Tutorial: Temperature Dependent Viscosity

Introduction

This tutorial examines the flow of liquid metal through a two dimensional channel. The viscosity of the liquid metal is modeled as a function of the temperature using a user-defined function (UDF).

Prerequisites

This tutorial assumes that you are familiar with the FLUENT interface and that you have a good understanding of the basic setup and solution procedures. If not, the FLUENT Tutorial Guide will provide you with the necessary experience. If you have not used UDFs, it would be helpful to refer to the FLUENT UDF Manual.

Problem Description

The problem considered in this tutorial is shown schematically in Figure 1. As the symmetry condition is imposed at the centerline, only half the channel is modeled, The wall of the channel is split into two parts: `wall-2`, which has a temperature of 280 K and `wall-3`, which has a temperature of 290 K. The temperature-dependent viscosity of the liquid metal will respond to this change in wall temperature.

The function, named cell_viscosity, is defined on a cell using DEFINE_PROPERTY. Two real variables are introduced: temp, the value of C_T(cell, thread), and mu, the laminar viscosity computed by the function. The value of the tempertaure is checked, and based upon the range into which it falls, the appropriate value of mu is computed. At the end of the function, the computed value for mu is returned to the solver.

The molecular viscosity of the liquid metal will be defined as a function of temperature, The relationship is given as follows:

$$\mu = \left\{ \begin{array}{cc} 5.5 \times 10^{-3} \text{ kg/m-s} & \text{T} > 288 \text{ K} \\ 143.2135 - 0.49725 \text{ T kg/m-s} & 286 \leq T \leq 288 \ K \\ \text{T} < 286 \text{ K} & 1 \text{ kg/m-s} \end{array} \right\} \tag{1}$$

where,

T = temperature of the fluid (K)

μ = molecular viscosity of the fluid (kg/m-s).

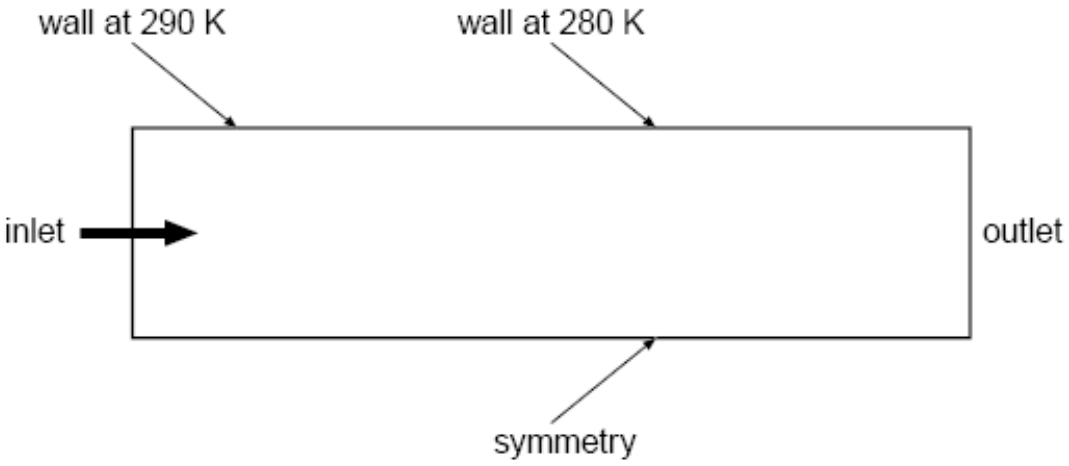


Figure 1: Schematic of the Problem

Preparation

1. Copy the files `user-vis.msh` and `viscosity.c` to your working folder.
2. Start the 2D (2d) version of FLUENT.

Setup and Solution

Step 1: Grid

1. Read the mesh file (`porous_plug.msh`).
2. Check the grid.
   Grid ⟶ Check
3. Display the grid (Figure 2).
   Display ⟶ Grid...

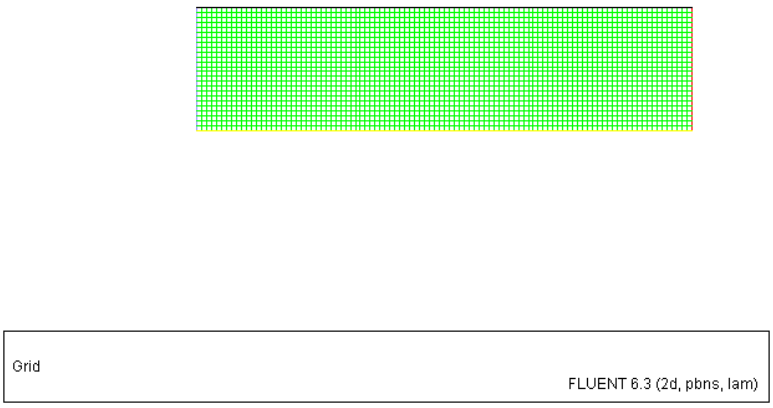


Figure 2: Grid Display

Step 2: Models

1. Retain the default solver settings.
2. Enable energy equation.
   Define ⟶ Models ⟶ Energy...

Step 3: Interpret the UDF

1. View the UDF.

   The user defined function, `viscosity.c` *is used to specify the temperature-dependent molecular viscosity in each cell. In a separate editor, you can view the contents of the UDF to understand its structure and function.*

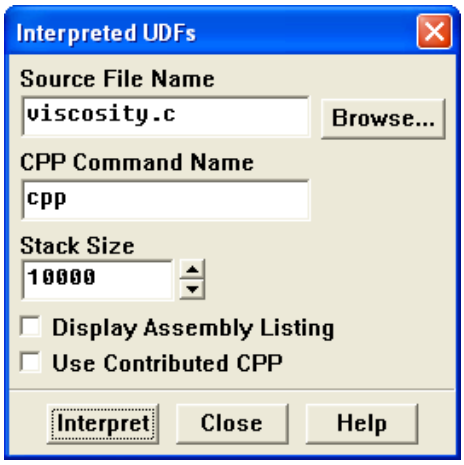
   The contents of the UDF file are as follows:

```
/************************************************************/
/*                                                          */
/* User-Defined Function for temperature-dependent viscosity */
/*                                                          */
/* Author: Liz Marshall                                     */
/*                                                          */
/************************************************************/

#include "udf.h"


DEFINE_PROPERTY(user_vis, cell, thread)

{

  float temp, mu;

  temp = C_T(cell, thread);

  {

/* If the temperature is high, use a small, constant viscosity */

  if ( temp > 288. )
      mu = 5.5e-3;

  else if ( temp >= 286. )
      mu = 143.2135 - 0.49725 * temp;

  else
      mu = 1.0;
}
  return mu;
}
```

   The above equations are applied to each and every cell which is associated to the thread (in this case a fluid zone). This user defined function will be called from the materials panel where the user defined function is specified for viscosity.

2. Interpret the UDF.

   Define ⟶ User-Defined ⟶ Functions ⟶ Interpreted...



(a) Click the Browse... button to open the Select File dialog panel.

(b) Select the file viscosity.c.

(c) Specify the C preprocessor to be used for CPP Command Name.

If you want to use the C preprocessor that Fluent Inc. has supplied, instead of using your own, you can select the Use Contributed CPP *option.*

(d) Retain the default Stack Size of 10000, unless the number of local variables in your function will cause the stack to overflow.

The Stack Size *should be set to a number that is greater than the number of local variables used.*

(e) Click Interpret.

Step 4: Materials

Define ⟶Materials...

1. Modify the fluid material.
   (a) Change Name to `liquid_metal`.
   (b) Enter 8000 for Density and 680 for Cp.
   (c) Enter 30 for Thermal Conductivity.
   (d) Select user-defined from the Viscosity drop-down list and click the Edit... button to open the User-Defined Functions panel.

i. Select user_vis.

ii. Click OK to close the User-Defined Functions panel.

(e) Click Change/Create and close the Materials panel.

Step 5: Boundary Conditions

Define ⟶ Boundary Conditions...

1. Set the boundary conditions for wall-2 zone.
   (a) Click the Thermal tab and select Temperature from the Thermal Conditions list.
   (b) Enter 280 K for Temperature.
   (c) Click OK to close the Wall panel.
2. Set the boundary conditions for wall-3 zone.
   (a) Click the Thermal tab and select Temperature from the Thermal Conditions list.
   (b) Enter 290 K for Temperature.
   (c) Click OK to close the Wall panel.
3. Set the boundary conditions for velocity-inlet-6 zone.
   (a) Select Components from the Velocity Specification Method drop-down list.
   (b) Enter 0.001 m/s for X-Velocity.
   (c) Click OK to close the Velocity Inlet panel.
4. Set the boundary conditions for pressure-outlet-7 zone.
   (a) Click the Thermal tab and enter 290 K for Backflow Total Temperature.
   (b) Click OK to close the Pressure Outlet panel.
5. Retain the default settings for the other zones.
6. Close the Boundary Conditions panel.

Step 6: Solution

1. Initialize the flow field from velocity-inlet-6.

   Solve ⟶ Initialize ⟶ Initialize...

2. Enable the plotting of residuals.

   Solve ⟶ Monitors ⟶ Residual...

3. Start by requesting 300 iterations (Figure 3).

   Solve ⟶ Iterate...

   The solution converges in approximately 200 iterations.

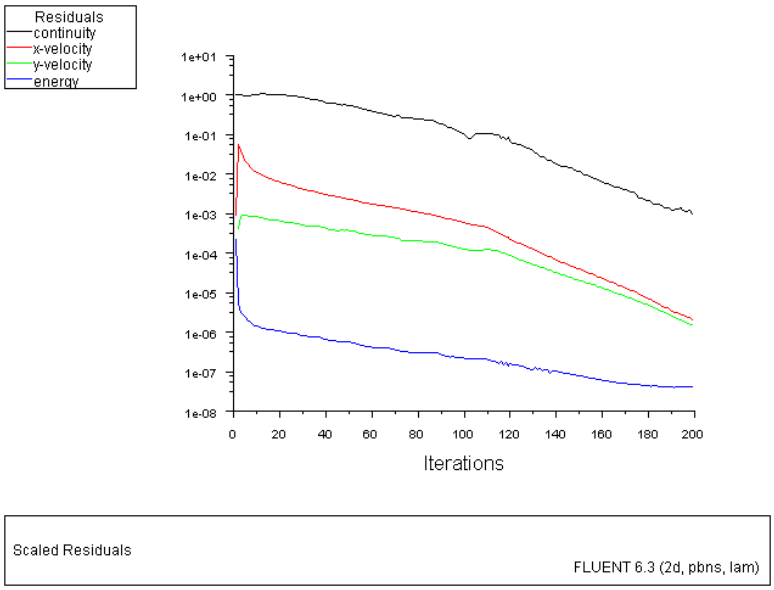


Figure 3: Scaled Residuals

Step 7: Postprocessing

1. Display filled contours of molecular viscosity (Figure 4)

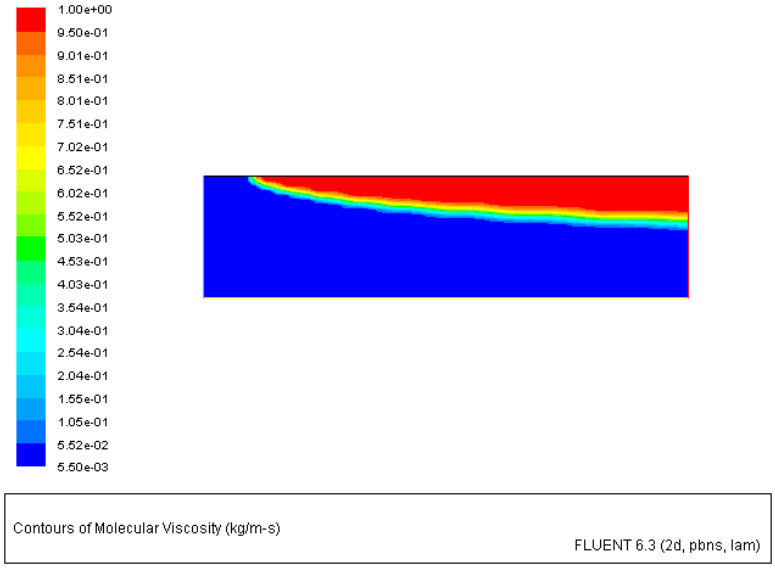


Figure 4: Contours of Molecular Viscosity

Results

Figure 4 shows that when the warmer fluid enters the channel from the left and encounters the cooler wall further on, its viscosity increases according to the user defined viscosity relation.

Summary

This tutorial demonstrated the use of UDFs for specifying a user defined property.

Note: *This capability is available for viscosity and thermal conductivity only (density and specific heat cannot be specified through UDFs).*

Extra: *When you are comfortable with the present exercise, you can try modifying the UDF to specify temperature-dependent thermal conductivity. you can copy the source code for the viscosity UDF and change the appropriate lines to calculate thermal conductivity. The thermal conductivity UDF is then accessed in the materials panel in the same way as viscosity.*