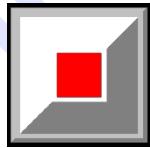


GT-SUITE

Cooling Systems and Thermal Management Application Manual and Tutorial

VERSION 2017



by
Gamma Technologies

Copyright 2016 © Gamma Technologies. All rights reserved.

All information contained in this manual is confidential and cannot be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Gamma Technologies.

GT SUPPORT

- TELEPHONE: (630) 325-5848
- FAX: (630) 325-5849
- E-MAIL: support@gtisoft.com
- Web Address: www.gtisoft.com
- Address: 601 Oakmont Lane, Suite 220
Westmont, IL 60559
USA

Telephone Support Hours

8:00 A.M. to 5:30 P.M. Central Time Monday – Friday



TABLE OF CONTENTS

CHAPTER 1: Introduction	1
1.1 Overview.....	1
1.2 Applications.....	1
1.3 Data Needed to Build a Cooling System Model	2
1.4 Data Needed to Build an Underhood Model (COOL3D)	3
CHAPTER 2: Modeling a Simple Open Flow Circuit.....	4
2.1 Getting Started	4
2.2 Introduction to the Modeling Environment	4
2.3 Importing Templates into the Project.....	6
2.4 Defining Objects	7
2.4.1 Building Pipes.....	8
2.4.2 Building Boundary Conditions	11
2.5 Placing Parts on the Project Map	12
2.6 Building a Connection	13
2.7 Linking Parts.....	14
2.8 Setting the Simulation Run Conditions.....	14
2.8.1 Run Setup.....	14
2.8.2 Case Setup	17
2.8.3 Output Setup	18
2.9 Run Simulation	18
2.10 Viewing Results in GT-POST	18
CHAPTER 3: Modeling Components of a Coolant Circuit	20
3.1 Modeling a Pump.....	20
3.1.1 Building a Pump	20
3.1.2 Testing the Behavior of a Pump	23
3.2 Modeling a Fan	26
3.2.1 Building a Fan.....	26
3.2.2 Testing the Behavior of a Fan.....	29
3.3 Modeling a Heat Exchanger.....	32
3.3.1 Theory of Heat Exchanger Modeling	32
3.3.2 Creating a Correlation from Measured Performance Data	33
3.3.3 Building a Heat Exchanger	34
3.3.4 Testing the Behavior of a Heat Exchanger	42
3.4 Modeling an Engine Block (or Heat Input Source)	46
3.4.1 Building a Heat Addition (No Mass).....	46
3.4.2 Theory of Engine Block (With Mass).....	48
3.4.3 Building an Engine Block (With Mass).....	50
3.4.4 Testing the Behavior of an Engine Block (With Mass)	54
3.5 Modeling a Thermostat	55
3.5.1 Building a Thermostat	55
3.5.2 Building a Thermostat Bypass.....	58
3.5.3 Testing the Behavior of a Thermostat.....	61
CHAPTER 4: Modeling a Hydraulic System (No Heat Transfer)	63



Table of Contents

4.1	Required Components for a Hydraulic System.....	63
4.2	Run Setup and Disabling Heat Transfer (Hydraulic Settings).....	64
4.3	Testing a Hydraulic System.....	66
CHAPTER 5:	Modeling a Steady-State Thermal System	70
5.1	Required Components for a Steady-State Thermal System.....	70
5.2	Heat Transfer in Pipes (Wall Temperature Solver)	75
5.3	Run Setup and Enabling Heat Transfer.....	77
5.4	Testing a Thermal System	78
CHAPTER 6:	Modeling an Underhood System with COOL3D.....	81
6.1	Getting Started	81
6.2	Introduction to the Modeling Environment	81
6.3	Modeling a Cooling Package	82
6.3.1	Linking to an Object Library	82
6.3.2	Building a Heat Exchanger	83
6.3.3	Building a Heat Addition.....	88
6.3.4	Building a Flow Space.....	91
6.3.5	Building a Blockage	92
6.3.6	Creating Boundary Conditions	94
6.3.7	Case Setup	97
6.4	Running an Underhood Model from GT-ISE	98
6.4.1	Placing Parts, Connections, and Links for an Underhood Model.....	98
6.4.2	Creating Circuits for an Underhood Model	100
6.4.3	Testing an Underhood Model	103
6.5	Calibrating an Underhood Model	104
6.5.1	Creating a Pressure Resistance Plane	105
6.5.2	Calibrating the System Pressure Drop (Flow Rate).....	108
6.6	Case Studies with Changes in Geometry	109
CHAPTER 7:	Modeling a Transient Thermal Model	112
7.1	Preparing a Transient Thermal Model	112
7.1.1	Modeling a Pressure Regulation Device (Accumulator or Overflow Tank)	112
7.1.2	Modeling the Engine and Vehicle	114
7.1.3	Imposing the Vehicle Speed Boundary Conditions for the Underhood System.....	114
7.1.4	Modeling a Simple Fan Controller	115
7.1.5	Setting the Driving Cycle Profile.....	116
7.1.6	Setting the CAC Boundary Conditions.....	117
7.1.7	Coloring a Circuit (Optional).....	117
7.2	Run Setup and Output Setup.....	117
7.2.1	Run Setup.....	117
7.2.2	Output Setup	118
7.3	Testing a Transient Thermal Model.....	119
CHAPTER 8:	Advanced Integration of Engine and Cooling Systems.....	122
8.1	Overview.....	122
8.2	Indirect (Partial) Integration	125
8.2.1	Engine (GT-POWER) Model Setup	125



Table of Contents

8.2.2	Cooling System Model Setup	127
8.3	Direct (Full) Integration.....	130
8.3.1	Engine Model Setup	130
8.3.2	Cooling Model Setup.....	130
8.3.3	Integrating Engine and Cooling Model	131
8.3.4	Other setup issues	132
CHAPTER 9:	Cooling System FRM Creation.....	133
9.1	Overview.....	133
9.2	Prepare System Model	133
9.3	Choose the First Flow Branch to Simplify	134
9.4	Setup and Run the "Virtual Flowbench" Model	135
9.4.1	Create Pressure Drop Reference Object	137
9.5	Create Simplified Parts in Full System Model.....	138
9.5.1	Run Model for Comparison to Full System.....	141
9.5.2	Simplify the remaining Flow Branches	142
9.6	Run FRM Model and Compare Results.....	143



CHAPTER 1: Introduction

1.1 Overview

Because engine thermal management tasks are typically distributed over several disciplines and they concern different sub-systems such as coolant, transmission, oil circuit, intercooler, heater and A/C system, and components such as thermostats, radiators, fans and pumps, a comprehensive CAE design tool is required to integrate all of these related activities. **GT-SUITE** is an advanced code for design and analysis of engine thermal management systems that has been specifically designed to handle this wide range of heat management issues and disciplines. As a result, GT-SUITE can be used to integrate all of the activities related to heat management to meet design objectives such as engine durability, emissions, fast warm-up, passenger cabin heating, component sizing and specification, and system control.

GT-SUITE is based on one-dimensional fluid dynamics, representing the flow and heat transfer in the piping and in the other components of a cooling system. Several parallel fluid circuits can be modeled simultaneously, each containing a different fluid (water/glycol mixture, oil, transmission fluid, air, etc.). These circuits interact through heat exchangers, transferring heat from one circuit to another, which allows the calculation of the overall heat balance in the system. In addition to the fluid flow and heat transfer capabilities, the code contains many other specialized models required for system analysis. By being comprehensive, the code is well suited for integration of all heat management activities arising in engine and vehicle development. In addition, it provides the flexibility needed to model advanced concepts, and it also includes a built-in vehicle/engine simulation for calculation of thermal loads under any driving cycle.

GT-SUITE features an object-based code design that provides a powerful model building facility and reduces user effort. Models are built by a highly versatile graphical user interface, GT-ISE (Integrated Simulation Environment), common to all applications which simplifies the task of managing object libraries and building, editing, executing and post-processing models. GT-ISE minimizes the amount of input data entry, as only *unique* geometrical elements must be defined; this reduces required data input. Models are built by this point-and-click GUI from a library of GT-supplied or user-defined reusable components. GT-SUITE is supported by an experienced team of developers and support staff.

1.2 Applications

GT-SUITE can be used for a wide range of activities relating to engine thermal management. Typical applications include:

- Analysis of the entire cooling circuit
- Transient or steady state engine operation
- Radiator sizing
- Pump sizing
- Pipe and orifice sizing
- Engine warm-up
- Thermostat specification
- Flow distribution of an underhood system
- System operation during a prescribed vehicle cycle (e.g. FTP):
 - vehicle speed
 - engine speed and torque
 - engine heat rejection to coolant



- ram air effects
- fan viscous clutch operation
- thermostat cycling
- Design of active and passive control systems

1.3 Data Needed to Build a Cooling System Model

A list of data generally used in a basic cooling system model can be found below. The list is not comprehensive as data requirements can change depending on the intended use of the model.

- General
 - Fluid properties
 - System schematic
 - Pressure regulation schematic (including tank volume)
- Pipes and Flowsplits (or CAD file with geometry)
 - Inlet and outlet diameter
 - Geometry for bent pipes
 - Surface roughness
 - Wall thermal properties
- Thermostats
 - For conventional wax thermostats – lift vs. wax temperature (opening and closing)
 - Pressure drop vs. flow rate at varying lifts, or effective flow area vs. lift
 - Relationship between lift of other valves for multiple valve thermostats
 - Time constant during transient operation
- Pumps and Fans
 - Pump performance data (volumetric flow rate as a function of pressure rise, speed, and efficiency)
 - Fan performance data (flow rate as a function of pressure ratio/rise, speed, and efficiency) with inlet reference conditions
- Engine and Engine Block
 - Heat rejection to engine block as a function of engine speed and load
 - Mass of structural material in engine block and head
 - Volume of fluid in engine block and head
 - Material properties
 - Pressure drop data: this can be one reference data point, a table of pressure drop vs. flow rate, or a power law equation of pressure drop vs. flow rate
 - Heat transfer coefficient between fluid and wall
- Heat Exchangers (HxMaster/HxSlave)
 - Volume of both fluid sides
 - Mass of structural material
 - Material properties
 - Geometry of heat exchanger (i.e. core height, core width, number of tubes, tube geometry, fin geometry, etc)
 - Experimental performance data containing the inlet temperature, pressure and flow rate for both fluid sides, and a performance quantity (effectiveness, overall heat transfer rate, outlet temperature for a single fluid side, or temperature ratio) and the fluids used in the tests
 - Pressure drop vs. flow rate for both fluid sides and the fluids used in the tests
- Experimental System Data for Calibration
 - Flow rates, pressures and temperatures at all inlets and outlet of major components and branches



1.4 Data Needed to Build an Underhood Model (COOL3D)

A list of data generally used in an underhood model can be found below. Most of the information below (dimensions, locations, etc) can be obtained from a CAD file.

- General
 - Location in 3D space and dimensions of heat exchangers
 - Location in 3D space and dimensions of fan and fan shroud
 - Location in 3D space and dimensions of any blockages (e.g. front fascia, grille openings, engine block, etc)
 - Location in 3D space and dimensions of any flow openings (e.g. underbody opening in engine compartment)
 - Inlet and outlet boundary conditions (ambient pressure, ambient temperature, C_p , and ram velocity, inlet and outlet pressure and temperature, or inlet and outlet flow rate and temperature)
 - Fan speed
 - External fluid type (including % humidity, if applicable)
- Heat Exchangers
 - Volume of both fluid sides
 - Mass of structural material
 - Material properties
 - Geometry of heat exchanger (i.e. core height, core width, number of tubes, tube geometry, fin geometry, etc)
 - Experimental performance data containing the inlet temperature, pressure and flow rate for both fluid sides, and a performance quantity (effectiveness, overall heat transfer rate, outlet temperature for a single fluid side, or temperature ratio) and the fluids used in the tests
 - Pressure drop vs. flow rate for both fluid sides and the fluids used in the tests
- Fans
 - Fan moment of inertia
 - Fan performance data (flow rate as a function of pressure ratio/rise, speed, and efficiency) with inlet reference conditions
- Experimental Data for Validation
 - Flow field results taken from various cross sections along the length of the cooling air flow path (pressure, temperature, and velocity)
 - Total air mass flow rate through each component
 - Outlet temperature for the internal fluid of all heat exchangers
 - Heat transfer rates of all heat exchangers



CHAPTER 2: Modeling a Simple Open Flow Circuit

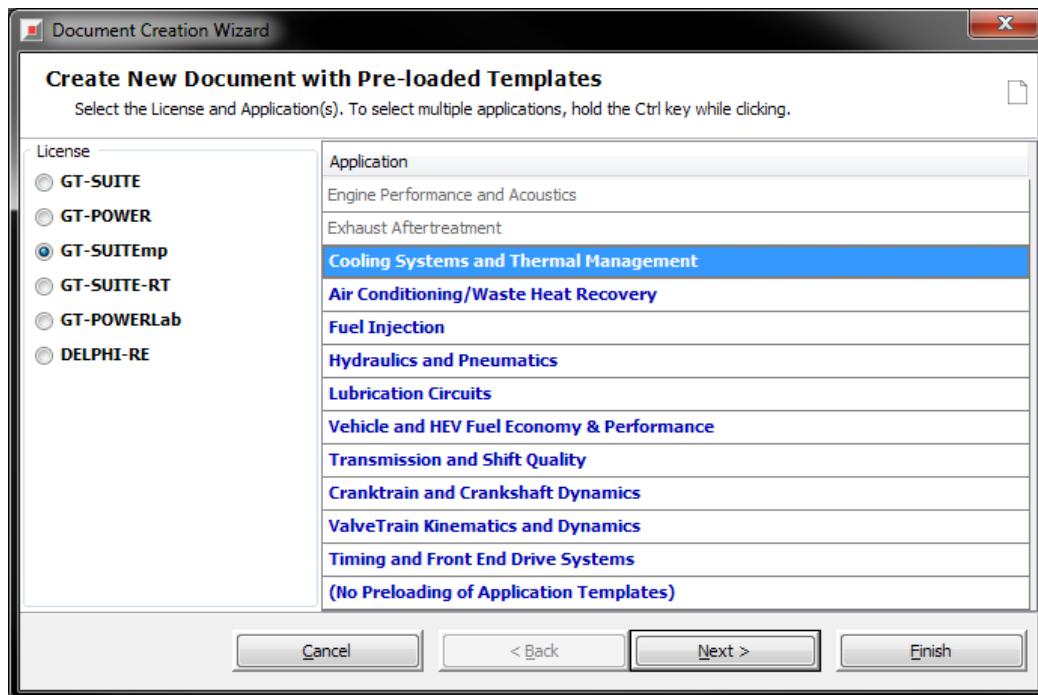
This tutorial has been prepared to assist new users of GT-SUITE for cooling system model building to learn how to use the software and get familiar with the user interface. A simple flow circuit model will be built to understand how the flow solver works by running simple steady flow model. Throughout this tutorial, it is recommended that one reads the description of each object and attribute from the online help for additional information related to that object.

2.1 Getting Started

Launch GT-ISE: If working on a PC, the graphical user interface for GT-SUITE and GT-SUITE-MP can be started by double-clicking on the GT-ISE (Gamma Technologies-Integrated System Environment) icon. If an icon has not been created, one can map a new icon by pointing to \$GTIHOME\Vx.x.x*\GTsuite\bin\win32\GTise.exe. From UNIX or PC, one can also launch the program by typing *gtise* at the command line.

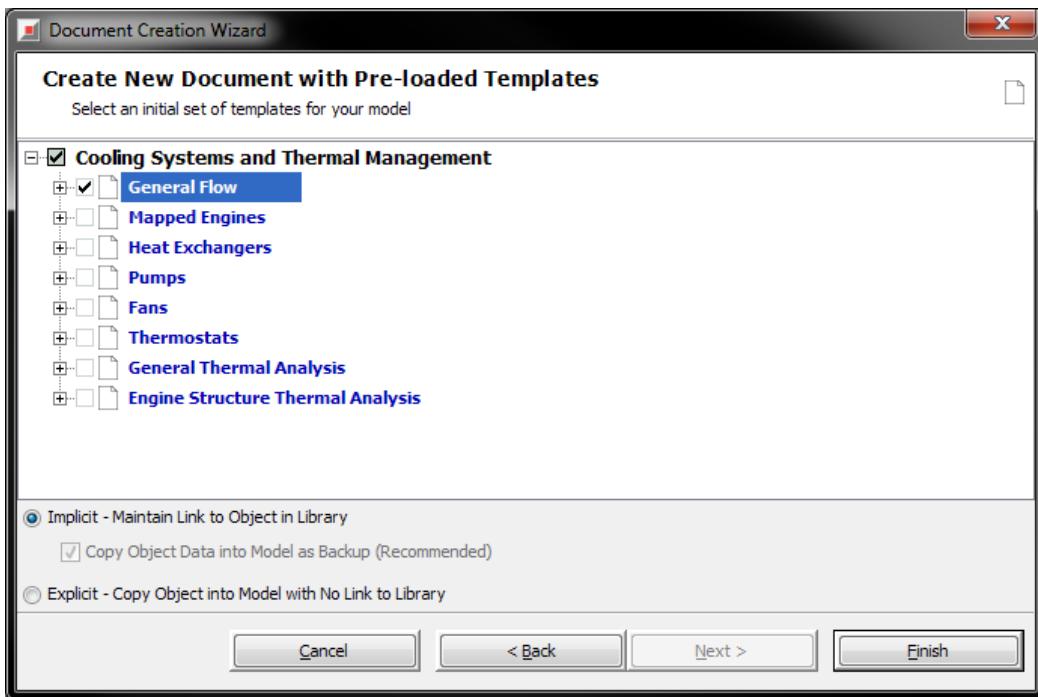
2.2 Introduction to the Modeling Environment

Once the program has started, a blank window will be created. In the **Home** tab, select **New**. This will load the "Document Creation Wizard" shown below. From this pane, one can see the applications supported by particular licenses highlighted in blue font, whereas unsupported applications are shown with light grey text. Templates related to cooling applications that are needed in this tutorial are available with either GT-SUITE or GT-SUITE-MP, so select one of these options. This license choice may be changed later without disruption provided no unsupported templates are involved. Also, from the Document Creation Wizard window, one may highlight a particular application to preload a set of templates for a project by clicking on a field, such as "Cooling Systems and Thermal Management". This feature is beneficial to use because it will automatically import multiple templates into the project tree without the need to search for them.



Tutorial 2: Modeling a Simple Open Flow Circuit

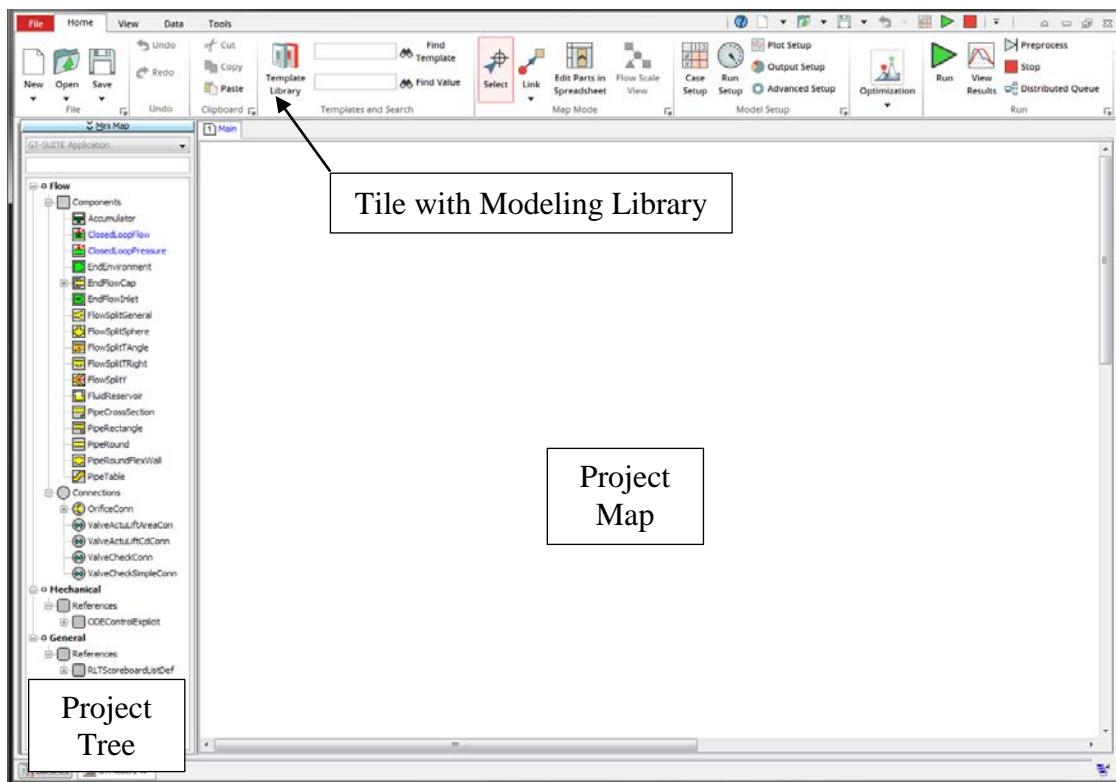
Click Next after selecting the "Cooling Systems and Thermal Management" option. In the next window, an option will be provided to select the templates that will be automatically imported into the new model. From this step, only select the option for **General Flow**, which include templates for boundary conditions, pipes, and flow connections. The simple model that will be built will only include flow boundary conditions and pipes. Click Finish after the **General Flow** section is selected to generate the new model.



The project view at this time should look like the following:



Tutorial 2: Modeling a Simple Open Flow Circuit

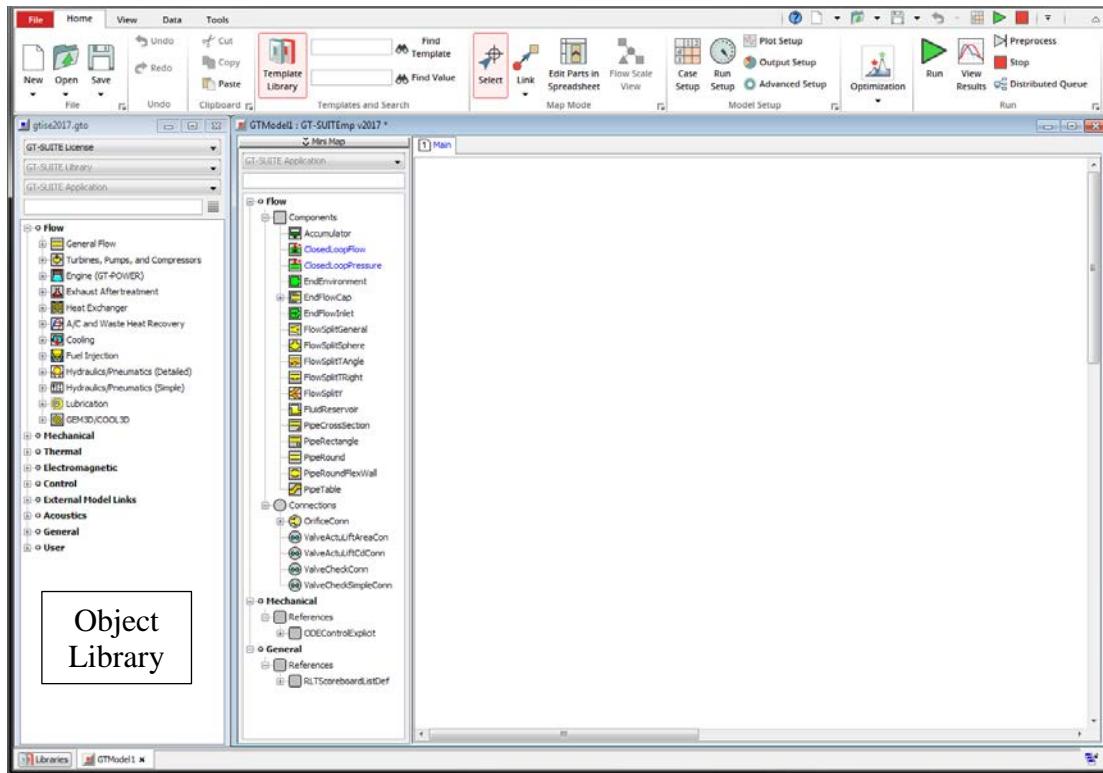


2.3 Importing Templates into the Project

Sometimes it is necessary to use other templates besides what is provided with the Document Creation Wizard, or to build on an already existing model. To add additional templates to the project, they will need to be located in the **Template/Object Library**. To access the template/object library, select the **Tile with Modeling Library** button. The template library contains all of the available templates that can be used in GT-SUITE.



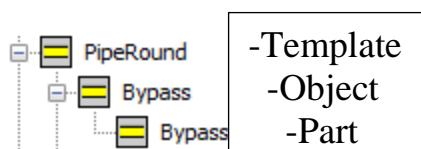
Tutorial 2: Modeling a Simple Open Flow Circuit



The **Find** feature can be used to help locate various templates and objects in the template/object library. There are two ways to use the Find feature. The first is to locate the **white box and binoculars** button on the toolbar, type part or the entire template/object name, and then pressure the binoculars button. The second way is to simultaneously press **Ctrl+F**. In order to use either Find feature, the template/object library must be the window that is in focus/active. Once a template/object has been found, it can simply be included into the project map by performing a drag-and-drop function.

2.4 Defining Objects

The modeling format in GT-ISE uses an object-oriented structure. This structure is comprised of a three-level hierarchy: **templates**, **objects**, and **parts**. Templates are provided which contain the unfilled attributes needed by the model within the software. The templates are made into objects, and when component and connection objects are placed on the project map they become parts and inherit their values from their parent objects. These parts may call reference objects like fluid properties, material properties, map lookups, etc. During the course of building the model many reference objects will be used, and most are automatically imported into the project at the time they are needed.

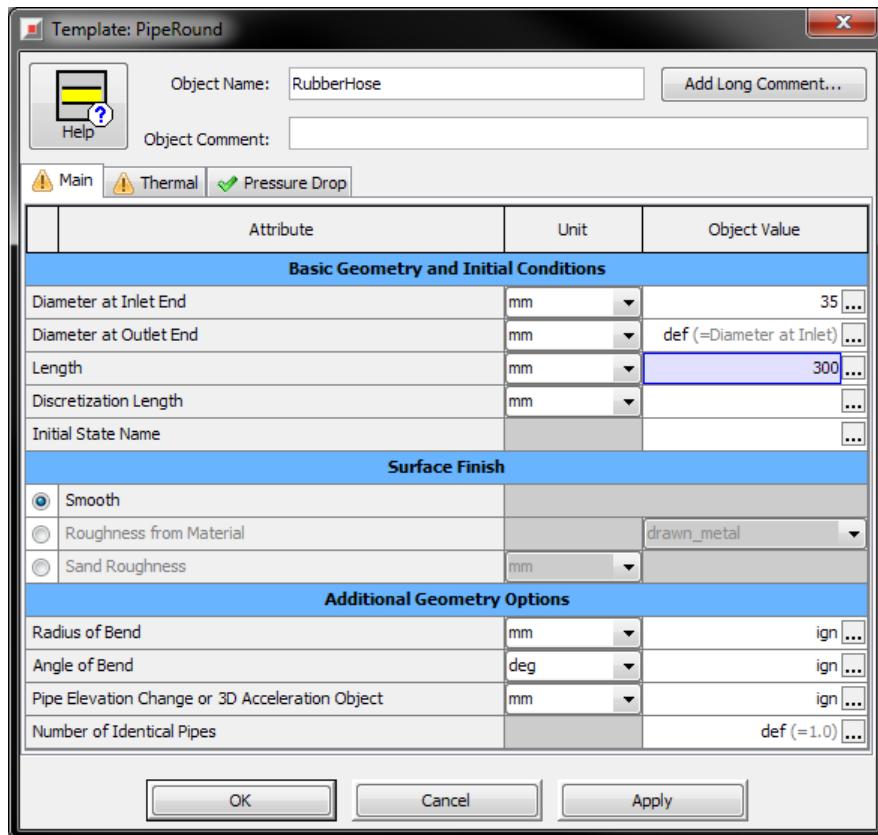


Before starting to build the simple flow circuit model, select **Save** while in the **Home** tab to save the project.



2.4.1 Building Pipes

The first step to modeling the open flow circuit will be to build a pipe. To do so, double-click on the 'PipeRound' template (or right-click on the template and select Add Object) in the project tree to open the object dialog. Name the object 'RubberHose' and fill out the pipe geometry and roughness as seen below. This information is typically acquired by measuring the pipe dimensions and taking note of the material.

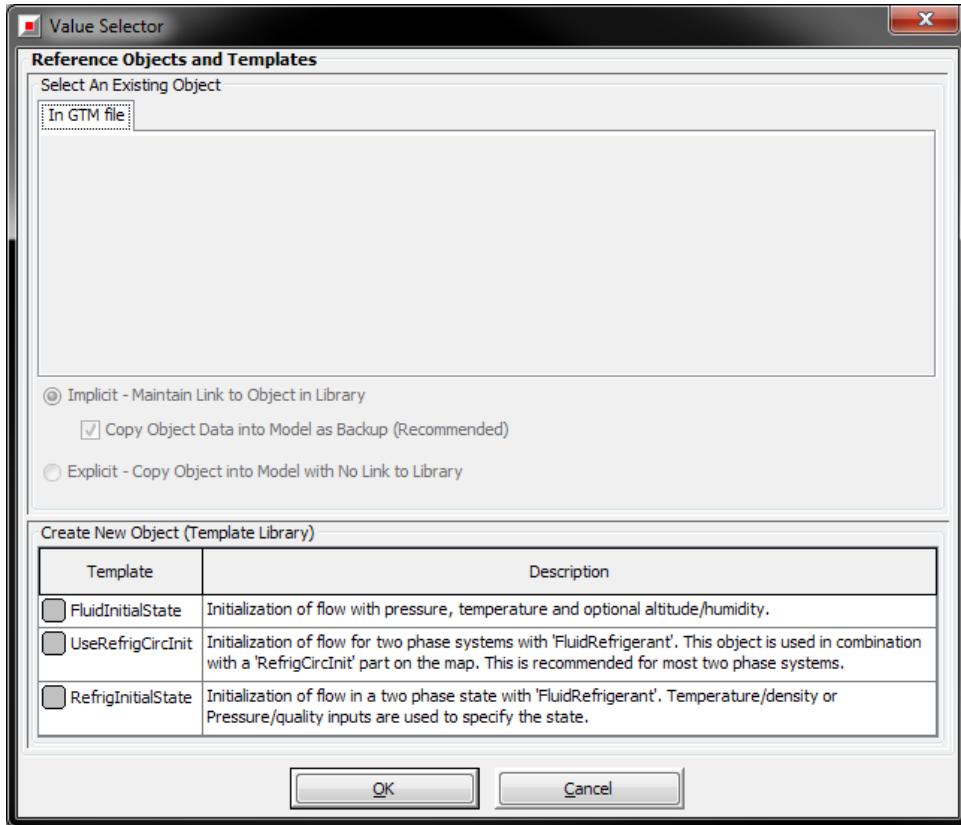


The attribute **Discretization Length** can play a role in the results of the model, but usually in the form of how the results are reported. In addition to this, the more subvolumes created by the discretization length can affect the simulation run time since each subvolume will be performing its own calculations. For more details on pipe discretization length, please refer to section **1.3 Discretization** in the Flow Theory manual (File→Manuals→Modeling_Theory). For cooling system models, a discretization length on the order of 30 - 50 mm is recommended (50 mm will be used in this case).

The attribute **Initial State Name** is used to set the initial fluid conditions at $t = 0$ for the flow volume. Each part on the map can have a unique object defined for the initial state, but it is common to reuse the same object for the complete circuit such that it is initialized to the same conditions. If running a model to steady-state, the initial conditions of the flow circuit will have little influence on the model results, but if running a transient warm-up, then the initial conditions can have a large impact in the performance of the model. To define the initial conditions for the pipe, simple click on the button that looks like three dots [...] to open the **Value Selector**.



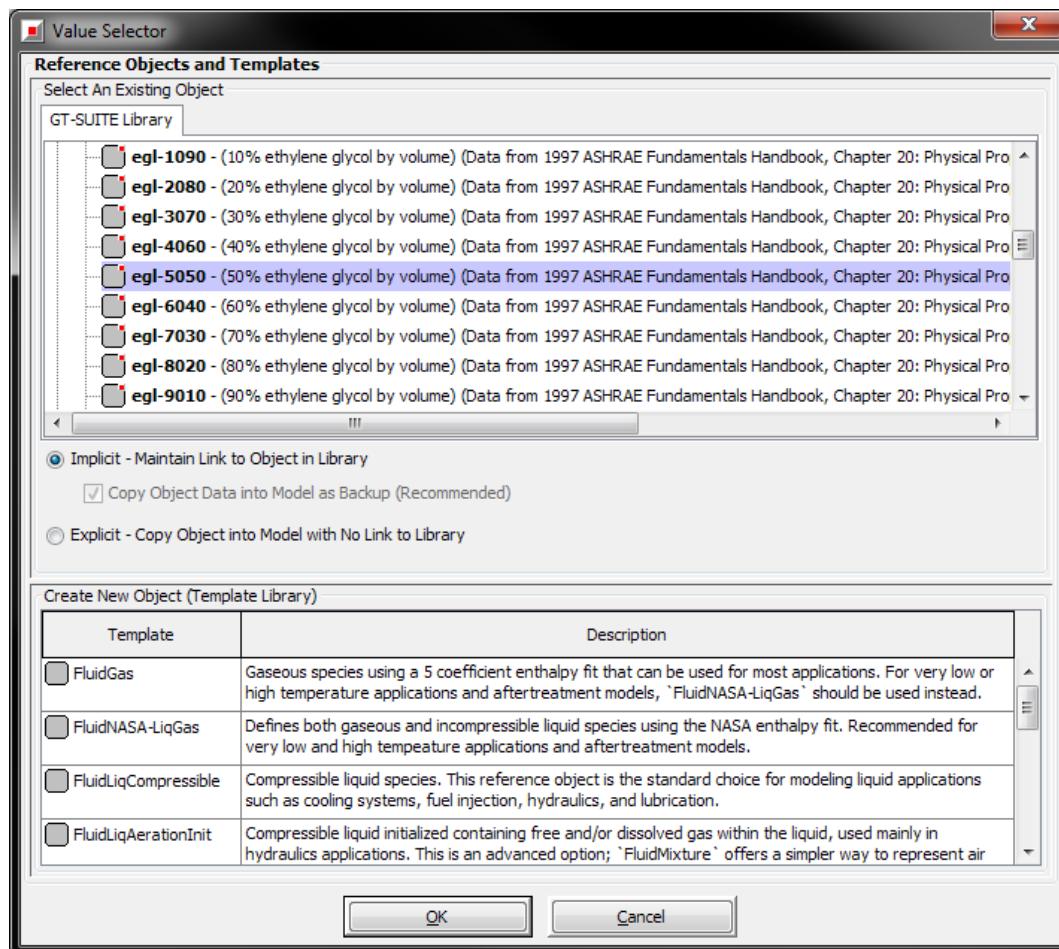
Tutorial 2: Modeling a Simple Open Flow Circuit



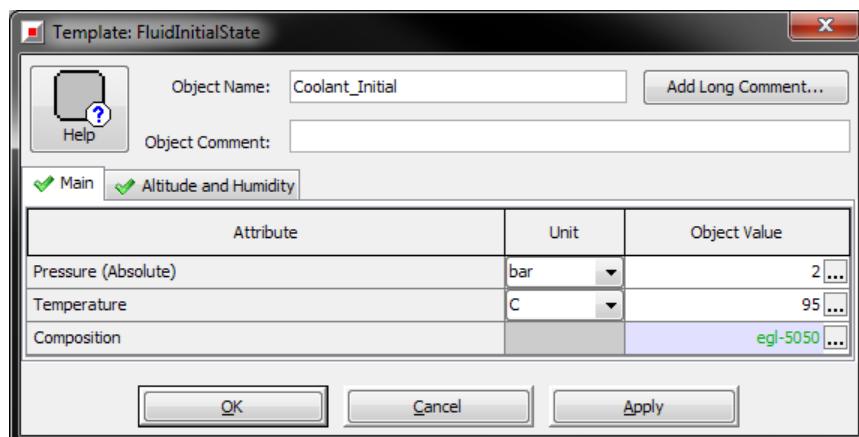
The Value Selector is used to quickly point to reference objects that exist in the model, or to point to objects that may exist in the template/object library. Reference objects are typically defined as an object that contains a set of data that is reused among different objects. This allows a model to be built quickly since the same information can be referenced without having to fill it out again. Select the 'FluidInitialState' object to define the initial fluid conditions in the pipe object. In the 'FluidInitialState' object dialog that opens, provide an object name of 'Coolant_Initial'. Set the initial fluid conditions as 2 bar and 95 C. For the **Composition**, use the Value Selector to find the fluid object 'egl-5050' ('FluidLiqCompressible') in the GT-SUITE Library.



Tutorial 2: Modeling a Simple Open Flow Circuit

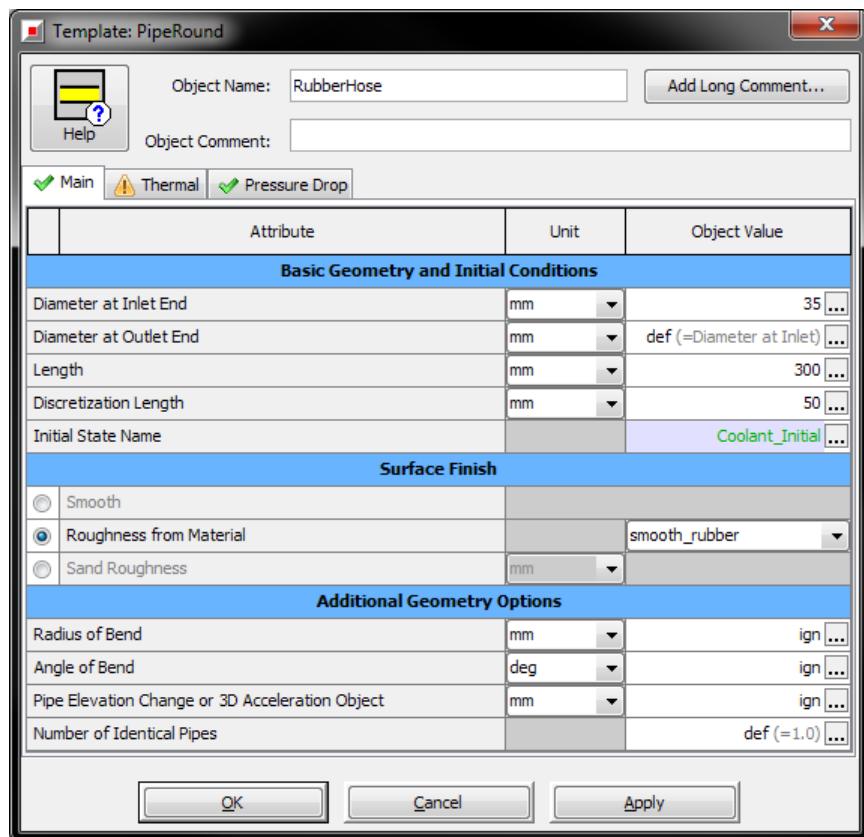


Upon confirming the object selection, the fluid properties for 'egl-5050' will be automatically imported to the project tree and the **Composition** will refer to the object name. The green text signifies the use of a reference object. Click OK when the initial conditions have been defined.



Back in the pipe object, the **Initial State Name** will be filled in with the name of the 'FluidInitialState' reference object that was just created. Every subvolume created in this pipe object will be initialized with the pressure, temperature, and composition defined in this object.





The **Roughness from Material** attribute can be set to 'smooth_rubber' to automatically select the surface roughness of a rubber pipe to model the pressure drop due to friction.

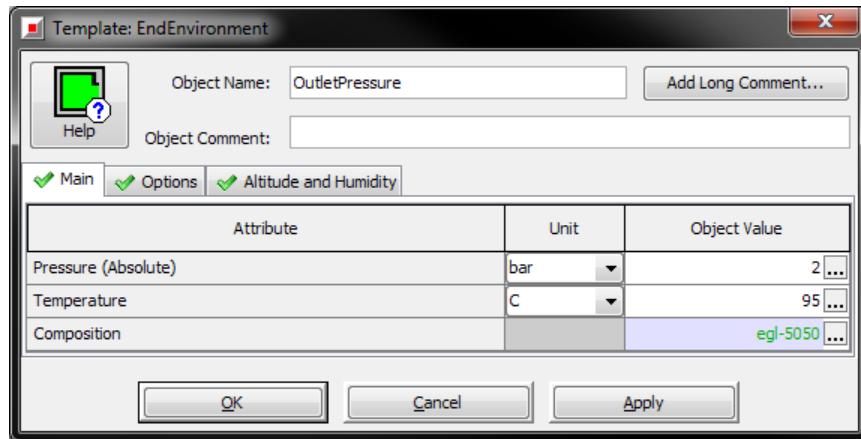
In the Thermal folder, set the Wall Temperature Method to **Adiabatic**. The heat transfer rate between the fluid and the wall will be ignored at this time. Click OK when the selection is made to finish defining the pipe.

2.4.2 Building Boundary Conditions

The next step will be to build the boundary conditions for the simple flow network. To do so, double-click on the 'EndEnvironment' template in the object tree to open the object dialog to set a pressure boundary condition. Name the object 'OutletPressure' and fill out the outlet Pressure, Temperature, and Composition for the boundary to define the outlet condition similar to what might have been used in a test bench. The Value Selector can be used to point to the 'egl-5050' object that was imported in the model.

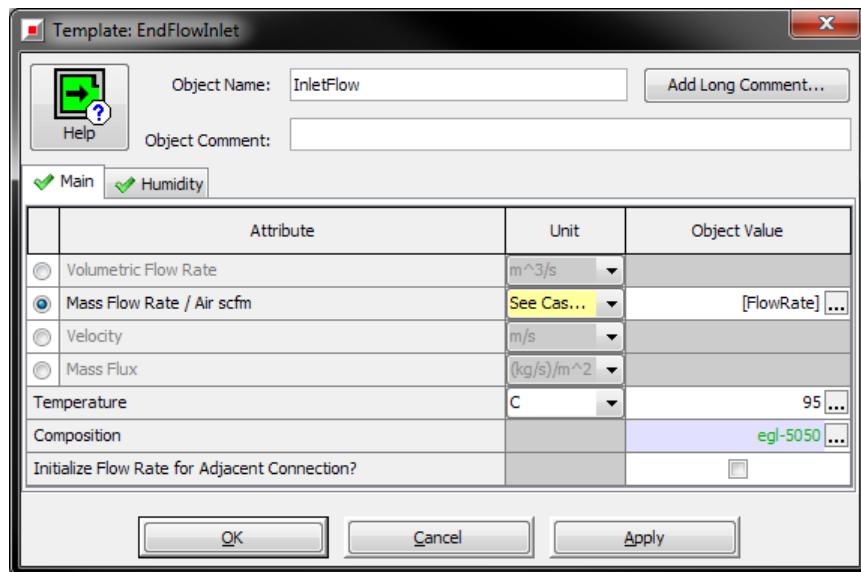


Tutorial 2: Modeling a Simple Open Flow Circuit



Because this object will be used as an outlet boundary, the Temperature and Composition will only matter if the flow reverses back into the model, or if a negative flow rate is seen at the neighboring connection. Click OK when the object is complete.

To define the inlet boundary condition, double-click on the 'EndFlowInlet' template in the object tree. This object will be used to define an inlet flow boundary condition (flow rate). Name the object 'InletFlow' and select the Mass Flow Rate radio button option. The mass flow rate will be imposed with a value that might have been measured on a test bench, but will be done with the use of a parameter so that a case sweep can be performed to test the pressure drop of the flow circuit at different flow rates. A parameter is defined with the start and end of brackets, [], that surround a character string. Type the parameter [FlowRate]. The values will be defined at a later time. Set the Temperature and Composition as seen below. Click OK when finished.



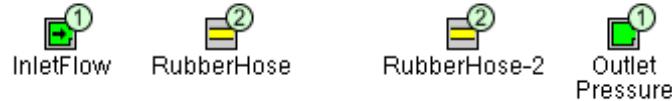
2.5 Placing Parts on the Project Map

To place parts on the map, a drag-and-drop operation can be used. Simply drag the objects 'OutletPressure', 'InletFlow', and 'RubberHose' from the tree to the map. For the 'RubberHose', two parts



Tutorial 2: Modeling a Simple Open Flow Circuit

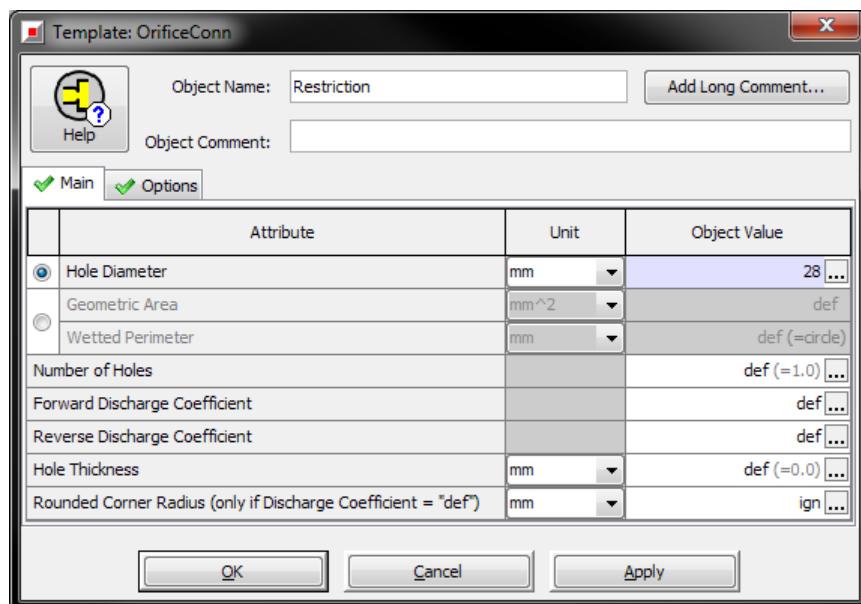
will be created from the one object. To do so, drag a second 'RubberHose' object from the tree to the map.



The numbers enclosed by a circle signify the number of required connections that are necessary to complete the connectivity.

2.6 Building a Connection

To create a connection, double-click on the connection template 'OrificeConn'. Name the object 'Restriction', and enter a hole diameter (as seen below) to model a restriction between the two pipes. This restriction will add an additional pressure drop to the simple flow circuit. For more details on orifice pressure drop, please refer to section **2.2.1 Orifices** in the Flow Theory manual (File→Manuals→Modeling_Theory). Click OK when finished.



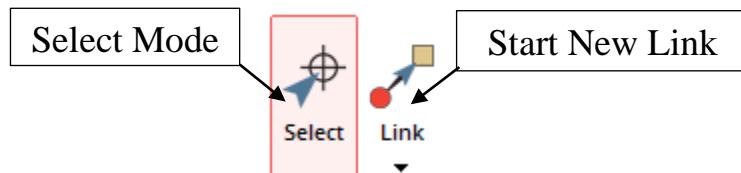
Drag the newly created orifice object onto the map between the two pipe parts.



It is not required to manually create a connection between two components on a map. In most cases, when a link is created to connect two components a default connection will be added automatically. In this case, the diameter of a default orifice connection will be the smaller diameter of the two neighboring flow parts.

2.7 Linking Parts

To connect two parts together, select the **Start New Link** button in the top toolbar, or right-click on the map and select the option from there. Up until now, the **Select Mode** was used which is selected to move parts around the map.



Connect the parts on the map from left to right (as seen in the image below). A default orifice connection will be placed between the parts if one does not already exist.



2.8 Setting the Simulation Run Conditions

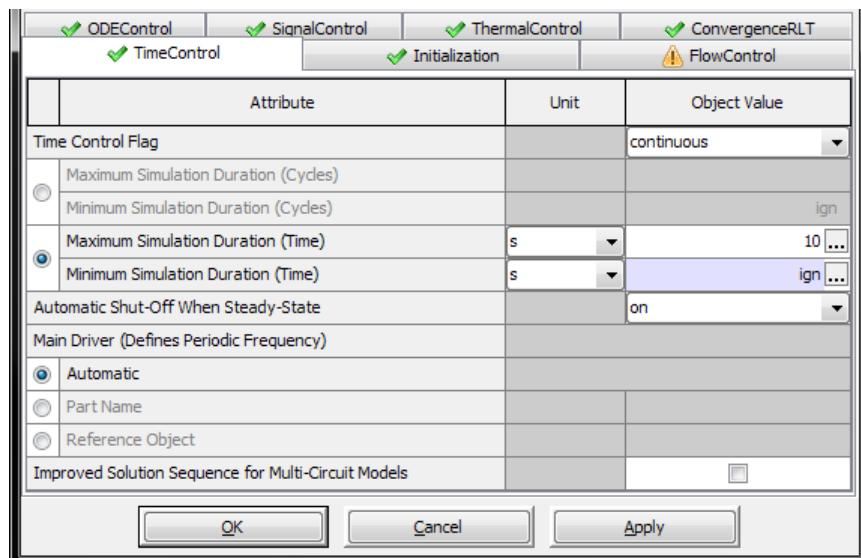
Now that the model has been built, the information such as the type of simulation, simulation duration, parameter input, and desired output must be declared. This will be accomplished in **Run Setup**, **Case Setup**, and **Output Setup**.

2.8.1 Run Setup

To define the type of simulation and simulation duration go to **Run Setup**, which can be found in the **Home** tab. There are many folders in Run Setup, but the most commonly used ones for cooling system simulations are **TimeControl**, **FlowControl**, and **ThermalControl**. The **ConvergenceRLT** folder is used on occasion. In the **TimeControl** folder one can define the simulation type, 'continuous' or 'periodic', as well as the simulation duration settings. Most cooling system simulations are time based, so select the 'continuous' option for the Time Control Flag. The 'periodic' option should only be selected if the simulation needs to be run in a cyclic nature (i.e. engine modeling). Since 'continuous' is selected, the second radio option for the Simulation Duration (Time) should be chosen. Set the maximum simulation duration to 10 sec.



Tutorial 2: Modeling a Simple Open Flow Circuit



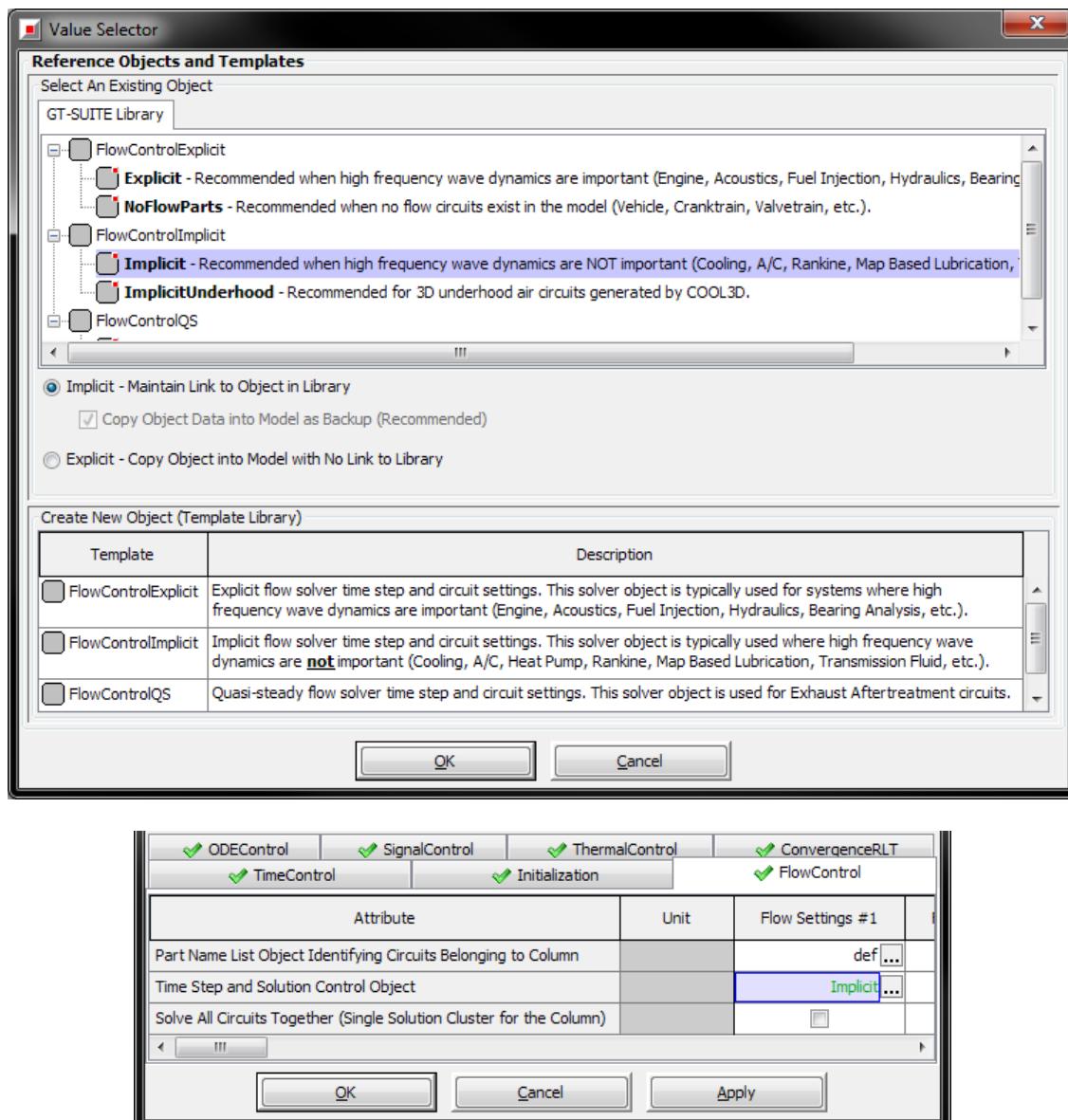
The Automatic Shut-Off When Steady-State will be left as 'on', which means it is possible for the simulation to end prior to the 10 sec defined earlier if a certain set of criteria is met for the fluid flow. These criteria will be defined in the **FlowControl** folder, and optionally in the **ConvergenceRLT** folder. If the Automatic Shut-Off When Steady-State is 'off', then the simulation will run for the full Maximum Simulation Duration regardless of the criteria set for steady-state.

The **FlowControl** folder is where the solver settings are defined for the flow circuit, as well as a basic setting for the steady-state convergence criteria. There are 3 types of solvers available in GT-SUITE: explicit, implicit, and quasi-steady. The explicit solver is used for models where the effects of wave dynamics are of interest (i.e. engine modeling). The implicit solver is used for models where wave dynamics are not of interest, and the flow is steady (i.e. cooling systems). The quasi-steady solver is used for models where chemistry is of interest (i.e. aftertreatment). The implicit solver uses an imposed time step calculation method to solve for the fluid dynamics throughout the simulation. There is the concept of time step convergence which is checked before advancing to the next time step. Additional information on the solver options can be found in the online help for each template, or in section **1.2 Time Step Calculation** in the Flow theory manual (File→Manuals→Modeling_Theory).

Use the Value Selector for the Time Step and Solution Control object to select the 'Implicit' object option from the library.



Tutorial 2: Modeling a Simple Open Flow Circuit



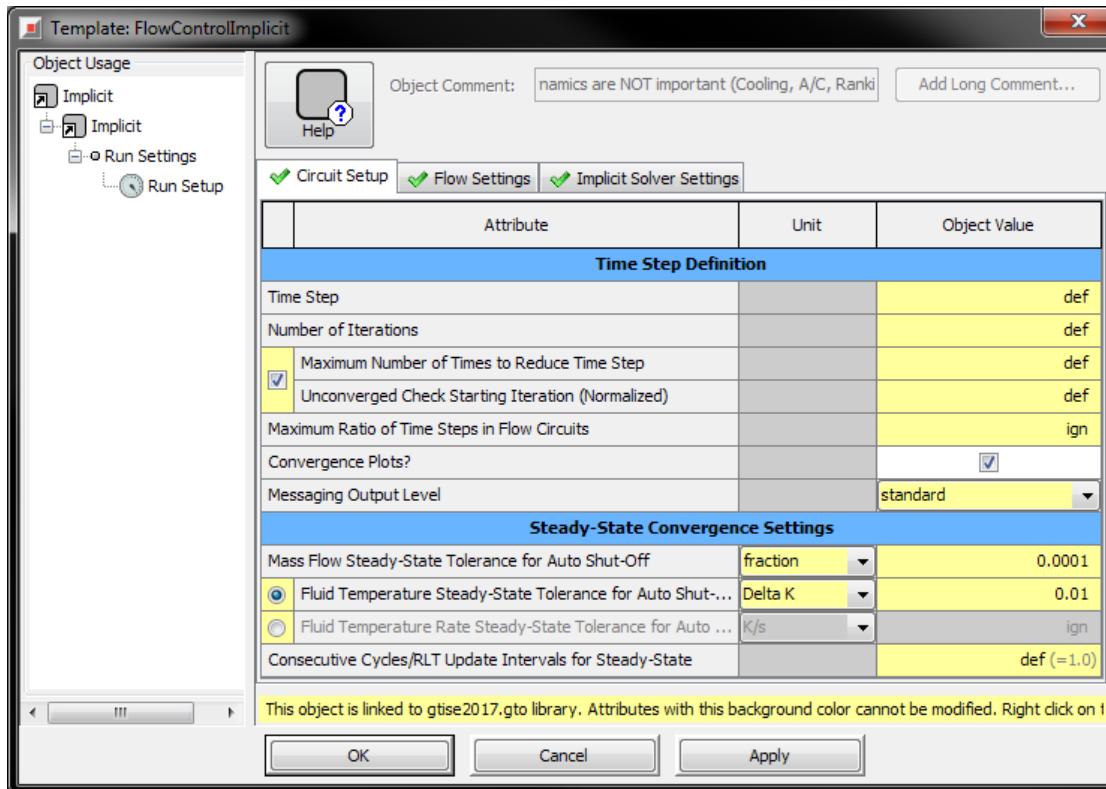
This object uses the recommended implicit solver settings for cooling system modeling, as well as a predefined set of criteria that will be used for the Automatic Shut-Off When Steady-State settings. The **Circuit Setup** folder contains the information related the calculation of the time step, as well as the definition for steady-state convergence criteria. The default (and recommended) time step for the implicit solver is 0.1 seconds.

The **Flow Settings** folder defines the global multipliers for friction and heat transfer for all flow volumes in the circuit. For example, the Global Heat Transfer Multiplier would be used to turn off all heat transfer in the model between the fluid and the wall for all flow volumes in the circuit.

The **Implicit Solver Settings** allows one to control how the implicit solver works. It is recommended to read the online help for each attribute prior to making adjustments to this folder.



Tutorial 2: Modeling a Simple Open Flow Circuit



There are multiple columns in the **FlowControl** folder so that it is possible to set a different solver type each flow circuit, or group of flow circuits. This tutorial only deals with a single circuit, so click OK to finish Run Setup.

2.8.2 Case Setup

In **Case Setup** (accessible from the **Home** menu), the parameter [FlowRate] that was set for the inlet boundary conditions can be defined. It is here that a case sweep can be set up to impose different flow rate conditions to predict the pressure drop in the flow circuit.

For this model, select Append Case to create an additional case, and enter a mass flow rate of 1 kg/s for Case 1, and 2 kg/s for Case 2.

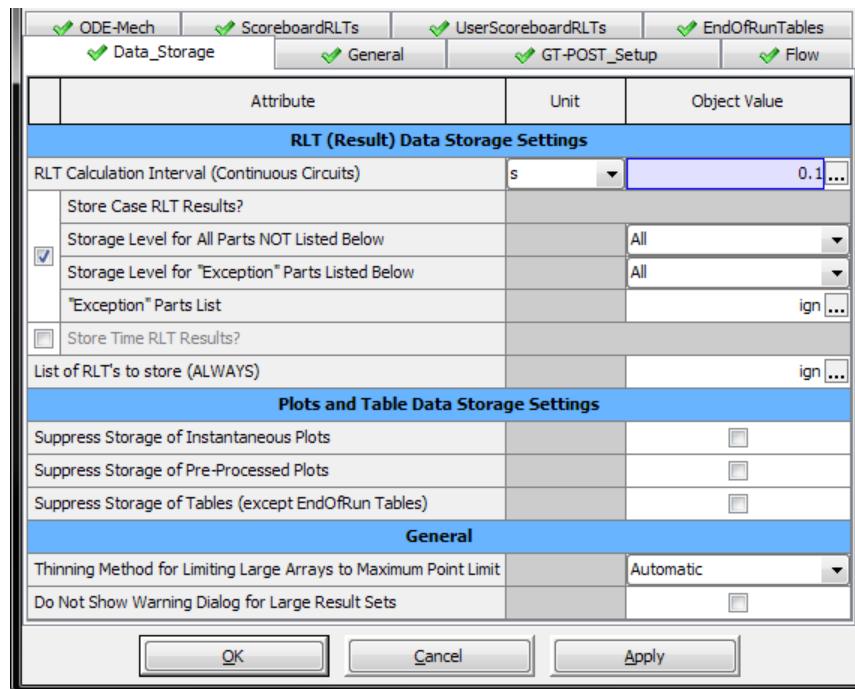
Main	All	+ New		
Parameter	Unit	Description	Case 1	Case 2
Case On/Off		Check Box to Turn Case On	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Case Label		Unique Text for Plot Legends		
FlowRate	kg/s	Mass Flow Rate / Air scfm	1 ...	2 ...

If there were other parameters in the model they would have appeared in a list form to be filled out as well. It is also possible to reorganize the parameters into different folders, similar to Excel. A Case # will only run in the corresponding checkbox is turned "on". Click OK when finished to complete and close Case Setup.



2.8.3 Output Setup

Output Setup (found in the **Home** tab) is where data storage is defined. RLTs (**ResuLTs**) are defined by the settings in the **Data_Storage** folder. It is recommended to change the RLT Calculation Interval to be on the same order of magnitude as the flow solver time step, which is 0.1 seconds. This way data storage, and data lookups, are performed at the same interval as the flow solver. In addition, this interval is used when performing the Automatic Shut-Off for Steady-State checks with the criteria defined.



The storage for Case RLTs is "on" by default. This allows the final, end of run values like pressure, temperature, flow rate, heat transfer rate, etc to be stored when the simulation completes. Turning this "off" will disable the storage of this data. It is optional to turn "on" the Time RLTs if there is interest to view RLTs at the RLT Calculation Interval. Click OK after the RLT Calculation Interval is defined to complete **Output Setup**.

2.9 Run Simulation

Now the model is ready to run. Press the **Run Simulation** button (▶) found in the **Home** toolbar to launch the solver. At the same time the model will be saved automatically. A new window will pop up to watch the progression of the model. Allow the simulation to finish running. It will be finished when the .gdx file (results file) is created.

2.10 Viewing Results in GT-POST

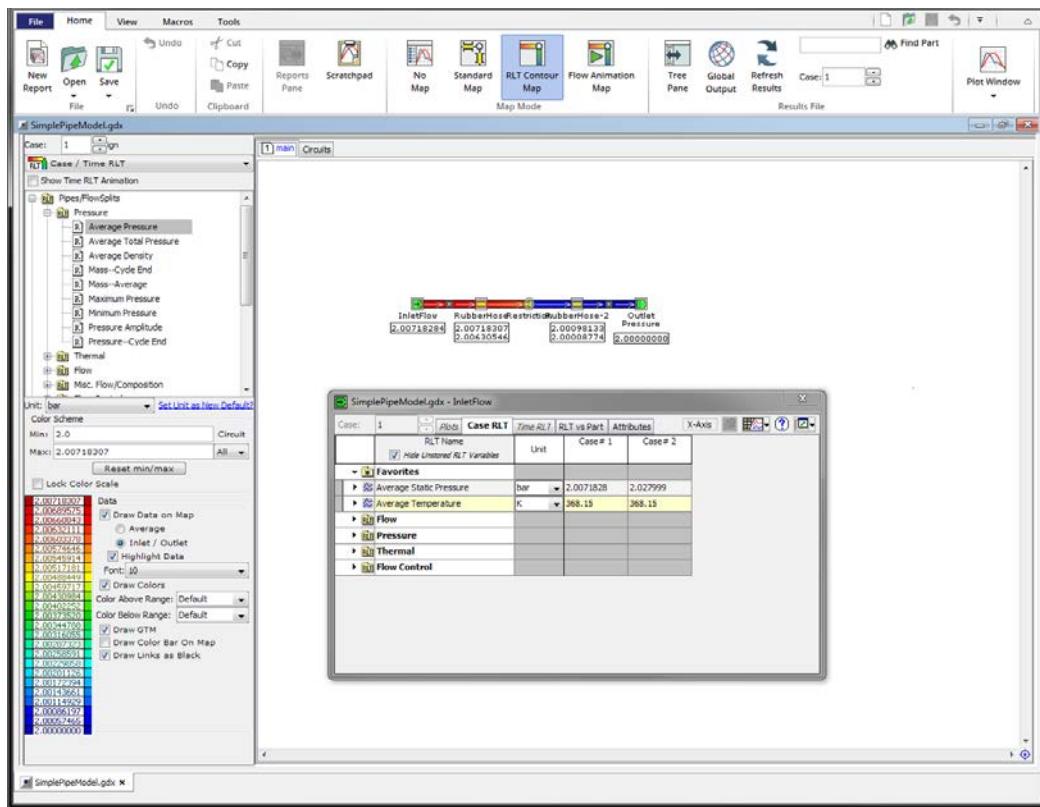
To view the results of the simulation, press the **View Results** button to launch GT-POST. Choose the filename of the .gdx file with the same file name as the model. The GT-POST post-processing program will launch automatically in a new window. When launching the results for the first time they will open in a map mode similar to GT-ISE. The GT-POST manual (File→Manuals→Graphical_Applications) can be used as a reference when viewing results.



Tutorial 2: Modeling a Simple Open Flow Circuit

Most results for cooling system simulations are case end (or end of run) results. It is possible to look at time based results, but this will be covered in CHAPTER 7: Modeling a Transient Thermal Model when a transient simulation is modeled.

Go to the RLT Contour mode by selecting the button (containing a color bar icon) in the toolbar to view the case end results of the pressure drop in the circuit. In the tree on the left, expand the **Pressure** option and highlight (select) the Average Pressure value. This will color the map to show the pressure in the flow circuit. To view the pressure results in each case one can use the case spinner in the toolbar to select a different case, or alternatively double-click on a part in the map to open a window that summarizes the results for that specific part.



This mode is useful when trying to compare the results between different parts in the model, or with measured results. In this case, one can see that the pressure drop in the flow circuit almost doubles when the flow rate doubles, with 75% of the pressure drop being attributed to the area restriction that was defined in the orifice. In cooling system modeling, the significant sources of pressure drop are typically caused by components, area changes, and friction, in that order. Knowing the expected pressure drop can assist in debugging a model if unexpected results are seen. If the pressure drop does not match in flow components, then the flow rate will not be correct. If the flow rate is not correct, then the heat balance in the system will not match results. CHAPTER 4: Modeling a Hydraulic System (No Heat Transfer) will cover modeling a hydraulic circuit for a complete system.

CHAPTER 3: Modeling Components of a Coolant Circuit

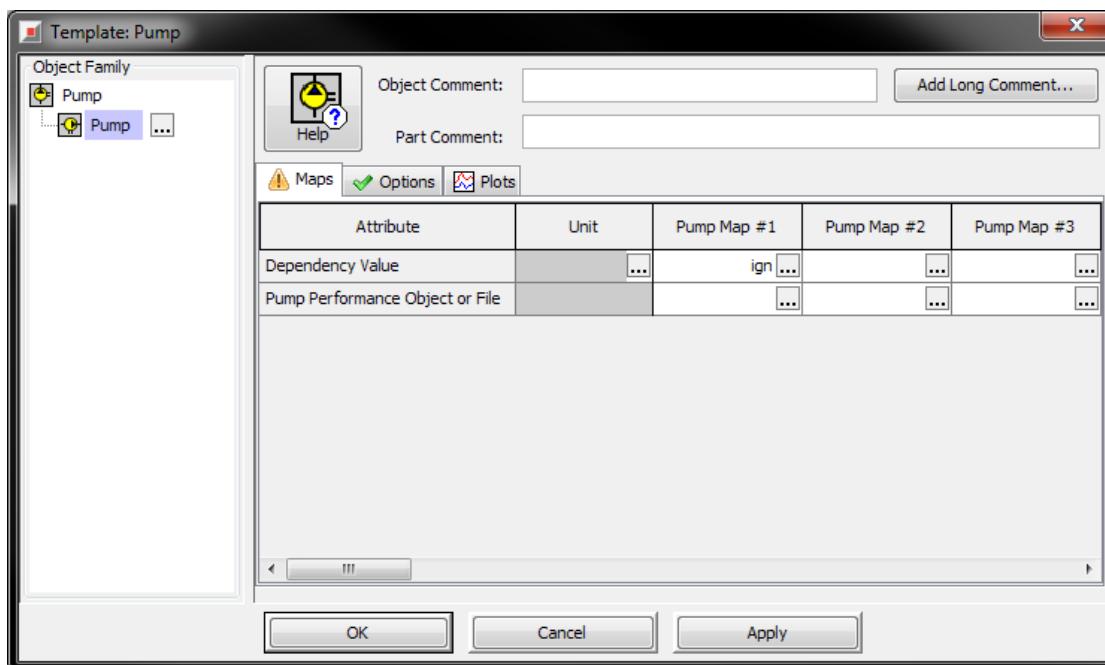
This tutorial has been prepared to provide guidelines on how to build a component of a coolant circuit and best practices. When building any cooling system, it is always recommended to build the components of the coolant circuit (i.e. heat exchanger) in a standalone model to ensure that the data entered was done so correctly, and the component is behaving as expected. Additionally, standalone components are recommended to be tested with the neighboring flow volumes as adiabatic and no friction pressure loss so that they do not influence the results of the test.

3.1 Modeling a Pump

Open the model Pump.gtm that can be found in the directory ..\tutorials\Modeling_Applications\Cooling_Thermal_Management\03-Components\. This tutorial is partially set up to test a Pump component standalone. The missing information to test a pump can be acquired by measuring the pump performance on a test bench. The pump performance typically contains a static pressure rise versus flow rate at various constant speed lines, as well as the efficiency at each operating point. Also required are the reference fluid conditions such as the pressure, temperature, and fluid type, and the inlet and outlet pipe diameter.

3.1.1 Building a Pump

Start by opening the part called 'Pump' found in the project map. The 'Pump' template allows various maps to be entered in each column of the **Maps** folder. Each map (or column) can be dependent on some variable that changes such that the solver can interpolate between each map provided. The variable that is used for the interpolation is defined in the **Options** folder. It is common for cooling system models to only have a single map. However, for the case when the pump performance is dependent on fluid temperature, this dependency is included in the map directly (more on this later).

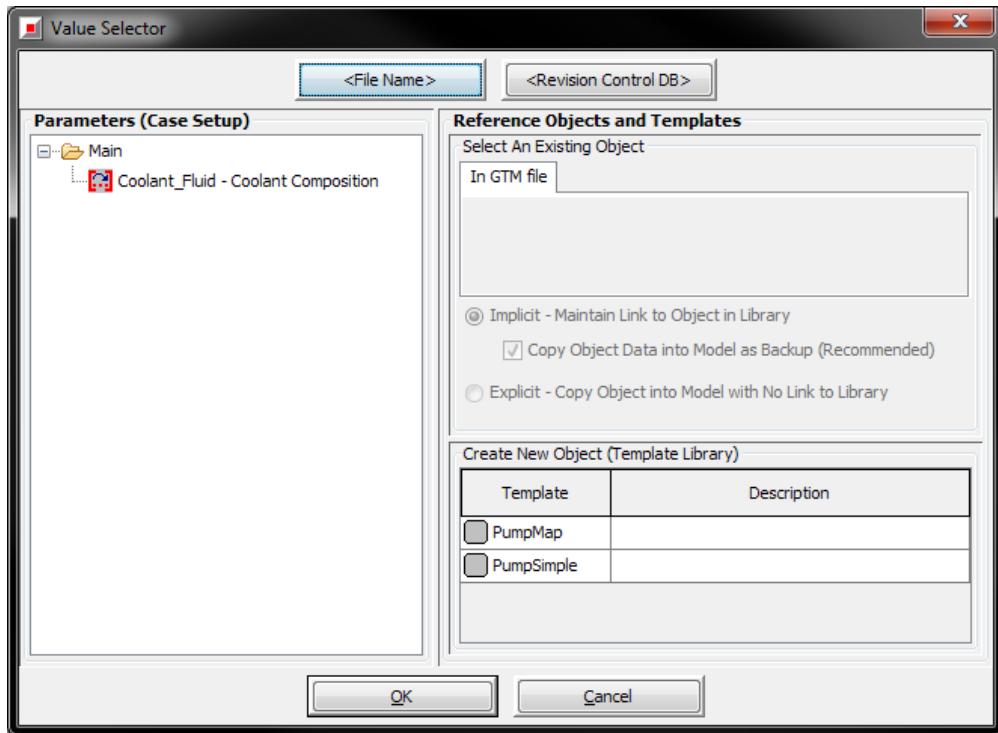


Click on the Value Selector for the attribute Pump Performance Object or File for column 1 in the **Maps** folder. The Value Selector will give the option to create a new object using 'PumpMap' or 'PumpSimple'.



Tutorial 3: Modeling Components of a Coolant Circuit

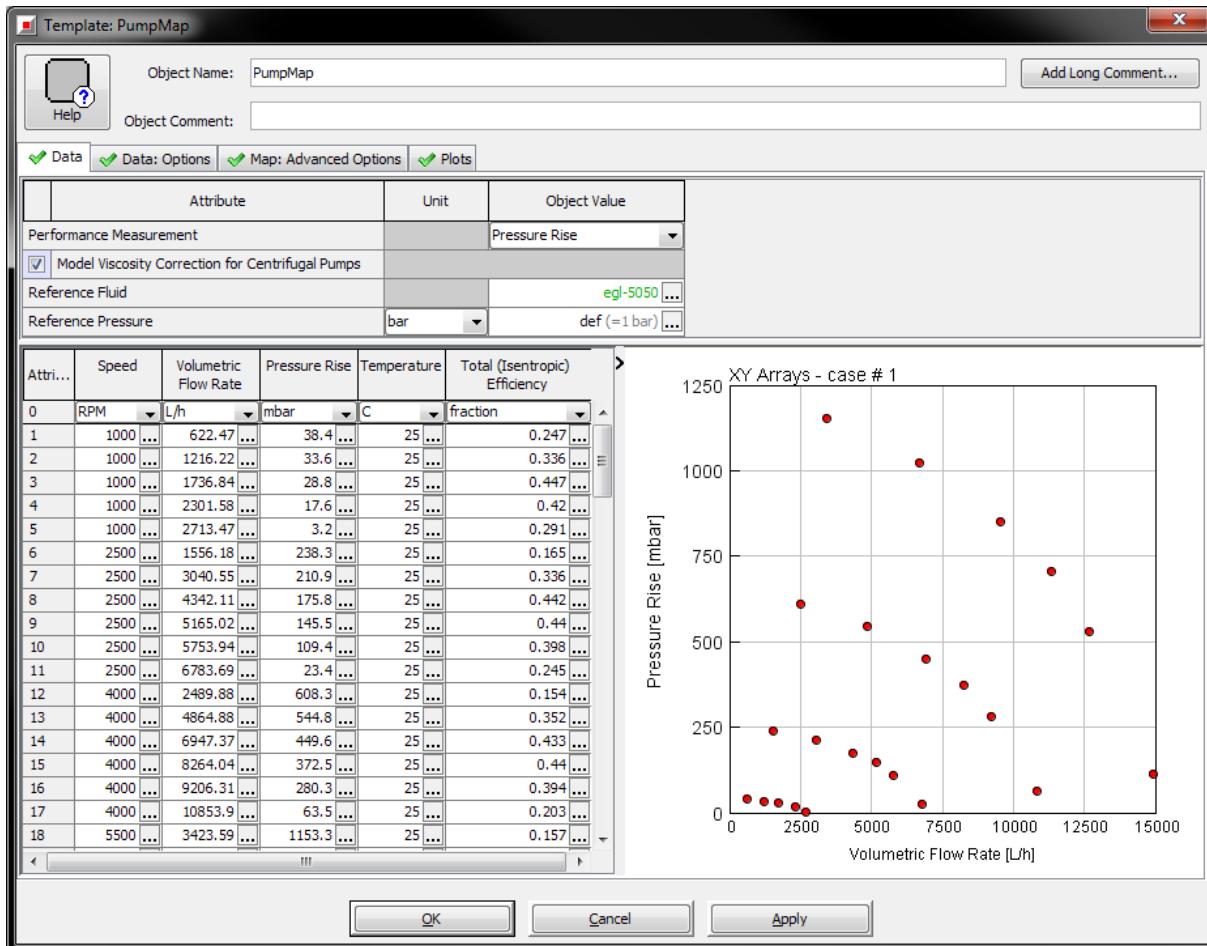
It is always recommended to use the template 'PumpMap' when the measured performance data is available. The template 'PumpSimple' is only useful when a general performance trend is known and the performance data is not readily available.



Select the 'PumpMap' object to open the data input dialog, and name the object 'PumpMap'. The pump performance data that will be copied over can be found in the file *DataSheets.xlsx* in the directory ..\tutorials\Modeling_Applications\Cooling_Thermal_Management\. The folder 'Pump' contains the measured performance data for the coolant pump. The pressure rise was measured by taking the static pressure difference across the pump, while the flow rate through the pump was recorded along with the operating speed. The pump efficiency is calculated as the ratio of work required to move the fluid from state 1 (inlet) to state 2 (outlet) against the power consumed by the pump. The excess energy will be added to the fluid at the outlet of the pump and will be seen in the form of a temperature increase. Copy and paste the data to the 'PumpMap' object, and select the correct fluid (egl-5050) from the library. *Warning: Be careful to select the correct units prior to pasting the data into the object.*



Tutorial 3: Modeling Components of a Coolant Circuit



Note: If data was measured at different operating temperatures, the Temperature column in the 'PumpMap' template can be used to group the pump performance curves together. The solver will then use the different temperature groups to calculate the correct pump performance. Usually only one fluid temperature is used to measure the pump performance unless the fluid viscosity plays a large role. If measured data is not available to take into account the effects of fluid viscosity, then the attribute **Model Viscosity Correction for Centrifugal Pumps** can be enabled to predict the changes in pump performance.

It is also possible to enter performance data for a reverse flow condition (negative flow rate through a pump), and for an overstreamed condition (inlet pressure is greater than outlet pressure). This input data is not commonly measured for a pump, but it is highly recommended doing so if there are multiple pumps existing in the same flow circuit. This is because there is a possibility that one pump will force the other to go into one of these regions. If this happens, and data is not provided, then the flow solution may not be accurate at these operating conditions.

Click OK on the 'PumpMap' object when finished.

If there are multiple pumps in the same flow circuit, then it is highly recommended to enable the Conservative pump model option in the **Options** folder of the 'Pump' template. For this tutorial, the Standard option is sufficient. Click OK on the 'Pump' object to accept the input.

3.1.2 Testing the Behavior of a Pump

Prior to running the model, the test conditions for the pump need to be included. As mentioned before, this model is partially set up to test these conditions. Parameters have already been created for all necessary inputs to test the pump performance. This includes the boundary conditions and reference geometry. Go to Case Setup to set values for the parameters. The input for the reference conditions can be found in the *DataSheets.xlsx* file. Use the Value Selector to import the correct fluid type. The parameter for [Coolant_Pressure] is set up as an equation to use the value of the existing parameter [Inlet_Pressure]. Mathematical equations are supported in Case Setup, and the expression must start with an =. The parameter [Outlet_Pressure] is set up as a math operation so that it is automatically calculated from the sum of [Inlet_Pressure] and [Pressure_Rise].

Main	All		
Parameter	Unit	Description	Case 1
Case On/Off		Check Box to Turn Case On	<input checked="" type="checkbox"/>
Case Label		Unique Text for Plot Legends	
Pipe_Diameter	mm	Pump Inlet/Outlet Pipe Diameter	35 <input type="button" value="..."/>
Pump_Speed	RPM	Pump Speed	<input type="button" value="..."/>
Inlet_Pressure	bar	Inlet Pump Pressure (static)	1 <input type="button" value="..."/>
Pressure_Rise	bar	Pressure Rise	
Outlet_Pressure	bar	Outlet Pump Pressure	EMPTY CELL <input type="button" value="..."/>
Coolant_Pressure	bar	Coolant Pressure	1 <input type="button" value="..."/>
Coolant_Temperature	C	Coolant Temperature	25 <input type="button" value="..."/>
Coolant_Fluid		Coolant Composition	egl-5050 <input type="button" value="..."/>

Next is to create the test conditions for the pump to ensure it is operating correctly. Only a few test points will be included, but feel free to test more. Append two more cases (a total of 3 cases), and set the parameter [Pump_Speed] to 4000 RPM, and [Pressure_Rise] to 0.5, 0.3, and 0.2 bar, respectively. The operating points were selected based on the performance data that was entered, and may vary from one performance map to another. Click OK on Case Setup when the operating points have been entered.

Main	All				
Parameter	Unit	Description	Case 1	Case 2	Case 3
Case On/Off		Check Box to Turn Case On	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Case Label		Unique Text for Plot Legends			
Pipe_Diameter	mm	Pump Inlet/Outlet Pipe Diameter	35 <input type="button" value="..."/>	35 <input type="button" value="..."/>	35 <input type="button" value="..."/>
Pump_Speed	RPM	Pump Speed	4000 <input type="button" value="..."/>	4000 <input type="button" value="..."/>	4000 <input type="button" value="..."/>
Inlet_Pressure	bar	Inlet Pump Pressure (static)	1 <input type="button" value="..."/>	1 <input type="button" value="..."/>	1 <input type="button" value="..."/>
Pressure_Rise	bar	Pressure Rise	0.5	0.3	0.2
Outlet_Pressure	bar	Outlet Pump Pressure	1.5 <input type="button" value="..."/>	1.3 <input type="button" value="..."/>	1.2 <input type="button" value="..."/>
Coolant_Pressure	bar	Coolant Pressure	1 <input type="button" value="..."/>	1 <input type="button" value="..."/>	1 <input type="button" value="..."/>
Coolant_Temperature	C	Coolant Temperature	25 <input type="button" value="..."/>	25 <input type="button" value="..."/>	25 <input type="button" value="..."/>
Coolant_Fluid		Coolant Composition	egl-5050 <input type="button" value="..."/>	egl-5050 <input type="button" value="..."/>	egl-5050 <input type="button" value="..."/>

Run the model at this time. Run Setup has already been filled out following GT recommendations for testing standalone components. After the model has finished running, open the results in GT-POST.

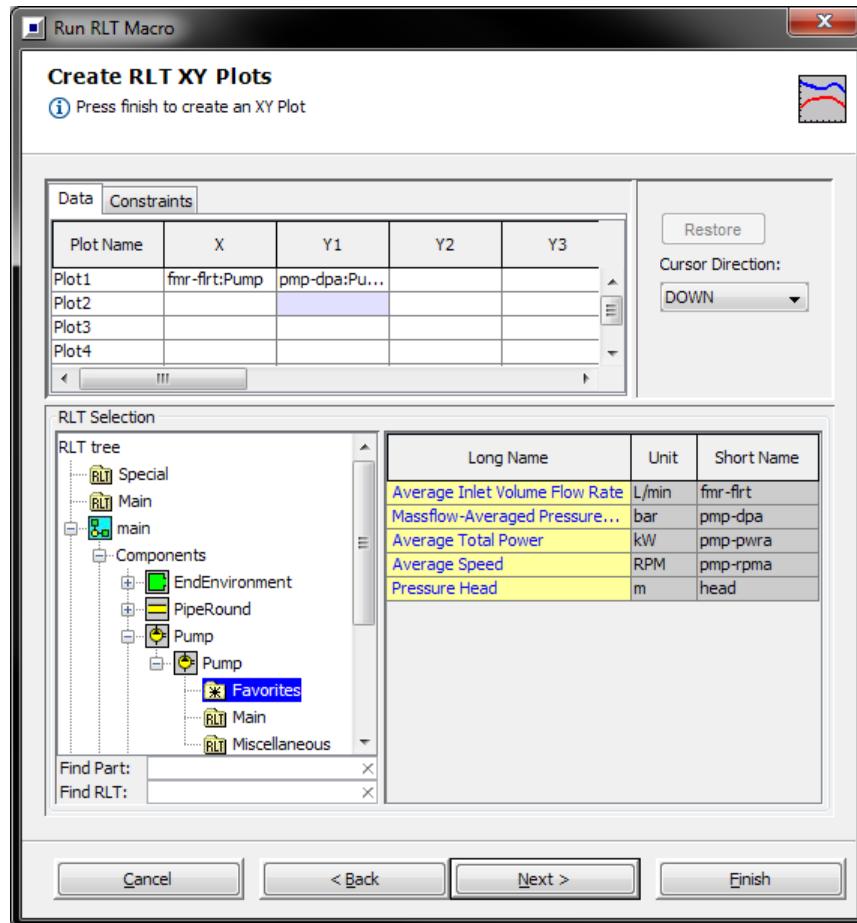
A preprocessed plot of the performance map data is available in the GT-POST tree. The plot is called **Pressure Rise vs. Volumetric Flow Rate**, and can be found under the PumpMap:PumpMap heading. This plot shows the raw data as well as the map fit created. To see how the results of the simulation stack



Tutorial 3: Modeling Components of a Coolant Circuit

up to the original data, one can either view the RLTs of the pump and match the operating points to the performance data, or merge RLTs with the preprocessed plot.

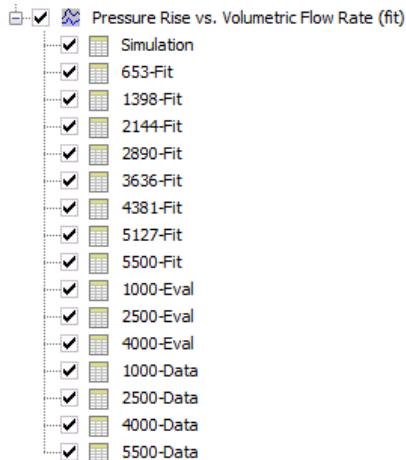
Select the RLT macro from the toolbar () to merge the RLTs with the performance data. Select the XY Scatter option, and click "Next" on the dialog that opens. In the next screen, locate the 'Pump' part by expanding the tree view, and select Average Inlet Volumetric Flow Rate for the X variable, and Massflow-Averaged Pressure Rise for the Y1 variable of Plot 1.



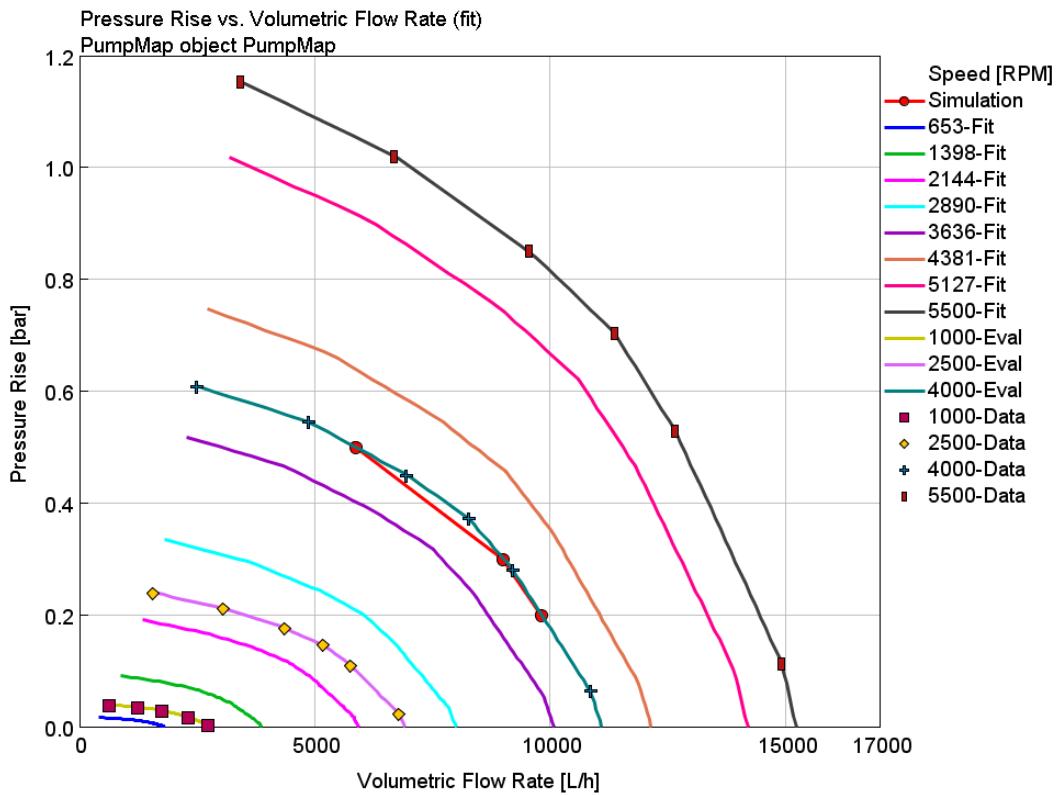
Click Finish after the selection is made. At this point the scratchpad will be opened with the newly created plot. Next, drag the preprocessed plot of the pump performance map from the .gdx file to the scratchpad. Then, drag the data set of the manually created Pressure Rise vs. Volume Flow Rate plot into the preprocessed plot that is now in the .gu file. The screenshot below shows what the final merge of the data sets will look like. Please note that the manually created data set was renamed to Simulation.



Tutorial 3: Modeling Components of a Coolant Circuit



If the plot is plotted (right-click and select View, or F4), then the operating points for the Simulation data set will appear along with the original performance data. (Please note that the Simulation data set was modified so symbols are displayed in order to make it easier to view the image below. If this is desired, then double-click on the Simulation data set and go to the Display folder.)



If the pump was successfully built, then the Simulation data set should match up with the speed line that was selected for testing. If not, then the test conditions and input data should be reviewed to ensure no mistakes were entered.

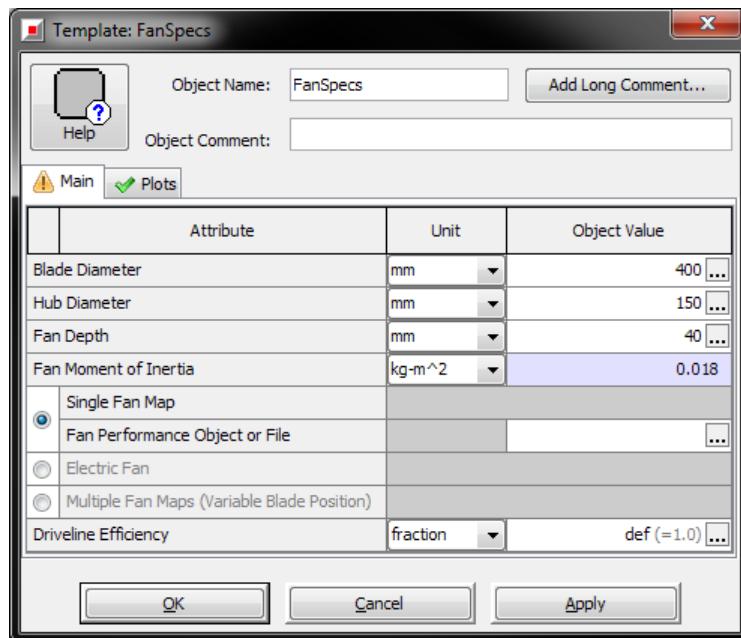


3.2 Modeling a Fan

Open the model `Fan.gtm` that can be found in the directory `..\tutorials\Modeling_Applications\Cooling_Thermal_Management\03-Components`. This tutorial is partially set up to test a Fan component standalone. The missing information to test a fan can be acquired by measuring the fan performance on a test bench. The fan performance typically contains a static pressure rise versus flow rate at various constant speed lines, as well as the efficiency at each operating point. Also required are the reference fluid conditions such as the pressure and temperature. The testing of a fan component is the same as testing a standalone pump component.

3.2.1 Building a Fan

Start by opening the part called 'Fan' found in the project map. The 'Fan' template has two functions: "As Tested" geometry and performance, and "As Used" (or scaled) geometry predictions. Using the "As Tested" input the performance of the fan can be predicted if the blade diameter is changed (scaled). For most applications though, the original geometry and performance data is enough to model the fan. Use the Value Selector for the attribute "As Tested" Fan Specifications Object. The **Geometry** folder contains the input for the fan geometry. The geometry (and performance data) for the fan can be found in the file `DataSheets.xlsx` in the directory `..\tutorials\Modeling_Applications\Cooling_Thermal_Management`.



There are three difference options to enter fan data:

- Single fan performance map (i.e. geared or imposed speed)
- Electric fan
- Variable blade position fan map (i.e. geared or imposed speed)

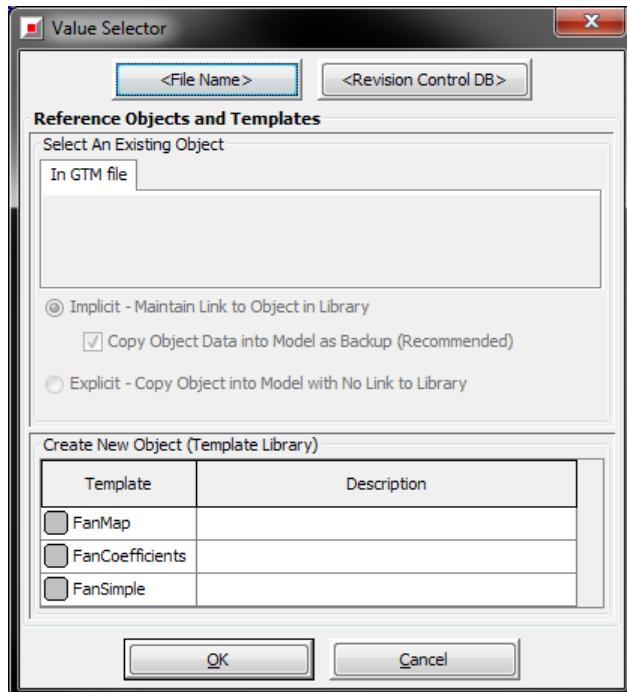
With the Single Fan Map option selected, use the Value Selector for the Fan Performance Object or File. Three options are available: 'FanCoefficients', 'FanMap', and 'FanSimple'.

'FanMap' is the recommended template to use because it allows data to be entered as it will be measured from a flow bench (pressure rise, flow rate, and speed). 'FanCoefficients' is another acceptable form, but



Tutorial 3: Modeling Components of a Coolant Circuit

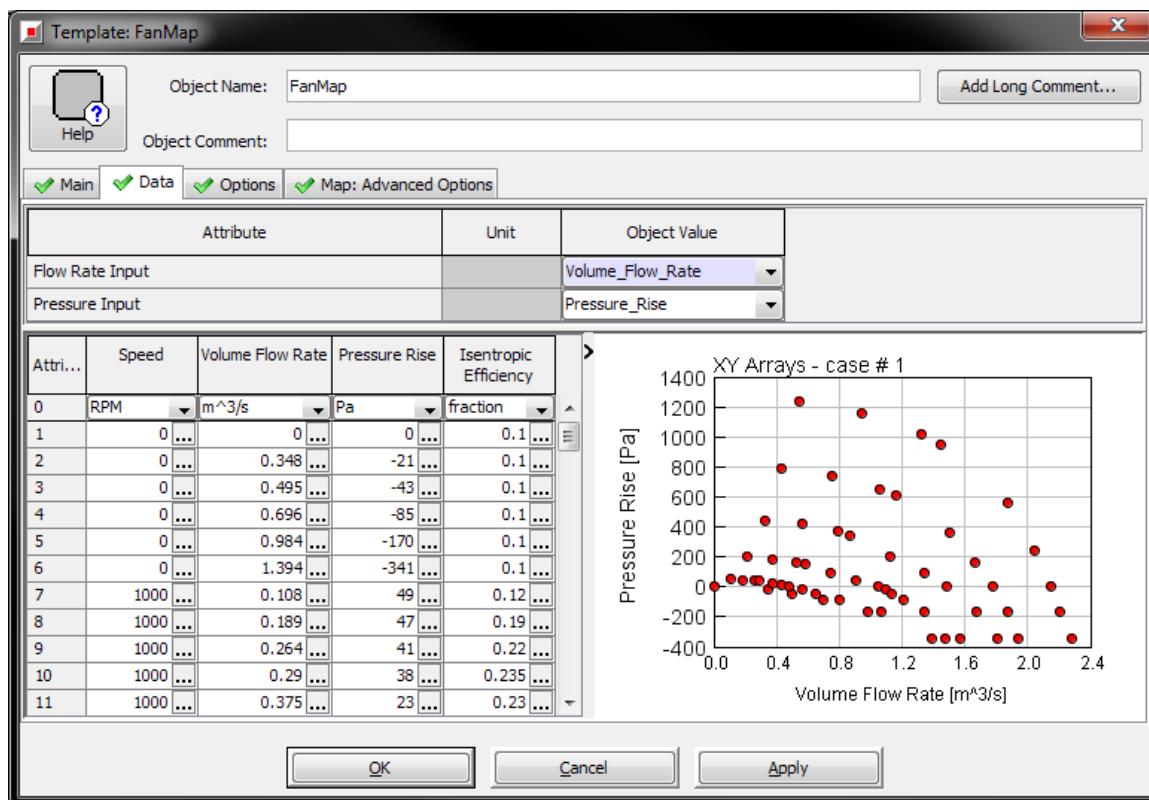
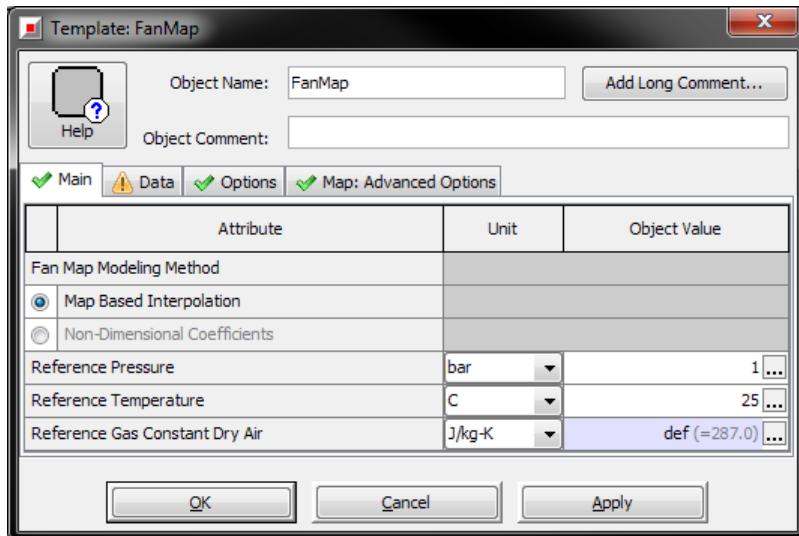
it requires the measured performance data to be manually converted to the necessary input ('FanMap' has the ability to do the conversion automatically if the option is selected). 'FanSimple' is only useful when a general performance trend is known and the performance data is not readily available.



After selecting 'FanMap', copy the reference conditions and performance data for the fan from *DataSheets.xlsx*. The folder 'Fan' contains the measured performance data for the fan. The reference conditions are necessary so the solver can correct the inlet conditions of the fan to use the map lookup. If the fan performance data was measured at different reference conditions, then they must all be corrected to the same reference condition before entering the data in the **Data** folder. The pressure rise was measured by taking the average static pressure difference across the fan, while the flow rate through the fan was recorded along with the operating speed. The fan efficiency is calculated as the ratio of work required to move the fluid from state 1 (inlet) to state 2 (outlet) against the power consumed by the fan. The excess energy will be added to the fluid at the outlet of the fan and will be seen in the form of a temperature increase. *Warning: Be careful to select the correct units prior to pasting the data into the object.*



Tutorial 3: Modeling Components of a Coolant Circuit



Note: Data has been entered for a negative pressure rise across the fan to represent an overblown condition. If data is not entered in this region, and the fan attempts to operate with a negative pressure rise, then the flow rate calculated will be the flow rate at a pressure rise of zero for the given speed.

Click OK when finished for both 'FanMap' and 'FanSpecs'.

Click OK on the 'Fan' template to complete the fan input.



3.2.2 Testing the Behavior of a Fan

Prior to running the model, the test conditions for the fan need to be included. As mentioned before, this model is partially set up to test these conditions. Parameters have already been created for all necessary inputs to test the fan performance. Go to Case Setup to set values for the parameters. The input for the reference conditions can be found in the *DataSheets.xlsx* file. The parameter for [Ambient_Pressure] is set up as an equation to use the value of the existing parameter [Inlet_Pressure]. Mathematical equations are supported in Case Setup, and the expression must start with an =. The parameter [Outlet_Pressure] is set up as a math operation so that it is automatically calculated from the sum of [Inlet_Pressure] and [Pressure_Rise].

Main	All		
Parameter	Unit	Description	Case 1
Case On/Off		Check Box to Turn Case On	<input checked="" type="checkbox"/>
Case Label		Unique Text for Plot Legends	
Fan_Speed	RPM	Fan Speed	<input type="button" value="..."/>
Inlet_Pressure	bar	Inlet Pressure	1 <input type="button" value="..."/>
Pressure_Rise	Pa	Pressure Rise	
Outlet_Pressure	bar	Outlet Pressure	EMPTY CELL <input type="button" value="..."/>
Ambient_Pressure	bar	Ambient Pressure	1 <input type="button" value="..."/>
Ambient_Temperature	C	Ambient Temperature	25 <input type="button" value="..."/>

Next is to create the test conditions for the fan to ensure it is operating correctly. Only a few test points will be included, but feel free to test more. Append two more cases (a total of 3 cases), and set the parameter [Fan_Speed] to 3000 RPM, and [Pressure_Rise] to 400, 300, and 200 Pa, respectively. The operating points were selected based on the performance data that was entered, and may vary from one performance map to another. Click OK on Case Setup when the operating points have been entered.

Main	All				
Parameter	Unit	Description	Case 1	Case 2	Case 3
Case On/Off		Check Box to Turn Case On	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Case Label		Unique Text for Plot Legends			
Fan_Speed	RPM	Fan Speed	3000 <input type="button" value="..."/>	3000 <input type="button" value="..."/>	3000 <input type="button" value="..."/>
Inlet_Pressure	bar	Inlet Pressure	1 <input type="button" value="..."/>	1 <input type="button" value="..."/>	1 <input type="button" value="..."/>
Pressure_Rise	Pa	Pressure Rise	400	300	200
Outlet_Pressure	bar	Outlet Pressure	1.004 <input type="button" value="..."/>	1.003 <input type="button" value="..."/>	1.002 <input type="button" value="..."/>
Ambient_Pressure	bar	Ambient Pressure	1 <input type="button" value="..."/>	1 <input type="button" value="..."/>	1 <input type="button" value="..."/>
Ambient_Temperature	C	Ambient Temperature	25 <input type="button" value="..."/>	25 <input type="button" value="..."/>	25 <input type="button" value="..."/>

Run the model at this time. Run Setup has already been filled out following GT recommendations for testing standalone components. After the model has finished running, open the results in GT-POST.

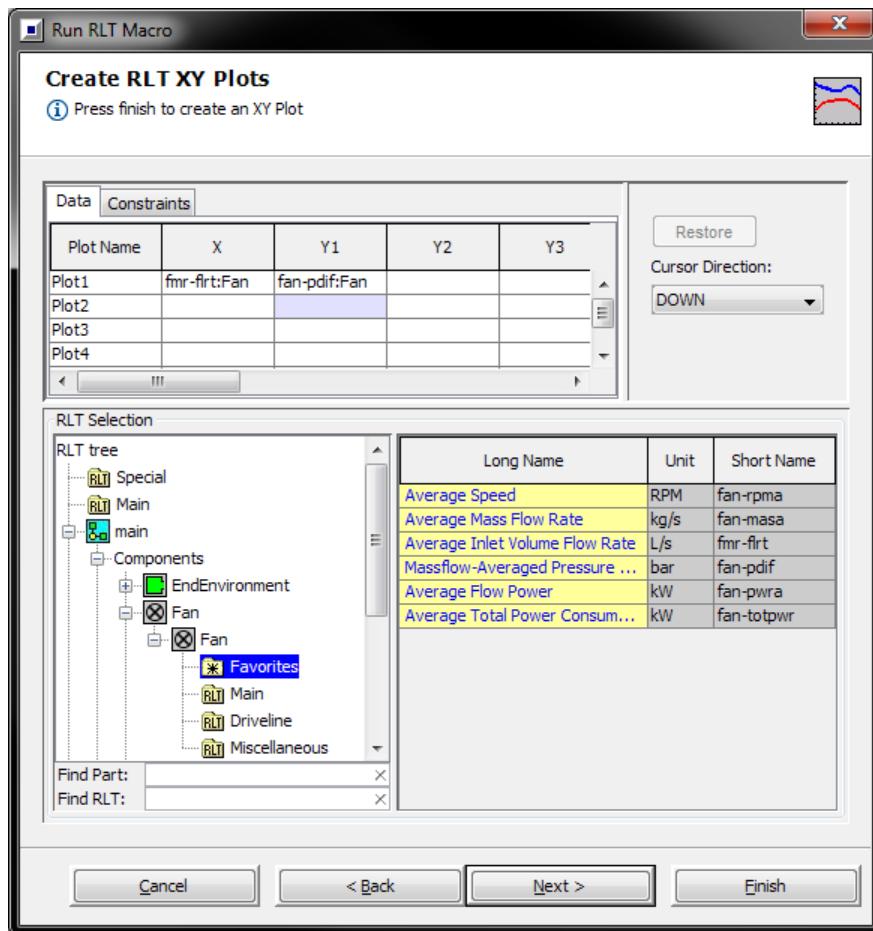
A preprocessed plot of the performance map data is available in the GT-POST tree. The plot is called **Pressure vs. Mass Flow Rate**, and can be found under the FanSpecs:FanMap heading. This plot shows the raw data as well as the map fit created. To see how the results of the simulation stack up to the original data, one can either view the RLTs of the fan and match the operating points to the performance data, or merge RLTs with the preprocessed plot.

Select the RLT macro from the toolbar () to merge the RLTs with the performance data. Select the XY Scatter option, and click "Next" on the dialog that opens. In the next screen, locate the 'Fan' part by



Tutorial 3: Modeling Components of a Coolant Circuit

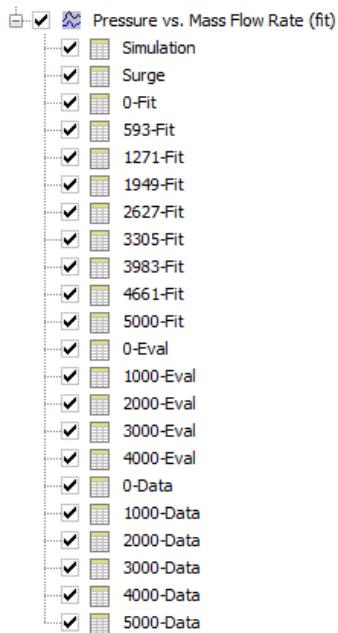
expanding the tree view, and select Average Mass Flow Rate for the X variable, and Massflow-Averaged Pressure Rise for the Y1 variable of Plot 1.



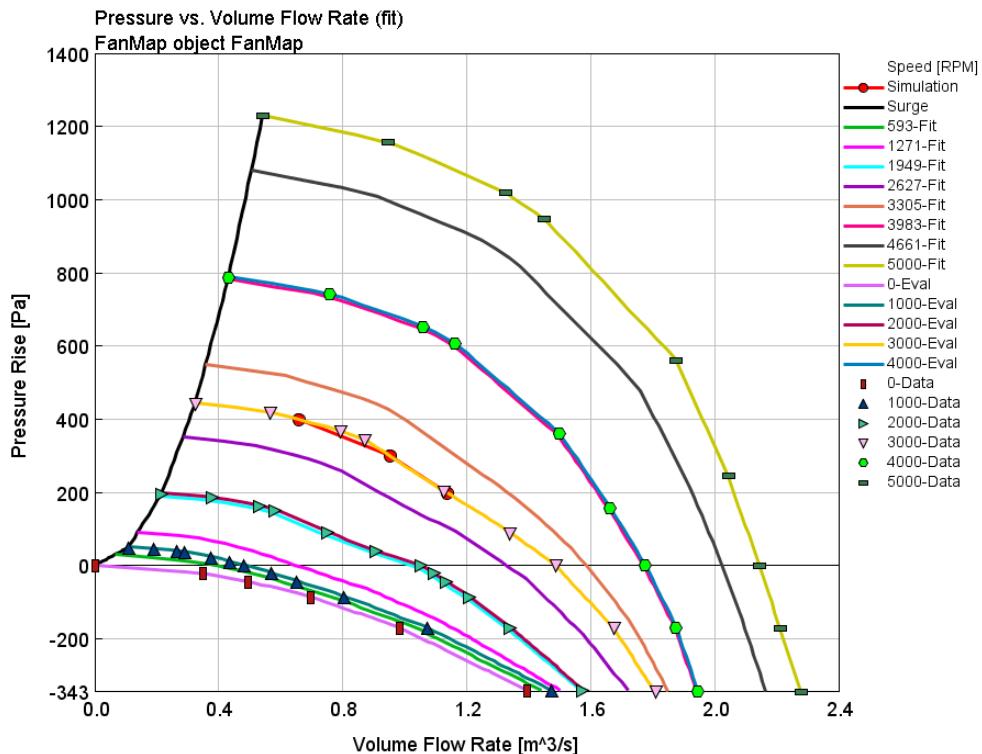
Click Finish after the selection is made. At this point a new file (.gu) will be opened with the newly created plot. Next, drag the preprocessed plot of the fan performance map from the .gdx file to the .gu file. Then, drag the data set of the manually created Pressure Rise vs. Mass Flow Rate plot into the preprocessed plot that is now in the .gu file. The screenshot below shows what the final merge of the data sets will look like. Please note that the manually created data set was renamed to Simulation.



Tutorial 3: Modeling Components of a Coolant Circuit



If the plot is plotted (right-click and select View, or F4), then the operating points for the Simulation data set will appear along with the original performance data. (Please note that the Simulation data set was modified so symbols are displayed in order to make it easier to view the image below. If this is desired, then double-click on the Simulation data set and go to the Display folder.)



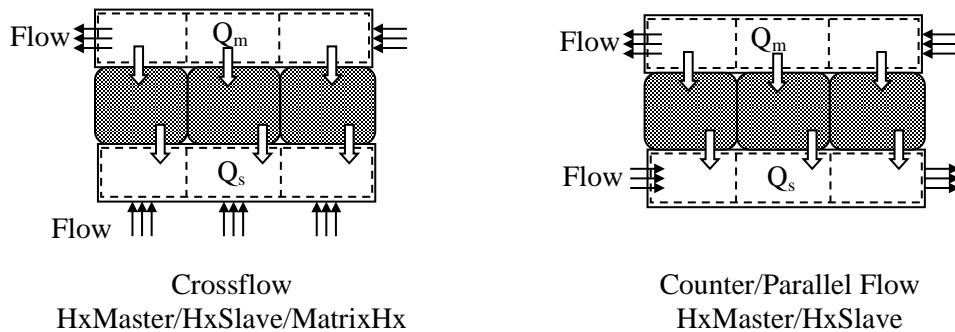
If the fan was successfully built, then the Simulation data set should match up with the speed line that was selected for testing. If not, then the test conditions and input data should be reviewed to ensure no mistakes were entered.

3.3 Modeling a Heat Exchanger

3.3.1 Theory of Heat Exchanger Modeling

Heat exchange between two fluid circuits can be modeled using the 'HxMaster' and 'HxSlave' pair, or the 'MatrixHx' object. The 'MatrixHx' object is exclusively used for 3D underhood flow, and can only be created as an output of a COOL3D model (.ghx) where 'COOLHeatExchanger' is used.

Lumped heat exchanger modeling can be used if the effects of high frequency pressure waves, temperature distribution, or two-phase flow are not of interest. However, it is highly recommended that all heat exchangers be discretized to a minimum of 3 subvolumes (ideally 5 subvolumes) so a more accurate wall temperature distribution can be calculated when solving for the heat transfer rate between two fluids. This is especially important when dealing with a parallel or counter flow heat exchanger type because of the large wall temperature variations that typically exist in these heat exchangers. If a heat exchanger is not discretized, then it is entirely possible that the incorrect wall temperature will be calculated, which may over or under estimate the heat transfer rate between the two opposing fluids. The methods of heat exchanger discretization can be seen below.



The heat transfer calculation includes the effects of the wall thermal capacitance, and the conductivity of the material (if modeled). The temperature of the structure in a heat exchanger is calculated from a balance of the heat transfer rates between the structure and the two fluids using the following equation:

$$\frac{dT_{wall}}{dt} = \frac{Q_m + Q_s}{\rho V C_p} = \frac{\left(hA\Delta T - \frac{2kA\Delta T_w}{t} \right)_m + \left(hA\Delta T - \frac{2kA\Delta T_w}{t} \right)_s}{\rho V C_p}$$

where:

h	heat transfer coefficient
A	heat transfer area
ΔT	temperature difference between the fluid and the wall
k	thermal conductivity of the wall material
ΔT_w	temperature difference between the surface and average wall temperature
t	tube thickness
ρ	density of the wall material



V	volume of the wall material
C_p	heat capacity of the wall material

The temperature difference above is dependent on the heat exchanger configuration. For example, parallel flow and counterflow heat exchangers use a modified form of the log mean temperature difference (LMTD). Other configurations typically use a correction to the LMTD. GT-SUITE has the capability to model parallel flow, counterflow, and crossflow (single-pass, unmixed) heat exchangers.

The heat transfer rate from each fluid to the wall is calculated using heat transfer coefficients defined by separate Nusselt number correlations of the form:

$$Nu = C Re^m Pr^{1/3}$$

where

$$Nu = \left(\frac{hL}{k} \right) \quad Re = \left(\frac{\rho UL}{\mu} \right) \quad Pr = \left(\frac{\mu C_p}{k} \right)$$

L	reference length
k	thermal conductivity of the fluid
ρ	fluid density
C_p	heat capacity of the fluid
μ	dynamic viscosity of the fluid

The coefficients for the correlation needed to calculate the dimensionless numbers can be stored in the 'HxNusseltCorr' reference object, although this will be done automatically if a heat exchanger file (.hx) is used in the 'HeatExchangerSpecs' object. The coefficients for the correlation are extracted from experimental performance data with the 'HxNuMap' or 'HxNuMapRefrig' reference objects.

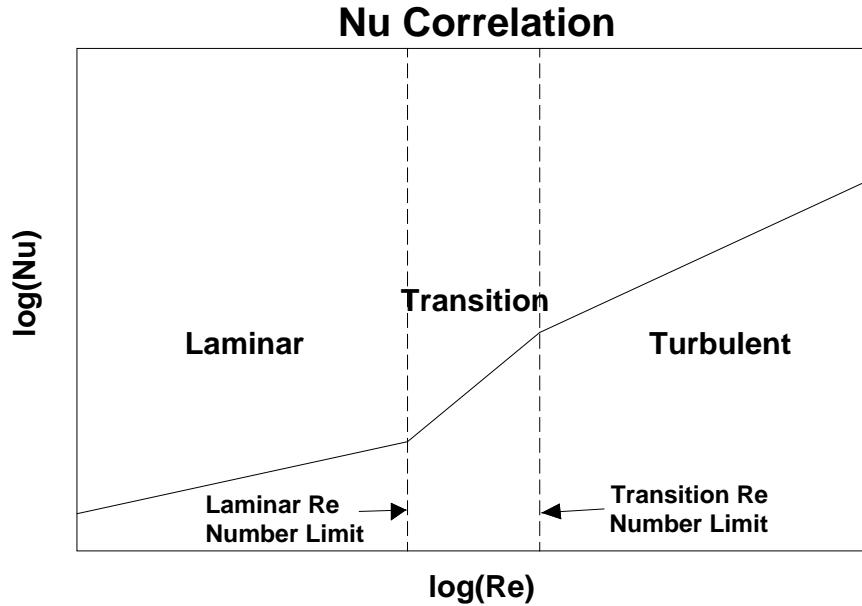
3.3.2 Creating a Correlation from Measured Performance Data

The heat transfer rates for a heat exchanger are calculated using heat transfer coefficients defined by Nusselt number correlations of the form:

$$Nu = C Re^m Pr^{1/3}$$

The 'HxNuMap' reference object can be used to extract the coefficient C and exponent m from steady-state experimental performance data. Different correlations (i.e. different coefficients and exponents) can be used for laminar, turbulent, and transitional flow regions as illustrated in the figure below. The coefficients and exponents of the correlation, as well as the Reynolds number defining the limits of the regions, are automatically determined by a regression analysis that best fits the data provided by minimizing the difference between the measured and predicted heat transfer rates. The regression analysis will always be performed at the start of a simulation unless a heat exchanger file (.hx) is created in the template 'HeatExchangerSpecs'. The quality of the fit can be viewed in GT-POST.





3.3.3 Building a Heat Exchanger

Open the model `HeatExchanger.gtm` that can be found in the directory `..\tutorials\Modeling_Applications\Cooling_Thermal_Management\03-Components`. This model is set up to test the performance of a standalone tube-fin heat exchanger (i.e. Radiator). The missing information such as the geometry and performance data can be acquired from a supplier, or by measuring the data on a test bench. There are always two sections of a heat exchanger modeled in GT-SUITE: the "master" side and the "slave" side. Each side represents a specific fluid, and exchange heat through the structure of the heat exchanger. The "master" side ('HxMaster') contains all of the information regarding the geometry and performance of the heat exchanger. The "slave" side ('HxSlave') exists on the map purely as a way to extract results for this specific fluid side.

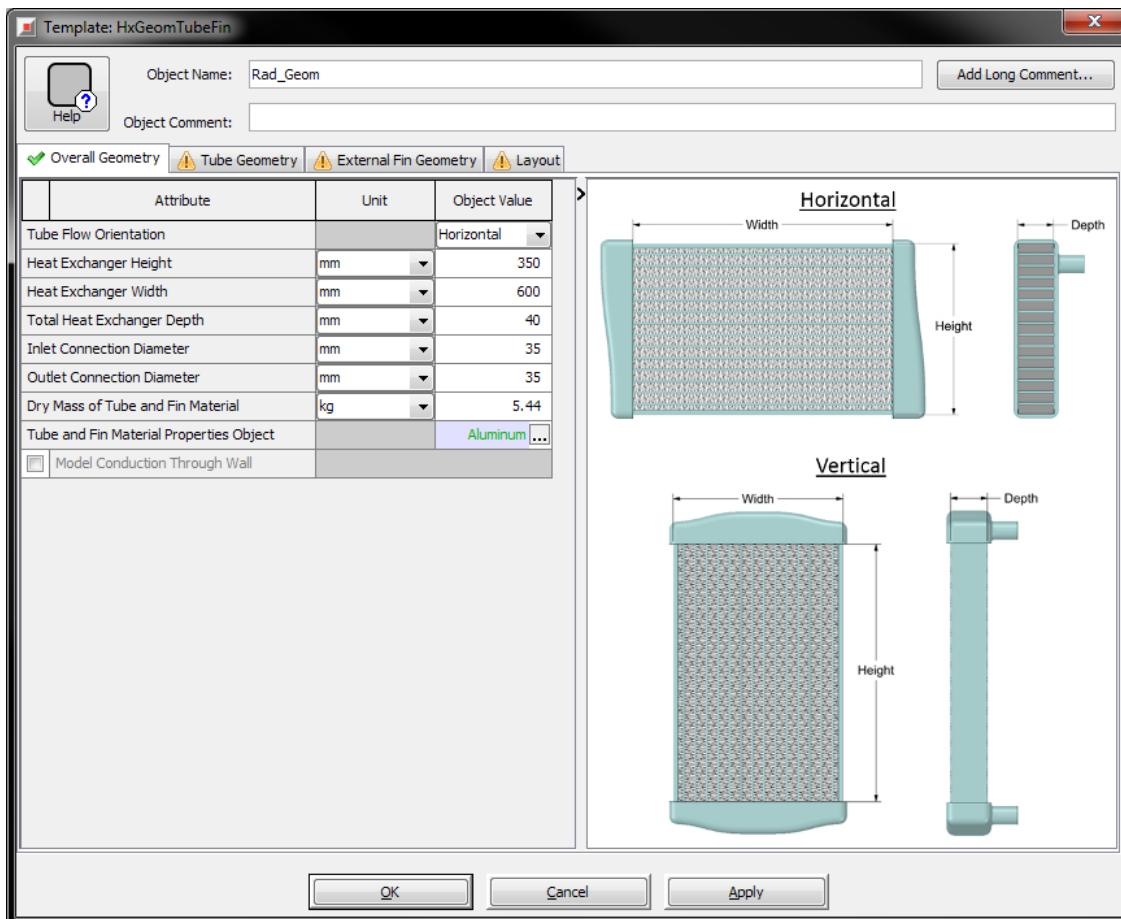
Open the part called 'Radiator' found in the project map. This part is the 'HxMaster' template, and the heat exchanger performance data is all contained in the attribute "As Tested" Heat Exchanger Specifications Object. To enter the geometry and performance data for the heat exchanger, use the Value Selector to create a 'HeatExchangerSpecs' object (provide a name for the 'HeatExchangerSpecs' object). The 'HeatExchangerSpecs' object is broken down into four attributes: geometry, heat transfer performance data, master pressure drop, and slave pressure drop. The attribute "As Tested" Heat Exchanger Geometry Object is where the geometry matching the performance data of your heat exchanger will be entered. "As Tested" is used to denote the original geometry, because the 'HxMaster' template has the ability to scale the geometry of the heat exchanger ("As Used") to predict the new performance of the heat exchanger. Use the Value Selector on the "As Tested" Heat Exchanger Geometry Object, and select the 'HxGeomTubeFin' object to create a tube-fin heat exchanger (the other heat exchanger options are plate, shell-tube, and general).

The **Overall Geometry** folder of 'HxGeomTubeFin' is where the overall dimensions of the heat exchanger are entered. This information can be measured directly, or acquired from a heat exchanger specification sheet sent by a supplier. The geometry for this heat exchanger can be found in the file `DataSheets.xlsx` in the directory `..\tutorials\Modeling_Applications\Cooling_Thermal_Management`. The Inlet and Outlet Connection Diameter is especially important because this diameter is used to determine



Tutorial 3: Modeling Components of a Coolant Circuit

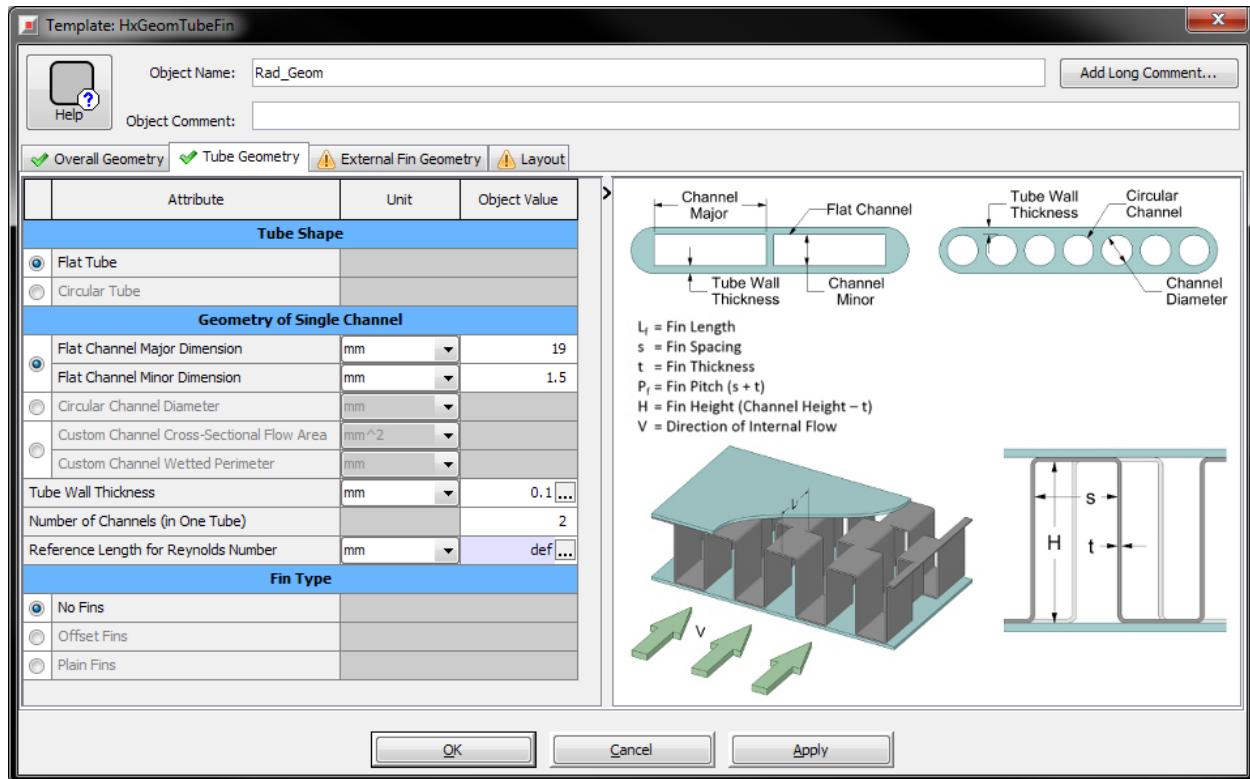
the velocity of the fluid entering the heat exchanger in the measured data when generating the Nusselt correlation and calibrating the pressure drop.



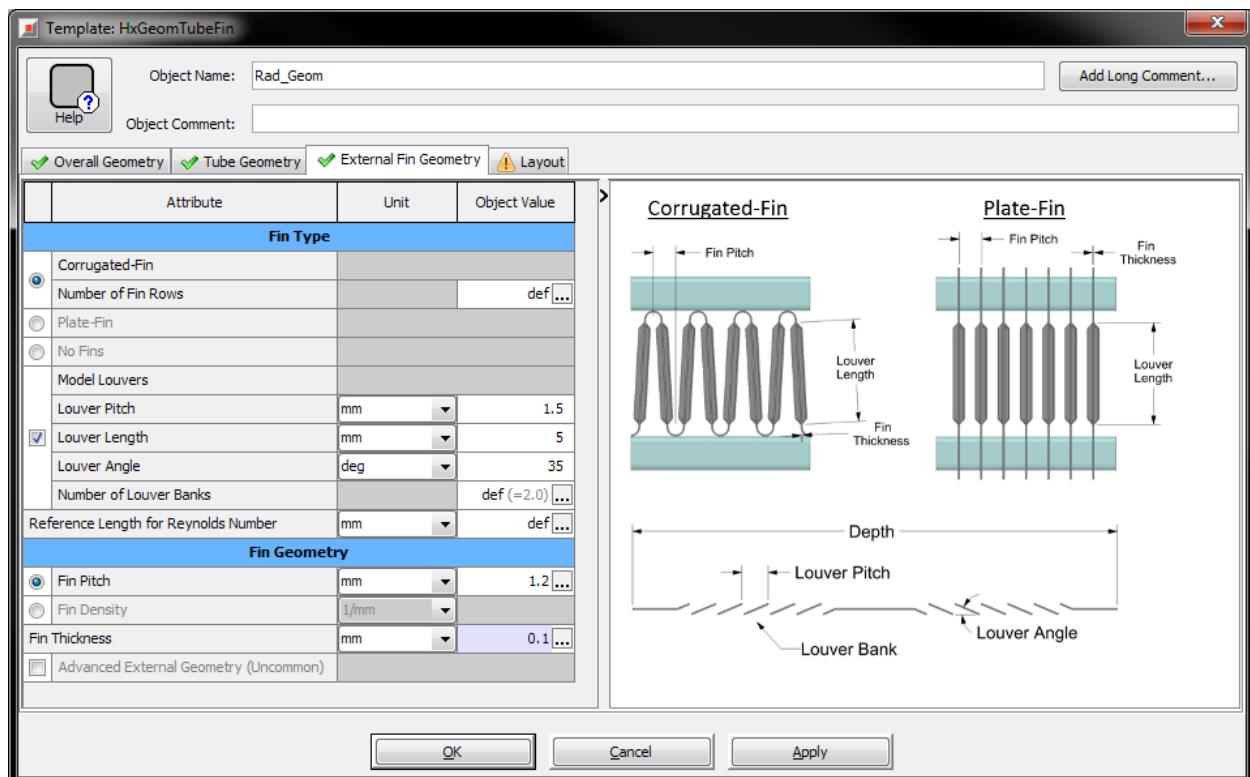
Use the **Tube Geometry** folder to define the internal tube geometry (including internal fins). Different tube shapes, as well as channel shapes, can be selected for use. The internal channel geometry for the heat exchanger tubes can be copied (or measured) from the *DataSheets.xlsx* file.



Tutorial 3: Modeling Components of a Coolant Circuit



The **External Fin Geometry** folder can be used to select the shape of the external fins. The external fin geometry for the heat exchanger can be copied (or measured) from the *DataSheets.xlsx* file.

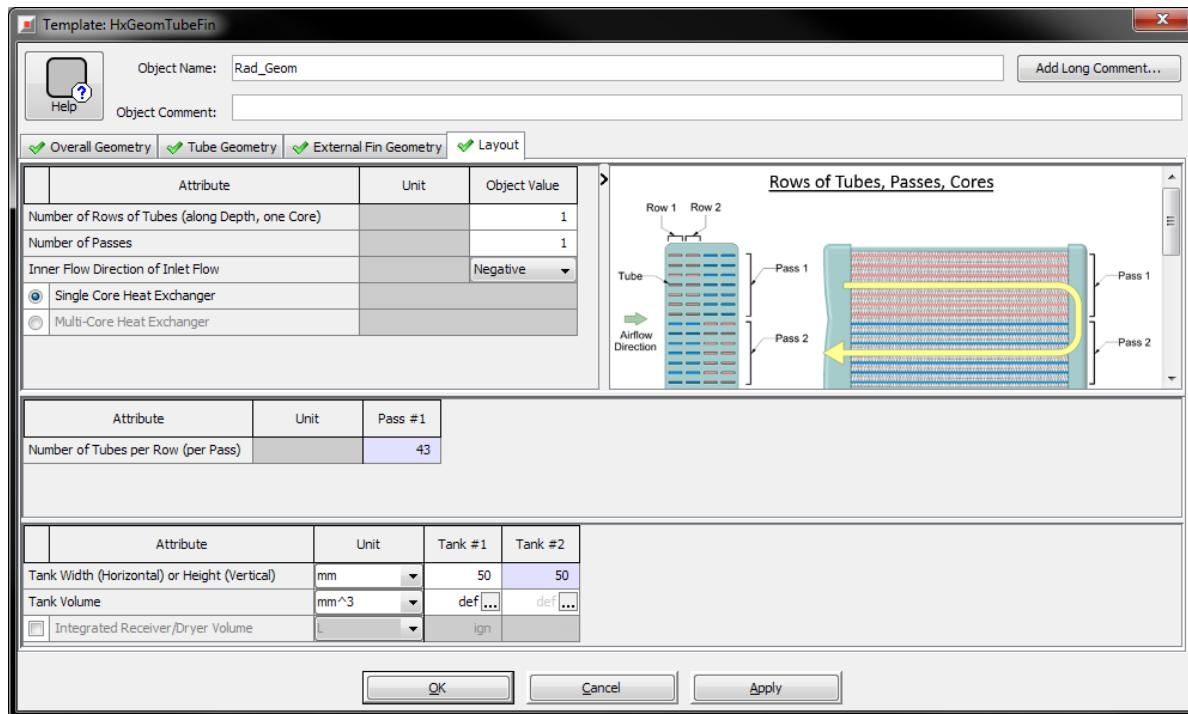


Tutorial 3: Modeling Components of a Coolant Circuit

The **Layout** folder is where the flow path configuration of the heat exchanger is defined. This includes the flow direction, number of tubes, and tank volume. The attribute **Number of Rows of Tubes** is measured along the heat exchanger depth, and is not to be confused with the number of tubes found along the height (horizontal) or width (vertical) of a heat exchanger, which is defined later. The visual image to the right of the dialog labels the **Number of Rows of Tubes**, in addition to the **Number of Passes**. Only a single heat exchanger pass will be modeled ("I" type) in this case, so the **Number of Passes** will be set to "1", which controls the column display of **Number of Tubes per Row (per Pass)** and **Tank Width or Height**. If the heat exchanger has two passes ("U" type) or three or more passes ("S" type), then more columns will be visible for these attributes. The options **Single** and **Multi-Core Heat Exchanger** should be selected to model the correct number of core (along the flow direction) that are present in the heat exchanger.

The attribute **Number of Tubes per Row (per Pass)** requires the tubes to be counted along the height or width of the heat exchanger for a single row and pass (43 tubes in this case). Knowing the total number of rows in the heat exchanger, along with the number of passes and/or cores will allow for the total number of tubes to be calculated.

The tank geometry (**Tank Width or Height**) is the last piece of information that is required to be entered, and is necessary to calculate the correct internal volume for the heat exchanger. This attribute is a quick method of calculating the tank volume, but the volume can be overridden by completing the attribute **Tank Volume** as well. Click OK when finished to complete the geometry input.

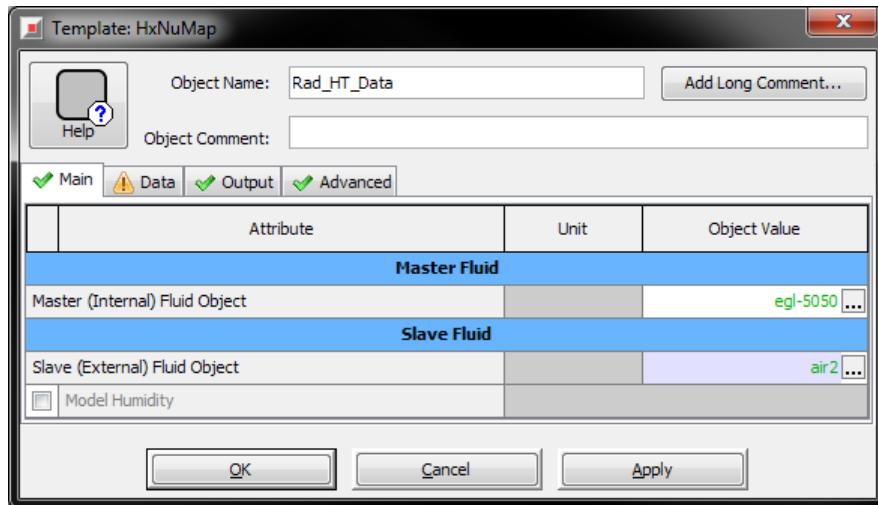


Back in the 'HeatExchangerSpecs' object, use the Value Selector for the Heat Transfer Data Object and select the 'HxNuMap' option to enter the heat transfer and pressure drop performance data for the heat exchanger. The object 'HxNuMapRefrig' should be used if one of the fluids used in the measurement of the performance data was a refrigerant.



Tutorial 3: Modeling Components of a Coolant Circuit

The **Main** folder of the 'HxNuMap' template is where the fluids used for the measurement of the performance data are entered. These fluids are used as the reference when calculating the fluid properties for the Nusselt correlation and friction factor that will be generated to represent this heat exchanger.

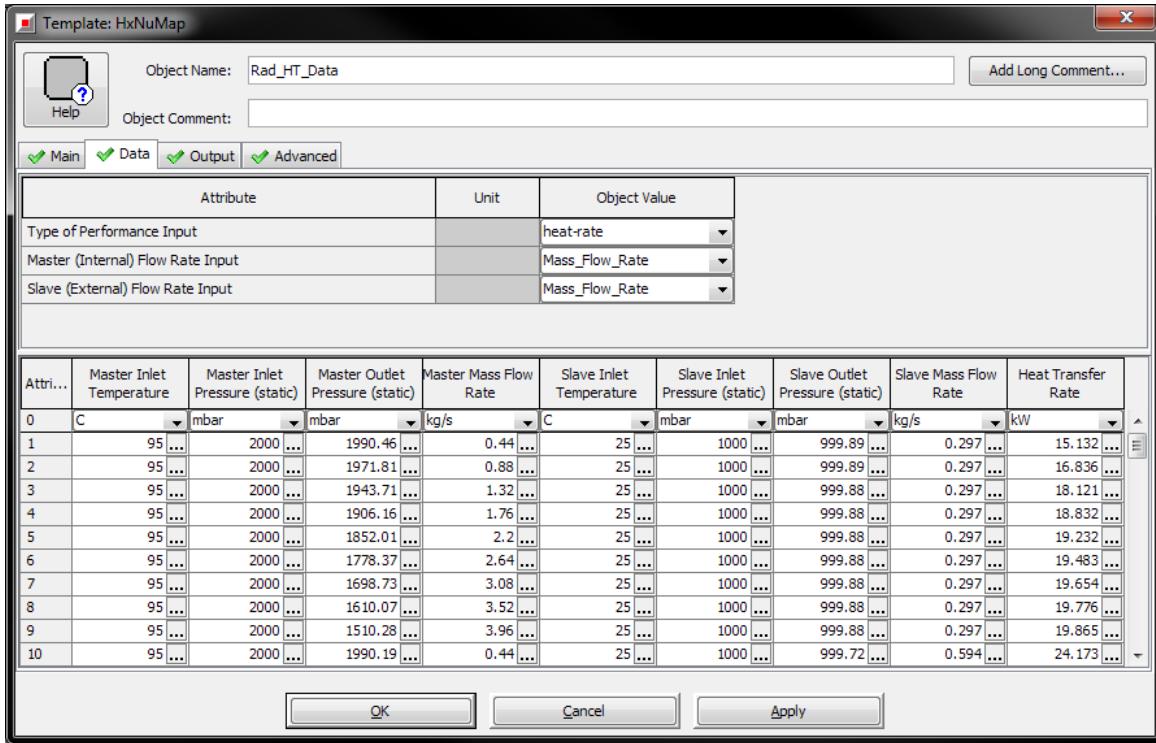


The **Data** folder is where the performance data for the heat exchanger is entered. The options selected for the Type of Performance Input, and Master and Slave Flow Rate Input will adjust the columns in the table to make it easier to enter measured data of different formats. *Warning: It is very important to ensure that the data entered has reached steady-state on the test bench, otherwise the heat balance for the heat exchanger may not be accurate for the calibration of the Nusselt correlation.*

With the combined data format of heat transfer and pressure drop, the Nusselt correlation will be created first, and with the known temperature resolution in the heat exchanger the pressure drop will be calibrated last. This method allows the preprocessing to more accurately predict the expected behavior for the heat exchanger. Click OK on the object to complete the data entry for the heat transfer performance data.



Tutorial 3: Modeling Components of a Coolant Circuit

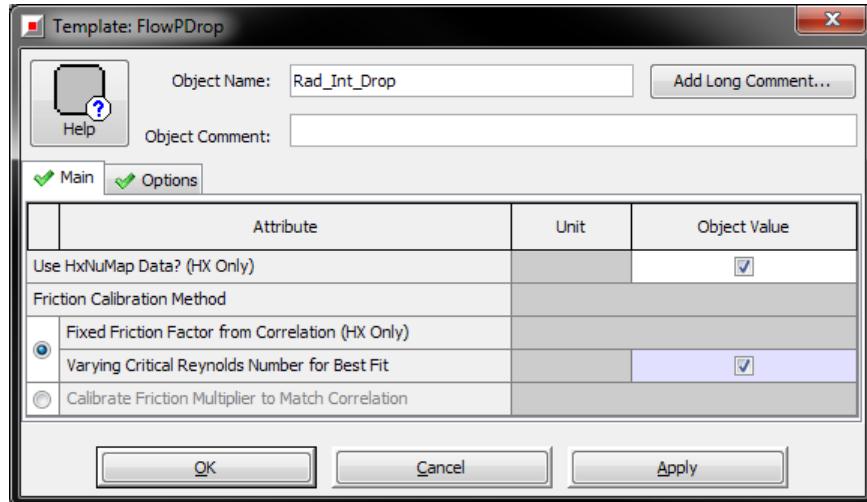


Back in the 'HeatExchangerSpecs' object, use the Value Selector for the Master (Internal) Pressure Drop Data Object and select the 'FlowPDrop' option to enter the internal pressure drop data for the heat exchanger. It is recommended to select the 'FlowPDrop' option because it allows a heat exchanger to be calibrated so that it is predictive at all operating conditions.

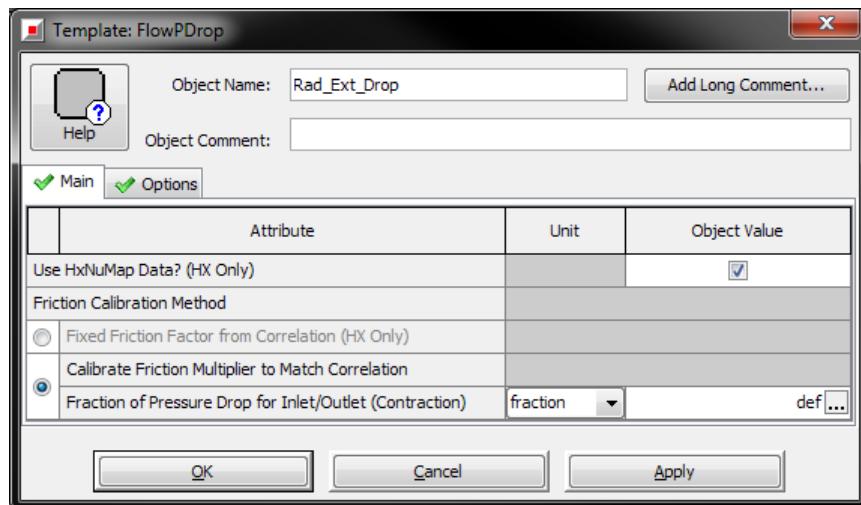
Because the inlet and outlet pressure were measured for both fluid sides in 'HxNuMap', the attribute **Use HxNuMap Data? (Hx Only)** can be enabled. When turned "on", the pressure drop will be calibrated after the Nusselt correlation so the temperature in the heat exchanger subvolumes can be taken into account. Additionally, since the internal heat exchanger is duct flow, the **Fixed Friction Factor from Correlation along with Varying Critical Reynolds Number for Best Fit** will be enabled. These options allow for the pressure drop of the heat exchanger to follow closer to Moody flow theory when the Reynolds number for transition flow may be something other than $Re = 2000$. This is highly recommend if scaled heat exchangers is of interest, or if a realistic friction pressure drop is warranted for duct flow. If it is unknown when to select this option (i.e. external fin flow), the default option **Calibrate Friction Multiplier to Match Correlation** will always attempt to find the lowest numerical error fit. Click OK on the object once the options have been selected.



Tutorial 3: Modeling Components of a Coolant Circuit



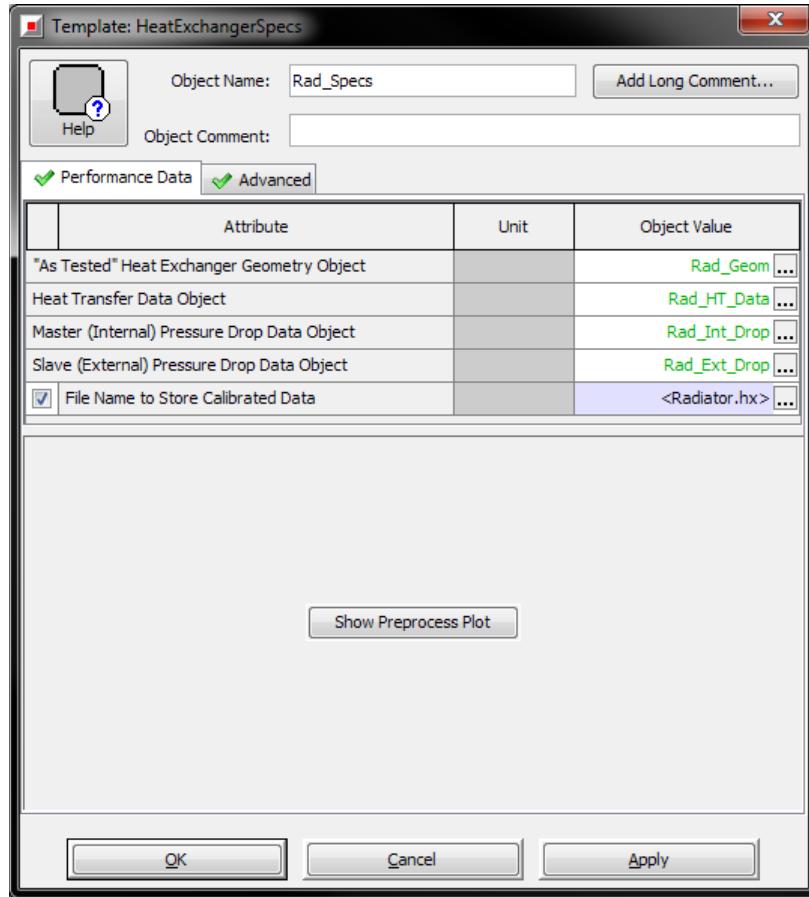
The last piece of information to be entered is the external pressure drop of the heat exchanger. Following the same procedure for the internal pressure drop, use the Value Selector for the Slave (External) Pressure Drop Data Object and select the 'FlowPDrop' option. The **Use HxNuMap Data? (HX Only)** option will be selected as well because the pressure drop data is contained in the 'HxNuMap' object. However, since the external side represents fin geometry (non-duct flow), the **Calibrate Friction Multiplier to Match Correlation** will be left as the prefill. Click OK on the object once the data has been entered.



At this time the performance data for the heat exchanger (heat transfer rate and pressure drop) has been entered. Turn on the attribute **File Name to Store Calibrated Data** and provide a filename with the extension .hx so a file can be generated after the first time the heat exchanger is calibrated and the results of the correlation can be stored for future use. This will save on preprocessing time for future model runs. If the preprocessed results of the heat exchanger need to be viewed right away, click on the button that says "Show Preprocess Plot". The heat exchanger preprocessing may take some time to complete, but when it is finished the fitting of the performance data can be analyzed immediately. Afterwards, click OK to complete the "As Tested" geometry and performance data entry.



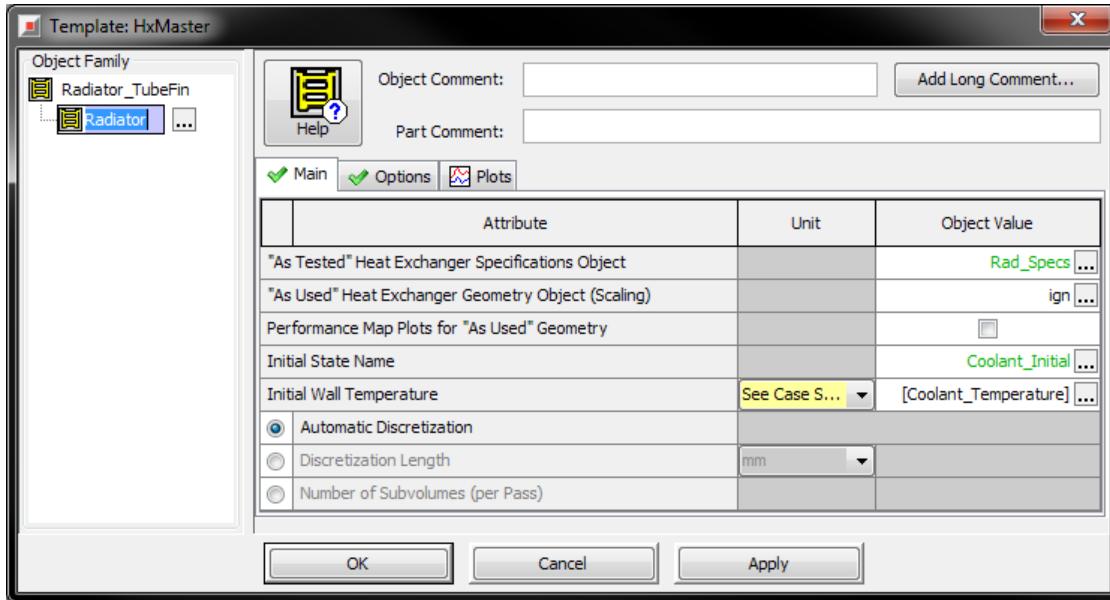
Tutorial 3: Modeling Components of a Coolant Circuit



Back in the 'HxMaster' part 'Radiator', the **"As Used" Heat Exchanger Geometry Object** is used only if scaling of the heat exchanger is of interest. If so, then a reference object will need to be created with the new heat exchanger geometry and GT-SUITE will predict the performance of the heat exchanger with any imposed boundary conditions. The attribute **Automatic Discretization** will automatically discretize the heat exchanger to a recommended number of subvolumes. Heat exchangers should be discretized to a minimum of 3 subvolumes, and ideally 5 subvolumes, to have multiple wall temperature present for the heat transfer solution. With the **Automatic Discretization** selected, the heat exchanger will be discretized to 5 subvolumes for each pass. *Warning: When discretizing a heat exchanger that is a crossflow type (i.e. HxGeomTubeFin), the upstream and downstream subvolumes around the 'HxSlave' part must be a 'FlowSplitGeneral' part unless the "Pipe Length for Inlet and Outlet Volumes" is enabled in 'HxSlave'. This is required because of the discretization rules used for the heat exchanger.*



Tutorial 3: Modeling Components of a Coolant Circuit



Click OK on the 'Radiator' part to complete the heat exchanger.

3.3.4 Testing the Behavior of a Heat Exchanger

Prior to testing the behavior of the heat exchanger, the test conditions for the component must be entered in Case Setup. Parameters have already been created for all necessary inputs to test the heat exchanger performance. Go to Case Setup to set the values for the parameters. The operating conditions selected to be used to test the heat exchanger can be anything, but it is recommended to select the same flow rate, pressure, temperature, and fluid composition so that it is easier to compare the results to the performance data. Use the Value Selector to import the correct fluid type for each side. The flow rate, pressure, and temperature can be selected from the input data in 'HxNuMap'. It is recommended to test a few operating point to ensure the heat exchanger is behaving correctly. Click OK on Case Setup when the operating points have been entered.

Main	All	+						
Parameter	Unit	Description	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Case On/Off		Check Box to Turn Case On	<input checked="" type="checkbox"/>					
Case Label		Unique Text for Plot Legends						
Coolant_Flow_Rate	kg/s	Coolant Flow Rate	1.32 ...	1.32 ...	1.32 ...	2.64 ...	2.64 ...	2.64 ...
Coolant_Pressure	bar	Coolant Pressure	2 ...	2 ...	2 ...	2 ...	2 ...	2 ...
Coolant_Temperature	C	Coolant Temperature	95 ...	95 ...	95 ...	95 ...	95 ...	95 ...
Coolant_Fluid		Coolant Composition	egl-5050 ...					
Air_Flow_Rate	kg/s	Air Flow Rate	0.594 ...	1.485 ...	2.376 ...	0.594 ...	1.485 ...	2.376 ...
Ambient_Pressure	bar	Ambient Pressure	1 ...	1 ...	1 ...	1 ...	1 ...	1 ...
Ambient_Temperature	C	Ambient Temperature	25 ...	25 ...	25 ...	25 ...	25 ...	25 ...
Ambient_Fluid		Ambient Composition	air2 ...					

Run the model at this time. Run Setup has already been filled out following GT recommendations for testing standalone components. If this is the first time the model is run, and a heat exchanger file (.hx) does not exist, then it may take a few seconds to a few minutes for the calibration of the heat exchanger to complete.



Tutorial 3: Modeling Components of a Coolant Circuit

```
*****
```

MASTER FLUID NUSSELT NUMBER CORRELATION

Re Lower Limit	= 518.624
Re Switch (Lam)	= 1360.86
Re Switch (Turb)	= 5835.49
Re Upper Limit	= 5846.17
Exponent (Lam)	= 0.599586
Exponent (Trans)	= 1.55083
Exponent (Turb)	= 0.663982
Coeff. (Lam)	= 4.106615E-02
Coeff. (Trans)	= 4.290002E-05
Coeff. (Turb)	= 9.384294E-02
Prandtl number	= 7.41786
Min hA_ratio (master)	= 6.091248E-02
Max hA_ratio (master)	= 0.940502

SLAVE FLUID NUSSELT NUMBER CORRELATION

Re Lower Limit	= 146.969
Re Switch (Lam)	= 0.0
Re Switch (Turb)	= 290.685
Re Upper Limit	= 1565.76
Exponent (Lam)	= 0.7
Exponent (Trans)	= 1.73579
Exponent (Turb)	= 0.954499
Coeff. (Lam)	= 0.0
Coeff. (Trans)	= 8.502007E-04
Coeff. (Turb)	= 7.147836E-02
Prandtl number	= 0.704575

REGRESSION ACCURACY OF OVERALL HEAT TRANSFER

Weighted Average Error (%) = 2.848504E-02

```
*****
```

INFO Processing FlowPDrop: Rad_Int_Drop (Master)

HeatExchangerSpecs object: Rad_Specs	
Calibrated Discharge Coefficient:	0.472299
Calibrated Friction Multiplier:	1.
Calibrated Critical Reynolds Number:	2799.4

INFO Processing FlowPDrop: Rad_Ext_Drop (Slave)

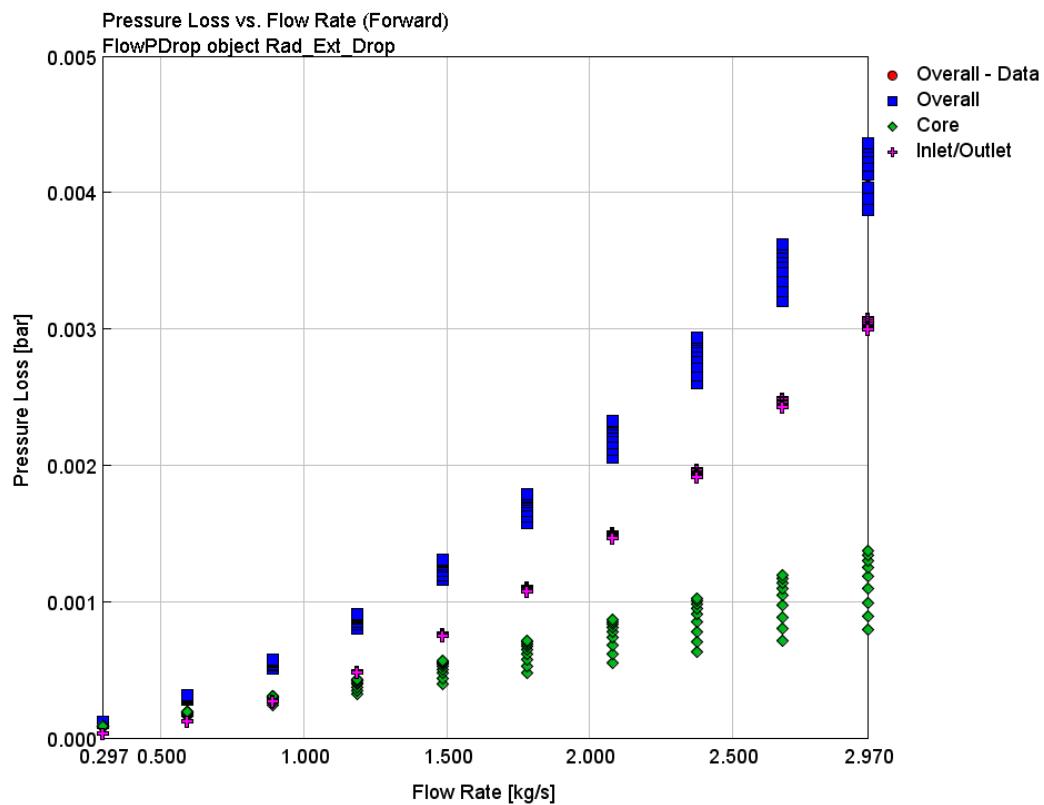
HeatExchangerSpecs object: Rad_Specs	
Calibrated Discharge Coefficient:	0.59447
Calibrated Friction Multiplier:	0.39672
Characteristic Length (mm):	1.5

However, as long as the heat exchanger file exists and is referenced, any subsequent run will skip the calibration (since it is not needed), and will use the previous calibration prior to running the model. After the model has finished running, open the results in GT-POST.

A preprocessed plot of the performance data for pressure drop and heat transfer rate is available in the GT-POST tree. The plots for pressure drop (internal and external) will be in the group(s) called FlowPDrop, and here the comparison of the pressure drop input data to the pressure drop calibration can be made. The image shown below is the fitting for the internal pressure drop.



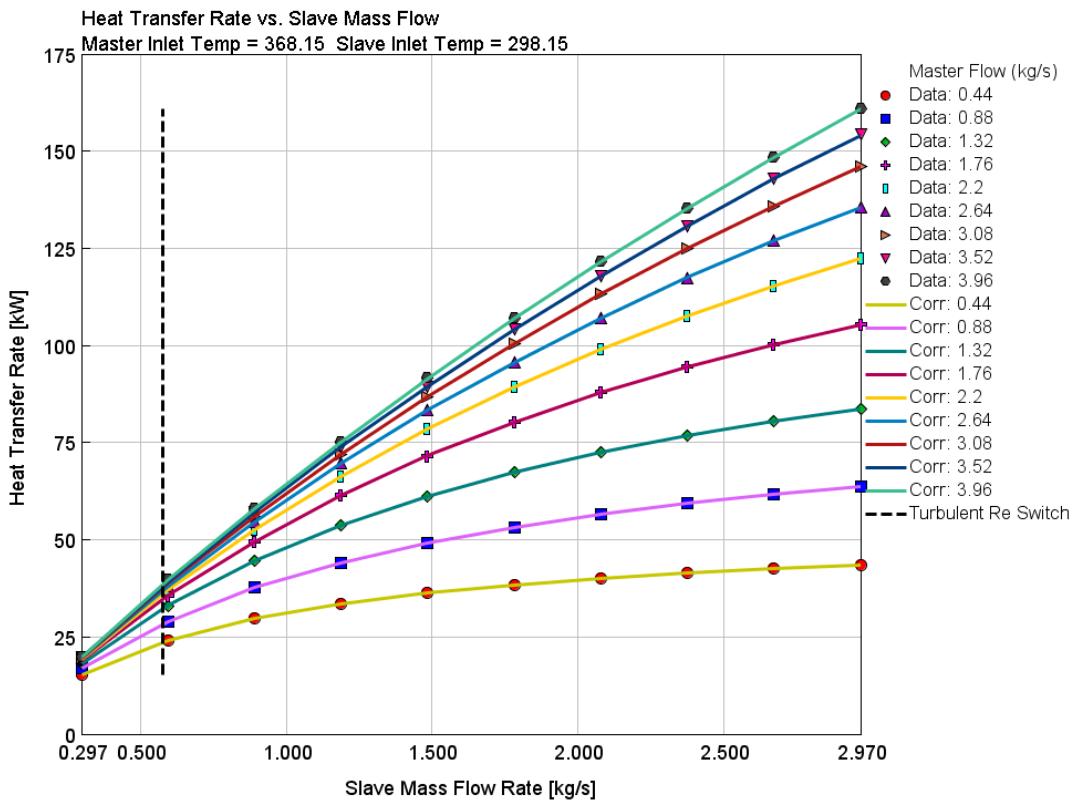
Tutorial 3: Modeling Components of a Coolant Circuit



The plots for the heat transfer rate will be in the group called 'HxNuMap', and here the comparison of the heat transfer rate performance data to the calibration can be made. The image shown below is the fitting for the heat transfer rate.



Tutorial 3: Modeling Components of a Coolant Circuit



To compare the results of the standalone heat exchanger to the input data, either use the RLT macro from the toolbar (⊕) to merge the RLTs with the performance data, or double-click on the part name in the GT-POST project map to view the Case RLTs directly and compare the values with the preprocessed plots.

HeatExchanger.gdx - Radiator								
Case:	1	Plots	Case RLT	Time RLT	RLT vs Part	Attributes	X-Axis	
		RLT Name	Unit	Case# 1	Case# 2	Case# 3	Case# 4	Case# 5
		Hide Unstored RLT Variables						
▼	Favorites							
▶	Average Pressure Drop	bar	0.05656946	0.057306726	0.05782212	0.22002289	0.21707122	0.21477376
▶	Mass Averaged Temperature (Inlet)	K	368.15	368.15	368.15	368.15	368.15	368.15
▶	Mass Averaged Temperature (Outlet)	K	361.10162	355.0586	351.70755	364.09973	359.27298	355.61075
▶	Average Wall Temperature	K	353.50607	339.17435	330.29645	361.90375	353.87842	347.25723
▶	Combined Energy Rate Out of Fluid	kW	33.09966	61.2723	76.81468	38.13684	83.3183	117.4407
▶	Average Mass Flow Rate (Inlet)	g/s	1320.0	1320.0	1320.0	2640.0	2640.0	2640.0
▶	Average Volume Flow Rate (Inlet)	L/s	1.2853676	1.2853676	1.2853675	2.5707147	2.5707152	2.5707154
▶	Pressure							
▶	Thermal							
▶	Flow							
▶	Flow Control							
▶	Init							

With either method, the results of the simulation should match with the preprocessed correlation results. If not, then review the input data to confirm that the test conditions are the same as the reference geometry and performance data entered for the heat exchanger.



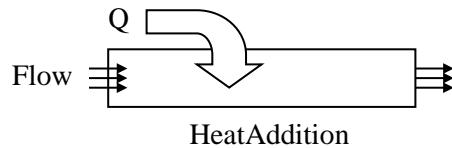
3.4 Modeling an Engine Block (or Heat Input Source)

There are two classes of templates available for modeling heat input to a fluid circuit. Heat input can be added directly to the fluid using the 'HeatAddition' template, and can be used when the rate of heat generation is known (a constant or function of time). This is ideal when attempting to model a steady-state setup where the transient response of the system is not of interest.

Alternatively, if the transient response of the system is of interest, then there are one of the three 'EngineBlock*' templates that can be used. Instead of imposing the heat input directly to the fluid, the heat input is deposited into the structure (usually in the form of a heat rejection map), which warms up, and then begins to reject heat to the fluid and also to the ambient boundary conditions.

3.4.1 Building a Heat Addition (No Mass)

Open the model HeatAddition.gtm that can be found in the directory ..\tutorials\Modeling_Applications\Cooling_Thermal_Management\03-Components\. The 'HeatAddition' template is used to impose a constant, function of time, or RLT dependent heat input rate. There is no internal solid mass included in this template, so the heat input rate is actuated directly on the fluid such that the fluid sees a temperature change following the formula $Q = \dot{m}c_p(T_{out} - T_{in})$. This is ideal when attempting to perform steady-state tests on a system to determine the final operating temperature. *Warning: If there is a significant amount of heat added to the fluid while the heat capacity rate is small, then it is possible to see sudden changes in fluid temperature that may exceed normal operating ranges.*



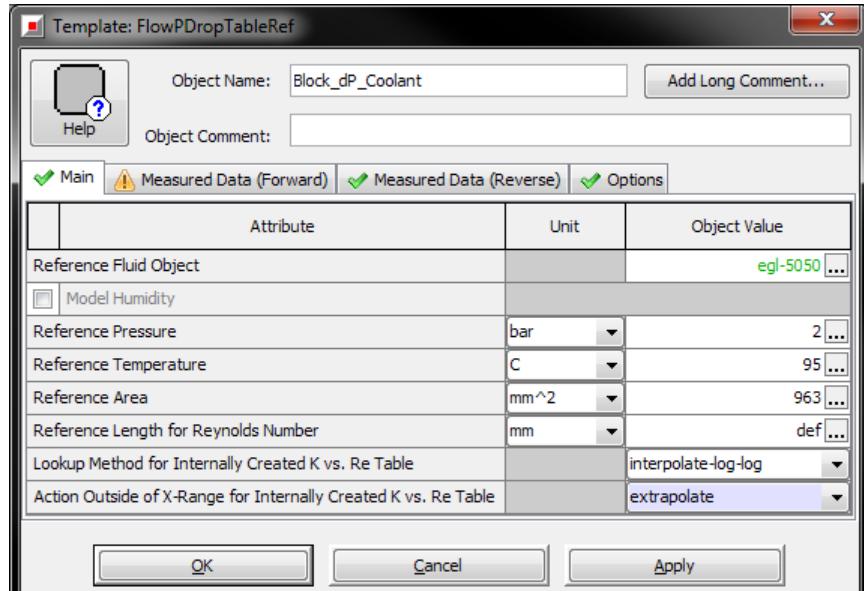
This object can be used to model auxiliary components such as an oil or transmission cooler, or cabin heater.

Double-click on the part called 'Block-NoMass' found in the project map. This part represents the engine of a system, and it requires very little input to complete the part. To complete the part, the Volume of Fluid Inside Component should be the total fluid volume of the engine block (2.2 L), which includes the water jackets and head passages. The attribute Heat Input Rate is where the heat source term can be entered as a constant rate, map, or transient, but will be left as zero for the time being. To model the pressure drop of the engine, click on the Value Selector for the attribute Pressure Drop Reference Object.

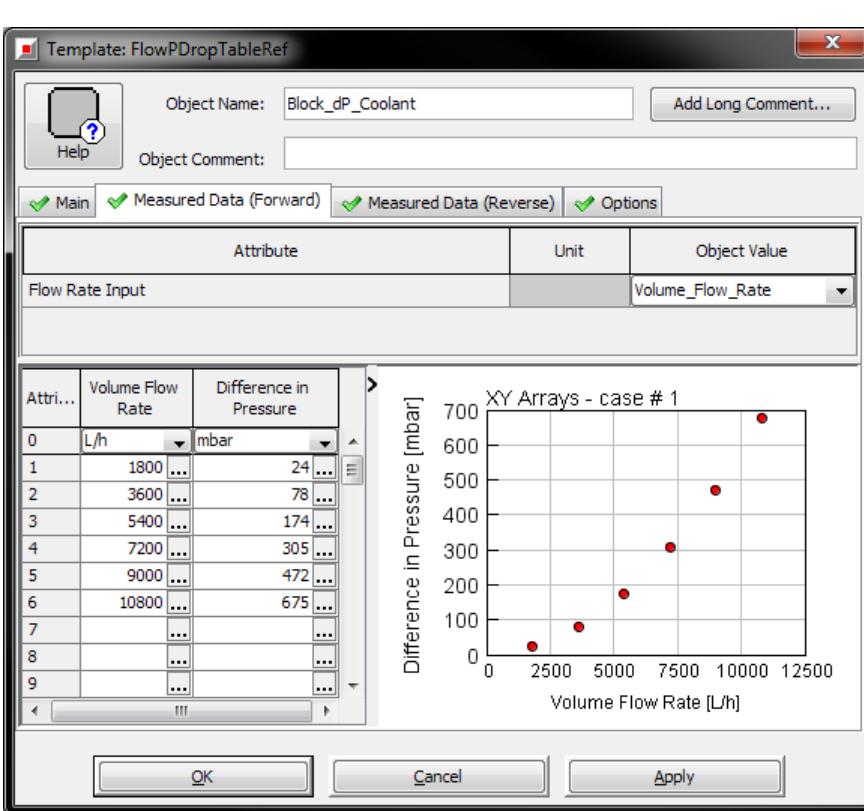
Select the template 'FlowPDropTableRef' so the pressure drop can be modeled while taking into account the changes in fluid properties as the pressure and temperature changes. This option converts the pressure drop data that is entered to a lookup table consisting of a pressure loss coefficient versus Reynolds number. The pressure drop data of the engine, which can be taken from a flow test bench, can be found in the file *DataSheets.xlsx* in the directory ..\tutorials\Modeling_Applications\Cooling_Thermal_Management\. Name the object 'Block_dP-Coolant', and copy the pressure drop data to the object from the reference file. The Reference Fluid Object, Reference Pressure, Reference Temperature, and Reference Area are necessary to create the pressure loss coefficient versus Reynolds lookup table. The table is applied to the part at run time so the pressure drop can be calculated using the instantaneous fluid properties that exist.



Tutorial 3: Modeling Components of a Coolant Circuit



The screenshot shows the 'Template: FlowPDropTableRef' dialog box with the 'Main' tab selected. The 'Object Name' field contains 'Block_dP_Coolant'. The 'Reference Fluid Object' is set to 'egl-5050'. Other settings include Reference Pressure (2 bar), Reference Temperature (95 °C), Reference Area (963 mm²), Reference Length for Reynolds Number (def), Lookup Method (interpolate-log-log), and Action Outside of X-Range (extrapolate). Buttons at the bottom are OK, Cancel, and Apply.

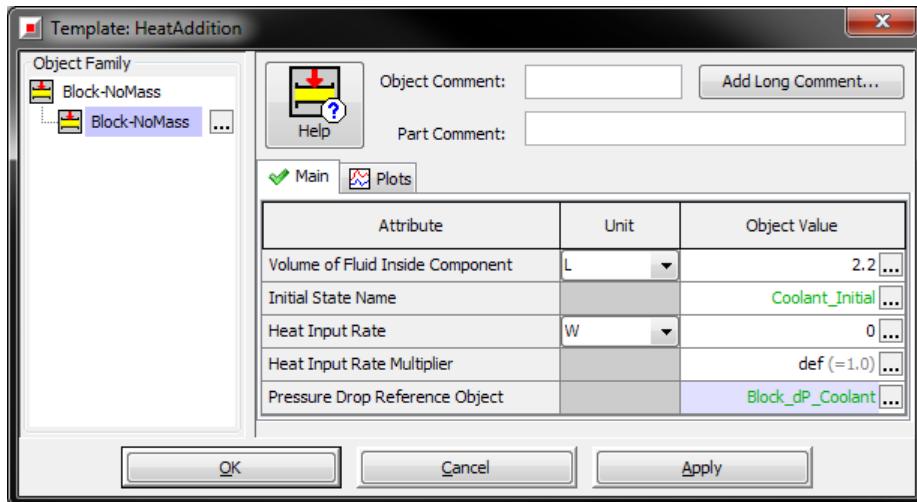


The screenshot shows the 'Template: FlowPDropTableRef' dialog box with the 'Measured Data (Forward)' tab selected. The 'Object Name' field contains 'Block_dP_Coolant'. The 'Flow Rate Input' attribute is set to 'Volume_Flow_Rate'. A table below lists Volume Flow Rate (L/h) and Difference in Pressure (mbar) for various flow rates. To the right is a scatter plot titled 'XY Arrays - case # 1' showing the relationship between Volume Flow Rate [L/h] and Difference in Pressure [mbar]. Buttons at the bottom are OK, Cancel, and Apply.

Click OK when finished to complete the 'Block_dP-Coolant' object, and then click OK again to accept the pressure drop object in the 'Block-NoMass' part.



Tutorial 3: Modeling Components of a Coolant Circuit



At this time, feel free to complete Case Setup with the operating conditions of the fluid circuit to test the behavior of the 'HeatAddition' pressure drop, or the temperature rise if a non-zero Heat Input Rate was entered.

Main	All							
Parameter	Unit	Description	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Case On/Off		Check Box to Turn Case On	<input checked="" type="checkbox"/>					
Case Label		Unique Text for Plot Legends						
Pump_Flow_Rate	L/h	Coolant Flow Rate	1800 <input type="button" value="..."/>	3600 <input type="button" value="..."/>	5400 <input type="button" value="..."/>	7200 <input type="button" value="..."/>	9000 <input type="button" value="..."/>	10800 <input type="button" value="..."/>
Coolant_Pressure	bar	Coolant Pressure	2 <input type="button" value="..."/>					
Coolant_Temperature	C	Coolant Temperature	95 <input type="button" value="..."/>					
Coolant_Fluid		Coolant Composition	egl-5050 <input type="button" value="..."/>					

After running the model, the RLT macro can be used to compare the resulting pressure drop to the data that was entered to ensure the correct behavior is achieved.

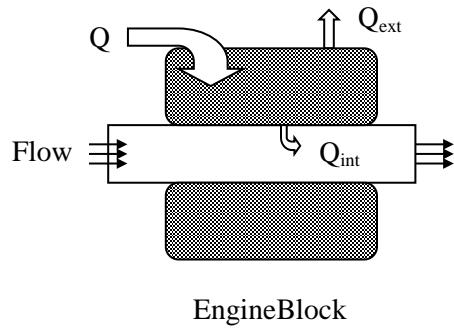
3.4.2 Theory of Engine Block (With Mass)

There are three templates available to model the heat input rate into the mass of a component to take into account the thermal inertia and transient delay of heat rejection to the fluid(s) in an engine block.

EngineBlock

When using the 'EngineBlock' template, heat is rejected from the engine and is passed to a lumped thermal mass model. This is the most basic template that can be used to have some sort of transient behavior in a model. For a more physical behavior, please see 'EngineBlock-3Mass' or 'EngineBlock-5Mass'.

Tutorial 3: Modeling Components of a Coolant Circuit



The engine block is assumed to have a uniform temperature, T_{wall} , which is calculated from a balance of the heat rejected by the engine, heat transfer to the coolant, and heat transfer to the environment.

$$\frac{dT_{wall}}{dt} = \frac{Q - Q_{int} - Q_{ext}}{\rho_{wall} C_{p-wall} V_{wall}}$$

The heat rejection rate, Q , can be obtained as a function of engine speed and load, or can be specified by the user as a constant, function of time, or RLT dependence. The heat transfer to the coolant, Q_{int} , is calculated as

$$Q_{int} = (hA)_{int}(T_{wall} - T_{flow})$$

where hA is calculated as

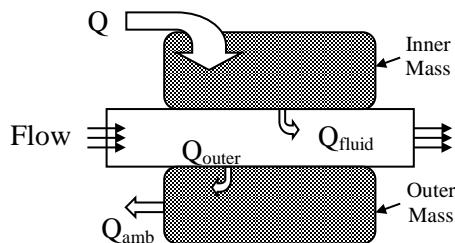
$$hA = (hA)_{ref} \left(\frac{\dot{m}}{\dot{m}_{ref}} \right)^{0.75}$$

The reference value of hA is calculated from the temperature increase in the coolant and the temperature difference between the coolant and the wall that are supplied by the user at the reference flow rate. The heat transfer to the environment, Q_{ext} , is calculated as

$$Q_{ext} = (hA)_{ext}(T_{wall} - T_{ext}) + \sigma A(T_{wall}^4 - T_{ext}^4)$$

EngineBlock-3Mass

When using the 'EngineBlock-3Mass' template, heat is rejected from the engine into an inner mass that represents the cylinder liner, to the cooling fluid as a single volume, to an outer mass that represents the remainder of the block material, and finally the ambient environment.



EngineBlock-3Mass



While the modeling approach is still primitive, it offers a more physical transient behavior when compared to the 'EngineBlock' template. It is recommended to use 'EngineBlock-3Mass' as starting point for any transient behavior in the engine block for a cooling model. This option does not handle thermal distribution in the engine block because the heat rate is imposed. If thermal distribution in the engine block is the primary consideration, then it is recommended to see the template 'EngCylStrucCond'.

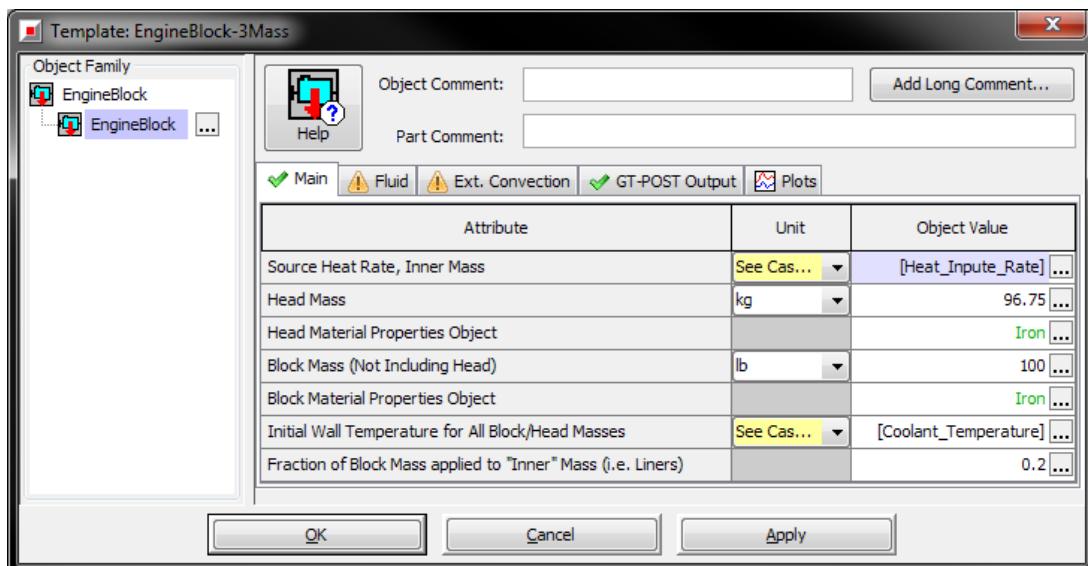
EngineBlock-5Mass

The 'EngineBlock-5Mass' is similar to the 'EngineBlock-3Mass' template, except it allows two fluids to be modeled simultaneously (i.e. coolant and oil) to capture some effect of the heat distribution in the engine block. However, it is still recommended to use the 'EngCylStrucCond' template if the main focus of the engine block is thermal distribution.

3.4.3 Building an Engine Block (With Mass)

Open the model EngineBlock.gtm that can be found in the directory ..\tutorials\Modeling_Applications\Cooling_Thermal_Management\03-Components\. Double-click on the part 'EngineBlock' found in the project map. This part will be used to model a 3 mass engine block ('EngineBlock-3Mass') to capture some thermal transient response.

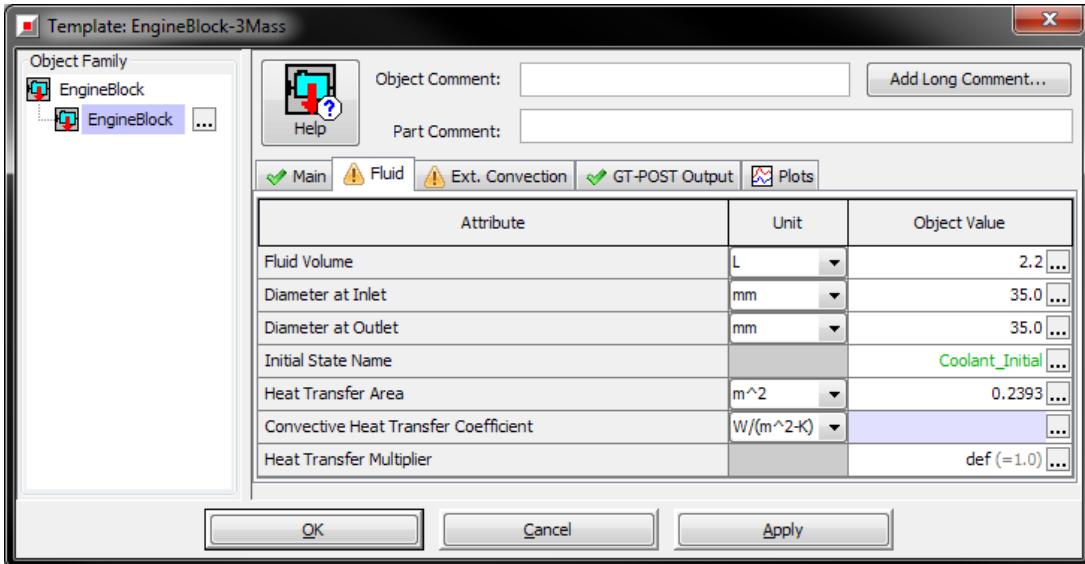
The **Main** folder contains the information for the source heat rate (combustion heat release), and the structural properties of the engine block. The source heat release will have to be measured from a test bench. Complete the **Main** folder using the information below.



The **Fluid** folder is where information concerning the cooling side geometry is entered. This includes the Fluid Volume, Heat Transfer Area, and Heat Transfer Coefficient between the cooling fluid and block material. The cooling Heat Transfer Area is typically measured from the CAD geometry.



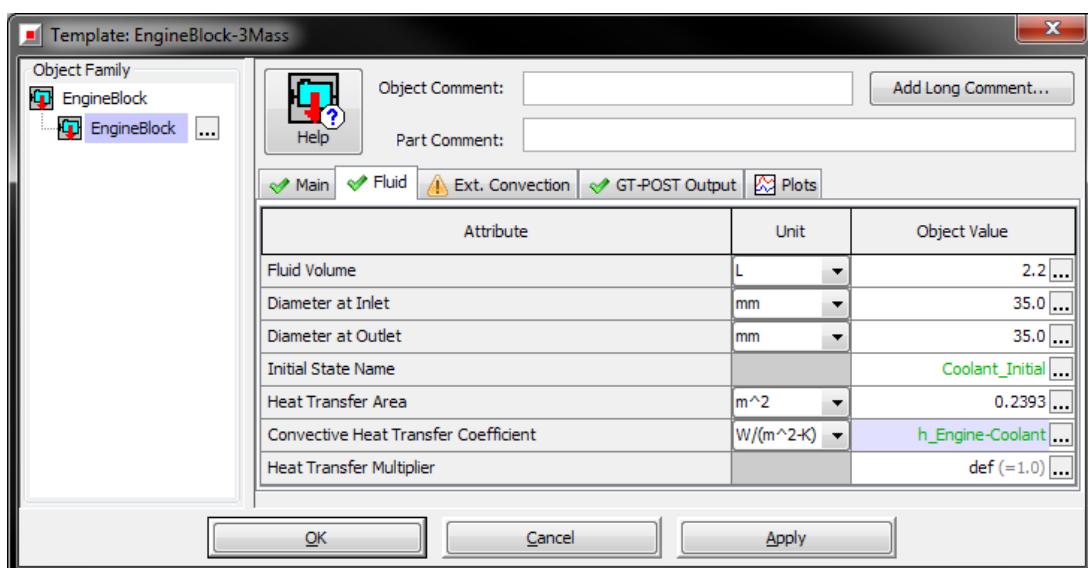
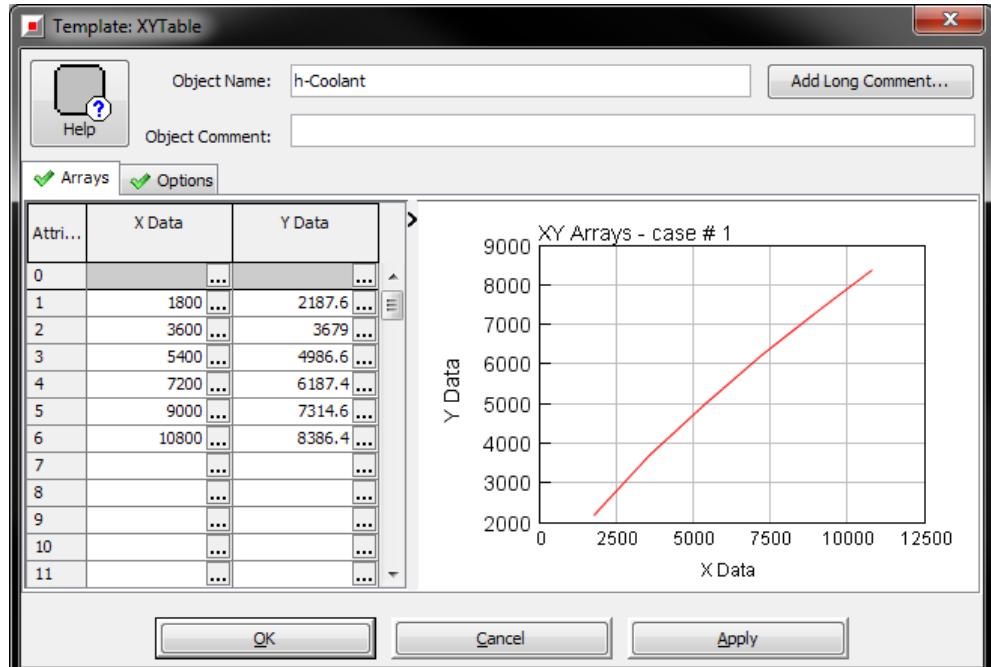
Tutorial 3: Modeling Components of a Coolant Circuit



To complete the cooling convection coefficient, which is measured from a test bench knowing the heat input rate and the inner mass and fluid temperature at various operating points, use the Value Selector and select the 'RLTDependencyXY' option to create a lookup table for the convection coefficient. An RLT lookup will be affected by the RLT calculation interval in Output Setup (**Home** tab), and will use the result of the previous time step as an input to the lookup table for the current time step. The input variable for the 'RLTDependencyXY' will be the coolant volume flow rate through the engine block. Use the Value Selector to select the Volume Flow Rate RLT in the 'PressureLossConn' connection called 'Block_dP'. Select units of [L/h] prior to accepting this variable. The Initial X Input will be 0, and the Dependence Object will point to a 'XYTable' object that is the lookup table for the coolant convection coefficient. The measured data for the convection coefficient can be copied from the file *DataSheets.xlsx* in the directory ..\tutorials\Modeling_Applications\Cooling_Thermal_Management. Click OK for the 'XYTable' when the data has been entered, and again on the 'RLTDependency' accept all changes and be brought back to the 'EngineBlock-3Mass'.



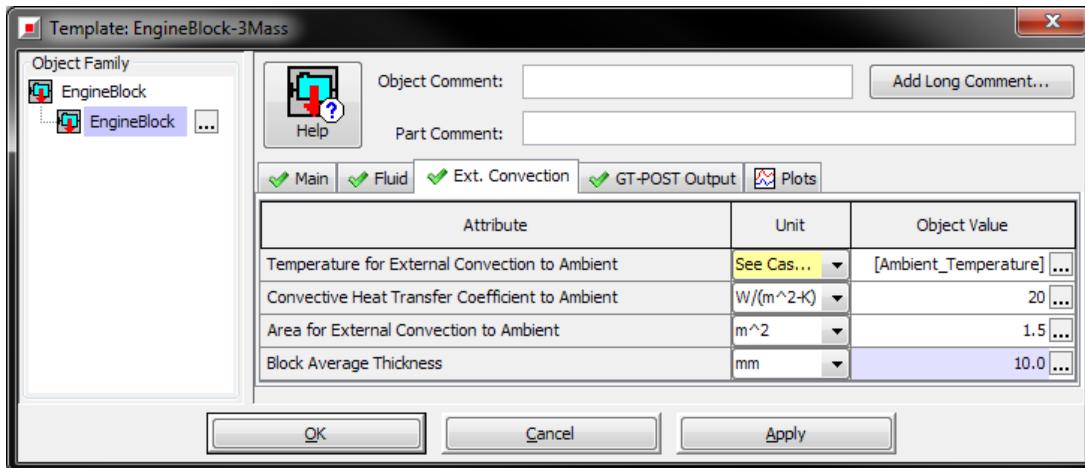
Tutorial 3: Modeling Components of a Coolant Circuit



The **Ext. Convection** folder is where the ambient boundary conditions are entered for the engine block object. The Convective Heat Transfer Coefficient to Ambient can usually be estimated to be about 20 W/m²-K for stagnant flow, but will be higher for forced convection across the engine block. The convection coefficient selected will affect the heat rate rejection from the outer mass. The Area for External Convection to Ambient can be measured from the CAD geometry of the engine block.

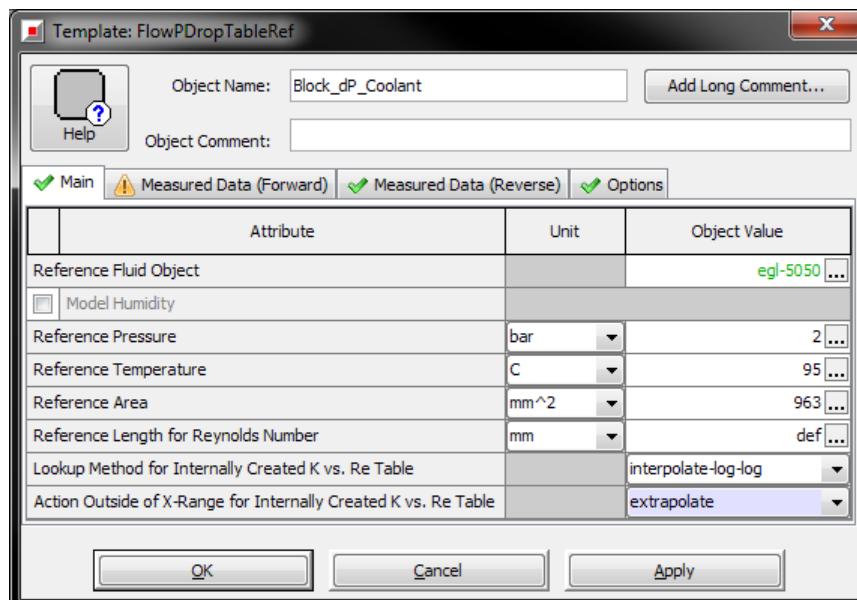


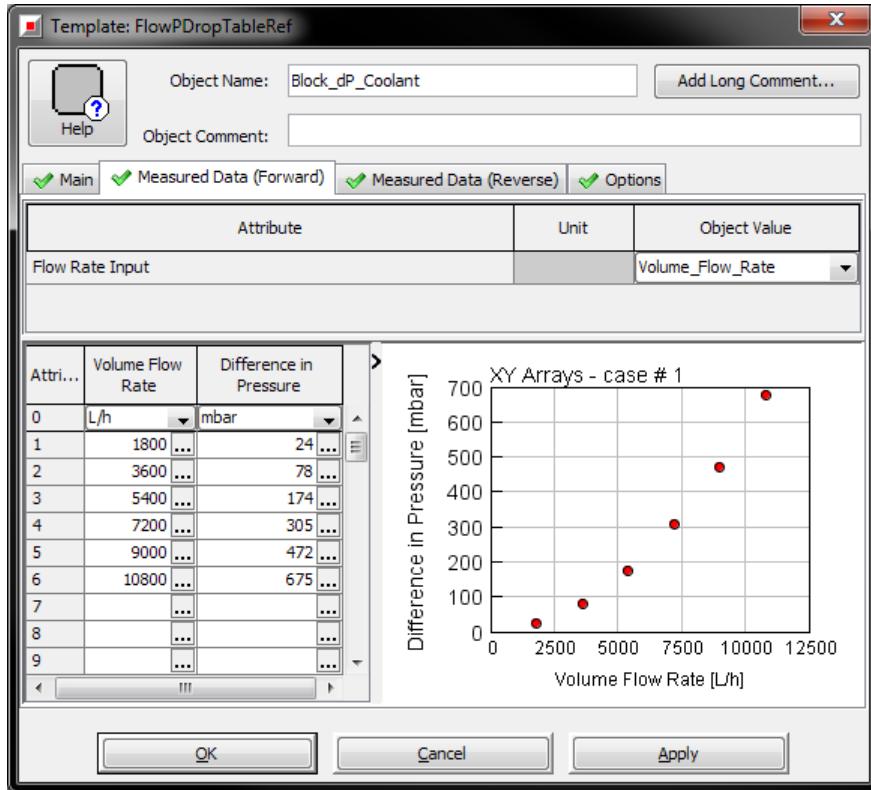
Tutorial 3: Modeling Components of a Coolant Circuit



Click OK in the dialog to accept all changes to model the engine block.

To complete the pressure drop for the engine block, open the connection part 'Block_dP' that is downstream of the 'EngineBlock' part. This object is a 'PressureLossConn', and is used to model the lumped pressure drop of a component. Select the template 'FlowPDropTableRef' so the pressure drop can be modeled while taking into account the changes in fluid properties as the pressure and temperature changes. This option converts the pressure drop data that is entered to a lookup table consisting of a pressure loss coefficient versus Reynolds number. The pressure drop data of the engine, which can be taken from a flow test bench, can be found in the file *DataSheets.xlsx* in the directory *..\tutorials\Modeling_Applications\Cooling_Thermal_Management*. Name the object 'Block_dP_Coolant', and copy the pressure drop data to the object from the reference file. The Reference Fluid Object, Reference Pressure, Reference Temperature, and Reference Area are necessary to create the pressure loss coefficient versus Reynolds lookup table. The table is applied to the part at run time so the pressure drop can be calculated using the instantaneous fluid properties that exist.





Click OK when finished to complete the 'Block_dP-Coolant' object, and then click OK again to accept the pressure drop object in the 'Block_dP' part.

3.4.4 Testing the Behavior of an Engine Block (With Mass)

A Case Sweep can be created in Case Setup to investigate the heat rate distribution in the engine block and thermal response under different operating conditions, or to test the pressure drop performance. Unlike the previous tutorials, the Automatic Shut-Off When Steady-State has been turned to "off" to force the simulation to run for the full duration in the Run Setup dialog. This will also cause the thermal inertia of the masses to be taken into account because the Thermal Solver in the **ThermalControl** folder of Run Setup is set to "automatic".

The image below shows one example with a constant heat input rate to study the temperature of the masses and how they respond with a constant inlet fluid temperature. To turn on the temperature plots, open the 'EngineBlock' part on the map and select the **Plots** folder.

The table lists parameters and their values. The 'Case On/Off' parameter is checked. The 'Case Label' is 'Unique Text for Plot Legends'. Other parameters include Heat_Input_Rate, Pump_Flow_Rate, Coolant_Pressure, Coolant_Temperature, Coolant_Fluid, and Ambient_Temperature.

Parameter	Unit	Description	Case 1
Case On/Off		Check Box to Turn Case On	<input checked="" type="checkbox"/>
Case Label		Unique Text for Plot Legends	
Heat_Input_Rate	kW	Source Heat Rate, Inner Mass	63
Pump_Flow_Rate	L/h	Coolant Flow Rate	10800
Coolant_Pressure	bar	Coolant Pressure	2
Coolant_Temperature	C	Coolant Temperature	95
Coolant_Fluid		Coolant Composition	egl-5050
Ambient_Temperature	C		25



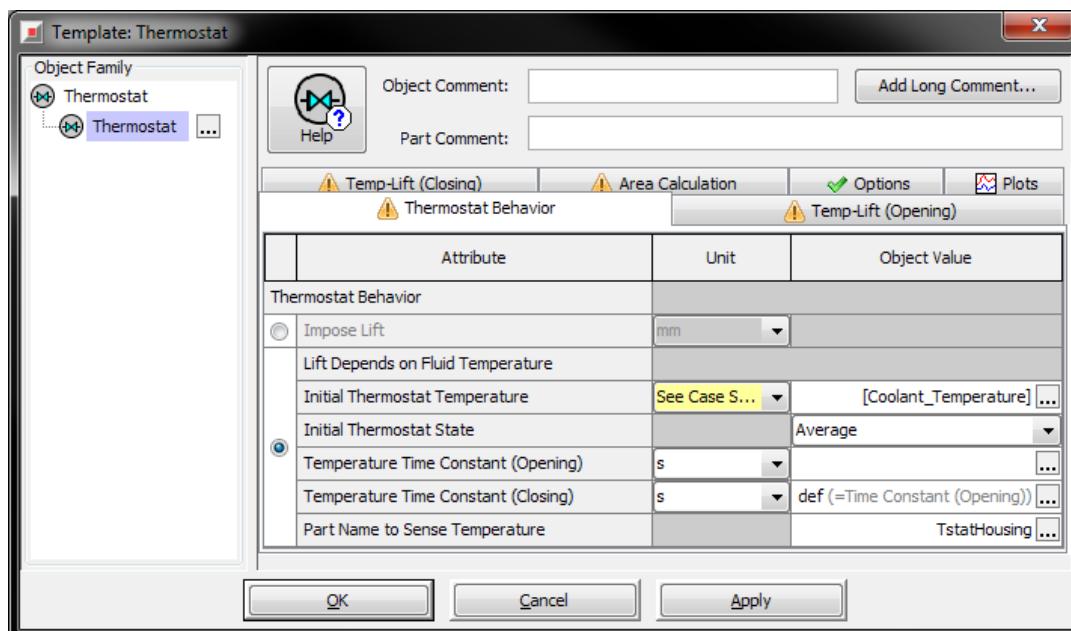
Alternatively, a detailed analysis can be performed by imposing a 'ProfileTransient' for the coolant temperature or heat input rate to study the effects of a step change in one of these variables.

3.5 Modeling a Thermostat

Open the model Thermostat.gtm that can be found in the directory ..\tutorials\Modeling_Applications\Cooling_Thermal_Management\03-Components\. This tutorial is partially set up to test the operation of a thermostat component that includes a bypass valve. The missing information to test a thermostat can be acquired by measuring thermostat behavior. This behavior typically includes the thermostat lift versus wax temperature, a calculated time constant for the thermostat response to a temperature change, and a pressure drop versus flow rate at various lift positions. Also required are the reference fluid conditions such as the pressure, temperature, and fluid type, and the reference thermostat diameters.

3.5.1 Building a Thermostat

Open the part called 'Thermostat' found in the project map. This part allows a single pair of opening and closing temperature-lift profiles to be entered in order to model the operation of a thermostat valve. The section Lift Depends on Fluid Temperature is selected so that the thermostat lift will adjust from the sensed temperature (Part Name to Sense Temperature). The attribute Initial Thermostat State can affect the starting behavior of the thermostat, so at times this value may need to be adjusted in order for the starting thermostat lift to be correct from using the entered Initial Thermostat Temperature. The Temperature Time Constant (Opening) will be calculated from measured data and mimic the response of the thermostat as the fluid temperature changes. The time constant in this case can be found in the file DataSheets.xlsx in the directory ..\tutorials\Modeling_Applications\Cooling_Thermal_Management\.

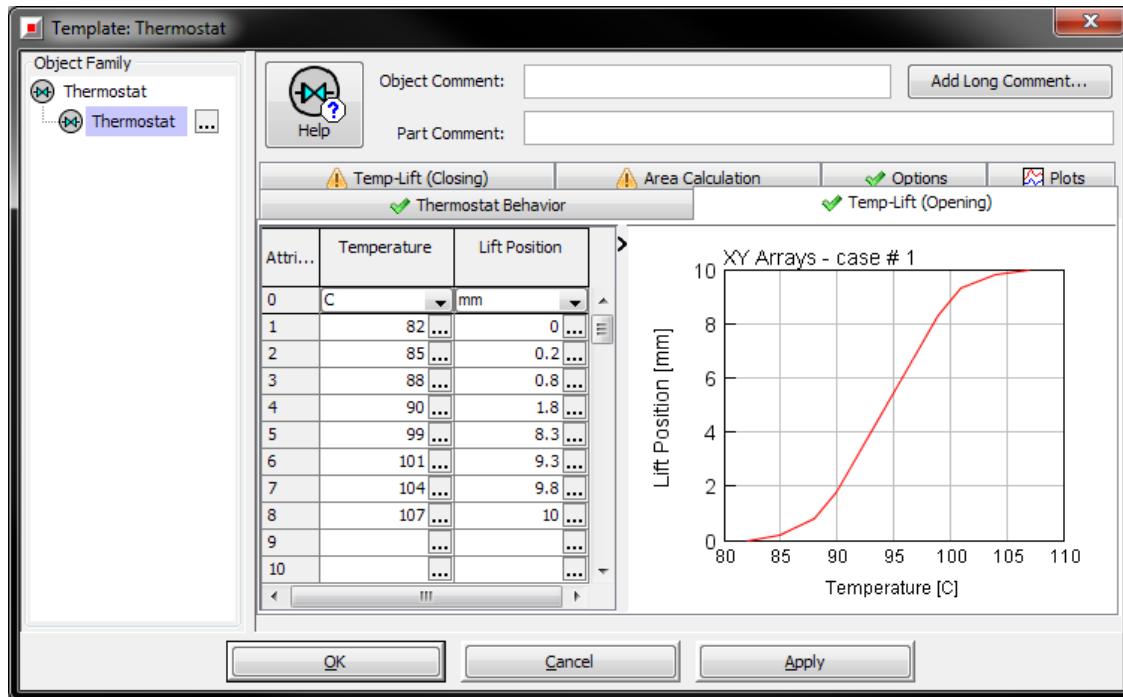


The section for Impose Lift is useful when a specific thermostat lift needs to be imposed, such as when trying to confirm the thermostat pressure as a specific position, or ensuring a branch in the system receives adequate flow at a specific fluid temperature.

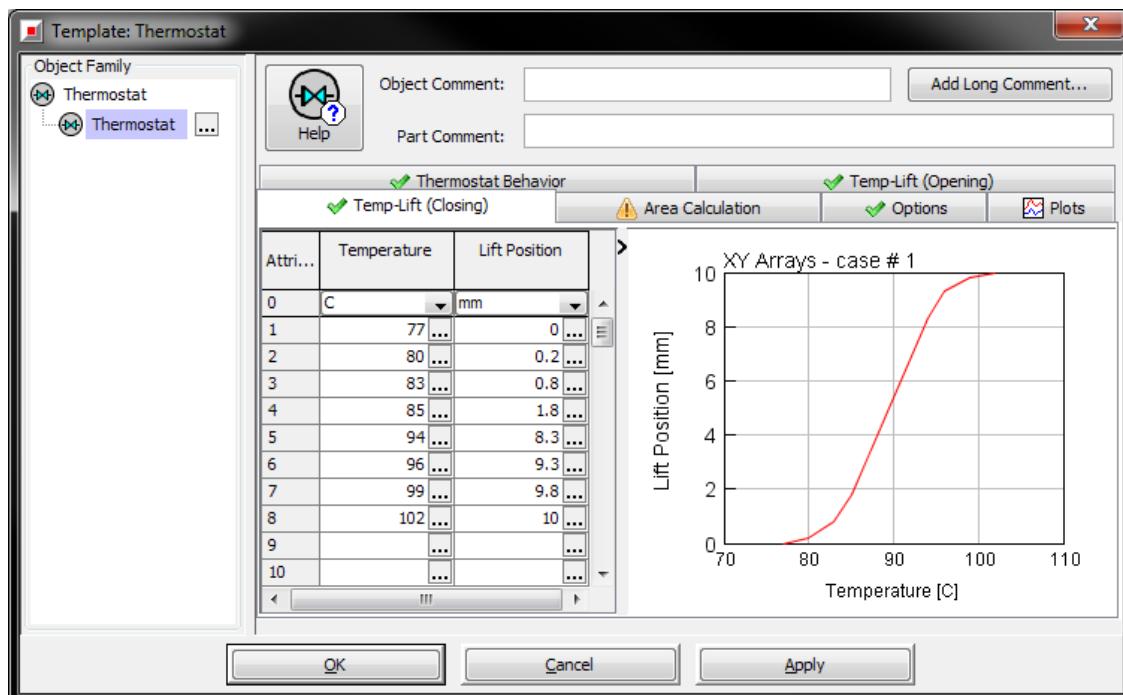


Tutorial 3: Modeling Components of a Coolant Circuit

The thermostat temperature-lift profiles for the opening curve (temperature change over time is positive) and the closing curve (temperature change over time is negative) can be entered independent of each other. The opening curve should be entered in the **Temp-Lift (Opening)** folder.



The closing curve should be entered in the **Temp-Lift (Closing)** folder.

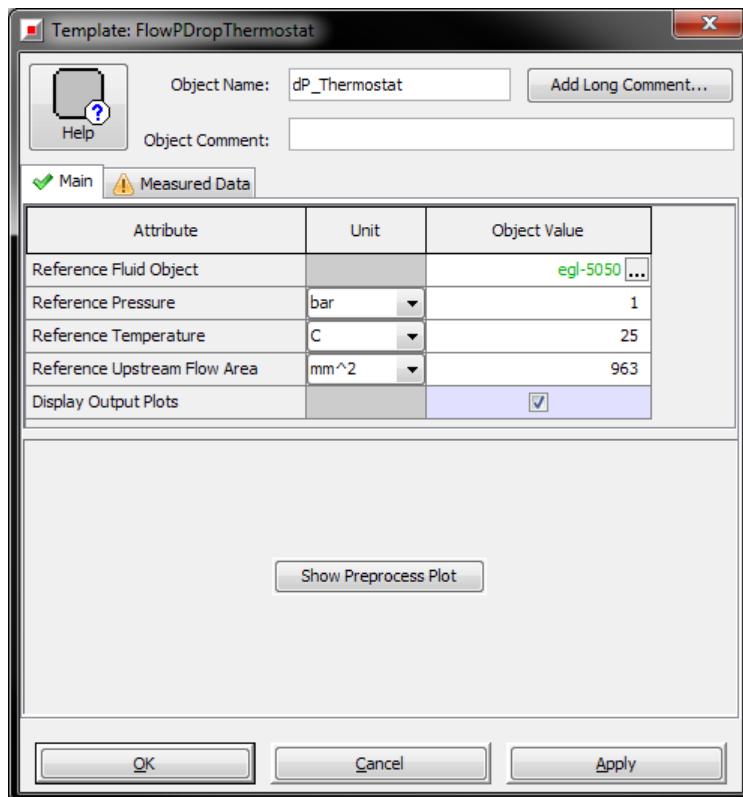


Tutorial 3: Modeling Components of a Coolant Circuit

The thermostat pressure drop can be entered in the **Area Calculation** folder, and in one of two forms. The more common data input will use the Calculate Effective Area Based on dP vs Flow attribute. This allows the thermostat pressure drop to be entered as pressure drop versus flow rate at constant lift positions. The solver will then calculate the average effective area for each lift position. Alternatively, the effective area can be entered directly if the Enter Effective Area is selected.

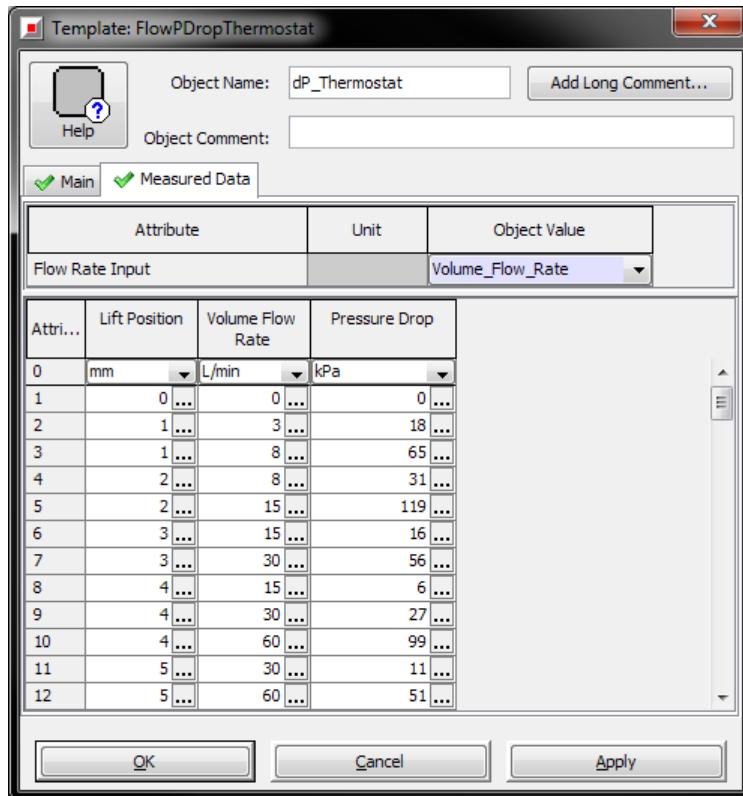
Using the Value Selector for the Calculate Effective Area Based on dP vs. Flow attribute, select the 'FlowPDropThermostat' template to enter the measured pressure drop data for the main thermostat valve. Name the object 'dP_Thermostat'.

The **Main** folder is where the reference conditions for the measured data are entered. The reference fluid conditions and type are necessary in order to calculate the correct fluid properties, and the Reference Upstream Flow Area is required in order to take into account the pressure recovery after the valve. The thermostat reference diameter of 35 mm provides the upstream flow area of 963 mm².



The pressure drop for the valve can be entered in the **Measured Data** folder. Separate pressure drop versus flow rates profiles were measured at constant lift positions.





Clicking on the "Show Preprocess Plot" button in the Main folder will display the calibrated effective area for the pressure drop data that is entered.

Click OK on the 'dP_Thermostat' object to accept all changes, and again on the 'Thermostat' object to finish building the valve.

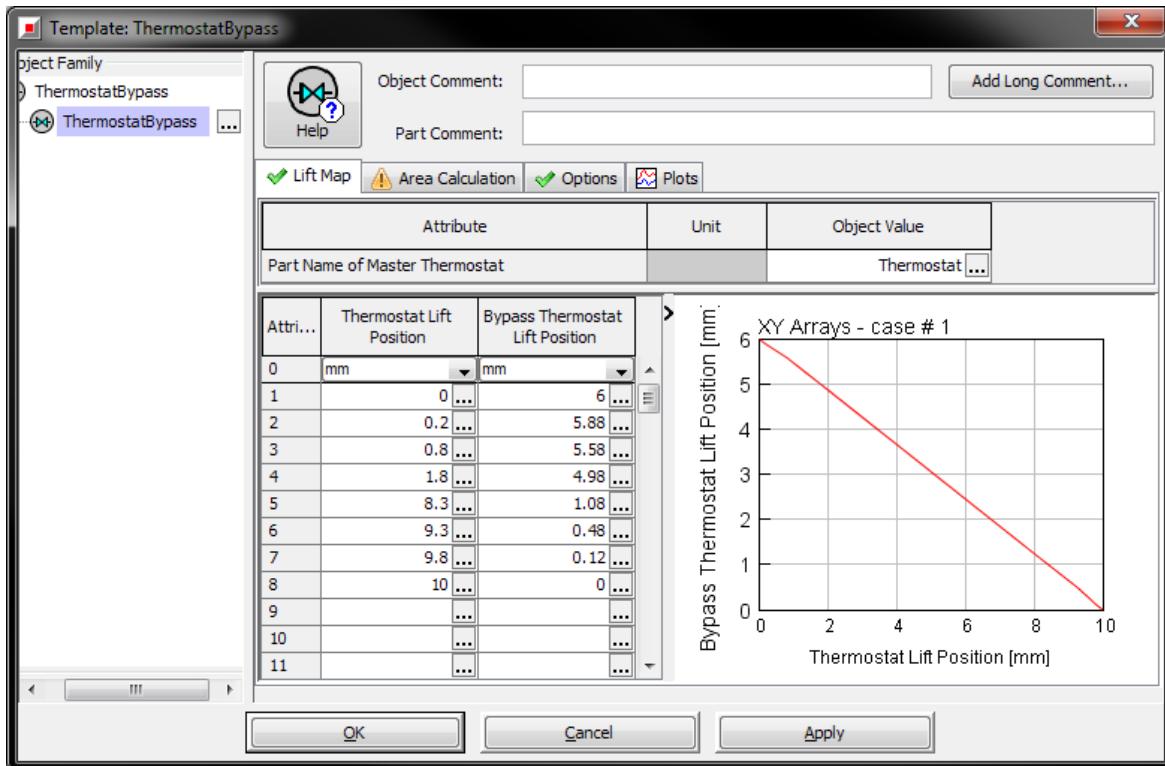
3.5.2 Building a Thermostat Bypass

In order to build a bypass thermostat valve there must be a 'Thermostat' or 'ThermostatHeated' part already on the map. The lift position of the bypass thermostat valve is dependent on the lift position of the reference part. This is because as the main thermostat valve changes position, an attached spring automatically adjusts the position of the bypass valve. This allows greater control over the flow distribution through the thermostat.

Open the part 'ThermostatBypass' to build the bypass valve. The attribute Part name of Master Thermostat should reference the part name of the 'Thermostat' or 'ThermostatHeated' part that is already on the map. This reference will link the two parts together so the lift position of the thermostat can be communicated with the bypass. The **Lift Map** folder is where the lift mapping of the thermostat and bypass is entered.



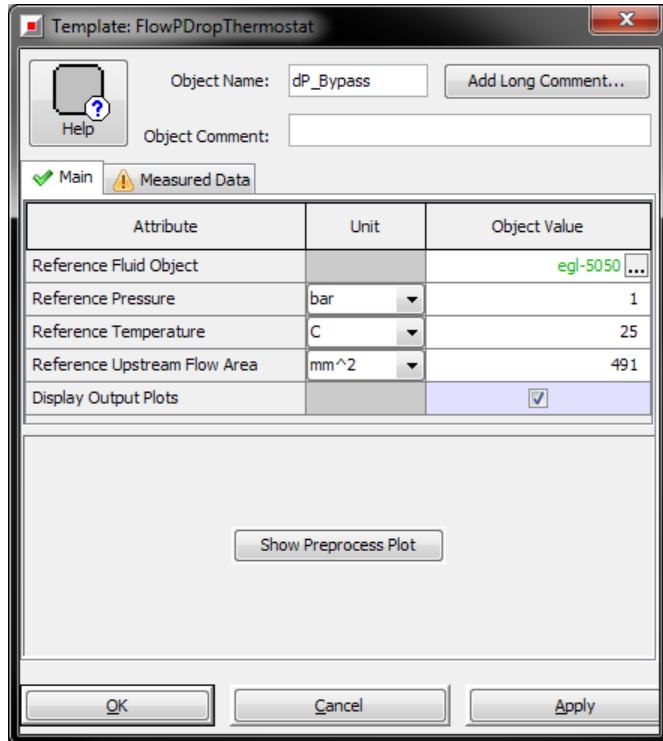
Tutorial 3: Modeling Components of a Coolant Circuit



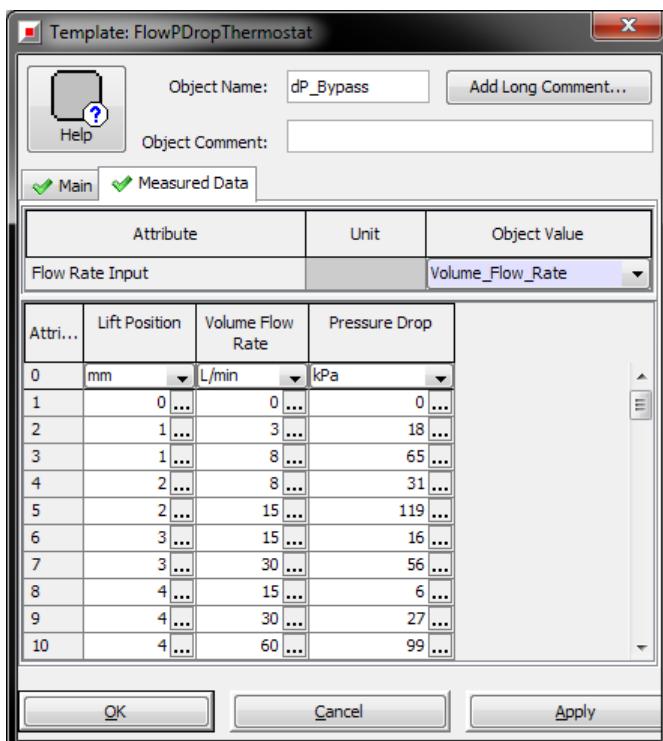
The pressure drop for the bypass valve can be entered by selecting the Value Selector for the attribute Calculate Effective Area Based on dP vs. Flow in the **Area Calculation** folder. Name the 'FlowPDropThermostat' object 'dP_Bypass'. The reference fluid conditions and type are necessary in order to calculate the correct fluid properties, and the Reference Upstream Flow Area is required in order to take into account the pressure recovery after the valve. The thermostat reference diameter of 25 mm provides the upstream flow area of 491 mm².



Tutorial 3: Modeling Components of a Coolant Circuit



The pressure drop for the bypass valve can be entered in the **Measured Data** folder. Separate pressure drop versus flow rates profiles were measured at constant lift positions.



Tutorial 3: Modeling Components of a Coolant Circuit

Clicking on the "Show Preprocess Plot" button in the Main folder will display the calibrated effective area for the pressure drop data that is entered.

Click OK on the 'dP_Thermostat' object to accept all changes, and again on the 'Thermostat' object to finish building the valve.

3.5.3 Testing the Behavior of a Thermostat

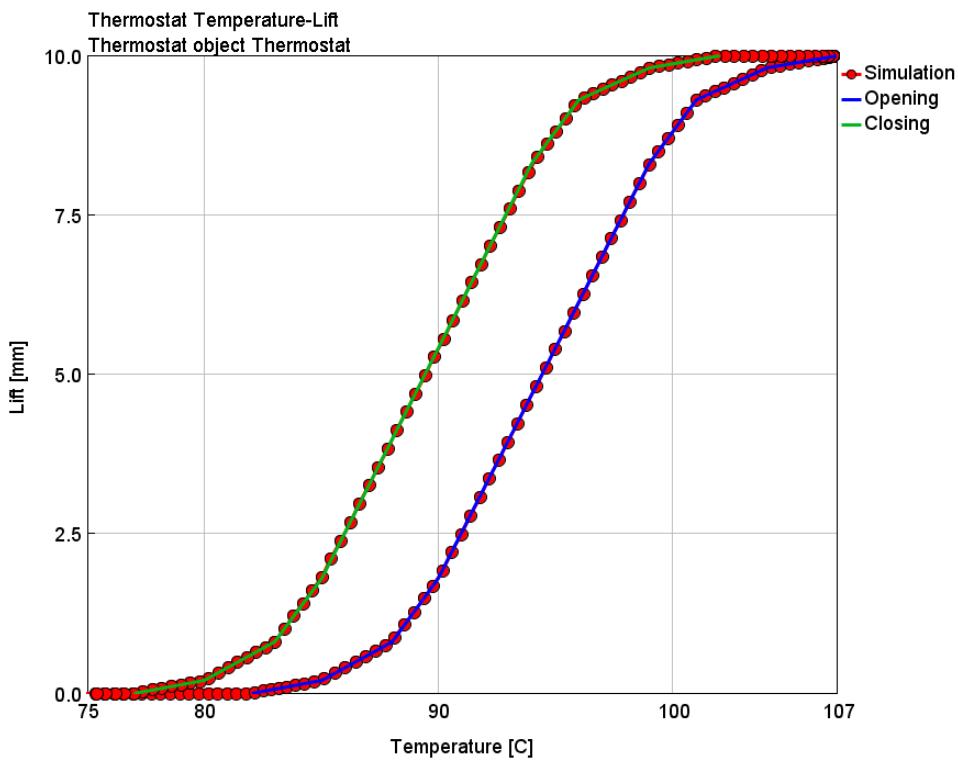
Prior to running the model, the test condition for the thermostat need to be included. Go to Case Setup and entered the test conditions as found in the *DataSheets.xlsx* file, or from your test bench measurements. Use the Value Selector to import the correct fluid type. The model is currently set up to adjust the fluid temperature imposed to follow the response of the thermostat left to ensure the profile is followed. Run the model at this time. Run Setup has already been filled out following GT recommendations for testing standalone components. After the model has finished running, open the results in GT-POST.

Main	All	+	
Parameter	Unit	Description	Case 1
Case On/Off		Check Box to Turn Case On	<input checked="" type="checkbox"/>
Case Label		Unique Text for Plot Legends	
Bypass_Diameter	mm	Bypass Pipe Reference Diameter	25 <input type="button" value="..."/>
Thermostat_Diameter	mm	Thermostat Reference Diameter	35 <input type="button" value="..."/>
Pump_Diameter	mm	Pump Reference Diameter	35 <input type="button" value="..."/>
Housing_Volume	mm^3	Thermostat Housing Volume	72500 <input type="button" value="..."/>
Coolant_Pressure	bar	Coolant Pressure	1 <input type="button" value="..."/>
Coolant_Temperature	C	Coolant Temperature	25 <input type="button" value="..."/>
Coolant_Fluid		Coolant Composition	egl-5050 <input type="button" value="..."/>

Once in GT-POST, use the RLT macro from the toolbar () to merge the Thermostat Time RLTs Temperature (X) and Lift (Y) with the preprocessed Thermostat plot that contains the thermostat temperature-lift profile. The image below was adjusted to make the operating points easier to see when plotted on top of the measured temperature-lift profile.



Tutorial 3: Modeling Components of a Coolant Circuit



A similar procedure can be done for the bypass thermostat valve as well.

The easiest way to test the pressure drop of the thermostat is to select the Impose Lift option in the 'Thermostat' part, and then impose the reference temperature in the end environment boundary conditions. After running the model again, the Case RLTs of the thermostat and thermostat bypass valve can be compared to the preprocessed pressure drop plots from 'FlowPDropThermostat'. If the thermostat is behaving correctly, then the Case RLTs should match the input data.



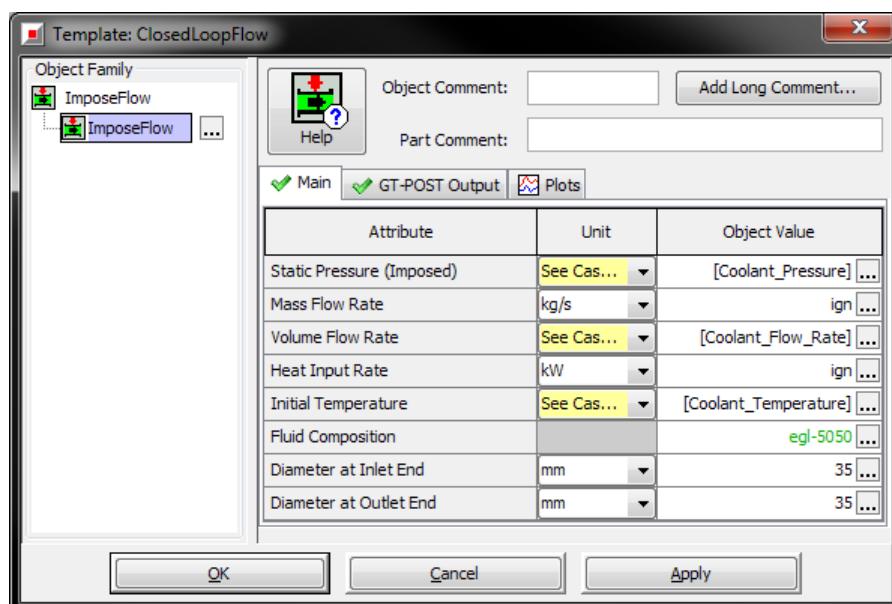
CHAPTER 4: Modeling a Hydraulic System (No Heat Transfer)

This tutorial has been prepared to provide guidelines on how to model a hydraulic system for a cooling circuit. This is typically the first test that is performed to confirm the system and the branches are seeing the correct pressure and flow distribution (i.e. system resistance) without the influence of a temperature increase from the engine. A hydraulic system requires the full cooling system to be built and connected together. However, there are two ways to run a model as a hydraulic system with no heat transfer.

- Method 1 involves spinning the coolant pump at a certain speed and then verifying the system pressure drop versus flow rate. The issue with this method is that if the system pressure drop is incorrect, then the total coolant flow rate will be incorrect as well. If the system pressure drop is incorrect, then diagnosing the model to determine the cause of the issue is fairly difficult. It also requires the coolant expansion tank to be modeled.
- Method 2 (recommended) involves replacing the pump in the model with the template 'ClosedLoopFlow'. This template mimics a closed loop system to determine the system pressure by calculating the pressure necessary to achieve a desired (imposed) flow rate. By imposing the total coolant flow rate at the location of the pump allows any issues with the pressure distribution in the model to be easily identified. Additionally, the coolant expansion tank is not required for this method, and should be removed if it exists.

4.1 Required Components for a Hydraulic System

Open the model `Hydraulic.gtm`, that can be found in the directory `..\tutorials\Modeling_Applications\Cooling_Thermal_Management\04-Hydraulic\`. The piping in this model was created by importing a CAD file of the hoses into GEM3D and then exporting the pipe volumes. Additionally, this model is already set up as a closed lookup system using the template 'ClosedLoopFlow'. This is represented by the part 'ImposeFlow' on the map (open the part). The information necessary to impose the flow rate for the cooling system is the reference pressure, flow rate, temperature, operating fluid, and the reference diameters upstream and downstream of the pump.



This information is already completed in the form of parameters, and the operating fluid will be 'egl-5050'. If the diameters are set to something different besides what is connected to/from this part, then an additional pressure drop may be included in the system because of changes in area. Click OK on the part after reviewing the entered data (nothing was changed).

Even though a heat exchanger is included in the model, the heat transfer rate will be disabled in the next section and the heat exchanger will be treated as a flow-only component. It is not required to build the full heat exchanger for this setup. If only the pressure drop for the heat exchanger is available, then a 'PressureLossConn' object can be used as a valid replacement in order to model the correct component pressure drop. However, if heat transfer will be modeled at a later time, and the heat exchanger geometry and performance data are available, then it is beneficial to include the heat exchanger because the project map will not need to be modified when heat transfer is included.

The thermostat lift will be imposed directly to control the system behavior instead of relying on the temperature to adjust it automatically. This is possible by selecting the Impose Lift attribute in the 'Thermostat' part.

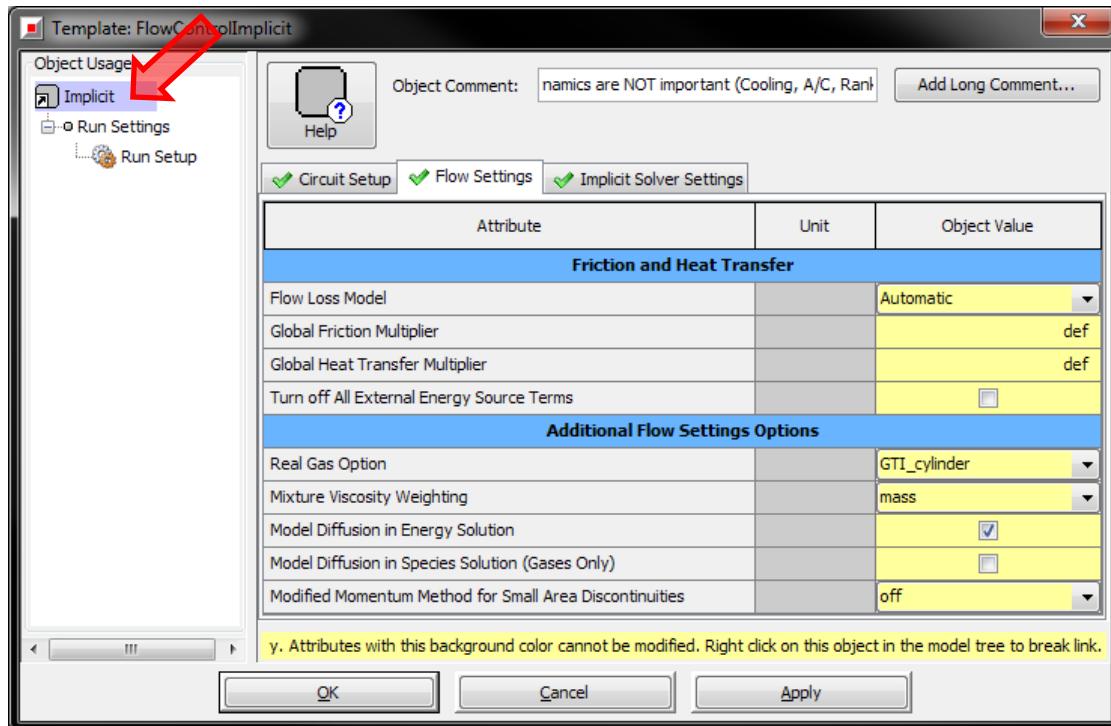
4.2 Run Setup and Disabling Heat Transfer (Hydraulic Settings)

To set the correct simulation input for a hydraulic system, open Run Setup in the **Home** tab. The **TimeControl** folder is prefilled to run for 60 seconds with Automatic Shut-Off When Steady-State turned "on". Assuming a model is built correctly, running a circuit for 60 seconds is enough time to reach steady-state when no heat transfer (or heat input) is included.

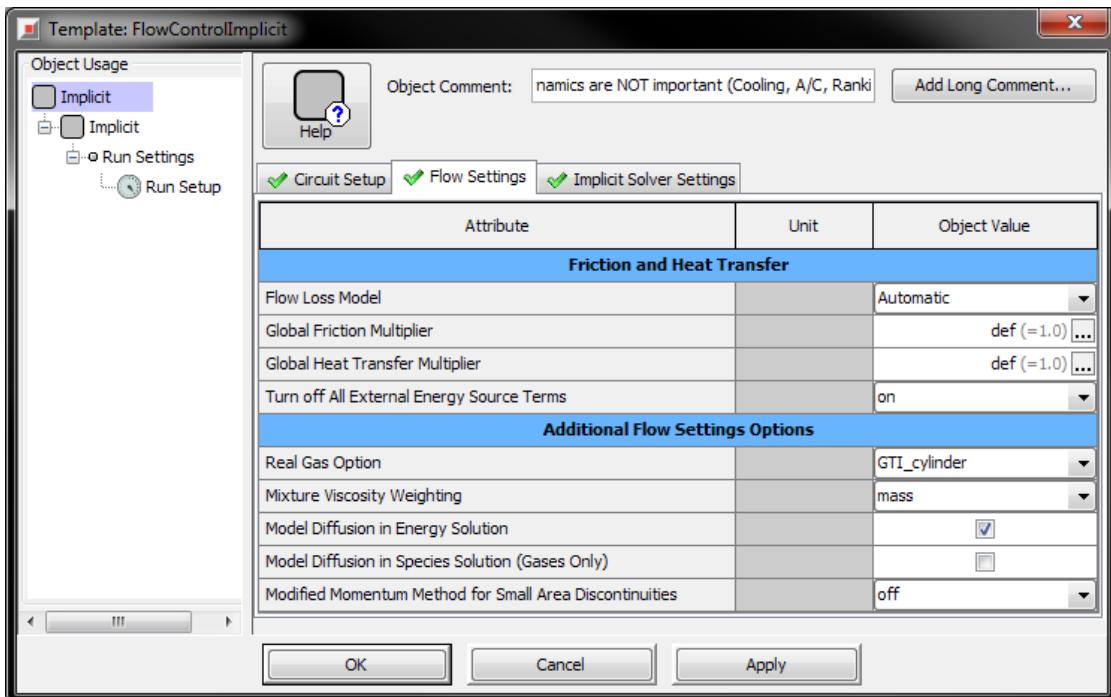
The **FlowControl** folder is where the adjustment needs to be made to model a hydraulic system. Double-click on the reference object "Implicit" for the attribute Time Step and Solution Control Object. When the object is opened, it will be marked as read-only because it is linked to the GT-SUITE library. The **Flow Settings** folder contains the attribute Global Heat Transfer Multiplier which must be set to zero in order to disable the calculations for heat transfer between the fluid and wall of the model. To make this change, right-click on the object name in the tree on the left, and select the option to **Break Implicit Link**.



Tutorial 4: Modeling a Hydraulic System (No Heat Transfer)



Once the link is broken, set the attribute **Turn off All External Energy Source Terms** to be "on", which will set all heat transfer rate terms between the fluid and wall, and as well as direct source terms, to be zero. This allows an adiabatic system to be modeled easily without any need to disable these terms individually in each part of a circuit. Click OK when finished to accept changes, and again in Run Setup.
Note: A value of "off" will enable the heat transfer term for the circuit.

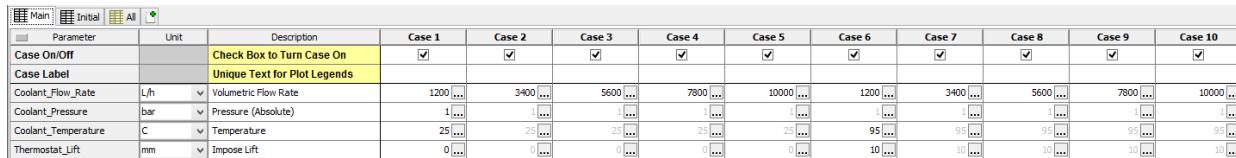


Tutorial 4: Modeling a Hydraulic System (No Heat Transfer)

Note: Even though heat transfer is disabled, it is still recommended to model both fluid sides of a heat exchanger. This allows a model to be switched easily between a thermal and hydraulic state without making changes to the parts on the project map.

4.3 Testing a Hydraulic System

Prior to running the model, the test conditions for the hydraulic system need to be set. The typical information needed to run a hydraulic system is the coolant flow rate, pressure, and temperature. For this test, a flow rate sweep is performed at two different coolant temperatures at an imposed thermostat lift. The [Coolant_Flow_Rate] is set to vary from 1200 - 10000 L/h at 2200 L/h increments, at a [Coolant_Temperature] of both 25 C and 95 C. The [Coolant_Pressure] was set to 1 bar. The [Thermostat_Lift] is imposed at 0 mm and 10 mm. Click OK on Case Setup when the operating points have been defined.



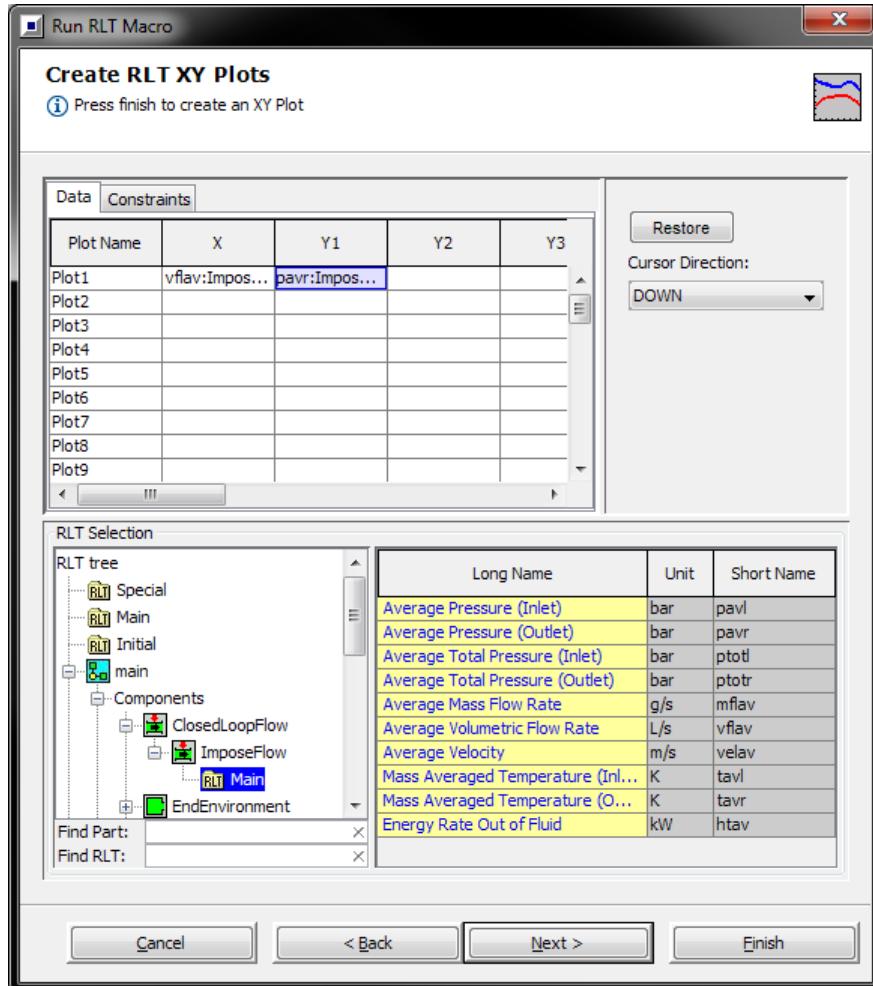
Main	Initial	All										
Parameter	Unit	Description	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8	Case 9	Case 10
Case On/Off		Check Box to Turn Case On	<input checked="" type="checkbox"/>									
Case Label		Unique Text for Plot Legends										
Coolant_Flow_Rate	L/h	Volumetric Flow Rate	1200...	3400...	5600...	7800...	10000...	1200...	3400...	5600...	7800...	10000...
Coolant_Pressure	bar	Pressure (Absolute)	1...	1...	1...	1...	1...	1...	1...	1...	1...	1...
Coolant_Temperature	C	Temperature	25...	25...	25...	25...	25...	95...	95...	95...	95...	95...
Thermostat_Lift	mm	Impose Lift	0...	0...	0...	0...	0...	10...	10...	10...	10...	10...

Run the model at this time. The simulation should be fairly quick because it is a hydraulic model with no dynamics and all heat transfer, and heat input, disabled. Open the results in GT-POST after the model has finished running.

Select the RLT macro from the toolbar () to create a plot of the system pressure drop versus flow rate using the Case RLTs at the two coolant temperatures (or thermostat lifts). Use the Volume Flow Rate (X) and the Outlet Pressure (Y) of the 'ImposeFlow' part.



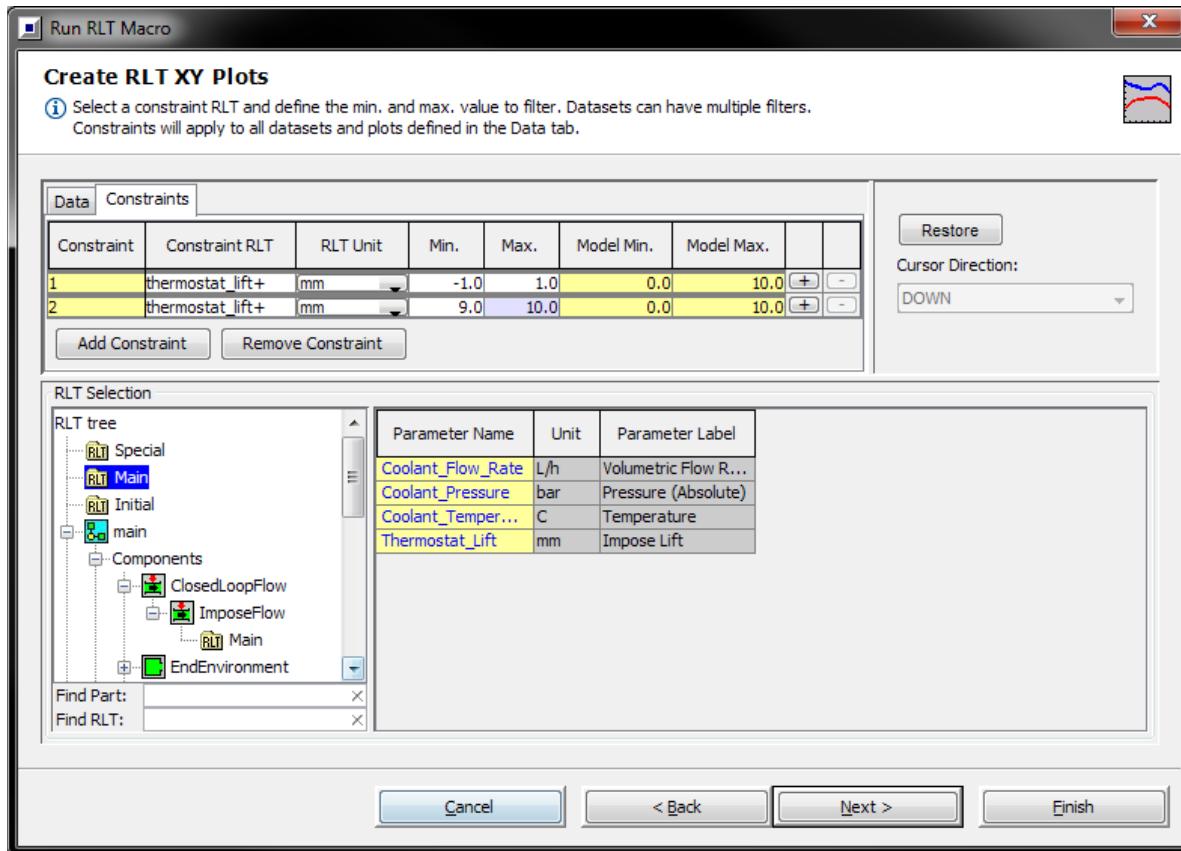
Tutorial 4: Modeling a Hydraulic System (No Heat Transfer)



Select the **Constraints** tab to organize the data sets from the simulation. Two constraints will be created using the parameter [Thermostat_Lift] found in the Main folder. The Min/Max for the first constraint should be -1/1, respectively, for the thermostat closed condition. The Min/Max for the second constraint should be 9/10 C, respectively, for the thermostat open condition. Click Finish when the constraints have been filled in.



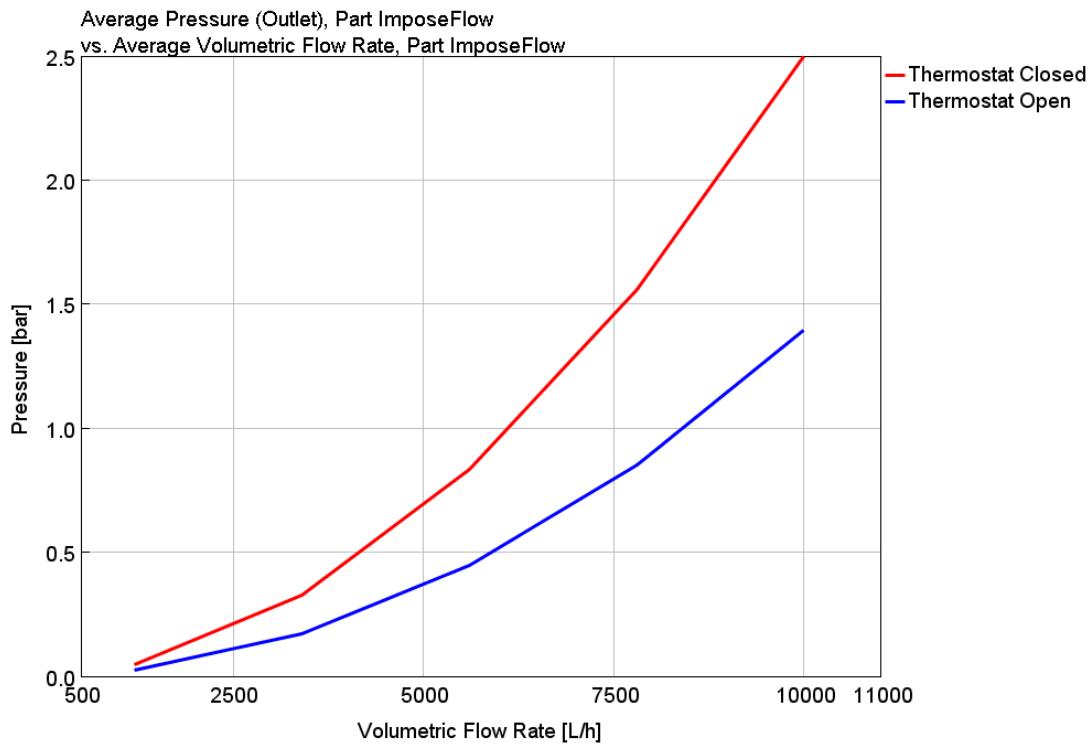
Tutorial 4: Modeling a Hydraulic System (No Heat Transfer)



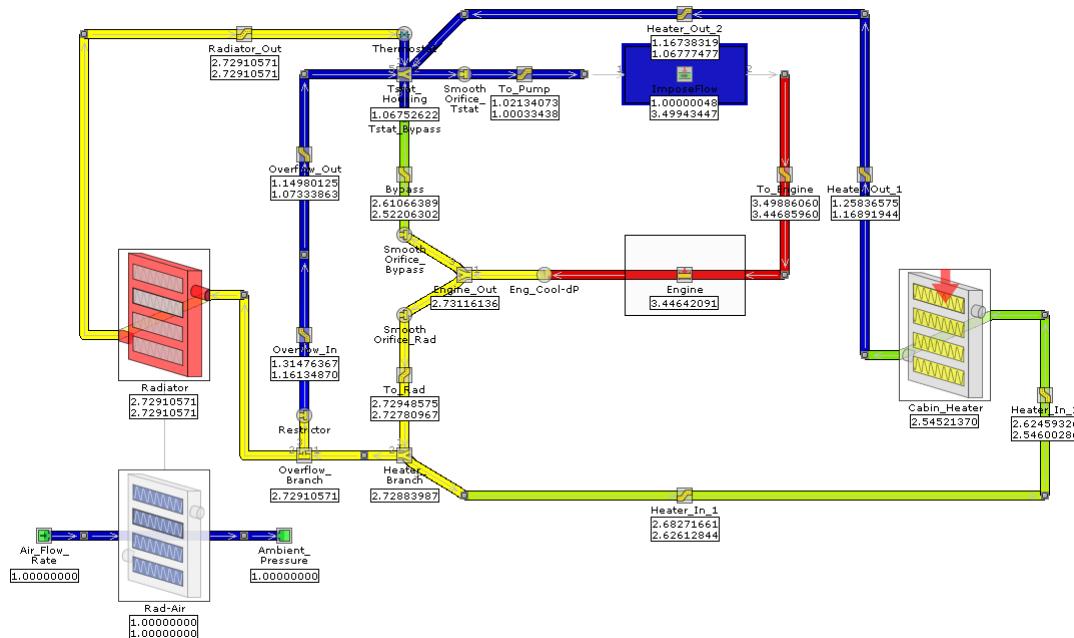
Upon clicking Finish, a new .gu and new plot will be automatically generated using the Case RLTs selected and the constraints. However, the Case RLT selected for the pressure (Y-axis) was the outlet pressure. This data set needs to be shifted by the reference pressure in order to get the correct system pressure drop versus flow rate plot. Double-click on each data set created, and in the **Operation** tree set the Y shift to -1 bar (1 bar was the reference pressure used for the test conditions). This will shift the data sets by -1 bar so that the plot now displays the system pressure drop versus flow rate at the thermostat closed and open lift positions. *Note: The plot displayed below has had the plot axis label, plot title, and dataset names modified.*



Tutorial 4: Modeling a Hydraulic System (No Heat Transfer)



This is a common study that is performed to understand the behavior of the cooling system in order to ensure various parts of the system receive adequate flow. The plot created can be used to compare to measured results of the system pressure drop. Additionally, the RLT Contour mode can be used to visualize the flow rate and/or pressure distribution in the circuit. The image below shows the pressure distribution (bar) in the circuit for Case 5.



CHAPTER 5: Modeling a Steady-State Thermal System

This tutorial has been prepared to provide guidelines on how to model a cooling system with heat transfer. If a hydraulic test has not been performed, then it is recommended to follow the steps in CHAPTER 4: Modeling a Hydraulic System (No Heat Transfer) to set up the model to run without heat transfer to ensure the correct system pressure drop. If a hydraulic system has been modeled, and the system pressure drop, flow distribution, and pressure distribution are all correct, then heat transfer (heat input) can be included in the circuit to study the steady-state temperature distribution in the system.

When modeling a circuit with a constant heat rejection rate at the engine to investigate the steady-state temperature distribution, it is recommended to use the 'HeatAddtion' template instead of an 'EngineBlock*' template since the coolant heat rejection can be directly imposed (this is a common test bench measurement). The coolant heat rejection rate can be directly imposed for known test cases, or set up as a map that is a function of engine speed and load. Both are viable methods, and it comes down to personal preference. This tutorial will show how to set up the coolant heat rejection rate as a map in order to show how to create a dependent RLT lookup.

There are three methods of setting the flow circuit to run when including heat transfer.

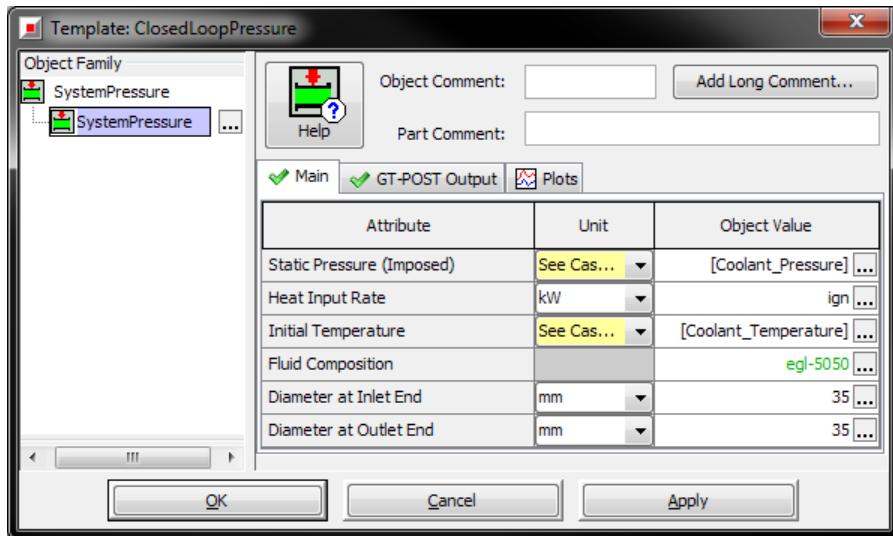
- Method 1 involves using the same setup as the hydraulic system model where there is a 'ClosedLoopFlow' template at the location of the pump to impose the total system flow rate. The coolant expansion tank should not be included with this method. This is a valid method, but it will not be used in this tutorial.
- Method 2 involves using the template 'ClosedLoopPressure' to impose the system pressure along with the coolant pump. This method is recommended if a coolant heat rejection map will be used because the engine speed will already be required. This allows the coolant flow rate to be automatically calculated instead of imposed, while controlling the system pressure in case there is an issue when introducing heat transfer to the system. The coolant expansion tank should not be included with this method. This is the method that will be used in this tutorial.
- Method 3 involves creating a closed loop system that involves the coolant pump. There will be no 'ClosedLoopFlow' or 'ClosedLoopPressure' templates included. This method is not recommended because if there is an issue with the heat transfer rate in the system then the system pressure may go out of control. It will then be difficult to diagnose the model to find the location of the issue. If this method is selected, then the coolant expansion tank must be modeled so the system pressure can be regulated.

5.1 Required Components for a Steady-State Thermal System

Open the model HeatTransfer.gtm, that can be found in the directory ..\tutorials\Modeling_Applications\Cooling_Thermal_Management\05-HeatTransfer\. This model is already set up as a closed lookup system using the template 'ClosedLoopPressure', and the pump that was created in 3.1 Modeling a Pump. The 'ClosedLoopPressure' is represented by the part 'SystemPressure' on the map (open the part). The information necessary to control the system pressure is only the reference pressure, operating fluid, and the reference diameters upstream at this location (i.e. pump reference diameter).

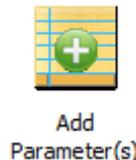


Tutorial 5: Modeling a Steady-State Thermal System



This information is already completed in the form of parameters, and the operating fluid will be 'egl-5050'. If the diameters are set to something besides what is connected to/from this part, then an additional pressure drop may be included in the system because of changes in area. Click OK on the part after reviewing the entered data (nothing was changed).

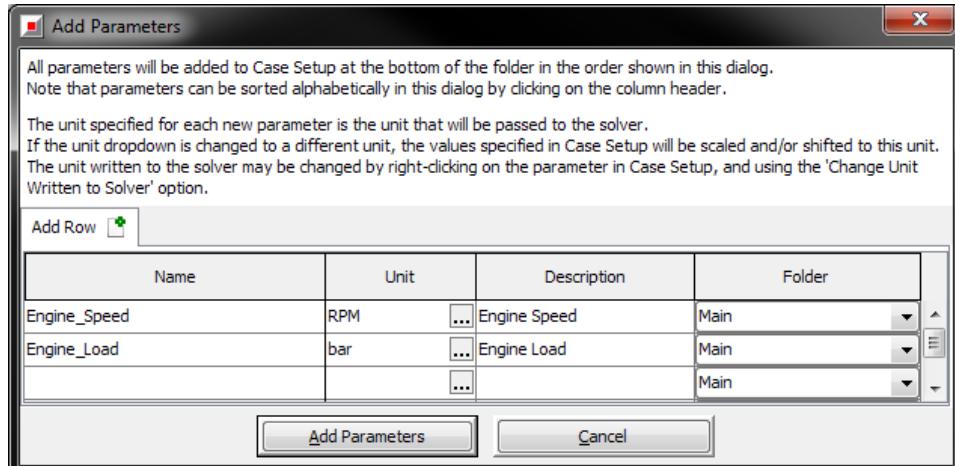
To create a coolant heat rejection map that is dependent on engine speed and load, the input parameters for engine speed and load must first be created. To do this, open Case Setup. To declare a parameter that does not yet exist, click on the button call Add Parameter(s).



This will open a new dialog where a parameter name and unit can be defined. Create the parameter Engine_Speed (the brackets, [], are not required) with the units of RPM (Angular Velocity). Create a second parameter called Engine_Load with the units of bar (Pressure). Click Add Parameters when finished to declare the parameter for engine speed and load.



Tutorial 5: Modeling a Steady-State Thermal System



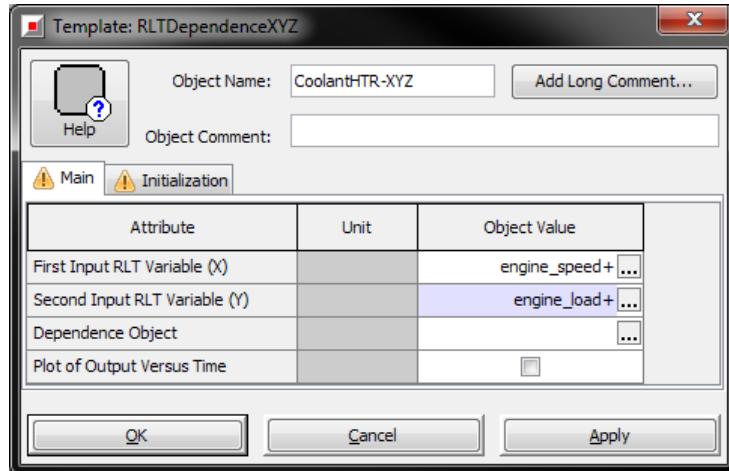
Parameter	Unit	Description	Case 1
Case On/Off		Check Box to Turn Case On	<input checked="" type="checkbox"/>
Case Label		Unique Text for Plot Legends	
Coolant_Temperature	C	Temperature	[...]
Air_Flow_Rate	kg/s	Mass Flow Rate / Air scfm	[...]
Ambient_Temperature	C	External Convection Temperature	[...]
Cabin_Heater_Rate	W	Heat Input Rate	[...]
Engine_Speed	RPM	Engine Speed	
Engine_Load	bar	Engine Load	
HeatRej_Block	kW	Heat Input Rate	[...]
Pump_Speed	RPM	Pump Speed	[...]

Click OK on Case Setup when finished.

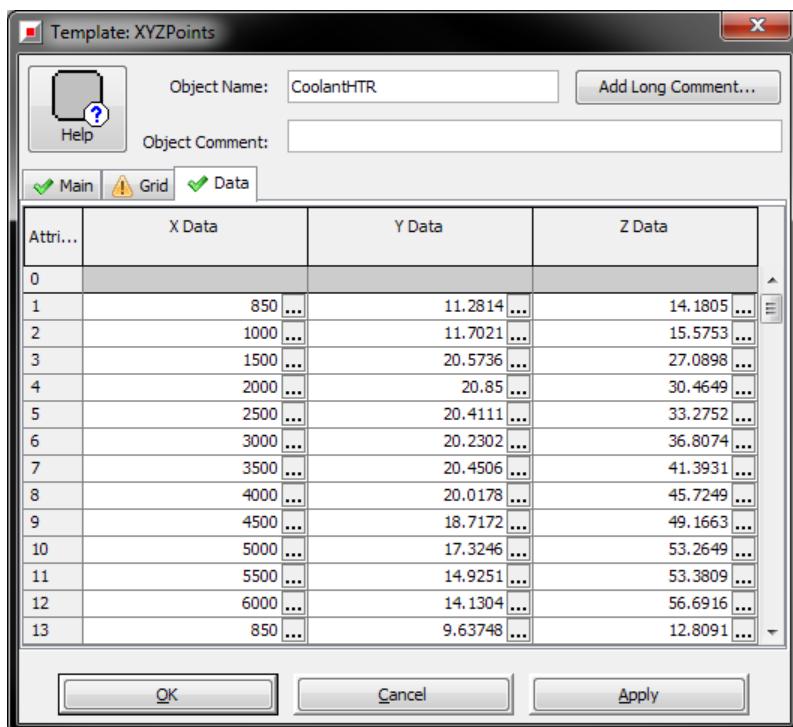
Open the part 'Engine' on the project map (a 'HeatAddition' template). The attribute Heat Input Rate is currently set as a parameter in case the coolant heat rejection rate would be modeled as a constant value that varies from cases to case. Because a map will be used instead, use the Value Selector to select the 'RLTDependenceXYZ' template. This template will allow engine speed (X) and engine load (Y) to be used to calculate the coolant heat rejection (Z). Use the Value Selector to select the engine speed parameter for the attribute First Input RLT Variable (X), and again to select the engine load parameter for the attribute Seconds Input RLT Variable (Y). The parameters can be found in the Main folder of the Parameters section.



Tutorial 5: Modeling a Steady-State Thermal System



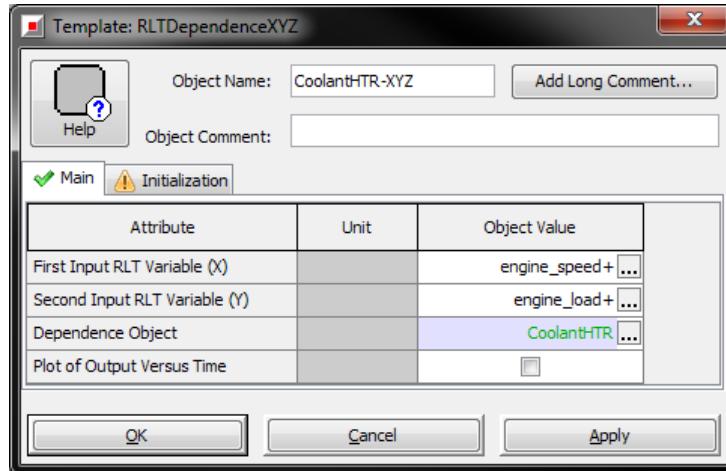
Use the Value Selector for the Dependence Object attribute and select the 'XYZPoints' option. Copy the heat rejection map data found in the file *DataSheets.xlsx* in the directory ..\tutorials\Modeling_Applications\Cooling_Thermal_Management\.



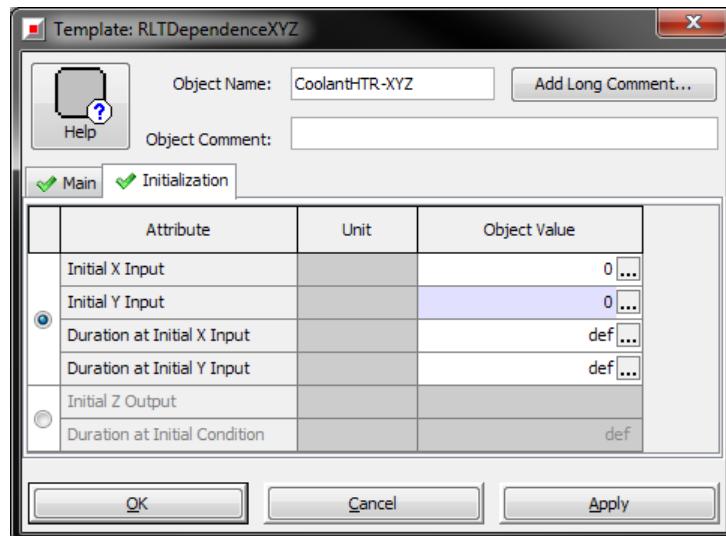
In the **Grid** folder of the object, select the radio button for the Internally Generate Y-Boundaries. This will automatically create the bounds for the map data so a normalized grid can be used for the lookup. Click OK when finished.



Tutorial 5: Modeling a Steady-State Thermal System

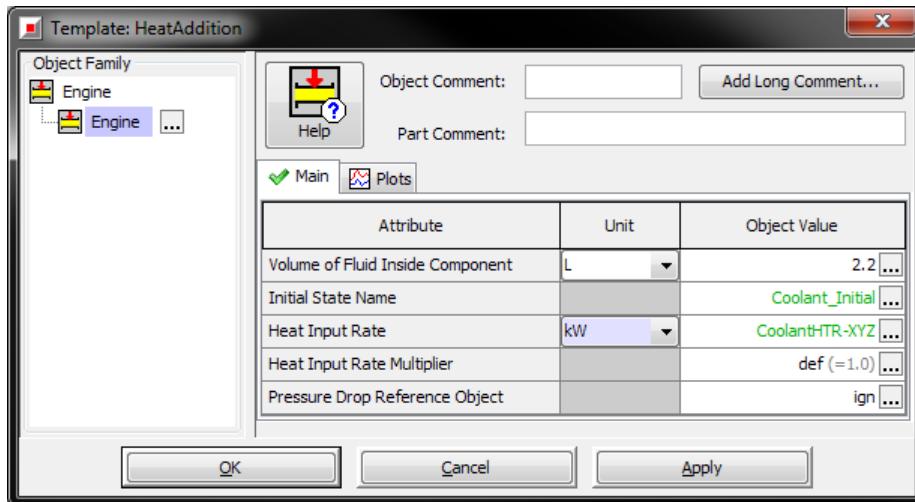


Back in the 'RLTDependenceXYZ' object, set the Initial X Input and Initial Y Input to 0 in the **Initialization** folder. Click OK when finished to complete the coolant heat rejection map.



Prior to clicking OK in the 'Engine' part to accept all changes, be sure to set the units for the Heat Input Rate to kW.





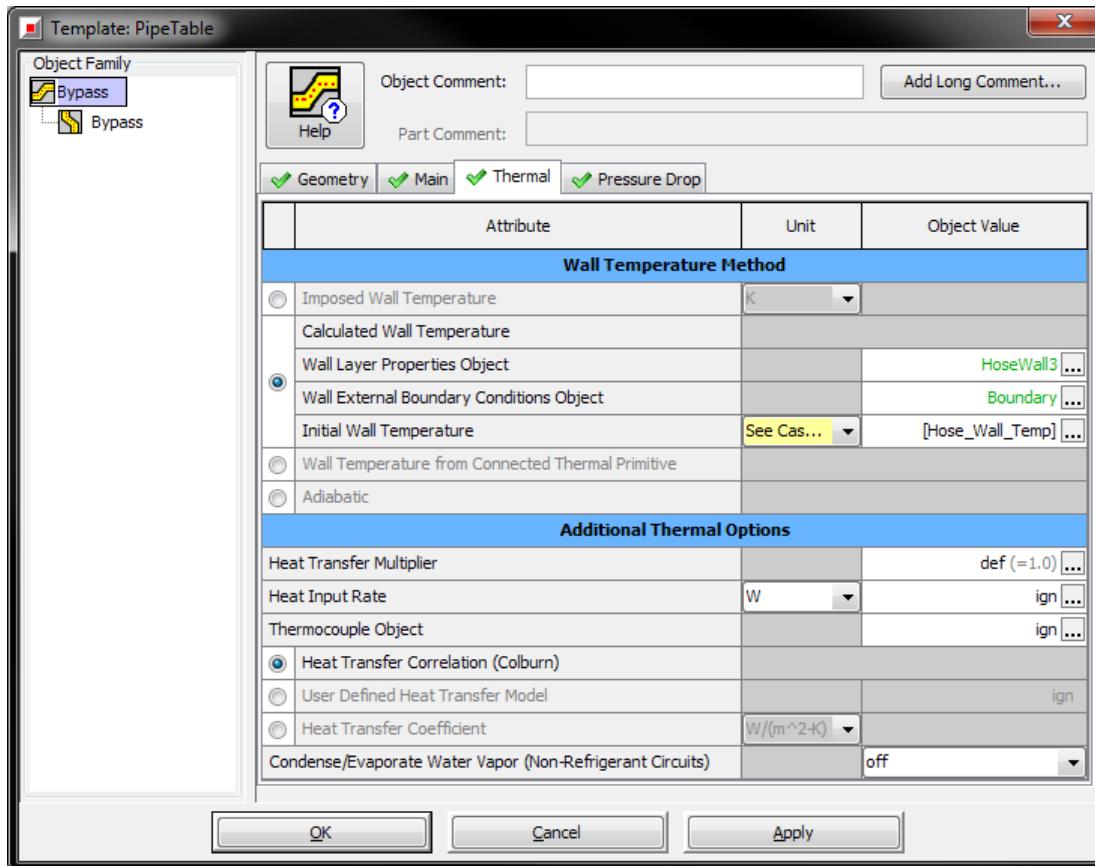
If coming from a hydraulic model where there was no heat exchanger and the pressure drop was modeled with a 'PressureLossConn', then the 'PressureLossConn' should be removed and replaced with a heat exchanger. This will be true for any heat transfer related component (i.e. 'HxMaster', 'HxSlave', 'HeatAddition', etc) where a significant exchange of energy will take place that can affect the simulation results.

5.2 Heat Transfer in Pipes (Wall Temperature Solver)

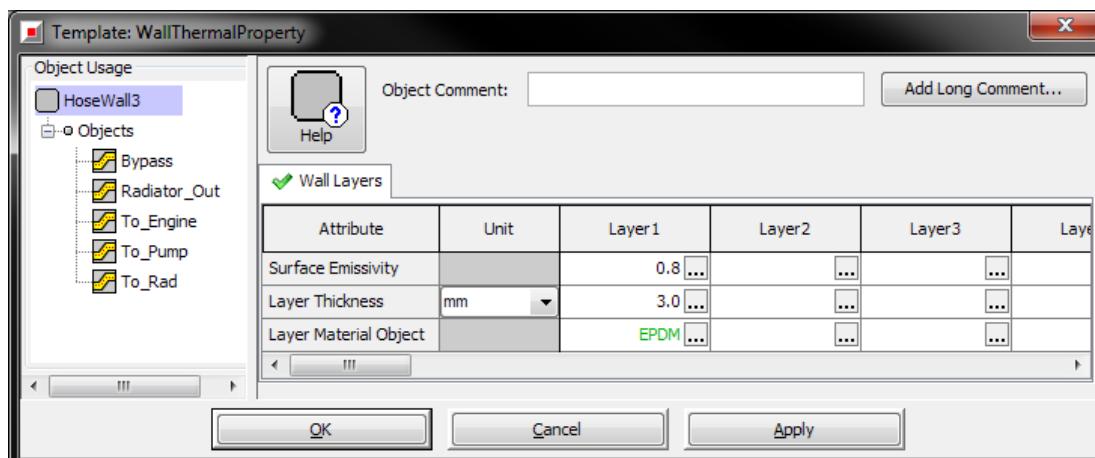
To model the thermal behavior in pipes, the correct Wall Temperature Method should be selected in all 'Pipe*' and FlowSplit*' parts. Open the part called 'Bypass'. The **Thermal** folder contains the different options for the Wall Temperature Method. It is common for a model to simply use the Imposed Wall Temperature or Adiabatic options to model the wall temperature. If attempting to achieve the correct heat balance in a system, it is recommended to use the Calculated Wall Temperature option. This option allows a user to model the material, thickness, and external boundary conditions of the pipe or flow split. The external boundary conditions can be similar to the conditions used in a test lab or on the road.



Tutorial 5: Modeling a Steady-State Thermal System

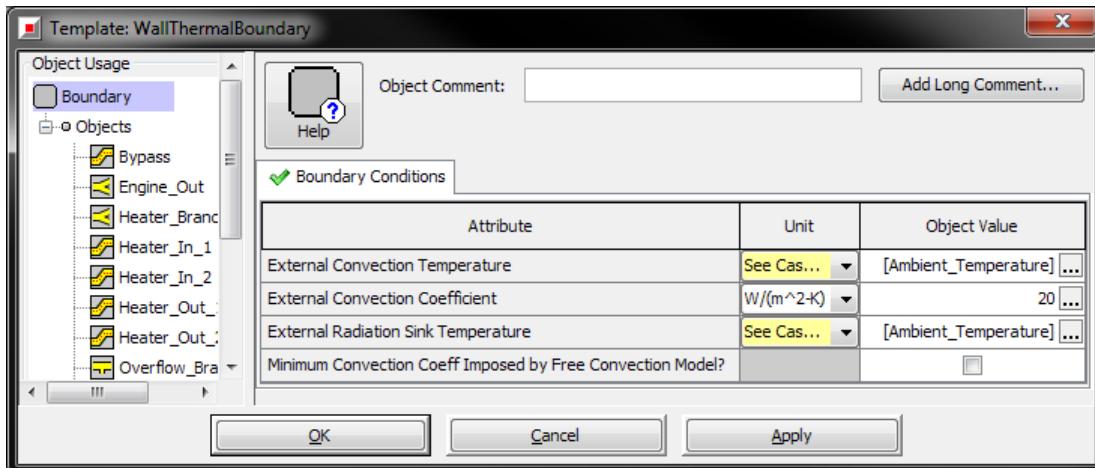


Double-click on the Wall Layer Properties Object reference called 'HoseWall3'. This reference object models the geometry (thickness and material) of the pipe. Each column in the **Wall Layers** folders can be used to model a different layer of the pipe or flow split wall (i.e. an air gap between two solid materials). Click OK when finished (nothing was changed).



The wall boundary conditions for the pipe are accessible from the Wall External Boundary Conditions Object reference called 'Boundary'. The external temperature and convection coefficient of the pipe are set here. Click OK when finished (nothing was changed).





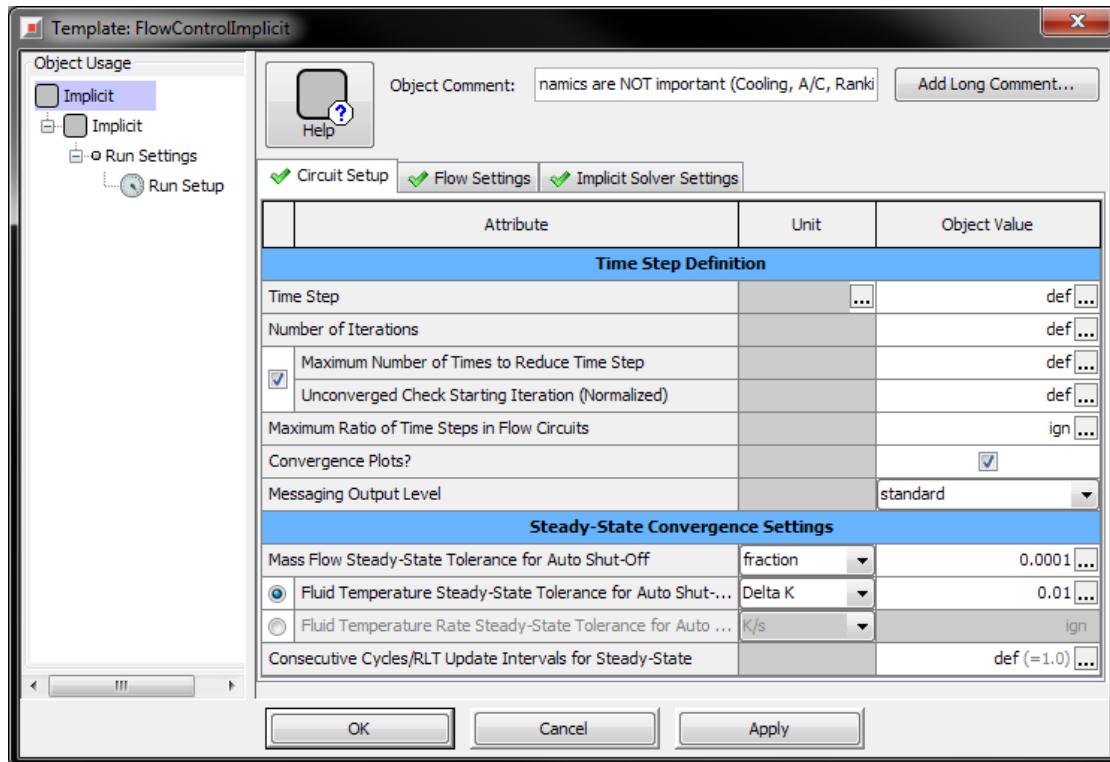
Each pipe and flow split in a flow circuit model can use the same Wall Layer Properties Object and Wall External Boundary Conditions Object, or a unique one. All pipes and flow splits in the model of this tutorial are using the Calculated Wall Temperature option. More information about the wall temperature solution can be found in section **2.1.5 Wall Thermal Solution (Reference Object)** in the Flow Theory manual (File→Manuals→Modeling_Theory).

5.3 Run Setup and Enabling Heat Transfer

To set the correct simulation input for a thermal system, open Run Setup which can be found in the **Home** tab. The **TimeControl** folder is prefilled to run for 300 seconds with Automatic Shut-Off When Steady-State turned "on", which is enough time for this model to reach steady-state. However, the simulation duration can vary from one model to the next when including heat transfer. Much of it has to do with the system volume. The larger the system volume, more time is required for a steady-state solution to be met. The initial temperature for the system can also play a role as well. A value of "on" for the Automatic Shut-Off When Steady-State forces the Thermal Solver in the **ThermalControl** folder to be "steady", which will ignore the inertia of all solid masses in the model. This can help some models reach steady-state faster since the masses will achieve steady-state almost instantaneously based on the neighboring temperature boundary conditions.

The **FlowControl** folder is where the settings need to be confirmed to ensure heat transfer will be included in the model. Unlike the settings for the hydraulic model, the attribute **Turn off All External Energy Source Term (Flow Settings folder)** for the flow circuit that is found in the reference object for the attribute **Time Step and Solution Control Object** should be set to "off". Click OK when finished to accept the changes, and again in Run Setup.





Enabling the External Energy Source Terms will allow heat transfer to be modeled between the fluid and wall of all pipe components in the model. Without it, the heat input source at the location of the 'Engine' will cause the fluid temperature to increase out of control because no heat can be removed at the location of the heat exchanger.

5.4 Testing a Thermal System

Testing a thermal system is no different than testing a hydraulic system. The test conditions for the system need to be defined in the model, which can be created in Case Setup. The test conditions selected may mimic a test bench setup where the engine speed and load will vary with a constant air flow rate across the radiator. The [Coolant_Temperature] is set to 75 C only to speed up the simulation run time. The value selected is set just before the thermostat starts to open, so the simulation will be treated as a warm-up condition. The initial coolant temperature is important for a thermostat because of the hysteresis. Setting the coolant temperature to a value such that the thermostat is already open may result in the incorrect lift, and consequently the incorrect heat balance on the system, because the thermostat lift responds in the wrong direction.



Tutorial 5: Modeling a Steady-State Thermal System

Main	Initial	All				
Parameter	Unit	Description	Case 1	Case 2	Case 3	Case 4
Case On/Off		Check Box to Turn Case On	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Case Label		Unique Text for Plot Legends				
Coolant_Temperature	C	Temperature	75 ...	75 ...	75 ...	75 ...
Air_Flow_Rate	kg/s	Mass Flow Rate / Air scfm	0.5 ...	0.5 ...	0.5 ...	0.5 ...
Ambient_Temperature	C	External Convection Temperature	25 ...	25 ...	25 ...	25 ...
Cabin_Heater_Rate	W	Heat Input Rate	0 ...	0 ...	0 ...	0 ...
Engine_Speed	RPM	Engine Speed	1650 ...	2700 ...	3300 ...	2600 ...
Engine_Load	bar	Engine Load	3.5 ...	5.6 ...	4 ...	6.5 ...
Pump_Speed	RPM	Pump Speed

The [Pump_Speed] can be set as a dependent of the engine speed with a gear ratio of 1.3. To do this, start by typing an equation for [Pump_Speed] that multiplies the [Engine_Speed] by 1.3 ($=1.3 * [Engine_Speed]$). Click OK on Case Setup when the run conditions are completed.

Main	Initial	All				
Parameter	Unit	Description	Case 1	Case 2	Case 3	Case 4
Case On/Off		Check Box to Turn Case On	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Case Label		Unique Text for Plot Legends				
Coolant_Temperature	C	Temperature	75 ...	75 ...	75 ...	75 ...
Air_Flow_Rate	kg/s	Mass Flow Rate / Air scfm	0.5 ...	0.5 ...	0.5 ...	0.5 ...
Ambient_Temperature	C	External Convection Temperature	25 ...	25 ...	25 ...	25 ...
Cabin_Heater_Rate	W	Heat Input Rate	0 ...	0 ...	0 ...	0 ...
Engine_Speed	RPM	Engine Speed	1650 ...	2700 ...	3300 ...	2600 ...
Engine_Load	bar	Engine Load	3.5 ...	5.6 ...	4 ...	6.5 ...
Pump_Speed	RPM	Pump Speed	=1.3*[Engine_Speed]	3510 ...	4290 ...	3380 ...

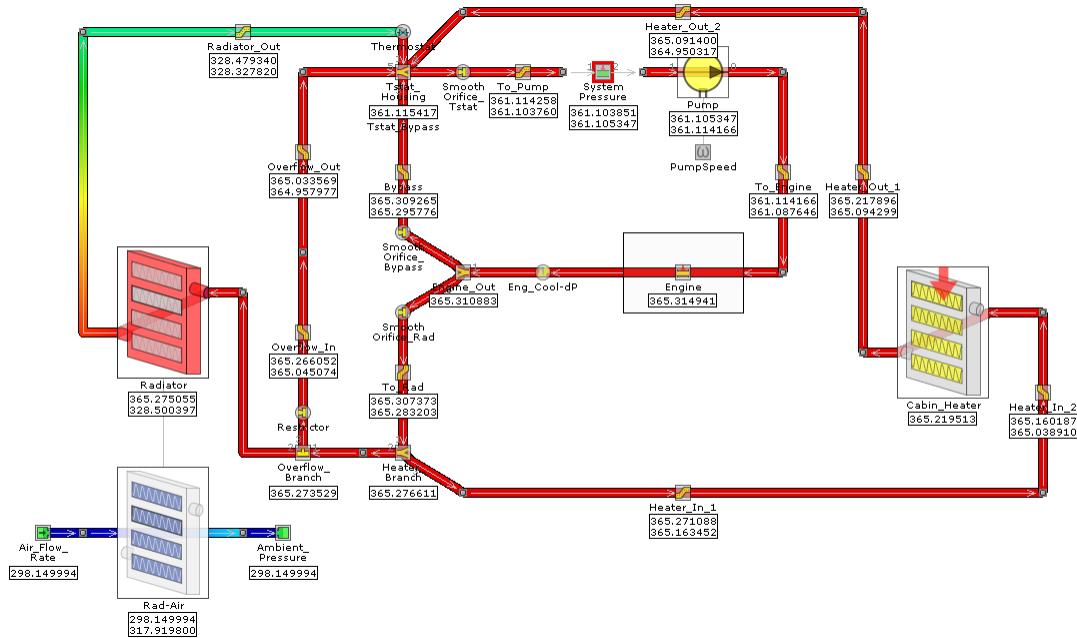
Main	Initial	All				
Parameter	Unit	Description	Case 1	Case 2	Case 3	Case 4
Case On/Off		Check Box to Turn Case On	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Case Label		Unique Text for Plot Legends				
Coolant_Temperature	C	Temperature	75 ...	75 ...	75 ...	75 ...
Air_Flow_Rate	kg/s	Mass Flow Rate / Air scfm	0.5 ...	0.5 ...	0.5 ...	0.5 ...
Ambient_Temperature	C	External Convection Temperature	25 ...	25 ...	25 ...	25 ...
Cabin_Heater_Rate	W	Heat Input Rate	0 ...	0 ...	0 ...	0 ...
Engine_Speed	RPM	Engine Speed	1650 ...	2700 ...	3300 ...	2600 ...
Engine_Load	bar	Engine Load	3.5 ...	5.6 ...	4 ...	6.5 ...
Pump_Speed	RPM	Pump Speed	2145 ...	3510 ...	4290 ...	3380 ...

Run the model at this time. The simulation may take a little longer to run when compared to the speed of a hydraulic system, but that is because the temperature of the coolant is changing to reach a steady-state value based on the heat balance of the system that includes the heat input from the engine, the heat rejection from the radiator, and the heat loss in all of the pipes. Open the results in GT-POST after the model has finished running.

The most common results to view can be seen in the RLT Contour mode. This will either be the mass-averaged temperature distribution in the cooling circuit, or the net heat transfer rate. The image below shows the temperature distribution (C) in the circuit. This can be used to compare to test measurements to ensure the steady-state temperature distribution is correct.



Tutorial 5: Modeling a Steady-State Thermal System



The sensitivity of the thermostat lift as it responds to changes in temperature can play a role in the final steady-state result. If the temperature distribution does not match measured results, the thermostat lift should be compared as well since it can affect the flow distribution in the circuit.



CHAPTER 6: Modeling an Underhood System with COOL3D

This tutorial has been prepared to assist new users of GT-SUITE for underhood model building using COOL3D. COOL3D is designed to provide the user a 3D building tool to solve the air flow and thermal distribution in underhood flow caused by heat exchanger stacking in order to understand the effects they have on heat exchanger performance and, consequently, cooling system design. COOL3D models are treated as an external file, and require the use of a "main" model built in GT-ISE that reference the 3D (.ghx) file.

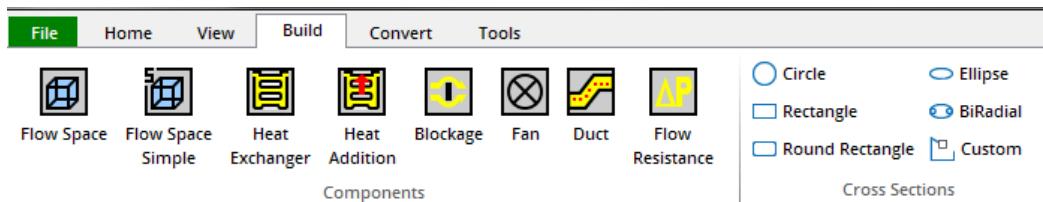
6.1 Getting Started

Launch COOL3D: If working on a PC, the graphical user interface for COOL3D can be started by double-clicking on the COOL3D icon. If an icon has not been created, one can map a new icon by pointing to \$GTIHOME\Vx.x.x*\ GTsuite\bin\win32\cool3d.exe. From UNIX or PC, one can also launch the program by typing *cool3d* at the command line.

If GT-ISE is already open, COOL3D can also be launched by going to the **Tools** tab, and then selecting COOL3D from the **GT Applications** dropdown.

6.2 Introduction to the Modeling Environment

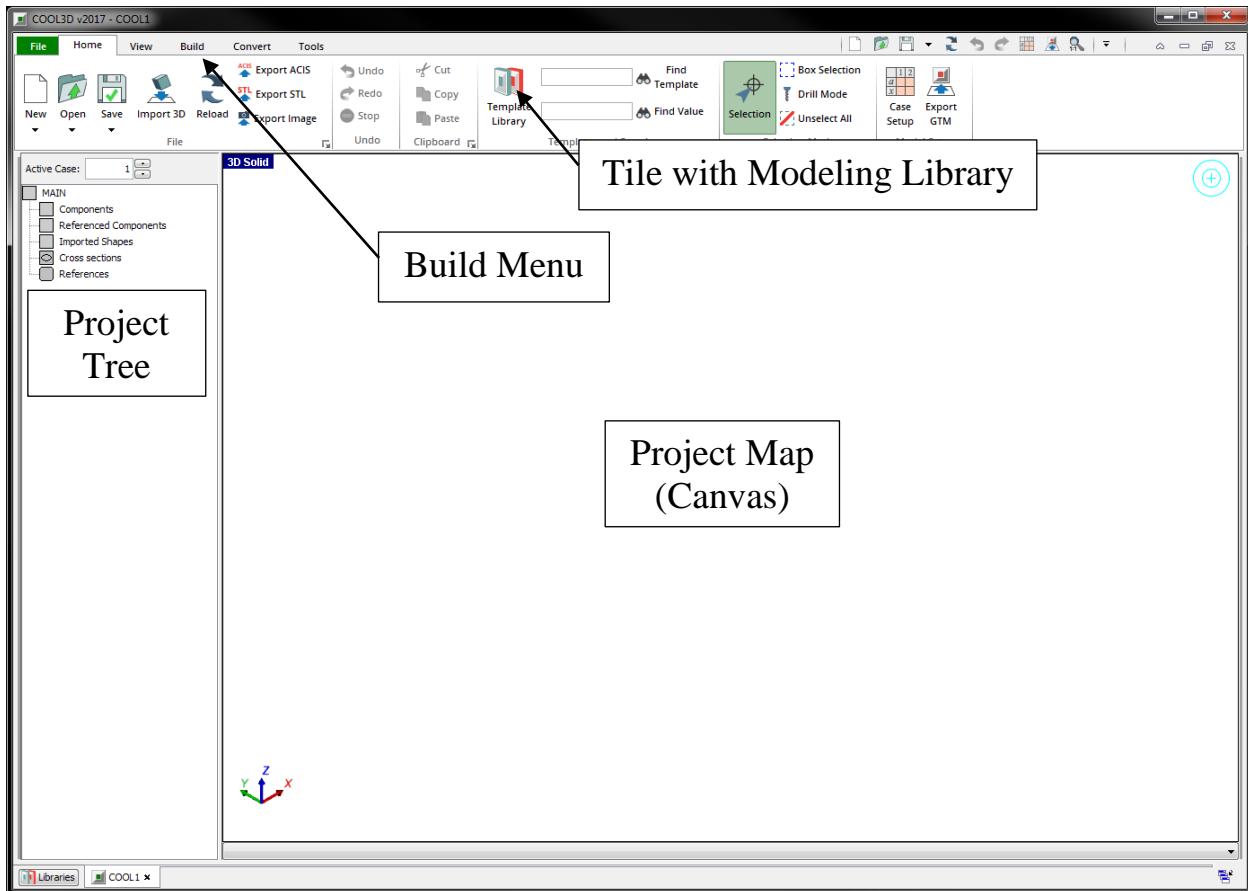
Once the program has started, a blank project map will be created. From the **Home** menu, select **New** to start a new project file. Unlike GT-ISE, to build components (objects) for the project they must be selected from the **Build** menu. The object choices are a flow space, heat exchanger, heat addition, blockage, fan, duct, and flow resistance, respectively.



When components are completed (built) they will be displayed in the project map (canvas) as a 3D element, and the Template/Object tree structure will appear in the Project Tree.



Tutorial 6: Modeling an Underhood System with COOL3D



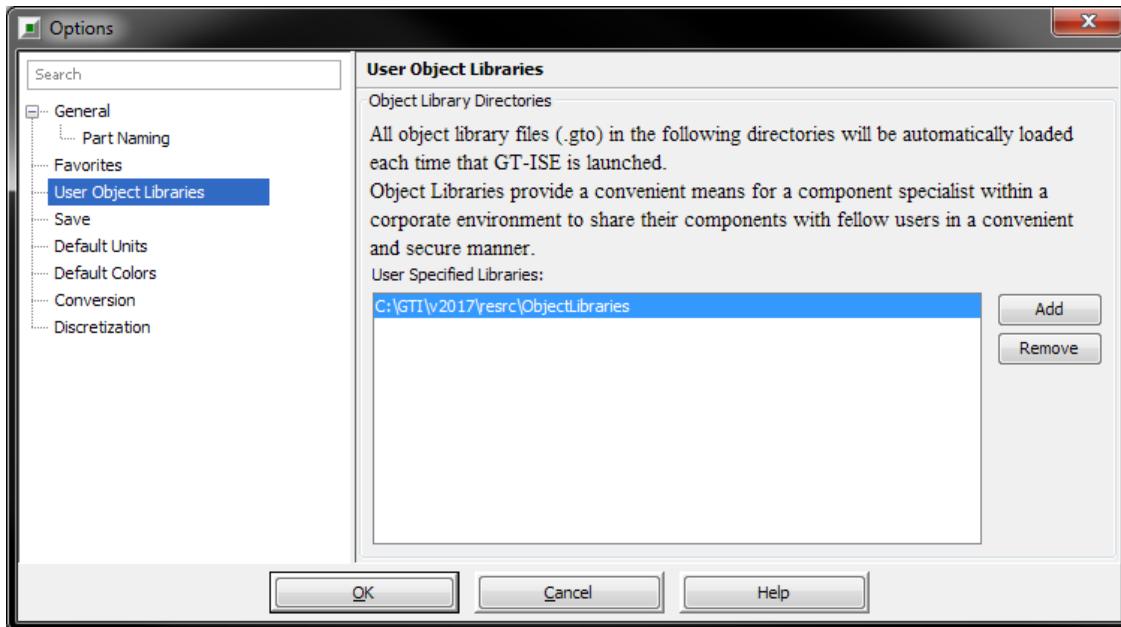
Similar to GT-ISE, there is access to the GT-SUITE library to import saved data such as lookup maps, or component geometry (i.e. heat exchangers).

6.3 Modeling a Cooling Package

6.3.1 Linking to an Object Library

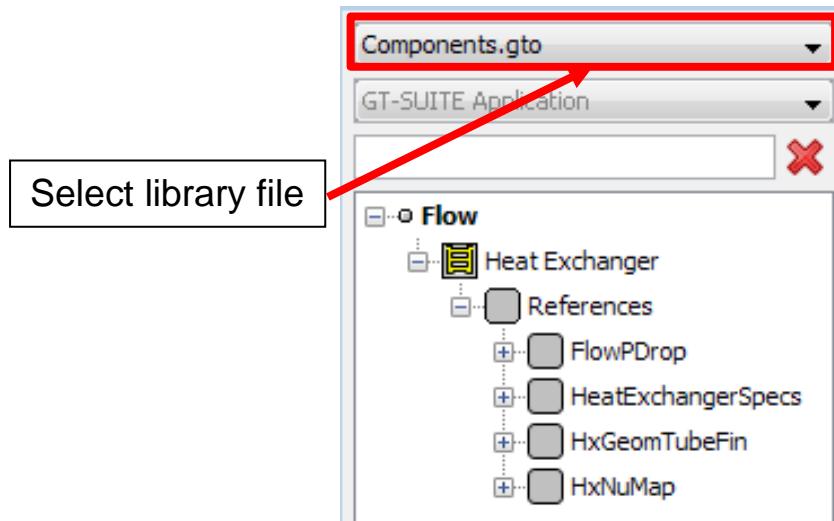
Before building components, an object library will be referenced in order to reuse some of the component geometry that was built in the previous tutorials, namely the heat exchanger. Go to **File → Options** to launch the Options dialog. The **User Object Libraries** section can be used to point to custom library files (.gto) that may contain component data to be shared among multiple files.





Click on the Add button to point to the directory where the .gto object *Components.gto* is found, which will be ..\tutorials\Modeling_Applications\Cooling_Thermal_Management\. This object library contains the information for two heat exchangers, which will be used in the next section. This ability is useful when attempting to keep a single source for all component data that may be commonly used. Click OK on the Options dialog once the directory is selected to accept all modifications.

Open the GT-SUITE library, and select the *Components.gto* from the library dropdown to see the information to build a heat exchanger.



6.3.2 Building a Heat Exchanger

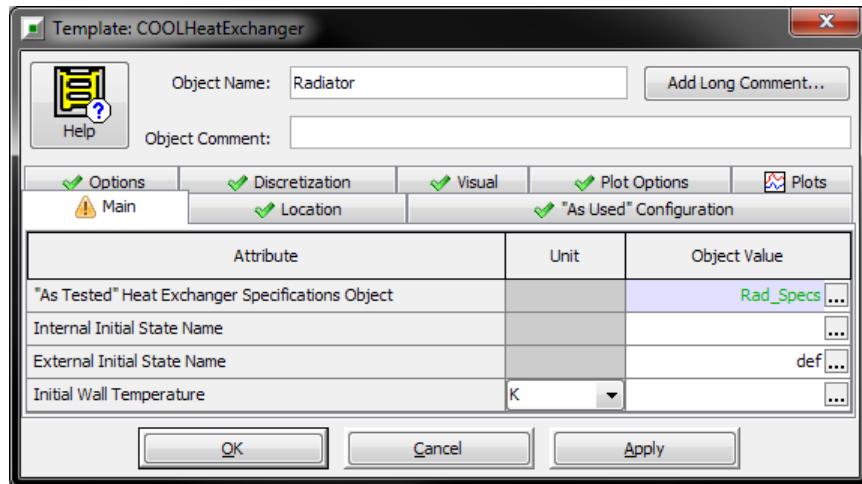
It is recommended to complete section 3.3 Modeling a Heat Exchanger first before continuing with this tutorial.



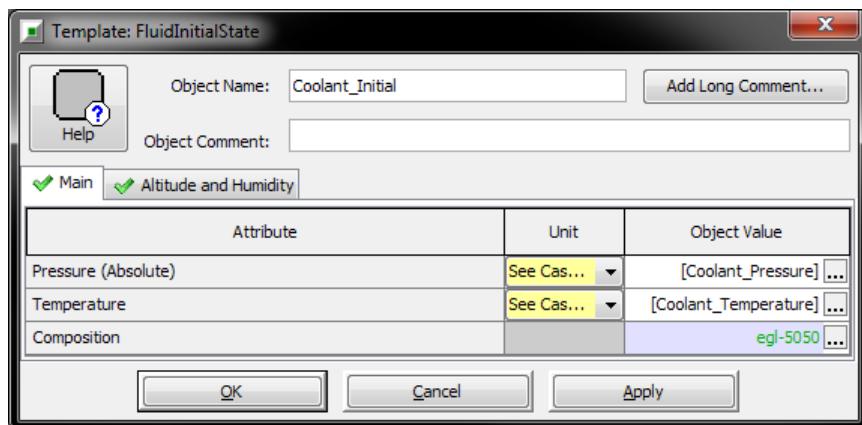
Tutorial 6: Modeling an Underhood System with COOL3D

Building a heat exchanger in COOL3D is almost identical to building a heat exchanger in GT-ISE. The only difference in the component building process is the requirement of a position in 3D space. To build a heat exchanger, select the heat exchanger icon (■) from the **Build** menu.

The **Main** folder is where the geometry and initial conditions are defined. Use the Value Selector for the "As Tested" Heat Exchanger Specification Object to import the geometry and performance data for the Radiator ('Rad_Specs') found in the *Components.gto*.



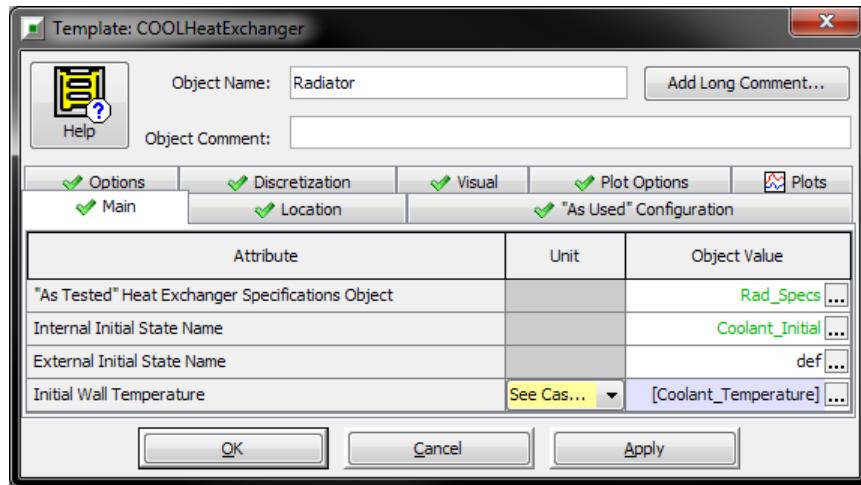
Like 'HxMaster', an initial state name for the fluid is required. Using 'FluidInitialState', define an Internal Initial State Name for the heat exchanger. Parameters were used for the pressure and temperature attributes such that they can be reused later, and 'egl-5050' is the initial fluid composition.



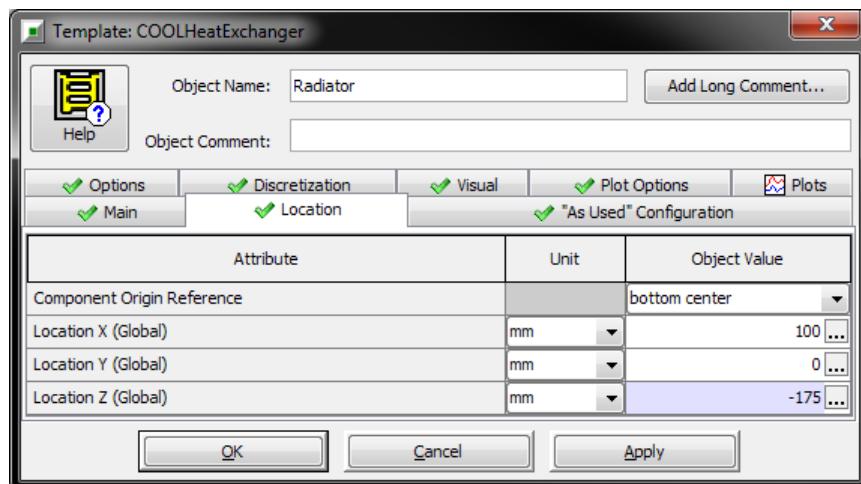
The same parameter that was created for the initial fluid temperature was also used for the Initial Wall Temperature.



Tutorial 6: Modeling an Underhood System with COOL3D



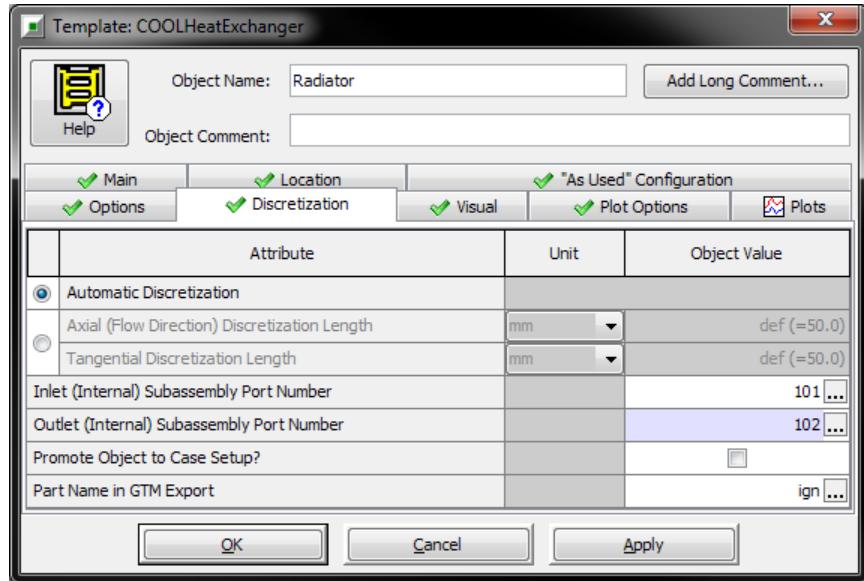
The **Location** folder is where the XYZ position of the heat exchanger can be set. The reference point for the heat exchanger can be picked from one of nine locations. In this case, the bottom center location was measured in CAD.



Optional: The **Discretization** folder is where the dangling port numbers (inlet/outlet connections) for the heat exchanger can be defined to make it easier to integrate the model with GT-ISE. While it is not required to enter a value for the Inlet Subassembly Port Number or Outlet Subassembly Port Number, it is recommended to do so to make it easier to create the necessary links in GT-ISE. The inlet and outlet port numbers for the 'Radiator' were set to 101 and 102, respectively. The numbers entered do not matter, and they can be a different value as long as they are not duplicated.

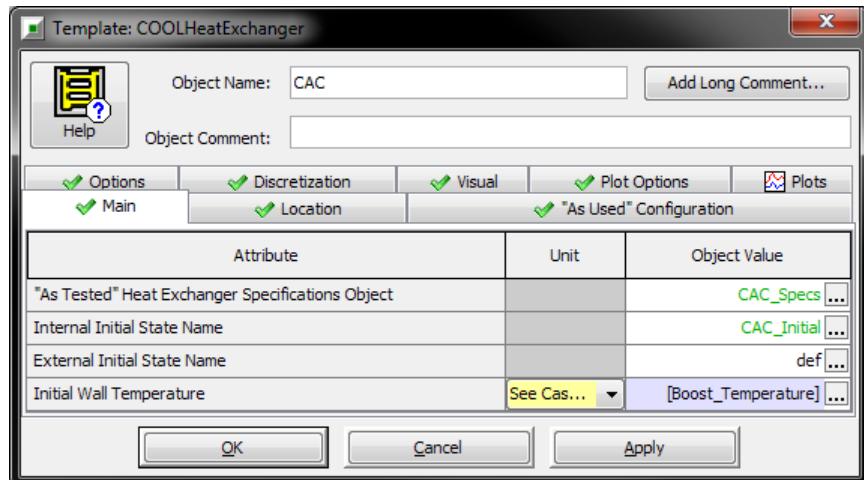


Tutorial 6: Modeling an Underhood System with COOL3D

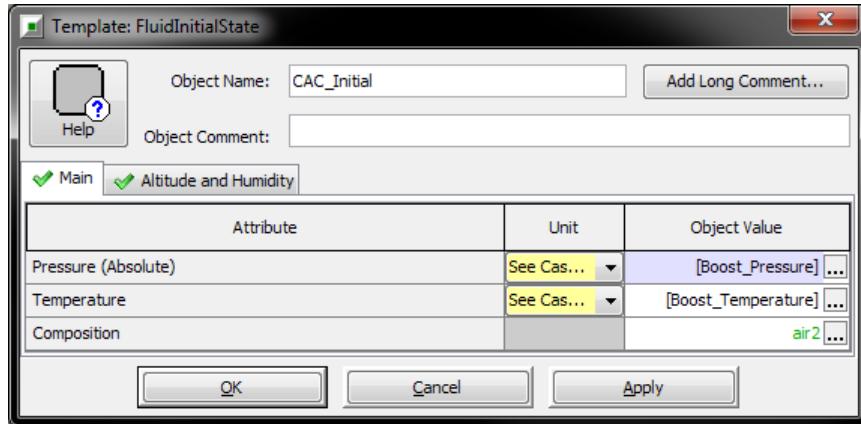


Click OK on the 'Radiator' dialog to complete the component. If successful, a 3D model representation of the heat exchanger should appear in the project map. (The **Visual** folder in the object can be used to change the color of the component.)

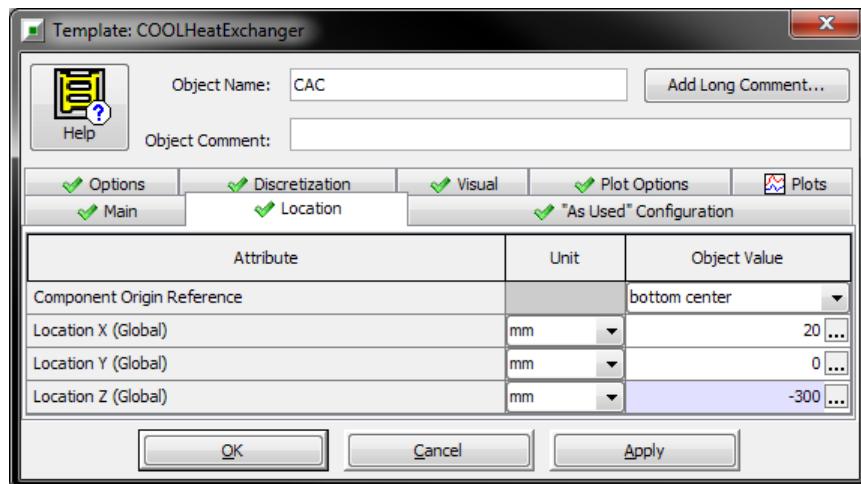
Before continuing, create a second heat exchanger to build a charge-air-cooler (CAC). Using the Value Selector, the 'CAC_Specs' object can be used for the "As Tested" Heat Exchanger Specification Object, the Initial State Name can use 'CAC_Initial', and the Initial Wall Temperature can use [Boost_Temperature].



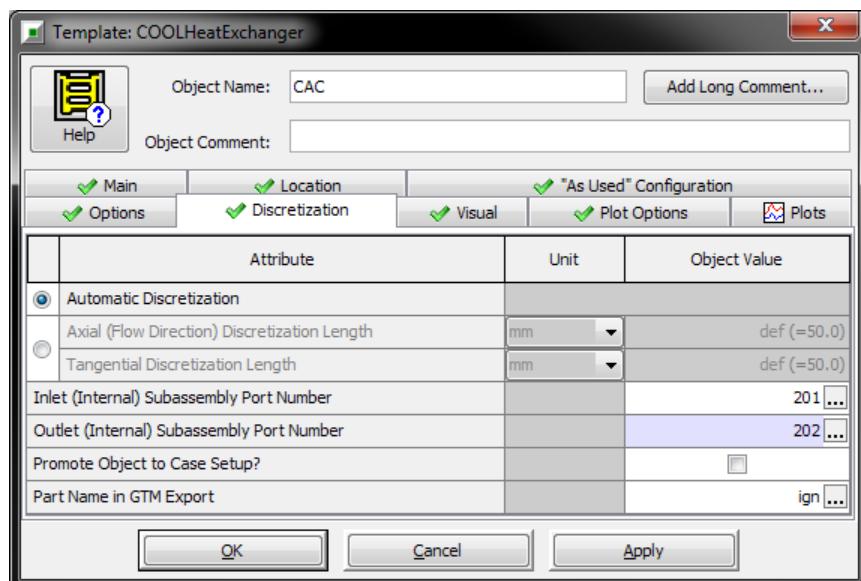
Tutorial 6: Modeling an Underhood System with COOL3D



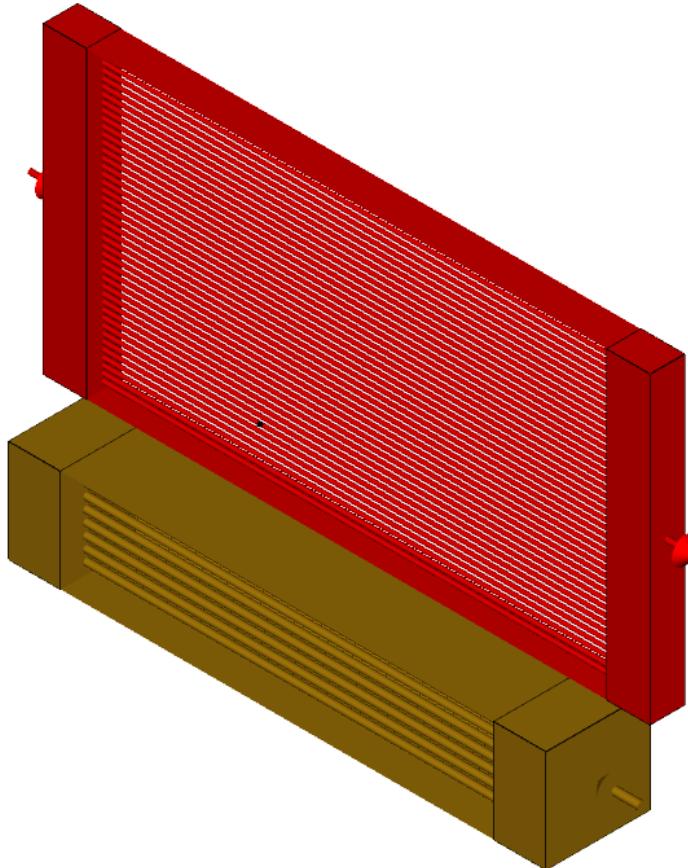
The location of the 'CAC' will also be defined from the bottom center.



Optionally, the Inlet and Outlet Subassembly Port Number can be defined. Click OK when finished.



The rendering of the heat exchangers will display the height, width, number of tubes, number of passes, number of cores, and internal flow direction as defined in the geometry object.



6.3.3 Building a Heat Addition

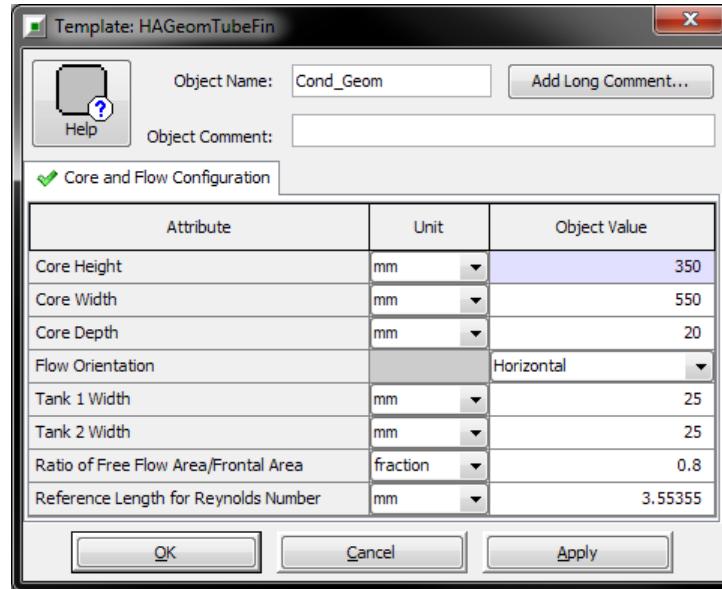
The 'COOLHeatAddition' is a simplified form of the 'COOLHeatExchanger'. It requires less geometry input, only models the external side, and imposes a heat input rate instead of using a performance map. This component is useful when the internal fluid performance is not of interest, not all of the information to model a heat exchanger is available, and/or the heat rejection is already known (i.e. a Condenser).

The 'COOLHeatAddition' object that will be built will model a Condenser in the cooling stack. To build a heat addition, select the heat addition icon (█) from the **Build** menu. The geometry and pressure drop data for the Condenser can be found in the file *DataSheets.xlsx* in the directory ..\tutorials\Modeling_Applications\Cooling_Thermal_Management\.

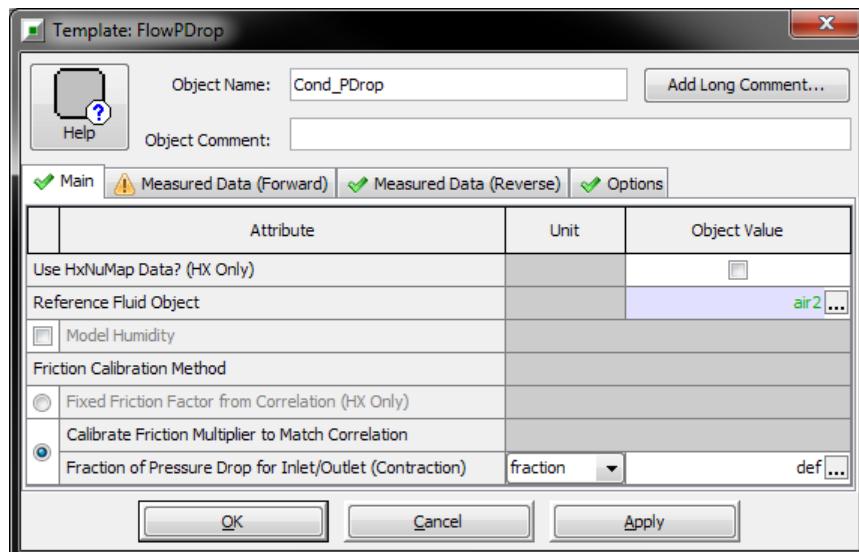
The **Main** folder is where the geometry and heat input rate to the underhood airflow is defined. Use the Value Selector for the "As Tested" Heat Addition Specifications Object to create a 'HeatAdditionSpecs' object. The "As Tested" Heat Addition Geometry Object is used to define the geometry of the heat addition component. Click OK when finished.



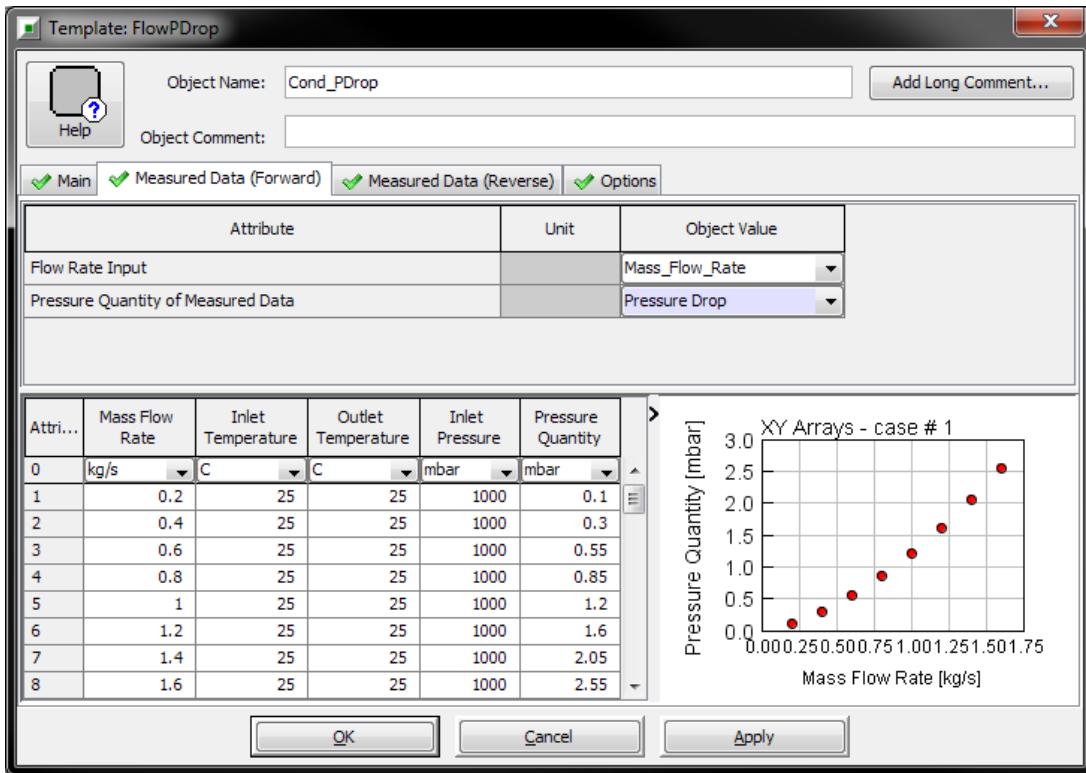
Tutorial 6: Modeling an Underhood System with COOL3D



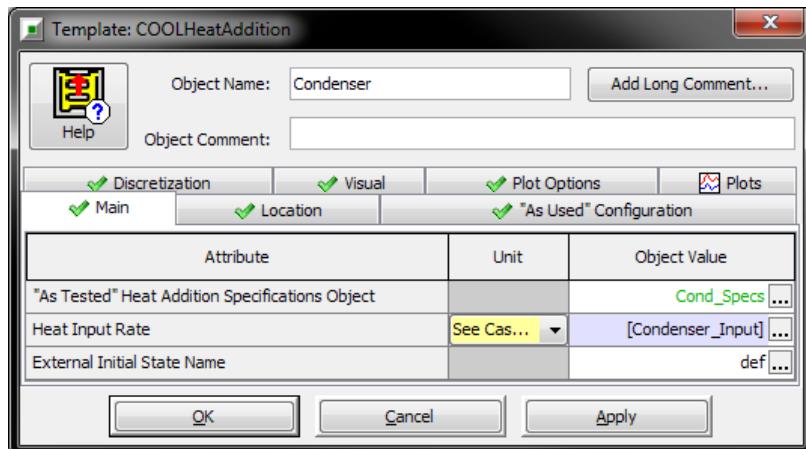
The external pressure drop for the Condenser can be entered for the attribute External Pressure Drop Data Object using the 'FlowPDrop' template. The pressure drop data can be copied from the *DataSheets.xlsx* file. Click OK on the object once the data has been entered, and again to accept the creation of the 'HeatAdditionSpecs' object.



Tutorial 6: Modeling an Underhood System with COOL3D



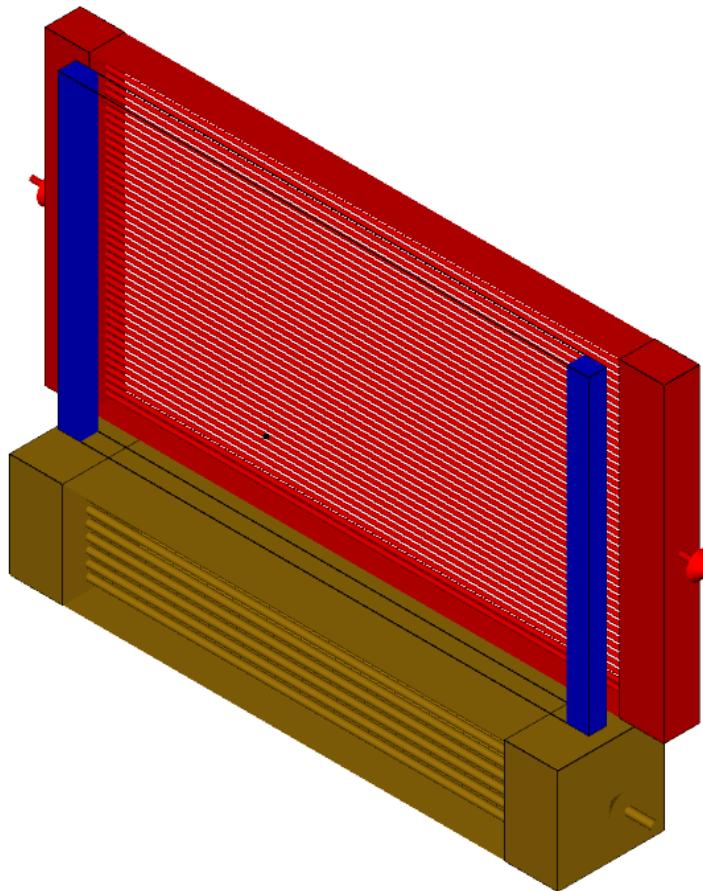
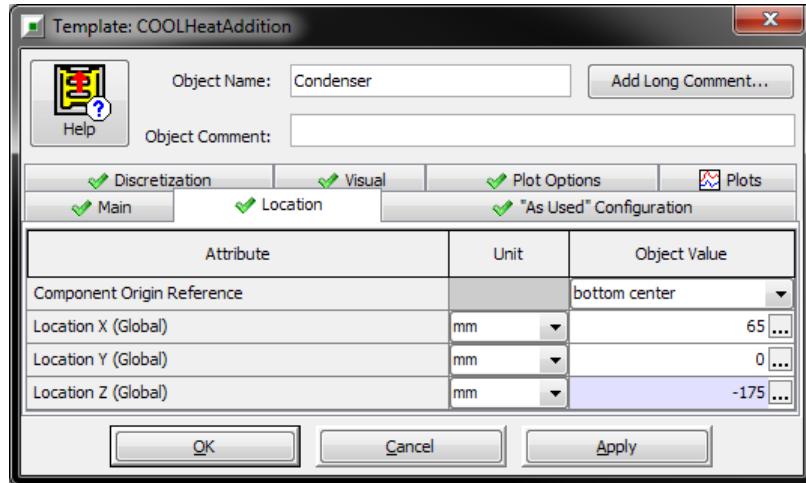
The Heat Input Rate attribute in the 'Condenser' object will be set as a parameter so the value can be varied.



The **Location** folder will set the position of the object in the model. Click OK when finished to create the 'Condenser' object.



Tutorial 6: Modeling an Underhood System with COOL3D



6.3.4 Building a Flow Space

A flow space is required to define the constraints of the underhood model. The model will not run without a flow space. There are two ways to build a flow space, both of which are acceptable to use:

- Method 1 involves using 'COOLFlowSpaceSimple' to build the flow space. The geometry of this template is automatically calculated based on the geometry of the components that exist in the

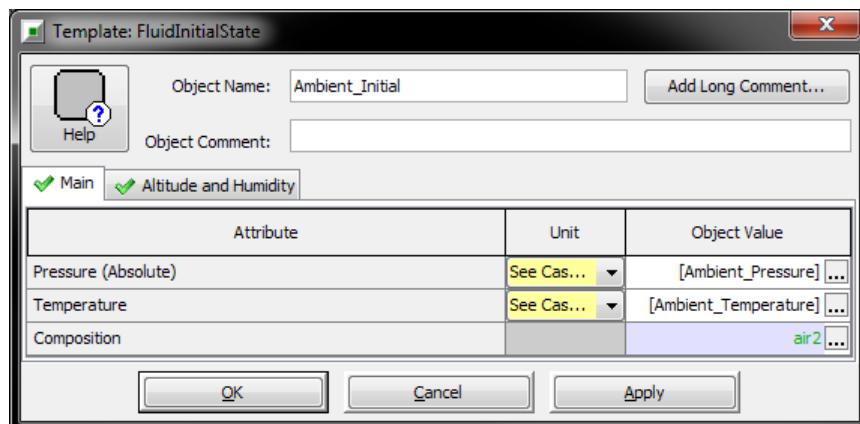


model. This is the simplest form to use, but the flexibility of the shape is limited. If this option is selected, only one flow space type is allowed in a model. This is the option that will be used in this tutorial.

- Method 2 involves using 'COOLFlowSpace' to build a customized flow space shape. The geometry of this template has more flexibility and is controlled through the use of cross sections extruded over distances. The location of the object is also explicitly set as opposed to be defined by a relative location. If this option is selected, multiple 'COOLFlowSpace' objects are allowed in the model.

The 'COOLFlowSpaceSimple' method will be used to define the constraints of the model. To build a simple flow space, select the flow space simple icon (F) from the **Build** menu.

The **Location** folder is where the relative distances for the location of the first and last cross section will exist for the flow space with respect to the components in the model. A value of 50 mm is a recommended value to use initially, but can be changed to adjust the flow volume in the circuit. Define the initial conditions of the underhood flow with the **Initial State Name** attribute in the **Initial State** folder using the 'FluidInitialState' template. Click OK when finished in both object to create the flow space.



The flow space object create will wrap about the components that have been created so far.

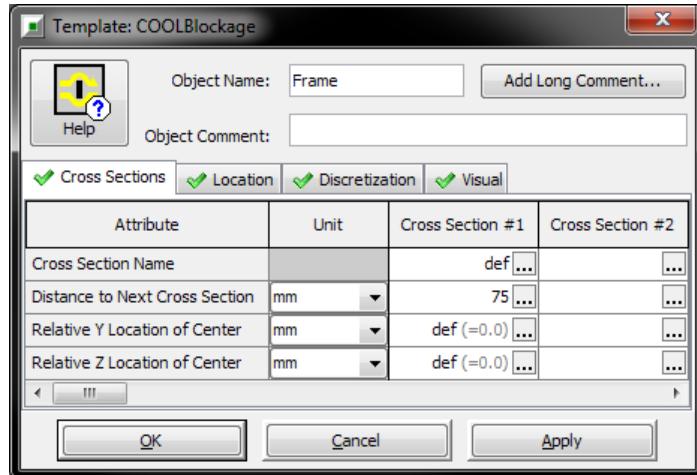
6.3.5 Building a Blockage

A blockage object can be used to build any shape to restrict (prevent) flow in a certain location, such as an engine block. Holes can also be added to blockages to allow flow through them, such as a fascia or the frame of a cooling package.

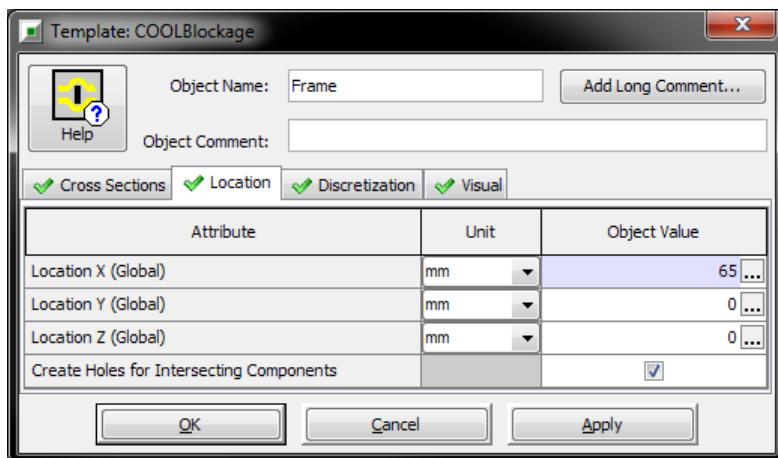
A blockage will be added to represent the frame holding the cooling package. To build a blockage ('COOLBlockage'), select the blockage icon (B) from the **Build** menu. The **Cross Sections** folder if used to create the custom shape of the blockage. Similar to the 'COOLFlowSpace' template, the shape can be defined by cross sections that are extruded over distances. However, one simple method of defining the cross section of a blockage for the use of a frame is to set the Cross Section Name to "def". A value of "def" will automatically use the shape of any flow space object that the blockage is placed inside. All that is required then is to define the extrusion distance.

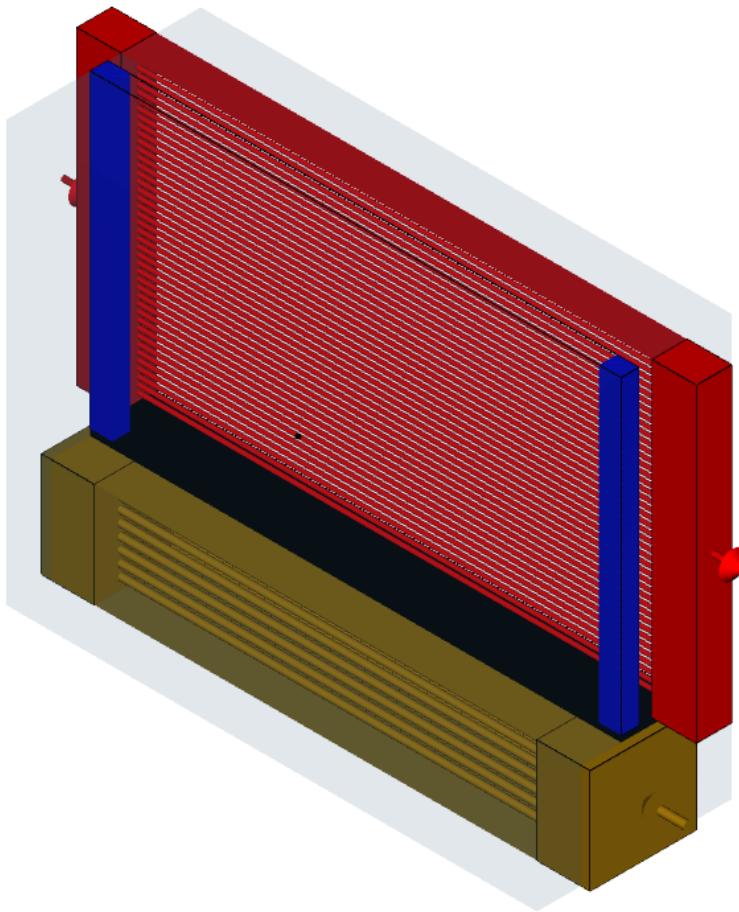


Tutorial 6: Modeling an Underhood System with COOL3D



The **Location** folder will be set to the following position to set the blockage location. In addition, by enabling the attribute Create Holes for Intersecting Components any component (i.e. heat exchanger) that intersects with the blockage will automatically create a hole in the blockage. This is especially useful when trying to quickly create a frame around a cooling package when it is possible for the components in the cooling package to change size and position. Any modification to these components will cause the hole to adjust appropriately. Click OK when finished to build the frame.





6.3.6 Creating Boundary Conditions

Boundary conditions are defined as a feature of a flow space (a child component). There are two types of boundary conditions that can be created in COOL3D.

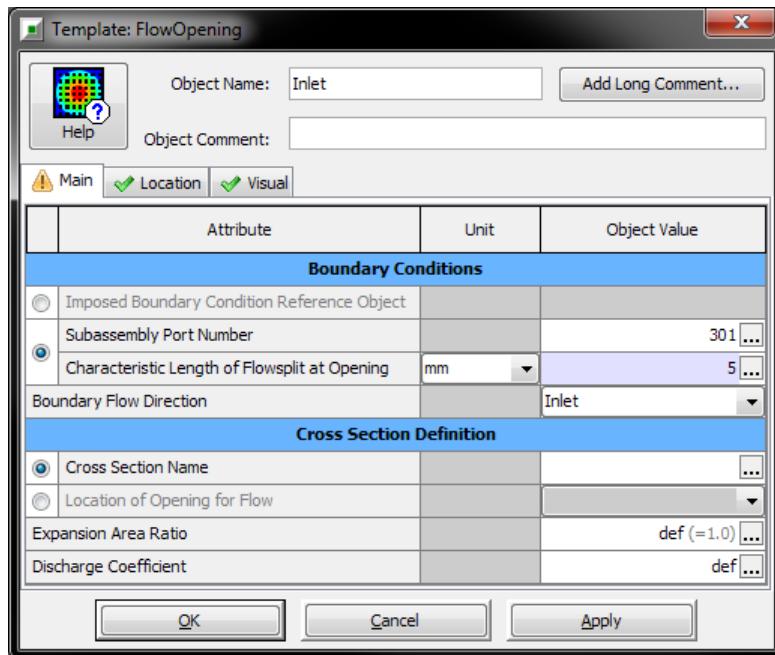
- Method 1 involves creating a reference object to set the pressure, temperature, and/or flow rate directly in COOL3D.
- Method 2 involves creating a dangling connection to integrate the underhood circuit with other flow parts in a GT-ISE model. This is the option that will be used in this tutorial.

To create a boundary condition, the flow space must first be selected. When it is selected, the boundary condition ('FlowOpening') icon (a square with a small circle) in the toolbar will be available from the **Component** menu. Similar to the heat exchangers, the Subassembly Port Number can be left as "def" and have a number automatically defined for the dangling connection. To make it easier to integrate the model, a value of 301 will be used for the inlet boundary (the number does not matter as long as it does not duplicate a number already selected). The Characteristic Length is prefilled with a recommended value to create a flow volume to assist in the integration that will be handled later. The value entered can be used to adjust the size of the flow volume created.

Note: Selecting the option Imposed Boundary Conditions Reference Object will allow the boundary condition to be defined as a pressure, temperature, and/or flow rate.



Tutorial 6: Modeling an Underhood System with COOL3D

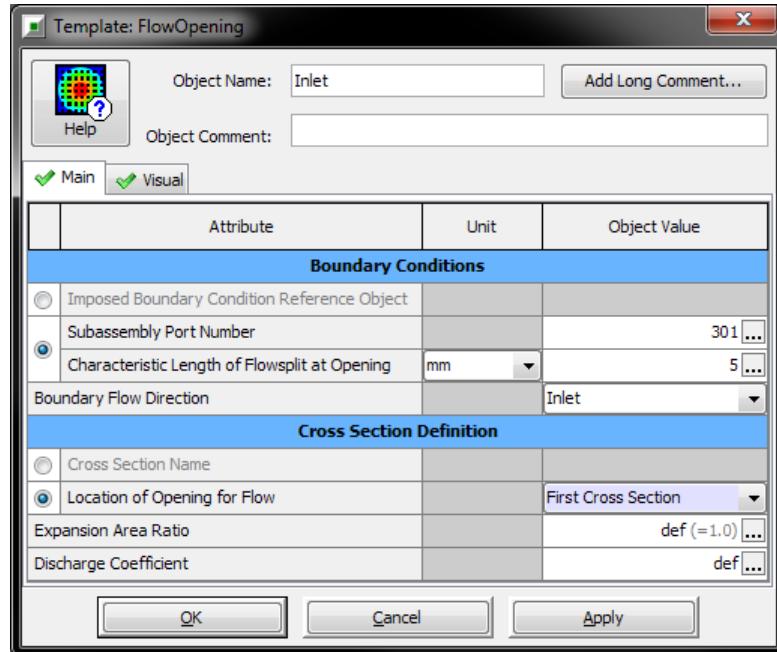


The Boundary Flow Direction values only matters for the sign used when reporting flow rate at the boundary locations. If flow goes along the direction of the connection, then the flow rate and velocity reported will be positive.

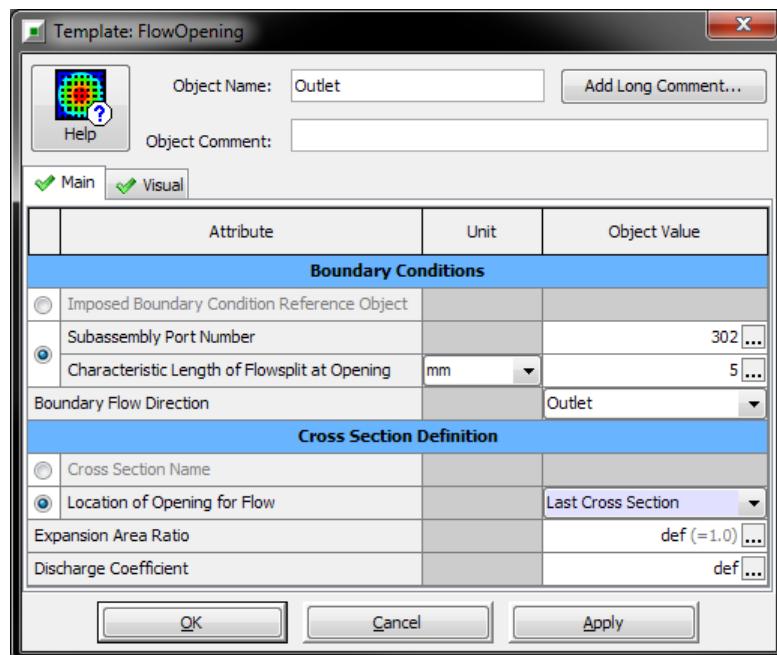
To define the shape of boundary conditions easily, select the Location of Opening for Flow option. The value selected here will automatically create an opening for the boundary that is the exact same size as the flow space at the location selected. If the Cross Section Name option was selected, then a custom shape can be defined for the boundary opening instead, which may be useful when attempting to model flow distribution patterns in an underhood model. Click OK when finished to create the inlet boundary opening for the flow space.

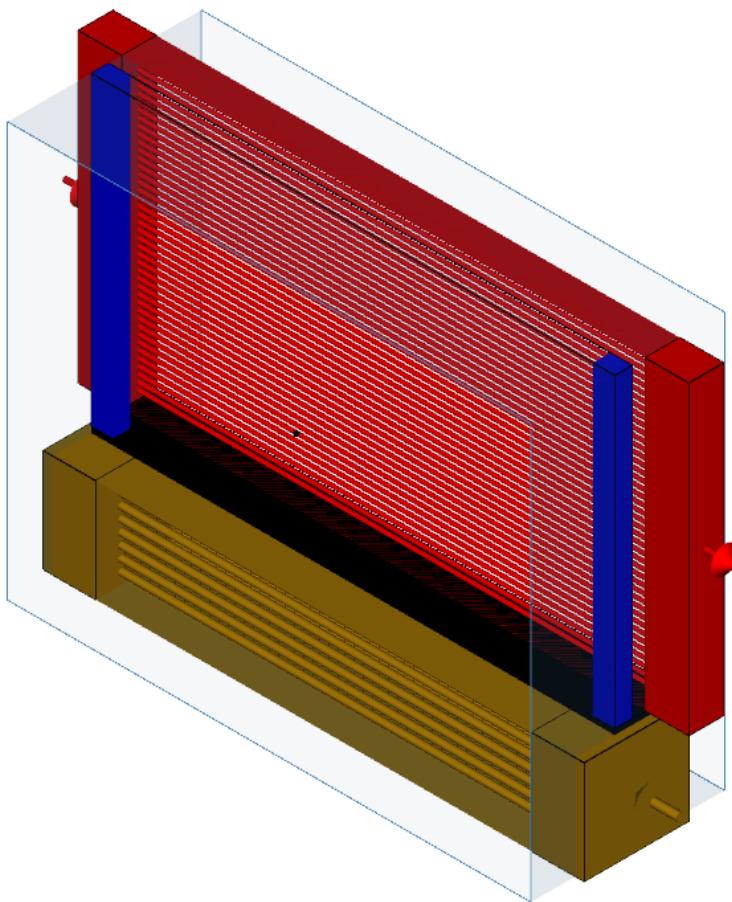


Tutorial 6: Modeling an Underhood System with COOL3D



Repeat the process for the outlet boundary using the information seen below.

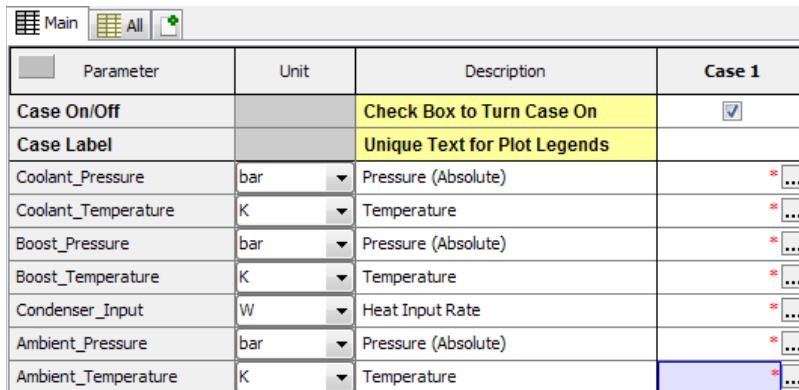




6.3.7 Case Setup

The Case Setup in COOL3D is a little different than GT-ISE. Each case in the Case Setup of COOL3D represents an underhood model that will be created when the model is discretized. This is useful when geometry or components are changing. However, it is possible to use GT-ISE to point directly to a COOL3D file (.ghx), which will be performed in the next section. If following this path, then only a single case can exist in the COOL3D file. This is because the different modeling permutations for the geometry will be set up in GT-ISE.

Open Case Setup and set the value of the parameters that were created to *. This will leave the parameters as floating so that GT-ISE will know to import them when working with the "main" .gtm model. However, not all attribute can be set to a *. Those that cannot are the ones related to geometry and location. In this case a value must be defined, but GT-ISE will still import them correctly when requested. At this time there are no parameters for any location or geometry attributes, so it is safe to set all parameters to a value of *. Click OK on Case Setup once this is done.



The screenshot shows a software interface with a toolbar at the top. The main window contains a table with columns for Parameter, Unit, Description, and Case 1. The table includes rows for Case On/Off (with a checked checkbox), Case Label (highlighted in yellow), Coolant_Pressure, Coolant_Temperature, Boost_Pressure, Boost_Temperature, Condenser_Input, Ambient_Pressure, and Ambient_Temperature. The last row, Ambient_Temperature, has a blue selection bar.

Parameter	Unit	Description	Case 1
Case On/Off		Check Box to Turn Case On	<input checked="" type="checkbox"/>
Case Label		Unique Text for Plot Legends	
Coolant_Pressure	bar	Pressure (Absolute)	* ...
Coolant_Temperature	K	Temperature	* ...
Boost_Pressure	bar	Pressure (Absolute)	* ...
Boost_Temperature	K	Temperature	* ...
Condenser_Input	W	Heat Input Rate	* ...
Ambient_Pressure	bar	Pressure (Absolute)	* ...
Ambient_Temperature	K	Temperature	

Optional: It is possible to discretize the COOL3D model to generate an equivalent .gtm model file by clicking on the **Run Simulation** button () found in the toolbar. This button will automatically discretize the model. This option is recommended if no studies will be performed on the underhood model (i.e. the design is fixed).

6.4 Running an Underhood Model from GT-ISE

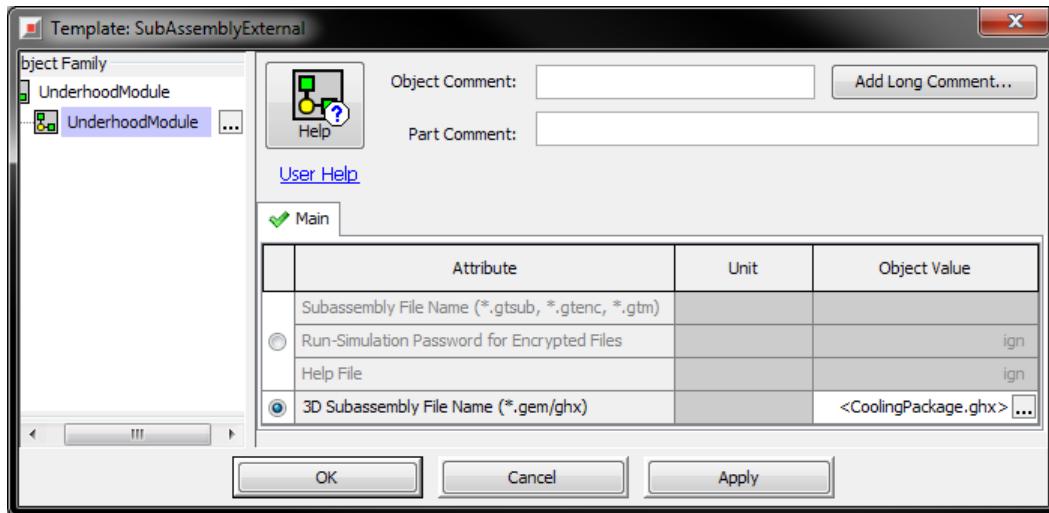
6.4.1 Placing Parts, Connections, and Links for an Underhood Model

Open the model UnderhoodModule.gtm found in the directory ..\tutorials\Modeling_Applications\Cooling_Thermal_Management\06-UnderhoodModule\. This model is prepared to run simple steady-state operating points for an underhood cooling package to determine the final coolant temperature. This is done by imposing the ambient air flow rate, CAC flow rate, pressure, and temperature, and the coolant flow rate and heat input (engine heat rejection). The boundary conditions for the ambient air ('Ambient_In') and CAC ('CAC_In') are imposed using the 'EndFlowInlet' template. The air flow rate through the underhood model is set up to impose the total flow rate, which is not common, but can be when running simple tests. Alternatively, the air side boundary conditions can be set up as a pressure boundary. The coolant boundary conditions for flow rate and heat input rate are set in the 'ImposeFlow' part which was created from the 'ClosedLoopFlow' template used earlier in CHAPTER 4: Modeling a Hydraulic System (No Heat Transfer).

The 'UnderhoodModule' part is an external subassembly (external model) that will reference the COOL3D model created in the previous step. Open the 'UnderhoodModule' part so the COOL3D model can be referenced. Once the dialog is open, use the Value Selector for the 3D Subassembly File Name (*.gem/ghx) to point to the COOL3D (.ghx) file that was created in the previous step. Click OK when finished.



Tutorial 6: Modeling an Underhood System with COOL3D



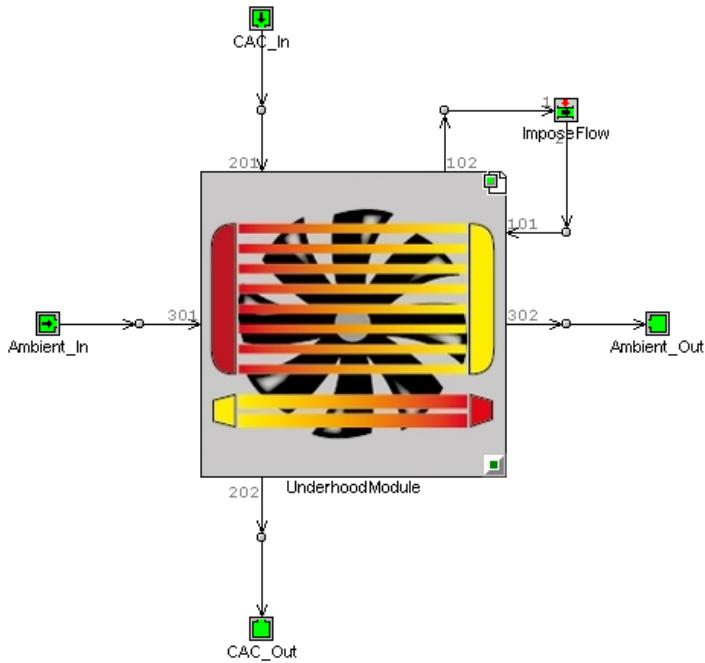
Alternative Method: If the COOL3D model was discretized in the previous step to generate a .gtsub file, then the Subassembly File Name (*.gtsub) attribute will need to be complete instead.

The important thing to remember when preparing a model to run an underhood model from COOL3D is the port number of all connections going to and from the external subassembly. This will be true for all external subassemblies in a model. When connecting links, the port number should match the subassembly port number defined in the COOL3D model.

- Radiator Inlet: 101
- Radiator Outlet: 102
- CAC Inlet: 201
- CAC Outlet: 202
- Ambient Inlet: 301
- Ambient Outlet: 302

The model is already prepared with the correct port numbers assuming the above list was followed.





Optional: If the port numbers are different, then double-click on the link connected to the external subassembly. Once the link dialog opens enter a new Link ID, and then click OK.

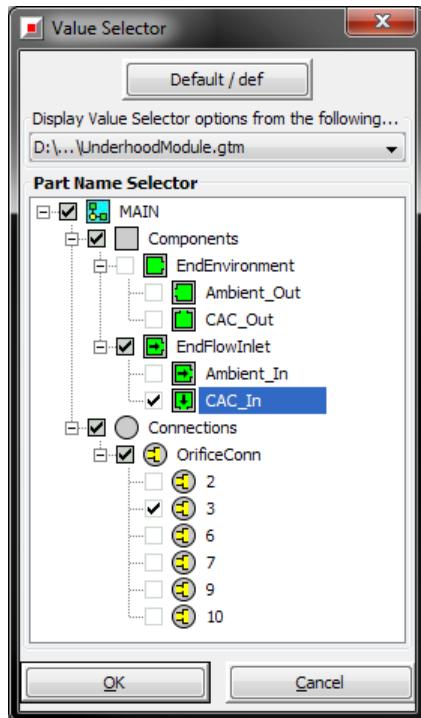
6.4.2 Creating Circuits for an Underhood Model

It is recommended to define the circuits of an underhood model to make it easier to diagnose any issues that arise, as well as allow the circuit to be solved/converge using a different set of criteria. Open Run Setup and access the **FlowControl** folder. Up until now only a single column (or flow setting) was defined. To define more than one flow setting a column needs to be created for each circuit(s). An underhood model with COOL3D should solve the internal flow using one solver setting, and the external (ambient) flow should solve with a different solver setting. To define the circuits for the internal flow, use the Value Selector for the Flow Settings #1 column for the attribute Part Name List Object, and select the 'FlowCircPartSelector' template.

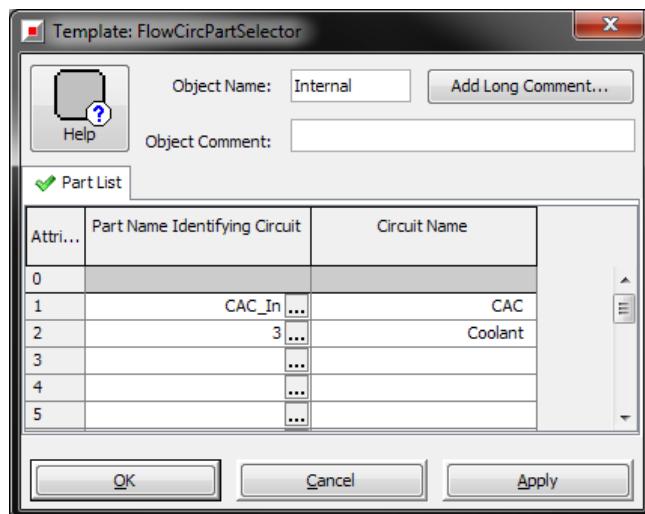
The 'FlowCircPartSelector' template allows parts to be selected to explicitly define the circuits that will be created, as well as their name. Use the Value Selector for the Part Name Identifying Circuit column and select two parts associated with the CAC and Coolant circuits (any part can be selected). If "def" is used, then a random part will be selected (not recommended). Click OK when finished.



Tutorial 6: Modeling an Underhood System with COOL3D

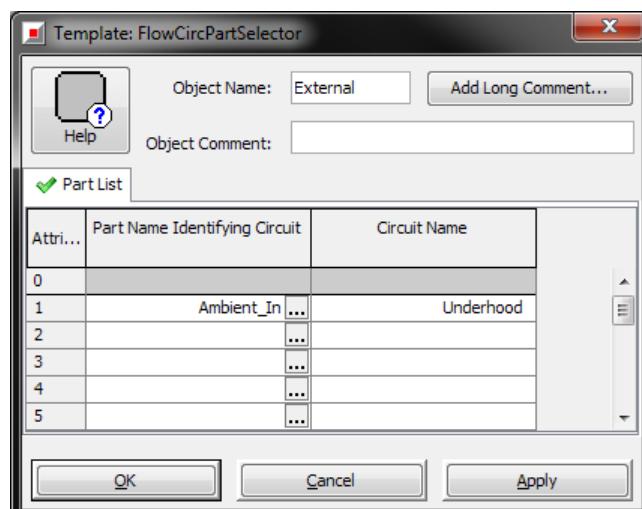
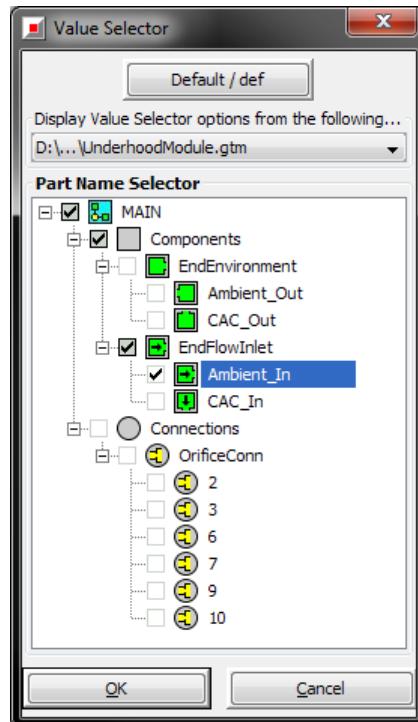


Once the part names are imported, provide a name in the Circuit Name column for each part that can be used to quickly identify the circuit when the solver runs. If the name is left as "def", then the part name selected will be used instead. Click OK when finished.



Do the same procedure for the second column (Flow Settings #2) to define/name the ambient air circuit. Click OK when finished.

Tutorial 6: Modeling an Underhood System with COOL3D



Attribute	Unit	Flow Settings #1	Flow Settings #2
Part Name List Object Identifying Circuits Belonging to Column		Internal <input type="button" value="..."/>	External <input type="button" value="..."/>
Time Step and Solution Control Object		<input type="button" value="..."/>	<input type="button" value="..."/>
Solve All Circuits Together (Single Solution Cluster for the Column)		<input type="checkbox"/>	<input type="checkbox"/>

To define the solver to be used for each column, use the Value Selector for the Time Step and Solution Control Object to select 'Implicit' for Flow Settings #1, and 'ImplicitUnderhood' for Flow Settings #2. These solver objects are optimized for underhood models with COOL3D, and are recommended to use. Click OK when finished to complete the circuit solver settings.

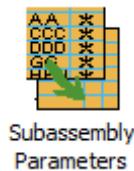


Tutorial 6: Modeling an Underhood System with COOL3D

Attribute	Unit	Flow Settings #1	Flow Settings #2
Part Name List Object Identifying Circuits Belonging to Column		Internal <input type="button" value="..."/>	External <input type="button" value="..."/>
Time Step and Solution Control Object		Implicit <input type="button" value="..."/>	ImplicitUnderhood <input type="button" value="..."/>
Solve All Circuits Together (Single Solution Cluster for the Column)		<input type="checkbox"/>	<input type="checkbox"/>

6.4.3 Testing an Underhood Model

Prior to running the model, the test conditions for the underhood model need to be created. Open Case Setup to create the boundary conditions for the underhood model to test two different operating points. These boundary conditions are typically taken from a flow test bench, or from CFD results, and will be used to validate the model. The ambient air flow rate will be imposed as a total flow rate. Before filling out the operating conditions, click on the Subassembly Parameters button to ensure all parameters are imported from the external COOL3D file.



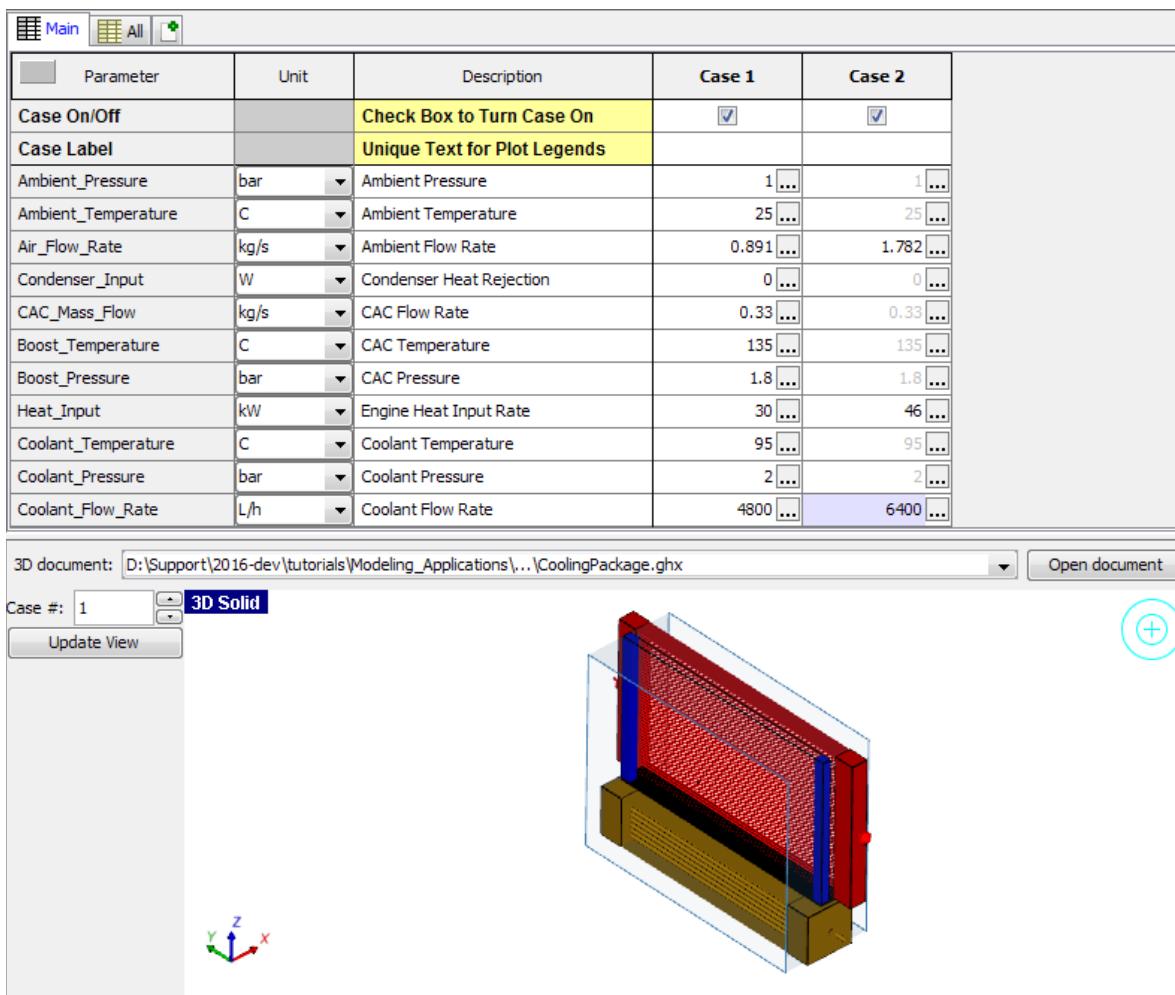
Subassembly
Parameters

Once all of the parameters are imported to the model, complete them by filling out the operating conditions. Click OK when finished.

Note: The panel along the bottom of Case Setup can be used to visualize the geometry of the COOL3D model if the location, dimensions, and/or components of the model were being changed from one case to another.



Tutorial 6: Modeling an Underhood System with COOL3D



Run the model at this time. The simulation will not start immediately because it is discretizing the COOL3D model. Once the process is completed (it should only take a few seconds) the solver will begin to run the two operating cases defined.

Use the RLT Contour Mode in GT-POST to view the steady-state coolant temperature. It is also possible to view other steady-state RLTs by double-clicking on any part to open the results summary dialog (i.e. flow rate, outlet heat exchanger temperatures, etc).

6.5 Calibrating an Underhood Model

Sometimes it is necessary to calibrate an underhood model to target a known measured air flow rate because of the simplified form in which the underhood (COOL3D) model is built. The calibration is usually added at locations that bypass components. As more detail is included in the COOL3D model to match the physical object(s), less calibration will be needed.

Open the model UnderhoodCalibrate.gtm found in the directory ..\tutorials\Modeling_Applications\Cooling_Thermal_Management\06-UnderhoodModule\. The external air flow rate will be calibrated to match a target flow distribution through the heat exchanger components.



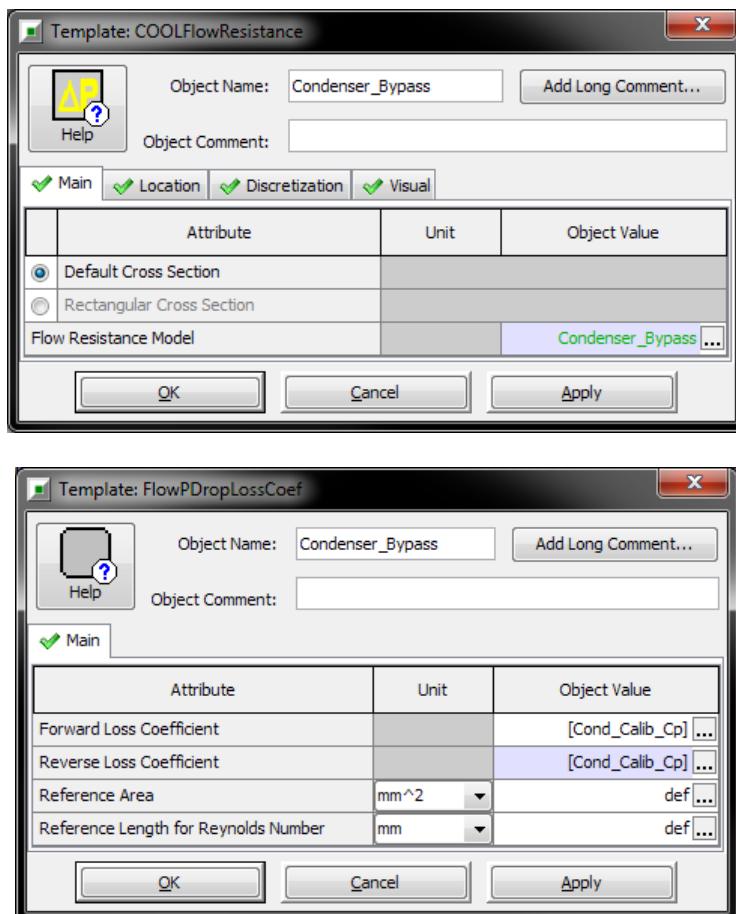
Tutorial 6: Modeling an Underhood System with COOL3D

A grille pressure loss and fan will also be included to build a model that can predict the underhood flow rate at various vehicle speeds.

The model is set up to run an imposed total air flow rate of 1.7 kg/s. It is important that the other boundary conditions are entered correctly to match the target test as well. After running the model, the flow rate through the Condenser is only 0.781 kg/s, whereas the target flow rate is 0.9 kg/s. To force more flow through the Condenser, a resistance will be added to the underhood model around the Condenser. Open the underhood model by right-clicking on the part 'UnderhoodModule' and selecting Open External Subassembly. This will automatically launch the underhood model in COOL3D.

6.5.1 Creating a Pressure Resistance Plane

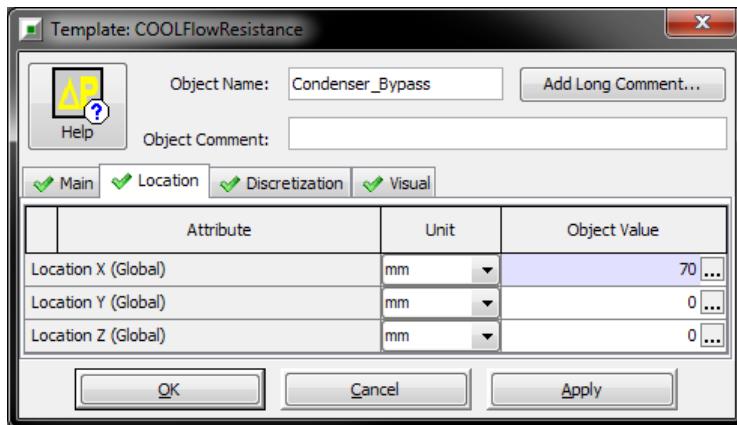
A resistance needs to be added around the Condenser. To add a resistance, select the resistance icon () from the **Build** menu. There are two options for the resistance; it can be built as a plane that takes the shape of the flow space, or as a rectangular shape. Select the Default Cross Section option to use the shape of the flow space. For the attribute Flow Resistance Model, use the Value Selector to pick the 'FlowPDropLossCoef' template. Selecting this template will allow a pressure loss coefficient to be imposed to model the pressure drop around the Condenser. Create a parameter for the Forward and Reverse Loss Coefficient such that it can be calibrated with the GT-SUITE solver. Click OK when finished.



In the **Location** folder, set the X position such that it is in the same plane as the Condenser. Click OK when finished to create the resistance plane around the Condenser.

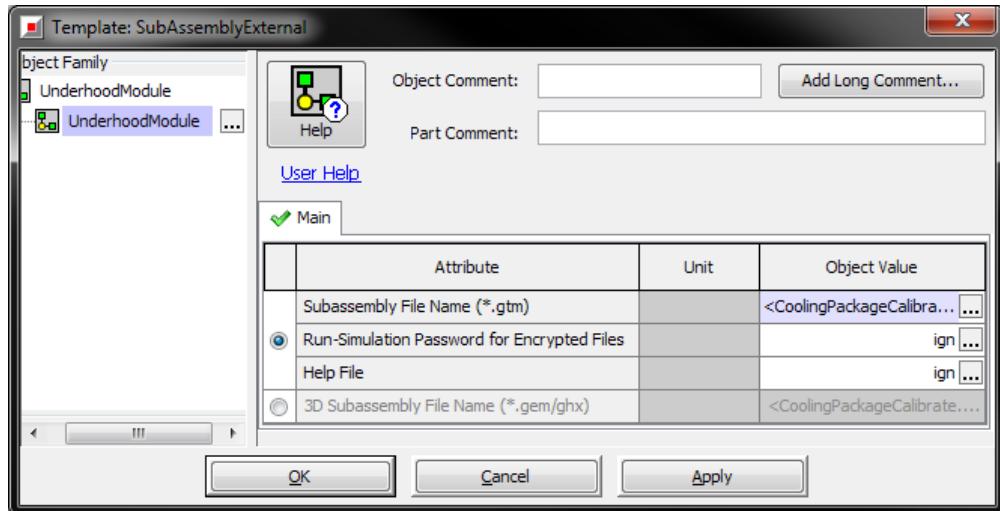


Tutorial 6: Modeling an Underhood System with COOL3D



Before going back to GT-ISE, access the Case Setup in COOL3D and set a value of * for the [Cond_Calib_Cp] parameter. Click OK when finished and discretize the model () to generate a .gtsub file.

Back in GT-ISE, open the 'UnderhoodModule' part and point to the .gtsub file that was created in the Subassembly File Name (*.gtsub, *.gtenc, *.gtm) attribute. Calibrating an underhood model involves working with the fixed location and geometry of the components. To speed up the calibration, the .gtsub file is being used instead. Click OK when finished.

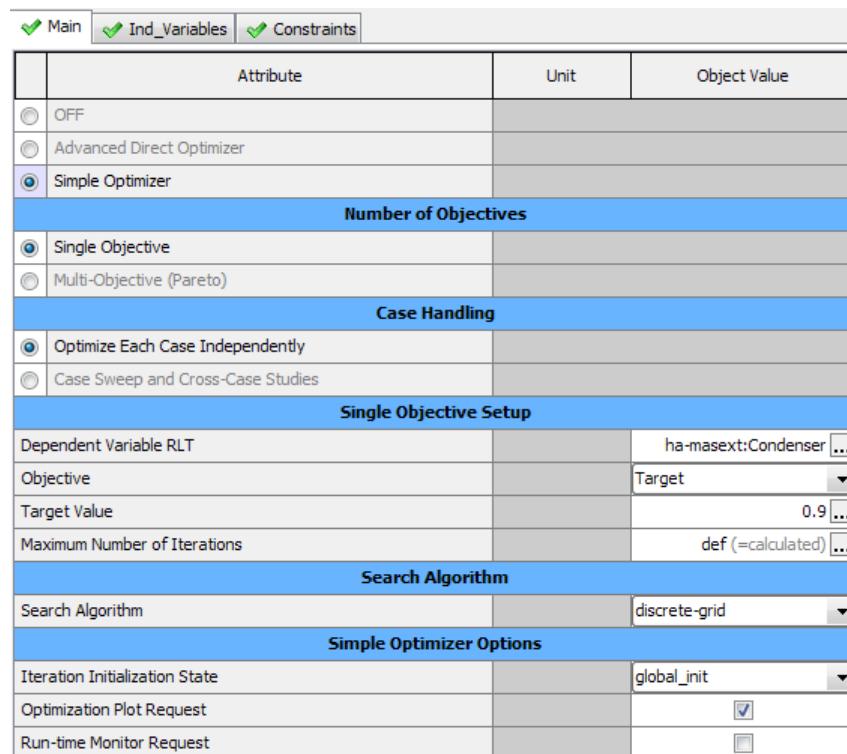
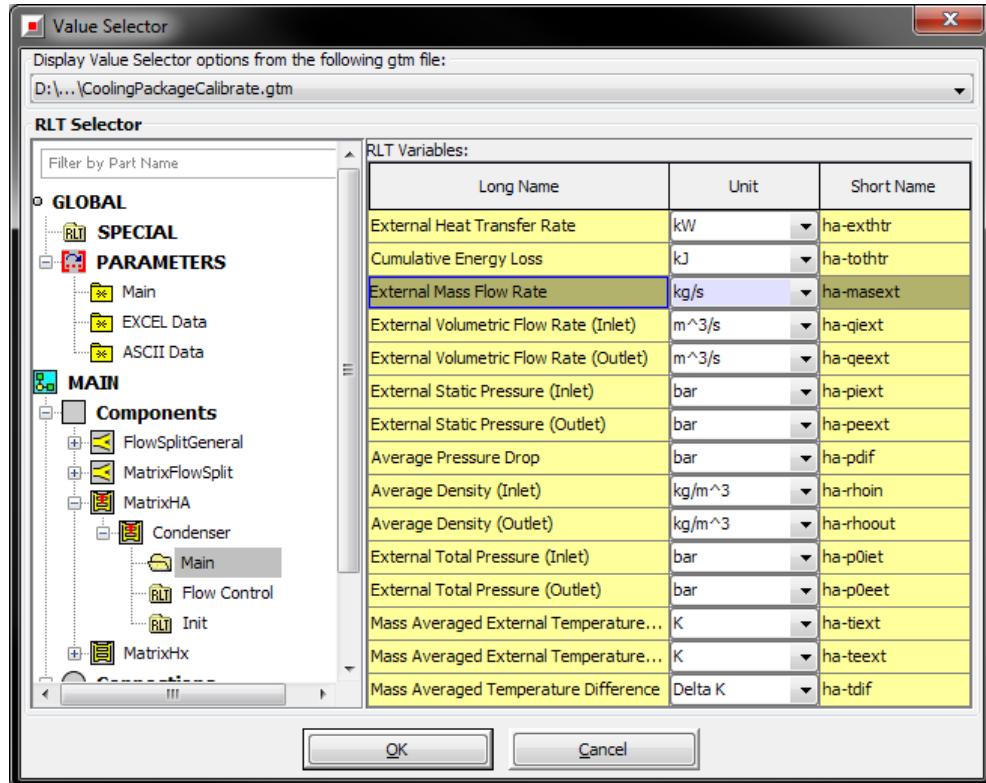


Access Case Setup and select the Subassembly Parameters button to import the newly created parameter. Set the initial value of the [Cond_Calib_Cp] parameter to 1 (this value will be overridden later on when calibrating). Click OK when finished.

To calibrate the pressure loss coefficient around the Condenser go to the **Home** tab and select **Optimization→Direct Optimizer**. Enable the Simple Optimizer, and set the Objective to "Target", and the Dependent Variable RLT to the external mass flow rate of the Condenser. Use the drop down at the top of the Value Selector dialog to locate the external .gtsub file where the Condenser is located. The Target Value of RLT Variable will be 0.9.



Tutorial 6: Modeling an Underhood System with COOL3D

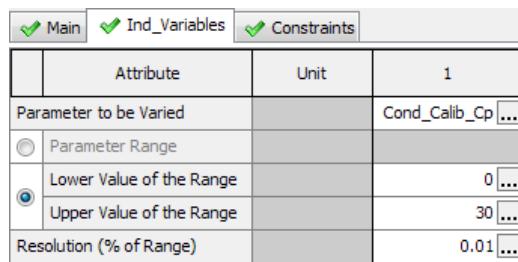


In the **Ind_Variables** folder, set the Parameter to be Varied to be [Cond_Calib_Cp] using the Value Selector, which is the value of the pressure loss coefficient that will be varied to match the target flow



Tutorial 6: Modeling an Underhood System with COOL3D

rate. The Lower and Upper Value of the Range, and the Resolution should be set to the image below. Click OK when finished.



Main		
Attribute	Unit	Value
Parameter to be Varied		Cond_Calib_Cp ...
Parameter Range		
Lower Value of the Range		0 ...
Upper Value of the Range		30 ...
Resolution (% of Range)		0.01 ...

At this time the model is prepared for the solver to find the correct pressure loss coefficient around the Condenser that will yield the correct Condenser flow rate. Run the model at this time. The optimizer will take a few iterations before it reached the correct value. Upon completion, the optimizer will report the pressure loss coefficient that is necessary to achieve the desired flow rate through the Condenser.

INFO Optimization successful

Independent var, Dependent var and Target: 5.94408 0.9 0.9

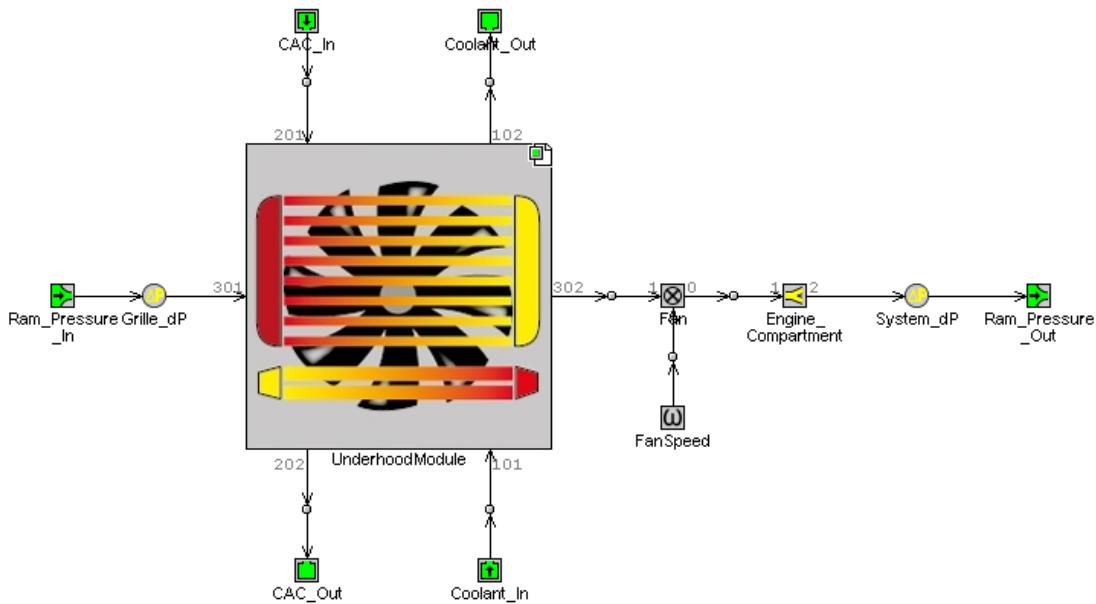
The target value in this case is 5.4317. Copy/paste this value into Case Setup in GT-ISE for the [Cond_Calib_Cp] parameter. If this calibration is sufficient, disable the optimizer in **Optimization→Direct Optimizer** in order to perform a Case Sweep with different operating conditions.

Now that the cooling package is calibrated, the pressure boundaries, grille pressure drop, and fan will be added to the model to calibrate the model for a vehicle speed condition in order to ensure the total underhood pressure drop (target flow rate) is correct.

6.5.2 Calibrating the System Pressure Drop (Flow Rate)

Open the model UnderhoodSystem.gtm found in the directory ..\tutorials\Modeling_Applications\Cooling_Thermal_Management\06-UnderhoodModule\. This model shows a typical setup for a model that will use a ram pressure boundary condition for the external air side along with a fan. However, the system pressure drop must be calibrated to target the correct total flow for the specific operating condition (vehicle speed = 100 kph, fan speed = 5000 RPM). The calibration procedure performed is identical to what was done with the bypass around the Condenser in the previous step; except the lumped pressure drop is now located between the fan and the outlet ram pressure boundary condition, and the target flow rate is through the fan (1.7 kg/s).





Running the model will result in an optimization performed on the 'System_dP' part to find the pressure loss coefficient that will yield the correct fan flow rate for the operating conditions defined.

```
*****
INFO Optimization successful
Independent var, Dependent var and Target: 1.65165 1.7 1.7
*****
```

Copy/paste this value in to Case Setup for the [Engine_Calib_Cp] parameter to represent the pressure loss in the engine compartment. This completes the calibration of the underhood model.

6.6 Case Studies with Changes in Geometry

Once the underhood model is calibrated to match known results, case studies can be performed to see how the cooling package responds to changes in geometry. One common example of this is scaling the geometry of a heat exchanger. When performing a study on change in geometry, the external model can once again point directly to the COOL3D (.ghx) model.

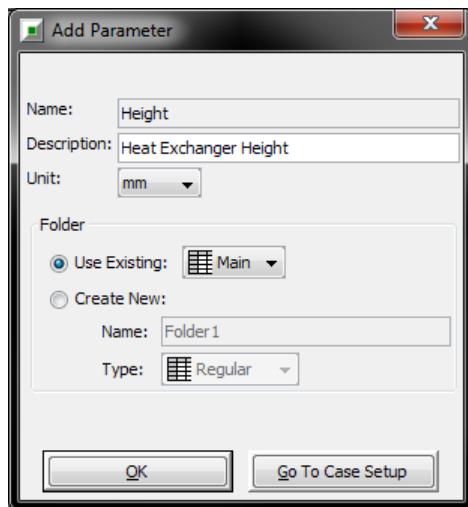
Open the model *UnderhoodScaling.gtm* found in the directory *..\tutorials\Modeling_Applications\Cooling_Thermal_Management\06-UnderhoodModule*. This model uses the results of the previous calibrations as the starting point for the heat exchanger scaling study, and already points directly to a COOL3D model that was previously built. Open COOL3D model to define the geometry changes for a heat exchanger, specifically the Radiator.

Once in COOL3D, open the 'Radiator' object and go to the "**As Used**" Configuration folder. This folder is used to define the modeling parameters for scaled heat exchanger. Use the Value Selector for the "As Used" Heat Exchanger Geometry Object to create a scaling reference object. The scaling options available for a tube-fin heat exchanger are the absolute dimensions, or the relative changes. Both options are acceptable forms to model the scaling of a heat exchanger, and it comes down to personal preference. Create parameters for the absolute dimensions for height and width. When the parameter is first created,

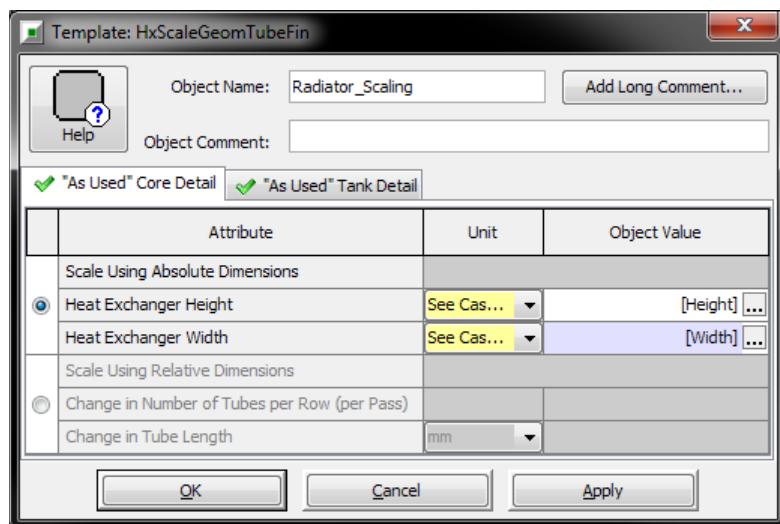


Tutorial 6: Modeling an Underhood System with COOL3D

a prompt to Case Setup will immediately open so the parameter value can be defined. This is required because these attribute will affect the display of the model and therefore require a value at all times.



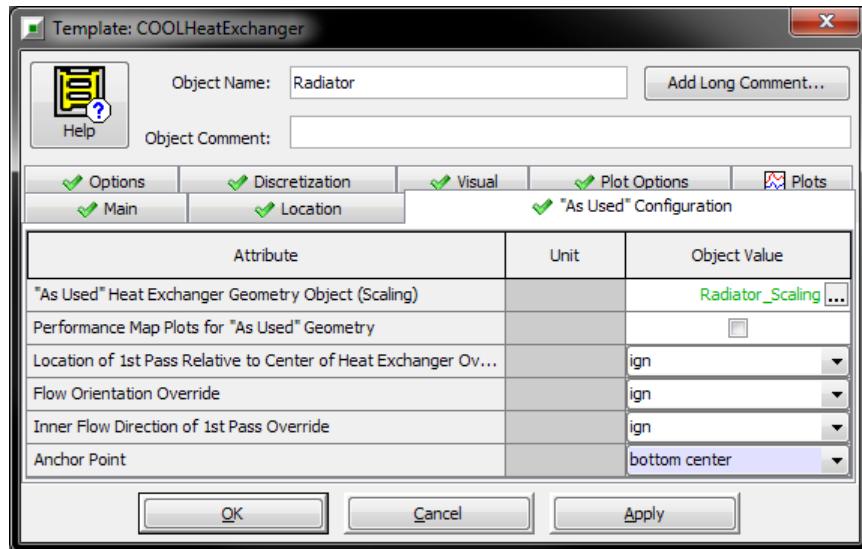
Select the Go To Case Setup option, and enter a value of 350 mm for the scaled height. Click OK in Case Setup to be brought back to the scaled reference object. Perform the same steps for the absolute width as well using a value of 600 mm. Click OK when finished.



Back in the 'Radiator' object, the Anchor Point will be adjusted to instead read "bottom center". The option selected here will fix the heat exchanger in place at the location selected before the geometry is modified. This is useful when there are constraints in the model that must be met. Click OK when finished.



Tutorial 6: Modeling an Underhood System with COOL3D

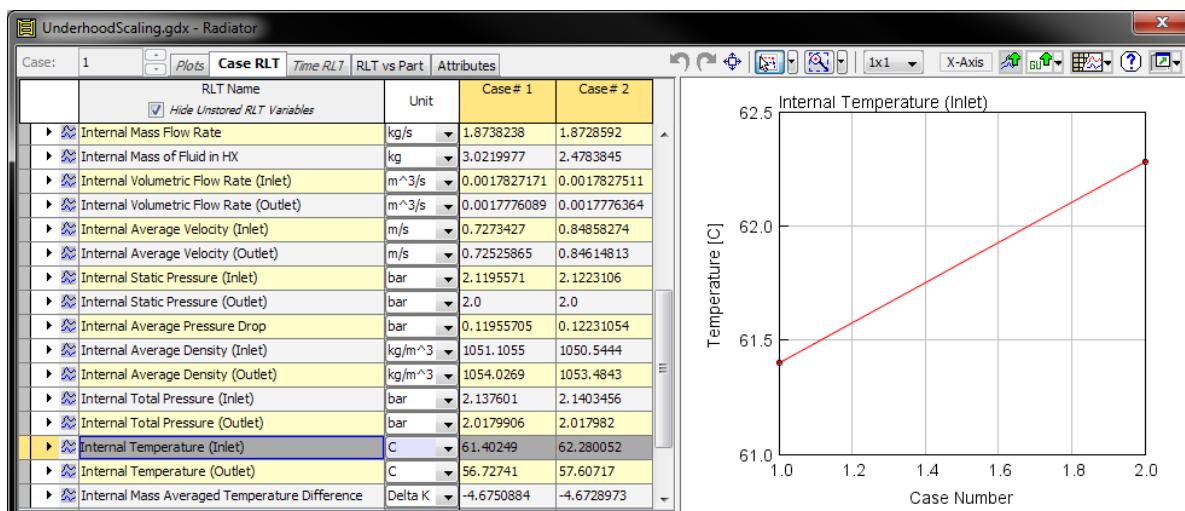


Nothing will change with the display of the model just yet. The dimensions entered for the heat exchanger geometry are such that no scaling is being performed (they were the same as the original dimensions). Save the model and go back to GT-ISE.

Once in GT-ISE, open Case Setup and select the Subassembly Parameters option to import the geometry scaling parameters from COOL3D. Now, the changes to the geometry of the heat exchanger can be applied for various studies. The image below shows Case 1 using the original geometry as the baseline, and Case 2 uses geometry that results in a shorter (height) heat exchanger. The remainder of the parameters will be left alone since they match the operating conditions for the test that will be performed. Click OK when finished, and run the model to test the scaled heat exchange performance.

Height	mm	Heat Exchanger Height	350	300
Width	mm	Heat Exchanger Width	600	550

Once the simulation is finished, the performance of the Radiator (i.e. top tank temperature) can be viewed in GT-POST to see how it changes with respect to the new geometry.



CHAPTER 7: Modeling a Transient Thermal Model

This tutorial has been prepared to provide guidelines on how to model a transient thermal system for a cooling circuit with an integrated vehicle to follow an imposed driving cycle. The vehicle will calculate the necessary engine speed and load to follow the driving cycle, which will then be used to determine the heat rejection to the cooling system. A simple fan controller has also been included.

This tutorial uses the completed models (and calibration) of the previous tutorials. It is recommended to complete those tutorials first prior to starting this tutorial, which goes through how to complete a model to run as a transient thermal system.

7.1 Preparing a Transient Thermal Model

Open the model `Transient.gtm`, that can be found in the directory `..\tutorials\Modeling_Applications\Cooling_Thermal_Management\07-Transient`. This model is already set up as a closed loop thermal system (CHAPTER 5: Modeling a Steady-State Thermal System) using an 'EngineBlock-3Mass' (CHAPTER 3: Modeling Components of a Coolant Circuit) to represent the thermal mass of the engine, and the calibrated underhood model (CHAPTER 6: Modeling an Underhood System with COOL3D) replaced the simple heat exchanger.

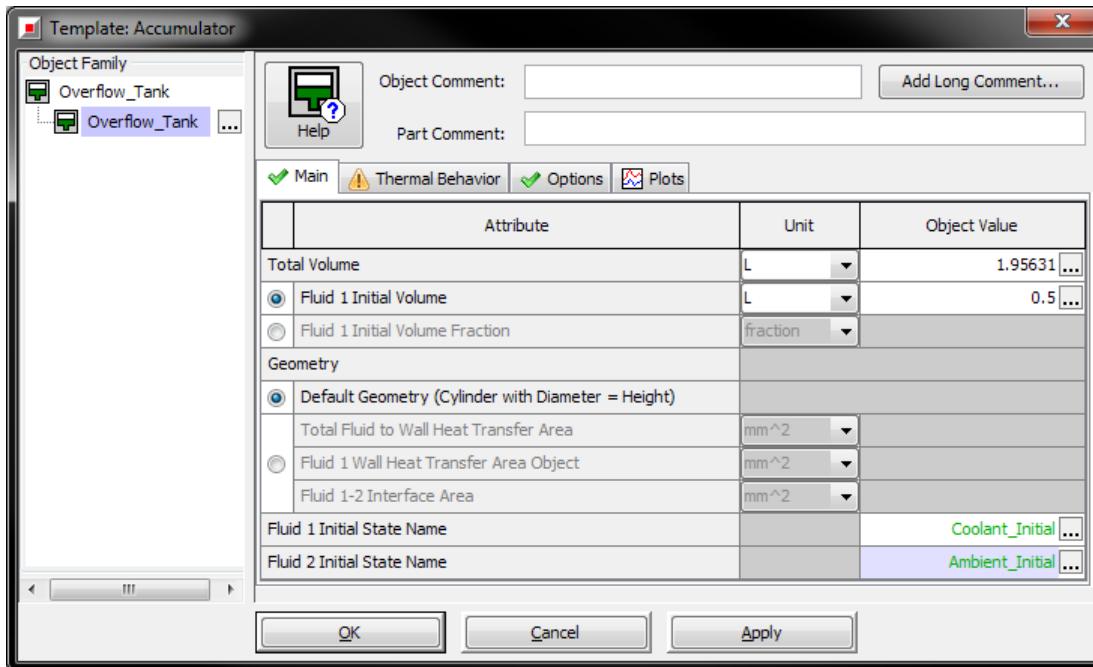
7.1.1 Modeling a Pressure Regulation Device (Accumulator or Overflow Tank)

An 'Accumulator' is also present in the model, which is required for a transient thermal system, to assist with the dynamic pressure regulation of the system. Unlike the previous tutorials, the 'Accumulator' is necessary in this case because the system is closed with no 'ClosedLoopPressure' or 'ClosedLoopFlow' parts, which were originally used to control the system pressure. As the engine rejects heat to the coolant, the fluid will expand. The 'Accumulator' (or expansion tank) is present to keep the system pressure in check.

Open the part 'Overflow_Tank', which connects the inlet of the radiator to the low pressure side of the cooling system. The **Main** folder contains the fluid volume, geometry, and fluid composition for the overflow tank. This information can be acquired from the CAD geometry of the tank or bottle. The non-default geometry option can be selected when heat transfer in the tank or bottle is extremely important, such as highly pressurized storage systems.

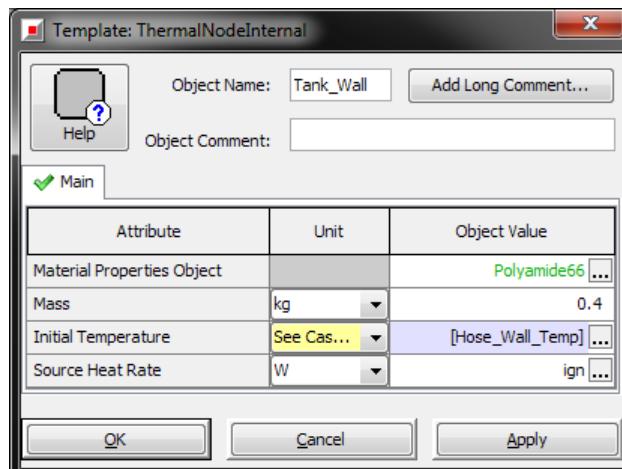


Tutorial 7: Modeling a Transient Thermal Model



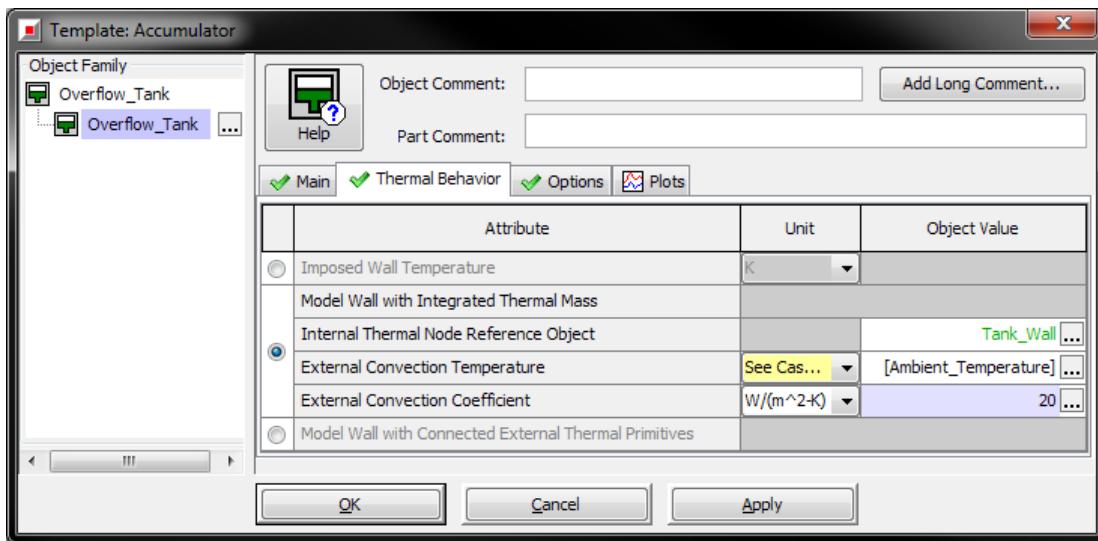
Note: The reference objects selected for the initial fluid states are important. Fluid 1 matches up with the Port 1 connection of the part, and fluid 2 matches up with the Port 2 connection of the part. If the references objects do not match with the correct port connection, then there can be solver instability in the model. In this model, the coolant side of the cooling system is connected at Port 1, and the ambient side is connected at Port 2 (to an 'EndEnvironment' boundary condition).

The **Thermal Behavior** folder is where the thermal solution for the overflow bottle can be created. Similar to the Calculated Wall Temperature in a pipe (5.2 Heat Transfer in Pipes (Wall Temperature Solver)), a material and external boundary conditions can be created to model the wall (material mass) of the overflow bottle. Select the 'ThermalNodeInternal' reference object for the attribute Internal Thermal Node Reference Object attribute. Complete the properties for the wall of the overflow bottle. Click OK when finished.



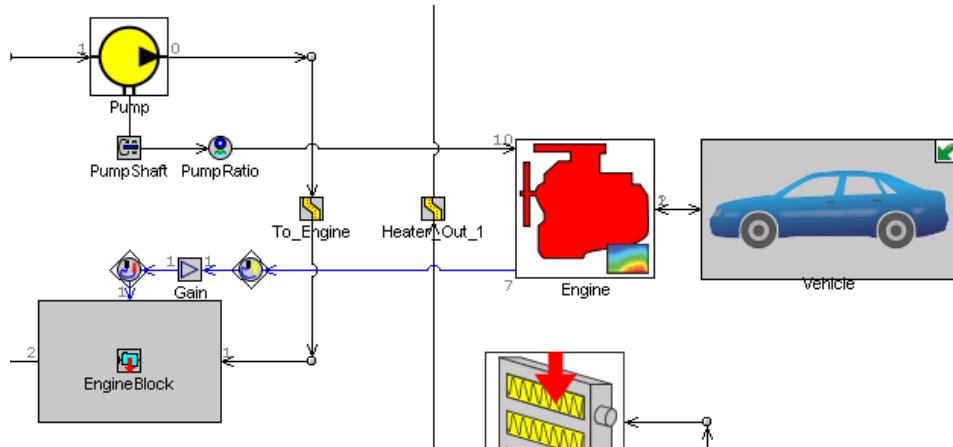
Tutorial 7: Modeling a Transient Thermal Model

Back in the 'Overflow_Bottle', complete the external boundary conditions for the wall. Click OK when finished.



7.1.2 Modeling the Engine and Vehicle

The mapped based engine (with heat rejection map) can be acquired from the engine performance group, or from a test bench rig. The vehicle (and transmission) model can be acquired from a vehicle group. Both the engine and the vehicle are integrated together with the cooling system. The pump is mechanically connected to the engine (gear ratio = 1.3), and the engine will pass the gas heat rejection to the engine block ('EngineBlock').



Note: Optionally, the engine can be modeled with a detailed (or FRM) GT-POWER model. Integration with the cooling system will be done through the template 'EngCylStrucCond'.

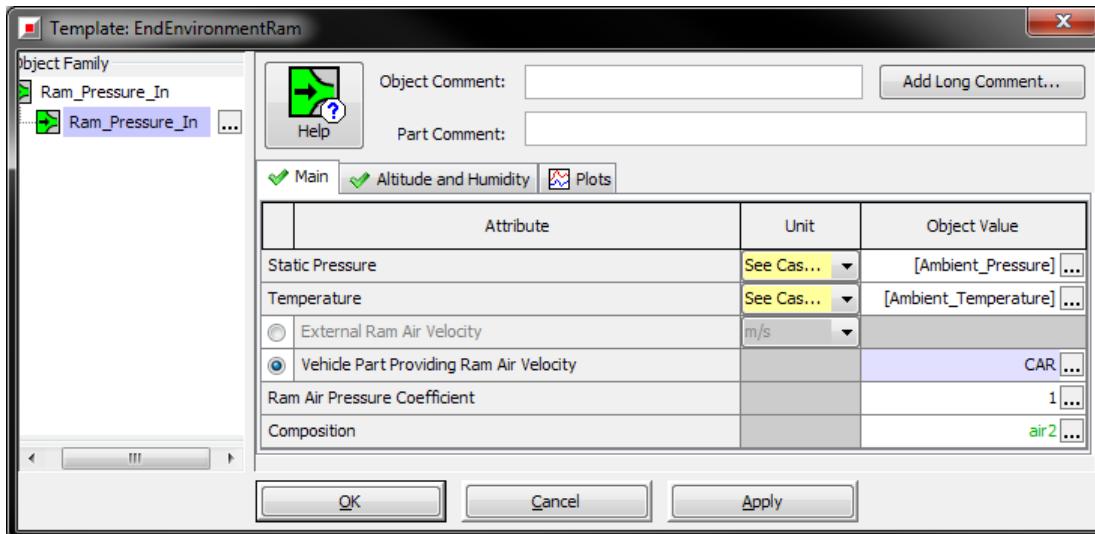
7.1.3 Imposing the Vehicle Speed Boundary Conditions for the Underhood System

With a vehicle in the model, the ram pressure boundary conditions for the underhood system can refer directly to the part name of the vehicle. This will allow the imposed velocity for the boundary conditions to automatically update as the vehicle speed adjusts.

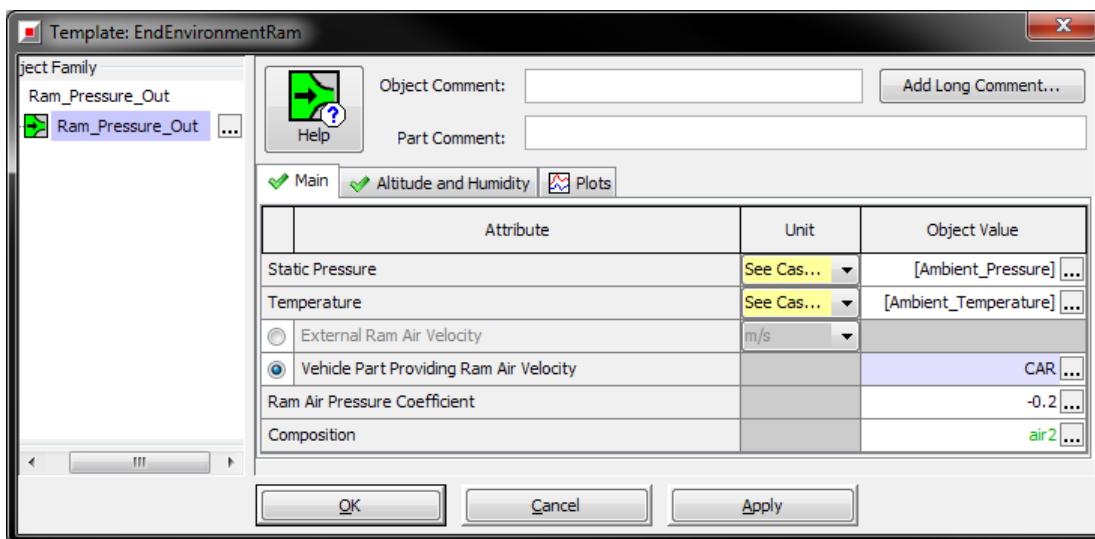


Tutorial 7: Modeling a Transient Thermal Model

Open the part 'Ram_Pressure_In', which is the inlet boundary condition for the underhood system. Select the attribute Vehicle Part Providing Ram Air Velocity, and point to the vehicle part (use the Value Selector to find the vehicle part name 'CAR'). This combination of inputs allows the vehicle velocity to be used for the inlet boundary condition. Click OK when finished.



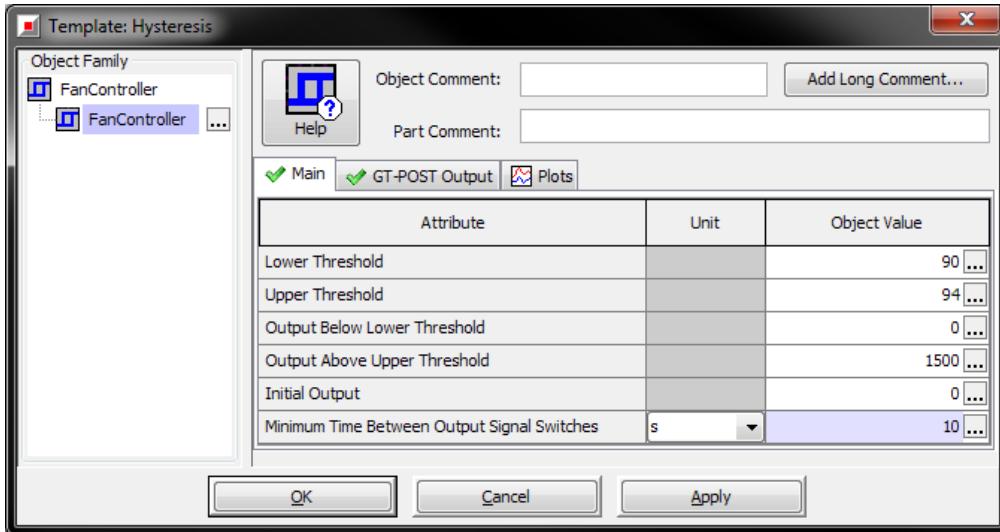
Perform the same steps to define the vehicle velocity boundary conditions for the outlet 'Ram_Pressure_Out' part. Click OK when finished.



7.1.4 Modeling a Simple Fan Controller

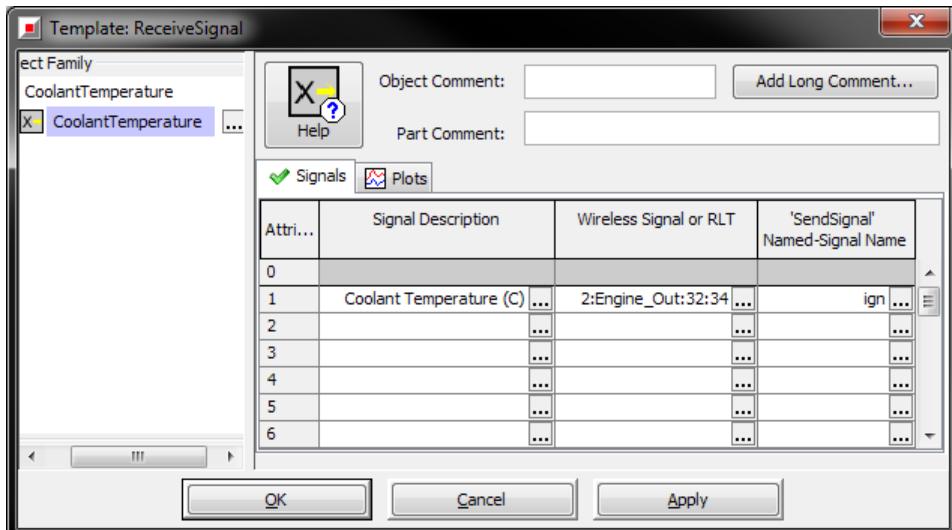
A simple fan controller (fan speed vs coolant temperature) is already linked to the speed boundary condition for the fan. Open the 'FanController' part, which is no more than a hysteretic lookup curve.





Using the coolant temperature as the input (not yet defined), the fan will turn on to 1500 RPM when the coolant temperature reaches 94 C, and will turn off (0 RPM) when the coolant temperature drops below 90 C. Click OK when finished.

Open the part 'CoolantTemperature', which is next to the 'FanController' part, to define the coolant temperature input location. Using the Value Selector for the Signal Name or RLT Name attribute, locate the Temperature RLT for the part 'Engine_Out', which is a 'FlowSplitGeneral' template. This part is found after the engine block in the project map. Select the units of C before clicking OK because the 'FanController' is set up to use Celsius as the unit. Click OK to accept the signal. Create a description for the signal in the Signal Description attribute, and then click OK to complete the coolant temperature signal.

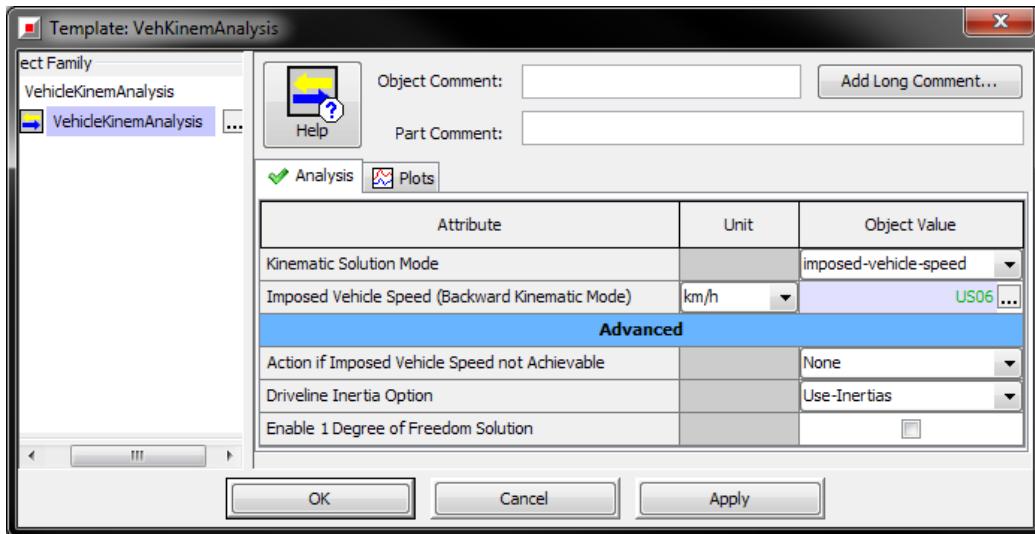


7.1.5 Setting the Driving Cycle Profile

The driving cycle profile is defined in the part 'VehicleKinemAnalysis', which can be found in the subassembly tab labeled 'Vehicle'. This part is where the vehicle settings are defined. Currently, the vehicle is set to follow an "imposed-vehicle-speed", and the profile can be set in the Imposed Vehicle



Speed attribute. Use the Value Selector to pick the 'US06' driving cycle from the template library. This profile is defined as km/h, so be sure to change the units of this attribute so it matched to the profile. Click OK when finished.



Note: It is possible to set the imposed driving cycle as a parameter so that it can be varied from one case to another in Case Setup.

7.1.6 Setting the CAC Boundary Conditions

The CAC boundary conditions in this case are imposed as a constant value. Realistically, the boundary conditions will be imposed as a profile transient to represent the driving cycle that is being used from measured test data, or integrated directly to a detailed GT-POWER engine model.

7.1.7 Coloring a Circuit (Optional)

Coloring a circuit is recommended to easily distinguish one fluid circuit from another in a model. This can be done by right-clicking on any flow volume part in the model, pick the Select Circuit for Part option, and then selecting the Color from the Part tab.

7.2 Run Setup and Output Setup

7.2.1 Run Setup

In Run Setup, the Maximum Simulation Duration needs to be defined to match the driving cycle that was selected. For the 'US06' driving cycle, the maximum simulation duration is 596 s. Additionally, because the goal is to model a transient thermal model, then Automatic Shut-Off When Steady State should be set to "off", which will automatically set the Thermal Solver in the **ThermalControl** folder to "transient". Click OK when finished.



Tutorial 7: Modeling a Transient Thermal Model

ODEControl		SignalControl		ThermalControl		ConvergenceRLT	
		TimeControl		Initialization		FlowControl	
	Attribute		Unit		Object Value		
	Time Control Flag				continuous		
<input type="radio"/>	Maximum Simulation Duration (Cycles)						
<input checked="" type="radio"/>	Maximum Simulation Duration (Time)	s		596	...		
	Automatic Shut-Off When Steady-State			off			
	Main Driver (Defines Periodic Frequency)						
<input type="radio"/>	Automatic						
<input type="radio"/>	Part Name						
<input type="radio"/>	Reference Object						
	Improved Solution Sequence for Multi-Circuit Models						

Note: The **FlowControl** folder is already set up to use multiple circuits as defined in 6.4.2 Creating Circuits for an Underhood Model. Column 1 is set for the internal fluid circuits (coolant and CAC), and column 2 is set for the external underhood circuit.

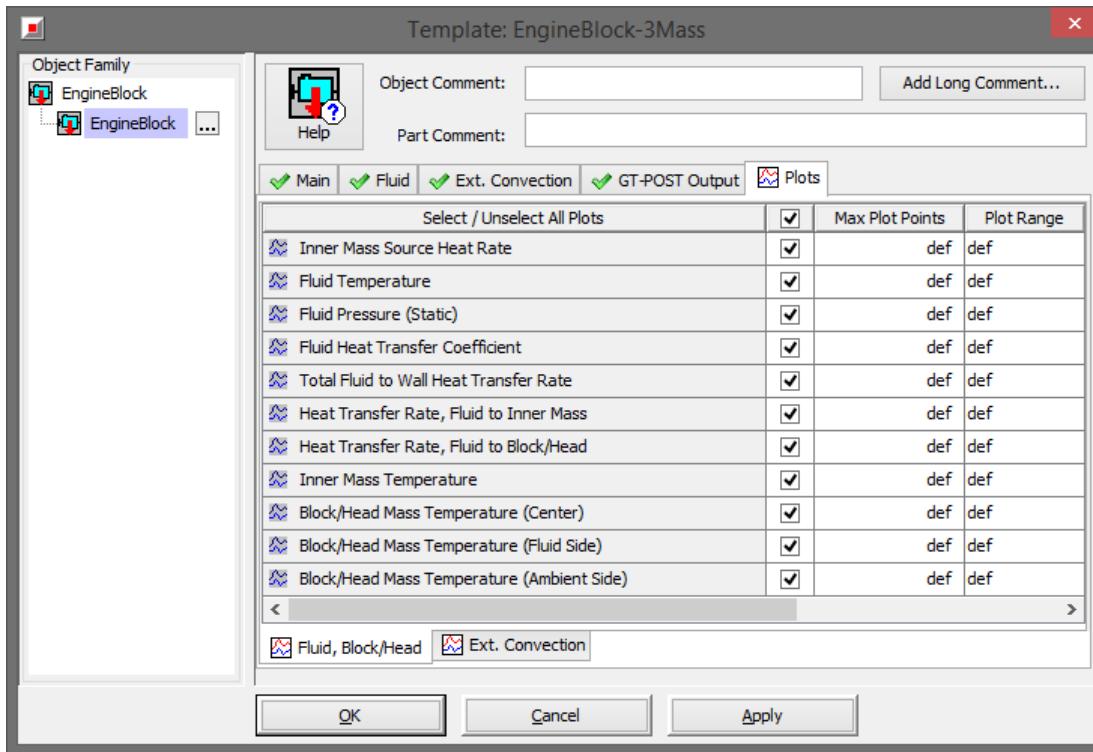
7.2.2 Output Setup

In Output Setup, the Time RLTs are already turned on to be stored for the simulation. This was done by enabling the box under the heading **Store Time RLT Results?**. The advantage to storing Time RLTs is that results for all parts in the model will be stored over the course of the simulation at the requested RLT Calculation Interval (to the Maximum Time RLT Data Storage Points) without needing to go to each part in the model to enable a plot (i.e. temperature, pressure, etc). However, this will result in a larger results file. Click OK when finished (nothing was changed).

ODE-Mech		ScoreboardRLTs		UserScoreboardRLTs		EndOfRunTables	
		Data_Storage		General		GT-POST_Setup	
	Attribute		Unit		Object Value		
RLT (Result) Data Storage Settings							
	RLT Calculation Interval (Continuous Circuits)	s		0.1	...		
	Store Case RLT Results?						
	Storage Level for All Parts NOT Listed Below			All			
<input checked="" type="checkbox"/>	Storage Level for "Exception" Parts Listed Below			All			
	"Exception" Parts List			ign	...		
	Store Time RLT Results?						
	Storage Level for All Parts NOT Listed Below			All			
	Storage Level for "Exception" Parts Listed Below			All			
<input checked="" type="checkbox"/>	"Exception" Parts List			ign	...		
	Maximum Time RLT Data Storage Points			500	...		
	Time RLT Storage Multiple (Periods)			1			
	List of RLT's to store (ALWAYS)			ign	...		
Plots and Table Data Storage Settings							
	Suppress Storage of Instantaneous Plots				<input type="checkbox"/>		
	Suppress Storage of Pre-Processed Plots				<input type="checkbox"/>		
	Suppress Storage of Tables (except EndOfRun Tables)				<input type="checkbox"/>		
General							
	Thinning Method for Limiting Large Arrays to Maximum Point Limit			Automatic			



Alternatively, if data storage is a concern, instantaneous time-based plots can be enabled in any part found on the project map. By double-clicking on a part and going to the Plots folder the requested plot will be stored if enabled. An example with 'EngineBlock' is seen below.



Data is only stored if the request is made. This is true for both instantaneous plots and RLTs. If it is found later that a results or plot is of interest, and it was not turned on, the model must be run again for the results and/or plot to be stored.

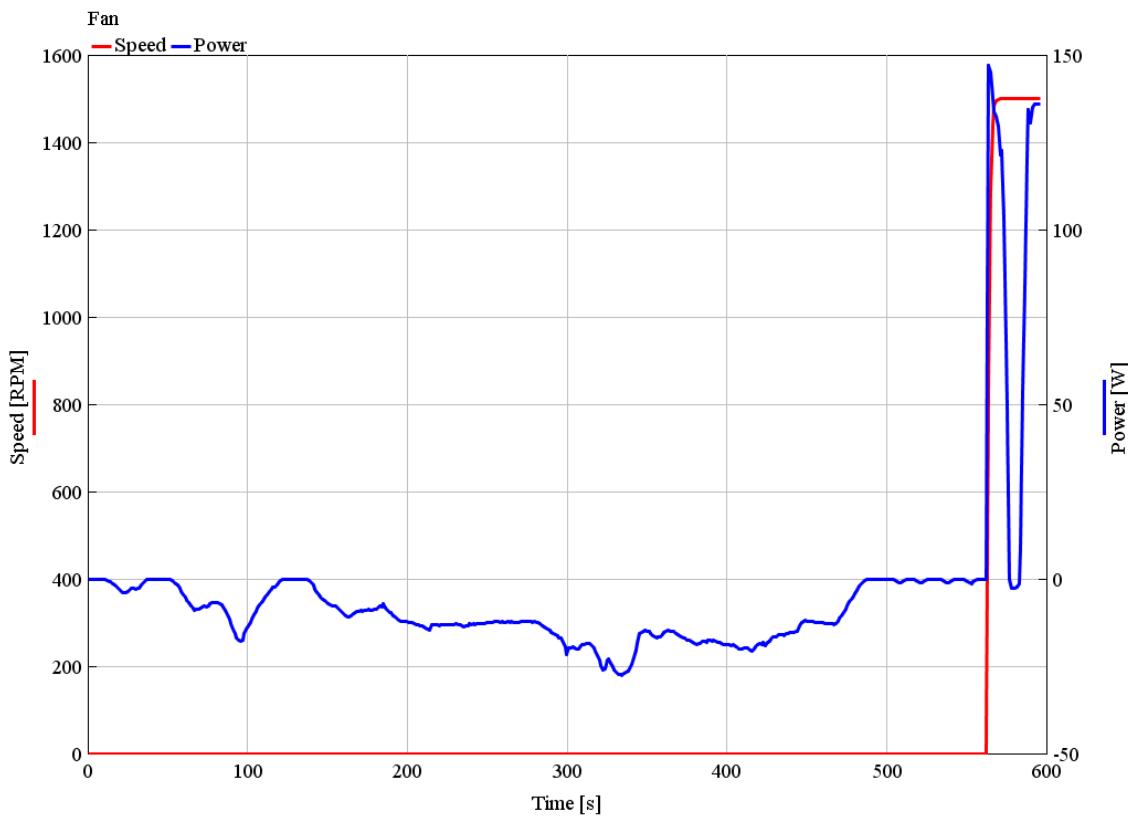
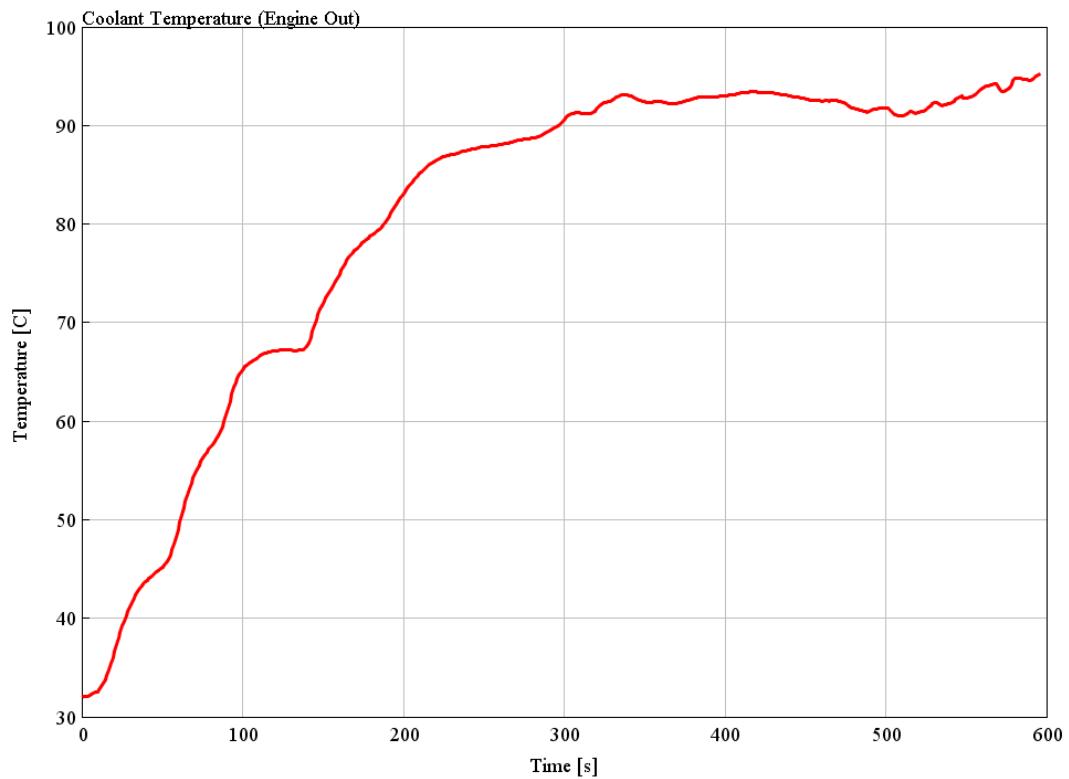
7.3 Testing a Transient Thermal Model

Case Setup is already complete with the test conditions for the model. If desired, feel free to add additional test conditions (i.e. another driving profile, swapping of parts in the underhood mode, etc.). Otherwise, the model can be run at this time to study the transient behavior of the cooling system over the 'US06' driving cycle. This simulation will take some time to run the full driving cycle. Open the results in GT-POST after the model has finished running.

With the model completed and open in GT-POST, one is able to view things like the warm-up time of the coolant temperature over the course of the driving cycle, the fan "on" time and power consumption, etc. (Note: The plots seen below have been created using a .gu file to merge Time RLT data sets.)



Tutorial 7: Modeling a Transient Thermal Model



Tutorial 7: Modeling a Transient Thermal Model

By being able to study these results, and others, one can use GT-SUITE to optimize the cooling system to achieve the desired warm-up rate, coolant operating temperature, fan power consumption, etc. for all operating conditions of the vehicle.

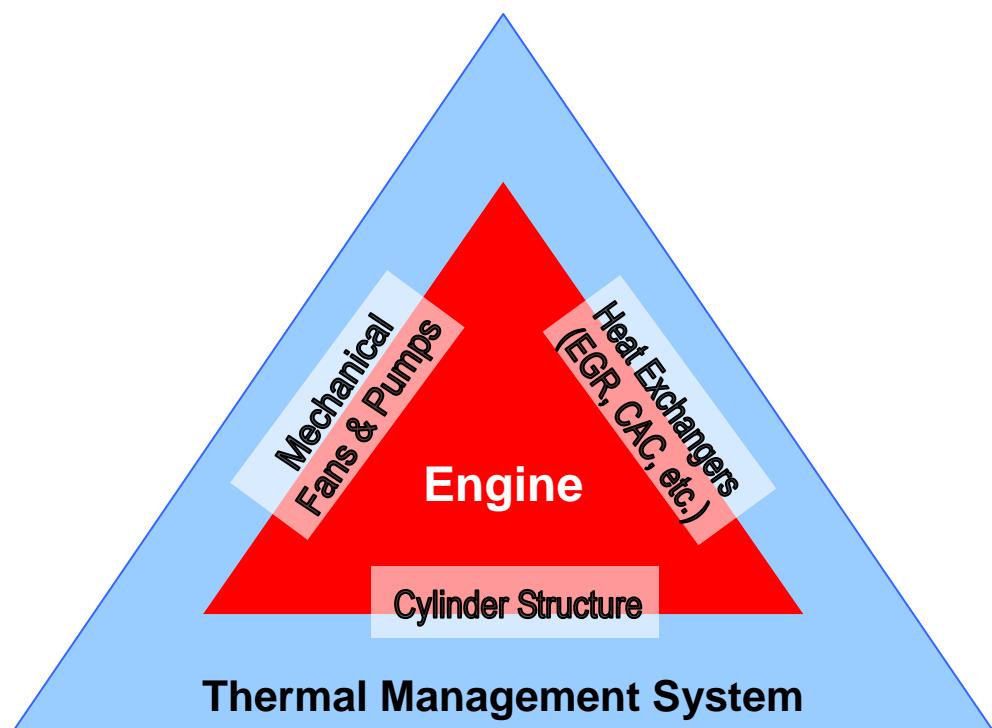


CHAPTER 8: Advanced Integration of Engine and Cooling Systems

8.1 Overview

An engine and the thermal management system that encompasses it have inherent interactions. These interactions can have a pronounced effect on both systems. For this reason, it is occasionally necessary to account for these interactions to accurately model the behavior of the entire assembly. This is possible in GT-SUITE by integrating the engine and cooling system models. In general, this involves the application of boundary conditions at several interfaces between the engine and thermal management system. GT-SUITE accounts for the interaction between the engine and the thermal management system in three key places as illustrated below.

1. **Cylinder structure:** This is the most obvious interaction. Heat transfer through the cylinder components affects both the brake power of the engine and the temperature rise of the coolant and oil.
2. **Heat exchangers:** This refers to heat exchangers which contain fluids from both the engine and thermal management system (i.e. charge-air-coolers, EGR coolers, etc.).
3. **Mechanical fan and pump shafts:** These require power and; hence, affect the accessory load on the engine based on the operating condition of the thermal management system. Also, the operating point of the fan and pump are limited by the engine operating point.



Tutorial 8: Advanced Integration of Engine and Cooling Systems

Some types of analysis do not require engines and thermal management systems to be integrated. In many instances, however, the integration of the engine model and the cooling system model is required. Examples of such analyses are listed below.

1. Allow realistic studies of impact on engine performance from ...
 - EGR cooler
 - charge-air-cooler
 - auxiliary load (mechanical pump, fan)
 - oil cooler
2. Vehicle performance in transient test cycles
 - Cylinder structure warm-up
 - Friction, Fuel Economy Studies
3. Cooling system control strategy evaluation
4. Engine block coolant flow strategies evaluation (series, parallel, cross-flow, etc.)
5. After-treatment devices

When an analysis does require integration, the fidelity necessary for the integration must also be considered. GT-SUITE offers two methods of integration, indirect integration and direct integration. These methods are shown in the table below. There are tradeoffs to both methods which must be considered when deciding which method to use. Indirect integration means that the engine model and the cooling system model are run separately. In contrast, direct integration means that the engine model and cooling system model are run simultaneously.

In general, during the initial stages of system design the indirect method is suitable. The indirect method is the faster of the methods with regards to solution time. The key trade off with the indirect method is that there is no feedback to the engine, so the effect the cooling system has on the engine performance is not accounted for. To completely capture the interactions of the two systems the direct method is necessary. The key trade off to direct integration is that the solution time is greatly increased as the engine model must be solved during the whole simulation.



Methods of Engine/Cooling System Integration		
	Indirect	Direct
Schematic	<pre> graph TD EM[Engine Model GT- POWER] --> DB[Database] DB --> CM[Cooling Model Cooling Systems] </pre>	<pre> graph TD subgraph DirectBox [] EM[Engine Model GT-SUITE] CM[Cooling Model GT-SUITE] EM <--> CM end </pre>
Main Features	<ul style="list-style-type: none"> • Engine and Cooling System models run separately • Run matrix of conditions with ENGINE, storing gas-side (steady-state) boundary condition data into database • Cooling System model extracts gas side boundary conditions during simulation and applies to finite element cylinder models • Fast solution times relative to direct method 	<ul style="list-style-type: none"> • Engine and Cooling System models merged and solved simultaneously • Fully transient boundary conditions (gas and coolant side) including charge-air-cooler and EGR cooler • Most accurate and complete integration between the engine and cooling performance • Easier to setup, but slower than indirect method
When to Use	<ul style="list-style-type: none"> • Models where interactions between cooling system and engine <u>are not</u> critical • Steady engine speed and load • Transient engine speed and/or load without turbocharging, CAC or EGR cooling • Passive cooling system controls 	<ul style="list-style-type: none"> • Models where interactions between cooling system and engine <u>are</u> critical • Transient engine speed and/or load with turbocharging, CAC and/or EGR cooling. • Active cooling system controls based on engine performance



8.2 Indirect (Partial) Integration

When using indirect integration, a cooling system model consisting of an engine cooling circuit and cylinder structures can reference in-cylinder gas boundary conditions from a previously run engine model. This is sometimes referred to as "partial" integration because the information is passed only in one direction – from engine to cooling system.

In lieu of GT-POWER engine simulation results to reference, a cooling system model can be run with user-imposed in-cylinder gas boundary conditions. If employing user-imposed gas boundary conditions you may skip the **Engine Model Setup** section and go directly to the **Cooling System Model Setup** section.

8.2.1 Engine (GT-POWER) Model Setup

Slight modifications to the engine model may be necessary to carry out the indirect integration. The following list describes the requirements of the engine model:

- **Wall Temperature Object:** The engine model must use the 'EngCylITWallSoln' object for the 'Wall Temperature defined by Reference Object' in each 'EngCylinder' part. This object parametrically defines the cylinder structure and approximates the coolant and oil boundary conditions at a steady state condition.
- **Intake and Exhaust Ports:** Care must be taken with 'Pipe*' parts directly upstream or downstream of the 'EngCylinder' parts. The thermal solution seen at the location of the 'Pipe*' parts must be passed to the FE model, and vice versa. It is recommended to check the option "Full Thermal Solution Between FE Walls and Gas" in the 'FECylinderStructure' object found in the corresponding 'EngCylinder' part.
- **Engine Friction:** Both the engine model and the cooling model will account for engine friction. To improve accuracy, both models should use the same friction model. See the table below for a description of how friction models can be applied.



Engine Friction Model Application

	Engine Only	Indirectly Integrated Engine/Cooling Systems	Directly Integrated Engine/Cooling Systems
Chen-Flynn Model	Automatically sends friction energy to the finite element cylinder structure component (independently calculated for each cylinder). Ratios for friction to piston rings and skirt are defined in the 'EngCylITWallSoln' object.	N/A	Must make connection from EngineCrankTrain to EngCylStrucCond(s). Each connection receives the total cylinder friction for the whole engine, so a multiplier must be applied to reduce the friction to an individual cylinder value. Ratios for friction to piston rings and skirt are defined in the EngCylStrucCond object.
EngineFrictionDetail Model	Automatically sends friction energy to the finite element cylinder structure component (evenly distributed to each cylinder). Ratios for friction to piston rings and skirt are defined in the EngCylITWallSoln object.	Must make connection from EngineState to EngCylStrucCond(s). A multiplier must be applied to reduce the friction to an individual cylinder value. Ratios for friction to piston rings and skirt are defined in the EngCylStrucCond	Must make connection from EngineCrankTrain to EngCylStrucCond(s). Each connection receives the total cylinder friction for the whole engine, so a multiplier must be applied to reduce the friction to an individual cylinder value. Ratios for friction to piston rings and skirt are defined in the EngCylStrucCond object.
External Controls	Actuate the FMEP input signal link on 'EngineCrankTrain' from an external friction lookup map or equation.	Actuate the FMEP or Friction Multiplier input signal link on 'EngineState' from an external friction lookup map or equation.	Actuate the FMEP input signal link on 'EngineCrankTrain' from an external friction lookup map or equation.

Once the above requirements are met, the GT-POWER engine model should be run through a set of cases that encompass the entire engine operating range or at least the engine operating range of interest. These cases can be easily set up using the Design of Experiments (DOE) feature in GT-ISE. The engine model will then generate a database of in-cylinder gas boundary conditions. This information is automatically stored and later extracted from the *.gdx file.



8.2.2 Cooling System Model Setup

The indirect integration in cooling systems is performed by referencing the in-cylinder gas boundary conditions for the finite element structure from a previously run GT-POWER model. The cooling system model used for this purpose will generally consist of six main sections as listed below. The important details of each section will be discussed.

- Engine cylinder structure
- In-cylinder gas boundary conditions
- Engine operating conditions
- Coolant circuit
- Oil circuit
- Connections

Engine Cylinder Structure - 'EngCylStrucCond': The cooling system model must contain a finite element model for each cylinder structure (each consists of a cylinder wall, head, piston, valves, and ports). This finite element model, 'EngCylStrucCond', is identical to the reference object 'EngCylITWallSoln' defined in the GT-POWER model. The same 'FECylinderStructure' reference object used in 'EngCylITWallSoln' of the GT-POWER model can be reused in 'EngCylStrucCond'.

Cylinder Gas Boundary Conditions - 'EngCylGasBCs': In-cylinder gas boundary conditions must available for each cylinder structure. The 'EngCylGasBCs' part provides these boundary conditions to the 'EngCylStrucCond' part. One 'EngCylGasBCs' part should be used for each 'EngCylStrucCond' part. To reference a GT-POWER model, the **Main** folder of the 'EngCylGasBCs' part must be set to use the lookup from previously run engine model results. The filename for the appropriate GT-POWER model output file (*.gdx) must be specified.

'EngCylGasBCs' allows a 3rd Independent RLT Variable to look up the in-cylinder boundary conditions (in addition to the engine speed and IMEP). Models that use this third variable require a large number of data points (i.e. many cases for the engine model). If the boundary conditions results are sparse, the look up method used to interpolate from the data may not return acceptable results for the in-cylinder boundary conditions.

Although it is not recommended, if GT-POWER results are not available or a user does not desire to reference GT-POWER results, the impose boundary conditions in the **Main** folder of the 'EngCylGasBCs' must be selected. The user must then enter temperature and heat transfer coefficient boundary conditions for all in-cylinder finite element structure surfaces. No 'EngineState' object is needed for this configuration.

Engine Operating Conditions - 'EngineState': An 'EngineState' part must be placed in the model to provide instantaneous engine operating conditions to the 'EngCylGasBCs' parts. The in-cylinder gas boundary conditions are interpolated from the GT-POWER output file based on engine speed and load (IMEP). The boundary conditions are then imposed on the FE structure defined by the 'EngCylStrucCond' part.



Tutorial 8: Advanced Integration of Engine and Cooling Systems

The **State** and **Secondary Maps** folders in the 'EngineState' part have three attributes related to engine maps that should be completed. The information for these three maps ('Mechanical Output Map', 'Engine Friction Map' and 'Fuel Consumption Map') may be extracted from the GT-POWER model outputs.

Coolant Circuit: The engine block and head coolant passages should be constructed from flow parts representing the fluid passages and thermal parts such as the 'ThermalMass'. The linkage of these components to the 'EngCylStrucCond' parts accounts for the coolant and other external boundary conditions on the cylinder structure. While linking the flow and thermal components to the 'EngCylStrucCond' part, the connecting port numbers must be selected properly by double clicking on the connection link. The port numbers indicate the boundary condition location (ex. water jacket, value guide coolant, etc.). The remaining vehicle thermal management system including pumps, heat exchangers, thermostats, etc. can also be constructed and connected as needed.

Oil Circuit: The engine oil circuit should be constructed with flow parts and thermal parts. The 'EngCylStrucCond' part should be linked to these components to model heat transfer from the lower cylinder wall and back of the piston to the oil. As with the coolant circuit, the connection port number must be selected by double clicking on the connection link. The oil circuit ports are 'Liner Oil' and 'Piston Oil'.

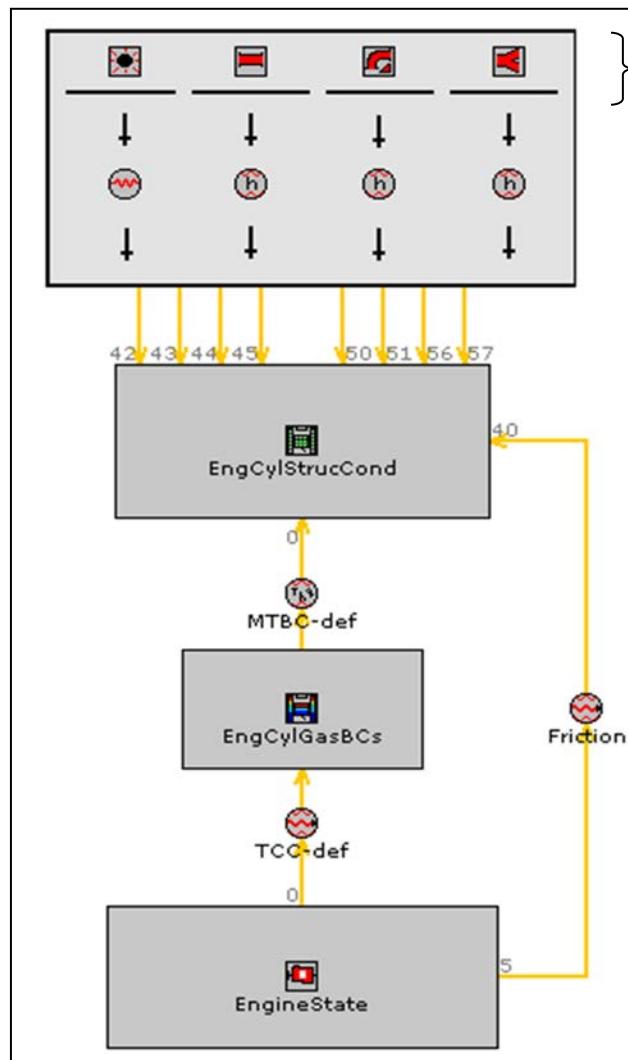
Connections: An automatic connection object ('ThermalCompConn') will be placed between the 'EngineState' part and each of the 'EngCylGasBCs' when in link mode. Also, an automatic connection object ('MultipleThermalBConn') will be placed between the 'EngCylGasBCs' parts and the 'EngCylStrucCond' parts when in link mode. This connection should be set to port zero at the 'EngCylStrucCond'. Ports 1 through 33 of 'EngCylStrucCond' should only be used in advanced simulations that do not reference GT-POWER results.

A connection at Port #40 of the 'EngCylStrucCond' parts with a 'ThermalCompConn' connection is also necessary to capture the friction energy from 'EngineState'. This connection transfers cylinder friction energy to the cylinder structure. Alternatively, port #40 on 'EngCylStrucCond' can be made from an external lookup map instead of 'EngineState'. CAUTION: The friction value from the 'EngineState' is the friction for all cylinders; therefore, a heat rate multiplier should be used in the 'ThermalCompConn' to reduce the value accordingly for each cylinder.

Both flow and thermal components can be connected to ports 34 through 66 of the 'EngCylStrucCond' part. See the online help for the 'EngCylStrucCond' template for a description of these ports. Only one connection can be created at each port. If one of these ports does not have a connection to it, then that port is assumed to be perfectly insulated.

The image below shows what the indirect integration of the cooling system with the GT-POWER results will look like.





These thermal components can be connected through thermal connections to the 'EngCylStrucCond' object to represent boundary conditions at the (ports related to different FE surfaces):

- Cylinder Friction (port #: 40)
- Head Coolant (port #: 43)
- Valve Guide Coolant Port (port #: 44-48)
- Pipe Port Coolant Port (port #: 50-54)
- Liner Water Jackets (port #: 56, 57)
- etc.

Fig.: 1 References:

MTBC-def ('MultipleThermalBConn') and TCC-def ('ThermalCompConn') default connection parts are set automatically when linking coupling objects. The connections may appear as 'dots' depending on the settings in the Tools -> Options -> View folder (Display default connection as icon).



8.3 Direct (Full) Integration

When using direct integration, a GT-POWER engine model and a cooling system model are combined (copy and paste) on one project map. This allows more accurate transient simulations to be performed. This is sometimes referred to as "full" integration because the information is passed in both directions and, therefore, available to both the engine and cooling model. A GT-SUITE license is required to run a "full" integrated model.

8.3.1 Engine Model Setup

Slight modifications to the engine model may be necessary to carry out the direct integration. The following list describes the requirements of the engine model:

- **Wall Temperature Object:** The engine model must use the 'Wall Temperature defined by FE Structure part' in each 'EngCylinder' part. This option will allow the matching 'EngCylStrucCond' part to share the same FE solution.
- **Engine Friction:** Three types of friction models are available for direct integrated models. See the table in the indirect integration section more information.

8.3.2 Cooling Model Setup

The cooling model used for this purpose will generally consist of four main sections as listed below. The important details of each section will be discussed.

- Engine cylinder structure
- Coolant circuit
- Oil circuit
- Connections

Engine Cylinder Structure - 'EngCylStrucCond': The cooling system model must contain a finite element model for each cylinder structure (each consists of a cylinder wall, head, piston, valves, and ports). The gas boundary conditions will be automatically determined from the connected 'EngCylinder' part.

Coolant Circuit: The engine block and head coolant passages should be constructed from flow parts representing the fluid passages and thermal parts such as the 'ThermalMass'. The linkage of these components to the 'EngCylStrucCond' parts accounts for the coolant and other external boundary conditions on the cylinder structure. While linking the flow and thermal components to the 'EngCylStrucCond' part, the connecting port numbers must be selected properly by double clicking on the connection link. The port numbers indicate the boundary condition location (ex. water jacket, value guide coolant, etc.). The remaining vehicle thermal management system including pumps, heat exchangers, thermostats, etc. can also be constructed and connected as needed.

Oil Circuit: The engine oil circuit should be constructed with flow parts and thermal parts. The 'EngCylStrucCond' part should be linked to these components to model heat transfer from the lower cylinder wall and back of the piston to the oil. As with the coolant circuit, the connection port number



must be selected by double clicking on the connection link. The typical oil circuit ports are 'Liner Oil' and 'Piston Oil'.

Connections: Both flow and thermal components can be connected to ports 34 through 66 of the 'EngCylStrucCond' part. See the online help for the 'EngCylStrucCond' template for a description of these ports. Only one connection can be created at each port. If one of these ports does not have a connection to it, then that port is assumed to be perfectly insulated.

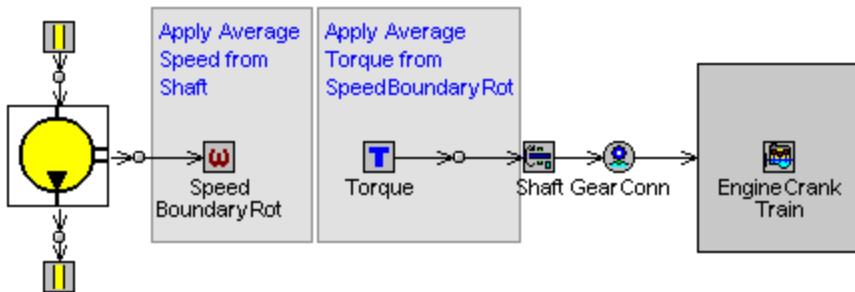
8.3.3 Integrating Engine and Cooling Model

The license used by one of the models should be changed to GT-SUITE. This can be done from the **Change License** option in the **File** tab. At this point, the second model can be copied and pasted onto the map of the first model.

Connections between the engine and the cooling system must be made at the interfaces between the engine and thermal management system (cylinder structure, heat exchangers, and mechanical fans and pumps).

- **Cylinder Structure:** A 'MultipleThermalBConn' is used to connect each 'EngCylinder' part to its corresponding 'EngCylStrucCond' part.
- **Cylinder Structure:** Each 'EngCylStrucCond' part may be connected to the 'EngineCrankTrain' with a 'ThermalCompConn' connection to model the friction. The port number used at the 'EngineCrankTrain' should be port #91. Port# 40 should be used at the 'EngCylStrucCond' parts. Alternatively, port #40 on 'EngCylStrucCond' can be made from an external lookup map instead of 'EngineCrankTrain'. CAUTION: The friction value from the 'EngineCrankTrain' is the friction for all cylinders; therefore, a heat rate multiplier should be used in the 'ThermalCompConn' to reduce the value accordingly for each cylinder.
- **Heat Exchangers:** A link should be created to connect 'HxMaster' and 'HxSlave' parts from the engine and cooling circuits together.
- **Mechanical Fans and Pumps:** Although it is possible to directly link auxiliary pumps or fans to the engine crankshaft, generally it is preferred to make an indirect link using 'RLTDependence*' objects or controls. This avoids aliasing of the engine speed (which will fluctuate within 1 engine cycle if the engine is in "load mode") when it is sampled at a much lower frequency for the cooling circuit (and not necessarily at the same point within each engine cycle). The speed of the pump/fan should be imposed using a 'SpeedBoundaryRot' part with an 'RLTDependence*' which points to a 'Shaft' (average shaft speed) that is connected to the 'EngineCrankTrain'. A 'Torque' on the 'Shaft' can be used to apply the average pump/fan load back to the 'EngineCrankTrain'.





8.3.4 Other setup issues

After combining and connecting the models on the same map there are a few additional issues to be addressed in Run Setup and Output Setup.

Run Setup: The Time Control Flag in Run Setup should generally be set to "periodic". Assuming that the model will be used for transient simulations (varying engine speed), the **Improved Solution Sequence for Multi-Circuit Models** should be turned "on". This setting enables the Time Control Flag for the cooling circuits to be overridden to "continuous" within the FlowControl folder (the engine and cooling circuits should be separated in the FlowControl folder). The time step for the cooling circuits may then be imposed in seconds (not in crank angle degrees) and can be left to "def" if desired. The **Thermal Solver** should also be set to "transient" if the **Improved Solution Sequence for Multi-Circuit Models** is "on".

After running the model for the first time, the time step should be checked for each circuit. If the cooling circuits do not have the expected step size that was imposed in Run Setup, then it is possible that the step size was limited by an ODE circuit (mechanical or controls). If this occurs, the **Maximum Ratio of Flow/ODE Time Step** attribute in the 'ODEControlExplicit' object (ODEControl folder of Run Setup) can be set to "ign" to avoid the unwanted time step limitation.

Output Setup: The RLT Calculation Interval (Continuous Circuits) should be set to define the interval over which RLT's are calculated for all continuous circuits. It is recommended to pick a value that is equivalent to the time step taken by the cooling circuits.



CHAPTER 9: Cooling System FRM Creation

9.1 Overview

This tutorial has been prepared to provide guidelines on how to convert a typical vehicle cooling circuit, or other hydraulic circuit model, to a Fast-Running-Model ("FRM") by reducing complexity while maintaining similar system results. This is useful when modeling a complex hydraulic system whose layout is "fixed," and detailed flow results are not of interest. Application examples include:

- Hydraulic systems within SiL or HiL applications
- Long transients, e.g. long drive cycle durations
- Integrated vehicle system models, e.g. energy management

The method of conversion/simplification presented here reduces runtime by reducing the total number of flow volumes in the model; it is most useful when modeling a cooling or hydraulic system containing many flow volumes. For example, models containing many hundreds of subvolumes have been observed to achieve up to 5x-10x speed improvement after conversion. Flow networks converted and imported from GEM3D often contain many complex pipe shapes, and use many subvolumes to model accurately; this type of model is a good candidate for FRM conversion.

9.2 Prepare System Model

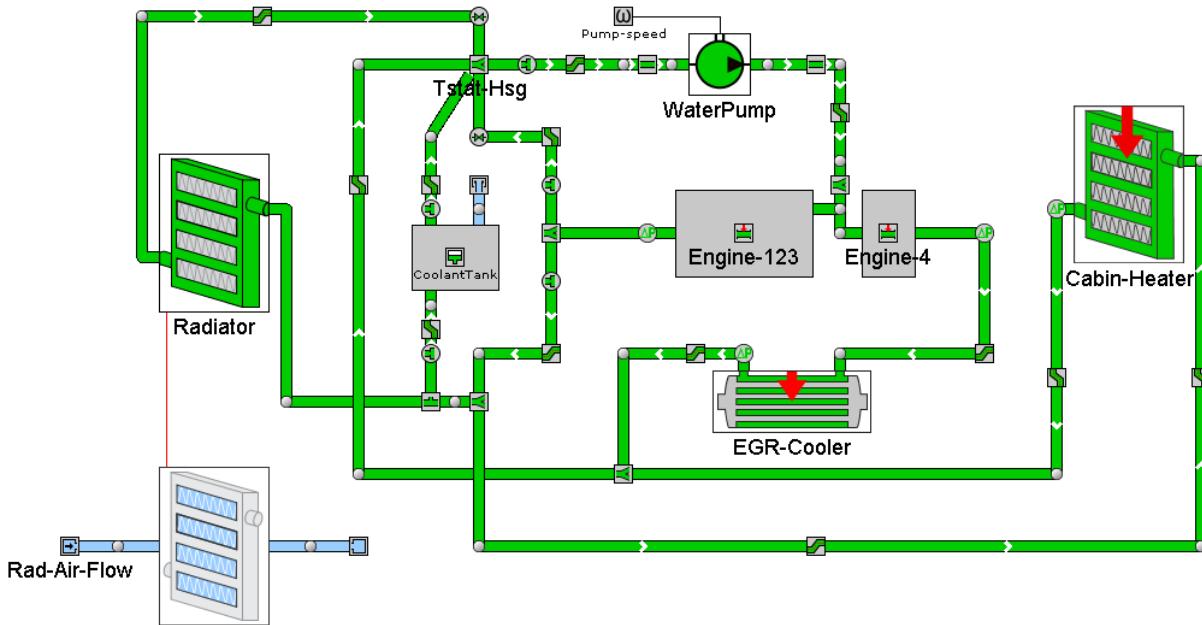
The creation of a FRM would typically only be of interest when a "full system" model of some complexity already exists. In this tutorial we will start with the full system model called *CoolingFRMTutorial.gtm*. Open this model in GT-ISE. Note that the model has been set up to run at several operating conditions to observe the minimum and maximum expected flow rate/Reynolds number conditions in the various flow branches of the system; each of the following parameters are run at a "high" and "low" condition (these limits may be adjusted as needed for other models - the goal is to capture the "extreme" running conditions of the full system model):

- Coolant temperature: 300K and 373K
- Pump speed: 1000 and 5500 RPM (lowest and highest mapped speeds)
- Thermostat lift: 0 and 10mm (fully closed and fully open)

In order to ensure that the fluid temperature remained constant in each case, and through the full system, all heat inputs were set to 0 in 'HeatAddition' parts (the engine and EGR cooler), and the "Ignore Energy Input to Fluid" attribute was checked in the "Options" folder of the 'WaterPump'. Additionally, the thermal wall solver was deactivated by setting the "Global Heat Transfer Multiplier" attribute to 0 in the "Flow Settings" folder of the 'FlowControlImplicit' reference object "Implicit."

To follow along with this tutorial, save a copy of the model with a new name, and run it.





Full system model, before changes (297 coolant circuit volumes)

View the results in GT-POST, and note the minimum and maximum coolant pressures in the system across all of the runs. This will help to select a meaningful choice of operating conditions when simplifying the model:

Minimum pressure (Case 3) = 0.91 bar
 Maximum pressure (Case 8) = 1.96 bar

9.3 Choose the First Flow Branch to Simplify

Before simplifying the model, we will first define what is meant by the term "flow branch." For the purposes of this process, a flow branch must:

- consist of one or more flow parts connected in series
- begin and end with a "component" (not a "connection")
- have only one inlet and one outlet (i.e. no 3-port flowsplits, flow nodes, etc.)
- not contain any variable-flow valves or other parts (e.g. thermostat, check valve, pump, etc.)
- not contain any parts whose heat transfer is significant (e.g. radiator or other active heat exchanger)

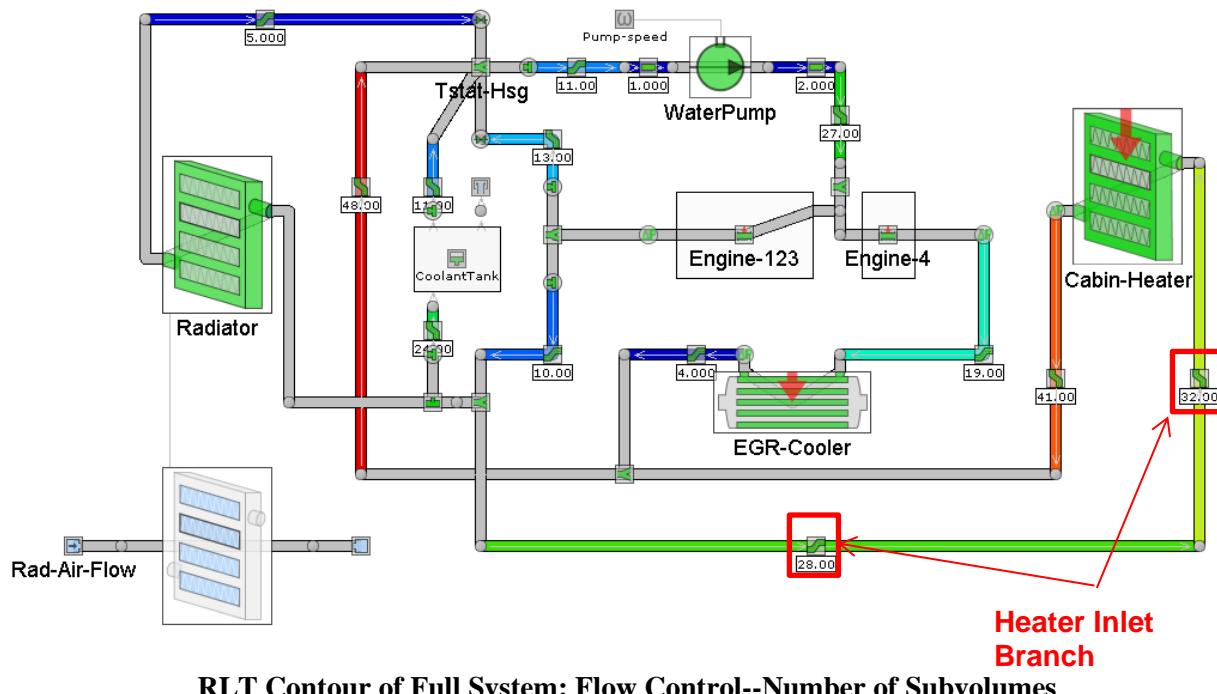
Note that these requirements do not exclude components such as a 'Hx*' or 'HeatAddition' parts, provided that heat transfer is not of interest and is not included in the full system model. Similarly, a non-default 'OrificeConn' or a valve may be included in a flow branch, provided that its position remains fixed across the entire range of full system operating conditions, and it is not the first or last part of a flow branch.

Viewing the "Flow Solution Numerics" table in GT-POST shows that the full system model contains 297 total flow volumes in the "Coolant" circuit. In order to judge which flow branches offer the best potential for improvement, in GT-POST, view the RLT Contour Map for Flow Control--Number of Subvolumes.



This will show which flow branches have the most subvolumes, and offer the most potential for simplification.

The heater inlet branch appears to offer the biggest potential improvement, as the 2 'PipeTable' parts ("Heater_In_1" and "Heater_In_2") contain a total of 60 subvolumes. We will choose this branch to simplify first. Note that the heater inlet branch could be combined with the "Cabin-Heater," "PrLoss_Cabin_Heater," and the outlet branch to form a single simplified flow branch if the heat transfer of the "Cabin-Heater" is not modeled in the full system model. But for this tutorial, we will assume it is of interest, and simplify the inlet and outlet branches separately.

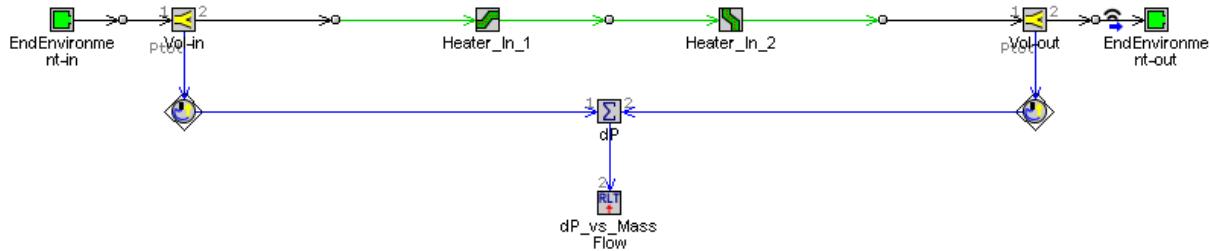


9.4 Setup and Run the "Virtual Flowbench" Model

From the tutorial directory, open the "virtual flowbench" model called *FlowBench-Template.gtm*. Save it with a new filename, for example, to indicate which branch is being simplified (you may wish to save a copy of the flowbench model for each branch), and follow these steps:

1. Select the 2 parts in the heater inlet branch (including the 'OrificeConn' between them) in the full system model and copy them. Paste them into the flow bench model and link them as shown below, making sure that flow direction through the branch is the same as in the system model. Any parameters present in the pasted parts may be pasted safely; if there are conflicting parameters, choose the "Keep Existing" option to maintain the values present in the flow bench model. If given the choice, do not append cases when pasting.





Heater inlet flow branch, as pasted and linked into flow bench model

2. In both 'PipeTable' parts:
 - In the "Main" folder, replace the "Initial State Name" attribute with the 'FlowBenchInit' reference object.
 - In the "Thermal" folder, delete the entries for "Wall Temperature Solver Object" and "Initial Wall Temperature" attributes, and select the "Adiabatic" option instead.
3. Open "Case Setup" and enter parameter values for Case 1 only in the "Inputs" folder (Cases 2-10 will be automatically populated correctly):
 - [P-max-system]: Maximum expected system pressure (absolute). Results from the baseline model show that the maximum system pressure was 1.96 bar; we will round up and enter 2. *Note that certain equations depend on this parameter, so it is recommended to maintain units of "bar." If different units are chosen, corrections must also be made for the parameters in the "FlowBenchParams" folder of Case Setup.*
 - [P-min-system]: Minimum expected system pressure (absolute). Results from the baseline model show that the minimum system pressure was 0.91 bar; we will enter 0.9.
 - [Exp-diam-inlet]: Diameter of first flow subvolume in system; in this case, the diameter at Distance = 0 (first column in the "Geometry" folder) for "Heater_In_1" is 15mm.
 - [Exp-diam-outlet]: Diameter of last flow subvolume in system; in this case, the diameter at Distance = 825.0105 (last column in the "Geometry" folder) for "Heater_In_2" is also 15mm.
4. In the "Inputs" folder of Case Setup, verify that fluid type and temperature are appropriate for the system being modeled. The default values are 'egl-5050' fluid at 50°C, which is appropriate for this model.
5. Run model.

Now open the results for the flow bench simulation in GT-POST. View the Case RLT results for Reynolds Number for either of the parts, and verify that it covers the Re range seen in the full model for that part. For example, the Re range in the flow bench model for "Heater_In_1" is approx. 190-64,000. The full system model shows a Re range of approx. 900-25,000 for the same part, so we can be confident that the full range is covered.

NOTE: By entering the max. and min. system pressures as the boundary conditions for a single flow branch, we are consistently over-estimating the total pressure drop through any single flow branch of a model. Although this is a fast and easy method, a better flow bench representation may be obtained by more accurately evaluating the max. and min. pressures across each individual branch. For example, in the full system model for this tutorial, evaluating these pressures across the heater inlet branch gives:

Minimum outlet pressure (Case 5) = 1.00 bar
 Maximum inlet pressure (Case 8) = 1.53 bar

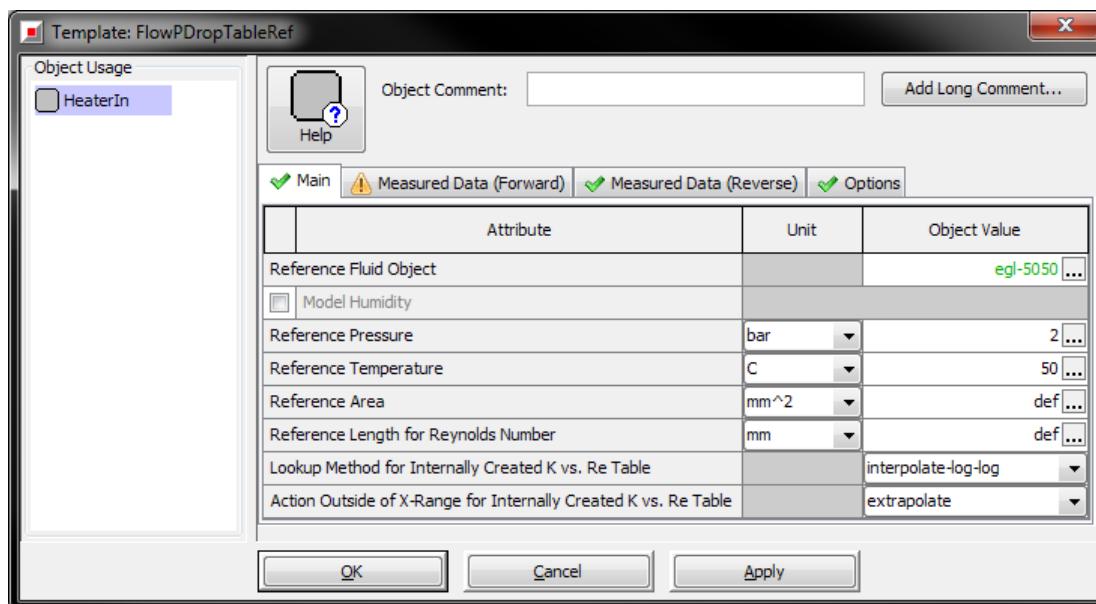


This gives a ΔP of 0.53 bar, rather than the 1.05 bar used here in the full system pressure method, which will result in a Re range more closely matching that of the real operating conditions.

9.4.1 Create Pressure Drop Reference Object

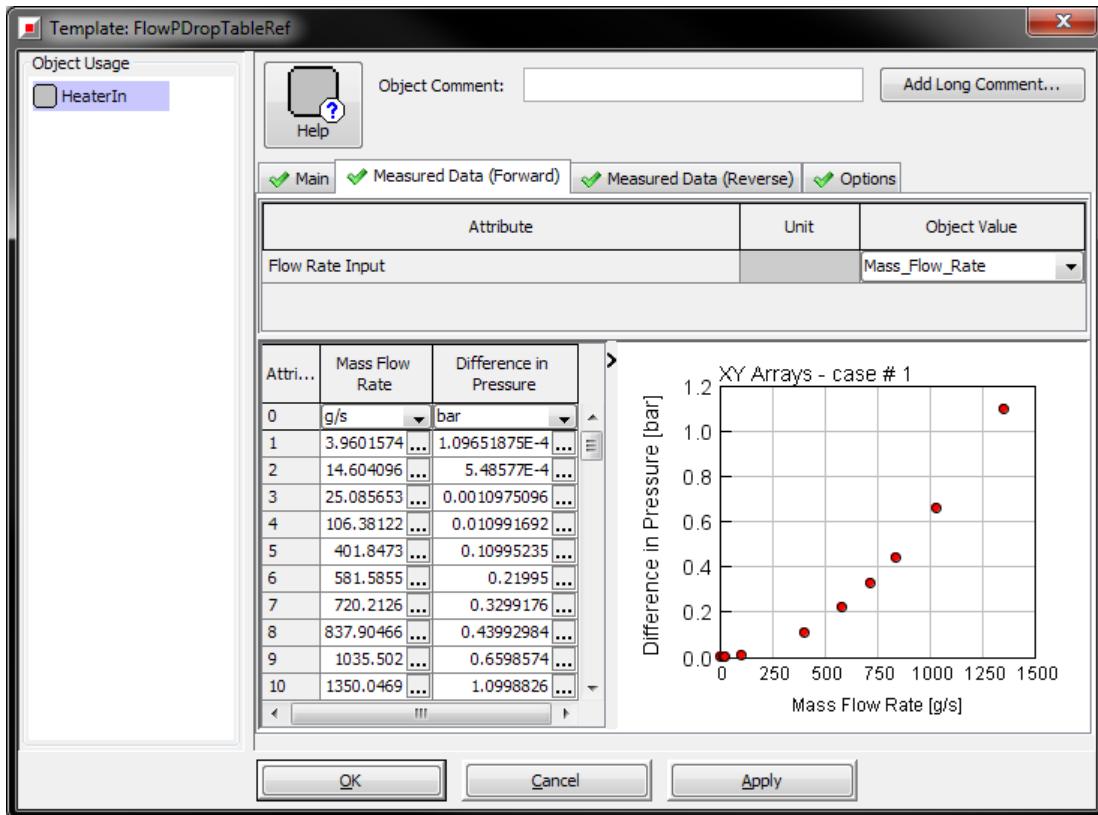
Next, use flow bench results to creating a pressure drop reference object in the full system model:

1. From the flow bench model, copy the 'FlowPDropTableRef' object called "Template," and paste it into the full system model. Rename the object to indicate the flow branch, for example, "HeaterIn" in this case.
2. Open the object, and in the "Main" folder, enter the reference fluid, pressure, and temperature values; note that these reference conditions should match the [Fluid], [T-Fluid], and [P-max-system] flow bench parameters:



3. In the "Measured Data (Forward)" folder, paste the Case RLT data from the 'RLTCreator' part called "dP_vs_MassFlow" in the flow bench results. Note the units of g/s and bar in the 'RLTCreator':





4. Click OK to finalize the 'FlowPDropTableRef' reference object.

NOTE: You could also copy the 'FlowPDropTableRef' template reference object into the working model, and customize the "Main" folder for that particular model, then using the "Copy and Edit Object..." option each time a new one is created for each flow branch.

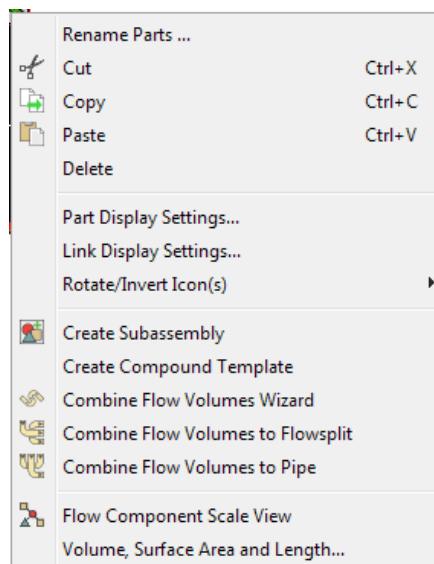
9.5 Create Simplified Parts in Full System Model

We will use the "Combine Flow Volumes" wizard to simplify each flow branch in the model. Note that the wizard is designed to combine multiple flow volumes into a single pipe or flowsplit; however, we will need to create 2 flow volumes per branch, in order to include a 'PressureLossConn' between them which accounts for the branch pressure drop (accordingly, $\frac{1}{2}$ of the total branch volume, surface area, and length will be included in each volume). For this reason, we will need to modify the wizard results slightly.

To simplify the heater inlet flow branch, follow these steps:

1. Select all flow parts to be simplified (including and 'OrificeConn' and other connecting parts between them), right-click, and select "Combine Flow Volumes." This will open the wizard dialog:





NOTE: If the flow branch being simplified consists of only a single part, a neighboring orifice must also be selected in order to make the wizard available. If no neighboring orifice exists, the branch may be unlinked, and a "dummy" orifice may be created and linked temporarily, simply to allow the wizard to run.

2. In the first screen of the wizard (*Template and Port Selection*), select "Combine flow volumes to **Flowsplit**" and click Next. In the next screen (*Port Assignments and Directions*), accept the default port positions, and click Next.
3. In the final screen (*Object Attribute Definition and Object Naming*), a list of objects contained in the flow branch is shown; one of them must be selected to supply non-geometric attribute values, such as fluid initial state, surface finish, thermal options, etc, to the resulting flowsplit.

NOTE: The thermal options are typically of greatest interest, as a flow branch containing many parts may use more than one different wall temperature solution method and/or 'WallTempSolver' object (for example, in this tutorial model the pump outlet flow branch consists of 2 parts using different 'WallTempSolver' objects). In this case, a warning will be given after completion of the wizard. A basic strategy here (used in this tutorial) is to choose the largest of the flow parts (i.e. that with the most surface area), to account for the greatest percentage of heat transfer to the branch. Alternatively, 2 different 'WallTempSolvers' may be used in the 2 resulting flowsplits, after the outlet flowsplit is completed.

4. To continue this conversion, select the first part in the list of parts presented ("Heater_In_1" in this case); this will present an object editor. In the "Main" folder of the object editor, edit the "Volume" and "Surface Area" attributes by using an equation to divide the existing values by 2:



Tutorial 9: Cooling System FRM Creation

	Attribute	Unit	Object Value
Basic Geometry and Initial Conditions			
Volume	mm ³		=274542.1/2 ...
Surface Area	mm ²		=73211.21/2 ...
Initial State Name			Coolant_Initial ...
Surface Finish			
<input checked="" type="radio"/> Smooth			
<input type="radio"/> Roughness from Material			smooth_rubber ...
<input type="radio"/> Sand Roughness	mm		
Additional Geometry Options			
Number of Identical Flowsplits			def (=1.0) ...

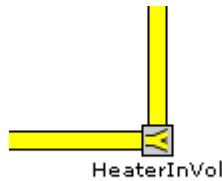
5. In the "Pressure Drop" folder, select the "No Friction Pressure Losses" option.
6. In the "Boundary Data" folder, edit the "Characteristic Length" attribute by using an equation to divide the existing value by 2, and ensure that the "Expansion Diameter" attribute for Boundary #2 is the same as #1 (this value should be the same as the [Exp-diam-inlet] flow bench parameter):

	Attribute	Unit	Boundary #1	Boundary #2
Boundary Data				
Link ID Number			1 ...	2 ...
<input checked="" type="radio"/> Angle (Planar Configuration)			180 ...	0 ...
<input type="radio"/> Angle wrt X-axis (3D)				
<input type="radio"/> Angle wrt Y-axis (3D)				
<input type="radio"/> Angle wrt Z-axis (3D)				
Characteristic Length	mm		=1553.59/2 ...	=1553.59/2 ...
Expansion Diameter	mm		15 ...	15 ...

NOTE: The expansion diameters for boundaries #1 and #2 must be the same (and equal to the inlet diameter of the full branch), even if the inlet and outlet diameters of the full flow branch are different. The outlet diameter of the branch will be duplicated in a 2nd flowsplit.

7. Give the object a name to reflect the flow branch ("HeaterInVol" for example), and click Finish. The flow parts have now been combined into a single 'FlowSplitGeneral' representing the "inlet" volume:

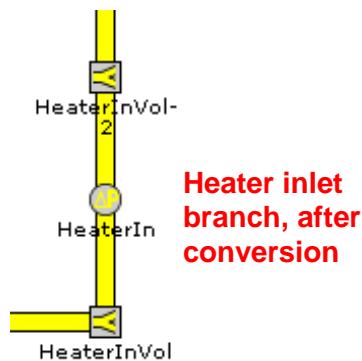




8. In order to complete the branch simplification, we must create a 2nd flowsplit for the "outlet" volume and a 'PressureLossConn' to place between them. In this example, because the inlet and outlet diameters of the original branch are the same, the "HeaterInVol" flowsplit part may simply be copied and pasted on the map, for use as the outlet flowsplit.

NOTE: If the inlet and outlet diameters of a flow branch are different, a 2nd flowsplit object must be created (most easily done with the "Copy and Edit Object..." command). For the outlet volume flowsplit, the expansion diameters for boundaries #1 and #2 must be the same, and equal to the outlet diameter of the full branch (and also equal to the [Exp-diam-outlet] flow bench parameter).

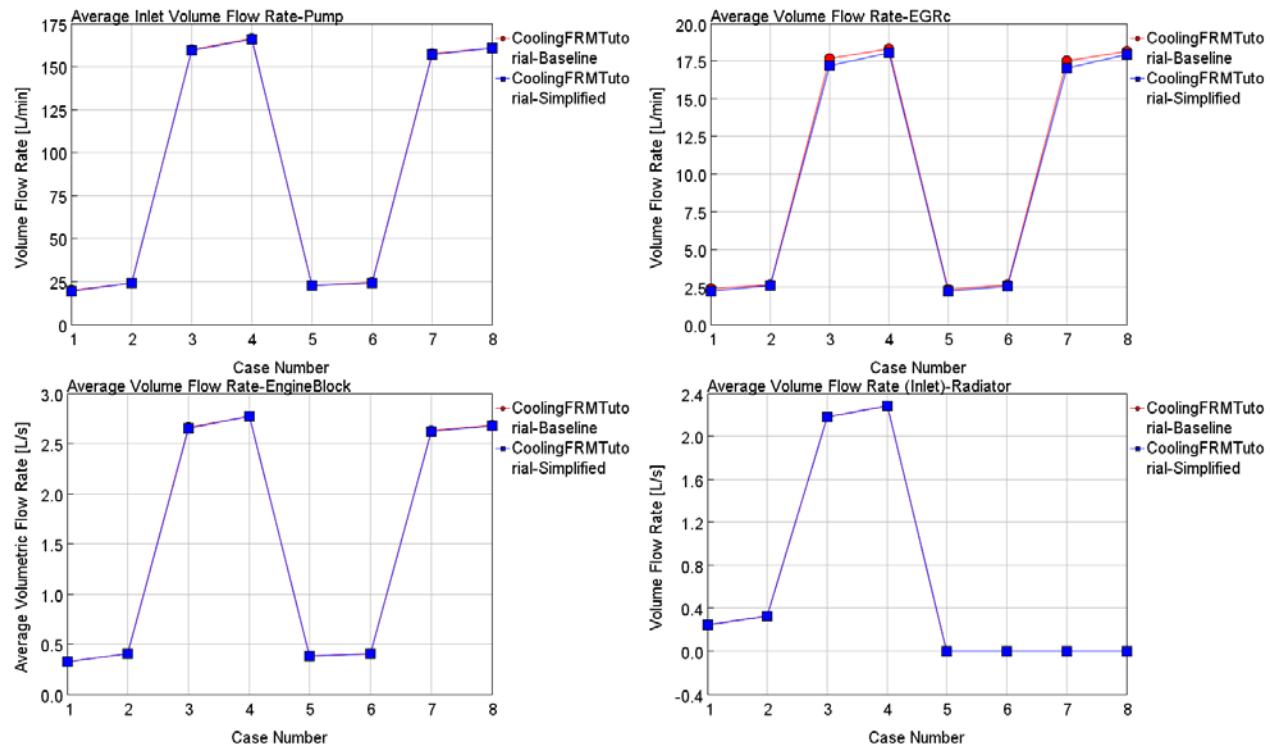
9. Drag a new 'PressureLossConn' object on to the map, and give it a name to reflect the branch. For the "Pressure Drop Reference Object" attribute, select the 'FlowPDropTableRef' object that was created in the previous step. Link the new parts to complete the conversion:



9.5.1 Run Model for Comparison to Full System

After simplifying a flow branch, it is recommended to save the full system model under a new name, and run all steady cases for comparison to the original model. For example, comparing flow rates through all of the major system components should show results that are nearly identical:



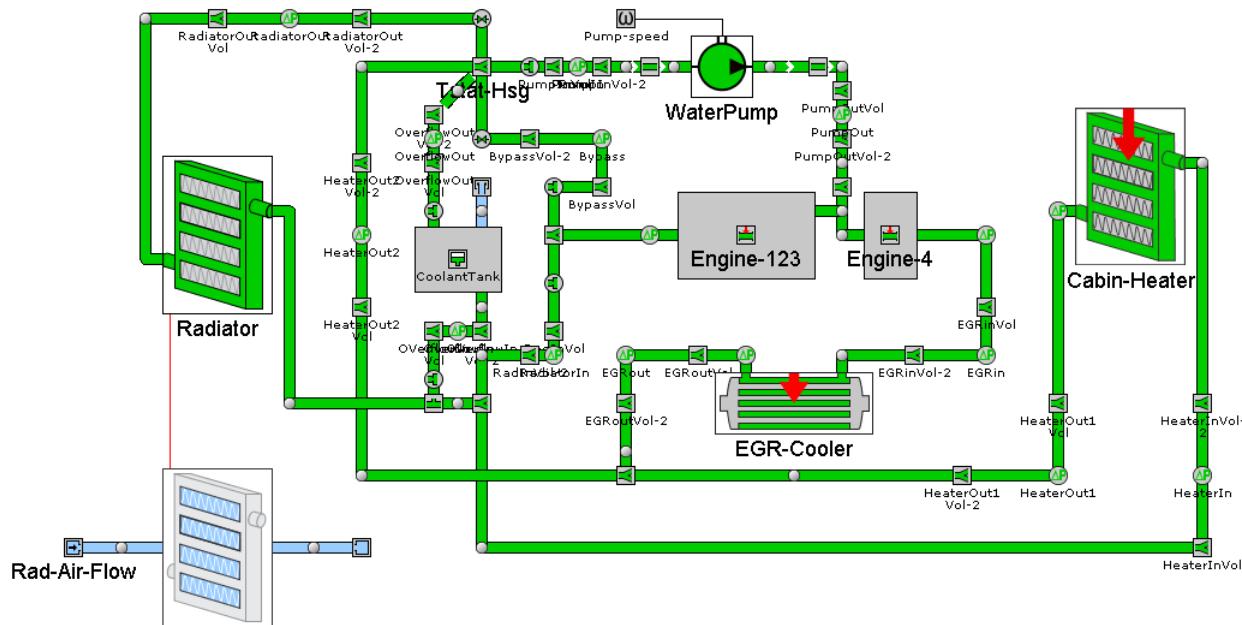


Doing this incremental comparison with each modified branch will make it easy to see if a change to a particular branch changed results, or if an error was made in the conversion process. The "before & after" steady-state results comparison of this tutorial can be viewed in the included report file *FRM_Comparisons.gu* in the tutorial directory.

9.5.2 Simplify the remaining Flow Branches

After all flow branches are converted and the model is run, the "Flow Solution Numerics" table in GT-POST shows that the FRM model contains 48 total flow volumes in the "Coolant" circuit, reduced from 297 volumes in the full system model.





Simplified FRM model, after changes (48 coolant circuit volumes)

9.6 Run FRM Model and Compare Results

To demonstrate the reduced run time of this FRM, a transient simulation (loosely based on the NEDC test) was performed using the full model, and compared to the simplified FRM. The following changes were made to convert the models to run transient simulations:

1. Engine speed was made transient.
2. Engine load was made transient.
3. Radiator air flow was made transient.
4. Engine heat input rate was made non-zero, and dependent upon engine speed and load.
5. Pump speed was made a function of engine speed, and a gear ratio added.
6. Thermostat lift dependent on temperature, rather than imposed.
7. Heat transfer turned ON in 'FlowControlImplicit' solution control object in Run Setup
8. Automatic shut-off turned OFF in Run Setup
9. TimeRLT results turned ON in Output Setup

The transient models are also included in the tutorial directory, and the "before & after" transient results comparison can be viewed in the included report file *FRM_Comparisons.gu* in the tutorial directory. Note that the converted FRM runs 2-3x faster than the baseline model.

