**MOOSE Project Parts 1, 2, & 3**

Part 1

In this first part of the project, we were given the dimensions of a fuel system (pellet, gap, cladding) and asked to solve for the temperature profile. A diagram of the system is illustrated below.

A diagram of a fuel gap and cladding

AI-generated content may be incorrect.

There were 5 ways with which to solve this problem:

1. Analytical steady-state solution, constant LHR, constant material properties
2. FEM steady-state solution, constant LHR, constant material properties
3. FEM transient solution, time-dependent LHR, constant material properties
4. FEM steady-state solution, constant LHR, temperature-dependent material properties
5. FEM transient solution, time-dependent LHR, temperature-dependent LHR

All FEM solutions were performed using the MOOSE program. Input and output files for all FEM solutions are attached in this submission.

Guidance for setting up the problem was as follows:

* Assume reasonable values for material properties
* Outer cladding constant temperature: 550 K
* Constant Linear Heat Rate (LHR): 350 W/cm
* Time-dependent LHR: 350\*EXP(-(t-20)^2/2) + 350 W/cm
* Determine transient solution for up to t = 100

Constant material properties were found through various resources and are given below.

|  |  |  |  |
| --- | --- | --- | --- |
| Material | Thermal Conductivity, k  (W/cm\*K) | Specific Heat, Cp  (J/g\*K) | Density (g/cm3) |
| UO2 (Fuel) | 0.03 | 0.33 | 10.97 |
| He gas (Gap) | 0.00152 | 5.193 | 0.1785 |
| Zr (cladding) | 0.23 | 0.35 | 6.511 |

Properties for the fuel and cladding were found in NE 533 Lecture 3. Properties for the gap were found at [Helium - Thermal Conductivity](https://www.periodic-table.org/Helium-thermal-conductivity/).

The only equations used to solve for temperature-dependent material properties are those given for the thermal conductivity of the fuel pellet, and the thermal conductivity of the gap (assuming the gap remains pure He gas) and are given in NE 533 Lecture 8 and Lecture 3, respectively. Those equations are presented for both below in W/cm\*K.

Where t = T / 1000, and

**Solution #1- Analytical**

The following equations were used to solve the steady-state temperature profile of the fuel system from the outside of the cladding to the fuel centerline. Subscripts denote the type of material (fuel, gap, cladding), and the radius/thickness of the materials are given in the diagram. The temperatures at each boundary were calculated, and the profile from the edge to the centerline of the pellet was modeled.

|  |  |  |
| --- | --- | --- |
| Boundary | Location (cm) | Temperature (K) |
| Outer Cladding | 0.605 | 550 |
| Inner Cladding | 0.505 | 615.5 |
| Surface of Fuel | 0.5 | 983.7 |
| Centerline of Fuel | 0 | 1912.1 |

**Solution #2- FEM, steady-state, constant properties**

There are six parts to each MOOSE input file:

* Mesh
* Variables
* Kernels
* BCs
* Executioner
* Outputs

All six of these parts will be described in this section. Since some of these parts will be repeated for the input files of other solutions, it is not necessary to cover them multiple times. This solution most closely correlates to the analytical solution, with a peak centerline temperature of about 1900K.

**Mesh**

Individual meshes were generated using *GeneratedMeshGenerator* that had the dimensions of each block outlined. Each block was assigned a subdomain ID using *SubDomainIDGenerator.* The blocks were stitched together at the interfaces using *StitchedMeshGenerator*. Meshes were split into 150x1, 10x1, and 100x1; the size of the meshes was varied to determine proper size for reasonable resolution.The coordinate type was changed from the default x-y system to the axisymmetric r-z system, with symmetry around the default y-axis.

**A screen shot of a computer program

AI-generated content may be incorrect.**

**Variables**

The variable was set as temperature, and an initial condition of 550 K was set because it made sense.

**Kernels**

The function *ADHeatConduction* was used to model the heat conduction of this closed system using automatic differentiation. The function *HeatSource* was required to add the volumetric heat generation rate, given to us in the problem as LHR. The LHR given was divided by the cross-sectional area of the pellet to give the value seeked.

**BCs**

There were two boundary conditions to be used in this problem. The centerline temperature of the fuel is at a peak, thus the derivative of temperature here is 0. This is reflected in the function *NeumannBC.* The outside of the cladding is held at 550 K, and is modeled using *DirichletBC.*

**Materials**

Three subsets of materials were chosen and their properties (i.e. thermal conductivity) were defined using *ADGenericConstantMaterial* as is done with most materials with constant properties.

**Executioner**

Since this was a steady-state solution, the function for the executioner was chosen as *Steady* with a Newton solve type as is common with other steady-state solutions in MOOSE. The following “petsc” lines utilize the Hypre-BoomerAMG preconditioners and solvers, and have been shown to help the solution converge.

A screen shot of a computer code

AI-generated content may be incorrect.

**Outputs**

The outputs to this input file were an exodus file that could simulate the solution in a program like Paraview, and a csv file that utilized the post processor built in. The Vector Post Processor *LineValueSampler* takes the data (temperature) of a line from one point to another (fuel centerline to cladding in middle of system). The 1D steady-state temperature profile of the fuel system for the analytical solution, the constant properties solution, and the temperature dependent properties solution can be seen below.

**Solution #3- FEM, transient, constant properties**

The parts of this solution identical to Solution #2 are: Variables, BCs, and Outputs. The max peak centerline temperature is a bit higher, almost 2100 K.

**Mesh**

The mesh was changed to be much rougher to make the computations easier (50x1, 5x1, 10x1).

**Kernels**

A kernel was added to include the time-dependence of heat conduction, *ADHeatConductionTimeDerivative.* The time-dependent formula for LHR in *HeatSource* was written out for the pellet.

**Materials**

Materials were still defined using *ADGenericConstantMaterial*, but additional properties (specific heat, density) were required to run the time derivative.

**Executioner**

The type for this problem was changed from Steady to Transient. The solve type used was PJFNK. The time step dt was chosen as 1 to get individual time steps. Therefore, from 0 to 100 is 100 steps. The non-linear relative tolerance and the non-linear absolute tolerance were both set at 1E-10. The linear tolerance was set at 1E-5, and a steady-state detection function was added in.

The fuel centerline temperature over time was plotted in Paraview and can be seen below.

A graph with a green line

AI-generated content may be incorrect.

**Solution #4- FEM, steady-state, changing properties**

The Mesh, Variables, BCs, and Outputs are identical to Solution #2. However, the max peak centerline temperature for this solution was only around 1760 K as reflected in the first plot. This is due to the changing value of thermal conductivity with temperature of the material.

**Kernels**

The only change is that the heat conduction function was changed from *ADHeatConduction* to just *HeatConduction*, since no AD materials were used.

**Materials**

The materials were changed from *ADConstantGenericMaterial* *to ParsedMaterial* for the fuel and gap, and *GenericConstantMaterial* for the cladding. *ParsedMaterial* was used for materials whose thermal conductivity changed with temperature, and the cladding was changed to reflect a non-AD heat conduction kernel.

**Executioner**

Instead of a steady-state solver, a transient solver was used with a time step of 1. The tolerances given in Solution #3 were also used here.

**Solution #5- FEM, transient, changing properties**

The Mesh has the same size elements as the other transient solution. The Variables, BCs, and Outputs are similar to Solution #2. The Executioner is the same as Solution #3.

**Kernels**

The same functions were used as in Solution #3 except they were converted from AD-functions to their non-AD form.

**Materials**

*ParsedMaterial* and *GenericConstantMaterial* were used as in Solution #4. However, both functions would have to be used for the fuel and the gap since they had both constant properties and changing properties that were required for the time derivative function.

The fuel centerline temperature over time was plotted in Paraview and can be seen below.

A graph with a green line

AI-generated content may be incorrect.

**Part 2**

In the second part of the project, the dimensions of the fuel system were changed. The system is now 1 m (100 cm) tall. The materials have material properties dependent on the temperature. The LHR changes as a function of axial elevation, and the outer cladding temperature changes with the two relationships below:

Values for LHR0 and Tincool are 350 W/cm and 500 K, respectively.

**Part 2 Solution- Steady State, k(T), LHR(z), Tcool(z)**

**Mesh**

The Mesh was stitched just like the other solutions. A mesh convergence study was done to determine the proper amount of finite elements to get a converging solution with reasonable resolution. The mesh sizes are 100x200, 10x200, and 50x200.

**Variables**

The initial condition was set for 500 K.

**Kernels**

Since this is a steady-state problem, only the heat conduction and heat source kernels are used. However, the heat source function is changed to be dependent on the axial elevation as above, in this case that is the y-coordinate.

**BCs**

The BCs are the same as in previous solutions except the Dirichlet BC value for the outer cladding has changed to a function that is dependent on axial LHR and axial elevation. The mass flow rate of 0.25 kg/s and heat capacity of water of 4200 J/kg\*K are built in to the function.

**Materials**

Ther materials are built the same way as they are in Solution #4.

**Executioner**

The same executioner is used as is in Solution #4.

**Outputs**

The outputs to this are an exodus file for simulation, and a csv file with a post processor built in to generate axial temperature profiles for the fuel centerline, fuel surface, and inner cladding as seen below.

**Part 3**

This part has parameters similar to Part 1. This includes a constant LHR of 350 W/cm, an outer cladding temperature of 500 K, and similar specific heat capacities and densities of all the materials. The geometry is also the same in this part.

The difference is that this is a thermo-mechanical simulation where temperature, stress, and strain are coupled. The thermal conductivity of the fuel pellet is dependent on temperature and burnup and is given in the following equations:

Burnup is a function of the variable time, and is given as:

The materials in this system undergo volumetric change. The pellet undergoes multiple effects that change its volume: thermal expansion, densification, solid fission product swelling, and gaseous fission product swelling. The strain formulas for these phenomena, excluding thermal expansion, are:

The thermal expansion of materials is modeled using a specific module. However, the three equations above were combined to equal the volumetric change in the pellet that is not due to thermal expansion. This volumetric change is also a function of temperature and burnup.

Mechanical properties of the materials can be seen below. While the cladding also undergoes deformation and expansion, the gap has no solid mechanical properties.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Material | Thermal Expansion Coefficient | Youngs Modulus | Poisson’s Ratio | Source |
| Fuel: UO2 | 1.2E-5 (1/K) | 192.9E+9 (Pa) | 0.302 | [NIST](https://srdata.nist.gov/CeramicDataPortal/Elasticity/UO2) |
| Gap: He | 0 (1/K) | 0 (Pa) | 0 | N/A |
| Clad: Zr-4 | 6E-6 (1/K) | 99.3E+9 (Pa) | 0.37 | [Azom](https://www.azom.com/article.aspx?ArticleID=7644) |

**Part 3 Solution- Transient, Thermo-Mechanical, Temperature and Burnup Dependent**

**Mesh**

The mesh was stitched just like the solutions in Part 1. The pellet, gap, and cladding mesh sizes were selected as 35x5, 5x5, and 10x5 respectively. Many finer meshes for these materials were chosen, but none of the solutions utilizing them converged. The rougher meshes than these also had trouble converging, and the ones that did had huge discrepancies in the simulation.

**Variables**

The variable temperature was set with an initial condition of 500 K.

**Kernels/Modules/Functions**

The kernels *HeatConduction, HeatConductionTimeDerivative, and HeatSource* were used.

A block was also created to all relevant solid mechanics modules. We wanted this command to add the solid mechanics variables we need automatically, set strain as finite, and utilize displacements in the r- and z- directions for calculations. The displacements were set at the top of the script as *GlobalParams.* Automatic differentiation was turned off to mitigate errors. Eigenstrain names were defined for thermal expansion and volumetric change due to densification, solid fission products, and gaseous fission products. An output of this module that was selected was “von Mises stress.” The temperature was also taken into this block.

A function was defined for burnup, which is given as a variable of time as previously discussed.

**BCs**

Boundary conditions for temperature were given similar to previous solutions from Part 1. Conditions were also set to restrict r-direction displacement in the middle of the pellet, as there is more solid material arresting displacement here than the outer edge of the pellet. Restriction of z-direction displacement was also set at the top and bottom of the pellet, as pellets are stacked and likely to arrest expansion.

**Materials**

Thermal properties of the gap and cladding were set identical to the Part 1 transient, temperature dependent solution. For the pellet, the thermal conductivity was set as a parsed material. This material utilized the expression for pellet conductivity with respect to temperature and our defined function burnup.

*ComputeIsotropicElasticityTensor* was used to define elasticity for the materials, taking in the elastic modulus and Poisson’s ratio. *ComputeFiniteStrainElasticStress* computed the stress and strain of the materials, without needing any inputs except the block. *ComputeThermalExpansionEigenstrain* was needed to compute just that. This eigenstrain would be the same as the thermal expansion eigenstrain we defined before in the solid mechanics block. The thermal expansion coefficients for materials were defined here per block.

For volumetric change in the pellet due to our strain phenomena, *ComputeVolumetricEigenstrain* was defined for each material, but only the pellet had useful inputs. The output went to the volumetric eigenstrain we defined in the solid mechanics block, and would add to thermal expansion. This module needed to call a volumetric material, so we defined in the next block a *ParsedMaterial* with the expression for pellet volumetric expansion being the sum of the three strain equations. This expression also was dependent on temperature and burnup.

**Executioner**

The transient type was used here, with a goal of simulating an hour (3600 seconds) in minute intervals (60 seconds). Automatic scaling was enabled for this.

Tolerances were adjusted to find the right combination for a complete convergence. The tolerances that finally worked for these meshes are below.

* Non-linear relative tolerance: 1E-6
* Non-linear absolute tolerance: 1E-6
* Linear tolerance: 1E-4
* Linear maximum iterations: 50

Preconditioners were also adjusted to get the solution to solve. These preconditioners worked as follows:

* -pc\_type: hypre
* -pc\_hypre\_type: boomeramg
* -pc\_hypre\_type\_boomeramg\_strong\_threshold: 0.5
* -mat\_mffd\_err: 1E-6

Steady-state detection was also turned on with a tolerance of 1E-8.

*IterativeAdaptiveDT* was also used to help get solutions to converge. The optimal iterations set were 1, the linear iteration ratio was 100 (default of 25), and dt was set to reset after a failed convergence.

**Outputs**

The output was an Exodus file. The code reached a steady-state solution of up to 50 seconds, utilizing 389 time steps. There was not enough time to go by for the gap to close.

Plots were created to analyze the von Mises stress, displacement-r, temperature, and displacement-z with time. These plots followed a straight line across the centerline of the system (y = 0.5).

The maximum von Mises stress occurred at the final time step in the centerline of the fuel at 2.1 GPa, but dropped to 0 at the outer pellet boundary. This is incorrect as the maximum stress in the fuel pellet should be the hoop stress at the surface.

The maximum r-direction displacement occurred at the final time step at the outer pellet boundary, with a value just a little greater than 0.005cm. Due to these differing results, it is unclear whether the pellet expanded enough to make contact with the cladding. There was no variation in z-direction displacement throughout space or time.

Based on the BCs selected and the results given, it is believed that the von Mises stress exceeds the fracture stress of 130 MPa (from NE533 lecture). However, the von Mises stress is incorrect so this cannot be proven at this time.

The input files and output files of all parts and solutions have been uploaded. There is not sufficient space to provide graphs on this final page, but better plots can be generated in Paraview that change with time.

I wish I had gotten a proper solution to converge, but I have run out of time and must submit something. Thank you Professor Beeler for your flexibility this semester, I found your class very interesting.