workbench

This chapter contains information relating to using beta features in Ansys Fluent 2021 R2 in Workbench.



**Note:**

The interactions between Ansys Workbench and Ansys Fluent for beta features are as follows:

- Enabling beta features in Ansys Workbench automatically enables beta features in Ansys Fluent systems that have an associated mesh or case file as long as there is not a session of Ansys Fluent already open or as long as a settings (`.set`) file has not yet been created (see the section on saving work in Fluent in the Fluent in Workbench User's Guide.
- The Ansys Fluent **Setup** cell must be reset if the beta features option is enabled in Ansys Workbench after an Ansys Fluent session is already open or after a settings (`.set`) file has been created.
- You can override the automatic enabling of beta features in Ansys Fluent with the text command `define> beta-feature-access` and entering no when prompted.
- Disabling beta features in Ansys Workbench does not automatically disable beta features in Ansys Fluent.

## 23.1. Performing Coupled Simulations with Fluent and Electronics Desktop Applications

If **Beta Options** are enabled in Workbench, additional options for coupling simulations are possible between Fluent and Ansys Electronics Desktop analysis systems. Note that once **Beta Options** are enabled in Workbench, you may also have to restart the Fluent application and/or Workbench to proceed with the simulation.

The following table illustrates the current support scenarios for Fluent and Ansys Maxwell coupling.

**Table 23.1: Current Supported Scenarios for Fluent and Maxwell Coupling**

| | Maxwell | Fluent | One-way Support? | Two-way Support? |
|---|---|---|---|---|
| 1 | Steady | Steady | Yes | Yes |
| 2 | Steady | Transient | Yes | Yes |

| | | | | |
|---|---|---|---|---|
| 3 | Transient | Steady | Yes | Yes |
| 4 | Transient | Transient | BETA | BETA |

Note that if **Beta Options** are disabled in Workbench, then Fluent will not generate the feedback temperature file for Maxwell.

When coupling Fluent with Ansys HFSS or Ansys Q3D Extractor, all of the scenarios are BETA.

## 23.2. Working with Custom Input Parameters

Various Ansys Fluent setup related input quantities can be assigned to an input parameter. You can define a series of simulations based on a set of parametric values that are managed both in Ansys Fluent and in Workbench. These parameters may be defined for numeric cell zone and boundary condition settings using the **New Input Parameter ...** option in the corresponding drop-down list or by a small “p” icon adjacent to a specific input setting. However, various Ansys Fluent settings are not supported by these methods.

If the `beta-feature-access` option is enabled in Ansys Fluent (as described in Introduction (p. 1)), you can mitigate this limitation using custom input parameters, and define input parameters for various Ansys Fluent simulation related settings. The `define/parameters/custom-input-parameters/create` command is used to define custom input parameters that will use other text user interface (TUI) commands in Fluent to change the desired simulation settings in a parametric manner. Each numerical component of the TUI command string can be marked and treated as a parameter. Setting up custom input parameters requires using Scheme functions that convert TUI commands into Scheme variables. Once defined, the Scheme variables are set as custom input parameters and are displayed in the **Parameters** dialog box alongside other input parameters. The input parameter passes a constant numeric value to the registered scheme function. Therefore, the associated Scheme function (and corresponding Fluent text command) uses the constant parameter values using the units that were already defined for the designated text command quantity.

**`/custom-input-parameters/`**

Enter the custom input parameters menu.

**`create/`**

Create a custom input parameter. The following example demonstrates the `create` command. where you create a custom input parameter using a Scheme file called `my-funct`:

```
/define/parameters/custom-input-parameters> create
Name of parameter ["parameter-1"]
parameter-1 value [0] 0.3
Enter the name of custom-input-var variable as symbol [custom-input-var1]
Enter the name of apply-function [()] my-funct
/solve/set/under-relaxation/pressure 0.3
```

where the `my-funct` Scheme file contents are:

```
(define my-funct
    (lambda (value )
        (ti-menu-load-string (format #f "/solve/set/under-relaxation/pressure ~g" value))))
```

**delete**

Delete a selected custom input variable, but not the associated input parameter (the input parameter has to be deleted separately). Using the wildcard '*' allows you to delete all custom input variable at once. For example:

```
/define/parameters/custom-input-parameters> delete
(custom-input-var3)
custom-input-var name(1) [custom-input-var3] *
custom-input-var name(2) [()]
Are you sure you want to delete input parameter ("custom-input-var1" "custom-input-var2" "custom-input-var3
```

**list**

Shows a list of defined custom input parameters along with their associated variables and apply functions. For example:

```
   parameter-name           custom-input-var                      apply-function
--------------------     ------------------------------        --------------------
   parameter-3              custom-input-var3                        my-funct
   parameter-2              custom-input-var2                        my-funct
   parameter-1              custom-input-var1                        my-funct
--------------------     ------------------------------        --------------------
```

## 23.3. Using UDFs to Compute Output Parameters

Ansys Fluent allows you to create output parameters that let you compare reporting values for different cases. Output parameters are typically defined and computed through the graphical user interface and not accessible through user defined functions (UDFs).

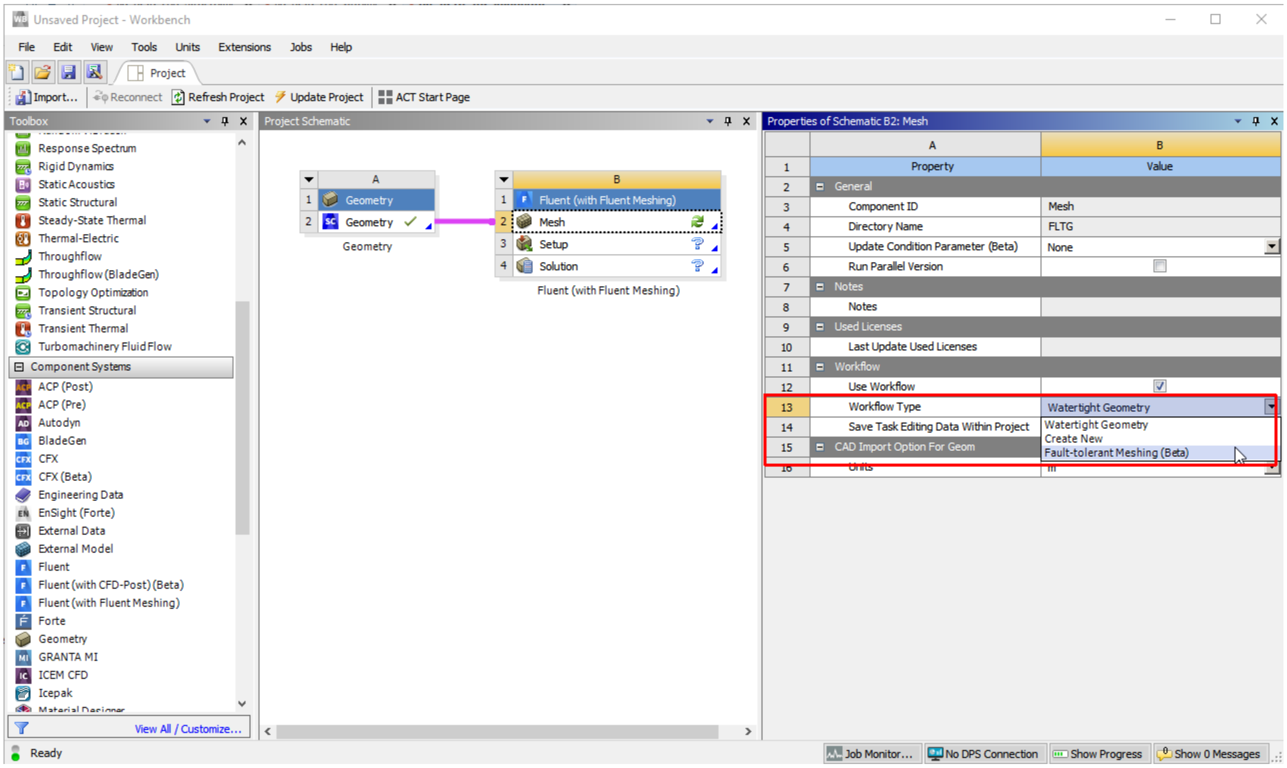
If the `beta-feature-access` option is enabled in Ansys Fluent (as described in Introduction (p. 1)), you can compute and publish real output parameter values computed by UDFs to Workbench (or Ansys Fluent). When the beta feature is enabled, a new output parameter type called `udf` is available once you invoke the `create` command in the `define/parameters/output-parameters` menu. The `create` command has been extended to allow you to create a UDF-based output parameter. Using a registered UDF, you can compute your own output quantity. Currently, this command is called by Fluent in Workbench at the end of a calculation. When you select the `udf` type, you are prompted for information regarding the name of the output parameter, the user-defined function, the name of any input parameters, etc. When you are finished, you are left with a specific output parameter that uses a UDF that in turn may consume one or more input parameters. For example:

```
define/parameters/output-parameters> create

Output Parameter Type>
drag-coefficient   lift-coefficient          udf
flux               moment-coefficient      volume-integral
force              surface-integral

Name of Output Parameter ["parameter-1"]
Available udf of type output-parameter: ("mylibudf::libudf")
output-parameter UDF function name ["mylibudf::libudf"]
Do you want to use Input Parameters in the UDF Output Parameter? [no] yes

Enter the no. of Input Parameters to be used in UDF [0] 2
Name of Input Parameter ["parameter-3"]
parameter-3 value [0] 2
Name of Input Parameter ["parameter-4"]
parameter-4 value [0] 3
```

To see the value of a particular output parameter, run the calculation for a few iterations (or initialize the solution), then type the `define/parameters/output-parameters/print-to-console` text command.

## 23.4. Fault-tolerant Meshing Workflow

When beta features are enabled, you can have access to the **Fault-tolerant Meshing** guided workflow as a **Workflow Type** from the **Properties** window for the **Meshing** cell in the **Fluent (with Fluent Meshing)** component system.



Note that when the **Create New** option is selected, the **Watertight Geometry** workflow is assumed and an **Import Geometry** task is added to the new workflow. Alternatively, if you want to start with the **Fault-tolerant Meshing** workflow, you can delete this task and, using the **Insert New** menu option, you can add your own **Import CAD and Part Management** task. The **CAD File** and **Display Units** properties for this task will be directly valuated based on the **Geometry** path and **Units** provided by the **Geometry** component in the Workbench.

# Chapter 24: User-Defined Functions

This chapter contains information about the beta features related to the following topics:



## 24.1. Six-DOF Motion Constraint Using UDFs

For each moving object you have an associated 6DOF properties UDF in which you set e.g. the mass and the moments of inertia:

```
 prop[SDOF_MASS] = 907.185;
prop[SDOF_IXX] = 27.116;
prop[SDOF_IYY] = 488.094;
prop[SDOF_IZZ] = 488.094;
```

and other 6DOF properties. The entries of the 6DOF properties array are explained in the section on `DEFINE_SDOF_PROPERTIES` in the Fluent Customization Manual.

To constrain the motion of a 6DOF object, i.e. to set the motion to 0, you simply set in the UDF the corresponding 6DOF property array entries to TRUE, e.g.

```
 prop[SDOF_ZERO_TRANS_X] = TRUE;
prop[SDOF_ZERO_ROT_Y] = TRUE;
```

where SDOF_ZERO_TRANS_X, SDOF_ZERO_TRANS_Y, and SDOF_ZERO_TRANS_Z are the components of the translation and SDOF_ZERO_ROT_Y, SDOF_ZERO_ROT_Y, and SDOF_ZERO_ROT_Z are the components of the rotation.

By default these entries are FALSE, i.e. the motion is not constrained.

# Chapter 25: Fluent as a Server

This chapter describes certain capabilities of the Fluent as a Server feature that are released as beta features.

## 25.1. Ansys Session Manager

Ansys Session Manager provides a flexible approach to starting and managing multiple Fluent as a Server sessions. It provides an abstraction layer through which the FLUENT Remote Console as well as client applications using the Fluent as a Server SDK can start, stop, and access Fluent solver sessions. Ansys Session Manager is typically run on a Server machine on which the Fluent as a Server sessions will be run. Once running, Ansys Session Manager listens for connections from client applications and performs the requested session management actions.

When Ansys Session Manager receives a request for a new solver session to be started, it creates a folder for the new solver session in the `TEMP` folder belonging to the user running Ansys Session Manager. It then starts the solver application in server mode and makes the connection information available to the client.

### 25.1.1. Using Ansys Session Manager

The Ansys Session Manager utility is provided as an executable and is located in the following directory within the Ansys FLUENT install tree:

```
v150\fluent\fluent15.0.0\addons\corba\<arch>
```

---

**Note:**

If you are using the Fluent Client Package instead of a full Fluent installation, the Ansys Session Manager files are located in a `Beta` subdirectory of the path above.

---

Usage of the Ansys Session Manager utility is as follows:

**Windows**

```
C:\> AnsysSessionManager.exe -f<configfile> -p<portnumber>
```

**Linux**

```
> AnsysSessionManager.cmd -f<configfile> -p<portnumber>
```

The following arguments and environment variables are recognized:

**<configfile>**

the name of a configuration file containing available installed solvers. For more information on the configuration file, see Configuring Ansys Session Manager (p. 132).

**<portnumber>**

the port on which Ansys Session Manager should listen for client connections.

**AAS_HOST**

an environment variable that is used to specify the IP address on which to listen for connections. Default value is `localhost` which will allow connections only from the local machine. To enable connections from remote clients, set this to the IP address of a network adapter on the local machine that is accessible remotely.

---

**Important:**

When started, Ansys Session Manager will display a message indicating two consecutive port numbers. The first of these is the port number used for connection from a client application.

---

### 25.1.2. Configuring Ansys Session Manager

Before using Ansys Session Manager you must create a configuration file. This file specifies which Ansys solvers are installed on the Server machine and how to instantiate them.

The configuration file is a text file of the following form where **`<userinput>`** indicates an item to be specified according to your specific installation:

```
__AnsysSessionManagerIni__
_NrAvailableApplications
<integer>
_Application_001
<solver 1 name>
<solver 1 executable> <arguments> -aas
CORBA ICoFluentUnit
_Application_002
<solver 2 name>
<solver 2 executable> <arguments> -aas
CORBA ICoFluentUnit
<etc...>
```

An example configuration file is included with the Ansys Fluent installation in the following location:

`%AWP_ROOT150%\fluent\fluent15.0.0\addons\corba\<arch>\AnsysSessionManager.ini`

You may need to edit or amend this file to reflect the local installation details on your Server Machine.

## 25.2. Fluent Remote Console

### 25.2.1. Connecting to Ansys Session Manager

To expose additional commands to connect to the Ansys Session Manager from Fluent Remote Console issue the following Fluent Remote Console command:

```
>beta.enable asm
```

The Ansys Session Manager commands are used to start, connect to, and disconnect from FLUENT solvers through an Ansys Session Manager instance running on a local or remote machine. Ansys Session Manager commands begin with the prefix `asm`.

**asm.connect_to_server <hostname> <portnumber>**

connect to the Ansys Session Manager listening on **`<hostname>`**:**`<portnumber>`**. The portnumber is that specified with the `-p` command line option to Ansys Session Manager.

**asm.list_applications**

list the available applications that can be started by the connected Ansys Session Manager.

**asm.start_application <appname>**

start a session of the application **`<appname>`** and connect to it.

**asm.list_sessions**

list the runnings sessions managed by the conntected Ansys Session Manager that are available for connection.

**asm.connect_to_session <session_name>**

connect to the session **`<session_name>`** that is being managed by the connected Ansys Session Manager.

Ansys Session Manager takes care of brokering the connection from the Fluent Remote Console to the Fluent as a Server session. However, once the client is connected to a solver session, it provides commands directly to the Component Session as described in the Fluent as a Server User's Guide without using the Session Manager as an intermediary.

## 25.3. Fluent as a Server SDK

---

**Note:**

If you are using the Fluent Client Package instead of a full Fluent installation, the SDK files described in the following sections are located in a `Beta` subdirectory of the paths below.

---

### 25.3.1. IAnsysSessionManager CORBA Interface

The IAnsysSessionManager interface is a CORBA interface that includes methods to communicate with a running Ansys Session Manager instance to query, start, and connect to solver sessions. In order to use the CORBA interface you must compile the `AnsysSessionManager.idl` file with a CORBA compiler suitable for your client development environment. This file is located in the following directory within the Fluent install tree:

`v212\fluent\fluent21.2.0\addons\corba\`**`<ARCH>`**`\AnsysSessionManager.idl`

**IAnsysSessionManager**

**long `getNrAvailableApplications`();**

returns the number of applications that Ansys Session Manager has registered.

**string `getApplicationNameByIndex`(in long *`p_iApplicationIndex`*);**

returns the name of the application with index *`p_iApplicationIndex`*.

**string `getApplicationInterfaceInfo`(in string *`p_stringApplicationName`*);**

returns information about the interface of the named application.

**CoArrayString `getApplicationAttributeNames`(in string *`p_stringApplicationName`*);**

returns the attribute names of the named application.

**CoArrayString `getApplicationAttributeDefaultValues`(in string *`p_stringApplicationName`*);**

returns the default values of the attributes of the named application.

**string `startSession`(in string *`p_stringApplicationName`*);**

starts a session of the application *`p_bstrApplicationName`*.

**string `startSessionWithAttributes`(in string *`p_stringApplicationName`*, in CoArrayString *`p_astringAttributeNames`*, in CoArrayString *`p_astringAttributeValues`*);**

starts a session of the application *`p_bstrApplicationName`*, with the specified attribute values.

**long `cleanUp`();**

attempt to connect to all sessions started by the current session manager. If a session fails to connect it will be removed from the list of running sessions.

**Important:**

If a Fluent session is active but slow to respond (due to a long process of reading a case, for example) CleanUp will wait until the session responds. Thus, depending on the status of the running sessions this call may take a long time to complete. Use with caution.

**long `getNrRunningSessions`();**

returns the number of running solver sessions being managed by Ansys Session Manager.

**string `getRunningSessionInterfaceInfo`(in string *`p_stringRunningSessionName`*);**

returns information about the interface of the named solver session.

**string `connectToRunningSession`(in string *`p_stringRunningSessionName`*);**

connect to the managed session named *`p_stringRunningSessionName`*.

**string `resuscitateSession`(in string *`p_stringApplicationName`*, in string *`p_stringLocation`*);**

similar to startSession, but instead of creating a new working folder, resuscitateSession will start the application in the remote folder designated by *`p_stringLocation`*.

**Note:**

*`p_stringLocation`* should be either the absolute path of an existing folder on the remote machine or a path relative to the folder in which Ansys Session Manager is running.

## 25.3.2. COM Connectors

The CORBA interfaces described in Fluent as a Server User's Guide are also available as pre-compiled COM connectors in DLL libraries included with the Fluent as a Server SDK.

### 25.3.2.1. Interfaces

You can make use of COM implementations of the ICoFluentUnit and ICoFluentSchemeController interfaces by including the following library in your application:

```
...\v150\fluent\fluent21.2.0\addons\corba\%ARCH%\COMCoFluentUnit.dll
```

This will make the following classes available:

**class `CCoFluentUnit`**

**`Calculate`(void);**

iterates the solution for the number of iterations specified with `put_NrIterations`. This is mainly for use with steady simulations as it does not perform dual-time iteration. For transient cases you can issue the solve/dual-time-iterate TUI command using `CCoSchemeController::DoMenuCommand`

**`get_ComponentName`(void);**

returns the name of the connected component

**`get_ComponentDescription`(void);**

returns the description of the connected component

**`get_NrInputParameters`(void);**

returns the number of input parameters defined in the current case

**`get_NrIterations`(void);**

returns the number of iterations currently set for a `Calculate` command to perform

**`get_NrOutputParameters`(void);**

returns the number of output parameters defined in the current case

**`getInputParameterNameByIndex`(VARIANT *`&p_variantInputParameterIndex`*);**

returns the name of the input parameter with index *`p_variantInputParameterIndex`*

**`getOutputParameterNameByIndex`(VARIANT *`&p_variantOutputParameterIndex`*);**

returns the name of the output parameter with index *`p_variantOutputParameterIndex`*

**`getOutputParameterValueByIndex`(VARIANT *`&p_variantOutputParameterIndex`*);**

returns the value of the output parameter with index *`IOutputParameterIndex`*

**`getOutputParameterValueByName`(LPCTSTR *`p_bstrOutputParameterName`*);**

returns a string containing the name of the output parameter with name *`p_bstrOutputParameterName`*

**`getSchemeControllerInstance`(void);**

returns an object that can be used to send TUI or scheme commands to the Fluent session and perform more advanced functions using the ICoFluentSchemeController Interface

**`LoadCase`(LPCTSTR *`p_bstrCaseFileName`*);**

load the case file *`p_bstrCaseFileName`* from the Fluent working directory into Fluent

**`LoadData`(LPCTSTR *`p_bstrDataFileName`*);**

load the data file *`p_bstrDataFileName`* from the Fluent working directory into Fluent

**`put_ComponentDescription`(LPCTSTR *`newValue`*);**

sets the connected component description to *`newValue`*

**`put_ComponentName`(LPCTSTR *`newValue`*);**

sets the connected component name to *`newValue`*

**`put_NrIterations`(VARIANT *`&newValue`*);**

sets the number of iterations that `Calculate` will perform to *`newValue`*.

**`SaveCase`(LPCTSTR *`p_bstrCaseFileName`*);**

save the current Fluent case to *`p_bstrCaseFileName`* in the Fluent working directory

**`SaveData`(LPCTSTR *`p_bstrDataFileName`*);**

save the current Fluent data to *`p_bstrDataFileName`* in the Fluent working directory

**`setInputParameterValueByIndex`(VARIANT *`&p_variantInputParameterIndex`*, VARIANT *`&p_variantInputParameterValue`*);**

sets the value of the input parameter with index *`p_variantInputParameterIndex`* to *`p_variantInputParameterValue`*

**`setInputParameterValueByName`(*`LPCTSTR p_bstrInputParameterName`*, VARIANT *`&p_variantInputParameterValue`*);**

sets the value of the input parameter with index *`p_lInputParameterIndex`* to *`p_lfInputParameterValue`*

**`Terminate`(void);**

terminate the connected Fluent as a Server session

**class `CCoFluentSchemeController`**

**`DoMenuCommand`(LPCTSTR *`p_bstrMenuCommand`*);**

issues a TUI command to the connected Fluent session. Output from the command is not returned

**`DoMenuCommandToString`(LPCTSTR *`p_bstrMenuCommand`*);**

issues a TUI command to the connected Fluent session and returns the output

**`DownloadFileToBuffer`(LPCTSTR *`p_bstrRemoteFileName`*);**

returns the contents of the file named *`p_bstrRemoteFileName`* in the remote Fluent session working directory.

**`DownloadFileToFile`(LPCTSTR *`p_bstrRemoteFileName`*, LPCTSTR *`p_bstrLocalFileName`*);**

writes the contents of the file named *`p_bstrRemoteFileName`* in the remote Fluent session working directory to the local file *`p_bstrLocalFileName`*.

**`ExecScheme`(LPCTSTR *`p_bstrSchemeCommand`*);**

issues a scheme command to the connected Fluent session. Output from the command is not returned

**`ExecSchemeToString`(LPCTSTR *`p_bstrSchemeCommand`*);**

issues a scheme command to the connected Fluent session and returns the output

**`SetRpVar`(LPCTSTR *`p_bstrRpVar`*, LPCTSTR *`p_szRpVarValue`*);**

sets the value of the rpvar *`p_szRpVar`* to *`p_szRpVarValue`*.

**`GetRpVar`(LPCTSTR *`p_bstrRpVar`*);**

returns a string with the value of the rpvar *`p_bstrRpVar`*.

**`UploadFileFromBuffer`(LPCTSTR *`p_bstrRemoteFileName`*, VARIANT *`&p_variantLocalBufferContent`*);**

writes a file named *`p_bstrRemoteFileName`* in the remote Fluent session working directory with the contents of *`p_variantLocalBufferContent`*. If *`p_bstrRemoteFileName`* exists, it is overwritten.

**`UploadFileFromFile`(LPCTSTR *`p_bstrRemoteFileName`*, VARIANT *`p_bstrLocalFileName`*);**

writes a file named *`p_bstrRemoteFileName`* in the remote Fluent session working directory with the contents of *`p_bstrLocalFileName`*. If *`p_bstrRemoteFileName`* exists, it is overwritten.

There is also a DLL library with an interface, IAnsysSessionManager, for connecting to Ansys Session Manager. You can use this interface by including the following library in your application:

```
...\v150\fluent\fluent21.2.0\addons\corba\%ARCH%\COMAnsysSessionManager.dll
```

**class `CAnsysSessionManager`**

**`CleanUp`(void);**

attempt to connect to all sessions started by the current session manager. If a session fails to connect it will be removed from the list of running sessions.

**Important:**

If a Fluent session is active but slow to respond (due to a long process of reading a case, for example) CleanUp will wait until the session responds. Thus, depending on the status of the running sessions this call may take a long time to complete. Use with caution.

**`ConnectToRunningSession`(LPCTSTR *`p_bstrRunningSessionName`*);**

connect to the managed session named *`p_bstrRunningSessionName`*.

**`ConnectToSessionManager`(LPCTSTR *`p_bstrHost`*, VARIANT *`&p_variantPort`*)**

establish a connection to the Ansys Session Manager listening on the specified host and port.

**`get_NrAvailableApplications`(void);**

returns the number of applications that Ansys Session Manager has registered.

**`get_NrRunningSessions`(void);**

returns the number of running solver sessions being managed by Ansys Session Manager.

**`getApplicationAttributeDefaultValues`(LPCTSTR *`p_bstrApplicationName`*);**

returns the default attribute values of the named application.

**`getApplicationAttributeNames`(LPCTSTR *`p_bstrApplicationName`*);**

returns the attribute names of the named application.

**`getApplicationInterfaceInfo`(LPCTSTR *`p_bstrApplicationName`*);**

returns information about the interface of the named application.

**`getApplicationNameByIndex`(VARIANT *`&p_variantApplicationIndex`*);**

returns the name of the application with index *`&p_variantApplicationIndex`*.

**`getRunningSessionInterfaceInfo`(LPCTSTR *`p_bstrRunningSessionName`*);**

returns information about the interface of the named solver session.

**`getRunningSessionNameByIndex`(VARIANT *`&p_variantRunningSessionIndex`*);**

returns the name of the solver session with index *`&p_variantRunningSessionIndex`*.

**`ResuscitateSession`(LPCTSTR *`p_bstrApplicationName`*, LPCTSTR *`p_bstrLocation`*);**

similar to StartSession, but instead of creating a new working folder, ResuscitateSession will start the application in the remote folder designated by *`p_bstrLocation`*.

> **Note:**
>
> *`p_bstrLocation`* should be either the absolute path of an existing folder on the remote machine or a path relative to the folder in which Ansys Session Manager is running.

**`StartSession`(LPCTSTR *`p_bstrApplicationName`*);**

starts a session of the application *`p_bstrApplicationName`*.

**`StartSessionWithAttributes`(LPCTSTR *`p_bstrApplicationName`*, VARIANT *`&p_variantAttributeNames`*, VARIANT *`&p_variantAttributeValues`*);**

starts a session of the application *`p_bstrApplicationName`* with attributes set to the specified values.

#### 25.3.2.2. Registering the COM Connectors

If you are using a Windows platform you can register the COM connectors by using the following commands:

```
C:\>cd <ANSYS Installation>\ANSYS Inc\v150\fluent\fluent15.0.0\addons\corba\%ARCH%\
C:\>regsvr32 ComCoFluentUnit.dll
C:\>regsvr32 ComAnsysSessionManager.dll
```

**Note:**

You must have Administrator privileges to register the COM objects

# Chapter 26: Population Balance

This chapter contains information relating to turbulence models available as beta features in Ansys Fluent 2021 R2.



## 26.1. Coulaloglou and Tavlarides Breakage

The breakage frequency is given by

$$g(v)=C_1\frac{\varepsilon^{1/3}}{(1+\alpha)d^{2/3}}\exp\left(-C_2\frac{\sigma(1+\alpha)^2}{\rho_d\varepsilon^{2/3}d^{5/3}}\right) \tag{26.1}$$

where $C_1$ and $C_2$ are constants with values of 0.00481 and 0.08, respectively, $\sigma$ is the surface tension, $\alpha$ is the volume fraction of the dispersed phase, $\varepsilon$ is the turbulence energy dissipation rate of the primary phase, and $\rho_d$ is the density of the dispersed phase [1].

**Important:**

Make sure you first enable beta feature access, as described in Introduction (p. 1).

Refer to Enabling the Population Balance Model in the *Fluent User's Guide* to learn how to enable the population balance model. Enable the **Breakage Kernel** option and select **tavlarides-model** from the **Frequency** drop-down list. Enter the desired **Surface Tension** in the **Surface Tension for Population Balance** dialog box.

### 26.1.1. References

1. Coulaloglou, C. A. and Tavlarides, L. L., Description of Interaction Processes in Agitated Liquid-Liquid Dispersions, Chem. Eng. Sci., 32 (1977) 1289-1297.

# Chapter 27: Fluent LB Method Client

This chapter covers the Fluent LB Method client, also known as Fluent Lattice Boltzmann. Information is organized into a how-to-use section, a best practices section, and a tutorial.



## 27.1. Using Fluent Lattice Boltzmann

This section describes how to use Fluent Lattice Boltzmann and is organized into the following sections:



### 27.1.1. Introduction

Fluent Lattice Boltzmann is a time-explicit flow solver for unsteady, single-phase, 3D flows that is based on a discrete form of the Boltzmann transport equation. This contrasts with a finite volume solver where mass and momentum conservation are described directly by the Navier-Stokes equations.

Ludwig Boltzmann (1844-1906) developed his transport equation as the foundation for the kinetic theory of gases at a time before the existence of atoms was accepted widely. The Boltzmann transport equation can describe the statistical properties of collections of atoms and molecules that make up a flow. The probability distribution of particle velocities is central to the approach, and these distributions evolve both in space and time according to Boltzmann's equation.

Collisions between particles are an important part of the physics that is represented. The nature of these collisions influences the macroscopic observed viscous properties of the medium. When a fluid reaches an equilibrium state where the probability distributions are homogeneous and isotropic, the velocity distribution is the classical Maxwell-Boltzmann distribution for a gas.

The familiar macroscopic quantities of a fluid (such as pressure, velocity and stresses) can be determined from moments of the probability distributions: the Boltzmann equation is solved for the probability distributions, and familiar flow quantities are then determined via a data reduction step.

The Boltzmann equation can be applied broadly to statistical physics problems, and some approximations are appropriate for the modeling of unsteady fluid flow. For example, the Knudsen number must be small. The Knudsen number is the ratio of the mean free path between collisions to the characteristic length scale of the geometry in the problem. Likewise, the collisions are modeled using the Bhatnagar-Gross-Krook (BGK) approximation, in which the velocity distributions are assumed to never be very far from an equilibrium state.

In the low Knudsen number limit, the macroscopic velocity and pressure associated with solutions of the Boltzmann equation can be demonstrated to correspond to solutions of the Navier-Stokes equations. There is a caveat: an extra term corresponding to a compressibility effect appears. This term resembles the artificial compressibility factor that is sometimes introduced into solution methods for the incompressible Navier-Stokes equations. The equations support an acoustic characteristic in which pressure and density fluctuations are proportional to each other. It is important to note that this term does not incorporate all the dynamic characteristics associated with the flow of an ideal gas.

A consequence of the compressibility effect is the presence of pressure waves in the flow domain for many problems. These traveling pressure disturbances often persist for many time steps and reverberate throughout the flow domain.

The discretization of the Boltzmann equation can be achieved using a spectral method to represent the probability distributions for the velocities. Using a Hermite polynomial basis for the velocity distribution, the Boltzmann equation discretizes into an explicit space-time marching scheme on a regular lattice—the Fluent Lattice Boltzmann (LB) equation. A consequence of the variable normalization and discretization is that the speed of sound is $1/\sqrt{3}$. Furthermore, the time step size chosen for the computation must be sufficiently small so that the numerical Mach number for the calculation is also small. The numerical Mach number is the ratio of the normalized flow speed in the domain to the speed of sound, and it is accepted generally that it should not exceed a value of 0.1 in LB calculations. This intimate connection between the scaling of physical quantities, the lattice being used, and the time step size is a consequence of mapping the Boltzmann equation onto the lattice so that the solutions approximate fluid physics well.

The LB algorithm itself is striking in its simplicity of form. There are two parts to it: streaming and collision. The streaming set involves transporting probability components to adjacent positions in the lattice. The collision involves determining a local representative equilibrium state and relaxing the current state towards that at a rate determined by the local effective viscosity. When that viscosity is chosen appropriately, the result can be behavior that corresponds to large eddy simulation. In Fluent Lattice Boltzmann, the consistent Smagorinsky formulation is used.

Different choices of the number of Hermite basis functions lead to different levels of fidelity in the discretization, and that choice has implications for how data flows in the discretized problem. More basis functions lead to higher fidelity at the price of more memory and computation. The standard D3Q19 discretization is used here. If each lattice point lies at the centroid of a cube, then the algorithm involves streaming to adjacent points through faces and edges (but not corners).

Fluent Lattice Boltzmann can be considered as running on a Cartesian octree mesh where there can be a hierarchical subdivision of individual cubes into eight sub-cubes. The lattice points for the LB calculations coincide with the cube centroid locations. The use of an octree provides the advantage that a locally adapted mesh can resolve important geometric and flow features at different spatial scales. The nature of the LB algorithm demands that the time step size on the spatially refined mesh be proportionately smaller than on coarser scales. As a result, more time steps are taken on finer scales. An appropriate numerical scheme is applied in order to exchange data between the scales at the octree mesh refinement transitions.

The boundary conditions that are supported within Fluent Lattice Boltzmann are velocity inlets, pressure outlets, symmetry planes, and walls. Even though the underlying lattice arrangement is Cartesian, the boundaries need not conform to Cartesian planes. Wall treatments for laminar and turbulent flows are provided, and the turbulent treatment makes use of a wall function approach.

The workflow for the setup and running of Fluent Lattice Boltzmann is very similar to that of an unsteady Fluent finite volume computation. Surface geometry, in the form of a surface mesh or case file, and boundary conditions are provided as input. Sizing for the mesh is defined by default or explicitly, for each boundary and the mesh and flow is initialized. A time step size is chosen and the calculation proceeds for a specified number of time steps. Various monitors (such as forces, moments, or mass flow rates) can be specified and are updated as the calculation progresses. A variety of postprocessing options are available to visualize the flow field.

#### 27.1.1.1. Program Capabilities

The current Fluent Lattice Boltzmann capabilities are:

- You can run on CPU cores or an NVIDIA GPU.
- You can start from a variety of different meshes: surface, volume or Ansys Fluent case and data.
- You can have local mesh refinement with the built-in octree mesher.
- You have access to much of Ansys Fluent's postprocessing capabilities.
- You can postprocess while the solver is running.

#### 27.1.1.2. Known Limitations

There are some limitations associated with the Fluent Lattice Boltzmann workspace, many of which are planned for resolution at a future release:

- Only one cell zone is allowed per simulation. While you can merge adjacent fluid cell zones to create a domain with a single cell zone, Fluent Lattice Boltzmann cannot merge zones that are fully separated by a wall. Refer to Defining the Domain Parameters for Meshing (p. 158) for additional information on merging cell zones.
- The solver is limited to incompressible, single-phase 3D flows.

- The solver is limited to a single multi-core CPU (shared memory parallel) or a single NVDIA GPU.
- Boundary types are limited to: velocity inlet, pressure outlet, symmetry, and wall.
- The solver expects all inputs in SI units.

## 27.1.2. Basic Steps for CFD Analysis Using Fluent Lattice Boltzmann

Before you begin your CFD analysis using Fluent Lattice Boltzmann, careful consideration of the following issues will contribute significantly to the success of your modeling effort. Also, when you are planning a CFD project, be sure to take advantage of the customer support available to all Ansys Fluent users.

For more information, see the following section:



### 27.1.2.1. Steps in Solving Your CFD Problem

After you have determined the important features of the problem you want to solve, perform the following basic procedural steps:

1. Define the modeling goals.
2. Create a surface mesh and/or 3D case file (with or without a corresponding data file) to represent the fluid domain.
3. Define the parameters that will be used to generate an octree mesh for the domain.
4. Set up the models, materials, cell zones, boundary properties, and report definitions.
5. Generate an octree mesh and initialize the solution flow fields.
6. Set up calculation activities.
7. Compute the solution.
8. Examine and save the results.
9. Consider revisions to the setup, if necessary.

Step 2. of the solution process requires a geometry modeler and/or mesh generator to produce a mesh file (such as a `.msh*`, `.stl`, or `.cas*` file). For example, you could use the meshing mode of Fluent or DesignModeler and Ansys Meshing within Ansys Workbench. For more information on creating geometry and/or meshes using each of these programs, refer to their respective manuals.

The details of the remaining steps are covered in the sections that follow.

## 27.1.3. Starting and Exiting Fluent Lattice Boltzmann

The following describe how to start Fluent Lattice Boltzmann on a Windows or Linux system:

### 27.1.3.1. Starting Fluent Lattice Boltzmann Using the Fluent Launcher

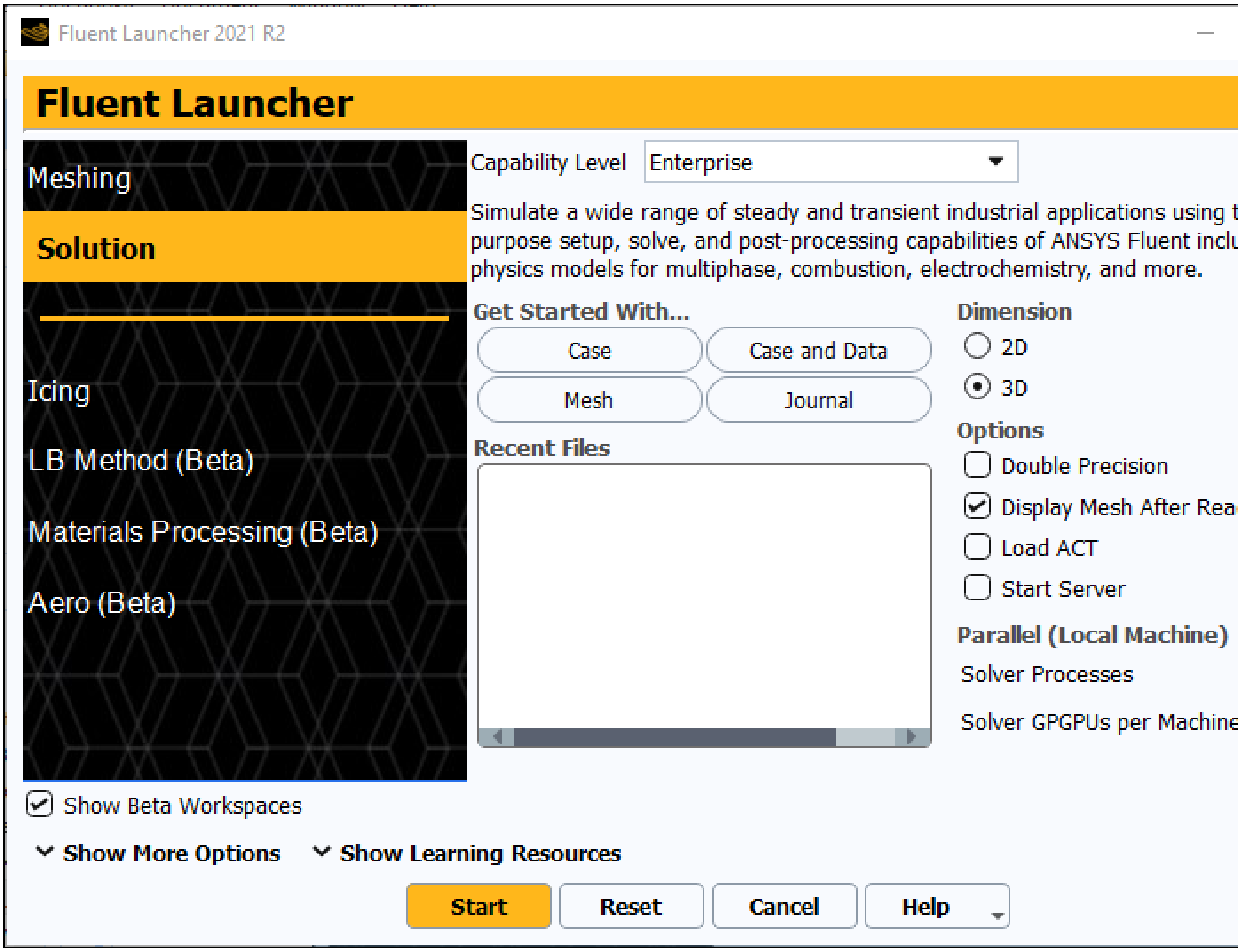
To start Fluent Lattice Boltzmann from the Fluent Launcher:

1. Open the Fluent Launcher. *Refer to Starting Ansys Fluent Using Fluent Launcher in the Fluent User's Guide for additional information on opening the Fluent Launcher.*

2. Select **Enterprise** from the **Capability Level** drop-down list.

3. Enable **Show Beta Workspaces**.

4. Select the **LB Method (Beta)** workspace in the Fluent Launcher.



5. (Optional) Use your system's GPU to run the solver by increasing the **Solver GPGPUs per Machine**. *Refer to* *Checking LB System Settings (p. 155)* *for additional information on controlling the GPU and CPU solver settings.*

   **Important:**

   Setting **Solver GPGPUs per Machine** to `1` uses 2 licenses (while keep it at `0` only uses a single license).

6. (Optional) Select a case | mesh | Fluent case | script to start. Selecting any of these starting files and launching Fluent LB Method automatically loads the selected file. *The **Start** button also becomes **Start With Selected Options**.*

7. Click **Start**.

**Important:**

Fluent Lattice Boltzmann is only available at the **Enterprise** licensing level.

#### 27.1.3.2. Starting Fluent Lattice Boltzmann from the Command Line

To start Fluent Lattice Boltzmann from the command line, enter the following in a command prompt window:

`fluent 3ddp [-t1] [-gpgpu=1] -app=lbapp [-appscript=` *file_name* `.py]`

Note that for this release, only a single processor may be specified (through the optional use of the `-t1` argument). Future releases will allow you to take advantage of multiple simultaneous processors.

Including `-gpgpu=1` is optional, and allows you to use a single general purpose graphics processing unit (GPGPU) to accelerate the calculation. Using a GPGPU requires HPC licenses; licensing details can be found in HPC Licensing in the *Ansys, Inc. Licensing Guide*. GPGPUs that are supported in the current release are posted on the Platform Support section of the Ansys Website.

You also have the option of including `-appscript=` *file_name* `.py` if you want to have the application immediately run a journal named *file_name* `.py`. For further details about journals, see Creating and Reading Journals / Scripts (p. 151).

By default, Fluent Lattice Boltzmann will use all of the cores on your machine. If you want to limit the number of cores used, you must set the `OMP_NUM_THREADS` environment variable equal to your desired number of cores.

**Important:**

For Windows, be sure the path to your Fluent Lattice Boltzmann home directory is in your command search path environment variable by executing the `setenv.exe` program located in the Ansys Fluent directory (for example, `C:\Program Files\ANSYS Inc\v212\fluent\ntbin\win64`).

**Note:**

For Linux, the best graphics driver is selected automatically, and the X11 driver is used by default when the required graphics support is not detected. If you feel it is necessary to use an alternate driver, you can include the `-setenv="HOOPS_PICTURE=`*driver_name* `"` command line option, where *driver_name* is the preferred graphics driver.

#### 27.1.3.3. Exiting Fluent Lattice Boltzmann

You can exit Fluent Lattice Boltzmann by selecting **Exit** in the **File** ribbon tab or by clicking the ✕ button in the top right corner of the application. For the latter, a **Question** dialog box will open to confirm if you want to proceed.

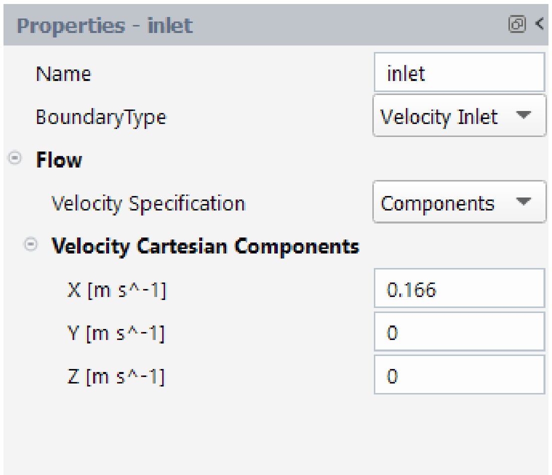
### 27.1.4. Graphical User Interface (GUI)

The graphical user interface (GUI) of Fluent Lattice Boltzmann is very similar to the solution mode of Fluent, in that:

- it includes a ribbon, an outline view, a graphics window, a console, toolbars, dialog boxes, and quick search (as described in GUI Components in the *Fluent User's Guide*)
- it can be modified (as described in Customizing the Graphical User Interface in the *Fluent User's Guide*)
- it can be set to match your preferences (as described in Setting User Preferences/Options in the *Fluent User's Guide*)
- it has access to the help system (as described in Using the Help System in the *Fluent User's Guide*)

When setting up your simulation, the majority of your actions will start by performing actions in the **Outline View**: you can right-click an item and use the menu that opens to perform a function (such as **Initialize** from the **Solution** item); and you can left-click an item to open a related **Properties** window. The **Properties** windows are similar to task pages or dialog boxes, in that they allow you to define settings, run the calculation, and postprocess results. Note that your settings are saved as you define them in a **Properties** window, unlike dialog boxes (which you may open from the **Outline View** or settings in the **Properties** window) which require you to click an **Ok** button.

**Figure 27.1: The Properties Window for a Boundary**




Note that the console does not interact with a text user interface (TUI), but does allow Python scripting; however, it is recommended that you instead read Python commands from a journal / script file (as described in Creating and Reading Journals / Scripts (p. 151)), as this allows for each command to complete before the next command is executed. This is especially important for interdependent commands. Typing or pasting a series of Python commands directly into the console could result in undesirable behavior, as there is no waiting for the previous command to complete before the next command is executed.

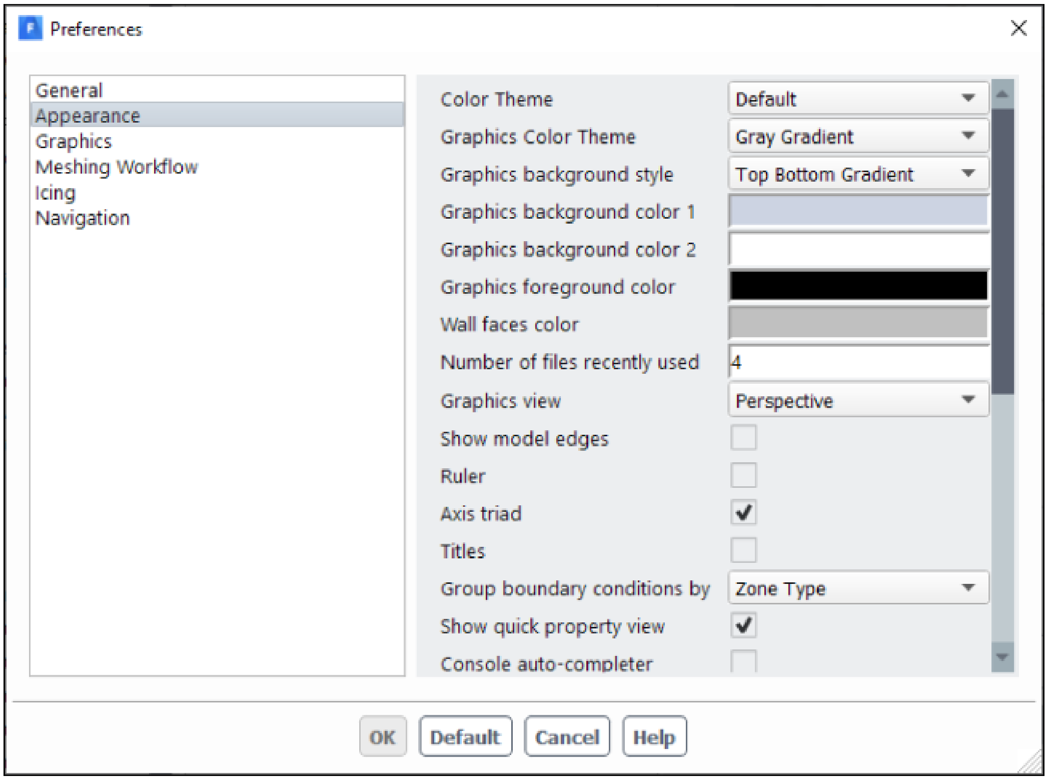
You can change how graphics objects are displayed by taking action directly in the graphics window, as well by using the **View** ribbon tab. The available actions and controls are similar to those described in Displaying Graphics and Modifying the Views in the *Fluent User's Guide*.

## 27.1.5. Setting Preferences

You can specify global settings that are applied whenever you are operating in Fluent Lattice Boltzmann. These settings are case-independent and are controlled using the **Preferences** dialog box.

**File** → **Preferences...**

**Figure 27.2: Preferences Dialog Box**




*Note that there is only one Preferences file that applies to all Fluent workspaces. Not all preference settings are relevant to all workspaces.* Refer to Setting User Preferences/Options in the *Fluent User's Guide* for additional information.

## 27.1.6. Creating and Reading Journals / Scripts

A journal contains a sequence of Fluent Lattice Boltzmann commands, arranged as they would be entered through the graphical user interface (GUI). The GUI commands are recorded in the journal file as Python 2.7 code lines. You can also create journal files manually with a text editor. If you want to include comments in your file, be sure to put a hash (#) at the beginning of each comment line.

The purpose of a journal file is to automate a series of commands. Another use is to produce a record of the input to a program session for later reference, although transcript files are often more useful for this purpose. (For details, see Creating Transcript Files (p. 152)).

Command input is taken from the specified journal file until its end is reached, at which time control is returned to the standard input (usually the mouse). Each line from the journal file is echoed to the standard output (usually the console) as it is read and processed.

When using journal files, note the following:

- Journal files written in one release of Fluent Lattice Boltzmann should only be read back in the same release; the use of a journal from a different release is not supported.
- A journal file is, by design, a record and playback facility. It contains no information about the state in which it was recorded or the state in which it is being played back.

- Be careful not to change the folder while recording a journal file. Also, try to re-create the state in which the journal was written before you read it into the application.

  For example, if your journal file includes an instruction to save a new file with a specified name, you should check to see if a file with that name exists in your folder before you read in your journal file. If a file with that name exists and you read in your journal file, when the program reaches the write instruction, it will prompt for a confirmation to overwrite the old file. Since the journal file does not contain any response to the confirmation request, Fluent Lattice Boltzmann cannot continue to follow the instructions of the journal file.

- Other conditions that may affect the application's ability to perform the instructions contained in a journal file can be created by modifications or manipulations that you make within the program.

  For example, if your journal file creates several surfaces and displays data on those surfaces, you must be sure to read in appropriate case and data files before reading the journal file.

- At a point of time, only one journal file can be open for recording, but you can write a journal and a transcript file simultaneously. You can also read a journal file at any time.

- Only successfully completed commands are recorded. For example, if you stopped the execution of a calculation using the **Interrupt** button, it will not be recorded in the journal file.

To start the journaling process, select the **File/Write/Start Journal...** ribbon tab item.

**File → Write → Start Journal...**

After you enter a name for the file in the **Select File** dialog box (being sure to include the `.py` extension, to make it easier to later filter by file type), journal recording begins. The **Start Journal...** menu item becomes the **Stop Journal** menu item. You can end journal recording by selecting **Stop Journal**, or by exiting the program.

**File → Write → Stop Journal**

After you have created a journal to capture a series of commands, you can modify it in a text editor so that it behaves as a script; that is, you can add Python commands so that it contains more complex programming constructs, such as loops and branching.

You can read a journal / script file into the program using the **Select File** dialog box opened by selecting the **File/Read/Journal/Script File...** ribbon tab item.

**File → Read → Journal/Script File...**

Alternatively, you can specify that a journal / script file is read immediately after Fluent Lattice Boltzmann opens, as described in Starting Fluent Lattice Boltzmann from the Command Line (p. 148).

## 27.1.7. Creating Transcript Files

A transcript file contains a complete record of all standard input to and output from Fluent Lattice Boltzmann, which is usually all keyboard and graphical user interface (GUI) input and all screen output. GUI commands are recorded as Scheme code lines in transcript files. Fluent Lattice Boltzmann creates a transcript file by recording everything typed as input or entered through the GUI, and everything printed as output in the console.

The purpose of a transcript file is to produce a record of the program session for later reference. Because they contain messages and other output, transcript files (unlike journal files), cannot be read back into the program.

---

**Important:**

Only one transcript file can be open for recording at a time, but you can write a transcript and a journal file simultaneously. You can also read a journal file while a transcript recording is in progress.

---

To start the transcription process, select the **File/Write/Start Transcript...** ribbon tab item.

**File → Write → Start Transcript...**

After you enter a name for the file in the **Select File** dialog box, transcript recording begins and the **Start Transcript...** menu item becomes the **Stop Transcript** menu item.

You can end transcript recording by selecting **Stop Transcript**, or by exiting the program.

**File → Write → Stop Transcript**

### 27.1.8. Reading, Writing, and Importing Case, Data, and Mesh Files

Fluent Lattice Boltzmann can read in case and data files created in Fluent Lattice Boltzmann (`*.cas.lb`).

**File → Read → Case... | Data... | Case & Data...**

Alternatively, you can read in Fluent Lattice Boltzmann case and data files by selecting the **Setup** branch in the Outline View tree, click **Read** and choose either **Case...** or **Case & Data...**.

Fluent Lattice Boltzmann allows you to import the following:

- mesh files: `.msh.h5`, `.msh`, `.msh.gz`, or `.stl` files
- case files: `.cas.h5`, `.cas`, or `.cas.gz`
- data files: `.dat.h5`, `.dat`, or `.dat.gz`

---

**Important:**

Fluent Lattice Boltzmann is only for 3D flows. Do not try to mimic 2D flow (by limiting the number of octs in the spanwise direction).

---

Details about such files can be found in Reading and Writing Files in the *Fluent User's Guide*.

You can read or write such files by using the menus under the **File** ribbon tab or the **Properties** window opened by left-clicking **Setup** in the **Outline View**.

**Figure 27.3: The Files Menu for Reading**

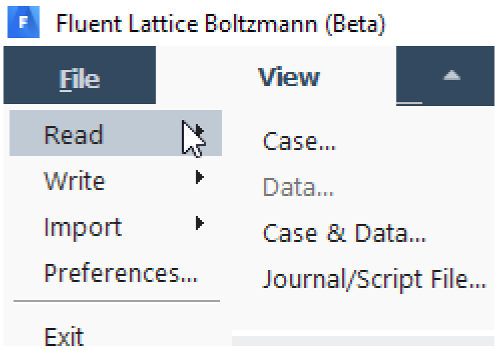



**Figure 27.4: The Files Menu for Writing**

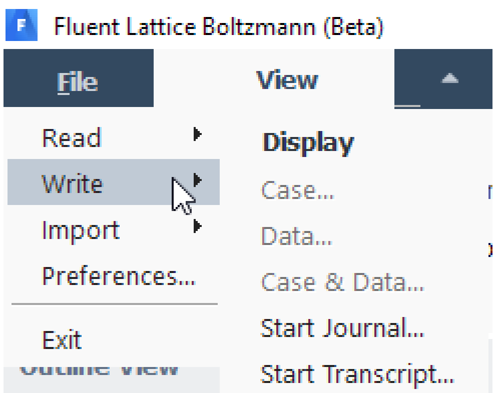
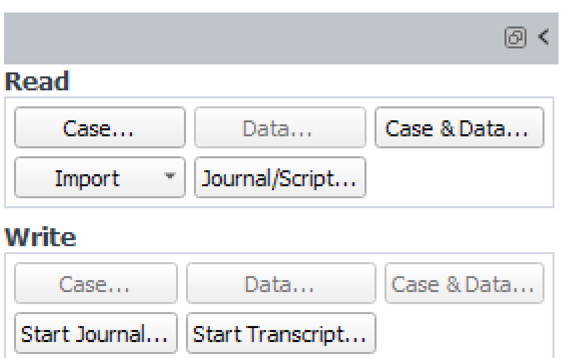



**Figure 27.5: The Properties Window for the Setup**




When reading/importing files, note the following:

- The files must be 3D.
- Volume meshes must be a single fluid cell zone.
- Surface meshes with the extension `.msh` must describe a single cell zone.

---

**Note:**

- Fluent Lattice Boltzmann cannot import boundary meshes exported from the Fluent **Solution** workspace.

- A boundary mesh written out in the default Common Fluids Format (`.msh.h5`) from the Fluent **Meshing** workspace cannot be imported into Fluent Lattice Boltzmann.

---

## 27.1.9. Checking LB System Settings

Depending on how you launched Fluent Lattice Boltzmann, you can either control the number of CPU cores that are used for the LB solver or which GPU is selected for the solver; Launching with **Solver GPGPUs per Machine** set to `0` (in the Fluent Launcher) lets you control the former, while launching with it set to `1` lets you control the latter. Once the application is open, you can control the setting in the **Properties - System** panel.

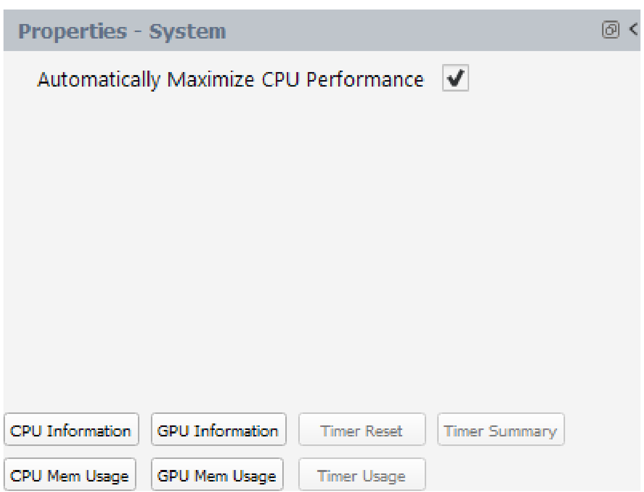
The **Properties - System** panel also allows you to print details on the CPU and GPU, information on memory usage, and print time-related statistics for the simulation.

### Controlling the Number of Cores:

To reduce the number of cores used in this LB solver session:

1. Open the system properties (**Properties - System**).

   **Setup** → **System**

2. Disable **Automatically Maximize CPU Performance**.

3. Specify how many CPU cores you want the solver to use in the **CPU solver threads** field.

   *You can click **CPU Information** to print information about the CPU to the **Console**, including the number of available cores.*

## Controlling the GPU:

---

**Important:**

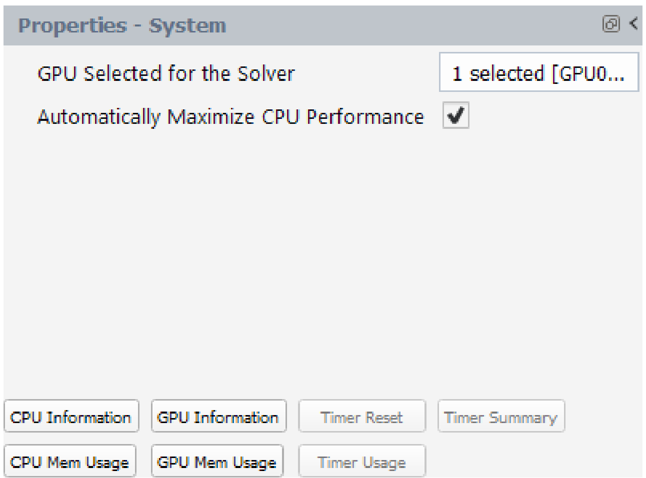
You can only change the selected GPU prior to initialization.

---

To change the selected GPU in this LB solver session:

1. Open the system properties (**Properties - System**).

   **Setup** → **System**

2. Click the **GPU selected for solver** field to open the **gpuList** dialog box.

3. Select the desired GPU and click **Ok**.

   *You can click **GPU Information** to print information about the GPU to the **Console**, including the available GPUs.*

The buttons available for printing information to the **Console** include:

- **CPU Information**—prints information about the hostname, numbers of CPUs, CPU cores, and available memory.

- **CPU Mem Usage**—prints information on CPU memory usage, including the number gigabytes used by each user at the current time, peak usage, and total usage.

- **GPU Information**—prints information on the GPUs installed on this machine, including the number of GPUs, the manufacturer, version, and so on.

- **GPU Mem Usage**—prints information on GPU memory usage, which will only show data if you specify one or more **Solver GPGPUs per Machine** when launching Fluent Lattice Boltzmann.

- **Timer Reset**—resets the timer that accumulates data as the solver is running.

- **Timer Usage**—prints information about the time that has ellapsed since the solver began running. An example is shown below:

```
=========== ========== ========== ========== ==========
Operations by hierarchy
=========== ========== ========== ========== ==========
Levels >>  |     0          1          2        Total
Operations |
----------- ---------- ---------- ---------- ----------
Solve      |        1          2          4          7
  Average  |        2          4          0          6
  lbmStep  |        1          2          4          7
----------- ---------- ---------- ---------- ----------


=========== ========== ========== ========== ==========
Octs per operation per time-step
=========== ========== ========== ========== ==========
Levels >>  |     0          1          2        Total
Operations |
----------- ---------- ---------- ---------- ----------
Average    |   464552      16128          0     480680
Stream     |   231616       5544      34272     271432
Collide    |   231616       5544      34272     271432
BC         |    48124       1136      22752      72012
----------- ---------- ---------- ---------- ----------


=========== ========== ========== ========== ==========
Time in s for 4600 time-steps
=========== ========== ========== ========== ==========
Levels >>  |     0          1          2        Total
Operations |
----------- ---------- ---------- ---------- ----------
Average    | 6.761e+01  7.458e+00  0.000e+00  7.507e+01
Stream     | 4.666e+01  1.487e+00  6.024e+00  5.417e+01
Collide    | 4.161e+01  1.871e+00  1.783e+01  6.131e+01
BC         | 2.738e+01  6.430e-01  1.300e+00  2.933e+01
----------- ---------- ---------- ---------- ----------
Total      | 1.833e+02  1.146e+01  2.515e+01  2.199e+02
----------- ---------- ---------- ---------- ----------


=========== ========== ========== ========== ==========
Time in us per oct per time-step
=========== ========== ========== ========== ==========
Levels >>  |     0          1          2        Mean
Operations |
----------- ---------- ---------- ---------- ----------
Average    | 3.164e-02  1.005e-01  0.000e+00  3.395e-02
Stream     | 4.379e-02  5.831e-02  3.821e-02  4.338e-02
Collide    | 3.906e-02  7.337e-02  1.131e-01  4.911e-02
BC         | 1.237e-01  1.230e-01  1.242e-02  8.853e-02
----------- ---------- ---------- ---------- ----------
```

- **Timer Summary**—prints a summary of the solver performance. An example is shown below:

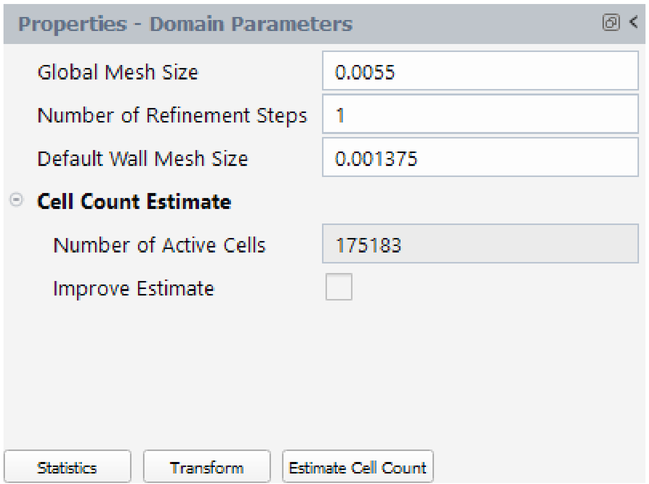
```
LB Solver Performance Summary
----------------------------------------------
      Number of timesteps : 4600
          Number of cells : 242956
Total wall clock time (s) : 2.198800e+02
              average     : 7.507202e+01
               stream     : 5.416801e+01
              collide     : 6.131298e+01
                   bc     : 2.932700e+01
                     CAPS : 5.082761e+00
----------------------------------------------
CAPS: [C]ells [A]dvanced by 1 second flowtime
      [P]er 1 [S]econd wall clock time
s
```

## 27.1.10. Defining the Domain Parameters for Meshing

You can specify the refinement of the octree mesh (which is generated for the domain when you initialize the solution).

- Open the domain parameter properties window (**Properties - Domain Parameters**).

  **Setup** → **Domain Parameters**




1. Clicking the **Statistics** button at the bottom of the window prints information about the current surface mesh sizing to the console, which can assist you with determining inputs for the **Global Mesh Size** and **Default Wall Mesh Size**.

2. Enter a value (in meters) for the **Global Mesh Size**, to define the size of the cells in the interior of the cell zone (away from the boundaries).

3. Enter a value for the **Number of Refinement Steps**. This unitless number controls how the mesh will transition between areas of refinement and coarseness: higher values result in more gradual mesh transitions (which are appropriate for modeling flows with thicker boundary layers).

4. Enter a value (in meters) for the **Default Wall Mesh Size** to define the cell size you want at wall boundaries.

5. (Optional) Enable **Improve Estimate** to improve the quality of the cell count estimate. *Note that the improved estimation process comes with an increase in estimation time.*

6. (Optional) Click **Estimate Cell Count** to update the **Cell Count Estimate** displayed in the **Number of Active Cells** field based on your settings.

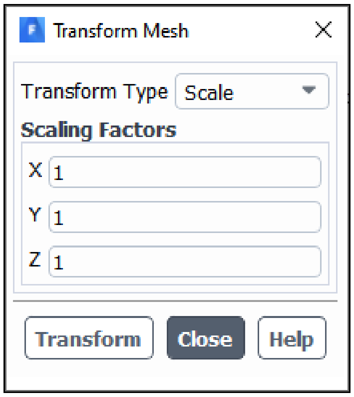
   ---

   **Note:**

   When estimating the cell count with **Improve Estimate** enabled, if Fluent Lattice Boltzmann determines that performing an accurate estimate will take prohibitively long, only an approximate estimate will be performed.

   ---

7. (Optional) Scale, rotate, and/or translate the mesh, as necessary, by clicking **Transform** to open the **Transform Mesh** dialog box.




   1. Select the type of mesh transformation that you want to perform from the **Transform Type** drop down list.

      - **Scale**—provide conversion factors to convert the model to SI units.
      - **Rotate**—provide the **Rotation Angle**, **Axis**, and **Origin** for the orientation of the model.
      - **Translate**—provide the offset for each axis.

   2. Click **Transform** and close the dialog box once you are done transforming the mesh.

   **Note:**

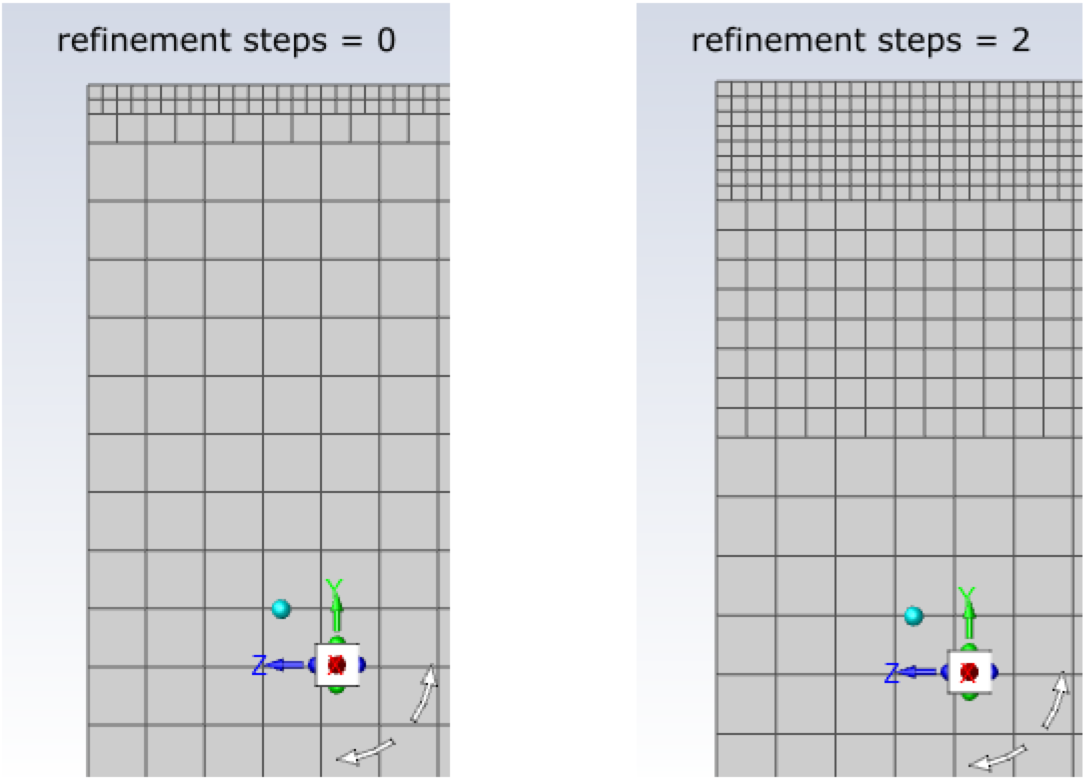
   You cannot transform the mesh if refinement boxes are defined.

After you initialize the solution (as described in Calculating a Solution (p. 175)), you can view the resulting octree mesh to ensure the refinement is appropriate. The following figure shows an example where a refined mesh was defined for the top boundary, and a larger mesh size was used for the global mesh and remaining boundaries. This figure illustrates how the number of refinement steps affects the mesh transitions.

**Important:**

You can change the mesh parameters and regenerate the mesh at any point in the setup process (by initializing again). Note that clicking **Initialize** again regenerates the mesh and reinitializes the flow field.

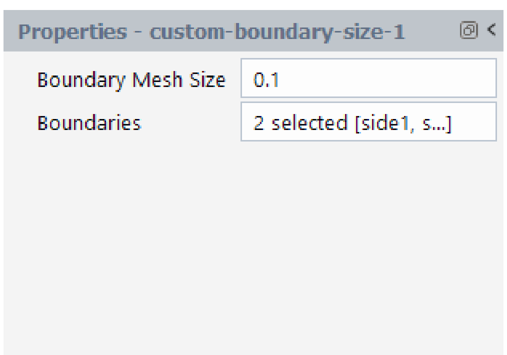
**Figure 27.6: Cross Sections Displaying Octree Mesh Transitions**




## Custom Mesh Boundary Sizes

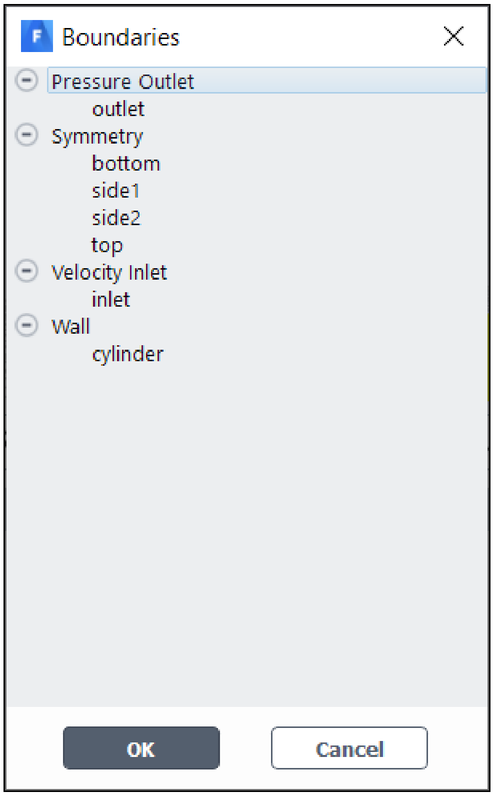
If you want to specify a mesh size for an individual boundary or a group of boundaries, you can define custom sizes for individual or groups of boundaries using the **Properties -custom-boundary-size-** <#> window:

**Setup** → **Domain Parameters** → **Custom Mesh Boundary Sizes** **New...**




1. Enter a value (in meters) for the **Boundary Mesh Size**.
2. Click the **Boundaries** field to open the **Boundaries** dialog box.

**Figure 27.7: The Boundaries Dialog Box**



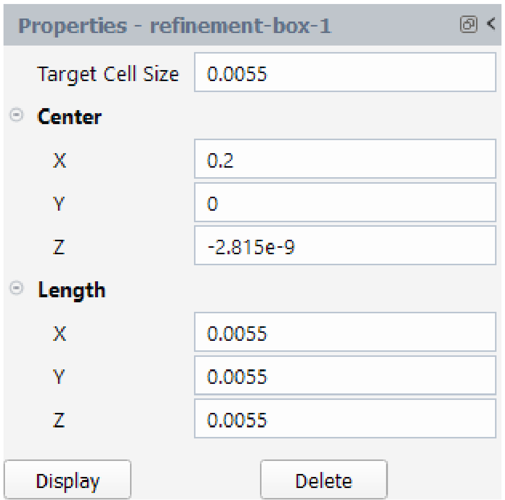

Select the boundaries where you want the custom mesh size and click **Ok**.

3. Repeat the previous steps for any other boundaries where you unique mesh size applied.

## Refinement Boxes

If you want specific areas of your simulation to be refined that are not tied to a specific boundary, you can define refinement boxes. *Note that once refinement boxes are defined, you cannot scale, rotate, or translate the mesh.*

**Setup** → **Domain Parameters** → **Refinement Boxes** **New...**




1. Specify the target cell size for the refinement box in the **Target Cell Size** field.

2. Enter the **X**, **Y**, and **Z** coordinates at the **Center** of the refinement box.

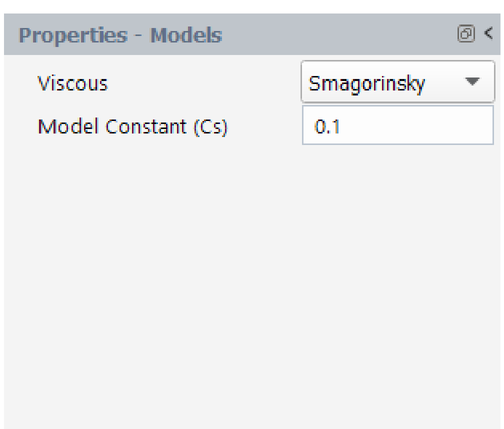
3. Enter **Length** of the **X**, **Y**, and **Z** dimensions of the refinement box.

4. Click **Display** to display the refinement box and confirm that it is located where you expect.

## 27.1.11. Modeling Turbulence

You can specify whether the flow is to be treated as laminar or turbulent by making a selection from the **Viscous** drop-down list in the **Properties - Models** window. The **Smagorinsky** turbulence model that is available in Fluent Lattice Boltzmann is a Large Eddy Simulation (LES) model based on a consistent Smagorinsky formulation [1] (p. 220). When the Smagorinsky model is enabled, the flow adjacent to the walls is modeled using a turbulent wall function.

**Setup → Models**




You can modify the value of the Smagorinsky **Model Constant (Cs)** to suit your case.

'The smagorinsky model constant is case dependent. For decaying turbulence absent of underlying shear, the coefficient is typically selected as Cs=0.18. For the more important class of shear flows, a value of Cs=0.1 is more appropriate.' [2] (p. 220). Refer to Smagorinsky-Lilly Model in the *Fluent Theory Guide* for additional information about the Smagorinsky-Lilly turblulence model.

### 27.1.12. Material Properties

As the Fluent Lattice Boltzmann solver is primarily intended for external aerodynamics, the default material is air. However, you may see that additional materials are present, if you import a Fluent case with additional materials defined.

**Note:**

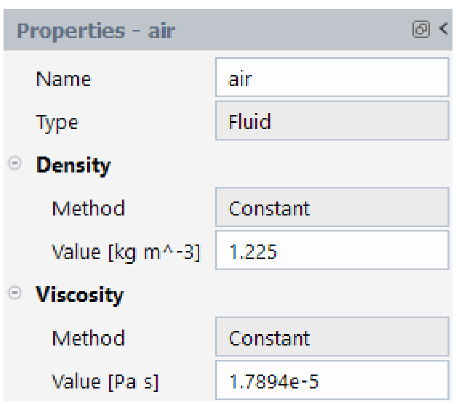
Solid materials are not available for specification.

To control the material properties for this simulation:

1. Open the properties table for air (**Properties - air**) by expanding **Materials** in the tree and selecting **Air**.

   **Setup** → **Materials** → **air**

   **Figure 27.8: The Properties Window for the Setup**

2. (Optional) Modify the density in the **Value [kg m^-3]** field under **Density**.
3. (Optional) Modify the viscosity in the **Value [Pa s]** field under **Viscosity**.
4. If you want to create additional materials, right-click the **Materials** branch in the Outline View tree and select **New...**. Now you can provide the material properties as outlined in the preceding steps.
5. You can delete materials by right-clicking them in the tree and selecting **Delete**.

**Note:**

Once you define a material, it must assigned to the cell zone before its properties are considered in any calculation.

## 27.1.13. Cell Zone and Boundary Conditions

Fluent Lattice Boltzmann only allows for a single fluid cell zone and no others. This cell zone may comprise of multiple boundaries. For example, if you are evaluating a car in a wind tunnel, the wind tunnel is your cell zone and the car is a hole in the cell zone (rather than a solid cell zone as it would be defined in the standard Ansys Fluent solver).

This chapter describes the types of boundaries that are available in Fluent Lattice Boltzmann and their appropriate use.



### 27.1.13.1. Setting up the Cell Zone

Although Fluent Lattice Boltzmann only allows for a single fluid cell zone, if you import a Fluent case file, it may contain more than one cell zone. As long as the additional zones are not related to the zone you want to simulate, they can be deleted by right-clicking them in the Outline View tree and selecting **Delete**. If all of the cell zones are related to your simulation, you can merge them together to create a single continuous fluid cell zone.

To setup a cell zone for your simulation:

1. Open the properties for <cell zone name> by selecting the cell zone in the Outline View tree (under **Setup**/**Cell Zones**).

2. Select the material for the cell zone from the **Material** drop-down list.

#### Merging and Deleting Cell Zones

Although Fluent Lattice Boltzmann only allows for a single fluid cell zone, if you import a Fluent case, it may contain more than one fluid cell zone. All of the solid cell zones will automatically be removed and all of the fluid zones will be imported. At this point, you must merge or delete all of the additional fluid zones, so that you have a single, uniform region.

---

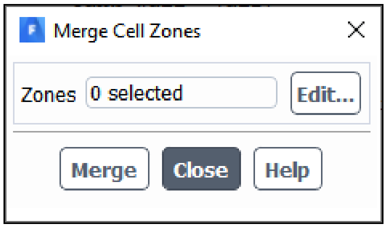
**Important:**

While you can merge adjacent fluid cell zones to create a domain with a single cell zone, Fluent Lattice Boltzmann cannot merge zones that are fully separated by a wall.

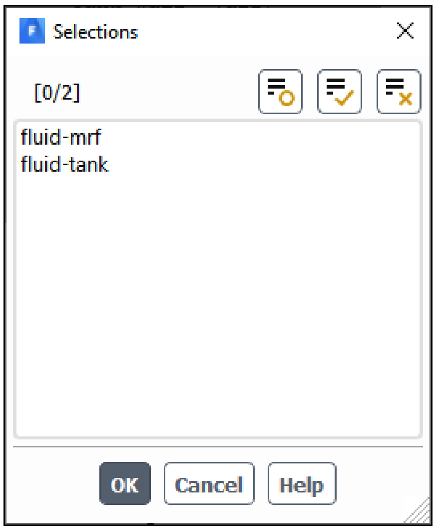
---

To merge cell zones:

1. Click **Merge Cell Zones** at the bottom of **Properties - Cell Zones**.




2. Click **Edit...** to open the **Selections** dialog box.



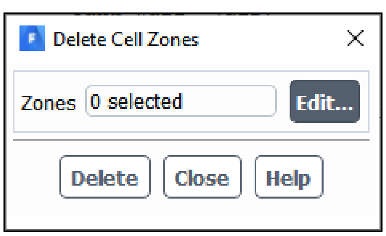

3. Select the zones that you want to merge and click **OK**.

4. Click **Merge**.

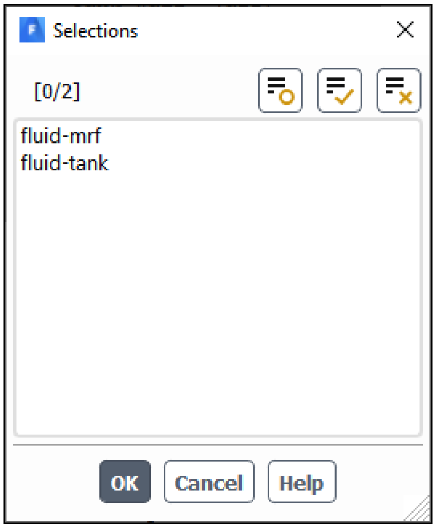
To delete cell zones you can either:

- Use the **Delete Cell Zones** dialog box.

  1. Click **Delete Cell Zones** at the bottom of **Properties - Cell Zones**.




  2. Click **Edit...** to open the **Selections** dialog box.

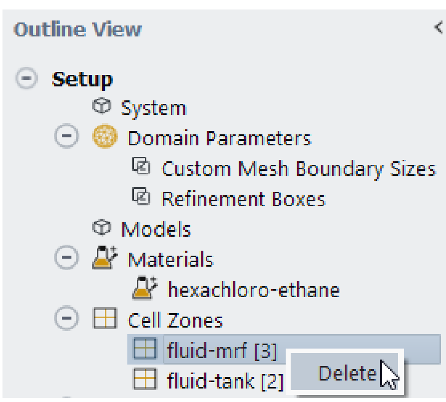



3. Select the zones you want to delete and click **OK**.

4. Click **Delete**.

- Use the right-click context menu in the Outline View tree.

  Right-click the cell zone(s) that you want to delete in the tree and click **Delete**.




### 27.1.13.2. Available Boundary Types

The following boundary types are available in Fluent Lattice Boltzmann:

- Velocity inlet
- Pressure outlet
- Wall

- Symmetry

---

**Important:**

For better solution stability, it is recommended that you avoid having velocity inlets in direct contact with pressure outlets.

---

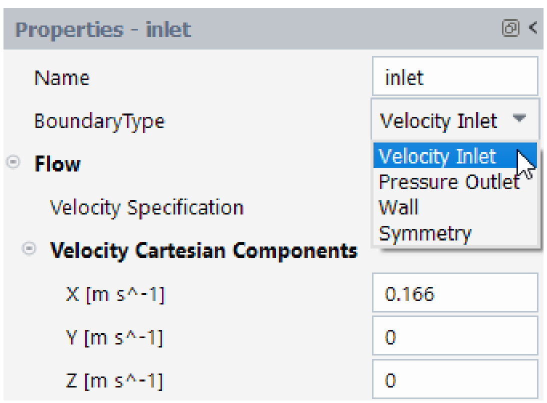
### 27.1.13.3. Changing Boundary Condition Types

Before you set any boundary conditions, you should check the zone types of all boundary zones and change any if necessary. For example, if your mesh includes a wall, but you want to use a velocity inlet instead, you will need to change the wall zone to a velocity-inlet zone.

The steps for changing a zone type using the graphical user interface are as follows:

1. Select the boundary or cell zone in the Outline View tree (under **Setup**/**Boundary Conditions**).

2. Select the correct zone type from the **Boundary Type** drop-down list.




### 27.1.13.4. Setting Boundary Conditions

To set boundary conditions:

1. Select the boundary in the Outline View tree.

   **Setup** → **Boundaries** → **<desired-boundary>**

2. Provide the inputs that are appropriate for the type of boundary that you selected.

Specific inputs for the following boundaries:

#### 27.1.13.4.1. Inputs for Velocity Inlets

Velocity inlets are used to define the velocity and scalar properties of the flow at inlet boundaries.

First, select how you want to define the velocity in the **Velocity Specification** drop-down list. The options are:

- **Magnitude and Direction**
  1. Enter the magnitude of the velocity in the **Velocity Magnitude [m s ^-1]** field.
  2. Provide the velocity direction in the **X**, **Y** and **Z** fields.
- **Components**

  Specify the X, Y, and Z velocity components in the **X [m s^-1]**, **Y [m s^-1]**, **Z [m s^-1]** fields.

**Important:**

For better solution stability, it is recommended that you avoid having velocity inlets in direct contact with pressure outlets.

#### 27.1.13.4.2. Inputs for Pressure Outlets

Pressure outlets are used to define the static pressure at flow outlets.

Provide the gauge pressure in the **Gauge Pressure [Pa]** field.

**Important:**

For better solution stability, it is recommended that you avoid having pressure outlets in direct contact with velocity inlets.

#### 27.1.13.4.3. Inputs for Walls

The required inputs vary depending on the **Wall Velocity Specification** method that you select:

- **Stationary**

  No further input is required.
- **Translational**

  Select the method for specifying the translational velocity from the **Translational Velocity Specification** drop-down list.

  - **Magnitude and Direction**
    1. Enter the magnitude of the translational velocity in the **Translational Velocity Magnitude [m s ^-1]** field.
    2. Provide the translational velocity direction in the **X**, **Y** and **Z** fields.

– **Components**

Specify the X, Y, and Z translational velocity components in the **X [m s^-1]**, **Y [m s^-1]**, **Z [m s^-1]** fields.

- **Rotational**

1. Enter the rotational velocity in the **Rotational Speed [radian s^-1]** field.

2. Provide the axis of rotation in the **X**, **Y**, and **Z** fields under **Rotation Axis Origin**.

3. Provide the direction of rotation axis in the **X**, **Y**, and **Z** fields under **Rotation Axis Direction**.

#### 27.1.13.4.4. Inputs for Symmetries

Symmetry boundaries are used when the physical geometry of interest, and the expected pattern of the flow/thermal solution have mirror symmetry.

With **Symmetry** selected for **Boundary Type**, no further inputs are required.

## 27.1.14. Operating Conditions

Operating conditions relate to pressure and it provides a reference pressure. The pressure specified at the pressure outlet is **Gauge Pressure**.

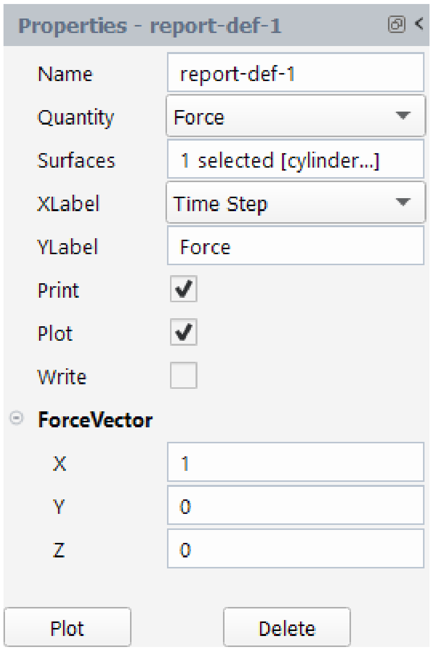
For closed systems without a pressure outlet, you must provide the **Reference Pressure Location** in addition to the **Operating Pressure**.

## 27.1.15. Setting Up Reports

Report definitions allow you to monitor quantities as the solution progresses. Report definitions can be printed to the Console, plotted in the graphics window, and/or written to a file.

To create a new report definition, select **Report Definitions** in the **Outline View** tree (under the **Solution** branch). Right-click **Report Definitions** in the tree and select **New...**, or click **New** in the **Properties - Report Definitions** panel.

**Solution** → **Report Definitions** → **New...**




Many of the setup steps are the same or similar when setting up the various types of report definitions.

To setup a force report definition:

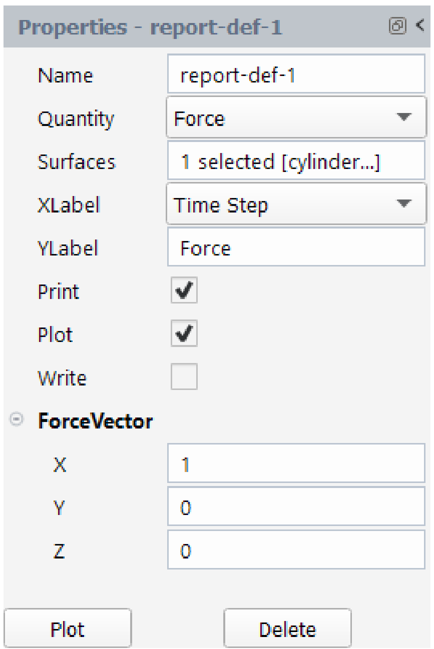
1. (Optional) Provide a **Name** for the report definition.
2. Select **Force** from the **Quantity** drop-down list.
3. Select the surfaces where you want the force calculated. *Click within the **Surfaces** list, which is colored yellow when it is empty.*
4. Choose how you want to present the report data:
   - Enable **Print** to have the results printed to the console.
   - Enable **Plot** to have the data plotted in the graphics window.
   - Enable **Write** to write the results to a file. *Once enabled, the a field appears for providing a name for the report definition file and a **Browse...** button for selecting a storage directory.*
5. (**Plot** enabled) Select the value (and label) for the X axis in the **XLabel** drop-down list.
   - **Time Step**—prints the value of the report definition with respect to the time step, which is retrieved from the solver every 100 time steps.
   - **Flow Time**—prints the value of the report definition with respect to the actual flow time, which is retrieved from the solver every 100 time steps.
6. (**Plot** enabled) Provide the label for the Y axis in the **YLabel** field.
7. Provide the components of the force vector in the **Force Vector** fields.

To delete a report definition, you can right-click it in the tree and select **Delete** or click **Delete** at the bottom of the **Properties - <report-definition-name>**.

You can create report definitions for the following quantities:

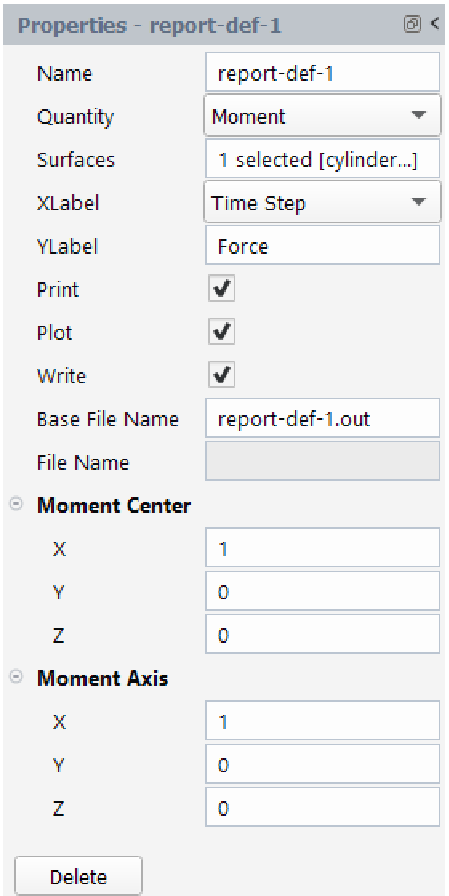


### 27.1.15.1. Force




The procedure for creating force report definitions is covered in Setting Up Reports (p. 169).

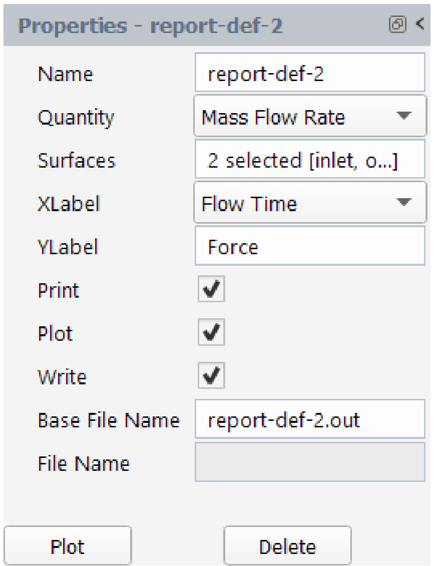
### 27.1.15.2. Moment




The common steps for creating a moment report definition are covered in Setting Up Reports (p. 169), and the moment report definition-specific steps are as follows:

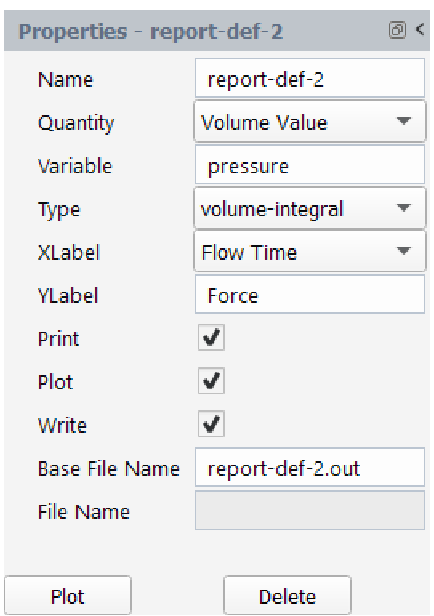
1. Select **Moment** from the **Quantity** drop-down list.
2. Provide the components of the moment center in the **Moment Center** fields.
3. Provide the components of the moment axis in the **Moment Axis** fields.

#### 27.1.15.3. Mass Flow Rate




The common steps for creating a mass flow rate report definition are covered in Setting Up Reports (p. 169), with the only difference being that you must select **Mass Flow Rate** from the **Quantity** drop-down list.

#### 27.1.15.4. Volume Value




The common steps for creating a volume value report definition are covered in Setting Up Reports (p. 169), and the volume value report definition-specific steps are as follows:

1. Select **Volume Value** from the **Quantity** drop-down list.

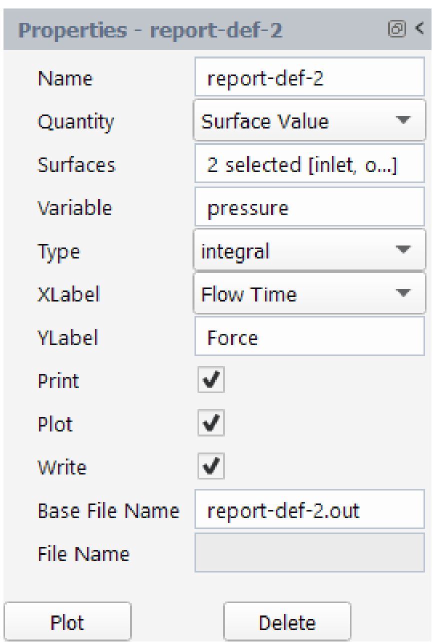
2. Select the specific variable that you want calculated from the **Variable** drop-down list.

   - **pressure**—calculates the gauge pressure.

- **x-velocity**—calculates the velocity in the x-direction.
- **y-velocity**—calculates the velocity in the y-direction.
- **z-velocity**—calculates the velocity in the z-direction.
- **velocity-magnitude**—calculates the magnitude of the velocity.

3. Specify how you want the variable calculated from the **Type** drop down list.
   - **sum**—calculates the sum of the variable values across the model
   - **min**—calculates the minimum variable value across the model.
   - **max**—calculates the maximum variable value across the model.
   - **volume**—computes the volume of the model. This selection ignores the selected **Variable**, as that selection does not affect this computation.
   - **volume-average**—calculates the average value of the variable across the model.
   - **volume-integral**—calculates the net value of the variable across the model.

### 27.1.15.5. Surface Value




The common steps for creating a surface value report definition are covered in Setting Up Reports (p. 169), and the surface value report definition-specific steps are as follows:

1. Select **Surface Value** from the **Quantity** drop-down list.
2. Select the specific variable that you want calculated from the **Variable** drop-down list.
   - **pressure**—calculates the gauge pressure.

- **x-velocity**—calculates the velocity in the x-direction.
- **y-velocity**—calculates the velocity in the y-direction.
- **z-velocity**—calculates the velocity in the z-direction.
- **velocity-magnitude**—calculates the magnitude of the velocity.

3. Specify how you want the variable calculated from the **Type** drop down list.

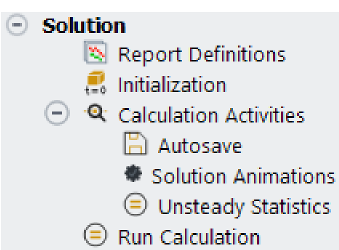
    - **area-average**—calculates the surface area-averaged value of the variable across the model surfaces.
    - **facet-average**—calculates the average value for the variable at the cell face centers.
    - **integral**—calculates the net surface value of the variable across the model.
    - **std-dev**—calculates the standard deviation of the surface value of the selected variable across the model.
    - **sum**—calculates the sum of the surface value of the variable across the model.
    - **vertex-average**—calculates average value for the variable at the cell nodes.

### 27.1.16. Calculating a Solution

The setup required prior to beginning a calculation consists of specifying how you want the solution to proceed, providing initial values for the domain, initializing the flow field (which also creates the mesh), and beginning the calculation.

These settings and operations are all specified in the sub-branches under the **Solution** branch—specifically, the **Initialization**, **Calculation Activities**, and **Run Calculation** branches.

**Figure 27.9: Solution Branch and Sub-Branches**




Further information on the steps required to begin a calculation are covered in the following sections:

27.1.16.1. Initial Flow Field Values
27.1.16.2. Calculation Activities: Autosave, Solution Animations, Unsteady Statistics
27.1.16.3. Run Calculation

#### 27.1.16.1. Initial Flow Field Values

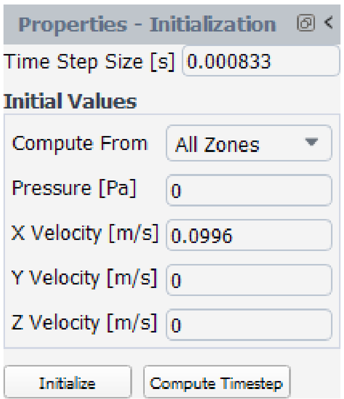
You can provide values for the flow field prior to beginning the calculation, which can help with the speed of the solution and convergence.

To initialize the flow field:

1. Open the initialization properties (**Properties - Initialization**) by selecting **Initialization** in the Outline View tree.

   **Solution** → **Initialization**

   **Figure 27.10: Initialization Properties**




2. Click **Compute Timestep** to have Fluent Lattice Boltzmann provide a **Time Step Size** based on the boundary conditions settings and intialization values. *The timestep size is required for properly initializing the flow field.*

   Alternatively, you can provide your own value for the **Time Step Size**.

3. Set the **Time Step Size**, which is required for properly initializing the flow field. Note that you can click **Compute Timestep** to have the solver provide a reasonable initial value for the timestep size.

4. Select where the values will be computed from in the **Compute From** drop-down list and provide the values, as needed.

   There are three ways that you can provide these initial values:

   - Directly setting the pressure and velocity components. You can do this by entering values for **Pressure**, **X Velocity**, **Y Velocity**, and **Z Velocity**.
   - Computing the pressure and velocity components from all of the zones. You can do this by selecting **All Zones** from the **Compute From** drop-down list.
   - Interpolating data from a finite volume (Ansys Fluent) data file generated on the same mesh. Fluent Lattice Boltzmann will interpolate this data onto the octree mesh. *You must import a data file for this option to appear in the drop-down list (**File** → **Import** → **Fluent Data File...**).*

5. Click **Initialize**.

### 27.1.16.2. Calculation Activities: Autosave, Solution Animations, Unsteady Statistics

With the rest of the case defined initialized as described in Initial Flow Field Values (p. 175), you are almost ready to begin calculating the solution.

1. Review the mesh (generated when you click **Initialize** in **Properties - Initialization**. If you are happy with it, then proceed to the next step. If you want a different mesh, go back to Defining the Domain Parameters for Meshing (p. 158) and update the mesh settings, then click **Initialize**.

2. (Optional) Define activities that you want to occur as the solution progresses, such as automatically saving data files, saving unsteady statistics, and creating animations. Refer to Autosave (p. 177), Solution Animations (p. 178), and Unsteady Statistics (p. 178) for additional information.

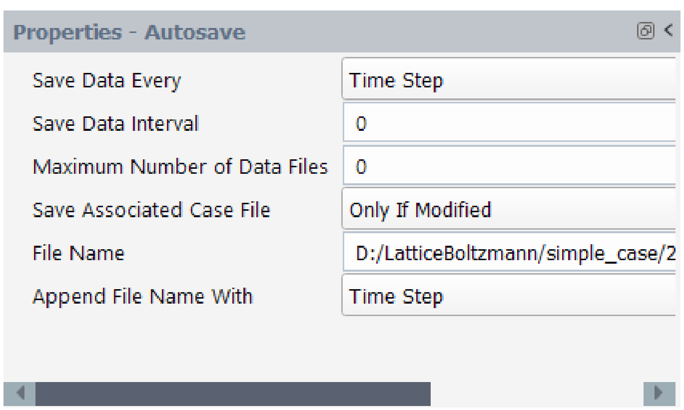
#### 27.1.16.2.1. Autosave

Calculation activities, including auto-saving are only available after you initialize the solution. Autosaving allows you to save case and data files at a specified interval without having to pause the solver and save the case and data files manually.

To enable autosaving:

1. Open the autosave properties (**Properties - Autosave**) by clicking the **Autosave** branch in the Outline View tree (under **Calculation Activities**).

   **Solution** → **Calculation Activities** → **Autosave**




2. Set the method that Fluent Lattice Boltzmann will use to determine when to save case and data files from the **Save Data Every** drop-down list.

   - **Time Step**—Data files and case files are saved at the specified timestep interval. *Note that saving only occurs every reporting interval, which is set as every 100 time steps.*

   - **Flow Time**—Data files and case files are saved at the specified flow time interval. *Note that saving only occurs every reporting interval, which is set as every 100 time steps.*

3. Specify whether you want the case file saved with the data file every time or only if the case is modified using the **Save Associated Case File** drop-down list.

4. Specify the name of the files that will be saved and their location in the **File Name** field. Click **Browse...** to open the **Select File** dialog box and navigate to the desired directory.

5. Chose whether you want the data files appended with the time step or flow time in the **Append File Name With** drop-down list.

6. (**Flow Time** only for appended file name) Enter how many decimal places you want for specifying the exact flow time appended to the data files in the **Number of Decimal Units in File Name** field.

#### 27.1.16.2.2. Unsteady Statistics

With unsteady statistics enabled you can visualize mean and root-mean-square-error (RSME) values of pressure and velocity. These quantities are typically used when looking at truly unsteady flows.

---

**Important:**

- Calculation activities, including unsteady statistics, are only available after you initialize the solution.

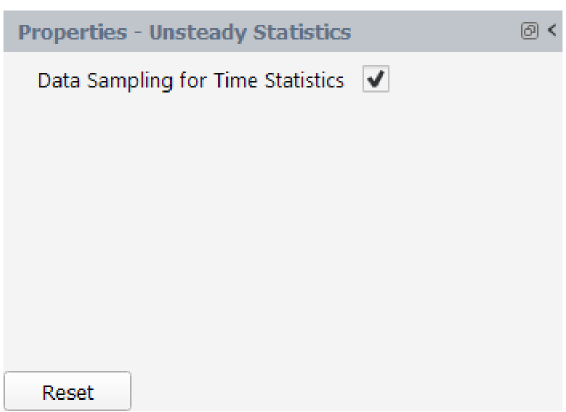
- Unsteady statistics are reset during remeshing.

---

To enable data sampling for unsteady statistics:

1. Open the unsteady statistics properties (**Properties - Unsteady Statistics**) by clicking the **Unsteady Statistics** branch in the Outline View tree (under **Calculation Activities**).

   **Solution** → **Calculation Activities** → **Unsteady Statistics**




2. Enable **Data Sampling for Time Statistics**.

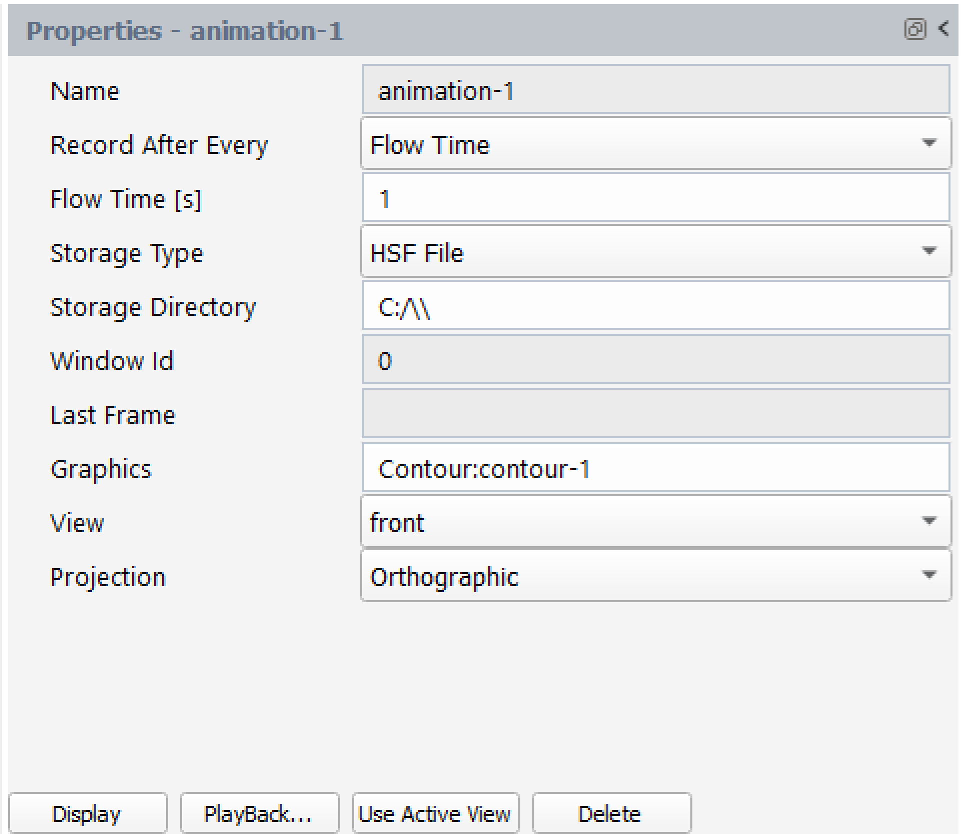
#### 27.1.16.2.3. Solution Animations

Calculation activities, including defining solution animations, are only available after you initialize the solution.

To create :

1. Open the animation properties (**Properties - animation-1** by opening the properties of **Solution Animation** branch in the Outline View tree (under **Calculation Activities**) and clicking **New**.

   **Solution** → **Calculation Activities** → **Solution Animations** → **New**

2. Specify whether you want the animation images saved based on time steps or flow time by selecting **TimeStep** or **Flow Time** from the **Record After Every** drop-down list.

3. Set how frequently you want the animation images saved by providing a value for **Index** or **Flow Time** depending on whether you selected **TimeStep** or **Flow Time** in the **Record After Every** field. *Note that animation images with **Record After Every** set to **TimeStep** are only saved in relation to the **Reporting Interval** specified in **Properties - Run Calculation**.*

4. Specify how you want Fluent Lattice Boltzmann to save the animation frames by selecting **HSF File** (3D image file), **In Memory**, or **PPM Image** (2D image file) from the **Storage Type** drop-down list.

   ---

   **Important:**

   The advantage to saving the animation sequence using the **HSF File** option is that these files are highly compressible and can be viewed and interacted with using a "HOOPS Viewer" application on iOS and Android equipped devices. The "HOOPS Viewer" application can be downloaded to iOS and Android equipped devices from the "App Store" and "Google Play" respectively.

An advantage to saving the animation sequence using the **PPM Image** option is that you can use the separate pixmap image files for the creation of a single GIF file. GIF file creation can be done quickly with graphics tools provided by other third-party graphics packages such as ImageMagick, that is, `animate` or `convert`. For example, if you save the PPM files starting with the string `sequence-2`, and you are using the ImageMagick software, you can use the `convert` command with the `-adjoin` option to create a single GIF file out of the sequence using the following command.

```
convert -adjoin sequence-2_00*.ppm sequence2.gif
```

5. Choose the **Storage Directory** where Fluent Lattice Boltzmann will save the animation frames.

6. Select the graphics object you want to animate by clicking in the **Graphics** field. If you want to animate an object that is not listed, you can create a new object, as described in Graphics Objects (p. 194).

7. Set the view for how the graphics object(s) will appear in the images captured during the simulation run. You can either select a view from the **View** drop-down list or display the object you plan to animate in the graphics window, orient it how you want, then click **Use Active View** at the bottom of the properties panel.

8. (Optional) Click **Display** to review how the animation images will appear once the solver run is complete.

#### Playing an Animation

Once you have reached a solution, you will want to review the animations that Fluent Lattice Boltzmann recorded during the simulation. Click **Playback...** (at the bottom of the **Properties - <animation-#>** panel) to open the **Playback** dialog box and play your animation(s). Refer to Playing an Animation Sequence in the *Fluent User's Guide* for additional information on using this dialog box.

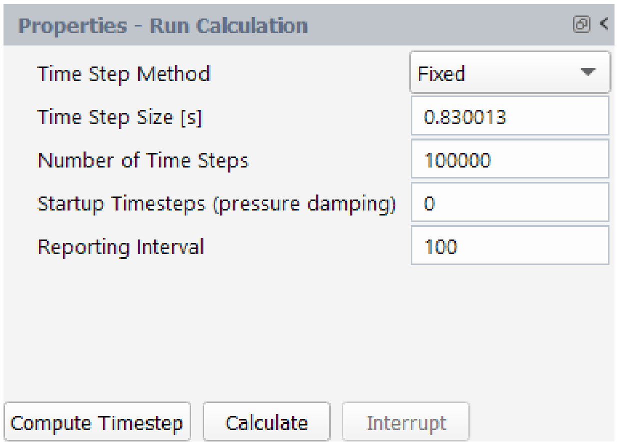
### 27.1.16.3. Run Calculation

After initializing and setting up your calculation activities, you can specify the settings for running the calculation.

1. Open the run calculation properties (**Properties - Run Calculation**) by selecting **Run Calculation** in the Outline View tree.

   **Solution** → **Run Calculation**

**Figure 27.11: Run Calculation Properties**




2. (Optional) Specify the number of **Startup Timesteps (pressure damping)** which can help to reduce early solution instability as the solver begins calculating.

3. (Optional) Enter a different value for the **Reporting Interval**. *Note that decreasing the reporting interval may increase the time to reach a converged solution.*

4. Click **Calculate**.

---

**Note:**

You can click **Interrupt** to pause the calculation at the end of the current reporting interval. Clicking **Calculate** again resumes the calculation.

---

## 27.1.17. Postprocessing Results

You can review your results in the graphics window using objects such as contours, vectors, pathlines, particle tracks and XY plots. You can also create surfaces for further exploring the results. *Note that you can begin postprocessing while the solver is running.*

---

**Important:**

- Y plus (y+) values are only saved in memory (not in the data file) and are therefore only available within the same session where the solution is computed.
- **Wall Y plus** and **Wall Fluxes** can only be computed on walls (not on user defined surfaces such as planes and points).

---

These different postprocessing tools are generally distributed into three groups: surfaces, views, and graphics objects, and they are discussed in the following sections:

### 27.1.17.1. Surfaces

You can create different types of surfaces to visualize your results, including points, lines, rakes, planes, and iso-surfaces. Note that these surfaces are "clipped", which means that they closely follow the boundaries of the model, rather than directly showing the jagged edges of the octree mesh. Values are interpolated to the smoothed out boundary lines.

---

**Note:**

To ensure updated surface definitions (for example, if you change the location of a line) are properly shown when included in a scene or other graphics object, display the surface before re-displaying the containing object.

---

Refer to the following sections for more details:



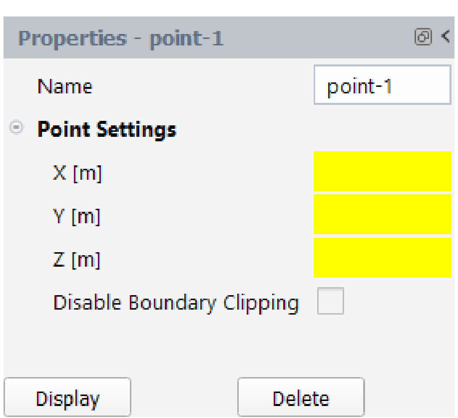
#### 27.1.17.1.1. Point Surfaces

You may be interested in displaying results at a single point in the domain. For example, you may want to monitor the value of some variable or function at a particular location. To do this, you must first create a "point" surface, which consists of a single point. When you display node-value data on a point surface, the value displayed is a linear average of the neighboring node values. If you display cell-value data, the value at the cell in which the point lies is displayed.

**Results → Surfaces** **New → Point**

To create a point surface, use the Figure 27.12: Properties of a Point Surface (p. 182).

**Figure 27.12: Properties of a Point Surface**

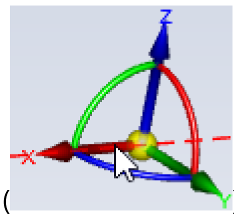


1. (Optional) Provide a name for the point surface if you do not want to use the default name.

   **Important:**

   The surface name that you provide must begin with an alphabetical letter. If the name of the surface begins with any other character or number, Fluent Lattice Boltzmann rejects the entry.

2. Specify the location of the point. There are two different ways to do this:

   - Enter the coordinates (**x, y, z**) under **Point Settings**.
   - Move the point tool () to the location in the domain where you want the point surface created. Refer to Using the Point Tool in the *Fluent User's Guide* for additional information.

3. (Optional) By default points are created at the closest boundary to the specified location. However, you can disable **Boundary Clipping** to have the point located at the exact coordinates specified, even if they do not fall precisely on a mesh boundary.

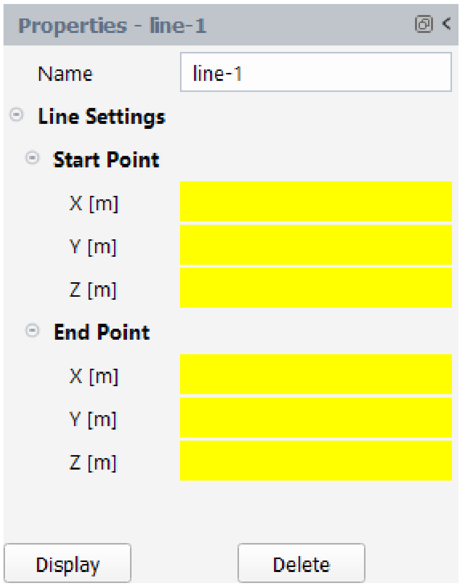
#### 27.1.17.1.2. Line Surfaces

A line is simply a line that extends up to and includes the specified endpoints; data points will be located where the line intersects the faces of the cell, and consequently may not be equally spaced.

**Results** → **Surfaces** **New** → **Line**

To create a line surface, use the Figure 27.13: Properties of a Line Surface (p. 184).

**Figure 27.13: Properties of a Line Surface**




1. (Optional) Provide a name for the line surface if you do not want to use the default name.

> **Important:**
>
> The surface name that you provide must begin with an alphabetical letter. If the name of the surface begins with any other character or number, Fluent Lattice Boltzmann rejects the entry.

2. Specify the location of the **Start Point**.
3. Specify the location of the **End Point**.

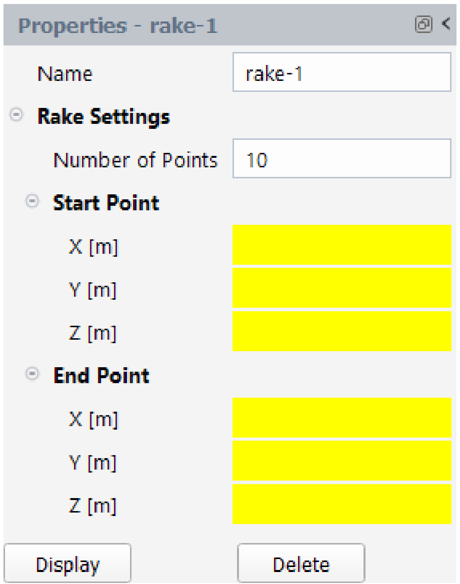
### 27.1.17.1.3. Rake Surfaces

A rake consists of a specified number of points equally spaced between two specified endpoints.

**Results** → **Surfaces** **New** → **Rake**

To create a rake surface, use the Figure 27.14: Properties of a Rake Surface (p. 185).

**Figure 27.14: Properties of a Rake Surface**




1. (Optional) Provide a name for the rake surface if you do not want to use the default name.

> **Important:**
>
> The surface name that you provide must begin with an alphabetical letter. If the name of the surface begins with any other character or number, Fluent Lattice Boltzmann rejects the entry.

2. Set the **Number of Points** to include in the rake surface.

3. Specify the location of the **Start Point**.

4. Specify the location of the **End Point**.

#### 27.1.17.1.4. Plane Surfaces

To display flow-field data on a specific plane in the domain, you will use a plane surface. You can create surfaces that cut through the solution domain along arbitrary planes.

> **Note:**
>
> Planes that are co-planar with a domain boundary may not appear as expected due to the clipping process for ensuring smooth boundaries.

There are three types of plane surfaces that you can create:

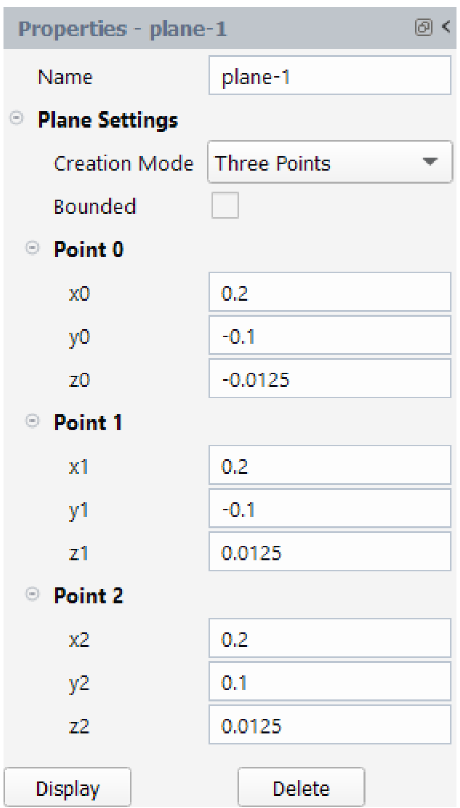
- Coordinate system-based—the plane is created in the YX, ZX, or XY directions, bounded by the extents of the domain. You can move the plane to the desired location in the domain using the plane tool. For example, if you are using the **YZ Plane** method, you can drag the plane in the (+) or (-) X direction.

- Point and normal—the plane orientation is determined by selecting a point and specifying a direction normal to that point. The extents of the plane are the edges of the domain. You have the option to control the orientation of the plane using the plane tool or you can compute the normal from a surface.
- Three points—the plane orientation and extents are bounded by three points that you can select. You also have the option to manipulate the points and orientation of the plane directly in the graphics window using the plane tool.

**Results → Surfaces New → Plane**

To create a plane surface, use the Figure 27.15: Properties of a Plane Surface (p. 186).

**Figure 27.15: Properties of a Plane Surface**




The procedure for creating the plane surface varies depending on the method you want to use.

### Point and Normal

1. (Optional) Provide a name for the plane.

---

**Important:**

The surface name that you enter must begin with an alphabetical letter. If the name of the surface begins with any other character or number, Fluent Lattice Boltzmann rejects the entry.

---

2. Drag the plane tool to the desired location, as shown in The Plane Tool for the Point and Normal Method in the *Fluent User's Guide*.

   *Alternatively, you can specify **Point 0** and the **Normal** by providing the coordinates of each.*

### Bounded

1. (Optional) Provide a name for the plane.

   **Important:**

   The surface name that you enter must begin with an alphabetical letter. If the name of the surface begins with any other character or number, Fluent Lattice Boltzmann rejects the entry.

2. Drag the plane tool to the desired location, as shown in The Plane Tool for the Three Points Method in the *Fluent User's Guide*.

   *Alternatively, you can specify **Point 0**, **Point 1**, and **Point 2** by providing the coordinates.*

### Three Points

1. (Optional) Provide a name for the plane.

   **Important:**

   The surface name that you enter must begin with an alphabetical letter. If the name of the surface begins with any other character or number, Fluent Lattice Boltzmann rejects the entry.

2. Drag the plane tool to the desired location, as shown in The Plane Tool for the Three Points Method in the *Fluent User's Guide*.

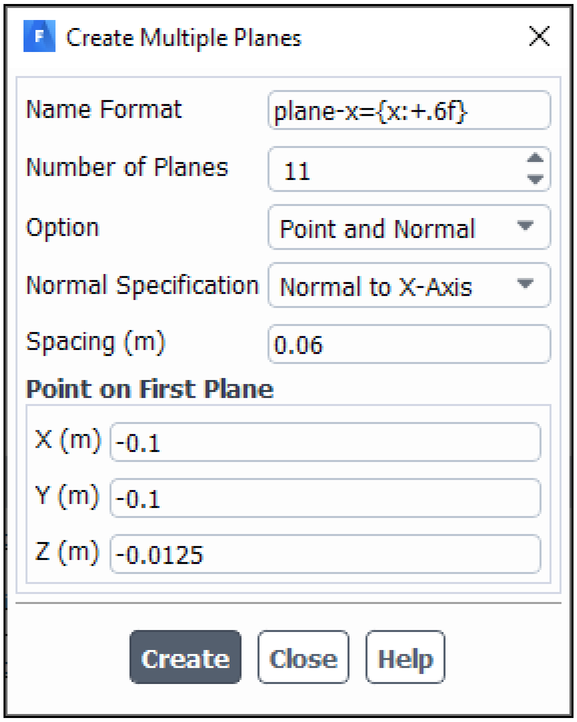
   *Alternatively, you can specify **Point 0**, **Point 1**, and **Point 2** by providing the coordinates.*

### Multiple Planes

Instead of creating plane surfaces one at a time as described above, you have the option to create multiple plane surfaces at once using the Figure 27.16: Create Multiple Planes Dialog Box (p. 188).

**Results** → **Surfaces** → **Create Multiple Planes...**

**Figure 27.16: Create Multiple Planes Dialog Box**




To use the **Create Multiple Planes** dialog box:

1. (Optional) Provide a format for naming the plane surfaces in the **Name Format** field.

2. Specify how many planes will be created in the **Number of Planes** field.

3. Select the method for how you want to create the planes in the **Option** drop-down list.

   - **Point and Normal**—similarly to the process described earlier in this section, you must provide a point and the direction normal to that point to define the first plane.

   - **First and Last Point**—you define the coordinates for the first and last point, which determines the orientation of the planes. The spacing is determined by how many planes you specify in the **Number of Planes** field.

4. (**Point and Normal** only) Specify the direction normal (perpendicular) to the plane in the **Normal Specification** drop-down list.

5. (**Point and Normal** only) Specify how far apart the planes are from each other in the **Spacing** field.

6. Provide the coordinate for the location of the first plane in the **Point on First Plane** group box.

7. (**First and Last Point** only) Provide the coordinates for the location of the last plane in the **Point on Last Plane** group box.

8. Click **Create** to create the new plane surfaces.

The new plane surfaces created using the **Create Multiple Planes** dialog box are added to the Outline View tree under the **Surfaces** branch and are now eligible for editing individually. *Once created, the multiple planes cannot be edited as a group.*

### 27.1.17.1.5. Iso-Surfaces

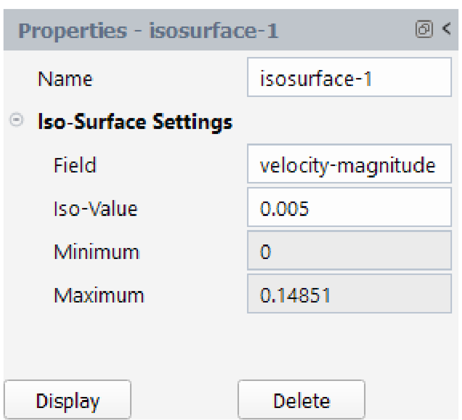
If you want to If you want to display results on cells that have a constant value for a specified variable, you will need to create an iso-surface of that variable. Generating an iso-surface based on $x$ , $y$ , or $z$ coordinate, for example, will give you an $x$ , $y$ , or $z$ cross-section of your domain; generating an iso-surface based on pressure will enable you to display data for another variable on a surface of constant pressure. You can create an iso-surface from an existing surface or from the entire domain. Furthermore, you can restrict any iso-surface to a specified cell zone.

> **Important:**
>
> Note that you cannot create an iso-surface until you have initialized the solution, performed calculations, or read a data file.

**Results → Surfaces New → Iso-Surface**

To create an iso-surface, use the Figure 27.17: Properties of an Iso-Surface (p. 189).

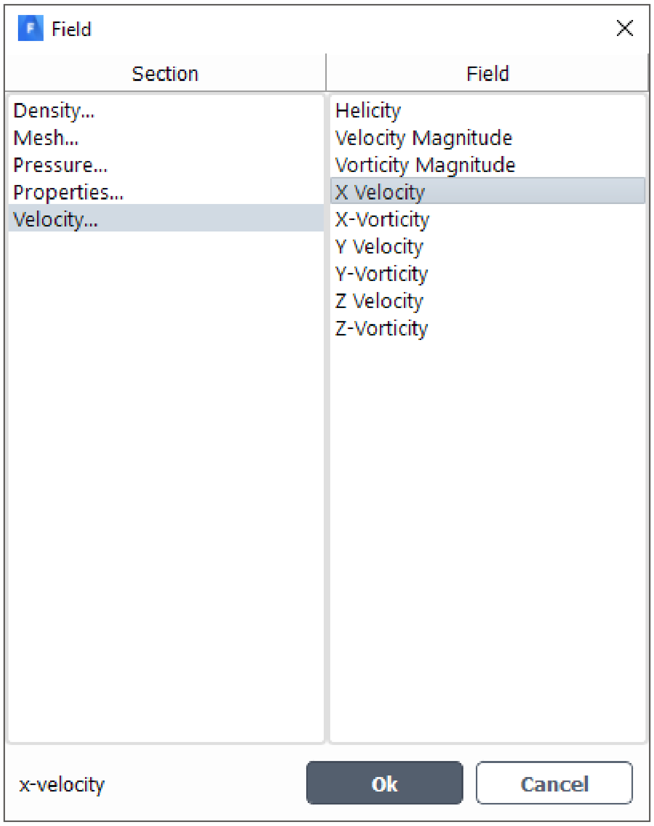
**Figure 27.17: Properties of an Iso-Surface**




1. (Optional) Provide a name for the iso-surface.

> **Important:**
>
> The surface name that you enter must begin with an alphabetical letter. If the name of the surface begins with any other character or number, Fluent Lattice Boltzmann rejects the entry.

2. Click **Field** to open the **Field** dialog box and specify the field you want to view.




3. Click **OK** to confirm your selected field and close the dialog box.

4. Enter the iso-values into the **Iso Value** field. You can enter multiple iso-values into the field separated by spaces.

5. Set the range by specifying the **Minimum** and **Maximum**.

Click **Delete** at the bottom of the iso-surface properties to delete an iso-surface.

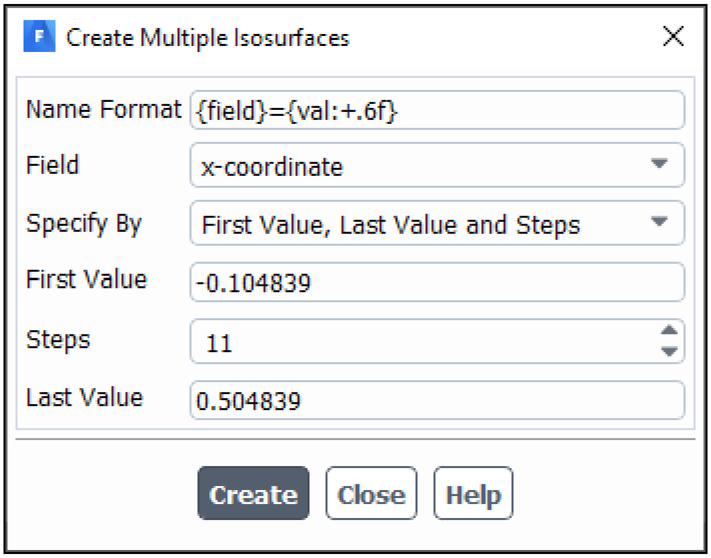
Instead of creating plane surfaces one at a time as described above, you have the option to create multiple plane surfaces at once using the Figure 27.16: Create Multiple Planes Dialog Box (p. 188).

### Create Multiple Iso-Surfaces

Instead of iso-surfaces one at a time as described above, you have the option to create multiple iso-surfaces at once using the Figure 27.18: Create Multiple Iso-Surfaces Dialog Box (p. 191).

**Results → Surfaces Create Multiple Iso-surfaces...**

**Figure 27.18: Create Multiple Iso-Surfaces Dialog Box**




To use the **Create Multiple Iso-Surfaces** dialog box:

1. (Optional) Provide a format for naming the iso-surfaces in the **Name Format** field.

2. Select the field that you want to use for creating the iso-surfaces from the **Field** drop-down list:

   - **rmse-y-velocity**—root mean square error of the y-component of the velocity.
   - **rmse-x-velocity**—root mean square error of the x-component of the velocity.
   - **rmse-pressure**—root mean square error of gauge pressure.
   - **mean-y-velocity**—average of the y-component of the velocity.
   - **mean-velocity-magnitude**—average of the velocity magnitude.
   - **mean-pressure**—average gauge pressure.
   - **y-coordinate**—value of the y-coordinate.
   - **mean-z-velocity**—average of the z-component of the velocity.
   - **total-pressure**—value of the total gauge pressure.
   - **z-coordinate**—value of the z-coordinate.
   - **axial-coordinate**—value of the axial-coordinate.
   - **rmse-z-velocity**—root mean square error of the z-component of the velocity.
   - **y-vorticity**—y-component of the vorticity.
   - **z-velocity**—z-component of the velocity.
   - **velocity-magnitude**—magnitude of the velocity.

- **x-coordinate**—value of the x-coordinate.
- **z-vorticity**—z-component of the vorticity.
- **helicity**—the combination of spin and velocity of the flow.
- **vorticity-mag**—magnitude of the vorticity.
- **cell-volume**—value of the mesh cell volume.
- **viscosity-lam**—laminar viscosity.
- **x-velocity**—x-component of the velocity.
- **y-velocity**—y-component of the velocity.
- **mean-x-velocity**—average of the x-component of the velocity.
- **dynamic-pressure**—value of the dynamic pressure.
- **x-vorticity**—x-component of the vorticity.
- **angular-coordinate**—value of the angular coordinate.
- **absolute-pressure**—value of the absolute pressure.
- **pressure**—value of the pressure.
- **rmse-velocity-magnitude**—root mean square error of the velocity magnitude.
- **abs-angular-coordinate**—absolute value of the angular coordinate.
- **density**—value of the density.

3. Specify the method you want to use for creating iso-surfaces in the **Specify By** drop-down list.

   - **First Value, Last Value and Steps**—using this method you must specify the **First Value** for the quantity that you selected in the **Field** drop-down list, specify how many iso-surfaces you want created by entering the number of **Steps**, and provide the final value for the selected quantity in the **Last Value** field.
   - **First Value, Last Value and Increment**—using this method you must specify the **First Value** for the quantity that you selected in the **Field** drop-down list, specify the size of the increments between the first and last value, which determines the number of iso-surfaces to be created, and provide the final value for the selected quantity in the **Last Value** field.
   - **First Value, Increment and Steps**—using this method you must specify the **First Value** for the quantity that you selected in the **Field** drop-down list, specify the size of the increments from the first value, and provide the total number of steps in the **Steps** field, which determines the total number of iso-surfaces to be created.
   - **Last Value, Decrement and Steps**—using this method you must specify the size of the decrement (negative increment) in the **Decrement** field, which will go backwards from the **Last Value**, provide the total number of steps in the **Steps** field, which determines the

total number of iso-surfaces, and provide the final value for the selected quantity in the **Last Value** field.

4. Click **Create** to create the new iso-surfaces.

The new iso-surfaces created using the **Create Multiple Iso-Surfaces** dialog box are added to the Outline View tree under the **Surfaces** branch and are now eligible for editing individually. *Once created, the multiple iso-surfaces cannot be edited as a group.*

### 27.1.17.2. Views

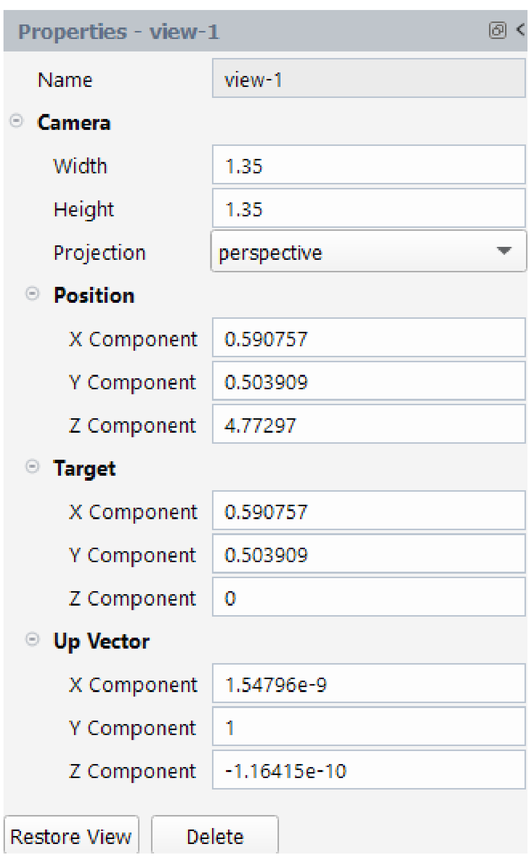
You can view your model and results from various angles in the graphics window. One way to change the view is by using your mouse in coordination with the graphics tools such as rotate ( ) and zoom ( ). Another is by using one of the predefined views.

You can use a predefined view by right-clicking it in the Outline View tree and selecting **RestoreView**.

**Results** → **Views** → **<Predefined-View>** → **RestoreView**

You can create a custom view by:

1. Right-click **Views** in the Outline View tree (under **Results**) and select **New...**.




2. (Optional) Provide a **Name** for this view.

3. Enter the **Width** and **Height** for the view under **Camera**.

4. Specify if the viewing angle is **perspective** or **orthographic** from the **Projection** drop-down list.

5. Specify the position of the view by entering the **X Component**, **Y Component**, and **Z Component** under **Position**.

6. Specify the target of the view by entering the **X Component**, **Y Component**, and **Z Component** under **Target**.

7. Set the orientation of the view by entering `1` for the positive vertical direction of the view in either the **X Component**, **Y Component**, or **Z Component** field under **Up Vector**.

8. Click **Restore View** at the bottom of the view properties panel to display the view.

You can delete a view by right-clicking it in the Outline View tree and selecting **Delete**.

### 27.1.17.3. Graphics Objects

Postprocessing graphics objects are available for visualizing the results of your Fluent Lattice Boltzmann simulation. You can display the mesh, contours, vectors, pathlines, and scenes. Scenes allow you to combine multiple graphics objects within a single graphics window.

---

**Note:**

Only field variables that are appropriate for Fluent Lattice Boltzmann simulations and compatible with octree meshes are available in the **Field** dialog box. Thus, Fluent Lattice Boltzmann offers a reduced, targeted set of field variables when compared with the general purpose Fluent solver.

---

The available graphics objects are described in greater detail in the following sections:



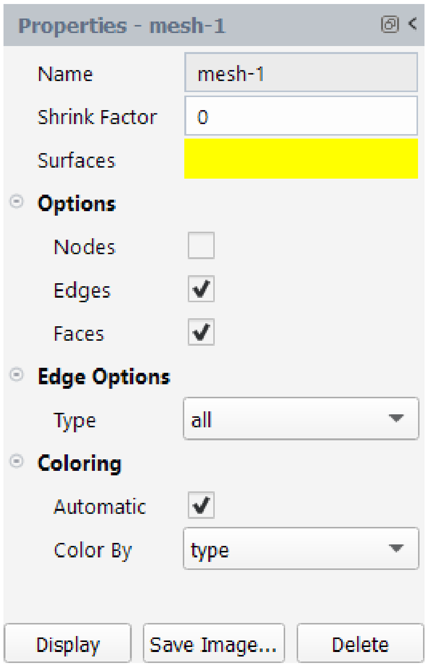
#### 27.1.17.3.1. Mesh Plots

Mesh plots allow you to visualize and inspect the mesh.

To create a mesh plot:

1. Right-click **Meshes** in the Outline View tree and select **New...**.

   **Results → Graphics → Meshes New...**




2. (Optional) Enter a **Name** for the mesh object.

3. (Optional) Enter a **Shrink Factor**.

4. Select the surfaces where you want the mesh displayed by clicking in the yellow **Surfaces** field, selecting the desired surfaces in the **Surfaces** dialog box and clicking **OK**.

5. Select the desired mesh **Options**:

   - **Nodes**—displays the mesh nodes.
   - **Edges**—displays the mesh edges.
   - **Faces**—displays the mesh faces.

6. Set which edges you want displayed from the **Type** drop-down list:

   - **all**—displays all of the edges.
   - **feature**—displays only the features and outline of the mesh.
   - **outline**—displays only the outline of the mesh.

7. Specify how you want the mesh to be colored:

   - **Automatic**—automatically colors the mesh based on the selection in the **Color by** drop-down list (either by **type** or **id**).
   - Manually (**Automatic** disabled)—allows you to set a color for the mesh faces and edges in the **Color Faces By** and **Color Edges By** drop-down lists.

8. Click **Display**.

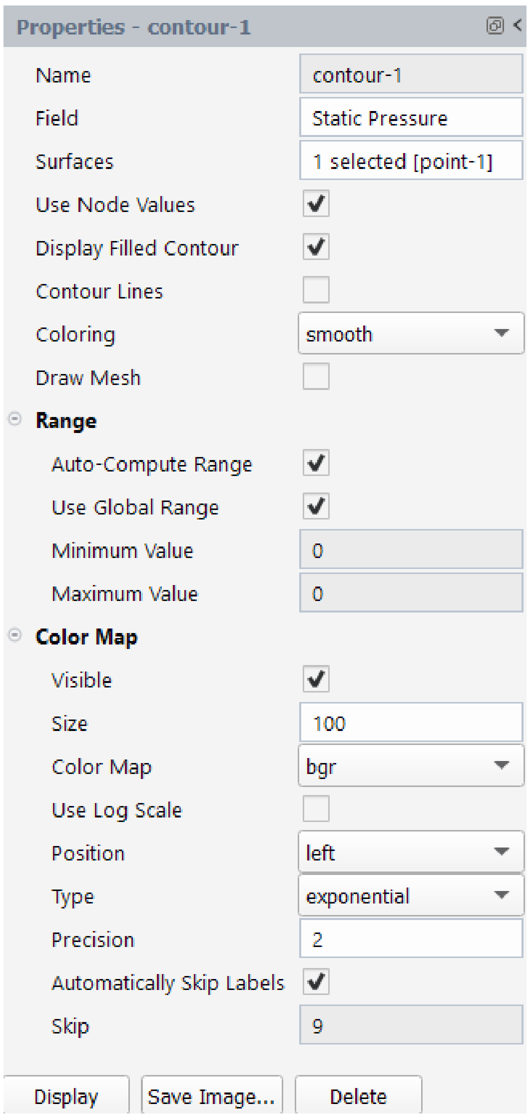
### 27.1.17.3.2. Contour Plots

Contour plots are a valuable postprocessing tool that allow you to use color to represent the values of the specified field variable on the selected surfaces.

To create a contour plot:

1. Right-click **Contours** in the Outline View tree and select **New...**.

   **Results** → **Graphics** → **Contours** → **New...**




2. (Optional) Enter a **Name** for the contour object.

3. Specify the field variable you want to display by clicking in the entry field for **Field**, selecting the variable from the **Field** dialog box and clicking **OK**.

4. Select the surfaces where you want the contours displayed by clicking in the yellow **Surfaces** field, selecting the desired surfaces in the **Surfaces** dialog box and clicking **OK**.

5. Select the desired contour options:

    - **Use Node Values**—displays the values at the nodes.
    - **Display Filled Contour**—displays contours that are fully colored.
    - **Contour Lines**—displays lines on the plot corresponding to the gradations in the colormap.
    - **Coloring**—lets you specify whether contours are **banded** or **smooth**.
    - **Draw Mesh**—displays whichever mesh you select in the **Overlayed Mesh** drop-down list, which includes the outline and any defined mesh objects.

6. Set the **Range** for the contour plot:

    - **Auto-Compute Range**—automatically calculates the range for the contour colormap. When disabled, it allows you to clip the range to a specified **Minimum** and **Maximum**.
    - **Use Global Range**—confirms that you are using the automatically computed range. However, if you do not want to use this range, you must disable **Auto-Compute Range**.
    - **Minimum Value** and **Maximum Value**—allows you to set the range for the contour plot values. *These fields only applies when you disable **Auto-Compute Range**.*

7. Set the **Color Map** fields:

    - **Size**—determines the size of the colormap.
    - **Color Map**—specifies the colors to be used in the colormap display.
    - **Use Log Scale**—uses a logarithmic scale.
    - **Position**—determines where the colormap is located in the graphics window.
    - **Type**—lets you set whether the colormap follows a general, float, or exponential scale.
    - **Precision**—lets you set the number of significant digits for the colormap.
    - **Skip**—allows you to set a spacing on the colormap scale.

8. Click **Display**.

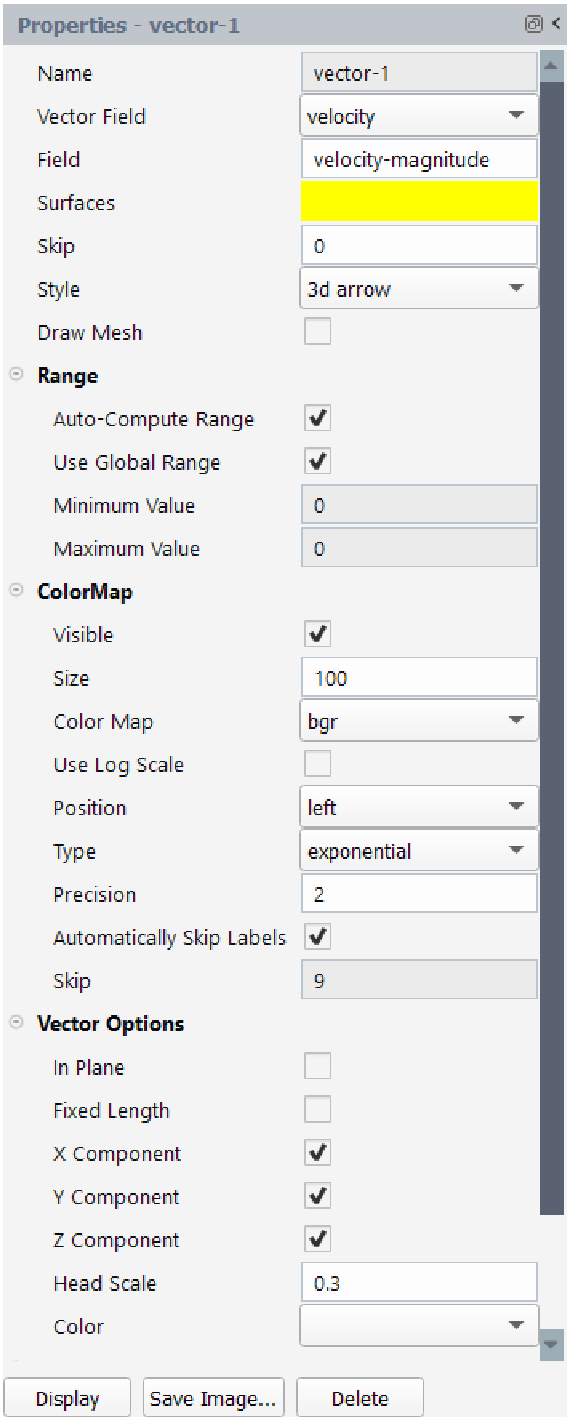
#### 27.1.17.3.3. Vector Plots

You can draw vectors in the entire domain, or on selected surfaces. By default, one vector is drawn at the center of each cell (or at the center of each facet of a data surface), with the length and color of the arrows representing the velocity magnitude . The spacing, size, and coloring of the arrows can be modified, along with several other vector plot settings. Velocity vectors are the default, but you can also plot vector quantities other than velocity. Note that cell-center values are always used for vector plots; you cannot plot node-averaged values.

To create a vector plot:

1. Right-click **Vectors** in the Outline View tree and select **New...**.

**Results** → **Graphics** → **Vectors** → **New...**




2. (Optional) Enter a **Name** for the vector object.

3. Specify the field variable you want to use to color the vectors by clicking in the entry field for **Field**, selecting the variable from the **Field** dialog box and clicking **OK**.

4. Select the surfaces where you want the vectors displayed by clicking in the yellow **Surfaces** field, selecting the desired surfaces in the **Surfaces** dialog box and clicking **OK**.

5. Set the **Range** for the vector plot:

   - **Auto-Compute Range**—automatically calculates the range for the vector colormap. When disabled, it allows you to clip the range to a specified **Minimum** and **Maximum**.
   - **Use Global Range**—confirms that you are using the automatically computed range. However, if you do not want to use this range, you must disable **Auto-Compute Range**.
   - **Minimum Value** and **Maximum Value**—allows you to set the range for the vector plot values. *These fields only applies when you disable* ***Auto-Compute Range****.*

6. Set the **ColorMap** fields:

   - **Size**—determines the size of the colormap.
   - **Color Map**—specifies the colors to be used in the colormap display.
   - **Use Log Scale**—uses a logarithmic scale.
   - **Position**—determines where the colormap is located in the graphics window.
   - **Type**—lets you set whether the colormap follows a general, float, or exponential scale.
   - **Precision**—lets you set the number of significant digits for the colormap.
   - **Skip**—allows you to set a spacing on the colormap scale.

7. Select the desired vector options:

   - **In Plane**—toggles the display of vector components in the plane of the selected surface. This feature is used for visualizing components that are normal to the flow.
   - **Fixed Length**—sets all the vectors to the same length.
   - **X Component**—toggles the display of the X component of vectors.
   - **Y Component**—toggles the display of the Y component of vectors.
   - **Z Component**—toggles the display of the Z component of vectors.
   - **Head Scale**—allows you to control the size of the vector arrow head in relation to the length of the vector.
   - **Color**—lets you specify a single color for all vectors.

8. Click **Display**.

### 27.1.17.3.4. Line Integral Convolution Plots (LICs)

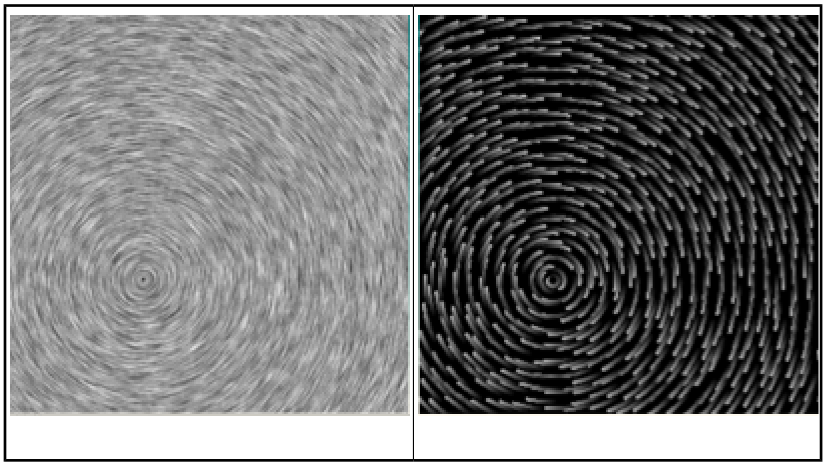
Line Integral Convolution (LIC) is a technique of using a flow field to blur or smear (convolve) a texture image mapped to a surface in the flow domain to provide a visual representation of the vector field on that surface. This results in an effect that is similar to a static picture of the flow field for moving smoke or fluid injected into the fluid. Generally, the integration is done using the flow velocity field, but you can use any other vector field.

There are two different types of LIC visualizations supported in Fluent Lattice Boltzmann; Standard LIC and Oriented LIC. The general methodology is the same for both types, but each uses different initial textures, on which the integration and convolution is performed. This results in very different visualization effects for the vector field, as shown in Table 27.1: Standard LIC (left) and Oriented LIC (right) (p. 200).

- Standard LIC uses an isotropic convolution and has a dense texture that is analogous to higher concentrations of smoke in the flow. It can show flow lines but not the specific orientation. Left to right flow will look the same as right to left flow. It can still be useful when the flow direction can easily be determined by geometric context.

- Oriented LIC uses an anisotropic convolution and a more sparse texture. This gives a result that shows both the flow direction and orientation and is similar to traditional short-length streamline or pathline visualizations.

**Table 27.1: Standard LIC (left) and Oriented LIC (right)**



---

**Note:**

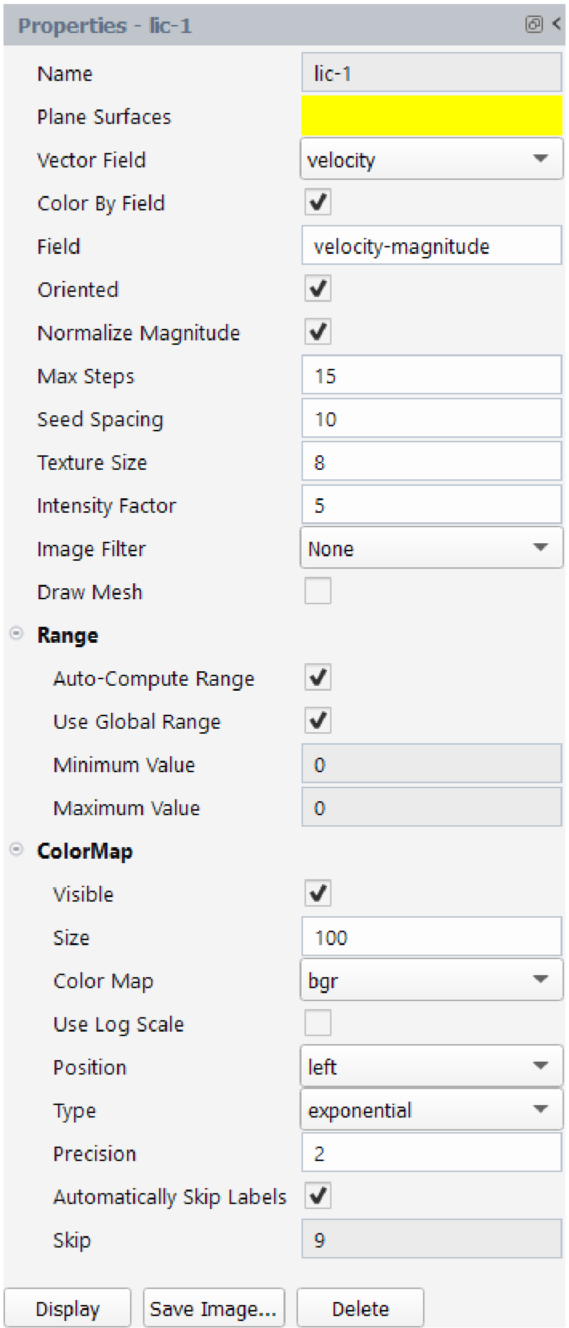
Before creating an LIC plot, you must create a plane surface for the LIC plot. Refer to Plane Surfaces (p. 185) for additional information on creating plane surfaces.

---

To create an LIC plot:

1. Right-click **LICs** in the Outline View tree and select **New...**.

   **Results** → **Graphics** → **LICs** → **New...**




2. (Optional) Enter a **Name** for the LIC object.

3. Select the vector field to use in the integration convolution from the **Vector Field** drop-down list.

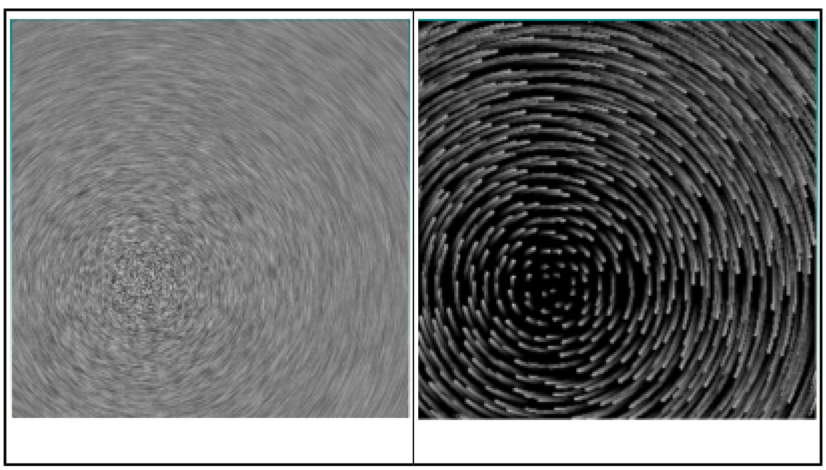
4. Select the surfaces where you want the LIC results displayed by clicking in the yellow **Surfaces** field, selecting the desired surfaces in the **Surfaces** dialog box and clicking **OK**.

**Note:**

LIC plots can only be displayed on flat plane surfaces. Refer to Plane Surfaces (p. 185) for additional information on creating plane surfaces.

5. (Optional) Enable **Color By Field** to have the color change based on the value of the field instead of just being displayed in a single color.
6. (If **Color By Field** is enabled) Specify the field variable you want to use to color the LIC results by clicking in the entry field for **Field**, selecting the variable from the **Field** dialog box and clicking **OK**.
7. (If **Color By Field** is disabled) Specify the color of the LIC results by clicking in the **Lic Color** field, selecting a color in the **Select Color** dialog box, and clicking **OK**.
8. (Optional) Disable **Oriented** to use Standard LIC calculations instead of the Oriented LIC calculations (see Table 27.1: Standard LIC (left) and Oriented LIC (right) (p. 200) for an example of how this impacts the display).
9. (Optional) Disable **Normalize Magnitude** to skip the normalization fo the vector field magnitude druing the integration. The effectiveness of the convolution can very sensitive to the vector field magnitude, so normalization is typically done to maintain a clear definition of the flow field and in particular, the location of singularities. *This truer for the Standard LIC method, so it is recommended that you only disable **Normalize Magnitude** when **Oriented** is enabled.*

**Table 27.2: Normalize Magnitude Disabled for Standard LIC (left) and Oriented LIC (right)**



10. Specify the maximum number of integration steps performed during the convolution. The **Max Steps** value is used independently for both the forward and backward integration, so the total number of steps for each convolution calculation can be double the number specified.
11. (If **Oriented** is enabled) Set the **Seed Spacing** to control the density of the texture seed points. The number specified is the general number of pixels separating the placement of the

seed points prior to integration. The actual displayed density will be controlled by both this field and the **Texture Size** field.

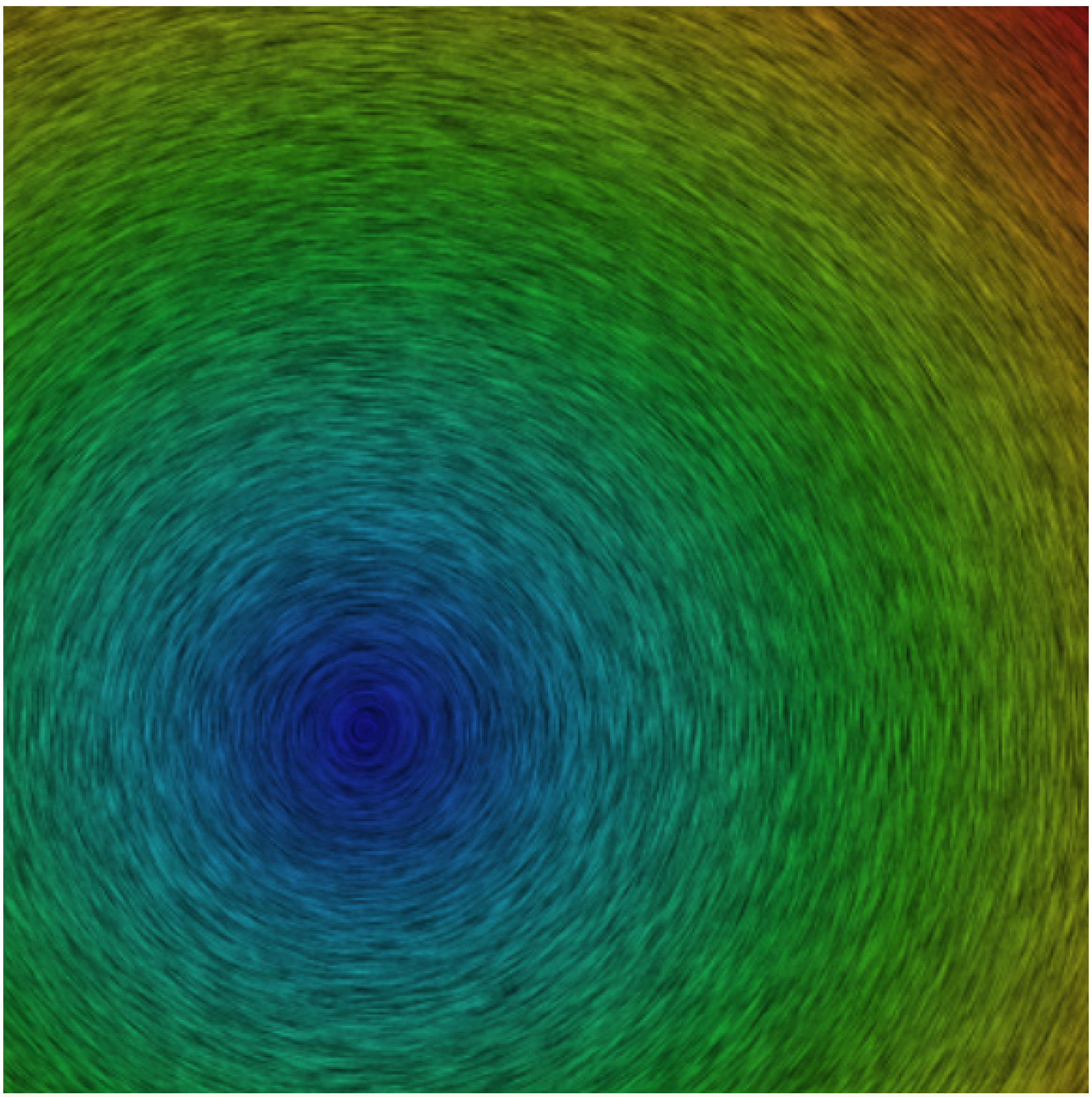
12. Set the **Texture Size** field to control the size (in 100's of pixels) of the texture image used in the convolution.

13. Use the **Intensity Factor** to scale the intensity levels of the convolution results up or down. You can use this to control the brightness or contrast of the resulting image.

14. (Optional) Select a **Image Filter** from the drop-down list. There are four filter options that can enhance or alter the resulting image texture. Image filters will tend to have more of an effect on the denser Standard LIC results than they will on the Oriented LIC results.

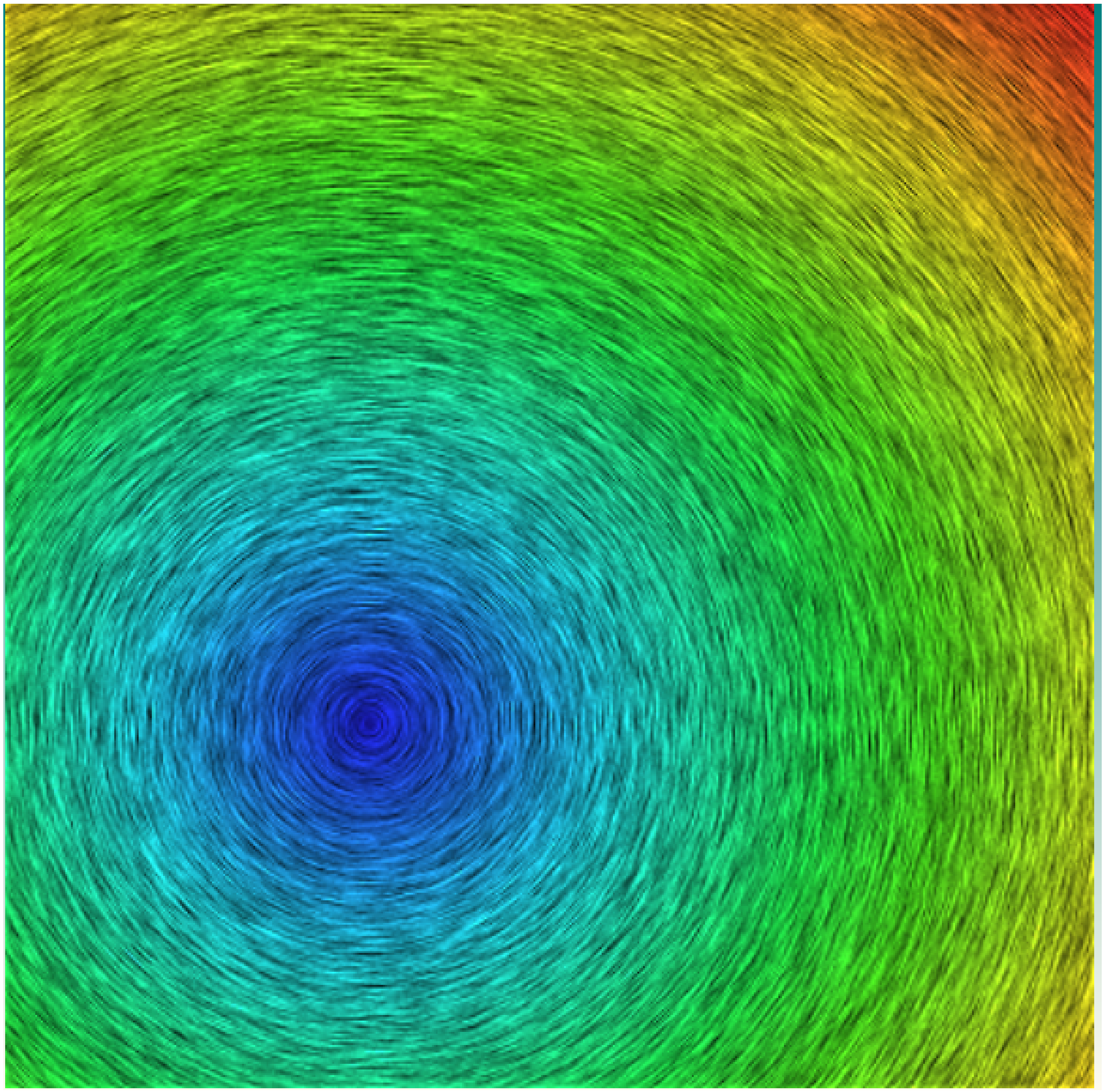
    - **None**—no filter applied.

    **Figure 27.19: Example with the Image Filter Set to None**

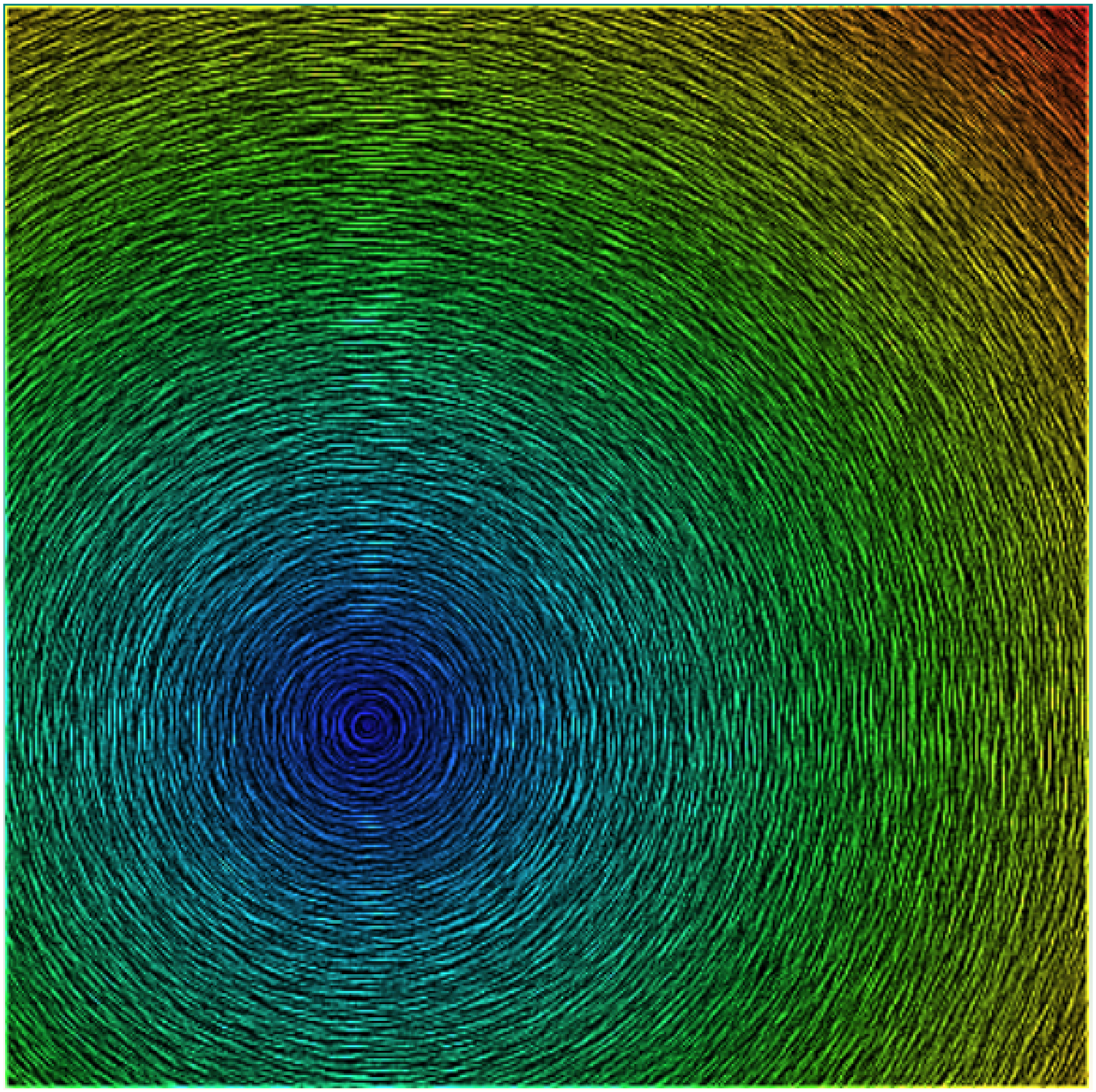
    - **Mild Sharpen**—provides a mild increase in the focus or contrast of the texture image.

**Figure 27.20: Example with the Image Filter Set to Mild Sharpen**



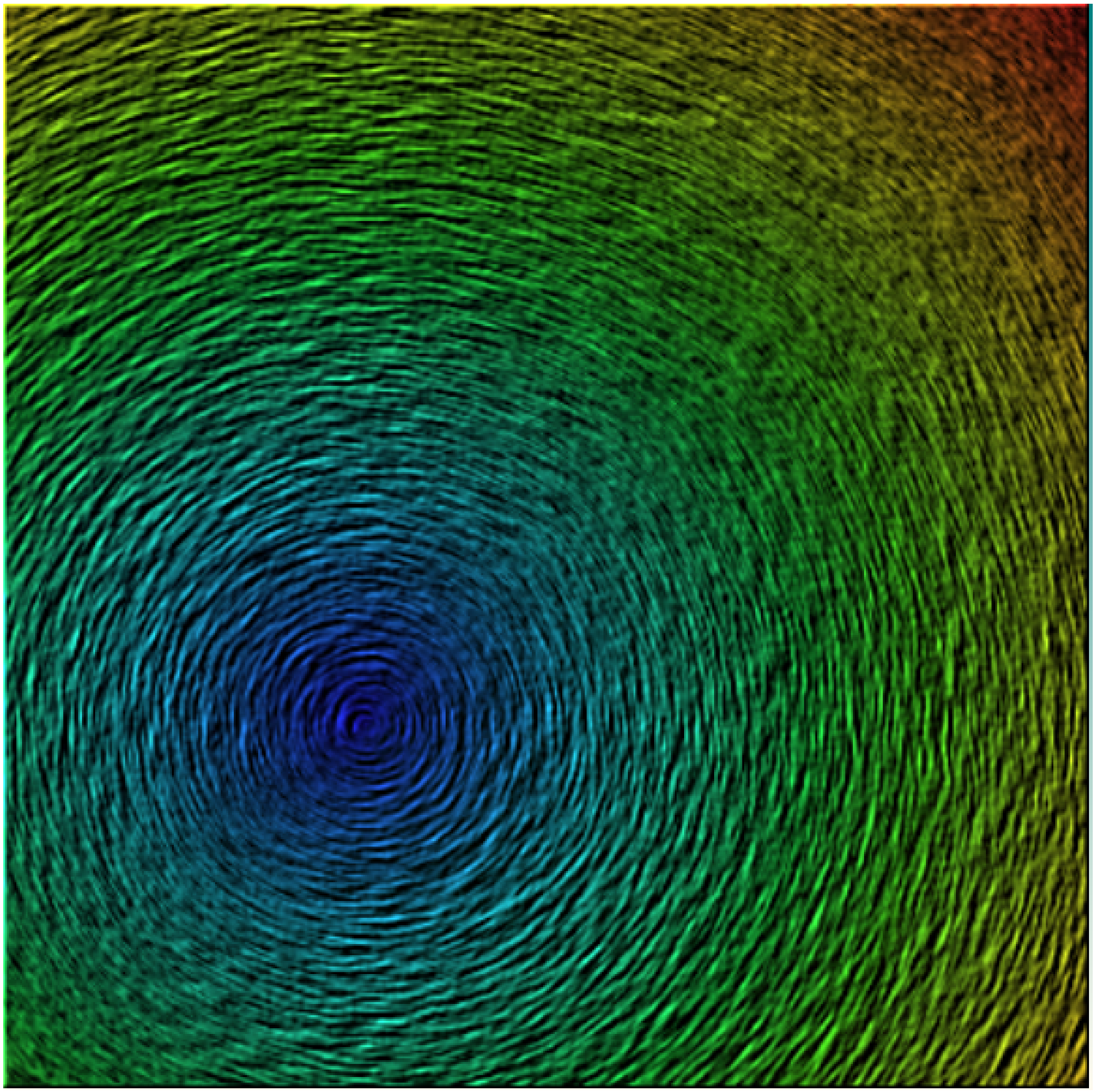
- **Strong Sharpen**—provides a stronger increase in the focus or contrast of the texture image.

**Figure 27.21: Example with the Image Filter Set to Strong Sharpen**



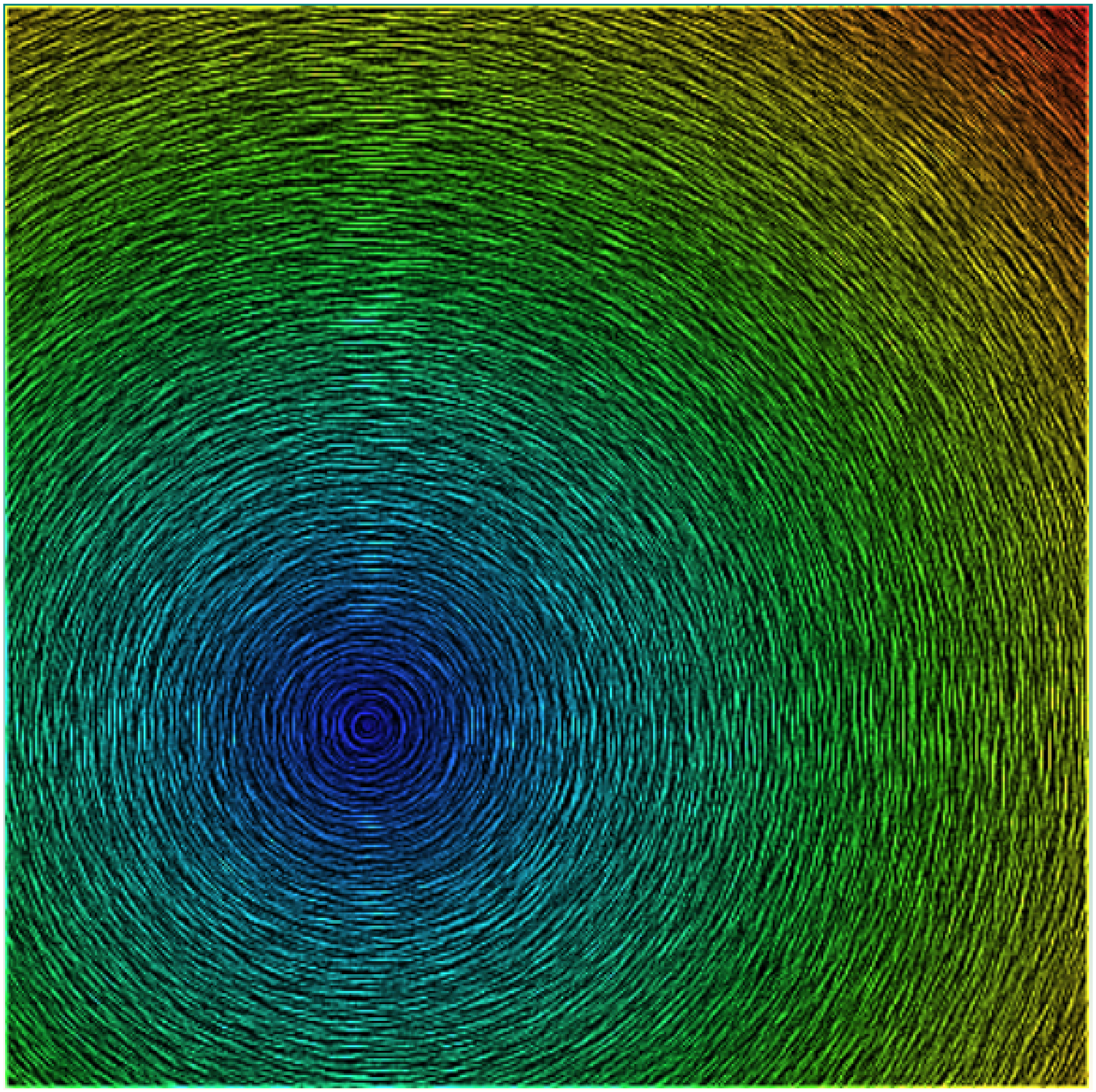
- **Mild Emboss**—creates shadowing and highlights to give edges a raised effect on features in the image.

**Figure 27.22: Example with the Image Filter Set to Mild Emboss**



- **Strong Emboss**—a stronger version of the **Mild Emboss** option. This will also result in the loss of color in areas with little definition before the filter was applied.

**Figure 27.23: Example with the Image Filter Set to Strong Emboss**



15. Set the **Range** for the vector plot:
    - **Auto-Compute Range**—automatically calculates the range for the vector colormap. When disabled, it allows you to clip the range to a specified **Minimum** and **Maximum**.
    - **Use Global Range**—confirms that you are using the automatically computed range. However, if you do not want to use this range, you must disable **Auto-Compute Range**.
    - **Minimum Value** and **Maximum Value**—allows you to set the range for the vector plot values. *These fields only applies when you disable **Auto-Compute Range**.*
16. Set the **ColorMap** fields:
    - **Size**—determines the size of the colormap.
    - **Color Map**—specifies the colors to be used in the colormap display.

- **Use Log Scale**—uses a logarithmic scale.
- **Position**—determines where the colormap is located in the graphics window.
- **Type**—lets you set whether the colormap follows a general, float, or exponential scale.
- **Precision**—lets you set the number of significant digits for the colormap.
- **Show All**—displays all of the colormap labels.
- **Skip**—allows you to set a spacing on the colormap scale.

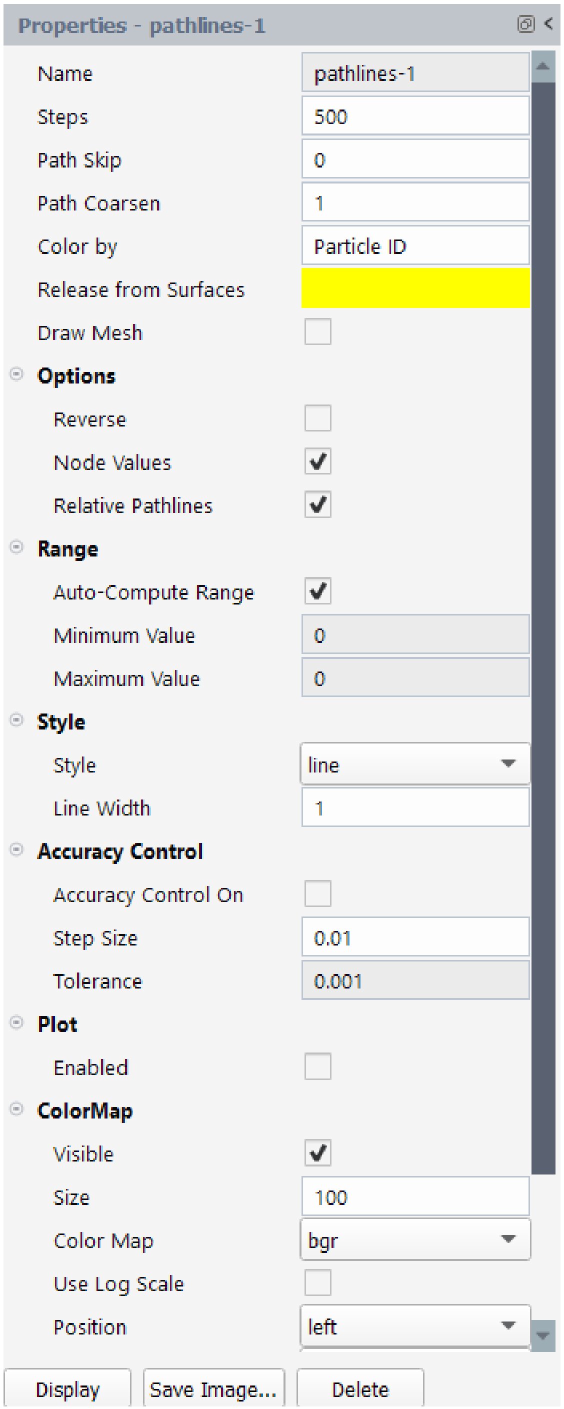
17. Click **Display**.

### 27.1.17.3.5. Pathline Plots

Pathlines are used to visualize the flow of massless particles in the problem domain. The particles are released from one or more surfaces that you have created as described in Surfaces (p. 182). A **line** or **rake** surface (see Line Surfaces (p. 183) and Rake Surfaces (p. 184)) is most commonly used.

1. Right-click **Pathlines** in the Outline View tree and select **New...**.

   **Results** → **Graphics** → **Pathlines** → **New...**




2. (Optional) Enter a **Name** for the pathline object.

3. Set the maximum number of steps that a particle can advance in the **Steps** field.

4. Set how many pathlines you want to skip in the **Path Skip** field to "thin out" a plot that is difficult to understand due to too many pathlines.

5. Set the coarsening factor in the **Path Coarsen** to further simplify the pathline plot.

Providing a coarsening factor of $n$, will result in each $n$ th point being plotted for a given pathline in any cell.

6. Specify the field variable you want to use to color the pathlines by clicking in the entry field for **Color by**, selecting the variable from the **Field** dialog box and clicking **OK**.
7. Select the release point for the particles in the pathlines from the **Release from Surfaces** list.
8. (Optional) Enable **Draw Mesh** to display whichever mesh you select in the **Overlayed Mesh** drop-down list, which includes the outline and any defined mesh objects.
9. Select the desired pathline options:
   - **Reverse**—reverses the flow of the pathlines. The only difference is that the surfaces you select from the **Release from surfaces** list should actually be the final destination of the particles and not their source.

     *If you are interested in determining the source of a particle for which you know the final destination (for example, a particle that leaves the domain through an exit boundary), you can reverse the pathlines and follow them from their destination back to their source.*
   - **Node Values**—specifies that node values should be interpolated to compute the scalar field at a particle location.
   - **Relative Pathlines**—allows you to display the pathlines relative to the rotating reference frame.
10. Set the **Range** for the pathline plot:
    - **Auto-Compute Range**—automatically calculates the range for the pathline colormap. When disabled, it allows you to clip the range to a specified **Minimum** and **Maximum**.
    - **Minimum Value** and **Maximum Value**—allows you to set the range for the pathline plot values. *These fields only applies when you disable **Auto-Compute Range**.*
11. Select how you want the pathlines displayed from the **Style** drop-down list.
12. Set the thickness of the pathlines in the **Line Width** field.
13. Set the **Accuracy Control**s:
    - **Accuracy Control On**—enables/disables accuracy control.
    - **Step Size**—sets the length interval used for computing the next position of the particle
    - **Tolerance**—is an accuracy used in the calculation of pathlines for each time step, when **Accuracy Control On** is enabled.
14. (Optional) Enable XY plotting along pathline trajectories.
15. Set the **ColorMap** fields:
    - **Size**—determines the size of the colormap.
    - **Color Map**—specifies the colors to be used in the colormap display.

- **Use Log Scale**—uses a logarithmic scale.
- **Position**—determines where the colormap is located in the graphics window.
- **Type**—lets you set whether the colormap follows a general, float, or exponential scale.
- **Precision**—lets you set the number of significant digits for the colormap.
- **Skip**—allows you to set a spacing on the colormap scale.

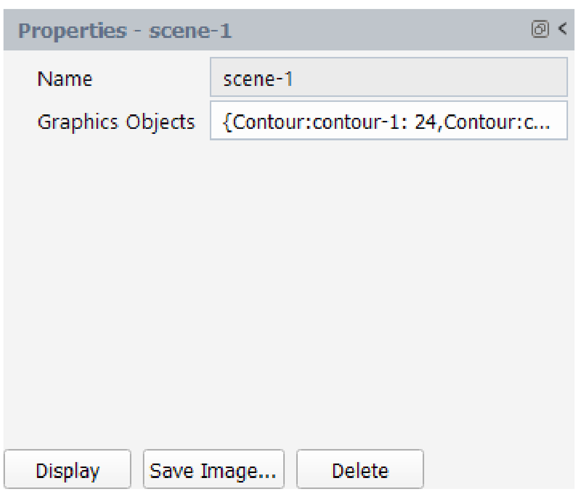
16. Click **Display**.

#### 27.1.17.3.6. Scenes

Scenes can be used to display multiple graphics plots within a single window. For example, you could overlay contours of pressure across a valve with velocity vectors and the mesh at the same location. Scenes allow you to modify the transparency of each plot so that you can emphasize a particular plot or view.

1. Right-click **Scenes** in the Outline View tree and select **New...**.

   **Results** → **Graphics** → **Scenes** → **New...**




2. (Optional) Enter a **Name** for the scene.
3. Select the graphics objects to include in the scene from the **Graphics Objects** drop-down list.
4. Set the transparencies for the selected graphics objects by clicking the **Transparencies** field.

   *The transparency setting is only available for mesh, contour, and LIC plots.*

5. Click **Display**.

### 27.1.17.4. Plots

You can create plots to visualize your results.

The following plots are available:



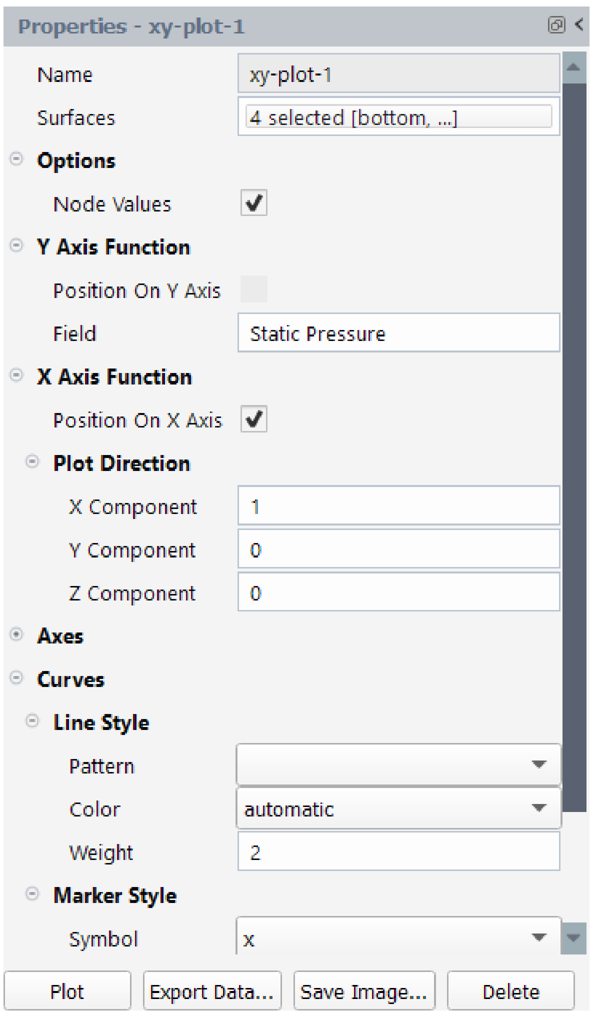
#### 27.1.17.4.1. XY Plots

You can create 2D XY plots of your results for analyzing one variable with respect to another variable.

To create an XY plot:

1. Right-click **XY Plots** in the Outline View tree and select **New...**.

   **Results** → **Graphics** → **XY Plots** → **New...**


2. (Optional) Enter a **Name** for the XY Plot.

3. Select the surfaces where the plot parameters apply by clicking in the yellow **Surfaces** field, selecting the desired surfaces in the **Surfaces** dialog box and clicking **OK**.

4. Specify whether the values should be calculated using cell-centered values alone or if they should also include **Node Values**.

5. Specify the field variable you want to display for the Y axis clicking in the entry field for **Field**, selecting the variable from the **Field** dialog box and clicking **OK**.

   Alternatively, you can set the Y axis to be the location on the Y axis, by enabling **Position On YAxis**.

6. Confirm the field variable you want to display for the X axis. By default, **Position On XAxis** is enabled. You can disable this option to select an alternate variable for the X Axis.

7. Specify the **Plot Direction** by confirming the values for the **X Component**, **Y Component**, and **Z Component**.

8. (Optional) Define **Axes** labels and options:

   **X Axis**

   a. Enter a **Label** for the x-axis.

   b. Choose the axis options:

      - **Log**—axis values are displayed on a logarithmic scale.
      - **Auto Range**—axis minimum, maximum, and increment are computed automatically.
      - **Major Rules**—draw lines through the plot at each major increment.
      - **Minor Rules**—draw lines through the plot at each minor increment.

   c. Set the **Number Format** for how axis values are displayed.

      i. Select the desired number output **Type** from the drop-down list.

      ii. Set the **Precision** to control the number of significant digits.

   d. Set the axis **Range** by entering values for **Minimum** and **Maximum**. *Note that these values are only used when the **Auto Range** option is disabled.*

   **Y Axis**

   a. Setup the Y axis by defining the settings as desired. The inputs fields are equivalent to the ones available for the X axis, as described above.

9. (Optional) Specify the **Curves** properties.

   - For lines (under **Line Style**) you can control the **Pattern**, **Color**, and **Weight** (equivalent to line thickness).
   - For markers (under **Marker Style**), you can control the **Symbol**, **Color**, and **Size**.

10. Click **Display** to plot the data.

You can also save the plot data to a file or as an image, by clicking **Export Data...** or **Save Image...** respectively.

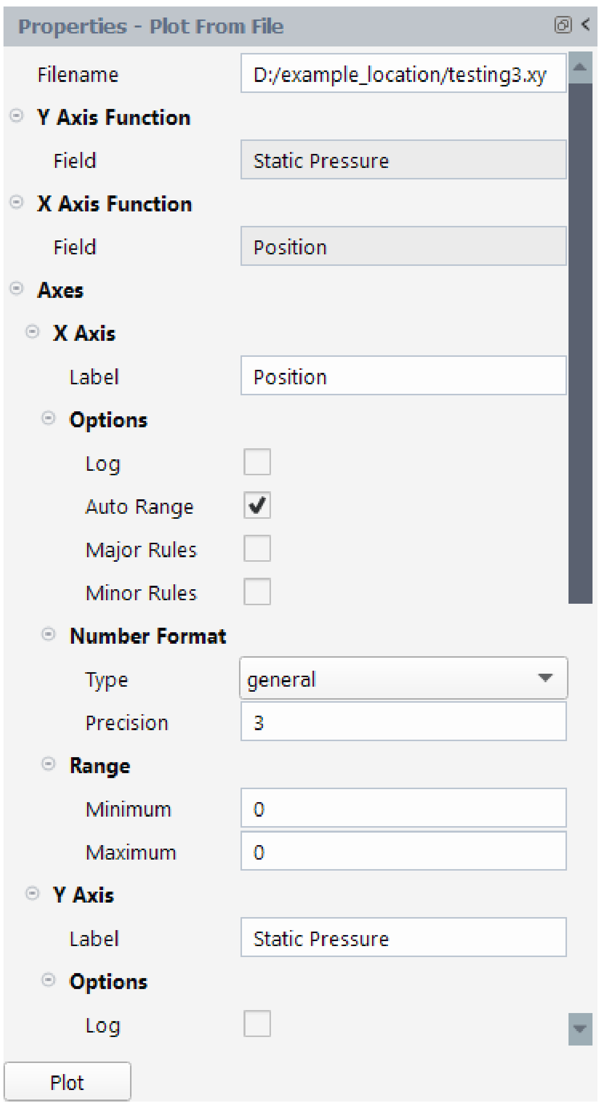
### 27.1.17.4.2. Plot from a File

You can read in `.xy` files to plot the data in the graphics window.

To plot data from a file:

1. Select **Plot From File** in the Outline View tree.

   **Results** → **Graphics** → **Plot From File**




2. Click in **Filename** and then click **Browse...** to open the **Select File** dialog box and select the file you want to plot.

   *The **Y Axis Function** and **X Axis Function** fields are populated automatically when Fluent reads the file.*

3. (Optional) Update the **Axes** and **Curves** settings as needed (the available settings are the same as those described in XY Plots (p. 212)).

4. Click **Plot**.

### 27.1.17.5. Reports

Static reports are available for computation of various postprocessing quantities.

The following report types are available for evaluation:

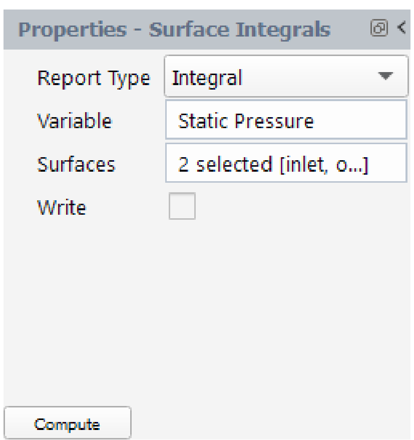


#### 27.1.17.5.1. Surface Integral

To compute a surface integral report:

1. Select **Surface Integrals** in the **Outline View** tree.

   **Results** → **Reports** → **Surface Integrals**




2. Select the **Report Type** you want to compute from the drop-down list.

   - **Area**—area of the selected surfaces.
   - **Integral**—surface integral of the specified variable over the selected surface(s).
   - **Sum**—sum of the specified variable over the selected surface(s).
   - **Standard Deviation**—standard deviation of the specified variable over the selected surface(s).
   - **Area-Weighted Average**—area-weighted average of the specified variable over the selected surface(s).

- **Facet Average**—facet average of the specified variable over the selected surface(s).
- **Facet Minimum**—facet minimum of the specified variable over the selected surface(s).
- **Facet Maximum**—facet maximum of the specified variable over the selected surface(s).
- **Vertex Average**—vertex average of the specified variable over the selected surface(s).
- **Vertex Minimum**—vertex minimum of the specified variable over the selected surface(s).
- **Vertex Maximum**—vertex maximum of the specified variable over the selected surface(s).

3. Click in the **Variable** field to open the **Field** dialog box.
4. Select the desired variable and click **OK**.
5. Click in the **Surfaces** list to open the **Surfaces** dialog box.
6. Select the desired surfaces and click **OK**.
7. (Optional) Enable **Write** to expose options for writing the report to a file.

   - **Append**—append the results of the current report computation to the same file (rather than overwriting an existing report of the same name).
   - **File Name**—specify the name of the saved report and the location where it is saved.

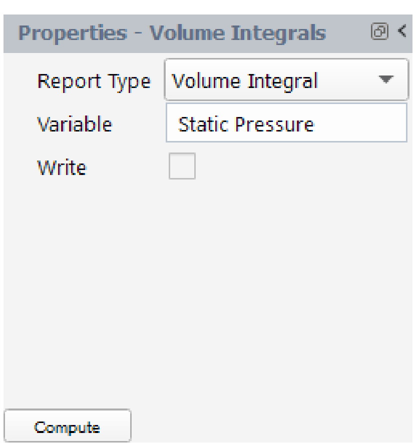
   Click **Compute** to write the report.
8. Click **Compute** and the result is printed in the **Console**.

#### 27.1.17.5.2. Volume Integral

To compute a volume integral report:

1. Select **Volume Integrals** in the **Outline View** tree.

   **Results** → **Reports** → **Volume Integrals**




2. Select the **Report Type** you want to compute from the drop-down list.

- **Volume**—volume of the domain.
- **Volume Integral**—volume integral of the specified variable over domain.
- **Sum**—sum of the specified variable over the domain.
- **Volume Average**—volume-averaged value of the specified variable across the domain.
- **Minimum**—minimum value of the specified variable over the domain.
- **Maximum**—maximum value of the specified variable over the domain.
- **Mass Integral**—mass-weighted integral of the specified variable across the domain.
- **Mass**—mass of the entire domain.

3. Click in the **Variable** field to open the **Field** dialog box.
4. Select the desired variable and click **OK**.
5. Click in the **Surfaces** list to open the **Surfaces** dialog box.
6. Select the desired surfaces and click **OK**.
7. (Optional) Enable **Write** to expose options for writing the report to a file.
   - **Append**—append the results of the current report computation to the same file (rather than overwriting an existing report of the same name).
   - **File Name**—specify the name of the saved report and the location where it is saved.

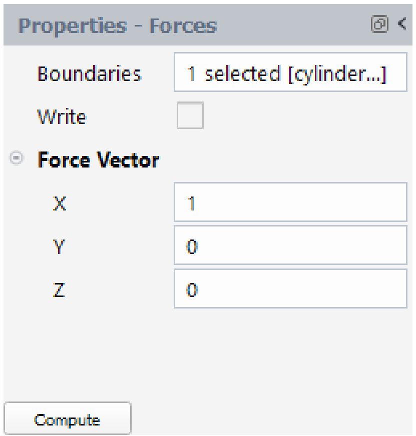
   Click **Compute** to write the report.
8. Click **Compute** and the result is printed in the **Console**.

#### 27.1.17.5.3. Force

To compute a force report:

1. Select **Forces** in the **Outline View** tree.

   **Results** → **Reports** → **Forces**




2. Click in the **Boundaries** list to open the **Boundaries** dialog box, select the desired boundaries, and click **OK** to confirm your selection(s).

3. (Optional) Enable **Write** to expose options for writing the report to a file.

   - **Append**—append the results of the current report computation to the same file (rather than overwriting an existing report of the same name).

   - **File Name**—specify the name of the saved report and the location where it is saved.

   Click **Compute** to write the report.

4. Specify the **X**, **Y**, and **Z** components of the **Force Vector**.

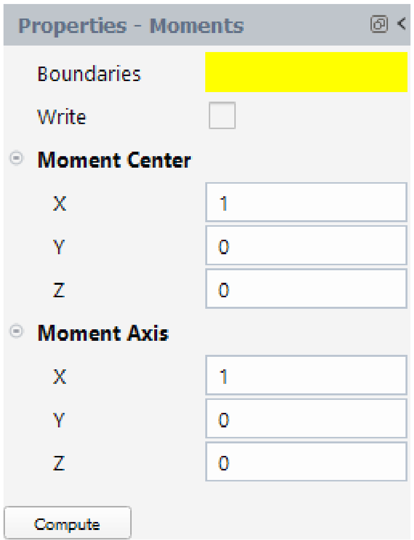
5. Click **Compute** and the result is printed in the **Console**.

#### 27.1.17.5.4. Moment

To compute a moment report:

1. Select **Moments** in the **Outline View** tree.

   **Results** → **Reports** → **Moments**




2. Click in the **Boundaries** list to open the **Boundaries** dialog box, select the desired boundaries, and click **OK** to confirm your selection(s).

3. (Optional) Enable **Write** to expose options for writing the report to a file.

   - **Append**—append the results of the current report computation to the same file (rather than overwriting an existing report of the same name).

   - **File Name**—specify the name of the saved report and the location where it is saved.

   Click **Compute** to write the report.

4. Specify the **X**, **Y**, and **Z** components of the **Moment Center**.

5. Specify the **X**, **Y**, and **Z** components of the **Moment Axis**.

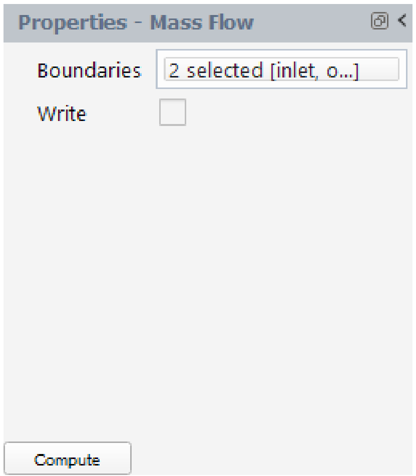
6. Click **Compute** and the result is printed in the **Console**.

### 27.1.17.5.5. Mass Flow

To compute a mass flow report:

1. Select **Mass Flow** in the **Outline View** tree.

   **Results** → **Reports** → **Mass Flow**




2. Click in the **Boundaries** list to open the **Boundaries** dialog box, select the desired boundaries, and click **OK** to confirm your selection(s).

3. (Optional) Enable **Write** to expose options for writing the report to a file.

   - **Append**—append the results of the current report computation to the same file (rather than overwriting an existing report of the same name).

   - **File Name**—specify the name of the saved report and the location where it is saved.

   Click **Compute** to write the report.

4. Click **Compute** and the result is printed in the **Console**.

### 27.1.18. References

## Bibliography

[1] O. Malaspinas and P. Sagaut. "Consistent subgrid scale modelling for lattice Boltzmann methods". *J. Fluid Mech*. 700. 514–542. 2012.

[2] U. Piomelli, P. Moin and J. H. Ferziger. "Model Consistency in Large-Eddy Simulation of Turbulent Channel Flow". *Physics of Fluids*. 31. 1884–1894. 1988.

# 27.2. Fluent LB Best Practices

This document provides general guidance on how to get the most out of the Fluent Lattice Boltzmann application and solver. The following areas are covered:

## 27.2.1. Input Mesh

Fluent LBM provides several options to read/import surface and volume meshes as well as Fluent case files. The boundary facets of these meshes provide the starting geometry for the Fluent LBM octree mesher. These surfaces will therefore be re-discretized to produce a consumable octree mesh. Geometric details of interest therefore need to be adequately resolved in the starting mesh for them to be correctly represented in Fluent LBM, including making sure that there is adequate curvature resolution on faceted surfaces.

To allow the visualization of solution quantities on regions of interest, Fluent LBM interpolates the solution back to the original boundary mesh. For most cases, the resolution of the original boundary mesh is sufficient for post-processing purposes; however, for cases where the resolution is much coarser than the LBM octree mesh (for example, large flat coarsely resolved surfaces or extruded 2.5D meshes) the interpolated solution on the original surface mesh may give unexpected results.

## 27.2.2. Boundary Conditions

This section describes best practices for defining boundary conditions in Fluent Lattice Boltzmann.



### 27.2.2.1. Model Orientation

In many situations, the orientation of the original geometry is aligned with global coordinate axes. This is advantageous where planar boundaries are present for two reasons:

1. The octree mesher in Fluent LBM is global coordinate system-aligned, and planar boundaries which are similarly aligned will be aligned with octree mesh faces, avoiding stair-stepping.

2. The numerical implementation of some boundary conditions, for example symmetry boundaries and outlets, means that they are more robust when aligned with the octree mesh.

### 27.2.2.2. Pressure Outlets

Axis-aligned pressure outlets are implemented using a non-reflecting condition to minimize the possibility of pressure waves re-entering the domain. For pseudo 2D (2.5D) cases where the outlet is adjacent to symmetry boundaries, this boundary option requires a sufficient local resolution (>10 octs) in the spanwise direction to prevent pressure and velocity artifacts developing locally.

#### 27.2.2.3. Mass Balance

The local calculation of mass flow at inlets and outlets can be affected by the local mesh resolution, particularly for boundaries that include curved surfaces/edges or those that are not axis-aligned, due to the cartesian nature of the octree mesh and it's intersection with the underlying geometry. To ensure a good mass balance, it is recommended that the inlet and outlet boundary resolutions are sufficient to adequately resolve the geometry. Additional local refinement at these boundaries will generally improve overall mass balance as well.

### 27.2.3. Turbulent Flows

For turbulent flows, the only model currently available in Fluent LBM is the **Smagorinsky** Large Eddy Simulation (LES) model. Like all LES models, the **Smagorinsky** model acts as a low-pass filter which resolves turbulent scales that are larger than the local mesh size, and modeling scales that are smaller than the local mesh size.

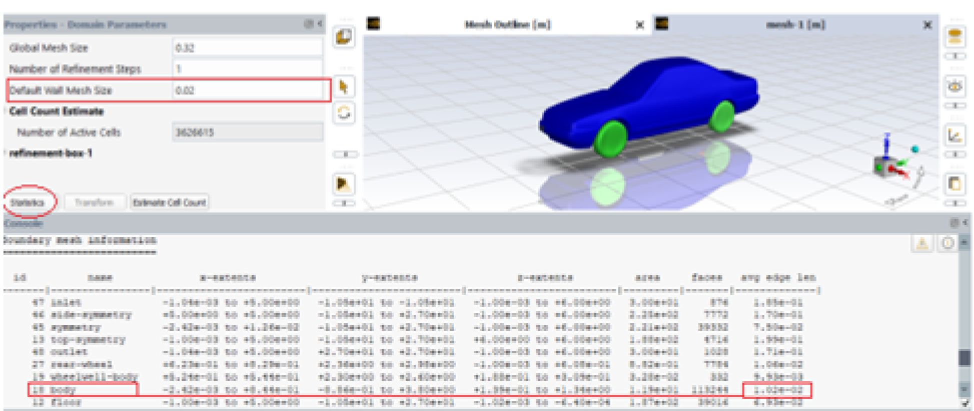
For velocity inlet conditions there is currently no option for specifying incoming fluctuations. In practice if incoming turbulence is required, initiation can be promoted with the use of trips, steps, or other turbulence generators upstream of the body of interest. Sufficient mesh resolution is therefore required to ensure not only that turbulence is initiated at the desired location but that it can also be sustained and allowed to propagate downstream without dissipating.

An appropriate mesh resolution is important for getting the right flow structures using Fluent LBM. You can look at the Yplus values on the wall surfaces to judge the mesh size. A small Yplus value, for example near 10, indicates that the mesh is highly resolved and could potentially be coarsened. A large Yplus value, for example more than 500, indicates that the mesh is not sufficiently resolved and the simulation would benefit from a refined mesh near the wall.

### 27.2.4. Mesh Settings

You need to specify mesh sizes before initializing the solution or before generating an octree mesh. A visual inspection of the boundary mesh, along with the knowledge of average face edge length, can be useful in determining the target mesh sizes. To get the average face edge length of the surface mesh, click **Statistics** in the properties of **Domain Parameters** panel.

**Figure 27.24: Fluent LBM Boundary Mesh Statistics**



In simulating flows past objects, you will need the wake region to be well resolved. Use a refinement box around the object extending a few object lengths downstream and specify a uniform mesh resolution in such a refinement box for best results.

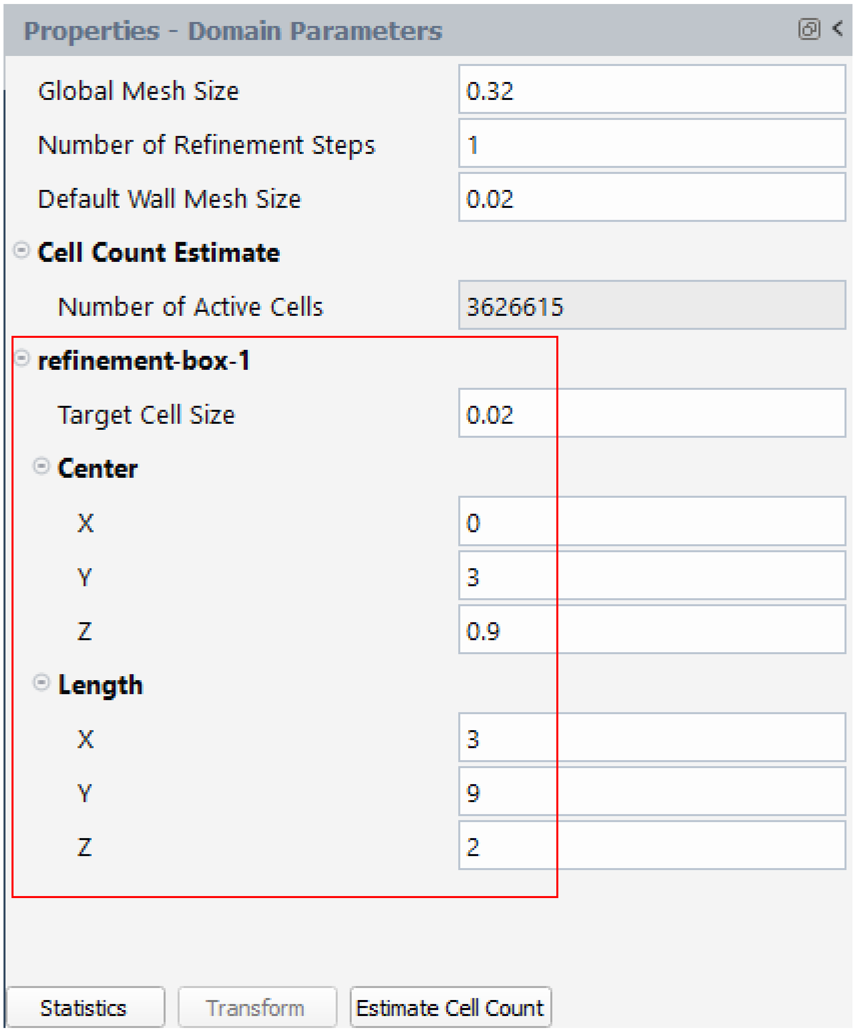
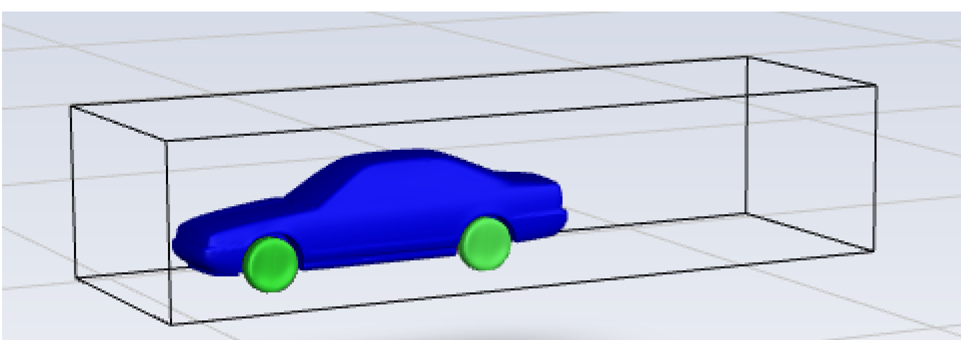



**Figure 27.25: Refinement Box Display**



## 27.2.5. Mesh Size Selection

To accurately calculate the flow and forces around objects with the Fluent LB Solver, it is recommended that the mesh is fine enough to capture all the pertinent characteristcs of the geometry and the flow.

For turbulent flows, this typically requires meshes that are suitable for LES calculations in regions of interest to capture resolved turbulence scales. In addition, it is recommended that refinement boxes are used around bluff bodies to avoid mesh transitions near wall boundaries of interest. Application of a constant sized mesh around bodies promotes better flow and force prediction than those where several mesh transitions occur near the surface.

When meshing thin gaps, it is important to ensure that sufficient mesh refinement is applied to surfaces to resolve the gap properly with the octree mesher. Insufficient resolution can lead to an incorrect mesh connection across the gap resulting in undesirable flows. Baffles (internal surfaces of zero thickness) do not require special attention.

After specifying the desired mesh sizes you can click **Estimate** to get an estimate of the octant count before generating the mesh via initialization. If the estimate predicts a prohibitively large octant count, you should change the mesh sizes until the octant count prediction is in an acceptable range.

Typical memory requirements for the core LB solver and the entire LB application are listed in the following table:

**Table 27.3: Fluent LBM Memory Requirements**

| | Single Precision | Double Precision |
|---|---|---|
| Solver | 400 MB/million octs | 600 MB/million octs |
| Total | 1.75 GB/million octs | 2.25 GB/million octs |

When running on a GPU, only the solver memory counts toward the GPU memory usage. A GPU such as the Quadro RTX 6000 with 24GB memory is therefore capable of performing a simulation with approximately 40 million octs in double precision.

## 27.2.6. Time Step Size Selection

Based on the mesh size and physics setup, the solver can compute an optimal time step size for you, based on a reference velocity (inlet velocity) and reference length scale (largest oct size). You can click **Compute Timestep** to get a **Time Step Size** before initializing your solution.

The Fluent LB Solver adopts a hierarchical time-stepping approach to solve the transient Lattice Boltzmann equations. Starting with the specified time step size for the largest oct, the Fluent LB solver works by halving the time step size on successively smaller octree levels to maintain an approximately consistent CFL number. Therefore, for every coarse octree time step, multiple time steps are taken at each refinement level to maintain a consistent flow time at all octree levels (the actual number of timesteps is 2n - 1, where n is the octree level with n = 1 being the coarsest level). Because of this, calculations performed on refined meshes do not always scale linearly with the total number of octants.

## 27.2.7. Pressure Damping

You will notice a pressure wave as the flow is developing in the domain which can be effectively damped using **Startup Timesteps (pressure damping)**. Typically 5,000 to 10,000 timesteps are required to damp the unwanted pressure waves.

To ensure that the solution is free from unwanted pressure waves, you can create a **Report Definition** for pressure on a point surface in the domain and monitor the pressure fluctuations on the point. When there are no more large-scale fluctuations it is safe to assume that the initial pressure waves have subsided.

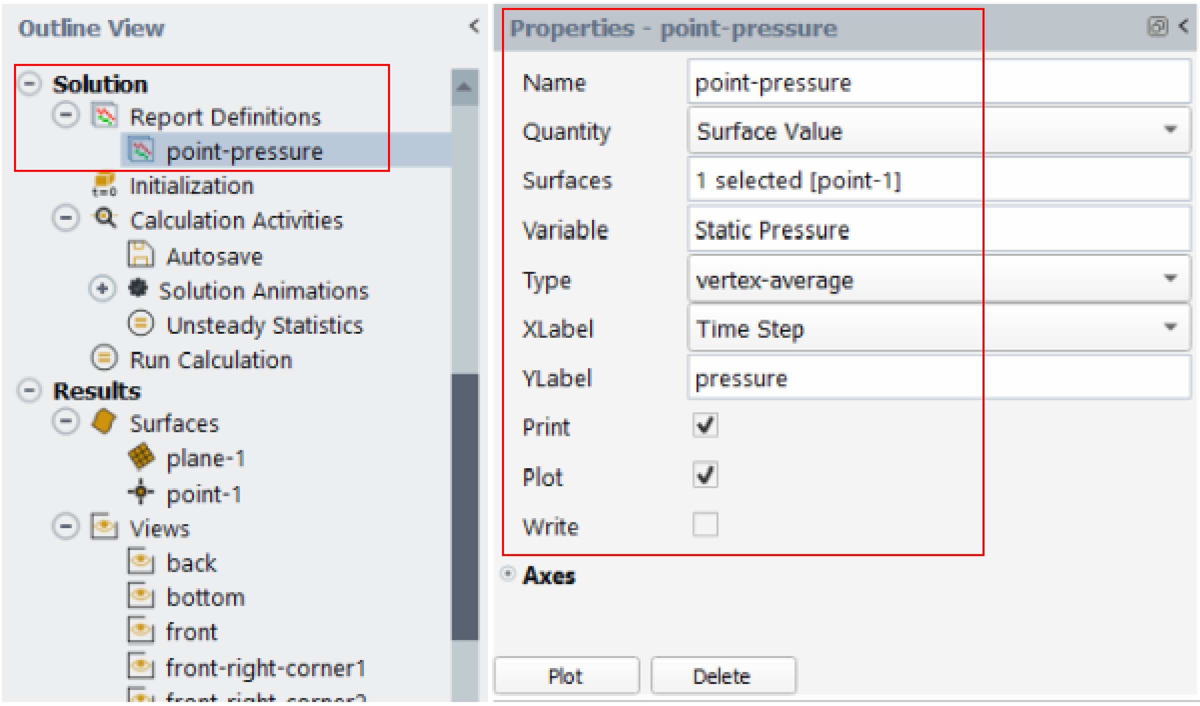
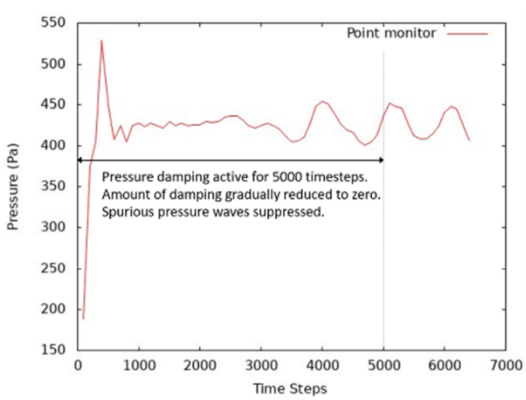



**Figure 27.26: Plot of Pressure at Point Monitor**




### 27.2.8. Unsteady Statistics

For many unsteady flows, quantities of interest are obtained by observing the statistical mean characteristics over a suitable time period after the flow has reached some statistically steady state. Statistical flow variables (mean and RMS for pressure and velocity) can be obtained by enabling **Unsteady Statistics** at a suitable point in the flow calculation.

## 27.3. Fluent LB Tutorial

This tutorial is divided into the following sections:

## 27.3.1. Introduction

The purpose of this tutorial is to use Fluent LB to compute the turbulent flow passed a skyscraper.

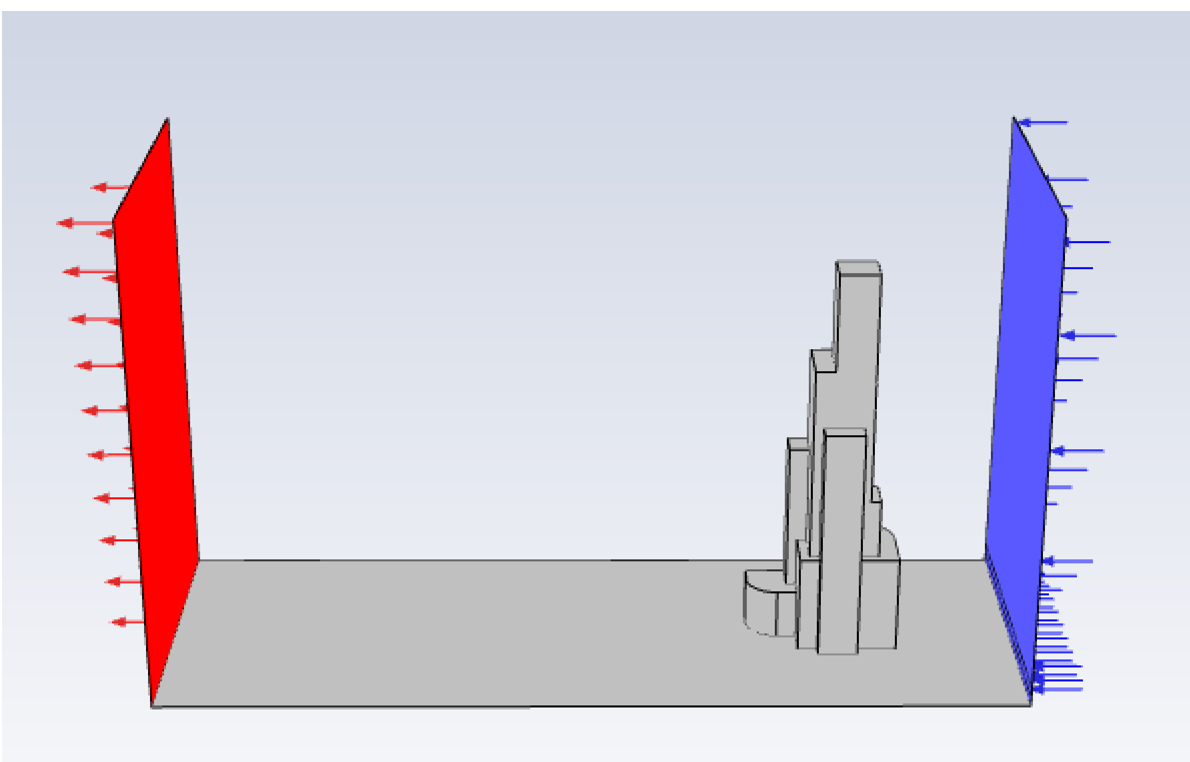
This tutorial demonstrates how to do the following:

- Read the surface mesh and define the parameters that will be used to generate an octree mesh for the domain.
- Setup the models and boundary conditions.
- Generate an octree mesh and initialize the solution flow field.
- Compute the solution.
- Examine the results.

## 27.3.2. Problem Description

The problem considers the turbulent flow of air around a skyscraper at a free stream velocity of 5 m/s. The skyscraper modeled here is shown in Figure 27.27: Problem Schematic (p. 226).

**Figure 27.27: Problem Schematic**



## 27.3.3. Setup and Solution

The following sections describe the setup and solution steps for this tutorial:

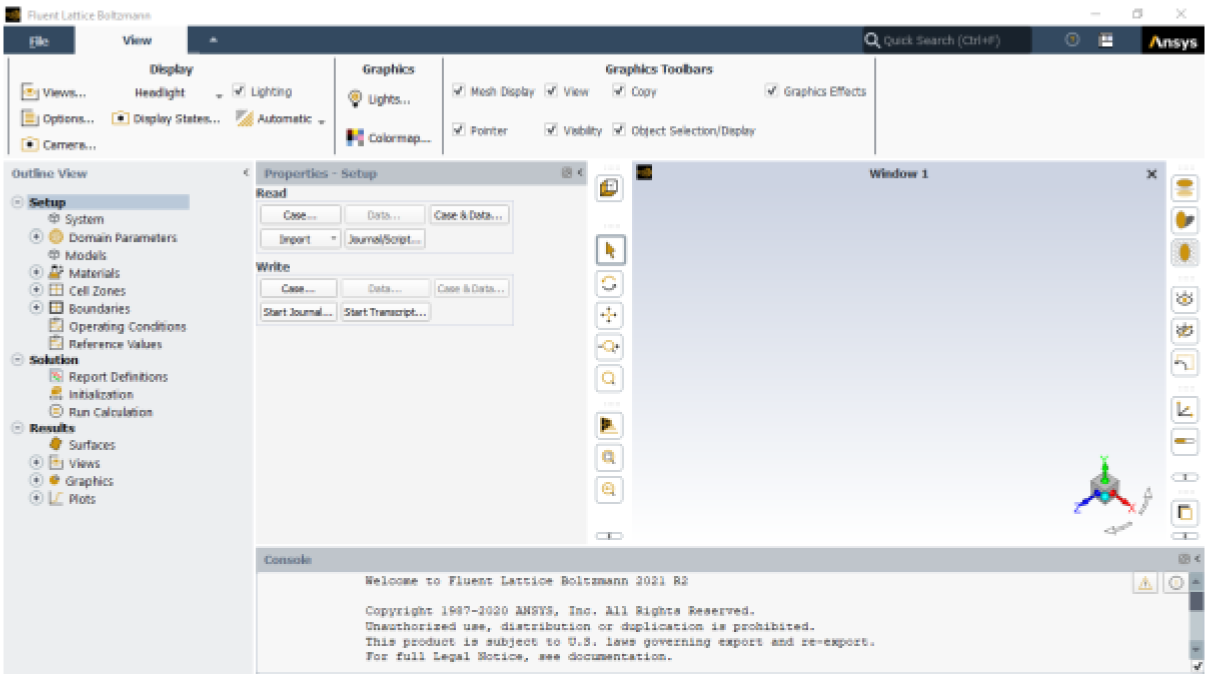


### 27.3.3.1. Preparation

To prepare for running this tutorial:

1. Download the `LB_Skyscraper.zip` file here.
2. Unzip `LB_Skyscraper.zip` to your working directory.
3. *The file, `Skyscraper.msh`, can be found in the folder.*
4. Use Fluent Launcher to start Ansys Fluent.
5. On the Fluent Launcher, set the **Capability Level** to **Enterprise**, then select **LB Method**.
6. Enable **Double Precision** under **Options**.
7. Set **Solver GPGPUs per machine** to 1 under **Parallel (Local Machine)**.
8. Click **Start**.

**Figure 27.28: Fluent Lattice Boltzmann Workspace**




The graphical user interface (GUI) of Fluent Lattice Boltzmann is very similar to Fluent, however there are some differences such as:

- The ribbon is reduced to only the **File** and **View** tabs and therefore most actions for setting up your simulation will be performed in the **Outline View**.
- Fluent Lattice Boltzmann does not use task pages or dialog boxes, rather settings are defined in the **Properties** window which is accessed by left-clicking an item in the **Outline View**. Note that settings in the **Properties** window are saved as you define them, unlike dialog boxes which require you to click an **OK** button.
- The console does not interact with the text-user-interface (TUI) but does allow for Python scripting. It is recommended that you read Python commands from a journal / script file.

## 27.3.3.2. Reading and Defining the Mesh

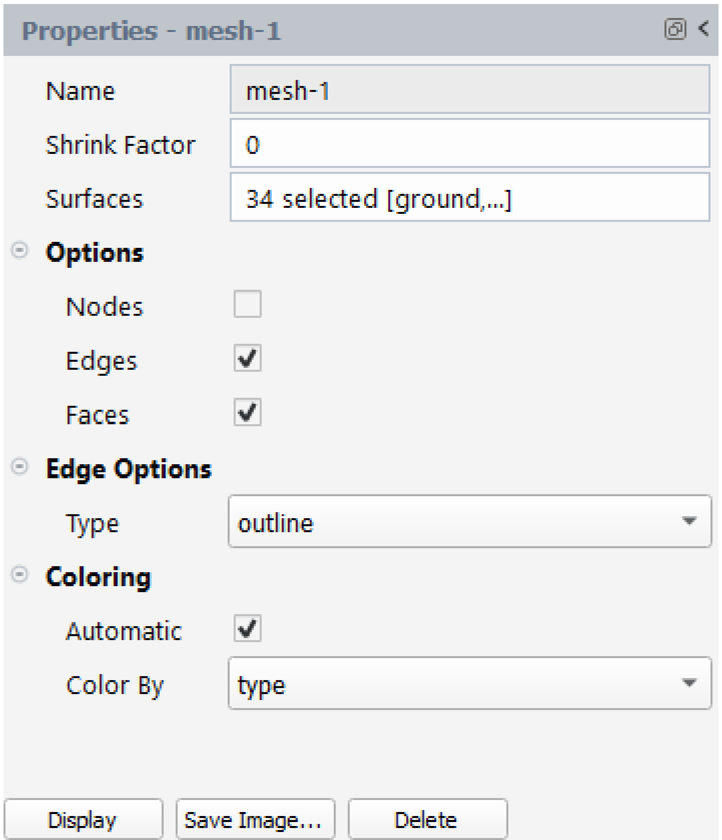
1. Import the mesh file `Skyscraper.msh`.

   **File → Import → Surface Mesh...**

   *As Fluent Lattice Boltzmann reads the surface mesh file, it will report the progress in the console.*

2. Display the mesh.

   **Results → Graphics → Meshes → New...**




a. Click the **Surfaces** field to open the **Surfaces** dialog box.

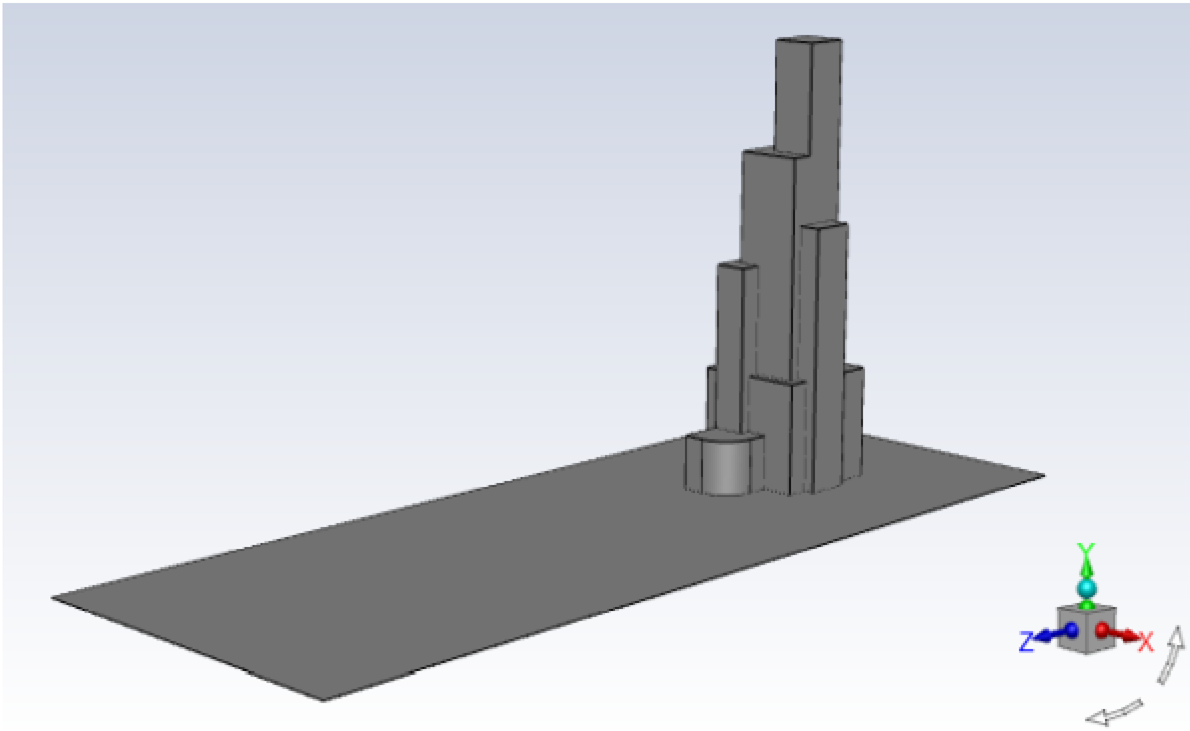
b. Select **Wall** and click **OK** to confirm the selected surfaces and close the **Surfaces** dialog box.

c. Ensure that **Faces** and **Edges** are enabled under **Options**.

d. Click the **Type** drop-down list under **Edge Options** and select **outline**.

e. Click **Display**.

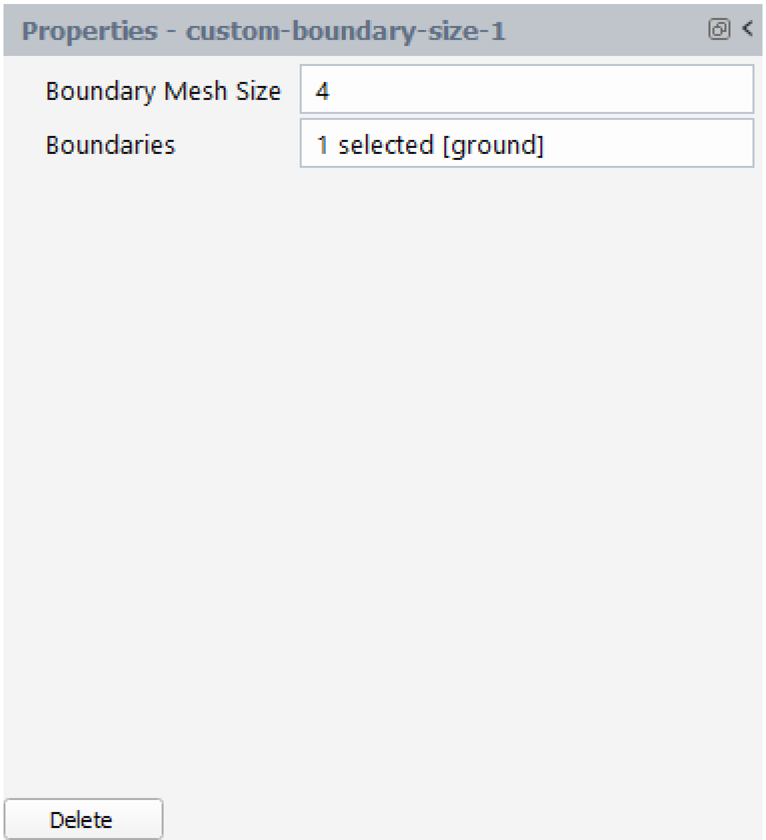
**Figure 27.29: Skyscraper Mesh Display**




3. Create a custom mesh boundary sizing for the ground.

   *We will create a custom mesh boundary sizing for the ground because it does not require the higher resolution needed for the skyscraper walls, which will be specified in* ***Domain Parameters****. Note that the custom mesh boundary sizing specifies the mesh size for a specific boundary and overrides the global mesh size.*

   **Setup** → **Custom Mesh Boundary Sizes** → **New...**




a. Enter 4 for **Boundary Mesh Size**.

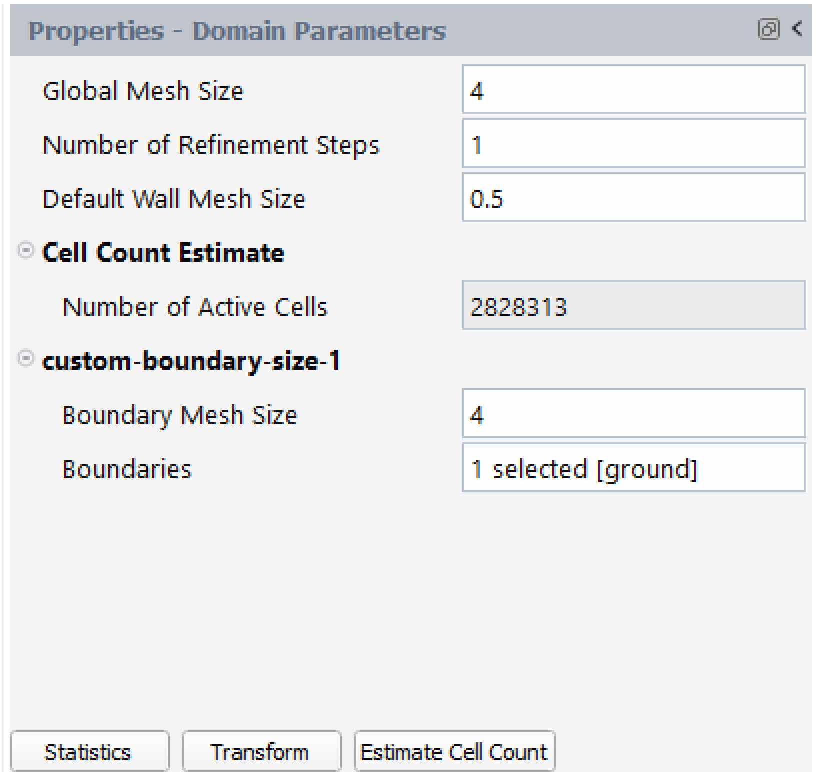
b. Click the **Boundaries** field to open the **Boundaries** dialog box.

c. Select **ground** and click **OK** to confirm the surface for the custom boundary sizing.

4. View the surface mesh statistics and define domain parameters.

*These parameters specify the refinement of the octree mesh, which is generated during initialization. For more information on setting these parameters see Defining the Domain Parameters for Meshing in the Fluent Beta Features Manual (p. 158).*

**Setup** → **Domain Parameters**




a. Click the **Statistics** button to print information about the sizing of the surface mesh to the console.

   *If the dimensions of the surface mesh are incorrect, the **Transform** button can be used to scale the domain.*

b. Enter 4 for **Global Mesh Size**.

c. Retain the value of `1` for **Number of Refinement Steps**.

   *This controls how the mesh transitions between areas of refinement and coarsness. Larger values result in more gradual mesh transitions, which are appropriate for modeling flows with thicker boundary layers.*

d. Enter `0.5` for **Default Wall Mesh Size**.

e. Click **Estimate Cell Count**.

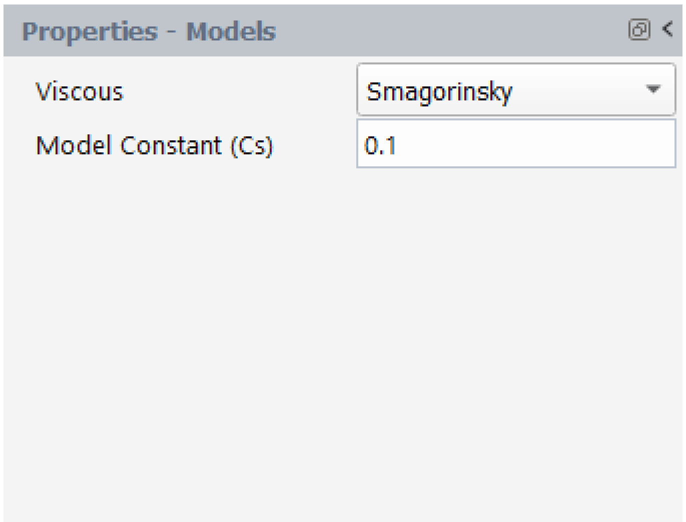
*Clicking **Estimate Cell Count** updates the **Number of Active Cells** field to show the estimated number of cells that will be present in the octree mesh, which is generated using these parameters.*

### 27.3.3.3. Models

1. Retain the default Smagorinsky turbulence model.

   *Fluent Lattice Boltzmann provides two options for the viscous model, **Laminar** and **Smagorinsky** (for modeling turbulence). Because the Lattice Boltzmann method is intrinsically transient and typically uses fine meshes, Reynolds-Averaged Navier-Stokes (RANS) models can not be used and only Large Eddy Simulation (LES) models are available for modeling turbulence.*

**Setup → Models**




a. Retain the default **Smagorinsky** model.

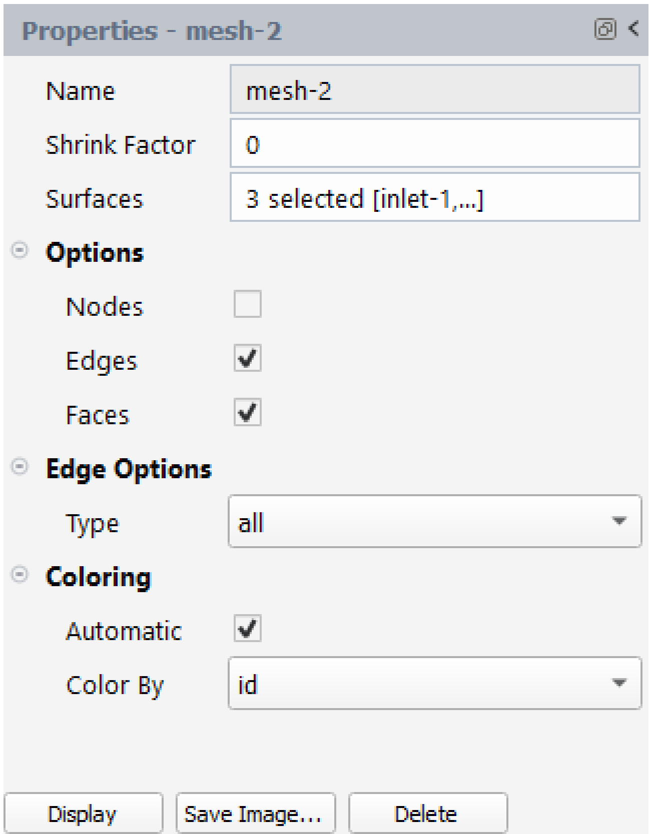
b. Retain the default entry of `0.1` for **Model Constant (Cs)**.

#### 27.3.3.4. Boundary Conditions

1. Display the velocity-inlets.

   *For this problem 3 inlets are used. Because the wind's velocity will be less near the ground, two narrow inlets **inlet-1** and **inlet-2**, will have a lower velocity than the wider **inlet-3** in order to approximate a shear wind profile.*

   **Results → Graphics → Meshes → New...**



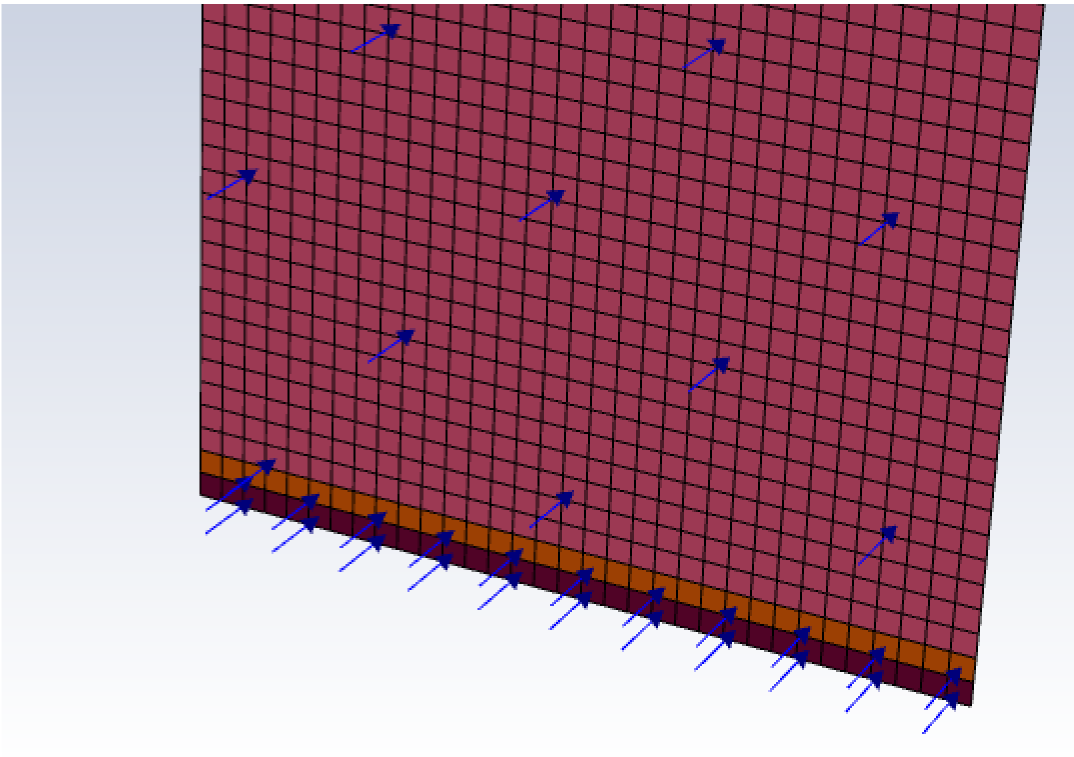

a. Click the **Surfaces** field to open the **Surfaces** dialog box.

b. Click **Inlet** to select all 3 inlets and click **OK** to confirm the selected surfaces and close the **Surfaces** dialog box.

c. Under **Coloring**, click the **Color By** drop-down list and select **id**.

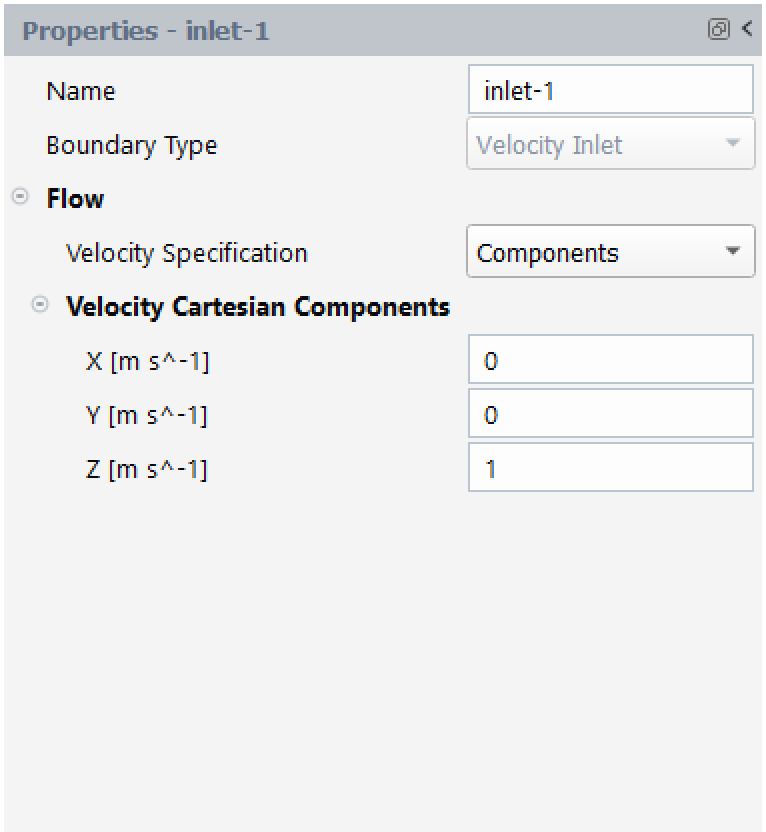
d. Click **Display**.

**Figure 27.30: Velocity-inlets**



2. Define the boundary conditions for inlet-1.

   **Setup** → **Boundaries** → **inlet-1**

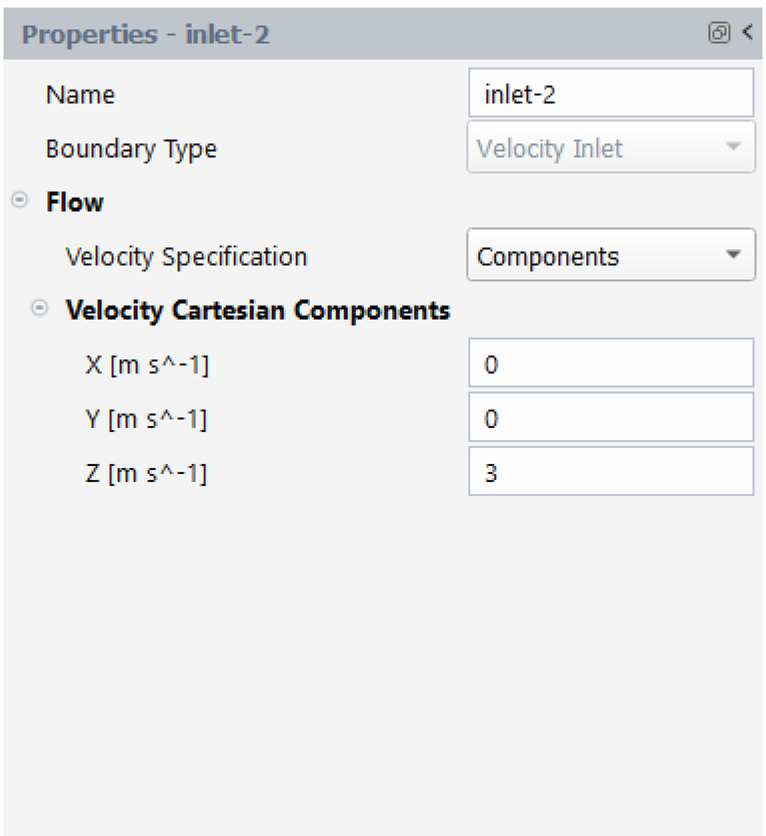



a. Select **Components** from the **Velocity Specification** drop-down list.

b. Set the values of the **Velocity Cartesian Components** to `0`, `0`, and `1` for **X**, **Y**, and **Z** respectively.

3. Define the boundary conditions for inlet-2.

**Setup** → **Boundaries** → **inlet-2**

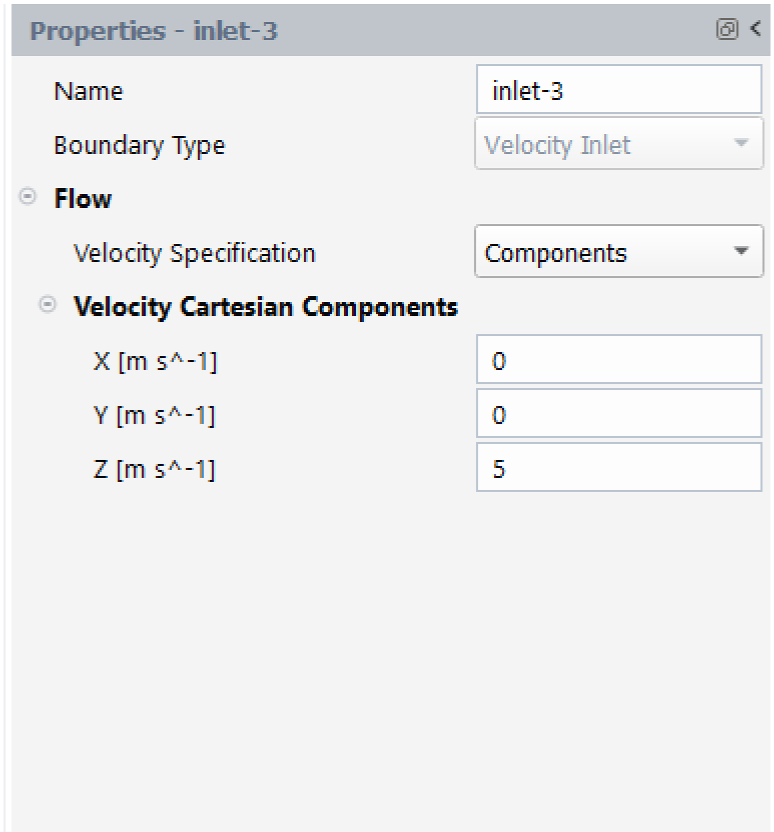



a. Select **Components** from the **Velocity Specification** drop-down list.

b. Set the values of the **Velocity Cartesian Components** to `0`, `0`, and `3` for **X**, **Y**, and **Z** respectively.

4. Define the boundary conditions for inlet-3.

**Setup** → **Boundaries** → **inlet-3**



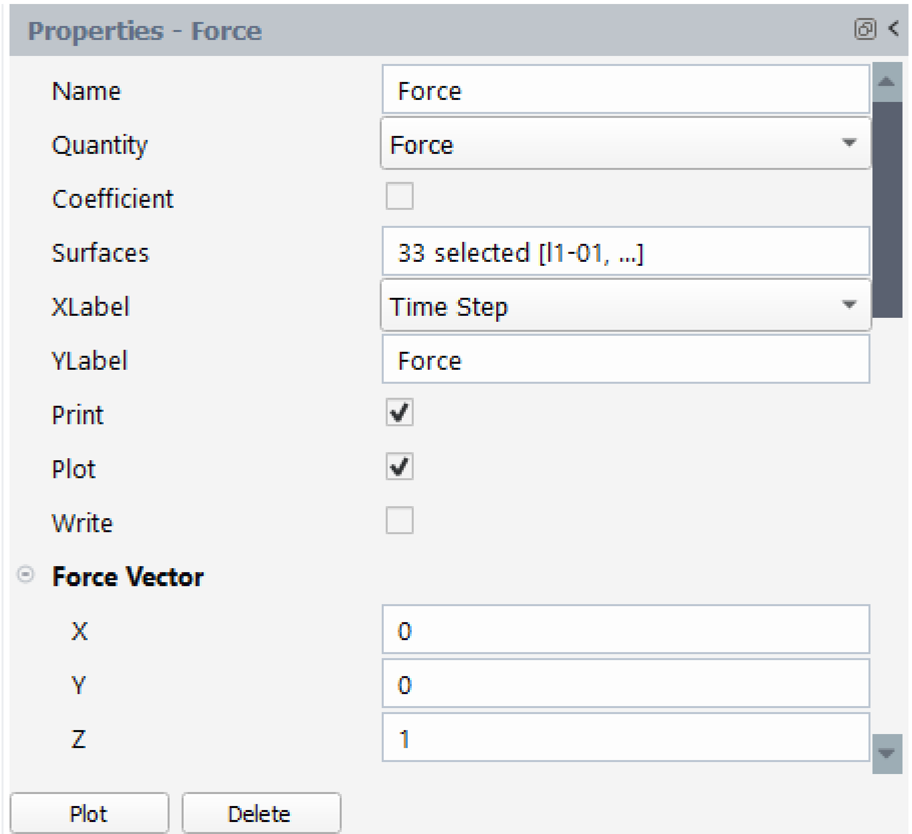

a. Select **Components** from the **Velocity Specification** drop-down list.

b. Set the values of the **Velocity Cartesian Components** to `0`, `0`, and `5` for **X**, **Y**, and **Z** respectively.

### 27.3.3.5. Solution

1. Create a force report definition to monitor the force on the skyscraper.

   **Solution** → **Report Definitions** → **New...**




a. Enter `Force` for **Name**.

b. Enable **Print** and **Plot**.

c. Ensure that **Force** is selected from the **Quantity** drop-down list.

d. Ensure that **Time Step** is selected for **XLabel** and enter `Force` for **YLabel**.

e. Click the **Surfaces** field to open the **Surfaces** dialog box.

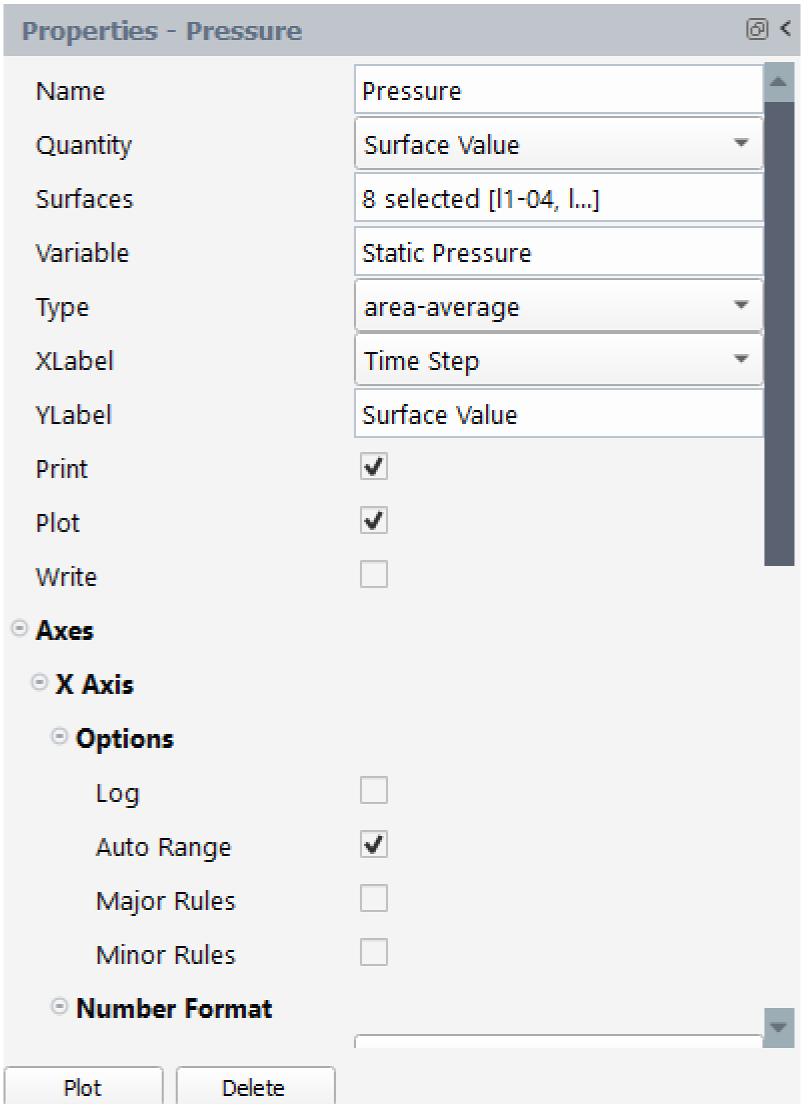
f. Click **Wall** to select all walls in the domain then disable **ground** which selects only the skyscraper walls.

g. Click **OK** to confirm the surfaces and close the **Surfaces** dialog box.

h. Set the values for **Force Vector** to `0`, `0`, and `1` for **X**, **Y**, and **Z** respectively.

2. Create a report definition to monitor the pressure on the back skyscraper walls.

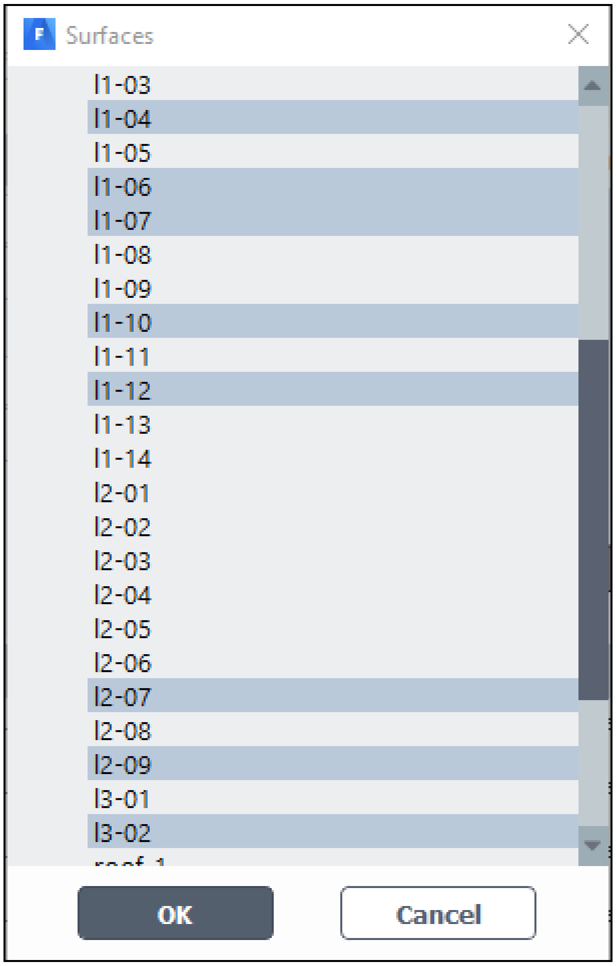
**Solution** → **Report Definitions** → **New...**




a. Enter `Pressure` for **Name**.

b. Select **Surface Value** from the **Quantity** drop-down list.

c. Click the **Surfaces** field to open the **Surfaces** dialog box.




d. Select `l1-04`, `l1-06`, `l1-07`, `l1-10`, `l1-12`, `l2-07`, `l2-09`, and `l3-02` from the list of surfaces then click **OK** to confirm the surfaces and close the **Surfaces** dialog box.

e. Click the **Variable** field to open the **Field** dialog box.

f. Ensure that **Pressure...** and **Static Pressure...** are selected from the **Section** and **Field** lists respectively.

g. Click **OK** to close the **Field** dialog box.

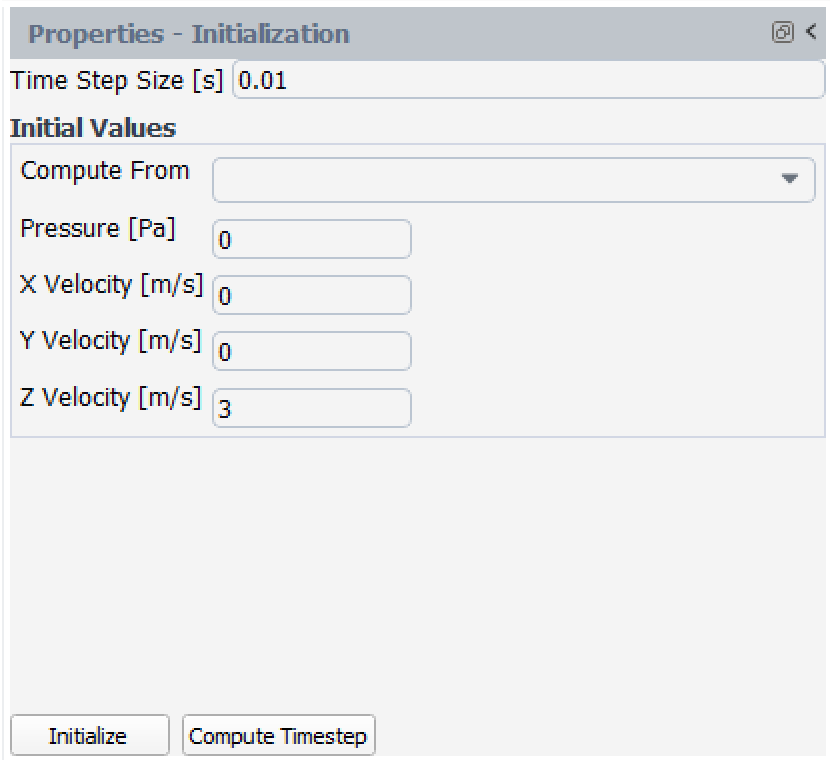
h. Click the **Type** drop-down list and select **area-average**.

i. Enable **Print** and **Plot**.

j. Ensure that **Time Step** is selected for **XLabel** and enter `Static Pressure` for **YLabel**.

3. Initialize the solution.

**Solution** → **Initialization**




a. Click **Compute Timestep** to estimate the time step size

b. Round down the time step size by entering `0.01` for **Time Step Size**.

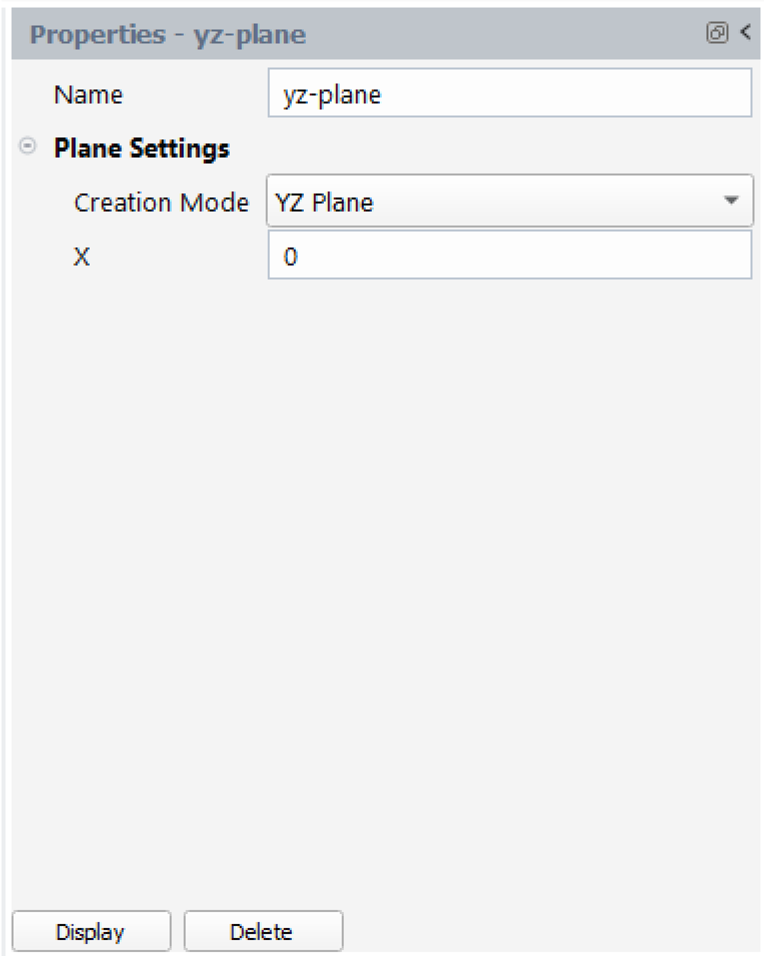
c. Enter 3 for **Z Velocity**.

d. Click **Initialize**.

*During initialization, the octree mesh for the domain is generated. Specifying the* ***Time Step Size*** *is required to properly initialize the flow field. The* ***Compute From*** *option can be used to specify* ***Initial Values*** *from all zones in the domain by selecting* ***All Zones****. Alternatively,* ***Initial Values*** *can be computed using data from an Ansys Fluent data file generated on the same mesh. Note that this option will only appear after the Ansys Fluent case file has been imported. For more information on specifying domain parameters for the octree mesh and initialization see sections* *Defining the Domain Parameters for Meshing (p. 158)* *and* *Calculating a Solution in the Fluent Beta Features Manual (p. 175)*.

4. Create YZ, ZX, and XY planes for solution animations and plotting of results.

**Results → Surfaces → New → Plane...**




a. Enter `yz-plane` for **Name**.

b. Select **YZ Plane** under the **Creation Mode** drop down list.

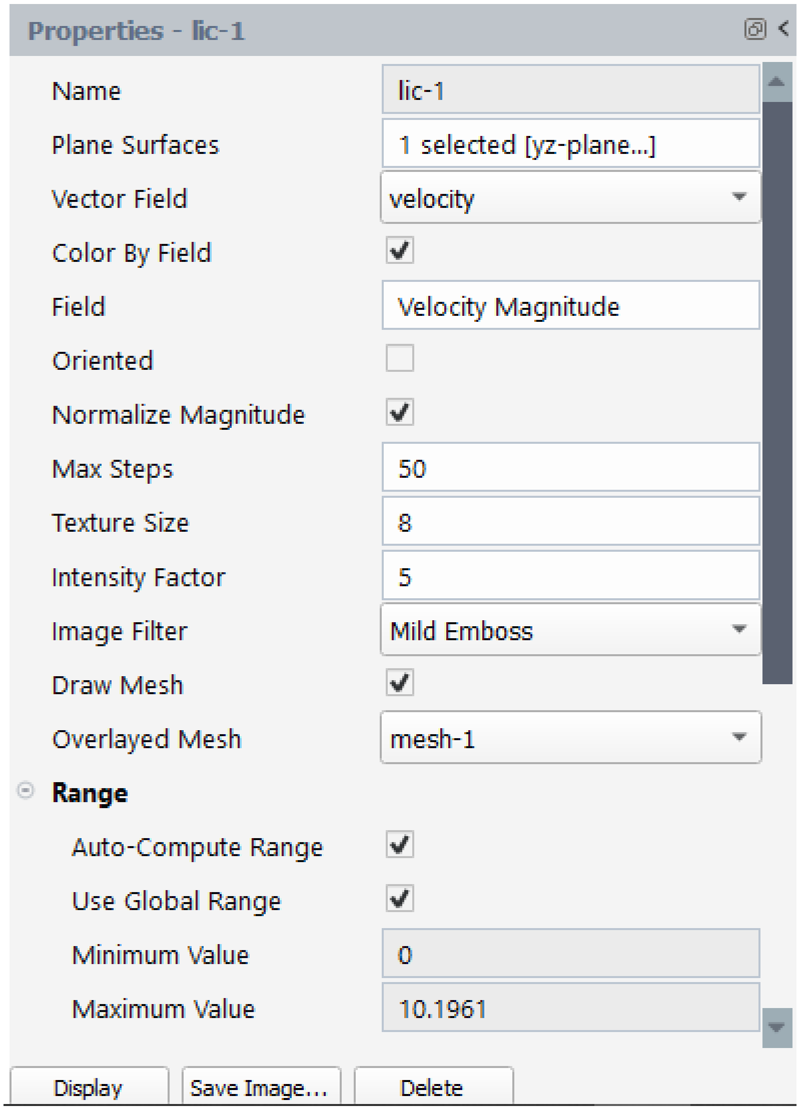
c. Retain the value of `0` for **X**.

d. Repeat the above steps for creating the `zx-plane` and the `xy-plane` by entering a value of 4 for **Y** and retaining the value of `0` for **Z** respectively.

*Alternatively, you can create multiple planes and/or multiple iso-surfaces at once, by clicking **Create Multiple Planes...** or **Create Multiple Iso-Surfaces...**. For details on creating multiple planes and iso-surfaces see Plane Surfaces (p. 185) and Iso-Surfaces in the Fluent Beta Features Manual (p. 189).*

5. Display LIC of velocity magnitude on the yz plane.

**Results** → **Graphics** → **LICs** → **New...**




a. Click the **Plane Surfaces** list and select **yz-plane**.

b. Click **OK** to confirm the selection and close the **Surfaces** dialog box.

c. Retain the default selections of **velocity** and **Velocity Magnitude** from the **Vector Field** and **Field** properties respectively.

d. Click the **Image Filter** list and select **Mild Emboss**.

e. Disable **Oriented**.

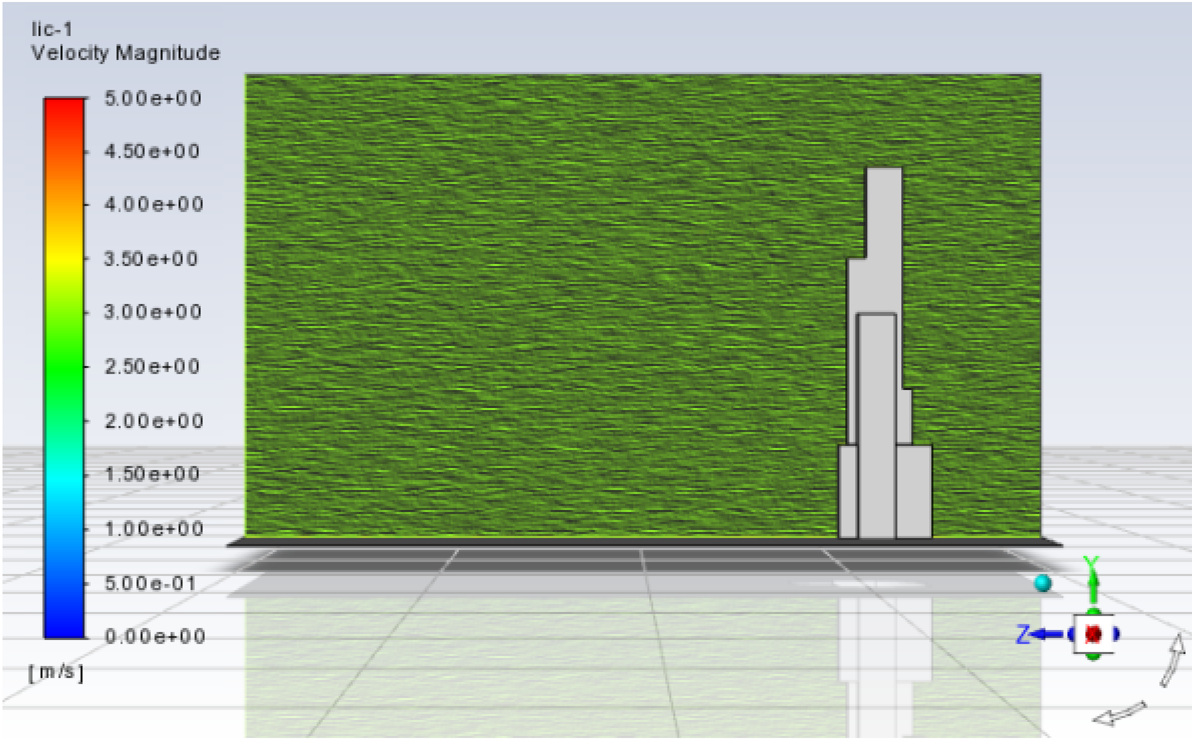
f. Enter `50` for **Max Steps**.

g. Enable **Draw Mesh** and select **mesh-1** from the **Overlayed Mesh** list.

h. Retain the remaining default settings.

i. Click **Display**.

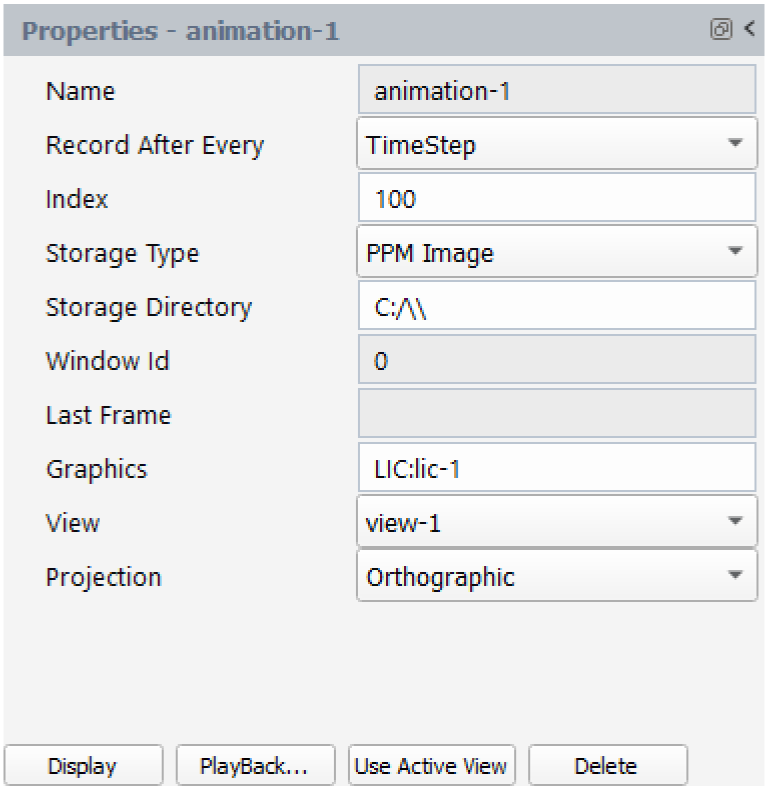
**Figure 27.31: LIC of Velocity Magnitude on the YZ Plane After Initialization**




*For more information on setting up an **LIC** plot in **Fluent (LB)** see* *Postprocessing Results in the Fluent Beta Features Manual (p. 181)* .

6. Create a solution animation for the LIC of velocity magnitude.

   **Solution** → **Calculation Activities** → **Solution Animations** → **New...**




a. Retain the default selection of **TimeStep** for **Record After Every**.

b. Retain the default value of `100` for **Index**.

c. Click the **Storage Type** drop-down list and select **PPM Image**.

   *For larger 2D or 3D cases, saving animation files with the PPM Image option is preferable, to avoid using too much of your machine's memory.*

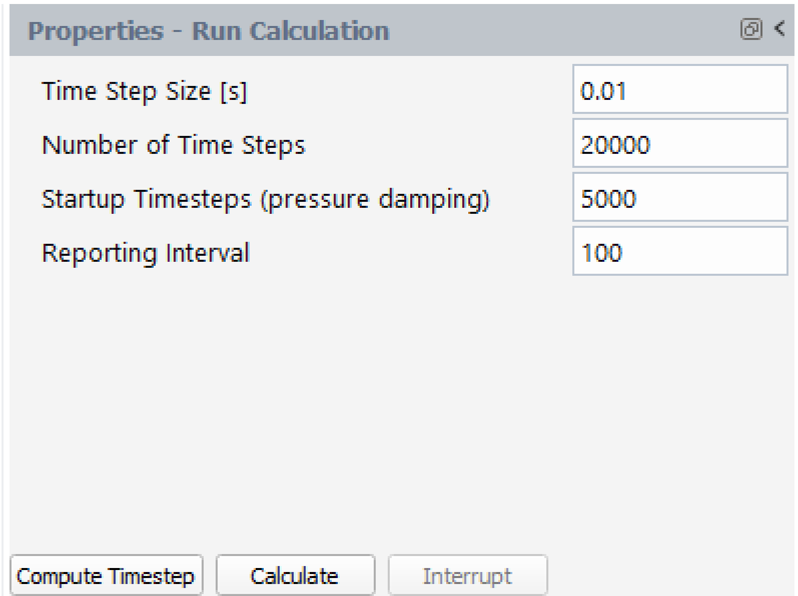
d. Click the **Graphics** field to open the **Graphics** dialog box.

e. Select LIC and `lic-1` from the **Type** and **Graphics** lists respectively.

f. With the LIC of velocity magnitude displayed as shown in Figure 27.31: LIC of Velocity Magnitude on the YZ Plane After Initialization (p. 245) above, click **Use Active View**.

7. Start the calculation by requesting 20,000 time steps.

   **Solution → Run Calculation**




a. Enter `20000` for **Number of Time Steps**.

b. Enter `5000` for **Startup Timesteps (pressure damping)**.

c. Click **Calculate**.

---

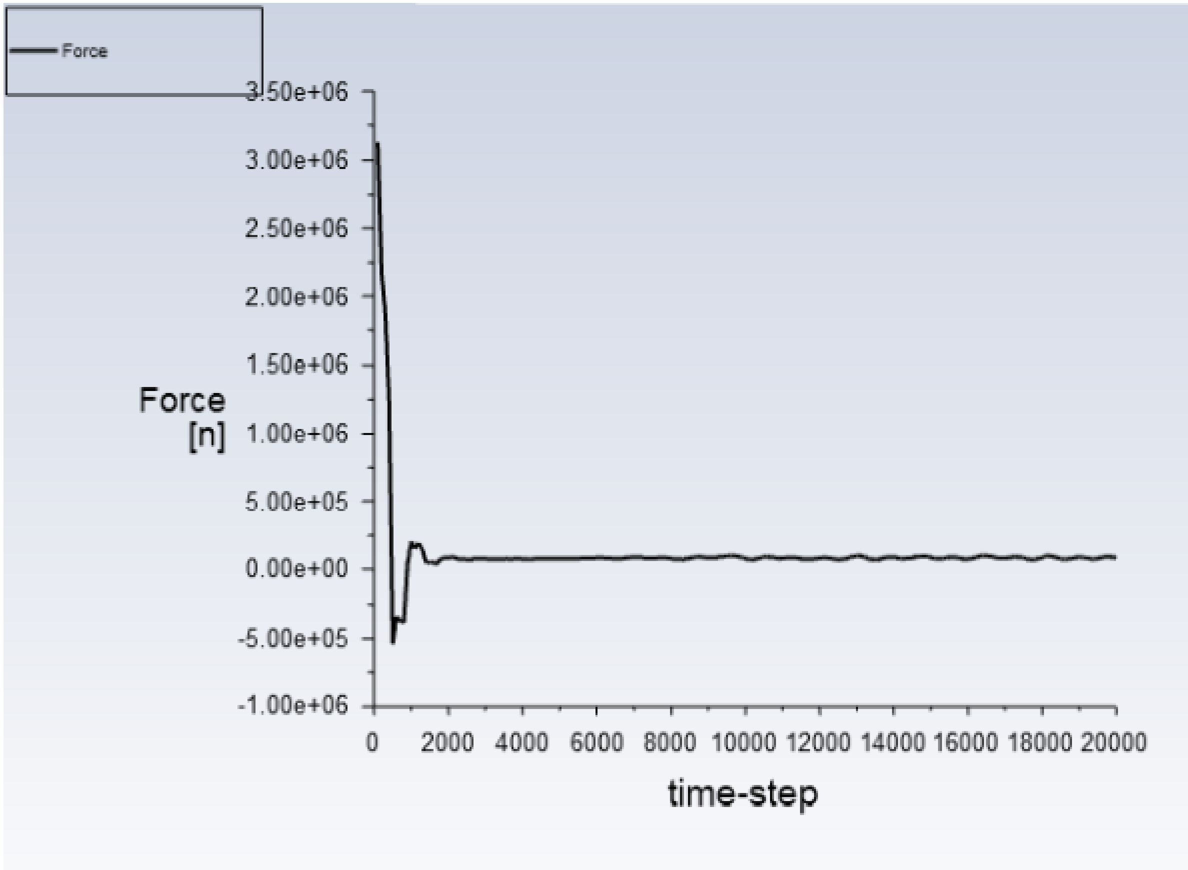
**Note:**

**Startup Timesteps (pressure damping)** helps to reduce early solution instability as the solver begins calculating.

---

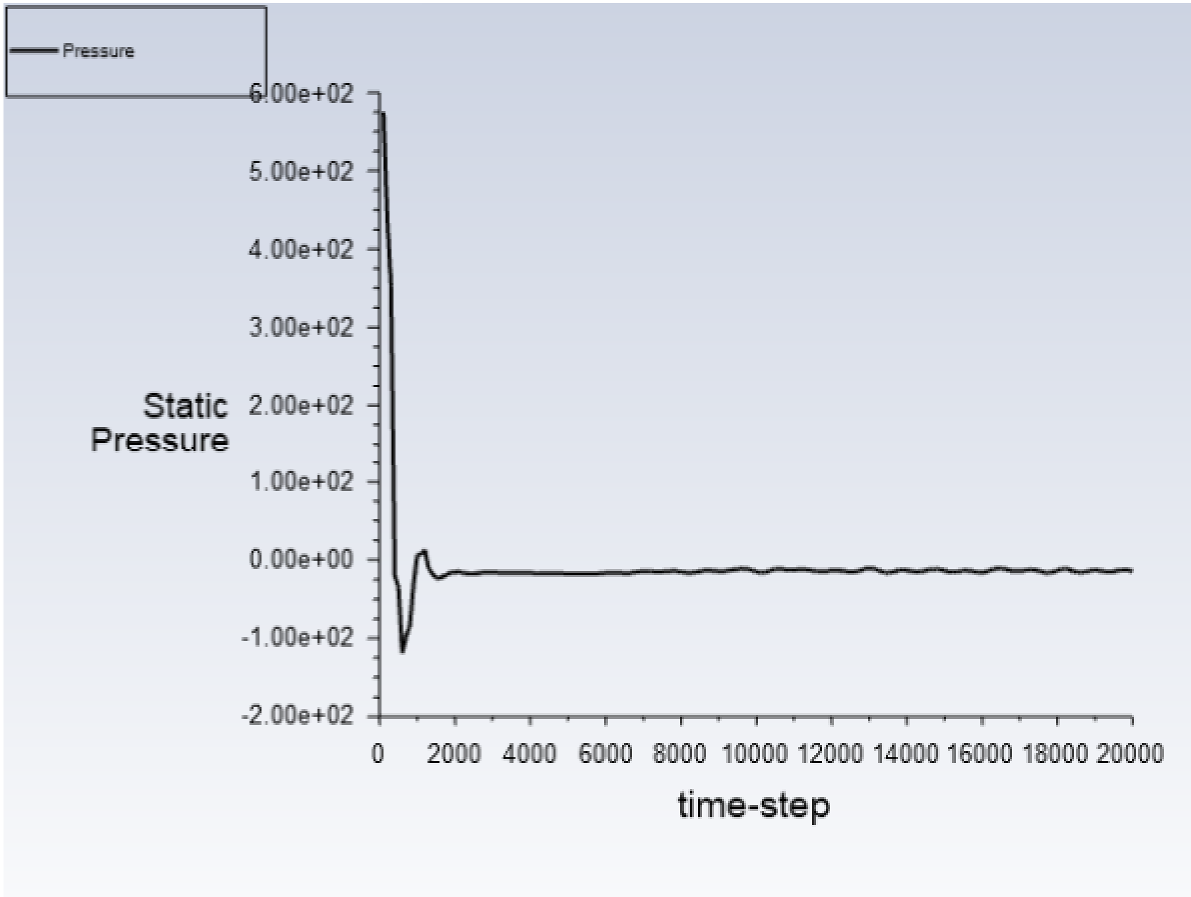
8. Convergence history of force on the front skyscraper walls.

**Figure 27.32: Convergence History of Force**




9. Convergence history of static pressure on the back skyscraper walls.

**Figure 27.33: Convergence History of Force**



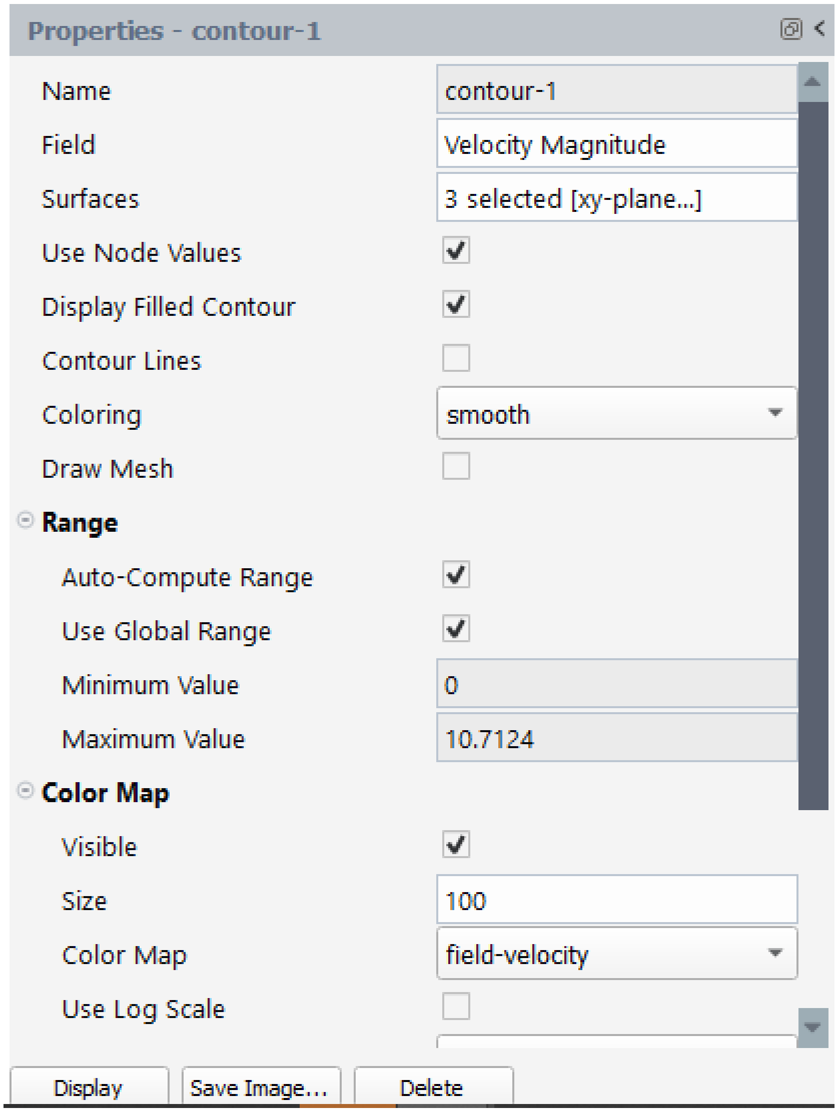

10. Save the case and data files (`Skyscraper.cas.lb` and `Skyscraper.dat.lb`).

**File → Write → Case & Data...**

## 27.3.3.6. Postprocessing

1. Display contours of velocity magnitude on the yz, zx, and xy planes.

**Results → Graphics → Contours → New...**




a. Click the **Field** list and select **Velocity** and **Velocity Magnitude** from the **Section** and **Field** columns, respectively.

b. Click **OK** to confirm the selection and close the **Field Variable** dialog box.

c. Click the **Surfaces** list and select **Plane**. Which selects all of the plane surfaces at once.

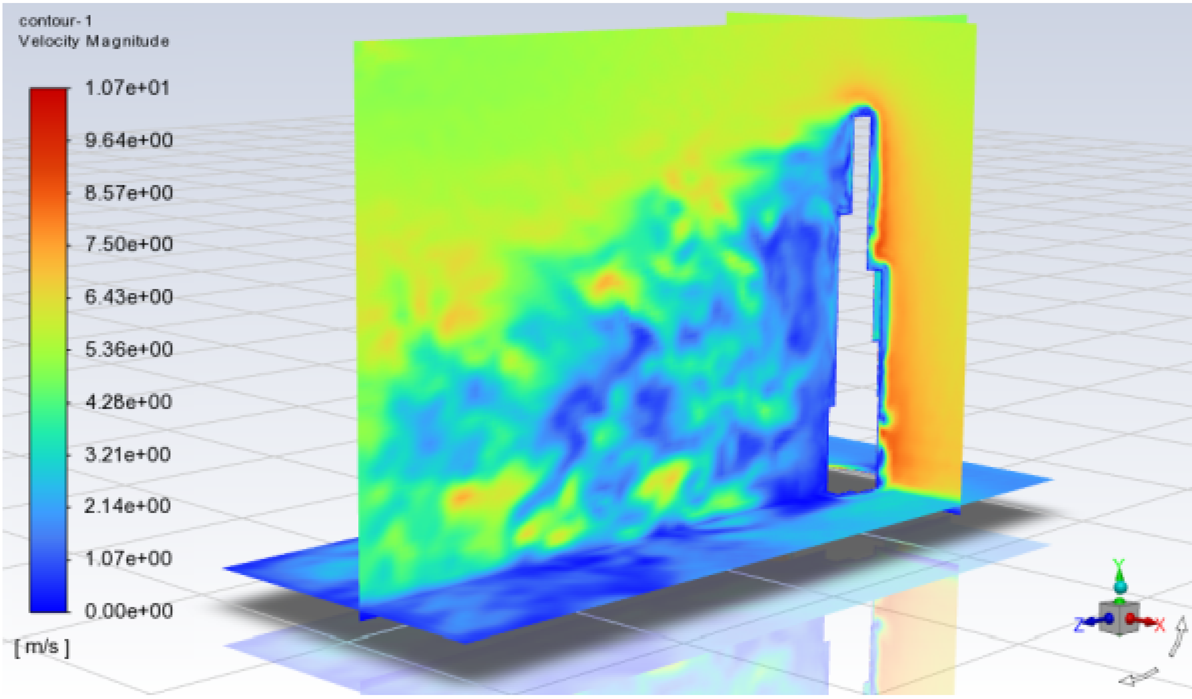
d. Click **OK** to confirm the plane selections and close the **Surfaces** dialog box.

e. Disable the **Headlight** and **Lighting** options in the **View** ribbon tab.

f. Retain the remaining default settings.

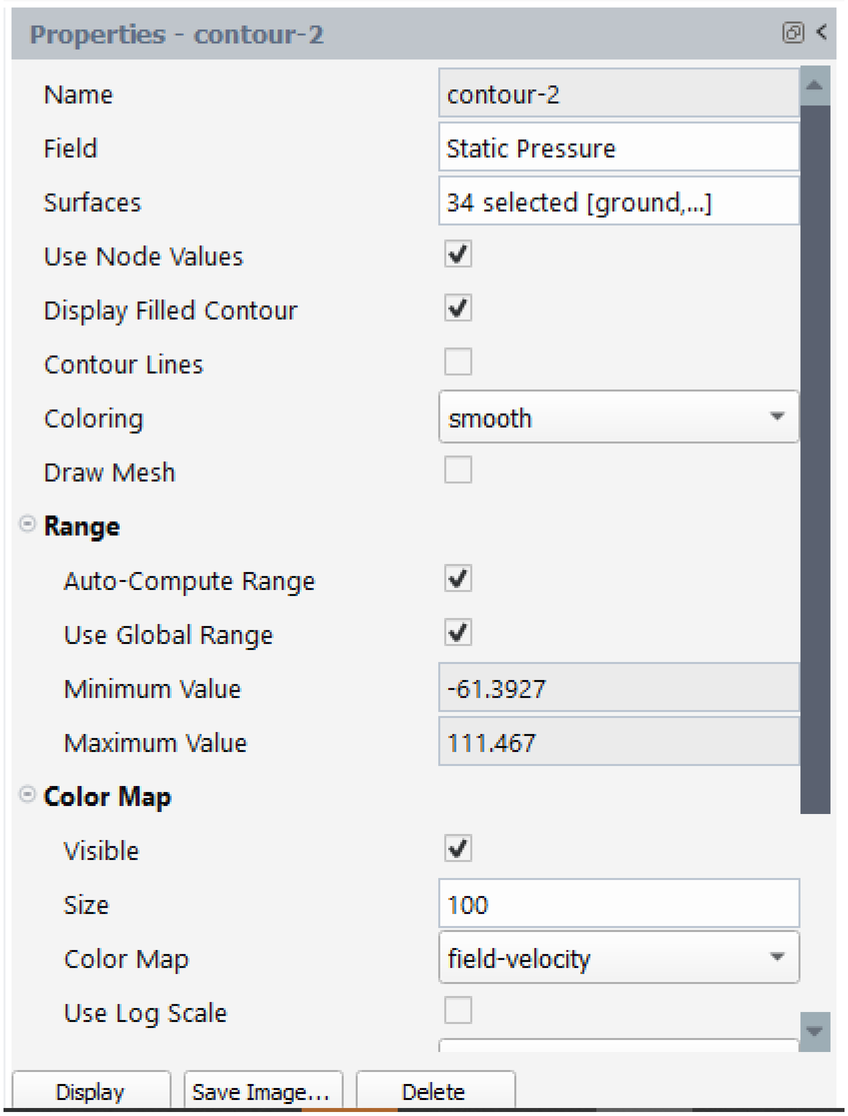
g. Click **Display**.

**Figure 27.34: Contours of Velocity Magnitude on the YZ, ZX, and XY Planes**




2. Display contour of static pressure on the ground and skyscraper walls.

   **Results** → **Graphics** → **Contours** → **New...**




a. Click the **Field** list and select **Pressure** and **Static Pressure** from the **Section** and **Field** columns, respectively.

b. Click **OK** to confirm the selection and close the **Field Variable** dialog box.

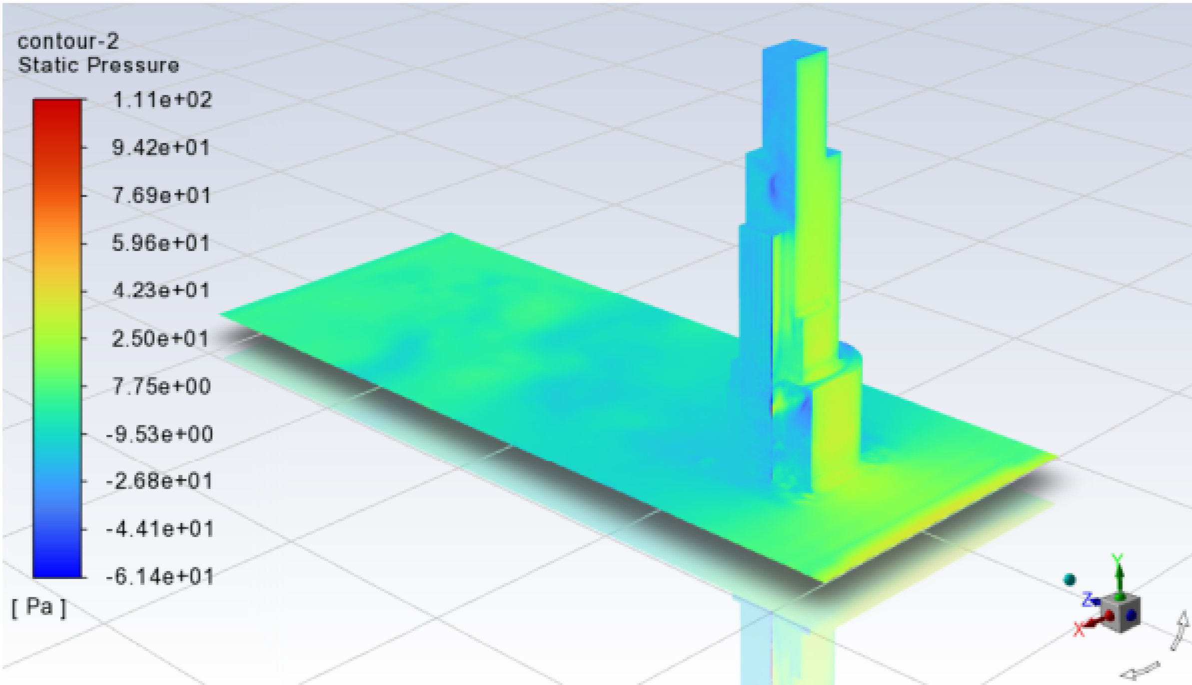
c. Click the **Surfaces** list and select **Wall**.

d. Click **OK** to confirm the wall selections and close the **Surfaces** dialog box.

e. Retain the remaining default settings.

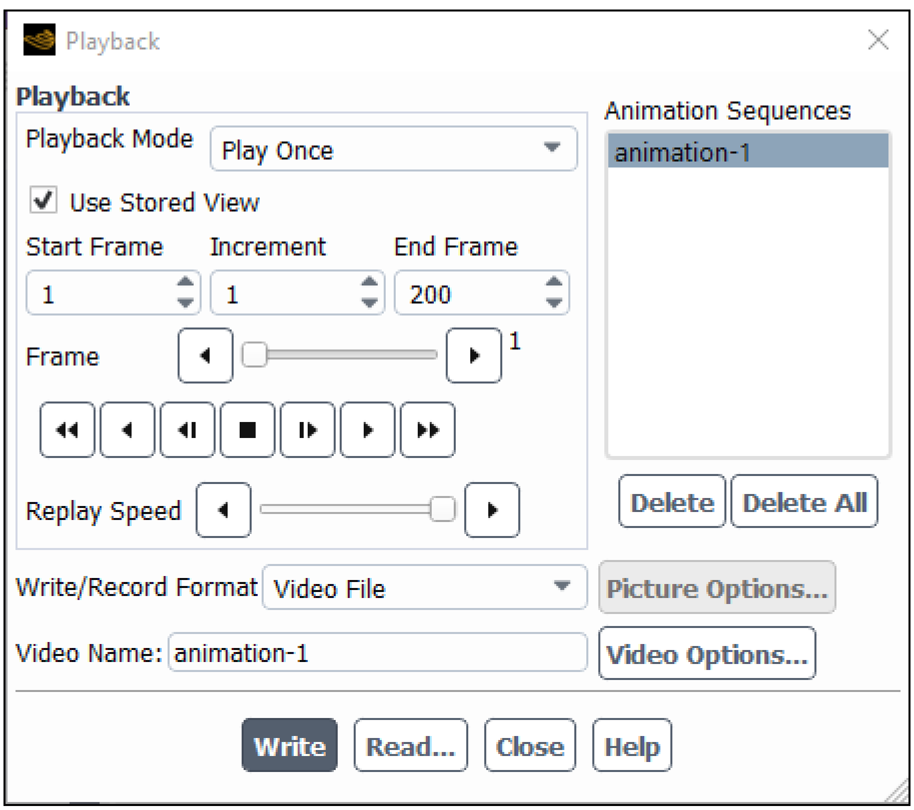
f. Click **Display**.

**Figure 27.35: Contours of Static Pressure on the Skyscraper Walls**




3. Playback the solution animation for the LIC of velocity magnitude.

   **Solution** → **Solution Animations** → **animation-1** → **PlayBack...**



a. Click the play button (the second from the right in the group of buttons in the **Playback** group box).

## 27.3.4. Summary

This tutorial demonstrated how to set up and solve an aerodynamics problem using Fluent Lattice Boltzmann with the Smagorinsky turbulence model.

# Chapter 28: Fluent Materials Processing Workspace

When beta features are enabled, you are able to simulate various types of polymer material processes (for example, extrusion and blowmolding) within the Fluent workspace environment.

This chapter discusses how to use the Fluent Materials Processing workspace. Information is organized into the following sections:



## 28.1. Introduction

The Fluent Materials Processing Workspace is a means to explore manufacturing applications such as polymer extrusion, blowmolding, fiber spinning, etc. within the Fluent environment. Many of the models and concepts in this workspace are based on Ansys Polyflow, details of which can be found in the Ansys Polyflow documentation.



### 28.1.1. Program Capabilities

The workspace supports the following features:

- Extrusion / Fiber spinning
- Blow Molding / Thermoforming
- Pressing / Forming

The supported geometry types include:

- 2D planar, 2D axisymmetric

- Shell
- 3D

The supported solver attributes include the following:

- Finite Element Method
- Steady-state
- Time-dependent / Transient
- Generalized Newtonian (isothermal & non-isothermal)

### 28.1.2. Installation and Licensing Requirements

The Fluent Materials Processing Workspace requires the following components.

- CFD-Enterprise license
- Ansys Polyflow
- Ansys EnSight
- A valid `%HOME%` environment variable assignment

### 28.1.3. Known Limitations

- When using a limited Academic License, you cannot load a mesh containing more than 512,000 elements.

## 28.2. Basic Steps for CFD Analysis Using the Fluent Materials Processing Workspace

Before you begin your CFD analysis using Fluent Materials Processing, careful consideration of the following issues will contribute significantly to the success of your modeling effort. Also, when you are planning a CFD project, be sure to take advantage of the customer support available to all Ansys Fluent users.

For more information, see the following section:



### 28.2.1. Steps in Solving Your Materials Processing CFD Problem

The primary workflow for using the Materials Processing workspace is as follows:

1. Define the modeling goals.
2. Create a surface mesh to represent the material processing domain (fluids, solids, and/or molds, etc.). This can be done using either a Polyflow or a Fluent mesh.

3. Setup your simulation (for example, simulation type/goals, cell zone definitions, material properties, boundary condition settings, solution constraints, and so on) using either of the following methods.
   - Manually create and define objects (using the Ribbon or the Outline View) to define all aspects of the simulation based on your requirements.
   - Use a specialized template that automatically defines most of your simulation objects based on key simulation type/goals. You can then revisit the settings and edit them as needed.
   - Use a specialized wizard that can set up a simulation (and optionally calculate a solution) once you provide a few key inputs (applicable for simple direct extrusion or inverse extrusion problems)
4. Compute the solution. Use default solution settings, or adjust them accordingly, in order to obtain an accurate CFD solution.
5. Examine and save the results. Use graphical analysis tools (such as contour plots, vector plots, and pathline plots) to better visualize your simulation results.
6. Consider revisions to the setup, if necessary.

The details of these steps are covered in the sections that follow.

---

**Note:**

You can also save and later open session case files (`*.mprcas`) that contain your simulation settings. Transcripts of your work (`*.trn`) within a session can also be saved. In addition, you record and later run journal files (`.py`).

---

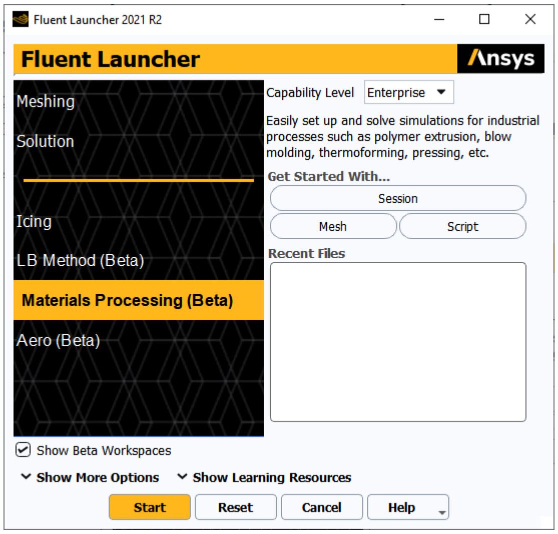
## 28.3. Starting and Exiting the Fluent Materials Processing Workspace

You can start and exit the workspace as described in the following sections:



### 28.3.1. Starting the Materials Processing Workspace Using the Fluent Launcher

1. Open the Fluent Launcher. *Refer to Starting Ansys Fluent Using Fluent Launcher in the Fluent User's Guide for additional information on opening the Fluent Launcher.*
2. Select **Enterprise** from the **Capability Level** drop-down list.
3. Select **Show Beta Workspaces**, located below the list of workspaces. Additional beta-level workspaces will then be made available in the list.
4. Select **Materials Processing (Beta)** from the list of workspaces.




5. Click **Start** to start the workspace. Alternatively, you can also:

   - Select a new **Session** file (`*.mprcas`). These files contain the commands needed to repopulate the workspace with your previous simulation settings.
   - Select a Polyflow or Fluent **Mesh** file.

     **Note:**

     Make sure that the mesh file you select exists in a folder where you have write permissions, otherwise, the Fluent Materials Processing workspace will automatically set this folder to be the working folder and will be unable to save files properly as a result.

   - Select a **Script** file.
   - Select an item in the **Recent Files** list (after you have used the workspace).

   If you use one of these methods, click **Start With Selected Options** to start the workspace.

**Important:**

Fluent Materials Processing is only available at the **Enterprise** licensing level.

## 28.3.2. Starting the Materials Processing Workspace Using the Command Line

To start Fluent Materials Processing from the command line, enter the following in a command prompt window:

```
../fluent/bin/matrpro [-prj] | [-R] | [-msh] <file-name>
```

where

**-prj**

Use this option when `<file-name>` is a session file.

**-R**

Use this option when `<file-name>` is a script file.

**-msh**

Use this option when `<file-name>` is a Polyflow or Fluent mesh file.

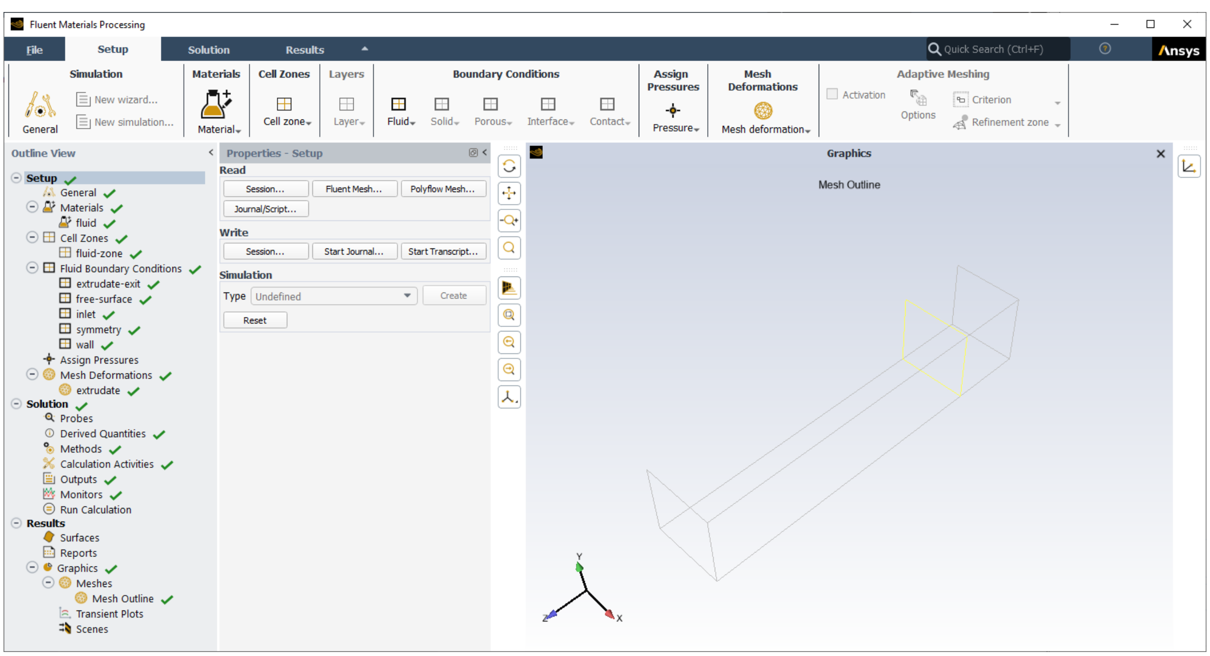
### 28.3.3. Exiting the Materials Processing Workspace

You can exit Fluent Materials Processing by selecting **Exit** in the **File** ribbon tab or by clicking the ✕ button in the top right corner of the application. For the latter, a **Question** dialog box will open to confirm if you want to proceed.

## 28.4. Graphical User Interface (GUI)

The graphical user interface (GUI) of Fluent Materials Processing is very similar to the solution mode of Fluent, in that:

- it includes a ribbon, an outline view, a graphics window, a console, toolbars, dialog boxes, and quick search (as described in GUI Components in the *Fluent User's Guide*)
- it can be modified (as described in Customizing the Graphical User Interface in the *Fluent User's Guide*)
- it can be set to match your preferences (as described in Setting User Preferences/Options in the *Fluent User's Guide*)
- it has access to the help system (as described in Using the Help System in the *Fluent User's Guide*)

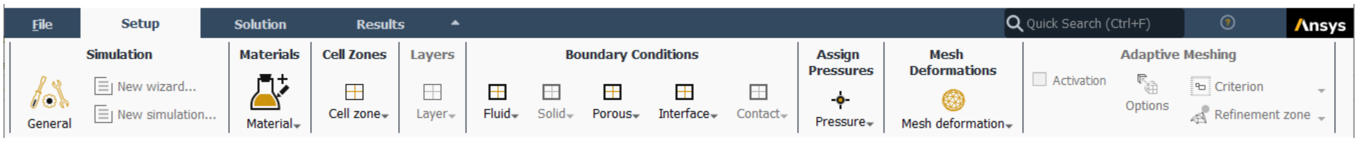


When setting up your simulation, the majority of your actions will start by performing actions in the **Ribbon** or the **Outline View**, and the associated **Properties** panels.

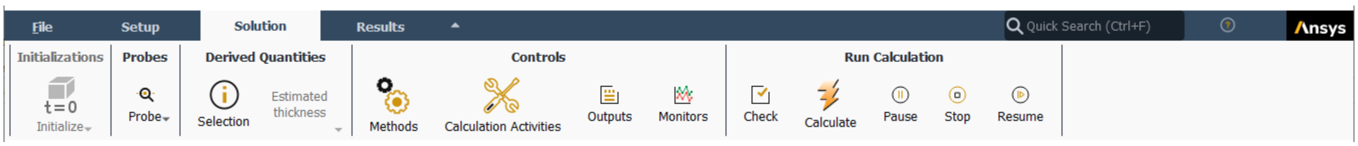
## Ribbon

The workspace Ribbon represents a horizontal workflow, and is structured similarly to, and interacts with, elements of the Outline View. As you progress from left to right, creating or editing objects (such as create a **New...** or **Edit...** an existing object from the **Fluid Boundary Condition** item), elements of the ribbon become available, and objects become visible in the Outline View.

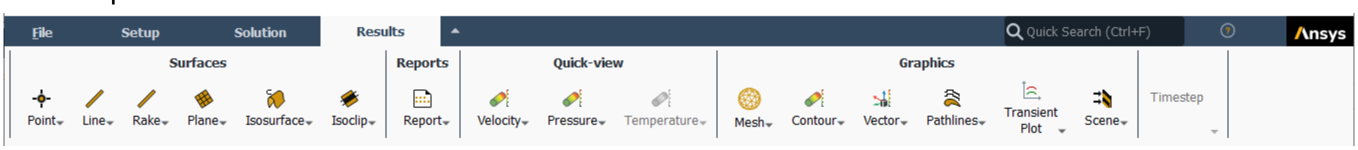
In the Ribbon, the simulation can be defined by adding objects to the **Setup** section of the ribbon.



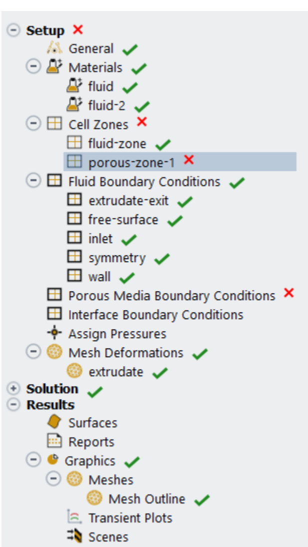

Solution-specific options are accessible from the **Solution** section




Data from a complete simulation can be analyzed by adding objects in the **Results** section. This includes predefined, 'quick-view' postprocessing objects such as velocity, pressure, or temperature (if applicable) contour plots.




## Outline View



In the Outline View, the simulation can be defined by adding objects to the **Setup** section of the tree, solution-specific options are accessible from the **Solution** section, while the data from a simulation can be analyzed by adding objects in the **Results** section.

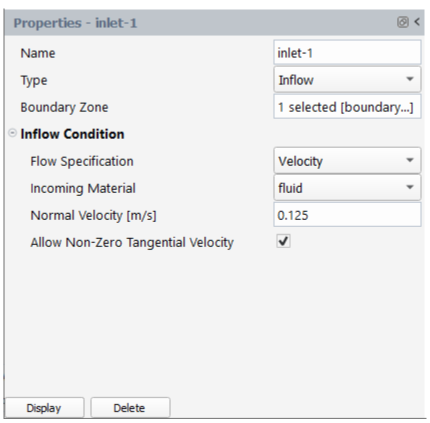
Icons help you visualize the status of an object in the Outline View. For instance,  represents the state of an object whose inputs are fully defined, whereas  represents the state where an object still requires attention, providing you with a quick visualization of the state of your simulation setup.

Right-click an item in the Outline View and use the menu that opens to perform a function (such as create a **New...** object from the **Fluid Boundary Condition** item); and you can left-click an item to open a related **Properties** window.

### Properties Window

The **Properties** windows are similar to task pages or dialog boxes, in that they allow you to define settings, run the calculation, and postprocess results. Note that your settings are saved as you define them in a **Properties** window, unlike dialog boxes (which you may open from the **Outline View** or settings in the **Properties** window) which require you to click an **Ok** button.

**Figure 28.1: The Properties Window for a Boundary**




Properties that require your attention will be highlighted in yellow.

### Console Window

Note that the console does not interact with a text user interface (TUI), but does allow Python scripting; however, you should instead read Python commands from a journal / script file, as this allows for each command to complete before the next command is executed. This is especially important for interdependent commands. Typing or pasting a series of Python commands directly into the console could result in undesirable behavior, as there is no waiting for the previous command to complete before the next command is executed.

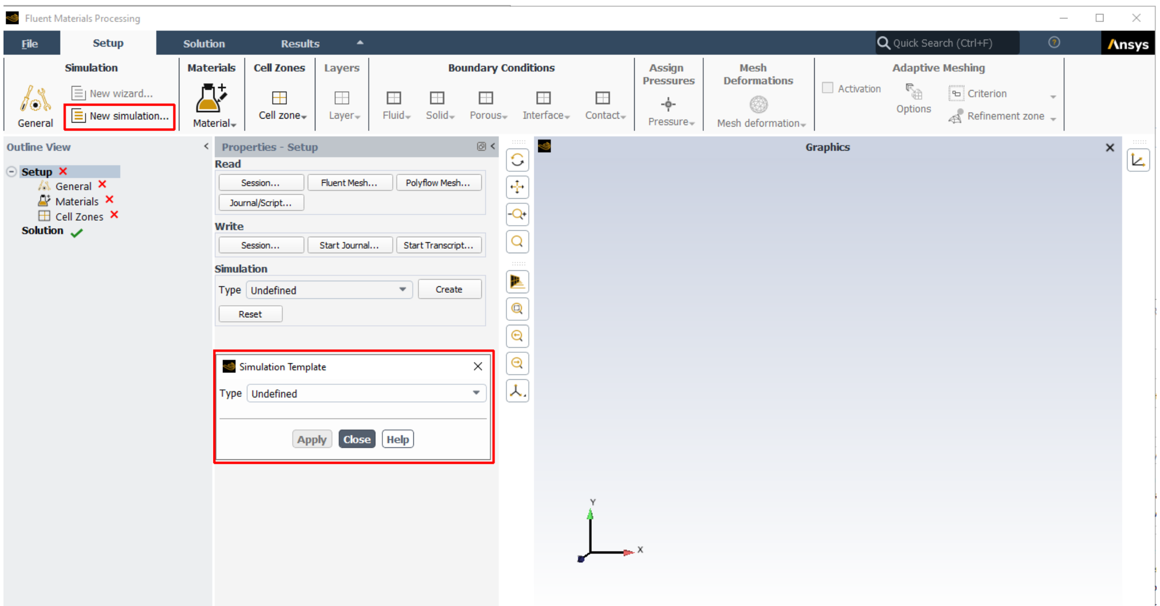
## 28.5. Getting Started

Upon launching the workspace, you typically can get started as follows:

1. Create a new simulation. There are several means of creating your simulation:

- Manually create and define objects such as simulation type/goals, cell zone definitions, material properties, boundary condition settings, solution constraints, and so on to define all aspects of the simulation based on your requirements. You can progress left-to-right in the Ribbon or from top-to-bottom using the Outline View in order to create and edit objects accordingly.
- Click the **New simulation...** button, under the **Simulation** category, in the **Setup** Ribbon. This displays the **Simulation Template** dialog.

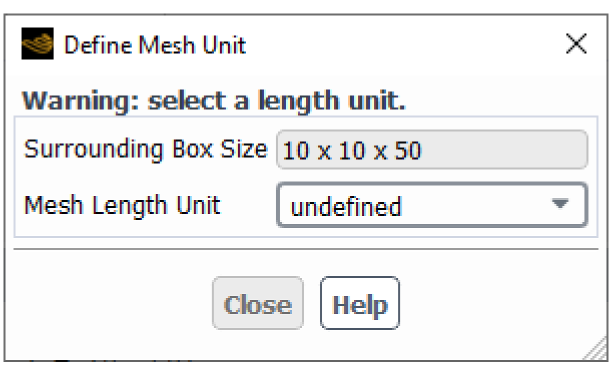
**Figure 28.2: Creating a New Simulation Using the Ribbon**



- Alternatively, you can also use the **Setup** properties panel.
  - Double-click **Setup** in the Outline View to expose the **Setup Properties**.
  - Read in a mesh (either a Fluent or a Polyflow mesh file). You can also read in an existing journal or Materials Processing session file.

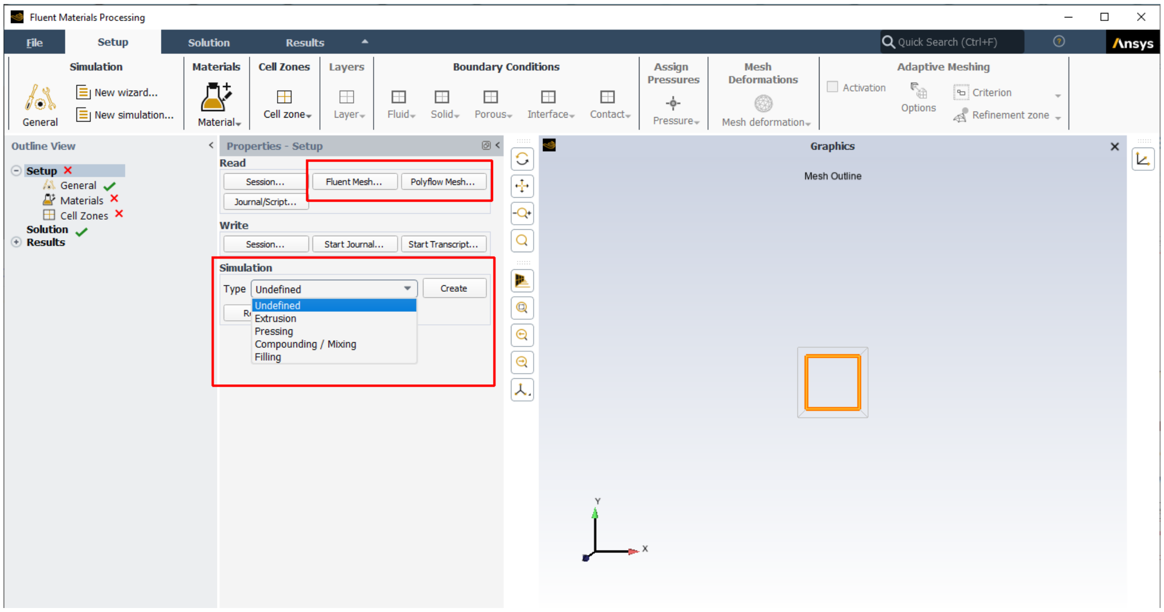
    If prompted, use the **Define Mesh Unit** dialog to specify the **Mesh Length Unit** for the selected mesh's length units.

**Figure 28.3: The Define Mesh Unit Dialog**



– Make your selection under **Simulation**

**Figure 28.4: Creating a New Simulation Using the Properties of the Setup**



2. In either case, you can then define a specific type of **Simulation** (see Choosing a Simulation Template (p. 263)).
3. Once the simulation type and associated settings have been made, click **Create** to create the simulation setup in the workspace.

   Setting up your simulation using this approach will automatically populate the Outline View with typical simulation-specific objects and predefined settings. If you are setting up your simulation without the aid of a template or the wizard, you will have to manually create the various objects and specify their properties yourself.

   You can revert to a clean workspace by clicking the **Reset** button. This will remove all objects from the Outline View and resets the workspace to a new state.

## 28.6. Choosing a Simulation Template

The following polymer simulation types are available using templates:

- **Extrusion** (Using the Extrusion Template (p. 264)) - available in 2D and 3D.
- **Blow Molding & Thermoforming** (Using the Blow Molding & Thermoforming Template (p. 265)) - available in shell only.
- **Pressing** (Using the Pressing Template (p. 265))
- **Compounding and Mixing** (Using the Compounding & Mixing Template (p. 266)) - available in 2D and 3D.
- **Film Casting** (Using the Film Casting Template (p. 266)) - available in 2D only.

- **Filling** (Using the Filling Template (p. 266)) - available in 2D and 3D.

### 28.6.1. Using the Extrusion Template

The **Extrusion** template helps you quickly set up a simulation where you can investigate the extrusion of molten polymers.

1. Specify the extrusion simulation **Goal**

   - Select **Analyze flow within die** when you want to investigate the flow of the molten polymer within just the die itself. With this option, several boundary conditions are already provided for you as part of the template:

     – **inlet**: an open boundary where fluid is entering the physics region.

     – **outlet**: an open boundary where fluid is exiting the physics region.

     – **wall**: a closed boundary across which fluid cannot flow into or out of the physics region.

   - Select **Predict extrudate shape** when you want to determine the shape of the extrudate based on the geometry of the die, material characteristics, and operating conditions. With this option, several boundary conditions are already provided for you as part of the template:

     – **extrudate deformation**: the deformation of the polymer extrudate.

     – **free surface**: a boundary that represents the outer surface of the polymer extrudate.

     – **extrudate exit**: an open boundary where the extrudate is exiting the physics region.

     – **inlet**: an open boundary where fluid is entering the physics region.

     – **wall**: a closed boundary across which fluid cannot flow into or out of the physics region.

   - Select **Determine die lip shape** when you want to determine the shape of the die lip corresponding to the desired cross section of the extrudate. With this option, several boundary conditions are already provided for you as part of the template:

     – **extrudate deformation**: the deformation of the polymer extrudate.

     – **free surface**: a boundary that represents the outer surface of the polymer extrudate.

     – **die deformation**: the deformation of the die.

     – **extrudate exit**: an open boundary where the extrudate is exiting the physics region.

     – **inlet**: an open boundary where fluid is entering the physics region.

     – **wall**: a closed boundary across which fluid cannot flow into or out of the physics region.

   **Note:**

   In either case, you may also want to take advantage of any symmetry in your simulation. In such cases, you can choose to add a separate **symmetry** boundary condition. A symmetry boundary represents a closed boundary at a plane of symmetry across which fluid is not expected to flow.

2. Specify the **Number of Fluids**.

3. Specify the **Number of Restrictors**.

4. Specify whether or nor to include **Thermal** considerations.

5. Specify whether or nor to include **Conjugate Heat Transfer** considerations.

When you have made your selections, click **OK** to get started. Each of the elements in the **Outline View** will have settings specific to the choices you make here in the template.

### 28.6.2. Using the Blow Molding & Thermoforming Template

The **Blow Molding & Thermoforming** template helps you quickly set up a simulation where you can investigate the blow molds and thermoformed polymers.

1. Specify the **Number of Fixed Molds**.

2. Specify the **Number of Moving Molds**.

3. Specify the **Number of Layers**.

4. Specify the **Duration** for the simulation.

5. Specify whether or nor to include **Thermal** considerations.

When you have made your selections, click **OK** to get started. Each of the elements in the **Outline View** will have settings specific to the choices you make here in the template.

### 28.6.3. Using the Pressing Template

The **Pressing** template helps you quickly set up a simulation where you can investigate polymer pressing phenomenon.

1. Specify the **Number of Fixed Molds**.

2. Specify the **Number of Moving Molds**.

3. Specify the **Duration** for the simulation.

4. Specify whether or nor to include **Thermal** considerations.

5. Specify whether or nor to include **Conjugate Heat Transfer** considerations.

When you have made your selections, click **OK** to get started. Each of the elements in the **Outline View** will have settings specific to the choices you make here in the template.

### 28.6.4. Using the Compounding & Mixing Template

The **Compounding & Mixing** template helps you quickly set up a simulation where you can investigate the combination of molten polymers.

1. Specify the **Type** of compounding and mixing simulation
   - Select **Batch Mixer** when you want to perform an analysis simulating a batch mixer.
   - Select **Extruder** when you want to perform an analysis simulating an extruder.
2. Specify the **Number of Moving Parts**.
3. Specify the **Duration** for the simulation.
4. Specify whether or nor to include **Thermal** considerations.

When you have made your selections, click **OK** to get started. Each of the elements in the **Outline View** will have settings specific to the choices you make here in the template.

### 28.6.5. Using the Film Casting Template

The **Film Casting** template helps you quickly set up a simulation where you can investigate polymer film casting.

1. Specify the **Number of Layers**.
2. Specify whether or nor to include **Thermal** considerations.

When you have made your selections, click **OK** to get started. Each of the elements in the **Outline View** will have settings specific to the choices you make here in the template.

### 28.6.6. Using the Filling Template

The **Filling** template helps you quickly set up a simulation where you can investigate polymer filling properties.

1. Specify the **Number of Inlets**.
2. Specify the **Number of Vents**.
3. Specify the **Duration** for the simulation.
4. Specify whether or nor to include **Thermal** considerations.

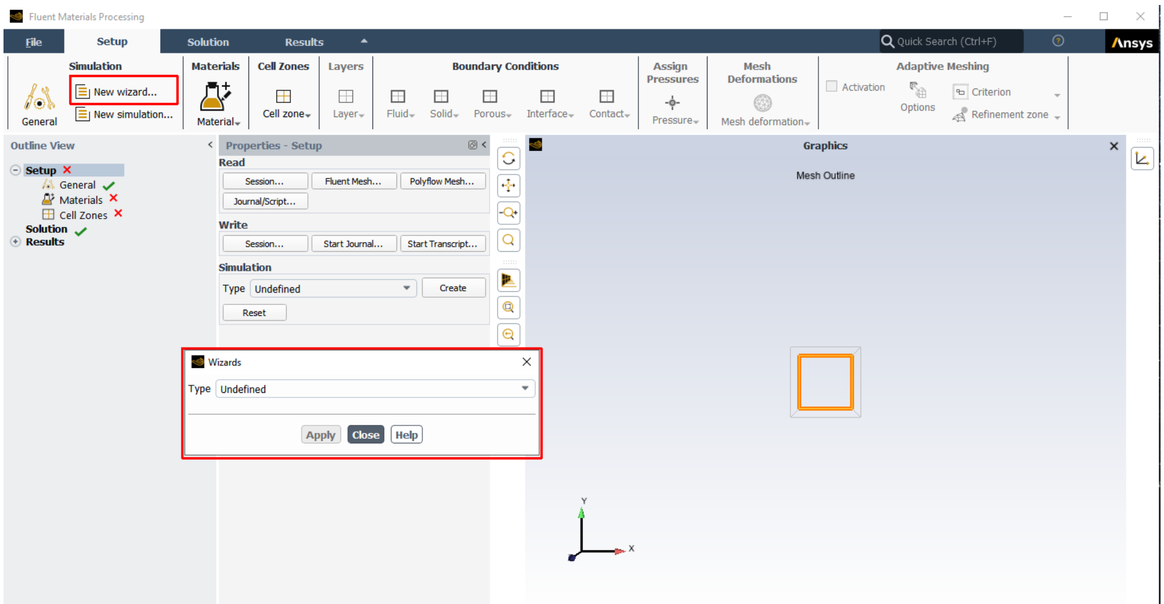
When you have made your selections, click **OK** to get started. Each of the elements in the **Outline View** will have settings specific to the choices you make here in the template.

## 28.7. Using the Simulation Wizard

The workspace provides a convenient way to set up the two common materials processing extrusion simulations through the use of specialized wizards.

1. Start the workspace.
2. Read in a mesh (either a Fluent or a Polyflow mesh file).
3. Click the **New wizard...** button, under the **Simulation** category, in the **Setup** Ribbon. This displays the **Wizards** dialog.

**Figure 28.5: Creating a New Simulation Using the Wizard**



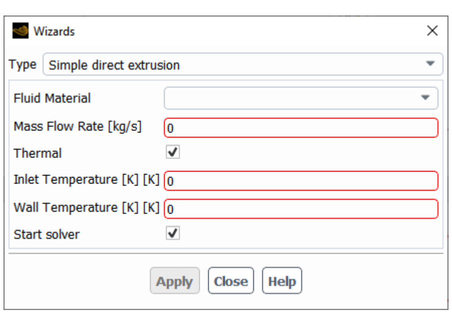

4. In the **Wizards** dialog, select an appropriate **Type**. You can choose from the **Simple direct extrusion** or the **Simple inverse extrusion** wizard.



## 28.7.1. Using the Simple Direct Extrusion Wizard

This wizard is available for you to quickly set up and solve an direct extrusion problem for a given mesh.

**Figure 28.6: The Simple Direct Extrusion Wizard**




1. Specify a **Fluid Material**.

2. Provide a value for the **Mass Flow Rate**.

3. Enable the **Thermal** option if temperatures are important in your simulation, in which case, you can then provide the **Inlet Temperature** and the **Wall Temperature**.

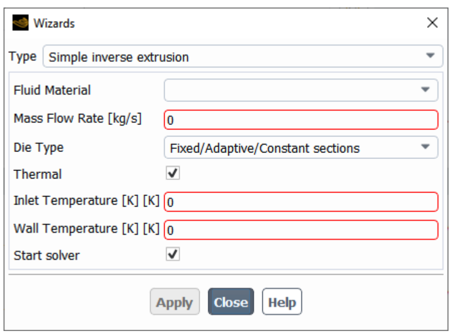
4. Use the **Start solver** option to automatically start solving the simulation once the simulation has been set up.

   If this option is enabled, once you click **Apply** and leave the wizard, the workspace takes your settings and sets up the extrusion simulation accordingly and solves the problem automatically. Once the calculations are completed, you can easily proceed to analyze the results.

## 28.7.2. Using the Simple Inverse Extrusion Wizard

This wizard is available for you to quickly set up and solve an inverse extrusion problem for a given mesh.

**Figure 28.7: The Simple Inverse Extrusion Wizard**




1. Specify a **Fluid Material**.

2. Provide a value for the **Mass Flow Rate**.

3. Specify the **Die Type**. Choices include:

   - **Fixed/Adaptive/Constant sections**
   - **Fixed/Adaptive sections**
   - **Adaptive/Constant sections**
   - **Adaptive sections**
   - **Constant sections**

4. Enable the **Thermal** option if temperatures are important in your simulation, in which case, you can then provide the **Inlet Temperature** and the **Wall Temperature**.

5. Use the **Start solver** option to automatically start solving the simulation once the simulation has been set up.

If this option is enabled, once you click **Apply** and leave the wizard, the workspace takes your settings and sets up the extrusion simulation accordingly and solves the problem automatically. Once the calculations are completed, you can easily proceed to analyze the results.

## 28.8. Setting Up Your Simulation

The details of your simulation setup depends on the type of template you have chosen, however, you can use the Outline View or the Ribbon to progress through the setup process. For instance, set your general simulation properties, followed by material properties, boundary condition properties, and solution properties, etc.

- **General** properties: used to define the overall simulation. Here, you can view and adjust any geometry, mesh, or calculation settings, as well as assorted physics options.
- **Material** properties: used to define the chemical and physical properties of a substance in your simulation.
  - Select **New...** to create a new object with its own unique properties.
  - Alternatively, select **Import from library** to open the **Library of Materials** dialog. Here, you can choose a material **Family**, and a corresponding **Material** to use in your simulation.
  - Alternatively, select **Import from file** to browse for and select a separate materials library file and add it to your simulation.
  - For existing materials, select **Export** to save the material definition to a separate file. Click **Fitting** to open Ansys Polymat where you can perform a fitting of your material data.
- **Cell Zone** properties: where you can categorize various portions of your geometry (mesh).
  - Use the **New...** button to create a new cell zone object with its own unique properties. Depending on the type of simulation you are setting up, cell zones can be one of the following types:
    - → **Fluid**
    - → **Solid**
    - → **Porous media**
    - → **Fixed mold**
    - → **Moving mold**
    - → **Restrictor**
    - → **Moving part**
  - Use the **Display** button to show an existing cell zone in the graphics window.
  - Use the **Delete** button to remove an existing cell zone

- **Boundary Condition** properties: where you define the physical conditions at specific boundaries in your simulation, depending on the type of simulation you are setting up.

  Use the **New...** button to create a new boundary condition object with its own unique properties. Depending on the type of simulation you are setting up, boundary conditions can be one of the following types:

  - **Inflow**
  - **Outflow**
  - **Wall**
  - **Symmetry**
  - **Free surface**
  - **Extrudate exit**
  - **Vent**
  - **Zero velocity**
  - **Zero force**
  - **Porous wall**

  If temperature changes are being considered, then there can also be **Thermal Conditions** associated at the boundaries as well.

- **Assign Pressures** properties: (optional) allow you to set the pressure level for a fluid flow simulation within a closed domain.

  Use the **New...** button to create a new pressure assignment object with its own pressure value and its own location point.

- **Mesh Deformations** properties: (optional) allow you to define how the mesh deforms in your simulation.

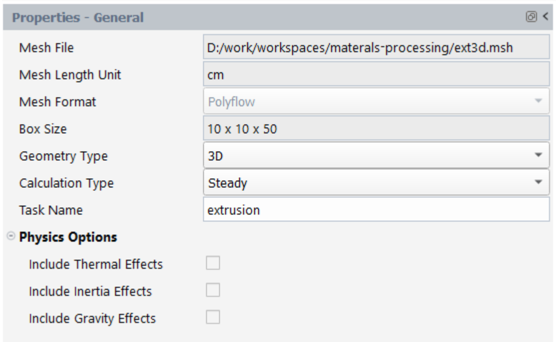
  Use the **New...** button to create a new mesh deformation object. Several types are available, including:

  - **Extrudate**
  - **Constant Die Section**
  - **Adaptive Die Section**

## 28.8.1. General Simulation Settings

Use the **General** node to access general properties of the selected simulation (Figure 28.8: General Simulation Properties (p. 271)).

**Figure 28.8: General Simulation Properties**

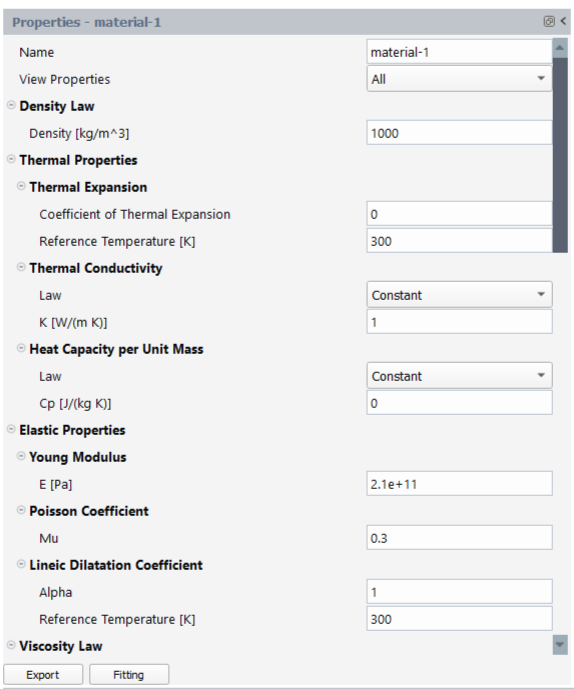



Here, most of the fields are informational, however, you can go to this page to change the **Geometry Type**, **Calculation Type**, **Task Name**, or any of the **Physics Options**.

## 28.8.2. Materials

Use the **Materials** node to access materials and their properties of the selected simulation (Figure 28.9: Material Properties (p. 271)).

**Figure 28.9: Material Properties**




Here, you can change the **Name** and elect to view and edit all (or a selection) of the material properties (such as viscoelastic and/or thermal properties) using the **View Properties** field.

For more information, see Material Properties in the Polyflow User's Guide.

## 28.8.3. Cell Zones

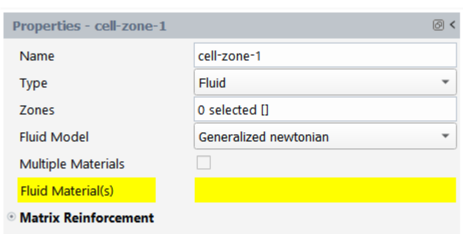
Use the **Cell Zones** node to access cell zone definitions for your simulation. Cell zones represent the domain of your simulation, identified by the following **Types**:



### 28.8.3.1. Fluid Cell Zones

When the **Type** is set to **Fluid**, use the **Cell Zones** node to access fluid cell zone definitions for your simulation (Figure 28.10: Fluid Cell Zone Properties (p. 272)).

**Figure 28.10: Fluid Cell Zone Properties**



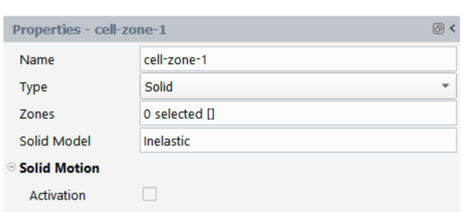

Here, you can change the **Name**, assign **Zones**, and change the **Fluid Model** (i.e., a viscoelastic model for the fluid zone: **Generalized Newtonian**, **Simplified viscoelastic**, **Differential viscoelastic**, or **Integral viscoelastic**).

For more information, see Generalized Newtonian Flow and Viscoelastic Flows in the Polyflow User's Guide.

### 28.8.3.2. Solid Cell Zones

When the **Type** is set to **Solid**, use the **Cell Zones** node to access solid cell zone definitions for your simulation (Figure 28.11: Solid Cell Zone Properties (p. 272)).

**Figure 28.11: Solid Cell Zone Properties**

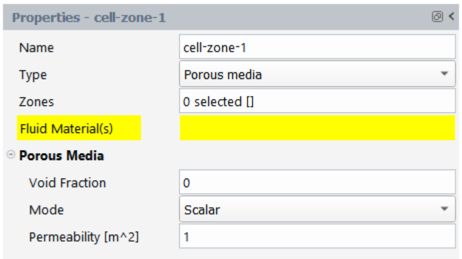



Here, you would typically set the **Name**, select the appropriate **Zones**, the **Solid Model**, and assorted **Solid Motion** properties, if applicable.

### 28.8.3.3. Porous Region Cell Zones

When the **Type** is set to **Porous media**, use the **Cell Zones** node to access porous cell zone definitions for your simulation (Figure 28.12: Porous Media Cell Zone Properties (p. 273)).

**Figure 28.12: Porous Media Cell Zone Properties**



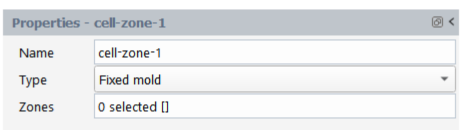

Here, you would typically set the **Name**, select the appropriate **Zones**, **Fluid Materials**, and assorted **Porous Media** properties..

For more information, see Porous Media in the Polyflow User's Guide.

### 28.8.3.4. Fixed Mold Cell Zones

When the **Type** is set to **Fixed mold**, use the **Cell Zones** node to access fixed mold cell zone definitions for your simulation (Figure 28.13: Fixed Mold Cell Zone Properties (p. 273)).

**Figure 28.13: Fixed Mold Cell Zone Properties**

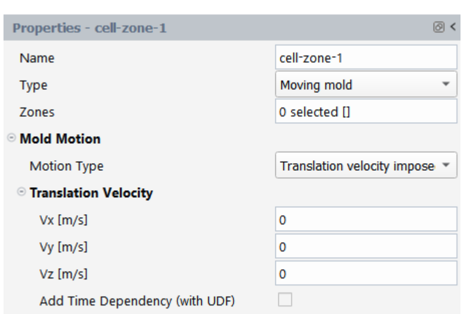



Here, you would typically set the **Name**, and select the appropriate **Zones** for your fixed mold.

### 28.8.3.5. Moving Mold Cell Zones

When the **Type** is set to **Moving mold**, use the **Cell Zones** node to access moving mold cell zone definitions for your simulation (Figure 28.14: Moving Mold Cell Zone Properties (p. 273)).

**Figure 28.14: Moving Mold Cell Zone Properties**

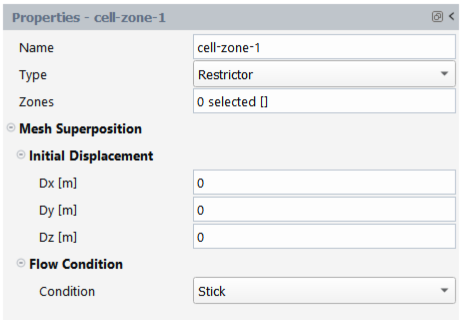


Here, you would typically set the **Name**, select the appropriate **Zones**, and assorted **Mold Motion** properties for the moving mold..

### 28.8.3.6. Restrictor Cell Zones

When the **Type** is set to **Restrictor**, use the **Cell Zones** node to access restrictor cell zone definitions for your simulation (Figure 28.15: Restrictor Cell Zone Properties (p. 274)).

**Figure 28.15: Restrictor Cell Zone Properties**



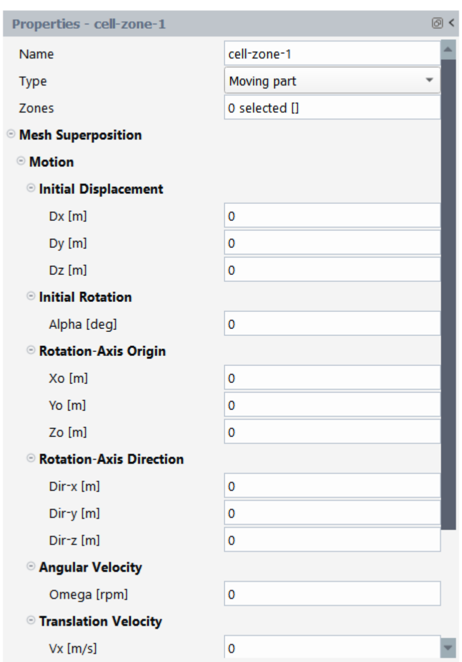

Here, you would typically set the **Name**, select the appropriate **Zones**, and assorted **Mesh Superposition** properties for the restrictor.

For more information, see Flows with Internal Moving Parts in the Polyflow User's Guide.

### 28.8.3.7. Moving Part Cell Zones

When the **Type** is set to **Moving part**, use the **Cell Zones** node to access moving part cell zone definitions for your simulation (Figure 28.16: Moving Part Cell Zone Properties (p. 274)).

**Figure 28.16: Moving Part Cell Zone Properties**



Here, you would typically set the **Name**, select the appropriate **Zones**, and assorted **Mesh Superposition** properties for the moving part.

For more information, see Flows with Internal Moving Parts in the Polyflow User's Guide.

## 28.8.4. Boundary Conditions

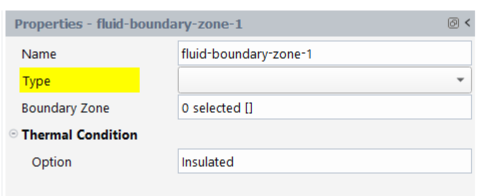
Depending on the type of simulation, boundary conditions can be of the following types:



### 28.8.4.1. Fluid Boundary Conditions

Specify conditions for the fluid material(s) at the various boundaries in your simulation (Figure 28.17: Fluid Boundary Properties (p. 275)).

**Figure 28.17: Fluid Boundary Properties**




#### 28.8.4.1.1. Fluid Boundary Zone Properties

Fluid boundary zone property settings for your simulation. Typically, you will change the following properties:

**Type**

Specify the type of boundary (inlet, outlet, wall, etc.)

**Boundary zone**

Specify one or more surfaces to assign to this boundary.

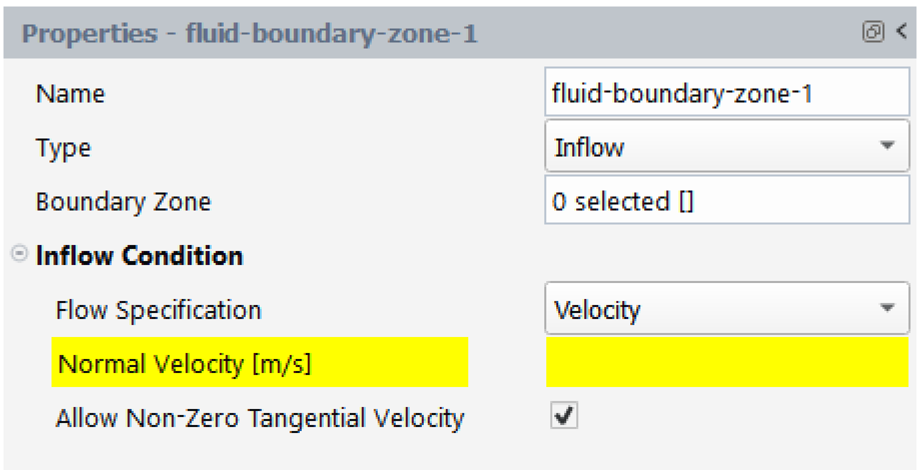
Fluid boundary types consist of the following:

#### 28.8.4.1.1.1. Inflow Fluid Boundary

Inlets define conditions where flow is expected to enter the solution domain.

When the **Type** is set to **Inflow**, use the **Fluid Boundary Zone** node to access inflow boundary definitions for your simulation (Figure 28.18: Inflow Fluid Boundary Properties (p. 276)).

**Figure 28.18: Inflow Fluid Boundary Properties**




Here, you would typically set the **Name**, select the appropriate **Boundary Zone**, and assorted **Inflow Condition** properties for the inflow fluid boundary.

##### 28.8.4.1.1.1.1. Inflow Properties

Properties related to the inflow fluid boundary condition.

**Flow specification**

Indicate the flow conditions at the inlet boundary. You can choose to impose a mass flow rate, a volume flow rate, or a velocity.

**Velocity profile**

You can choose to pre-compute the velocity profile corresponding to the given flow rate (fully developed), to compute it with the main flow (computed dynamically) or let the program decide automatically (program controlled).

**Incoming material**

Specify the material that flows through the inlet.

**Volume flow rate**

Provide the flow rate in terms of the volume of material that flows through the inlet.

**Mass flow rate**

Provide the flow rate in terms of the mass of material that flows through the inlet.

**CSV Filename**

Specify the name and location of the comma separated file that contains the inflow velocity profile.

**Field Name**

Specify the field variable from the CSV file that you want to apply to the inflow velocity profile.

**Normal velocity**

Provide the velocity of material that flows through the inlet.

**Allow non-zero tangential velocity**

Deselecting this option forces the velocity to be strictly normal to the inlet.

**Add Time/Continuation Dependency (with UDF)**

help text for add time/continuation dependency (with udf)

**UDF Id**

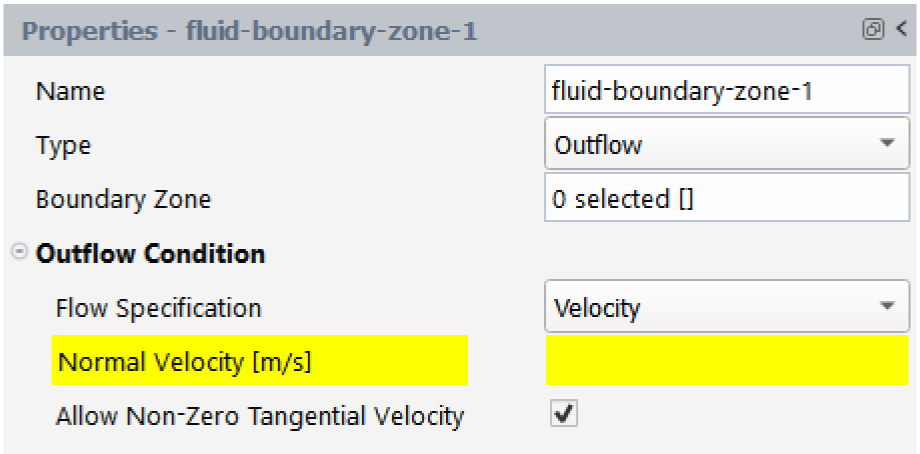
help text for udf id

##### 28.8.4.1.1.2. Outflow Fluid Boundary

For polymer extrusion simulations, when the outlet represents an actual exit from which the fluid can freely flow, tangential velocities at the outlet should be allowed (such as a die exit). However, when the outlet is at the end of the computation domain, but not at the end of the channel in which the fluid flows, tangential velocities at outlets should not be allowed because the fluid is still confined after the outlet.

When the **Type** is set to **Outflow**, use the **Fluid Boundary Zone** node to access outflow boundary definitions for your simulation (Figure 28.19: Outflow Fluid Boundary Properties (p. 278)).

**Figure 28.19: Outflow Fluid Boundary Properties**




Here, you would typically set the **Name**, select the appropriate **Boundary Zone**, and assorted **Outflow Condition** properties for the outflow fluid boundary.

### 28.8.4.1.1.2.1. Outflow Properties

Properties related to the outflow fluid boundary condition.

**Flow specification**

Indicate the flow conditions at the outlet boundary. You can choose to impose a mass flow rate, a volume flow rate, or a pressure (default).

**Volume flow rate**

Provide the flow rate in terms of the volume of material that flows through the outlet.

**Mass flow rate**

Provide the flow rate in terms of the mass of material that flows through the outlet.

**Gauge static pressure**

Provide the pressure that you want to impose at the outlet. That pressure will determine the actual flow rate if pressure is imposed at inlet(s) or the level of pressure in the die if flow rate is imposed at inlet(s).

**CSV Filename**

Specify the name and location of the comma separated file that contains the outflow velocity profile.

**Field Name**

Specify the field variable from the CSV file that you want to apply to the outflow velocity profile.

**Allow non-zero tangential velocity**

Deselecting this option forces the velocity to be strictly normal to the outlet.

**Add Time/Continuation Dependency (with UDF)**

help text for add time/continuation dependency (with udf)

**UDF Id**

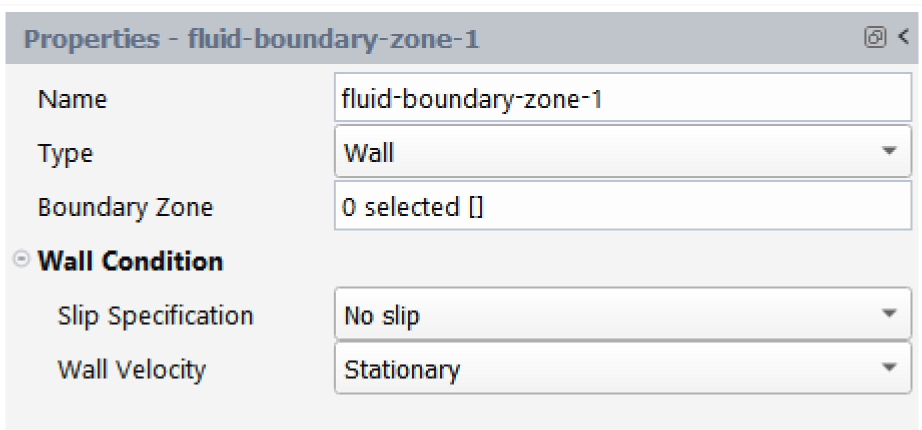
help text for udf id

### 28.8.4.1.1.3. Wall Fluid Boundary

Wall boundaries assigned to surfaces of a fluid region prevent flow through those surfaces. The conditions that best describe the forces applied to a wall are summarized below.

When the **Type** is set to **Wall**, use the **Fluid Boundary Zone** node to access wall boundary definitions for your simulation (Figure 28.20: Wall Fluid Boundary Properties (p. 279)).

**Figure 28.20: Wall Fluid Boundary Properties**




Here, you would typically set the **Name**, select the appropriate **Boundary Zone**, and assorted **Wall Condition** properties for the wall fluid boundary.

**Slip Specification** includes the following options:

- **No slip**: the fluid sticks to the wall. If the wall is moving, the fluid moves with the same velocity as the wall.
- **Partial slip**: the fluid partially sticks to the wall.
- **Free slip**: a frictionless wall.

#### 28.8.4.1.1.3.1. Wall Properties

Properties related to the wall fluid boundary condition.

**Slip specification**

Indicate the flow conditions at the wall boundary. You can choose whether to impose a zero velocity (no slip), a friction force (partial slip), or no friction (free slip).

**Wall velocity**

Indicate whether the wall boundary is a stationary wall, or a moving wall with an associated velocity.

**Slip model**

For slip conditions, specify how the shear force is calculated with respect to the tangential relative velocity.

**Friction coefficient**

Specify the friction coefficient for slip conditions.

**Friction Coefficient**

help text for friction coefficient

**Scaling Factor**

help text for scaling factor

**Exponent**

help text for exponent

**First friction coefficient**

Specify the first friction coefficient for slip conditions.

**Second friction coefficient**

Specify the second friction coefficient for slip conditions.

**Friction coefficient**

Specify the friction coefficient for slip conditions.

**Scaling factor**

A scaling factor with the dimensions of the velocity. It affects the slope of the slip-velocity curve.

**Critical stress**

The critical force density at which the friction coefficient changes. When this stress is exceeded, the second friction component is used.

**First Friction Coefficient**

help text for first friction coefficient

**First Scaling Factor**

help text for first scaling factor

**Second Friction Coefficient**

help text for second friction coefficient

**Second Scaling Factor**

help text for second scaling factor

**Exponent**

help text for exponent

**Temperature dependence**

Specify whether the thermal dependency is a constant, or it obeys a first-order approximation of the Arrhenius law, or it obeys the Arrhenius law.

**Activation energy ratio**

The ratio of the activation energy to the thermodynamic constant.

**Reference temperature**

A reference temperature for which the thermal dependency function H(T) is 1.

**Pressure dependence**

Specify the wall pressure dependency as either exponential or linear.

**Alpha**

Specify a value for the linear pressure dependency.

**Beta**

Specify a value for the exponential pressure dependency.

###### 28.8.4.1.1.3.1.1. First Point of Axis

Define the axis of rotation by specifying the coordinates for the first of two points.

**X1**

Specify the X component of the rotation axis for the first point.

**Y1**

Specify the Y component of the rotation axis for the first point.

**Z1**

Specify the Z component of the rotation axis for the first point.

#### 28.8.4.1.1.3.1.2. Second Point of Axis

Define the axis of rotation by specifying the coordinates for the second of two points.

**X1**

Specify the X component of the rotation axis for the first point.

**Y1**

Specify the Y component of the rotation axis for the first point.

**Z1**

Specify the Z component of the rotation axis for the first point.

**X2**

Specify the X component of the rotation axis for the second point.

**Y2**

Specify the Y component of the rotation axis for the second point.

**Z2**

Specify the Z component of the rotation axis for the second point.

#### 28.8.4.1.1.3.1.3. Angular Velocity

Represents the angular velocity for the rotating wall.

**Omega [rad/s]**

Specify the magnitude of the angular velocity at which the wall is rotating about the axis defined by the two points Pt1 and Pt2.

#### 28.8.4.1.1.3.1.4. Translation Velocity

Represents the translational velocity for the wall boundary.

**Vx**

Specify the X component of the translational velocity for the wall boundary.

**Vy**

Specify the Y component of the translational velocity for the wall boundary.

**Vz**

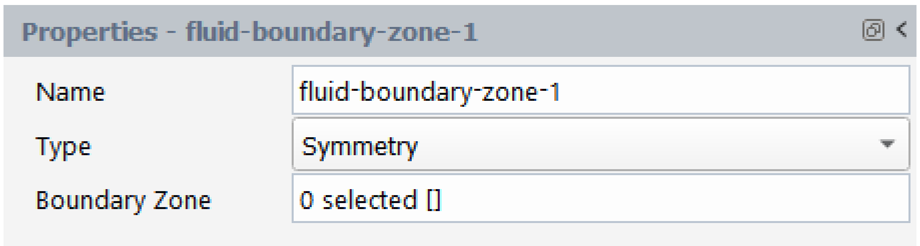
Specify the Z component of the translational velocity for the wall boundary.

#### 28.8.4.1.1.4. Symmetry Fluid Boundary

A symmetry boundary imposes constraints that mirrors the expected pattern of the flow or thermal solution on either side of it. A symmetry boundary is equivalent to imposing zero values for the normal velocity and tangential force.

When the **Type** is set to **Symmetry**, use the **Fluid Boundary Zone** node to access symmetry boundary definitions for your simulation (Figure 28.21: Symmetry Fluid Boundary Properties (p. 283)).

**Figure 28.21: Symmetry Fluid Boundary Properties**




Here, you would typically set the **Name**, and select the appropriate **Boundary Zone** for the symmetry fluid boundary.

##### 28.8.4.1.1.4.1. Symmetry Properties

Properties related to the symmetry fluid boundary condition.

**Plane of symmetry**

Specify the type of symmetry plane as either normal in the X, Y, or Z direction, or in an arbitrary normal direction.

**Nx**

Specify the X value for the arbitrary normal direction of the symmetry plane.

**Ny**

Specify the Y value for the arbitrary normal direction of the symmetry plane.

**Nz**

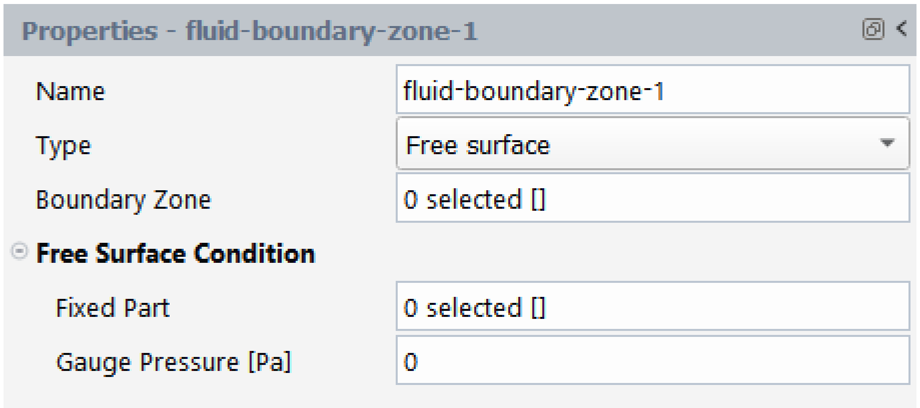
Specify the Z value for the arbitrary normal direction of the symmetry plane.

#### 28.8.4.1.1.5. Free Surface Fluid Boundary

In an extrusion problem, where the shape of the extrudate is not known in advance, a free surface is used to represent the outer surface of the extrudate. A free-surface problem involves a boundary whose position is computed as part of the solution, since it is not known in advance.

When the **Type** is set to **Free surface**, use the **Fluid Boundary Zone** node to access free surface boundary definitions for your simulation (Figure 28.22: Free Surface Fluid Boundary Properties (p. 284)).

**Figure 28.22: Free Surface Fluid Boundary Properties**




Here, you would typically set the **Name**, select the appropriate **Boundary Zone**, and assorted **Free Surface Condition** properties for the free surface fluid boundary.

#### 28.8.4.1.1.5.1. Free Surface Properties

Properties related to the free surface fluid boundary condition.

**Fixed part**

Specify one or more surfaces to which the free surface is attached/fixed.

**Gauge Pressure**

Specify the pressure applied on the free surface.

**Direction of Displacement**

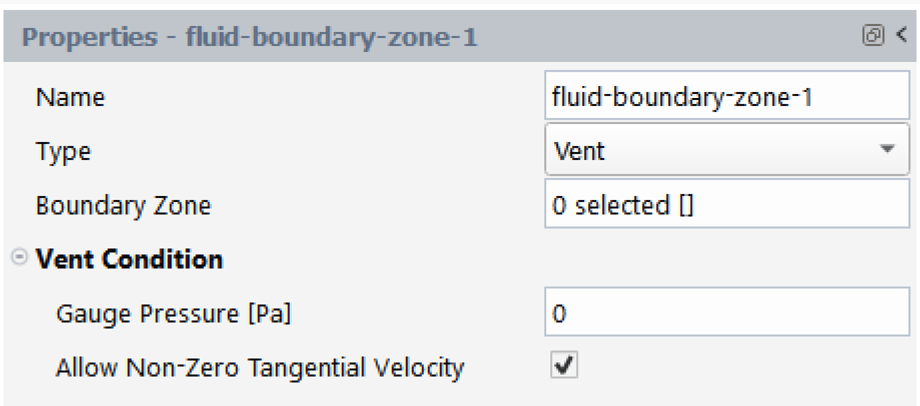
help text for direction of displacement

### 28.8.4.1.1.6. Vent Fluid Boundary

Vent boundary conditions are used to model inlet/outlet vents with a specified static pressure, and potential tangential velocity.

When the **Type** is set to **Vent**, use the **Fluid Boundary Zone** node to access vent boundary definitions for your simulation (Figure 28.23: Vent Fluid Boundary Properties (p. 285)).

**Figure 28.23: Vent Fluid Boundary Properties**




Here, you would typically set the **Name**, select the appropriate **Boundary Zone**, and assorted **Vent Condition** properties for the vent fluid boundary.

###### 28.8.4.1.1.6.1. Vent Properties

Properties related to the vent fluid boundary condition.

**Gauge static pressure**

Specify the pressure applied on the vent.

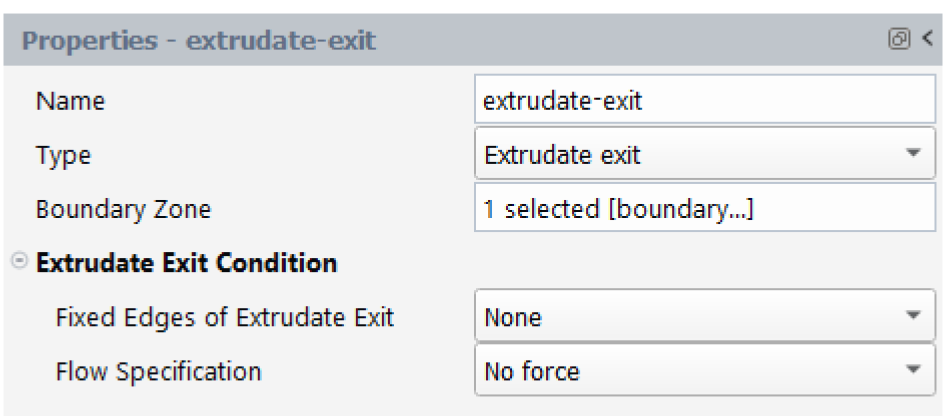
**Allow non-zero tangential velocity**

Deselecting this option forces the velocity to be strictly normal to the vent.

###### 28.8.4.1.1.7. Extrudate Exit Fluid Boundary

For polymer extrusion simulations where you want to predict the shape of the extrudate, or determine the die lip shape, you can also specify the conditions of the extrudate at the exit boundary (Figure 28.23: Vent Fluid Boundary Properties (p. 285)).

**Figure 28.24: Extrudate Exit Fluid Boundary Properties**




Here, you would typically set the **Name**, select the appropriate **Boundary Zone**, and assorted **Extrudate Exit** properties for the extrudate exit fluid boundary.

#### 28.8.4.1.1.7.1. Extrudate Exit Properties

Properties related to the extrudate exit fluid boundary condition.

**Fixed edges of extrudate exit**

Specify whether to fix all edges of the free-jet exit in the frame of a die lip shape design, to fix none of the edges, or to specify your own surfaces.

**Free surfaces with fixed edges**

Specify the applicable surfaces.

**Flow specification**

Specify the flow conditions at the free-jet exit. You can choose to impose a take-up force, a take-up velocity or a take-up force per unit area when the extrudate is pulled by the extrusion line. Take-up force(velocity) if the force(velocity) applied at the end of the free-jet to reach an extrusion speed. It can been seen as a model simulating the rolls or transportation belt of the extrusion line.

**Take-up force per unit area**

Specify a value for the take-up force per unit area.

#### 28.8.4.1.1.7.1.1. Take Up Velocity

The take-up velocity is the velocity applied at the end of the extrudate to reach an extrusion speed. It can been seen as a model simulating the rolls or transportation belt of the extrusion line.

**Vx**

Specify the X component of the take-up velocity.

**Vy**

Specify the Y component of the take-up velocity.

**Vz**

Specify the Z component of the take-up velocity.

#### 28.8.4.1.1.7.1.2. Take Up Force

The take-up force is the force applied at the end of the extrudate to reach an extrusion speed. It can been seen as a model simulating the rolls or transportation belt of the extrusion line.

**Fx**

Specify the X component of the take-up force.

**Fy**

Specify the Y component of the take-up force.

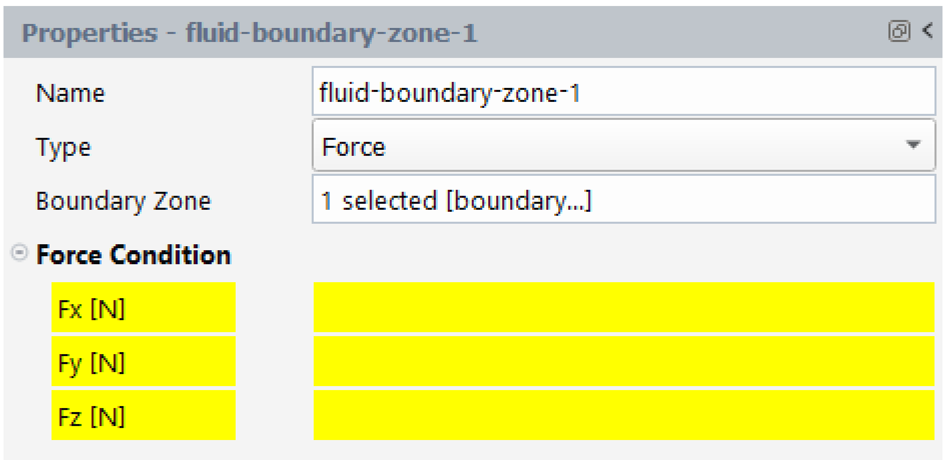
**Fz**

Specify the Z component of the take-up force.

#### 28.8.4.1.1.8. Force Fluid Boundary

A force boundary condition is used in cases when the conditions at a boundary can be represented by a force (Figure 28.25: Force Fluid Boundary Properties (p. 287)).

**Figure 28.25: Force Fluid Boundary Properties**




Here, you would typically set the **Name**, select the appropriate **Boundary Zone**, and assorted **Force Condition** properties for the force fluid boundary.

##### 28.8.4.1.1.8.1. Force Properties

Properties related to the force fluid boundary condition.

**Fx**

Specify the X component of the force.

**Fy**

Specify the Y component of the force.

**Fz**

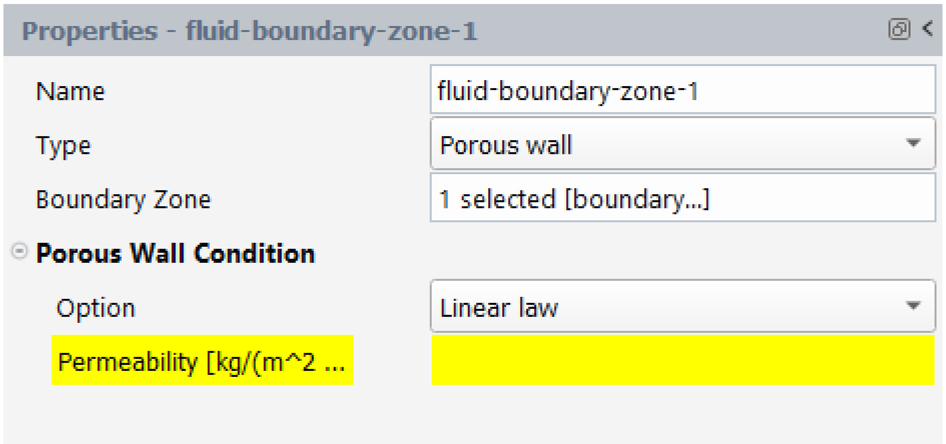
Specify the Z component of the force.

#### 28.8.4.1.1.9. Porous Wall Fluid Boundary

Porous wall conditions are used to model a thin "membrane" that has a known normal force on it. Examples of uses for the porous wall condition include screens and filters.

For the porous wall condition (Figure 28.26: Porous Wall Fluid Boundary Properties (p. 288)), a zero tangential velocity component is imposed simultaneously with one of three relationships between the normal force and the normal velocity. See Porous Wall Condition for details.

**Figure 28.26: Porous Wall Fluid Boundary Properties**




Here, you would typically set the **Name**, select the appropriate **Boundary Zone**, and assorted **Porous Wall Condition** properties for the porous wall fluid boundary.

Porous wall **Options** include:

- **Linear**: represents a linear relationship for the permeability coefficient of the porous wall.
- **Threshold**: represents a porous definition based on two different permeabilities and a critical stress threshold value.
- **Asymptotic** : represents a relationship between the permeability and a scaling factor.

### 28.8.4.1.1.9.1. Porous Wall Properties

Properties related to the porous wall fluid boundary condition.

**Option**

Indicate how you would like to calculate the normal force on the porous boundary: using the linear law, the threshold law, or the asymptotic law.

**Permeability**

Specify the permeability for the linear law.

**First permeability**

Specify a value for the first permeability for the threshold law.

**Second permeability**

Specify a value for the second permeability for the threshold law.

**Critical stress**

Specify a value for the critical force density for the threshold law.

**Permeability**

Specify a value for the permeability using the asymptotic law.

**Scaling factor**

Specify a value for the scaling factor.

**Temperature dependence**

Choose how the temperature dependency is managed. You can choose **one**, or either of the Arrhenius laws.

**Activation energy ratio**

The ratio of the activation energy to the thermodynamic constant.

**Reference temperature**

A reference temperature for which the thermal dependency function H(T) is 1.

##### 28.8.4.1.1.10. Thermal Condition

Thermal considerations for your simulation. When temperature effects are included in your simulation, each of the fluid boundary conditions will have **Thermal Condition** properties.

**Option**

Specify the thermal conditions at this wall boundary. You can choose an insulated wall, or impose a heat flux (constant and/or convective), or impose the temperature of the wall, and provide the parameters of the selected energy boundary condition.

**Temperature**

Specify the temperature at the boundary.

**Heat flux**

Specify the heat flux that is independent of temperature. This heat flux is constant on the whole boundary.

**Heat transfer coefficient**

Specify the heat convection coefficient. The convection heat flux will be evaluated at a given location of the boundary by multiplying the difference between the calculated local temperature and the reference temperature by the convection coefficient.

**Convection temperature**

Specify the heat convection temperature at the boundary.

**Minimum Normal Velocity**

help text for minimum normal velocity

**Penalty Coefficient**

help text for penalty coefficient

**CSV Filename**

help text for csv filename

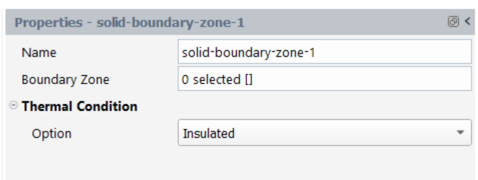
**Field Name**

help text for field name

#### 28.8.4.2. Solid Boundary Conditions

Specify conditions for the solid material(s) at the various boundaries in your simulation (Figure 28.27: Solid Boundary Properties (p. 290)).

**Figure 28.27: Solid Boundary Properties**

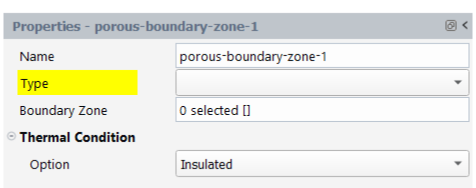



Here, you would typically set the **Name**, select the appropriate **Boundary Zone**, and, if applicable, any **Thermal Condition** properties for the solid boundary.

#### 28.8.4.3. Porous Media Boundary Conditions

Specify conditions for the porous media at the various boundaries in your simulation (Figure 28.28: Porous Media Boundary Properties (p. 290)).

**Figure 28.28: Porous Media Boundary Properties**



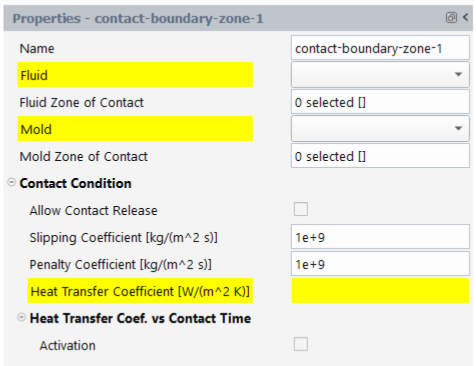

Here, you would typically set the **Name**, select the appropriate **Boundary Zone**, the **Type**, and, if applicable, any **Thermal Condition** properties for the porous media boundary.

The **Type** can include: **Pressure**, **Normal velocity**, **Wall**, and **Symmetry**.

#### 28.8.4.4. Contact Boundary Conditions

Use the properties of the **Contact Boundary Condition** (Figure 28.29: Contact Boundary Condition Properties (p. 291)) to describe how and where the fluid and mold come into contact with each other in your simulation.

**Figure 28.29: Contact Boundary Condition Properties**

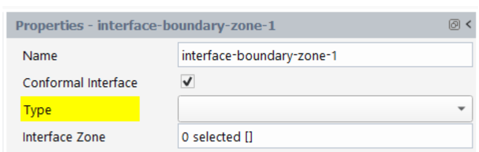



Here, you would typically set the **Name**, select the appropriate **Fluid** and its **Fluid Zone of Contact**, the **Mold**, and its **Mold Zone of Contact**, and any **Contact Condition** (and, if applicable, any **Thermal Condition**) properties for the contact boundary.

## 28.8.4.5. Interface Boundary Conditions

Use the properties of the **Interface Boundary Condition** object (Figure 28.30: Interface Boundary Condition Properties (p. 291)) to describe the interaction between fluids, solids, or porous regions in your simulation.

**Figure 28.30: Interface Boundary Condition Properties**

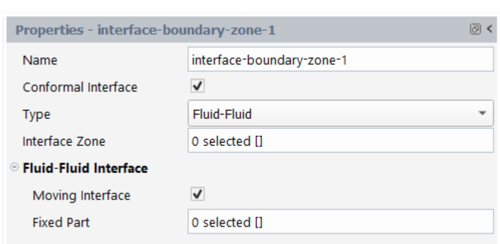



Here, you would typically set the **Name**, the **Type**, and select the appropriate **Interface Zone** for the interface boundary.

## 28.8.4.6. Fluid-Fluid Interface Boundary Conditions

Use the properties of the **Fluid-Fluid Interface Boundary Condition** object (Figure 28.31: Fluid-Fluid Interface Boundary Condition Properties (p. 291)) to describe the interaction between fluids in your simulation.

**Figure 28.31: Fluid-Fluid Interface Boundary Condition Properties**

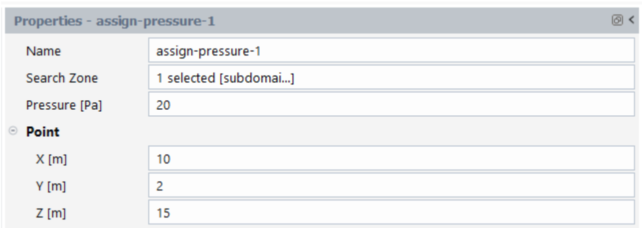



Here, you would typically set the **Name**, the **Type**, and select the appropriate **Interface Zone** for the interface boundary, along with **Fluid-Fluid Interface** conditions.

### 28.8.5. Pressure Assignment

Use the properties of the **Assign Pressures** object (Figure 28.32: Assign Pressure Properties (p. 292)) to describe a fixed pressure to apply to your enclosed flow domain.

**Figure 28.32: Assign Pressure Properties**

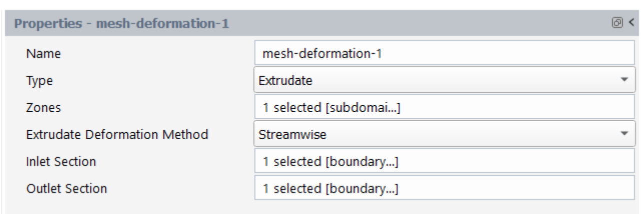



Here, you would typically set the **Name**, the specific **Search Zone** or region, the **Pressure** value for the assignment boundary, along with distinct **Point** values where the pressure will be assigned.

### 28.8.6. Mesh Deformations

Use the properties of the **Mesh Deformation** object (Figure 28.33: Mesh Deformation Properties (p. 292)) to describe how and where the extrusion deforms during your simulation.

**Figure 28.33: Mesh Deformation Properties**




Here, you would typically set the **Name**, the **Type** of mesh deformation, select the **Zones**, specify the **Extrudate Deformation Method**, as well as make assignments for the **Inlet Section** and the **Outlet Section** for the mesh deformation boundary.

## 28.9. Setting Solution Options

The **Solution** section of the Outline View and the Ribbon allow you access to common solution-specific controls, such as:

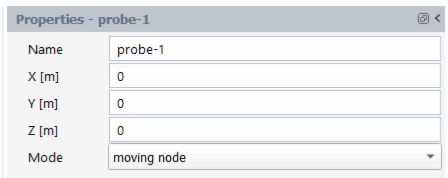
- 28.9.1. Creating Solution Probes
- 28.9.2. Generating Derived Quantities From Your Solution
- 28.9.3. Accessing Solution Methods
- 28.9.4. Accessing Calculation Activities
- 28.9.5. Accessing Solution Outputs
- 28.9.6. Accessing Solution Monitors
- 28.9.7. Running the Calculation

### 28.9.1. Creating Solution Probes

Use the **Probes** task to create discretely located points in your domain to monitor your simulation.

Use the **New...** button to create a new point object (Figure 28.34: Properties of Probe (p. 293)). Once you specify the x, y, and Z coordinates, you can indicate whether the point will a **Moving node**, or a **Fixed geometric location**.

**Figure 28.34: Properties of Probe**

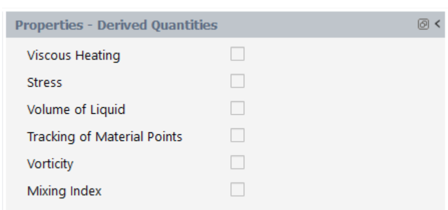



Here, you would typically set the **Name**, and the **X**, **Y**, and **Z** coordinates of the solution probe.

### 28.9.2. Generating Derived Quantities From Your Solution

Use the **Derived Quantities** task to determine which of the available derived quantities you want to create an output field for within your simulation. For more information, see Computing Derived Quantities

**Figure 28.35: Properties of Derived Quantities**




Depending on the type of simulation (blow molding extrusion, etc.) the choices for **Derived Quantities** can include:

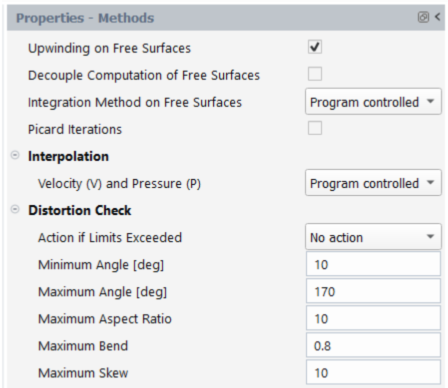
- **Viscous Heating**
- **Stress**
- **Volume of Liquid**
- **Tracking of Material Points**
- **Vorticity**
- **Mixing Index**

Once selected, when a solution is generated, these quantities will be available for postprocessing (see Postprocessing Results (p. 296)).

## 28.9.3. Accessing Solution Methods

Use the **Methods** task to determine specific solution techniques that can be applied to your simulation.

**Figure 28.36: Properties of Methods**

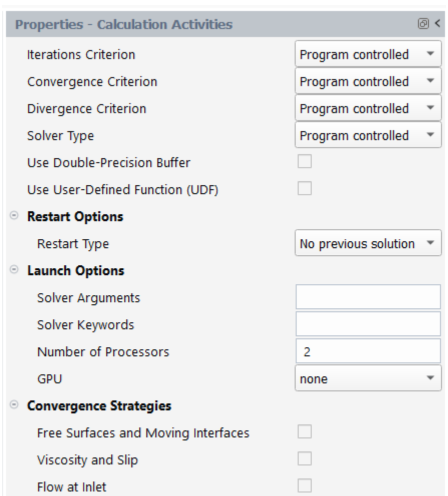



In most cases, the default settings are adequate.

## 28.9.4. Accessing Calculation Activities

Use the **Calculation Activities** task to determine what, if any, activities will be performed during the calculation of the solution.

**Figure 28.37: Properties of Calculation Activities**

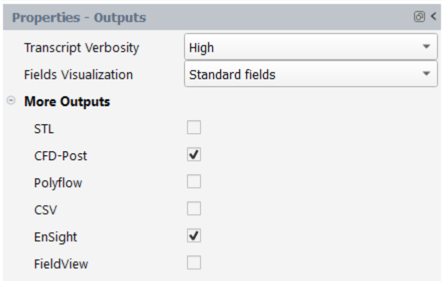



In most cases, the default settings are adequate.

## 28.9.5. Accessing Solution Outputs

Use the **Outputs** task properties (Figure 28.38: Properties of Outputs (p. 295)) to configure any output-specific settings that will accompany the solution.

**Figure 28.38: Properties of Outputs**




In most cases, the default settings are adequate.

## 28.9.6. Accessing Solution Monitors

Use the **Monitors** task properties (Figure 28.39: Properties of Monitors (p. 295)) to configure any monitors that will describe the solution's convergence.

**Figure 28.39: Properties of Monitors**

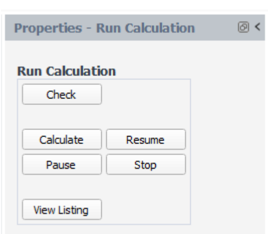

Depending on the type of simulation (blow molding, extrusion, etc.) some convergence criteria are already enabled by default.

In most cases, the default settings are adequate.

## 28.9.7. Running the Calculation

Once you have set up the simulation, use the **Run Calculation** task in the **Outline View** to access the Figure 28.40: Properties of Run Calculation (p. 295).

**Figure 28.40: Properties of Run Calculation**




This task includes options to:

- **Check** the setup of your simulation.

  Click the **Check** button to check the setup of your simulation. Any relevant messages are displayed in the console window, and you are notified if and when the solver can be started.

- **Calculate** a solution.

Once you have checked your simulation setup, click the **Calculate** button to start the solver.

- **Interrupt** the calculation.

  Once you have started your calculation, click the **Interrupt** button to pause the solver.

- **Stop** the calculation.

  Once you have started your calculation, click the **Stop** button to halt the solver.

- **View Listing** of your calculations.

  Once your calculation is complete, click the **View Listing** button to see the solution transcript, available as a docked tab next to the graphics window.

## 28.10. Postprocessing Results

You can review your results in the graphics window using objects such as contours, vectors, pathlines, particle tracks and XY plots. You can also create surfaces for further exploring the results.

These different postprocessing tools are generally distributed into three groups: surfaces, reports, graphics objects, and views, and they are discussed in the following sections:



### 28.10.1. Surfaces

You can create different types of surfaces to visualize your results, including points, lines, rakes, planes, and iso-surfaces.

Refer to the following sections for more details:



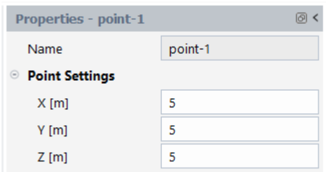
#### 28.10.1.1. Point Surfaces

You may be interested in displaying results at a single point in the domain. For example, you may want to monitor the value of some variable or function at a particular location. To do this, you must first create a "point" surface, which consists of a single point. When you display node-value data on a point surface, the value displayed is a linear average of the neighboring node values. If you display cell-value data, the value at the cell in which the point lies is displayed.

**Results → Surfaces New → Point**

To create a point surface, use the Figure 28.41: Properties of a Point Surface (p. 297).

**Figure 28.41: Properties of a Point Surface**




Here, you would typically set the **Name**, and the **X**, **Y**, and **Z** coordinates of the point.

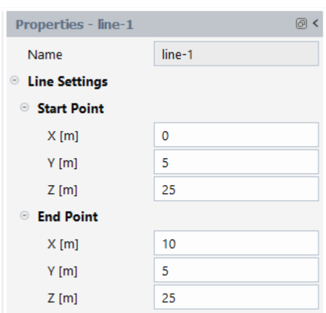
### 28.10.1.2. Line Surfaces

A line is simply a line that extends up to and includes the specified endpoints; data points will be located where the line intersects the faces of the cell, and consequently may not be equally spaced.

**Results → Surfaces New → Line**

To create a line surface, use the Figure 28.42: Properties of a Line Surface (p. 297).

**Figure 28.42: Properties of a Line Surface**




Here, you would typically set the **Name**, and various **Line Settings**, such as the **X**, **Y**, and **Z** coordinates of the **Start Point** and the **End Point** of the line surface.

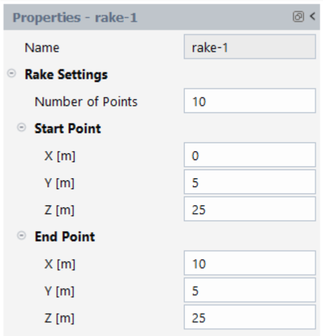
### 28.10.1.3. Rake Surfaces

A rake consists of a specified number of points equally spaced between two specified endpoints.

**Results → Surfaces New → Rake**

To create a rake surface, use the Figure 28.43: Properties of a Rake Surface (p. 298).

**Figure 28.43: Properties of a Rake Surface**




Here, you would typically set the **Name**, and various **Rake Settings**, such as the **X**, **Y**, and **Z** coordinates of the **Start Point** and the **End Point** of the rake surface.

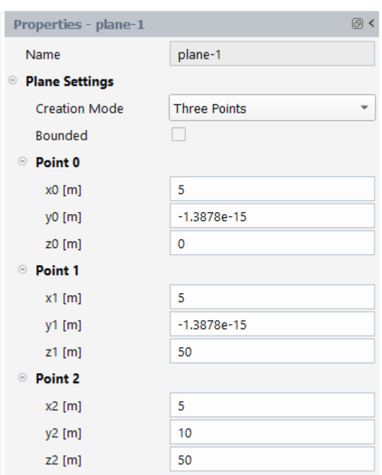
### 28.10.1.4. Plane Surfaces

To display flow-field data on a specific plane in the domain, you will use a plane surface. You can create surfaces that cut through the solution domain along arbitrary planes.

**Results → Surfaces Plane...**

To create a plane surface, use the Figure 28.44: Properties of a Plane Surface (p. 298).

**Figure 28.44: Properties of a Plane Surface**




Here, you would typically set the **Name**, and various **Plane Settings** of the plane surface.

There are three types of plane surfaces that you can create (via the **Creation Mode**):

- Coordinate system-based—the plane is created in the YX, ZX, or XY directions, bounded by the extents of the domain. You can move the plane to the desired location in the domain using the plane tool. For example, if you are using the **YZ Plane** method, you can drag the plane in the (+) or (-) X direction.

- Point and normal—the plane orientation is determined by selecting a point and specifying a direction normal to that point. The extents of the plane are the edges of the domain. You have the option to control the orientation of the plane using the plane tool or you can compute the normal from a surface.
- Three points—the plane orientation and extents are bounded by three points that you can select. You also have the option to manipulate the points and orientation of the plane directly in the graphics window using the plane tool.

#### 28.10.1.5. Iso-Surfaces

If you want to If you want to display results on cells that have a constant value for a specified variable, you will need to create an iso-surface of that variable. Generating an iso-surface based on *x-* , *y-* , or*z-* coordinate, for example, will give you an *x* , *y* , or *z* cross-section of your domain; generating an iso-surface based on pressure will enable you to display data for another variable on a surface of constant pressure. You can create an iso-surface from an existing surface or from the entire domain. Furthermore, you can restrict any iso-surface to a specified cell zone.

---

**Important:**

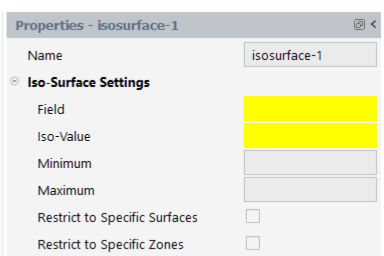
Note that you cannot create an iso-surface until you have initialized the solution, performed calculations, or read a data file.

---

**Results** → **Surfaces** **New** → **Iso-Surface**

To create an iso-surface, use the Figure 28.45: Properties of an Iso-Surface (p. 299).

**Figure 28.45: Properties of an Iso-Surface**




Here, you would typically set the **Name**, and various **Iso-Surface Settings** of the iso-surface.

Click **Display** at the bottom of the iso-surface properties to display the iso-surface in the graphics window.

Click **Delete** at the bottom of the iso-surface properties to delete an iso-surface.

Instead of creating plane surfaces one at a time as described above, you have the option to create multiple plane surfaces at once using the Figure 28.47: Create Multiple Planes Dialog Box (p. 300).

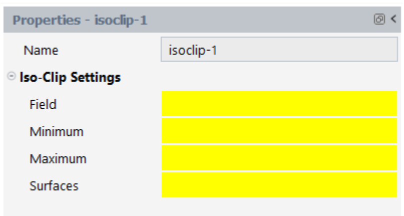
### 28.10.1.6. Iso-Clip Surfaces

Iso-clip surfaces are clipped surface that consist of those points on the selected surface where the scalar field values are within the specified range.

**Results** → **Surfaces** **New** → **Iso-Clip**

To create an iso-clip surface, use the Figure 28.46: Properties of an Iso-Clip Surface (p. 300).

**Figure 28.46: Properties of an Iso-Clip Surface**




Here, you would typically set the **Name**, and various **Iso-Clip Settings** of the iso-clip surface.

Click **Display** at the bottom of the iso-clip properties to display the iso-clip in the graphics window.

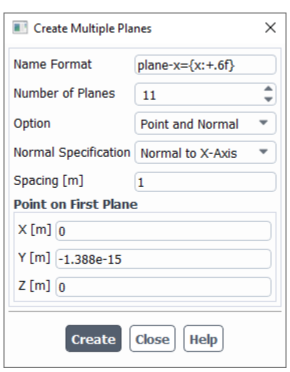
Click **Delete** at the bottom of the iso-clip properties to delete an iso-clip.

### 28.10.1.7. Creating Multiple Planes

Instead of creating plane surfaces one at a time as described above, you have the option to create multiple plane surfaces at once using the Figure 28.47: Create Multiple Planes Dialog Box (p. 300).

**Results** → **Surfaces** **Create Multiple Planes...**

**Figure 28.47: Create Multiple Planes Dialog Box**




To use the **Create Multiple Planes** dialog box:

1. (Optional) Provide a format for naming the plane surfaces in the **Name Format** field.

2. Specify how many planes will be created in the **Number of Planes** field.

3. Select the method for how you want to create the planes in the **Option** drop-down list.

- **Point and Normal**—similarly to the process described earlier in this section, you must provide a point and the direction normal to that point to define the first plane.

- **First and Last Point**—you define the coordinates for the first and last point, which determines the orientation of the planes. The spacing is determined by how many planes you specify in the **Number of Planes** field.

4. (**Point and Normal** only) Specify the direction normal (perpendicular) to the plane in the **Normal Specification** drop-down list.

5. (**Point and Normal** only) Specify how far apart the planes are from each other in the **Spacing** field.

6. Provide the coordinate for the location of the first plane in the **Point on First Plane** group box.

7. (**First and Last Point** only) Provide the coordinates for the location of the last plane in the **Point on Last Plane** group box.

8. Click **Create** to create the new plane surfaces.

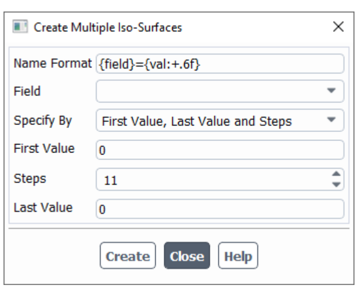
The new plane surfaces created using the **Create Multiple Planes** dialog box are added to the Outline View tree under the **Surfaces** branch and are now eligible for editing individually. *Once created, the multiple planes cannot be edited as a group.*

#### 28.10.1.8. Creating Multiple Iso-Surfaces

Instead of iso-surfaces one at a time as described above, you have the option to create multiple iso-surfaces at once using the Figure 28.48: Create Multiple Iso-Surfaces Dialog Box (p. 301).

**Results** → **Surfaces** **Create Multiple Iso-surfaces...**

**Figure 28.48: Create Multiple Iso-Surfaces Dialog Box**




To use the **Create Multiple Iso-Surfaces** dialog box:

1. (Optional) Provide a format for naming the iso-surfaces in the **Name Format** field.

2. Select the field that you want to use for creating the iso-surfaces from the **Field** drop-down list:

   - **local shear rate**—value of the local shear rate.

   - **velocity-magnitude**—magnitude of the velocity.

   - **x-coordinate**—value of the x-coordinate.

- **y-coordinate**—value of the y-coordinate.
- **z-coordinate**—value of the z-coordinate.
- **x-velocity**—x-component of the velocity.
- **y-velocity**—y-component of the velocity.
- **z-velocity**—z-component of the velocity.
- **viscosity**—value of the viscosity.
- **pressure**—value of the pressure.
- **density**—value of the density.

3. Specify the method you want to use for creating iso-surfaces in the **Specify By** drop-down list.

   - **First Value, Last Value and Steps**—using this method you must specify the **First Value** for the quantity that you selected in the **Field** drop-down list, specify how many iso-surfaces you want created by entering the number of **Steps**, and provide the final value for the selected quantity in the **Last Value** field.
   - **First Value, Last Value and Increment**—using this method you must specify the **First Value** for the quantity that you selected in the **Field** drop-down list, specify the size of the increments between the first and last value, which determines the number of iso-surfaces to be created, and provide the final value for the selected quantity in the **Last Value** field.
   - **First Value, Increment and Steps**—using this method you must specify the **First Value** for the quantity that you selected in the **Field** drop-down list, specify the size of the increments from the first value, and provide the total number of steps in the **Steps** field, which determines the total number of iso-surfaces to be created.
   - **Last Value, Decrement and Steps**—using this method you must specify the size of the decrement (negative increment) in the **Decrement** field, which will go backwards from the **Last Value**, provide the total number of steps in the **Steps** field, which determines the total number of iso-surfaces, and provide the final value for the selected quantity in the **Last Value** field.

4. Click **Create** to create the new iso-surfaces.

The new iso-surfaces created using the **Create Multiple Iso-Surfaces** dialog box are added to the Outline View tree under the **Surfaces** branch and are now eligible for editing individually. *Once created, the multiple iso-surfaces cannot be edited as a group.*

## 28.10.2. Reports

Static reports are available for computation of various postprocessing quantities.

**Results** → **Reports** → **New...**

Once a report is created, click **Print Report** and the result is printed in the **Console**. Click **Save Report** and the result is saved to a file and location of your choice. Click **Plot Report** and the result is printed in the **Graphics** window.

## 28.10.3. Graphics Objects

Postprocessing graphics objects are available for visualizing the results of your simulation. You can display the mesh, contours, vectors, pathlines, and scenes. Scenes allow you to combine multiple graphics objects within a single graphics window.

> **Note:**
>
> Only field variables that are appropriate and compatible for the specific workspace simulations are available in the **Field** dialog box.

Once you have created your graphics objects and/or made changes to their properties, you can:

- Click **Display** to visualize the object in the graphics window.
- Click **Save Image** to open the **Save Image** dialog box and proceed to save an image file of the object in the graphics window.

The available graphics objects are described in greater detail in the following sections:

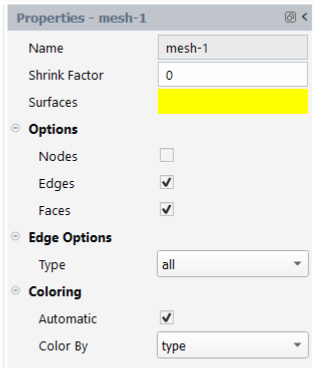


### 28.10.3.1. Mesh Plots

Mesh plots allow you to visualize and inspect the mesh.

To create a mesh plot, right-click **Meshes** in the Outline View tree and select **New...**.

**Results** → **Graphics** → **Meshes** → **New...**



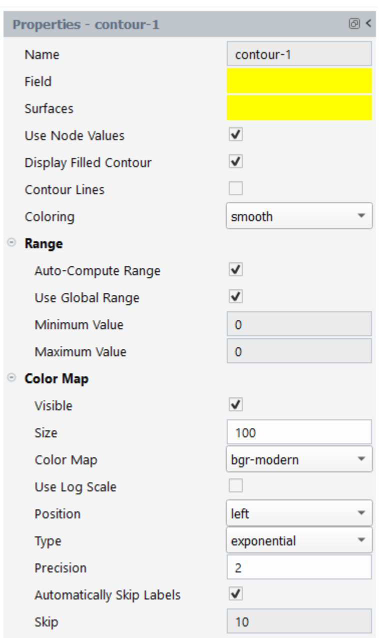

Once you have made your property settings, click **Display** to visualize the object in the graphics window. You can also click **Save Image** to save an image file of the object in the graphics window.

### 28.10.3.2. Contour Plots

Contour plots are a valuable postprocessing tool that allow you to use color to represent the values of the specified field variable on the selected surfaces.

To create a contour plot, right-click **Contours** in the Outline View tree and select **New...**.

**Results** → **Graphics** → **Contours** **New...**



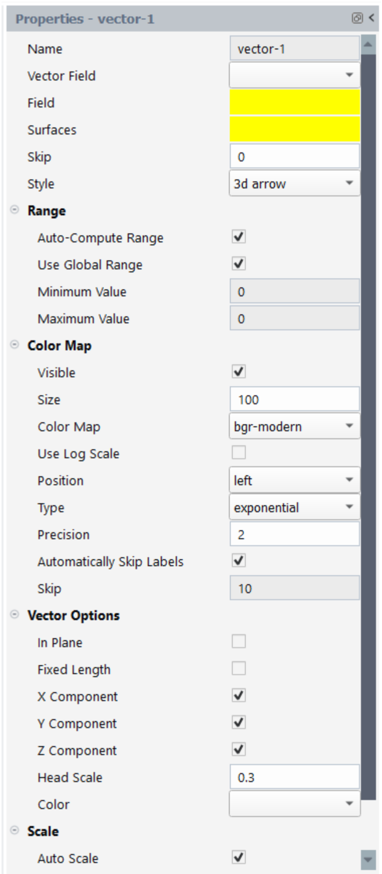

Once you have made your property settings, click **Display** to visualize the object in the graphics window. You can also click **Save Image** to save an image file of the object in the graphics window.

### 28.10.3.3. Vector Plots

You can draw vectors in the entire domain, or on selected surfaces. By default, one vector is drawn at the center of each cell (or at the center of each facet of a data surface), with the length and color of the arrows representing the velocity magnitude . The spacing, size, and coloring of the arrows can be modified, along with several other vector plot settings. Note that cell-center values are always used for vector plots; you cannot plot node-averaged values.

To create a vector plot, right-click **Vectors** in the Outline View tree and select **New...**

**Results** → **Graphics** → **Vectors** **New...**



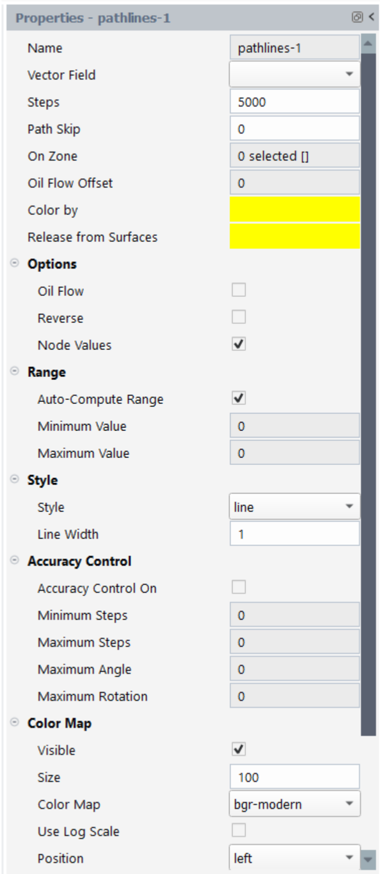

Once you have made your property settings, click **Display** to visualize the object in the graphics window. You can also click **Save Image** to save an image file of the object in the graphics window.

### 28.10.3.4. Pathline Plots

Pathlines are used to visualize the flow of massless particles in the problem domain. The particles are released from one or more surfaces that you have created as described in Surfaces (p. 296). A **line** or **rake** surface (see Line Surfaces (p. 297) and Rake Surfaces (p. 297)) is most commonly used.

To create a pathline plot, right-click **Pathlines** in the Outline View tree and select **New...**.

**Results** → **Graphics** → **Pathlines** **New...**



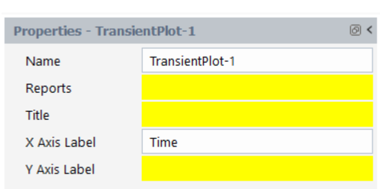

Once you have made your property settings, click **Display** to visualize the object in the graphics window. You can also click **Save Image** to save an image file of the object in the graphics window.

## 28.10.3.5. Transient Plots

Transient plots are a valuable postprocessing tool that allow you to use study time-dependant aspects of your simulation.

To create a contour plot, right-click **Contours** in the Outline View tree and select **New...**.

**Results** → **Graphics** → **Transient Plots** **New...**




Once you have set your properties:

- Click **Print** to create a hard copy of the plot.
- Click **Export** to save the report in another format.

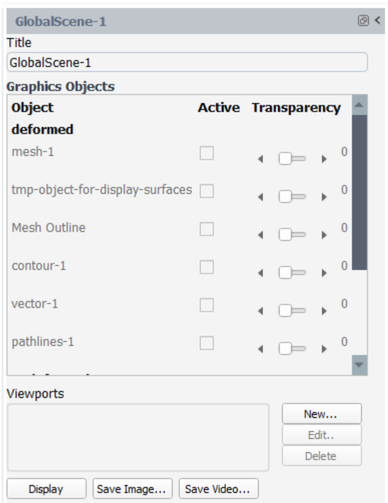
- Click **Plot** to display the plot in the graphics window.

### 28.10.3.6. Scenes

Scenes can be used to display multiple graphics plots within a single window. For example, you could overlay contours of pressure across a valve with velocity vectors and the mesh at the same location. Scenes allow you to modify the transparency of each plot so that you can emphasize a particular plot or view.

To create a new scene, right-click **Scenes** in the Outline View tree and select **New...**.

**Results** → **Graphics** → **Scenes** **New...**




1. (Optional) Enter a **Name** for the scene.

2. Select the graphics objects to include in the scene from the **Graphics Objects** list.

3. Set the transparencies for the selected graphics objects by clicking the **Transparencies** field.

Once you have made your property settings, click **Display** to visualize the object in the graphics window. You can also click **Save Image** to save an image file of the object in the graphics window.

# Chapter 29: Fluent Materials Processing Workspace: 3D Polymer Extrusion Tutorial

This tutorial is divided into the following sections:



## 29.1. Introduction

This tutorial illustrates the simulation of a 3D extrusion process using the Fluent Materials Processing workspace. The workspace will allow you to set up and solve a polymer extrusion problem so you can easily obtain accurate prediction of the extrudate shape for a given die geometry under prescribed operating conditions.

In this tutorial you will learn how to use the Fluent Materials Processing workspace to:

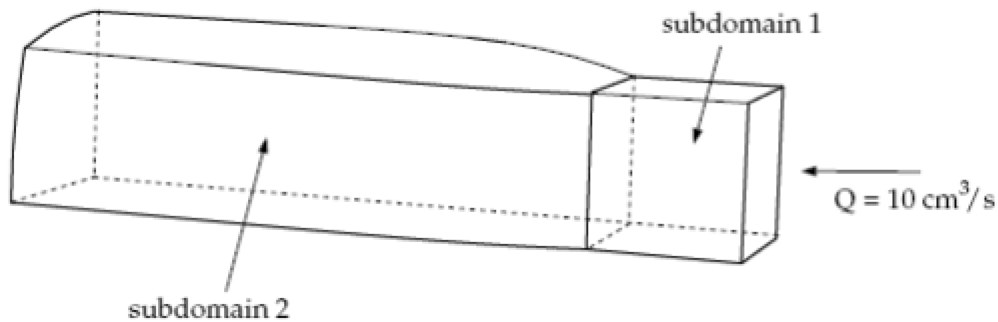
- Easily set up your extrusion simulation
- Calculate a solution
- Analyze the results.

## 29.2. Problem Description

This problem deals with the flow of a Newtonian fluid through a three-dimensional die. Due to the symmetry of the problem (the cross-section of the die is a square), the computational domain is defined for a quarter of the geometry and two planes of symmetry are defined.

The melt enters the die as shown in Figure 29.1: Problem Description (p. 310) at a flow rate of $Q$ = 10 cm$^3$/s (a quarter of the actual flow rate) and the extrudate is obtained at the exit. At the end of the computational domain, it is assumed that the extrudate is fully deformed and that it will not deform any further. It is assumed that subdomain 2 is long enough to account for all the deformation of the extrudate.

**Figure 29.1: Problem Description**

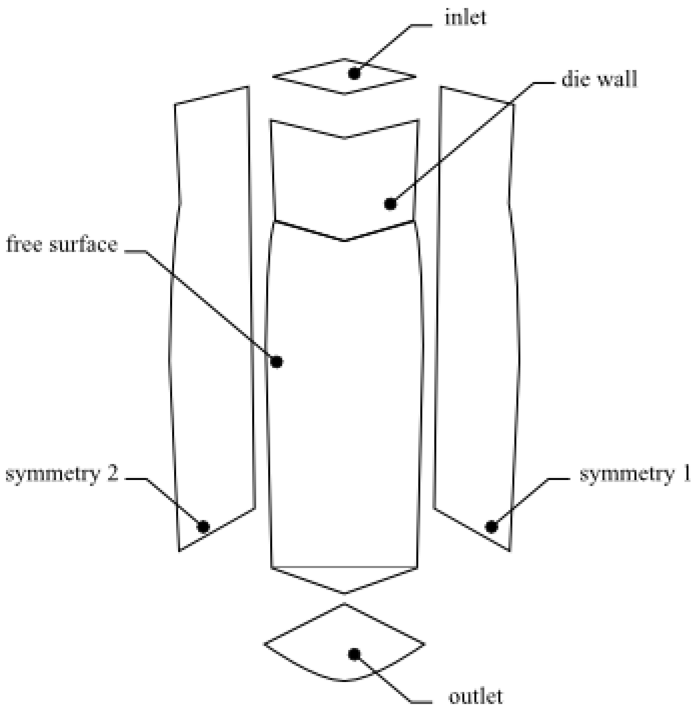



The incompressibility and momentum equations are solved over the computational domain. The domain for the problem is divided into two subdomains (as shown in Figure 29.1: Problem Description (p. 310)) so that the remeshing algorithm can be applied only to the portion of the mesh that will be deformed. The subdomain 1 represents the die where the fluid is confined. The subdomain 2 corresponds to the extrudate that is in contact with the air and can deform freely. The main aim of the calculation is to find the location of the free surface (the skin of the extrudate).

The boundary sets for the problem are shown in Figure 29.2: Boundary Sets for the Problem (p. 311), and the conditions at the boundaries of the domains are as follows.

- boundary 1: flow inlet, volumetric flow rate $Q$ = 10 cm$^3$/s
- boundary 2: zero velocity
- boundary 3: symmetry plane
- boundary 4: symmetry plane
- boundary 5: free surface
- boundary 6: flow exit

**Figure 29.2: Boundary Sets for the Problem**




# 29.3. Setup and Solution

The following sections describe the setup and solution steps for this tutorial:



## 29.3.1. Preparation

To prepare for running this tutorial:

1. Prepare a working folder for your simulation.

2. Download the `3d_extrusion.zip` file here.

3. Unzip the `3d_extrusion.zip` file you have downloaded to your working folder.

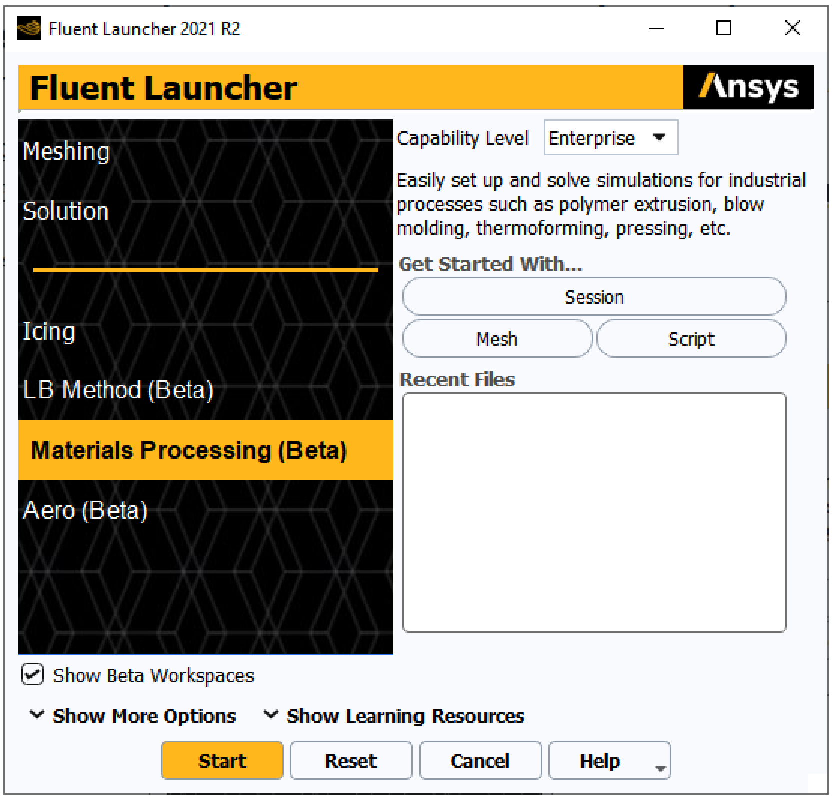
4. *The mesh file* `ext3d.msh` *can be found in the unzipped folder.*

## 29.3.2. Launching Ansys Fluent

1. Use the Fluent Launcher to start Ansys Fluent.

2. Enable **Show Beta Workspaces**

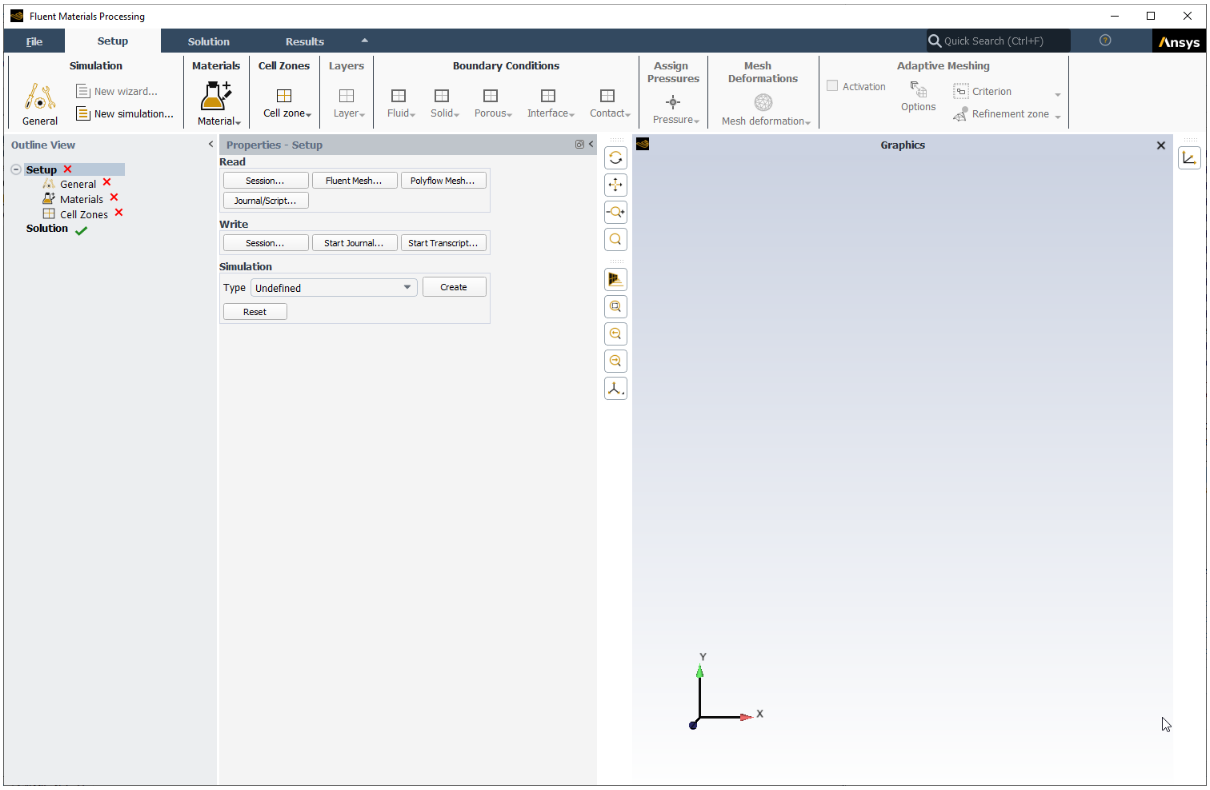
3. Select Fluent **Materials Processing (Beta)** in the list of Fluent workspaces.




4. Click **Start**.

## 29.3.3. Setup Your Simulation

Clicking **Start** in the Fluent Launcher opens the Fluent **Materials Processing** workspace.




The workspace is a version of Ansys Fluent that utilizes the power of the Ansys Polyflow solver to simulate polymer flows such as extrusion, blow molding, pressing, and so forth.

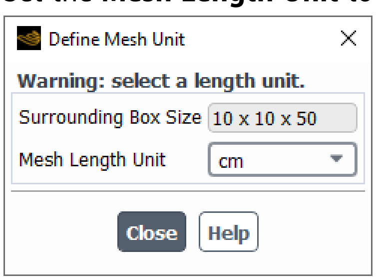
1. Open an extrusion geometry by reading in a Polyflow mesh file.

   a. In the **Setup** properties, under **Read**, click the **Polyflow Mesh** button.

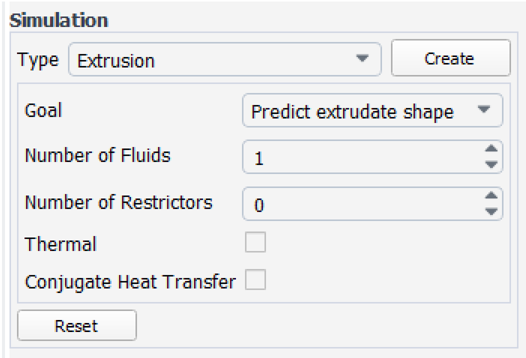
      *You can also use the **File** menu, and choose **Read** > **Polyflow Mesh**.*

   b. Locate and select the mesh file from your working folder (`extrusion.msh`)

   c. Set the **Mesh Length Unit** to **cm**, and close the dialog box.




2. Set up an extrusion simulation.




a. Under **Simulation**, set the **Type** to **Extrusion**.

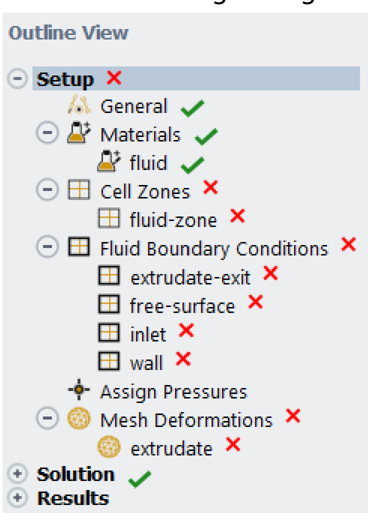
b. Set the **Goal** to be **Predict extrudate shape**.

c. Keep the **Number of fluids** as 1.

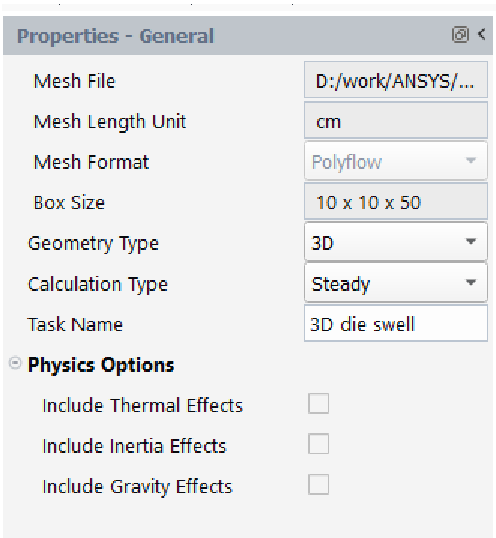
d. Keep the **Number of restrictors** as 0.

e. Click **Create**.

This will instruct the workspace to set up the appropriate objects and settings based on your selections. Notice the Outline View's use of status icons. A green check mark indicates the properties of that object are satisfactory. A red 'x' indicates that attention is required for that object. You can progress down the Outline View (or across the Ribbon) to complete your simulation settings using each object's property pages.



## 29.3.4. General Properties




1. Select **General** in the Outline View to review general properties of the simulation.

2. Review the settings in the properties page.

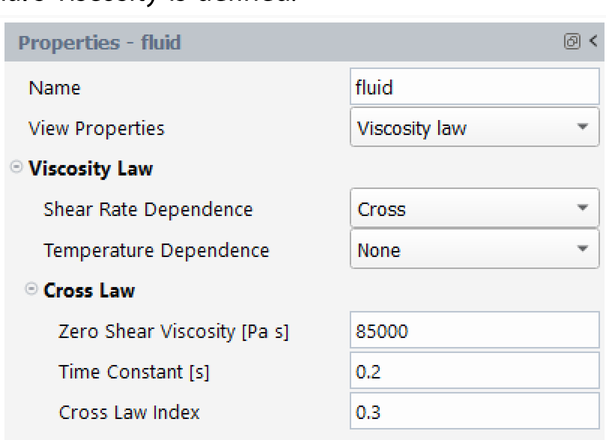
3. Make sure that **Steady** is selected for the **Calculation Type**.

4. Enter `3D die swell` for the **Task Name**.

## 29.3.5. Material Properties

*Fluent indicates which material properties are relevant by graying out the irrelevant properties. For this model you will only define the viscosity of the material.*

Even though the default material's (**fluid**) properties are up-to-date, you will change how that material's viscosity is defined.




1. Select **fluid** under **Materials** in the Outline View to review material properties of the simulation.

2. For **View Properties**, select **Viscosity Law**.

3. For **Shear Rate Dependence**, select **Cross**.

*The Cross law exhibits shear-thinning (the decrease in viscosity as the shear rate increases) that is a characteristic of many polymers. The viscosity in this tutorial is given by the Cross law:*

$$\eta = \frac{\eta_0}{1 + \left(\lambda \dot{\gamma}\right)^m} \quad (29.1)$$

where:

$\eta_0$ = zero shear-rate viscosity = 85000 poise

$\lambda$ = natural time = 0.2 s

$m$ = Cross law index = 0.3

$\dot{\gamma}$ = shear rate

4. For **Zero Shear Viscosity** ( $\eta_0$), enter `8500`.

5. For **Time Constant** ( $\lambda$), enter `0.2`.

6. For **Cross Law Index** ( $m$), enter `0.3`.

## 29.3.6. Cell Zone Properties

1. Select **fluid-zone** under **Cell Zones** in the Outline View to review cell zone properties of the simulation.

2. For **Zones**, select the field to open a selection dialog.

3. In the selection dialog, select both available subdomains: **subdomain1** and **subdomain2** and click **OK** to keep your selections and close the dialog.

## 29.3.7. Boundary Condition Properties

*In the following steps you will set the conditions at each of the boundaries of the simulation.*

1. Set inlet conditions.

   a. Select **inlet** under **Fluid Boundary Conditions** in the Outline View to review inlet boundary conditions for the simulation.

   b. Select **boundary1** as the **Boundary Zone**.

   c. Select **Volume Flow** as the **Flow Specification**.

   d. Select **fluid** as the **Incoming material**.

   e. Enter `0.00001` m^3/s (10 cm^3/s) as the **Volume Flow**.

2. Set outlet conditions.

   a. Select **extrudate-exit** under **Fluid Boundary Conditions** in the Outline View to review exit boundary conditions for the simulation.

b. Select **boundary6** as the **Boundary Zone**.

3. Set symmetry conditions.

   a. Select **Fluid Boundary Conditions** in the Outline View, and click the **New...** button in its properties window, or right-click in the tree and add a new condition to the tree.

   b. Rename the new condition to **Symmetry**.

   c. Set the **Type** to **Symmetry**.

   d. Select **boundary3-subdomain1**, **boundary3-subdomain2**, **boundary4-subdomain1**, and **boundary4-subdomain2** as the **Boundary Zone**.

4. Set free surface conditions.

   a. Select **free-surface** under **Fluid Boundary Conditions** in the Outline View to review free surface boundary conditions for the simulation.

   b. Select **boundary5** as the **Boundary Zone**.

   c. Select **boundary2** as the **Fixed Part**.

5. Set wall conditions.

   *At a solid-liquid interface, the velocity of the liquid is that of the solid surface. Hence the fluid is assumed to stick to the wall. This is known as the no-slip condition because the liquid is assumed to adhere to the wall, and hence, has no velocity relative to the wall.*

   a. Select **wall** under **Fluid Boundary Conditions** in the Outline View to review wall boundary conditions for the simulation.

   b. Select **boundary2** as the **Boundary Zone**.

## 29.3.8. Mesh Deformation Properties

*This model involves a free surface, whose shape is unknown a priori, which will be calculated together with the flow equations. A portion of the mesh is affected by the relocation of this boundary, so a remeshing technique is applied on this part of the mesh. The free surface is entirely contained within subdomain 2, therefore only subdomain 2 is affected by the relocation of the free surface.*

1. Under **Mesh Deformations**, select **extrudate** to edit the default properties of the extrudate exit for your simulation.

2. Select **subdomain2** for **Zones**.

3. Select **Optimesh 3D** for the **Extrudate Deformation Method**.

4. Select **interface-subdomain2-subdomain1** for the **Inlet Section**.For the Inlet section, select "interface subdomain2

5. Select **boundary6** for the **Outlet Section**.

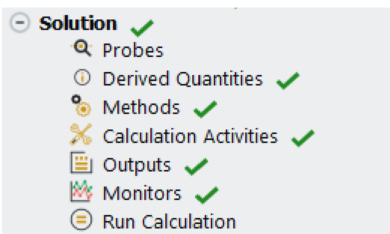
*The purpose of the remeshing technique is to relocate internal nodes according to the displacement of boundary nodes due to the motion of the free surface, since a part of the mesh is deformed. For 3D extrusion problems where large deformations of the extrudate are expected, the optimesh remeshing technique is recommended*

*The optimesh remeshing technique requires the direction of extrusion to be parallel to the $x$, $y$, or $z$ axis, and all slices into which the remeshing domain is cut must be perpendicular to the extrusion axis.*

*The domain to be remeshed is cut into a series of 2D slices (planes) in a direction perpendicular to the direction of extrusion, and each plane is remeshed independently. For this process, Polyflow requires the selection of the initial plane and the final plane. In this problem, the initial plane is the intersection of subdomain 2 with subdomain 1, and the final plane is the intersection of subdomain 2 with the flow exit (boundary 6).*

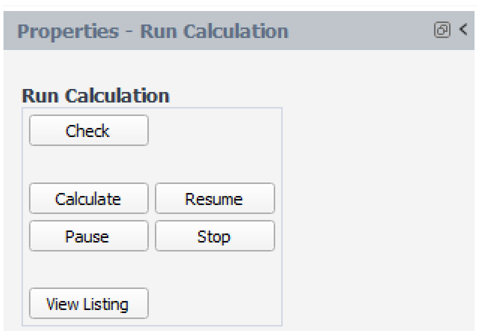
## 29.3.9. Solution

Open the **Solution** branch of the Outline View (or use the Ribbon). Here, you can review problem setup and other solution properties. Most items indicate that current default values are appropriate.




1. Review your problem setup.

    a. Select **Run Calculation** to see available options.

    b. Click **Check** to review your simulation settings. Fluent will inform you as to whether or not your setup contains any problems and will provide guidance to solve any issues.

2. Calculate a solution.

    a. Click **Calculate** to begin the computing a solution.

    b. Once the calculations are complete, click **View Listing** to check the solution's listing to confirm its success. You can confirm that the solution proceeded as expected by looking for the following printed at the bottom of the listing file:

    ```
    The computation succeeded.
    ```

## 29.4. Results

Review results using the various tools in the **Results** section of the Outline View or the Ribbon where you can setup contours, vectors, and so on.

1. Display the velocity distribution on the boundaries.

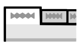
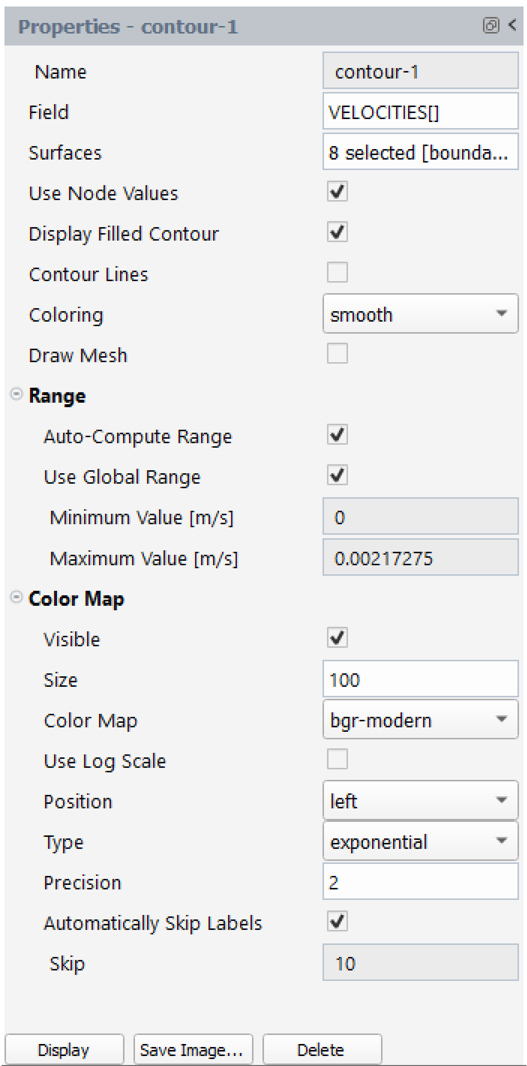
    **Results** → **Contours** → **New...**

   **Figure 29.3: Contour Properties**

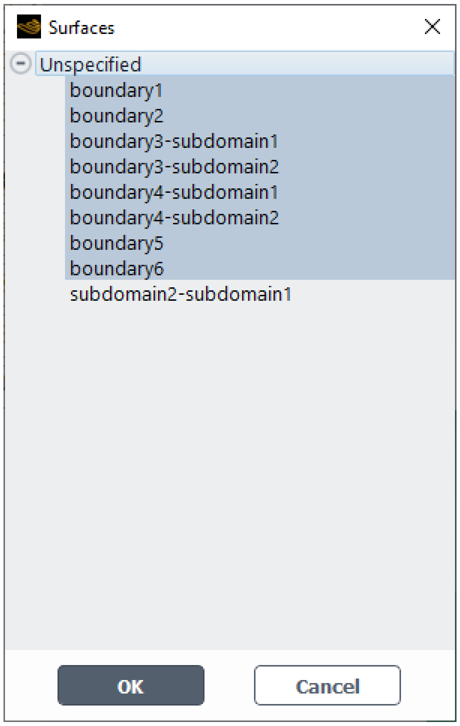
   
   

   a. Keep the default **Name** (`contour-1`).

   b. For the **Field**, select **VELOCITIES**.

   c. For the **Surfaces**, select he field to open the **Surfaces** dialog where you can select the relevant boundaries. Once your selections are complete, click **OK** to close the dialog.

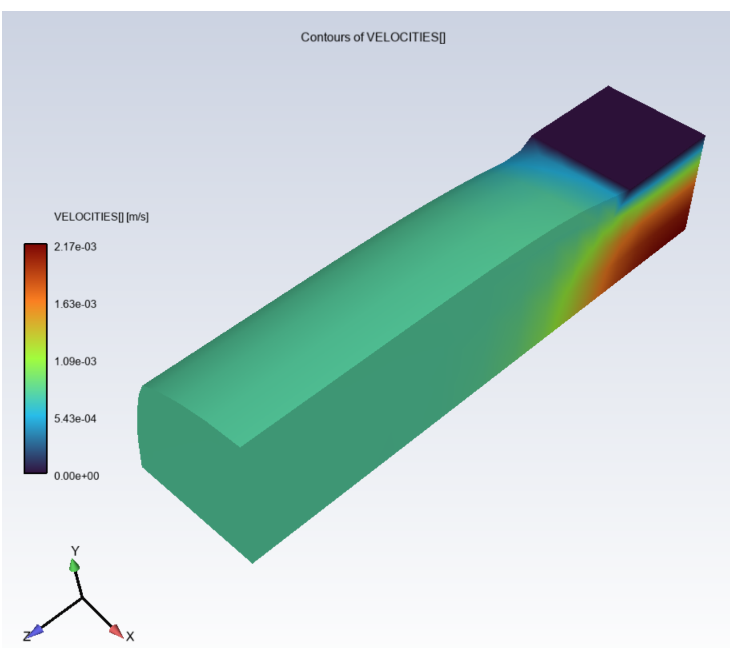
**Figure 29.4: Selecting Boundaries in the Surfaces Dialog**




d. Keep the remaining defaults and click **Display** to see the velocity magnitude contour plot in the graphics window.

   Use the toolbars in the graphics window to adjust the display so that it is in an isometric orientation, and fits the window.

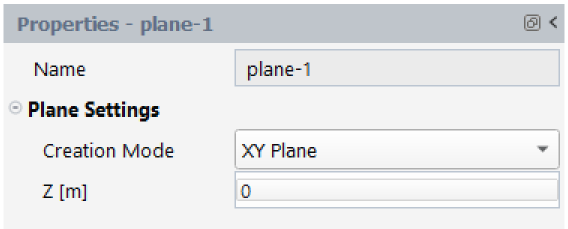
**Figure 29.5: Contours of Velocity Magnitude**



2. Display contours of velocity in cross-sectional planes, at Z = 0, 0.08, 0.15, and 0.45 m.

   a. Create a cross-section planes at Z = 0 m.

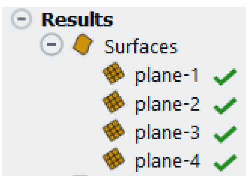
      **Results** → **Surfaces** → **New Plane...** →

      In the properties of **plane-1**, under **Plane Settings**, select **XY Plane** for the **Creation Mode**, enter `0` for **Z**, and click **Display**

   b. Create additional cross-section planes for Z = 0.08, 0.15, and 0.45 m.

      Once complete, you should have four separate plane surfaces available.

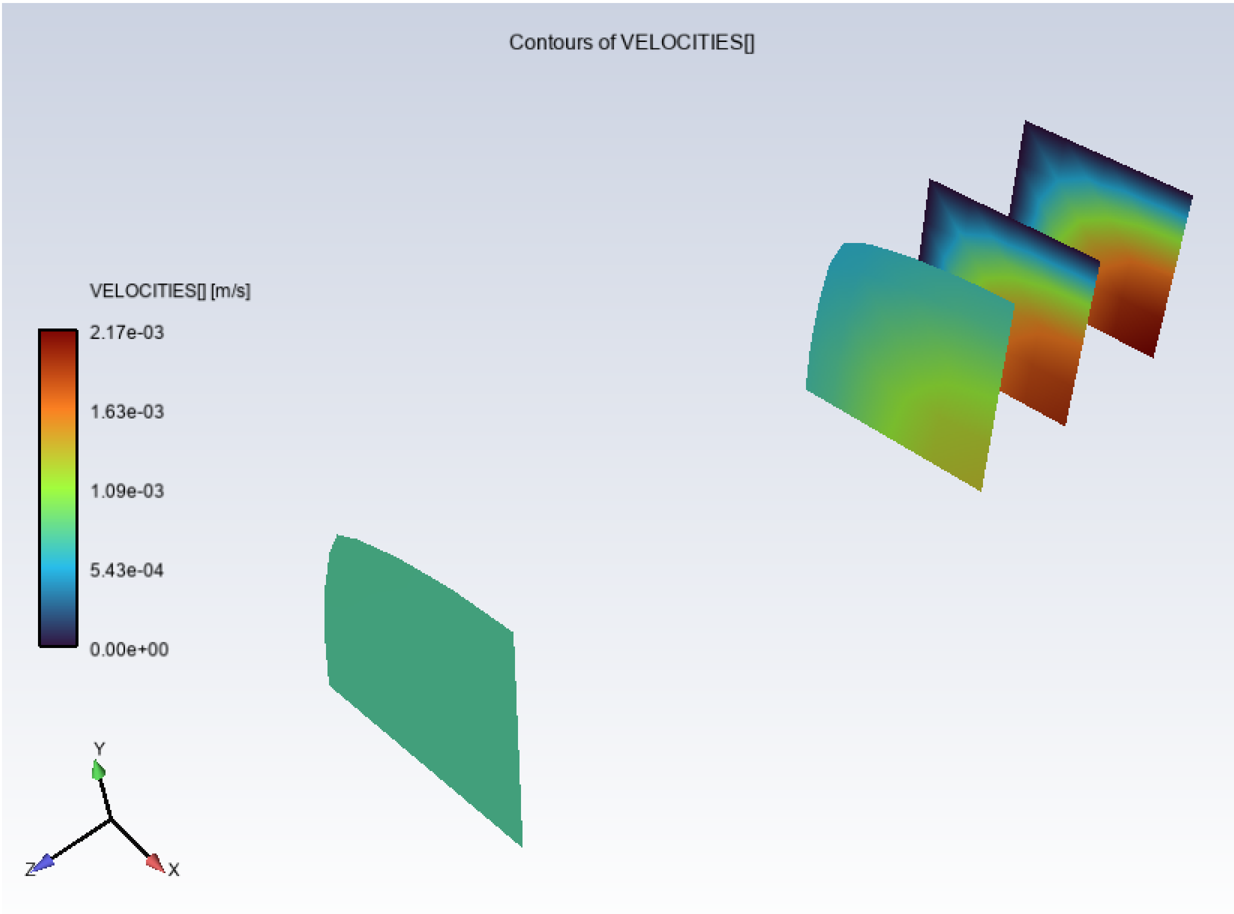
      *You could also have used the **Create Multiple Surface** dialog to easily create multiple surfaces.*

   c. Display contours of velocity in the cross-sectional planes.

      **Results** → **Contours** → **New...**

      In the properties of **contour-2**, select **VELOCITIES** for the **Field**, and for the **Surfaces**, select the four plane surfaces you just created, keep the remaining defaults, and click **Display**..

**Figure 29.6: Velocity Profiles at Cross-Sections**




## 29.5. Summary

This tutorial introduced the concept of a 3D extrusion problem using the Fluent Materials Processing workspace. You solved the problem using a specific 3D geometry for the die and made suitable assumptions about the physics of the problem. You analyzed the factors affecting the extrudate shape. In addition you learned how to use the optimesh remeshing method, which is recommended for 3D extrusion problems.

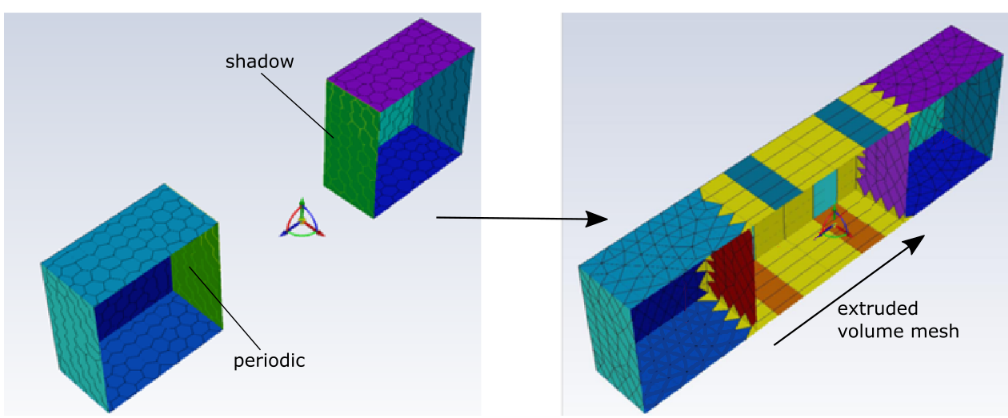
# Chapter 30: Fluent Meshing

This chapter contains information relating to using beta features in Ansys Fluent 2021 R2 in Meshing mode.



## 30.1. Volume Mesh Extrusion for Periodic Boundaries

When beta features are enabled, the Fluent Meshing guided workflow's **Extrude Volume Mesh** task includes an additional **Method** when extruding volume meshes. If translational periodic boundaries have been defined in the **Set Up Periodic Boundaries** task (Setting Up Periodic Boundaries), you can extrude the selected translational periodic face to its shadow face using the **Extend to Periodic** method, and setting the remaining options. Note that when defining the translational periodic boundary, the periodic and shadow faces should be connected to two *different* cell zones.




## 30.2. Fault-tolerant Meshing: Managing Zones

When beta features are enabled, the Fluent Fault-tolerant Meshing guided workflow includes a **Manage Zone** task to help you better manage your zones, and is similar to the **Manage Zone** task in the Watertight Geometry workflow (Managing Zones).

After you have generated your volume mesh, you can use the **Manage Zones** task make additional changes to your cell and/or face zones (such as renaming or merging) prior to proceeding to the Fluent solver. This can especially useful for large models with numerous zones.

---

**Note:**

Labels and bodies should **not** have the same name, since doing so would lead to ambiguities. Bodies whose name contains the string 'fluid' will automatically be assigned as fluid regions, however, the same is not true for solids, since that is usually the default name for any body in both SpaceClaim and DesignModeler.

---

1. Set the **Type** of zone to **Cell Zone** or **Face Zone**.

2. Apply type filtering for the zones:

   - For cell zones, use the **Type Filter** to filter the list of available objects as **Fluid** or **Solid** or **All**.

   - For face zones,use the **Type Filter** to filter the list of available objects as **Internal**, **Fluid-Fluid**, **Solid-Fluid**, **Fluid-Solid**, **External-Solid**, **External-Fluid**, or **External**.

3. Apply value filtering for the zones:

   - For cell zones, use the **Volume Filter** to only display a list of cell zones within a certain volume.

   - For face zones: use the **Area Filter** to only display a list of face zones within a certain area.

   - You can set either filter to **All**, **Less than**, **More than**, or **Equal to** the corresponding **Area** value (for face zones) or **Volume** (for cell zones). When using the **Equal to** option, for the**Equal within a range of (%)** field, specify a percentage range for the volume/area.

4. Additional filtering options and wildcards are available in the list as well to help in selecting cell zones.

5. To merge selected cell/face zones:

   i. Select **Merge** as the **Operation**.

   ii. Specify the portion of the **Name** of the selected zones that you want to merge, or keep the default name. The task attempts to create an appropriate name. If the target name already exists, then a numerical value will be appended to the name (such as `my_label_1`, `my_label_2`, etc.).

   iii. For **Do you want to merge adjacent faces?**, select either **Yes** or **No**.

   ---

   **Note:**

   The merging of cell zones also allows you to merge the underlying face zone(s). Zones with face labels (named selections), however, are not supported during the face merge.

   ---

6. To change the prefix of the names of selected cell/face zones:

    i. Select **Change Prefix** as the **Operation**.

    ii. Use the **From** field to indicate the prefix that you want to change, and use the **To** field to indicate what you would like to change the prefix to. The task attempts to change the prefix accordingly. If the target name already exists, then a numerical value will be appended to the name (such as `my_label_1`, `my_label_2`, etc.).

7. To rename selected cell/face zones:

    i. Select **Rename** as the **Operation**.

    ii. Specify a new **Name** for the selected zone. The task attempts to create an appropriate name. If the target name already exists, then a numerical value will be appended to the name (such as `my_label_1`, `my_label_2`, etc.).

8. Click **Manage Zone** to apply your changes for this task. Use the **Draw Mesh** button to display the surface mesh to inspect any zone changes.

    If you need to make adjustments to any of your settings in this task, click **Revert and Edit**, make your changes and click **Update**, or click **Cancel** to cancel your changes.

## 30.3. Enhanced Orthogonal Quality

When beta features are enabled, you can employ an enhanced definition of the orthogonal quality measure using the `report/enhanced-orthogonal-quality` text user interface command. When enabled, the enhanced definition of the Orthogonal Quality measure combines the properties of squish, cell shape quality, and cell face warpage, and is optimal for thin prism cells. The range is from -1 to 1. Under ideal conditions, the orthogonal quality is close to 1, whereas valid cells require the quality to be more than 0. The squish reflects the orthogonal quality of one face with the edge vector of the face centroid and the cell centroid. The cell shape quality reflects the shape of the cell. It is designed to detect bad cell shape at one local edge, and it can detect twisted cells or edge concave cells. The cell face warp reflects the variation of normals between the faces that can be constructed from the cell face.

As noted, the worst cells will have an Orthogonal Quality closer to 0 and the best cells will have an Orthogonal Quality closer to 1. The following table lists the range of orthogonal quality values and the corresponding cell quality.

**Table 30.1: Orthogonal Quality Ranges and Cell Quality**

| Orthogonal Quality | Cell Quality |
|---|---|
| 1 | orthogonal |
| 0.9–<1 | excellent |
| 0.75–0.9 | good |
| 0.5–0.75 | fair |
| 0.25–0.5 | poor |
| >0–0.25 | bad (sliver) |

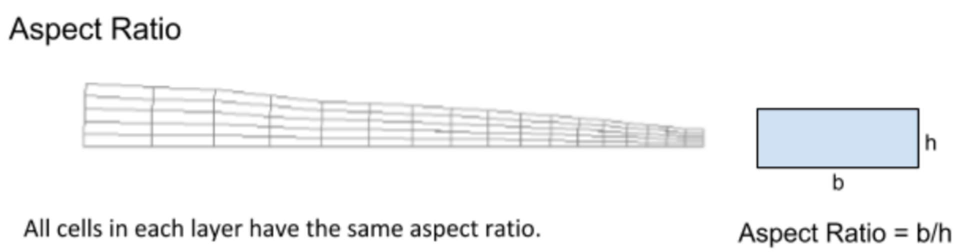
| Orthogonal Quality | Cell Quality |
|---|---|
| 0 | degenerate |

## 30.4. Fault-tolerant Meshing: Overset Meshing & Boundary Layer Prism Growth

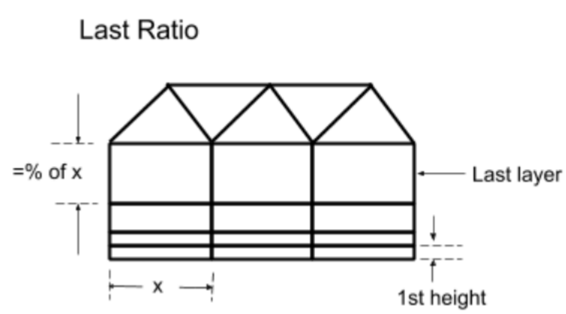
When beta features are enabled, the Fault-tolerant Meshing guided workflow's **Create Component Meshes** task includes an additional **Type** when creating component meshes. Use the **Grow Prisms** option to create a component mesh by growing a boundary layer (prism layer) around one or more selected objects from the imported geometry.

When the **Type** is set to **Grow Prisms**, additional settings are required:

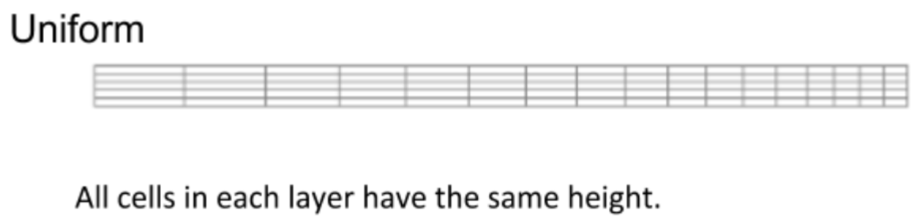
- Choose an **Offset Method Type**. The offset method that you choose determines how the mesh cells closest to the boundary are generated. Choices include:
  - **aspect-ratio**: allows you to control the aspect ratio of the boundary layer cells (or prism cells) that are extruded from the base boundary zone. The aspect ratio is defined as the ratio of the prism base length to the prism layer height.

  - **last-ratio**: allows you to control the aspect ratio of the boundary layer cells (or prism cells) that are extruded from the base boundary zone. You can specify **First Height** for the first prism layer.

  - **uniform**: allows you to generate every new node (child) to be initially the same distance away from its parent node (that is, the corresponding node on the previous layer, from which the direction vector is pointing).




i. Specify the **Number of Layers**. This value determines the maximum number of boundary layers to be created in the mesh.

ii. If the **Offset Method Type** is set to **uniform**, specify the **First Height**. This value is the height of the first layer of cells in the boundary layer.

iii. If the **Offset Method Type** is set to **aspect-ratio**, specify the **Aspect Ratio**. You can control the heights of the inflation layers by defining the aspect ratio of the inflations that are extruded from the inflation base. The aspect ratio is defined as the ratio of the local inflation base size to the inflation layer height.

iv. If the **Offset Method Type** is *not* set to **last-ratio**, provide the **Growth Rate**. This value determines the relative thickness of adjacent inflation layers. As you move away from the face to which the inflation control is applied, each successive layer is approximately one growth rate factor thicker than the previous one.

Additionally, you can provide a value for the **Last Ratio Percentage**.

## 30.5. Reference Frames (TUI)

When beta features are enabled, the `reference-frames/` text command menu is available in the meshing mode of Fluent. With this text command menu and the corresponding subcommands, you have the ability to perform basic manipulations of reference frames.

```
Meshing/reference-frames>
add                     hide                    list-properties
display                 list
```

- `add` allows you to add an object
- `display` allows you to display a reference frame
- `hide` allows you to hide a reference frame
- `list` allows you to list objects
- `list-properties` allows you to list the properties of an object

## 30.6. CAD Import Options (TUI)

When beta features are enabled, the `file/import/cad-options/` text command menu contains the `create-label-per-body-during-cad-faceting?` command, that creates labels for the selected objects individually during faceting upon import.

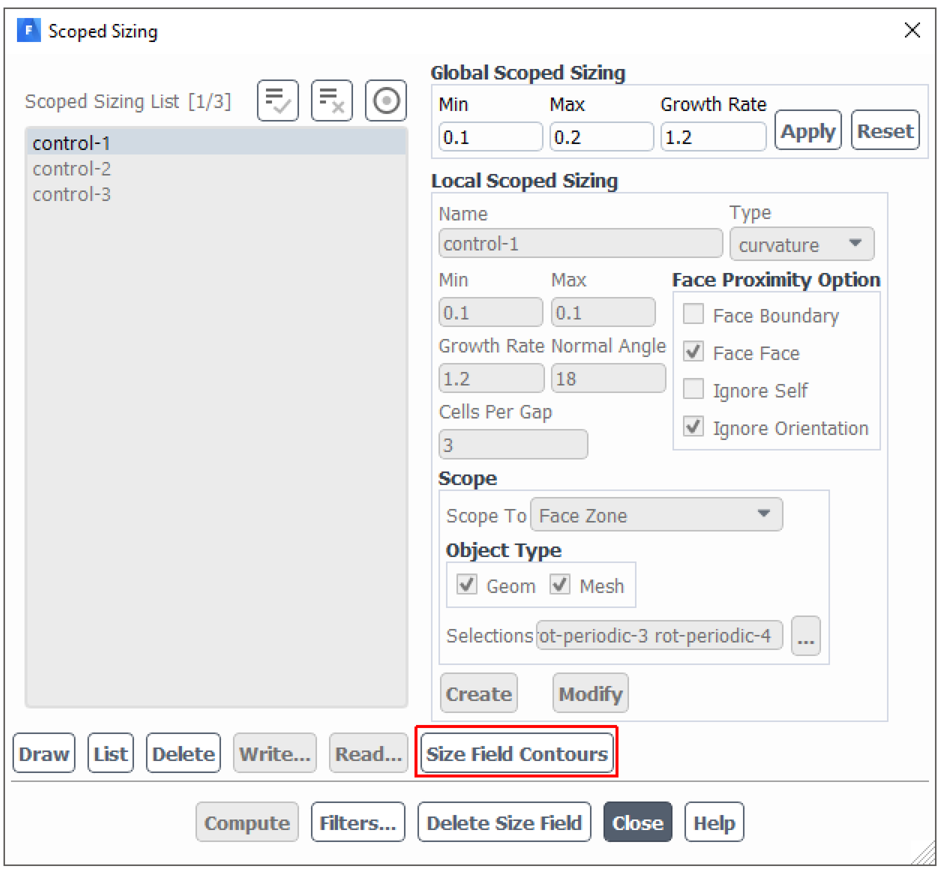
## 30.7. Saving Data and Checkpoints (TUI)

When beta features are enabled, the `file/checkpoint/` text command menu is available and contains options for saving data in your workflow. The term *checkpointing* refers to a means of storing the mesh in the memory (at selective checkpoints), instead of writing the mesh data to a file. Available subcommands are:

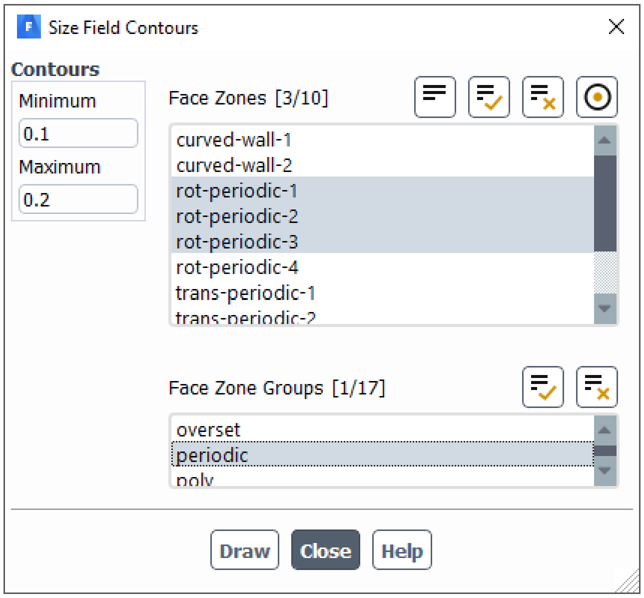
- `delete-checkpoint` removes a specifically named checkpoint.
- `list-checkpoint-names` provides a list of checkpoint names.
- `restore-checkpoint` restores to specifically named checkpoint.
- `write-checkpoint` writes a specifically named checkpoint.

## 30.8. Accessing Size Field Contours

When beta features are enabled, you can have access to size field contours directly from the **Scoped Sizing** dialog box.




Once a scoped sizing control is computed in the dialog box, you can click the **Size Field Contours** button to open the **Size Field Contours** dialog box.




Here, you can set the **Minimum** and **Maximum** contour values, and select **Face Zones** and/or **Face Zone Groups** in order to draw the contours for the size field.

# Chapter 31: Fluent's Virtual Blade Model

A method for analyzing the aerodynamic interaction between multiple rotors and airframes has been implemented as an add-on module for the general-purpose CFD solver Fluent. With this technique, the 3D rotors are replaced with actuator disks that introduce the effect of the rotors via implicit momentum source terms in the governing equations. Since the blades are not physically modeled, the technique is called the Virtual Blade Model (VBM). The model supports hybrid unstructured meshes for easy handling of multiple rotor geometries in close proximity, with convenient local mesh clustering. The non-linear aerodynamic interaction of rotor wakes with each other and with other structural components is obtained by coupling the VBM with Fluent's Navier-Stokes solver. Accurate aerodynamic predictions of helicopter rotors are possible only if the rotors operate at desired thrust and zero moment about the hub. Trimming is performed in an automatic and robust fashion using an iterative method to account for the non-linear relation between blade pitch and rotor performance. A propeller and a well-studied, simplified single rotor helicopter model in forward flight are presented at the end of the document as examples of the capabilities of the Fluent VBM.

## 31.1. Introduction

The flow around a rotorcraft is strongly unsteady and three-dimensional. This complexity is attributable to the unique aerodynamic characteristics of rotary wings and to the reciprocal aerodynamic interaction of the stationary and rotating components that make up a rotorcraft. For a simple helicopter, for example, strong aerodynamic interactions can be observed between the main rotor and the airframe, between the main rotor and the tail rotor, and between the tail rotor and airframe. These interactions dominate the overall flow field and therefore the performance of the helicopter.

Due to the complexity of the flow around rotorcraft, engineers require powerful tools to optimize its performance and stability. Wind tunnel time is difficult to obtain and very expensive, and often of limited effectiveness. Hence, the ability to accurately simulate these complicated flow fields with computational tools becomes increasingly important. Numerical methods can handle many levels of complexity, therefore the computational approach is dictated by the objective of the simulation, which can range from obtaining detailed blade characteristics (stall behavior, noise, FSI, etc.), to cases where the time-averaged cumulative effects of the rotors on each other and the airframe are more than sufficient. For the latter, typical applications would be the prediction of fuselage drag and tail forces for a range of operating conditions, as well the effect of engine exhaust plume distribution and impingement on skin heating and IR-signature prediction. These time-averaged analyses are carried out using reduced-order models such as the Virtual Blade Model.

Historically, in time-averaged simulations, the rotors and their blades are replaced by rotor disk surrogates of equal diameter, or actuator disks, that are coupled with three-dimensional Navier-Stokes or Euler solvers. Since the rotors and their blades are not modelled directly, the computational grids are much smaller, less complex and require significantly less mesh generation effort. With the time-averaged approach, the computational time is also drastically reduced.

Two distinct actuator disk approaches exist. The pressure-disk rotor model simulates helicopter rotors or propellers in a time-averaged manner using a disk composed of two surfaces, one side representing an outflow boundary, the other an inflow boundary. The disk does not need to have a finite thickness, and the grid nodes of the two surfaces need not be paired and coincident. A pressure jump, function of radius and azimuth, is imposed across the disk, subject to the constraint that mass be explicitly conserved through the disk surfaces.

Zori et al.6 (References (p. 361)) developed an alternate technique that replaces the rotor system with momentum sources acting on a one-cell-thick actuator disk that is an integral part of the mesh and does not have inflow and outflow boundaries. The advantage of this approach is that mass is automatically conserved across the disk. Both methods compute the rotor forces according to the Blade Element Theory, using lift and drag coefficient look-up tables for the stacks of airfoils that represent the blades. The Virtual Blade Element formulation is introduced into Ansys Fluent with a User-Defined Function (UDF).

Accurate aerodynamic predictions are possible only if the rotors operate at the desired thrust and zero moments about the hub. These targets can be met by varying the collective (thrust) and the cyclic (moments) blade pitch angles through user input or by a trimming algorithm included in the rotor disk model. Typically, the relationship between the thrust coefficient and the collective pitch angle, and between the hub moments and the cyclic pitch angles is assumed to be linear, making the trim routine easy to implement but numerically unstable. The lack of a robust trim routine historically limited the simulation to single rotor configurations, therefore ignoring the tail rotor. Only recently, Yang et al.6, suggested an automatic trim routine based on a Newton-Raphson iterative method to account for the non-linear relation between blade pitch and rotor performance. The current implementation of the Fluent VBM UDF is based on the models developed by Zori et al. and Yang et al. (References (p. 361)), with additional enhancements.

The Fluent VBM allows for the specification of rotor blades represented by stacks of 2D airfoil sections varying in twist, chord and airfoil type. The airfoil look-up tables, containing lift and drag coefficients vs. angle of attack, can also be functions of Mach and Reynolds number, allowing the accurate treatment of both low- and high-speed regimes. Furthermore, the UDF can handle up to twenty-five individual rotors simultaneously, permitting simulations of complete rotorcraft with both main and tail rotors, or other configurations, such as quadcopters, quad-tiltrotors, or other exotic multi-rotor configurations. Simulations of multiple rotorcraft operating in close proximity are also possible. Furthermore, in the current implementation, the rotor disks can also be meshed with hybrid unstructured grids, simplifying mesh construction for multiple rotors in close proximity and with convenient individual local mesh clustering.

## 31.2. The Virtual Blade Model (VBM)

This technique models the effect of the rotor on the flow field in a time-averaged manner through momentum equation source terms placed in a disk volume swept by the spinning rotor. Two tasks must be addressed simultaneously in this approach: 1) time-averaged solution of the flow field through and around the rotor; 2) reduced-order solution of the rotor blade aerodynamics to compute the required momentum source terms. In our approach for the first task outlined above, the flow field is solved using Ansys Fluent. Inviscid, or viscous laminar or turbulent simulations can be obtained, assuming that the fluid is either compressible or incompressible. The following is, therefore, an outline of our approach to the formulation and implementation of Task 2.

The momentum source terms, unknown at the start of the simulations, are linked to the flow solution through the actuator disks and therefore evolve as part of the iterative solution. The momentum source terms are evaluated using the Blade Element Theory by replacing the rotor blades with a stack of airfoil sections in the spanwise direction. Each section is treated as if it were a 2D airfoil. The geometric properties of each section, such as chord length, airfoil type and twist can vary along the span of the blade.

In typical CFD simulations, rotorcraft are assumed to be stationary and immersed in a moving airstream, hence airflow solutions around rotorcraft are computed in the absolute frame of reference. Since the blades are rotating around an axis with velocity $\Omega$, they are best described in a relative frame of reference defined by their rotation axis and origin, the radial direction along the blade and the azimuthal angle that defines the position of the blade in its circle of rotation. In VBM simulations, a third local 2D frame of reference is needed, defined by the tangential velocity vector at the blade section and the normal vector of the blade. In order to compute the forces acting on a blade section, it is necessary to transform the velocity field $\vec{U}$ computed in the absolute frame of reference to the local blade section frame of reference to obtain the relative velocity field $2\pi$, from which the local angle of attack ($\alpha$), Mach number ($Ma$), and Reynolds number ($Re$), can be determined. Look-up tables corresponding to each blade section are then used to extract the local $C_L$ and $C_D$ values, from which the instantaneous rotor forces can be computed in the form:

$$f_{L,D}=C_{L,D}\left(\alpha,Ma,\mathrm{Re}\right)\cdot c\left(r/R\right)\cdot\frac{1}{2}\rho u_{rel}^2$$

$u_{rel}$ being the instantaneous relative airflow velocity experienced by each blade cross-section in the frame of reference of the rotor. However, for the time-averaged solution in the absolute frame of reference, these forces must be time-averaged over one rotation. Assuming that the rotational speed is constant, time averaging over one period is identical to geometric averaging over an angle $2\pi$. Therefore, the resultant forces per-cell become

$$F_{L,D}\Big|_{cell}=N_b\frac{\Delta r\cdot r\Delta\phi}{2\pi r}f_{L,D}$$

where $N_b$, $r$ and $\phi$ are the number of blades and the spanwise and the radial and azimuthal coordinates, respectively.

The above force vector $F$ is transformed back into the absolute flow field reference frame vector $F'$ so that it can be included in the airflow calculation. Let $V_{cell}$ be the cell volume, then:

$$S'_{cell}=\frac{F'_{cell}}{V_{cell}}$$

is the time-averaged source term per unit volume that is added to the momentum equations in every cell attached to the rotor disk. Once these source terms have been computed, the solution of the flow field can be obtained, and the iterative procedure is repeated until the convergence criteria are met. This generalized implementation supports both structured and hybrid unstructured mesh topologies with hexahedral and prismatic elements.

The rotor performance is controlled by the thrust and moment coefficients, which are defined as

$$C_T=\frac{T}{C_{NA-EU}\rho V_{tip}^2\pi\frac{d^2}{4}}$$

$$C_{Mx}=\frac{M_x}{C_{NA-EU}\rho V_{tip}^2\pi\frac{d^2}{4}\frac{d}{2}}$$

$$C_{My}=\frac{M_y}{C_{NA-EU}\rho V_{tip}^2\pi\frac{d^2}{4}\frac{d}{2}}$$

where $C_{NA\text{–}EU}$ = 1 in North American (NA) practice and $C_{NA\text{–}EU}$ = ½ in European practice. In NA practice, the coefficient ½ is absorbed into the thrust and moment coefficients, whereas in European practice the formulae explicitly contain the coefficient ½, therefore

$$\left(c_T, c_{Mx}, c_{My}\right)\Big|_{EU} = 2\left(c_T, c_{Mx}, c_{My}\right)\Big|_{NA}$$

In the addon, $C_{NA\text{–}EU}$ is set to 1.

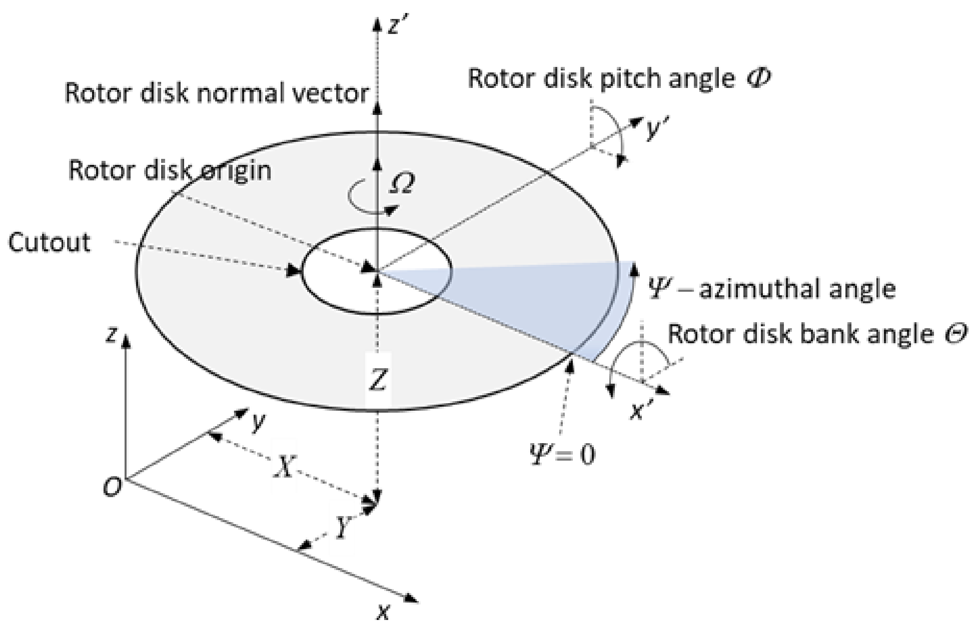
Finally, a comment should be made regarding turbulence modeling. CFD solvers such as Ansys Fluent employ a wide range of turbulence models to account for the effects of turbulence on the mean flow for steady-state fluid flow simulations. For applications where the VBM is used, it is suggested that standard two equation models such as $k-\varepsilon$ or $k-\omega$ be employed. It should be noted that the VBM model itself does not affect the turbulence model directly by increasing the production of turbulence kinetic energy within the modeled rotor zone. However, turbulent mixing is modeled downstream of the VBM rotor zones, and therefore turbulence is indirectly influenced by the momentum generated by the rotors.

## 31.2.1. Rotor Disks

The current implementation of the VBM allows a maximum of twenty-five rotor disks. This enables the simulation of complex rotorcraft with multiple rotors, from simple helicopters with main and tail rotor, to multiple tilt rotors, to quadcopters and even to multiple rotorcraft, and single- or multi-engine propeller aircraft.

The rotor disk geometry is characterized by the location of its center {*X*,*Y*,*Z*}, its radius and its pitch and bank angles, as shown in Figure 31.1: Rotor Disks Schematic (p. 334).

**Figure 31.1: Rotor Disks Schematic**




The natural orientation of the disk's normal vector points in the positive *Z*-direction and the tail of the helicopter points in the positive *X*-direction. Should the orientation of the normal vector be different, the disk must be rotated from its natural orientation using the disk pitch angle $\phi$ (rotation around the Y-axis) and bank angle $\Theta$ (rotation around the *x'*-axis $\{cos\phi, 0, sin\phi\}$). In other words, if a pitch rotation is applied, then the *x'*-axis of the rotor will no longer be parallel to the *X*-axis.

**Note:**

- Rotations are not additive.
- Rotations are non-commutative and must be applied consistently in this order.
- Rotations follow the right-hand rule.

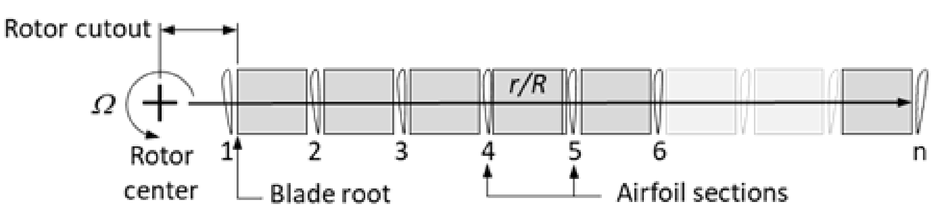
This also applies to the inclination of the rotor during CAD construction – the rotations must be applied in the same order. For example, in the case of an aircraft equipped with a propeller whose normal axis is aligned in the negative *X*-direction, the pitch and bank angles must be set to $\phi$ = –90˚, $\Theta$ = 0˚. Similarly, if the rotor axis is aligned in the positive *Y*-direction, the pitch and bank angles should be set to $\phi$ = 0˚, $\Theta$ = -90˚.

Rotor disk position can be also set using disk normal vector components instead of disk pitch and bank angles. In that case, Fluent's VBM computes disk pitch and bank angles based on above mentioned natural orientation and disk normal vector.

## 31.2.2. Blade Geometry

The blades are represented as stacks of airfoils which can vary in chord, twist and airfoil type along the normalized blade span $r/R$, as shown in Figure 31.2: 3D Blade Geometry Represented by a Stack of 2D Airfoils (p. 335). The VBM assumes a linear distribution of chord, twist and $C_L$ and $C_D$ between the two ends of each section. Simulation accuracy increases if the airfoil look-up tables contain data for a range of Reynolds ($Re$) and Mach numbers ($Ma$) covering all possible operating conditions, since the UDF can determine the local $Re$ and $Ma$ on the blade and interpolate $C_L$ and $C_D$ from the tables accordingly.

**Figure 31.2: 3D Blade Geometry Represented by a Stack of 2D Airfoils**




## 31.2.3. Blade Pitch

The blade pitch angle *q*(*y*), not to be confused with the rotor pitch angle, is the angle of the blade with respect to the rotor disk plane. To a first-order approximation (neglecting blade torsional flexing and other harmonics), it consists of the baseline collective angle $\theta_o$ and a cyclic pitch component expressed as a function of the sine and cosine of the azimuthal angle $\psi$:

$$\theta(\psi)=\theta_0-\theta_{lc}\cos\psi-\theta_{ls}\sin\psi$$

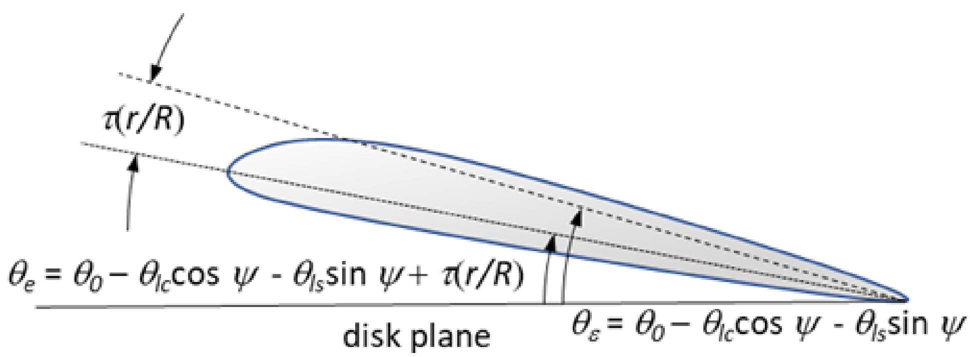
The blade pitch angles $\theta_0$, $\theta_{1c}$, and $\theta_{1s}$ can either be prescribed by you or calculated by the automatic rotor trim routine. The collective angle $\theta_0$ is the angle of the blade at some point on its span with respect to the disk surface. The angles $\theta_{1c}$ and $\theta_{1s}$, the amplitudes of the longitudinal and lateral components of the cyclic pitch, are directly related to the angles of the swash plate found on most helicopter rotors and therefore control the pitch and roll attitude of the rotorcraft. Most rotor/propeller blades also have a spanwise twist. The reference point for the pitch and twist angles can be defined arbitrarily at any position along the blade as long as they provide correct rotor pitch angle. In general, the reference point for the twist is located where the collective pitch angle is defined. The blade twist angle is usually negative. Negative twist progressively reduces the angle of attack towards the tip, increases the stall and buffeting margins, unloads the outer blade region and increases flapping stability.

The UDF provides the means of displaying the effective pitch angle over the entire rotor as one of several User-Defined data fields that can be added to the solution file at your discretion. The effective pitch angle is defined as

$$\theta_e(\psi,\tau,r)=\theta_0-\theta_{lc}\cos\psi-\theta_{ls}\sin\psi+\tau(r/R)$$

where $\tau(r/R)$) is the twist angle as a function of the normalized radius, as shown in Figure 31.3: Effective Pitch Angle as a Function of r/R and $\psi$ (p. 336).

**Figure 31.3: Effective Pitch Angle as a Function of r/R and $\psi$**




## 31.2.4. Blade Flapping

Articulated helicopter rotor blades can flap in response to the centrifugal and aerodynamic forces they endure. The flapping motion is not computed by the model, however, if the characteristics of the flapping motion is known, the model can account for the rotor coning angle and the first flapping harmonics by transforming the velocity components from the rotor disk plane to the actual tip path plane. Therefore, the baseline coning angle $\beta_0$ and the longitudinal and lateral flapping coefficients $\beta_{1c}$ and $\beta_{1s}$ determine the resultant coning angle $\beta(\psi)$) as

$$\beta(\psi)=\beta_0-\beta_{lc}\cos\psi-\beta_{ls}\sin\psi$$

Figure 31.4: Blade Flapping – Baseline $\beta_0$ (Top); Longitudinal and Lateral Components $\beta_{1c}$, $\beta_{1s}$ (Bottom) (p. 337) shows a schematic of the flapping blades in the *x-z* and *y-z* planes.

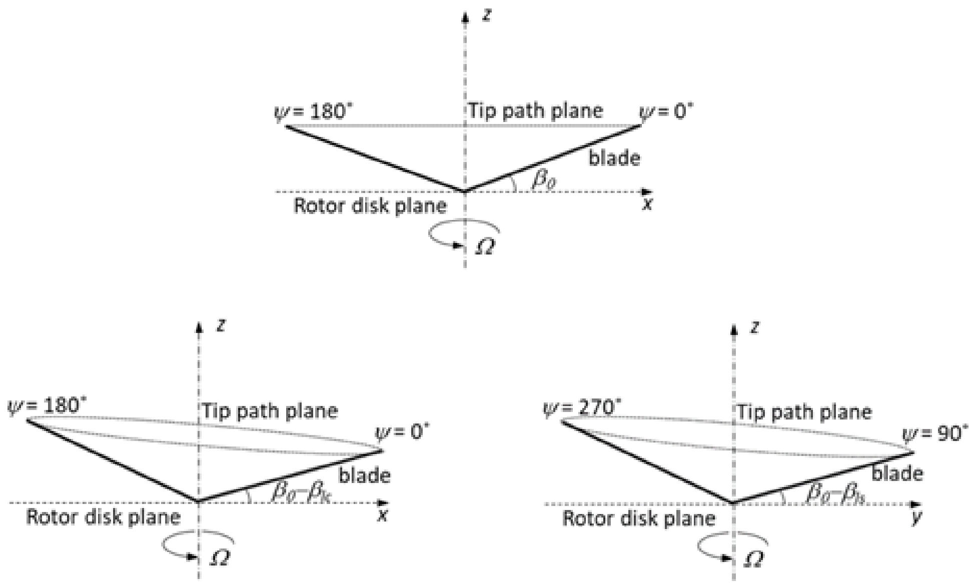
---

**Note:**

The disk of the rotor remains flat and the coning of the blades is simulated.

---

**Figure 31.4: Blade Flapping – Baseline $\beta_0$ (Top); Longitudinal and Lateral Components $\beta_{1c}$, $\beta_{1s}$ (Bottom)**




The change in the angle of attack due to flapping is computed, however the relatively small flapping velocity component $\partial\beta / \partial t$ that should be added to the velocity component normal to the blade path is neglected. Furthermore, due to its relatively small effect, the lead-lag motion of the blade is also neglected.

## 31.2.5. Rotor Trimming

The rotors of helicopters operating in forward flight generate asymmetrical load distributions with respect to the plane defined by the forward velocity vector and the rotor normal vector. In addition to the collective pitch control required to maintain level flight at the desired altitude, cyclic pitch is necessary to balance the load across the rotor and control the pitching and rolling moments. Accurate aerodynamic simulations are only possible when the rotors are operating at the desired thrust and pitching/rolling moment settings. These settings, however, must be transformed into rotor blade pitch angles - a non-trivial operation. An automatic trim routine is available to compute the correct rotor settings without costly empirical guesswork. Thrust and moments with respect to the center of the rotor can be formulated in terms of non-dimensional thrust and moment coefficients.

#### 31.2.5.1. Rotorcraft

$$T = C_T \rho_\infty V_{tip}{}^2 A_{rotor}$$

$$M_x = C_{Mx} \rho_\infty V_{tip}{}^2 A_{rotor} \frac{d_{rotor}}{2}$$

$$M_y = C_{My} \rho_\infty V_{tip}{}^2 A_{rotor} \frac{d_{rotor}}{2}$$

$$V_{tip} = \Omega \frac{d_{rotor}}{2}$$

$$A_{rotor} = \pi \frac{d_{rotor}{}^2}{4}$$

#### 31.2.5.2. Propellers

$$T = C_T \rho_\infty V_a^2 A_{prop}$$

$$M_x = C_{Mx} \rho_\infty V_a^2 A_{prop} \frac{d_{prop}}{2}$$

$$M_y = C_{My} \rho_\infty V_a^2 A_{prop} \frac{d_{prop}}{2}$$

$$A_{rotor} = \pi \frac{d_{prop}{}^2}{4}$$

where $V_{tip}$ is the rotor tip speed and $V_a$ is the aircraft forward speed.

Since rotorcrafts in hover have no forward velocity, the dynamic pressure must be formulated in terms of the rotor tip speed, whereas for propellers the aircraft forward speed is normally used.

**Important:**

In the add-on, the thrust and moment coefficients are defined as functions of the rotor tip speed for both propellers and rotors.

The reference area of the rotor $A_{rotor}$ is defined solely by its diameter *d* and includes the disc cutout, or the spinner cutout in the case of a propeller.

The Virtual Blade Model trimming routine calculates the collective and cyclic pitch angles to achieve the desired thrust and pitching (y-axis) and rolling (x-axis) moments around the hub. Let the three coefficients be expressed as functions of the collective and cyclic angles:

$$c_T = c_T\left(\theta_0, \theta_{lc}, \theta_{ls}\right)$$
$$c_{Mx} = c_{Mx}\left(\theta_0, \theta_{lc}, \theta_{ls}\right)$$
$$c_{My} = c_{My}\left(\theta_0, \theta_{lc}, \theta_{ls}\right)$$

**Note:**

The effect of the flapping hinge offset is neglected in the calculation of the moments.

Since the relationship between the rotor aerodynamic parameters and the blade pitch is non-linear, an iterative technique is used to drive the trim procedure to convergence. Following Yang et al. (References (p. 361)), a Newton-Raphson iterative method is employed to automatically trim any number of rotors in the computational domain to achieve the user-set target values $\left\{c_T, c_{Mx}, c_{My}\right\}_i^{\text{target}}$.

The simulation begins with a user-supplied initial guess for the angles $\left\{\theta_0, \theta_{lc}, \theta_{ls}\right\}_i^0$ of each rotor $i$. After every $n$ airflow iterations, the flow solver pauses and the UDF computes the rotor parameters $\left\{c_T, c_{Mx}, c_{My}\right\}_i^n$. A new set of angles is computed for each rotor *i* with the Newton-Raphson iterative approach. Let

$$\begin{Bmatrix} c_T \\ c_{Mx} \\ c_{My} \end{Bmatrix}_i^n + \begin{bmatrix} \frac{\partial c_T}{\partial \theta_0} & \frac{\partial c_T}{\partial \theta_{lc}} & \frac{\partial c_T}{\partial \theta_{ls}} \\ \frac{\partial c_{Mx}}{\partial \theta_0} & \frac{\partial c_{Mx}}{\partial \theta_{lc}} & \frac{\partial c_{Mx}}{\partial \theta_{ls}} \\ \frac{\partial c_{My}}{\partial \theta_0} & \frac{\partial c_{My}}{\partial \theta_{lc}} & v\frac{\partial c_{My}}{\partial \theta_{ls}} \end{bmatrix}_i^n \begin{Bmatrix} \Delta\theta_0 \\ \Delta\theta_{lc} \\ \Delta\theta_{ls} \end{Bmatrix}_i^n = \begin{Bmatrix} c_T \\ c_{Mx} \\ c_{My} \end{Bmatrix}_i^{\text{target}}$$

The system of equations can be solved easily by recasting it in the form

$$\begin{Bmatrix} \Delta\theta_0 \\ \Delta\theta_{lc} \\ \Delta\theta_{ls} \end{Bmatrix}_i^n = \sigma_u^{-1}\left[J_i^n\right]^{-1} \begin{Bmatrix} \Delta c_T \\ \Delta c_{Mx} \\ \Delta c_{My} \end{Bmatrix}_i^n$$

With initial angles $\left\{\theta_0, \theta_{lc}, \theta_{ls}\right\}^0$ The angles are updated as follows

$$\begin{Bmatrix} \theta_0 \\ \theta_{lc} \\ \theta_{ls} \end{Bmatrix}^{n+1} = \begin{Bmatrix} \theta_0 \\ \theta_{lc} \\ \theta_{ls} \end{Bmatrix}^n + \sigma_u \begin{Bmatrix} \Delta\theta_0 \\ \Delta\theta_{lc} \\ \Delta\theta_{ls} \end{Bmatrix}^n$$

where $\sigma_u$ is a tunable under-relaxation factor, such that $0 \le \sigma_u \le 1$. The recommended value is 0.8.

The UDF then recomputes the source terms and the flow solution resumes. Since there are no analytical equations for computing the coefficients of the Jacobian matrix, the derivatives are obtained in the frozen flow field at time level $n$ by computing the thrust and moment coefficients after perturbing each angle independently by the amount $\pm a$. For example, the first coefficient, representing the derivative of the thrust coefficient with respect to the collective angle, would be approximated by

$$\frac{\partial c_T}{\partial \theta_0} \approx \frac{c_T(\theta_0+\alpha)-c_T(\theta_0-\alpha)}{(\theta_0+\alpha)-(\theta_0-\alpha)}$$

$$\approx \frac{c_T(\theta_0+\alpha)-c_T(\theta_0-\alpha)}{2\alpha}$$

Since the derivatives are computed numerically, the algorithm is really the Secant Method. This procedure is repeated at every n iterations, until the rotor performance parameters reach their target values and the flow solution is converged. The L2 norms of $\Delta C_T$, $\Delta C_{Mx}$ and $\Delta C_{My}$ are computed and printed on the Fluent Console to show the convergence of the trimming routine. The trimming routine is robust and yields converged results for cases simulating multiple rotors.

## 31.2.6. Tip Losses

Rotor blades usually have very high aspect ratios, therefore the lift and drag forces can be computed to a reasonable approximation at each spanwise location by assuming two-dimensional flow. However, near the tip, the circulation around the blade induced by the lift force is turned by 90˚ and shed downstream. The turning of the circulation creates the tip vortex, which introduces lift and drag losses that the VBM is unable to model directly.

To obtain a more realistic solution, the tip loss effects are incorporated in VBM using a tip loss factor $F$ that corrects lift and drag coefficients near the tip of the disk, such that:

$$C_{lcor}=F^{*}C_l(Re,Ma,\alpha\ )$$

where the lift coefficient $C_l$ is interpolated in airfoil data tables based on the $Re,Ma,\alpha$ at the cell centroid of each element on the disk. Corrected drag coefficient $C_{dcor}$ is interpolated in airfoil data tables based on the $Re,Ma$ at the cell centroid and the corrected angle of attack $\alpha_{cor}$ that corresponds to the corrected lift coefficient $C_{lcor}$; this angle of attack is also interpolated in airfoil data tables.

$$C_{lcor}\overset{table}{\Longrightarrow}\alpha_{cor}(Re,Ma,C_{lcor})\ \overset{table}{\Longrightarrow}C_{dcor}(Re,Ma,\ \alpha_{cor})$$

Two options are available in VBM to calculate the tip loss factor as follows:

- **Quadratic Tip Loss Function**: Using this option, tip losses are approximated by a damping function inversely proportional to the square of the distance to the blade tip, starting from a user-specified radial threshold, named tip loss limit $r_{te}$, by default set at 96% of the rotor span ($R$).

- **Modified Prandtl's Tip Loss Function**: Prandtl's tip loss function is generally used in Blade Element Momentum Theory (BEMT) that equates two set of equations of the momentum balance on a rotating annular stream tube passing on a disk (momentum theory, or MT) and the forces acting on the blade element at the corresponding sections along the blade span (blade element theory or BET). In VBM, however, Reynolds-averaged Navier–Stokes equations (RANS) and BET are applied such that RANS can capture the tip loss effects near the tip of rotor disk; however, that does not represent the tip loss near the tip of the actual rotor. Therefore, a modified version of this function is used in VBM such that the user can tune it for each application. In VBM, the tip loss factor $F$ is calculated using:

$$F=\frac{2}{\pi}\cos^{-1}\left(e^{-gf}\right) \;\;\&\;\; f=\frac{N}{2}\frac{R-r}{r sin(\varphi)}$$

where $g$ is the tuning coefficient and $f$ is given in terms of the number of the blades $N$, the radial position of blade element $r$, the rotor radius $R$, and induced inflow angle $\varphi$, defined as the angle between the normal and tangential components of relative incident velocity at each blade section where an element exists.

$$\varphi=\tan^{-1}\left(\frac{V_n}{V_t}\right)$$

The tuning coefficient $g$ should be larger than 1, set by user in VBM user interface.

## 31.3. Output Data

The VBM add-on prints useful data on the console, the `Rotor_xx_Loads.csv` files and optionally in the solution file. The data printed to the console after each rotor trimming update are the collective and longitudinal and lateral components of the cyclic angles and their changes from the previous update. The L2 residuals of the three equations are also printed to facilitate the assessment of the convergence of $C_T$, $C_{Mx}$ and $C_{My}$.

The `Rotor_xx_Loads.csv` files are written for each active rotor `xx` and contain the `Thrust(N)`, `Torque(N·m)`, `Power(W)`, `Mx(N·m)` and `My(N·m)` components of the moments around the rotor center for each *Iteration* or *Elapsed Time* step, depending on whether the simulation is steady-state or unsteady. The file is printed in comma-separated format (CSV) and can be imported directly into virtually any spreadsheet application.

The VBM add-on allocates 15 additional User Defined Memory (UDM) units for each computational cell in the entire domain to store the results of the VBM. The UDM variables are cell-based and will not display correctly as node variables. They are calculated only in the cells attached to the actuator disks. The labels of the UDM parameters are listed in Table 31.1: UDM Variables (p. 341).

**Table 31.1: UDM Variables**

| Parameter | Name | Description |
|---|---|---|
| 0 | `radius` | Radius of the cell centroid |
| 1 | `\|V\|_blade` | Relative airflow velocity at the cell centroid |
| 2 | `pitch` | Pitch angle as defined in Blade Pitch (p. 335) |
| 3 | `chord` | Blade chord at the cell centroid |
| 4 | `Re no.` | Reynolds number |
| 5 | `Ma no.` | Mach number |
| 6 | `AoA` | Angle of attack |

| Parameter | Name | Description |
|---|---|---|
| 7 | `C_l` | Lift coefficient of blade section at the cell centroid |
| 8 | `C_d` | Drag coefficient of the blade section at the cell centroid |
| 9 | `Fn_blade` | Normal component of the force acting on the blade at the cell centroid |
| 10 | `Ft_blade` | Tangential component of the force acting on the blade at the cell centroid |
| 11 | `Fr_blade` | Radial component of the force acting on the blade at the cell centroid |
| 12 | `Fx` | The time-averaged force component in the x-direction |
| 13 | `Fy` | The time-averaged force component in the y-direction |
| 14 | `Fz` | The time-averaged force component in the z-direction |

In case of loading any UDF library for a simulation while VBM add-on is activated, memory conflicts may arise if that UDF is writing UDMs to the data file. This VBM add-on takes advantage of the `udm_offset` variable provided by Fluent. This variable modifies the addressing of the UDM fields to prevent conflicts between libraries. The offset value is written to the Fluent **Console** after the **Load** command is issued and it should be `udm_offset` $\geq$ `0`. If there is any UDM conflict, the value could remain at its negative default value and a crash may occur as soon as the add-on or UDF attempts to write the extra data.

## 31.4. Meshing Guidelines

The VBM supports both structured and unstructured hybrid grids. An internal surface in the shape of a flat disk should be created in the flow domain at the location of the rotor. In the CAD, this surface should be marked as an internal wall or internal surface that will be discretized by a single layer of nodes, such that the mesh is continuous across the surface (for example, no duplicate nodes). The dead part of the rotor near the hub - the cutout - can be neglected. Figure 31.5: Schematic of a Cut Through a Structured VBM Grid. The Green Cells Are Assigned to the VBM (p. 343) shows the disk analogue of a simple rotor and its cutout.

A single uniform layer of prisms or hexahedral cells should be extruded from the cell faces attached to the disk in the direction normal to the disk. Although there is no definitive rule regarding the thickness of these cells, it is recommended that the layer remains between 1/40th to 1/100th of the diameter of the disk. The layer of cells in the direction downstream of the disk should be marked as a separate zone (rotor zone).

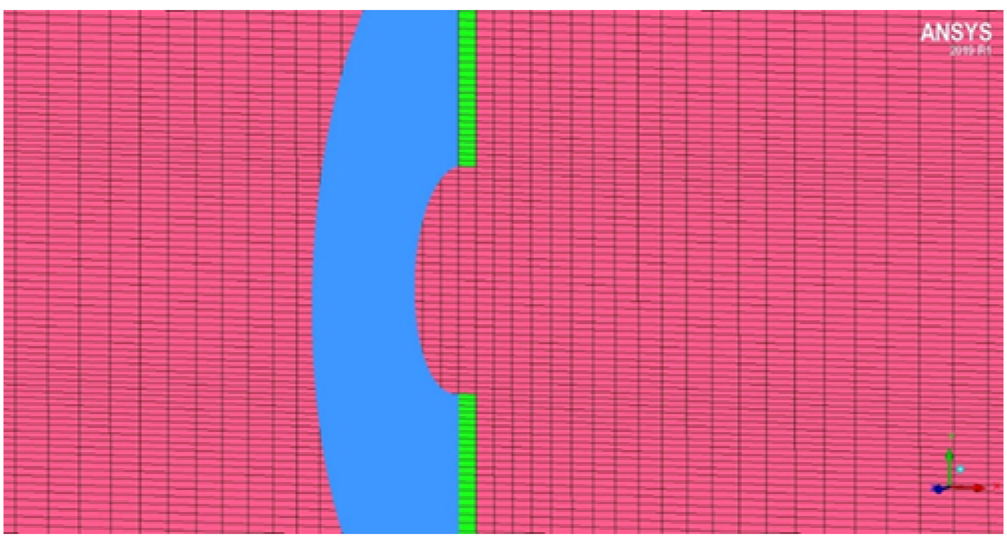
**Note:**

Pyramids and tetra cells attached to the disk cells to provide the blending with the rest of the grid are part of the flow domain, not the rotor zone.

Mesh clustering on the inner and outer circumference of the disk is recommended since these are regions where interference effects and loss models are usually acting. The current simple tip loss model, for example, will not produce smooth solutions if the grid in the range $0.96 \leq r/R \leq 1.0$ is very coarse.

Attention must be paid to the size of the inlet/exit boundaries of the computational domain, particularly for heavy helicopters in hover, to ensure that the mass flow rate of air passing through the domain is many times larger than the mass flow through all the rotors. If this detail is overlooked the overall simulation might diverge, converge very poorly, or the rotors may not be able to produce the desired amount of thrust. The reason is that VBM simulations (and also higher-fidelity methods, such as the MRF) are over-specified: while rotorcraft or propeller-driven aircraft will fly at a speed resulting from the balance of forces acting on them, in the CFD simulation this speed is specified by you. It is then possible to specify speeds that are incompatible with the settings of the rotor. For this reason, for example, a suitable non-zero mass flow rate should be specified on the farfield boundaries of a helicopter in hover.

**Figure 31.5: Schematic of a Cut Through a Structured VBM Grid. The Green Cells Are Assigned to the VBM**




Finally, the grid in the downstream of the rotor should have sufficient clustering to capture the complex effects of the momentum change imparted by the VBM and the interaction of the wake of the rotor with other rotors (for example, main and tail rotors of a helicopter) or with other components of the aircraft/rotorcraft. Since the load distribution of a rotor is generally non-uniform, secondary vortices are likely to appear and should be captured in as much detail as possible.

## 31.5. Airfoil File Format

The results of the VBM will only be as accurate as the lift and drag data supplied for each blade section. The blade is represented by a stack of airfoils and therefore multiple airfoil files can be used. Rotating components can have high tip velocities, therefore, if possible, the airfoil data file should contain lift and drag data for a range of Reynolds and Mach numbers sufficient to cover the entire spectrum of aerodynamic conditions that will be encountered. Since the receding blades of helicopters in forward-flight can experience reversed flow, the data must be provided for the full $-180^{\circ} \leq \text{AoA} \leq +180^{\circ}$ spectrum of angles of attack. The names of the airfoil data files present in the working directory should always have the `.dat` suffix, as in `naca0012.dat`. The airfoil file follows the (C81-compatible) format listed in Table 31.2: Format of the Airfoil Data File (p. 344).

**Table 31.2: Format of the Airfoil Data File**

| | |
|---|---|
| `Airfoilxyz` | Name of the airfoil (same as the airfoil file name and 30 characters maximum) |
| `Itot` | Total number of $C_L$ and $C_D$ tables in the file ( 25 airfoil tables maximum (cl plus cd)) |
| `Cl` | Label at the beginning of the $C_L$ table. (10 characters maximum) |
| `Re` | Reynolds number of the table |
| `Ma` | Mach number of the table |
| `Jtot` | Number of lines in the subsequent table |
| `-180.0 0.00` | Angle of attack and $C_L$ values (two blanks-separated columns and 250 data points maximum) |
| ... | |
| `-45.0 -1.50` | |
| ... | |
| `0.0 0.0` | |
| ... | |
| `45.0 1.50` | |
| ... | |
| `180.0 0.00` | |
| `Cd` | Label at the beginning of the $C_D$ table. (10 characters maximum) |
| `Re` | Reynolds number of the table |
| `Ma` | Mach number of the table |
| `Jtot` | Number of lines in the subsequent table |
| `-180.0 0.006` | Angle of attack and $C_D$ values (two blanks-separated columns and 250 data points maximum) |
| ... | |
| `180.0 0.006` | |

| | |
|---|---|
| ... | Repeat, if more $C_L$ and $C_D$ tables are available. |

The following shows an example of an airfoil data file.

```
naca0015
2
cl
100000.0
0.1
 41
     -180.00  0.0000
     -172.00  0.7800
     :
        0.00  0.0000
     :
      172.50 -0.7800
      180.00  0.0000
cd
100000.0
0.1
71
     -180.00 0.0220
     -175.00 0.0620
     :
        0.00 0.0088
      :
      175.00 0.0620
      180.00 0.0220
```

## 31.6. Loading the VBM Add-on Module

The VBM add-on module is a beta feature found under the `addons/vbm` directory. Therefore, it is required to enable the feature through the text user interface (TUI) using the following command in the Fluent console:

**`(enable-feature 'vbm)`**

VBM add-on module can now be loaded through TUI.

> **Note:**
>
> The module can be loaded only when a valid Ansys Fluent case file or mesh has been set or read.

To load the add-on, use the following TUI command in the Fluent console:

**`define`** → **`models`** → **`addon-module`**

A list of Ansys Fluent add-on modules is displayed in the console. Select the Virtual Blade Model by entering **`12`** as the module number.

```
> define/models/addon-module
Fluent Addon Modules:
     0. None
     1. MHD Model
     2. Fiber Model
     3. Fuel Cell and Electrolysis Model
     4. SOFC Model with Unresolved Electrolyte
     5. Population Balance Model
     6. Adjoint Solver
     7. Single-Potential Battery Model
     8. Dual-Potential MSMD Battery Model
     9. PEM Fuel Cell Model
     10. Macroscopic Particle Model
     11. Reduced Order Model
     12. Virtual Blade Model
Enter Module Number: [0] 12
Fast-loading "C:/PROGRA~1/ANSYSI~1/v211/fluent/fluent21.1.0/addons/vbm\lib\addon.bin"
Preset all vbm model specific UDF hooks? [no] yes
Done.
```

Next, the add-on will prompt you to setup UDF hooks. Answering `yes` will setup the UDM to 15 and set the Function Hooks. During the loading process, a Scheme library containing the graphical and text user interface, and a UDF library containing a set of user-defined functions (UDFs) for the VBM add-on module are automatically loaded into Ansys Fluent.

This process is reported in the Fluent console and UDM offset should be equal to 0 at the end. Once the add-on is loaded into Ansys Fluent, the **Virtual Blade Model** appears under the **Models** tree branch.

**Note:**

The VBM is only available in 3D and Parallel mode. The **Serial** option is deprecated.

## 31.7. Cell Zone Condition

After loading the add-on, source terms should be assigned for each rotor zone:

1. Click the **Physics** button within the main ribbon. Click the **Cell Zones** button in the **Zones** column on the right-hand side of the main ribbon.

2. Select the rotor zone and click on the **Edit** button at the bottom.

3. Tick mark the **Source Terms** radio button.

4. Click the **Source Terms** button that has become active on the right-hand side of the ribbon.

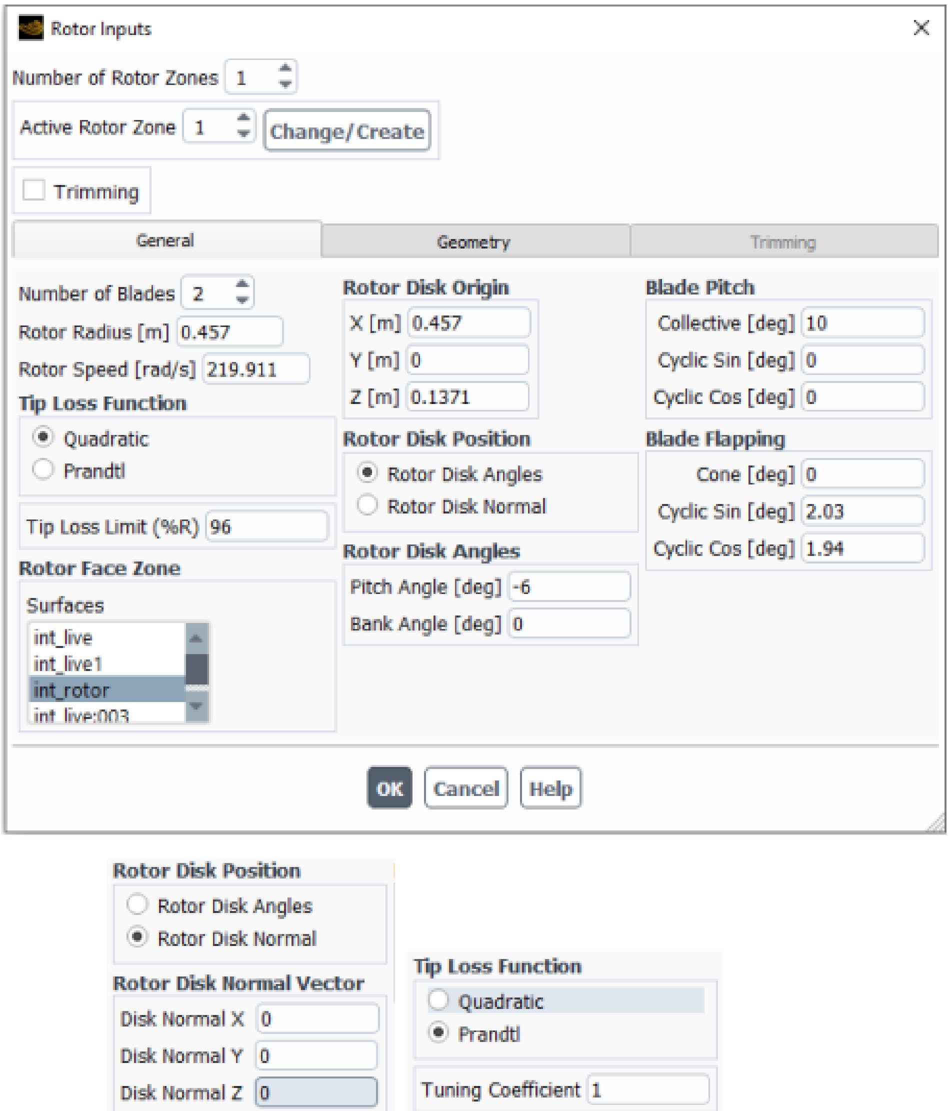
5. For each of the X, Y and Z Momentum, click the **Edit** button. Enter **1** for the number of sources in the small window at the right, and select the corresponding source term. For example, for each of the **X Momentum** sources, click on the down-arrow and select **udf xmom_srcN::vbm** appropriate for the respective rotor number from the menu.

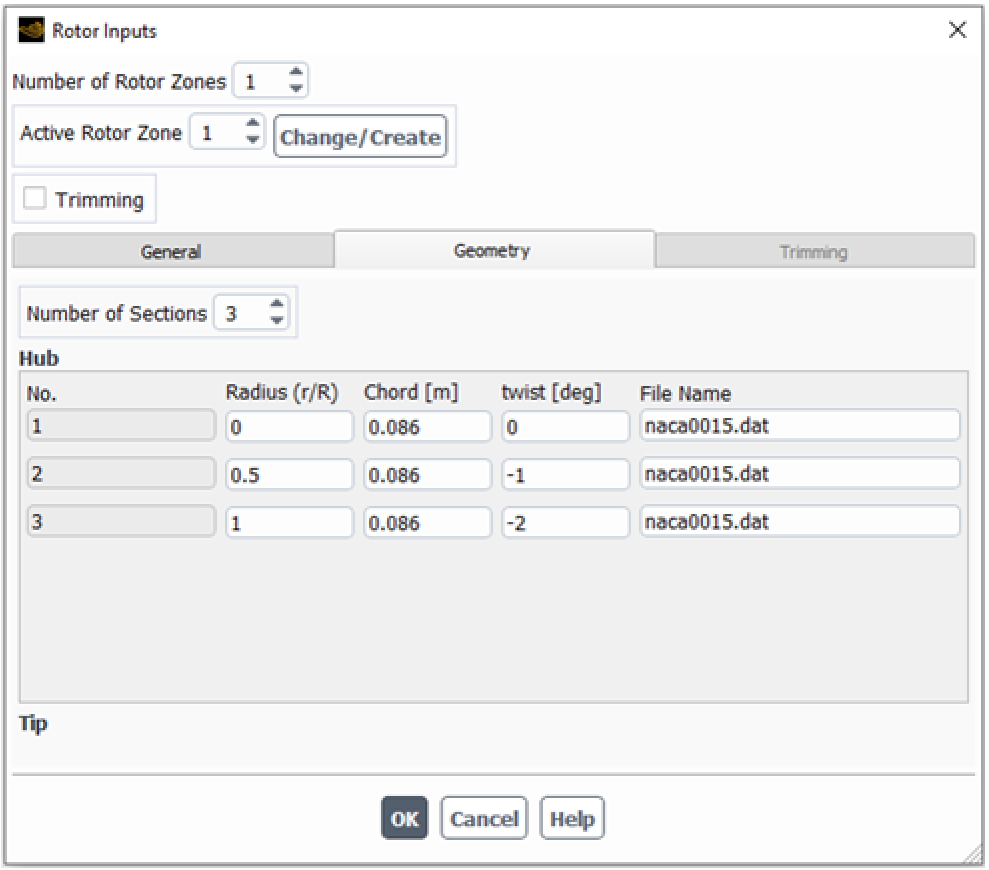
6. Click the **Apply** button and close the **Fluid** dialog box.

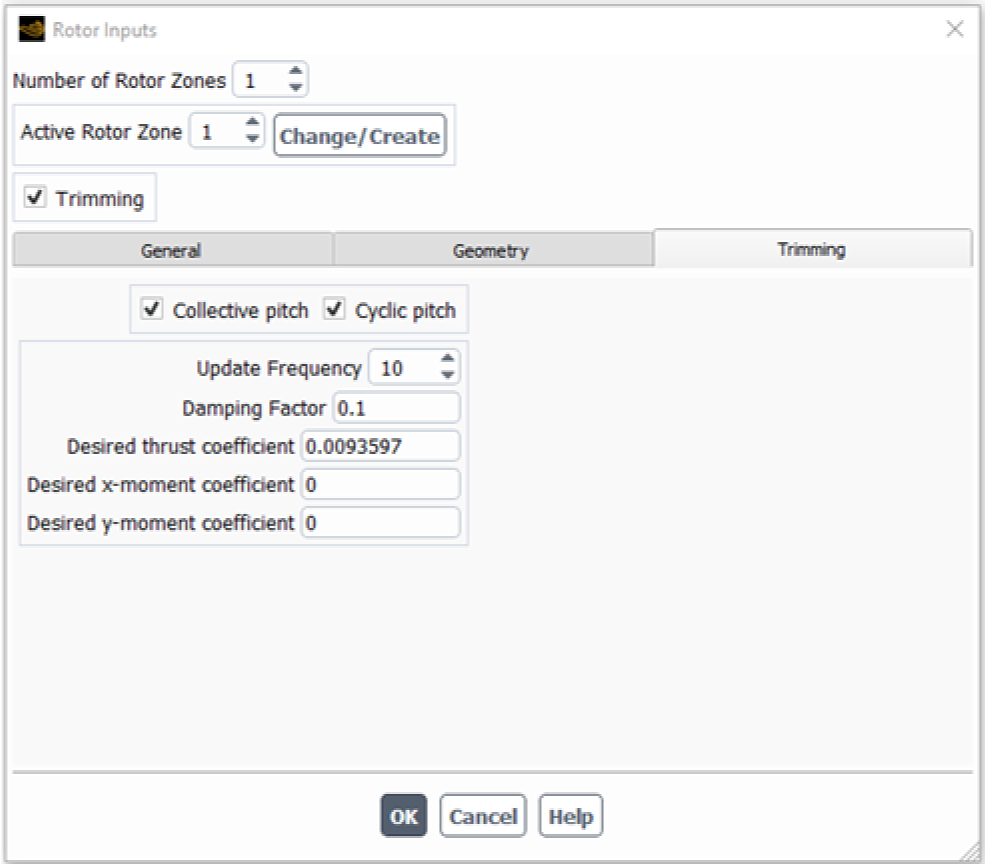
## 31.8. VBM Configuration

The add-on module can be configured through its own mini-graphical user interface. Depending on the running conditions, the mini-graphical user interface will display up to three configuration panels, accessible through the ribbon: General, Geometry and Trimming, as shown in Figure 31.6: UDF Mini-Graphical User Interface Configuration Panels (p. 347).

**Figure 31.6: UDF Mini-Graphical User Interface Configuration Panels**




Rotor Inputs
Number of Rotor Zones 1
Active Rotor Zone 1
Change/Create
Trimming
General
Geometry
Trimming
Number of Sections 3
Hub
No.
Radius (r/R)
Chord [m]
twist [deg]
File Name
1
0
0.086
0
naca0015.dat
2
0.5
0.086
-1
naca0015.dat
3
1
0.086
-2
naca0015.dat
Tip
OK
Cancel
Help




**Note:**

The input to the add-on is always in SI units.

To configure the rotors contained in a grid, follow these steps:

1. Enter the **Number of Rotor Zones** present in the grid. The UDF is configured by default for 10 rotor zones, but it can support up to 25.

2. Select the **Active Rotor Zone** to be configured.

3. Checkmark the **Trimming** box if collective/cyclic trimming is desired.

4. Enter the **Number of Blades**.

5. Enter the **Rotor Radius (m)**.

6. Enter the **Rotor Speed (rad/s)** (rpm or rad/s, depending on the Fluent units configuration).

7. Select **Tip Loss Function**; enter the **Tip Loss Limit** (% of normalized rotor radius, default is 96%. A value of 100% means that tip losses will not be introduced) if choosing **Quadratic** function; enter **Tuning Coefficient** if using **Prandtl**'s function.

8. Enter **X**, **Y**, **Z** coordinates of the center of the rotor.

9. Select the internal disk surface representing the rotor from the **Surfaces** list.

10. To set **Rotor Disk Position**, enter the **Pitch and Bank Angle [deg]** if choosing the **Rotor Disk Angles** option (see Figure 31.1: Rotor Disks Schematic (p. 334)) or enter **Disk Normal X, Y, Z** if choosing the **Rotor Disk Normal** option.

11. Enter the initial blade **Collective (deg)** angle $\theta_0$.

12. Enter the **Cyclic Sin (deg)** (lateral) and **Cyclic Cos (deg)** (longitudinal) components $\theta_{lc}$ and $\theta_{ls}$ of the **Blade Pitch** angle.

13. Enter the **Cone (deg)** angle $\beta_0$.

14. Enter the **Coning Sin (deg)** and **Coning Cos (deg)** components $\beta_{1c}$ and $\beta_{1s}$ of the **Blade Flapping**.

15. Click on the **Geometry** button on the ribbon.

16. Enter the **Number of Sections**. Up to 25 are supported by default.

17. For each section, enter the **Radius (r/R)**, starting at innermost location (cutout), **Chord (m)** length, **twist (deg)** angle and **File Name** of the airfoil file containing the CL and CD data for the section (with or without the `.dat` suffix).

---

**Note:**

All files containing the airfoil tables must have `.dat` suffix. If omitted from the airfoil file name entered in the module mini-graphical user interface, it will be added automatically. Whether the name **`naca0012`** or **`naca0012.dat`** is entered into the module mini-graphical user interface, a file named `naca0012.dat` must exist in the project directory.

---

18. Click the **Trimming** tab on the ribbon, if trimming has been activated.

19. Tick-mark the **Collective pitch** and/or the **Cyclic pitch** boxes. Depending on the trimming operation selected, additional input boxes will be activated.

20. Select the **Damping Factor**. Default value is 0.1, but faster convergence can be achieved by increasing this value. Max value is 1, recommended value is 0.8.

21. Enter values for the **Desired thrust coefficient**.

22. The **Desired x-moment** and **Desired y-moment** coefficients are normally left at 0, however non-zero values can be entered if a longitudinal or lateral moment is required.

23. Click the **Change/Create** button to save the data for this rotor.

24. Repeat steps 2 – 23 to configure the other rotor zones.

25. Click the **OK** button only to terminate input and close the mini-graphical user interface.

26. Save the `.cas` file.

After the initial configuration, every time a change is made to one or more of the parameters, the **Change/Create** button must be clicked before switching to another **Active Rotor Zone** or clicking on the **OK** button. Clicking on the **Change/Create** button saves the data of each rotor to a pointer, whereas clicking on the **OK** button stores the data in a Fluent memory pointer and closes the mini-graphical user interface. When the `.cas` file is saved, the information of all the rotors will be saved as a pointer in the file.

# 31.9. Validation Examples

## 31.9.1. Propeller

This validation case consists of a 6-blade propeller. The 3D geometry of the propeller was simulated with Ansys CFX for a range of advance ratios $1.0 \leq J \leq 1.7$ and a fixed inflow velocity $V = 84.5452$ m/s. The rotational speed varies from 5,073 at the lowest to 2,984 rpm at the maximum advance ratio. The design advance ratio is $Jd = 1.4208$, which yields a rotational speed $W = 3570.32$ rpm (373.8831 rad/s). The blade pitch and blade chord distributions are mildly non-linear and the NACA 16016 airfoil section used for the blades has a thickness of 6%. The details of the rotor and running conditions are listed in Table 31.3: Propeller Characteristics (p. 351).

**Figure 31.7: Six-Blade Propeller Geometry for the 3D CFX Simulation**

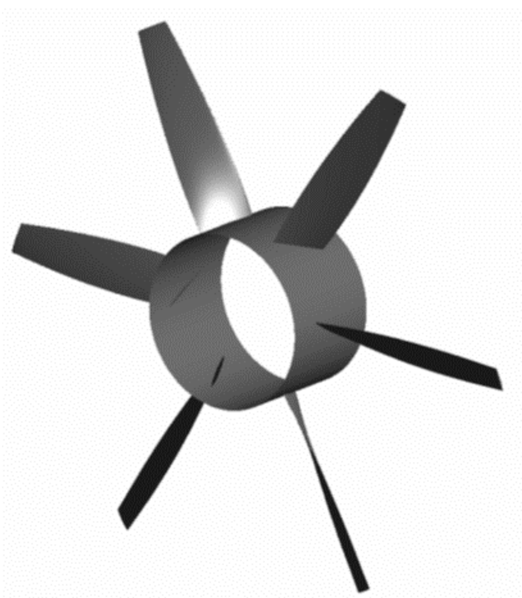


**Table 31.3: Propeller Characteristics**

| **Geometric Characteristics** | |
|---|---|
| Propeller diameter | 1.0 m |
| Cutout diameter | 0.3 m |
| Blade root angle | 61.85345° |
| Blade tip angle | 26.72414° |
| Root chord | 0.10 m |
| Tip chord | 0.05 m |
| Blade airfoil | NACA 16016 |

| Operating Conditions | |
|---|---|
| Mach number | 0.25 |
| Forward speed | 84.5452 m/s |
| Static pressure | 97,013.91 Pa |
| Static temperature | 284.5926 K |

Test conditions:

1. Compute the performance of the propeller with fixed-pitch blades over the range of advance ratios.

2. At the design advance ratio, turn the problem around and use the computed thrust to trim the now variable-pitch blades and recover the fixed-pitch blade angle.

Figure 31.8: Computational Domain of the VBM Propeller Simulation. The Rotor Is the Small Green Dot at the Center (p. 352) shows the computational domain built for this VBM simulation, with a diameter 50 times larger than the actuator disk. The domain was discretized with a block-structured grid composed of 2.15 M nodes and 2.17 M cells.

**Figure 31.8: Computational Domain of the VBM Propeller Simulation. The Rotor Is the Small Green Dot at the Center**

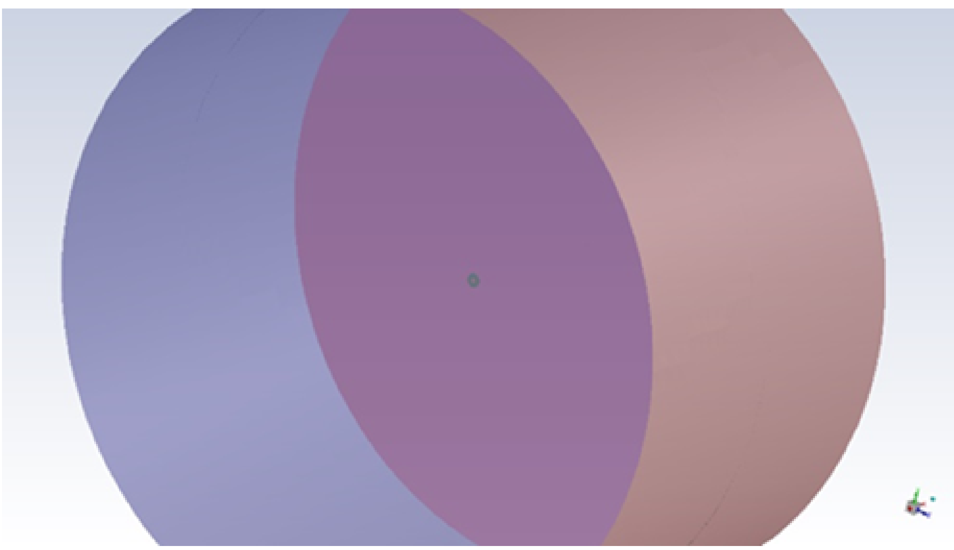
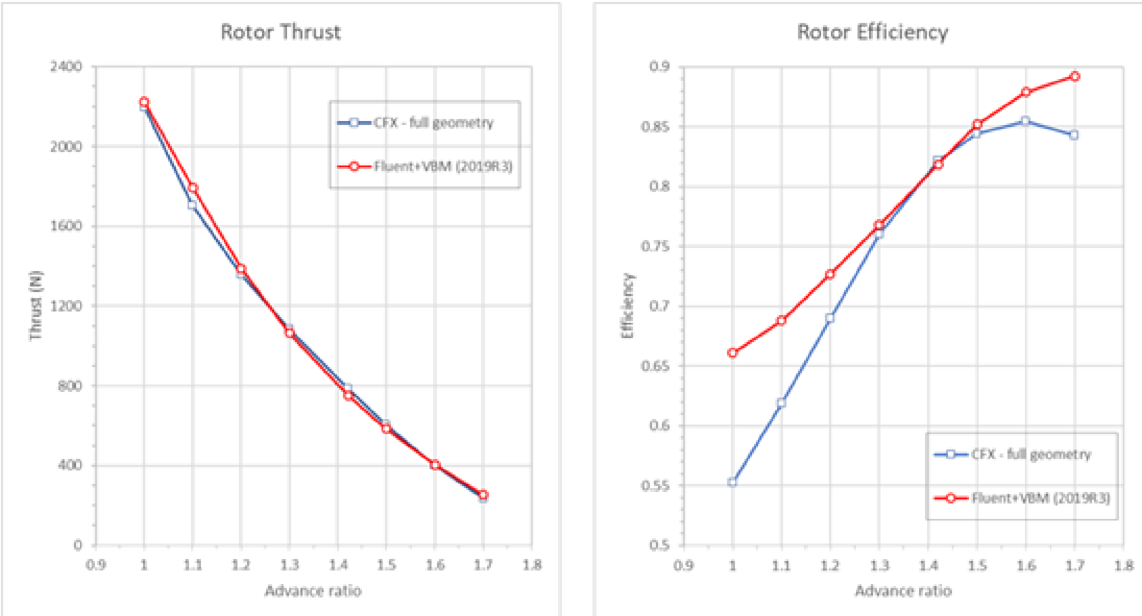


Figure 31.9: VBM Performance Compared to the 3D Propeller (p. 353) shows the results of the simulations for the range of advance ratios. The thrust produced by the VBM is in good agreement with the results of CFX, particularly at the propeller design advance ratio $Jd$ = 1.4208. The efficiency curve, however, indicates that the tip and hub losses are not yet sufficiently well-modeled to predict the peak efficiency of 85% at $J$ = 1.6.

**Figure 31.9: VBM Performance Compared to the 3D Propeller**




The propeller simulation has also been used to validate the blade angle (collective) trimming algorithm at the design advance ratio $Jd$ = 1.4308. At this advance ratio, the actuator disk simulation with fixed pitch produced a thrust of 751.0118 N, which corresponds to a thrust coefficient $CT$ = 0.02304. The virtual blades were allowed to change pitch from a starting angle of 55˚ to match the target thrust coefficient and recover the original tip blade angle $q$tip = 61.85345˚. Figure 31.10: Root Blade Angle Convergence History (p. 353) shows the blade angle convergence history. At convergence, the measured blade angle was $qcomp$ = 61.853327˚, a difference of less than 0.0002%. Hence, we can conclude that the trimming algorithm is accurate and converges very quickly, as shown in Figure 31.11: Thrust Coefficient Convergence History (p. 354) and Figure 31.12: Fluent Convergence with Blade Pitch Trimming Enabled (p. 354).

**Figure 31.10: Root Blade Angle Convergence History**

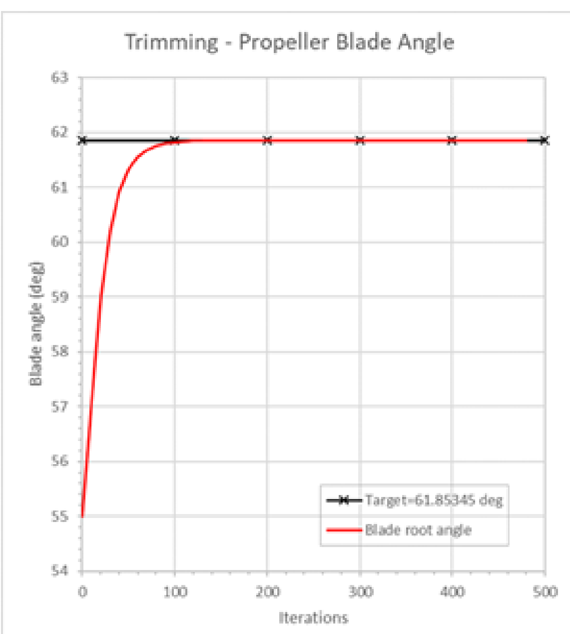


**Figure 31.11: Thrust Coefficient Convergence History**

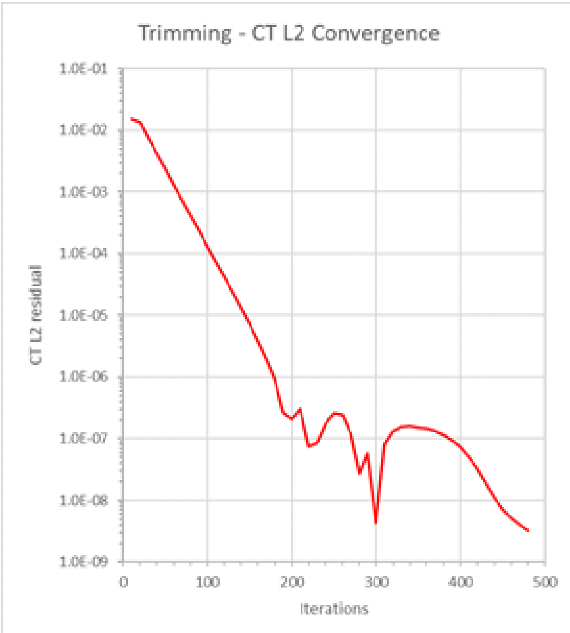
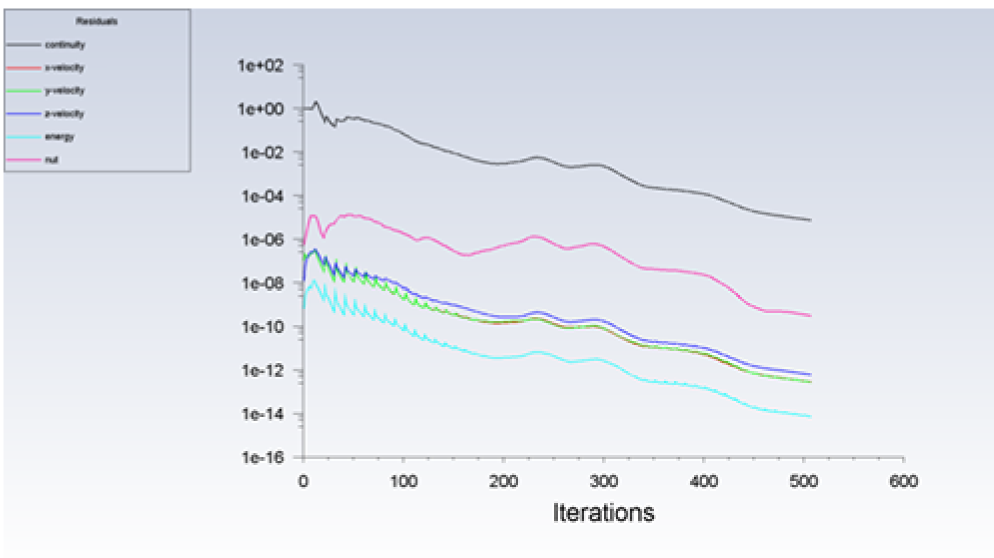



**Figure 31.12: Fluent Convergence with Blade Pitch Trimming Enabled**




## 31.9.2. Simple Helicopter

In order to validate the current implementation of the model, we consider the following rotor airframe interaction case in forward-flight. The airframe consists of a cylindrical body, 0.134m in diameter, with hemispherical nose. It was tested in the Georgia Institute of Technology 7x9 ft low-speed tunnel, shown in Figure 31.13: Simple Helicopter Geometry in the 7x9 FT Georgia Institute of Technology Low-Speed Wind Tunnel (p. 355). The rotor consists of two stiff teetering blades that are allowed to flap. The flapping angles will be imposed using the results of the experiments. The geometric and operating conditions are listed in Table 31.4: Simple Helicopter Model (p. 355).

**Figure 31.13: Simple Helicopter Geometry in the 7x9 FT Georgia Institute of Technology Low-Speed Wind Tunnel**

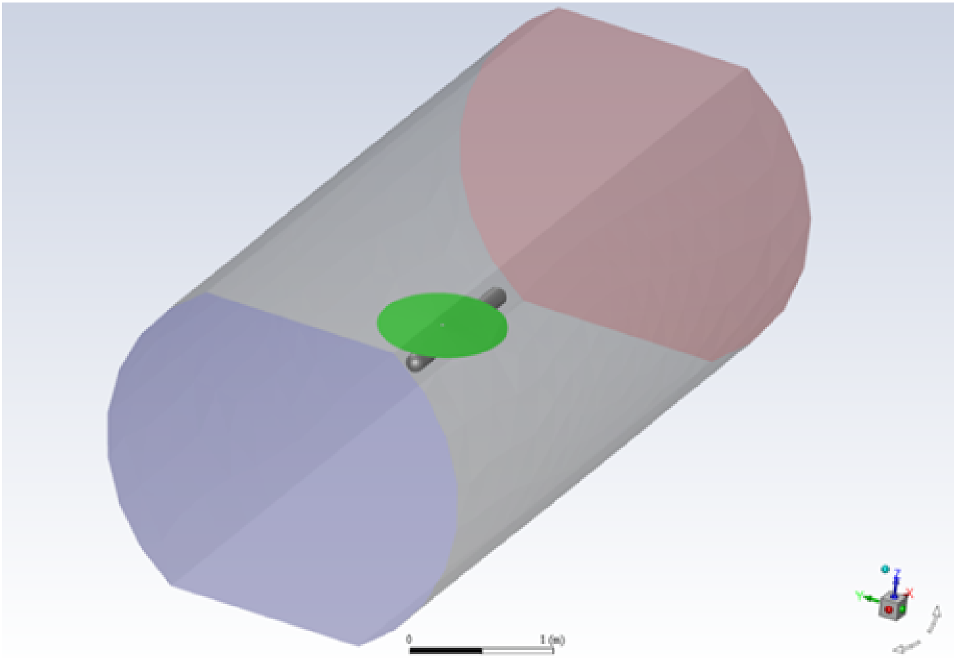



**Table 31.4: Simple Helicopter Model**

| **Geometric Characteristics** | |
|---|---|
| Rotor diameter | 0.914 m |
| Collective blade angle | 10.0˚ (fixed) |
| Blade twist | 0.0˚ |
| Root chord | 0.086 m |
| Tip chord | 0.086 m |
| Advance ratio | 0.1 |
| Thrust coefficient | 0.0092 |
| Speed | 10.04995 m/s |
| Rotor pitch angle | -6.0˚ |
| Flap angle, longitudinal | 1.94˚ |
| Flap angle, lateral | 2.03˚ |
| Rotational speed | 2,100 rpm<br>(219.9115 rad/s) |
| Blade airfoil section | NACA 0015 |

| **Operating Conditions** | |
|---|---|
| Advance ratio (*J*) | 0.1 |
| Reference pressure | 101,263.15 Pa |
| Reference temperature | 288.0997 K |

| Operating Conditions | |
|---|---|
| Reference density | 1.22451 kg/m3 |
| Entrance velocity *uref* | 10.04996 m/s |
| Tip speed | 100.49955 m/s |

This validation case uses a fully unstructured, body-fitting mesh consisting of less than 350k cells and converges in less than 500 iterations using Fluent's coupled solver. Ideal-gas conditions are assumed for the working fluid and turbulence is modeled using the Spalart-Allmaras model including viscous heating and rotation and strain tensors in the eddy viscosity production term. The PRESTO scheme is selected for the pressure discretization, the Green-Gauss Node-Based approach for gradient discretization, second-order upwind for the momentum equations and first-order upwind for the turbulence model.

Figure 31.14: Locations of the Cutting Planes Across the Fuselage (p. 356) and Figure 31.15: Locations of the Plotting Curves Long the Fuselage (p. 356) show the locations of the cutting planes and plotting curves used in the subsequent illustrations.

**Figure 31.14: Locations of the Cutting Planes Across the Fuselage**

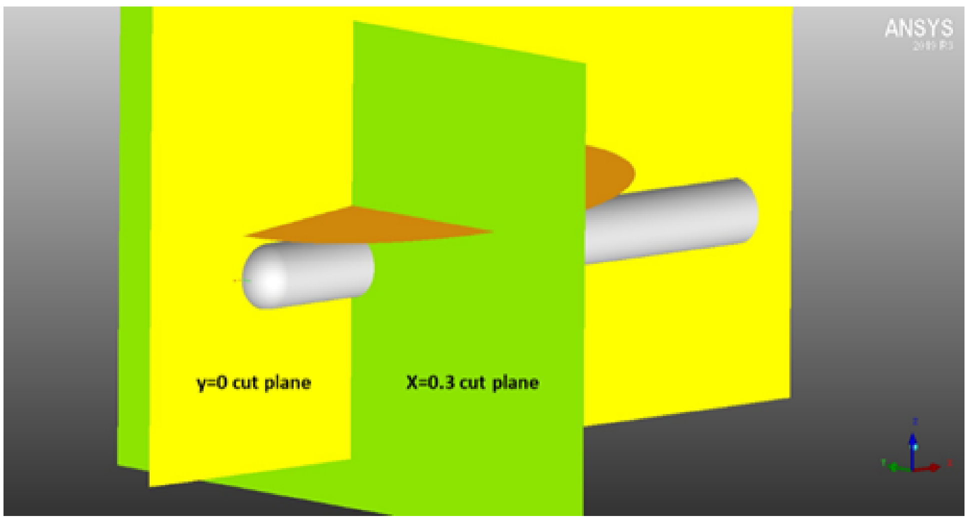
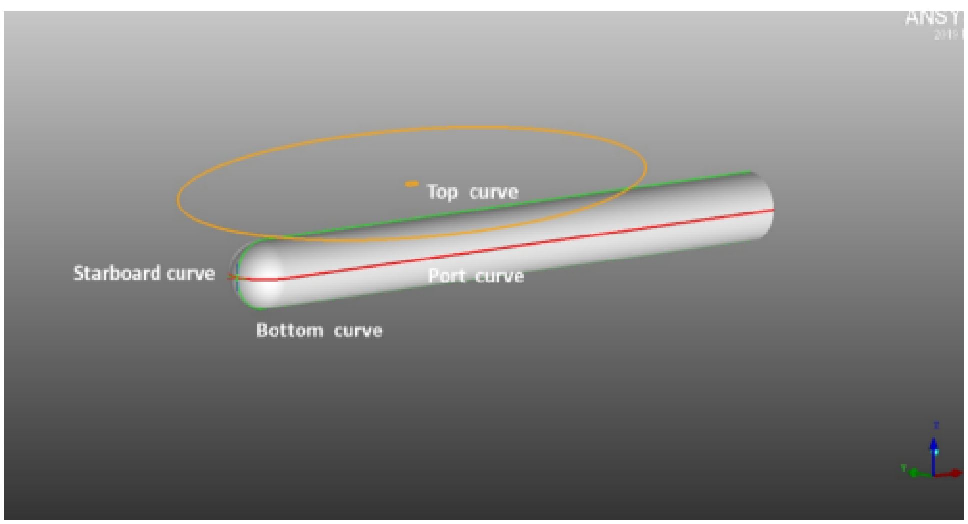



**Figure 31.15: Locations of the Plotting Curves Long the Fuselage**



Two cases are compared with the experimental data:

1. Fixed-pitch rotor blades.

2. Rotor blades trimmed for collective and cyclic angles.

Figure 31.16: Static Pressure at the X=0.3 Cutting Plane, Seen from the Front (p. 357) shows the static pressure distribution in a cross-section cutting across the rotor at x=0.3m. The fixed-pitch case, Figure 31.16: Static Pressure at the X=0.3 Cutting Plane, Seen from the Front (p. 357) (a), clearly shows a larger pressure jump and therefore larger lift on the advancing side than on the retreating side, causing an imbalance in the moments. Figure 31.17: Static Pressure at the Y=0 Cutting Plane, Seen from the Side (p. 357) (a) shows that there is a pressure imbalance on the rotor also in the longitudinal direction. This result is not representative of the experiment and yields pressure predictions on the fuselage that do not agree well with the experimentally observed values, as shown in Figure 31.18: Fixed-Pitch Helicopter Rotor Simulation (p. 359). Figure 31.16: Static Pressure at the X=0.3 Cutting Plane, Seen from the Front (p. 357) (b) and Figure 31.17: Static Pressure at the Y=0 Cutting Plane, Seen from the Side (p. 357) (b), on the other hand, show that the case with collective and cyclic trimming exhibits a more balanced pressure distribution over the rotor disk, both laterally and longitudinally.

**Figure 31.16: Static Pressure at the X=0.3 Cutting Plane, Seen from the Front**

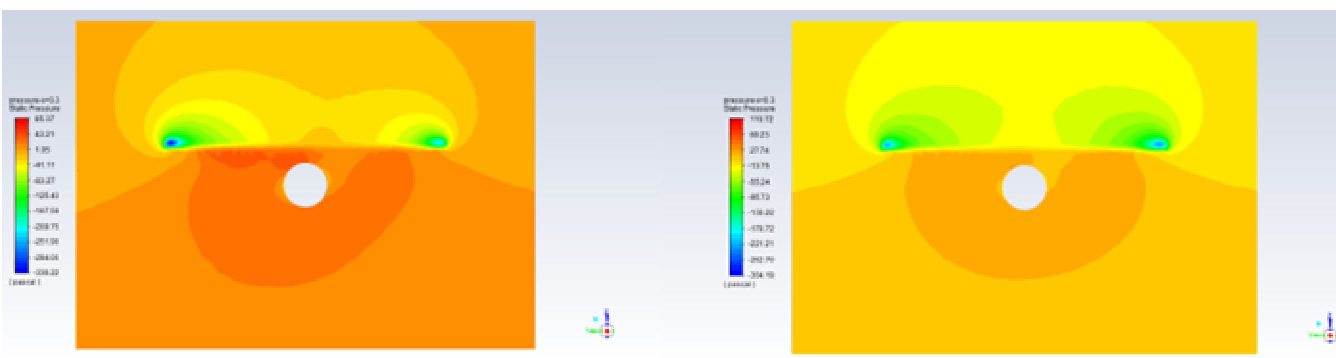
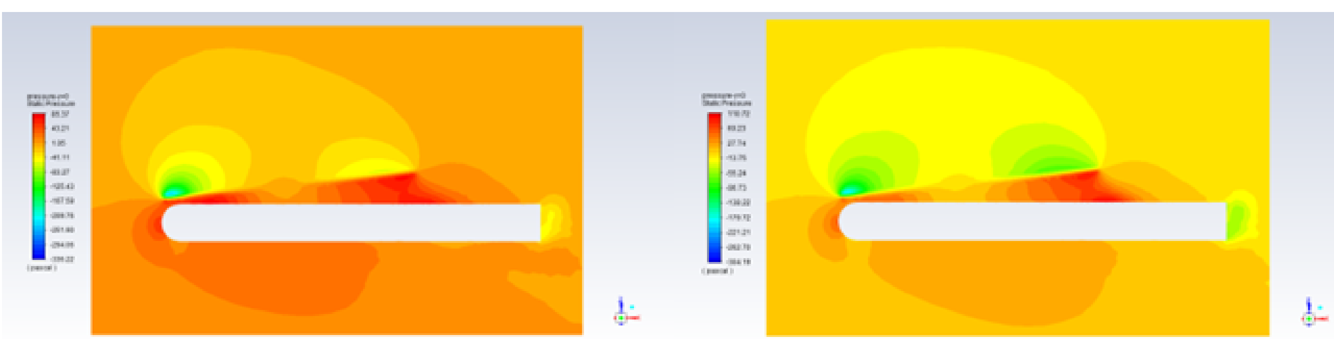


**Figure 31.17: Static Pressure at the Y=0 Cutting Plane, Seen from the Side**



It is important to point out that the pitch of the teetering rotor in the experiment is fixed at 10° collective with no possibility to control the cyclic component, while the simulation performs trimming by adjusting the cyclic pitch. This is a valid approach due to the equivalence of flapping and cyclic pitching: as far as rotor aerodynamics is concerned, one degree of cyclic pitch produces the same effect as one degree of flapping.

To evaluate the performance of the VBM with and without the trimming algorithm, Figure 31.18: Fixed-Pitch Helicopter Rotor Simulation (p. 359) and Figure 31.19: Collective and Cyclic Trimmed Helicopter Rotor Simulation (p. 360) compare in detail the experimental (symbols) and numerical (lines) pressure coefficient along the fuselage at port, starboard, top and bottom positions, as function of *x/R*. Here *x* is the streamwise distance starting at the tip of the airframe and *R* is the radius of the rotor. The pressure coefficient $c_p$ is defined as

$$c_p = \frac{p - p_{ref}}{0.5\,\rho_{ref} u_{ref}^2}$$

where the reference values are taken to be equivalent to the values at the entrance of the wind tunnel test section. While both approaches yield reasonable agreement with the experimental data, it is obvious that the thrust and moment-trimmed results match the experiment significantly better.

In particular the pressure distribution on the top of the fuselage agrees well in the trimmed case apart from the pronounced pressure peak due to the rotor wake impinging on the forward fuselage. A possible explanation for this discrepancy is the proximity of the rotor blade tip to the nose of the airframe, which results in a strong rotor-airframe interaction, especially since the untwisted and un-tapered blade produces a very strong tip vortex. In the VBM simulation, the effect of blade thickness is ignored, even though it may be significant when the blade passes within a few chord lengths from the fuselage. Furthermore, the present time-averaged method is unable to resolve the discrete tip vortex observed in the experiment. The experiments, as well as unsteady analysis, reveal a two-to-seven times higher $c_p$ value over the steady-state solution due to the blade passage effect. Besides this artificial diffusion of vorticity, grid-based methods inherently introduce numerical diffusion and therefore artificial damping of convected vorticity. This theory is supported by a comparison with the work of Marvis et al. (References (p. 361)), who were able to capture this strong peak significantly better using a gridless lifting line/free wake analysis.

The port and starboard side agree very well in the trimmed case, apart from the pronounced pressure drop caused by the rotor wake-leading edge, as the flow that impinges on the top of the fuselage accelerates around the sides. The failure of the VBM in this region is presumably of similar origin as discussed above.

Along the bottom of the airframe, the severe rotor-wake interaction is mostly blocked by the airframe itself and the pressure disturbance is minimal. The interference from the edges of the rotor-wake can still be seen, but they are swept rearward. Curiously, the result at the bottom seems to get worse with trimming enabled, but this might be due to the underprediction of the rotor-wake leading edge, discrepancies in the cylinder length between experiment and analysis and the omission of the airframe support mechanism.

**Figure 31.18: Fixed-Pitch Helicopter Rotor Simulation**

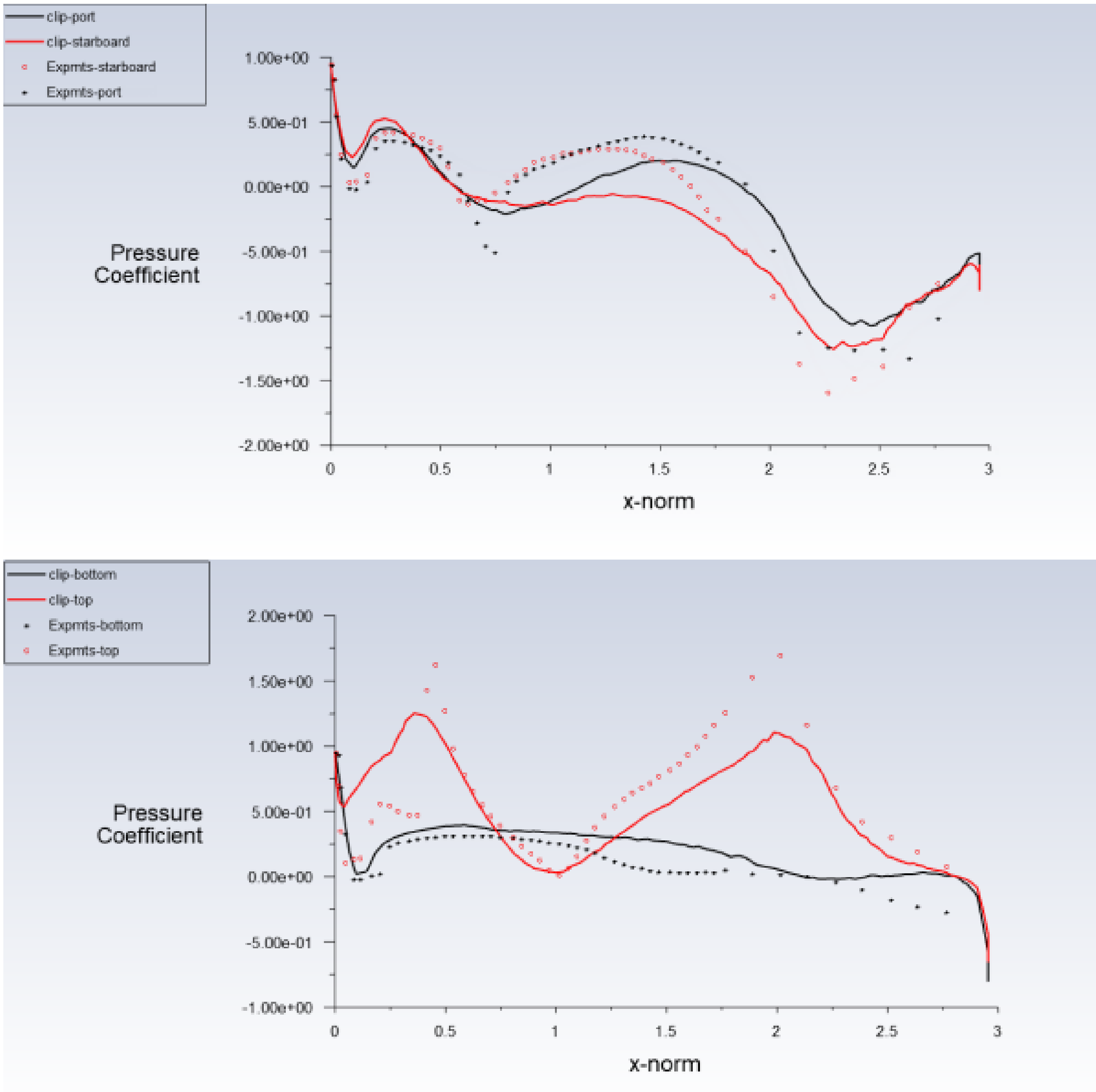
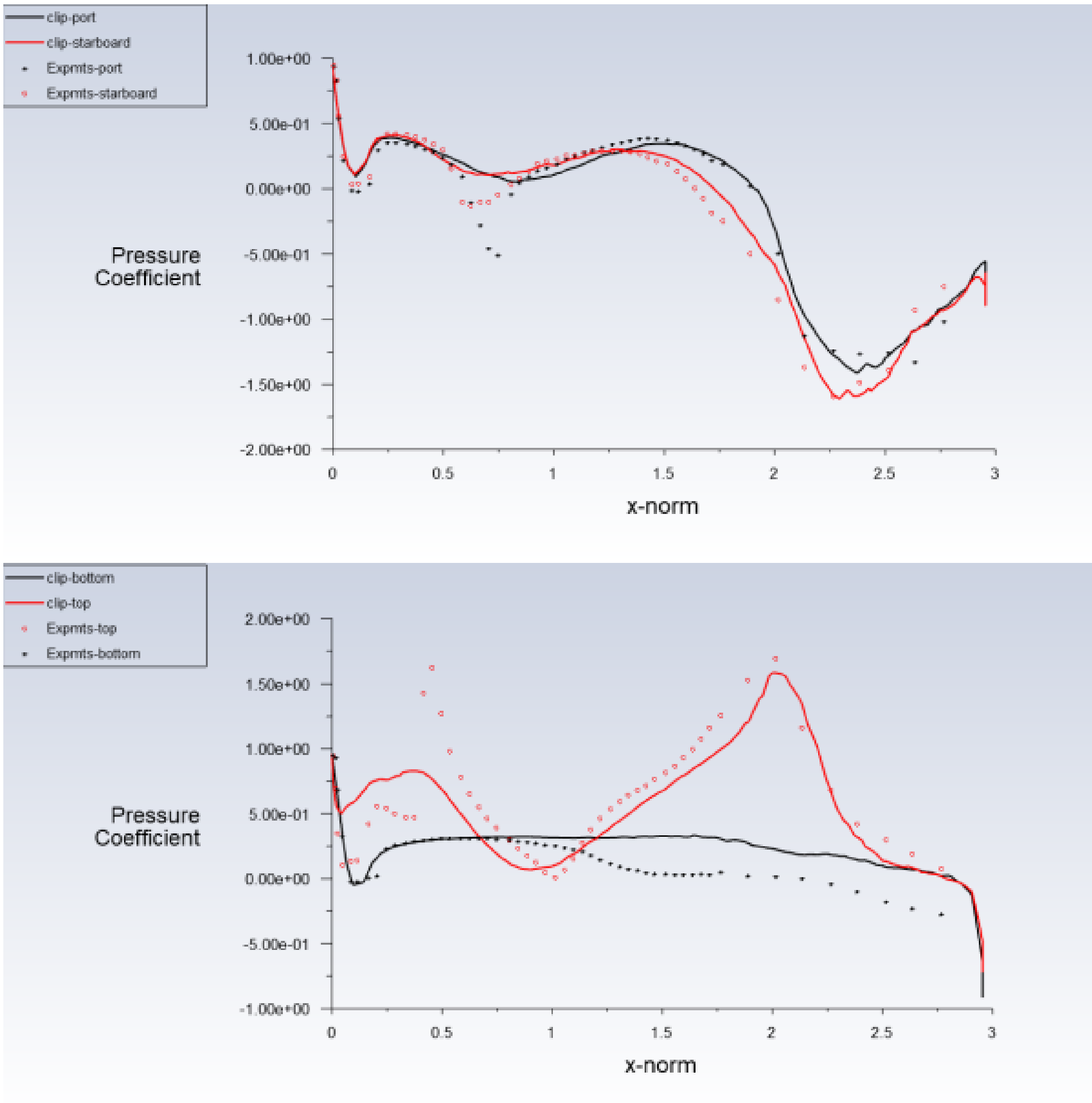


**Figure 31.19: Collective and Cyclic Trimmed Helicopter Rotor Simulation**




## 31.10. Summary

We present a method for analyzing the mutual aerodynamic interaction between multiple rotors and airframes implemented into the CFD solver Ansys Fluent. The model is based on the rotor disk momentum approach introduced by Zori et al. and the work of Yang et al. (References (p. 361)) but introduces some improvements. The current model permits simultaneous simulations of multiple rotors using an unstructured mesh topology inside the rotor disk. Furthermore, complex rotor blade geometries can be considered and an automatic and robust trim routine for thrust and moment trimming is available. As validation examples, an isolated propeller simulation and a rotor-airframe interaction case in forward flight are shown. Good agreement with the experimentally-obtained pressure distribution is observed.

## 31.11. References


1. Chaffin, M.S., "A guide to the use of the pressure disk rotor model as implemented in INS3D-UP", NASA CR-4692, September 1995.

2. Zori, L.A.J., Rajagopalan, R.G., "Navier-Stokes Calculation of Rotor-Airframe Interaction in Forward Flight", Journal of the American Helicopter Society, Vol.40, April 1995.

3. Stepniewski, W.Z. and Keys, C.N., "Rotary-Wing Aerodynamics," Dover Publications Inc., New York, USA, 1984.

4. Leishman, J.G., "Principles of Helicopter Aerodynamics", Cambridge Aerospace Series, Cambridge University Press, 2000.

5. "Ansys Fluent Customization Manual 2019R3," ANSYS, Inc., Canonsburg, PA, USA.

6. Yang, Z., Sankar L.N., Smith, M., Bauchau, O., "Recent Improvements to a Hybrid Method for Rotors in Forward Flight", AIAA Conference Reno, 2000.

7. "Fluent User's Guide 2019R3," ANSYS, Inc., Canonsburg, PA, USA.

8. Capitao Patrao, A., "Description and Validation of the rotorDiskSource Class for Propeller Performance Estimation," Proceedings of CFD with Open Source Software, edited by Nilsson, H., 2017.

9. Liou, S.G., Komerath, N.M., McMahon, H.M, "Velocity Measurements of Airframe Effects on a Rotor in Low-Speed Forward Flight" J. Aircraft Vol.26, No.4, 1989.

10. Brand, A., Komerath, N., McMahon, H., "Results from Laser Sheet Visualization of a Periodic Rotor Wake", J. Aircraft Vol.26, No.5, 1989.

11. Liou, S.G., Komerath, N.M., McMahon, H.M., "Velocity Field of a Cylinder in the Wake of a Rotor in Forward Flight", J. Aircraft Vol.27, No.9, 1990.

12. Liou, S.G., Komerath, N.M., McMahon, H.M, "Measurements of the Interaction Between a Rotor Tip Vortex and a Cylinder", AIAA J. Vol.28, No.6, 1990.

13. Marvis, D.N., Komerath, N.M., McMahon, H.M., "Prediction of Aerodynamic Rotor-Airframe Interaction in Forward Flight", Journal of the American Helicopter Society, Vol.34, No.4, 1989.

14. Brand, A.G., Komerath, N.M., McMahon, H.M., "Windtunnel data from a rotor wake/airframe interaction study", Georgia Institute of Technology, US Army research contract No. DAAG 29-82-K-0094, 1986.

15. Prouty, R., "Helicopter Performance, Stability and Control", Notes by The University of Kansas Continuing Education, September 13-17, 2004.

16. Lorber, P.F., Egolf, T.A., "An Unsteady Helicopter Rotor-Fuselage Aerodynamic Interaction Analysis", Journal of the American Helicopter Society, Vol.35, No.3, 1990.

# 31.12. Fluent's Virtual Blade Model Tutorials

## 31.12.1. Fluent's Virtual Blade Model Helicopter Tutorial

This tutorial provides guidelines for setting up and solving a simple helicopter simulation using Fluent's Virtual Blade Model (VBM). The physical rotor is replaced with an actuator disk of finite thickness that provides the framework to simulate the thrust and torque of the actual rotor using the momentum source terms in Fluent's governing equations. Local flow characteristics in the actuator disk are extracted from the 3D flow solution generated by Fluent and used by the VBM to compute the forces acting on each blade section from airfoil look-up tables, then applied to the cells composing the actuator disk. Hence the unsteady rotor problem is replaced with a much simpler time-averaged procedure that can be used very effectively for the initial design of a real helicopter. The simplified helicopter geometry in this tutorial consists of a cylinder with a hemispherical nose and a flat disk to simulate the simple tethering rotor and rotor/fuselage flow interaction in the Georgia Institute of Technology (GIT) wind tunnel test chamber. The primary purpose of this tutorial is to illustrate the methodology for conducting this type of simulation, however, even though the mesh is very coarse, very reasonable agreement with the published experimental data can be obtained.

### 31.12.1.1. Introduction

This tutorial illustrates the methodology for configuring and solving flow on a simple helicopter using Fluent's Virtual Blade Model (VBM). The VBM is an intermediate method for modeling the time-averaged cumulative effects of the rotating blades, between the simpler FAN and the more complex Single or Multiple Rotating Frames (SRF/MRF) approaches.

The VBM replaces the rotating blades with analogues composed of stacks of 2D airfoil sections. The aerodynamic properties of the airfoil sections, in the form of tables of lift and drag coefficients as functions of the angle of attack, are used to model the effect of the blades according to the Blade Element Theory (BET). Hence, the rotational velocity, local blade angle, twist, and chord effects on the airflow passing through the rotor disk can be included in the overall Fluent simulation of the aircraft/rotorcraft. The Fluent VBM is significantly more accurate than the FAN model and its results compare favorably to the SRF/MRF model, without the very large cell count required to mesh the complete blades with sufficiently fine resolution.

The VBM introduces the forces generated by the blades on the working fluid via momentum sources applied in the grid cells attached to the rotor disk, also called the actuator disk, allowing the pressure jump across the disk to vary with radius and azimuth. This eliminates the need to generate individual meshes over each of the rotor blades, significantly reducing complexity, computational effort and mesh generation time. The magnitude of the momentum sources is computed according to the BET, allowing for the variation of twist angle, chord and airfoil types along the span. The non-linear aerodynamic interaction between the rotor and other structural components is solved by introducing the VBM into Fluent through a User-Defined Function (UDF). The airfoil $C_L$, $C_D$ tables required by the BET can also be functions of Mach and Reynolds number, yielding greater fidelity for both in-compressible and compressible flow simulations. The rotor disk pitch and bank angle with respect to the incoming flow velocity can also be considered. Finally, the blade pitch angle can be updated automatically by a trimming routine to match the specified thrust coefficient.

Although the Fluent VBM has been specifically developed for rotorcraft, it is also applicable to general rotating blade applications (propellers, wind power, HVAC, automotive, marine, etc.), for flows characterized by:

- Low to moderate blade loading
- Predominantly axial flow
- Negligible geometrical blockage

This tutorial will cover the following topics:

- Loading the VBM add-on module in Fluent.
- VBM meshing requirements.
- Setting up and running a test case using the VBM.
- Running an untrimmed, thrust-trimmed, and thrust-and-moment-trimmed simulation.
- Post-processing the three different sets of results and interpreting the differences.

The results shown in this tutorial have been obtained on 10 cores of a 12-core Intel Xeon® Gold 6136 CPU @ 3.00 GHz running Windows 10 Enterprise, version 10.0.18362 build 18362.

### 31.12.1.2. Problem Description

This tutorial will demonstrate the simulation of a simple helicopter in forward-flight in the Georgia Institute of Technology wind tunnel [1-6] shown in Figure 31.20: Simple Helicopter in the GIT Wind Tunnel Test Section (p. 364). The geometric and operating conditions are listed in Table 31.5: Geometric Data and Operating Conditions (p. 363).

**Table 31.5: Geometric Data and Operating Conditions**

| **Geometric Data** | |
|---|---|
| Tunnel dimensions | 2.134 x 2.743 x 4.877 m |
| Body diameter | 0.134 m |
| Body length | 1.35 m |
| Body/rotor clearance | 0.135m |
| Rotor blades | 2 |
| Rotor radius $r_{tip}$ | 0.4570 m |
| Cutout radius | 0.0125 m |
| Hinge offset | 0 m |
| Blade section | NACA 0015 |
| Blade chord | 0.086 m |
| Disk pitch angle | -6° |
| Disk bank angle | 0° |
| Collective pitch | 10° (fixed) |
| Coning angle | 0° |
| Longitudinal flapping angle | 1.94° |
| Lateral flapping angle | 2.03° |

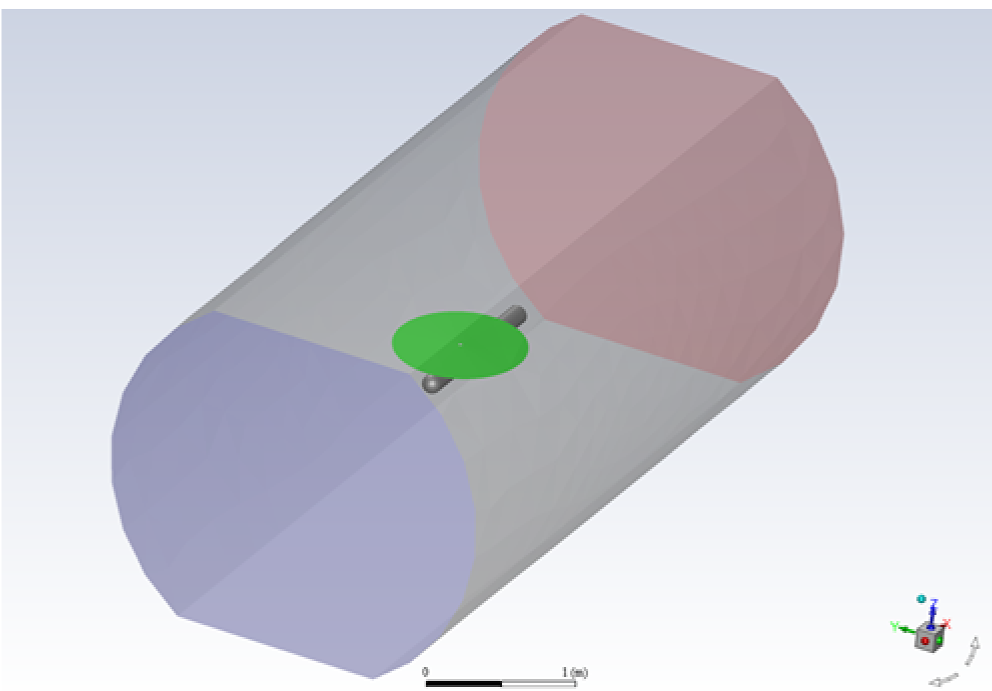
| Geometric Data | |
|---|---|
| Blade twist | 0° |

| Operating Conditions | |
|---|---|
| Advance ratio (*J*) | 0.1 |
| Reference pressure | 101263.15 Pa |
| Reference temperature | 288.0997 K |
| Reference density | 1.22451 kg/m$^3$ |
| Entrance velocity (*Vx*) | 10.04996 m/s |
| Rotational speed | 2,100 rpm<br>($\Omega$ = 219.9115 rad/s) |
| Tip speed | 100.49955 m/s |

For this simulation, the advance ratio is defined as:

$$J = \frac{V_x}{\Omega r_{tip}}$$

**Figure 31.20: Simple Helicopter in the GIT Wind Tunnel Test Section**




Three operating cases will be considered.

- No trimming
- Collective trimming
- Collective and cyclic trimming

Download the `vbm_helicopter_tutorial.zip` file here.

Unzip `vbm_helicopter_tutorial.zip` to your working directory.

The following files are required for the simulation and should be present in the working directory:

`naca0015.dat`

`VBM_helicopter_tutorial.msh.gz`

`xnr_phi0.xy`

`xnr_phi090.xy`

`xnr_phi180.xy`

`xnr_phi270.xy`

---

**Note:**

The coefficient of the dynamic pressure in the thrust and moment coefficients follows the North-American practice in the add-on module (set to 1.0).

---

### 31.12.1.3. Setting up the Calculation

Launch the Fluent executable, then:

1. Select the **3D**, **Double Precision** and **Parallel (Local Machine)** options.
2. Choose a suitable number of **Processes**, or only one if only a single CPU is available.
3. Select the appropriate **Working Directory** in the **General Options** panel.
4. Make sure that all of the files listed at the end of Problem Description (p. 363) are located in the **Working Directory**.
5. Click **Start**.

#### 31.12.1.3.1. Reading the Grid

1. To read the mesh file, go to:

   **File** → **Read** → **Mesh...**

   Select the `VBM_helicopter_tutorial.msh.gz` file and click **OK**.

2. To display the mesh, click on the following button in the ribbon.

   **Domain** → **Display...**

   Deselect **inlet**, **outlet** and **tunnel** in the **Surfaces** list, select **int_rotor** and **fuselage** and click **Display**.

The computational domain must be divided into two separate but communicating fluid domains (**Physics** → **Zones** → **Cell Zones**). The VBM only acts on the cells that form the **live1** cell zone attached to one side of the actuator disk.

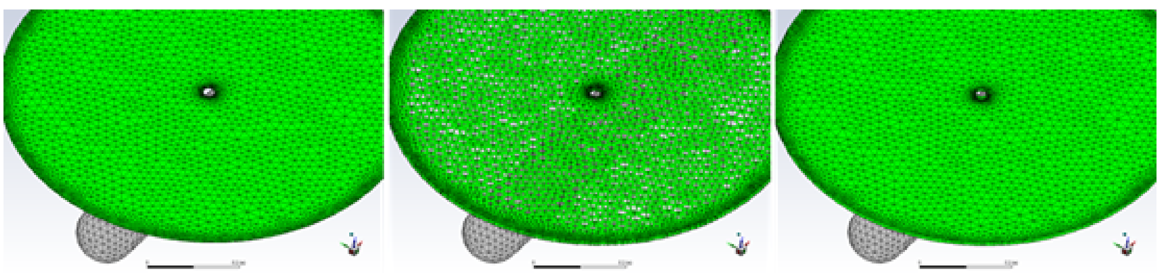
As shown in Figure 31.21: From Left to Right, the Three Components of the live1 Cell Zone: Int_Rotor, Int_live1 and Int_Live:003. (p. 366), Fluent automatically subdivides the **live1** cells into three separate entities.

- **int_rotor** contains the internal single-sided cell faces lying on the actuator disk surface;
- **int_live:003** contains the cell faces shared by the **live** and **live1** domains, excluding the actuator disk;
- **int_live1** contains the internal cell faces perpendicular to the disk;

3. To display the first two surfaces, select each in the list of **Surfaces**, and click **Display**.
4. For the third surface, however, click on **Adjacency** in the **Mesh Display** dialog box to open the **Adjacency** dialog box; select **live1** from the list of **Cell Zone(s)** and **int_live1** from the list of the **Adjacent Face Zones**; and then click on the **Display Face Zones**.

If this arrangement is not respected, the VBM will not work properly.

**Figure 31.21: From Left to Right, the Three Components of the live1 Cell Zone: Int_Rotor, Int_live1 and Int_Live:003.**



The following mesh topology characteristics must be respected:

- The entire 360° azimuth of the rotor must be modeled.
- The actuator disk surface (**int_rotor**) must have the interior boundary condition.
- The cells attached to one side of the disk must be marked as a separate domain (**live1**).
- These cells must have one complete face attached to the disk. Only hexa and prisms are allowed.
- A continuum fluid zone (**live**) must completely envelop the rotor zone (**live1**).
- The fluid zone and rotor zone must have different BC index (13 for the fluid zone and 1 for the rotor zone).

#### 31.12.1.3.2. Loading VBM Add-on Module

The VBM add-on module is a beta feature found under the `addons/vbm` directory. Therefore, it is required to enable the feature through the text user interface (TUI).

1. Enter the following TUI command in the console to enable the feature:

   ```
   (enable-feature 'vbm)
   ```

2. Enter the following TUI command in the console to load the VBM add-on:

   ```
   Define/models/addon-module 12
   ```

   ---

   **Note:**

   The VBM module can be loaded only after a valid Ansys Fluent mesh or case file has been set or read.

   VBM module is the 12th module listed in the console.

   During the loading process, a Scheme library containing the graphical and text user interface, and a UDF library containing a set of user-defined functions (UDFs) for the VBM add-on module are automatically loaded into Ansys Fluent.

   ---

3. Answer `yes` when asked to setup UDF hooks in the console. This will setup the UDM to 15 and set the Function Hooks.

   The loading processes are reported in the Fluent console and UDM offset should be equal to 0 at the end. Once the addon is loaded into Ansys Fluent, the Virtual Blade Model appears under the **Models** tree branch.

#### 31.12.1.3.3. Setup Units

Since this tutorial is in the SI system of units, and the rotor disk, blade pitch and blade flapping angles are provided in degree, go to:

**Domain** → **Mesh** → **Units...**

1. Click the **si** tab.
2. Click **angle** in the **Quantities** list and choose **deg** in the **Units** list.
3. Close the **Set Units** dialog box.

#### 31.12.1.3.4. Cell Zone Conditions

In the **Physics** section of the Fluent ribbon, go to:

**Zones** → **Cell Zones**

1. Select **live1** in the **Cell Zone Conditions** section in the **Task Page** and click the **Edit...** button
2. Ensure that the **Zone Name** is **live1** and the **Material Name** is **air**.
3. Tick mark the **Source Terms** box and click the **Source Terms** button in the ribbon just below.
4. Enable the **X Momentum** source term by clicking on the **Edit...** button. The **X Momentum Sources** window will open.

5. Set the **Number of X Momentum sources** to **1** with the up/down arrows or by entering the number in the box. Click on the down arrow just below and select the **udf xmom_src_1::vbm** option from the pull-down menu and press the **OK** button.

6. Repeat the operation above to configure the **Y Momentum** and **Z Momentum** sources with **udf ymom_src_1::vbm** and **udf zmom_src_1::vbm**, respectively. The confirmation message **1 source** should appear in the boxes next to the **X**, **Y** and **Z Momentum** sources.

7. Click **Apply** and **Close** the **Fluid** dialog box.

#### 31.12.1.3.5. Operating Conditions

To set the operating conditions, go to:

**Physics** → **Operating Conditions...**

1. Set a value of `101263.15` Pa in the **Operating Pressure** box.

2. Set the value of the **X** component of the **Reference Pressure Location** to -`1` m.

3. Click **OK**.

#### 31.12.1.3.6. Physical Modeling

1. To configure the Fluent solver settings, go to:

   **Physics** → **General...**

   Select the following options in **General Task** page:

   **Time** → **Steady**

   **Type** → **Pressure-Based**

   **Velocity Formulation** → **Absolute**

2. To enable energy equation, go to:

   **Physics** → **Models**

   Tick-mark the **Energy** equation box.

3. To select the turbulence model, go to:

   **Physics** → **Models** → **Viscous...**

   a. Select the **Spalart-Allmaras (1 eqn**) turbulence model

   b. Enable the **Strain/Vorticity – Based** production option.

   c. Tick mark the **Viscous Heating** and **Curvature Correction** boxes in the **Options** section.

   d. Click **OK** to accept all the other default settings and close the **Viscous Model** dialog box.

For vorticity-dominated flows, the default **Vorticity-Based**-production option overpredicts the production of eddy viscosity in the vortex cores. Adding the strain tensor to the vorticity reduces the production of turbulent viscosity in regions where the measure of vorticity exceeds that of strain rate.

### 31.12.1.3.7. Materials

This simulation features a high-speed flow regime, hence compressibility must be enabled. Go to:

**Physics** → **Materials** → **Create/Edit...**

1. Select **air** as the working fluid in the **Fluent Fluid Materials** pull-down menu
2. Select **ideal-gas** in the **Density** pull-down menu.
3. Click on the **Change/Create** button.
4. Close the **Create/Edit Materials** dialog box.

> **Note:**
>
> The VBM also works with the **constant** density and **incompressible-ideal-gas** options, however since rotors usually operate in the compressible regime, the **ideal-gas** option is more appropriate.

### 31.12.1.3.8. Boundary Conditions

To configure the boundary conditions, go to

**Physics** → **Zones** → **Boundaries**

1. Set the boundary condition at the inlet.
   a. Select the **inlet** boundary in the **Task Page**, ensure that its **Type** is **velocity-inlet** and click on the **Edit...** button.
   b. In the **Momentum** panel of the ribbon, select the **Components** option from the **Velocity Specification Method** pull-down menu.
   c. Select **Absolute** in the **Reference Frame** pull-down menu.
   d. Input the values (`10.049955`, `0`, `0`) m/s for the **X-**, **Y-** and **Z-Velocity** components, respectively.

   > **Note:**
   >
   > The velocity is calculated using the definition of the advance ratio **J** = 0.1.

   e. In the **Turbulence** section, select the **Intensity and Hydraulic Diameter** option from the **Specification Method** Pull-down menu.

f. Set the **Turbulence Intensity** value to **1**% (or **Turbulence Intensity (fraction)** value to **0.01**) and the **Hydraulic Diameter** value to **0.134** m (diameter of the fuselage). The diameter of the fuselage is assumed to represent the characteristic turbulent macroscopic length scale.

g. In the **Thermal** panel of the ribbon, set a **Temperature** value of **288.0997** K.

h. Click **Apply** and close the **Velocity Inlet** dialog box.

2. Set the boundary conditions at the outlet.

   a. Select the **exit** boundary, ensure that the **Type** is **pressure-outlet**, then click the **Edit...** button.

   b. Select **Absolute** in the **Backflow Reference Frame** pull-down menu.

   c. Set the **Gauge Pressure** value to **0** Pascal

   d. In the **Turbulence** section, select the **Modified Turbulent Viscosity** option in the **Specification Method** pull-down menu.

   e. Set a value of **0.0001** ($m^2/s$) in the **Backflow Modified Turbulent Viscosity** box. There is very little chance that backflow may occur, however it is good practice not to skip this operation.

   f. In the **Thermal** panel of the ribbon, set the **Backflow Total Temperature** to **288.15** K.

   g. Click **Apply** and close the **Pressure Outlet** dialog box.

3. Set wall boundary conditions.

   a. Select the **tunnel** boundary and click the **Edit...** button.

   b. In the **Momentum** panel of the ribbon, set the **Shear Condition** to **Specified Shear** with {**0.0.0**} **X-**, **Y-** and **Z-Components**, respectively, and click the **OK** button.

   c. In the **Thermal** panel of the ribbon, select **Heat Flux** in the **Thermal Conditions** and ensure that the **Heat Flux** value is **0** $W/m^2$.

   d. Click **Apply** and close the **Wall** dialog box.

### 31.12.1.3.9. Reference Values

To set the reference values, go to:

**Physics** → **Reference Values...**

1. Select **inlet** from the **Compute from** pull-down menu.

2. Set the **Area** value to **0.65612** $m^2$ (disk area).

3. Set the **Length Value** to **0.086** m (blade chord).

4. Select **live** from the **Reference Zone** pull-down menu.

#### 31.12.1.3.10. Discretization and Solution Controls

1. To set the discretization options, go to the **Solution Methods** task page,

   **Solution** → **Solution** → **Methods...**

   Use the pull-down menus to set the following options:

   - **Scheme** → **SIMPLE**
   - **Gradient** → **Green-Gauss Node Based***
   - **Pressure** → **PRESTO!****
   - **Density** → **Second Order Upwind**
   - **Momentum** → **Second Order Upwind**
   - **Modified Turbulent Viscosity** → **First Order Upwind**
   - **Energy** → **Second Order Upwind**

   **Note:**

   *The node-based averaging scheme is more accurate than the default cell-based scheme, especially on unstructured meshes, and most notably for triangular and tetrahedral meshes.

   **The **PRESTO!** pressure discretization scheme is recommended for highly rotating flows.

2. Retain the default solver parameters in:

   **Solution** → **Controls** → **Controls...** → **Solution Controls**.

#### 31.12.1.3.11. Convergence Monitoring

1. To configure the residuals monitors that will appear in the Fluent graphics window and console, go to:

   **Solution** → **Reports** → **Residuals...**

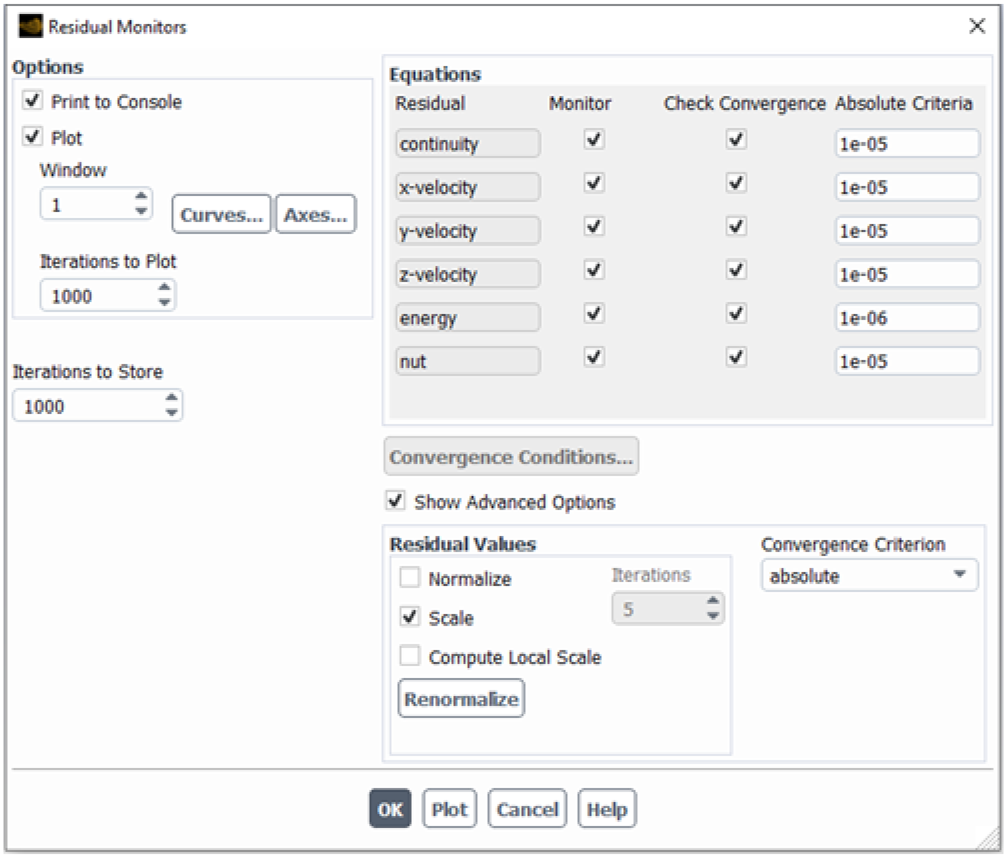
   a. Ensure that **Plot** and **Print to Console** options are enabled in the **Options** group box.

   b. Enable **Show Advanced Options** and select **absolute** from the **Convergence Criterion** drop-down list.

   c. Set the **Absolute Criteria** values to `1e-6` for **Energy equation** and to `1e-5` for other equations as shown in Figure 31.22: Solution Residuals Configuration (p. 372).

   d. Click **OK** to close the **Residual Monitors** dialog box.

**Figure 31.22: Solution Residuals Configuration**




2. Additionally, you may want to monitor pressure convergence on the actuator disk **int_rotor**. Go to:

   **Solution** → **Reports** → **Definitions** → **New...** → **Surface Report** → **Integral...**

   a. Enter `rotor-pressure` in the **Name** box.

   b. Tick-mark the **Report File**, **Report Plot** and **Print to Console** boxes in the **Create** section.

   c. Select **Pressure...** and **Static Pressure** with the pull-down menus in the **Field Variable** section.

   d. Select **int_rotor** in the **Surfaces** section.

   e. Click **OK** to close the **Surface Report Definition** dialog box.

The pressure convergence history will be displayed on screen and written in the file `rotor-pressure-rfile`. Additionally, the convergence histories of the thrust, torque, power and moments are written to the `Rotor_1_Loads.dat` file.

The value of the pressure integral and the convergence monitors will appear in the graphics window. The ribbon at the top left of the graphics window can be used to change the view from the residual monitors to the pressure integral and vice-versa.

#### 31.12.1.3.12. Solution Initialization

Initialize the solution from inlet boundary values. Go to:

**Solution → Initialization**

1. Select **Standard** and click **Options...** to open the **Solution Initialization** task page, which provides access to further settings.
2. Select **inlet** from the **Compute from** drop-down list.
3. Set the **Reference Frame** to **Absolute**.
4. Click the **Initialize** button.

### 31.12.1.4. Rotor Inputs

The last step before running the case is the configuration of the physical parameters of the helicopter rotor.

To set up the rotor parameter, go to:

**Physics → Models → More...**

1. Select **Virtual Blade Model** in the **Model** task page and click the **Edit...** button. This will open the VBM mini-graphical user interface.
2. Ensure the **Number of Rotor Zones** and **Active Rotor Zone** are set to **1**.
3. Enter the parameters shown in Figure 31.23: General Disk Data Configuration Window (p. 374).
4. Click **int_rotor** in the list of **Surfaces** to select the actuator disk surface.
5. Click the **Geometry** button of the ribbon and enter the parameters shown in Figure 31.24: Geometry Configuration Window (p. 375).
6. Click **Change/Create** to save the settings.
7. Click **OK** to close the **Rotor Input** dialog box.

---

**Note:**

The **Change/Create** button must be clicked before moving on to the next rotor (when present) or before pressing the **OK** button. This sequence must always be respected, even if the mini-graphical user interface is re-opened to simply edit a parameter.

When the **OK** button is clicked, the **User-Defined → Execute on Demand** command is executed automatically.

The UDF will append the `.dat` suffix to the airfoil file names if it is omitted in the mini-graphical user interface.

Consult Fluent's Virtual Blade Model (p. 331) for more information on the geometry of the rotor disk and the effect of the parameters.

**Figure 31.23: General Disk Data Configuration Window**

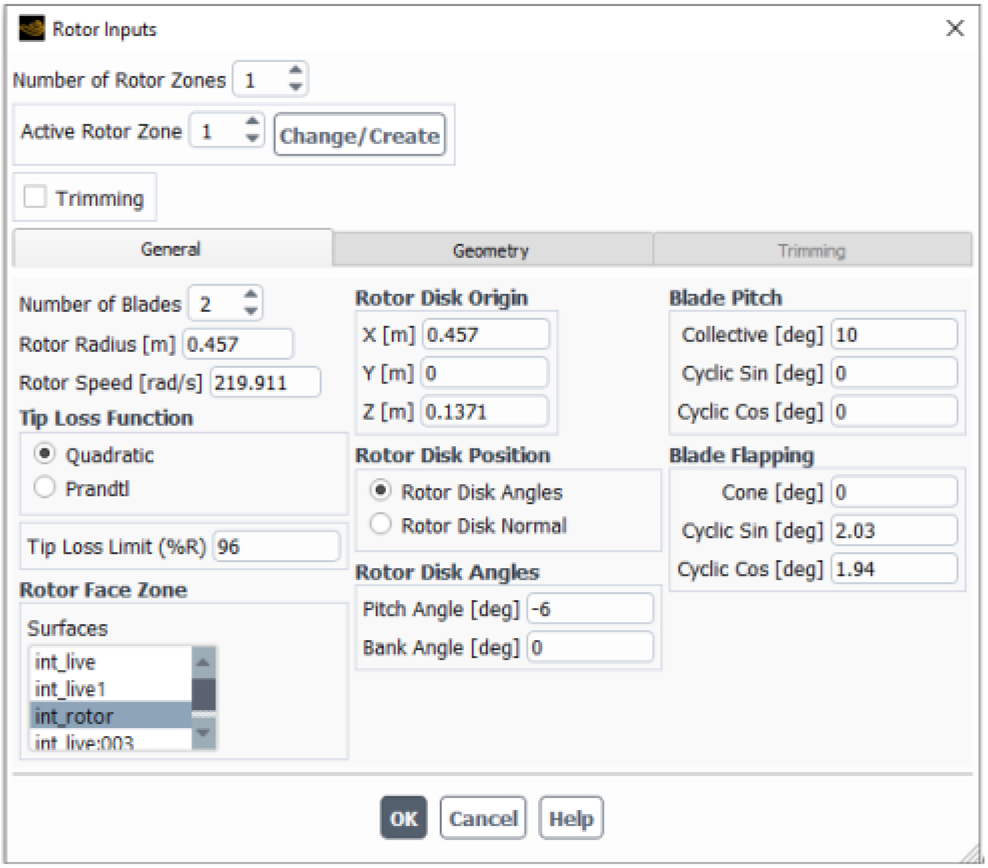
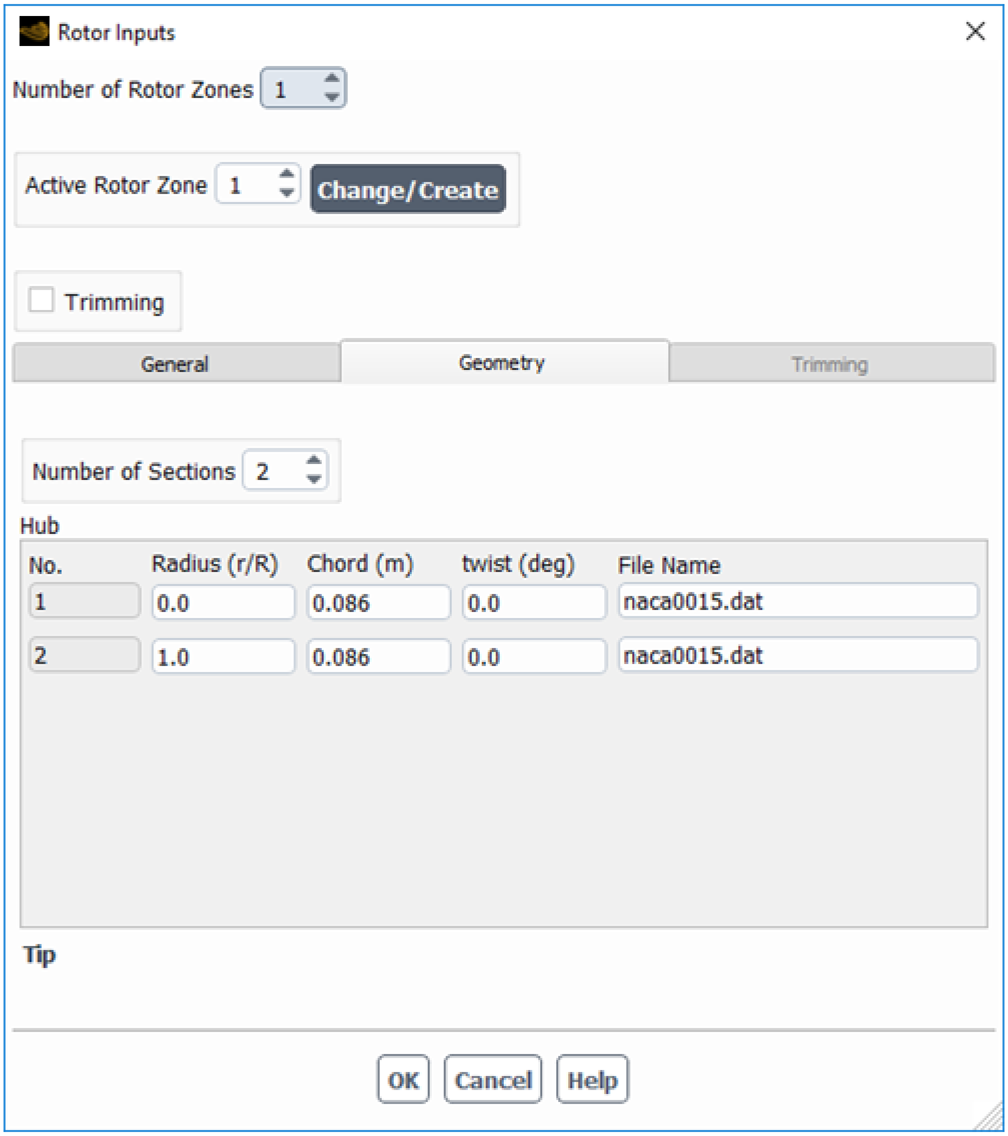


**Figure 31.24: Geometry Configuration Window**




## 31.12.1.5. Post-Processing

The following sections explain how to create a framework for extracting data from the solution and comparing it to the experimental data. The post-processing will be configured and saved before moving on to the actual solution process.

### 31.12.1.5.1. Curves for the Pressure Coefficient

Four curves will be created along the port and starboard sides and at the top and bottom of the cylindrical fuselage to enable the extraction of the pressure coefficient. The curves are saved so that they can be reused for the subsequent cases. Go to:

**Results** → **Surface** → **Create Iso-Surface...**

The **Iso-Surface** panel is shown in Figure 31.25: Iso-Surface Creation Panel (p. 376).

1. Create clip-z.

a. Enter the name **clip-z** in the **New Surface Name** box.

b. Select **Mesh...** and then **Z-Coordinate** from the **Surface of Constant** drop-down list.

c. Select **fuselage** in the **From Surface** list and **live** in the **From Zones** list.

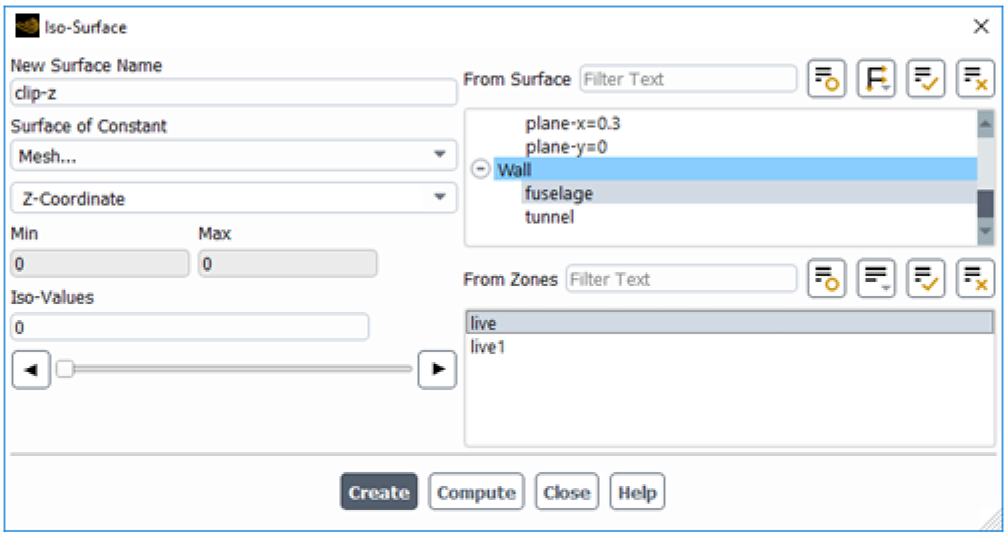
d. Set the **Iso-Values** to **0** m.

e. Click the **Create** button.

This operation creates a curve at the intersection of the cylindrical fuselage and an X-Y cutting plane at position **Z** = 0 m.

To define the next curve, do not close **Iso-Surface** dialog box.

**Figure 31.25: Iso-Surface Creation Panel**




2. Create clip-y.

   a. Name the curve around the fuselage **clip-y** in the **New Surface Name** box.

   b. Select **Mesh...** and then **Y-Coordinate** in the **Surface of Constant** drop-down list.

   c. Select **fuselage** in the **From Surface** list and **live** in the **From Zones** list

   d. Set the **Iso-Values** to **0** m.

   e. Click the **Create** button.

   f. Close the **Iso-Surface** dialog box.

   This operation creates a curve at the intersection of the cylindrical fuselage and an X–Z cutting plane at position **Y** = 0 m.

3. Create clip-port.

The curves are not useful in their present form, they need to be divided into port and starboard, top and bottom sides. Go to:

**Results** → **Surface** → **Create** → **Iso-Clip...**.

The panel is shown in Figure 31.26: Iso-Clip Panel for the Creation of the Clip-Port Curve (p. 377).

a. Enter the name `clip-port` in the **New Surface Name** box.

b. Select **clip-z** in the **Clip Surface** list.

c. Select **Mesh...** and then **Y-Coordinate** in the **Clip to Values of** pull-down menus.

d. Click the **Compute** button. The maximum and minimum $Y$ limits of the curve **clip-z** are displayed under the two circular dials.

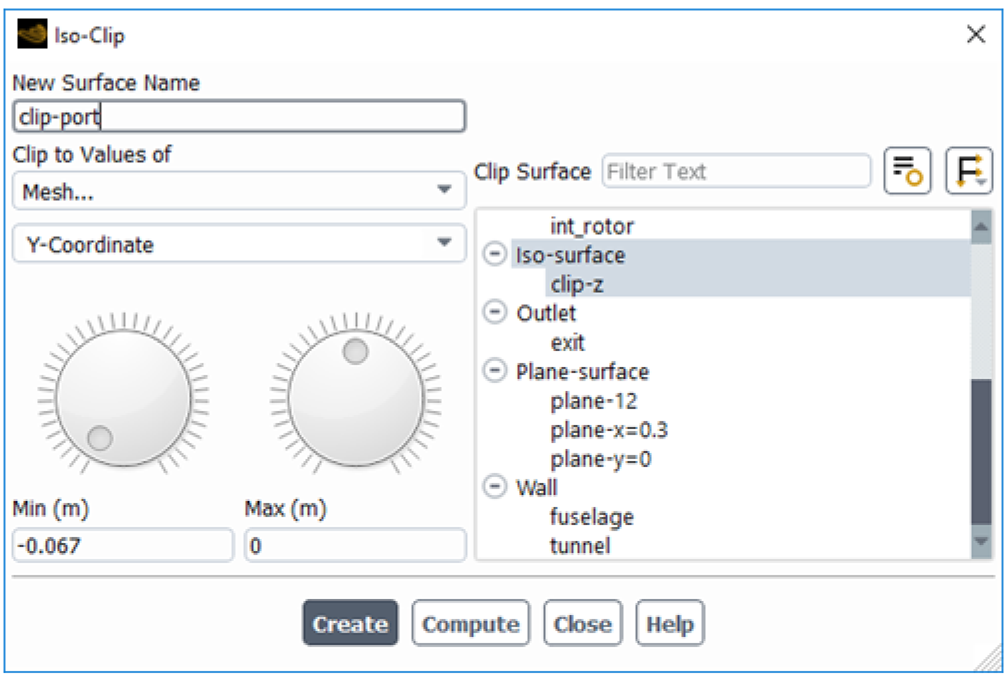
e. Change the **Max (m)** limit to `0` and press the **Enter** key.

f. Leave the **Min (m)** value at `-0.067`.

g. Click the **Create** button.

To define the next curve, do not close the **Iso-Clip** dialog box.

**Figure 31.26: Iso-Clip Panel for the Creation of the Clip-Port Curve**




4. Create clip-starboard.

   a. Enter the name `clip-starboard` in the **New Surface Name** box.

   b. Retain **clip-z** in the **Clip Surface** list.

   c. Click the **Compute** button again.

d. Change the **Min (m)** to **`0`** and press the **Enter** key.

e. Leave the **Max (m**) value at **`0.06699026`**.

f. Click the **Create** button.

To define the next curve, do not close the **Iso-Clip** dialog box.

5. Create clip-bottom.

a. Enter the name **`clip-bottom`** in the **New Surface Name** box.

b. Select **clip-y** in the **Clip Surface** list.

c. Select **Mesh...** and then **Z-Coordinate** in the **Clip to Values of** pull-down menus.

d. Click the **Compute** button. The maximum and minimum **Z** limits of the curve **clip-y** are displayed under the two circular dials.

e. Change the **Max (m**) limit to **`0`** and press the **Enter** key.

f. Leave the **Min (m**) value at **`-0.06693347`**.

g. Click the **Create** button.

To define the next curve, do not close the **Iso-Clip** dialog box.

6. Create clip-top.

a. Enter the name **`clip-top`** in the **New Surface Name** box.

b. Retain **clip-y** in the **Clip Surface** list.

c. Click on the **Compute** button again.

d. Change the **Min (m)** to **`0`** and press the **Enter** key.

e. Leave the **Max (m)** value at **`0.06698362`**.

f. Click the **Create** button.

g. Close the **Iso-Clip** dialog box.

7. Delete **clip-y** and **clip-z**.

The **clip-y** and **clip-z** curves are no longer needed and therefore can be deleted. Go to:

**Results** → **Surface** → **Manage...**

a. Select **clip-y** and **clip-z** in the **Surfaces** list and click on the **Delete** button.

b. **Close** the **Surfaces** dialog box.

#### 31.12.1.5.2. Cutting Planes for the Pressure Distributions

Create two cutting planes to visualize the rotor-fuselage interaction. One cutting plane will be oriented perpendicular to the centerline of the fuselage, and the other one will be parallel to the centerline of the fuselage. The two cutting planes will enable a comparison of the effect of trimming on the solution (no trimming, collective only, collective and cyclic trimming). Go to:

**Results** → **Surface** → **Create** → **Plane...**

1. Create plane-y=0

    a. Enter **`plane-y=0`** in the **Contour Name** box.

    b. Select **Three Points** from the **Method** drop-down list and tick-mark the **Bounded** box in the **Options** list. This will allow the creation of a bounded cutting plane using three corner points.

    c. Enter the coordinates of the corner points in the **Points** section: **P(0)** = {**`-0.25`**, **`0`**, **`-0.40`**}, **P(1)** = {**`1.5`**, **`0`**, **`-0.40`**} and **P(2)** = {**`1.5`**, **`0`**, **`0.70`**}.

    d. Click the **Create** button.

    To define the next plane, do not close the **Plane Surface** dialog box.

2. Create plane-x=0.3

    a. Enter the name **`plane-x=0.3`** in the **Contour Name** box.

    b. Enter the coordinates of the corner points in the **Points** section: **P(0)** = {**`0.3`**, **`0.7`**, **`-0.5`**}, **P(1)** = {**`0.3`**, **`-0.7`**, **`-0.5`**} and **P(2)** = {**`0.3`**, **`-0.7`**, **`0.5`**)}.

    c. Click the **Create** button.

    d. Close the **Plane Surface** dialog box.

#### 31.12.1.5.3. Custom Field Function

The experimental pressure coefficient data is provided with respect to the normalized x-coordinate x/r, where r is the radius of the rotor. It is therefore necessary to define a Custom Field Function to facilitate the direct comparison. Go to:

**User-Defined** → **Custom...**

1. Select **Mesh...** and then **X-Coordinate** in the **Field Functions** pull-down menus. Click the ***Select*** button. An **x** will appear in the **Definition** box.

2. Enter the name **`x-norm`** in the **New Function Name** box.

3. Click the **/** button.

4. Enter the value **`0.457`** with the keys of the numeric keypad, as shown in Figure 31.27: Custom Field Function Configuration Panel (p. 380).

5. Click the **Define** and **Close** buttons.

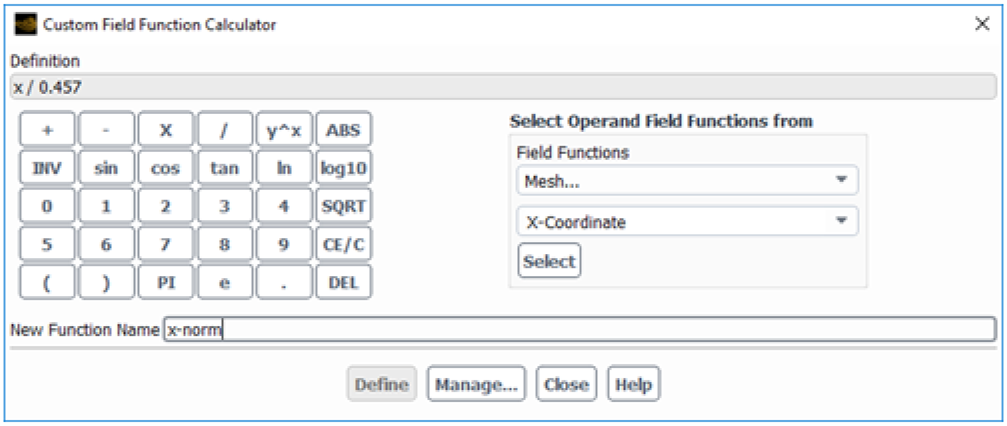
---

**Note:**

To change/modify an already defined custom field function, click on the **Manage...** button.

---

**Figure 31.27: Custom Field Function Configuration Panel**




### 31.12.1.6. Solution

The three simulations can now be executed to demonstrate the capabilities of the VBM. Before starting the simulation, it is a good idea to save the work at this point. Go to:

**File** → **Write** → **Case...**

and save the settings into the `VBM_helicopter_tutorial.cas` file. Alternately, the **Case & Data...** option can be used.

#### 31.12.1.6.1. Rotor Simulation with Fixed-Pitch

1. For the first simulation, the helicopter rotor is operating in fixed-pitch mode without trimming. Go to:

   **Solution** → **Run Calculation** → **Run Calculation...**

   Set the **No. of Iterations** value to **500** and click on the **Calculate** button.

   The convergence history of the residuals is shown in Figure 31.28: Convergence History (p. 381). Figure 31.29: Pressure Monitor Convergence History (p. 381) shows the convergence history of the integral of the static pressure on the **int_rotor** surface.

**Figure 31.28: Convergence History**

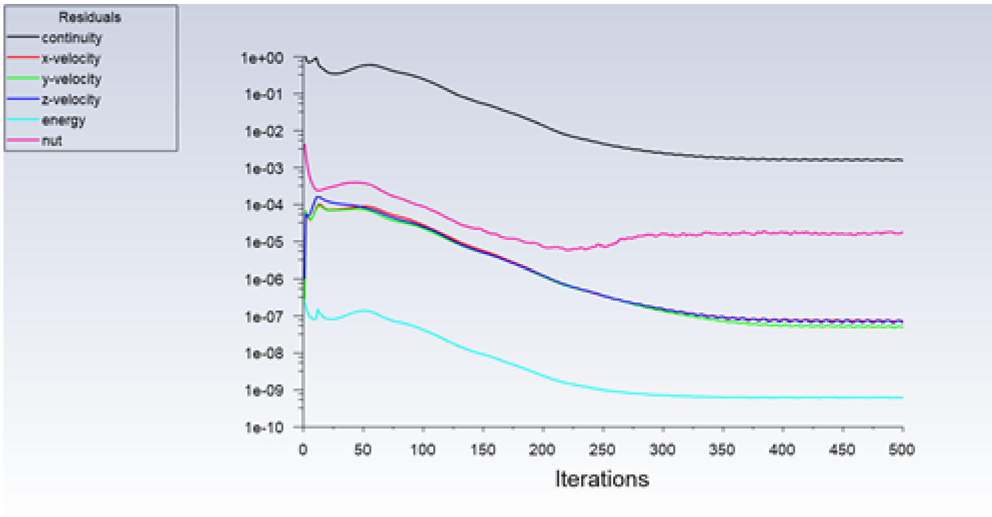
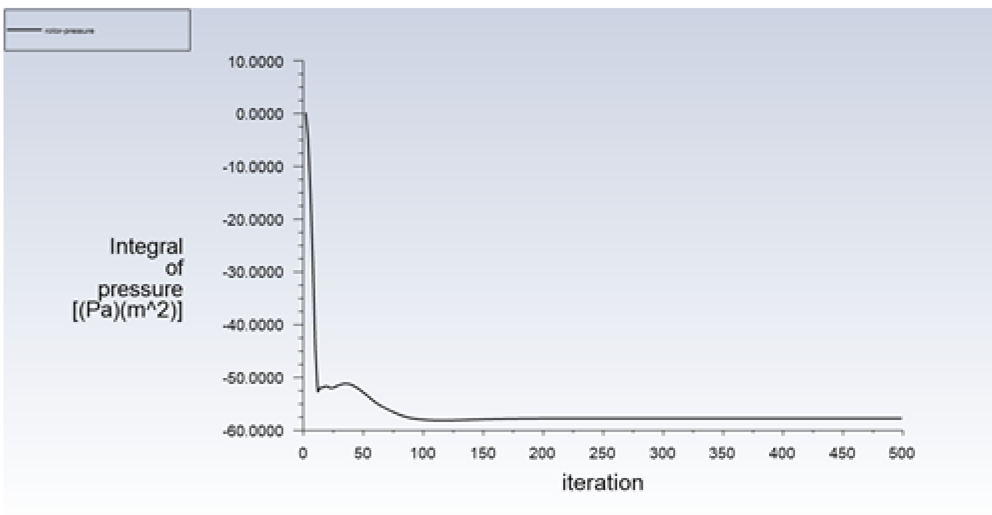



**Figure 31.29: Pressure Monitor Convergence History**




### Note:

In case the simulation stops with the following error message, "`The fl process could not be started`", proceed with the following steps from now on:

- Exit Fluent and re-launch it using a different number of CPUs. Sometimes a computer restart is required.
- Read the case and data files which were saved in the previous step.
- Verify if the UDF library is loaded and linked properly into Fluent ( (Loading VBM Add-on Module (p. 366) and Loading VBM Addon Module (p. 396)).
- Verify **Rotor Inputs** (Rotor Inputs (p. 373)), click **Change/Create** and then **OK**.
- Verify all other settings and continue.

2. Save the Fluent case and data files (`VBM_helicopter_tutorial.cas` and `.dat`):

   **File** → **Write** → **Case & Data...**

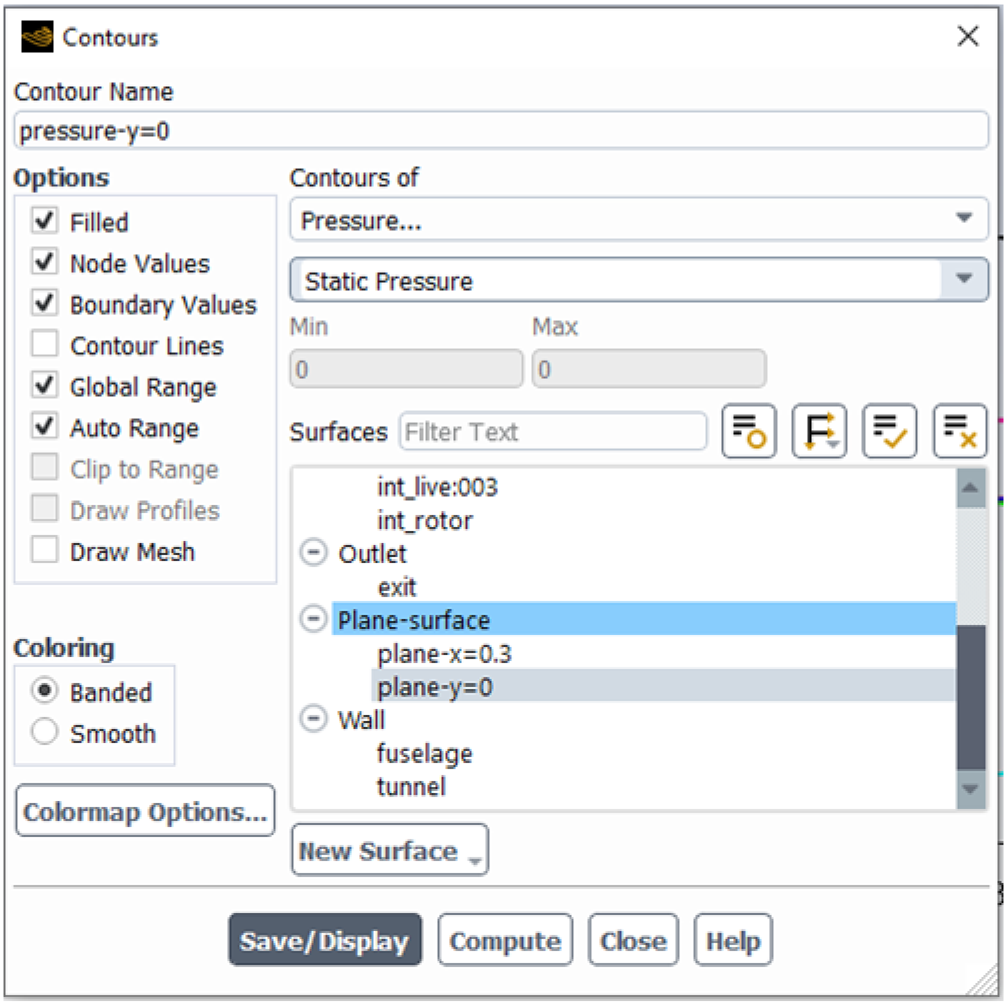
3. To display contours of pressure on the cutting planes that have been created in Cutting Planes for the Pressure Distributions (p. 379), go to:

   **Results** → **Graphics** → **Contours** → **New ...**

   a. Configure the window as shown in Figure 31.30: Display the Pressure Distribution on the Y=0 Plane (p. 382).

   b. Click on the **Save/Display** button to display the image in the graphics window.

**Figure 31.30: Display the Pressure Distribution on the Y=0 Plane**

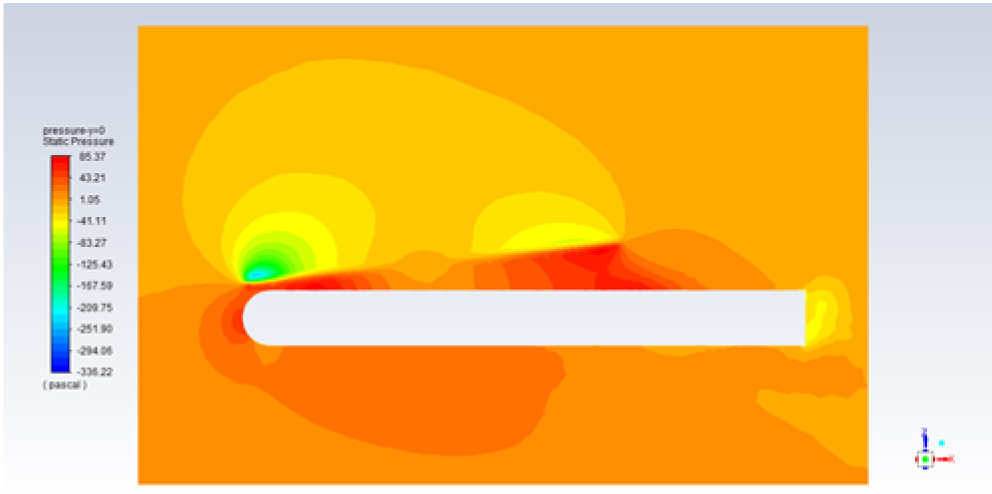



   c. To see the contours on plane-y=0, as shown in Figure 31.31: Pressure Distribution with Fixed Blade Pitch; Y=0 Cutting Plane (p. 383), click the green y-axis arrow in the axis triad twice and click the **Fit to Window** button in the graphics toolbar.

   d. To disable the light on the contour, go to the **Lights** dialog box, and de-select **Light On**:

   **View** → **Graphics** → **Lights...**

**Figure 31.31: Pressure Distribution with Fixed Blade Pitch; Y=0 Cutting Plane**




e. Create a similar contour for the plane-x=0.3 to see the image of Figure 31.32: Pressure Distribution with Fixed Blade Pitch; X=0.3 Cutting Plane (p. 383). Use the axis triad, rotation controls, and Lights dialog box as needed.

**Figure 31.32: Pressure Distribution with Fixed Blade Pitch; X=0.3 Cutting Plane**

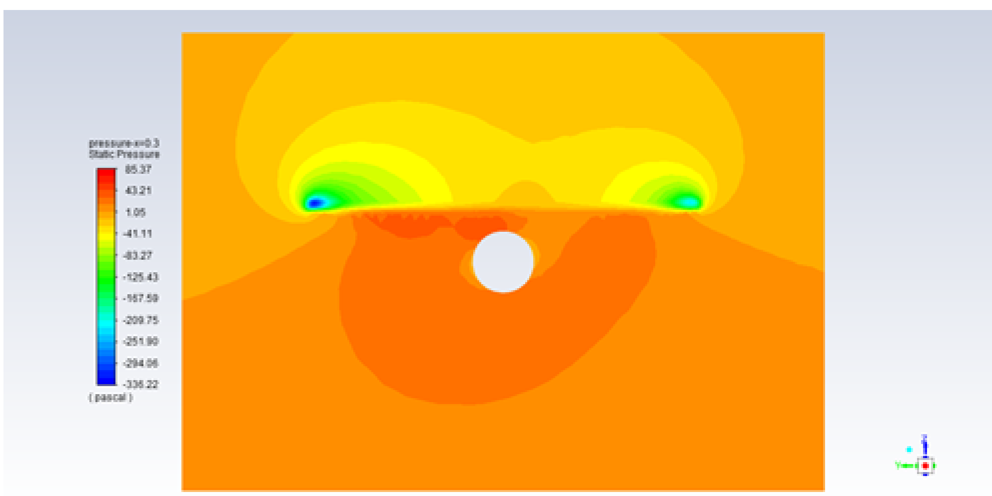



Figure 31.31: Pressure Distribution with Fixed Blade Pitch; Y=0 Cutting Plane (p. 383) and Figure 31.32: Pressure Distribution with Fixed Blade Pitch; X=0.3 Cutting Plane (p. 383) show that there is a clear pressure load imbalance on the disk, therefore if the helicopter were free to fly, it would not be able to maintain a straight-and-level course, rather it would tilt up and roll to the left. To achieve stability, the rotor needs to be trimmed so that the pressure integrated over the rotor produces no net moments, only lift and thrust in the forward direction. This can only be achieved by fine-tuning the rotor collective and cyclic pitch (thrust and moment trimming).

#### 31.12.1.6.2. Rotor Simulation with Collective Trimming

The next simulation should be identical to the previous one, with the exception that now the VBM add-on module will trim the collective angle of the rotor. It is expected that the trimming will recover the exact blade pitch angle of the fixed-pitch rotor and produce the same thrust. Before enabling the collective trimming, the thrust coefficient of the fixed-pitch rotor needs to be computed from the fixed-pitch solution with the following equation:

$$C_T = \frac{T}{C_{US-EU}\rho V_{tip}^2 \pi \frac{d^2}{4}}$$

Table 31.6: Values from the Fixed-Pitch Solution (p. 384) lists the values of the variables appearing in the formula. The rotor thrust $T$ can be recovered from column 3 of the last line of the VBM log file `Rotor_1_Loads.csv`. $C_{US-EU}$ is the coefficient of the dynamic pressure set to 1 in the code. The blade tip velocity and the reference density are listed in Table 31.5: Geometric Data and Operating Conditions (p. 363).

**Table 31.6: Values from the Fixed-Pitch Solution**

| Variable | | Value |
|---|---|---|
| $T$ | rotor thrust | 75.95131 N |
| $C_{US-EU}$ | dynamic pressure coefficient | 1 |
| $V_{tip}$ | blade tip velocity | 100.49955 m/s |
| $\rho$ | reference density | 1.22451 kg/m3 |
| $d$ | rotor diameter | 0.914 m |

The computed thrust coefficient value **CT** is 0.0093597.

1. To enable collective trimming, go to:

   **Physics → Models → More...**

   a. Select the **Virtual Blade Model** and click on the **Edit...** button to open the mini-graphical user interface.

   b. Tick-mark the **Trimming** box, then select the **Trimming** panel in the ribbon.

   c. Tick-mark the **Collective pitch** option, set the **Update Frequency value** to **`10`**, the **Damping Factor** to **`0.7`** and enter the value **`0.0093597`** in the **Desired thrust coefficient** box.

   d. Click the **Change/Create** button, then press the **OK** button.

2. To run the calculation, go to:

   **Solution → Run Calculation...**

   Set a value of **`200`** in the **No. of Iterations** box and press the **Calculate** button.

   The residuals and pressure monitor values should not change for the next 200 iterations. This short run is required only to verify that the thrust coefficient value is precise and the collective angle printed to the console after 200 iterations converges to the original 10-degrees fixed-pitch blade angle.

3. When the execution terminates, save the case and data files (`VBM_helicopter_tutorial_VP.cas` and `.dat`).

   **File** → **Write** → **Case & Data...**

#### 31.12.1.6.3. Rotor Simulation with Collective and Cyclic Trimming

1. To enable cycling trimming, go to:

   **Physics**→ **Models** → **More**

   a. Select the **Virtual Blade Model** in the **Task Page** and click on the **Edit...** button to open the mini-graphical user interface.

   b. Select the **Trimming** option in the ribbon. Tick-mark the **Cyclic pitch** option, then enter the value **`0`** in the **Desired x-moment coefficient** and **Desired y-moment coefficient boxes**.

   c. Click the **Change/Create** button, then press the **OK** button.

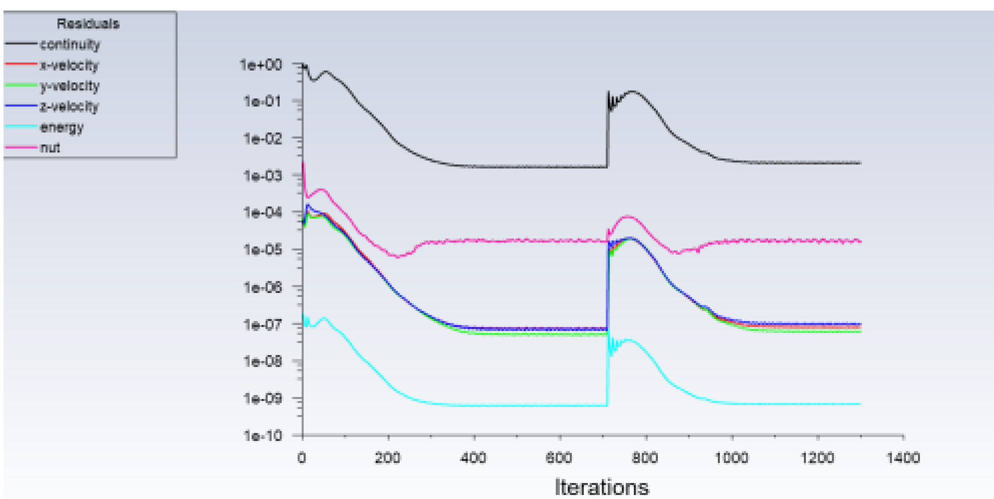
2. To run the calculation, go to:

   **Solution**→ **Run Calculation...**

   Set a value of **`600`** in the **No. of Iterations** box and press the **Calculate** button.

   At convergence the residual history will look like Figure 31.33: Convergence History with Collective and Cyclic Angles Trimming (p. 385).

   **Figure 31.33: Convergence History with Collective and Cyclic Angles Trimming**




3. When the execution terminates, save the case and data files (`VBM_helicopter_tutorial_Co_Cy.cas` and `.data`):

   **File** → **Write** → **Case & Data...**

The converged rotor thrust with collective and cyclic trimming is listed in column 3 of the `Rotor_1_Loads.csv` log file as 75.95093 N, a difference of -0.0005% with respect to the thrust obtained with fixed pitch. Because of the cyclic pitch trimming, however, the collective angle changes from 10˚ to 10.093873˚ and the longitudinal and lateral cyclic angles are 2.013306˚ and 2.616949˚, respectively.

Figure 31.34: Pressure Distribution with Collective and Cyclic Trimming; Y=0 Cutting Plane (p. 386) and Figure 31.35: Pressure Distribution with Collective and Cyclic Trimming;X=0.3 Cutting Plane (p. 386) show that the cyclic trimming has balanced the pressure distribution almost equally along, and especially across, the rotor.

**Figure 31.34: Pressure Distribution with Collective and Cyclic Trimming; Y=0 Cutting Plane**

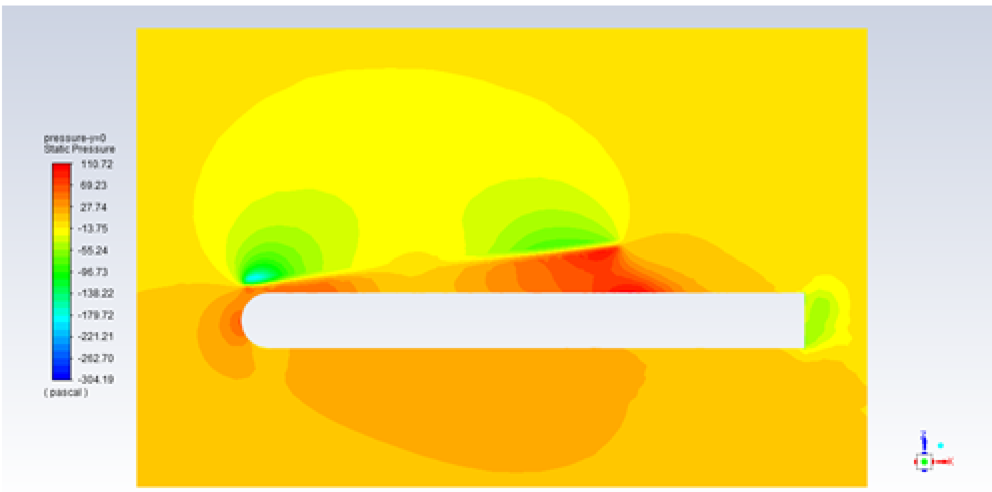
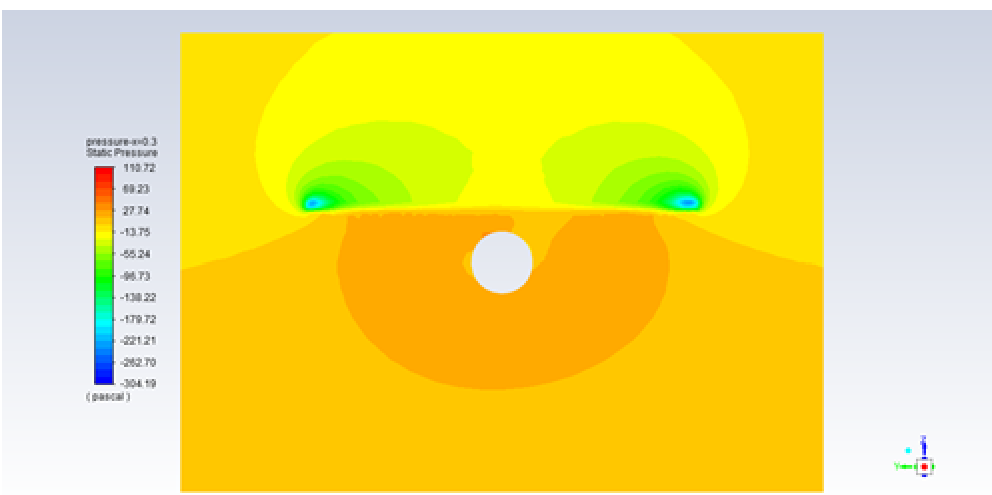



**Figure 31.35: Pressure Distribution with Collective and Cyclic Trimming;X=0.3 Cutting Plane**




## 31.12.1.6.4. Comparison with Experimental Results

Even though the purpose of this tutorial is to illustrate the usage of the Ansys Fluent VBM and the computational grid is rather coarse in order to keep the execution times as low as possible, the computational results can be compared to the experimental data for validation.

#### 31.12.1.6.4.1. Rotor Simulation with Fixed Pitch

1. To re-read the case and solution files, go to:

   **File→ Read→ Case & Data...**

   and read the `VBM_helicopter_tutorial.cas` and `.dat` files.

2. To compare the results along the top and bottom sides of the fuselage, using the XY plot panel, go to:

   **Results→ Plots→ XY Plot→ New...**

   a. Deselect the **Position** on **X Axis** in the **Options** section.

   b. Select **Pressure** and then **Pressure Coefficient** from the **Y Axis Function** pull-down menu.

   c. Select **Custom Field Functions...** and then x-norm from the **X Axis Function** pull-down menu.

   d. Select **clip-bottom** and **clip-top** in the **Surfaces** list.

   e. Click the **Load File...** button and select the files: `xnr_phi0.xy` and `xnr_phi180.xy` containing the experimental data, then click **OK**.

   f. Click the **Save/Plot** button.

   It is possible to change curve style. Click **Curves** in the **Solution XY Plot** dialog box; then select the **curve #** and define the **Line Style** and **Marker Style** from the options in the **Curve – Solution XY Plot** dialog box.

3. To compare the results along the port and starboard sides of the fuselage:

   a. Select the two files in the **File Data** list and click the **Free Data** button.

   b. Deselect the **clip-top** and **clip-bottom** entities in the **Surfaces** list. Select **clip-port** and **clip-starboard** in the **Surfaces** list.

   c. Click the **Load File...** button and select the `xnr_phi090.xy` and `xnr_phi270.xy` files containing the experimental data, the click **OK**.

   d. Click on the **Save/Plot** button.

Figure 31.36: Pressure Coefficient Distribution Along the Top and Bottom of the Fuselage (p. 388) shows the comparison of the numerical and experimental pressure coefficient along the top and bottom of the fuselage. Figure 31.36: Pressure Coefficient Distribution Along the Top and Bottom of the Fuselage (p. 388) shows the comparison of the numerical and experimental pressure coefficient distribution along the port and starboard sides of the fuselage.

**Figure 31.36: Pressure Coefficient Distribution Along the Top and Bottom of the Fuselage**

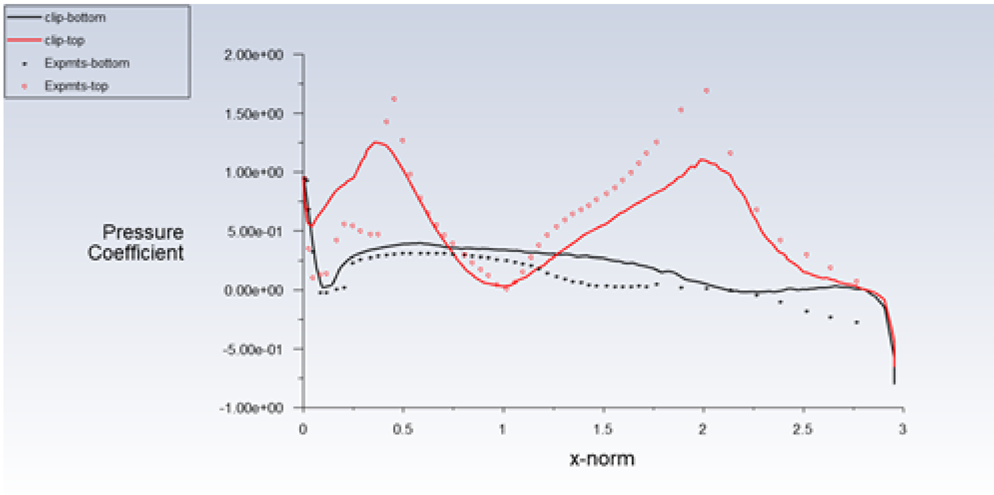
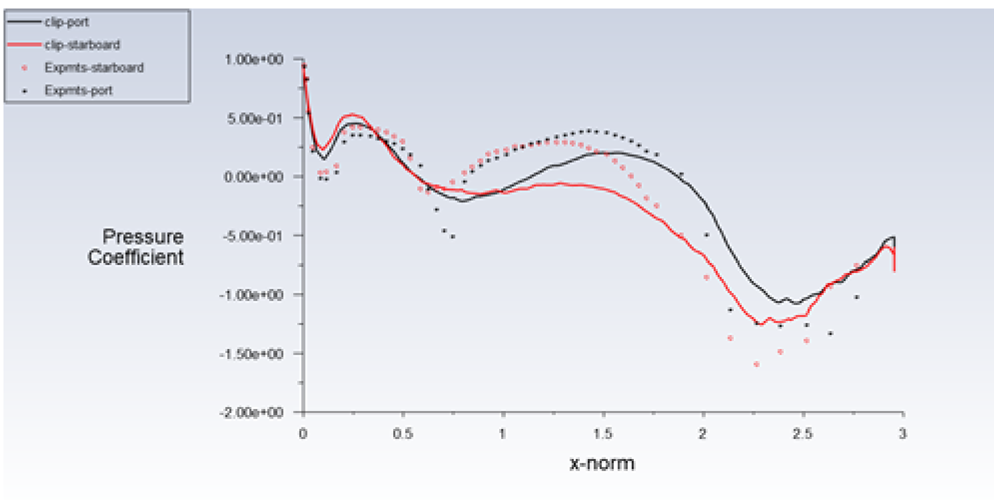



**Figure 31.37: Pressure Coefficient Distribution Along the Port and Starboard Sides of the Fuselage**




### 31.12.1.6.4.2. Rotor Simulation with Collective and Cyclic Trimming

The same procedures can be used to compare the numerical results with collective and cyclic trimming to the experimental pressure coefficient along the fuselage. Go to:

**File→ Read→ Case & Data...**

and re-read the `VBM_helicopter_tutorial_Co_Cy.cas` and `.dat` files, then follow the same procedure outlined above.

Figure 31.38: Pressure Coefficient Distribution Along the Top and Bottom of the Fuselage Obtained with Collective and Cyclic Trimming (p. 389) shows the comparison of the numerical and experimental pressure coefficient along the top and bottom of the fuselage. Figure 31.39: Pressure Coefficient Distribution Along the Port and Starboard Sides of the Fuselage Obtained with Collective and Cyclic Trimming (p. 389) shows the comparison of the numerical and experimental pressure coefficient along the port and starboard sides of the fuselage.

**Figure 31.38: Pressure Coefficient Distribution Along the Top and Bottom of the Fuselage Obtained with Collective and Cyclic Trimming**

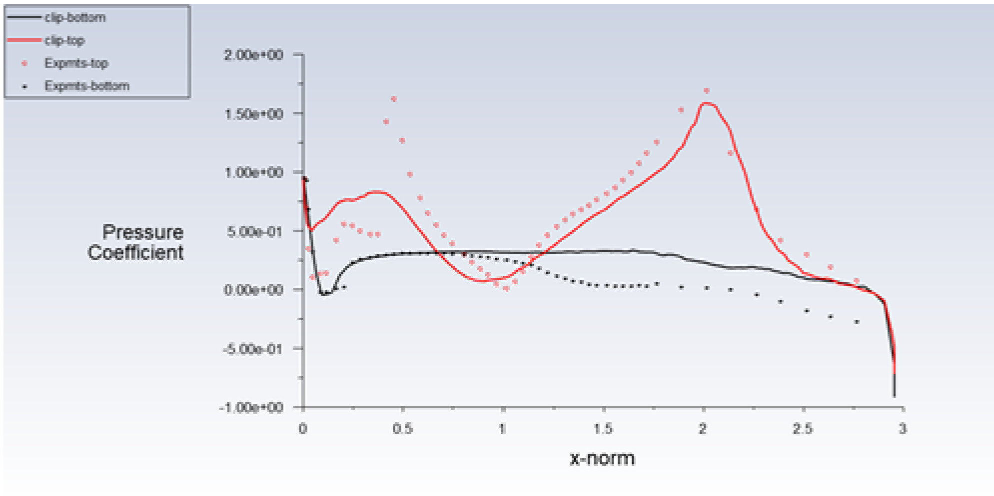
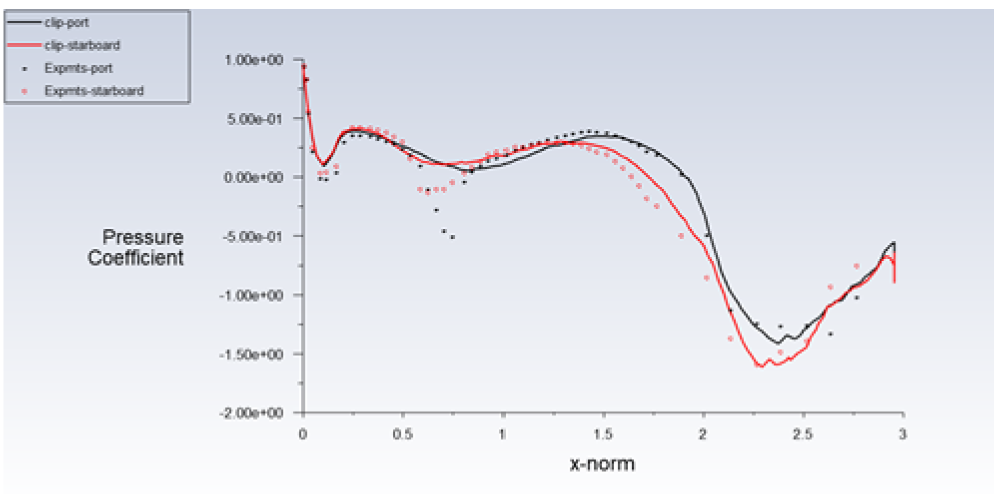



**Figure 31.39: Pressure Coefficient Distribution Along the Port and Starboard Sides of the Fuselage Obtained with Collective and Cyclic Trimming**




Some improvements in the agreement between the experimental and numerical results with collective and cyclic trimming are seen at the top of the fuselage. The numerical results on the port and starboard side have also improved considerably.

### 31.12.1.7. Summary

This tutorial explains the usage of Fluent's VBM for a generic helicopter case and compares the numerical results with fixed blade pitch and with collective and cyclic trimming to the published experimental data.

Step-by-step instructions for the Fluent and VBM set up are provided to enable you to conduct realistic rotorcraft simulations.

The comparison of the fixed-pitch results with the collective trimming case implicitly demonstrates that the VBM trimming routine is stable and accurate.

Considering the coarseness of the grid, which has been tailored towards portability, simplicity and fast turnaround of the tutorial runs, rather than accuracy, a very reasonable agreement between the results of the simulation obtained with collective and cyclic trimming and the experimental data has been demonstrated.

#### 31.12.1.8. References


1. Liou, S.G., Komerath, N.M. and McMahon, H.M. "Velocity Measurements of Airframe Effects on a Rotor in Low-Speed Forward Flight", AIAA Journal of Aircraft, vol. 26, no. 4, 1989.

2. Glauert, H. "The Elements of Aerofoil and Airscrew Theory," Second Edition, Cambridge University Press, New York, USA, 1947.

3. Stepniewski, W.Z. and Keys, C.N., "Rotary-Wing Aerodynamics," Dover Publications Inc., New York, USA, 1984.

4. Leishman, J.G., "Principles of Helicopter Aerodynamics," Cambridge University Press, New York, USA, 2000.

5. Zori, L.A.J., Rajagopalan, R.G., "Navier-Stokes Calculation of Rotor-Airframe Interaction in Forward Flight", Journal of the American Helicopter Society, Vol.40, April 1995.

6. Brand, A.G., Komerath, N. and McMahon, H., "Results from Laser Sheet Visualization of a Periodic Rotor Wake", AIAA Journal of Aircraft, vol. 26, no. 5, 1989.

7. Liou, S.G., Komerath, N.M. and McMahon, H.M. "Velocity Field of a Cylinder in the Wake of a Rotor in Forward Flight", AIAA Journal of Aircraft, vol. 27, no. 9, 1990.

8. Liou, S.G., Komerath, N.M. and McMahon, H.M. "Measurements of the Interaction Between a Rotor Tip Vortex and a Cylinder," AIAA Journal, vol. 28, no. 6, 1990.

9. D.N. Mavris, Komerath, N.M. and McMahon, H.M. "Prediction of Aerodynamic Rotor-Airframe Interactions in Forward Flight", Journal of the American Helicopter Society, October 1989.

10. Brand, A.G., Komerath, N.M. and McMahon, H.M., "Windtunnel Data From a Rotor Wake/Airframe Interaction Study", Georgia Institute of Technology, US Army Research Contract No. DAAG 29-82-K-0094, 1986.


### 31.12.2. Fluent's Virtual Blade Model Propeller Tutorial

Although the VBM was written specifically to simulate helicopter rotor flow/airframe interaction, it can easily handle any type of rotating component, such as aircraft and marine propellers, axial wind turbines, etc. This tutorial provides guidelines for setting up and solving a simple propeller simulation using Fluent's Virtual Blade Model (VBM). The physical propeller is replaced with an actuator disk of finite thickness that provides the framework to simulate the thrust and torque of the actual propeller using the momentum source terms in Fluent's governing equations. Local flow characteristics in the actuator disk are extracted from the 3D flow solution generated by Fluent and used by the VBM to compute the forces acting on each blade section from airfoil look-up tables, then applied to the cells composing the actuator disk. The unsteady propeller problem is replaced with a much simpler time-averaged procedure that can be used very effectively for the initial design of a real propeller-driven aircraft. Although the mesh is very coarse, the purpose of this tutorial is to illustrate the methodology for conducting this type of simulation, rather than as a validation of the Virtual Blade Model that should be conducted with much finer grids.

#### 31.12.2.1. Introduction

This tutorial illustrates the methodology for configuring and solving flow on a 6-bladed aircraft propeller using Fluent's Virtual Blade Model (VBM). The VBM is an intermediate method for modeling the time-averaged cumulative effects of the rotating blades, between the simpler FAN and the more complex Single or Multiple Rotating Frames (SRF/MRF) approaches.

The VBM replaces the rotating blades with analogues composed of stacks of 2D airfoil sections. The aerodynamic properties of the airfoil sections, in the form of tables of lift and drag coefficients as functions of the angle of attack, are used to model the effect of the blades according to the Blade Element Theory (BET). Hence, the rotational velocity, local blade angle, twist and chord on the airflow passing through the rotor disk can be included in the overall Fluent simulation of the aircraft/rotorcraft. The Fluent VBM is significantly more accurate than the FAN model and its results compare favorably to the SRF/MRF model, without the very large cell count required to mesh the complete blades with sufficiently fine resolution.

The VBM introduces the forces generated by the blades on the working fluid via momentum sources applied in the grid cells attached to the propeller disk, also called the actuator disk, allowing the pressure jump across the disk to vary with radius and azimuth. This eliminates the need to generate detailed meshes over each of the propeller blades, significantly reducing grid size, computational effort and mesh generation time. The magnitude of the momentum sources is computed according to the BET, allowing for the variation of twist angle, chord and airfoil types along the span. The non-linear aerodynamic interaction between the propeller and other structural components is solved by introducing the VBM into Fluent through a User-Defined Function (UDF). The airfoil $C_L$ and $C_D$ tables required by the BET can also be functions of Mach and Reynolds number, yielding greater fidelity for both incompressible and compressible flow simulations. The propeller disk pitch and bank angle with respect to the natural orientation of the actuator disk can also be considered. Finally, the blade pitch angle can be updated automatically by a trimming routine to match the specified thrust coefficient.

Although the VBM has been specifically developed for rotorcraft simulations, it is also applicable to general rotating machinery (propellers, wind power, HVAC, automotive, marine, etc.), for flows typically characterized by:

- Low-to-moderate blade loading
- Predominantly axial flow
- Negligible geometrical blockage

This tutorial will cover the following topics:

- Loading VBM add-on module in Fluent.
- VBM meshing requirements.
- Setting up and running a test case using the VBM.
- Running a fixed-pitch and thrust-trimmed simulation.

- Post-processing the two sets of results and interpreting the differences.

The results shown in this tutorial have been obtained on 10 cores of a 12-core Intel® Xeon® Gold 6136 CPU @ 3.00 GHz running Windows 10 Enterprise, version 10.0.18362 build 18362.

### 31.12.2.2. Problem Description

This tutorial will demonstrate the simulation of a simple propeller modeled by an actuator disk, as shown in Figure 31.40: Simple Propeller, Modeled by an Actuator Disk (Green), Inside a Cylindrical Domain (p. 393). The geometric and operating conditions are listed in Table 31.7: Propeller Geometric Data and Operating Conditions (p. 392).

**Table 31.7: Propeller Geometric Data and Operating Conditions**

| **Geometric Data** | |
|---|---|
| Tunnel dimensions | 50 x 30 m |
| Tunnel turbulence intensity | 1% |
| Rotor blades | 6 |
| Rotor radius | 0.5 m |
| Cutout radius | 0.15 m |
| Hinge offset | 0 m |
| Blade section | NACA 16016 (6% camber) |
| Blade root chord | 0.1 m |
| Blade tip chord | 0.05 m |
| Disk pitch angle | -90° |
| Disk bank angle | 0° |
| Collective pitch angle | 61.85345° (fixed) |
| Coning angle | 0° |
| Longitudinal flapping angle | 0° |
| Lateral flapping angle | 0° |
| Blade twist | -35.12931° |

| **Operating Conditions** | |
|---|---|
| Design advance ratio (*Jd*) | 1.4208 |
| Reference pressure | 97013.91 Pa |
| Reference temperature | 284.5926 K |
| Reference density | 1.187584 kg/m$^3$ |
| Inflow velocity (*Vx*) | 84.54519 m/s |
| Rotational speed | 3570.321 rpm<br>(373.8831 rad/s) |
| Tip velocity | 186.94155 m/s |

The conditions in Table 31.7: Propeller Geometric Data and Operating Conditions (p. 392) were computed using the following definition of the advance ratio.

$$J=\frac{V_x}{nd}=\frac{2\pi V_x}{\Omega d}$$

where

$d$ – propeller diameter (m)

$n$ – rotational speed (rev/s)

$\Omega$ – rotational speed (rad/s)

$V_x$ – propeller axial flow velocity (m/s)

**Figure 31.40: Simple Propeller, Modeled by an Actuator Disk (Green), Inside a Cylindrical Domain**

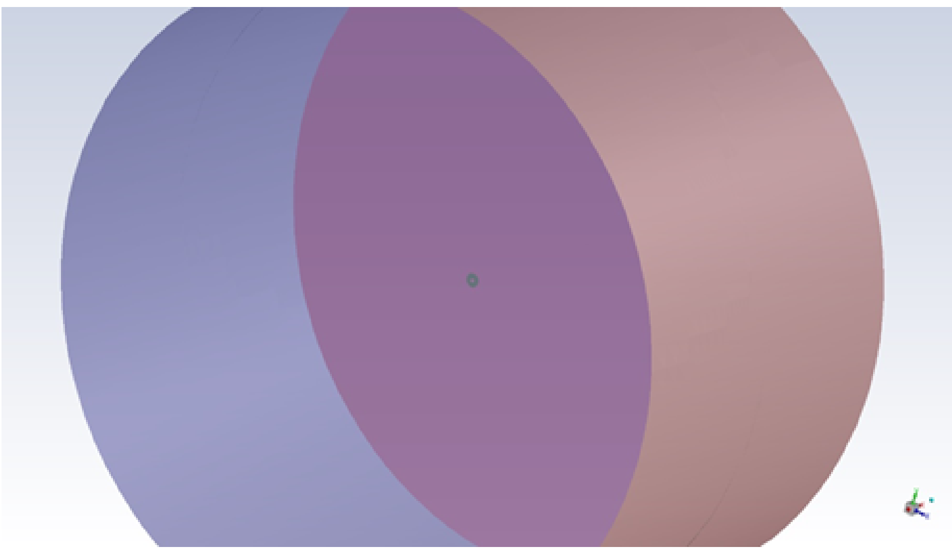


**Table 31.8: Geometric Characteristics of the Propeller Blade**

| Section | r/R | Chord (m) | Twist (deg) | Airfoil |
|---|---|---|---|---|
| 0 | 0.30 | 0.100000 | 0.00000 | naca16016 |
| 1 | 0.40 | 0.098922 | -6.55173 | naca16016 |
| 2 | 0.50 | 0.095905 | -12.11207 | naca16016 |
| 3 | 0.60 | 0.090733 | -17.06897 | naca16016 |
| 4 | 0.70 | 0.083621 | -21.16379 | naca16016 |
| 5 | 0.80 | 0.074353 | -24.91379 | naca16016 |
| 6 | 0.90 | 0.063308 | -28.53448 | naca16016 |
| 7 | 0.95 | 0.056519 | -30.99138 | naca16016 |
| 8 | 1.00 | 0.050000 | -35.12931 | naca16016 |

The geometric characteristics of the propeller blade are listed in Table 31.8: Geometric Characteristics of the Propeller Blade (p. 393). Figure 31.41: Radial Distribution of Blade Chord and Twist (p. 394)

shows that the radial distributions of chord and twist are non-linear. Figure 31.42: Lift and Drag Coefficients of the Modified NACA 16016 Airfoil (p. 394) shows that the active range of the aerodynamic coefficients of the modified NACA 16016 airfoil is limited to ±8˚. However, the VBM requires that they be provided over the full ±180˚ range. At the end of the simulation, it will be good practice to verify that the angle of attack has not exceeded the ±8˚ range across the entire actuator disk.

**Figure 31.41: Radial Distribution of Blade Chord and Twist**

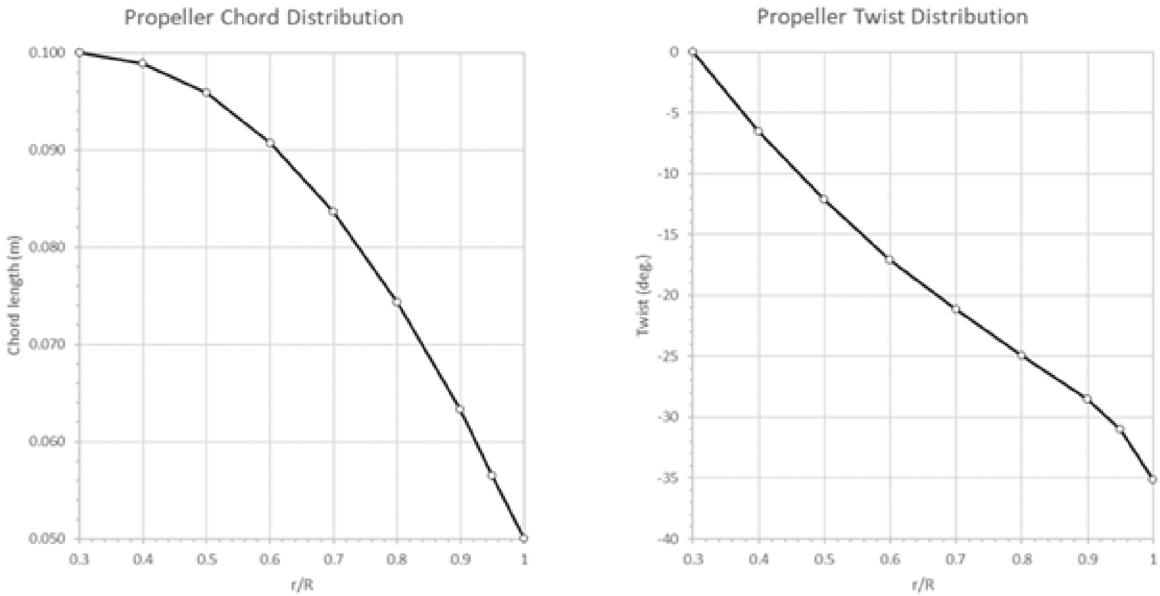
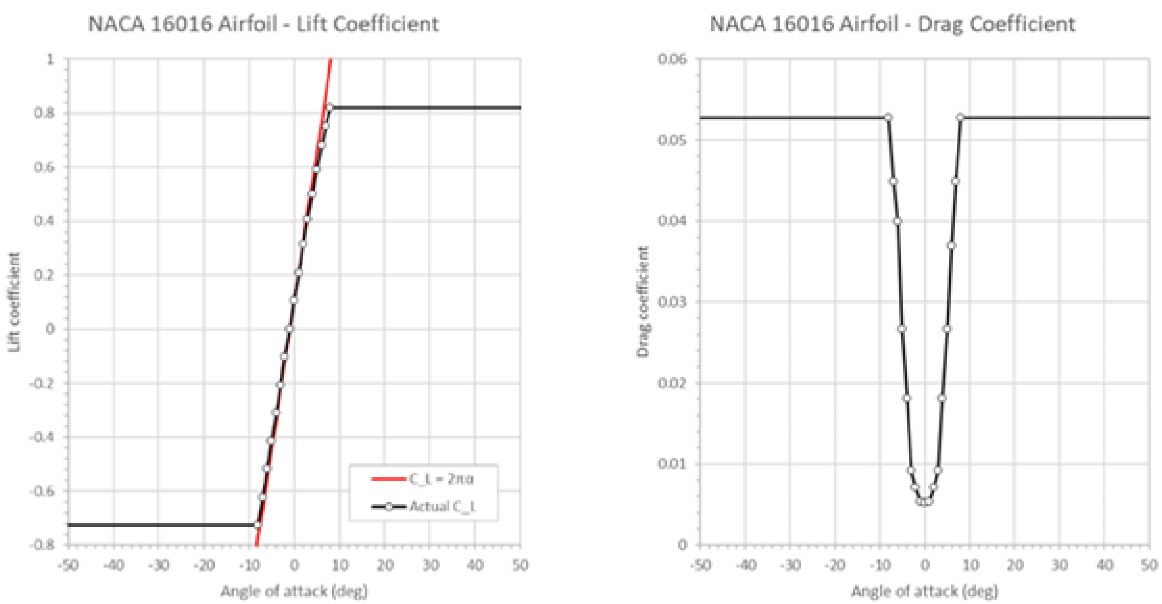



**Figure 31.42: Lift and Drag Coefficients of the Modified NACA 16016 Airfoil**




Two operating cases will be considered:

- Fixed pitch
- Blade pitch angle trimming (collective) - reverse simulation

Download the `vbm_propeller_tutorial.zip` file here.

Unzip `vbm_propeller_tutorial.zip` to your working directory.

The following files are required for the simulation and should be present in the working directory:

`naca16016.dat`

`VBM_propeller_tutorial.msh.gz`

---

**Note:**

The coefficient of the dynamic pressure in the thrust and moment coefficients follows the North-American practice in the addon module (set to 1.0).

---

### 31.12.2.3. Setting up the Calculation

Launch the Fluent executable, then:

1. Select the **3D**, **Double Precision** and **Parallel (Local Machine)** options.
2. Choose a suitable number of **Processes**, or only one if only a single CPU is available.
3. Select the appropriate **Working Directory** in the **General Options** panel.
4. Make sure that all the files listed at the end of Problem Description (p. 392) are located in the **Working Directory**.
5. Click **Start**.

#### 31.12.2.3.1. Reading the Grid

1. To read the mesh file, go to:

   **File** → **Read** → **Mesh...**

   Select the `VBM_propeller_tutorial.msh.gz` file, and click **OK**.

2. To display the mesh, click on the following button in the ribbon.

   **Domain** → **Display...**

   Deselect **inflow**, **outer** and **outflow** in the **Surfaces** list and select **int_acdisk** and click on **Display**.

   The computational domain must be divided into two separate but communicating fluid domains (**Physics** → **Zones** → **Cell Zones**). The VBM only acts on the cells that form the **disk** cell zone attached to one side of the actuator disk.

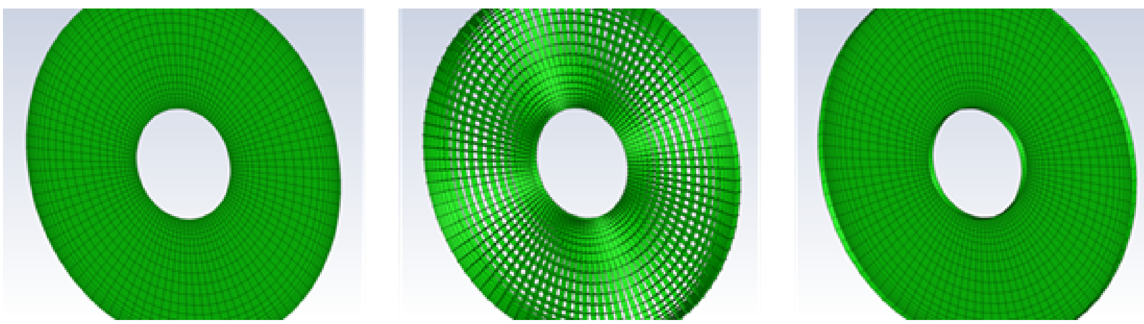
   As shown in Figure 31.43: From Left to Right, the Three Components of the Disk Cell Zone: Int_Acdisk, Int_Disk and Int_Disk:003 (p. 396), Fluent automatically subdivides the disk cells into three separate entities:

- **int_acdisk** contains the internal single-sided cell faces lying on the actuator disk surface.
- **int_disk:003** contains the cell faces shared by the **live** and **disk** domains, excluding the actuator disk.
- **int_disk** contains the internal cell faces perpendicular to the disk.

3. To display the first two surfaces, select each in the list of **Surfaces**, and click **Display**.
4. For the third surface, however, click on **Adjacency** in the **Mesh Display** dialog box to open the **Adjacency** dialog box; select disk from the list of **Cell Zone(s)** and **int_disk** from the list of the **Adjacent Face Zones** and then click on the **Display Face Zones**.

   If this arrangement is not respected, the VBM will not work properly.

**Figure 31.43: From Left to Right, the Three Components of the Disk Cell Zone: Int_Acdisk, Int_Disk and Int_Disk:003**



The following mesh topology requirements must be respected:

- The entire 360˚ azimuth of the propeller disk must be modeled.
- The actuator disk surface (**int_acdisk**) must have the interior boundary condition.
- The cells attached to one side of the disk must be marked as a separate domain (**disk**).
- These cells must have one complete face attached to the disk. Only hexa and prisms are allowed.
- A continuum fluid zone (**live**) must completely envelop the rotor zone (**disk**).
- The fluid zone and rotor zone must have different BC index (10 for the fluid zone and 1 for the rotor zone).

### 31.12.2.3.2. Loading VBM Addon Module

The VBM add-on module is a beta feature and installed in a directory called `addons/vbm`. Therefore, it is required to enable the feature through the text user interface (TUI).

1. Enter the following TUI command in the console to enable the feature:

   ```
   (enable-feature 'vbm)
   ```

2. Enter the following TUI command in the console to load the VBM addon:

```
Define/models/addon-module 12
```

---

**Note:**

The VBM module can be loaded only when a valid Ansys Fluent mesh or case file has been set or read.

VBM module is the 12th module listed in the console.

During the loading process, a Scheme library containing the graphical and text user interface, and a UDF library containing a set of user-defined functions (UDFs) for the VBM addon module are automatically loaded into Ansys Fluent.

---

3. Answer **yes** when asked to setup UDF hooks in the console. This will setup the UDM to 15 and set the Function Hooks.

   The loading processes are reported in the Fluent console and UDM offset should be equal to 0 at the end. Once the addon is loaded into Ansys Fluent, the Virtual Blade Model appears under the Models tree branch.

### 31.12.2.3.3. Setup Units

Since this tutorial is in the SI system of units, and the rotor disk, blade pitch and blade flapping angles are provided in degree, go to:

**Domain** → **Mesh** → **Units...**

1. Click the **si** tab.
2. Click **angle** in the **Quantities** list and choose **deg** in the **Units** list.
3. Close the **Set Units** dialog box.

### 31.12.2.3.4. Cell Zone Conditions

In the **Physics** section of the Fluent ribbon, go to:

**Zones** → **Cell Zones**

1. Select **disk** in the **Cell Zone Conditions** section in the **Task Page** and click on the **Edit...** button
2. Ensure that the **Zone Name** is **disk** and the **Material Name** is **air**.
3. Tick mark the **Source Terms** box and click the **Source Terms** button in the ribbon just below.
4. Enable the **X Momentum** source term by clicking on the **Edit...** button. The **X Momentum Sources** window will open.
5. Set the **Number of X Momentum** sources to **1** with the up/down arrows or by entering the number in the box. Click on the down arrow just below and select the **udf xmom_src_1::vbm** option from the pull-down menu and press the **OK** button.

6. Repeat the operation above to configure the **Y Momentum** and **Z Momentum** sources with **udf ymom_src_1::vbm** and **udf zmom_src_1::vbm**, respectively. The confirmation message **1 source** should appear in the boxes next to the **X**, **Y** and **Z** Momentum sources.

7. Click **Apply** and **Close** the **Fluid** dialog box.

#### 31.12.2.3.5. Operating Conditions

To set the operating conditions, go to:

**Physics** → **Operating Conditions...**

1. Set a value of `97013.91` Pascal in the **Operating Pressure** box.

2. Set a value of `-1` m in the **X** component of the **Reference Pressure Location** section.

3. Click **OK**.

#### 31.12.2.3.6. Physical Modeling

1. To configure the Fluent solver settings, go to:

   **Physics** → **General**

   and select the following options in the **General** task page:

   **Time** → **Steady**

   **Type** → **Pressure-Based**

   **Velocity Formulation** → **Absolute**

2. To enable the energy equation, go to:

   **Physics** → **Models**

   and tick-mark the **Energy** equation box.

3. To select the turbulence model, go to:

   **Physics** → **Models** → **Viscous...**

   a. Select the **Spalart-Allmaras (1-eqn)** turbulence model.

   b. Enable the **Strain/Vorticity-Based** production option.

   c. Tick mark the **Curvature Correction** in the **Options** section.

   d. Click **OK** to accept all the other default settings and close the **Viscous Model** dialog box.

For vorticity-dominated flows, the default **Vorticity-Based**-production option overpredicts the production of eddy viscosity in the vortex cores. Adding the strain tensor to the vorticity reduces the production of turbulent viscosity in regions where the measure of vorticity exceeds that of strain rate.

#### 31.12.2.3.7. Materials

This simulation features a high-speed flow regime, hence compressibility must be enabled. Go to:

**Physics** → **Materials** → **Create/Edit...**

1. Select **air** as the working fluid in the **Fluent Fluid Materials** pull-down menu.

2. Select **ideal-gas** in the **Density** pull-down menu.

3. Click the **Change/Create** button.

4. Close the **Create/Edit Materials** dialog box.

---

**Note:**

The VBM also works with the **constant** density and **incompressible-ideal-gas** options, however since rotors usually operate in the compressible regime, the **ideal-gas** option is more appropriate.

---

#### 31.12.2.3.8. Boundary Conditions

To configure the boundary conditions, go to:

**Physics** → **Zones** → **Boundaries**

1. Set the boundary condition at the inlet.

   a. Select the **inflow** boundary in the **Task Page**, ensure that its **Type** is **velocity-inlet** and click on the **Edit...** button.

   b. In the **Momentum** panel of the ribbon, select the **Components** option from the **Velocity Specification Method** pull-down menu.

   c. Select **Absolute** in the **Reference Frame** pull-down menu.

   d. Input the values (`84.54519`,`0`,`0`) m/s for the **X-**, **Y-** and **Z-Velocity** components, respectively.

   ---

   **Note:**

   The axial velocity is calculated using the freestream Mach number (from Table 31.7: Propeller Geometric Data and Operating Conditions (p. 392)).

   ---

   e. In the **Turbulence** section, select the **Turbulent Viscosity Ratio** option from the **Specification Method** pull-down menu and set the **Turbulent Viscosity Ratio** to **1**.

   f. In the **Thermal** panel of the ribbon, set a **Temperature** value of `284.5926` K.

   g. Click **Apply** and close the **Velocity Inlet** dialog box.

2. Set the boundary condition at the outlet.

   a. Select the **outflow** boundary, ensure that the **Type** is **pressure-outlet**, then click on the **Edit...** button.

   b. Select **Absolute** in the **Backflow Reference Frame** pull-down menu.

   c. Set the **Gauge Pressure** value to **`0`** Pascal.

   d. In the **Turbulence** section, select the **Modified Turbulent Viscosity** option in the **Specification Method** pull-down menu.

   e. Set a value of **`0.0001`** ($m^2/s$) in the **Backflow Modified Turbulent Viscosity** box.

   f. There is very little chance that backflow may occur, however it is good practice not to skip this operation.

   g. In the **Thermal** panel of the ribbon, set the **Backflow Total Temperature** to **`288.15`** K.

   h. Click **Apply** and close the **Pressure Outlet** dialog box.

3. Set the **Wall** boundary conditions.

   a. Select the **Outer** boundary and click on the **Edit...** button.

   b. In the **Momentum** panel of the ribbon, set the **Shear Condition** to **Specified Shear** with {**`0`**,**`0`**,**`0`**} **X-**, **Y-** and **Z-Components**, respectively, and click the **OK** button.

   c. In the **Thermal** panel of the ribbon, select **Heat Flux** in the **Thermal Conditions** and ensure that the **Heat Flux** value is **`0`** $W/m^2$.

   d. Click **Apply** and close the **Wall** dialog box.

### 31.12.2.3.9. Reference Values

To set the reference values, go to:

**Physics** → **Reference Values...**

1. Select **inflow** from the **Compute from** pull-down menu.

2. Set the **Area** value to **`0.785398`** $m^2$ (disk area).

3. Set the **Length Value** to **`0.05`** m (blade tip chord).

4. Select **live** from the **Reference Zone** pull-down menu.

### 31.12.2.3.10. Discretization and Solution Controls

1. To set the discretization options, go to:

   **Solution** → **Solution Methods...**

   In the **Solution Methods** task page, use the pull-down menus to set the following options:

- **Scheme** → **Coupled**
- **Gradient** → **Green-Gauss Node Based***
- **Pressure** → **PRESTO!****
- **Density** → **Second Order Upwind**
- **Momentum** → **Second Order Upwind**
- **Modified Turbulent Viscosity** → **First Order Upwind**
- **Energy** → **Second Order Upwind**

---

**Note:**

*The node-based averaging scheme is more accurate than the default cell-based scheme, especially for unstructured meshes, and most notably for triangular and tetrahedral meshes.

**The **PRESTO!** pressure discretization scheme is recommended for rotating flows.

---

2. Retain the default solver parameters in:

   **Solution** → **Controls** → **Controls...**→ **Solution Controls**.

   .

### 31.12.2.3.11. Convergence Monitoring

1. To configure the residuals monitors that will appear in the Fluent graphics window and console, go to:

   **Solution** → **Reports** → **Residuals...**

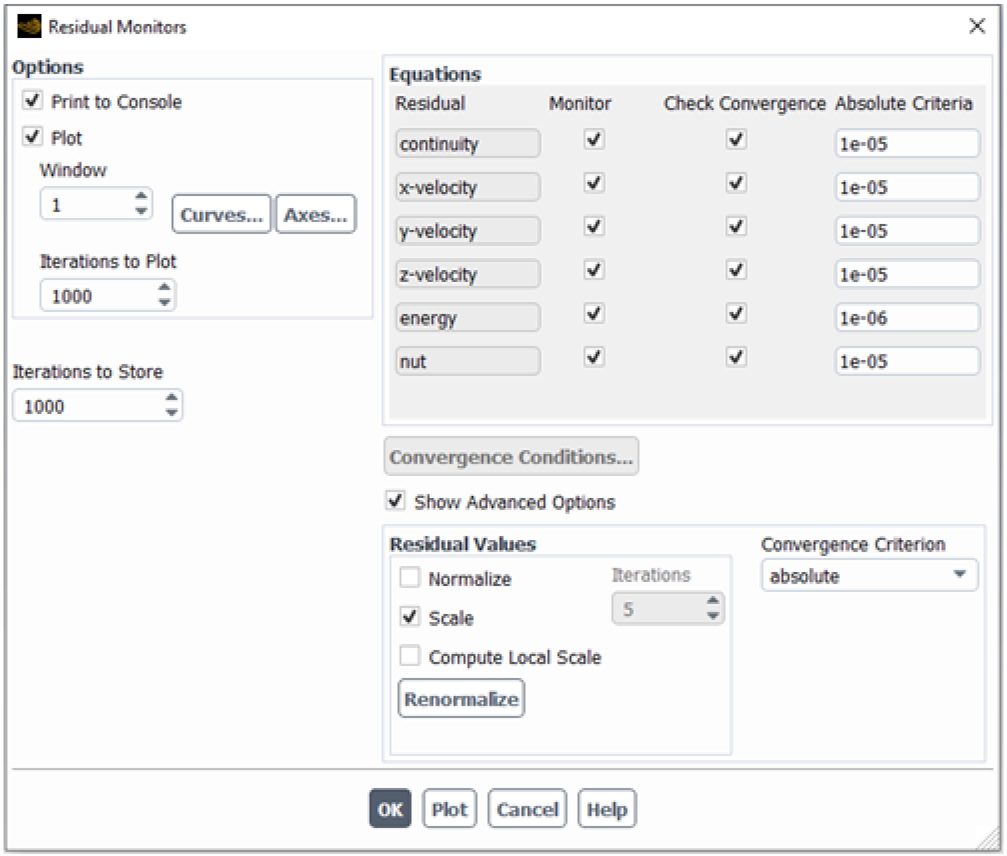
   a. Ensure that **Plot** and **Print to Console** options are enabled in the **Options** group box.

   b. Enable **Show Advanced Options** and select **absolute** from the **Convergence Criterion** drop-down list.

   c. Set the **Absolute Criteria** values to `1e-6` for the energy equation and to `1e-5` for the other equations as shown in Figure 31.44: Solution Residuals Configuration (p. 402).

   d. Click **OK** to close the **Residual Monitors** dialog box.

**Figure 31.44: Solution Residuals Configuration**




2. Additionally, you may want to monitor pressure convergence on the actuator disk **int_acdisk**. Go to:

   **Solution** → **Reports** → **Definitions** → **New...** → **Surface Report** → **Integral...**

   a. Enter `rotor-pressure` in the **Name** box.

   b. Tick-mark the **Report File**, **Report Plot** and **Print to Console** boxes in the **Create** section.

   c. Select **Pressure...** and **Static Pressure** with the pull-down menus in the **Field Variable** section.

   d. Select **int_acdisk** in the **Surfaces** section.

   e. Click **OK** to close the **Surface Report Definition** dialog box.

The pressure convergence history will be displayed on screen and written in the file `rotor-pressure-rfile.out`. Additionally, the convergence histories of the thrust, torque, power and moments are written to the `Rotor_1_Loads.dat` file.

The value of the pressure integral and the convergence monitors will appear in the graphics window. The ribbon at the top left of the graphics window can be used to change the view from the residual monitors to the pressure integral and vice-versa.

#### 31.12.2.3.12. Solution Initialization

Initialize the solution from inlet boundary values. Go to:

**Solution → Initialization**

1. Select **Standard** and click **Options...** to open the **Solution Initialization** task page, which provides access to further settings.

2. Select **inlet** from the **Compute from** drop-down list.

3. Set the **Reference Frame** to **Absolute**.

4. Click the **Initialize** button.

### 31.12.2.4. Rotor Inputs

The last step before running the case is the configuration of the physical parameters of the propeller. To set up rotor parameter, go to:

**Physics → Models → More...**

1. Select the **Virtual Blade Model** in the **Model** task page and click the **Edit...** button. This will open the VBM mini-graphical user interface.

2. Ensure **Number of Rotor Zones** and the **Active Rotor Zone** are set to **1**.

3. Enter the parameters shown in Figure 31.45: General Disk Data Configuration Window (p. 404).

4. Click on **int_acdisk** in the list of **Surfaces** to select the actuator disk surface.

5. Click on the **Geometry** button of the ribbon and enter the parameters shown in Figure 31.46: Geometry Configuration Window (p. 405).

6. Click **Change/Create** to save the settings.

7. Click **OK** to close the **Rotor Input** dialog box.

---

**Note:**

The **Change/Create** button must be clicked before moving on to the next rotor (when present) or before pressing the **OK** button. This sequence must always be respected, even if the mini-graphical user interface is re-opened to simply edit a parameter.

When the **OK** button is clicked, the **User-Defined → Execute on Demand** command is executed automatically.

The UDF will append the `.dat` suffix to the airfoil file names if it is omitted in the mini-graphical user interface.

Consult Fluent's Virtual Blade Model (p. 331) for more information on the geometry of the rotor disk and the effect of the parameters.

---

**Figure 31.45: General Disk Data Configuration Window**

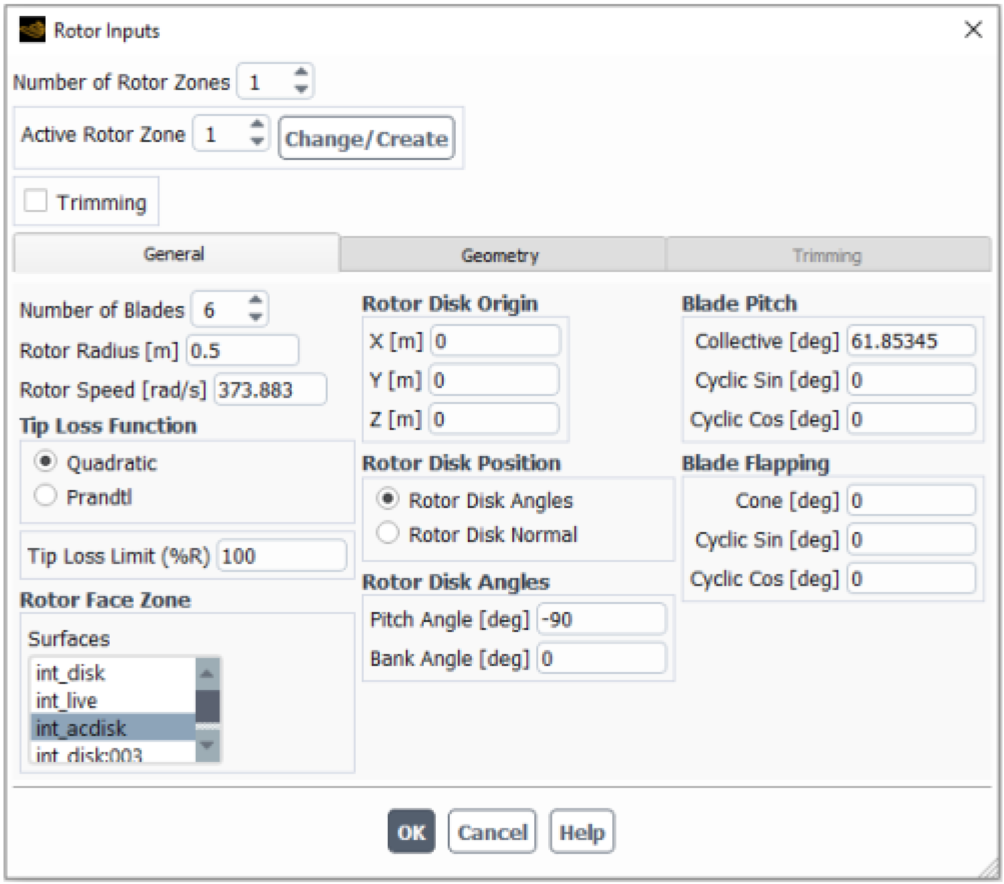
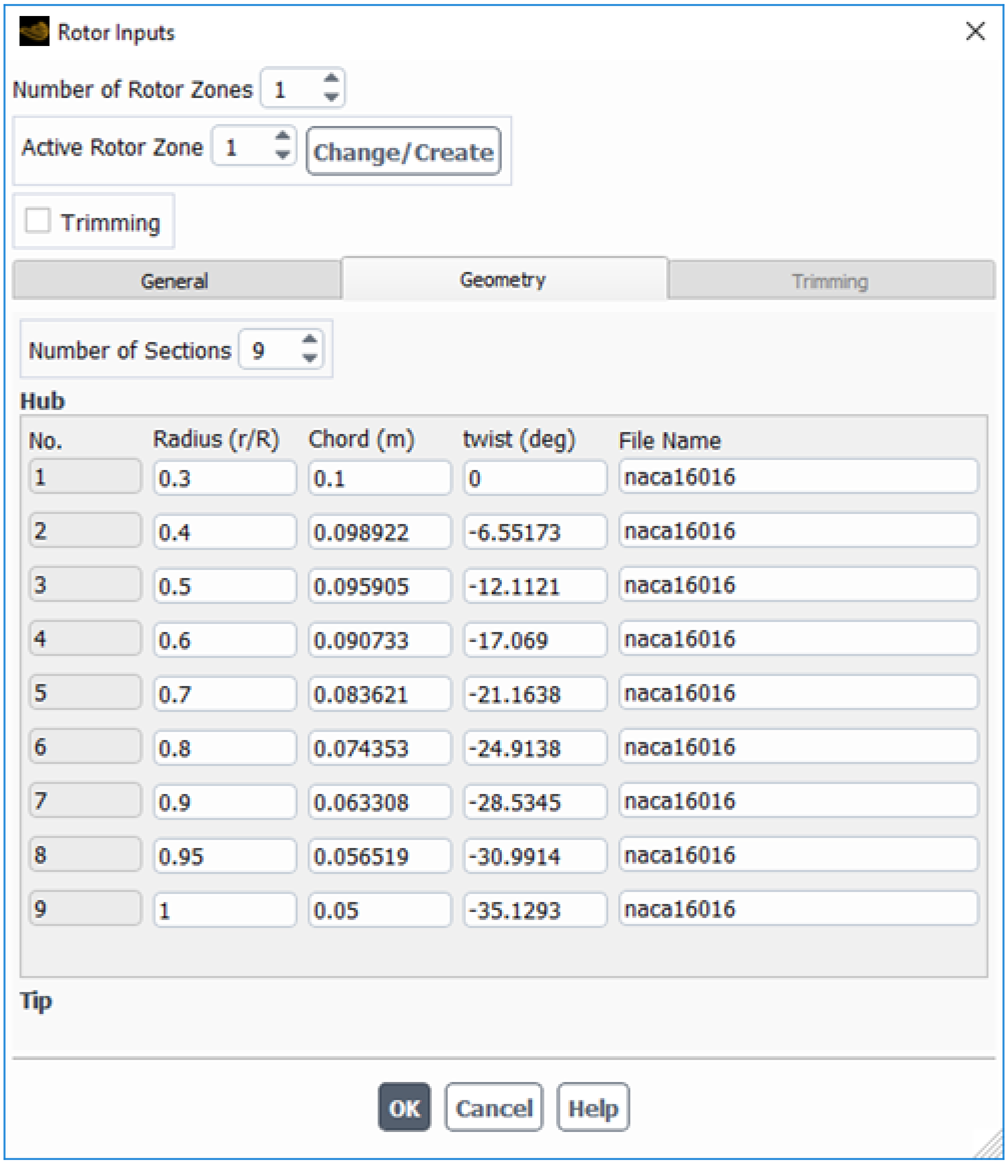


**Figure 31.46: Geometry Configuration Window**




## 31.12.2.5. Post-Processing

The following section explains how to create a framework for generating solution images. The post-processing set-up will be configured and saved before moving on to the actual solution process.

### 31.12.2.5.1. Cutting Plane for the Velocity Distributions

Create a cutting plane through the wake of the propeller to visualize the momentum changes produced by the actuator disk. The cutting plane will also enable a comparison of the effect of trimming on the solution. Go to:

**Results** → **Surface** → **Create** → **Plane...**

1. Tick-mark the **Point and Normal** box in the **Options** list. This will allow the creation of an unbounded cutting plane.

2. Enter the name **`plane-z=0`** in the **Contour Name** box.

3. Enter the coordinates of the origin of **int_acdisk** in the **Points** section: P(0) = {**0**,**0**,**0**}

4. Enter the components of the cutting plane **Normal** vector **n** = {**0**,**0**,**1**}

5. Click the **Create** button and close the **Plane Surface** dialog box

### 31.12.2.6. Solution

The two simulations can now be executed to demonstrate the capabilities of the VBM. Before starting the simulation, it is a good idea to save the work at this point. Go to

**File** → **Write** → **Case...**

and save the settings into the `VBM_propeller_tutorial.cas` file. Alternately, the **Case & Data...** option can be used.

#### 31.12.2.6.1. Propeller Simulation with Fixed-Pitch

1. For the first simulation, the propeller is operating in fixed-pitch mode without trimming. Go to:

   **Solution** → **Run Calculation** → **Run Calculation...**

   Set the **No. of Iterations** value to **600** and click on the **Calculate** button. The convergence history of the residuals of the governing equations is shown in Figure 31.47: Convergence History with Fixed Pitch (p. 406). Figure 31.48: Pressure Monitor Convergence History with Fixed Pitch (p. 407) shows the convergence history of the integral of the static pressure on the **int_acdisk** surface.

   **Figure 31.47: Convergence History with Fixed Pitch**

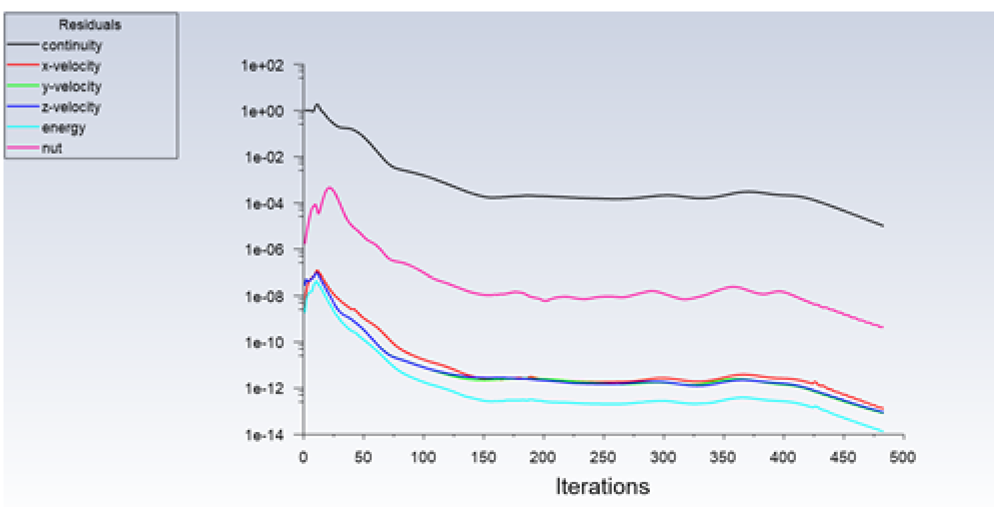
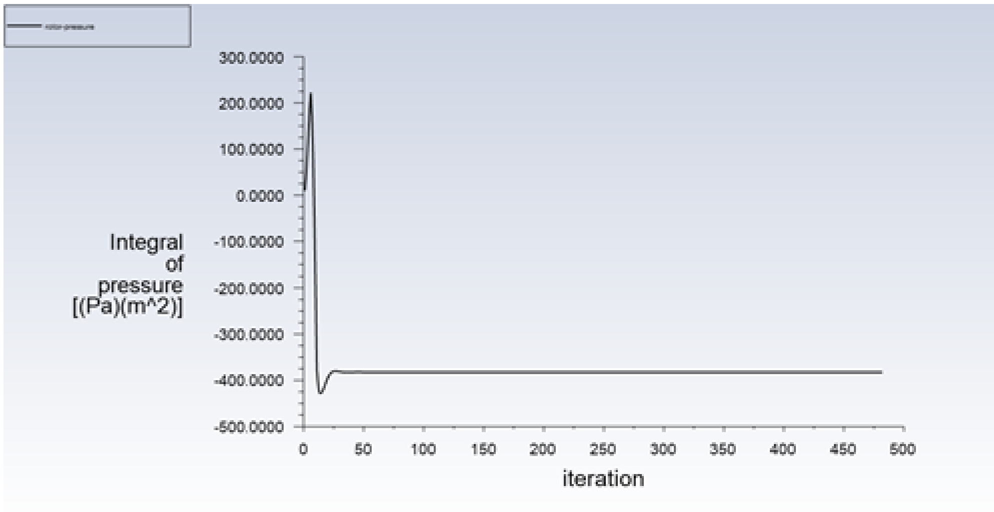


**Figure 31.48: Pressure Monitor Convergence History with Fixed Pitch**




---

**Note:**

In case the simulation stops with the following error message, `"The fl process could not be started"`, proceed with the following steps from now on:

- Exit Fluent and re-launch it using a different number of CPUs. Sometimes a computer restart is required.
- Read the case and data files which were saved in the previous step.
- Verify if the UDF library is Loaded and linked properly into Fluent (Loading VBM Add-on Module (p. 366) and Loading VBM Addon Module (p. 396)).
- Verify **Rotor Inputs** (Rotor Inputs (p. 403)), click **Change/Create** and then **OK**.
- Verify all other settings and continue.

---

2. Save the Fluent case and data files (`VBM_propeller_tutorial_FP.cas` and `.dat`):

   **File → Write → Case & Data...**

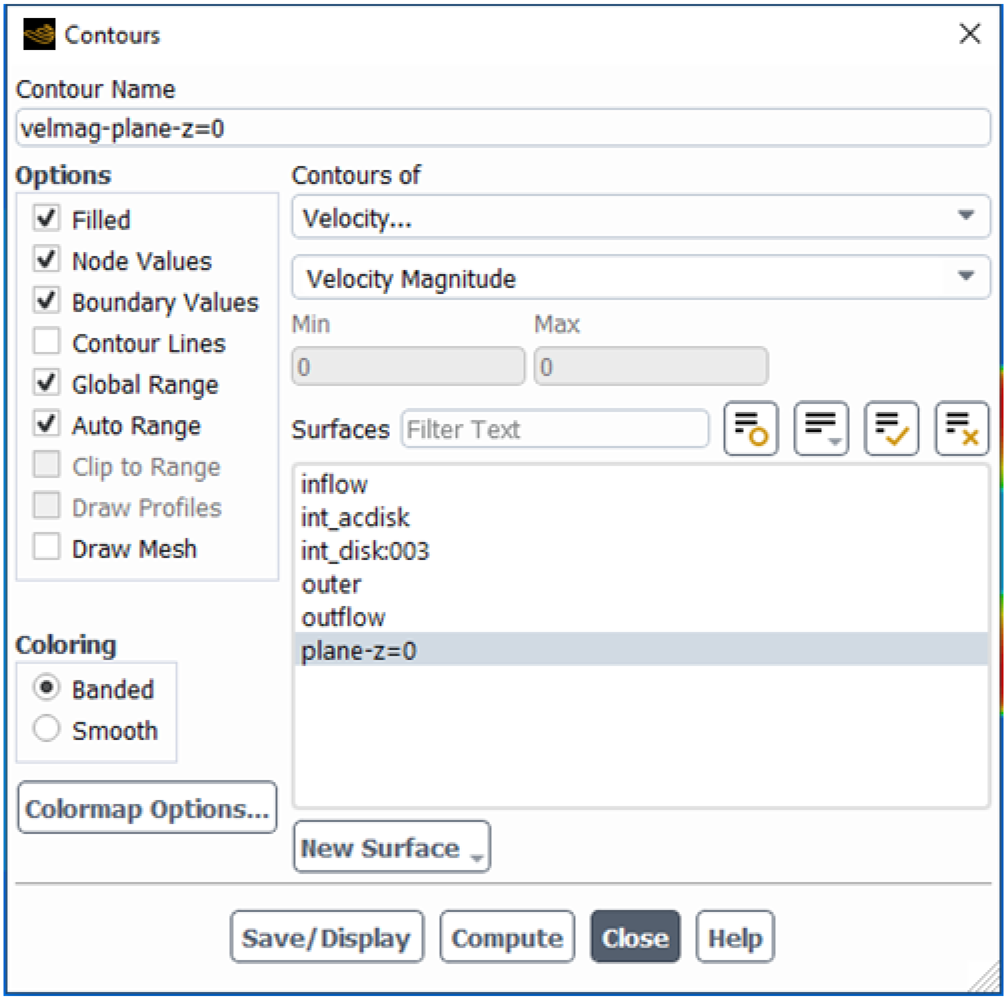
3. To display contours of flow velocity on the cutting plane that has been created in Cutting Plane for the Velocity Distributions (p. 405), go to:

   **Results → Graphics → Contours → New ...**

   a. Configure the window as shown in Figure 31.49: Display the Velocity Distribution on the Z = 0 Cutting Plane (p. 408).

   b. Click on the **Save/Display** button to display the image in the graphics window.

**Figure 31.49: Display the Velocity Distribution on the Z = 0 Cutting Plane**




c. Click the Blue z-axis arrow in the axis triad and zoom in to see the velocity contours on plane-z=0 near the actuator disk, as shown in Figure 31.50: Velocity Magnitude Distribution Around the Actuator Disk; Z = 0 Cutting Plane (p. 409).

4. To display the UDM data (see Output Data under Fluent's Virtual Blade Model within the Fluent Customization Manual), go to:

   **Results** → **Graphics** → **Contours** → **New ...**

   a. Ensure **Node Value** is deselected in the **Options** group box.

   ---

   **Note:**

   The UDM variables are cell-based, therefore the **Node Value** check box must be de-selected in order for the values to display correctly. Re-orient the viewing position along the positive x-axis.

   ---

   b. Select **int_acdisk** in the list of **Surfaces**, then select the **User-Defined Memory...** and desired options from the **Contours of** pull-down menus.

c. Click the **Save/Display** button.

The contours of **Angle of Attack (AoA)** are shown in Figure 31.51: Angle of Attack Distribution on the Actuator Disk (p. 409). This is a useful visual verification that the angle of attack is within the active region of the aerodynamic properties of the airfoil section (see Problem Description (p. 392)). Figure 31.52: Blade Pitch Angle Distribution on the Actuator Disk (p. 410) shows the radial distribution of the blade pitch angle.

**Figure 31.50: Velocity Magnitude Distribution Around the Actuator Disk; Z = 0 Cutting Plane**

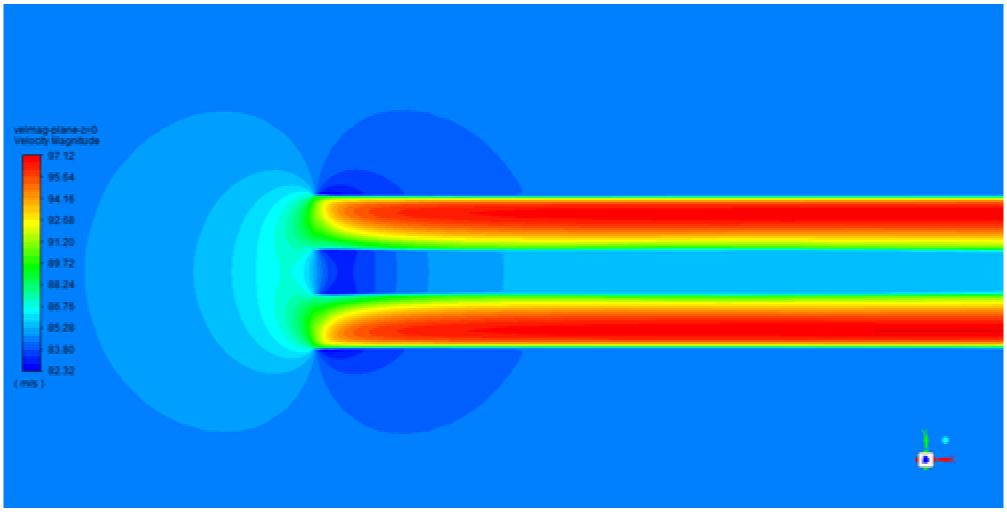



**Figure 31.51: Angle of Attack Distribution on the Actuator Disk**

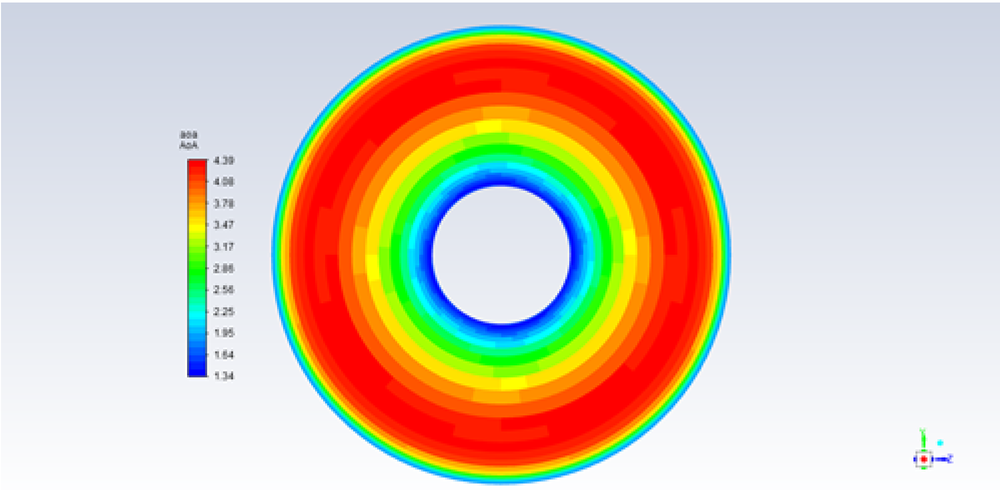
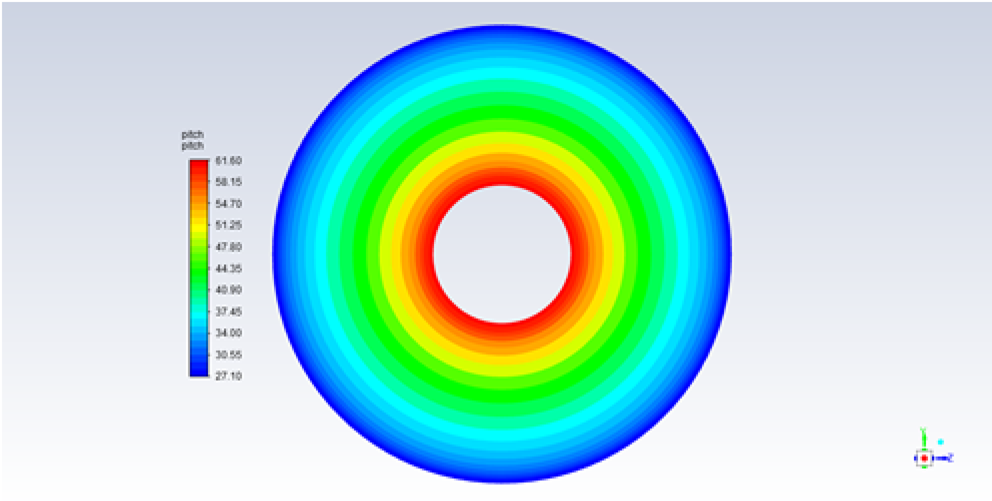


**Figure 31.52: Blade Pitch Angle Distribution on the Actuator Disk**




### 31.12.2.6.2. Rotor Simulation with Pitch Trimming (Collective Trimming)

The next simulation should yield identical results to the previous one, with the exception that now the VBM will trim the blade pitch angle (collective angle) of the propeller to match the same thrust. It is expected that the trimming will recover the exact blade pitch angle of the fixed-pitch propeller.

Before enabling the collective trimming, the thrust coefficient of the fixed-pitch propeller needs to be computed from the fixed-pitch solution with the following equation:

$$C_T = \frac{T}{C_{US\text{–}EU}\rho V_{tip}^2 \pi \frac{d^2}{4}}$$

Table 31.9: Values from the Fixed-Pitch Solution (p. 410) lists the values of the variables to insert in the formula. The rotor thrust $T$ can be recovered from column 3 of the last line of the VBM log file `Rotor_1_Loads.csv`.

$C_{US\text{–}EU}$ is the coefficient of the dynamic pressure, set to **1** in the code. The blade tip velocity and the reference density are taken from Table 31.7: Propeller Geometric Data and Operating Conditions (p. 392).

**Note:**

The usage of tip speed rather than the customary propeller axial velocity.

**Table 31.9: Values from the Fixed-Pitch Solution**

| Variable | Value | |
|---|---|---|
| $T$ | rotor thrust | 761.9504 N |
| $C_{US\text{–}EU}$ | dynamic pressure coefficient | 1 |
| $V_{tip}$ | blade tip velocity | 186.9414 m/s |
| $\rho$ | reference density | 1.187584 kg/m$^3$ |

| Variable | Value | |
|---|---|---|
| $d$ | rotor diameter | 1 m |

The computed thrust coefficient value **CT** is 0.0233756.

1. To enable collective trimming, go to:

   **Physics** → **Models** → **More**

   a. Select the **Virtual Blade Model** and click on the **Edit...** button to open the mini-graphical user interface.

   b. Set the **Collective** angle to **55**°.

   c. Tick-mark the **Trimming** box and click on the **Trimming** button in the ribbon.

   d. Tick-mark the **Collective pitch** box, set the **Update Frequency** value to **10**, the **Damping Factor** to **0.7** and enter the value **0.0233756** in the **Desired thrust coefficient** box.

   e. Click the **Change/Create** button, then press the **OK** button.

2. To run the calculation, go to:

   **Solution** → **Run Calculation** → **Run Calculation...**

   Enter a value of **600** in the **No. of Iterations** box and press the **Calculate** button.

   This run is intended to verify that the thrust coefficient value is precise and the collective angle printed to the console after 600 iterations converges towards the original 61.85345-degrees fixed-pitch blade angle.

3. When the execution terminates, save the case and data files (`VBM_propeller_tutorial_VP.cas` and `.dat`):

   **File** → **Write** → **Case & Data...**

At convergence the residual history and pressure monitor history will resemble Figure 31.53: Convergence History with Collective Trimming (p. 412) and Figure 31.54: Pressure Monitor Convergence History with Collective Trimming (p. 412).

**Figure 31.53: Convergence History with Collective Trimming**

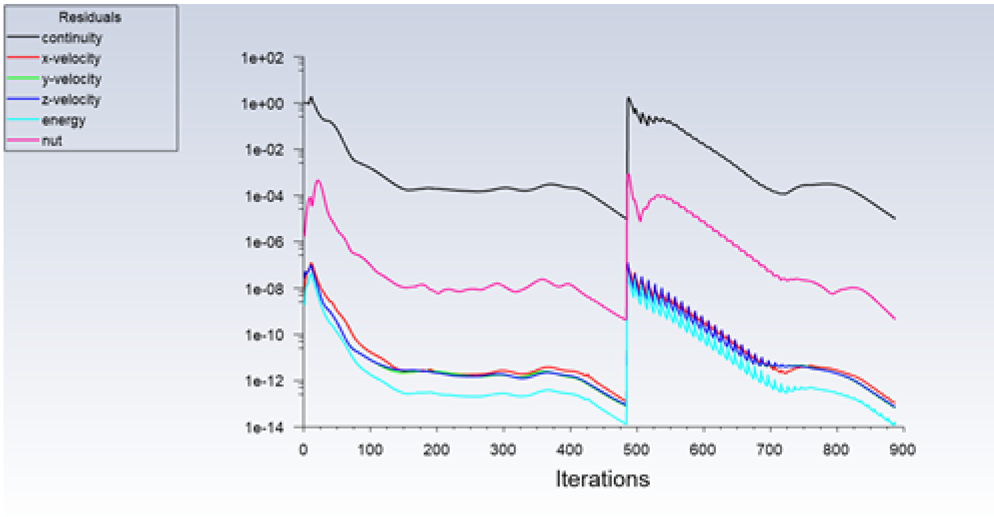
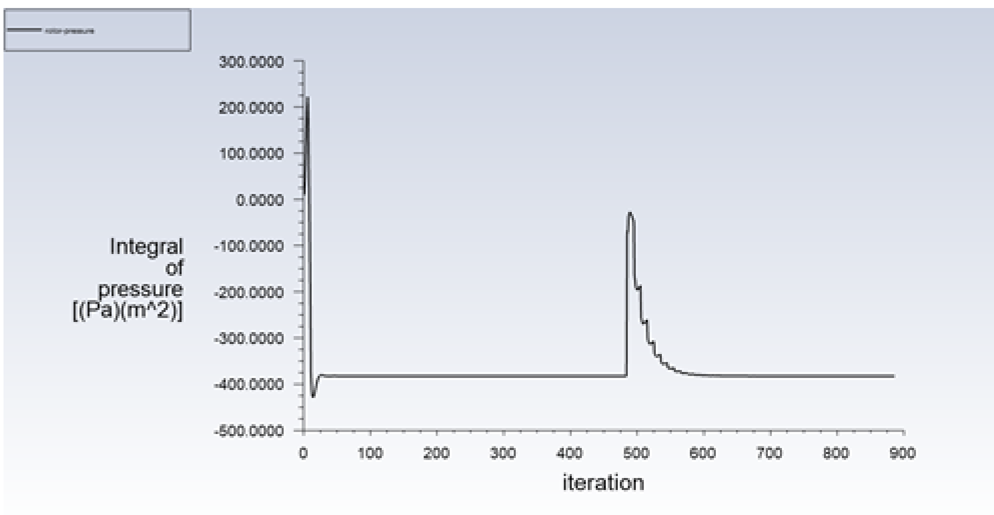



**Figure 31.54: Pressure Monitor Convergence History with Collective Trimming**




The converged rotor thrust with collective trimming is listed in column 3 of the `Rotor_1_Loads.csv` log file as 761.9533 N, a difference of 0.00038% with respect to the thrust of 761.9504 N obtained with fixed pitch. At convergence, the collective angle reported on the Fluent console is 61.853478, compared to the fixed-pitch setting of 61.85345˚ a difference of 0.00005%.

As expected, Figure 31.55: Velocity Magnitude Distribution Around the Actuator Disk with Collective Trimming; Z = 0 Cutting Plane (p. 413), Figure 31.56: Angle of Attack Distribution on the Actuator Disk with Collective Trimming (p. 413) and Figure 31.57: Blade Pitch Angle Distribution on the Actuator Disk with Collective Trimming (p. 413) show that, for the same thrust, the collective trimming has produced a solution virtually identical to the fixed-pitch case.

**Figure 31.55: Velocity Magnitude Distribution Around the Actuator Disk with Collective Trimming; Z = 0 Cutting Plane**

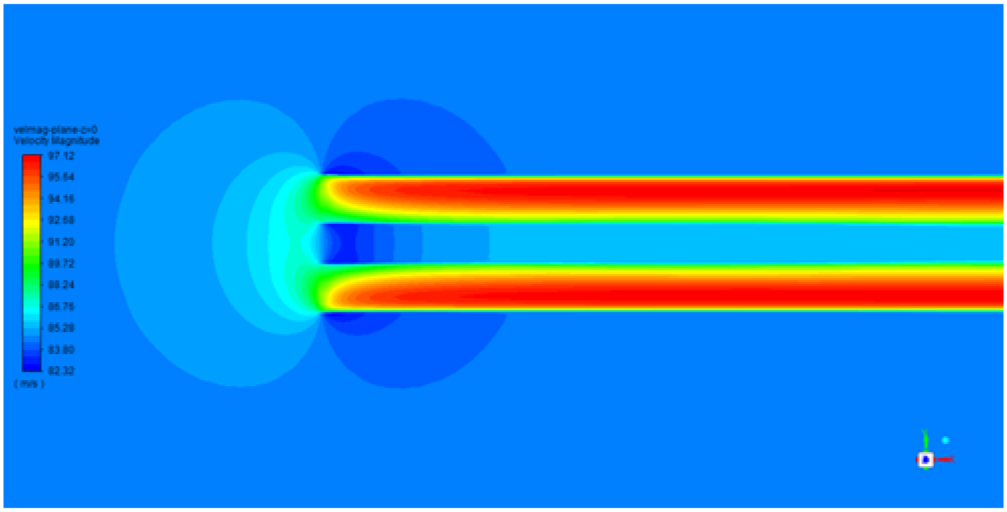


**Figure 31.56: Angle of Attack Distribution on the Actuator Disk with Collective Trimming**

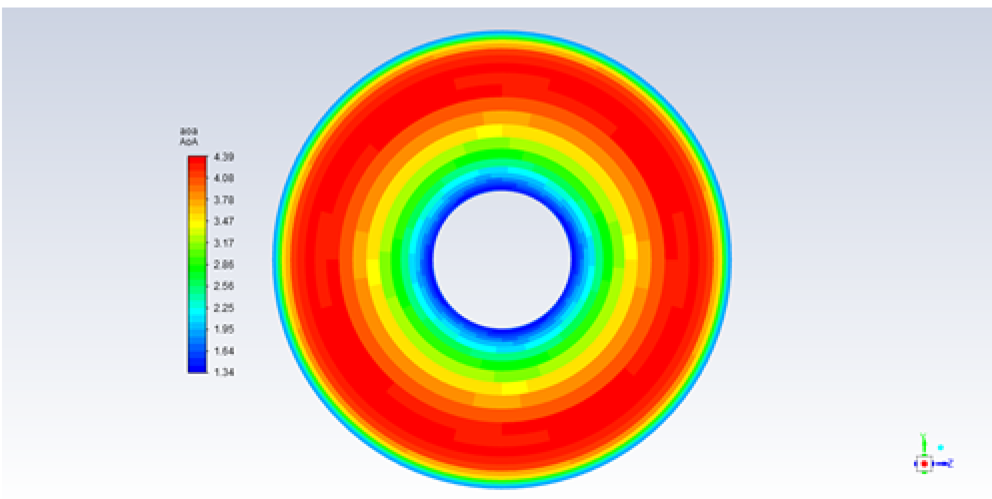
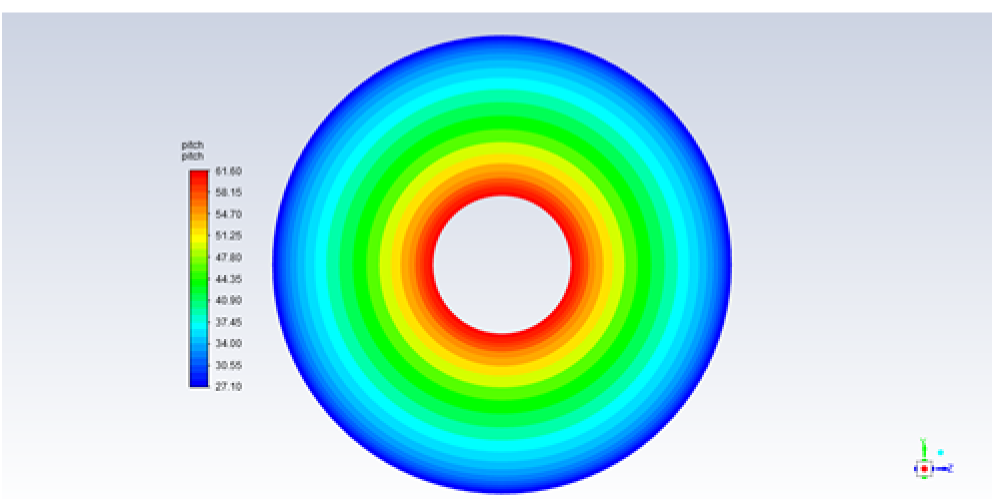



**Figure 31.57: Blade Pitch Angle Distribution on the Actuator Disk with Collective Trimming**



### 31.12.2.7. Summary

This simple tutorial shows that Fluent's VBM, which was originally written for helicopter rotor simulations, can also be used to simulate propellers, even when the actuator disk, the propeller analogue, is not aligned with the natural helicopter rotor disk orientation (Z-axis).

Step-by-step instructions for the Fluent and VBM set-up are provided to enable you to conduct realistic propeller simulations.

The comparison of the fixed-pitch results with the collective trimming case implicitly demonstrates that the VBM trimming routine is stable and accurate.

Considering the coarseness of the grid, which has been tailored towards portability, simplicity and fast turnaround of the tutorial runs, rather than accuracy, very reasonable results have been obtained for both the fixed-pitch and collective trimming simulations, compared to the validation shown in Fluent's Virtual Blade Model within the Fluent Customization Manual.

### 31.12.2.8. References

1. Capitao Patrao, A., "Description and Validation of the rotorDiskSource Class for Propeller Performance Estimation," In: Proceedings of CFD with OpenSource Software, 2017, Edited by Nilsson., H.

2. Glauert, H. "The Elements of Aerofoil and Airscrew Theory," Second Edition, Cambridge University Press, New York, USA, 1947.

3. Stepniewski, W.Z. and Keys, C.N., "Rotary-Wing Aerodynamics," Dover Publications Inc., New York, USA, 1984.

4. Leishman, J.G., "Principles of Helicopter Aerodynamics," Cambridge University Press, New York, USA, 2000.

5. Zori, L.A.J., Rajagopalan, R.G., "Navier-Stokes Calculation of Rotor-Airframe Interaction in Forward Flight", Journal of the American Helicopter Society, Vol.40, April 1995.

# Chapter 32: Multishot Icing with Automatic Remeshing Tutorial

The following sections of this chapter are:



## 32.1. Limitations

- Fluent report files are not supported when running multishot remeshing simulations. Disable these report files from your `.cas[.h5]` files using a Fluent Solution Workspace.

## 32.2. Multishot Glaze Ice with Automatic Remeshing Using Fluent Meshing

This tutorial demonstrates Multishot ice accretion in glaze icing conditions on the 3D Onera M6 swept wing using automatic remeshing with Fluent Meshing. Automatic remeshing provides improved robustness than the grid displacement approach used in Multi-Shot Ice Accretion with Automatic Mesh Displacement by effectively maintaining and improving the mesh resolution over an ice shape that continuously grows with time.

This tutorial is divided into two sections:

1. Setting up a Fluent Airflow Simulation on a Clean Onera M6 Wing (p. 416) demonstrates how to setup a clean (zero wall roughness) airflow case using a Fluent Solution Workspace for typical icing calculations.

2. Multishot Icing with Automatic Remeshing on the Onera M6 Wing (p. 420) shows how Fluent Icing can be setup to perform an 18-minute Multishot ice accretion calculation divided into 6 quasi-steady shots, of 3 minutes each. It is recommended to have a 64GB RAM machine to run this multishot icing simulation since the mesh size increases with the number of shots.

Download the `fluent_multishot.zip` file here.

Unzip `fluent_multishot.zip` to your working directory.

For convenience, `oneram6-wing.cas.h5` has already been setup accordingly for Setting up a Fluent Airflow Simulation on a Clean Onera M6 Wing (p. 416). You could therefore skip Setting up a Fluent Airflow Simulation on a Clean Onera M6 Wing (p. 416). However, if this is your first time performing an icing calculation with Fluent on a large 3D geometry, it is recommended to first continue with the steps shown in Setting up a Fluent Airflow Simulation on a Clean Onera M6 Wing (p. 416).

### 32.2.1. Setting up a Fluent Airflow Simulation on a Clean Onera M6 Wing

1. Launch Fluent on your computer. In the **Fluent Launcher** window, select **Solution**. Set the **Dimension** as **3D**, select **Double Precision** under **Options**. Set the number of **Solver Processes** to at least **12** CPUs. Click **Start** to launch the Fluent Solution Workspace.

2. Read the case file by going to **File** → **Read** → **Case**. Browse to and select the extracted file `fluent_multishot/oneram6-wing.cas.h5`.

   The table below provides the flight conditions:

   **Table 32.1: Flight Conditions**

| | |
|---|---|
| **Characteristic Length (MAC)** | 0.65 m |
| **Speed** | 103 m/s |
| **Angle of Attack ($\alpha$)** | 4° |
| **Pressure (Altitude = 5,000 ft)** | 84312.727 Pa |
| **Temperature** | 268 K (-5.15 °C) |

3. From the ribbon menu on the top, select **Physics** → **Solver** → **Operating Conditions...**. Set the **Operating Pressure** to **0** Pa. Press **OK**.

4. From the **Outline View** on the side, select **General** under **Setup**. Ensure that the **Solver** is set to **Type** → **Pressure-Based**, **Velocity-Formulation** → **Absolute**, and **Time** → **Steady**.

5. Under **Models**,

   - Select **Energy** and ensure that it is turned on.

   - Double-click **Viscous** to open the **Viscous Model** menu. There are different turbulence models that can be selected. For icing applications, it is strongly recommended to use the popular k-$\omega$ SST model, which is turned on by default. Enable **Viscous-Heating** and **Production Limiter** under **Options**. Under the **Model Constants** section, change the **Energy Prandtl Number** and **Wall Prandtl Number** to **0.9**, to be consistent with FENSAP-ICE, and the **Production Limiter Clip Factor** to **10**. Press **OK**.

6. Next, click **Materials** → **Fluid** and double-click **air** to open the air properties.

   - Set the **Density** to **ideal-gas**. The other airflow properties can be set from Fluent Icing.

   - Click **Change/Create** to save the air properties, then press **Close**.

7. Next, click **Boundary Conditions**,

   - Click **Inlet** and double-click **pressure-far-field-4** to set the far field boundary conditions.

     a. In the **Momentum** panel, set the **Gauge Pressure (pascal)** to **84,312.727** Pa and the **Mach Number** to **0.31385**. To define a 4-degree angle of attack ($\alpha$), set the **Coordinate System** to **Cartesian (X, Y, Z)** and the **X**, **Y** and **Z-Component**'s to **0.997564078**, **0.069756311**, and **0**. Under **Turbulence**, set the **Specification Method** to **Intensity and**

**Viscosity Ratio**, the **Turbulence Intensity** to `0.08`% and the **Turbulent Viscosity Ratio** to `1e-05`.

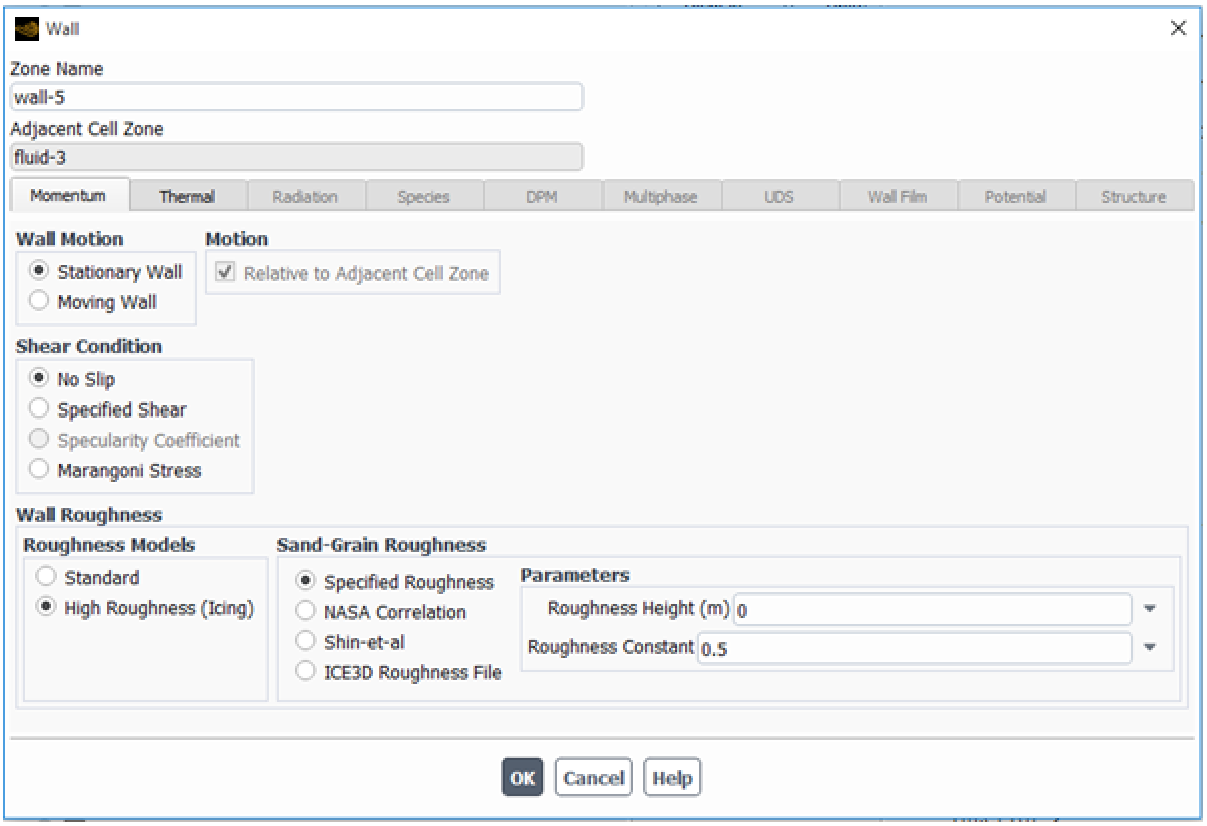
b. In the **Thermal** panel, set the **Temperature** to **268** K. Press **OK**.

- Click **Outlet** and double-click **pressure-outlet-8** to set the subsonic outlet boundary

  a. In the **Momentum** panel, set the **Gauge Pressure** to `84,312.727` Pa and enable the **Prevent Reverse Flow** option.

- Click **Wall** and double-click **wall-5**.

  a. In the **Momentum** panel, set the **Shear Condition** to **No Slip** and the **Roughness Models** to **High Roughness (Icing)**. Under **Sand-Grain Roughness**, keep the default settings and make sure that the **Roughness Height (m)** is set to `0` since the surface of the Onera M6 is considered clean.




---

**Note:**

It is also possible to enable the **High Roughness (Icing)** model as well as its **Roughness Height (m)** directly in Fluent Icing.

---

b. In the **Thermal** panel, set the **Thermal Conditions** to **Temperature**. Set the **Temperature** to `283.2797` K, and press **OK**. This temperature corresponds to the **Adiabatic stagnation temperature + 10K**, as required for standard external icing simulations with FENSAP-ICE.

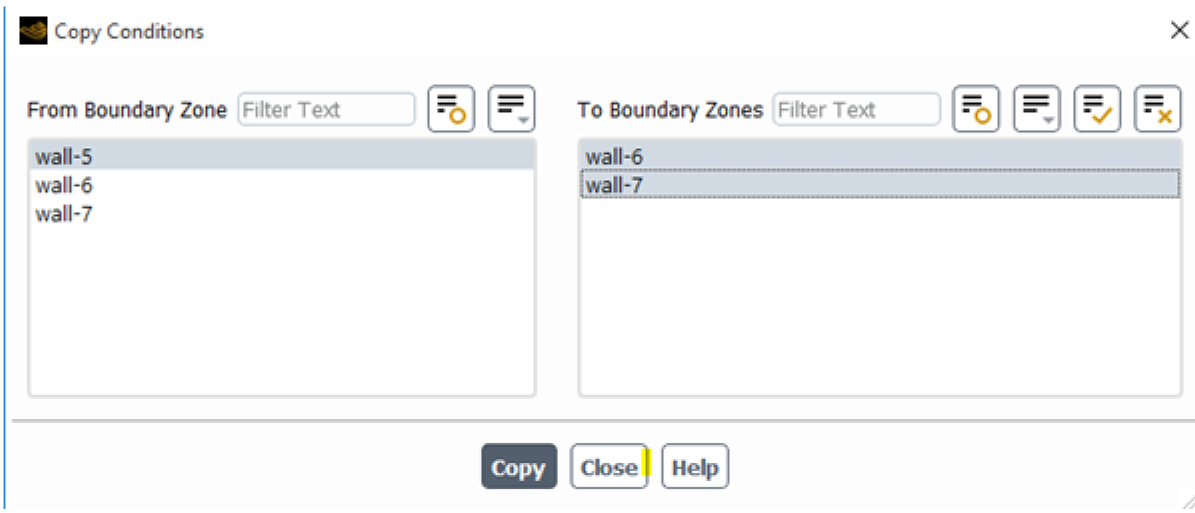
---

**Note:**

To save time, any arbitrary wall boundary condition can be set here as Fluent Icing can easily overwrite this by using the option **Adiabatic stagnation temperature + 10K** that automatically computes the correct value and imposes it at a selected wall.

---

- Copy the same boundary condition settings of **wall-5** on to **wall-6** and **wall-7** by right-clicking **wall-5** and select **Copy**. Under the **To Boundary Zones** select both **wall-6** and **wall-7** and click **Copy**.




8. Next, double-click on the **Reference Values**. Under **Compute from**, select **pressure-far-field-4**. Set the **Area** to `0.76` m2, the planform area of the wing. Set the **Length** to `0.65` m, the mean aerodynamic chord of the wing. Adjust the **Velocity** to `103` m/s.

---

**Note:**

These reference values will be used for post-processing purposes such as aerodynamic coefficients.

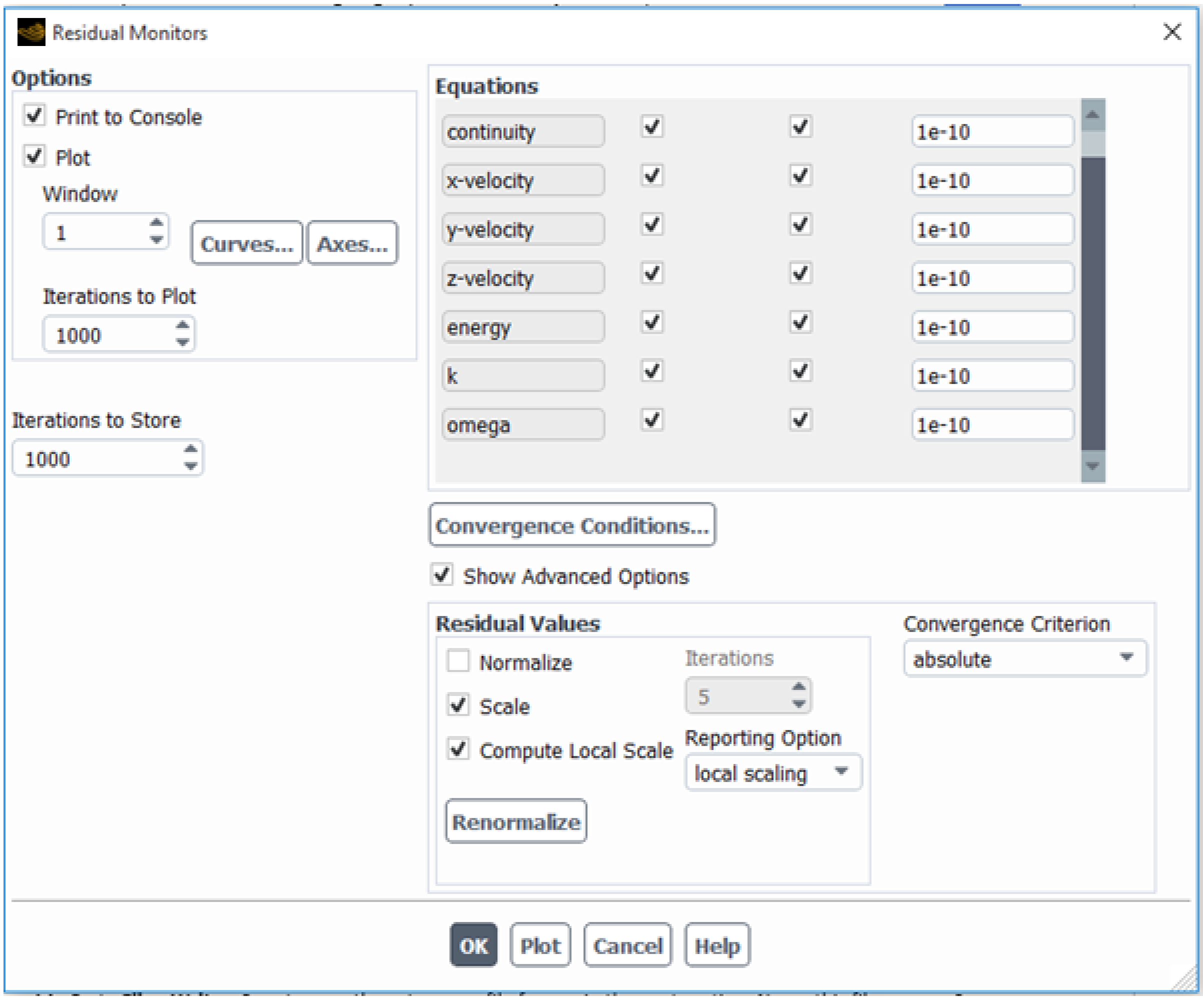
---

9. Continue to the **Solution** menu on the **Outline View**, double-click **Methods**. Set the **Pressure-Velocity Coupling** scheme to **Coupled**. Under **Spatial Discretization**, set the **Gradient** to **Green-Gauss Node Based** and the remaining options to **Second Order** or **Second Order Upwind**. Enable **Pseudo Transient** and **High Order Term Relaxation** to improve convergence.

10. Next, double-click the **Monitors** menu and then the **Residual** menu.

- Modify the **Absolute Criteria** for convergence to `1e-10` for all residuals. Ensure that **Monitor** and **Check Convergence** are selected for all residuals.

- Make sure that the **Print to Console** and the **Plot** are enabled.
- Enable **Show Advanced Options**. In **Residual Values**, check **Scale** and **Compute Local Scale**, and select **local scaling** under **Reporting Option**.




11. You can also monitor the drag and lift coefficients during the simulation.
    - In the **Outline View**, double-click **Report Definitions**. Select **New** → **Force Report** → **Drag**. Change the **Name** to `report-cd` and set the **Force Vector** to `0.99756405`, `0.069756474`, and `0`. Enable **Report Plot** and **Print to Console** under **Create**. Activate **Drag Coefficient** under **Report Output Type** and select **wall-5**, **wall-6** and **wall-7** under the **Wall Zones** section. Press **OK**.
    - In the **Report Definitions** window, select **New** → **Force Report** → **Lift**. Change the **Name** to `report-cl` and set the **Force Vector** to `-0.069756474`, `0.99756405`, and `0`. Enable **Report Plot** and **Print to Console** under **Create**. Activate **Lift Coefficient** under **Report Output Type** and select **wall-5**, **wall-6**, and **wall-7** under the **Wall Zones** section. Press **OK** and close the **Report Definitions** window.
12. Next, double-click the **Initialization** menu. Select **Hybrid Initialization** under **Initialization Methods**. Click **More Settings...**. In the **Hybrid Initialization** window, increase the **Number of**

**Iterations** to **20** and enable **Use External-Aero Favorable Settings** under **Initialization Options** in order to have a better starting airflow solution over the Onera M6 wing. Click **OK**.

13. Finally, double-click **Run Calculation**. Make sure that, in the **Pseudo Transient Settings**, the **Time Step Method** is set to **Automatic** and the **Length Scale Method** to **Conservative**. Change the **Time Scale Factor** to **0.1** in order to improve convergence. Under **Parameters**, set the **Number of Iterations** to **300**.

14. Go to **File** → **Write** → **Case** to save the setup case file for use in the next section. Name this file **oneram6-wing.clean.cas.h5**.

15. Go to **File** → **Exit** to close Fluent.

### 32.2.2. Multishot Icing with Automatic Remeshing on the Onera M6 Wing

In this section, you will learn how to set-up and perform a Multishot icing calculation on the Onera M6 wing with automatic remeshing in Fluent Icing. The automatic remeshing approach uses Fluent Meshing to remesh entirely the new iced surface and volumetric mesh while maintaining a high level of mesh quality and allowing complex features such as ice horns to be captured more precisely. It should be noted, that the current beta release has some limitations with the Fluent Icing approach.

---

**Note:**

The Automatic Remeshing approach shown in this tutorial is different to the automatic mesh displacement (ALE) method shown in Multi-Shot Ice Accretion with Automatic Mesh Displacement. In the ALE method, only the ice surface and the cells around the ice shape are displaced, therefore keeping the same number of nodes and cells as the original grid. The inherent disadvantage of the ALE method is the deterioration of the mesh quality (coarsening, increase in aspect ratio of cells, etc.) especially for longer ice accretion times where the possibility of complex ice shapes such as horns can develop.

---

The table below shows the In-flight icing conditions.

**Table 32.2: Simulation In-Flight Icing Conditions**

| | |
|---|---|
| **Characteristic Length (MAC)** | 0.65 m |
| **Speed** | 103 m/s |
| **Angle of Attack ($\alpha$)** | 4° |
| **Pressure (Altitude = 5,000 ft)** | 84,312.727 Pa |
| **Temperature** | 268 K (-5.15 °C) |
| **MVD** | 20 µm |
| **LWC** | Appendix C – Continuous Maximum |
| **Ice Accretion Time** | 18 minutes |

1. Launch Fluent on your computer. In the Fluent Launcher window, select **Icing**. Set the number of processes to at least **12** CPUs. Click on **Show More Options** and, in **General Options**, select a suitable **Working Directory** for your new multishot remeshing. Click **Start** to launch Fluent Icing.

2. Once Fluent Icing opens, the **Project** tab will be displayed by default. In the **Project**'s top ribbon panel, select **Project** → **New...** and enter **`Fluent_multishot_oneram6`** to create a new **Project** folder.

---

**Note:**

A Fluent project allows you to save and manage multiple **Simulations** and **Runs** in a single project folder.

---

3. From the top ribbon panel, go to **File** and select **Preferences...**. In the **Preferences** window, click **Icing** and enable **Beta features**. The automatic remeshing capabilities is a hidden feature that only appears in the graphical user interface of Fluent Icing when **Beta features** are active.

4. Go to the **Project** ribbon menu and select **Simulations** → **Import Case** and browse to and select the `fluent_multishot/oneram6-wing.cas.h5` file from the extracted `fluent_multishot.zip` archive.

   - A **New simulation** window will appear. Enter the **Name** of the new simulation as **`oneram6_icing`**, and check to enable **Load in solver**. Press **OK**.

   In this manner, a new **Simulation** folder will be created in your **Project** folder, and the `oneram6-wing.cas.h5` file will be imported.

   You should now see the **oneram6_icing (loaded)** tree displayed in the **Outline View** tab.

5. Select the **Setup** tree menu under **oneram6_icing (loaded)** and in the **Setup** panel, check **Airflow**, **Particles** and **Ice**.

6. Continue further down and right-click the **Airflow** tree menu located under **Setup** and select **Update with Fluent Case settings** to make sure that the previously setup Fluent simulation settings are properly transferred to Fluent Icing.

7. Click **Setup** → **Airflow** to display the **Airflow** panel.

   - Under **General**, select **Fluent** as the **Airflow solver**.

   - Under **Conditions**, make sure that the airflow conditions under **Conditions** correspond to the conditions in (Table 32.2: Simulation In-Flight Icing Conditions (p. 420)).

   - Under **Direction**, go to **Vector mode** and select **Angle of attack** to specify the orientation of the reference airflow. Set the **Lift Direction** and **Drag Direction** to **Y+** and **X+** respectively. Set the **AoA [deg.]** to **4** and the **Velocity magnitude [m/s]** to **`103`** (Table 32.2: Simulation In-Flight Icing Conditions (p. 420)).

8. Inside the **Outline View** window, right-click the **Fluent** icon located under **Airflow** and select **Set to default Air properties**. A message appears, to confirm that the air properties will be computed using the airflow temperature of 268 K. Press **OK**. This automatically sets the air properties to those suggested for icing simulations by using the current reference air temperature conditions. The values of air properties have been computed using the equations presented in Airflow within the Fluent User's Guide.

9. Click **Setup** → **Airflow** → **Fluent** to display the **Fluent** panel. Under **Materials**, select **Air** as the **Fluid** used in this simulation. The airflow properties computed in the previous step are shown.

10. Click **Setup** → **Particles** to display the **Particles** panel. In this panel, check only **Droplets**.

11. Click **Setup** → **Particles** → **Droplets** to display the **Droplets** panel.

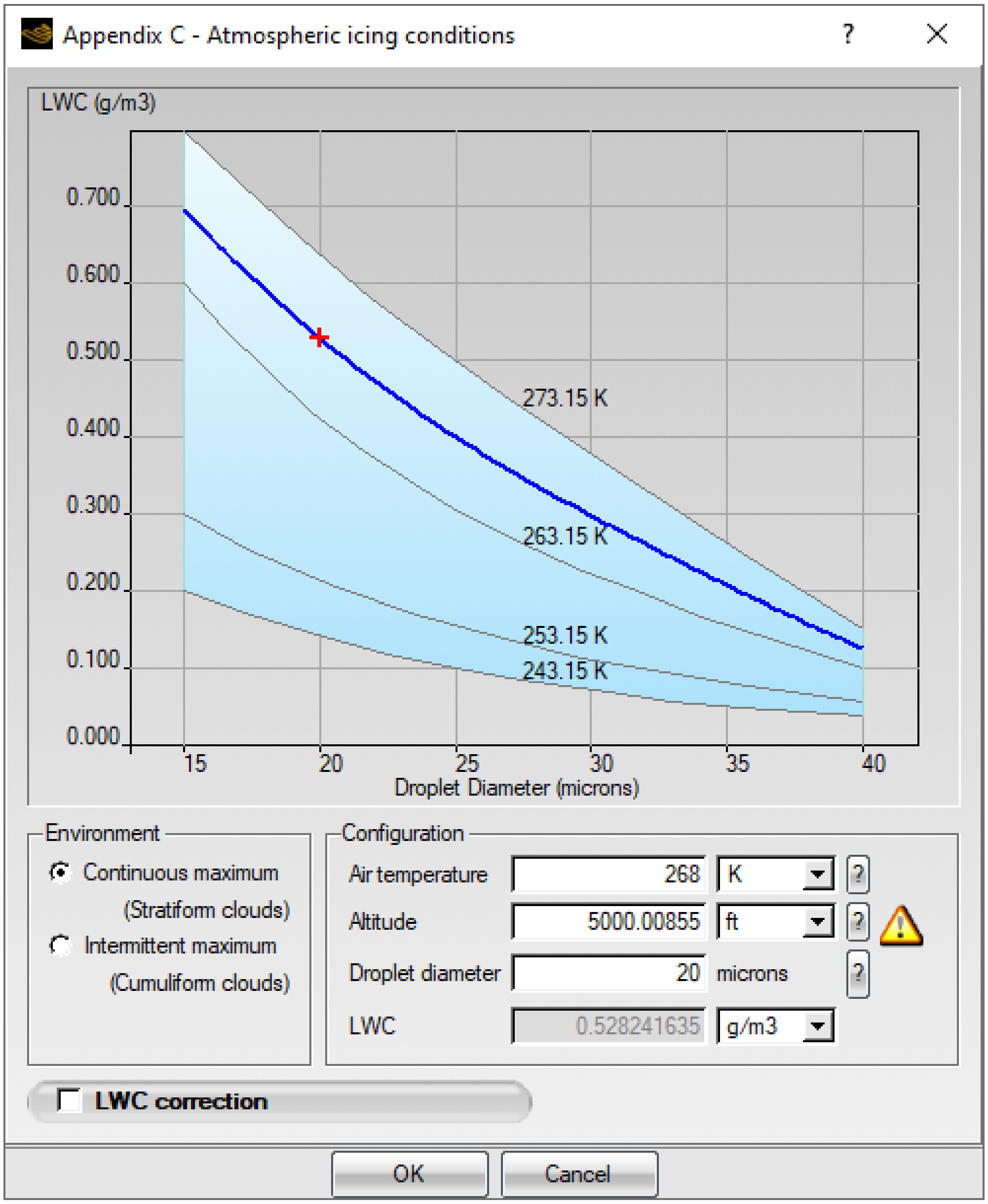
    - In this panel,

        a. Under **Droplet conditions**, set the **Droplet diameter [microns]** to `20`.

        b. Under **Appendix Conditions**, select **Appendix C**. Click the **Configure Appendix C** button located at the bottom left of this panel. An **Appendix C** window appears. In this window, select **Continuous maximum (Stratiform clouds)** in **Environment**. The **LWC** shown in this window automatically changes to 0.528241635 g/m3. Press **OK**. A message appears to confirm your selection. Click **Yes**. The **LWC [kg/m3]** value in the **Properties - Droplets** window is now set to `0.000528242` kg/m3 to respect the **Appendix C** continuous maximum cloud.

        c. Under **Particles Distribution**, keep **Monodispersed** since we will conduct a water catch simulation using a single droplet size.

        d. Under **Model**, keep **Water** as the **Droplet drag model**. This is the default drag law for droplet particles.




12. Click **Setup** → **Ice** to display the **Ice** panel. In this panel. Under **Model**, enable **Beading**.

13. Under **Setup** → **Boundary Conditions**,

- Go to **Inlets** → **pressure-far-field-4**,

  a. Under **Airflow**, select **Edit** besides **Conditions**. Click on **Import ref. conditions** located at the bottom of this property panel, to populate the airflow boundary conditions using the airflow conditions specified in **Setup** → **Airflow**.

  b. Under **Particles**, **From ref. conditions** is enabled and **Droplet velocity vector** remains unchecked. The **From ref. conditions** option will apply the **Droplet conditions** at the inlet of the pressure-far-field, in this case, the LWC and the MVD. If **Droplet velocity vector** remains unchecked, the airflow velocity is imposed as the droplet velocity at the inlet. In other words, the relative velocity between air and droplets is zero at farfield.

- Go to **Walls** → **wall-5**,

  a. Under **Airflow**, select **High roughness for Icing** next to **Wall Roughness** and set the **Roughness Height (m)** to **0.0005**. This is the initial roughness height that is applied on the first shot only. Other shots will have a roughness distribution applied over their iced surfaces since **Beading** was enabled in **Setup** → **Ice**.

  b. Make sure that under **Ice**, **Icing** is **Enabled**. This will allow ice accretion over this wall. Do not check **Specify Heat Flux**.

  c. To properly set the wall temperature for icing simulations, go to the **Outline View**, right-click on **wall-5** and select **Set temperature to Adiabatic + 10**. By doing so, the **Temperature [K]** under **Airflow** in the **Properties - wall-5** panel is correctly set.

- Repeat the process described above for wall boundaries **wall-6** and **wall-7**.

14. In the **Outline View**, under **Solution**,

- Click **Airflow** and set **Number of Iterations** to **300**.

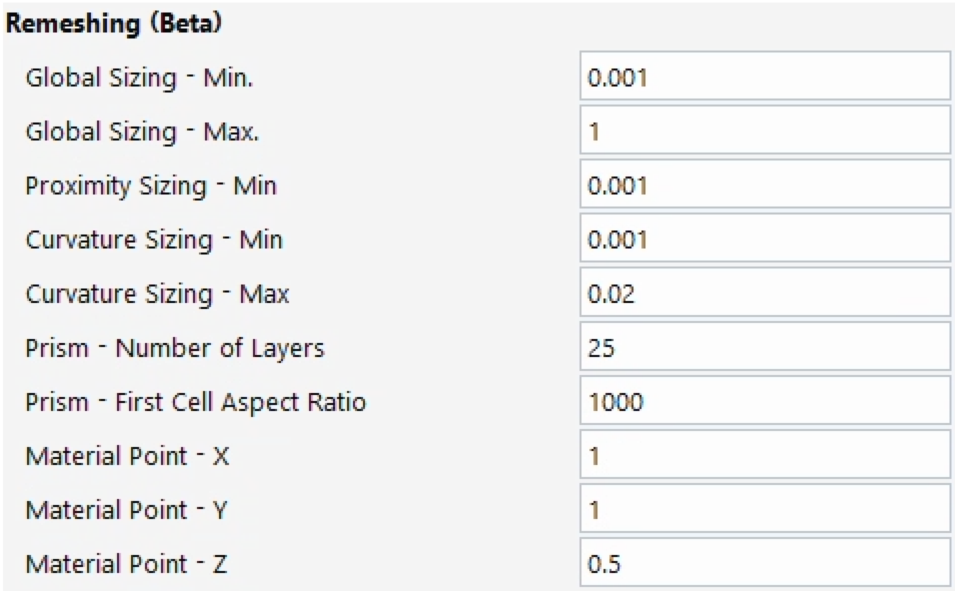
- Click **Particles** and set **Number of Iterations** to **120**.

- Click **Ice**.

  – Set **Total time of ice accretion [s]** to **180** which corresponds to 1/6th of the total time in Table 32.2: Simulation In-Flight Icing Conditions (p. 420)

  – Select **Set-up remeshing (beta)**, located at the bottom of the **Properties – Ice** panel. This automatically sets **Fluent Meshing** as the **Remeshing (Beta)** solver, under the **Output** option, and adds extra options, under **Remeshing (Beta)**, to control the type of mesh refinement to be done at each shot.

  – Below **Remeshing (Beta)**, several options are available to control the minimum and maximum surface tetra sizes as well as the prism layer, and to define a material point inside the volume that is used during the remeshing process. Enter the following settings as shown in the image below.

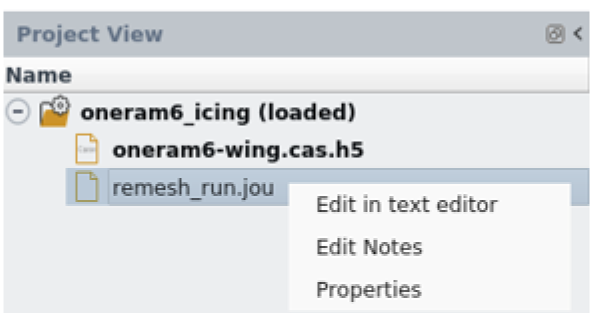



In this example, the mesh settings were created to match the requirements of this case. If you would like to further refine your 3D meshes to improve the level of precision of your simulations, you can modify these settings and increase the number of CPUs to compute your simulations. For more information, consult Automatic Remeshing Using Fluent Meshing.

---

**Note:**

In addition, a `remesh_run.jou` file is automatically created in the **Simulation** folder. This file contains the automatic remeshing steps that Fluent Meshing uses to run. It can be modified, as needed, by going to the **Project View** tab and right-clicking it and then by selecting **Edit in text editor**. This provides more flexibility and allows you to adapt the journal file according to your workflows.




---

15. Go back to the **Outline View**,

- Click **Solution**. In the **Solution** window, under **Multi-shot**,

  a. Set **Number of shots** to **6**.

b. Check **Save files at Each Shot** to examine the steady-state solutions at the end of each shot.

---

**Note:**

The airflow solver can be reinitialized at each shot or can continue from an interpolated solution coming from the previous shot to start its calculations. This can be done by choosing one of these options under **Airflow Restart**. In the current simulation, leave it to **Reinitialize**.

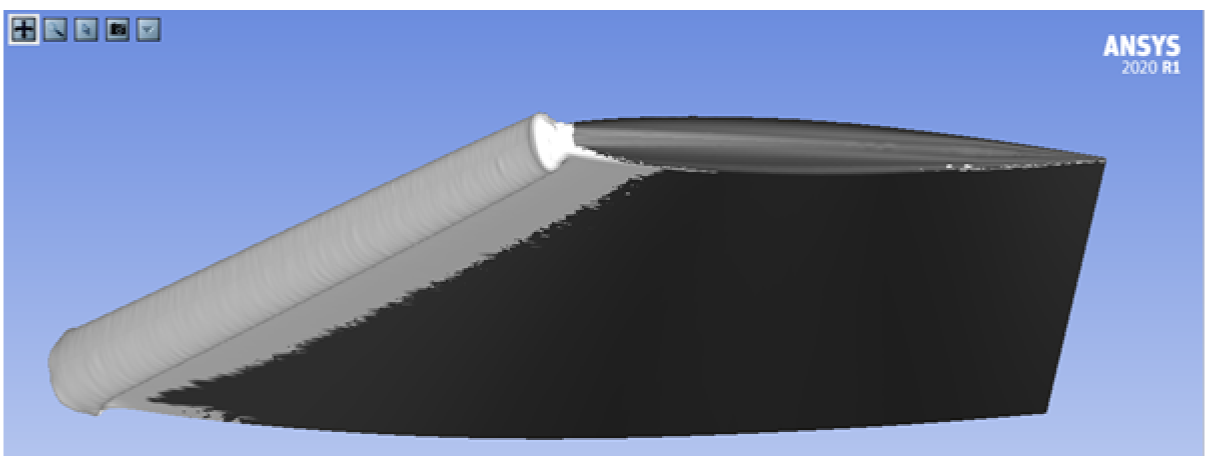
---

- Right-click **Solution**. Select **Run Multishot** to launch the Multishot calculation. A **New run** window will appear. Set the **Name of the new run** to `multishot_remeshing`. Click **OK**.

16. Go to the **Project View** by clicking the **Project** tab in the top ribbon menu. Under the **oneram6 icing (loaded)** simulation, a new run named **multishot_remeshing** now appears and is specified as **(current)**. Expand the run by clicking the **+** icon to the left of **multishot_remeshing** to show the files associated with the run. Each shot is represented by a grey folder and includes a two digit number to specify the shot number that it is associated with. Each grey folder contains a mesh file (**Case**), and 3 solution files (**Airflow**, **Droplets** and **Ice**).

17. Once all calculations are complete, view the final ice shape by right-clicking the **Ice** icon located under **shot.6** in the **Project View**. Select **View result** → **View with Viewmerical**. Choose **Ice Cover** in the **Select View Type** dialog.

**Figure 32.1: Multishot Automatic Remeshing Ice Shape**




Additionally, you can look at the convergence history of this simulation in the **Plots** display window located at the right of your screen. **Global-Air/-Particles/-Ice** provide a quick overview of the convergence history of all shots.

---

**Note:**

In the event your calculation stopped during the multishot simulation, you can resume your calculations from a given step within a shot. In this case, go to the **Project View**, right-click the **shot.xx** folder, and select **Set Restart Shot (Beta)**. Inside the **Outline View**, click on **Solution**. Inside its **Properties – Solution** panel, set the **Step** from where you would like to resume your multi-shot simulation. There are 4 steps: **Airflow**, **Particles**, **Ice**, and **Mesh Update**. Then, click the **Run multi-shot** button located at the left corner of the **Properties – Solution** panel to relaunch your calculations and click **Yes** to continue the current run. Calculations will start at the beginning of the step that was selected.

18. Do not close Fluent Icing if you would like to further post-process the Multishot ice solution.

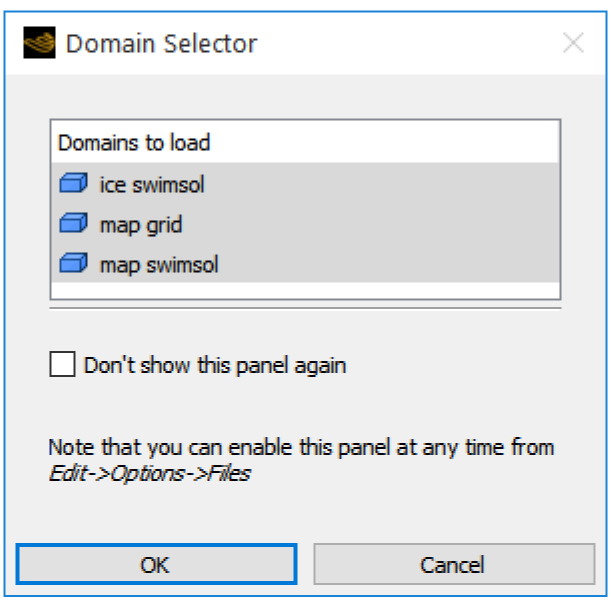
## 32.3. Multishot Glaze Ice with Automatic Remeshing - Postprocessing Using CFD-Post

In this tutorial, you will learn how to postprocess and generate figures and animations of a 3D Multishot ice accretion calculation (ice shape and ice solution fields) using two dedicated CFD-Post macros: **Ice Cover – 3D-View** and **Ice Cover – 2D-Plot** For this purpose, the Multishot icing solutions of the **multishot_remeshing** run are needed and, therefore, completion of Multishot Glaze Ice with Automatic Remeshing Using Fluent Meshing (p. 415) is required.

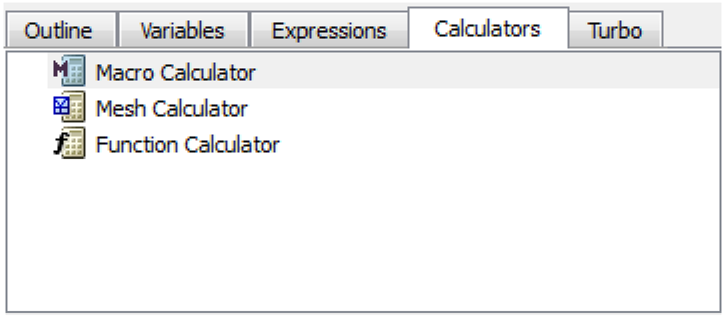
For more information regarding these macros, consult the CFD-Post Macros section within the FENSAP-ICE User Manual.

1. In your Fluent Icing window, go to the **Project View** and right-click **multishot_remeshing**. Select **View with CFD-Post** → **Ice** → **Ice cover**.

2. After opening CFD-Post, a **Domain Selector** window will request confirmation to load the following domains: **ice swimsol**, **map grid**, and **map swimsol**. Click **OK** to proceed.



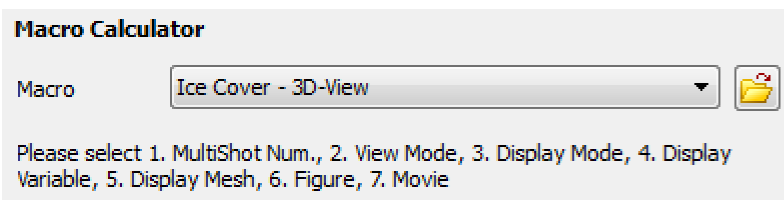

3. Go to the **Calculators** tab and double-click on **Macro Calculator**. The **Macro Calculator**'s interface panel will be activated and displayed.




**Note:**

The **Macro Calculator** can also be accessed by selecting **Tools** → **Macro Calculator** from the CFD-Post's main menu.

4. Select the **Ice Cover – 3D-View** macro script from the Macro drop-down list. This will bring up the user interface which contains all input parameters required to view ICE3D output solutions in the CFD-Post 3D Viewer.

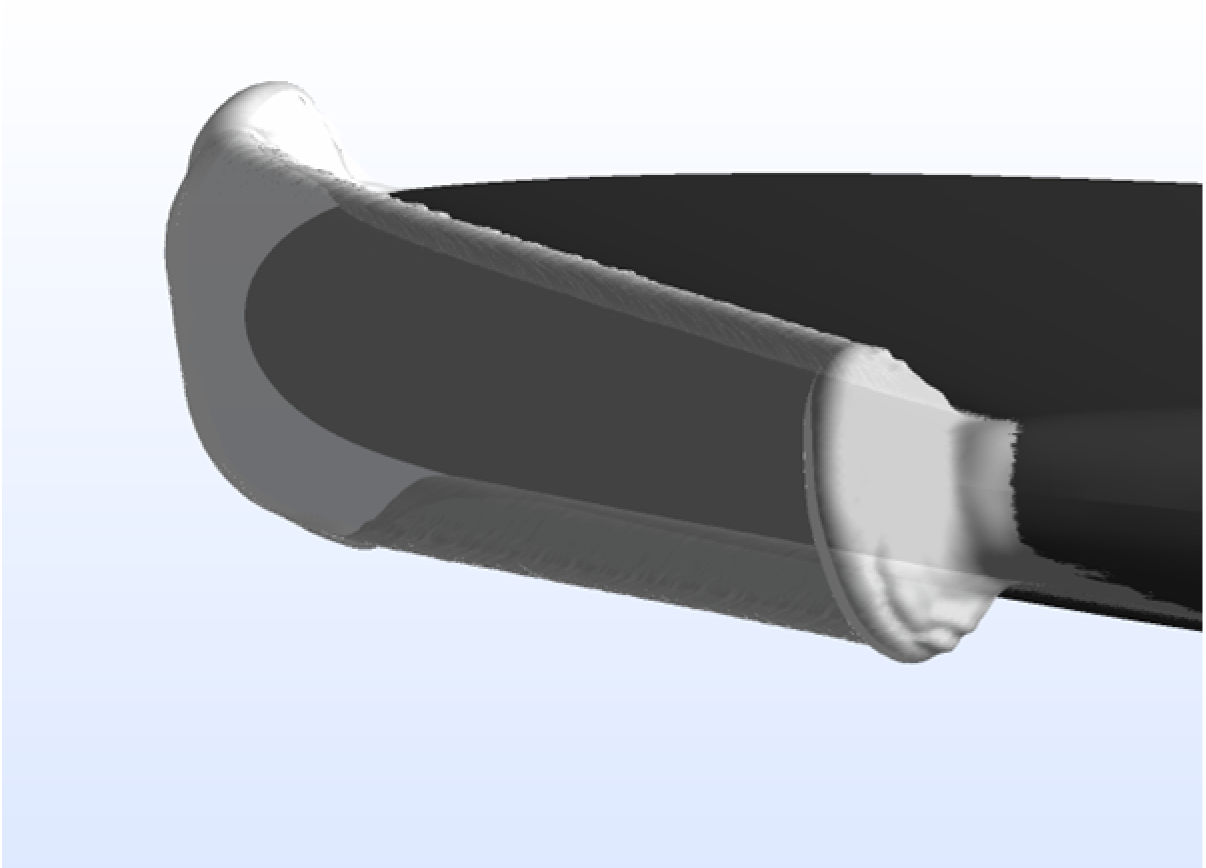



5. The default settings inside the **Macro Calculator** panel will allow you to automatically output the ice shape of a first shot of the Multishot simulation. Output the ice shape at the end of the Multishot simulation of Multishot Icing with Automatic Remeshing on the Onera M6 Wing (p. 420), this corresponds to the ice shape of shot 6. In this manner, set **Multi-shot #** to **6**.

6. Under **Display Mode**, enter **`0.2`** in **Transparency** to output a semi-transparent ice shape that will allow you to see the swept wing beneath the ice shape.

7. Leave the other settings unchanged. Click **Calculate** to execute the macro and view the ice shape in **3D Viewer**. Figure 32.2: Ice View in CFD-Post, Transparent Ice Cover of the Final Ice Shape (p. 429) shows the output of the macro.

**Figure 32.2: Ice View in CFD-Post, Transparent Ice Cover of the Final Ice Shape**



**Note:**

To change the style of the ice shape display, go to **Display Mode** and select one of the following options: **Ice Cover**, **Ice Cover – shaded**, **Ice Cover – No Orig**, **Ice Cover (only)** or **Ice Cover (only) - shaded**. To output the surface mesh of the ice shape, go to **Display Mesh** and select **Yes**.

8. To output the solution fields of your icing simulation, you can either select **Ice Solution – Overlay**, **Ice Solution** or **Surface Solution** under **Display Mode**. In this case, you will output the ice accretion rate over the ice layer without transparency.

9. To do this, select **Ice Solution – Overlay** in **Display Mode**.

10. Set a value of `0` to **Transparency** under **Display Mode** as this will provide a more solid view on the displayed surfaces.

11. Select **Instant. Ice Growth (kg s^-1 m^-2)** in the **Display Variable** drop-down list.

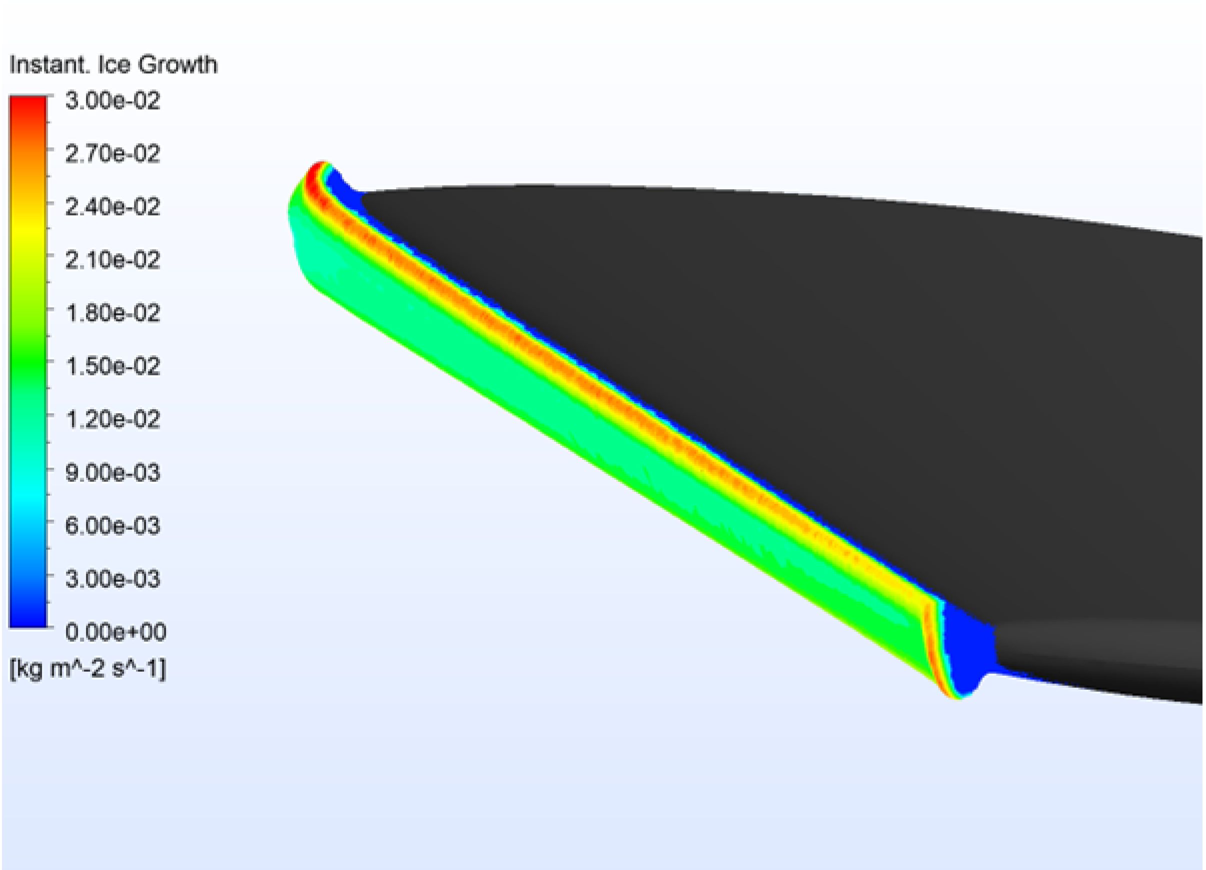
12. You will change the range and number of contours of the ice solution field. Under **Display Variable**,

    • Set **Number of Contours** to **21**.

- Change **Range** from **Global** to **User Specified**.
- Enter **0.03** and **0** in the **(Usr. Specif.) Max** and **(Usr. Specif.) Min** input boxes, respectively.

13. Click **Calculate** to view the instantaneous ice growth over the final ice shape. Figure 32.3: Ice View in CFD-Post, Instantaneous Ice Growth over the Final Ice (p. 430) shows the output of the macro.

**Figure 32.3: Ice View in CFD-Post, Instantaneous Ice Growth over the Final Ice**




14. Go to **Save Figure** to save this figure in a file,

- Select **Yes** beside **Save Figure**.
- Keep **Screen Shot** under **By**. If you select **Size**, you must specify width and height of the image.
- Keep the default **Format**. There are three types of format supported: **PNG**, **JPEG**, and **BMP**. The default format is **PNG**.
- Specify a **Filename** for the figure.

15. Click **Calculate** to generate and save the figure. A message will appear to notify the user of the location where the figure is saved.

    **Note:**

    If CFD-Post was opened through Fluent Icing, the figure will be saved in the run folder. If CFD-Post was opened in standalone mode, the figure will be saved in the Window's system default folder.

16. You can also generate and save animations that highlight the ice shape evolution of your Multishot simulation while displaying an ice solution field over it. Follow these steps to create and save a custom animation.

    - Go to **Save Figure** and select **No**.
    - Go to **Multi-shot #** and set it to **1**. The animation starts at the assigned shot number in **Multi-shot #** to the last shot of the simulation.
    - Set **(Multi-shot) Movie** to **On** and click **Calculate** to see the animation on the **3D Viewer** window.
    - To save this animation, in **(Multishot) Movie**,
      - Set **Save** to **Yes**.
      - Keep the default **Format**. Two formats are supported, **WMV** and **MPEG4**. The default is **WMV**.
      - Specify a **Filename**.
    - Click **Calculate** to generate and save the animation. A message will appear to notify the user of the location where the animation is saved and at which shot the animation starts.

      **Note:**

      If CFD-Post was opened through Fluent Icing, the animation will be saved in the run folder. If CFD-Post was opened in standalone mode, the animation will be saved in the Window's system default folder.

17. Select **Ice Cover – 2D-Plot** from the **Macro** drop-down list to create 2D-plots of the Multishot simulation. You will create 2D-Plots at various locations along the swept wing using a single shot solution or multiple shot solutions.

18. Set **Multi-shot #** to **3** since you will output the ice shape of the third shot.

19. Change the **Plot's Title** to `Multishot – Remeshing – Shot No.3`.

20. Keep the default setting of **2D-Plot (with)**. The default setting is **Single Shot**. The other options of **2D-Plot (with)** allow the creation of multiple shot results within the same 2D-Plot.

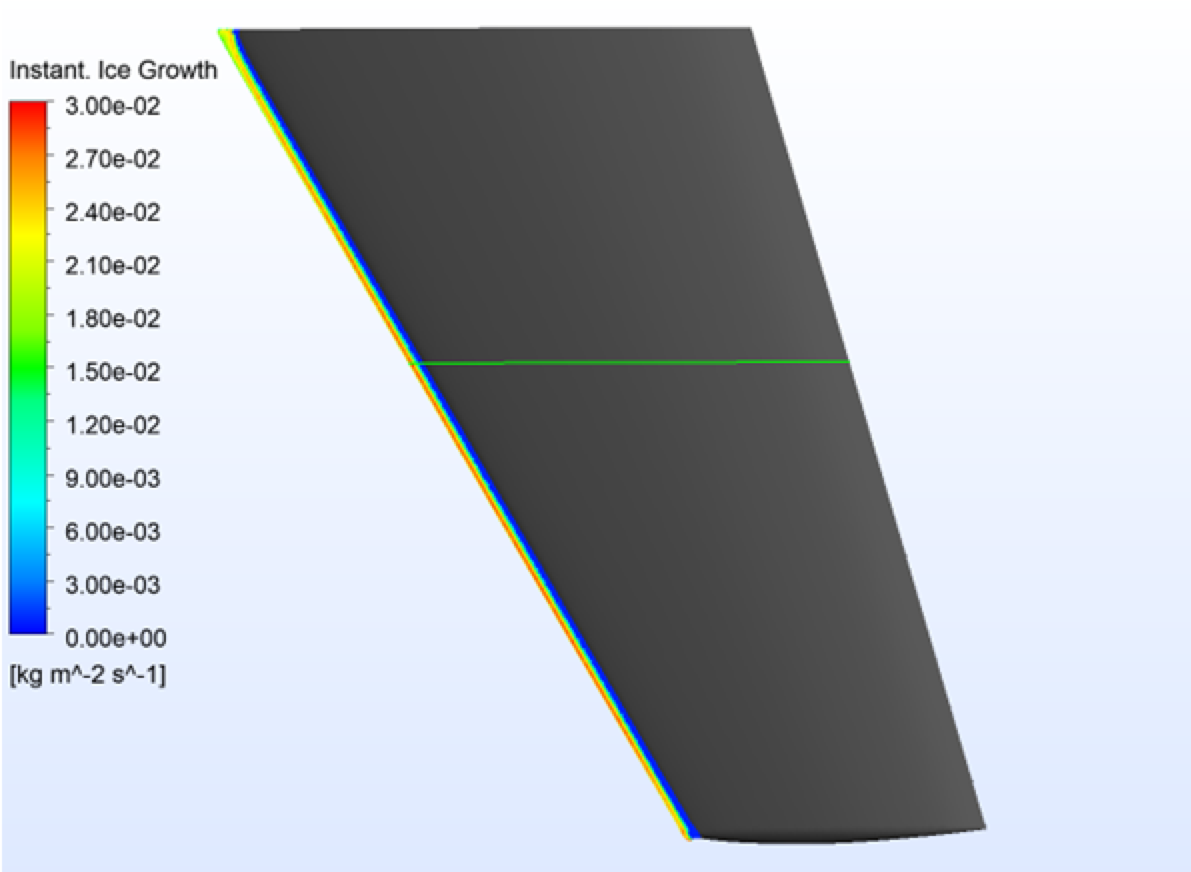
21. Under **2D-Plot (with)**,

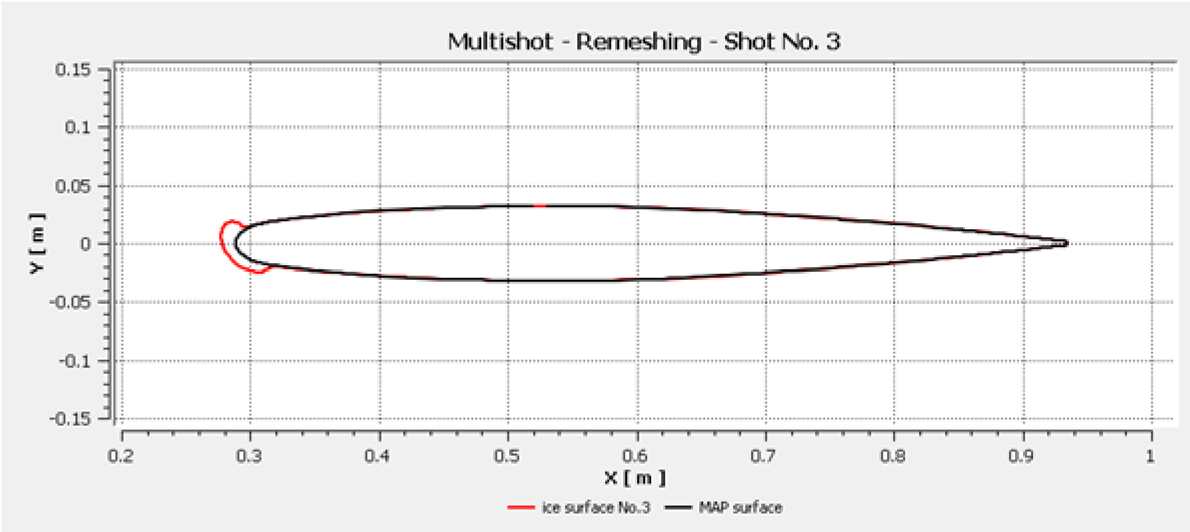
    - Keep the default setting of **Mode**. The default setting is **Geometry** to output the ice shape.

- Make sure **Cutting Plane By** is set to the default **Z Plane**.
- Set **X/Y/Z Plane Point** to `0.5`. In this case, this corresponds to a Z=0.5 plane.
- Set the **X-Axis** to **X** and the **Y-Axis** to **Y**.

22. To center the 2D-Plot around the wing section at Z=0,
    - Keep the **(x)Range** of the **X-Axis** to **Global**.
    - Change the **(y)Range** of the **Y-Axis** from **Global** to **User Specified**. Specify values of `0.15` and `-0.15` in the input boxes of **(Usr.Specif.y)Max** and **(Usr.Specif.y)Min**, respectively.
23. Leave the other default settings unchanged and click **Calculate** to create a 2D-Plot of the 3rd shot ice shape at Z=0.5 in a floating **ChartViewer** of CFD-Post. Adjust the output window's size. The figure below shows the cutting plane location in the **3D Viewer** window and the output of the macro in the **ChartViewer**.

**Figure 32.4: 2D-Plot in CFD-Post, 3rd Shot Ice Shape at Z=0.5**






24. You will now create a 2D-Plot of all the ice shapes of the multishot simulation using a user-defined cutting plane and save this image in a file. First, set **Multi-shot #** to **1**.

25. Change the **Plot's Title** to `Multishot – Remeshing – All Shots`.

26. Select **Multi-Shots** from the **2D-Plot (with)** drop-down list. This will generate a series of 2D plot curves, starting from the assigned shot number in **Multishot Num.** to the last shot of the multishot simulation.

27. Under **2D-Plot (with)**,

    - Keep the default setting of **Mode**. The default setting is **Geometry** to output the ice shape.

    - Set the **Cutting Plane By** to **Point and Normal** to define an arbitrary cutting plane. The normal does not need to be a unit vector.

    - Set the coordinate of the point (**`0.533646`**, **`-0.022681`**, **`0.898708`**) inside **(Pt. & Nml.)Pnt.X/Y/Z**, respectively.

    - Set the normal vector coordinates (**`0.5`**, **`0`**, **`0.866025`**) inside **(Pt. & Nml.)Nml.X/Y/Z**, respectively.

    - Set the **X-Axis** to **X** and the **Y-Axis** to **Y**.

28. To center the 2D-Plot around the leading edge of the wing section defined by the arbitrary cutting plane,

    - Change the **(x)Range** of the **X-Axis** from **Global** to **User Specified**. Specify values of **`0.65`** and **`0.45`** in the input boxes of **(Usr.Specif.x)Max** and **(Usr.Specif.x)Min**, respectively.

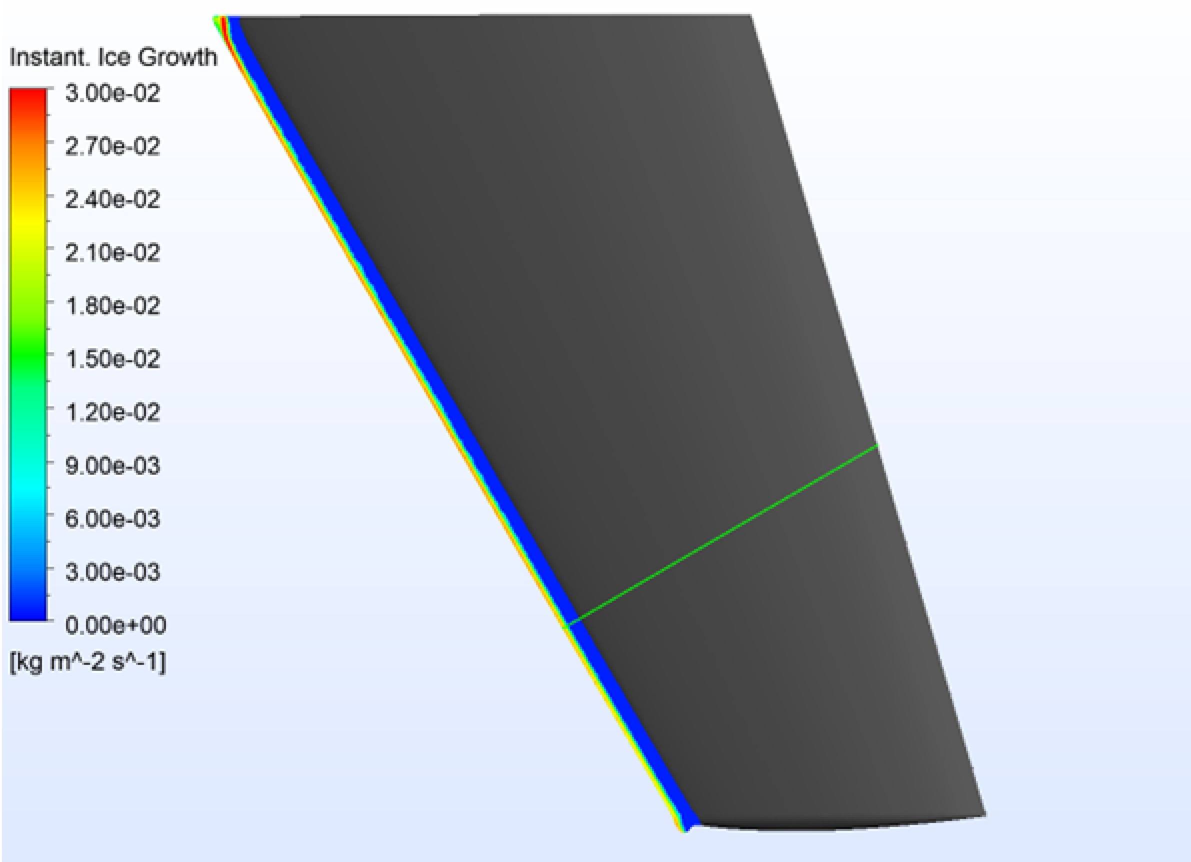
    - Change the **(y)Range** of the **Y-Axis** from **Global** to **User Specified**. Specify values of **0.06** and **-0.06** in the input boxes of **(Usr.Specif.y)Max** and **(Usr.Specif.y)Min**, respectively.

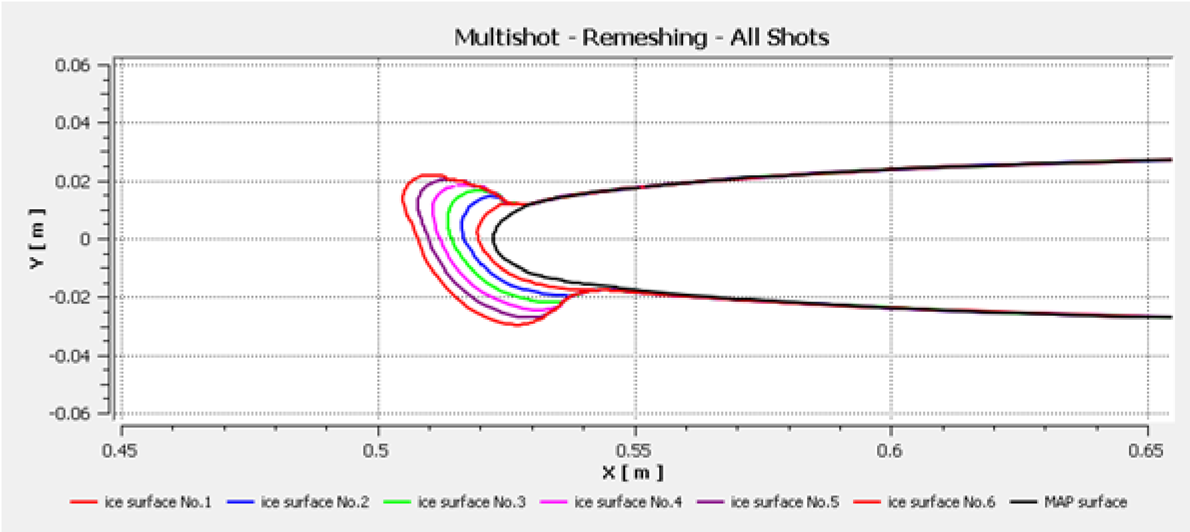
29. To simultaneously save this 2D-Plot as a figure, go to **Save Figure**,

- Select **Yes** beside **Save Figure**.
- Keep the default **Format**. There are three types of format supported: **PNG**, **JPEG**, and **BMP**. The default format is **PNG**.
- Specify a **Filename** for the figure.

30. Leave the other default settings unchanged and click **Calculate** to create a 2D-Plot of all the ice shapes at a user-defined cutting plane in a floating **ChartViewer** of CFD-Post. A message will appear to notify the user of the location where the figure is saved. Adjust the output window's size. The figure below shows the cutting plane location in the **3D Viewer** window and the output of the macro in the **ChartViewer**.

**Figure 32.5: 2D-Plot in CFD-Post, Multishot Ice Shapes at a User-Defined Cutting Plane**




Multishot - Remeshing - All Shots
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
Y [ m ]
0.45
0.5
0.55
0.6
0.65
X [ m ]
ice surface No.1
ice surface No.2
ice surface No.3
ice surface No.4
ice surface No.5
ice surface No.6
MAP surface

# Chapter 33: Fluent Aero

The following sections of this chapter are:



## 33.1. Known Issues and Limitations in Fluent Aero 2021 R2

- Rarely, the user interface may freeze during the calculation of the first Design Point. The user interface will unfreeze after the first Design Points has been calculated.

- The following message, `Warning: Pressure-far-field boundary conditions may only be used with ideal gas law`, may appear when you first create a new aero workflow. This warning can be ignored as Fluent Aero will automatically enable the ideal gas law when **Calculate** is pressed.

- Adiabatic walls are automatically applied when **Calculate** is pressed. If your imported case file contains wall thermal boundary conditions that are not adiabatic, they will be overwritten by Fluent Aero.

- Only steady state absolute reference frame solutions are supported.

- The **Flow Direction** setting of the **Flight Conditions** inside **Properties - Simulation Conditions** cannot be modified when using a **WindTunnel Domain Type**. It is recommended to impose a flow direction that is orthogonal to the inlet of the wind tunnel computational or aligned with the orientation of the test section walls of your domain. For **WindTunnel** type simulations, if a user would like to change the orientation of the flow with respect to the aircraft, they must rotate the surface mesh of the aircraft model and use in a new aero workflow simulation.

- When you modify a boundary zone, the user interface may have some issues in updating the **Properties** panel associated with that boundary zone.

  - Changing to a new zone type may not immediately reveal the new settings associated with the new zone type. To display them, simply click another item in the **Outline View**, then click back on the zone that was just changed, and the new settings will be displayed.

  - A note will correctly appear in the **Properties** panel specifying that a zone type is not yet directly supported in Fluent Aero. However, if you modify the same boundary back to a supported type (for example, back to **pressure-outlet**), the note will still be present. Simply ignore this note, as it is incorrect and will not have any effect on your simulation.

- It might be possible to edit columns in the **Input:Design Points Table** that are protected and should not be modified by you (such as the **DP** column) by using some combination of mouse click and mouse drag operations. If you mistakenly changed this value, use the **Reload DP Table** command to refresh the values to their intended state.

- The first time you calculate a new simulation, a message will be printed to the console stating that a boundary type used in previous results has changed and that the `run.settings` file will be deleted, even though no previous results exist. In this situation, the message should not have be printed. Since no previous results exist, it will have no effect on your simulation, so the message can be ignored.

## 33.2. Overview of Fluent Aero

Fluent Aero allows users to easily explore the aerodynamic performance of aircraft under a wide range of flight regimes, from subsonic to hypersonic conditions, all within a dedicated Fluent Application Client environment. A streamlined workflow guides the user through the creation of a matrix of flight conditions or design points where single and multiple flight parameters, such as angle of attack, Mach number, altitude, etc., can vary. Most common models, solvers and convergence settings of Fluent are tuned using the latest best practices for external aerodynamic problems and are available in Fluent Aero's user interface. Simulations can then be conducted in a quicker and more user-friendly environment. Furthermore, the full capabilities of the Fluent Solution Workspace remain accessible when its session is displayed through the Fluent Aero ribbon's **Workspaces** → **Solution** button.

## 33.3. Quick Start

Complete work procedures, with example dataset and conditions, are available as a separate tutorial guide - Fluent Aero Tutorial (p. 566).

### To Set-up and Run Simulations with Fluent Aero

- Launch **Fluent** on your computer. In the **Fluent Launcher** window that appears, check to enable **Show Beta Workspaces**. Select **Aero (beta)** from the left panel. Set the number of **Solver Processes**, and click **Start**.

- In the Fluent Aero user interface, go to **File** → **Preferences...**. In the **Preferences** window, go to the **Aero** menu. Enable **Advanced Settings**. If you would like to calculate your simulations on your local machine, disable **Use Custom Solver Launch Settings**. Disable **Use Custom Solver Launch Settings**, **Show Solution Workspace** and **Enable Solution Workspace Graphics**, as these are not required for typical runs on your local machine.

- In the **Project** ribbon, select **Project** → **New**, and choose an appropriate name and location for the Project file, and click **OK**.

- Go to **Simulations** → **New Aero Workflow**, and select a suitable (`.cas`, `.cas.h5`) file.

---

**Note:**

This case file must:

– Contain a 3D mesh that is oriented such that the geometry is aligned with the cartesian axis.

– For a **Freestream** type domain, contain a pressure-farfield or a pressure-farfield with a pressure outlet type boundary to define the main flow inlet/outlet boundary conditions.

– For a **WindTunnel** type domain, contain a velocity-inlet and a pressure outlet type boundary to define the main flow inlet/outlet boundary conditions of the test section in the tunnel, and wall or symmetry type boundary conditions to define the side boundaries of this test section. Name these boundaries **`windtunnel-inlet`**, **`windtunnel-outlet`**, **`windtunnel-wall*`**, and **`windtunnel-symm*`** to ensure that the automatic domain type setup works properly.

– Be set to material type (Air.)

---

- A new Simulation is created. When prompted, choose an appropriate name for the new simulation, and select **Yes**. A simulation folder with the chosen name is then created inside the project, and the case file is loaded into a remote Solver session.

- Once the **Solver** is loaded, the **Simulation** ribbon view will be displayed. Use the aero application to review and apply the conditions and settings.

  – **Geometric Properties**

  → Ensure the correct domain type for your geometry has been selected (**Freestream** or **WindTunnel**).

  → Define the geometric grid orientation with respect to the cartesian axis, and the reference length and area.

  – **Simulation Conditions**

  → Choose the input **Parameter** types and setup the atmospheric **Flight Conditions** to use in your simulation for each **Design Point**.

  → Choose a **Distribution** of each parameter

  - **Constant**: specify conditions that will be common to all **Design Points**.

  - **Uniform**: specify conditions that will vary uniformly per **Design Point**.

  - **Custom**: specify conditions that will vary non-uniformly per **Design Point**.

  → Use the **Input: Design Points** table to set the custom input values for each design point.

  – **Component Groups**

→ Organize the domain's boundary zones into aircraft **Component Groups**, such as **Wing**, **Engine**, **Fuselage**, etc., to help the setup of boundary conditions and the analysis of individual aircraft components.

→ Ensure that the **Freestream** or **WindTunnel** group correctly defines the external boundary of your domain.

→ Specify zone specific boundary conditions (such as a mass flow on an engine exhaust).

– Set **Solve**

→ Set the number of Iterations to run for each design point.

→ Check to enable **Show Advanced Settings** if you would like to change the default value of various CFD solver parameters such as Turbulence Models or Convergence settings.

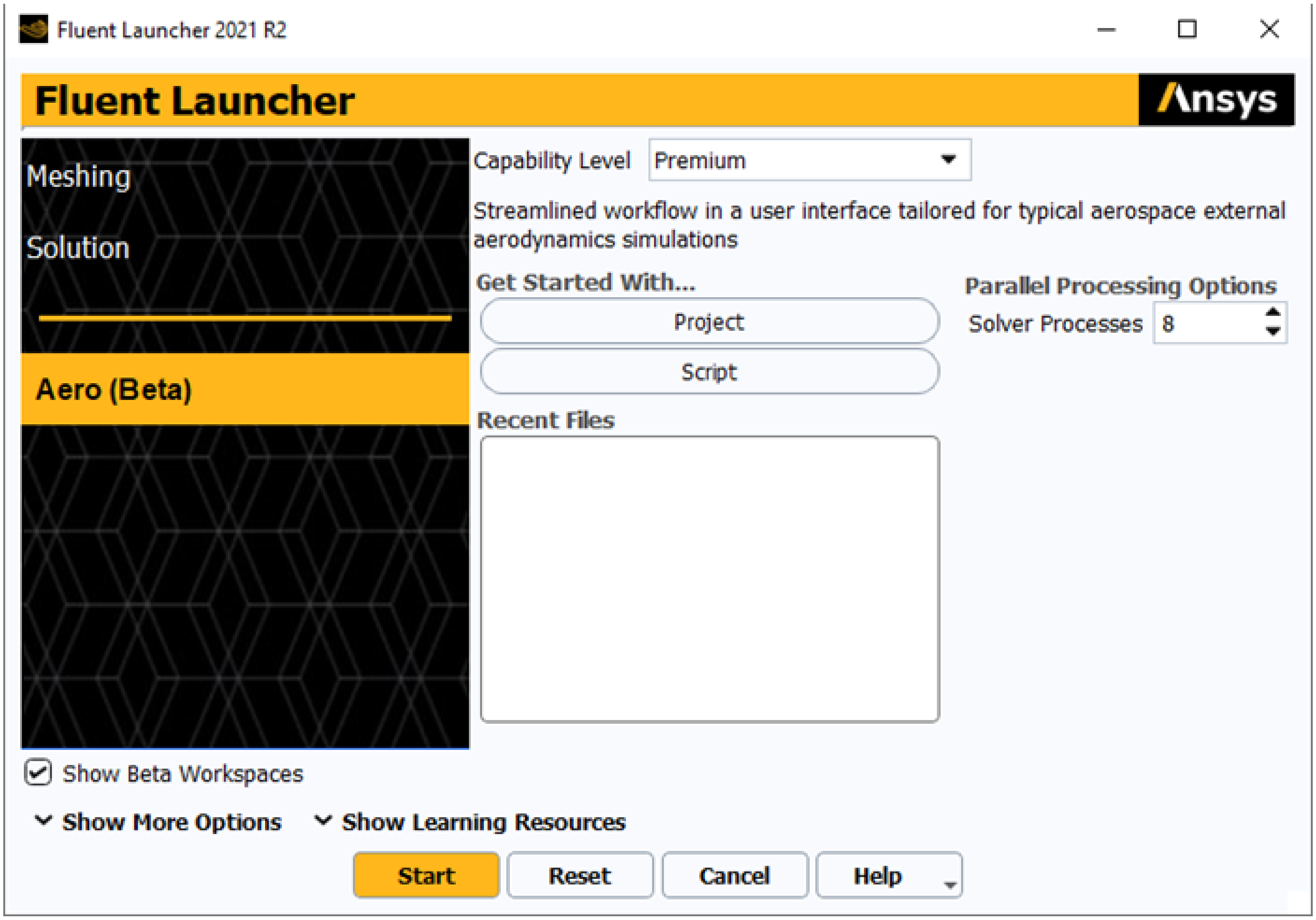
- Once the appropriate conditions and settings are applied, click on **Solve** and then click **Calculate** to start the calculations.

- Once the Simulation has started, monitor them using the **Convergence** window. A **Results** section becomes available in the **Outline View** tree – use the various **Tables**, **Graphs**, **Plots** and **Contours** options to investigate the results.

## 33.4. Starting Fluent Aero

Fluent Aero can be started from either:

- The **Fluent Launcher**
  - Launch Fluent. In the **Fluent Launcher** window, enable **Show Beta Workspaces**, and select **Aero (Beta)**. Set an appropriate number of **Solver Processes** and click **Start** to launch Fluent Aero.




- Your Ansys installation folder:
  - Windows

    `fluent/bin/aero.bat` in the `Ansys Inc/v212` folder
  - Linux

    `fluent/bin/aero` in the `Ansys Inc/v212` folder

## 33.4.1. Solver and License Requirements

When you load a case file into Fluent Aero, the Fluent solver launches using the number of CPUs that you specified. This uses the following license keys:

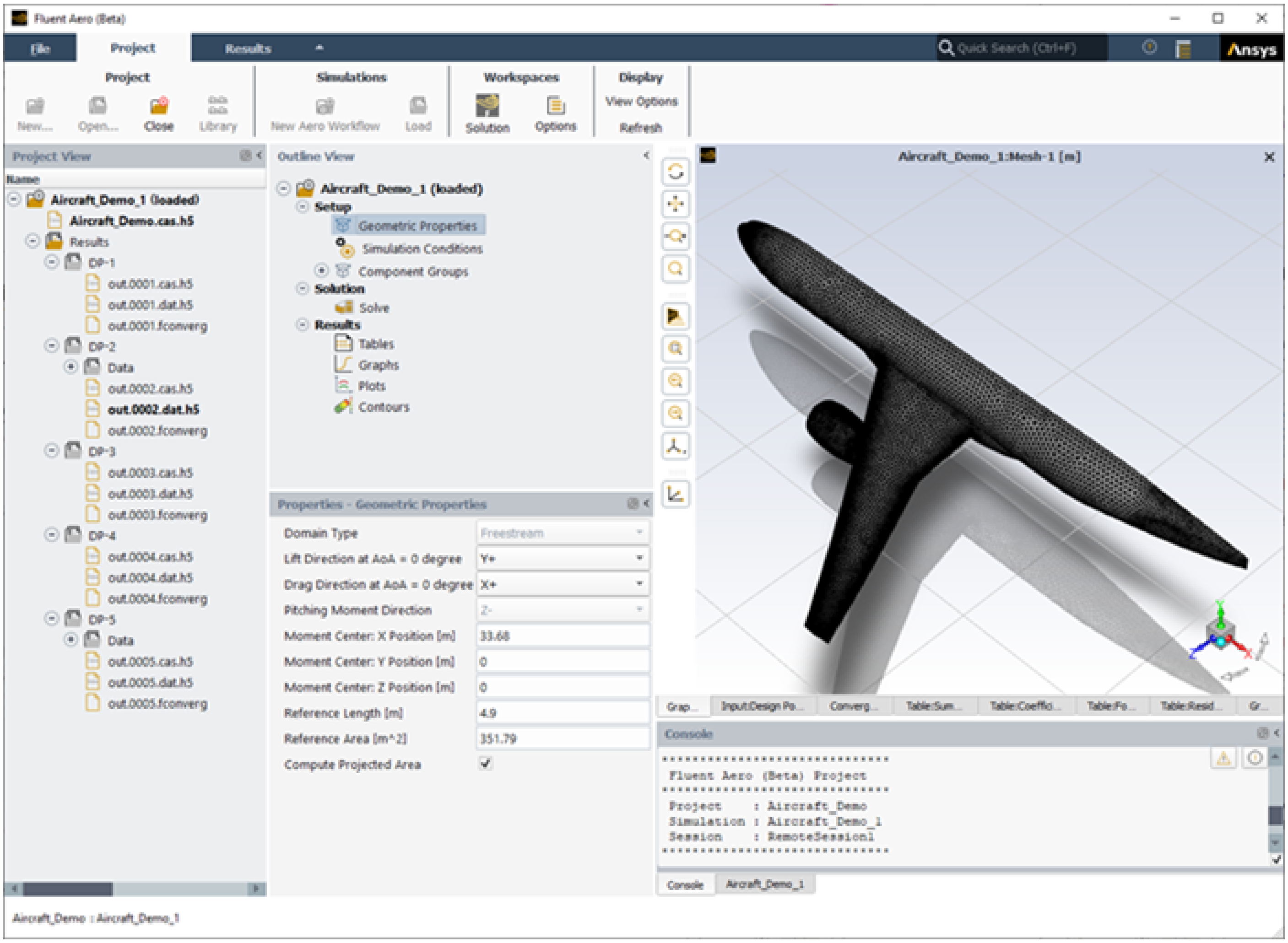
- CFD Solver Level 2 (Premium)
- HPC tokens linked to the number of CPUs requested

The Fluent Aero application does not consume license keys and can be used to invoke external post-processors. A single post-processor can be launched from Fluent Aero and share the license seat as the Fluent solver. Additional post-processing tools will consume their own license seat.

## 33.5. Fluent Aero Graphical User Interface

The Fluent Aero graphical user interface consists of the following components.

**Figure 33.1: The Fluent Aero Graphical User Interface**



- **File** (Top-Left)

  Consists of file management commands (**Open Project...**, **Save Case**, **Import file...**, etc.)

- **Ribbon** (Top)

  – **Project**

  → Project management commands

  → Simulation management commands

  → Workspaces management commands

  → Display options

– **Results**

→ Select post-processing commands and quick-view actions

- **Project View** (Left)

  The left side panel shows the **Project View** by default. Before opening a project, this view shows available and recent projects that can be opened. Once a project is open, this view contains a list of all simulations, runs and output files contained in your Fluent Aero project.

  – **Simulation folders**

  The **Project View** will display a list of Simulation folders, which are created when the **New Aero Workflow...** option is used. If a Solver session is currently loaded, **(loaded)** will be displayed next to the corresponding **Simulation** folder name.

  – **Results folder**

  → If a Simulation has been calculated, a Results folder will appear inside the Simulation folder. All output files of a particular run will be contained within the Results folder.

- **Outline View** (Top-Center)

  The **Outline View** is displayed in the top-center of the user interface, towards the upper right of the **Project View** by default. This view contains the currently loaded case. A loaded case is connected to a loaded Solver session and provides access to:

  – **Setup**

  Contains steps to setup and organize your geometry and specify reference parameters and simulation conditions.

  – **Geometric Properties**

  Settings to define the geometry orientation and reference sizes.

  – **Simulation Conditions**

  Settings to define the freestream flight conditions of each design point of your aerodynamic simulation (such as altitude, Mach number, etc.)

  – **Component Groups**

  Settings to organize your mesh boundary zones into groups of aircraft components (such as **Engine**, or **Wing** components) and to define group or boundary zone specific conditions (such as engine exhaust temperature and mass flow).

  – **Solution**

  Contains steps to setup your CFD solver parameters.

  – **Solve**

  Settings to define and start the calculation process including hidden advanced solver and convergence parameters, if necessary.

– **Results**

  Access to quick post-processing features to explore the results of each design point.

- **Properties window** (Bottom-Center)

  Contains the settings of a selected item in the **Outline View**.

- **Graphic windows** (Right)

  Displays mesh, numerical, post-processed and monitoring data.

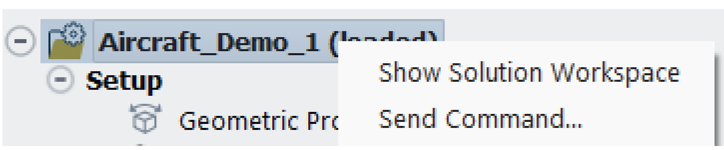
- **Console window** (Bottom-Right)

  Displays the progress of your simulation and allows scripting of commands. The default console is a Python console.

---

**Note:**

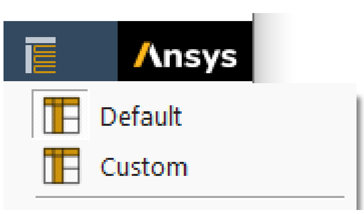
In the case of Solver sessions, each session has a separate console window, which displays a Fluent solver execution log. This console cannot be used for commands. To send a text command to the Solver, right-click the simulation case in the **Outline View** and choose **Send Command...**




---

## 33.5.1. Layout Menu

The top-right corner of the main window contains a layout menu. This menu permits to toggle between different panel distributions. Upon start-up, Fluent Aero will setup these panels using the view that has previously been selected.

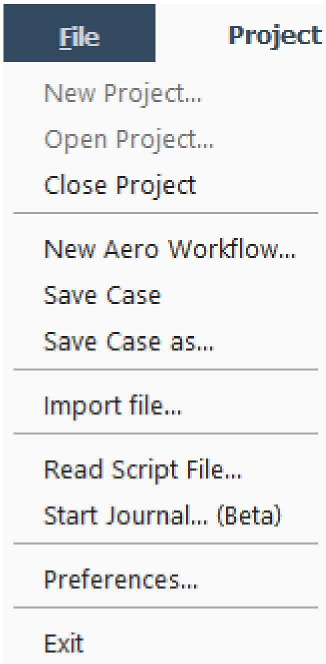



- **Default**: The default view of the panels of Fluent Aero.

- **Custom**: If selected, the window layout is automatically saved when you close your project and restored the next time you launch Fluent Aero.

## 33.5.2. File Menu

The Fluent Aero **File** menu consists of the following options.

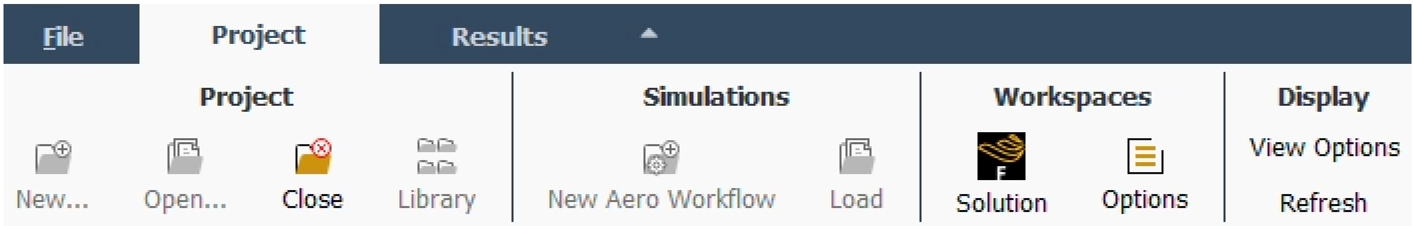



- **New Project...**: Creates a new Fluent Aero project.
- **Open Project...**: Opens an existing project.
- **Close Project**: Closes a project.
- **New Aero Workflow...**: Imports a Fluent case or mesh file, creates a new simulation, and loads a solver session.
- **Save Case**: Saves the current settings to the Fluent case file stored in the Simulation.
- **Save Case as...**: Saves the current settings to a new case file. The new case file must be stored in the same simulation folder and will become the default case file of the simulation.
- **Import file...**: Copy a file external to the project, into the current simulation folder.
- **Read Script File...**: Selects a python script file to execute.
- **Start Journal... (Beta)**: Records a journal file of supported commands.
- **Preferences...**: Opens the **Preferences** window. The **Aero** preference window is specific to Fluent Aero.
- **Exit**: Exits Fluent Aero. All Solver sessions will also be terminated.

### 33.5.3. Ribbon Commands

There are two ribbon commands: **Project** and **Results**.

- **Project**




– **Project**

→ **New...**: Creates a new Fluent Aero project.

→ **Open...**: Opens an existing project.

→ **Close**: Closes a project.

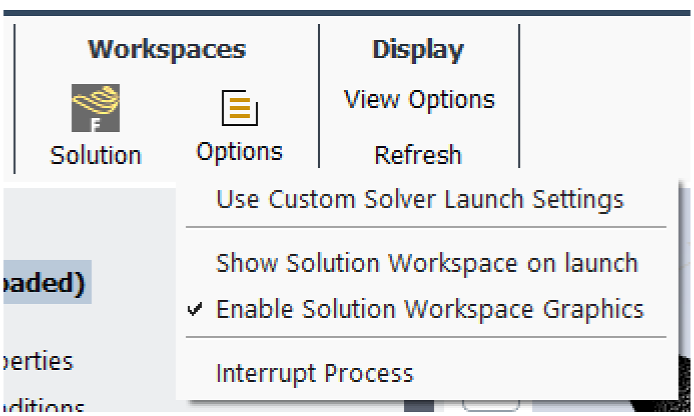
→ **Library**: Displays a list of previously opened projects. Double clicking on a project, inside this list, opens it.

– **Simulations**

→ **New Aero Workflow**: Imports a new case file and begins a new aero workflow simulation.

→ **Load**: Loads the simulation that has been selected in the **Project** panel.

– **Workspaces**




→ **Solution**: Displays the Fluent solver window.

– **Options**: Settings related to the Fluent solver workspace.

→ **Use Custom Solver Launch Settings**:

When enabled, the `Import case` or `Load Simulation` command will first open a **Fluent Launcher** window, where additional settings can be applied to determine how the Solver is launched. This allows the user to load the Solver on a different machine than where Fluent Aero is open, or to use a job scheduler to launch the Solver on a cluster with a queuing system or with a different number of CPUs than what is specified as default. This option is also accessible in the **Preferences** panel.

→ **Show Solution workspace on launch**:

The Fluent solver window will be shown as the solver is launched.

→ **Enable Solution Workspace Graphics**:

With this option enabled, the Fluent window will also have a graphics window.

→ **Interrupt process**:

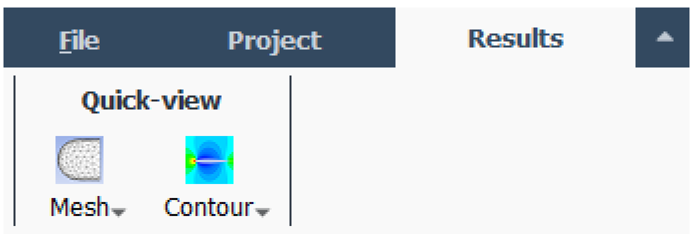
This terminates the current Fluent solver and stops any process on the local or remote machine.

– **Display**

→ **View Options**: Provides options to change the display in the **Project View**.

→ **Refresh**: Refreshes the Fluent Aero user interface.

- **Results**

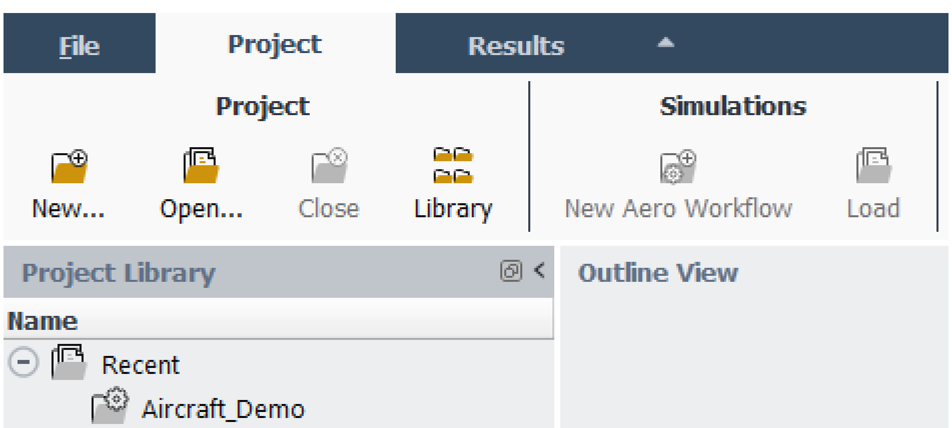



– **Quick-view**: Accesses the post-processing shortcuts.

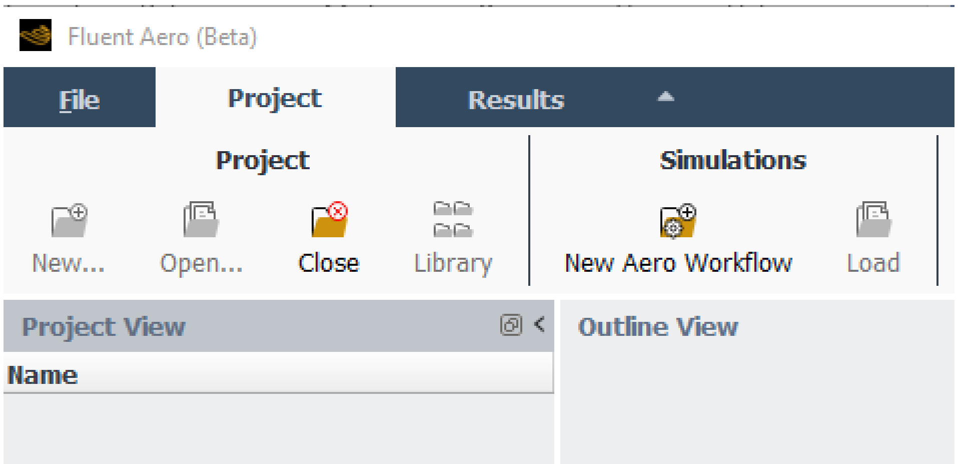
# 33.6. Creating or Opening a Fluent Aero Project

## 33.6.1. Creating a Fluent Aero Project

Upon startup, the Fluent Aero application has no Project loaded, and presents an empty Project window.

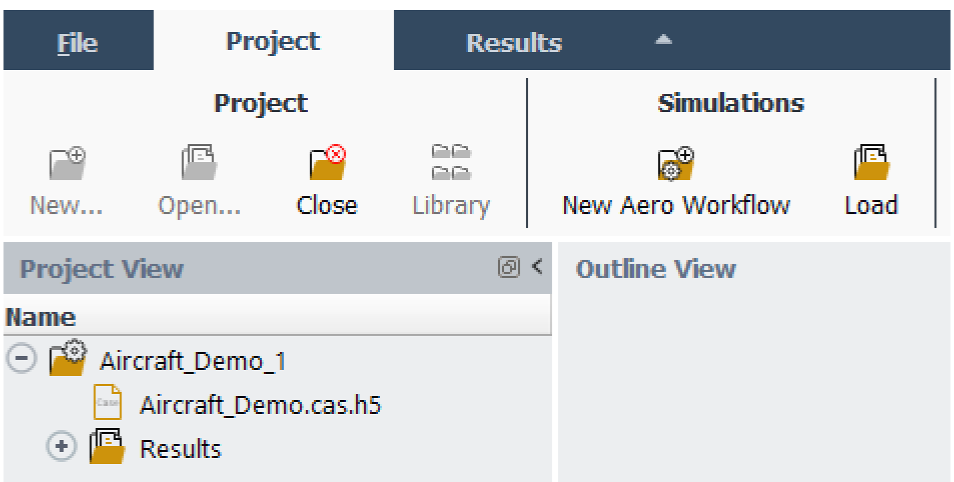


**Project** → **New...** or **File** → **New Project...** creates a new project. A window will appear prompting the user to set a location and name for the new project folder. A project file (`projectname.flprj`) and folder (`projectname.cffdb/`) is then written to the disk in the location selected, and the new project opens in Fluent Aero. Since this is a new project, **Project View** remains empty. Once the project is open, the Simulations commands become available.




## 33.6.2. Opening a Fluent Aero Project

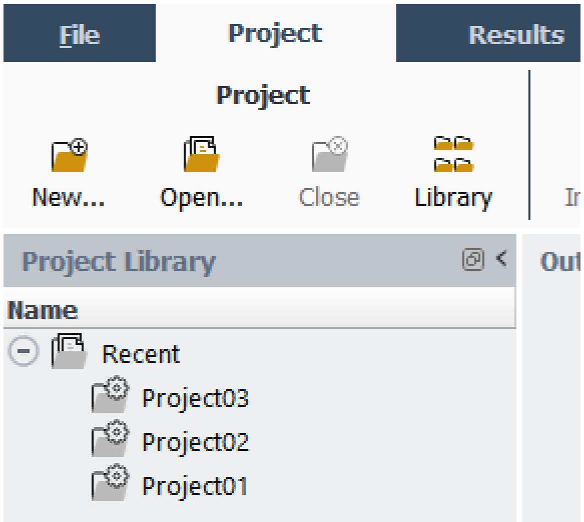
**Project** → **Open...** or **File** → **Open Project...** opens a preexisting Fluent Aero project. A window appears allowing the user to navigate to and select a preexisting project file (`projectname.flprj`). All Simulations contained within the project will be displayed in the left-side **Project View**.



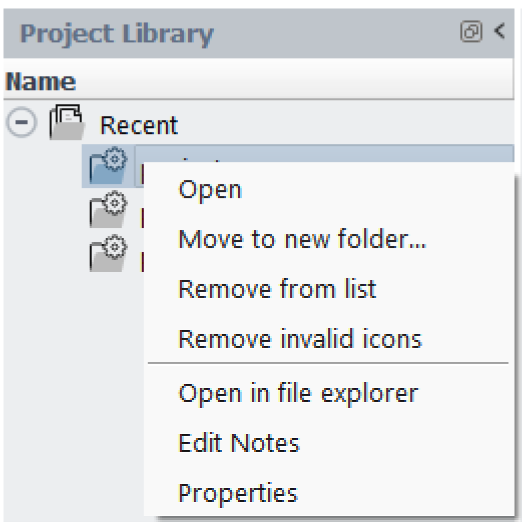

After opening a **Project**, the project name will be displayed in the bottom left corner of the Fluent Aero window.

### 33.6.3. Project Library

**Project** → **Library** displays a list of projects that were previously opened in Fluent Aero. Double-clicking a project in this list opens it.




In the **Project Library**, newly opened projects are added to the **Recent** folder.




The following options are available when right-clicking a given project in the **Project Library** window:

- **Open**: Opens the project.
- **Move to new folder...**: Creates a new folder in the **Project Library**, and moves the project to that folder. The project location on the disk is unchanged.
- **Remove from list**: Removes the project from the **Project Library** list. The project is not erased.
- **Remove invalid icons**: If a project is moved or erased on the disk, it will display as a broken icon, which is an invalid icon. This command will remove all invalid project references from the **Project Library**.

- **Open in file explorer**: Opens a Windows file explorer to the project location.
- **Edit Notes**: Opens the **Properties** panel for the selected item in the **Notes** section. The **Notes** section permits to record text notes for a project item. If a project item has a note attached to it, it will display with a * in the file name suffix. **Ctrl** double-click also permits to display the **Notes** panel directly.
- **Properties**: Opens a window that shows the project location in the disk. This window also allows you to write some notes for future reference.
- **Library management:**
  - Drag and drop projects from one folder to another, to move them across categories. The project is added at the top of the list.
  - Drag and drop with the **Alt** key, to reorder projects.
  - The contextual menu (right-click) of folders that contain projects permits to sort the folder content by **Sort by name**.

> **Note:**
>
> Each release uses its own **Project Library** to track it's recently used projects. However, the **Project Library** from the previous releases will be initially imported as a new folder called **-Imported-** in the **Project Library**, so that older projects can be easily accessed.

### 33.6.4. Project Close

To close a currently open **Project**, select **Project** → **Close** or **File** → **Close Project...**

## 33.7. Creating or Loading a Fluent Aero Simulation

To create a Simulation, a Fluent `.cas[.h5]` or a Fluent `.msh[.h5]` file must be loaded. This file has certain requirements to be compatible with a Fluent Aero simulation.

### 33.7.1. General Case File or Mesh File Requirements

The Fluent `.cas[.h5]` or `.msh[.h5]` file must respect the following requirements:

- 3D only.

  2D Fluent mode is not supported. 2D geometries are supported via 3D meshes with symmetry or periodic planes.
- If you would like to simulate a **Freestream** type domain, a pressure-farfield boundary type or a pressure-farfield with pressure-outlet boundary type should be used to define the main farfield flow boundary. See more details in Freestream or WindTunnel Domain Type Requirements (p. 451).

- If you would like to simulate a **WindTunnel** type domain, a velocity-inlet boundary type should be used to define the inlet flow and a pressure-outlet boundary type should be used to define the outlet flow. See more details in Freestream or WindTunnel Domain Type Requirements (p. 451).

---

**Note:**

It is beneficial for the boundary zone names and types to be set up properly inside a Fluent Solution Workspace or a Fluent Meshing Workspace before importing the `.cas[.h5]` the `.msh[.h5]` file into Fluent Aero. Doing so will help Fluent Aero automatically setup the domain type as it imports the domain.

---

## 33.7.2. Freestream or WindTunnel Domain Type Requirements

Fluent Aero supports two domain types, **Freestream** and **WindTunnel**, each with a different set of requirements for the input mesh or case file. Furthermore, if the boundary zones are set up as recommended, Fluent Aero will be able to automatically determine the domain type and automatically setup the Simulation and its associated **Freestream** or **WindTunnel** component group.

### 33.7.2.1. Freestream

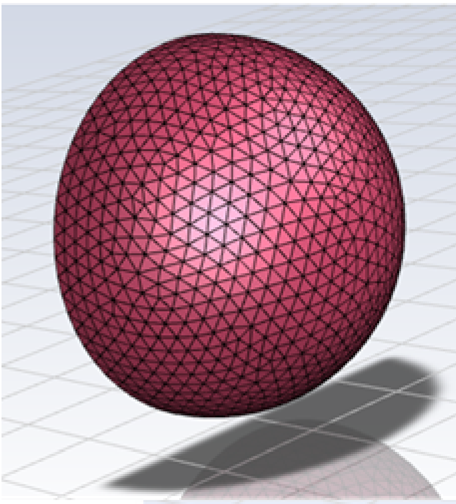
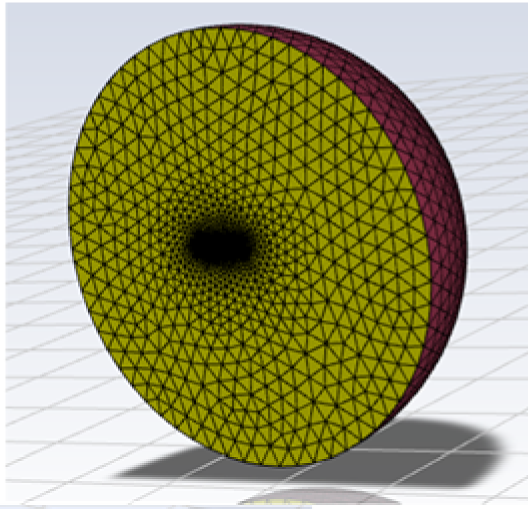
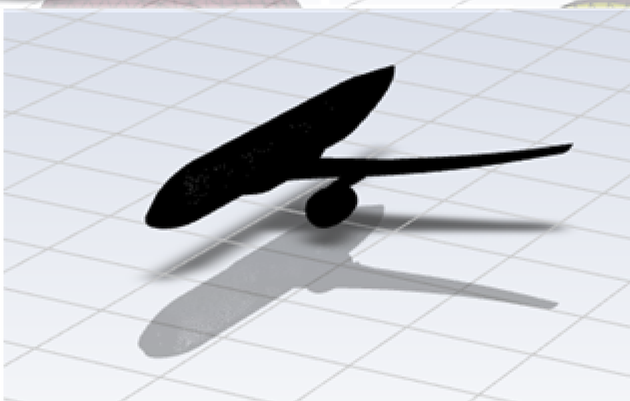
A **Freestream** domain type is used to simulate the airflow around an aircraft during typical in-flight conditions. The computational domain is setup such that its external boundaries are located far enough away from the aircraft model so that their presence does not impact the computational solution near the aircraft model. Typically, the distance between the aircraft model surface and the external boundaries is recommended to be at least 30 times the characteristic length of the aircraft.

#### Freestream Domain Type Requirements

Fluent Aero requires the following restrictions for the external boundary zones:

- At least one **pressure-far-field** type boundary zone must be used to define the external freestream boundary of the domain.
- A **pressure-outlet** type boundary zone may be used to define the general outlet flow region of the freestream domain. However, it must be attached to a pressure-far-field.
- A **symmetry** type boundary zone may be used to represent a symmetric geometry.

In the images below, a hemispherical pressure-far-field type boundary zone (red) with a circular symmetry type boundary zone surrounds the wall zones (grey) defining and aircraft geometry. In this case, no pressure-outlet type boundary zone is used to define the freestream domain.

### Automatically Setup of a Freestream Group

Once Fluent Aero loads a mesh or case file, it will attempt to automatically determine if a **Freestream** domain type is being used.

- First, it will search for a **pressure-far-field** type boundary zone that defines the freestream boundary of the domain. If one is found, Fluent Aero will determine that the user is importing a **Freestream** type domain, and the pressure-far-field zone will be added to the **Freestream** group.

- Next, it will search for a **pressure-outlet** boundary that is attached to the pressure-far-field, and if found, this boundary will also be added to the **Freestream** group.

A console message will appear to summarize the auto domain type setup. If there were any issues in setup, a warning will be printed, and the user will have the opportunity to modify the **Freestream** group manually.

```
Console

------------------------------------------------------------
-                   Simulation Domain Type                 -
------------------------------------------------------------
Fluent Aero has been set to use a Freestream domain type.

- The following boundary zones have been detected:
  > Freestream Inlet (type: pressure-far-field)
    - farfield
  > Freestream Outlet, not required (type: pressure-outlet):
    - None

- The Domain Type will be set to Freestream,
  and the group will be created in Component Groups
-----------------------------------------------------------
```

The **Domain Type** will be set to **Freestream** in the **Geometric Properties** step.

| Properties - Geometric Properties | |
|---|---|
| Domain Type | Freestream |

And, the **pressure-far-field** type zone and **pressure-outlet** type zones will be added to the **Freestream** group under **Component Groups**.


### 33.7.2.2. WindTunnel

A **WindTunnel** domain type is used to simulate the airflow around an aircraft model inside an experimental wind tunnel test section. The computational domain is setup such that its external boundaries appropriately match those used for an experimental wind tunnel test, which may be located close enough to the aircraft model so that their presence has some impact on the computational solution near the aircraft model. The object of such a simulation would be to match as closely as possible the wall-effects associated with the wind tunnel test.

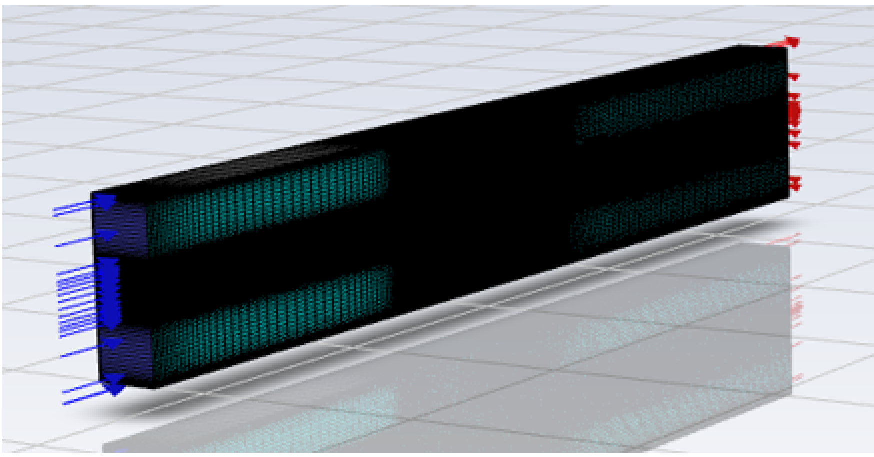
#### WindTunnel Domain Type Requirements

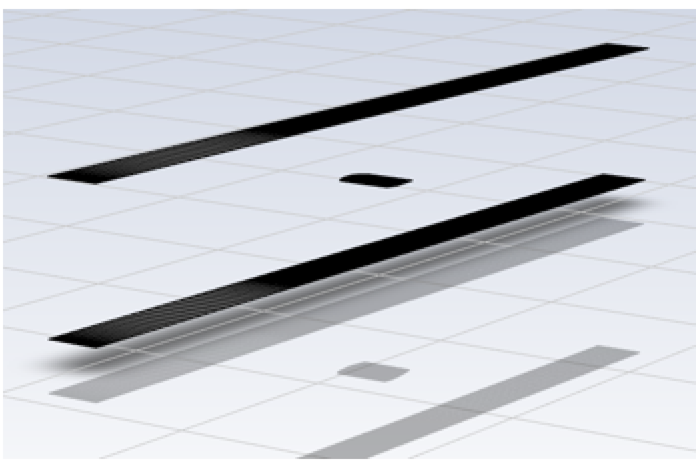
Fluent Aero requires the following restrictions for the external boundary zones:

- A **velocity-inlet** type boundary zone must be used to define the wind tunnel inlet flow upstream of the aircraft model. It is recommended that this boundary be named **windtunnel-inlet** so that the automatic setup can register the boundary.

- A **pressure-outlet** type boundary zone may be used to define the wind tunnel outlet flow downstream of the aircraft model. It is recommended that this boundary be named **wind-tunnel-outlet** so that the automatic setup can register this boundary.
- **wall** or **symmetry** or **periodic** type boundary zones may be used to represent the wind tunnel walls. It is recommended that this boundary be named **`windtunnel-wall*`**, **`windtunnel-symmetry*`** so that the automatic setup can register these boundaries.

In the images below, a **velocity-inlet** type boundary zone (blue), a **pressure-outlet** type boundary zone (red), **wall** type (grey) and **periodic** type (teal) boundary zones define the wind tunnel domain while additional **wall** type (grey) boundary zones define an airfoil inside the test section.





## Automatic Setup of a WindTunnel Group

Once Fluent Aero loads a mesh or case file, it will attempt to automatically determine if a **WindTunnel** domain type is being used.

- First, it will search for any boundary that has **windtunnel** in its name (case insensitive and can include "-" in the name). If a boundary is found, Fluent Aero will determine that the **WindTunnel Domain Type** is being used.
- Next, it will search for a **velocity-inlet** with **windtunnel** in its name, and add that boundary to the **WindTunnel** group.

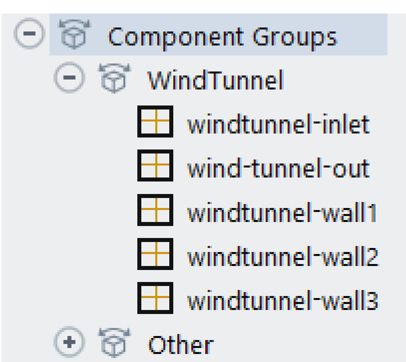
- Next, it will search for a **pressure-outlet** with **windtunnel** in its name, and add that boundary to the **WindTunnel** group
- Finally, it will search for any **wall** or **symmetry** type boundaries with **windtunnel** in its name, and add those boundaries to the **WindTunnel** group.

A console message will appear to summarize the auto domain type setup. If there were any issues in setup, a warning message will be printed, and the user will have the opportunity to modify the **WindTunnel** group manually.


```
----------------------------------------------------------
-                  Simulation Domain Type                 -
----------------------------------------------------------
Fluent Aero has been set to use a Wind Tunnel domain type.

- The following boundary zones have been detected:
  > Wind Tunnel Inlet (type: velocity-inlet):
    - windtunnel-inlet
  > Wind Tunnel Outlet (type: pressure-outlet):
    - wind-tunnel-out
  > Wind Tunnel Sides (types: symmetry, wall):
    - windtunnel-wall3
    - windtunnel-wall2
    - windtunnel-wall1

- The Domain Type will be set to WindTunnel,
  and the group will be created in Component Groups
----------------------------------------------------------
```

The **Domain Type** will be set to **WindTunnel** in the **Geometric Properties** step.

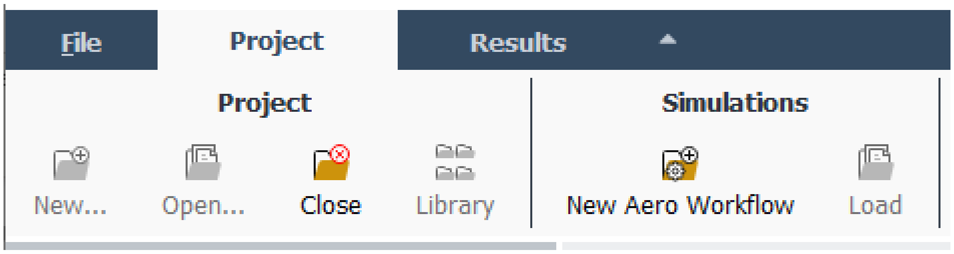

And, the **windtunnel-*** named zones will be added to the **WindTunnel** group under **Component Groups**.



### 33.7.3. Creating a New Simulation by Using New Aero Workflow

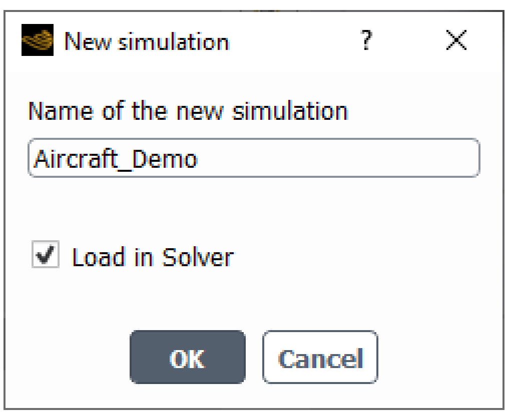
If a new project has been created, the Fluent Aero application has no case loaded and presents an empty **Project View** window. A new Fluent Aero Simulation can be created by clicking the **New Aero Workflow** button.

**Figure 33.2: New Aero Workflow**




Select **Simulation** → **New Aero Workflow** to import a case file and create a new simulation. A window will appear where a case file (`.cas`, `.cas.h5`) or mesh file case file (`.msh`, `.msh.h5`) can be selected.

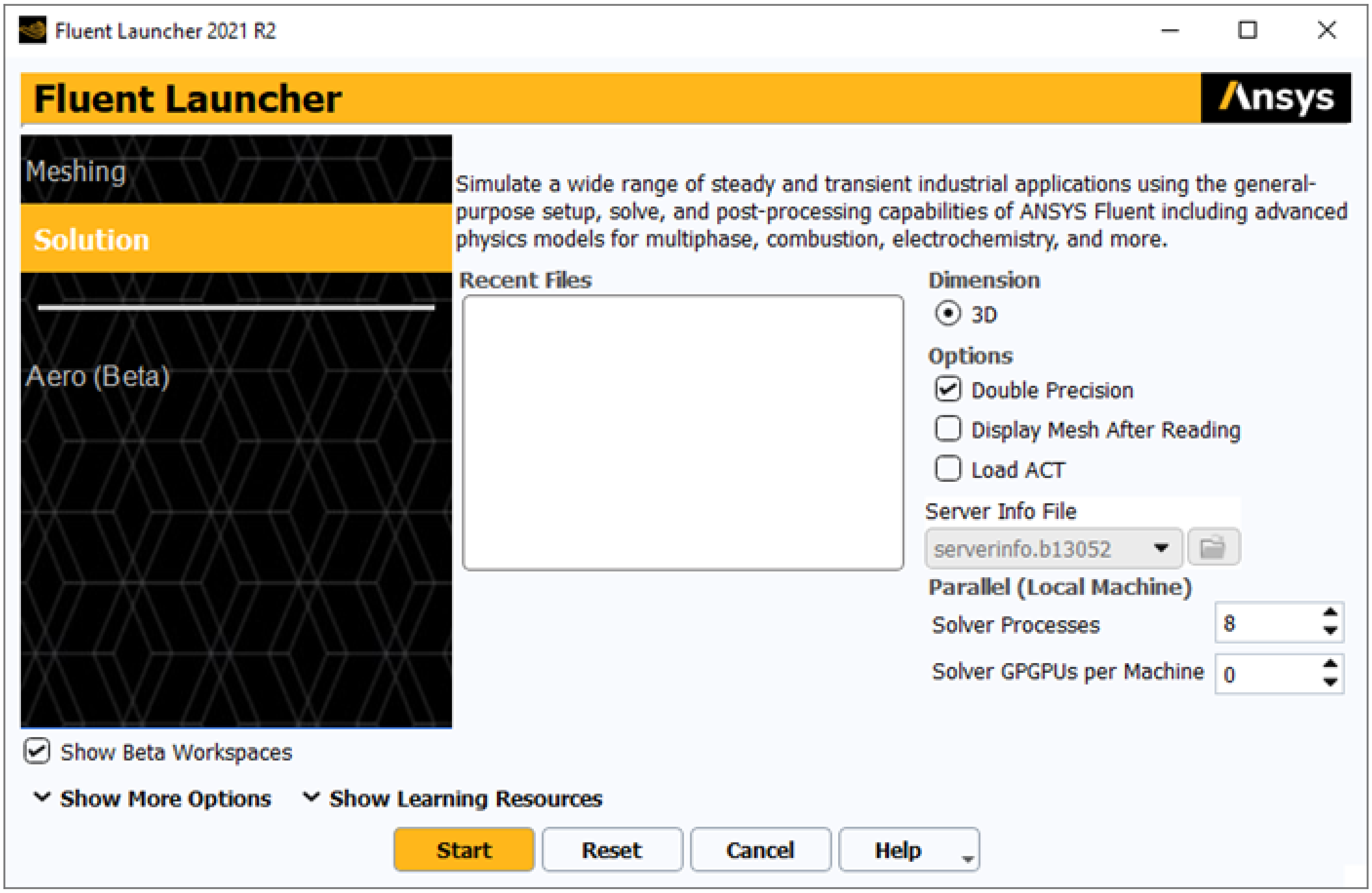
After a case file has been selected, a window will appear requesting a name for the new Simulation. Clicking **OK** will create a new Simulation in the currently open Project folder. The case file will be copied into the new Simulation folder and will be used as the primary case file for that Simulation.



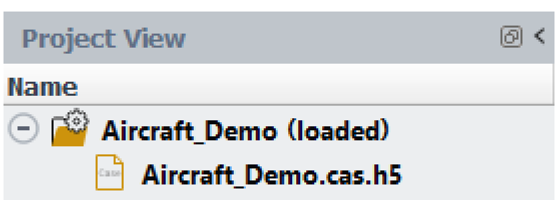

- **Load in solver**: If enabled, the solver is loaded with the selected case file. If disabled, the solver is not loaded, and you can choose to load the solver at a later time.

If **Use Custom Solver Launch Settings** is disabled in **Preferences**, the Solver will be loaded on your local machine, and no further input will be required.

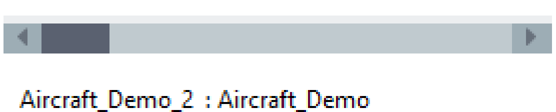
However, if **Use Custom Solver Launch Settings** is enabled in **Preferences**, a **Fluent Launcher** window will appear. If so, set the appropriate number of **Solver Processes**, and if necessary click **Show More Options** to specify the **Remote Machine** or **Job Scheduler** and click **Start**.




When the Import case is complete, the new Simulation folder appears in your Project view. This folder contains the case file that has been imported. If **Load in Solver** was selected, a Solver session opens in the background, and the Simulation folder is listed with **(loaded)** next to its name.

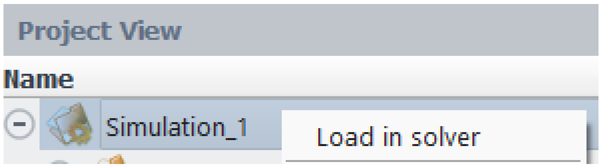



The Simulation that's currently selected will be displayed in the bottom left corner of the Fluent Aero window, next to the currently open Project.



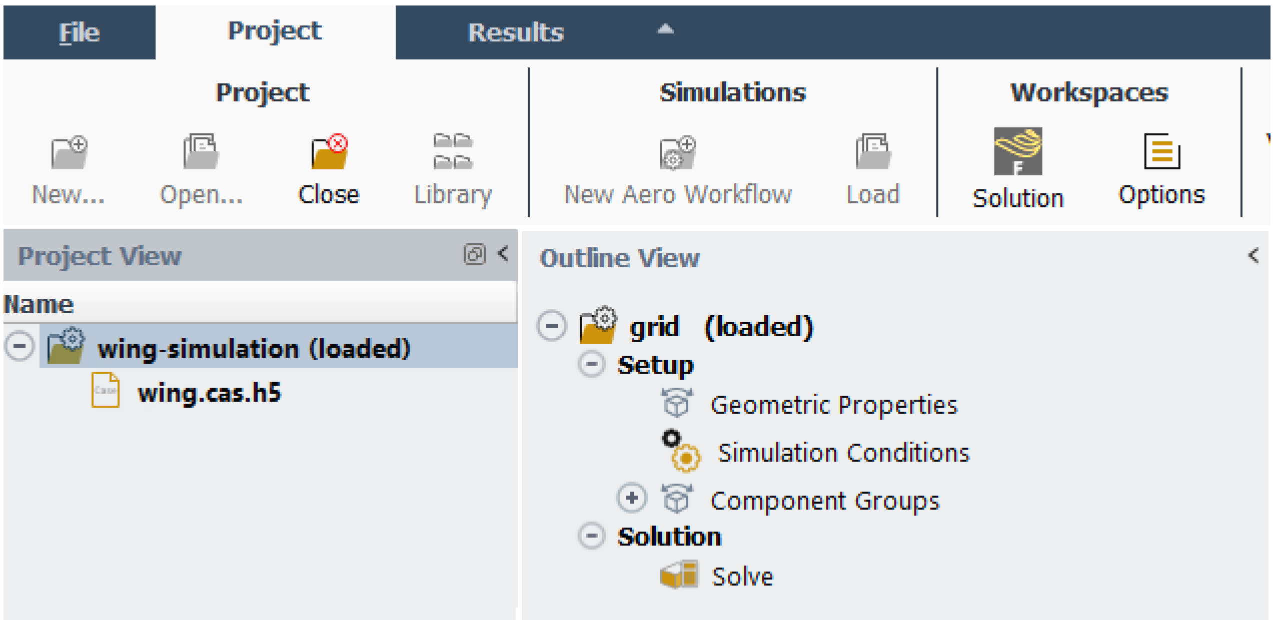

## 33.7.4. Loading a Simulation

From the Project view, you can load a simulation by right clicking on a Simulation folder and selecting **Load in solver**, by double-clicking it, or by using **Project** → **Simulations** → **Load**.




By default, **Load in solver** will launch a Solver on the same machine where Fluent Aero is open with your default number of CPUs. Alternatively, if **Use Custom Solver Launch Options** has been enabled in the **Preferences**, a **Fluent Launcher** window will appear allowing the specification of different options to launch the solver.

Once the Simulation is loaded, **(loaded)** will appear next to the Simulation folder name in your **Project View**, and the Aero Workflow steps will be displayed in the **Outline View**.



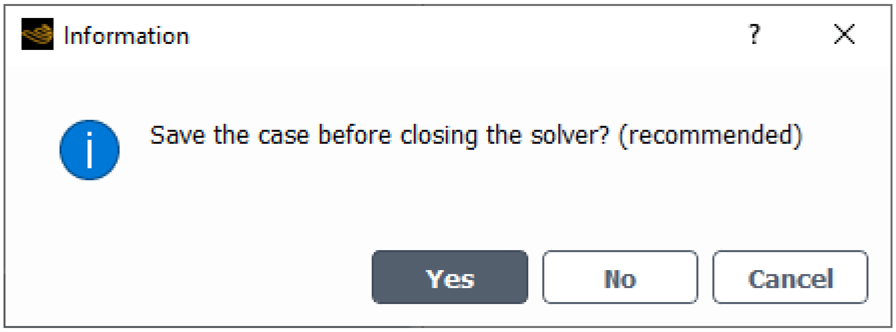

## 33.7.5. Closing a Simulation

From the **Project View**, you can close a Simulation by right-clicking on a Simulation folder and selecting **Close Solver**.

From the **Outline View**, you can close a Simulation by right-clicking on a Simulation folder and selecting **Close Solver**.

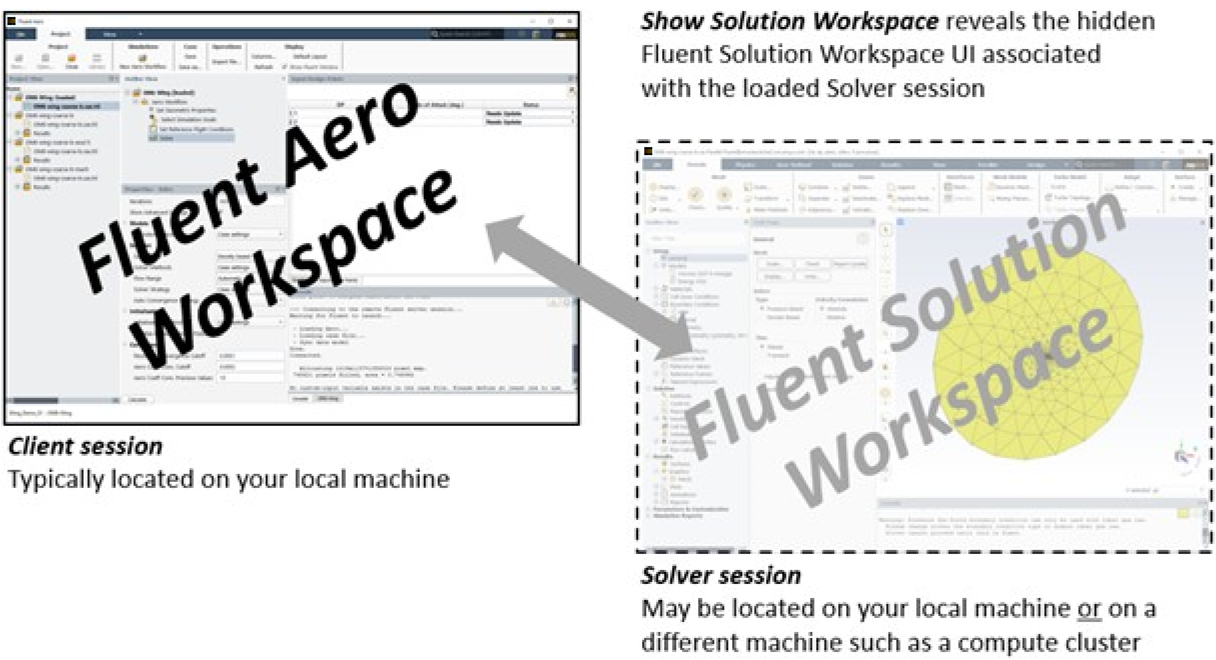
In both cases, the Solver and the Simulation will be closed and (loaded) will no longer appear next to the Simulation.

A panel may appear asking if you would like to save the case file before closing. Selecting **Yes** is generally recommended, as it ensures that your Simulation's case file will be updated using any recent changes to the simulation.

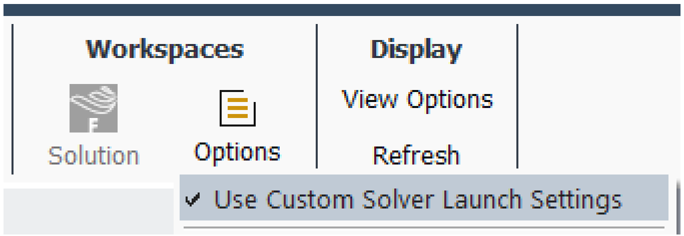



## 33.7.6. Fluent Aero Workspace, the Solver Session, and the Fluent Solution Workspace

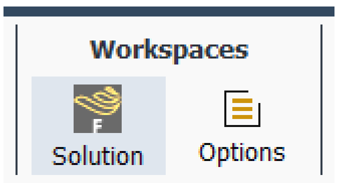
The image below visually represents the interaction between the Fluent Aero Workspace, the Solver session and the Fluent Solution Workspace.




When you first open Fluent Aero, the user interface window that is presented is the Fluent Aero Workspace. However, when you create or load a **Simulation** using **New Aero Workflow** or **Load in Solver**, a *Solver session* will be loaded. By default, this Solver session will be launched and your case file will be loaded in the background on your local machine – the same machine where your Fluent Aero Workspace (or Client session) is located. However, it is also possible to launch the Solver on a different machine or with a different number of processes. In this way, it is possible for the user to have the lightweight Fluent Aero Workspace (where the majority of the user interaction takes place) located on a local machine with a more capable graphics display, but have the Solver session (where the majority of the computational work takes place), located on the compute node of a cluster, which may have more CPU power and memory, but not have access to a graphics display. If a user would like to launch the solver on a different machine, they may do so by enabling **Use Custom Solver Launch Options**. This option can be found in **Project** → **Workspaces** → **Options** or from **File** → **Preferences** → **Aero**.



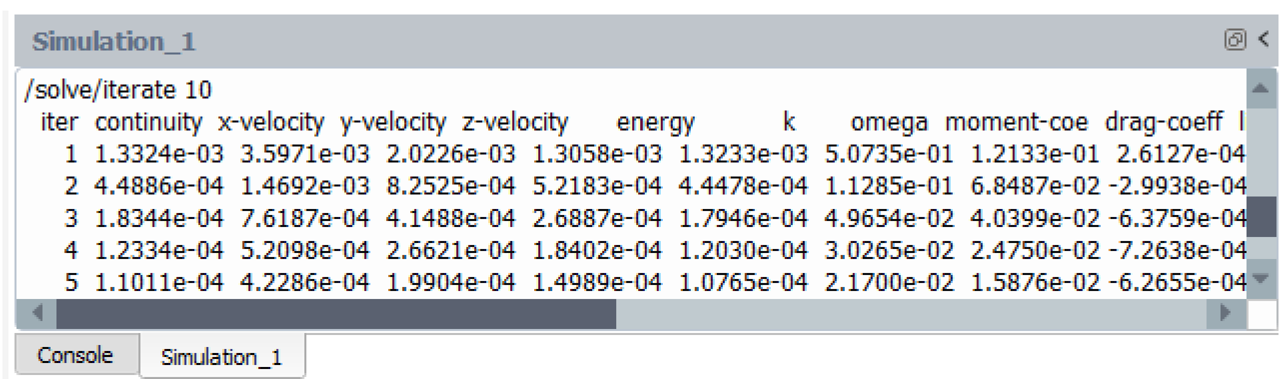

Furthermore, just like the Fluent Aero Workspace represents the graphical user interface for the Client session, the Fluent Solution Workspace represents the graphical user interface for the Solver session. Typically, the Fluent Solution Workspace is hidden, because its interface is not required to setup and interact with Fluent Aero simulations. However, if you would like to interact with the Fluent Solution Workspace, for example to setup more advanced settings that are not available from within the Fluent Aero Workspace, you may do so by clicking **Project** → **Workspaces** → **Solution**.




---

**Note:**

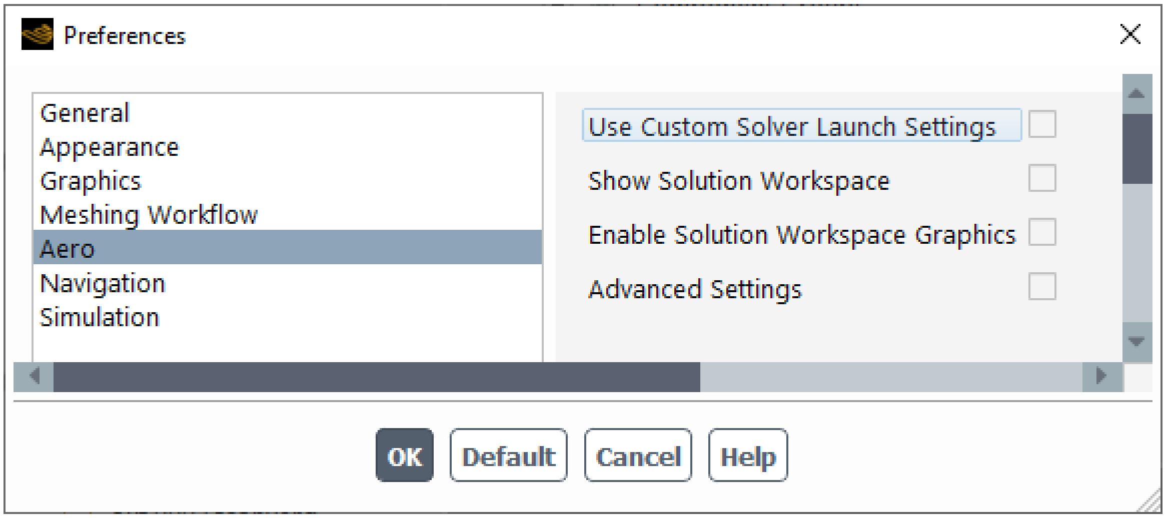
Each Solver session creates a separate console window tab inside Fluent Aero Workspace which displays the solver execution log. The title of this console window will be the same as the title of your Simulation. This is different than Fluent Aero Workspace's Client session console, which is called Console. The Solver session console cannot be used for commands. Instead, to send a text command to the Solver session, right-click on the Simulation in the **Outline View** and choose **Send Command...**

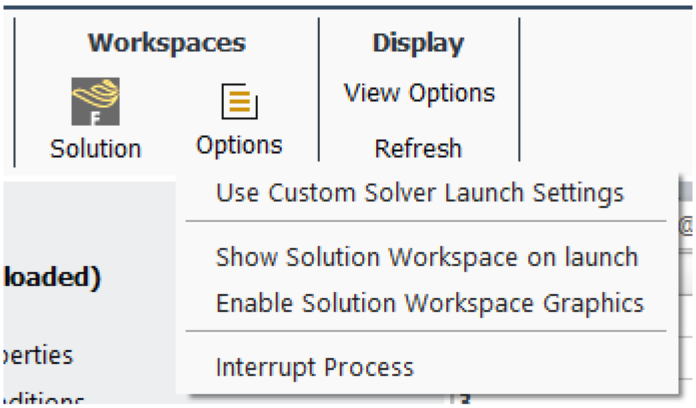



---

## 33.7.6.1. Preferences Related to the Solver Session and the Solution Workspace

Preferences related to the Solver session and the Solution Workspace are available in **Project** → **Workspaces** and in **File** → **Preferences...** → **Aero**.



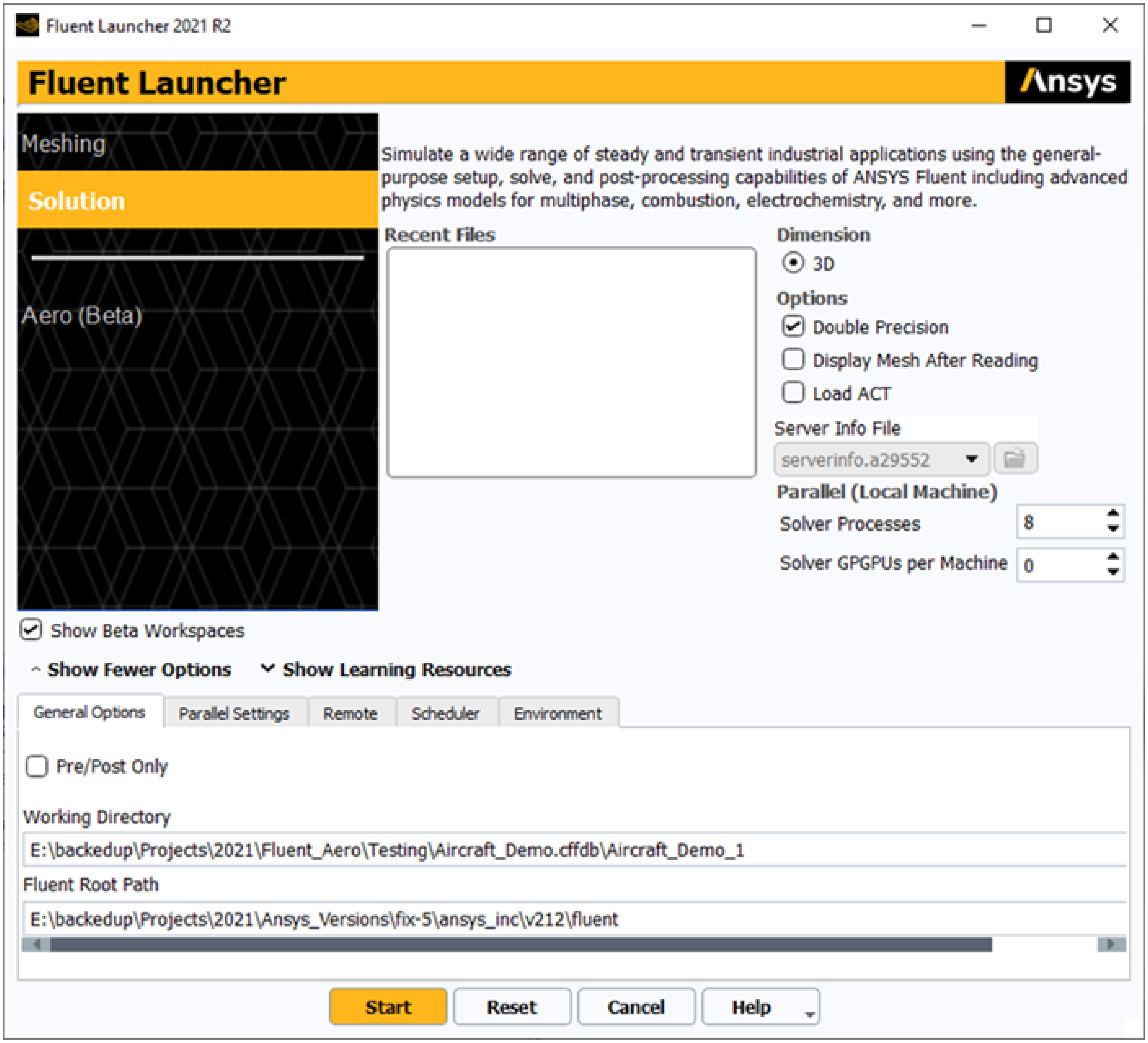




- **Use Custom Solver Launch Settings**

  By default, when you create or load a **Simulation** using **New Aero Workflow** or **Load in Solver**, the Fluent Aero **Simulation** will load a Solver session that is launched in the background on the same machine and with the same number of processes you used to launch Fluent Aero. However, it is also possible to use different settings to launch the Solver on a different machine or with a different number of processes.

  To do this, go to **File** → **Preferences...**, select **Aero**, and enable **Use Custom Solver Launch Settings**.

  If **Use Custom Solver Launch Settings** is enabled, the **New Aero Workflow** or **Load in solver** command will first launch a Fluent Launcher window, where additional settings can be applied to determine how the Solver is launched. This allows the user to load the Solver session on a different machine than where Fluent Aero is open, or to specify the **Remote Machine** or use a **Job Scheduler** (within **Show More Options**) to launch the Solver on a cluster with a queuing system or with a different number of CPUs than what is specified as default.




When you **Use Custom Solver Launch Settings** to **Load in Solver**, it is required to set the following settings in the **Fluent Launcher** that appears:

– **Mode** → **Solution**

→ The Solver session must be launched in Solution mode, as Fluent Aero is expecting to connect to a Solution Workspace Solver session in the background.

– **Dimension** → **3D**

- **Options** → **Double Precision**

**Note:**

When Fluent Aero is launched using the `bin/aero` (linux) or `bin/aero.bat` (windows) file, the **Fluent Launcher** will always appear when using **New Aero Workflow** or **Load in solver** regardless of the setting specified in the **Preferences** panel.

- **Solution** (button) or **Show Solution Workspace**

  When this option is disabled, the Fluent Solution Workspace associated with the Solver session will be hidden. Enabling this option will cause a Solution Workspace window to be shown, allowing the use of advanced settings that may not be available in the Fluent Aero Workspace by itself.

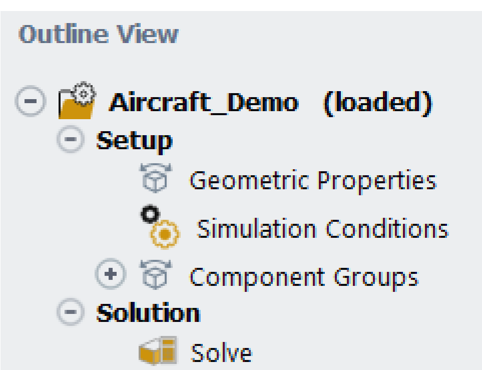
- **Enable Solution Workspace Graphics**

  When this option is disabled, the Fluent Solution Workspace associated with the Solver session will have its post-processing graphics window disabled. Enabling this option will cause a Solution Workspace graphics window to be shown, allowing the user to make post-processing graphical plots within the Solution Workspace window.

## 33.8. Setting up a Fluent Aero Simulation

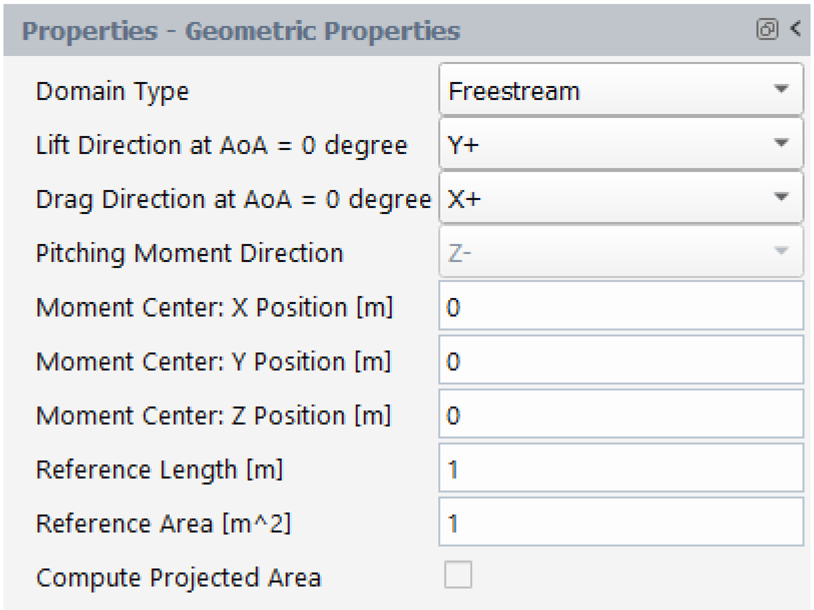
Once a Simulation is loaded, a simulation tree appears in the **Outline View**. The name of the Simulation you have loaded appears at the top of the tree along with the text **(loaded)**. Underneath the name of the Simulation, the **Outline View** tree is displayed.

The **Outline View** tree contains steps that guide a user through the setup of Fluent Aero simulation. These steps should be worked through sequentially to completely setup the problem. This tree is composed of four steps: three steps under **Setup** (**Geometric Properties**, **Simulation Conditions**, **Component Groups**), and one step under **Solution** (**Solve**).




### 33.8.1. Geometric Properties

The first step in the Fluent Aero Workflow is to define geometric properties, including orientation and reference sizes.




- **Domain Type**

  This setting is used to describe the type of computational domain that surrounds your object of interest. There are two available settings: **Freestream** or **WindTunnel**. If you follow the guidelines for name and boundary type outlined in Freestream or WindTunnel Domain Type Requirements (p. 451), this setting may be set automatically upon importing a case file. This setting cannot be changed after a calculation has been performed.

- **Lift Direction at AoA = 0 degree**

  This setting defines which cartesian direction corresponds to the direction of the Lift vector in the scenario that the geometry is operating at a 0 degree angle of attack. This setting provides Fluent Aero with an on orientation so that the Angle of Attack input variable can be applied correctly. The geometry must be oriented as aligned with a cartesian axis such that a cartesian direction can be selected.

- **Drag Direction at AoA = 0 degree**

  This setting defines which cartesian direction corresponds to the direction of the Drag vector in the scenario that the geometry is operating at a 0 degree angle of attack. This setting provides Fluent Aero with an orientation so that the Angle of Attack input variable can be applied correctly. The geometry must be oriented as aligned with a cartesian axis such that a cartesian direction can be selected.

- **Pitching Moment Direction**

  This setting defines which cartesian direction corresponds to the direction of the Pitching Moment. It is setup automatically depending on the Lift and Drag direction settings.

- **Moment Center: X-, Y-, Z- Positions [m]**

  These settings define the position in meters of the Pitching Moment using the cartesian coordinates of the mesh.

- **Reference Length [m]**

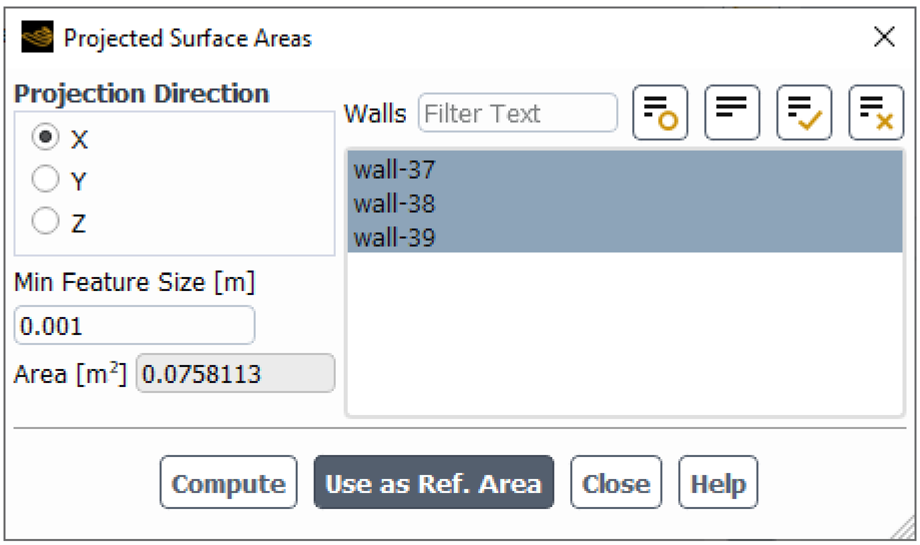
  This setting defines the reference length in units of meters, which is used for the computation of aerodynamic coefficients.

- **Reference Area [m^2]**

  This setting defines the reference length in units of square meters, which is used for the computation of aerodynamic coefficients.

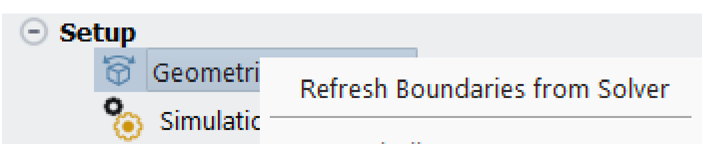
- **Compute Projected Area**

  Enabling this option opens the **ProjectedSurface Areas** tool window that allows the user to compute the reference area of your geometry. Set the **Projection Direction** and select the **Walls** that you would like to use for the reference area computation. Click **Compute**. If you are satisfied with the value that appears in the **Area [m^2]** box, click on **Use as Ref. Area** to fill the **Reference Area [m^2]** box with this value.




The **Set Grid Properties** step also contains the following optional command:

- **Refresh Boundaries from Solver**




  As described in the **Case File Requirements**, it is recommended that the boundary types of a `.cas[.h5]` file have been setup properly in a Fluent Solution Workspace or a Fluent Meshing Workspace before it is imported into Fluent Aero. However, if the boundary types of a `.cas[.h5]` file have not yet been setup before loading into Fluent Aero, the **Show Solution Workspace** command can be used to open the background Solver window, where the boundary types can be redefined. These boundary types can then be updated in Fluent Aero by right-clicking on **Set Geometric Properties** and selecting **Refresh Boundaries from Solver.**

Once the geometric properties have been defined, the user continues to the next step.

## 33.8.2. Simulation Conditions

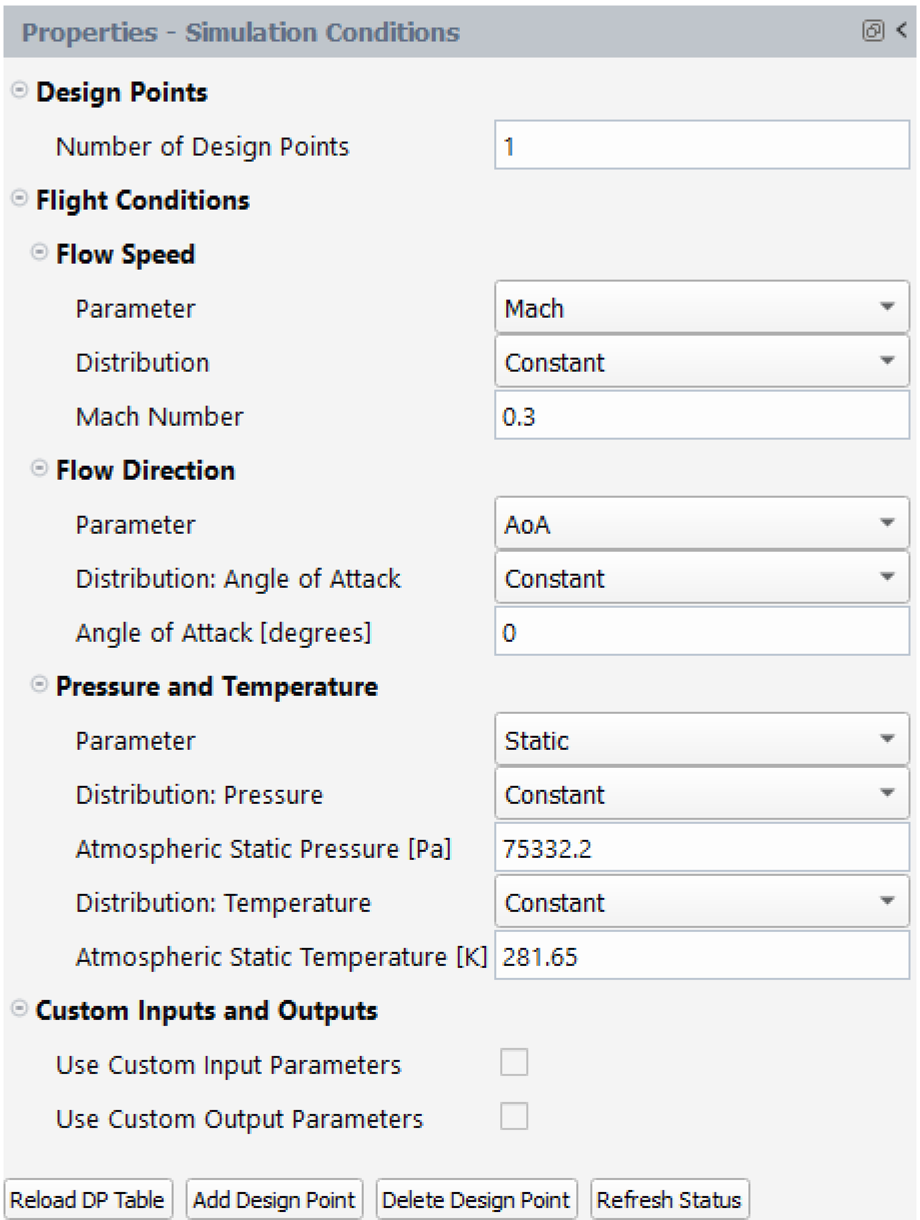
In this step, you define a matrix of flight or wind tunnel conditions (referred to in Fluent Aero as Design Points). For each Design Point, Fluent Aero will calculate and post processing an aerodynamic CFD solution.

---

**Note:**

The conditions applied in this section will be applied to the inlet and outlet boundary zones that are part of the **Freestream** or **Wind Tunnel** component groups.

---




- **Design Points**
  - **Number of Design Points**

The Single Simulation mode is used to investigate the results of a simulation of a single design point using a single flight condition. The **Number of Design Points** entered here will determine the number of rows in the **Input: Design Points** table.

- **Flight Conditions**

  This section contains inputs to define the conditions used to perform your aerodynamic simulations. Settings applied in this section will control the inlet and outlet boundary zones that are part of the **Freestream** or **Wind Tunnel** component groups.

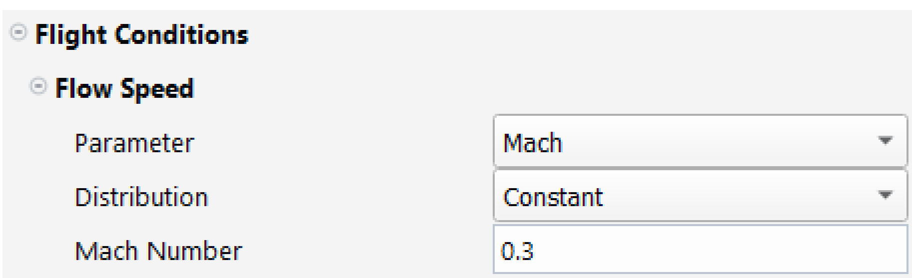
  There are three categories that are used to input these conditions settings: **Flow Speed**, **Flow Direction** and **Pressure and Temperature**. Each category contains; **Parameter** settings - where the specific input parameter associated with that category can be selected, a **Distribution** setting - where the value used for the input can be defined as **Constant** for all Design Points or varied with a **Uniform** distribution or **Custom** distribution for all Design points.

  – **Using the Distribution Setting in Flight Conditions**

    The **Distribution** setting works similarly for all three categories of flight conditions, and is described below.

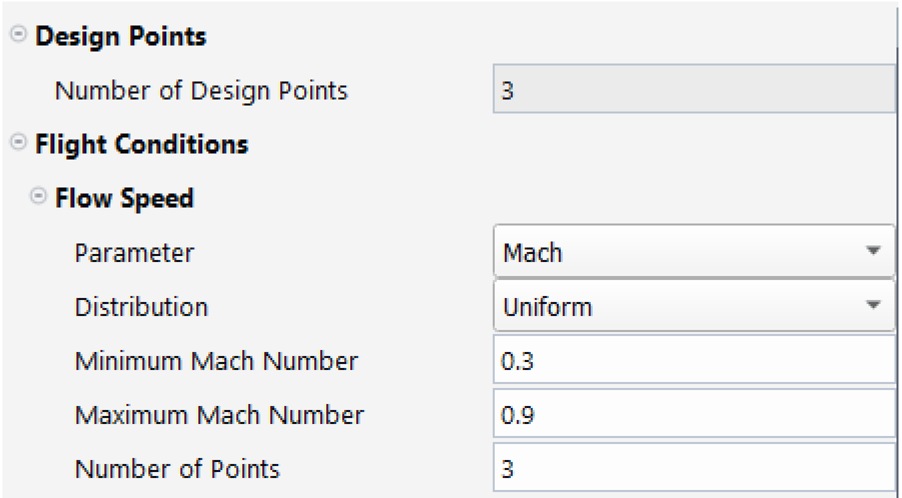
    → **Distribution: Constant**

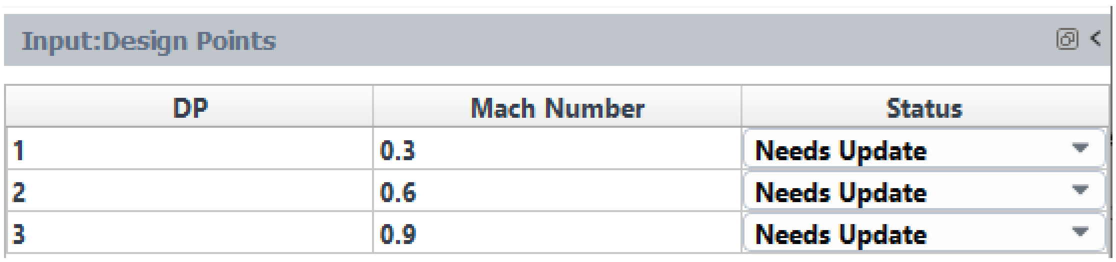
    If a **Distribution** is set to **Constant**, an input is revealed to enter the constant value of selected parameter to be used for all design points. This is shown in the image below using the **Flow Speed** → **Parameter** → **Mach** as an example.




    → **Distribution: Uniform (Single Parameter)**

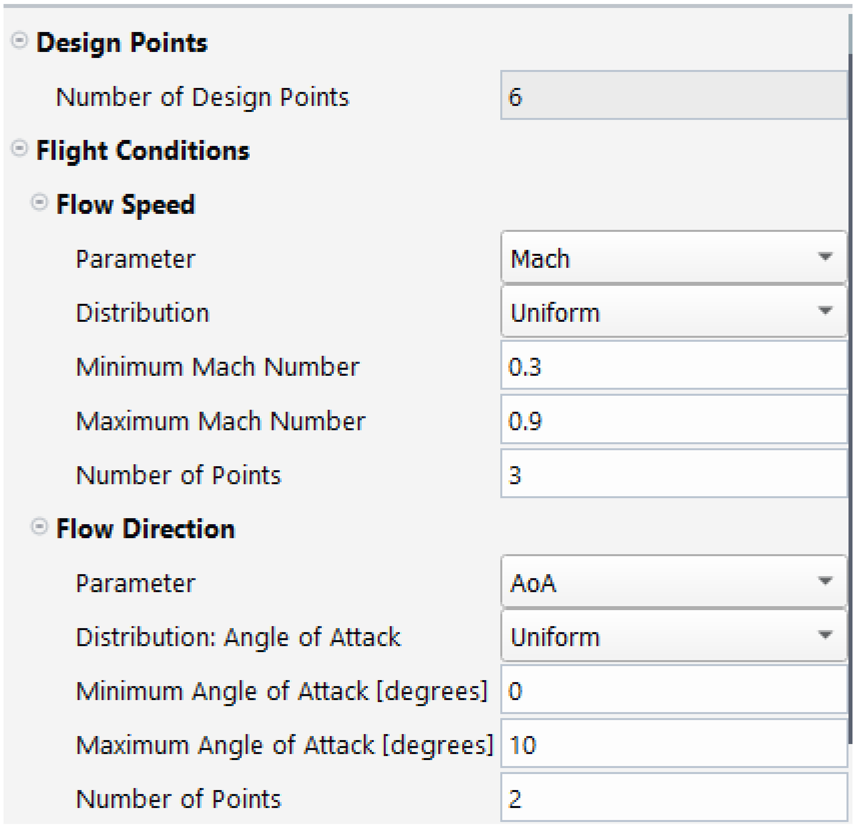
    If a **Distribution** is set to **Uniform**, inputs are revealed to enter the **Minimum Mach Number** and **Maximum Mach Number** value as well as the **Number of Points** to use for the selected parameter. These values will be used to fill the **Input: Design Point Table** that appears above the graphics window in the user interface. Furthermore, the **Number of Design Points** setting will be greyed out and automatically set to match the **Number of Points** set for the uniform distribution of the selected parameter.




→ **Distribution: Uniform (Multiple Parameters)**

If the **Distribution** of multiple input parameters are set to **Uniform**, the **Input: Design Point Table** will be generated to contain Design Points for every combination of inputs from each parameter that is set to use a uniform distribution. Furthermore, the **Number of Design Points** setting will be updated to match the number generated by using multiple uniform input parameters. In the image below, both **Mach** and **AoA** are set to use a **Distribution** → **Uniform**, and a total of six design points are generated.

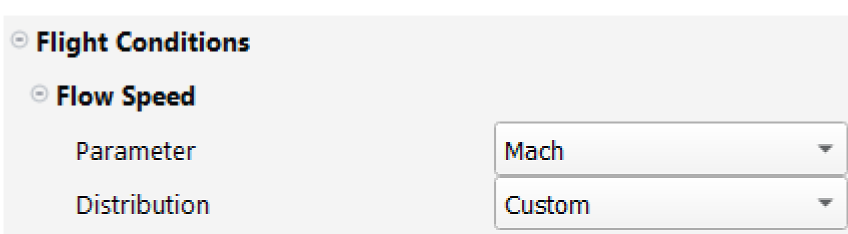


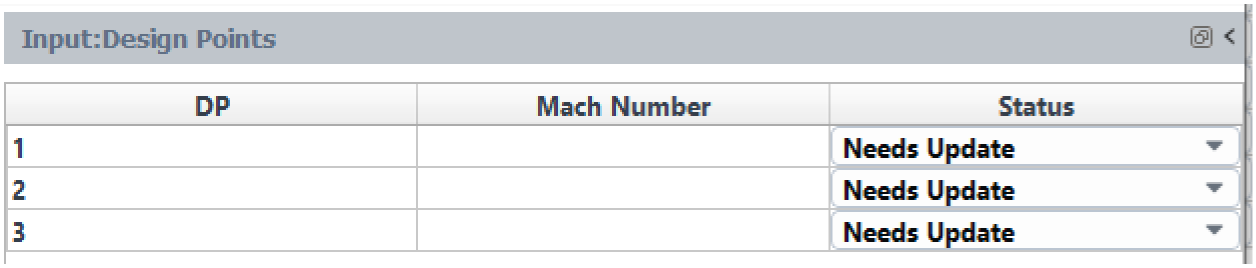

**Input:Design Points**

| DP | Mach Number | Angle of Attack [deg.] | Status |
|---|---|---|---|
| 1 | 0.3 | 0.0 | Needs Update |
| 2 | 0.3 | 10.0 | Needs Update |
| 3 | 0.6 | 0.0 | Needs Update |
| 4 | 0.6 | 10.0 | Needs Update |
| 5 | 0.9 | 0.0 | Needs Update |
| 6 | 0.9 | 10.0 | Needs Update |

→ **Distribution: Custom**

If a **Distribution** is set to **Custom**, empty cells are generated in the **Input: Design Points** table where the user can enter a custom value for each design point.






– **Flow Speed**

This section contains inputs to define the flow speed of the ambient flight conditions used to perform your aerodynamic simulations. Settings applied in this section will be applied to the **Inlet** boundary zones that are part of the **Freestream** or **WindTunnel** component groups.

→ **Parameter**

This setting defines the type of input parameter that will be used to define the flow speed of the condition. **Mach** number or **True Airspeed** (m/s) can be selected.

- **Mach**

  Mach number is used to define the speed condition for each Design Point.

- **True Airspeed**

  True Airspeed (m/s) is used to define the speed condition for each Design Point. If True Airspeed is used for the **Flow Speed Parameter**, the **Pressure and Temperature** category is limited to use either the **Static** or **Altitude Parameter** setting under **Flow Direction**.

→ **Distribution**

This setting defines how the value of the input parameter selected above is distributed across each Design Point. **Constant**, **Uniform** or **Custom** can be selected. Refer to Using the Distribution Setting in Flight Conditions under Simulation Conditions (p. 466) for a more complete description of how this setting is used.

→ **Parameter and Distribution Specific Input Values**

Depending on what is selected for **Parameter** and **Distribution**, further inputs to control the values to use for the **Flow Speed** input parameter will be revealed under **Flow Speed** or in the **Input Design Point Table**.

– **Flow Direction**

This section contains inputs to define the flow direction of each condition. Settings applied in this section will be applied to the **Inlet** boundary zones that are part of the **Freestream** or **WindTunnel** component groups.

→ **Parameter**

This setting defines the type of input parameter that will be used to define the flow direction. **AoA**, **AoS** and **AoA+AoS** can be selected. This setting is only available for a **Freestream Do-**

**main Type**. The **Flow Direction** is fixed to a **0 Angle of Attack [degrees]** when using a **WindTunnel Domain Type**.

- **AoA**

  Angle of attack (degrees) is used to define the flow direction for each Design Point. Increasing the angle of attack will increase the pitch of the aircraft with respect to the incoming flow.

- **AoS**

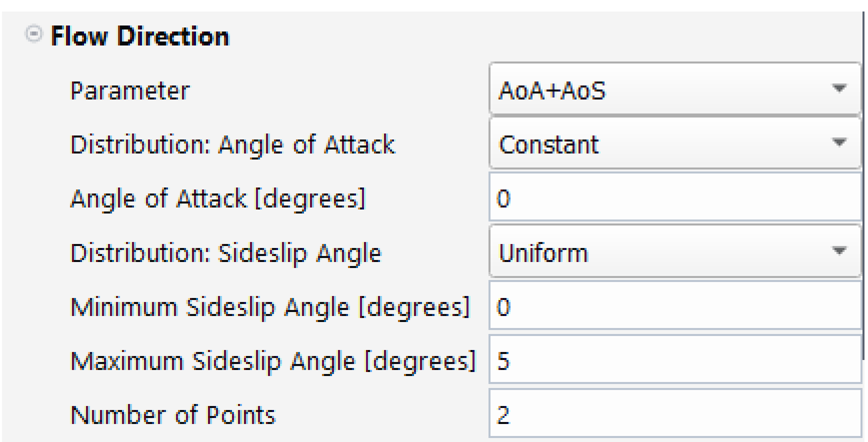
  Sideslip angle (in degrees) is used to define the flow direction for each Design Point. Increasing the sideslip angle will increase the yaw of the aircraft with respect to the incoming flow. If you would like to use a non-zero value for sideslip angle, they must ensure that they don't use a symmetry boundary to define their domain.

- **AoA+AoS**

  Both angle of attack and sideslip angle (in degrees) are used to define the flow direction for each Design Point. Increasing the angle of attack will increase the pitch of the aircraft with respect to the incoming flow. Increasing the sideslip angle will increase the yaw of the aircraft with respect to the incoming flow. If you would like to use a non-zero value for sideslip angle, they must ensure that they don't use a symmetry boundary to define their domain.

→ **Distribution**

This setting defines how the value of the input parameters selected above is distributed across each Design Point. **Constant**, **Uniform** or **Custom** can be selected. If the **Parameter** is set to **AoA+AoS**, a **Distribution** setting is revealed and must be defined for both inputs. For example, in the image below, a **Constant** distribution is used for angle of attack, and a **Uniform** distribution is used for the sideslip angle.




Refer to Using the Distribution Setting in Flight Conditions under Simulation Conditions (p. 466) for a more complete description of how this setting is used.

→ **Parameter and Distribution Specific Input Values**

Depending on what is selected for **Parameter** and **Distribution**, further inputs to control the values to use for the **Flow Speed** input parameter will be revealed under **Flow Speed** or in the **Input Design Point Table**.

– **Pressure and Temperature**

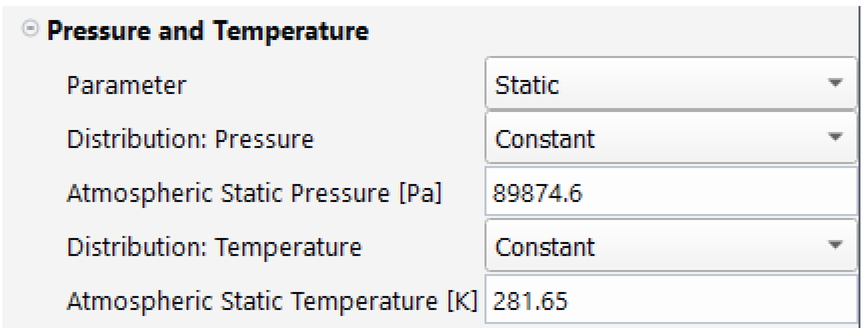
This section contains inputs to define the static pressure and temperature to perform your aerodynamic simulations. Settings applied in this section will be applied to the **Inlet** and **Outlet** boundary zones that are part of the **Freestream** or **WindTunnel** component groups.

→ **Parameter**

This setting defines the type of input parameters that will be used to define the static pressure and temperature at each design point. **Static**, **Total**, **Altitude** and **Reynolds** can be selected.

- **Static**

  **Static Pressure [Pa]** and **Static Temperature [K]** are directly used to define the condition at each Design Point.




- **Total**

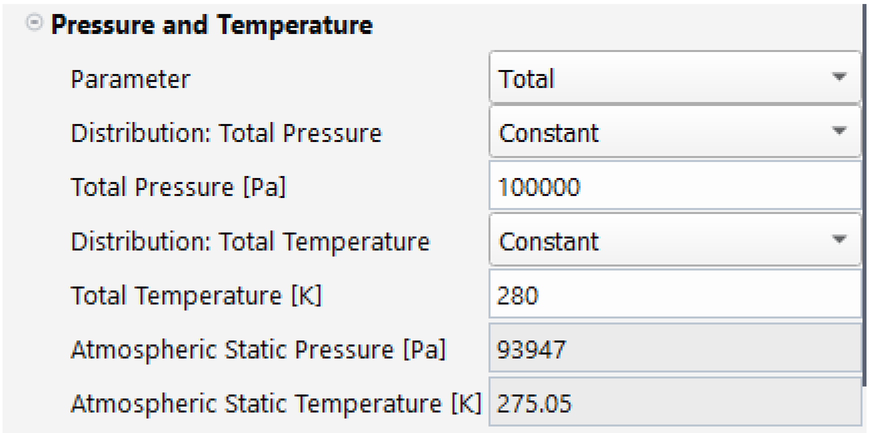
  **Total Pressure [Pa]** and **Total Temperature [K]** are used to define the static pressure and static temperature of the condition for each Design Point. The static temperature and static pressure will be computed from the total temperature, total pressure and mach number, using the following correlations.

$$\frac{p_t}{p} = \left(1+\frac{\gamma-1}{2}M^2\right)^{\frac{\gamma}{\gamma-1}} \qquad \frac{T_t}{T} = 1+\frac{\gamma-1}{2}M^2$$

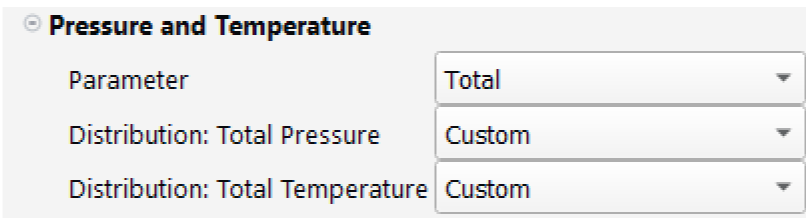
When **Total** is selected, **True Airspeed** cannot be selected for the **Flow Speed Parameter**. Only **Mach** number is available.

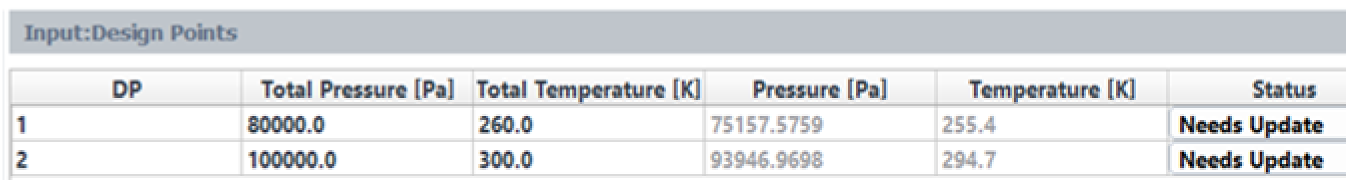
Depending on each parameter's **Distribution** setting, the value computed for the **Static Pressure** and **Static Temperature** will be displayed in the **Pressure and Temperature** section of the **Properties** panel or in the **Input Design Points Table**.

For example, in the image below, both distributions are set to **Constant**, and therefore the computed values of **Static Pressure** and **Static Temperature** are listed in grey in the **Pressure and Temperature** of the **Properties** panel.




In the next example, both distributions are set to **Custom**, and therefore the computed values of **Static Pressure** and **Static Temperature** are listed in grey inside the **Input Design Points Table**.



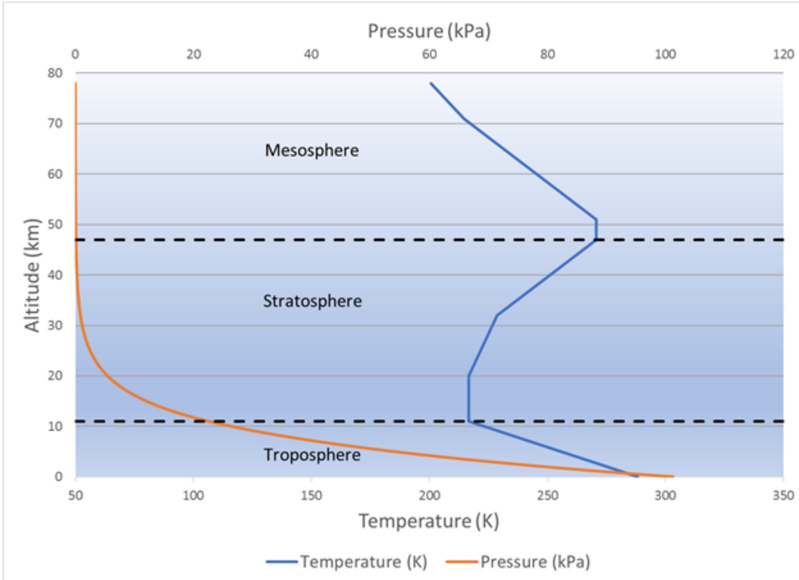
Input:Design Points

| DP | Total Pressure [Pa] | Total Temperature [K] | Pressure [Pa] | Temperature [K] | Status |
|---|---|---|---|---|---|
| 1 | 80000.0 | 260.0 | 75157.5759 | 255.4 | Needs Update |
| 2 | 100000.0 | 300.0 | 93946.9698 | 294.7 | Needs Update |

- **Altitude**

  Using the International Standard Atmosphere, Flight Altitude (in meters) is used to define the **Static Pressure** and **Static Temperature** of the ambient flight conditions for each Design Point. Depending on each parameter's **Distribution** setting, the value computed for the **Static Pressure** and **Static Temperature** will be displayed in the **Pressure and Temperature** section of the **Properties** panel or in the **Input Design Points Table**.

  Currently,this option is supported for the following atmospheric regions: Troposphere, Stratosphere and Mesosphere – up to a maximum of 71,000 meters.




- **Reynolds**

  Reynolds number is used to define the **Static Pressure** and **Static Temperature** of the ambient flight conditions for each Design Point. When **Reynolds** is selected, you must input values for Reynolds number and Static Temperature. The Static Pressure will then be computed from the Reynolds number, Static Temperature, Mach number and Reference Length (specified in Geometric Properties), using the following correlations, while using the Ideal Gas Law to compute density and Sutherland's Law to compute viscosity.

$$Re = \rho V L / \mu$$

$$\rho = P / RT \quad (ideal\ gas\ law)$$

$$\mu = \mu_{ref} \left( \frac{T}{T_{ref}} \right)^{\frac{3}{2}} \frac{T_{ref}+S}{T+S} \quad (Sutherland's\ Law)$$

$$R = 287.05$$

$$\mu_{ref} = 0.0000179$$

$$T_{ref} = 288$$

$$S = 110$$

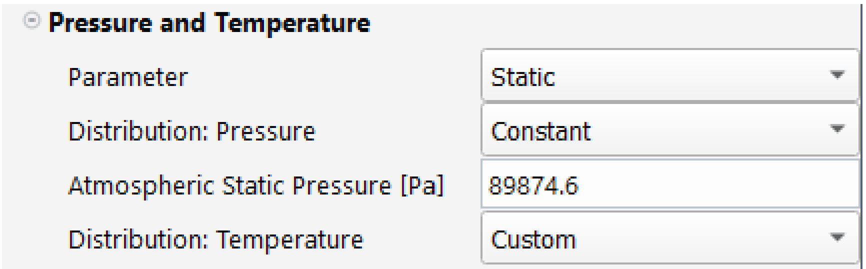
$$\gamma = 1.4$$

When **Reynolds** is selected, **True Airspeed** cannot be selected for the **Flow Speed Parameter**. Only **Mach** number is available.

Depending on each parameter's **Distribution** setting, the value computed for the Static Pressure and Static Temperature will be displayed in the **Pressure and Temperature** section of the **Properties** panel or in the **Input Design Points Table**.

– **Distribution**

This setting defines how the value of the input parameters selected above is distributed across each Design Point. **Constant**, **Uniform** or **Custom** can be selected. If the **Parameter** is set to **Static**, **Total** or **Reynolds**, a **Distribution** setting is revealed and must be defined for multiple inputs. For example, in the image below, the **Parameter** is set to **Static** and a **Constant** distribution is used for **Pressure**, and a **Custom** distribution is used for **Temperature**.




Refer to Using the Distribution Setting in Flight Conditions under Simulation Conditions (p. 466) for a more complete description of how this setting is used.

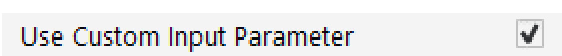
– **Parameter and Distribution Specific Input Values**

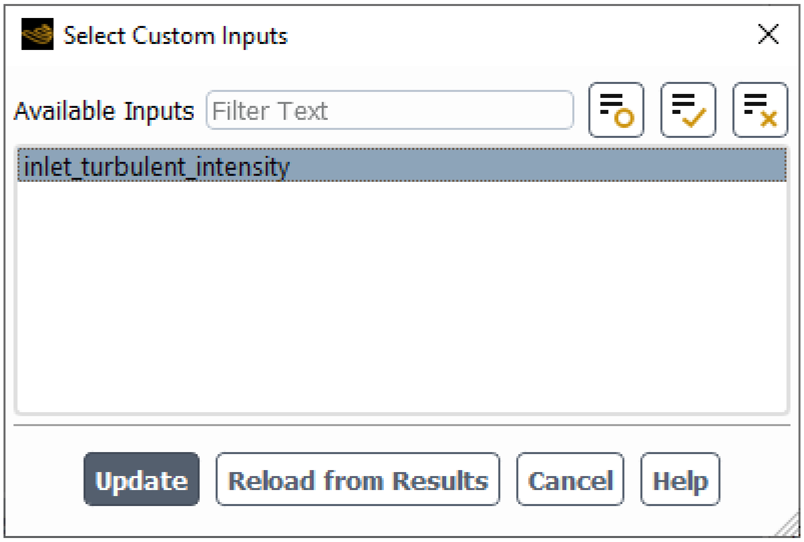
Depending on what is selected for **Parameter** and **Distribution**, further inputs to control the values to use for the **Flow Speed** input parameter will be revealed under **Flow Speed** or in the **Input Design Point Table**.

- **Custom Input and Output Parameters**

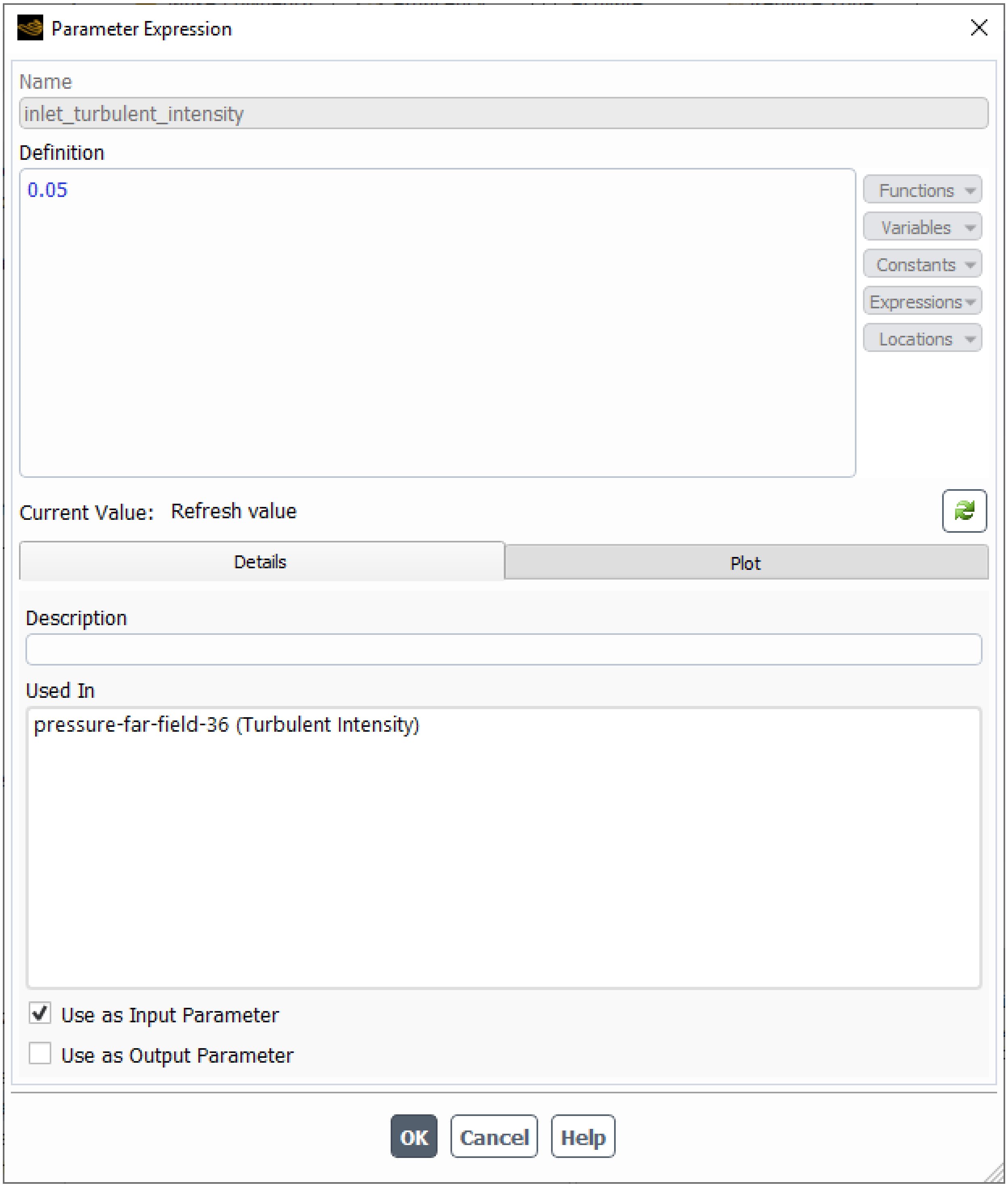
  – **Use Custom Input Parameter**

  This option allows the use of custom input parameters to vary at each design point. When this option is enabled, a window appears with a list of all custom inputs available. Select the Inputs you would like to use, and click **Update**. The selected input will appear as a fillable data column entry in the **Input:Design Points** table. If previous **Results** have already been created in your Simulation, clicking **Reload from Results** will reset the custom input selection to be consistent to what was used in the previous results.



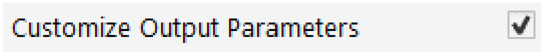


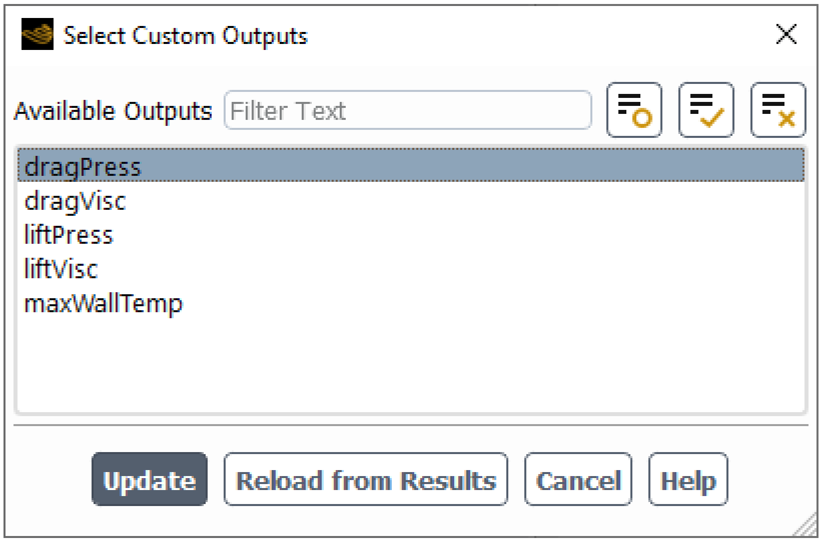

The list of custom input parameters will be filled by any Input Parameters defined in the initial case file using the Fluent Solution Workspace's Parameter Expression framework. If a user would like to add another custom input parameter to their current Fluent Aero session, they can use the **Workspaces** → **Solution** button to go to the **Solution Workspace** window and add new **Parameter Expression** that are set to **Use as Input Parameter**. The image below shows an example **Parameter Expression** defined in Fluent Solution workspace that works correctly as a custom input in Fluent Aero.




– **Customize Output Parameters**

This option allows the use of custom output parameters to vary at each design point. When this option is enabled, a window appears with a list of all custom outputs available. Select the outputs you would like to use, and click **Update**. When your **Results** are calculated, a Custom Output table will be produced listing the value of each selected custom output for each design point. If previous **Results** have already been created in your Simulation, clicking **Reload from Results** will reset the custom output selection to be consistent to what was used in the previous results.




The list of custom output parameters will be filled with both the custom outputs available in Fluent Aero by default, and by any Output Parameters defined in the initial case file using the Fluent Solution Workspace's Parameter Expression framework. The following default custom outputs are available in Fluent Aero:

→ **dragPress**

The pressure component of the total drag force.

→ **dragVisc**

The viscous component of the total drag force.

→ **liftPress**

The pressure component of the total lift force.

→ **liftVisc**

The viscous component of the total lift force.

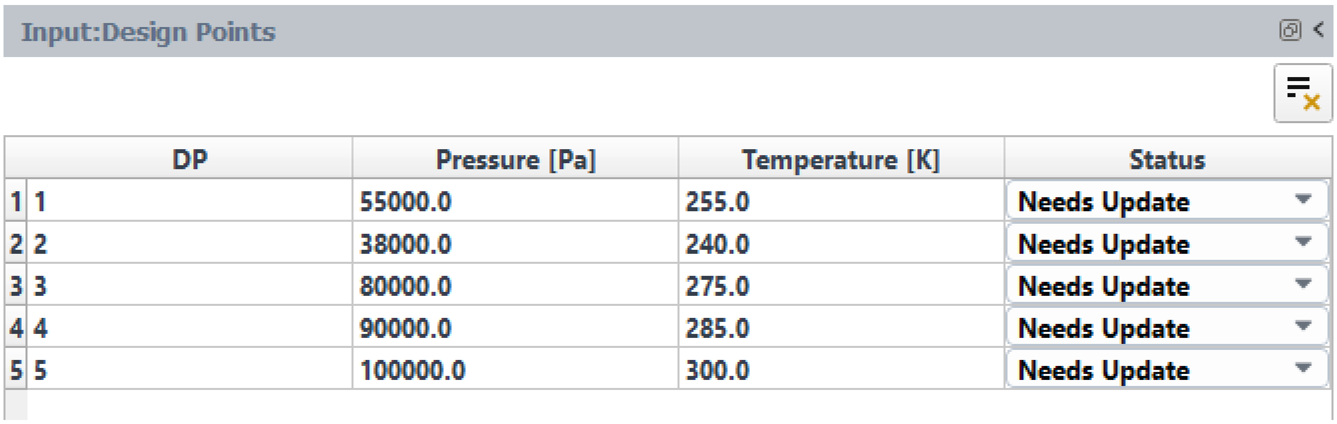
→ **maxWallTemp**

The maximum wall temperature that occurs on all walls in the domain.

If a user would like to add another custom output parameter to their current Fluent Aero session, they can use the **Show Solution Workspace** button to go to the Solution Workspace window and add new Parameter Expressions that are set to Use as Output Parameter.

- **Input:Design Points Table**

  When you modify the settings of the **Select Simulation Goals** step, an **Input: Design Points** table will be generated in the top-right area of the Fluent Aero user interface and modified to match the settings you have selected. This table will consist of a row for each Design Point you have requested to calculate, and a column for each input parameter that will be varied. In this table, the first column lists the Design Point number, the middle columns list the variable input parameters, and the last column list the **Status** of each Design Point calculation. The input variables in the middle columns can be edited by double clicking on a cell and entering a value. The Status of the final column can be changed by selecting an item from the drop down list. In the image below, there are 5 Design Points that feature a range of input **Pressure** and **Temperature** values and each **Design Point Status** is currently set to **Needs Update**, signifying that these Design Points have not yet been calculated.




| | DP | Pressure [Pa] | Temperature [K] | Status |
|---|---|---|---|---|
| 1 | 1 | 55000.0 | 255.0 | Needs Update |
| 2 | 2 | 38000.0 | 240.0 | Needs Update |
| 3 | 3 | 80000.0 | 275.0 | Needs Update |
| 4 | 4 | 90000.0 | 285.0 | Needs Update |
| 5 | 5 | 100000.0 | 300.0 | Needs Update |

– **Design Point Status**

The **Status** column of the **Input:Design Point** table is an important way of assessing and interacting with the calculation of each Design Point. This column both enables the user to check if a design point has been calculated successfully or not, and to request changes to the design point calculation. The following design point statuses are possible:

→ **Needs Update**

The **Status** will be set to **Needs Update** when either:

- The Design Point has not yet been calculated, or
- A setting in the **Geometric Properties**, **Simulation Conditions**, or **Component Groups** steps has changed since the last time the design point has been calculated, or
- The input values used in the **Input:Design Points** table have changed since the last time the design point has been calculated.

Notably, changes to the **Solve** step will not cause the **Status** to change to **Needs Update**. This is because a user may want to change the Solve settings midway through a design point calculation before continuing with the calculation.

Lastly, a user can manually set a **Design Point Status** to **Needs Update**, even if the above mentioned settings have not been changed.

Any **Design Point** whose **Status** is set to **Needs Update** will be calculated when a user clicks the **Calculate** command, and any previous solutions that may have existed for that Design Point will not be used.

→ **Updated**

The **Status** will be set to **Updated** when a design point has been calculated successfully, and a solution has been obtained.

Any **Design Point** whose **Status** is set to **Updated** will not be calculated when a user clicks on the **Calculate** command.

→ **Interrupted**

The **Status** will be set to **Interrupted** if you have used the **Interrupt** command during the calculation of a design point.

Any **Design Point** whose **Status** is set to **Interrupted** will not be calculated when a user clicks on the **Calculate** command.

---

**Note:**

When a Design Point calculation is interrupted, by default, the results files (`out.0*.dat[.h5]` and `out.0*.cas[.h5]`) are not saved, and the values in the results tables (**Table:Summary**, **Table:Coefficients**, **Table:Residuals**) are not updated. If you would like to save your current solution to the results files and update the results tables, the **Save Results** command should be used.

---

→ **Error**

The **Status** will be set to **Error** if an error occurs during the calculation of a design point that causes the calculation to stop.

Any **Design Point** whose **Status** is set to **Error** will not be calculated when a user clicks on the **Calculate** command.

---

**Note:**

If a user experienced an error during calculation, the **Console** output, or the `.trn` file located in the Simulation folder, can be investigated to find more information on the cause of the error.

---

→ **Do Not Update**

You can set the **Status** of a **Design Point** to **Do No Update** when the user does not want to calculate a given design point after clicking on the **Calculate** command.

→ **Continue to Update**

You can set the **Status** to **Continue to Update** if the user would like to continue the calculation of a previously updated design point. Any **Design Point** whose **Status** is set to **Continue to Update** will load and continue calculating from the previous solution file (`out.0*.dat[.h5]`) when the user clicks on the **Calculate** command.

→ **Initialize**

Set the **Status** of a **Design Point** to **Initialize** to only initialize the solution at a given design point after pressing **Calculate**.

Any **Design Point** whose **Status** is set to **Initialize** will be initialized and no solver iterations will be performed. This allows you to investigate the Initial solution of a design point before continuing on with the full calculation.

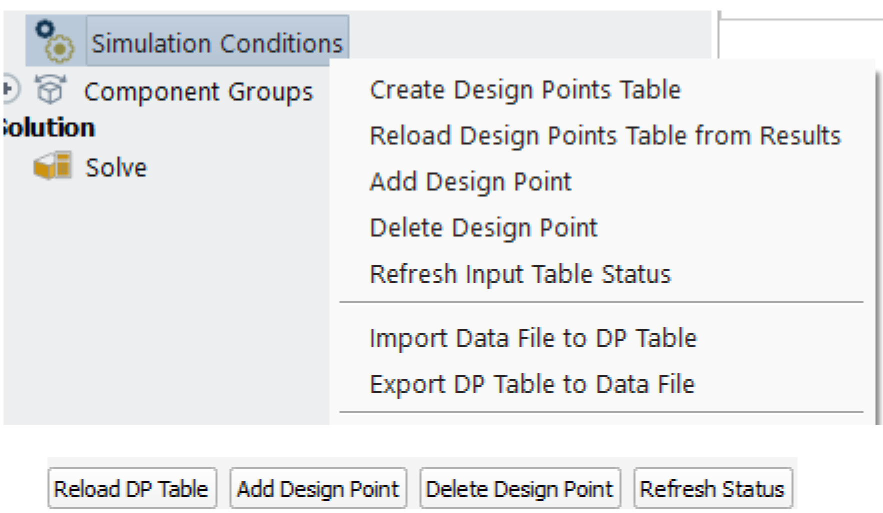
→ **Initialized**

The **Status** will be set to **Initialized** when a design point has been initialized successfully, and an initial solution has been obtained.

Any **Design Point** whose **Status** is set to **Initialized** will continue to be calculated from the current initial solution when you click **Calculate**.

- **Commands Available from Simulation Conditions**

  A number of commands are available from the **Select Simulation Goals** step. These can be accessed either from the buttons at the bottom of the **Properties** panel, or by right-clicking on **Select Simulation Goals** in the **Outline View** panel. Most of these commands are used to interact with the **Input Design Points** table.




  – **Create Design Points Table**

    This command can be used to generate the **Input: Design Points** table in the top-right area of the Fluent Aero user interface. Typically, this table is generated automatically, however, this option can be used if the table is not generated as expected.

  – **Reload DP Table** or **Reload Design Points Table from Results**

    If **Results** have already been computed for your Simulation, this command will reload the design point table values from the previously generated results, thereby removing any changes that may have been made by the user since.

  – **Add Design Point**

    This command will add a design point to the bottom of the **Input: Design Points** table.

– **Delete Design Point**

This command will delete a design point from the **Input:Design Points** table. A panel will appear, allowing you to enter the **Design Point Number** that you would like to delete. The design point will then be removed from the table.

If previous results have been calculated, all files associated with that design point and listed in the **DP-#** folder shown in the **Project View** will be deleted (including `.cas`, `.dat` and `.fconverg` files, as well as any `.csv` plot files that were generated). The numbering of all other design point folders and files will be readjusted such that they remain sequential. For example, if the previous **Results** included three Design Point Folders (**DP-1**, **DP-2** and **DP-3**), and **Design Point 2** is deleted, the previous **DP-2** folder and all its contents will be removed, and the previous **DP-3** folder will be renamed to **DP-2** to ensure that sequential numbering of folders and files is maintained.

– **Refresh Status** or **Refresh Input Table Status**

This command will refresh the **Status** of each **Design Point** to the value that was obtained at the end of your most recent calculation, or to **Needs Update** if no calculation has been performed, thereby removing any changes that may have been made by the user since.

– **Import Data File to DP Table**

This command can be used to fill the contents of your **Input:Design Points** table by importing a comma separated value (`.csv`) file. The first row of this file should contain a header line starting with a '#' comment symbol. The remaining rows should contain the Design Point number and all values you would like to fill in the table, separated by commas. An example file is shown in the image below.

```
# Design Point Table Contents
 1,20.65,67300,7.0319,225.01
 2,19.33,62900,13.4753,237.33
 3,17.71,58200,26.0318,250.49
 4,15.66,54600,42.1269,260.57
 5,13.29,52400,56.0235,266.73
```

– **Export DP Table to Data File**

This command can be used to export the contents of your **Input:Design Points** table to a comma separated values (`.csv`) file.

– **Show Output Selection Panel**

This command will cause the **Select Custom Outputs** panel to appear.

Once the goals have been specified, the user continues to the next step.

## 33.8.3. Component Groups

In this step, two important setup tasks are performed:

- The boundary zones of the geometry and mesh can be organized into groups of zones that represent entire aircraft components. This allows you to focus more on interacting with the aircraft geometry, rather than interacting with an unorganized list of boundary zones. It can help facilitate simulation setup and post processing. For example, an engine intake, exhaust and nacelle wall boundaries can be organized into an **Engine Component**.

- Any boundary zone or component specific boundary conditions can be applied. For example, the engine exhaust mass flow, which may change per design point, can be setup.

For simple geometries, where no organization of boundary zones is required, this step is not required and can be skipped.

---

**Note:**

It is highly recommended to ensure the organization of **Component Groups** and the selection of boundary zones names and types is completed before calculating any design point. If changes are made to any boundary zone type or boundary zone name after any **Results** have been calculated, this will be registered as a substantial change to the setup of the simulation, and the Status of all Design Points will be set to **Needs Update**. Furthermore, the run.settings file, which stores the component specific boundary condition setup, will be deleted. After this, a **Save Case** command or **Calculate** command must be performed so that an updated run.settings file is saved.

---

### 33.8.3.1. Organizing Component Groups and Boundaries

The first setup task that the **Component Groups** step is capable of is to organize the boundary zones of the geometry into groups of zones that represent aircraft components. Organizing boundary zones in this way will make interacting with the geometry and mesh more intuitive during setup and post processing.

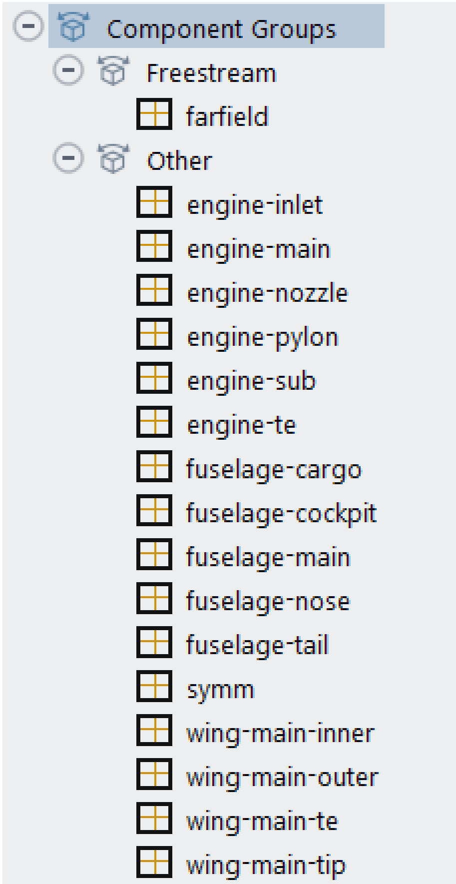
#### Default Component Groups

By default, there are at least two **Component Groups** present when you first creating a new simulation. They are:

- A **Freestream** or **WindTunnel** group that should contain all external boundaries of your computational mesh. This group is either automatically detected and setup when you first connect to your case file, or is modified and created manually. Refer to Freestream or WindTunnel Domain Type Requirements (p. 451) for more details.

- An **Other** group that should contain all other boundaries.

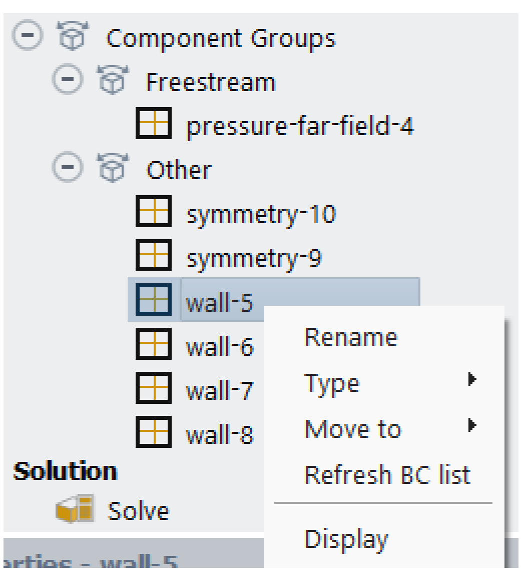
The image below shows an example of Freestream Domain Type setup where the **pressure-far-field** type zone called **farfield** is part of the **Freestream** group and all other walls are part of the **Other** group.




Further organization of these boundaries can be done by right-clicking a boundary zone and by using the **Component Manager** panel.

### Using the Boundary Zone Right-Click Commands

The following boundary zone organization commands are available by right-clicking a boundary in the **Component Groups** list.



- **Rename**

  Renames the selected boundary.

- **Type**

  Allows to change the boundary type of the selected boundary.

- **Move to**

  Allows to move the selected boundary to another **Component Group**.

- **Refresh BC list**

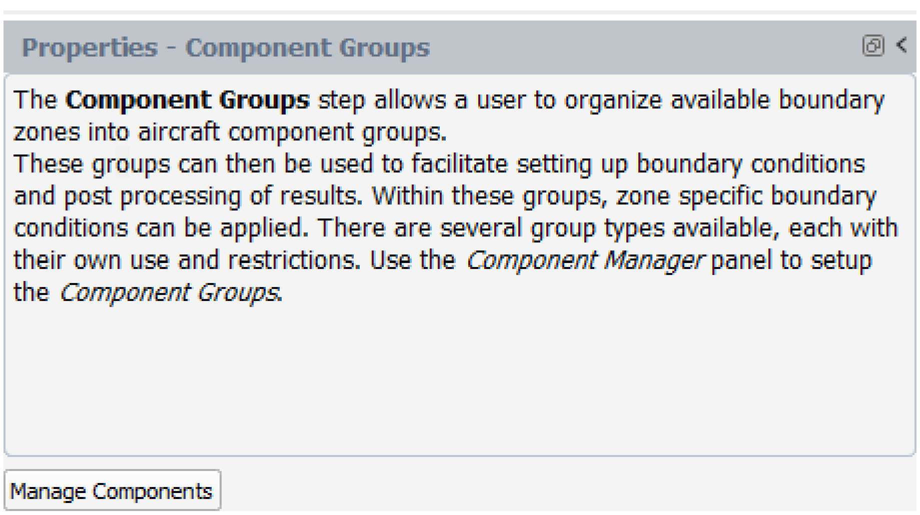
  This command will refresh the boundary list. This should update the Fluent Aero's boundary conditions list if any boundaries have been changed from inside the Fluent Solution Workspace.

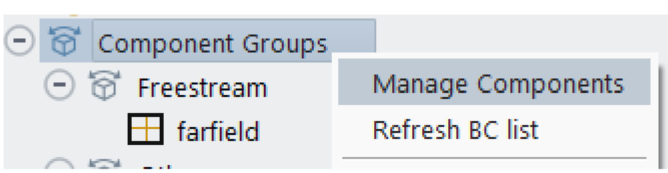
- **Display**

  Displays the selected boundary in the **Graphics** panel.

### Using the Component Manager

The **Component Manager** panel can be used to create groups of components and reorganize boundary zones into these groups. It can be accessed by clicking the **Manage Components** command button at the bottom of the **Properties – Components Groups** panel, or by right-clicking **Component Groups** and selecting **Manage Components**.

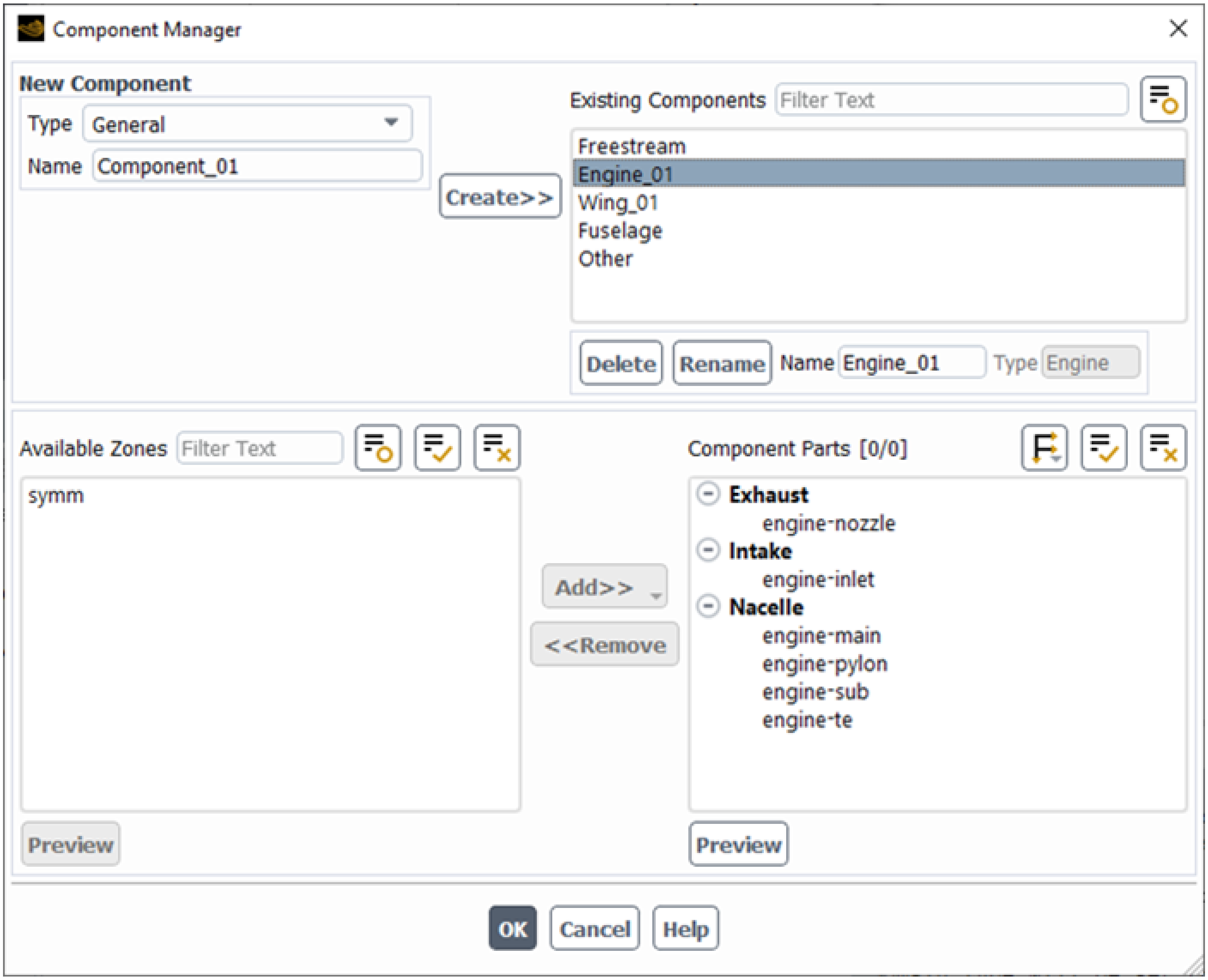


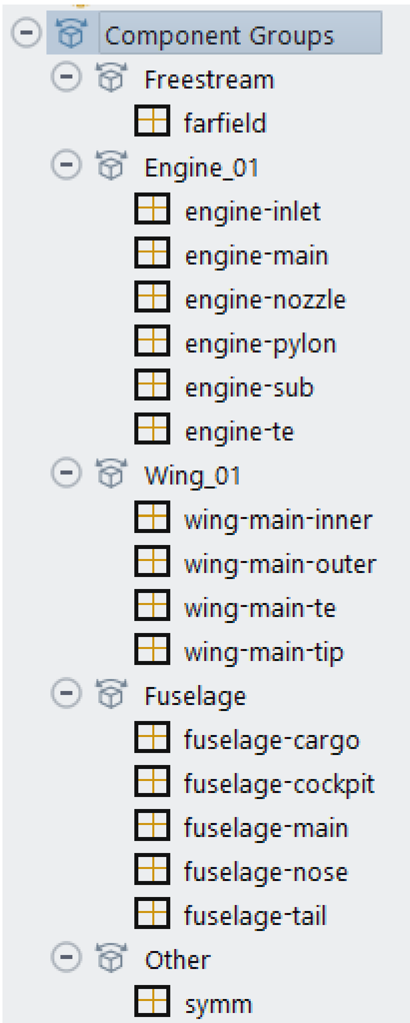

After selecting the command, a **Component Manager** panel will appear. There are four main areas of the **Component Manager**:

- **New Component** (top-left), which contains options to create a new component, including selecting the Type and setting the Name of the new component.
- **Existing Components** (top-right), which contains a selectable list of existing components, as well as commands to Delete or Rename existing components.
- **Available Zones** (bottom-left), which shows a selectable list of zones that can be added to a component using the **Add>>** button.
- **Component Parts** (bottom-right), which shows the list of zones that are contained within the currently selected Existing Component to. Depending on the type of component selected, this list of zones will be organized into specific Component Parts that only allow specific boundary types. Zones can be removed from a component group by selecting the zone in Component Parts and using the **Remove>>** button.

The image below shows the **Component Manager** panel after it has been used to setup **Engine**, **Fuselage** and **Wing** components in addition to the **Freestream** and **Other** components that are listed by default. A complete walkthrough of how to setup the components can be found in Introduction to Aircraft Component Groups and Computing Aerodynamic Coefficients on an Aircraft at Different Flight Altitudes and Engine Regimes (p. 611).




The image below shows the corresponding **Outline View** list of **Components**, after the **Component Manager** was used to organize the boundary zones.



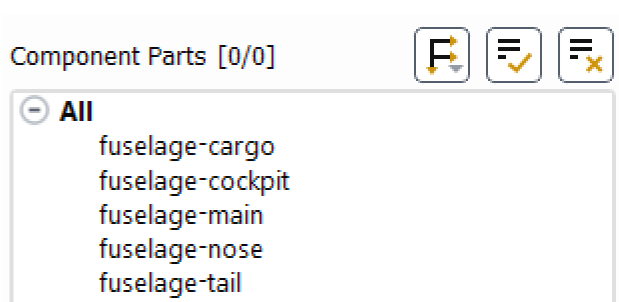

### Types of Components

Currently, there are three available **Types** of **Components**, **General**, **Wing** and **Engine**.

- **General**

  A **General** type component has no specific requirements. It features a single component part called **All**, to which any type of Fluent Aero supported boundary zone can be added.

  The **Component Parts** of the **General** type component called **fuselage** from the example above is shown in the image below:

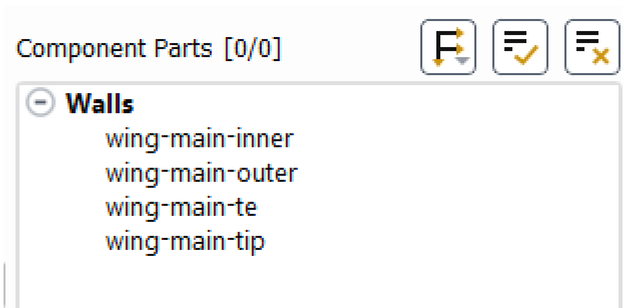


- **Wing**

  A **Wing** type component contains the following **Component Parts** with the following requirements:

  – **Walls**: this component part must contain **wall** type boundary zones.

  The **Component Parts** of the **Wing** type component called **wing-01** from the example above is shown in the image below:




- **Engine**

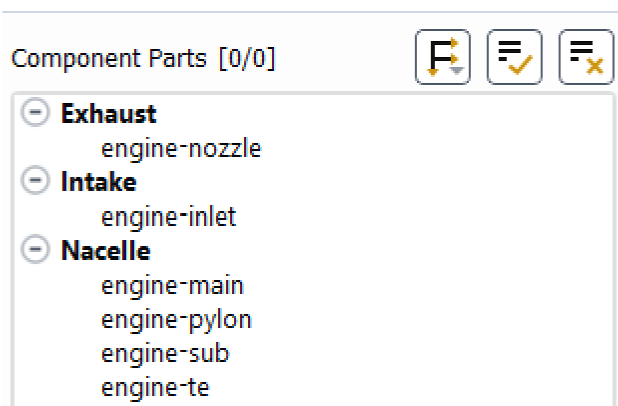
  An **Engine** type component contains the following **Component Parts** with the following requirements:

  – **Exhaust**: this component part must contain a **mass-flow-inlet** type boundary zone.

  – **Intake**: this component part must contain a **pressure-outlet** type boundary zone.

  – **Nacelle**: this component part is not required, but may only have **wall**-type boundary zones.

  The **Component Parts** of the **Engine** type component called **engine-01** from the example above is shown in the image below:




### 33.8.3.2. Applying Component Zone Specific Boundary Conditions

The second setup task that the **Component Groups** step is capable of is to apply specific boundary conditions to component zones.

While the **Simulation Conditions** step is used to apply **Freestream** or **WindTunnel Inlet** and **Outlet** conditions that represent flight or wind tunnel conditions of the aircraft, the zone specific boundary conditions of a **Component Groups** allows you to apply boundary conditions onto aircraft components. These boundary conditions may be constant or varying (custom) for each Design Point simulation. The mass flow rate of the engine exhaust is an example of a type of boundary condition that will be applied in the **Component Groups** step.

The zone specific boundary conditions setup options provided in the **Component Groups** step is only available on Fluent Aero's supported boundary types and only for certain supported boundary variables. To setup zone specific boundary conditions on unsupported boundaries, the **Use Custom Input Parameter** functionality provided by the **Simulation Conditions** step can be used.

### 33.8.3.3. Boundary Zone Types Supported by Fluent Aero

Zones using a type that is directly supported from inside Fluent Aero's user interface will appear with a black square with gold cross icon next to its zone name inside.


The following boundary zone types are supported by Fluent Aero.

- **velocity-inlet**
- **pressure-inlet**
- **mass-flow-inlet**
- **wall**
- **pressure-outlet**
- **symmetry**
- **periodic**

For some supported zone type, Fluent Aero may display some common boundary variables that can be controlled directly from inside Fluent Aero.

---

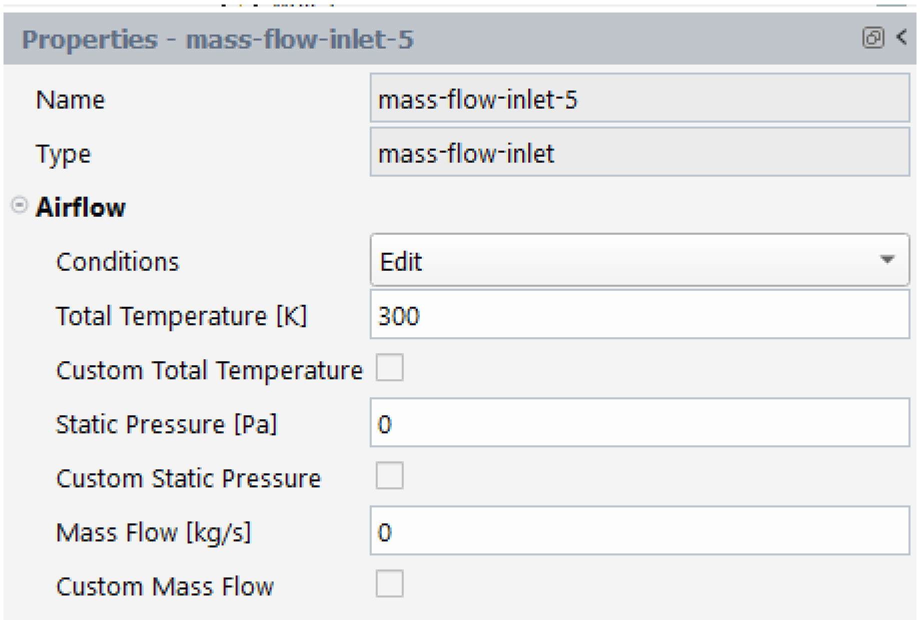
**Note:**

To control any variable that is not displayed directly from inside Fluent Aero, use the **Workspaces** → **Solution** button to show the Fluent Solution Workspace and modify the setting there, and/or use the **Simulation Conditions** → **Use Custom Input Parameter** functionality to modify the variable per design point.

---

The following supported zone type have the following settings available from inside Fluent Aero.

- **Mass-flow-inlet**




– **Conditions**

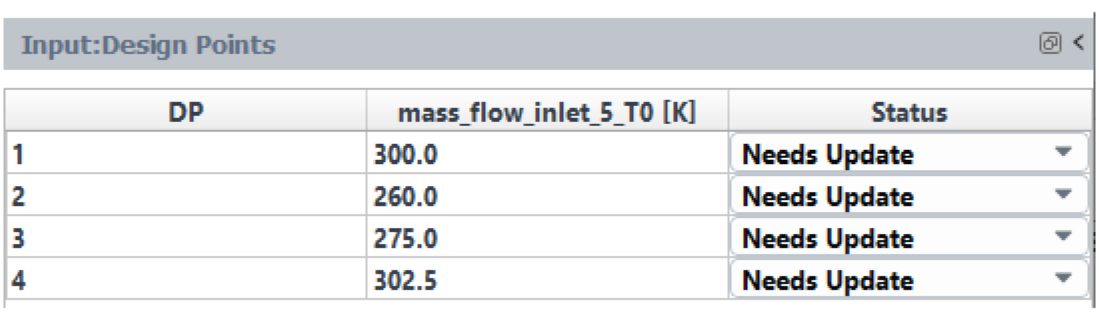
  Setting **Conditions** to **Edit** will cause Fluent Aero to reveal the list of settings available to specify directly inside the properties panel. Setting **Conditions** to **Case settings** will cause Fluent Aero to use the zone settings that were setup in the initial case file or from within the Fluent Solution Workspace.

– **Total Temperature [K]**

  Sets a constant total temperature in Kelvin to use on this boundary condition for all Design Points.

– **Custom Total Temperature**

  Enabling this option will insert an extra column in the **Input: Design Points** table where a custom total temperature can be specified for each Design Point, as shown below.


Input:Design Points

| DP | mass_flow_inlet_5_T0 [K] | Status |
|---|---|---|
| 1 | 300.0 | Needs Update |
| 2 | 260.0 | Needs Update |
| 3 | 275.0 | Needs Update |
| 4 | 302.5 | Needs Update |

– **Total Temperature Expression Name**

  If **Custom Total Temperature** is enabled, this option is revealed. In this case, you can set the name of the custom variable that is used to control the total temperature of the mass flow inlet zone. The default name of the expression uses the following format: **zone_name_T0**.

- **Static Pressure [Pa]**

  Sets a constant pressure in Pascal to use on this boundary condition for all Design Points.

- **Custom Static Pressure**

  Enabling this option will insert an extra column in the **Input: Design Points** table where a custom static pressure can be specified for each Design Point.

- **Static Pressure Expression Name**

  If **Custom Static Pressure** is enabled, this option is revealed. In this case, you can set the name of the custom variable that is used to control the static pressure of the mass flow inlet zone. The default name of the expression uses the following format: **zone_name_P**.

- **Mass Flow [kg/s]**

  Sets a constant mass flow in kilograms per second to use on this boundary condition for all Design Points.

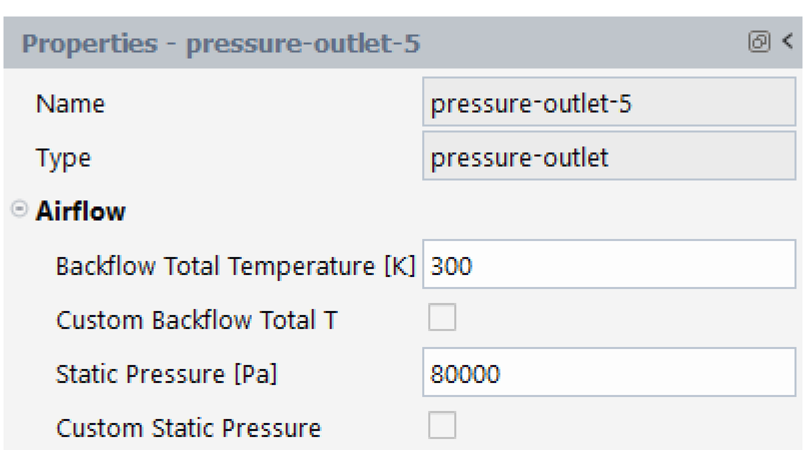
- **Custom Mass Flow**

  Enabling this option will insert an extra column in the **Input: Design Points** table where a custom mass flow can be specified for each Design Point.

- **Mass Flow Expression Name**

  If **Custom Mass Flow** is enabled, this option is revealed. In this case, you can set the name of the custom variable that is used to control the mass flow of the mass flow inlet zone. The default name of the expression uses the following format: **zone_name_massflow**.

• **Pressure-outlet**




- **Backflow Total Temperature [K]**

  Sets a constant backflow total temperature in Kelvin to use on this boundary condition for all Design Points.

- **Custom Backflow Total Temperature**

Enabling this option will insert an extra column in the **Input: Design Points** table where a custom backflow total temperature can be specified for each Design Point, as shown in the example image below.

**Input:Design Points**

| DP | pressure_outlet_5_T0 [K] | Status |
|---|---|---|
| 1 | 300.0 | Needs Update |
| 2 | 260.0 | Needs Update |
| 3 | 275.0 | Needs Update |
| 4 | 302.5 | Needs Update |

– **Backflow Total Temperature Expression Name**

If **Custom Backflow Total Temperature** is enabled, this option is revealed. In this case, you can set the name of the custom variable that is used to control the total temperature of the **pressure-outlet** zone. The default name of the expression uses the following format: **zone_name_T0**.

– **Static Pressure [Pa]**

Sets a constant pressure in Pascal to use on this boundary condition for all Design Points.

– **Custom Static Pressure**

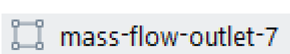
Enabling this option will insert an extra column in the **Input: Design Points** table where a custom static pressure can be specified for each Design Point.

– **Static Pressure Expression Name**

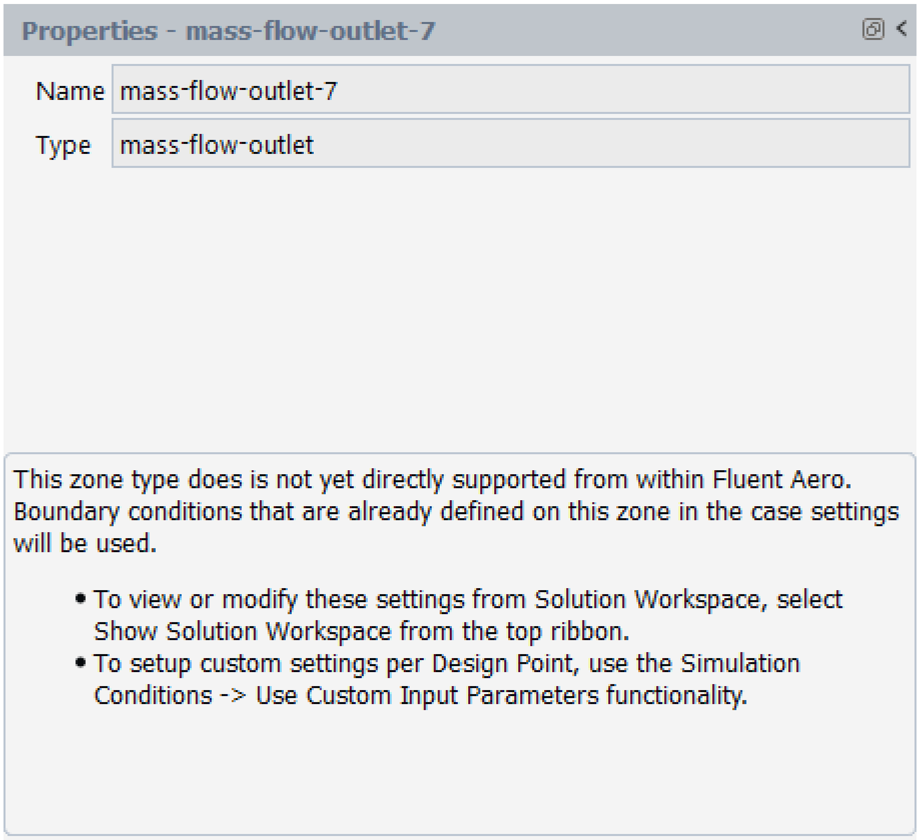
If **Custom Static Pressure** is enabled, this option is revealed. In this case, you can set the name of the custom variable that is used to control the static pressure of the **pressure-outlet** zone. The default name of the expression uses the following format: **zone_name_P**.

### 33.8.3.4. Boundary Zone Types Not Directly Supported by Fluent Aero

Zones using a type that is not directly supported inside Fluent Aero's UI will appear with a grey square icon next to its zone name inside.



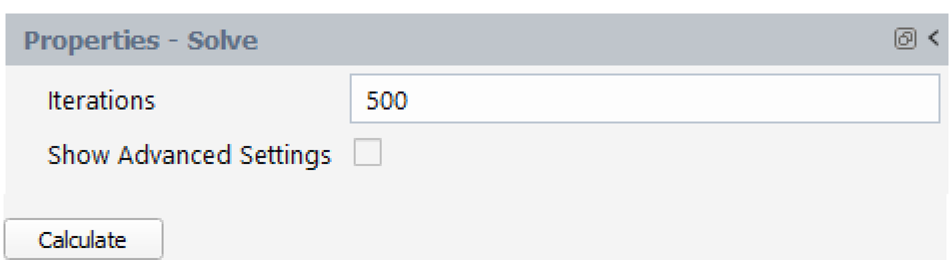

Also, a note will appear in the **Properties** panel specifying that the zone is not yet supported, along with recommendations on how to modify the input values on this boundary zone type, if needed.




For any zone type that is not directly supported by Fluent Aero, the boundary conditions that are already defined on this zone in the case settings will be used. If you would like to modify these settings, they can do so by selecting the **Workspaces** → **Solution** button from the top ribbon. Furthermore, to setup custom settings per design point, the **Simulation Conditions** → **Use Custom Input Parameters** functionality can be used.

### 33.8.4. Solve

In this step, you instruct Fluent Aero to calculate the design points specified in the previous steps. Clicking on **Solve** will reveal the **Solve** properties window.

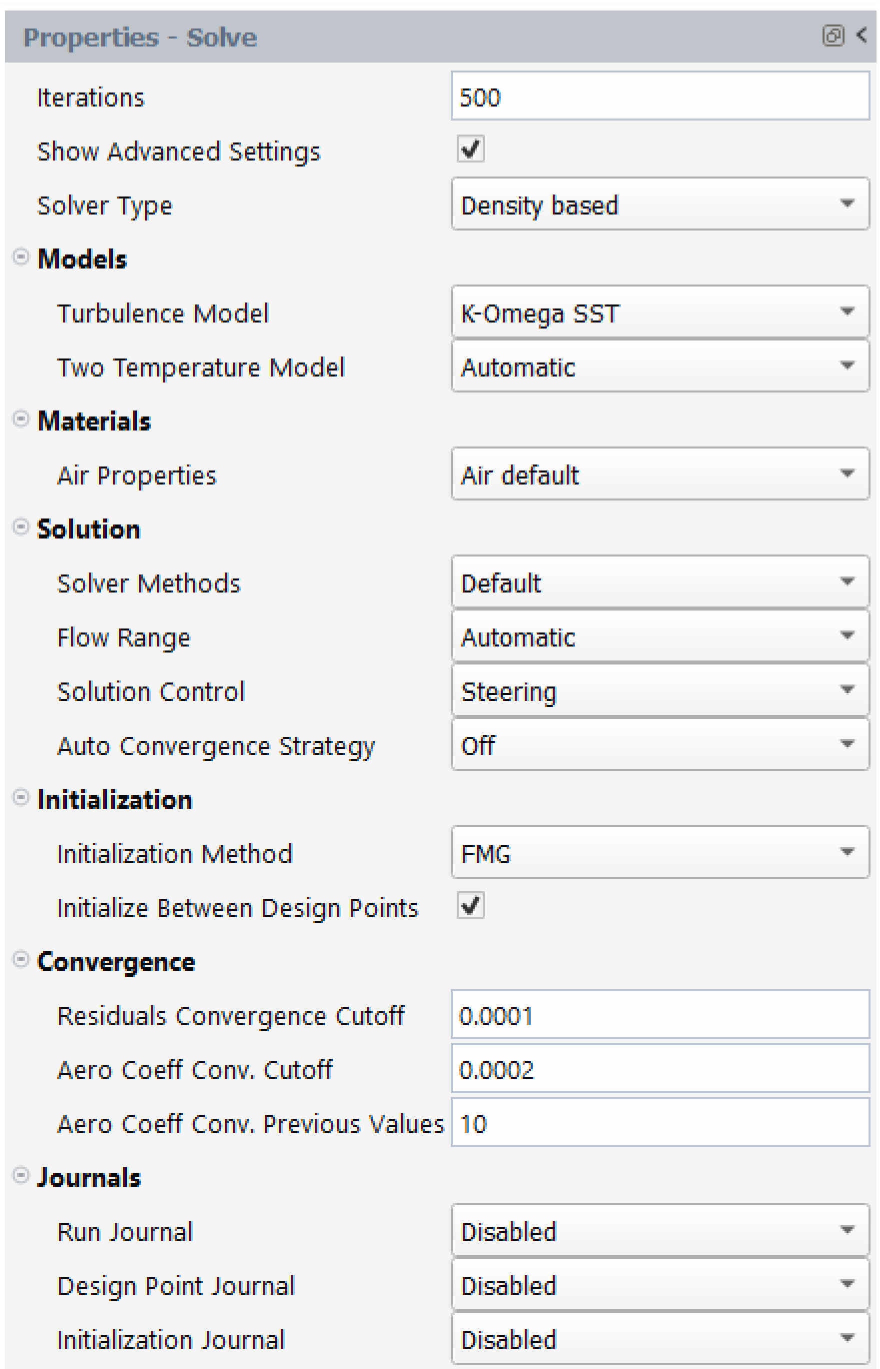



- **Iterations**

  This setting defines the maximum number of solver iterations that will be performed for each design point. The default number of iterations is 500. If the calculation of a single design point reaches the maximum solver iterations and the convergence criteria are satisfied, the calculation will stop, and the next design point will be automatically simulated.

- **Show Advanced Settings**

  **Show Advanced Settings** can be enabled to have access to and change certain CFD Solver settings. This is not mandatory, as Fluent Aero will otherwise calculate with its default parameters. By enabling **Show Advanced Settings**, the following options will be revealed.




- **Solver Type**

  This setting defines the solver settings used to perform the calculations. **Density based** is set by default. **Pressure based** can also be selected. The selection here may affect which other options listed below appear. For more information regarding these options, consult Overview of Flow Solvers.

- **Models**

  – **Turbulence Model**

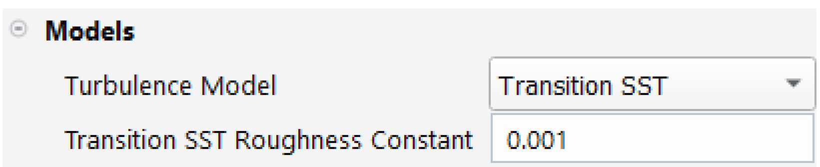
  This setting defines the **Turbulence Model**.

  → **K-Omega SST**

  This is the default setting. By selecting this default model, the viscous heating, and production limiter will be activated in the background Solver session.

  → **Transition SST**

  When selected, viscous heating and **Transition SST** Roughness Constant correlation are applied by default. The model constant of the latter will appear as another input option in the **Solve** panel. This value will have a significant effect on the laminar to turbulent transition location.




  → **K-Omega WJ-BSL-EARSM (Beta)**

  This is a more advanced turbulence model that is available as a beta option. This model is a 2 equations hybrid RSM that allows the capture of secondary flow motion driven by turbulence anisotropies or to better represent flows with strong swirl or streamline curvature. This requires prior enabling of the beta options in the Fluent Solution Workspace. By selecting this default model, the viscous heating, and production limiter will be activated in the background.

  → **Case Settings**

  When selected, Fluent Aero will not apply settings to the turbulence model. Instead, it will use the turbulence model setting setup in the initial case file, or from the Solution Workspace window (in **Models** → **Viscous**). This option allows an advanced user to have full access to the list of turbulence models inside the Fluent Solution Workspace. To make advanced changes to the turbulence model while the case file is already loaded, use the **Show Solution Workspace** command, and in the window that appears, make the appropriate changes in the **Viscous** panel.

  – **Two Temperature Model**

  This setting controls the **Two Temperature Model** in your Fluent Aero simulations. This model is only available if **Solver Type** is set to **Density based**. For more information regarding this model, consult The Two-Temperature Model for Hypersonic Flows.

  → **Automatic**

When selected, Fluent Aero uses the **Flow Range** for each Design Point to determine if the **Two Temperature Model** is enabled or disabled. If the **Mach** number for the design point is greater than or equal to 4.5, Fluent Aero will register that the **Flow Range** is hypersonic, and will activate the **Two Temperature Model**. If the **Mach** number for the design point is less than 4.5, Fluent Aero will register that the **Flow Range** is not hypersonic, and will disable the **Two Temperature Model**.

→ **Enabled**

When selected, Fluent Aero will ensure that the **Two Temperature Model** is enabled for all Design Point calculations.

→ **Disabled**

When selected, Fluent Aero will ensure that the **Two Temperature Model** is disabled for all Design Point calculations.

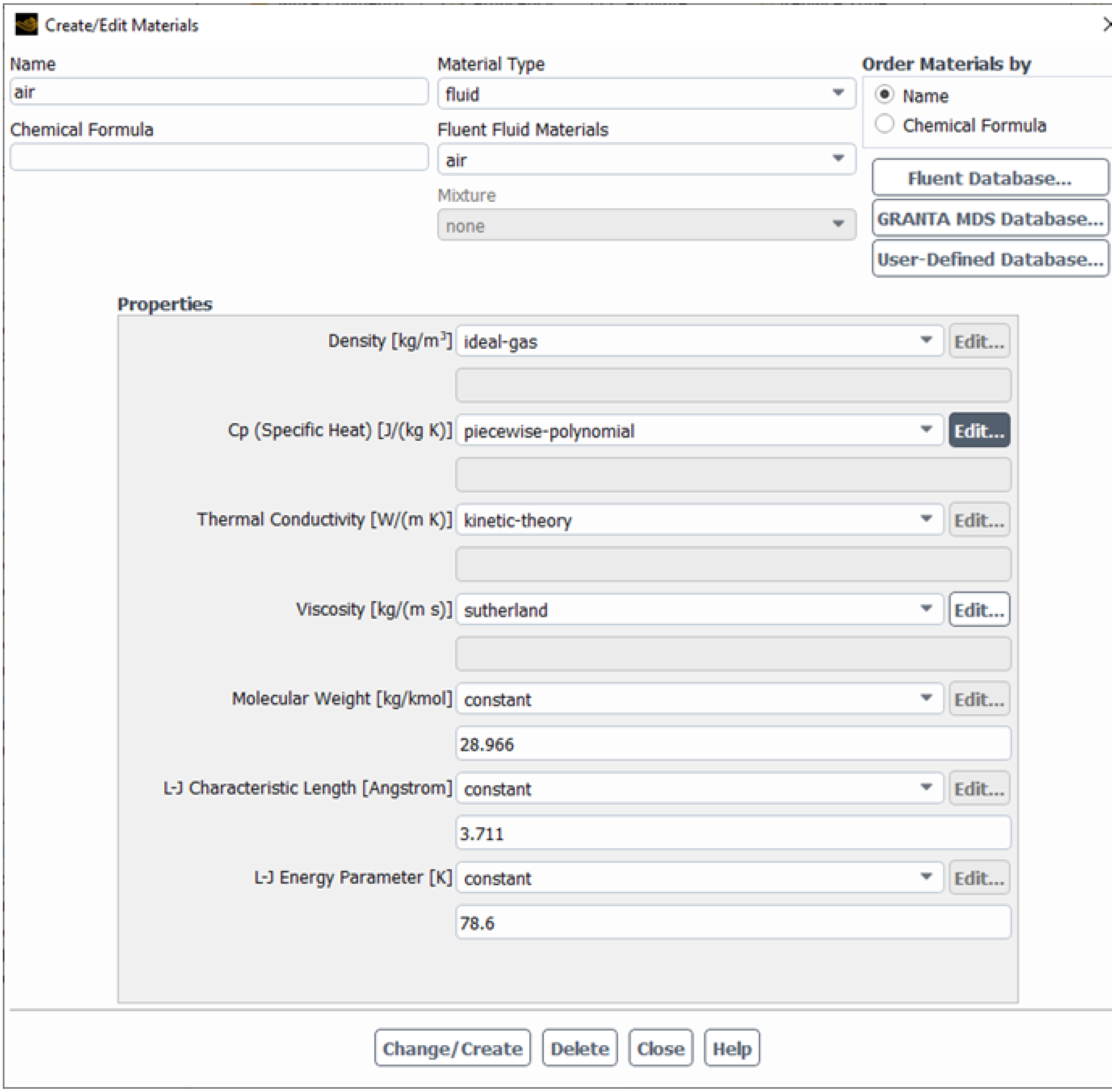
→ **Case Settings**

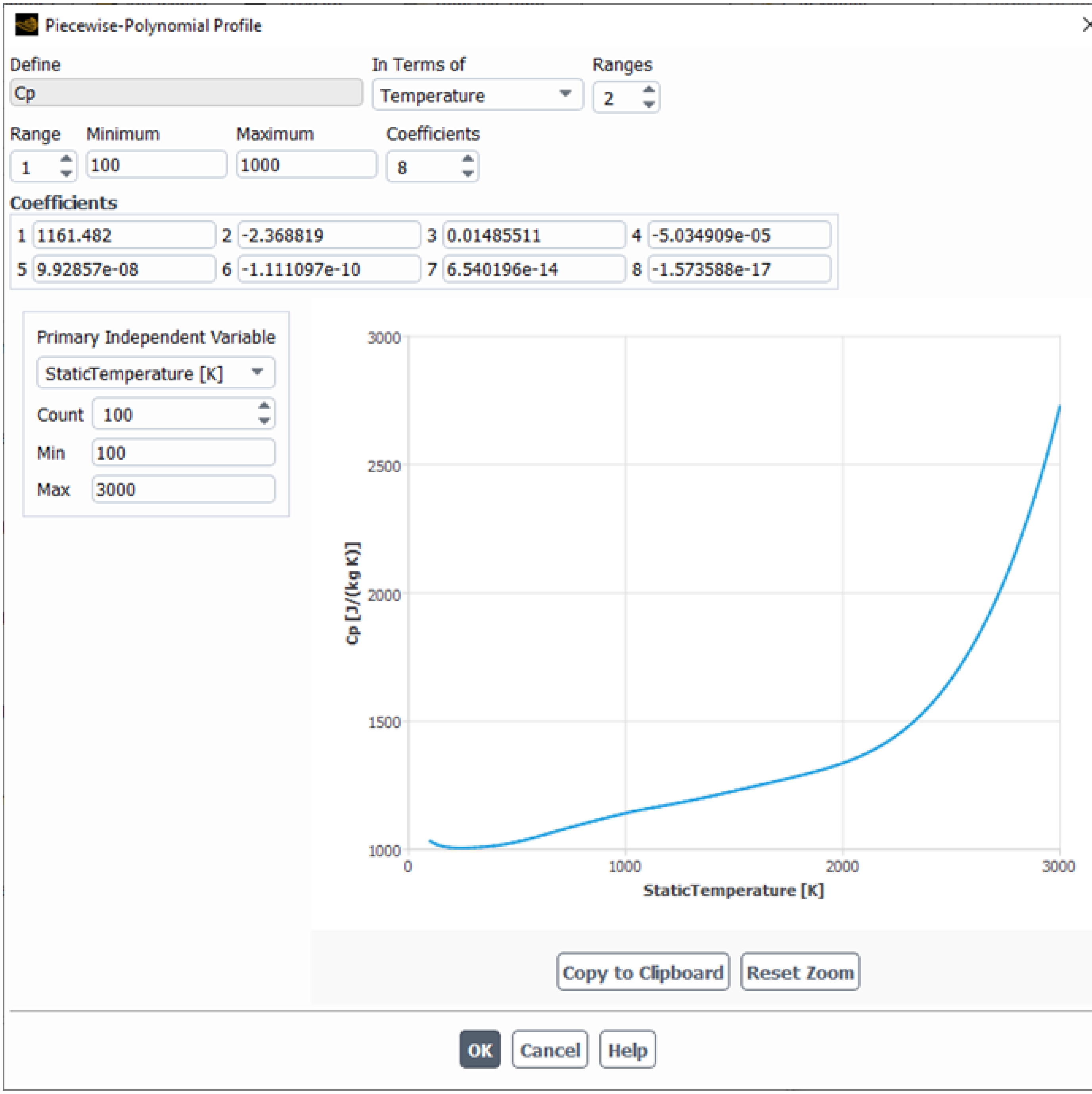
When selected, Fluent Aero will not apply settings to the **Two Temperature Model**. Instead, it will use the **Two Temperature Model** setting setup in the initial case file, or from the **Solution Workspace** window (in **Models** → **General**). If the **Two Temperature Model** is set to **Case settings**, the **Air Properties** must also be set to **Case settings**.

- **Materials**
  - **Air Properties**

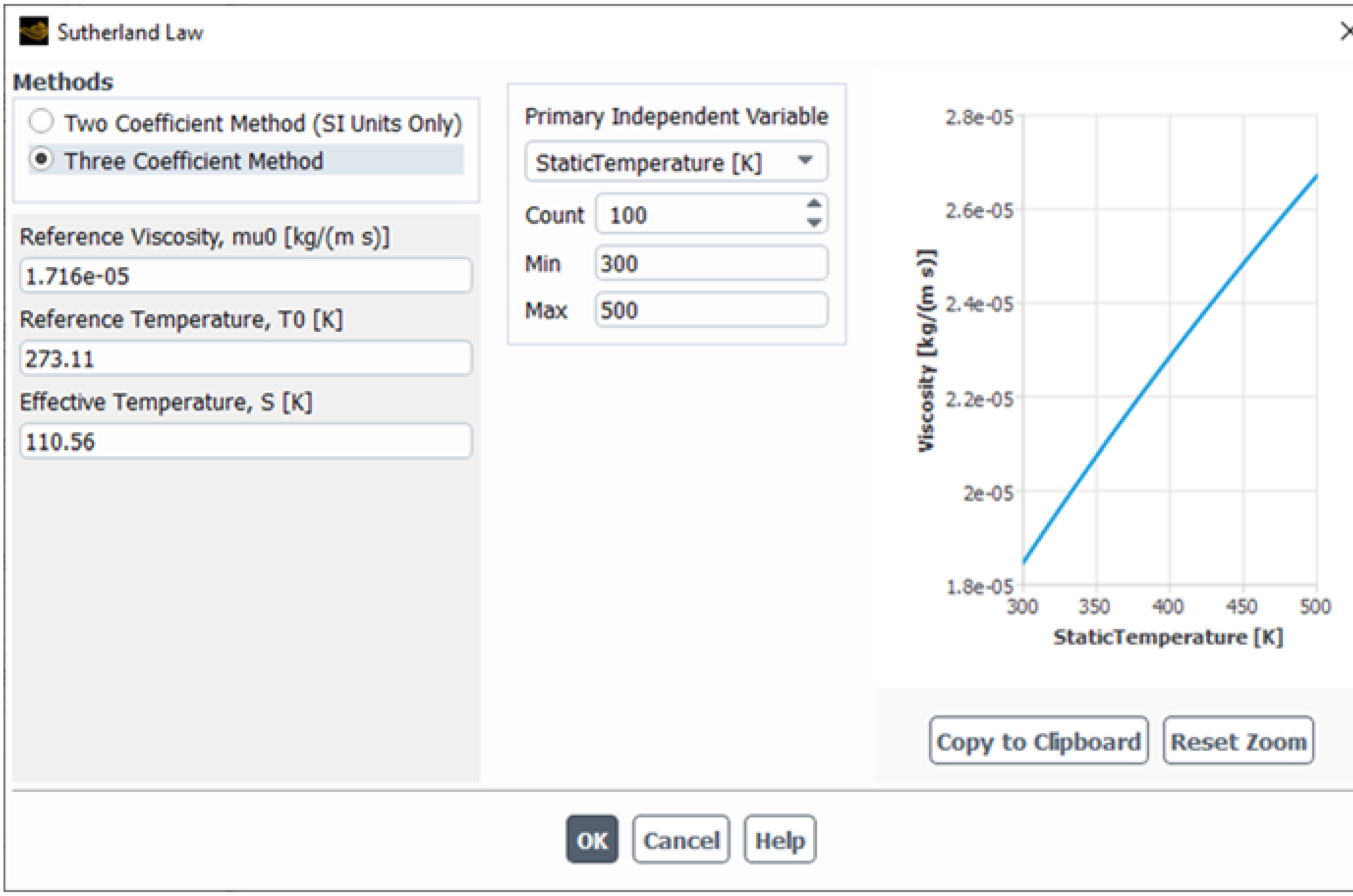
→ **Air default**

When selected, Fluent Aero will set two types of air material properties depending on the activation of the **Two Temperature Model**. Fluent Aero's typical default air properties, when the **Two Temperature Model** is disabled, are shown in the image below:

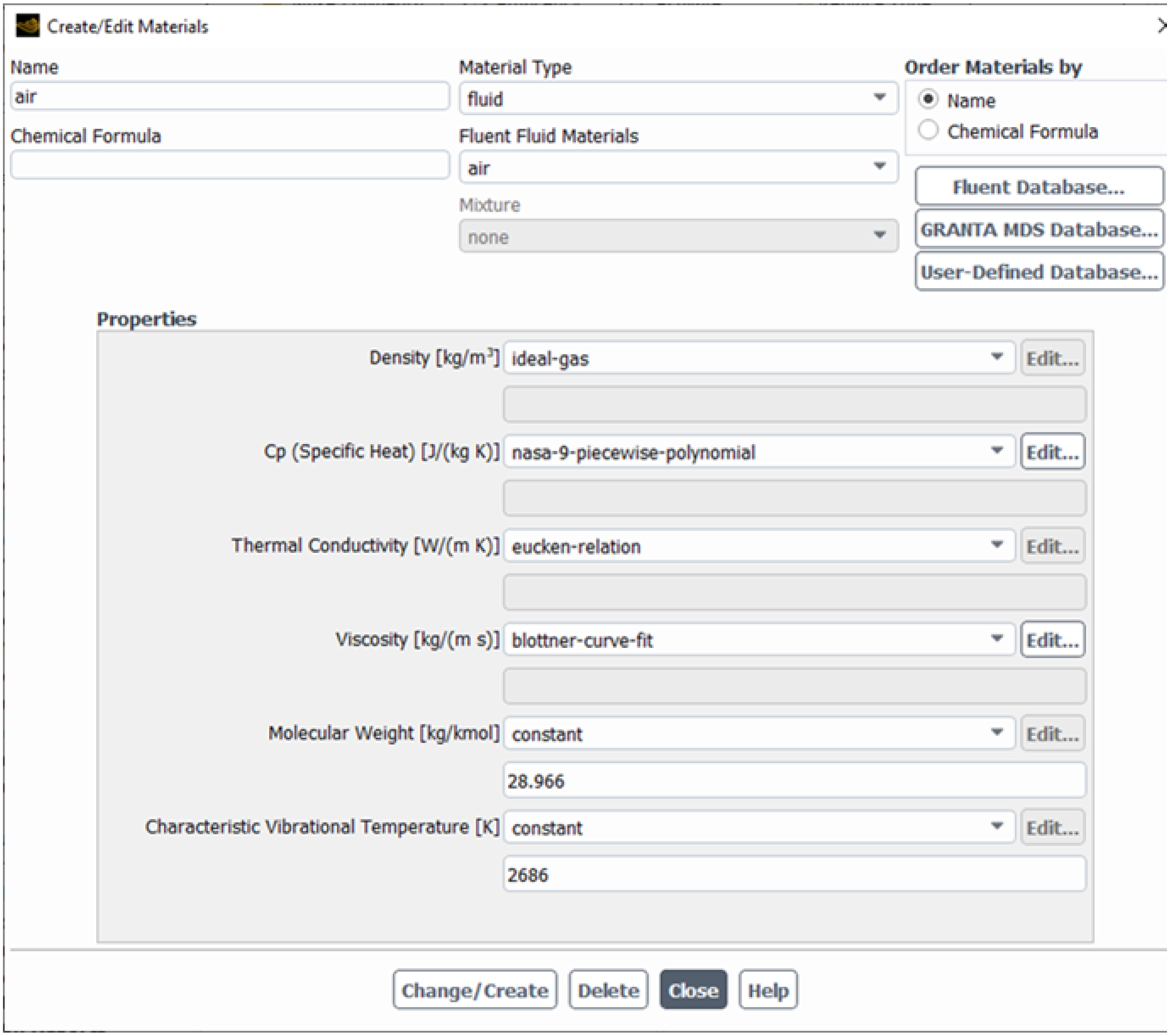

Create/Edit Materials
Name
air
Material Type
fluid
Order Materials by
Name
Chemical Formula
Chemical Formula
Fluent Fluid Materials
air
Mixture
none
Fluent Database...
GRANTA MDS Database...
User-Defined Database...
Properties
Density [kg/m3]
ideal-gas
Edit...
Cp (Specific Heat) [J/(kg K)]
piecewise-polynomial
Edit...
Thermal Conductivity [W/(m K)]
kinetic-theory
Edit...
Viscosity [kg/(m s)]
sutherland
Edit...
Molecular Weight [kg/kmol]
constant
Edit...
28.966
L-J Characteristic Length [Angstrom]
constant
Edit...
3.711
L-J Energy Parameter [K]
constant
Edit...
78.6
Change/Create
Delete
Close
Help



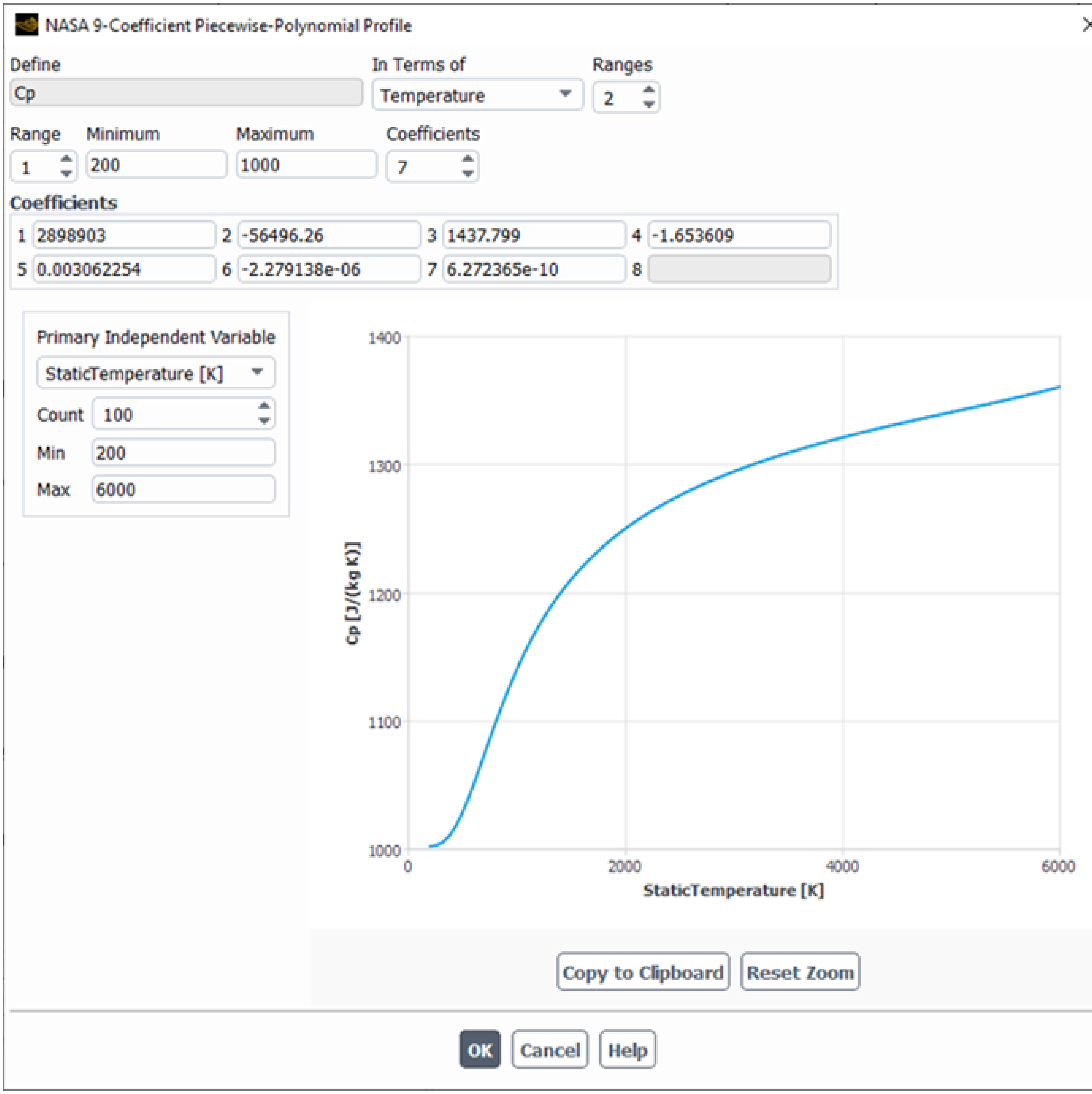
Piecewise-Polynomial Profile
Define
Cp
In Terms of
Temperature
Ranges
2
Range
Minimum
Maximum
Coefficients
1
100
1000
8
Coefficients
1 1161.482
2 -2.368819
3 0.01485511
4 -5.034909e-05
5 9.92857e-08
6 -1.111097e-10
7 6.540196e-14
8 -1.573588e-17
Primary Independent Variable
StaticTemperature [K]
Count 100
Min 100
Max 3000
Cp [J/(kg K)]
3000
2500
2000
1500
1000
0
1000
2000
3000
StaticTemperature [K]
Copy to Clipboard
Reset Zoom
OK
Cancel
Help

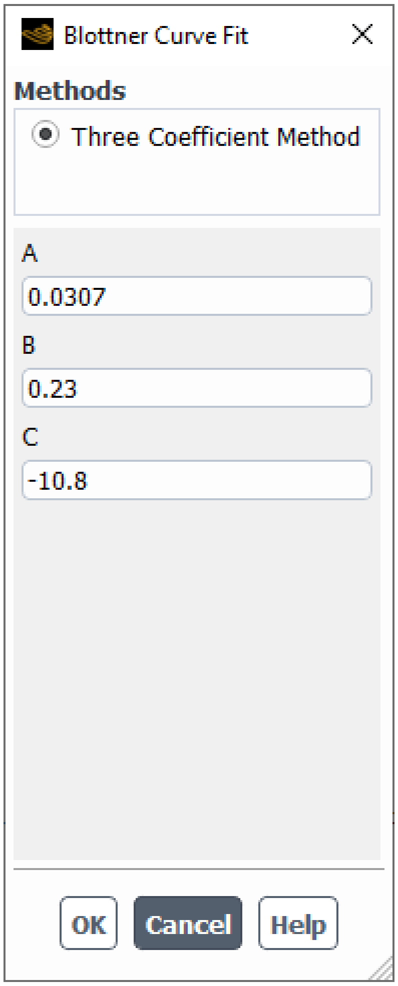



Fluent Aero's alternate default air properties, when the **Two Temperature Model** is enabled, is shown below:



Create/Edit Materials
Name
air
Material Type
fluid
Order Materials by
Name
Chemical Formula
Chemical Formula
Fluent Fluid Materials
air
Fluent Database...
Mixture
none
GRANTA MDS Database...
User-Defined Database...
Properties
Density [kg/m3]
ideal-gas
Edit...
Cp (Specific Heat) [J/(kg K)]
nasa-9-piecewise-polynomial
Edit...
Thermal Conductivity [W/(m K)]
eucken-relation
Edit...
Viscosity [kg/(m s)]
blottner-curve-fit
Edit...
Molecular Weight [kg/kmol]
constant
Edit...
28.966
Characteristic Vibrational Temperature [K]
constant
Edit...
2686
Change/Create
Delete
Close
Help


NASA 9-Coefficient Piecewise-Polynomial Profile
Define
Cp
In Terms of
Temperature
Ranges
2
Range
1
Minimum
200
Maximum
1000
Coefficients
7
Coefficients
1 2898903
2 -56496.26
3 1437.799
4 -1.653609
5 0.003062254
6 -2.279138e-06
7 6.272365e-10
8
Primary Independent Variable
StaticTemperature [K]
Count
100
Min
200
Max
6000
Cp [J/(kg K)]
1400
1300
1200
1100
1000
0
2000
4000
6000
StaticTemperature [K]
Copy to Clipboard
Reset Zoom
OK
Cancel
Help




→ **Case Settings**

When selected, Fluent Aero will not apply settings to the **Air Properties**. Instead, it will use the **Material Properties for Air** as setup in the initial case file, or from the **Solution Workspace** window (in **Materials** → **Fluid** → **air**). If the **Two Temperature Model** is set to **Case settings**, then the **Air Properties** is also set to **Case settings**.

- **Solution**

  – **Solver Methods**

  This setting defines the solver settings used to perform the calculations.

  → **Default**

  When **Default** is selected, Fluent Aero will apply a subset of default Solution Methods to the background Solver session for Density based and Pressure based, respectively. More information on these default settings can be found in Table 33.1: Mapping of Fluent Aero Solve Settings to Their Equivalent Settings in Fluent Solution Workspace: Material Properties, Iterations, Turbulence Model, Solver Type and Initialization Method (p. 513) and Table 33.2: Mapping of Fluent Aero Solve Settings to Their Equivalent Settings in Fluent

Solution Workspace: Solver Methods and Solver Strategy When Solver Type Is Set to Density Based (p. 515).

→ **Case settings**

When **Case settings** is selected, Fluent Aero will not apply any settings to the Solution Methods of the background Solver session. The Solution Methods already specified in the background Solver session will be used. This option allows an advanced user to have full access to the list of Solution Methods available of the Solver. To make changes to the Solution Methods to the background Solver session, use the Show Solution Workspace command, and in the Window that appears, make the appropriate changes in the Methods panel.

- **Flow Range**

  This option only appears if **Solver Type** is set to **Density based**.

  This setting allows the user to specify the expected flow range. By default, this is set to **Automatic**, which will use the design point conditions to set **the** flow range for each design point. In **Automatic** mode, the inlet **Mach** number will be used to select the flow range associated with each Design Point, using the following criteria:

  → $0.0 < M < 0.7$ → **Subsonic**

  → $0.7 \leq M < 1.4$ → **Transonic**

  → $1.4 \leq M < 2.5$ → **Low-Supersonic**

  → $2.5 \leq M < 4.5$ → **High-Supersonic**

  → $4.5 \leq M$ → **Hypersonic**

  However, you can manually set the flow range to either **Subsonic**, **Transonic**, **Low-Supersonic**, **High-Supersonic**, and **Hypersonic** if they know that their design points operate in a given flow regime. This option will help Fluent Aero tune various solver options for each type of simulation.

- **Solution Control**

  This setting is used to select the solver convergence strategy. The default option is Steering, but **Pseudo Transient**, **CFL** and **Case settings** can also be selected. **Steering** is not available if the **Solver Type** is set to **Pressure based**. If **Case settings** is selected, the solver strategy setup in the initial case file, or from the Solution Workspace window, will be used. Depending on the selection, a number of other settings may be revealed.

  → The following settings appear when **Solver Strategy** is set to **Pseudo Transient**:

  **Time Scale Factor**: controls the speed of convergence. Lower values result in a slower convergence speed but also potentially a better stability. The default value is 1. See Using the Solver within the Fluent User Guide for more information.

  → The following settings appear when **Solver Strategy** is set to **CFL**:

**Courant Number**: controls the speed of convergence. Lower values result in a slower convergence speed but also potentially a better stability. The default value is 5. See Using the Solver within the Fluent User Guide for more information.

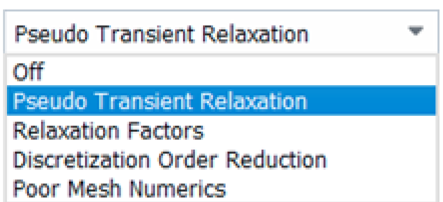
→ The following settings may appear when **Solver Strategy** is set to **Steering**:

→ **First to Second Order Blending**, **Initial Courant Number**, **Maximum Courant Number**, and **Explicit Under-Relaxation Factor** all control the speed and aggressiveness of convergence. The default values will change based on the selection made for the **Flow Range**. See Using the Solver within the Fluent User Guide for more information.

– **Auto Convergency Strategy**

This setting attempts to improve the convergence of cases that do not meet the convergence conditions in the first set of iterations. After the first set of iterations have been completed for a design point (500 iterations by default), the convergence conditions are checked. If convergence is not met, certain solver settings are modified to attempt to improve convergence. The settings that are modified depend on the option selected. This setting modification process can be performed a single time, or multiple times, depending on the selection. When a new Design Point calculation is started, the setting modifications are removed. By default, the **Auto Convergence Strategy** is set to **Off**.

The following options are available




→ **Off**

After a first set of iterations are performed, no attempt to improve convergence is made. No settings are modified, and no additional set of iterations is performed. The Design Point will be reported as not meeting the convergence conditions, and the next Design Point calculation will start.

→ **Pseudo Transient Relaxation**

If the convergence conditions of a Design Point are not met after the first set of iterations, the pseudo transient time scale factor is reduced by 50%, and another set of iterations is performed. If the convergence conditions are not met after the second set of iterations, the pseudo transient time scale factor is reduced by another 50% and a third set of iterations is performed. If the convergence conditions are not met after the third set of iterations, the Design Point calculation is stopped, and the next Design Point is calculated with the pseudo transient time scale factor set back at its initial value.

---

**Note:**

This setting is only effective if **Solver Strategy** is set to **Pseudo Transient**.

---

→ **Relaxation Factors**

If the convergence conditions of a Design Point are not met after the first set of iterations, the pseudo transient explicit relaxation factors are set to values which are 50% smaller than their default values, and another set of iterations is performed. If the convergence conditions are not met after the second set of iterations, the higher order term relaxation is set to all variables, and a third set of iterations is performed. If the convergence conditions are not met after the third set of iterations, the Design Point calculation is stopped, and the next Design Point is calculated with the pseudo transient explicit relaxation factors set back at their initial values, and the higher order term relaxation set back to flow variables only.

→ **Discretization Order Reduction**

If the convergence conditions of a Design Point are not met after the first set of iterations, all of the Spatial Discretization settings are set to First Order, and another set of iterations is performed. If the convergence conditions are not met after the second set of iterations, the Design Point calculation is stopped, and the next Design Point is calculated with the Spatial Discretization settings set back to Second Order.

→ **Poor Mesh Numerics**

If the convergence conditions of a Design Point are not met after the first set of iterations, the Design Point solution is first re-initialized, then poor mesh numerics is enabled, and another set of iterations is performed. If the convergence conditions are not met after the second set of iterations, the Design Point calculation is stopped, and the next Design Point is calculated with the poor mesh numerics disabled.

- **Initialization**

  – **Initialization Method**

  This setting defines the method used to initialize the simulation. Currently, **FMG**, **Hybrid**, **Standard**, and C**ase settings** are available. The default setting is **FMG**. If **Case settings** is selected, the initialization method setup in the initial case file, or from the Solution Workspace window, is used. More information on these settings associated with each **Initialization Method** can be found in Table 33.1: Mapping of Fluent Aero Solve Settings to Their Equivalent Settings in Fluent Solution Workspace: Material Properties, Iterations, Turbulence Model, Solver Type and Initialization Method (p. 513).

  – **Initialize Between Design Points**

  This setting is enabled by default. In this case, an Initialization will be performed at the beginning of the calculation of each Design Point. When unchecked, the current design point will begin to calculate using the results of the previous design point simulation.

- **Convergence**

  – **Convergence Cutoff**

  This setting defines the convergence residual value at which the calculation of a design point will stop if all the solver residuals drop below this value. By default, the convergence cutoff is set to 1e-4. The convergence condition is reached only when both **Residuals Convergence Cutoff** and **Aero Coeff. Conv. Cutoff** are satisfied.

– **Aero Coeff. Conv. Cutoff**

  This setting defines the convergence cutoff for the Lift, the Drag and the Moment coefficients. The default value is 2e-4. The convergence condition is reached only when both **Residuals Convergence Cutoff** and **Aero Coeff. Conv. Cutoff** are satisfied.

– **Aero Coeff. Conv. Previous Values**

  This setting defines the number of iterations prior to the current iteration to be considered to estimate the convergence of the monitored coefficients. The default value is 10. The coefficient of the current iteration will be compared with each of the last 10 values and the maximum difference will be used to compare with the cutoff setting. Note that this maximum difference is normalized by the value of current iteration.

- **Journals**

  This section allows you to setup Fluent journal (`.jou`) files that can be used to add extra control and setup to the Fluent Solver at various points within the Design Point calculation process. These journal files will be read by the Fluent Solver directly so they can only include settings that would affect the Fluent Solution Workspace. The journal files cannot be used to change any settings that are located inside the Fluent Aero Workspace. The default location to put these journal files is inside the **Simulation** folder (alongside the Simulation case file).

  – **Run Journal**

    This setting allows you to execute a journal file at the beginning of a run, after you press the **Calculate** command, and after all Fluent Aero's default solver setup has been applied, but before the initialization or calculation of any Design Point has been carried out. The default name to use for this journal file is `run.jou`.

  – **Design Point Journal**

    This setting allows you to execute a journal file right before the initialization or calculation of each design Point. The default name to use for this journal file is `designpoint.jou`.

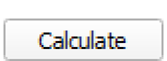
  – **Initialization Journal**

    This setting allows you to execute a journal file to replace the initialization routine that Fluent Aero would use for a Design Point. When this setting is **Enabled**, the **Initialization Method** setting will be set to **Journal file**. The default name to use for this journal file is `initialization.jou`.

  The following command buttons are available at the bottom of the **Solve** → **Properties** panel.

  The following **Calculate** command button is available when Fluent Aero is waiting for the user to start the calculation:

- **Calculate**

Once all the parameters and variables are set, Fluent Aero calculations can be started by clicking on the **Calculate** button which is located on the bottom of the **Solve** panel, or by right-clicking on **Solve** in the **Outline View** tree and selecting the **Calculate** command.

When the calculation starts, Fluent Aero will refer to the **Input: Design Points** table **Status** column to determine if and how a particular Design Point will be calculated.

– **Design Points** with **Status** set to **Needs Update** will be calculated starting from the beginning, including initialization.

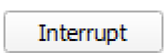
– **Design Points** with **Status** set to **Continue to Update** or **Initialized** will be calculated starting from the previously written solution file (`out.0*.dat[.h5]`) associated with that design point.

– **Design Points** with **Status** set to **Initialize** will be only be initialized but no solver iterations will be calculated.

– **Design Points** with **Status** set to **Updated**, **Do Not Update**, **Interrupted** or **Error**, will not be calculated.

The following **Interrupt** command button is available when Fluent Aero is calculating:

- **Interrupt**




While the calculation is being performed, an **Interrupt** button appears at the bottom of the **Solve** panel. If Interrupt is selected, the calculation will be interrupted at the next iteration.

---

**Important:**

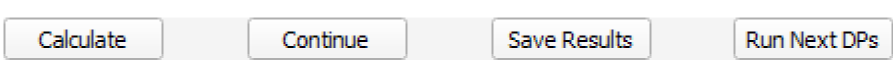
Interrupting a calculation does not save the results files (`out.0*.dat[.h5]` and `out.0*.cas[.h5]`) to the **Design Point** folder, and does not update the values in the results tables (**Table:Summary**, **Table:Coefficients**, **Table:Residuals**). If you would like to save your current solution to the results files and update the results tables, select **Save Results**.

---

After the calculation is complete or interrupted, additional commands will appear in the **Solve** panel.

The following command buttons are available after a calculation has been interrupted:

- **Continue**




By clicking on the **Continue** button, Fluent Aero will continue the interrupted design point calculation from their previous state. The solution will not be initialized, and an additional set of iterations will be performed.

Before a **Continue** is performed, any setting in the **Solve** panel can be adjusted, and these settings will be applied to the calculation. For example, a user can change the **Time Scale Factor**, and this new **Time Scale Factor** will be used for the remainder of the continued calculation.

However, while using the **Continue** command, you cannot make any changes to the **Geometric Properties**, **Simulation Conditions**, **Component Groups** panel, or the **Input Design Points Table**, as changes to these settings will set the **Design Point Status** to **Needs Update**, and therefore will require a full update from a new initial solution.

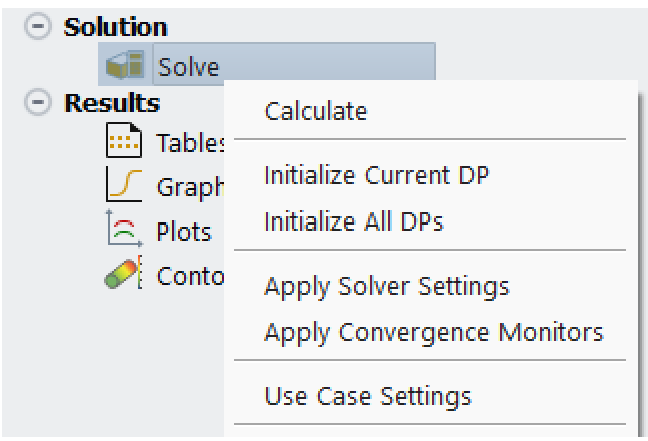
- **Save Results**

  When the calculation of a design point is interrupted, the data solution file is not saved and the tabulated results values are not computed or filled. If the user wants to save the results of the design point at the interrupted location, clicking **Save Results** will cause the data file to be saved and the tabulated results to be computed and filled.

- **Run Next DPs**

  This command will move on from the current design point that was interrupted, and cause the next design points in the sequence to be calculated. The data solution file and the tabulated results of the current design point will not be saved, unless the user selects the **Save Results** command described above.

  The following right-click commands are available by right-clicking the **Solve** step.




> **Note:**
>
> Commands such as **Calculate** and **Interrupt**, which are also available as command buttons, are instead described in Simulation Conditions (p. 466).

- **Initialize Current DP**

  This command will initialize the current design point only. If there has not yet been any Design Points calculated, the first Design Point will be initialized. No iterations will be calculated. This allows the user to verify if the initialization procedure is working well for a single Design Point before continuing with the calculation.

  The Current Design Point is typically the most recent Design Point that was calculated or loaded, and will have its `.dat[.h5]` file shown in bold in the **Project View**. It is also possible to see the Current Design Point by right-clicking on the simulation folder, selecting **Properties** to view the simulation metadata, and finding the current key.

- **Initialize All DPs**

  This command will initialize all design points. No iterations will be calculated. This allows you to verify if the initialization procedure is working well for all Design Points before continuing with the calculation.

- **Apply Solver Settings**

  In a typical workflow, when the **Calculate** command is used, Fluent Aero will first setup all settings in the Solver and Solution Workspace and then initialize and calculate the design points. If you would instead like to first verify what solver settings are setup inside the solution workspace before continuing with the calculation, this **Apply Solver Settings** command can be used. By selecting this command, Fluent Aero will immediately setup all solver settings inside the Solution Workspace, allowing you to open the Solution Workspace to verify the settings before calculating.

- **Apply Convergence Monitors**

  In a typical workflow, when the **Calculate** command is used, Fluent Aero will first setup all convergence monitors in the Solver and Solution Workspace and then initialize and calculate the design points. If you would instead like to first verify what convergence monitors are setup inside the solution workspace before continuing with the calculation, this **Apply Convergence Monitors** command can be used. By selecting this command, Fluent Aero will immediately setup all convergence monitors inside the Solution Workspace, allowing you to open the Solution Workspace to verify the settings before calculating.

- **Use Case Settings**

  This command will set every option in the **Properties – Solve** panel, under the **Show Advanced Settings** flag to **Case settings**, if available. Setting a **Solve** option to **Case settings** will cause Fluent Aero to not apply any settings to that equivalent option in the Fluent Solution Workspace. The setting already specified in the background Solver and Solution Workspace will be used. By setting all possible **Solve** commands to **Case settings**, you ensure that the majority of Solver settings from their initial Case file can be used.

  **Important:**

  Care should be taken when using this option, because doing so may cause Fluent Aero to calculate with settings that may not be fully supported.

  Therefore, it is only recommended for advanced Fluent Aero users. A full list of the mapping between Fluent Aero's Solve settings and their equivalent setting in Fluent Solution Workspace can be found in Mapping of the Solve Settings of Fluent Aero Workspace to Their Equivalent Settings in Fluent Solution Workspace (p. 513).

## 33.8.5. Modifying Settings After Results Have Been Calculated

After **Results** have been calculated, you may go back and modify settings inside **Setup** (**Geometric Properties**, **Simulation Conditions**, **Component Groups**), **Solution** (**Solve**) or in the **Input: Design Point Table**. Depending on which settings are modified, the **Results** may no longer be up to date, and the status of the Design Points may change.

**Modifying settings that define all Design Points will cause all Design Points to be set to Needs Update.**

Modifying any of the following settings will cause the **Status** of all Design Points to be set to **Needs Update**.

- Most settings in the **Geometric Properties** step will cause all Design Points to be set to **Needs Update**, except for the following exceptions:
  - **Domain Type** cannot be changed after **Results** have been calculated.
  - Changing **Reference Length** and **Reference Area** trigger a special dialog, which is described in an upcoming section.
- Changing any input **Parameter** type in **Simulation Conditions** will cause all Design Points to be set to **Needs Update**.
- Changing the constant value of any parameter that is set to **Distribution** → **Constant** in **Simulation Conditions** will cause all Design Points to be set to **Needs Update**.
- Changing the constant value of any input parameter defined on a specific zone in **Component Groups** will cause all Design Points to be set to **Needs Update**.
- Changing the name of any custom input parameter defined on a specific zone in **Component Groups** will cause all Design Points to be set to **Needs Update**.
- Changing the zone type or renaming any zone listed in **Component Groups** will cause all Design Points to be set to **Needs Update**. This procedure will also trigger additional changes, such as a deletion of the `run.settings` file, as described in an upcoming section.

**Modifying settings in the Input Design Point Table will cause an individual Design Point to be set to Needs Update.**

Changing an input value in the **Input Design Point Table** that represents a single Design Point will cause the **Status** of that Design Point to be set to **Needs Update**. If you would like to refresh the values of the Design Point Table to match those that were calculated, you may use the **Reload DP Table** command.

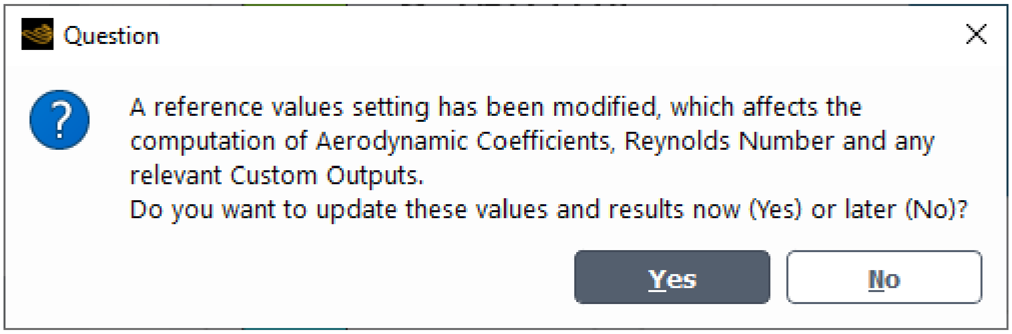
**Modifying settings Solution → Solve will not change the status of a Design Point.**

Modifying any settings in the **Solve** panel will not change the status of any Design Point. This is because a user may want to calculate a Design Point simulation using a preliminary set of solver settings, then modify the settings and continue the calculation in order to follow a specific convergence method. A user must take care to ensure that the settings used to calculate each design point are as intended. To load the most recent **Solve** settings used to calculate an individual Design Point, they can right-click on a design point folder in the **Project View** and select the **Load Design Point Solve Settings** command.

**Modifying Reference Length and Reference Area**

The **Reference Length** and **Reference Area** has no impact on the CFD solution itself (`.dat[.h5]` file), but does have an impact on the aerodynamic coefficients because they are used to non-dimentionalize the forces. Therefore, if a user modifies the **Reference Length** or **Reference Area** after calculating any **Results**, a procedure allows the user to update these coefficients only without having to recalculate any Design Point solution.

If **Reference Length** or **Reference Area** are modified, a panel will appear.




If a user selects **Yes**, Fluent Aero will immediately update all Aerodynamic Coefficients and the Reynolds number in all results tables or files. If any relevant Custom Output is used, Fluent Aero will also load the solution file for every Design Point, and recompute the custom output value before updating the tables (this process may take some time).

If a user selects **No**, Fluent Aero will not yet update the Aerodynamic Coefficients, and the Status of all Design Points will be set to **Reference Values Modified**. When the user is ready to update these values, they should use the **Update Results** command button located in the **Results** tables properties panel. This will update the values of the Aerodynamic Coefficients, Reynolds Number and relevant Custom Outputs and set the status of each Design Point back to **Updated**. It is important for a user to update the results before they close a simulation to ensure that the values are updated appropriately.

**Modifying a Zone Type or Name**

Generally, it is recommended for the user to setup the fully zone names and types before calculating any **Results**. If the user modifies the type or name of any zone listed in **Component Groups** after calculating **Results**, the following will occur:

- The `run.settings` file, which contains the list of all zones, their names, types and their zone specific settings, will be deleted.
- The **Status** of all Design Points will be set to **Needs Update**. Furthermore, in this case, the status will also be saved to the Design Point folders, and therefore the Status cannot be refreshed by using the **Refresh Status** or **Reload DP Table** command.
- The following message will be printed to the console.

```
A boundary type used in previous Results has changed :
   - Status for all DPs will be set to Needs Update
   - The current run.settings file is out of date and will be deleted

 It is recommended to save the case file to update and sync with the run.settings.
```

The user must then re-save the `run.settings` to re-sync it with the case file by doing one of the following:

- Selecting **File** → **Save Case**, or
- Clicking **Calculate**, or
- Clicking **Yes** when asked to save the case after closing the solver.

If you changed a zone type or name after calculating the results, it is important to re-save the `run.settings` file in one of the above recommended ways. Failing to do so could cause issues the next time you load your simulation.

## 33.8.6. Mapping of the Solve Settings of Fluent Aero Workspace to Their Equivalent Settings in Fluent Solution Workspace

When a user clicks **Calculate**, the settings inside the **Solve** panel of Fluent Aero cause their equivalent setting inside the loaded Solver session to be updated. The following tables describes the mapping between the settings shown in Fluent Aero and their equivalent settings inside the Fluent Solution Workspace.

**Note:**

Many Fluent Aero **Solve** panel settings control a group of settings inside the Fluent Solution Workspace.

**Table 33.1: Mapping of Fluent Aero Solve Settings to Their Equivalent Settings in Fluent Solution Workspace: Material Properties, Iterations, Turbulence Model, Solver Type and Initialization Method**

| **Fluent Aero applies the following equivalent settings to Solution Workspace** | | | |
|---|---|---|---|
| **Setting in Fluent Aero** | | **Equivalent Group of Settings in Solution Workspace** | |
| **Solve → Iterations** | (value) | Run Calculation → Number of Iterations | (value) |
| **Solve → Solver Type** | **Density based** | General → Solver → Type | Density-Based |
| | **Pressure based** | General → Solver → Type | Pressure-Based |
| **Solve → Turbulence Model** | **K-Omega SST** | Models → Viscous | K-omega (2 eqn) |
| | | Models → Viscous → k-omega (2eqn) | SST |
| | | Models → Viscous → options → Viscous Heating | Enabled |
| | | Models → Viscous → options → Production Limiter | Enabled |
| | **Transition SST** | Models → Viscous | Transition SST (4-eqn) |
| | | Models → Viscous → options → Viscous Heating | Enabled |
| | | Models → Viscous → Roughness Correlation | Enabled |

| | | | |
|---|---|---|---|
| | **Case settings** | Keep as is currently defined in Solution Workspace | |
| **Solve → Two Temperature Model** | **Automatic** | Models → Energy → Two Temperature Model | If Flow Range = Hypersonic → Enabled. If Flow Range is not Hypersonic → Disabled |
| | **Enabled** | Models → Energy → Two Temperature Model | Enabled |
| | **Disabled** | Models → Energy → Two Temperature Model | Disabled |
| | **Case settings** | Keeps as currently defined in Solution Workspace | |
| **Solve → Air Properties** | **Air default (standard)** | Models → Energy | On |
| | | Materials → Fluid | air |
| | | Materials → Fluid → air → Density | Ideal Gas |
| | | Materials → Fluid → air → Cp | Piecewise polynomial |
| | | Materials → Fluid → air → Thermal Conductivity | Kinetic theory |
| | | Materials → Fluid → air → Viscosity | Sutherlands |
| | | Materials → Fluid → air → Molecular Weight | Constant = 28.966 |
| | **Air default (if Two Temp. Model Enabled or activated while set to Automatic)** | Models → Energy | On |
| | | Materials → Fluid | air |
| | | Materials → Fluid → air → Density | Ideal Gas |
| | | Materials → Fluid → air → Cp | nasa-9-piecewise-polynomial |
| | | Materials → Fluid → air → Thermal Conductivity | euken relation |
| | | Materials → Fluid → air → Viscosity | blottner curve fit |
| | | Materials → Fluid → air → Molecular Weight | Constant = 28.966 |
| | **Case settings** | Keep the air material properties that are currently defined in Solution Workspace | |
| **Solve → Initialization Method** | **Standard** | Initialization → Initialization Method | Standard Initialization method with Initial Values filled with inlet conditions per Design Point |
| | **Hybrid** | Initialization → Initialization Method | Hybrid Initialization |
| | | Initialization → Initialization Method → More Settings… → Number of Iterations | 10 |
| | | Initialization → Initialization Method → More Settings… → Use External-Aero Favorable Settings | Enabled |

| | | | |
|---|---|---|---|
| | **FMG** | Initialization → Initialization Method | Standard Initialization method with Initial Values filled with inlet conditions per Design Point |
| | | Run Calculation → Options → Use FMG Initialization | Enabled |
| | | Run Calculation → Solution Steering → More Settings… → FMG Settings | Default for Flow Range: (Subsonic, Transonic, Low-Supersonic, High-Supersonic): /solve/initialize set-fmg-initialization 4 0.001 100 0.001 200 0.001 400 0.001 500 0.75 yes, (Hypersonic): /solve/initialize set-fmg-initialization 2 0.001 100 0.001 200 0.75 yes |
| | **Case Settings** | Keep as is currently defined in Solution Workspace | |

**Table 33.2: Mapping of Fluent Aero Solve Settings to Their Equivalent Settings in Fluent Solution Workspace: Solver Methods and Solver Strategy When Solver Type Is Set to Density Based**

| **If Solver Type is set to Density Based, the following settings are applied** | | | |
|---|---|---|---|
| **Setting in Fluent Aero** | | **Equivalent Group of Settings in Solution Workspace** | |
| **Solve → Solver Methods** | **Default** | Methods → Solution Methods → Pressure-Velocity Coupling → Flux Type | Rhie-Chow |
| | | Methods → Solution Methods → Pressure-Velocity Coupling → Formulation | Implicit |
| | | Methods → Solution Methods → Spatial Discretization → Gradient | Green-Gauss Node Based |
| | | Methods → Solution Methods → Spatial Discretization → Flow | Second Order Upwind |
| | | Methods → Solution Methods → Spatial Discretization → Turbulent Kinetic Energy | Second Order Upwind |
| | | Methods → Solution Methods → Spatial Discretization → Specific Dissipation Rate | Second Order Upwind |
| | | Methods → Solution Methods → Higher Order Term Relaxation | Enabled |
| | | Methods → Solution Methods → Convergence Acceleration For Stretched Meshes | Enabled |
| | | /solve/set convergence-acceleration-for-stretched-meshes yes 2 | |
| | **Case Settings** | Keep as is currently defined in Solution Workspace | |

| | | | |
|---|---|---|---|
| **Solve → Flow Range** | **Automatic** | Sets Flow Range per Design Point based on the Freestream Mach number. The setting per DP subsequently affects the Solve → Steering settings | 0 < M < 0.7 → Subsonic, 0.7 < M < 1.4 → Transonic, 1.4 < M < 2.5 → Low-Supersonic, 2.5 < M < 4.5 → High-Supersonic, 4.5 < M → Hypersonic |
| | **Subsonic** | Selecting one of these settings will force Fluent Aero to use Solve → Steering settings associated with the selected Flow Range | |
| | **Transonic** | | |
| | **Low-Supersonic** | | |
| | **High-Supersonic** | | |
| | **Hypersonic** | | |
| **Solve → Solution Control** | **Pseudo Transient** | Methods → Solution Methods → Pseudo Transient | Enabled |
| | | Methods → Solution Methods → Convergence Acceleration For Stretched Meshes | Disabled |
| | | Run Calculation → Pseudo Transient Settings → Time Step Method | Automatic |
| | | Run Calculation → Options → Solution Steering | Disabled |
| | | (rpsetvar 'physical-time-step 0) | |
| | | (rpsetvar 'amg/group-size 8) | |
| | | (rpsetvar 'amg/post-relaxations 3) | |
| | | (rpsetvar 'relaxation-method "ilu") | |
| | | /solve/set/high-speed-numerics/expert 5 | |
| | **Solve → Time Scale Factor → (value)** | Run Calculation → Pseudo Transient Settings → Time Scale Factor | (value) |
| | **CFL** | Methods → Solution Methods → Pseudo Transient | Disabled |
| | | Run Calculation → Options → Solution Steering | Disabled |
| | | (rpsetvar 'amg/group-size 2) | |
| | | (rpsetvar 'amg/post-relaxations 1) | |
| | | (rpsetvar 'relaxation-method "gauss-seidel") | |
| | | /solve/set/high-speed-numerics/expert 5 | |
| | **Solve → Courant number → (value)** | Controls → Solution Controls → Courant Number | (value) |
| | **Steering** | Run Calculation → Options → Solution Steering | Disabled |

| | | | |
|---|---|---|---|
| | | Methods → Solution Methods → Pseudo Transient | Disabled |
| | | (rpsetvar 'amg/group-size 2) | |
| | | (rpsetvar 'amg/post-relaxations 1) | |
| | | (rpsetvar 'relaxation-method "gauss-seidel") | |
| | | /solve/set/high-speed-numerics/expert 5 | |
| | | Run Calculation → Solution Steering → More Settings… → Steering Settings → Stage 2 → Courant Number Update Interval | Default per Flow Range: Subsonic: 20, Transonic: 20, Low-Supersonic: 30, High-Supersonic: 30, Hypersonic: 30 |
| | | Run Calculation → Solution Steering → More Settings… → Steering Settings → Stage 1 → After Iterations | Default per Flow Range: Subsonic: 0, Transonic: 0, Low-Supersonic: 100, High-Supersonic: 100, Hypersonic: 200 |
| | | Run Calculation → Solution Steering → More Settings… → Steering Settings → Stage 1 → Duration | Default per Flow Range: Subsonic: 100, Transonic: 100, Low-Supersonic: 100, High-Supersonic: 100, Hypersonic: 100 |
| | **Solve → First to Second Order Blending → (value)** | Run Calculation → Solution Steering → First to Higher-Order Blending | (value) or default per Flow Range: Subsonic: 1, Transonic: 1, Low-Supersonic: 1, High-Supersonic: 1, Hypersonic: 0.5 |
| | **Solve → Initial Courant Number → (value)** | Run Calculation → Solution Steering → More Settings… → Steering Settings → Courant Number → Initial | (value) or default per Flow Range: Subsonic: 5, Transonic: 5, Low-Supersonic: 5, High-Supersonic: 5, Hypersonic: 0.5 |
| | **Solve → Maximum Courant Number → (value)** | Run Calculation → Solution Steering → More Settings… → Steering Settings → Courant Number → Maximum | (value) or default per Flow Range: Subsonic: 200, Transonic: 200, Low-Supersonic: 200, High-Supersonic: 200, Hypersonic: 4 |
| | **Solve → Under-Relaxation Factor → (value)** | Run Calculation → Solution Steering → More Settings… → Steering Settings → Explicit Under-Relaxation Factor | (value) or default per Flow Range: Subsonic: 0.75, Transonic: 0.75, Low-Supersonic: 0.75, High-Supersonic: 0.75, Hypersonic: 0.5 |

| | | |
|---|---|---|
| | **Case Settings** | Keep as is currently defined in Solution Workspace |

**Table 33.3: Mapping of Fluent Aero Solve Settings to Their Equivalent Settings in Fluent Solution Workspace: Solver Methods and Solver Strategy When Solver Type Is Set to Pressure Based**

| **If Solver Type is set to Pressure Based, the following settings are applied** | | | |
|---|---|---|---|
| **Setting in Fluent Aero** | | **Equivalent Group of Settings in Solution Workspace** | |
| **Solve → Solver Methods** | **Default** | Methods → Solution Methods → Pressure-Velocity Coupling → Flux Type | Rhie-Chow |
| | | Methods → Solution Methods → Pressure-Velocity Coupling → Scheme | Coupled |
| | | Methods → Solution Methods → Spatial Discretization → Gradient | Green-Gauss Node Based |
| | | Methods → Solution Methods → Spatial Discretization → Pressure | Second Order Upwind |
| | | Methods → Solution Methods → Spatial Discretization → Density | Second Order Upwind |
| | | Methods → Solution Methods → Spatial Discretization → Momentum | Second Order Upwind |
| | | Methods → Solution Methods → Spatial Discretization → Turbulent Kinetic Energy | Second Order Upwind |
| | | Methods → Solution Methods → Spatial Discretization → Specific Dissipation Rate | Second Order Upwind |
| | | Methods → Solution Methods → Spatial Discretization → Energy | Second Order Upwind |
| | | Methods → Solution Methods → Higher Order Term Relaxation | Enabled |
| | **Case Settings** | Keep as is currently defined in Solution Workspace | |
| **Solve → Solution Control** | **Pseudo Transient** | Methods → Solution Methods → Pseudo Transient | Enabled |
| | | Run Calculation → Pseudo Transient Settings → Time Step Method | Automatic |
| | | Methods → Solution Methods → Convergence Acceleration For Stretched Meshes | Disabled |
| | | (rpsetvar 'physical-time-step 0) | |
| | | (rpsetvar 'amg/group-size 2) | |
| | | (rpsetvar 'amg/post-relaxations 1) | |
| | | (rpsetvar 'relaxation-method "gauss-seidel") | |
| | **Solve → Time Scale Factor → (value)** | Run Calculation → Pseudo Transient Settings → Time Scale Factor | (value) |
| | **CFL** | Methods → Solution Methods → Pseudo Transient | Disabled |
| | | (rpsetvar 'amg/group-size 2) | |
| | | (rpsetvar 'amg/post-relaxations 1) | |

| | | | |
|---|---|---|---|
| | | (rpsetvar 'relaxation-method "gauss-seidel") | |
| | **Solve → Courant number → (value)** | Controls → Solution Controls → Flow Courant Number | (value) |
| | **Case Settings** | Keep as is currently defined in Solution Workspace | |

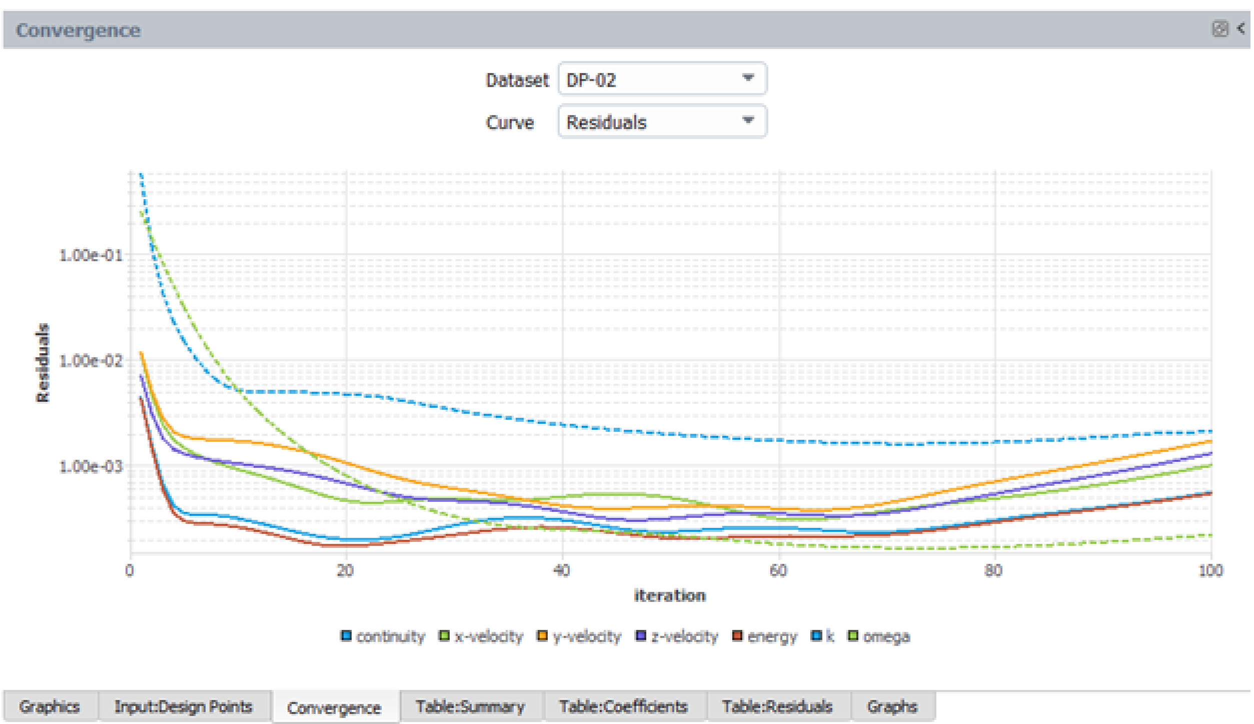
## 33.9. Viewing the Results of a Fluent Aero Simulation

The following section describes the possible ways to interact with and postprocess the results of a Fluent Aero simulation.

### 33.9.1. Convergence Plots in the Graphics Window

During and after the Fluent Aero Simulation, the **Convergence** window can be used to display the convergence plots of Residuals as well as any monitors such as Lift, Drag and Moment. The **Convergence** window is shown by default when a user begins a calculation. It is located in the same location as the Graphics window, and can be accessed at any time by selecting the **Convergence** tab.

The image below shows a **Convergence** window showing the **Residuals** of **Design Point 2**.




There are two items that can be used to control the content displayed in the **Convergence** plot window:

- **Dataset**

  Select the **Design Point** that you would like to see the **Residuals** or **Convergence** Monitors. Any **Design Point** that has been simulated will appear in the list of selectable **Design Points**.

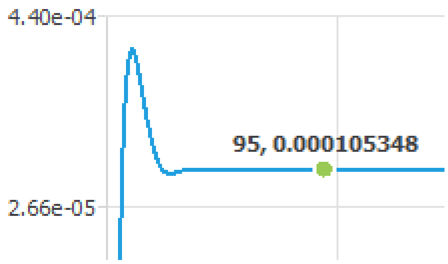
- **Curve**

  Select the **Residuals** or **Convergence** Monitors of the selected. **Residuals**, **lift-coefficient**, **drag-coefficient**, and **pitch-**, **yaw-** and **roll-moment-coefficient** are always available as selections in this menu when using Fluent Aero. Individual solver residuals and other solution monitors may also be available depending on your settings.

The convergence plots can be interacted with in the following ways:

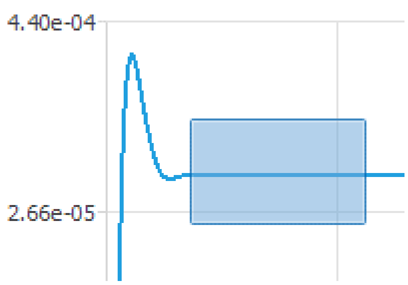
- **Display the value of a point on the curve**

  The value of a point on a curve can be displayed by clicking on a curve near the point of interest.



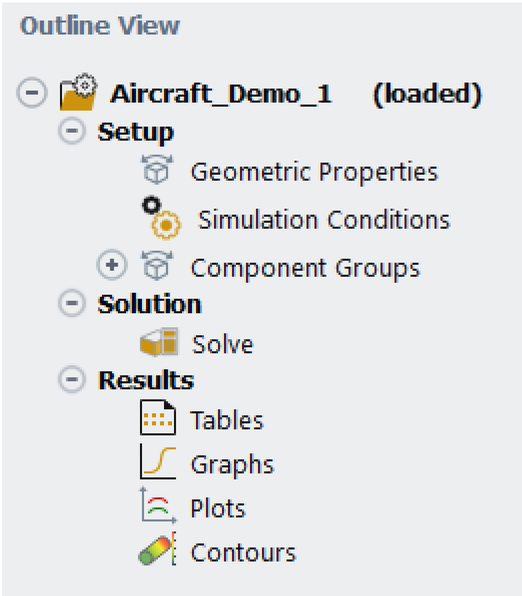

- **Zoom into a part of the curve**

  A section of the curve can be investigated further by zooming onto it using a **Shift**+Left-Click+Mouse-Drag.



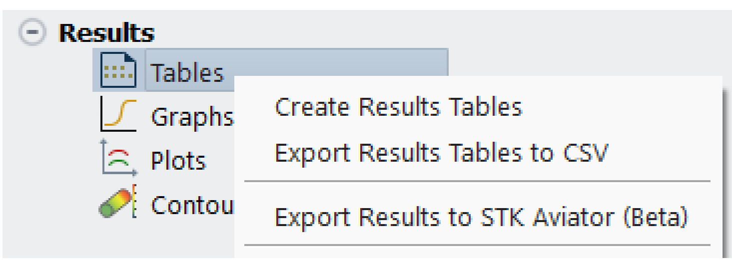

## 33.9.2. Results Menu

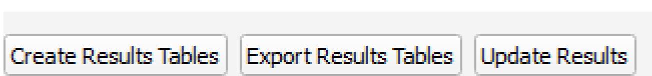
Once a Fluent Aero Simulation begins to calculate, a **Results** component menu will appear at the end of the **Outline View** tree. The **Results** component menu contains four sub-components that can be used to post-process the results; **Tables**, **Graphs**, **Plots**, and **Contours**.




### 33.9.2.1. Tables

The **Tables** component contains commands to display tabulated results of a Simulation. These commands are accessible by right-clicking on Tables in the **Outline View** window, or as buttons located at the bottom of the Tables properties window.



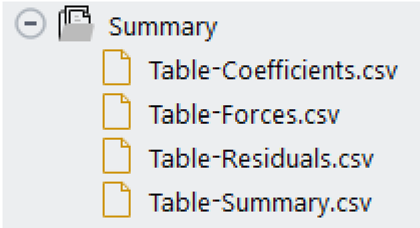




- **Create Results Table**

  This function will create a set of results tables called **Table:Summary**, **Table: Coefficients**, **Table:Forces**, **Table:Residuals**, and if required, **Table:Custom Output** in the same area as the **Graphics** window. This table presents selected input and output values associated with each Design Point that was calculated in the Simulation.

- **Export Results Tables to CSV**

  This function will export the results tables using comma separated values files `Table-Coefficients.csv`, `Table-Forces.csv`, `Table-Residuals.csv` and `Table-Summary.csv` located in a **Summary** folder inside **Results**.




The following figure shows an example of `Table-Summary.csv` file to illustrate the file format used.

```
DP,Mach,AoA [deg],AoS [deg],P [Pa],T [K],Reynolds,Avg Coeff Conv,Avg Residual,Criteria Met?
1,0.43,2,0,54019.9036,255.65,2.1340e+07,3.5523e-04,1.2988e-04,partially
2,0.43,4,0,54019.9036,255.65,2.1340e+07,1.9492e-04,1.0765e-04,partially
3,0.43,6,0,54019.9036,255.65,2.1340e+07,8.3860e-05,1.6652e-04,partially
4,0.43,8,0,54019.9036,255.65,2.1340e+07,3.0434e-03,1.6267e-04,no
5,0.43,10,0,54019.9036,255.65,2.1340e+07,1.6753e-02,2.1401e-04,no
```

- **Export Results to STK Aviator**

  This function will export the results of aerodynamic coefficients to a simple format `.aero` file supported by STK Aviator called `STK-Aviator.aero`. This file can be used to define the aerodynamic performance of an aircraft during an AGI STK Aviator simulation. The format of this file varies if **Altitude** was used as an input parameter or not.

  **Example 33.1: Format of a Simple STK-Aviator.aero File (without Altitude)**

```
Version:      10.1
RefAreaM2: 1
# Mach   Cl   Cd   AOArad
0.3 0.0228809 0.000595742 0.0872664626
0.3 0.0228809 0.000595742 0.0872664626
```

  **Example 33.2: Format of a Simple STK-Aviator.aero File (with Altitude)**

```
Version:      10.1
RefAreaM2: 1
AltitudeDependence: yes
# Alt   Mach   Cl   Cd   AOArad
1000 0.3 0.0228809 0.000595742 0.0872664626
1000 0.3 0.0228809 0.000595742 0.0872664626
```

  An `.aero` file has a number of requirements on the dataset used to generate a complete and valid file. These restrictions change depending if **Altitude** was used as an input parameter or not.

  Requirements for a complete and valid `STK-Aviator.aero` file (without Altitude):

  – A minimum of 3 Design Points are required.

  – No points with a repeated combination of Mach and AoA should be used.

  – At least 3 distinct non-colinear points on the Mach vs. Cl plane are required.

Requirements for a complete and valid `STK-Aviator.aero` file (with Altitude):

- A minimum of 6 Design Points are required.
- No points with a repeated combination of Altitude, Mach and AoA should be used.
- Each combination of Altitude and Mach requires at least 2 AoA points.
- At least 3 distinct non-colinear points on the Altitude + Mach vs. Cl plane are required.

The more advanced .aero file format type, called an `Advanced Fix Wing .aero` file, is not yet supported by Fluent Aero so it cannot be automatically generated using this command.

Refer to the STK Help at https://help.agi.com/stk/ for more information on how a `.aero` file can be used in STK Aviator.

- **Update Results**

  If a user has changed a workflow setting that effects some of the tabulated results or has added a new Custom Output Parameter, **Update Results** can be used to update all the results inside these tables without re-calculating each Design Point. For example, if a user would like to change the Reference length of a group of Design Point solutions that have already been obtained in order to update the aerodynamic coefficients, the user can simply change the **Reference Length** inside **Geometric Properties** and use **Update Results** to update these coefficients such that they are computed with the new value of **Reference Length**.

  ---

  **Note:**

  Clicking **Update Results** after adding a new **Custom Output Parameter** will cycle through each Design Point and load each solution file, so this process could take some time to complete, especially for larger cases.

  ---

There are four tables produced by Fluent Aero to help post-process the Results: **Table:Summary**, **Table: Coefficients**, **Table:Residuals**, and if required, **Table:Custom Output**.

- **Table: Summary**

  An example of Table: Summary is provided below along with a description of each item in the table.

Table:Summary

| | DP | Mach | AoA [deg] | AoS [deg] | P [Pa] | T [K] | Reynolds | Avg Coeff Conv | Avg Residual | Criteria Met? |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0.43 | 2.0 | 0.0 | 54019.9036 | 255.65 | 2.1340e+07 | 3.5523e-04 | 1.2988e-04 | partially |
| 2 | 2 | 0.43 | 4.0 | 0.0 | 54019.9036 | 255.65 | 2.1340e+07 | 1.9492e-04 | 1.0765e-04 | partially |
| 3 | 3 | 0.43 | 6.0 | 0.0 | 54019.9036 | 255.65 | 2.1340e+07 | 8.3860e-05 | 1.6652e-04 | partially |
| 4 | 4 | 0.43 | 8.0 | 0.0 | 54019.9036 | 255.65 | 2.1340e+07 | 3.0434e-03 | 1.6267e-04 | no |
| 5 | 5 | 0.43 | 10.0 | 0.0 | 54019.9036 | 255.65 | 2.1340e+07 | 1.6753e-02 | 2.1401e-04 | no |

  - **DP**

    The Design Point number associated with the row in the table.

  - **Mach**

The Mach Number used to calculate the Design Point.

– **AOA [deg.]**

The Angle of Attack used to calculate the Design Point.

– **P [Pa]**

The Static Pressure used to calculate the Design Point.

– **T [K]**

The Static Temperature used to calculate the Design Point.

– **Reynolds**

The Reynolds number associated with Design Point flight conditions.

– **Avg. Coeff. Conv.**

This value is the average of the convergence of the lift coefficient, drag coefficient and moment coefficient at the final iteration computed for the design point.

$$Avg.\ Coeff\ Conv. = \left(Cl\ Conv. + Cd\ Conv.\right) / 2$$

– **Avg. Residual**

This value is the average of all residuals (from Table: Residuals) at the final iteration computed for the design point.

– **Criteria Met?**

This value will list yes if the Cl and Cd **Coefficients Convergence (Cl. Conv. and Cd. Conv.)** (from **Table: Coefficients**) and all **Residuals** (from **Table: Residuals**) meet their respective convergence conditions specified in **Solve** → **Advanced Settings** → **Aero Convergence Cutoff** (default 1e-4) and **Aero Coeff. Convergence Cutoff** (default 2e-4). It will list no if these conditions have not been met, and partially if some of the conditions have been met.

---

**Note:**

The convergence of the Cm-y, Cm-p, and Cm-r values (partially reported in the **Max. Cm Conv.** column) are not used to determine if the convergence criteria is met. If interested, a user should investigate these values manually with the **Max. Cm Conv.** value in the table or using the **Convergence** graphs.

---

- **Table: Coefficients**

An example of **Table: Coefficients** is provided below along with a description of each item in the table.

Table:Coefficients

| | DP | Cl | Cd | Cm-y | Cm-p | Cm-r | Cl Conv. | Cd Conv. | Max. Cm Conv. |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1.7018e-01 | 2.4693e-02 | -1.1206e-01 | -3.0087e-02 | 3.5157e-01 | 4.7937e-04 | 6.2610e-04 | 5.7888e-04 |
| 2 | 2 | 2.6923e-01 | 3.0073e-02 | -1.1431e-01 | -2.8379e-02 | 5.8325e-01 | 3.4108e-04 | 2.5861e-04 | 2.9511e-04 |
| 3 | 3 | 3.6078e-01 | 3.7741e-02 | -1.0935e-01 | -2.7211e-02 | 7.8905e-01 | 8.6636e-05 | 1.9238e-04 | 7.3440e-05 |
| 4 | 4 | 4.3799e-01 | 4.6759e-02 | -9.9970e-02 | -2.5288e-02 | 9.5277e-01 | 6.2536e-03 | 4.1073e-03 | 4.0943e-03 |
| 5 | 5 | 4.7379e-01 | 6.5784e-02 | -1.0813e-01 | -2.1888e-02 | 1.0040e+00 | 3.4296e-02 | 1.1700e-02 | 2.9746e-02 |

– **DP**

The Design Point number associated with the row in the table.

– **Cl**

The Lift Coefficient computed for the Design Point. This coefficient considers the pressure and shear forces on all walls in the geometry.

– **Cd**

The Drag Coefficient computed for the Design Point. This coefficient takes into account the pressure and shear forces on all walls in the geometry.

– **Cm-y**

The Yaw Moment Coefficient computed for the Design Point. This coefficient considers the pressure and shear forces on all walls in the geometry.

– **Cm-p**

The Pitch Moment Coefficient computed for the Design Point. This coefficient takes into account the pressure and shear forces on all walls in the geometry.

– **Cm-r**

The Roll Moment Coefficient computed for the Design Point. This coefficient considers the pressure and shear forces on all walls in the geometry.

– **Cl Conv.**

A value measuring the convergence of the lift coefficient at the final iteration computed in the simulation of the design point. This value is computed using the following equation,

$$Cl\ \ Conv. = abs\left(Cl_n - Cl_i\right) / Cl_n$$

where $Cl_n$ is the lift coefficient at the final iteration, and $Cl_i$ is the lift coefficient that produces the maximum difference when compared to $Cl_n$ from among the most recent previous iterations. By default, the 10 most recent iterations are used, and this can be specified in **Solve** → **Advanced Settings** → **Aero Conv. Coeff Previous Values**.

– **Cd Conv.**

A value measuring the convergence of the drag coefficient at the final iteration computed in the simulation of the design point. This value is computed using the following equation,

$$Cd\ \ Conv.{=}abs\left(Cd_n{-}Cd_i\right)/Cd_n$$

where $Cd_n$ is the drag coefficient at the final iteration, and $Cd_i$ is the drag coefficient that produces the maximum difference when compared to $Cd_n$ from among the most recent previous iterations. By default, the 10 most recent iterations are used, and this can be specified in **Solve** → **Advanced Settings** → **Aero Conv. Coeff Previous Values**.

– **Max Cm Conv.**

A value measuring the convergence of the moment coefficients at the final iteration in the simulation of the design point. This value is computed using the following equation,

$$Max\ Cm\ Conv. = max(\ \left[abs\left(Cmy{-}n - Cmy{-}i\right)\ /\ Cmy{-}n\right], \left[abs\left(Cmp{-}n - Cmp{-}i\right)\ /\ Cmp{-}n\right], \left[abs\left(Cmr{-}n\right.\right.$$

Where each $\mathrm{Cm}_n$ (Cmy, Cmp, Cmr) is each moment coefficient at the final iteration, and each $\mathrm{Cm}_i$ is the moment coefficient that produces the maximum difference when compared to each $\mathrm{Cm}_n$ from among the most recent previous iterations. Therefore, the **Max Cm Conv.** value in the table will show the moment coefficient with the highest value. By default, the 10 most recent iterations are used, and this can be specified in **Solve** → **Show Advanced Settings** → **Aero Conv. Coeff Previous Values**.

- **Table: Forces**

An example of **Table:Coefficients** is provided below along with a description of each item in the table.

Table:Forces

| | DP | Lift [N] | Drag [N] | Mom. Yaw [N.m] | Mom. Pitch [N.m] | Mom. Roll [N.m] |
|---|---|---|---|---|---|---|
| 1 | 1 | 4.1858e+05 | 6.0735e+04 | -1.3506e+06 | -3.6262e+05 | 4.2372e+06 |
| 2 | 2 | 6.6220e+05 | 7.3970e+04 | -1.3777e+06 | -3.4203e+05 | 7.0294e+06 |
| 3 | 3 | 8.8739e+05 | 9.2831e+04 | -1.3179e+06 | -3.2795e+05 | 9.5098e+06 |
| 4 | 4 | 1.0773e+06 | 1.1501e+05 | -1.2049e+06 | -3.0478e+05 | 1.1483e+07 |
| 5 | 5 | 1.1653e+06 | 1.6180e+05 | -1.3032e+06 | -2.6380e+05 | 1.2101e+07 |

– **DP**

The Design Point number associated with the row in the table.

– **Lift [N]**

The total Lift force computed for the Design Point. This force takes into account the pressure and shear forces on all walls in the geometry.

– **Drag [N]**

The total Drag force computed for the Design Point. This force takes into account the pressure and shear forces on all walls in the geometry.

– **Mom. Yaw [N.m]**

– The total Yaw Moment force computed for the Design Point. This force takes into account the pressure and shear forces on all walls in the geometry.

– **Mom. Pitch [N.m]**

The total Pitch Moment force computed for the Design Point. This force takes into account the pressure and shear forces on all walls in the geometry.

– **Mom. Roll [N.m]**

The total Roll Moment force computed for the Design Point. This force takes into account the pressure and shear forces on all walls in the geometry.

- **Table: Residuals**

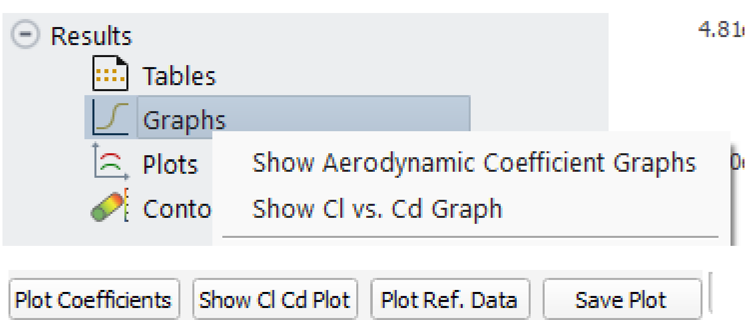
An example of **Table:Residuals** is shown below. The residual names listed may change depending on the models that are used in your simulation.

Table:Residuals

| | DP | iteration | continuity | x-velocity | y-velocity | z-velocity | energy | k | omega |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 500 | 7.1953e-07 | 3.1617e-07 | 5.0368e-07 | 7.0564e-06 | 7.0392e-06 | 1.7212e-05 | 3.4462e-06 |
| 2 | 2 | 65 | 2.2473e-06 | 1.1511e-05 | 1.8454e-05 | 6.4995e-06 | 1.0613e-05 | 3.4809e-05 | 4.5664e-06 |
| 3 | 3 | 77 | 1.8092e-06 | 1.0186e-05 | 8.5661e-06 | 1.7414e-05 | 6.5436e-06 | 2.3707e-05 | 4.9580e-06 |
| 4 | 4 | 1500 | 5.8639e-05 | 1.1425e-03 | 3.0240e-04 | 3.5305e-04 | 4.0992e-04 | 2.1731e-03 | 9.1129e-05 |

## 33.9.2.2. Graphs

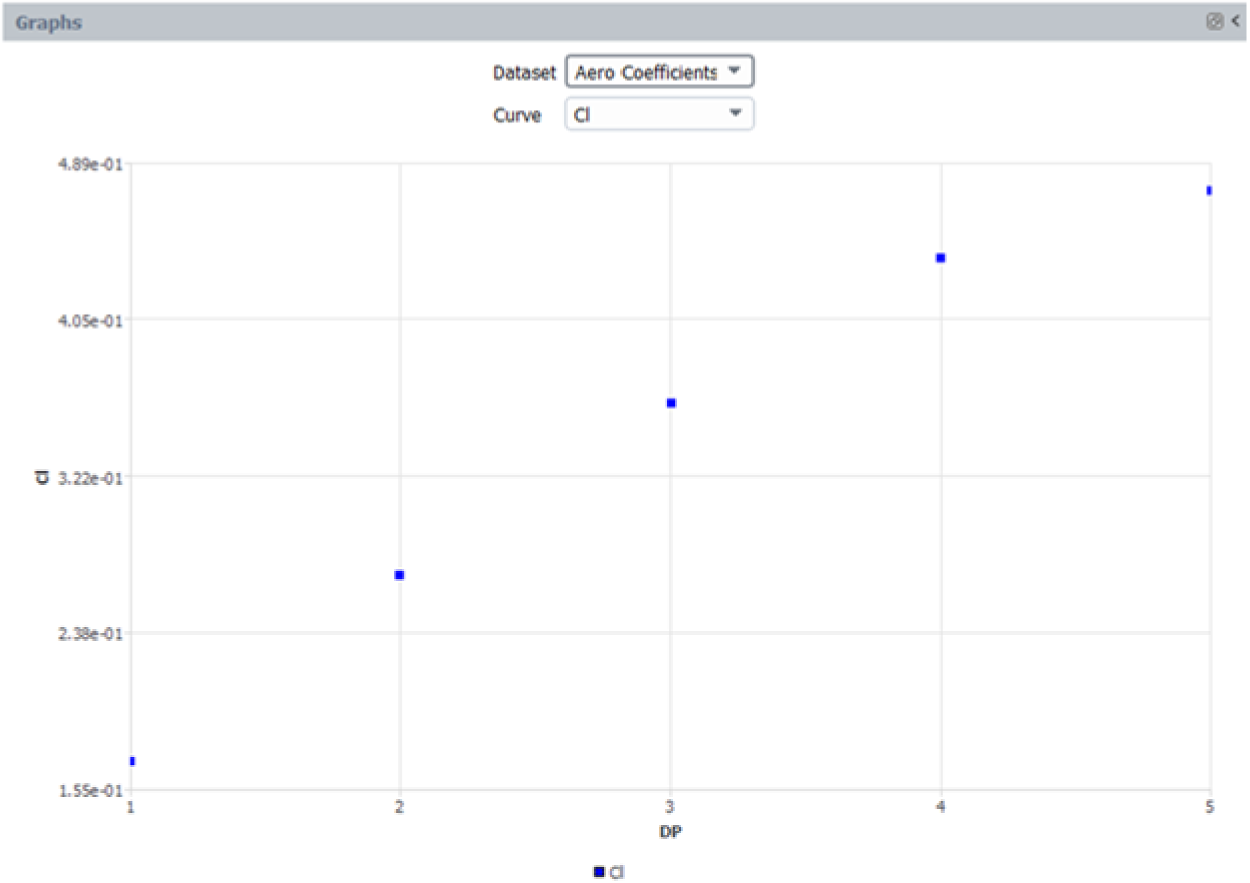
The **Graphs** component contains commands to display aerodynamic coefficient plots using the results of your simulations inside the Graphics window. These commands are accessible by right-clicking on **Graphs** in the **Outline View**, or as buttons located at the bottom of the **Graphs** properties window.




- **Plot Coefficients** / **Show Aerodynamic Coefficient Graphs**

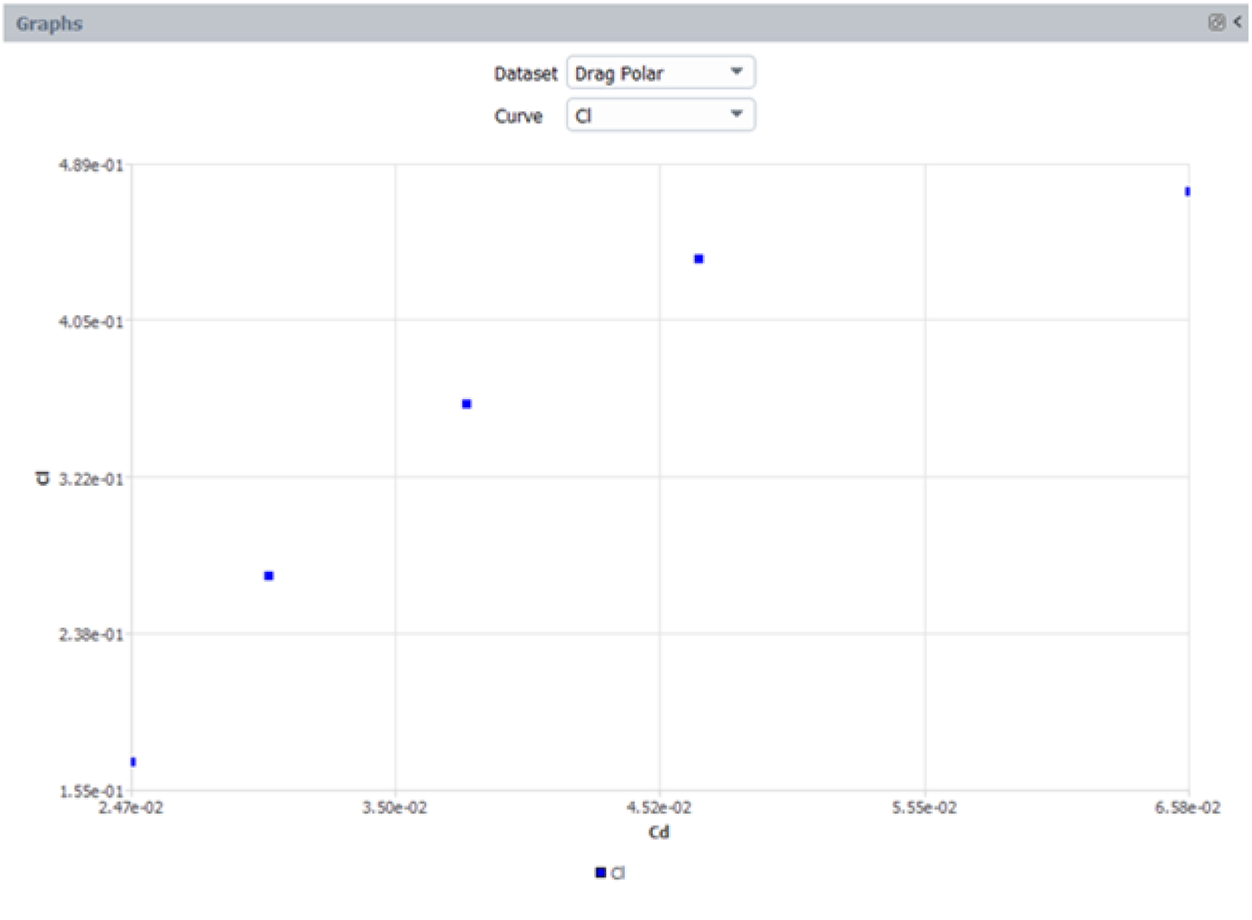
This function will cause the graphics window to present the **Graphs** window showing the aerodynamic coefficients with respect to the Angle of Attack, Mach number, or Design Point number for all Design Points that were calculated in the Simulation.

An example of **Graphs** panel showing a Coefficient graph is shown below. The y-axis definition can be modified by making a selection from the **Curve** selection box at the top of the graph. **Cl**, **Cd**, **Cm**, and **Cl/Cd** can be selected.




- **Show Cl vs. Cd Graph / Show Cl Cd Plot**

  This function will cause the graphics window to present the lift coefficient with respect to the drag coefficient for all Design Points that were calculated in the Simulation. An example of **Graphs** panel showing a Drag Polar, **Cl** vs. **Cd** graph is shown below.

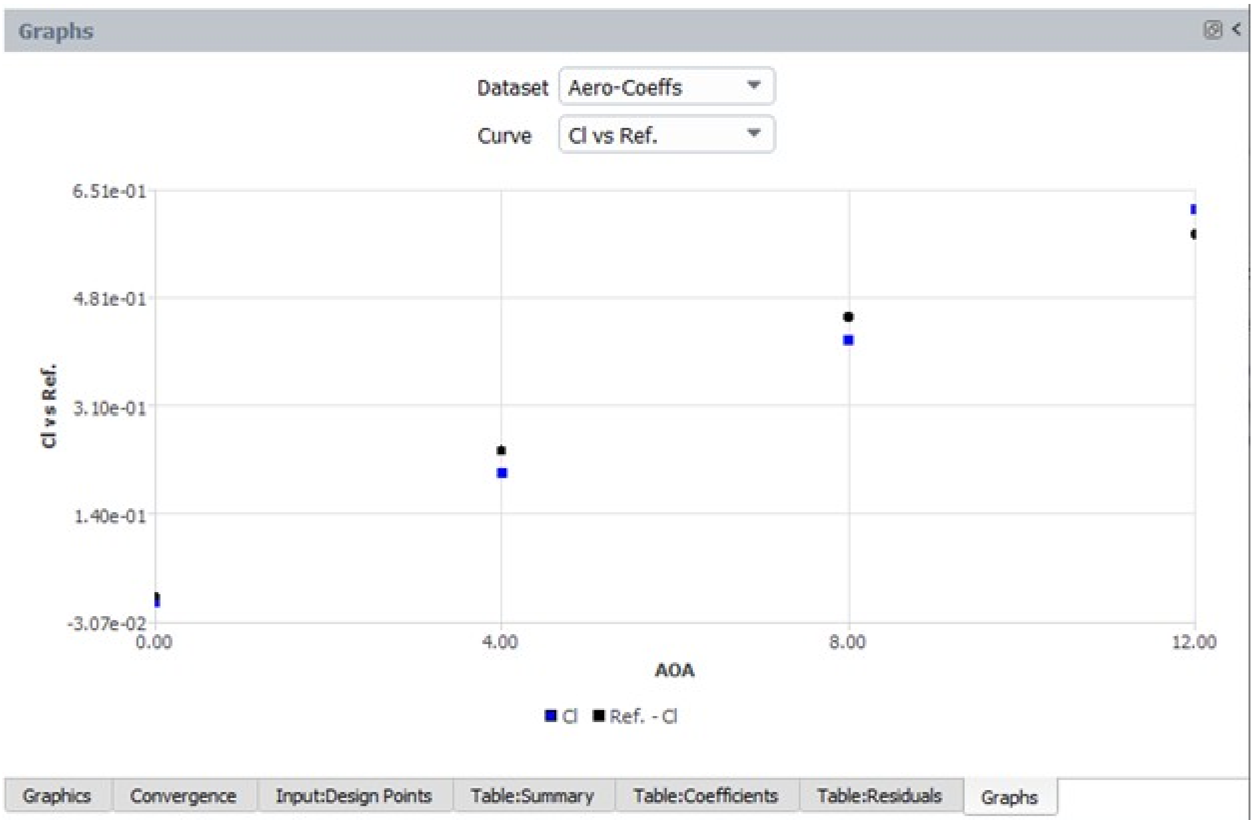



- **Plot Ref. Data**

  This function will open a file navigation window where a comma separated value (`.csv`) file containing reference data can be selected. This file will be then read and its content added to the active graph panel.

  The `.csv` file must be formatted such that the first line contains the x-axis value (either AOA, Mach or DP) and the y-axis value (either Cl, Cd, Cm or Cl/Cd) separated by a comma. The remaining lines of the file can contain the reference data points. An example `.csv` reference data file and associated **Graphs** panel is shown below.

```
AOA,Cl
0,0.1e-01
4.0,0.24
8,0.45
12,0.58
```




- **Save Plot**

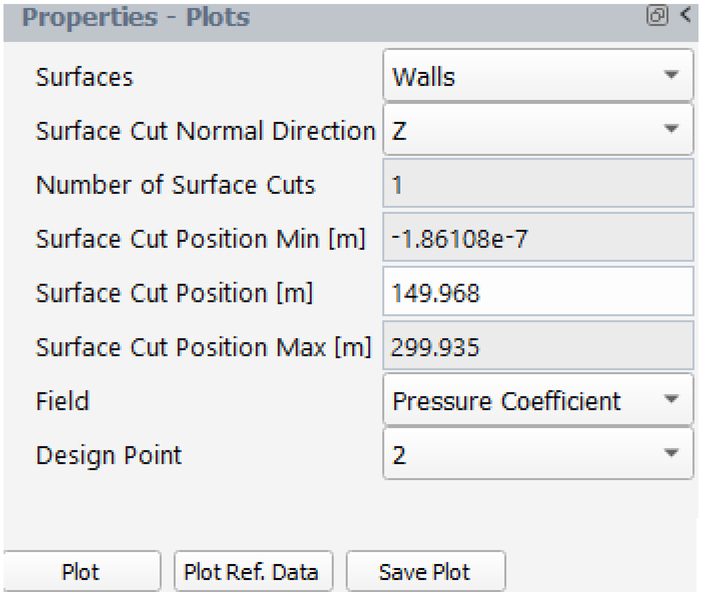
  This function will save the content of the **Graphs** panel to a `.png` image file on the disk.

### 33.9.2.3. Plots

The **Plots** component contains commands to display 2D surface cut plots of results of a Simulation in the Graphics window.

By default, the plane used to create the surface cut plots will be oriented coplanar with the Lift and Drag Direction Axis, as setup in the Grid Properties step.

Clicking on the **Plots** item will cause the **Plots** properties window to appear.




- **Surfaces**

  Defines the surface to use on which to display the surface cut 2D plot values. Choosing **Walls** plots on all walls of the domain. Choosing **Selected Surfaces** will reveal another option where individual surfaces can be selected to use for the plot.

- **Surface Cut Normal Direction**

  Set the normal direction of the cut plane surface.

- **Number of Surface Cuts**

  Cuts Currently, only **1** surface cut plot can be created at a time, so this item is not editable.

- **Surface Cut Position Min [m]**

  This value specifies the minimum global grid position, in meters, available in the direction normal to the plane that can be used to create the surface cut plot.

- **Surface Cut Position [m]**

  Specify the global grid position, in meters, where a surface cut plot will be plotted.

- **Surface Cut Position Max [m]**

  This value specifies the maximum global grid position, in meters, available in the direction normal to the plane that can be used to create the surface cut plot.

- **Field**

  Specify the field value to plot.

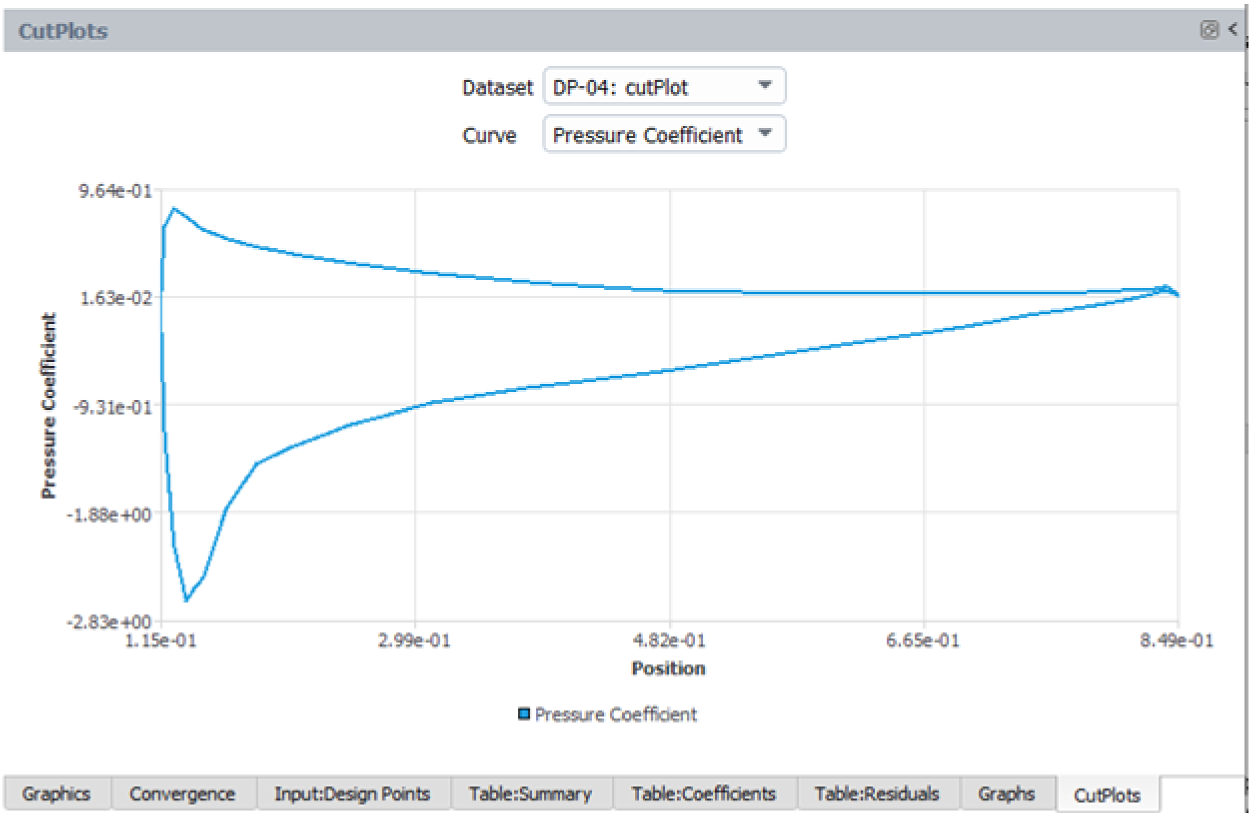
- **Design Point**

Specify the Design Point solution to use for the plot. If this value is changed to a different design point solution than is currently loaded, the solution will be first loaded when **Show Cut Plot** is selected.

- **Plot**

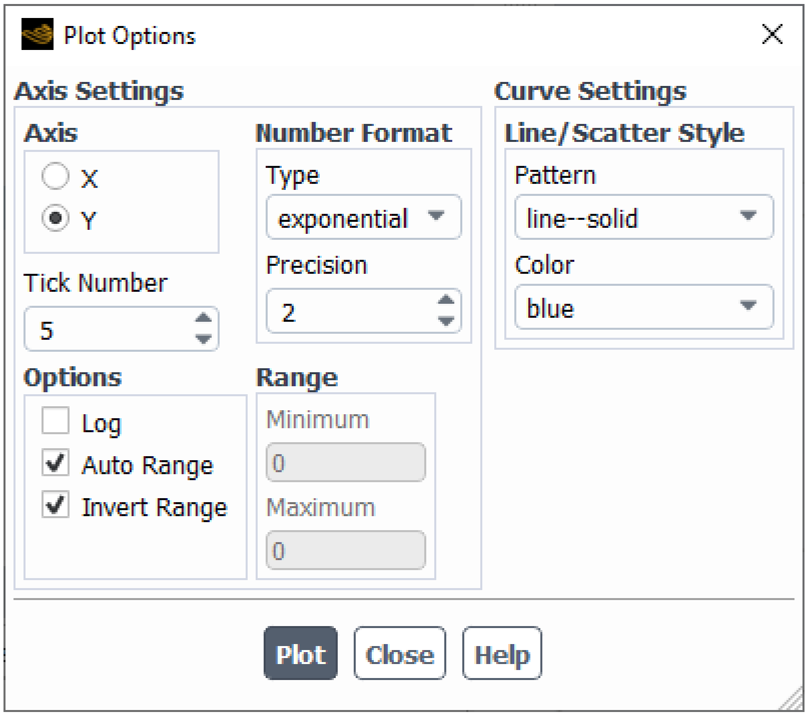
This command will create the 2D surface cut plot in the Graphics window. If the Design Point number listed in the selection box is different than the solution that is currently loaded, the appropriate solution will be loaded before the plot is created.

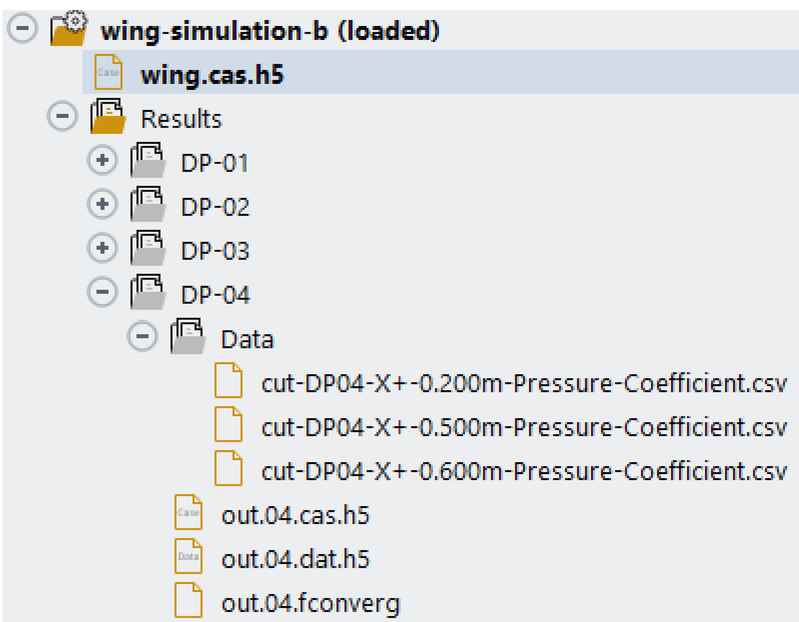
An example **Plot** is shown below.




When **Plot** is used, a comma separated value file is also created with the x-y data used to generate the plot. This file will be saved to a **Data** folder inside the Design Point folder associated with the solution used to generate the plot.

Also, a **Plot Options** window will appear which allows the user to modify the line and the point plot display format to use for the plot. Modify any setting and click the **Plot** button in the **Plot Options** window to update the plot format.



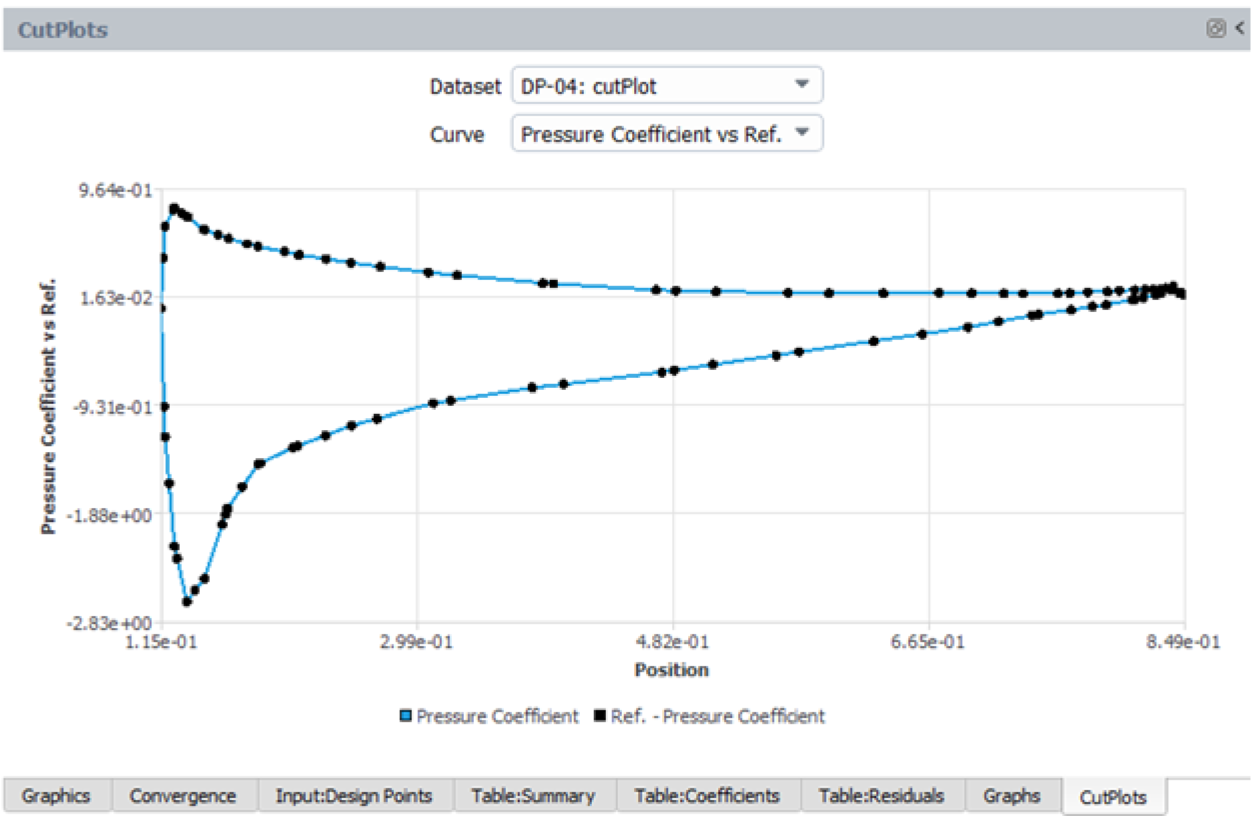




- **Plot Ref. Data**

  This function will open a file navigation window where a comma separated value (`.csv`) file containing reference data can be selected. The reference data will then be added to the active and representative graph panel.

  The `.csv` file must be formatted such that the first line contains **Position** then the y-axis value (use the same name as it is listed in the y-axis of the plot) separated by a comma. The remaining lines of the file can contain the reference datapoints. An example of `.csv` reference data file and associated **Graphs** panel is shown below.

```
Position,Pressure Coefficient
0.483831,0.0674613
0.469754,0.0738452
0.396474,0.129155
0.388995,0.134842
0.327676,0.202651
0.307008,0.22573
0.272619,0.278544
0.251476,0.311725
0.233691,0.346139
0.214212,0.382598
0.204001,0.408341
0.184946,0.453359
0.177516,0.477416
0.163748,0.523822
0.156577,0.55808
```




- **Save Plot**

  This function will save the content of the **Graphs** panel to a `.png` image file on the disk.

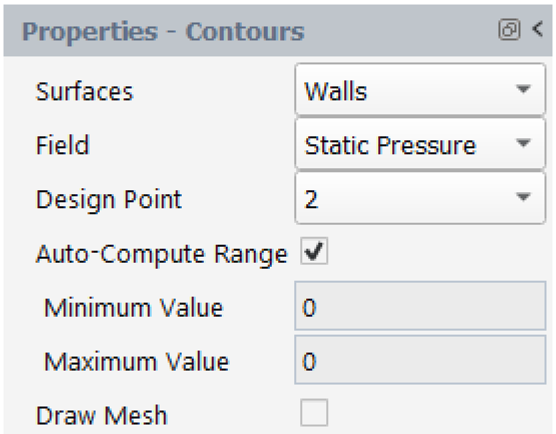
## 33.9.2.4. Contours

The **Contours** component contains commands to display 3D contour plots of results of a Simulation in the Graphics window.

Clicking on the **Contours** item will cause **Properties - Contours** to appear. The first setting in the **Properties - Contours** is **Surfaces**, which has two possible options: **Walls**, and **Cutting Plane**. The settings and functions available in the **Contour** category will change depending on which of these options is selected.

- **Surfaces – Walls**

  When **Surfaces** is set to **Walls** in **Properties - Contours**, the following options are available:




  – **Contours: Surfaces**

    This option specifies the **Surfaces** to use for the 3D contour plots. In this section, **Walls** is selected. The **Walls** option will cause the contour to be plotted on all wall boundaries in the domain.

  – **Field**

    This option specifies the solution field that will be plotted. The following solution fields are available when **Surfaces** is set to **Walls: Pressure**, **Temperature**, **Heat Flux**, **Shear Stress**, **Y Plus**, and **Shear Stress Vectors**.

  – **Design Point**

    Specify the **Design Point** solution to use for the contour. If this value is changed to a different design point solution than is currently loaded, the solution will be first loaded when **Plot Wall Contour** is selected.

  – **Auto-Compute Range**

    Enabling this option will cause the contour to automatically compute the min and max range for the contour plot based on the solution that is shown.

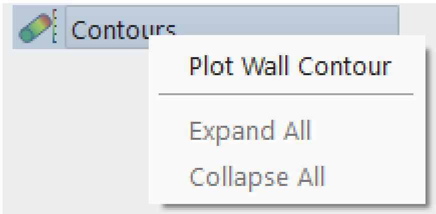
  – **Minimum Value**, **Maximum Value**

    Defines the minimum and maximum range for the contour to display. These settings are automatically computed if **Auto-Compute Range** is enabled.

  – **Draw Mesh**

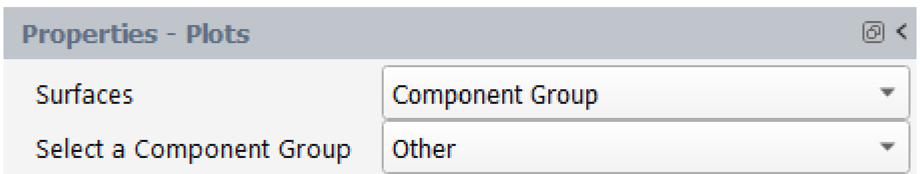
    Enabling this option will cause the mesh lines to be drawn on top of the contour plot.

Subsequently, the following command are available by right-clicking on **Contours** or at the bottom of the **Contours** properties window:



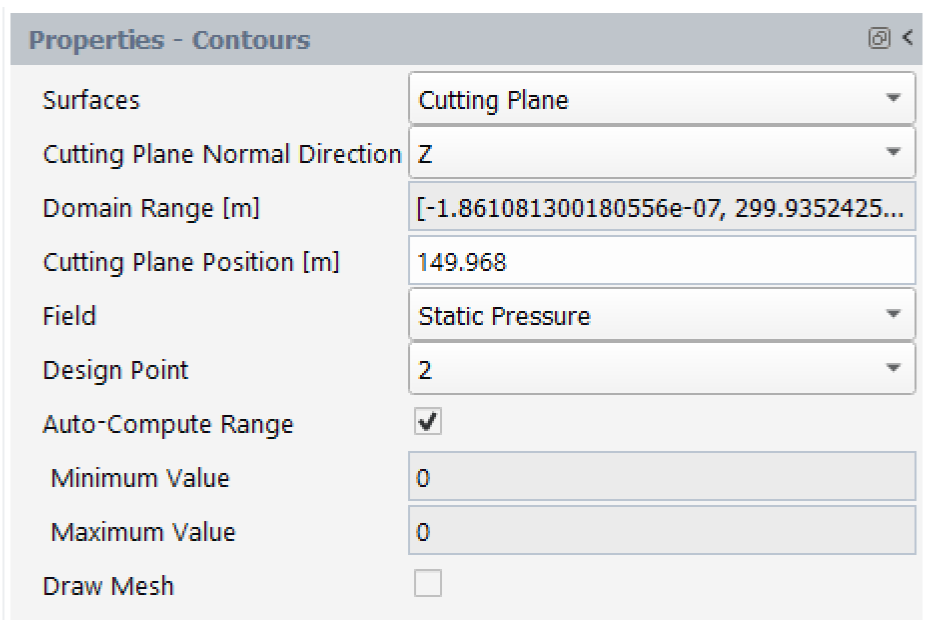

- **Surfaces – Component Group**

  When **Surfaces** is set to **Component Group** in **Properties- Contours**, one extra option is available where the user can **Select a Component Group** to be displayed:




- **Contours: Surfaces – Cutting Plane**

  When **Surfaces** is set to **Cutting Plane** in **Properties - Contours**, the following options are available:




  - **Surfaces**

    This option specifies the Surfaces to use for the 3D contour plots. In this section, **Cutting Plane** is selected. The **Cutting Plane** option will cause the contour to be plotted on a cutting plane in the Graphics window. This cutting plane will be coplanar to the Lift and Drag direction Vectors setup in **Set Grid Properties**.

– **Cutting Plane Position Min [m]**

  This value tells the user the minimum grid position the cutting plane can be set to.

– **Cutting Plane Position [m]**

  This option is used to specify the grid position of the cutting plane that will be used for contour plotting.

– **Cutting Plane Position Max [m]**

  This value tells the user the maximum grid position the cutting plane can be set to.

– **Field**

  This option specifies the solution field that will be plotted. The following solution fields are available when **Surfaces** is set to **Cutting Plane: Pressure**, **Temperature**, **Velocity Magnitude**, **Mach Number**, **Turbulent Intensity**, **Turbulent Viscosity**, and **Velocity Vectors**.

– **Design Point**

  Specify the **Design Point** solution to use for the contour. If this value is changed to a different design point solution than is currently loaded, the solution will be first loaded when **Plot Cutting Plane Contour** is selected.

– **Auto-Compute Range**

  Enabling this option will cause the contour to automatically compute the min and max range for the contour plot based on the solution that is shown.

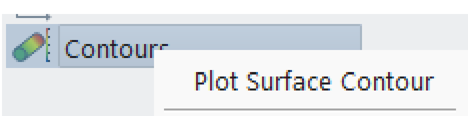
– **Minimum Value**, **Maximum Value**

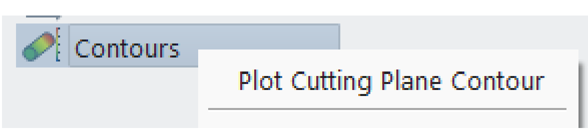
  Defines the minimum and maximum range for the contour to display. These settings are automatically computed if **Auto-Compute Range** is enabled.

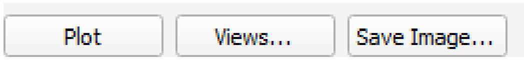
– **Draw Mesh**

  Enabling this option will cause the mesh lines to be drawn on top of the contour plot.

The following **Contours** command options are available as command buttons or as right-click commands:






– **Plot, Plot Surface Contour, Plot Cutting Plane Contour**

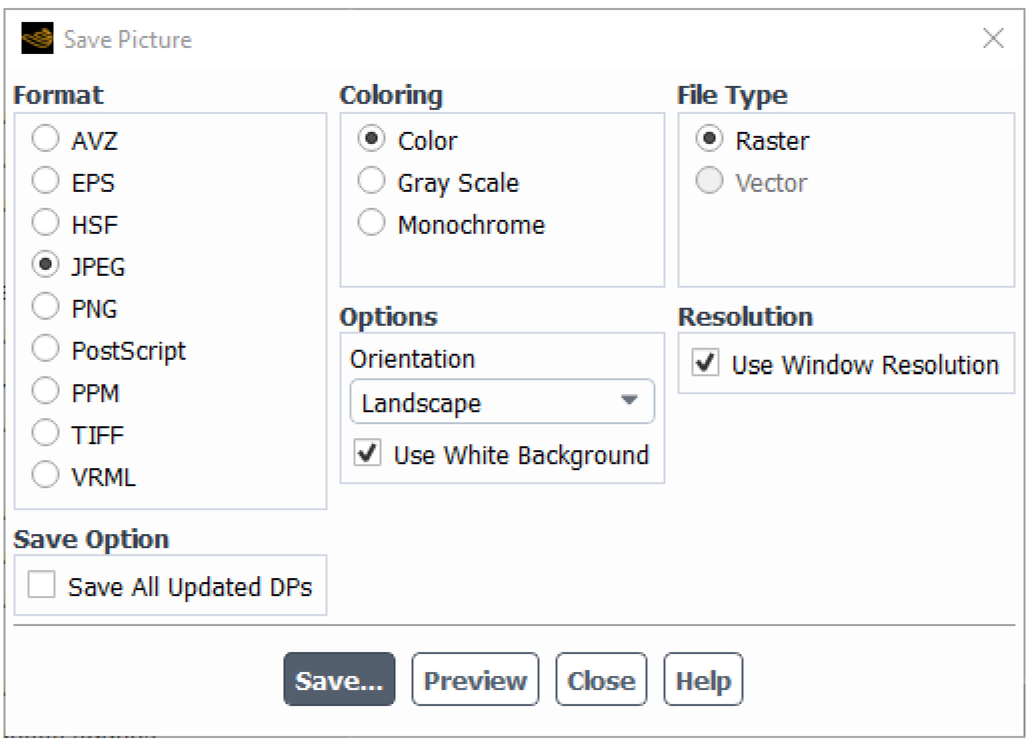
These commands will plot the contour requested as setup in **Properties - Contours**. If the Design Point number listed in the selection box is different than the solution that is currently loaded, the appropriate solution will be loaded before the contour is created.

– **Views...**

Opens a panel where a graphics window view orientation can be selected, recalled, or saved.

– **Save Image...**

Opens a panel where the user can save the content of the graphics window into a picture file. The panel includes several format options that can be used to generate the picture. It also includes a **Save All Updated DPs** check box, which when enabled, will cause Fluent Aero to output a picture file using the same contour plot setup for every Design Point that has a **Status** of **Updated**.

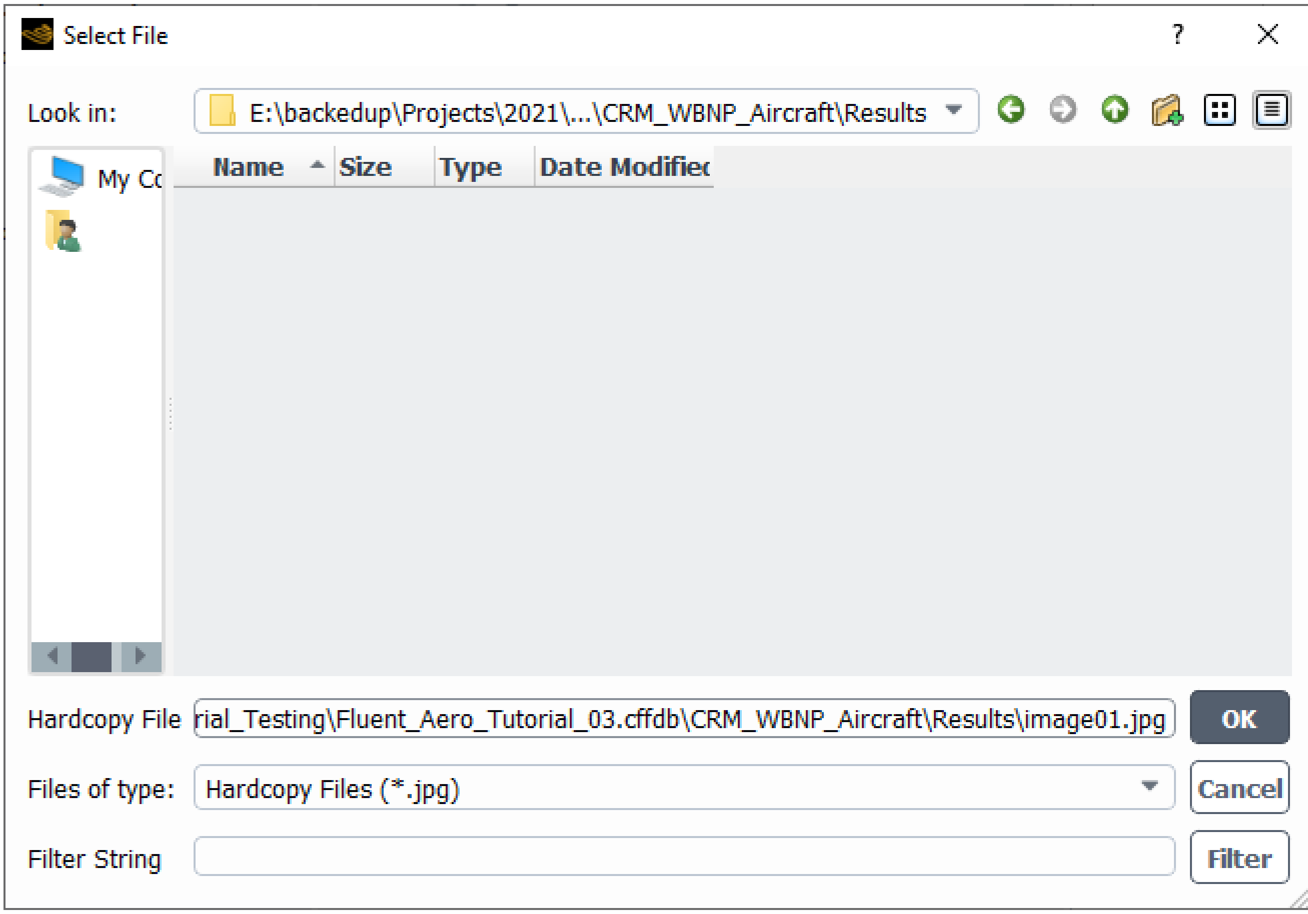



After clicking the **Save...** button, a window will appear.

---

**Note:**

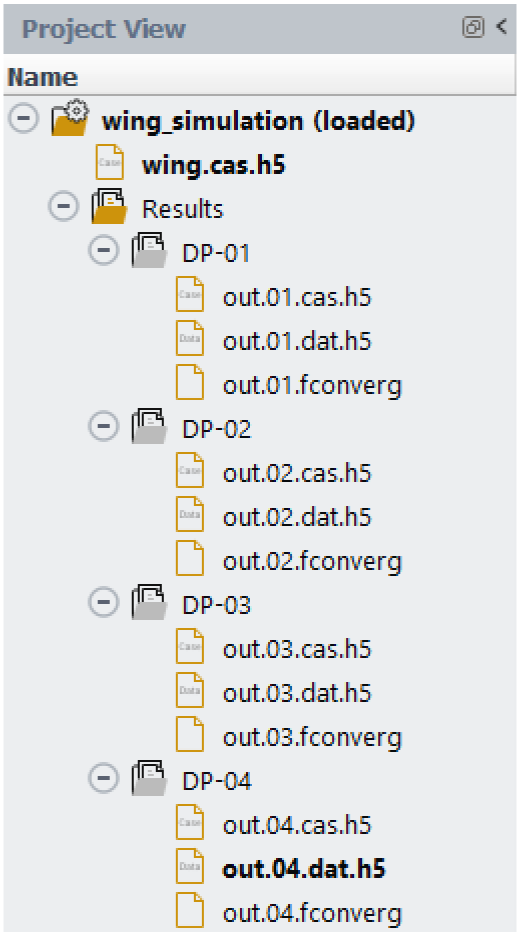
You must enter the full file path into **Hardcopy File** to the image they would like to output. Clicking **OK** will save the image to the desired location.




### 33.9.3. Results Folder

Once a Fluent Aero Simulation is complete, clicking on the **Project View** will now display a **Results** folder inside the **Simulation** folder. The **Results** folder will contain subfolders corresponding to each simulated Design Point, and each of these subfolders will contain associated case (`.cas[.h5]`), data (`.dat[.h5]`) and convergence (`.fconverg`) files. The `.fconverg` file contains the output of the residuals and monitors and is used to create the **Convergence** plots.

The image below shows an example of a **Results** folder.

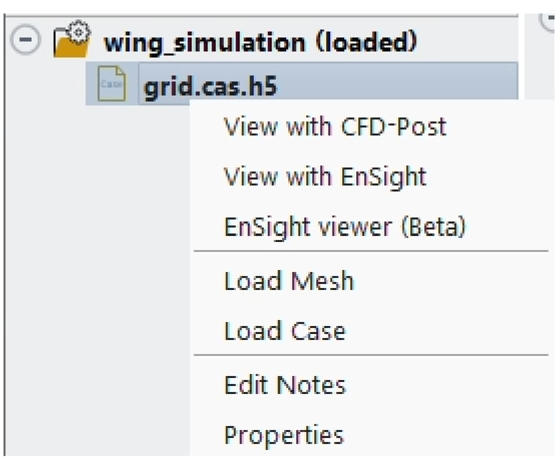



## 33.10. Using the Project View to Interact with Fluent Aero Simulations

The following section describes the options available to interact with your Simulations, Runs and results files from the **Project View**.

### 33.10.1. Simulation Folder Commands

When a Simulation is not yet loaded, the following options are available:



- **Load in Solver**

  Loads the Simulation. This will launch the Solver session in the background. (See Loading a Simulation (p. 457) for more information).

- **Delete**

  Deletes the Simulation folder. This will remove the Simulation from the **Project View** and delete all child files and folders from the disk.

- **Clean-up folder**

  Erases all simulation files such as **cleanup-***, and **serverinfo-***.

- **Open in file explorer**

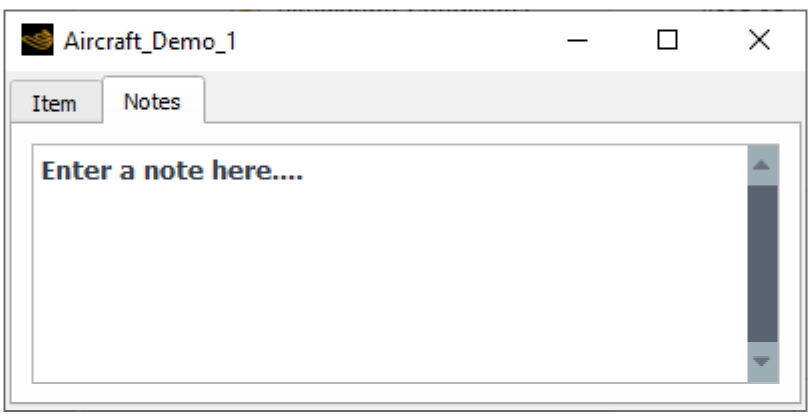
  Opens the simulation directory in the file explorer of the local operating system.

- **Sort by name**

  Sorts all files in the Simulation alphabetically by name. By default, the Simulation files are initially sorted by time of creation.

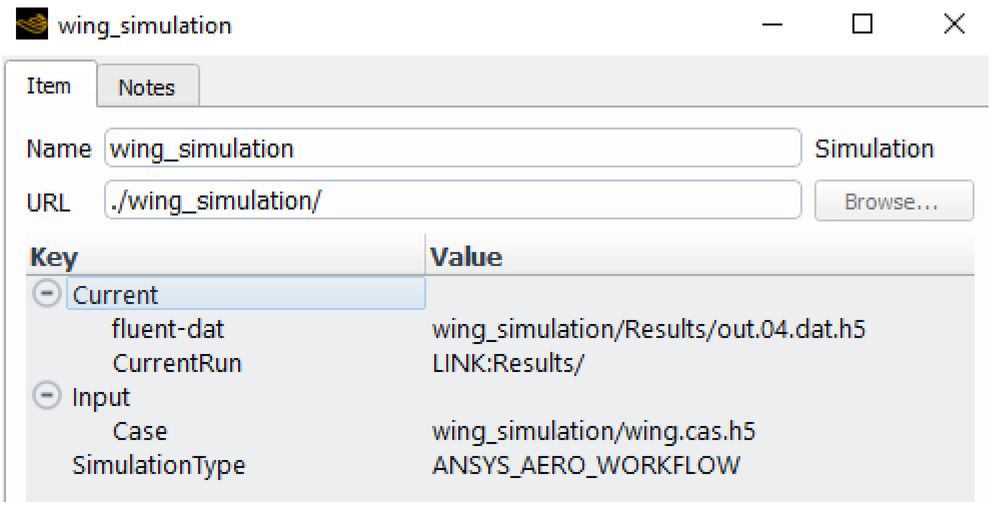
- **Edit Notes**

  Opens a **Properties** window, showing the **Notes** panel, where a user can add text notes to the selection. If a text note is added, the item inside the project view will be displayed with a * icon next to its name, signifying a note has been added.




- **Properties**

  Opens a **Properties** window showing the Metadata and properties of a Simulation. The figure below shows an example of the contents of a Simulation Properties window.



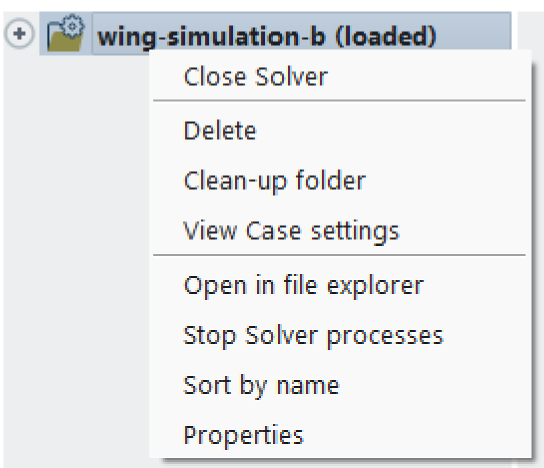

In the above figure,

– The **Name** of the Simulation and its **URL** path are shown at the top of the **Item** tab.

– **Input** shows the input case file used in the Simulation.

– The **Notes** tab can be used to write any notes that you would like to add to better describe this Simulation.

When a Simulation is loaded, the following additional options are available:




- **Close Solver**

  Closes the Simulation and closes the background Solver session if enabled.

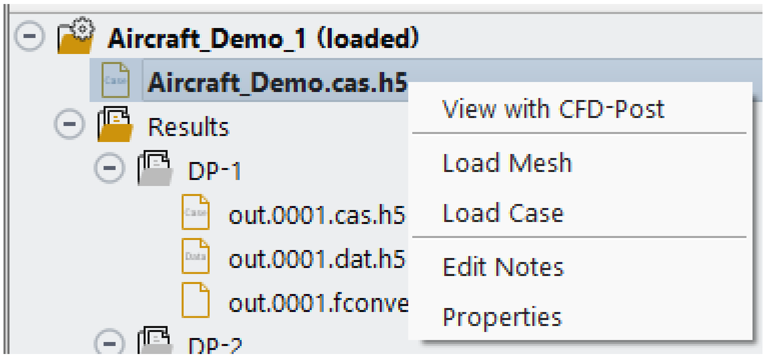
- **View Case settings**

  Shows the **Outline View** window associated with the Simulation, where the Simulation settings can be investigated.

- **Stop Solver processes**

Launches the **cleanup-fluent-*** script file associated with the simulation, which will close the associated background Solver session.

### 33.10.2. Case File Commands




- **View with CFD-Post**

  Opens a CFD-Post postprocessing session in a new window and loads the selected case file.

- **Load Mesh**

  Load the mesh from the selected `.cas[.h5]` file.

- **Load Case**

  Loads the selected `.case[.h5]` file.

- **Delete**

  Deletes the selected case file.

---

**Caution:**

If a user deletes the case file that is used as the input case for the Simulation, the Simulation can no longer be loaded.

---

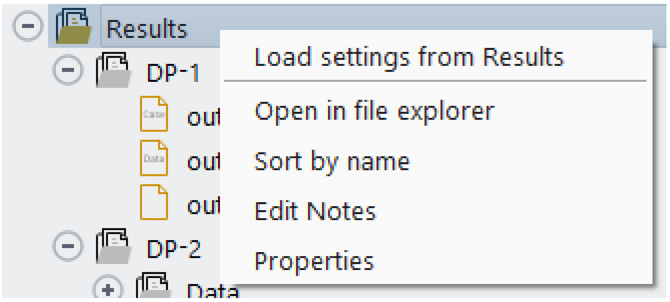
- **Edit Notes**

  Opens a **Properties** window showing the **Notes** panel, where a user can add text notes to the selection. If a text note is added, the item inside the project view will be displayed with an * icon next to its name, signifying that a note has been added.

- **Properties**

  Opens a **Properties** window showing the Metadata and properties of the case file.

### 33.10.3. Results Folder Commands and Metadata




- **Load settings from Results**

  Loads the active settings of the Aero Workflow steps from the **Results** folder **Metadata** (**Settings** → **Results**, **Setup**, **Solution**). These settings are generally those that are consistent with the most recent settings that were active when **Calculate** was last used.

- **Open in file explorer**

  Opens the **Results** directory in the file explorer of the local operating system.

- **Sort by name**

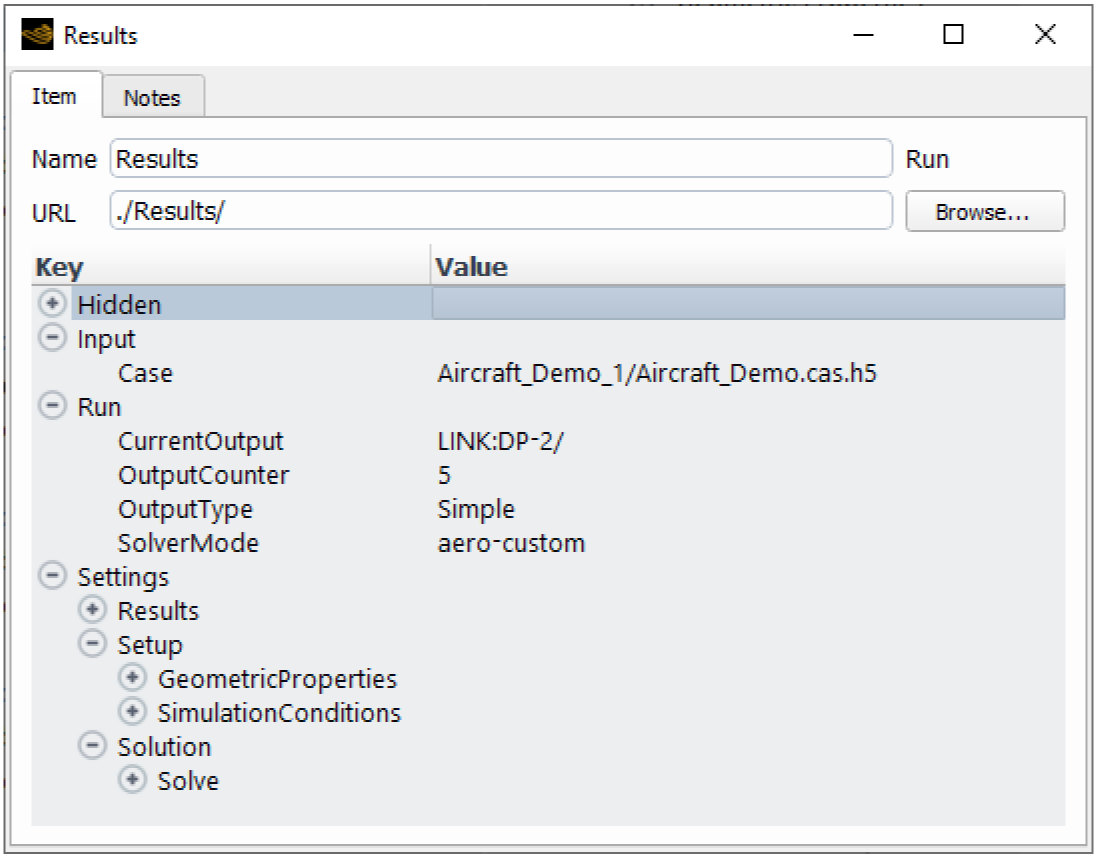
  Sorts all files in the **Results** folder alphabetically by name. By default, the Simulation files are sorted by time of creation.

- **Edit Notes**

  Opens a **Properties** window showing the **Notes** panel, where a user can add text notes to the selection. If a text note is added, the item inside the project view will be displayed with an * icon next to its name, signifying that a note has been added.

- **Properties**

  Opens a properties window showing the Metadata and properties of the **Results**. The figure below shows an example of the contents of a **Results** properties window.




- **Hidden** contains flags used by the datamodel to know when to expose certain elements of the user interface. This information should not generally be required by the user.

- **Input** contains metadata that specifies the location of the input `.cas[.h5]` file used to generate the **Results** folder.

- **Run** contains metadata that specifies the number of Design Points contained within the **Results** folder, and the **CurrentOutput** directory, related to the most recent Design Point that was calculated.

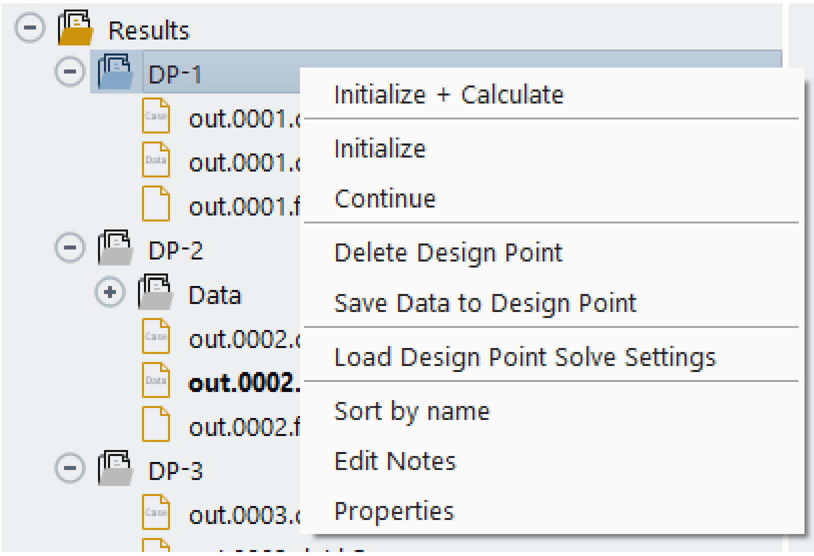
- **Settings** → **Setup** → **Geometric Properties** contains metadata that stores the settings used in the **Geometric Properties** step when the **Results** were calculated. This metadata is used to restore the settings when loading a simulation.

- **Settings** → **Setup** → **Simulation Conditions** contains metadata that stores the settings used in the Simulation Conditions step when the **Results** were calculated. This metadata is used to restore the settings when loading a simulation.

- **Settings** → **Solution** → **Solve** contains metadata that stores the settings used in the Solve step when the **Results** were calculated. This metadata is used to restore the settings when loading a simulation.

- **Settings** → **Results** contains metadata associated with options within the **Results** step in the **Outline View**.

### 33.10.4. Design Point Folder Commands and Metadata




- **Initialize + Calculate**

  Initializes and then calculates the selected Design Point. This is equivalent to using the Solve → Calculate command from the **Outline View** with the respective Design Point Status in the **Input:Design Points** table set to **Needs Update**. This option will execute a fresh start of your calculation. Any solution file that is already present will not be loaded – it will be deleted and will not be used as an initial solution. The calculation will take into account any changes in settings that were made to the Aero Workflow steps in the **Outline View**.

- **Initialize**

  Initializes the selected Design Point. This is equivalent to using the Solve → Calculate command from the **Outline View** with the respective Design Point Status in the **Input:DesignPoints Table** set to **Initialize**. Any solution file that is already present will not be loaded – it will be deleted and will not be used as an initial solution. The initialization will take into account any changes in settings that were made to the Aero Workflow steps in the **Outline View.**

- **Continue**

  Calculates the selected Design Point. This is equivalent to using the Solve → **Calculate** command from the **Outline View** with the respective Design Point Status in the **Input:Design Points** table set to **Continue to Update**. This calculation will first load any previous solution file and continue the calculation from this previous solution. The calculation will take into account any changes in settings that were made to the Aero Workflow steps in the **Outline View**.

- **Delete Design Point**

  Deletes the selected Design Point folder along with all the files contained within it. The remaining Design Point Folder numbers and their files will be renamed to ensure that the project continues to have complete and sequential numbering. For example, if DP-2 is deleted, the previously named DP-3 folder and its content will be renamed to DP-2. This is equivalent to using the **Simulation Conditions** → **Delete Design Point** command from the **Outline View**.

- **Save Data to Design Point**

This command will save the current state of the solution in memory to the `.dat` file of the selected DP folder. This allows a user to perform custom operations in the Solution Workspace to prepare a data file, which can then subsequently save to a Design Point folder of choice. This may be helpful, for example, if a user would like to prepare a solution with a custom initialization routine, before using **Continue to Update** to continue the Design Point calculation from that state. This option is only recommended for advanced users of Fluent Aero.

- **Load Design Point Solve Settings**

  Load all **Properties - Solve** panel settings from the Design Point Metadata (**Settings** → **Solution** → **Solve**) and populates the Aero Workflow Solve step (including anything listed in **Show Advanced Settings**) with these settings. This is useful if you want to revert back to use **Solve** settings that were used for that particular Design Point.

- **Sort by name**

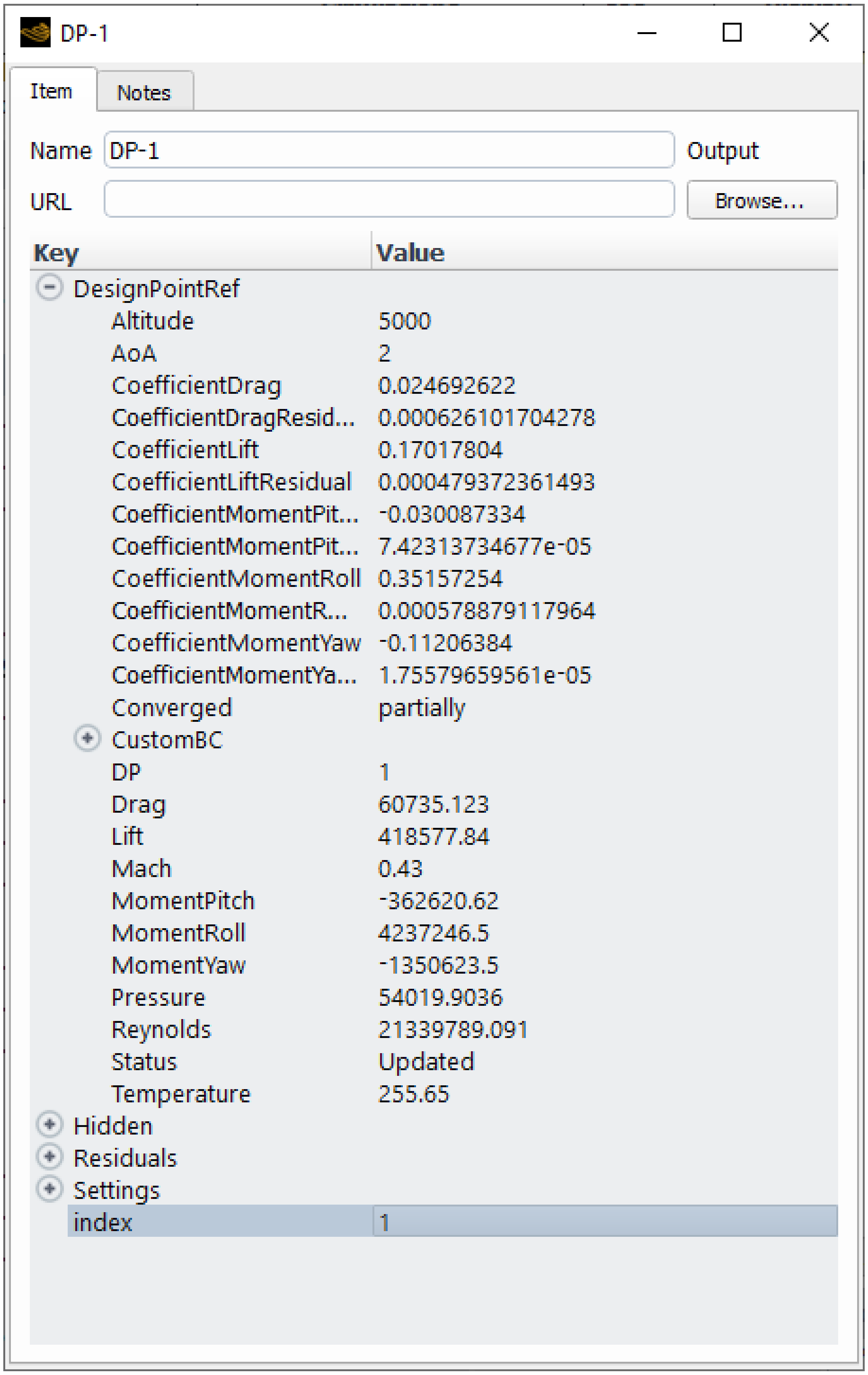
  Sorts the Design Point folders alphabetically by name.

- **Edit Notes**

  Opens a **Properties** window showing the **Notes** panel, where a user can add text notes to the selection. If a text note is added, the item inside the project view will be displayed with an * icon next to its name, signifying that a note has been added.

- **Properties**

  Opens a properties window showing the Metadata and properties of the Design Point. The figure below shows an example of the contents of a **Design Point** properties window.




The contents of the Design Point Properties window is similar to those of the **Results** properties, with the following differences.

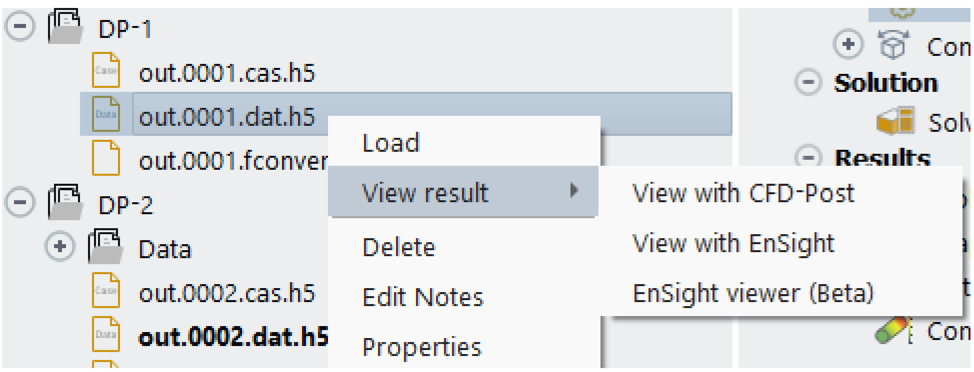
- **DesignPointRef** contains the input and output parameters associated with the Design Point. These values are used to generate the **Table:Coefficients** and **Table:Summary** accessible from the **Tables** → **Create Results Tables** in the **Outline View**.

- **DesignPointOutput** is similar to **DesignPointRef** but specifically contains the results of any custom output parameters.

– Residuals contains the final residual values achieved at the final iteration calculated for the Design Point. These values are used to generate the **Table:Residuals** accessible from the **Tables** → **Create Results Tables** in the **Outline View**.

– All **GeometricProperties, SimulationConditions** metadata will generally be the same as contained in it's parent **Results** folder. However, if any of these settings were modified and a recalculate **Selected Design Point** was performed, these settings will be locally updated on the Design Point folder to correspond to these settings changes. If the metadata is the same as that of its parent **Results** folder, the text will be in italic font. If the metadata is different than that of its parent **Results** folder, the text will be in standard font.

## 33.10.5. Solution File Commands




- **Load**

  Loads the selected solution file to memory, and shows the solution file as bold in the **Project View** menu. This allows the selected solution to be post-processed from the **Results** menu in the **Outline View**.

- **View results** → **View with CFD-Post**

  Opens a CFD-Post post-processing session in a new window and loads the selected case file.

- **View result** → **View with EnSight**

  Opens an EnSight post-processing session in a new window and loads the selected case file.

- **View result** → **EnSight viewer (Beta)**

  Opens an EnSight post-processing session inside the Fluent Aero User Interface. An object tree for EnSight will be loaded in the **Outline View** (below the **Simulation Workflow** steps), and the EnSight viewer will be loaded inside the Fluent Aero graphics window.

- **Delete**

  Deletes the selected file.

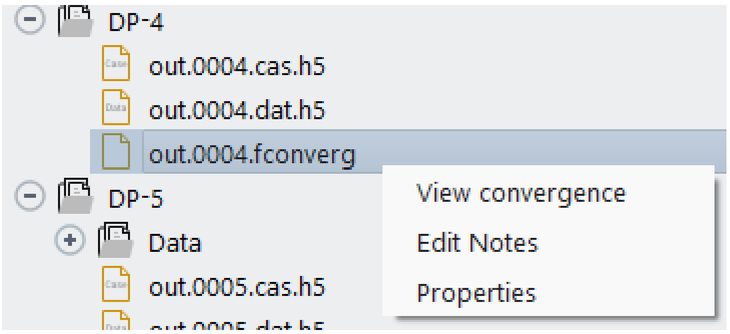
- **Edit Notes**

  Opens a **Properties** window showing the **Notes** panel, where a user can add text notes to the selection. If a text note is added, the item inside the project view will be displayed with an * icon next to its name, signifying that a note has been added.

- **Properties**

  Opens a **Properties** window showing the Metadata and the properties of the case file.

### 33.10.6. Convergence File Commands




- **View convergence**

  Load the convergence file in the convergence **Plots** window, using the **Dataset: Custom**. This allows the user to conveniently investigate the convergence of a particular Design Point calculation directly from the **Project View**.

- **Edit Notes**

  Opens a **Properties** window showing the **Notes** panel, where a user can add text notes to the selection. If a text note is added, the item inside the project view will be displayed with an * icon next to its name, signifying that a note has been added.

- **Properties**

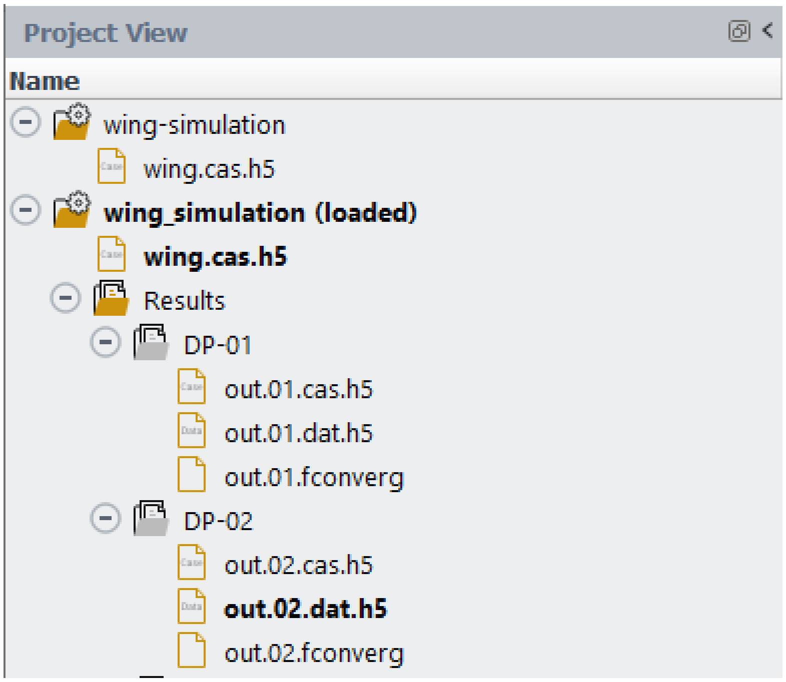
  Opens a **Properties** window showing the Metadata and properties of the convergence file.

### 33.10.7. The Use of Bold in Project View

The **Project View** window will make use of bold text to provide additional information on **Simulations**, **Case** and **Data** file status.

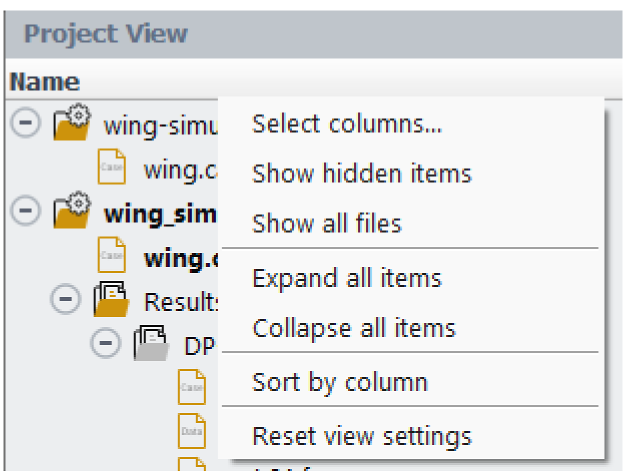
- A **Simulation** that is currently loaded and connected to a Solver session will have its name be displayed in Bold with **(loaded)** listed next to the name.
- The **.cas[.h5]** and **.dat[.h5]** files that are currently loaded will be displayed in bold. These files will also be specified as the **Current files** in the **Simulation** properties metadata window.

The image below shows an example of a **Project View** window that shows that the Simulation **wing_simulation** is currently loaded, along with its case file **wing.cas.h5**, and its design point data file **out.02.dat.h5**.




## 33.10.8. Project View Organization Options

The **Project View** window can be sorted in various ways to help with file organization. By default, the **Project View** only displays the Name column in the **Project View** file organization panel. Right-clicking on the **Name** column header will display additional file organization settings.




- **Select columns…**

  Opens a window that allows the user to select the information to display in the **Project View**. Alternatively, this option can be selected by choosing **Display** → **Columns** from the Project ribbon. See Using Columns in Project View (p. 552) for more info.

- **Show hidden items**

  Shows additional folders and files that exist inside the **Simulation** and **Results** folders. See Hidden Items in **Project View** (p. 555) for more info.

- **Show all files**

  Shows the full content of the **Simulation** and **Results** folders, including temporary files, log files, etc.

- **Expand all items**

  Expands all items in the **Project View**

- **Collapse all items**

  Collapses all items in the **Project View**

- **Sort by column**

  Sorts all files by the selected **Column**. If the **Name** column is selected, the files will be sorted alphabetically. By default, the files in the **Project View** are sorted by order of creation.

- **Reset view settings**

  Resets the **Project View** to its initial settings. This will display only the **Name** column, and all files will be sorted by order of creation.

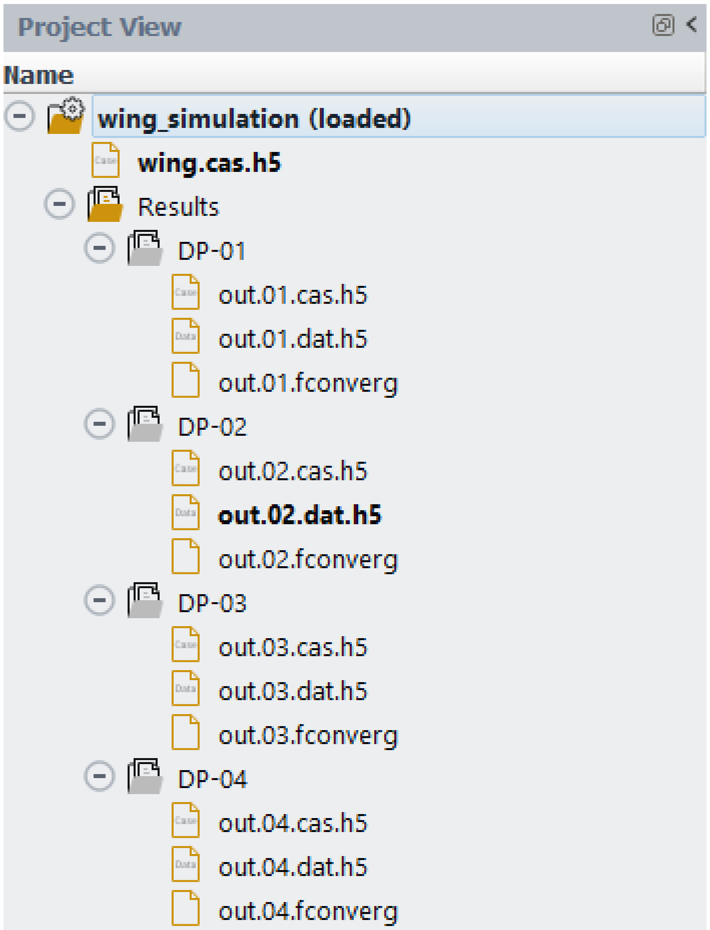
- **Sorting by Alt-select-drag**

  In addition to the sorting options described above, all files can be moved up or down in the **Project View** by **Alt**-selecting the file and dragging it to a new position. This operation can only be performed on the default file organization view.

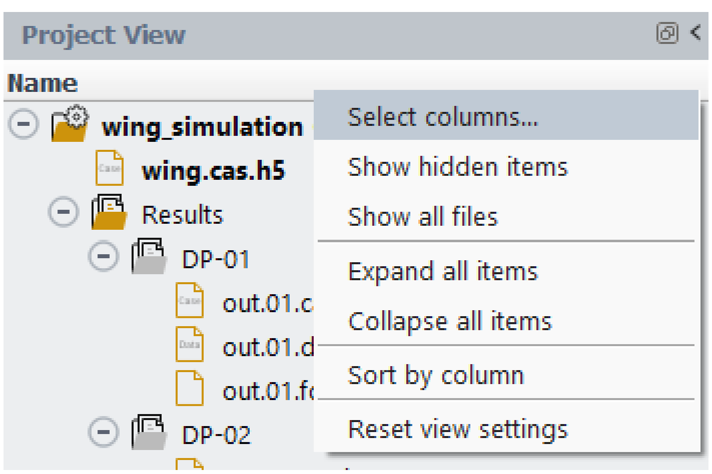
## 33.10.9. Using Columns in Project View

Sortable columns are available in the **Project View** window to show additional information and to help organize Simulations, Results and solution files.

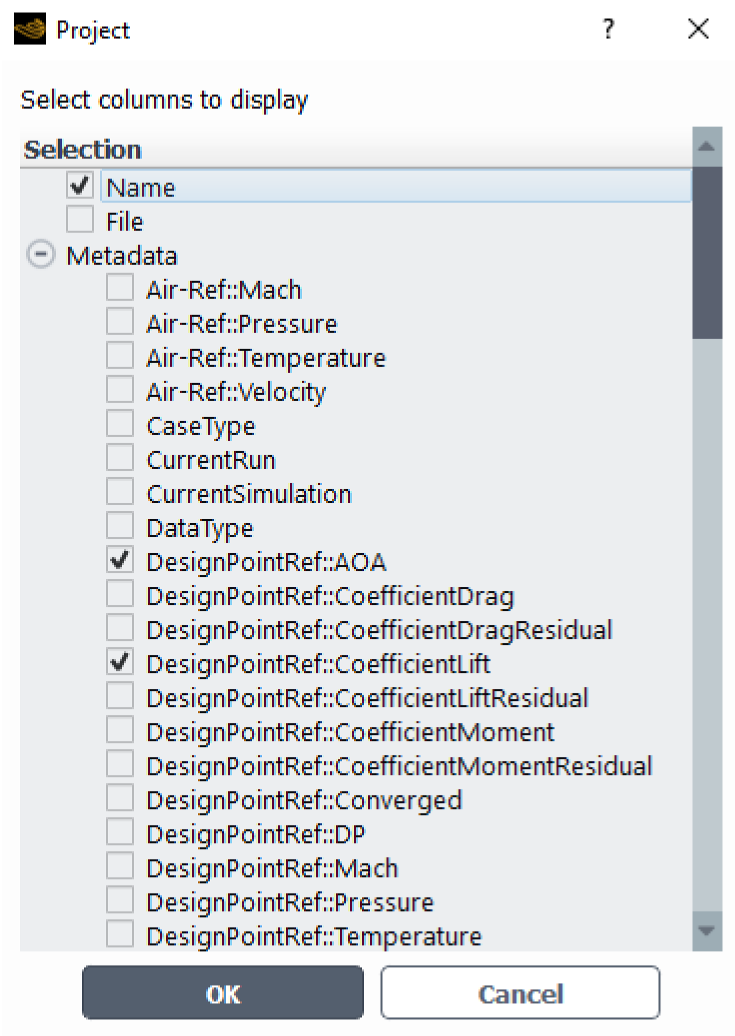
By default, **Project View** only displays the **Name** column, which displays the **Name** of each item.



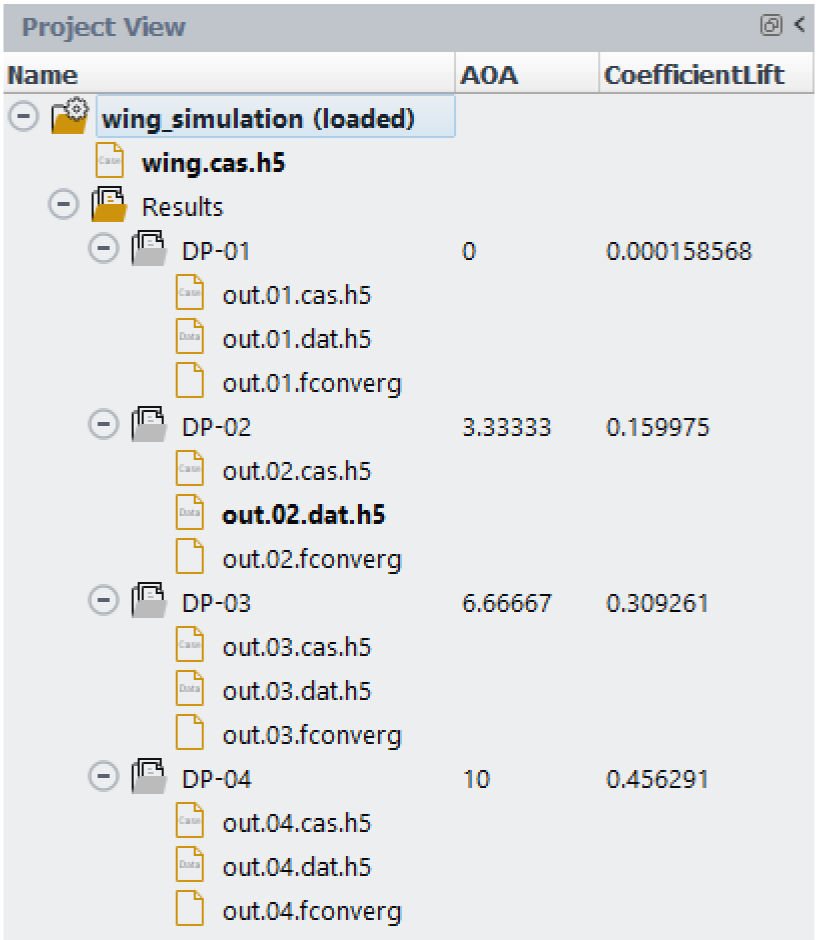

Right-click on the **Name** column in the header and choose **Select columns...** to open the column display menu. Alternatively, this window can be accessed by selecting **Display** → **Columns** in the **Project** ribbon menu.




A window appears showing the list of metadata settings that can be selected to display as a column in the **Project View**. For example, **DesignPointRef::AOA** and **DesignPointRef::CoefficientLift** can be selected to display the Angle of Attack and Lift Coefficient associated with each Design Point.

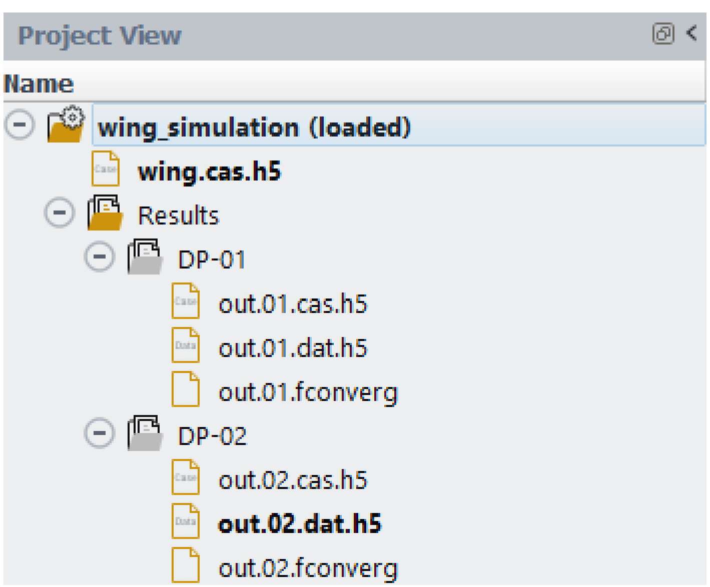



The selected information will then be displayed in a column to the right of the **Name** column. If the file contains a value associated with a display column, the value will be shown in the associated cell. If the file does not contain a property associated with a displayed column, the associated cell will be empty. In the image below, the columns show that 4 design points were calculated with different Angles of Attack (0, 3.33, 6.66 and 10 degrees) and resulted in a range of Lift Coefficients.

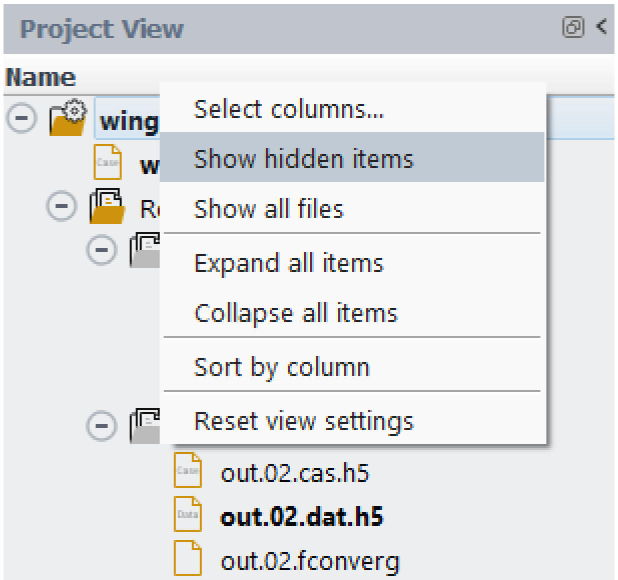



## 33.10.10. Hidden Items in Project View

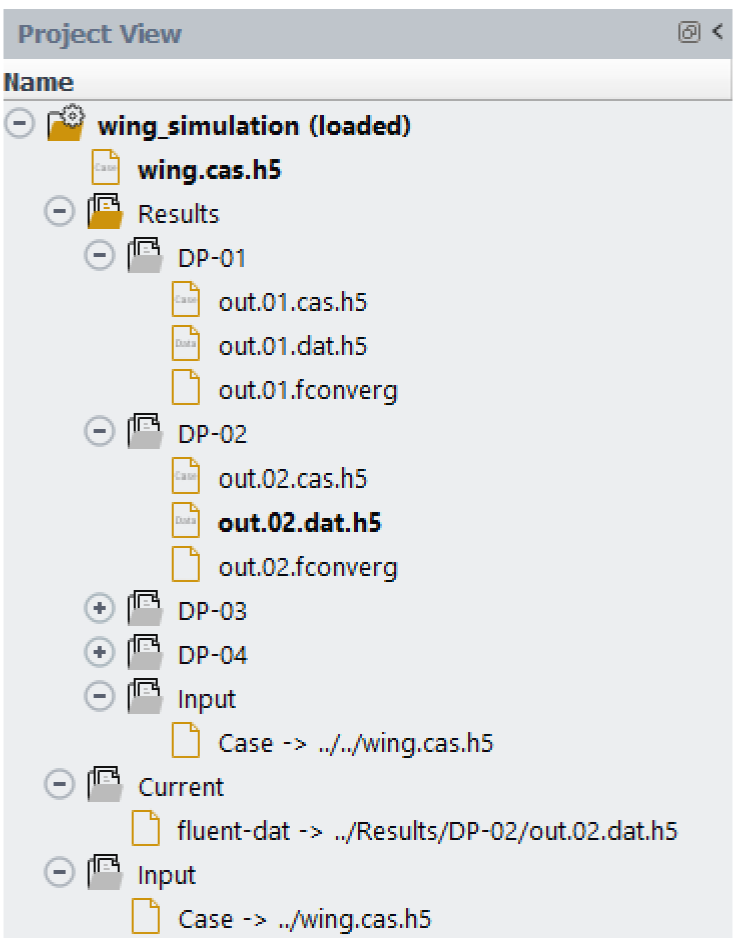
By default, the **Project View** only displays a selected list of case, result and convergence files as well as **Simulation**, **Results** and **Design Point** folders.



However, there are additional files associated with a Project that a user may want to interact with that are hidden by default. To display these files, right-click on the **Name** column header and select **Show hidden items**.




Hidden folders and files will be shown.



Some of the hidden folders include the following:

- **Current folder**

  This folder contains links to all the current solutions, which typically will be the solution file that was most recently obtained from a calculation, or which was most recently loaded using a load command. When **Load in solver** is used to open a Simulation, the solutions listed in the Current folder will be loaded and displayed in bold in the **Project View** menu.

- **Input folder**

  This folder contains links to all case files or solution files that were used as an input to the Simulation or Results.

# 33.11. Post-processing With CFD-Post and EnSight From Fluent Aero

The following sections of this chapter are:



In addition to the post-processing options available directly from the **Outline View**, the external tools CFD-Post and EnSight can also be used to post-process the simulation results that are present in the **Project View**.

## 33.11.1. CFD-Post From Fluent Aero

CFD-Post is a general post-processor for Ansys Fluids and can read Fluent case and data files. Usage is detailed in CFD-Post User's Guide.

There are two ways to open CFD-Post from Fluent Aero; by using View with CFD-Post from the **Results** → **Quick-view** ribbon menu, or by using **View with CFD-Post** from the **Project View**. Each method has a slightly different usage and is described in the following sections.

---

**Note:**

CFD-Post is installed by default with any Ansys Fluids product and uses a Post-processing license token. Launching it requires the **AWP_ROOT*** environment variable to be set-up. This is automatically done on Windows during the installation process. See Ansys, Inc. Installation Guides.

---

### 33.11.1.1. Accessing CFD-Post From Results → Quick-View

Select **View with CFD-Post** from the **Results** → **Quick-view** ribbon drop down menu. A new CFD-Post window will appear. The last written solution of the selected type will be loaded.

#### 33.11.1.2. Accessing CFD-Post from Project View

It is also possible to launch CFD-Post from the **Project View** by right-clicking on a solution file, and selecting **View with CFD-Post**. CFD-Post is then launched from the **Simulation** folder, a new CFD-Post window will appear, and the selected solution file will be loaded.

### 33.11.2. EnSight From Fluent Aero

EnSight is a general post-processor for Ansys Fluids and can read icing solutions directly via a command file or by selecting the files in its own **Open File** dialog. Usage is detailed in the Ansys EnSight User Manual.

There are three ways to open CFD-Post from Fluent Aero; by using **View with EnSight** from the **Project View**, or by using **EnSight viewer (Beta)** from the **Project View**. Each method has a slightly different usage and is described in the following sections.

---

**Note:**

EnSight is installed by default with any Ansys Fluids product and uses a Post-processing license token. Launching it requires the **AWP_ROOT*** ** environment variable to be set-up. This is automatically done on Windows during the installation process. See the Ansys, Inc. Installation Guides for more details.

---

#### 33.11.2.1. View with EnSight From Project View

Launch EnSight from the **Project View** by right-clicking on a solution file, and selecting **View with EnSight**. EnSight is then launched from the **Simulation** folder, a new EnSight window will appear, and the selected solution file will be loaded. Use the EnSight window's selectable options to further post process the solution file.

#### 33.11.2.2. EnSight Viewer (Beta) From Project View

Launch EnSight from the **Project View** by right-clicking on a solution file, and selecting **View with EnSight**. EnSight is then launched from the **Simulation** folder, an EnSight post processing object tree will appear in the **Outline View**, and the selected solution file will be loaded. Use the selectable EnSight post processing options available in the **Outline View** to further post-process the solution file.

---

**Note:**

This is a beta functionality, and may therefore have additional limitations.

---

# 33.12. Appendix

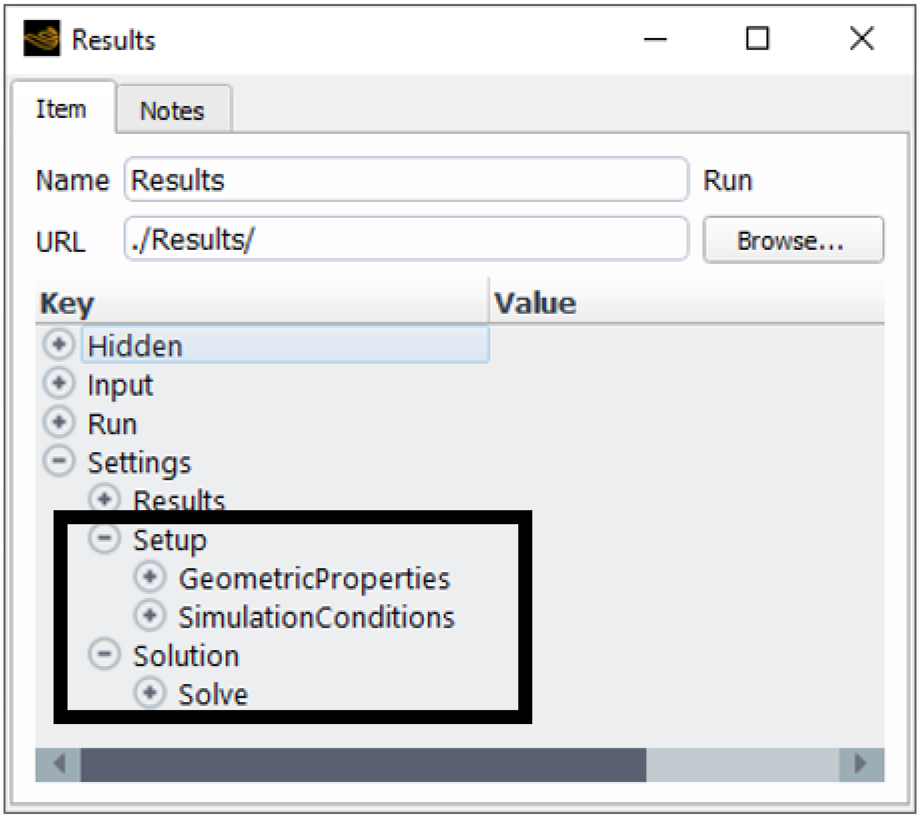
## Settings Files: projectname.flprj and run.settings

Two files are used to store the simulation and design point settings; the `projectname.flprj` file, and the `run.settings` file. The following section describes the breakdown of which settings that are included in each file.

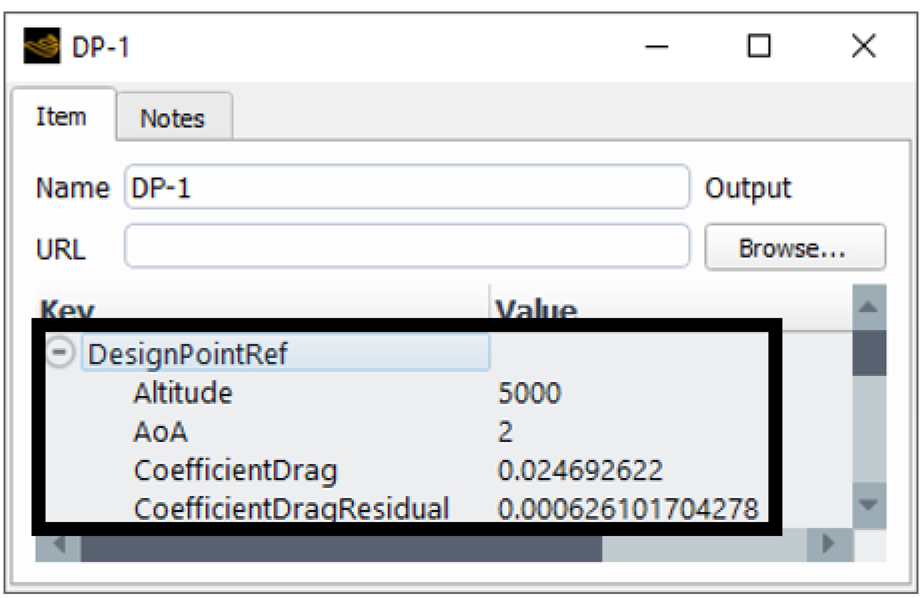
### `[projectname].flprj` file stored settings

Every Fluent Aero project is defined by a project folder `[projectname].cffdb`, which is a directory that contains all the simulation folders and files, and a project file `[projectname].flprj`, which is an XML format file that contains information about the files contained in the project folder including their dependencies, location, and additional metadata useful to each item. This metadata is used by Fluent Aero to store information related to the settings used to run a simulation and values of results produced. The `[projectname].flprj` file includes, but is not limited to, the following settings:

- **Datamodel settings** used by the **Geometric Properties**, **Simulation Conditions**, and **Solve** steps associated with the Results folder in each simulation.
  - These settings will be saved to the `[projectname].flprj` project file when calculating results.
  - The settings contained in this file will then be used to re-populate the datamodel settings for each of these steps in the **Outline View** when reconnecting to a previous simulation.
  - These values will be used by Fluent Aero to check to see if any Design Points need to be set to **Needs Update** when the datamodel step input value has changed.

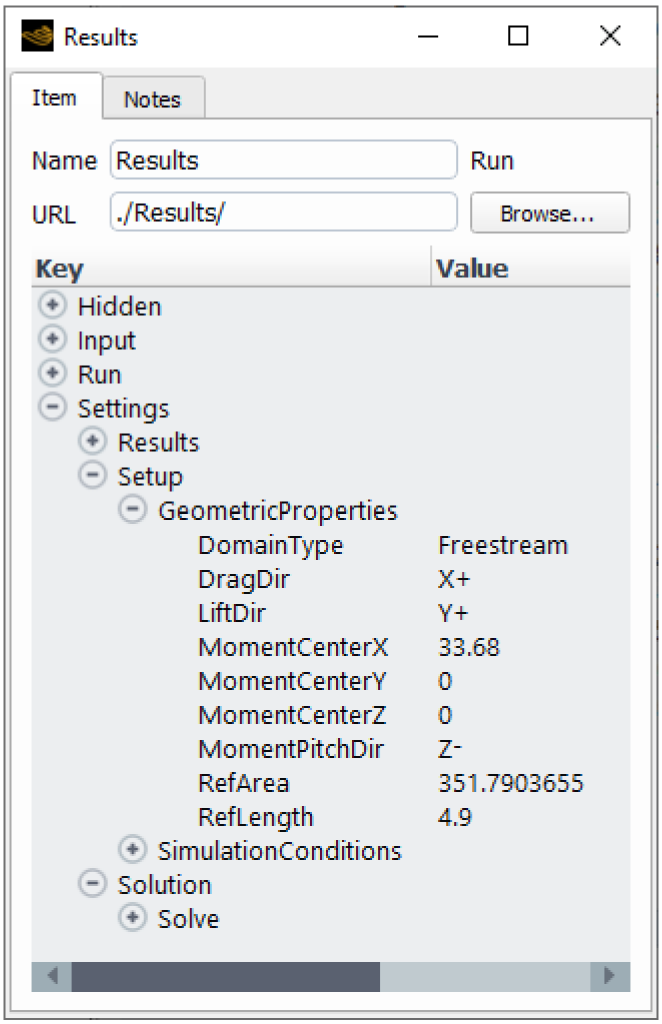



- **DesignPointRef metadata associated with each DP folder in each Simulation.**
  - These settings will be saved to the `[projectname].flprj` project file when calculating results.
  - The settings contained in this file will then be used to re-populate the cells of the **Input: Design Point Table** and all **Results** tables when reconnecting to a previous simulation.
  - These values will be used by Fluent Aero to check if a Design Point should be set to **Needs Update** when a cell value in the **Input Design Point Table** has changed.



These metadata settings can be viewed either directly inside the `[projectname].flprj` file using a text editor, or by right-clicking on an object inside the **Project View** and selecting **Properties**. The images below show an example of **Geometric Properties** settings from a **Results** folder in the `[projectname].flprj` and metadata properties panel, respectively.

```
<Results type="object" class="PFolder">
    <URL class="string" value="./Results/"/>
    <Subclass class="string" value="Run"/>
    <Metadata type="object" class="ansys::Project::MetadataHoof">
        <Run::OutputType class="string" value="Simple"/>
        <Run::SolverMode class="string" value="aero-custom"/>
        <Settings::Setup::GeometricProperties::DomainType class="string" value="Freestream"/>
        <Settings::Setup::GeometricProperties::LiftDir class="string" value="Y+"/>
        <Settings::Setup::GeometricProperties::DragDir class="string" value="X+"/>
        <Settings::Setup::GeometricProperties::MomentPitchDir class="string" value="Z-"/>
        <Settings::Setup::GeometricProperties::MomentCenterX class="double" value="33.68"/>
        <Settings::Setup::GeometricProperties::MomentCenterY class="double" value="0"/>
        <Settings::Setup::GeometricProperties::MomentCenterZ class="double" value="0"/>
```



## run.settings file stored settings

Inside the **Results** folder of every Fluent Aero simulation is a `run.settings` file that contains settings related to the setup of the boundary zones and the **Component Groups** step. The `run.settings` file includes the following settings:

- **Datamodel settings** used by the **Component Groups** step associated with the **Results** folder, which define the name and type of each boundary zone, the group name that each zone is a member of, and any input settings applied to a specific zone.
  - These settings will be saved to the `run.settings` file when calculating results.
  - The settings contained in this file will then be used to re-populate the datamodel settings for the **Component Groups** step in the **Outline View** when reconnecting to a previous simulation.
  - These values will be used by Fluent Aero to see if any Design Points needs to be set to **Needs Update** when the **Component Groups** step input value has changed.
  - Notably, if a user changes the name or type of a zone after **Results** have been calculated, the `run.settings` file will be deleted, and needs to be re-saved. See Modifying a Zone Type or Name under Modifying Settings After Results Have Been Calculated (p. 510) for more information.

A portion of an example `run.settings` file opened using a text editor, is shown in the image below.

```
{
    "ROOT": {
        "_RSESSION_NAME_": {
            "Case": {
                "App": {
                    "BC:BC1": {
                        "BCType": "pressure-far-field",
                        "IsExit": "false",
                        "IsInlet": "true",
                        "IsWall": "false",
                        "_name_": "farfield",
                        "AirflowMassFlowInlet": {
                            "BCSync": "Case settings",
                            "DirectionMode": "Case settings",
                            "FlowX": "0",
                            "FlowY": "0",
                            "FlowZ": "0",
                            "MassFlow": "0",
                            "MassFlowCustom": "EnterExpressionName",
                            "MassFlowCustomFlag": "false",
                            "Pressure": "0",
                            "PressureCustom": "EnterExpressionName",
                            "PressureCustomFlag": "false",
                            "SettingsEditable": "false",
                            "SettingsVisible": "false",
                            "Temperature": "0",
                            "TemperatureCustom": "EnterExpressionName",
                            "TemperatureCustomFlag": "false"
                        },
```

In future releases, it is possible that only the `run.settings` file will be used to store a larger portion of the simulation and design point settings than is currently being stored.

### 33.12.1. Python Console

Some of the Fluent Aero features and settings can be accessed and changed from the console using Python commands.

---

**Note:**

The Python Console support from the 2021R2 Fluent Aero release should be considered as a beta feature, since it does not yet contain sufficient commands to set up and run all cases. Additional support will be added in a future release.

---

The same commands can be used to automate Fluent Aero, by launching it with the `-R` argument,as in:

**Linux**

```
bin/aero-R file.py -I -t CPU
```

**Windows**

```
bin/aero.bat-R file.py -I -t CPU
```

By default, `-R` will execute the Python script then exit the application (the exit status code of the application will be non-zero if the Python script has thrown an exception or some other error occurred).

Adding the `-I` argument runs the Python script in interactive mode, the application will not close after the execution of the Python script, and is currently always required by Fluent Aero. The `-wait` argument waits until the execution is completed before resuming batch execution. (This is not required when using the `-R`argument)Without the`-wait`command, Fluent Aero is sent to background and the return code of the application cannot be used. The CPU count to use (`-tCPU`) is required, as this launch procedure does not use the **Fluent Launcher**.

#### Python Console & Operations

The **Console** window at the bottom right of Fluent Aero permits to enter commands interactively. The following sections describe example commands available in the Python console.

**dir() and the command/attribute list**

```
dir()
```

Display the list of global commands and top-level objects.

**'AddSession', 'Project', 'ReadScriptFile', 'Sim', 'StartTranscript', 'StopTranscript', 'csim', 'current-Simulation'**

```
dir(object)
```

Display the list of commands & attributes of the object.

**Example 33.3: Entering Commands Interactively (Assuming airfoil-demo is already loaded) dir(Sim[“airfoil-demo”])**

```
dir(Sim[“airfoil-demo”])
```

**'AeroWorkflow', 'Connect', 'ConnectionInfo', 'Disconnect', ‘SendCommand’**

```
r= currentSimulation()
```

A temporary variable can be used to simplify command lines.

```
dir(r.AeroWorkflow.Setup.GeometricProperties)
```

**'DomainType', 'DragDir', 'LiftDir', ‘RefArea', 'RefLength', `RefreshBCs’**

**Accessing/Changing values**

The () operator will return the current value of the attribute.

*Result:*1

The= operator will set the current value of the attribute.

```
r.AeroWorkflow.Setup.GeometricProperties.RefArea= 0.05
```

**Commands**

If an item listed in dir() is a verb, it is a command. For example:

```
r.AeroWorkflow.Setup.GeometricProperties.RefreshBCs()
```

**Sample Script**

```
r = RemoteSession["RemoteSession1"]
r0 = "RemoteSession1"

# Geometric Properties
r.AeroWorkflow.Setup.GeometricProperties.LiftDir = "Y+"
r.AeroWorkflow.Setup.GeometricProperties.DragDir = "X+"
r.AeroWorkflow.Setup.GeometricProperties.MomentCenterX = 0.2
r.AeroWorkflow.Setup.GeometricProperties.MomentCenterY = 0
r.AeroWorkflow.Setup.GeometricProperties.MomentCenterZ = 0
r.AeroWorkflow.Setup.GeometricProperties.RefLength = 0.5334
r.AeroWorkflow.Setup.GeometricProperties.RefArea = 0.05334

# Simulation Conditions
r.AeroWorkflow.Setup.SimulationConditions.FlightConditions.FlowSpeed.Parameter = "Mach"
r.AeroWorkflow.Setup.SimulationConditions.FlightConditions.FlowSpeed.Distribution = "Constant"
r.AeroWorkflow.Setup.SimulationConditions.FlightConditions.FlowSpeed.Mach = 0.3
r.AeroWorkflow.Setup.SimulationConditions.FlightConditions.FlowDirection.Parameter = "AoA"
r.AeroWorkflow.Setup.SimulationConditions.FlightConditions.FlowDirection.DistributionAoa = "Uniform"
r.AeroWorkflow.Setup.SimulationConditions.FlightConditions.FlowDirection.AoaMin = 3
r.AeroWorkflow.Setup.SimulationConditions.FlightConditions.FlowDirection.AoaMax = 5
r.AeroWorkflow.Setup.SimulationConditions.FlightConditions.FlowDirection.AoaNum = 2
r.AeroWorkflow.Setup.SimulationConditions.FlightConditions.PresTempInput.Parameter = "Static"
r.AeroWorkflow.Setup.SimulationConditions.FlightConditions.PresTempInput.DistributionPressure = "Constant"
r.AeroWorkflow.Setup.SimulationConditions.FlightConditions.PresTempInput.DistributionTemperature = "Constant"
r.AeroWorkflow.Setup.SimulationConditions.FlightConditions.PresTempInput.Pressure = 101325
r.AeroWorkflow.Setup.SimulationConditions.FlightConditions.PresTempInput.Temperature = 300

# Solve
r.AeroWorkflow.Solution.Solve.Iterations = 300
r.AeroWorkflow.Solution.Solve.ShowAdvanced = True
r.AeroWorkflow.Solution.Solve.SolverType = "Pressure based"

# Calculate
r.AeroWorkflow.Solution.Solve.AeroCalculate()
```

## 33.12.2. Data Structure Hierarchy

- RemoteSession (Simulation)
  - AeroWorkflow
    - → Setup
      - GeometricProperties
      - SimulationConditions
      - ComponentGroups
    - → Solution
      - Solve
    - → Results

## 33.12.3. Global Functions

- `currentSimulation()`

  Returns the current `RemoteSession`.If multiple Fluent connections exist, the current is related to the current selection in the **Outline View**.

- `csim()`
- Returns the `currentSimulation().Case.App`

## 33.13. Fluent Aero Tutorial

The following sections of this chapter are:



### 33.13.1. Computing Aerodynamic Coefficients on an ONERA M6 Wing at a Range of Angles of Attack Using AOA Exploration

The objective of this tutorial is to use Fluent Aero to obtain lift, drag and moment coefficients on an ONERA M6 swept wing geometry at a range of Angles of Attack.

Download the `fluent_aero_tutorial.zip` file here.

Unzip `fluent_aero_tutorial.zip` to your working directory.

1. Extract the `oneram6-wing-coarse.cas.h5` file and the `reference_data` folder for this tutorial. The `reference_data` folder contains several `.csv` formatted text files that will be compared with the results from the current calculation.

   **Note:**

   These reference data files have been generated for demonstration purposes only, and may not necessarily be accurate.

   The `oneram6-wing-coarse.cas.h5` file contains a grid of an ONERA -M6 swept wing that consists of 35,000 nodes, and 142,000 cells. Tetrahedral cells define most of the computational domain. Four layers of prisms are grown off the wing's wall boundaries. The limits of the computational domain are defined by a hemispherical boundary defined as a pressure-farfield boundary type, and a flat circular boundary defined as a symmetry plane in the Z direction.

   **Note:**

   This is a very coarse mesh. Its sole purpose is to quickly demonstrate a typical workflow in Fluent Aero, and should not be relied upon for accurate simulations. For more accurate simulations, a finer mesh appropriate for external aerodynamic simulations, featuring more prism layers, higher mesh surface refinement, and increased mesh density in the wake region, should be used.

In order for Fluent Aero to automate the setup of boundary conditions, it is recommended to use input case files that follow the domain requirements associated with either a **Freestream** or **WindTunnel**, type domain, depending on what is being used. For a **Freestream** type domain, the external boundary of the domain must be defined by a **pressure-far-field** type domain, and optionally a **pressure-outlet** type domain that is attached to the **pressure-far-field**. If these boundary types are setup in Fluent Meshing or Fluent Solution before importing into Fluent Aero, Fluent Aero will automatically place these boundaries into the **Freestream** boundary group which will be used to define the **Freestream** simulation conditions. However, Fluent Aero will not automatically set any boundary conditions on any other boundary zones that are not a part of the **Freestream** or **WindTunnel** group. For all other zones, the user can define a boundary condition manually in the **Component Groups** step or Fluent Aero will use the boundary condition setup of the initial case file or as setup from the Solution Workspace. Refer Freestream or WindTunnel Domain Type Requirements (p. 451) for more information.

**Figure 33.3: View of the Surface Mesh around the ONERA M6 Wing**

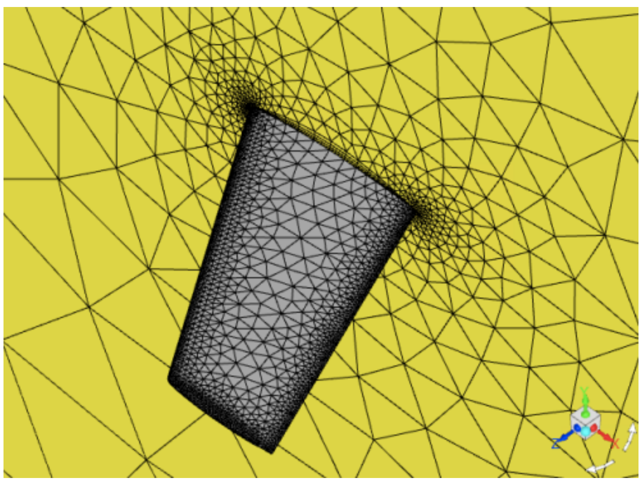
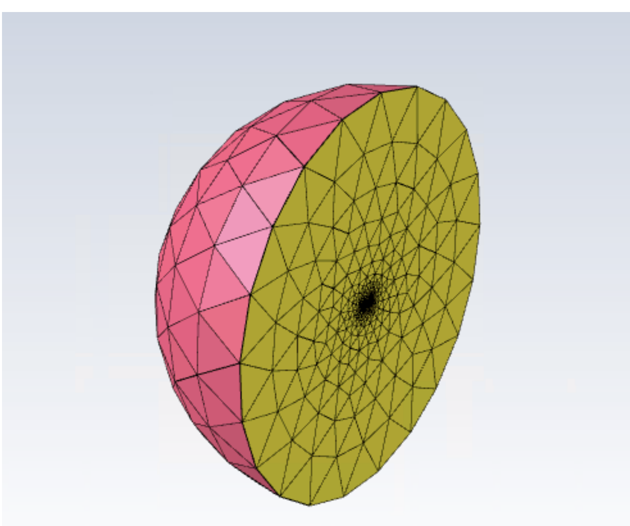


**Figure 33.4: Boundary Surface Mesh of the ONERA M6 Domain**



2. Launch **Fluent 2021 R2** on your computer. On the **Fluent Launcher** panel, set the **Capacity Level** to **Premium** or **Enterprise**. Then enable the **Show Beta Workspaces** option and select **Aero (Beta)**. Set the number of **Solver Processes** to **4-16**. Click **Start**. Alternatively, go to your Ansys installation folder and double-click the `aero` (on Linux) or `aero.bat` (on Windows) file inside the `fluent/bin/` folder.

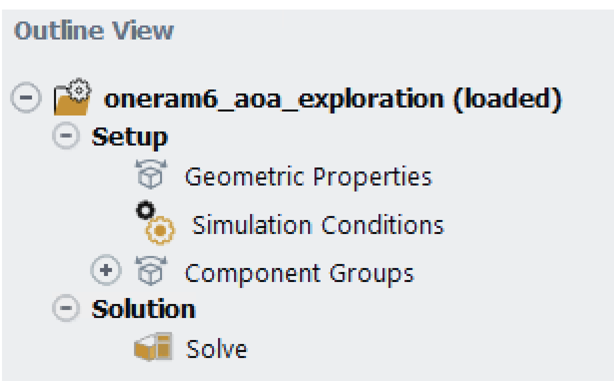
3. In the Fluent Aero Workspace, go to **File** → **Preferences**. In the **Preference** window, go to the **Aero** menu and enable **Use Custom Solver Launch Settings**. If this option is enabled, a **Fluent Launcher** window will open when loading a case file, allowing you to schedule a calculation on a server and/or to specify a different number of **Solver Processes** to use in your simulation. Alternatively, if this setting is disabled, Fluent Aero will load the case file in a Solver session on your local machine.

   Also, in the **Preferences** → **Aero** menu, make sure **Show Solution Workspace** and **Enable Solution Workspace Graphics** are disabled. Click **OK**.

4. When Fluent Aero first opens, the **Project** tab will be displayed by default. In the **Project**'s top ribbon panel, select **Project** → **New...** and enter **`Fluent_Aero_Tutorial_01`** to create a new Project folder. In the **Project**'s top ribbon, select **Simulations** → **New Aero Workflow**, and browse to and select the `oneram6-wing-coarse.cas.h5` file. A **New simulation** window will appear. Enter the **Name of the new simulation** as **`oneram6_aoa_exploration`**, and check to enable **Load in Solver**. Click **OK**.

5. If a **Fluent Launcher** window appears, set the **Solver Processes** to **4-16** and click **Start**. The case file will be opened and a background Solver session will be loaded. A new **Simulation** folder will be created in your **Project** folder, and the `oneram6-wing-coarse.cas.h5` file will be imported.

   After the `.cas.h5` file has been successfully loaded, a new **Outline View** tree appears under **oneram6_aoa_exploration (loaded)**.

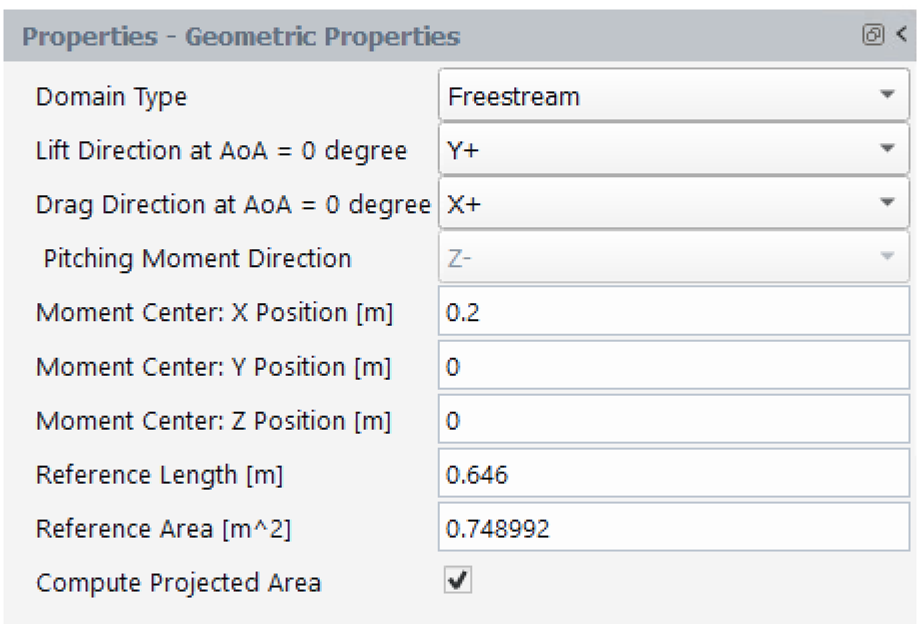
   
   

   While importing, Fluent Aero will search for and find the **pressure-far-field** zone that defines the external boundary of the domain. If present, this will cause Fluent Aero to determine that this case is using a **Freestream** domain type, and the following message will be reported in the console.

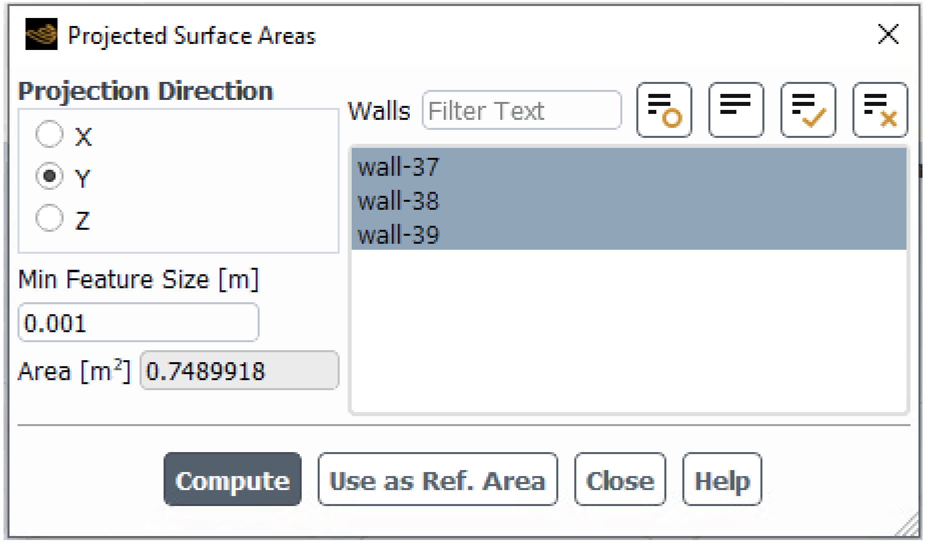
```
----------------------------------------------------------------
-                    Simulation Domain Type                    -
----------------------------------------------------------------
Fluent Aero has been set to use a Freestream domain type.

- The following boundary zones have been detected:
  > Freestream Inlet (type: pressure-far-field)
    - pressure-far-field-36
  > Freestream Outlet, not required (type: pressure-outlet):
    - None

- The Domain Type will be set to Freestream,
  and the group will be created in Component Groups
---------------------------------------------------------------
```

6. In the **Outline View** window, click **Geometric Properties**. A **Properties - Geometric Properties** window appears below the **Outline View** window. At the top of this new properties window, notice that the **Domain Type** has been automatically set to **Freestream**.

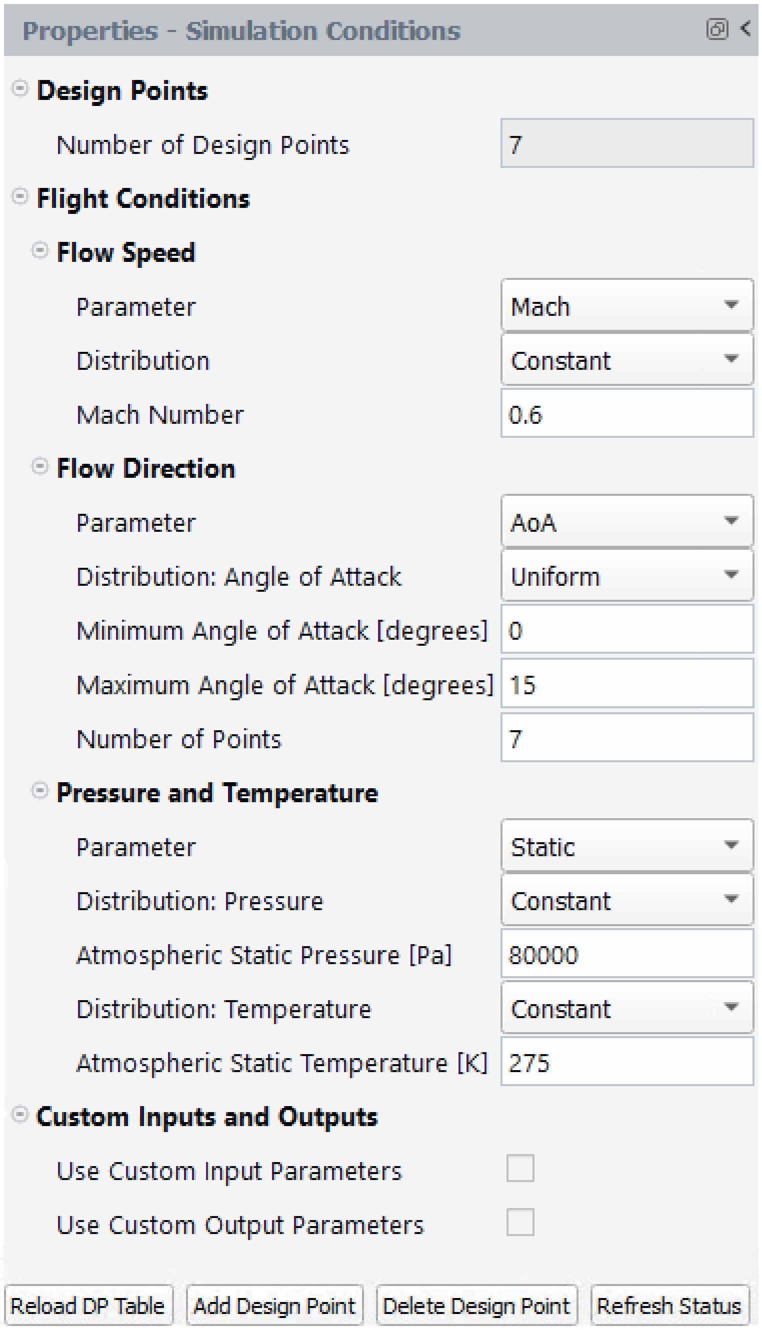
   Define the orientation of the geometry within the computational domain, which will be used to compute the aerodynamic forces. In this case, set **Lift Direction at AoA = 0 degree** to **Y+**, set **Drag Direction at AoA = 0 degree** to **X+**. **Pitching Moment Direction** will automatically be set to **Z-**. Set the **Moment Center X-**, **Y-** and **Z-Position [m]** to `0.2`, `0`, and `0`, respectively. Set the **Reference Length [m]** to `0.646`, which corresponds to the mean chord length.




   Set the **Reference Area [m^2]** to `0.748992`. Alternatively, the reference area can be computed by enabling the **Compute Projected Area** option. A pop-up panel will appear. Set the **Projection Direction** to **Y** and select all the **wall** surfaces. Click on the **Compute** button and then click on the **Use as Ref. Area** button to copy the computed area to the **Reference Area [m^2]** box.




7. In the **Setup** tree, go to **Simulation Conditions**. In the **Properties – Simulation Conditions** window, go to **Flight Conditions**. In the **Flow Speed** section, set **Parameter** to **Mach**, **Distribution** to **Constant** and the **Mach Number** to **`0.6`**. In the **Flow Direction** section, set **Parameter** to **AoA** and **Distribution: Angle of Attack** to **Uniform**. Set the **Minimum Angle of Attack [degrees]** to **`0`**, the **Maximum Angle of Attack [degrees]** to **15** and the **Number of Points** to **7**. Go to the **Pressure and Temperature** section. Set **Parameter** to **Static**, **Distribution: Pressure** to **Constant** and the **Atmospheric Static Pressure [Pa]** to **`80000`**. Set **Distribution: Temperature** to **Constant** and the **Atmospheric Static Temperature [K]** to 275.



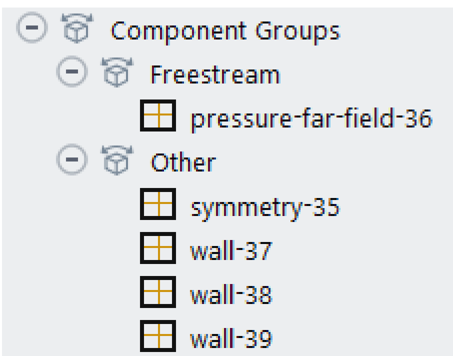

An **Input: Design Points** table will be created in the graphic window on the right-hand side of the user interface. This table shows all the Design Points that will be simulated. The initial status of each Design Point (DP) has been set as **Needs Update**. The status can be set as **Do Not Update** if you decide not to update one or several Design Points. In this tutorial, we will keep all Design Points as **Needs Update**.

**Figure 33.5: Initial Input: Design Points Table**

Input:Design Points

| DP | Angle of Attack [deg.] | Status |
|---|---|---|
| 1 | 0.0 | Needs Update |
| 2 | 2.5 | Needs Update |
| 3 | 5.0 | Needs Update |
| 4 | 7.5 | Needs Update |
| 5 | 10.0 | Needs Update |
| 6 | 12.5 | Needs Update |
| 7 | 15.0 | Needs Update |

8. Go to **Component Groups**. Two default **Component Groups** have been created after the simulation is loaded. The **Freestream** group contains the pressure-farfield zone that defines the external boundary of the domain, and is where the Freestream atmospheric flight conditions for each Design Point are defined as boundary conditions. The **Other** group contains the remainder of the boundary zones which are the walls of oneram6 wing geometry and the symmetry plan of the simulation domain.




**Note:**

When using more complex geometries, it can be useful to re-organize zones and create additional **Component Groups** using the **Component Manager**. This type of procedure is described in detail in Introduction to Aircraft Component Groups and Computing Aerodynamic Coefficients on an Aircraft at Different Flight Altitudes and Engine Regimes (p. 611). However, this is not necessary for this simple case. Therefore, continue to the next step.

9. Go to **Solve**. The default number of iterations for each DP is **500**. Check to enable **Show Advanced Settings**. This reveals model and solver parameters that advanced users can edit. For now, the default settings for these parameters will be considered. Uncheck to disable **Show Advanced Settings**.

**Note:**

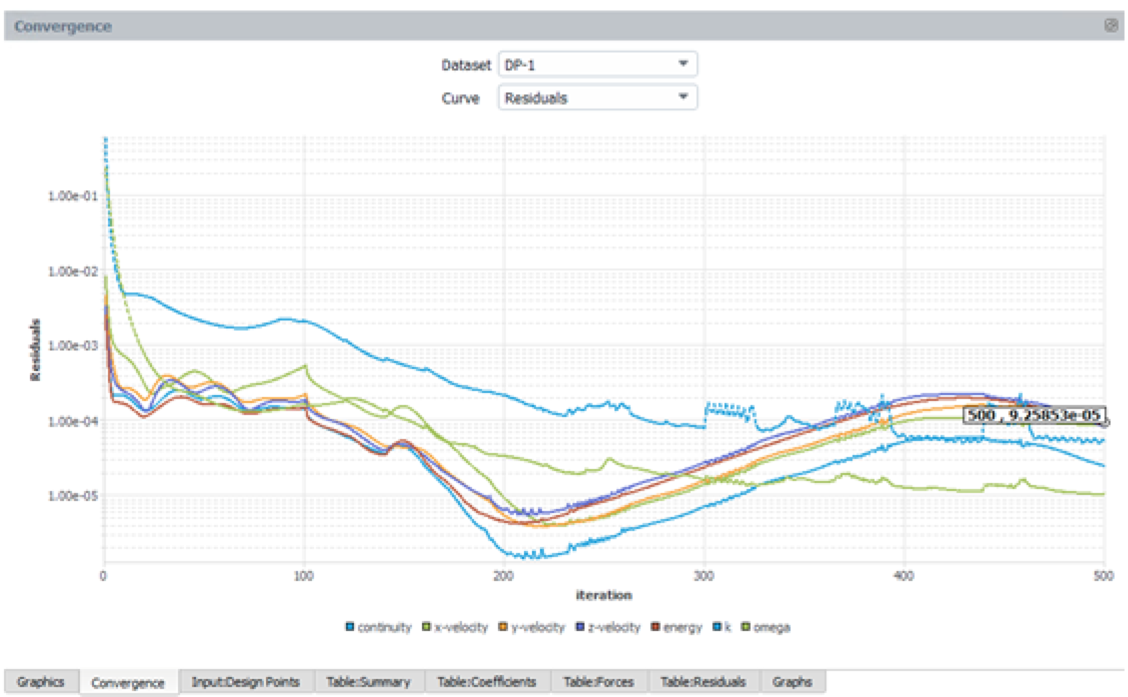
The advanced settings also include the convergence cut-off criteria for the residuals and the aerodynamic coefficients (lift and drag coefficients). These settings can be adjusted to achieve more relaxed or stricter convergence, as desired. For more information, refer to Solve (p. 494).

Click on the **Calculate** button at the bottom of the **Properties - Solve** panel. Alternatively, right-click on **Solve** from the **Outline View** tree and select **Calculate** from the drop-down menu. The calculation will start, and the first Design Point, **DP-1**, featuring the minimum Angle of Attack, will be simulated. A **Convergence** window will appear in place of the Graphics window and display the residuals and monitors for **DP-1**.

Design Point **DP-1** will calculate until the total **Iterations** (500) or the **Residuals Convergence Cutoff** (1e-4) and **Aero Coeff Conv. Cutoff** (2e-4) are reached, whichever comes first. In this example, DP-1 will calculate for 500 iterations. At that point, the angle of attack will be updated for the next Design Point and the calculation will resume. This process repeats until all Design Points are simulated.

10. Look at the convergence history of the simulation in the **Convergence** window located on the right of your screen. Set the **Dataset** to **DP-1** and the **Curve** to **Residuals**, to view the continuity, x- y- z-velocity, energy, and turbulence residuals for the first Design Point. You can left-click on a residual curve to show the iteration number and the corresponding residual value. Here, the residual of the y-velocity at iteration 500 is shown and equals to 9.25853e-5 which is slightly lower than the residual convergence cutoff (1e-4). The remaining residuals of the DP-1 have met the **Residuals Convergence Cutoff** criteria after 500 interations.

**Figure 33.6: Convergence of Residuals for Design Point 1**

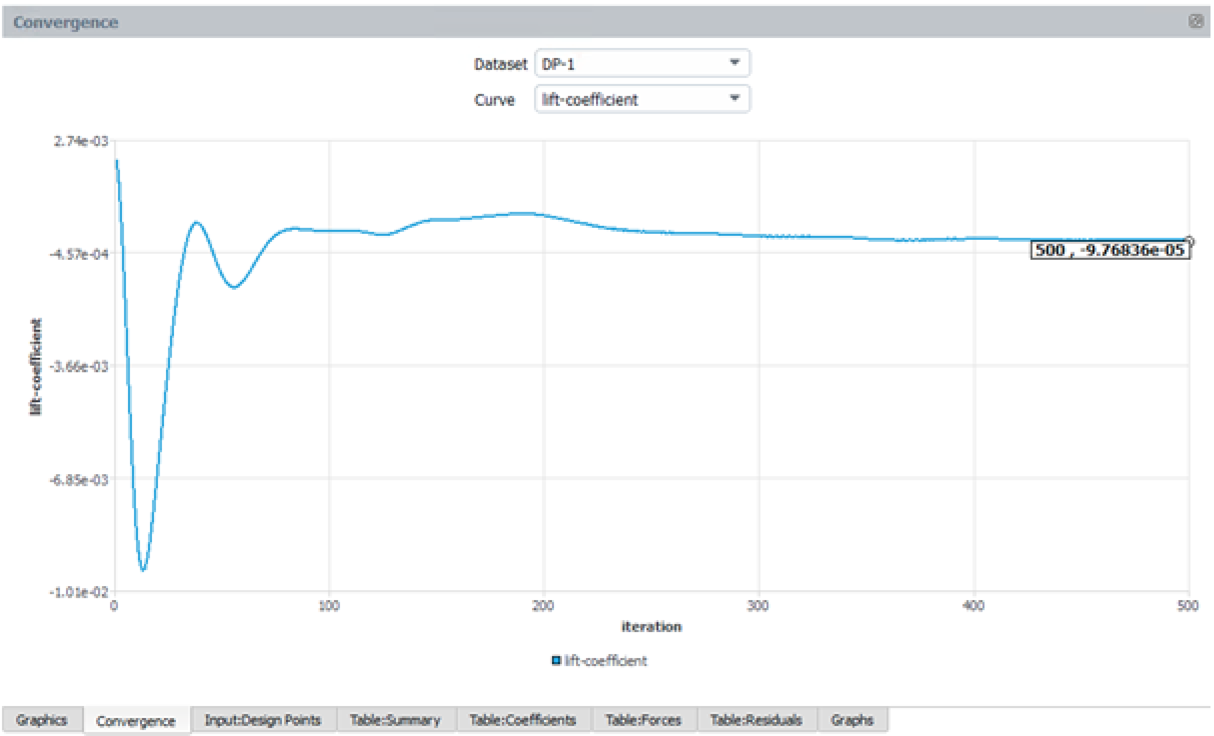



**Note:**

While all residuals in the image above meet the residual convergence cutoff criterion of 1e-4, it is still recommended for users to investigate their solutions to ensure that appropriate convergence levels have been achieved and that convergence remains stable. A residual convergence cutoff of 1e-4 may be appropriate for some cases, but not for others, and therefore care should be taken when selecting this value. If more strict convergence is required, a lower value (1e-5, or 1e-6 for example) could be set for the **Solve** → **Show Advanced Properties** → **Residuals Convergence Cutoff** option.

In the **Convergence** window, set **Curve** to **lift-coefficient**. The evolution of the lift coefficient for DP-1 will be displayed. Left-click on last iteration of the lift-coefficient plot, to show the lift coefficient value.

**Figure 33.7: Convergence History of the Lift Coefficient for Design Point 1**




When the calculation of DP-1 is complete, the status column in row 1 of the **Input: Design Points** table will be set to **Updated**, and the calculation of DP-2 will begin.

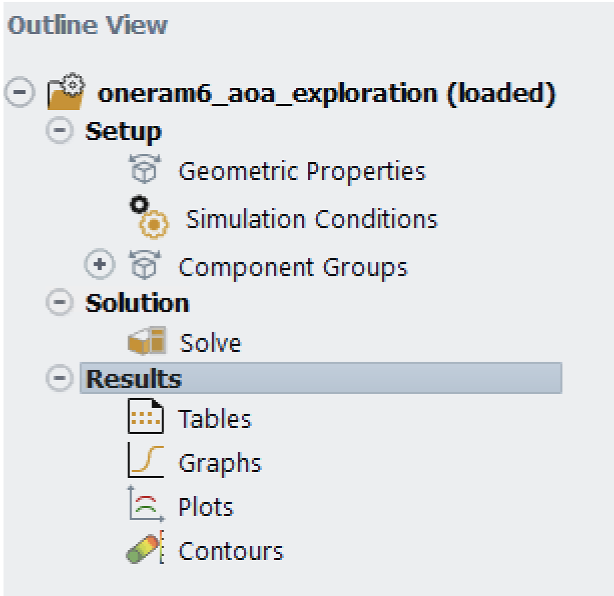
11. After all the Design Points have been updated, the status of the **Input: Design Points** table will be set to **Updated** for all the Design Points.

**Figure 33.8: Input: Design Points Table after Calculation**

Input:Design Points

| DP | Angle of Attack [deg.] | Status |
|---|---|---|
| 1 | 0.0 | Updated |
| 2 | 2.5 | Updated |
| 3 | 5.0 | Updated |
| 4 | 7.5 | Updated |
| 5 | 10.0 | Updated |
| 6 | 12.5 | Updated |
| 7 | 15.0 | Updated |

12. A **Results** section will be displayed in the **Outline View** tree after the simulation starts. This allows a user to quickly post-process design point solutions, by obtaining aerodynamic coefficient plots, creating contour plots of the solution fields, comparing solution fields to experimental data, and more.




13. The first element in the **Results** section is **Tables**. In the current tutorial, 4 different **Tables** will automatically be created in the **Graphics** window area when the calculation is complete.

**Note:**

You can also click the **Tables** → **Export Results Tables** button and the results tables will be exported to the **Results** folder of the current simulation and they will be visible in the **Project View** under the **Summary** folder.

Click on the **Table: Summary**, **Table: Coefficients**, **Table: Forces** and **Table: Residuals** tabs at the bottom of the **Graphics** window to reveal each table.

- **Table: Summary** summarizes the flight conditions and convergence information for each Design Point. In the current simulation, most design points have met or partially met the convergence criteria.

**Figure 33.9: Summary Table of the Flight Conditions and Convergence Information**

Table:Summary

| | DP | Mach | AoA [deg] | AoS [deg] | P [Pa] | T [K] | Reynolds | Avg Coeff Conv | Avg Residual | Criteria Met? |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0.6 | 0.0 | 0.0 | 80000.0 | 275.0 | 6.5846e+06 | 7.9415e-06 | 6.1315e-05 | yes |
| 2 | 2 | 0.6 | 2.5 | 0.0 | 80000.0 | 275.0 | 6.5846e+06 | 1.1576e-04 | 2.6549e-05 | yes |
| 3 | 3 | 0.6 | 5.0 | 0.0 | 80000.0 | 275.0 | 6.5846e+06 | 9.5842e-05 | 2.8081e-05 | yes |
| 4 | 4 | 0.6 | 7.5 | 0.0 | 80000.0 | 275.0 | 6.5846e+06 | 1.1055e-04 | 4.2039e-05 | yes |
| 5 | 5 | 0.6 | 10.0 | 0.0 | 80000.0 | 275.0 | 6.5846e+06 | 3.2709e-05 | 1.9390e-04 | partially |
| 6 | 6 | 0.6 | 12.5 | 0.0 | 80000.0 | 275.0 | 6.5846e+06 | 7.1977e-06 | 3.9127e-04 | no |
| 7 | 7 | 0.6 | 15.0 | 0.0 | 80000.0 | 275.0 | 6.5846e+06 | 9.8584e-05 | 1.6287e-05 | yes |

- **Table:Coefficients** contains the lift, drag, yaw moment, pitching moment and rolling moment coefficients. The **Cl Conv.** and **Cd Conv.** columns measure the convergence of the lift and drag coefficients, which are used to determine if the convergence criteria is met. The last column shows the maximum value of the convergence of the yaw, pitching and rolling moments.

**Note:**

The convergence of the moment coefficients are provided for reference only and are not used to verify if the convergence criteria is met.

**Figure 33.10: Results Table of Aerodynamic Coefficients**

Table:Coefficients

| | DP | Cl | Cd | Cm-y | Cm-p | Cm-r | Cl Conv. | Cd Conv. | Max. Cm Conv. |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | -9.7684e-05 | 1.1302e-02 | -3.3640e-03 | 9.4080e-05 | -4.0088e-05 | 1.4060e-06 | 1.4477e-05 | 3.8717e-06 |
| 2 | 2 | 1.6823e-01 | 1.4598e-02 | 2.0573e-03 | -6.5980e-02 | 1.3653e-01 | 5.8111e-05 | 1.7342e-04 | 2.3500e-03 |
| 3 | 3 | 3.3003e-01 | 2.6455e-02 | 1.3256e-02 | -1.2943e-01 | 2.6531e-01 | 1.7083e-04 | 2.0856e-05 | 7.8643e-04 |
| 4 | 4 | 4.7373e-01 | 5.5007e-02 | 1.4129e-02 | -1.9016e-01 | 3.6766e-01 | 5.4805e-05 | 1.6629e-04 | 2.8987e-03 |
| 5 | 5 | 5.6470e-01 | 9.5076e-02 | 6.4287e-03 | -2.2844e-01 | 4.1199e-01 | 2.2386e-05 | 4.3031e-05 | 3.0810e-04 |
| 6 | 6 | 6.2391e-01 | 1.3853e-01 | 1.3978e-03 | -2.5987e-01 | 4.4234e-01 | 9.6768e-06 | 4.7187e-06 | 5.4021e-04 |
| 7 | 7 | 6.4510e-01 | 1.8045e-01 | -3.9201e-03 | -2.8369e-01 | 4.6250e-01 | 9.7881e-05 | 9.9286e-05 | 4.8006e-05 |

- **Table:Forces** contains the lift, drag and moment forces.

**Figure 33.11: Results Table of Aerodynamic Forces**

Table:Forces

| | DP | Lift [N] | Drag [N] | Mom. Yaw [N.m] | Mom. Pitch [N.m] | Mom. Roll [N.m] |
|---|---|---|---|---|---|---|
| 1 | 1 | -1.4750e+00 | 1.7066e+02 | -3.2814e+01 | 9.1769e-01 | -3.9103e-01 |
| 2 | 2 | 2.5402e+03 | 2.2043e+02 | 2.0067e+01 | -6.4359e+02 | 1.3317e+03 |
| 3 | 3 | 4.9834e+03 | 3.9946e+02 | 1.2930e+02 | -1.2625e+03 | 2.5879e+03 |
| 4 | 4 | 7.1532e+03 | 8.3059e+02 | 1.3782e+02 | -1.8549e+03 | 3.5863e+03 |
| 5 | 5 | 8.5268e+03 | 1.4356e+03 | 6.2708e+01 | -2.2283e+03 | 4.0187e+03 |
| 6 | 6 | 9.4209e+03 | 2.0917e+03 | 1.3635e+01 | -2.5348e+03 | 4.3148e+03 |
| 7 | 7 | 9.7407e+03 | 2.7247e+03 | -3.8239e+01 | -2.7672e+03 | 4.5114e+03 |

- **Table: Residuals** shows the final residuals as well as the number of iterations run for each Design Point.

**Figure 33.12: Results Table of Final Residuals**

Table:Residuals

| | DP | iteration | continuity | x-velocity | y-velocity | z-velocity | energy | k | omega |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 500 | 2.4198e-05 | 8.4784e-05 | 9.2585e-05 | 8.0515e-05 | 8.4183e-05 | 5.2753e-05 | 1.0189e-05 |
| 2 | 2 | 266 | 5.7663e-06 | 1.1647e-05 | 1.5672e-05 | 2.1388e-05 | 2.0086e-05 | 9.7490e-05 | 1.3796e-05 |
| 3 | 3 | 226 | 7.5550e-06 | 1.9328e-05 | 2.0316e-05 | 2.4955e-05 | 2.2236e-05 | 8.9441e-05 | 1.2740e-05 |
| 4 | 4 | 266 | 1.6714e-05 | 3.0539e-05 | 4.2423e-05 | 5.2749e-05 | 6.1322e-05 | 8.0202e-05 | 1.0321e-05 |
| 5 | 5 | 500 | 9.4833e-05 | 6.0867e-04 | 1.6235e-04 | 7.2177e-05 | 3.9394e-04 | 1.9471e-05 | 5.8505e-06 |
| 6 | 6 | 500 | 1.8658e-04 | 1.2400e-03 | 3.2253e-04 | 1.4703e-04 | 8.2509e-04 | 6.7086e-06 | 1.0941e-05 |
| 7 | 7 | 296 | 2.1438e-06 | 7.6292e-06 | 6.4201e-06 | 7.1006e-06 | 7.4621e-06 | 7.6558e-05 | 6.6945e-06 |

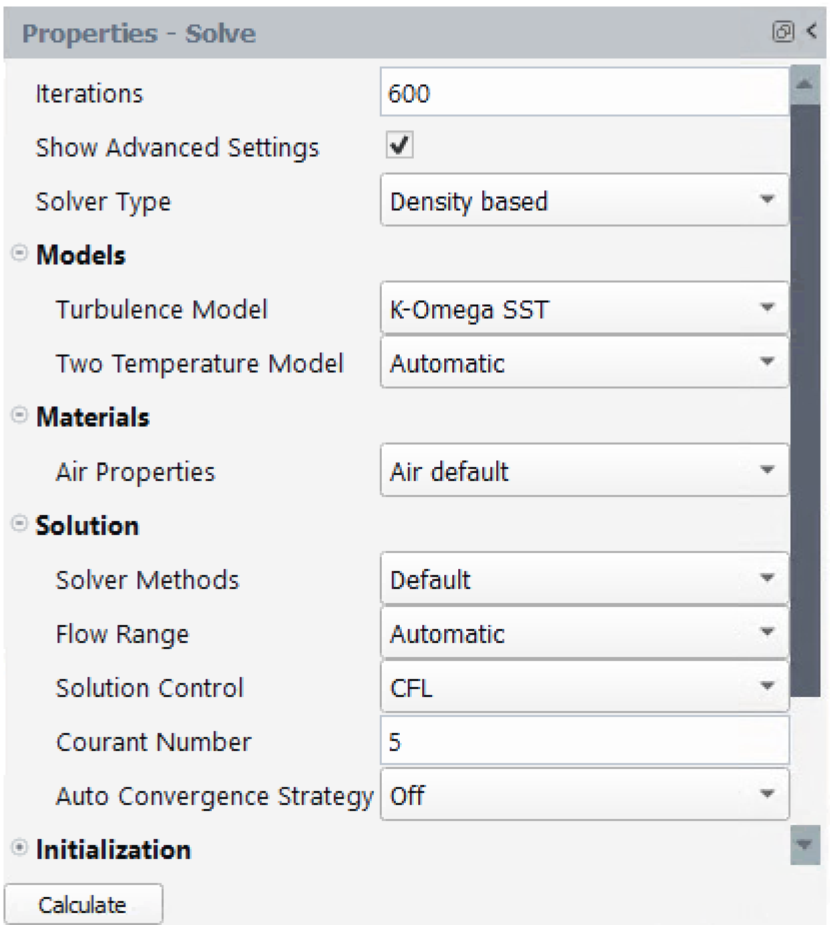
14. Based on the convergence status, the user can continue to calculate selected Design Points from the current results. Here, since the convergence criteria has not been met for Design Point 6, we will continue to calculate the **DP-6** from its last solution. In the **Input: Design Points** table, set the **Status** column of **DP-6** to **Continue To Update**.

**Figure 33.13: Set a Design Point to Continue to Update**

Input:Design Points

| DP | Angle of Attack [deg.] | Status |
|---|---|---|
| 1 | 0.0 | Updated |
| 2 | 2.5 | Updated |
| 3 | 5.0 | Updated |
| 4 | 7.5 | Updated |
| 5 | 10.0 | Updated |
| 6 | 12.5 | Continue To Update |
| 7 | 15.0 | Updated |

Click on **Solve** in the **Outline View** and go to the **Properties - Solve** panel. Set **Iterations** to `600`. Enable **Show Advanced Settings**. Set **Solution Control** to **CFL**. Set the **Courant Number** to **5**. These new settings will now be used for any subsequent Design Point calculations. Click **Calculate** to launch the computation. Since only **DP-6** has been set to **Continue To Update**, the calculation will only be performed for **DP-6**.




In a **Continue To Update** calculation, new residuals will be appended to the previous convergence plots. After about 578 iterations, all the convergence criteria will be met for **DP-6**, and the calculation will stop. Fluent Aero will update results in the tables and aerodynamic coefficient plots. For example, the status column of the **Table: Summary** will be changed from **No** to **Yes**.

**Figure 33.14: Convergence of the Residuals for Design Point 6 After a Continue to Update**


As shown in the new **Table: Summary** table, the **Avg. Coeff. Conv** and **Avg. Residual** have been improved and the **Criteria Met?** changes to **yes** for **DP-6**.

**Figure 33.15: Summary Table After a Continue to Update of DP-6**

Table:Summary

| | DP | Mach | AoA [deg] | AoS [deg] | P [Pa] | T [K] | Reynolds | Avg Coeff Conv | Avg Residual | Criteria Met? |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0.6 | 0.0 | 0.0 | 80000.0 | 275.0 | 6.5846e+06 | 7.9415e-06 | 6.1315e-05 | yes |
| 2 | 2 | 0.6 | 2.5 | 0.0 | 80000.0 | 275.0 | 6.5846e+06 | 1.1576e-04 | 2.6549e-05 | yes |
| 3 | 3 | 0.6 | 5.0 | 0.0 | 80000.0 | 275.0 | 6.5846e+06 | 9.5842e-05 | 2.8081e-05 | yes |
| 4 | 4 | 0.6 | 7.5 | 0.0 | 80000.0 | 275.0 | 6.5846e+06 | 1.1055e-04 | 4.2039e-05 | yes |
| 5 | 5 | 0.6 | 10.0 | 0.0 | 80000.0 | 275.0 | 6.5846e+06 | 3.2709e-05 | 1.9390e-04 | partially |
| 6 | 6 | 0.6 | 12.5 | 0.0 | 80000.0 | 275.0 | 6.5846e+06 | 7.4207e-06 | 4.2997e-05 | yes |
| 7 | 7 | 0.6 | 15.0 | 0.0 | 80000.0 | 275.0 | 6.5846e+06 | 9.8584e-05 | 1.6287e-05 | yes |

15. Click on **Graphs** in the **Outline View** to show the plots of the aerodynamic coefficients defined in Fluent Aero. At the bottom of the **Properties - Graphs** window, click **Plot Coefficients**. An X-Y plot of Lift Coefficient (**Cl**) vs. Design Point (**DP**) will appear in the **Graphics** window. The Drag and Moment Coefficients can be shown by selecting **Cd** and **Cm-r/y/p** from the **Curve** selection drop-down list.

**Figure 33.16: Lift Coefficient vs. Design Point**

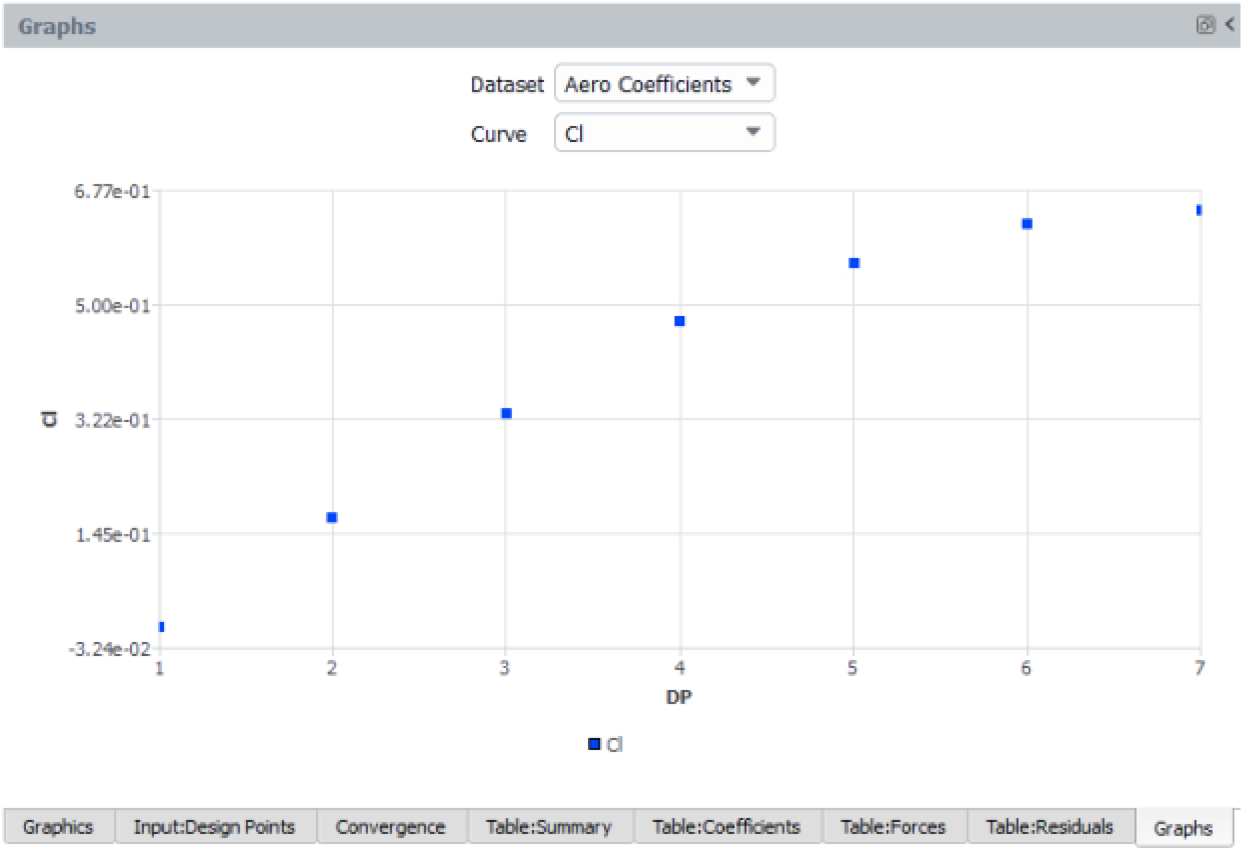


**Figure 33.17: Drag Coefficient vs. Design Point**

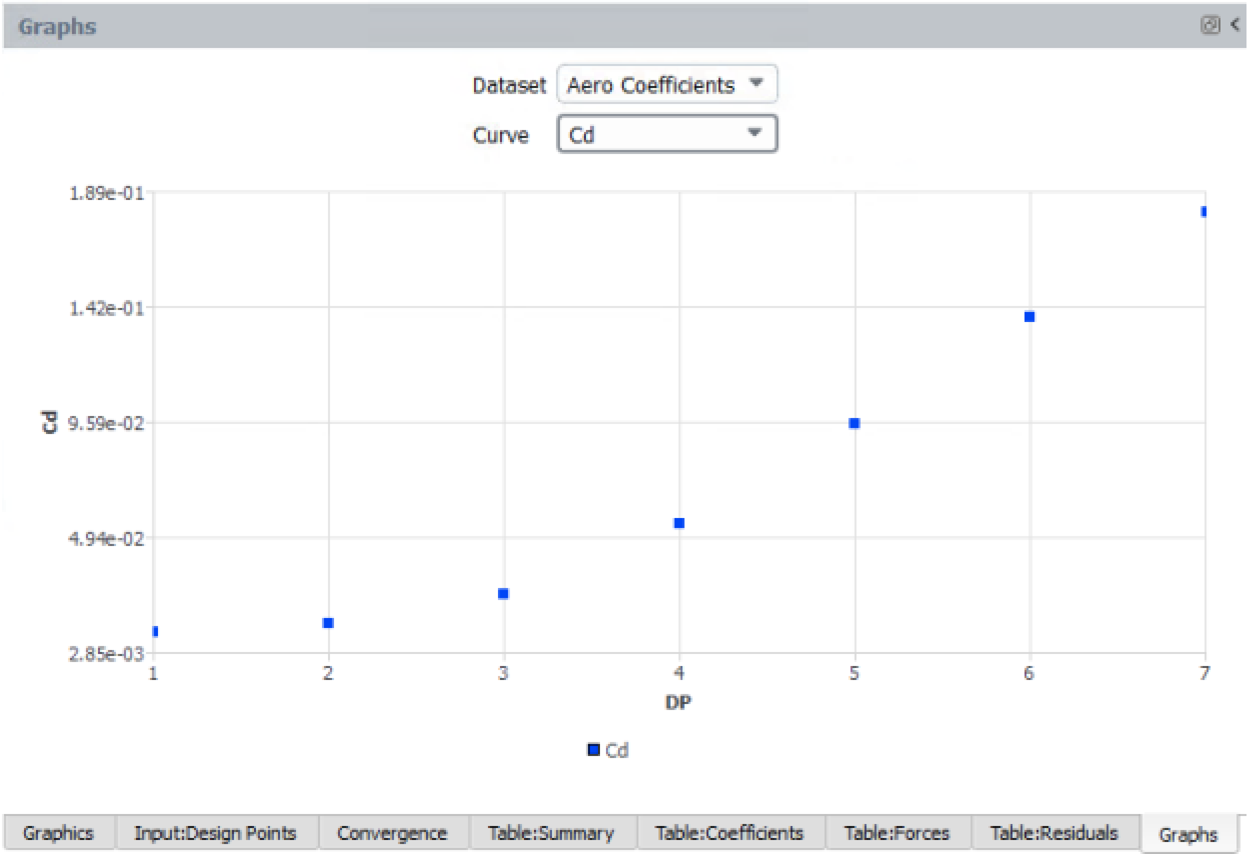
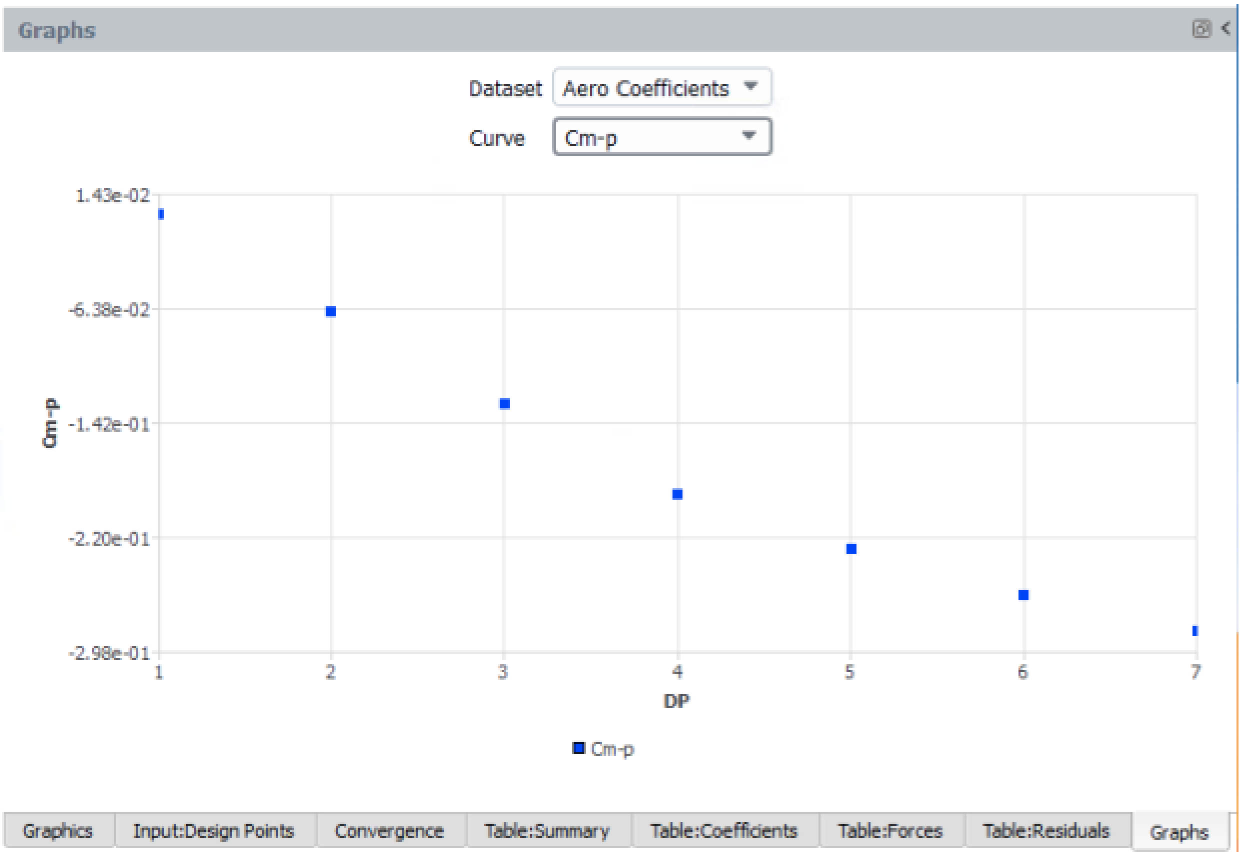
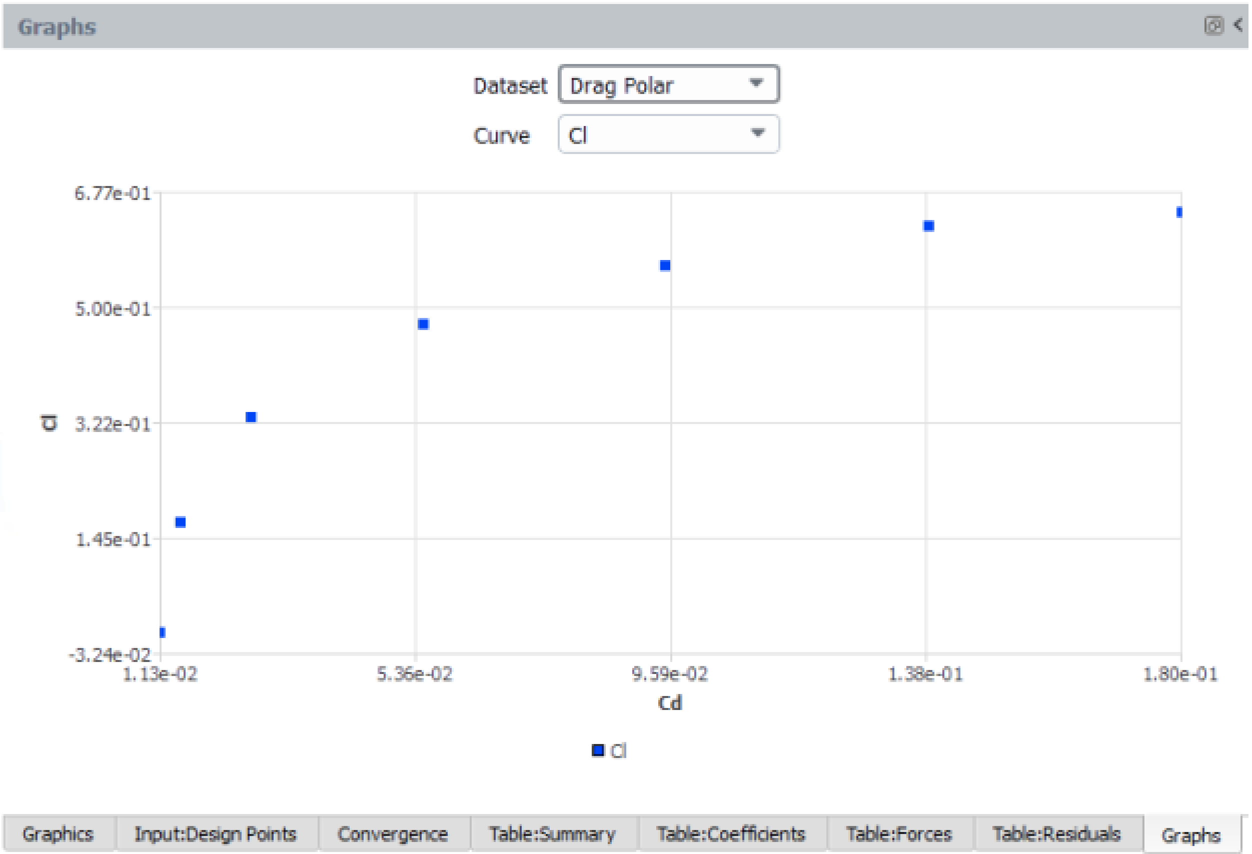


**Figure 33.18: Moment Coefficient vs. Design Point**




Click the **Show Cl Cd Plot** button. An X-Y plot of Lift Coefficient (Cl) vs. Drag Coefficient (Cd) will appear in the Graphics window. Alternatively, you can simply change **Dataset** to **Drag Polar** from the **Graphics** window to show the drag polar plot.

**Figure 33.19: Lift Coefficient vs. Drag Coefficient**




16. In the **Properties – Graphs** window, you can load and plot a reference dataset to compare with the simulation results by using the **Plot Ref. Data** function. To compare the **Cl vs DP** curve from simulation with a reference data, set **Dataset** to **Aero-Coefficients** and **Curve** to **Cl** from the **Graphs** plot window. Then click **Plot Ref. Data** and a file browser will appear. For this tutorial, we use the results from a finer mesh to demonstrate this functionality. Browse to the `reference_data` folder inside the tutorial folder and select the `ref-onera-wing-Cl-vs-dp.csv` file. The reference data will be loaded to the **Cl** curve.

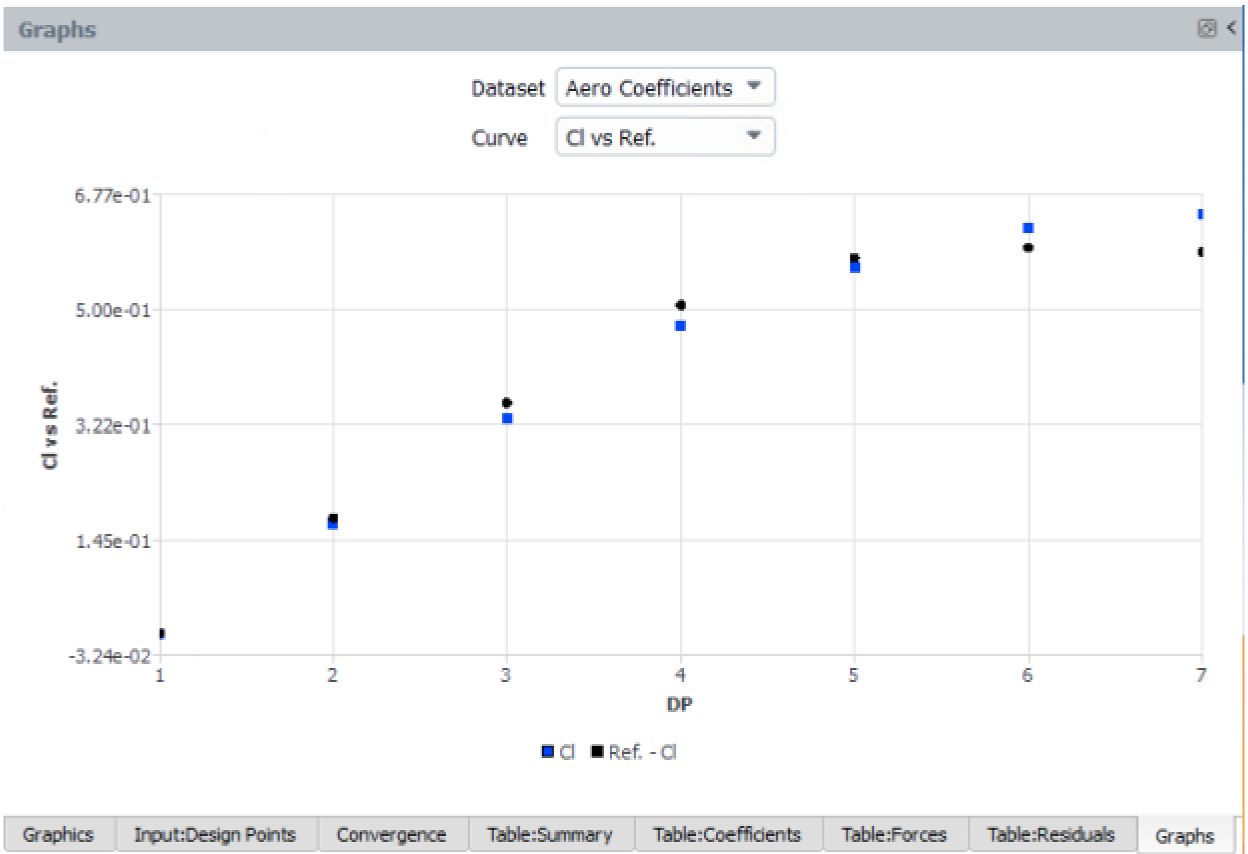
---

**Note:**

The reference file should contain row data separated by commas. The first line contains the x- and y- axis names of the plot that you want to compare to. The remaining lines contain the data values you would like to plot. The format of the file used here is shown in the image below:

```
DP,Cl
1,0.000383332
2,0.177969
3,0.35412
4,0.504979
5,0.577948
6,0.594107
7,0.587674
```

**Figure 33.20: Comparing a Reference Dataset to the Lift Coefficient Curve**




17. The **Plots** options can be used to quickly display simple 2D plots of selected design points and solution variables. Left-click on **Plots**. The **Properties - Plots** window will be displayed. Set **Surfaces** to **Walls**, **Surface Cut Normal Direction** to **Z** and set **Surface Cut Position [m]** to `0.25`. Then set **Field** to **Pressure Coefficient** and **Design Point** to **2**. Click **Plot** and the pressure coefficient on the walls of DP-2 at Z=0.25m will be plotted in the **CutPlots** window.

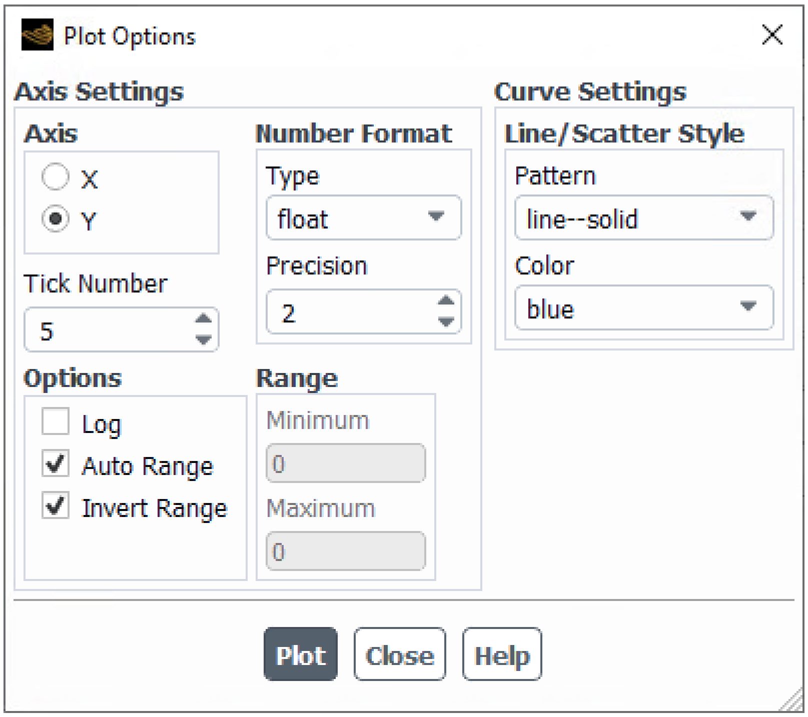
A **Plot Options** panel will appear after clicking the **Plot** button. Users can use this panel to customize plot settings for both axes.

---

**Note:**

The pressure coefficient is selected in **Field**, the range of y-axis will be reverted with higher value on the bottom and lower value on the top. The pattern and color of the curve can be modified as well.

---



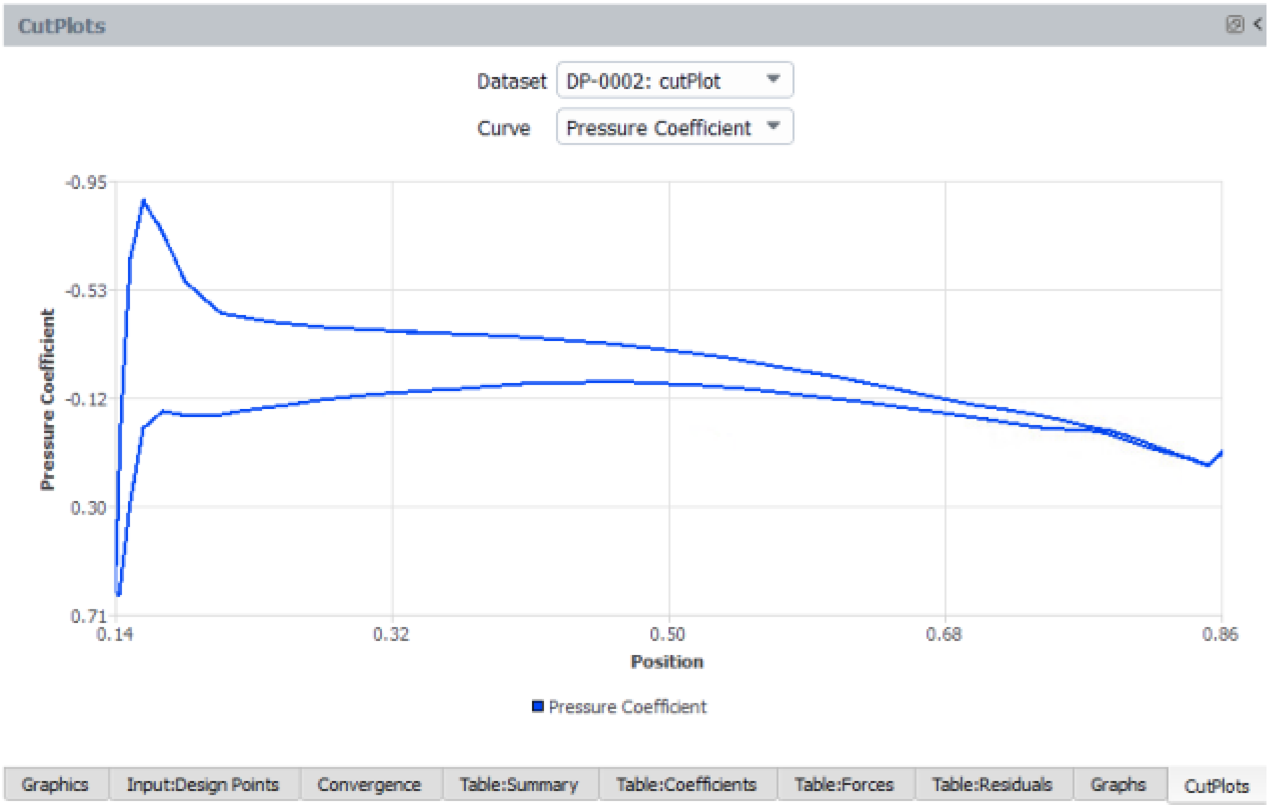

---

**Note:**

A `.csv` file will be saved in the results folder after clicking the **Plot** button, which is visible within **DP-2/Data** folder in the **Project View**. There is a known issue of the current tool to detect the discontinuities on the curves, which will be fixed in the next release.

---

**Figure 33.21: Distribution of the Wall Pressure Coefficient at Z=0.25m for DP-2**

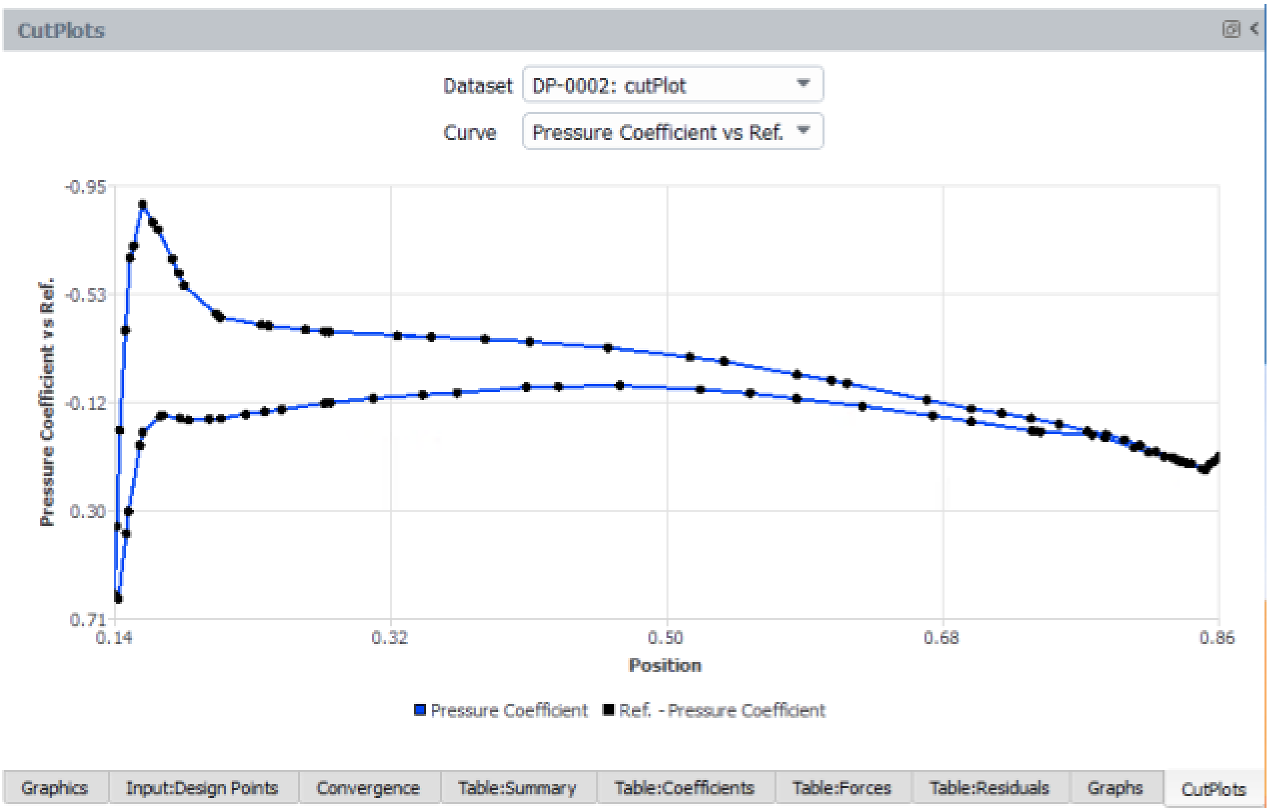



Click the **Plot Ref. Data** button to load and plot a reference dataset in the current plot. In the dialog window that appears, browse to the reference file `ref-onera-wing-Cp-2.5deg-section-0.25m.csv` in the `reference_data` folder, and click **OK**. You can use the **Save Plot** button to export the current plot to a `.png` file on disk. The reference data will be imported in the **DP-2/Data** folder in the **Project View**.

**Note:**

The same plot settings will be applied for the Fluent Aero results and black dots will be used to plot the reference data.

**Figure 33.22: Load a Reference Data to the Pressure Coefficient Plot of DP-2 at Z=0.25m**




18. The **Contours** options can be used to quickly display simple contour plots of selected design points and solution variables. Left-click on **Contours** from **Outline View** to display the **Properties - Contours** setup window.

    Set **Surfaces** to **Walls**, **Field** to **Static Pressure** and **Design Point** to 2. Click on the **Plot** button. The selected wall contour will be displayed in the **Graphics** window. In the **Graphics** window, use the mouse to set the view of the contour. Click **View...** and change **Save Name** to `onera-wing-view-1`. Click **Save** to save the current contour view. The saved view can later be applied to other contour plots.

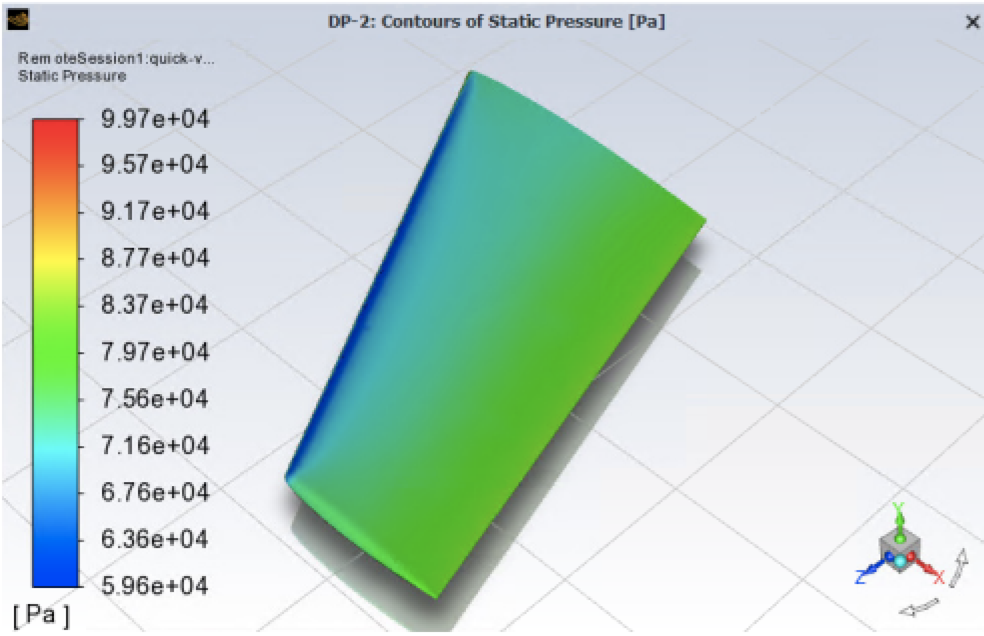
    ---

    **Note:**

    To change graphics display settings, you can go to **File** → **Preferences** → **Graphics**. For example, in the **Lighting** section, you can set **Headlight** to **On**, and set appropriate values to both **Headlight intensity** and **Ambient light intensity** to personalize the graphics object rendering.

    ---

**Figure 33.23: Wall Static Pressure Contour of Design Point 2**



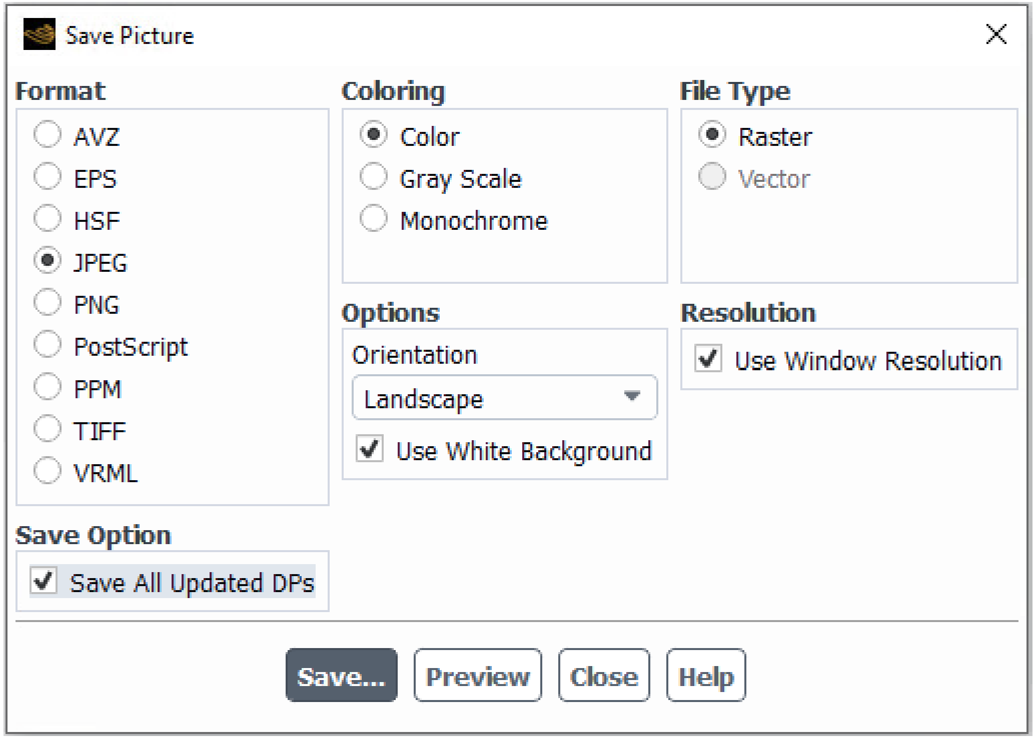

---

**Note:**

The mesh used in this tutorial is a very coarse mesh. Its sole purpose is to quickly demonstrate a typical workflow in Fluent Aero and should not be relied upon for accurate simulations. In particular, this mesh may not capture well viscous effects (such as viscous drag) or complex flow features (such as separation). This should be kept in mind while the user investigates the solutions. For more accurate simulations, a finer mesh appropriate for external aerodynamic simulations, featuring more prism layers, higher mesh surface refinement, and increased mesh density in the wake region, should be used.

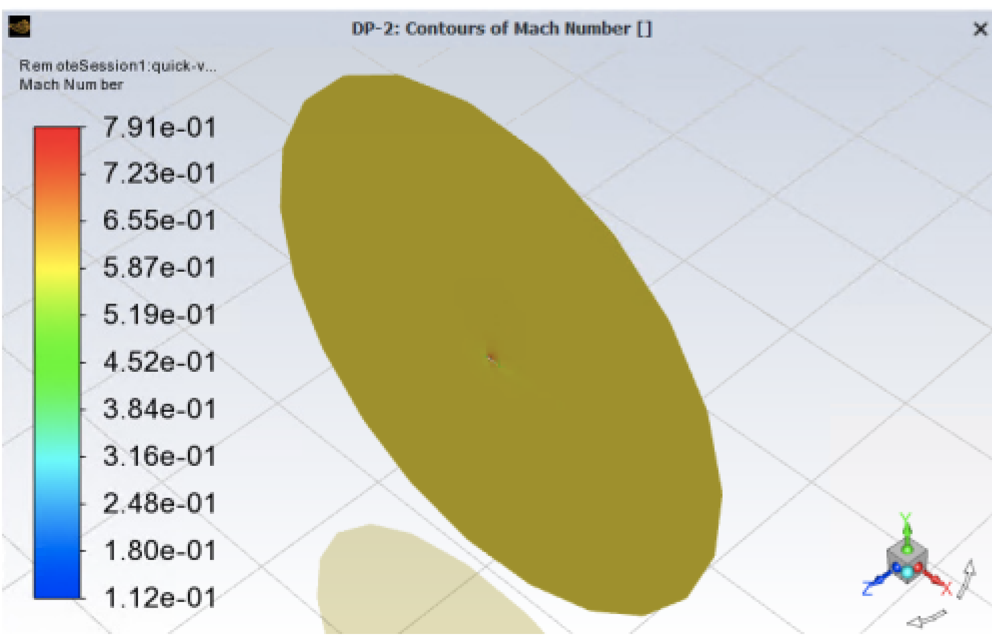
---

19. In the **Properties - Contours** area, click **Save Image...** and a **Save Picture** dialog will appear. From this dialog, check to enable **Save All Updated DPs** and set **Format** to **JPEG**. Click the **Save...** button. Set the image name to **`onera-wing-static-pressure`** from the **Select File** dialog and the contours of the static pressure for all the 7 Design Points will be saved to the **Results** folder as `onera-wing-static-pressure-DP-1` to `7.jpg`.




20. In the **Properties - Contours** area, set **Surfaces** to **Cutting Plane**, **Cutting Plane Normal Direction** to **Z**, **Cutting Plane Position [m]** to `0.2` and **Field** to **Mach Number**. Click on the **Plot** button to show the cutting plane contour in the **Graphics** window.

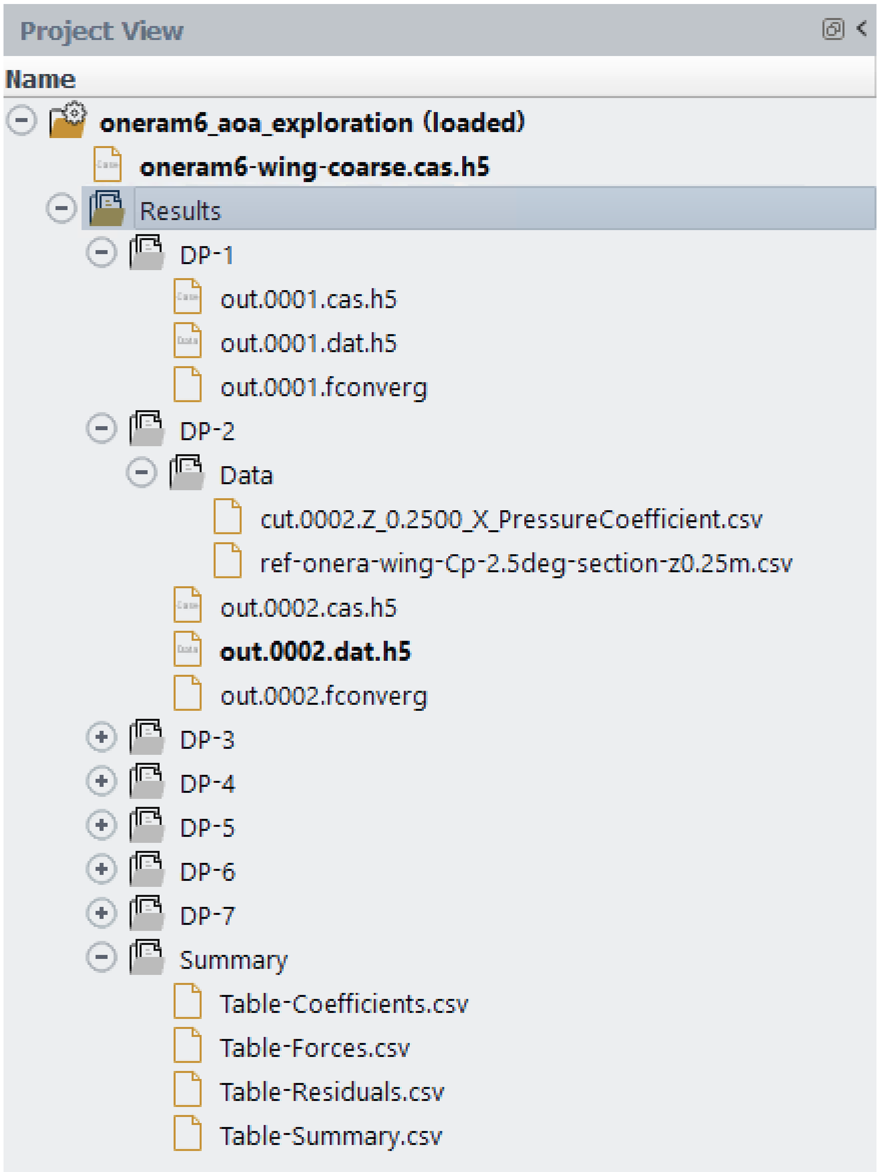
**Figure 33.24: Mach Number Cutting Plane Contour of Design Point 2**




21. In the **Project View** menu, a **Results** folder will be created after the calculation starts. A folder for each Design Point along with an associated case file, data file, and convergence file will appear inside the **Results** folder. A **Data** folder which contains the result (a `.csv` formatted file) of the

2-D plot of the pressure coefficient is created. A **Summary** folder containing the results tables (in `.csv` formatted files) is created after clicking the **Export Results Tables** button.

**Figure 33.25: Project View Panel After Calculation and Post-Processing**




These `.dat.h5` solution files can be further investigated using external post processing tools such as CFD-Post and EnSight. By left clicking on a `.dat.h5` file, a dropdown menu appears. From there, go to **View result** menu and a sub-menu appears containing **View with CFD-Post**, **View with EnSight** and **EnSight viewer (Beta)**. Right-click on the `out.00001.dat.h5` solution file and select **View result** → **View with CFD-Post** to view the result file in CFD-Post. A CFD-Post window will appear where the data file can be further post-processed.

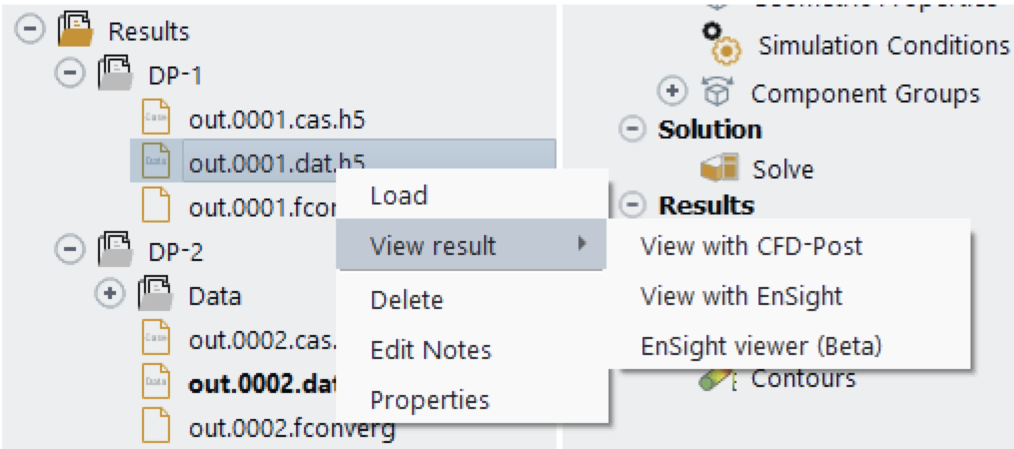
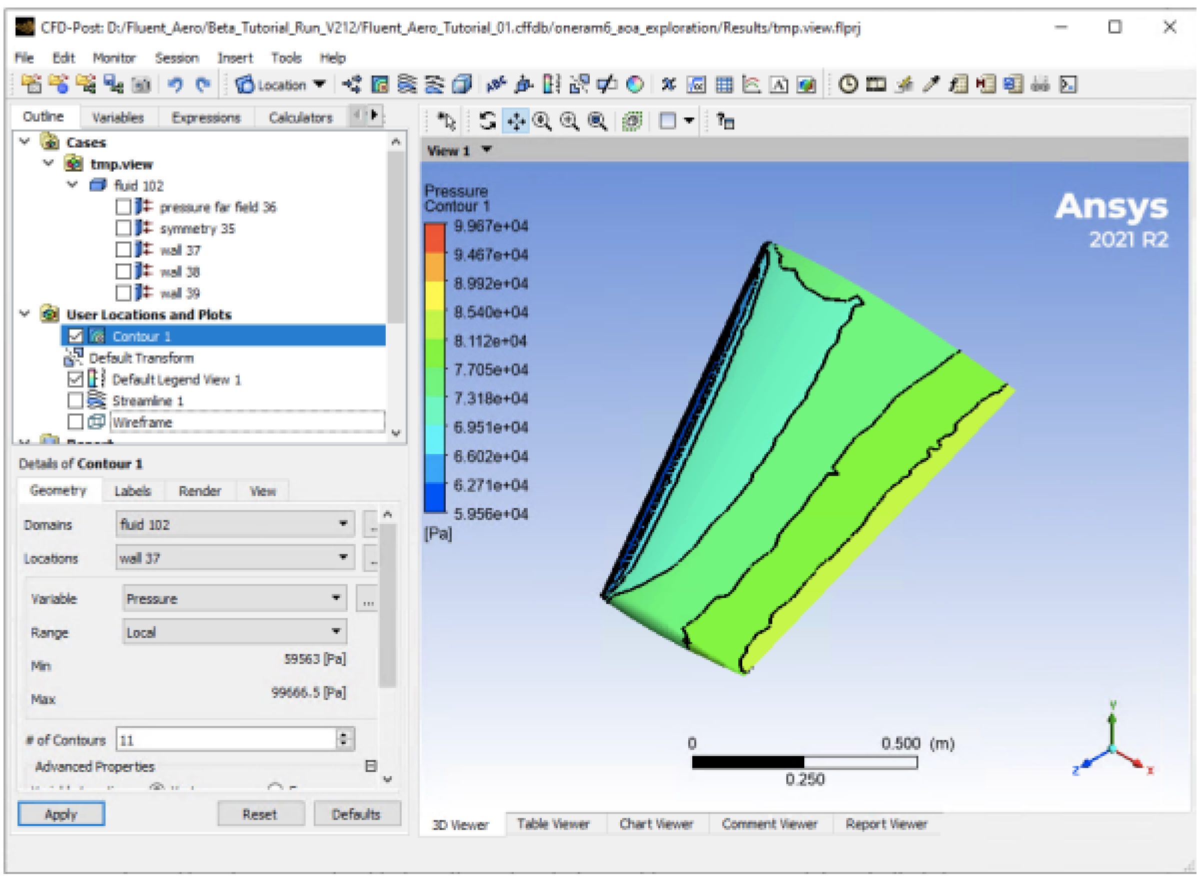



**Figure 33.26: Solution of Design Point 2 in CFD-Post**




22. Left-click on **out.0001.dat.h5** and select **View result** → **EnSight Viewer (Beta)**. EnSight appears in the **Outline View**, and an EnSight window will appear in Fluent Aero's **Graphics** area, where you can continue post-processing. Once you are finished, you can close the dataset in EnSight by

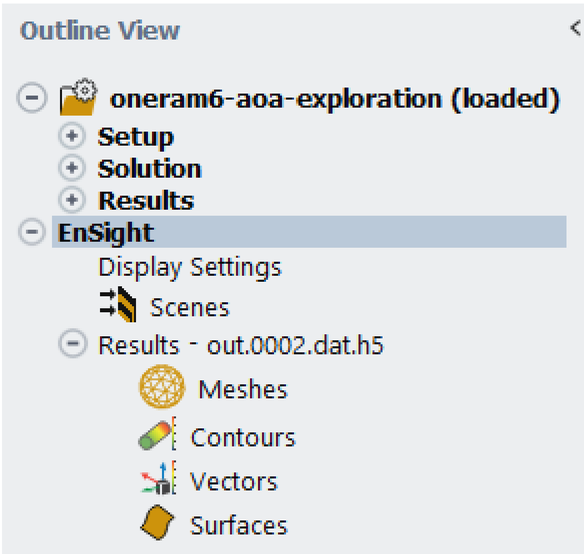
right-clicking on the dataset name (**Results – out.0002.dat.h5**) under the EnSight node in the **Outline View** and selecting **Delete**.

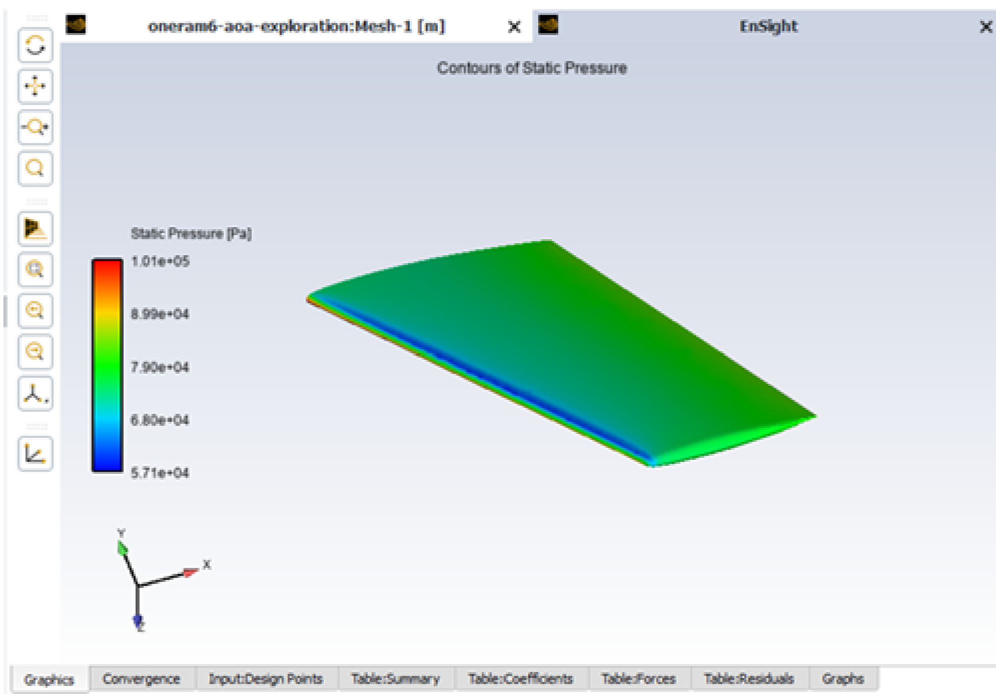
---

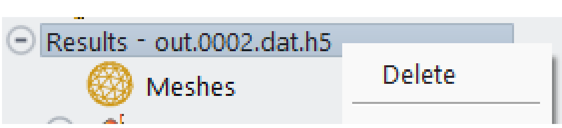
**Note:**

The **Ensight Viewer (Beta)** option is still a beta feature in development, and therefore a user may experience some limitations during use.

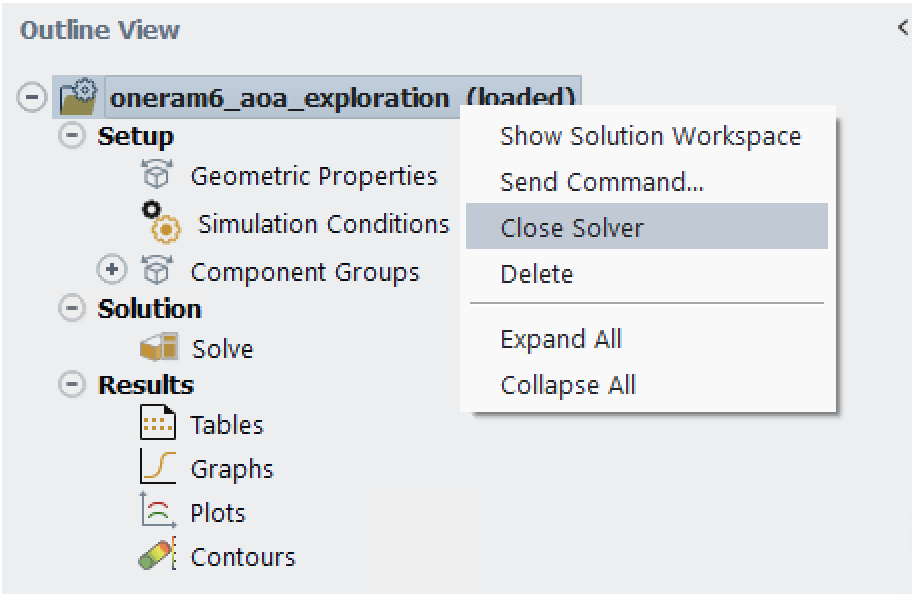
---



23. After completing the post-processing of the current simulation, you can add new simulations to the same project. At the moment, Fluent Aero can only load one simulation in the solver at the same time. Therefore, you should close the solver before adding a new simulation. Left click on **oneram6_aoa_exploration** from the **Outline View** and then select **Close Solver**. An information panel will appear to ask you if you want to save the case file or not. Click **Yes** to save the case file.




If you would like to create a new Simulation, select **Simulations** → **New Aero Workflow** from the **Project**'s top ribbon, and browse to and select a file to create a new simulation.

24. Once you have completed all your Simulations, close the project and exit Fluent Aero. From the ribbon, select **Project** → **Close** to close a project. Next, select **File** → **Exit** and the Fluent Aero Workspace will be closed.

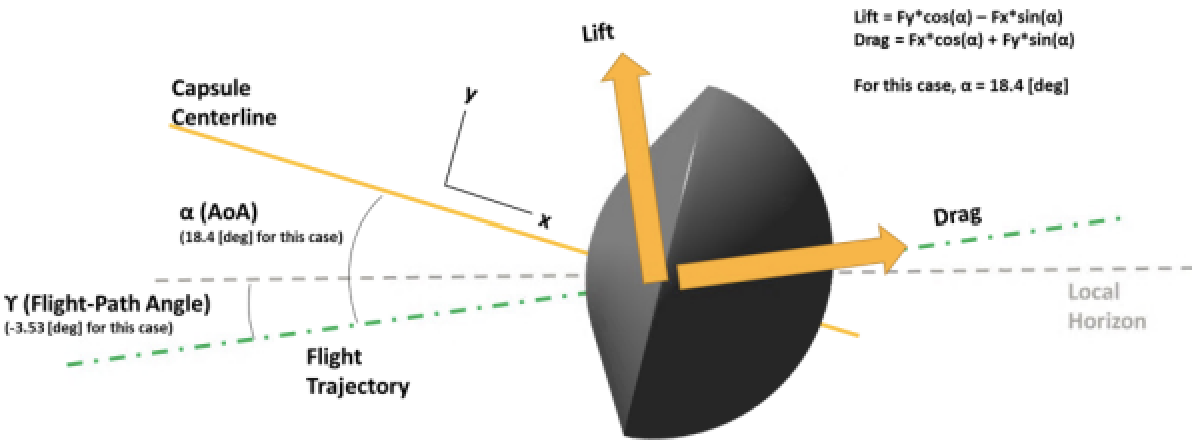
## 33.13.2. Computing Aerodynamic Coefficients and Maximum Wall Temperature on a Re-Entry Capsule at Different Altitudes Using Custom Exploration

The objective of this tutorial is to use Fluent Aero to compute the flow around a re-entry capsule in a range of hypersonic flight conditions. The flight conditions used in this tutorial correspond to conditions experienced at different altitudes during an example flight path of a reference re-entry capsule, at altitudes ranging from approximately 50-70 [km].

**Note:**

This case file shares the same mesh as the Modeling Hypersonic Flow tutorial. However, some of the solver settings and boundary conditions in the current tutorial are different from those in the Fluent tutorial.

**Figure 33.27: Re-entry Capsule Problem Specification**

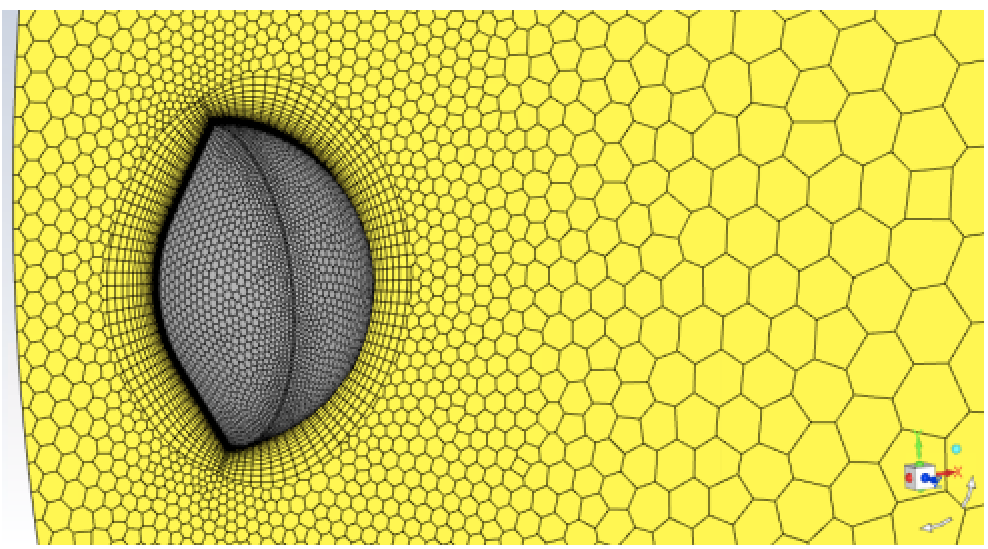



Download the `fluent_aero_tutorial.zip` file here.

Unzip `fluent_aero_tutorial.zip` to your working directory.

1. Extract the `Capsule.cas.h5` file for this tutorial. The grid is an all-poly mesh which consists of 620,000 nodes and 180,000 cells. The limits of the computational domain are defined by a pressure-farfield as inlet and pressure-outlet as outlet, and a symmetry plane in the Z direction.

   **Figure 33.28: View of the Mesh around the Capsule**

2. Launch **Fluent 2021 R2** on your computer. On the **Fluent Launcher** panel that appears, set the **Capacity Level** to **Premium** or **Enterprise**. Then enable the **Show Beta Workspaces** option and select **Aero (Beta)**. Set the number of **Solver Processes** to **4-16**. Click **Start**.

3. In the **Fluent Aero** Workspace, go to **File** → **Preferences**. In the **Preference** window, go to the **Aero** menu and ensure **Use Custom Solver Launch Settings** is disabled.

   **Note:**

   If this option is enabled, a **Fluent Launcher** window will open when loading a case file, allowing you to schedule a calculation on a server and/or to specify a different number of **Solver Processes** to use in your simulation. If this setting is disabled, Fluent Aero will load the case file in a Solver session on your local machine with the same number of CPUs used to open Fluent Aero.

   Also, in the **Preferences** → **Aero** menu, make sure that **Show Solution Workspace** and **Enable Solution Workspace Graphics** are disabled. Click **OK**.

4. When Fluent Aero first opens, the Project tab will be displayed by default. In the Project's top ribbon panel, select **Project** → **New...** and enter **`Fluent_Aero_Tutorial_02`** to create a new **Project** folder.

5. In the **Project**'s top ribbon, select **Simulations** → **New Aero Workflow**, and browse to and select the `Capsule.cas.h5` file. A **New simulation** window will appear. Enter the **Name of the new simulation** as **`Capsule_custom_exploration`**, and check to enable **Load in solver**.

6. The case file will be opened and a background Solver session will be loaded. A new Simulation folder will be created in your **Project** folder, and the `Capsule.cas.h5` file will be imported.

   After the `.cas.h5` file has been successfully loaded, a new **Outline View** tree appears under **Capsule_custom_exploration (loaded)**.

   While importing, Fluent Aero will search for and find the pressure-far-field zone that defines the external boundary of the domain. It's presence will cause Fluent Aero to determine that this case is using a **Freestream Domain Type**, and the following message will be reported in the console.

```
Console

-------------------------------------------------------------
-                   Simulation Domain Type                  -
-------------------------------------------------------------
Fluent Aero has been set to use a Freestream domain type.

- The following boundary zones have been detected:
  > Freestream Inlet (type: pressure-far-field)
    - inflow
  > Freestream Outlet, not required (type: pressure-outlet):
    - outflow

- The Domain Type will be set to Freestream,
  and the group will be created in Component Groups
-------------------------------------------------------------
```

7. From the **Outline View** window, go to **Geometric Properties**. At the top of the **Properties – Geometric Properties** window, notice that the **Domain Type** has been automatically set to **Freestream**.

   Define the orientation of the geometry within the computational domain. This is used to compute the aerodynamic forces. In this manner, set **Lift Direction at AoA = 0 degree** to **Y+** and set **Drag Direction at AoA = 0 degree** to **X+**. The **Pitching Moment Direction** will automatically set to **Z-**. Set the **Moment Center X-**, **Y-** and **Z-Position [m]** to `0`, `0`, and `0`, respectively.

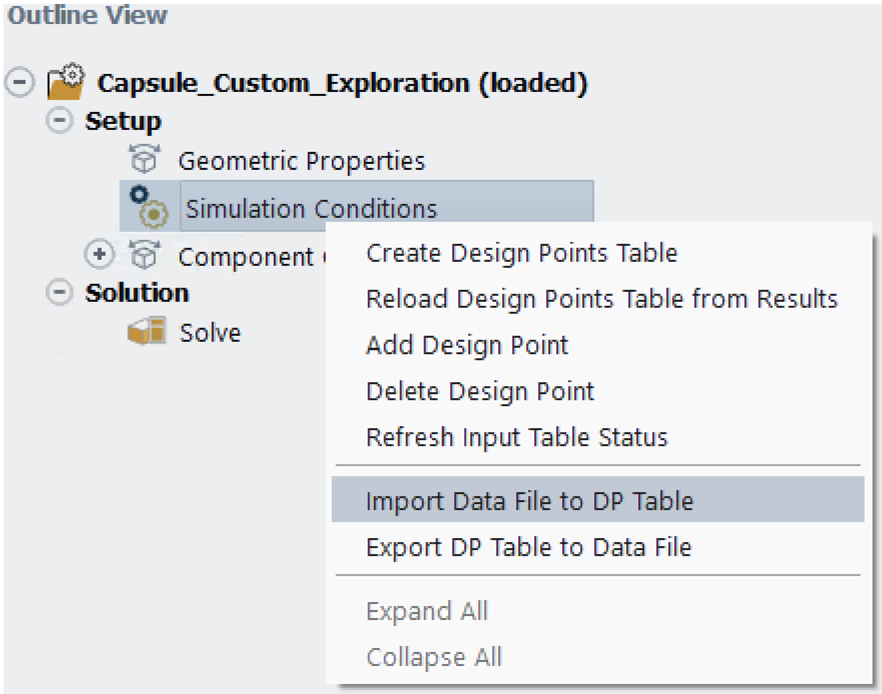
   Set the **Reference Length [m]** to `3.5` and the **Reference area [m^2]** to `3.0129`. Alternatively, the user can use **Compute Projected Area** to calculate the reference area.

8. In the **Setup** tree, select **Simulation Conditions**. In the **Properties – Simulation Conditions** window, set the **Number of Design Points** to **3**. In **Flow Speed** section, set **Parameter** to **Mach** and **Distribution** to **Custom**. For **Flow Direction**, set **Parameter** to **AoA**, **Distribution: Angle of Attack** to **Constant** and **Angle of Attack [degrees]** to `18.4`. In the **Pressure and Temperature** section, set **Parameter** to **Altitude** and **Distribution** to **Custom**.

   An empty **Input:Design Points** table will be created with 3 rows, one for each Design Point. The user can manually fill the **Input:Design Points** table by clicking on each entry cell and entering a value. Notice that there are 6 columns in the table. The first column specifies the **Design Point** number, and cannot be edited. The second and third columns are for specifying the variable inputs of **Mach Number** and **Altitude [m]**, respectively. Since **Altitude** was selected under **Pressure and Temperature**, **Pressure [Pa]** and **Temperature [K]** are also shown in the table. These columns cannot be edited, as they will be automatically calculated and filled based on the Altitude input, which uses the International Standard Atmosphere. The final column lists the Status of each Design Point calculation.

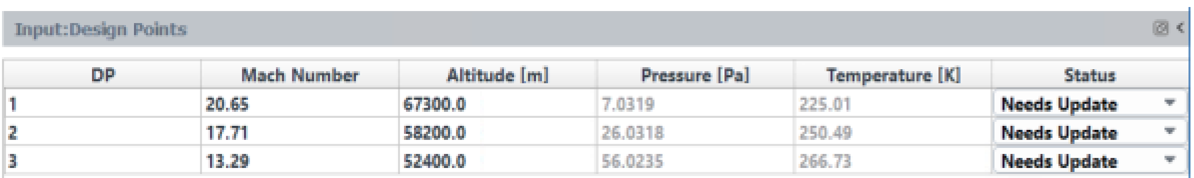
   Begin to manually fill the table by clicking on the **Mach Number** cell of Design Point 1, and enter `20.65`. Next, click on the **Altitude [m]** cell and enter `67300`. Notice that the **Pressure [Pa]** and **Temperature [K]** cells are automatically filled with **7.0319** and **225.01**, respectively.

   Fill the remainder of the table by importing the data from a `.csv` file. Right-click the **Simulation Conditions** and select **Import Data File to DP Table** from the drop-down list of commands.




Navigate to and load the `Capsule_Input_3_Design_Points.csv` from the `FLU-ENT_AERO_TUTORIALS` folder and the remainder of the **Input:Design Points** Table will be filled.

**Figure 33.29: Input:Design Points Table of a Custom Exploration with 3 Design Points**



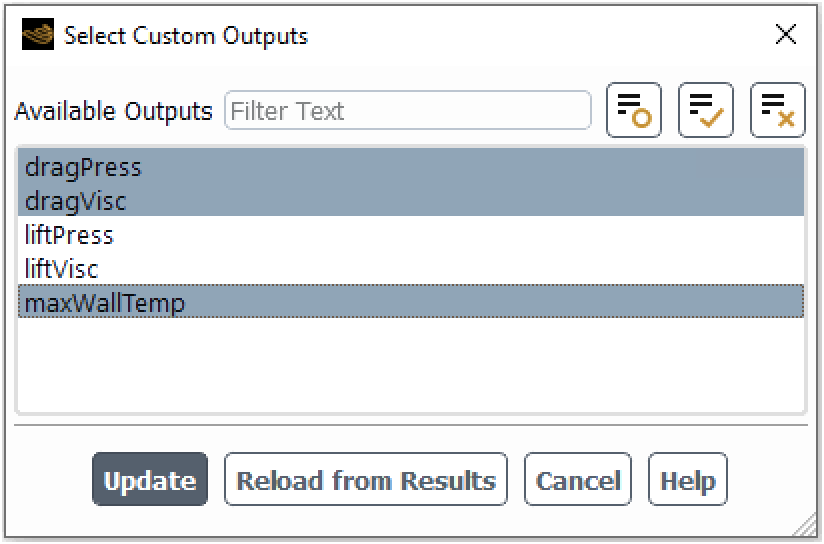
Input:Design Points

| DP | Mach Number | Altitude [m] | Pressure [Pa] | Temperature [K] | Status |
|---|---|---|---|---|---|
| 1 | 20.65 | 67300.0 | 7.0319 | 225.01 | Needs Update |
| 2 | 17.71 | 58200.0 | 26.0318 | 250.49 | Needs Update |
| 3 | 13.29 | 52400.0 | 56.0235 | 266.73 | Needs Update |

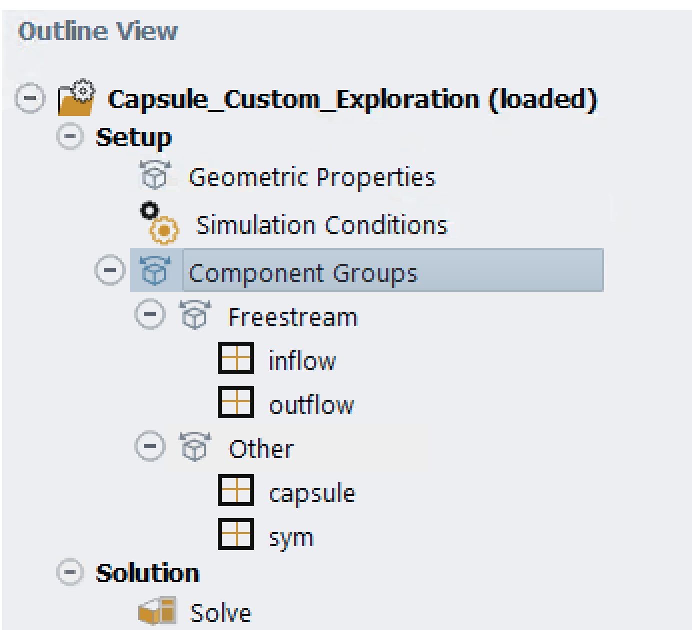
Notice that the **Status** of each **Design Point** is currently set to **Needs Update**, because they have not yet been calculated.

9. From the **Properties - Simulation Conditions** window, enable **Use Custom Output Parameters**. A selection panel will appear which contains a list of pre-defined custom-output variables. Select **dragPress** (pressure induced drag force), **dragVisc** (wall shear stress induced drag force), and **maxWallTemp** (maximum wall temperature), and click **Update**. Fluent Aero will create the selected variables in the Solver and the results of these output parameters will be shown in a table at the end of the calculation. You may refer to the Fluent Aero manual for more details about the functionality of the custom input/output parameters.

**Figure 33.30: Select Custom Outputs**




10. Go to **Component Groups**. Two default **Component Groups** have been created after the simulation is loaded. The **Freestream** group contains a pressure-far-field and a pressure-outlet zone that define the external boundary of the domain, and is where the Freestream atmospheric flight conditions for each Design Point are defined as boundary conditions. The **Other** group contains the remainder of the boundary zones which are the walls of re-entry capsule and a symmetry plane of the simulation domain.




**Note:**

When using more complex geometries, it can be useful to re-organize zones and create additional **Component Groups** using the **Component Manager**. This type of procedure is described in detail in Introduction to Aircraft Component Groups and Computing Aerodynamic Coefficients on an Aircraft at Different Flight Altitudes and Engine Regimes (p. 611). However, this is not necessary in this simple case. Therefore, continue to the next step.

11. Go to **Solve**. Set **Iterations** to **`1000`**. Check to enable **Show Advanced Settings**. Set the **Solver Type** to **Density based**. Set the **Turbulence Model** to **K-Omega SST** and the **Two Temperature Model** to **Enabled**. Change the **Flow Range** from **Automatic** to **Hypersonic** and set **First to Second Order Blending** to **`1`**.

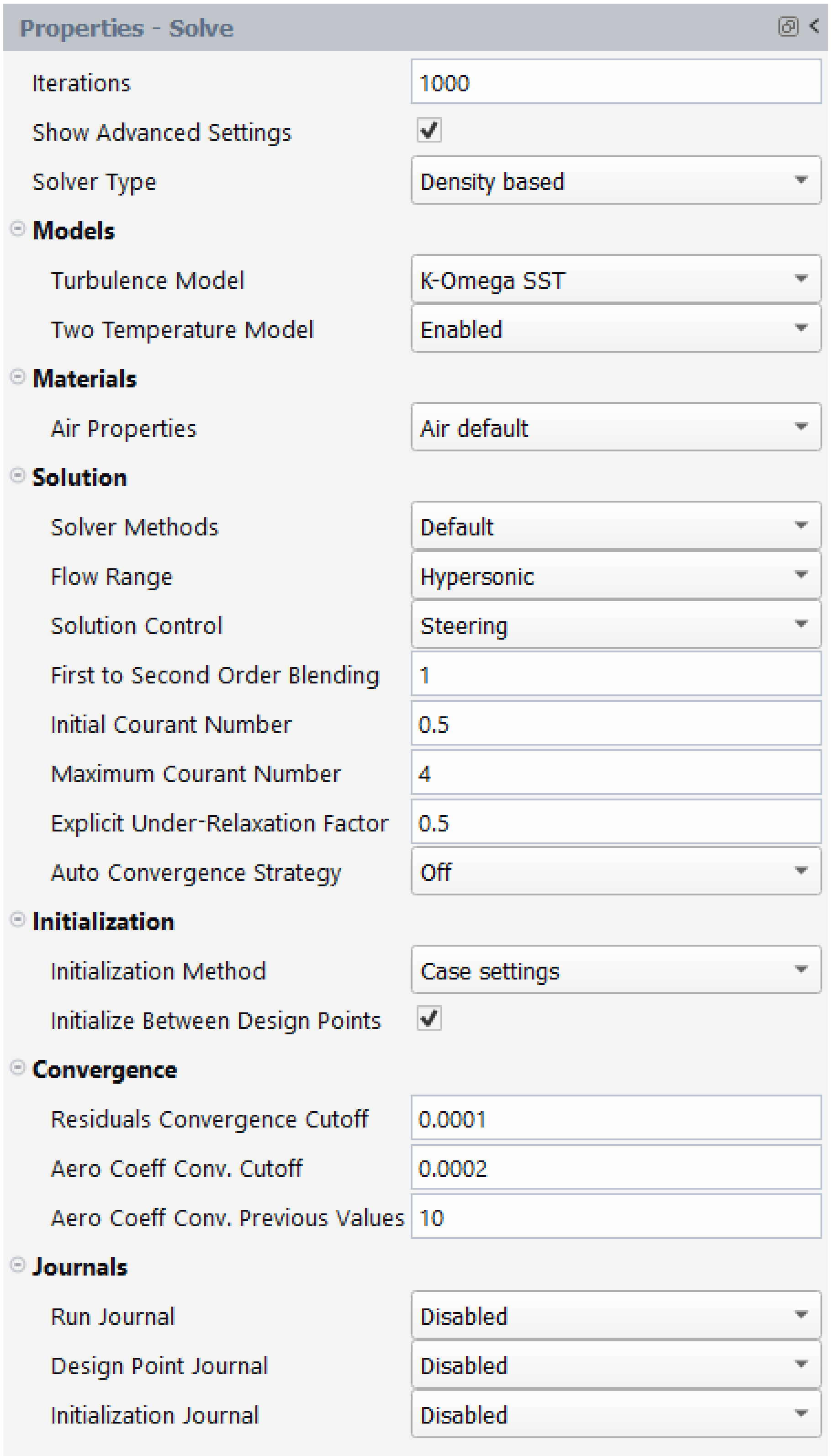
    Set **Initialization Method** to **Case settings**.

    **Note:**

    By setting **Initialization Method** to **Case setttings**, Fluent Aero will use the initialization method that has been setup in the initial `.cas` file, or from within the **Solution Workspace** using the **Workspace** → **Solution** command. In this example, the `Capsule.cas.h5` file already contains the appropriate custom initialization settings that should be used for this case.

    The **Solver Strategy** is set to **Steering**. This strategy differs from what is used in Modeling Hypersonic Flow within the Fluent Tutorials.

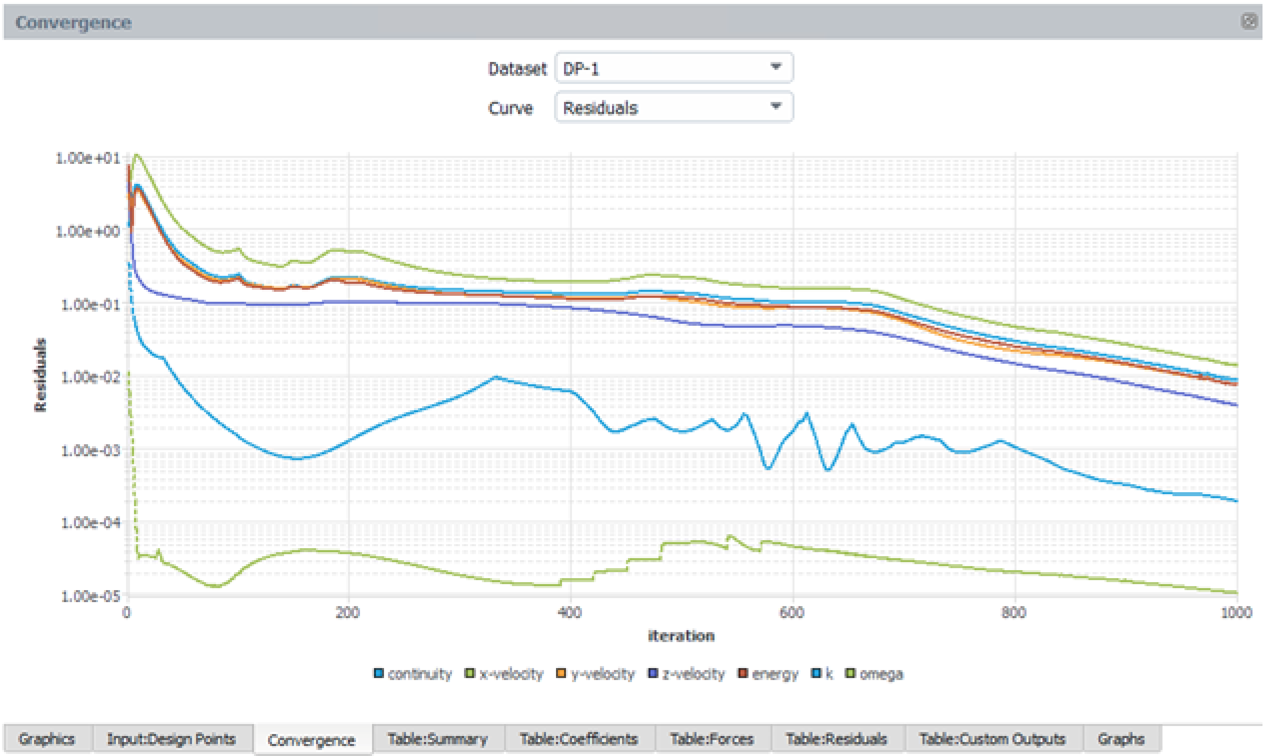
**Figure 33.31: Set Advanced Settings in the Properties – Solve Window**




12. Click the **Calculate** button at the bottom of the **Properties - Solve** panel. The simulation will first initialize using the flight conditions of DP-1 and the custom FMG initialization settings and then

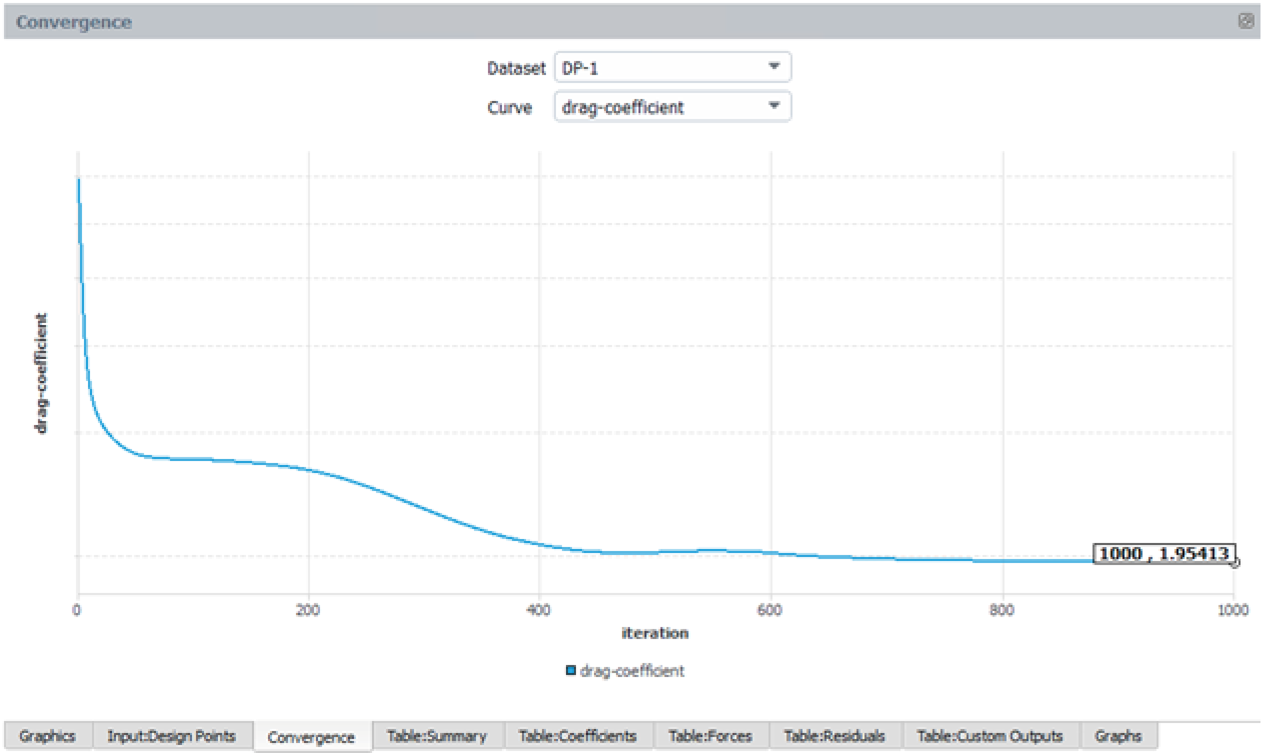
DP-1 will begin to iterate. The residuals plot of DP-1 will appear in the **Convergence** window located on the right of the screen.

**Figure 33.32: Convergence of the Residuals for Design Point 1**




In the **Convergence** window, set **Dataset** to **DP-1** and **Curve** to **drag-coefficient**. The evolution of the drag coefficient for DP-1 will be displayed. The user can query the value of the drag coefficient by left-clicking on the curve.

**Figure 33.33: Convergence History of the Drag Coefficient for Design Point 1**




After 1000 iterations, the calculation of DP-1 is complete, the status column in row 1 of the **Input: Design Points** table will be set to **Updated**, the results data file (`.dat[.h5]`) will be saved, and calculation of DP-2 will begin.

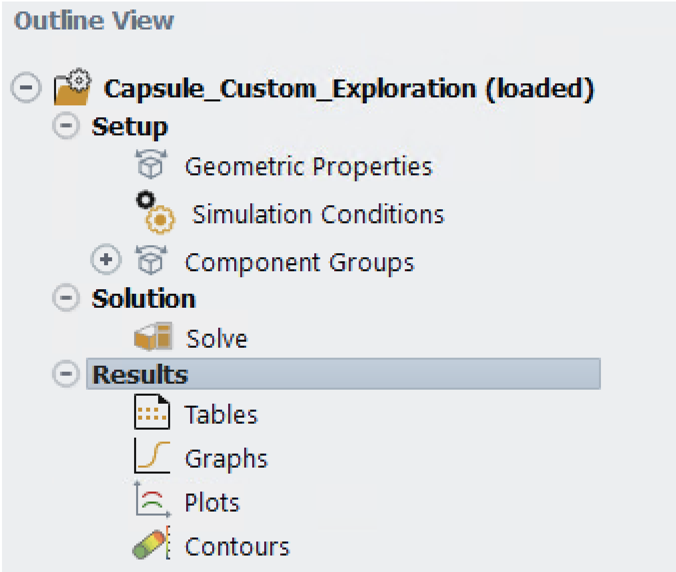
13. After all the Design Points have been updated, the status of the **Input: Design Points** table will be set to **Updated** for all the Design Points.

**Figure 33.34: Input: Design Points Table after Calculation**

Input:Design Points

| DP | Mach Number | Altitude [m] | Pressure [Pa] | Temperature [K] | Status |
|---|---|---|---|---|---|
| 1 | 20.65 | 67300.0 | 7.0319 | 225.01 | Updated |
| 2 | 17.71 | 58200.0 | 26.0318 | 250.49 | Updated |
| 3 | 13.29 | 52400.0 | 56.0235 | 266.73 | Updated |

14. A **Results** section will be displayed in the **Outline View** after the simulation starts. This allows a user to quickly postprocess all Design Point solutions, by obtaining aerodynamic coefficient plots, creating contour plots of the solution fields, comparing solution fields to experimental data, and more.




15. The first element in the **Results** section is **Tables**. In the current tutorial, 5 different **Tables** will be automatically created in the Graphic window area when the calculation is complete.

> **Note:**
>
> You can also click on **Tables** → **Export Results Tables** button and the results tables will be exported to the **Results** folder of the current simulation and they will be visible in the **Project View** under the **Summary** folder.

Click on the **Table: Summary**, **Table: Coefficients**, **Table:Forces**, **Table: Residuals**, and **Table: Custom Outputs** tabs at the bottom of the **Graphics** window to reveal each table.

- **Table: Summary** summarizes the flight conditions and convergence information for each Design Point.

> **Note:**
>
> In the current simulation, while we can see from the convergence curves that the convergence of lift and drag was reasonable. However, you may notice in the final column of this table that all 3 Design Points have not met the default convergence criteria that was set in **Solve** → **Show Advanced Properties** → **Convergence**. To have the **Criteria Met?** column set to **yes**, a user could relax the convergence criteria by increasing the values, or calculate for more iterations until the default criteria is met.

**Figure 33.35: Summary Table of the Flight Conditions and Convergence Information**

Table:Summary

| | DP | Mach | AoA [deg] | AoS [deg] | P [Pa] | T [K] | Reynolds | Avg Coeff Conv | Avg Residual | Criteria Met? |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 20.65 | 18.4 | 0.0 | 7.0319 | 225.01 | 7.6839e+04 | 4.8960e-05 | 5.3042e-03 | no |
| 2 | 2 | 17.71 | 18.4 | 0.0 | 26.0318 | 250.49 | 2.9224e+05 | 2.4967e-05 | 4.5531e-03 | no |
| 3 | 3 | 13.29 | 18.4 | 0.0 | 56.0235 | 266.73 | 5.2520e+05 | 2.3186e-05 | 2.0738e-03 | no |

- **Table: Coefficients** contains the lift, drag and moment coefficients. The last three columns show the convergence information of the lift and drag coefficients and the maximum value among the three moment coefficients.

**Figure 33.36: Results Table of the Aerodynamic Coefficients**

Table:Coefficients

| | DP | Cl | Cd | Cm-y | Cm-p | Cm-r | Cl Conv. | Cd Conv. | Max. Cm Conv. |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | -3.9237e-01 | 1.9541e+00 | -8.6349e-01 | -2.7644e-01 | 6.2691e-03 | 2.2206e-05 | 7.5715e-05 | 1.7197e-03 |
| 2 | 2 | -4.0272e-01 | 1.9446e+00 | -8.5817e-01 | -2.6634e-01 | 3.3563e-03 | 3.1043e-06 | 4.6830e-05 | 1.8554e-03 |
| 3 | 3 | -4.1198e-01 | 1.9502e+00 | -8.5064e-01 | -2.5914e-01 | 2.2817e-03 | 3.1898e-06 | 4.3182e-05 | 1.7825e-03 |

- **Table: Forces** contains the lift, drag and moment dimensional forces.

**Figure 33.37: Results Table of the Aerodynamic Forces**

Table:Forces

| | DP | Lift [N] | Drag [N] | Mom. Yaw [N.m] | Mom. Pitch [N.m] | Mom. Roll [N.m] |
|---|---|---|---|---|---|---|
| 1 | 1 | -2.4814e+03 | 1.2358e+04 | -1.9113e+04 | -6.1188e+03 | 1.3876e+02 |
| 2 | 2 | -6.9348e+03 | 3.3486e+04 | -5.1721e+04 | -1.6052e+04 | 2.0228e+02 |
| 3 | 3 | -8.5976e+03 | 4.0700e+04 | -6.2132e+04 | -1.8928e+04 | 1.6666e+02 |

- **Table: Residuals** summarizes the final residuals as well as the number of iterations run for each Design Point.

**Figure 33.38: Results Table of Final Residuals**

Table:Residuals

| | DP | iteration | continuity | x-velocity | y-velocity | z-velocity | energy | k | omega | vib-ele-energy |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1000 | 8.7716e-03 | 1.3885e-02 | 7.6826e-03 | 3.9859e-03 | 7.6161e-03 | 1.9408e-04 | 1.0596e-05 | 2.8795e-04 |
| 2 | 2 | 1000 | 7.6175e-03 | 1.1999e-02 | 6.5473e-03 | 3.2253e-03 | 6.7193e-03 | 1.1795e-05 | 1.4107e-05 | 2.9062e-04 |
| 3 | 3 | 1000 | 3.4209e-03 | 5.1902e-03 | 3.0970e-03 | 1.6536e-03 | 2.9609e-03 | 1.2804e-05 | 8.3067e-06 | 2.4638e-04 |

- **Table: Custom Outputs** shows the final values of the custom outputs.

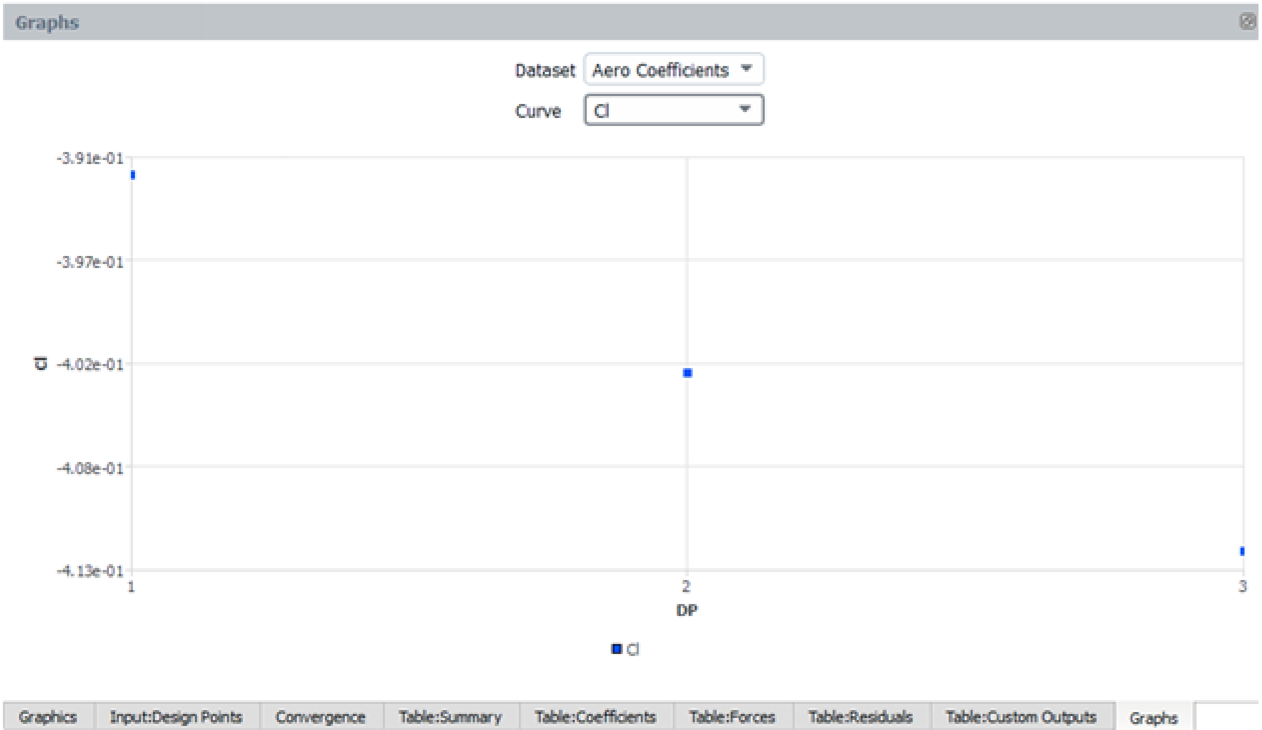
**Figure 33.39: Results Table of Custom Outputs**

Table:Custom Outputs

| | DP | dragPress | dragVisc | maxWallTemp |
|---|---|---|---|---|
| 1 | 1 | 12105.473 | 252.36934 | 19144.033 |
| 2 | 2 | 33136.962 | 350.58068 | 16908.406 |
| 3 | 3 | 40421.748 | 282.74573 | 10388.129 |

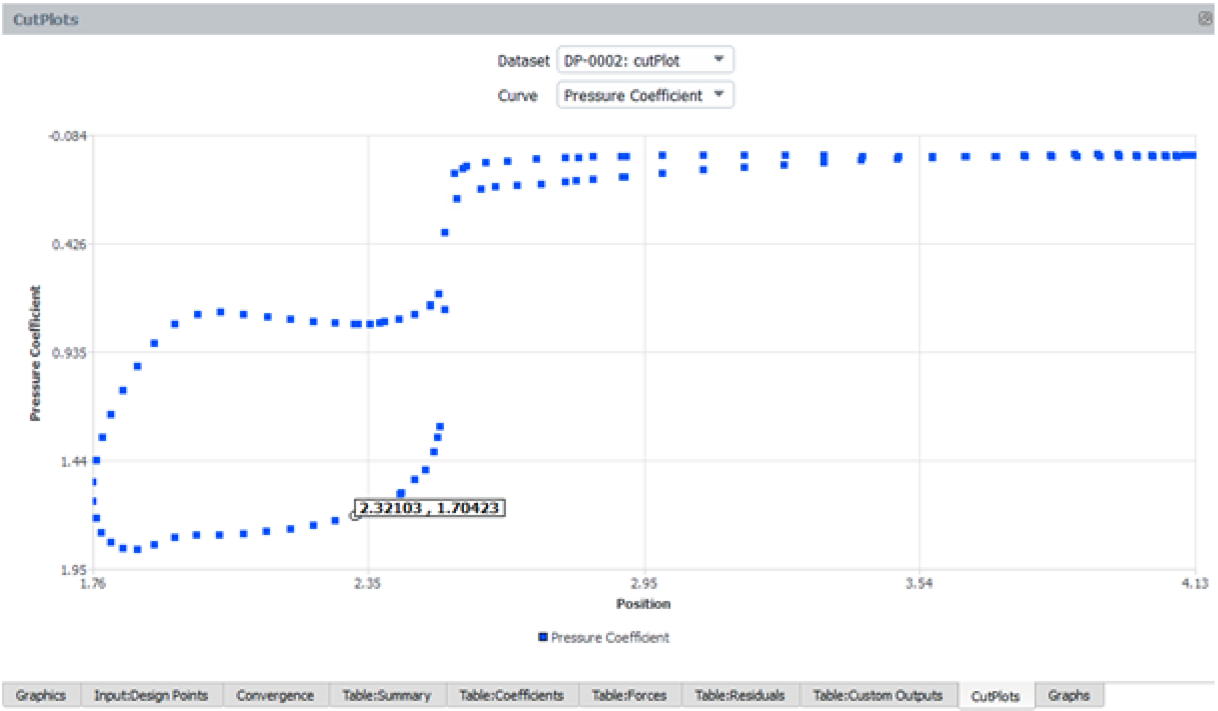
Click on **Graphs** in the **Outline View** to show the plots of the aerodynamic coefficients defined in Fluent Aero. At the bottom of the **Properties - Graphs** window, click **Plot Coefficients**. An X-Y plot of Lift Coefficient (Cl) vs. Design Point (DP) will appear in the **Graphics** window. The Drag and Moment Coefficients can be shown by selecting **Cd** and **Cm-y/p/r** from the **Curve** selection drop-down list.

**Figure 33.40: Lift Coefficient vs. Design Point**



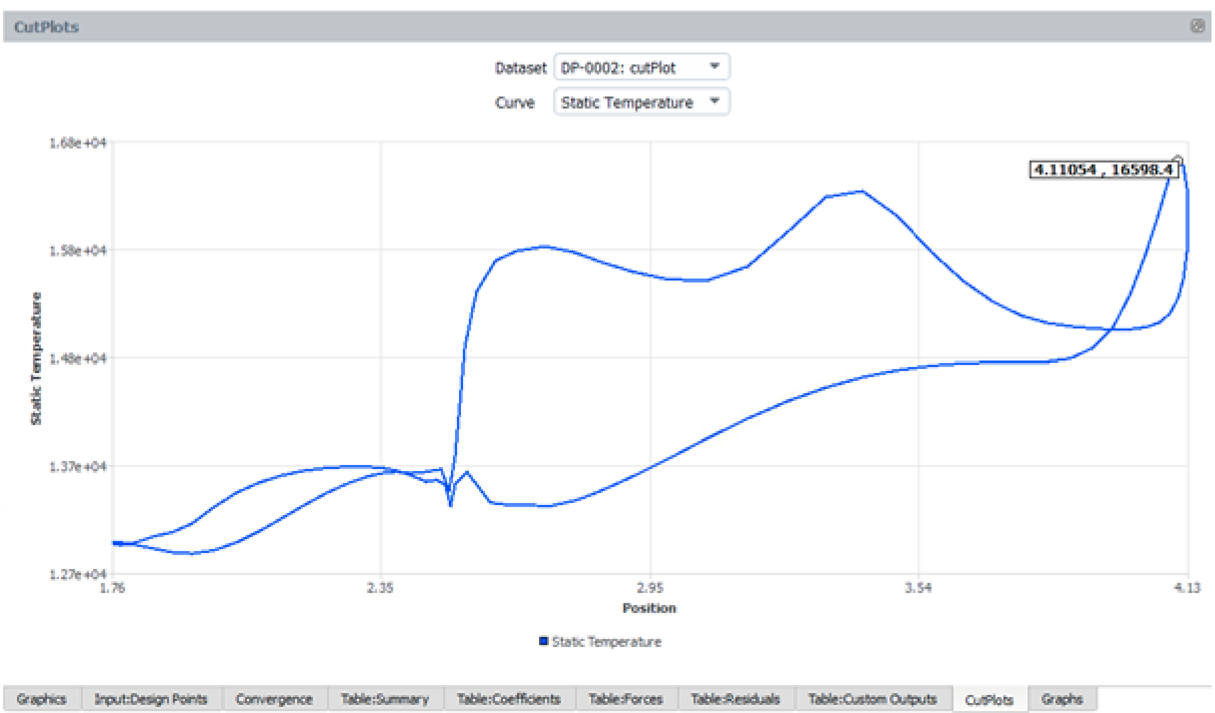

16. The **Plots** options can be used to quickly display simple 2D plots of selected design points and solution variables. Left-click on **Plots**. The **Properties - Plots** window will be displayed. Set **Surfaces** to **Walls** and set **Surface Cut Position [m]** to `0.1`. Then set **Field** to **Pressure Coefficient** and **Design Point** to **2**. Click **Plot**. Since the solution for DP-2 at Z=0.1m will be plotted in the **CutPlots** window. A cut plot using default plot options will be created and a popup panel which allows the user to customize certain plot options will appear. If the field is set to **Pressure Coefficient**, the **Invert Range** option for the y-axis will be enabled. From the **Plot Options** panel, set the **Pattern in Curve Settings** to **scatter--squares**. Click **Plot**. The 2d cut plot in the graphic window will be updated using these new settings.

**Figure 33.41: Distribution of the Wall Pressure Coefficient at Z = 0.1m for DP-2**



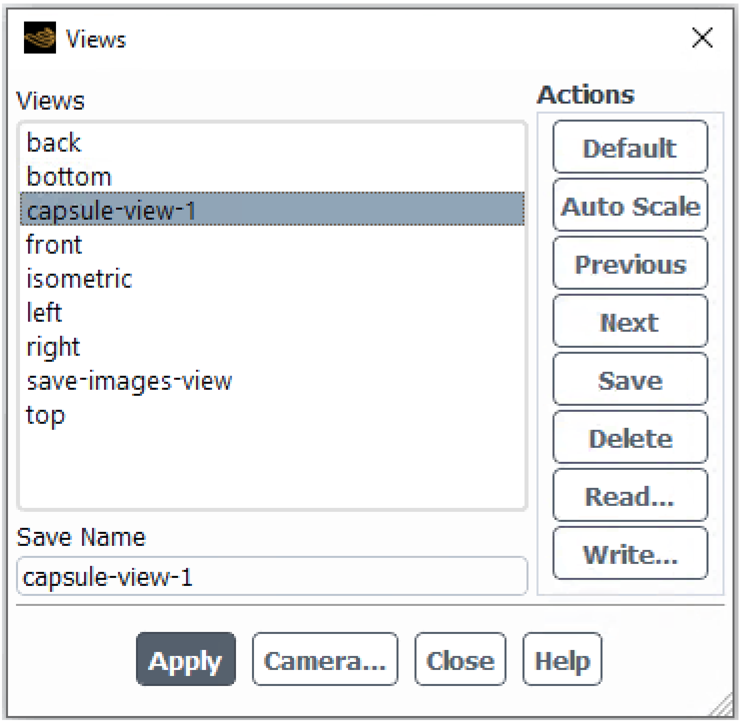

Then set **Field** to **Static Temperature** and click **Show Cut Plot** to show the walls static temperature of DP-2 at Z=0.1m.

**Figure 33.42: Distribution of the Wall Static Temperature at Z = 0.1m for DP-2**



17. The **Contours** options can be used to quickly display simple contour plots of selected design points and solution variables. Left-click on **Contours** from the **Outline View** to display the **Properties – Contours** setup window.

    Set **Surfaces** to **Walls**, **Field** to **Static Temperature** (and then **Static Pressure**) and **Design Point** to **3**. Uncheck to disable **Auto-Compute Range** and set the **Minimum** and **Maximum Value** to **`8200`** and **`10200`** respectively. Check to enable **Draw Mesh**. Click on the **Plot Wall Contour** button. The selected wall contour will be displayed in the **Graphics** window. In the **Graphics** window, use the mouse to set the view of the contour. Click **View...** and change **Save Name** to **`capsule-view-1`**. Click **Save** to save the current contour view.



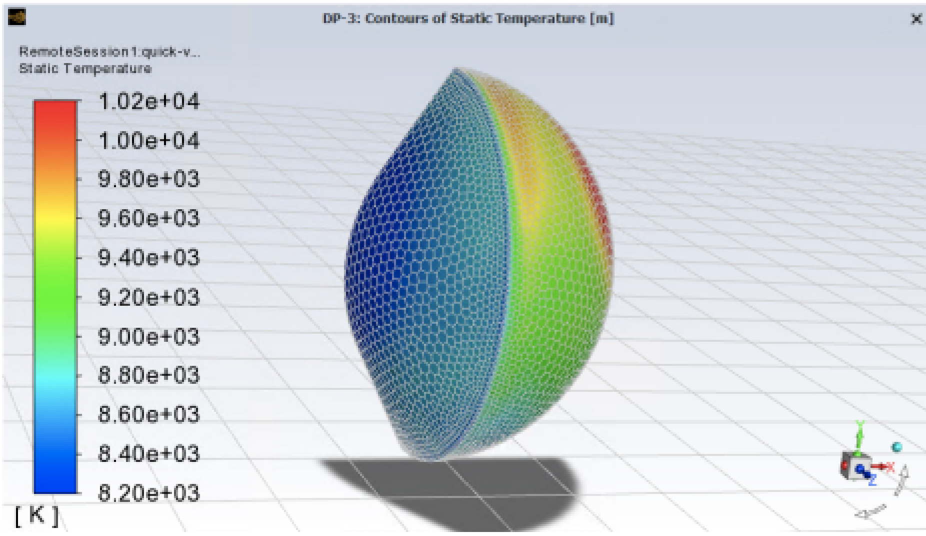

---

**Note:**

To change graphics display settings, you can go to **File** → **Preferences** → **Graphics**. For example, in the **Lighting** section, you can set **Headlight** to **On**, and set appropriate values to both **Headlight intensity** and **Ambient light intensity** to personalize the graphics object rendering.

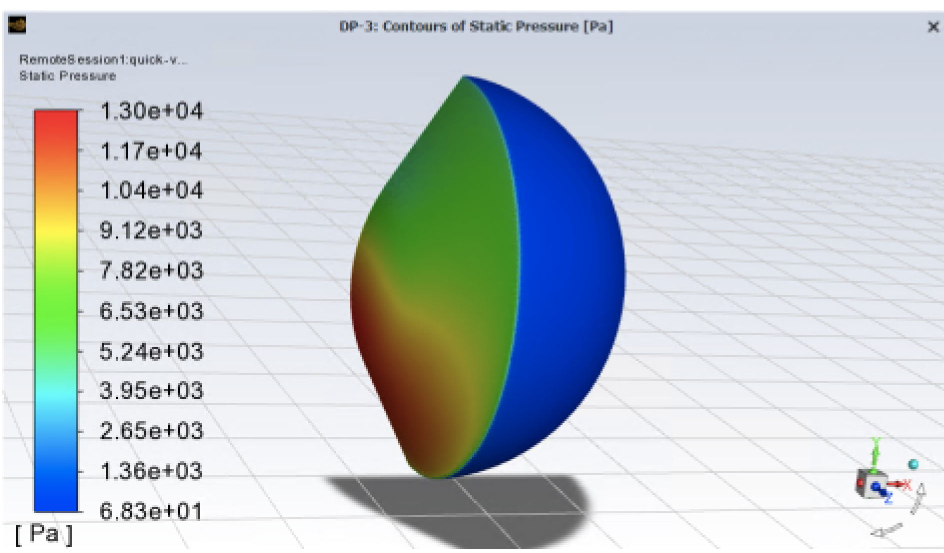
---

**Figure 33.43: Wall Static Temperature Contour of DP-03**




Set **Surfaces** to **Walls**, **Field** to **Static Pressure** and **Design Point** to **3**. Check to enable **Auto-Compute Range** and uncheck the **Draw Mesh** option. Click **Plot** and then click **Views...** after the contour is created. From the **Views** panel, select **view capsule-view-1** and click **Apply**. Close the **Views** panel. Click **Save Image...** and a **Save Picture** option panel will appear. From this panel, check to enable **Save All Updated DPs** and set **Format** to **JPEG**. Click the **Save...** button. Set image name to **`capsule-static-pressure`** from the **Select File** dialog and the contours of the static pressure for all the 3 Design Points will be saved to the **Results** folder as `capsule-static-pressure-DP-1`, `2` and `3.jpg`.

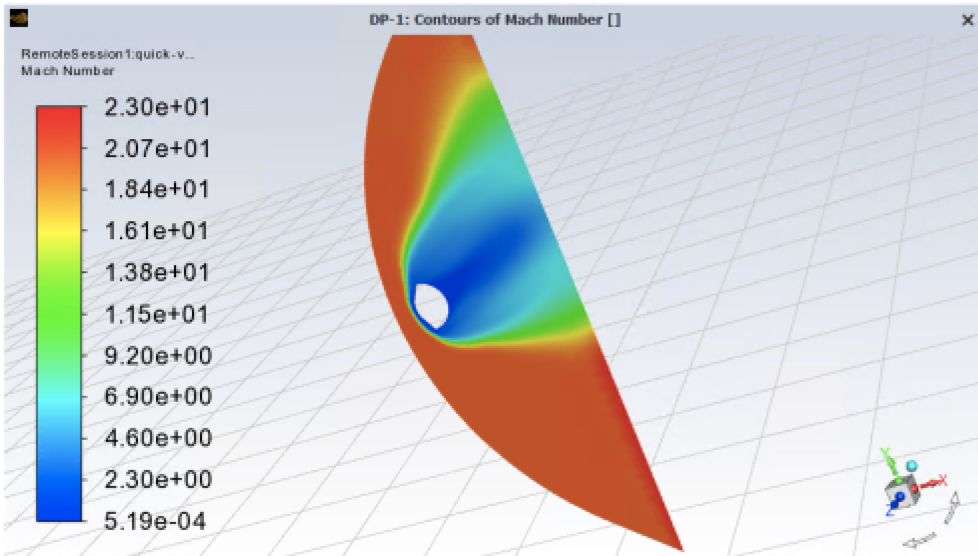
**Figure 33.44: Wall Static Pressure contour of DP-03**




In this solution, notice the high pressure region around the stagnation point located towards the front of the capsule (left side in image), and the low pressure region in the backside of the capsule, which is surrounded by the wake.

18. In the **Properties - Contours** area, set **Surfaces** to **Cutting Plane**, **Cutting Plane Normal Direction** to **Z**, **Cutting Plane Position [m]** to **`0.1`** and **Field** to **Mach Number**. Click on the **Plot** button to show the cutting plane contour in the **Graphics** window.

**Figure 33.45: Mach Number Cutting Plane Contour Plot**




19. Some of the selectable settings available in the **Solve** panel of Fluent Aero control a group of equivalent settings in the Fluent Solution Workspace. For example, in this case, the **Solver Methods** is set to **Default**, which will setup a range of equivalent settings inside the **Methods** panel of the background Solver session's Solution Workspace. This includes setting the **Spatial Discretization Gradient** to **Green-Gauss Node Based**, enabling **High Order Term Relaxation**, enabling **Convergence Acceleration for Stretched Meshes (CASM)**, and more. All the equivalent settings are applied to the Solution Workspace once the **Calculate** command is used in Fluent Aero.

---

**Note:**

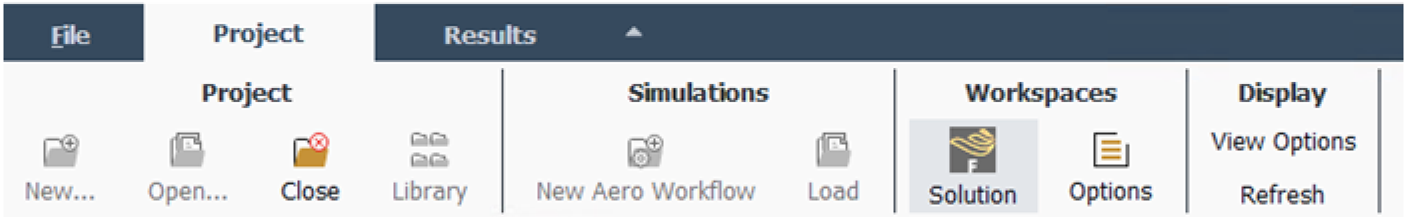
A full mapping of the available **Solve** settings in the Fluent Workspace and their equivalent settings in the Fluent Solution Workspace can be found in Mapping of the Solve Settings of Fluent Aero Workspace to Their Equivalent Settings in Fluent Solution Workspace (p. 513) .

---

Since we have already calculated some Design Points, having Solver Methods set to Default has already caused the equivalent group of Solution Workspace settings to be applied. However, we would now like to modify one of these default settings to use while continuing the calculation of one of our design points.

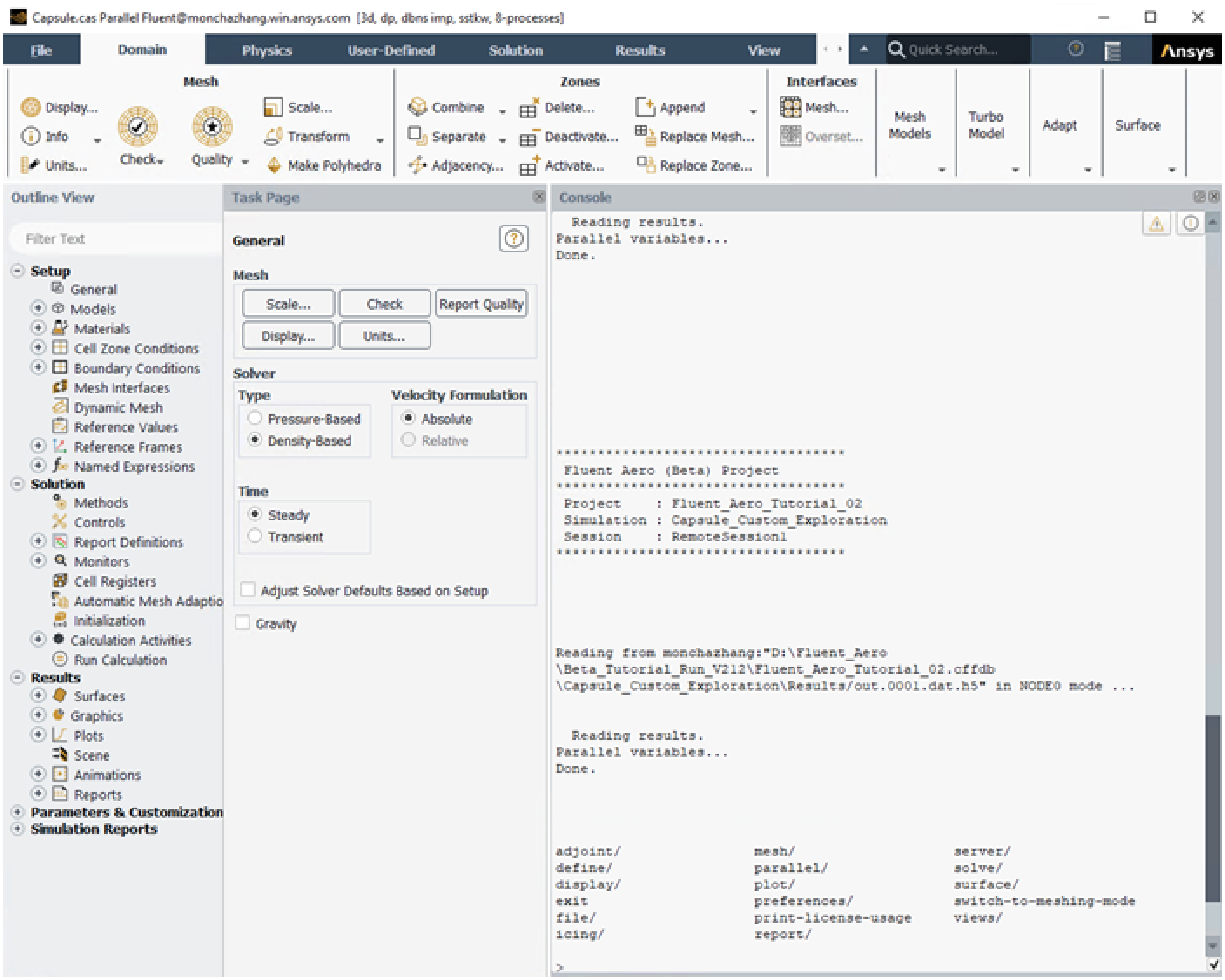
Go to the **Properties - Solve** panel. Change the **Solver Methods** to **Case settings**. From **Project** → **Workspaces**, click **Solution** to show the solution workspace.

**Figure 33.46: Enable Solution Workspace from the Project Ribbon**




Go to the solution workspace window console.

**Figure 33.47: Fluent Solution Workspace Window**



Enter the following text user interface command into the Solution Workspace console.

- `/solve/set/convergence-acceleration-for-stretched-meshes yes 100`

**Figure 33.48: Applying the TUI Command to the Solution Workspace Console**

Console

```
file/                   print-license-usage     views/
icing/                  report/

> /solve/set/convergence-acceleration-for-stretched-meshes yes 100
.
--------------------------------------------------------------------------------
 Convergence Accelerationfor for Stretched Meshes (CASM) has been selected.
 - The use of FMG initialization is highly recommended to provide good initial
   solution field before the start of calculations with CASM option.
 - Maximum benefit of CASM option is realized when local flow is aligned
   with mesh stretching.
--------------------------------------------------------------------------------

>
```

This will change the **CASM cut-off multiplier** from **2** (Fluent Aero's default value) to a more aggressive value of **100**, which could help to speed up convergence.

### Note:

If a new calculation was being performed from the beginning with the **Solver Methods** set to **Case settings**, Fluent Aero would not have applied any default Solver Methods settings to Solution Workspace. Therefore, the user would need to ensure that appropriate settings are applied to the relevant settings in the Solution Workspace. Using the **Apply Solver Settings** command accessed by right-clicking on **Solve** in the **Outline View** could be helpful to apply default solver settings without calculating. However, in this case, a calculation has already been performed while **Solver Methods** was set to **Default**, so appropriate equivalent settings have already been applied in the **Solution Workspace**, and a user can have confidence that changing the **CASM cut-off multiplier** is all that is required.

First, go to the **Input: Design Points** table and set the status of **DP-1** to **Continue To Update**.

Input:Design Points

| DP | Mach Number | Altitude [m] | Pressure [Pa] | Temperature [K] | Status |
|---|---|---|---|---|---|
| 1 | 20.65 | 67300.0 | 7.0319 | 225.01 | Continue To Update |
| 2 | 17.71 | 58200.0 | 26.0318 | 250.49 | Updated |
| 3 | 13.29 | 52400.0 | 56.0235 | 266.73 | Updated |

Next, go to the **Properties - Solve** panel, and relax the convergence conditions for this Design Point by changing the **Residuals Convergence Cutoff** and **Aero Coeff Conv. Cutoff** to `0.001`.

**Convergence**

| | |
|---|---|
| Residuals Convergence Cutoff | 0.002 |
| Aero Coeff Conv. Cutoff | 0.002 |

Finally, keep the **Iterations** set to `1000` and click **Start**.

**Figure 33.49: Convergence of Residuals for Design Point 1 Using Continue to Update**


Notice that after relaxing the convergence conditions and completing a **Continue to Update** calculation, the **Criteria Met?** is not set to **yes** in the **Summary** table for **DP 1**.

Table:Summary

| | DP | Mach | AoA [deg] | AoS [deg] | P [Pa] | T [K] | Reynolds | Avg Coeff Conv | Avg Residual | Criteria Met |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 20.65 | 18.4 | 0.0 | 7.0319 | 225.01 | 7.6839e+04 | 3.2891e-06 | 6.3923e-04 | yes |
| 2 | 2 | 17.71 | 18.4 | 0.0 | 26.0318 | 250.49 | 2.9224e+05 | 2.4967e-05 | 4.5531e-03 | no |
| 3 | 3 | 13.29 | 18.4 | 0.0 | 56.0235 | 266.73 | 5.2520e+05 | 2.3186e-05 | 2.0738e-03 | no |

20. After completing work on the current simulation, you can close the solver by left-clicking on **Capsule_custom_exploration** from the **Outline View** and then selecting **Close Solver**. An information panel will appear to ask you if you want to save the case file or not. Click **Yes** to save the case file.

21. Close the project and exit Fluent Aero. From the ribbon, select **Project** → **Close** to close a project. Next, select **File** → **Exit** and the Fluent Aero Workspace will be closed.

## 33.13.3. Introduction to Aircraft Component Groups and Computing Aerodynamic Coefficients on an Aircraft at Different Flight Altitudes and Engine Regimes

The objective of this tutorial is to introduce the aircraft **Component Groups** feature in Fluent Aero and to compute the flow around a common commercial aircraft at different flight Altitudes, Mach and engine regimes. The aircraft considered is a NASA Common Research Model (CRM) which is a wing-body-nacelle-python (WBNP) configuration. The flight conditions used in this tutorial represent example climb and nominal cruise conditions.

---

**Note:**

The flight conditions in the current tutorial have been defined for demonstration purposes only.

---

Download the `fluent_aero_tutorial.zip` file here.

Unzip `fluent_aero_tutorial.zip` to your working directory.

Extract the `CRM_WBNP_Aircraft.msh` file for this tutorial. The grid consists of 459,448 nodes and 2,124,630 cells. Three layers of prisms are grown off the aircraft's wall boundaries and the remainder of the computational domain is filled with tetrahedral cells. The limits of the computational domain are defined by a hemispherical boundary defined as a pressure-farfield boundary type, and a flat circular boundary defined as a symmetry plane in the Z direction.

---

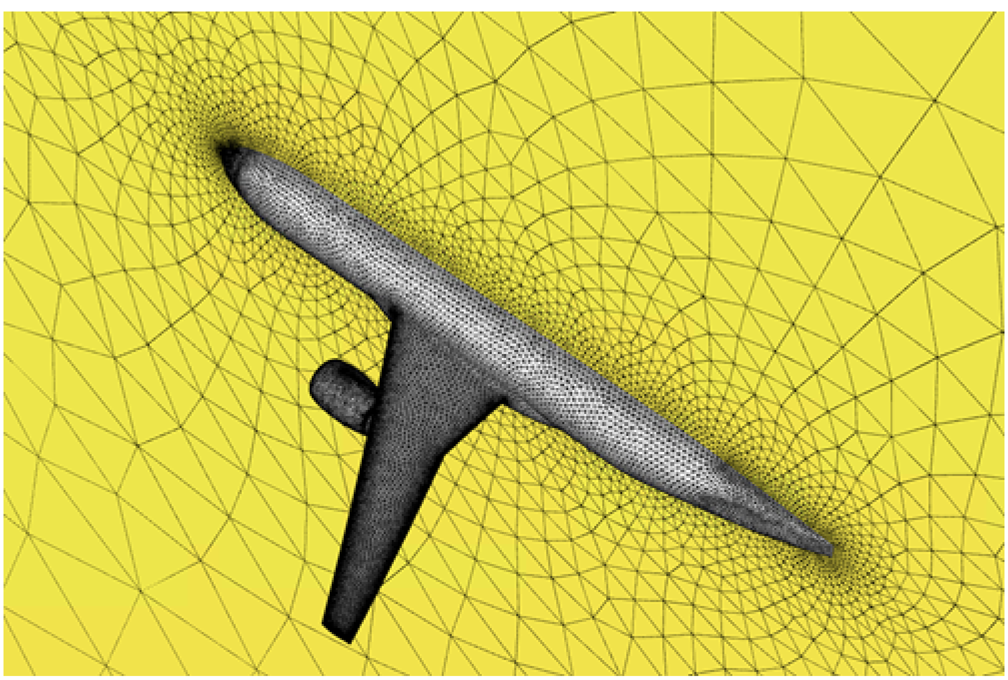
**Note:**

When using as an input, a Fluent format `.msh [.h5]` file should meet the Input Case File or Mesh File Requirements outlined in the Fluent Aero Manual.

The mesh used in this tutorial is a very coarse mesh. Its purpose is to quickly demonstrate a typical workflow and the **Component Groups** feature in Fluent Aero, and should not be relied upon for accurate simulations. For more accurate simulations, a finer mesh appropriate for external aerodynamic simulations, featuring more prism layers, higher mesh surface refinement, and increased mesh density in the wake region, should be used.

---

**Figure 33.50: View of the Mesh Around the Aircraft**



1. Launch Fluent on your computer. On the **Fluent Launcher** panel that appears, set the **Capacity Level** to **Premium** or **Enterprise**. Then enable the **Show Beta Workspaces** and select **Aero (Beta)**. Set the number of **Solver Processes** to **4**. Click **Start**.

2. In the Fluent Aero Workspace, go to **File** → **Preferences**. In the **Preference** window, go to the **Aero** menu and ensure **Use Custom Solver Launch Settings** is disabled.

   ---

   **Note:**

   If **Use Custom Solver Launch Settings** is enabled, a **Fluent Launcher** window will open when loading a case file, allowing you to schedule a calculation on a server and/or to specify a different number of **Solver Processes** to use in your simulation. if this setting is disabled, Fluent Aero will load the case file in a Solver session on your local machine with the same number of CPUs used to launch Fluent Aero.

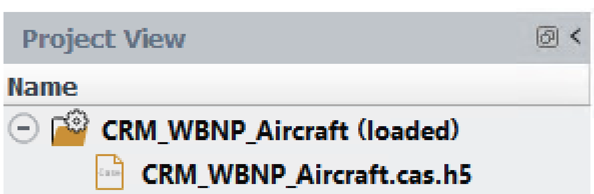
   ---

   Also, in the **Preferences** → **Aero** menu, make sure that **Show Solution Workspace** and **Enable Solution Workspace Graphics** are disabled. Click **OK**.

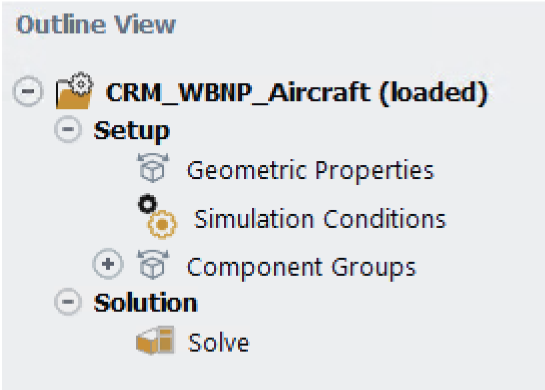
3. When Fluent Aero first opens, the **Project** tab will be displayed by default. In the **Project**'s top ribbon panel, select **Project** → **New...** and enter **`Fluent_Aero_Tutorial_03`** to create a new **Project** folder.

4. In the **Project**'s top ribbon, select **Simulations** → **New Aero Workflow**, and browse to and select the `CRM_WBNP_Aircraft.msh` file. A **New simulation** window will appear. Enter the **Name of the new simulation** as **`CRM_WBNP_Aircraft`**, and check to enable **Load in solver**.

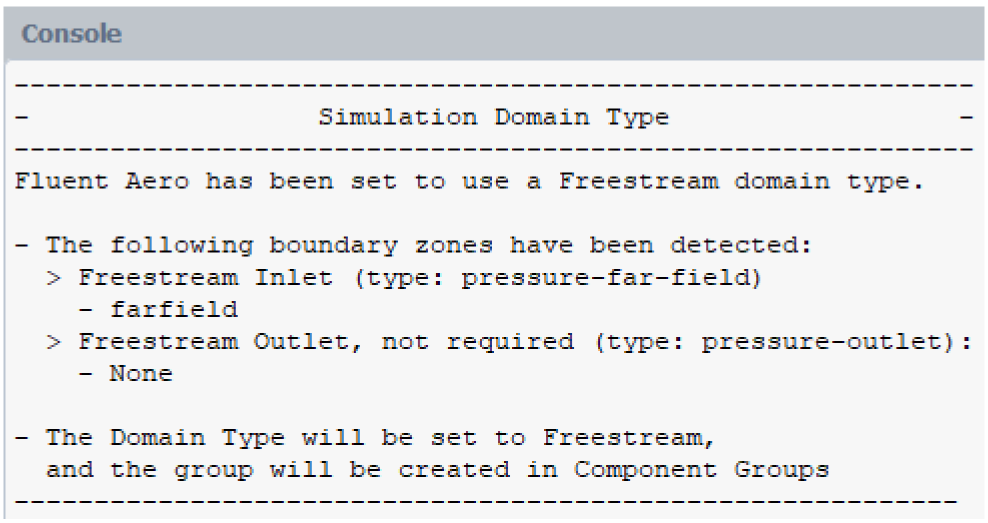
5. The case file will be opened and a background Solver session will be loaded. A new Simulation folder will be created in your **Project** folder. Fluent Aero will convert the `.msh` grid file to a `.cas.h5` format case file and the latter will be imported in the simulation folder as **CRM_WBNP_Aircraft.cas.h5**.

6. After the case file has been successfully loaded, a new **Setup** tree appears under **CRM_WBNP_Aircraft (loaded)** in the **Outline View** window.



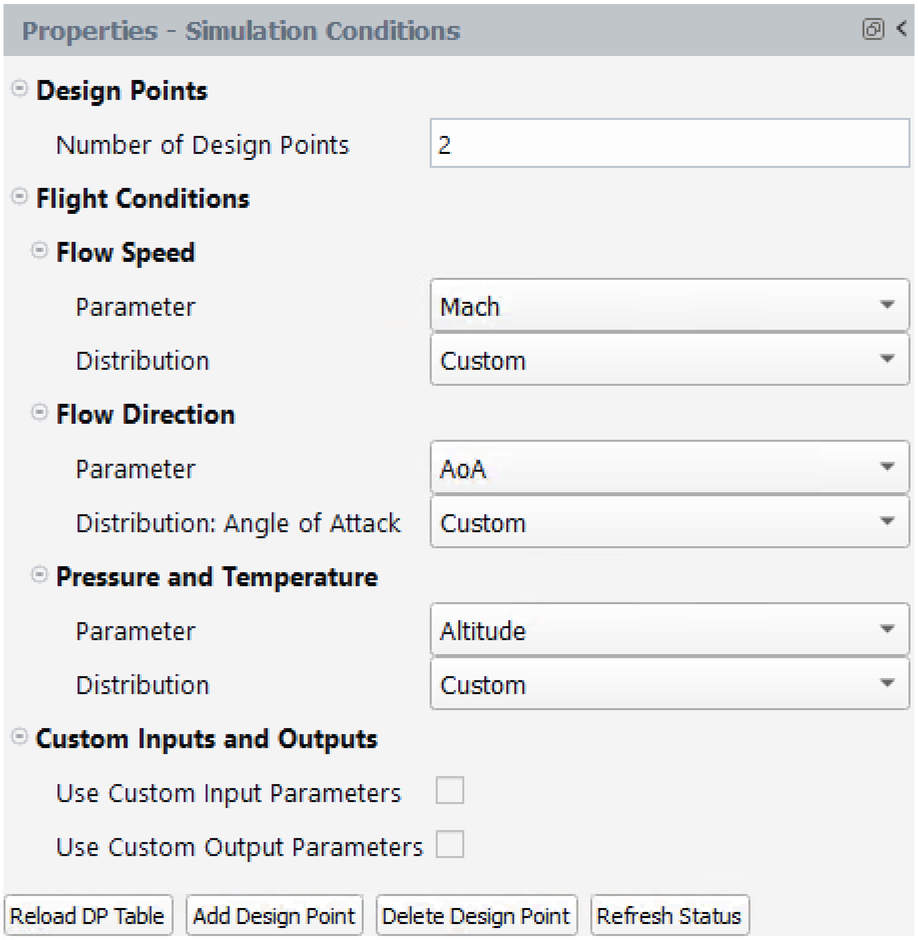

While importing, Fluent Aero will search for and find the pressure-far-field zone that defines the external boundary of the domain. It’s presence will cause Fluent Aero to determine that this case is using a **Freestream** domain type, and the following message will be reported in the console.



```
Console
------------------------------------------------------------
-                 Simulation Domain Type                   -
------------------------------------------------------------
Fluent Aero has been set to use a Freestream domain type.

- The following boundary zones have been detected:
  > Freestream Inlet (type: pressure-far-field)
    - farfield
  > Freestream Outlet, not required (type: pressure-outlet):
    - None

- The Domain Type will be set to Freestream,
  and the group will be created in Component Groups
-----------------------------------------------------------
```

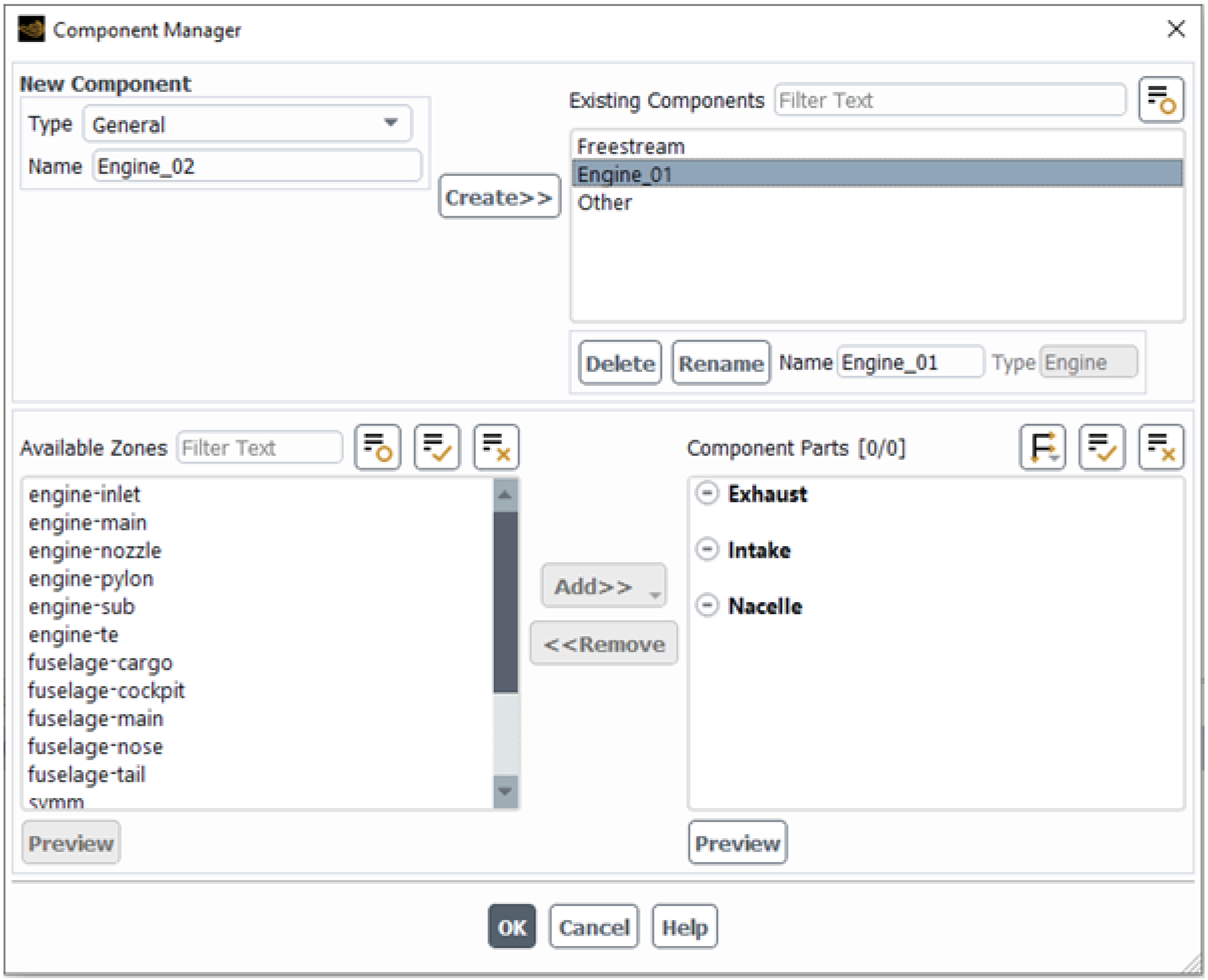
7. In the **Outline View** window, in the **Setup** tree, go to **Geometric Properties**. A **Properties – Geometric Properties** window appears below the **Outline View**. Define the orientation of the geometry within the computational domain, which is used to compute the aerodynamic forces. Set **Lift Direction at AoA = 0 degree** to **Y+**, set **Drag Direction at AoA = 0 degree** to **X+**. Set the **Moment Center X-**, **Y-** and **Z-Position [m]** to **`33.678`**, **`4.520`**, and **`0`**, respectively. Set the **Reference Length [m]** to **`7.005`**. Set the **Reference area [m^2]** to **`351.79`**. Alternatively, the user can use the **Compute Projected Area** tool to calculate the reference area.

8. In the **Setup** tree, go to **Simulation Conditions**. In the **Properties – Simulation Conditions** window, set the **Number of Design Points** to **2**. In the **Flow Speed** section, set **Parameter** to **Mach**, and **Distribution** to **Custom**. In the **Flow Direction** section, set **Parameter** to **AoA** and **Distribution: Angle of Attack** to **Custom**. In the **Pressure and Temperature** section, set **Parameter** to **Altitude**, **Distribution** to **Custom**.

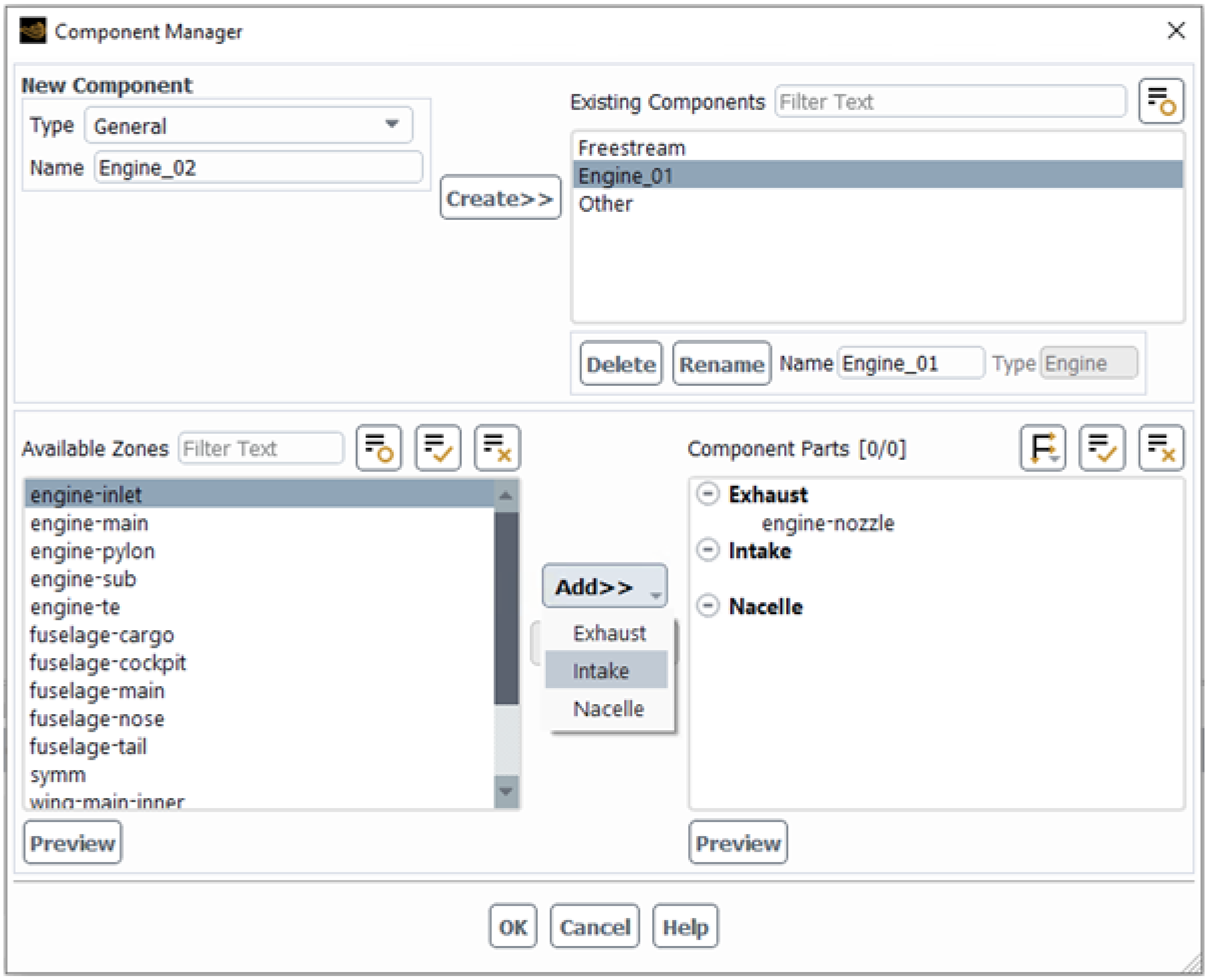



9. In the **Setup** tree, go to **Component Groups**. In the **Properties – Component Groups** window, click **Manage Components**. The component groups manger panel will appear.

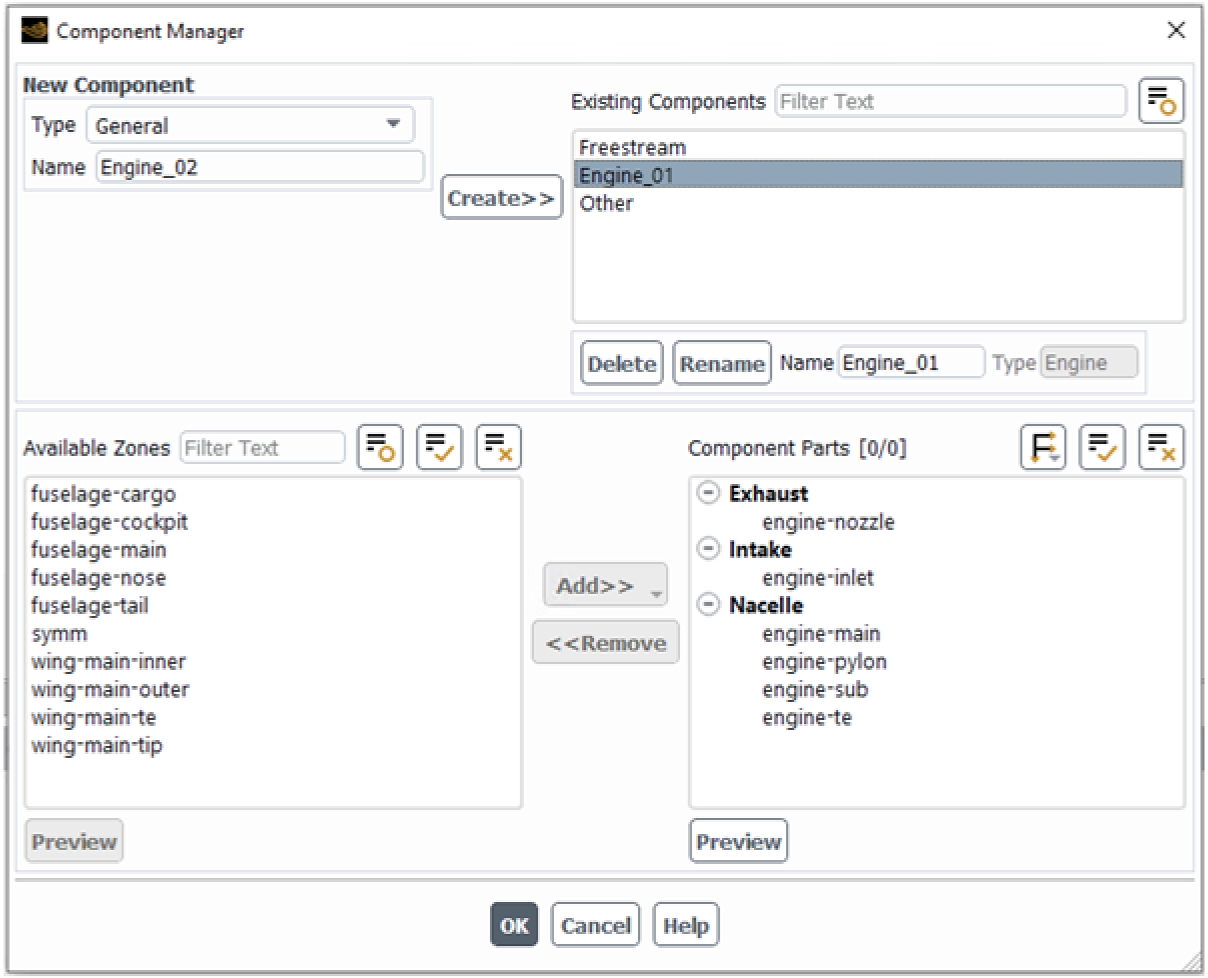
   To create an **Engine** type component, set **Type** to **Engine** in the **New Component** region and a default group name **Engine_01** will appear. Use **Create>>** to add **Engine_01** into **Existing Components**. All the boundary zones that have not been assigned to any group are listed in the **Available Zones**. An **Engine** type group is a specific type of group which contains three different **Component Parts**: **Exhaust**, **Intake** and **Nacelle**. At least one boundary zone must be added to each component part.



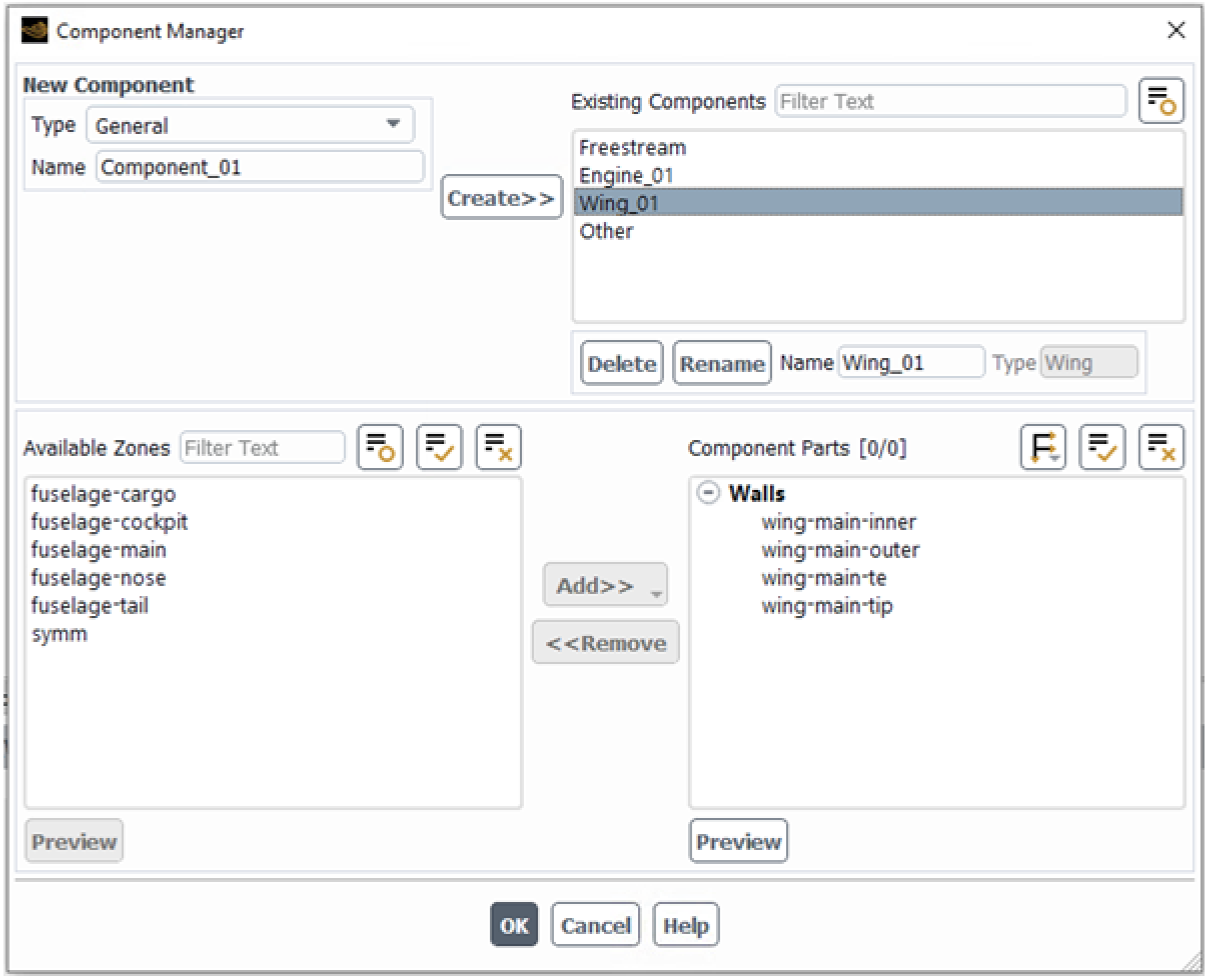

From **Available Zones**, select **engine-inlet** and use the **Add>>** button to add it to the Intake component part.



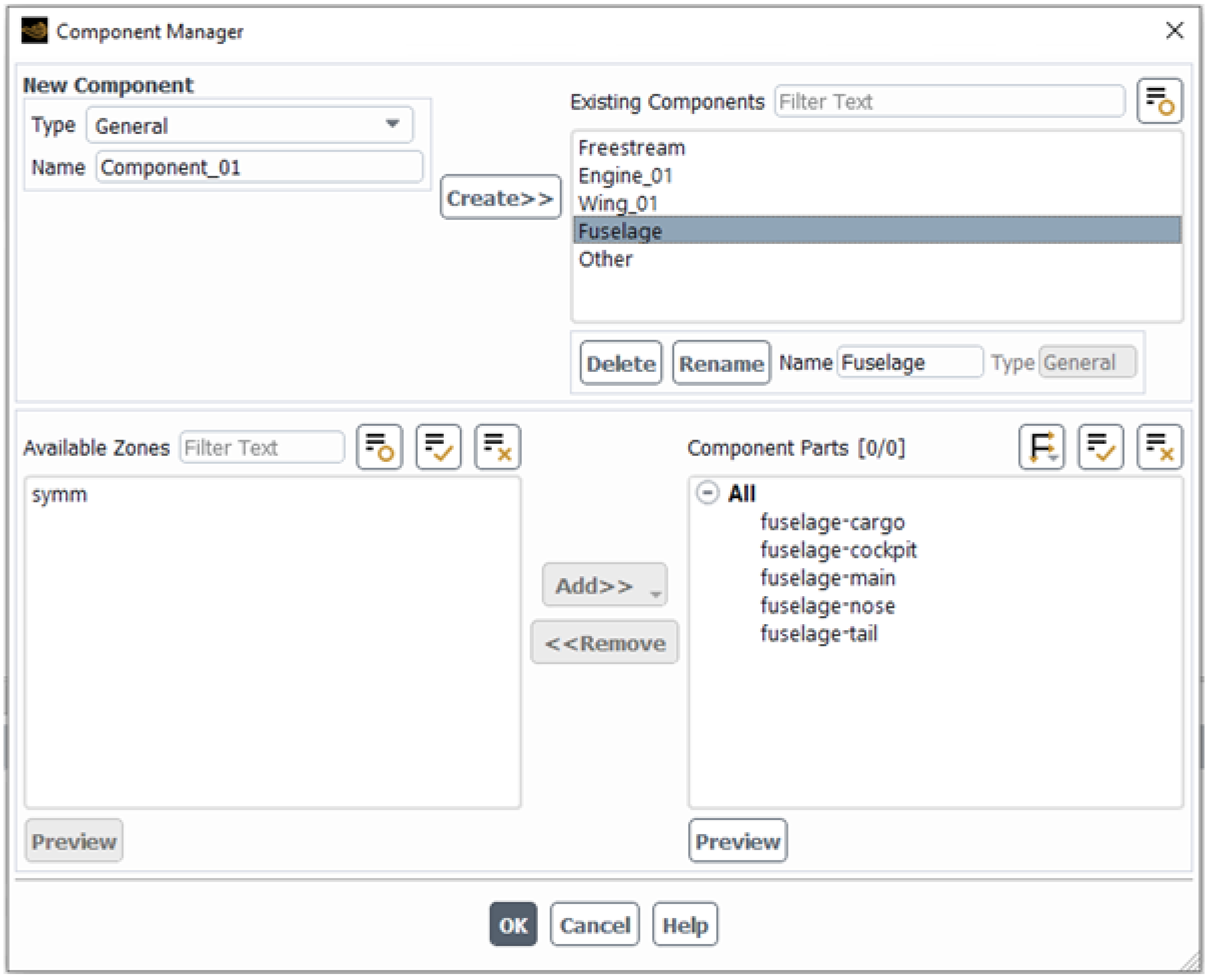

Select and add **engine-nozzle** to the **Exhaust** part. Select and add all the remaining boundary zones starting with **engine-** to the **Nacelle** part. The **Engine_01** component group is now successfully created.




10. From the **New Component** section, change **Type** to **Wing**. Use **Create>>** to add **Wing_01** to the **Existing Components**. From **Available Zones**, select all the boundary zones that start with **wing-** and use **Add>>** to add them to the **Walls** component parts.

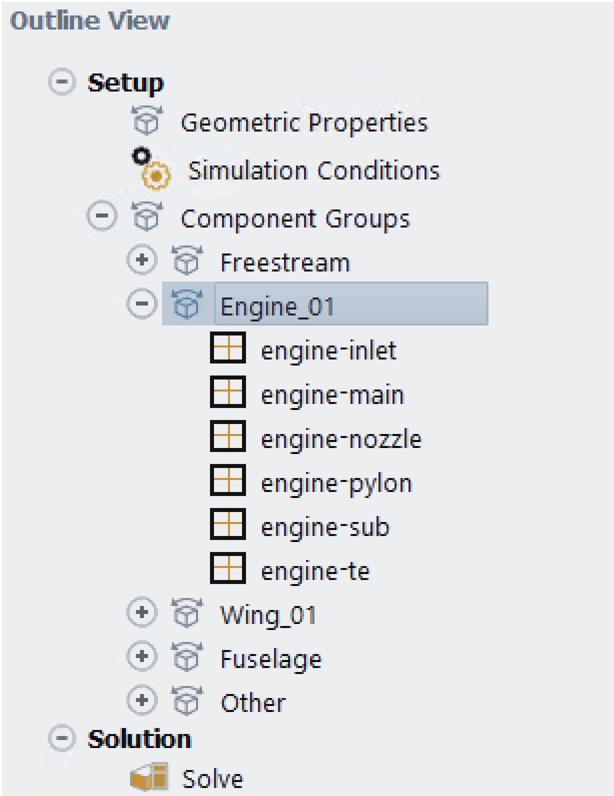



11. Change **Type** in the **New Component** to **General** and change **Name** to `Fuselage`. Use **Create>>** to add the **Fuselage** group to **Existing Components**. From the **Available Zones**, select all the boundary zones with the **fuselage-** prefix and use **Add>>** to add them to the **All** component part.

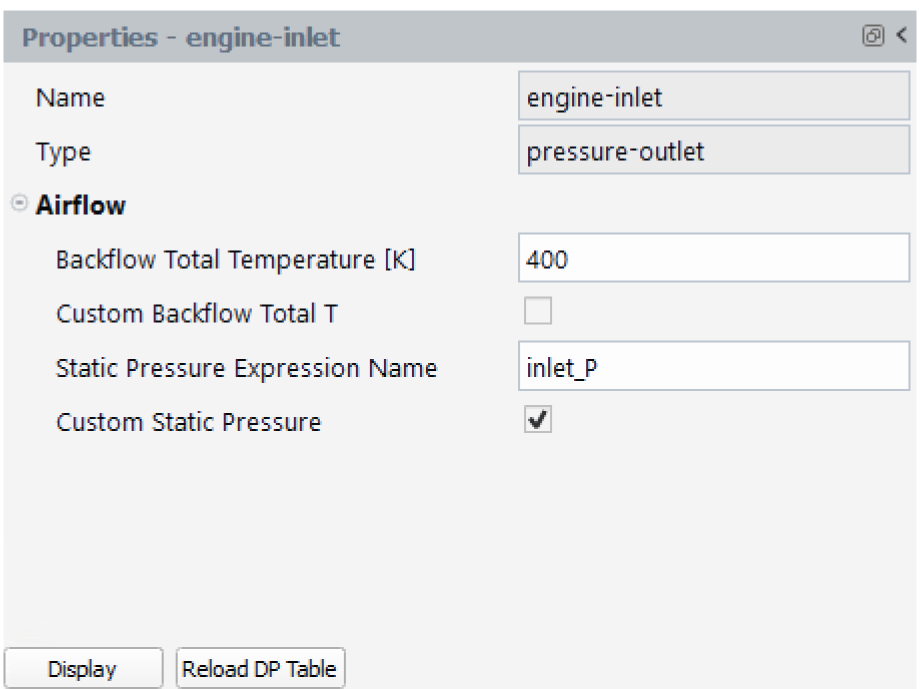



Click **OK** to close the **Component Manager**.

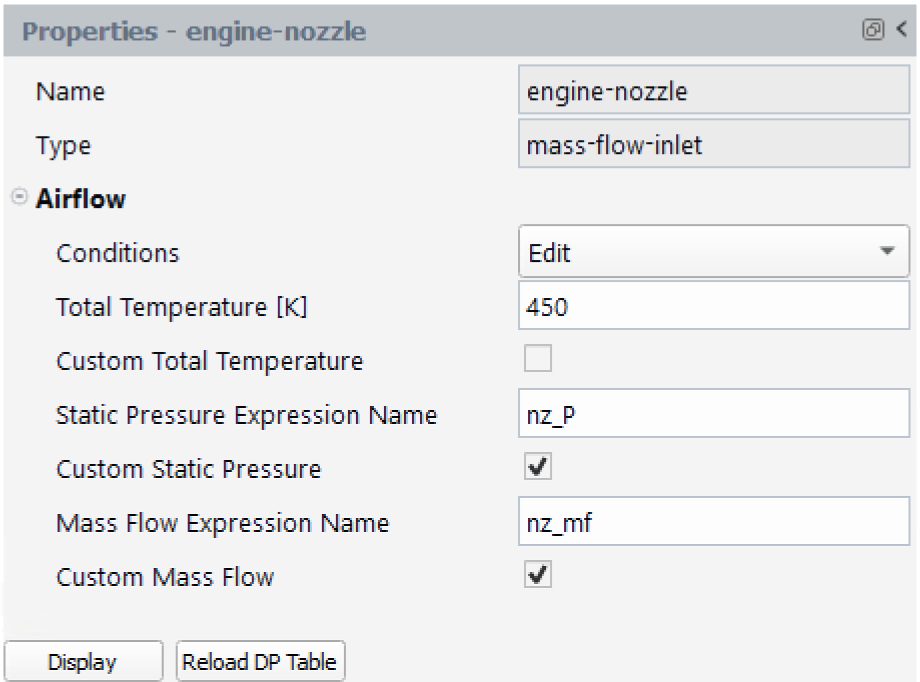
12. The three new **Component Groups** will appear in the **Outline View** in addition to the default **Freestream** and **Other** groups. From **Outline View** → **Component Groups**, expand the **Engine_01** component and select **engine-inlet** which is a pressure-outlet type boundary zone.




In the **Properties – engine-inlet** window, set **Backflow Total Temperature [K]** to **`400`**. Check to enable **Custom Static Pressure** to apply different **Static Pressure [Pa]** for each Design Point. Change the default name to **`inlet_P`**. A custom expression for the **Static Pressure [Pa]** will be created in the solution workspace with the name **inlet_P**. A column named **inlet_P** will be added to the **Input:Design Points** table.



13. Select **engine-nozzle** to set boundary conditions at the engine exhaust. In the **Properties – engine-nozzle** window, change **Conditions** to **Edit**. Set **Total Temperature [K]** to **450**. Check to enable **Custom Static Pressure** and **Custom Mass Flow** in order to apply different **Static Pressure [Pa]** and **Mass Flow [kg/s]** for each Design Point. Change the default expression names to **nz_P** and **nz_mf** respectively. Two custom expressions will be created in the solution workspace and they will can edited from the **Input:Design Points** table.




14. The user can manually fill the **Input:Design Points** table. Notice that there are 9 columns in the table. The first column specifies the **Design Point** number, and cannot be edited. The second to fourth columns are for specifying the variable inputs of **Mach Number**, **Angle of Attack [deg.]** and **Altitude [m]**, respectively. The **Pressure [Pa]** and **Temperature [K]** columns cannot be edited, as they will be automatically calculated and filled based on the **Altitude** input. The next three columns define the boundary conditions from the **Engine_01** group. They allow the user to input conditions corresponding to different engine regimes. The last column is set to **Needs Update** for both of the Design Points since they have not yet been calculated.

    Fill the table with the values as shown below.

    **Figure 33.51: Input:Design Points Table With 2 Design Points**

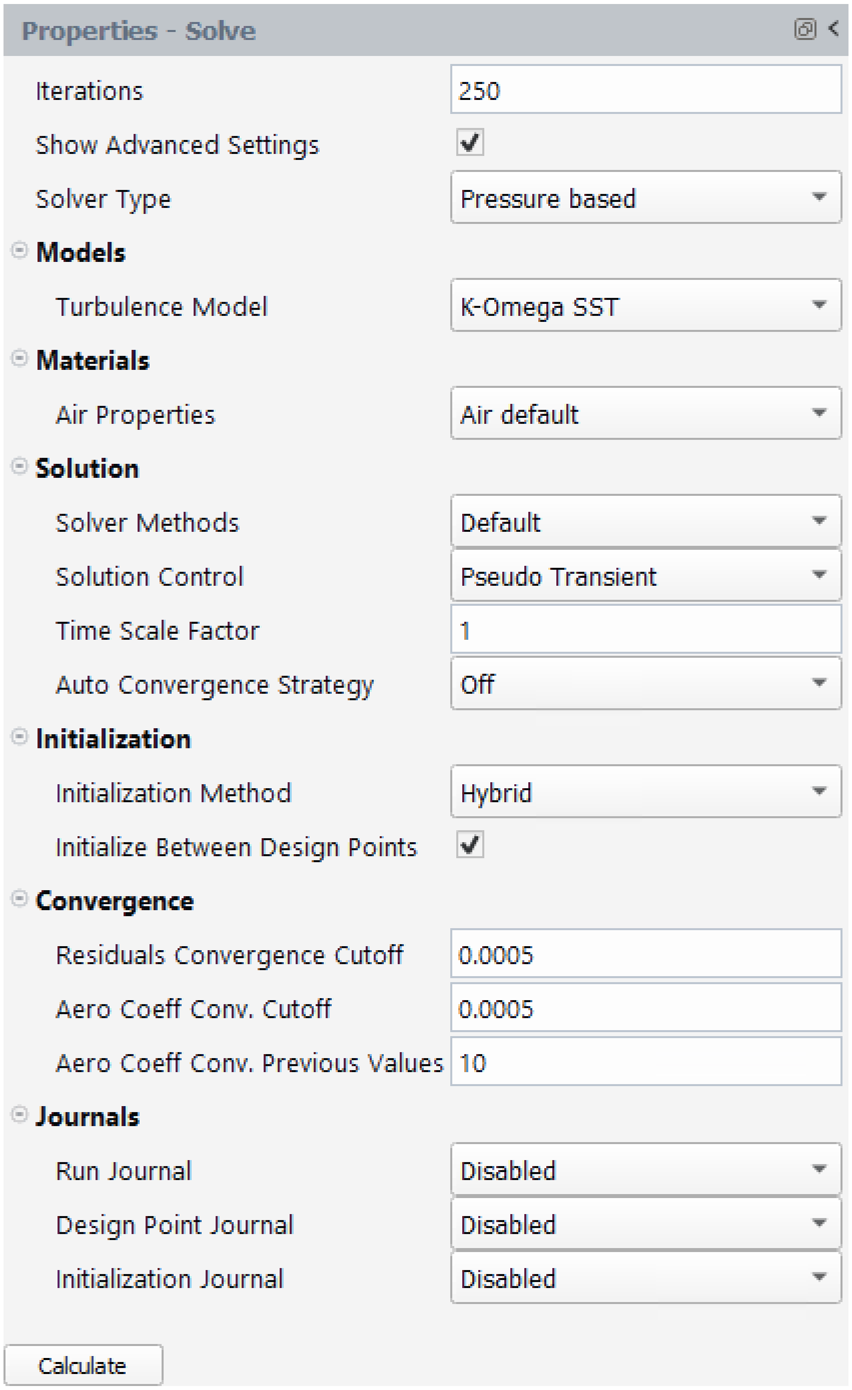
Input:Design Points

| DP | Mach Number | Angle of Attack [deg.] | Altitude [m] | Pressure [Pa] |
|---|---|---|---|---|
| 1 | 0.45 | 5.25 | 5000.0 | 54019.9036 |
| 2 | 0.85 | 2.1 | 11200.0 | 21929.4349 |

| Temperature [K] | nz_mf [kg s^-1] | nz_P [kg m^-1 s^-2] | inlet_P [kg m^-1 s^-2] | Status |
|---|---|---|---|---|
| 255.65 | 800.0 | 51500.0 | 57200.0 | Needs Update |
| 216.65 | 600.0 | 20000.0 | 24000.0 | Needs Update |

15. In the **Outline View**, go to **Solution** → **Solve**. Change the number of **Iterations** to **250**. Check to enable **Show Advanced Settings**. This reveals model and solver parameters that advanced

users can edit. Change the **Solver Type** to **PBNS**. Since this is a coarse mesh, we will relax the convergence cutoff criteria. Set both **Residuals Convergence Cutoff** and **Aero Coeff Conv. Cutoff** to `0.0005`. For the remaining solver parameters, Fluent Aero's default PBNS settings will be applied.

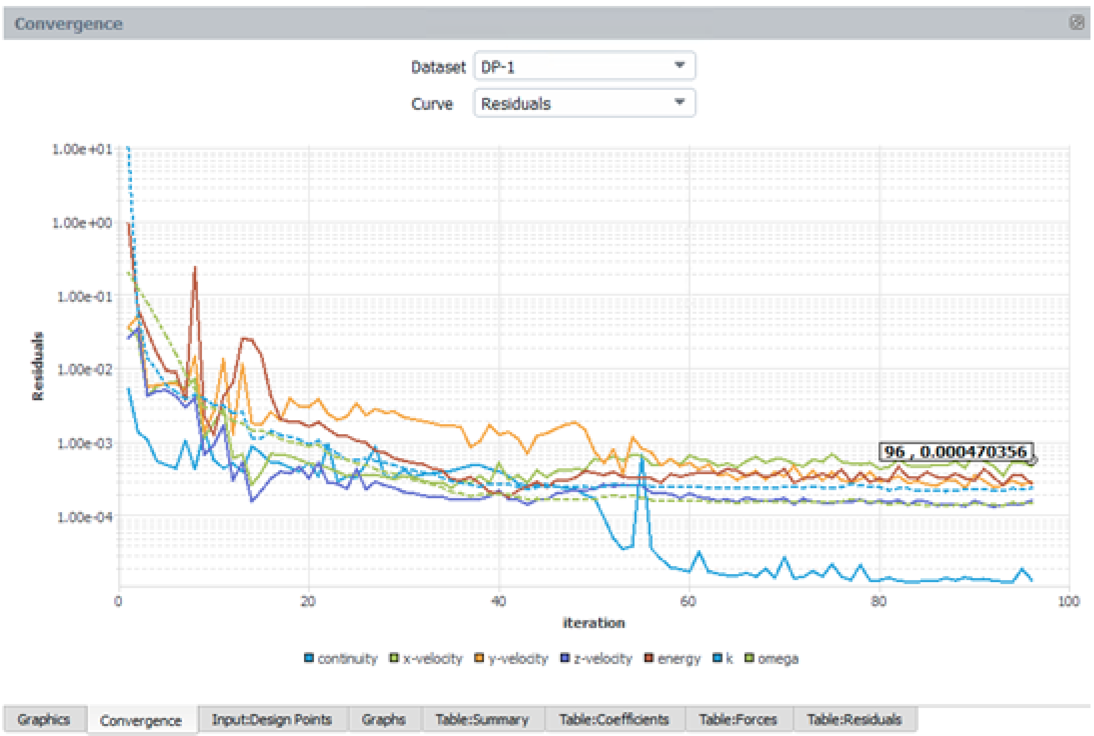



16. Click on the **Calculate** button at the bottom of the **Properties - Solve** panel. Alternatively, right-click on **Solve** from the **Outline View** tree and select **Calculate** from the drop-down menu. The calculation will start, and the first Design Point, **DP-1** will be simulated. A **Convergence** window will appear in place of the **Graphics** window and display the residuals and monitors for **DP-1**.

17. Design Point **DP-1** will calculate until the total **Iterations** (250) or the **Residuals Convergence Cutoff** (5e-4) and **Aero Coeff Conv. Cutoff** (5e-4) are reached, whichever comes first. In this example, **DP-1** will run 250 iterations.

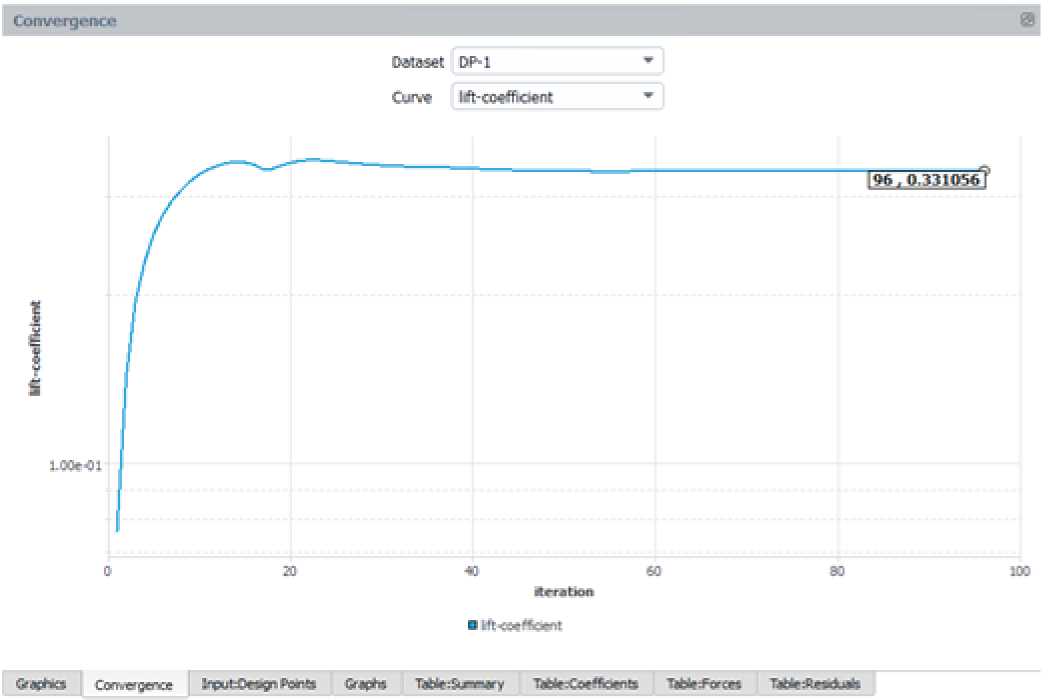
18. Look at the convergence history of the simulation in the **Convergence** window located on the right of your screen. Set the **Dataset** to **DP-1** and the **Curve** to **Residuals** to view the continuity, x- y- z-velocity, energy, and turbulence residuals for the first Design Point. You can left-click on a residual curve to show the iteration number and the corresponding residual value. Here, the residual of the x-velocity at iteration 96 is shown and equals to 4.70356e-4 which is slightly lower than the residual convergence cutoff (5e-4). The remaining residuals of the **DP- 1** have met the **Residuals Convergence Cutoff** criteria after 96 iterations.

**Figure 33.52: Convergence of Residuals for Design Point 1**




In the **Convergence** window, set **Curve** to **lift-coefficient**. The evolution of the lift coefficient for **DP-1** will be displayed. Left-click on last iteration of the lift-coefficient plot, to show the value of the lift coefficient at iteration 96.

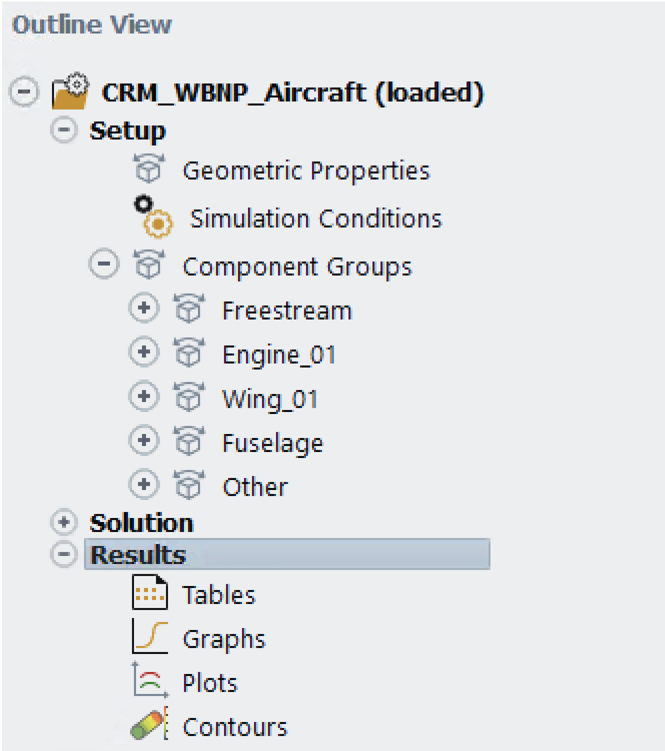
**Figure 33.53: Convergence History of the Lift Coefficient for Design Point 1**




When the calculation of DP-1 is complete, the status column in row 1 of the **Input: Design Points** table will be set to **Updated**, and the calculation of DP-2 will begin.

After all the Design Points have been updated, the status of the **Input: Design Points** table will be set to **Updated** for all the Design Points.

19. A **Results** section will be displayed in the **Outline View** tree after the simulation starts. This allows a user to quickly post-process all Design Point solutions, by obtaining aerodynamic coefficient plots, creating contour plots of the solution fields, comparing solution fields to experimental data, and more.




20. The first element in the **Results** section is **Tables**. In the current tutorial, 4 different Tables will be automatically created in the **Graphics** window area when the calculation is complete.

    Click on the **Table: Summary**, **Table: Coefficients**, **Table: Forces**, and **Table: Residuals** tabs at the bottom of the **Graphics** window to reveal each table.

    - **Table: Summary** summarizes the flight conditions and convergence information for each Design Point. In the current simulation, both Design Points have partially met the convergence criteria.

      **Figure 33.54: Summary Table of the Flight Conditions and Convergence Information**

      Table:Summary

      | | DP | Mach | AoA [deg] | AoS [deg] | P [Pa] | T [K] | Reynolds | Avg Coeff Conv | Avg Residual | Criteria Met? |
      |---|---|---|---|---|---|---|---|---|---|---|
      | 1 | 1 | 0.45 | 5.25 | 0.0 | 54019.9036 | 255.65 | 3.1926e+07 | 3.2904e-04 | 2.2845e-04 | yes |
      | 2 | 2 | 0.85 | 2.1 | 0.0 | 21929.4349 | 216.65 | 1.8533e+07 | 4.4786e-04 | 3.7584e-04 | partially |

    - **Table: Residuals** summarizes the final residuals as well as the number of iterations run for each Design Point. For each Design Point, only the residuals of the y-velocity are slightly higher than the cutoff criteria.

      **Figure 33.55: Results Table of Final Residuals**

      Table:Residuals

      | | DP | iteration | continuity | x-velocity | y-velocity | z-velocity | energy | k | omega |
      |---|---|---|---|---|---|---|---|---|---|
      | 1 | 1 | 96 | 1.3404e-05 | 4.7036e-04 | 2.9263e-04 | 1.6281e-04 | 2.7378e-04 | 2.3617e-04 | 1.4996e-04 |
      | 2 | 2 | 250 | 3.2762e-05 | 7.8733e-04 | 6.7725e-04 | 3.3170e-04 | 2.7506e-04 | 2.8632e-04 | 2.4046e-04 |

- **Table: Coefficients** contains the lift, drag and moment coefficients. The last three columns show the convergence information of the lift and drag coefficients and the maximum value among the three moment coefficients.

**Note:**

Only the convergence of lift and drag coefficients are used to verify if the cutoff criteria is met. For the Design Point 1, the convergence of both lift and drag coefficient have met the cutoff criteria. For the Design Point 2, the convergence of the lift coefficient has not yet been satisfied.

**Figure 33.56: Results Table of the Aerodynamic Coefficients**

Table:Coefficients

| | DP | Cl | Cd | Cm-y | Cm-p | Cm-r | Cl Conv. | Cd Conv. | Max. Cm Conv. |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 3.3106e-01 | 2.1489e-02 | -6.6588e-02 | -1.9484e-02 | 5.4581e-01 | 3.1256e-04 | 3.4553e-04 | 4.6514e-04 |
| 2 | 2 | 2.0948e-01 | 1.9553e-02 | -8.4375e-02 | -4.1032e-02 | 3.4301e-01 | 4.7858e-04 | 4.1713e-04 | 5.4469e-04 |

**Note:**

While most of the residuals and the convergence of the aerodynamic coefficients meet the residual convergence cutoff criterion of 5e-4, it is still recommended for users to investigate their solutions to ensure that appropriate convergence levels have been achieved and that convergence remains stable. A residual convergence cutoff of 5e-4 may be appropriate for some cases, but not for others, and therefore care should be taken when selecting this value.

- **Table: Forces** contains the dimensional forces of lift, drag and moment.

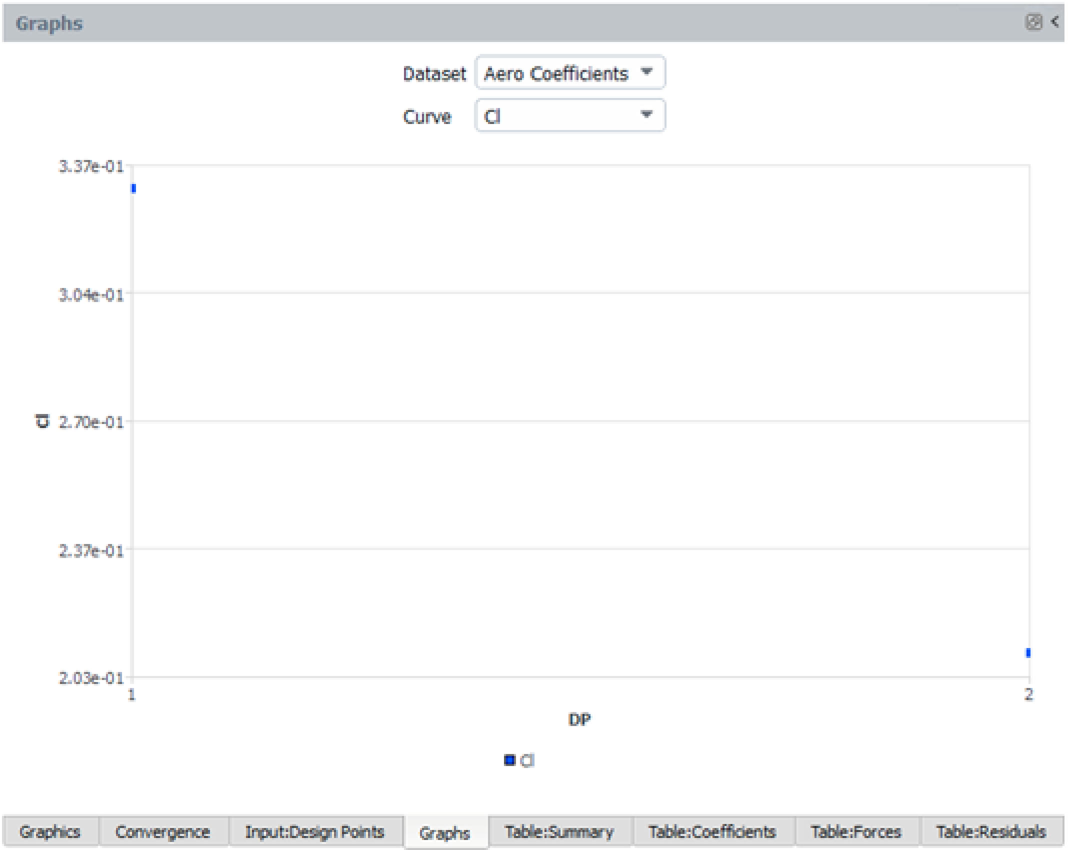
**Figure 33.57: Results Table of the Aerodynamic Forces**

Table:Forces

| | DP | Lift [N] | Drag [N] | Mom. Yaw [N.m] | Mom. Pitch [N.m] | Mom. Roll [N.m] |
|---|---|---|---|---|---|---|
| 1 | 1 | 8.9179e+05 | 5.7885e+04 | -1.2565e+06 | -3.6766e+05 | 1.0299e+07 |
| 2 | 2 | 8.1730e+05 | 7.6288e+04 | -2.3060e+06 | -1.1214e+06 | 9.3749e+06 |

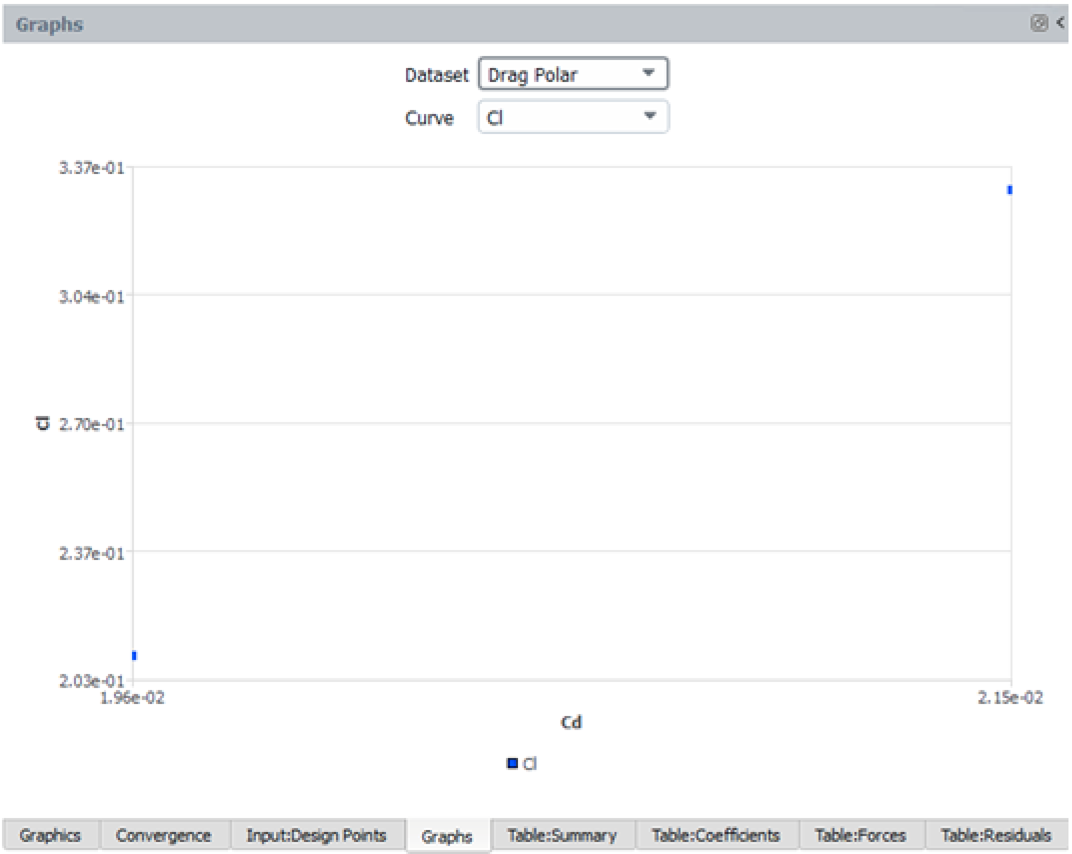
21. Click on **Graphs** in the **Outline View** to show the plots of the aerodynamic coefficients defined in Fluent Aero. At the bottom of the **Properties - Graphs** window, click **Plot Coefficients**. An X-Y plot of Lift Coefficient (**Cl**) vs. Design Point (**DP**) will appear in the **Graphics** window. The Drag and Moment Coefficients can be shown by selecting **Cd** and **Cm-r/y/p** from the **Curve** selection drop-down list. Select **Cl**/**Cd** from the **Curve** to show the lift to drag ratio as a function of the Design Point.

**Figure 33.58: Lift to Drag Ratio vs. Design Point**

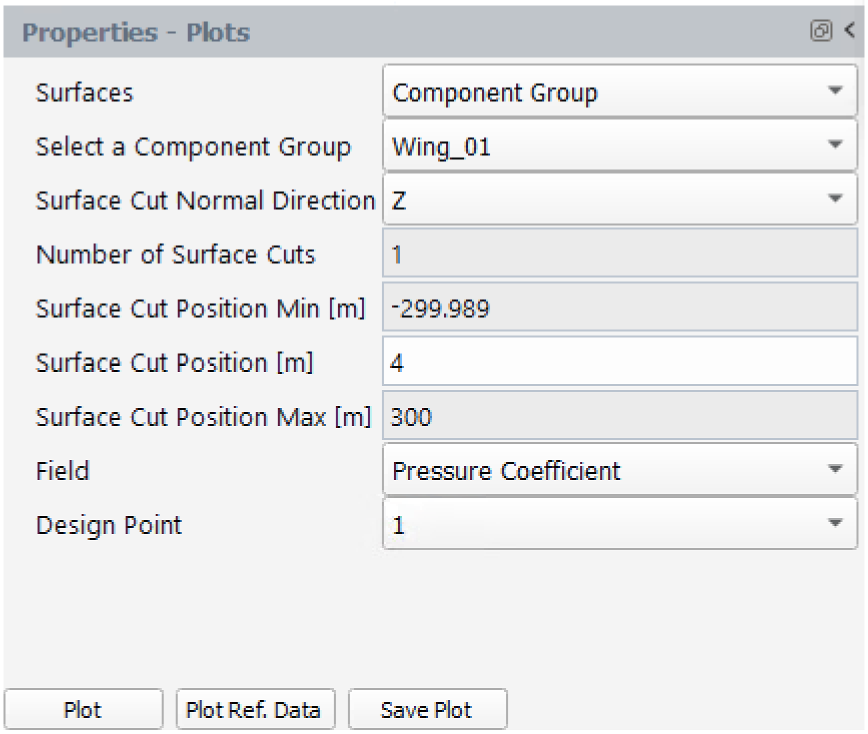



Click the **Show Cl Cd Plot** button. An X-Y plot of Lift Coefficient (Cl) vs. Drag Coefficient (Cd) will appear in the **Graphics** window. Alternatively, you can simply change **Dataset** to **Drag Polar** from the **Graphics** window to show the drag polar plot.

**Figure 33.59: Lift Coefficient vs. Drag Coefficient**




22. The **Plots** options can be used to quickly display simple 2D plots of selected design points and solution variables. Left-click on **Plots**. The **Properties - Plots** window will be displayed. Set **Surfaces** to **Component Groups**. Change **Select a Component Group** to **Wing_01**. Set **Surface Cut Normal Direction** to **Z** and set **Surface Cut Position [m]** to **4** which is a position close to the fuselage. Then set **Field** to **Pressure Coefficient** and **Design Point** to **1**. Click **Plot** and the pressure coefficient on the walls of DP-1 at Z=4m will be plotted in the **CutPlots** window.




A **Plot Options** panel will appear after clicking the **Plot** button. Users can use this panel to personalize some plot settings for both axes.

---

**Note:**

If the pressure coefficient is selected in **Field**, the range of y-axis will be reverted with higher value on the bottom and lower value on the top. The pattern and color of the curve can be modified as well.

---

Set **Axis** to **X** and uncheck **Auto Range**. Set **Range/Minimum** and **Maximum** to **25.5** and **37.5** respectively. Set **Number Format/Type** to **general** and **Precision** to **3**. Set **Pattern** to **line--dash-dotted** from **Curve Settings**. Click **Plot** from the **Plot Options** panel.

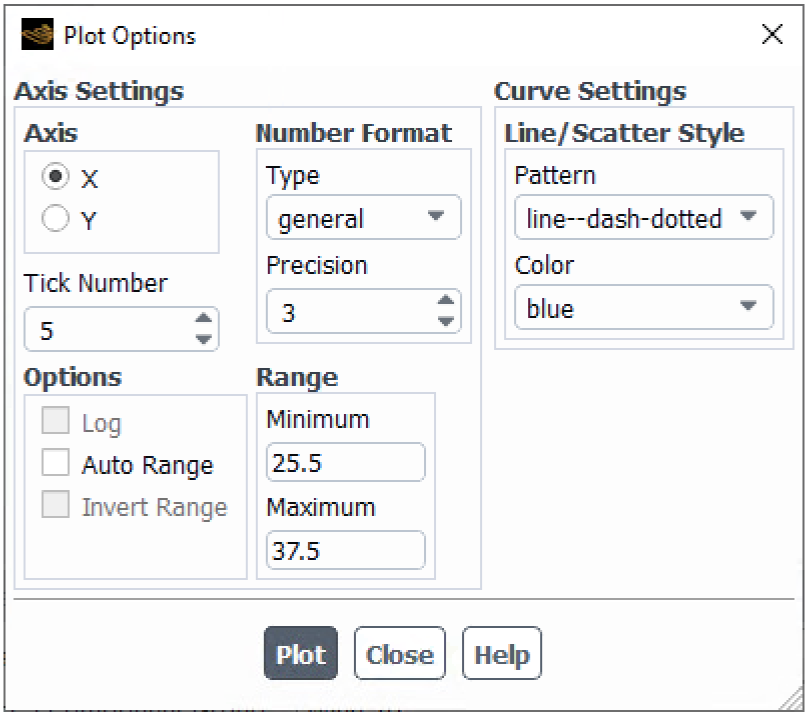
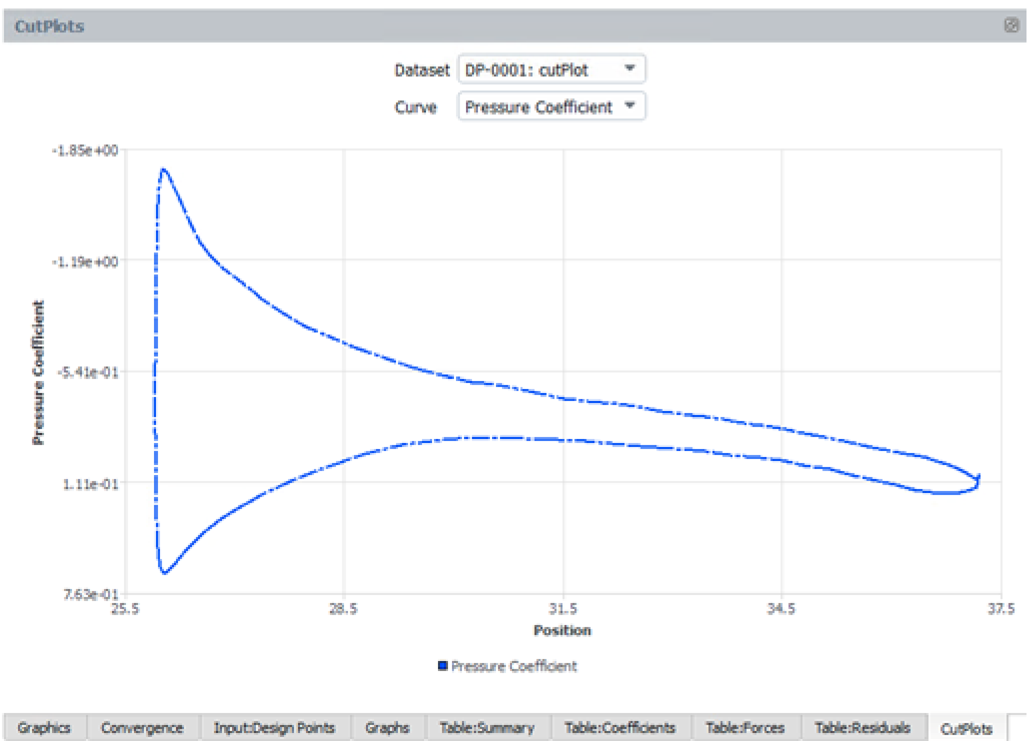


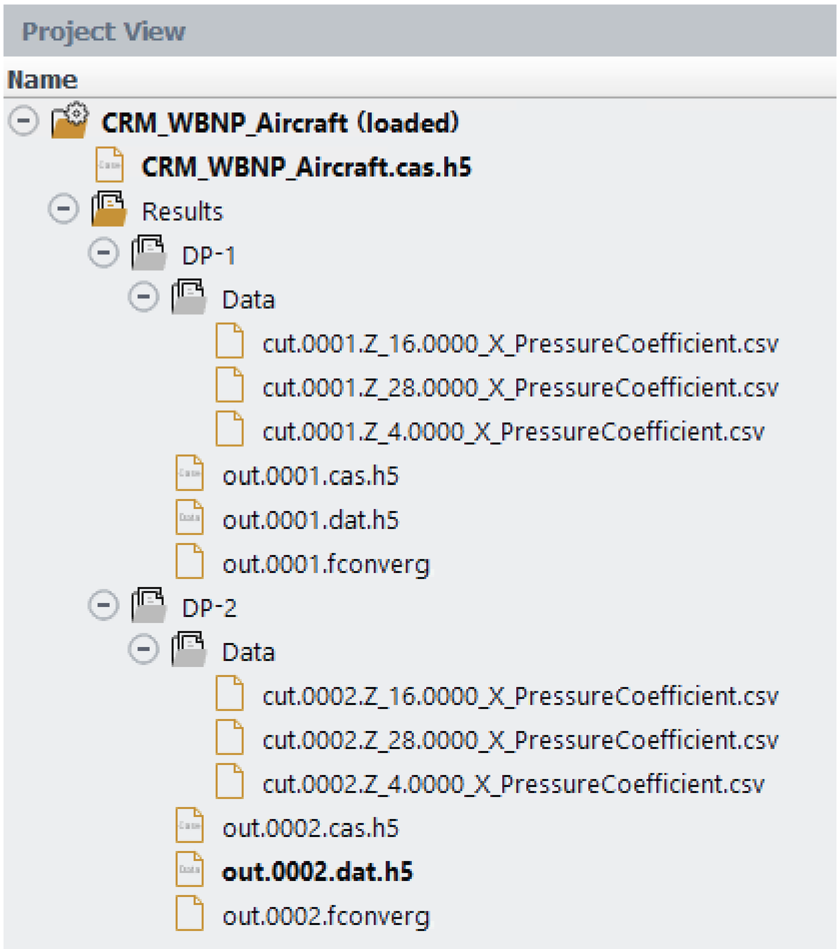

**Figure 33.60: Distribution of the Wall Pressure Coefficient at Z=4m for DP-1**



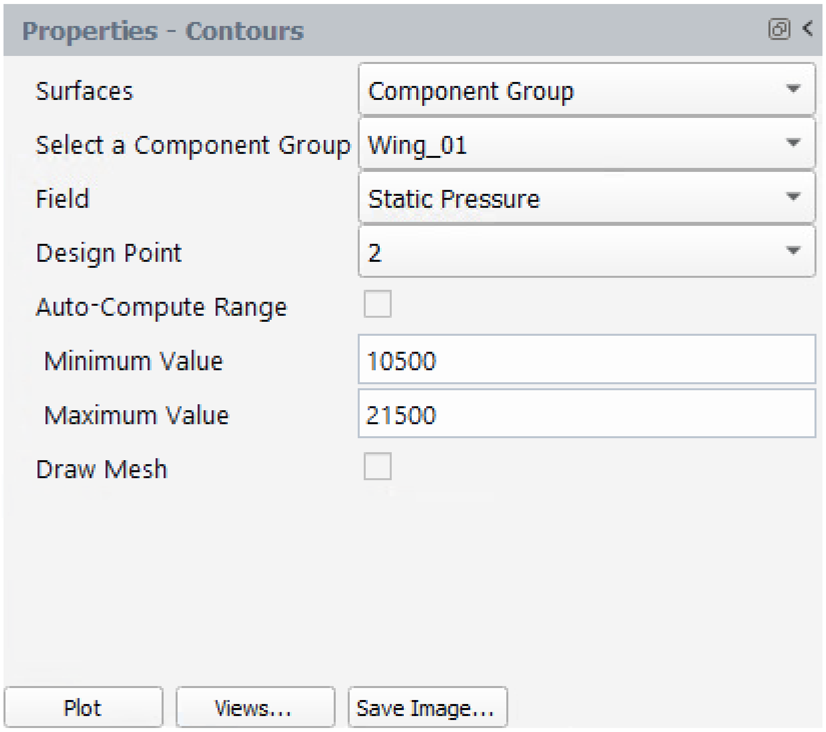

## Note:

A `.csv` file will be saved in the results folder after clicking the **Plot** button, which is visible within **DP-1**/**Data** folder in the **Project View**. There is a known issue of the current tool to detect the discontinuities on the curves, which will be fixed in the next release.

Repeat the above step for **Surface Cut Position [m]** of **16** and **28** which are positions close to the mid-range and the tip of the wing respectively. After plotting the pressure-coefficient at these wing-span positions, set the **Design Point** to **2** and create the same plots for **DP-2**. After completing these plots, the correponding `.csv` files are saved to the **DP-1**/**Data** and **DP-2**/**Data** folder in the **Results** directory.

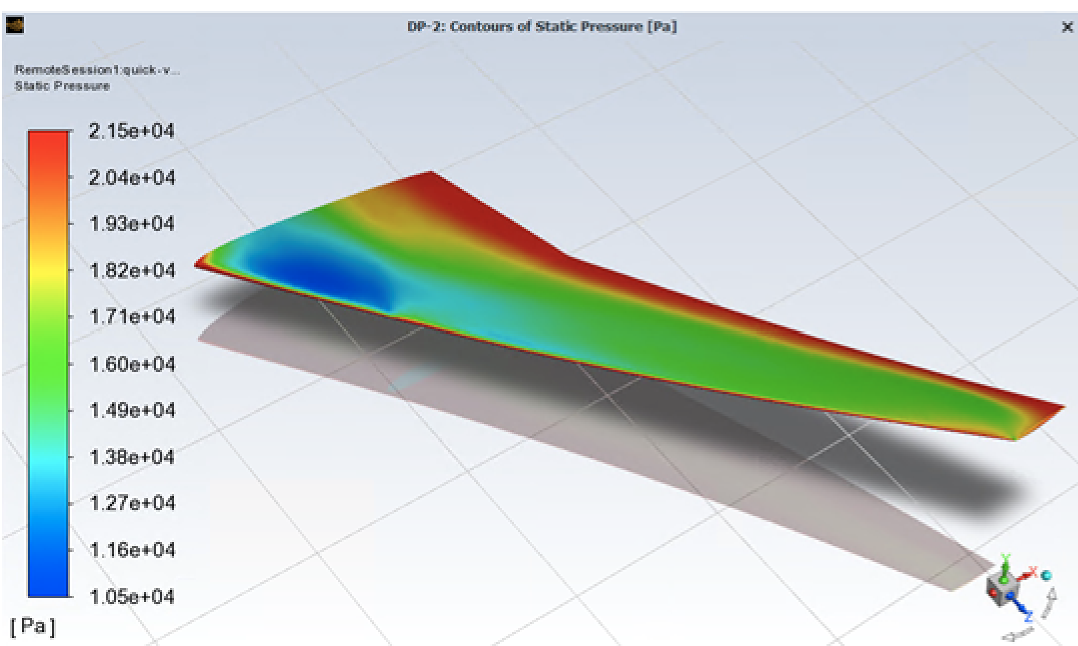



23. The **Contours** options can be used to quickly display simple contour plots of selected design points and solution variables. Left-click on **Contours** from **Outline View** to display the **Properties - Contours** window. If the **Surfaces** option is set to **Walls**, all the wall type boundary zones will be used to display the desired contour. For this tutorial, we will set **Surfaces** to **Component Group** and set **Select a Component Group** to **Wing_01**. This will show the contour of selected parameter on all the boundary zones in the **Wing_01** group. Set **Field** to **Static Pressure** and **Design Point** to **2**. Uncheck the **Auto-Compute Range** option and set the **Minimum** and **Maximum Value** to **`10500`** and **`21500`** respectively.




Click on the **Plot** button. The selected contour will be displayed in the **Graphics** window. In the **Graphics** window, use the mouse to set the view of the contour. As we can see from the contour, some of the common flow features are captured. Due to the flow on the suction side between the fuselage and the nacelle, a strong flow acceleration is present. Flow encounters adverse pressure gradient moving downstream to the trailing edge. At the wing tip, the pressure is higher in the outer region than in the inner region which indicates the presence of transverse flow.

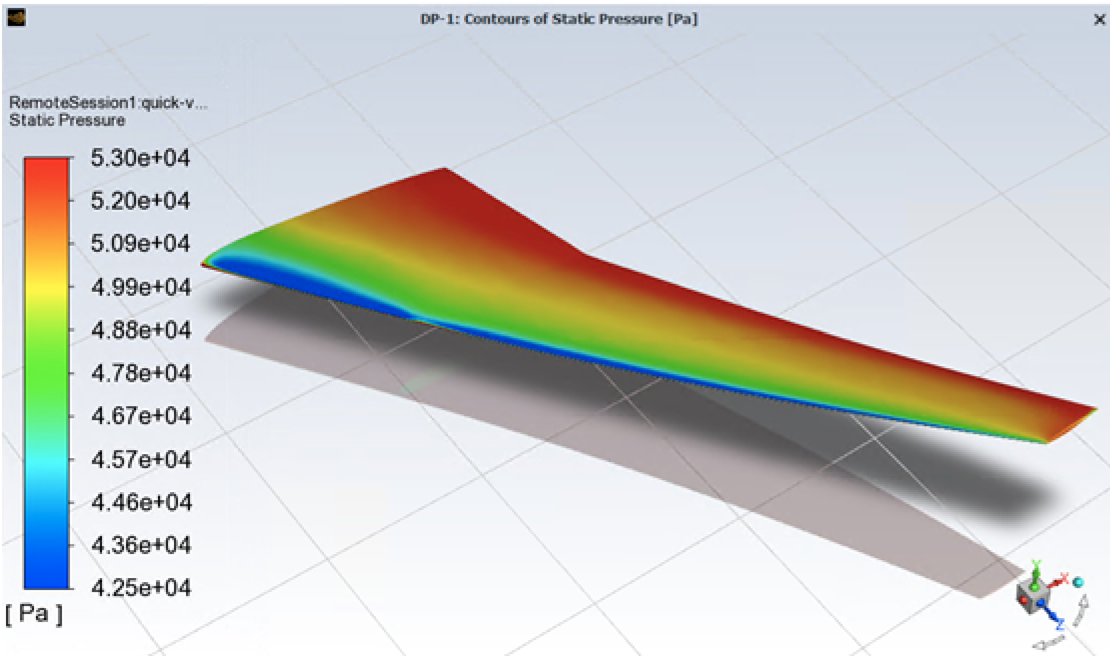
**Figure 33.61: Wall Static Pressure Contour of the Wing_01 Component of Design Point 2**




Click **View...** and change **Save Name** to `crm-wing-view-1`. Click **Save** to save the current contour view. Set **Field** to **Static Pressure** and **Design Point** to **1**. Uncheck the **Auto-Compute**

**Range** option and set the **Minimum** and **Maximum Value** to `42500` and `53000` respectively. Click the **Plot** button to create the contour and then click **View...** to apply the `crm-wing-view-1` to the contour.

**Figure 33.62: Wall Static Pressure Contour of the Wing_01 Component of Design Point 1**




---

**Note:**

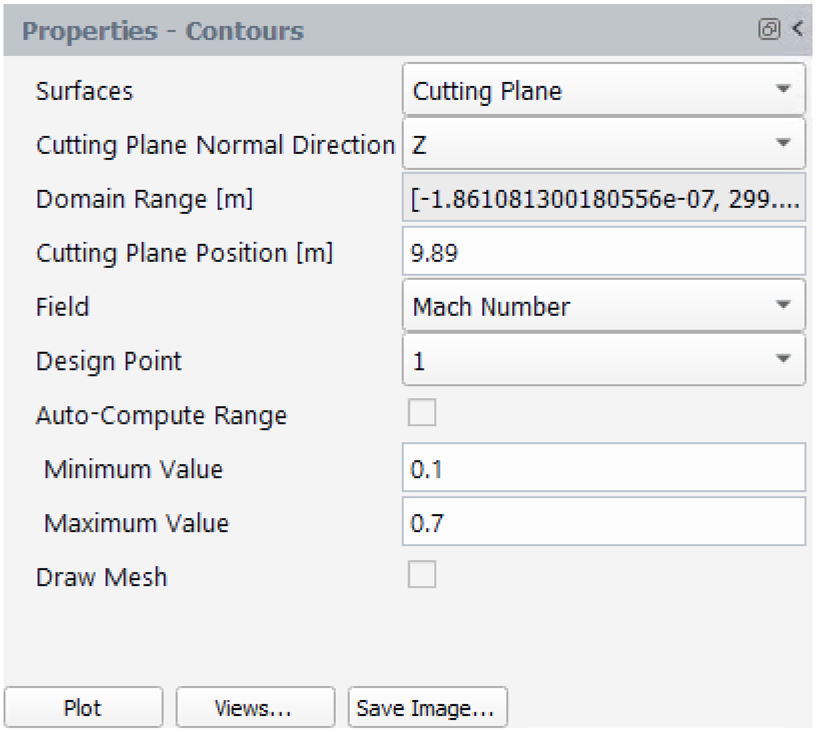
To change graphics display settings, you can go to **File** → **Preferences** → **Graphics**.

For example, in the **Lighting** section, you can set **Headlight** to **On**, and set appropriate values to both **Headlight intensity** and **Ambient light intensity** to personalize the graphics object rendering.

The mesh used in this tutorial is a very coarse mesh. Its purpose is to quickly demonstrate a typical workflow in Fluent Aero and should not be relied upon for accurate simulations. In particular, this mesh may not capture well viscous effects (such as viscous drag) or complex flow features (such as separation). This should be considered while the user investigates the solutions. For more accurate simulations, a finer mesh appropriate for external aerodynamic simulations, featuring more prism layers, higher mesh surface refinement, and increased mesh density in the wake region, should be used.

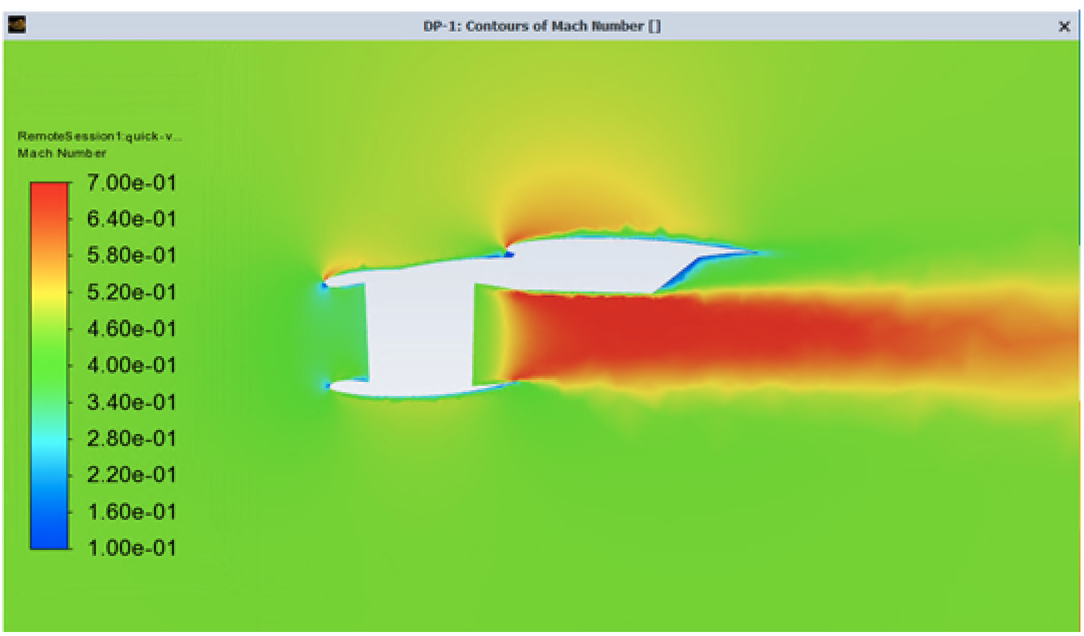
---

24. In the **Properties - Contours** area, set **Surfaces** to **Cutting Plane**, **Cutting Plane Normal Direction** to **Z**, **Cutting Plane Position [m]** to `9.89` which corresponds to the pylon installation position. Set **Field** to **Mach Number** and the **Design Point** to **1**. Uncheck **Auto-Compute Range** and set the **Minimum** and **Maximum Value** to `0.1` and `0.7` respectively.

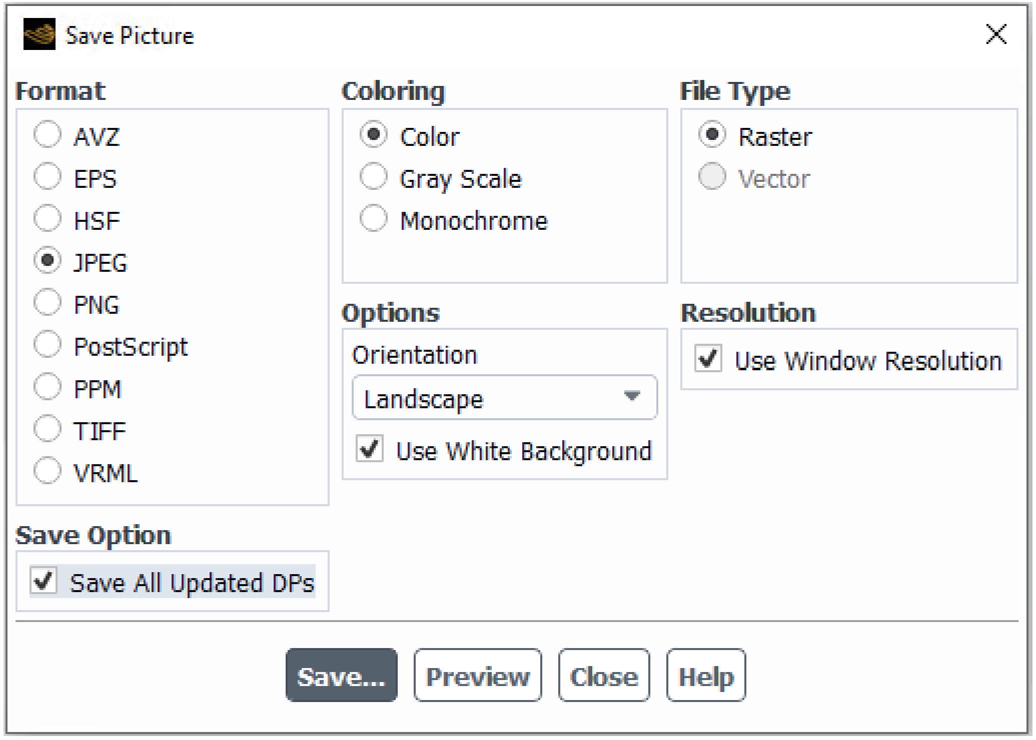



Click on the **Plot** button. The solution of the Design Point 1 will be loaded, and the cutting plane contour will be displayed in the Graphics window.

**Figure 33.63: Mach Number Cutting Plane Contour of Design Point 1**




In the **Properties - Contours** area, click **Save Image...** and a **Save Picture** option panel will appear. From this panel, check to enable **Save All Updated DPs** and set **Format** to **JPEG**. Click the **Save...** button. Set image name to **`CRM-engine-mach-Z9.89m`** from the **Select File** dialog and the contours of the static pressure for all the 2 Design Points will be saved to the **Results** folder as `CRM-engine-mach-Z9.89m-DP-1` and `2.jpg`.


Save Picture
Format
AVZ
EPS
HSF
JPEG
PNG
PostScript
PPM
TIFF
VRML
Coloring
Color
Gray Scale
Monochrome
File Type
Raster
Vector
Options
Orientation
Landscape
Use White Background
Resolution
Use Window Resolution
Save Option
Save All Updated DPs
Save...
Preview
Close
Help

# Chapter 34: Mesh Adaptation With Fluent Icing

The following sections of this chapter are:



## 34.1. Adaptive Simulation: Transonic Turbulent Flow over the ONERA M6

The current section describes how to setup and run a sequence of airflow mesh adaptation cycles within Fluent Icing. For this purpose, the ONERA M6 wing under transonic fully turbulent condition (M = 0.84 and Re = 11.8 million) is used. Despite the simple geometry, a lambda-shock develops over the suction side of the wing. The computational domain of the ONERA M6 wing is made of 1,533,678 nodes and 4,266,117 cells. The wing surface is discretized by approximately 99,000 triangles and the height of the first layer of cells guarantee a Y+ that is below 1 under this condition.

**Figure 34.1: ONERA M6 – Mesh: Farfield (Left) and Wing (Right) Surfaces**

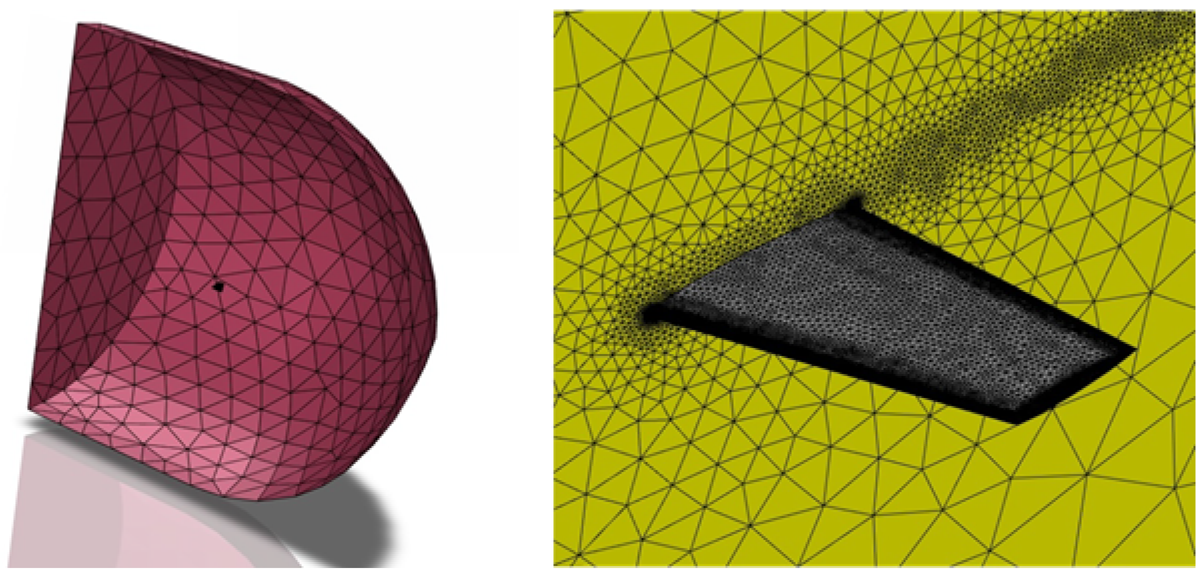
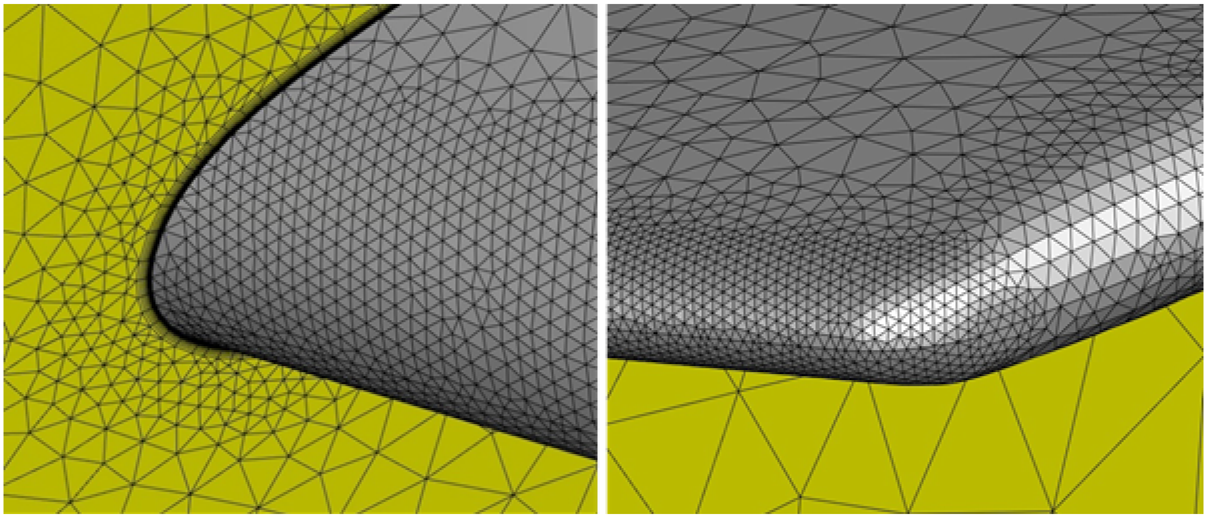


**Figure 34.2: ONERA M6 wing – Mesh Detail: Leading Edge at the Wing Root (Left) and Wing Tip (Right)**



A mesh, such as this one, is considered relatively coarse to capture shocks. Therefore, this tutorial will demonstrate how a sequence of airflow mesh adaptation cycles improves the level of precision of CFD results by capturing not only the lambda shock, but also the wake and wing tip vortex produced by the ONERA M6.

This tutorial is divided into two sections:

1. **Initial Fluent Airflow Simulation**: This section shows how to setup a clean transonic airflow simulation using Fluent Icing and its corresponding Fluent Solution session that is suitable for mesh adaptation.

2. **Sequence of Mesh Adaptation Cycles**: This section demonstrates how to configure Fluent Icing to perform 2 airflow mesh adaptation cycles. Each cycle is composed of an airflow (Fluent) and mesh adaptation (OptiGrid) solver. For more information regarding OptiGrid, see OptiGrid - Mesh Adaptation within the FENSAP-ICE User Manual.

Download the `fluent_adaptation.zip` file here.

Unzip `fluent_adaptation.zip`, preferably inside your working directory.

## 34.1.1. Initial Fluent Airflow Simulation

In this section, you will learn how to set-up a clean transonic airflow simulation within Fluent Icing and its corresponding Fluent Solution session. The latter will allow you to access Fluent settings that are not exposed inside Fluent Icing and that enhance the quality and accuracy of your airflow and adapted mesh results.

The table below shows the transonic conditions of this tutorial. These conditions yield a Mach number of ~0.84 and a Reynolds number of ~11.8 million.

**Table 34.1: Transonic Condition**

| | |
|---|---|
| **Characteristic Length (MAC)** | 0.65 m |
| **Speed** | 269 m/s |

| | |
|---|---|
| **Angle of Attack ($\alpha$)** | 3.06 ˚ |
| **Pressure** | 80,600 Pa |
| **Temperature** | 255 K |

1. Launch Fluent on your computer. In the **Fluent Launcher** window, select **Icing**. Set the number of processes to at least **8** or **12** CPUs. Click on **Show More Options** and, in **General Options**, select a suitable **Working Directory** for your Fluent Icing project. Click **Start** to launch Fluent Icing.

   **Note:**

   On Windows, the path to the working directory must not contain spaces (for instance, `New Folder`)

2. Once Fluent Icing opens, the **Project** tab will be displayed. In the **Project**'s top ribbon panel, select **Project** → **New...** and enter **`OneraM6_Adaptation`** to create a new **Project** folder.

   **Note:**

   A Fluent project allows you to save and manage multiple **Simulations** and **Runs** in a single project folder.

3. From the top ribbon panel, go to **File** and select **Preferences...**. In the **Preferences** window, select **Icing** and inside its panel.

   - Click **Show Solution Workspace** and **Enable Solution Workspace Graphics** to display the Fluent Solution session user interface, including its graphics window, that is connected to Fluent Icing.
   - **Enable Beta features**. Mesh adaptation is a hidden feature that only appears in the graphical user interface of Fluent Icing when **Beta features** are active.
   - Click **Advanced Settings** to access extra smoothing parameters that will be used during mesh adaptation.
   - Press **OK**.

4. Go to the **Project** ribbon menu and select **Simulations** → **Import case** and browse to `fluent_adaptation/workshop_input_files/Input_Grid` and select the `oneram6.cas.h5` file from the extracted `fluent_adaptation.zip` archive.

5. A **New simulation** window will appear. Enter the **Name** of the new simulation as **`oneram6_adaptation`**, and check **Load in Solver** to enable this option. Press **OK**.

   In this manner, a new **Simulation** folder will be created in your **Project** folder, and the `oneram6.cas.h5` file will be imported. You should now see the **oneram6_adaptation (loaded)** tree under the **Outline View** tab.

Moreover, a new Fluent Solution window will appear. This window is connected to your Fluent Icing window.

6. Inside the Fluent Icing window, select the **Setup** tree menu under **oneram6_adaptation (loaded)** and, in the **Properties - Setup** panel, check **Airflow**. Disable all other simulation types.

7. Continue further down and right-click the **Airflow** tree menu located under **Setup** and select **Update with Fluent Case settings** to make sure that the previously Fluent simulation settings are properly transferred to Fluent Icing.

8. Click **Setup** → **Airflow** to display the **Properties - Airflow** panel.

   - Under **General**, select **Fluent** as the **Airflow** Solver.

   - Under **Conditions**, make sure that the airflow conditions under **Conditions** correspond to the transonic conditions mentioned in Table 34.1: Transonic Condition (p. 638). To work in absolute pressure, set the **Operating Pressure [Pa]** to **`0`**. You can verify that these conditions satisfy the Mach number of ~0.84 and Reynolds number of ~1.18 million by looking at the bottom of this submenu.

   - Under **Direction**, go to **Vector Mode** and select **Angle of attack** to specify the orientation of the reference airflow. Set the **Lift Direction** and **Drag Direction** to **Y+** and **X+** respectively. Set the **AoA [deg.]** to **`3.06`** and the **Velocity Magnitude [m/s]** to **`269`** (Table 34.1: Transonic Condition (p. 638)).

9. Click **Setup** → **Airflow** → **Fluent** to display the **Properties – Fluent** panel.

   - Under **Materials**, select **Case settings** next to **Fluid**. This will use the fluid properties of your case file. By default, Fluent Icing only uses constant air properties. In transonic regimes, these properties are not constant. Therefore, to improve accuracy, you will later select temperature dependent air properties inside the Fluent Solution window that is connected to Fluent Icing.

   - Under **Models**, make sure that **Energy**, **Viscous Heating** and **Turb. Production Limiter** are enabled, that **Turbulence** is set to **K-Omega 2-eqn** and that **k-omega Model** to **SST**.

10. Go to **Setup** → **Boundary Conditions**,

    - Inside **Inlets**, click **pressure-far-field-4** to display its properties panel.

      – Under **Airflow**, select **Edit** next to **Conditions**.

      – Go to the bottom of this panel and click **Import ref. conditions**. This will copy the **Conditions** set inside the **Properties – Airflow** panel as boundary conditions.

    - Inside **Walls**, select **wall-5** boundary,

      – Under **Airflow**,

        → Set **Thermal Conditions** to **Heat Flux** and impose a **Heat Flux [W/m2]** of **`0`**.

        → Set the **Wall Roughness** to **Case settings**; all the walls inside the case file were set to **`0`** **Roughness Height [m]**.

- Repeat the process described above for wall boundaries **wall-6** and **wall-7**. All wall surfaces of the ONERA M6 are adiabatic and smooth.

11. Go to the Fluent Solution window that is connected to Fluent Icing and, inside its **Outline View**,

- Double-click **Setup** → **Materials** → **Fluid** → **Air** to display the Material properties of air.

  - Under **Properties**,

    → Set **Density [kg/m$^3$]** to **ideal-gas**.

    → Set **Cp (Specific Heat) [J/kg K)]** to **piecewise-polynomial**.

    → Set **Thermal Conductivity [W/(m K)]** to **kinetic-theory**.

    → Set **Viscosity [kg/(m s)]** to **sutherland**.

    → Leave the other properties as they are.

    → Click the **Change/Create** button to save your changes.

  - These air properties are temperature dependent and have been used in a wide variety of validation cases.

- Go to **Setup** → **Boundary Conditions** → **Wall**

  - Click on **wall-5** and **wall-6** while pressing on the **Ctrl** key of your keyboard. This automatically selects these two boundary surfaces.

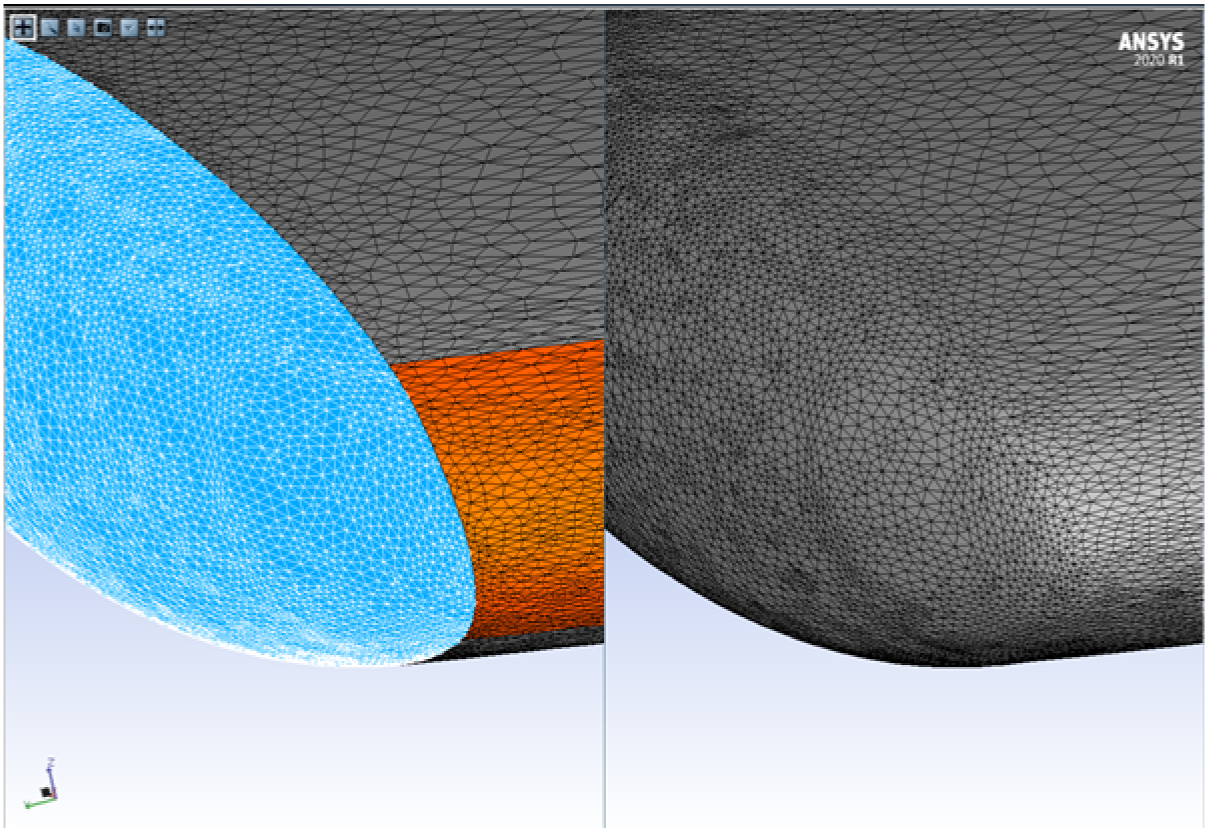
  - Right-click on **wall-6**, a submenu appears. Inside this submenu, select **Merge** to combine these 2 wall surfaces into one.

  - Wall-6 and wall-5 represent the upper and lower surfaces of the ONERA M6 wing. They meet at the leading edge of the wing where the curvature is expected to be continuous. CAD surfaces created internally by OptiGrid for node projection will have discontinuous curvature across this edge, which may produce a visible crease in the mesh after adaptation. To avoid this, such edges can be removed by combining their associated surfaces before mesh adaptation. See the figure below for an example of mesh adaptation with and without this merging step. Wall-7 should not be merged with the other walls as its surface represents a blunt trailing edge and the curvature between the wing and its trailing edge is discontinuous.

  - Double click on **wall-7** and, inside its **Wall** panel, change its **Zone Name** to **trailing-edge**. Click **Apply**.

  - Double-click on **wall-5** or **wall-6** (automatically generated surface name given by Fluent after the merge) and, inside its **Wall** panel, change its **Zone Name** to **wing**. Click **Apply**.

**Figure 34.3: Surface Mesh at the Wing Tip after One Mesh Adaptation: Without (Left) and with (Right) Merging Wall Surfaces**




**Note:**

The imported case file has been set to obtain a robust Pressure-Based 2$^{nd}$ order airflow solution and to monitor convergence of lift and drag coefficients. In this manner,

- Inside **Setup** → **Reference Values**, the **Area [m$^2$]** is set to 0.7728935 in order to compute the aerodynamic coefficients.
- Inside **Solution** → **Methods**, the **Pressure- Velocity Coupling** scheme is set to **Coupled**. In **Spatial Discretization**, the **Gradient** is set to **Green- Gauss Node Based** and the remaining options to **Second Order** or **Second Order Upwind**. **Pseudo Transient** has been enabled as well as **High Order Term Relaxation** to improve convergence.
- Inside **Solution** → **Controls**, the explicit relaxation factors are set to 0.5 for all parameters except **Density**, which is set to 1.
- Inside **Solution** → **Initialization**, **Use External-Aero Favorable Settings** have been enabled in **Hybrid Initialization**.
- Inside **Solution** → **Run Calculation**, the **Time Scale Factor** of the **Conservative Length Scale Method** has been reduced to 0.05.

This setup is used as is by Fluent Icing.

---

12. Go back to the Fluent Icing window and, inside the **Outline View**, right-click on **symmetry-8** under **Setup** → **Boundary Conditions**. Select **Refresh BC list** to sink the boundaries of Fluent Icing with those of Fluent Solution and to reflect the merge and the renaming of wall boundaries.

13. In the ribbon, go to **File** and select **Save Case** to save your changes to the `oneram6.cas.h5` file. This is an important step as it will save your mesh topology changes that will be later used by the mesh adaptation solver.

14. Inside the **Outline View**, select **Airflow** under **Solution**. Inside the **Properties – Airflow** panel, set **Number of Iterations** to **`500`** and press the **Calculate** button located at the bottom of that panel.

15. A **New run** window will appear. Set the **Name** of the new run to **`flow_orig`** and click **Ok**. The calculations will start with hybrid initialization.

16. Monitor the convergence of this calculation inside Fluent Icing. Go to the **Plots** window located at the right of your Fluent Icing window and look at the residuals and aerodynamic coefficient curves. The following three figures show the convergence of residuals and lift and drag coefficients as well as their values at the end of the simulation. After 500 iterations, lift and drag coefficients should have converged to the following values ~0.2514 and ~0.01785 respectively.

**Figure 34.4: Convergence of Scaled Residuals**

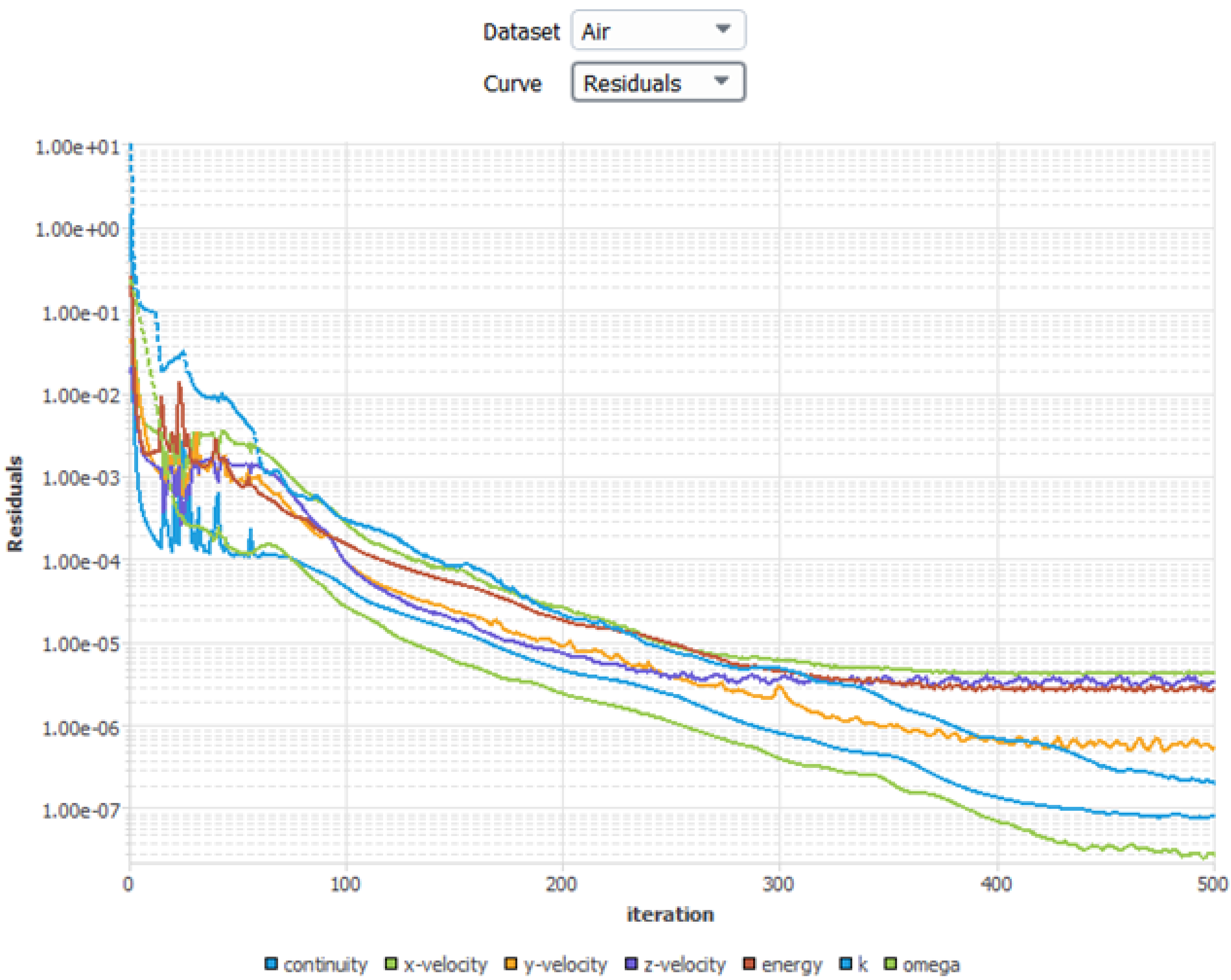



**Figure 34.5: Convergence of Lift and Drag Coefficients**

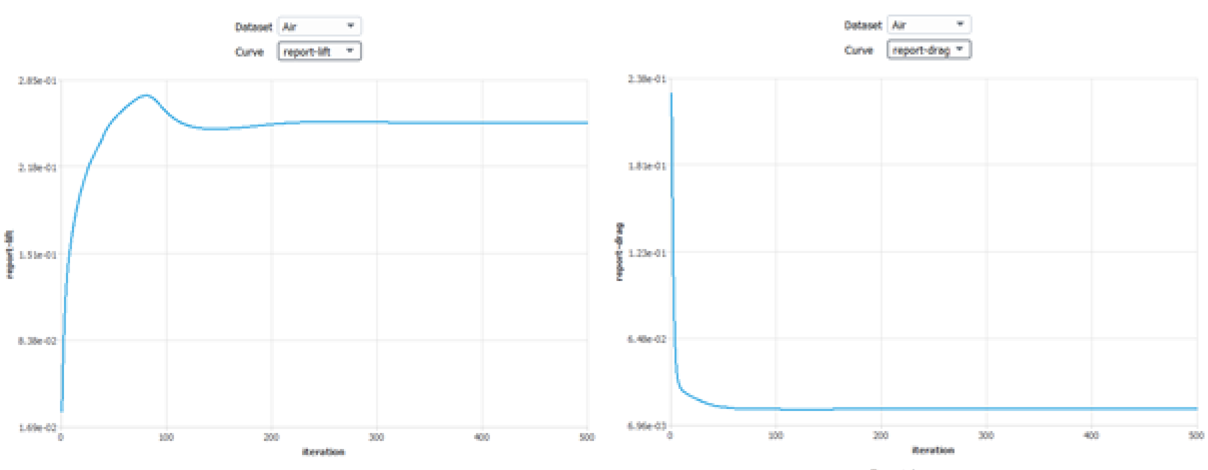


**Figure 34.6: Residual and Lift and Drag Coefficients near the End of the Simulation**

```
iter  continuity  x-velocity  y-velocity  z-velocity      energy          k       omega  report-dra  report-lif     time/iter
 496  7.9506e-08  4.3723e-06  5.9411e-07  3.1514e-06  2.8205e-06  2.1584e-07  2.6109e-08  1.7854e-02  2.5139e-01  0:01:58   4
 497  7.9623e-08  4.3348e-06  5.8751e-07  3.1406e-06  2.6975e-06  2.1585e-07  2.8036e-08  1.7854e-02  2.5139e-01  0:01:29   3
 498  7.9855e-08  4.2986e-06  5.5924e-07  3.2658e-06  2.6971e-06  2.1368e-07  2.9175e-08  1.7854e-02  2.5139e-01  0:00:59   2
 499  8.1037e-08  4.3531e-06  5.3395e-07  3.3482e-06  2.7082e-06  2.1059e-07  2.9272e-08  1.7854e-02  2.5139e-01  0:00:30   1
 500  8.2886e-08  4.2898e-06  5.3743e-07  3.3874e-06  2.8059e-06  2.0490e-07  2.8205e-08  1.7854e-02  2.5139e-01  0:00:00   0
```

17. Once the computation is complete, an **Airflow** icon will appear under the **flow_orig** run folder in the **Project View**. This icon shows that a solution has been obtained and has been saved inside the run directory, **oneram6_adaptation/flow_orig**. The filename of this solution is **oneram6.dat.h5**.

18. Go to **Project View** and right-click on the **Airflow** icon located inside the **flow_orig** run folder. Choose **View result** → **View with Viewmerical** from the drop-down menu. This will open the flow solution using Viewmerical.

19. In Viewmerical, follow these steps to output the pressure coefficient and Mach number.

   - Pressure coefficient

     – To create contours along the surface of the wing and at the symmetry plane (Z = 0 m).

       → Inside the **Objects** panel, disable **BC_1000**. Enable **BC_2000**, **BC_2001** and **BC_4300**.

       → Inside the **Data** panel, set **Data** to **Pressure Coefficient** and change the **Color range** to **Spectrum 2 -32**.

     – To create contours along the surface of the wing and at Z = 0.75 m.

       → Inside the **Objects** panel, disable **BC_1000** and **BC_4300**. Enable **BC_2000** and **BC_2001**. Go to the **Cutting Plane** submenu, select **Z-coordinate** and set **Pos** to `0.75`.

       → Inside the **Data** panel, set **Data** to **Pressure Coefficient** and change the **Color range** to **Spectrum 2 -32**.

     – 2D plots along the wing at Z = 0 m and Z = 0.75 m.

       → Inside the **Query** panel, set **2D Plot** to **Enabled**.

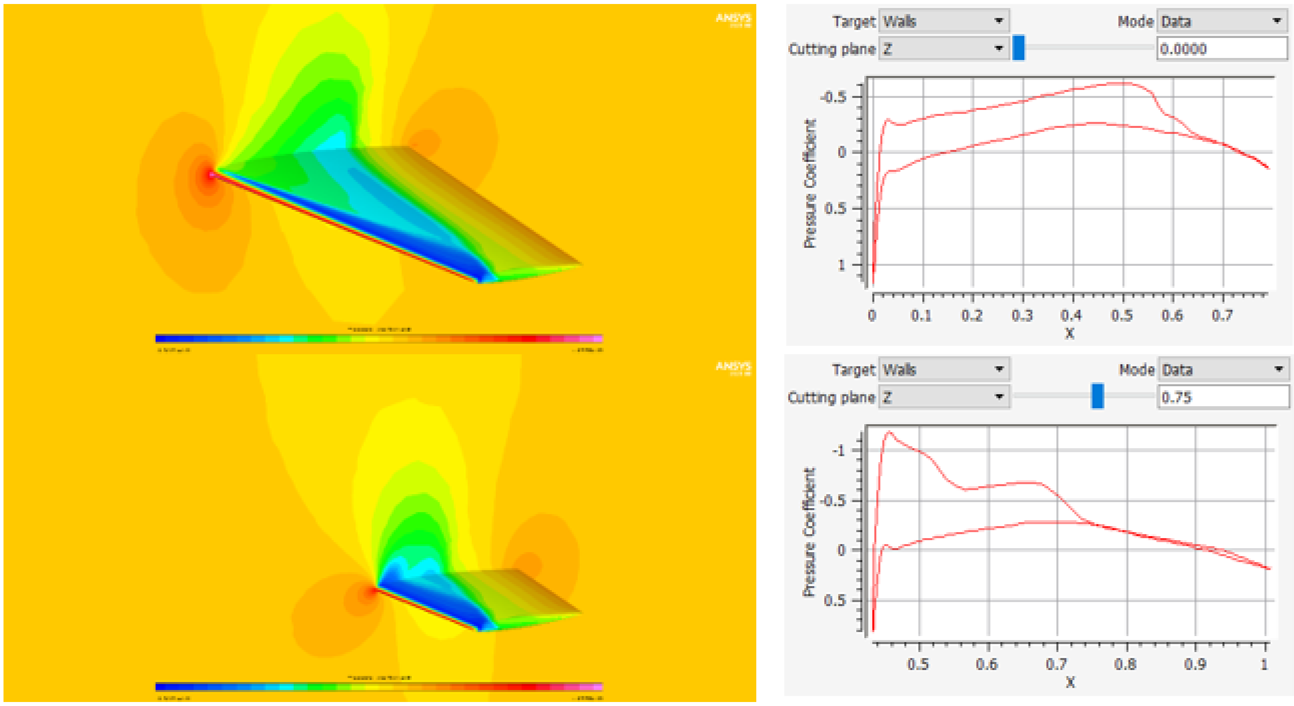
       → Make sure that **Target** is set to **Walls** and that **Cutting plane** is set to **Z**.

       → Select **X** as the horizontal axis. To do so, set **X** in the bottom left corner of your plot.

       → Set the text box to the right of **Cutting plane** to `0` and `0.75` to see the pressure coefficient along the wing at Z = 0 m and Z = 0.75 m respectively.

     – The figure below shows the post-processed figures described in the previous steps.

**Figure 34.7: Pressure Coefficient Contours and 2D Plots at Z = 0 m Plane (Top) and Z = 0.75 m Plane (Bottom)**




- Mach number
  - To create contours at the symmetry plane (Z = 0 m).

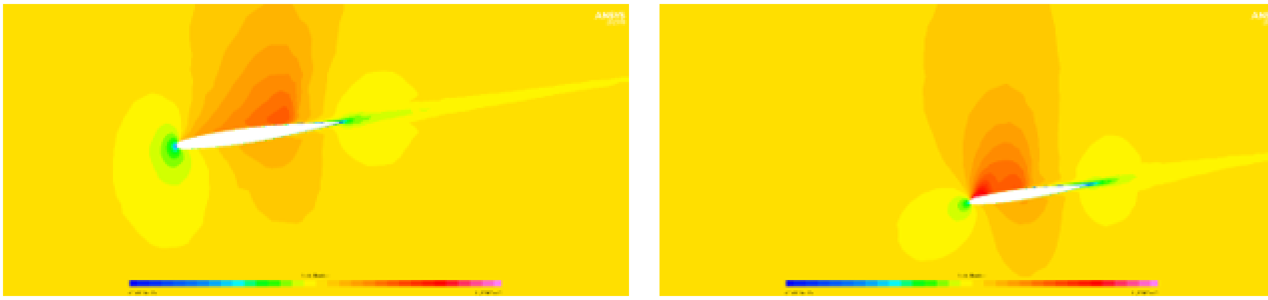
    → Inside the **Objects** panel, disable **BC_1000**, **BC_2000** and **BC_2001**. Enable **BC_4300**.

    → Inside the **Data** panel, set **Data** to **Mach Number** and change the **Color range** to **Spectrum 2 -32**.
  - To create contours at Z = 0.75 m.

    → Inside the **Objects** panel, disable **BC_1000**, **BC_2000**, **BC_2001** and **BC_4300**. Go to **Cutting Plane** submenu, select **Z-coordinate** and set **Pos** to **`0.75`**.

    → Inside the **Data** panel, set **Data** to **Mach Number** and change the **Color range** to **Spectrum 2 -32**.
  - The figure below shows the post-processed figures described in the previous steps.

**Figure 34.8: Mach Contours at Z = 0 m Plane (Left) and Z = 0.75 m Plane (Right)**



As observed in the previous pressure coefficient and Mach figures, the coarse mesh was able to roughly capture the lambda shock of the airflow. The sharp pressure drop across the shocks is not well represented nor is the airflow around the wing and at the wake. Also, Mach number contours reveal too much numerical noise in the solution. In the next section, you will use mesh adaptation cycles to improve this solution.

20. Close Viewmerical without closing Fluent Icing.

## 34.1.2. Mesh Adaptation Cycles

In this section, you will learn how to set-up a sequence of mesh adaptation cycles from an existing airflow solution in order to improve its precision. In this case, two (2) cycles are sufficient to better capture the pressure drop across the lambda shock, the airflow acceleration along curved surfaces such as the leading edge and the wing tip, and the wake produced by the wing.

**Note:**

Inside Fluent Icing, mesh adaptation is done using Ansys OptiGrid. The latter allows anisotropic mesh optimization by aligning and stretching cells and edges with flow features to minimize and homogenize the Hessian-based mesh error of selected solution fields. Only unstructured meshes (prisms and tetrahedrons) are supported. Several mesh operations are done to reduce this error: mesh coarsening, mesh refinement, edge swapping and node movement. The most relevant parameters of Ansys OptiGrid are exposed in Fluent Icing. For more information, see OptiGrid - Mesh Adaptation within the FENSAP-ICE User Manual.

1. After you obtain the airflow solution in Initial Fluent Airflow Simulation (p. 638), in the **Outline View** of Fluent Icing, select **Setup** under **oneram6_adaptation (loaded)**. In the **Properties – Setup** panel, enable **Mesh Adaptation (Beta)** and make sure that **Airflow** is also enabled. This will automatically create a **Mesh Adaptation** entry under **Setup** and **Solution**.

2. Go to **Setup** → **Mesh Adaptation**. In its **Properties** panel,

- Click **Generate Geometry**, located at the bottom of the panel, to create a CAD from the mesh of the `oneram6.cas.h5` file. This CAD contains the surfaces that bound the computational domain and is used as the geometry to preserve in all mesh adaptation cycles.

> **Note:**
>
> Since the CAD is reconstructed from the initial mesh, it is important that this mesh possesses a high surface mesh resolution on rounded surfaces. This will improve the accuracy of the generated CAD. In the current tutorial, this has not been done in order to reduce mesh size and therefore computational time and resources.

- Under **Options**,
  - Set **Mode** to **Solution** to conduct mesh adaptations using airflow solutions.
  - Set **Adaptation Variable** to **Multiple Scalars**.
  - In **Variables**, select **Density**, **Pressure**, **Temperature**, **X Velocity**, **Y Velocity**, **Z Velocity** and **Turbulent Viscosity**. The first 6 fields represent the primitive variables of the airflow solver. The turbulent viscosity is a post-processed field predicted by the turbulence model. Press **OK**.
  - Set **Scalar Smoothing** to **Enabled**. This gives you access to parameters that remove numerical noise or enhance flow discontinuities.
    - → Set **Convolution Iterations** to **`1`**. Convolution is used to filter the solution and remove numerical noise. A value of one represents the number of time steps required to solve a Laplacian equation that smooths the solution fields. This is usually required when the original grid is coarse, which is the case in this tutorial.
    - → Keep **Deconvolution Iterations** and **Post-Deconvolution Iterations** to **`0`**.
- Under **Constraints**,
  - Take a moment to look at the **Current Min. Edge Length** and **Current Max. Edge Length** of the original ONERA M6 mesh. OptiGrid provides these lengths as guidelines to set the minimum and maximum edge length constraints to respect during mesh adaptation.
  - Set the **Min. Edge Length** of cells to **`2e-4`**. This represents approximately a tenth of the thickness of the trailing edge at the wing tip. This is the smallest geometric dimension along the wing and should be enough to highlight the surface mesh refinement improvements made specially at the leading edge of the wing and along the shock.
  - Set the **Max. Edge Length** of cells to **`1`**. This represents approximately a fifth of the current maximum length that is located at farfield. Reducing the maximum length of cells at farfield will improve the transition in mesh refinement between the wing and the farfield as mesh adaptation cycles are executed.
  - Keep **Settings** as **Default**.
- Under **Boundaries**,

  - In **Boundary Correction Zones**, select trailing-edge and wing. This will preserve the prism layer height and growth ratio of the walls selected.
  - In **Frozen Zones**, select **trailing-edge** to prevent surface mesh adaptation on the trailing edge of the ONERA M6. This surface has been specified as a frozen zone to reduce the mesh size of adapted meshes and to show its functionality.

3. Go to **Solution** and, inside its properties window, under **Mesh Adaptation (beta)**, set **Number of Solver-Adaptation Loops** to **2**. This will allow you to conduct two adaptation cycles. Each cycle or loop is composed of an adapted mesh and its corresponding airflow solution.

4. Go to **Solution** → **Mesh Adaptation**. In its properties panel,

   - Under **Target**,
     - Take a moment to look at the **Current Number of Nodes** and **Current Number of Cells** of the ONERA M6 mesh. OptiGrid provides these numbers as guidelines to control the mesh size per mesh adaptation cycle.
     - Set **Mode** to **Number of Nodes** to control the size of the adapted meshes.
     - Set **Target Number Nodes** to `2,500,000` to set the target number of nodes of the first adapted mesh.
     - Set **Additional Nodes per Cycle** to `500,000` to set the number of nodes to add at each mesh adaptation cycle after the first adapted mesh.
   - Under **Controls**,
     - Set **Iterations** to **15**. In this manner, fifteen mesh adaptation iterations will be carried out per mesh adaptation cycle.
     - Set **Mode** to **Advanced**. New options appear that will allow you to control the type of mesh optimization and the number of mesh movement operations per iteration.
     - Set **Type of Optimization** to **Full**. In this mode, all supported mesh operations (add/remove nodes, edge swapping and node movement) can be performed.
     - Set **Node Movement – Pre** to **15**. This sets the maximum number of node movement, edge refinement and swapping loops to smooth and optimize the mesh based on different constraints per iteration.
     - Set **Node Movement – Post** to **75**. This sets the maximum number of node movement loops to smooth the adapted mesh per iteration.
     - Set **Edge Swapping** to **5**.
     - Enable **Adapt on Curvature** and disable **Compute Error Distribution**.

**Note:**

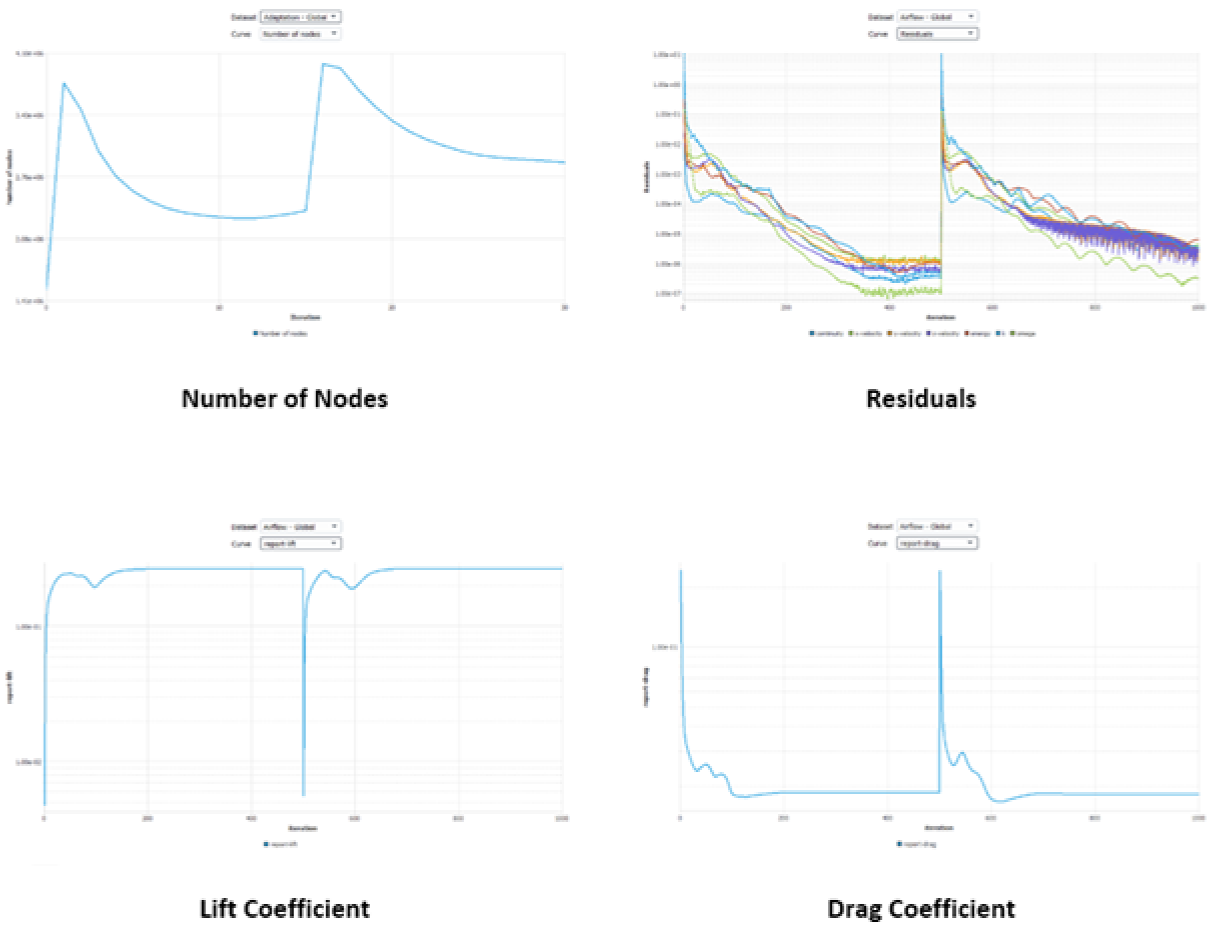
**Compute Error Distribution** is currently supported for single mesh adaptations.

5. In the **Outline View**, right-click **Solution** and select **Run adaptation loop**. A **New run** window will appear. Set the **Name** of the new run to **`adapt_loop_oneram6`**. Press **OK**.

6. Monitor the convergence of this calculation inside Fluent Icing. Go to the **Plots** window located at the right of your Fluent Icing window and look at convergence plots of mesh adaptation and airflow by navigating through the **Dataset** list. **Adaptation – Global** and **Airflow – Global** contain the convergence history of all adaptation cycles per solver. Inside this list, numbers displayed next to Adaptation or Airflow link an adaptation cycle to its corresponding convergence plot.

   The following figures show the convergence of the number of nodes, residuals and lift coefficient during adaptation cycles.

   **Figure 34.9: Convergence Plots of Mesh Adaptation and Airflow Solvers during Mesh Adaptation Cycles**


   As the mesh size increases and regions become finer with mesh adaptation, the initial convergence settings, in this case, the time scale factor and the relaxation parameters, should be adjusted and become more robust in order to guarantee good convergence of the airflow solver throughout the mesh adaptation cycles/loops. In this tutorial, a time scale factor of 0.05 guarantees a good convergence of the airflow within 500 iterations as residuals are below 1e-5 and lift and drag coefficients plateaued. The table below shows the lift and drag coefficients produced by the original mesh and each mesh adaptation cycle.

**Table 34.2: Aerodynamic Coefficients during Mesh Adaptation Cycles**

| | $C_L$ | $\Delta C_L$ | $C_D$ | $\Delta C_D$ |
|---|---|---|---|---|
| **Original Mesh** | 0.251391 | --- | 0.017854 | --- |
| **1st Adapted Mesh** | 0.264926 | 5.38% | 0.018516 | 3.71% |
| **2nd Adapted Mesh** | 0.266362 | 0.54% | 0.018207 | 1.67% |

This table shows that differences in lift and drag coefficients between successive adapted meshes decay with mesh adaptation.

### Note:

If more adaptation cycles are sought, for instance to reduce the difference in drag coefficient below 1%, the time scale factor of the airflow solver should be further reduced and its number of iterations per cycle should increase consequently. Moreover, it is recommended to either increase the number of adaptation iterations per cycle and/or to increase the number of node movement pre- and post- iterations to guarantee a smooth mesh during adaptations.

If accurate mesh independent solutions are sought, enable surface mesh adaptation at the trailing edge. This will require a larger target number of nodes.

7. Once all simulations are completed, load the two adapted solutions and the original solution inside Viewmerical. To do this, go to **Project View** and,

   - Right-click the **Airflow** file located inside the **flow_orig** subfolder. Select **View result** → **View with Viewmerical**.
   - Right-click the **Airflow** file located inside the **adapt-loop-oneram6/out_01** subfolder. Select **View result** → **View with Viewmerical**. A message asking if you would like to append this solution to Viewmerical will appear. Click **Yes** to load this solution into the same Viewmerical window that contains the original solution.
   - Repeat the above step for the **Airflow** solution file located inside the **adapt-loop-oneram6/out_02** subfolder.

8. Rename the first solution that was loaded to `orig`, the second to `1st_adapt` and the third to `2nd_adapt`. Compare pressure coefficient contours over the wing, symmetry plane and the Z = 0.75m plane between `orig` and `2nd_adapt` solutions. Enable **Shaded** and **Shaded + Wireframe** in **Object** to see the contours in standalone mode and contours with their respective surface mesh. Moreover, output their 2D plots at Z = `0` m and Z = `0.75` m. You should obtain the following figures.

**Figure 34.10: Pressure Coefficient Contours and Surface Mesh Along the Wing and Symmetry Plane - Original Solution (Left) and 2nd Adapted Solution (Right)**

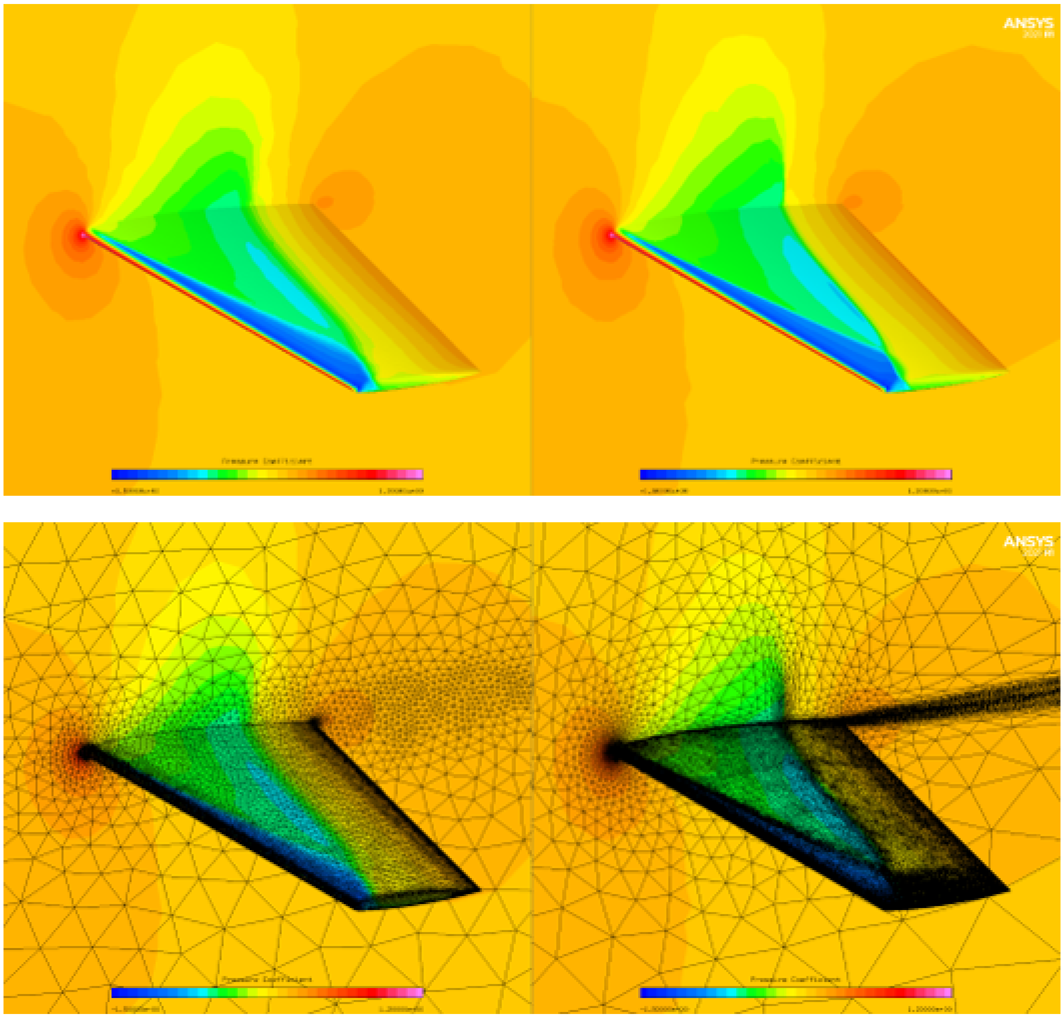


**Figure 34.11: Pressure Coefficient Contours and Surface Mesh Along the Wing and Z = 0.75 m Plane - Original Solution (Left) and 2nd Adapted Solution (Right)**

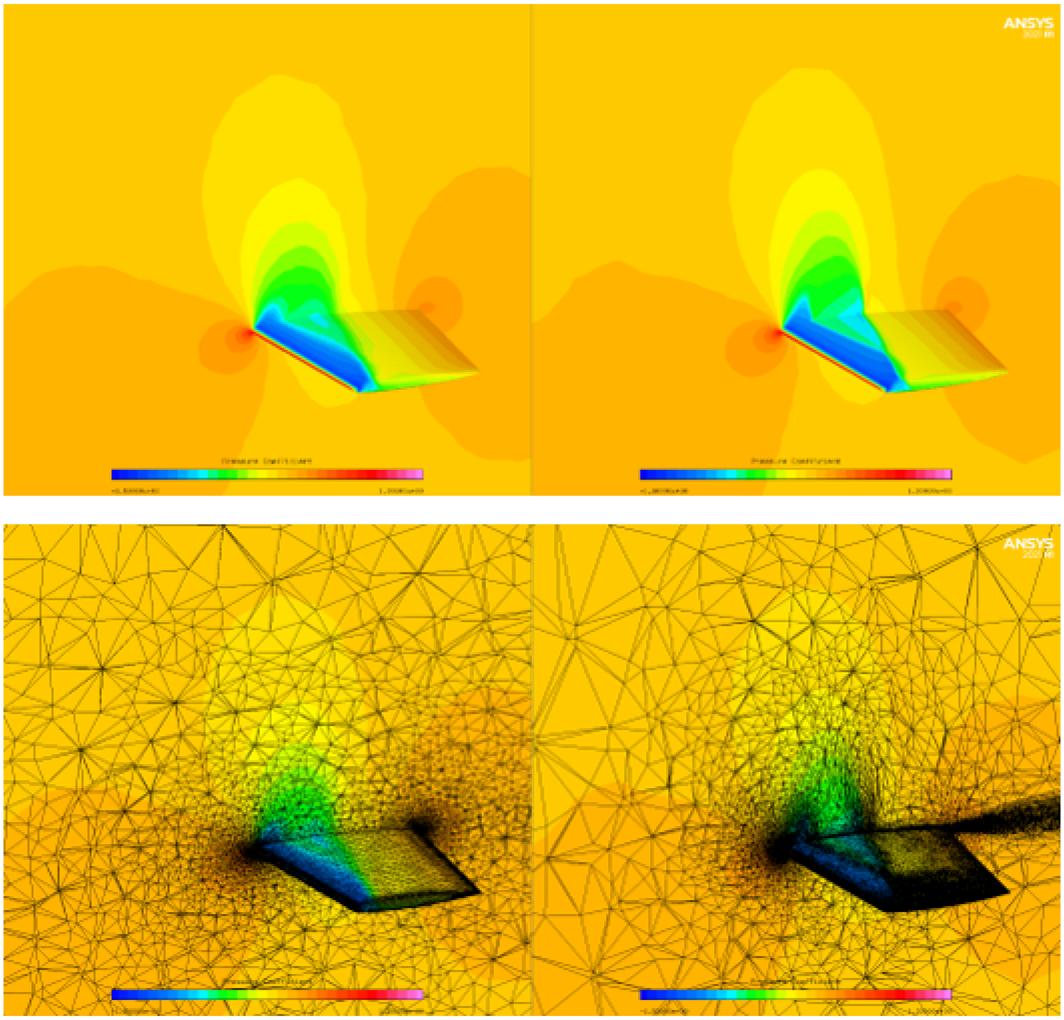
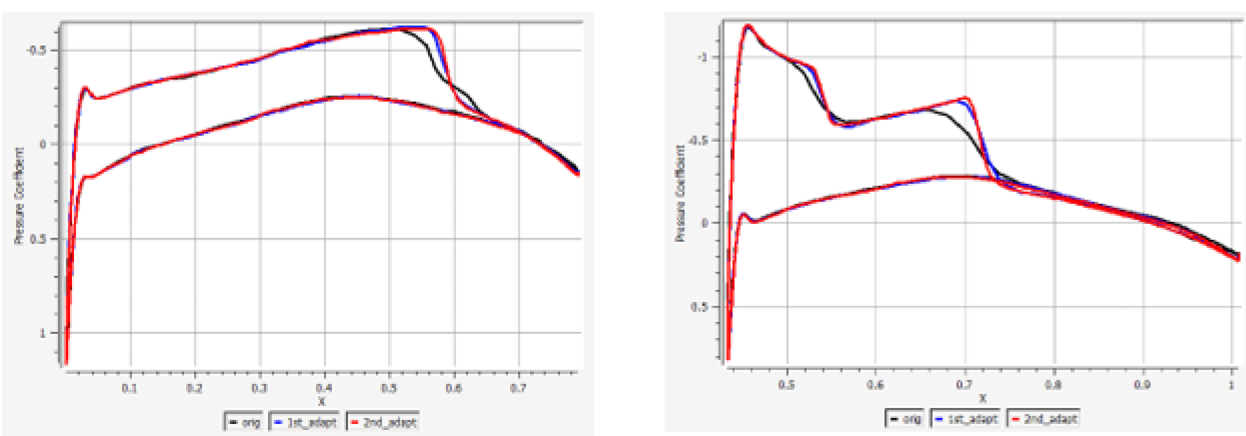
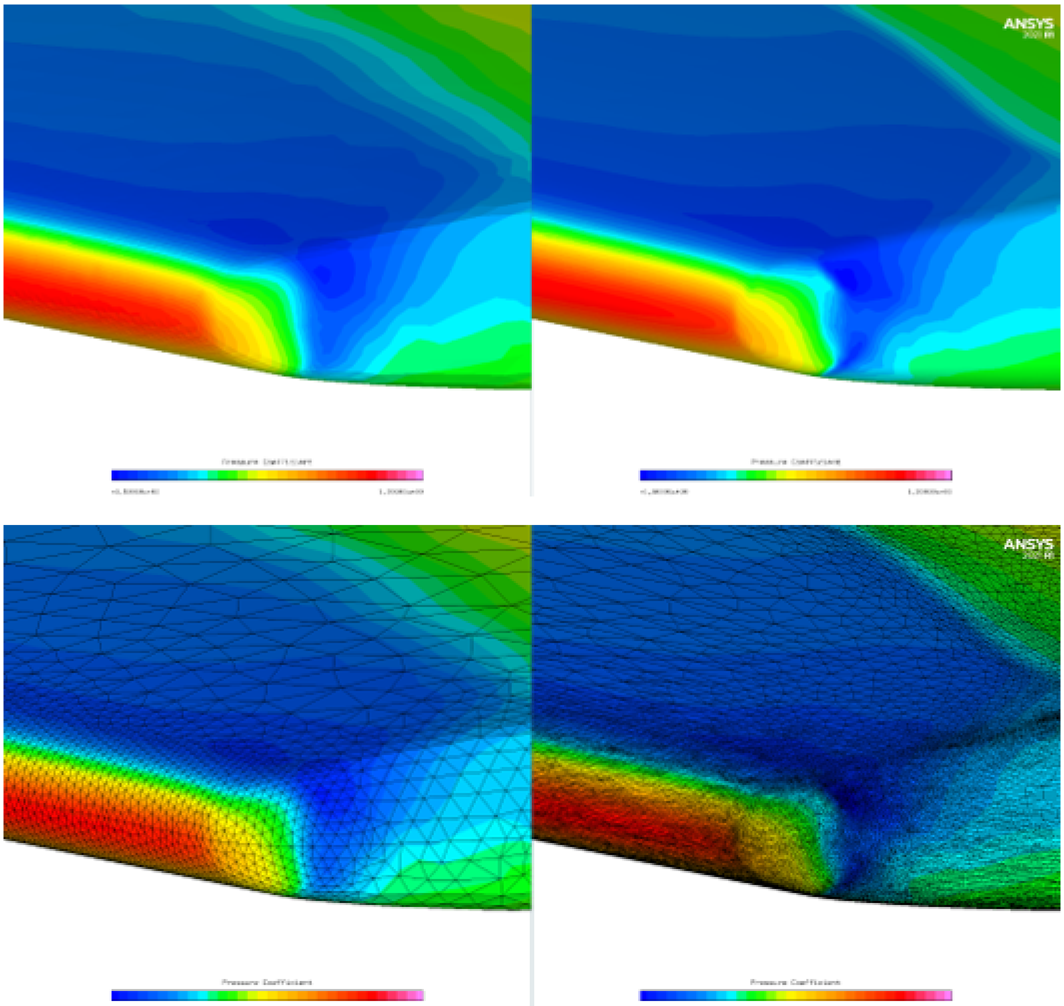


**Figure 34.12: 2D Surface Pressure Coefficient Distributions Along the Wing at Z = 0 m (Left) and Z = 0.75 m (Right)**




As seen from the above figures, mesh adaptation further refines the mesh at the location of the lambda shock as well as the wake produced by the wing. This extra refinement allows the airflow solver to capture the abrupt pressure drop across the lambda shock (see 2D plot figures). Other wing locations have also been refined, for instance, the leading edge and the wing tip. In View-merical, zoom in near the leading edge of the wing tip. You should obtain a figure similar to the one shown below.

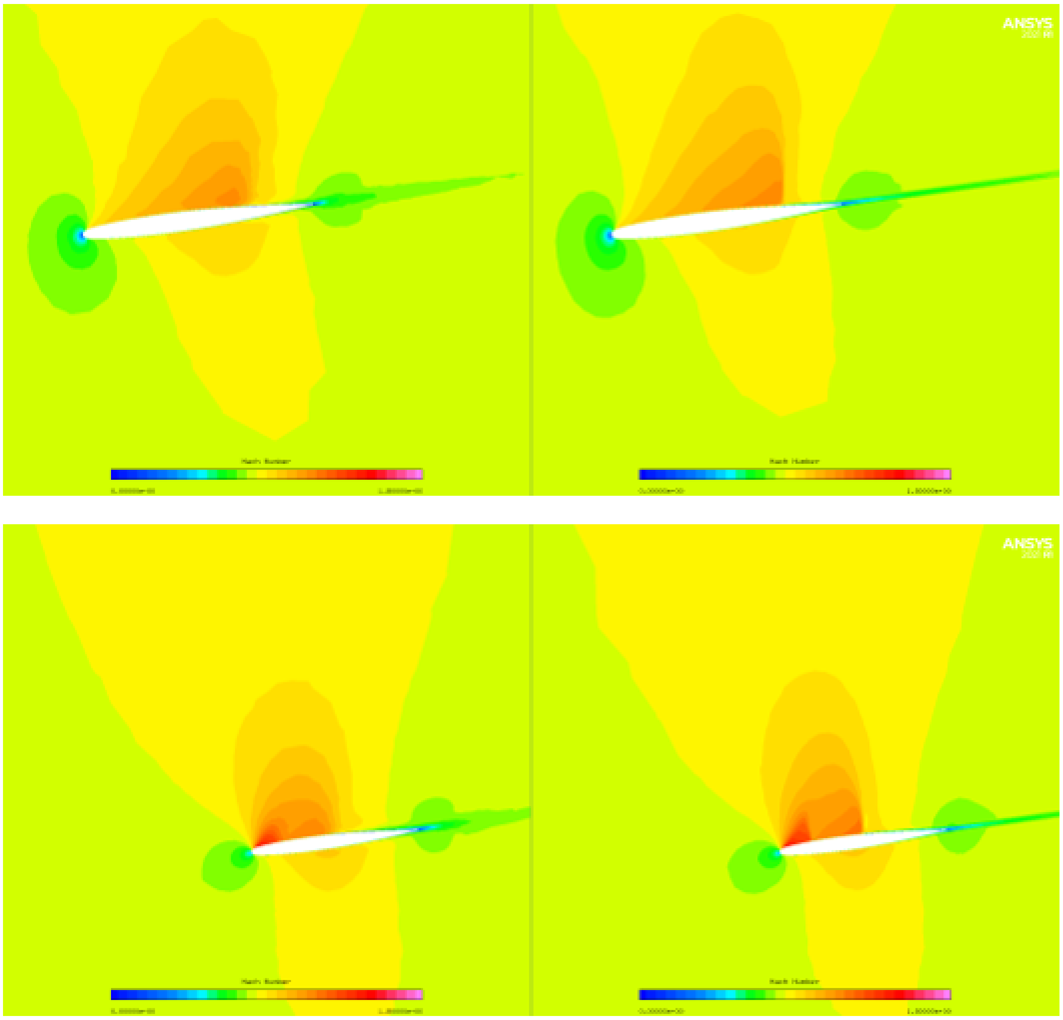
**Figure 34.13: Pressure Coefficient Contours and Surface Mesh Along the Leading Edge of the Wing Tip - Original Solution (Left) and 2$^{nd}$ Adapted Solution (Right)**




This figure shows how mesh adaptation improved the mesh refinement of the curvature of the leading edge and wing tip. As a result, a more continuous distribution of pressure coefficient along these high velocity gradient locations can be achieved.

9. Compare Mach number contours at the symmetry plane and the Z = 0.75 m plane between the original and the final adapted solution. Enable **Shaded** in the **Object** panel to only see the contours. You should obtain the following figures.

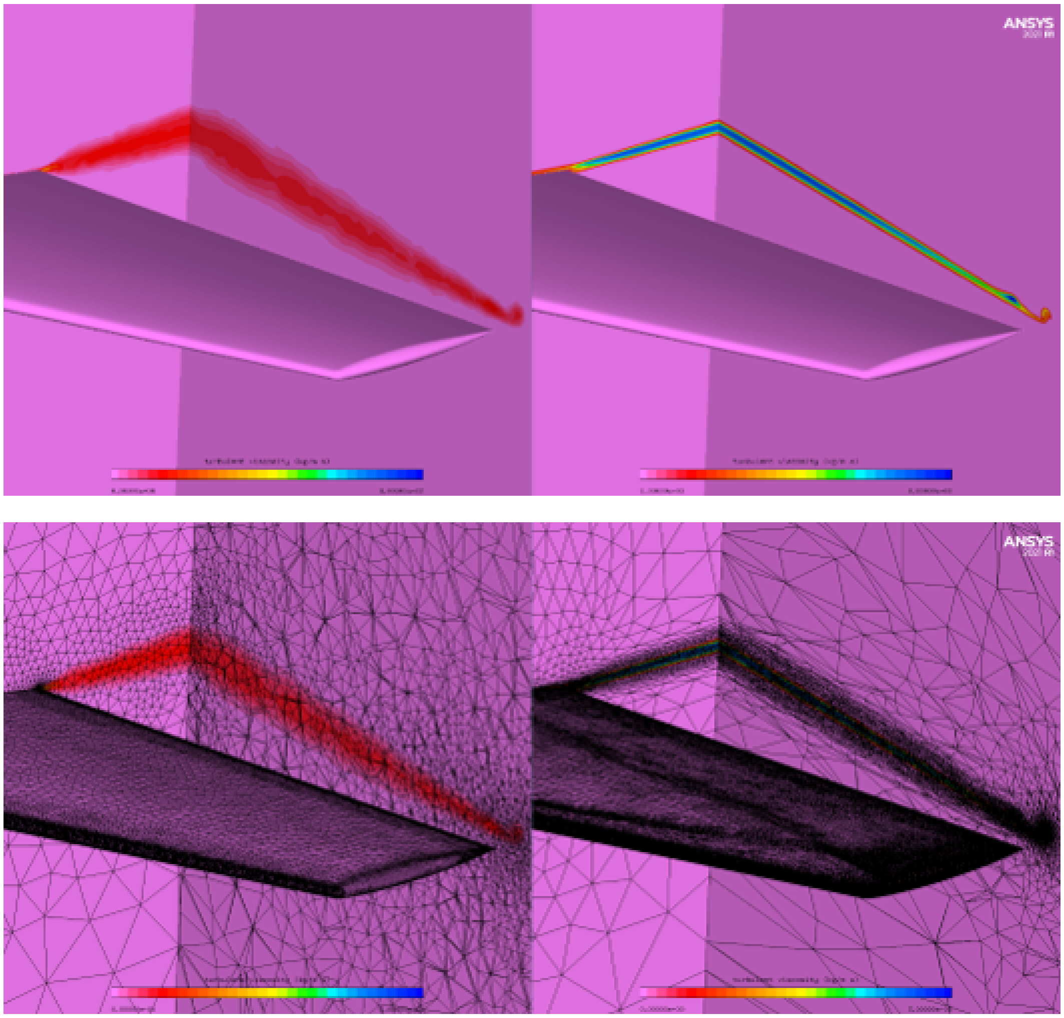
**Figure 34.14: Mach Number Contours at the Symmetry Plane (Top) and the Z = 0.75 m Plane (Bottom) - Original Solution (Left) and 2nd Adapted Solution (Right)**



Comparison of Mach number contours show that mesh adaptation provides a clear representation of the airflow acceleration around the wing and the sudden deceleration across the shocks. Numerical noise and dissipation have been extensively reduced and the wake has been neatly captured.

10. Compare turbulent viscosity contours at the symmetry plane and the X = 1.2 m plane between the original and the final adapted solution. Enable **Shaded** and **Shaded + Wireframe** in **Object** to see the contours in standalone mode and contours with their respective surface mesh. You should obtain the following figures.

**Figure 34.15: Turbulent Viscosity Contours and Surface Mesh Along the Wing, Symmetry Plane and X = 1.2 m Plane - Original Solution (Left) and 2nd Adapted Solution (Right)**




Zoom in near the location of the wing tip vortex on the X = 1.2m plane to have a better view of the mesh refinement and the turbulent viscosity at this location. Go back to the **Data** panel and change **Data** to **Velocity vectors** to clearly see the orientation of the airflow recirculation at the wake.

**Figure 34.16: Turbulent Viscosity at X = 1.2m Plane, Wing Tip Vortex - Original Solution (Left) and 2nd Adapted Solution (Right)**

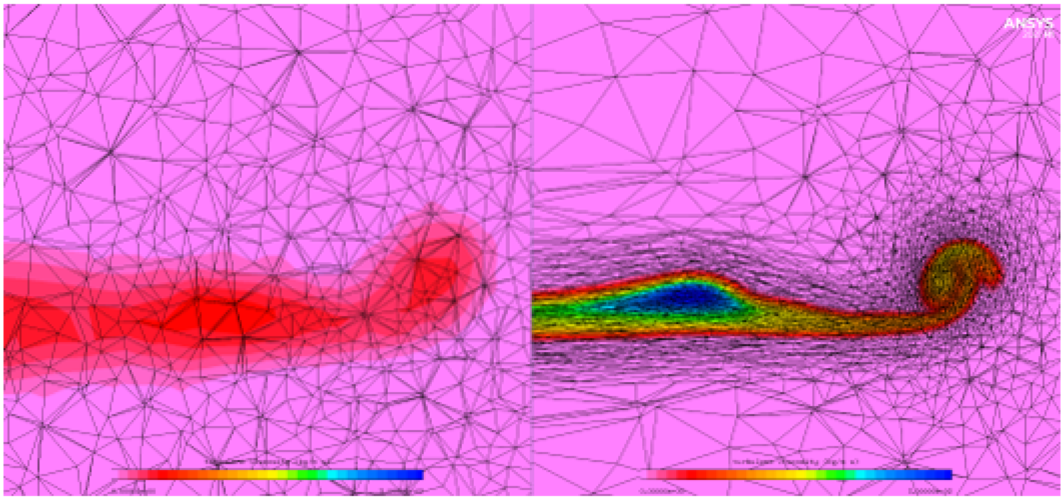
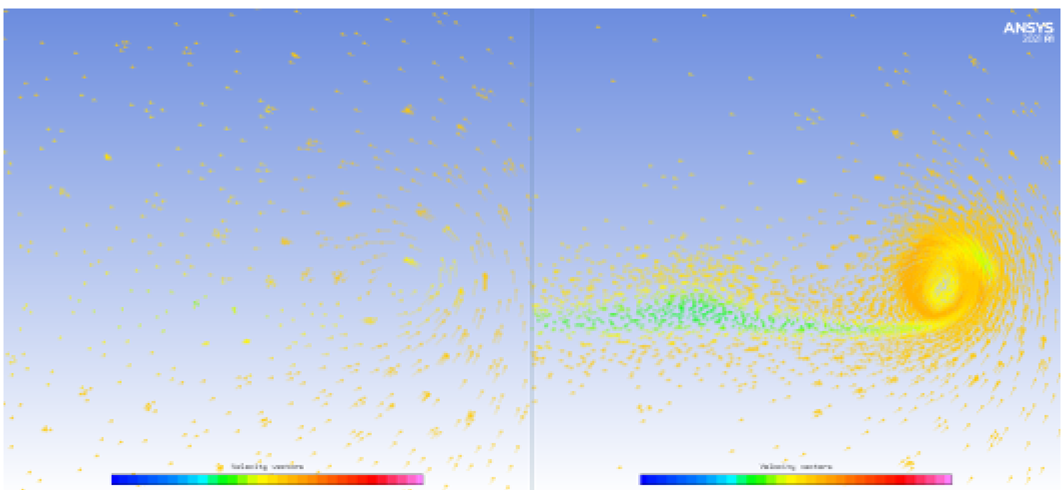


**Figure 34.17: Velocity Vectors at X = 1.2m Plane, Wing Tip Vortex - Original Solution (Left) and 2nd Adapted Solution (Right)**



The above figures show the level of mesh refinement that was added during mesh adaptation cycles in order to precisely capture the wake and the wing tip vortex produced by the wing. These phenomena are important to capture especially for high lift configurations or aircraft components that are located in the wake of upstream surfaces.