INL/EXT-21-62507

Assessment Approach to Advanced Fuel Models

Technical Report

Stephen Novascone, Daniel VanWasshenova, Albert Casagranda, Gyanender Singh, and Pierre-Clement Simon

Idaho National Laboratory

Idaho National Laboratory

INL is a U.S. Department of Energy National Laboratory operated by Battelle Energy Alliance, LLC

DISCLAIMER

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, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness, of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the U.S. Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the U.S. Government or any agency thereof.

Assessment Approach to Advanced Fuel Models

Technical Report

Stephen Novascone, Daniel VanWasshenova, Albert Casagranda, Gyanender Singh, and Pierre-Clement Simon

Idaho National Laboratory

15 September 2021

Idaho National Laboratory
Computational Mechanics and Materials Department
Idaho Falls, Idaho 83415

http://www.inl.gov

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

Page intentionally left blank

Abstract

This document constitutes completion of the NEAMS milestone M3MS-21IN0201016, which is titled: *Assessment of advanced fuel improvements*. In this report we present: (1) simulations of separate effects creep tests for uranium alloy and HT-9, (2) examples of standardizing and updating metallic fuel assessments to include the latest models and key output figures of merit, which is coordinated with LANL to better evaluate advanced models, (3) examples of how the FIPD database at ANL is integrated into metallic fuel assessment cases, and finally (4) we document fuel-specific figures of merit and the BISON review process for assessments with best practices for preparing and reviewing finite element simulations in the appendix.

Acknowledgments

This manuscript has been authored by Battelle Energy Alliance, LLC, under Contract No. DE-AC07-05ID14517 with the U.S. Department of Energy. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for U.S. Government purposes.

This research made use of the resources of the High Performance Computing Center at Idaho National Laboratory, which is supported by the Office of Nuclear Energy of the U.S. Department of Energy and the Nuclear Science User Facilities under Contract No. DE-AC07-05ID14517.

Contents

List of Figures

List of Tables

1. Introduction

As mechanistic models of advanced fuel types emerge, assessing performance in separate effects and integrated simulations must be done in parallel for all emerging fuel types and cladding models. Separate effects experiments such as diffusion couple, tube burst, and swelling are key to determining how well individual material models perform. Two such separate effect experiments are presented here for evaluating creep models in uranium alloy and HT-9 cladding. Examples of standardizing and updating metallic fuel assessments to include the latest models and key output figures of merit are also presented. In addition, the Experimental Breeder Reactor-II experiments and legacy data will be used in developing additional metallic fuel assessments. This work was coordinated with Argonne National Laboratory assessment activities and advanced model development at Los Alamos National Laboratory.

This report also documents ideas for developing assessment cases for the BISON fuel performance code, along with features that should be included in those models. This is the first step toward creating a robust system for incorporating simulations of advanced fuel types as well as separate effects experiments supporting advanced reactor designs in BISON. These simulations, along with the comparison of the numerical solution to experimental results, will provide the basis for validating the code and increasing confidence in the use of BISON for fuel design.

2. Background

The current light-water reactor (LWR) assessment models in BISON were developed over numerous years by various individuals using different methods. This approach produced a significant number of assessment cases for BISON, but the input files lacked consistency and were not updated uniformly as the material models and numerical methods in BISON evolved. Therefore, the LWR assessment experience is being used to inform the recommendations in this report. A consistent methodology must be established for determining when an assessment case should be added to the BISON repository, how the model should be structured, and how the solution should be evaluated compared to experimental data. This document represents an initial attempt to create such a procedure.

3. Separate Effects Tests

Separate effects tests are simulations of experiments focused on individual components of a fuel pin (i.e., an experiment that focus on fuel or cladding in isolation). Developing a robust set of separate effects simulations will result in better understanding of the individual material models. These simulations are also a good template to test new material models. Having a broad set of separate effects simulations improves understanding of material model behavior and provide a more effective validation base for fuel performance codes.

3.1 Metallic fuel creep

The thermomechanical performance of the fuel is the most important feature in an integrated fuel pin simulation. In this section, we take a closer look at the creep behavior of metallic fuel.

A constitutive model that captures the combined thermal and irradiation creep response of uranium-alloy fuel is described in the Metallic Fuels Handbook (MFH) [1]:

$$\dot{\epsilon} = A_1 \left(1 + 7.9p + 470p^2\right) \exp\left(\frac{-Q_1}{RT}\right) \sigma + A_2 \left(1 - p^{0.67}\right)^{-4.5} \exp\left(\frac{-Q_2}{RT}\right) \sigma^{4.5} + A_3 \dot{F} \sigma \tag{1}$$

where $\dot{\epsilon}$ is the creep rate (1/s), σ is the effective (Mises) stress (MPa), T is the temperature (K), p is the porosity fraction (-), $\dot{F}$ is the volumetric fission rate (fissions/cm^3/s), Q_i are the activation energies (cal/mol), R is the universal gas constant (1.987 cal/mol/K), and A_{1-3} are empirical constants given as $A_1 = 5 \times 10^3$, $A_2 = 6$, and $A_3 = 7.7 \times 10^{-23}$.

The terms on the right-hand side of Eq. 1 respectively represent alpha-U phase diffusion, dislocation, and irradiation creep. The activation energies for the thermal creep terms (Q_1 and Q_2) are given as $Q_1 = Q_2 = 52\,000$ cal/mol.

These U and Pu alloys have a phase change temperature of 923.15 K. Above this temperature, the creep mechanism is considered to be dominated by gamma-phase dislocation, so the first two terms in 1 are replaced by the first term in the following equation:

$$\dot{\epsilon} = A_4 \left(1 - p^{0.67}\right)^{-3} \exp\left(\frac{-Q_3}{RT}\right) \sigma^3 + A_3 \dot{F} \sigma \tag{2}$$

where $A_4 = 8.0 \times 10^{-2}$ and $Q_3 = 28\,500$ cal/mol.

Summarized in MFH [1] and described in more detail in Foote [2] (UZr) and Saller [3] (UPuZr) are results of creep tests on uranium alloy. In the tests, the specimen is subjected to an effective stress of 9.8 MPa while the temperature varies from 870 K to 1100 K, and the creep strain rate is recorded. The results of the experiment are shown in Figure 1 and where used to generate Equations 1 and 2, which have been added to BISON in prior years.

When these equations where added to BISON, there were accompanying regression tests and documentation. However, the regression tests were limited to test basic functionality and convergence. Part of the purpose of this milestone is to revisit material models and perform a more comprehensive evaluation of the installment of the model. In that vein, a BISON simulation of the experiment documented in [2] and [3] was run and compared to the experiment measurements. For reference, Figure 1 also shows plots of equations 1 and 2 that vary over the same temperature range as the experiment and with porosity values of 0, 0.1, and 0.2. The fission rate is set to zero in this case. The BISON simulation matched the inputs to the equations.

Figure 1 shows reasonable agreement between experiment measurements and the output from the equations and their installation in BISON. One key difference between the BISON result and the equations and measurements is the transition between creep mechanisms (the transition from Equation 1 to 2 as the phase changes due to the rise in temperature). The data and corresponding equations show an instant change from one phase to another. Such an abrupt transition would cause convergence trouble, so a smoothing function was used around the phase transition temperature (923 K). The take-away from this section is that the uranium alloy creep model from [1] was implemented correctly as indicated by its expected behavior in the range of experiment measurements. Further validation is required to determine the general-use value of the model.

The importance of accurately representing the mechanical response of the fuel motivates further consideration of the relevant literature. A good overview of metallic fuel properties is found in Janney [4], which points out one of the few experiments performed that sheds light on the inelastic response of uranium alloy as described by McDeavitt [5]. In the experiment uranium-zirconium pellets are constructed by sintering into a right circular cylinder, then mechanically loaded by the application of a hydrostatic external pressure at a temperature of 973 K. The change in the height of the cylindrical pellet due to the pressure was recorded. A plot digitization of the measurements presented in [5] are shown in Figure 2. McDeavitt [5] reports that the mechanical response of the pellet can be described by the coupling of grain boundary diffusion and creep cavitation.

It is evident that an accurate engineering scale model should account for this behavior because the mechanical load on the fuel in a sodium-cooled fast reactor is similar to the experiment described in [5]. To show how the creep model used in BISON compares to this experiment, a demonstration simulation was performed. In the simulation a hollow spherical geometry was used to account for the uniform loading and the trapped gas within the pellet from the sintering process. The volume of the hollow sphere is equivalent to the solid cylindrical pellet and the volume within the sphere is equivalent to the gas volume trapped in the pellet after it was sintered to 90% theoretical density. The mechanical loads applied to the sphere are the external hydrostatic pressure and an internal pressure that corresponds to the sintering temperature,

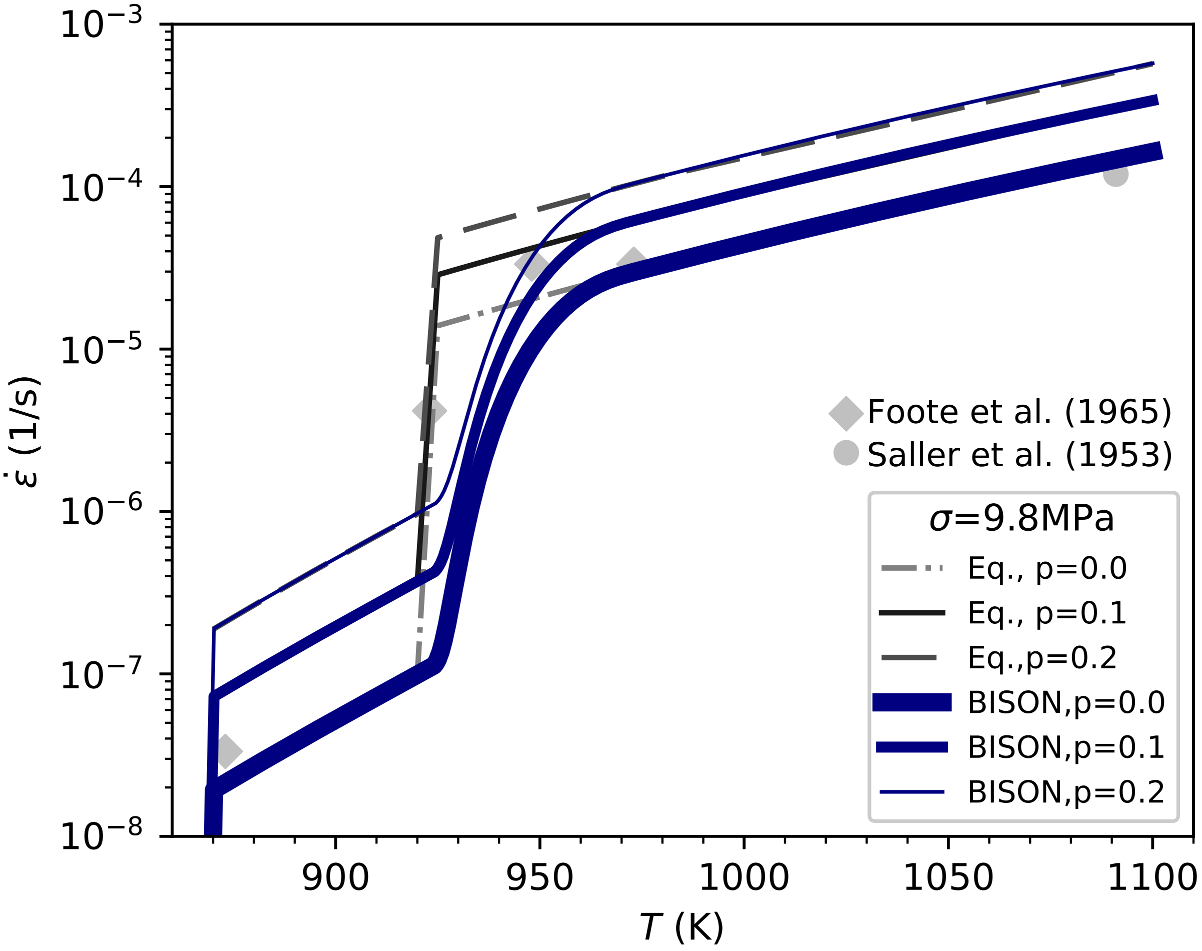

Figure 1. The steady-state creep rates as a function of temperature and porosity (*experimental data from [2, 3]*).

pressure, and gas volume of the pellet (1100°C, 3.5 MPa, and volume that corresponds to 10% theoretical density). The pressure calculated using these numbers in the ideal gas law resulted in a internal pressure of 2.3 MPa, which is applied to the inner volume of the sphere and responds to temperature and volume changes via the ideal gas law. In this way, the pressure within the sphere increases as the hydrostatic load is applied and the volume of the sphere is correspondingly reduced. The results of this simulation and the experiment measurements for the highest and lowest pressures of the experiment (17.3 and 3.5 MPa respectively) are shown in Figure 3, where it is clear, and expected, that the creep model in BISON is not accurately representing the mechanical response of the pellet. Allowing for the fact that the simulation is ill-posed as an accurate representation of the experiment, the calculation results show that the steep drop-off in strain rate as time increases is not captured by the creep model. This fact indicates that a constitutive

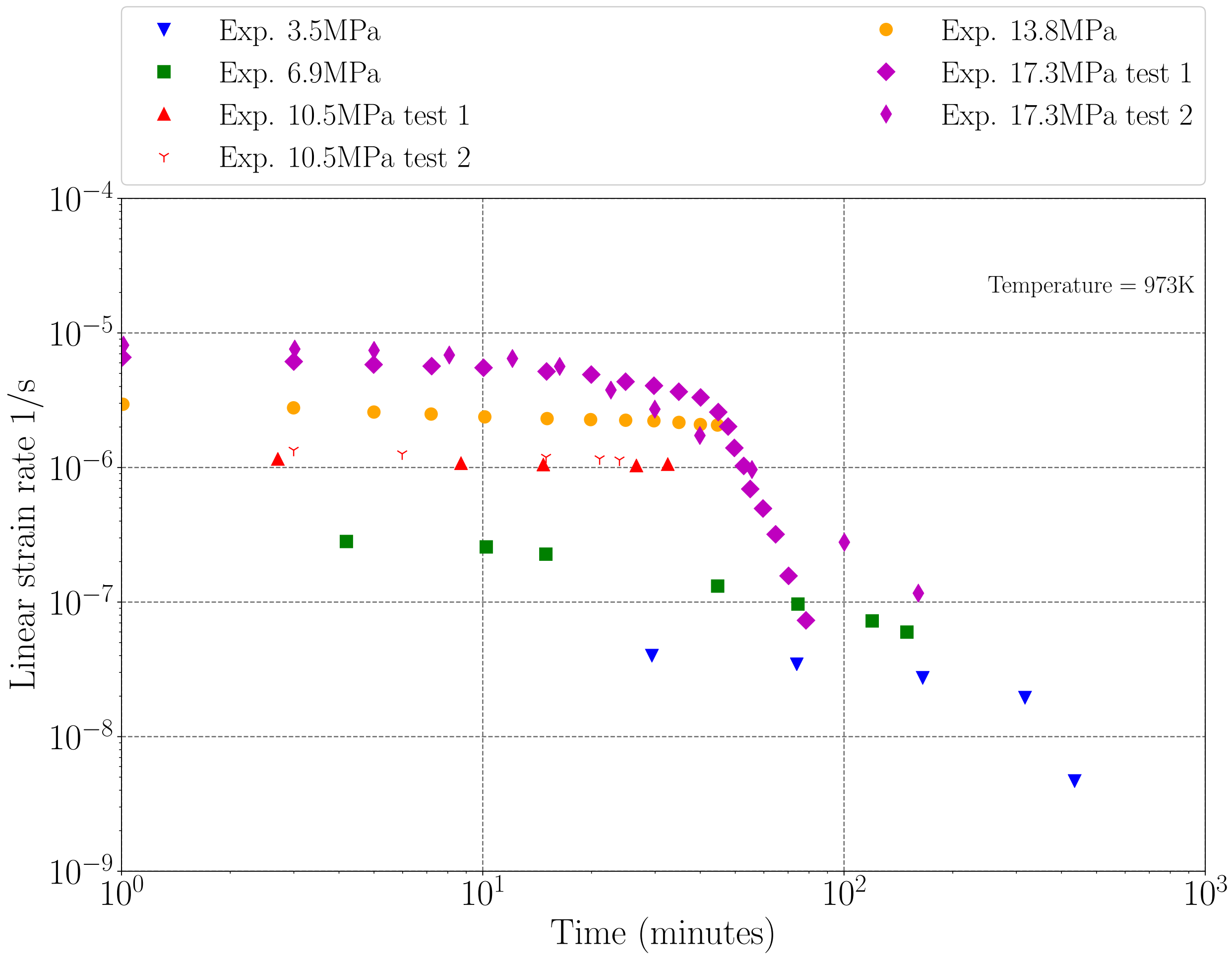

Figure 2. Strain rate vs. time for varying hydrostatic pressure of U-10Zr metallic fuel alloy from McDeavitt [5].

model capable of simulating the coupling between grain boundary diffusion and creep cavitation is required. A model, such as the MOOSE class ADViscoplasticityStressUpdate, or a reduced order model may account for the physics and will be the subject of investigation in FY 2022.

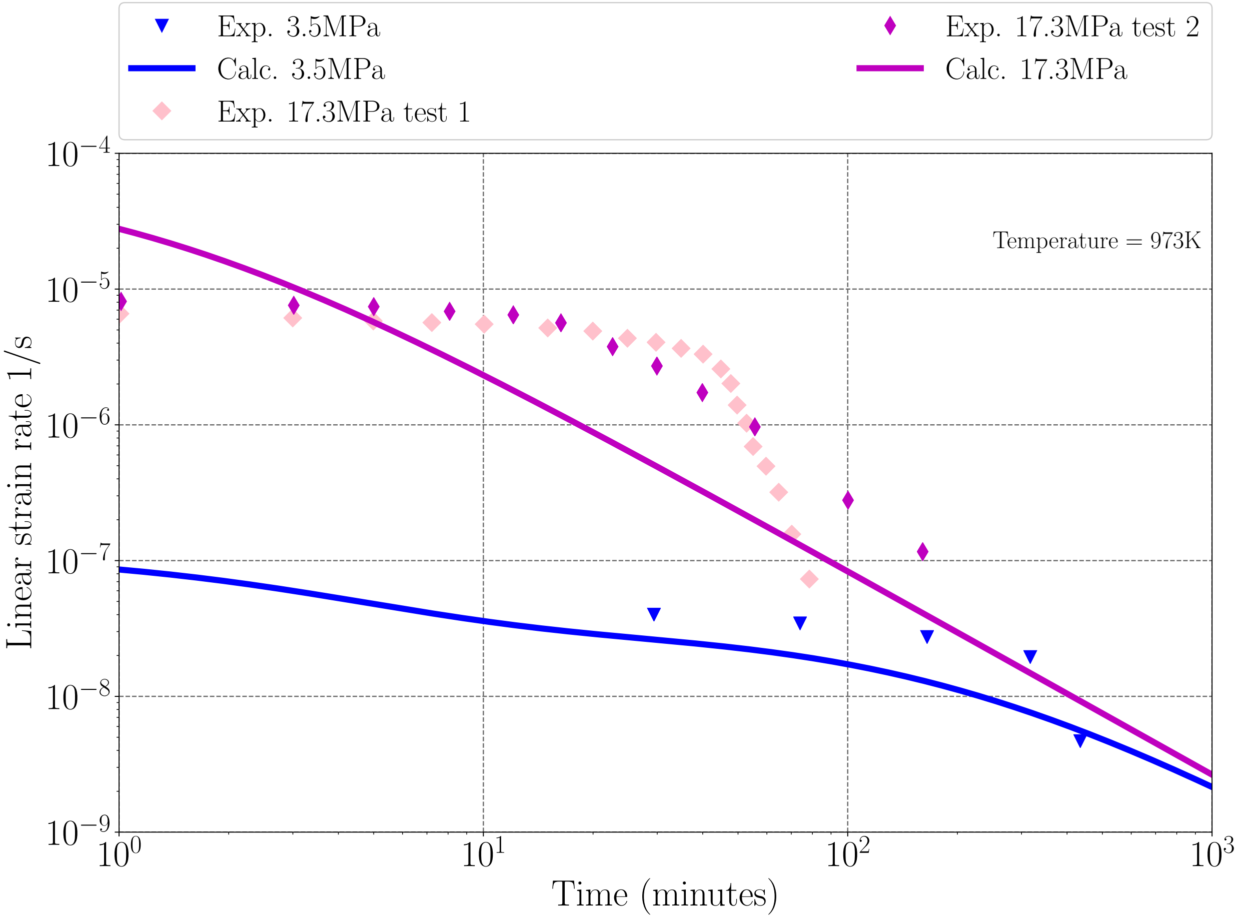

Figure 3. Strain rate vs. time for varying hydrostatic pressure of U-10Zr metallic fuel alloy from McDeavitt [5] with BISON calculations that are only accounting for the creep response and neglecting the hot isostatic pressings response.

3.2 HT-9 Cladding

The cladding material typically used in fast reactor applications is the martensitic stainless steel called HT-9. The equations used to represent the creep response of HT-9 in BISON are presented here.

The primary creep rate $\dot{\epsilon}_{p,mfh}$ from [1] is calculated by:

$$\dot{\epsilon}_{p,mfh} = \left[13.4 \exp\left(-\frac{15027}{RT}\right)\sigma + 8.43 \times 10^{-3} \exp\left(-\frac{26451}{RT}\right)\sigma^4 + 4.08 \times 10^{18} \exp\left(-\frac{89167}{RT}\right)\sigma^{0.5}\right] \cdot 1.6 \times 10^{-8} \exp\left(1.6 \times 10^{-6} t\right), \tag{3}$$

where T is temperature in K, $R = 1.987$ cal/K/mol, σ is the von Mises stress in MPa, and t is the simulation time in s.

The secondary creep rate $\dot{\epsilon}_{s,mfh}$ from [1] is calculated by:

$$\dot{\epsilon}_{s,mfh} = 1.17 \times 10^{7} \exp\left(-\frac{83142}{RT}\right)\sigma^2 + 8.33 \times 10^{7} \exp\left(-\frac{108276}{RT}\right)\sigma^5, \tag{4}$$

where T is temperature in K, $R = 1.987$ cal/K/mol, and σ is the von Mises stress in MPa.

The irradiation creep rate $\dot{\epsilon}_{irr,mfh}$ from [1] is calculated by:

$$\dot{\epsilon}_{irr,mfh} = \left[1.83 \times 10^{-6} + 2.59 \times 10^{12} \exp\left(-\frac{73000}{RT}\right]\right)\phi\sigma^{1.3}, \tag{5}$$

where T is temperature in K, $R = 1.987$ cal/K/mol, σ is the von Mises stress in MPa, and ϕ is the fast flux in (10^{22} n/cm^2).

The primary creep rate $\dot{\epsilon}_{p,rky}$ from [6] is calculated by:

$$\dot{\epsilon}_{p,rky} = 0.01 * \left(10^{0.52 - 2647.31/T}\sigma^{1.09 + 31.48/T}\right) m \exp(-mt), \tag{6}$$

where T is temperature in K, σ is the von Mises stress in MPa, t is time in s, and $m = -2.11 \times 10^{-6}$ 1/s.

The secondary creep rate $\dot{\epsilon}_{s,rky}$ from [6] is calculated by:

$$\dot{\epsilon}_{s,rky} = 10^{-5.58 - 5562.28/T}\sigma^{1.5} \tag{7}$$

where T is temperature in K and σ is the von Mises stress in MPa.

An example of a cladding separate effects test is an irradiated thermomechanical pressure load test of HT-9, a martensitic stainless steel used as cladding material in sodium-cooled fast reactors [7]. The HT-9 test specimen a hollow right-circular cylinder that is internally pressurized at various temperature and fast neutron fluence. Under these conditions, measurements of permanent hoop strain are recorded at the corresponding values of hoop stress, temperature, and fast neutron fluence. A BISON fuel performance model that represents the salient geometry, material models and loading conditions is used to calculate permanent hoop strain and the results compared to the experiment measurements.

The finite element mesh is designed to reproduce the loading conditions of the experiment with 2D-RZ coordinates and an inner and outer diameter of 10.75 mm and 9.3 mm, respectively. The cylinder length is 0.5 m. The mesh density is four elements through the thickness of the cylinder and 100 elements along the length resulting in a maximum aspect ratio of 2.8. Spatial and temporal resolution studies show that the mesh density and time step size are adequate for solution convergence.

This report focuses on three experiments with the following loading, thermal, and irradiation conditions. In the first test, the temperature is set to 425^{o}C, the hoop stress is 110 MPa, and fluence ranges from 0 to $1{\cdot}10^{27}$ n/m^2. The second test temperature is 590^{o}C at 55 MPa with the same range in fluence. In these first two tests, no primary creep model is used and the RKY secondary thermal creep model is used with the MFH irradiation creep model. Rounding-out the cladding separate effects tests is a stress ramp test, in which the temperature is 540 ^{o}C, the fluence remains the same as tests one and two, and the hoop stress varies from 0 to about 100 MPa. This third test uses the same creep models as Tests 1 and 2, with the addition of the RKY primary creep model. Results from these calculations and the corresponding measurements from the experiment are shown in Figures 4 - 6.

Observation of the results shown in Figures 4 - 6 lead to the conclusion that the HT-9 material models in BISON capture the general mechanical response, but could be significantly improved. Figures 4 and 5 show the material models over estimate hoop strain. For the conditions of Figure 4 with relatively low temperature (425^{o}C) and high hoop stress (110 MPa) the BISON calculation overestimates the permanent hoop strain by about 30%. In Figure 5, for relatively high temperature (590^{o}C) and low hoop stress (55 MPa) BISON overestimates final permanent hoop strain by about 20%. Surprisingly, as the hoop stress is increased from 0 to 100 MPa at 540^{o}C and $1{\cdot}10^{27}$ n/m^2 BISON appreciably *under estimates* permanent hoop strain through the range of stress values. The last points on Figure 6 show a difference of about 70% between the measurement and calculation. Given this large discrepancy and since the stress is constantly changing, a speculative conclusion is that the creep strain rate model in BISON is increasingly in error as the stress rises. This will be the subject of further investigation next year. Moreover, this simulation should be repeated with the new reduced order models from Los Alamos, which include the evolution of microstructure as the stress, temperature, and irradiation conditions change. The reduced order models have demonstrated a better comparison between EBR-II cladding strain measurements and calculations.

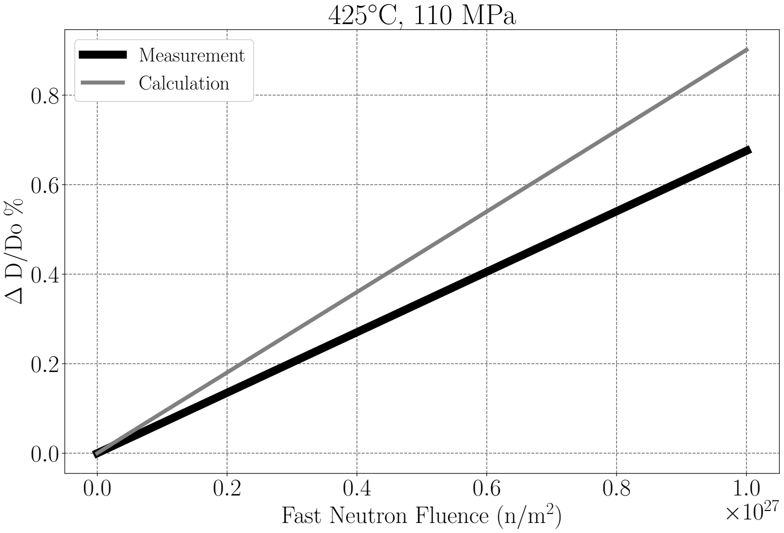

Figure 4. BISON calculations compared to measurements in [7] for permanent hoop strain versus increasing fast neutron fluence in HT-9 at 425^{o}C and 110 MPa.

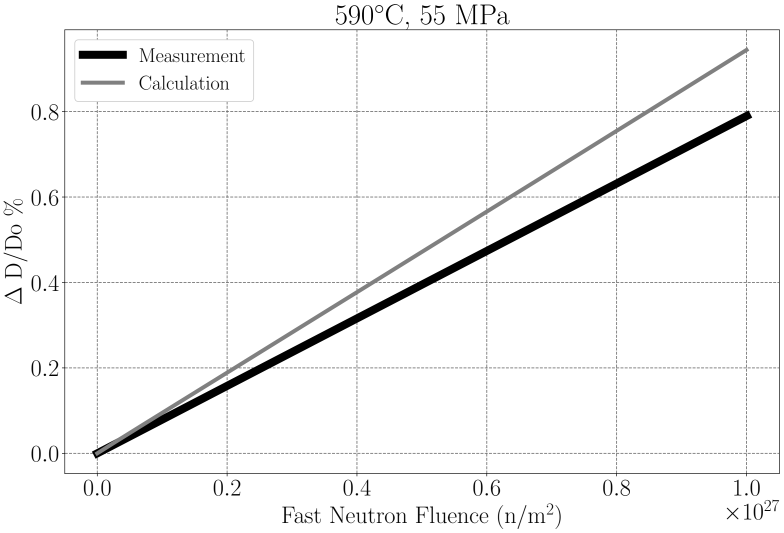

Figure 5. BISON calculations compared to measurements in [7] for permanent hoop strain versus increasing fast neutron fluence in HT-9 at 590^{o}C and 55 MPa.

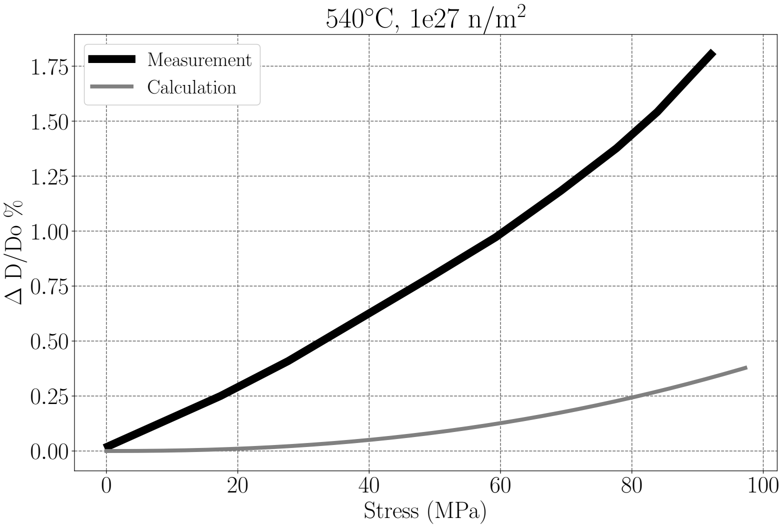

Figure 6. BISON calculations compared to measurements in [7] for permanent hoop strain versus increasing hoop stress in HT-9 at 540^{o}C and fast neutron fluence of 1e27 n/m^2

.

4. Metallic Fuel Assessment Standard

A large-scale standardization of input file creation for metallic fuel assessment cases was undertaken for this milestone following the guidelines discussed previously. In ensuring that this was both an effective and accurate tool, and user-friendly at the same time, extensive use of actions were developed. Action classes allow for a "behind-the-scenes" creation of the necessary input file blocks, for given conditions, greatly reducing input file length and complexity, as well as providing a standard.

The method of utilizing actions in standardizing BISON simulations is beneficial to end-users for two main reasons. The primary reason is that it allows a simpler method of creating input files with a reduced amount of code-specific knowledge. This allows a "plug-and-play" approach for beginning users in nuclear fuel simulations. The secondary reason is a more standard method of what and how simulations are run and how they can be achieved. By giving the user-specific methods for desired simulations potential differences are eliminated. The specific actions used in this effort are discussed below.

4.1 Nuclear Materials

As with all NuclearMaterial action classes, the type of, or lack of fission process involved is required. This required parameter describes the characteristic type of fission study that will occur. For metallic fuel-type simulations, this is simply labeled as **fission_operation = Normal**. This must be placed under the NuclearMaterials block heading and cannot be placed under sub-blocks such as UPuZr. It is assumed that all blocks under the NuclearMaterials block heading will be undergoing the same fission process (e.g., the fuel cannot be undergoing normal fission while the cladding is experiencing LOCA).

The NuclearMaterials actions also allow the creation of additional blocks outside of Materials, further reducing the complexity and length of input files. The creation of Tensor Mechanics Action, and thermal components (e.g., Kernels, Variables, and ICs) have been integrated into all NuclearMaterials. Additionally, common parameters can be applied (e.g., temperature or material outputs) under the NuclearMaterials tab, which are applied to all blocks that fall under this header.

4.1.1 UPuZr Fuel Blocks

This **NuclearMaterialUPuZr** action reduces BISON input file length by internally generating the materials required for simulating metallic nuclear fuel, specifically UPuZr. The user may elect to use the **Nuclear-**

MaterialUPuZr action to generate the necessary material classes for common metallic fuel simulations, with or without additional NuclearMaterial actions for cladding. All of the material blocks, generated with the default parameter settings, are shown in Table 1.

The flexibility of the action classes allows for continual refinement of what and how simulations are performed by BISON without end-users determining which blocks need to be utilized in correctly simulating specific models. As such, users obtain the standard methods through the input parameter **upuzr_models = 'FIPD Burnup Elastic Creep Swelling ThermalExpansion'**, with the currently available models listed.

Table 1. Material classes created by the NuclearMaterialUPuZr action.

Material classes created by the NuclearMaterialUPuZr action		
Created Classes	**Pre-Set Parameters**	**Block Name**
UPuZrFissionRate		fission_rate
UPuZrBurnup	burnup_name = burnup	burnup
UPuZrElasticityTensor		fuel_elasticity_tensor
ComputeMultipleInelasticStress		fuel_inelastic_stresx
UPuZrCreepUpdate	max_inelastic_increment = 1e-2	fuel_upuzrcreep

ComputeThermalExpansionEigenstrain	thermal_expansion_coeff = 1.18e-5
	eigenstrain_name = fuel_thermal_strain
		fuel_thermal_expansion

UPuZrGaseousEigenstrain	bubble_number_density = N_bubbles
	interconnection_initiating_porosity = 0.23
	interconnection_terminating_porosity = 0.25
	eigenstrain_name = gas_swelling_strain
		gas_swelling

BurnupDependentEigenstrain	eigenstrain_name = solid_swelling_strain	solid_swelling
UPuZrThermal		metal_fuel_thermal
Density	initial_fuel_density = 15800	fuel_density

UPuZrFissionGasRelease	fission_rate = fission_rate
	critical_porosity = 0.24
	fractional_fgr_initial = 0.8
	fractional_fgr_post = 1.0
		fission_gas_release

The eigenstrain have been automatically parsed and included by wrapping the TensorMechanics Master Action in side the NuclearMaterials following the naming convention created by this class.

4.1.2 HT9 Cladding Blocks

This NuclearMaterialHT9 action reduces BISON input file length by internally generating the Materials required for simulating metallic nuclear cladding, specifically HT9. The user may elect to use the **NuclearMaterialHT9** action to generate the necessary material classes for common metallic fuel simulations. All of the material blocks, generated with the default parameter settings, are shown in Table 2.

Again, as discussed previously, the required parameter for the type of fission describes the characteristic type of fission study that will occur. For metallic fuel type simulations, this is simply labeled as **fission_operation = Normal**. This must be placed under the NuclearMaterials block heading and cannot be placed under sub-blocks, such as HT9.

Table 2. Material classes created by the NuclearMaterialHT9 action.

Material classes created by the NuclearMaterialHT9 action		
Created Classes	**Pre-Set Parameters**	**Block Name**

ComputeIsotropicElasticityTensor	poissons_ratio = 0.3
	youngs_modulus = 7.5e10
		clad_elasticity_tensor

ComputeMultipleInelasticStress	tangent_operator = elastic
	inelastic_models = clad_zrycreep
		clad_stress

ZryCreepLimbackHoppeUpdate	absolute_tolerance = 1e-10
	max_iterations = 50
	fast_neutron_flux = fast_neutron_flux
	fast_neutron_fluence = fast_neutron_fluence
		clad_zrycreep

ZryThermalExpansionMATPROEigenstrain	burnup_function = burnup
	eigenstrain_name = clad_thermal_strain
		clad_thermal_expansion

ZryIrradiationGrowthEigenstrain	fast_neutron_fluence = fast_neutron_fluence
	eigenstrain_name = fuel_irradiation_strain
		clad_irradiation_swelling

Density	density = 6551.0	clad_density

HeatConductionMaterial	thermal_conductivity = 16.0
	specific_heat = 330.0
		clad_thermal

The eigenstrain have been automatically parsed and included by wrapping the TensorMechanics Master Action in side the NuclearMaterials following the naming convention created by this class.

4.1.3 Example Input Files

Use of action classes greatly help in the reduction of input file length and complexity. Through the use of the NuclearMaterials for metallic fuels, a standardization of how the inputs files are constructed is realized as well. As an example of this process, input files for the same tests parameters ”before” and ”after” are shown in Listings 1 and 2, respectively.

Through use of the NuclearMaterials action, full functionality is achieved with a great reduction on input file length. This reduction also ensures that the classes used to construct the simulation are the vetted and standard methods for the physical models being investigated.

Listing 1. Example of the input file length before reduction via `NuclearMaterialsUPuZr`
and `NuclearMaterialsHT9`

```
[Variables]
  [disp_x]
  []
  [disp_y]
  []
  [temp]
    initial_condition = 298
  []
[]
[AuxVariables]
  [porosity]
    order = CONSTANT
    family = MONOMIAL
  []
    [solid_swell]
    block = fuel
    order = CONSTANT
    family = MONOMIAL
  []
  [gas_swell]
    block = fuel
    order = CONSTANT
    family = MONOMIAL
  []
  [volumetric_strain]
    block = fuel
    order = CONSTANT
    family = MONOMIAL
  []
  [hoop_stress]
    order = CONSTANT
    family = MONOMIAL
  []
  [hoop_creep_strain]
    order = CONSTANT
    family = MONOMIAL
  []
  [hoop_elastic_strain]
    order = CONSTANT
    family = MONOMIAL
  []
  [total_hoop_strain]
    order = CONSTANT
    family = MONOMIAL
  []
  [func_val1] # Just for visualization
  []
  [func_val2] # Just for visualization
  []
[]
[Modules/TensorMechanics/Master]
  [fuel]
    strain = FINITE
    add_variables = true
    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'
    extra_vector_tags = 'ref'
    block = fuel
    eigenstrain_names = 'fuel_thermal_strain fuel_volumetric_strain'
  []
  [clad]
    strain = FINITE
    add_variables = true
    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'
    extra_vector_tags = 'ref'
    block = cladding
    eigenstrain_names = 'clad_thermal_eigenstrain'
  []
[]
[Kernels]
  [gravity]
    type = Gravity
    variable = disp_y
    value = -9.81
    extra_vector_tags = 'ref'
  []
  [heat]
    type = HeatConduction
    variable = temp
    extra_vector_tags = 'ref'
  []
  [heat_ie_f]
    type = HeatConductionTimeDerivative
    variable = temp
    extra_vector_tags = 'ref'
    block = fuel
    density_name = 16000
  []
    [heat_ie_c]
    type = HeatConductionTimeDerivative
    variable = temp
    extra_vector_tags = 'ref'
    block = cladding
    density_name = 7890
  []
  [heat_source]
    type = FissionRateHeatSource
    variable = temp
    fission_rate = fission_rate
    block = fuel
    extra_vector_tags = 'ref'
  []
[]
[AuxKernels]
  [gas_swell]
    type = MaterialRealAux
    variable = gas_swell
    property = gas_swelling
    execute_on = timestep_end
  []
  [solid_swell]
    type = MaterialRealAux
    variable = solid_swell
    property = solid_swelling
    execute_on = timestep_end
  []
  [volumetric_strain]
    type = RankTwoScalarAux
    rank_two_tensor = total_strain
    variable = volumetric_strain
    scalar_type = VolumetricStrain
    execute_on = timestep_end
    block = fuel
  []
  [hoop_stress]
    type = RankTwoAux
    rank_two_tensor = stress
    variable = hoop_stress
    index_j = 2
    index_i = 2
    execute_on = timestep_end
  []
  [hoop_creep_strain]
    type = RankTwoAux
    rank_two_tensor = creep_strain
    variable = hoop_creep_strain
    index_j = 2
    index_i = 2
    execute_on = timestep_end
    block = cladding
  []
  [hoop_elastic_strain]
    type = RankTwoAux
    rank_two_tensor = elastic_strain
    variable = hoop_elastic_strain
    index_j = 2
    index_i = 2
    execute_on = timestep_end
    block = cladding
  []
  [total_hoop_strain]
    type = RankTwoAux
    rank_two_tensor = total_strain
    variable = total_hoop_strain
    index_j = 2
    index_i = 2
    execute_on = timestep_end
    block = cladding
  []
  [func_val1]
    type = FunctionAux
    function = id_vpp_func
    variable = func_val1
    block = cladding
  []
  [func_val2]
    type = FunctionAux
    function = od_vpp_func
    variable = func_val2
    block = cladding
  []
[]
[Materials]
  [fission_rate]
    type = UPuZrFissionRate
    block = fuel
    rod_linear_power = power_history
    axial_power_profile = pwr_axial_peaking_factors
    use_metadata = true
    mesh_generator = fipd_mesh
    outputs = all
  []
  [fission_rate_elongate]
    type = UPuZrFissionRate
    block = cladding
    fission_rate_name = fission_rate
    rod_linear_power = power_history
    axial_power_profile = pwr_axial_peaking_factors_elongate
    use_metadata = true
    mesh_generator = fipd_mesh
    outputs = all
  []
  [burnup]
    type = UPuZrBurnup
    initial_X_Pu = ${fuel_pu}
    initial_X_Zr = ${fuel_zr}
    density = ${fuel_density}
    outputs = all
    block = fuel
  []
  [burnup_elongate]
    type = UPuZrBurnup
    initial_X_Pu = ${fuel_pu}
    initial_X_Zr = ${fuel_zr}
    density = ${fuel_density}
    outputs = all
    block = cladding
    burnup_name = burnup
  []
  [fast_neutron_flux]
    type = FastNeutronFlux
    calculate_fluence = true
    rod_ave_lin_pow = flux_history
    axial_power_profile = fflux_axial_peaking_factors
    block = fuel
    factor = 1.0
    outputs = all
  []
  [fast_neutron_flux_elongate]
    type = FastNeutronFlux
    calculate_fluence = true
    rod_ave_lin_pow = flux_history
    axial_power_profile = fflux_axial_peaking_factors_elongate
    block = cladding
    factor = 1.0
    outputs = all
  []
  [fuel_elasticity_tensor]
    type = UPuZrElasticityTensor
    block = fuel
  []
  [fuel_elastic_stress]
    type = ComputeMultipleInelasticStress
    tangent_operator = nonlinear
    inelastic_models = 'fuel_upuzrcreep'
    block = fuel
  []
  [fuel_upuzrcreep]
    type = UPuZrCreepUpdate
    block = fuel
    porosity = porosity
    max_inelastic_increment = 2e-3
  []
  [fuel_thermal_expansion]
    type = ComputeThermalExpansionEigenstrain
    block = fuel
    thermal_expansion_coeff = 1.18e-5
    stress_free_temperature = 295.0
    eigenstrain_name = fuel_thermal_strain
  []
  [fuel_volumetric_swelling] # Use LIFE-METAL Empirical model
    type = UPuZrVolumetricSwellingEigenstrainLM
    block = fuel
    use_preset_bubble_size = true
    anisotropic_factor = 1.26
    fission_rate = fission_rate
    burnup = burnup
    fis_gas_ret = fis_gas_ret
    hydrostatic_stress = hydrostatic_stress
    eigenstrain_name = fuel_volumetric_strain
    gas_swelling_scale_factor = 1.0
    outputs = all
  []
  [metal_fuel_thermal]
    type = UPuZrThermal
    block = fuel
    spheat_model = savage
    thcond_model = lanl
    porosity = porosity
  []
  [fuel_density]
    type = Density
```

```
    displacements = 'disp_x disp_y'
    block = fuel
  []
  [Fission_Gas_Release]
    type = FgrUPuZrLM
    block = fuel
    fission_rate = fission_rate
    epsilon_c = 0.36
  []
  [clad_elasticity_tensor]
    type = HT9ElasticityTensor
    id_wastage_degradation_function = id_vpp_func
    od_wastage_degradation_function = od_vpp_func
    block = cladding
  []
  [clad_stress]
    type = ComputeMultipleInelasticStress
    tangent_operator = nonlinear
    inelastic_models = 'clad_ht9creep'
    block = cladding
  []
  [clad_ht9creep]
    type = HT9CreepUpdate
    block = cladding
    fast_neutron_flux = fast_neutron_flux
    id_wastage_degradation_function = id_vpp_func
    od_wastage_degradation_function = od_vpp_func
  []
  [thermal_expansion]
    type = ComputeThermalExpansionEigenstrain
    block = cladding
    thermal_expansion_coeff = 1.2e-5
    stress_free_temperature = 295.0
    eigenstrain_name = clad_thermal_eigenstrain
  []
  [clad_thermal]
    type = ThermalHT9
    block = cladding
  []
  [clad_density]
    type = Density
    block = cladding
    density = 7874.0
  []
[]
```

Listing 2. Example of the input file length reduction achieved with the use of the NuclearMaterialsUPuZr and NuclearMaterialHT9 actions

```
[NuclearMaterials]
  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 hoop_stress'
  add_variables = true
  fission_operation = 'Normal'
  physics = 'Thermal Mechanics'
  extra_vector_tags = 'ref'
  stress_free_temperature = 295.0
  initial_temperature = 298
  initial_X_Zr = ${fuel_pu}
  initial_X_Pu = ${fuel_zr}
  [UPuZr]
    [fuel]
      block = fuel
      decomposition_method = TaylorExpansion
      upuzr_models = 'FIPD Elastic Burnup Creep Swelling ThermalExpansion'
      density = 15800
      bubble_number_density = 1e20
      max_inelastic_increment = 2e-3
      rod_linear_power = power_history
      axial_power_profile = axial_peaking_factors
      output_properties = 'porosity gaseous_porosity'
      additional_generate_output = 'volumetric_strain'
    []
  []
  [HT9]
    [clad]
      block = cladding
      decomposition_method = TaylorExpansion
      fast_flux_factor = 2.47e19
      thermal_expansion_coeff = 1.2e-5
      ht9_models = 'FIPD Elastic Creep ThermalExpansion'
      additional_generate_output = 'hoop_creep_strain hoop_elastic_strain hoop_strain'
    []
  []
[]
```

4.2 Standard Metallic Fuel Rod Outputs

The StandardMetallicFuelRodOutputs action simplifies the input file of a metallic fuel simulation by creating up to 11 postprocessors, vector postprocessors, and user objects, which are commonly used to analyze the results of BISON simulations. The standard names given to these postprocessors and vector postprocessors by this action aid in the generation of plots with a common plotting tool. This action is designed for use with only Metallic Fuel simulations run with a mesh generated by either the mesh script or the SmearedPelletMesh; and as such, this action relies on the use of the boundary condition naming conventions in these files.

The user may elect to use the StandardMetallicFuelRodOutputs action to generate output for only the cladding, only the fuel pellets, or both as appropriate to the simulation. All of the possible outputs, generated with the default parameter setting **rod_component = Both**, are shown in Table 3.

Additionally, this action can create the PlenumPressure by including the **initial_pressure**. For simplicity, the gas constant for this situation is assumed to be R = 8.3143 which is in units of $J \cdot K^{-1} \cdot mol^{-1}$. Additional unmeshed volumes and their corresponding temperatures can be included when calculating plenum pressure (e.g., the volume between dish and chamfer surfaces of adjacent pellets) by using the parameters **additional_volumes** and **temperature_of_additional_volumes**. In calculating the volume of the plenum gas, additional volumes can be included via the parameter **internal_volume_addition**. This can apply to situations such as the addition of sodium within the internal volume.

The action accommodates different naming conventions. The amount of fission gas produced can be called **fis_gas_prod** or **fgm_produced**, while the amount of fission gas released can be named either **fis_gas_rel** or **fgm_released**.

4.2.1 Standard Output Blocks

Table 3. Correspondence Among Action Functionality and MooseObjects for the StandardMetallicFuelRodOutputs action.

Material classes created by the StandardMetallicFuelRod action		
Created Classes	**Functionality**	**Generated Postprocessor Name**
SideAverageValue	Average temperature of the cladding interior	average_interior_clad_temperature
AxisymmetricCenterlineAverageValue	Average centerline fuel pellet temperature	average_centerline_fuel_temperature
SideAverageValue	Average plenum temperature	plenum_temperature
InternalVolume	Volume of the plenum gases	plenum_volume
InternalVolume	Volume of the fuel pellets	pellet_volume
NodalMaxValue	Maximum axial displacement of the clad interior	maximum_clad_elongation
NodalMaxValue	Maximum axial displacement of the fuel pellets	maximum_fuel_elongation
NodalValueSampler	Radial displacement of the outer cladding at the simulation end	input file name + _clad_radial_displacement_FINAL
NodalValueSampler	Radial displacement of the outer fuel pellet at the simulation end	input file name + _fuel_outer_radial_displacement_FINAL
ElementExtremeValue	Max Total Hoop Strain	max_total_hoop_strain
LineValueSampler	Cladding Total Hoop Strain	clad_total_hoop_strain
ElementIntegralMaterialProperty	Generated fission gas	fission_gas_generated
ElementIntegralMaterialProperty	Released fission gas	fission_gas_released
FGRPercent	Percentage of released fission gas	fission_gas_released_percentage
ElementAverageValue	Rod average burnup	average_burnup

4.2.2 Example Input File Syntax

The default behavior of the StandardMetallicFuelRodOutputs action will create a standard set of outputs for both the fuel mesh block component of the rod and the cladding. However, it is also possible to use the StandardMetallicFuelRodOutputs action to generate outputs for only the fuel or only the cladding, depending on the **rod_component** selected.

For cladding burst tests that are run only on a hollow tube mesh, the advanced parameters **plenum_boundary_name** and **external_clad_boundary_name** should be set to ensure the correct boundary or boundaries are used in the calculation of the plenum volume, plenum temperature, the average clad interior temperature, and the radial displacement of the cladding exterior.

Note that when **plenum_boundary_name** is set, the boundaries specified by arguments for this parameter are used for both the plenum quantities and the average clad interior temperature.

4.2.3 Example Input Files

As was discussed earlier, the StandardMetallicFuelRodOutputs action allows the flexibility of determining which components (i.e., fuel and/or cladding) will be used in creating the necessary outputs. An example of only fuel components is shown in /refexample:BeforeStandardOutputs and /refexample:PostStandardOutputs. This same process is applicable when only the cladding is selected, or alternatively both components are selected. Regardless, the correct output classes are constructed for the rod components, eliminating possible errors.

Listing 3. Example of the input file length before reduction via `StandardMetallicFuelRodOutputs` action

```
[BCs]
  [PlenumPressure]
    [plenumPressure]
      boundary = 9
      initial_pressure = 0.084e6 # Pa
      startup_time = 0
      R = 8.3143
      temperature = plenum_temperature
      volume = plenum_volume
      output = plenum_pressure
      material_input = fission_gas_released
    []
  []
[]
[Postprocessors]
  [plenum_temperature]
    type = SideAverageValue
    boundary = 9
    variable = temp
    execute_on = 'initial linear'
  []
    [pellet_volume]
    type = InternalVolume
    boundary = 8
  []
  [plenum_volume]
```

```
    type = InternalVolume
    boundary = 9
    execute_on = 'initial timestep_end'
    addition = -1.53703e-6
  []
  [average_burnup]
    type = ElementAverageValue
    block = pellet
    variable = burnup
  []
  [fission_gas_produced]
    type = ElementIntegralMaterialProperty
    mat_prop = fis_gas_prod
    block = pellet
  []
  [fission_gas_released]
    type = ElementIntegralMaterialProperty
    mat_prop = fis_gas_rel
    block = pellet
    execute_on = 'initial timestep_end'
  []
  [fission_gas_released_percentage]
    type = FGRPercent
    fission_gas_released = fission_gas_released
    fission_gas_generated = fission_gas_produced
  []
  [max_total_hoop_strain]
    type = ElementExtremeValue
    value_type = max
    block = clad
    variable = total_hoop_strain
  []
[]
[VectorPostprocessors]
  [clad_radial_displacement]
    type = NodalValueSampler
    variable = disp_x
    boundary = 2
    sort_by = y
    outputs = 'vec2'
  []
[]
```

Listing 4. Example of the input file length reduction achieved with the use of the `StandardMetallicFuelRodOutputs` action

```
[StandardMetallicFuelRodOutputs]
  initial_pressure = 0.084e6 # Pa
  fuel_pellet_blocks = 'pellet'
  plenum_boundary_name = 'inside_surfaces'
[]
```

4.3 FIPD Benchmarking

The FIPD action simplifies the input file for FIPD datasets. From ANL(https://fipd2.ne.anl.gov/),”FIPD is an organized collection of EBR-II test pin data and documentation. The database includes pin operation conditions calculated using a collection of ANL analysis codes developed during the IFR program, including axial distributions for power, temperatures, fluences, burnup, and isotopic densities. The database also contains pin measured data from post-irradiation examination, including pin fission gas release and gas chemistry measurements, and axial distributions from profilometry, gamma scans, and neutron radiography. There is also an extensive collection of documents associated with different pins and experiments, including raw PIE data, design descriptions, safety analysis, and operational reports.”

The FIPD action is used by correlating two sets of information, ”function_purposes” and ”functions_used.” When these two sets are correlated, the action uses the information to pass the FIPD csv files into the correctly constructed Functions, AuxVariables, BCs, PostProcessors, VectorPostprocessors, and Kernels. This greatly reduces the complexity and length of input files, while allowing for experimental verification of BISON simulations. An overview of the constructed MooseObjects is shown in 4.3.1.

The user must select the model used for wastage from the following list: flux_ht9, flux_ss316, flux_d9, burnup_ht9_legacy, burnup_ht9_opt, burnup_ht9_gap, and flux_burnup_ht9 with burnup_ht9_opt being the default. In addition, the cladding material should be slected from the following list: HT9, SS316, HT9_HEDL, and custom with HT9 being the default. These user inputs help to further refine the input parameters for the constructed classes.

4.3.1 FIPD Action Blocks

Table 4. Correspondence Among Action Functionality and MooseObjects for the FIPD action

Moose Objects created by the FIPD action		
Function Purposes	**Functions Type**	**Associated Constructed Classes**
clad_od_temp	FIPDAxialProfileFunction	FunctionDirichletBC
clad_inn_temp	VectorPostprocessorFunction	FunctionDirichletBC NodalValueSampler
power_history	PiecewiseLinear	
pwr_axial_peaking_factors	FIPDAxialProfileFunction	
pwr_axial_peaking_factors_elongate	FIPDAxialProfileFunction	NodalMaxValue
fflux_axial_peaking_factors	FIPDAxialProfileFunction	
fflux_axial_peaking_factors_elongate	FIPDAxialProfileFunction	NodalMaxValue
flux_history	PiecewiseLinear	
flow_rate_history	PiecewiseLinear	
id_vpp_func	MetallicFuelWastageDegradationFunction	FuelRodLineValueSampler MetallicFuelWastage FunctionAux AuxVariable
od_vpp_func	MetallicFuelWastageDegradationFunction	FuelRodLineValueSampler MetallicFuelCoolantWastage FunctionAux AuxVariable
na_vol	MeshPropertyFunction	

4.3.2 Example Input Files

Listing 5. Example of the input file length before reduction via `FIPD` action

```
[Functions]
  [clad_od_temp]
    type =  FIPDAxialProfileFunction
    data_file = clad_od_temp_history_DP11.csv
    use_metadata = true
    mesh_generator = fipd_mesh
  []
  [power_history]
    type = PiecewiseLinear
    data_file = power_history_DP11.csv
  []
```

```
[pwr_axial_peaking_factors]
  type = FIPDAxialProfileFunction
  data_file = peakingfactor_power_relative_DP11.csv
  use_metadata = true
  mesh_generator = gen
  zero_ends = true
  data_shift_type = peaking
[]
[pwr_axial_peaking_factors_elongate]
  type = FIPDAxialProfileFunction
  data_file = peakingfactor_power_relative_DP11.csv
  use_metadata = true
  mesh_generator = gen
  zero_ends = true
  data_shift_type = peaking
  fuel_elongation_pp = max_fuel_elongation
[]
[fflux_axial_peaking_factors]
  type = FIPDAxialProfileFunction
  data_file = peakingfactor_flux_relative_DP11.csv
  use_metadata = true
  mesh_generator = fipd_mesh
  zero_ends = true
  data_shift_type = peaking
  extrapolate_to_zero = true
[]
[fflux_axial_peaking_factors_elongate]
  type = FIPDAxialProfileFunction
  data_file = peakingfactor_flux_relative_DP11.csv
  use_metadata = true
  mesh_generator = fipd_mesh
  zero_ends = true
  data_shift_type = peaking
  extrapolate_to_zero = true
  fuel_elongation_pp = max_fuel_elongation
[]
[flux_history]
  type = PiecewiseLinear
  data_file = flux_history_DP11.csv
[]
[flow_rate_history]
  type = PiecewiseLinear
  data_file = flow_history_DP11.csv
[]
[id_vpp_func]
  type = MetallicFuelWastageDegradationFunction
  vectorpostprocessor_name = id_wastage
  argument_column = y
```

```
    wastage_type = ID
    value_column = wastage_thickness
    use_metadata = true
    degradation_factor = 0.001
    mesh_generator = 'fipd_mesh'
    transition_width = 1E-4
  []
  [od_vpp_func]
    type = MetallicFuelWastageDegradationFunction
    vectorpostprocessor_name = od_wastage
    argument_column = y
    wastage_type = OD
    value_column = cc_wastage_thickness
    use_metadata = true
    degradation_factor = 0.001
    mesh_generator = 'fipd_mesh'
    transition_width = 1E-4
  []
  [ci_temp]
    type = VectorPostprocessorFunction
    argument_column = y
    component = y
    value_column = temp
    vectorpostprocessor_name = clad_inn_temp
  []
  [na_vol]
    type = MeshPropertyFunction
    mesh_generator = fipd_mesh
    mesh_property_name = sodium_volume
    scale_factor = -1.0
  []
[BCs]
  [fuel_top_temp]
    type = FunctionDirichletBC
    boundary = fuel_top
    variable = temp
    function = ci_temp
  []
  [surf]
    type = FunctionDirichletBC
    variable = temp
    boundary = 'cladding_outside_bottom cladding_outside_right cladding_outside_top'
    function = clad_od_temp
  []
[]
[Materials]
  [wastage_thickness]
    type = MetallicFuelWastage
    method = burnup_ht9_opt
    burnup = burnup
    k0_ht9_opt = 39.127250252.
    Qb_ht9_opt = 179068.59493
    scale_factor = 1
    boundary = cladding_inside_right
    outputs = all
  []
  [cc_wastage_thickness]
    type = MetallicFuelCoolantWastage
    clad_material = HT9
    use_effective_method = true
    scale_factor = 1
    boundary = cladding_outside_right
    outputs = all
  []
[]
[VectorPostprocessors]
  [id_wastage]
    type = FuelRodLineValueSampler
    variable = wastage_thickness
    material = 'clad'
    fraction = 0.0
    num_points = 600
    orientation = 'vertical'
    fuel_pin_geometry = 'pin_geometry'
    execute_on = 'initial timestep_end'
    allow_duplicate_execution_on_initial = true
  []
  [od_wastage]
    type = FuelRodLineValueSampler
    variable = cc_wastage_thickness
    material = 'clad'
    fraction = 1.0
    num_points = 600
    orientation = 'vertical'
    fuel_pin_geometry = 'pin_geometry'
    execute_on = 'initial timestep_end'
    allow_duplicate_execution_on_initial = true
  []
[]
```

Listing 6. Example of the input file length reduction achieved with the use of the `FIPD` action

```
[FIPD]
  function_purposes = 'clad_od_temp power_history
    pwr_axial_peaking_factors pwr_axial_peaking_factors_elongate
    fflux_axial_peaking_factors fflux_axial_peaking_factors_elongate
    flux_history flow_rate_history'
  functions_used = 'clad_od_temp_history_DP11.csv power_history_DP11.csv
    peakingfactor_power_relative_DP11.csv peakingfactor_power_relative_DP11.csv
    peakingfactor_flux_relative_DP11.csv peakingfactor_flux_relative_DP11.csv
    flux_history_DP11.csv flow_history_DP11.csv'
  Qb_ht9_opt = 179068.59493
  k0_ht9_opt = 39.127250252
[]
```

5. Assessment Model Development

This section contains a list of questions and issues to consider when beginning the development of a BISON assessment case. It should serve as a step-by-step guide when creating a new assessment.

1. Why is this assessment case necessary?
   (a) What purpose will this model serve?
2. What feature(s)/capability of BISON will this case test?
   (a) Does an existing assessment case already cover this feature/capability? If so, is there value in adding a similar case? Is the extent of the physics involved different enough to warrant another assessment? Are the conditions of the test or experiment varied? Is the consistency of BISON simulations using this feature/capability in question, or will a statistical analysis of the results be made?
   (b) Is the feature/capability being tested fuel-type- and/or cladding- or barrier-material-specific (e.g., TRISO, metallic, and LWR)? Is the feature/capability specific to a particular set of operating conditions?
   (c) Does the feature/capability being tested relate to a specific type of analysis (e.g., steady-state, reactivity-initiated accident, or loss-of-coolant accident)?
3. How will the assessment case be validated?
   (a) What data will be used to determine if BISON is producing correct results?
      i. Have the experimental data been corroborated?
      ii. Are there conflicting experimental results?
      iii. What is the QA status of the data?
      iv. How much uncertainty is there in the data?
      v. Will the BISON results be compared with those of other fuel performance codes?
   (b) How will the BISON output be compared to the data or results from other fuel performance software?
   (c) Can the data be released publicly?
4. Is there an assessment case that can be used as a template for this model?
   (a) If so, what changes need to be made to the template model to simulate the case being developed?
   (b) Are those changes significant enough to warrant a new assessment case, or should this be a subcase of that assessment?

   (c) Can several similar cases be made more consistent by creating a template and setting up subcases with specific geometric, material, and/or operating condition changes?

5. Are all the relevant parameters (e.g., geometry and operating conditions) for creating the simulation available?
   (a) Where are the parameters referenced (e.g., journal papers and reports)?
   (b) If this assessment is fuel-type-specific, do you use the standard set of outputs for that fuel type?
   (c) Are there outputs that, despite lacking any experimental data for comparison, should be output and compared for numerical accuracy and model consistency?
   (d) Are comments and/or documentation in the input file appropriate for making this assessment case more easily understandable to typical users?
   (e) Does this assessment case fit within an existing group definition, or should it be used to create a new group?
6. Has a separate documentation page been created describing the simulation and experimental data and providing a comparison between the two?
   (a) Model geometry (e.g., fuel and cladding)?
   (b) Operating conditions (e.g., power, coolant temperature, and flow rate)?
   (c) Material properties (e.g., structural and thermal)?
   (d) Experimental description and potential limitations?
7. Will a mesh/time-step/tolerance convergence study be conducted to verify the solution?
8. Can a short-running version of the assessment case be created?
   (a) Can the end_time be set to a short value (e.g., at the end of the ramp to power) and output generated for comparison at that time?
   (b) Can a coarser mesh be used to speed up the run?

5.1 Fuel-Specific Figures of Merit

This section outlines the possible outputs for each fuel type, thus affording figures of merit when setting up an assessment case. Some generic outputs that apply to all fuel types will also be listed in this section.

1. Generic Outputs
   (a) Maximum, average, and peak fuel temperature
   (b) Fuel centerline temperature
   (c) Maximum, average, and peak burnup
   (d) Fission gas release percentage
   (e) Total power
   (f) Number of linear and nonlinear iterations per time step
   (g) Total number of linear and nonlinear iterations
   (h) Simulation runtime.
2. LWR Fuel

   - (a) Maximum, average, and peak cladding strains (hoop)
   - (b) Maximum, average, and peak cladding stresses (von Mises, hoop, hydrostatic)
   - (c) Mid-plane contact pressure
   - (d) Pellet volume
   - (e) Plenum temperature and volume
   - (f) Average clad interior temperature
   - (g) Average gap width
   - (h) Maximum fuel elongation
   - (i) Maximum clad elongation
   - (j) Average hydrogen and hydride concentrations
   - (k) Rod average linear power
   - (l) Average axial fuel displacement
   - (m) Average and maximum oxide thickness.

3. Metallic Fuel
   - (a) Maximum, average, and peak fuel strains
   - (b) Cladding axial and radial growth
   - (c) Fuel minimum and maximum axial growth
   - (d) Maximum eutectic penetration in fuel
   - (e) Zr concentration as a function of position and time
   - (f) Fuel cladding chemical interaction behavior
   - (g) Cladding strain and core damage frequency
   - (h) Plenum pressure.
4. TRISO
   - (a) Failure probability (Weibull-based)
   - (b) Release fraction from failed and intact particles
   - (c) Fission product release of specific species (e.g., Ag, Cs, and Sr)
   - (d) Fission product retention of specific species (e.g., Ag, Cs, and Sr)
   - (e) CO production
   - (f) Tangential stress in SiC layer
   - (g) Maximum stresses in layers (e.g., xx, yy, and xy)
   - (h) Strength of layers (IPyC and SiC)
   - (i) Layer volumes and void volume fractions
   - (j) Maximum fluence.
5. Mixed Oxide
   - (a) Average and maximum porosity
   - (b) Minimum and maximum pore velocity
   - (c) Average thermal conductivity
   - (d) Central void formation

(e) Columnar grain region

(f) Equiaxed grain region.

5.2 BISON Review Process

This section lists a series of questions for BISON reviewers of merge requests related to assessment cases being considered. These can apply to both new assessment cases and changes to existing cases.

1. Has the assessment case been appropriately justified? Is it necessary?

2. Does the assessment case feature documentation (e.g., geometry, operating conditions, material properties, and experimental results/measurements)?

3. Are the references for this assessment case available? Have they been checked in terms of consistency with the values used in the model(s)?

4. Is the assessment case based on a previous case? Are there enough differences to justify a separate model?

5. Does the input file follow the fuel type formatting template? Do the names of blocks, variables, materials, and other inputs follow a convention? As an example, fuel relocation strain

```
[Modules/TensorMechanics/Master]
  [./pellets]
    block = 3
    strain = FINITE
    eigenstrain_names = ’fuel_relocation_strain fuel_thermal_strain fuel_volumetric_strain’
    generate_output = ’vonmises_stress hydrostatic_stress stress_xx stress_yy stress_zz strain_xx
                       strain_yy strain_zz’
    extra_vector_tags = ’ref’
  [../]
  [./clad]
    block = 1
    strain = FINITE
    eigenstrain_names = ’clad_thermal_eigenstrain clad_irradiation_strain’
    generate_output = ’vonmises_stress stress_xx stress_yy stress_zz creep_strain_xx creep_strain_yy
                       creep_strain_xy creep_strain_zz strain_xx strain_yy strain_zz’
    extra_vector_tags = ’ref’
  [../]
[]

[Materials]
  .
  .
  .
  [./fuel_relocation]
    type = UO2RelocationEigenstrain
    block = 3
    burnup_function = burnup
    diameter = 0.0095631 #Fuel pellet diameter in m
    linear_heat_rate_function = q
    gap = 190.5e-6 #diametral gap in m
    relocation_activation1 = 5000
    burnup_relocation_stop = 0.029
    eigenstrain_name = fuel_relocation_strain
  [../]
[]
```

6. Are the outputs of BISON defined correctly? Do they depend on other postprocessors or auxvariables? Are those defined correctly as well?

7. Does the assessment case run on Falcon or another high-performance computing system? Does the case pass the nightly testing system? Does the case run using the same PETSc version and other libraries as the nightly system? Does the model finish running in a reasonable amount of time?

8. Are all the necessary files included in the Pull Request?

9. Is there a separate mesh file? If so, can the internal mesh generator be used instead? Does the model use the fuel pin geometry user object? If not, can it be modified to do so?

10. Have plots been generated to compare BISON results with experimental data? Do they work correctly? What plotting system do they use? For example, see Figure 7, which shows a plot generated via the following matplotlib script:

```
#!/usr/bin/env python
#/************************************************/
#/*          DO NOT MODIFY THIS HEADER           */
#/*                                              */
#/*                   BISON                      */
#/*                                              */
#/*    (c) 2015 Battelle Energy Alliance, LLC    */
#/*           ALL RIGHTS RESERVED                */
#/*                                              */
#/*   Prepared by Battelle Energy Alliance, LLC  */
#/*     Under Contract No. DE-AC07-05ID14517     */
#/*     With the U. S. Department of Energy      */
#/*                                              */
#/*     See COPYRIGHT for full restrictions      */
#/************************************************/

import matplotlib as mpl
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.ticker as mtick
import pylab
from matplotlib.ticker import AutoMinorLocator

data_nom = np.genfromtxt(’group_A/gold/x441_group_A_nominal_chkfile.csv’, delimiter=’,’, names=True)
data01 = np.genfromtxt(’group_B/gold/x441_group_B_nominal_chkfile.csv’, delimiter=’,’, names=True)
data02 = np.genfromtxt(’group_C/gold/x441_group_C_nominal_chkfile.csv’, delimiter=’,’, names=True)
data_leg = np.genfromtxt(’group_A/gold/x441_group_A_legacy_swell_chkfile.csv’, delimiter=’,’, names=True)
data01_leg = np.genfromtxt(’group_B/gold/x441_group_B_legacy_swell_chkfile.csv’, delimiter=’,’, names=True)
data02_leg = np.genfromtxt(’group_C/gold/x441_group_C_legacy_swell_chkfile.csv’, delimiter=’,’, names=True)
data_15D = np.genfromtxt(’group_A/gold/x441_group_A_1_5D_chkfile.csv’, delimiter=’,’, names=True)
data01_15D = np.genfromtxt(’group_B/gold/x441_group_B_1_5D_chkfile.csv’, delimiter=’,’, names=True)
data02_15D = np.genfromtxt(’group_C/gold/x441_group_C_1_5D_chkfile.csv’, delimiter=’,’, names=True)
```

```
x_bison = [1.5, 2.1, 1.1]
x = [1.5, 1.5, 1.5, 2.1, 2.1, 1.1, 1.1]
EBR2_data = [0.35, 0.26, 0.30, 0.30, 0.26, 0.30, 0.35]
pin_label = [’DP21’, ’DP50’, ’DP55’, ’DP56’, ’DP37’]
bison = [data_nom[’max_total_hoop_strain’]*100,
         data01[’max_total_hoop_strain’]*100,data02[’max_total_hoop_strain’]*100]
bison_leg = [data_leg[’max_total_hoop_strain’]*100,
             data01_leg[’max_total_hoop_strain’]*100, data02_leg[’max_total_hoop_strain’]*100]
bison_15D = [data_15D[’max_total_hoop_strain’]*100,
             data01_15D[’max_total_hoop_strain’]*100, data02_15D[’max_total_hoop_strain’]*100]

fig = plt.figure()
ax1 = fig.add_subplot(111)
mpl.rcParams.update({’font.size’: 10})
ax1.set_xlabel("Plenum to Fuel Volume Ratio")
ax1.set_ylabel("Maximum Cladding Strain (%)")
ax1.set_xlim(1.0,2.2)
ax1.minorticks_on()
plt.xticks(np.arange(1.0,2.2,step=0.1))
ax1.set_ylim(0.0,0.7)
ax1.scatter(x, EBR2_data, label=’EBR-II’, marker=’x’, color=’blue’)
ax1.scatter(x_bison, bison, label=’Bison (New Swell)’, marker=’o’, color=’red’)
ax1.scatter(x_bison, bison_leg, label=’Bison (Legacy Swell)’, marker=’o’, color=’black’)
ax1.scatter(x_bison, bison_15D, label=’Bison (1.5D)’, marker=’o’, color=’green’)
ax1.grid(True,linestyle=’--’)
leg = ax1.legend(loc=’upper right’)
plt.savefig(’plot_clad_strain_vs_plen_fuel_ratio.pdf’, format=’pdf’, bbox_inches=’tight’, pad_inches=0.5)
```

11. Are the experimental data located in the correct place in the BISON repository? Is it releasable to the public?

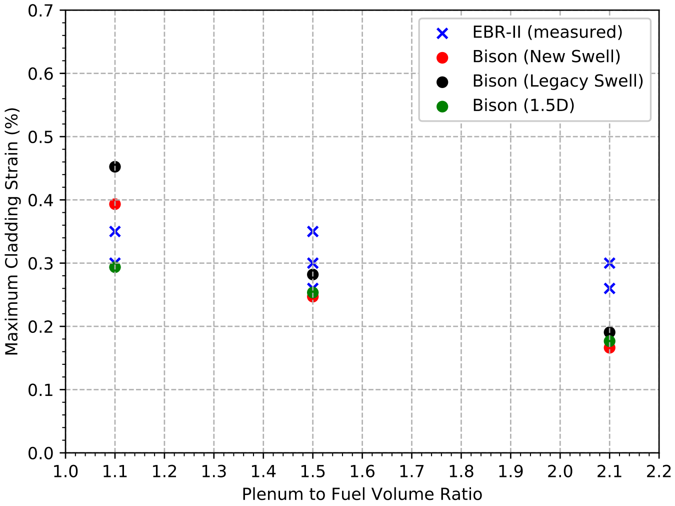

Figure 7. Maximum cladding strain versus plenum-to-fuel-volume ratio for a variety of BISON models and measurements from Experimental Breeder Reactor-II.

6. Summary and Future Work

The objective of this milestone is to demonstrate how emerging fuel performance models should be assessed and documented. In this milestone report we have shown results for separate effects tests and corresponding simulations of creep behavior in metallic fuel and cladding, metallic fuel assessment standardization and FIPD integration, and general guidance for developing new assessment cases. As the models and assessment cases for UN/UC, TRISO, doped-fuel, and other fuel-types emerge, the approach documented here should be referenced and expanded.

Bibliography

[1] MFH. *Metallic Fuels Handbook*. Argonne National Laboratory. Compiled by: IFR Property Evaluation Working Group. June 1988.

[2] FG Foote, HH Chiswik, and RE Macherey. *Annual progress report for 1965 metallurgy division*. Tech. rep. ANL-7155. Argonne National Laboratory, 1965.

[3] HA Saller et al. *Creep strengths of uranium alloys at 1500 and 1800F*. Tech. rep. BMI-834. Battelle Memorial Inst., Columbus, Ohio, 1953.

[4] DE Janney and SL Hayes. "Experimentally known properties of U-10Zr alloys: A critical review". In: *Journal of Nuclear Technology* 203.2 (Aug. 2018), pp. 109–128. DOI: `https://doi.org/10.1080/00295450.2018.1435137`.

[5] SM McDeavitt and AA Solomon. "Hot-isostatic pressing of U-10Zr by a coupled grain boundary diffusion and creep cavitation mechanism". In: *Journal of Nuclear Materials* 228.2 (Oct. 1996), pp. 184–200. DOI: `https://doi.org/10.1016/S0022-3115(95)00220-0`.

[6] Ho Jin Ryu, Yeon Soo Kim, and A M Yacout. "Thermal creep modeling of HT9 steel for fast reactor applications". In: *Journal of Nuclear Materials* 409.3 (Feb. 2011), pp. 207–213.

[7] JL Straalsund and DS Gelles. *Assessment of the performance potential of the martensitic alloy HT-9 for liquid-metal fast-breeder-reactor applications.* Tech. rep. HEDL-SA–2771-FP DE83 015477. Hanford Engineering Development Laboratory, 1983.

A. Best Practices for Finite Element Analysis Model Development

- Before beginning work
  1. What is the aim of the simulation? What needs to be calculated? What degree of accuracy is required?
  2. How can the results of this model be checked? Are any changes in the boundary conditions or dimensions of the model expected to alter the results in a predictable way? Can a simple hand calculation be performed as a check to estimate a simulation result?
  3. How will the model be developed? Is there a subset of the domain and/or physics involved that can be simulated first?
  4. Must the model be made parametrically? Will the dimensions potentially change? Does the meshing of the domain need to be biased into a particular region?
  5. Do you anticipate the necessity for a parameter sensitivity study? Will an optimization of some aspect of the simulation be necessary?
  6. Do you know the material properties for the model? How accurately? Is there any uncertainty in the values? Are the units for properties consistent?
  7. Does a similar model or simulation exist? Can an existing model be used as a starting point and modified in steps to perform the simulation?
  8. Does any geometric symmetry exist for making the model smaller and simpler?

- Setting up the simulation
  1. How is the geometry specified? Do you have a CAD geometry or a drawing? What are the units?
  2. Will you need to create the geometry in Cubit? What boundary conditions will be needed? What sidesets/nodesets will be needed? Are multiple blocks needed? What element order should be used?
  3. Is a mesh convergence study appropriate? Can the problem be accurately solved using a large number of lower-order elements, or fewer higher-order elements? Is the simulation run time an issue?
  4. Can standard solver options be used, or does the problem require more sophisticated solver capa-

bilities and/or time-stepping strategies? Preconditioners? Predictors?

5. Are dampers needed for this simulation? What type? Maximum increment or bounding values to be specified?
6. Creating an input file
   (a) Know your goals
      i. What question are you trying to answer?
      ii. What quantities need to be calculated and output?
   (b) Start simple
      i. For mechanics problems, begin with elastic material
      ii. For contact, use frictionless behavior
      iii. Use a simple ramp-and-hold function for boundary conditions, instead of a more complex, realistic functional form
      iv. Linear elements
   (c) Check the default values of unspecified parameters
   (d) Look at regression tests, examples, and assessment cases.

- Running the simulation

  1. Must the simulation be run serially, or is parallel processing an option?
  2. Does the size and/or complexity of the model require a certain number of computational resources (e.g., nodes and processors on a cluster)?

- Postprocessing the results

  1. Are all relevant outputs saved in a format that can be plotted or tabulated?
  2. Are scripts necessary to automate the postprocessing of results?
  3. Have the plotting tools and formats been decided?