ACFD/Heat Transfer Case Involving a Long, Narrow Rectangular Channel, Illustrating the Capabilities Various Analysis Packages Included with CAELinux 2010

By:

Charles Warner

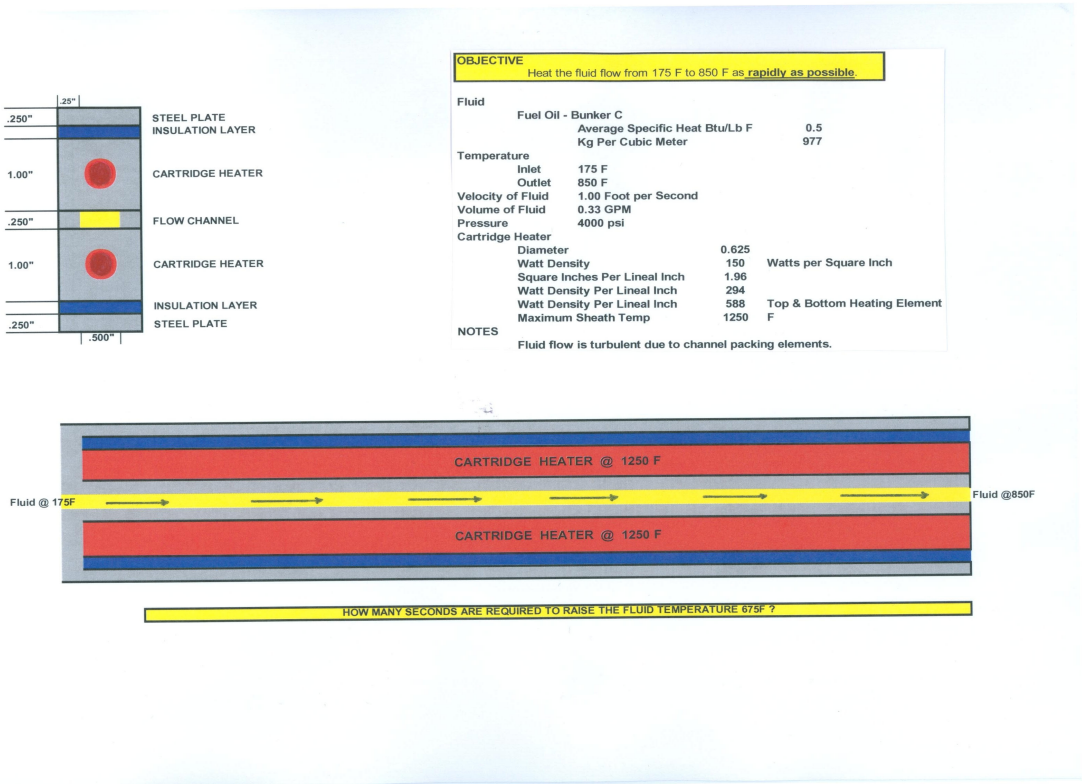
April 21, 2011

CW Engineering, SA

6578-2124/cwarner7_11@hotmail.com

Problem Statement

Our investigation involves heating a fluid (similar to Bunker 3) from 175 ºF (80 ºC) to 850 ºF (454.44444 ºC) in a 1/4" x 1/2" rectangular capillary, which in turn is heated by a steel structure in contact with two cartridge heaters. The capillary will have an outlet pressure of about 4000 psi (27579029 Pa, or 272.2 atmospheres). Documentation provided by the originator:



We note that the environment in which we will be working is extreme in both pressure and temperature, which complicates the problem.

The question is, how long does the channel have to be to achieve a fluid temperature of 850 ºF?

Fluid Properties

We have been given values for the fluid of:

- **Fluid velocity** will be 1 ft/s (.33 gpm; 0.498040376 m/s at the inlet).
- **Average Specific Heat** Btu/lb ºF = 0.5 , or it will range from 1.67 to 2.09 kJ/kg-ºK use 2090 J/kg-ºK
- **Thermal Conductivity** = 0.6069 W/(m ºK) from previous work by others on the same problem (the "Tunisia Study"
- E = **bulk modulus fluid elasticity** (N/m2)~ 1.07 to 1.5 x 10^9 N/m²
- **Density** 977 kg/m³

Due to the fact that we are dealing with a physical regime where the fluid properties will vary over the domain, we will use mass flow rate rather than fluid velocity in our calculations. Given a cross-sectional area of the inlet of .5*.25 inches, or 8.0645E-05 m², we have 0.000040164 m³ per second, or 0.039240683 kg/s mass flow rate for entire channel for the specified density (assumed at normal

conditions) and flow rate. We will be using ¼ of the channel, so our desired mass flow rate will be 0.009810171 kg/s in the ¼ channel.

We expect density to vary due to pressure and temperature. Searching the web, we do not find a source of relative parameters for our fluid domain, so we will have to come up with some sort of approximation.

We also anticipate that the primary effect of the extreme pressure will be on the fluid density. We make the assumption that fluid pressures will be fairly constant (at least relatively) throughout the domain, and evaluate the anticipated effect of pressure on density:

$$\rho_1 = \rho_0 / (1 - (p_1 - p_0) / E)$$

$$\rho_1 = \rho_0(1+\Delta p/E)$$

where

E = *bulk modulus fluid elasticity* (N/m^2)

ρ_1 = *final density* (kg/m^3)

ρ_0 = *initial density* (kg/m^3)

p_1 = *final pressure* (N/m^2) *(27579029 Pa)*

p_0 = *initial pressure* (N/m^2) *(101325 Pa)*

Density at 4000 psi using $E = 1.07\ 10^9$ N/m² > 1002 kg/m³

Density at 4000 psi using $E = 1.5\ x10^9$ N/m² > 994 kg/m³

To validate this, we consider the compressibility of our fluid, which is given by various sources to range from .0045% to .007% per atmosphere. 4000 psi equals 272.18385 atmospheres. so the volume of our fluid should decrease by a factor of 0.012248273 to 0.01905287. This predicts a density due to elevated pressure of 989 to 1016 kg/m³, which is a bit larger range than that arrived at using the bulk modulus, but within the same general magnitude.

We have analyzed the anticipated variation in density over the temperature range and at the pressure specified;

$$\rho_1 = \rho_0 / (1 + \beta (t_1 - t_0))$$

β = volumetric temperature expansion coefficient ($m^3/m^3\ ^oC$) = 0.000045 to 0.00007

For distillate-type fluids, standard properties are generally given at 40 ºC. and we find that the variation in density for our model fluid to closely follow the relationship:

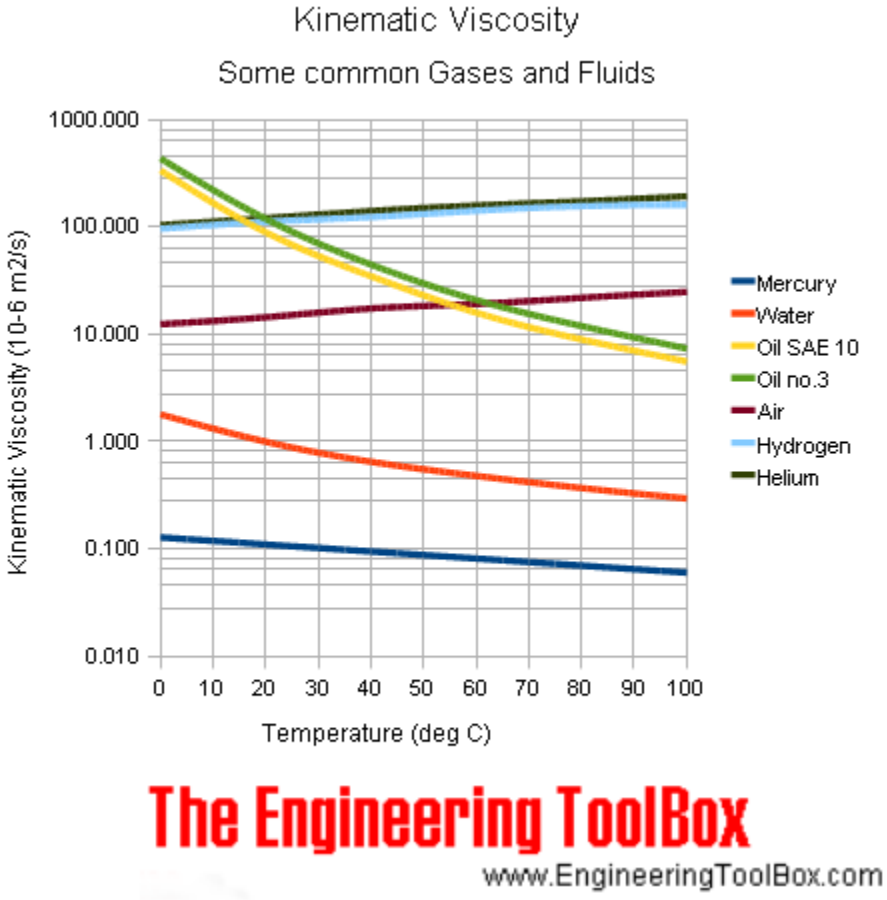
$$\rho = 992.8 - 0.068*T$$

Adjusting the slope for proper units, we find that the Coefficient of Thermal Expansion for our fluid is expected to be approximately 0.0000685 /ºC.

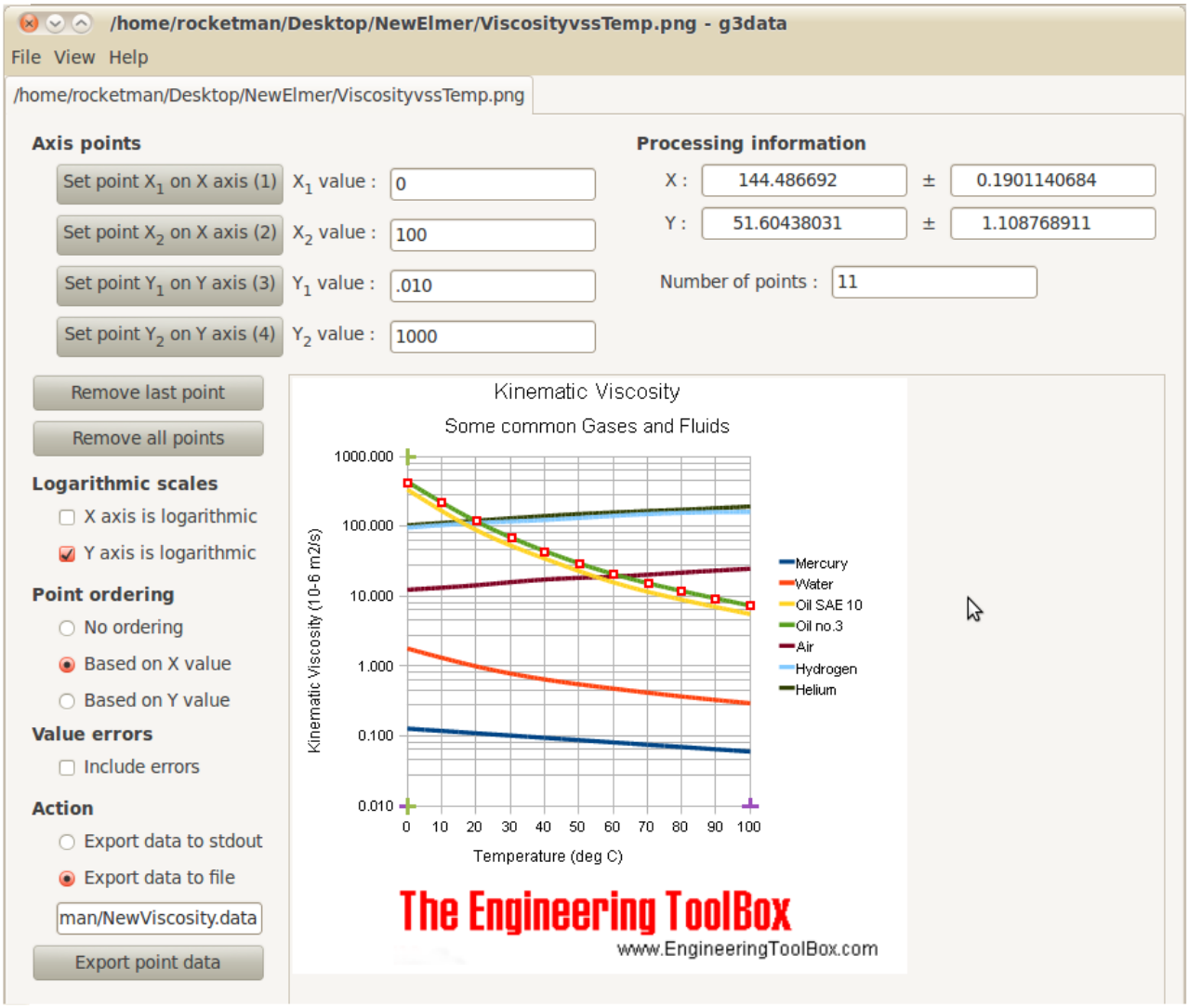
At the specified pressure and over the temperature range of interest, we expect the density to vary from 986 to 960 kg/m³ in a linear manor. Using the compressibility model, we expect a range from 1013 to 986 kg/m³ (also linear). We will want to run multiple studies to evaluate the sensitivity to density variation of the heat transfer to the fluid.

Dynamic Viscosity

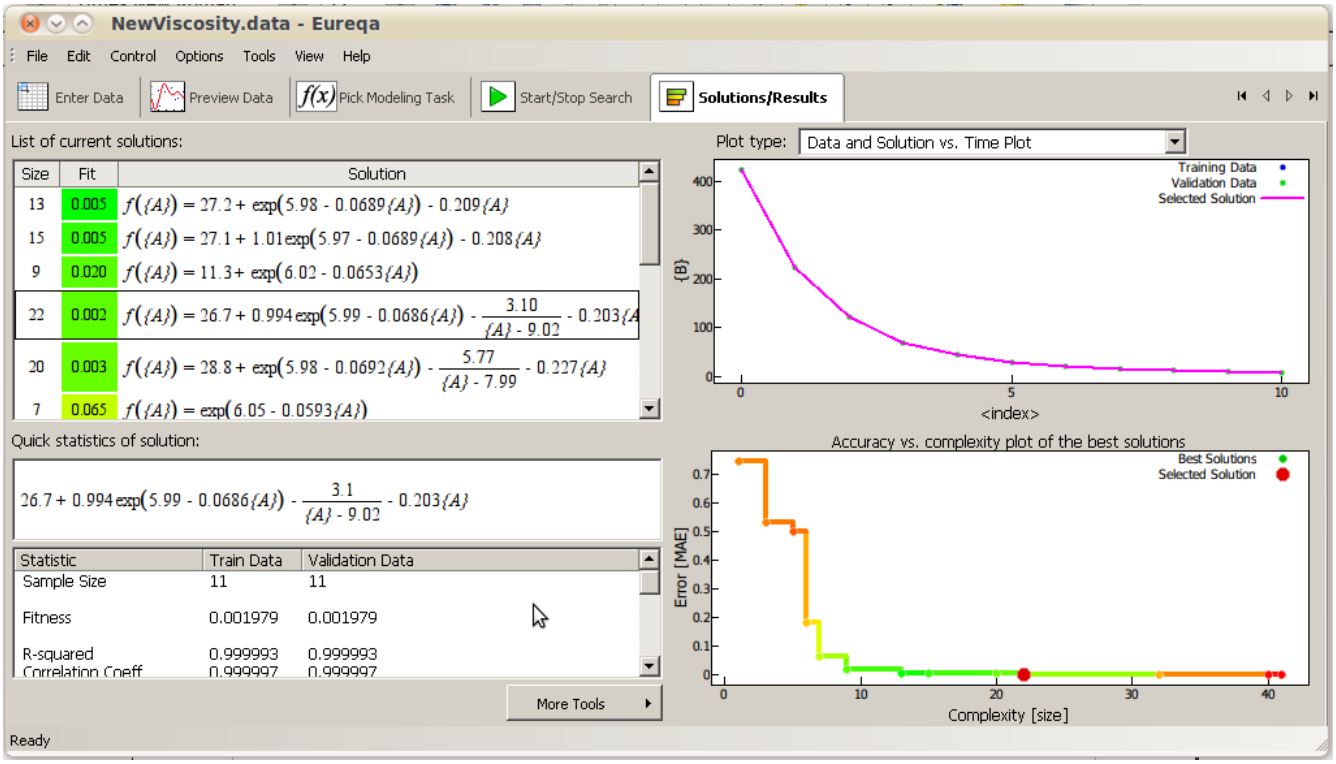
Determining the dynamic viscosity of the fluid has been a bit more involved. The best data we could come up with was a graph of kinematic viscosity for similar fluids from [The Engineering ToolBox]:



To convert this to useful information, and specifically to extend it to the temperature range of interest, we incorporated a couple of Open Source tools. First, we use a program called program called [g3data] to extract the data from the above chart:



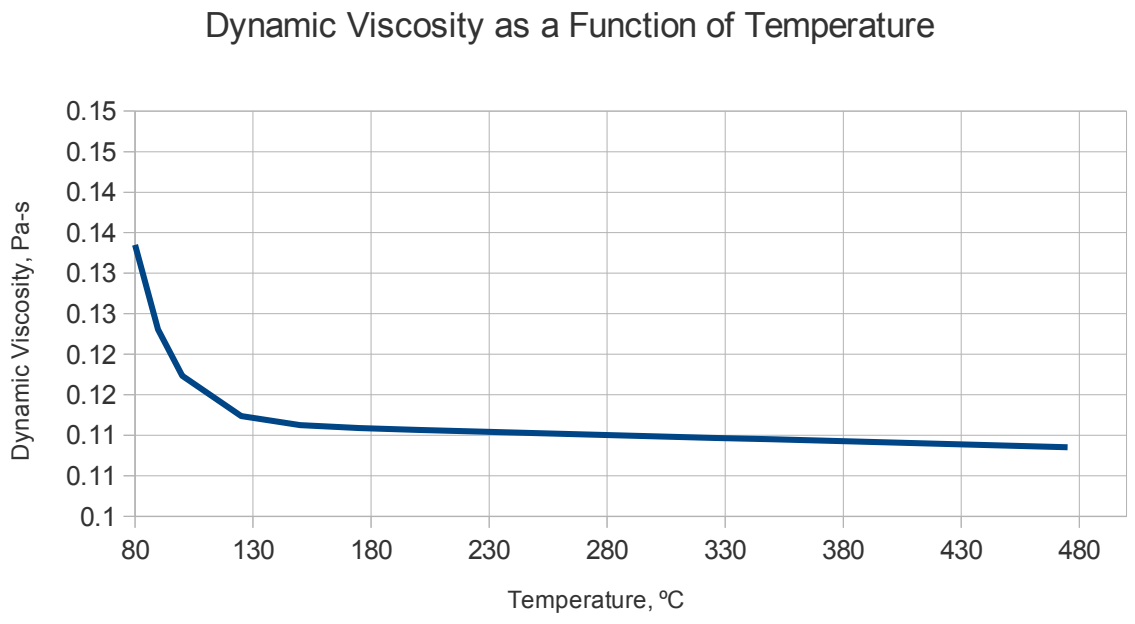
We then export this data to a program called Eureqa to convert the data to a mathematical relationship:



After adjusting the resulting relationship to convert to dynamic viscosity (unit adjustment and multiplying the kinematic viscosity by the density function derived above), we come up with the following approximation:

$$\mu(T) = 10^{-6} * (11.3 + \exp(6.02 - .0653 * T)) * (992.8 - 0.068 * T)$$

The following figure illustrates the viscosity function defined above:



This gives us a higher variation at lower temperatures, leveling off to a nearly constant value above about 150 ºC. Again, we may want to run multiple studies to evaluate the effect of this variable.

Solid Properties

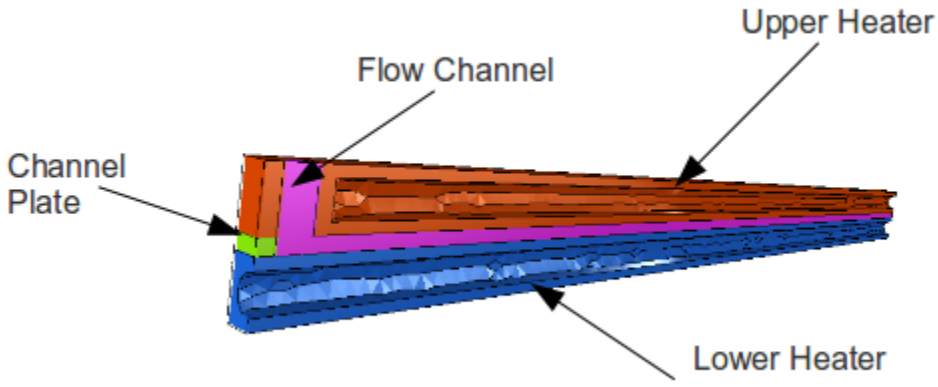
The main solid body actually consists of three plates in intimate contact with each other. The material for the plates will be Bohler-Uddeholm SUPERIOR® H13 Hot Work Tool Steel:

- **Density:** 7.61 g/cc @ Temperature 593 °C (7610 kg/m³)
- **Heat Capacity:** 0.590 J/g-°C @ Temperature 593 °C (590 J/kg-ºC)
- **Thermal Conductivity:** 33.4 W/m-K @ Temperature 704 °C
- **Modulus of Elasticity:** 179 Gpa @499 ºC
- **Yield Strength:** 331 Mpa @ 649 ºC

Actual Operating temperature will be on the order of 677 ºC. Outer surfaces will be insulated with Pyrogel XT, with a Thermal conductivity of 0.09 W/m-K. The heater cartridges will have a constant temperature of 1250 ºF (676.66667 ºC).

The Problem Domain

The full domain will essentially consist of a solid and a fluid domain, with the geometry shown here:

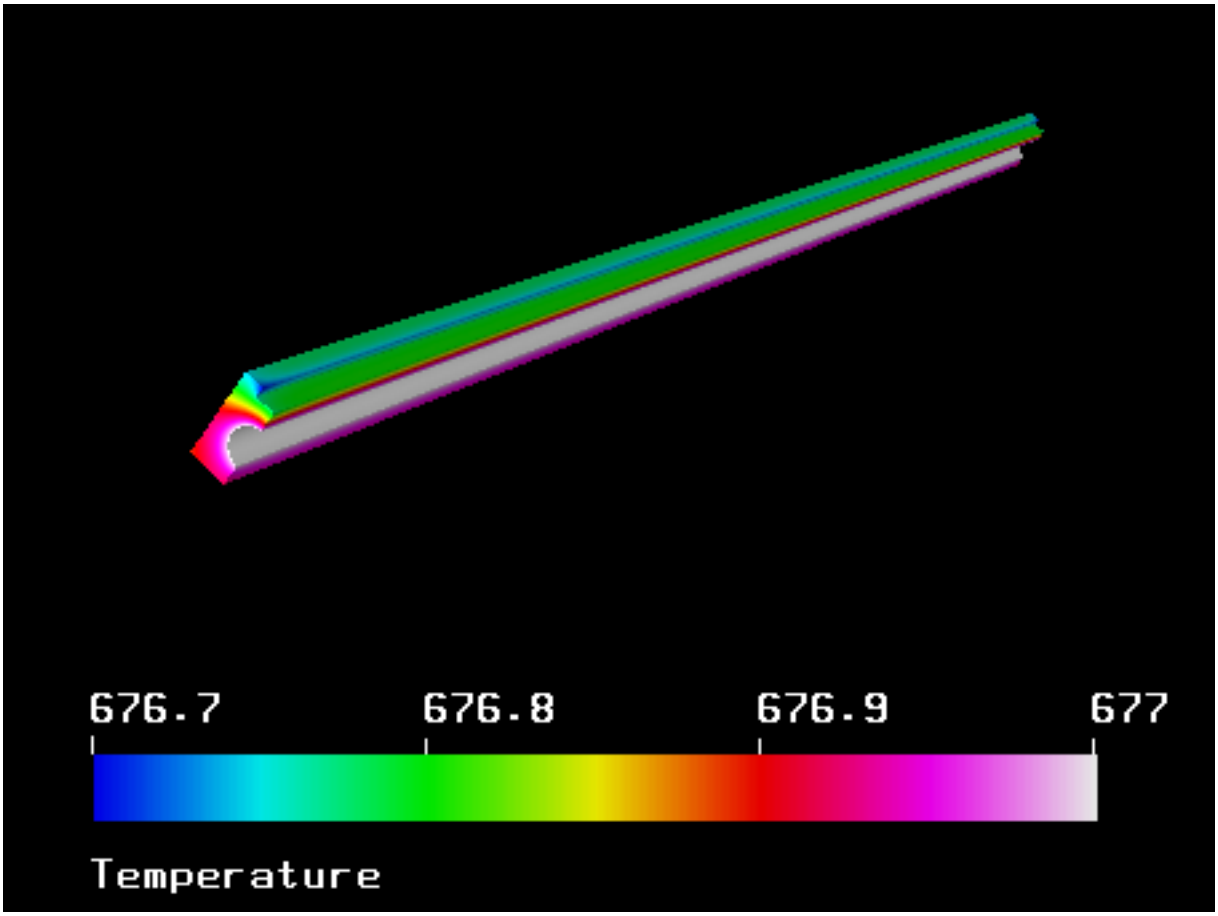


Note that the fluid channel begins with a circular inlet that transitions through a 90º turn into a rectangular channel. We have used the Salome package to create the geometry.

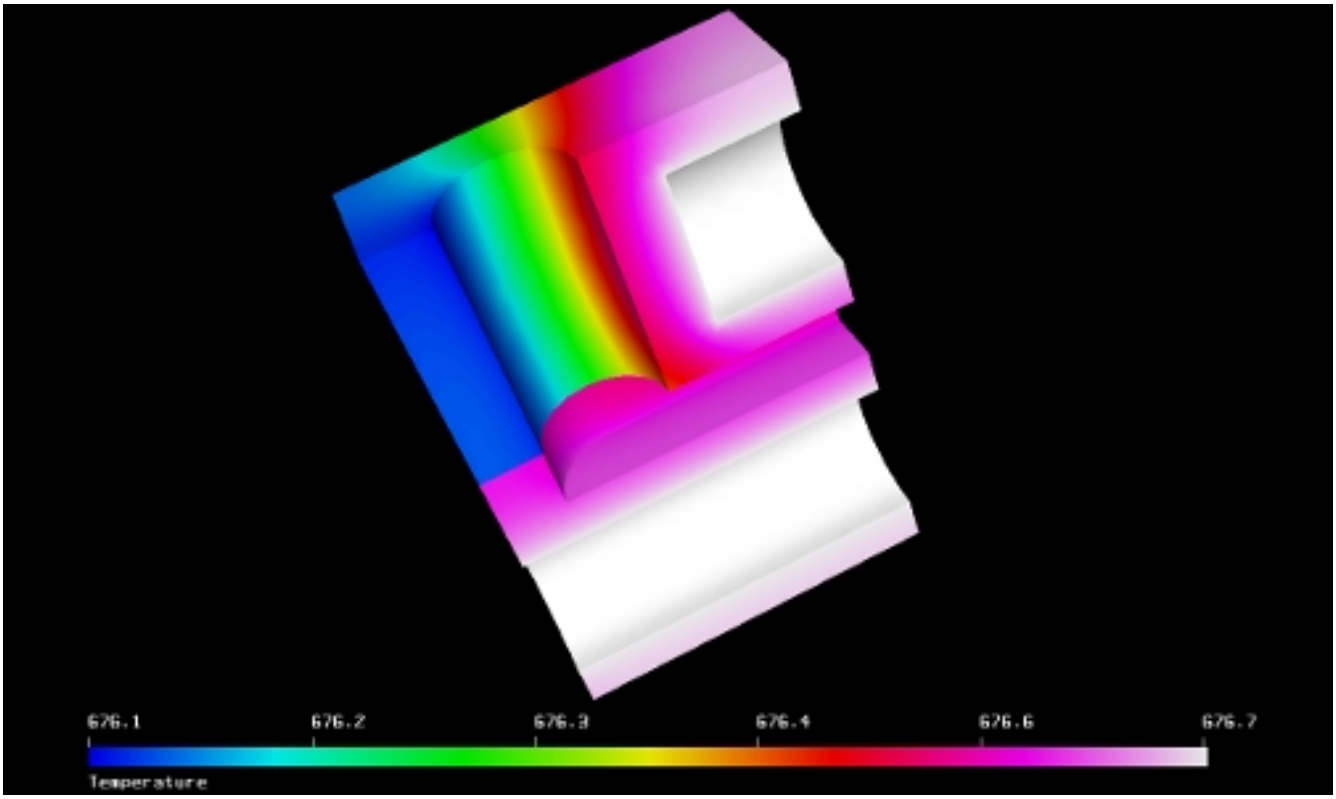
Preliminary Thermal Analysis

We will begin our solution by analyzing the steady state thermal characteristics of a section of the solid domain in order to evaluate the anticipated thermal environment expected for the fluid. We use the Open Source CFD package Elmer, distributed by CSC — IT Center for Science Ltd (administered by the Ministry of Education and Culture of Finland, and first developed in collaboration with Finnish Universities, research institutes and industry). Note that this package is included in the Open Source Computer Aided Engineering package, CAELinux 2010 distribution (an extensive collection of Open Source engineering programs, developed by an independent group of developers without external financial support, including sophisticated FEA, CFD and other multi-physics solvers). A part of this investigation includes the goal of comparing the functionality and capabilities of some of the various packages, and their applicability to particular investigations. We selected Elmer for the initial thermal

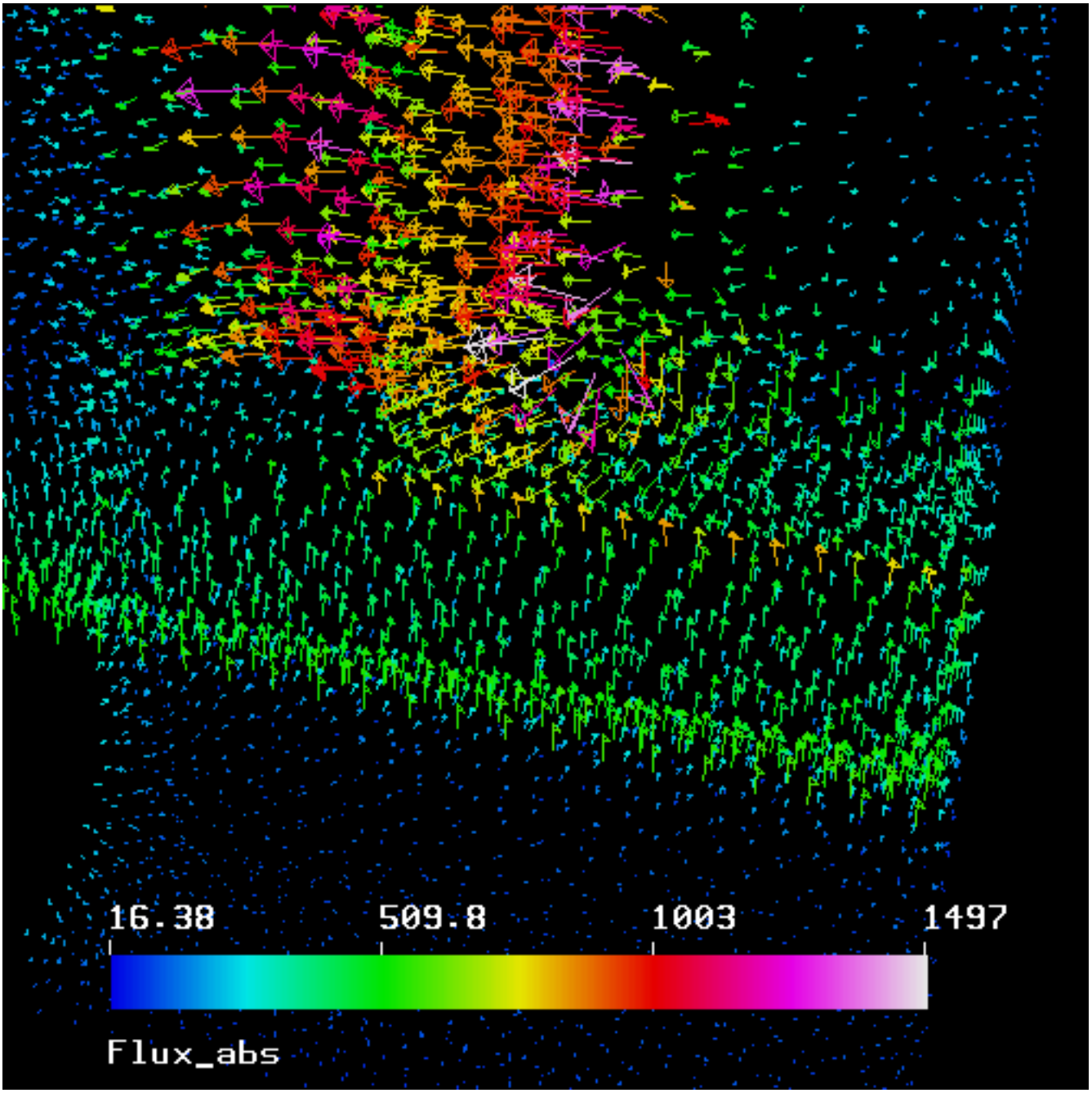
modeling because it is relatively easy, with its included GUI, to set up a study case, and integrates a (limited) post-processor that can give us preliminary results quite quickly on our laptop with a dual-core, 64-bit processor and 3GB of memory. While the package allows for exporting the results to other formats such as *.vtk for use with the ubiquitous Paraview package (also included in CAELinux), this is an additional step that it would be nice to avoid, as will be discussed below.



This first image shows ¼ of the heater assembly, taking advantage of the symmetry of the structure. The radiused region is where the actual heater cartridge is installed, and the surface temperature in was set constant at the cartridge heater temperature (which ignores any heat transfer inefficiencies in the contact between the heater and the steel plate). The step on top represents the fluid channel, while the back and the bottom are the insulated surfaces. Information gleaned from this exercise is that the fluid channel wall temperatures are fairly uniform; that is, we should be able to come up with a reasonable approximation of the heat transfer to the fluid by using a simplified model of fixed wall temperatures. The temperature for the bottom of the fluid channel will be taken as 676.8 ºC and the sidewall will taken as 676.6 ºC.



This next image shows the inlet and heater end. Again, we note that the variation of wall temperatures within the channel quite small.



This image gives us an indication of the heat flux we can expect into the channel, based on the fluid properties and temperature of the heater assembly. Estimating the color of the flux vectors to be in the region of 510 W/m², and given our estimated temperature difference of (676.8-80), we determine a flux of approximately 0.85 W/m²-ºK.

It should be noted that to get the flux image from the Elmer package, it was necessary to do a bit of editing of the case file. Many of the "solvers" that come with the Elmer package are accessible and editable from the GUI, but, in this case, it was necessary to hand edit the case file. Fortunately, Elmer comes with extensive documentation of the various solvers, and stores the case file in a plain text *.sif file. This is the case file from this particular simulation:

```
Header
  CHECK KEYWORDS Warn
```

```
  Mesh DB "." "."
  Include Path ""
  Results Directory "."
End

Simulation
  Max Output Level = 4
  Coordinate System = Cartesian
  Coordinate Mapping(3) = 1 2 3
  Simulation Type = Steady state
  Steady State Max Iterations = 1
  Output Intervals = 1
  Timestepping Method = BDF
  BDF Order = 1
  Solver Input File = case.sif
  Post File = case.ep
End

Constants
  Gravity(4) = 0 -1 0 9.82
  Stefan Boltzmann = 5.67e-08
  Permittivity of Vacuum = 8.8542e-12
  Boltzmann Constant = 1.3807e-23
  Unit Charge = 1.602e-19
End

Body 1
  Target Bodies(1) = 8
  Name = "Body 1"
  Equation = 1
  Material = 1
End

Solver 1
  Equation = Heat Equation
  Variable = -dofs 1 Temperature
  Procedure = "HeatSolve" "HeatSolver"
  Exec Solver = Always
  Stabilize = True
  Bubbles = False
  Lumped Mass Matrix = False
  Optimize Bandwidth = True
  Steady State Convergence Tolerance = 1.0e-5
  Nonlinear System Convergence Tolerance = 1.0e-8
  Nonlinear System Max Iterations = 20
  Nonlinear System Newton After Iterations = 3
  Nonlinear System Newton After Tolerance = 1.0e-3
  Nonlinear System Relaxation Factor = 1
  Linear System Solver = Iterative
  Linear System Iterative Method = BiCGStab
  Linear System Max Iterations = 500
```

```
  Linear System Convergence Tolerance = 1.0e-8
  Linear System Preconditioning = ILU0
  Linear System ILUT Tolerance = 1.0e-3
  Linear System Abort Not Converged = False
  Linear System Residual Output = 1
  Linear System Precondition Recompute = 1
End

Solver 2
  Equation = Flux Solver
  Procedure = "FluxSolver" "FluxSolver"
  Exec Solver = Always
  Linear System Solver = Iterative
  Linear System Iterative Method = BiCGStab
  Linear System Max Iterations = 500
  Linear System Convergence Tolerance = 1.0e-8
  Linear System Preconditioning = None
End

Equation 1
  Name = "Equation 1"
  Active Solvers(2) = 1 2
End

Material 1
  Name = "Austenitic stainless steel (AK Steel 201)"
  Poisson ratio = 0.3
  Heat expansion Coefficient = 15.7e-6
  Youngs modulus = 197.0e9
  Heat Conductivity = 33.4
  Heat Capacity = 590.0
  Mesh Poisson ratio = 0.3
  Density = 7610.0
  Poisson ratio = 0.3
  Youngs modulus = 197.0e9
End

Boundary Condition 1
  Target Boundaries(1) = 1
  Name = "Insulated Surface"
  External Temperature = 20
  Heat Transfer Coefficient = .09
End

Boundary Condition 2
  Target Boundaries(1) = 2
  Name = "Fluid_Interface"
  External Temperature = 79.4444
  Heat Transfer Coefficient = .6069
End
```

```
Boundary Condition 3
  Target Boundaries(1) = 3
  Name = "Symmetry"
  Heat Flux = 0.
End

Boundary Condition 4
  Target Boundaries(1) = 6
  Name = "HeaterSurface"
  Temperature = 676.66667
End

Boundary Condition 5
  Target Boundaries(1) = 7
  Name = "HeaterSurface"
  Temperature = 676.66667
End

Boundary Condition 6
  Target Boundaries(1) = 8
  Name = "Symmetry"
  Heat Flux = 0.
End
```

The portions of the file that had to be edited to provide the flux output are highlighted. The rest of the file was built automatically from the GUI.

The Fluid Domain

We are now going to switch to a different analysis program, although Elmer has the capabilities to complete the entire project, including combined heat and fluid flow. The reason we are switching is because we want to explore Code_Saturne, developed by EDF of France, which is integrated with the Salome-MECA platform (also from EDF) in the CAELinux distribution. We also note that the post processing facilities integrated with the Elmer package (of which there are two) appear to be somewhat limited and difficult to manipulate, especially for the geometry with which we are working. We happen to have a personal preference for the post-processor packaged with Salome. In our opinion, an integrated package is preferable, since there is no issue with file formats when moving into a new phase of the project. As noted, we are also interested in exploring the capabilities of the different packages.

But first, a little more manual calculation. We need to determine the Reynolds Number to establish the flow regime, and the Reynolds Number requires a calculation of the hydraulic diameter (which we also happen to need for our simulation):

For flow in a pipe or tube, the Reynolds number is generally defined as:

$$\mathrm{Re} = \frac{\rho \mathrm{V} D_H}{\mu} = \frac{\mathrm{V} D_H}{\nu} = \frac{\mathrm{Q} D_H}{\nu A}$$

where:

- D_H is the hydraulic diameter of the pipe
- Q is the volumetric flow rate
- A is the pipe *cross-sectional* area

For shapes such as squares, rectangular or annular ducts (where the height and width are comparable) the hydraulic diameter is given by:

$$D_H = \frac{4A}{P}.$$

Where A is the cross-sectional area and P is the wetted perimeter.

From the above, we determine that our Reynolds Number is works out to be on the order of 37. For low Reynolds number, Re $\ll 1$ the fluid motion is regular and laminar. For intermediate Reynolds number Re of order ~ 1 to $\sim 10^2$ complicated flows are observed with, depending on the precise set-up, some of the symmetries permitted by the equations of motion and the boundary conditions broken.

Our hydraulic diameter is calculated to be .00847 m.

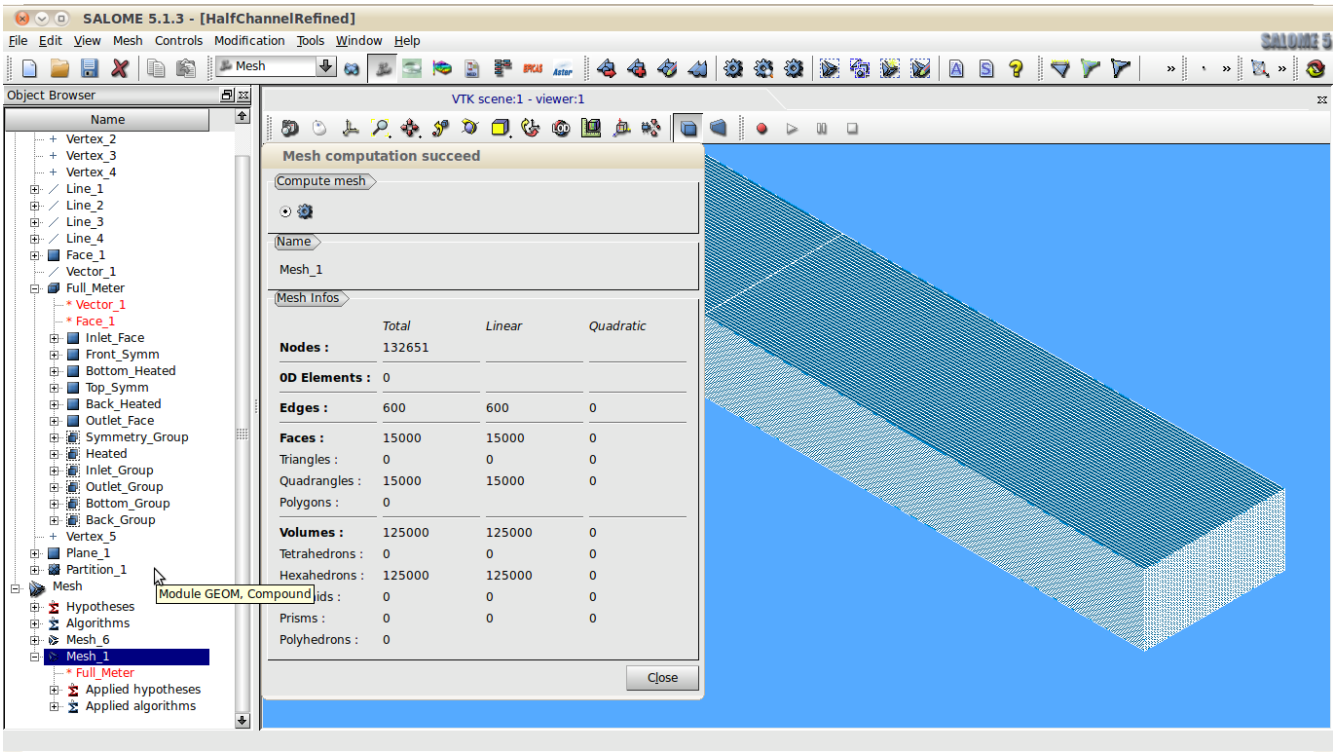
To review the additional fluid properties:

Fluid Properties:

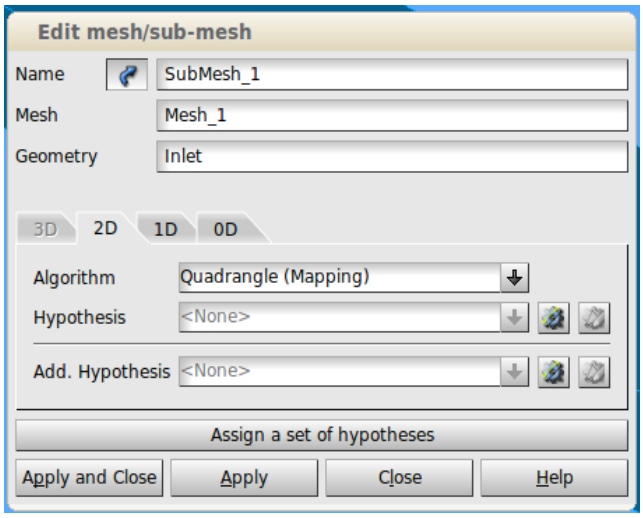
- **Mass Flow Rate**: 0.009810171 kg/s
- **Average Specific Heat:** Btu/lb ºF = 0.5 , or 1.67 to 2.09 kJ/kg-ºK (use 2090 J/kg-ºK)
- **Thermal Conductivity:** 0.6069 W/(m ºK)
- **Bulk Modulus Fluid Elasticity:** ~ 1.07 to 1.5 x 10^9 N/m²
- **Density:** (992.8 -0.068*T) kg/m³
- **Viscosity:** 10^{-6} *(11.3 + exp (6.02 - .0653 * T)) * (992.8 -0.068*T)

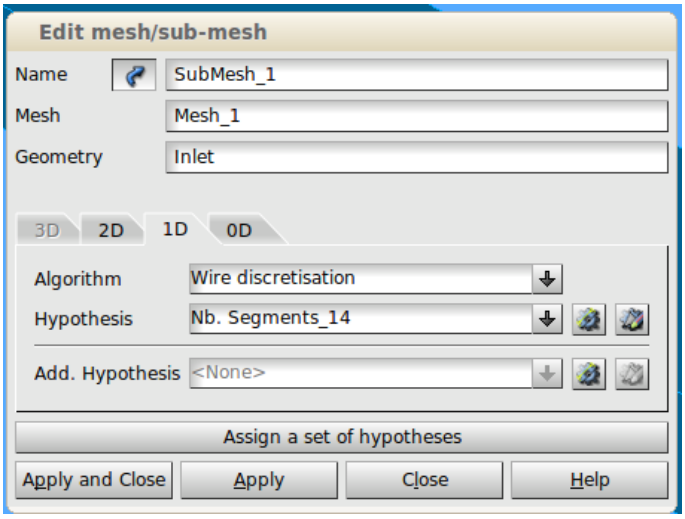
Experimenting with a variety of mesh configurations, and considering that we are anticipating laminar flow in a rectangular channel, we have, with experimentation, developed what appears to be a reasonable mesh for the fluid study, consisting of 250,000 hexahedral elements (25 x 25 elements in cross section by 400 elements along the length). Preliminary test runs of coarser meshes did not appear to give adequate resolution. The model will be a meter long channel representing ¼ of the full fluid domain, taking advantage of two symmetry planes. Final dimensions are 1 meter by 0.00635 meter by 0.003175 meter.

Actually, to achieve this mesh with the limitations of our computer, there are a couple of tricks that are not well-documented. The first step is to create a course mesh, using thee “Automatic Hexahedralization” option in the “Create Mesh” menu, setting the “Nb. of Segments” to 50, and compute the mesh:

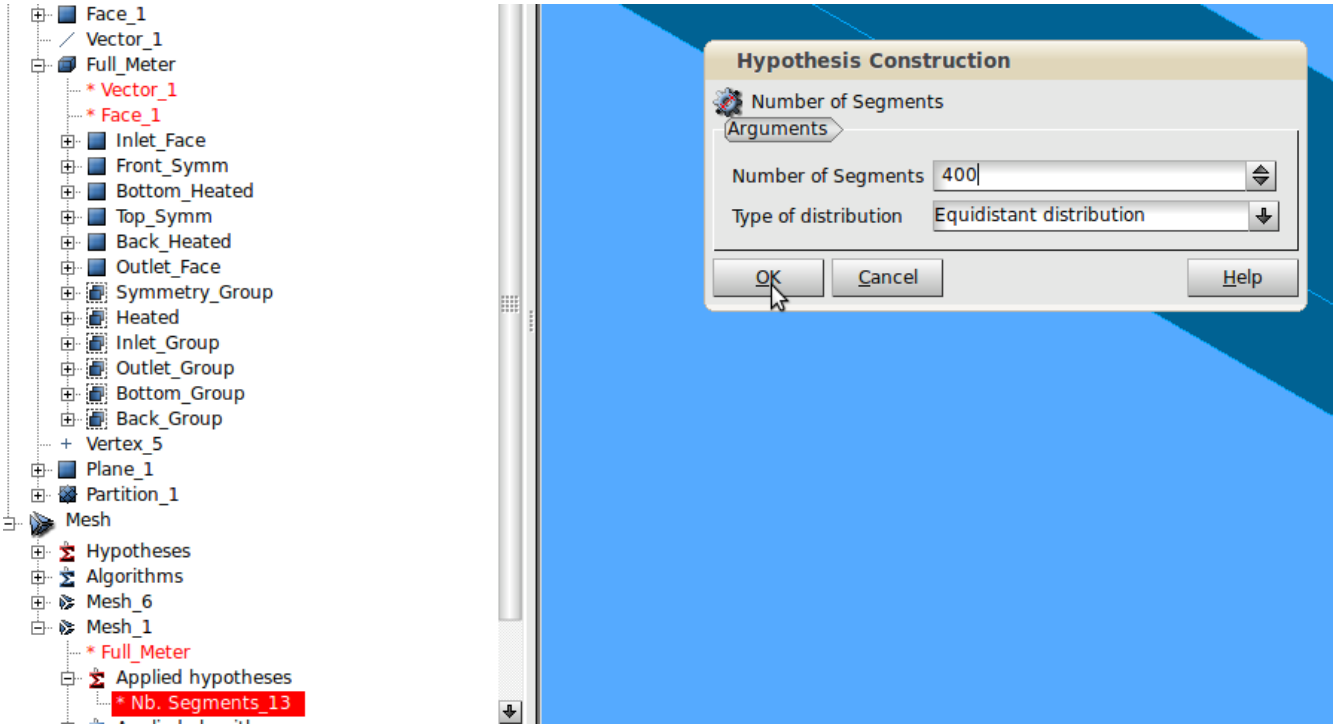


Next, we create identical sub-meshes on the inlet and outlet faces:

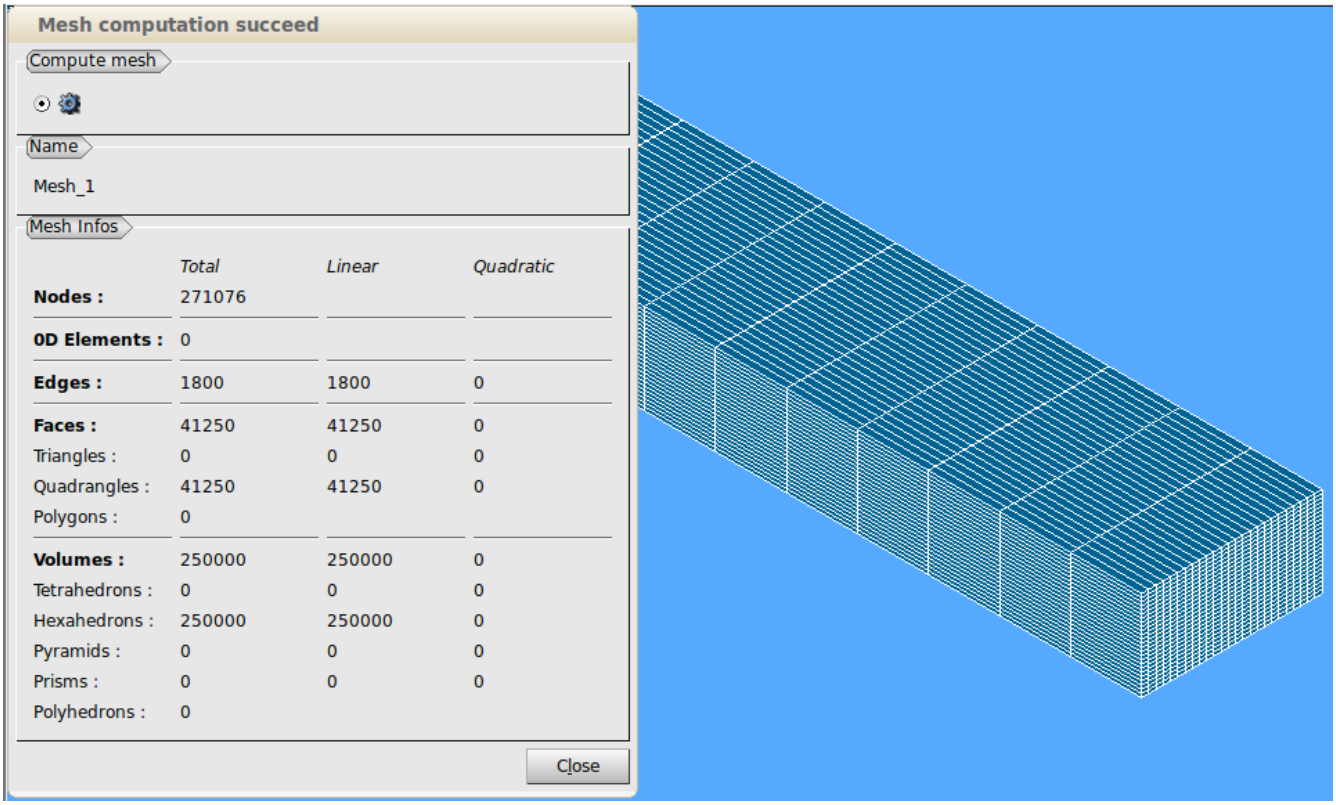




After recomputing them mesh with the sub-meshes included, we edit the main mesh hypothesis “Nb Segments”, changing the original 50 to 400:



After which we recompute the mesh, winding up with the following:



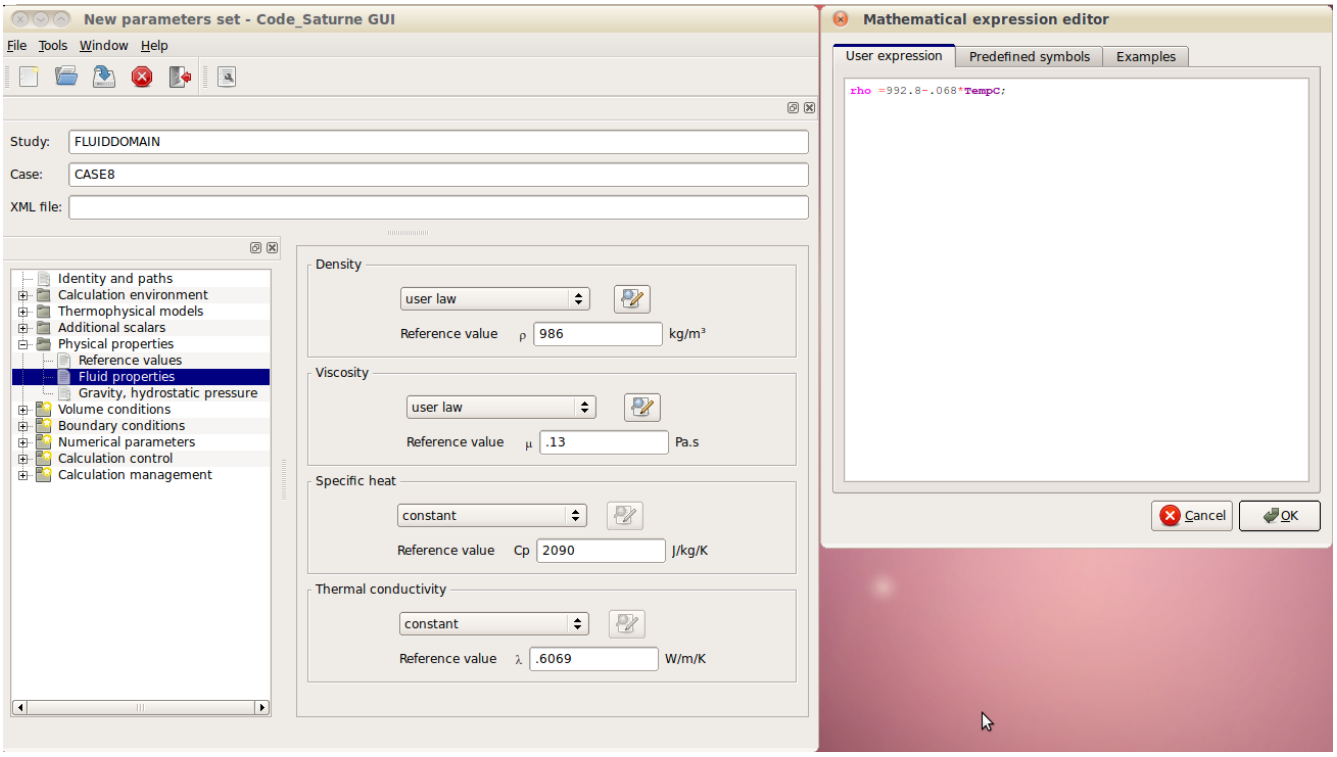
Before exporting the mesh in the *.med format which we will be using for our CFD analysis, it is necessary to define the boundary groups (groups on faces), a procedure well documented in the literature.

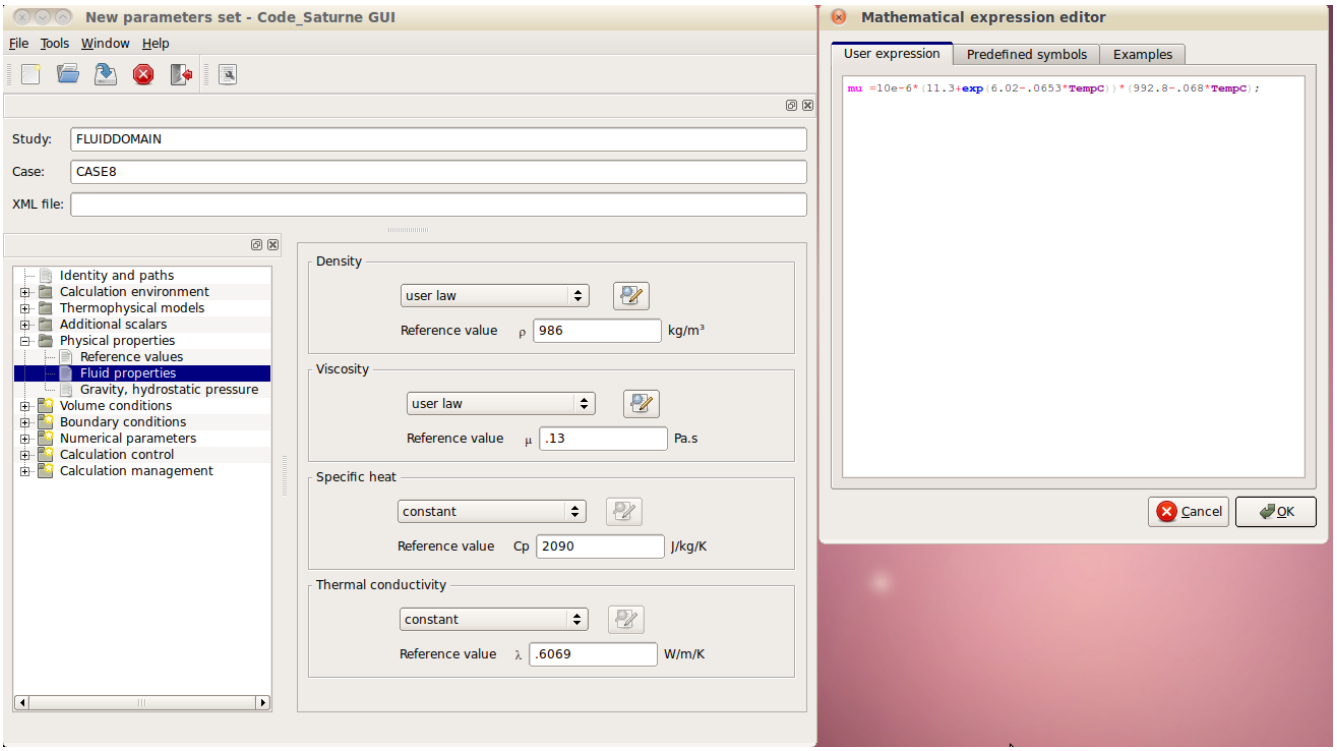
An analysis of the definition of the formula for the Reynolds Number suggests that Re will decrease with temperature. We also suspect that, since the cooler fluid at the inlet will have a higher density than the heated fluid, we should be able to improve thermal mixing by injecting the fluid at the top of the channel oriented with the long dimension vertical (this was substantiated with preliminary test runs of the model, and through comparison with injecting the fluid at the bottom of the channel).

We experimented with different turbulence models, especially a low-Reynolds model known as "v^2-f" in Code_Saturne, which is similar to the standard k-epsilon model, but incorporating also some near-wall turbulence anisotropy as well as non-local pressure-strain effects. It is a general turbulence model for low Reynolds numbers that does use wall functions because it is valid up to solid walls. It can provide the right scaling for the representation of the damping of turbulent transport close to the wall. Further research uncovered a study by Juan Uribe (available at http://cfd.mace.manchester.ac.uk/) that the Shear Stress Transport (SST) model may be more appropriate for our study. The SST model is in reality a combination of two eddy viscosity models, based on purely empirical blending functions. The great advantage of the SST model is that it allows for a direct integration over the viscus sub-layer without the need of damping functions. It is important to note that, although the model can be used in the viscous sublayer, it has not been designed to take into account any of the wall effects. The SST model is good in attached boundary layers. Several investigators have demonstrated that the SST model predicts recirculation better than traditional models. For the long tall cavity the SST model does a good job in predicting values for the vertical velocity and the temperature. In this case it is interesting to note that the model acts in κ-ω mode in most of the domain, so the predictions are very similar to the original model. We also experimented with other turbulence models, without significant differences.

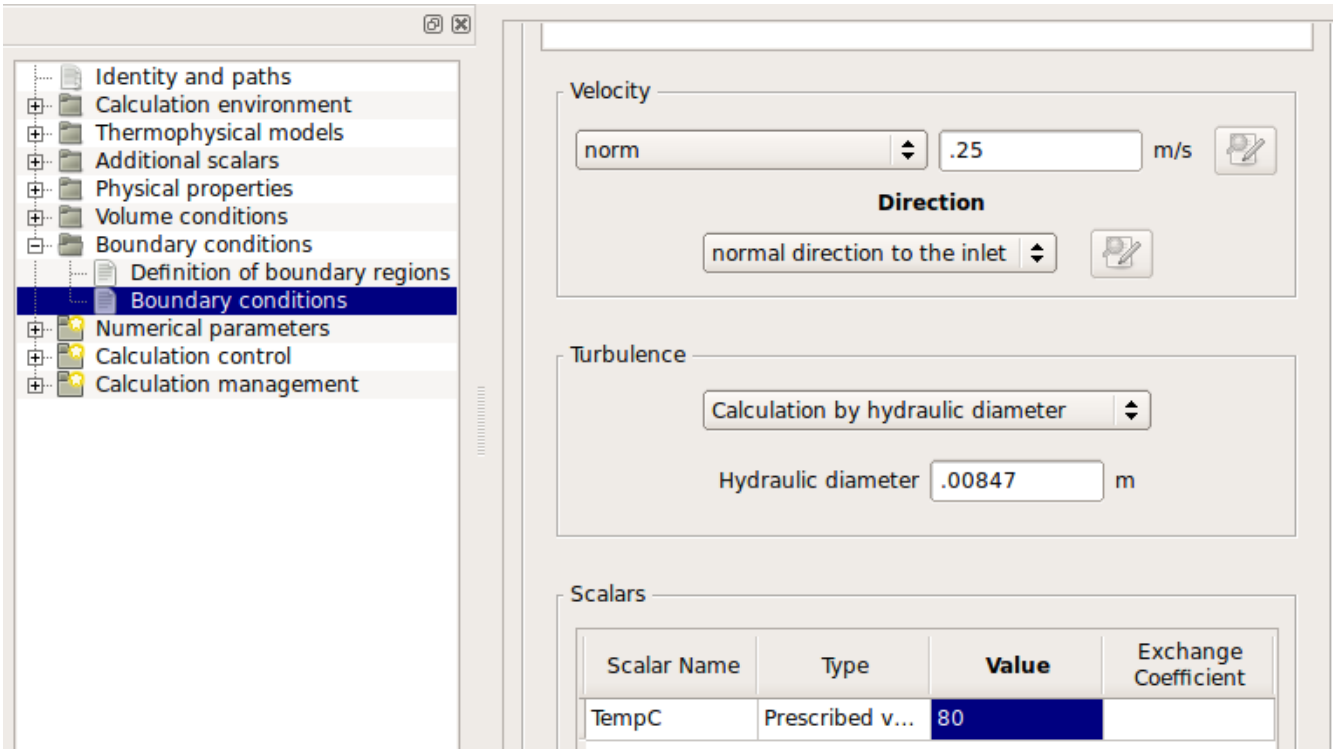
Preliminary runs also demonstrated that the system achieves steady state in somewhat less than 200 time steps (using the default 1 second time step), but to be safe, as we modified various parameters and experimented with alternative turbulence models and physical orientations, we settled on letting the process run for 300 time steps (note that a typical run requires about 1.8 hours on our system).

Some critical parameter settings used in Code_Saturne include the variable density and variable viscosity, set by using "User Law" for the appropriate parameter, thus:



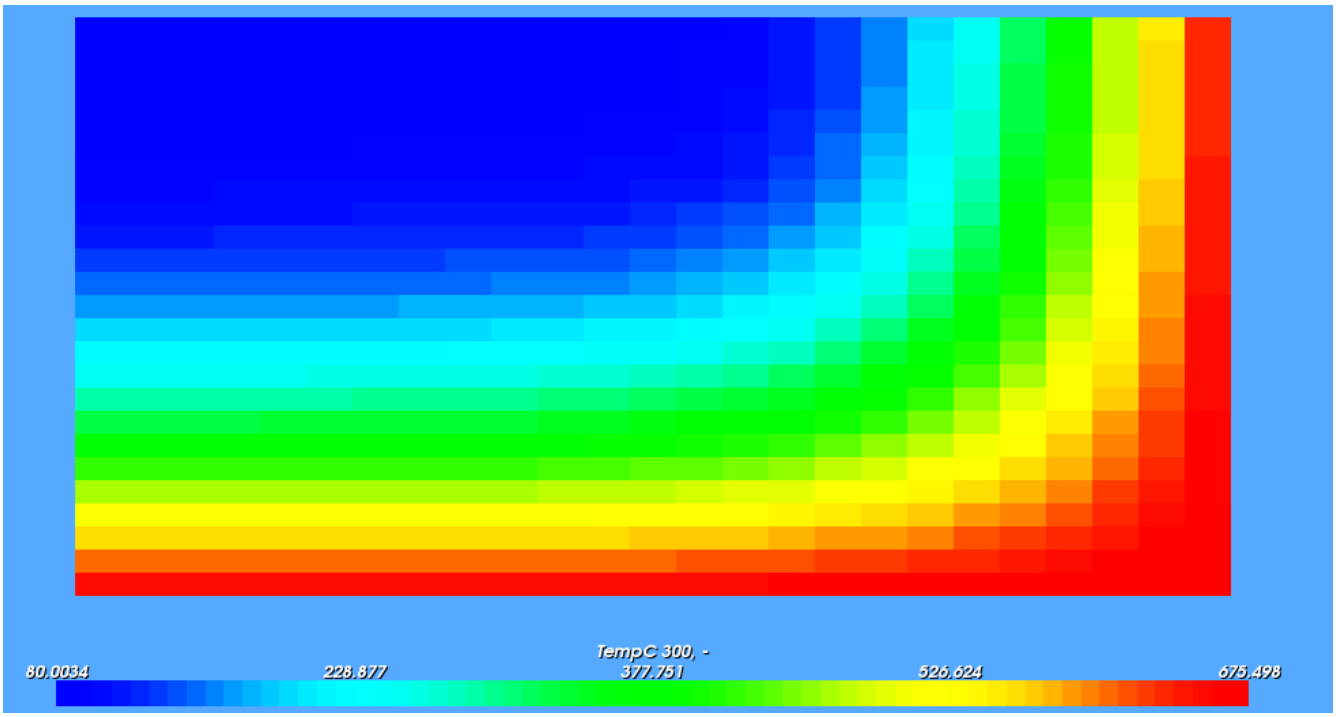


And, finally, primarily to provide a reference point, we look at the input boundary conditions:

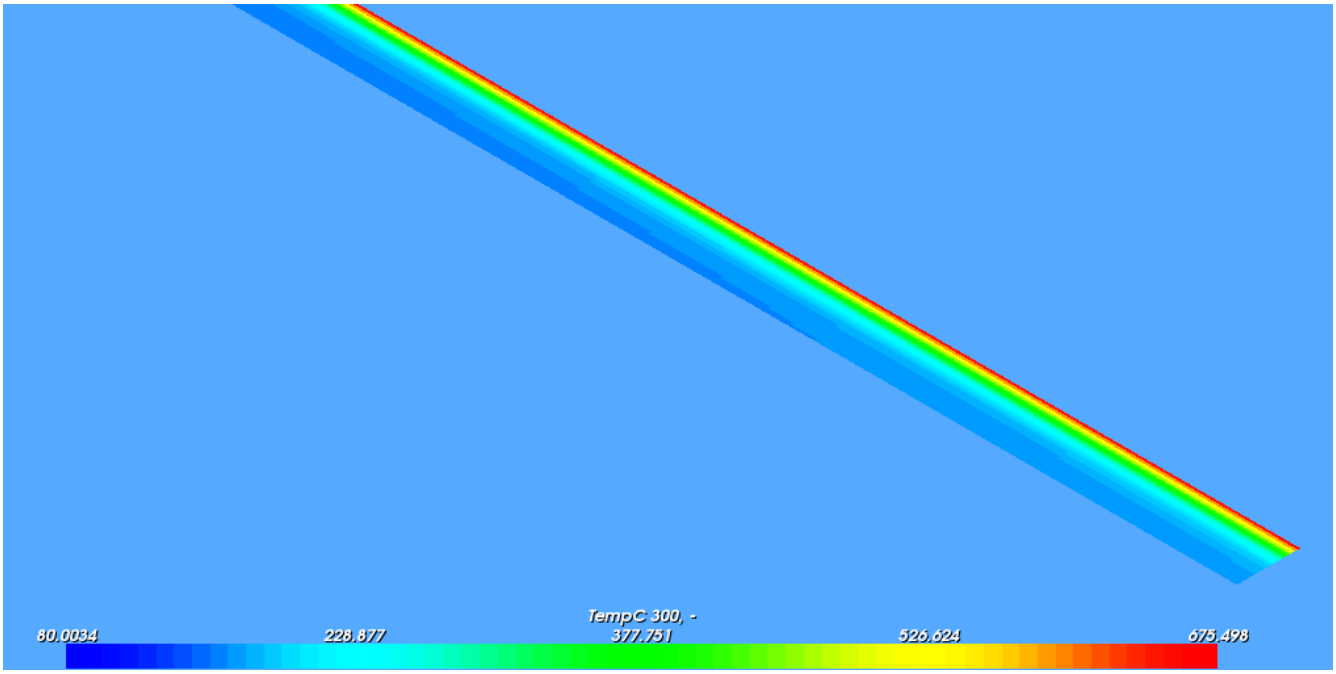


Results

Looking at the results of our final run in the Salome post-processor (this is the outlet face):



We see that we are not getting very good thermal mixing. Another view of a cut plane through the middle of the channel shows that low heat transfer through the fluid is likely the limiting factor:



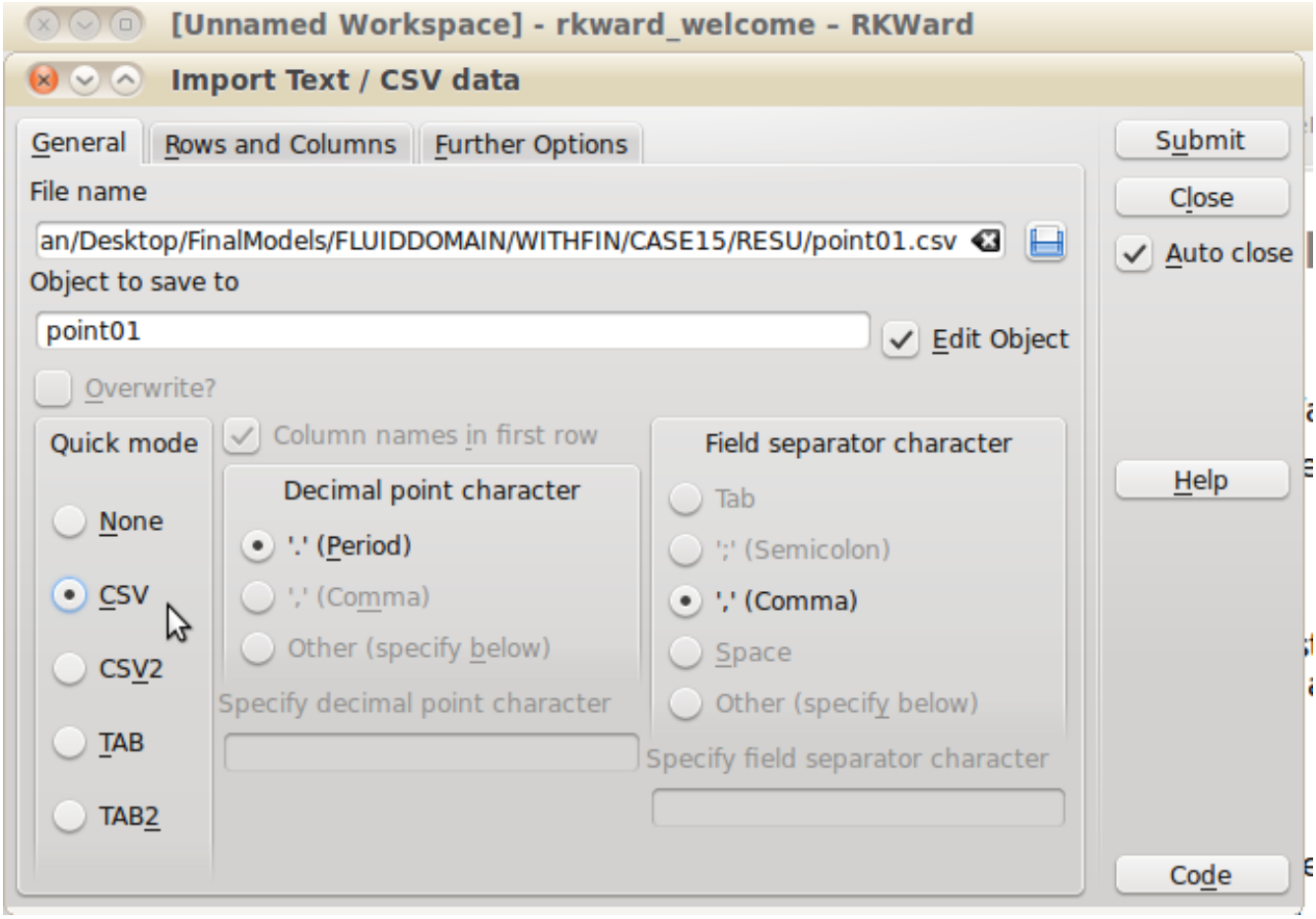
We note that the boundary layer heats up very rapidly near the inlet (not shown), but beyond the boundary layer, the thermal transfer is fairly slow.

What we really want to know is the average fluid temperature as a function of distance along the channel. At this point, we switch to the Paraview visualization package because it offers a function "Integrate Variables" that we can use to estimate average temperature along the channel. This is accomplished By creating slices at various distances along the channel, selecting the "Integrate Variables" from the "Filters" menu, then exporting the data for further analysis in our spreadsheet. The data exported from Paraview gives us 1250 temperature values at each cut plane, and these are evaluated using the relationship,

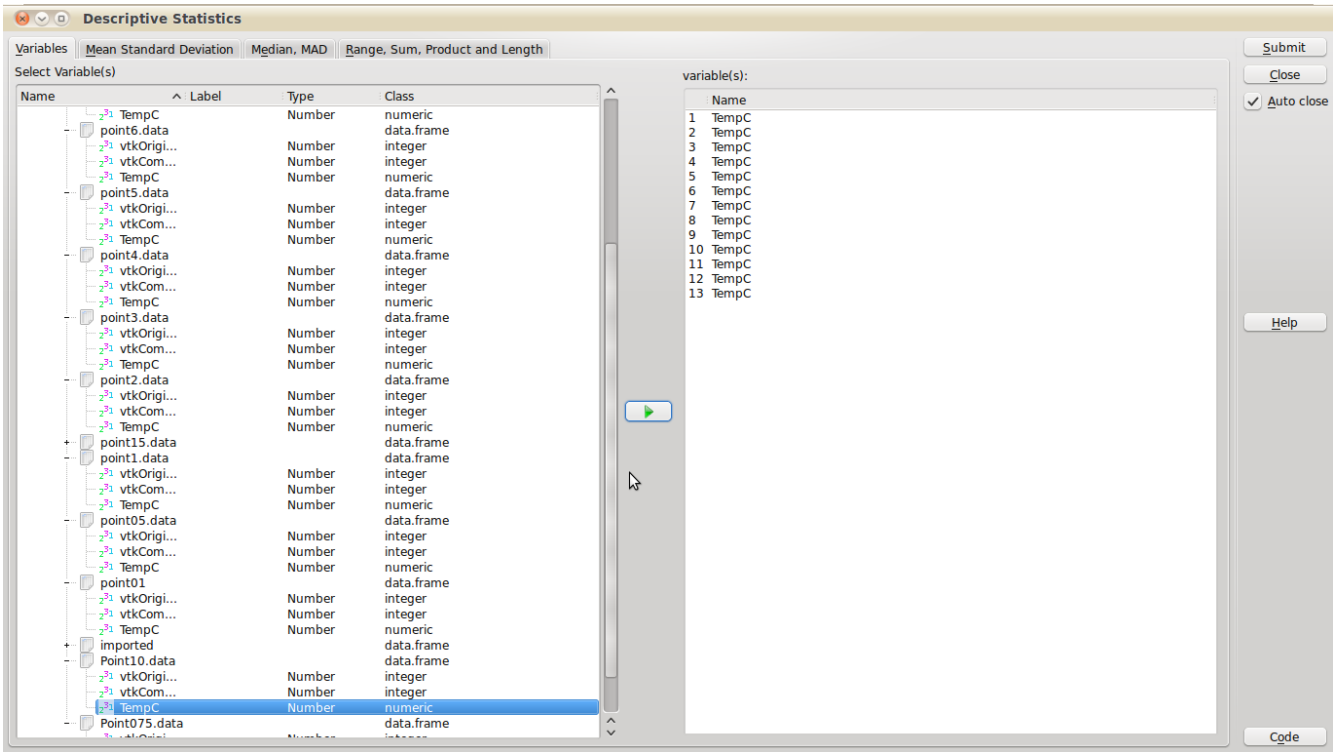
$$T_{avg} = (1/A) * \Sigma (T * dA)$$

Where A is the total cross sectional area, and dA is 1/1250 * A. Rather than use the traditional spreadsheet, however, since all of our area increments are the same size, we can use R (a statistical analysis package) with the RKWard GUI interface (both of which are included in the CAELinux package) to more conveniently calculate the arithmetic mean.

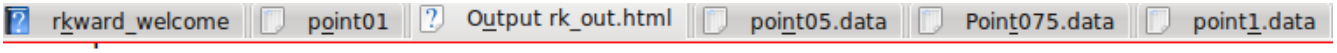
We begin by importing our data (which was exported from Paraview as *.csv files) into RKWard using the "Files/Import/Import format/Import Text 7 CSV data" menu selection:



Next, we select the variables we want to analyze (in this case, the temperature) by selecting the menu "Analysis/Descriptive Statistics" and select the appropriate variable from each file to be analyzed:



Note that the program doesn't seem to mind that all of the variables have the same name, since they come from different files. Selecting the "Submit" button in the upper right corner of the illustrated window results in:



Run again

Descriptive statistics

Parameters

- Trim of mean: 0

Sat Apr 23 03:12:27 2011

Object	mean	min	max	standard deviation	length of sample	number of NAs
TempC	148.9577	80.0134	650.809	132.8251	1250	0
TempC	195.9088	80.0156	667.018	174.7423	1250	0
TempC	211.8716	80.0166	669.079	182.9757	1250	0
TempC	224.4431	80.0177	670.262	188.2899	1250	0
TempC	259.5482	80.0227	672.417	198.5205	1250	0
TempC	283.6417	80.0299	673.334	202.1901	1250	0
TempC	302.4918	80.0426	673.867	203.3749	1250	0
TempC	318.1711	80.0709	674.225	203.2942	1250	0
TempC	331.6948	80.1408	674.486	202.4681	1250	0
TempC	343.9175	80.3048	674.69	201.1282	1250	0
TempC	354.4119	80.6169	674.845	199.5192	1250	0
TempC	364.3644	81.1701	674.979	197.6265	1250	0
TempC	373.3966	82.1483	675.101	195.5822	1250	0

Run again

We copy and paste the "mean" entry for each point into a new dataset sheet, and manually enter the value for "Position":

point01 | Output rk_out.html | Output.data

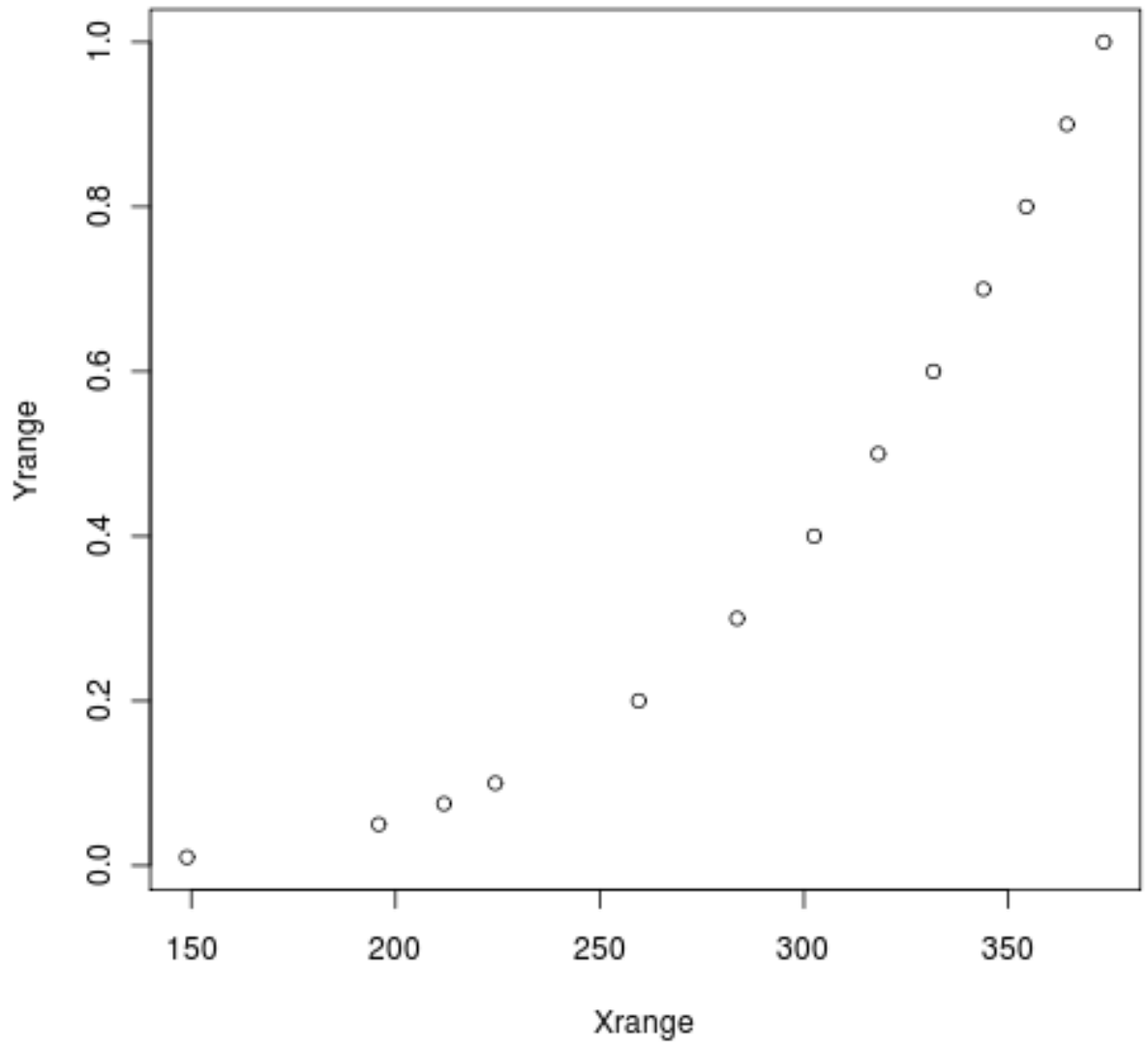
	Temperature	Position	var2	var3	var4
Name	Temperature	Position	var2	var3	var4
Label					
Type	Number	Number	Number	Number	Number
Format					
Levels					
1	148.9577	0.01			
2	195.9088	0.05			
3	211.8716	0.075			
4	224.4431	0.1			
5	259.5482	0.2			
6	283.6417	0.3			
7	302.4918	0.4			
8	318.1711	0.5			
9	331.6948	0.6			
10	343.9175	0.7			
11	354.4119	0.8			
12	364.3644	0.9			
13	373.3966	10			

And, finally, we select a scatter plot from the “Plot” menu just to have a look at the data:

point01 | Output rk_out.html | Output.data

- X variables: Temperature
- Y variables: Position

Sat Apr 23 03:21:34 2011



Run again

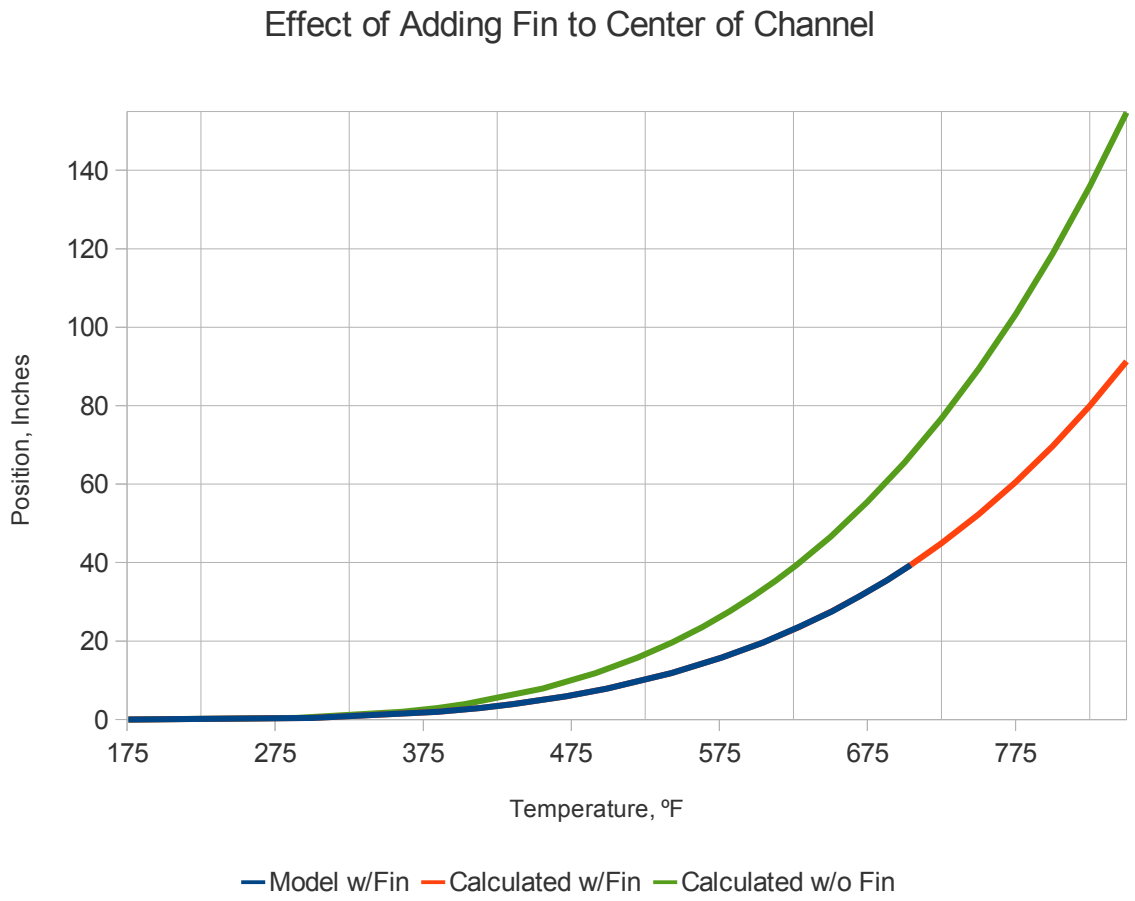
We then pass these results back to Eureqa to come up with a best-fit curve to match the data. Converting back to the units of interest, we see that the length of channel required to achieve an average temperature is:

At this resolution, it would appear that both equations derived from the Eureqa application fit the actual data from the study quite well, but Eureqa also reports that the curve "Fit 2" provides a slightly better fit to the data. Fit 2 is defined by the relationship:

$$x = 2.86e{-}10 * T^4 + 1.380e{-}7 * T^3 - 1.46e{-}4 * T^2 + 0.03325 * T - 2.36$$

This results in a prediction that the channel must be about 155 inches long to achieve the target temperature of 850 ºF.

As an additional exercise, we modified the physical geometry by adding an additional fin longitudinally in the center of the channel and repeated the analysis. Our final results:



The equation derived for the case with the central fin is:

$$x = 183.6e{-}15 * T^5 - 1.95e{-}10 * T^4 + 332.1e{-}9 * T^3 - 1.596e{-}4 * T^2 + 0.0293 * T - 1.87$$

which predicts a length of 91- ¼ inches of channel to achieve the desired temperature. This suggests that additional fins might further decrease the required length.

References:

Special Thanks to Kai Velten and Claus Meister of Hochschule RheinMain (the University of Wiesbaden, Germany), for invaluable assistance, providing key advice regarding the final data analysis.

1. CAELinux 2010, the Open Source Computer Aided Engineering Software Distribution, http://www.caelinux.com/CMS/

2. The Engineering Toolbox, a source of materials properties and other engineering information, http://www.engineeringtoolbox.com/

3. Eureqa, a powerful Open Source data modeling solution, http://ccsl.mae.cornell.edu/eureqa

4. Elmer, a multi-physics Finite Elements analysis package, included with CAELinux, http://www.csc.fi/english/pages/elmer

5. Salome, a 3D CAD, Meshing, Post Processing, and Multi-physics FE analysis package, included with CAELinux, www.salome-platform.org

6. Code_Saturne. 3D CFD solver, included with CAELinux, http://www.code-saturne.org

7. Paraview, a general purpose 3D visualization software, included with CAELinux, www.kitware.com or www.paraview.org

8. http://cfd.mace.manchester.ac.uk/, an excellent source of information about various turbulence models and other CFD details.

9. http://www.r-project.org/

10. http://rkward.sourceforge.net/

11.

Additional Links that the reader may find useful (user forums, tutorials, etc.):

Ubuntu Support Home
Ubuntu Wiki (community-edited website)
Documentation - IPythonCFD tutorial: laminar flow along a 2D channel - Part II | Free your CFD
Elmer examples — CSC
Grid Spacing Calculator
TestCases < Saturne < TWiki
European Research Community
ParaView - Elmer Wiki
CFD General Notation System
CFD-Wiki
PhD course in CFD with OpenSource software, 2010
ParaView
ParaView - KitwarePublic
ParaView: Alphabetical List
WebHome < Saturne < TWiki
THTLAB
iCFDdatabase —
ParaView/Users Guide/Table Of Contents - KitwarePublic
Use — SALOME Platform
Salome Tutorials — SALOME Platform
Python Programming Tutorial, by Richard G Baldwin
Dive Into Python

Software Carpentry » Python
Online Materials Information Resource - MatWeb
CAElinux- Adv Examples
Doc:CAETutorials - CAELinuxWiki
Open Source Physics
Free Online Course Materials | OCW Scholar | MIT OpenCourseWare
Free Online Computer Science and Programming Books, Textbooks, and Lecture Notes ::
FreeTechBooks.com
O'Reilly Open Books Project
SALOME Mesh User's Guide
CFD Discussion Forums
Forums — Code_Saturne
The Python Tutorial
The Python Standard Library
Python Glossary
ResearchGate
Scirus - for scientific information