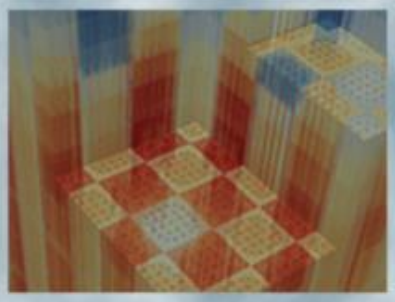
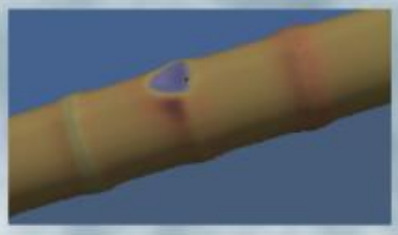
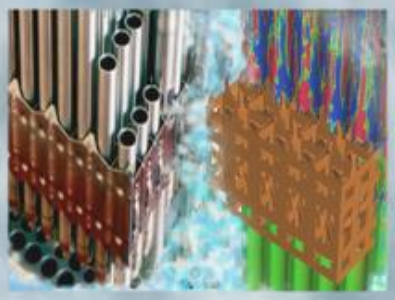
SAND2022-14794R

CASL
A DOE Energy Innovation Hub

CASL-I-2015-0321-000
ORNL/SPR-2015/609







Executing BISON-CASL Fuel Performance Cases Using VERA-CS Output

Shane Stimpson, ORNL
Kevin Clarno, ORNL
Jeff Powers, ORNL
Roger Pawlowski, SNL

September 30, 2015

U.S. DEPARTMENT OF ENERGY | Nuclear Energy

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

REVISION LOG

Revision	Date	Affected Pages	Revision Description
0	09/30/2015	All	Original Report

Document pages that are:

Export Controlled NO

IP/Proprietary/NDA Controlled NO

Sensitive Controlled NO

Approved for Public Release YES

Requested Distribution:

To:

Copy:

EXECUTIVE SUMMARY

As the Consortium for Advanced Simulation of Light Water Reactors (CASL) moves forward with more complex multiphysics simulations, there is increased focus on incorporating fuel performance analysis methods. The coupled neutronics/thermal-hydraulics capabilities within the Virtual Environment for Reactor Applications Core Simulator (VERA-CS) have become relatively stable and major advances have been made in analysis efforts, including the simulations of twelve cycles of Watts Bar Nuclear Unit 1 (WBN1) operation [1]. However, VERA-CS approaches for treating fuel pin heat transfer have well-known limitations that could be eliminated through better integration with the BISON-CASL fuel performance code. Several approaches are being taken to improve or replace the VERA-CS fuel models and improve integration with BISON-CASL. Tiamat [2,3,4] is being developed to replace the fuel models with a direct coupling of BISON-CASL and the neutronics and coolant heat transfer of VERA-CS. Tiamat will execute in an inline mode that allows for in-memory transfer of VERA-CS data to BISON-CASL. This will provide fuel performance results during core simulation. However, fuel performance typically undergoes an independent analysis using a stand-alone fuel performance code with manually specified input defined from an independent core simulator solution or set of assumptions.

This milestone covers initial efforts to facilitate using VERA for core simulation and fuel performance to operate in this mode of execution. This approach will be used for several CASL fuel performance challenges, such as RIAs and DNB. These issues require time-dependent full core results to establish the boundary conditions for the high-fidelity BISON-CASL simulations. The approach will also be used to improve fuel temperature models in VERA-CS, guide the development of Tiamat, and simplify the use of BISON-CASL within VERA for industry users.

Prior to this work, an xml2moose preprocessor was used to generate partially complete BISON-CASL inputs for each unique pin-type for use in Tiamat. Useful extensions have been made to the xml2moose preprocessor to incorporate source and boundary conditions (pin power and moderator temperature) data from the VERA-CS HDF5 output files (VERAout) and create corresponding input files for each fuel rod for BISON-CASL to execute. More work was performed to streamline the execution of many BISON-CASL pins and incorporate the output into VERAout.

While this tool is still being developed and extended, many of the capabilities needed to simulate full/quarter core multicycle depletions have been developed, such as the ability to link outputs from multiple cycles, and the ability to implement assembly shuffling in full symmetry. To demonstrate progress to date, several sets of results are presented:

1. A simple single pin is analyzed to highlight the output data processed into BISON-CASL inputs. A brief scaling study is included to validate the planned approach to execute larger cases with each pin run in serial.
2. A 3 × 3 assembly cross multicycle depletion problem is presented which is similar to cases shown in previous Tiamat publications [2,3,4]. Results for this problem are provided using quarter symmetry without fuel shuffling between cycles.
3. A 3 × 3 test problem with 3 × 3 pin assemblies is presented to demonstrate the initial shuffling capabilities.

CONTENTS

FIGURES

TABLES

ACRONYMS

ASCII	American Standard Code for Information Interchange
BOC	beginning of cycle
CASL	Consortium for Advanced Simulation of Light Water Reactors
CMFD	coarse mesh finite difference
COBRA	Coolant Boiling in Rod Arrays
CS	core simulator
CSV	comma separated value
CTF	COBRA two-phase flow
DNB	departure from nucleate boiling
DOE	US Department of Energy
EFPD	effective full power day
EOC	end of cycle
EPRI	Electric Power Research Institute
HDF	hierarchical data format
IFBA	integral fuel burnable absorber
INL	Idaho National Laboratory
JFNK	Jacobian Free Newton-Krylov
LWR	light water reactor
MOOSE	Multiphysics Object Oriented Simulation Environment
MPI	message passing interface
MW	megawatt
NEM	nodal expansion method
ORNL	Oak Ridge National Laboratory
PCI	pellet-clad interaction
PHI	physics integration
PWR	pressurized water reactor
RIA	reactivity-insertion accident
SP_3	simplified P_3
TH	thermal-hydraulics
VERA	Virtual Environment for Reactor Applications
WBN1	Watts Bar Nuclear Unit 1
XML	extended markup language

1. INTRODUCTION

1.1 Motivation

As the Consortium for Advanced Simulation of Light Water Reactors (CASL) moves forward with more complex multiphysics simulations, there is increased focus on incorporating fuel performance analysis methods into the Virtual Environment for Reactor Applications Core Simulator (VERA-CS). The coupled neutronics/thermal-hydraulics (TH) capabilities within VERA-CS have become relatively stable, and major advances have been made in analysis efforts, including the simulations of twelve cycles of Watts Bar Nuclear Unit 1 (WBN1) operation [1]. Several different approaches are being taken to consider fuel performance, including a more direct, multiway coupling with Tiamat [2,3,4], as well as a more loosely coupled, one-way approach with standalone BISON-CASL cases.

This milestone focuses on generating and executing BISON-CASL fuel performance cases using output power and temperature distributions generated by coupled neutronics/TH simulations with VERA-CS. While ongoing work is underway to develop Tiamat, the primary motivation for this approach is to provide quicker results without the need to rerun previous calculations. For example, output from the twelve cycles of WBN1 simulated with MPACT and CTF within VERA-CS [5] could be used as a basis for independent BISON-CASL cases to assess parameters of pellet-clad interaction or accuracy of global fuel temperature distributions.

Previous work to support WBN1 depletion used standalone BISON-CASL to generate a fuel temperature table. This table was used by VERA-CS with representative, core-averaged power/temperature distributions, and it was used for all pins in the core. Once multiple fuel temperature tables can be entered, the ability to generate cases for each pin will help provide more accurate fuel temperature tables, and different fuels will be able to use separate tables. Alternatively, it values for thermal properties, such as fuel and gap conductivity, could be generated which could be used by CTF.

1.2 VERA-CS Description

The VERA simulation environment being developed by CASL is comprised of codes that are collectively used for nuclear reactor modelling and simulation. This work uses the MPACT neutron transport solver, the CTF TH solver, and the BISON-CASL fuel performance code, each of which is briefly described below. Figure 1.2.1 shows the components of VERA. At present, the components that make up the core simulator (VERA-CS) are the neutronics, TH, and fuel performance packages.

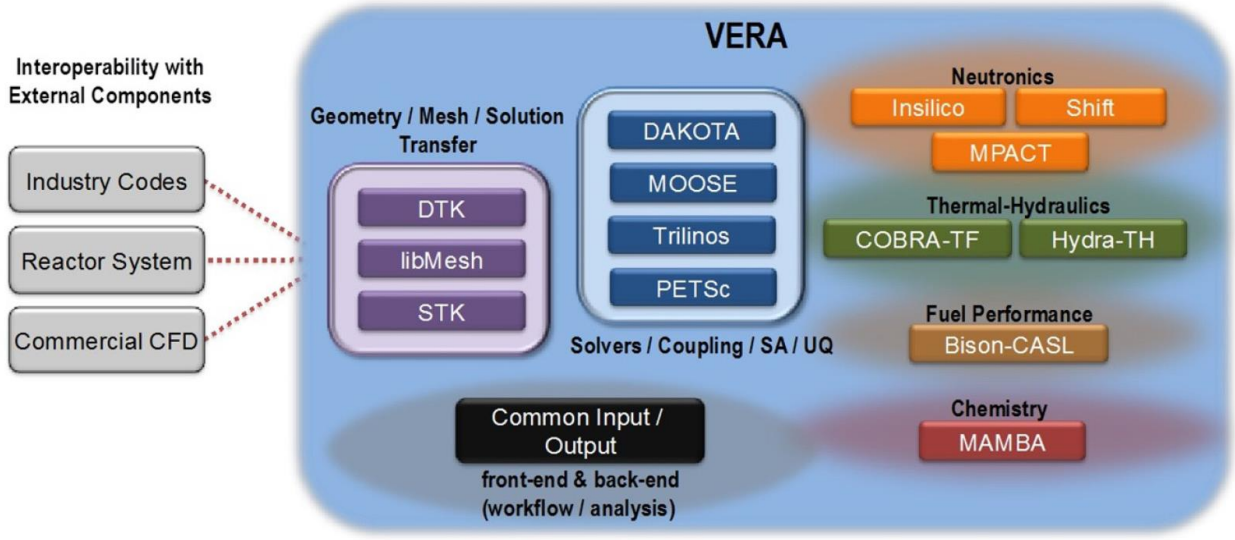


Figure 1.2.1. VERA Components.

1.2.1 MPACT

The MPACT neutron transport solver, being developed collaboratively by Oak Ridge National Laboratory (ORNL) and the University of Michigan (UM), provides pin-resolved flux and power distributions [6]. To solve three-dimensional (3D) problems, it employs the 2D/1D method, which decomposes the problem into a 1D-axial stack of 2D-radial planes [7]. Typically, 2D-MOC is used to solve each radial plane, and 1D-nodal methods are used to solve axially along each rod. While there are a variety of axial solvers available, the nodal expansion method (NEM)-simplified P_3 (SP_3) solver is the default, which wraps a one-node NEM kernel [8]. These 2D and 1D solvers are coupled together through transverse leakage terms to ensure neutron conservation, and they are accelerated using 3D coarse mesh finite difference (CMFD).

Some of the initial results were obtained using a simplified TH model instead of coupling to CTF. The simplified TH model enforces mass, momentum, and energy conservation on assembly-size nodes compared to the subchannels used in CTF. It also does not account for any crossflow between assemblies. This capability is being extended to perform calculation on a more resolved basis. Simplified TH was primarily used with consideration of wall time limitations on Titan [9]. However, CTF results are also presented, but with cross flow disabled to ensure completion within wall time.

1.2.2 CTF

CTF is a subchannel TH code being developed by ORNL and Pennsylvania State University specifically for light water reactor (LWR) analysis [10]. It simulates two-phase flow with a three-field representation—liquid, droplet, and vapor—assuming that the liquid and droplet fields are in dynamic equilibrium, leaving two energy conservation equations.

CTF provides significantly higher resolution than simplified TH, but it is more currently more limited in parallelization. Ongoing work is focused on increasing parallelization from the assembly level to the subchannel level, significantly alleviating runtime concerns for cases presented in this work.

1.2.3 BISON-CASL

The BISON-CASL fuel performance code is being developed by Idaho National Laboratory (INL) to provide single-rod fuel performance modeling capability so that users can assess safety margins and the impact of plant operation and fuel rod design on thermo-mechanical behavior such as pellet-cladding interaction (PCI) failures in pressurized water reactors (PWRs) [11,12]. PCI is controlled by the complex relationship between the mechanical, thermal, and chemical behaviors of a fuel rod during operation. Consequently, modeling PCI requires an integral fuel performance code to simulate the fundamental processes of these behaviors. BISON-CASL is built on INL's Multiphysics Object Oriented Simulation Environment (MOOSE) / BISON packages [13,14], which use the finite element method for geometric representation and a Jacobian Free Newton-Krylov (JFNK) scheme to solve systems of partial differential equations [13]. For this work, BISON-CASL uses a 2D azimuthally-symmetric (R-Z), smeared-pellet thermomechanical fuel pin model with output data from VERA-CS to generate the time-dependent power shape/history and moderator temperature inputs needed for BISON-CASL.

1.3 Milestone Objectives

There are several primary objectives for this milestone:

1. create a unified BISON-CASL template for standard UO_2 fuel;
2. modify the react2xml preprocessor capability to account for new parameters in the BISON block;
3. modify the xml2moose preprocessor to create an input for each pin, updating the values in the template based on the parameters in the extended markup language (XML) file; and
4. enable xml2moose to read the VERA-CS output hierarchical data format HDF5 file and populate input files that BISON-CASL can process.

The following two items are stretch goals needed in future extensions of the capability, but are only noted for this milestone:

1. Consider multicycle BISON-CASL cases without shuffling and
2. Extend the multicycle capability to include fuel shuffling.

1.4 Acknowledgements

The authors wish to acknowledge Nathan Capps (University of Tennessee) and Danielle Perez (INL) for assistance with the BISON-CASL input file specification.

The authors also wish to acknowledge that this work was supported by the Consortium for Advanced Simulation of Light Water Reactors (www.casl.gov), an energy innovation hub (http://www.energy.gov/hubs) for modeling and simulation of nuclear reactors under U.S. Department of Energy (DOE) Contract No. DE-AC05-00OR22725.

This work also made use of resources of the Oak Ridge Leadership Computing Facility at ORNL, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725

2. CREATION OF A CONSOLIDATED TEMPLATE

Within CASL, BISON-CASL has been used for several different applications and problems. Standalone BISON-CASL has been made available to industry and used in a recent Electric Power Research Institute (EPRI) test stand, the standalone version was recently used to generate fuel temperature tables for simulations of the WBN1 reactor [1], it is being used within Tiamat for coupled MPACT/CTF/BISON-CASL simulations [2,3,4], and it was being used by FMC for modeling 3D missing pellet surface problems in the Braidwood Nuclear Power Plant. In FY16, it will be used in an inline version of Tiamat for completing the Level 1 PCI milestone in FY16. Unfortunately, the inputs/templates used in these efforts were very different, so the first task of this milestone was to consolidate these inputs into a common template, which is shown in Appendix A.

There are several different blocks that are enabled when the template is being processed for Tiamat compared to standalone BISON-CASL cases. Tiamat has several different post-processors that are only used for in-memory transfers, including one for the conservation of power, which is essential to ensuring stability and convergence. Anything specific to Tiamat use is purged from the input for standalone cases. Additionally, there are a couple different boundary conditions that are supported: fixed axial distribution of the clad temperature, fixed axial distribution of the coolant temperature, and simplified, axial coolant flow. Tiamat uses a Dirichlet boundary condition to specify the clad surface temperature, which is determined by CTF, whereas the standalone cases currently use a Dirichlet boundary condition to specify the bulk coolant temperature, since this data is stored in the output HDF5 files. It would be possible to use the clad surface temperature if it were stored, but only the average clad temperature is currently available. The last boundary condition, which is currently not supported through the template, is based on the `CoolantFlow` block of the BISON-CASL input, which has been used by EPRI in the test stand evaluation cases.

3. PREPROCESSOR MODICATIONS

Several preprocessors convert VERA ASCII input into a form that each code reads. The first preprocessor is react2xml, which converts VERA input into an XML file. Some codes such as MPACT have been modified to fully process the XML file directly. However, codes such as CTF and BISON-CASL have additional preprocessors that convert the XML file into a format native to each code.

3.1 VERA Input to XML (react2xml)

In the work to consolidate the inputs into a common template, input parameters were identified as necessary to be modifiable through the BISON block of the VERA input. These parameters relate to options for various purposes (e.g., meshing, physics coefficients, and solver selection). In total there are 125 modifiable input parameters denoted by "#VERA_MODIFIABLE" in the template (Appendix A). Each parameter can be entered to the VERA ASCII input and added to the XML file during conversion with react2xml.

3.2 XML to BISON-CASL (xml2moose)

With the various input parameters now in the XML file, xml2moose will populate the template accordingly, replacing the default values with the new values as appropriate. Furthermore, xml2moose will modify the template based on other nonoptional, inputs such as geometry and temperature.

3.2.1 Creating Inputs for Each Fuel Rod

To accommodate Tiamat's needs, xml2moose was initially developed so that a new BISON-CASL input is created for each unique pin type. Only a few inputs were actually generated for any particular problem because Tiamat handles much of the data passing and other rod-specific characteristics internally, so it does not need an input for each pin. However, the rod in this work needs a separate input since the power/temperature profiles and histories are input externally. Extending xml2moose to generate an input for each rod only required minor modifications to the internal identification of unique rods and file nomenclature.

3.2.2 Pulling Data from Output Files

Pulling the solution data from the output files has been the most substantial development of this work. Essentially, the pin power and moderator temperature distributions are pulled from the output HDF5 files produced by VERA-CS and stored in comma separated value (CSV) files to be read by BISON-CASL. Three files were produced: (1) the normalized axial power distribution, (2) axial distribution of the moderator temperature, and (3) the power history of the rod. Because axial power input must be normalized, the normalization coefficient is factored into the power history. More specific examples are shown in the next section.

Unlike VERA-CS, BISON-CASL cannot immediately begin at-power operation: it needs sufficient time to ramp to power, or nonconvergence will be encountered. Therefore, an 8-hour time period is included prior to start-up to adjust from cold zero-power to hot zero-power, and another 24-hour period power ramp is imposed to go from zero power to initial power. Even if the VERA-CS input/output used a ramp to power, the BISON-CASL inputs would still impose an additional power ramp.

3.2.3 Multicycle Capability

To account for multicycle depletion, the preprocessor has been modified to take in a series of output files, which it will read together, appending the output data together. In each VERA input, the cycle operation start and end dates are specified (`op_date`), which are also placed into the output file. xml2moose reads these dates and places an appropriate outage time between cycles. Similarly, 24-hour down- and up-ramps are introduced at the beginning and end of each outage. Currently, this 2-day period is being deducted from the total outage time, but that decision may be revisited in the future. Initial development focused on multicycle capability without shuffling, as shown in the demonstration results. However, work has begun on incorporating the shuffling capability, and results are included for full symmetry shuffling.

3.2.4 Shuffling Capability

In each cycle of an operating reactor (excluding the first), some of the fuel is replaced with fresh fuel and the bundles are shuffled to provide better fuel utilization. When fresh fuel is inserted, new BISON-CASL inputs are generated, and the fuel is tracked through the life of the assembly. To consider an example of this, cycle 1 assemblies in a 3 × 3 assembly core may be labeled as follows:

```
A-1 B-1 C-1
A-2 B-2 C-2
A-3 B-3 C-3
```

When cycle 1 simulation is complete and the assemblies are being shuffled for cycle 2, the following example `shuffle_label` would shift row 1 to row 3, row 2 to row 1, and fresh fuel is inserted along row 2 (denoted by "+"):

```
restart_shuffle ../c1_depl/cycle1.res EOC

shuffle_label  A-2 B-2 C-2
                +   +   +
               A-1 B-1 C-1
```

In many cases, the fuel rod is used in the next two cycles (with shuffling), and the data from the new location are used. However, in some cases, the assembly may be placed into the spent fuel pool for several cycles before being reinserted. For example, in cycle 3, one might place an assembly from row 3 of cycle 1 into the center:

```
restart_shuffle ../c2_depl/cycle2.res EOC
                ../c1_depl/cycle1.res EOC

shuffle_label  A-1 B-1 C-1
               +  1A-3   +
               A-2 B-2 C-2
```

It is possible for any particular cycle to include fuel from several previous cycles or from different reactors. Currently, the accepted procedure for core shuffling specifies that shuffling is performed in a separate case from cycle depletions, so the `shuffle_label` data are not available in the output HDF5 files. For this reason, the XML files for the shuffle input must be specified. Furthermore, there may be geometric differences in the fresh assemblies in other cycles, so the XML files for those also must be specified.

3.2.4 Parallelization

To help decrease the time spent processing inputs and creating CSV files which could yield thousands of files generated, message passing interface (MPI) parallelization has been incorporated. There are two levels of loops in xml2moose that require parallelization: (1) the loop over unique assemblies, which then has (2) a nested loop over unique pins. Based on the number of MPI processes available, the bounds of these loops are determined by a simple partitioning scheme, and they support an arbitrary number of processors.

3.3 Unit Testing

To ensure that these capabilities are retained through future development, two unit tests were added to the MOOSEExt package. First is the `bison_from_vera` test, a single fuel rod problem that uses an HDF5 output and runs BISON-CASL for a limited timeframe. The results of this case, which are also reported in a CSV file, are compared to a reference results file with a specified tolerance. Second, a 3 × 3 cross with 3 × 3 pin assemblies is tested for multicycle capability in `bison_from_vera_multi`.

4. CAPABILITY DEMONSTRATION

To consider these capabilities in action, a few test problems will be considered.

The first is a single fuel rod depleted to 440 effective full power days (EFPDs) at full power. This particular pin cell came from a 3 × 3 cross case (the next problem considered). Evaluating the single pin cell case allows for a better understanding of inputs for each case, and a small scaling study has been performed.

Next, a larger 3 × 3 cross problem with 17 × 17 assemblies is evaluated. This problem also includes a baffle and reflector, which would be used in a quarter core model. All VERA-CS results in this work used the 8-group cross section library for fast turnaround time. The important part of these figures is not necessarily the accuracy of the values, particularly since the 3 × 3 cross is a fairly atypical problem, but rather that they demonstrate the mechanics of the process to produce them. In the figures below, any maximum or minimum parameters reported are axial. Average quantities are nodal-averaged.

4.1 Single Pin Depletion

4.1.1 Input Data

As mentioned above, there are three primary inputs formulated from the output of VERA-CS cases:

1. axial power distribution (Figure 4.1.1),
2. moderator temperature distribution (Figure 4.1.2), and
3. rod linear power history (Figure 4.1.3).

The following figures demonstrate the time dependence (in EFPD) of the inputs.

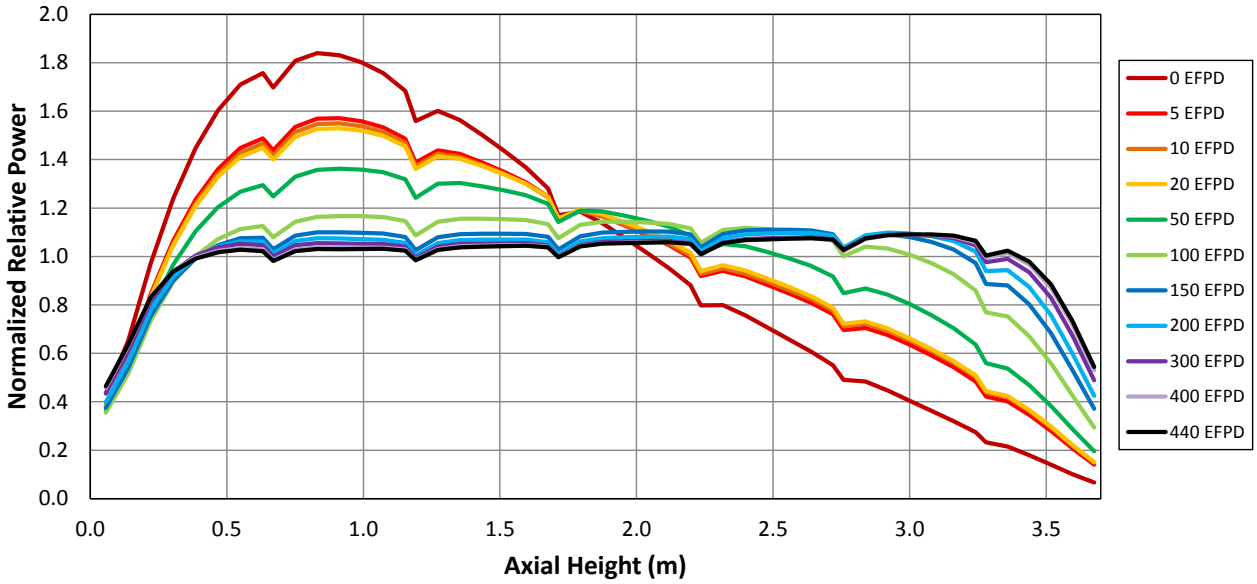


Figure 4.1.1. Single Pin – Normalized Axial Power Distribution.

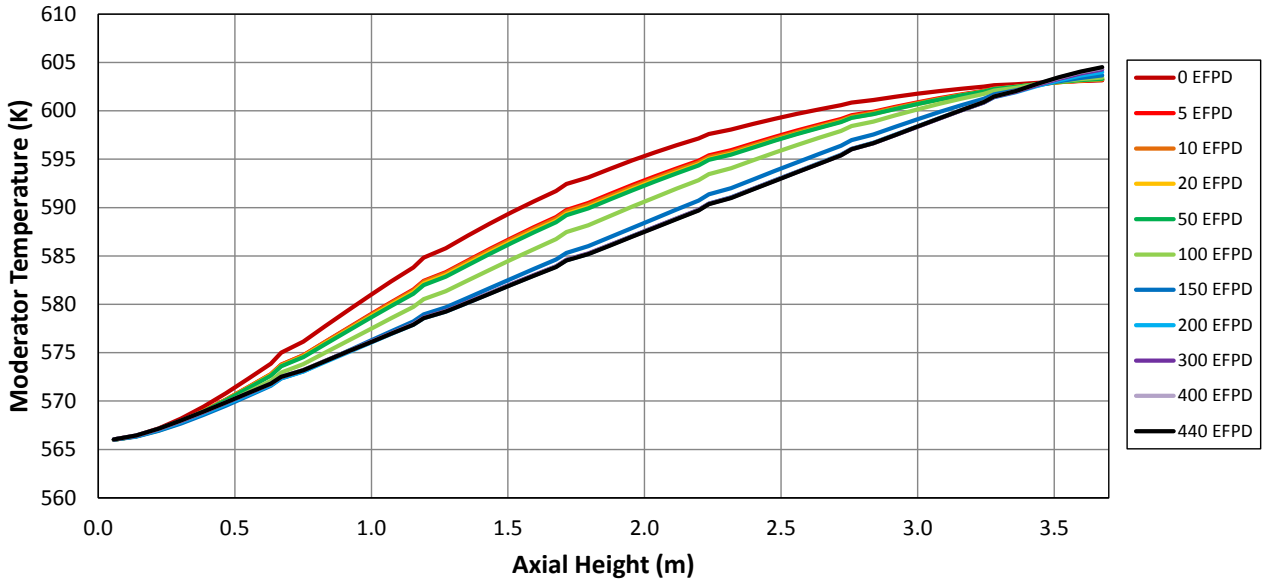


Figure 4.1.2. Single Pin – Moderator Temperature Distribution (K).

The relative power for the pin depicted in Figure 4.1.3 increased over time. At beginning of cycle (BOC), the relative power is nearly unity, but as the higher power pins burn out more, the power in this pin slightly increases.

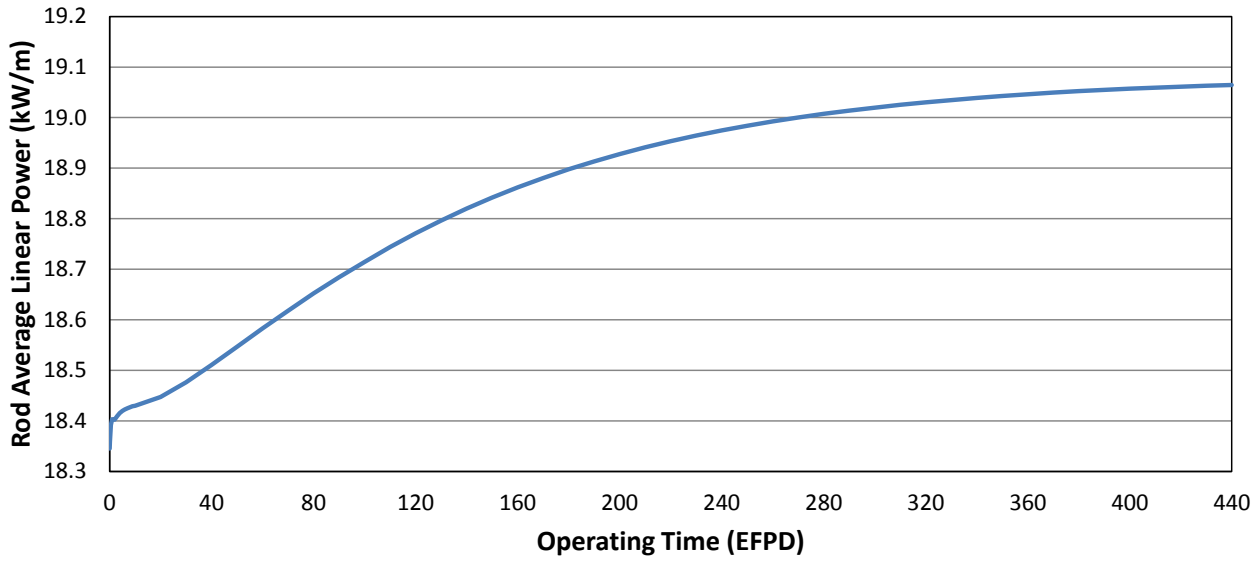


Figure 4.1.3. Single Pin – Rod Average Linear Power (kW/m).

4.1.3 Output Data and Scaling Study

Because of the number of cases that will be generated, the plan is to have at least one pin per process. However, the parallel performance of each BISON-CASL case should be understood in cases with a very large number of processors. Table 4.1.1 shows the timing and efficiency results for the single pin cases that include 1–64 processors as run on ifba.ornl.gov. As can be seen, efficiency drops very quickly, further justifying the approach of limiting one pin per process. This is not terribly surprising since any efficiency loss would support this idea.

Table 4.1.1. Single Pin – Scaling Study Timing/Efficiency Results

Proc.	Time (min)	Eff. (%)
1	59.2	100.0
2	39.9	74.1
4	24.0	61.8
8	14.1	52.5
16	10.7	34.5
32	8.7	21.4
64	10.1	9.2

Figures 4.1.4 (maximum centerline fuel temperature), Figure 4.1.5 (minimum clad hoop stress), Figure 4.1.6 (average fuel temperature), and Figure 4.1.7 (minimum gap distance) generally show very consistent results except for the minimum gap distance, which demonstrates severe discrepancies. While using 1-4 processors yields consistent results, anything more implies that contact is made between fuel and cladding. These discrepancies need to be understood, but they do not impede future cases since parallel execution of each case is not planned. This discrepancy may be an error in the internal BISON-CASL post-processor routine since other results are not affected. If the code were seeing contact, temperatures would likely be substantially different than without contact. Negative values for gap distance are possible, which would indicate that the fuel and clad mesh are overlapping.

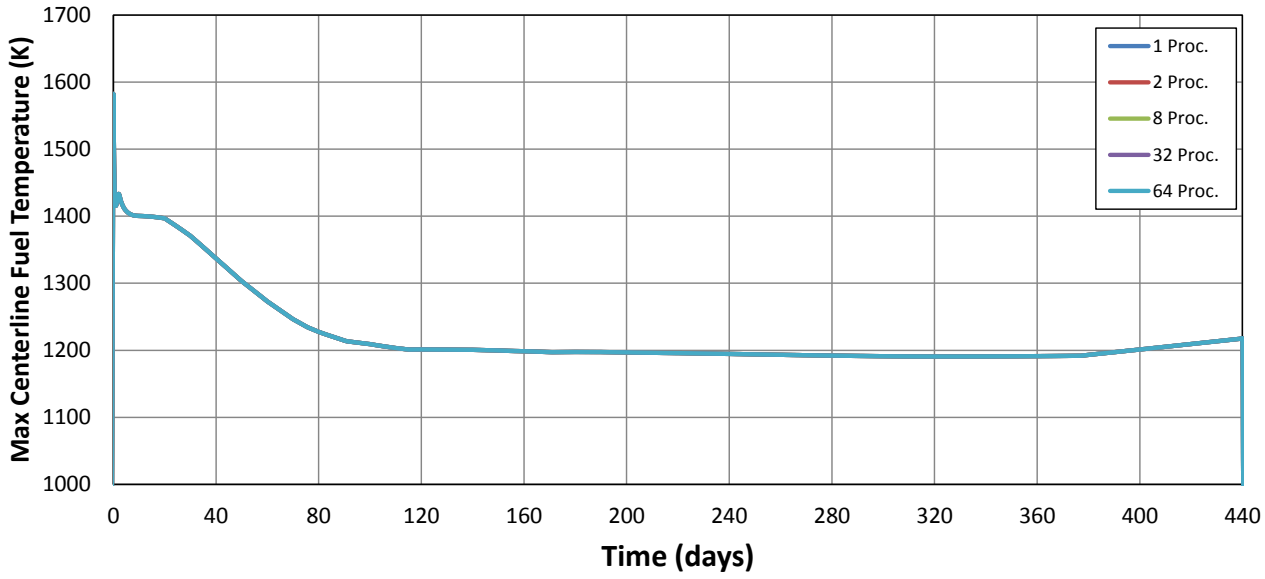


Figure 4.1.4. Single Pin - Maximum Centerline Fuel Temperature (K).

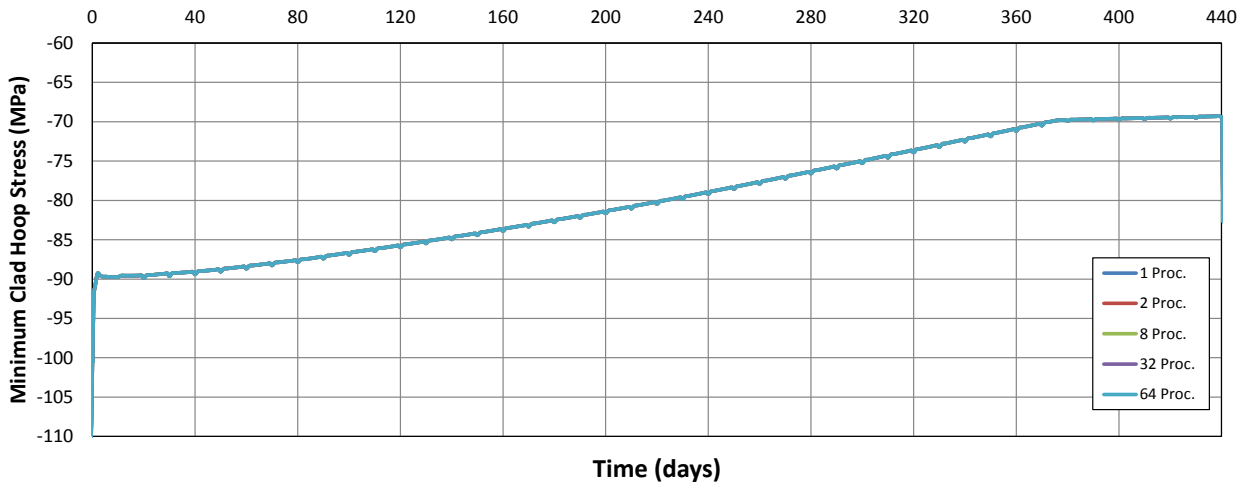


Figure 4.1.5. Single Pin – Minimum Clad Hoop Stress (MPa).

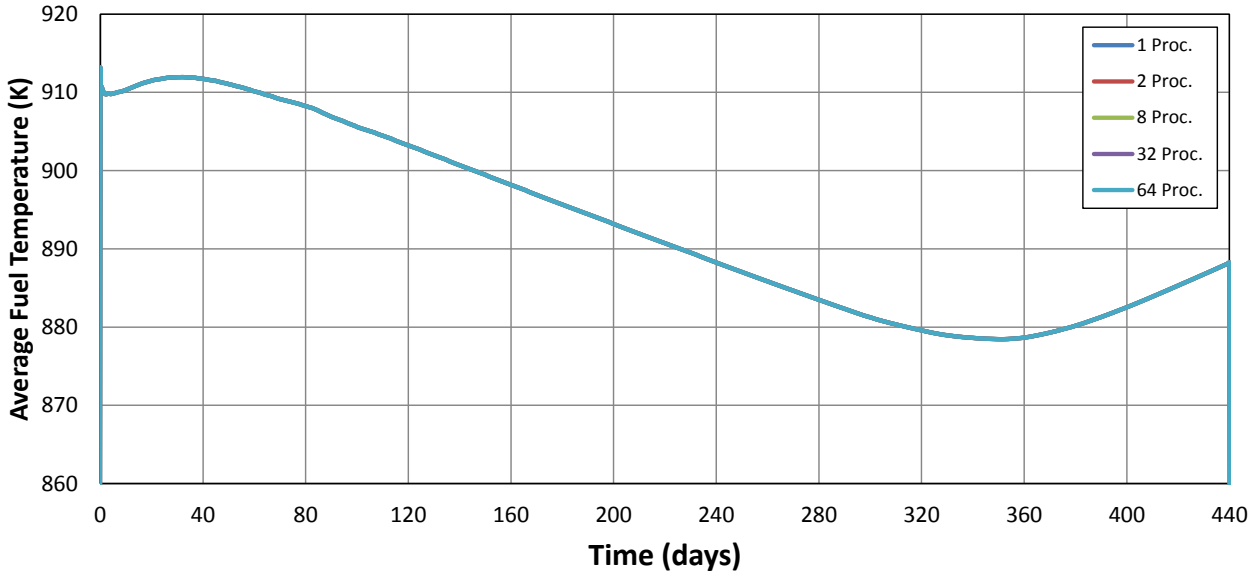


Figure 4.1.6. Single Pin - Average Fuel Temperature (K).

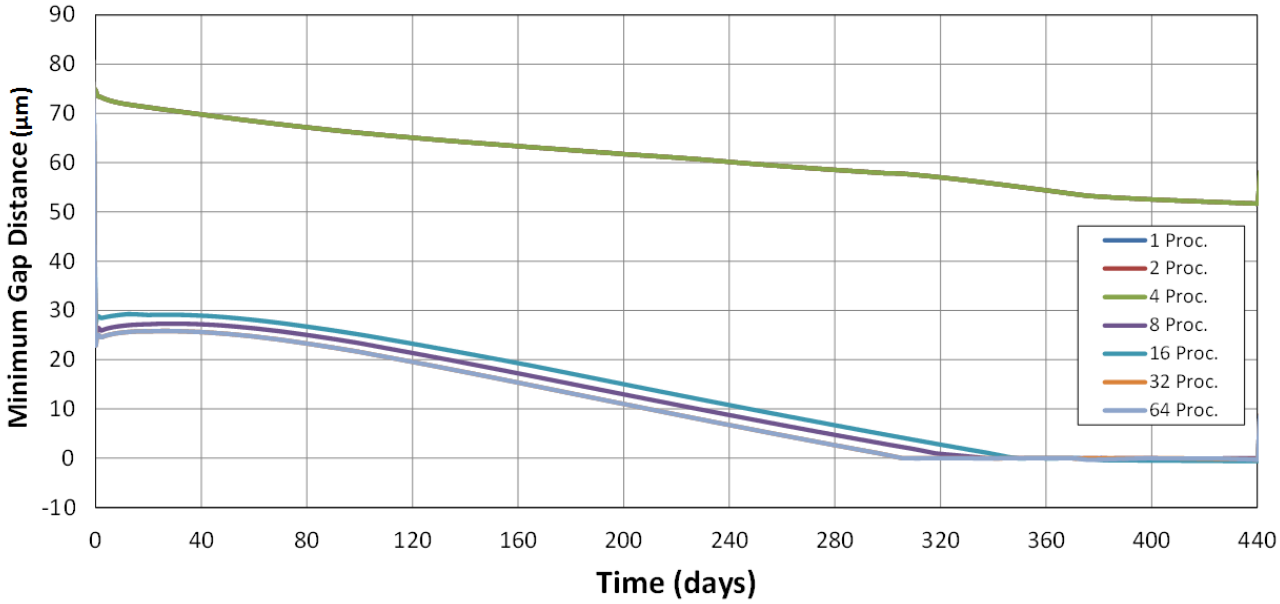


Figure 4.1.7. Single Pin – Minimum Gap Distance (µm).

4.2 3 × 3 Cross Multicycle Depletion without Shuffling

4.2.1 Input Data

The 3 × 3 cross problem presented here is inspired by the 3 × 3 cross problem that was evaluated by Tiamat previously [2,3,4]. Figure 4.2.1 shows the geometry and material layout of a 2D slice of the problem. This problem was run with quarter symmetry and baffle/reflector assemblies.

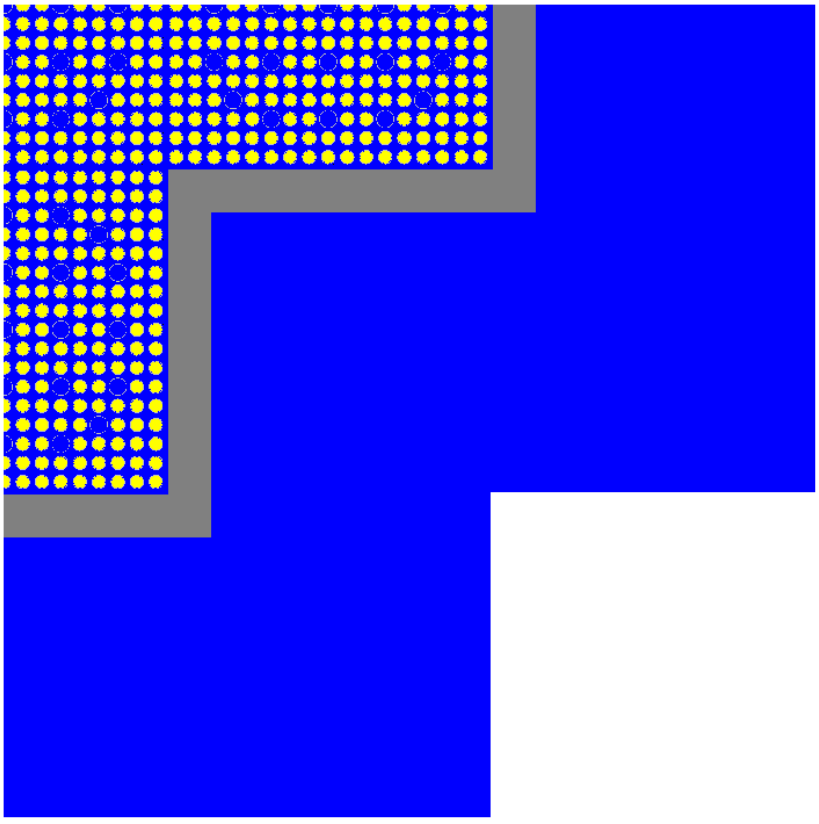


Figure 4.2.1. 3 × 3 Cross – Geometry

To generate a representative linear heat rate profile as might have been seen in the highest power pin [15], the core was run at 67 megawatts (MW) for the first cycle and 50 MW for the second cycle. With this configuration, Figure 4.2.2 shows the linear heat rate history for one of the centermost pins. Both cycles were run at 100% power for 440 EFPDs with a 44-day outage between the cycles.

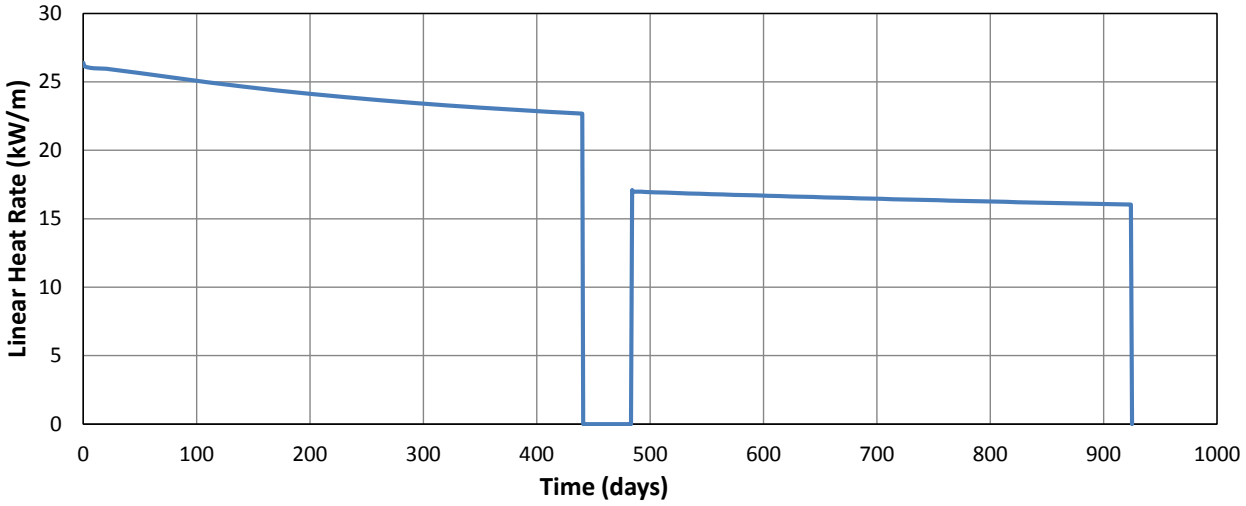


Figure 4.2.2. 3 × 3 Cross (Simplified TH) – Linear Heat Rate (kW/m) of Highest Power Pin.

As noted previously, the VERA-CS results are generated with the 8-group test library, so the computational burden of the radiation transport was minimized. Because the gradients in this core are fairly substantial and CTF is currently limited in its parallelization, the runtime with CTF was considerably larger than with the internal simplified TH model available in MPACT. Therefore, for the initial demonstration, results were obtained using the simplified TH model, which uses assembly-sized nodes for calculation. The VERA-CS results were obtained from Titan [9], running on 1,302 cores for a total of 2.1 hours of wall time between cycles 1 and 2. The 348 BISON-CASL cases were run on 60 processors in roughly 600 core hours. Eventually it was decided to run CTF without cross flow, which reduced the run time considerably (approximately 4.1 hrs on 1,302 cores), so single cycle CTF results are provided to supplement the simplified TH results. This change in TH solver does not affect the mechanics of xml2mooose in any way since the same data are still stored on the HDF5 file.

4.2.2 Output Data

Figures 4.2.3–4.2.9 show the output data obtained from the CSV file produced by BISON-CASL with the VERA-CS results with simplified TH. These figures show data for several parameters of interest, such as maximum centerline fuel temperature, average fuel temperature, average clad temperature, plenum pressure, minimum gap distance, and minimum hoop stress. The minimum hoop stress is reported since it appears to be the most limiting in clad. Figures 4.2.10–4.2.15 show the corresponding results for the first cycle with CTF.

As shown in Figures 4.2.3–4.2.9, two pins (one in the center assembly and one in the lower assembly) demonstrated convergence issues during the outage between cycles 1 and 2. This is particularly surprising since both pins are not at very high power during operation, and during outage, all pins are at zero power. It is difficult to determine that simplified TH was used, but Figure 4.2.6, which shows the average clad temperature, clearly demonstrates some of the approximations being made and that assembly-sized nodes are being used. This can be compared to Figure 4.2.12,

which shows the average clad temperature for CTF, where a much more resolved distribution is observed.

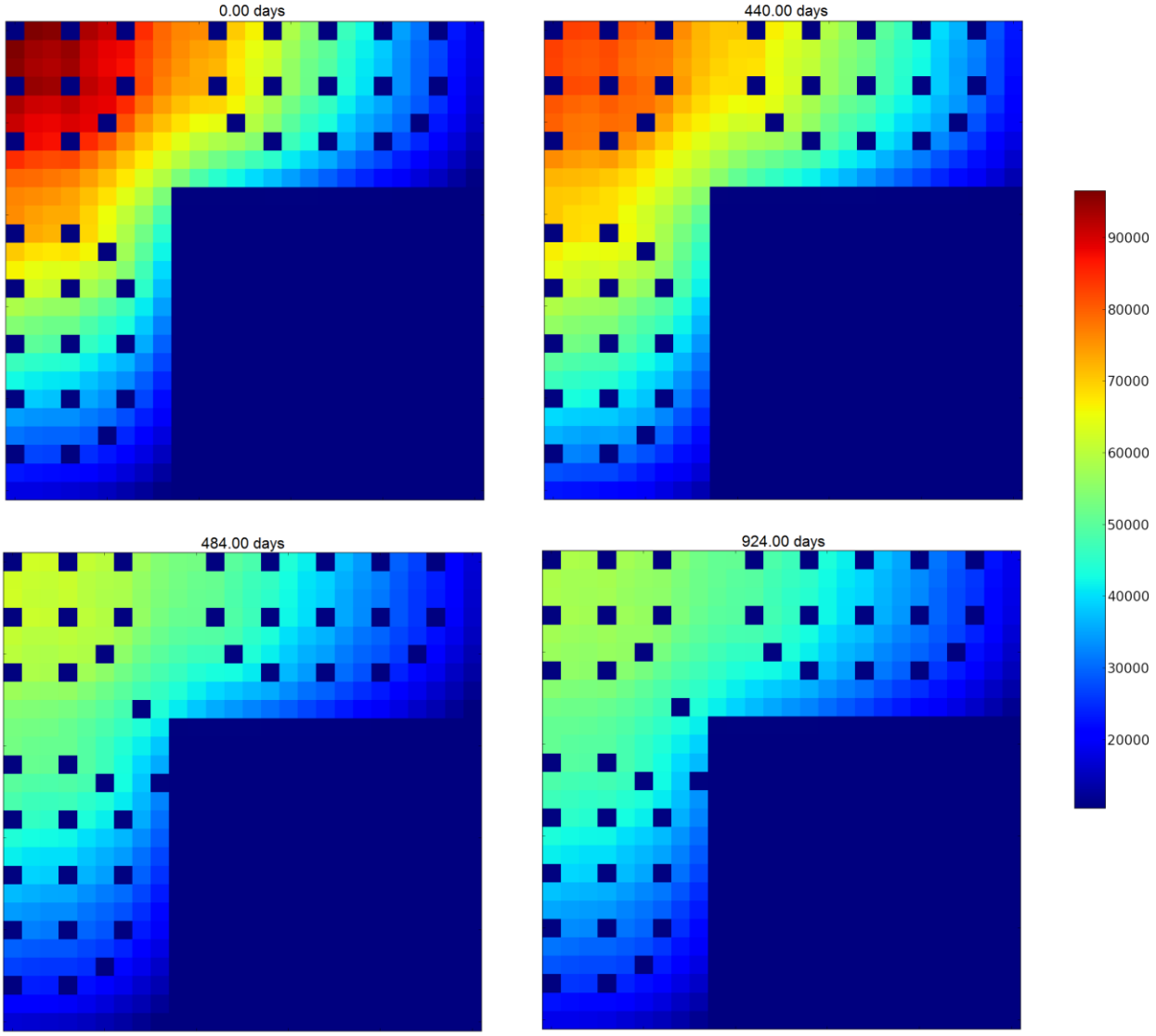


Figure 4.2.3. 3 × 3 Cross (Simplified TH) – Rod Power (W).

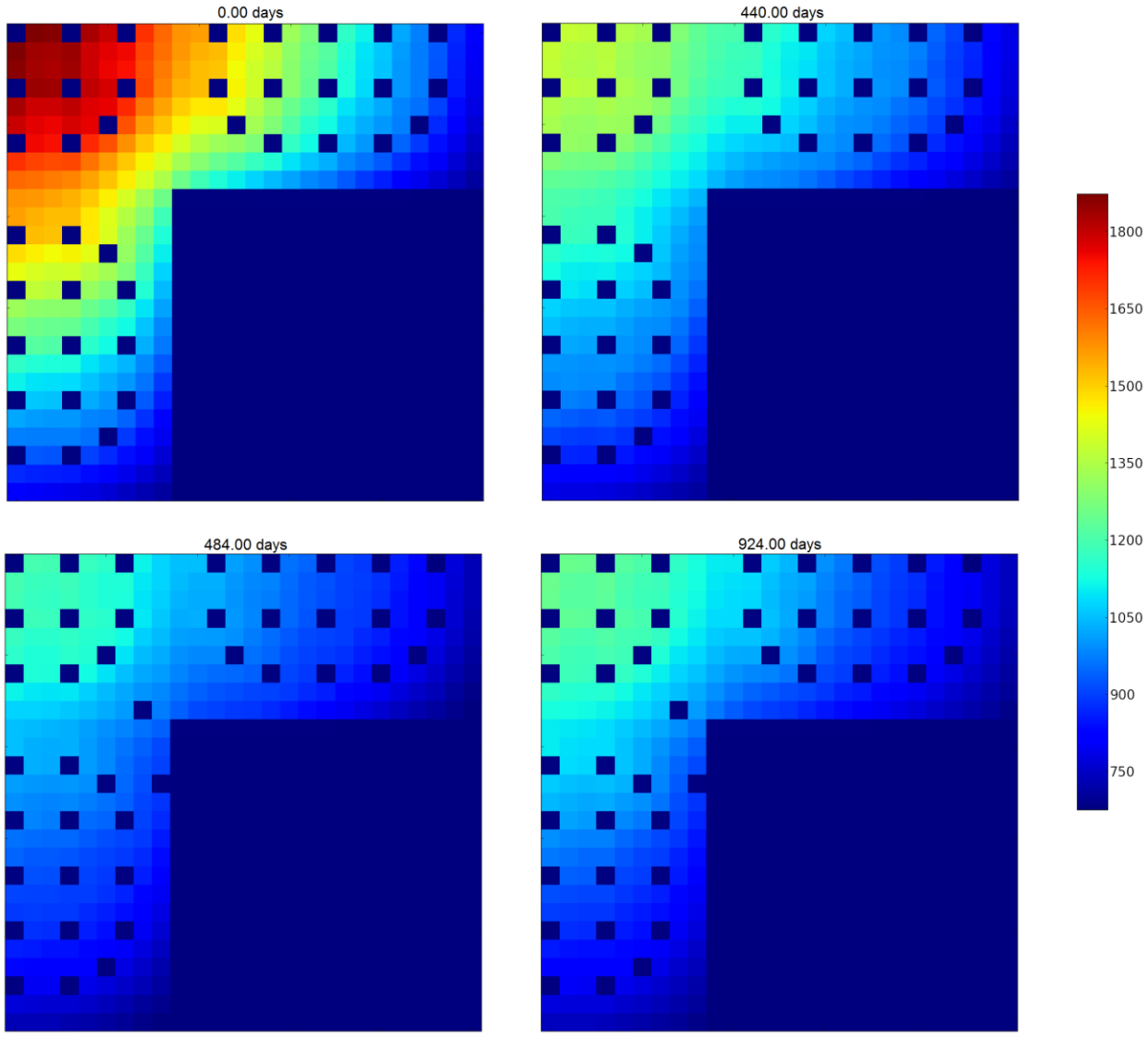


Figure 4.2.4. 3 × 3 Cross (Simplified TH) – Maximum Centerline Fuel Temperature (K).

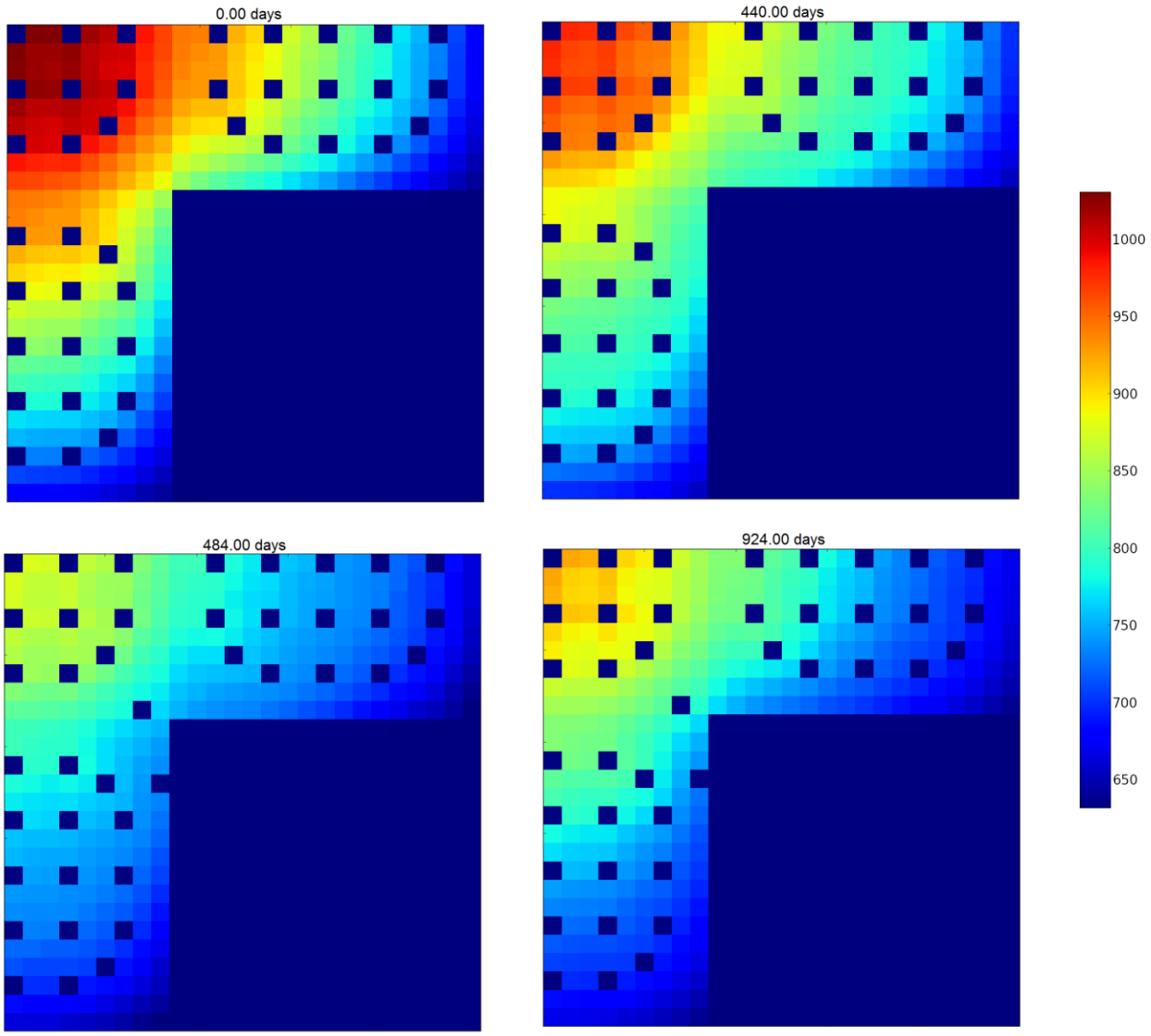


Figure 4.2.5. 3 × 3 Cross (Simplified TH) – Average Fuel Temperature (K).

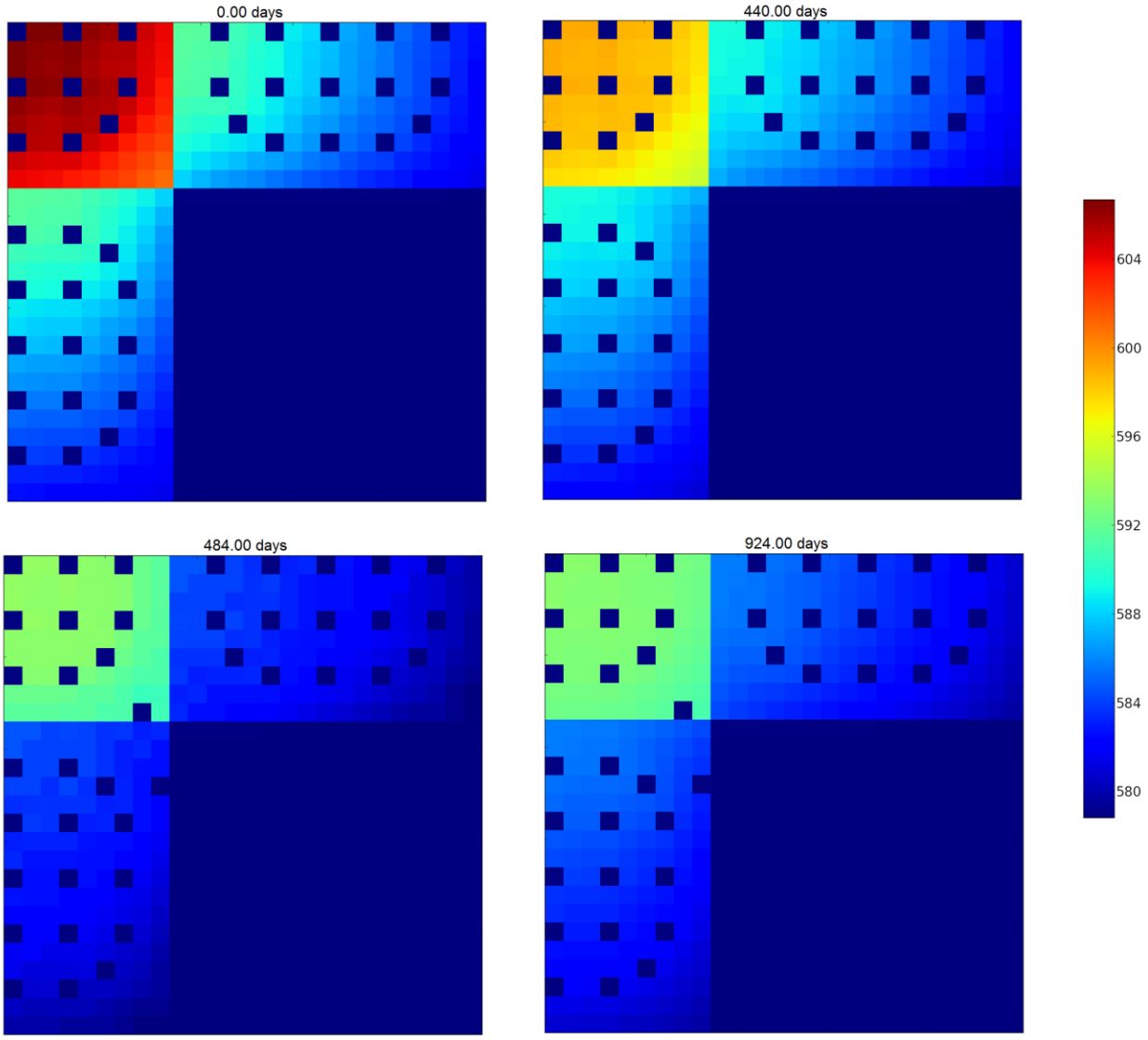


Figure 4.2.6. 3 × 3 Cross (Simplified TH) – Average Clad Temperature (K).

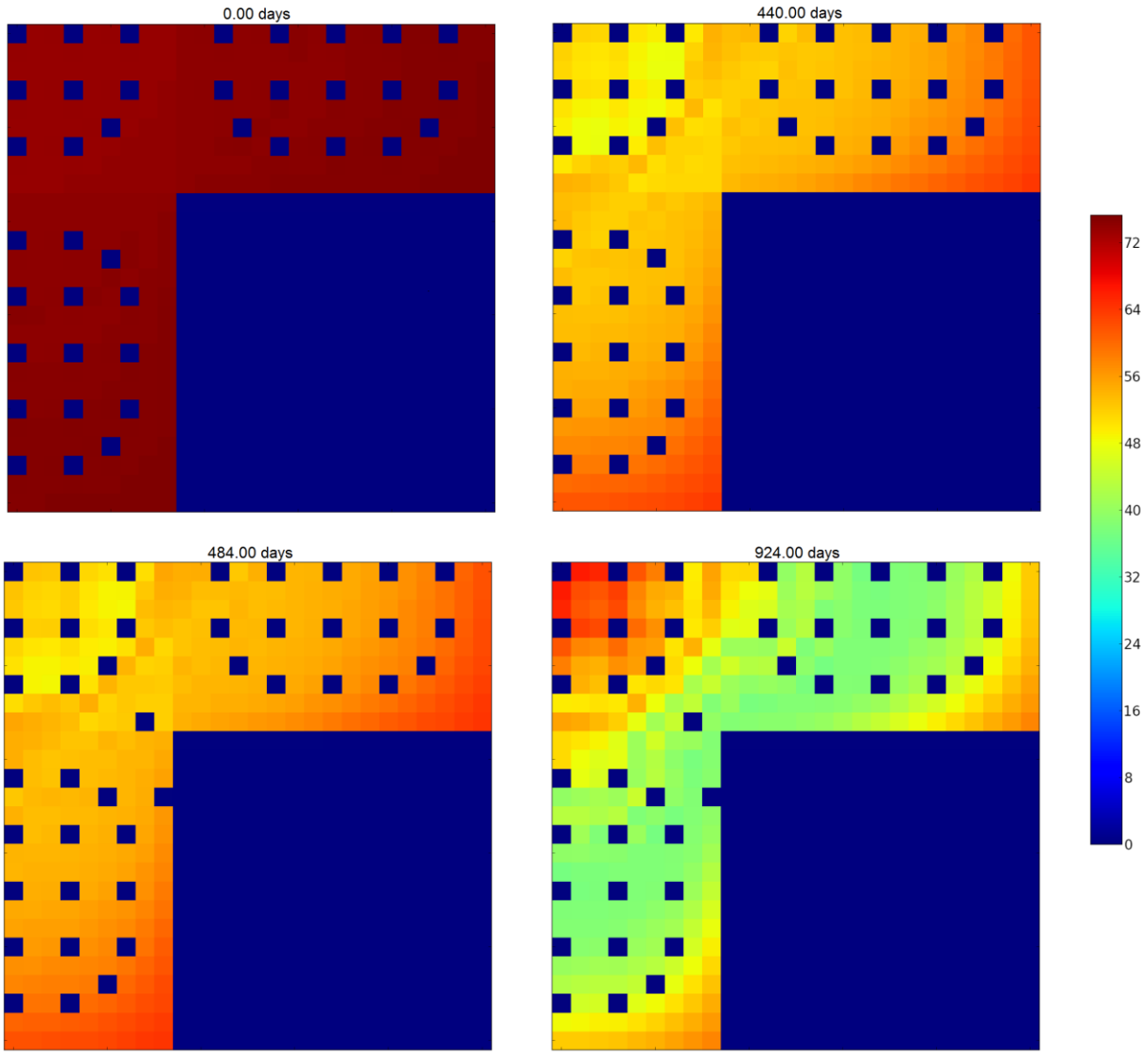


Figure 4.2.7. 3 × 3 Cross (Simplified TH) – Minimum Gap Distance (µm).

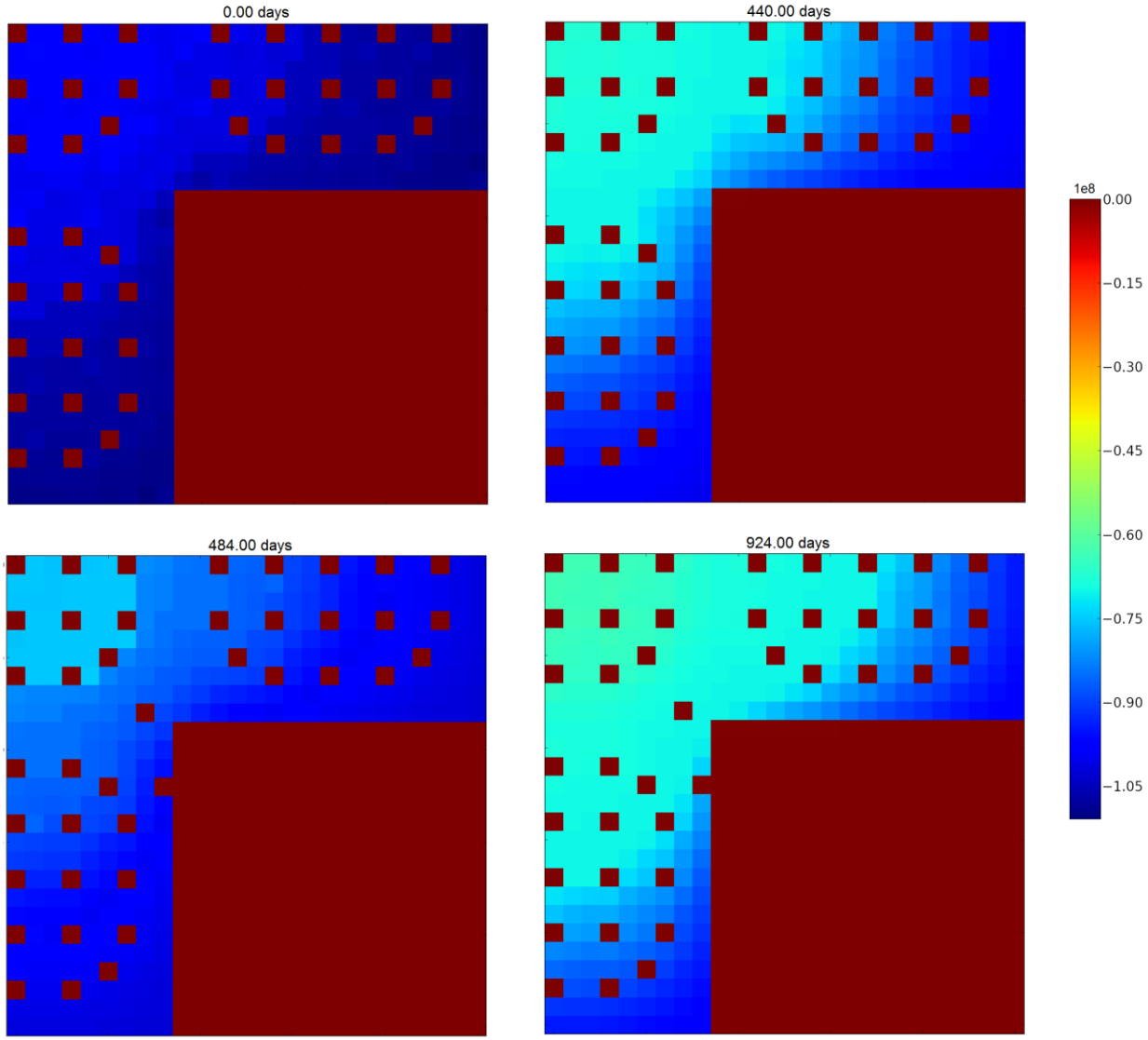


Figure 4.2.8. 3 × 3 Cross (Simplified TH) – Minimum Hoop Stress (Pa).

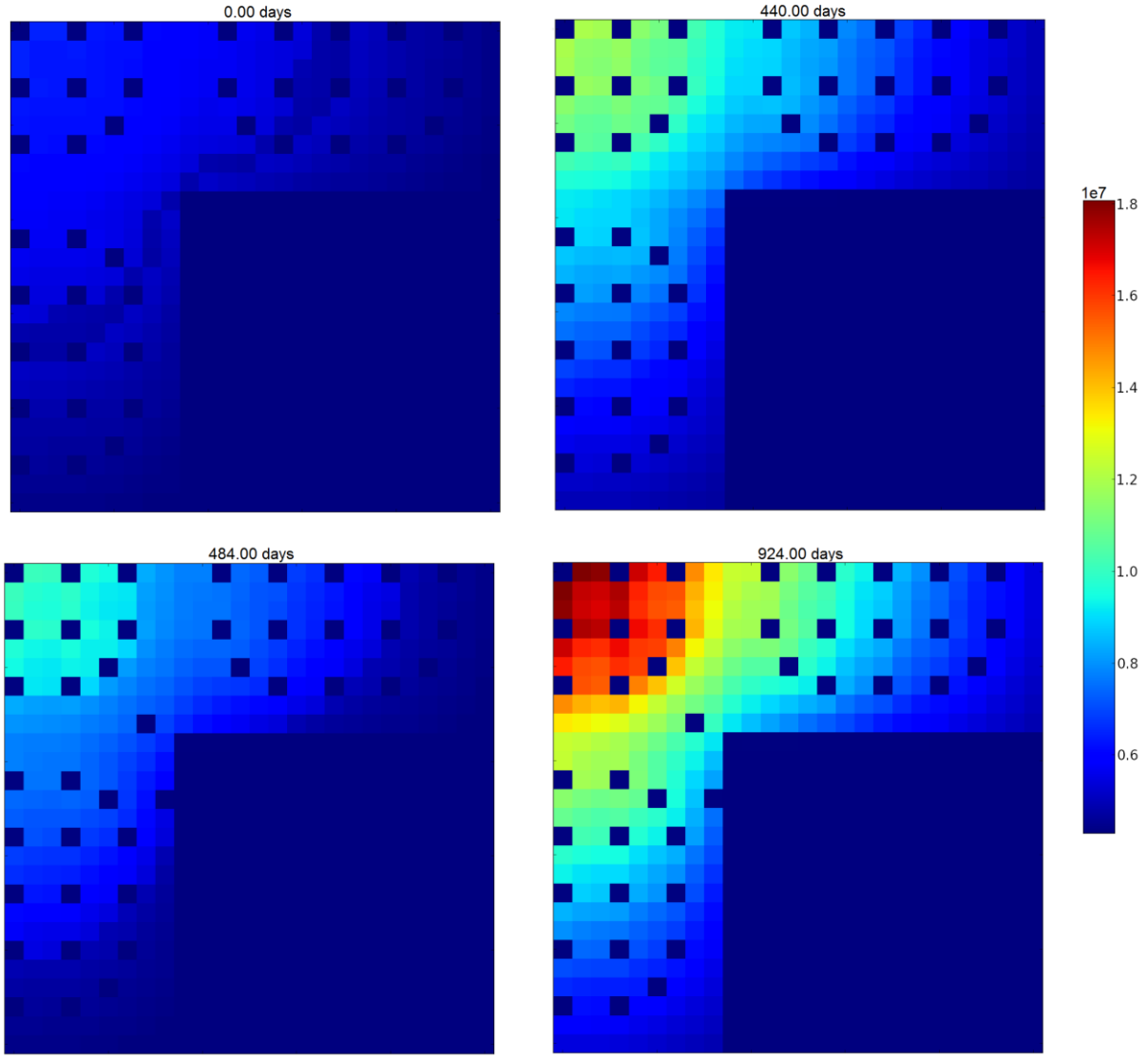


Figure. 4.2.9. 3 × 3 Cross (Simplified TH) – Plenum Pressure (Pa).

Figures 4.2.10–4.2.15 show similar results using CTF with no cross flow.

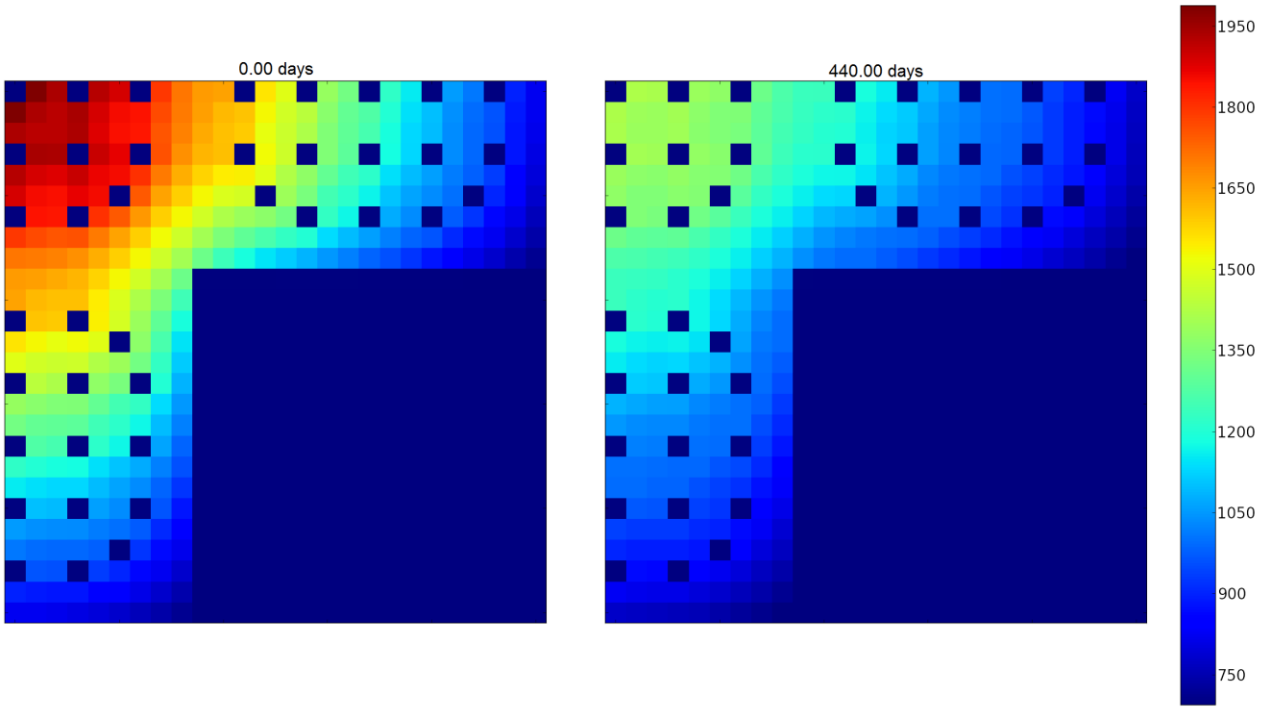


Figure 4.2.10. 3 × 3 Cross (CTF) – Maximum Centerline Fuel Temperature (K).

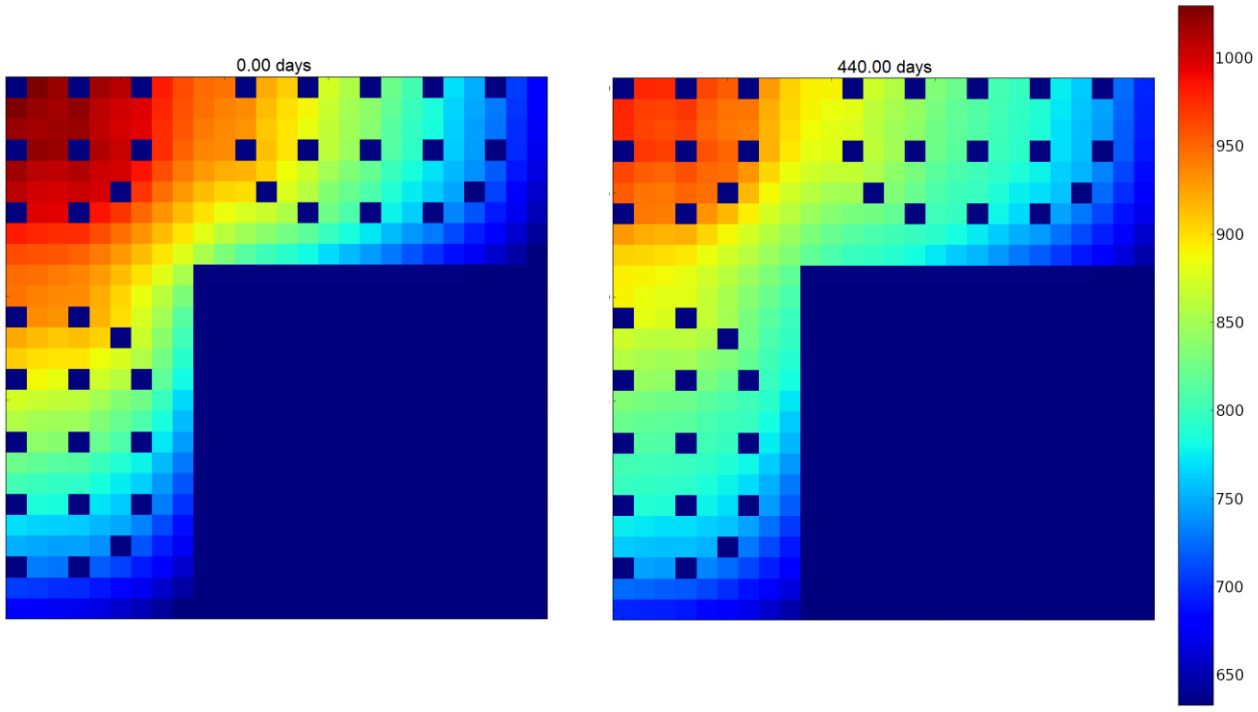


Figure 4.2.11. 3 × 3 Cross (CTF) – Average Fuel Temperature (K).

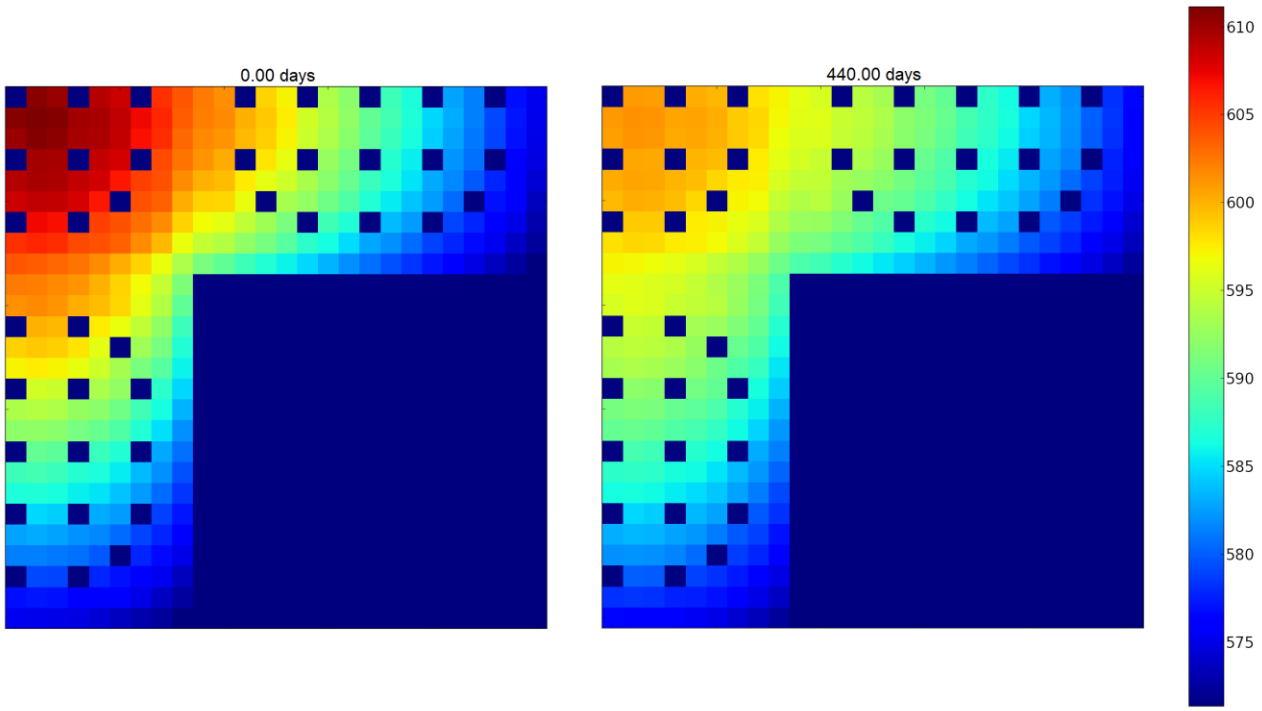


Figure 4.2.12. 3 × 3 Cross (CTF) – Average Clad Temperature (K).

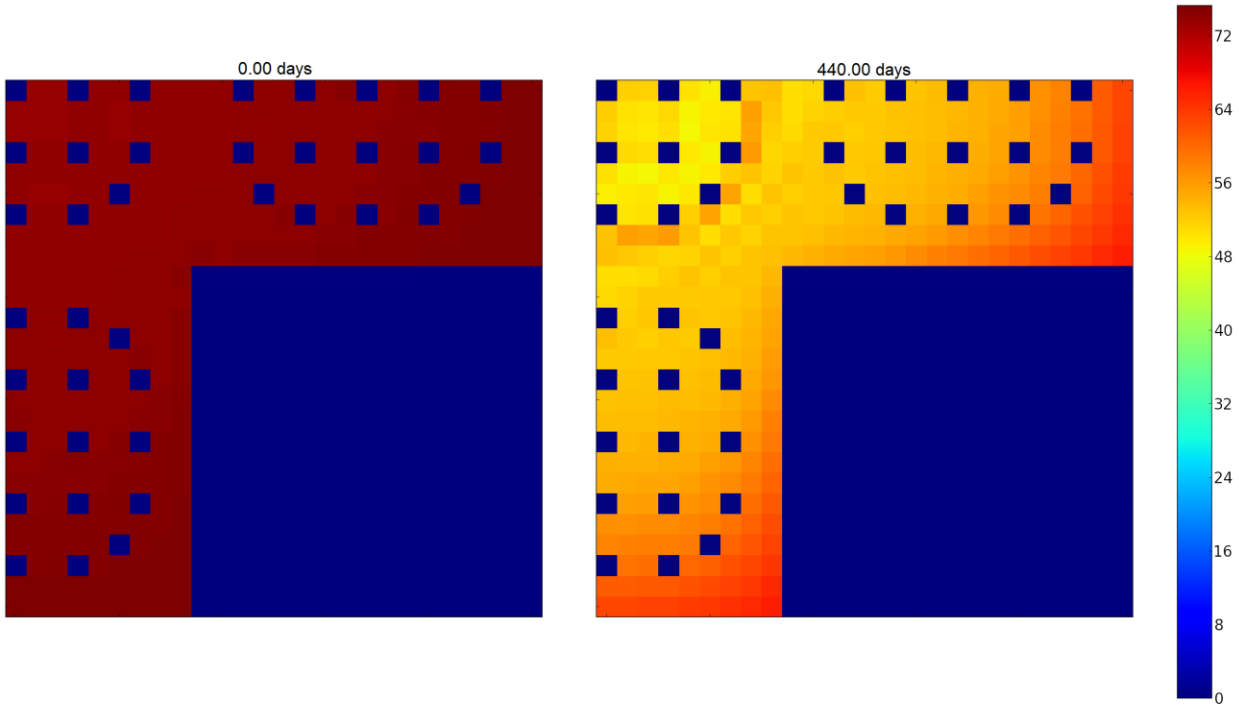


Figure 4.2.13. 3 × 3 Cross (CTF) – Minimum Gap Distance (µm).

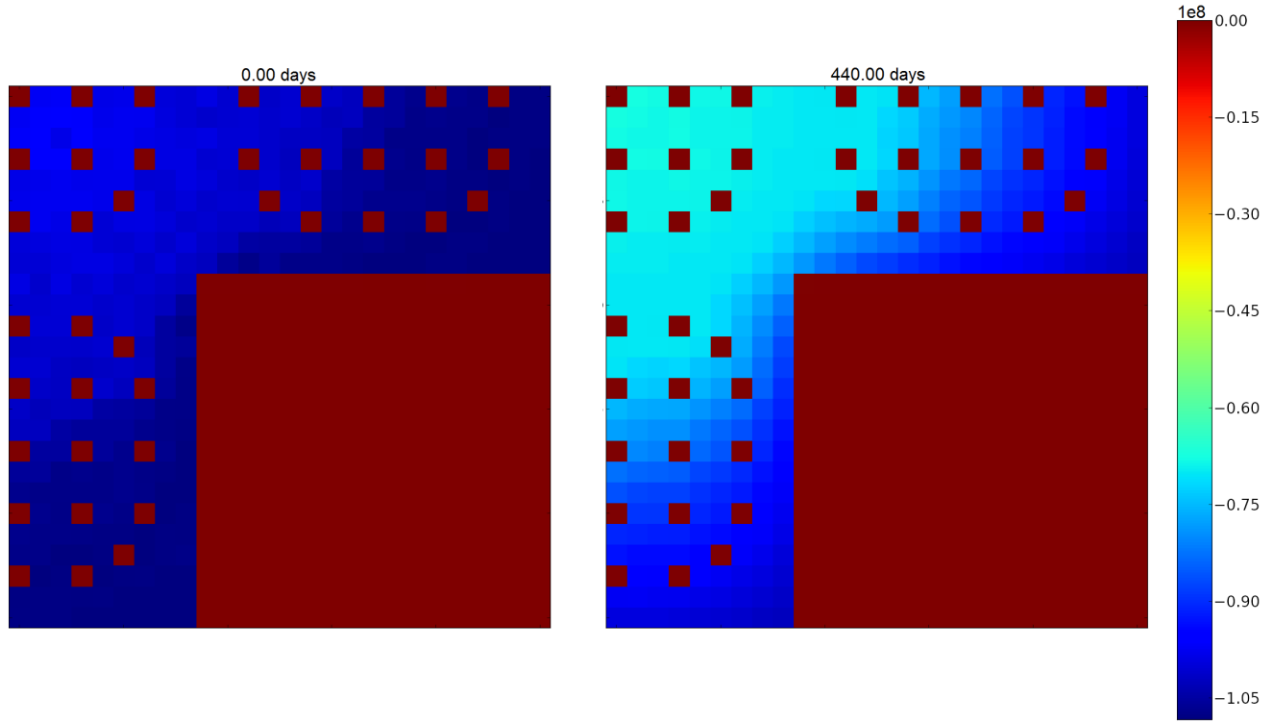


Figure 4.2.14. 3 × 3 Cross (CTF) – Minimum Hoop Stress (Pa).

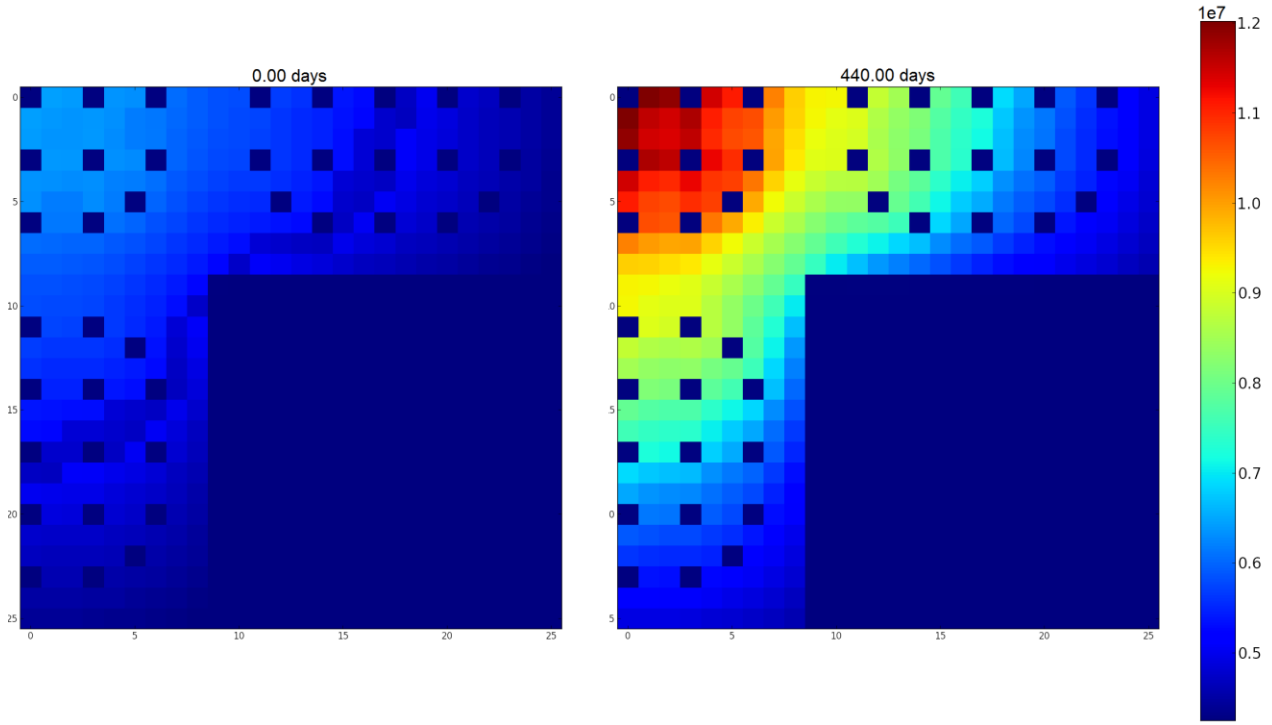


Figure 4.2.15. 3 × 3 Cross (CTF) – Plenum Pressure (Pa).

4.3 3 × 3 Core / 3 × 3 Pin Assemblies Multicycle Depletion with Shuffling

The shuffling capability was recently added, and larger scale testing is underway. However, preliminary results for a 3 × 3 assembly core with 3 × 3 pins per assembly (Figure 4.3.1) are presented to provide a basic understanding of the mechanics. Because the shuffling capability is currently limited to full symmetry, this test problem was also limited to full symmetry.

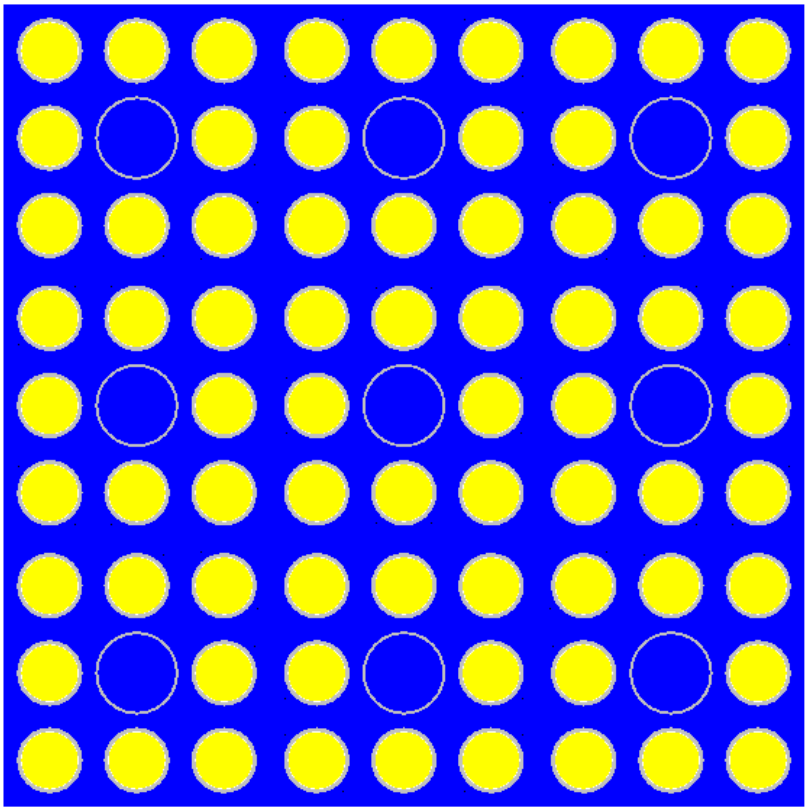


Figure 4.3.1. 3 × 3 Core / 3 × 3 Pin – Geometry.

Regarding the shuffling modifications to the xml2moose preprocessor (Section 3.2.4), one example shuffling map moved the fuel from row 2 in cycle 1 to row 1 in cycle 2, the fuel from row 1 to row 3, and inserted fresh fuel in the second row, as shown below:

```
shuffle_label  A-2 B-2 C-2
                +   +   +
               A-1 B-1 C-1
```

While this is not a very physical shuffling pattern, it is sufficient for this demonstration. As before, each cycle is run at 100% power for 440 EFPDs with a 44-day outage in between. Figures 4.3.2–4.3.4 show the rod power, average fuel temperature, and average clad temperature for each fuel rod. Not surprisingly, during the first cycle, the results are fairly flat, with some variation depending on proximity to a guide tube. In cycle two, however, since rows 1 and 3 now have once-burned fuel, and row 2 has fresh fuel, there is a larger difference in power and temperature.

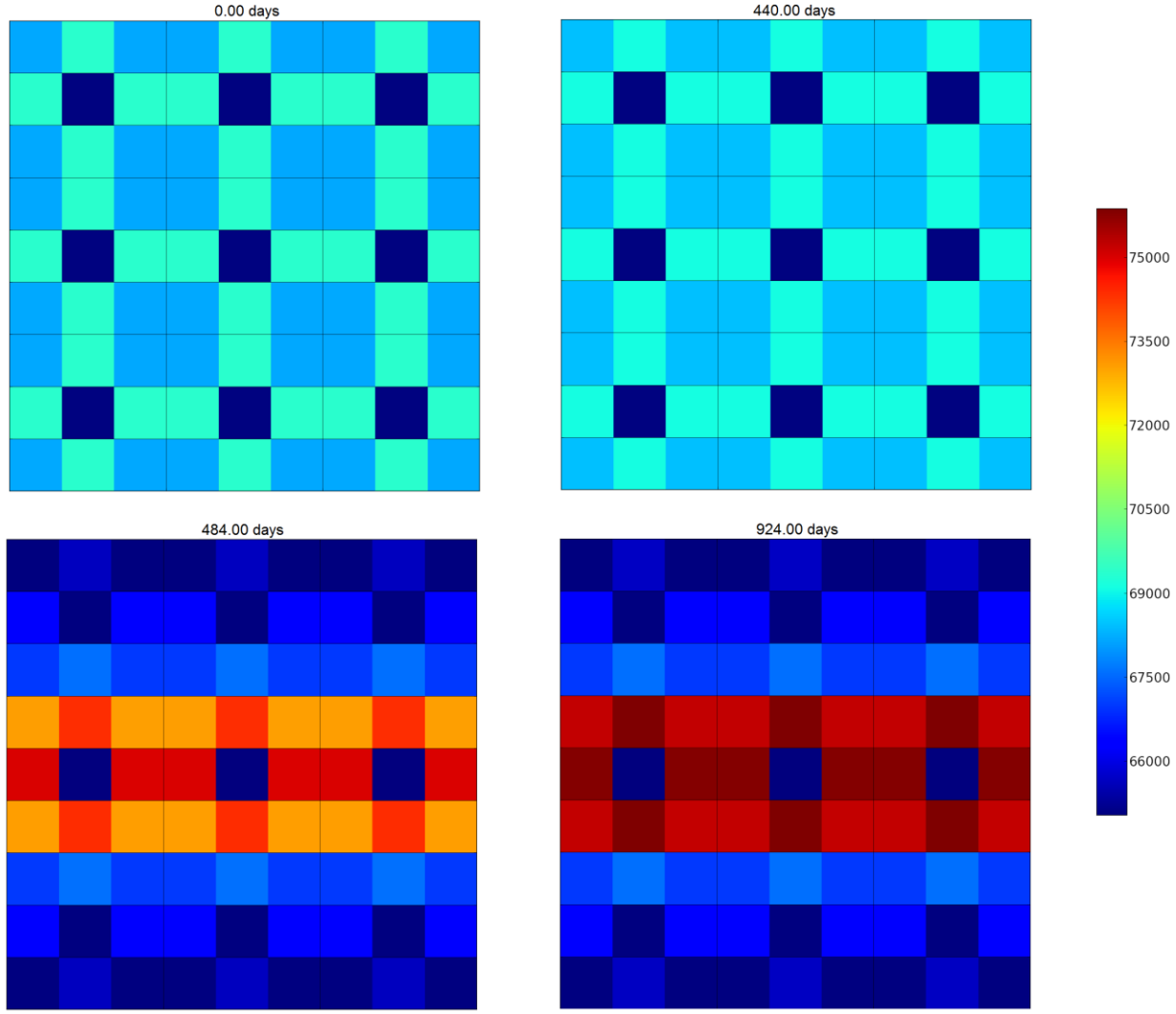


Figure 4.3.2. 3 × 3 Core / 3 × 3 Pin – Rod Power (W).

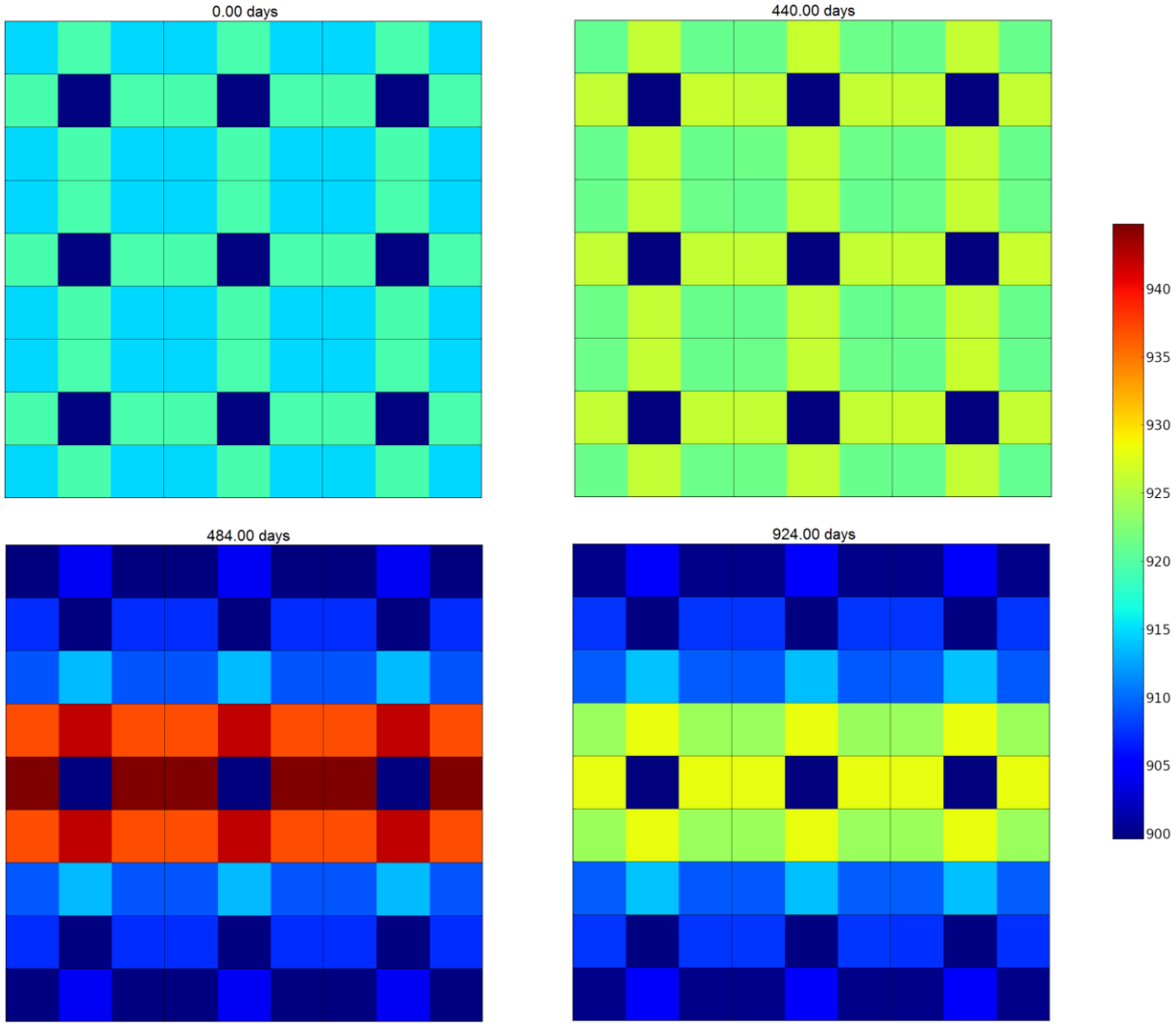


Figure 4.3.3. 3 × 3 Core / 3 × 3 Pin – Average Fuel Temperature (K).

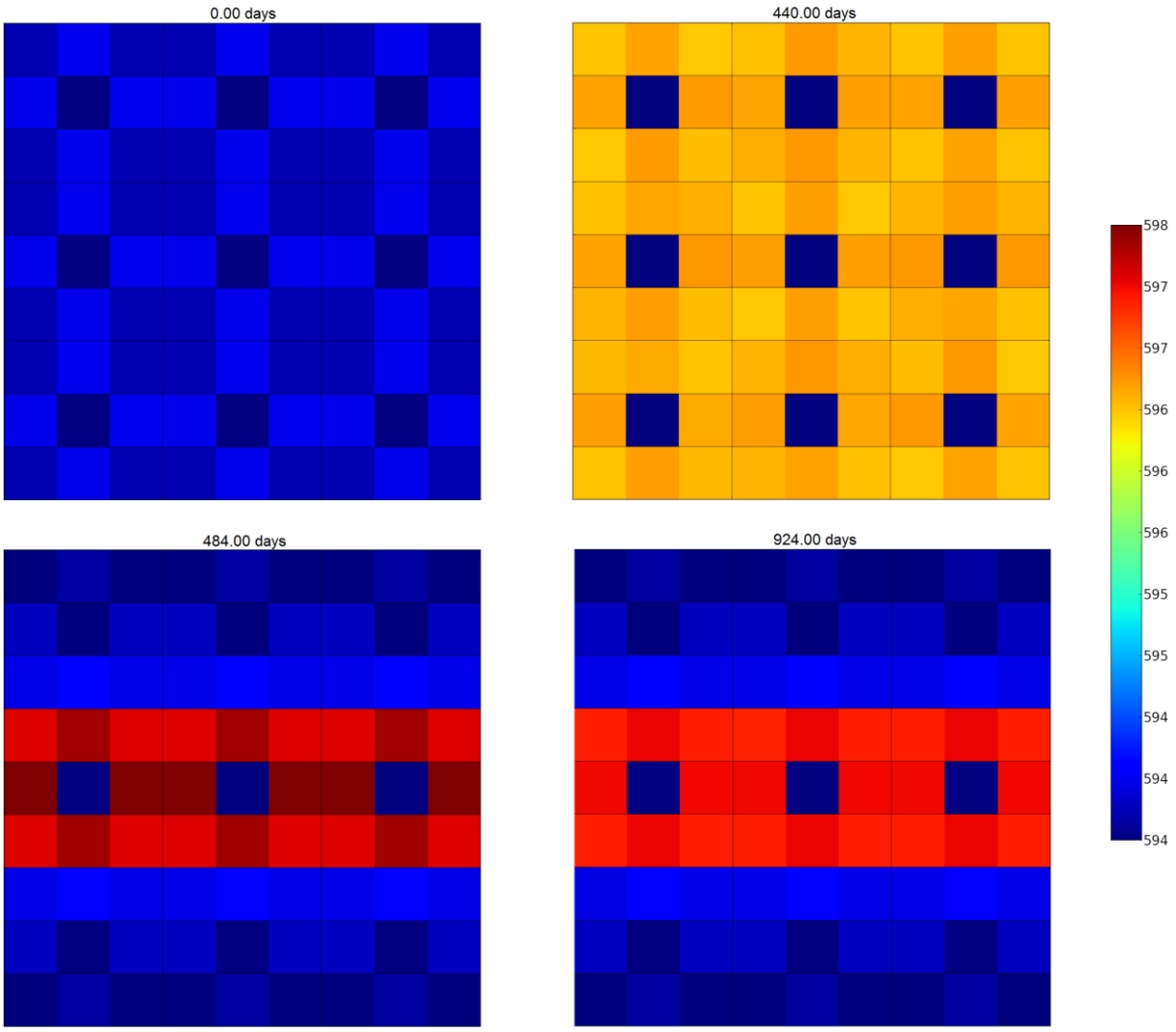


Figure 4.3.4. 3 × 3 Core / 3 × 3 Pin – Average Clad Temperature (K).

5. CONCLUSIONS, CONCERNS, AND FUTURE WORK

5.1 Summary and Conclusions

This report describes recent efforts to enable BISON-CASL inputs to be generated for each fuel rod in a specified problem based on the output from VERA-CS coupled neutronics/TH simulations. The first task was to consolidate the templates being used in several independent projects. Then modifications to the react2xml preprocessor were made to allow inputs to be sent from the VERA ASCII input to the XML file which xml2moose converts into a BISON-CASL input. The changes to xml2moose comprise most of this work, as it has been extended from use with Tiamat to generate an input for every rod in the problem with supporting CSV file inputs for power and moderator temperature. It also includes multicycle capability and initial shuffling capability, which is limited to full symmetry. These capabilities were demonstrated for a single pin and for 3 × 3 cross problems. Parameters of interest that can be obtained from the outputs include centerline fuel temperature, clad hoop stress, and plenum pressure. This information provide insights into PCI and can also be useful when taking the next steps within VERA to improve fuel temperatures (either through temperature tables or in generating informed TH parameters for CTF).

The goals of this milestone have been met, and some progress has been made on the stretch goals which will accelerate work on future related milestones.

5.2 Concerns

5.2.1 Running on Leadership Class Clusters

The BISON-CASL results presented in this report were obtained on a development cluster (ifba.ornl.gov). Future cases, which will be considerably broader in scope, need to be run on leadership class clusters such as Titan [9] and Eos [16]. However, such machines have limitations on job queueing, making the current approach of using the Python multiprocessor module to partition inputs impossible. The use of the MultiApp capability within MOOSE is being pursued as an option to accommodate the needs on these machines. MultiApp essentially spawns a subapplication for each case, and the subapplications are partitioned across all cores. While this could be sufficient for these machines, initial testing has demonstrated convergence issues, and more testing is necessary before proceeding confidently. Members of the BISON development team at INL have made suggestions and recommendations to address these issues. It may also be helpful to obtain an updated version of BISON and to ensure that an approved version of PETSc is used.

Several BISON-CASL input options must be disabled for execution on Eos. These are concerning, because some fidelity may be lost without them.

5.2.2 Convergence Issues with BISON-CASL

There are several cases with demonstrated convergence issues. Some issues may be machine specific, but should be understood regardless. Two cases failed to converge during cycle outages. These are at hot zero power without and have nothing else changing in the solution. Additional cases show severe issues when operating for several cycles.

5.3 Future Work

5.3.1 Extending Shuffle Capability

At this writing, the shuffling capability has been limited to full symmetry, so only assembly shuffling has been considered. In quarter symmetry, assemblies can originate from different

quadrants, so assembly rotation must be taken into consideration. While this will not likely be challenging, it will require additional logic and testing.

5.3.2 Accounting for More Fuel Rod Types

The current BISON-CASL template is limited to use for standard UO_2 fuel pins. Additional work should be performed to generalize the input for other types of rods, such those with integral fuel burnable absorber (IFBA)-coated fuel and potentially gadolinium pins (UO_2-Gd_2O_3).

5.3.3 Running Quarter Core Models

Shuffle capability must be extended (Section 5.3.1) and more fuel rod types must be accounted for (Section 5.3.2) in order to reach the target application of running quarter core models, particularly based on the output from the twelve cycles of WBN1. Many of the concerns with convergence and running on leadership class facilities also must be resolved.

REFERENCES

[1] A. Godfrey et al. *VERA Benchmarking Results for Watts Bar Nuclear Plant Unit 1 Cycles 1-12*. Technical Report, CASL-U-2015-0206-000, Oak Ridge National Laboratory. http://www.casl.gov/docs/CASL-U-2015-0206-000.pdf (2015).

[2] R. P. Pawlowski, K. T. Clarno, and R. O. Montgomery. *Demonstrate Integrated VERA-CS for the PCI Challenge Problem.* Technical Report, CASL-I-2014-0153-000, Oak Ridge National Laboratory (2014).

[3] R. P. Pawlowski et al. "Design of a high fidelity core simulator for analysis of pellet clad interaction." In: *Proceedings of the ANS Joint International Conference on Mathematics and Computation (M&C 2015), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method*. Nashville, TN, USA (2015).

[4] K. T. Clarno et al. "High fidelity modeling of pellet-clad interaction using the CASL virtual environment for reactor applications." In: *Proc. ANS Joint International Conference on Mathematics and Computation (M&C 2015), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method*. Nashville, TN, USA (2015).

[5] J. A. Turner. *Virtual Environment for Reactor Applications (VERA): Snapshot 3.1.* Technical Report, CASL-U-2013-0164-000, Oak Ridge National Laboratory. http://www.casl.gov/docs/CASL-U-2013-0164-000.pdf (2013).

[6] *MPACT Theory Manual*. Technical Report, University of Michigan (2013).

[7] B. Collins et al. "Assessment of 2D/1D Capability in MPACT," *Proc. PHYSOR 2014*, Kyoto, Japan, September 28–October 3 (2014).

[8] S. G. Stimpson, B. S. Collins, T. J. Downar. "Axial Transport Solvers for the 2D/1D Scheme in MPACT," *Proc. PHYSOR 2014*, Kyoto, Japan, September 28 –October 3 (2014).

[9] Oak Ridge Leadership Computing Facility. "Introducing Titan - The World's #1 Open Science Supercomputer" (2014), http://www.olcf.ornl.gov/titan/.

[10] M. N. Avramova. *CTF: A Thermal Hydraulic Sub-Channel Code for LWR Transient Analyses, Users Manual.* Technical Report, Pennsylvania State University, Department of Nuclear Engineering (2009).

[11] R. O. Montgomery et al. "Peregrine: Advanced modeling of pellet-cladding interaction (pci) failure in lwrs." In: *Proc. TopFuel 2012 Reactor Fuel Performance Conference.* Manchester, United Kingdom (2012).

[12] R. O. Montgomery et al. "Advanced pellet-cladding interaction modeling using the US DOE CASL fuel performance code: Peregrine." In: *Transactions of the American Nuclear Society Annual Meeting*. Reno, Nevada (2014).

[11] D. Gaston et al. "Moose: A parallel computational framework for coupled systems of nonlinear equations." *Nuclear Engineering Design*, 239: pp. 1768–1778 (2009).

[14] R. Williamson et al. "Multidimensional multiphysics simulation of nuclear fuel behavior." *Journal of Nuclear Materials*, 423: pp. 149–163 (2012).

[15] I. Porter, T. W. Knight, and P. Reynaud. "Potential Impacts of Modeling Full Reactor Cores Using Combined Fuel Performance and Thermal Hydraulics Codes," *Nuclear Technology*, 190: pp. 174–182 (2015).

[16] Oak Ridge Leadership Computing Facility, *Eos User Guide* (2014), http://www.olcf.ornl.gov/support/system-user-guides/eos-user-guide/.

APPENDIX A – BISON-CASL INPUT TEMPLATE

```
# ================================================== #
# BISON-CASL Fuel Pin Input Template
#             (see the bottom for uses and assumptions)
# ================================================== #
# Nomenclature:
#  -- #VERA_DEFINED    = this value will  be overwritten by xml2moose
#  -- #VERA_MODIFIABLE = this value could be overwritten by xml2moose
#  -- #VERA_PREPARED   = this value is specified when created for Tiamat or from VERA-CS output; blank otherwise
#  -- #TIAMAT          = enabled for Tiamat
#  -- #BISON_CASL      = enabled for stand-alone BISON-CASL
#  -- #CLAD_SURFACE    = enabled when Dirichlet BCs on the clad surface are used.
#  -- #COOLANT_TEMP    = enabled when Dirichlet BCs for the bulk coolant temperature are used.
#  -- #COOLANT_FLOW    = enabled when Coolant Flow BCs are used.
# ================================================== #

# ================================================== #
# Global Parameters used throughout the input file
# ================================================== #
[GlobalParams]
  density =                                                 #VERA_DEFINED
  disp_x = disp_x
  disp_y = disp_y
  disp_r = disp_x
  disp_z = disp_y
  order = FIRST                                             #VERA_MODIFIABLE as globalparams_order
  family = LAGRANGE
  energy_per_fission = 3.2e-11    # J/fission               #VERA_MODIFIABLE as globalparams_energy_per_fission
  a_lower =                                                 #VERA_DEFINED
  a_upper =                                                 #VERA_DEFINED
  initial_porosity =                                        #VERA_DEFINED (used in ./fuel_thermal and ./fission_gas_release)
[]

# ================================================== #
# Mesh (and Geometry, if internally-meshed)
# ================================================== #
[Mesh]
  displacements = 'disp_x disp_y'
  file =                                                    #BISON_CASL (conditionally included if mesh_file is specified)
  patch_size = 20      # For contact algorithm              #VERA_MODIFIABLE as mesh_patch_size
  type = SmearedPelletMesh                                  # (omitted if file is specified)
  clad_mesh_density = customize                             # (omitted if file is specified)
  pellet_mesh_density = customize                           # (omitted if file is specified)
  clad_thickness =                                          #VERA_DEFINED (omitted if file is specified)
  pellet_outer_radius =                                     #VERA_DEFINED (omitted if file is specified)
  clad_bot_gap_height =                                     #VERA_DEFINED (omitted if file is specified)
  clad_top_gap_height =                                     #VERA_DEFINED (omitted if file is specified)
  pellet_quantity =                                         #VERA_DEFINED (omitted if file is specified)
  pellet_height =                                           #VERA_DEFINED (omitted if file is specified)
  clad_gap_width =                                          #VERA_DEFINED (omitted if file is specified)
  top_bot_clad_height =                                     #VERA_DEFINED (omitted if file is specified)
  nx_p =  6   # number of radial elements in the fuel  #VERA_MODIFIABLE as mesh_nx_p  (omitted if file is specified)
  ny_p = 100  # number of axial elements in the fuel      #VERA_MODIFIABLE as mesh_ny_p  (omitted if file is specified)
  nx_c =  3   # number of elements in the clad thickness        #VERA_MODIFIABLE as mesh_nx_c  (omitted if file is specified)
  ny_c = 100  # number of elements in the axially in the clad #VERA_MODIFIABLE as mesh_ny_c  (omitted if file is specified)
  ny_cu = 1                                                 #VERA_MODIFIABLE as mesh_ny_cu (omitted if file is specified)
  ny_cl = 1                                                 #VERA_MODIFIABLE as mesh_ny_cl (omitted if file is specified)
  intervals =                                               #VERA_DEFINED (omitted if file is specified)
  elem_type = QUAD4                                         #VERA_MODIFIABLE as mesh_elem_type (omitted if file is specified)
[]

# ================================================== #
# Dimensions and Primary Variables
# ================================================== #
[Problem]
  coord_type = RZ
[]
[Variables]
  [./disp_x]
  [../]
  [./disp_y]
  [../]
  [./temp]
    initial_condition = 3.000000e+02                        #VERA_MODIFIABLE as variables_temp_initial_condition
  [../]
[]
[SolidMechanics]
  [./solid]
    temp = temp
  [../]
[]

# ================================================== #
# Auxiliary Variables
# ================================================== #
[AuxVariables]

  # ================================================== #
  # Nodal Quantities
  # ================================================== #

  [./htcl]
    initial_condition = 500.0                               #VERA_MODIFIABLE as auxvariables_htcl_initial_condition
  [../]

  [./htcv]
    initial_condition = 0.0                                 #VERA_MODIFIABLE as auxvariables_htcv_initial_condition
  [../]
```

```
[./Tl]
  initial_condition = 565.0                                  #VERA_MODIFIABLE as auxvariables_Tl_initial_condition
[../]

[./Tv]
  initial_condition = 565.0                                  #VERA_MODIFIABLE as auxvariables_Tv_initial_condition
[../]

[./burnup]
  block = 3
[../]

[./fast_neutron_flux]
  block = 1
[../]

[./fast_neutron_fluence]
  block = 1
[../]

[./fission_rate]
  initial_condition =  0
[../]

[./grain_radius]
  block = 3
  initial_condition = 5.240000e-06                           #VERA_MODIFIABLE as auxvariables_grain_radius_initial_condition
[../]

[./casl_fission_rate] #>> PURGED FROM STANDALONE CASES
  initial_condition = 0.0
[../]

[./casl_clad_surface_temperature] #>> PURGED FROM STANDALONE CASES
  #Hot-Zero Power coolant temperature.
  initial_condition = 565.0                                  #VERA_MODIFIABLE as auxvariables_casl_clad_surface_temperature
[../]

# ================================================ #
# Constant Monomial Quantities (Mechanics)
# ================================================ #
[./hydrostatic_stress]
  order = CONSTANT
  family = MONOMIAL
[../]

[./radial_stress]
  order = CONSTANT
  family = MONOMIAL
[../]

[./axial_stress]
  order = CONSTANT
  family = MONOMIAL
[../]

[./hoop_stress]
  order = CONSTANT
  family = MONOMIAL
[../]

[./hoop_strain]
  order = CONSTANT
  family = MONOMIAL
[../]

[./radial_strain]
  order = CONSTANT
  family = MONOMIAL
[../]

[./axial_strain]
  order = CONSTANT
  family = MONOMIAL
[../]

[./vonmises]
  order = CONSTANT
  family = MONOMIAL
[../]

[./creep_strain_mag]
  order = CONSTANT
  family = MONOMIAL
[../]

[./creep_strain_radial]
  order = CONSTANT
  family = MONOMIAL
[../]

[./creep_strain_axial]
  order = CONSTANT
  family = MONOMIAL
[../]

[./creep_strain_hoop]
  order = CONSTANT
  family = MONOMIAL
```

```
  [../]

  # ================================================== #
  # Constant Monomial Quantities (Non-Mechanics)
  # ================================================== #
  [./pellet_id]
    order = CONSTANT
    family = MONOMIAL
    block = 3
  [../]

  [./axial_fission_rate]
    order = CONSTANT
    family = MONOMIAL
  [../]

  [./axial_burnup]
    order = CONSTANT
    family = MONOMIAL
  [../]

  [./axial_temperature]
    order = CONSTANT
    family = MONOMIAL
  [../]

  [./gap_conductivity]
    order = CONSTANT
    family = MONOMIAL
  [../]

  [./fuel_conductivity]
    order = CONSTANT
    family = MONOMIAL
  [../]

  [./porosity]
    order = CONSTANT
    family = MONOMIAL
    initial_condition =                                     #VERA_DEFINED
  [../]

  [./gap_distance]
    order = CONSTANT
    family = MONOMIAL
    initial_condition =                                     #VERA_DEFINED
  [../]
[]

# ================================================== #
# Time- and Space-Dependent Source and BCs
# ================================================== #
[Functions]
  [./linear_heat_rate_profile]  #>> replaced power_history and power_profile
    type = PiecewiseLinear
    x = '-100 0 5000'                                       #VERA_MODIFIABLE as functions_linear_heat_rate_profile_x
    y = '0 0 25000'                                         #VERA_MODIFIABLE as functions_linear_heat_rate_profile_y
    data_file =                                             #BISON_CASL as functions_linear_heat_rate_profile_data_file
    format = columns                                        #BISON_CASL as functions_linear_heat_rate_profile_format
    scale_factor = 1                                        #VERA_MODIFIABLE as functions_linear_heat_rate_profile_scale_factor
  [../]

  [./axial_peaking_factors]
    type =                                                  #TIAMAT/BISON_CASL as functions_axial_peaking_factors_type (defaults
to ParsedFunction if TIAMAT, PiecewiseBilinear if BISON_CASL (data_file necessary))
    value = 1                                               #VERA_MODIFIABLE as functions_axial_peaking_factors_value for TIAMAT
    data_file =                                             #VERA_MODIFIABLE as functions_axial_peaking_factors_data_file for
BISON_CASL
    axis = 1                                                #BISON_CASL
    scale_factor = 1                                        #BISON_CASL
  [../]

  [./coolant_temperature]
    type = PiecewiseLinear                                  #VERA_MODIFIABLE as functions_coolant_temperature_type (defaults to
PiecewiseBilinear if BISON_CASL (data_file necessary))
    data_file = user_coolant_temp.csv                       #VERA_MODIFIABLE as functions_coolant_temperature_data_file (omitted
otherwise)
    x = '-100 0'                                            #VERA_MODIFIABLE as functions_coolant_temperature_x (omitted if
data_file included)
    y = '293 565'                                           #VERA_MODIFIABLE as functions_coolant_temperature_y (omitted if
data_file included)
    axis = 1
  [../]

  [./coolant_pressure_ramp]                                 #>> CHECK: Should start at atmospheric, not Zero pressure
    # used in coolantPressure BC
    type = PiecewiseLinear
    scale_factor = 1
    x = '-100 0'                                            #VERA_MODIFIABLE as functions_coolant_pressure_ramp_x
    y = '0 1.0'                                             #VERA_MODIFIABLE as functions_coolant_pressure_ramp_y
  [../]

  [./q]
    type = CompositeFunction
    functions = 'linear_heat_rate_profile axial_peaking_factors' #VERA_MODIFIABLE as functions_q_functions
  [../]
[]

# ================================================== #
# Burnup Equation Set
```

```
# ================================================== #
[Burnup]
  [./burnup]
    block = 3
    rod_ave_lin_pow = linear_heat_rate_profile                #VERA_MODIFIABLE as burnup_burnup_rod_ave_lin_pow
    axial_power_profile = axial_peaking_factors               #VERA_MODIFIABLE as burnup_burnup_axial_power_profile
    num_radial = 6                                            #VERA_MODIFIABLE as burnup_burnup_num_radial
    num_axial = 48                                            #VERA_MODIFIABLE as burnup_burnup_num_axial
    fuel_inner_radius = 0.0                                   #VERA_MODIFIABLE as burnup_burnup_fuel_inner_radius
    fuel_outer_radius =                                       #VERA_DEFINED
    fuel_volume_ratio = 1.0                                   #VERA_MODIFIABLE as burnup_burnup_fuel_fuel_volume_ratio
    i_enrich = '3.100e-02 9.690e-01 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00'  #VERA_DEFINED (overridable) or VERA_MODIFIABLE as
burnup_burnup_i_enrich
    RPF = RPF
  [../]
[]

# ================================================== #
# Coolant Channel Equation Set (#COOLANT_FLOW)
# ================================================== #
#[CoolantChannel]
#
#  [./convective_clad_surface]
#    boundary = '1 2 3'
#    variable = temp
#    inlet_temperature =                                    #VERA_DEFINED    TODO: check units
#    inlet_pressure =                                       #VERA_DEFINED    TODO: check units
#    inlet_massflux =                                       #VERA_DEFINED    TODO: check units
#    rod_diameter =                                         #VERA_DEFINED    TODO: check units
#    rod_pitch =                                            #VERA_DEFINED    TODO: check units
#    linear_heat_rate = linear_heat_rate_profile            #VERA_MODIFIABLE as
coolantchannel_convective_clad_surface_linear_heat_rate_profile
#    axial_power_profile = axial_peaking_factors            #VERA_MODIFIABLE as
coolantchannel_convective_clad_surface_axial_power_profile
#  [../]
#[]

# ================================================== #
# Primary Kernels used in Heat Transfer
# ================================================== #
[Kernels]

  [./gravity]
    type = Gravity
    variable = disp_y
    value = -9.81
  [../]

  [./heat]
    # gradient term in heat conduction equation
    type = HeatConduction
    variable = temp
  [../]

  [./heat_ie]
    # time term in heat conduction equation
    type = HeatConductionTimeDerivative
    variable = temp
  [../]

  [./heat_source]
    type = NeutronHeatSource                                  #CASL_MOOSE_MK_Source if TIAMAT, NeutronHeatSource if BISON_CASL
    fission_rate = fission_rate                               #BISON_CASL
    aux_var =                                                 #TIAMAT sets to casl_fission_rate
    variable = temp
    block = 3
  [../]
[]

[AuxKernels]

  # ================================================== #
  # Pre-Defined Types
  # ================================================== #

  [./pelletid]
    type = PelletIdAux
    block = 3
    variable = pellet_id
    number_pellets = 1
    execute_on = initial
  [../]

  [./FissionRateAux]   #>>PURGED FROM TIAMAT CASES
    type = FissionRateAux
    value = 0.59158e15                                          #VERA_MODIFIABLE as auxkernels_fissionrateaux_value
    function = q                                                #VERA_MODIFIABLE as auxkernels_fissionrateaux_function
    variable = fission_rate
    block = 3
  [../]

  [./fast_neutron_flux]
    type = FastNeutronFluxAux
    variable = fast_neutron_flux
    block = 1
    rod_ave_lin_pow = linear_heat_rate_profile                  #VERA_MODIFIABLE as auxkernels_fast_neutron_flux_rod_ave_lin_pow
    axial_power_profile = axial_peaking_factors                 #VERA_MODIFIABLE as
auxkernels_fast_neutron_flux_axial_power_profile
    # CHECK: ratio of LHGR and fast neutron flux?
    factor = 1.6e12 # (n/m2-s per W/m)                          #VERA_MODIFIABLE as auxkernels_fast_neutron_flux_factor
```

```
  execute_on = timestep_begin
[../]

[./fast_neutron_fluence]
  type = FastNeutronFluenceAux
  variable = fast_neutron_fluence
  block = 1
  fast_neutron_flux = fast_neutron_flux
  execute_on = timestep_begin
[../]

[./fuel_porosity]
  type = PorosityAuxUO2
  block = 3
  variable = porosity
  execute_on = timestep_begin                          #VERA_MODIFIABLE as auxkernels_fuel_porosity_execute_on
[../]

[./grain_radius]
  type = GrainRadiusAux
  block = 3
  variable = grain_radius
  temp = temp
  execute_on = timestep_begin                          #VERA_MODIFIABLE as auxkernels_grain_radius_execute_on
[../]

[./gap_conductance]
  type = MaterialRealAux
  property = gap_conductance
  variable = gap_conductivity
  boundary = 10  # fuel radially-outer surface
[../]

[./fuel_conductance]
  type = MaterialRealAux
  property = thermal_conductivity
  variable = fuel_conductivity
  block = 3
[../]

# ================================================= #
# General Mechanics Types
# ================================================= #

[./hydrostatic_stress]
  block = 3
  type = MaterialTensorAux
  tensor = stress
  variable = hydrostatic_stress
  quantity = hydrostatic
  execute_on = timestep_begin                          #VERA_MODIFIABLE as auxkernels_hydrostatic_stress_execute_on
[../]

[./hoop_stress]
  type = MaterialTensorAux
  tensor = stress
  variable = hoop_stress
  index = 2
  execute_on = timestep_begin                          #VERA_MODIFIABLE as auxkernels_hoop_stress_execute_on
[../]

[./radial_stress]
  type = MaterialTensorAux
  tensor = stress
  variable = radial_stress
  index = 0
  execute_on = timestep_begin                          #VERA_MODIFIABLE as auxkernels_radial_stress_execute_on
[../]

[./axial_stress]
  type = MaterialTensorAux
  tensor = stress
  variable = axial_stress
  index = 1
  execute_on = timestep_begin                          #VERA_MODIFIABLE as auxkernels_axial_stress_execute_on
[../]

[./radial]
  type = MaterialTensorAux
  tensor = stress
  variable = radial_strain
  quantity = radial
  # what is this and why are all three the same.
  point1 = '0, 0, 0'
  point2 = '0, 1, 0'
  execute_on = timestep_begin                          #VERA_MODIFIABLE as auxkernels_radial_execute_on
[../]

[./axial]
  type = MaterialTensorAux
  tensor = stress
  variable = axial_strain
  quantity = axial
  # what is this and why are all three the same.
  point1 = '0, 0, 0'
  point2 = '0, 1, 0'
  execute_on = timestep_begin                          #VERA_MODIFIABLE as auxkernels_axial_execute_on
[../]

[./hoop]
```

```
  type = MaterialTensorAux
  tensor = stress
  variable = hoop_strain
  quantity = hoop
  # what is this and why are all three the same.
  point1 = '0, 0, 0'
  point2 = '0, 1, 0'
  execute_on = timestep_begin                                  #VERA_MODIFIABLE as auxkernels_hoop_execute_on
[../]

[./elastic_radial_strain]
  type = MaterialTensorAux
  tensor = elastic_strain
  variable = radial_strain
  index = 0
  execute_on = timestep_end
[../]

[./elastic_axial_strain]
  type = MaterialTensorAux
  tensor = elastic_strain
  variable = axial_strain
  index = 1
  execute_on = timestep_end
[../]

[./elastic_hoop_strain]
  type = MaterialTensorAux
  tensor = elastic_strain
  variable = hoop_strain
  index = 2
  execute_on = timestep_end
[../]

[./vonmises]
  type = MaterialTensorAux
  tensor = stress
  variable = vonmises
  quantity = vonmises
  execute_on = timestep_begin                                  #VERA_MODIFIABLE as auxkernels_vonmises_execute_on
[../]

[./creep_strain_mag]
  type = MaterialTensorAux
  tensor = creep_strain
  variable = creep_strain_mag
  quantity = plasticstrainmag
  execute_on = timestep_begin                                  #VERA_MODIFIABLE as auxkernels_creep_strain_mag_execute_on
  # block = 1
[../]

[./creep_strain_radial]
  type = MaterialTensorAux
  tensor = creep_strain
  variable = creep_strain_radial
  index = 0
  execute_on = timestep_end                                    #VERA_MODIFIABLE as auxkernels_creep_strain_radial_execute_on
[../]

[./creep_strain_axial]
  type = MaterialTensorAux
  tensor = creep_strain
  variable = creep_strain_axial
  index = 1
  execute_on = timestep_end                                    #VERA_MODIFIABLE as auxkernels_creep_strain_axial_execute_on
[../]

[./creep_strain_hoop]
  type = MaterialTensorAux
  tensor = creep_strain
  variable = creep_strain_radial
  index = 2
  execute_on = timestep_begin
[../]

[./conductance]  #>> NOT IN OTHER TEMPLATE
  type = MaterialRealAux
  property = gap_conductance
  variable = gap_conductivity
  boundary = 10
[../]

# ================================================= #
# Other General Types
# ================================================= #

[./axial_burnup]
  type = SpatialUserObjectAux
  block = 3
  variable = axial_burnup
  execute_on = timestep_begin
  user_object = axial_burnup
[../]

[./axial_temperature]
  type = SpatialUserObjectAux
  block = 3
  variable = axial_temperature
  execute_on = timestep_begin
  user_object = axial_temperature
```

```
  [../]

  [./gap_distance]
    type = PenetrationAux
    variable = gap_distance
    boundary = 10
    paired_boundary = 5
  [../]
[]


# ================================================== #
# Mechanical and Thermal Contact
# ================================================== #
[Contact]
  [./pellet_clad_mechanical]
    master = 5                                                  # clad
    slave = 10                                                  # fuel
    disp_x = disp_x
    disp_y = disp_y
    penalty = 1e8                                               #VERA_MODIFIABLE as pellet_clad_mechanical_penalty
    tangential_tolerance = 1e-4                                 #VERA_MODIFIABLE as pellet_clad_mechanical_tangential_tolerance
(omitted otherwise)
    normal_smoothing_distance = 0.1                             #VERA_MODIFIABLE as pellet_clad_mechanical_normal_smoothing_distance
    model = frictionless                                        #VERA_MODIFIABLE as pellet_clad_mechanical_model
    formulation = penalty                                       #VERA_MODIFIABLE as pellet_clad_mechanical_formulation
    system = Constraint                                         #VERA_MODIFIABLE as pellet_clad_mechanical_system (omitted otherwise)
 [../]
[]

[ThermalContact]

  [./thermal_contact]
    type = GapHeatTransferLWR
    variable = temp
    master = 5                                                  # clad
    slave = 10                                                  # fuel
    roughness_coef = 3.200000e+00                               #VERA_MODIFIABLE as thermalcontact_roughness_coef
    roughness_fuel = 2.000000e-06                               #VERA_MODIFIABLE as thermalcontact_roughness_fuel
    roughness_clad = 1.000000e-06                               #VERA_MODIFIABLE as thermalcontact_roughness_clad
    jump_distance_model = KENNARD
    plenum_pressure = plenum_pressure
    contact_pressure = contact_pressure
    initial_moles = initial_moles
    initial_gas_fractions = '1.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
0.0000e+00' #VERA_MODIFIABLE as thermalcontact_initial_gas_fraction
    gas_released = fis_gas_released
    tangential_tolerance = 1.0e-6                               #VERA_MODIFIABLE as thermalcontact_tangential_tolerance
    normal_smoothing_distance = 0.1
    quadrature = false                                          #VERA_MODIFIABLE as thermalcontact_quadrature
    order = FIRST                                               #VERA_MODIFIABLE as thermalcontact_order
  [../]
[]

[BCs]

  [./no_x_all]
    type = DirichletBC
    variable = disp_x
    boundary = 12
    value = 0.0
  [../]

  [./no_y_fuel_bottom]
    type = DirichletBC
    variable = disp_y
    boundary = 1020                                             #VERA_MODIFIABLE as bcs_no_y_fuel_bottom_boundary
    value = 0.0
  [../]

  [./no_y_clad_bottom]
    type = DirichletBC
    variable = disp_y
    boundary = 1
    value = 0.0
  [../]

  [./clad_coolant_surface]
    type = ConvectiveFluxFunction                               #CASL_MOOSE_MK_AuxVarDirichletBC for TIAMAT and
ConvectiveFluxFunction for BISON_CASL
    coefficient = 1e5                                           #BISON_CASL
    coefficient_function = 1.0e6                                #BISON_CASL
    T_infinity = coolant_temperature                            #BISON_CASL
    aux_var = 'casl_clad_surface_temperature'                   #TIAMAT
    boundary = '2'
    variable = temp
  [../]

  # CHECK: this seems arbitrary, why?
  [./convective_clad_surface_bottom] # apply convective boundary to clad outer
    type = ConvectiveFluxBC
    boundary = '1 3'
    variable = temp
    rate = 38200.0    #convection coefficient (h)
    initial = 565.0
    final = 585.0
    duration = 1.0e4  #duration of initial power ramp
  [../]

  [./Pressure]
```

```
    [./coolantPressure]
      boundary = '1 2 3'
      factor = 15.5132e6                                    #VERA_DEFINED
      function = coolant_pressure_ramp                      #VERA_MODIFIABLE as bcs_pressure_coolantpressue_function
    [../]
  [../]

  [./PlenumPressure]
    # apply plenum pressure on clad inner walls and pellet surfaces
    [./plenumPressure]
      boundary = '9'                                        #VERA_MODIFIABLE as bcs_plenumpressure_plenumpressure_boundary
      initial_pressure = 2.62e6                             #VERA_MODIFIABLE as
bcs_plenumpressure_plenumpressure_initial_pressure
      startup_time = 0                                      #VERA_MODIFIABLE as bcs_plenumpressure_plenumpressure_startup_time
      R = 8.314462
      output_initial_moles = initial_moles   # coupling to post processor to get initial fill gas mass
      temperature = ave_temp_interior        # coupling to post processor to get temperature for plenum gas
      volume = gas_volume                    # coupling to post processor to get gas volume
      material_input = fis_gas_released      # coupling to post processor to get fission gas added
      output = plenum_pressure               # coupling to post processor to output plenum/gap pressure
    [../]

  [../]

  #>> CLAD_SURFACE
  [./right] #>>PURGED FROM STANDALONE CASES
    type = CASL_MOOSE_MK_AuxVarDirichletBC
    variable = temp
    boundary = 2
    aux_var = 'casl_clad_surface_temperature'
  [../]
[]

# ================================================== #
# Specification of Material Properties
# ================================================== #
[Materials]

  [./fuel_thermal]
    type = UO2Therm
    block = 3
    temp = temp
    burnup = burnup
  [../]

  [./fuel_swelling]
    type = VSwellingUO2
    block = 3
    temp = temp
    burnup = burnup
    total_densification = 1.00000e-02                       #VERA_MODIFIABLE as materials_fuel_swelling_total_densification
#    gas_swelling_type = Sifgrs
  [../]

  [./fuel_mech]
    type = MechUO2
    block = 3
    temp = temp
    burnup = burnup
    fission_rate =                                          #casl_fission_rate for TIAMAT, fission_rate for BISON_CASL
    youngs_modulus = 2.e11
    poissons_ratio = .345
    thermal_expansion = 10e-6
    grain_radius = 5.240000e-06                             #VERA_MODIFIABLE as materials_fuel_mech_grain_radius
    oxy_to_metal_ratio = 2.0
    max_its =                                               #VERA_MODIFIABLE as materials_fuel_mech_max_its (omitted otherwise)
    output_iteration_info = false                           #VERA_MODIFIABLE as materials_fuel_mech_output_iteration_info
    model_thermal_expansion = true
    model_swelling = true                                   #VERA_MODIFIABLE as materials_fuel_mech_model_swelling
    name_swelling_model = fuel_swelling                     #VERA_MODIFIABLE as materials_fuel_mech_name_swelling_model (omitted
otherwise)
    model_relocation = true
    name_relocation_model = fuel_relocation
    model_creep = true                                      #VERA_MODIFIABLE as materials_fuel_mech_model_creep
    matpro_youngs_modulus = true                            #VERA_MODIFIABLE as materials_fuel_mech_matpro_youngs_modulus
    matpro_poissons_ratio = true
    volumetric_strain = deltav_v0_swe
  [../]

  [./fuel_relocation]
    type = RelocationUO2
    model = ESCORE
    block = 3
    burnup = burnup
    q = q                                                   #VERA_MODIFIABLE as materials_fuel_relocation_q
    diameter = 0.008192                                     #VERA_DEFINED
    gap = 0.000168                                          #VERA_DEFINED
    relocation_activation1 = 16404                          #VERA_MODIFIABLE as materials_fuel_relocation_relocation_activation1
    # Average burnup at which fuel comes into contact with clad at 25kW/m
    # NOTE: this should likely require rerunning BISON after a more accurate value has been obtained
    burnup_relocation_stop = 0.0315                         #VERA_MODIFIABLE as materials_fuel_relocation_burnup_relocation_stop
  [../]

  [./clad_thermal]
    type = ThermalZry
    block = 1
    temp = temp
  [../]

  [./clad_solid_mechanics]
```

```
    type = MechZryPP
    block = 1
    temp = temp
    fast_neutron_flux = fast_neutron_flux
    fast_neutron_fluence = fast_neutron_fluence
    youngs_modulus = 7.5e10
    poissons_ratio = 0.3
    max_its = 50
    material_type = 0
    absolute_tolerance = 1e-12                              #VERA_MODIFIABLE as materials_clad_solid_mechanics_absolute_tolerance
    relative_tolerance = 1e-5                               #VERA_MODIFIABLE as materials_clad_solid_mechanics_relative_tolerance
(omitted if BISON_CASL)
    stress_free_temperature = 297                           #VERA_MODIFIABLE as
materials_clad_solid_mechanics_stress_free_temperature (omitted if BISON_CASL)
    model_primary_creep = true                              #VERA_MODIFIABLE as
materials_clad_solid_mechanics_model_primary_creep (omitted otherwise)
    model_thermal_creep = true                              #VERA_MODIFIABLE as
materials_clad_solid_mechanics_model_thermal_creep (omitted otherwise)
    model_irradiation_creep = true                          #VERA_MODIFIABLE as
materials_clad_solid_mechanics_model_irradiation_creep (omitted otherwise)
    model_irradiation_growth = true                         #VERA_MODIFIABLE as
materials_clad_solid_mechanics_model_irradiation_growth
    model_thermal_expansion = true                          #VERA_MODIFIABLE as
materials_clad_solid_mechanics_model_thermal_expansion
    output_iteration_info = false
    model_elastic_modulus = false                           #VERA_MODIFIABLE as
materials_clad_solid_mechanics_model_elastic_modulus
    cold_work_factor = 4.812700e-02                         #VERA_MODIFIABLE as materials_clad_solid_mechanics_cold_work_factor
(omitted otherwise)
    oxygen_concentration = 0.0                              #VERA_MODIFIABLE as
materials_clad_solid_mechanics_oxygen_concentration (omitted otherwise)
  [../]

  [./fission_gas_release]
    type = Sifgrs
    axial_power_profile = axial_peaking_factors             #VERA_MODIFIABLE as materials_fission_gas_release_axial_power_profile
(omitted otherwise)
    block = 3
    burnup = burnup
    end_densification_burnup =                              #VERA_MODIFIABLE as
materials_fission_gas_release_end_densification_burnup (omitted otherwise)
    file_name =                                             #VERA_MODIFIABLE as materials_fission_gas_release_file_name (omitted
otherwise)
    fission_rate = casl_fission_rate                        #casl_fission_rate with TIAMAT, fission rate with BISON_CASL
    grain_radius = grain_radius                             #VERA_MODIFIABLE as materials_fission_gas_release_grain_radius
    hydrostatic_stress =                                    #VERA_MODIFIABLE as materials_fission_gas_release_hydrostatic_stress
(omitted otherwise)
    initial_grain_radius = 5.240000e-06                     #VERA_MODIFIABLE as
materials_fission_gas_release_initial_grain_radius
    pellet_brittle_zone =                                   #VERA_MODIFIABLE as materials_fission_gas_release_pellet_brittle_zone
(omitted otherwise)
    pellet_id =                                             #VERA_MODIFIABLE as materials_fission_gas_release_pellet_id (omitted
otherwise)
    rod_ave_lin_pow =                                       #VERA_MODIFIABLE as materials_fission_gas_release_rod_ave_lin_pow
(omitted otherwise)
    temp = temp
    total_densification =                                   #VERA_MODIFIABLE as materials_fission_gas_release_total_densification
(omitted otherwise)
    compute_swelling = true
  [../]

  [./clad_density]
    type = Density
    block = 1
    density = 6551.0                                        #VERA_DEFINED or VERA_MODIFIABLE as materials_clad_density_density
  [../]

  [./fuel_density]
    type = Density
    block = 3
  [../]
[]

# ================================================== #
# User Objects for Output Processing
# ================================================== #
[UserObjects]
  [./pbz]
    type = PelletBrittleZone
    block = 3
    pellet_id = pellet_id
    temp = temp
    pellet_radius =                                         #VERA_DEFINED
    number_pellets = 1
    execute_on = timestep_begin                             #VERA_MODIFIABLE as userobjects_pbz_execute_on
  [../]

  [./averagefissionrate]
    type = LayeredAverage
    block = 3
    variable = fission_rate
    direction = y
    num_layers = 49
  [../]

  [./average_temp]
    type = LayeredAverage
    block = 3
    variable = temp
    direction = y
```

```
    num_layers = 49
  [../]

  [./averagebu]
    type = LayeredAverage
    block = 3
    variable = burnup
    direction = y
    num_layers = 49
  [../]

  [./casl_average_fission_rate]
    variable = casl_fission_rate                    #casl_fission_rate with TIAMAT, fission rate with BISON_CASL
    type = LayeredAverage
    block = 3
    direction = y
    bounds =                                        #VERA_DEFINED
  [../]

  [./surface_temp]
    type = LayeredSideAverage
    boundary = 2
    variable = temp
    direction = y
    bounds =                                        #VERA_DEFINED
    use_displaced_mesh = 0
  [../]

  [./axial_temperature]
    type = LayeredAverage
    block = 3
    variable = temp
    direction = y
    bounds =                                        #VERA_DEFINED
  [../]

  [./axial_burnup]
    type = LayeredAverage
    block = 3
    variable = burnup
    direction = y
    bounds =                                        #VERA_DEFINED
  [../]

  [./integral_temperature]
    type = LayeredAverage
    block = 3
    variable = temp
    direction = y
    num_layers = 1
  [../]

  [./integral_burnup]
    type = LayeredAverage
    block = 3
    variable = burnup
    direction = y
    num_layers = 1
  [../]

  [./average]
    type = LayeredAverage
    block = 3
    variable = temp
    direction = y
    bounds =                                        #VERA_DEFINED
  [../]

  [./axial_surface_temperature]
    type = LayeredSideAverage
    boundary = 2
    variable = temp
    direction = y
    bounds =                                        #VERA_DEFINED
    use_displaced_mesh = 0
  [../]

  [./rod_avg_fast_fluence]
    type = LayeredSideAverage
    boundary = 2
    variable = fast_neutron_fluence
    direction = y
    num_layers = 1
    use_displaced_mesh = 0
  [../]

  [./casl_clad_surface_heat_flux]
    type = LayeredSideFluxAverage
    variable = temp
    boundary = 2
    direction = y
    bounds =                                        #VERA_DEFINED
    diffusivity = thermal_conductivity
  [../]
[]

[Dampers]
  [./limitT]
    type = MaxIncrement
    max_increment = 50.0                            #VERA_MODIFIABLE as dampers_limitt_max_increment
```

```
    variable = temp
  [../]
[]

# ================================================== #
# Solver Options
# ================================================== #
[Executioner]
  type = Transient
  solve_type = 'PJFNK'

  petsc_options = '-pc_type_asm'                                  #VERA_MODIFIABLE as executioner_petsc_options (omitted otherwise)
  petsc_options_iname = '-ksp_gmres_restart -pc_type -pc_hypre_type -pc_hypre_boomeramg_max_iter'  #VERA_MODIFIABLE as
executioner_petsc_options_iname
  petsc_options_value = '201               hypre    boomeramg      4'                              #VERA_MODIFIABLE as
executioner_petsc_options_value
  line_search = 'none'                                            #VERA_MODIFIABLE as executioner_line_search
  verbose = false                                                 #VERA_MODIFIABLE as executioner_verbose

  # controls for linear iterations
  l_max_its = 100                                                 #VERA_MODIFIABLE as executioner_l_max_its
  l_tol = 8e-3                                                    #VERA_MODIFIABLE as executioner_l_tol

  # controls for nonlinear iterations  #>>MODIFIABLE???
  nl_max_its = 20                                                 #VERA_MODIFIABLE as executioner_nl_max_its
  nl_rel_tol = 1e-4                                               #VERA_MODIFIABLE as executioner_nl_rel_tol
  nl_abs_tol = 1e-10                                              #VERA_MODIFIABLE as executioner_nl_abs_tol

  # ================================================== #
  # Time Step Control - needs work.
  # ================================================== #
  start_time = -100                                               #VERA_MODIFIABLE as executioner_start_time
  end_time   = 6e+07                                              #VERA_PREPARED as executioner_end_time
  num_steps  = 1                                                  #VERA_PREPARED as executioner_num_steps (omitted if BISON_CASL)
  dtmax      =                                                    #VERA_MODIFIABLE as executioner_dtmax (omitted otherwise)
  dtmin      =                                                    #VERA_MODIFIABLE as executioner_dtmin (omitted otherwise)
  [./TimeStepper]
    type = IterationAdaptiveDT                                    #IterationAdaptiveDT with TIAMAT, FunctionDT with BISON_CASL
    time_t = '0     1.0e4 53200 1.0e5'                            #BISON_CASL as executioner_timestepper_time_t
    time_dt = '1.0e3 1.0e3 1.0e3 1.0e5'                           #BISON_CASL as executioner_timestepper_time_dt
    dt = 1.0e1                                                    #TIAMAT or as executioner_timestepper_dt
    optimal_iterations = 8                                        #TIAMAT or as executioner_optimal_iterations
    linear_iteration_ratio = 100                                  #TIAMAT or as executioner_timestepper_linear_iteration_ratio
  [../]

# # Quadrature block needs to be activated if you are using higher order elements (i.e. QUAD8 in 2D-RZ)
#   [./Quadrature]
#     order = FIFTH                                                 #VERA_MODIFIABLE as executioner_quadrature_order
#   [../]

[]

# ================================================== #
# Postprocessors: for internal averages and output.
# ================================================== #
[Postprocessors]

  # ================================================== #
  # Required for Fission Gas Release Models
  # ================================================== #

  [./ave_temp_interior]
    # used to compute temperature of plenum
    type = SideAverageValue
    #>> CHECK: is '9' the top of fuel?
    boundary = 9
    variable = temp
    outputs = 'exodus'
  [../]

  [./fis_gas_released]
    type = ElementIntegralFisGasReleasedSifgrs
    variable = temp
    block = 3
    outputs = 'exodus'
  [../]

  [./gas_volume]
    type = InternalVolume
    boundary = 9
    outputs = 'exodus'
  [../]

  [./fis_gas_grain]
    type = ElementIntegralFisGasGrainSifgrs
    variable = temp
    block = 3
    outputs = exodus
  [../]

  [./fis_gas_boundary]
    type = ElementIntegralFisGasBoundarySifgrs
    variable = temp
    block = 3
    outputs = exodus
  [../]

  # ================================================== #
  # Output-Only: Fission Gas Release Data
  # ================================================== #
```

```
[./average_grain_radius]
  type = ElementAverageValue
  block = 3
  outputs = 'exodus'
  variable = grain_radius
[../]

[./clad_inner_vol]
  type = InternalVolume
  boundary = 7
  outputs = 'exodus'
[../]

[./pellet_volume]
  type = InternalVolume
  boundary = 8
  outputs = 'exodus'
[../]

[./avg_clad_temp]
  type = SideAverageValue
  boundary = 7
  variable = temp
  outputs = 'exodus'
[../]

[./fis_gas_produced]
  type = ElementIntegralFisGasGeneratedSifgrs
  variable = temp
  block = 3
  outputs = 'exodus'
[../]

# ================================================== #
# Output-Only: Gap-Conductance Related
# ================================================== #

[./flux_from_clad]
  type = SideFluxIntegral
  variable = temp
  boundary = 5
  diffusivity = thermal_conductivity
  outputs = 'exodus'
[../]

[./flux_from_fuel]
  type = SideFluxIntegral
  variable = temp
  boundary = 10
  diffusivity = thermal_conductivity
  outputs = 'exodus'
[../]

# ================================================== #
# Output-Only: Solver Related
# ================================================== #

[./_dt]
  type = TimestepSize
  outputs = 'exodus'
[../]

[./nonlinear_its]
  type = NumNonlinearIterations
  outputs = 'exodus'
[../]

[./linear_its] # number of nonlinear iterations at each timestep
  type = NumLinearIterations
[../]

# ================================================== #
# Output-Only: Power
# ================================================== #

[./rod_average_fuel_temp]
   type =  ElementAverageValue
   block = 3
   variable = temp
[../]

[./rod_total_power]
  type = ElementIntegralPower
  variable = temp
  fission_rate = casl_fission_rate                          #casl_fission_rate with TIAMAT, fission_rate with BISON_CASL
  block = 3
[../]

[./rod_input_power]
  # Watts
  type = PlotFunction
  function = linear_heat_rate_profile
  scale_factor = 3.6576 # rod height                        #VERA_DEFINED
[../]

[./average_fission_rate]
  # CHECK: Units?
  type = AverageFissionRate
  rod_ave_lin_pow = linear_heat_rate_profile
```

```
    outputs = 'exodus'
  [../]

  # ================================================== #
  # Tiamat: Conservation of Energy (#TIAMAT)
  # ================================================== #

  [./volume_casl_fission_rate] #>> PURGED FROM STANDALONE CASES
    type = ElementIntegralVariablePostprocessor
    variable = 'casl_fission_rate'
    execute_on = custom
  [../]

  [./casl_max_rod_temperature] #>> PURGED FROM STANDALONE CASES
    # for global convergence checking
    type = NodalMaxValue
    variable = temp
    block = 3
  [../]

  # ================================================== #
  # PCI Indicators
  # ================================================== #
  [./clad_hoop_stress_max]
    type = ElementExtremeValue
    value_type = MAX
    variable = hoop_stress
    block = 1
    outputs = 'csv'
  [../]

  [./clad_hoop_stress_min]
    type = ElementExtremeValue
    value_type = MIN
    variable = hoop_stress
    block = 1
    outputs = 'csv'
  [../]

  [./fuel_centerline_temp]
    type = NodalExtremeValue
    variable = temp
    boundary = 12
    outputs = 'csv'
  [../]

  [./gap_distance]
    type = NodalExtremeValue
    value_type = MAX
    variable = gap_distance
    outputs = 'csv'
  [../]

  [./average_clad_temp]
    type =  ElementAverageValue
     block = 1
     variable = temp
  [../]

  [./max_clad_temp]
    type = ElementExtremeValue
    value_type = MAX
    block = 1
    variable = temp
  [../]

  [./min_clad_temp]
    type = ElementExtremeValue
    value_type = MIN
    block = 1
    variable = temp
  [../]

[]

# ================================================== #
# Location and format of output
# ================================================== #
[Outputs]
  output_initial = true
  csv = true
  exodus = true
  file_base =                                          #BISON_CASL as outputs_file_base (omitted otherwise)
  interval = 1

  [./console]
    type = Console
    perf_log = true
    max_rows = 25
    output_linear = true
    # output_linear = false
  [../]

#  [./CSV]
#    type = CSV
#    initial = false
#    output_postprocessors_on = false
#  [../]
[]
```

```
# ================================================== #
# Debug Output
# ================================================== #
[Debug]
#  show_var_residual_norms = true
#  show_var_residual = 'disp_x disp_y temp'
[]

# ================================================== #
# Use of this input template
# ================================================== #
#
# This is a VERA input template for BISON-CASL.
#
# A VERA input file (input.vera) can be used to produce
# a VERA xml input file (input.xml) using the VERAIn
# script:
#
# > VERA/VERAIn/verain/scripts/react2xml.pl input.vera input.xml
#
# The xml2moose preprocessors can be used to produce
# BISON-CASL input files using the xml input (input.xml)
# and a template (defined in input.vera):
#
# > BUILD/MOOSE/xml2moose -c input
#
# Currently, xml2moose generates an input for use with
# Tiamat, which assumes that power-history comes from
# the Neutronics code, clad surface temperature comes
# from Cobra-TF, the average fuel temperature on the
# axial_edit_bounds computed for passing to the
# Neutronics code, and the average heat flux on the
# clad outer surface at the axial_edit_bounds must be
# computed for passing to Cobra-TF.
#
# In the near future, xml2moose will have additional
# options to modify this input to support:
#
# - Stand-alone BISON-CASL with power history, shape, and clad temperatures from a VERA-CS HDF5 output file
# - Stand-alone BISON-CASL with unspecified power history, shape, and clad temperature BCs
# - Stand-alone BISON-CASL with unspecified power history, shape, and bulk-coolant temperature BCs
# - Stand-alone BISON-CASL with unspecified power history, shape, and coolant channel BCs
# ================================================== #

# ================================================== #
# Assumptions embedded in this input template
# ================================================== #
# 1. There is one fuel type:
#   -- UO2
#   -- single density    (VERA_DEFINED)
#   -- single enrichment (VERA_DEFINED)
# 2. There is one clad type:
#   -- Zircalloy
#   -- single density    (VERA_DEFINED)
#   -- single enrichment (VERA_DEFINED)
# 3. The fuel is modeled as a single smeared mesh
#   -- The axial_edit_bounds in the VERA input file define
#      the minimum axial mesh and BOUNDS for post-processing
#   -- The internal mesh generator is used, but the
#      axial and radial mesh in the fuel and cladding can be
#      specified by the user (VERA_MODIFIABLE)
# 4. The current template is limited to 2D-RZ PWR standard
#    fuel rod models. Future work will be necessary to
#    include annular fuel, IFBA coating, and BWR rods.
#
# ================================================== #
```