Model created in COMSOL Multiphysics 6.4

Temperature Distribution in a Vacuum Flask

This model is licensed under the COMSOL Software License Agreement 6.4.
All trademarks are the property of their respective owners. See www.comsol.com/trademarks.

Introduction

The following example solves for the temperature distribution within a vacuum flask holding hot coffee. The main interest here is to illustrate how to use MATLAB® functions to define material properties and boundary condition directly within the COMSOL model.

Two MATLAB functions are used to define the temperature dependent thermal conductivity of the vacuum flask shell and the insulation foam, while a third function defines the heat transfer coefficient that corresponds to a natural convective cooling for a vertical plate and surrounding air.

Model Definition

Assume axial symmetry for this simulation to reduce the model geometry to the 2D cross section of the vacuum flask geometry shown in Figure 1.

The vacuum flask consists of a steel shell isolated with a foam material, and a cork made of nylon. On the inside wall apply a constant temperature, assuming that the vacuum flask is filled with coffee of constant temperature.

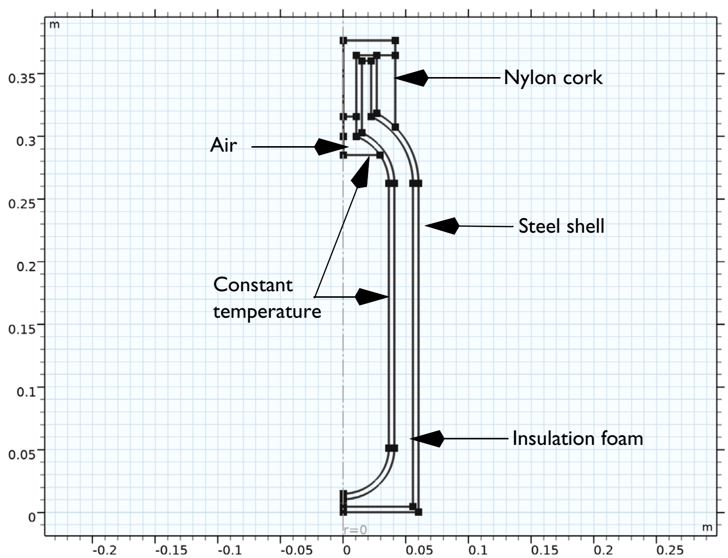

Figure 1: Cross section of the vacuum flask geometry.

Define the temperature dependent thermal conductivity of the insulating foam according to the following polynomial expression:

$$k_{\mathrm{foam}} = -2.141 + 1.87e^{-2}T - 5.3e^{-5}T^2 + 4.945e^{-8}T^3$$

The above expression is plotted in Figure 2 for the temperature interval 293 K – 400 K.

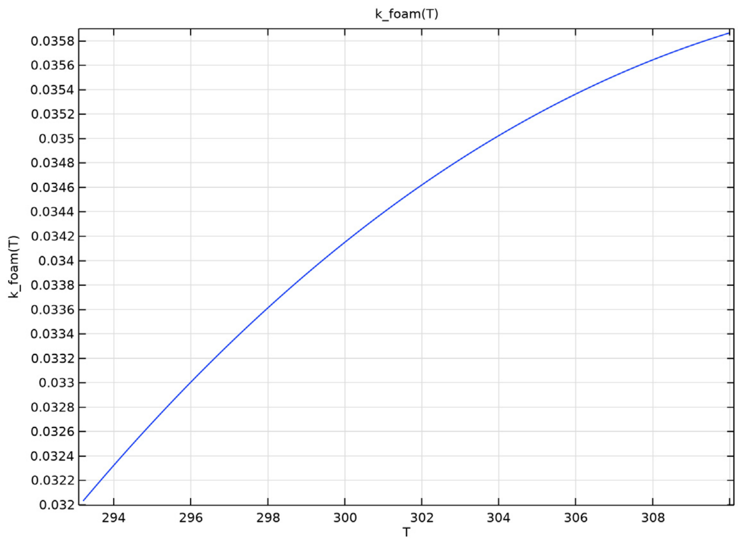

Figure 2: Thermal conductivity of the insulating foam versus temperature.

Use a different polynomial expression for the thermal conductivity of steel according to

$$k_{\mathrm{steel}} = 71.12 - 0.115T + 1.16e^{-4}T^2 - 4.25e^{-8}T^3$$

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_{surface}, engineers must estimate T_{surface} and then iterate between h and T_{surface} to obtain a converged value for h. The following expression for h is a typical handbook correlation for the case of natural convection in air on a vertical heated wall:

$$h = \frac{k \cdot \mathrm{Nu}}{L}$$

$$\mathrm{Nu} = \left(825e^{-3} + \frac{387e^{-3} \cdot \mathrm{Ra}^{\frac{1}{6}}}{\left(1 + \left(\frac{492e^{-3}}{\mathrm{Pr}}\right)^{\frac{9}{16}}\right)^{\frac{8}{27}}}\right)^2$$

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

In the above equations, T_{ambient} is the temperature of the air surrounding the vacuum flask, and all other variables are defined in the function script, under step 6 in the section Modeling Instructions — MATLAB®.

Figure 3 illustrates the expression of heat transfer coefficient as function of the wall temperature, the ambient temperature and the wall length.

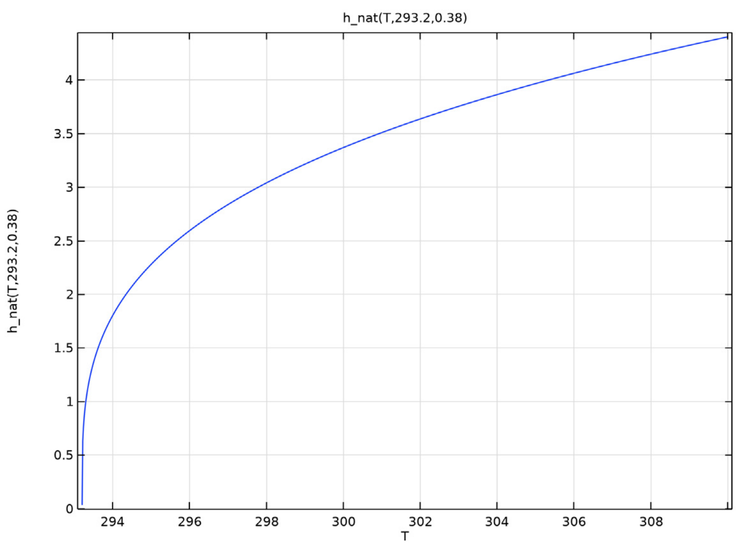

Figure 3: Heat transfer coefficient for natural convection cooling of a vertical wall versus the surface temperature. The ambient temperature and the wall length are given constant, $T_{\mathrm{ambient}} = 293.2\ K$ *and* $L = 38\ cm$*, respectively.*

Results and Discussion

From the temperature distribution in the vacuum flask wall shown in Figure 4 you can see that most of the temperature gradients are in the foam, which shows that the material works well for insulating the vacuum flask.

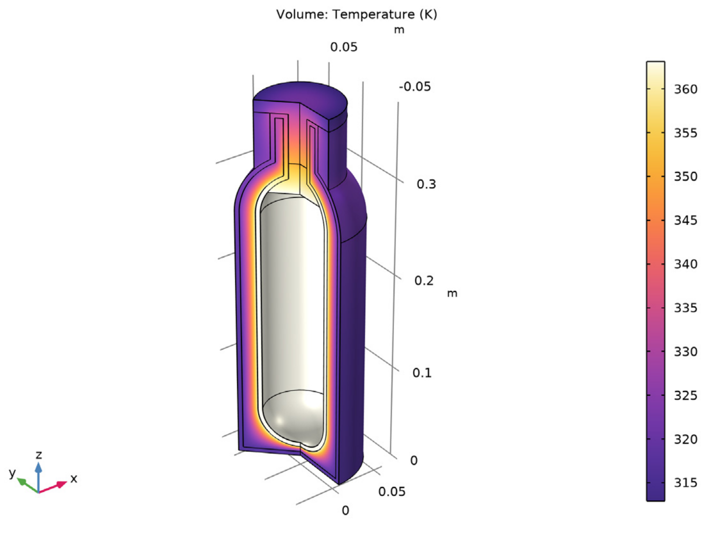

Figure 4: Temperature distribution in the vacuum flask.

Figure 5 shows the heat transfer coefficient along the vacuum flask surface. Its value depends on the temperature surface, and the discontinuity in the curve is at the boundary between the cap and the steel wall.

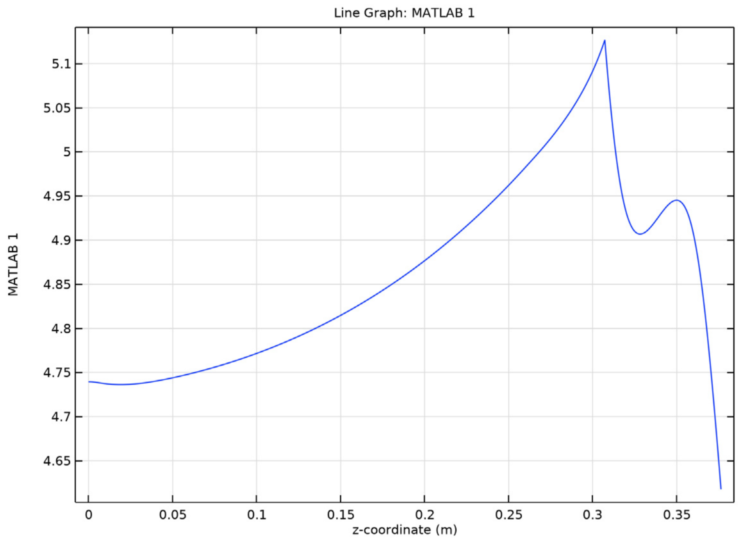

Figure 5: Heat transfer coefficient along the surface of the vacuum flask.

Notes About the COMSOL Implementation

To use an external MATLAB function in a COMSOL model add a MATLAB feature node to the tree in the model builder, and specify the functions there. Once you solve the model, COMSOL automatically starts a MATLAB process that evaluates the functions and returns the value to the COMSOL model.

Application Library path: `COMSOL_Multiphysics/LiveLink_for_MATLAB/Tutorials/vacuum_flask_llmatlab`

Modeling Instructions — MATLAB®

Before the implementation of the model in the COMSOL Desktop®, define the MATLAB functions for the thermal conductivity of the insulating foam, and the convective heat transfer coefficient.

1 Start the MATLAB editor or any text editor.

2 In a new file enter the following:

```
function out = k_foam(T)
% k_foam returns a value for the insulation foam thermal
% conductivity that vary with the local temperature
out = -2.141+1.87e-2*T-5.3e-5*T.^2+4.945e-8*T.^3;
```

3 Save the file as `k_foam.m`.

4 Define now the function derivative with respect to T. Open a new file and enter the lines below:

```
function out = k_foam_dT(T)
% Derivative with respect of T of the function k_foam
out = 1.87e-2-10.6e-5*T+14.835e-8*T.^2;
```

5 Save the file as `k_foam_dT.m`.

6 Define the heat transfer coefficient for the cooling by natural convection:

```
function out = h_nat(T, Tamb, L)
% h_nat returns the value of the heat transfer coefficient
% corresponding to the cooling of a vertical plate by air in
% natural convection. The heat transfer coefficient depends
% on the surface temperature T, the temperature of the
% surrounding air Tamb and the length of the wall

% Air properties
k = 2e-2; % Thermal conductivity [W/(m*K)]
rho = 1; % Density [kg/m^3]
cp = 1e3; % Heat capacity [J/(kg*K)]
mu = 1e-6; % Dynamic viscosity [Pa*s]
g = 9.81; % Gravity [m/s^2]

% Definition of Prandt number
Pr = cp*mu/k;
% Definition of Rayleigh number
Ra = 2*g*rho^2*cp/(mu*k).*abs((T-Tamb)./(T+Tamb)).*L.^3;
% Definition of Nusselt number
num = 387e-3*Ra.^(1/6);
denom = (1+(492e-3/Pr)^(9/16))^(8/27);
Nu = (825e-3+num/denom).^2;
% Return the heat transfer coefficient value
out = k.*Nu./L;
```

7 Save the file as `h_nat.m`.

8 Finally, create one more file to define the function derivative with respect to T. In this model we assume the ambient temperature and the wall length as constant. In a new file enter:

```
function out = h_nat_dT(T, Tamb, L)
```

```
% Air properties
k = 2e-2; % Thermal conductivity [W/(m*K)]
rho = 1; % Density [kg/m^3]
cp = 1e3; % Heat capacity [J/(kg*K)]
mu = 1e-6; % Dynamic viscosity [Pa*s]
g = 9.81; % Gravity [m/s^2]

Pr = cp*mu/k;
Ra = 2*g*rho^2*cp/(mu*k).*abs((T-Tamb)./(T+Tamb)+eps).*L.^3;
Ra_dT = 2*g*rho^2*cp/(mu*k)*2.*Tamb./(T+Tamb).^2.*L.^3;
num = 387e-3*Ra.^(1/6);
num_dT = 387e-3/6*Ra.^(-5/6).*Ra_dT;
denom = (1+(492e-3/Pr)^(9/16))^(8/27);
Nu_dT = num_dT.*(2*825e-3+2*num/denom)/denom;
out = k.*Nu_dT./L;
```

9 Save the file as `h_nat_dT.m`.

Note: It is important to have the same size for the function output as for the input argument. As COMSOL Multiphysics assembles the problem in blocks, the variables have the size of an array. To prevent possible size related issues, use pointwise operators (`.*`, `./`, `.^`) with the input arguments.

SETTING THE DIRECTORY PATH IN MATLAB

To be able to solve the model in the COMSOL Desktop, you need to make sure that the path to the MATLAB functions that are used in the model is set in MATLAB. Refer to the section *Setting the Function Directory Path in MATLAB®* in the *LiveLink™ for MATLAB® User's Guide* for details on how do to this.

Modeling Instructions — COMSOL Desktop

From the **File** menu, choose **New**.

NEW

In the **New** window, click **Model Wizard**.

MODEL WIZARD

1 In the **Model Wizard** window, click **2D Axisymmetric**.

2 In the **Select Physics** tree, select **Heat Transfer > Heat Transfer in Solids (ht)**.

3 Click **Add**.

4 Click **Study**.

5 In the **Select Study** tree, select **General Studies** > **Stationary**.

6 Click **Done**.

GLOBAL DEFINITIONS

Parameters 1

1 In the **Model Builder** window, under **Global Definitions** click **Parameters 1**.

2 In the **Settings** window for **Parameters**, locate the **Parameters** section.

3 In the table, enter the following settings:

Name	Expression	Value	Description
Length	38[cm]	0.38 m	Height of the vacuum flask
T_amb	20[degC]	293.15 K	Temperature of surrounding air

MATLAB 1

1 In the **Home** toolbar, click f(X) **Functions** and choose **Global** > **MATLAB**.

2 In the **Settings** window for **MATLAB**, locate the **Functions** section.

3 In the table, enter the following settings:

Function name	Arguments
k_foam	T

4 Click to expand the **Plot Parameters** section. In the table, enter the following settings:

Lower limit	Upper limit
293.2	310

5 Click **Plot**.

6 Locate the **Functions** section. In the table, enter the following settings:

Function name	Arguments
k_foam	T
h_nat	T, Tamb, L

7 Click **Move Up**.

8 Locate the **Plot Parameters** section. In the table, enter the following settings:

Lower limit	Upper limit
293.2	293.2
0.38	0.38

9 Click **Plot**.

10 Click to expand the **Derivatives** section. In the table, enter the following settings:

Function name	Argument	Partial derivative
k_foam	T	k_foam_dT(T)
h_nat	T	h_nat_dT(T,T_amb,Length)
h_nat	Tamb	0
h_nat	L	0

11 Locate the **Functions** section. In the table, enter the following settings:

Function name	Arguments
h_nat	T, Tamb, L
k_foam	T
k_foam_dT	T
h_nat_dT	T, Tamb, L

GEOMETRY 1

Import 1 (imp1)

1. In the **Home** toolbar, click **Import**.
2. In the **Settings** window for **Import**, locate the **Source** section.
3. Click **Browse**.
4. Browse to the model's Application Libraries folder and double-click the file `vacuum_flask_llmatlab.dxf`.
5. Click **Import**.

Form Union (fin)

1. In the **Model Builder** window, right-click **Form Union (fin)** and choose **Build Selected**.
2. Click the **Zoom Extents** button in the **Graphics** toolbar.
3. In the **Settings** window for **Form Union/Assembly**, click **Build Selected**.

DEFINITIONS

Variables 1

1 In the **Definitions** toolbar, click **Local Variables**.

2 In the **Settings** window for **Variables**, locate the **Variables** section.

3 In the table, enter the following settings:

Name	Expression	Unit	Description
k_steel	71.12[W/(m*K)]-0.115[W/(m*K^2)]*T+1.16E-4[W/(m*K^3)]*T^2-4.25E-8[W/(m*K^4)]*T^3	W/(m·K)	Steel thermal conductivity

MATERIALS

In the **Model Builder** window, under **Component 1 (comp1)** right-click **Materials** and choose **Browse Materials**.

MATERIAL BROWSER

1 In the **Material Browser** window, select **Built-in** > **Nylon** in the tree.

2 Click **Add to Component**.

3 In the tree, select **Built-in** > **Air**.

4 Click **Add to Component**.

5 Click **Done**.

6 Select Domain 3 only.

MATERIALS

Steel

1 In the **Materials** toolbar, click **Blank Material**.

2 Select Domain 1 only.

3 In the **Settings** window for **Material**, locate the **Material Contents** section.

4 In the table, enter the following settings:

Property	Variable	Value	Unit	Property group
Thermal conductivity	k_iso ; kii = k_iso, kij = 0	`k_steel`	W/(m·K)	Basic
Density	rho	`7850`	kg/m³	Basic
Heat capacity at constant pressure	Cp	`475`	J/(kg·K)	Basic

5 Right-click **Material 3 (mat3)** and choose **Rename**.

6 In the **Rename Material** dialog, type `Steel` in the **New label** text field.

7 Click **OK**.

Foam

1 In the **Materials** toolbar, click **Blank Material**.

2 Select Domain 2 only.

3 In the **Settings** window for **Material**, locate the **Material Contents** section.

4 In the table, enter the following settings:

Property	Variable	Value	Unit	Property group
Thermal conductivity	k_iso ; kii = k_iso, kij = 0	`k_foam(T)`	W/(m·K)	Basic
Density	rho	`24`	kg/m³	Basic
Heat capacity at constant pressure	Cp	`2.3`	J/(kg·K)	Basic

5 Right-click **Material 4 (mat4)** and choose **Rename**.

6 In the **Rename Material** dialog, type `Foam` in the **New label** text field.

7 Click **OK**.

HEAT TRANSFER IN SOLIDS (HT)

Temperature 1

1 In the **Physics** toolbar, click **Boundaries** and choose **Temperature**.

2 Select Boundaries 7, 20, 27, and 32 only.

3 In the **Settings** window for **Temperature**, locate the **Temperature** section.

4 In the T_0 text field, type `90[degC]`.

Heat Flux 1

1 In the **Physics** toolbar, click **Boundaries** and choose **Heat Flux**.

2 Select Boundaries 11, 22, 23, 25, and 33 only.

3 In the **Settings** window for **Heat Flux**, locate the **Heat Flux** section.

4 From the **Flux type** list, choose **Convective heat flux**.

5 In the h text field, type `h_nat(T,T_amb,Length)`.

6 In the T_{ext} text field, type `T_amb`.

MESH 1

1 In the **Model Builder** window, under **Component 1 (comp1)** click **Mesh 1**.

2 In the **Settings** window for **Mesh**, locate the **Physics-Controlled Mesh** section.

3 From the **Element size** list, choose **Extremely fine**.

4 Click **Build All**.

STUDY 1

In the **Study** toolbar, click **Compute**.

RESULT TEMPLATES

1 In the **Results** toolbar, click **Result Templates** to open the **Result Templates** window.

2 Go to the **Result Templates** window.

3 In the tree, select **Study 1/Solution 1 (sol1)** > **Heat Transfer in Solids** > **Temperature (ht)**.

4 Click the **Add Result Template** button in the window toolbar.

5 In the **Results** toolbar, click **Result Templates** to close the **Result Templates** window.

RESULTS

1D Plot Group 3

In the **Results** toolbar, click **1D Plot Group**.

Line Graph 1

1 In the **Model Builder** window, right-click **1D Plot Group 3** and choose **Line Graph**.

2 Select Boundaries 22, 23, 25, and 33 only.

3 In the **Settings** window for **Line Graph**, locate the **y-Axis Data** section.

4 In the **Expression** text field, type `h_nat(T,T_amb,Length)`.

5 Locate the **x-Axis Data** section. From the **Parameter** list, choose **Expression**.

6 In the **Expression** text field, type `z`.

7 In the **1D Plot Group 3** toolbar, click **Plot**.