

# HEAT TRANSFER MODULE

USER'S GUIDE

**VERSION 3.5**

**How to contact COMSOL:**

**Benelux**

COMSOL BV  
Röntgenlaan 19  
2719 DX Zoetermeer  
The Netherlands  
Phone: +31 (0) 79 363 4230  
Fax: +31 (0) 79 361 4212  
info@comsol.nl  
www.comsol.nl

**Denmark**

COMSOL A/S  
Diplomvej 376  
2800 Kgs. Lyngby  
Phone: +45 88 70 82 00  
Fax: +45 88 70 80 90  
info@comsol.dk  
www.comsol.dk

**Finland**

COMSOL OY  
Arabianranta 6  
FIN-00560 Helsinki  
Phone: +358 9 2510 400  
Fax: +358 9 2510 4010  
info@comsol.fi  
www.comsol.fi

**France**

COMSOL France  
WTC, 5 pl. Robert Schuman  
F-38000 Grenoble  
Phone: +33 (0)4 76 46 49 01  
Fax: +33 (0)4 76 46 07 42  
info@comsol.fr  
www.comsol.fr

**Germany**

COMSOL Multiphysics GmbH  
Berliner Str. 4  
D-37073 Göttingen  
Phone: +49-551-99721-0  
Fax: +49-551-99721-29  
info@comsol.de  
www.comsol.de

**Italy**

COMSOL S.r.l.  
Via Vittorio Emanuele II, 22  
25122 Brescia  
Phone: +39-030-3793800  
Fax: +39-030-3793899  
info.it@comsol.com  
www.it.comsol.com

**Norway**

COMSOL AS  
Søndre gate 7  
NO-7485 Trondheim  
Phone: +47 73 84 24 00  
Fax: +47 73 84 24 01  
info@comsol.no  
www.comsol.no

**Sweden**

COMSOL AB  
Tegnérsgatan 23  
SE-111 40 Stockholm  
Phone: +46 8 412 95 00  
Fax: +46 8 412 95 10  
info@comsol.se  
www.comsol.se

**Switzerland**

FEMLAB GmbH  
Technoparkstrasse 1  
CH-8005 Zürich  
Phone: +41 (0)44 445 2140  
Fax: +41 (0)44 445 2141  
info@femlab.ch  
www.femlab.ch

**United Kingdom**

COMSOL Ltd.  
UH Innovation Centre  
College Lane  
Hatfield  
Hertfordshire AL10 9AB  
Phone: +44-(0)-1707 636020  
Fax: +44-(0)-1707 284746  
info.uk@comsol.com  
www.uk.comsol.com

**United States**

COMSOL, Inc.  
1 New England Executive Park  
Suite 350  
Burlington, MA 01803  
Phone: +1-781-273-3322  
Fax: +1-781-273-6603

COMSOL, Inc.  
10850 Wilshire Boulevard  
Suite 800  
Los Angeles, CA 90024  
Phone: +1-310-441-4800  
Fax: +1-310-441-0868

COMSOL, Inc.  
744 Cowper Street  
Palo Alto, CA 94301  
Phone: +1-650-324-9935  
Fax: +1-650-324-9936

info@comsol.com  
www.comsol.com

For a complete list of international  
representatives, visit  
www.comsol.com/contact

**Company home page**  
www.comsol.com

**COMSOL user forums**  
www.comsol.com/support/forums

*Heat Transfer Module User's Guide*

© COPYRIGHT 1994–2008 by COMSOL AB. All rights reserved

Patent pending

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from COMSOL AB.

COMSOL, COMSOL Multiphysics, COMSOL Script, COMSOL Reaction Engineering Lab, and FEMLAB are registered trademarks of COMSOL AB.

Other product or brand names are trademarks or registered trademarks of their respective holders.

Version: September 2008 COMSOL 3.5

Part number: CM020801

# C O N T E N T S

## Chapter 1: Introduction

<b>Introduction</b>	<b>2</b>
New Features in Heat Transfer Module 3.5 . . . . .	3
Typographical Conventions . . . . .	4

## Chapter 2: Theory of Heat Transfer

<b>Theory Background</b>	<b>6</b>
What is Heat Transfer? . . . . .	6
The Heat Equation . . . . .	6
Boundary Conditions . . . . .	9
Radiative Heat Transfer . . . . .	10
Convective Cooling/Heating and Heat Transfer Coefficients . . . . .	15
References . . . . .	18

## Chapter 3: Application Mode Overview

<b>Application Modes and Predefined Multiphysics Couplings</b>	<b>20</b>
--	-----------

## Chapter 4: General Heat Transfer

<b>The General Heat Transfer Application Mode</b>	<b>24</b>
Variables and Space Dimensions . . . . .	25
Application Mode Properties . . . . .	26
Scalar Variables . . . . .	28
<b>Convection and Conduction</b>	<b>29</b>
PDE Formulation . . . . .	29

Subdomain Settings . . . . .	30
Loading Material Properties from the Library . . . . .	42
The Material Libraries . . . . .	44
The Liquids and Gases Library and Material Property Functions . . . . .	44
Using Material Property Functions . . . . .	45
Editing Material Properties . . . . .	50
Turbulence Modeling . . . . .	51
References . . . . .	53
<b>Boundary Conditions</b>	<b>54</b>
Overview and Equations . . . . .	54
Boundary Settings . . . . .	56
Using h Coefficients in the Heat Transfer Module . . . . .	61
Heat Transfer Coefficients Library Expressions . . . . .	65
References . . . . .	66
<b>Radiative Heat Transfer</b>	<b>67</b>
The Radiosity Method . . . . .	67
View Factor Evaluation . . . . .	68
Boundary Settings . . . . .	71
Grouping Boundaries . . . . .	74
Guidelines for Solving Surface-to-Surface Radiation Problems . . . . .	76
<b>Highly Conductive Layers</b>	<b>77</b>
PDE Formulation—Highly Conductive Layers . . . . .	78
Boundary Settings—Highly Conductive Layer . . . . .	79
Reference . . . . .	80
<b>Out-of-Plane Heat Transfer</b>	<b>81</b>
PDE Formulation—Out-of-Plane Heat Transfer . . . . .	81
Subdomain Settings—Out-of-Plane Heat Transfer . . . . .	83
<b>Building and Solving General Heat Transfer Models</b>	<b>86</b>
<b>Building and Solving a Conduction Model</b>	<b>87</b>
Model Definition—Conduction in a Cylinder . . . . .	87
Results. . . . .	88
Reference . . . . .	88

Modeling Using the Graphical User Interface . . . . .	89
<b>Building and Solving a Convection and Conduction Model</b>	<b>91</b>
Model Definition—Heat Exchanger . . . . .	91
Results and Discussion. . . . .	93
Modeling Using the Graphical User Interface . . . . .	97
Modeling Using the Graphical User Interface—Extended Model . . . . .	102
<b>Building and Solving a Natural Convection Model—Laminar Flow</b>	<b>109</b>
Model Definition—Natural Convection Cooling in a Thermos. . . . .	110
Approach 1—Loading a Heat Transfer Coefficient Function. . . . .	112
Approach 2—Modeling the External Flow . . . . .	113
Results and Discussion. . . . .	116
Modeling Using the Graphical User Interface . . . . .	119
<b>Building and Solving a Radiation Model</b>	<b>131</b>
Model Definition—Radiation in a Cavity . . . . .	131
Results. . . . .	132
Reference . . . . .	134
Modeling Using the Graphical User Interface . . . . .	135
Comparing Results with Theory . . . . .	139
<b>Building and Solving a Model with Highly Conductive Layers</b>	<b>142</b>
Model Definition—Copper Layer on Silica Glass . . . . .	142
Results. . . . .	143
Modeling Using the Graphical User Interface . . . . .	144
<b>Building and Solving an Out-of-Plane Heat Transfer Model</b>	<b>146</b>
Model Definition—Thin Plate . . . . .	146
Results. . . . .	147
Modeling Using the Graphical User Interface . . . . .	149

## Chapter 5: Thin Conductive Shell

<b>The Thin Conductive Shell Application Mode</b>	<b>154</b>
Properties . . . . .	155
Scalar Variables . . . . .	155
Material . . . . .	156
Out-of-Plane Heat Transfer . . . . .	160
Edge Conditions . . . . .	161
<b>Building and Solving a Thin Conductive Shell Model</b>	<b>163</b>
Model Definition—Shell Conduction . . . . .	163
Results . . . . .	164
Reference . . . . .	165
Modeling Using the Graphical User Interface . . . . .	166

## Chapter 6: The Bioheat Equation

<b>The Bioheat Equation Application Mode</b>	<b>172</b>
Variables and Space Dimensions . . . . .	172
PDE Formulation . . . . .	172
Subdomain Settings . . . . .	174
Boundary Condition Types . . . . .	176
Boundary Settings . . . . .	178
Example Models . . . . .	178
Reference . . . . .	178

## Chapter 7: Momentum Transport

<b>Introduction</b>	<b>180</b>
References . . . . .	181
<b>The Incompressible Navier-Stokes Equations</b>	<b>182</b>
Subdomain Equations . . . . .	182
Boundary Conditions . . . . .	183

Point Settings . . . . .	193
Numerical Stability—Stabilization Techniques . . . . .	193
Corner Smoothing . . . . .	197
The Boussinesq Approximation . . . . .	198
Solver Settings . . . . .	199
References . . . . .	201
<b>Swirl Flow</b>	<b>203</b>
The Swirl Velocity Property . . . . .	204
<b>Turbulent Flow</b>	<b>206</b>
Subdomain Equations . . . . .	206
The $k$ - $\epsilon$ Turbulence Model . . . . .	209
The $k$ - $\omega$ Turbulence Model . . . . .	210
Boundary Conditions . . . . .	211
Numerical Stability—Stabilization Techniques . . . . .	216
Realizability Constraints . . . . .	219
The Turbulence Model Property . . . . .	221
Solver Settings . . . . .	221
References . . . . .	222
<b>Weakly Compressible Navier-Stokes</b>	<b>223</b>
Stabilization Settings for Weakly Compressible Navier-Stokes . . . . .	225
Other Settings . . . . .	226
Turbulent Weakly Compressible Flow . . . . .	226
References . . . . .	227
<b>Non-Isothermal Flow</b>	<b>228</b>
Laminar Non-Isothermal Flow . . . . .	228
Turbulent Non-Isothermal Flow . . . . .	229
<b>Building and Solving a Non-Isothermal Laminar Flow Model</b>	<b>231</b>
Introduction . . . . .	231
Model Definition—Heated Plate . . . . .	231
Results and Discussion. . . . .	233
Comparing Results with Empirical Heat Transfer Coefficients . . . . .	236
Reference . . . . .	240
Modeling Using the Graphical User Interface . . . . .	241

<b>Building and Solving a Non-Isothermal Turbulent Flow Model</b>	<b>248</b>
Introduction . . . . .	248
Model Definition . . . . .	248
Results and Discussion. . . . .	250
Reference . . . . .	253
Modeling Using the Graphical User Interface . . . . .	253

## Chapter 8: Application Mode Implementation

<b>Application Mode Overview</b>	<b>266</b>
<b>Application Mode Equations</b>	<b>268</b>
<b>Application Mode Variables</b>	<b>269</b>
General Heat Transfer . . . . .	271
Thin Conductive Shell . . . . .	276
The Bioheat Equation . . . . .	278
Momentum Transport . . . . .	280

## Chapter 9: Glossary

<b>Glossary of Terms</b>	<b>288</b>
<b>INDEX</b>	<b>291</b>



# Introduction

This chapter provides an introduction to the Heat Transfer Module.

# Introduction

The Heat Transfer Module is an optional package that extends the COMSOL Multiphysics® modeling environment with customized user interfaces and functionality optimized for the analysis of heat transfer. We have developed it for a wide audience including researchers, developers, teachers, and students. To assist users at all levels of expertise, this module comes with a library of prewritten ready-to-run application modes and example models that appear in the companion *Heat Transfer Module Model Library*.

Heat transfer is involved in almost every kind of physical process, and in fact it can be the limiting factor for many processes. Thus its study is of vital importance, and the requirement for powerful heat transfer analysis tools is virtually universal. Further, heat transfer often appears together with, or as a result of, other physical phenomena, making this module useful for engineers in all disciplines.

Indeed, the modeling of heat transfer effects has become ever more important in the design of products in many areas including the electronics, automotive, and medical industries. The combination of experimental work in the laboratory along with theoretical analyses through computer models has proven to be effective in accelerating the understanding of problems as well as helping decrease development costs for new processes.

We developed this dedicated Heat Transfer Module to greatly expand upon the base capabilities available in COMSOL Multiphysics, in particular by adding support for radiative heat transfer. The module supports all fundamental mechanisms including *conductive*, *convective*, and *radiative* heat transfer (both *surface-to-surface* and *surface-to-ambient* radiation). Using the application modes in this module along with the inherent multiphysics capabilities of COMSOL Multiphysics, you can model a temperature field in parallel with other physics—a powerful combination that makes your models even more accurate and representative of the real world.

The documentation set for the Heat Transfer Module consists of two books. The one in your hands, the *Heat Transfer Module User's Guide*, introduces the basic modeling process. The tutorial sections present the different application modes available in the module. It also discusses the modeling strategy for various cases. This section covers different combinations of conductive, convective, and radiative heat transfer. It also reviews special modeling techniques for highly conductive layers, thin conductive shells, and out-of-plane heat transfer. Throughout the manual the models

increase in complexity by combining several heat transfer mechanisms and also by coupling these to application modes describing the flow of fluids.

The second book in the set, the *Heat Transfer Module Model Library*, is divided into three broadly defined application areas where heat transfer plays an important role:

- Electronics and power systems
- Processing and manufacturing
- Medical technology

Most of the models involve multiple heat transfer mechanisms and are often coupled to other physical phenomena, for example, fluid dynamics or electromagnetics. The authors developed several state-of-the-art examples by reproducing models that have appeared in international scientific journals.

The library also contains a set of tutorial models that should prove very helpful for new users who want to step through simple examples that illustrate how to simulate different heat-transfer mechanisms. Each model comes with some theoretical background as well as step-by-step instructions that illustrate how to set it up. Further, we supply these examples as COMSOL Multiphysics Model MPH-files so you can open them in COMSOL Multiphysics for immediate access. This way you can follow along with the printed discussion as well as use them as a jumping-off point for your own modeling needs.

We hope the Heat Transfer Module becomes a valuable tool in your modeling work, and we are convinced that the effort you put into understanding COMSOL Multiphysics will be repaid several times over. If you have any feedback on the models in this set, please let us know. Likewise, if you have any ideas for additional models that we could add to the library, we welcome your suggestions. In any case, feel free to contact us at [info@comsol.com](mailto:info@comsol.com).

### *New Features in Heat Transfer Module 3.5*

---

- Support for heat transfer in unbounded domains using infinite elements.
- Improved stabilization for nonisothermal flow and convective heat transfer
- Mass source term for the Brinkman equation
- New models:
  - Model of a light bulb using surface-to-surface radiation and free convection
  - Model of a displacement ventilation system in a room using turbulent flow coupled to heat transfer

## *Typographical Conventions*

---

All COMSOL manuals use a set of consistent typographical conventions that should make it easy for you to follow the discussion, realize what you can expect to see on the screen, and know which data you must enter into various data-entry fields. In particular, you should be aware of these conventions:

- A **boldface** font of the shown size and style indicates that the given word(s) appear exactly that way on the COMSOL graphical user interface (for toolbar buttons in the corresponding tooltip). For instance, we often refer to the **Model Navigator**, which is the window that appears when you start a new modeling session in COMSOL; the corresponding window on the screen has the title **Model Navigator**. As another example, the instructions might say to click the **Multiphysics** button, and the boldface font indicates that you can expect to see a button with that exact label on the COMSOL user interface.
- The names of other items on the graphical user interface that do not have direct labels contain a leading uppercase letter. For instance, we often refer to the Draw toolbar; this vertical bar containing many icons appears on the left side of the user interface during geometry modeling. However, nowhere on the screen will you see the term “Draw” referring to this toolbar (if it were on the screen, we would print it in this manual as the **Draw** menu).
- The symbol **>** indicates a menu item or an item in a folder in the **Model Navigator**. For example, **Physics>Equation System>Subdomain Settings** is equivalent to: On the **Physics** menu, point to **Equation System** and then click **Subdomain Settings**. **COMSOL Multiphysics>Heat Transfer>Conduction** means: Open the **COMSOL Multiphysics** folder, open the **Heat Transfer** folder, and select **Conduction**.
- A Code (monospace) font indicates keyboard entries in the user interface. You might see an instruction such as “Type 1.25 in the **Current density** edit field.” The monospace font also indicates COMSOL Script codes.
- An *italic* font indicates the introduction of important terminology. Expect to find an explanation in the same paragraph or in the Glossary. The names of books in the COMSOL documentation set also appear using an italic font.

# Theory of Heat Transfer

In this chapter you find a background to the theory of heat transfer.

# Theory Background

This chapter reviews the theory behind the equations with which you model heat transfer problems. It describes in detail the theory behind radiative heat transfer between surfaces, that is, surface-to-surface radiation, because this functionality is unique to this module. For more detailed discussions of the fundamentals of heat transfer, see Ref. 1 and Ref. 2.

## *What is Heat Transfer?*

---

Heat transfer is defined as the movement of energy due to a difference in temperature. It is characterized by the following three mechanisms, all of which the Heat Transfer Module supports:

- *Conduction*—Heat conduction takes place through different mechanisms in different media. Theoretically it takes place in a gas through collisions of the molecules; in a fluid through oscillations of each molecule in a “cage” formed by its nearest neighbors; in metals mainly by electrons carrying heat and in other solids by molecular motion which in crystals take the form of lattice vibrations known as phonons. Typical for heat conduction is that the heat flux is proportional to the temperature gradient.
- *Convection*—Heat convection takes place through the net displacement of a fluid, which transports the heat content in a fluid through the fluid’s own velocity. The term convection is also used for the heat dissipation from a solid surface to a fluid, where the heat transfer coefficient and the temperature difference across a fictitious film describe the flux. The Heat Transfer Module includes both descriptions.
- *Radiation*—Heat transfer by radiation takes place through the transport of photons, which solid surfaces can absorb or reflect. The Heat Transfer Module includes surface-to-surface radiation, which accounts for shadowing and diffuse reflections between radiating surfaces. It also includes surface-to-ambient radiation where the ambient surroundings are treated as a black body with known temperature.

## *The Heat Equation*

---

The fundamental law governing all heat transfer is the first law of thermodynamics, commonly referred to as the principle of conservation of energy. However, internal

energy,  $U$ , is a rather inconvenient quantity to measure and use in simulations. Therefore, the basic law is usually rewritten in terms of temperature,  $T$ . For a fluid, the resulting *heat equation* is:

$$\rho C_p \left( \frac{\partial T}{\partial t} + (\mathbf{u} \cdot \nabla) T \right) = -(\nabla \cdot \mathbf{q}) + \tau : \mathbf{S} - \left. \frac{T \partial \rho}{\rho \partial T} \right|_p \left( \frac{\partial p}{\partial t} + (\mathbf{u} \cdot \nabla) p \right) + Q \quad (2-1)$$

where

- $\rho$  is the density (kg/m<sup>3</sup>)
- $C_p$  is the specific heat capacity at constant pressure (J/(kg·K))
- $T$  is absolute temperature (K)
- $\mathbf{u}$  is the velocity vector (m/s)
- $\mathbf{q}$  is the heat flux by conduction (W/m<sup>2</sup>)
- $p$  is pressure (Pa)
- $\tau$  is the viscous stress tensor (Pa)
- $\mathbf{S}$  is the strain rate tensor (1/s):

$$\mathbf{S} = \frac{1}{2}(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)$$

- $Q$  contains heat sources other than viscous heating (W/m<sup>3</sup>)

In deriving Equation 2-1, a number of thermodynamic relations have been used. The equation also assumes that mass is always conserved, which means that density and velocity must be related through:

$$\frac{\partial \rho}{\partial t} + \nabla \bullet (\rho \mathbf{v}) = 0$$

The General Heat Transfer application mode uses Fourier's law of conduction which states that the conductive heat flux,  $\mathbf{q}$ , is proportional to the temperature gradient:

$$q_i = -k \frac{\partial T}{\partial x_i} \quad (2-2)$$

where  $k$  is the thermal conductivity (W/(m·K)). In a solid, the thermal conductivity can be different in different directions. Then  $k$  becomes a tensor

$$k = \begin{bmatrix} k_{xx} & k_{xy} & k_{xz} \\ k_{yx} & k_{yy} & k_{yz} \\ k_{zx} & k_{zy} & k_{zz} \end{bmatrix}$$

and the conductive heat flux is given by

$$q_i = -\sum_j k_{ij} \frac{\partial T}{\partial x_j}$$

The second term on the right of Equation 2-1 represents viscous heating of a fluid. An analogous term arises from the internal viscous damping of a solid. The operation “:” is a contraction and can in this case be written on the following form:

$$\mathbf{a}:\mathbf{b} = \sum_n \sum_m a_{nm} b_{nm} \quad (2-3)$$

The third term represents pressure work and is responsible for the heating of a fluid under adiabatic compression and for some thermo-acoustic effects. It is generally small for low Mach number flows. A similar term can be included to account for thermo-elastic effects in solids.

Inserting Equation 2-2 into Equation 2-1, reordering the terms and ignoring viscous heating and pressure work puts the heat equation on a perhaps more familiar form:

$$\rho C_p \frac{\partial T}{\partial t} + \nabla \cdot (-k \nabla T) = Q - \rho C_p \mathbf{u} \cdot \nabla T \quad (2-4)$$

The General Heat Transfer application mode solves this equation for the temperature,  $T$ . When convective heat transfer is active, you can provide the velocity  $\mathbf{u}$  as a mathematical expression of the independent variables or calculate it within COMSOL Multiphysics by a coupling to a momentum-transfer application mode such as Incompressible Navier-Stokes or Weakly Compressible Flow application mode. If the velocity is set to zero, you finally get the equation governing pure conductive heat transfer in a solid:

$$\rho C_p \frac{\partial T}{\partial t} + \nabla \cdot (-k \nabla T) = Q$$



## Boundary Conditions

---

The heat equation accepts two basic types of boundary conditions: *specified temperature* and *specified heat flux*. The former is of Dirichlet type and prescribes the temperature at a boundary:

$$T = T_0 \quad \text{on } \partial\Omega$$

while the latter specifies the inward heat flux

$$-\mathbf{n} \cdot \mathbf{q} = q_0 \quad \text{on } \partial\Omega$$

where:

- $\mathbf{q}$  is the total heat flux vector ( $\text{W}/\text{m}^2$ ),

$$\mathbf{q} = -k\nabla T + \rho C_p \mathbf{u}T$$

- $\mathbf{n}$  is the normal vector of the boundary,
- $q_0$  is inward heat flux ( $\text{W}/\text{m}^2$ ), normal to the boundary.

Note that when convective heat transfer is active, the heat flux boundary condition is a mixed, or Robin type, boundary condition rather than a pure Neumann boundary condition.

The special case  $q_0 = 0$  is called *thermal insulation*. Another special case is  $q_0 = -\rho C_p \mathbf{u}T$ , or equivalently  $-\mathbf{n} \cdot (-k\nabla T) = 0$ , which is known as *convective flux*. This is usually the appropriate condition on an outflow boundary in a model with convection. If the velocities are zero, thermal insulation and convective flux are equivalent conditions.

The inward heat flux  $q_0$  is normally a sum of contributions from different heat transfer processes. It is often convenient to split the heat flux boundary condition as

$$-\mathbf{n} \cdot \mathbf{q} = q_0 + q_r + q_s + h(T_{\text{inf}} - T) \quad \text{on } \partial\Omega \quad (2-5)$$

where  $q_r$  represents incoming radiation and  $q_s$  is a contribution from a thin but highly conducting shell in contact with the boundary. The last term is a product of a heat transfer coefficient,  $h$ , and the difference between the surface temperature  $T$  and a reference temperature  $T_{\text{inf}}$ . It can be used to model a thin shell with low thermal conductivity or, more commonly, the convective cooling of a surface exposed to a flowing fluid with bulk temperature  $T_{\text{inf}}$ .

The next two sections take a closer look at the contributions from radiation and from convective heat transfer across a boundary layer.

### *Radiative Heat Transfer*

This discussion has thus far considered heat transfer by means of conduction and convection. The third mechanism for heat transfer is radiation. Thermal radiation denotes the stream of electromagnetic waves emitted from a body at a certain temperature. This section studies the theory behind the radiative heat transfer process that occurs on the surface of a body.

#### **DERIVING THE RADIATIVE HEAT FLUX**

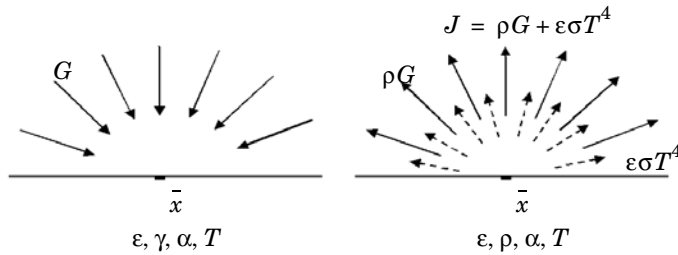


Figure 2-1: Arriving irradiation (left), leaving radiosity (right).

Consider a point  $\bar{x}$  located on an opaque surface that has an emissivity  $\varepsilon$ , reflectivity  $\rho$ , absorptivity  $\alpha$ , and temperature  $T$ . Assume the body is opaque, which means that no radiation is transmitted through the body. This is true for most solid bodies.

The total *arriving* radiative flux at  $\bar{x}$  is named the irradiation,  $G$ . The total *outgoing* radiative flux  $\bar{x}$  is named the radiosity,  $J$ . The radiosity is the sum of the reflected radiation and the emitted radiation:

$$J = \rho G + \varepsilon \sigma T^4 \quad (2-6)$$

The *net inward* radiative heat flux,  $q$ , is then given the difference between the irradiation and the radiosity:

$$q = G - J \quad (2-7)$$

Using Equation 2-6 and Equation 2-7 you can eliminate  $J$  and obtain a general expression for the net inward heat flux into the opaque body based on  $G$  and  $T$ .

$$q = (1 - \rho)G - \varepsilon \sigma T^4 \quad (2-8)$$

Most opaque bodies also behave as ideal gray bodies, meaning that the absorptivity and emissivity are equal, and the reflectivity is therefore given from the following relation:

$$\alpha = \varepsilon = 1 - \rho \quad (2-9)$$

Thus, for ideal gray bodies,  $q$  is given by:

$$q = \varepsilon(G - \sigma T^4) \quad (2-10)$$

This is the equation used in COMSOL Multiphysics in the General Heat Transfer application as radiation boundary condition.

### **RADIATION TYPES**

The Heat Transfer Module distinguishes between two types of radiative heat transfer: *surface-to-ambient* radiation, and *surface-to-surface* radiation. Equation 2-10 holds for both radiation types, but the irradiation term,  $G$ , is different for them. The next sections derive the irradiation term and the resulting radiative heat flux for both radiation types.

### **SURFACE-TO-AMBIENT RADIATION**

Surface-to-ambient radiation assumes the following:

- The ambient surroundings in view of the surface have a constant temperature,  $T_{\text{amb}}$ .
- The ambient surroundings behave as a black body. This means that the emissivity and absorptivity are equal to 1, and zero reflectivity.

These assumptions allow you to explicitly express the irradiation as

$$G = \sigma T_{\text{amb}}^4 \quad (2-11)$$

Inserting Equation 2-11 into Equation 2-10 results in the net inward heat flux for surface-to-ambient radiation

$$q = \varepsilon\sigma(T_{\text{amb}}^4 - T^4) \quad (2-12)$$

For boundaries where you have specified surface-to-ambient radiation, COMSOL Multiphysics adds this term to the right side of Equation 2-5.

### SURFACE-TO-SURFACE RADIATION

Surface-to-surface radiation is more complex. It includes radiation from both the ambient surroundings and from other surfaces. A generalized equation for the irradiative flux is:

$$G = G_m + F_{\text{amb}} \sigma T_{\text{amb}}^4 \quad (2-13)$$

where  $G_m$  is the *mutual irradiation* arriving from other surfaces in the modeled geometry and  $F_{\text{amb}}$  is the ambient view factor. The latter describes the portion of the view from each point that is covered by ambient conditions.  $G_m$  on the other hand is determined from the geometry and the local temperatures of the surrounding boundaries. The following sections derive the equations for  $G_m$  and  $F_{\text{amb}}$  for a general 3D case:

Consider a point  $\bar{x}$  on a surface as in Figure 2-2. Point  $\bar{x}$  can see points on other surfaces as well as the ambient surrounding. Assume that the points on the other surfaces have a local radiosity,  $J'$ , while the ambient surrounding has a constant temperature,  $T_{\text{amb}}$ .

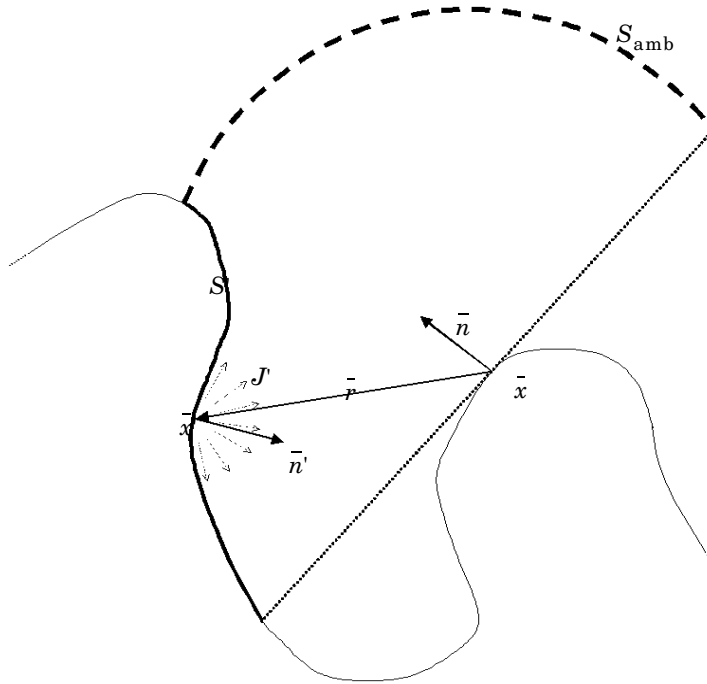


Figure 2-2: Example geometry for surface-to-surface radiation.

The mutual irradiation at point  $\bar{x}$  is given by the following surface integral:

$$G_m = \int_{S'} \frac{(-\bar{n}' \cdot \bar{r})(\bar{n} \cdot \bar{r})}{\pi|\bar{r}|^4} J' dS \quad (2-14)$$

The heat flux that arrives from  $\bar{x}'$  depends on the local radiosity  $J'$  projected onto  $\bar{x}$ . The projection is computed using the normal vectors  $\bar{n}$  and  $\bar{n}'$  along with the vector  $\bar{r}$ , which points from  $\bar{x}$  to  $\bar{x}'$ .

The ambient view factor,  $F_{\text{amb}}$ , is determined from the integral of the surrounding surfaces  $S'$ , here denoted as  $F'$ , determined from the integral below:

$$F_{\text{amb}} = 1 - F' = 1 - \int_{S'} \frac{(-\bar{n}' \cdot \bar{r})(\bar{n} \cdot \bar{r})}{\pi|\bar{r}|^4} dS \quad (2-15)$$

The two last equations plug into Equation 2-13 to yield the final equation for irradiative flux.

The equations used so far apply to the general 3D case. 2D geometries results in simpler integrals. For 2D the resulting equations for the mutual irradiation and ambient view factor are:

$$G_m = \int_{S_{\perp}'} \frac{(-\bar{n}' \cdot \bar{r}_{\perp})(\bar{n} \cdot \bar{r}_{\perp})}{2|\bar{r}_{\perp}|^3} J dS_{\perp} \quad (2-16)$$

$$F_{\text{amb}} = 1 - \int_{S_{\perp}'} \frac{(-\bar{n}' \cdot \bar{r}_{\perp})(\bar{n} \cdot \bar{r}_{\perp})}{2|\bar{r}_{\perp}|^3} dS_{\perp} \quad (2-17)$$

where the integral over  $S_{\perp}'$  denotes the line integral along the boundaries of the 2D geometry.

In axisymmetric geometries, the irradiation and ambient view factor cannot be computed directly from a closed-form expression. Instead, a virtual 3D geometry must be constructed, and the view factors evaluated according to Equation 2-16.

### SOLVING FOR THE RADIOSITY

The previous section derived equations for the irradiation  $G$  at an arbitrary surface point  $\bar{x}$ . Now recall the expression for the radiosity leaving from  $\bar{x}$

$$J = \rho G + \varepsilon \sigma T^4 \quad (2-18)$$

Inserting the expression for  $G$  in Equation 2-18 gives the following equation for the radiosity:

$$J = \rho \{ G_m + F_{\text{amb}} \sigma T_{\text{amb}}^4 \} + \varepsilon \sigma T^4 \quad (2-19)$$

Assuming an ideal gray body, the last equation becomes:

$$J = (1 - \varepsilon) \{ G_m + F_{\text{amb}} \sigma T_{\text{amb}}^4 \} + \varepsilon \sigma T^4 \quad (2-20)$$

This is the equation used in the General Heat Transfer application mode to solve for the radiosity,  $J$ . It applies to boundaries that participate in surface-to-surface radiation. Equation 2-20 results in a linear equation system in  $J$  that is solved in parallel with the equation for the temperature,  $T$ .

## *Convective Cooling/Heating and Heat Transfer Coefficients*

---

One of the most common boundary conditions when modeling heat transfer is convective cooling or heating whereby a fluid cools a surface by natural or forced convection. In principle, it is possible to model this process in two ways:

- Use a heat transfer coefficient on the convection-cooled surfaces
- Extend the model to describe the flow and heat transfer in the cooling fluid

The first method is very powerful and efficient. In addition, the General Heat Transfer application mode provides built-in functions for using heat transfer coefficients. For most engineering purposes, the use of these coefficients is an accurate and numerically efficient modeling approach. You then model convection cooling by specifying the heat flux on the boundaries that interface with the cooling fluid as being proportional to the temperature difference across a fictitious thermal boundary layer. You describe the flux in terms of a heat transfer coefficient,  $h$ , according to the equation

$$-\mathbf{n} \cdot (-k\nabla T) = h(T_{\text{inf}} - T).$$

The second approach, which implies including the cooling fluid's flow and heat transfer in the model, can be useful in certain cases. You might want to try this method if, for example, the geometry causes the heat transfer coefficient to vary substantially with position, or if the temperature in the fluid is of particular interest. In such cases, the boundary condition at the interface is continuity and requires no attention.

However, the model's complexity increases significantly due to the necessity of solving the flow field as well as the temperature field. In addition, memory requirements and calculation time generally grow significantly. This section focuses on the method of using heat transfer coefficients to describe convective cooling.

The main difficulty in using heat transfer coefficients is in calculating or specifying the appropriate value of the  $h$  coefficient. That coefficient depends on the cooling fluid, the fluid's material properties, and the surface temperature—and, for forced-convection cooling, also on the fluid's flow rate. In addition, the geometrical configuration affects the coefficient.

### **HEAT TRANSFER COEFFICIENT THEORY**

It is possible to divide convection cooling into four main categories depending on the type of convection conditions (natural or forced) and on the type of geometry (internal or external convection flow). In addition, these four cases can all experience either

laminar or turbulent flow conditions, resulting in a total of eight types of convection, as in Figure 2-3.

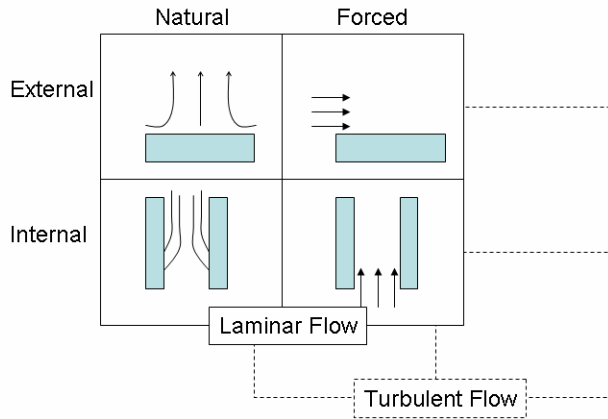


Figure 2-3: The eight possible categories of convection cooling.

The difference between natural and forced convection is that in the latter case an external force such as a fan creates the flow. In natural convection, buoyancy forces induced by temperature differences and the thermal expansion of the fluid drive the flow.

For each category, various relationships for the heat transfer coefficient have been proposed in the literature. For most cases, the equation describing the  $h$  coefficient varies significantly with the geometrical shape. The expression differs, for example, for forced laminar internal convection between a tube and a set of parallel plates.

Heat transfer handbooks generally contain a large set of empirical and theoretical correlations for  $h$  coefficients. The Heat Transfer Module's Heat Transfer Coefficients library includes a subset of them. The Heat Transfer Coefficients library uses handbook expressions based on the following set of dimensionless numbers:

- The Nusselt number,  $Nu_L(Re, Pr, Ra) = hL/k$
- The Reynolds number,  $Re_L = \rho UL/\eta$
- The Prandtl number,  $Pr = \eta C_p/k$
- The Rayleigh number,  $Ra = Gr Pr = \rho^2 g \beta C_p \Delta T L^3 / (\eta k)$

where

- $h$  is the heat transfer coefficient ( $W/(m^2 \cdot K)$ ).



- $L$  is the characteristic length (m).
- $\Delta T$  is the temperature difference between surface and cooling fluid bulk (K).
- $g$  is the gravitational constant ( $\text{m/s}^2$ ).
- $k$  is the thermal conductivity of the fluid ( $\text{W}/(\text{m}\cdot\text{K})$ ).
- $\rho$  is the fluid density ( $\text{kg}/\text{m}^3$ ).
- $U$  is the bulk velocity ( $\text{m/s}$ ).
- $\eta$  is the viscosity ( $\text{Pa}\cdot\text{s}$ ).
- $C_p$  equals the heat capacity of the fluid ( $\text{J}/(\text{kg}\cdot\text{K})$ ).
- $\beta$  is the thermal expansivity ( $1/\text{K}$ ).

Further,  $\text{Gr}$  refers to the Grashof number, which is defined as the ratio between the buoyancy force and the viscous force.

#### **NATURE OF THE FLOW—THE GRASHOF NUMBER**

In any fluid-flow modeling, before setting up the model it is necessary to know the nature of the flow, an aspect that influences the choice of mathematical model.

In cases of externally driven flow, such as forced convection, the flow's nature is characterized by the *Reynolds number*,  $\text{Re}$ , which describes the ratio of the inertial to viscous forces. It is based on the velocity, viscosity, density, and length scale.

However, the velocity is largely unknown for internally driven flows such as natural convection. In such cases the *Grashof number*,  $\text{Gr}$ , characterizes the flow. It describes the ratio of the internal driving force (buoyancy force) to a viscous force acting on the fluid. Similarly to the Reynolds number it requires the definition of a length scale, the fluid's physical properties, and the temperature scale (temperature difference). The Grashof number is defined as:

$$\text{Gr}_L = \frac{g\beta(T_s - T_0)L^3}{\left(\frac{\eta}{\rho}\right)^2}$$

where  $g$  is the constant of gravitational acceleration,  $\beta$  is the fluid's expansion coefficient,  $T_s$  denotes the temperature of the hot surface,  $T_0$  equals the temperature of the surrounding air,  $L$  is the length scale,  $\eta$  represents the fluid's dynamic viscosity, and  $\rho$  is the density.

For ideal gases, the expansion coefficient is given by

$$\beta = 1/T.$$

The transition from laminar to turbulent flow occurs at a Gr value of  $10^9$ ; the flow is turbulent for larger values.

### **NUSSELT NUMBER CORRELATIONS**

In the Heat Transfer Coefficients library included with the Heat Transfer Module, the  $h$  coefficient is based on Nusselt number correlations from handbooks and is expressed as a function of the material properties, temperature, flow rate, and geometry.

For natural convection, the relationship for the Nusselt number typically has the form

$$\text{Nu} = C'(\text{GrPr})^n .$$

The parameter  $C'$  depends on the geometry. The exponent  $n$  is typically equal to 0.25 for laminar flow and 0.33 for turbulent flow. Nusselt relationships for forced convection vary significantly and have no general formula.

There are two types of Nusselt numbers: average,  $\overline{\text{Nu}}_L$ ; and local,  $\text{Nu}_y$ . The average is an integral form. It is based on the total length of the cooled surface and yields an average heat transfer coefficient,  $h_{\text{ave}}$ . Local Nusselt numbers result in a local heat transfer coefficient depending on position. In this case, the variable  $L$  in the expressions is replaced by  $y$ , the distance from the leading edge (or distance from the first contact point along the flow direction). In the Heat Transfer Coefficients library both types are present.

### *References*

---

1. F.P. Incropera and D.P. DeWitt, *Fundamentals of Heat and Mass Transfer*, 4th ed., John Wiley & Sons, 1996.
2. A. Bejan, *Heat Transfer*, Wiley, 1993.

## Application Mode Overview

This short chapter provides an overview of the application modes and predefined multiphysics couplings in the Heat Transfer Module.

# Application Modes and Predefined Multiphysics Couplings

The Heat Transfer Module consists of a number of application modes that describe the temperature field of a non-isothermal system. These application modes treat problems that involve heat transfer by conduction, convection, and radiation.

In addition to the application modes that describe the temperature field, the Heat Transfer Module also includes an application mode for fluid dynamics that is capable of modeling the velocity field of a non-isothermal fluid. With this application mode you can describe the convective field of a heat transfer problem.

APPLICATION MODE	NAME	1D	2D	3D	STATIONARY	TIME-DEPENDENT
General Heat Transfer, Cartesian	htgh	√	√	√	√	√
General Heat Transfer, axisymmetry	htgh	√	√		√	√
Bioheat Equation, Cartesian	htbh		√	√	√	√
Bioheat Equation, axisymmetry	htbh		√		√	√
Thin Conductive Shell	htsh			√	√	√
Weakly Compressible Navier-Stokes, Cartesian	chns		√	√	√	√
Weakly Compressible Navier-Stokes, axisymmetry	chns		√		√	√

The Heat Transfer Module also includes the following predefined multiphysics couplings:

PREDEFINED MULTIPHYSICS COUPLING	APPLICATION MODES	STATIONARY	TIME-DEPENDENT
Electro-Thermal Interaction—Joule Heating	General Heat Transfer, Conductive Media DC	√	√
Fluid-Thermal Interaction—Non-Isothermal Flow	General Heat Transfer, Weakly Compressible Navier-Stokes	√	√
Fluid-Thermal Interaction—Turbulent Non-Isothermal Flow, k-ε	General Heat Transfer, k-ε Turbulence Model	√	√
Fluid-Thermal Interaction—Turbulent Non-Isothermal Flow, k-ω	General Heat Transfer, k-ω Turbulence Model	√	√

All predefined multiphysics couplings are available for 2D, 2D axisymmetric, and 3D geometries. For more information about the Fluid-Thermal Interaction predefined multiphysics couplings, see “Non-Isothermal Flow” on page 228. The Joule Heating predefined Multiphysics coupling is documented under “Electro-Thermal Interaction—Joule Heating” on page 380 of the *COMSOL Multiphysics Modeling Guide*.

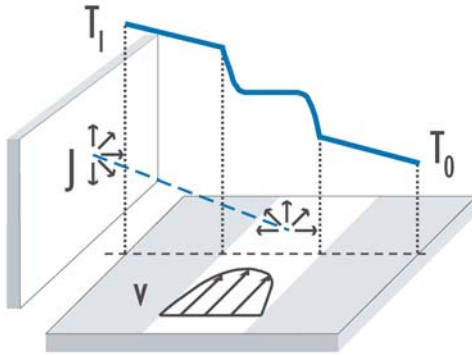
The next chapters describe the equations and boundary conditions for the application modes in the Heat Transfer Module. Each chapter also presents tutorial examples that illustrate how to work with the equations to build and solve models with a specific application mode.



# General Heat Transfer

In this chapter you find a description of the General Heat Transfer application mode, which is the most fundamental application mode in the Heat Transfer Module. It covers all three types of heat transfer: conduction, convection, and radiation.

# The General Heat Transfer Application Mode



General Heat Transfer is the main application mode in the Heat Transfer Module. It handles problems involving the three fundamental heat-transfer mechanisms: *conduction*, *convection*, and *radiation*. These mechanisms can be active in adjacent domains and exchange heat at a common boundary, or they can all interact in the same domain. The General Heat Transfer application mode also supports special modeling techniques for treating heat transfer in thin *highly conductive layers* and for keeping account of *out-of-plane* heat transfer in 2D and 1D models.

The following table shows in which spatial dimensions the various heat transfer mechanisms and modeling features can be put to work:

HEAT TRANSFER PROCESS	1D	AXI 1D	2D	AXI 2D	3D
Conduction (enabled by default)	√	√	√	√	√
Convection	√	√	√	√	√
Radiation (surface-to-ambient)	√	√	√	√	√
Radiation (surface-to-surface)	√		√	√	√
Out-of-plane heat transfer	√		√		
Highly conductive layer			√	√	√

The first half of this chapter starts by treating convection and conduction together with material properties and other subdomain settings (see “Convection and Conduction”



on page 29). Then follows four sections dealing with boundary conditions in general (“Boundary Conditions” on page 54), radiation (“Radiative Heat Transfer” on page 67), highly conductive layers (“Highly Conductive Layers” on page 77), and out-of-plane heat transfer (“Out-of-Plane Heat Transfer” on page 81).

The second half of the chapter contains tutorials on how to use the General Heat Transfer application mode. It starts with the section “Building and Solving General Heat Transfer Models” on page 86. The accompanying *Heat Transfer Module Model Library* also includes many examples using this application mode.

### *Variables and Space Dimensions*

---

To start modeling with the General Heat Transfer application mode, go to the **Model Navigator**, open the **Heat Transfer Module** folder, and then select **General Heat Transfer**.

The General Heat Transfer application mode is available in 1D, 2D, and 3D, as well as for axisymmetric models using cylindrical coordinates in 1D and 2D. The dependent variables are the temperature,  $T$ , and the radiosity,  $J$ . The radiosity variable is necessary only in models involving radiative heat transfer.

When this application mode starts, only conductive heat transfer is active. To include convection, you must switch it on explicitly for the subdomains where it applies. Because radiation is a boundary feature, you enable it when specifying boundary conditions.

## Application Mode Properties

Application Mode Properties are global settings that control the behavior of the General Heat Transfer application mode. Open the **Application Mode Properties** dialog box (Figure 4-1) by going to the **Physics** menu and selecting **Properties**.

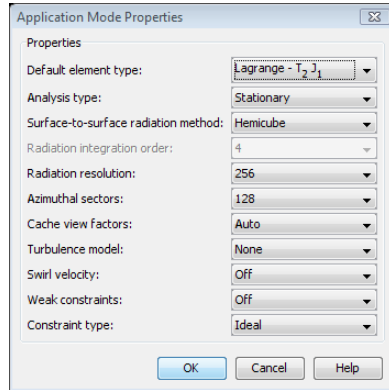


Figure 4-1: The Application Mode Properties dialog box.

Not all properties are active in all space dimensions and for all analysis types. The following settings are available:

- **Default element type:** A list of elements, where the selected element is the default on all new subdomains. Available elements include
  - **Lagrange -  $T_2 J_1$**
  - **Lagrange -  $T_3 J_1$**
  - **Lagrange -  $T_4 J_1$**
  - **Lagrange - Linear**
  - **Lagrange - Quadratic**
  - **Lagrange - Cubic**
  - **Lagrange - Quartic**
  - **Lagrange - Quintic**

The default element is **Lagrange -  $T_2 J_1$** . The three first elements all use order 1 for the radiosity variable; the remaining elements use the same order for both the temperature and radiosity variable.

- **Analysis type:** Choose between **Stationary** and **Transient** formulations.

- **Out-of-plane heat transfer:** Use this property to enable or disable out-of-plane heat transfer; by default it is disabled. This property is available in 1D Cartesian and 2D Cartesian coordinate systems. See “Out-of-Plane Heat Transfer” on page 81 for more details on out-of-plane heat transfer.
- **Surface-to-surface radiation method:** Available methods are **Direct area integration** and **Hemicube**. This setting controls the way view factors (Equation 2-14 and Equation 2-15) are evaluated. For more information see “View Factor Evaluation” on page 68.
- **Radiation integration order:** This property is available only for the Direct area integration radiation method, for which it controls the order of the quadrature rule that COMSOL Multiphysics applies when evaluating the surface integral.
- **Radiation resolution:** For the Hemicube radiation method, you can set the linear resolution of the z-buffer (see page 69).
- **Azimuthal sectors:** When evaluating surface-to-surface radiation in an axisymmetric geometry, the General Heat Transfer application mode must first construct a virtual 3D geometry by rotating the 2D mesh about the model axis. The value of **Azimuthal sectors** controls the resolution of this sweep by specifying the number of sectors to a full revolution.
- **Cache view factors:** Evaluating view factors for surface-to-surface radiation is usually very time consuming. Therefore, saving view factors between irradiation evaluation calls is normally beneficial. The **Auto** option generally leaves the cache on. If you experience inexplicably long assembly times, you can force a saving of view factors by setting the property to **Yes**. By setting it to **No**, you can save memory at the cost of increased assembly overhead for the nonlinear and time-dependent solvers.
- **Turbulence model:** Use this property to add terms for modeling of turbulent heat transfer. See the section “Turbulence Modeling” on page 51 for a brief description of the associated settings.
- **Swirl velocity:** This property is available in 1D Axisymmetric and 2D Axisymmetric coordinate system. It makes it possible to account for an azimuthal velocity which, if non-constant, gives a contribution to viscous heating.
- **Weak constraints:** A property that controls whether or not weak constraints are active. Weak constraints introduce Lagrange multipliers, which provide extremely accurate flux quantities. In contrast, the standard way of evaluating fluxes involves taking derivatives on boundaries, which is mesh-dependent and less accurate. When

weak constraints are enabled, all constraints are weak by default, but it is possible to change this setting for individual domains.

- **Constraint type:** Constraints can be ideal or non-ideal (see “Ideal vs. Non-Ideal Constraints” on page 359 in the *COMSOL Multiphysics Modeling Guide*).

### Scalar Variables

The General Heat Transfer application mode requires two scalar variables: the Stefan-Boltzmann constant,  $\sigma$ , and the universal gas constant,  $R_g$ . The former applies to radiative heat transfer, where it relates the radiative flux to surface temperature. The latter is central to the *ideal gas* fluid model.

To open the **Application Scalar Variables** dialog box (Figure 4-2), go to the **Physics** menu and choose **Scalar Variables**.

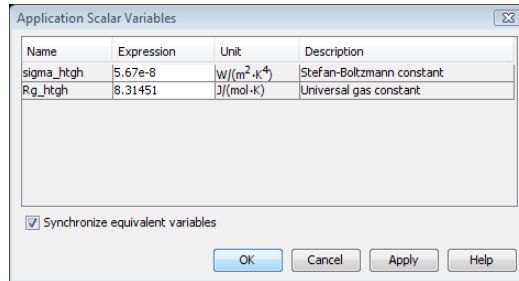


Figure 4-2: The Application Scalar Variables dialog box.

The default values of the constants are:

PROPERTY	VARIABLE	DESCRIPTION	DEFAULT VALUE	SI UNIT
$\sigma$	sigma_htgh	Stefan-Boltzmann constant	$5.67 \cdot 10^{-8}$	$W/(m^2 \cdot K^4)$
$R_g$	Rg_htgh	Universal gas constant	8.31451	$J/(mol \cdot K)$

Change the default values only if you model with the **Base unit system** model setting **None** using non-SI units. If you use any of the other base unit systems, COMSOL Multiphysics converts the values in SI units for these constants to the corresponding values in the selected base unit system.

# Convection and Conduction

The General Heat Transfer application mode models heat transfer by convection, conduction, and radiation. This section describes the equation governing convection and conduction, its parameters, and how to access them in the COMSOL Multiphysics user interface.

## *PDE Formulation*

---

When all features are active, the General Heat Transfer application mode solves the following equation on subdomains:

$$\rho C_p \frac{\partial T}{\partial t} + \nabla \cdot (-k \nabla T) = Q - \rho C_p \mathbf{u} \cdot \nabla T + \tau : \mathbf{S} - \frac{T \partial \rho}{\rho \partial T} \bigg|_p \left( \frac{\partial p}{\partial t} + \mathbf{u} \cdot \nabla p \right) \quad (4-1)$$

For some background, see “The Heat Equation” on page 6. In the above equation:

- $\rho$  is the density (kg/m<sup>3</sup>)
- $C_p$  is the specific heat capacity at constant pressure (J/(kg·K))
- $T$  is absolute temperature (K)
- $k$  is the thermal conductivity (W/(m·K))
- $\mathbf{u}$  is the velocity vector (m/s)
- $p$  is pressure (Pa)
- $\mathbf{S}$  is the strain rate tensor (1/s):

$$\mathbf{S} = \frac{1}{2}(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)$$

- $\tau$  is the viscous stress tensor (Pa):

$$\tau = \eta \left( 2\mathbf{S} - \frac{2}{3}(\nabla \cdot \mathbf{u})\mathbf{I} \right)$$

with viscosity  $\eta$  (Pa·s) and  $\mathbf{I}$  denotes the identity tensor

- $\tau : \mathbf{S} \equiv \sum \tau_{ij} \mathbf{S}_{ij}$  is the contraction of the two tensors  $\tau$  and  $\mathbf{S}$
- $Q$  contains heat sources other than viscous heating (W/m<sup>3</sup>)

For a steady-state problem, the temperature does not change over time and the first term on the left-hand side of Equation 4-1 disappears. The second term on the

right-hand side represents convective heat transfer and is not active by default, neither are the third and fourth terms, which represent viscous heating and pressure work, respectively. These last two terms can only be switched on when convective heat transfer is active.

The pressure-work term, furthermore, is only available for transient analysis of a compressible gas. Note also that compared to Equation 2-1, a term representing transport of acoustic waves has been dropped. This assumption is valid for small Mach numbers.

### Subdomain Settings

To enter material properties for conductive heat transfer, go to the **Physics** menu, select **Subdomain Settings** to open the like-named dialog box, and then go to the **General** page (Figure 4-3).

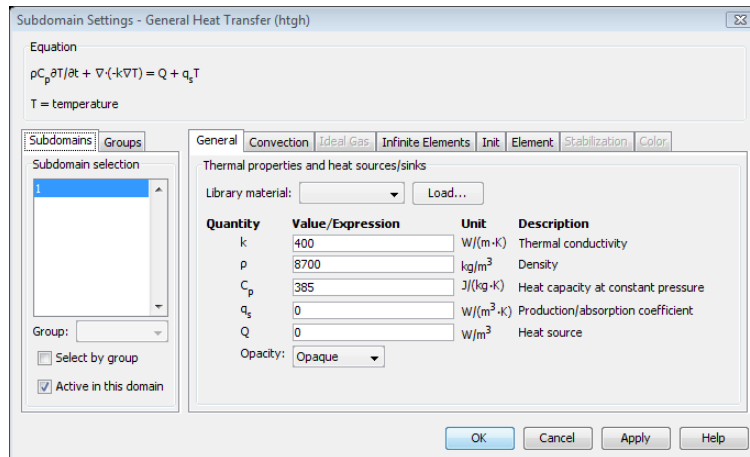


Figure 4-3: The General page of the Subdomain Settings dialog box.

To enable convective heat transfer and define the convection properties of the problem, go to the **Convection** page and select the **Enable convective heat transfer** check box at the top of the page (Figure 4-4).

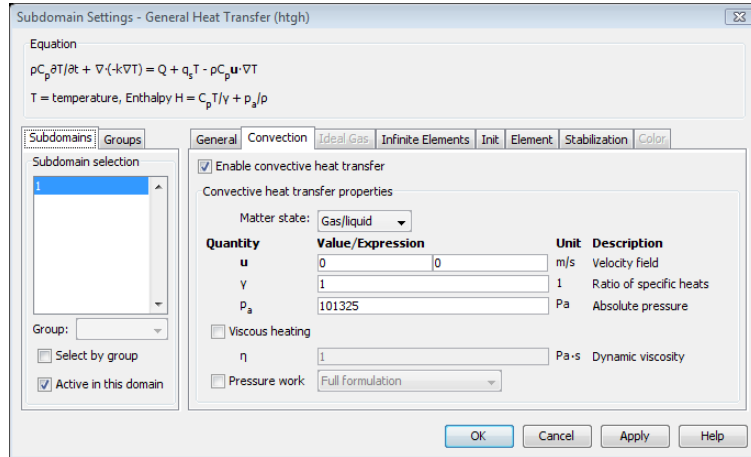


Figure 4-4: The Convection page of the Subdomain Settings dialog box.

The subdomain quantities available on the **General** and **Convection** pages are:

QUANTITY	VARIABLE	DESCRIPTION
$\rho$	rho	Density
$C_p$	C	Heat capacity at constant pressure
$k$	k	Thermal conductivity
$\underline{k}$		Thermal conductivity tensor
$k_{ij}$	$kx_i x_j$	Thermal conductivity tensor, $x_i x_j$ component
$Q$	Q	Heat source
$q_s$	qs	Production/absorption coefficient
$\gamma$	gamma	Ratio of specific heats
$u, v, w$	u, v, w	$x_1$ -, $x_2$ -, $x_3$ -velocity
$p_a$	p	Absolute pressure
$\eta$	eta	Dynamic viscosity

In addition, there are controls for **Opacity**, **Matter state**, **Pressure work**, and **Viscous heating**.

**Density** The density can be a function of the temperature, and also of the pressure in a coupled flow simulation. When **Matter state** is set to **Ideal gas**, the density is defined through the ideal gas law and the edit field is unavailable.

**Heat capacity**  $C_p$  describes the amount of heat energy required to produce a unit temperature change in a unit mass at constant pressure. The heat capacity is normally a function of temperature. For an ideal gas, you can choose to specify either  $C_p$  or the ratio of specific heats,  $\gamma$ , but not both.

**Thermal conductivity** You can enter  $k$  either as a scalar, which is appropriate for fluids and isotropic solids, or as a tensor describing an anisotropic material. If you have activated a turbulence model,  $k$  is replaced by  $(k + k_T)$  where  $k_T$  is the turbulent heat conductivity.

**Heat source**  $Q$  describes heat generation within the domain. Express heating and cooling with positive and negative values, respectively.

**Production/absorption coefficient** In many senses, Equation 4-1 is much like an advection-diffusion-reaction equation. If such equations have strong source terms, they can become numerically unstable. GLS stabilization can however stabilize with respect to source terms of the type  $q_s T$ . Theoretically,  $q_s = q_s(\mathbf{x}, t)$ , but some extra stability can be gained as long as  $q_s$  is non-singular.

**Ratio of specific heats**  $\gamma$  is the ratio of heat capacity at constant pressure,  $C_p$ , to heat capacity at constant volume,  $C_v$ . When using the ideal gas law to describe a fluid, specifying  $\gamma$  is enough to evaluate  $C_p$ . For common diatomic gases such as air,  $\gamma = 1.4$  is the standard value. Apart from its significance in the ideal gas case, the value of  $\gamma$  is important for GLS stabilization (see “Numerical Stability—Stabilization Techniques” on page 193). When you select GLS stabilization, the  $\gamma$  field becomes enabled also for **Matter state** settings other than **Ideal gas**.

**Velocity** These components describe the velocity field in the subdomain. For 1D enter  $u$ ; for 2D enter  $u, v$ ; and for 3D enter  $u, v, w$ . The velocities are central to heat transfer by convection, but they can also generate viscous heating if the velocity field is nonconstant.

**Absolute pressure** The absolute pressure,  $p_a$ , is used to calculate enthalpy, which in turn is used to calculate the convective heat flux post processing variable. It is also used in the **Ideal gas** law and in calculating the **Pressure work**. Observe that the edit field can



contain a dual pressure variable that via an ODE keeps track of the pressure level in the domain.

**Dynamic viscosity** The dynamic viscosity  $\eta$  is used together with the velocity expressions to evaluate the viscous heating when you have selected the corresponding check box.

**Opacity** This setting is relevant only for problems involving radiative heat transfer, described later in this manual.

**Matter state** The matter state setting affects the way the application mode treats density in the equations. The following options are available:

- **Gas/Liquid**—the density is expected to be a general function of pressure and Temperature (default).
- **Ideal gas**—the density is defined through the ideal gas law. Selecting this option activates the **Ideal Gas** page on which you can specify related settings.
- **User defined**—the same as **Gas/Liquid**.
- **Solid**—the density is a function of temperature but not of pressure. The enthalpy is approximated by  $C_p T/\gamma$  because the term  $p_a/\rho$  is very small. Hence, you only need to specify the **Absolute pressure**,  $p_a$ , when **Pressure work** is activated. **Viscous heating** is disabled because it does not apply to solids.

**Pressure work** This check box activates the pressure work term in Equation 4-1. For transient simulation, it exists in two versions: a **Full formulation** and a **Low-Mach number formulation**. The latter excludes the term  $\mathbf{u} \cdot \nabla p$  which is small for most flows with low Mach number. The pressure work is calculated using the **Absolute pressure** variable.

**Viscous heating** Select the **Viscous heating** check box to include the corresponding heat source term in Equation 4-1.

**Enthalpy** As can be seen in Figure 4-4, the enthalpy,  $H$ , is approximated as  $H = C_p T/\gamma - p/\rho$  which is an exact relation for an incompressible substance with respect to the reference state  $T = 0, p = 0$ . This approximation does not affect the calculations but is merely used to calculate the convective heat flux post processing variable,  $\rho \mathbf{u} H_0$  where  $H_0 = H + (1/2)\mathbf{u} \cdot \mathbf{u}$  is the total enthalpy.

## IDEAL GAS SETTINGS

If you select **Ideal gas** from the **Matter state** list, the corresponding page is activated.

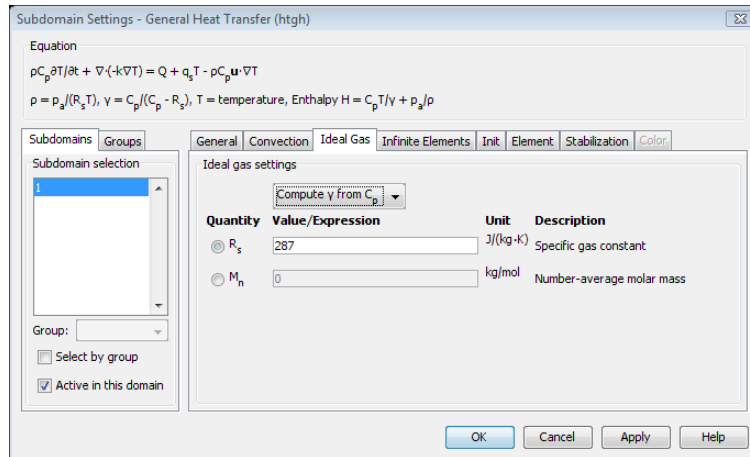


Figure 4-5: The Ideal Gas page of the Subdomain Settings dialog box.

On the **Ideal Gas** page you enter data required for calculating  $\rho$  and either  $C_p$  (heat capacity at constant pressure) or  $\gamma$  (ration of specific heats). You decide which one of  $C_p$  and  $\gamma$  that is specified and which one that is calculated from the ideal gas law using the list at the top of the **Ideal Gas** page.

The ideal gas law utilizes the **Absolute pressure** defined on the **Convection** page.

Finally, you can choose to specify either the specific gas constant,  $R_s$ , or the number-average molar mass,  $M_n$ . If you choose the latter, the Heat Transfer Module uses the molar gas constant  $R_g = 8.314 \text{ J}/(\text{mol}\cdot\text{K})$  by default.

## INFINITE ELEMENTS

Many environments that are modeled with finite elements are in reality unbounded or open, meaning that the heat exchange stretches toward infinity. The easiest approach to modeling an unbounded domain is to extend the simulation domain “far enough” that the influence of the terminating boundary conditions at the far end becomes negligible. This approach can create unnecessary mesh elements and make the geometry difficult to mesh due to large differences between the largest and smallest object.

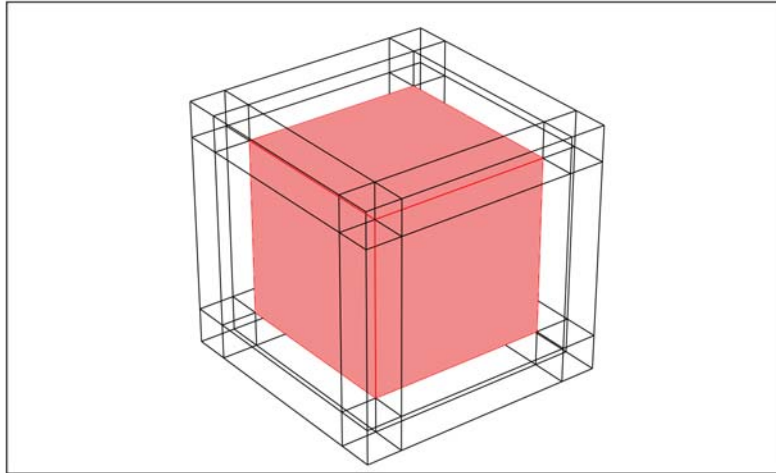
Another approach is to use *infinite elements*. There are many implementations of infinite elements available, and the elements used in the Heat Transfer Module are

often referred to as *mapped infinite elements* (see Ref. 1). This implementation maps the model coordinates from the local, finite-sized domain to a stretched domain. The inner boundary of this stretched domain coincides with the local domain, but at the exterior boundary the coordinates are scaled toward infinity:

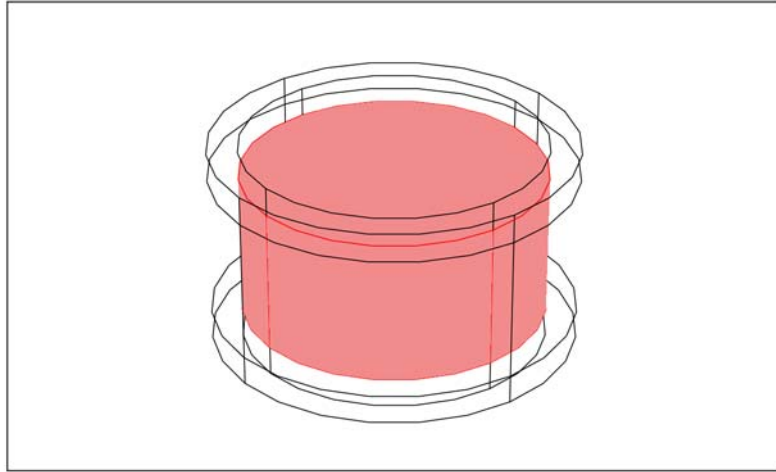
$$t' = t_0 \frac{\delta t}{t_0 + \delta t - t}$$

The inner coordinate,  $t_0$ , and the width of the infinite element region,  $\delta t$ , are input parameters for each region. The software uses default values for these properties for geometries that are Cartesian, cylindrical, or spherical. However, these default parameters might not work well for complex geometries, so it might be necessary to define other parameters. The following figures show typical examples of infinite element regions that work nicely for each of the infinite element types. These types are:

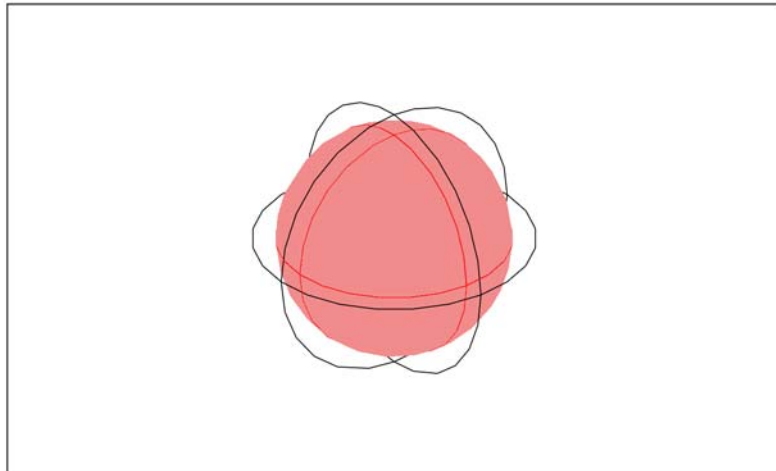
- Stretching in Cartesian coordinate directions, labeled **Cartesian**.
- Stretching in cylindrical directions, labeled **Cylindrical**.
- Stretching in spherical direction, labeled **Spherical**.
- User-defined coordinate transform for general infinite elements.



*Figure 4-6: A square surrounded by typical infinite-element regions of Cartesian type.*



*Figure 4-7: A cylinder surrounded by typical cylindrical infinite-element regions.*



*Figure 4-8: A sphere surrounded by a typical spherical infinite-element region.*

If you use other shapes for the infinite element regions not similar to the shapes shown in the previous figures, it might be necessary to define the infinite-element parameters manually. The software stores the default parameters in variables with the naming convention `param_guess_suffix`, where `param` is the name of the parameter, and

*suffix* is the application mode suffix. You can check their values by choosing **Equation System>Subdomain Settings** from the **Physics** menu. Click the **Variables** tab and look for variables with `_guess_` in the name.

Enabling infinite elements in a domain simultaneously disables some features that are not compatible with the coordinate mapping. These features are:

- Streamline diffusion, Crosswind diffusion, and Isotropic diffusion found on the **Stabilization** page (see Figure 4-10).
- Pressure work and Viscous heating found on the **Convection** page (see Figure 4-4).

### Using Infinite Elements

Select the subdomains where you want to place infinite elements, and then select the appropriate type of infinite element from the **Type of infinite element** list, for example, **Cartesian** (see Figure 4-9). Define which space coordinates to stretch by selecting the corresponding check boxes. Also, specify the center of your simulation system in the **Center point** edit fields; this is typically the location or center point of your heat source in the physical domain. The estimated inner coordinates, e.g. `S0x_guess_htgh` in Figure 4-9, is calculated with respect to the defined **Center point**.

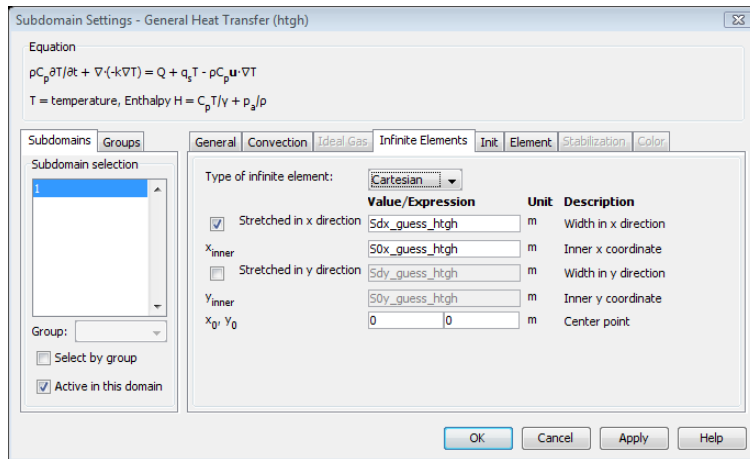


Figure 4-9: The subdomain settings for infinite elements.

### Known Issues When Modeling Using Infinite Elements

When modeling with infinite elements you should be aware of the following:

- The expressions resulting from the stretching get quite complicated for spherical and cylindrical infinite elements in 3D. This increases the time for the assembly stage

in the solution process. After the assembly, the computation time and memory consumption are comparable to a problem without infinite elements. The number of iterations for iterative solvers might increase if the infinite element regions have a coarse mesh.

- Infinite element regions deviating significantly from the typical configurations shown in the beginning of this section can cause the automatic calculation of the infinite element parameter to give erroneous result. Enter the parameter values manually if you find that this is the case.
- The infinite element region is designed to model uniform regions extended toward infinity. Avoid using objects with different material parameters or boundary conditions that influence the solution inside an infinite element region.
- 1D and 2D cases, and also 3D cases with multiple symmetry planes that effectively make them behave as 1D or 2D, cannot have only Dirichlet conditions at infinity.

### STABILIZATION SETTINGS

The **Stabilization** page of the **Subdomain Settings** dialog box is activated when convection is enabled (see Figure 4-4 on page 31).

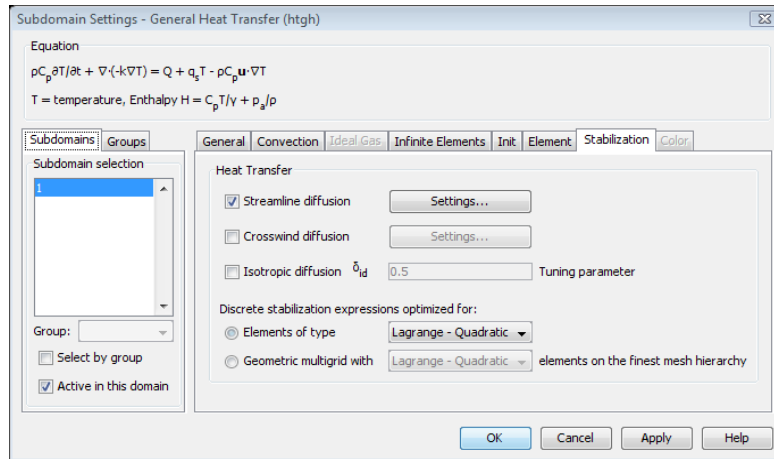


Figure 4-10: The Stabilization page of the Subdomain Settings dialog box.

There are three types of stabilization available:

- Streamline diffusion
- Crosswind diffusion
- Isotropic diffusion

Stabilization is needed because pure Galerkin discretization is unstable for convection-dominated or source-term dominated transport equations. The stabilization methods and their effects on the numerical solution are described in Chapter 17, “Stabilization Techniques,” in the *COMSOL Multiphysics Modeling Guide*.

For optimal functionality, the exact weak formulations and constants of GLS, SUPG, and crosswind diffusion should depend on the order of the basis functions (elements). You can control this dependence through two option buttons, each with its corresponding list, at the bottom of the **Stabilization** page shown in Figure 4-10. In most cases, the model uses only one element type. If this is the case, select this type from the **Elements of type** list. This is the default setting for all dimensions but 3D.

However, when using geometric multigrid (either as a solver or as a preconditioner), the equations can be discretized with different elements on the different multigrid hierarchies. Then, select the element type that is used on the finest mesh hierarchy from the **Geometric multigrid with** list. This is the default setting for 3D, because the default solver for General Heat in 3D is BiCGStab with geometric multigrid as preconditioner. The resulting weak expressions then take forms adapted to multigrid hierarchies created in a way equivalent to the hierarchy generation method **Lower element order first (any)** (see “Constructing a Multigrid Hierarchy” on page 558 in the *COMSOL Multiphysics Reference Guide*).

The values of the GLS and SUPG constants follow Ref. 3 and Ref. 4 and cannot be changed without performance losses. In contrast, the constant  $C^e$  for Crosswind Diffusion (see Equation 4-2) is a tuning constant. Ref. 5 gives the recommendations 0.7 for quadratic elements and 0.35 for linear elements. In general, the higher the value of this constant, the more diffusion it introduces. COMSOL uses the values for  $C^e$  listed in Table 4-1.

TABLE 4-1: VALUES OF  $C^e$  FOR CROSSWIND DIFFUSION IN COMSOL MULTIPHYSICS

ELEMENT ORDER	$C^e$
1	0.7
2	0.4
3	0.2
4	0.1
5	0.05

### Streamline Diffusion

Streamline diffusion is active by default. Figure 4-11 shows the **Streamline Diffusion Settings** dialog box. There are three types of streamline diffusion available:

- Galerkin least-squares (GLS)—GLS is the default choice and is by all references superior to any of the other methods available. GLS for the heat transfer equation can be made a part of the GLS method for the fully compressible Navier-Stokes (see “Stabilization Settings for Weakly Compressible Navier-Stokes” on page 225).
- Streamline upwind Petrov-Galerkin (SUPG)—a subset of GLS that stabilizes only with respect to convection, not reaction as well. SUPG is computationally less expensive than GLS but its stabilization properties are not as good (Ref. 2).
- Anisotropic diffusion—this is the only streamline diffusion method that needs a user defined constant. The **Tuning parameter** edit field becomes active when you select **Anisotropic diffusion** from the list.

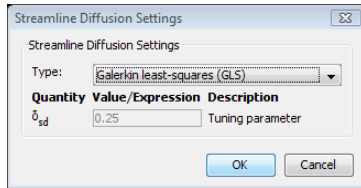


Figure 4-11: The Streamline Diffusion Settings dialog box.

The constants of GLS and SUPG are controlled as described in the section “Stabilization Settings” on page 38. For anisotropic diffusion, there is no theory on what the value of this constant is, and the optimal value depends on the model.

### Crosswind Diffusion

Crosswind diffusion provides extra diffusion in the region of sharp gradients. There are two types of crosswind diffusion available:

- Scalar stabilization—the method adds the following contribution to the weak formulation (Ref. 5):

$$-\sum_{e=1}^{N_{el}} \int_{\Omega^e} \frac{1}{2} \max\left(0, C^e - \frac{2k}{h|\beta|}\right) \frac{h|R|}{|\nabla T|} \nabla \hat{T} \left( I - \frac{\mathbf{u} \otimes \mathbf{u}}{|\mathbf{u}|^2} \right) \nabla T d\Omega \quad (4-2)$$

where  $R$  is the PDE residual,  $T$  is the test function for  $T$ , and



$$\beta = \begin{cases} \frac{\rho C_p (\mathbf{u} \cdot \nabla T)}{|\nabla T|^2} \nabla T & \text{if } \nabla T \neq 0 \\ 0 & \text{if } \nabla T = 0 \end{cases} \quad (4-3)$$

The added diffusion is orthogonal to the streamline diffusion, so you can use streamline diffusion and crosswind diffusion simultaneously. Scalar stabilization does not explicitly take any multiphysics coupling into account and is therefore the appropriate choice if the temperature equation is the only equation solved for or, if the temperature is a passive scalar (that is, if other equations are completely independent of the temperature) or, if other equations depend only weakly on the temperature.

- System stabilization—this setting is available when the model contains at least one Navier-Stokes application mode. It is effective only when the following conditions are met:
  - The Navier-Stokes application mode has Weakly compressible flow set to On.
  - The Navier-Stokes application mode uses Crosswind diffusion and has its stabilization adapted to non-isothermal flow (see “Stabilization Settings for Weakly Compressible Navier-Stokes” on page 225).

This option regards the heat transfer equation as a part of the compressible Navier-Stokes equations and add diffusion accordingly. Hence, system stabilization is the appropriate choice for non-isothermal compressible flows.

Figure 4-12 shows the **Crosswind Diffusion Settings** dialog box.

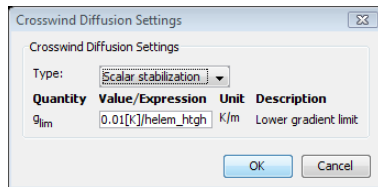


Figure 4-12: The Crosswind Diffusion Settings dialog box.

The list in Figure 4-12 is available only when the model contains at least one Navier-Stokes application mode. It is used to switch between **Scalar stabilization** and **System stabilization**. When the list is not available, scalar stabilization will be added.

When **Scalar stabilization** is selected from the list in Figure 4-12 or when the list is not present, the  $g_{lim}$  edit field is activated. The variable  $g_{lim}$  is needed because both Equation 4-2 and Equation 4-3 contain terms of the form  $1/|\nabla T|$ , which become

singular if  $\nabla T = 0$ . Hence, all occurrences of  $1/|\nabla T|$  are replaced by  $1/\max(|\nabla T|, g_{\text{lim}})$  where  $g_{\text{lim}}$  is a measure of a small gradient.

### Isotropic Diffusion

Isotropic diffusion is described in the section “Isotropic Diffusion” on page 500 of the *COMSOL Multiphysics Modeling Guide*.

## Loading Material Properties from the Library

For most application modes in COMSOL Multiphysics, you can load a material and its accompanying properties directly from within the **Subdomain Settings** dialog box. First click the **Load** button as indicated in Figure 4-13.

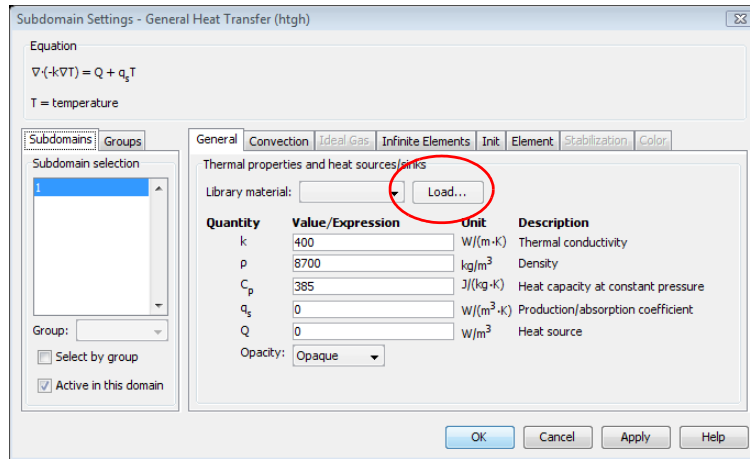


Figure 4-13: The Load button associated with library materials in the Subdomain Settings dialog box.

This action opens the **Materials/Coefficients Library** dialog box (see Figure 4-14). From there you can select a material to load.

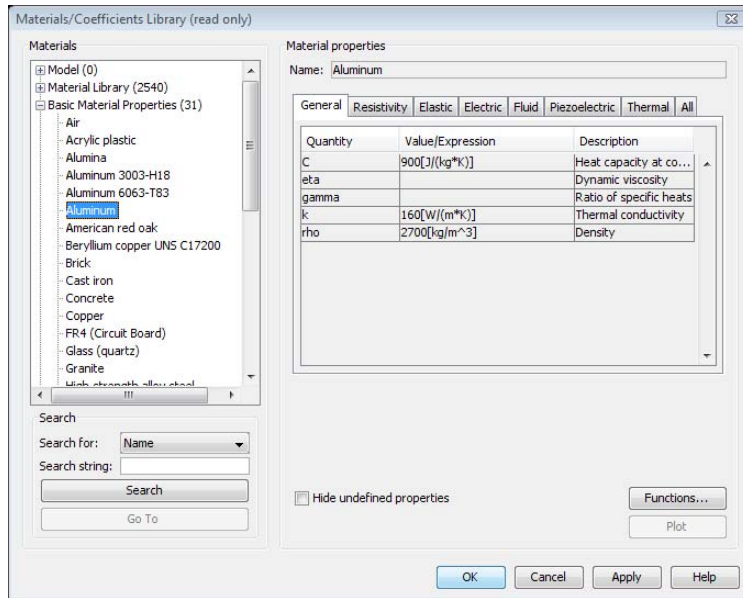


Figure 4-14: The **Materials/Coefficients Library** dialog box.

The **Materials** list on the left side contains the installed library folders plus a model folder that contains already-selected materials and coefficients. To load a material into the **Model** folder, select it from the **Materials** list and click **Apply** or **OK**; it now appears in the **Model** folder, which shows all the materials available for the model.

After clicking **OK**, the **Materials/Coefficients Library** dialog box closes and you return to the **Subdomain Settings** dialog box. Now the software has loaded the selected material properties into the corresponding edit fields of that dialog box. The edit fields that

contain data taken from the materials library you just selected appear in bold (see Figure 4-15).

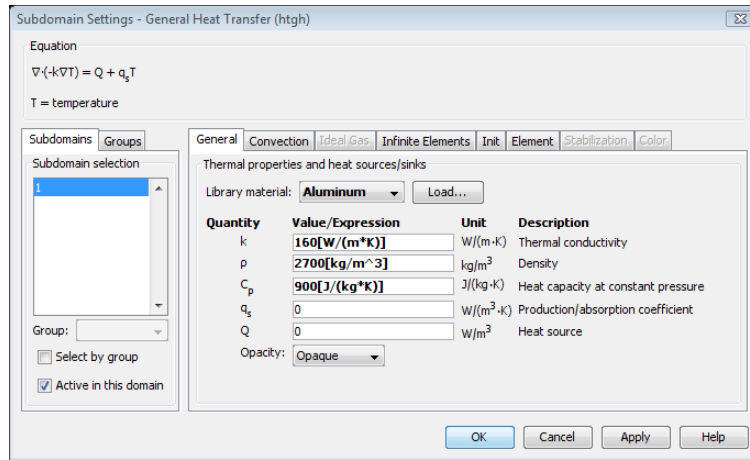


Figure 4-15: The Subdomain Settings dialog box after loading aluminum from the material library. Boldface characters indicate that the library material is active.

### The Material Libraries

As noted earlier, the **Materials** list in the **Materials/Coefficients Library** dialog box contains multiple expandable folders. The list includes **Model**, **Basic Material Properties**, and additional libraries depending on the modules that your license includes.

Also, if your license includes the COMSOL Material Library, this also appears as a separate folder in the **Materials** list.

---

**Note:** The COMSOL Material Library is a separate add-on product that includes over 2500 materials and about 20,000 properties, most of which are temperature-dependent functions.

---

### The Liquids and Gases Library and Material Property Functions

In many cases of modeling the material properties varies with the dependent variables describing the state (temperature, pressure, concentration, potential, stress, and so on). COMSOL Multiphysics is capable of describing material properties as functions

of the dependent variables. The following discussion exemplifies use and access a library with material property functions (in this case Liquids and Gases), and also how to edit it and create your own material property functions.

The Liquids and Gases material library contains thermal and fluid dynamic properties for a set of common fluids. All properties are given as functions of temperature and at atmospheric pressure, except the density, which for gases is also a function of the local pressure. The library also contains surface and interface tensions for a selected set of liquid/gas and liquid/liquid systems. All functions are based on data collected from scientific publications. The literature references are noted in the text file `fluids_lib.txt`, located in the data folder of your local COMSOL installation.

If you click the **Load** button in the **Subdomain Settings** dialog box of any application mode and select to expand **Liquids and Gases** you can choose from various fluids, both liquids or gases (as depicted in Figure 4-16). Their properties are described at atmospheric pressure conditions.

### *Using Material Property Functions*

---

In many modeling situations, the material properties vary with the dependent variables describing the state (temperature, pressure, concentration, potential, stress, and so on). In COMSOL Multiphysics you can describe material properties as functions of the dependent variables. The following discussion exemplifies how to use a library with material property functions (in this case Liquids and Gases), and also how to edit it and create your own material property functions.

---

**Note:** The data-fitted functions expect temperature and pressure arguments to be expressed in the SI units *kelvin* (K) and *pascal* (Pa), respectively, and return values in appropriate SI units. Unit expressions are automatically inserted to handle the conversions to and from the model's base unit system.

---

If you click the **Load** button in the **Subdomain Settings** dialog box of any application mode and select to expand **Liquids and Gases** you can choose from various fluids, both liquids and gases (see Figure 4-16).

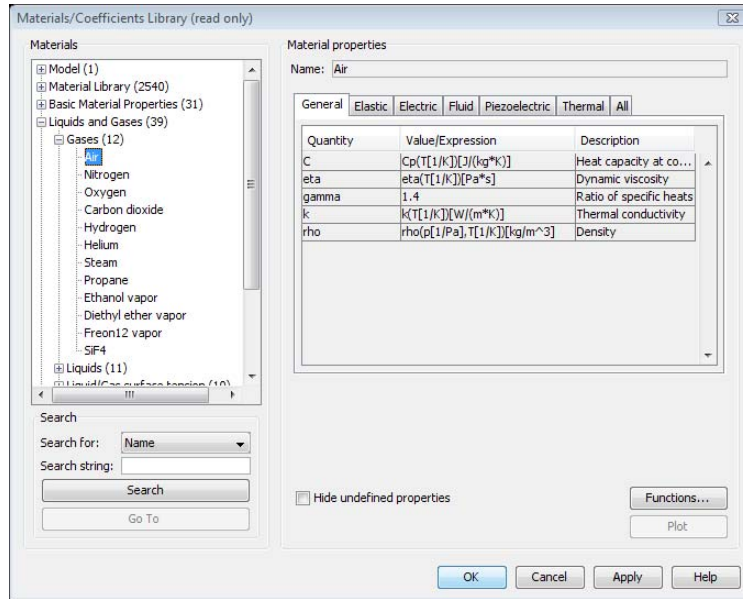


Figure 4-16: The Materials/Coefficients Library dialog box, with Liquids and Gases expanded.

If you load a material that uses a function such as those in the Liquids and Gases library, the **Subdomain Settings** dialog box looks like Figure 4-17, with function calls in the edit fields for the material properties.

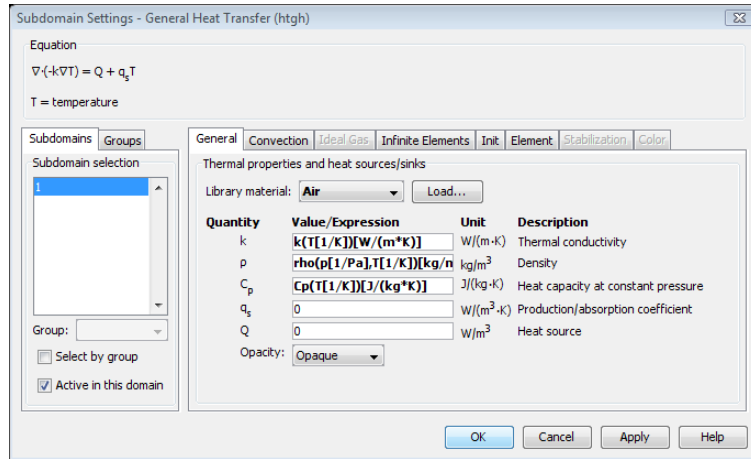


Figure 4-17: The Subdomain Settings dialog box after loading a material that uses function calls, for example, the materials from the fluid library.

In this example, the software specifies the material property for density with the function call

$$\rho(p[1/Pa], T[1/K])[kg/m^3]$$

which is a function call to the material loaded, in this case **Air**. The function uses two inputs: pressure,  $p$ , and temperature,  $T$ . The default settings are based on the assumption that the temperature variable in the model is  $T$  and that there is a pressure variable named  $p$ . Being dependent variables, these are expressed in the model's selected base unit system. The unit expressions inside the function calls convert the values from the model's unit system to nondimensional numbers corresponding to SI units, while the expression between the last brackets makes sure the returned SI value is interpreted correctly in the model's unit system.

In many cases you must change these function inputs. For example, if you model only heat transfer, there is no variable for pressure. In that case you must either specify the pressure directly in the function input or set up a constant or expression for the variable  $p$ . This constant or expression variable must have the dimension of pressure, which you achieve by adding a pressure unit to the expression, such as  $135[kPa]$ .

It is easy to alter the function input values and variables: simply click inside the parentheses delimiting the function argument and replace the default symbol with the desired value or variable. In the following figure you can see such a modification where a numerical value replaces the pressure variable.

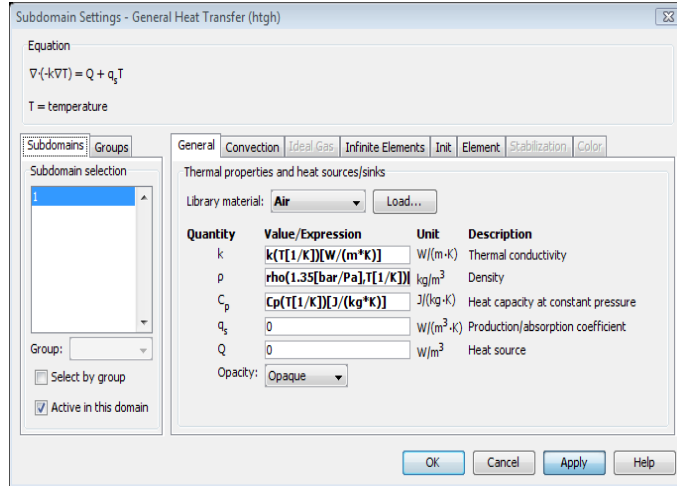


Figure 4-18: Modifying the function inputs. Notice how the unit syntax can convert argument values from any pressure unit to pascal.

In this case the function output is the density for air at the specified pressure and at the temperature given by the variable T.

---

**Note:** If you override one property defined by a library material by changing anything outside the function arguments, all other property expressions loose their connection to the material library. In particular, material functions appearing in other edit fields (now in plain text instead of in bold face) stop working. The proper way to edit one property of a loaded material is to change the material functions, as described later in this section, rather than editing the edit field in the **Subdomain Settings** dialog box.

---

## USING MATERIAL PROPERTY FUNCTION CALLS OUTSIDE THE SUBDOMAIN SETTINGS

You can also use a library material property function in a model in places other than the **Subdomain Settings** dialog box. One example might be to combine several



properties in an expression in the **Scalar Expressions** dialog box. To do so, you must first load the library material into the model using the **Materials/Coefficients Library** dialog box. This dialog box opens either, as described above, from the **Subdomain Settings** dialog box (then in read only mode), or directly from the **Options** menu. The currently loaded materials and their assigned names are listed in the **Model** folder in the **Materials** tree. (Figure 4-19).

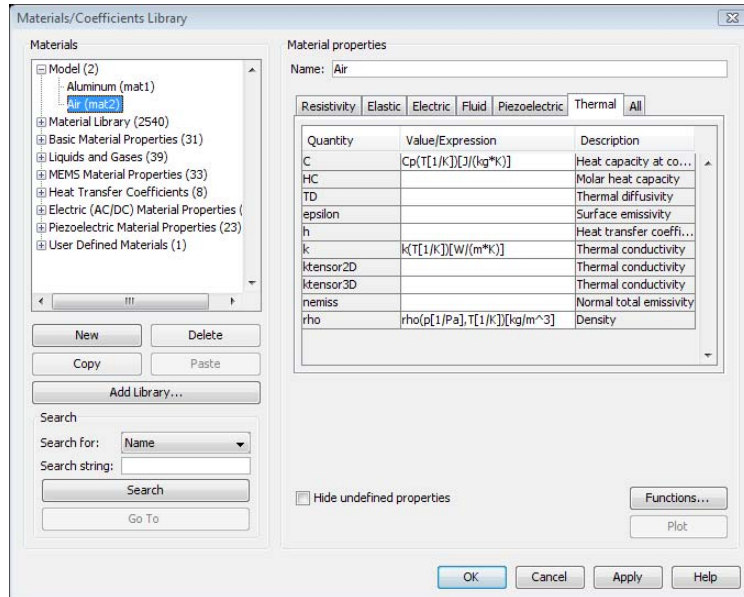


Figure 4-19: Viewing the materials loaded into a model along with their names in COMSOL Multiphysics.

In the above figure, two materials are already loaded into the model. COMSOL Multiphysics allocates local material names on the form **matX**, where X is a running number assigned in the order in which materials were loaded into the model. In this case, the user loaded **Aluminum** first, so it takes the name **mat1**.

To use a particular material-property function, you can start by copying the syntax shown in the **Value/Expression** column in the **Materials/Coefficients Library** dialog box. Then you must add **matX\_** in front of the function call. Thus, referring to Figure 4-19, to evaluate the function for thermal conductivity of air at 350 K, the syntax is

`mat2_k(350) [W/m*K]`

## Editing Material Properties

To change a property of a loaded material, choose **Materials/Coefficients Library** from the **Options** menu. This opens the dialog box in edit mode. Then select the desired material in the **Materials** list, click the **Value/Expression** field of interest, and change the expression as depicted in Figure 4-20. In this example the user is adding `k_turb` to the original function for thermal conductivity, where `k_turb` is a variable that must be defined elsewhere in the model.

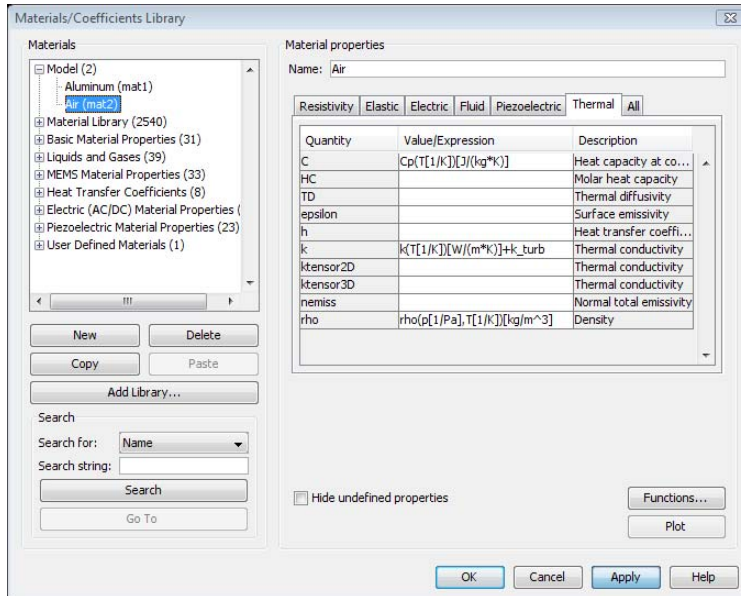


Figure 4-20: Editing a material property.

If you want to edit the function describing some material property, click the **Functions** button. This opens the **Functions** dialog box, where you can view and edit any function describing the material.

After changing a material property in this way, you need to reload all subdomain settings that use the material; otherwise the function call will not work. To reload the material, simply go to **Physics>Subdomain Settings** and select the modified material from the **Library material** list. The new expression then shows up in the edit field for

the corresponding material property in the **Subdomain Settings** dialog box (see Figure 4-21).

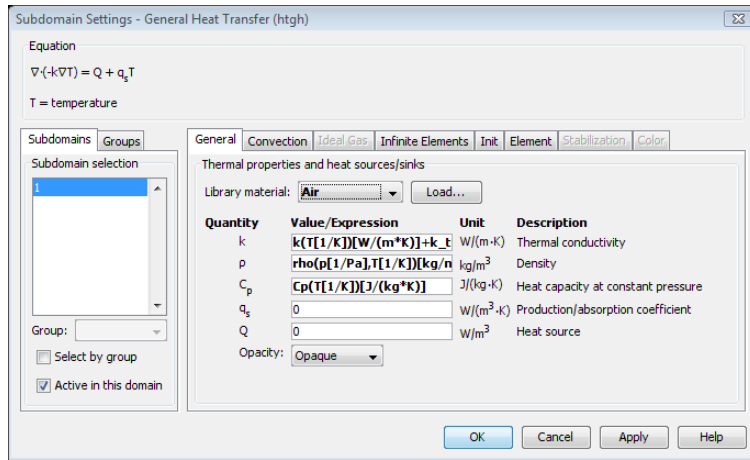


Figure 4-21: The update of a material property expression in the Subdomain Settings dialog box.

Note that you can only edit materials currently in the **Model** folder in this way and that any changes are local to the current model. The original material in the library file remains intact. Within the **Materials/Coefficients Library** dialog box you can, however, change a material’s name and then **Copy** and **Paste** it into one of the other libraries. Afterwards, clicking **OK** saves the new material for future use in the corresponding library text file.

You can also set up new materials by creating a new library file, as well as change the existing files using any text editor. A detailed description of this process appears in the *COMSOL Multiphysics User’s Guide*, where you also find complete documentation on the functionality of the Materials/Coefficients Library.

### Turbulence Modeling

Figure 4-22 shows the **Convection** page of the **Subdomain Settings** dialog box when the **Turbulence model** application mode property is activated. The laminar and turbulent viscosity must both be specified because they are used not only in the viscous heating but also in the turbulent thermal wall functions and in the models for the turbulent Prandtl number.

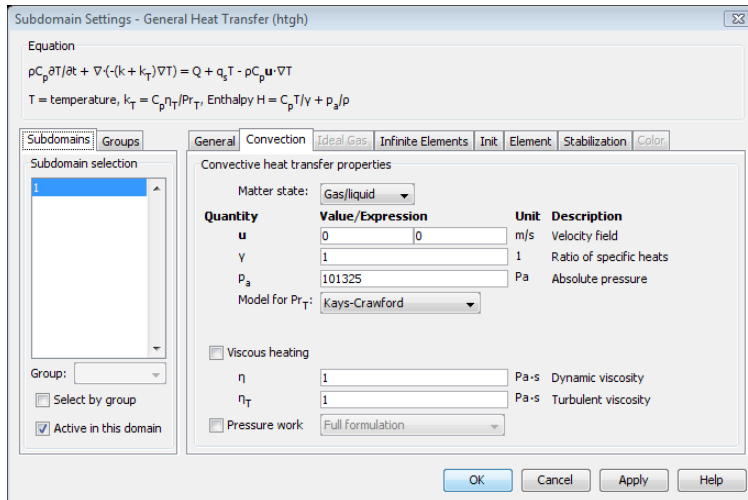


Figure 4-22: The Convection page of the Subdomain Settings dialog box when the Turbulence model application mode property is activated.

There are three options for modeling the turbulent Prandtl number:

- Kays-Crawford (default)
- Extended Kays-Crawford
- User defined

#### KAYS-CRAWFORD

This is a relatively exact model for  $Pr_T$ , still simple. In Ref. 6, it is compared to other models for  $Pr_T$  and found to be good for most kind of turbulent wall bounded flows except for liquid metals. The model is given by

$$Pr_T = \left( \frac{1}{2Pr_{T\infty}} + \frac{0.3}{\sqrt{Pr_{T\infty}}} \frac{C_p \eta_T}{k} - \left( 0.3 \frac{C_p \eta_T}{k} \right)^2 (1 - e^{-k/(0.3C_p \eta_T \sqrt{Pr_{T\infty}})}) \right)^{-1} \quad (4-4)$$

where  $Pr_{T\infty} = 0.85$  and  $k$  is the conductivity.

#### EXTENDED KAYS-CRAWFORD

Weigand and others (Ref. 7) suggested an extension of Equation 4-4 to liquid metals by introducing

$$\text{Pr}_{T_\infty} = 0.85 + \frac{100k}{C_p \eta \text{Re}_\infty^{0.888}} \quad (4-5)$$

where  $\text{Re}_\infty$  must be provided either as a constant or as a function of the flow field.

### References

---

1. O.C. Zienkiewicz, C. Emson, and P. Bettess, "A Novel Boundary Infinite Element," *Int. J. Num. Meth. Engrg*, vol. 19(3), pp. 393–404, 1983.
2. R. P. Bonet Chaple, "Numerical Stabilization of Convection-Diffusion-Reaction Problems," *DIAM Report, Delft University of Technology*, 2006.
3. I. Harari and T.J.R. Hughes, "What are  $C$  and  $h$ ? Inequalities for the analysis and design of finite element methods," *Comp.Meth. Appl. Mech. Engrg*, vol. 97, pp. 157–192, 1992
4. Y. Bazilevs, V.M. Calo, T.E. Tezduyar, and T.J.R. Hughes, "YZ $\beta$  discontinuity capturing for advection-dominated processes with application to arterial drug delivery," *Int.J.Num. Meth. Fluids*, vol. 54, pp. 593–608, 2007.
5. R. Codina, "Comparison of some finite element methods for solving the diffusion-convection-reaction equation," *Comp. Meth.Appl. Mech. Engrg*, vol. 156, pp. 185–210, 1998.
6. W.M. Kays, "Turbulent Prandtl Number — Where Are We?," *ASME J. Heat Transfer*, vol. 116, pp. 284–295, 1994.
7. B. Weigand, J.R. Ferguson, and M.E. Crawford, "An extended Kays and Crawford turbulent Prandtl number model," *Int. J. Heat Mass Transfer*, vol. 40 no. 17, pp. 4191–4196, 1997.

# Boundary Conditions

This section describes the overall structure of the boundary conditions you can apply on a General Heat Transfer model. The important contributions from radiation and highly conductive layers are discussed separately in subsequent sections.

## *Overview and Equations*

---

The General Heat Transfer application mode supports boundary conditions of two fundamental types—the Dirichlet condition

$$T = T_0 \quad \text{on } \partial\Omega$$

and the heat-flux (or Neumann) condition

$$-\mathbf{n} \cdot \mathbf{q} = q_0 + q_r + q_s + h(T_{\text{inf}} - T) \quad \text{on } \partial\Omega \quad (4-6)$$

where

- $\mathbf{q} = -k\nabla T$  is the conductive flux vector ( $\text{W}/\text{m}^2$ ).
- $\mathbf{n}$  is the normal vector of the boundary, pointing out from the domain.
- $q_0$  is inward heat flux ( $\text{W}/\text{m}^2$ ), normal to the boundary, from external sources.
- $q_r$  is net influx from radiation ( $\text{W}/\text{m}^2$ ).
- $q_s$  represents energy transferred from other parts of the boundary through a thin, highly conductive shell ( $\text{W}/\text{m}^2$ ).
- $h$  is a heat transfer coefficient ( $\text{W}/(\text{m}^2 \cdot \text{K})$ ).
- $T_{\text{inf}}$  is a reference bulk temperature (K).

Note that Equation 4-6 is not the whole truth in the presence of a Dirichlet condition. If you specify the temperature on a boundary, an additional term,  $\mu$ , known as a Lagrange multiplier appears on the right-hand side. This term represents the “reaction force” necessary to impose the constraint on the temperature but is not directly visible in the equation.

## **INTERIOR BOUNDARIES AND CONTINUITY**

An *interior boundary* is a boundary between two subdomains where heat transfer is active, as opposed to an external geometry boundary or a boundary where the General Heat Transfer application mode is active on one side only. You can specify the

temperature using the same type of Dirichlet condition on both exterior and interior boundaries. Equation 4-6, on the other hand, transforms into a heat flux discontinuity condition on interior boundaries:

$$-\mathbf{n}_{\text{up}} \cdot \mathbf{q}_{\text{up}} - \mathbf{n}_{\text{down}} \cdot \mathbf{q}_{\text{down}} = q_0 + q_r + q_s + h(T_{\text{inf}} - T) \quad \text{on } \partial\Omega$$

or equivalently

$$-\mathbf{n}_{\text{up}} \cdot (\mathbf{q}_{\text{up}} - \mathbf{q}_{\text{down}}) = q_0 + q_r + q_s + h(T_{\text{inf}} - T) \quad \text{on } \partial\Omega \quad (4-7)$$

since the normals on the up and down side of the boundary,  $\mathbf{n}_{\text{up}}$  and  $\mathbf{n}_{\text{down}}$ , differ only by a sign. The right-hand side represents a flux discontinuity, or equivalently, a heat source or heat sink depending on sign. The temperature is always continuous due to the continuity of the finite element field. If the right-hand side is zero, Equation 4-7 specifies continuity also in the normal heat flux. This is the default boundary condition on interior boundaries.

#### BOUNDARY CONDITIONS ON PAIRS

A *boundary pair* can be created where parts in an assembly are in contact with each other, either permanently through an *identity pair* or as a result of structural deformations constrained by a *contact pair*. Usually, a pair of boundaries connected in this way behaves like a single interior boundary, and the same boundary conditions can be applied. The default setting specifies continuity in both temperature and normal heat flux across the pair, just like on any interior boundary. Alternately, a fixed temperature can be specified, as well as a heat source or sink according to Equation 4-7.

All pair boundary conditions borrowed from interior boundaries require the temperature to be continuous across the pair. There are, however, other possibilities. The General Heat Transfer application mode includes a condition which models the behavior of a thin thermally resistive layer. Assuming that the layer is thin and tangential heat flux in the layer negligible, the jump in temperature across the boundary becomes proportional to the normal heat flux. You can express this as two separate but symmetric heat flux boundary conditions:

$$\begin{aligned} -\mathbf{n}_{\text{down}} \cdot (-k\nabla T)_{\text{down}} &= \frac{k_{\text{res}}}{d_{\text{res}}}(T_{\text{up}} - T_{\text{down}}) \quad \text{on } \partial\Omega_{\text{down}} \\ -\mathbf{n}_{\text{up}} \cdot (-k\nabla T)_{\text{up}} &= \frac{k_{\text{res}}}{d_{\text{res}}}(T_{\text{down}} - T_{\text{up}}) \quad \text{on } \partial\Omega_{\text{up}} \end{aligned} \quad (4-8)$$

where  $k_{\text{res}}$  is the thermal conductivity (W/(m·K)) and  $d_{\text{res}}$  the thickness (m) of the thin thermally resistive layer. The two boundaries together making up the pair are here referred to as *up* and *down*. In other contexts, *source* and *destination* or *master* and *slave* are more common as labels. This particular boundary condition is, however, symmetric with respect to the two sides.

## Boundary Settings

To specify boundary conditions, go to the **Physics** menu, select **Boundary Settings** to open the dialog box shown in Figure 4-23.

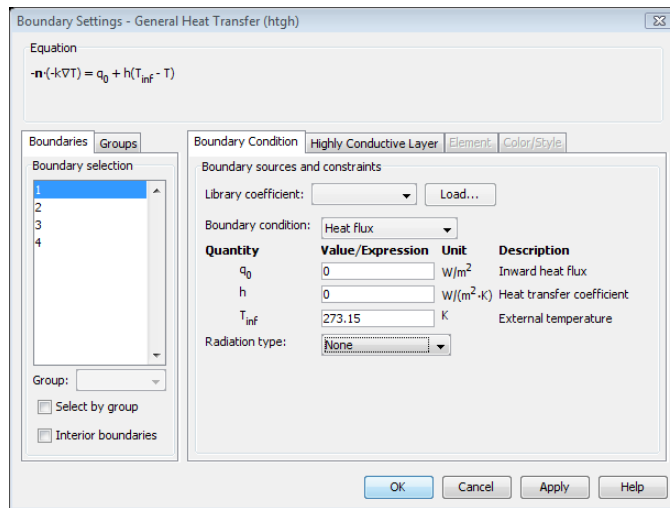


Figure 4-23: The Boundary Condition page in the Boundary Settings dialog box.

### BOUNDARY CONDITION TYPES

Select boundary condition type from the **Boundary condition** list on the **Boundary Condition** page. This setting controls what other parameters are available. The available boundary condition types are:

#### Heat Flux

Selecting the Heat flux boundary condition gives you access to all terms in Equation 4-6. The contributions from radiation and highly conductive layers are not enabled by default and must be switched on to take effect. These terms are treated separately below, see “Radiative Heat Transfer” on page 67 and “Highly Conductive Layers” on page 77.



The default form of the Heat flux boundary condition is

$$-\mathbf{n} \cdot \mathbf{q} = q_0 + h(T_{\text{inf}} - T) \quad \text{on } \partial\Omega$$

The first term,  $q_0$ , is a general heat flux entering the domain. It can model, for example, a heat source on the boundary, such as an electrical heating element, or a known inflow of energy, such as radiation with known intensity. Note that  $q_0$  can be a function of the temperature  $T$  and of other dependent variables.

In principle, you can enter the second term,  $h(T_{\text{inf}} - T)$  directly into  $q_0$ , but boundary conditions of this form are so common that the application mode supports them explicitly. They arise in at least three different situations:

- A solid exchanges energy with the surrounding environment through convective heat transfer, where  $h$  is the *heat transfer coefficient*, and  $T_{\text{inf}}$  is the external bulk temperature. The value of  $h$  depends on the geometry and the ambient flow conditions. See “Convective Cooling/Heating and Heat Transfer Coefficients” on page 15 for an introduction to the theory and “Using h Coefficients in the Heat Transfer Module” on page 61 for instructions on using the Heat Transfer Coefficient Library included with the Heat Transfer Module.
- The boundary is covered by a thin layer of a thermally insulating material and the temperature on the outside of this layer is known. If the thermal conductivity,  $k$ , of the layer is considerably lower than the thermal conductivity of the bulk material, a layer of thickness  $d$  corresponds to a heat transfer coefficient  $h = k/d$ .
- The solid boundary under consideration is in contact with another solid with known temperature  $T_{\text{inf}}$ , but the contact is not perfect and therefore acts as a resistive layer. The corresponding heat transfer coefficient is difficult to estimate but can be entered if it is empirically known.

#### *Insulation/Symmetry*

Insulation/Symmetry is a special case of the above Heat flux condition where all heat transfer mechanisms across the boundary have been disabled such that  $-\mathbf{n} \cdot \mathbf{q} = 0$ . This means that there is no temperature gradient across the boundary and no mass flow in the normal direction. In addition, the boundary does not take part in radiative heat transfer.

In a typical model, you apply the Insulation/Symmetry boundary condition to boundaries that do not exchange energy with the surroundings, either because they are thermally well insulated or because they lie on a symmetry plane that has been introduced to reduce model size.

### *Convective Flux*

The Convective flux boundary condition is also a special case of the general Heat flux condition and is actually identical to the Insulation/Symmetry boundary condition. In a model with convective heat transfer, this condition states that the only heat transfer over a boundary is by convection. The temperature gradient in the normal direction is zero, and there is no radiation. This is usually a good approximation of the conditions at an outlet boundary in a heat transfer model with fluid flow.

### *Temperature*

This boundary condition prescribes the temperature  $T_0$  at a boundary. The finite element algorithm returns a solution in which the condition is either true at each node point or in an integral sense, depending on the **Weak constraints** setting in the **Application Mode Properties** dialog box. Using weak constraints allows you to postprocess the heat flux over a boundary with a fixed temperature much more accurately. See “Using Weak Constraints” on page 358 in the *COMSOL Multiphysics Modeling Guide*.

The Temperature boundary condition is compatible with radiative heat transfer where it can act as a radiation source but also reflect incoming radiation. Because the temperature is given explicitly, the amount of radiation that enters or leaves the boundary does not affect the conditions at the boundary itself. It can, however, irradiate other boundaries. It is therefore possible to use a Temperature boundary as a fixed-temperature radiation source (see “Radiative Heat Transfer” on page 67).

### *Axial Symmetry (Axisymmetric Models Only)*

The Axial symmetry condition is available only for axisymmetric versions of the heat transfer application modes. Apply this boundary condition at the symmetry axis  $r = 0$ . Note that from the mathematical point of view, this condition is identical to the Insulation/Symmetry condition.

### *Thermal Wall Functions (Turbulence Models Only)*

The thermal wall functions are available only when the application mode property **Turbulence model** is activated.

The heat flux across the fluid-solid interface is assumed to be proportional to the temperature difference between the wall and the fluid on the opposite side of the viscous boundary layer:

$$q = \frac{\rho C_p C_\mu^{1/4} k_w^{1/2} (T_w - T)}{T^+}$$

Where the dimensionless temperature  $T^+$  is given by (Ref. 1):

$$T^+ = \begin{cases} \text{Pr} \delta_w^+ & \text{for } \delta_w^+ < \delta_{w1}^+ \\ 15\text{Pr}^{2/3} - \frac{500}{(\delta_w^+)^2} & \text{for } \delta_{w1}^+ \leq \delta_w^+ < \delta_{w2}^+ \\ \frac{\text{Pr}}{\kappa} \ln(\delta_w^+) + \beta & \text{for } \delta_{w2}^+ \leq \delta_w^+ \end{cases} \quad (4-9)$$

where in turn

$$\begin{aligned} \delta_w^+ &= \frac{\delta_w \rho C_\mu^{1/4} k^{1/2}}{\eta} & \delta_{w1}^+ &= \frac{10}{\text{Pr}^{1/3}} \\ \delta_{w2}^+ &= 10 \sqrt{10 \frac{\kappa}{\text{Pr}_T}} & \text{Pr} &= \frac{C_p \eta}{k} \\ \beta &= 15 \text{Pr}^{2/3} - \frac{\text{Pr}_T}{2\kappa} \left( 1 + \ln \left( 1000 \frac{\kappa}{\text{Pr}_T} \right) \right) \end{aligned} \quad (4-10)$$

The constants  $C_\mu$  and  $\kappa$  equal 0.09 and 0.42, respectively.

#### *Heat Source/Sink*

The Heat source/sink boundary condition is the interior-boundary counterpart of the Heat flux boundary condition discussed above. On interior boundaries, you cannot specify the actual heat flux but only the flux difference between the up and down sides of the boundary, as shown in Equation 4-7. If this difference has positive sign, it acts as a heat source embedded in the boundary, otherwise as a heat sink.

Radiation and a highly conductive layer can contribute to the boundary condition but are disabled by default. Those terms are discussed in separate sections later in this chapter. The remaining condition is

$$-\mathbf{n}_{\text{up}} \cdot (\mathbf{q}_{\text{up}} - \mathbf{q}_{\text{down}}) = q_0 + h(T_{\text{inf}} - T) \quad \text{on } \partial\Omega$$

where  $q_0$  represents a general heat source, for example, an embedded electrical heater. The second term,  $h(T_{\text{inf}} - T)$ , models heat exchanged with a fluid flowing in a network of thin channels embedded in the boundary. An appropriate value for  $h$  can, for example, be found from a local analysis of a unit cell.

This boundary condition is available on interior boundaries and assembly pairs only.

### Continuity

The default setting for interior boundaries is Continuity, which is a special case of the above Heat source/sink condition. In the absence of sources or sinks, that condition becomes  $-\mathbf{n}_{\text{up}} \cdot (\mathbf{q}_{\text{up}} - \mathbf{q}_{\text{down}}) = 0$ . This means that the heat flux in the normal direction is continuous across the boundary.

The temperature is naturally continuous on an internal boundary following the continuity of the finite element field. Therefore, the Continuity boundary condition is identical to the condition that applies between any two neighboring elements in the mesh. In fact, as long as you have not selected **Enable interior boundaries** in the **Boundary Settings** dialog box, the interior boundaries are not in any way different from any other mesh element boundaries, where the Continuity condition effectively applies.

This boundary condition is available on interior boundaries and assembly pairs only.

### Thin Thermally Resistive Layer

On pairs, you can apply the Temperature, Continuity and Heat source/sink boundary conditions in the same way as on any interior boundary. In addition, you can model the jump in temperature caused by a thin layer of a thermally insulating material inserted between the pair boundaries. The Thin thermally resistive layer boundary condition implements Equation 4-8 on the parts of the boundaries where the two sides are in contact with each other.

This boundary condition is available on assembly pairs only.

## BOUNDARY PROPERTIES

Depending on the selected boundary condition type, some of the following boundary properties are available:

QUANTITY	VARIABLE	DESCRIPTION
$q_0$	q	Inward heat flux
$h$	h	Convective heat transfer coefficient
$T_{\text{inf}}$	Tinf	External (bulk) temperature
$T_0$	T0	Prescribed temperature
$k_{\text{res}}$	kres	Layer thermal conductivity
$d_{\text{res}}$	dres	Layer thickness

The remaining quantities on the boundary condition page are grayed out as long as the **Radiation type** is set to **None**. A discussion of the settings for radiative heat transfer

appears in the section “Radiative Heat Transfer” on page 67. If you enable the highly conductive layer feature, additional properties become available on the **Highly Conductive Layer** page. These are described in detail in the section “Highly Conductive Layers” on page 77.

### Using $h$ Coefficients in the Heat Transfer Module

The Heat Transfer Module includes a library of predefined heat transfer coefficients for common flow situations. For a background, discussing why these are necessary and how they are estimated, see “Convective Cooling/Heating and Heat Transfer Coefficients” on page 15. The Heat Transfer Coefficients library is technically a material library similar to, for example, the Liquids and Gases library also included in the product. See “Loading Material Properties from the Library” on page 42 for a general introduction to using and editing material properties and defining functions.

To access the library, open the **Physics>Boundary Settings** dialog box and click the **Load** button (see Figure 4-24). Select to expand **Heat Transfer Coefficients** in the **Materials** tree and select the correlation best describing your convection conditions.

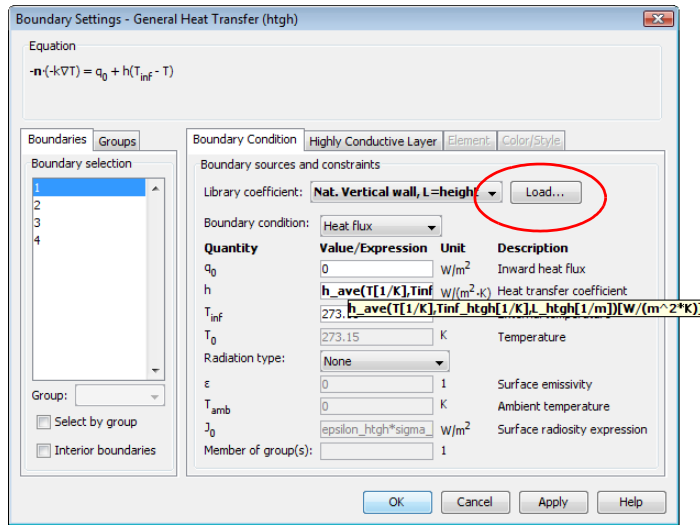


Figure 4-24: The Boundary Settings dialog box with a heat transfer coefficient loaded from the materials/coefficients library.

Once you have chosen the appropriate  $h$  coefficient correlation (convection condition) from the library list, the software updates the edit field for  $h$  with the appropriate

library function call. The following step-by-step describes the necessary settings and details on using this feature.

Follow these steps to use an  $h$  coefficient in a model:

- 1 Open the menu item **Physics>Boundary Settings**.
- 2 Select the appropriate boundaries, then select the **Heat flux** condition.
- 3 Specify the **External temperature** (the bulk temperature of the coolant) by entering a value in the  $T_{inf}$  edit field.
- 4 Click the **Load** button.
- 5 Select **Heat Transfer Coefficients**, which is the  $h$  coefficient library. From the list of  $h$  coefficient correlations, select the appropriate convection condition you want to model (Figure 4-25). Click **OK**.

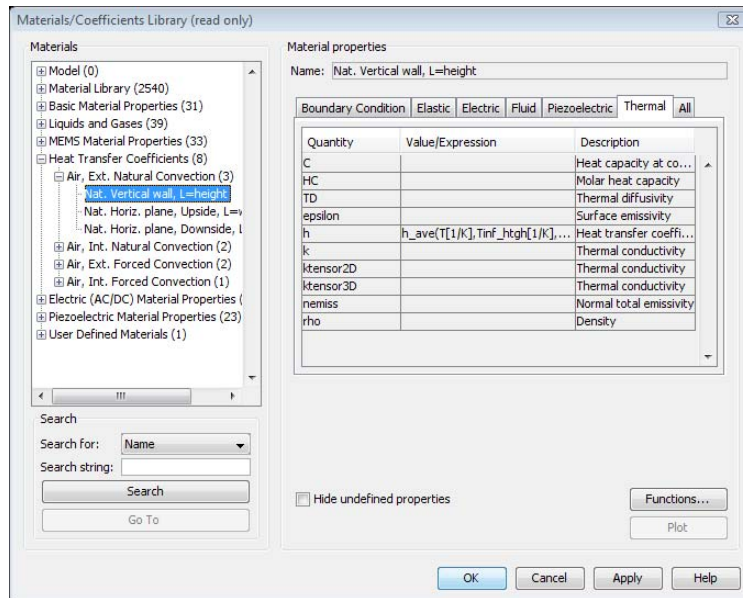


Figure 4-25: The materials/coefficients library for heat transfer coefficients.

A function call now appears in the edit field for **h**. Figure 4-26 shows a typical function call.

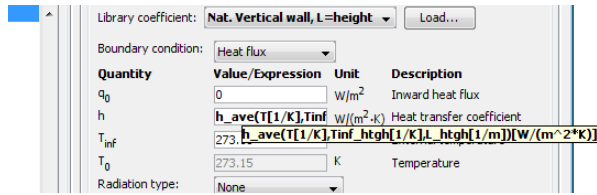


Figure 4-26: The default function call for an  $h$  coefficient loaded from the library (figure shows part of dialog box only).

The default function call generates a solution without any attention, but to get the proper results it is important to set up the correct inputs for the call, the most important being the length scale,  $L$ , of the convection problem.

- 6 Edit the function arguments within the brackets by clicking the edit field for  $h$  and entering your own values or variables.

The parameter  $L\_htgh$  defines the **Convection length scale**. The dimension for this parameter appears in the **Library coefficient** list next to the **Load** button. In some cases there are two length scales. The default parameter,  $L\_htgh$ , has the value 1 m.

If modeling forced convection, you must also replace the default parameter for the **Convection fluid velocity**,  $U\_htgh$ , with the actual velocity in your case. The default has the value 1 m/s.

If modeling external forced convection, use the parameter  $s\_htgh$  to define the **Convection local coordinate**. Its default value is 1 m in 1D,  $s$  m in 2D/2D-axisymmetry, and  $\sqrt{s_1^2 + s_2^2}$  m in 3D. As with the previous parameters, you must replace this one with the value that applies to your problem.

The default settings assume that the temperature variable is  $T$  and that the application mode name is  $htgh$  (all default settings for the General Heat Transfer application mode). This requires changes only if the application mode name differs from  $htgh$ .

- 7 To finish the process, click **OK**. This closes the **Boundary Settings** dialog box.

#### DETAILS ABOUT THE HEAT TRANSFER COEFFICIENTS LIBRARY

The Heat Transfer Coefficients library is a part of the **Materials/Coefficients Library** dialog box in COMSOL Multiphysics, which you open from the **Options** menu (and in read-only mode from by clicking **Load** in the **Boundary Settings** dialog box). It contains  $h$  coefficient functions for the most common geometrical situations. At this time the library is limited to air, the most common cooling fluid.

The library covers all categories of convection (see Figure 2-3 on page 16). COMSOL Multiphysics smoothly switches between library expressions when the flow turns from laminar to turbulent. This means that COMSOL Multiphysics reduces the original eight categories down to four:

- External natural convection
- Internal natural convection
- External forced convection
- Internal forced convection

Once you have selected the appropriate coefficient correlation from the list, the software sets up the  $h$  coefficient edit field of the **Boundary Conditions** dialog box with the necessary library function call. The function uses the following input arguments: the surface temperature,  $T$  (K); the bulk temperature of the coolant,  $T_{\text{inf}}$  (K); the length scale,  $L$  (m); and for forced convection also the bulk coolant velocity,  $U$  (m/s). In some cases there are two length scales. The software uses these parameters to calculate the  $h$  coefficient during the problem solution. The library functions also return the appropriate Jacobian used in the solution process.

If you want to use the  $h$  coefficient functions in some other way in a model, you can call them with a normal function call that has new function arguments. To make this work properly, you must first load the library coefficient. For example, to call the library function depicted in Figure 4-26, use the format

```
mat1_h_ave(arg1, arg2, arg3)
```

where  $\text{arg1}$ ,  $\text{arg2}$ , and  $\text{arg3}$  are the input arguments for  $T$ ,  $T_{\text{inf}}$ , and  $L$ , respectively. The first part of the function name,  $\text{mat1\_}$ , refers to the loaded coefficient number. You can find this name under the **Model** node of the **Materials/Coefficients Library** dialog box.

If you want to specify the coefficient yourself based on an arbitrary Nusselt correlation, you can specify the relationship as an expression in the **Scalar Expressions** dialog box or as a function in the **Functions** dialog box, and then directly enter it in the  $h$  edit field in the **Boundary Settings** dialog box. Another possibility is to create your own library coefficient that you can use also in future modeling sessions. This option is described in the section “Using Your Own Material Data” on page 233 in the *COMSOL Multiphysics User’s Guide*.



## Heat Transfer Coefficients Library Expressions

The following list gives the Heat Transfer Coefficients library expressions presently implemented in the Heat Transfer Module; it contains the original expressions as described in the given references. In COMSOL Multiphysics you specify the material properties as polynomials or other functions of temperature, fitted to the material data according to the fluid materials library in the Heat Transfer Module. In addition, some dimensionless numbers are also represented in the library functions as fitted functions of the temperature, as specified in the table.

### MATERIAL: AIR, 1 ATMOSPHERE

TABLE 4-2: HEAT TRANSFER COEFFICIENTS LIBRARY EXPRESSIONS

TYPE	GEOMETRY	FLOW	$h$ -EXPRESSION	REF.
Natural external	vertical wall	laminar	$h_{\text{ave}} = 0.56F_{\text{lam}}\left(\frac{\Delta T}{L}\right)^{0.25}$	Ref. 2 eq. 9.92
	vertical wall	turbulent ( $Ra_L > 10^9$ )	$h_{\text{ave}} = 0.12F_{\text{turb}}\Delta T^{0.33}$	Ref. 2 eq. 9.92
	horizontal surface, upside	laminar	$h_{\text{ave}} = 0.54F_{\text{lam}}\left(\frac{\Delta T}{L}\right)^{0.25}$	Ref. 2 eq. 9.92
	horizontal surface, upside	turbulent ( $Ra_L > 10^9$ )	$h_{\text{ave}} = 0.14F_{\text{turb}}\Delta T^{0.33}$	Ref. 2 eq. 9.92
	horizontal surface, downside	laminar*	$h_{\text{ave}} = 0.25F_{\text{lam}}\left(\frac{\Delta T}{L}\right)^{0.25}$	Ref. 2 eq. 9.92
Natural internal	chimney flow, parallel plates	laminar/ turbulent	$h_{\text{ave}} = \frac{kRa}{192L}$	Ref. 3 eq. 7.96
	chimney flow, tube	laminar/ turbulent	$h_{\text{ave}} = \frac{kRa}{128L}$	Ref. 3 eq. 7.96
Forced external	plate	laminar	$h_{\text{ave}} = \left(\frac{k}{L}\right) \frac{0.928Pr^{0.33}Re^{0.5}}{\left(1 + \left(\frac{0.0207}{Pr}\right)^{0.67}\right)^{0.25}}$	Ref. 3 eq. 5.83
	plate	turbulent ( $Re_L \geq 5 \cdot 10^5$ )	$h_{\text{ave}} = \left(\frac{k}{L}\right)(0.037Pr^{0.33}Re^{0.8} - 23550)$	Ref. 3 eq. 5.134
	plate	laminar	$h_{\text{loc}} = \left(\frac{k}{L}\right)0.332Pr^{0.33}Re^{0.5}$	Ref. 3 eq. 5.79

TABLE 4-2: HEAT TRANSFER COEFFICIENTS LIBRARY EXPRESSIONS

TYPE	GEOMETRY	FLOW	$h$ -EXPRESSION	REF.
	plate	turbulent ( $Re_y \geq 5 \cdot 10^5$ )	$h_{loc} = \left(\frac{k}{L}\right) 0.0296 Pr^{0.33} Re^{0.8}$	Ref. 3 eq. 5.131
Forced internal	isothermal tube	laminar	$h_{ave} = \frac{3.66k}{L}$	Ref. 3 table 6.2
	isothermal tube	turbulent ( $Re_D > 2300$ )	$h_{ave} = \left(\frac{k}{L}\right) 0.023 Pr^{0.33} Re^{0.8}$	Ref. 3 eq. 6.90

\* Turbulent flow was never observed.

In this table,  $F_{lam} = k \left(\frac{Ra}{L^3 \Delta T}\right)^{1/4}$  and  $F_{turb} = \frac{k}{L} \left(\frac{Ra}{\Delta T}\right)^{1/3}$ .

For  $F_{lam}$ ,  $F_{turb}$ ,  $Ra$ , and  $Pr$ , the table uses function fits based on the material properties at 1 atmosphere. For the material **Air** these are

$$F_{lam} = 6.3126 - 1.4322 \log T$$

$$F_{turb} = 121.9 - 69.518 \log T + 10.255 (\log T)^2$$

$$Ra = \Delta T L^3 10^{(19.058 - 4.4449 \log T)}$$

$$Pr = 2.8469 - 1.3494 \log T + 0.1949 (\log T)^2$$

where  $\log$  represents  $\log_{10}$ , and  $L$  refers to the length scale value in meters and  $T$  to the temperature value in kelvin.

### References

1. D. Lacasse, É. Turgeon, and D. Pelletier, "On the judicious use of the  $k$ - $\epsilon$  model, wall functions and adaptivity," *Int. J. Therm. Sci.*, vol. 43, pp. 925–938, 2004.
2. J.M. Coulson and J.F. Richardson, *Chemical Engineering*, vol. 1, 4th ed., Pergamon Press, 1990.
3. A. Bejan, *Heat Transfer*, Wiley, 1993.

# Radiative Heat Transfer

The General Heat Transfer application mode can account for heat transfer by radiation, in addition to convection and conduction. Radiation is treated as a process that transfers energy directly between boundaries and which therefore contributes to the boundary conditions rather than to the heat equation itself. This section describes the additional boundary equation for the radiation balance, the available view factor evaluation methods, and the boundary properties and settings associated with radiation.

You can read a more detailed explanation and background to the radiation equations under “Radiative Heat Transfer” on page 10 in the chapter on heat transfer theory.

---

**Note:** In the General Heat Transfer application mode COMSOL Multiphysics works under the assumption that the media in subdomains does not participate in the radiation process. Note also that you must work with absolute (thermodynamical) temperature units. See “Using Units” on page 188 of the *COMSOL Multiphysics User’s Guide*.

---

## *The Radiosity Method*

---

The radiation interacts with convective and conductive heat transfer through the source term  $q_r$  in the Heat flux and Heat source/sink boundary conditions, Equation 4-11 and Equation 4-12. By definition, this source must be the difference between incident radiation and radiation leaving the surface. According to Equation 2-10 it is given by

$$q_r = \varepsilon(G - \sigma T^4) \quad (4-11)$$

where

- $\varepsilon$  is the surface emissivity, a dimensionless number in the range  $0 \leq \varepsilon \leq 1$ .
- $G$  is the incoming radiative heat flux, or *irradiation* ( $\text{W}/\text{m}^2$ )
- $\sigma$  is the Stefan-Boltzmann constant by default equal to  $5.67 \cdot 10^{-8} \text{ W}/(\text{m}^2 \cdot \text{T}^4)$ .

The irradiation,  $G$ , at a point can in general be written as a sum according to Equation 2-13:

$$G = G_m + F_{\text{amb}} \sigma T_{\text{amb}}^4$$

where

- $G_m$  is the mutual irradiation, coming from other boundaries in the model ( $\text{W}/\text{m}^2$ ).
- $F_{\text{amb}}$  is an ambient view factor whose value is equal to the fraction of the field of view that is not covered by other boundaries. Therefore, by definition,  $0 \leq F_{\text{amb}} \leq 1$  must hold at all points.
- $T_{\text{amb}}$  is the assumed far-away temperature in the directions included in  $F_{\text{amb}}$ .

The Heat Transfer Module allows two different settings: *surface-to-ambient radiation* and *surface-to-surface radiation*:

- Surface-to-ambient radiation is appropriate for convex exterior surfaces and assumes from the outset that  $G_m = 0$  and  $F_{\text{amb}} = 1$ . Inserting this into Equation 4-11, the total radiative heat flux simplifies to

$$q_r = \varepsilon \sigma (T_{\text{amb}}^4 - T^4) \quad (4-12)$$

- Surface-to-surface radiation, on the other hand, requires accurate evaluation of the mutual irradiation,  $G_m$ . The incident radiation at one point on the boundary is a function of the exiting radiation, or *radiosity*,  $J$  ( $\text{W}/\text{m}^2$ ), at every other point in view. The radiosity, in turn, is a function of  $G_m$ , which leads to an implicit radiation balance. According to Equation 2-20:

$$J = (1 - \varepsilon) \{ G_m(J) + F_{\text{amb}} \sigma T_{\text{amb}}^4 \} + \varepsilon \sigma T^4 \quad (4-13)$$

The General Heat Transfer application mode treats  $J$  as an additional dependent variable. When you enable surface-to-surface radiation on a boundary, you also introduce shape functions for  $J$  and add the above equation to be solved together with the heat equation, Equation 4-1.

### *View Factor Evaluation*

---

The strategy for evaluating *view factors* is central to any radiation simulation. Loosely speaking, a view factor is a measure of how much influence the radiosity at a given part of the boundary has on the irradiation at some other part.

The quantities  $G_m$  and  $F_{\text{amb}}$  in Equation 4-13 are not strictly view factors in the traditional sense.  $F_{\text{amb}}$  is the view factor of the ambient portion of the field of view, which is considered to be a single boundary with constant radiosity

$$J_{\text{amb}} = \sigma T_{\text{amb}}^4$$

$G_m$ , on the other hand, is the integral over all visible points of a differential view factor times the radiosity of the corresponding source point (see “Surface-to-Surface Radiation” on page 12). In the discrete model, you can think of it as a product of a view factor matrix and a radiosity vector. This is, however, not necessarily the way the calculation is performed.

A separate evaluation is performed for each unique point where  $G_m$  or  $F_{\text{amb}}$  is requested, typically for each quadrature point during solution. Differential view factors are normally computed only once, the first time they are needed, and then stored in memory until next time the model definition or the mesh is changed. This behavior is usually desirable but can be turned off using the **Cache view factors** setting in the **Application Mode Properties** dialog box.

The Heat Transfer Module supports two different surface-to-surface radiation methods, which you can select in the **Application Mode Properties** dialog box:

**Direct area integration:** COMSOL Multiphysics evaluates the integrals in Equation 2-14 and Equation 2-15 directly, without considering which face elements are obstructed by others. This means that shadowing effects (that is, surface elements being obstructed in nonconvex cases) are not taken into account. Elements facing away from each other are, however, excluded from the integrals.

Direct area integration is fast and accurate for simple geometries with no shadowing, or where the shadowing can be handled by manually assigning boundaries to different groups (see “Grouping Boundaries” on page 74). Note that if shadowing is ignored, global energy is not conserved. You can control the accuracy by specifying an integration order. Sharp angles and small gaps between surfaces may require a higher integration order for accuracy but also more time to evaluate the irradiation.

**Hemicube:** The more sophisticated and general hemicube method uses a  $z$ -buffered projection on the sides of a hemicube (with generalizations to 2D and 1D) to account for shadowing effects. You can think of the method as rendering digital images of the geometry in five different directions (in 3D; in 2D only three directions are needed), and counting the pixels in each mesh element to evaluate its view factor.

Hemicube is the default method in the General Heat Transfer application mode. You can influence its accuracy by setting the resolution of the virtual snapshots. The number of  $z$ -buffer pixels on each side of the 3D hemicube equals the specified resolution squared. Thus the time required to evaluate the irradiation increases

quadratically with resolution. In 2D, the number of  $z$ -buffer pixels is proportional to the resolution property, and thus the time is, as well.

---

**Note:** View factors are always calculated directly from the mesh, which is a polygonal, flat-faceted representation of the geometry. To improve the accuracy of the radiative heat transfer simulation, you must therefore refine the mesh rather than raise the element order.

---

### **RADIATION IN AXISYMMETRIC GEOMETRIES**

For an axisymmetric geometry,  $G_m$  and  $F_{amb}$  must be evaluated in a corresponding 3D geometry obtained by revolving the 2D boundaries about the axis. COMSOL Multiphysics creates this virtual 3D geometry by revolving the 2D boundary mesh into a 3D mesh. You control the resolution in the azimuthal direction by setting the number of azimuthal sectors, which is the same as the number of elements to a full revolution. Try to balance this number, accessible from the **Application Mode Properties** dialog box, against the mesh resolution in the  $rz$ -plane.

You can select between the hemicube and the direct area integration methods also in axial symmetry. Their settings work the same way as in 3D. Note that while  $G_m$  and  $F_{amb}$  are in fact evaluated in a full 3D, the number of points where they are requested is limited to the quadrature points on the boundary of a 2D geometry. The savings compared to a full 3D simulation are therefore substantial despite the full 3D view factor code being used.

## Boundary Settings

To enable radiative heat transfer, go to the **Physics** menu and select **Boundary Settings**. In the resulting dialog box go to the **Boundary Condition** page (Figure 4-27).

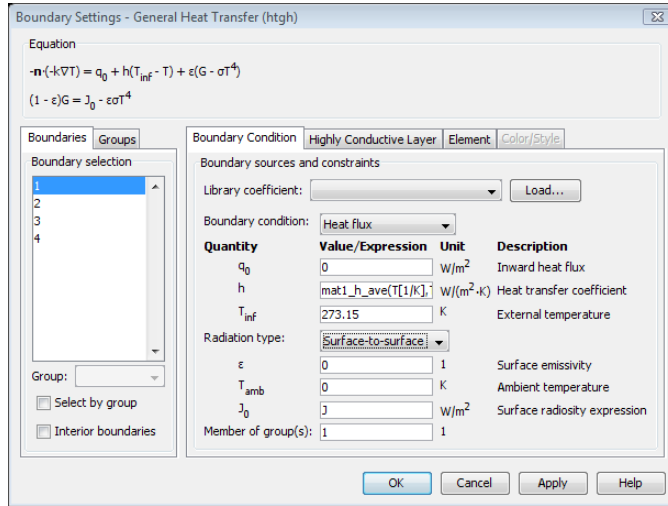


Figure 4-27: The Boundary Condition page in the Boundary Settings dialog box.

The following **Boundary condition** types (see “Boundary Condition Types” on page 56) are compatible with radiation:

- **Heat flux**
- **Heat source/sink**
- **Temperature**

After selecting one of the above boundary conditions, you can enable radiation by setting **Radiation type** to either **Surface-to-ambient** or **Surface-to-surface**. This enables edit fields for the following radiation properties:

QUANTITY	VARIABLE	DESCRIPTION
$\epsilon$	epsilon	Surface emissivity
$T_{amb}$	$T_{amb}$	Ambient temperature
$J_0$	$J_0$	Surface radiosity expression
Member of group(s)	-	-

**Surface emissivity:** The surface emissivity is a property of the material surface that depends both on the material itself and the structure of the surface. Set  $\epsilon$  to a number between 0 and 1, where 0 represents diffuse mirror and 1 is appropriate for a perfect black body. The proper value for a physical material lies somewhere in-between and can be found from tables or measurements.

**Ambient temperature:** Set  $T_{\text{amb}}$  to the far-away temperature in directions where no other boundaries obstruct the view. Inside a closed cavity, the ambient view factor,  $F_{\text{amb}}$ , is theoretically zero and the value of  $T_{\text{amb}}$  therefore should not matter. It is, however, good practice to set  $T_{\text{amb}}$  to  $T$  in such cases because that minimizes errors introduced by the finite resolution of the view factor evaluation.

**Surface radiosity expression:** The surface radiosity expression  $J_0$  specifies how the radiosity of a boundary is evaluated when that boundary is visible in the calculation of the irradiation onto another boundary in the model.

Note that  $J_0$  does not directly affect the boundary condition on the boundary where it is specified, but rather how that boundary affects others through radiation. Therefore, a boundary can be specified as accepting only surface-to-ambient radiation but still give a contribution to surface-to-surface radiation on other boundaries.

**Member of group(s):** The **Member of group(s)** field lets you specify roughly which boundaries are visible to each other. This can speed up the radiation calculations in many cases. Read more under “Grouping Boundaries” on page 74.

#### **BOUNDARY CONDITION—RADIATION TYPE COMBINATIONS**

The combined selection of **Boundary condition** and **Radiation type** affects the definition of the radiative heat flux  $q_r$  in Equation 4-13 and the default values of  $J_0$  and **Member of groups**. In addition, selecting **Surface-to-surface** radiation enables Equation 4-13 and lets you change the shape function for the radiosity  $J$  on the **Element** page. The following combinations are possible:

##### *Heat Flux with Surface-to-Ambient Radiation*

The radiative flux is defined according to Equation 4-12 as

$$q_r = \epsilon\sigma(T_{\text{amb}}^A - T^A)$$

By default, the radiosity of the boundary is set to  $J_0 = \epsilon\sigma T^A$  and no equation for the radiosity is solved. If the **Member of group(s)** field is left empty, the boundary does not take part in surface-to-surface radiation. It can, however, be assigned to one or more



groups, in which case it contributes the radiosity specified in  $J_0$  to the group but does not receive any irradiation except for the above  $q_r$ .

#### *Heat Flux with Surface-to-Surface Radiation*

The radiative flux is defined according to Equation 4-11 as

$$q_r = \varepsilon(G - \sigma T^4)$$

where the evaluation of  $G$  requires the solution of Equation 4-13. The default value of  $J_0$  is therefore  $J$ , which also appears as a dependent variable on the **Element** page. This setting applies to nearly all kinds of radiation problems. Only in rare cases would you use values for  $J_0$  other than  $J$ . By default all boundaries taking part in surface-to-surface radiation are part of the same group, number 1.

Note that surface-to-surface radiation can take place only between boundaries that border on exactly one *opaque* subdomain. Therefore, make sure that the adjacent subdomain has **Opacity** set to **Opaque** in the **Subdomain Settings** dialog box (see “Subdomain Settings” on page 30).

#### *Temperature with Surface-to-Ambient Radiation*

You can combine the Dirichlet boundary condition  $T = T_0$  with radiation. In the majority of situations this does not affect the conditions on the boundary itself, because the temperature there has been specified once and for all. Instead, switching on surface-to-ambient radiation for a Temperature boundary allows the boundary to act as a radiation source for other boundaries taking part in surface-to-surface radiation. For this to happen, you must assign the boundary to a radiation group because the **Member of group(s)** field is empty by default.

There is, however, a case when adding radiation terms changes the meaning of the Temperature boundary condition itself. If you set  $T_0 = T_2$ , where  $T_2$  is the temperature variable from another application mode, you normally get continuity both in the combined temperature field  $[T, T_2]$  and in the normal heat flux over the boundary. Radiation terms introduced on the boundary in that case act like a source or sink on an interior boundary and cause a heat flux discontinuity. Compare to Equation 4-7 and the discussion about the Heat source/sink boundary condition on page 59.

#### *Temperature with Surface-to-Surface Radiation*

A boundary with a Temperature boundary condition can also take part in surface-to-surface radiation. Just like the case with Heat flux boundary condition and surface-to-surface radiation, an additional equation must be solved for the radiosity,  $J$ . As in the case above, the radiation terms do not affect the conditions at the boundary

itself unless the Temperature condition is used to enforce continuity between two different temperature fields.

#### *Heat Source/Sink with Surface-to-Ambient Radiation*

Using surface-to-ambient radiation together with the Heat source/sink boundary condition on an interior boundary is similar to using the combination of Surface-to-ambient and Heat flux on an exterior boundary, see “Heat Flux with Surface-to-Ambient Radiation” on page 72 above.

Note that as the equation is written, the radiation is assumed to affect only one side of the boundary. Therefore, this condition is directly applicable only to a boundary between an opaque and a transparent domain. When there are transparent domains on both sides of the boundary, radiation will go both ways. If the ambient temperature is the same on both sides, you can model this bidirectional radiation just by summing the emissivities of the two surfaces. Otherwise, you can use the source term  $q_0$  to manually implement surface-to-ambient radiation in the more general case.

#### *Heat Source/Sink with Surface-to-Surface Radiation*

This combination is the interior-boundary counterpart of “Heat Flux with Surface-to-Surface Radiation” on page 73. Note that surface-to-surface radiation can take place only between boundaries that border on exactly one *opaque* subdomain. The radiation takes place in the direction away from the opaque domain, possibly into a neighboring domain marked as *transparent*. The void surrounding the geometry and any inactivated domains are considered to be transparent in this sense.

### *Grouping Boundaries*

---

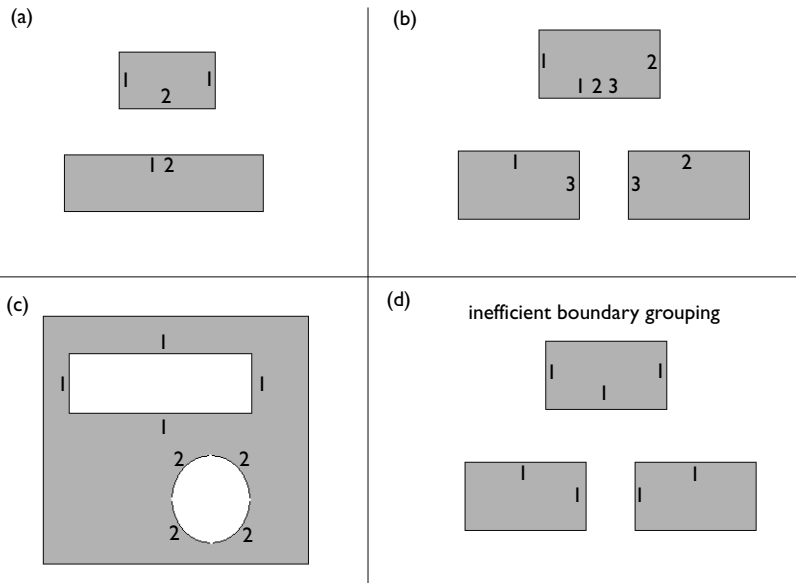
For radiation problems you can apply a boundary grouping to save computational time. You specify it in the **Boundary Settings** dialog box by going to the **Boundary Condition** page and then making an appropriate entry in the **Member of group(s)** edit field.

A boundary grouping specifies which boundaries in a radiation problem have a chance to see one another, and they must be members of the same group. For each boundary that has its radiation type set to surface-to-ambient or surface-to-surface, use the **Member of group(s)** edit field to specify to which groups the boundary belongs.

---

**Note:** Be careful when grouping boundaries in axially symmetric geometries. The grouping cannot be based on which boundaries have a free view toward each other in the 2D geometry. Instead, you must consider the full 3D geometry, obtained by revolving the model geometry about the  $z$  axis, when defining groups. For example, parallel vertical boundaries must typically belong to the same group in 2D axisymmetric models, but to different groups in a planar model using the same 2D geometry.

---



*Figure 4-28: Examples of boundary grouping.*

Figure 4-28 shows four examples of possible boundary groupings. On boundaries that have no number, the user has set the surface-to-ambient radiation type. These boundaries do not irradiate other boundaries, neither do other boundaries irradiate them. On boundaries that have one or several group numbers, the user has set the surface-to-surface radiation type. The numbers on each boundary specify in which groups the boundary belongs.

On boundaries where the user has set the surface-to-surface radiation type, the default setting for **Member of group(s)** is 1. To specify that a boundary is a member of more

than one group, you can enter a space-separated list of integers in the edit field, for example, 1 2 3.

To obtain optimal computational performance, it is good practice to specify as many groups as possible as opposed to specifying few but large groups. For example, in Figure 4-28, case (b) is more efficient than case (d).

On boundaries where the user has set surface-to-ambient radiation, the default for **Member of group(s)** is an empty edit field, which denotes that the boundary does not interact with other boundaries through radiation. If you specify one or more group numbers on these boundaries, the radiation that leaves the boundary irradiates the surrounding boundaries, but it receives no radiation from the surrounding boundaries.

Using the surface-to-ambient radiation type with boundary grouping can be efficient if the boundary acts as a pure emitter that irradiates an object because this setup does not require COMSOL Multiphysics to solve the equations for the radiosity on the emitting boundary.

### *Guidelines for Solving Surface-to-Surface Radiation Problems*

---

The following guidelines are helpful when selecting solver settings for models that involve surface-to-surface radiation:

- Surface-to-surface radiation makes the Jacobian matrix of the discrete model partly filled as opposed to the usual sparse matrix. The additional nonzero elements in the matrix appear in the rows and columns corresponding to the radiosity degrees of freedom. It is therefore common practice to keep the element order of the radiosity variable,  $J$ , low. By default, linear Lagrange elements are used irrespective of the shape function order specified for the temperature. When you need to increase the resolution of your temperature field, it might be worth considering raising the order of the temperature elements instead of refining the mesh.
- The **Assembly blocksize** parameter (found in the **Solver Parameters** dialog box on the **Advanced** page) can have a major influence on memory usage during the assembly of problems where surface-to-surface radiation is enabled. You might consider a blocksize as small as 100. Using a smaller blocksize also leads to more frequent updates of the progress bar.

The section “Building and Solving a Radiation Model” on page 131 gives an example of how to build and solve a model involving both conductive and radiative heat transfer.

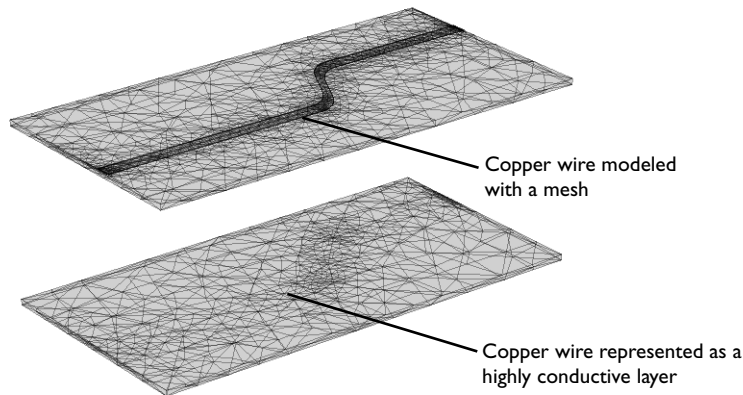
# Highly Conductive Layers

The highly conductive layer feature is efficient for modeling heat transfer in thin layers without the need to create a fine mesh for them. The material in the thin layer should be a good thermal conductor. A good example is a copper trace on a printed circuit board, where the traces are good thermal conductors compared to the board's substrate material. More generally you can apply the highly conductive layer feature in a part of a geometry with the following properties:

- The part is a thin layer compared to the thickness of the adjacent geometry
- The part is a good thermal conductor compared to the adjacent geometry

Because the layer is very thin and has a high thermal conductivity, you can assume that no variations in temperature and in-plane heat flux exist along the layer's thickness. Furthermore, you can think of the difference in heat flux in the layer's normal direction between its upper and lower face as a heat source or sink that is smeared out along the layer thickness.

The Heat Transfer Module supports heat transfer in highly conductive layers in 2D and 3D as well as 2D axisymmetry. A significant benefit is that you can represent a layer as a boundary instead of a subdomain. Doing so simplifies the geometry and significantly reduces the required number of mesh elements. Figure 4-29 shows an example where a highly conductive layer reduces the mesh density significantly.



*Figure 4-29: Modeling a copper wire as a subdomain (top) requires a denser mesh compared to modeling it as a boundary with a highly conductive layer (bottom).*

*PDE Formulation—Highly Conductive Layers*

---

To describe heat transfer in highly conductive layers, the Heat Transfer Module uses a variant of the heat equation that describes the in-plane heat flux in the layer (see also “Thin Conductive Shell” on page 153)

$$d_s \rho_s C_{p,s} \frac{\partial T}{\partial t} + \nabla_t \cdot (-d_s \underline{k}_s \nabla_t T) = q_{\partial\Omega} - q_\Omega = -q_s. \quad (4-14)$$

Here the operator  $\nabla_t$  denotes the *del* or *nabla* operator projected onto the plane of the highly conductive layer. The properties in the equation are:

- $\rho_s$  is the layer density (kg/m<sup>3</sup>).
- $C_{p,s}$  is the layer heat capacity (J/(kg·K)).
- $\underline{k}_s$  is the layer thermal conductivity tensor (W/(m·K)).
- $d_s$  is the layer thickness (m).
- $q_{\partial\Omega}$  is the heat flux from the surroundings into the layer (W/m<sup>2</sup>).
- $q_\Omega$  is the heat flux from the layer into the subdomain (W/m<sup>2</sup>).
- $q_s$  is the net outflux of heat through the top and bottom faces of the layer (W/m<sup>2</sup>).

The rightmost variable,  $q_s$ , can be recognized as one of the source terms in Equation 4-6 page 54. With the above boundary equation inserted, the general heat flux boundary condition becomes

$$-\mathbf{n} \cdot \mathbf{q} = q_0 + q_r + h(T_{\text{inf}} - T) - d_s \rho_s C_{p,s} \frac{\partial T}{\partial t} - \nabla_t \cdot (-d_s \underline{k}_s \nabla_t T) \quad \text{on } \partial\Omega$$

## Boundary Settings—Highly Conductive Layer

To create a highly conductive layer, go to the **Physics** menu and select **Boundary Settings**. In the resulting dialog box go to the **Highly Conductive Layer** page (Figure 4-30).

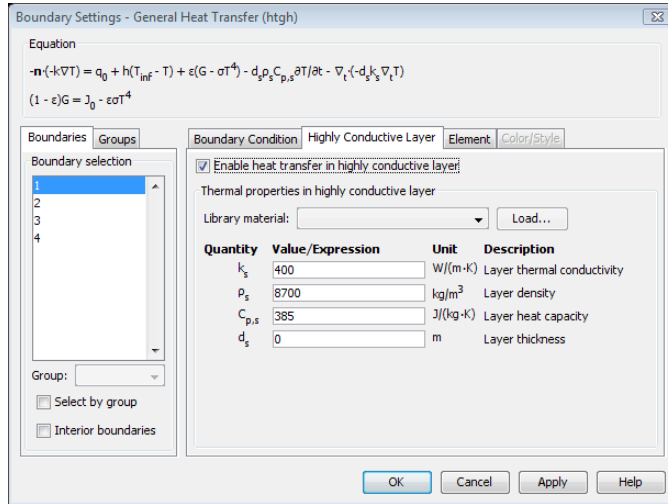


Figure 4-30: The Highly Conductive Layer page in the Boundary Settings dialog box.

Select the **Enable heat transfer in highly conductive layer** check box at the top of the page to activate heat transfer in the highly conductive layer. The quantities associated with a highly conductive layer are:

QUANTITY	VARIABLE	DESCRIPTION
$\rho$	rhos	Layer density
$C_p$	Cps	Layer heat capacity
$k$	ks	Layer thermal conductivity
$\vec{k}_s$		Layer thermal conductivity tensor
$k_{s,ij}$	ksxixj	Layer thermal conductivity tensor, $x_i x_j$ component
$d_s$	ds	Layer thickness

**Layer density**  $\rho$  specifies the material's density. Note that you enter density as mass per unit volume and not as a surface density, or mass per area.

**Layer heat capacity**  $C_{p,s}$  describes the amount of heat energy required to produce a unit temperature change in a unit mass of the layer material at constant pressure.

**Layer thermal conductivity**  $k$  describes the relationship between the heat flux vector  $\mathbf{q}$  and the tangential temperature gradient  $\nabla_t T$  as in

$$\mathbf{q} = -d_s k \nabla_t T.$$

When modeling anisotropic thermal conductivity properties, note that the Heat Transfer Module uses a 3-by-3 matrix in 3D and a 2-by-2 matrix in 2D to represent the thermal conductivity tensor. Enter the elements in the global coordinate system.

**Layer thickness**  $d_s$  is zero by default.

---

**Note:** If the thickness is zero, the highly conductive layer does not take effect.

---

See the section “Building and Solving a Model with Highly Conductive Layers” on page 142 for an example on how to build a model involving a highly conductive layer.

### *Reference*

---

I. F.P. Incropera and D.P. DeWitt, *Fundamentals of Heat and Mass Transfer*, 4th ed., John Wiley & Sons, 1996.



# Out-of-Plane Heat Transfer

When the object you intend to model in COMSOL Multiphysics is thin or slender enough along one of its geometry dimensions, there is usually only a small variation in temperature along the object's thickness or cross section. For such objects, it is efficient to reduce the model geometry to 2D or even 1D and use the out-of-plane heat transfer mechanism. Figure 4-31 shows two examples of likely situations where you can apply this type of geometry reduction.

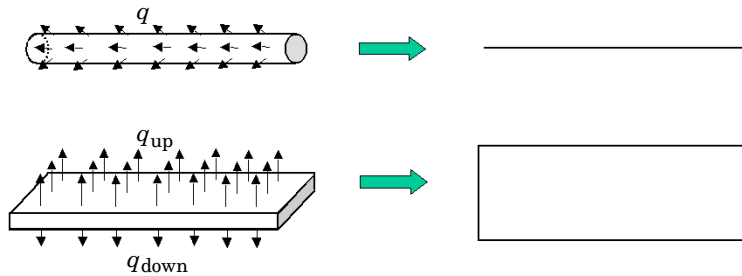


Figure 4-31: Geometry reduction from 3D to 1D (top) and from 3D to 2D (bottom).

The reduced geometry does not include all the boundaries of the original 3D geometry. For example, the reduced geometry does not represent the upside and downside surfaces of the plate in Figure 4-31 as boundaries. Instead, heat transfer through these boundaries appears as sources or sinks in the thickness-integrated version of the heat equation used when out-of-plane heat transfer is active.

## *PDE Formulation—Out-of-Plane Heat Transfer*

The following equation shows the heat equation for a 2D heat transfer problem that includes conduction and out-of-plane heat transfer:

$$\delta_{ts} d_z \rho C_p \frac{\partial T}{\partial t} + \nabla \cdot (-d_z \underline{k} \nabla T) = d_z Q + d_z q_s T + h_u (T_{ext, u} - T) + h_d (T_{ext, d} - T) + \varepsilon_u \sigma (T_{amb, u}^4 - T^4) + \varepsilon_d \sigma (T_{amb, d}^4 - T^4)$$

Mathematically this is the heat equation integrated along the thickness of the full 3D geometry where  $d_z$  (m) represents the thickness.

The first two out-of-plane terms

$$h_u(T_{\text{ext, u}} - T) + h_d(T_{\text{ext, d}} - T)$$

represent the convective heat flux through the upside and downside surfaces of the full 3D geometry. Here:

- $h_u$  is the heat transfer coefficient for the upside ( $\text{W}/(\text{m}^2 \cdot \text{K})$ ).
- $h_d$  is the heat transfer coefficient for the downside ( $\text{W}/(\text{m}^2 \cdot \text{K})$ ).
- $T_{\text{ext, u}}$  is the external temperature for the upside (K).
- $T_{\text{ext, d}}$  is the external temperature for the downside (K).

In addition, the final two out-of-plane terms

$$\varepsilon_u \sigma (T_{\text{amb, u}}^4 - T^4) + \varepsilon_d \sigma (T_{\text{amb, d}}^4 - T^4)$$

represent the heat flux due to surface-to-ambient radiation where

- $\varepsilon_u$  is the surface emissivity for the upside (dimensionless).
- $\varepsilon_d$  is the surface emissivity for the downside (dimensionless).
- $\sigma$  is the Stefan-Boltzmann constant ( $\text{W}/(\text{m}^2 \cdot \text{K}^4)$ ).
- $T_{\text{amb, u}}$  is the ambient temperature for the upside (K)
- $T_{\text{amb, d}}$  is the ambient temperature for the downside (K).

In 1D, the corresponding subdomain equation is:

$$\begin{aligned} \delta_{\text{ts}} A_c \rho C_p \frac{\partial T}{\partial t} + \nabla \cdot (-A_c k \nabla T) \\ = A_c (Q + q_s T) + P_c h_z (T_{\text{ext, z}} - T) + P_c \varepsilon_z \sigma (T_{\text{amb, z}}^4 - T^4) \end{aligned}$$

Mathematically this is the heat equation integrated over the cross-sectional area of the full 3D geometry, where

- $A_c$  is the cross-sectional area ( $\text{m}^2$ ).
- $P_c$  is the cross-sectional perimeter (m).
- $h_z$  is the heat transfer coefficient ( $\text{W}/(\text{m}^2 \cdot \text{K})$ ).
- $T_{\text{ext, z}}$  is the external temperature (K).
- $\varepsilon_z$  is the surface emissivity (dimensionless)

- $\sigma$  is the Stefan-Boltzmann constant ( $W/(m^2 \cdot K^4)$ )
- $T_{amb,z}$  is the ambient temperature (K).

### Subdomain Settings—Out-of-Plane Heat Transfer

To enable out-of-plane heat transfer, go to the **Physics** menu and choose **Properties**. In the **Application Mode Properties** dialog box, locate the **Out-of-plane heat transfer** option and select **Enabled** from the list.

To enter properties for out-of-plane heat transfer, first choose **Physics>Subdomain Settings**. In the resulting dialog box go to the **Out-of-Plane** page (Figure 4-32).

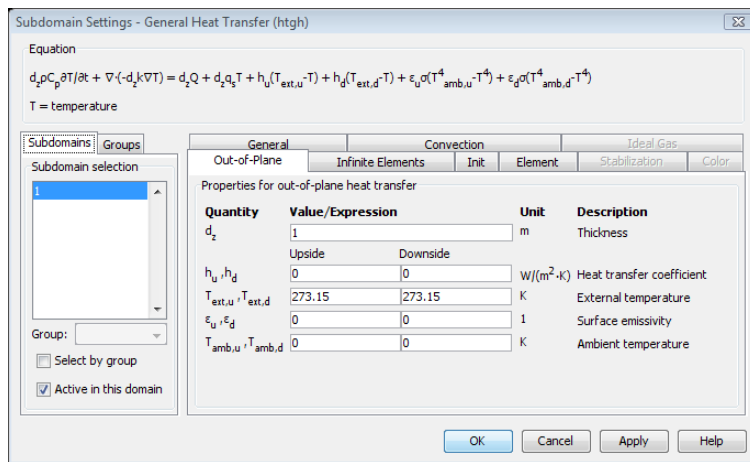


Figure 4-32: The Out-of-Plane page in the Subdomain Settings dialog box (in 2D).

### SUBDOMAIN QUANTITIES—2D

In 2D the subdomain quantities are:

QUANTITY	VARIABLE	DESCRIPTION
$h_u$	hu	Heat transfer coefficient, upside
$h_d$	hd	Heat transfer coefficient, downside
$T_{ext,u}$	Textu	External temperature, upside
$T_{ext,d}$	Textd	External temperature, downside
$\epsilon_u$	epsilonu	Surface emissivity, upside
$\epsilon_d$	epsilond	Surface emissivity, downside
$T_{amb,u}$	Tambu	Ambient temperature, upside

QUANTITY	VARIABLE	DESCRIPTION
$T_{amb,d}$	Tambd	Ambient temperature, downside
$d_z$	dz	Thickness

**Heat transfer coefficient, upside/downside**  $h_u$  is the heat transfer coefficient for the upside surface; use  $h_d$  for the downside surface.

**External temperature, upside/downside**  $T_{ext,u}$  is the external bulk temperature on the upside ( $T_{ext,d}$  on the downside) that the heat transfer coefficient uses as a reference to define the convective heat flux through the surface.

**Surface emissivity, upside/downside** The surface emissivity of the upside surface,  $\epsilon_u$ , ( $\epsilon_d$  for the downside surface) is a dimensionless number between 0 and 1.

**Ambient temperature, upside/downside** The upside surface sees the temperature of its ambient surrounding as  $T_{amb,u}$ , and the downside surface sees its ambient as  $T_{amb,d}$ .

**Thickness**  $d_z$  is the thickness of the full 3D object that is reduced to a 2D model using out-of-plane heat transfer.

#### SUBDOMAIN QUANTITIES—1D

In 1D the subdomain quantities are:

QUANTITY	VARIABLE	DESCRIPTION
$h_z$	hz	Heat transfer coefficient
$T_{ext,z}$	Textz	External temperature
$\epsilon_z$	epsilonz	Surface emissivity
$T_{amb,z}$	Tambz	Ambient temperature
$A_c$	Ac	Cross-sectional area
$P_c$	Pc	Cross-sectional perimeter

**Heat transfer coefficient**  $h_z$  is the heat transfer coefficient for the surface. When using SI units, enter the value in  $W/(m^2 \cdot K)$ .

**External temperature**  $T_{ext,z}$  is the external bulk temperature that the heat transfer coefficient uses as a reference to define the convective heat flux through the surface.

**Surface emissivity**  $\epsilon_z$  is a dimensionless number between 0 and 1.

**Ambient temperature** The surface sees a the surrounding ambient at a temperature of  $T_{\text{amb},z}$ .

**Cross-sectional area**  $A_c$  is the area of the cross section of the full 3D object that out-of-plane heat transfer reduces to a 1D model.

**Cross-sectional perimeter**  $P_c$  is the perimeter of the cross section of the full 3D object that out-of-plane heat transfer reduces to a 1D model.

See the section “Out-of-Plane Heat Transfer” on page 81 for an example of how to build a model involving both conductive and out-of-plane heat transfer.

See the section “Building and Solving an Out-of-Plane Heat Transfer Model” on page 146 for an example on how to build a model involving out-of-plane heat transfer.

# Building and Solving General Heat Transfer Models

The following sections describe how to build and solve heat transfer problems using the General Heat Transfer application mode.

The discussion starts with an example on conductive heat transfer and further examples add other heat transfer mechanisms. The following list points out in which section you can read about how to build models with different heat transfer mechanisms:

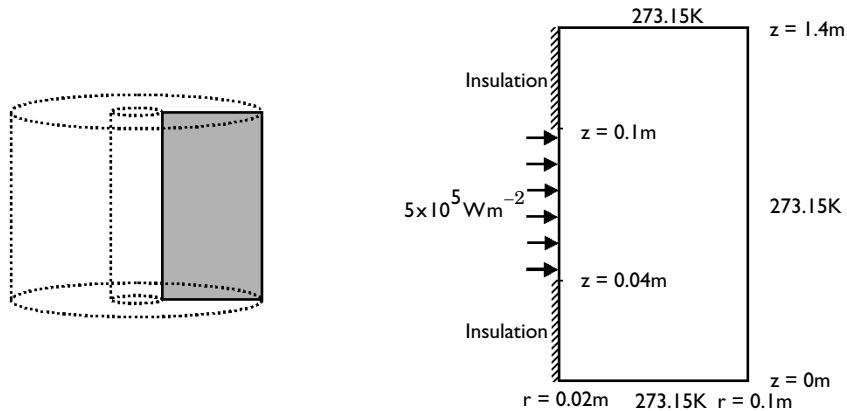
- Conduction—see “Building and Solving a Conduction Model” on page 87.
- Conduction and convection—see “Building and Solving a Natural Convection Model—Laminar Flow” on page 109.
- Conduction and radiation—see “Building and Solving a Radiation Model” on page 131.
- Conduction and out-of-plane heat transfer—see “Out-of-Plane Heat Transfer” on page 81.
- Conduction with heat transfer in highly conductive layers—see “Building and Solving a Model with Highly Conductive Layers” on page 142.

# Building and Solving a Conduction Model

The following example illustrates how to build and solve a conductive heat transfer problem using the General Heat Transfer application mode. The model, taken from a NAFEMS benchmark collection (Ref. 1), shows an axisymmetric steady-state thermal analysis. As opposed to the NAFEMS benchmark model, we use the temperature unit kelvin instead of degrees Celsius for this model.

## *Model Definition—Conduction in a Cylinder*

The modeling domain describes the cross section of a 3D solid as shown in Figure 4-33.



*Figure 4-33: Model geometry and boundary conditions.*

You set three types of boundary conditions:

- Prescribed heat flux
- Insulation/Symmetry
- Prescribed temperature

The governing equation for this problem is the steady-state heat equation for conduction with the volumetric heat source set to zero:

$$\nabla \cdot (-k\nabla T) = 0$$

The thermal conductivity  $k$  is 52 W/(m·K).

### Results

---

The plot in Figure 4-34 shows temperature as a function of position:

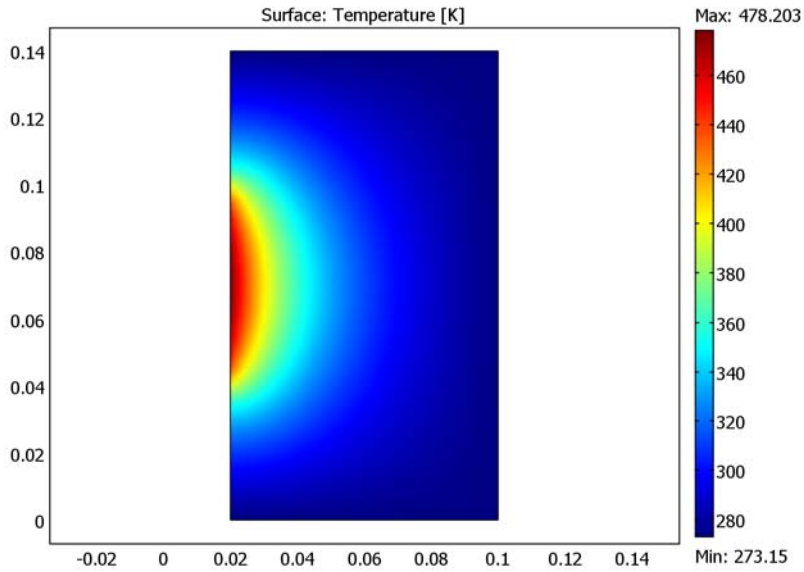


Figure 4-34: Temperature distribution.

The benchmark result for the target location ( $r = 0.04$  m and  $z = 0.04$  m) is a temperature of 59.82 °C, or  $(59.82 + 273.15)$  K = 332.97 K. The COMSOL Multiphysics model, using a default mesh with about 540 elements, gives a temperature of 332.947 K at the same location.

### Reference

---

1. A.D. Cameron, J.A. Casey, and G.B. Simpson, *NAFEMS Benchmark Tests for Thermal Analysis (Summary)*, NAFEMS Ltd., 1986.



---

**Model Library path:** Heat\_Transfer\_Module/Tutorial\_Models/  
cylinder\_conduction

---

### *Modeling Using the Graphical User Interface*

---

- 1 Open the **Model Navigator** and click the **New** tab.
- 2 From the **Space dimension** list select **Axial symmetry (2D)**.
- 3 From the list of application modes select **Heat Transfer Module>General Heat Transfer>Steady-state analysis**. Click **OK**.

#### **GEOMETRY MODELING**

- 1 Open the **Rectangle** dialog box by pressing the Shift key and clicking the **Rectangle/Square** button on the Draw toolbar.
- 2 Specify the following rectangle; when done, click **OK**.

PROPERTY	VALUE
Width	0.08
Height	0.14
Base	Corner
r	0.02
z	0

Next split the left vertical boundary into three boundaries by adding two points.

- 3 Press the Shift key and click the **Point** button on the Draw toolbar. In the resulting dialog box go to the **r** edit field and enter 0.02 0.02, then go to the **z** edit field and enter 0.04 0.1. Click **OK**.
- 4 Click the **Zoom Extents** button on the Main toolbar.

#### **PHYSICS SETTINGS**

##### *Boundary Conditions*

- 1 From the **Physics** menu select **Boundary Settings**.

2 In the resulting dialog box apply these boundary conditions; when done, click **OK**.

SETTINGS	BOUNDARIES 1, 4	BOUNDARIES 2, 5, 6	BOUNDARY 3
Boundary condition	Insulation/Symmetry	Temperature	Heat flux
$T_0$		273.15	
$q_0$			5e5

#### Subdomain Settings

1 From the **Physics** menu select **Subdomain Settings**.

2 Select Subdomain 1, then enter the following properties; when done, click **OK**.

PROPERTY	VALUE
k (isotropic)	52
$\rho$	0
$C_p$	0

### COMPUTING THE SOLUTION

To compute the solution click the **Solve** button on the Main toolbar.

### POSTPROCESSING AND VISUALIZATION

The default plot (Figure 4-34) shows the cylinder's temperature. The NAFEMS benchmark value at the coordinates  $r = 0.04$ ,  $z = 0.04$  is 332.97 K. To compare the computed solution to the NAFEMS solution, follow these steps:

1 From the **Postprocessing** menu, select **Data Display>Subdomain**.

2 From the **Predefined quantities** list select **Temperature**.

3 In both the **r** and **z** edit fields type 0.04.

4 Click **OK**.

The temperature at  $r = 0.04$ ,  $z = 0.04$  appears in the message log at the bottom of the user interface.

# Building and Solving a Convection and Conduction Model

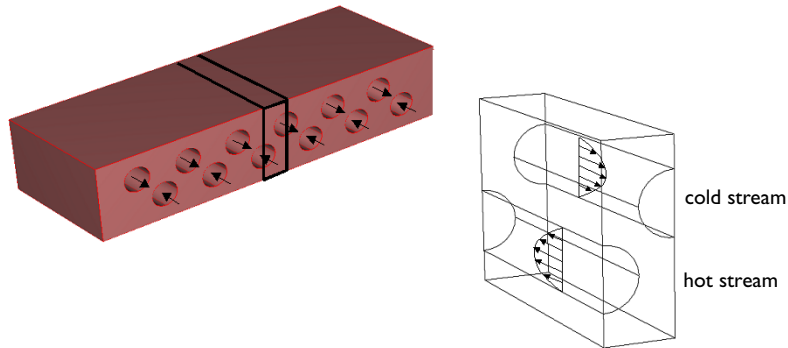
The following example builds and solves a conduction and convection heat transfer problem using the General Heat Transfer application mode.

The example concerns a stainless-steel MEMS heat exchanger, which you can find in lab-on-a-chip devices in biotechnology and in microreactors such as for micro fuel cells. This model examines the heat exchanger in 3D, and it involves heat transfer through both convection and conduction.

## *Model Definition—Heat Exchanger*

---

Figure 4-35 shows the geometry of the heat exchanger. It is necessary to model only one unit cell because they are all almost identical except for edge effects in the outer cells.



*Figure 4-35: Depiction of the modeled part of the heat exchanger (left).*

The governing equation for this model is the heat equation for conductive and convective heat transfer

$$\nabla \cdot (-k \nabla T) = Q - \rho C_p \mathbf{u} \cdot \nabla T \quad (4-15)$$

where  $C_p$  denotes the specific heat capacity ( $\text{J}/(\text{kg}\cdot\text{K})$ ),  $T$  is the temperature ( $\text{K}$ ),  $k$  is the thermal conductivity ( $\text{W}/(\text{m}\cdot\text{K})$ ),  $\rho$  is the density ( $\text{kg}/\text{m}^3$ ),  $\mathbf{u}$  is the velocity vector

(m/s), and  $Q$  is a sink or source term (which you set to zero because there is no production or consumption of heat in the device).

In the solid part of the heat exchanger the velocity vector,  $\mathbf{u} = (u, v, w)$ , is set to zero in all directions. In the channels the velocity field is defined by an analytical expression that approximates fully-developed laminar flow for a circular cross section. For both the hot and cold streams, you set the velocity components in the  $x$  and  $z$  directions to zero.

For the hot stream, the expression

$$v = v_{\max} \left( 1 - \left( \frac{r}{R} \right)^2 \right) \quad (4-16)$$

gives the  $y$ -component of the velocity where

- $v_{\max}$  is the maximum velocity (m/s), which arises in the middle of the channel
- $r$  is the distance from the center of the channel (m)
- $R$  is the channel radius (m)

You describe velocity in the cold stream in the same manner but in the opposite direction

$$v = -v_{\max} \left( 1 - \left( \frac{r}{R} \right)^2 \right)$$

In an extended approach, instead of using the analytical expression for the velocity field, the fluid in the channels can be simulated using the Weakly Compressible Navier-Stokes application mode, which allows the fluid density to vary with temperature. Here the density is defined as

$$\rho = \rho_m \left( 1 - \frac{T - T_m}{T_m} \right) \quad (4-17)$$

where  $\rho_m$  the mean density ( $\text{kg}/\text{m}^3$ ),  $T$  is the temperature (K), and  $T_m = (T_{\text{cold}} + T_{\text{hot}})/2$  is the mean fluid temperature.

The boundary conditions are insulating for all outer surfaces except for the inlet and outlet boundaries in the fluid channels. At the inlets, you specify constant temperatures for the cold and hot streams,  $T_{\text{cold}}$  and  $T_{\text{hot}}$ , respectively. At the outlets, convection dominates the transport of heat so you apply the convective flux boundary condition:

$$-k \nabla T \cdot \mathbf{n} = 0$$

## *Results and Discussion*

---

Figure 4-36 shows the temperature isosurfaces and the heat flux arrows for the conductive heat flux in the device. The temperature isosurfaces clearly show the convective term's influence in the channels. The arrow field indicates that the heat flows from the geometry's hot region to its cold region. Figure 4-37 displays the corresponding results for the extended model. As the plot shows, the temperature distribution is very similar to that in the initial case, which can therefore be concluded to be a good approximation of the extended case.

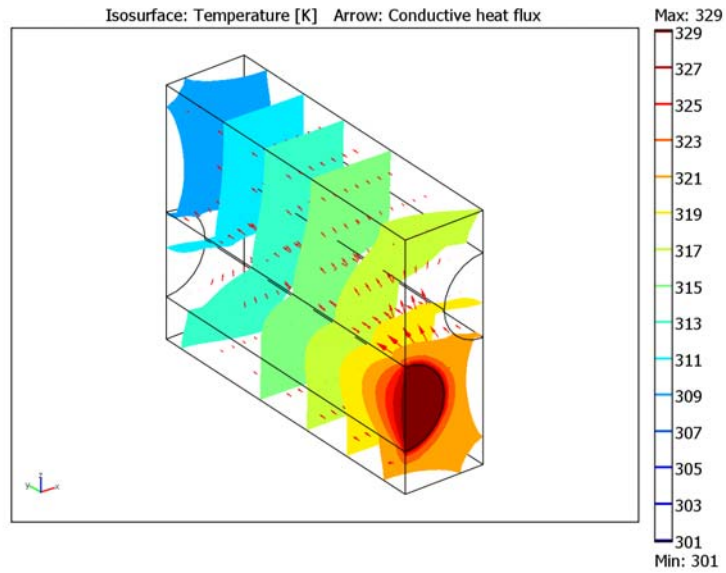


Figure 4-36: Isotherms and conductive heat flux arrow field in the cell unit's geometry.

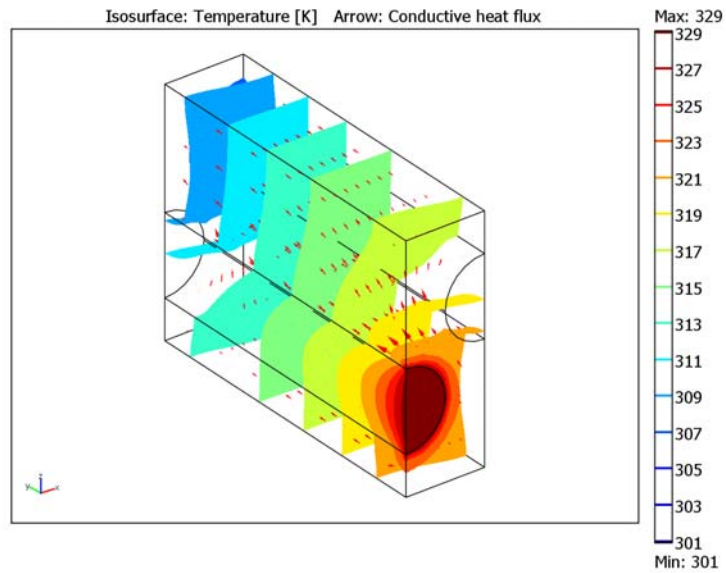
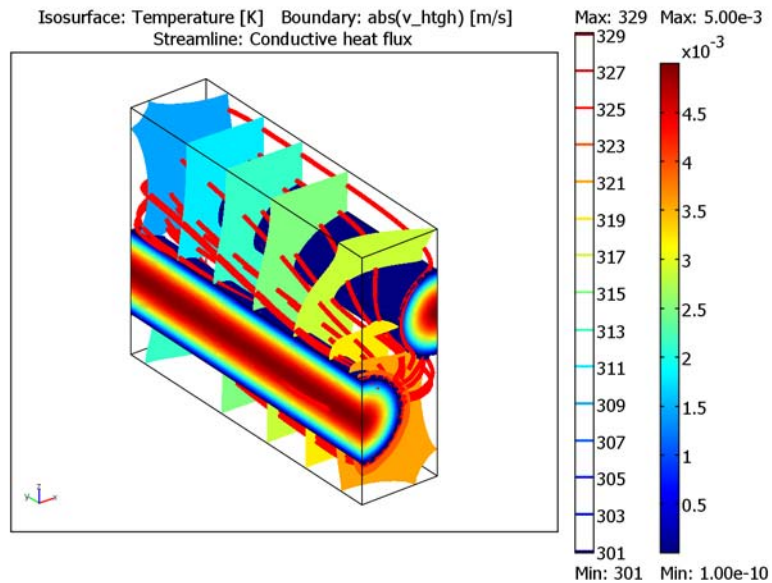


Figure 4-37: Extended model results; isotherms and conductive heat flux arrow field in the cell unit's geometry.

To visualize the velocity field in the channels, Figure 4-38 and Figure 4-39 (extended case) plot the velocity on the tubes' boundaries. The same figures also visualize some streamlines for the conductive heat flux. Comparing the initial and extended models reveals that the channel velocities are slightly different. The analytical velocity profile is constant in the axial direction, as specified, with a centerline velocity magnitude of 5 mm/s (the velocity at the channel centerpoint). In the extended model the centerline velocity varies due to the temperature dependence. In the hot channel the centerline velocity decreases slightly from the inlet. This is due to a net increase of the fluid density as the temperature decreases. The opposite situation occurs in the cold channel. Figure 4-40 shows the velocity magnitude at the channel centerlines in the extended model.



*Figure 4-38: Boundary plot of the velocity field in the channels combined with temperature isosurfaces and heat conduction streamlines in the solid.*

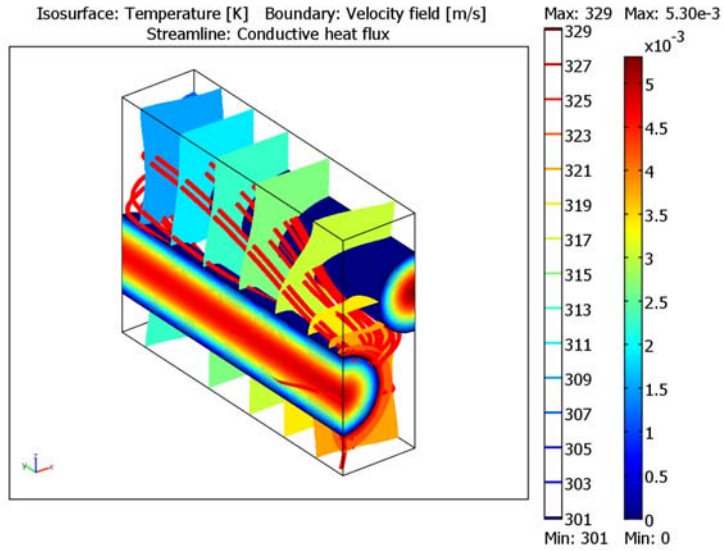


Figure 4-39: Extended model results; boundary plot of the velocity field in the channels combined with temperature isosurfaces and heat conduction streamlines in the solid.

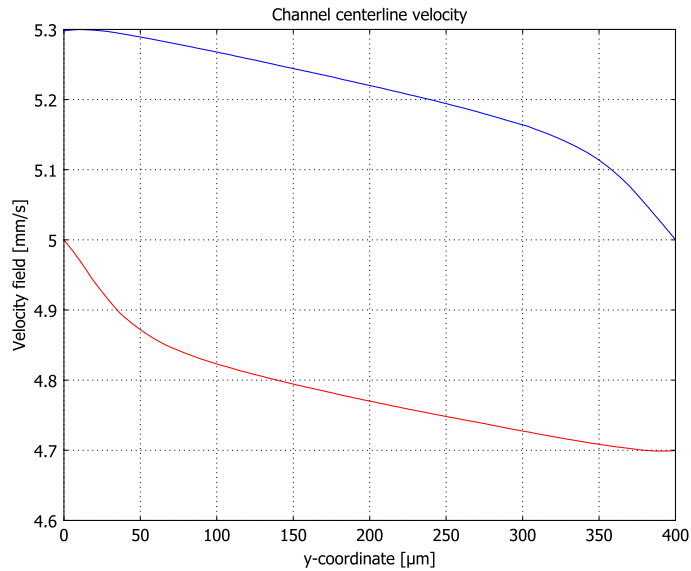


Figure 4-40: Velocity magnitude profiles along the centerline in the cold (blue line) and hot (red line) fluid channel.



The name and location of the model using the analytical fluid velocity are:

---

**Model Library path:** Heat\_Transfer\_Module/Tutorial\_Models/  
heat\_exchanger\_iso

---

The name and location of the extended model, which uses a Weakly Compressible Navier-Stokes application mode to describe the fluid flow, are:

---

**Model Library path:** Heat\_Transfer\_Module/Tutorial\_Models/  
heat\_exchanger\_ni

---

### *Modeling Using the Graphical User Interface*

---

Model the heat exchanger according to the following instructions. Assume that the heat exchanger material is steel and that the channel fluid is water.

#### **MODEL NAVIGATOR**

- 1 Open the **Model Navigator** and go to the **New** page.
- 2 From the **Space dimension** list select **3D**.
- 3 From the list of application modes select  
**Heat Transfer Module>General Heat Transfer>Steady-state analysis**.
- 4 Click **OK**.

#### **OPTIONS AND SETTINGS**

- 1 From the **Options** menu select **Constants**.
- 2 Define these constants (the descriptions are optional); when done, click **OK**.

<b>NAME</b>	<b>EXPRESSION</b>	<b>DESCRIPTION</b>
v_max	5[mm/s]	Maximum velocity
R	5e-5[m]	Channel radius
T_hot	330[K]	Hot stream temperature
T_cold	300[K]	Cold stream temperature
rho_w	1e3[kg/m^3]	Mean density of water

NAME	EXPRESSION	DESCRIPTION
k_w	0.6[W/(m*K)]	Thermal conductivity of water
Cp_w	4200[J/(kg*K)]	Heat capacity of water

## GEOMETRY MODELING

- 1 From the **Draw** menu select **Work-Plane Settings**.
- 2 Click the **Quick** tab, and in the **Plane** area click the **z-x** button.
- 3 Click **OK**.
- 4 Working on the **Geom2** page, press the Shift key and click the **Rectangle/Square** button on the Draw toolbar.
- 5 In the resulting dialog box, enter these properties; when done, click **OK**.

PROPERTY	VALUE
Width	3e-4
Height	1e-4

- 6 Click the **Zoom Extents** button on the Main toolbar.
- 7 Press the Shift key and click the **Ellipse/Circle (Centered)** button in the Draw toolbar.
- 8 In the dialog box that appears, enter the following properties for the circle; when done, click **OK**.

OBJECT DIMENSIONS	EXPRESSION
Radius	5e-5
Base	Center
x	1e-4
y	0

- 9 In the same manner, create a second circle with the following properties:

OBJECT DIMENSIONS	EXPRESSION
Radius	5e-5
Base	Center
x	2e-4
y	1e-4

- 10 Press Ctrl+A to select all three objects, and then click the **Union** button on the Draw toolbar to create one composite object, CO1.
- 11 Press the Shift key and click the **Rectangle/Square** button on the Draw toolbar.

12 Enter these properties; when done, click **OK**.

PROPERTY	VALUE
Width	$2 \cdot 5e-5$
Height	$5e-5$
x	$5e-5$
y	$-5e-5$

13 In the same manner, create a second rectangle with these properties:

PROPERTY	VALUE
Width	$2 \cdot 5e-5$
Height	$5e-5$
x	$1.5e-4$
y	$1e-4$

14 Using the mouse, select both geometry objects CO1 and R2, then click the **Difference** button on the Draw toolbar. This creates the composite object CO2.

15 Select the geometry objects CO2 and R1, then click the **Difference** button on the Draw toolbar.

16 Press Ctrl+A to select the resulting objects.

17 From the **Draw** menu select **Extrude**.

18 In the **Distance** edit field type  $4e-4$ , then click **OK**.

## PHYSICS SETTINGS

### *Subdomain Settings*

To be able to express the velocity profile in the channels, you need an expression variable for the distance from the center of the channels.

1 From the **Options** menu select **Expressions>Subdomain Expressions**.

2 Select Subdomains 2 and 3. In the **Name** edit field type  $r$ , then select each subdomain in turn and in the **Expression** edit field enter the appropriate value from the following table; when done, click **OK**.

SUBDOMAIN 2	SUBDOMAIN 3
$\text{sqrt}(x^2+(z-1e-4)^2)$	$\text{sqrt}((x-1e-4)^2+(z-2e-4)^2)$

3 From the **Physics** menu select **Subdomain Settings**.

- 4 Select Subdomain 1.
- 5 Go to the **General** page and click the **Load** button. From the **Materials** list select **Basic Material Properties>Steel AISI 4340**.
- 6 Click **OK** to close the **Materials/Coefficients Library** dialog box.
- 7 Select Subdomains 2 and 3. On the **General** page, enter these material properties:

PROPERTY	VALUE
k	k_w
$\rho$	rho_w
$C_p$	Cp_w
Q	0

- 8 Go to the **Convection** page. Select the **Enable convective heat transfer** check box.
- 9 Select Subdomain 2 In the **v** edit field type  $v_{max} * (1 - (r/R)^2)$ .
- 10 Select Subdomain 3. In the **v** edit field type  $-v_{max} * (1 - (r/R)^2)$ , then click **OK**.

#### *Boundary Conditions*

- 1 From the **Physics** menu select **Boundary Settings**.
- 2 Enter these boundary conditions; when done, click **OK**.

SETTINGS	BOUNDARY 5	BOUNDARY 15	BOUNDARIES 11, 14	ALL OTHERS
Boundary condition	Temperature	Temperature	Convective flux	Insulation/Symmetry
$T_0$	T_hot	T_cold		

#### **MESH GENERATION**

To resolve the velocity and pressure fields in the boundary-layer regions, use a finer mesh at the fluid-solid interface boundaries.

- 1 From the **Mesh** menu, choose **Free Mesh Parameters**.
- 2 On the **Global** page, click the **Custom mesh size** button.
- 3 Type 1.6 in the **Element growth rate** edit field.
- 4 Click the **Boundary** tab.
- 5 Select Boundaries 6, 8, 12, and 13.
- 6 Type  $1e-5$  in the **Maximum element size** edit field.
- 7 Click the **Remesh** button and then click **OK**.

## COMPUTING THE SOLUTION

Click the **Solve** button on the Main toolbar.

## POSTPROCESSING AND VISUALIZATION

The default plot shows temperature slices. Create Figure 4-36 in the “Results and Discussion” section by following these steps:

- 1 From the **Postprocessing** menu select **Plot Parameters**.
- 2 Go to the **General** page, clear the **Slice** plot and select the **Isosurface** and **Arrow** plots.
- 3 On the **Isosurface** page find the **Predefined quantities** list and select **Temperature**.
- 4 In the **Isosurface levels** area click the **Vector with isolevels** button, then in the corresponding edit field enter 301:2:329.
- 5 On the **Arrow** page, click the **Subdomain Data** tab. Find the **Predefined quantities** list and select **Conductive heat flux**.
- 6 Go to the **Arrow type** list and select **3D arrow**.
- 7 Click **OK**.

The following steps explain how to reproduce Figure 4-38 in the results section:

- 1 From the **Postprocessing** menu select **Plot Parameters**.
- 2 On the **General** page clear the **Arrow** check box, then select the **Boundary** and **Streamline** check boxes.
- 3 Go to the **Boundary** page, find the **Expression** edit field, then enter  $\text{abs}(v\_htgh)$ .  
The reason to use the absolute values is to get a smoother plot that avoids some extremely small negative values that might occur due to the discretization.
- 4 Click the **Range** button.
- 5 In the dialog box that appears, clear the **Auto** check box and enter the following values; when done, click **OK**.

RANGE LIMIT	VALUE
Min	1e-10
Max	5e-3

- 6 On the **Streamline** page, find the **Predefined quantities** list and select **Conductive heat flux**.
- 7 Select **Tube** from the **Line type** list.
- 8 Click **OK**.

Now model the heat exchanger using a Weakly Compressible Navier-Stokes application mode to describe the fluid flow and a General Heat Transfer application mode for the heat equation. Use the predefined combination Non-Isothermal Flow, which contains both application modes with the relevant couplings already entered.

#### MODEL NAVIGATOR

- 1 Open the **Model Navigator** and go to the **New** page.
- 2 From the **Space dimension** list, select **3D**.
- 3 From the **Application Modes** list, select **Heat Transfer Module>Fluid-Thermal Interaction>Non-Isothermal Flow>Steady-state analysis**.
- 4 Click **OK**.

#### OPTIONS AND SETTINGS

- 1 From the **Options** menu, select **Constants**.
- 2 Define these constants (the descriptions are optional); when done, click **OK**.

NAME	EXPRESSION	DESCRIPTION
v_max	5[mm/s]	Maximum velocity
R	5e-5[m]	Channel radius
T_hot	330[K]	Hot stream temperature
T_cold	300[K]	Cold stream temperature
T_mean	(T_hot+T_cold)/2	Mean temperature
eta_w	6.32e-4[Pa*s]	Dynamic viscosity
rho_mean_w	1e3[kg/m^3]	Mean density of water
k_w	0.63[W/(m*K)]	Thermal conductivity of water
Cp_w	4200[J/(kg*K)]	Heat capacity of water

#### GEOMETRY MODELING

- 1 From the **Draw** menu, select **Work-Plane Settings**.
- 2 Click the **Quick** tab, and in the **Plane** area click the **z-x** button.
- 3 Click **OK**.
- 4 Working on the **Geom2** page, press the Shift key and click the **Rectangle/Square** button on the Draw toolbar.

5 In the resulting dialog box, enter these properties; when done, click **OK**.

PROPERTY	VALUE
Width	3e - 4
Height	1e - 4

6 Click the **Zoom Extents** button on the Main toolbar.

7 Press the Shift key and click the **Ellipse/Circle (Centered)** button on the Draw toolbar.

8 In the dialog box that appears, enter the following properties for the circle; when done, click **OK**.

OBJECT DIMENSIONS	EXPRESSION
Radius	5e - 5
Base	Center
x	1e - 4
y	0

9 In the same manner, create a second circle with the following properties:

OBJECT DIMENSIONS	EXPRESSION
Radius	5e - 5
Base	Center
x	2e - 4
y	1e - 4

10 Press Ctrl+A to select all three objects, and then click the **Union** button on the Draw toolbar to create one composite object, CO1.

11 Press the Shift key and click the **Rectangle/Square** button on the Draw toolbar.

12 Enter these properties; when done, click **OK**.

PROPERTY	VALUE
Width	2*5e - 5
Height	5e - 5
x	5e - 5
y	-5e - 5

13 In the same manner, create a second rectangle with these properties:

PROPERTY	VALUE
Width	$2 \times 5e-5$
Height	$5e-5$
x	$1.5e-4$
y	$1e-4$

14 Using the mouse, select both geometry objects CO1 and R2, then click the

**Difference** button on the Draw toolbar. This creates the composite object CO2.

15 Select the geometry objects CO2 and R1, then click the **Difference** button on the Draw toolbar.

16 Press Ctrl+A to select the resulting objects.

17 From the **Draw** menu, select **Extrude**.

18 In the **Distance** edit field type  $4e-4$ , then click **OK**.

## PHYSICS SETTINGS

### Boundary Expressions

1 From the **Options** menu, select **Expressions>Boundary Expressions**.

2 Select Boundary 5 and enter the following names and expressions:

NAME	EXPRESSION
v_inflow	$v_{\max} \cdot (1 - (r/R)^2)$
r	$\sqrt{x^2 + (z - 1e-4)^2}$

3 Select Boundary 15 and enter the following expressions in the **Expression** column for the boundary expressions you created in Step 2; when done, click **OK**.

NAME	EXPRESSION
v_inflow	$-v_{\max} \cdot (1 - (r/R)^2)$
r	$\sqrt{(x - 1e-4)^2 + (z - 2e-4)^2}$

### Subdomain Expressions

1 From the **Options** menu, select **Expressions>Subdomain Expressions**.

2 Select Subdomains 2 and 3. Define the following expression; when done click **OK**.

NAME	EXPRESSION
rho_w	$\rho_{\text{mean\_w}} \cdot (1 - (T - T_{\text{mean}}) / T_{\text{mean}})$



#### *Subdomain Settings—Fluid Flow*

- 1 Select **Multiphysics>Weakly Compressible Navier-Stokes (chns)**.
- 2 From the **Physics** menu, select **Subdomain Settings**.
- 3 Select Subdomain 1 and set it to **Solid domain** in the **Group** list.
- 4 Select Subdomains 2 and 3. On the **Physics** page, set the **Dynamic viscosity** to  $\eta_w$ .
- 5 Click **OK**.

#### *Boundary Conditions—Fluid Flow*

- 1 From the **Physics** menu, select **Boundary Settings**.
- 2 Select Boundaries 4 and 17 and set the **Boundary type** to **Symmetry boundary**.
- 3 Select Boundaries 6, 8, 12, and 13 and set the **Boundary type** to **Wall** with a **No slip Boundary condition**.
- 4 Select Boundaries 5 and 15 and set the **Boundary type** to **Inlet** and the **Boundary condition** to **Velocity**.
- 5 Set the  $\mathbf{v}_0$  velocity component to  $v_{inflow}$ .
- 6 Select Boundaries 11 and 14 and set the **Boundary type** to **Outlet** and the **Boundary condition** to **Pressure**.
- 7 Set the pressure  $p_0$  to 0, then click **OK**.

#### *Subdomain Settings—Heat Transfer*

- 1 Select **Multiphysics>General Heat Transfer (htgh)**.
- 2 From the **Physics** menu, select **Subdomain Settings**.
- 3 Select Subdomain 1 and set it to **Solid domain** in the **Group** list.
- 4 On the **General** page, click the **Load** button to load a **Library material**.
- 5 Under **Basic Material Properties** choose **Steel AISI 4340**.
- 6 Select Subdomains 2 and 3. Set the **Thermal conductivity** to  $k_w$ , the **Density** to  $\rho_w$ , and the **Heat capacity at constant pressure** to  $Cp_w$ .
- 7 Click **OK**.

#### *Boundary Conditions—Heat Transfer*

- 1 From the **Physics** menu, select **Boundary Settings**.
- 2 Select Boundaries 11 and 14. Set the **Boundary condition** to **Convective flux**.
- 3 Select Boundary 15. Set the **Boundary condition** to **Temperature** and set  $T_0$  to  $T_{cold}$ .
- 4 Select Boundary 5. Set the **Boundary condition** to **Temperature** and set  $T_0$  to  $T_{hot}$ .

- 5 Click **OK**.

### **MESH GENERATION**

To resolve the velocity and pressure fields in the boundary-layer regions, use a finer mesh at the fluid-solid interface boundaries.

- 1 From the **Mesh** menu, choose **Free Mesh Parameters**.
- 2 On the **Global** page, click the **Custom mesh size** button.
- 3 Type 1.6 in the **Element growth rate** edit field.
- 4 Click the **Boundary** tab.
- 5 Select Boundaries 6, 8, 12, and 13.
- 6 Type  $1.7 \times 10^{-5}$  in the **Maximum element size** edit field.
- 7 Click the **Remesh** button and then click **OK**.

### **COMPUTING THE SOLUTION**

Click the **Solve** button on the Main toolbar.

### **POSTPROCESSING AND VISUALIZATION**

The default plot shows temperature slices. Create Figure 4-37 by following these steps:

- 1 From the **Postprocessing** menu, select **Plot Parameters**.
- 2 On the **General** page, clear the **Slice** plot and select the **Isosurface** and **Arrow** plots.
- 3 On the **Isosurface** page, find the **Predefined quantities** list and select **General Heat Transfer (htgh)>Temperature**.
- 4 In the **Isosurface levels** area, click the **Vector with isolevels** button and in the corresponding edit field enter 301:2:329.
- 5 Click the **Arrow** tab. On the **Subdomain Data** page, find the **Predefined quantities** list and select **General Heat Transfer (htgh)>Conductive heat flux**.
- 6 Go to the **Arrow type** list and select **3D arrow**. Click **OK**.

The following steps explain how to reproduce Figure 4-39:

- 1 From the **Postprocessing** menu, select **Plot Parameters**.
- 2 On the **General** page clear the **Arrow** check box, then select the **Boundary** and **Streamline** check boxes.
- 3 Go to the **Boundary** page, find the **Predefined quantities** list, and select **Weakly Compressible Navier-Stokes (chns)>Velocity field**.

4 Go to the **Streamline** page, find the **Predefined quantities** list and select **General Heat Transfer (htgh)>Conductive heat flux**.

5 Select **Tube** from the **Line type** list, then click **OK**.

Finally, follow these instructions to reproduce the plots in Figure 4-40:

1 From the **Postprocessing** menu, choose **Cross-Section Plot Parameters**.

2 On the **General** page, select the **Keep current plot** check box.

3 Click the **Line/Extrusion** tab.

4 From the **Predefined quantities** list in the **y-axis data** area, select **Weakly Compressible Navier-Stokes (chns)>Velocity field**.

5 From the **Unit** list, select **mm/s**.

6 In the **x-axis data** area, click the lower option button and then click the **Expression** button.

7 In the **X-Axis Data** dialog box, type  $y$  in the **Expression** edit field. Select  $\mu\text{m}$  from the **Unit** list, then click **OK**.

8 In the **Cross-section line data** area, enter the following settings:

PROPERTY	VALUE
x0	0
x1	0
y0	0
y1	4e-4
z0	1e-4
z1	1e-4

9 Click the **Line Settings** button. From the **Line color** list, select **Color**, then click **OK**.

10 Click **Apply** to generate the first line.

11 Change the following settings in the **Cross-section line data** area:

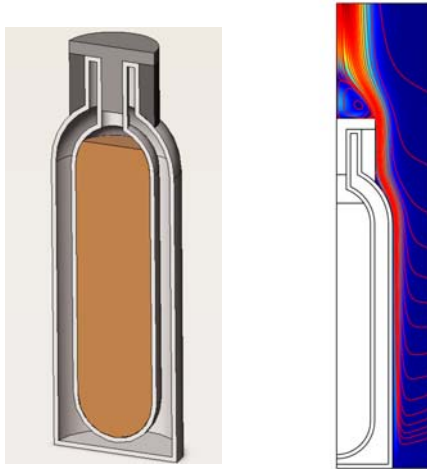
PROPERTY	VALUE
x0	1e-4
x1	1e-4
z0	2e-4
z1	2e-4

12 Click the **Line Settings** button. From the **Line color** list, select **Cycle**, then click **OK**.

- 13** Click **OK** to plot the second line and close the dialog box.
- 14** Click the **Edit Plot** toolbar button in the figure window and edit the title and axis labels as desired. You may also select the **Show grid** check box in the **Grid** tab.

# Building and Solving a Natural Convection Model—Laminar Flow

The following example solves a free-convection problem in which a thermos holding hot coffee dissipates thermal energy. The main interest is to calculate the thermos' cooling power; that is, how much heat it loses per unit time.



This example treats the natural convection cooling using two approaches:

- Using heat transfer coefficients to describe the thermal dissipation
- Modeling the convective flow of air outside the thermos to describe the thermal dissipation

The first approach describes the outside heat flux using a heat transfer coefficient function from the Heat Transfer Coefficients library that the Heat Transfer Module includes. This approach results in a rather simple model that predicts the stationary cooling well and produces accurate results in terms of temperature distribution and cooling power.

The second approach solves for both the total energy balance and the flow equations of the outside cooling air. This model produces detailed results concerning the flow field around the thermos as well as the temperature distribution and cooling power. However, it is more complex and requires significantly more computational resources than the first model.

### GEOMETRY CREATION: IMPORT FROM A CAD PACKAGE

The thermos was designed with a 2D CAD package. You can import that geometry as a DXF file (Vacuum\_Thermos.dxf).

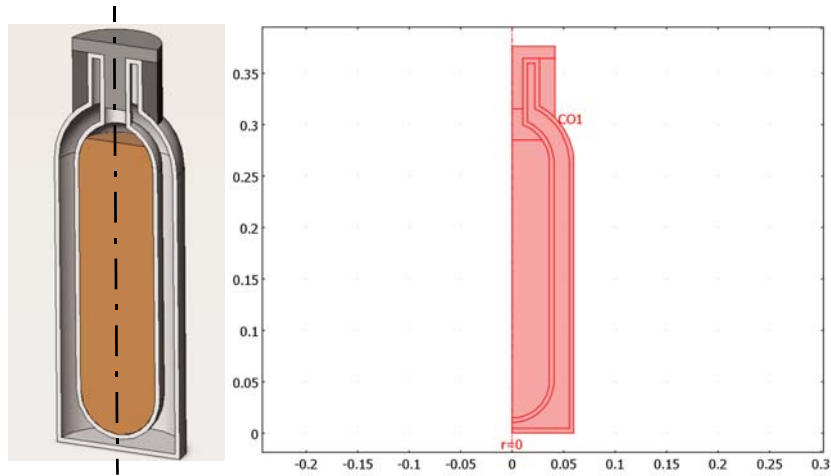


Figure 4-41: A 3D cut-away drawing (left figure) of the thermos with coffee; a stainless-steel shell (light gray) that can be filled with an insulating material such as foam; an insulating ring around the thermos neck (black); and a screw stopper (dark gray). The 2D-axisymmetric representation appears on the right.

### CONTROL VOLUME

The model uses a control volume around the thermos to represent the domain of the surrounding air. Choosing an appropriate control volume for natural convection models is difficult. Your choice strongly influences the model, the mesh, the convergence, and especially the flow behavior. The real-world air domain surrounding the thermos is the entire room or atmosphere in which the thermos is sitting. To make an appropriate truncation, it is important to consider the expected flow behavior and thermal conditions. In this case you expect the hot thermos to induce an upward flow. This means that the flow enters the modeled control volume radially, flowing toward the thermos, and then it exits axially upward. At the entry, the temperature is that of the surrounding air, and the heat flux is convection dominated because the flow exits the control volume upward. For the flow, you know the static pressure is constant around the thermos. However, the flow direction is largely unknown locally around the thermos.

An option is to make the rectangle infinitely large or as large as the external room—but this would result in a very large model, one requiring a supercomputer to solve. At the other extreme, if you make the control volume too small, the solution is affected by the imposed artificial boundary conditions, and there might also be a truncation of flow eddies, making convergence difficult. One way to approach this task is to start with a small control volume, set up and solve the model, then expand the control volume, solve the model again, and see if the results change. This example uses a sufficient control volume by truncating the air domain at  $r = 0.1$  m and  $z = 0.5$  m.

### MATERIAL PROPERTIES

Next consider the materials that make up the thermos model. The screw stopper and insulation ring are made of nylon. The thermos bottle consists of stainless steel, and the filling material between the inner and outer walls is a plastic foam. The surrounding fluid is atmospheric air.

The Heat Transfer Module supplies a library of data for material properties that includes all materials used in this model except the foam, which you specify manually. Table 4-3 provides a list of a standard foam's thermal properties:

TABLE 4-3: FOAM MATERIAL PROPERTIES

PROPERTY	VALUE
Conductivity	0.01 W/(m·K)
Density	60 kg/m <sup>3</sup>
Specific heat capacity	200 J/(kg·K)

### HEAT TRANSFER PHYSICS

This example neglects the convection flows in the hot liquid (coffee). The thermal conductivity in the liquid is large enough to make the temperature gradients rather small. The first two models study the thermos as being thermally stationary, although a real thermos continuously loses heat and cools down. However, because the heat-loss process in the hot liquid is slow compared to the convection flow, a quasi-stationary model assuming that the coffee has a constant temperature of 95 °C (203 °F) is applicable. Such a model reveals the temperature gradients and cooling power. The model results show where the main losses appear, and thus where improvements are possible—which is what a thermos designer is really interested in.

---

**Note:** This model works with the default SI base unit system. However, by using COMSOL Multiphysics' bracket syntax for units, you can enter temperatures in the unit of your choice, whether that is kelvin, degrees Fahrenheit, or degrees Celsius. You can also display the results using any of these temperature units.

---

The general (conduction and convection) heat balance equation is:

$$\rho C_p \frac{\partial T}{\partial t} + \nabla \cdot (-k \nabla T) = Q - \rho C_p (\mathbf{u} \cdot \nabla T)$$

where  $\rho$  is the density,  $C_p$  denotes the specific heat capacity,  $k$  equals the thermal conductivity,  $Q$  is the heat source term, and  $\mathbf{u}$  represents the fluid velocity.

### *Approach 1—Loading a Heat Transfer Coefficient Function*

---

The first model uses a simplified approach and solves the thermal-conduction equation making use of a heat transfer coefficient,  $h$ , to describe the natural convection cooling on the outside surface of the thermos. This approach is very powerful in many situations, especially if the main interest is not so much the flow's behavior but rather its cooling power. By using the appropriate  $h$  correlations, you can generally arrive at accurate results at a very low computational cost. In addition, many correlations are valid for the entire flow regime, from laminar to turbulent flow. This makes it possible to approach the problem directly without predicting whether the flow is laminar or turbulent.

In the first model, the convective term in the equation is zero. There are no heat sources in the thermos, making  $Q$  zero. In addition, the equation is solved for as being stationary, making the time derivative zero. Thus the equation simplifies to

$$\nabla \cdot (-k \nabla T) = 0$$

The second model, which includes the flow, takes the convection in the surrounding air into account.

### **BOUNDARY CONDITIONS**

You assign the vertical boundaries along the axis of symmetry with the symmetry condition (zero gradients); the wetted surface of the stainless-steel shell is held at a constant temperature of 95 °C (203 °F); assume that the bottom is perfectly insulated (zero flux). The thermos surfaces are exposed to air and are convection cooled.



The only remaining energy-balance boundary condition is for the thermos surface. In the first approach a convective heat-transfer coefficient together with the ambient temperature, 20 °C (68 °F), describes the heat flux.

### *Convective Heat Transfer Coefficient*

The outer surface dissipates heat via natural convection. This loss is characterized by the convective *heat transfer coefficient*,  $h$ , which in practice you often determine with empirical handbook correlations. Because these correlations depend on the surface temperature,  $T_{\text{surface}}$ , engineers must estimate  $T_{\text{surface}}$  and then iterate between  $h$  and  $T_{\text{surface}}$  to obtain a converged value for  $h$ . Most of these correlations require tedious computations and property interpolations that make this iterative process quite unpleasant and labor intensive.

Following is a typical handbook correlation for  $h$  for the case of natural convection in air on a vertical heated wall; it is rather complex and requires some skill to set up properly:

$$h = \frac{k \cdot \text{Nu}_L}{L}$$

$$\text{Nu}_L = \left( 825 \cdot 10^{-3} + \frac{387 \cdot 10^{-3} \cdot \text{Ra}_L^{1/6}}{\left[ 1 + \left( \frac{492 \cdot 10^{-3}}{\text{Pr}} \right)^{9/16} \right]^{8/27}} \right)^2$$

$$\text{Ra}_L = 2g \left( \frac{\rho^2 C_p}{\eta k} \right) \frac{T_{\text{surface}} - T_{\text{ambient}}}{T_{\text{surface}} + T_{\text{ambient}}} L^3$$

COMSOL Multiphysics handles these types of nonlinearities internally and adds much convenience to such computations, so there is no need to iterate. In the final model of this thermos example you set up this type of complex correlation manually. However, this first example uses built-in functionality to do the job.

The Heat Transfer Module comes with a library of heat transfer coefficient functions that you can access easily. The model example takes advantage of this functionality.

### *Approach 2—Modeling the External Flow*

---

The second approach to simulate the cooled thermos is to produce a model that computes the convective velocity field around the thermos in detail. Before proceeding with a simulation of this kind, it is a good idea to try to estimate the Grashof number

(see “Nature of the Flow—the Grashof Number” on page 17) because that number influences the choice between assuming laminar flow and applying a turbulence model.

The material properties for air at ambient conditions (293 K, 1 atm) appear in the following table:

TABLE 4-4: AIR MATERIAL PROPERTIES

PROPERTY	VALUE
Dynamic viscosity	$1.77 \cdot 10^{-5} \text{ Pa}\cdot\text{s}$
Density	$1.178 \text{ kg/m}^3$

The model’s length scale is the length of the heated fluid’s flow path, in this example 0.5 m. Notice that this value increases if the modeled flow domain is extended in the direction of the flow.

If you insert these values into the equation for the Grashof number together with the coffee’s temperature (95 °C) for  $T_s$  and 20 °C as  $T_0$ , the result is  $3.1 \cdot 10^9$ , which indicates that the flow is turbulent if the surface temperature is 95 °C.

However, with a more realistic surface temperature, of say 318 K (45 °C), the local Grashof number becomes  $1 \cdot 10^9$  at the top of the modeled domain. This points to a flow that just reaches transition conditions at the top of the modeled domain (where  $L = 0.5 \text{ m}$ ). It is thus possible to describe the model with laminar flow equations.

#### LAMINAR BUOYANCY-DRIVEN FLOW

To model nonisothermal laminar buoyancy-driven flow, the following example uses COMSOL Multiphysics’ Weakly Compressible Navier-Stokes application mode. This application mode implements a generalized Navier-Stokes formulation that takes varying density into account. The equations are:

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho \mathbf{u} \cdot \nabla \mathbf{u} = \nabla \cdot [-p \mathbf{I} + \eta (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) - (2\eta/3 - \kappa)(\nabla \cdot \mathbf{u})\mathbf{I}] + \mathbf{F}$$

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$$

where  $\mathbf{u}$  is the velocity (m/s),  $p$  is the pressure (Pa),  $\kappa$  denotes the dilational viscosity (Pa·s),  $\mathbf{I}$  gives the identity matrix, and  $\mathbf{F}$  equals the internal force.

The buoyancy forces are included by specifying  $\mathbf{F}$  in this way:

$$\mathbf{F} = (\rho - \rho_{\text{ref}})\mathbf{g}$$

with  $\mathbf{g}(r, z) = (0, -9.81) \text{ m/s}^2$ .

You can assume that the dilational viscosity equals zero for atmospheric air. The density is given by the ideal gas law

$$\rho = \frac{pM}{RT}$$

where  $M$  is the average molecular weight of air, and  $R$  is the gas constant. In the model example you use one of the available material libraries to describe the temperature-dependent properties of air.

### **BOUNDARY CONDITIONS**

When solving flow problems numerically, your engineering intuition is crucial in setting good boundary conditions. In this problem, the warm thermos drives vertical air currents along its walls, and they eventually join in a thermal plume above the top of the thermos. Air is pulled from the surroundings toward the thermos where it eventually feeds into the vertical flow. Thus you can expect air to enter the flow domain from the side and exit through the top. You would expect this flow to be quite weak and therefore do not anticipate any significant changes in dynamic pressure.

#### *Flow Boundary Conditions*

- On the top boundary, use a pressure condition. The static pressure is 1 atmosphere ( $1.013 \cdot 10^5$  Pa). It is important to specify the pressure in terms of the absolute pressure because it influences the evaluation of the density function.
- On the right boundary, the viscous stress is zero.
- On the upper-left boundary, the flow domain coincides with the axis of symmetry where you apply the insulation/symmetry condition.
- All other boundaries (the thermos surface and the bottom horizontal line) are walls where you use the No slip condition.

#### *Thermal Boundary Conditions*

- The right boundary of the flow domain is replenished with ambient air, so constrain its temperature to the ambient temperature, in this case 20 °C.
- The top boundary is the exit of the flow domain where convection dominates; accordingly, use a boundary condition for a convective heat flux.
- Again, the top left is best described by insulation/symmetry.
- Assume that the bottom is perfectly insulated.
- All other boundaries (the thermos surface) have continuity in temperature and flux, which COMSOL Multiphysics assigns automatically.

## Results and Discussion

Figure 4-42 shows the temperature distribution in both the thermos and the surrounding air. The results are taken from the model that includes the fluid-flow simulation. However, the temperature results in the solid parts are close to identical for the two models of this example.

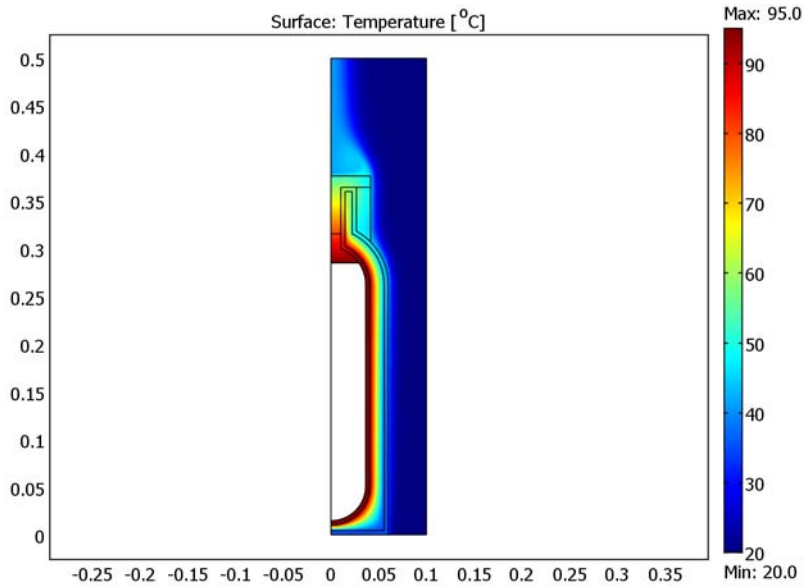


Figure 4-42: Temperature results for the model including the fluid flow.

Your main interest with the model is to calculate the integral cooling power, which indicates how long the coffee stays hot. The first model, using the  $h$ -coefficient to describe the cooling, resulted in a total cooling power of 11.1 W. The second model, including the fluid flow resulted in 11.08 W. The models agree very well.

A second question concerns how the cooling power is distributed on the thermos surface. The heat transfer coefficient represents this property. Figure 4-43 shows a

comparison of the predicted distribution of  $h$  along the height of the thermos between the two models.

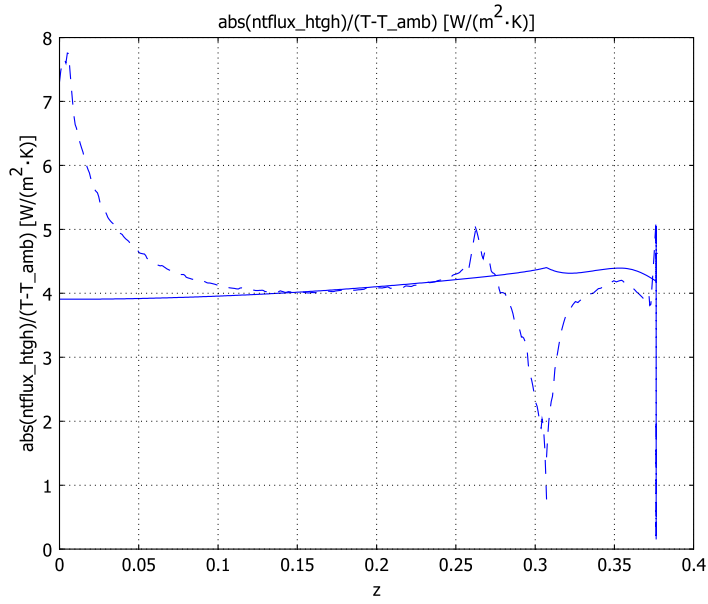


Figure 4-43: Comparison between the predicted heat transfer coefficient for the two models. The dashed line represents the model including the fluid flow.

The fluid flow model does a much better job at describing local cooling power.

Figure 4-44 depicts the flow of air around the thermos calculated from the flow model.

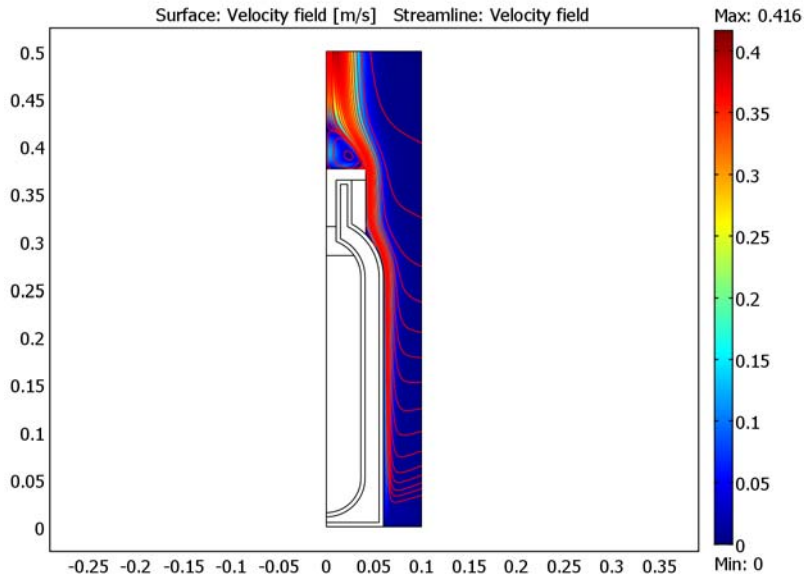


Figure 4-44: The fluid velocity and streamlines for air around the thermos.

One interesting result is the vortex formed above the lid. It reduces the cooling in this region.

### CONCLUSIONS

The predicted heat transfer coefficient,  $h$ , from the fluid-flow model agrees well in an integral sense with that given by the library heat transfer function. However, the library functions do not describe the local effects as well as the flow model does. In fact, the flow model is much more accurate. This means that you can use this type of model to create and calibrate library functions for heat transfer coefficients for complex geometries. Once calibrated, the functions allow you to use the first approach when solving large-scale and perhaps time-dependent models.

The suggested technique is a useful way to model complex geometries with convection cooling for which no accurate heat transfer coefficient descriptions are available. You then solve the “full model” with both heat and fluid flow and postprocess for the local heat transfer coefficient as described earlier. From these plots you can set up your own heat transfer coefficient functions that simplify the model without loss of accuracy in the thermal cooling.

---

**Model Library path:** Heat\_Transfer\_Module/Tutorial\_Models/  
thermos\_laminar\_hcoeff

---

### *Modeling Using the Graphical User Interface*

---

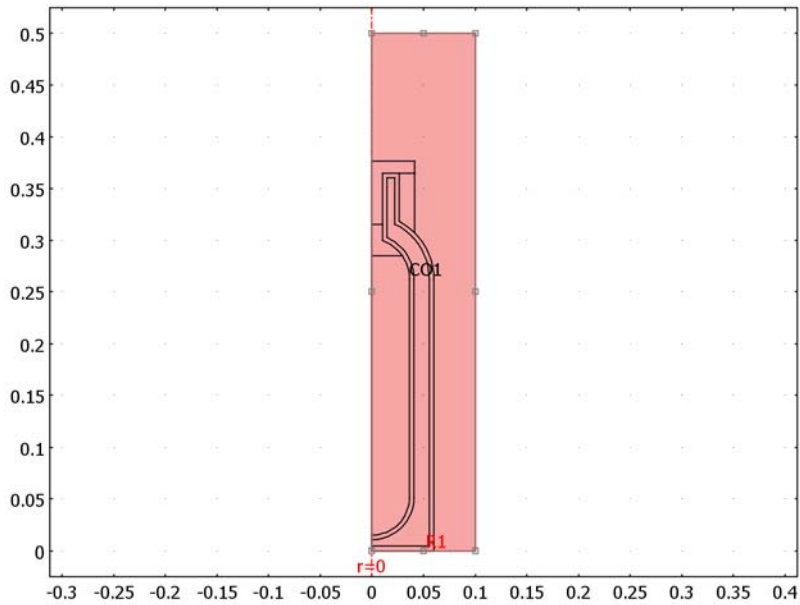
#### **MODEL NAVIGATOR**

- 1 Launch COMSOL Multiphysics. In the **Model Navigator** go to the **Space dimension** list and select **Axial symmetry (2D)**.
- 2 From the list of application modes select **Heat Transfer Module>General Heat Transfer>Steady-state analysis**.
- 3 Click **OK**.

#### **GEOMETRY**

- 1 Select **File>Import>CAD Data From File**.
- 2 Browse to the folder models/Heat\_Transfer\_Module/Tutorial\_Models in the COMSOL installation directory on your system.
- 3 Select the file Vacuum\_Thermos.dxf.
- 4 Click **Import**.
- 5 Using the mouse, select the lower left corner point of the imported drawing and drag it to the coordinates  $r = 0, z = 0$ . The geometry must align with the symmetry axis.
- 6 Select **Draw>Specify Objects>Rectangle**. Enter a **Width** of 0.1 and a **Height** of 0.5, then click **OK**.

- Click the **Zoom Extents** button on the Main toolbar. The following plot then appears:



### CONSTANTS

- From the **Options** menu select **Constants**, then define the following constants (the descriptions are optional):

NAME	EXPRESSION	DESCRIPTION
k_foam	0.01[W/(m*K)]	Foam thermal conductivity
p_atm	1e5[Pa]	Air pressure
Length	0.38[m]	Height of thermos wall
T_amb	20[degC]	Temperature of surrounding air

- Click **OK**.

### SUBDOMAIN SETTINGS

- Select **Physics>Subdomain Settings**.
- From the **Subdomain selection** list select Subdomains 3 and 6 by pressing the Ctrl key while clicking the entries. These subdomains represent the outer air and hot coffee, respectively.



- 3 Clear the **Active in this domain** check box.  
At this point the heat balance is not modeled in these regions, and no thermal equations are set up for these subdomains.
- 4 From the **Subdomain selection** list, select Subdomain 1 (the stainless-steel shell).
- 5 Open the **Materials/Coefficients Library** by clicking the **Load** button.
- 6 Load the material properties by selecting **Basic Material Properties>Steel AISI 4340**. Click **OK**.
- 7 From the **Subdomain selection** list select Subdomain 2 (the insulating cavity).
- 8 Enter the material properties for the plastic foam (as given in Table 4-3 on page 111): type  $k_{\text{foam}}$  in the **Thermal conductivity** edit field, type 60 in the **Density** edit field, and type 200 in the **Heat capacity** edit field.
- 9 From the **Subdomain selection** list select Subdomain 4 (the air above the coffee).
- 10 Click the **Load** button. Select the material **Basic Material Properties>Air**, which is one of the first materials in the library. Click **OK**.
- 11 In the expression in **Density** edit field, replace  $p$  with  $p_{\text{atm}}$  so the function call reads  $\text{rho}(p_{\text{atm}}[1/\text{Pa}], T[1/\text{K}]) [\text{kg}/\text{m}^3]$ . In this way, the density is calculated under atmospheric conditions.
- 12 From the **Subdomain selection** list select Subdomain 5 (the screw stopper) and Subdomain 7 (the plastic ring).
- 13 Click the **Load** button. Select the material **MEMS Material Properties>Polymers>Nylon**, then click **OK** to close the **Materials/Coefficients Library** dialog box.
- 14 Click **OK** to close the **Subdomain Settings** dialog box.

## BOUNDARY CONDITIONS

- 1 Select **Physics>Boundary Settings**.
- 2 From the **Boundary selection** list, select the boundaries that enclose the coffee except for the ones that coincide with the axis of symmetry. Press and hold the Ctrl key when making this multiple selection. Verify that you have selected Boundaries 8, 23, 32, and 37.
- 3 Change the selection in the **Boundary condition** list to **Temperature**.
- 4 In the  $T_0$  edit field for **Temperature** enter 95[degC].
- 5 Select the bottom of the thermos (Boundary 2). Notice that **Insulation/Symmetry** is the default setting, which you can keep. Analogously, the boundaries on the axis of symmetry are already set up correctly as **Insulation/Symmetry**.

- 6 Click **OK**.
- 7 Select **File>Save As** and save the model with the name `Thermos_start`.

### LOADING A HEAT TRANSFER COEFFICIENT FUNCTION

- 1 Select **Physics>Boundary Settings**.
- 2 From the **Boundary selection** list, select the outer surfaces of the thermos (Boundaries 13, 25, 26, 28, and 38).
- 3 Select **Heat flux** from the **Boundary condition** list.
- 4 Click the **Load** button to access the **Materials/Coefficients Library**.
- 5 Select **Heat Transfer Coefficients>Nat. Vertical wall, L=height**, then click **OK**. This loads the heat transfer coefficient function to the boundary-condition settings.

The function you loaded is valid for cases of natural convection to air for vertical surfaces. It requires you to input the surface and ambient air temperatures (in kelvin) plus the wall height in meters. The temperatures are set to the variable  $T$  and the value in the  $T_{inf}$  edit field (which has the variable name `Tinf_htgh`), respectively; you can keep these values but you need to modify the wall height entry:

- 6 In the **h** edit field replace `L_htgh` (the third argument for `h_ave`) with your constant for the height, `Length`.

The function in the **h** edit field should read

**`h_ave(T[1/K],Tinf_htgh[1/K],Length[1/m])[W/(m^2*K)]`** in a bold font.

- 7 In the  $T_{inf}$  edit field type `T_amb`.
- 8 Click **OK**.

### MESHING AND SOLVING

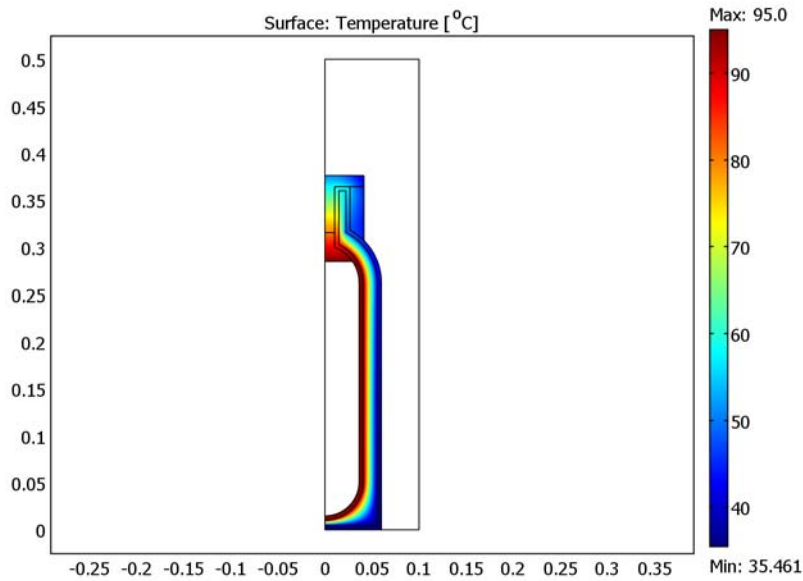
Click the **Solve** button on the Main toolbar.

### POSTPROCESSING

The default plot shows the temperature. To change the unit in the plot to degrees Celsius, do as follows:

- 1 Click the **Plot Parameters** button on the Main toolbar.
- 2 In the dialog box that opens, click the **Surface** tab. In the **Unit** list select **°C**, then click **OK**.

This results in the following plot:



Your main interest with the model is to calculate the cooling power, which indicates how long the coffee stays hot. It is easy to compute the heat loss to the air using the **Postprocessing>Boundary Integration** dialog box according to the following steps.

- 1 Select **Postprocessing>Boundary Integration**.
- 2 Change the selection in the **Predefined quantities** list to **Normal total heat flux**. Note that the outer surfaces of the thermos (Boundaries 13, 25, 26, 28, and 38) are still selected from the previous operation.
- 3 Select the **Compute surface integral (for axisymmetric modes)** check box, then click **OK** to integrate.

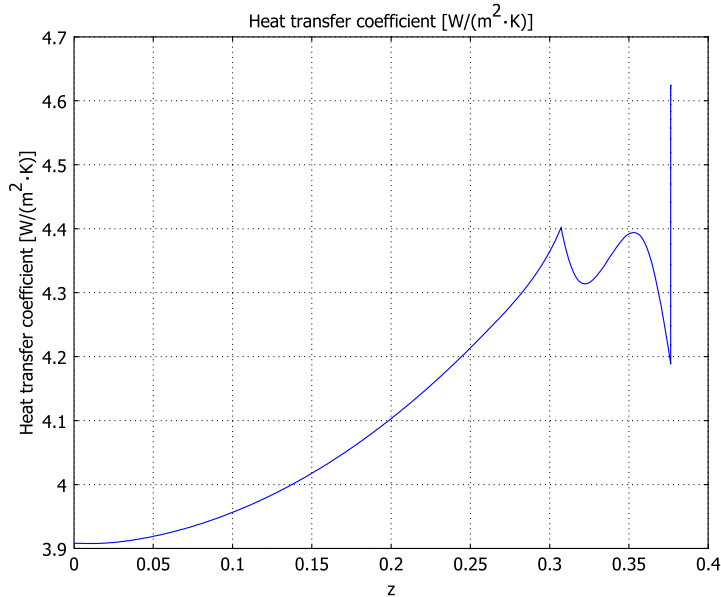
COMSOL Multiphysics posts the results of the integration in the message log at the bottom of the user interface. The heat-loss rate of the thermos is 11.1 W.

A second question concerns how the cooling power is distributed on the thermos surface. The heat transfer coefficient represents this property.

- 1 To visualize the value of the heat transfer coefficient on the thermos surface, select **Postprocessing>Domain Plot Parameters**.

- 2 On the **Line/Extrusion** page go to the **Predefined quantities** list and select **Heat transfer coefficient**.
- 3 From the **Boundary selection** list select Boundaries 13, 25, 26, 28, and 38 (the outer surface of the thermos).
- 4 From the **x-axis data** list select **z**, then click **Apply**.

These steps produce the following plot of the heat transfer coefficient of the thermos surface:



As you can see, the value of  $h$  is in the range of 4 to 4.5  $W/(m^2 \cdot K)$ .

Do not close the plot window; later you plot more information in the same figure.

Finally, plot the temperature of the thermos surface.

- 1 In the **Domain Plot Parameters** dialog box click the **Line/Extrusion** tab. In the **Predefined quantities** list select **Temperature** and in the **Unit** list select **°C**.
- 2 On the **General** page, select **New figure** from the **Plot in** list, then click **OK**.  
The temperature of the thermos surface is in the range of 36 °C to 45 °C along its height.
- 3 Choose **File>Save As** and save the model with the name **Thermos\_load\_h**.

**Model Library path:** Heat\_Transfer\_Module/Tutorial\_Models/  
thermos\_laminar\_flow

---

#### MODEL NAVIGATOR

The following instructions are based on continuing with the previous model.

- 1 Select **Multiphysics>Model Navigator**.
- 2 From the list of application modes select **Heat Transfer Module>Weakly Compressible Navier-Stokes>Steady-state analysis**.
- 3 Click the **Add** button, then click **OK**.

The flow application mode is valid only in the air, Subdomain 6. Deactivate the flow equations in the other subdomains.

- 1 Select **Physics>Subdomain Settings**.
- 2 Select Subdomains 1–5, and 7 (all subdomains except the air, Subdomain 6).
- 3 Clear the **Active in this domain** check box, then click **Apply**.
- 4 In the **Subdomain selection** list, select Subdomain 6.
- 5 Select **Air** from the **Library material** list.
- 6 Edit the  $\rho$  edit field for the pressure so that it reads  $\rho((p+p_{atm})[1/Pa], T[1/K])[kg/m^3]$ .
- 7 In the **Fz** edit field type  $9.81[m/s^2]*(\rho_{air\_ref}-\rho_{chns})$ .

Doing the above steps models the buoyancy force. The variable  $\rho_{chns}$  refers to the material-property edit field for density in this application mode (named  $chns$ , thus the designation  $\rho_{chns}$ ). It uses the material-library function for the density and so it is temperature dependent.

This is done so that the variable  $p$  corresponds to the pressure variations.

- 8 Click **OK**.

#### EXPRESSIONS

Next create the variable  $\rho_{air\_ref}$ , the density of air at the ambient pressure and temperature.

- 1 Select **Options>Expressions>Scalar Expressions**.

- 2 Set up a new variable with **Name** rho\_air\_ref defined by the **Expression**  $\text{mat2\_rho}(p_{\text{atm}}[1/\text{Pa}], T_{\text{amb}}[1/\text{K}]) [\text{kg}/\text{m}^3]$ . Optionally, enter Density of air at ambient pressure and temperature in the **Description** column.
- 3 Click **OK**.

This expression calls the density function of Material 2 in the model, which here is air. It evaluates the density at the temperature  $T_{\text{amb}}$  and the pressure  $p_{\text{atm}}$ . If you want to check the assigned number of a loaded material, choose **Options>Materials/Coefficients Library** and look in the **Model** folder. The material **Air** is second in the list and thus is Material 2.

## PHYSICS SETTINGS

### *Subdomain Settings—General Heat transfer*

Now reactivate the heat transfer equations in the surrounding air domain, couple them to the fluid-flow velocities, and define the material properties.

- 1 Select **Multiphysics>General Heat Transfer**.
- 2 Select **Physics>Subdomain Settings**.
- 3 In the **Subdomain selection** list, select Subdomain 6 (the surrounding air).
- 4 Select the **Active in this domain** check box.
- 5 Select **Air** from the **Library material** list.
- 6 Edit the  $\rho$  edit field so that it reads  $\text{rho}((p+p_{\text{atm}})[1/\text{Pa}], T[1/\text{K}]) [\text{kg}/\text{m}^3]$ .
- 7 On the **Convection** page select the **Enable convective heat transfer** check box.
- 8 Enter  $u$  and  $v$  in the **Velocity field** edit fields. These are the flow variables. Now you have coupled the two application modes and created a multiphysics model.
- 9 Click **OK**.

### *Boundary Conditions—General Heat Transfer*

- 1 Select **Physics>Boundary Settings**.
- 2 Select Boundary 12 and set the **Boundary condition** to **Axial symmetry**.
- 3 Select Boundary 14 and set the **Boundary condition** to **Convective flux**. This means that the outflux of heat is dominated by convection and that the temperature gradients are zero.
- 4 Select Boundary 29. Verify that the boundary condition defaults to **Insulation/Symmetry**.
- 5 Select Boundary 30 and change the **Boundary condition** to **Temperature**.

6 In the **Temperature** edit field type  $T_{amb}$ , then click **OK**.

*Boundary Conditions—Weakly Compressible Navier-Stokes*

1 Select **Multiphysics>Weakly Compressible Navier-Stokes**.

2 Select **Physics>Boundary Settings**. Set boundary conditions according to the table

SETTING	BOUNDARY 12	BOUNDARY 14	BOUNDARY 30	BOUNDARIES 13, 25, 26, 28, 29, 38
Boundary type	Symmetry boundary	Outlet	Open boundary	Wall
Boundary condition	Axial symmetry	Normal stress	Normal stress	No slip
$f_0$		0	0	

3 Click **OK**.

## MESHING AND SOLVING

Natural-convection problems can be difficult to solve because the flow is internally driven with a certain boundary layer. A successful solution requires a sufficient resolution of the boundary layer that forms on all no-slip surfaces.

You can build successful meshes in two ways. First, with enough experience you can set up the mesh yourself in the **Mesh Parameters** dialog box. If the problem does not converge, the mesh probably needs more attention. The second approach is to use COMSOL Multiphysics' adaptive solver, which is handy. However, it requires a converged solution to refine the mesh and typically needs some tuning.

In this case, a mesh with quad-elements proves very efficient.

1 Select **Mesh>Free Mesh Parameters**.

2 Click the **Subdomain** tab.

3 Select Subdomain 6.

4 In the **Maximum element size** edit field, type 0.005.

5 From the **Method** list, select **Quad**.

6 Click the **Remesh** button and then click **OK**.

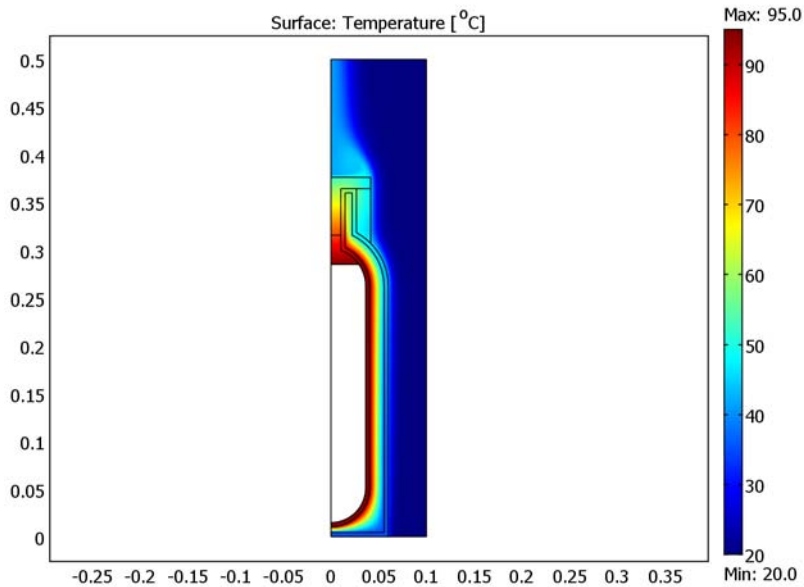
The solver settings are inherited from the General Heat Transfer application mode. The system including the Weakly Compressible Navier-Stokes is more complicated, however, and requires some tuning of the solver settings.

1 Select **Solve>Solver Parameters**.

- 2 Click the **Stationary** tab.
- 3 Select the **Highly nonlinear problem** check box.
- 4 Click the **Advanced** tab.
- 5 From the **Type of scaling** list, select **None** because no scaling provides better convergence for models that include fluid flow.
- 6 Click **OK**.
- 7 Click the **Solve** button on the Main toolbar.

## POSTPROCESSING

In the drawing area a plot similar to the one in the following figure appears.



Evaluate the integral heat flux for this final model.

- 1 Once again select **Postprocessing>Boundary Integration**.
- 2 Select the outer surfaces of the thermos (Boundaries 13, 25, 26, 28, and 38).
- 3 In the **Expression** edit field type `abs(ntflux_htgh)`, then click **OK**.

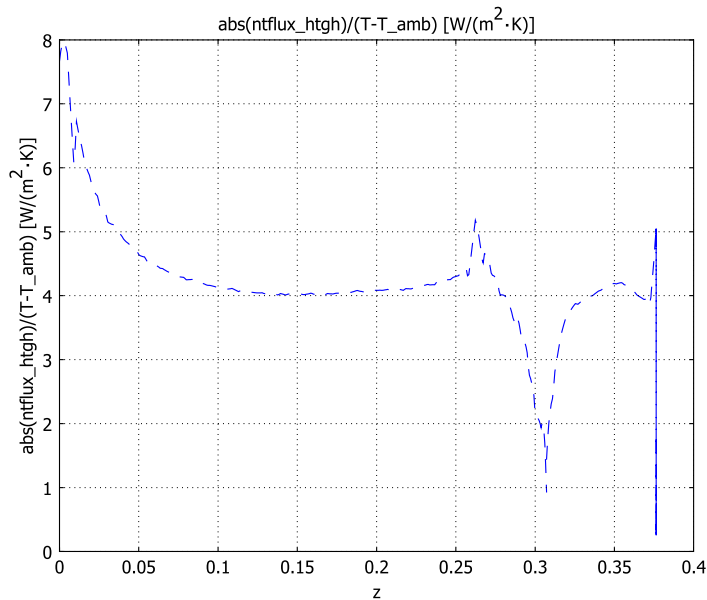
The value in the message log is approximately 11.08 W.



The following steps show how to plot the distribution of the  $h$  coefficient along the thermos surface:

- 1 Select **Postprocessing>Domain Plot Parameters**.
- 2 If you still have the results for  $h$  from the previous model open in a plot figure, go to the **General** page, select **Figure 1** from the **Plot in** list, and select the **Keep current plot** check box.
- 3 Click the **Line/Extrusion** tab and verify that Boundaries 13, 25, 26, 28, and 38 are still selected.
- 4 Click the **Line Settings** button, select the **Dashed line** style, and then click **OK**.
- 5 In the **Expression** edit field, type  $\text{abs}(\text{ntflux\_htgh}) / (\text{T}-\text{T\_amb})$ , which corresponds to a calculated heat transfer coefficient.
- 6 Select **z** in the **x-axis data** list, then click **OK**.

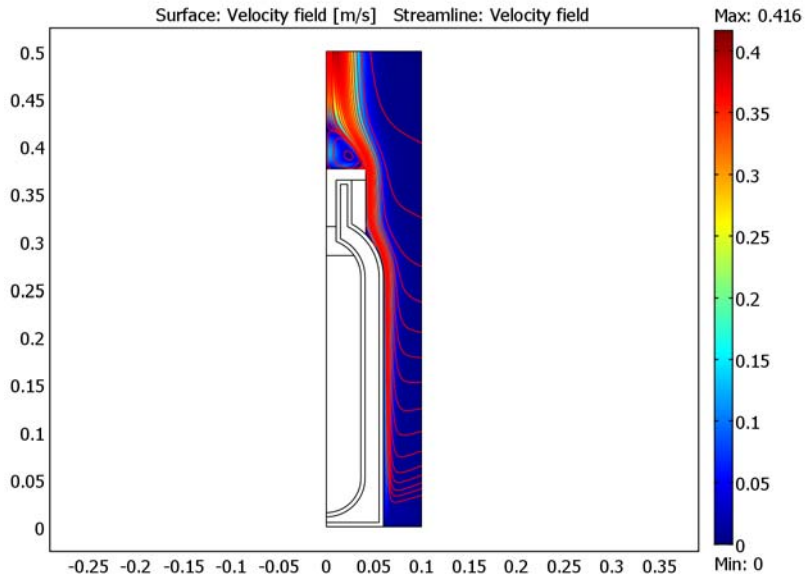
You get the following plot (with the result from the current model only):



Now move on to study the velocity field and the flow lines.

- 1 Click the **Plot Parameters** button on the Main toolbar.
- 2 On the **Surface** page, select **Weakly Compressible Navier-Stokes (chns)>Velocity field** from the **Predefined quantities** list.

- 3 Go to the **Streamline** page and select the **Streamline plot** check box to enable the plot.
- 4 In the **Predefined quantities** list select **Weakly Compressible Navier-Stokes (chns)>Velocity field**.
- 5 In the **Streamline plot type** list select **Magnitude controlled**.
- 6 Click **OK** to get the following plot:



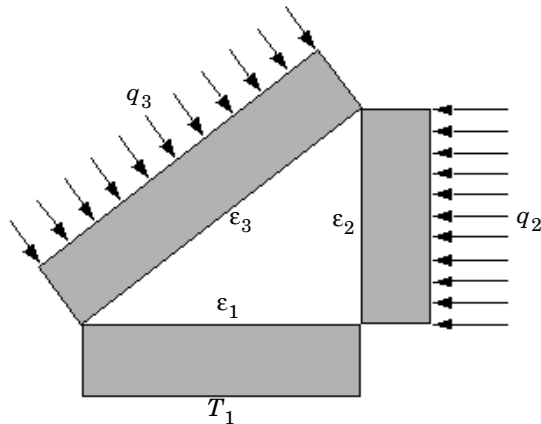
# Building and Solving a Radiation Model

The following example shows how to build and solve a radiative heat transfer problem using the General Heat Transfer application mode. In particular, this 2D model illustrates the use of the surface-to-surface radiation feature. This example's simple geometry allows a comparison of results from COMSOL Multiphysics with a theoretical solution.

## *Model Definition—Radiation in a Cavity*

---

The geometry consists of three rectangles with lengths 3 m, 4 m, and 5 m, and all have a width of 1 m. The rectangles are situated such that they form a triangular cavity (see Figure 4-45).



*Figure 4-45: Three subdomains form a triangular cavity to study heat radiation.*

The rectangles are made of copper, which is a good thermal conductor, and they transfer heat internally by conduction. Further, the three inner boundaries that form the cavity can exchange heat only by means of surface-to-surface radiation. The surface emissivities are  $\epsilon_1 = 0.4$  on the bottom boundary,  $\epsilon_2 = 0.6$  on the vertical boundary, and  $\epsilon_3 = 0.8$  on the inclined boundary.

The system holds the temperature on the (outer) lower boundary at  $T_1 = 300$  K. Further, the system experiences an inward heat flux of  $q_2 = 2000$  W/m<sup>2</sup> on the outer boundary of the vertical rectangle and  $q_3 = 1000$  W/m<sup>2</sup> on the outer boundary of the tilted one. All remaining outer boundaries are insulated.

This model examines the steady state of the cavity's interior boundaries. Specifically, you know the value of the heat flux on the inclined and vertical boundaries, and you want to find the temperature along these boundaries. Conversely, you know the temperature of the bottom boundary and want to determine the heat flux that leaves through it.

### Results

---

Figure 4-46 shows the temperature field resulting from a simulation. If you run the model as described later in this section you can see from the color display that the temperatures are fairly constant within the three rectangles. The theoretical results predict that a heat flux  $Q_1 = -11,000$  leaves the cavity through the bottom boundary, that the vertical boundary has a temperature of  $T_2 = 641$  K, and that the inclined boundary has a temperature of  $T_3 = 600$  K. The results obtained are very close to these theoretical values. The section “Comparing Results with Theory” on page 139 provides more details on the verification of these results.

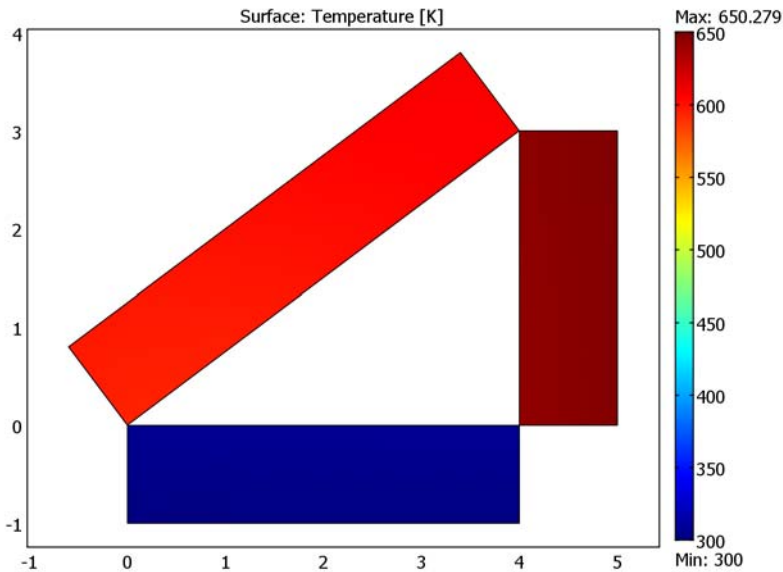
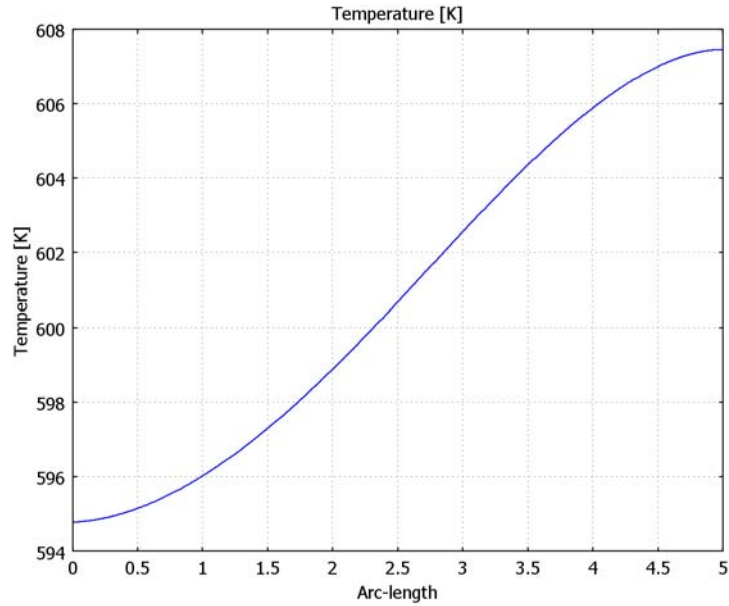


Figure 4-46: Temperature as a function of position.

Figure 4-47 details the temperature distribution along the inclined boundary of the cavity. Note that while the temperature is not constant, it varies by only a few percent. The lowest temperature appears on the bottom left, and the highest temperature is on the right, which seems reasonable because that portion of the boundary is located adjacent to the vertical boundary, which has the highest inward heat flux.



*Figure 4-47: Temperature distribution along the inclined boundary of the cavity.*

Figure 4-48 plots the radiosity along the inclined boundary (in other words, the total heat flux that leaves the boundary into the cavity). The radiosity is the sum of the heat flux the boundary *emits* plus the heat flux it *reflects*.

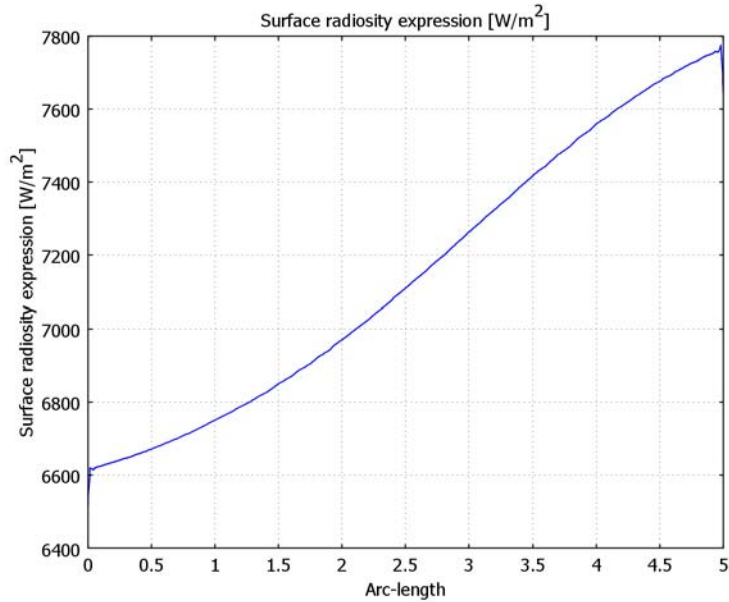


Figure 4-48: Radiosity along the inclined boundary of the cavity.

Like temperature, the radiosity is fairly constant along the boundary. The small variations have roughly the same distribution as the temperature in Figure 4-47.

### Reference

1. A. Bejan, *Heat Transfer*, Wiley, 1993.

**Model Library path:** Heat\_Transfer\_Module/Tutorial\_Models/  
cavity\_radiation

### MODEL NAVIGATOR

- 1 Open the **Model Navigator** and click the **New** tab.
- 2 Select **2D** from the **Space dimension** list.
- 3 In the list of application modes select **Heat Transfer Module>General Heat Transfer**.
- 4 Click **OK**.

### OPTIONS AND SETTINGS

- 1 From the **Options** menu, select **Axes/Grid Settings**.
- 2 On the **Axis** page in the resulting dialog box enter these limits; when done, click **OK**.

PROPERTY	VALUE
x min	-2
x max	8
y min	-2
y max	6

### GEOMETRY MODELING

The geometry consists of three rectangles positioned such that they create a triangular cavity. To ensure that their corners do not touch, offset the rectangles by a very small amount to prevent them from exchanging heat by conduction.

- 1 Press the Shift key and click the **Rectangle/Square** button on the Draw toolbar.
- 2 In the resulting dialog box enter these properties; when done, click **OK**.

PROPERTY	VALUE
Width	4
Height	1
x	0
y	-1.001

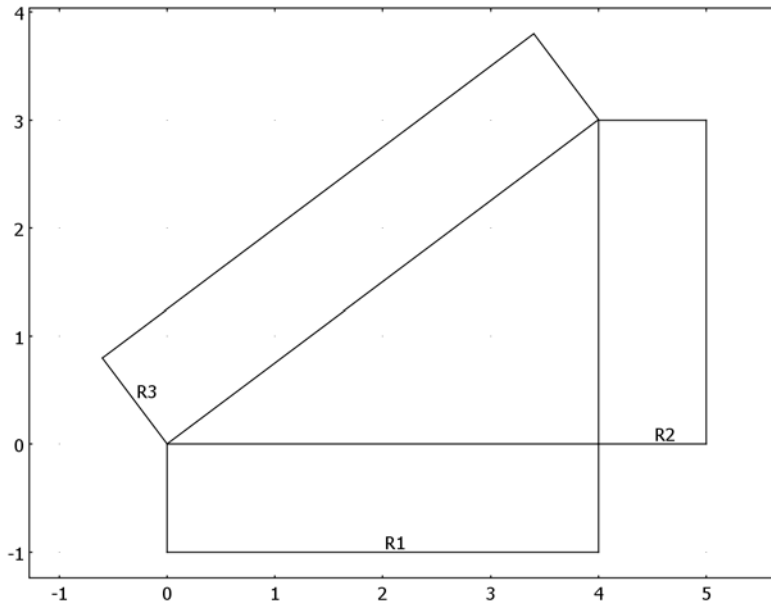
- 3 In the same manner, create a second rectangle using these properties:

PROPERTY	VALUE
Width	1
Height	3

x	4.001
y	0

4 For the third rectangle you must also specify a rotation angle. Define the rectangle with these properties; when done, click **OK**.

PROPERTY	VALUE
Width	5
Height	1
x	-0.001
y	0.001
$\alpha$	$180/\pi*\text{atan}(3/4)$



*The completed geometry for the radiation model.*



## PHYSICS SETTINGS

### Subdomain Settings

Conduction is the only heat transfer mechanism in all three subdomains. COMSOL Multiphysics sets its default thermal material properties for copper, which is a good thermal conductor. Even so, you must set a few parameters.

- 1 Go to the **Physics** menu and select **Subdomain Settings**.
- 2 In the dialog box select Subdomains 1–3, then verify that the **Enable convective heat transfer** check box on the **Convection** tab is cleared.
- 3 Click the **Init** tab and set the initial **Temperature** to 300.
- 4 Click **OK**.

### Boundary Conditions

From the **Physics** menu, open the **Boundary Settings** dialog box and enter boundary conditions from the following two tables; when done, click **OK**.

SETTINGS	BOUNDARY 2	BOUNDARY 12	BOUNDARY 5	BOUNDARIES 1, 4, 7, 8, 10, 11
Type	Heat flux	Heat flux	Temperature	Insulation/Symmetry
$T_0$			300	
$q_0$	1000	2000		
Radiation type	None	None	None	

SETTINGS	BOUNDARY 3	BOUNDARY 6	BOUNDARY 9
Type	Heat flux	Heat flux	Heat flux
$q_0$	0	0	0
Radiation type	Surface-to-surface	Surface-to-surface	Surface-to-surface
$\epsilon$	0.8	0.4	0.6
$T_{\text{amb}}$	300	300	300
Surface radiosity expression	J	J	J
Member of group(s)	1	1	1

## MESH GENERATION

- 1 From the **Mesh** menu select **Free Mesh Parameters**.
- 2 On the **Boundary** tab, select Boundaries 3, 6, and 9, and then enter a **Maximum element size** of  $5e-2$ .

- 3 On the **Point** tab, select Points 2, 4, 6, 8, 9, and 10, and then enter a **Maximum element size** of  $1e-2$ .
- 4 Click **Remesh**, then click **OK**.

#### COMPUTING THE SOLUTION

- 1 From the **Solve** menu select **Solver Parameters**.
- 2 Click the **Stationary** tab and enter a value of 50 in the **Maximum number of iterations** edit field.
- 3 Click **OK**.
- 4 Click the **Solve** button on the Main toolbar.

#### POSTPROCESSING AND VISUALIZATION

The default plot is the temperature field (Figure 4-46). The following steps reproduce Figure 4-47, which shows the temperature on the inclined boundary:

- 1 From the **Postprocessing** menu select **Domain Plot Parameters**.
- 2 On the **General** page, click the **Line/Extrusion plot** button in the **Plot type** area.
- 3 Click the **Line/Extrusion** tab.
- 4 In the **Boundary selection** list select Boundary 3.
- 5 From the list of **Predefined quantities** select **Temperature**, then click **Apply**.

To reproduce Figure 4-48, which shows the radiosity on the inclined boundary, follow these steps:

- 1 From the **Postprocessing** menu return to the **Domain Plot Parameters** dialog box.
- 2 From the list of **Predefined quantities** select **Surface radiosity expression**. Click **OK**.

## Comparing Results with Theory

---

As mentioned at the start of this section, you can compare the results from this model to theoretical values, which are based on the sketch in Figure 4-49 and whose theory appears in the section “Enclosures with More Than Two Surfaces” in Ref. 1.

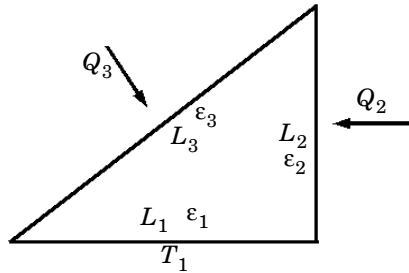


Figure 4-49: Problem sketch for theoretical analysis of radiation in a triangular cavity.

The theoretical model considers only the three boundaries that form the cavity. On these boundaries, the model either holds the temperature at a constant value or specifies a heat flux. Note that this approach differs somewhat from the model, which sets temperatures and heat fluxes on the rectangles’ *outer* boundaries.

The COMSOL Multiphysics model shows that the temperatures and heat fluxes are nearly equal on the inner and outer boundaries of the rectangles. Therefore, the COMSOL Multiphysics and the theoretical model show good agreement.

Using the notation from Figure 4-49 with the same assumptions as for the model, you obtain the following lengths and emissivities:

$$L_1 = 4 \text{ m}, \quad \varepsilon_1 = 0.4$$

$$L_2 = 3 \text{ m}, \quad \varepsilon_2 = 0.6$$

$$L_3 = 5 \text{ m}, \quad \varepsilon_3 = 0.8$$

The boundary conditions define either the temperature or the heat flux on each boundary. Now apply the same boundary conditions as in the problem where you set values for  $T_1$ ,  $Q_2$ , and  $Q_3$ . You must, however, make a small adjustment for  $T_1$  because the theoretical configuration sets it on the outer boundary and not on the cavity side.  $T_1$  is therefore slightly higher than 300 K in the theoretical analysis:

$$T_1 = 307 \text{ K}$$

$$Q_2 = q_2 L_2 = 2000 \frac{\text{W}}{\text{m}} \times 3 \text{ m} = 6000 \text{ W}$$

$$Q_3 = q_3 L_3 = 1000 \frac{\text{W}}{\text{m}} \times 5 \text{ m} = 5000 \text{ W}.$$

You describe the heat flux from a boundary with the following two equations:

$$Q_i = L_i \frac{\varepsilon_i}{1 - \varepsilon_i} (\sigma T_i^4 - J_i),$$

$$Q_i = L_i \sum_k F_{ik} (J_i - J_k).$$

For a triangular cavity, the following equation gives the view factor between surface  $i$  and surface  $j$ :

$$F_{ij} = \frac{L_i + L_j - L_k}{2L_i}.$$

Substituting this expression for the view factor into the second equation for the heat fluxes results in a linear systems of six equations. The six unknowns for this particular problem are  $J_1, J_2, J_3, Q_1, T_2,$  and  $T_3,$  and you would like to compare the three last to those from the solution. Solving the linear system yields the following values:  $Q_1 = -11,000 \text{ W/m}, T_2 = 641 \text{ K},$  and  $T_3 = 600 \text{ K}.$

To compare these results with the model, you must find the total heat flux through the cavity's bottom (horizontal) boundary and also calculate the average temperatures on the other two boundaries. Compute these values by integrating along the boundaries.

First find the total heat flux through the horizontal boundary in the model.

- 1 From the **Postprocessing** menu select **Boundary Integration**.
- 2 From the **Boundary selection** list select Boundary 6, then go to the **Predefined quantities** list and select **Normal conductive heat flux**. Click **Apply**.

The total heat flux now appears in the lower left corner of the main graphical user interface. The result should be approximately  $-10,980 \text{ W/m}.$

Next find the mean temperatures on the vertical and inclined boundaries of the cavity.

- 1 While still in the **Boundary Integration** dialog box, go to the **Boundary selection** list and select **9** (the vertical boundary). In the **Expression** edit field enter  $T/3$  [m]; division by three meters corresponds to the boundary length.
- 2 Click **Apply**. The resulting temperature, read from the bottom of the user interface, is approximately 645 K.
- 3 Similarly, to obtain the mean temperature for the inclined boundary, go to the **Boundary selection** list and choose number 3. In the **Expression** edit field you this time enter  $T/5$  [m].
- 4 Click **Apply**. The resulting temperature is approximately 601 K.

The following table compares the results from COMSOL Multiphysics with the theoretical values:

QUANTITY	COMSOL MULTIPHYSICS	THEORY
$Q_1$	-10,980 W/m	-11,000 W/m
$T_2$	645 K	641 K
$T_3$	601 K	600 K

The differences are quite small. Also note that the theoretical model includes some simplifications. For example, it assumes that heat fluxes and temperatures are constant along each boundary. It also assumes a constant view factor for each pair of boundaries, whereas COMSOL Multiphysics computes a local view factor at each point of the cavity.

# Building and Solving a Model with Highly Conductive Layers

This section demonstrates how to use the highly conductive layer feature of the General Heat Transfer application mode.

## *Model Definition—Copper Layer on Silica Glass*

This example constructs a 2D time-dependent model of a silica glass block that is coated with a thin copper layer. Figure 4-50 shows the model geometry and boundary conditions.

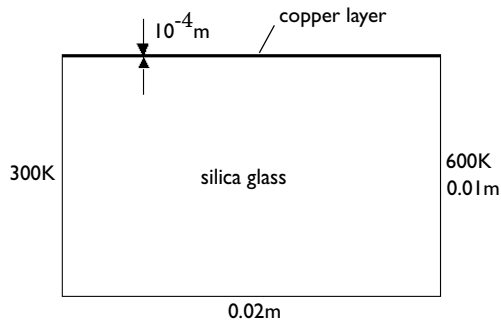


Figure 4-50: Model geometry for a silica block with a copper layer.

The model sets the initial temperature to 300 K. The following table shows the thermal properties for silica glass and copper:

QUANTITY	SILICA GLASS	COPPER	DESCRIPTION
$\rho$	2203 kg/m <sup>3</sup>	8700 kg/m <sup>3</sup>	Density
$C_p$	703 J/(kg·K)	385 J/(kg·K)	Specific heat
$k$	1.38 W/(m·K)	400 W/(m·K)	Thermal conductivity

The thermal conductivity of copper is much higher than that for silica glass. That fact, and also considering that the copper layer is thin, makes it possible to model it as a highly conductive layer. Using this feature you do not need to resolve the thin layer with an extremely fine mesh, which would require a significantly larger computation time.

## Results

Figure 4-51 shows the temperature field after 1 s, 3 s, 10 s, and 60 s. The results show that the temperature rise is faster in the copper layer than in the silica glass. After 60 s the temperature field has almost reached steady state, and the temperature field varies linearly between the two vertical boundaries.

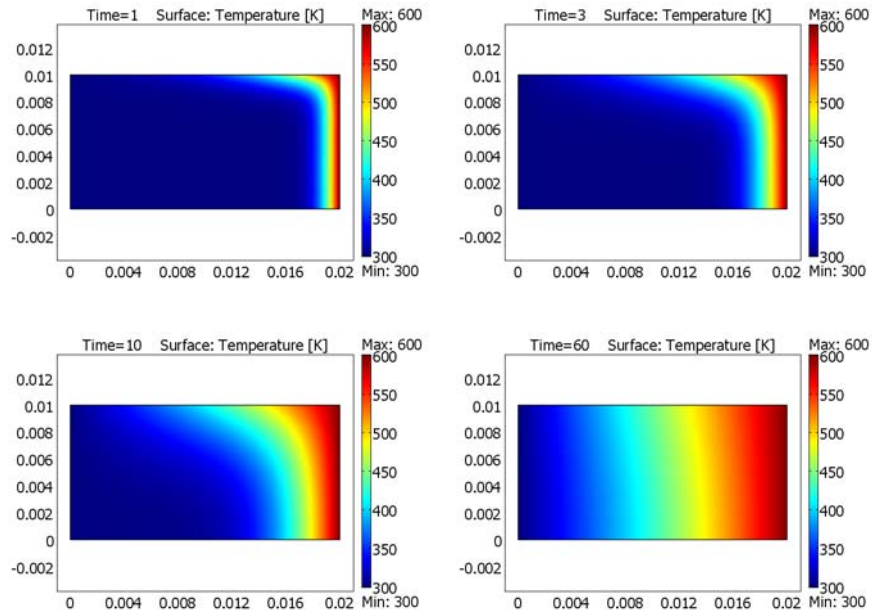


Figure 4-51: Temperature field after 1, 3, 10, and 60 seconds.

---

**Model Library path:** Heat\_Transfer\_Module/Tutorial\_Models/copper\_layer

---

You can compare the results in this model with a model where the copper layer has been meshed instead of using the highly conductive layer feature. This model produces the same results, but requires a denser mesh and longer computation time.

The reference model can be found at the following location:

---

**Model Library path:** Heat\_Transfer\_Module/Tutorial\_Models/  
copper\_layer\_meshed

---

### *Modeling Using the Graphical User Interface*

---

#### **MODEL NAVIGATOR**

- 1 Open the **Model Navigator** and click the **New** tab.
- 2 From the **Space dimension** list select **2D**.
- 3 In the list of application modes select **Heat Transfer Module>General Heat Transfer>Transient analysis**.
- 4 Click **OK**.

#### **GEOMETRY MODELING**

- 1 Shift-click the **Rectangle/Square** button on the Draw toolbar.
- 2 In the dialog box that appears, enter these rectangle properties; when finished, click **OK**.

OBJECT DIMENSIONS	EXPRESSION
Width	0.02
Height	0.01

- 3 Click the **Zoom Extents** button on the Main toolbar.

#### **PHYSICS SETTINGS**

##### *Boundary Conditions*

- 1 From the **Physics** menu select **Boundary Settings**.
- 2 From the **Boundary selection** list select **1**. From the **Boundary condition** list select **Temperature**, then in the **T<sub>0</sub>** edit field type 300.
- 3 Repeat this procedure for Boundary 4, but in the **T<sub>0</sub>** edit field type 600.
- 4 From the **Boundary selection** list select **3**, then click the **Highly Conductive Layer** tab.
- 5 Select the **Enable heat transfer in highly conductive layer** check box.
- 6 Click the **Load** button, go to the **Materials** list and select **Basic Material Properties>Copper**. Click **OK**.



7 In the  $d_s$  edit field type  $1e-4$ , then click **OK**.

#### *Subdomain Settings*

- 1 From the **Physics** menu select **Subdomain Settings**.
- 2 In the **Subdomain selection** list choose number **1**.
- 3 Click the **Load** button. In the **Materials** list select **Basic Material Properties>Silica glass**. Click **OK**.
- 4 Click the **Init** tab. In the **T(t<sub>0</sub>)** edit field type **300**, then click **OK**.

#### **COMPUTING THE SOLUTION**

- 1 From the **Solve** menu select **Solver Parameters**.
- 2 Examine the **Solver** list and verify that **Time dependent** is selected.
- 3 In the **Times** edit field type **0:1:60**, then click **OK**.
- 4 Click the **Solve** button on the Main toolbar.

#### **POSTPROCESSING AND VISUALIZATION**

The default plot shows the temperature field after 60 s. The following steps describe how to view the temperature field at other times.

- 1 From the **Postprocessing** menu select **Plot Parameters**.
- 2 Click the **General** tab.
- 3 From the **Solution at time** list select the desired solution time. Click **Apply**.

You can also generate an animation of the process in this way:

- 1 In the **Plot Parameters** dialog box click the **Animate** tab.
- 2 In the **Solutions to use** list make sure all the times are selected.
- 3 Click the **Start Animation** button.

# Building and Solving an Out-of-Plane Heat Transfer Model

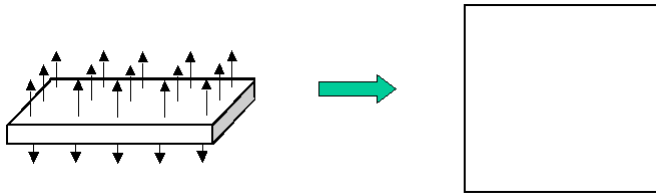
This section demonstrates how to use the out-of-plane heat transfer feature of the General Heat Transfer application mode.

This example models heat transfer in a thin rectangular metal plate. Because the plate's thickness is only 1/100 of its length and width, you simulate the process using a 2D approximation (that is, assume the temperature is constant along the thickness). The plate has a fixed temperature at one end and is isolated at the other end. A surrounding liquid cools the plate by convection. In addition, the model considers surface-to-ambient radiation.

## *Model Definition—Thin Plate*

---

The plate measures 1-by-1 m, with a thickness of 0.01 m. Figure 4-52 describes the simplification from 3D to 2D.



*Figure 4-52: Model geometry reduction from 3D to 2D.*

The stationary heat transfer equation for this 2D problem is

$$\nabla \cdot (-d_z k \nabla T) = d_z (Q + q_s T) + h_u (T_{\text{ext}, u} - T) + h_d (T_{\text{ext}, d} - T) + \epsilon_u \sigma (T_{\text{amb}, u}^4 - T^4) + \epsilon_d \sigma (T_{\text{amb}, d}^4 - T^4)$$

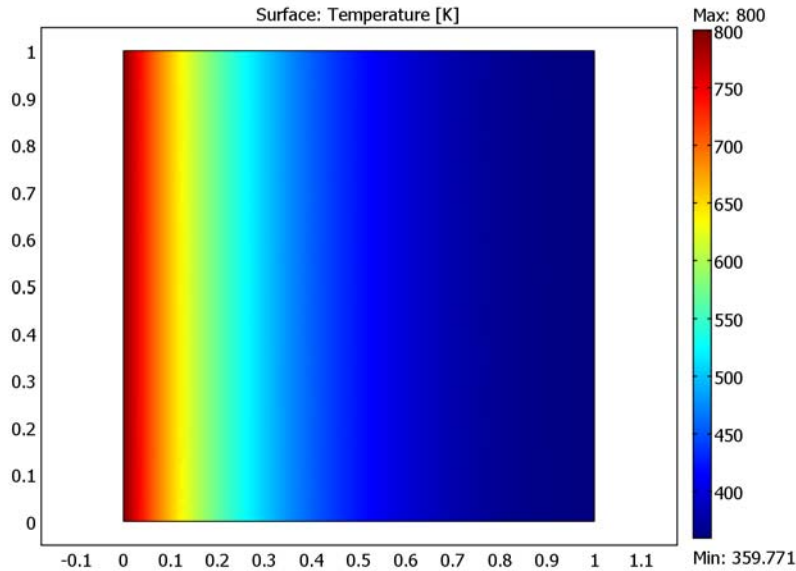
In this case the problem is symmetric, making the upside constants equal to the downside ones. For constants set  $h$ , the heat transfer film coefficient, to  $10 \text{ W}/(\text{m}^2 \cdot \text{K})$ , and set the emissivity,  $\epsilon$ , to 0.5. The ambient and external temperatures,  $T_{\text{amb}}$  and  $T_{\text{ext}}$ , are both 300 K. This model also uses the default material properties for the application mode, and they correspond to copper. The variable  $d_z$  denotes the plate's thickness.

Finally, set the left boundary to a fixed temperature of 800 K, while the other boundaries are thermally insulated.

## Results

---

Figure 4-53 shows the steady-state temperature distribution.



*Figure 4-53: Stationary temperature distribution in the plate.*

You can also plot the out-of-plane fluxes for the convection and radiation.

It is interesting to compare the results from this 2D approximation with the equivalent full 3D model. Figure 4-54 shows the temperature along the bottom edge of the 2D model and the corresponding 3D result. The results are in such good agreement that except for the colors on the computer screen you cannot even distinguish between the two curves.

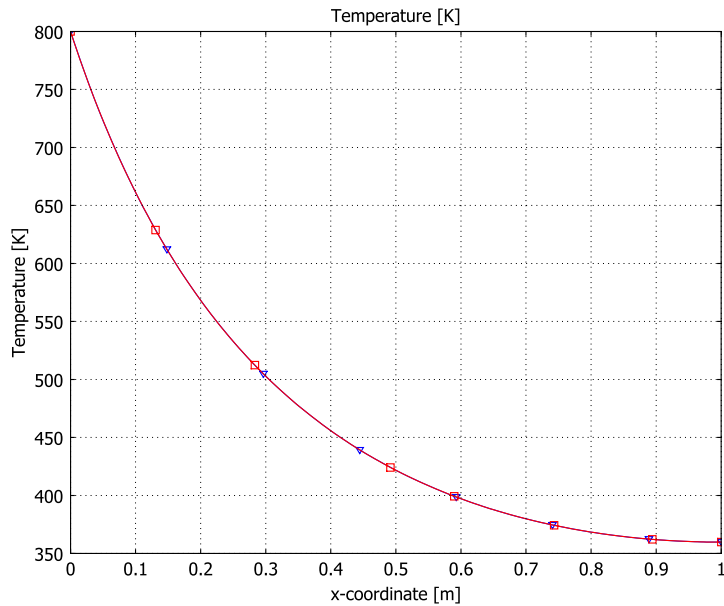


Figure 4-54: Comparison of the temperature (from left to right in the plate) from both the 2D out-of-plane model (blue/triangles) and the full 3D model (red/squares). The curves coincide to a very high degree, making it difficult to tell them apart.

This agreement with the 3D result means that the approximation you made about zero temperature gradients along the plate thickness is accurate. Thus, for slender geometries, the out-of-plane application mode significantly reduces the numerical load with negligible effects on accuracy.

---

**Model Library path:** Heat\_Transfer\_Module/Tutorial\_Models/  
thin\_plate\_2d

---



---

**Model Library path:** Heat\_Transfer\_Module/Tutorial\_Models/  
thin\_plate\_3d

---

### MODEL NAVIGATOR

- 1 Open the **Model Navigator** and click the **New** tab. Verify in the **Space dimension** list that you have selected **2D**.
- 2 From the list of application modes select  
**Heat Transfer Module>General Heat Transfer>Out-of-plane heat transfer**.
- 3 Click **OK**.

### GEOMETRY MODELING

- 1 Press the Shift key and click the **Rectangle/Square** button in the Draw toolbar.
- 2 In the dialog box that appears, accept the default rectangle properties by clicking **OK**.

OBJECT DIMENSIONS	EXPRESSION
Width	1
Height	1
Base	Corner
x	0
y	0

- 3 Click **Zoom Extents** on the Main toolbar.

### PHYSICS SETTINGS

#### *Subdomain Settings—General Heat Transfer*

- 1 From the **Physics** menu select **Subdomain Settings**.
- 2 From the **Subdomain selection** list choose **1**. On the **General** page, accept the default material properties (that is, make no changes).
- 3 On the **Out-of-Plane** page, enter the following settings; when done, click **OK**.

PROPERTIES	VALUES
$h_u, h_d$	10, 10
$T_{ext,u}, T_{ext,d}$	300, 300
$\epsilon_u, \epsilon_d$	0.5, 0.5
$T_{amb,u}, T_{amb,d}$	300, 300
$d_z$	0.01

### Boundary Conditions—General Heat Transfer

- 1 Go to the **Physics** menu and select **Boundary Settings**.
- 2 From the **Boundary selection** list choose 1. Enter the following settings; when done, click **OK**.

Boundary condition	Temperature
$T_0$	800
Radiation type	None

### MESH GENERATION

Click the **Initialize Mesh** button on the Main toolbar.

### COMPUTING THE SOLUTION

Click the **Solve** button on the Main toolbar.

### POSTPROCESSING AND VISUALIZATION

The default plot shows the temperature distribution along the plate (Figure 4-53).

To generate the plot in Figure 4-54 follow these steps:

- 1 From the **Postprocessing** menu select **Domain Plot Parameters**.
- 2 Click the **Line/Extrusion** tab. From the **Boundary selection** list select number 2.
- 3 Click the **Line Settings** button at the bottom of the dialog box. From the **Line marker** list, select **Triangle**. Click **OK** to close the **Line Settings** dialog box.
- 4 Click **OK** to close the **Domain Plot Parameters** dialog box and generate the first trace.
- 5 Save this model by clicking the **Save** button on the Main toolbar and giving it a name of your choice.
- 6 Open the equivalent 3D model by going to the **File** menu and selecting **New**. The **Model Navigator** opens up. Click the **Model Library** tab and from the list select **Heat Transfer Module>Tutorial Models>thin plate 3D**. Click **OK**.
- 7 From the **Postprocessing** menu select **Domain Plot Parameters**.
- 8 On the **General** page, go to the **Plot in** list at the bottom of the dialog box and select **Figure 1**. In addition, select the **Keep current plot** check box to plot in Figure 1 without deleting the existing trace.
- 9 Click the **Line/Extrusion** tab. From the **Edge selection** list, select 5.
- 10 In the **x-axis data** area, click first the lower option button and then the **Expression** button.

- 11** In the **X-Axis Data** dialog box, type  $x$  in the **Expression** edit field. Click **OK**.
- 12** Click the **Line Settings** button. From the **Line color** list, select **Color**.
- 13** From the **Line marker** list, select **Square**. Click **OK**.
- 14** Click **OK** to create the dual-trace plot.



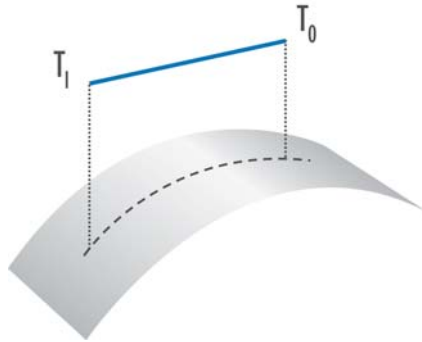


## Thin Conductive Shell

Thin conductive shells are useful for modeling heat transfer in structure. This chapter describes how to use the Thin Conductive Shell application mode.

# The Thin Conductive Shell Application Mode

The Thin Conductive Shell application mode is suitable for solving thermal-conduction problems in thin structures.



This application modes support two types of heat transfer: conduction and out-of-plane heat transfer. Because the thermal conductivity across the shell thickness is very large or the shell is so thin, you can assume constant temperature through the shell thickness. The dependent variable is the temperature  $T$ . The application mode is defined on 3D faces. The governing equation for heat transfer in thin shells is:

$$\delta_{ts} \rho d_z C_p \frac{\partial T}{\partial t} + d_z \nabla_T \cdot (-k_g \nabla_T T) = Q + h_u (T_{\text{ext, u}} - T) + h_d (T_{\text{ext, d}} - T) + \epsilon_u \sigma (T_{\text{amb, u}}^4 - T^4) + \epsilon_d \sigma (T_{\text{amb, d}}^4 - T^4) + q_u + q_d$$

This section shows how to work with this application mode by solving this equation. The discussion describes the application mode in this order:

- Properties
- Scalar Variables
- Material
- Out-of-Plane Heat Transfer
- Edge Conditions

## Properties

---

Open the **Application Mode Properties** dialog box by going to the **Physics** menu and selecting **Properties**.

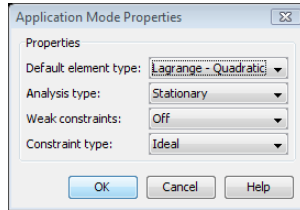


Figure 5-1: The Application Mode Properties dialog box.

In the resulting dialog box you control various global settings for the model:

- **Default element type** When you make a selection from this list of available elements it becomes the default on all new boundaries but does not affect faces you already created. Available elements are:
  - **Lagrange - Linear**
  - **Lagrange - Quadratic** (default)
  - **Lagrange - Cubic**
  - **Lagrange - Quartic**
  - **Lagrange - Quintic**
- **Weak constraints:** Controls whether or not weak constraints are active. Weak constraints introduce Lagrange multipliers, which provide extremely accurate flux quantities; the standard way of evaluating them involves taking derivatives on boundaries, which is a mesh-dependent and less accurate method. When weak constraints are enabled, all constraints are weak by default, but it is possible to change this setting for individual domains.
- **Constraint type:** Constraints can be ideal or non-ideal (see “Ideal vs. Non-Ideal Constraints” on page 359 in the *COMSOL Multiphysics Modeling Guide*).

## Scalar Variables

---

This application mode only has one scalar variable, the Stefan-Boltzmann constant,  $\sigma$ , which specifies the out-of-plane ambient radiation

$$\epsilon_u \sigma (T_{\text{amb,u}}^4 - T^4)$$

where  $T_{\text{amb,u}}$  is the ambient temperature, and  $\epsilon_u$  the surface emissivity on the upside of the shell.

To open the **Scalar Variables** dialog box, go to the **Physics** menu and select **Scalar Variables**.

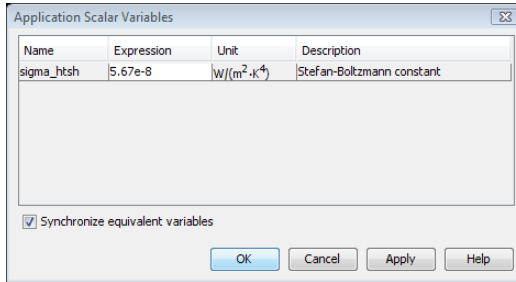


Figure 5-2: The Application Scalar Variables dialog box.

The following table shows the unit and default value in SI units of the Stefan-Boltzmann constant:

PROPERTY	VARIABLE	DESCRIPTION	DEFAULT VALUE	SI UNITS
$\sigma$	sigma_htsh	Stefan-Boltzmann constant	5.67e-8	W/(m <sup>2</sup> ·K <sup>4</sup> )

Change the default value only if the model uses no base unit system and the data that you enter are based on a non-SI unit system.

### *Material*

You specify the material properties and heat source on the **Physics** page in the **Boundary Settings** dialog box (Figure 5-3).

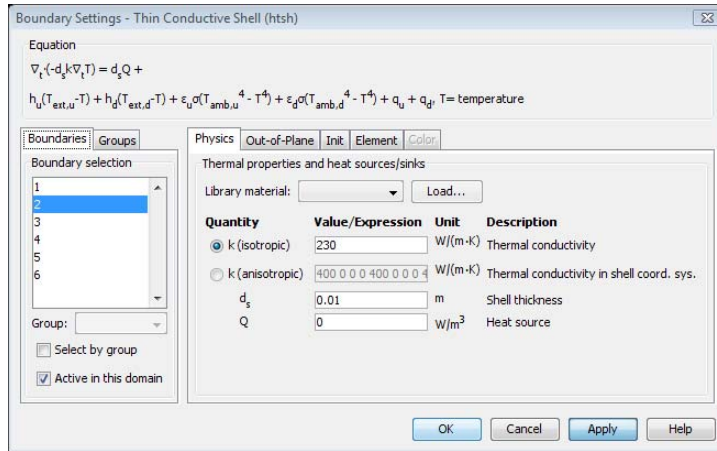


Figure 5-3: The Boundary Settings dialog box.

The following table contains the shell properties together with their default values.

QUANTITY	VARIABLE	DESCRIPTION	DEFAULT VALUE	SI UNIT
$\delta_{ts}$	Dts	Time-scaling coefficient	1	-
$k$	k	Thermal conductivity	400	W/(m·K)
$k_{ij}$	$k_{ij}$	Thermal conductivity tensor comp. in shell coord. sys.	400	W/(m·K)
$d_s$	ds	Shell thickness	0.01	m
$\rho$	rho	Density	8700	kg/m <sup>3</sup>
$C_p$	Cp	Heat capacity	385	J/(kg·K)
$Q$	Q	Heat source	0	W/m <sup>3</sup>

**Time-scaling coefficient** You rarely change the value of  $\delta_{ts}$ , but you can use it to change the time scale, for example, from seconds to minutes by setting it to 1/60.

---

**Note:** If you change the value of the time-scaling coefficient, any quantity involving time derivatives or explicit references to time must be reinterpreted accordingly—not only in the Thin Conductive Shell application mode, but also in any coupled physics. You must multiply all occurrences of time derivatives with the time-scaling coefficient, and convert all uses of time (as a variable) to the consistent base unit system in order to get consistent units in your equations and postprocessing expressions.

---

**Thermal conductivity** The variable  $k$  describes the relationship between the heat flux vector  $\mathbf{q}$  and the temperature gradient  $\nabla T$  as in

$$\mathbf{q} = -k\nabla T,$$

which is *Fourier's law of heat conduction*. The conductivity can be either isotropic or an anisotropic tensor. In the latter case, you specify the tensor components in the *shell local coordinate system*, which is defined from the geometric tangent and normal vectors. The local  $x$  direction is the surface tangent vector  $\mathbf{t}_1$  and the local  $z$  direction is the normal vector  $\mathbf{n}$ . Their cross product defines the third orthogonal direction such that

$$\begin{aligned}\mathbf{e}_{xl} &= \mathbf{t}_1 \\ \mathbf{e}_{yl} &= \mathbf{e}_{zl} \times \mathbf{e}_{xl} = \mathbf{n} \times \mathbf{t}_1 \\ \mathbf{e}_{zl} &= \mathbf{n}.\end{aligned}$$

is an orthonormal basis aligned with the shell. You can then express the transformation matrix from the shell's local coordinate system to the global system using the local basis vectors' components in the global system:

$$A = \begin{bmatrix} e_{xlx} & e_{yly} & e_{zlx} \\ e_{xly} & e_{yly} & e_{zly} \\ e_{xlz} & e_{yly} & e_{zly} \end{bmatrix}$$

The global thermal-conductivity tensor used in the equations consists of the local tensor entered in the dialog box transformed with  $A$ :

$$k_g = A k A^t$$

Since the conductivity in the direction normal to the shell is from the outset taken to be sufficiently high to assume a constant temperature across the thickness, its actual value does not matter. Therefore, you can safely ignore the third row and third column of the thermal conductivity tensor in local coordinates.

---

**Note:** You cannot use anisotropic thermal conductivity together with a moving frame, since the Moving Mesh application mode does not define the tangential direction vectors.

---

**Shell thickness**  $d_s$  defines the shell thickness.

**Density**  $\rho$  specifies the material's density.

**Heat capacity**  $C_p$  describes the amount of heat energy required to produce a unit temperature change in a unit mass.

**Heat source**  $Q$  describes internal heat generation within the shell. It is not a material property.

## Out-of-Plane Heat Transfer

You specify the out-of-plane heat transfer using the **Out-of-Plane** page in the **Boundary Settings** dialog box (Figure 5-4).

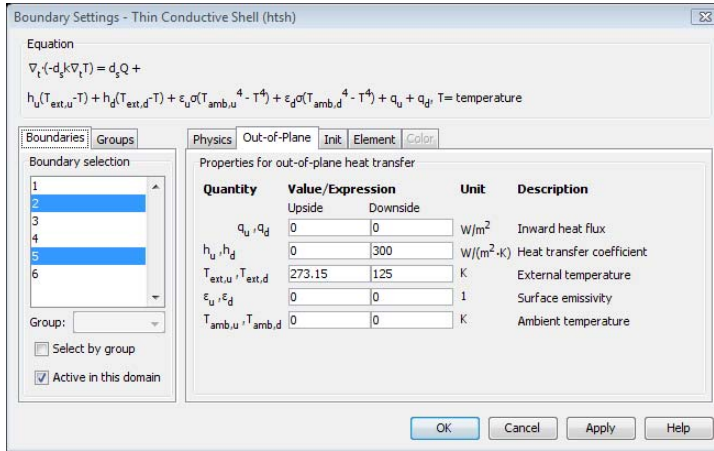


Figure 5-4: The Out-of-Plane page.

The out-of plane heat transfer properties are:

QUANTITY	VARIABLE	DESCRIPTION	DEFAULT VALUE	SI UNITS
$q_u$	qu	Out-of-plane inward heat flux, upside	0	W/m <sup>2</sup>
$q_d$	qd	Out-of-plane inward heat flux, downside	0	W/m <sup>2</sup>
$h_u$	hu	Out-of-plane heat transfer coefficient, upside	0	W/(m <sup>2</sup> ·K)
$h_d$	hd	Out-of-plane heat transfer coefficient, downside	0	W/(m <sup>2</sup> ·K)
$T_{ext,u}$	Textu	Out-of-plane external temperature, upside	0	K
$T_{ext,d}$	Textd	Out-of-plane external temperature, downside	0	K
$\epsilon_u$	epsilonu	Out-of-plane surface emissivity, upside	0	-
$\epsilon_d$	epsilond	Out-of-plane surface emissivity, downside	0	-



QUANTITY	VARIABLE	DESCRIPTION	DEFAULT VALUE	SI UNITS
$T_{amb,u}$	Tambu	Out-of-plane ambient temperature, upside	0	K
$T_{amb,d}$	Tambd	Out-of-plane ambient temperature, downside	0	K

You work with these quantities to model heat flux from the surroundings into the shell from either its upside or the downside. Because there is no temperature difference through the shell thickness, it makes no difference from which side the flux is coming. The total out-of-plane flux is:

$$q_{total} = q_u + q_d + h_u(T_{ext,u} - T) + h_d(T_{ext,d} - T) + \epsilon_u \sigma (T_{amb,u}^4 - T^4) + \epsilon_d \sigma (T_{amb,d}^4 - T^4).$$

### Edge Conditions

You specify edge conditions in the **Edge Settings** dialog box (Figure 5-5).

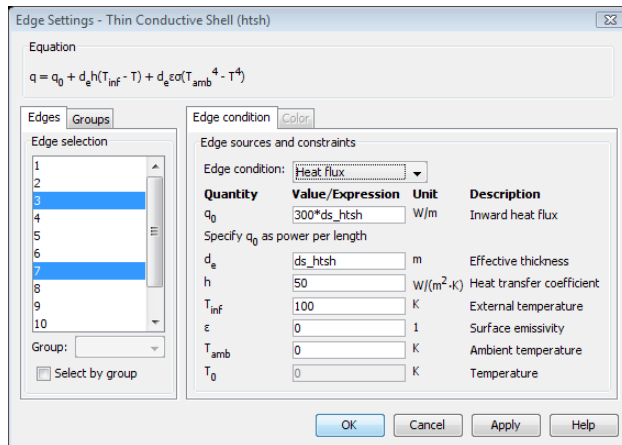


Figure 5-5: The Edge Settings dialog box.

The available edge conditions are:

EDGE CONDITION	EXPRESSION	DESCRIPTION
Heat flux	$q = q_0 + d_e h (T_{\text{inf}} - T) + d_e \varepsilon \sigma (T_{\text{amb}}^4 - T^4)$	Heat flux through inward specified flux, convective heat transfer, and radiation
Thermal insulation	$q = 0$	Thermal insulation or symmetry
Temperature	$T = T_0$	Prescribed temperature

The quantities specifying these different edge conditions are:

QUANTITY	VARIABLE	DESCRIPTION	DEFAULT VALUE	SI UNIT
$q_0$	q0	Inward heat flux	0	W/m
$d_e$	de	Effective edge thickness	ds	m
$h$	h	Heat transfer coefficient	0	W/(m <sup>2</sup> ·K)
$T_{\text{inf}}$	Tinf	External temperature	0	K
$\varepsilon$	epsilon	Edge emissivity	0	-
$T_{\text{amb}}$	Tamb	Ambient temperature	0	K

Note that the inward heat flux  $q_0$  is specified as power per length. The effective edge thickness  $d_e$  is the height of the part of the edge that is exposed to the ambient surroundings. On a free edge,  $d_e$  is equal to the shell thickness  $d_s$ , which is also the default value.

# Building and Solving a Thin Conductive Shell Model

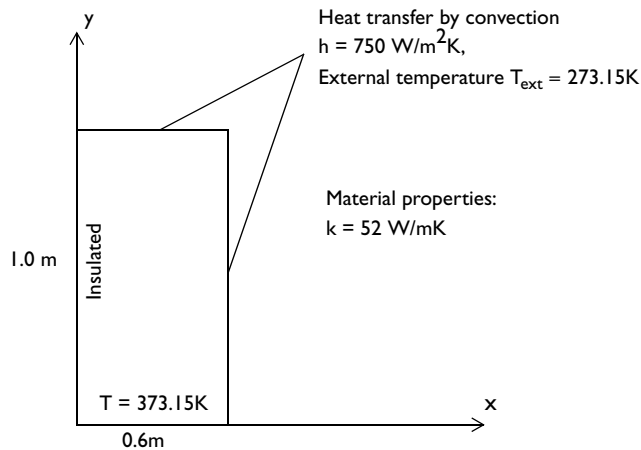
The following example illustrates how to build and solve a model using the Thin Conductive Shell application mode. This model is a 2D NAFEMS benchmark (Ref. 1), which the author has transformed to 3D. The author also used the unit kelvin (K) for temperature, as opposed to the NAFEMS benchmark model, which uses degrees Celsius.

## *Model Definition—Shell Conduction*

---

### **GEOMETRY**

Figure 5-6 describes the 2D benchmark example.



*Figure 5-6: A 2D benchmark example for a thin conductive shell.*

The 3D model bends this plate so that it becomes a quarter of a cylinder (Figure 5-7).

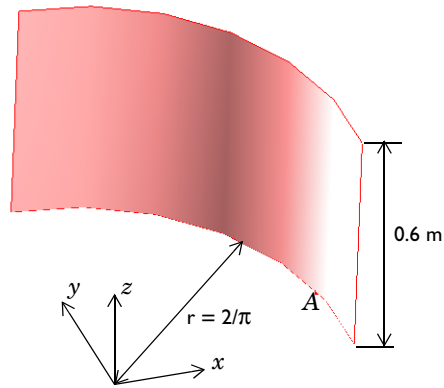


Figure 5-7: The 3D geometry based on the 2D model.

### Results

The temperature at point  $A$  in Figure 5-7 is in close agreement with that from the NAFEMS benchmark (Ref. 1) as is evident from this table:

RESULT	COMSOL MULTIPHYSICS	NAFEMS
T (at A)	291.42 K (18.27 °C)	291.40 K (18.25 °C)

Figure 5-8 shows the temperature distribution.

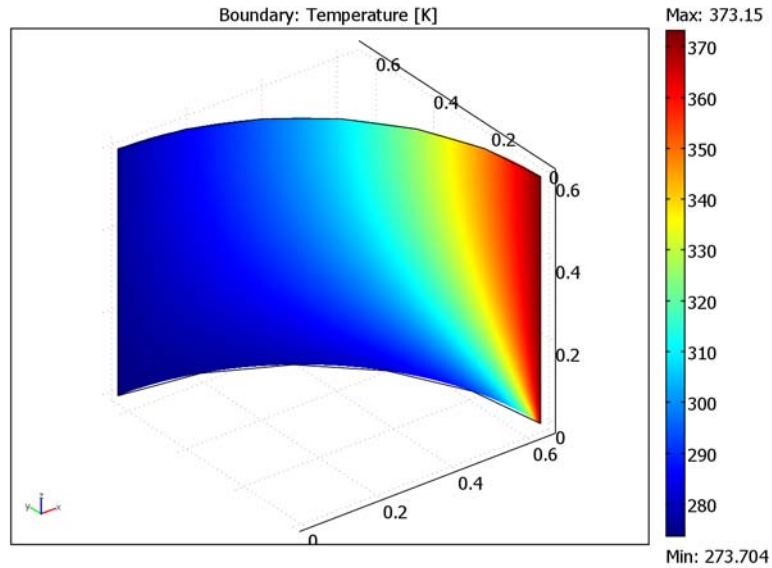


Figure 5-8: The resulting temperature field of the 3D model.

### Reference

---

I. J.A. Casey and G.B Simpson, "Two-dimensional steady state," *Benchmark Tests for Thermal Analysis*, NAFEMS, 1986, Test 10, p. 2.9.

---

**Model Library path:** Heat\_Transfer\_Module/Tutorial\_Models/  
shell\_conduction

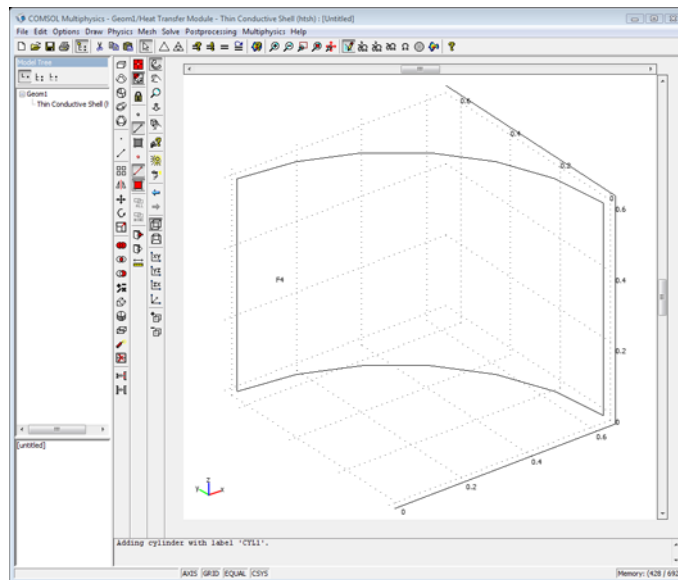
---

### MODEL NAVIGATOR

- 1 Open the **Model Navigator**, click the **New** tab, and from the **Space dimension** list select **3D**.
- 2 From the list of application modes select **Heat Transfer Module>Thin Conductive Shell>Steady-state analysis**. Click **OK**.

### GEOMETRY MODELING

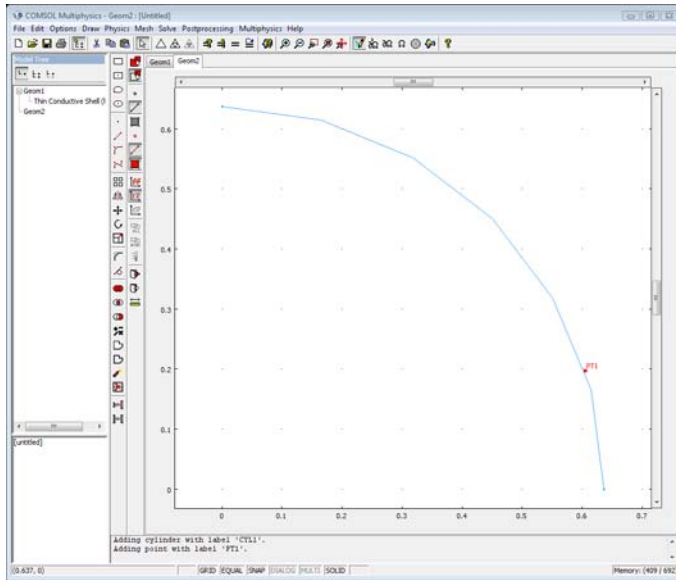
- 1 Click the **Cylinder** button on the Draw toolbar.
- 2 To create an open cylinder, go to the **Style** area and select **Face**.
- 3 In the **Radius** edit field enter  $2/\pi$ , and in the **Height** edit field enter 0.6. Click **OK**.
- 4 Click the **Split Object** button on the Draw toolbar.
- 5 Using the mouse select faces F1, F2, and F3. Press the Delete key to remove those three faces, leaving just one.
- 6 Click the **Zoom Extents** button on the Main toolbar.



*The geometry after removing three faces of a cylinder.*

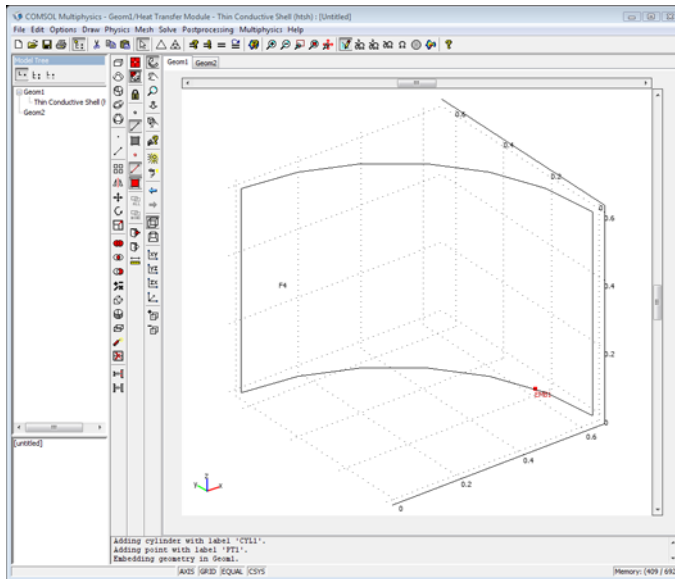
- 7 From the **Draw** menu select **Work-Plane Settings**.

- 8 In the resulting dialog box click **OK**, which brings up the default work plane.
- 9 Click the **Point** button on the Draw toolbar. Then click the cursor on the lower right end of the projected 3D face.
- 10 Click the **Rotate** button on the Draw toolbar.
- 11 In the **Rotation angle** area, enter  $0.2 \times 180$  in the  $\alpha$  edit field. Click **OK**.



*The work plane with the repositioned point.*

- From the Draw menu select **Embed** then click **OK**. This step embeds the point where you compare the calculated solution with the benchmark.



*The embedded measurement point shown on the 3D geometry.*

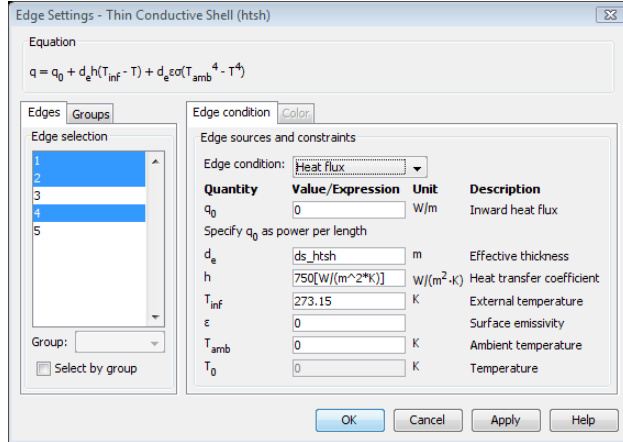
## PHYSICS SETTINGS

### Edge Settings

- From the **Physics** menu select **Edge Settings**.
- Specify conditions on the edges as follows; when done, click **OK**.

SETTINGS	EDGES 1, 2, 4	EDGE 3	EDGE 5
Edge condition	Heat flux	Insulation/continuity	Temperature
$d_e$	ds_htsh		
$h$	750[W/(m <sup>2</sup> *K)]		
$T_{inf}$	273.15		
$T_0$			373.15





*Edge Settings dialog box with settings for Edges 1, 2, and 4.*

### Boundary Conditions

- 1 From the **Physics** menu select **Boundary Settings**.
- 2 In the resulting dialog box go to the **Boundary selection** list and choose **1**. In the edit field for **k (isotropic)** enter 52.
- 3 Click **OK**.

### MESH GENERATION

Click the **Initialize Mesh** button on the Main toolbar to mesh the geometry with the default settings.

### COMPUTING THE SOLUTION

Click the **Solve** button on the Main toolbar.

### POSTPROCESSING AND VISUALIZATION

The default plot is the boundary plot of the temperature  $T$  (see Figure 5-8).

The following steps show how to obtain the temperature at the benchmark verification point:

- 1 From the **Postprocessing** menu choose **Point Evaluation**.
- 2 In the **Point selection** list select **3**.

**3** Click **OK**.

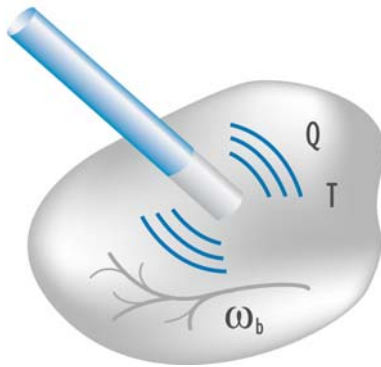
COMSOL Multiphysics displays the value of the temperature at Point 3, about 291.4 K in the message log at the bottom of the user interface.

## The Bioheat Equation

This chapter describes the Bioheat Equation application mode for modeling heat transfer within tissue using the bioheat equation.

# The Bioheat Equation Application Mode

This section describes the Bioheat Equation application mode in the Heat Transfer Module, which is designed to model heat transfer within tissue.



This application mode uses Pennes' approximation to represent heat sources from metabolism and blood perfusion. In addition, it adds a spatial heating source to the equation. This third source is useful when modeling how an external source, such as an electromagnetic field, heats tissue.

## *Variables and Space Dimensions*

---

The Bioheat Equation application mode is available in 2D and 3D as well as for axisymmetric models using cylindrical coordinates in 2D. The dependent variable is the temperature,  $T$ .

## *PDE Formulation*

---

The *bioheat equation* is similar to the heat equation for heat conduction except that it splits the heat-source term on the right-hand side into three terms that are of interest in bioheat transfer problems. Further, the material properties in this equation are better suited to the study of heat transfer in tissue. The equation thus takes this form:

$$\delta_{ts} \rho C \frac{\partial T}{\partial t} + \nabla \cdot (-k \nabla T) = \rho_b C_b \omega_b (T_b - T) + Q_{\text{met}} + Q_{\text{ext}}$$

Where on the left-hand side:

- $\delta_{ts}$  is a time-scaling coefficient (dimensionless).
- $\rho$  is the *tissue density* ( $\text{kg}/\text{m}^3$ ).
- $C$  is the *specific heat of tissue* ( $\text{J}/(\text{kg}\cdot\text{K})$ ).
- $k$  is the tissue's *thermal conductivity tensor* ( $\text{W}/(\text{m}\cdot\text{K})$ ).

On the right side of the equation,  $\rho_b C_b \omega_b (T_b - T)$  is a source term for the blood perfusion where:

- $\rho_b$  is the *density of blood* ( $\text{kg}/\text{m}^3$ ).
- $C_b$  is the *specific heat of blood* ( $\text{J}/(\text{kg}\cdot\text{K})$ ).
- $\omega_b$  is the *blood perfusion rate* ( $\text{m}^3/(\text{m}^3\cdot\text{s})$ ).
- $T_b$  is the *arterial blood temperature* (K).

Finally, examine the second and third terms on the right side where:

- $Q_{\text{met}}$  is the *heat source from metabolism* ( $\text{W}/\text{m}^3$ ).
- $Q_{\text{ext}}$  is the *spatial heat source* ( $\text{W}/\text{m}^3$ ).

For a steady-state problem, the temperature does not change over time, and the first term in the bioheat equation disappears.

## Subdomain Settings

Figure 6-1 shows the dialog box for the **Subdomain Settings** of the Bioheat Equation application mode; here you input properties for the bioheat equation.

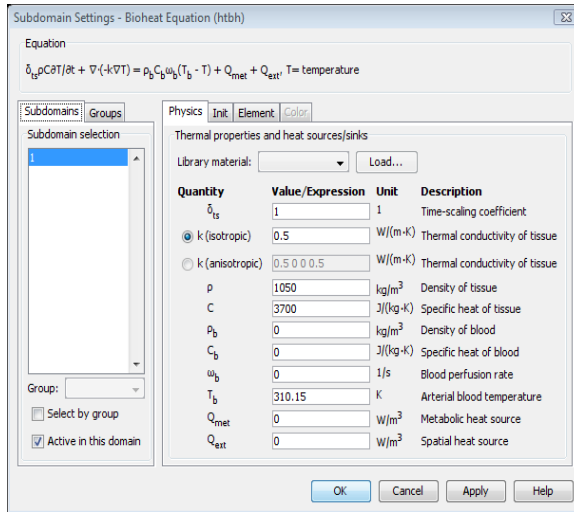


Figure 6-1: The Subdomain Settings dialog box.

The subdomain quantities are:

QUANTITY	VARIABLE	DESCRIPTION
$\delta_{ts}$	Dts	Time-scaling coefficient
$\rho$	rho	Tissue density
$C$	C	Specific heat of tissue
$k$	k	Thermal conductivity of tissue
$\underline{k}$		Tissue thermal conductivity tensor
$k_{ij}$	$kx_ix_j$	Tissue thermal conductivity tensor, $x_i x_j$ component
$\rho_b$	rhob	Density of blood
$C_b$	Cb	Specific heat of blood
$\omega_b$	wb	Blood perfusion rate
$T_b$	Tb	Arterial blood temperature
$Q_{met}$	Qmet	Metabolic heat source
$Q_{ext}$	Qext	Spatial heat source

**Time-scaling coefficient**  $\delta_{ts}$  is normally 1, but you can change the time scale, for example, from seconds to minutes by setting it to 1/60. See also the note on page 158.

**Density of tissue** The effective density of the tissue is for all practical purposes equal to the density of the tissue-blood continuum, because the vascular volume is much smaller than that of the solid tissue.

**Specific heat of tissue**  $C$  describes the amount of heat energy required to produce a unit temperature change in a unit mass of tissue.

**Thermal conductivity of tissue**  $k$  describes the relationship between the heat flux vector  $\mathbf{q}$  and the temperature gradient  $\nabla T$  as in

$$\mathbf{q} = -k\nabla T,$$

which is *Fourier's law of heat conduction*. Enter this quantity as the unit power per unit length and unit temperature.

**Density of blood** Enter  $\rho_b$  as the unit mass per unit volume.

**Specific heat of blood**  $C_b$  describes the amount of heat energy required to produce a unit temperature change in a unit mass of blood.

**Blood perfusion rate**  $\omega_b$  describes the volume of blood per second that flows through a unit volume of tissue.

**Arterial blood temperature**  $T_b$  is the temperature at which blood leaves the arterial blood veins and enters the capillaries.

**Metabolic heat source**  $Q_{\text{met}}$  describes heat generation from metabolism. Enter this quantity as the unit power per unit volume.

**Spatial heat source**  $Q_{\text{ext}}$  describes heat generation from an external source. Enter this quantity as the unit power per unit volume.

## Boundary Condition Types

The available boundary conditions for the Bioheat Equation application mode are:

BOUNDARY CONDITION	DESCRIPTION
$-\mathbf{n} \cdot (-k\nabla T) = q_0 + h(T_{\text{inf}} - T)$	Heat flux
$-\mathbf{n} \cdot (-k\nabla T) = 0$	Insulation or symmetry
$T = T_0$	Prescribed temperature
$r = 0$	Axial symmetry

The boundary conditions available on interior boundaries are:

INTERIOR BOUNDARY CONDITION	DESCRIPTION
$\mathbf{n} \cdot (k_1\nabla T_1 - k_2\nabla T_2) = q_0 + h(T_{\text{inf}} - T)$	Heat flux discontinuity
$\mathbf{n} \cdot (k_1\nabla T_1 - k_2\nabla T_2) = 0$	Continuity
$T = T_0$	Prescribed temperature

### HEAT FLUX

$$-\mathbf{n} \cdot (-k\nabla T) = q_0 + h(T_{\text{inf}} - T)$$

The heat flux condition accounts for a general heat flux, as well as that from convection as defined by a convective heat transfer coefficient. The application mode interprets the heat flux in the direction of the inward normal. Therefore a positive value of  $q_0$  corresponds to a heat source.

- A positive value of  $q_0$  represents a heat source on the boundary, for example an electrical heater or a known inflow of energy, such as radiation with specified intensity. Conversely, a negative value will represent a heat sink.
- The term  $h(T_{\text{inf}} - T)$  models convective heat transfer with the surrounding environment, where  $h$  is the *heat transfer coefficient* and  $T_{\text{inf}}$  is the *external bulk temperature*. The value of  $h$  depends on the geometry and the ambient flow conditions. For an introduction on how to calculate and use heat transfer coefficients, see “Convective Cooling/Heating and Heat Transfer Coefficients” on page 15.

### INSULATION OR SYMMETRY

$$-\mathbf{n} \cdot (-k\nabla T) = 0$$



This condition applies to boundaries where the domain is well insulated. It can also be used to reduce model size by taking advantage of symmetry.

### **AXIAL SYMMETRY**

This boundary condition is available only for axisymmetric versions of heat transfer application models. Use it on the symmetry axis  $r = 0$ . Note that from the mathematical point of view, this condition is identical to thermal insulation.

### **PRESCRIBED TEMPERATURE**

$$T = T_0$$

This boundary condition prescribes the temperature  $T_0$ .

### **HEAT FLUX DISCONTINUITY**

$$\mathbf{n} \cdot (k_u \nabla T_u - k_d \nabla T_d) = q_0 + h(T_{\text{inf}} - T)$$

On interior boundaries an assembly pairs you can specify a heat flux discontinuity, which is the same thing as a heat source or heat sink. In the equation for this boundary condition, the terms on the left are subindexed with “u” for upside and “d” for downside. The equation states that the difference between the heat flux on the left and the right is equal to the heat source on the boundary.

- Specify  $q_0$  to represent a heat source or sink on the interior boundary. Enter a positive quantity for a source or a negative quantity for a sink.
- The second term,  $h(T_{\text{inf}} - T)$ , models heat exchanged with a fluid flowing in a network of thin channels embedded in the boundary. An appropriate value for  $h$  can for example be found from a local analysis of a unit cell.

This boundary condition is only available at interior boundaries and for assembly pairs at boundaries between parts of an assembly.

### **CONTINUITY**

$$\mathbf{n} \cdot (k_u \nabla T_u - k_d \nabla T_d) = 0$$

The default setting for interior boundaries and assembly pairs is Continuity, which is a special case of the above Heat flux discontinuity condition. In the absence of sources or sinks, that condition specifies that the heat flux in the normal direction is continuous across the boundary. This boundary condition is only available at interior boundaries and for assembly pairs at boundaries between parts of an assembly.

## Boundary Settings

You specify the boundary conditions in the **Boundary Settings** dialog box (Figure 6-2).

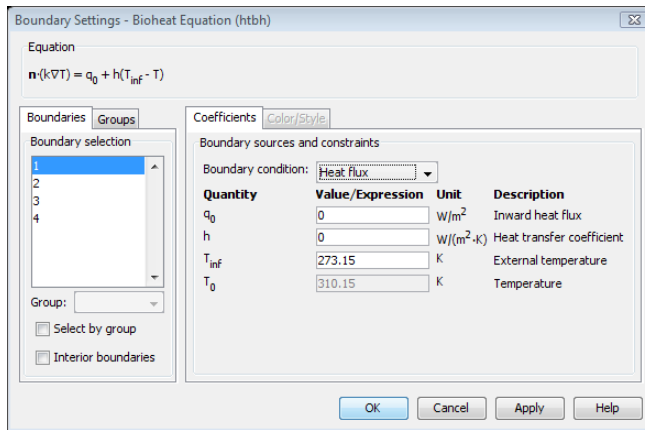


Figure 6-2: The Boundary Settings dialog box.

The quantities and variables that specify a boundary condition are:

QUANTITY	VARIABLE	DESCRIPTION
$q_0$	q	Inward heat flux
$h$	h	Convective heat transfer coefficient
$T_{\text{inf}}$	Tinf	External (bulk) temperature
$T_0$	T0	Prescribed temperature

## Example Models

For examples of models using the Bioheat Equation application mode, see Microwave Cancer Therapy on page 278 and Tumor Removal on page 291 of the *Heat Transfer Module Model Library*.

## Reference

1. A. Bejan, *Heat Transfer*, Wiley, 1993.

# Momentum Transport

The Heat Transfer Module extends the CFD capability of COMSOL Multiphysics by adding turbulence modeling and support for weakly compressible flows. This enables modeling of forced or temperature gradient-driven flows in both laminar and turbulent regimes.

# Introduction

All application modes described in this chapter are based on the Navier-Stokes equations, which in their most general form read

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (7-1)$$

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho (\mathbf{u} \cdot \nabla) \mathbf{u} = \nabla \cdot [-p \mathbf{I} + \boldsymbol{\tau}] + \mathbf{F} \quad (7-2)$$

$$\rho C_p \left( \frac{\partial T}{\partial t} + (\mathbf{u} \cdot \nabla) T \right) = -(\nabla \cdot \mathbf{q}) + \boldsymbol{\tau} : \mathbf{S} - \frac{T}{\rho} \frac{\partial \rho}{\partial T} \bigg|_p \left( \frac{\partial p}{\partial t} + (\mathbf{u} \cdot \nabla) p \right) + Q \quad (7-3)$$

where

- $\rho$  is the density (kg/m<sup>3</sup>)
- $\mathbf{u}$  is the velocity vector (m/s)
- $p$  is pressure (Pa)
- $\boldsymbol{\tau}$  is the viscous stress tensor (Pa)
- $\mathbf{F}$  is the body force vector (N/m<sup>3</sup>)
- $C_p$  is the specific heat capacity at constant pressure (J/(kg·K))
- $T$  is absolute temperature (K)
- $\mathbf{q}$  is the heat flux vector (W/m<sup>2</sup>)
- $Q$  contains the heat sources (W/m<sup>3</sup>)

$\mathbf{S}$  is the strain rate tensor:

$$\mathbf{S} = \frac{1}{2} (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)$$

The operation “:” denotes a contraction between tensors defined by

$$\mathbf{a} : \mathbf{b} = \sum_n \sum_m a_{nm} b_{nm} \quad (7-4)$$

Equation 7-1 is the continuity equation and represents the conservation of mass.

Equation 7-2 is a vector equation and represent the conservation of momentum.

Equation 7-3 describes the conservation of energy, formulated in terms of

temperature. This is an intuitive formulation that facilitates specification of boundary conditions.

To close the equation system 7-1 through 7-3, some constitutive relations are needed. A common relation is derived by assuming that the fluid is Newtonian. Together with Stokes' assumption, the viscous stress tensor becomes:

$$\boldsymbol{\tau} = 2\eta\mathbf{S} - \frac{2}{3}\eta(\nabla \cdot \mathbf{u})\mathbf{I} \quad (7-5)$$

The dynamic viscosity  $\eta$  (Pa-s) is allowed to depend on the thermodynamic state, but not on the velocity field. All gases and many liquids can be considered Newtonian. Examples of non-Newtonian fluids are honey, mud, blood, liquid metals, and most polymer solutions. Support for modeling of non-Newtonian fluids can be found in the COMSOL Chemical Engineering Module, while the Heat Transfer Module treats all fluids as Newtonian according to Equation 7-5. Other commonly used constitutive relations are Fourier's law of conduction and the ideal gas law.

There are several books where the Navier-Stokes equations are derived and concepts such as Newtonian fluids and the Stokes assumption are explained in detail. See, for example, the classical text by Batchelor (Ref. 1) and the more recent work by Panton (Ref. 2).

Many applications describe isothermal flows where Equation 7-3 is decoupled from Equation 7-1 and Equation 7-2. The temperature equation is described in the section "The Heat Equation" on page 6, and the fully-coupled system is treated in the section "Non-Isothermal Flow" on page 228.

Each section discusses a specific form of the continuity and momentum equations. The properties of the equations are then used as a foundation to clarify their range of applicability.

### *References*

---

1. G.K. Batchelor, "*An Introduction To Fluid Dynamics*," Cambridge University Press, 1967.
2. R.L. Panton, "*Incompressible Flow*," second edition, John Wiley & sons, inc., 1996.

# The Incompressible Navier-Stokes Equations

The incompressible Navier Stokes equations in their basic form describe flow of a Newtonian fluid whose density can be assumed to be constant or nearly constant. In the Heat Transfer Module, you cannot directly access any dedicated application mode for incompressible laminar flow. The Incompressible Navier-Stokes application mode is, however, the common platform on which all flow application modes are based and to which they return if you choose to turn off all special features. This section therefore describes the basic functionality which is shared between the flow application modes in the context of the Incompressible Navier Stokes application mode.

In theory, the same equations describe laminar as well as turbulent flows. In practice however, the mesh resolution required to simulate turbulence with this application mode makes such an approach impractical. This is discussed further in the section “Numerical Stability—Stabilization Techniques” on page 193.

Buoyant flows in general should be treated by the Non-Isothermal Flow predefined multiphysics coupling, see “Non-Isothermal Flow” on page 228. It is possible, however, to treat some simple cases of buoyant flows with the Incompressible Navier-Stokes application mode by invoking the Boussinesq approximation. “The Boussinesq Approximation” on page 198 describes this in more detail.

## *Subdomain Equations*

---

The Incompressible Navier-Stokes application mode assumes that the fluid is incompressible; that is, that  $\rho$  is constant or nearly constant. This is the case for all fluids under normal conditions and also for gases at low velocities. For constant  $\rho$ , Equation 7-1 reduces to

$$\nabla \cdot \mathbf{u} = 0 \quad (7-6)$$

and the stress tensor  $\boldsymbol{\tau}$  in Equation 7-5 becomes

$$\boldsymbol{\tau} = \eta(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)$$

These assumptions put together give the following momentum equations:

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla) \mathbf{u} = \nabla \cdot [-p \mathbf{I} + \eta(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)] + \mathbf{F} \quad (7-7)$$

The equations used by the application mode are displayed in the **Subdomain Settings** dialog box. Here, you also specify the physical properties of the fluid (see Figure 7-1).

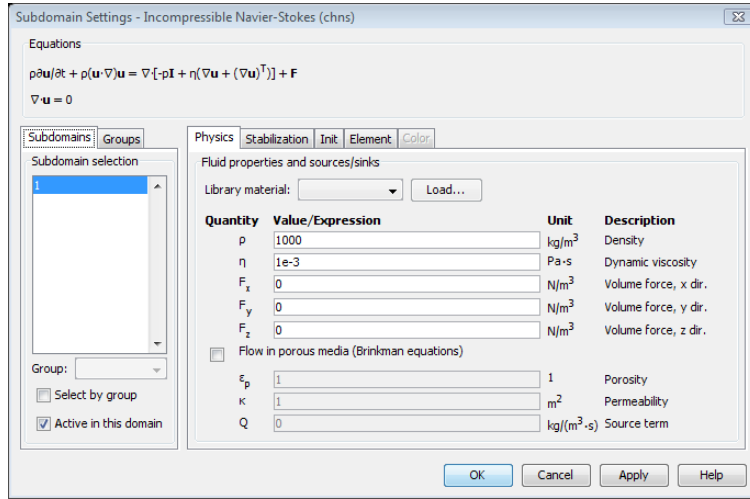


Figure 7-1: The Subdomain Settings dialog box for 3D incompressible Navier-Stokes equations.

## Boundary Conditions

The boundary conditions for the Incompressible Navier-Stokes application mode are grouped into the following types:

- Wall
  - No slip (Default)
  - Slip
  - Sliding wall
  - Moving/leaking wall
- Inlet
  - Velocity (Default)
  - Pressure, no viscous stress
  - Laminar inflow

- Outlet
  - Velocity
  - Pressure
  - Pressure, no viscous stress (Default)
  - No viscous stress
  - Normal stress
  - Laminar outflow
- Symmetry boundary
  - Symmetry (Default)
  - Axial symmetry
- Open boundary
  - Normal stress (Default)
  - No viscous stress
- Stress
  - General stress (Default)
  - Normal stress
  - Normal stress, normal flow



You specify a boundary condition in the **Boundary Settings** dialog where you first select the appropriate **Boundary type** and then a **Boundary Condition**.

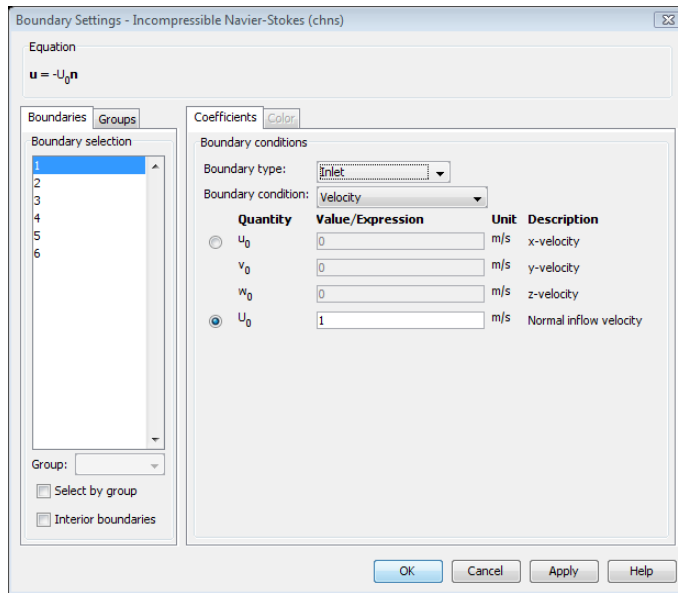


Figure 7-2: Boundary Settings dialog box for the Incompressible Navier-Stokes application mode.

If a mathematical formulation describes more than one type of physical boundary condition, it can appear in more than one boundary type. However, every possible use of a single mathematical formulation cannot be covered. Hence, the boundary types should be regarded as guidelines, not as restrictions on the applicability of the formulations.

The theory of most boundary conditions can be found in Ref. 1.

## WALL

These boundary conditions describe the existence of a solid wall.

### *No Slip*

This is the standard and default boundary condition for a stationary solid wall. The condition prescribes

$$\mathbf{u} = \mathbf{0}$$

that is, that the fluid at the wall is not moving.

### *Moving/Perforated Wall*

If the wall moves, so must the fluid. Hence, this boundary condition prescribes

$$\mathbf{u} = \mathbf{u}_w$$

Note that setting this boundary condition does not automatically cause the associated wall to move. The section “The Moving Mesh Application Mode” on page 463 of the *COMSOL Multiphysics Modeling Guide* describes how to set up a model with moving boundaries.

You can also use the Moving/perforated wall boundary condition to simulate a wall where fluid is leaking into or leaving through a perforated wall.

### *Sliding Wall*

If you use this boundary condition, the wall is assumed to behave like a conveyor belt, that is, that the surface is sliding in its tangential direction. The wall does not have to actually move in the coordinate system.

In two space dimensions (2D), the tangential direction is unambiguously defined by the direction of the boundary. However, the situation becomes more complicated in 3D. For this reason, this boundary condition has slightly different definitions in the different space dimensions.

**2D and Axial Symmetry** The velocity is given as a scalar  $U_w$  and the condition prescribes

$$\mathbf{u} \cdot \mathbf{n} = 0, \quad \mathbf{u} \cdot \mathbf{t} = U_w$$

where  $\mathbf{t} = (-n_y, n_x)$  for 2D and  $\mathbf{t} = (-n_z, n_r)$  for axial symmetry. When the Swirl flow property is activated (see section “Swirl Flow” on page 203), the swirl wall component is specified separately.

**3D** The velocity is set equal to a given vector  $\mathbf{u}_w$  projected onto the boundary plane:

$$\mathbf{u} = \mathbf{u}_w - (\mathbf{n} \cdot \mathbf{u}_w)\mathbf{n}$$

### *Slip*

The slip condition assumes that there are no viscous effects at the slip wall and hence, no boundary layer develops. From a modeling point of view, this may be a reasonable approximation if the important effect of the wall is to prevent fluid from leaving the domain. Mathematically, the constraint can be formulated as:

$$\mathbf{u} \cdot \mathbf{n} = 0, \quad \mathbf{t} \cdot (-p\mathbf{I} + \eta(\nabla\mathbf{u} + (\nabla\mathbf{u})^T))\mathbf{n} = 0$$

where  $\mathbf{t}$  is a tangential vector to the boundary.

### **INLET**

This boundary type contains different ways to specify conditions on a boundary where the fluid is supposed to enter the domain. Notice that the formulations contained in this boundary type all appear, some of them slightly modified, in the Outflow boundary type as well. Hence, there is nothing in the mathematical formulations that prevents a fluid from leaving the domain through boundaries where you have specified the Inlet boundary type.

#### *Velocity*

This boundary condition offers two ways to specify an inlet velocity. The first is to set the velocity equal to a given vector  $\mathbf{u}_0$ :

$$\mathbf{u} = \mathbf{u}_0$$

The other is to specify a normal inflow velocity:

$$\mathbf{u} = -\mathbf{n}U_0$$

Note that the boundary normal,  $\mathbf{n}$ , points out of the domain.

#### *Pressure, No Viscous Stress*

This boundary condition specifies vanishing viscous stress along with a Dirichlet condition on the pressure:

$$\eta(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)\mathbf{n} = \mathbf{0}, \quad p = p_0$$

It is a numerically stable boundary condition that admits total control of the pressure level along the entire boundary. However, if the inflow is not normal to the boundary, this condition yields an overspecification of the problem. In the case that your solution turns out to have a non-normal inflow velocity, there are two choices. Either, move the boundary farther away to a location where the inflow is normal to the boundary or, use a stress type boundary condition described on page 192.

Note that this condition is identical to the Pressure, no viscous stress condition for Outflow boundaries. Hence, depending on the pressure field in the rest of the subdomain, a boundary with this condition can very well become an outflow boundary.

### Laminar Inflow

This boundary conditions adds a weak contribution of the form

$$-L_{\text{entr}}((-p\mathbf{I} + \eta(\nabla_{\mathbf{t}}\mathbf{u} + (\nabla_{\mathbf{t}}\mathbf{u})^T)), \nabla_{\mathbf{t}}\hat{\mathbf{u}})_{\Gamma} - p_{\text{entr}}(\mathbf{n}, \hat{\mathbf{u}})_{\Gamma} - L_{\text{entr}}((\nabla_{\mathbf{t}} \cdot \mathbf{u}), \hat{p})_{\Gamma} \quad (7-8)$$

Where  $(\cdot, \cdot)_{\Gamma}$  is the  $L_2$  inner product on the boundary,  $\nabla_{\mathbf{t}}$  is the gradient in the tangential direction of the boundary, and  $\hat{\mathbf{u}}$  and  $\hat{p}$  are the test functions for the velocities and pressure, respectively. The addition of Equation 7-8 corresponds to the situation shown in Figure 7-3: a fictitious domain of length  $L_{\text{entr}}$  is assumed to be attached to the inlet of the computational domain. The flow in this fictitious domain is assumed to be laminar plug flow. If you select the option **Constrain outer edges to zero**, the flow in the fictitious domain is instead assumed to be fully developed laminar channel flow (in 2D) or fully developed laminar internal flow (in 3D). This does not affect the boundary condition in the real domain,  $\Omega$ , where the boundary conditions are always fulfilled.

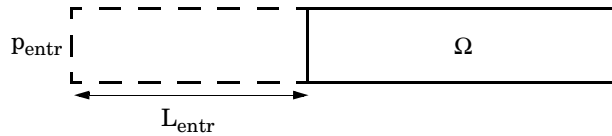


Figure 7-3: Sketch of the physical situation simulated when using the Laminar inflow boundary condition.  $\Omega$  is the actual computational domain while the dashed domain is a fictitious domain.

If you specify an average inlet velocity or inlet mass flow instead of the pressure, COMSOL Multiphysics adds an ODE that calculates a pressure,  $p_{\text{entr}}$ , such that the desired inlet velocity or mass flow is obtained.

### OUTFLOW

This boundary type contains different ways to specify conditions on a boundary where the fluid exits the domain. Note that all of the formulations in this type can be found, possibly slightly modified, in other boundary types as well. Hence, there is nothing in the mathematical formulations that prevent a fluid from entering the domain through boundaries where you have set the Outflow boundary type.

Setting outlet conditions for the Navier-Stokes equations is not a trivial task. A general rule of thumb, however, is that if there is something interesting happening at an outflow boundary, extend the computational domain to include this phenomenon.

### Velocity

This boundary condition offers two ways to specify an outlet velocity. The first is to set the velocity equal to a given vector  $\mathbf{u}_0$ :

$$\mathbf{u} = \mathbf{u}_0$$

The other is to specify a normal outlet velocity:

$$\mathbf{u} = \mathbf{n}U_0$$

Observe that the boundary normal,  $\mathbf{n}$ , is pointing out of the domain.

### Pressure, No Viscous Stress

This boundary condition specifies vanishing viscous stress along with a Dirichlet condition on the pressure:

$$\eta(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)\mathbf{n} = \mathbf{0}, \quad p = p_0$$

It is a numerically stable boundary condition that admits total control of the pressure level at the whole boundary. However, if the outflow is not normal to the boundary, this condition is an overspecification. In the case that your solution turns out to have a non-normal outflow velocity, there are two choices. Either move the boundary farther away to a location where the outflow is normal to the boundary or use a stress type boundary condition described on page 192.

Observe that this condition is identical to the Pressure, no viscous stress condition for Inflow boundaries. Hence, depending on the pressure field in the rest of the subdomain, a boundary with this condition can very well become an inflow boundary.

### Pressure

This boundary condition prescribes only a Dirichlet condition for the pressure:

$$p = p_0$$

Use this boundary condition only for high Reynolds number outflow boundaries, that is  $\text{Re}^c = \rho|\mathbf{u}|h/(2\eta) \gg 1$ . It is far less stable than the Pressure, no viscous stress boundary condition, but it is consistent with a non-normal outflow velocity.

### No Viscous Stress

Prescribes vanishing viscous stress:

$$\eta(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)\mathbf{n} = \mathbf{0}$$

This condition can be useful in some situations because it does not impose any constraint on the pressure. A typical example is a model with volume forces that give rise to pressure gradients that are hard to prescribe in advance. It should however be combined with a point constraint on the pressure to be numerically stable (see “Point Settings” on page 193).

#### Normal Stress

The total stress on the boundary is set equal to a stress vector of magnitude,  $f_0$ , oriented in the negative normal direction:

$$(-p\mathbf{I} + \eta(\nabla\mathbf{u} + (\nabla\mathbf{u})^T))\mathbf{n} = -f_0\mathbf{n}$$

This implies that the total stress in the tangential direction is zero. This boundary condition implicitly sets a constraint on the pressure that for 2D flows can be written

$$p = 2\eta\frac{\partial u_n}{\partial n} + f_0 \quad (7-9)$$

If  $\partial u_n/\partial n$  is small, Equation 7-9 can be interpreted as  $p \approx f_0$ .

#### Laminar Outflow

This boundary conditions adds a weak contribution of the form

$$-L_{\text{exit}}((-p\mathbf{I} + \eta(\nabla_t\mathbf{u} + (\nabla_t\mathbf{u})^T)), \nabla_t\hat{\mathbf{u}})_\Gamma - p_{\text{exit}}(\mathbf{n}, \hat{\mathbf{u}})_\Gamma - L_{\text{exit}}((\nabla_t \cdot \mathbf{u}), \hat{p})_\Gamma \quad (7-10)$$

Where  $(\cdot, \cdot)_\Gamma$  is the  $L_2$  inner product on the boundary,  $\nabla_t$  is the gradient in the tangential direction of the boundary, and  $\hat{\mathbf{u}}$  and  $\hat{p}$  are the test functions for the velocities and pressure, respectively. The addition of Equation 7-10 corresponds to the situation shown in Figure 7-4: a fictitious domain of length  $L_{\text{exit}}$  is assumed to be attached to the outlet of the computational domain. The flow in this fictitious domain is assumed to be laminar plug flow. If you select the **Constrain outer edges to zero** check box, the flow in the fictitious domain is instead assumed to be fully developed laminar channel flow (in 2D) or fully developed laminar internal flow (in 3D). This does not

affect the boundary condition in the real domain,  $\Omega$ , where the boundary conditions are always fulfilled.

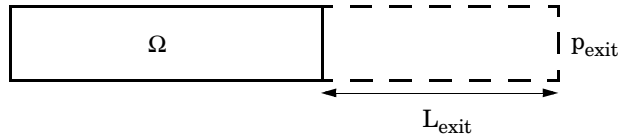


Figure 7-4: Sketch of the physical situation simulated when using Lamellar outflow boundary condition.  $\Omega$  is the actual computational domain while the dashed domain is a fictitious domain.

If you specify an average outlet velocity or inlet mass flow instead of the pressure, the software adds an ODE that calculates  $p_{\text{exit}}$  such that the desired outlet velocity or mass flow is obtained.

#### SYMMETRY BOUNDARY

Prescribes no penetration and vanishing shear stresses:

$$\mathbf{u} \cdot \mathbf{n} = 0, \quad \mathbf{t} \cdot (-p\mathbf{I} + \eta(\nabla\mathbf{u} + (\nabla\mathbf{u})^T))\mathbf{n} = 0$$

In 2D Axial Symmetry, the above formulation is called **Symmetry**.

#### Axial Symmetry

This boundary condition is only available in 2D Axial Symmetry. Use it on all boundaries with coordinate  $r = 0$ . It prescribes  $u_r = 0$  and vanishing stresses in the  $z$  direction.

#### OPEN BOUNDARY

You can use this boundary type on boundaries that are open to large volumes of fluid. Fluid can both enter and leave the domain on boundaries with this type of condition.

#### No Viscous Stress

Prescribes vanishing viscous stress:

$$\eta(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)\mathbf{n} = \mathbf{0}$$

This condition can be useful in some situations because it does not impose any constraint on the pressure. A typical example is a model with volume forces that give rise to pressure gradients that are hard to prescribe in advance. It should however be combined with a point constraint on the pressure to be numerically stable (see “Point Settings” on page 193).

### Normal Stress

The total stress on the boundary is set equal to a stress vector of magnitude,  $f_0$ , oriented in the negative normal direction:

$$(-p\mathbf{I} + \eta(\nabla\mathbf{u} + (\nabla\mathbf{u})^T))\mathbf{n} = -f_0\mathbf{n}$$

This implies that the total stress in the tangential direction is zero. This boundary condition implicitly sets a constraint on the pressure that for 2D flows can be written

$$p = 2\eta \frac{\partial u_n}{\partial n} + f_0 \quad (7-11)$$

If  $\partial u_n / \partial n$  is small, Equation 7-11 can be interpreted as  $p \approx f_0$ .

### STRESS

This type of boundary condition represents a very general class of conditions also known as traction boundary conditions.

### General Stress

The total stress on the boundary is set equal to a given stress  $\mathbf{F}$ :

$$(-p\mathbf{I} + \eta(\nabla\mathbf{u} + (\nabla\mathbf{u})^T))\mathbf{n} = \mathbf{F}$$

This boundary condition implicitly sets a constraint on the pressure that for 2D flows can be written

$$p = 2\eta \frac{\partial u_n}{\partial n} - \mathbf{n} \cdot \mathbf{F} \quad (7-12)$$

If  $\partial u_n / \partial n$  is small, Equation 7-12 can be interpreted as  $p \approx -\mathbf{n} \cdot \mathbf{F}$ .

### Normal Stress

The total stress on the boundary is set equal to a stress vector of magnitude,  $f$ , oriented in the normal direction:

$$(-p\mathbf{I} + \eta(\nabla\mathbf{u} + (\nabla\mathbf{u})^T))\mathbf{n} = -f_0\mathbf{n}$$

This implies that the total stress in the tangential direction is zero. This boundary condition implicitly sets a constraint on the pressure that for 2D flows can be written

$$p = 2\eta \frac{\partial u_n}{\partial n} + f_0 \quad (7-13)$$

If  $\partial u_n / \partial n$  is small, Equation 7-13 can be interpreted as  $p \approx f_0$ .



### *Normal Stress, Normal Flow*

In addition to the stress condition set in the Normal stress condition, this condition also prescribes that there must be no tangential velocities on the boundary:

$$(-p\mathbf{I} + \eta(\nabla\mathbf{u} + (\nabla\mathbf{u})^T))\mathbf{n} = -f_0\mathbf{n}, \quad \mathbf{t} \cdot \mathbf{u} = 0$$

Also this boundary condition implicitly sets a constraint on the pressure that for 2D flows can be written

$$p = 2\eta \frac{\partial u_n}{\partial n} + f_0 \quad (7-14)$$

If  $\partial u_n / \partial n$  is small, Equation 7-14 can be interpreted as  $p \approx f_0$ .

### *Point Settings*

---

If it is not possible to specify the pressure level using a boundary condition, the pressure must be set in some other way, for example, by specifying a fixed pressure at a point. You find a dialog box for **Point Settings** on the **Physics** menu.

### *Numerical Stability—Stabilization Techniques*

---

The momentum equations (Equation 7-7) are (nonlinear) convection-diffusion equations. As described in Chapter 17, “Stabilization Techniques,” in the *COMSOL Multiphysics Modeling Guide* such equations are unstable if discretized using the Galerkin finite element method. Stabilized finite element methods are therefore necessary in order to obtain physical solutions. You control the stabilization settings from the **Stabilization** page in the **Subdomain Settings** (see Figure 7-5).

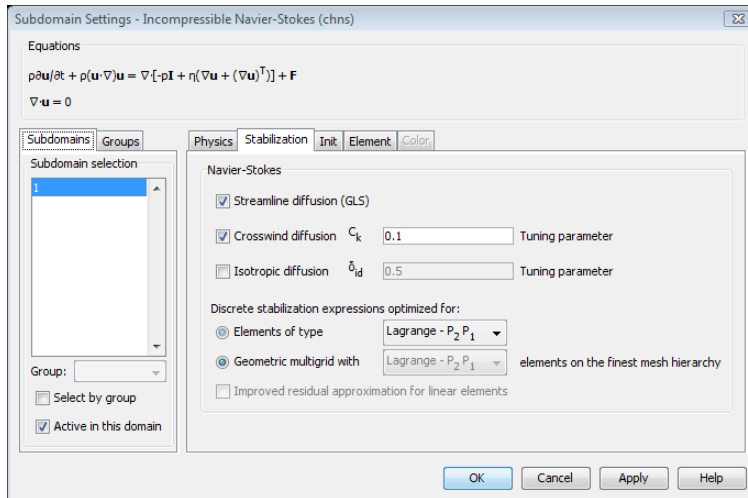


Figure 7-5: The Stabilization page of the Subdomain Settings dialog box.

There are three types of stabilization available for Navier-Stokes:

- Streamline diffusion (GLS)
- Crosswind diffusion
- Isotropic diffusion

For optimal functionality, the exact weak formulations and constants of GLS and crosswind diffusion should depend on the order of the basis functions (elements). You can obtain this using the two option buttons, each with its corresponding list, at the bottom of the **Stabilization** page (see Figure 7-5). In most cases, the model uses only one element type. If this is the case, select this type from the **Elements of type** list. This is the default setting for all space dimensions except 3D.

However, when using the geometric multigrid solver/preconditioner (either as a solver or as a preconditioner), it is possible to discretize the equations with different elements on the different multigrid hierarchies. Then select the element type that is used on the finest mesh level from the **Geometric multigrid with** list. This is the default setting for 3D because the default solver for in 3D is BiCGStab with geometric multigrid as preconditioner. The resulting weak expressions then take forms adapted to multigrid hierarchies created in a way equivalent to the hierarchy generation method that you get by selecting **Lower element order first (any)** (see “Constructing a Multigrid Hierarchy” on page 558 in the *COMSOL Multiphysics Reference Guide*).

The values of constants of GLS and crosswind diffusion follow Ref. 2 and Ref. 3 and cannot be changed without performance losses.

Streamline diffusion and crosswind diffusion are functional only for the element types available in the lists on the **Stabilization** page.

### **STREAMLINE DIFFUSION**

For strongly coupled systems of equations, the streamline diffusion must be applied to the whole system of equations, not only to each equation separately. These ideas were first explored by Hughes and Mallet (Ref. 4) and were later extended to Galerkin least-squares (GLS) applied to the Navier-Stokes equations (Ref. 5), which is the form that COMSOL Multiphysics supports. The time-scale tensor is the diagonal tensor presented in Ref. 6.

Streamline diffusion is active per default because it is necessary when convection is the dominating part of the flow.

The unstabilized incompressible Navier-Stokes equations are subject the Babuska-Brezzi condition, which states that the basis functions for the pressure must be of lower order than the basis functions for the velocity. If the incompressible Navier-Stokes equations are stabilized by GLS, it is possible to use equal-order interpolation. Hence, streamline diffusion is necessary when using linear Lagrange elements. This applies also if the model is solved using geometric multigrid (either as a solver or as a preconditioner) and at least one multigrid hierarchy level uses linear Lagrange elements.

### **CROSSWIND DIFFUSION**

Crosswind diffusion can also be formulated for systems of equations, and when applied to the Navier-Stokes equations it becomes a shock-capturing operator. COMSOL supports the formulation in Ref. 5 with shock capturing viscosity taken from Ref. 7.

Incompressible flows do not contain shock waves, but crosswind diffusion is still useful to introduce extra diffusion in sharp boundary layers and shear layers that otherwise would require a very dense mesh to resolve.

The tuning parameter,  $C_k$ , controls the amount of crosswind diffusion introduced in the model. The recommended range for low Mach number flows and incompressible flows is  $0 < C_k < 0.5$  ( $C_k=0$  means no diffusion at all). The value must be neither space dependent nor time dependent.

Crosswind diffusion is active per default because it makes it easier to obtain a solution even if the mesh is not perfect. It is advisable, however, to investigate the influence  $C_k$ .

Crosswind diffusion also has the effect that iterative solvers can use inexpensive presmoothers such as SSOR (see “Solver Settings” on page 199). If you deactivate crosswind diffusion, you must also change all applications of SSOR to Vanka (see “Preconditioners for the Iterative Solvers” on page 553 in the *COMSOL Multiphysics Reference Guide*).

### ISOTROPIC DIFFUSION

Isotropic diffusion adds diffusion to the momentum equations of the Navier-Stokes equations in the same way as described in “Isotropic Diffusion” on page 500 in the *COMSOL Multiphysics Modeling Guide*. The continuity equation gains no extra stability.

### IMPROVED RESIDUAL APPROXIMATION FOR LINEAR ELEMENTS

Both GLS and crosswind diffusion evaluate the residuals of the Navier-Stokes equations to determine the amount of diffusion. The residuals of the momentum equations contain second-order derivatives of the velocity, which, if linear elements are used, are evaluated to zero. The impact of this is small if the flow is dominated by convection where

$$\partial \mathbf{u} / \partial t + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p \quad (7-15)$$

is a good approximation of the momentum equations. Equation 7-15 is not a good approximation for flows dominated by viscous effects, and if the mesh is coarse, the result can be that the pressure field contains small oscillations.

You can remove the oscillations by refining the mesh. An alternative is to select the **Improved residual approximation for linear elements** check box (see Figure 7-5). The program then reconstructs the viscous terms by polynomial-preserving recovery (see “Using Special Operators” on page 163 in the *COMSOL Multiphysics User’s Guide*). Polynomial-preserving recovery makes the equation system more complicated and thereby more expensive to solve.

The improved residuals can be useful if the Navier-Stokes equations are part of a multiphysics problem and you cannot afford to refine the mesh.

## Corner Smoothing

You find the **Corner smoothing** property in the **Application Mode Properties** dialog box. It can be a useful property when the model contains walls with slip conditions, as described below.

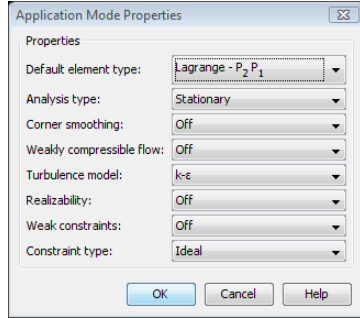


Figure 7-6: The *Application Mode Properties* dialog box where *Corner smoothing* can be turned on and off.

Consider the situation sketched in Figure 7-7. At the point where the boundaries  $\Gamma_1$  and  $\Gamma_2$  intersect, there will be two boundary normals, one for  $\Gamma_1$  and one for  $\Gamma_2$ . These two normals are denoted  $\mathbf{n}_{\Gamma_1}$  and  $\mathbf{n}_{\Gamma_2}$  in Figure 7-7. If the boundaries now both have no-penetration condition, there will be two Dirichlet conditions at the point of intersection, namely

$$\mathbf{n}_{\Gamma_1} \cdot \mathbf{u} = 0 \quad (7-16)$$

and

$$\mathbf{n}_{\Gamma_2} \cdot \mathbf{u} = 0 \quad (7-17)$$

The only way that both Equation 7-16 and Equation 7-17 can be fulfilled is if  $\mathbf{u} = \mathbf{0}$  at the point of intersection. This is not always the expected solution, however.

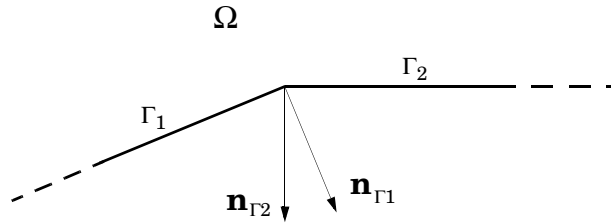


Figure 7-7: Intersection between the boundaries  $\Gamma_1$  and  $\Gamma_2$ .  $\mathbf{n}_{\Gamma_1}$  and  $\mathbf{n}_{\Gamma_2}$  are the boundary normals prescribed by  $\Gamma_1$  and  $\Gamma_2$  respectively.  $\Omega$  is the computational domain.

When corner smoothing is activated, any Dirichlet condition  $d(\mathbf{n}) = 0$  is replaced by  $d(\mathbf{n}_w) = 0$ , where  $\mathbf{n}_w$  is a vector of dependent boundary variables whose solution in each point is the average of all normals in that point. In the current example, equations 7-16 and 7-17 are replaced with

$$\mathbf{n}_w \cdot \mathbf{u} = 0 \quad (7-18)$$

and  $\mathbf{n}_w$  has the solution  $\mathbf{n}_w = (1/2)(\mathbf{n}_{\Gamma_1} + \mathbf{n}_{\Gamma_2})$ . Equation 7-18 can then be satisfied for  $\mathbf{u} \neq \mathbf{0}$ .

### The Boussinesq Approximation

---

The Boussinesq approximation is a way to treat some simple cases of buoyant flows without having to use a compressible formulation of the Navier-Stokes equations.

The Boussinesq approximation assumes that variations in density have no effect on the flow field except that they give rise to buoyant forces. The density is taken to be a reference value,  $\rho_0$ , except in the body force term, which is set to

$$\mathbf{F} = (\rho_0 + \Delta\rho)\mathbf{g} \quad (7-19)$$

where  $\mathbf{g}$  is the gravity vector. You can enter an expression for Equation 7-19 in the **Volume force** edit fields in the **Subdomain Settings** dialog box (see Figure 7-1 on page 183). However, further simplifications are often possible. Because  $\mathbf{g}$  can be written in terms of a potential,  $\Phi$ , it is possible to write Equation 7-19 as:

$$\mathbf{F} = -\nabla(\rho_0\Phi) + \Delta\rho\mathbf{g}$$

The first part can be canceled out by splitting the true pressure,  $p$ , as a sum of a hydrodynamic component,  $P$ , and a hydrostatic component,  $-\rho_0\Phi$ . Then you can write equations 7-6 and 7-7 in terms of the hydrodynamic pressure  $P = p + \rho_0\Phi$ :

$$\nabla \cdot \mathbf{u} = 0 \quad (7-20)$$

$$\rho_0 \frac{\partial \mathbf{u}}{\partial t} + (\rho_0 \mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla P + \nabla \cdot (\eta(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)) + \mathbf{g} \Delta p \quad (7-21)$$

Hence, to obtain the Boussinesq approximation on this form, you only have to enter the expression for  $\mathbf{g} \Delta p$  in the **Volume Force** edit fields in the **Subdomain Settings** dialog box.

In practice, the shift from  $p$  to  $P$  can be ignored except where the pressure appears in boundary conditions. The pressure that you specify at boundaries is the hydrodynamic pressure in this case. For example, on a vertical outflow or inflow boundary, the hydrodynamic pressure is typically a constant, while the true pressure is a function of the vertical coordinate.

The system that Equation 7-20 and Equation 7-21 form has its limitations. The main assumption is that the density fluctuations must be small, that is,  $\Delta\rho/\rho_0 \ll 1$ . There are also some more subtle constraints that, for example, makes the Boussinesq approximation unsuitable for systems of very large dimensions. An excellent discussion of the Boussinesq approximation and its limitations appears in Chapter 14 of Ref. 8.

### *Solver Settings*

---

#### **LINEAR SYSTEM SOLVER**

The recommended solver type for small and medium-sized fluid-flow problems is a direct solver, and the PARDISO direct solver is the default solver for 1D and 2D models. It is not as robust as UMFPACK or SPOOLES but is more memory efficient.

3D problems are often too large to solve using a direct solver. The default solver is BiCGStab with the geometric multigrid (GMG) preconditioner. The GMG preconditioner uses the SSOR preconditioner/smoothers as the presmoothers and postsmoothers. It is possible to use SSOR thanks to the GLS streamline diffusion and to the crosswind diffusion (“Numerical Stability—Stabilization Techniques” on page 193). If you deactivate any of these stabilization techniques, you must replace the SSOR smoothers by the Vanka preconditioner/smoothers (see “The Vanka Algorithm” on page 568 in the *COMSOL Multiphysics Reference Guide*).

For more information about the solvers, see Chapter 6, “Solving the Model,” on page 377 in the *COMSOL Multiphysics User’s Guide*.

### TRANSIENT SOLVER SETTINGS

The transient Incompressible Navier-Stokes application mode uses per default the Generalized  $\alpha$  method (see “The Time-Dependent Solver” on page 391 in the *COMSOL Multiphysics User’s Guide*) with manual time step control.

The time step for the transient Incompressible Navier-Stokes application mode is calculated from the CFL number:

$$\text{CFL} = \frac{|\mathbf{u}|\Delta t}{h}$$

The Navier-Stokes application mode contains a variable `dt_cfl_gn_app` that corresponds to a CFL number equal to one.  $n$  is the index of the geometry on which the application mode is active and `app` is the application mode name. As an example, assume that the application mode name is `chns` and that the application mode is active on geometry 2. The time step variable is then `dt_cfl_g2_chns`.

`dt_cfl_gn_app` is defined by

$$\text{dt\_cfl\_gn\_app} = \min_{\Omega_e} \left( \frac{h}{\max(|\mathbf{u}|, 10\mathbf{u}_{\text{diff}})} \right)$$

where  $\Omega_e$  is the set of all mesh cells in geometry  $n$  on subdomains where the Navier-Stokes is active.  $\mathbf{u}_{\text{diff}}$  is a diffusive velocity scale equal to  $\eta/(\rho h)$ . The argument  $10\mathbf{u}_{\text{diff}}$  is necessary to prevent the time step from becoming too large when  $|\mathbf{u}|$  is small.

A necessary condition for explicit time stepping techniques to be stable is that the CFL number is smaller than one. COMSOL uses implicit schemes which are stable for any  $\Delta t$ . Large  $\Delta t$  has the advantage that the computation can be completed in few time steps. There are however two drawbacks:

- The solver will introduce numerical diffusion that increases with  $\Delta t$ . In other words, using large  $\Delta t$  means that the result will be less accurate compared to using small  $\Delta t$ .
- The equation system becomes harder to solve for large  $\Delta t$ . That is, more computational work is required to complete each time step.

A compromise between efficiency and accuracy is to use a  $\Delta t$  such that  $2 \leq \text{CFL} \leq 10$ . The default is  $\Delta t = 5 \cdot \text{dt\_cfl\_gn\_app}$ , and the setting can be found on the **Time Stepping** page in the **Solver Parameters** dialog box.



The time step can be increased if you expect the model to feature only very slow transients. If the solver on the other hand repeatedly fails to solve the nonlinear system (see “The Time-Dependent Solver Log” on page 452 in the *COMSOL Multiphysics User’s Guide*), then the time step should be decreased.

#### ADVANCED SOLVER SETTINGS

For a flow that is parallel to a coordinate axis, the automatic scaling feature in COMSOL Multiphysics does not work if you use the nonlinear stationary solver. For such cases, turn off the scaling feature or use manual scaling; see “Scaling of Variables and Equations” on page 529 in the *COMSOL Multiphysics Reference Guide*. The problem occurs when one solution component is on average zero.

#### References

---

1. P.M. Gresho and R.L. Sani, *Incompressible Flow and the Finite Element Method, Volume 2: Isothermal Laminar Flow*, John Wiley and Sons, LTD, 2000.
2. I. Harari and T.J.R. Hughes, “What are  $C$  and  $h$ ? Inequalities for the analysis and design of finite element methods,” *Comp. Meth. Appl. Mech. Engrg*, vol. 97, pp. 157–192, 1992.
3. Y. Bazilevs, V.M. Calo, T.E. Tezduyar, and T.J.R. Hughes, “ $\text{YZ}\beta$  discontinuity capturing for advection-dominated processes with application to arterial drug delivery,” *Int.J.Num. Meth. Fluids*, vol. 54, pp. 593–608, 2007.
4. T.J.R. Hughes and M. Mallet, “A new finite element formulation for computational fluid dynamics: III. The Generalized Streamline Operator for Multidimensional Advective-Diffusive System,” *Comp. Meth. Appl. Mech. Engrg*, vol. 58, pp. 305–328, 1986.
5. G. Hauke and T.J.R. Hughes, “A unified approach to compressible and incompressible flows” *Comp. Meth. Appl. Mech. Engrg*, vol. 113, pp. 389–395, 1994.
6. G. Hauke, “Simple stabilizing matrices for the computation of compressible flows in primitive variables,” *Comp. Meth. Appl. Mech. Engrg*, vol. 190, pp. 6881–6893, 2001.
7. A. Soulaïmani and M. Fortin, “Finite element solution of compressible viscous flows using conservative variables,” *Comp. Meth. Appl. Mech. Engrg*, vol. 118, pp. 319–350, 2001.

8. D.J. Tritton, *Physical Fluid Dynamics*, second edition, Oxford University Press, 1988.

# Swirl Flow

Swirl flow is an extension of the Incompressible Navier-Stokes application mode for axially symmetric geometries. By default, application modes based on the Navier-Stokes equations assume that the azimuthal velocity  $u_\phi$  in a 2D axisymmetric model is zero, but you can change this using the *Swirl velocity* application mode property. With swirl flow active, the only remaining assumption is that there are no variations in any model variable in the azimuthal direction.

For a cylindrical coordinate system, under the assumption that  $\partial/\partial\phi \equiv 0$ , Equation 7-6 and Equation 7-7 can be written

$$\frac{\partial u_r}{\partial r} + \frac{u_r}{r} + \frac{\partial u_z}{\partial z} = 0 \quad (7-22)$$

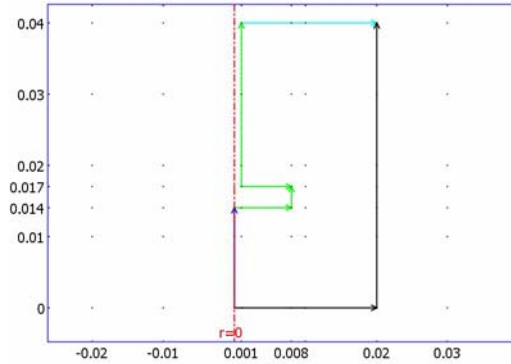
$$\begin{aligned} \rho \left( \frac{\partial u_r}{\partial t} + u_r \frac{\partial u_r}{\partial r} + u_z \frac{\partial u_r}{\partial z} - \frac{u_\phi^2}{r} \right) &= -\frac{\partial p}{\partial r} + \frac{1}{r} \frac{\partial}{\partial r} (r \tau_{rr}) + \frac{\partial \tau_{zr}}{\partial z} - \frac{\tau_{\phi\phi}}{r} + F_r \\ \rho \left( \frac{\partial u_z}{\partial t} + u_r \frac{\partial u_z}{\partial r} + u_z \frac{\partial u_z}{\partial z} \right) &= -\frac{\partial p}{\partial z} + \frac{1}{r} \frac{\partial}{\partial r} (r \tau_{rz}) + \frac{\partial \tau_{zz}}{\partial z} + F_z \\ \rho \left( \frac{\partial u_\phi}{\partial t} + u_r \frac{\partial u_\phi}{\partial r} + \frac{u_r u_\phi}{r} + u_r \frac{\partial u_\phi}{\partial z} \right) &= \frac{1}{r^2} \frac{\partial}{\partial r} (r^2 \tau_{r\phi}) + \frac{\partial \tau_{z\phi}}{\partial z} + F_\phi \end{aligned} \quad (7-23)$$

The Newtonian viscous stress tensor (Equation 7-5) in cylindrical coordinates assuming  $\partial/\partial\phi \equiv 0$  is given by

$$\begin{aligned} \tau_{rr} &= 2\eta \frac{\partial u_r}{\partial r}, & \tau_{zr} &= \tau_{rz} = \eta \left( \frac{\partial u_z}{\partial r} + \frac{\partial u_r}{\partial z} \right) \\ \tau_{zz} &= 2\eta \frac{\partial u_z}{\partial z}, & \tau_{r\phi} &= \tau_{\phi r} = \eta r \frac{\partial}{\partial r} \left( \frac{u_\phi}{r} \right) \\ \tau_{\phi\phi} &= 2\eta \frac{u_r}{r}, & \tau_{z\phi} &= \tau_{\phi z} = \eta \frac{\partial u_\phi}{\partial z} \end{aligned}$$

When the azimuthal velocity component  $u_\phi$  is included in the equations, the application mode by default denotes it  $w$ . The remaining components are then  $u = u_r$  and  $v = u_z$  just as when the swirl velocity is not present.

In an axisymmetric geometry, COMSOL Multiphysics assumes the symmetry axis to be at  $r = 0$ . This means that you have to build all models using the axisymmetry modes to the right of the axis  $r = 0$ . A correctly drawn example is shown in Figure 7-8.



*Figure 7-8: A geometry drawn in the 2D Axial Symmetry space dimension. Notice that the whole domain must be contained to the right of the symmetry line  $r = 0$ .*

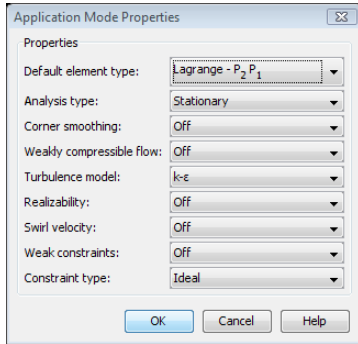
The Swirl velocity property does not affect boundary conditions, point settings and stabilization functionality, which are the same as for the standard Navier-Stokes implementation. See “Boundary Conditions” on page 183, “Point Settings” on page 193, and “Numerical Stability—Stabilization Techniques” on page 193. It is also possible to use corner smoothing with swirl flow. See “Corner Smoothing” on page 197.

### *The Swirl Velocity Property*

---

Most application modes for momentum transport have a swirl flow formulation that can be activated through the **Swirl velocity** property. Figure 7-9 shows the **Application**

**Mode Properties** dialog box, which you open by choosing **Properties** from the **Physics** menu.



*Figure 7-9: The Application Mode Properties dialog box for the k-ε Turbulence Model application mode. The Swirl velocity property is set to Off by default.*

The **Swirl velocity** property is **Off** by default. Set it to **On** to turn on swirl flow and include the azimuthal velocity in the model.

# Turbulent Flow

Turbulence often appears in industrial applications that involve fluid flow. It can be considered a characteristic state of the flow that you can only describe in a statistical manner. In this statistical representation of turbulent flow, the mean quantities are of main interest. However, fluctuations in different field quantities also play an important role.

In order to predict the behavior of turbulent systems, numerical modeling can reduce the costs of experiments and prototype equipment. In these turbulence models it is important to account for the nature of the fluctuations around the mean flow. This is included in the Heat Transfer Module as the Reynolds-averaged Navier-Stokes (RANS) equations.

The Navier-Stokes equations describe the basic phenomena of mass and momentum transport. These equations can also be used for turbulent flow simulations, although that requires a large number of elements to capture all the dynamics of the flow. An alternative is to consider the averaged equations, resulting in a hierarchy of equations and statistical unknowns. These unknown terms are modeled by equations known as *closure relations*, of which the eddy viscosity approach is the most common one. The Heat Transfer Module includes two such closure schemes:  $k$ - $\epsilon$  and  $k$ - $\omega$ .

Because the  $k$ - $\epsilon$  Turbulence Model application mode and the  $k$ - $\omega$  Turbulence Model application mode depend on the same fundamental principles, this section describes both.

## *Subdomain Equations*

---

Both application modes assume that the flow is incompressible and that the fluid is Newtonian. The flow is then in theory guided by the incompressible Navier-Stokes equations:

$$\begin{aligned}\rho \frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla)\mathbf{u} &= \nabla \cdot [-p\mathbf{I} + \eta(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)] + \mathbf{F} \\ \nabla \cdot \mathbf{u} &= 0\end{aligned}\tag{7-24}$$

Fundamental to the analysis of fluid flow is the Reynolds number:

$$\text{Re} = \frac{\rho UL}{\eta}\tag{7-25}$$

where  $U$  denotes a velocity scale, and  $L$  denotes a representative length. The Reynolds number represents the ratio between inertial and viscous forces. At low Reynolds numbers, viscous forces dominate and tend to damp out all disturbances, which leads to laminar flow. At high Reynolds numbers, the damping in the system is very low giving small disturbances the possibility to grow by nonlinear interactions. If the Reynolds number is high enough, the fluid flow field eventually ends up in a chaotic state called turbulence.

### REYNOLDS-AVERAGED NAVIER-STOKES (RANS) EQUATIONS

Once flow has become turbulent, all quantities fluctuate in time and space, and there is little advantage in having an exact and detailed picture of the flow. Instead, an averaged representation almost always provides sufficient information about the flow.

The Reynolds-averaged representation of turbulent flows divides the flow quantities into an averaged value and a fluctuating part,

$$\phi = \bar{\phi} + \phi' \quad (7-26)$$

where  $\phi$  can represent any scalar quantity of the flow. In general, the mean value can vary in space and time. This is exemplified in Figure 7-10, which shows time averaging of one component of the velocity vector for nonstationary turbulence. The unfiltered flow has a time scale  $\Delta t_1$ . After a time filter with width  $\Delta t_2 \gg \Delta t_1$  has been applied, there is a fluctuating part,  $u'_i$ , and an average part,  $U_i$ . Because the flow field also varies on a time scale longer than  $\Delta t_2$ ,  $U_i$  is still time dependent but is much smoother than the unfiltered velocity  $u_i$ .

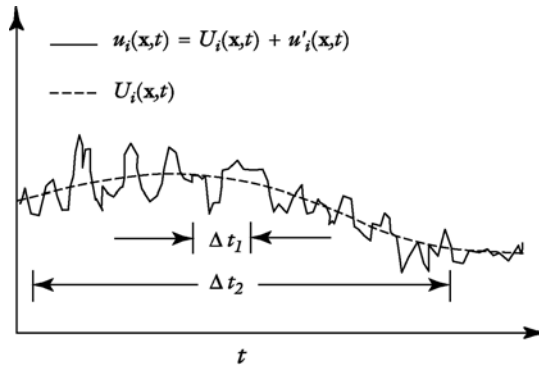


Figure 7-10: The unfiltered velocity component  $u_i$ , with a time scale  $\Delta t_1$  and the averaged velocity component,  $U_i$ , with time scale  $\Delta t_2$ .

Decomposition of flow fields into an averaged part and a fluctuating part, followed by insertion into the Navier-Stokes equation, then averaging, gives the Reynolds-averaged Navier-Stokes (RANS) equations:

$$\rho \frac{\partial \mathbf{U}}{\partial t} + \rho \mathbf{U} \cdot \nabla \mathbf{U} + \nabla \cdot (\overline{\rho \mathbf{u}' \otimes \mathbf{u}'}) = -\nabla P + \nabla \cdot \eta (\nabla \mathbf{U} + (\nabla \mathbf{U})^T) + \mathbf{F} \quad (7-27)$$

$$\nabla \cdot \mathbf{U} = 0$$

where  $\mathbf{U}$  is the averaged velocity field.  $\otimes$  is the outer vector product. A comparison with Equation 7-24 indicates that the only difference is the appearance of the last term on the left hand side of Equation 7-27. This term represents interaction between the fluctuating velocities and is called the Reynolds stress tensor. This means that to obtain the mean flow characteristics, you need information about the small-scale structure of the flow. In this case, that information is the correlation between fluctuations in different directions.

The governing equations for the components of the Reynolds stress tensor follow from a procedure similar to that above. In 3D, this introduces six additional equations and six additional unknowns. Because the resulting equations contain additional unknowns emanating from higher-order statistics, this is still not sufficient for solving flow problems. Resolving this dilemma requires making some assumptions about the flow, a procedure known as imposing a *closure* of a turbulence model.

### EDDY VISCOSITY CLOSURE

The most common by far way to model turbulence is by assuming the turbulence to be of a purely diffusive nature. The deviating part of the Reynolds stress is then expressed by

$$\rho (\overline{\mathbf{u}' \otimes \mathbf{u}'}) - \frac{\rho}{3} \text{trace}(\overline{\mathbf{u}' \otimes \mathbf{u}'}) \mathbf{I} = -\eta_T (\nabla \mathbf{U} + (\nabla \mathbf{U})^T) \quad (7-28)$$

where  $\eta_T$  is the *eddy viscosity*, also known as the turbulent viscosity. The spherical part can be written

$$\frac{\rho}{3} \text{trace}(\overline{\mathbf{u}' \otimes \mathbf{u}'}) \mathbf{I} = \frac{2}{3} \rho k \quad (7-29)$$

where  $k$  is the turbulent kinetic energy. In simulations of incompressible flows, this term is included in the pressure, but when the absolute pressure level is of importance (in compressible flows, for example) this term must be explicitly modeled.



### The $k$ - $\varepsilon$ Turbulence Model

The  $k$ - $\varepsilon$  model is one of the most used turbulence models for industrial applications. The Heat Transfer Module includes the standard  $k$ - $\varepsilon$  model (Ref. 1). This model introduces two additional transport equations and two dependent variables: the turbulent kinetic energy,  $k$ , and the dissipation rate of turbulence energy,  $\varepsilon$ . Turbulent viscosity is modeled by

$$\eta_T = \rho C_\mu \frac{k^2}{\varepsilon} \quad (7-30)$$

where  $C_\mu$  is a model constant.

The transport equation for  $k$  can be derived by taking the trace of the equations for the Reynolds stresses:

$$\rho \frac{\partial k}{\partial t} - \nabla \cdot \left[ \left( \eta + \frac{\eta_T}{\sigma_k} \right) \nabla k \right] + \rho \mathbf{U} \cdot \nabla k = \frac{1}{2} \eta_T (\nabla \mathbf{U} + (\nabla \mathbf{U})^T)^2 - \rho \varepsilon \quad (7-31)$$

An equation for  $\varepsilon$  can be derived in a similar manner. That equation is, however, impossible to model on a term-by-term basis. Instead, all terms that do not have an equivalent term in the  $k$  equation are discarded. The resulting equation reads:

$$\rho \frac{\partial \varepsilon}{\partial t} - \nabla \cdot \left[ \left( \eta + \frac{\eta_T}{\sigma_\varepsilon} \right) \nabla \varepsilon \right] + \rho \mathbf{U} \cdot \nabla \varepsilon = \frac{1}{2} C_{\varepsilon 1} \frac{\varepsilon}{k} \eta_T (\nabla \mathbf{U} + (\nabla \mathbf{U})^T)^2 - \rho C_{\varepsilon 2} \frac{\varepsilon^2}{k} \quad (7-32)$$

The model constants in the above equations are determined from experimental data (Ref. 1); their values are listed in Table 7-1.

TABLE 7-1: MODEL CONSTANTS IN EQUATION 7-30, EQUATION 7-31, AND EQUATION 7-32

CONSTANT	VALUE
$C_\mu$	0.09
$C_{\varepsilon 1}$	1.44
$C_{\varepsilon 2}$	1.92
$\sigma_k$	1.0
$\sigma_\varepsilon$	1.3

The  $k$ - $\varepsilon$  turbulence model relies on several assumptions, the most important of which are that the Reynolds number is high enough and that the turbulence is in equilibrium in boundary layers, which means that production equal dissipation. The assumptions limit the accuracy of the model, since this is not always true. The spatial extension of recirculation zones, for example, is underpredicted by the  $k$ - $\varepsilon$  turbulence model.

Furthermore, in the description of rotating flows, the model often shows poor agreement with experimental data. In most cases, the limited accuracy is often a fair trade for the amount of computational resources saved compared to more complicated turbulence models.

### *The $k$ - $\omega$ Turbulence Model*

---

Much of the  $k$ - $\varepsilon$  model's inaccuracy can be attributed to the rather inaccurate modeling of the  $\varepsilon$  equation. An alternative to modeling the dissipation is to model transport of the dissipation per unit turbulent kinetic energy,  $\omega$ . The turbulent viscosity is modeled by:

$$\eta_T = \rho \frac{k}{\omega} \quad (7-33)$$

The transport of turbulent kinetic energy and  $\omega$  is given by

$$\rho \frac{\partial k}{\partial t} + \rho \mathbf{U} \cdot \nabla k = \nabla \cdot [(\eta + \sigma_k \eta_T) \nabla k] + \frac{1}{2} \eta_T (\nabla \mathbf{U} + \nabla \mathbf{U}^T)^2 - \rho \beta_k k \omega \quad (7-34)$$

and

$$\rho \frac{\partial \omega}{\partial t} + \rho \mathbf{U} \cdot \nabla \omega = \nabla \cdot [(\eta + \sigma_\omega \eta_T) \nabla \omega] + \frac{\alpha}{2} \eta_T \frac{\omega}{k} (\nabla \mathbf{U} + \nabla \mathbf{U}^T)^2 - \beta \rho \omega^2 \quad (7-35)$$

where the previous equations use the following closure constants and functions:

$$\alpha = \frac{13}{25}, \quad \beta = \beta_0 f_\beta, \quad \beta_k = \beta_{k,0} f_{\beta,k}, \quad \sigma_\omega = \sigma_k = \frac{1}{2}$$

$$\beta_0 = \frac{9}{125}, \quad f_\beta = \frac{1 + 70\chi_\omega}{1 + 80\chi_\omega}, \quad \chi_\omega \equiv \left| \frac{\Omega_{ij} \Omega_{jk} S_{ki}}{(\beta_{k,0} \omega)^3} \right|$$

$$\beta_{0,k} = \frac{9}{100}, \quad f_{\beta,k} = \begin{cases} 1 & \chi_k \leq 0 \\ \frac{1 + 680\chi_k^2}{1 + 400\chi_k^2} & \chi_k > 0 \end{cases}, \quad \chi_k \equiv \frac{1}{\omega^3} \nabla k \cdot \nabla \omega$$

The tensors  $\Omega_{ij}$  and  $S_{ij}$  are the mean rotation-rate tensor and the mean strain-rate tensor, respectively, defined by

$$\Omega_{ij} = \frac{1}{2} \left( \frac{\partial U_i}{\partial x_j} - \frac{\partial U_j}{\partial x_i} \right), \quad S_{ij} = \frac{1}{2} \left( \frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \quad (7-36)$$

The use of the correction factors  $f_\beta$  and  $f_{\beta,k}$  is a modification of the original (1988)  $k$ - $\omega$  model that Wilcox made for the updated model from 1998. The corrections improve the accuracy for free-shear flows significantly by decreasing the spreading of such flows.

There are no guarantees that one turbulence model provides better results than the other for a certain flow case. As a general rule, the  $k$ - $\omega$  model behaves better close to walls and usually predicts free shear flows more accurately than the  $k$ - $\epsilon$  model. For external flows, on the other hand, results obtained by the  $k$ - $\omega$  model have a tendency to be strongly dependent on the free-stream values of  $k$  and  $\omega$ .

It is often harder to reach convergence with the  $k$ - $\omega$  model than with the  $k$ - $\epsilon$  model. A way to reach convergence is to first solve the problem with the  $k$ - $\epsilon$  model and then change to the  $k$ - $\omega$  model. The application mode automatically changes the initial values so that the  $k$ - $\epsilon$  result becomes the initial guess for the  $k$ - $\omega$  model.

### *Boundary Conditions*

---

The turbulence application modes have the same boundary types as the Incompressible Navier-Stokes application modes; see “Boundary Conditions” on page 183. With exception for the boundary type **Wall**, the functionality described for the incompressible Navier-Stokes equations applies to the Reynolds-averaged Navier-Stokes equations as well. Note that any expression that in the incompressible Navier-Stokes formulation is of the form  $g(\eta)$  is of the form  $g(\eta + \eta_T)$  for the turbulence application modes.

All boundary conditions need additional conditions for the turbulence transport equations. The following sections describe these boundary conditions for each boundary type.

#### **WALL**

##### *Slip*

The Slip boundary condition assumes that there is no viscous interaction between the wall and the fluid. In addition to the conditions prescribed for the momentum equations, this boundary condition also prescribes

$$\mathbf{n} \cdot \left( \left( \eta + \frac{\eta_T}{\sigma_k} \right) \nabla k - \rho \mathbf{U} k \right) = 0 \quad / \quad (\mathbf{n} \cdot ((\eta + \sigma_k \eta_T) \nabla k - \rho \mathbf{U} k) = 0)$$

$$\mathbf{n} \cdot \left( \left( \eta + \frac{\eta_T}{\sigma_\varepsilon} \right) \nabla \varepsilon - \rho \mathbf{U} \varepsilon \right) = 0 \quad / \quad (\mathbf{n} \cdot ((\eta + \sigma_\omega \eta_T) \nabla \omega - \rho \mathbf{U} \omega) = 0)$$

where the expressions to the left are used for the  $k$ - $\varepsilon$  Turbulence Model and the expressions to the right are used for the  $k$ - $\omega$  Turbulence Model.

#### *Logarithmic Wall Function*

Turbulence close to a solid wall is very different from isotropic free-stream turbulence. This must be accounted for in a proper model.

There are basically two approaches to account for solid walls subjected to turbulent flow. The first approach, used for low Reynolds number turbulence models, modifies the equations by additional terms and factors that account for near-wall effects. In such a case, you must refine the mesh near the wall to resolve the viscous sublayer. Such methods are of interest for moderate Reynolds numbers, where near-wall resolution lead to a reasonable number of elements.

In the second approach, which is used by these application modes, an empirical relation between the value of velocity and wall friction replaces the thin boundary layer near the wall. Such relations are known as wall functions and are accurate for high Reynolds numbers and situations where pressure variations along the wall are not very large. However, this approach can often be used outside its frame of validity with reasonable success.

Logarithmic wall functions applied to finite elements assume that the computational domain begins a distance  $\delta_w$  from the real wall. They also assume that the flow is parallel to the wall and that the velocity can be described by

$$U^+ = \frac{U}{u_\tau} = \frac{1}{\kappa} \ln \left( \frac{\delta_w}{l^*} \right) + C^+ \quad (7-37)$$

Here  $U$  is the velocity parallel to the wall,  $u_\tau$  is the friction velocity defined by

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}}, \quad (7-38)$$

$\kappa$  denotes the Kármán's constant (about 0.42), and  $C^+$  is a universal constant for smooth walls. In the application mode,  $C^+$  is defined as an application scalar variable named `Cp1us_chns` and has a default value of 5.5 (it is possible to change this value in

the **Application Scalar Variables** dialog box). Further,  $l^*$  is known as the viscous length scale and is defined by

$$l^* = \frac{\eta}{\rho u_\tau} \quad (7-39)$$

You must specify the distance  $\delta_w$  or its equivalent in viscous units,  $\delta_w^+ = \delta_w/l^*$ . Their internal relation is given by

$$\delta_w^+ = \frac{\rho C_\mu^{1/4} k^{1/2} \delta_w}{\eta} \quad (7-40)$$

The logarithmic wall functions are formally valid for values of  $\delta_w^+$  between 30 and 100. For high Reynolds number, the upper limit can be extended to a few hundred.  $\delta_w^+$  is available as the variable `dwp1us_chns` and is defined on the boundaries. Because the wall function assumes that the flow is parallel to the wall, the velocity component perpendicular to the wall is zero.

The boundary conditions for  $k$ ,  $\varepsilon$ , and  $\omega$  are derived from the assumption that turbulent production equals the dissipation:

$$\nabla k \cdot \mathbf{n} = 0, \quad \varepsilon = \frac{C_\mu^{3/4} k^{3/2}}{\kappa \delta_w}, \quad \omega = \frac{k^{1/2}}{C_\mu^{1/4} \kappa \delta_w} \quad (7-41)$$

### Sliding Wall

This boundary condition assumes that the wall is moving with a given velocity tangential to the wall. It extends the logarithmic wall function concept by changing Equation 7-37 to

$$\frac{(U - U_w)}{u_\tau} = \frac{1}{\kappa} \ln \left( \frac{\delta_w}{l^*} \right) + C^+$$

where  $U_w$  is calculated in the same way as described for the Incompressible Navier-Stokes application mode in the section “Sliding Wall” on page 186.

## INLET BOUNDARIES

At inlet boundaries, the functionality for the RANS equations are the same as described for the incompressible Navier-Stokes in the section “Inlet” on page 187. In addition, values for the two turbulent quantities need to be specified ( $k$  and  $\varepsilon$ , or  $k$  and  $\omega$ ). Alternatively, you can specify a turbulent length scale,  $\mathbf{L}_T$ , and a turbulence intensity,  $\mathbf{I}_T$ . They are related to the turbulence variables via

$$k = \frac{3}{2}(|\mathbf{U}|I_T)^2, \quad \varepsilon = C_\mu^{3/4} \frac{k^{3/2}}{L_T}, \quad \omega = C_\mu^{-1/4} \frac{k^{1/2}}{L_T} \quad (7-42)$$

A value below 0.1% is a low intensity. Good wind tunnels can produce values of  $I_T$  as low as 0.05%. Fully turbulent flows have intensities between five and ten percent.

The turbulent length scale is a measure of the size of the eddies that are not resolved. For free-stream flows these are typically very small (in the order of centimeters). The length scale cannot be zero, however, because that would imply infinite dissipation. You can use the following table as a guideline when specifying  $L_T$  (Ref. 3):

TABLE 7-2: TURBULENT LENGTH SCALES FOR TWO-DIMENSIONAL FLOWS

FLOW CASE	$L_T$	$L$
Mixing layer	$0.07L$	Layer width
Plane jet	$0.09L$	Jet half width
Wake	$0.08L$	Wake width
Axisymmetric jet	$0.075L$	Jet half width
Boundary layer ( $\partial p/\partial x = 0$ )		
– Viscous sublayer and log-layer	$\kappa l_w(1 - \exp(-l_w^+/26))$	Boundary layer thickness
– Outer layer	$0.09L$	
Pipes and channels (fully developed flows)	$0.07L$	Pipe diameter or channel width

where  $l_w$  is the wall distance and  $l_w^+ = l_w/l^*$  is the wall distance in viscous units.

### OUTLET BOUNDARIES

At outlet boundaries, the functionality for the RANS equations are the same as described for the incompressible Navier-Stokes equations in the section “Outflow” on page 188. In addition, convective flux conditions are prescribed for the turbulence variables:

$$\begin{aligned} \mathbf{n} \cdot \nabla k &= 0 \\ \mathbf{n} \cdot \nabla \varepsilon &= 0 \quad / \quad \mathbf{n} \cdot \nabla \omega = 0 \end{aligned}$$

### SYMMETRY BOUNDARIES

The equations for this condition read

$$\begin{aligned}
& (-p\mathbf{I} + (\eta + \eta_T)(\nabla\mathbf{U} + (\nabla\mathbf{U})^T)) \cdot \mathbf{t} = 0 \\
& \mathbf{n} \cdot \mathbf{U} = 0 \\
& \mathbf{n} \cdot \left( \left( \eta + \frac{\eta_T}{\sigma_k} \right) \nabla k - \rho \mathbf{U} k \right) = 0 \quad / \quad (\mathbf{n} \cdot ((\eta + \sigma_k \eta_T) \nabla k - \rho \mathbf{U} k) = 0) \\
& \mathbf{n} \cdot \left( \left( \eta + \frac{\eta_T}{\sigma_\varepsilon} \right) \nabla \varepsilon - \rho \mathbf{U} \varepsilon \right) = 0 \quad / \quad (\mathbf{n} \cdot ((\eta + \sigma_\omega \eta_T) \nabla \omega - \rho \mathbf{U} \omega) = 0)
\end{aligned}$$

where the constraints on the turbulent quantities express zero total normal flux. The expressions to the left are used for the  $k$ - $\varepsilon$  Turbulence Model and the expressions to the right are used for the  $k$ - $\omega$  Turbulence Model.

### OPEN BOUNDARIES

The expressions for the momentum equations are the same as for the Incompressible Navier-Stokes (see page 191), except that  $\eta + \eta_T$  replaces  $\eta$ . Homogeneous Neumann conditions are prescribed for the turbulence variables

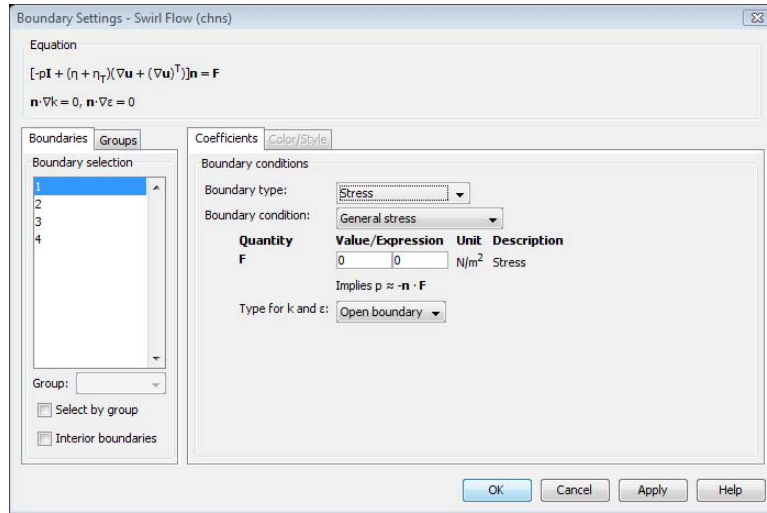
$$\begin{aligned}
& \mathbf{n} \cdot \nabla k = 0 \\
& \mathbf{n} \cdot \nabla \varepsilon = 0 \quad / \quad \mathbf{n} \cdot \nabla \omega = 0
\end{aligned}$$

Note that these are the same boundary conditions used for outlet boundaries. This expresses that the turbulent characteristic of whatever is outside the computational domain is guided by the flow inside the computational domain. Such an assumption is physically reasonable as long as relatively small amounts of fluid enters via an open boundary. Large amount of fluids may, however, never enter via an open boundary. If this is the case, switch to an inlet condition instead.

### STRESS CONDITIONS

Because this type of boundary condition is very general, different conditions have to be prescribed for the turbulence variables depending on what stresses that are prescribed. As Figure 7-11 shows, there are three choices for the turbulence variables:

**Inlet, Outlet, and Open boundary.** The functionality of these boundary types are described in the previous sections.



*Figure 7-11: The Boundary Settings dialog box for the  $k$ - $\epsilon$  Turbulence Model application mode. For the Stress boundary type, there are three choices for the type for the turbulence variables.*

### *Numerical Stability—Stabilization Techniques*

For turbulence models, the Stabilization page in the Subdomain Settings dialog box contains two sub-pages; one for the Navier-Stokes equations and one for the turbulence equations. These sub-pages are shown in Figure 7-12 and Figure 7-13. You can switch between the sub-pages by using the list at the top of the Stabilization



page. The Navier-Stokes settings are described in the section Numerical Stability—Stabilization Techniques on page 193.

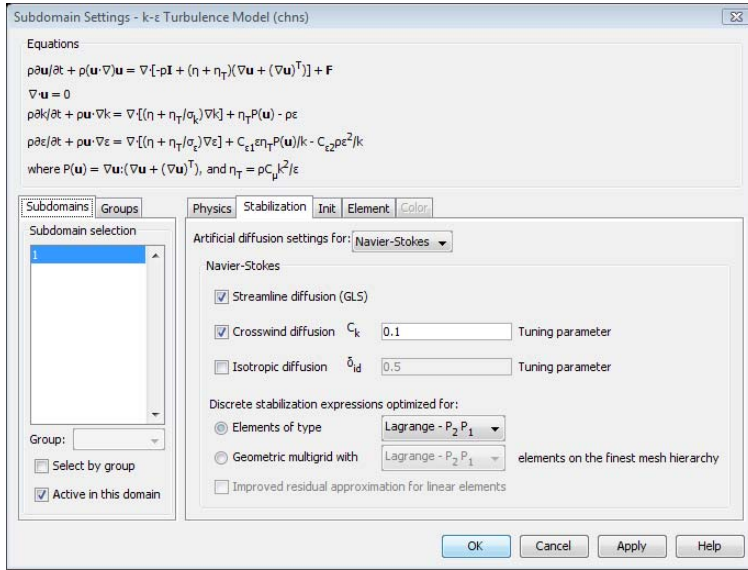


Figure 7-12: The artificial diffusion settings for Navier-Stokes on the Stabilization page of the Subdomain Settings dialog box.

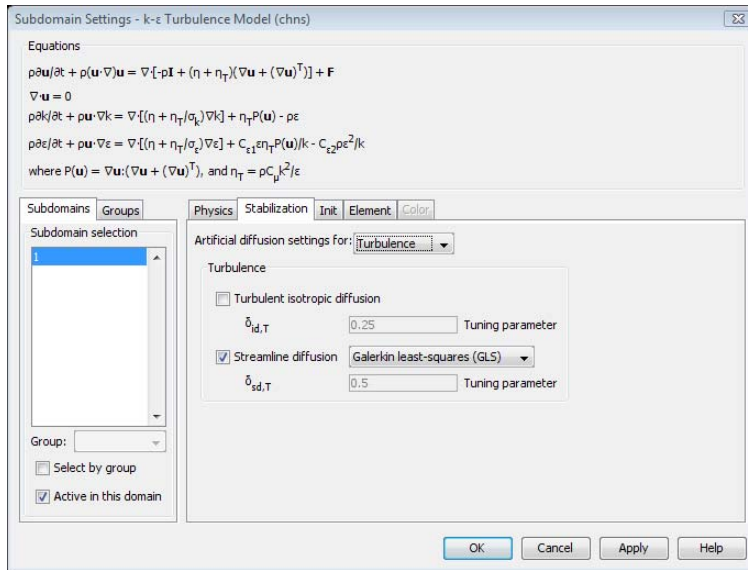


Figure 7-13: The Turbulence sub-page of the Stabilization page of the Subdomain Settings dialog box.

The turbulence equations can be stabilized by

- Isotropic Diffusion
- Streamline Diffusion

#### Isotropic Diffusion

Oscillations in the turbulence equations can often lead to divergence of the whole simulation. If a turbulence simulation diverges, convergence can often be obtained by adding isotropic diffusion to the turbulence equations.

The isotropic diffusion for the turbulence equations is proportional to the production of turbulent kinetic energy,  $k$ . (The isotropic diffusion described in the section “Isotropic Diffusion” on page 500 of the *COMSOL Multiphysics Modeling Guide* is proportional to the velocity).

A rule of thumb is that the tuning parameter  $\delta_{sd,T}$  should be in the range (0, 0.5] and set as small as possible to obtain an accurate solution.

## STREAMLINE DIFFUSION

There are four versions of streamline diffusion available for the turbulence equations

- Anisotropic diffusion
- Galerkin least-squares (GLS)
- Petrov-Galerkin
- Petrov-Galerkin/Compensated

Anisotropic diffusion and GLS are described in Chapter 17, “Stabilization Techniques,” in the *COMSOL Multiphysics Modeling Guide*. GLS is the default technique.

### *Petrov-Galerkin*

The Petrov-Galerkin streamline diffusion is almost the same as the SUPG method described in the section “Streamline Upwind Petrov-Galerkin (SUPG)” on page 502 in the *COMSOL Multiphysics Modeling Guide*. The only difference is an intrinsic time scale that the method uses. The turbulence equations use the original time scale designed for transport equations dominated by convection (see Ref. 4).

There is no strict theory governing the value of the tuning parameter,  $\delta_{sd,T}$ , but a good rule of thumb is to set  $\delta_{sd,T} = 0.5/p$  where  $p$  is the order of the basis functions.

This version of Petrov-Galerkin give results similar to GLS if the flow is everywhere dominated by convection. In regions that are dominated by diffusion, this method does however introduce a bit too much diffusion. Hence, GLS is in most cases the preferred method.

### *Petrov-Galerkin/Compensated*

The Petrov-Galerkin/Compensated method uses a modified intrinsic time scale that completely eliminates the streamline diffusion contributions in regions dominated by diffusion (Ref. 4).

There is no strict theory governing the value of the tuning parameter,  $\delta_{sd,T}$ , but a good rule of thumb is to select  $\delta_{sd,T} = 0.5/p$  where  $p$  is the order of the basis functions.

The drawback is that this formulation can cause discontinuities and that it introduces additional nonlinearities. Hence, GLS is in most cases the preferred method.

### *Realizability Constraints*

---

A stagnation point is defined as a point on a surface where a streamline ends. Typical examples are the leading edge of a flat plate or the front edge of a wing profile. The

equilibrium assumption made when deriving the turbulent transport equations are not valid in such points. The result is an excessive production of turbulence known as the stagnation point anomaly. To cope with this issue, you must activate a realizability constraint.

The eddy-viscosity model of the Reynolds stress tensor can be written

$$\overline{\rho u_i u_j} = -2\eta_T S_{ij} + \frac{2}{3}\rho k \delta_{ij}$$

where  $\delta_{ij}$  is the Kronecker delta and  $S_{ij}$  is the strain rate tensor. Durbin (Ref. 5) showed that the fact that diagonal elements of the Reynolds stress tensor must be nonnegative can be used to derive the following constraint on the turbulent viscosity:

$$\eta_T \leq \frac{\rho k}{3 \cdot \max(\lambda_\alpha)} \quad (7-43)$$

Here,  $\max(\lambda_\alpha)$  is the largest eigenvalue of the strain rate tensor. When using realizability constraint for the  $k$ - $\varepsilon$  model, the turbulent viscosity is given by

$$\eta_T = \min\left(\rho C_\mu \frac{k^2}{\varepsilon}, \frac{\rho k}{3 \cdot \max(\lambda_\alpha)}\right) \quad (7-44)$$

The corresponding equivalence for the  $k$ - $\omega$  model is also available. In two dimensions, it is possible to derive an exact expression for the right-hand side of Equation 7-44:

$$\eta_T \leq \frac{\rho k \sqrt{2}}{3 \sqrt{S_{ij} S_{ij}}} \quad (7-45)$$

In three dimensions, however, you can only find an approximate upper limit without calculating all the eigenvalues of  $S_{ij}$ :

$$\eta_T \leq \frac{\rho k}{\sqrt{6} \sqrt{S_{ij} S_{ij}}} \quad (7-46)$$

To activate the realizability constraint, use the **Application Mode Properties** dialog box (Figure 7-14), which you open by choosing **Properties** from the **Physics** menu.

Because an active realizability constraint implies a lower turbulent viscosity than the unconstrained case, the convergence will be hampered. If an active realizability

constraint hinders convergence, try to first solve without realizability and then activate the constraint.

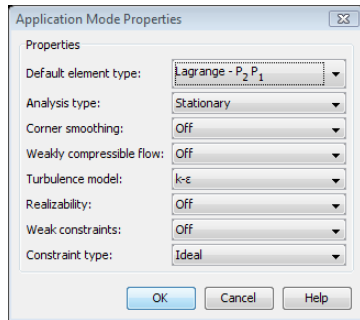


Figure 7-14: The Application Mode Properties dialog box for the  $k$ - $\epsilon$  Turbulence Model application mode.

### The Turbulence Model Property

---

The **Turbulence model** property in the **Application Mode Properties** dialog box (see Figure 7-14) has the following options

- **None**
- **$k$ - $\epsilon$**
- **$k$ - $\omega$**

This property can be used to change between the  $k$ - $\epsilon$  Turbulence Model application mode and the  $k$ - $\omega$  Turbulence Model application mode. This can be useful because it can be hard to reach convergence using the  $k$ - $\omega$  model. A standard trick is then to start from a  $k$ - $\epsilon$  solution. When changing turbulence model, COMSOL automatically updates the **Initial value** edit fields in the **Subdomain Settings** dialog box.

The **Turbulence model** property is available for many application modes that per default do not utilize any turbulence model.

### Solver Settings

---

Navier-Stokes equations and the turbulence equations form an equation system that is very hard to solve in a coupled manner. It is therefore praxis to split the equation systems in two parts; one part that contains the Navier-Stokes equations and another part with the turbulence equations. Hence, the default solver for turbulence problems is the segregated solver.

The default setting for the turbulence application modes is that for each iteration made for the Navier-Stokes equations, three iterations are made for the turbulence equations. The reason is the nonlinear source terms in the turbulence equations that must be in balance before the next iteration can be made for the Navier-Stokes equations. For very complicated problems, it can be necessary to increase the number of iterations made for the turbulence equations.

For more information on the segregated solvers see The Solvers and Their Settings on page 385 in the *COMSOL Multiphysics User's Guide*.

### *References*

---

1. D.C. Wilcox, *Turbulence Modeling for CFD*, DCW Industries Inc., 1998.
2. D.M. Driver and H.L. Seegmiller, "Features of a reattaching turbulent shear layer in diverging channel flow," *AIAA Journal*, vol. 23, pp. 163–171, 1985.
3. H.K. Versteeg and W. Malalasekera, *An introduction to Computational Fluid Dynamics*, Prentice Hall, 1995.
4. C. Johnson, *Numerical solution of partial differential equations by the finite element method*, Studentlitteratur, 1987.
5. A. Durbin, "On the  $k$ - $\epsilon$  stagnation point anomaly," *International Journal of Heat and Fluid Flow*, vol. 17, pp. 89–90, 1986.

# Weakly Compressible Navier-Stokes

The Weakly Compressible Navier-Stokes application mode contains the fully compressible formulation of the continuity equation and momentum equations:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) &= 0 \\ \rho \frac{\partial \mathbf{u}}{\partial t} + \rho \mathbf{u} \cdot \nabla \mathbf{u} &= -\nabla p + \nabla \cdot \left( \eta (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) - \left( \frac{2}{3} \eta - \kappa_{dv} \right) (\nabla \cdot \mathbf{u}) \mathbf{I} \right) + \mathbf{F} \end{aligned} \quad (7-47)$$

The stress tensor used in Equation 7-47 describes a Newtonian fluid, with an added term  $\kappa_{dv}$ . This term expresses the deviation from Stokes' assumption, which states that the fluid particles are in thermodynamic equilibrium with their neighbors. It is very rare that a fluid shows a significant deviation from Stokes' assumption, and  $\kappa_{dv}$  is therefore by default set to zero.

---

**Note:** Note that the Weakly Compressible Navier-Stokes application was designed solely to model flow with variable density, where a second application mode—typically a heat balance—controls the density. This coupling is automatically set up by the Non-Isothermal Flow predefined multiphysics coupling (see “Non-Isothermal Flow” on page 228).

---

## THE MACH NUMBER LIMIT

An important dimensionless number in fluid dynamics is the Mach number,  $\text{Ma}$ , defined by

$$\text{Ma} = \frac{|\mathbf{u}|}{a} \quad (7-48)$$

where  $a$  is the speed of sound. A flow is formally incompressible when  $\text{Ma} = 0$ . This is theoretically achieved by letting the speed of sound tend to infinity. The Navier-Stokes equations will then have the numerical property that a disturbance anywhere in the computational domain will instantaneously spread to the entire domain. This results in a parabolic equation system.

The fully compressible Navier-Stokes equations, Equation 7-1 through Equation 7-3, have a finite speed of sound and hence a Mach number larger than zero. This will have

no numerical significance as long as the Mach number is well below one. However, when the Mach number approaches unity, the equations will turn from parabolic to hyperbolic. When this happens, the numerical properties of the equation changes, one of several implications being that the boundary conditions used for incompressible Navier-Stokes equations become invalid. The Weakly Compressible Navier-Stokes application mode uses the same boundary condition as the Incompressible Navier-Stokes application mode, which implies that the Weakly Compressible Navier-Stokes application mode cannot be used for flows with Mach number larger than or equal to one.

The practical Mach number limit is however lower than one. The first reason for this is the sound wave transport term that has been neglected in the General Heat Transfer application mode (see “Convection and Conduction” on page 29). This term becomes important already at moderate Mach numbers. The second reason is that already at moderate Mach number, the fully compressible Navier-Stokes equations start to display very sharp gradients. To handle these gradient, special numerical techniques are needed. It is impossible to give an exact limit where the low Mach number regime ends and the moderate Mach number regime begins, but a rule of thumb is that the Mach number effects start to appear at  $Ma = 0.3$ .



## Stabilization Settings for Weakly Compressible Navier-Stokes

The **Stabilization** page in the **Subdomain Settings** dialog box for the Weakly Compressible Navier-Stokes application mode is shown in Figure 7-15.

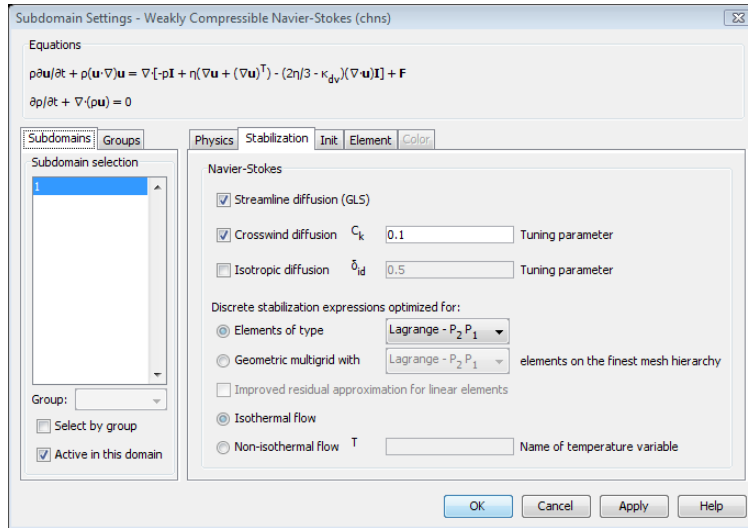


Figure 7-15: The Stabilization page for the Weakly Compressible Navier-Stokes application mode.

All settings and their functionality, except the two buttons at the bottom of the page, are the same as for the Incompressible Navier-Stokes application mode (see “Numerical Stability—Stabilization Techniques” on page 193).

When the **Isothermal flow** button is selected, the stabilization expressions for the Navier-Stokes equations will assume that the density,  $\rho$ , is independent of the temperature.

If the density is a function of the temperature, the momentum equations, continuity equation and temperature equation form a closely coupled system of equations. For optimal performance, this coupling must be accounted for in the stabilization expressions. This is achieved by clicking the **Nonisothermal flow** button and entering the name of the temperature variable in the corresponding edit field. Full functionality is achieved if:

- 1 Galerkin least-squares streamline diffusion is activated for both the Weakly Compressible Navier-Stokes application mode and for the General Heat Transfer application mode.
- 2 If crosswind diffusion is activated for both the Weakly Compressible Navier-Stokes application mode and for the General Heat Transfer application mode. The crosswind diffusion type for the General Heat Transfer application mode must be System stabilization.

See “Stabilization Settings” on page 38 for stabilization settings for the General Heat Transfer application mode.

### *Other Settings*

---

The Weakly Compressible Navier-Stokes application mode shares boundary conditions, point settings and, solver settings with the Incompressible Navier-Stokes application mode; see “Boundary Conditions” on page 183, “Point Settings” on page 193, and Solver Settings on page 199. The Corner Smoothing property can be applied to the Weakly Compressible Navier-Stokes application mode as well; see “Corner Smoothing” on page 197.

### *Turbulent Weakly Compressible Flow*

---

If the Reynolds average (see the section “Turbulent Flow” on page 206) is applied to Equation 7-47, terms of the form  $\overline{\rho' u'}$  will appear and need to be modeled. To avoid this, a density based average, known as the Favre average, is introduced:

$$\tilde{u}_i = \frac{1}{\bar{\rho}} \lim_{T \rightarrow \infty} \frac{1}{T} \int_t^{t+T} \rho(\mathbf{x}, \tau) u_i(\mathbf{x}, \tau) d\tau \quad (7-49)$$

It follows from Equation 7-49 that

$$\bar{\rho} \tilde{u}_i = \overline{\rho u_i} \quad (7-50)$$

and a variable  $u_i$  is decomposed in a mass averaged component,  $\tilde{u}_i$ , and a fluctuating component,  $u_i''$  according to

$$u_i = \tilde{u}_i + u_i'' \quad (7-51)$$

Using Equation 7-50 and Equation 7-51 along with some modeling assumption for compressible flows (Ref. 1), Equation 7-47 can be written in the form

$$\begin{aligned} \frac{\partial \bar{p}}{\partial t} + \frac{\partial}{\partial x_i}(\bar{\rho} \tilde{u}_i) &= 0 \\ \bar{\rho} \frac{\partial \tilde{u}_i}{\partial t} + \bar{\rho} \tilde{u}_j \frac{\partial \tilde{u}_i}{\partial x_j} &= -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j} \left( \eta \left( \frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} \right) - \frac{2}{3} \eta \frac{\partial \tilde{u}_k}{\partial x_k} \delta_{ij} - \overline{\rho u_j'' u_i''} \right) + \bar{F}_i \end{aligned} \quad (7-52)$$

The Favre-averaged Reynolds stress tensor is modeled using the same argument as for incompressible flows:

$$-\overline{\rho u_j'' u_i''} = \eta_T \left( \frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} \right) - \frac{2}{3} \left( \eta_T \frac{\partial \tilde{u}_k}{\partial x_k} + \bar{\rho} k \right) \delta_{ij}$$

where  $k$  is the turbulent kinetic energy. Comparing Equation 7-52 to its incompressible counterpart (Equation 7-27), it can be seen that apart from the term  $-(2/3)\bar{\rho}k\delta_{ij}$ , the compressible and incompressible formulations are exactly the same, except that the free variables are  $\tilde{u}_i$  instead of  $U_i = \bar{u}_i$ . More information on modeling of turbulent compressible flows can be found in Ref. 1 and Ref. 2.

The turbulent transport equations ( $k$ - $\varepsilon$  or  $k$ - $\omega$ ) are used in their fully compressible formulations (Ref. 3).

## References

---

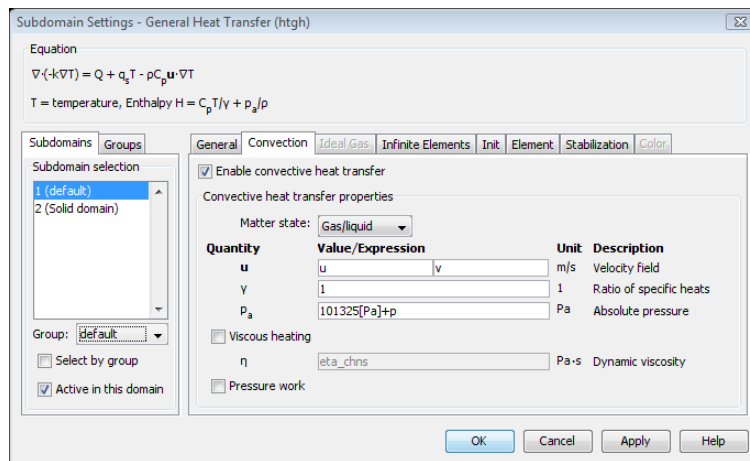
1. J. Larsson, "Numerical Simulation of Turbulent Flows for Turbine Blade Heat Transfer," *Doctoral thesis for the degree of Doctor of Philosophy*, Chalmers University of Technology, Sweden, 1998.
2. D.C. Wilcox, "Turbulence Modeling for CFD," 2nd ed., DCW Industries, 1998.
3. L. Ignat, D. Pelletier, and F. Ilinca, "A universal formulation of two-equation models for adaptive computation of turbulent flows," *Computer methods in applied mechanics and engineering*, vol. 189, pp. 1119–1139, 2000.

# Non-Isothermal Flow

The Heat Transfer Module includes predefined Multiphysics couplings that combine the heat equation with either a laminar or a turbulent weakly compressible flow. The advantage of using the Multiphysics entry points—over loading the application modes separately—is that you will find predefined couplings in both directions. In particular, the application modes use the same definition of density, which can therefore be a function of both pressure and temperature. Solving this coupled system of equation usually requires numerical stabilization, which is also set up by the predefined multiphysics coupling.

## *Laminar Non-Isothermal Flow*

The Non-Isothermal Flow predefined multiphysics coupling is a combination of two other application modes: Weakly Compressible Navier-Stokes (see page 223) and General Heat Transfer (see page 24). It generates two predefined subdomain group settings: **default** and **Solid domain**. This is illustrated in Figure 7-16, which shows the **Subdomain Settings** dialog box for the General Heat Transfer application mode. The **Group** list is in the lower-left corner.



*Figure 7-16: The Subdomain Settings dialog box for the General Heat Transfer application mode loaded via the Non-Isothermal Flow predefined multiphysics coupling. The velocities are predefined to be consistent with the selected group.*

Assign any fluid domain to the **default** group. The velocities are then taken from the Weakly Compressible Navier-Stokes application mode. Use **Solid domain** to model solid materials. This deactivates convective heat transfer and stabilization on the subdomain.

The groups **default** and **Solid domain** are available in the **Subdomain Settings** dialog box for the Weakly Compressible Navier-Stokes application mode as well. The effect is that the Navier-Stokes equations are active if the **default** group is selected while they are inactive (that is, velocities are not solved for) if the **Solid domain** group is selected.

Note that you have to assign each subdomain to the proper group in both the Weakly Compressible Navier-Stokes and the General Heat Transfer application mode.

The Non-Isothermal Flow predefined multiphysics coupling allows the same boundary conditions as the separate application modes. On boundaries between fluid and solid domains, the temperature is always continuous, which is consistent with the default no-slip condition on the fluid velocity.

### *Turbulent Non-Isothermal Flow*

---

Two variants of the Turbulent Non-Isothermal Flow predefined multiphysics coupling are available. The first one uses the  $k$ - $\epsilon$  turbulence model and the second one uses the  $k$ - $\omega$  turbulence model.

Both predefined multiphysics couplings combine three application modes: one Weakly Compressible Navier-Stokes and two General Heat Transfer application modes. The difference compared to loading these application modes separately is that suitable solver parameters will be set and that you will find two predefined subdomain groups: **default** and **Solid domain**.

You must assign each subdomain to the proper group in each of the application modes. Selecting **Solid domain** for a subdomain has the following effect in each application mode:

- General Heat Transfer application mode 1 is deactivated.
- Convection is disabled in General Heat Transfer application mode 2, giving a pure conduction equation.
- The Turbulence Model application mode is deactivated.

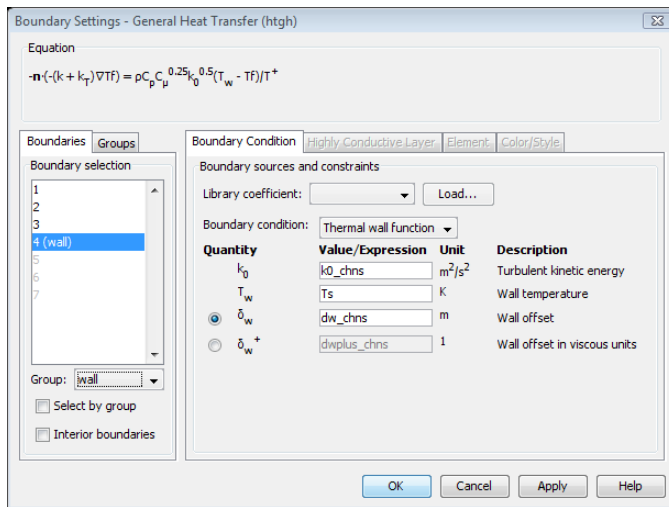
Selecting **default** has the following effects:

- In General Heat Transfer application mode 1, convection is enabled and the velocities taken from the Turbulence Model application mode.

- General Heat Transfer application mode 2 is deactivated.
- The laminar and turbulent kinematic viscosities are taken from the Turbulence Model.
- The density in the Turbulence Model application mode is taken from the active General Heat Transfer application mode.

The last point is to ensure that the same fluid model applies to both the active application modes, for example if you have selected **Ideal gas** from the **Fluid type** list in the General Heat Transfer application mode.

There is also a boundary group, **wall**, available in the **Boundary Settings** dialog boxes of the General Heat Transfer application modes. This group setting for General Heat Transfer application mode 2 appears in Figure 7-17.



*Figure 7-17: The Boundary Settings dialog box for General Heat application mode 2 loaded via a turbulent Non-Isothermal Flow predefined multiphysics coupling. The Wall group helps you to set up the thermal wall function boundary condition.*

The **wall** settings apply to boundaries that separate a fluid domain from a solid domain. They implement a thermal wall function corresponding to the Logarithmic wall function for turbulent flow explained on page 212. All necessary information is automatically taken from the Weakly Compressible Navier-Stokes application mode.

# Building and Solving a Non-Isothermal Laminar Flow Model

## *Introduction*

---

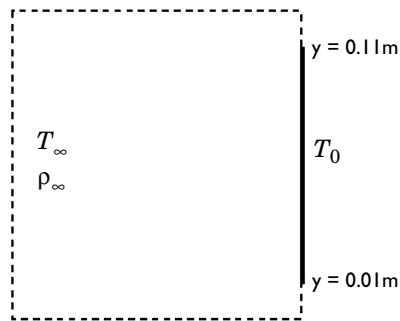
This model shows the application of COMSOL Multiphysics in modeling the non-isothermal flow of gases at relatively low flow velocities. It assumes that the work the gas does is negligible; that the variations in temperature arise through external heating; and that the gas is ideal. These assumptions are excellent at small flow velocities and low pressure, for example, near atmospheric pressure.

The model analyses the free-convection process outside a heated isothermal vertical plate. It illustrates how to couple the heat transfer equation to the equations for non-isothermal flow. It also compares the results to the empirical expressions for the heat flux from an isothermal vertical plate as presented by A. Bejan (Ref. 1).

## *Model Definition—Heated Plate*

---

A vertical plate is held at a constant temperature, which is higher than that of the surrounding air. When the warm plate heats the air near its surface, the air starts rising upward due to its change in density. This is called a *free convection* or *natural convection* process. When modeling this process, consider a rectangular subdomain that consists of air. The 10-cm plate is located on the right vertical wall. Figure 7-18 shows the model geometry.



*Figure 7-18: The model geometry consists of a vertical plate and the air outside the plate surface.*

The steady-state Navier-Stokes equations for non-isothermal flow describe the air flow, while the heat equation for conductive and convective heat transfer describes the heat balance equation. Assuming that the dilatational viscosity of the fluid is zero, the model equations are formulated as follows:

$$\begin{aligned}\rho \mathbf{u} \cdot \nabla \mathbf{u} &= \nabla \cdot [-p \mathbf{I} + \eta (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) - (2\eta/3)(\nabla \cdot \mathbf{u}) \mathbf{I}] + \mathbf{F} \\ \nabla \cdot (\rho \mathbf{u}) &= 0 \\ \rho C_p \mathbf{u} \cdot \nabla T - \nabla \cdot (k \nabla T) &= 0\end{aligned}$$

In the above equations,  $\eta$  denotes the dynamic viscosity of air (Pa·s),  $\mathbf{u}$  is the velocity vector (m/s),  $\rho$  refers to the density ( $\text{kg}/\text{m}^3$ ), and  $p$  is the pressure (Pa). Furthermore,  $k$  denotes the thermal conductivity ( $\text{W}/(\text{m}\cdot\text{K})$ ),  $C_p$  represents the heat capacity ( $\text{J}/(\text{kg}\cdot\text{K})$ ), and  $\mathbf{F}$  is the volume force ( $\text{N}/\text{m}^3$ ).

Now set the volume force to

$$\mathbf{F} = (\rho_\infty - \rho) \mathbf{g}.$$

This equation expresses the lift force in terms of the local density change. The term  $\rho_\infty$  is the density far away from the plate where the warm plate has no influence on the air, and  $\mathbf{g}$  is the gravitational acceleration vector.

The equations also include the expression for density, which you obtain using the Boussinesq approximation:

$$\rho = \rho_\infty \left( 1 - \frac{T - T_\infty}{T_\infty} \right)$$

where  $T_\infty$  is the temperature far away from the plate. Using the Boussinesq approximation to establish the relationship between temperature and density is often more stable than using the ideal gas equation because the Boussinesq approximation does not include the pressure. It thus assumes that the pressure is constant, which is a very good approximation in this model. The variation of  $C_p$  in the temperature range is less than one percent; it is therefore a reasonable simplification to assume it to be constant. The thermal conductivity and the viscosity of air vary somewhat more, but this model keeps them constant in order to capture the effect of expansion only.

For the heat equation, the model keeps the heated plate at a constant temperature,

$$T = T_0.$$

The short boundaries below and above the vertical plate are thermally insulated so that



$$-\mathbf{n} \cdot (-k\nabla T) = 0.$$

Ideally you would not include the insulated plates, but they are needed to create a smooth flow near the heated plate's edges. On the bottom and left boundary, the model sets the temperature equal to the far-field temperature

$$T = T_{\infty}.$$

Air rises upward through the upper horizontal boundary. Therefore assume that convection dominates the transport of heat,

$$\mathbf{n} \cdot (-k\nabla T) = 0.$$

For the non-isothermal flow, assume that the velocity is zero at the wall (including its leading and trailing plates), which corresponds to the no-slip boundary condition

$$\mathbf{u} = \mathbf{0}.$$

The remaining boundaries all have the open boundary condition, meaning that no forces act on the fluid:

$$(-p\mathbf{I} + \eta(\nabla\mathbf{u} + (\nabla\mathbf{u})^T) - (2\eta/3)(\nabla \cdot \mathbf{u})\mathbf{I})\mathbf{n} = \mathbf{0}$$

The open boundary condition in this case defines that you assume that the computational domain extends to infinity.

### *Results and Discussion*

---

The model computes results for ten plate temperatures ranging from 10 °C to 100 °C above the far-field temperature. The far-field conditions hold for dry-air properties at room temperature, 293 K. The results from the simulation show a clear influence on the temperature field near the plate. It also shows the velocity field resulting from free convection due to variations in density. Figure 7-19 and Figure 7-20 display the temperature and velocity fields in the air for a plate temperature of 323 K, which is 30 °C above the far-field temperature. The temperature variations are fairly large near the plate, and that situation of course has a direct impact on the flow profile near the

plate. The far-field temperature remains at room temperature, and the far-field velocities are also fairly small.

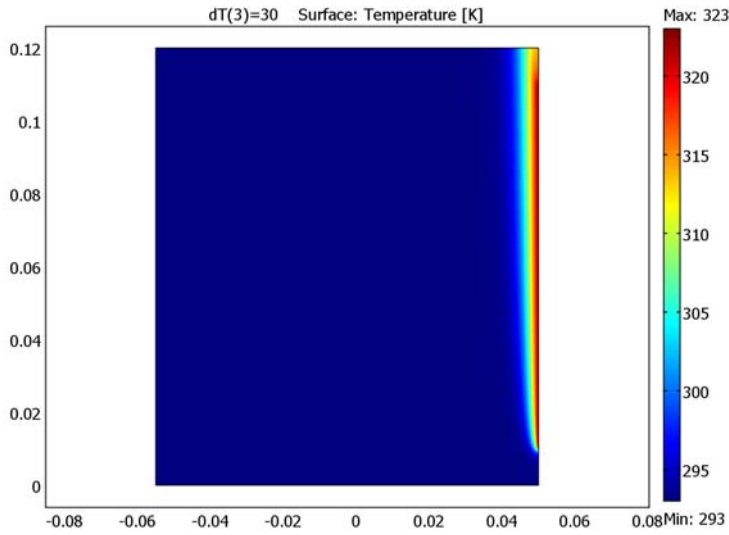


Figure 7-19: Temperature field for a plate at  $30^{\circ}\text{C}$  above the far-field temperature.

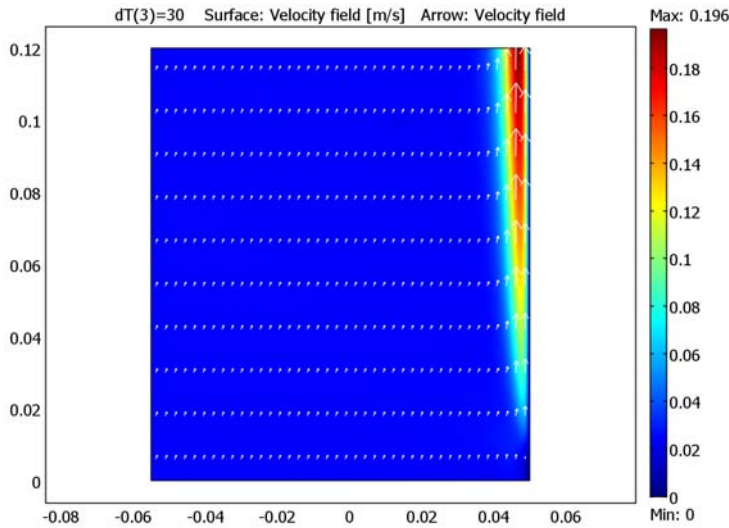


Figure 7-20: Velocity field for a plate at  $30^{\circ}\text{C}$  above the far-field temperature.

Now examine the boundary region in detail. The following two figures show cross-section plots of the temperature and velocity profile, respectively, at  $y = 0.09$  m.

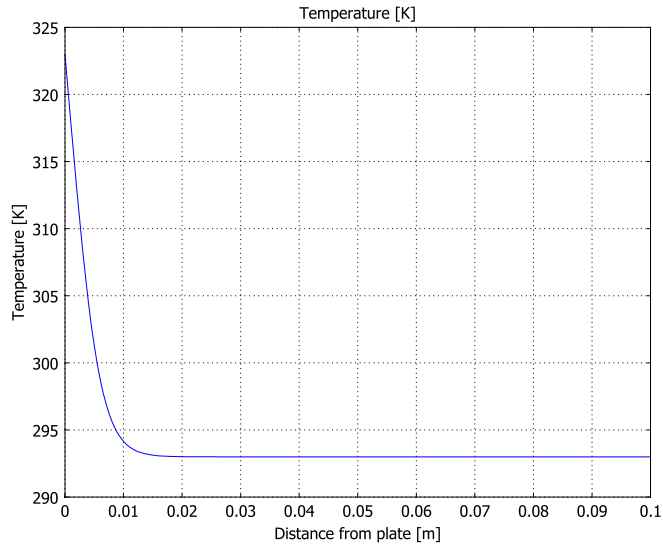


Figure 7-21: Temperature profile for a plate temperature  $30^{\circ}\text{C}$  above the far-field temperature.

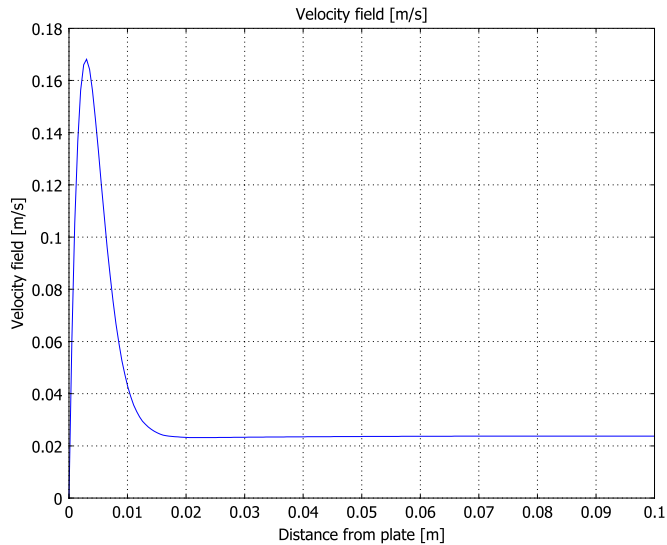


Figure 7-22: Velocity profile for a plate temperature  $30^{\circ}\text{C}$  above the far-field temperature.

Within the boundary region, the temperature starts at the plate temperature and then decreases smoothly to the far-field temperature. The velocity, on the other hand, is zero at the plate boundary (because of the no-slip condition); it then increases with distance from the plate (because of the lift force on the fluid due to density variations). The velocity then reaches a maximum and starts decreasing again. Ideally, the far-field velocity should be zero.

Figure 7-23 illustrates the resulting heat flux from the isothermal plate into the fluid. The heat flux from the plate is very high at the leading edge, and it then decreases along the plate. This is expected because the further up along the plate the air rises, the warmer the air is. Next compare the resulting heat flux that COMSOL Multiphysics calculates compared to the empirical expressions available for free convection from an isothermal plate.

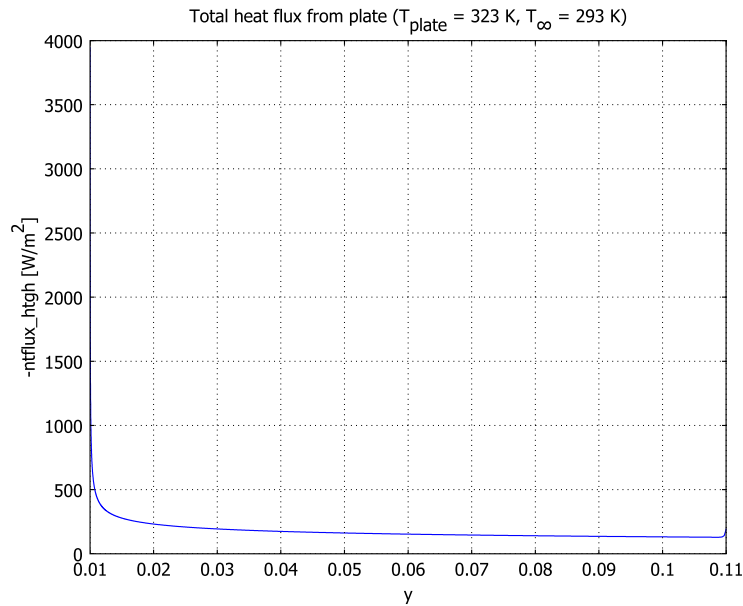


Figure 7-23: The local heat flux from the isothermal plate into the fluid when the plate is 30 °C above the far-field temperature. The horizontal axis goes from the leading edge to the trailing edge.

### Comparing Results with Empirical Heat Transfer Coefficients

This model works with a simple geometry that makes it possible to compare the results with empirical expressions for the heat flux from the isothermal plate. This discussion

uses empirical expressions from Ref. 1 (Section 7.2.3, “Isothermal Wall (Laminar Flow),” to compute the local heat flux along a flat vertical plate as shown in Figure 7-24.

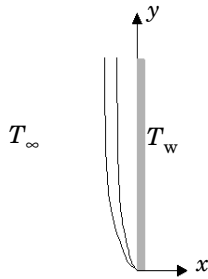


Figure 7-24: Boundary layer for temperature and velocity.

The local Nusselt number for an isothermal wall (laminar flow) is

$$\text{Nu}_y = 0.503 \left[ \frac{\text{Pr}}{\text{Pr} + 0.986\text{Pr}^{0.5} + 0.492} \right]^{0.25} \text{Ra}_y^{0.25} \quad (7-53)$$

where  $\text{Pr}$  is the Prandtl number and  $\text{Ra}$  is the Rayleigh number. These are defined as

$$\text{Pr} = \frac{\eta C_p}{k} \quad (7-54)$$

$$\text{Ra}_y = \frac{\rho^2 g \beta C_p}{\eta k} (T_w - T_\infty) y^3. \quad (7-55)$$

The variables and constants in these expressions are:

- $\eta$  is the viscosity
- $C_p$  is the specific heat
- $k$  is the thermal conductivity
- $\beta$  is the coefficient of volumetric thermal expansion
- $T_w$  is the wall temperature
- $T_\infty$  is the far-field temperature
- $g$  is the gravitational acceleration
- $y$  is the coordinate along the plate

Inserting Equation 7-55 into Equation 7-53 gives the following expression for the local Nusselt number:

$$\text{Nu}_y = 0.503 \left[ \frac{\text{Pr}}{\text{Pr} + 0.986\text{Pr}^{0.5} + 0.492} \right]^{0.25} \left[ \frac{g\beta}{\alpha\nu} (T_w - T_\infty) y^3 \right]^{0.25}. \quad (7-56)$$

Further, evaluate all properties of air at the mean temperature  $(T_w + T_\infty)/2$ . Appendix D of Ref. 1 lists  $\text{Pr}$  and the entire factor  $g\beta/(\alpha\nu)$  for dry air at different temperatures, which makes the evaluation of these properties less cumbersome.

The local heat transfer coefficient is obtained from

$$h_y = \frac{k\text{Nu}_y}{y}. \quad (7-57)$$

The heat flux is:

$$q_y = h_y(T_w - T_\infty). \quad (7-58)$$

Inserting Equation 7-57 into Equation 7-58 gives

$$q_y = \frac{k\text{Nu}_y}{y}(T_w - T_\infty). \quad (7-59)$$

Inserting Equation 7-56 into Equation 7-59 now gives the following expression for the heat flux from the plate into the air:

$$\begin{aligned} q_y &= 0.503 \frac{k}{y} \left( \frac{\text{Pr}}{\text{Pr} + 0.986\text{Pr}^{0.5} + 0.492} \right)^{0.25} \left( \frac{g\beta}{\alpha\nu} y^3 \right)^{0.25} (T_w - T_\infty)^{1.25} \\ &= 0.503 \frac{k}{y^{0.25}} \left( \frac{\text{Pr}}{\text{Pr} + 0.986\text{Pr}^{0.5} + 0.492} \right)^{0.25} \left( \frac{g\beta}{\alpha\nu} \right)^{0.25} (T_w - T_\infty)^{1.25} \end{aligned} \quad (7-60)$$

The following discussion shows the detailed computations for the case  $T_w = T_\infty + 30 \text{ K} = 323 \text{ K}$ . First evaluate the properties of air at the mean temperature so that

$$\frac{T_w + T_\infty}{2} = \frac{323 \text{ K} + 293 \text{ K}}{2} = 308 \text{ K}.$$

At this temperature, the properties of dry air are:

- $\text{Pr} = 0.7168$

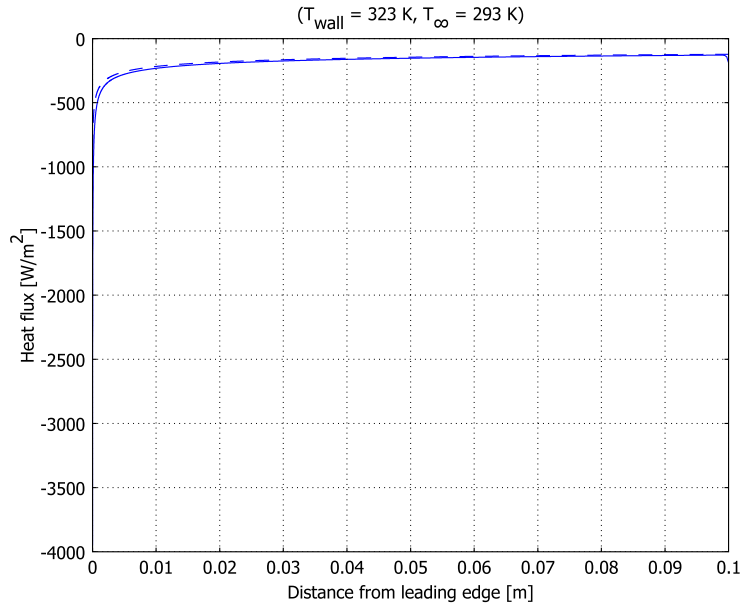
- $\frac{g\beta}{\alpha\nu} = 8.53 \cdot 10^7 \text{ (m}^3 \cdot \text{K)}^{-1}$
- $k = 0.0263 \text{ W/(m} \cdot \text{K)}$ .

Inserting these material properties into Equation 7-60 gives the following expression for the heat flux, expressed as a function of the coordinate  $y$ :

$$q_y = 0.503 \frac{0.0263}{y^{0.25}} \left( \frac{0.7168}{0.7168 + 0.986 \cdot 0.7168^{0.5} + 0.492} \right)^{0.25} (8.53 \cdot 10^7)^{0.25} (323 - 293)^{1.25}$$

$$= 68.7y^{-0.25}$$

The following graph plots the local heat flux,  $q_y$ , together with the local heat flux obtained in the model for the plate temperature of 323 K.



*Figure 7-25: Comparison of COMSOL Multiphysics results (solid line) and the empirical expression (dashed line) for the heat flux from a vertical plate. The plate is 30 degrees hotter than the far-field temperature.*

The results in Figure 7-25 show very good agreement between the heat flux obtained with COMSOL Multiphysics and that computed with the empirical expression.

The next figure shows the same comparison between the results of COMSOL Multiphysics and empirical results for five temperatures ranging from 20 to 100 °C hotter than the far-field temperature.

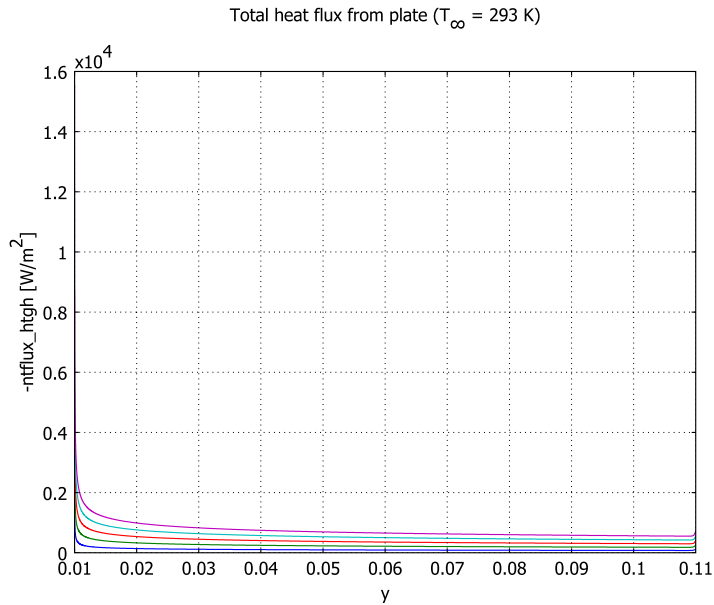


Figure 7-26: Comparison of COMSOL Multiphysics results and empirical expression for the heat flux from a vertical plate for five different temperatures.

As expected, the heat flux into the fluid increases with wall temperature. The agreement between the results of COMSOL Multiphysics and the empirical computation is good, even for wall temperatures as high as 100 °C above the far-field temperature.

### Reference

- 
1. A. Bejan, *Heat Transfer*, Wiley, 1993.
- 

**Model Library path:** Heat\_Transfer\_Module/Tutorial\_Models/heated\_plate

---



### MODEL NAVIGATOR

- 1 Start COMSOL Multiphysics to bring up the **Model Navigator**.
- 2 Select **2D** from the **Space dimension** list.
- 3 From the list of application modes select **Heat Transfer Module>Fluid-Thermal Interaction>Non-Isothermal Flow>Steady-state analysis**.
- 4 Click **OK**.

### OPTIONS AND SETTINGS

Continue by creating a temporary database for constants the model uses.

- 1 From the **Options** menu, choose **Constants**.
- 2 Define the following constants; when finished, click **OK**.

NAME	EXPRESSION
Tinf	293[K]
dT	10[K]
rho0	1.21 [kg/m <sup>3</sup> ]
eta	1.81e-5[Pa*s]
k	27e-3[W/(m*K)]
cp	1006[J/(kg*K)]
gamma	1.4
p0	0[Pa]
g	9.81[m/s <sup>2</sup> ]

### GEOMETRY MODELING

- 1 Shift-click the **Rectangle/Square** button on the Draw toolbar on the left side of the user interface.
- 2 In the dialog box that appears enter the following rectangle properties; when finished, click **OK**.

OBJECT DIMENSIONS	EXPRESSION
Width	0.105
Height	0.12
Base	Corner

OBJECT DIMENSIONS	EXPRESSION
x-position	-0.055
y-position	0

- 3 Click the **Zoom Extents** button on the Main toolbar.
- 4 Shift-click the **Point** button on the Draw toolbar.
- 5 In the dialog box that appears enter the following point properties; when finished, click **OK**.

OBJECT DIMENSIONS	EXPRESSION
x	0.05
y	0.01

- 6 In the same manner, create the a second point with the following point properties; when finished, click **OK**.

OBJECT DIMENSIONS	EXPRESSION
x	0.05
y	0.11

### PHYSICS SETTINGS

- 1 Go to the **Options** menu, find **Expressions**, and then open **Subdomain Expressions**.
- 2 Select Subdomain 1, then enter the following expression; when finished, click **OK**.

NAME	EXPRESSION
rho	$\text{rho0} * (1 - (T - T_{\text{inf}}) / T_{\text{inf}})$

#### *Subdomain Settings—Weakly Compressible Navier-Stokes*

- 1 From the **Multiphysics** menu, select **Weakly Compressible Navier-Stokes (chns)**.
- 2 From the **Physics** menu, select **Subdomain Settings**.
- 3 Select Subdomain 1, then enter the following properties.

SETTINGS	SUBDOMAIN I
$\eta$	eta
$F_x$	0
$F_y$	$(\text{rho0} - \text{rho}) * g$

- 4 Click **OK**.

*Boundary Conditions—Weakly Compressible Navier-Stokes*

- 1 From the **Physics** menu, select **Boundary Settings**.
- 2 Enter the data given in the following table; when done, click **OK**.

SETTINGS	BOUNDARIES 1-3	BOUNDARIES 4-6
Boundary type	Open boundary	Wall
Boundary condition	Normal stress	No slip
$f_0$	0	

*Subdomain Settings—General Heat Transfer*

- 1 From the **Multiphysics** menu, select **General Heat Transfer (htgh)**.
- 2 From the **Physics** menu, select **Subdomain Settings**.
- 3 On the **General** page, enter the following properties (for properties not listed keep the default settings):

SETTINGS	SUBDOMAIN 1
k (isotropic)	k
$\rho$	rho
$C_p$	cp

- 4 Click the **Convection** tab. From the **Matter state** list, select **Gas/liquid**.
- 5 In the  $\gamma$  edit field, type gamma.
- 6 Click the **Init** tab.
- 7 Select Subdomain 1 and type  $T_{inf}$  in the  **$T(t_0)$**  edit field.
- 8 Click **OK**.

*Boundary Conditions—General Heat Transfer*

- 1 From the **Physics** menu, select **Boundary Settings**.
- 2 Enter boundary settings as in the following table; when done, click **OK**.

SETTINGS	BOUNDARIES 1, 2	BOUNDARY 3	BOUNDARIES 4, 6	BOUNDARY 5
Type	Temperature	Convective flux	Insulation/symmetry	Temperature
$T_0$	$T_{inf}$			$T_{inf}+dT$

## MESH GENERATION

The following steps describe how to generate a mesh that properly resolves the velocity field near the wall without using an overly dense mesh in the far field.

- 1 From the **Mesh** menu select **Free Mesh Parameters**.
- 2 From the **Predefined mesh sizes** list, select **Fine**.
- 3 Click the **Boundary** tab, then select Boundaries 4–6.
- 4 In the **Maximum element size** edit field, type  $3e-4$ .
- 5 Click the **Point** tab, then select Point 4.
- 6 In the **Maximum element size** edit field, type  $2e-5$ .
- 7 Click **Remesh**, then click **OK**.

### COMPUTING THE SOLUTION

To easily find an initial guess for the solution, start by solving the problem for a higher viscosity than the true value for air. Then decrease the viscosity until you reach the true value for air. Make the transition from the start value to the true value using the parametric solver in the following way:

- 1 From the **Solve** menu, select **Solver Parameters**.
- 2 In the dialog box that appears, go to the **Solver** list and select **Parametric**.
- 3 On the **General** page, type  $dT$  in the **Parameter name** edit field.
- 4 In the **Parameter values** edit field, type  $10:10:100$ .
- 5 Click **OK**.
- 6 Click the **Solve** button on the Main toolbar.

### POSTPROCESSING AND VISUALIZATION

The default plot shows the velocity field for  $T_{\text{wall}} = 393$  K.

To create Figure 7-20 that shows the velocity field for  $T_{\text{wall}} = 323$  K, follow these steps:

- 1 Click the **Plot Parameters** button on the Main toolbar.
- 2 On the **General** page, find the **Solution to use** area and from the **Parameter value** edit field select **30**.
- 3 Go to the **Arrows** page. Select the **Arrow plot** check box.
- 4 From the **Predefined expressions** list, select **Weakly Compressible Navier-Stokes (chns)>Velocity field**.
- 5 In the **Arrow positioning** area type 40 in the edit field for the number of **x points** and 10 in the edit field for the number of **y points**.

- 6 In the **Arrow parameters** area, clear the **Auto** check box for the **Scale factor**, then type 2 in the associated edit field.
- 7 Click the **Color** button. In the **Arrow Color** dialog box select white, then click **OK**.
- 8 Click **Apply**.

To create Figure 7-19, which displays the temperature field for  $T_{\text{wall}} = 323 \text{ K}$ , continue with the following steps:

- 1 Clear the **Arrow plot** check box.
- 2 Go to the **Surface** page. In the **Predefined quantities** list select list select **General Heat Transfer (htgh)>Temperature**.
- 3 Click **OK**.

The following steps describe how to create Figure 7-21 and Figure 7-22, which show the temperature and velocity profiles, respectively:

- 1 From the **Postprocessing** menu select **Cross-Section Plot Parameters**.
- 2 On the **General** page, from the **Solution to use** list select **30**.
- 3 Click the **Title/Axis** button. Select the option button next to the **First axis label** edit field, then enter the label **Distance from plate [m]**.
- 4 On the **Line/Extrusion** page, in the **Predefined quantities** list select **General Heat Transfer (htgh)>Temperature**.
- 5 In the **Cross-section line data** area enter the following properties:

FIELD	VALUE
x0	0.05
x1	-0.05
y0	0.09
y1	0.09

- 6 Click **Apply**.
- 7 To generate the velocity-profile plot, from the **Predefined expressions** list select **Weakly Compressible Navier-Stokes (chns)>Velocity field**, then click **OK**.

The following steps describe how to create Figure 7-23, which shows the heat flux from the heated plate:

- 1 From the **Postprocessing** menu, select **Domain Plot Parameters**.
- 2 On the **General** page, select **30** from the **Solutions to use** list.

- 3 Click the **Title/Axis** button. Click the option button next to the **Title** edit field, then enter the title Total heat flux from plate ( $T_{\text{wall}} = 323 \text{ K}$ ,  $T_{\infty} = 293 \text{ K}$ ).
- 4 Go to the **Line/Extrusion** page. Select Boundary 5.
- 5 In the **Expression** edit field type `-ntflux_htgh`. This is the negative of the normal total heat flux *out of* the modeling domain and thus equals the total heat flux from the plate *into* the modeling domain.
- 6 Click **OK**.

The following steps describe how to create Figure 7-25.

- 1 From the **Postprocessing** menu, select **Domain Plot Parameters**.
- 2 On the **General** page, from the **Solutions to use** list select **30**.
- 3 From the **Plot in** list, select **New figure**.
- 4 Click the **Title/Axis** button. Click the option button next to the **Title** edit field, then enter the title Total heat flux from plate ( $T_{\text{wall}} = 323 \text{ K}$ ,  $T_{\infty} = 293 \text{ K}$ ).
- 5 Click the option button next to the **First axis label** field, then enter Distance from leading edge [m].
- 6 Click the option button next to the **Second axis label** field, then enter Heat flux [ $\text{W/m}^2$ ].
- 7 Click **OK**.
- 8 Click the **Line/Extrusion** page, then select Boundary 5.
- 9 In the **Expression** edit field, type `ntflux_htgh` and click **Apply**.
- 10 On the **General** page, select the **Keep current plot** check-box.
- 11 Click the **Line/Extrusion** tab.
- 12 In the **Expression** edit field, type `-68.7*(y-0.01)^-0.25`.
- 13 Click the **Line Settings** button.
- 14 In the **Line style** list, select **Dashed line**. Click **OK**.
- 15 Click **OK**.

The following steps describe how to create Figure 7-26.

- 1 From the **Postprocessing** menu, select **Domain Plot Parameters**.
- 2 On the **General** page, go to the **Solutions to use** list and select **20, 40, 60, 80, and 100**.
- 3 In the **Plot in** list select **New figure**.

- 4 Click the **Title/Axis** button.
- 5 Click the option button next to the **Title** edit field, then enter the title: Total heat flux from plate ( $T_{\infty} = 293 \text{ K}$ ).
- 6 Click **OK**.
- 7 Click the **Line/Extrusion** tab, then select Boundary 5.
- 8 In the **Expression** edit field, type `-ntflux_htgh`.
- 9 Click **OK**.

# Building and Solving a Non-Isothermal Turbulent Flow Model

## *Introduction*

---

The present model investigates the performance of a displacement ventilation system. In the model, you set up the coupled heat transfer and turbulent flow equations describing the system using the predefined multiphysics coupling Turbulent Non-Isothermal Flow,  $k$ - $\epsilon$ . Given measured values for inlet velocity, inlet temperature, and heat flux, this model yields field configurations of air temperature and velocity that are consistent with experimental measurements and analytic global models (Ref. 1).

## *Model Definition*

---

In general, there are two classes of ventilation: mixing ventilation and displacement ventilation. In displacement ventilation, air enters a room at the floor level and displaces warmer air to achieve the desired temperature. Heating sources in the room can include running electronic devices, or inlet jets of warm air. A potential issue with the displacement ventilation approach is that significant temperature variation and strong stratification may arise.

The model geometry consists of a test chamber with the dimensions 2.5 m by 3.65 m by 3 m. A warm jet enters the chamber from an inlet located at the floor center. Fresh air, at constant temperature and relatively low velocity, is supplied through a wall inlet. Heat exits the chamber through an exhaust located in the center of the ceiling. The walls of the chamber are assumed to be thermally insulated.



Symmetry reduces the modeling domain to half of the chamber, as depicted in Figure 7-27. The warm jet feeds  $0.028 \text{ m}^3/\text{s}$  of air at a temperature of  $45 \text{ }^\circ\text{C}$  into the room. The temperature of the fresh air is  $21 \text{ }^\circ\text{C}$  and has a flow rate of  $0.05 \text{ m}^3/\text{s}$ .

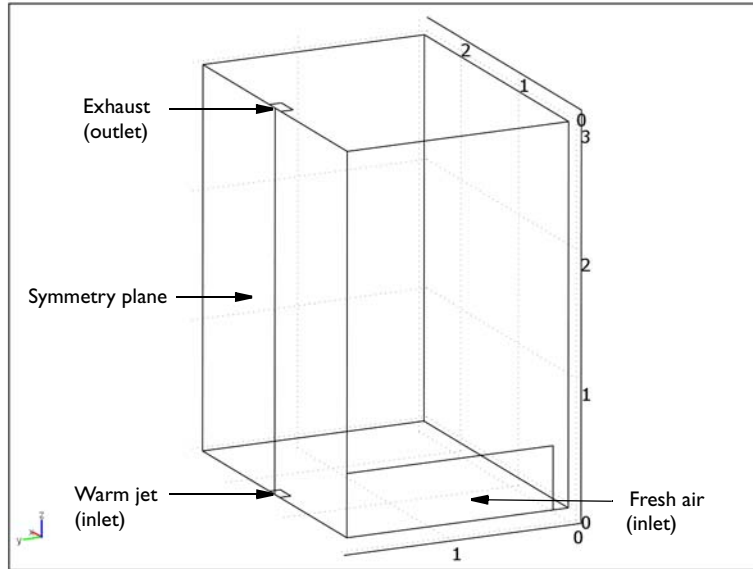


Figure 7-27: The modeling domain is reduced to half the chamber size due to symmetry.

Convection of heat can be either forced or free. Forced convection occurs if

$$\frac{g\beta\Delta T}{U^2/L} \ll 1 \quad (7-61)$$

where  $g$  is the gravity ( $\text{m}/\text{s}^2$ ),  $\beta$  ( $1/\text{K}$ ) is the coefficient of thermal expansion,  $T$  (K) the temperature,  $U$  ( $\text{m}/\text{s}$ ) the velocity, and  $L$  (m) refers to the characteristic length. Equation 7-61 states that the buoyancy force is small compared to the inertial force. In such a situation, the character of the flow field is described by the Reynolds number,  $\text{Re} = UL/\nu$ . Natural convection occurs if Equation 7-61 is not fulfilled, in which case the flow field character is described by the Grashof number,

$$\text{Gr} = \frac{g\beta\Delta TL^3}{\nu^2}$$

where  $\nu$  ( $\text{m}^2/\text{s}$ ) is the kinematic viscosity. If the convective forces and buoyant forces are of the same order of magnitude, then  $\text{Gr}^{1/2}$  can be interpreted as the ratio between the inertial and viscous forces. That is, when the Grashof number is large, the flow becomes turbulent.

To investigate if Equation 7-61 holds, the air can be approximated as an ideal gas in which case  $\beta = 1/T$ . Furthermore,  $\Delta T \approx 20$  K,  $U \approx 1$  m/s, and  $L \approx 2$  m. This gives

$$\frac{g\beta\Delta T}{U^2/L} \approx \frac{9.8 \cdot 20 \cdot 2}{1^2 \cdot 300} = 1.3$$

Hence, it is the Grashof number that determines whether the flow is turbulent or laminar. Using the same approximations as above:

$$\text{Gr} \approx \frac{9.8 \cdot 20 \cdot 2^3}{(1.6 \cdot 10^{-5})^2 \cdot 300} = 2 \cdot 10^{10} \quad (7-62)$$

Equation 7-62 clearly indicates that the flow is turbulent.

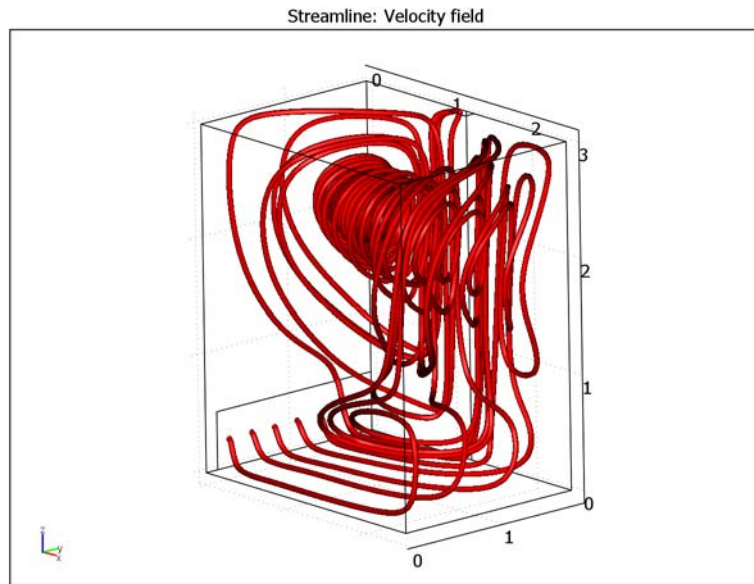
You can model natural turbulent convection with the predefined multiphysics coupling Turbulent Non-Isothermal Flow,  $k$ - $\epsilon$ . This application mode solves the Favre averaged Navier-Stokes equations including the energy equation and Favre averaged transport equations for the turbulent kinetic energy,  $k$ , and the turbulent dissipation rate,  $\epsilon$ .

### *Results and Discussion*

---

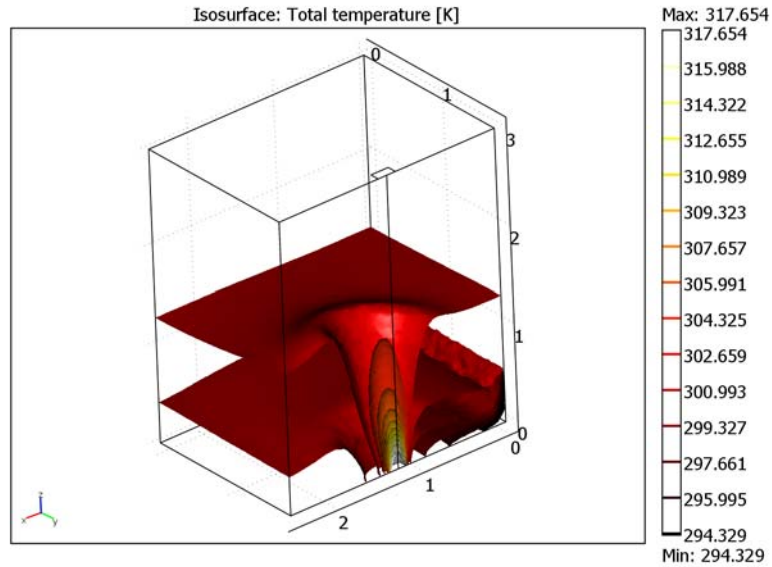
Figure 7-28 shows the streamlines of the velocity field in the test chamber. The cold air that enters through the diffuser sinks towards the floor. As the air approaches the jet it is heated and entrained into the jet. The rising air creates a circulating pattern.

Some of the circulating exits through the vent in the roof while some sinks back to the floor.



*Figure 7-28: Streamlines illustrating the velocity field in the test chamber.*

Figure 7-29 shows isosurfaces of the temperature. The recirculation in the upper half of the room gives a fairly homogeneous temperature for  $z > 1.5$  m.



*Figure 7-29: Isosurfaces of the temperature.*

Figure 7-30 shows a plot of the temperature in a straight line from the center of the inlet jet to the center of the outlet vent. The result is compared to experimental data from Ref. 1. The model captures the general trend and the curves compare very well in the lower part of the room. The temperature of the experimental curve does however fall to 26 °C—four degrees below the computational results. One possible explanation is that the experimental jet drifts more to the side than indicated by the

simulation. Another possibility mentioned in Ref. 1 is that the experimental facility is not so well isolated as intended.

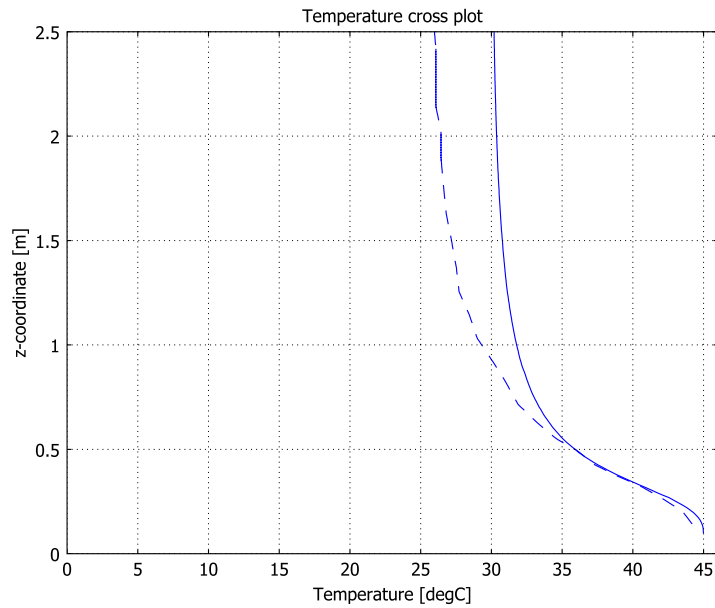


Figure 7-30: Air temperature in a straight line from the center of the inlet jet to the center of the outlet vent. The solid line is the result from the simulation while the dashed line is experimental results from Ref. 1.

### Reference

I. D. Mazoni and P. Guittou, "Validation of Displacement Ventilation Simplified Models," *Proceedings of 'Building Simulation '97'*, the Fifth International IBSPA Conference, vol. I, pp. 233–239, International Building Performance Simulation Association (IBSPA), 1997.

**Model Library path:** Heat\_Transfer\_Module/Tutorial\_Models/  
displacement\_ventilation

### Modeling Using the Graphical User Interface

I Start by clicking the COMSOL Multiphysics icon on the desktop.

- 2 In the **Model Navigator**, set the **Space dimension** to **3D**.
- 3 In the list of application modes, select **Heat Transfer Module>Fluid Thermal Interaction>Turbulent Non-Isothermal Flow, k-ε**.
- 4 From the **Element** list, select **Lagrange - Linear**.
- 5 Click the **Multiphysics** button.
- 6 Click the **Add** button.
- 7 Select the **General Heat Transfer (htgh2)** application mode, then click the **Remove** button.  
 This application mode accounts for heat transfer in solids and can be removed because the computational domain will contain only fluid.
- 8 Click **OK**.

### GEOMETRY MODELING

- 1 Hold down Shift and click the **Block** button on the Draw toolbar.
- 2 In the **Length** area, enter the following values:

LENGTH	VALUE
X	2.5
Y	1.825
Z	3

- 3 Click **OK**.
- 4 From the **Draw** menu, select **Work-Plane Settings**.
- 5 In the **Work-Plane Settings** dialog box, click **OK** to accept the default settings.  
 This creates a 2D *xy*-plane at  $z = 0$  in a new geometry, *Geom2*.
- 6 Hold down Shift and click the **Rectangle/Square** button on the Draw toolbar.
- 7 In the dialog box that appears, enter the following rectangle properties:

PROPERTY	EXPRESSION
Width	0.18
Height	0.09
Base	Corner
x-position	1.16
y-position	1.735

- 8 Click **OK**.

9 Again, hold down Shift and click the **Rectangle/Square** button.

10 Enter the following rectangle properties:

PROPERTY	EXPRESSION
Width	0.2
Height	0.1
Base	Corner
x-position	1.15
y-position	1.725

11 Click **OK**.

12 From the **Draw** menu, choose **Embed**.

13 Select both objects from the list, then click **OK**.

14 Make sure you are on the **Geom1** page, then select the embedded rectangle EMB2 (the larger of the two rectangles).

15 Click the **Move** button on the Draw toolbar. In the **Displacement** area, type 3 in the **z** edit field.

16 Click **OK**.

17 From the **Draw** menu, select **Work-Plane Settings** and choose **y-z plane**.

18 Click **OK**.

This creates a 2D *yz*-plane at  $x = 0$ .

19 Hold down Shift and click the **Rectangle/Square** button on the Draw toolbar.

20 In the dialog box that appears, enter the following rectangle properties:

PROPERTY	EXPRESSION
Width	1.7
Height	0.5
Base	Corner
x-position	0.125
y-position	0

21 Click **OK**.

22 From the **Draw** menu, select **Embed**.

23 Select **R3** from the list, then click **OK**.

24 From the **Draw** menu, select **Line**.

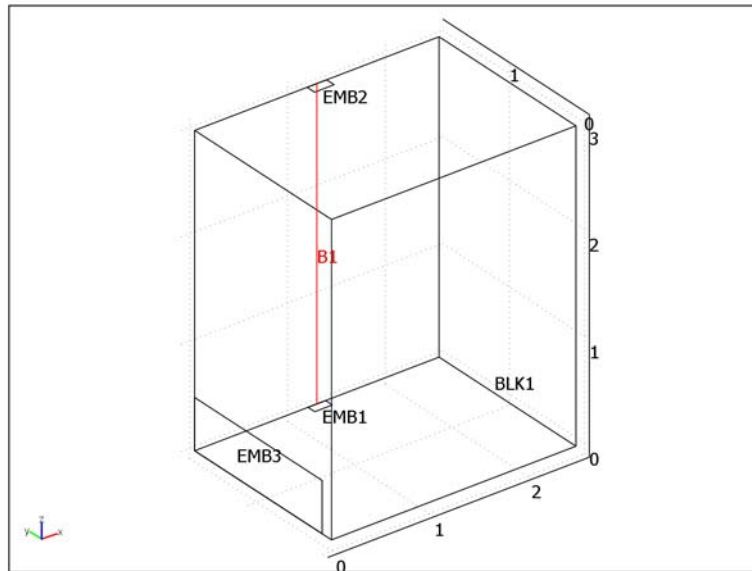
25 In the dialog box that appears, enter the following line coordinates:

DIRECTION	EXPRESSION
x	1.25 1.25
y	1.825 1.825
z	0 3

You will use this line to control the mesh along the direction of the jet.

26 Click the **Zoom Extents** button on the Main toolbar.

The resulting geometry is shown below.



#### OPTIONS AND SETTINGS

- 1 From the **Options** menu, open the **Constants** dialog box.
- 2 Specify the following names, expressions, and descriptions (optional):

NAME	EXPRESSION	DESCRIPTION
Ms	0.028[m <sup>3</sup> /s]	Source flow rate
Md	0.05[m <sup>3</sup> /s]	Diffuser flow rate
As	0.0324[m <sup>2</sup> ]	Source inlet area
Ad	1.7[m <sup>2</sup> ]	Diffuser inlet area



NAME	EXPRESSION	DESCRIPTION
Us	Ms/As	Source flow velocity
Ud	Md/Ad	Diffuser flow velocity
Tdiff	21[degC]	Diffuser air temperature
Tsource	45[degC]	Source air temperature
p0	101325[Pa]	Reference pressure
g	-9.81[m/s^2]	Gravity acceleration constant

3 Click **OK**.

### PHYSICS SETTINGS

#### *Subdomain Settings—k-ε Turbulence Model*

- 1 From the **Physics** menu, select **Subdomain Settings**.
- 2 Select Subdomain 1.
- 3 Click the **Load** button next to the **Library material** list.
- 4 Select **Liquids and Gases>Gases>Air**, then click **OK**.
- 5 Edit the expression for  $\rho$  so that it reads  $\rho(p0[1/Pa], Tf[1/K]) [kg/m^3]$ .
- 6 In the edit field for  $F_z$ , enter  $g*\rho_{chns}$ .
- 7 Click the **Stabilization** tab.
- 8 In the **Tuning parameter** edit field for **Crosswind diffusion**, enter 0.3.
- 9 From the **Artificial diffusion settings for** list, select **Turbulence**.
- 10 Select the **Turbulent isotropic diffusion** check box.

Buoyant flows often feature shear layers and the location of these shear layers are hard to predict in advance. Some extra diffusion is necessary to capture the shear layers and prevent the computation from diverging.

11 Click **OK**.

Because the flow contains a jet impinging on the ceiling, avoiding stagnation point anomalies requires realizability constraints (see “Realizability Constraints” on page 219).

- 1 From the **Physics** menu, select **Properties**.
- 2 Set **Realizability** to **On**.
- 3 Click **OK**.

#### *Boundary Settings—k-ε Turbulence Model*

- 1 From the **Physics** menu, select **Boundary Settings**.

- 2 Select Boundary 5. From the **Boundary type** list, select **Inlet**.
- 3 In the  $U_0$  edit field, type  $U_d$ , in the  $L_T$  edit field, type  $0.01$ , and in the  $I_T$  edit field, type  $0.1$ .
- 4 Select Boundaries 6 and 9. From the **Boundary type** list, select **Symmetry boundary**.
- 5 Select Boundary 7. From the **Boundary type** list, select **Outlet** and then, from the **Boundary condition** list, select **Pressure**.
- 6 Select Boundary 8. From the **Boundary type** list, select **Inlet**.
- 7 In the  $U_0$  edit field, type  $U_s$ , and in the  $L_T$  edit field, type  $0.05$  and in the  $I_T$  edit field, type  $0.1$ .
- 8 Select the remaining boundaries (Boundaries 1, 2, 3, 4, and 10).
- 9 Click the  $\delta_w^+$  button, then enter  $1000$  in the corresponding edit field.
- 10 Click **OK**.

The inlet boundary conditions for the hot jet can be understood by examining the temperature profile in Figure 7-30. The quick drop in temperature indicates that a rather large amount of colder air is quickly entrained. The only way to achieve this is to have a long turbulent length scale and high turbulence intensity.

The inflow from the diffuser is modeled as if there were a grate there, that is, with high turbulence intensity and a short turbulent length scale.

The large offset at the walls implies that you will not obtain maximum accuracy for the wall shear stresses. However, the influence of the wall shear stresses on the solution is small, and decreasing the offset would require much denser mesh on the walls.

#### *Subdomain Settings—General Heat Transfer*

- 1 From the **Multiphysics** menu, select the **General Heat Transfer (htgh)** application mode.
- 2 From the **Physics** menu, select **Subdomain Settings**.
- 3 Select Subdomain 1.
- 4 From the **Library material** list, select **Air**.
- 5 Edit the expression for  $\rho$  so that it reads  $\rho_0(p_0[1/\text{Pa}], T_f[1/\text{K}])$  [kg/m<sup>3</sup>].
- 6 Click the **Init** tab and enter  $T_{diff}$  in the edit field for **Tf( $t_0$ )**.
- 7 Click **OK**.

#### *Boundary Settings—General Heat Transfer*

- 1 From the **Physics** menu, select **Boundary Settings**.

- 2 Select Boundary 5.
- 3 From the **Boundary condition** list, select **Temperature**.
- 4 In the **Temperature** edit field, type  $T_{diff}$ .
- 5 Select Boundary 7.
- 6 From the **Boundary condition** list, select **Convective flux**.
- 7 Select Boundary 8.
- 8 From the **Boundary condition** list, select **Temperature**.
- 9 In the **Temperature** edit field, type  $T_{source}$ .
- 10 Click **OK**.

### MESH GENERATION

You need to generate a mesh that resolves the hot jet and also take care to refine it at the edges of the diffuser and at the outlet.

- 1 From the **Mesh** menu, select **Free Mesh Parameters**.
- 2 Click the **Custom mesh size** button.
- 3 In the **Maximum element size** edit field, type 0.12.
- 4 In the **Element growth rate** edit field, type 1.2.
- 5 Click the **Boundary** tab.
- 6 Select Boundaries 7 and 8.
- 7 In the **Maximum element size** edit field, type 0.02.
- 8 Click the **Edge** tab.
- 9 Select Edges 6, 7, and 8.
- 10 In the **Maximum element size** edit field, type 0.05.
- 11 Select Edge 19, then click the **Distribution** tab.
- 12 Select the **Constrained edge element distribution** check box.
- 13 In the **Number of edge elements** edit field, type 100.
- 14 Select the **Distribution** check box.
- 15 In the **Element ratio** edit field, type 5.
- 16 From the **Distribution** list, select **Exponential**.
- 17 Select the **Symmetric** check box.
- 18 Click the **Remesh** button.
- 19 Click **OK**.

## COMPUTING THE SOLUTION

The temperature variable is in buoyant flows strongly coupled to the rest of the Navier-Stokes equations. It is therefore recommended to solve the temperature in the same group as the velocities and the pressure for these kind of flows.

- 1 Click the **Solver Parameters** button on the Main toolbar.
- 2 In the **Group 1 Components** edit field, add  $T_f$ .
- 3 For **Group 3**, click the **Delete** button.
- 4 For **Group 1**, click the **Linear Solver Settings** button.
- 5 In the dialog box that appears, select **Preconditioner** from the list to the left.
- 6 In the **Number of levels** edit field, type 3.

The iterative solver uses a direct solver to solve a correction problem on the coarsest multigrid hierarchy. This is the reason why you must increase the number of levels for large problems. Manually created multigrid hierarchies often give the best performance, but automatic mesh generation works fine for simple geometries like the current one.

- 7 Click **OK**.
- 8 Back in the **Solver Parameters** dialog box, select the **Manual specification of segregated steps** check box.
- 9 For **Group 1**, click the **Damping/Termination Settings** button.
- 10 In the dialog box that appears, type 0.3 in the **Damping constant** edit field and then click **OK**.
- 11 For **Group 2**, click the **Damping/Termination Settings** button.
- 12 In the dialog box that appears, type 0.3 in the **Damping constant** edit field and then click **OK**.

The complex flow patterns of buoyant flows often makes it necessary to use low damping constants to find a stationary solution.

- 13 Click the **Advanced** tab.
- 14 From the **Type of scaling** list, select **Manual**.
- 15 In the **Manual scaling** edit field, type  $T_f / 100$   $p / 10$ .

The temperature  $T_f$  is approximately 300 K,  $p$  ranges between 0 and 10, while the velocities are on the order of unity. If the variables are not scaled, the resulting equation system can become ill conditioned.

- 16 Click **OK**.

17 Click the **Solve** button on the Main toolbar.

---

**Note:** The solution takes a little more than an hour to compute and the computation requires 1.2 GB of RAM.

---

## POSTPROCESSING AND VISUALIZATION

Figure 7-28 is produced by the following steps:

- 1 From the **Postprocessing** menu, choose **Plot Parameters**.
- 2 On the **General** page, clear the **Slice** check box and select the **Streamline** check box.
- 3 Click the **Streamline** tab.
- 4 From the **Predefined quantities** list, select **k-ε Turbulence Model (chns)>Velocity field**.
- 5 Click the **Specify start point coordinates** option button.
- 6 Enter the following coordinates:

PROPERTY	VALUE
x	zeros(6,1)
y	0.25:0.25:1.5
z	ones(6,1)*0.25

- 7 From the **Line type** list, select **Tube**.
- 8 Click **OK**.
- 9 Click the **Headlight** button in the Camera toolbar.
- 10 Use the mouse to rotate the geometry.

To reproduce Figure 7-29 follow these steps:

- 1 From the **Postprocessing** menu, choose **Plot Parameters**.
- 2 On the **General** page, clear the **Streamline** check box and select the **Isosurface** check box.
- 3 Click the **Isosurface** tab.
- 4 In the **Number of levels** edit field, type 15.
- 5 From the **Colormap** list, select **hot**.
- 6 Click **OK**.
- 7 Use the mouse to rotate the geometry.

To create Figure 7-30 proceed as follows:

- 1 From the **Option** menu, select **Functions**.
- 2 Click the **New** button.
- 3 Select the **Interpolation** button.
- 4 From the **Use data from** list, select **File** and click the **Browse** button.
- 5 Locate and select the file temp\_experiment.txt then click **Open**.
- 6 In the **Function name** edit field, type temp\_exp and click **OK**.
- 7 From the **Interpolation method** list, select **Linear**.
- 8 Click **OK** to close the **Functions** dialog box.
- 9 From the **Solve** menu, select **Update model**.
- 10 From the **Postprocessing** menu, choose **Cross-Section Plot Parameters**.
- 11 Select **Keep current plot**.
- 12 Go to the **Line/Extrusion** page.
- 13 In the **Expressions** edit field, type z.
- 14 Enter the following coordinates in the **Cross-section line data** area:

PROPERTY	VALUE
x0	1.25
y0	1.825
z0	0.1
x1	1.25
y1	1.825
z1	2.5

- 15 Click option button next to the **Expression** button.
- 16 From the **Predefined quantities** list, select **General Heat Transfer (htgh)>Temperature**.
- 17 From the **Unit** list, select **°C**.
- 18 Click **Apply**.
- 19 Return to the **Cross-Section Plot Parameters** dialog box and click the **Expression** button.
- 20 In the **Expression** edit field, type temp\_exp(z) and then click **OK**.
- 21 Click the **Line Settings** button.
- 22 From the **Line style** list, select **Dashed line** and click **OK**.

23 Click **OK**.

24 In the **Figure 1** window, click the **Edit plot** button.

25 Clear the **x limits** check box and in the left edit field that becomes active, type 0.

26 In the **Title** edit field, type Temperature cross plot.

27 In the **x label** edit field, type Temperature [degC].

28 Click **OK**.





# Application Mode Implementation

The Heat Transfer Module is built completely on top of COMSOL Multiphysics. All application modes are based on the same generic PDE and weak form equations defined in COMSOL Multiphysics. This chapter acts as a reference, where you can look up the variables defined and used by each application mode.

# Application Mode Overview

The application modes in the Heat Transfer Module fall into one of the module's two main branches: *heat transfer* and *momentum transport*. There are three separate application modes in the first group:

- The General Heat Transfer application mode offers the most complete heat transfer functionality available in COMSOL Multiphysics, including radiative heat transfer.
- Thin Conductive Shell application mode can only be applied to thin 3D surfaces, where it can considerably reduce model size and computational complexity.
- The Bioheat Equation is tailored for modeling heat transfer in living tissue. It uses Pennes's model to account for blood perfusion and metabolism.

For momentum transport, there is—technically speaking—only one application mode, which behaves rather differently depending on its property settings. Three combinations of property settings and defaults have their own names and entry points in the Model Navigator, and therefore, for all purposes, act as separate application modes:

- The Weakly Compressible Navier-Stokes application mode describes the momentum transport part of a coupled fluid-thermal problem where the fluid density is a function of pressure and temperature.
- The  $k$ - $\epsilon$  Turbulence Model application mode models turbulent flow using the standard  $k$ - $\epsilon$  turbulence model and logarithmic wall functions.
- The  $k$ - $\omega$  Turbulence Model application mode implements Wilcox's  $k$ - $\omega$  model.

Note that you can change one into another using the settings in the **Application Mode Properties** dialog box. In the same way, you can also remove both turbulence and compressibility to reduce the application mode to laminar incompressible Navier-Stokes, or add a swirl velocity component to model swirl flow.

In addition to the pure application modes, the Heat Transfer Module includes three predefined Multiphysics couplings:

- Non-Isothermal Flow combines a General Heat Transfer application mode with a Weakly Compressible Navier-Stokes application mode to describe laminar flow with variable density.

- Turbulent Non-Isothermal Flow in addition to turbulent Weakly Compressible Flow requires two separate General Heat Transfer application modes to model the jump in temperature across the turbulent thermal boundary layer.

For more information about the Fluid-Thermal Interaction predefined multiphysics couplings, see “Non-Isothermal Flow” on page 228.

- Joule Heating applies the resistive heating generated in a Conductive Media DC application mode from COMSOL Multiphysics, or the AC/DC Module if included in your license, as a source in a General Heat Transfer application mode. The reverse coupling, giving a temperature-dependent conductivity, can also be included. The Joule Heating predefined Multiphysics coupling is documented under “Electro-Thermal Interaction—Joule Heating” on page 380 of the *COMSOL Multiphysics Modeling Guide*.

# Application Mode Equations

The underlying PDEs, boundary conditions, and theoretical background for each application mode are described in the chapters “The General Heat Transfer Application Mode” on page 24, “The Thin Conductive Shell Application Mode” on page 154, “The Bioheat Equation Application Mode” on page 172 and “Momentum Transport” on page 179.

All subdomain equations for the application modes in the Heat Transfer Module are based on either the Coefficient form equation or the General form equation. See “What Equations Does COMSOL Multiphysics Solve?” on page 507 in the *COMSOL Multiphysics Reference Guide* for a concise listing of the different forms. The sections “Using the Coefficient Form PDEs” on page 247 and “Using the General Form PDE” on page 257, both in the *COMSOL Multiphysics Modeling Guide*, provide a more thorough introduction including examples.

You can find out which equation form a particular application mode uses by choosing **Model Settings** from the **Physics** menu and looking at the default **Equation system form**. Note that you can always choose to display equations in a more general equation system form than the one used by the application mode. That is, you can always switch from Coefficient form to General form, and from General form to Weak form, but not in the opposite direction. In a Multiphysics model, the application mode implemented in the least general form decides which equation forms are available.

To see how the equations for the application mode you are using are implemented in the selected equation system form, choose **Equation System>Subdomain Settings** from the **Physics** menu. In the dialog box, you will find a number of coefficient pages, where you can inspect how the application mode defines the terms in the generic equation of the selected form, as well as two pages marked **Element** and **Variables**. On the **Element** page, you can examine the shape functions used by the application mode to represent its the dependent variables. The **Variables** page, which is discussed in more detail in the next section, contains a list of variables defined by the application mode.

Under **Physics>Equation System**, you will also find similar dialog boxes for boundary conditions and variables, as well as for edge and point settings. Note that you can always modify the subdomain equations, boundary conditions, and definition of the application mode variables by editing the coefficient expressions and variables in the different **Equation System** dialog boxes.

# Application Mode Variables

The application modes in the Heat Transfer Module define a large set of variables. This chapter contains lists of the variables that each application mode defines. Other information, like the theoretical background for the application modes, can be found elsewhere in the *Heat Transfer Module User's Guide*.

Most application modes define variables of three different types. The *dependent variables* are the unknowns solved for. They are represented using shape functions, and are essential to any problem, that is, they cannot be replaced by numerical values or expressions in terms of other variables. You can specify names for the dependent variables when you select an application mode in the Model Navigator. Below, the default names are listed for each application mode.

The *application scalar variables* are normally numerical values, for example universal constants, model parameters and user-controlled constants such as the frequency of a harmonic oscillation. In rare cases, they can also be globally defined expressions in terms of other application mode variables. In the following sections, any application scalar variables are presented in a table together with their default value, unit and description.

Remaining variables defined by the application mode, collectively referred to as *application mode variables*, are expressions defined in terms of other variables and constants on domains of a specific dimension. Further, the definition can vary from domain to domain, depending on the currently applied physics settings. The application mode variable tables in the following sections are organized as follows:

- The **Name** column lists the names of the variables that you can use in expressions in the equations or for postprocessing. Note that some variables represent vector- or tensor-valued expressions. In that case, there is one variable for each component, using the independent variable names as index. For example, if the table writes the variable name as  $T_{x_i}$ , the actual variables are typically called  $T_x$  and  $T_y$  in 2D Cartesian coordinates and  $T_r$ ,  $T_z$  in axisymmetric geometries.

---

**Note:** All variables defined by the application mode, except the dependent variables, get an underscore plus the application mode name appended to them. For example, the default application mode name for the Incompressible Navier-Stokes application mode is `chns`, hence, the density variable name is `rho_chns`.

---

- The **Type** column indicates if the variable is defined on subdomains (S), boundaries (B), edges (E), or points (P). The column indicates the levels where the variable is explicitly declared. Variables that are not explicit on domains of a given dimension can nevertheless be evaluated if they are declared in an adjacent domain of higher dimension. When there is more than one adjacent domain, an average value is computed. For example, all subdomain variables can also be evaluated on boundaries, but on interior boundaries take on the mean of the values in the two adjacent subdomains.

The **Type** column can also show a (V) or (T) to indicate that the variable is a component of a vector or tensor. Vector variables are included in the lists of predefined expressions for plot types such as Arrow, Deform and Streamline.

- The **Description** column gives a description of the variables.
- The **Expression** column gives the expression of the variables in terms of other variables or physical quantities, or a reference to an explanation elsewhere.

The application mode variables listed in the following sections are all available both for postprocessing and when formulating the equations. Note that you can also find the definitions of these variables directly in the graphical user interface of COMSOL Multiphysics. From the **Physics** menu, choose **Scalar Variables** to inspect the application scalar variables, or select one of the items in the **Equation System** submenu to open the **Equation System** dialog box for the corresponding dimension. On the **Variables** page, you will find the application mode variable names and their definitions (which you can also change, if you like).

Note that the actual variable names in your model may depend on the spatial coordinate names, as well as on the order in which application modes are added to a Multiphysics model. The names and definitions may also be affected by application mode property settings and by the physics settings for the particular subdomain or boundary. Therefore, it is good practice to always check the variable names for your particular model in the GUI, before using them in equations.

## General Heat Transfer

---

### APPLICATION MODE NAME

htgh

---

TABLE 8-1: DEPENDENT VARIABLES, GENERAL HEAT TRANSFER

NAME	DESCRIPTION
T	Temperature
J	Radiosity

---

**Note:** The dependent variable for the radiosity,  $J$ , is always present but is used only if the model includes surface-to-surface radiation.

---

TABLE 8-2: SCALAR VARIABLES, GENERAL HEAT TRANSFER

NAME	VALUE	UNIT	DESCRIPTION
sigma	$5.67 \cdot 10^{-8}$	$W/(m^2 \cdot K^4)$	Stefan-Boltzmann constant
Rg	8.31451	$J/(mol \cdot K)$	Universal gas constant

TABLE 8-3: APPLICATION MODE VARIABLES, GENERAL HEAT TRANSFER

NAME	TYPE	DESCRIPTION	EXPRESSION
rho	S	Density	$\rho$
C	S	Heat capacity	$C_p$
k	S	Thermal conductivity	$k$
$kxixj$	S	Thermal conductivity tensor, $x_i x_j$ component	$k_{ij}$
Q	S	Heat source	$Q$
qs	S	Production/absorption coefficient	$q_s$
gamma	S	Ratio of specific heats	$\gamma$
u, v, w	S	$x_1$ -, $x_2$ -, $x_3$ -velocity	$u, v, w$
p	S	Pressure	$p$
eta	S	Dynamic viscosity	$\eta$
hz	S	Out-of-plane heat transfer coefficient (ID)	$h_z$
hu	S	Out-of-plane heat transfer coefficient, upside	$h_u$

---

TABLE 8-3: APPLICATION MODE VARIABLES, GENERAL HEAT TRANSFER

NAME	TYPE	DESCRIPTION	EXPRESSION
hd	S	Out-of-plane heat transfer coefficient, downside	$h_d$
Textz	S	Out-of-plane external temperature (1D)	$T_{ext,z}$
Textu	S	Out-of-plane external temperature, upside	$T_{ext,u}$
Textd	S	Out-of-plane external temperature, downside	$T_{ext,d}$
epsilonz	S	Out-of-plane emissivity (1D)	$\epsilon_z$
epsilonu	S	Out-of-plane emissivity, upside	$\epsilon_u$
epsilon_d	S	Out-of-plane emissivity, downside	$\epsilon_d$
Tambz	S	Out-of-plane ambient temperature (1D)	$T_{amb,z}$
Tambu	S	Out-of-plane ambient temperature, upside	$T_{amb,u}$
Tambd	S	Out-of-plane ambient temperature, downside	$T_{amb,d}$
Ac	S	Cross-sectional area	$A_c$
Pc	S	Cross-sectional perimeter	$P_c$
dz	S	Thickness	$d_z$
da	S	Total time scale factor	$\rho C_p$
gradT	S	Temperature gradient	$ \nabla T $
pA	S	Absolute pressure	$p_a$
H	S	Enthalpy	$H = C_p T / \gamma + p_a / \rho$
dflux	S	Conductive flux	$ k \nabla T $
cflux	S	Convective flux	$ \rho \mathbf{u} (H + 0.5 \mathbf{u}^2) $
tflux	S	Total heat flux	$\left  \rho u_i \left( H + \frac{1}{2} u_j u_j \right) - \tau_{ij} u_j - k_{ij} \frac{\partial T}{\partial x_j} \right $
dflux_xi	S (V)	Conductive flux, $x_i$ component	$\sum_j -k_{ij} \frac{\partial T}{\partial x_j}$
cflux_xi	S (V)	Convective flux, $x_i$ component	$\rho u_i (H + 0.5  \mathbf{u} )$



TABLE 8-3: APPLICATION MODE VARIABLES, GENERAL HEAT TRANSFER

NAME	TYPE	DESCRIPTION	EXPRESSION
tflux_xi	S (V)	Total heat flux, $x_i$ component	$\sum_j -k_{ij} \frac{\partial T}{\partial x_j} + \rho u_i (H + 0.5 \mathbf{u}^2)$
beta_xi	S	Convective field, $x_i$ component	$\rho C_p u_i$
Dm	S	Mean diffusion coefficient	$\frac{\sum_{i,j} k_{ij} \beta_i \beta_j}{ \beta ^2}$
cellPe	S	Cell Peclet number	$\left  \frac{\rho C_p \mathbf{u} h}{2k} \right $
res	S	Equation residual	$\begin{aligned} & \rho C_p \frac{\partial T}{\partial t} + \rho C_p \mathbf{u} \cdot \nabla T - \\ & \tau : \mathbf{S} + \frac{T \partial \rho}{\rho \partial T} \Big _p \left( \frac{\partial p}{\partial t} + \mathbf{u} \cdot \nabla p \right) - \\ & Q - q_s T - \nabla \cdot (k \nabla T) \end{aligned}$
qconv	S	Out-of-plane convective heat flux (1D, 2D)	$\begin{aligned} & h_z (T_{\text{ext}z} - T) , \\ & h_u (T_{\text{ext},u} - T) + \\ & h_d (T_{\text{ext},d} - T) \end{aligned}$
grad	S	Out-of-plane radiative heat flux (1D, 2D)	$\begin{aligned} & \varepsilon_z \sigma (T_{\text{amb}z}^4 - T^4) , \\ & \varepsilon_u \sigma (T_{\text{amb},u}^4 - T^4) + \\ & \varepsilon_d \sigma (T_{\text{amb},d}^4 - T^4) \end{aligned}$
da	S	Total time-scale factor	$\rho C_p$
d	S	Thickness	$d$
dr_guess	S	Default guess for width in radial direction of infinite element domain	$\Delta_r$
R0_guess	S	Default guess for inner radius of infinite element domain	$R_0$
Si	S	Infinite element $x_i$ coordinate	$s_i$

TABLE 8-3: APPLICATION MODE VARIABLES, GENERAL HEAT TRANSFER

NAME	TYPE	DESCRIPTION	EXPRESSION
$S0i\_guess$	S	Default guess for inner $x_i$ coordinate of infinite element domain	$S_{0i}$
$Sdi\_guess$	S	Default guess for width along $x_i$ coordinate of infinite element domain	$\Delta_i$
$dVol$	S	Volume integration contribution	$dV$
$q0$	B	Inward heat flux, heat source/sink	$q_0$
$T0$	B	Prescribed temperature	$T_0$
$h$	B	Heat transfer coefficient	$h$
$Tinf$	B	External temperature	$T_{inf}$
$epsilon$	B	Emissivity	$\epsilon$
$sigma$	B	Stefan-Boltzmann constant	$\sigma$
$Tamb$	B	Ambient temperature	$T_{amb}$
$J0$	B	Surface radiosity expression	$J_0$
$rhos$	B	Density of highly conductive layer	$\rho_s$
$Cps$	B	Specific heat of highly conductive layer	$C_{p,s}$
$ks$	B	Thermal conductivity of highly conductive layer	$k_s$
$ksxixj$	B	Thermal conductivity of highly conductive layer, tensor $x_i x_j$ component	$k_{s,ij}$
$ds$	B	Thickness of highly conductive layer	$d_s$
$ndflux$	B	Normal conductive flux	$\mathbf{n} \cdot (-k \nabla T)$
$ncflux$	B	Normal convective flux	$\mathbf{n} \cdot \rho \mathbf{u} (H + 0.5 \mathbf{u}^2)$
$ntflux$	B	Normal total heat flux	$n_i \left( \rho u_i \left( H + \frac{1}{2} u_j u_j \right) - \tau_{ij} u_j - k_{ij} \frac{\partial T}{\partial x_j} \right)$

TABLE 8-3: APPLICATION MODE VARIABLES, GENERAL HEAT TRANSFER

NAME	TYPE	DESCRIPTION	EXPRESSION
G	B	Surface irradiation	$G$
Gm	B	Mutual surface-to-surface irradiation	$G_m$
Famb	B	Ambient view factor	$F_{amb}$
rflux	B	Radiative flux	$J_0 - G$
U	B	Convection fluid velocity for h	1
L	B	Convection length scale for h	1
s	B	Convection local coordinate for h (3D)	$\sqrt{s_1^2 + s_2^2}$
dVolbnd	B	Area integration contribution	$dA$

**Note:** The variables  $G_m$  and  $F_{amb}$  are evaluated by the view factor evaluation code, which is encapsulated in an *elirradiation* element object (see *elirradiation* on page 94 in the *COMSOL Multiphysics Reference Guide*). After exporting the FEM structure to the COMSOL Script command line or MATLAB, you can inspect its element syntax representation by locating the element in the `fem.elemmph` field.

## Thin Conductive Shell

---

### APPLICATION MODE NAME

htsh

---

TABLE 8-4: DEPENDENT VARIABLES, GENERAL HEAT TRANSFER

NAME	DESCRIPTION
T	Temperature

TABLE 8-5: SCALAR VARIABLES, GENERAL HEAT TRANSFER

NAME	VALUE	UNIT	DESCRIPTION
sigma	$5.67 \cdot 10^{-8}$	$W/(m^2 \cdot K^4)$	Stefan-Boltzmann constant

TABLE 8-6: APPLICATION MODE VARIABLES, INCOMPRESSIBLE NAVIER-STOKES

NAME	TYPE	DESCRIPTION	EXPRESSION
Dts	B	Time-scale factor	$\delta_{ts}$
rho	B	Density	$\rho$
C	B	Heat capacity	$C_p$
ds	B	Thickness	$d_s$
k	B	Thermal conductivity	$k$
$kxixj$	B	Thermal conductivity tensor, local system $x_i x_j$ component	$k_{ij}$
Q	B	Heat source/sink	$Q$
qu	B	Out-of-plane inward heat flux, upside	$q_u$
qd	B	Out-of-plane inward heat flux, downside	$q_d$
hu	B	Out-of-plane heat transfer coefficient, upside	$h_u$
hd	B	Out-of-plane heat transfer coefficient, downside	$h_d$
Textu	B	Out-of-plane external temperature, upside	$T_{ext,u}$
Textd	B	Out-of-plane external temperature, downside	$T_{ext,d}$
epsilonu	B	Out-of-plane emissivity, upside	$\epsilon_u$
epsilond	B	Out-of-plane emissivity, downside	$\epsilon_d$

TABLE 8-6: APPLICATION MODE VARIABLES, INCOMPRESSIBLE NAVIER-STOKES

NAME	TYPE	DESCRIPTION	EXPRESSION
Tambu	B	Out-of-plane ambient temperature, upside	$T_{amb,u}$
Tambd	B	Out-of-plane ambient temperature, downside	$T_{amb,d}$
da	B	Total time-scale factor	$\delta_{ts}\rho C_p$
gradT	B	Temperature gradient	$ \nabla T $
tflux	B	Total heat flux	$ -k\nabla T $
tflux_xi	B (V)	Total heat flux, $x_i$ component	$\sum_j -k_{ij}\frac{\partial T}{\partial x_j}$
qconv	B	Out-of-plane convective heat flux	$h_{zu}(T_{ext,u} - T) +$ $h_{zd}(T_{ext,d} - T)$
grad	B	Out-of-plane radiative heat flux	$\epsilon_{zu}\sigma(T_{amb,u}^4 - T^4) +$ $\epsilon_{zd}\sigma(T_{amb,d}^4 - T^4)$
q0	E	Heat source/sink	$q_0$
de	E	Effective edge thickness	$d_e$
T0	E	Prescribed temperature	$T_0$
h	E	Heat transfer coefficient	$h$
Tinf	E	External temperature	$T_{inf}$
epsilon	E	Emissivity	$\epsilon$
Tamb	E	Ambient temperature	$T_{amb}$

## The Bioheat Equation

### APPLICATION MODE NAME

htbh

TABLE 8-7: DEPENDENT VARIABLES, GENERAL HEAT TRANSFER

NAME	DESCRIPTION
T	Temperature

TABLE 8-8: APPLICATION MODE VARIABLES, INCOMPRESSIBLE NAVIER-STOKES

NAME	TYPE	DESCRIPTION	EXPRESSION
Dts	S	Time-scale factor	$\delta_{ts}$
rho	S	Density of tissue	$\rho$
C	S	Specific heat of tissue	$C$
k	S	Thermal conductivity of tissue	$k$
$k_{xixj}$	S	Thermal conductivity tensor of tissue, $x_i x_j$ component	$k_{ij}$
rhob	S	Density of blood	$\rho_b$
Cb	S	Specific heat of blood	$C_b$
wb	S	Blood perfusion rate	$w_b$
Tb	S	Arterial blood temperature	$T_b$
Qmet	S	Metabolic heat source	$Q_{met}$
Qext	S	Spatial heat source	$Q_{ext}$
gradT	S	Temperature gradient	$ \nabla T $
flux	S	Total heat flux	$ -k\nabla T $
flux_xi	S (V)	Total heat flux, $x_i$ component	$\sum_j -k_{ij} \frac{\partial T}{\partial x_j}$
Qb	S	Blood perfusion heat source	$\rho_b C_b w_b (T_b - T)$
hb	S	Blood perfusion heat transfer coefficient	$\rho_b C_b w_b$
de	B	Effective edge thickness	$d_e$
q0	B	Inward heat flux	$q_0$
T0	B	Prescribed temperature	$T_0$
h	B	Heat transfer coefficient	$h$

TABLE 8-8: APPLICATION MODE VARIABLES, INCOMPRESSIBLE NAVIER-STOKES

NAME	TYPE	DESCRIPTION	EXPRESSION
Tinf	B	External temperature	$T_{\text{inf}}$
nflux	B	Normal heat flux	$\mathbf{n} \cdot (-k \nabla T)$

**INCOMPRESSIBLE NAVIER-STOKES**

**APPLICATION MODE NAME**

chns

TABLE 8-9: DEPENDENT VARIABLES, INCOMPRESSIBLE NAVIER-STOKES

NAME	DESCRIPTION
u, v, w	Velocity in the $x_1$ , $x_2$ , and $x_3$ directions
p	Pressure

TABLE 8-10: APPLICATION MODE VARIABLES, INCOMPRESSIBLE NAVIER-STOKES

NAME	TYPE	DESCRIPTION	EXPRESSION
rho	S	Density	$\rho$
eta	S	Dynamic viscosity	$\eta$
F	S	Volume force	$\mathbf{F}$
U	S	Velocity field	$ \mathbf{u} $
V	S	Vorticity	$\nabla \times \mathbf{u}$
K	B	Viscous force per unit area	$[\eta(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)] \cdot \mathbf{n}$
T	B	Total force per unit area	$[\eta(\nabla \mathbf{u} + (\nabla \mathbf{u})^T) - p\mathbf{I}] \cdot \mathbf{n}$
cellRe	S	Cell Reynolds number	$\frac{\rho \mathbf{u} h}{\eta}$
res	S	Equation residual	$\rho \mathbf{u}_t + \rho(\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p - \mathbf{F} - \nabla \cdot [\eta(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)]$
res_sc	S	Shock capturing residual	$\rho \mathbf{u}_t + \rho(\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p - \mathbf{F}$
beta	S	Convective term	$\rho \mathbf{u}$
da	S	Total time-scale factor	$\rho$
divU	S	Divergence of velocity field	$\nabla \cdot \mathbf{u}$
taum	S	GLS time scale	$\min\left(\frac{\Delta t}{\rho}, \frac{0.5h}{\max(\rho \mathbf{u} , \frac{6\eta}{h})}\right)$



TABLE 8-10: APPLICATION MODE VARIABLES, INCOMPRESSIBLE NAVIER-STOKES

NAME	TYPE	DESCRIPTION	EXPRESSION
tauc	S	GLS time scale	$0.5 \mathbf{u} h \min\left(1, \frac{\rho \mathbf{u} h}{\eta}\right)$
Dm	S	Mean diffusion coefficient	$\eta$
u0, v0, w0	B	Velocity (Velocity BC)	$\mathbf{u}_0$
p0	B	Pressure (Pressure BC)	$p_0$
f0	B	Normal stress (Normal stress BC)	$f_0$
Fbnd	B	Stress (General stress BC)	$\mathbf{F}$
U0in	B	Normal inflow velocity (Velocity BC)	$U_0$
U0out	B	Normal outflow velocity (Velocity BC)	$U_0$
uvw	B	Velocity of tangentially moving wall (Sliding wall BC, 2D)	$\mathbf{u}_w$
uw, vw, ww	B	Velocity of tangentially moving wall (Sliding wall BC, 3D)	$\mathbf{u}_w$
uwall	B	Velocity of moving wall (Moving/Leaking wall BC)	$\mathbf{u}_w$
U0	B	Average velocity (laminar inflow/outflow BC)	$U_0$
V0	B	Volume per time unit (laminar inflow/outflow BC 3D)	$V_0$
Lentr	B	Entrance length (laminar inflow BC)	$L_{entr}$
Lexit	B	Exit length (laminar outflow BC)	$L_{exit}$
p0_entr	B	Entrance pressure (laminar inflow BC)	$p_{0,entr}$
p0_exit	B	Exit pressure (laminar outflow BC)	$p_{0,exit}$

## SWIRL FLOW

---

### APPLICATION MODE NAME

---

chns

---

TABLE 8-11: DEPENDENT VARIABLES, SWIRL FLOW

NAME	DESCRIPTION
u, v, w	Velocity in the $r$ , $z$ , and $\varphi$ directions
$p$	Pressure

The application mode variables are identical to those of the Incompressible Navier-Stokes, see Table 8-10.

## TURBULENT FLOW, K- $\epsilon$ TURBULENCE MODEL

---

### APPLICATION MODE NAME

---

chns

---

TABLE 8-12: DEPENDENT VARIABLES, K- $\epsilon$  TURBULENCE MODEL

NAME	DESCRIPTION
u, v, w	Velocity in the $x_1$ , $x_2$ , and $x_3$ directions
$p$	Pressure
logk	Logarithm of turbulent kinetic energy
logd	Logarithm of turbulent dissipation rate

Table 8-13 shows the different scalar variables used in the  $k$ - $\epsilon$  Turbulence Model application mode:

TABLE 8-13: SCALAR VARIABLES, K- $\epsilon$  TURBULENCE MODEL

NAME	DESCRIPTION	EXPRESSION
C $_{\mu}$	Turbulence modeling constant (= 0.09)	$C_{\mu}$
C $_{d1}$	Turbulence modeling constant (= 1.44)	$C_{d1} = C_{\epsilon 1}$
C $_{d2}$	Turbulence modeling constant (= 1.92)	$C_{d2} = C_{\epsilon 2}$
sigma $_{k}$	Turbulence modeling constant (= 1.0)	$\sigma_k$
sigma $_{\epsilon}$	Turbulence modeling constant (= 1.3)	$\sigma_{\epsilon}$
kappa	von Kármán constant (=0.42)	$\kappa$
C $_{p+}$	Logarithmic wall function constant (= 5.5)	$C^+$

In addition to the application mode variables available for the Incompressible Navier-Stokes application mode, the following variables are available.

TABLE 8-14: APPLICATION MODE VARIABLES, K- $\epsilon$  TURBULENCE MODEL

NAME	TYPE	DESCRIPTION	EXPRESSION
k0	S	Turbulent kinetic energy	$k = e^{\log k}$
d0	S	Turbulent dissipation rate	$\epsilon = e^{\log d}$
etaT	S	Turbulent viscosity	$\rho C_{\mu} \frac{k^2}{\epsilon}$

### TURBULENT FLOW, K- $\omega$ TURBULENCE MODEL

#### APPLICATION MODE NAME

chns

TABLE 8-15: DEPENDENT VARIABLES, K- $\omega$  TURBULENCE MODEL

NAME	DESCRIPTION
u, v, w	Velocity in the $x_1$ , $x_2$ , and $x_3$ directions
p	Pressure
logk	Logarithm of turbulent kinetic energy
logw	Log of specific turbulent dissipation rate

Table 8-16 summarizes the different scalar variables used in the  $k$ - $\omega$  Turbulence Model application mode:

TABLE 8-16: SCALAR VARIABLES, K- $\omega$  TURBULENCE MODEL

NAME	DESCRIPTION	EXPRESSION
Cmu	Turbulence modeling constant (= 0.09)	$C_{\mu}$
alpha	Turbulence modeling constant (= 13/25)	$\alpha$
beta0	Turbulence modeling constant (= 9/125)	$\beta_0$
beta0k	Turbulence modeling constant (= 0.09)	$\beta_{0,k}$
sigmaw	Turbulence modeling constant (= 0.5)	$\sigma_{\omega}$
kappa	von Kármán constant (=0.42)	$\kappa$
Cplus	Logarithmic wall function constant (= 5.5)	$C^+$

In addition to the application mode variables available for the Incompressible Navier-Stokes application mode, the following variables are available.

TABLE 8-17: APPLICATION MODE VARIABLES, K- $\omega$  TURBULENCE MODEL

LABEL	TYPE	DESCRIPTION	EXPRESSION
k0	S	Turbulent kinetic energy	$k = e^{\log k}$
omega0	S	Specific turbulent dissipation rate	$\omega = e^{\log \omega}$
etaT	S	Turbulent viscosity	$\rho e^{\log k - \log \omega}$

### WEAKLY COMPRESSIBLE NAVIER-STOKES

APPLICATION MODE NAME
chns

TABLE 8-18: DEPENDENT VARIABLES, WEAKLY COMPRESSIBLE NAVIER-STOKES

NAME	DESCRIPTION
u, v, w	Velocity in the $x_1$ , $x_2$ , and $x_3$ directions
p	Pressure

The application mode variables are:

TABLE 8-19: APPLICATION MODE VARIABLES, WEAKLY COMPRESSIBLE NAVIER-STOKES

NAME	TYPE	DESCRIPTION	EXPRESSION
rho	S	Density	$\rho$
eta	S	Dynamic viscosity	$\eta$
kappadv	S	Dilatational viscosity	$\kappa$
F	S	Volume force	$\mathbf{F}$
U	S	Velocity field	$ \mathbf{u} $
V	S	Vorticity	$\nabla \times \mathbf{u}$
K	B	Viscous force per area	$\left[ \eta(\nabla \mathbf{u} + (\nabla \mathbf{u})^T) - \left( \frac{2\eta}{3} - \kappa \right) (\nabla \cdot \mathbf{u}) \mathbf{I} \right] \cdot \mathbf{n}$
T	B	Total force per area	$\left[ -p \mathbf{I} + \eta(\nabla \mathbf{u} + (\nabla \mathbf{u})^T) - \left( \frac{2\eta}{3} - \kappa \right) (\nabla \cdot \mathbf{u}) \mathbf{I} \right] \cdot \mathbf{n}$
cellRe	S	Cell Reynolds number	$\frac{\rho  \mathbf{u}  h}{\eta}$

TABLE 8-19: APPLICATION MODE VARIABLES, WEAKLY COMPRESSIBLE NAVIER-STOKES

NAME	TYPE	DESCRIPTION	EXPRESSION
res	S	Equation residual	$\rho \mathbf{u}_t + \rho(\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p - \mathbf{F} -$ $\nabla \cdot \left[ \eta(\nabla \mathbf{u} + (\nabla \mathbf{u})^T) - \left( \frac{2\eta}{3} - \kappa \right) (\nabla \cdot \mathbf{u}) \mathbf{I} \right]$
res_sc	S	Shock capturing residual	$\rho \mathbf{u}_t + \rho(\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p - \mathbf{F}$
beta	S	Convective field	$\rho \mathbf{u}$
Dm	S	Mean diffusion coefficient	$\eta$
da	S	Total time-scale factor	$\rho$
divU	S	Divergence of velocity field	$\nabla \cdot \mathbf{u}$
taum	S	GLS time scale	$\min \left( \frac{\Delta t}{\rho}, \frac{0.5h}{\max(\rho \mathbf{u} , \frac{6\eta}{h})} \right)$
tauc	S	GLS time scale	$0.5 \mathbf{u} h \min \left( 1, \frac{\rho \mathbf{u} h}{\eta} \right)$



## Glossary

This glossary contains application-specific terms used in the Heat Transfer Module software and documentation. For information about terms relating to finite element modeling, mathematics, geometry, and CAD, see the glossary in the *COMSOL Multiphysics User's Guide*. For references to more information about a term, see the Index in this or other manuals.

# Glossary of Terms

**anisotropy** The condition of exhibiting properties with different values when measured in different directions.

**bioheat equation** An alternative form of the heat equation that incorporates the effects of blood perfusion, metabolism, and external heating. The equation describes heat transfer in tissue.

**blackbody** A blackbody is a surface that absorbs all incoming radiation, that is, it does not reflect radiation. The blackbody also emits the maximum possible radiation.

**conduction** Heat conduction takes place through different mechanisms in different media. Theoretically, conduction takes place through collisions of molecules in a gas, through oscillations of each molecule in a “cage” formed by its nearest neighbors in a fluid, and by the electrons carrying heat in metals or by molecular motion in other solids. Typical for heat conduction is that the heat flux is proportional to the temperature gradient.

**convection** Heat convection takes place through the net displacement of a fluid, which translates the heat content in a fluid through the fluid's own velocity. The term convection is also used for the heat dissipation from a solid surface to a fluid, where the heat transfer coefficient and the temperature difference across a fictitious film describes the flux. The Heat Transfer Module includes both descriptions.

**emissivity** A dimensionless factor between 0 and 1 that specifies the ability of a surface to emit radiative energy. The value 1 corresponds to an ideal surface, which emits the maximum possible radiative energy.

**heat capacity** See *specific heat*.

**highly conductive layer** A highly conductive layer is a thin layer on a boundary. It has much higher thermal conductivity than the material in the adjacent subdomain. This allows you to assume that the temperature is constant across the layer's thickness. The General Heat Transfer application mode supports heat transfer in highly conductive layers.



**irradiation** The total radiation that arrives at a surface.

**Navier-Stokes equations** The equations for the momentum balances coupled to the equation of continuity for a Newtonian incompressible fluid are often referred to as the Navier-Stokes equations. The most general versions of the Navier-Stokes equations do, however, describe fully compressible flows.

**opaque material** An opaque body does not transmit any radiative heat flux, that is, the surface of an opaque body has a transmissivity equal to 0.

**radiation** Heat transfer by radiation takes place through the transport of photons, which can be absorbed or reflected on solid surfaces. The Heat Transfer Module includes surface-to-surface radiation, which accounts for effects of shading and reflections between radiating surfaces. It also includes surface-to-ambient radiation where the ambient radiation can be fixed or given by an arbitrary function.

**radiosity** The total radiation that leaves a surface, that is, both the emitted and the reflected radiation.

**specific heat** Refers to the quantity that represents the amount of heat required to change one unit of mass of a substance by one degree. It has units of energy per mass per degree. This quantity is also called *specific heat* or *specific heat capacity*.

**specific heat capacity** See *specific heat*.

**thin conductive shell** An application mode for modeling heat transfer in a thin shell. “Thin” means that the shell is thin enough, or has high enough thermal conductivity, for you to assume that the temperature is constant across the shell’s thickness. See also *highly conductive layer*.

**transparent material** A transparent body transmits radiative heat flux, that is, the surface of a transparent body has a transmissivity greater than 0.

**thermal conductivity** The definition of thermal conductivity is given by Fourier’s law, which relates the heat flux to the temperature gradient. In this equation, the thermal conductivity is the proportional constant.



# I N D E X

- A**
    - absolute pressure 32
    - ambient temperature 72
    - application mode properties
      - azimuthal sectors 27
      - cache view factors 27
      - radiation integration order 27
      - radiation resolution 27
      - surface-to-surface radiation method 27
      - swirl velocity 27
      - turbulence model 27
      - weak constraints 27, 155
    - application modes
      - Bioheat Equation 172
      - General Heat Transfer 24, 271
      - Incompressible Navier-Stokes 182
      - Thin Conductive Shell 154, 276
      - Weakly Compressible Navier-Stokes 223
    - application scalar variables 269
    - application variables 269
    - arterial blood temperature 173, 175
    - artificial diffusion 193
    - assembly pairs, boundary conditions on 55
    - average Nusselt number 18
    - axial symmetry boundary condition 58, 177
    - axisymmetric radiation 70
    - azimuthal sectors 27, 70
  - B**
    - Babuska-Brezzi condition 195
    - bioheat equation 172
    - blood density 175
    - blood perfusion rate 173, 175
    - boundaries, grouping of 74
    - boundary conditions 54
    - axial symmetry 58
    - compatible with radiation 71
    - continuity 60
    - convective flux 58
    - Dirichlet type 9
    - for turbulence models 211
    - heat flux 56
    - heat source/sink 59, 74
    - Neumann type 9
    - on pairs 55
    - specified heat flux 9
    - specified temperature 9
    - specifying 56
    - symmetric 176
    - temperature 58
    - thin thermally resistive layer 55, 60
    - types of 56
  - boundary properties 60
  - boundary settings
    - turbulence intensity 213
    - turbulent length scale 213
  - Boussinesq approximation 198, 232
  - bulk coolant velocity 64
  - bulk temperature of coolant 64
  - buoyancy forces 114
  - buoyancy-driven flow 114
- C**
  - caching view factors 27
  - CFL number 200
  - characteristic length 17
  - chock capturing 195
  - closure 208
  - closure relations 206
  - COMSOL Material Library 44
  - conduction 20
    - in heat transfer 6
  - conductive heat transfer 30

- conductive shells 154
- contact pair 55
- continuity 55
- continuity boundary condition 60
- continuity equation 180
- control volumes 110
- convection 20
  - forced vs. natural 16
  - in heat transfer 6
- convection length scale 63
- convective cooling 15
- convective flux 9
- convective flux boundary condition 58
- convective heat transfer 31
- cooling power 116
- correlations, of Nusselt numbers 18
- crosswind diffusion 195
- crosswind diffusion stabilization 40
- D** density 32
  - of blood 175
  - of tissue 175
- dependent variables 269
- diatomic gases 32
- direct area integration method 69
- Dirichlet boundary condition 9
- displacement ventilation system 248
- dissipation rate 209
- E** eddy viscosity 206, 208
- element order 76
- energy, transferred through shell 54
- enthalpy 33
- extended Kays-Crawford model 52
- external bulk temperature 176
- external forced convection 64
- external natural convection 64
- F** forced convection 16
  - Fourier's law of conduction 7
  - Fourier's law of heat conduction 158, 175
  - free convection 231
- G** General Heat Transfer application mode
  - 24, 271
  - gradient, of temperature 80, 175
  - Grashof number 17
  - grouping of boundaries 74
- H** heat capacity 32, 78, 173
- heat equation 7
- heat flux
  - boundary condition for 56, 177
  - vector 80, 175
- heat flux discontinuity 55, 177
- heat source from metabolism 173
- heat source/sink boundary condition 59
  - with radiation 74
- heat sources 32
  - from metabolism 175
- heat transfer 20
  - by conduction 6
  - by convection 6
  - by radiation 6
  - conductive 20
  - convective 20
  - radiative 20
- heat transfer and turbulent flow, coupled 248
- heat transfer coefficient 54, 57, 176
  - for convective cooling 15
  - predefined 61
- Heat Transfer Coefficients library 18, 61
  - details of 63
  - materials in 65
- Heat Transfer Module 3.5, new features
  - in 3
- hemicube method 69
- highly conductive layers 77
  - model 142

- highly conductive shells, energy transferred through 54
- I**
  - ideal gas 33
    - settings for 34
  - identity pair 55
  - Incompressible Navier-Stokes application mode 182
  - incompressible Navier-Stokes equations 182
  - infinite elements 34
  - influx from radiation 54
  - integral cooling power 116
  - interior boundary 54
  - internal forced convection 64
  - internal natural convection 64
  - inward heat flux 54
  - irradiation 10, 67
  - isotropic diffusion 196
  - isotropic diffusion stabilization 42
- K**
  - Kármán's constant 212
  - Kays-Crawford model 52
  - k-epsilon turbulence model 209
  - k-omega turbulence model 210
- L**
  - Lagrange multipliers 54
  - laminar buoyancy-driven flow 114
  - length scales 63, 64
  - Liquids and Gases material library 44
  - local coordinates, in shells 158
  - local Nusselt number 18
  - logarithmic wall functions 212
  - low Mach number flows 8
- M**
  - Mach number 223
  - mapped infinite elements 35
  - master and slave pairs 56
  - material libraries 44
  - material properties
    - from material libraries 42
  - matter state 33
    - gas/liquid 33
    - ideal gas 33
    - solid 33
  - metabolic heat sources 175
  - micro fuel cells 91
  - molar mass, number-average 34
- N**
  - NAFEMS benchmark 87
  - natural convection 16, 231
  - natural-convection problems 127
  - Navier-Stokes equations 180, 182
    - residual approximation for 196
  - Neumann boundary condition 9
  - new features in version 3.5 3
  - non-isothermal flow 231
  - Non-Isothermal Flow predefined multi-physics coupling 228
  - non-Newtonian fluids 181
  - normal heat flux continuity 55
  - normal vector of the boundary 54
  - number-average molar mass 34
  - numerical stability 193
  - Nusselt correlations 64
  - Nusselt number 16, 237
    - average 18
    - local 18
- O**
  - opaque subdomains 73
  - out-of-plane ambient radiation 155
  - out-of-plane heat transfer 81
    - model 146
- P**
  - Pennes' approximation 172
  - perfectly matched layers 34
  - perfusion rate of blood 175
  - Prandtl number 16, 237
  - pressure work 30, 33
  - production/absorption coefficient 32
- R**
  - radiation 20

- in axisymmetric geometries 70
- in heat transfer 6
- surface-to-ambient 11
- surface-to-surface 11
- radiation integration order 27
- radiation resolution 27
- radiative heat transfer 67
  - enabling 71
- radiosity 10, 25
- radiosity method 67
- RANS 207
- ratio of specific heats 32
- Rayleigh number 16, 237
- realizability constraints 219
- reference bulk temperature 54
- residuals, of the Navier-Stokes equations 196
- Reynolds number 16, 17, 206
- Reynolds stress tensor 208
- Reynolds-averaged Navier-Stokes equations 206, 207

## S

- shells
  - local coordinate system for 158
  - thin conductive 154
- SI units 47
- solving natural-convection problems 127
- solving surface-to-surface radiation problems 76
- source and destination pairs 56
- spatial heat source 175
- specific heat
  - of blood 175
  - of tissue 175
- specified heat flux 9
- specified temperature 9
- SSOR preconditioner/smoothen 199
- stabilization 38
  - crosswind diffusion 40

- isotropic diffusion 42
  - streamline diffusion 40
- stagnation point anomaly 220
- Stefan-Boltzmann constant 28, 67
- strain rate tensor 180
- streamline diffusion 195
- streamline diffusion stabilization 40
- subdomain settings
  - ideal gas 34
  - stabilization 38
- surface emissivity 72
- surface radiosity expression 72
- surface temperature 64
- surface-to-ambient radiation 11, 68
- surface-to-surface radiation 11, 68
  - direct area integration method 69
  - guidelines for solving 76
  - hemicube method 69
- surface-to-surface radiation method 27
- swirl flow 203
- swirl velocity 27
- swirl velocity property 27, 203
- symmetry boundary condition 176

## T

- temperature
  - boundary condition for 73
  - temperature boundary condition 58, 73, 177
  - temperature continuity 55
  - temperature gradient 80, 175
  - thermal wall functions 58
  - thermal conductivity 17, 32, 78, 91, 173
  - thermal insulation 9, 176
  - thermally resistive layer 55, 60
  - thermos model 109
  - Thin Conductive Shell 276
  - Thin Conductive Shell application mode 154, 276
  - thin conductive shells 154

- time-scaling coefficient
  - in the bioheat equation 175
  - in the heat equation 84
- tissue density 175
- turbulence modeling 51
- turbulence models
  - k-epsilon 209
  - k-omega 210
  - k-omega vs. k-epsilon 211
  - property for selecting 27
  - switching 221
- turbulent flow 206
- turbulent heat conductivity 32
- turbulent kinetic energy 209
- turbulent length scale 214
- Turbulent Non-Isothermal Flow prede-
  - finied multiphysics coupling 229
- turbulent Prandtl number
  - extended Kays-Crawford model 52
  - Kays-Crawford model 52
  - models for 52
- turbulent viscosity 208
- turbulent weakly compressible flow 226
- typographical conventions 4
- U** unbounded domains, modeling of 34
- universal gas constant 28
- up and down pairs 56
- V** Vanka preconditioner/smoothen 199
- view factors 68
  - caching 27
- viscous heating 30, 33
- viscous length scale 213
- viscous stress tensor 181
- viscous sublayer 212
- volume force 232
- W** wall functions 211
  - thermal 58
- weak constraints 27, 155
- weakly compressible flow 223
- Wilcox k-omega model 210

