NE 533 MOOSE Project: Part 1

Alexandra Doherty
02.28.2025

Abstract

Understanding the thermal behavior of fuel rods in a nuclear reactor environment is imperative for the knowledge of how to safely operate nuclear power plants. In order to tabulate the temperature profiles of fuel rods, we first simulate a single fuel pellet behavior with a steady-state linear heat rate and a transient linear heat rate, with both constant, and temperature-dependent thermal conductivity values in the fuel, gap, and cladding using INL's MOOSE Framework[3].

Introduction

The goal of this report is to outline centerline temperature/temperature profiles of fuel pellets using the MOOSE Framework. The prompt of these simulations is based on the fuel pellet shown in the figure below. The height of the pellet is 1.0 cm, the R_f is 0.5 cm, the R_g is 0.005 cm, and the R_c is 0.1 cm.

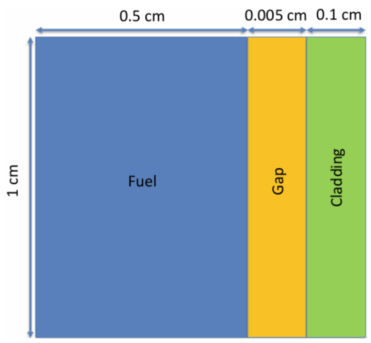


Figure 1: This figure shows the dimensions of the fuel pellet outlined in the proposal of this simulation design.

Four separate conditions were to be simulated:

- Steady-state linear heat rate with a constant thermal conductivity
- Steady-state linear heat rate with a temperature-dependent thermal conductivity
- Transient linear heat rate with a constant thermal conductivity
- Transient linear heat rate with a temperature-dependent thermal conductivity

The methods for simulating each condition are outlined throughout this report, comparisons are included, equations are displayed, and the full MOOSE codes are listed in the appendix.

A UO_2 fuel pellet was used for these simulations due to being a traditional form of nuclear fuel (along with UN). The gap was assumed to be entirely Helium, however it is possible for there to be Xenon in the gap as well, just not for this application. The cladding used was Zirconium, another option being stainless steel, but Zirconium is another material traditionally used. The material properties of each material are listed below.

	Thermal Conductivity	Specific Heat	Density
Fuel (UO2)	0.03 W/(cm*K)	0.33 J/(g*K)	10.98 g/cm^3
Gap (He)	0.153e-2 W/(cm*K)	5.1932 J/(g*K)	0.1786e-3 g/cm^3
Clad (Zr)	0.17 W/(cm*K)	0.35 J/(g*K)	6.5 g/cm^3

Figure 2: [1][4]This figure shows the material properties used in the MOOSE programs for the fuel, gap, and cladding.

Methodology

Analytical Solution

The analytical solution was calculated using different simplifications of the overall heat conduction equation to form equations for the temperature in the fuel, gap, and cladding. The steady-state LHR with constant thermal conductivity was the only solution able to be tabulated analytically, which is what was used to determine the mesh in each program moving forward. The analytical solution is graphed using Excel, with commands determining which equation is applied to which section. The equations are outlined in the Equations portion of this report.

General Properties for Defining Programs

Some properties of these separate scenarios are applicable across both steady-state and transient functions regardless of thermal conductivity temperature-dependence. Those properties consist of the mesh, preconditioning, variables, and boundary conditions.

Mesh Determination

The xmin and ymin are determined by the coordinate (0,0) which is the bottom corner of the fuel rod pellet, and the xmax and ymax are the outer dimensions of the top right corner of the fuel rod pellet cross section, containing the radius of the fuel pellet (0.5 cm), gap thickness (0.05 cm) and cladding thickness (0.1 cm). "dim" represents 2-dimensional, and coord-type 'RZ' denotes the geometric orientation of the problem.

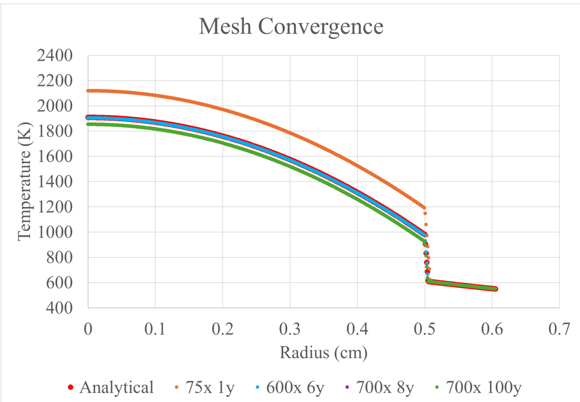


Figure 3: This figure shows the results of the Mesh Convergence test compared to the analytical results.

Subdomain1 creates a box around the fuel and the gap, and subdomain2 creates a box around the fuel itself. This forms three boxes; an outer box from the outer part of subdomain1 to the xmax and ymax representing the cladding, a central box between subdomain1's outer limits and subdomain2's outer limits representing the gap, and a left-oriented box between the origin and subdomain2's outer limits representing the fuel. Block ID "2" is associated with the fuel component, Block ID "1" with the Gap, and Block ID "0" with the Cladding.

Mesh nx and ny were determined using a mesh convergence analysis comprised of a series of tests comparing results of the temperature profile of the steady-state LHR with constant thermal conductivity to its corresponding analytical solution. The parameters contributing to creating the temperature profile of the static LHR with constant thermal conductivity component are outlined further along in this report. Figure 2 displays the effect of mesh size on quality of analysis. Using a coarser mesh such as nx=75 and ny=1, the temperature profile reported slightly higher values than the analytical solution, and a finer mesh such as nx=700 and ny=100 reported slightly lower values than the analytical solution and had an extended run time that was inefficient. The mesh corresponding most closely with the analytical solution was nx=600 and ny=6.

Preconditioning

The preconditioning system in MOOSE allows a user to define the type of preconditioning matrix to build (type of system of equations to apply). The Preconditioning system chosen for this application is a solve type of "NEWTON". Newton's method applies a full Jacobian to the solve to the system and is allows for greater convergence especially in nonlinear situations. It is easier for smaller applications due to the greater stored memory compared to "PJFNK" which was the solve type this program was originally attempted with.

Variables

The only variable defined throughout these programs is temperature, denoted as "T". It is a first order variable with an initial condition of 550 K, which is the outer cladding temperature provided.

Boundary Conditions

For this application, the Dirichlet and Neumann boundary conditions are used. Dirichlet BC indicates that the value of the Temperature (T) is fixed, and Neumann BC indicates that the derivative of the Temperature (T) is 0. That shows that there is no heat flux across the boundary, so that the outer cladding temperature will remain 550K (given value) throughout the calculation process.

The boundary conditions effectively fix the temperature profile into a controlled region to evaluate. It is also possible to include top and bottom boundary conditions, however it was not necessary for this stage of the process. Top and bottom boundary conditions would fix the temperature on all sides, however we are focused on axial thermal gradients.

Functions

The "Parsed Function" command was used to calculate the volumetric heat rate (VHR) from the linear heat rate (LHR) in the steady-state program. This was necessary because MOOSE operates using the finite element method (FEM) which operates using volumetric quantities. For the steady-state application, the VHR was constant, and could have been declared also using a "Constant" expression, alternatively the "Parsed Function" was used for ease of replicating for the transient application, which is time dependent. "ParsedFunction" is beneficial when solving expressions as a function of time or linear coordinate. The conversions from LHR to VHR for steady state and transient LHR values are displayed in the "Equations" section of this report, which is the LHR divided by the cross sectional area of the fuel pellet. The transient LHR equation was provided in the context of the proposed project.

Steady-State Linear Heat Rate

Kernels

Kernels are used to solve the heat conduction equation more simply. Each kernel is used to solve pieces o fthe residual equation in the form of a partial derivative. In the steady-state case, two kernels are used; "HeatSource" and "ADHeatConduction". In the heat source segment, block 2 is identified as the heat source, which corresponds with the fuel pellet. The variable T is then associated with the heat source as a function changing by means of LHR. In the steady state, the LHR is constant, the highest temperature exists at the fuel centerline (radius = 0) and the lowest temperature is on the outer cladding (radius = max). "ADHeatConduction" is used opposed to "HeatConduction" due to the automatic differentiation (AD) that is able to compute derivatives of each temperature dependent property per iteration. The heat conduction variable is used to incorporate the diffusion term of the heat equation, once again denoting temperature as the associated variable. The heat conduction kernel is useful for steady-state and transient heat conduction as well as temperature-dependent thermal conductivity.

Constant Thermal Conductivity

Materials

Three separate materials were used for these programs, the fuel (UO_2), Gap (He), and Cladding (Zr). In the steady-state process with a constant thermal conductivity, the materials were more simple, only requiring the constant thermal conductivity to be declared (each value for thermal conductivity is an accepted value). To differentiate which block is which, there is a block declaration. The type of material for each is an "ADHeatConductionMaterial" because the thermal conductivity is being considered and contributes to a function to determine temperature at each point in the fuel pellet.

Postprocessor

A VectorPostprocessor is used for both steady-state programs. It is set up to take 500 points between declared start/end points, which are along the centerline of the fuel. Each point is paired with the value of the temperature variable. The "sortby" option is used to order the values from least to greatest. This is consistent for both the constant and variable thermal conductivity programs.

Executioner

Executioner type should be steady for steady-state programs. The difference between the constant and variable thermal conductivities is that the variable thermal conductivity program requires additional parameters to help it converge. Nonlinear and linear relative/absolute tolerances are determined through trial and error to find the maximum decimal places the function can converge to in a reasonable duration of time with adequate accuracy.

Outputs

Exodus is the function used in outputs that stores simulation results, and should be marked as true. Then to tabulate data and manipulate for further analysis, it can be exported to a CSV file and named, all is typed in the output block.

Variable Thermal Conductivity

Materials

The fuel, gap, and cladding are declared as ADHeatConductionMaterials consistent with each iteration of programs. The difference is that thermal conductivity is in terms of a function. The functions for thermal conductivity are shown in the ”Equations” portion of this report. The nomenclature for the function uses ”t”, although ”t” is indicative of temperature rather than time, this is because the operator must use functions of t or linear coordinates, and through trial and error, ”t” was the most functional operating variable. A minimum temperature of 550 Kelvin is also noted so that the function can converge easily.

Transient Linear Heat Rate

Kernels

The ”HeatSource” and ”ADHeatConduction” kernels are the same as the steady-state LHR program, the difference for Transient being the addition of the ”ADHeatConductionTimeDerivative” kernel. The time derivative serves to make adjustments to each of the parameters declared, in this case temperature, over each time step. This allows for the formation of a temperature profile over a series of timesteps.

Postprocessor

The postprocessor used for the transient LHR uses the ”PointValue” function to measure the simulated temperature at each timestep to count as a data point each iteration.

Executioner

The transient LHR executioner is the same as steady-state for the constant thermal conductivity section. For the variable thermal conductivity, the ”TimeStepper” function block type ”IterationAdaptiveDT” was added to further optimize the block. This starts with a smaller time step and after the optimal iterations can adjust to speed up or slow down the simulation time.

Outputs

The only difference between the steady-state and transient programs output file is the steady-state executes on the final iteration, where the transient executes when it converges after each timestep.

Constant Thermal Conductivity

Materials

Transient LHR with constant thermal conductivity uses a similar Materials block as the steady-state LHR with constant thermal conductivity, except it requires the declaration of thermal conductivity, specific heat, and density. Each were acquired through accepted values. For these blocks, an ADGenericConstantMaterial function was used. This is useful for declaring material properties that do not have temperature dependence, so it could have been used for the steady-state LHR with constant thermal conductivity as well. This function will not be effective for the programs with a temperature dependent thermal conductivity, and more creative solutions are required to declare material properties when some are temperature dependent and others are not.

Variable Thermal Conductivity

Materials

For the transient LHR with a variable thermal conductivity program, ADHeatConductionMaterial functions are used to declare the thermal conductivity temperature and specific heat functions/values. This is because they are considered temperature dependent in MOOSE, whereas densities are treated as constant or a predefined function in MOOSE, and are included in ADGenericConstantMaterial function blocks.

Equations

Analytical

$$T_F(r) = \frac{Q_{\text{avg}}(R_f^2 - r^2)}{4k_f} + T_{F0}$$

$$T_G(r) = T_{CI} - \ln\left(\frac{r}{R_g}\right)\frac{LHR}{2\pi k_g}$$

$$T_C(r) = T_{CO} - \ln\left(\frac{r}{R_c}\right)\frac{LHR}{2\pi k_c}$$

$$Q_{\text{avg}} = \frac{LHR}{\pi R_f^2}$$

Figure 4: $T_F(r)$ is the fuel temperature, $T_G(r)$ is the gap temperature, $T_C(r)$ is the cladding temperature, T_{F0} is the fuel centerline temperature, T_{CI} is the inner cladding temperature T_{CO} is the outer cladding temperature, k_f is the fuel thermal conductivity, k_g is the gap thermal conductivity, k_c is the cladding thermal conductivity, R_f is the fuel pellet radius, R_g is the gap radius, R_c is the cladding thickness, Q_{ave} is the average VHR, LHR is the linear heat generation, r is the radial position.

Volumetric Heat Rate

$$VHR_{\text{linear}} = \frac{LHR}{\pi \times R_f^2}$$

$$VHR_{\text{transient}} = \frac{LHR \times \exp\left(-\frac{(t-20)^2}{2}\right) + LHR}{\pi \times (R_f)^2}$$

Figure 5: Conversions from linear heat rate (LHR) to volumetric heat rate (VHR) to adjust the value to be more representative of heat transfer through the bulk. R_f represents the fuel pellet radius and t is time.

Thermal Conductivity

Fuel (UO_2)

$$k(T) = \frac{1}{100}\left(\frac{100}{7.5408 + 17.629\left(\frac{T}{1000}\right) + 3.6142\left(\frac{T}{1000}\right)^2} + \frac{6400}{\left(\frac{T}{1000}\right)^{5/2}}\exp\left(\frac{-16.35}{T/1000}\right)\right)$$

Gap (Helium)

$$k(T) = 16 \times 10^{-6} \times T^{0.79}$$

Cladding (Zirconium)

$$k(T) = \frac{8.8527 + 7.0820 \times 10^{-3}T + 2.5329 \times 10^{-6}T^2 + \frac{2.9918 \times 10^3}{T}}{100}$$

Figure 6: [2].Equations depicting how temperature helps to describe thermal conductivity for conditions accounting for a variable k. T is Temperature, and k is units of W/cm*K

Heat Conduction

$$\textbf{Steady-state:} \quad \nabla \cdot (k\nabla T) = Q$$

$$\textbf{Transient:} \quad \rho C_p \frac{\partial T}{\partial t} = \nabla \cdot (k\nabla T) + Q$$

Figure 7: This figure shows the heat conduction equations for steady-state and transient LHR. ρC_p is the thermal inertia, where ρ is the density, $\frac{\partial T}{\partial t}$ is the rate of temperature change over time, $\nabla \cdot (k\nabla T)$ is the heat flux divergence, and Q is the volumetric heat source.

Results

Steady-State Linear Heat Rate

The figure below shows how the temperature profile of a steady-state LHR presents itself. A maximum temperature can be observed in the centerline of the material (radius = 0 cm), which makes sense when comparing to literature expectations, as well as the analytical solution. The constant thermal conductivity curve yielded a slightly higher centerline temperature of 1900 K, whereas the variable thermal conductivity curve yielded a centerline temperature of 1750 K.

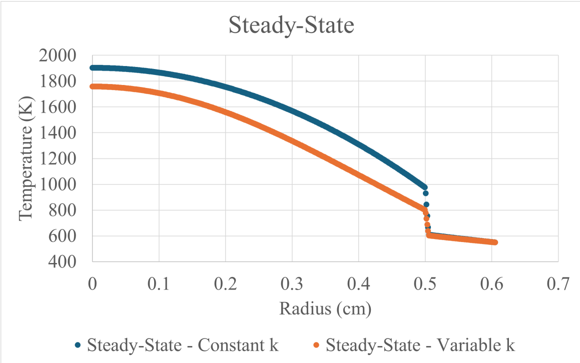


Figure 8: This figure shows the results of the steady-state LHR programs, comparing the temperature profile resulting from a constant vs a variable thermal conductivity.

Transient Linear Heat Rate

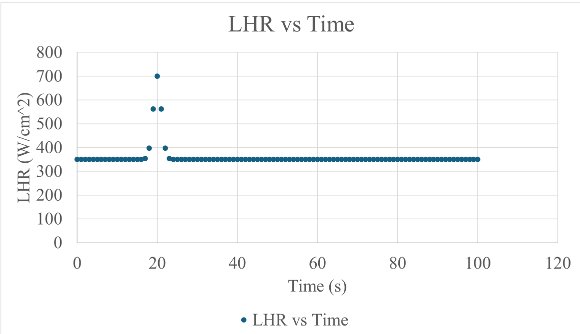


Figure 9: This figure shows the fluctuation of LHR based on the function provided.

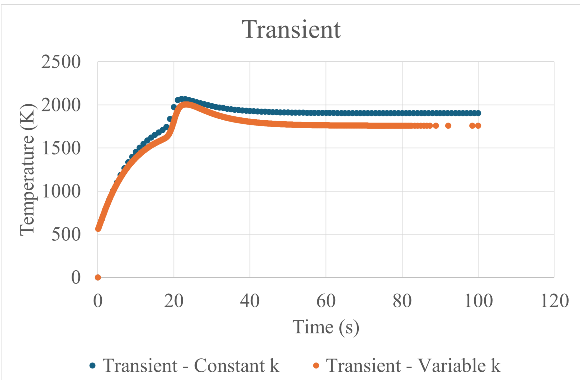


Figure 10: This figure shows the results of the transient LHR programs, comparing the temperature profile resulting from a constant vs a variable thermal conductivity.

Discussion

The lower temperature profiles observed in the constant thermal conductivity plots versus the steady-state temperature profiles can be attributed to the overall increase in thermal conductivity with temperature. A higher thermal conductivity means a greater amount of heat is leaving the fuel pellet, at a higher rate. this leads to the slightly steeper gradient and the higher average temperature profile when k is unchanging. At higher temperatures, heat is conducted more efficiently and the centerline temperatures will decrease when the thermal conductivity is a function of temperature. The thermal conductivity equations represented in the "Equations" section of this report, show how thermal conductivity increases with temperature for the gap and cladding blocks, but not the fuel block. This is because ceramic materials do not conduct heat the same way as metals, and the thermal conductivity increase in the cladding overrules the decrease in thermal conductivity in the fuel, resulting in a lower temperature profile/centerline temperature over time.

It can be observed that the peak in the transient temperature profile is at about the same timestamp as the LHR vs Time plot peaks. This suggests a relationship with LHR increase and temperature spikes over time, there is an initial sharp increase in temperature as the power input in the LHR is increased and once the heat generation equals heat removal, thermal equilibrium is established and the system reaches steady-state, aka when the curve flattens out.

The importance behind studying both steady-state and transient behavior is to show a stable temperature profile to use for long term operation and understanding burn-up of individual fuel rods based on temperature, which can help logically configure fuel rods for the most efficient operation. Transient profiles show the response to changes in power, in this case LHR. It is important to understand the difference between variable and constant thermal conductivity as well, because the variable is more accurate as to what is occurring in the fuel rod, but constant can make for more straight forward assumptions. However, it is necessary to note that using a constant k would result in a consistent overestimation of temperature no matter which case is being looked at.

Conclusion

Using a fuel pellet consisting of UO_2 fuel, entirely Helium gap, and a Zirconium cladding, the maximum centerline temperatures for each program are as follows:

- Steady-state LHR with constant k: 1903 K
- Steady-state LHR with temperature-dependent k: 1758 K
- Transient LHR with constant k: 2070 K
- Transient LHR with temperature-dependent k: 2005 K

The temperature dependent thermal conductivity curves were collectively lower than the constant thermal conductivity curves for both steady-state and transient due to the higher net thermal conductivity in the gap and cladding conducting heat out of the fuel pellet. A transient linear heat rate results in a peak centerline temperature at about the 20-23 second timestep due to the spike in LHR at that time from the transient LHR equation. It is important to simulate each of these situations to understand fuel at different linear points throughout the fuel pellet and after different lengths of time in the reactor. This helps predict fuel behavior and ensure safe operation of nuclear reactors.

References

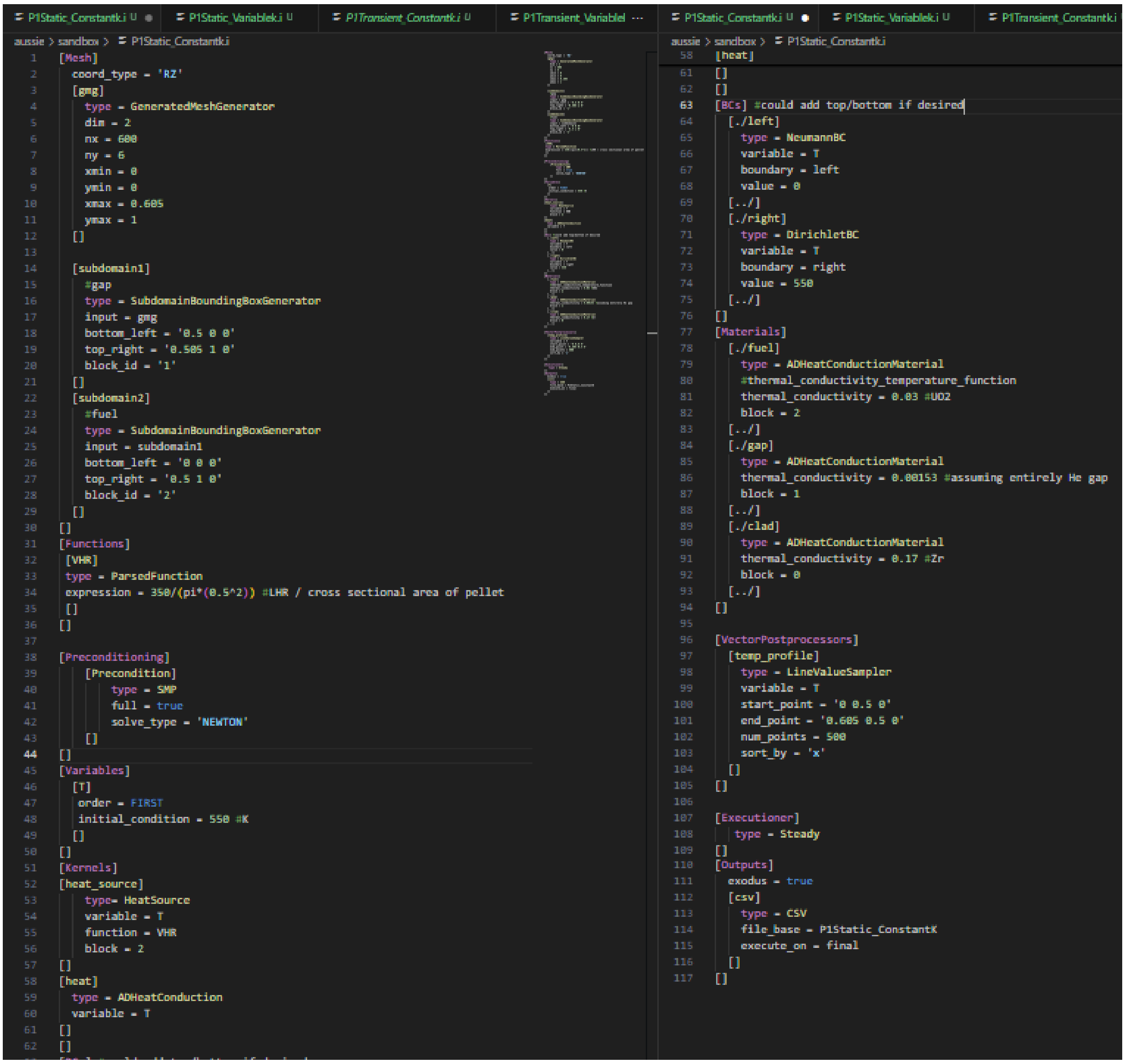
[1] Angstrom Sciences. Thermal conductivity of elements, 2025.

[2] J. K. Fink and L. Leibowitz. Thermal conductivity of zirconium. *Journal of Nuclear Materials*, 226:44–50, Oct 1995.

[3] Idaho National Laboratory. Moose: Multiphysics object oriented simulation environment, 2025.

[4] University of Massachusetts Amherst, Chemistry Department. Appendix: Specific heats, 2025.

Appendix



```
[Mesh]
  coord_type = 'RZ'
  [gmg]
    type = GeneratedMeshGenerator
    dim = 2
    nx = 600
    ny = 6
    xmin = 0
    ymin = 0
    xmax = 0.605
    ymax = 1
  []

  [subdomain1]
    #gap
    type = SubdomainBoundingBoxGenerator
    input = gmg
    bottom_left = '0.5 0 0'
    top_right = '0.505 1 0'
    block_id = '1'
  []
  [subdomain2]
    #fuel
    type = SubdomainBoundingBoxGenerator
    input = subdomain1
    bottom_left = '0 0 0'
    top_right = '0.5 1 0'
    block_id = '2'
  []
[]
[Functions]
 [VHR]
 type = ParsedFunction
 expression = 350/(pi*(0.5^2)) #LHR / cross sectional area of pellet
 []
[]

[Preconditioning]
    [Precondition]
        type = SMP
        full = true
        solve_type = 'NEWTON'
    []
[]
[Variables]
  [T]
   order = FIRST
   initial_condition = 550 #K
  []
[]
[Kernels]
[heat_source]
    type= HeatSource
    variable = T
    function = VHR
    block = 2
[]
[heat]
  type = ADHeatConduction
  variable = T
[]
[]
```

```
[heat]
[]
[]
[BCs] #could add top/bottom if desired
  [./left]
    type = NeumannBC
    variable = T
    boundary = left
    value = 0
  [../]
  [./right]
    type = DirichletBC
    variable = T
    boundary = right
    value = 550
  [../]
[]
[Materials]
  [./fuel]
    type = ADHeatConductionMaterial
    #thermal_conductivity_temperature_function
    thermal_conductivity = 0.03 #UO2
    block = 2
  [../]
  [./gap]
    type = ADHeatConductionMaterial
    thermal_conductivity = 0.00153 #assuming entirely He gap
    block = 1
  [../]
  [./clad]
    type = ADHeatConductionMaterial
    thermal_conductivity = 0.17 #Zr
    block = 0
  [../]
[]

[VectorPostprocessors]
  [temp_profile]
    type = LineValueSampler
    variable = T
    start_point = '0 0.5 0'
    end_point = '0.605 0.5 0'
    num_points = 500
    sort_by = 'x'
  []
[]

[Executioner]
   type = Steady
[]
[Outputs]
  exodus = true
  [csv]
    type = CSV
    file_base = P1Static_ConstantK
    execute_on = final
  []
[]
```

Figure 11: This figure shows the MOOSE program for the steady-state LHR with a constant thermal conductivity temperature profile.

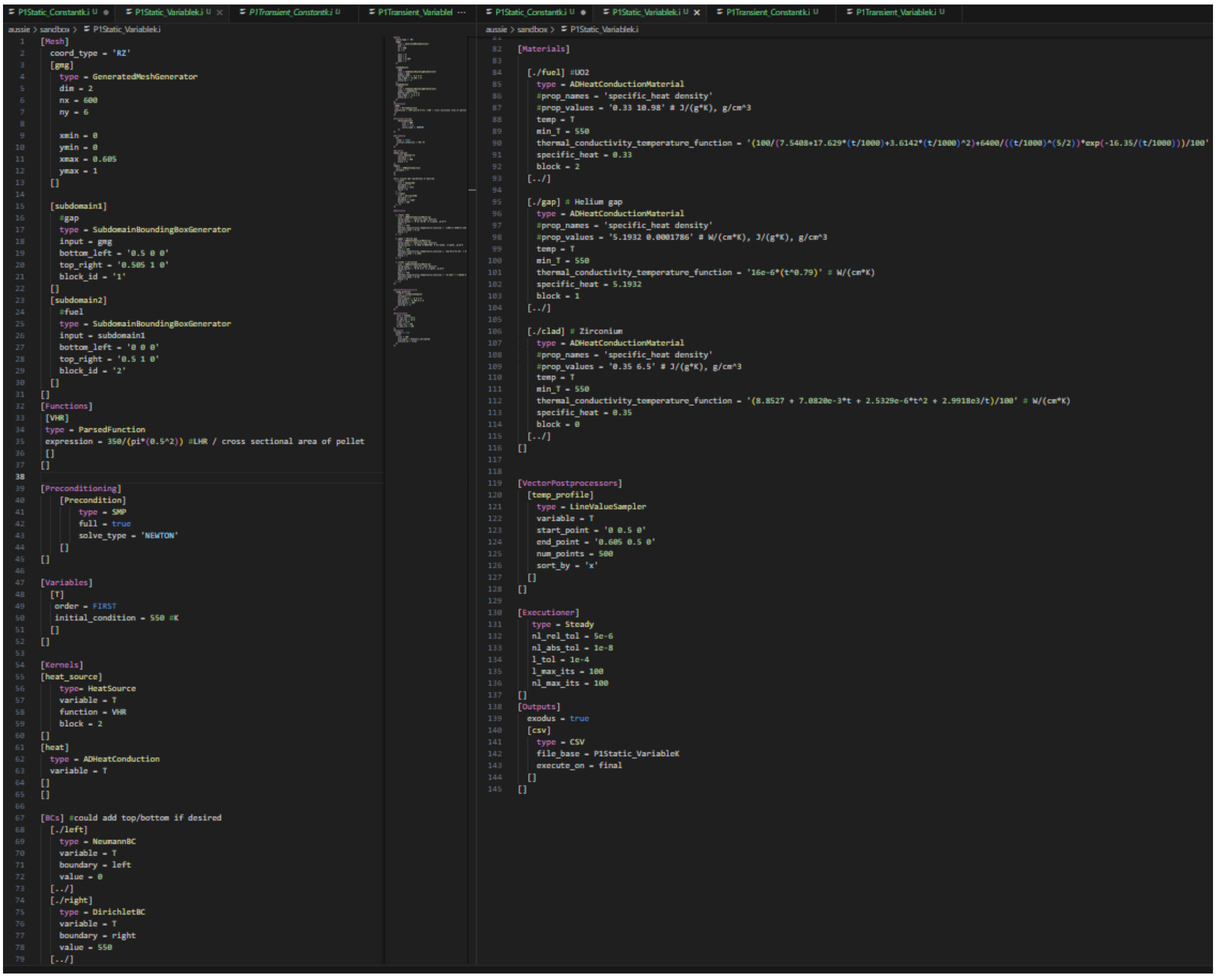
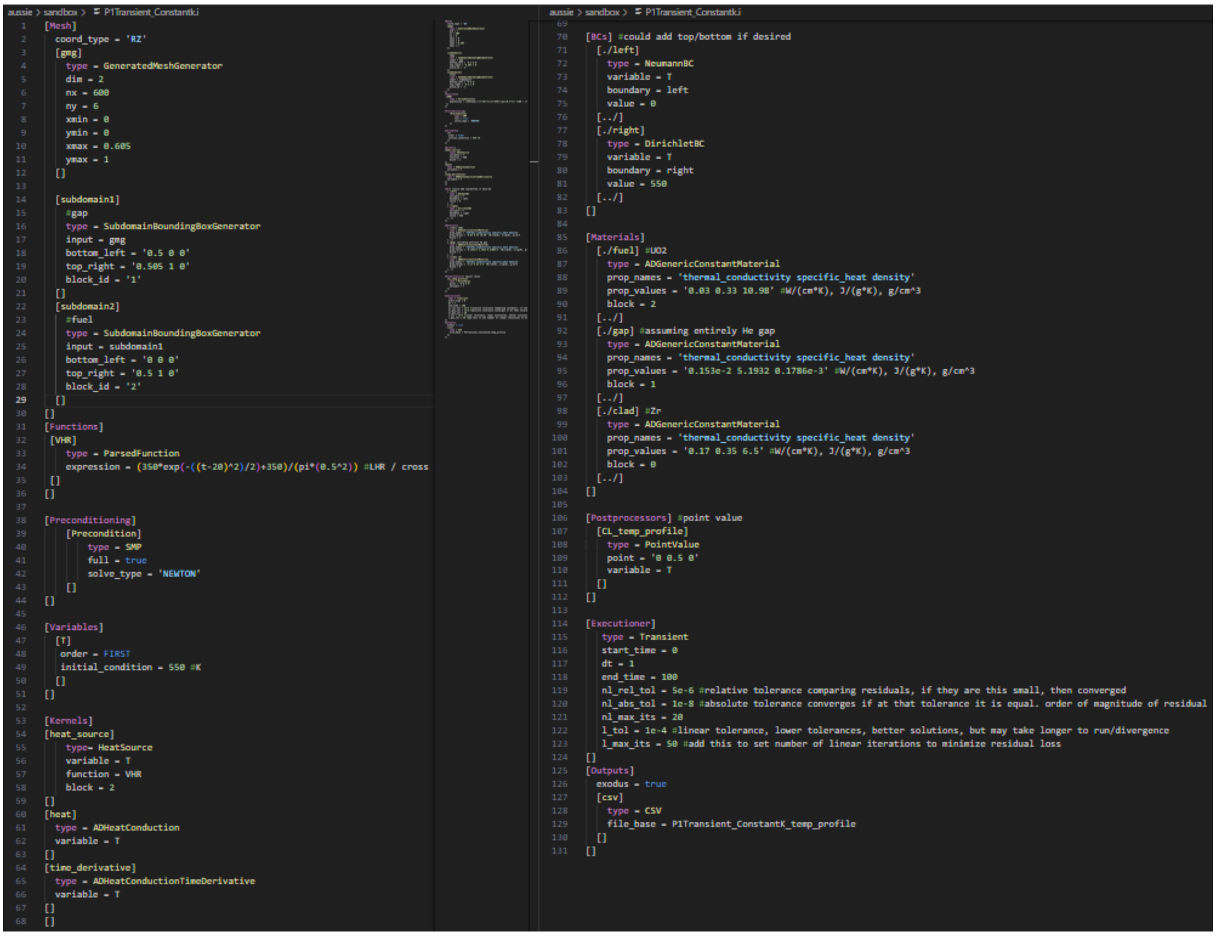
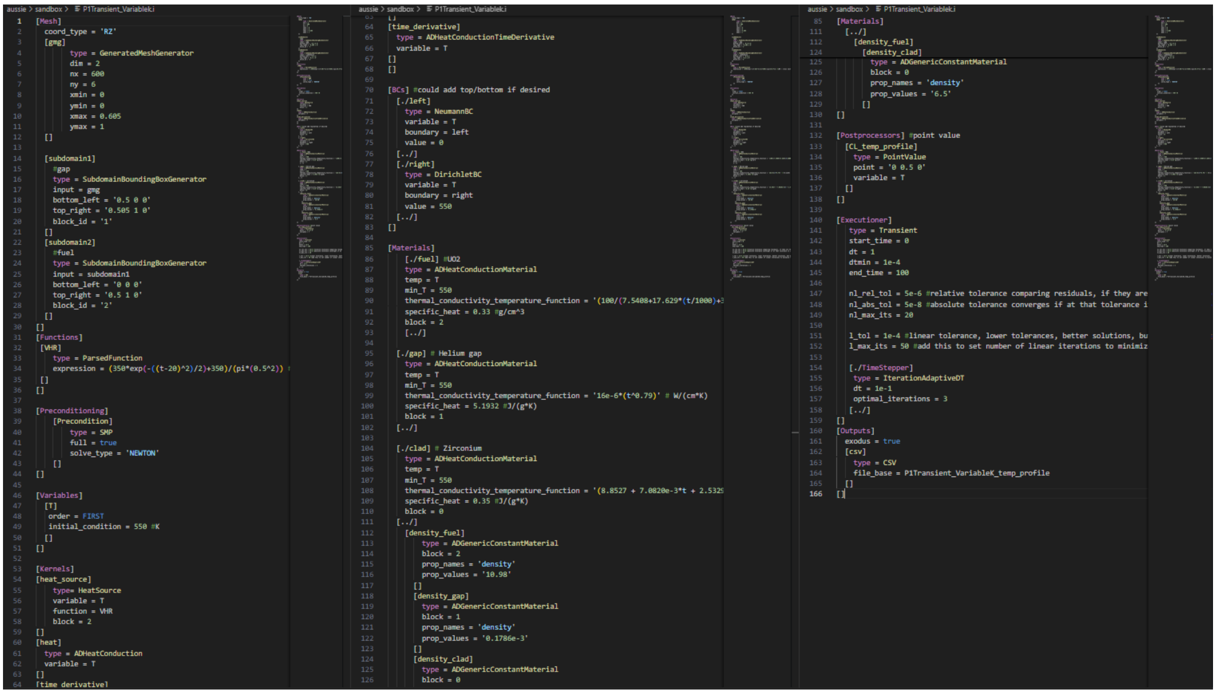


Figure 12: This figure shows the MOOSE program for the steady-state LHR with a variable thermal conductivity temperature profile.



```
aussie > sandbox > P1Transient_ConstantK.i
[Mesh]
  coord_type = 'RZ'
  [gmg]
    type = GeneratedMeshGenerator
    dim = 2
    nx = 600
    ny = 6
    xmin = 0
    ymin = 0
    xmax = 0.605
    ymax = 1
  []

  [subdomain1]
    #gap
    type = SubdomainBoundingBoxGenerator
    input = gmg
    bottom_left = '0.5 0 0'
    top_right = '0.505 1 0'
    block_id = '1'
  []
  [subdomain2]
    #fuel
    type = SubdomainBoundingBoxGenerator
    input = subdomain1
    bottom_left = '0 0 0'
    top_right = '0.5 1 0'
    block_id = '2'
  []
[]
[Functions]
 [VHR]
    type = ParsedFunction
    expression = (350*exp(-((t-20)^2)/2)+350)/(pi*(0.5^2)) #LHR / cross
 []
[]

[Preconditioning]
    [Precondition]
        type = SMP
        full = true
        solve_type = 'NEWTON'
    []
[]

[Variables]
  [T]
   order = FIRST
   initial_condition = 550 #K
  []
[]

[Kernels]
[heat_source]
    type= HeatSource
    variable = T
    function = VHR
    block = 2
[]
[heat]
  type = ADHeatConduction
  variable = T
[]
[time_derivative]
  type = ADHeatConductionTimeDerivative
  variable = T
[]
[]

[BCs] #could add top/bottom if desired
  [./left]
    type = NeumannBC
    variable = T
    boundary = left
    value = 0
  [../]
  [./right]
    type = DirichletBC
    variable = T
    boundary = right
    value = 550
  [../]
[]

[Materials]
  [./fuel] #UO2
    type = ADGenericConstantMaterial
    prop_names = 'thermal_conductivity specific_heat density'
    prop_values = '0.03 0.33 10.98' #W/(cm*K), J/(g*K), g/cm^3
    block = 2
  [../]
  [./gap] #assuming entirely He gap
    type = ADGenericConstantMaterial
    prop_names = 'thermal_conductivity specific_heat density'
    prop_values = '0.153e-2 5.1932 0.1786e-3' #W/(cm*K), J/(g*K), g/cm^3
    block = 1
  [../]
  [./clad] #Zr
    type = ADGenericConstantMaterial
    prop_names = 'thermal_conductivity specific_heat density'
    prop_values = '0.17 0.35 6.5' #W/(cm*K), J/(g*K), g/cm^3
    block = 0
  [../]
[]

[Postprocessors] #point value
  [CL_temp_profile]
    type = PointValue
    point = '0 0.5 0'
    variable = T
  []
[]

[Executioner]
   type = Transient
   start_time = 0
   dt = 1
   end_time = 100
   nl_rel_tol = 5e-6 #relative tolerance comparing residuals, if they are this small, then converged
   nl_abs_tol = 1e-8 #absolute tolerance converges if at that tolerance it is equal. order of magnitude of residual
   nl_max_its = 20
   l_tol = 1e-4 #linear tolerance, lower tolerances, better solutions, but may take longer to run/divergence
   l_max_its = 50 #add this to set number of linear iterations to minimize residual loss
[]
[Outputs]
  exodus = true
  [csv]
    type = CSV
    file_base = P1Transient_ConstantK_temp_profile
  []
[]
```

Figure 13: This figure shows the MOOSE program for the transient LHR with a constant thermal conductivity temperature profile.



```
aussie > sandbox > P1Transient_VariableK.i
[Mesh]
  coord_type = 'RZ'
  [gmg]
        type = GeneratedMeshGenerator
        dim = 2
        nx = 600
        ny = 6
        xmin = 0
        ymin = 0
        xmax = 0.605
        ymax = 1
  []

  [subdomain1]
    #gap
    type = SubdomainBoundingBoxGenerator
    input = gmg
    bottom_left = '0.5 0 0'
    top_right = '0.505 1 0'
    block_id = '1'
  []
  [subdomain2]
    #fuel
    type = SubdomainBoundingBoxGenerator
    input = subdomain1
    bottom_left = '0 0 0'
    top_right = '0.5 1 0'
    block_id = '2'
  []
[]
[Functions]
 [VHR]
    type = ParsedFunction
    expression = (350*exp(-((t-20)^2)/2)+350)/(pi*(0.5^2))
 []
[]

[Preconditioning]
    [Precondition]
        type = SMP
        full = true
        solve_type = 'NEWTON'
    []
[]

[Variables]
  [T]
   order = FIRST
   initial_condition = 550 #K
  []
[]

[Kernels]
[heat_source]
    type= HeatSource
    variable = T
    function = VHR
    block = 2
[]
[heat]
  type = ADHeatConduction
  variable = T
[]
[time_derivative]
  type = ADHeatConductionTimeDerivative
  variable = T
[]
[]

[BCs] #could add top/bottom if desired
  [./left]
    type = NeumannBC
    variable = T
    boundary = left
    value = 0
  [../]
  [./right]
    type = DirichletBC
    variable = T
    boundary = right
    value = 550
  [../]
[]

[Materials]
    [./fuel] #UO2
    type = ADHeatConductionMaterial
    temp = T
    min_T = 550
    thermal_conductivity_temperature_function = '(100/(7.5408+17.629*(t/1000)+3
    specific_heat = 0.33 #g/cm^3
    block = 2
    [../]

  [./gap] # Helium gap
    type = ADHeatConductionMaterial
    temp = T
    min_T = 550
    thermal_conductivity_temperature_function = '16e-6*(t^0.79)' # W/(cm*K)
    specific_heat = 5.1932 #J/(g*K)
    block = 1
  [../]

  [./clad] # Zirconium
    type = ADHeatConductionMaterial
    temp = T
    min_T = 550
    thermal_conductivity_temperature_function = '(8.8527 + 7.0820e-3*t + 2.5329
    specific_heat = 0.35 #J/(g*K)
    block = 0
  [../]
    [density_fuel]
        type = ADGenericConstantMaterial
        block = 2
        prop_names = 'density'
        prop_values = '10.98'
      []
      [density_gap]
        type = ADGenericConstantMaterial
        block = 1
        prop_names = 'density'
        prop_values = '0.1786e-3'
      []
      [density_clad]
        type = ADGenericConstantMaterial
        block = 0
        prop_names = 'density'
        prop_values = '6.5'
      []
[]

[Postprocessors] #point value
  [CL_temp_profile]
    type = PointValue
    point = '0 0.5 0'
    variable = T
  []
[]

[Executioner]
   type = Transient
   start_time = 0
   dt = 1
   dtmin = 1e-4
   end_time = 100

   nl_rel_tol = 5e-6 #relative tolerance comparing residuals, if they are
   nl_abs_tol = 5e-8 #absolute tolerance converges if at that tolerance i
   nl_max_its = 20

   l_tol = 1e-4 #linear tolerance, lower tolerances, better solutions, bu
   l_max_its = 50 #add this to set number of linear iterations to minimiz

   [./TimeStepper]
    type = IterationAdaptiveDT
    dt = 1e-1
    optimal_iterations = 3
   [../]
[]
[Outputs]
  exodus = true
  [csv]
    type = CSV
    file_base = P1Transient_VariableK_temp_profile
  []
[]
```

Figure 14: This figure shows the MOOSE program for the transient LHR with a variable thermal conductivity temperature profile.