INL/EXT-19-55892 Rev. 0

September 2019

Select Transient Metallic Fuel Bison Benchmarks

A. X. Zabriskie
K. D. Weaver

Idaho National Laboratory

NOTICE

This information was prepared as an account of work sponsored by an agency of the U.S. Government. Neither the U.S. Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for any third party's use, or the results of such use, of any information, apparatus, product, or process disclosed herein, or represents that its use by such third party would not infringe privately owned rights. The views expressed herein are not necessarily those of the U.S. Nuclear Regulatory Commission.

INL/EXT-19-55892 Rev. 0

Select Transient Metallic Fuel Bison Benchmarks

A. X. Zabriskie
K. D. Weaver

September 2019

Idaho National Laboratory
Fuel Modeling and Simulation Department
Idaho Falls, Idaho 83415

Prepared for the
U.S. Department of Energy
Office of Nuclear Energy
Under U.S. Department of Energy-Idaho Operations Office
Contract DE-AC07-05ID14517

Abstract

To explore the current capabilities of Bison with respect to transient metallic fuel modeling, four transient experiments are compared with their respective Bison model. The four experiments are the OPT-1 experiment of the Experimental Breeder Reactor II (EBR-II) and the M5, M6, and M7 experiments of the Transient Reactor Test facility (TREAT). Each benchmark is a 2D-RZ geometry of a single pin employing coupled multiphysics. Bison models of the benchmark agree well and illustrate areas needing development. The areas of suggested development deal with better material models, better coolant channel module for transients, proper treatment of material melting, improved feedback with failure models, and an ability to set initial conditions from post irradiation examination (PIE) data. Two Fast Flux Test Facility (FFTF) pins are also modeled with M7 transient conditions to illustrate Bison being applied to investigate potential experiments.

Contents

1 Introduction **1**

2 Benchmarks **1**

2.1 Benchmark of EBR-II 2

2.2 Benchmarks of TREAT 7

2.3 Benchmark Outcomes 13

3 Baselines **14**

4 Conclusion **18**

Acknowledgments **18**

References **20**

Appendix **21**

4.1 OPT-1 Benchmark Bison Input 21

4.2 M5 Benchmark Bison Input 32

4.3 M6 Benchmark Bison Input 44

4.4 M7 Benchmark Bison Input 56

1 Introduction

With the restart of the Transient Reactor Test facility (TREAT) [1] at Idaho National Laboratory (INL), transient testing, including metallic fuel types, may begin again. To increase the effectiveness of TREAT experimental campaigns, advance modeling and simulation tools not previously available are being developed and applied to guide and understand TREAT experiments. The fuel performance code Bison [2] is one such tool. Bison is built on a modular and multiphysics framework [3] and is currently in development and improvement.

As the interface between transient metallic testing and modeling in Bison is new, defining routes of information between modeling and experimentation provides motivation for collaborative efforts between models and experiments. The data provided from experiments could be consumed by models in setting initial conditions, correlation inputs, and validation. The results from modeling could show holes in data, improve experiment efficiency, and provide a source of data, which may not be measurable during an experiment. Exercises of modeling experiments shows which data Bison uses, what physics Bison models, and current code development needs.

To explore the current abilities of Bison, transient benchmark simulations of four historic experiments are compared with experiment data. The four experiments are:

1. *OPT-1:* Experiment X512 in EBR-II
2. *M5:* Transient M5-F1 in TREAT
3. *M6:* Transient M6 in TREAT
4. *M7:* Transient M7 in TREAT

The benchmarks show areas where Bison agrees with experimental data and areas for further development and investigation. The benchmarks are from two different reactors, Experimental Breeder Reactor II (EBR-II) and TREAT, and feature different metallic fuel materials in both the fuel slug and cladding.

To illustrate some unique capabilities of Bison, transient baseline simulations of Fast Flux Test Facility (FFTF) pins in TREAT transients are provided. The baseline simulations provide examples of predictive modeling, and alternate approaches to transient Bison boundary conditions.

A brief description of each reactor illustrates the modularity of Bison in simulating fuel types and forms from different reactors. Table 1 shows major differences between each reactor. TREAT fuel length and coolant varies due to experiments remaining separate from the reactor [4]. Irradiated experiments in TREAT contain fuel pins base-irradiated in other reactors.

Table 1: Reactor Features

Reactor	Capability	Fuel Length	Coolant
EBR-II	Base Irradiation Slow Ramp Transient	0.63 m	Sodium
TREAT	Fast Shaped Transient	Varies	Varies
FFTF	Base Irradiation	2.38 m	Sodium

2 Benchmarks

The benchmarks selected focused on transient experiments of used fuel pins that had previously been base-irradiated. Data from post irradiation examination (PIE) of pins or sister pins provides values for initial conditions of each pin going into transient experiments. Initial conditions of each Bison benchmark are taken as much as possible from this post base-irradiation PIE data.

Before comparing each benchmark to the corresponding experiment, an overview of the similarities and differences between each is provided in Table 2. Bison inputs (see appendix 4) are tailored to each experiment. An effort to keep the same coupled physics for each benchmark proved somewhat difficult and will be discussed in detail when discussing each benchmark.

Table 2: Benchmark Characteristics

Benchmark	Reactor	Fuel Type	Cladding	Coolant Channel	Peak Burnup
OPT-1	EBR-II	U-19Pu-10Zr	HT9	Triangular	11.1 at.%
M5	TREAT	U-19Pu-10Zr	D9	Annular Tube	0.8 at.%
M6	TREAT	U-19Pu-10Zr	D9	Annular Tube	1.9 at.%
M7	TREAT	U-10Zr	HT9	Annular Tube	2.9 at.%

2.1 Benchmark of EBR-II

The OPT-1 benchmark models a slow-ramp overpower transient in EBR-II designed to determine the performance capability of metallic fuel [5]. The specific fuel pin for this benchmark is DP-55. The fuel pin is base-irradiated in experiment X441 in EBR-II.

The OPT-1 benchmark couples tensor mechanics, heat transfer, and zirconium redistribution. Generated heat from fission conducts outward into the cladding where a ”coolant channel” boundary condition cools the cladding. Zirconium distribution in a fuel slug changes the fission power production profile. This results in a temperature profile, which drives zirconium redistribution. Zirconium concentration affects mechanical strength and thermal conductivity of a fuel slug.

The Bison model is a 2D-RZ single pin simulation geometry representing a fuel slug and cladding. The outer cladding boundary condition represents a coolant subchannel, including both pressure and convection. Table 3 provides geometric dimensions and their sources for this benchmark.

Table 3: OPT-1 Benchmark Geometry

Dimension	Value	Source
Slug Radius	2.545 mm	[6]
Slug Length	28.5 cm	[6]
Gap Width	Closed	[7]
Clad Width	0.381 mm	[6]
Plenum Length	58.85 cm	[6]

Initial conditions of the Bison model are determined from PIE data of sister pins to DP-55. Table 4 lists initial conditions and their respective sources. Material property sources are also listed.

As the model is a 2D-RZ geometry and symmetric about the center of the pin, Neumann boundaries at the center-line provide symmetric boundary conditions. Free-body-motion is restrained by anchoring the center-line bottom corner with zero displacement in all directions. No zirconium leaves or enters the fuel slug. An outer cladding boundary condition uses Bison's `CoolantChannel` representing sodium convection of heat. Table 5 provides the input for `CoolantChannel`.

The X512 experiment had a 24-hour preconditioning phase before a transient power ramp. The Bison model also simulates this 24-hour preconditioning phase providing time for zirconium to establish a proper profile. Zirconium is initially set to a uniform concentration throughout the fuel slug, which is not proper for an irradiated fuel pin. The power ramp occurs for 311.3 s from 43 $\mathrm{kW\,m^{-1}}$ to 56.4 $\mathrm{kW\,m^{-1}}$ before a SCRAM reduces power [7]. Figure 1 shows the transient portion of the power density profile for the OPT-1 benchmark.

The X512 experiment's PIE on pin DP-55 is currently being pursued. As such, the OPT-1 benchmark results are compared with the pre-experiment COBRA thermal analysis results [5].

Table 4: OPT-1 Benchmark Initial Conditions and Properties

Description	Value	Source
	Slug Characteristics	
Initial Zr	22.5 at.%	[6]
Initial Pu	16.3 at.%	[5]
D_{Zr}	`ZrDiffusivityUPuZr`	[8]
E	`UPuZrElasticityTensor`	[8]
ν	`UPuZrElasticityTensor`	[8]
α_{th}	1.18e−5 K^{-1}	[8]
ρ	15 800 $\mathrm{kg\,m}^{-3}$	[8]
k_{th}	`ThermalUPuZr`	[8]
c_p	`ThermalUPuZr`	[8]
	Cladding Characteristics	
E	188 MPa	[8]
ν	0.236	[8]
α_{th}	1.2e−5 K^{-1}	[8]
ρ	7874 $\mathrm{kg\,m}^{-3}$	[8]
k_{th}	`ThermalHT9`	[8]
c_p	`ThermalHT9`	[8]
Bond Sodium k_{th}	62.9 $\mathrm{W\,m}^{-1}\,\mathrm{K}^{-1}$	[9]

Table 5: OPT-1 Benchmark `CoolantChannel` Inputs

Input	Value	Source
Coolant Material	Sodium	[5]
Subchannel Geometry	Triangular	[5]
T_{inlet}	642 K	[7]
P_{inlet}	0.439 MPa	[10]
$\dot{m}_{inlet}$	4173 $\mathrm{kg\,m}^{-2}\,\mathrm{s}^{-1}$	[5]
Rod Pitch	6.92 mm	[5]

The COBRA analysis provides predicted pre-transient and peak transient temperatures for each pin in the experiment, including DP-55. Peak temperatures from COBRA at multiple locations are compared in figure 2.

Further results from the OPT-1 benchmark include profilometry of the outer cladding. However, this profilometry is a ”delta” profilometry as the initial profilometry of the cladding is set as flat. Bison is currently unable to assign a profilometry to a cladding boundary condition taken from PIE data. This is for good reason as the task is not trivial.

PIE measurements for profilometry determine radial displacement of the outer edge of cladding for a pin. The measurement does not provide any information on a displacement profile radially inward from a measured outer edge. Bison needs some method to calculate inner displacements to provide a realistic stress state in a model as an initial condition if a PIE profilometry is used to set initial cladding profilometry.

A popular option to overcome this is to use a simulation of a base-irradiation to set initial conditions for a transient. This approach forgoes any information from PIE on an irradiated pin. However, if simulation results match PIE data for a base-irradiation, then this is a good option, but requires extra modeling efforts.

With the OPT-1 benchmark attempting to only use PIE data, the maximum radial displacement from cladding with an initial zero displacement is 1.07×10^{-2} mm at 85 % of the slug height

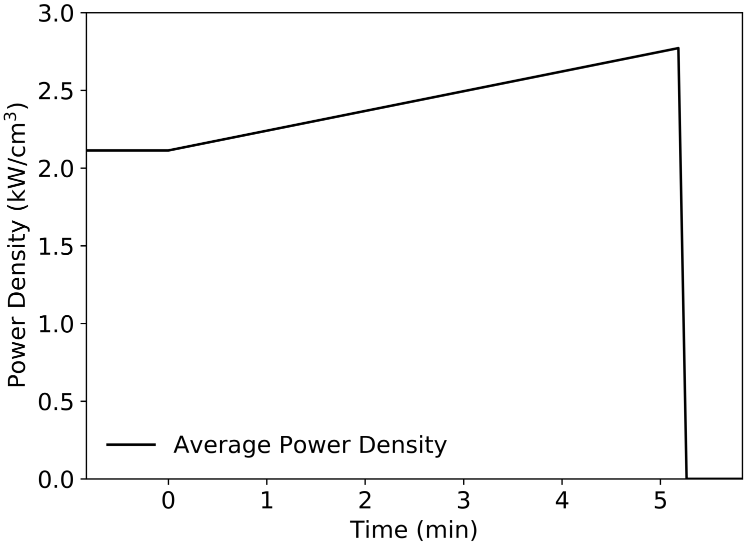


Figure 1: Power density profile for transient portion of the OPT-1 benchmark.

during the transient peak. At the same location during the transient peak, the axial displacement is 0.502 mm. The contact between the slug and cladding is approximated as "glued" which does not allow slip or separation. Figure 3 shows the radial and axial displacement profiles at the transient peak.

The OPT-1 benchark has an initially flat zirconium distribution that established a profile during the 24 h preconditioning phase. This zirconium profile is unaffected during the transient as shown in figure 4, where the radial axis is scaled 20× and the plenum and its cladding is not shown.

Even though the X512 experiment did not cause any pin's to fail, the slow (compared to the other transient benchmarks) ramp transient of X512 lets the OPT-1 benchmark include everything Bison has to offer for longer base-irradiation (steady-state) simulations. With multiphysics coupling, a cumulative damage fraction (CDF) failure correlation and eutectic thickness correlation are included in the input [8]. As the failure and eutectic options are relatively new in Bison and still under development, a non-failing test without proper initial conditions (base-irradiation accumulated damage and initial eutectic thickness) for failure models provides no value. The inclusion does show if the pieces work together and if erroneous results are produced, i.e., failure indicated when no failure occurred.

The eutectic thickness correlation shows no increase in eutectic thickness during either preconditioning or transient phases. This is somewhat expected as temperatures predicted by both COBRA and Bison (see figure 2) indicate the cladding remains below 900 K. Eutectic formation and melting is thought to begin near 988 K [11]. The eutectic thickness value calculated in Bison does not currently provide feedback affecting the solution, e.g., thinning a cladding providing less mechanical strength.

The CDF correlation also shows no failure. Peak CDF values occur in the cladding at 72.1 % of the fuel slug length. The preconditioning phase accumulates 0.202 CDF at that location and the transient increases the value to 0.233. Failure occurs when the CDF is 1 or greater. These values are based off an initial CDF value of zero, which is likely not proper as base irradiation does increase CDF from zero.

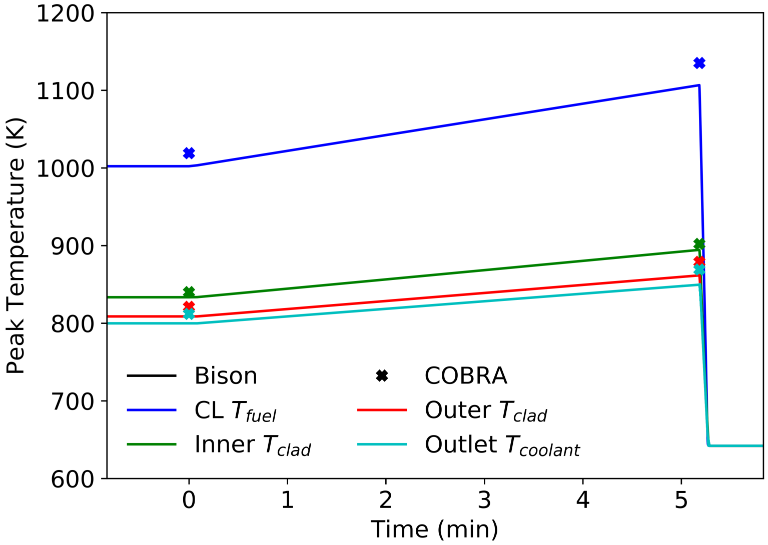


Figure 2: Peak temperature comparison of COBRA and Bison for OPT-1.

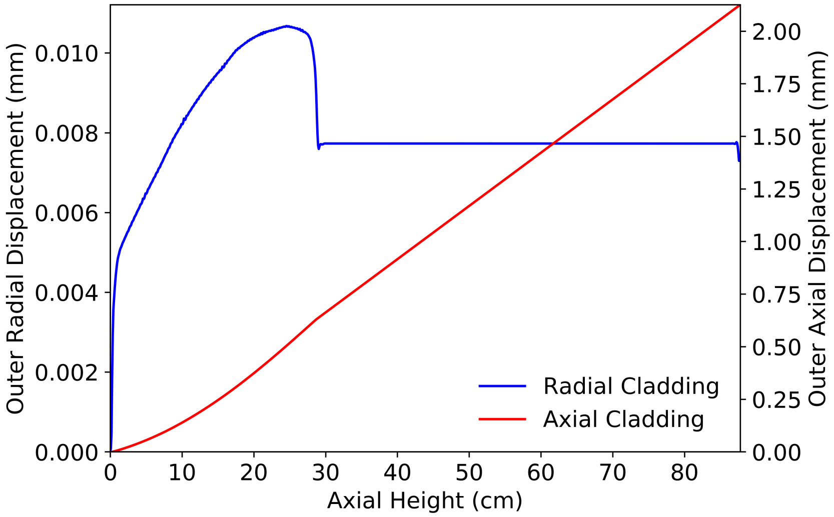
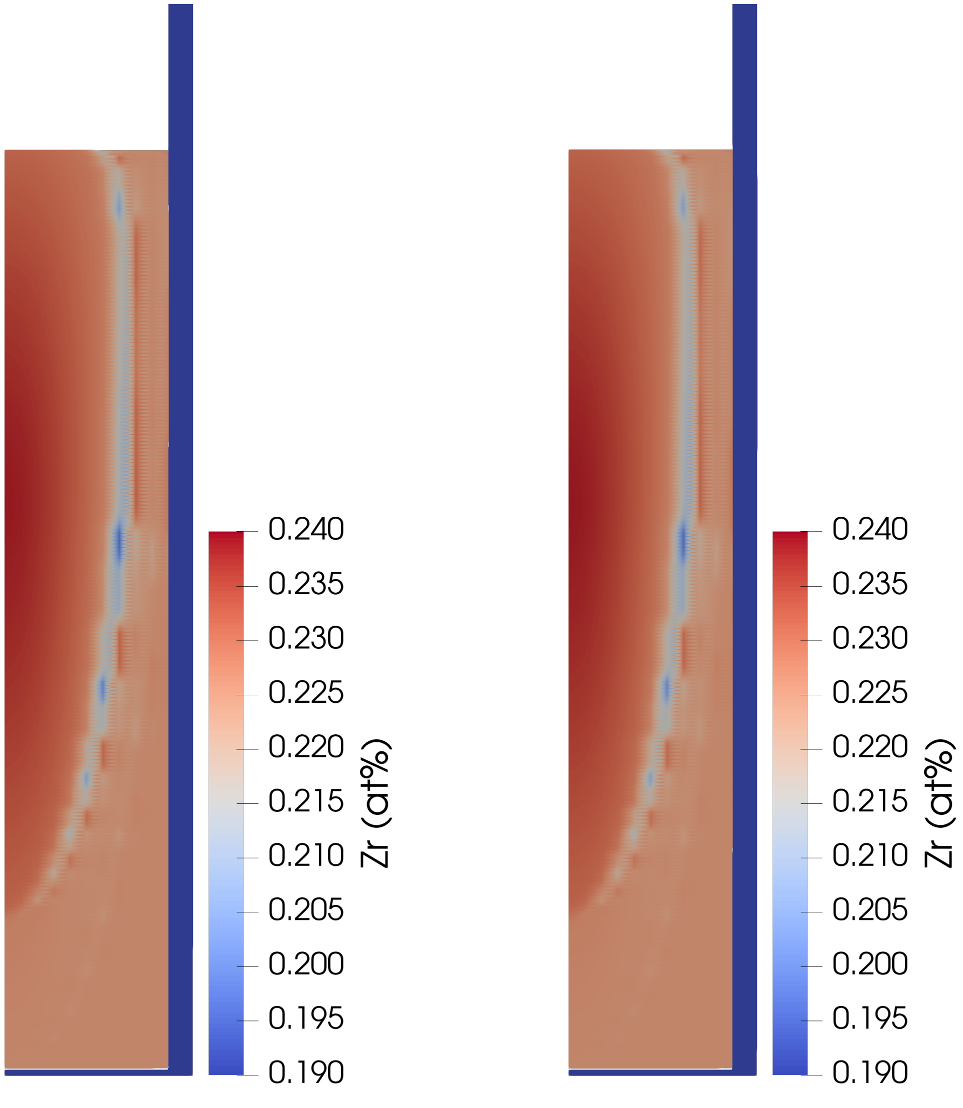


Figure 3: Outer cladding displacements at the transient peak.



(a)

(b)

Figure 4: No change in zirconium distribution from 24-hour preconditioning profile (a) to transient peak profile (b).

2.2 Benchmarks of TREAT

Three benchmarks of un-failed pins in TREAT model shaped transients meant to melt metallic fuel. The TREAT experiments M5, M6, and M7 [12] are modeled in Bison with the same coupled physics. As with the OPT-1 benchmark, initial conditions are taken from PIE data as much as possible.

The M5 benchmark models pin T-280 from the X419A experiment of EBR-II. The M6 benchmark models pin T-186 from the X419 experiment of EBR-II. The M7 benchmark models pin T-427 from the X425 experiment of EBR-II. Each pin's initial conditions in the Bison model are set to conditions in TREAT and not to conditions at the end of each EBR-II experiment.

All the TREAT benchmarks couple tensor mechanics and heat transfer; they treat zirconium distribution as uniform and constant. A "coolant channel" boundary condition representing a single pin in a flow channel annular tube configuration for each TREAT experiment is included as some heat is removed during the transients. Material strength and properties are temperature dependent.

All benchmarks use a Bison 2D-RZ single pin geometry with the fuel slug in initial contact with cladding using a frictionless contact boundary condition between the two. Table 6 provides geometric dimensions and their source. Table 7 lists initial conditions and material properties for each benchmark.

Table 6: TREAT Benchmark Geometries

Dimension	M5	M6	M7	Source
Slug Radius (mm)	2.54	2.54	2.54	[12]
Slug Length (cm)	35.1	34.7	37.3	[12]
Gap Width	Closed	Closed	Closed	[12]
Clad Width (mm)	0.38	0.38	0.38	[12]
Plenum Length (cm)	24.4	25.24	25.24	[12]

Neumann boundaries at the fuel slug center-line provide symmetric boundary conditions. Free-body-motion is restrained by anchoring the center-line bottom corner of the fuel slug and cladding to experience zero displacement in any direction. An outer cladding boundary condition uses Bison's `CoolantChannel` to correlate heat transfer from a tube inside another tube [14]. Table 8 provides input for `CoolantChannel`.

The M5, M6, and M7 TREAT experiments all use shaped transients. Unlike the OPT-1 benchmark, these TREAT benchmark models do not simulate a preconditioning phase, but start when TREAT data starts. Average power densities for the TREAT benchmarks are provided in figure 5.

All M-series experiments are designed to melt some portion of fuel in a pin [12]. This melting is troubling for Bison because it does not currently contain the capabilities to model molten metallic fuel. Bison does use temperature-dependent properties, but does not change properties when material melts. Currently, Bison may be set to limit property evaluation when temperatures rise above a threshold for a material. This allows simulation to progress beyond melt, but fidelity is sacrificed as no feedback from melted material affects solution results.

Temperature thermocouple data is available from each experiment. Each thermalcouple measures the temperature of a thin-walled flowtube, effectively measuring bulk coolant temperature, T_∞. Bison's `CoolantChannel` has the ability to calculated T_∞ and the convection heat transfer coefficient (HTC). Even though coolant temperature measurements along the length of an experiment pin are available, the benchmarks are meant to test predictive capabilities in Bison. As such, thermocouple data taken during an experiment will not be used as a boundary condition, though Bison does have this capability.

`CoolantChannel` is designed for longer time scale simulations, e.g., steady-state and base-irradiation simulations. As such, TREAT transients are technically outside the original intent.

Table 7: TREAT Benchmark Initial Conditions and Properties

Description	M5	M6	M7	Source
Slug Characteristics				
Zr (at.%)	24.55	25.08	22.00	[11]
Pu (at.%)	15.61	15.27	0.0969	[11]
Porosity	0.27	0.27	0.31	[11]
$T_{liquidus}$ (K)	1515	1515	1600	[11]
E	`UPuZrElasticityTensor`			[11]
ν	`UPuZrElasticityTensor`			[11]
α_{th} (K^{-1})	1.18e-5	1.18e-5	1.18e-5	[8]
ρ (kg m^{-3})	15400	15500	15700	[8]
k_{th}	`ThermalUPuZr`			[8]
c_p	`ThermalUPuZr`			[8]
Cladding Characteristics				
E (MPa)	$f(T)$	181	188	[13] [13] [8]
ν	$f(T)$	0.313	0.236	[13] [13] [8]
α_{th} (K^{-1})	1.61e-5	1.62e-5	1.2e-5	[13] [13] [8]
ρ (kg m^{-3})	7851	7847	7874	[13] [13] [8]
k_{th}	`ThermalD9`		`ThermalHT9`	[8]
c_p	`ThermalD9`		`ThermalHT9`	[8]
Bond Sodium Characteristics				
k_{th} (W m^{-1} K^{-1})	73.7	73.7	73.7	[9]

This is manifest in instability during simulation solves making the choice of time step critical. A comparison of M5, M6, and M7 benchmarks to experiment data for coolant temperature is shown in figure 6. COBRA-PI calculated values post-experiment are also provided [12].

The comparison is not ideal as each thermalcouple, COBRA-PI, and Bison benchmark value is not located at the same elevation. The COBRA-PI value is from the closest calculated value to each thermocouple. Benchmark values are peak values likely originating from just above the fuel slug, which is at a lower elevation from each thermocouple,

Even though Bison does not account for melting and zirconium distribution is not modeled in these TREAT benchmarks, an attempt to compare fuel melt profiles is made in figure 7 by post-processing a temperature profile and flagging cells red that have surpassed $T_{liquidus}$ of table 7. Overlayed on this profile in figure 7 is PIE solidus measurement data and calculated COBRA-PI results.

Bison does not account for the heat of fusion. As such, temperature values rise as if a material does not melt. The larger melt zone of each benchmark of figure 7 reflects this. The uniform zirconium distribution of the TREAT benchmarks does not represent burned fuel whose melt temperature is affected by concentration of zirconium in a region of material. Including both a proper zirconium distribution and heat of fusion (or melting physics) would increase Bison's fidelity.

Profilometry data is available for only the M7 experiment. The data is a post-transient measurement with no comparison to the pre-transient data. The profile exhibits an interesting shape as there is a slight reduction in radius before a large increase in radius until the plenum beings. The M7 benchmark once again has an initial displacement as a uniform zero. As figure 8 indicates, the final M7 benchmark displacements, though small in magnitude, have a reduction region before the large increase region also present in the experimental measurement.

The maximum axial expansion of fuel slugs compared to each benchmark's axial expansion is presented in table 9. Currently, the axial growth of metallic slugs in Bison is transitioning to a new method informed by lower-length scale modeling. These benchmark results do not use this

Table 8: TREAT Benchmarks `CoolantChannel` Inputs

Input	M5	M6	M7	Source
Coolant Material	Sodium	Sodium	Sodium	[12]
Correlation	Seban	Seban	Seban	[8]
T_{inlet} (K)	573	583	586	[12]
P_{inlet} (kPa)	413.7	413.7	455.1	[12]
$\dot{m}_{inlet}$ ($\mathrm{kg\,m^{-2}\,s^{-1}}$)	3761	3761	4521	[12]
Flow Area (cm)	0.222	0.222	0.222	[12]
D_{hydr} (mm)	2.057	2.057	2.057	[12]
Heated Perimeter (mm)	18.35	18.35	18.35	[12]
D_{heat} (mm)	4.84	4.84	4.84	[12]

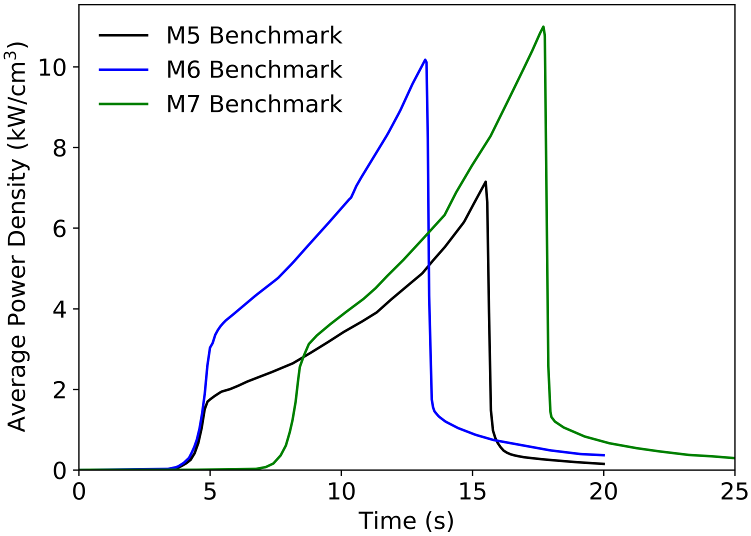


Figure 5: Power density profile for M5, M6, and M7 benchmarks.

new method as it is in heavy development.

The TREAT benchmarks feature non-failed pins, but these pins were brought close to failure [12]. As such, eutectic thickness should not be zero as the experiments clearly exceeded the eutectic melt temperature. As such, Bison's eutectic correlation shows that each benchmark's peak attack on cladding is located at the top of the fuel slug.

As indicated by eutectic thicknesses greater than cladding thickness, the simple temperature correlation in Bison predicts failure in M6 and M7 benchmarks when the experiments did not fail. In contrast, a Bison HT9 CDF failure model using short Dorn parameters [8] predicts no failure in the M7 benchmark. There is currently no failure model for D9 cladding in Bison.

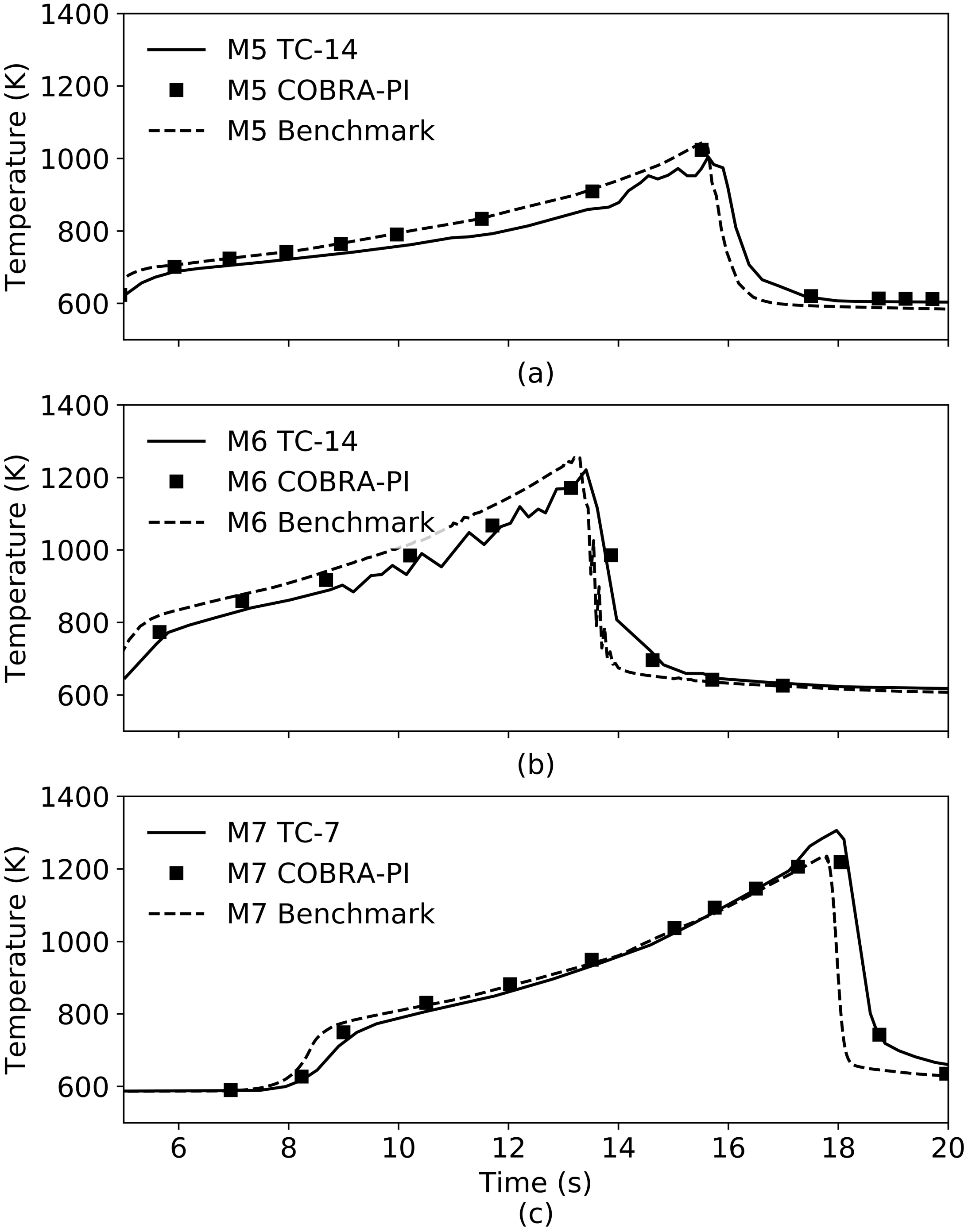


Figure 6: Comparison of benchmark peak coolant temperatures with measured flow tube temperatures for M5 (a), M6 (b), and M7 (c).

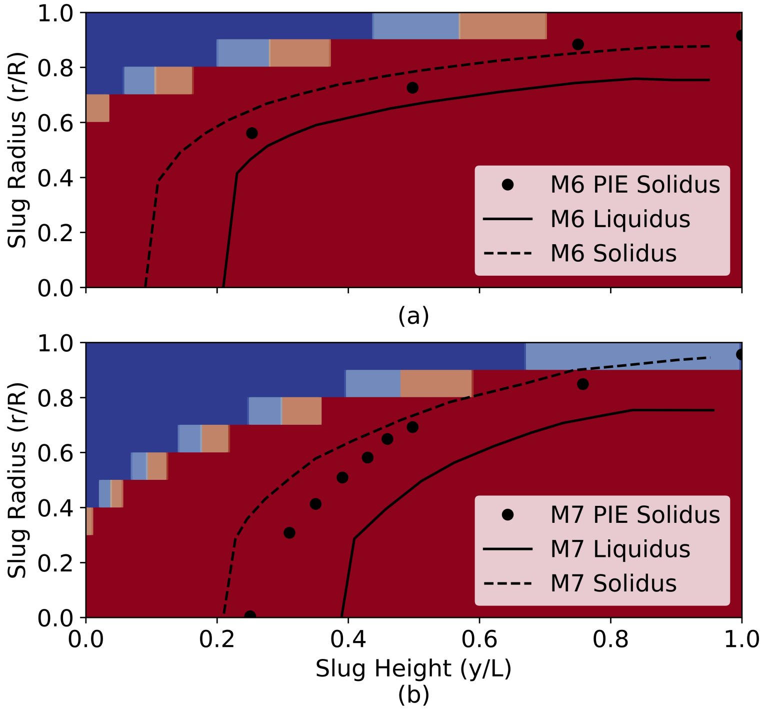


Figure 7: Benchmark melt (red) profile for M6 (a) and M7 (b) compared with PIE data and COBRA-PI calculated values.

Table 9: Axial Slug Expansion Comparison

Name	Experiment (%)	Benchmark (%)
M5	1	0.04
M6	2 - 4	0.09
M7	2 - 4	1.34

Table 10: TREAT Benchmark Eutectic Thickness Results

Benchmark	Thickness (μm)	Wastage (%)
M5	0.33	0.087
M6	710	187
M7	530	139

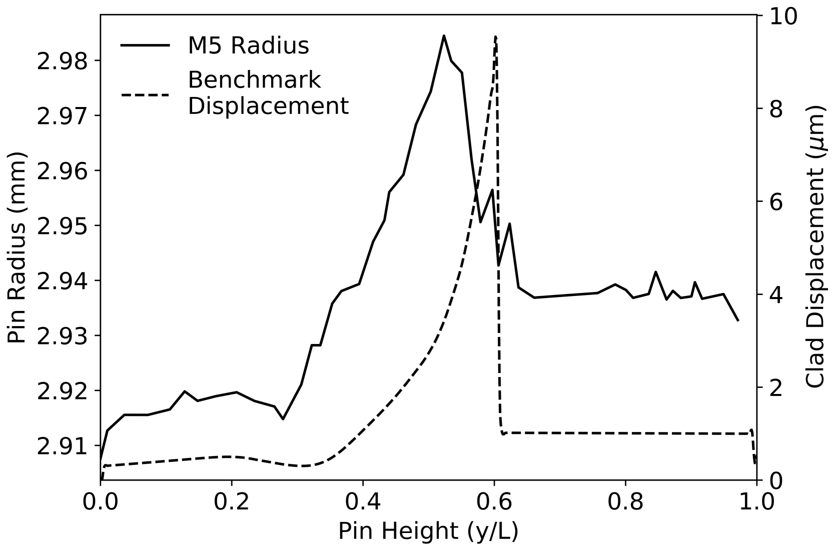


Figure 8: M7 experiment profilometry and benchmark displacements along pin.

2.3 Benchmark Outcomes

A unique aspect of Bison is that it is under development and is a modular simulation package. Bison's modularity allows users flexibility to enable or restrict physics and methods for simulations. This modularity also enables developers to address issues effectively and improve aspects of Bison that are lacking. As such, benchmarks are a tool to change the "current" state of Bison identifying areas needing improvement to be quickly developed.

The OPT-1 benchmark is in good agreement with pre-test results from COBRA. The benchmark proved that modules and methods developed for base-irradiation modeling could be used in a transient. The benchmark also illustrated that pre-conditioning time steps may provide good initial conditions for a transient much the same way a long inlet area provides fully-developed flow in computational fluid dynamics.

The OPT-1 benchmark shows Bison has the ground work already in place to model metallic fuel. Multiple pathways of interacting feedback between three different governing physics are present in the OPT-1 benchmark. Heat transport and tensor mechanics are joined by mass transport with each one providing feedback to the others, though zirconium redistribution is still in early development. Failure models, also in early development, are included in the OPT-1 benchmark even though no feedback occurs.

Of great concern is a proper convection boundary condition. The current coolant channel model proves it can handle the transient of OPT-1. Further investigation into Bison's manual indicates the coolant channel model is also used in water cooled reactors. Sodium coolant has been added making use of already existing convection heat transfer code developed for water coolant. Transients with water coolant have found success using Bison's coolant channel model [15].

Another concern from the OPT-1 benchmark is proper initial conditions taken directly from PIE data. Bison's ability to set initial displacements to match profilometry data has already been identified as an opportunity for improvement. Another opportunity could be using PIE cross sections image to form an initial zirconium distribution profile. Bison does have the ability to set initial profiles radially and axially. With this ability to set the profile and information from PIE cross sections, interpolation may fill in needed data for initial profiles between PIE cross sections.

The TREAT benchmarks challenge Bison's transient metallic fuel modeling capabilities with faster and stronger transients. The transient time scale reduces from minutes in OPT-1 to seconds in the M-series, having an effective energy deposition rate of 14 kW to 35 kW. The TREAT benchmarks enter fuel melting temperatures, which is a capability Bison does not have nor is currently planned to have. To proceed, an added limit to material property evaluation allows simulations to progress past melting.

Along with improvements suggested from the OPT-1 benchmark, the TREAT benchmarks identified a few more areas for possible improvement. The first is a proper "short" transient coolant channel model. Bison does have other convection boundary conditions, but they require knowing HTC and T_∞ values along a cladding. The coolant channel method requires inlet conditions and calculates convection conditions up the cladding from an inlet. The TREAT benchmarks show the current coolant channel model may become unstable producing oscillations in temperature if proper time step sizes are not provided. Some oscillations sill remain and could not be removed and are visible in the M6 benchmark, see figure 6(b).

The TREAT benchmarks are made of varying combinations of materials. Bison has modules for both binary and ternary metallic fuel thermal and mechanical properties using what data is available for composition effects. Bison has modules for thermal and mechanical properties of HT9 cladding. Bison has a module for thermal properties of D9 cladding. The M5 and M6 (see table 7) benchmarks show that even without a proper mechanical properties Bison module, constant values (M6) and temperature-dependent properties (M5) may be input readily into Bison. However, development of proper material property modules in Bison is preferred and a suggested improvement.

The M6 and M7 benchmarks also show that eutectic thickness correlations based solely on temperature are not adequate enough. Having a eutectic form and then thin cladding to provide feedback into the mechanics solution is a desirable improvement. Cladding failure with CDF is a historic correlation and currently only available for HT9 cladding. Other methods for determining cladding failure specific to transient conditions [16] may be added to Bison to improve failure predictions and move away from correlated failure models. Bison's current failure models, including eutectic penetration, aim to locate a site of failure along with a time of failure.

Overall, without calibrating any of Bison's model inputs to experimental data, all the benchmarks show promise in providing predictive information useful to experimentalists. The comparison of experiments to benchmarks highlight areas of possible improvement. Such a process improves both experiment efficiency and modeling fidelity.

3 Baselines

Transient testing of FFTF pins in TREAT has never been done. As FFTF pins are longer than EBR-II pins, transient performance differences between FFTF and EBR-II pins is an unanswered question. To illustrate Bison's relevance, a baseline attempting to answer this question is presented.

The FFTF baseline imposes the M7 benchmark boundary conditions and power profile onto two FFTF pins from MFF-3 and MFF-5 experiment tests [17]. The baseline attempts to preserve power density and coolant conditions. As such, more energy is produced by each MFF pin as their fuel slug is longer than M7's fuel slug.

Pin 193045 is chosen from MFF-3 and pin 195011 is chosen from MFF-5. Both are binary U-10Zr metallic fuel forms with HT9 cladding, similar to the M7 benchmark. The peak burnup for 193045 is 13.8 at.% and for 195011 is 10.1 at.%. Both are higher than the burnup for the M7 benchmark.

As the MFF-3 and MFF-5 inputs for Bison are based off the M7 benchmark, their is little difference between the inputs. The differences are mainly in the geometry of each pin, provided in Table 11, and that the baseline uses `ConvectiveFluxFunction` as an outer cladding boundary condition. This replaces the `CoolantChannel` of the M7 benchmark, but HTC and T_∞ values from M7 are dimensionally normalized and applied in the FFTF baseline of MFF-3 and MFF-5.

Table 11: FFTF Baseline Geometry

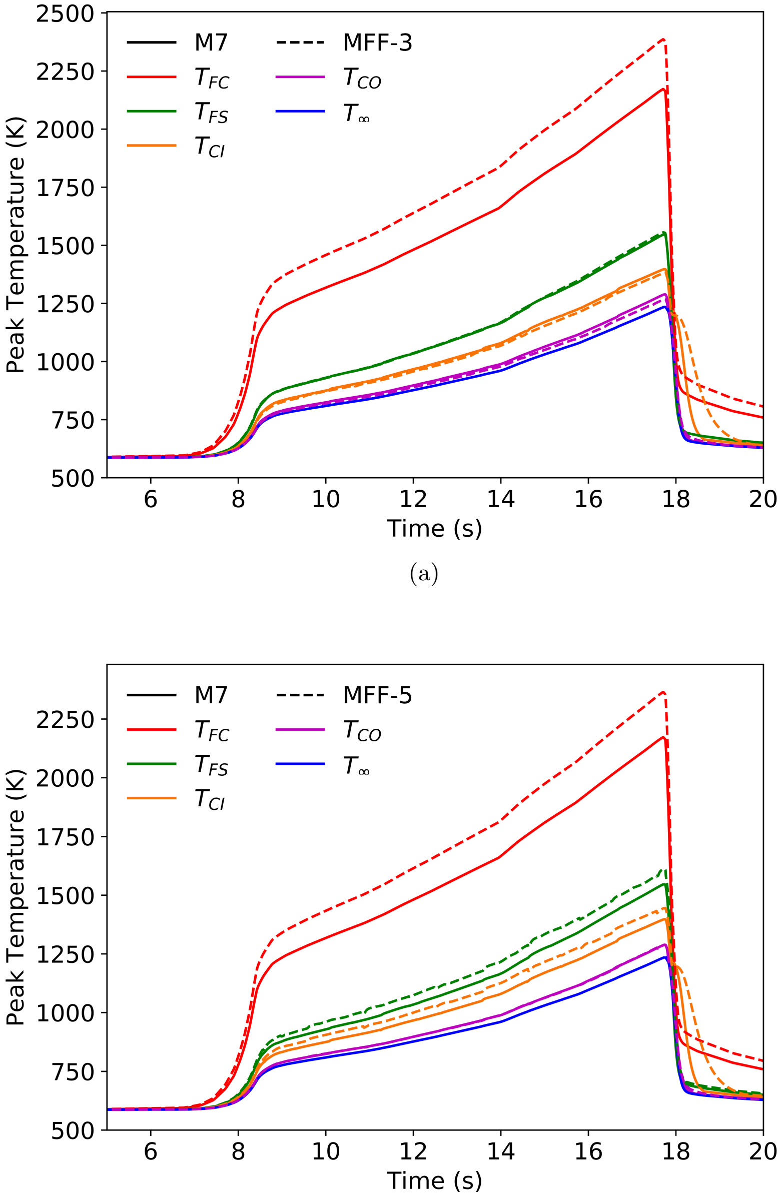
Description	MFF-3	MFF-5	Source
Slug Radius (mm)	2.87	2.87	[17]
Slug Length (cm)	92.8	94.8	[17]
Gap Width	Closed	Closed	[17]
Clad Width (mm)	0.559	0.559	[17]
Plenum Length (cm)	1.484	1.617	[17]
Zr (at.%)	25.4	19.5	[17]
Pu (at.%)	2.16	1.80	[17]

The FFTF baseline is a simple attempt at modeling FFTF pins in TREAT transients. A much more detailed analysis of power coupling factors between TREAT reactor power and experiment pin power is needed. Modification of coolant channel size and flow is also necessary to achieve targeted temperature profiles. The results from the FFTF baseline are not to be taken as high fidelity results but initial exploration of potential possibilities.

A comparison of peak temperatures of interest is shown in figure 9. Immediately, the temperature profiles show both baseline pins would have a portion of the fuel slug melt. Eutectic penetration is also possible due to cladding temperatures passing the onset of eutectic melting temperature.

As in the M7 benchmark, a flat profilometry is the initial condition for both MFF pins. As such, figure 10 is provided purely as a comparison and an indicator of behavior possibilities. The displacements shown are in the radial direction and are at 25 s, which is after reactor SCRAM by more than 5 s.

From figure 10, negative displacements indicate that the cladding has shrunk. This result is puzzling as internal plenum pressure remains higher than outer cladding coolant pressure. A possible explanation involves Bison not handling melting. Material property correlations are applied at these high melt temperatures. Some properties are limited, but creep rates are not. As such, non-physical creep during the transient peak when temperatures are highest distorts stress states. When cooling removes thermal expansion effects, non-physical displacements are calculated. Data about temperature and mechanical behavior of metallic fuel at post-melt conditions could guide users and developers to make better assumptions.



(a)

(b)

Figure 9: Comparison of M7 benchmark and FFTF baseline peak temperatures for the MFF-3 pin (a) and MFF-5 pin (b).

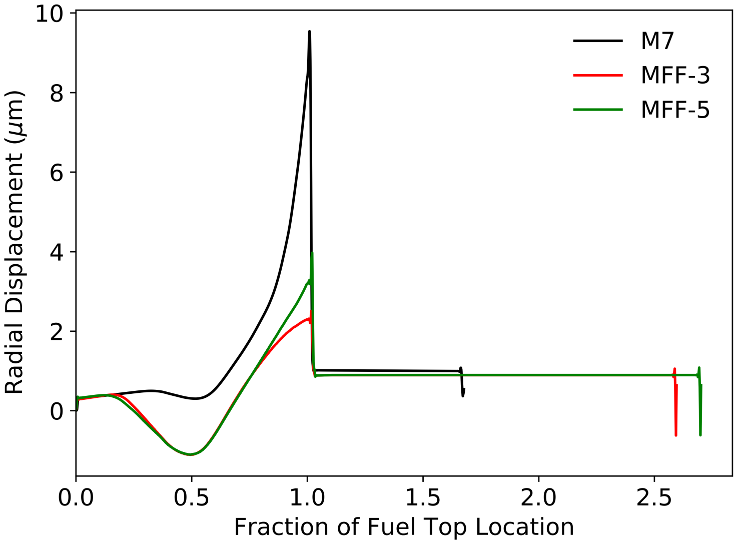


Figure 10: Radial cladding displacement comparison of the M7 benchmark and the FFTF baseline at end of simulation (25 s).

4 Conclusion

As transient experiments stress metallic fuel with difficult conditions, simulation models of transient experiments stress modeling tools in a similar manner. With resumption of transient testing in TREAT and Bison being developed to model metallic fuel, knowing the current capabilities of Bison and possible improvement areas allows focused efforts toward modeling and development of metallic fuel simulation using Bison.

Benchmarks from EBR-II (OPT-1) and TREAT (M5, M6, and M7) compare well with experiment results. The Bison benchmarks approach model development as if they are predicting an experiment's outcome with only initial conditions (after base-irradiation) and experiment conditions available. No calibration of Bison input parameters occurs with input values being found from sources in published data.

Specific areas identified by benchmarks for improvement are:

- Transient specific coolant channel convective boundary condition.
- Ability to inform initial conditions from PIE of base-irradiation pins or sister pins, e.g., profilometry and zirconium distribution.
- Proper material property and failure Bison modules for possible fuel and cladding materials, e.g., D9 cladding material.
- Improved eutectic failure, i.e., fuel-clad chemical interaction, methods perhaps tracking species mass transport in fuel and clad.
- Feedback linkage back from failure or melt occurrences to alter solution progression.
- Intelligent treatment of material melt for simulation through melt and then cooling.

Acknowledgments

This work was funded by the TREAT Integral Transient Fuel Performance Testing program. The submitted manuscript has been authored by a contractor of the U.S. Government under Contract DE-AC07-05ID14517. Accordingly, the U.S. Government retains a non-exclusive, royalty free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.

References

[1] US Department of Energy. DOE National Laboratory Resumes Operation of U.S. Transient Test Reactor.

[2] G. Hansen, C. Newman, D. Gaston, and c. Permann. An implicit solution framework for reactor fuel performance simulation. Technical Report INL/CON-09-15900, Idaho National Laboratory, 2009.

[3] Derek Gaston, Chris Newman, Glen Hansen, and Damien Lebrun-Grandie. MOOSE: A parallel computational framework for coupled systems of nonlinear equations. *Nuclear Engineering and Design*, 239(10):1768–1778, 2009.

[4] N. E. Woolstenhulme, C. C. Baker, J. D. Bess, C. B. Davis, C. M. Hill, G. K. Housley, C. B. Jensen, N. D. Jerred, R. C. O'Brien, S. D. Snow, and D. M. Wachs. Capabilities development for transient testing of advanced nuclear fuels at TREAT. Technical Report INL/CON-16-37608, Idaho National Laboratory, Idaho Falls, ID 83415, September 2016.

[5] H. Tsai, J. P. Herzog, L. L. Briggs, and M. C. Billone. Data package for the irradiation of IFR overpower transient test OPT-1 (X512) in EBR-II. Technical Report IPS-58-00-00, Argonne National Laboratory, 1991.

[6] H. Tsai and M. C. Billone memo dated Oct. 10, 1990. DP-1 pin cladding strain data as related to the Mk-V fuel design. Argonne National Laboratory, 1990.

[7] J. I. Sackett letter dated March 25, 1992 to E. J. Hughes. Preliminary look at: EBR-II transient test OPT-1. Argonne National Laboratory, 1992.

[8] Bison inter-laboratory development team. Bison manual. `https://mooseframework.org/bison`, 2019.

[9] J. K. Fink and L. Leibowitz. Thermodynamic and transport properties of sodium liquid and vapor. Technical Report ANL/RE-95/2, Argonne National Laboratory, 1995.

[10] J. R. Davis, G. E. Deegan, J. D. Leman, and W. H. Perry. Operating experience with sodium pumps at EBR-II. Technical Report ANL/EBR-027, Argonne National Laboratory, 1970.

[11] T. H. Bauer, G. R. Fenske, and J. M. Kramer. Cladding failure margins for metallic fuel in the integral fast reactor. Technical Report CONF-870812–22, Idaho National Laboratory, July 1987.

[12] T. H. Bauer, W. R. Robinson, J. W. Holland, E. A. Rhodes, and A. E. Wright. First overpower tests of metallic IFR fuel in TREAT: Data and analysis from tests M5, M6, and M7. Technical Report ANL-IFR-124, Argonne National Laboratory, 1989.

[13] G. L. Hofman, M. C. Billone, J. F. Koenig, J. M. Kramer, J. D. B. Lambert, L. Leibowitz, Y. Orechwa, D. R. Pedersen, D. L. Porter, H. Tsai, and A. E. Wright. Metallic fuels handbook. Technical Report ANL-NSE-3, Argonne National Laboratory, 2019.

[14] V. I. Subbotin, A. K. Papovyants, P. L. Kirillov, and N. N. Ivanovskii. A study of heat transfer to molten sodium in tubes. *Soviet Atomic Energy*, 13(4):991–994, 1963.

[15] RIA Benchmark Phase II Task Group. Reactivity initiated accident (RIA) fuel codes benchmark phase-II. Technical Report NEA/CSNI/R(2016)6/VOL1, March 2016.

[16] Theodore H. Bauer, Arthur E. Wright, William R. Robinson, John W. Holland, and Edgar A. Rhodes. Behavior of modern metallic fuel in TREAT transient overpower tests. *Nuclear Technology*, 92(3):325–352, 1990.

[17] W. J. Carmack, D. L. Porter, D. G. Cummings, H. J. M. Chichester, S. L. Hayes, and D. W. Wootan. Examination of legacy metallic fuel pins (U-10Zr) tested in FFTF. Technical Report INL/LTD-13-28987-R.1, Idaho National Laboratory, 2013.

Appendix - Benchmark Bison Inputs

As Bison is in active development, benchmark input files are only guaranteed to produce consistent results with those presented if the exact same Bison version is used. As Bison's version control uses Git, the exact version of Bison is identified by a Git hash and this may be used to check it out from the INL code repository.

The specific Bison commit hash is `b2c08d93ce93141c57f049a0d6448ce87920a34a` from August 7, 2019. All simulation results are produced with this version of Bison. Each benchmark input is executed in parallel on 4 processors.

4.1 OPT-1 Benchmark Bison Input

```
# OPT-1 Benchmark of X512 experiment of DP-55 pin
# Not transferring initial condition from base irradiation case
# Units: m, W, kg, Pa

[Debug]
  show_var_residual = ’disp_x disp_y temperature’
  show_var_residual_norms = true
[]

[GlobalParams]
  order = SECOND
  family = LAGRANGE
  energy_per_fission = 3.2e-11 # J/fission
  displacements = ’disp_x disp_y’
  stress_free_temperature = 642.039 # K
[]

[Modules/TensorMechanics/Master]
  add_variables = true
  save_in = ’saved_x saved_y’
  strain = FINITE
  generate_output = ’stress_xx stress_yy stress_zz vonmises_stress
      hydrostatic_stress creep_strain_xx creep_strain_yy creep_strain_zz
      elastic_strain_xx elastic_strain_yy elastic_strain_zz strain_xx
      strain_yy strain_zz’
  [./slug_mech]
    block = pellet
    eigenstrain_names = ’slug_thermal_strain slug_gas_swell’
  [../]
  [./clad_mech]
    block = clad
    eigenstrain_names = ’clad_thermal_eigenstrain’
  [../]
[]

[Problem]
  coord_type = RZ
  type = ReferenceResidualProblem
  solution_variables = ’disp_x disp_y temperature atom_zr’
  reference_residual_variables = ’saved_x saved_y saved_t saved_zr’
  group_variables = ’disp_x disp_y’
[]

[Mesh]
  type = SmearedPelletMesh
  pellet_mesh_density = customize
  clad_mesh_density = customize
  clad_bot_gap_height = 0.381e-3 # m
  clad_gap_width = 1e-6 # m
  clad_thickness = 0.381e-3 # m
  clad_top_gap_height = 588.5e-3 # m
  pellet_height = 28.5e-2 # m
  pellet_outer_radius = 2.545e-3 # m
  pellet_quantity = 1
  top_bot_clad_height = 1.9e-3 # m

  nx_c = 4 # radial clad elements
  nx_p = 10 # radial fuel pellet elements
  ny_c = 260 # axial clad elements
  ny_cl = 3 # axial clad lower plug elements
  ny_cu = 3 # axial clad upper plug elements
  ny_p = 260 # axial fuel pellet elements

  elem_type = QUAD8
  partitioner = centroid
  patch_update_strategy = auto
  centroid_partitioner_direction = y
[]

[Functions]
  [./power_history]
    type = PiecewiseLinear
    x = ’0     86400 86711 86716 87000’ # seconds
    y = ’43000 43000 56400 0     0’ # W/m
  [../]
  [./axial_peak_factors]
    type = PowerPeakingFunction
    fit = EBRII_ROW_3
    pellet_length = 28.5e-2 # m
    pellet_y_start = 0.002281 # m
  [../]
  [./q_heat]
    type = CompositeFunction
    functions = ’power_history axial_peak_factors’
    scale_factor = 49141.369 # m^-2
  [../]
  [./dt_fun]
    type = PiecewiseLinear
    x = ’0   86300 86400 86700 86720 87000’ # Time
    y = ’900 900   5     0.1   0.1   5’ # dt
  [../]
[]

[Variables]
  [./temperature]
    initial_condition = 642.039 # K
  [../]
  [./atom_zr]
    block = pellet
    initial_condition = 0.225
    scaling = 1.0e10
  [../]
[]
```

```
[Kernels]
  [./gravity]
    type = Gravity
    variable = disp_y
    value = -9.81
    save_in = ’saved_x saved_y’
  [../]
  [./heat_slug]
    type = ConstitutiveHeatConduction
    block = pellet
    variable = temperature
    save_in = ’saved_t’
    thermal_conductivity = ’thermal_conductivity’
    thermal_conductivity_args = ’temperature atom_zr’
    thermal_conductivity_derivs = ’thermal_conductivity_dT
        thermal_conductivity_dZr’
  [../]
  [./heat_clad]
    type = ConstitutiveHeatConduction
    block = clad
    variable = temperature
    save_in = ’saved_t’
    thermal_conductivity = ’thermal_conductivity’
    thermal_conductivity_args = ’temperature’
    thermal_conductivity_derivs = ’thermal_conductivity_dT’
  [../]
  [./heat_ie]
    type = ConstitutiveHeatConductionTimeDerivative
    variable = temperature
    save_in = ’saved_t’
    specific_heat = ’specific_heat’
    specific_heat_derivs = ’specific_heat_dT’
    specific_heat_args = ’temperature’
  [../]
  [./vol_heat_source]
    save_in = ’saved_t’
    block = pellet
    type = FissionRateHeatSource
    variable = temperature
    fission_rate = fission_rate
    fission_rate_args = ’atom_zr’
    fission_rate_derivs = ’fission_rate_dZr’
  [../]
  [./slug_zr_diffusion]
    type = ZirconiumDiffusion
    block = pellet
    temp = temperature
    variable = atom_zr
    save_in = ’saved_zr’
  [../]
  [./slug_zr_diffusion_ie]
    type = TimeDerivative
    block = pellet
    variable = atom_zr
    save_in = ’saved_zr’
  [../]
[]
```

```
[AuxVariables]
  [./saved_t]
  [../]
  [./saved_x]
  [../]
  [./saved_y]
  [../]
  [./saved_zr]
    block = pellet
  [../]
  [./power_density]
    block = pellet
  [../]
  [./fissions]
    order = CONSTANT
    family = MONOMIAL
    block = pellet
  [../]
  [./gap_cond]
    order = CONSTANT
    family = MONOMIAL
  [../]
  [./coolant_htc]
    order = CONSTANT
    family = MONOMIAL
  [../]
  [./peak_coolant_temp]
    order = CONSTANT
    family = MONOMIAL
    block = clad
  [../]
  [./cumulative_damage_index]
    order = CONSTANT
    family = MONOMIAL
  [../]
  [./element_failed]
    order = CONSTANT
    family = MONOMIAL
  [../]
  [./eutectic_thickness]
    order = CONSTANT
    family = MONOMIAL
    block = clad
  [../]
[]

[AuxKernels]
  [./calc_power_density]
    type = FunctionAux
    function = q_heat
    variable = power_density
    execute_on = timestep_end
    block = pellet
  [../]
  [./fissions]
    type = MaterialRealAux
    block = pellet
    property = fission_rate
    variable = fissions
    execute_on = timestep_end
  [../]
  [./conductance]
    type = MaterialRealAux
    property = gap_conductance
    variable = gap_cond
    boundary = 10
  [../]
  [./coolant_temperature]
    type = MaterialRealAux
    property = coolant_temperature
    variable = peak_coolant_temp
    boundary = 2
  [../]
  [./cdf_amount]
    boundary = '1 2 3'
    type = MaterialRealAux
    property = cdf_failure
    variable = cumulative_damage_index
  [../]
  [./failed_element]
    boundary = '1 2 3'
    type = MaterialRealAux
    property = failed
    variable = element_failed
  [../]
  [./fcci_eutectic]
    execute_on = timestep_end
    type = EutecticThicknessFCCI
    boundary = 5
    variable = eutectic_thickness
    temperature = temperature
  [../]
[]

[Postprocessors]
  [./_dt]
    type = TimestepSize
  [../]
  [./num_lin_it]
    type = NumLinearIterations
  [../]
  [./num_nonlin_it]
    type = NumNonlinearIterations
  [../]
  [./tot_lin_it]
    type = CumulativeValuePostprocessor
    postprocessor = num_lin_it
  [../]
  [./tot_nonlin_it]
    type = CumulativeValuePostprocessor
    postprocessor = num_nonlin_it
  [../]
  [./alive_time]
    type = PerfGraphData
    section_name = Root
    data_type = TOTAL
  [../]

  [./ave_temp_interior]
    type = SideAverageValue
    boundary = 9
    variable = temperature
    execute_on = ’initial linear’
  [../]
  [./avg_clad_temp]
    type = ElementAverageValue
    variable = temperature
    outputs = ’exodus csv’
    block = clad
  [../]
  [./avg_fuel_temp]
    type = ElementAverageValue
    variable = temperature
    outputs = ’exodus csv’
    block = pellet
  [../]
  [./max_outer_clad_temp]
    type = NodalExtremeValue
    variable = temperature
    value_type = max
    boundary = 2
  [../]
  [./peak_outer_fuel_temp]
    type = NodalExtremeValue
    variable = temperature
    value_type = max
    boundary = 10
  [../]
  [./peak_coolant_temperature]
    type = ElementExtremeValue
    variable = peak_coolant_temp
    value_type = max
    block = clad
    outputs = ’all’
  [../]
  [./peak_clad_temp]
    type = ElementExtremeValue
    variable = temperature
    value_type = max
    block = clad
    outputs = ’exodus csv’
  [../]
  [./peak_fuel_temp]
    type = ElementExtremeValue
    variable = temperature
    value_type = max
    block = pellet
    outputs = ’exodus csv’
  [../]
  [./flux_from_clad]
    type = SideFluxIntegral
    variable = temperature
    boundary = 5
```

```
    diffusivity = thermal_conductivity
    outputs = ’exodus csv’
  [../]
  [./flux_from_fuel]
    type = SideFluxIntegral
    variable = temperature
    boundary = 10
    diffusivity = thermal_conductivity
    outputs = ’exodus csv’
  [../]
  [./pellet_volume]
    type = InternalVolume
    boundary = 8
    outputs = ’console csv’
  [../]
  [./rod_total_power]
    type = ElementIntegralPower
    variable = temperature
    block = pellet
    use_material_fission_rate = true
    fission_rate_material = fission_rate
    outputs = ’all’
  [../]
  [./avg_power_density]
    type = ElementAverageValue
    variable = power_density
    block = pellet
    outputs = ’all’
  [../]
  [./avg_fission_rate]
    type = ElementAverageValue
    variable = fissions
    block = pellet
    outputs = ’all’
  [../]
  [./fis_gas_released]
    type = ElementIntegralFisGasProduce
    block = pellet
    execute_on = ’initial timestep_end’
  [../]
  [./gas_volume]
    type = InternalVolume
    boundary = 9
    execute_on = ’initial linear’
    addition = -1.404661e-6 # m3; Volume of displaced sodium
  [../]
  [./clad_inner_vol]
    type = InternalVolume
    boundary = 7
    outputs = ’exodus’
  [../]
  [./max_eutectic_pen]
    type = ElementExtremeValue
    variable = eutectic_thickness
    block = clad
    value_type = max
    execute_on = timestep_end
    outputs = ’all’
  [../]
[]

[Contact]
  [./pellet_clad_mechanical]
    master = 5
    slave = 10
    penalty = 1e12
    model = glued
    normalize_penalty = true
    tangential_tolerance = 1e-3
    normal_smoothing_distance = 0.1
    system = constraint
  [../]
[]

[ThermalContact]
  [./thermal_contact]
    type = GapHeatTransfer
    variable = temperature
    master = 5
    slave = 10
    quadrature = true
    gap_conductivity = 62.90 # W/mK
    min_gap = 1e-3 # m
  [../]
[]

[BCs]
  [./no_x_all]
    type = DirichletBC
    variable = disp_x
    boundary = 12
    value = 0.0
  [../]
  [./no_y_fuel]
    type = PenaltyDirichletBC
    penalty = 1e10
    variable = disp_y
    boundary = 20
    value = 0.0
  [../]
  [./no_y_clad]
    type = DirichletBC
    variable = disp_y
    boundary = 1
    value = 0.0
  [../]

  [./Pressure]
    [./coolantPressure]
      boundary = ’1 2 3’
      factor = 0.439168e6 # Pa
    [../]
  [../]
  [./PlenumPressure]
    [./plenumPressure]
      boundary = 9
      initial_pressure = 5.85E+06 # Pa
      initial_temperature = 883.0 # K
      startup_time = 0.0
      R = 8.3143 # J/mol K
      temperature = ave_temp_interior
      volume = gas_volume
      output = plenum_pressure
      material_input = fis_gas_released
      output_initial_moles = plenum_moles
    [../]
  [../]
[]

[CoolantChannel]
  [./convective_clad_surface]
    boundary = ’1 2 3’
    variable = temperature
    coolant_material = sodium # Do NOT add water...
    inlet_temperature = 642.039
    inlet_pressure = 0.439168e6 # Pa; Assuming pumps are near full power
    subchannel_geometry = triangular
    rod_diameter = 0.00585216
    rod_pitch = 0.00692216
    inlet_massflux = 4.17292E+03 # kg/s; Not same as base irradiation
    number_axial_zone = 50
    compute_enthalpy = true # Material property: coolant_enthalpy
  [../]
[]

[Materials]
  # Fuel Slug Properties
  [./slug_phase]
    type = PhaseUPuZr
    block = pellet
    temp = temperature
    X_Zr = atom_zr
    X_Pu = 0.163
  [../]
  [./slug_massdif]
    type = ZrDiffusivityUPuZr
    block = pellet
    temp = temperature
    X_Zr = atom_zr
    X_Pu = 0.163
  [../]
  [./fission_rate]
    type = UPuZrFissionRate
    block = pellet
    rod_linear_power = power_history
    axial_power_profile = axial_peak_factors
    pellet_radius = 2.545e-3 # m
    X_Zr = atom_zr
    initial_X_Zr = 0.225
  [../]
  [./slug_elasticity_tensor]
    type = UPuZrElasticityTensor
    block = pellet
    temperature = temperature
    X_Zr = atom_zr
    X_Pu = 0.163
    output_properties = 'youngs_modulus poissons_ratio'
    outputs = all
  [../]
  [./slug_stress] # Strain done by Action
    type = ComputeMultipleInelasticStress
    tangent_operator = nonlinear
    inelastic_models = 'slug_creep'
    block = pellet
  [../]
  [./slug_creep]
    type = UPuZrCreepUpdate
    block = pellet
    temperature = temperature
    porosity = porosity
    max_inelastic_increment = 1e-4
  [../]
  [./slug_thermal_expansion]
    type = ComputeThermalExpansionEigenstrain
    block = pellet
    thermal_expansion_coeff = 1.18e-5
    temperature = temperature
    eigenstrain_name = slug_thermal_strain
  [../]
  [./slug_swelling_gas]
    type = UPuZrGaseousEigenstrain
    eigenstrain_name = 'slug_gas_swell'
    temperature = temperature
    initial_porosity = 0.0
    bubble_number_density = 1e20
    outputs = all
    output_properties = 'porosity gaseous_porosity'
    block = pellet
  [../]
  [./slug_thermal]
    type = ThermalUPuZr
    block = pellet
    spheat_model = savage
    temp = temperature
    thcond_model = lanl
    porosity = porosity # for now set to zero
    X_Zr = atom_zr
    X_Pu = 0.163
  [../]
  [./slug_density]
    type = Density
    block = pellet
    density = 15800.0
  [../]
  [./slug_fission_gas_release]
    block = pellet
    type = FgrUPuZr
    fission_rate = fission_rate
  [../]

  # Cladding Properties
  [./clad_elasticity_tensor]
```

```
    type = ComputeIsotropicElasticityTensor
    youngs_modulus = 1.88e11
    poissons_ratio = 0.236
    block = clad
  [../]
  [./clad_stress]
    type = ComputeMultipleInelasticStress
    tangent_operator = nonlinear
    inelastic_models = 'clad_creep'
    block = clad
  [../]
  [./clad_fast_flux]
    type = FastNeutronFlux
    block = clad
    factor = 2.47e19
  [../]
  [./clad_creep]
    type = HT9CreepUpdate
    block = clad
    temperature = temperature
  [../]
  [./clad_thermal_expansion]
    type = ComputeThermalExpansionEigenstrain
    block = clad
    thermal_expansion_coeff = 1.2e-5
    temperature = temperature
    eigenstrain_name = clad_thermal_eigenstrain
  [../]
  [./clad_thermal]
    type = ThermalHT9
    block = clad
    temp = temperature
  [../]
  [./clad_density]
    type = Density
    block = clad
    density = 7874.0
  [../]

  [./failclad]
    block = clad
    type = FailureCladHT9
    hoop_stress = stress_zz
    temperature = temperature
    method = cdf_short
  [../]
[]

[Preconditioning]
  [./SMP]
    type = SMP
    full = true
    solve_type = 'PJFNK'
    petsc_options = '-snes_ksp_ew -snes_converged_reason'
    petsc_options_iname = '-pc_type -pc_factor_mat_solver_package -
        ksp_gmres_restart'
    petsc_options_value = 'lu       superlu_dist                  51'
  [../]
```

```
[]

[Executioner] # Kind of ominous, eh?
  line_search = 'none'
  type = Transient
  l_max_its = 100
  l_tol = 1e-3
  nl_max_its = 200
  nl_rel_tol = 1e-5
  nl_abs_tol = 1e-8

  start_time = 0.0
  end_time = 87000.0 # seconds
  dtmin = 0.01 # s
  dtmax = 30000.0 # s

  [./TimeStepper]
    type = FunctionDT
    function = dt_fun
  [../]
  [./Quadrature]
    order = FIFTH
    side_order = SEVENTH
  [../]
[]

[Outputs]
  perf_graph = true
  interval = 1
  csv = true
  [./console]
    type = Console
    output_linear = true
    output_nonlinear = true
  [../]
  [./exodus]
    type = Exodus
  [../]
[]
# vi:filetype=moose_fw
```

4.2 M5 Benchmark Bison Input

```
# TREAT M5 U-19Pu-10Zr pin clad in D9 transient only simulation. (Not
   using transferred initial condition.)
# ANL-IFR-124 Test M5 T-280 in flow tube #1. First tests only (M5-F1)
# Units: m, W, kg, Pa

[Debug]
  show_var_residual = 'disp_x disp_y temperature'
  show_var_residual_norms = true
[]

[GlobalParams]
  order = SECOND
  family = LAGRANGE
  energy_per_fission = 3.28451e-11  # J/fission
  displacements = 'disp_x disp_y'
  stress_free_temperature = 573.0 # K
  # u_weight = 0.7019
  # pu_weight = 0.1877
  # zr_weight = 0.1104
  # X_U = 0.59837
  X_Pu = 0.15610
  X_Zr = 0.24553
[]

[Problem]
  coord_type = RZ
  type = ReferenceResidualProblem
  solution_variables = ’disp_x disp_y temperature’
  reference_residual_variables = ’saved_x saved_y saved_t’
  group_variables = ’disp_x disp_y’
[]

[Mesh]
  type = SmearedPelletMesh
  clad_bot_gap_height = 0.38e-3 # m
  clad_gap_width = 1e-6 # m
  clad_thickness = 0.38e-3 # m
  clad_top_gap_height = 2.44e-1 # m
  pellet_height = 3.51e-1 # m
  pellet_outer_radius = 2.54e-3 # m
  pellet_quantity = 1
  top_bot_clad_height = 1.9e-3 # m

  elem_type = QUAD8
  clad_mesh_density = customize
  pellet_mesh_density = customize
  # Clad is usually coarser than fuel
  nx_c = 4 # radial clad elements
  nx_p = 10 # radial fuel pellet elements
  ny_c = 300 # axial clad elements (in active region)
  ny_cl = 3 # axial clad lower plug elements
  ny_cu = 3 # axial clad upper plug elements
  ny_p = 350 # axial fuel pellet elements

  partitioner = centroid
  patch_size = 25
  patch_update_strategy = auto
  centroid_partitioner_direction = y
[]

[Functions]
  [./power_history] # Peak power density
    type = PiecewiseLinear
    data_file = ’TREATpow_M5_F1.csv’
    format = columns
    scale_factor = 359.8637 # 5.22 W/g/MW PCF * 77.46 g * 0.89 TCF
  [../]
  [./axial_peaking_factors] # Peak factor (1 max)
    type = PiecewiseBilinear
    data_file = ’m7_adjm5_axial_tranAlone.csv’
    axis = 1 # y (no MooseEnum, so integer)
  [../]
  [./q_heat]
    type = CompositeFunction
    functions = ’power_history axial_peaking_factors’
    scale_factor = 140564.48 # m^-3; volume of fuel slug, 1/V, to make
        power density
  [../]
  [./dt_fun]
    type = PiecewiseLinear
    x = ’0   5.5  15.4 20’
    y = ’0.5 0.05 0.05 0.5’
  [../]
[]

[Variables]
  [./temperature]
    initial_condition = 573.0 # K
  [../]
[]

[Modules/TensorMechanics/Master]
  add_variables = true
  save_in = ’saved_x saved_y’
  strain = FINITE
  generate_output = ’stress_xx stress_yy stress_zz vonmises_stress
      hydrostatic_stress elastic_strain_xx elastic_strain_yy
      elastic_strain_zz strain_xx strain_yy strain_zz’
  [./slug_mech]
    block = pellet
    eigenstrain_names = ’slug_thermal_strain’
  [../]
  [./clad_mech]
    block = clad
    eigenstrain_names = ’clad_thermal_eigenstrain’
  [../]
[]

[Kernels]
  [./gravity]
    type = Gravity
    variable = disp_y
    value = -9.81
    save_in = ’saved_x saved_y’
  [../]
  [./heat]
    type = ConstitutiveHeatConduction
    variable = temperature
    save_in = ’saved_t’
    thermal_conductivity = ’thermal_conductivity’
    thermal_conductivity_args = ’temperature’
    thermal_conductivity_derivs = ’thermal_conductivity_dT’
  [../]
  [./heat_ie]
    type = ConstitutiveHeatConductionTimeDerivative
    variable = temperature
    save_in = ’saved_t’
    specific_heat = ’specific_heat’
    specific_heat_args = ’temperature’
    specific_heat_derivs = ’specific_heat_dT’
  [../]
  [./vol_heat_source]
```

```
    type = HeatSource
    block = pellet
    function = q_heat
    variable = temperature
    save_in = 'saved_t'
  [../]
[]

[AuxVariables]
  [./power_density]
    block = pellet
  [../]
  [./fission_rate]
    block = pellet
  [../]
  [./fast_neutron_flux]
    block = clad
  [../]
  [./fast_neutron_fluence]
    block = clad
  [../]

  # AuxVariables for Reference Residual Problem
  [./saved_x]
  [../]
  [./saved_y]
  [../]
  [./saved_t]
  [../]

  # Aux variables for output
  [./energy_density]
    block = pellet
    initial_condition = 0.0
  [../]

  [./gap_cond]
    order = CONSTANT
    family = MONOMIAL
  [../]
  [./coolant_htc]
    order = CONSTANT
    family = MONOMIAL
    block = clad
  [../]
  [./coolant_temperature]
    order = CONSTANT
    family = MONOMIAL
    block = clad
  [../]
  [./element_failed]
    order = CONSTANT
    family = MONOMIAL
    block = pellet
  [../]
  [./eutectic_thickness]
    order = CONSTANT
    family = MONOMIAL
    block = clad
  [../]
[]

[AuxKernels]
  [./calc_fission_rate]
    type = FissionRateFromPowerDensity
    block = pellet
    function = q_heat
    variable = fission_rate
    execute_on = ’initial timestep_end’
  [../]
  [./fast_neutron_flux]
    type = FastNeutronFluxAux
    variable = fast_neutron_flux
    block = clad
    factor = 0.6e19
    execute_on = ’initial timestep_end’
  [../]
  [./fast_neutron_fluence]
    type = FastNeutronFluenceAux
    variable = fast_neutron_fluence
    block = clad
    fast_neutron_flux = fast_neutron_flux
    execute_on = ’initial timestep_end’
  [../]
  [./calc_power_density]
    type = FunctionAux
    block = pellet
    function = q_heat
    variable = power_density
    execute_on = ’initial timestep_end’
  [../]
  [./calc_energy_density]
    type = VariableTimeIntegrationAux
    block = pellet
    order = 2
    variable_to_integrate = power_density
    variable = energy_density
    execute_on = timestep_end
  [../]
  # Hoop stress_zz.

  [./conductance]
    type = MaterialRealAux
    property = gap_conductance
    variable = gap_cond
    boundary = 10
  [../]
  [./coolant_temperature]
    type = MaterialRealAux
    property = coolant_temperature
    variable = coolant_temperature
    boundary = 2
  [../]
  [./failed_element]
    type = MaterialRealAux
    property = failed
    variable = element_failed
    block = pellet
  [../]
  [./fcci_eutectic_in]
    execute_on = timestep_end
    type = EutecticThicknessFCCI
    boundary = 5
    variable = eutectic_thickness
    temperature = temperature
  [../]
  [./fcci_eutectic_out]
    execute_on = timestep_end
    type = EutecticThicknessFCCI
    boundary = 2
    variable = eutectic_thickness
    temperature = temperature
  [../]
[]

[Postprocessors]
  [./_dt]
    type = TimestepSize
  [../]
  [./num_lin_it]
    type = NumLinearIterations
  [../]
  [./num_nonlin_it]
    type = NumNonlinearIterations
  [../]
  [./tot_lin_it]
    type = CumulativeValuePostprocessor
    postprocessor = num_lin_it
  [../]
  [./tot_nonlin_it]
    type = CumulativeValuePostprocessor
    postprocessor = num_nonlin_it
  [../]

  [./fis_gas_released]
    type = ElementIntegralFisGasProduce
    block = pellet
    execute_on = 'initial timestep_end'
  [../]
  [./ave_temp_interior]
    type = SideAverageValue
    boundary = 9
    variable = temperature
    execute_on = 'initial linear'
  [../]
  [./avg_clad_temp]
    type = ElementAverageValue
    variable = temperature
    outputs = 'exodus csv'
    block = clad
  [../]
  [./avg_fuel_temp]
    type = ElementAverageValue
    variable = temperature
    outputs = ’exodus csv’
    block = pellet
  [../]
  [./max_outer_clad_temp]
    type = NodalExtremeValue
    variable = temperature
    value_type = max
    boundary = 2
  [../]
  [./peak_clad_temp]
    type = ElementExtremeValue
    variable = temperature
    value_type = max
    block = clad
    outputs = ’exodus csv’
  [../]
  [./peak_fuel_temp]
    type = ElementExtremeValue
    variable = temperature
    value_type = max
    block = pellet
    outputs = ’exodus csv’
  [../]
  [./peak_outer_fuel_temp]
    type = NodalExtremeValue
    variable = temperature
    value_type = max
    boundary = 10
  [../]
  [./peak_coolant_temperature]
    type = ElementExtremeValue
    variable = coolant_temperature
    value_type = max
    block = clad
    outputs = ’all’
  [../]
  [./flux_from_clad]
    type = SideFluxIntegral
    variable = temperature
    boundary = 5
    diffusivity = thermal_conductivity
    outputs = ’exodus csv’
  [../]
  [./flux_from_fuel]
    type = SideFluxIntegral
    variable = temperature
    boundary = 10
    diffusivity = thermal_conductivity
    outputs = ’exodus csv’
  [../]
  [./max_power_density]
    type = ElementExtremeValue
    variable = power_density
    value_type = max
    block = pellet
  [../]
  [./avg_power_density]
    type = ElementAverageValue
    variable = power_density
    block = pellet
    outputs = ’exodus csv’
  [../]
  [./total_deposit_pin_energy]
    type = ElementIntegralVariablePostprocessor
    variable = energy_density
    block = pellet
  [../]
  [./clad_inner_vol]
    type = InternalVolume
    boundary = 7
    outputs = ’exodus’
  [../]
  [./pellet_volume]
    type = InternalVolume
    boundary = 8
    outputs = ’exodus’
    execute_on = ’initial timestep_end’
  [../]
  [./gas_volume]
    type = InternalVolume
    boundary = 9
    execute_on = ’initial linear’
    outputs = ’exodus csv’
    addition = -2.0e-6 # m3; assuming all bond sodium displaced by closed
        gap
  [../]
  [./clad_fuel_gap]
    type = NodalMaxValue
    variable = penetration
    boundary = 10
    outputs = ’exodus csv’
  [../]
  [./max_eutectic_pen]
    type = ElementExtremeValue
    variable = eutectic_thickness
    block = clad
    value_type = max
    execute_on = timestep_end
    outputs = ’all’
  [../]
[]

[Contact]
  [./pellet_clad_mechanical]
    master = 5
    slave = 10
    penalty = 1e12
    model = frictionless
    normalize_penalty = true
    tangential_tolerance = 1e-3
    normal_smoothing_distance = 0.1
    system = constraint
  [../]
[]

[ThermalContact]
```

```
  [./thermal_contact]
    type = GapHeatTransfer
    variable = temperature
    master = 5
    slave = 10
    quadrature = true
    gap_conductivity = 73.70 # W/mK
    min_gap = 1e-3 # m
  [../]
[]

[BCs]
  [./no_x_all]
    type = DirichletBC
    variable = disp_x
    boundary = 12
    value = 0.0
  [../]
  [./no_y_fuel]
    type = PenaltyDirichletBC
    penalty = 1e10
    variable = disp_y
    boundary = 20
    value = 0.0
  [../]
  [./no_y_clad]
    type = DirichletBC
    variable = disp_y
    boundary = 1
    value = 0.0
  [../]

  [./Pressure]
    [./coolantPressure]
      boundary = ’1 2 3’
      factor = 413685.0 # Pa
    [../]
  [../]
  [./PlenumPressure]
    [./plenumPressure]
      boundary = 9
      initial_pressure = 268896.0 # Pa
      initial_temperature = 309.0 # K
      startup_time = 0.0
      R = 8.3143 # J/mol K
      temperature = ave_temp_interior
      volume = gas_volume
      output = plenum_pressure
      material_input = fis_gas_released
      output_initial_moles = plenum_moles
    [../]
  [../]
[]

[CoolantChannel]
  [./convective_clad_surface]
    boundary = ’1 2 3’
    variable = temperature
    inlet_temperature = 573.0 # K
    inlet_pressure = 413685.0 # Pa
    inlet_massflux = 3761.261 # kg/m^2 sec
    coolant_material = sodium
    flow_area = 2.22e-5 # m^2
    hydraulic_diameter = 2.057e-3 # m
    heated_perimeter = 1.835e-2 # m
    heated_diameter = 4.84e-3 # m
    number_axial_zone = 50
    htc_correlation_type = 3 # Seban’s for sodium
    compute_enthalpy = true # Material property: coolant_enthalpy
  [../]
[]

[Materials]
  # Fuel Slug Properties
  [./set_porosity]
    type = GenericConstantMaterial
    block = pellet
    prop_names = porosity
    prop_values = 0.27 # fractional porosity
  [../]
  [./set_mat_fission_rate]
    type = ParsedMaterial
    block = pellet
    args = ’fission_rate’
    function = ’fission_rate * 1.0’
    f_name = ’fission_rate’
  [../]
  [./melted]
    type = GenericMaterialFailure
    block = pellet
    compared = greater_than
    variable_check = temperature
    constant_criteria = 1515.0 #K
  [../]
  [./for_youngs]
    type = ParsedMaterial
    block = pellet
    outputs = all
    output_properties = ’youngs_modulus’
    f_name = ’youngs_modulus’
    args = ’temperature’
    material_property_names = ’porosity’
    constant_names = ’T_limit E_U T_meltU W_Zr W_Pu B_E Ta_start Ta_end’
    constant_expressions = ’1350.0 1.6e11 1405.0 0.1132 0.1843 1.2 923.0
        973.0’
    function = ’E_p := 1.0 - B_E * porosity; E_W := (1.0 + 0.17 * W_Zr) /
        (1.0 + 1.34 * W_Zr) - W_Pu; T_act := if(temperature > T_limit,
        T_limit, temperature); x_smooth := (T_act - Ta_start) / (Ta_end -
        Ta_start); f_smooth := if(T_act < Ta_start, 0.0, if(T_act > Ta_end,
         1.0, 6.0 * pow(x_smooth, 5) - 15.0 * pow(x_smooth, 4) + 10.0 * pow
        (x_smooth, 3))); E_T := 1.0 - 1.06 * (T_act - 588.0) / T_meltU -
        f_smooth * 0.3 * (1.0 - 1.06 * (Ta_end - 588.0) / T_meltU); E_U *
        E_T * E_p * E_W’
  [../]
  [./for_poissons]
    type = ParsedMaterial
    block = pellet
    outputs = all
    output_properties = 'poissons_ratio'
    f_name = 'poissons_ratio'
    args = 'temperature'
    material_property_names = 'porosity'
    constant_names = 'T_limit nu_U T_meltU W_Zr B_nu'
    constant_expressions = '1350.0 0.24 1405.0 0.1132 0.8'
    function = 'nu_p := 1.0 - B_nu * porosity; nu_W := (1.0 + 3.4 * W_Zr)
        / (1.0 + 1.9 * W_Zr); T_act := if(temperature > T_limit, T_limit,
        temperature); nu_T := 1.0 + 1.2 * (T_act - 588.0) / T_meltU; nu_U *
        nu_p * nu_W * nu_T'
  [../]
  [./slug_elasticity_tensor]
    type = ComputeVariableIsotropicElasticityTensor
    block = pellet
    args = temperature
    youngs_modulus = youngs_modulus
    poissons_ratio = poissons_ratio
  [../]
  [./slug_stress] # Strain done by Action
    type = ComputeFiniteStrainElasticStress
    block = pellet
  [../]
  [./slug_thermal_expansion]
    type = ComputeThermalExpansionEigenstrain
    block = pellet
    thermal_expansion_coeff = 1.18e-5 # FIND SOURCE
    temperature = temperature
    eigenstrain_name = slug_thermal_strain
  [../]
  [./slug_thermal]
    type = ThermalUPuZr
    block = pellet
    Na_depth = 0.0 # fraction
    fuel_outer_radius = 2.54e-3 # m
    spheat_model = savage
    temp = temperature
    thcond_model = billone
    outputs = all
    output_properties = thermal_conductivity
  [../]
  [./slug_density]
    type = Density
    block = pellet
    density = 15400.0 # kg/m3
  [../]
  [./slug_fission_gas_release]
    block = pellet
    type = FgrUPuZr
    fission_rate = fission_rate
  [../]

  # Cladding Properties
  [./clad_youngs]
    type = ParsedMaterial
    block = clad
    outputs = all
    output_properties = ’youngs_modulus’
    f_name = ’youngs_modulus’
    args = ’temperature’
    function = ’1.0e6 * (2.01e5 - 79.29 * (temperature - 273.0))’
  [../]
  [./clad_poissons]
    type = ParsedMaterial
    block = clad
    outputs = all
    output_properties = ’poissons_ratio’
    f_name = ’poissons_ratio’
    args = ’temperature’
    function = ’cel := temperature - 273.0; E_mpa := 2.01e5 - 79.29 * cel;
         G_mpa := 7.86e4 - 38.41 * cel; E_mpa / (2.0 * G_mpa) - 1.0’
  [../]
  [./clad_elasticity_tensor]
    type = ComputeVariableIsotropicElasticityTensor
    block = clad
    args = temperature
    youngs_modulus = youngs_modulus
    poissons_ratio = poissons_ratio
  [../]
  [./clad_stress]
    type = ComputeFiniteStrainElasticStress
    block = clad
  [../]
  # No D9 creep model yet.
  [./clad_fast_flux]
    type = FastNeutronFlux
    block = clad
    factor = 0.6e19
  [../]
  [./clad_thermal_expansion]
    type = ComputeThermalExpansionEigenstrain
    block = clad
    thermal_expansion_coeff = 1.61e-5
    temperature = temperature
    eigenstrain_name = clad_thermal_eigenstrain
  [../]
  [./clad_thermal]
    type = ThermalD9
    block = clad
    temp = temperature
  [../]
  [./clad_density]
    type = Density
    block = clad
    density = 7851.0 # kg/m3
  [../]
[]

[Preconditioning]
  [./SMP]
    type = SMP
    full = true
    solve_type = ’PJFNK’
    petsc_options = ’-snes_ksp_ew -snes_converged_reason’
    petsc_options_iname = ’-pc_type -pc_factor_mat_solver_package -
```

```
       ksp_gmres_restart'
    petsc_options_value = 'lu       superlu_dist                  51'
  [../]
[]

[Executioner] # Kind of ominous, eh?
  type = Transient
  line_search = 'none'

  l_max_its = 100
  l_tol = 1e-3

  nl_max_its = 200
  nl_rel_tol = 1e-6
  nl_abs_tol = 1e-9

  start_time = 0.0
  end_time = 20.0 # second
  dtmin = 0.0001
  dtmax = 1.0

  [./TimeStepper]
    type = FunctionDT
    function = dt_fun
    growth_factor = 1.5
  [../]
  [./Quadrature]
    order = FIFTH
    side_order = SEVENTH
  [../]
[]

[Outputs]
  perf_graph = true
  interval = 1
  csv = true
  [./console]
    type = Console
    output_linear = true
    output_nonlinear = true
  [../]
  [./exodus]
    type = Exodus
  [../]
[]
# vi:filetype=moose_fw
```

4.3 M6 Benchmark Bison Input

```
# TREAT M6 U-19Pu-10Zr pin clad in D9 transient only simulation. (Not
   using transferred initial condition.)
# ANL-IFR-124 Test M6 T-186
# Units: m, W, kg, Pa

[Debug]
  show_var_residual = 'disp_x disp_y temperature'
  show_var_residual_norms = true
[]
```

```
[GlobalParams]
  order = SECOND
  family = LAGRANGE
  energy_per_fission = 3.28451e-11  # J/fission
  displacements = ’disp_x disp_y’
  stress_free_temperature = 583.0 # K
  # u_weight = 0.7025
  # pu_weight = 0.1843
  # zr_weight = 0.1132
  # X_U = 0.59653
  X_Pu = 0.15266
  X_Zr = 0.25081
[]

[Problem]
  coord_type = RZ
  type = ReferenceResidualProblem
  solution_variables = ’disp_x disp_y temperature’
  reference_residual_variables = ’saved_x saved_y saved_t’
  group_variables = ’disp_x disp_y’
[]

[Mesh]
  type = SmearedPelletMesh
  clad_bot_gap_height = 0.38e-3 # m
  clad_gap_width = 1e-6 # m
  clad_thickness = 0.38e-3 # m
  clad_top_gap_height = 252.35e-3 # m
  pellet_height = 34.7e-2 # m
  pellet_outer_radius = 2.54e-3 # m
  pellet_quantity = 1
  top_bot_clad_height = 1.9e-3 # m

  elem_type = QUAD8
  clad_mesh_density = customize
  pellet_mesh_density = customize
  # Clad is usually coarser than fuel
  nx_c = 4 # radial clad elements
  nx_p = 10 # radial fuel pellet elements
  ny_c = 300 # axial clad elements (in active region)
  ny_cl = 3 # axial clad lower plug elements
  ny_cu = 3 # axial clad upper plug elements
  ny_p = 350 # axial fuel pellet elements

  partitioner = centroid
  patch_size = 25
  patch_update_strategy = auto
  centroid_partitioner_direction = y
[]

[Functions]
  [./power_history] # Peak power density
    type = PiecewiseLinear
    data_file = ’powerTREAT_M6.csv’
    format = columns
    scale_factor = 386.46405 # 5.53 W/g/MW PCF * 77.65 g * 0.9 TCF; grams
        of fuel in pellet for W/g to W
  [../]
  [./axial_peaking_factors] # Peak factor (1 max)
    type = PiecewiseBilinear
    data_file = 'm7_adjm6_axial_tranAlone.csv'
    axis = 1 # y (no MooseEnum, so integer)
  [../]
  [./q_heat]
    type = CompositeFunction
    functions = 'power_history axial_peaking_factors'
    scale_factor = 142184.816 # m^-3; volume of fuel slug, 1/V, to make
        power density
  [../]
  [./dt_fun]
    type = PiecewiseLinear
    x = '0   11  13    14.5 15  20'
    y = '0.1 0.1 0.05  0.05 0.1 0.1'
  [../]
[]

[Variables]
  [./temperature]
    initial_condition = 583.0 # K
  [../]
[]

[Modules/TensorMechanics/Master]
  add_variables = true
  save_in = 'saved_x saved_y'
  strain = FINITE
  generate_output = 'stress_xx stress_yy stress_zz vonmises_stress
      hydrostatic_stress elastic_strain_xx elastic_strain_yy
      elastic_strain_zz strain_xx strain_yy strain_zz'
  [./slug_mech]
    block = pellet
    eigenstrain_names = 'slug_thermal_strain'
  [../]
  [./clad_mech]
    block = clad
    eigenstrain_names = 'clad_thermal_eigenstrain'
  [../]
[]

[Kernels]
  [./gravity]
    type = Gravity
    variable = disp_y
    value = -9.81
    save_in = 'saved_x saved_y'
  [../]
  [./heat]
    type = ConstitutiveHeatConduction
    variable = temperature
    save_in = 'saved_t'
    thermal_conductivity = 'thermal_conductivity'
    thermal_conductivity_args = 'temperature'
    thermal_conductivity_derivs = 'thermal_conductivity_dT'
  [../]
  [./heat_ie]
    type = ConstitutiveHeatConductionTimeDerivative
    variable = temperature
    save_in = ’saved_t’
    specific_heat = ’specific_heat’
    specific_heat_args = ’temperature’
    specific_heat_derivs = ’specific_heat_dT’
  [../]
  [./vol_heat_source]
    type = HeatSource
    block = pellet
    function = q_heat
    variable = temperature
    save_in = ’saved_t’
  [../]
[]

[AuxVariables]
  [./power_density]
    block = pellet
  [../]
  [./fission_rate]
    block = pellet
  [../]
  [./fast_neutron_flux]
    block = clad
  [../]
  [./fast_neutron_fluence]
    block = clad
  [../]

  # AuxVariables for Reference Residual Problem
  [./saved_x]
  [../]
  [./saved_y]
  [../]
  [./saved_t]
  [../]

  # Aux variables for output
  [./energy_density]
    block = pellet
    initial_condition = 0.0
  [../]
  [./gap_cond]
    order = CONSTANT
    family = MONOMIAL
  [../]
  [./coolant_htc]
    order = CONSTANT
    family = MONOMIAL
    block = clad
  [../]
  [./coolant_temperature]
    order = CONSTANT
    family = MONOMIAL
    block = clad
  [../]
  [./element_failed]
    order = CONSTANT
    family = MONOMIAL
    block = pellet
  [../]
  [./eutectic_thickness]
    order = CONSTANT
    family = MONOMIAL
    block = clad
  [../]
[]

[AuxKernels]
  [./calc_fission_rate]
    type = FissionRateFromPowerDensity
    block = pellet
    function = q_heat
    variable = fission_rate
    execute_on = ’initial timestep_end’
  [../]
  [./fast_neutron_flux]
    type = FastNeutronFluxAux
    variable = fast_neutron_flux
    block = clad
    factor = 0.6e19
    execute_on = ’initial timestep_end’
  [../]
  [./fast_neutron_fluence]
    type = FastNeutronFluenceAux
    variable = fast_neutron_fluence
    block = clad
    fast_neutron_flux = fast_neutron_flux
    execute_on = ’initial timestep_end’
  [../]
  [./calc_power_density]
    type = FunctionAux
    block = pellet
    function = q_heat
    variable = power_density
    execute_on = ’initial timestep_end’
  [../]
  [./calc_energy_density]
    type = VariableTimeIntegrationAux
    block = pellet
    order = 2
    variable_to_integrate = power_density
    variable = energy_density
    execute_on = timestep_end
  [../]
  # Hoop stress_zz.
  [./conductance]
    type = MaterialRealAux
    property = gap_conductance
    variable = gap_cond
    boundary = 10
  [../]
  [./coolant_temperature]
    type = MaterialRealAux
    property = coolant_temperature
    variable = coolant_temperature
    boundary = 2
  [../]
  [./failed_element]
    type = MaterialRealAux
    property = failed
    variable = element_failed
    block = pellet
  [../]
  [./fcci_eutectic_in]
    execute_on = timestep_end
    type = EutecticThicknessFCCI
    boundary = 5
    variable = eutectic_thickness
    temperature = temperature
  [../]
  [./fcci_eutectic_out]
    execute_on = timestep_end
    type = EutecticThicknessFCCI
    boundary = 2
    variable = eutectic_thickness
    temperature = temperature
  [../]
[]

[Postprocessors]
  [./_dt]
    type = TimestepSize
  [../]
  [./num_lin_it]
    type = NumLinearIterations
  [../]
  [./num_nonlin_it]
    type = NumNonlinearIterations
  [../]
  [./tot_lin_it]
    type = CumulativeValuePostprocessor
    postprocessor = num_lin_it
  [../]
  [./tot_nonlin_it]
    type = CumulativeValuePostprocessor
    postprocessor = num_nonlin_it
  [../]

  [./fis_gas_released]
    type = ElementIntegralFisGasProduce
    block = pellet
    execute_on = 'initial timestep_end'
  [../]
  [./ave_temp_interior]
    type = SideAverageValue
    boundary = 9
    variable = temperature
    execute_on = 'initial linear'
  [../]
  [./avg_clad_temp]
    type = ElementAverageValue
    variable = temperature
    outputs = 'exodus csv'
    block = clad
  [../]
  [./avg_fuel_temp]
    type = ElementAverageValue
    variable = temperature
    outputs = 'exodus csv'
    block = pellet
  [../]
  [./max_outer_clad_temp]
    type = NodalExtremeValue
    variable = temperature
    value_type = max
    boundary = 2
  [../]
  [./peak_clad_temp]
    type = ElementExtremeValue
    variable = temperature
    value_type = max
    block = clad
    outputs = 'exodus csv'
  [../]
  [./peak_fuel_temp]
    type = ElementExtremeValue
    variable = temperature
    value_type = max
    block = pellet
    outputs = 'exodus csv'
  [../]
  [./peak_outer_fuel_temp]
    type = NodalExtremeValue
    variable = temperature
    value_type = max
    boundary = 10
  [../]
  [./peak_coolant_temperature]
    type = ElementExtremeValue
    variable = coolant_temperature
    value_type = max
    block = clad
    outputs = 'all'
  [../]
  [./flux_from_clad]
    type = SideFluxIntegral
    variable = temperature
    boundary = 5
    diffusivity = thermal_conductivity
    outputs = 'exodus csv'
  [../]
  [./flux_from_fuel]
    type = SideFluxIntegral
    variable = temperature
    boundary = 10
    diffusivity = thermal_conductivity
    outputs = 'exodus csv'
  [../]
  [./max_power_density]
    type = ElementExtremeValue
    variable = power_density
    value_type = max
    block = pellet
  [../]
  [./avg_power_density]
    type = ElementAverageValue
    variable = power_density
    block = pellet
    outputs = ’exodus csv’
  [../]
  [./total_deposit_pin_energy]
    type = ElementIntegralVariablePostprocessor
    variable = energy_density
    block = pellet
  [../]
  [./clad_inner_vol]
    type = InternalVolume
    boundary = 7
    outputs = ’exodus’
  [../]
  [./pellet_volume]
    type = InternalVolume
    boundary = 8
    outputs = ’exodus’
    execute_on = ’initial timestep_end’
  [../]
  [./gas_volume]
    type = InternalVolume
    boundary = 9
    execute_on = ’initial linear’
    outputs = ’exodus csv’
    addition = -2.0e-6 # m3; assuming all bond sodium displaced by closed
       gap
  [../]
  [./clad_fuel_gap]
    type = NodalMaxValue
    variable = penetration
    boundary = 10
    outputs = ’exodus csv’
  [../]
  [./max_eutectic_pen]
    type = ElementExtremeValue
    variable = eutectic_thickness
    block = clad
    value_type = max
    execute_on = timestep_end
    outputs = ’all’
  [../]
[]

[Contact]
  [./pellet_clad_mechanical]
    master = 5
    slave = 10
    penalty = 1e12
    model = frictionless
    normalize_penalty = true
    tangential_tolerance = 1e-3
    normal_smoothing_distance = 0.1
    system = constraint
  [../]
[]

[ThermalContact]
  [./thermal_contact]
    type = GapHeatTransfer
    variable = temperature
    master = 5
    slave = 10
    quadrature = true
    gap_conductivity = 73.70 # W/mK
    min_gap = 1e-3 # m=1 mm
  [../]
[]

[BCs]
  [./no_x_all]
    type = DirichletBC
    variable = disp_x
    boundary = 12
    value = 0.0
  [../]
  [./no_y_fuel]
    type = PenaltyDirichletBC
    penalty = 1e10
    variable = disp_y
    boundary = 20
    value = 0.0
  [../]
  [./no_y_clad]
    type = DirichletBC
    variable = disp_y
    boundary = 1
    value = 0.0
  [../]

  [./Pressure]
    [./coolantPressure]
      boundary = '1 2 3'
      factor = 413685.0 # Pa
    [../]
  [../]
  [./PlenumPressure]
    [./plenumPressure]
      boundary = 9
      initial_pressure = 648107.0 # Pa
      initial_temperature = 309.0 # K
      startup_time = 0.0
      R = 8.3143 # J/mol K
      temperature = ave_temp_interior
      volume = gas_volume
      output = plenum_pressure
      material_input = fis_gas_released
      output_initial_moles = plenum_moles
    [../]
  [../]
[]
```

```
[CoolantChannel]
  [./convective_clad_surface]
    boundary = ’1 2 3’
    variable = temperature
    inlet_temperature = 583.0 # K
    inlet_pressure = 413685.0 # Pa
    inlet_massflux = 3761.261 # kg/m^2 sec
    coolant_material = sodium # Do NOT add water...
    flow_area = 2.22e-5 # m^2
    hydraulic_diameter = 2.057e-3 # m
    heated_perimeter = 1.835e-2 # m
    heated_diameter = 4.84e-3 # m
    number_axial_zone = 50
    htc_correlation_type = 3 # Seban’s for sodium
    compute_enthalpy = true # Material property: coolant_enthalpy
  [../]
[]

[Materials]
  # Fuel Slug Properties
  [./set_porosity]
    type = GenericConstantMaterial
    block = pellet
    prop_names = porosity
    prop_values = 0.27 # fractional porosity
  [../]
  [./set_mat_fission_rate]
    type = ParsedMaterial
    block = pellet
    args = ’fission_rate’
    function = ’fission_rate * 1.0’
    f_name = ’fission_rate’
  [../]
  [./melted]
    type = GenericMaterialFailure
    block = pellet
    compared = greater_than
    variable_check = temperature
    constant_criteria = 1515.0 # K
  [../]
  [./for_youngs]
    type = ParsedMaterial
    block = pellet
    outputs = all
    output_properties = ’youngs_modulus’
    f_name = ’youngs_modulus’
    args = ’temperature’
    material_property_names = ’porosity’
    constant_names = ’T_limit E_U T_meltU W_Zr W_Pu B_E Ta_start Ta_end’
    constant_expressions = ’1350.0 1.6e11 1405.0 0.1132 0.1843 1.2 923.0
       973.0’
    function = ’E_p := 1.0 - B_E * porosity; E_W := (1.0 + 0.17 * W_Zr) /
       (1.0 + 1.34 * W_Zr) - W_Pu; T_act := if(temperature > T_limit,
       T_limit, temperature); x_smooth := (T_act - Ta_start) / (Ta_end -
       Ta_start); f_smooth := if(T_act < Ta_start, 0.0, if(T_act > Ta_end,
        1.0, 6.0 * pow(x_smooth, 5) - 15.0 * pow(x_smooth, 4) + 10.0 * pow
       (x_smooth, 3))); E_T := 1.0 - 1.06 * (T_act - 588.0) / T_meltU -
```

```
     f_smooth * 0.3 * (1.0 - 1.06 * (Ta_end - 588.0) / T_meltU); E_U *
     E_T * E_p * E_W’
[../]
[./for_poissons]
  type = ParsedMaterial
  block = pellet
  outputs = all
  output_properties = ’poissons_ratio’
  f_name = ’poissons_ratio’
  args = ’temperature’
  material_property_names = ’porosity’
  constant_names = ’T_limit nu_U T_meltU W_Zr B_nu’
  constant_expressions = ’1350.0 0.24 1405.0 0.1132 0.8’
  function = ’nu_p := 1.0 - B_nu * porosity; nu_W := (1.0 + 3.4 * W_Zr)
     / (1.0 + 1.9 * W_Zr); T_act := if(temperature > T_limit, T_limit,
     temperature); nu_T := 1.0 + 1.2 * (T_act - 588.0) / T_meltU; nu_U *
     nu_p * nu_W * nu_T’
[../]
[./slug_elasticity_tensor]
  type = ComputeVariableIsotropicElasticityTensor
  block = pellet
  args = temperature
  youngs_modulus = youngs_modulus
  poissons_ratio = poissons_ratio
[../]
[./slug_stress] # Strain done by Action
  type = ComputeFiniteStrainElasticStress
  block = pellet
[../]
[./slug_thermal_expansion]
  type = ComputeThermalExpansionEigenstrain
  block = pellet
  thermal_expansion_coeff = 1.18e-5 # FIND SOURCE
  temperature = temperature
  eigenstrain_name = slug_thermal_strain
[../]
[./slug_thermal]
  type = ThermalUPuZr
  block = pellet
  Na_depth = 4.08e-2 # fraction
  fuel_outer_radius = 2.54e-3 # m
  spheat_model = savage
  temp = temperature
  thcond_model = billone
  outputs = all
  output_properties = thermal_conductivity
[../]
[./slug_density]
  type = Density
  block = pellet
  density = 15500.0 # kg/m3
[../]
[./slug_fission_gas_release]
  block = pellet
  type = FgrUPuZr
  fission_rate = fission_rate
[../]
```

```
  # Cladding Properties
  [./clad_elasticity_tensor]
    type = ComputeIsotropicElasticityTensor
    youngs_modulus = 1.81e11 # at 523 K
    poissons_ratio = 0.313 # at 523 K
    block = clad
  [../]
  [./clad_stress] # Strain done by Action
    type = ComputeFiniteStrainElasticStress
    block = clad
  [../]
  # No D9 creep model yet.
  [./clad_fast_flux]
    type = FastNeutronFlux
    block = clad
    factor = 0.6e19
  [../]
  [./clad_thermal_expansion]
    type = ComputeThermalExpansionEigenstrain
    block = clad
    thermal_expansion_coeff = 1.62e-5
    temperature = temperature
    eigenstrain_name = clad_thermal_eigenstrain
  [../]
  [./clad_thermal]
    type = ThermalD9
    block = clad
    temp = temperature
  [../]
  [./clad_density]
    type = Density
    block = clad
    density = 7846.7 # kg/m3
  [../]
[]

[Preconditioning]
  [./SMP]
    type = SMP
    full = true
    solve_type = 'PJFNK'
    petsc_options = '-snes_ksp_ew -snes_converged_reason'
    petsc_options_iname = '-pc_type -pc_factor_mat_solver_package -
        ksp_gmres_restart'
    petsc_options_value = 'lu       superlu_dist                  51'
  [../]
[]

[Executioner] # Kind of ominous, eh?
  type = Transient
  line_search = 'none'

  l_max_its = 100
  l_tol = 1e-3

  nl_max_its = 200
  nl_rel_tol = 1e-6
  nl_abs_tol = 1e-9
  start_time = 0.0
  end_time = 20.0 # second
  dtmin = 0.0001
  dtmax = 1.0

  [./TimeStepper]
    type = FunctionDT
    function = dt_fun
    growth_factor = 1.5
  [../]
  [./Quadrature]
    order = FIFTH
    side_order = SEVENTH
  [../]
[]

[Outputs]
  perf_graph = true
  interval = 1
  csv = true
  [./console]
    type = Console
    output_linear = true
    output_nonlinear = true
  [../]
  [./exodus]
    type = Exodus
  [../]
[]
# vi:filetype=moose_fw
```

4.4 M7 Benchmark Bison Input

```
# TREAT M7 U-10Zr pin clad in HT9 transient only simulation. (Not using
   transferred initial condition.)
# ANL-IFR-124 Test M7 T-427
# Units: m, W, kg, Pa

[Debug]
  show_var_residual = 'disp_x disp_y temperature'
  show_var_residual_norms = true
[]

[GlobalParams]
  order = SECOND
  family = LAGRANGE
  energy_per_fission = 3.28451e-11  # J/fission
  displacements = 'disp_x disp_y'
  stress_free_temperature = 586.0 # K
  # u_weight = 0.90126
  # pu_weight = 0.00115
  # zr_weight = 0.0976
  # X_U = 0.779
  X_Pu = 0.000969
  X_Zr = 0.22
[]

[Problem]
```

```
  coord_type = RZ
  type = ReferenceResidualProblem
  solution_variables = 'disp_x disp_y temperature'
  reference_residual_variables = 'saved_x saved_y saved_t'
  group_variables = 'disp_x disp_y'
[]

[Mesh]
  type = SmearedPelletMesh
  clad_bot_gap_height = 0.38e-3 # m
  clad_gap_width = 1e-6 # m
  clad_thickness = 0.38e-3 # m
  clad_top_gap_height = 252.35e-3 # m
  pellet_height = 37.3e-2 # m
  pellet_outer_radius = 2.54e-3 # m
  pellet_quantity = 1
  top_bot_clad_height = 1.9e-3 # m

  elem_type = QUAD8
  clad_mesh_density = customize
  pellet_mesh_density = customize
  # Clad is usually coarser than fuel
  nx_c = 4 # radial clad elements
  nx_p = 10 # radial fuel pellet elements
  ny_c = 300 # axial clad elements (in active region)
  ny_cl = 3 # axial clad lower plug elements
  ny_cu = 3 # axial clad upper plug elements
  ny_p = 350 # axial fuel pellet elements

  partitioner = centroid
  patch_size = 25
  patch_update_strategy = auto
  centroid_partitioner_direction = y
[]

[Functions]
  [./power_history] # Peak power density
    type = PiecewiseLinear
    data_file = 'm7fa_digitized.csv'
    format = columns
    scale_factor = 78.8 # g; grams of fuel in pellet for W/g to W
  [../]
  [./axial_peaking_factors] # Peak factor (1 max)
    type = PiecewiseBilinear
    data_file = 'm7_axial_tranAlone.csv'
    axis = 1 # y (no MooseEnum, so integer)
  [../]
  [./q_heat]
    type = CompositeFunction
    functions = 'power_history axial_peaking_factors'
    scale_factor = 132273.8097 # m^-3; volume of fuel slug, 1/V, to make
        power density
  [../]
  [./dt_fun]
    type = PiecewiseLinear
    x = '0   6   7    9    15   20    25'
    y = '0.5 0.5 0.01 0.01 0.01 0.001 0.1'
  [../]
```

```
[]

[Variables]
  [./temperature]
    initial_condition = 586.0 # K
  [../]
[]

[Modules/TensorMechanics/Master]
  add_variables = true
  save_in = 'saved_x saved_y'
  strain = FINITE
  generate_output = 'stress_xx stress_yy stress_zz vonmises_stress
      hydrostatic_stress elastic_strain_xx elastic_strain_yy
      elastic_strain_zz strain_xx strain_yy strain_zz'
  [./slug_mech]
    block = pellet
    additional_generate_output = 'creep_strain_xx creep_strain_yy
        creep_strain_zz'
    eigenstrain_names = 'slug_thermal_strain'
  [../]
  [./clad_mech]
    block = clad
    additional_generate_output = 'creep_strain_xx creep_strain_yy
        creep_strain_zz'
    eigenstrain_names = 'clad_thermal_eigenstrain'
  [../]
[]

[Kernels]
  [./gravity]
    type = Gravity
    variable = disp_y
    value = -9.81
    save_in = 'saved_x saved_y'
  [../]
  [./heat]
    type = ConstitutiveHeatConduction
    variable = temperature
    save_in = 'saved_t'
    thermal_conductivity = 'thermal_conductivity'
    thermal_conductivity_args = 'temperature'
    thermal_conductivity_derivs = 'thermal_conductivity_dT'
  [../]
  [./heat_ie]
    type = ConstitutiveHeatConductionTimeDerivative
    variable = temperature
    save_in = 'saved_t'
    specific_heat = 'specific_heat'
    specific_heat_args = 'temperature'
    specific_heat_derivs = 'specific_heat_dT'
  [../]
  [./vol_heat_source]
    type = HeatSource
    block = pellet
    function = q_heat
    variable = temperature
    save_in = 'saved_t'
  [../]
[]

[AuxVariables]
  [./power_density]
    block = pellet
  [../]
  [./fission_rate]
    block = pellet
  [../]
  [./fast_neutron_flux]
    block = clad
  [../]
  [./fast_neutron_fluence]
    block = clad
  [../]
  [./creep_rate_aux]
    order = CONSTANT
    family = MONOMIAL
  [../]

  # AuxVariables for Reference Residual Problem
  [./saved_x]
  [../]
  [./saved_y]
  [../]
  [./saved_t]
  [../]

  # Aux variables for output
  [./energy_density]
    block = pellet
    initial_condition = 0.0
  [../]
  [./gap_cond]
    order = CONSTANT
    family = MONOMIAL
  [../]
  [./coolant_htc]
    order = CONSTANT
    family = MONOMIAL
    block = clad
  [../]
  [./coolant_temperature]
    order = CONSTANT
    family = MONOMIAL
    block = clad
  [../]
  [./cumulative_damage_index]
    order = CONSTANT
    family = MONOMIAL
    block = clad
  [../]
  [./a_crack]
    order = CONSTANT
    family = MONOMIAL
    block = clad
  [../]
  [./element_failed]
    order = CONSTANT
    family = MONOMIAL
  [../]
  [./eutectic_thickness]
    order = CONSTANT
    family = MONOMIAL
    block = clad
  [../]
[]

[AuxKernels]
  [./calc_fission_rate]
    type = FissionRateFromPowerDensity
    block = pellet
    function = q_heat
    variable = fission_rate
    execute_on = 'initial timestep_end'
  [../]
  [./fast_neutron_flux]
    type = FastNeutronFluxAux
    variable = fast_neutron_flux
    block = clad
    factor = 0.6e19
    execute_on = 'initial timestep_end'
  [../]
  [./fast_neutron_fluence]
    type = FastNeutronFluenceAux
    variable = fast_neutron_fluence
    block = clad
    fast_neutron_flux = fast_neutron_flux
    execute_on = 'initial timestep_end'
  [../]
  [./calc_power_density]
    type = FunctionAux
    block = pellet
    function = q_heat
    variable = power_density
    execute_on = 'initial timestep_end'
  [../]
  [./calc_energy_density]
    type = VariableTimeIntegrationAux
    block = pellet
    order = 2
    variable_to_integrate = power_density
    variable = energy_density
    execute_on = timestep_end
  [../]
  # Hoop stress_zz.
  [./creep_rate_aux]
    type = MaterialRealAux
    property = creep_rate
    variable = creep_rate_aux
    execute_on = timestep_end
  [../]

  [./conductance]
    type = MaterialRealAux
    property = gap_conductance
    variable = gap_cond
    boundary = 10
  [../]
  [./coolant_temperature]
    type = MaterialRealAux
    property = coolant_temperature
    variable = coolant_temperature
    boundary = 2
  [../]
  [./coolant_htc]
    type = MaterialRealAux
    property = coolant_channel_htc
    variable = coolant_htc
    boundary = 2
  [../]
  [./cdf_amount]
    block = clad
    type = MaterialRealAux
    property = cdf_failure
    variable = cumulative_damage_index
  [../]
  [./a_crack]
    type = MaterialRealAux
    block = clad
    property = a_radius
    variable = a_crack
    execute_on = 'timestep_end'
  [../]
  [./failed_element]
    type = MaterialRealAux
    property = failed
    variable = element_failed
  [../]
  [./fcci_eutectic_in]
    execute_on = timestep_end
    type = EutecticThicknessFCCI
    boundary = 5
    variable = eutectic_thickness
    temperature = temperature
  [../]
  [./fcci_eutectic_out]
    execute_on = timestep_end
    type = EutecticThicknessFCCI
    boundary = 2
    variable = eutectic_thickness
    temperature = temperature
  [../]
[]

[Postprocessors]
  [./_dt]
    type = TimestepSize
  [../]
  [./num_lin_it]
    type = NumLinearIterations
  [../]
  [./num_nonlin_it]
    type = NumNonlinearIterations
  [../]
  [./tot_lin_it]
    type = CumulativeValuePostprocessor
    postprocessor = num_lin_it
  [../]
  [./tot_nonlin_it]
    type = CumulativeValuePostprocessor
    postprocessor = num_nonlin_it
  [../]

  [./fis_gas_released]
    type = ElementIntegralFisGasProduce
    block = pellet
    execute_on = ’initial timestep_end’
  [../]
  [./ave_temp_interior]
    type = SideAverageValue
    boundary = 9
    variable = temperature
    execute_on = ’initial linear’
  [../]
  [./avg_clad_temp]
    type = ElementAverageValue
    variable = temperature
    outputs = ’exodus csv’
    block = clad
  [../]
  [./avg_fuel_temp]
    type = ElementAverageValue
    variable = temperature
    outputs = ’exodus csv’
    block = pellet
  [../]
  [./max_outer_clad_temp]
    type = NodalExtremeValue
    variable = temperature
    value_type = max
    boundary = 2
  [../]
  [./peak_clad_temp]
    type = ElementExtremeValue
    variable = temperature
    value_type = max
    block = clad
    outputs = ’exodus csv’
  [../]
  [./peak_fuel_temp]
    type = ElementExtremeValue
    variable = temperature
    value_type = max
    block = pellet
    outputs = ’exodus csv’
  [../]
  [./peak_outer_fuel_temp]
    type = NodalExtremeValue
    variable = temperature
    value_type = max
    boundary = 10
  [../]
  [./peak_coolant_temperature]
    type = ElementExtremeValue
    variable = coolant_temperature
    value_type = max
    block = clad
    outputs = 'all'
  [../]
  [./flux_from_clad]
    type = SideFluxIntegral
    variable = temperature
    boundary = 5
    diffusivity = thermal_conductivity
    outputs = 'exodus csv'
  [../]
  [./flux_from_fuel]
    type = SideFluxIntegral
    variable = temperature
    boundary = 10
    diffusivity = thermal_conductivity
    outputs = 'exodus csv'
  [../]
  [./max_power_density]
    type = ElementExtremeValue
    variable = power_density
    value_type = max
    block = pellet
  [../]
  [./avg_power_density]
    type = ElementAverageValue
    variable = power_density
    block = pellet
    outputs = 'exodus csv'
  [../]
  [./total_deposit_pin_energy]
    type = ElementIntegralVariablePostprocessor
    variable = energy_density
    block = pellet
  [../]
  [./clad_inner_vol]
    type = InternalVolume
    boundary = 7
    outputs = 'exodus'
  [../]
  [./pellet_volume]
    type = InternalVolume
    boundary = 8
    outputs = 'exodus'
    execute_on = 'initial timestep_end'
  [../]
  [./gas_volume]
    type = InternalVolume
    boundary = 9
    execute_on = 'initial linear'
    outputs = 'exodus csv'
    addition = -2.0e-6 # m3; assuming all bond sodium displaced by closed
        gap
  [../]
  [./clad_fuel_gap]
    type = NodalMaxValue
    variable = penetration
    boundary = 10
    outputs = 'exodus csv'
  [../]
  [./max_eutectic_pen]
    type = ElementExtremeValue
    variable = eutectic_thickness
    block = clad
    value_type = max
    execute_on = timestep_end
    outputs = 'all'
  [../]
[]

[Contact]
  [./pellet_clad_mechanical]
    master = 5
    slave = 10
    penalty = 1e12
    model = frictionless
    normalize_penalty = true
    tangential_tolerance = 1e-3
    normal_smoothing_distance = 0.1
    system = constraint
  [../]
[]

[ThermalContact]
  [./thermal_contact]
    type = GapHeatTransfer
    variable = temperature
    master = 5
    slave = 10
    quadrature = true
    gap_conductivity = 73.70 # W/mK
    min_gap = 1e-3 # m
  [../]
[]

[BCs]
  [./no_x_all]
    type = DirichletBC
    variable = disp_x
    boundary = 12
    value = 0.0
  [../]
  [./no_y_fuel]
    type = PenaltyDirichletBC
    penalty = 1e10
    variable = disp_y
    boundary = 20
    value = 0.0
  [../]
  [./no_y_clad]
    type = DirichletBC
    variable = disp_y
    boundary = 1
    value = 0.0
  [../]

  [./Pressure]
    [./coolantPressure]
      boundary = '1 2 3'
      factor = 455054.0 # Pa
    [../]
  [../]
  [./PlenumPressure]
    [./plenumPressure]
      boundary = 9
      initial_pressure = 1.41343e6 # Pa
      initial_temperature = 309.0 # K
      startup_time = 0.0
      R = 8.3143 # J/mol K
      temperature = ave_temp_interior
      volume = gas_volume
      output = plenum_pressure
      material_input = fis_gas_released
      output_initial_moles = plenum_moles
    [../]
  [../]
[]

[CoolantChannel]
  [./convective_clad_surface]
    boundary = '1 2 3'
    variable = temperature
    inlet_temperature = 586.0 # K
    inlet_pressure = 455054.0 # Pa
    inlet_massflux = 4520.72 # kg/m^2 sec
    coolant_material = sodium # Do NOT add water...
    flow_area = 2.22e-5 # m^2
    hydraulic_diameter = 2.057e-3 # m
    heated_perimeter = 1.835e-2 # m
    heated_diameter = 4.84e-3 # m
    number_axial_zone = 50
    htc_correlation_type = 3 # Seban's for sodium
    compute_enthalpy = true # Material property: coolant_enthalpy
  [../]
[]

[Materials]
  # Fuel Slug Properties
  [./set_porosity]
    type = GenericConstantMaterial
    block = pellet
    prop_names = porosity
    prop_values = 0.31 # fractional porosity
  [../]
  [./set_mat_fission_rate]
    type = ParsedMaterial
    block = pellet
    args = 'fission_rate'
    function = 'fission_rate * 1.0'
    f_name = ’fission_rate’
  [../]
  [./melted]
    type = GenericMaterialFailure
    block = pellet
    compared = greater_than
    variable_check = temperature
    constant_criteria = 1600.0 #K
  [../]
  [./for_youngs]
    type = ParsedMaterial
    block = pellet
    outputs = all
    output_properties = ’youngs_modulus’
    f_name = ’youngs_modulus’
    args = ’temperature’
    material_property_names = ’porosity’
    constant_names = ’T_limit E_U T_meltU W_Zr W_Pu B_E Ta_start Ta_end’
    constant_expressions = ’1550.0 1.6e11 1405.0 0.0976 0.00115 1.2 923.0
        973.0’
    function = ’E_p := 1.0 - B_E * porosity; E_W := (1.0 + 0.17 * W_Zr) /
        (1.0 + 1.34 * W_Zr) - W_Pu; T_act := if(temperature > T_limit,
        T_limit, temperature); x_smooth := (T_act - Ta_start) / (Ta_end -
        Ta_start); f_smooth := if(T_act < Ta_start, 0.0, if(T_act > Ta_end,
         1.0, 6.0 * pow(x_smooth, 5) - 15.0 * pow(x_smooth, 4) + 10.0 * pow
        (x_smooth, 3))); E_T := 1.0 - 1.06 * (T_act - 588.0) / T_meltU -
        f_smooth * 0.3 * (1.0 - 1.06 * (Ta_end - 588.0) / T_meltU); E_U *
        E_T * E_p * E_W’
  [../]
  [./for_poissons]
    type = ParsedMaterial
    block = pellet
    outputs = all
    output_properties = ’poissons_ratio’
    f_name = ’poissons_ratio’
    args = ’temperature’
    material_property_names = ’porosity’
    constant_names = ’T_limit nu_U T_meltU W_Zr B_nu’
    constant_expressions = ’1550.0 0.24 1405.0 0.0976 0.8’
    function = ’nu_p := 1.0 - B_nu * porosity; nu_W := (1.0 + 3.4 * W_Zr)
        / (1.0 + 1.9 * W_Zr); T_act := if(temperature > T_limit, T_limit,
        temperature); nu_T := 1.0 + 1.2 * (T_act - 588.0) / T_meltU; nu_U *
        nu_p * nu_W * nu_T’
  [../]
  [./slug_elasticity_tensor]
    type = ComputeVariableIsotropicElasticityTensor
    block = pellet
    args = temperature
    youngs_modulus = youngs_modulus
    poissons_ratio = poissons_ratio
  [../]
  [./slug_stress] # Strain done by Action
    type = ComputeMultipleInelasticStress
    tangent_operator = nonlinear
    inelastic_models = ’slug_creep’
    block = pellet
  [../]
  [./slug_creep]
```

```
  type = UPuZrCreepUpdate
  block = pellet
  temperature = temperature
  porosity = porosity
  max_inelastic_increment = 1e-3
[../]
[./slug_thermal_expansion]
  type = ComputeThermalExpansionEigenstrain
  block = pellet
  thermal_expansion_coeff = 1.18e-5
  temperature = temperature
  eigenstrain_name = slug_thermal_strain
[../]
[./slug_thermal]
  type = ThermalUPuZr
  block = pellet
  Na_depth = 0.0673
  fuel_outer_radius = 2.54e-3
  spheat_model = savage
  temp = temperature
  thcond_model = billone
  outputs = all
  output_properties = thermal_conductivity
[../]
[./slug_density]
  type = Density
  block = pellet
  density = 15700.0 # kg/m3
[../]
[./slug_fission_gas_release]
  block = pellet
  type = FgrUPuZr
  fission_rate = fission_rate
[../]

# Cladding Properties
[./clad_elasticity_tensor]
  type = ComputeIsotropicElasticityTensor
  youngs_modulus = 1.88e11
  poissons_ratio = 0.236
  block = clad
[../]
[./clad_stress] # Strain done by Action
  type = ComputeMultipleInelasticStress
  block = clad
  tangent_operator = nonlinear
  inelastic_models = 'clad_creep'
[../]
[./clad_fast_flux]
  type = FastNeutronFlux
  block = clad
  factor = 0.6e19
[../]
[./clad_creep]
  type = HT9CreepUpdate
  block = clad
  temperature = temperature
[../]
```

```
  [./clad_thermal_expansion]
    type = ComputeThermalExpansionEigenstrain
    block = clad
    thermal_expansion_coeff = 1.2e-5
    temperature = temperature
    eigenstrain_name = clad_thermal_eigenstrain
  [../]
  [./clad_thermal]
    type = ThermalHT9
    block = clad
    temp = temperature
  [../]
  [./clad_density]
    type = Density
    block = clad
    density = 7874.0 # kg/m3
  [../]

  [./failclad]
    block = clad
    type = FailureCladHT9
    hoop_stress = stress_zz
    temperature = temperature
    method = cdf_short
  [../]
[]

[Preconditioning]
  [./SMP]
    type = SMP
    full = true
    solve_type = ’PJFNK’
    petsc_options = ’-snes_ksp_ew -snes_converged_reason’
    petsc_options_iname = ’-pc_type -pc_factor_mat_solver_package -
       ksp_gmres_restart’
    petsc_options_value = ’lu       superlu_dist                  51’
  [../]
[]

[Executioner] # Kind of ominous, eh?
  type = Transient
  line_search = ’none’

  l_max_its = 100
  l_tol = 1e-3

  nl_max_its = 200
  nl_rel_tol = 1e-6
  nl_abs_tol = 1e-9

  start_time = 0.0
  end_time = 25.0 # seconds
  dtmin = 0.0001
  dtmax = 1.0

  [./TimeStepper]
    type = FunctionDT
    function = dt_fun
  [../]
  [./Quadrature]
    order = FIFTH
    side_order = SEVENTH
  [../]
[]

[Outputs]
  perf_graph = true
  interval = 1
  csv = true
  [./console]
    type = Console
    output_linear = true
    output_nonlinear = true
  [../]
  [./exodus]
    type = Exodus
  [../]
  [./vec_coolant_htc]
    type = CSV
    file_base = vec_htc
    time_data = true
  [../]
  [./vec_coolant_Tinf]
    type = CSV
    file_base = vec_Tinf
    time_data = true
  [../]
  [./Checkpoint]
    type = Checkpoint
    num_files = 3
  [../]
[]

[VectorPostprocessors]
  [./coolant_htc_vals]
    type = SideValueSampler
    boundary = 2
    variable = coolant_htc
    sort_by = y
    outputs = ’vec_coolant_htc’
    contains_complete_history = false # Need to compile after
  [../]
  [./coolant_Tinf_vals]
    type = SideValueSampler
    boundary = 2
    variable = coolant_temperature
    sort_by = y
    outputs = ’vec_coolant_Tinf’
    contains_complete_history = false # Need to compile after
  [../]
[]
# vi:filetype=moose_fw
```