Rashed Almasri NE533 03/19/2024

Project Part 1

The Geometry & Variable

To design this geometry, we need to write three different inputs into the mesh block. Firstly, a general mesh to cover up the entire geometry, titled 'total', with dimension set at 2, the type of this mesh is GeneratedMeshGenerator for standard rectangular mesh. Then the two extra meshes are subdomains with type SubdomainBoundingBoxGenerator, where we identify the fuel block and the gap block. Here, the cladding block will be automatically identified by default.

If we run the code with the command –mesh-only added to the end of the command line, we can generate an output file for the mesh to make sure this part of the code is functioning well, and the geometry is well defined. The geometry can be seen in Figure 1.

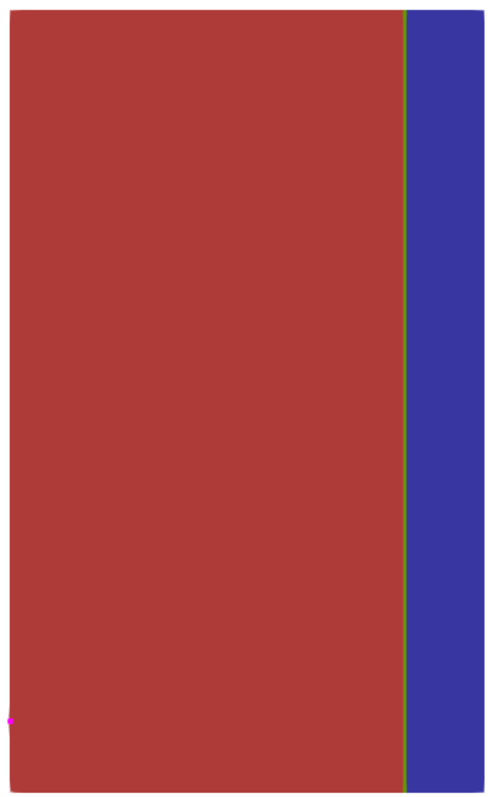

Figure 1: Geometry composition within the input file.

For the variables, we only introduce one, which is the temperature, with a first-order equation that will behave just like a diffusion variable in our case.

Functions & Kernels

In the steady state, the function type will be ConstantFunction. We have a value for LHR, but we need to input Q here in our code, so by simply dividing LHR by pi and the fuel radius squared, we get Q at 445.6338 w/cm.K.

For the kernels, I have added four different kernels, one for generated heat from the fuel, but three for the conduction of the three materials, even though conduction values were also defined later in the materials block which perhaps show two different methods of inputting the conduction in our case.

For the kernel blocks, we need to define the specific material or block this kernel takes effect. So, for the generated heat, it is only in the fuel, but for the conduction, it is defined separately at each block.

Boundary Conditions and Materials

Two boundary conditions will be defined. First, the temperature of the coolant, which is 550 K, and will be the temperature of the surface of the cladding. This one is straightforward, and has a type of ADDirichletBC, which reflects a constant value, g = h.

The second one is the in the center of the fuel, or on the left edge of the geometry. We know here that the temperature will be at a maximum, which means the derivative of the temperature will be zero. This boundary condition type is ADNeumannBC, which is the derivative of the variable at this point, which is set to zero.

Now to the materials, we define each block, name them, and set the type to HeatConductionMaterial, as we are only concerned with their heat conductance for the moment.

Problem, Executioner and Outputs

The problem will be FEProblem, which is the default Finite Element problem. The executioner type is set on Steady for calculating the values in a steady state. And The output is set on exodus, which is the output file we will obtain.

Output File and Comparison

First let us check the analytical solution for our problem.

LHR = 350 W/cm^2

R-fuel = 0.5 cm

t-gap = 0.005 cm

t-clad = 0.1 cm

kf = 0.03 W/cm

kg = 0.0026 W/cm

kc = 0.17 W/cm

Tcl = 550 K

$$\frac{LHR}{2\pi R_F} = \frac{350}{2\pi 0.5} = 111.41$$

$$T_{Co} - T_{ci} = 111.41 \times \frac{0.1}{0.17} = 65.534$$

$$T_{ci} = 615.534\ K$$

$$T_{Ci} - T_{fo} = 111.41 \times \frac{0.005}{0.0026} = 214.25$$

$$T_{fo} = 829.784\ K$$

$$T_{fo} - T_s = \frac{LHR}{4\pi K_f} = \frac{350}{4\pi \times 0.03} = 928.4$$

$$\boldsymbol{T_s = 1758.19\ K}$$

Now, for our output file, we can see the results in Figure 2. But to find the centerline temperature, we just find the maximum value within Paraview.

The centerline temperature from the code is: **1747.61 K.**

Calculating the variance between the two values, we have only a 0.6% difference between the two values, indicating that our code is pretty accurate.

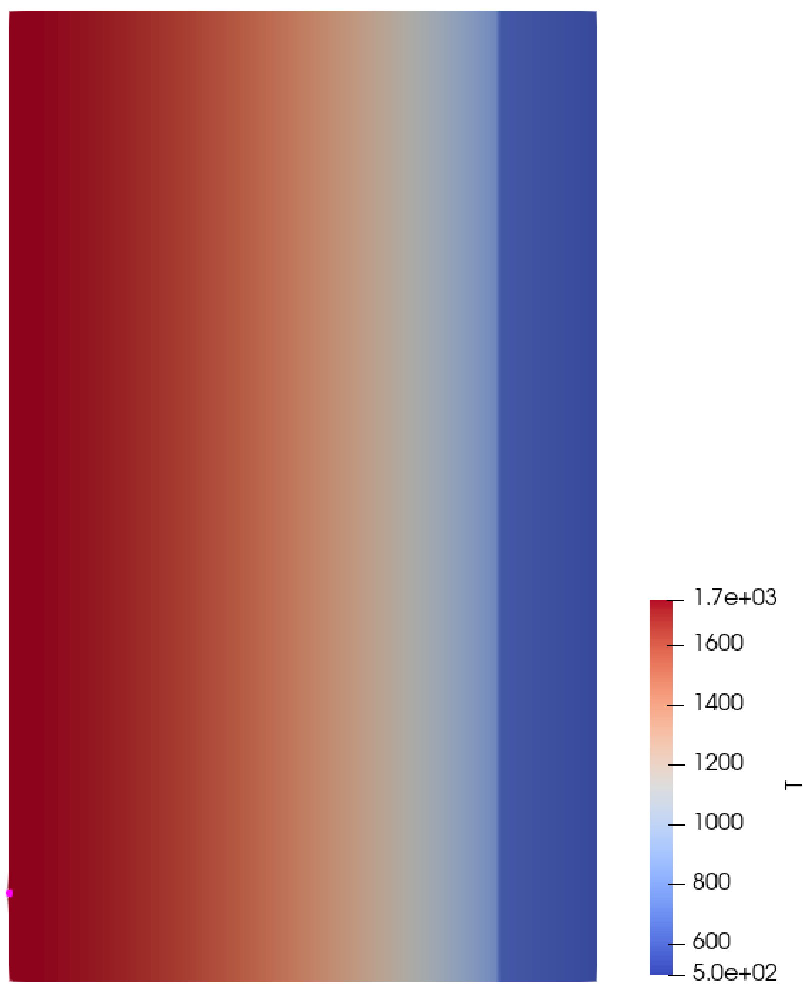

Figure 2: Temperature profile output file.