NRC Multiphysics Analysis Capability Deployment FY2020 - Part 3

Paolo Balestra[1], Zachary Prince[1], Javier Ortensi[2], Sebastian Schunert[1], Yaqi Wang[1], William M. Hoffman[3], Albert Casagranda[3], Mark D DeHart[2]

[1] Nuclear Engineering Methods Development
[2] Reactor Physics Design and Analysis
[3] Computational Mechanics and Materials

Idaho National Laboratory
P.O. Box 1625
Idaho Falls, ID 83415-3840

June 2020

NRC Multiphysics Analysis Capability Deployment FY2020 - Part 3

Paolo Balestra, Zachary Prince, Javier Ortensi, Sebastian Schunert, Yaqi Wang, William M. Hoffman, Albert Casagranda, Mark D DeHart

June 2020

Idaho National Laboratory
Nuclear Science & Technology Division
Idaho Falls, Idaho 83415

http://www.inl.gov

Abstract

This report details progress and activities of Idaho National Laboratory (INL) on the Nuclear Regulatory Commission (NRC) project "Development and Modeling Support for Advanced Non-Light Water Reactors."

The tasks completed for this report are:

- Task 2c: Explicit modeling of pebble transient temperature response. In this simulation, the 400 MWth Pebble-Bed Modular Reactor (PBMR) design, PBMR-400, experiences a 20-second power ramp from 100% to 150% power. This is followed by a similar reduction in the power back to 100%. Several multiscale pebble coupling approaches are tested with one pebble per mesh element in the active core region. The results show good conservation behavior and the stability of the coupling.

- Extended scope part 1: An assessment of the computational efficiency of the Discontinuous Finite Element Method (DFEM) heat transfer solver shows good scalability. The DFEM solver is a factor of 4 more expensive in solution time than the Finite Element Method (FEM) solver for heat transfer problems due to the increased number of degrees of freedom. Nonetheless, the DFEM approach provides the user with the flexibility to model gap heat transfer problems.

- Extended scope part 2: The *GapHeatTransferInterfaceMaterial* was improved to give the user increased flexibility with the modeling of heat transfer through gaps with the DFEM solver. A number of gap parameters can now be coupled both through functions and variables.

- Extended scope part 3: Demonstration of how the gap width between hexagonal fuel cells can be calculated during a heat-up transient and used in the *GapHeatTransferInterface* model. A full-domain DFEM model with gap expansion is coupled to a SubApp that models the thermal expansion of the base plate. The results show the expected physical behavior, although have not been fully benchmarked at this point in time.

Contents

iii

Acronyms

AMG Algebraic Multi-Grid. 12–14

CFEM Continuous Finite Element Method. vii, ix, 11–13

DFEM Discontinuous Finite Element Method. ii, iii, vii, ix, 1, 11–15, 19–21, 23, 24, 26–31

DoF Degrees-of-Freedom. 12, 14

FEM Finite Element Method. ii, 12

HTR High Temperature Reactor. 13

INL Idaho National Laboratory. ii, 1, 31

MOOSE Multiphysics Object Oriented Simulation Environment. 2, 12, 13, 19, 26

NRC Nuclear Regulatory Commission. ii, 1, 11, 31

PBMR Pebble-Bed Modular Reactor. ii, 2

SOW Statement-of-Work. 1

SuperLU Super Lower–Upper. 11, 12

TRISO TRi-structural ISOtropic. 2

List of Figures

List of Tables

x

1 Introduction

This report details progress and activities of Idaho National Laboratory (INL) on the Nuclear Regulatory Commission (NRC) project "Development and Modeling Support for Advanced Non-Light Water Reactors."

Table 1 provides a summary of the tasks completed for this report. It matches the deliverable number, the Statement-of-Work (SOW) number, and a short description of the deliverable to the relevant section in this report.

Table 1: List of Deliverables.

Deliverable Number	SOW Task	Report Section	Description
2b	2c*	2	Explicit modeling of pebble transient temperature response
Extended scope	NA	3	Improvements to gap heat transfer model

* Task 2c advanced from 9/6/2020 and Task 3c delayed until 9/6/2020.

The tasks completed for this report are:

- Task 2c: Develop a sample problem demonstrating the explicit modeling of the pebble transient temperature response. One prototypic pebble should be modeled for every element in the pebble bed; the pebble transient conduction equation should be coupled to the porous medium solid energy equation with the solid temperature used as the boundary condition for the pebble. Coupling should be bidirectional, that is, porous medium solid temperature to pebble, and pebble heat flux as volumetric heat source to porous medium.

- Extended scope part 1: The new *GapHeatTransferInterface* model requires the usage of the discontinuous Galerkin form for the finite element analysis. Assess the computational efficiency for DFEM in heat transfer problems.

- Extended scope part 2: Enhancements to the *GapHeatTransferInterfaceMaterial* to accept variables to model the gap heat transfer phenomena.

- Extended scope part 3: Develop a problem demonstrating how the gap width between hexagonal fuel cells can be calculated during a heat-up transient and used in the *GapHeatTransferInterface* model.

2 Explicit modeling of pebble transient temperature response (Task 2c)

The inherent flexibility in the Multiphysics Object Oriented Simulation Environment (MOOSE) framework allows the development of various models with different degrees of approximation for explicit pebble temperature calculations during transient simulations. These explicit pebble coupling approaches leverage the MultiApp features available in the MOOSE framework. Using MultiApps the pebble model fidelity level is limited only by the computational resources (i.e., the same Pronghorn pebble bed input file can be coupled to a homogeneous pebble model or to a 3-D BISON model, using the same coupling schemes). In addition, it is possible to use a large variety of predefined materials, to model the different layers of the pebble and TRi-structural ISOtropic (TRISO) fuel.

2.1 Model and transient example description

Simplified models for a pebble and a pebble bed porous medium have been developed to demonstrate and test these capabilities. The size of the simplified models (number of mesh elements and degrees of freedoms) is that of a typical 2-D R-Z model developed for transient simulations. To check the accuracy of the different coupling schemes, a simple transient scenario involving a change in the heat generation rate (power) is selected.

2.1.1 Pebble bed porous media and pebble model description

The Pebble bed geometry and material properties reported in Table 2 are based on PBMR-400 benchmark specifications [1] and some steady state calculations published in [2]. As shown in Figure 1 (left) a linear axial power distribution following the (1) is imposed. A non-uniform power distribution is imposed to promote the heat transfer by diffusion and to have a different power density in each pebble making the coupling more challenging.

$$q'''(y,t) = \frac{P(t)}{V_{pb}} \cdot (0.5 + y/h_{pb}) \tag{1}$$

where $P(t)$ is the total pebble bed power as a function of time, V_{pb} is the total pebble bed volume, h_{pb} is the total bed height, and y is the axial coordinate. The 2-D R-Z Pebble bed model (active core) has been discretized using a 22 x 5 regular rectangular mesh elements

for a total of 110 elements. Each one of the 110 pebble bed porous media elements is coupled with a 1-D spherical pebble model shown in Figure 1 (right), discretized using 60 radial mesh points.
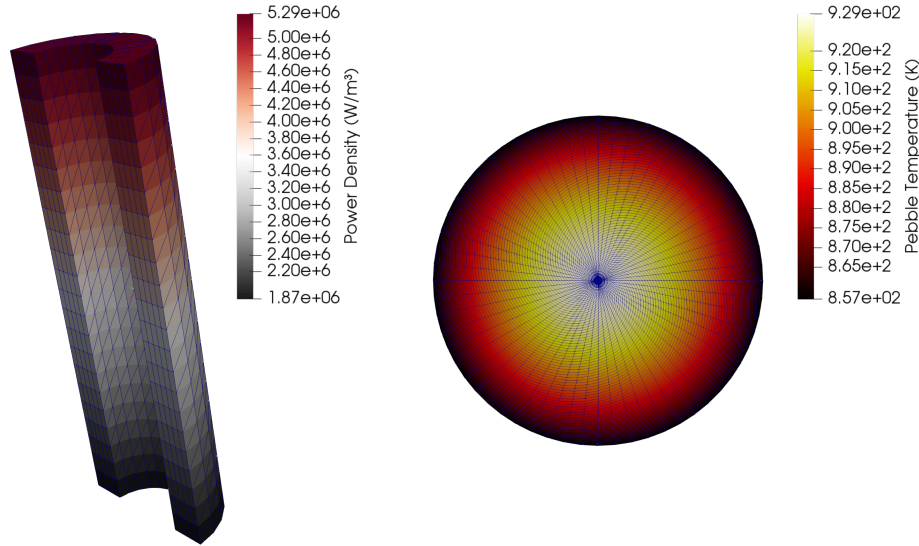
Figure 1: Annular pebble bed model geometry with linear power distribution (left); Pebble model with example temperature distribution (right).

2.1.2 Transient description — power ramp

A significant change in one of the conditions imposed on the model is necessary to test the transient response of the coupling schemes. The particular transient of interest is initiated with a change to the total heat generation rate (power) that is shown in Figure 2. The first 500s are used to stabilize the model solution to a steady state level. This is followed by a linear power ramp with a duration of 20s of duration from 100% to 150% (450.0 MW) rated power. After another 500s of solution stabilization at the new power level, a new linear power ramp is applied for 20s to bring the power back down to 100%. The remaining 1000s are used to let the model solution stabilize again and to verify that the models reach the same pre-ramp operation conditions.

Table 2: Model geometry and material properties.

Description	Symbol	Units	Value
Pebble bed height	h	m	11.0
Pebble bed inner radius	r_{in}	m	1.0
Pebble bed outer radius	r_{out}	m	1.85
Pebble bed total power	P	MW	300.0
Pebble bed total mass flow rate	Γ	kg/m^3	150.0
Pebble bed porosity	ε	//	0.39
Pebble bed total equivalent conductivity (contact, convection, radiation)	$k_{eq,s}$	W/m/K	180.0
Pebble bed volumetric convection heat exchange coefficient	α	W/m^3/K	2.6e+5
Pebble bed pressure drop coefficient ($dP = w \cdot \rho v^2 dL$)	w	m^{-1}	3.5
Pebble bed coolant	//	//	Helium
Pebble bed He inlet temperature	T_{in}	K	773.15
Pebble bed He outlet Pressure	P_{out}	Pa	9.0e+6
Pebble radius	r_p	m	3.0e-2
Pebble conductivity	k_s	W/m/K	26.0
Pebble specific heat	$c_{p,s}$	W/kg/K	1.697e+3
Pebble density	ρ_s	kg/m^3	1.78e+3

Figure 2: Pebble bed total power as function of time, 20s linear power ramp to 150% of the power and back to the 100% after 500s

2.2 Coupling schemes description

Different coupling schemes have been tested with the aim of studying the effect of the different approximations on the model output parameters, such as the temperature of the pebble center during the transient. Three different approaches have been studied in order of complexity:
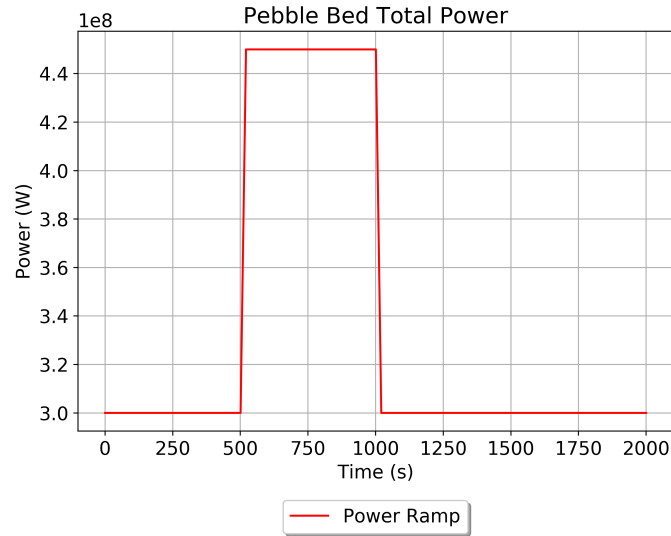
1. Steady state model.

2. A porous media transient model one way coupled to a steady state pebble model.

3. A pebble transient model fully coupled with the porous media model.

Shown in Figure 3a, the steady state model excludes the time derivative terms $\frac{\partial T}{\partial t}$ in both the pebble bed and pebble model equations. The power q''' is imposed equally in the two models, since the absence of thermal inertia (i.e., $\rho * C_p$ is neglected) makes unnecessary the transfer of power from the pebble model to the pebble bed porous media model. The only transfer needed is the pebble surface temperature that is calculated as `T_solid` in the porous media, which is transferred to the pebble model as (`T_surface`) and used in a Dirichlet boundary condition on the pebble surface.

The next coupling scheme utilizes the porous media transient solver (see Figure 3b). The time derivative term is active in the porous media model in which the heat capacity and density are a volume-weighted mixture of the properties used for the pebble model; in this case it's just one material but there is no limitation on the number of materials, or how mix them. The time-derivative term in the pebble model is deactivated since the thermal inertia is already taken into account in the porous media model, and, again, the surface temperature is transferred. The big advantage of this methodology is that the one-way coupling removes the need to: (1) iterate between the two models and (2) transfer the power, since the pebble has no thermal inertia. The drawback is that there is no guarantee that during the transient, the pebble outgoing thermal flux will match the sum of the heat deposited in the fluid by convection, and the diffusive flux within the porous media. Another issue with this approach, is that using the porous media transient model is not possible to reproduce perfectly the real thermal inertia of the pebble, especially when the properties depend on the temperature because the average temperature of the porous media is used to calculate the properties.

The last coupling scheme (see Figure 3c) is the most sophisticated because it removes the approximations introduced in the one-way coupling. The time derivative term is inactive

in the porous media model, but it is active in the pebble model. The heat generation rate is only imposed in the pebble model and therefore a modified heat generation rate must be transferred to the porous media as part of the coupling. The simplest coupling of the two models is done by transferring the `T_solid` from the porous media and imposing it as a boundary condition in the pebble model (as in the previous two coupling schemes), then returning the outgoing thermal flux and imposing it as source term in the porous media. However, this approach is unstable, since a small change in the surface temperature can produce an big change in the pebble surface thermal flux, destabilizing the solution of the porous media model, which does not contain a time derivative term. As shown in (see Figure 3c) a different coupling scheme that relaxes the convergence has been used. First, the thermal flux from conduction and convection by the solid part in the porous media is transferred to the pebble model and imposed using a Neumann boundary condition. Then, a modified heat generation term is sent back to the MasterApp. This modified term consists of the heat generation rate, based on the heat flux, plus an additional volumetric penalty term that is proportional to the difference between the porous media `T_solid` and the just calculated pebble surface temperature `T_surface`. The two models iterate at each time step until the additional term vanishes, when the Porous media `T_solid` and the pebble surface temperature `T_surface` converge to the same value.

(a) Steady State

$\boxed{\cancel{\dfrac{\partial T}{\partial t}} \quad q'''}$ $\boxed{\cancel{\dfrac{\partial T}{\partial t}} \quad q'''}$

Porous Media

Unidirectional

$T_{solid} = T_{surface}$ as BC

Pebbles

(b) Transient - One Way

$\boxed{\bigcirc\!\!\!\dfrac{\partial T}{\partial t} \quad q'''}$ $\boxed{\cancel{\dfrac{\partial T}{\partial t}} \quad q'''}$

Porous Media

Unidirectional

$T_{solid} = T_{surface}$ as BC

Pebbles

(c) Transient - Full

$\boxed{\cancel{\dfrac{\partial T}{\partial t}} \quad \cancel{q'''}}$ $\boxed{\bigcirc\!\!\!\dfrac{\partial T}{\partial t} \quad q'''}$

Porous Media

Picard Iterations

q'' as BC, T_{solid}

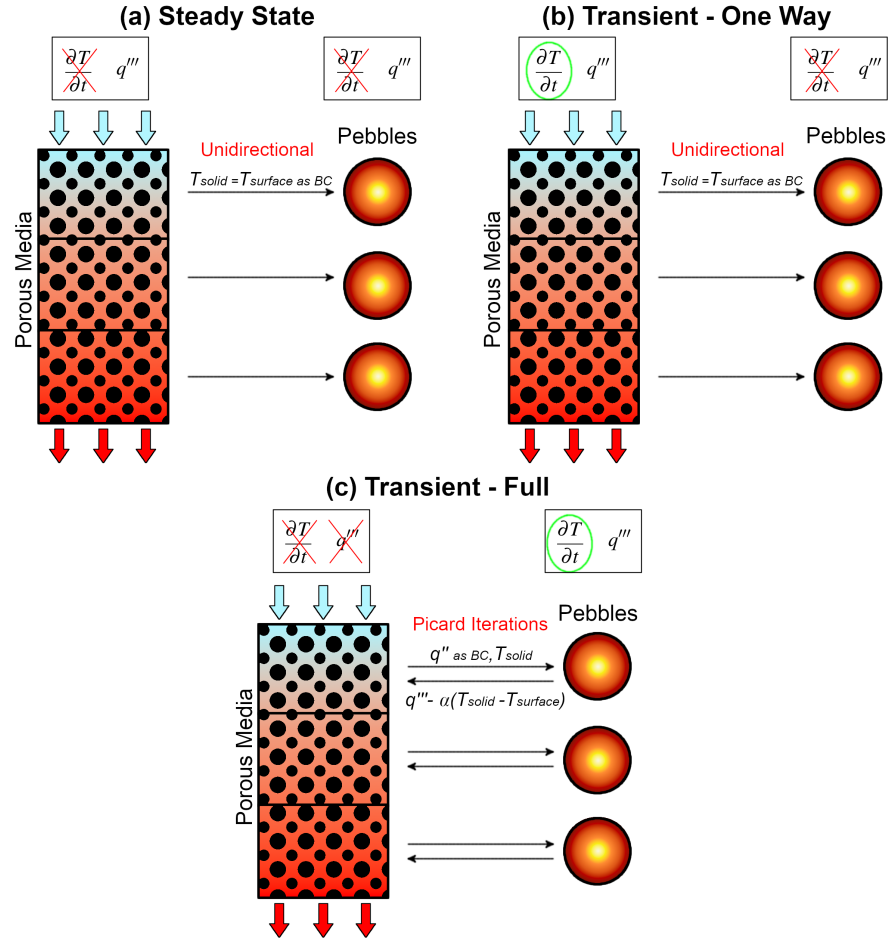$q''' - \alpha(T_{solid} - T_{surface})$

Pebbles

Figure 3: Coupling schemes between the porous medium and the pebble models. The black arrows represent the transferred quantities (the ones used as boundary conditions are followed by *as BC*).

7

2.3 Transient results analysis

The three coupling schemes described in Section 2.2 have been tested using the model and the transient described in Section 2.1. Figure 4 shows the total (all the pebbles) integral of the heat flux on the pebble surface. This plot confirms that the power is preserved during the transfer with the 110 MultiApps since all three simulations reach the total power that is generated into the system (300 MW initially and 450 MW at the end of the ramp) asymptotically. The only coupling scheme that shows a delay in the total pebble outlet power is the fully coupled one since it is the only scheme that contains a time derivative term in the pebble equations. The steady state coupling and the one-way coupling instead match immediately the imposed total power. Shown in Figures 5, 6, and 7, for all the
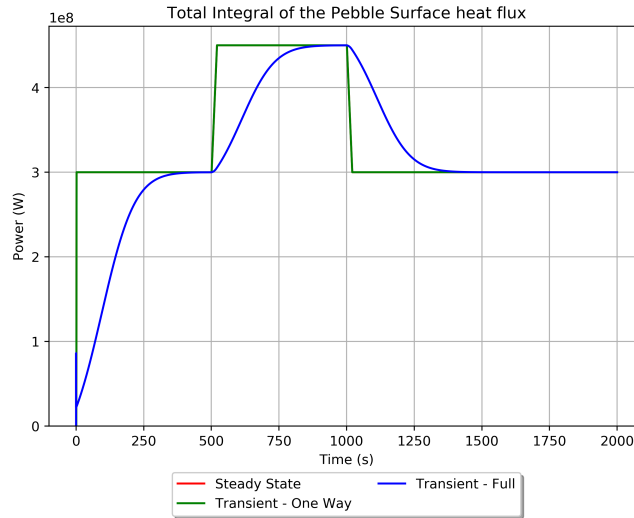
Figure 4: Average integral of the heat flux at the pebble surface.

different schemes, the pebble center temperatures, pebbles surface temperatures, and fluid temperatures converge to the same value after 500s, thus confirming the three coupling schemes are correctly implemented and preserve energy during the various transfers. The steady state model shape follows perfectly the power ramp timing, since no thermal inertia is simulated. The one-way coupling scheme instead shows a temperature trend very close to the fully coupled model, even if the pebble surface flux is not correctly simulated and there is no inertia in the pebble model. This demonstrates that simulating the thermal inertia using the porous medium is still possible for a slow transient without diverging too much from the fully coupled solution. The imbalance between the heat generated and removed

8

from the pebble during the transient generates a discrepancy between the two models. The fluid temperature is almost the same as shown in Figure 7, on the contrary the discrepancy in the pebble center temperature is clearly visible in Figure 5. The fluid temperature change is smaller in magnitude because it is driven by the equilibrium in the pebble heat generation rate, which is delayed with respect to the change in the pebble center temperature. Finally, as shown in Figure 5(b), the maximum pebble center line temperature discrepancy between the two coupling schemes (the temperature change delay between the one-way and the fully coupled scheme) is bigger than the discrepancy in the average value demonstrating the temperature shape (not only the average value) changes during the transient, as shown in Figure 5(a).
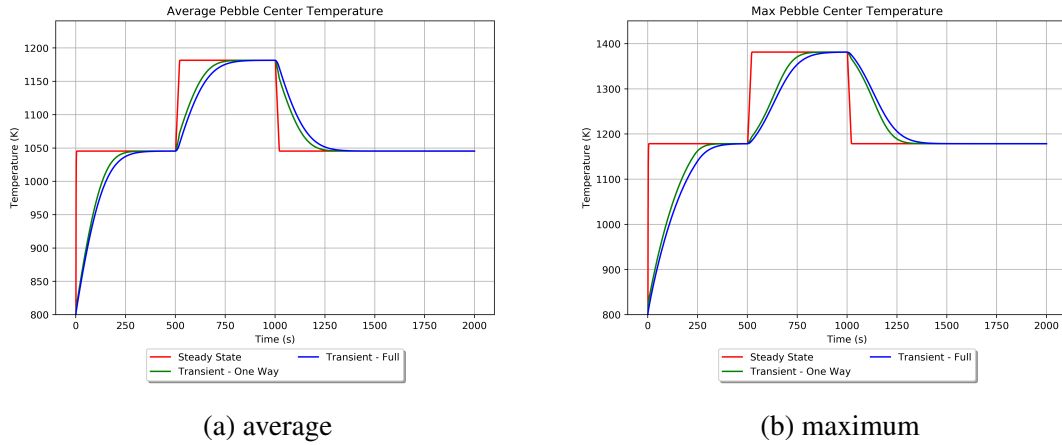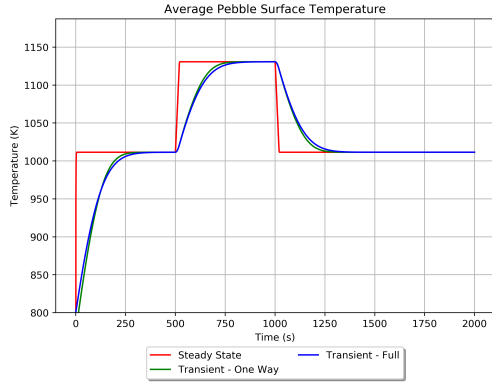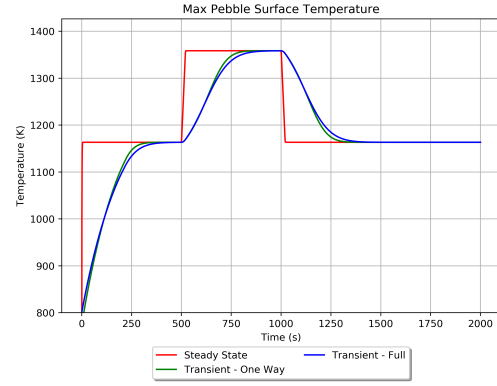

(a) average

(b) maximum

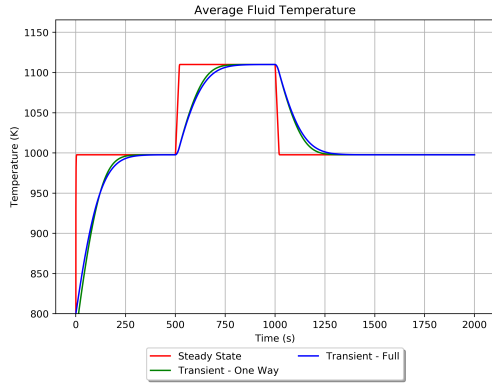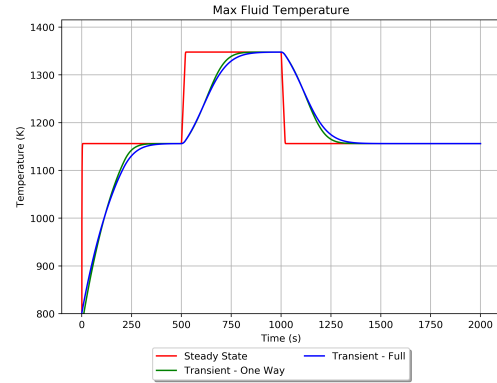Figure 5: Pebble center temperatures [K].

(a) average

(b) maximum

Figure 6: Pebble surface temperatures [K].

(a) average

(b) maximum

Figure 7: Fluid temperatures [K].

3 Gap heat transfer model

Several heat pipe microreators with hexagonal cladding have been proposed in recent years [3, 4]. In these designs, the fuel cell is surrounded by a gap that can present a significant resistance to heat transfer in the case of a heat pipe failure. There are two phenomena that can improve or degrade the heat transfer across these gaps. First, as the cladding heats up, it tends to thermally expand and reduce the effective gap distance between the cladding. This tends to improve the heat transfer between fuel cells. Second, as the support plate heats up, it tends to increase the cell pitch, thus increasing the size of the gap and hindering heat transfer. The NRC identified the need to model the gap thermal expansion from both phenomena with an approximated Discontinuous Finite Element Method (DFEM) approach. This analysis methodology does not require a thermomechanical solution for the full core and has been generalized (i.e., is applicable to any reactor design with gaps). The NRC proposed an extended scope of work to test and improve this capability. First, the efficiency of the DFEM heat transfer problem in tested in Section 3.1. Several improvements requested were added to DFEM model and are discussed in Section 3.2. Finally, several tests are performed with a representative hexagonal geometry in Section 3.3.

3.1 Computational efficiency for DFEM

The most pressing concern about the gap heat transfer model implementation is how it affects simulation time. The model requires the use of the DFEM because the nature of the model produces discontinuous solutions. Traditionally, elliptic problems with smooth solutions (typical for heat transfer) are discretized with a Continuous Finite Element Method (CFEM). However, DFEM produces many more degrees of freedom for the same mesh and will therefore require much more effort to solve. The NRC compared the runtime of a CFEM model (no gap heat transfer) and a DFEM (with gap model) and observed that the DFEM runtime was 14 times that of CFEM, despite only having 4 times the number of degrees of freedom and no significant difference in the number of linear iterations.

The NRC sent the input files that created these results and the Griffin team profiled the simulation time of both methods. Upon this investigation, it became clear that the solver technique was the main culprit in the runtime discrepancy. The solver used a SuperLU preconditioner, which directly inverts the Jacobian matrix at every nonlinear iteration. This is why there is very little difference in the number of linear iterations between CFEM and DFEM. However, SuperLU does not scale linearly with the number of degrees of freedom,

which is why DFEM is taking so much more time building the preconditioner. Switching to a Algebraic Multi-Grid (AMG) preconditioner would cut down on the time spent building the preconditioner significantly. However, AMG has poor linear performance with monomial basis functions (more linear iterations required versus CFEM). Therefore, using L_2 Lagrange basis functions is preferable for an **AMG** preconditioner. Table 3 shows a comparison of DFEM and CFEM when switching the preconditioner and variable type.

Table 3: Runtime performance comparison between CFEM and DFEM with gap heat transfer

	SuperLU		AMG	
		DFEM		DFEM
	CFEM	(Monomial)	CFEM	(L_2 Lagrange)
CPU Time (s) (4 cores)	13.66	154.77	10.27	80.31
CPU Time (s) (16 cores)	7.77	54.14	4.52	24.26
CPU Time (s) (32 cores)	7.00	34.76	3.83	14.68
No. of DoF	46,704	168,896	46,704	337,792
Nonlinear Iterations	3	3	3	3
Linear Iterations	6	12	32	25

Here we see that the number of degrees of freedom increases even more when using L_2 Lagrange elements. However, this element type paired with an AMG preconditioner reduces runtime by half, despite there being double the number of Degrees-of-Freedom (DoF)s. The runtime per DoF is now comparable to CFEM. Furthermore, the large number of DoFs allows more processors to be used without degrading parallel efficiency. In keeping 10,000 DoFs per CPU, CFEM-AMG with four cores and DFEM-AMG with 32 cores have approximately the same runtime. These results will vary from problem to problem, but the gap heat transfer model implementation does not seem to change the performance of what is expected from DFEM in general. It is important to note that first-order monomial elements always have four DoFs per element, while first-order L_2 Lagrange changes depending on the element type: four for tetrahedra, eight for hexahedra, six for prisms, and five for pyramids. It is also important to point out that AMG does not work well with a monomial basis, since AMG preconditioning is not well suited for the resulting operator. Also, SuperLU further degrades performance with a L_2 Lagrange basis since the number of DoFs increases significantly.

The simulations for the AMG runs were also profiled to determine the most time consuming tasks during the simulation. In both simulations, it was quite apparent that mapping and evaluating shape functions within libMesh (the FEM library MOOSE uses), which occurs

at every residual evaluation was the most CPU intensive task. Although this functionality is not optimized for this particular problem, it allows for a lot of flexibility for the user, including being able to perform adaptive mesh refinement and accounting for displaced meshes. With AMG, 96.5% of the total solve time for CFEM and DFEM is spent computing the residual and <0.1% is spent computing the gap heat transfer kernel. Therefore, improving performance for this type of problem requires a significant refactor in MOOSE so that the effect of the gap heat transfer model on a DFEM mesh is negligible.

In order to fairly compare performance between CFEM and DFEM, a study was designed so the same model was used for both schemes, different levels of refinement were applied, and the error for the simulation was compared. The model used is the 14-block High Temperature Reactor (HTR) prismatic geometry from the deliverables in a previous report [5]. The gap heat transfer model was applied for CFEM and DFEM. It is very important to note that applying the model to CFEM was created for this study only. It is not possible to generalize this technique for complex geometries. Figure 8 shows the volumetric average error versus simulation runtime for various uniform mesh refinements.

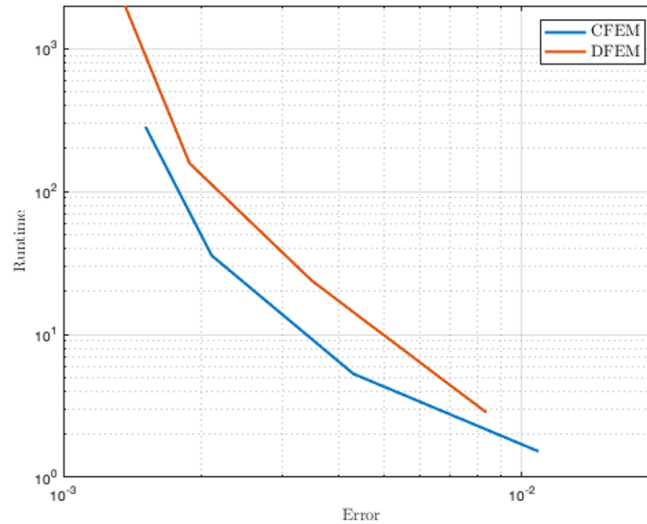Figure 8: Runtime performance comparison between CFEM and DFEM gap heat transfer

A highly refined DFEM simulation was used as reference for error calculation. Each simulation was run with the same number of processors, which is why DFEM appears slower for a given error. This is generally the case with smooth elliptic problems. However, DFEM runtime ranges from 1 to 4 times that of CFEM, despite having approximately 9 times

the number of DoFs. Consequently, using more processors will decrease the difference between the two methods.

To use DFEM with L_2 Lagrange basis and AMG preconditioner, this is what is needed in the variable and executioner blocks of the input file:

```
[Variables]
  [temperature]
    order = FIRST
    family = L2_LAGRANGE
  []
[]
```

```
[Executioner]
  type = Steady
  solve_type = 'PJFNK'
  petsc_options_iname = '-pc_type -pc_hypre_type'
  petsc_options_value = 'hypre boomeramg'
[]
```

3.2   Enhancements to GapHeatTransferInterfaceMaterial

To perform a full gap heat transfer model, considering conduction, convection, and radiation, the user is required to define five material properties:

1. Fluid conductivity (conduction)

2. Gap width (conduction)

3. Gap fluid temperature (convection)

4. Convective heat transfer coefficients (convection)

14

5. Surface emissivities/reflectivities (radiation).

Previously, the functionality of these material definitions was quite limited, as the user could only define them as a function of space. Additionally, the user needed to know which side of the sideset was the master and neighbor in order to define the surface properties (e.g., heat transfer coefficient, emissivity, etc.) properly. The GapHeatTransferInterfaceMaterial since has been substantially improved by allowing temperature dependent properties and block restricted surface properties.

The rest of the subsection will go over each of the possible configurations for a given problem. The example to showcase the functionality is a simple 1-D problem with a gap separating two blocks. For reference, the equation describing the heat transfer across each surface of the gap is given as:

$$
q''^{-}_{\text{gap}} = -k_{\text{gap}} \frac{T_+ - T_-}{\delta - \Delta d^+_{\text{gap}} - \Delta d^-_{\text{gap}}} + h^-_{\text{gap}}(T_- - T_{\text{bulk}}) + \frac{\varepsilon^- \sigma T^4_-(1-\rho^+) - \varepsilon^+ \sigma T^4_+(1-\rho^-)}{1 - \rho^- \rho^+},
$$
(2)

$$
q''^{+}_{\text{gap}} = -k_{\text{gap}} \frac{T_+ - T_-}{\delta - \Delta d^+_{\text{gap}} - \Delta d^-_{\text{gap}}} - h^+_{\text{gap}}(T_+ - T_{\text{bulk}}) + \frac{\varepsilon^- \sigma T^4_-(1-\rho^+) - \varepsilon^+ \sigma T^4_+(1-\rho^-)}{1 - \rho^- \rho^+},
$$
(3)

where δ can be defined as $\bar{d}_{\text{gap}} - \Delta d^+_{\text{gap}} - \Delta d^-_{\text{gap}}$. The corresponding options in GapHeatTransferInterfaceMaterial are defined in Table 4. Note that the convergence of the DFEM solver degrades significantly as the gap closes, the *min_gap_width* is provided to control the convergence.

3.2.1 Simplest form — constant properties

In this example of the material definition, the properties do not have dependence on temperature and each surface has the same properties:

```
[Materials]
  [gap]
    type = GapHeatTransferInterfaceMaterial
    boundary = 'interface0'
```

Table 4: Input parameter description for GapHeatTransferInterfaceMaterial

Input	Symbol	Units	Data Type	Dependence	Default
boundary	N/A	N/A	BoundaryName	N/A	None
conductivity	k_gap	W/m/K	Function	T_bulk	0
gap_temperature	T_bulk	K	VariableName	N/A	$\frac{T_+ + T_-}{2}$
gap_width	δ	m	VariableName	N/A	Empty
nominal_gap_width	$\bar{d}_\text{gap}$	m	Function	T_bulk	1
min_gap_width	$\bar{d}_\text{gap,min}$	m	Parameter	N/A	0
surface_blocks	N/A	N/A	SubdomainNames	N/A	Empty
surface_temperature	$T_\pm$	K	VariableNames	N/A	0
gap_displacement	$\Delta d_\text{gap}^\pm$	m	Functions	$T_\pm$	0
heat_transfer_coefficient	$h_\text{gap}^\pm$	W/m^2/K	Functions	$T_\pm$	0
emissivity	$\varepsilon^\pm$	N/A	Functions	$T_\pm$	0
reflectivity	$\rho^\pm$	N/A	Functions	$T_\pm$	0

```
    gap_temperature = Tbulk
    conductivity = 1.5
    nominal_gap_width = 1.0
    heat_transfer_coefficient = 1
    emissivity = 1
  []
[]
```

Here, Tbulk is an auxiliary variable which can be calculated using auxiliary kernels or set to a constant value. Note that since surface_blocks has not been defined, the properties are the same for each surface.

3.2.2 Gap temperature dependence

In this example, conductance properties are set to depend on the gap temperature (T_bulk):

```
[Materials]
  [gap]
    type = GapHeatTransferInterfaceMaterial
```

```
      boundary = 'interface0'
      gap_temperature = Tbulk
      conductivity = kgap
      nominal_gap_width = dgap
      heat_transfer_coefficient = 1
      emissivity = 1
    []
[]
```

`kgap` and `dgap` are defined in the Functions block of the input file.

3.2.3 Surface temperature dependence

In this example, the properties defined on the surface depend on the surface temperature
($T_\pm$):

```
[Materials]
  [gap]
    type = GapHeatTransferInterfaceMaterial
    boundary = 'interface0'
    gap_temperature = Tbulk
    conductivity = kgap
    nominal_gap_width = dgap
    surface_blocks = '0 1'
    surface_temperature = 'T T'
    gap_displacement = 'd1funct d2funct'
    heat_transfer_coefficient = 'h1funct h2funct'
    emissivity = 'eps1funct eps2funct'
    reflectivity = '0.0 0.0'
  []
[]
```

Since the `surface_blocks` parameter is set, all of the surface properties must be defined
for each of the listed blocks: `surface_temperature`, `gap_displacement`,

17

`heat_transfer_coefficient`, `emissivity`, and `reflectivity`. `surface_blocks` also defines the proper ordering of the properties, here, `d1funct`, `h1funct`, and `eps1funct` define the properties on subdomain 0, since each of these are the parameter's first entry, and subdomain 0 is the first entry of `surface_blocks`. All of the "funct" arguments are defined in the Functions block and depend on the surface temperatures prescribed by `surface_temperature`. Typically, `surface_temperature` will contain a list of the same variable, unless there are separate temperature variables in different blocks, which is uncommon.

3.2.4 Gap width as variable

In this example, the gap with (δ) is a variable, which can be calculated outside the gap heat transfer model by a mechanics calculations:

```
[Materials]
  [gap]
    type = GapHeatTransferInterfaceMaterial
    boundary = 'interface0'
    gap_temperature = Tbulk
    conductivity = kgap
    gap_width = gapw
    surface_blocks = '0 1'
    surface_temperature = 'T T'
    heat_transfer_coefficient = 'h1funct h2funct'
    emissivity = 'eps1funct eps2funct'
    reflectivity = '0.0 0.0'
  []
[]
```

Note that the user cannot define `nominal_gap_width` or `gap_displacement` if the `gap_width` parameter is set.

3.3 Modeling of heat pipe microreactors with hexagonal cladding

A set of tests have been developed to investigate an approach to modeling the thermal expansion phenomena in MOOSE without relying on a thermomechanical solution for the entire core. Section 3.3.1 introduces the basic geometry and the specific test problems. The deployment of the DFEM modeling approach requires a funcionalization of the displacement of the heat transfer surfaces as a function of surface temperature, which is described in Section 3.3.2. Finally, Section 3.3.3 contains the results and discussion.

3.3.1 Problem description

The geometry for the test problems consists of seven hexagonal fuel cells without the heat pipes, as shown in Figure 9. The failed heat pipes 1 and 7 are labeled in red. A support plate is modeled at the bottom of the geometry and each fuel cell includes a fuel region surrounded by clad. The reference mesh includes the explicit gap, whereas the mesh for the DFEM model is very similar, but with the gaps removed.
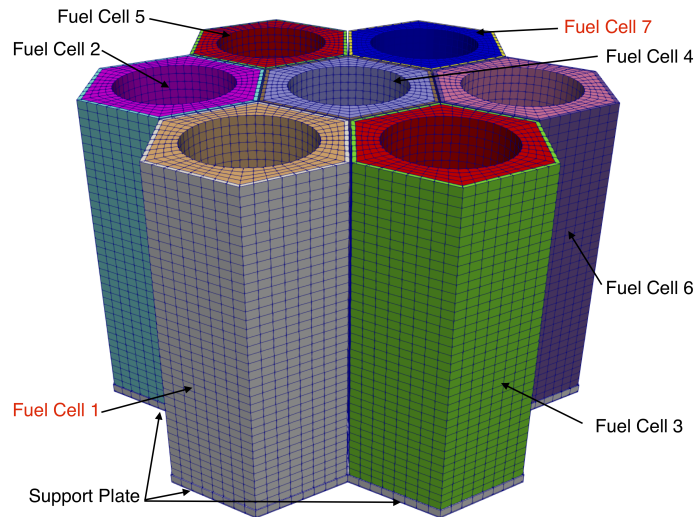

Figure 9: Reference mesh (1/10 axial scale)

A cosine-shaped heat generation profile is applied above the base plate in the fuel section to provide a heat source for the problem. A Robin (convective heat flux) boundary condition is imposed on the inner cylindrical surfaces of the model to emulate the heat removal by

the heat pipes. In addition to the failure of the heat pipes, an asymmetric temperature distribution is obtained by altering the heat transfer coefficient to each of the heat pipe bulk temperatures. The cladding surfaces facing other fuel cell clad surfaces are used in the gap heat transfer model. A Dirichlet boundary condition of 900.0 K is imposed on the rest of the cladding surfaces. The top and bottom surfaces use adiabatic boundary conditions. Material properties for 316 stainless steel are used throughout this model, including the fuel region. The Bison *SS316ThermalExpansionEigenstrain* and *SS316ElasticityTensor* material are used for the thermomechanics models.

The problem matrix is shown in Table 5. Problem 1a is a benchmark to ensure the approximated DFEM model has similar heat transfer characteristics as the reference problem. Problem 1b incorporates the thermal expansion of the gap surfaces without the expansion of the support plate. This is achieved with the addition of the *GapHeatTransferInterface-Material* and the functionalization of the gap distance as a function of temperature that is developed in Section 3.3.2. Problem 2a incorporates the displacement from the thermal expansion of the support plate and 2b adds the displacement from the thermal expansion of the gap surfaces. In problems 2a and 2b the DFEM solution is the MainApp, whereas the thermomechanics solution is performed in the SubApp. The temperature field from the DFEM solution was transferred to the thermomechanics. The x, y, and z displacements and the gap width (distance) are transferred from the SubApp back to the MainApp.

Table 5: Problem description for thermomechanics analysis

Problem	Description
1a	heat transfer
1b	thermomechanics - expansion of the cladding
2a	thermomechanics - expansion of the support plate
2b	thermomechanics - expansion of the cladding and support plate

The thermomechanics solution domain is limited to the support plate, depicted in Figure 10. All components of the displacement are set to zero at the bottom surfaces of the support plate to prevent translation and rotation of the solid body. A weight-bearing pressure is imposed on the top surface to include the effect of the weight of the heat pipe cells. The gap distance is computed with the *PenetrationAux* and *ParsedAux* kernels for a number of sidesets placed around each gap, shown in Figure 11. Penetration is considered to exist when the surfaces come into contact (i.e. when the gap closes). If the surfaces are not in contact, then a gap exists and the penetration variable reports a negative value. The gap size is, therefore, equal to the penetration value multiplied by negative one.

It is essential to note that the convergence of problems with gaps, either with the fully explicit reference or the DFEM models, relies on setting a minimum gap distance, which for these problems is set to 0.01 mm.
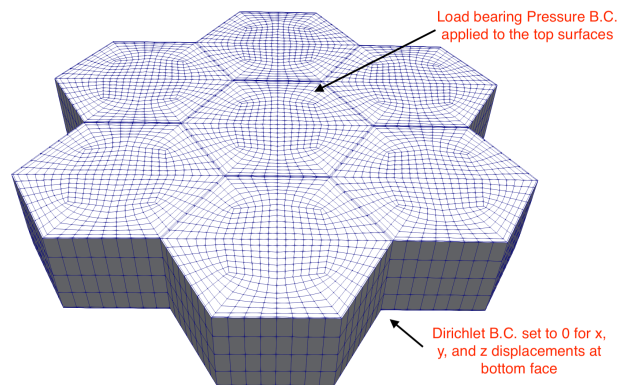

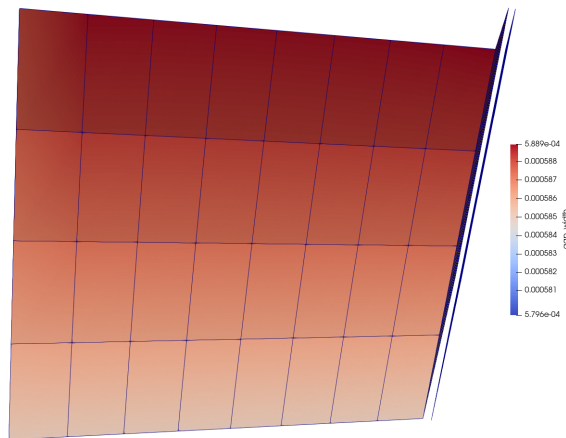Figure 10: Mesh and boundary conditions for the support plate SubApp.

Figure 11: Sidesets used in the calculation of the gap distance.

3.3.2 Functionalizing the gap distance

A dedicated model is developed to determine the displacement of the cladding surface as a function of surface temperature. The mesh shown in Figure 9 is modified by removing all the fuel regions, leaving just the cladding and support plate. Two Dirichlet boundary conditions are used to obtain a linear temperature distribution over the length of the geometry. All base plate surfaces (left) are fixed at the stress free temperature of 295 K, and the surfaces at the top of the geometry (right) are fixed at 1000 K. An adiabatic condition is imposed on the outer side surfaces. The temperature distribution is shown in Figure 12. The boundary conditions for the thermomechanics fix the displacement components at the bottom surface of the support plate (far left face).
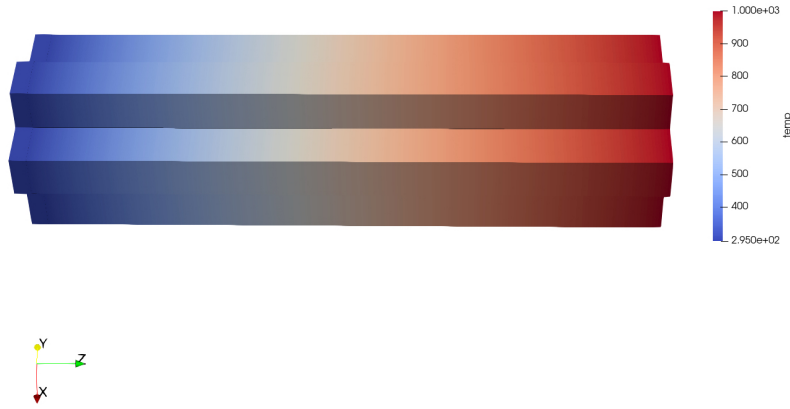

Figure 12: Temperature distribution used in the determination of the surface displacement as a function of temperature.

The gap distance is computed with the *PenetrationAux* and *ParsedAux* between two parallel clad surfaces along the length of the model. At the gap nearest to the support plate, the temperature is fixed at the stress-free temperature, thus there is no thermal expansion. A number of parameter values are outputted along the axial nodes of the edge of the clad surface using the *NodalValueSampler* postprocessor. The parameters include the temperature, penetration, gap width, and surface displacement (half-gap-width needed by *GapHeatTransferInterfaceMaterial*). The gap width and the surface displacement are shown in Figure 13. The negative gap size above 940 K indicated that the surface penetrated the volume on its opposite side. The function relating the surface displacement and temperature was obtained via a polynomial fit provided in Figure 13(b).
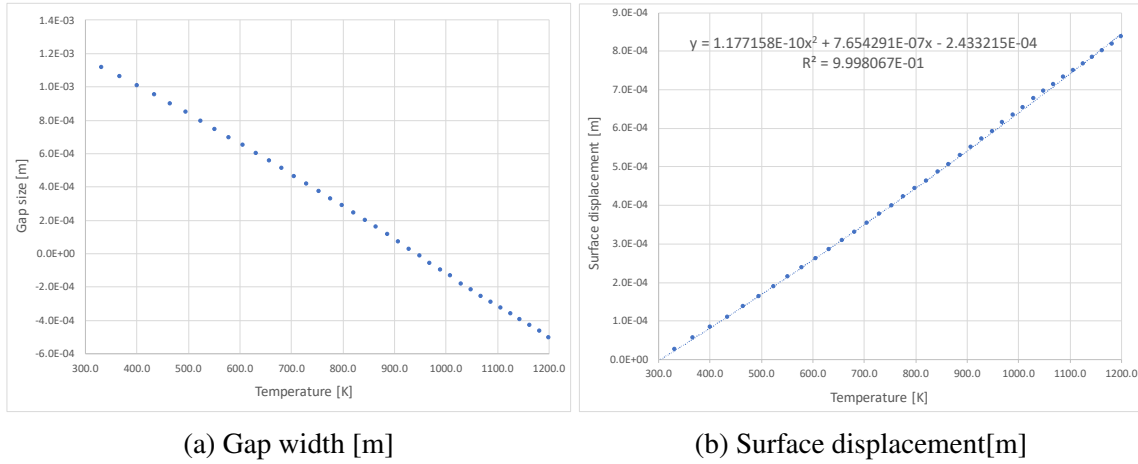
(a) Gap width [m]   (b) Surface displacement[m]

Figure 13: Behavior of the gap as a function of temperature.

3.3.3 Results

The results from the various tests are included in Tables 6 and 7. A case with no heat transfer across the gaps, *no HT*, is provided for comparison. The heat transfer solution from the DFEM 1a and the reference model are consistent within 0.1% on clad temperature average value and within 0.3% of the maximum value. The addition of the gap thermal expansion function in the DFEM model shows the average and maximum temperatures significantly drop in the failed fuel cells and increased temperatures in the neighboring fuel cells. The addition of the support plate thermal expansion decreases the temperature in the failed fuel cells and has minimal effect of the neighbours. Finally, modeling of both the gap and support plate thermal leads to same solutions that in problem 1b, which indicates that the minimum gap distance has been reached in multiple locations of the model. The reference calculation for problem 2b confirms that the cladding comes into contact at various locations, as shown in Figure 14. The reference model also indicates that the DFEM model over-predicts the gap closure. This leads to better than expected heat transfer with lower average (-0.3 %) and maximum (-1.0 %) temperatures in the failed fuel cells. This fact is further verified with an additional calculation with a 50% penalty on the surface displacement function. In these cases (1b* and 2b*) the expansion of the clad surface does not appear to reach the minimum gap distance and leads to different results. Improvements to both the gap width variable transfer and the functionalizaton of the surface displacement can further reduce these differences.

Table 6: Average clad temperatures.

Test	cell 1	cell 2	cell 3	cell 4	cell 5	cell 6	cell 7
no HT	1009.38	925.04	925.04	989.85	923.68	923.68	1009.37
ref 1a	964.49	932.99	932.99	1003.22	932.14	932.14	964.10
ref 2b	949.56	934.80	934.93	999.14	934.08	934.26	948.92
DFEM 1a	963.54	933.31	933.31	1003.36	932.47	932.47	963.14
DFEM 1b	946.85	937.09	937.09	1002.09	936.67	936.67	946.24
DFEM 2a	961.18	933.83	933.83	1003.32	933.05	933.05	960.75
DFEM 2b	946.81	937.10	937.10	1002.08	936.69	936.69	946.20
DFEM 1b*	958.51	934.38	934.38	1003.49	933.65	933.65	958.05
DFEM 2b*	951.47	935.94	935.94	1003.15	935.37	935.37	950.95

* gap thermal expansion decreased by 50%

Table 7: Maximum clad temperatures.

Test	cell 1	cell 2	cell 3	cell 4	cell 5	cell 6	cell 7
no HT	1245.10	983.18	983.18	1019.33	979.27	979.27	1245.10
ref 1a	1105.62	1019.58	1019.58	1044.43	1017.06	1017.06	1104.63
ref 2b	1063.10	1033.62	1033.59	1046.34	1031.67	1031.75	1061.56
DFEM 1a	1102.23	1020.91	1020.91	1044.94	1018.43	1018.43	1101.22
DFEM 1b	1052.56	1037.23	1037.23	1049.99	1035.55	1035.55	1051.14
DFEM 2a	1095.48	1023.13	1023.13	1045.57	1020.76	1020.76	1094.39
DFEM 2b	1052.56	1037.23	1037.23	1049.99	1035.54	1035.54	1051.13
DFEM 1b*	1086.60	1025.88	1025.88	1046.90	1023.62	1023.62	1085.48
DFEM 2b*	1064.05	1033.28	1033.28	1049.67	1031.37	1031.37	1062.83

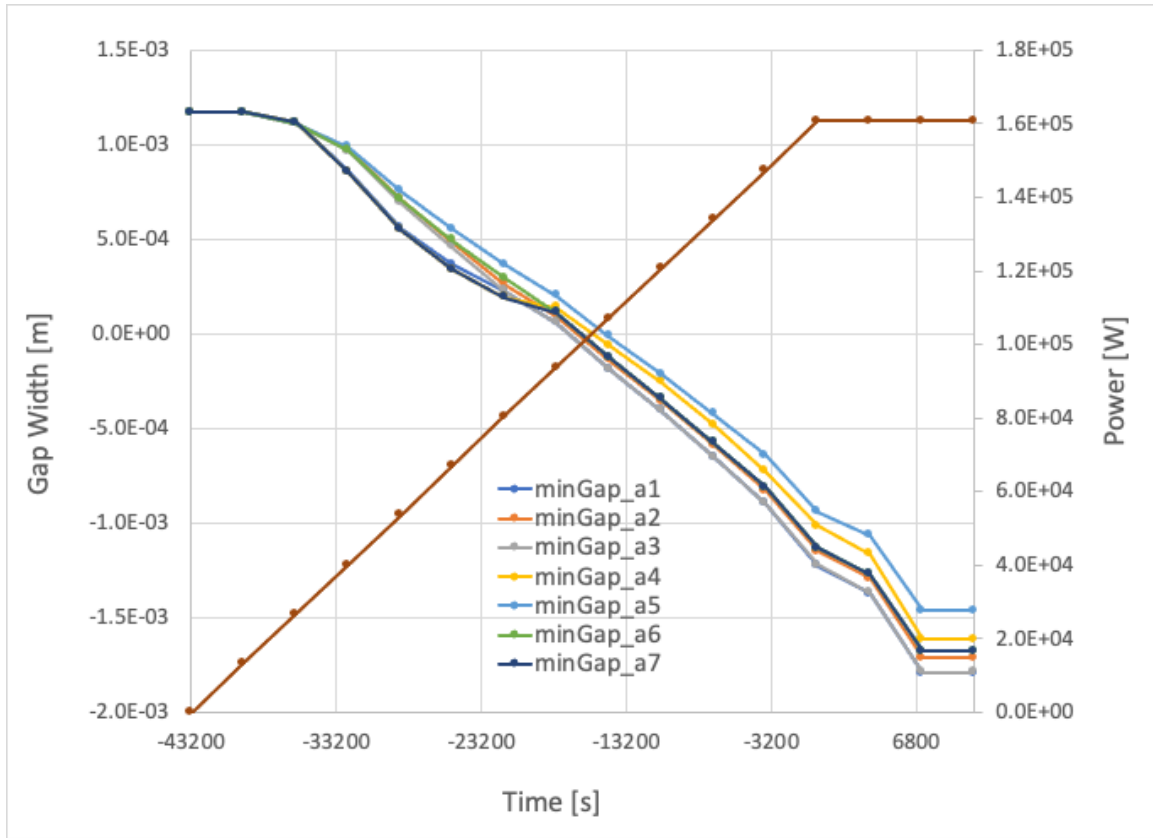* gap thermal expansion decreased by 50%

Figure 14: Closure of the gap in the reference calculation for problem 2b. Minimum gap is for all nodes in each cladding volume. Negative value of the gap width indicates penetration.

Results from DFEM problem 2b* are included in Figures 15 to 19. The support plate maximum displacements in x and y are roughly 1.23 mm, whereas it is 1.04 mm in the axial direction due to the weight loading. Figure 19 shows the gap with variable that is transferred from the SubApp. The maximum gap width value of 1.18 mm is correctly transferred to the bottom of the geometry, but is interpolated along the length of the domain and its value reduced by 40%. This is due to the interpolation transfer since the two domains do not align. Research by the MOOSE team is ongoing to improve the transfer of the gap width variable.
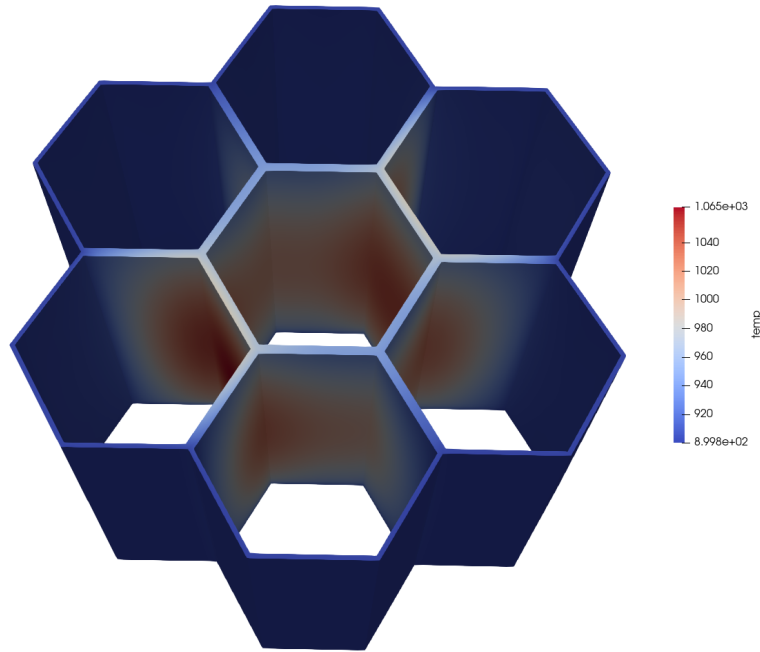

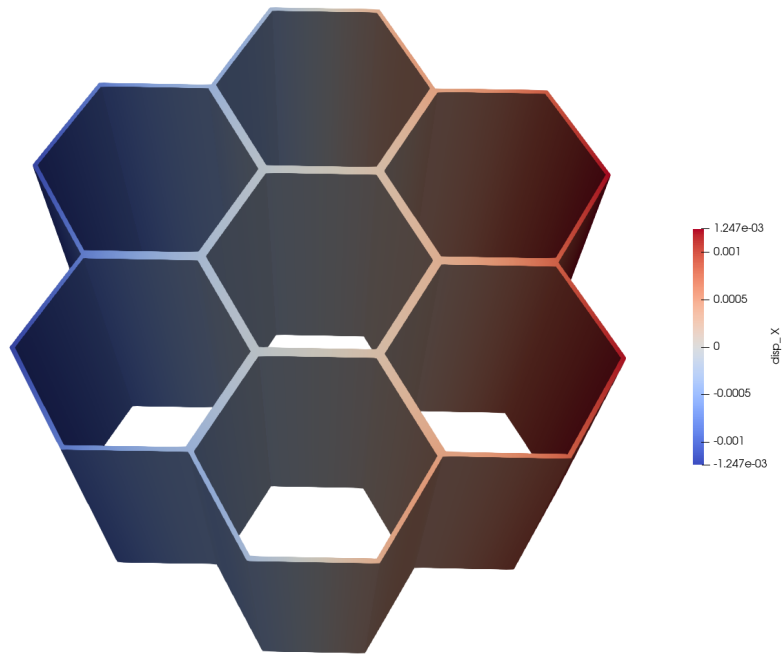Figure 15: Clad temperature [K] in DFEM 2b* model (1/10th scale in z).

Figure 16: Clad x displacement [m] in DFEM 2b* model transferred from the support plate themomechanics SubApp (1/10th scale in z).
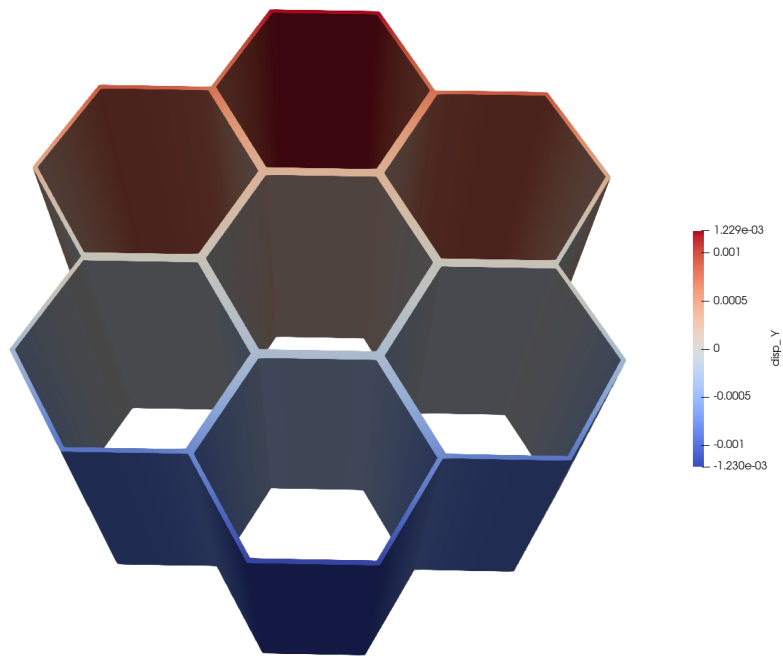
Figure 17: Clad y displacement [m] in DFEM 2b* model transferred from the support plate themomechanics SubApp (1/10th scale in z).
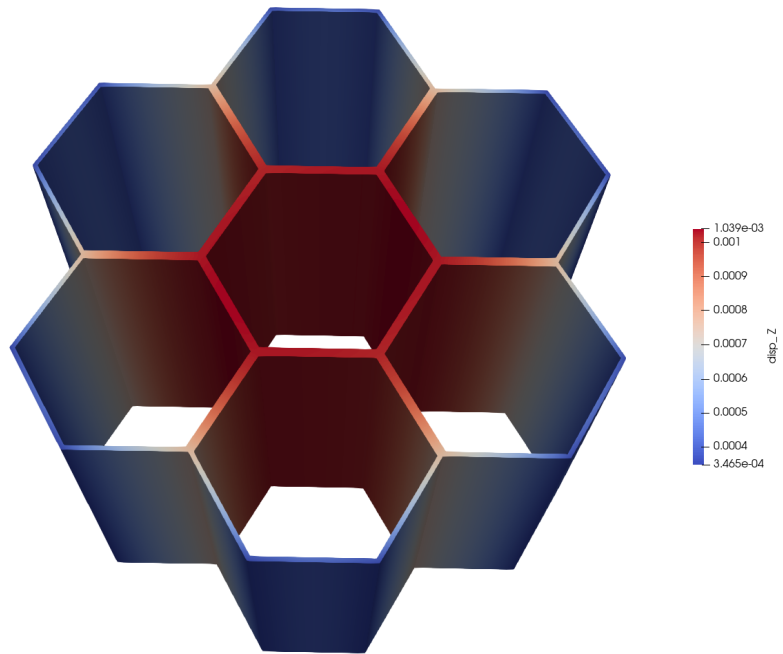
Figure 18: Clad z displacement [m] in DFEM 2b* model transferred from the support plate themomechanics SubApp (1/10th scale in z).
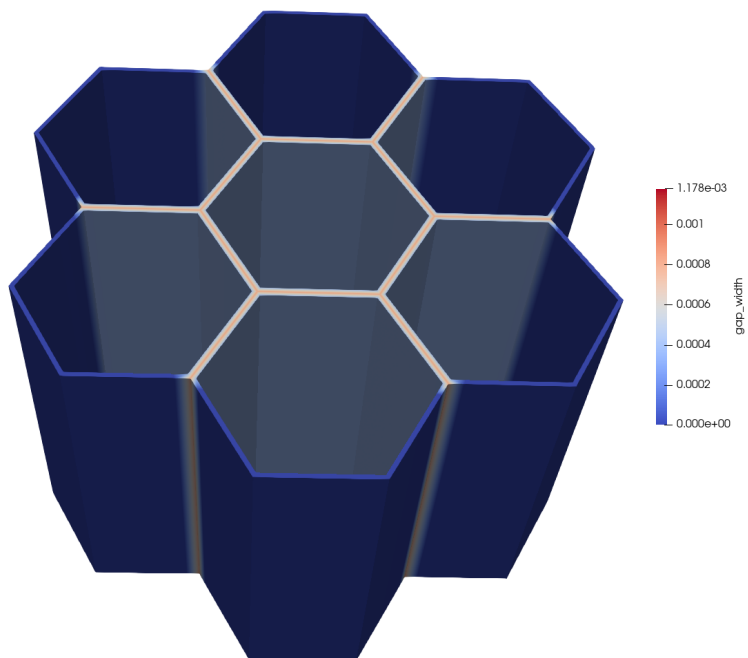
Figure 19: Gap width [m] (in DFEM 2b* model) transferred from the support plate themo-mechanics SubApp (1/10th scale in z).

4 Conclusions

This report details progress and activities of Idaho National Laboratory (INL) on the Nuclear Regulatory Commission (NRC) project "Development and Modeling Support for Advanced Non-Light Water Reactors."

We report the successful completion of one task of the NRC project "Development and Modeling Support for Advanced Non-Light Water Reactors." In addition, the extended scope tasks have also been completed. The following tasks' completion is reported:

- Task 2c: Explicit modeling of pebble transient temperature response. Several multiscale pebble coupling approaches are tested with one pebble per mesh element in the active core region. The results show good conservation behavior and stability of the coupling schemes.

- Extended scope part 1: An assessment of the computational efficiency of the DFEM heat transfer solver shows good scalability. The DFEM solver is a factor of 4 more expensive in solution time than the FEM solver for heat transfer problems due to the increased number of degrees of freedom. Nonetheless, the DFEM approach provides the user with the flexibility to model gap heat transfer problems.

- Extended scope part 2: The *GapHeatTransferInterfaceMaterial* was improved to give the user increased flexibility with the modeling of heat transfer through gaps with the DFEM solver. A number of gap parameters can now be coupled both through functions and variables.

- Extended scope part 3: Demonstration of how the gap width between hexagonal fuel cells can be calculated during a heat up transient and used in the *GapHeatTransferInterface* model. A full domain DFEM model with gap expansion is coupled to a SubApp that models the thermal expansion of the base plate. The results show the expected physical behavior, although they have not been fully benchmarked at this point in time.

References

[1] OECD/NEA. *PBMR Coupled Neutronics/Thermal-hydraulics Transient Benchmark The PBMR-400 Core Design*. Tech. rep. NEA/NSC/DOC(2013)10. OECD/NEA, July 2013.

[2] Paolo Balestra et al. "PBMR-400 benchmark solution of exercise 1 and 2 using the moose based applications: MAMMOTH, Pronghorn". In: Cambridge, UK: PHYSOR 2020, Transition to a scalable nuclear future, 2020.

[3] *Oklo Power Combined Operating License Application for the Aurora at INL: Part II, Final Safety Analysis Report*. ML20075 Project 99902046. Oklo Inc., Mar. 2020.

[4] J.W. Sterbentz et al. *Preliminary Assessment of Two Alternative Core Design Concepts for the Special Purpose Reactor*. INL report INL/EXT-17-43212, rev. 1. Idaho National Laboratory, May 2018.

[5] S. Schunert et al. *NRC Multiphysics Analysis Capability Deployment FY2020 - Part 1*. INL report INL/EXT-19-56762, rev. 0. Idaho National Laboratory, Dec. 2019.