Institutionen för datavetenskap
Umeå universitet
Martin Berggren

3 oktober 2011
Teknisk beräkningsvetenskap II
Fallstudie 1

Lab 1: Equation-based Modeling in Comsol Multiphysics

1 Introduction

The labs for Case Study 1 will use Comsol Multiphysics, a highly flexible general-purpose finite-element package: it can be used as an engineering tool for predefined modeling situations, but also as a numerical solver for partial differential equations, for which the user explicitly defines the equation or the associated variational form. Comsol can either be run from its own Graphical User Interface (GUI) or can be set up to interface with Matlab. In this course, we will only use the GUI.

For engineering use, there are readymade "canned solutions," such as the stress in a loaded wrench or the sound field in a muffler. The user may use these as a starting point and modify the geometry (it is for instance possible to import CAD descriptions), associated parameter values (Young's modulus, speed of sound, . . .), and boundary conditions in order to obtained something close to what is requested. Using Comsol in this way, the user does not need to know exactly what equations, what variational form, or what elements that Comsol uses. (All this information is available under the hood within the GUI, however). In order to see the available predefined solutions, activate a tab in the Comsol window denoted Model Library. (If there is no such tab, it can be created by selecting **Model Library** under the **View** menu.) Then click on the ▶ to the left of **COMSOL Multiphysics** in order to see the predefined models from Acoustics, Chemical Engineering, etc. Additional "modules" (CFD, Structural Mechanics. . .) containing even more predefined models are sold separately. Your available version probably contains the **Structural Mechanics Module** in addition to the basic **COMSOL Multiphysics** folder.

Within this course, however, we will not use the readymade solutions, but an "expert mode" which allows precise specifications of the equations, the boundary conditions, and the elements used. There are three conceptually different ways in the "expert mode" to specify equations and boundary conditions, namely through the *coefficient form*, the *general form*, or the *weak form*. In the coefficient form, the problem is specified through the setting of individual coefficients in a quite general system of partial differential equations and associated boundary conditions. In the general form, the equation is specified through the definition of a so-called flux function in a what is known as a "conservation-law" formulation of the equation. The weak form is the most general and precise way to specify the equations in a finite element context, and it is this formulation we will use below. In the weak form, the equation is specified by specifying the integrands that occur when writing the equation in a variational form, as in expression (2) below.

Remark 1 (regarding terminology). In the lectures of this course, we use the term "variational form" for equations like (2), whereas Comsol uses the term "weak form" for the same object. Note also that the use of the term "weak form" does not in any way indicate that this form is inferior![1] The situation is in fact the opposite; the "weak form" is the most general and precise way to specify the equation to be used in a finite-element context, since all finite-element methods operate through approximations of a properly specified weak (or variational) form.

2 Getting familiar with Comsol Muliphysics

This lab constitutes an introduction to the use of Comsol Multiphysics. This lab is useful even if you have some experience of Comsol from previous courses. The version of the software used

[1] The reason for the terminology is that the solution to a variational problem constitutes a *weak* (or *generalized*) solution to the partial differential equation; that is, a solution that may contain less smoothness than the form of the partial differential equation would initially suggest.

in this course (v4.2) is likely different from the one you previously used, and the GUI layout and the workflow has changed considerably since the introduction of v4.0. Moreover, the way we will use the software (through specification of the variational form) is probably new to you.

When using Comsol Multiphysics, as well as any other finite-element software, there are conceptually five consecutive steps that needs to be taken for each problem:

1. geometry definition,
2. meshing,
3. problem specification,
4. solution,
5. post processing and visualization.

2.1 Geometry definition.

In order to learn how to carry out parameterized solid geometry modeling, work through the printout *Creating a 2D Geometry Model* from the Comsol Multiphysics Modeling Guide. Once you have created the 2D geometry as described in the above document, save it to file, since you will need it for the steps below.

2.2 Meshing

In this course, we will only need to create simple so-called unstructured triangular meshes (which Comsol calls "Free triangular meshes"). In the Model Builder, click on **Mesh 1**, and the **Mesh** page appears in the Settings tab. On the **Mesh** page, select **User-controlled mesh** under **Sequence type**. Then click on the ▶ symbol to the left of **Mesh 1** in the Model Builder to open up more options under **Mesh 1**. Click on **Size**, which opens the **Size** page in the Settings tab. On the **Size** page, under **Calibrate for**, select **Fluid dynamics**. Then click again on the **Build All** button on the Settings tab in order to build the mesh, which then will be shown in the Graphics window. Play around with the options on the **Size** page (Maximum element size e.g.) to see what happens with the mesh. When you have changed any parameters, you need to press the **Build All** button again.

Remark 2. If you would like to learn more about meshing (or about any other issue in the following), select **Documentation** in the **Help** meny, which brings up a web page with the available documentation. For instance, there is a Meshing section in the Comsol Multiphysics User's Guide (it is Chapter 19 in the pdf version).

2.3 Problem specification

We will work with the following boundary-value problem for Laplace's equation:

$$\begin{aligned} -\Delta u &= 0 && \text{in } \Omega, \\ u &= u_D && \text{on } \Gamma_D \\ \frac{\partial u}{\partial n} &= 0 && \text{on } \Gamma_i \\ \frac{\partial u}{\partial n} + \alpha u &= \alpha u_a && \text{on } \Gamma_o \end{aligned} \tag{1}$$

on the domain Ω that you constructed in § 2.1 (figure 1).

Problem (1) constitute a simple model for steady (that is, thermal equilibrium has been reached) heat conduction in the metal object occupying domain Ω, where u is the temperature. We assume that boundary Γ_D is connected to a device that holds a fixed, given temperature u_D, that boundary Γ_i is thermally insulated, and that boundary Γ_o is facing air at a fixed, given temperature u_a. The value of α is related to the so-called heat transfer coefficient for the interface between the solid material and air.

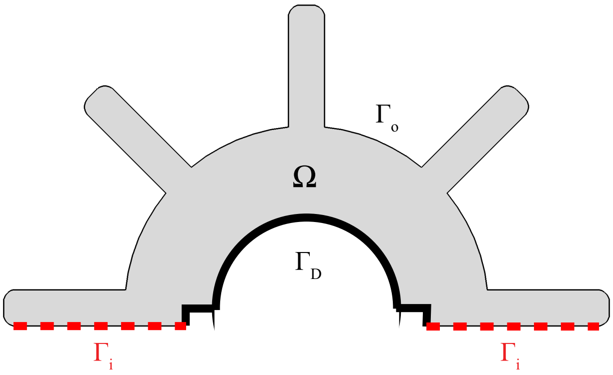

FIGURE 1. The domain associated with boundary value problem (1). The boundary of Ω comprises Γ_D (thick solid), Γ_i (thick dashed), and Γ_o (the rest, thin solid).

Task 1. By applying integration by parts, show that boundary-value problem (1) can be written in the following variational form:

$$\begin{aligned}&\text{Find } u \in V_{u_D} \text{ such that}\\ &\int_\Omega \nabla v \cdot \nabla u \, d\Omega + \alpha \int_{\Gamma_0} v u \, d\Gamma = \alpha \int_{\Gamma_0} v u_a \, d\Gamma \qquad \forall v \in V_0,\end{aligned} \tag{2}$$

where

$$\begin{aligned} V_{u_D} &= \left\{ v \mid \int_\Omega |\nabla v|^2 \, d\Omega \le \infty, v = u_D \text{ on } \Gamma_D \right\}, \\ V_0 &= \left\{ v \mid \int_\Omega |\nabla v|^2 \, d\Omega \le \infty, v = 0 \text{ on } \Gamma_D \right\}. \end{aligned} \tag{3}$$

When using the weak form in Comsol, the integrands in the variational form (2) are specified in symbolic form. More precisely, the expression inside the domain integral is written in one place, and the expression inside the boundary integral over Γ_0 is written in another place. For instance, the integrand inside the integral over Ω can be written in components as

$$\nabla v \cdot \nabla u = v_x u_x + v_y u_y, \tag{4}$$

where the subscripts denote differentiation. In Comsol Multiphysics, expression (4) is written

```
test(ux)*ux + test(uy)*uy                    (5)
```

where `test(u)` denotes a test function associated with the unknown u (which is denoted v in variational form (2)), and a trailing x or y denotes differentiation with respect to the coordinate x or y.

Implementation in Comsol Multiphysics

We assume that the geometry and a mesh has been created as specified in § 2.1 and 2.2. The steps below guides you through the remaning steps in order to numerically solve variational problem (2).

1. In the Model Builder (the leftmost tab on the Comsol GUI), right-click on **Model 1** and select **Add Physics**.

2. In the **Add Physics** page that appears under the **Model Wizard** tab, select **Mathematics** > **PDE Interfaces** > **Weak Form PDE** and click the ✚ sign under the **Add Physics** window. This action associates the default dependent variable name `u` with a variational (weak) form. You may change the name of the variable in the box labeled **Field** name. (You can also choose more than one field variable here. For the Maxwell equations, for instance, you would choose 3 variables in two space dimensions and 6 variables in three space dimensions). Then click the **Finish** button (marked with a black-and-white flag) on the **Add Physics** page, which will add the PDE leaf into the tree in the Model builder.

3. In order to define values for parameters α, u_D, and u_a, used in the boundary conditions, in the **Model Builder**, click on **Parameters** under **Global Definitions**. Enter values of the variables `alpha`, `uD`, and `ua`, in the table on the **Parameters** page in the Settings tab. We assume that the device attached to Γ_D holds a temperature of 100°, and that the room temperature is 20°. Moreover, we assume that the heat transfer coefficient is such that $\alpha = 1$. Thus,
$$u_D = 100, \quad u_a = 20, \quad \alpha = 1. \tag{6}$$

4. Click on the ▶ to the left of **PDE** in the Model Builder to open more selections, then select **Weak Form PDE 1**.

5. In the **Weak Form PDE** page that then appear in the **Settings** tab, replace the default expression in the field **weak** with expression (5). This action defines the contribution of the integrals over Ω to variational form (2).

6. Right-click on **PDE** in the **Model Builder** and select **More** > **Weak Contribution**. (There are two **More** appearing when you right-click; select the *second* one, the one below **Periodic conditions**).

7. Click on the **Weak Contribution 1** that will appear under **PDE** in the **Model Builder**. A page labeled **Weak Contribution** will appear in the **Settings** tab. On the **Weak Contribution** page, make sure that **Manual** is selected among the **Selection** options. Click on the boundary segments (there are quite a few) that correspond to boundary Γ_o and add them to the box labeled **Selection:** (can be done by clicking on the + sign e.g.). Enter the text `alpha*test(u)*(u-ua)` in the field labeled **Weak Contribution**. This action defines the contribution of the integrals over Γ_o to variational form (2).

8. Right-click on **PDE** in the **Model Builder** and select **Dirichlet Boundary Condition**. A page **Dirichlet Boundary Condition** will appear in the Settings tab. On the **Dirichlet Boundary Condition** page, make sure that **Manual** is selected among the **Selection** options. Click on the boundary segments that correspond to boundary Γ_D and add them to the box labeled **Selection:** (can be done by clicking on the + sign e.g.). Add `uD` in the box below the text **Prescribed value of** `u`.

9. Note that you do not need to do anything to set the boundary condition at Γ_i. Why?

10. Right-click the root object in the Model Builder (the root object has the same name as the file in which the model is saved) and choose **Add Study**. In the **Select Study Type** page that appears in the Model Wizard tab, select **Stationary**, and click on the **Finish** flag.

11. Right-click on **Study 1** in the **Model Builder**, select **Compute**, wait for the solution to be computed. (Alternatively, you can click on the green equal sign at the top of the window).

12. When the solution is computed, a color surface plot will appear in the graphics window, in which the colors visualizes the temperature distribution in the domain. Check so that the temperatures appear reasonable with respect to what can be expected from the boundary conditions on Γ_D and Γ_o.

13. Select **Results** > **2D Plot Group 1** > **Surface 1**. On the **Surface** page that appears in the Settings tab, you can see the default settings for surface plots.

Fourier's law is a constitutive relation for heat conduction that models the heat conduction in materials, such as metals, in which the heat conducting properties are *homogeneous* (same in each point) and *isotropic* (same in each direction). Fourier's law is analogous to Darcy's law for porous-media flows and states that the *heat flux vector* $\boldsymbol{q}$ is proportional to the negative gradient of the temperature,

$$\boldsymbol{q} = -\kappa \nabla u, \tag{7}$$

where $\kappa > 0$ is the material's *thermoconductivity*. Expression (7) tells us that the heat energy will flow in the same direction as negative temperature gradients, that is, heat flows from high temperature to low temperature.

Task 2.

1. Here we will visualize how the heat flows in Ω. Right-click **Results** > **2D Plot Group 1** in the Model Builder and select **Arrow Surface**.
2. In the **Arrow Surface** page that appears on the Settings tab, replace the default expressions in the fields labeled **x component** and **y component** with the ones obtained from expression (7). (You may assume that $\kappa = 1$). After entering the expressions, click on the **Plot** button (it is marked with a small pen on the top of the Arrow Surface page). The small arrows that then show up in the graphics window indicates the direction and magnitude of the heat flux vector. Play around with the options for arrow plots, for instance the number of points for the vectors and the color and size. Make sure that the plot you obtain seems reasonable in terms of the direction you expect the heat to flow.
3. Another way of visualizing the heat flux field is through *streamlines*. The streamlines associated with the $\boldsymbol{q}$ field are curves through the domain that are such that their tangent at each point coincides with the $\boldsymbol{q}$ vectors and such that the heat flow is always constant between two consecutive curves. The latter property means that the heat flow is more intense in regions such that the streamlines lie close together, compared to regions where the distance between consecutive streamlines is larger.

   Plot streamlines of the $\boldsymbol{q}$ field by right-clicking on **Results** > **2D Plot Group 1** in the Model Builder and selecting **Streamline**. In the **Streamline** page that appears in the Settings tab, replace the default expressions in fields labeled **x component** and **y component** with the ones obtained from expression (7). (Again, you may assume that $\kappa = 1$). When plotting streamlines, you need to specify the boundaries through which the streamlines should pass. In this case, heat will flow through all boundaries except the insulated Γ_i boundary. Press the symbol ⏻ to the right of Selection on the Streamline page in order to activate the choice of boundary segments for streamline intersections. Then, to the **Selection** box, add all boundary segments except those that constitute Γ_i. (In this case, it is possible to choose all boundaries, including Γ_i, without seeing to much of a difference. In general, however, it can happen that the streamlines become clearer if the insulated boundaries are deleted). Then click on the **Plot** button in order to see the streamlines. Also here play around with the options, such as the number of streamlines. (You may want to increase the number from the default value). It can be easier to see the streamlines by disabling the arrow plots: right-click on **Results** > **2D Plot Group 1** > **Arrow Surface 1** and choose **Disable**.
4. Compute solutions for $\alpha = 0, 0.2, 0.5, 1, 100$. Observe how the solution depends on α. What happens for a very large value of α?