Plan of Record
Period 2: January – June 2011
Volume 1
EPRI Test Stand Report
Results and Feedback from the EPRI Test Stand

Brenden Mervin, EPRI
Martin Pytel, EPRI
Dennis Hussey, EPRI
Stephen Hess, EPRI
1 August 2014
Please complete sections appropriate for this record.

REVISION LOG

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Affected Pages</th>
<th>Revision Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>8/1/2014</td>
<td>All</td>
<td>Original Report</td>
</tr>
<tr>
<td>1</td>
<td>12/17/2014</td>
<td>REVISION LOG</td>
<td>Changed control entries from TBD to N/A to approve for public release</td>
</tr>
</tbody>
</table>

Document pages that are:

- Export Controlled ____ None____________________________________________________
- IP/Proprietary/NDA Controlled ____ None__________________________________________
- Sensitive Controlled ____ None__________________________________________________

Requested Distribution:

To:   Doug Kothe, CASL Director  
      Doug Burns, CASL Deputy Director  
      Paul Turinsky, CASL Chief Scientist  
      Jeff Banta, CASL Program Manager

Copy: Zeses Karoutas – AMA FA Lead  
      Rose Montgomery – AMA FA Deputy Lead  
      Stephen Hess – AMA FA Deputy Lead  
      Chris Stanek – MPO FA Lead  
      Brian Wirth – MPO FA Deputy Lead  
      Robert Montgomery – Peregrine Principal Investigator  
      Jess Gehin – PHI FA Lead  
      Matt Sieger – CASL Quality Manager  
      Dennis Hussey – IC Chairman
EXECUTIVE SUMMARY

This report details the Electric Power Research Institute’s (EPRI) Test Stand experience with the Peregrine fuel performance code, a component of the Virtual Environment for Reactor Applications (VERA). The purpose of this report is to provide an assessment of the VERA software suite, specifically the fuel performance component, Peregrine. This includes providing both technical and administrative feedback to the Consortium for Advanced Simulation of Light Water Reactors (CASL) in order to aid in the development and deployment of VERA software and components. As a CASL industry partner, EPRI feedback will also discuss VERA’s applicability to modern industry issues in commercial power plant operation and design. Some of the feedback topics relating to the assessment of Peregrine include discussions on the simulation capabilities and its ability to perform pellet-cladding interaction (PCI) analysis, compatibility with industry computing platforms, usability, and performance relative to existing industry tools.

The objective of this Test Stand is to evaluate Peregrine as a state-of-the-art fuel performance code by guiding it through a series of fuel performance progression problems. These progression problems are performed using two-dimensional (2D) R-Z axisymmetric models and focus on examining the thermal and mechanical responses of the fuel and cladding to an imposed axially-varying power history. The progression begins with a constant axial power profile imposed during a single cycle ramp up to power followed by steady-state operation for a short length test rod and concludes with the most complex case studied by the Test Stand: an axially-varying power history containing a first cycle ramp to full power steady-state operation followed by a shutdown and a second-cycle ramp to full power. The methods and procedures used for this final progression are also used to examine a generic pressurized water reactor (PWR) fuel rod as well as a Watts Bar Unit 1 fuel rod.

The ultimate goal of this Test Stand application is to evaluate Peregrine with respect to its ability to perform pellet-cladding interaction (PCI) analyses. To accomplish this, an extensive set of simulations were executed in order to perform a thorough assessment of the Peregrine code. These simulations used a variety of different models and power histories to evaluate various aspects of the code and its ability to perform analyses similar to those typically performed by nuclear power plant operators using currently available fuel performance codes. In the absence of measured data, Peregrine is run alongside the Falcon fuel performance code, and the resulting thermal and mechanical trends as a function of time are compared for parameters such as temperature, displacement, and hoop stress for select locations. As an illustrative example, Figure 1 displays the hoop stress comparison for a generic PWR rod model during the down power following cycle 1 steady-state operation as well as the second cycle restart (progression problem 6). In comparison with the Falcon analysis, notable observations from Figure 1 include:

- Peregrine predicts less stress relaxation during the down power as well as the first step in power during the second-cycle restart,
- Peregrine predicts more stress relaxation during the final hold period after the second cycle power ramp is complete,
- Peregrine predicts less stress buildup during the first step increase of the second-cycle power ramp,

Reactor data for this final case were used with permission from Westinghouse and the Tennessee Valley Authority (TVA).
• Peregrine predicts more stress buildup during the second step increase of the second-cycle power ramp, and
• The peak inside cladding surface hoop stress predictions by the two codes are within 0.5% of each other.

Figure 1. Cladding Inside Surface Hoop Stress Taken at a Point along the Axial Centerline of the Active Fuel Region for Progression Problem 6

The results from the seven progression problems were used to assess the position of Peregrine relative to industry standard codes. Note that 15,000+ core hours were used on EPRI’s high-performance computing (HPC) cluster Phoebe to perform the simulations for this Test Stand. The primary conclusions derived from this Test Stand are:

• Differences are observed in the magnitude of the temperature, displacement, and hoop stress estimates by Peregrine and Falcon for similar input cases.
• General agreement between Falcon and Peregrine is observed in the temperature, displacement, and hoop stress trends both spatially and temporally.
• The post-processing and analysis tools developed by the EPRI Team for this Test Stand automated many of the tasks required to rapidly compare results among different versions of Peregrine.
• Support from CASL and the Peregrine development team was essential to the success of the Test Stand.

Noting that the Peregrine code is still under development, these initial results are promising and suggest with a good degree of confidence that Peregrine will provide a significantly advanced fuel performance analysis capability within VERA. Because of the level of active development, work is required to understand the reasons for the observed differences and to conduct additional verification and validation benchmarking studies as the code matures.
In addition to technical results and conclusions, feedback is also provided in Appendix B on various subjects including installation, testing, input file construction, output file post-processing, execution, and documentation. Select key recommendations include:

- Develop verification and validation testing procedures for Peregrine and integrate these test suites into the continuous development process for VERA.

- Prepare additional documentation detailing input file specifications, available models/options, and best practices for fuel performance modeling and simulation using Peregrine. Major focus areas include meshing, time step selection, and model selection.

- Improve the workflow between repositories at Idaho National Laboratory (INL), Pacific Northwest National Laboratory (PNNL), and Oak Ridge National Laboratory (ORNL) to facilitate ongoing software development during a Test Stand.
ACKNOWLEDGEMENTS

The following individuals and organizations provided valuable contributions to this report. Their expert advice and active support regarding the research performed during this Test Stand is greatly appreciated. The EPRI-CASL Test Stand Team expresses gratitude to Westinghouse for contributing fuel rod design and ANC core follow fuel rod axial shape power history data for Watts Bar Unit 1. These data are instrumental for the advancement of Peregrine development. The EPRI-CASL Test Stand Team expresses gratitude to TVA for allowing this data to be used to exercise and extend Peregrine capabilities for this Test Stand.

ANATECH Corp.
Wenfeng Liu

Electric Power Research Institute
Heather Feldman

Idaho National Laboratory
Jason Hales, Danielle Perez, Ben Spencer, Richard Williamson,

Oak Ridge National Laboratory
Mark Baird, Ross Bartlett, Roger Pawlowski, Matt Sieger,

Pacific Northwest National Laboratory
Robert Montgomery, Dion Sunderland

Tennessee Valley Authority
Rose Montgomery

University of Tennessee
Nathan Capps, Brian Wirth

Westinghouse
Zeses Karoutas, David Mitchell
# CONTENTS

1 INTRODUCTION ...........................................................................................................................1

2 TEST STAND DEPLOYMENT ..........................................................................................................3
  2.1 Phoebe .................................................................................................................................3
  2.2 VERA ...................................................................................................................................4
  2.3 Peregrine ..............................................................................................................................6

3 METHODOLOGY ........................................................................................................................11
  3.1 Analysis Methodology .........................................................................................................11
    3.1.1 Progression Problems ..................................................................................................11
    3.1.2 Peregrine/Falcon Comparisons ................................................................................15
  3.2 Modeling Options ................................................................................................................16
    3.2.1 Relocation ....................................................................................................................17
    3.2.2 Smeared Cracking .......................................................................................................17
    3.2.3 Fission Gas Release .....................................................................................................18
    3.2.4 Contact and Friction ....................................................................................................19
  3.3 Boundary Conditions ..........................................................................................................19
    3.3.1 Power History ..............................................................................................................20
    3.3.2 Coolant Channel Model ..............................................................................................21
  3.4 Solution Control ...................................................................................................................21
    3.4.1 Meshing .......................................................................................................................21
    3.4.2 Parallelism ....................................................................................................................23
    3.4.3 Time Stepping .............................................................................................................23
    3.4.4 Post-Processing ..........................................................................................................23
  3.5 Automation Tools ..................................................................................................................26
    3.5.1 Falcon to Peregrine Scripts ........................................................................................26
    3.5.2 Code-to-Code Comparisons .......................................................................................26

4 RESULTS ......................................................................................................................................27
  4.1 Progression Problem 1: Super-Ramp PK2-1 – Single Cycle Ramp .......................................28
  4.2 Progression Problem 2: PCI Example Case – Single Cycle Ramp .......................................35
  4.3 Progression Problem 3: PCI Example Case – Increased Power ..........................................44
  4.4 Progression Problem 4: PCI Example Case – Down Power ...............................................49
  4.5 Progression Problem 5: PCI Example Case – Second Cycle Restart without Down Power ..........................................................................................................................53
  4.6 Progression Problem 6: PCI Example Case – Second Cycle Restart with Down Power ..........................................................................................................................58
  4.7 Progression Problem 7: Watts Bar - Multi-Cycle Analysis ..................................................63

5 CONCLUSIONS AND FUTURE WORK ......................................................................................69
  5.1 Conclusions ........................................................................................................................69
  5.2 Recommendations for Future Work ....................................................................................70
FIGURES

Figure 1. Cladding Inside Surface Hoop Stress Taken at a Point along the Axial Centerline of the Active Fuel Region for Progression Problem 6 ................................................................. iv

Figure 2. Diagram of the EnginFrame Workflow [3] .......................................................................................... 4

Figure 3. Diagram of VERA/VERA-CS (as of 5/28/2014) [5] ........................................................................... 5

Figure 4. Coupling among the Neutronics, Thermal Hydraulics, and Fuel Performance Components of VERA ................................................................................................................ 5

Figure 5. Peregrine Integration/Application Diagram [5] .............................................................................. 6

Figure 6. MOOSE TPL Dependency Diagram [11] ........................................................................................ 7

Figure 7. 2D Axisymmetric Finite Element Fuel Rod Model [13] ................................................................ 9

Figure 8. 2D Radial-Azimuthal Finite Element Fuel Rod Model [14] ............................................................. 9

Figure 9. 3D Fuel Rod Model Representation [13] .................................................................................... 10

Figure 10. Power History for the Super-Ramp Case ..................................................................................... 12

Figure 11. Power History for the PCI Example Case, Specifically Progression Problem 6 .................. 13

Figure 12. Power History for the Watts Bar Case [16], [17], [18], [19] ................................................. 14

Figure 13. Behavioral Interactions and Dependencies [21] ....................................................................... 16

Figure 14. Fuel Pellet Crack Plane Modes [24] ..................................................................................... 18

Figure 15. LWR Fuel Rod Diagram [24] ........................................................................................................ 20

Figure 16. Finite Element Diagram ............................................................................................................. 22

Figure 17. Finite Element Mesh Diagrams for Falcon (left) and Peregrine (right) ........................... 24

Figure 18. Sampled Locations for Thermal/Mechanical Analyses ............................................................... 28

Figure 19. Power History for Progression Problem 1 ........................................................................ 29

Figure 20. Bulk Coolant Temperature History Applied Uniformly as the Outer Cladding Boundary Condition for Progression Problem 1 .................................................................. 30

Figure 21. Final Temperature Profile for Progression Problem 1 .............................................................. 31

Figure 22. Fuel Centerline Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 1 .............................................................. 32
Figure 23. Fuel Mid-Plane Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 1

Figure 24. Fuel Outer Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 1

Figure 25. Cladding Inner Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 1

Figure 26. Cladding Outer Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 1

Figure 27. Axial Temperature Profile along the Fuel Centerline at 1126 days for Progression Problem 1

Figure 28. Axial Temperature Profile along the Fuel Outer Surface at 1126 days for Progression Problem 1

Figure 29. Power History for Progression Problem 2

Figure 30. Fuel Centerline Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 2

Figure 31. Fuel Mid-Plane Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 2

Figure 32. Fuel Outer Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 2

Figure 33. Difference in the Fuel Centerline Temperatures Plotted in Figure 30 for Progression Problem 2

Figure 34. Cladding Inner Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 2

Figure 35. Cladding Outer Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 2

Figure 36. Coolant Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 2

Figure 37. Temperature as a Function of Radial Position Taken at Points along the Axial Mid-Plane of the Active Fuel Region at 460 days for Progression Problem 2

Figure 38. Axial Temperature Profile along the Fuel Centerline at 460 days for Progression Problem 2

Figure 39. Axial Temperature Profile along the Fuel Outer Surface at 460 days for Progression Problem 2
Figure 40. Gap Thickness Predicted by Falcon for the Final Time Step (~460 days) for Progression Problem 2

Figure 41. Gap Thickness Predicted by Peregrine for the Final Time Step (~460 days) for Progression Problem 2

Figure 42. Temperature as a Function of Radial Position Taken at Points along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 3

Figure 43. Axial Temperature Profile along the Fuel Centerline for Progression Problem 3

Figure 44. Axial Temperature Profile along the Fuel Outer Surface for Progression Problem 3

Figure 45. Gap Thickness at the Axial Mid-Plane of the Active Fuel Region for Progression Problem 3

Figure 46. Gap Thickness at the Axial Mid-Plane of the Active Fuel Region for Progression Problem 3 Plotted for the First 5 Days of the Simulation Only

Figure 47. Temperatures (left) and Displacements (right) for the Fuel Outer Surface and Cladding Inner Surface Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 3

Figure 48. Power History for Progression Problem 4

Figure 49. Fuel Centerline Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 4

Figure 50. Fuel Outer Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 4

Figure 51. Temperature Difference between Falcon and Peregrine for 3 Different Location along the Fuel Axial Mid-Plane for Progression Problem 4

Figure 52. Gap Thickness at the Axial Mid-Plane of the Active Fuel Region for Progression Problem 4

Figure 53. Radial Displacements for the Fuel Outer Surface (left) and Cladding Inner Surface (right) Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 4

Figure 54. Power History for Progression Problem 5

Figure 55. Fuel Centerline Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 5

Figure 56. Fuel Outer Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 5

Figure 57. Fuel Outer Surface and Cladding Inner Surface Radial Displacements Taken at a Point along the Axial Centerline of the Active Fuel Region for Progression Problem 5
Figure D-2. Axial Temperature Profile along the Cladding Outer Surface at 1126 days for Progression Problem 1 .......................................................................................................................100

Figure D-3. Axial Temperature Profile along the Cladding Inner Surface at 460 days for Progression Problem 2 ...........................................................................................................................................101

Figure D-4. Axial Temperature Profile along the Cladding Inner Surface at 460 days for Progression Problem 2 ...........................................................................................................................................101

Figure D-5. Fuel Centerline Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 3 .............................................................................................102

Figure D-6. Fuel Mid-Plane Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 3 .............................................................................................102

Figure D-7. Fuel Outer Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 3 .................................................................................103

Figure D-8. Cladding Inner Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 3 .................................................................................103

Figure D-9. Cladding Outer Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 3 .................................................................................104

Figure D-10. Coolant Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 3 .....................................................................................................104

Figure D-11. Hoop Stress on the Cladding inside Surface Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 3 .............................................................105

Figure D-12. Fuel Mid-Plane Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 3 .............................................................................................106

Figure D-13. Cladding Inner Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 3 .................................................................................106

Figure D-14. Cladding Outer Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 3 .................................................................................107

Figure D-15. Fuel Mid-Plane Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 5 .................................................................................................108

Figure D-16. Cladding Inner Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region ............................................................................................................................108

Figure D-17. Cladding Outer Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region ............................................................................................................................109

Figure D-18. Temperature as a Function of Radial Position Taken at Points along the Axial Mid-Plane of the Active Fuel Region ............................................................................................................................109
TABLES

Table 1. Specifications for Phoebe ........................................................................................................3
Table 2. Progression Problem Descriptions..........................................................................................12
Table 3. Select Test Conditions for the Super-Ramp Test Problem [15], [13].....................................13
Table 4. Select Test Conditions for the PCI Example Test Problem.....................................................14
Table 5. Peregrine Material and Behavioral Models .................................................................................17
Table 6. Finite Element Parameters Used by Peregrine for Each Progression Problem ......................23
Table 7. Comparison of Output Sources for Falcon and Peregrine .....................................................26
Table 8. Computational Run Time Statistics for the Progression Problems..........................................28
Table B-1. Unit Comparison between Peregrine and Typical Industry Units .......................................79
Table B-2. Select Input Parameters for Mesh Generation in Peregrine..................................................82
Table B-3. Results from Cases 123 through 127 ..................................................................................86
## ACRONYMS

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D</td>
<td>Two-Dimensional</td>
</tr>
<tr>
<td>3D</td>
<td>Three-Dimensional</td>
</tr>
<tr>
<td>ANL</td>
<td>Argonne National Laboratory</td>
</tr>
<tr>
<td>CASL</td>
<td>Consortium for Advanced Simulation of Light Water Reactors</td>
</tr>
<tr>
<td>CS</td>
<td>Core Simulator</td>
</tr>
<tr>
<td>CSV</td>
<td>Comma-Separated Values</td>
</tr>
<tr>
<td>EPRI</td>
<td>Electric Power Research Institute</td>
</tr>
<tr>
<td>FE</td>
<td>Finite Element</td>
</tr>
<tr>
<td>FLOP</td>
<td>Floating-Point Operations per Second</td>
</tr>
<tr>
<td>GB</td>
<td>Gigabyte</td>
</tr>
<tr>
<td>GHz</td>
<td>Gigahertz</td>
</tr>
<tr>
<td>HBS</td>
<td>High-Burnup Structure</td>
</tr>
<tr>
<td>HDF</td>
<td>Hierarchical Data Format</td>
</tr>
<tr>
<td>HPC</td>
<td>High-Performance Computing</td>
</tr>
<tr>
<td>HYPRE</td>
<td>High Performance Preconditioners</td>
</tr>
<tr>
<td>IFPE</td>
<td>International Fuel Performance Experiments</td>
</tr>
<tr>
<td>INL</td>
<td>Idaho National Laboratory</td>
</tr>
<tr>
<td>JFNK</td>
<td>Jacobian-Free Newton-Krylov</td>
</tr>
<tr>
<td>LGPL</td>
<td>Lesser General Public License</td>
</tr>
<tr>
<td>LHGR</td>
<td>Linear Heat Generation Rate</td>
</tr>
<tr>
<td>LWR</td>
<td>Light Water Reactor</td>
</tr>
<tr>
<td>MOAB</td>
<td>Mesh-Oriented Database</td>
</tr>
<tr>
<td>MOOSE</td>
<td>Multiphysics Object Oriented Simulation Environment</td>
</tr>
<tr>
<td>MS</td>
<td>Modeling and Simulation</td>
</tr>
<tr>
<td>NQA1</td>
<td>Nuclear Quality Assurance Level 1</td>
</tr>
<tr>
<td>NRC</td>
<td>Nuclear Regulatory Commission</td>
</tr>
<tr>
<td>ORNL</td>
<td>Oak Ridge National Laboratory</td>
</tr>
<tr>
<td>PCI</td>
<td>Pellet-Cladding Interaction</td>
</tr>
<tr>
<td>PCMI</td>
<td>Pellet-Cladding Mechanical Interaction</td>
</tr>
<tr>
<td>PDE</td>
<td>Partial Differential Equation</td>
</tr>
<tr>
<td>PETSC</td>
<td>Portable, Extensible Toolkit for Scientific Computation</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------</td>
<td>------------</td>
</tr>
<tr>
<td>PIE</td>
<td>Post-Irradiation Examination</td>
</tr>
<tr>
<td>PNNL</td>
<td>Pacific Northwest National Laboratory</td>
</tr>
<tr>
<td>PWR</td>
<td>Pressurized Water Reactor</td>
</tr>
<tr>
<td>QA</td>
<td>Quality Assurance</td>
</tr>
<tr>
<td>QDR</td>
<td>Quad Data Rate</td>
</tr>
<tr>
<td>SI</td>
<td>International System of Units</td>
</tr>
<tr>
<td>SIFGRS</td>
<td>Simple Integrated Fission Gas Release and Swelling</td>
</tr>
<tr>
<td>SNL</td>
<td>Sandia National Laboratories</td>
</tr>
<tr>
<td>TB</td>
<td>Terabyte</td>
</tr>
<tr>
<td>TPL</td>
<td>Third-Party Library</td>
</tr>
<tr>
<td>TVA</td>
<td>Tennessee Valley Authority</td>
</tr>
<tr>
<td>UTK</td>
<td>University of Tennessee in Knoxville</td>
</tr>
<tr>
<td>VERA</td>
<td>Virtual Environment for Reactor Applications</td>
</tr>
<tr>
<td>WBN1</td>
<td>Watts Bar Nuclear Unit 1</td>
</tr>
</tbody>
</table>
1 INTRODUCTION

There are many phenomena, both macroscopic and microscopic, that have significant effects on the operations, safety, and performance of a nuclear reactor. The objective of the Consortium for Advanced Simulation of Light Water Reactors (CASL) is to enhance the performance of light water reactors (LWRs) through high-fidelity modeling and simulation. Accomplishing this objective involves developing a “virtual reactor” environment capable of performing the necessary calculations to simulate the operation of a nuclear reactor and therefore provide a platform for understanding and addressing significant reactor design, operational, and safety issues in the industry.

This report describes the Electric Power Research Institute’s (EPRI) Test Stand experience with the Peregrine fuel performance code, a component of the Virtual Environment for Reactor Applications (VERA). The purpose of this Test Stand is to serve as an alpha deployment of the VERA software suite with a CASL industry partner to gain enhanced feedback and testing of the software in a real world environment with analyses using both proprietary and non-proprietary reactor data. This report provides an assessment of the VERA software suite, specifically the fuel performance component, Peregrine. The Test Stand also provides feedback to CASL in order to aid in the development and deployment of future VERA software and components and their applicability to industry issues in commercial power plant operation and design. Both technical and programmatic feedback and discussions on subjects such as simulation capabilities, compatibility with industry computing platforms, usability of the software, and performance relative to existing industry tools are discussed. Detailed feedback is located in Appendix B.

Evaluation of Peregrine as a state-of-the-art fuel performance code is accomplished by evaluating a series of fuel performance progression problems. These progression problems use two-dimensional (2D) axisymmetric models and focus on examining the thermal and mechanical responses of the fuel and cladding to an imposed axially-varying power history. The progression begins with a constant axial power profile imposed during a single cycle ramp up to power for a short length test rod and concludes with the most complex Test Stand case, an axially-varying power history containing a first cycle ramp and hold followed by a short period of down time and a subsequent second-cycle ramp to full power. The methods and procedures used for this final progression problem is also used to examine the fuel performance of a Watts Bar Unit 1 fuel rod. Reactor data provided for this final case was used with permission from Westinghouse and the Tennessee Valley Authority (TVA). The ultimate goal is to evaluate Peregrine and its ability to perform pellet-cladding interaction (PCI) analyses. For a more in-depth discussion of fuel performance simulations and PCI, refer to Reference [1].

All Peregrine Test Stand simulations were run using EPRI’s industry class computing cluster Phoebe, which was purchased as part of the CASL program to provide EPRI with a high-performance computing (HPC) capability. VERA software was retrieved from CASL source code repositories. Specific versions of each software component were identified by CASL to designate each EPRI Test-Stand deployment version.

The report is organized into the following sections. Section 2 discusses the deployment of the Peregrine Test Stand on EPRI’s high-performance computer, Phoebe. Section 3 covers the modeling and analysis methodology applied which includes Peregrine-specific modeling options, boundary conditions, and solution methods and options. Section 4 reviews the results for the
progression problems. Section 5 presents the conclusions and recommendations for future work identified from this work.
2 TEST STAND DEPLOYMENT

The initial deployment of this Test Stand was performed in October 2013. During this initial deployment period, VERA was installed on Phoebe, EPRI’s industry class computing cluster. Details about Phoebe, VERA, and Peregrine are provided in the following subsections.

2.1 Phoebe

Phoebe\(^1\) is a Hewlett-Packard high-performance computing cluster with 31 ProLiant SL230s compute nodes. Each compute node has two Intel\(^\circledR\) Xeon\(^\circledR\) E5-2670 processors (8 cores, 2.6 gigahertz (GHz) each) and 64 gigabytes (GBs) of shared memory for a total of 496 compute cores, 2 terabytes (TBs) of memory, and an approximate computational speed of 8 billion floating point operations per second (FLOPS), or 8 teraFLOPS. The nodes are connected to the storage server using quad data rate (QDR) Infiniband interconnects and 10 GB Ethernet interconnects are used between all nodes [2]. A table containing the specifications for Phoebe is provided in Table 1.

<table>
<thead>
<tr>
<th>Processors</th>
<th>Intel(^\circledR) Xeon(^\circledR)</th>
</tr>
</thead>
<tbody>
<tr>
<td># of nodes</td>
<td>31</td>
</tr>
<tr>
<td># of compute cores</td>
<td>496</td>
</tr>
<tr>
<td>Storage</td>
<td>100 TB</td>
</tr>
<tr>
<td>Memory</td>
<td>2 TB</td>
</tr>
<tr>
<td>Speed</td>
<td>8 teraflops</td>
</tr>
<tr>
<td>Communication</td>
<td>QDR Infiniband (10 GB/s)</td>
</tr>
<tr>
<td>Operating systems</td>
<td>Linux, Windows</td>
</tr>
</tbody>
</table>

Phoebe features both Windows (Server 2008 HPC Edition) and Linux (Red Hat 6.4) operating systems so that both Windows and Linux applications can be run simultaneously. To accomplish this, there is a ProLiant DL380p Linux head node, a ProLiant SL230s Windows head node, and a ProLiant SL4540 storage node. Of the 31 compute nodes, 15 of them are dual-bootable, supporting both Windows and Linux operation systems. Computing resources for Phoebe for both Linux and Windows jobs are managed through the EnginFrame [3] web interface or via the Torque [4] queuing system. A workflow diagram for EnginFrame is provide in Figure 2.

Over the course of this Test Stand, 15 000+ core hours were used on Phoebe to perform the simulations for this Test Stand.

\(^1\) Named after the Greek goddess of prophecy and oracular intellect.
The Virtual Environment for Reactor Applications (VERA) is a suite of software components that can be coupled to model quasi-steady-state LWR conditions in high-performance computing (HPC) applications. VERA includes multiple independent software modules that perform neutron transport, cross section processing, thermal-hydraulics, thermo-mechanical (fuel performance), and isotopic depletion calculations. The VERA Core Simulator, VERA-CS, features neutronics, sub-channel thermal hydraulics, and fuel performance components. The applications are then integrated into VERA-CS using a coupled framework which includes VERAIn, which unifies input file structure across all VERA applications, the Data Transfer Kit (DTK), which unifies the mesh representations across all VERA applications, as well as other infrastructural components. A diagram showing the various components of VERA is provided in Figure 3, and a diagram describing the coupling among neutronics, thermal hydraulics, and fuel performance is provided in Figure 4.

The initial installation of VERA at EPRI, VERA 3.3 DEV, was completed in October 2013. This installation included a standalone installation of Peregrine because this was the only component under examination during this Test Stand. Note that updates to this installation were performed in February 2014, April 2014, and July 2014 as additional capabilities became available.
Figure 3. Diagram of VERA/VERA-CS (as of 5/28/2014) [5]

Figure 4. Coupling among the Neutronics, Thermal Hydraulics, and Fuel Performance Components of VERA
2.3 Peregrine

Peregrine, the fuel performance code contained within VERA, is an LWR fuel performance code developed to investigate the effects of reactor operation on individual fuel rods and assess safety and operational margins [6]. Peregrine can couple various thermal, mechanical, chemical, and irradiation-induced effects models to provide high-fidelity fuel performance modeling solutions for LWR applications. A diagram with some of the physics models included in Peregrine and how they are coupled to VERA-CS is provided in Figure 5.

![Figure 5. Peregrine Integration/Application Diagram](image)

Peregrine is built upon the Multiphysics Object Oriented Simulation Environment (MOOSE) [7], developed at Idaho National Laboratory (INL). MOOSE is a computational framework for multiphysics simulation that uses a set of general-use finite element methods and scalable solvers and preconditioners to solve systems of partial differential equations (PDEs) in parallel. The third-party library (TPL) libMesh [8], developed by the University of Texas at Austin and distributed under the GNU Lesser General Public License (LGPL), is an open-source environment for finite element programming applications and is used to provide the parallel adaptive finite element capabilities within MOOSE. In addition, MOOSE uses the Portable, Extensible Toolkit for Scientific Computation (PETSc) [9], developed at Argonne National Laboratory (ANL), to provide a suite of parallel linear/nonlinear solvers used to solve systems of PDEs in parallel. PETSc, in addition to its own solvers, utilizes the HYPRE [10] software library.
for additional solvers and preconditioners used for solving linear systems of equations on massively parallel machines. A dependency diagram for MOOSE and its required TPLs is provided in Figure 6.

Figure 6. MOOSE TPL Dependency Diagram [11]

The MOOSE framework was chosen as a foundation for Peregrine for the following reasons:

1. The existing fuel performance code BISON, already written using the MOOSE framework, served as an architectural starting point to quickly develop Peregrine.
2. Maintenance of the underlying code is managed by other projects and leveraged to make Peregrine more robust.

3. The object oriented framework is easily extensible and therefore allows other fuel and cladding models to be quickly incorporated into the code base.

4. MOOSE currently meets all NQA-1 (Nuclear Quality Assurance Level 1) requirements [12].

Traditionally, fuel performance modeling and simulation (M&S) applied a 1.5D representation of a fuel rod at several axial sections with a radial one-dimensional description. In fact, 1.5D M&S is the preferred method for certified safety analysis because it is simple and several vendor codes have been certified by the Nuclear Regulatory Commission (NRC) for this purpose.

Advanced in the computational community over recent decades have made 2D axisymmetric M&S (illustrated in Figure 7) available with reasonable solution times and are now bringing three-dimensional (3D) M&S into the realm of practicality. There are many advantages to 2D axisymmetric models when the simplifications needed by traditional methods no longer apply, and this concept is extended when moving to 3D space. For example, to approach 3D modeling, local effects associated with pellet radial cracks are investigated using plane strain 2D radial-azimuthal cross section (slice) models (Figure 8). 3D modeling in Peregrine offers the ability to represent the geometry and boundary conditions for a reactor fuel rod to a higher degree. An illustration of a shortened fuel rod model consisting of five individual fuel pellets is provided in Figure 9. Note that only 2D R-Z axisymmetric models are used for this Test Stand.
Figure 7. 2D Axisymmetric Finite Element Fuel Rod Model [13]

Figure 8. 2D Radial-Azimuthal Finite Element Fuel Rod Model [14]
Figure 9. 3D Fuel Rod Model Representation [13]
3 METHODOLOGY

This section describes the problem statements, modeling options, and solution methods applied during the EPRI Test Stand. An overview of the Test Stand fuel performance progression problem strategy, which was developed specifically for this Test Stand, is presented along with the modeling options, boundary conditions, solution methods, and analysis tools used in the subsections to follow.

Note that italicized words are used to represent Peregrine input file keywords throughout this report.

3.1 Analysis Methodology

3.1.1 Progression Problems

A series of progression problems was developed in order to evaluate Peregrine’s thermo-mechanical 2D axisymmetric fuel performance modeling capabilities. Details on the purpose of each progression problem are provided in Table 2. The progression begins in problem 1 with a flat axial power profile imposed on a shortened test rod for a single cycle ramp. Relevant model parameters for this case, denoted as the Super-Ramp [15], [13] case throughout this report, are provided in Table 3. Progression problem 2 then moves to a generic PWR fuel rod example case, denoted as the PCI example\(^1\) case throughout this document, is used to investigate the thermal and mechanical responses predicted when an axially-varying power history is imposed on a full length fuel rod. The imposed power history for this progression problem includes a ramp up to steady-state operation and hold for a full cycle. In progression problem 3, the power is increased to achieve gap closure and enable investigation of the thermo-mechanical responses estimated by Peregrine with respect to pellet-cladding contact. Progression problem 4 then appends a period of down power to the power history and evaluates the thermo-mechanical response as the model ramps down from steady-state to zero power. Progression problem 5 takes progression problem 3 and appends an additional ramp period to the end of the steady-state operational period. For progression problem 6, the down power and second cycle ramp are both appended (in that order) to the end of steady-state operation for the first cycle to investigate thermo-mechanical responses under second-cycle restart conditions. Lastly, in progression problem 7, operational data provided by Westinghouse and TVA for Watts Bar Nuclear Unit 1 (WBN1) are used to perform fuel performance analysis for a real PWR reactor fuel rod.

The ultimate goal is to evaluate Peregrine and its ability to perform pellet-cladding interaction (PCI) analyses. For a more in-depth discussion of fuel performance simulations, refer to Reference [1].

\(^1\) This case was used as a non-proprietary Falcon example case provided by EPRI to CASL to facilitate an open exchange between CASL entities interested in developing Peregrine: INL, Pacific Northwest National Laboratory (PNNL), ANATECH, Westinghouse, and the University of Tennessee in Knoxville (UTK).
### Table 2. Progression Problem Descriptions

<table>
<thead>
<tr>
<th>Problem</th>
<th>Model</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Super-Ramp</td>
<td>Thermal analysis of shortened rod with a flat axial profile</td>
</tr>
<tr>
<td>2</td>
<td>PCI Example</td>
<td>Single-cycle thermo-mechanical analysis</td>
</tr>
<tr>
<td>3</td>
<td>PCI Example</td>
<td>Gap closure analysis with an elevated power history.</td>
</tr>
<tr>
<td>4</td>
<td>PCI Example</td>
<td>Thermo-mechanical analysis including down power</td>
</tr>
<tr>
<td>5</td>
<td>PCI Example</td>
<td>Thermo-mechanical analysis with additional power ramp</td>
</tr>
<tr>
<td>6</td>
<td>PCI Example</td>
<td>Thermo-mechanical analysis with down power and 2(^{nd}) cycle ramp</td>
</tr>
<tr>
<td>7</td>
<td>Watts Bar</td>
<td>Thermo-mechanical analysis using a Watts Bar fuel rod model</td>
</tr>
</tbody>
</table>

**Figure 10. Power History for the Super-Ramp Case**
Table 3. Select Test Conditions for the Super-Ramp Test Problem [15], [13]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coolant temperature at core inlet</td>
<td>566 K</td>
</tr>
<tr>
<td>Mass flow rate</td>
<td>6833 kg/s</td>
</tr>
<tr>
<td>Average system pressure</td>
<td>14.5 MPa</td>
</tr>
<tr>
<td>Active length of core</td>
<td>0.3174 m</td>
</tr>
<tr>
<td>Fuel density</td>
<td>10340 kg/m</td>
</tr>
<tr>
<td>Pellet outer diameter</td>
<td>9.138 mm</td>
</tr>
<tr>
<td>Cladding outer diameter</td>
<td>10.75 mm</td>
</tr>
<tr>
<td>Cladding thickness</td>
<td>0.74 mm</td>
</tr>
</tbody>
</table>

Figure 11. Power History for the PCI Example Case, Specifically Progression Problem 6
Table 4. Select Test Conditions for the PCI Example Test Problem

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coolant temperature at core inlet</td>
<td>560 K</td>
</tr>
<tr>
<td>Mass flow rate</td>
<td>3024 kg/s</td>
</tr>
<tr>
<td>Average system pressure</td>
<td>15.5 MPa</td>
</tr>
<tr>
<td>Active length of core</td>
<td>3.6576 m</td>
</tr>
<tr>
<td>Fuel density</td>
<td>10465 kg/m</td>
</tr>
<tr>
<td>Pellet outer diameter</td>
<td>8.1915 mm</td>
</tr>
<tr>
<td>Cladding outer diameter</td>
<td>8.3566 mm</td>
</tr>
<tr>
<td>Cladding thickness</td>
<td>0.5715 mm</td>
</tr>
</tbody>
</table>

Figure 12. Power History for the Watts Bar Case [16], [17], [18], [19]
3.1.2 Peregrine/Falcon Comparisons

Many of the results produced during this Test Stand are displayed as comparisons between Peregrine and Falcon simulations, with both cases using similar inputs. Version 1.2 of the Falcon fuel performance code [20] represents the version of Falcon used throughout this Test Stand. For details on the VERA versions used to generate the results within this report, see Appendix B.5.1.

Considering that a number of physics models are the same in both Peregrine and Falcon it is a reasonable assumption that the same input in both codes would result in essentially the same output. However, because the two codes are coupled and solved with different solution methods and procedures, the results are not expected to be identical although they are expected to display similar trends with reasonably close output values. These results are also provided in this format in order to compare Peregrine results to an industry standard tool as well as to help identify differences between the codes that may guide developers in determining further verification and validation needs. Other differences in the way solutions are reached for these time-based analyses may highlight the potential advantages or disadvantages associated with the advanced solution methods used by Peregrine. Note that Falcon results are not being considered as a benchmark solution against which Peregrine is being measured. The Falcon results are provided to illustrate differences between Peregrine and an industry-standard fuel performance code with over 40 years of development history and industry use. More information on Falcon is provided in Appendix A.

Output results throughout Section 4 of this report are plotted as a function of time (in days) or position (in meters). The various thermal and mechanical responses are reported using the International System of Units (SI). Some notable exceptions include using days instead of seconds because of time scale practicalities and kW/ft for linear heat generation rates (LHGRs) instead of W/m because of industry traditional use and familiarity.
3.2 Modeling Options

Fuel performance modeling is a coupled thermal-chemical-mechanical process that results in a rather complex modeling environment containing a large number of different physical phenomena taking place over the course of a given calculation, with several exhibiting nonlinear behaviors. Figure 13 illustrates a high-level dependency diagram of the different mechanisms present in a fuel performance code. The individual models relevant to this work for fuel pellets, cladding, internal voids (gaps), and coolant conditions, are listed in Table 5 for both Falcon and Peregrine.

Figure 13. Behavioral Interactions and Dependencies [21]
<table>
<thead>
<tr>
<th>Material</th>
<th>Property</th>
<th>Peregrine</th>
<th>Falcon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuel</td>
<td>Thermal Conductivity</td>
<td>Falcon/NFIR</td>
<td>MATPRO [22] + Literature</td>
</tr>
<tr>
<td></td>
<td>Heat Capacity</td>
<td>FPE [23]</td>
<td>MATPRO</td>
</tr>
<tr>
<td></td>
<td>Relocation</td>
<td>Falcon</td>
<td>ESCORE</td>
</tr>
<tr>
<td></td>
<td>Thermal/Irradiation Creep</td>
<td>Falcon</td>
<td>Modified MATPRO</td>
</tr>
<tr>
<td></td>
<td>Smeared Cracking</td>
<td>Falcon</td>
<td>EPRI</td>
</tr>
<tr>
<td></td>
<td>Thermal Expansion</td>
<td>MATPRO</td>
<td>MATPRO</td>
</tr>
<tr>
<td></td>
<td>Solid Swell</td>
<td>Falcon</td>
<td>Literature</td>
</tr>
<tr>
<td></td>
<td>Densification</td>
<td>ESCORE</td>
<td>MATPRO, ESCORE</td>
</tr>
<tr>
<td></td>
<td>Fission Gas Release</td>
<td>SIFGRS</td>
<td>Modified Forsberg-Massih</td>
</tr>
<tr>
<td></td>
<td>Radial Power/Burnup</td>
<td>TUBRNP</td>
<td>TUBRNP</td>
</tr>
<tr>
<td>Cladding</td>
<td>Thermal Conductivity</td>
<td>MATPRO</td>
<td>MATPRO</td>
</tr>
<tr>
<td></td>
<td>Irradiation Growth</td>
<td>Falcon</td>
<td>MATPRO, ESCORE</td>
</tr>
<tr>
<td></td>
<td>Thermal/Irradiation Creep</td>
<td>Limback/Falcon</td>
<td>Limback</td>
</tr>
<tr>
<td></td>
<td>Thermal Expansion</td>
<td>MATPRO</td>
<td>MATPRO</td>
</tr>
<tr>
<td>Gap</td>
<td>Gap Conductivity</td>
<td>MATPRO</td>
<td>Literature</td>
</tr>
<tr>
<td></td>
<td>Void Volume</td>
<td>Ideal Gas Law</td>
<td>Ideal Gas Law</td>
</tr>
<tr>
<td>Coolant</td>
<td>Single Channel Enthalpy Rise Model</td>
<td>CASL</td>
<td>EPRI</td>
</tr>
</tbody>
</table>

### 3.2.1 Relocation

Relocation, the process by which some of the initial fuel-cladding gap is reduced via the distribution of cracking within the fuel pellet, has an effect on the thermal and mechanical behavior of the fuel pellet region over time. Relocation is a function of fuel rod design, fuel properties, power level, and burnup [1]. A relocation threshold of 5 kW/ft (1.64 × 10⁴ W/m) was used for all analyses presented in Section 4 for both Falcon and Peregrine. Investigation of the relocation model in Peregrine which supports this decision is presented in Appendix C.1 as a supplemental analysis.

### 3.2.2 Smeared Cracking

As uranium dioxide (UO₂) fuel pellets heat up in a reactor, a large temperature gradient develops from the fuel centerline to the fuel outer surface. This large temperature gradient induces tensile stresses that contribute to cracking in the fuel. Predicting the location of cracks, considered a stochastic event, could be extremely complex as it would require a large number of simulations to determine the average behavior using traditional statistical simulation techniques such as Monte Carlo. Furthermore, considering the level of information typically provided by the manufacturer and lack of stress-strain in tension for fuel pellets, these predictions would be suspect. Thus, a smeared-cracking model is used when the cracking is viewed as a mechanism that changes the material behavior from isotropic to orthotropic and the material stiffness normal to the crack surface drops to zero while full stiffness parallel to the crack is maintained. Cracks in the fuel can be represented in three different planes and develop with independent histories, including closing and possible healing if temperature rises above the sintering temperature (Figure 14).
Peregrine’s smeared cracking model compares a critical stress to each principal stress at quadrature points in the fuel. If a given principal stress is above the specified critical stress, the fuel is considered cracked at that point in space in the direction of the principal stress. At this point, the stress is reduced to zero and the material will have no strength unless the strain becomes compressive around the imposed crack. Lastly, because the solution difficulty increases as the number of cracks increases for a given integration point, control parameters are available to limit the number of directions of imposed cracks [25].

Peregrine’s smeared cracking model was not used during the Test Stand. This decision was made at the recommendation of the Peregrine development team. An investigation of the effects of the smeared cracking model in Falcon is presented in Appendix C.2 as a supplemental analysis to illustrate the effects Falcon’s smeared cracking model has on the estimated results. Future work should involve performing Peregrine simulations with the smeared cracking model enabled and recreating the comparisons illustrated in Section 4.

### 3.2.3 Fission Gas Release

Fission products are continuously produced within the fuel during power operation. The fission products of concern for fuel performance modeling are xenon and krypton. These fission products, which exist in the gaseous phase, have significant effects on the thermal and mechanical properties of the fuel rod. Gaseous fission products lead to fuel swelling, rod internal pressure increases, and changes to the fill gas composition (which degrades the thermal conductivity across the fuel-cladding gap).

The model used within Peregrine for all simulations reported on in this report is the Simple Integrated Fission Gas Release and Swelling model (SIFGRS) developed at INL\(^2\). It is used to

---

\(^2\) A modified version of the Forsberg-Massih fission gas release model is also included in Peregrine, however the SIFGRS model is the recommended model [23].
determine the fission gas production, release, and swelling in UO$_2$. The SIFGRS fission gas release model incorporates gaseous diffusion, precipitation in grains, growth and coalescence of gas bubbles at grain faces, as well as thermal, athermal, steady-state, and transient fission gas release [25]. In-depth analysis of this model and the fission gas release solution provided by Peregrine was not included in the scope of this Test Stand. For more information and feedback on Peregrine’s fission gas release capabilities, see Appendix B.3.6.

3.2.4 Contact and Friction

Peregrine has two contact/friction formulations, penalty and kinematic, and three contact models, glued, coulomb, and frictionless. The combination of kinematic/frictionless$^3$ appears most appropriate for modeling the gap behavior between fuel pellets and the cladding tube for this work as it predicts both gap closure and reopening as well as the resulting buildup and relaxation of cladding stresses. This combination also predicts gap reopening during power ramp down periods. Another important parameter required for the contact/friction specification is the penalty factor. This parameter represents the stiffness of a constraint that prevents the fuel and cladding surfaces from penetrating one another by applying a normal force along the contact surface [26]. The penalty is a function of the material stiffness and the finite element size.

Progression problems 1 through 7 used the kinematic formulation and frictionless contact model. The frictionless contact model used for progression problem 1 at the time the simulations were performed did not allow the pellet and cladding to separate once they came in contact; however progression problem 1 only focused on thermal results and furthermore does not include a ramp down period which would prompt gap reopening. Progression problems 2 through 7 used the constraint-based contact system$^4$. Complete analysis of the contact and friction models available within Peregrine is beyond the scope of this report.

3.3 Boundary Conditions

A diagram of a 2D fuel performance model is provided in Figure 15. This type of geometric model representation is used for both Falcon and Peregrine. The only exception is that the end plugs in Falcon are modeled as having different thicknesses while Peregrine models both end plugs as having the same thickness. This choice simplified the mesh generation procedure for the Peregrine simulations and is not anticipated to significantly affect any results presented in this report.

---

$^3$ Note that the frictionless model existed under the name experimental throughout this Test Stand. This model was integrated into the frictionless model on July 17$^{th}$, 2014 by the BISON development team.

$^4$ The constraint-based contact system was made available on July 1$^{st}$, 2014 to the EPRI Test Stand Team and was deemed the best implementation for pellet-cladding contact modeling by the BISON and Peregrine development teams.
3.3.1 Power History

Perhaps the most important forcing function input to thermo-mechanical fuel performance simulations is the fuel rod power history. The power history is used to calculate the volumetric heat generation in the pellet resulting from fission reactions. The rod average power history for a fuel rod is typically entered and augmented by a time dependent axial profile to define the local power at discrete elevations. Due to the importance of the power history on the fuel performance simulation, this is the primary input parameter that is varied over the course of the progression problems.
One observation is that Peregrine can only use linear interpolation to estimate the power history between discrete power changes, while Falcon can use linear interpolation or apply step changes to the power level after a change by interjecting a small ramp to with duration of 0.1 hr. This changes the integrated power (i.e. the area under the power history curves), which affects all results related to temperature and burnup (such as heat transfer, creep rates, stress, and other material properties). This highlights the need to understand the way the power history is provided so that it can be properly represented by the software.

A second notable point is that the calculation of the volumetric heat generation rate in the pellet from fission reactions represents a source term calculation that has not been fully verified within Peregrine, and thus the volumetric heat generation rate may be a source of difference between the two codes which will be more relevant near the center of the pellet.

### 3.3.2 Coolant Channel Model

During power operation, coolant constantly flows along the length of the fuel rod on the cladding outer surface in order to extract the heat generated by fission reactions within the fuel rod. The cladding outer surface defines the control volume boundary, and the values used to formulate the boundary condition on the outer surface of the cladding are specified by a coolant channel model. The model is a single channel enthalpy rise coolant model that, given the inlet temperature, pressure, and mass flux, along with the geometry of the channel, estimates the coolant temperature and corresponding heat transfer coefficient for each outer cladding mesh element face modeled. The heat generated by the fuel is ultimately transferred to the bulk coolant flowing past the fuel rod [25].

The coolant channel model was used as the cladding outside boundary condition for progression problems 2 through 7. Progression problem 1 used a cladding outside surface temperature profile as the cladding outside boundary condition.

### 3.4 Solution Control

Peregrine estimates a numerical solution for differential equations that vary spatially and temporally. In general, the smaller the step size in space and time, the more accurate the result; however, this required more computational resources. The spatial and temporal discretization is defined by two factors: meshing and time stepping. These two topics are briefly discussed below.

#### 3.4.1 Meshing

Peregrine is a finite element (FE) based code that uses libMesh [27] to provide the underlying finite element formulation. Finite element codes have many options for defining the finite element formulation. To illustrate these differences, Figure 16 shows a 9-node element. There are three different types of elements used during this Test Stand: QUAD4, QUAD8, and QUAD9. QUAD4 elements are rectangular elements consisting of 4 nodes at points 1 to 4 (i.e. one node at each corner) in Figure 16. Likewise, QUAD8 and QUAD9 elements are rectangular elements consisting of nodes at points 1 to 8 and 1 to 9 in Figure 16, respectively. For the simulations performed during this Test Stand, Peregrine primarily used QUAD8 elements while Falcon used QUAD9 elements. This choice was made because QUAD8 elements represented the

---

5 These numbers are referring to the subscripts of the variable θ in Figure 16.
element type most often used by the development teams while QUAD9 elements represented the default for Falcon simulations.

![Figure 16. Finite Element Diagram](image)

In addition to the type of element, the order of the functions used to represent the basis functions must also be specified. QUAD4 elements are restricted to first-order basis functions because only two quadrature points are available along each line segment to formulate the function. For QUAD8 and QUAD9 elements, a second-order representation is normally sufficient. Higher orders were required in some progression problems in order to facilitate changes imposed by the constraint-based contact system.

For all progression problems, 5 radial elements were used in the fuel and 3 radial elements were used in the cladding. Additional relevant finite element options used for each progression problem are provided in Table 6 and includes the finite element type, the order of the basis functions, and the number of axial elements used to model the fuel and cladding regions. All Falcon simulations used 24 axial elements in the fuel and cladding regions and QUAD9 elements with second-order basis functions. Both Falcon and Peregrine used a single element row to represent the upper and lower end plugs and Falcon used a single element row to represent the upper and lower plena. More information on the difference in the FE implementation used by Falcon and Peregrine is provided in Section 3.4.4 as well as Appendix B.4.1.

---

6 Also note that QUAD4 elements are no longer used in Falcon and that the maximum number of axial elements is fixed at 50. The selection of QUAD9 elements and 24 axial elements in the fuel and cladding is recommended by the Falcon development team.
### 3.4.2 Parallelism

Peregrine uses a Jacobian-Free Newton-Krylov (JFNK) method for solving the finite element representation of the fuel performance models. One of the key benefits of this solver is that it can be executed in parallel, and the solutions obtained should be independent of the number of processors used to perform the simulation. Additional analyses regarding the scalability and reproducibility of Peregrine simulations is provided in Appendix B.4.1.5.

All Peregrine cases were run on EPRI’s high-performance computer Phoebe using between 1 and 16 cores on a single node.

### 3.4.3 Time Stepping

Determining the proper time step size for a solution is difficult in any multiphysics simulation environment. Throughout the simulation, time steps need to be small enough to capture any significant thermal, mechanical, or chemical changes. While some of these are gradual, others occur quickly, and the exact time at which a given phenomenon may occur is generally unknown before the simulation. The primary time-dependent data known a priori are the power history, axial shape history, and any time-varying boundary conditions that may be applied such as cladding outside surface temperatures.

Concerning the results contained in this report, progression problem 1 used time step intervals suggested by the Peregrine development team. Progression problems 2 through 5 used the IterationAdaptiveDT time stepping option in Peregrine. The time step limiting function used was a custom function which represented the time steps used by the Falcon simulation. Progression problems 6 and 7 used the power history as the time step limiting function since using a custom time stepping function constructed from the Falcon time steps resulted in convergence issues encountered by the solver.

### 3.4.4 Post-Processing

Multiple different output formats are available for Peregrine. The primary output format used by the Peregrine user community is an Exodus-II file [28], which can be processed using visualization software such as Paraview [29] or VisIt [30]. Additionally, a comma-separated values (CSV) file is used to provide a summary of post-processor variables defined in the input file.

To facilitate rapid analysis of Peregrine and Falcon results, several element- and point-based post-processors were created within the Peregrine input files to report sufficient information to the CSV files. This was required for three reasons: (1) Falcon does not produce output in a

---

<table>
<thead>
<tr>
<th>Progression Problem</th>
<th>Element Type</th>
<th>Basis Function Order</th>
<th># Axial Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>QUAD4</td>
<td>1st</td>
<td>48</td>
</tr>
<tr>
<td>2</td>
<td>QUAD8</td>
<td>2nd</td>
<td>70</td>
</tr>
<tr>
<td>3</td>
<td>QUAD8</td>
<td>2nd</td>
<td>70</td>
</tr>
<tr>
<td>4</td>
<td>QUAD8</td>
<td>7th</td>
<td>70</td>
</tr>
<tr>
<td>5</td>
<td>QUAD8</td>
<td>7th</td>
<td>70</td>
</tr>
<tr>
<td>6</td>
<td>QUAD8</td>
<td>2nd</td>
<td>70</td>
</tr>
<tr>
<td>7</td>
<td>QUAD8</td>
<td>7th</td>
<td>70</td>
</tr>
</tbody>
</table>
format readable by Paraview, (2) point-based postprocessors were required to ensure that comparisons made between the two codes were comparing points at the same location, and (3) no tools capable of deciphering Exodus-II formatted files were available\(^7\). To elaborate, Figure 17 illustrates a finite element mesh produced by both Falcon and Peregrine for a model with 5 radial elements in the fuel, 3 radial elements in the cladding, and 8 axial elements in both the fuel and cladding. The difference between the two model formulations is that Falcon uses gap elements to model the fuel-clad gap as well as the upper and lower plenum. Because Falcon is using gap elements in this way, the FE mesh includes an extra couple of rows of elements in the cladding tube to represent the upper and lower plena and an extra column of elements to represent the cladding gap. This arrangement of elements affects the location of the Peregrine cladding elements as they instead overlap the upper and lower plena. Thus, a given Peregrine element or node in the cladding will not correspond to the same location in the Falcon model. To manage this situation for node-based values, point post-processors were placed at locations identical to the node locations in the Falcon model. Unfortunately, no such post-processor fix is available for element-based data. Thus, when comparing element-based data, the element occupying most of the area where the corresponding Falcon element would reside (if superimposed on the Peregrine mesh) is used for comparison.

\[^7\] Although Paraview and VisIt are capable of deciphering Exodus-II formatted files, these tools are meant for visualizing data for a single output file and do not have the capacity to rapidly extract data for use in comparisons with Falcon.
Table 7 lists the output parameters compared within this report and the source of the information for each code (i.e. element, node, or point-based values). Coolant temperature and hoop stress represent two parameters where the data extracted are not from the exact same spatial location because of constraining differences in each code.
### Table 7. Comparison of Output Sources for Falcon and Peregrine

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Falcon</th>
<th>Peregrine</th>
<th>Exact Comparison</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuel Temperature</td>
<td>Node</td>
<td>Point</td>
<td>True</td>
</tr>
<tr>
<td>Clad Temperature</td>
<td>Node</td>
<td>Point</td>
<td>True</td>
</tr>
<tr>
<td>Coolant Temperature</td>
<td>Node</td>
<td>Element</td>
<td>False</td>
</tr>
<tr>
<td>Fuel Displacement</td>
<td>Node</td>
<td>Point</td>
<td>True</td>
</tr>
<tr>
<td>Clad Displacement</td>
<td>Node</td>
<td>Point</td>
<td>True</td>
</tr>
<tr>
<td>Gap Thickness</td>
<td>Element</td>
<td>Point</td>
<td>True</td>
</tr>
<tr>
<td>Hoop Stress</td>
<td>Element</td>
<td>Element</td>
<td>False</td>
</tr>
</tbody>
</table>

#### 3.5 Automation Tools

A collection of post-processing tools were developed to facilitate construction of Peregrine cases, extraction of both Peregrine and Falcon output data, and to automate the process of comparing the output data from the individual codes.

##### 3.5.1 Falcon to Peregrine Scripts

Python scripts were developed to convert Falcon input files into Peregrine input files. The driver for these scripts is that the original format of the proprietary models provided to EPRI for this Test Stand application are in the form of Falcon input files. The developed infrastructure allows the flexibility to perform simulations for hundreds of different rods. Furthermore, it ensures that matching Peregrine models could be developed once the Peregrine equivalent for each option/parameter used by Falcon is identified and documented. Development of this capability also aided in the understanding of the Peregrine input file structure and available options.

##### 3.5.2 Code-to-Code Comparisons

A collection of C++ and Python modules were developed in order to facilitate rapid post-processing, visualization, and analysis of hundreds of different models. Throughout the Test Stand, progression problems were executed and analyzed automatically as code updates were provided. The developed modules perform the following tasks:

1. Add *Postprocessors* to the Peregrine input file to create element-, node-, and point-based output fields which can be dumped to the CSV output file.
2. Extract Falcon output from the hierarchical data format (HDF) database and store this data in a generic format.
3. Extract Peregrine output from the CSV output file and store this data in a generic format.
4. Map Peregrine elements to Falcon elements and Peregrine nodes/points to Falcon nodes to ensure data comparisons are for similar physical locations.
5. Generate preset comparison plots for temperature, displacement, and hoop stress at select locations.

The resulting comparison plots are then documented and analyzed in iPython Notebooks. Lastly, a Git repository and accompanying Excel spreadsheet are used to organize, track, and store the relevant simulation data for future reference.
4 RESULTS

A fuel performance code is best validated with good post-irradiation examination (PIE) data that sufficiently covers the conditions that are expected to be modeled. These data should include separate effects testing as well as integral testing of parameters that are important indicators of satisfactory performance of fuel rods. Unfortunately, sufficient integrated fuel rod performance PIE data are not available to perform this assessment for the challenge problems that were studied. Furthermore, Test Stand activities are designed to focus on industry application of the software, not validation. Any validation would soon be obsolete and need to be repeated because of the changing configuration of models, solvers, and input structure experienced on account of the work being performed by both the MOOSE, BISON, and Peregrine development teams.

To assess the results delivered by Peregrine without integrated fuel rod performance PIE measurements, Falcon is being used as a benchmark fuel performance code. As stated previously, the intent of these benchmarks is not to compare absolute results, but to obtain an understanding of Peregrine capabilities by comparing results to a standard fuel performance modeling code in widespread use by the industry. The objective of these progression problem demonstrations is to evaluate Peregrine capabilities at this stage of development and to inform and prioritize future development efforts. Results from seven progression problems are presented in the following subsections. The first progression represents a shortened test rod with a flat axial power profile. In progression problems 2 through 6, a generic full-length PWR fuel rod is modeled with an increasingly complex applied power history. The final progression problem represents a full-length PWR model using Watts Bar Unit 1 fuel rod design and reactor operational data. These data were provided by permission from Westinghouse and the TVA and were used to test the readiness of Peregrine to analyze a typical PCI fuel performance case using plant operating data.

Because fuel performance results have both time and spatial dependence, and because both the Falcon and Peregrine codes produce output in different formats, most results are presented as a time series analysis for a single point in space or at specific points along a given geometric plane for a specified time. The locations along the axial mid-plane of the active fuel region where data were extracted for time-series analysis are indicated in Figure 18. From left to right, these points represent the fuel centerline, fuel radial mid-plane, fuel outer surface, cladding inner surface, and cladding outer surface.

Lastly, it is important to note that four different versions of VERA were used throughout the course of this Test Stand due to ongoing Peregrine development activities. Thus, results for progression problem 1 used the third version of VERA obtained by EPRI in April 2014 while results for progression problems 2 through 7 used the final version of the Peregrine code obtained in July 2014. For more information on the versions of VERA used, see Appendix B.5.1.

1 Note that fuel performance code comparisons are difficult because of differences in modeling and input parameter specification.
The number of cores and timing statistics for each progression problem are provided in Table 8. Note that significant differences exist in the amount of output provided by each case, which affects the computational time.

### Table 8. Computational Run Time Statistics for the Progression Problems

<table>
<thead>
<tr>
<th>Progression Problem</th>
<th># Cores</th>
<th>Wall Time (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16</td>
<td>24</td>
</tr>
<tr>
<td>2</td>
<td>16</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>16</td>
<td>50</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>59</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>177</td>
</tr>
<tr>
<td>6</td>
<td>16</td>
<td>49</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>54</td>
</tr>
</tbody>
</table>

#### 4.1 Progression Problem 1: Super-Ramp PK2-1 – Single Cycle Ramp

The first progression problem is a test reactor experimental fuel rod case derived from the Studsvik International Fuel Performance Experiments (IFPE)/Super-Ramp Project [31]. This test case represents a fuel rod (modeled as a single pellet) that is less than $1/10^{th}$ the size of a commercial PWR fuel rod. This case also includes further simplification in that the power history is applied uniformly along the axial length of the rod. A plot of the power history for this case is provided in Figure 19. The coolant boundary conditions were imposed using a uniform...
bulk coolant temperature history (Figure 20) with a specified heat transfer coefficient\(^2\). Note that the heat transfer coefficients

![Figure 19. Power History for Progression Problem 1](image)

\(^2\) It was recognized late in the review process that the heat transfer coefficients were significantly different in the Falcon and Peregrine input files. This is believed to be the cause of the discrepancies in Figure 25 and Figure 26, especially since the cladding temperatures calculated by both codes are much closer to each other in progression problems 2 through 7.
The following thermal results obtained from Peregrine are compared with Falcon results for a similarly modeled case. Note that since the imposed axial power profile is flat along the length of the rod, the results are similar at any radial location along the axial length of the rod that is sufficiently far away\(^3\) from the top and bottom of the fuel stack. To illustrate this point, Figure 21 plots the final temperature profile for progression problem 1.

\(^3\) Using Figure 21, “sufficiently far away” is determined to be approximately 4 cm along the fuel centerline and approximately 2 m along the fuel outer surface and cladding regions.
Figure 21. Final Temperature Profile for Progression Problem 1

Figure 22 plots the fuel centerline temperature as a function of time. Similar to the results presented in the reference [6], Falcon is generally predicting hotter fuel temperatures than Peregrine. This difference in temperature is less at the radial mid-plane in Figure 23 and is reversed at the outside of the fuel and at select cladding locations (reported in Figure 24 through Figure 26). Recall from Section 3.1.1 that the power history import is different between the codes. This is clear in Figure 22 at approximately 850 days (or anywhere where there is a significant change in power) where Falcon has a step change in centerline temperature and Peregrine changes gradually.

The power history used by the Peregrine development team for this case was misrepresented from its source, a Falcon case file. The original Falcon input was based on constant power input where changes to power from one time point to the next are quickly ramped using a 0.1-hr ramp time. Instead, linear changes were assumed for this Peregrine case\(^4\). This difference is apparent when comparing Peregrine results that show ramped temperature changes where Falcon results do not (Figure 22 through Figure 26).

In all cases, reasonable agreement is achieved between the codes if one accounts for the difference in power input noted above. A consistent difference in temperature is exhibited as the shape in temperature history is examined. The axial profiles at 1126 days into the simulated fuel exposure, shown in Figure 27 through Figure 28, are consistent with this assessment.

\(^4\) The power history input can be modified to mirror the 0.1-hour ramps used by Falcon, however the Peregrine solvers had difficulty finding converged solutions using this strategy and therefore this technique was not pursued further.
Figure 22. Fuel Centerline Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 1

Figure 23. Fuel Mid-Plane Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 1
Figure 24. Fuel Outer Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 1

Figure 25. Cladding Inner Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 1
Figure 26. Cladding Outer Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 1

Figure 27. Axial Temperature Profile along the Fuel Centerline at 1126 days for Progression Problem 1
4.2 Progression Problem 2: PCI Example Case – Single Cycle Ramp

Progression problem 2 uses the PCI Example case, which represents a generic PWR fuel rod model. For this progression problem, the simulated power history includes an initial ramp up to power followed by approximately 450 days of steady-state operation. This particular case is focused on an analysis of the thermal results for the approximately 460-day cycle. The power history for this progression problem is proved in Figure 29.
Figure 30 through Figure 32 illustrate temperature comparisons between Falcon and Peregrine for three locations along the axial mid-plane of the fuel: the centerline, radial mid-plane, and outer surface. Beginning with the fuel centerline temperature comparison in Figure 30, Falcon is predicting temperatures that are on average 30 K higher than Peregrine, and this difference increases as a function of time to a maximum difference of approximately 55 K after 460 days. Noting that the power level slowly increases with time, it appears that the temperature differences are consistent and proportional to the linear heat generation rate. This can be more easily observed in Figure 33, which plots the difference in fuel centerline temperature between the two codes alongside the linear heat generation rate.

When looking at the temperatures at the radial mid-plane of the fuel, which are plotted in Figure 31, a similar trend is observed where the Peregrine temperatures are lower than the Falcon temperatures. In this case however, the temperature differences are on average 10 K smaller. Moving to the fuel outer surface temperatures (Figure 32), the temperature differences drops to within 10 K of each other.

Figure 34 through Figure 36 show the cladding and coolant temperature histories for a point along the axial mid-plane of the active fuel region. A small difference is observed between the two codes, however this difference is indecipherable using Figure 34 through Figure 36. To observe how the temperature profiles compare axially along the cladding inside and outside surfaces, see Figure D-3 and Figure D-4 in Appendix D.

Differences in the calculated local fuel thermal conductivities between the two codes is one possible cause for the difference in the fuel centerline temperature observed. Another is a possible difference in the way local fission heat generation rates are determined. This could not be assessed at this time because these variables are not output as a function of time and space in the Peregrine simulations used for this analysis. Further research should be performed to
compare these values to verify whether or not there is a correlation between the difference in thermal conductivity or fission heat generation with the observed difference in fuel centerline temperature. The local burnup calculated by each code could also be used as an indicator of fission heat generation.

In order to gain more insight into the solutions obtained by the two codes, temperature profiles were plotted as a function of position in Figure 37 through Figure 39 using data from the final time step at the end of the simulation (approximately 460 days). Figure 37 illustrates more clearly the observation that fuel centerline temperatures are hotter in Falcon than in Peregrine. Furthermore, the most important difference illustrated by Figure 37 is the difference in the temperature gradient across the fuel. Given that both codes are using the same power history and axial shape history, inspection of the conduction equation for Cartesian and cylindrical coordinates below:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial z^2} + \frac{\dot{s}}{\lambda} = \frac{\rho c_p}{\lambda} \frac{\partial T}{\partial t}$$

Equation 1

or in cylindrical coordinates,

$$\frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial T}{\partial r} \right) + \frac{\partial^2 T}{\partial z^2} + \frac{\dot{s}}{\lambda} = \frac{\rho c_p}{\lambda} \frac{\partial T}{\partial t}$$

Equation 2

shows the heat generation rate ($\dot{S}$) in W/m$^3$ as a function of thermal conductivity ($\lambda$) in W/(m-K), density ($\rho$) in kg/m$^3$, heat capacity ($c_p$) in J/(kg-K), and distance ($r, z$ or $x$) in meters. This suggests that differences between the codes are either related to geometry (i.e. the gap distances are different, changing the heat generation rate), property differences (the thermal expansion coefficient, heat capacity or thermal conductivity may differ as a function of temperature, or the densification models may be different), or localized power/burnup differences (which ultimately affect the heat generation rate). One known difference between the codes is that Falcon assumes steady-state, which sets the transient partial derivative, $\frac{\rho c_p}{\lambda} \frac{\partial T}{\partial t}$, equal to zero, while Peregrine maintains the transient partial as non-zero. Although this transient value is relatively small, it would have an effect on the results during fast transients and less during steady state or normal operational power changes.

This result suggests that more heat is being generated from the fuel at this elevation in the Peregrine case relative to the Falcon case because the cladding and coolant temperatures are in such good agreement. Figure 38 shows the fuel centerline temperature as a function of axial position at the end of the simulation. From these results an initial conclusion could be that there is a slight difference in the axial shape for the two codes as the fuel centerline temperature differences appears greater in the low region of the fuel rod compared to the upper region. However, Figure 39 provides more insight into the differences in the solutions produced by both codes. Along the fuel outer surface, a significant difference in the temperature in observed along the upper region of the fuel rod. These observed differences are likely a direct result of the coupled mechanical response where the fuel-cladding gap is closing faster for Falcon than in the Peregrine case. Even though the power level is higher in this section, a closed gap means better heat transfer and lower fuel outside surface temperature. Displaced model plots, in Figure 40 and Figure 41, illustrate the gap width for the final time step for Falcon and Peregrine, respectively.

Note that the axial shape near the end of the simulation also shows a higher power in the upper segment of the fuel rod relative to the lower segment.
The Falcon result in Figure 40 shows a closed gap in the upper region of the fuel rod. This allows for better heat transfer between the fuel and cladding in the Falcon case for the upper segment of the fuel rod. Figure 41 illustrates an unclosed gap for the Peregrine simulation that translates to a buildup of heat within the upper region of the fuel (Figure 39), which is alleviated in the Falcon case due to gap closure.

Thus it can be concluded that a different solution with respect to the axial temperature profile is being obtained by the two codes, and direct comparisons at a given position are only comparable to the extent that they allude to the difference in the solutions for the two codes. It is important to note that these differences are acceptable with the understanding that the nonlinearity associated with the solution complicates code comparisons. It is also important to realize that, in this progression problem, the linear heat generation rates used as input were calculated without feedback or coupling to the detailed fuel rod behavior simulated by either code. The reduction in temperature is a result of heat conduction differences and may impact the fission rates in ways that may be unaccounted for. This is a source of uncertainty that is beyond the scope of this Test Stand but is a phenomenon that could be better studied when Peregrine is coupled with both a reactor physics code and a thermal hydraulics code.

Figure 30. Fuel Centerline Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 2
Figure 31. Fuel Mid-Plane Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 2

Figure 32. Fuel Outer Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 2
Figure 33. Difference in the Fuel Centerline Temperatures Plotted in Figure 30 for Progression Problem 2.

Figure 34. Cladding Inner Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 2.
Figure 35. Cladding Outer Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 2

Figure 36. Coolant Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 2
Figure 37. Temperature as a Function of Radial Position Taken at Points along the Axial Mid-Plane of the Active Fuel Region at 460 days for Progression Problem 2

Figure 38. Axial Temperature Profile along the Fuel Centerline at 460 days for Progression Problem 2
Figure 39. Axial Temperature Profile along the Fuel Outer Surface at 460 days for Progression Problem 2

Figure 40. Gap Thickness Predicted by Falcon for the Final Time Step (~460 days) for Progression Problem 2
4.3 Progression Problem 3: PCI Example Case – Increased Power

Progression problem 3 extends the analysis presented in the previous progression problem to investigate gap closure and cladding hoop stress. Because gap closure is not achieved by Peregrine within the first cycle given the power history presented for progression problem 2, the power history data are scaled up by a factor of 1.2 for progression problem 3. This results in gap closure being achieved within the first cycle.

Analysis of the temperature as a function of time for the six relevant locations along the axial mid-plane (fuel centerline, mid-plane, and outer surface as well as the cladding inner and outer surfaces and the coolant) yields results and trends that are quite similar to progression problem 2. These plots are available in Appendix D.3. In general, Peregrine is estimating cooler temperatures than Falcon along the fuel centerline, and this trend dissipates as one moves radially towards the fuel outer surface. Temperatures throughout the cladding and coolant for the two codes are again comparing quite well. A plot of the temperature as a function of radial position along the axial mid-plane is provided in Figure 42. This plot illustrates the difference in fuel temperatures throughout the fuel along the axial mid-plane and is similar to what is observed in Figure 37 for progression problem 2. Figure 43 and Figure 44 plot the temperature profiles as a function of axial position for the fuel centerline and fuel outer surface, respectively. The primary difference here relative to what is observed in Figure 38 and Figure 39 for progression problem 2 is that the temperature profiles (for progression problem 3) have the same shape. This is because gap closure has been achieved in both codes for the time step being plotted, which was not the case for progression problem 2.

The important phenomena being examined by this progression problem is the fuel-clad gap width, which is plotted as a function of time in Figure 45. This result shows a difference in the
gap width of approximately 8 to 10 microns. Furthermore, this difference appears once the reactor comes up to power and then persists up until the gap is closed, which in the Falcon simulation is approximately 80 days before the gap closes in the Peregrine case.

To investigate this difference further, the gap thickness results over the first five days of the simulation are re-plotted in Figure 46. This helps to illustrate the similarities and differences in the solutions obtained by both codes. From Figure 46, the difference in the solutions appears to occur between 2 and 4 days into the simulation. Peregrine finds a solution showing a larger gap thickness at a time before Falcon shows a significant decrease in gap thickness. Both Peregrine and Falcon show about the same size gap when Falcon first shows this decrease. However, the change in gap thickness beyond this point is greater for Falcon until both codes show a constant gap thickness just before the 4 day mark. This more rapid change in gap thickness exhibited by Falcon results in a smaller gap for Falcon from this point on when compared to the Peregrine result.

Figure 47 plots the temperatures and displacements for the fuel outer surface and cladding inner surface over the first 5 days of the simulation. The observable temperature differences are that the cladding temperatures are in acceptable agreement while the fuel temperature in Peregrine exceeds the Falcon estimate. Comparing the radial displacements, both the fuel and the cladding differ by an initial offset. The position of the cladding and the fuel at the start of the simulation should be the same because this is by definition the initial start conditions at hot zero power. Ramping up from cold zero power to hot zero power is automatically performed within Falcon and manually performed by the user for Peregrine.

Following the initial condition, the fuel expands more in Falcon than in Peregrine. Given the higher fuel temperatures estimated by Falcon in Figure 42, more expansion in the Falcon simulation should be expected; however, it cannot be determined whether or not this is the only contributor to the observed differences in the radial displacements between the two codes because densification, relocation, smeared cracking, and swelling are all playing a significant role in the early stages of the simulations. Brief supplemental analyses of relocation and smeared cracking models are available in Appendix C.
Figure 42. Temperature as a Function of Radial Position Taken at Points along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 3

Figure 43. Axial Temperature Profile along the Fuel Centerline for Progression Problem 3
Figure 44. Axial Temperature Profile along the Fuel Outer Surface for Progression Problem 3

Figure 45. Gap Thickness at the Axial Mid-Plane of the Active Fuel Region for Progression Problem 3
Figure 46. Gap Thickness at the Axial Mid-Plane of the Active Fuel Region for Progression Problem 3 Plotted for the First 5 Days of the Simulation Only

Figure 47. Temperatures (left) and Displacements (right) for the Fuel Outer Surface and Cladding Inner Surface Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 3
4.4 Progression Problem 4: PCI Example Case – Down Power

Progression problem 4 focuses on observing the thermo-mechanical solution, primarily in the form of gap reopening, as the simulation comes down from a steady-state power of approximately 7 kW/ft to zero power. Noting that this is simply a continuation of progression problem 3, all results given in this section will begin 463 days into the simulation and end as soon as zero power is reached, which is after approximately 464.3 days. The power history for this time period is presented in Figure 48.

![Figure 48. Power History for Progression Problem 4](image)

Figure 48 illustrates the fuel centerline temperature profile as the reactor begins to come down in power. Once the ramp down begins, both codes successfully ramp down the temperatures accordingly. A noteworthy observation from Figure 49 is that the rate of temperature decrease in Peregrine is slower than the rate observed in Falcon. Furthermore, the two codes have similar temperatures at the point in which zero power is reached.

Figure 50 plots the fuel outer surface temperatures for both models. Again it is observed that the temperature profiles decrease at different rates but converge on approximately the same temperature. Figure 51 plots the difference in temperature between Falcon and Peregrine for three different locations along the axial mid-plane of the active fuel regions. Figure 51 verifies the models converging on the same temperature. Note that the cladding and coolant temperatures for the time frame being examined by progression problem 4 are all within 2 K of each other, and are thus not examined in depth for this progression problem.

The mechanical response of the ramp down period is shown in Figure 52. When the reactor power begins to ramp down, both codes reopen the fuel-cladding gap. However, Falcon estimates a faster rate than Peregrine as there is a larger gap opening. Figure 53 plots the fuel outer surface and cladding inner surface radial displacements. This result shows the fuel and the
cladding contracting faster in the radial direction for the Falcon case during the ramp down period. In the case of the fuel, this rate is fast enough to create the larger gap observed in Figure 52 relative to Peregrine. Because of this faster rate, the gap width in Peregrine at the time when zero power is reached is approximately 1.4 microns smaller than Falcon. In other words, the larger reduction in fuel outer radius combined with an offset in the cladding inside radius determined by Falcon leads to a smaller gap opening estimated by Peregrine at zero power.

![Figure 49. Fuel Centerline Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 4](image)

Figure 49. Fuel Centerline Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 4
Figure 50. Fuel Outer Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 4

Figure 51. Temperature Difference between Falcon and Peregrine for 3 Different Location along the Fuel Axial Mid-Plane for Progression Problem 4
Figure 52. Gap Thickness at the Axial Mid-Plane of the Active Fuel Region for Progression Problem 4

Figure 53. Radial Displacements for the Fuel Outer Surface (left) and Cladding Inner Surface (right) Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 4
4.5 Progression Problem 5: PCI Example Case – Second Cycle Restart without Down Power

Progression problem 5 is similar to progression problem 3 but appends the second cycle restart directly to the steady-state first cycle power history. There is no down power period for this progression problem; the power history is monotonically increasing throughout the simulation. The power history for progression problem 5 for the appended ramp period is provided in Figure 54. Note that this power ramp consists of 4 distinct features. Using the annotation provided in Figure 54, segment $\overline{AB}$ represents a small but noticeable power ramp, segment $\overline{BC}$ represents a sharp power ramp, segment $\overline{CD}$ represents a short-duration hold, and segment $\overline{DE}$ represents a second sharp power ramp. These four segments will be used in the ensuing discussion to more easily clarify which region of the power ramp is being discussed.

This progression problem is used to analyze the buildup of hoop stress on the inside cladding surface as the fuel rod undergoes an additional ramp period. Given that this is an extension of progression problem 3 and that the first 460 days of the simulation are exactly the same, it may be concluded from Figure 45 that this secondary ramp period is occurring after the gap has closed in both the Falcon and Peregrine simulations.

![Figure 54. Power History for Progression Problem 5](image)

Figure 54 plots the fuel centerline temperature as a function of time during the subsequent ramp period for progression problem 5. As with the previous progression problems, similar trends were observed in both Falcon and Peregrine with Falcon predicting hotter temperatures than Peregrine. The fuel outer surface temperature plotted in Figure 56 for both Falcon and Peregrine shows some interesting differences in both the magnitudes and predicted changes that occur during the ramp in power by the two codes. Similar to the earlier progression problems for the PCI Example case (progression problems 2 through 4), Falcon predicts fuel outer surface temperatures that are cooler than what Peregrine is predicting. During segment $\overline{AB}$, the
temperature change predicted by Falcon is slower in response than Peregrine. This indicates that heat generated in the fuel is not being removed as fast through the gap in the Peregrine case at this point relative to the Falcon case. During the subsequent sharp power ramp (segment $BC$), intermediate hold period (segment $CD$), and second sharp power ramp (segment $DE$), the temperature changes seem to be quite similar for the two codes. Once the appended power ramp is complete, a difference in the outer fuel temperature response is again observed. Peregrine seems to have reached a temperature where further increase is slow while Falcon is at a temperature where a more rapid rise in temperature is occurring. Furthermore, once the ramp is complete, the Falcon temperature continues increasing at a nonlinear rate while the Peregrine fuel outer surface temperature increases linearly.

Reviewing the mechanical solution, Figure 57 plots the fuel outer surface and cladding inner surface radial displacements as a function of time during the second cycle ramp period. The interesting aspect of the results presented in Figure 57 is that the Peregrine displacement is flat and unchanging after the ramp period ends whereas the Falcon results show a slight but steady decrease in the fuel radial displacement and a slight but steady increase in the cladding displacement. Because the gap remains closed in the Falcon case, it can be concluded that Falcon is modeling a change in fuel response to the high cladding stress not accounted for in Peregrine. The decrease in fuel radius calculated by Falcon is equivalent to the decrease in cladding inside radius. The fuel-cladding interface that seems to be moving inward in response to the stress exhibited on the fuel by the cladding is not observed in the Peregrine results. This change may explain the parabolic temperature change during the final steady state power hold exhibited by Falcon in Figure 56. This may be related to the fact that Falcon is calculating higher temperatures.

The hoop stress comparison is plotted in Figure 58. From this result illustrates that both codes predict stress buildup and relaxation at the appropriate times with respect to changes in the power history. The notable differences between the estimates of the two codes are the difference in the rate of stress relaxation following each of the two sharp power ramps (segment $BC$ and segment $DE$) and the difference in the magnitude of the stress buildup during the second sharp power ramp (segment $DE$) of the simulated operating history. In addition, after the first sharp ramp (segment $BC$), Peregrine predicts a linear relaxation in the stress level while the stress relaxation predicted by Falcon appears nonlinear. Following the second sharp ramp (segment $DE$), the stress relaxation profile predicted by both codes is nonlinear, with Falcon being the more pronounced of the two.

During the second sharp power ramp (segment $DE$), the change in hoop stress for Falcon is approximately 70 MPa while the increase in the hoop stress calculated by Peregrine is approximately 115 MPa. It should be expected that the cladding would relax more in the Peregrine case because of this larger change, however the Falcon case exhibits a more significant relaxation in cladding hoop stress. This may be explained in connection with the fuel-cladding interface’s inward movement that seems to be in response to the high stress that would be expected on the fuel. Future investigation is needed to examine which components within both the fuel and the cladding are contributing to the faster cladding inside surface hoop stress relaxation.

The rate of change in hoop stress may be calculated to observe the stress buildup and relaxation occurring at each time step. This stress buildup and relaxation is examined for the Falcon case in
the 463- to 469-day range in Figure 59. This result shows how Falcon’s use of a fixed numbers of thermal, mechanical and time step solution iterations, to favor faster runtimes, may lead to convergence artifacts for some solution steps where stress/power are at high levels. Falcon has not implemented a true convergence criteria because experience has shown good results have been achieved even under transient conditions.

Figure 55. Fuel Centerline Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 5
Figure 56. Fuel Outer Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 5

Figure 57. Fuel Outer Surface and Cladding Inner Surface Radial Displacements Taken at a Point along the Axial Centerline of the Active Fuel Region for Progression Problem 5
Figure 58. Cladding Inside Surface Hoop Stress Taken at a Point along the Axial Centerline of the Active Fuel Region for Progression Problem 5

Figure 59. Change in Cladding Inside Surface Hoop Stress at a Point along the Axial Centerline of the Active Fuel Region for Progression Problem 5
4.6 Progression Problem 6: PCI Example Case – Second Cycle Restart with Down Power

The final progression problem utilizing the PCI Example model involves the simulation of a complete first cycle, ramp down to zero power, and a subsequent second cycle ramp. The power history for this progression problem is provided in Figure 60. Note that the same line segment notation used in progression problem 5 will be used in progression problem 6 to denote the different regions of the power ramp.\(^6\)

![Figure 60. Power History for Progression Problem 6](image)

Figure 61 is a plot of the fuel centerline temperature for progression problem 6. Similar to the previous progression problems, the trends in the fuel centerline temperature throughout the ramp down and second cycle restart for both Falcon and Peregrine show good agreement. The magnitudes, as in the previous progression problems, differ significantly. Figure 62 plots the difference between the Falcon and Peregrine temperature estimates by subtracting the Peregrine temperature estimate from the Falcon temperature estimate. Figure 62 illustrates that the magnitude of the difference in temperature has a strong dependence on the magnitude of the temperature which is reflective of the power history. This result requires further evaluation to obtain a better understanding of what is causing this temperature difference and why it appears to be correlated to the power history.

The fuel radial mid-plane temperatures, illustrated in Figure 63, show results similar to those presented for the fuel centerline temperatures. The fuel outer surface temperature, illustrated in

\(^6\) Note that the time stepping strategy used for progression problems 2 through 5 for the PCI Example case could not be used for this progression problem due to convergence issues experienced by the solver during the ramp down to zero power. Thus fewer data points are available to represent the Peregrine results for progression problem 6 relative to the other progression problems.
Figure 64, shows a difference in the trend during the second cycle restart and subsequent hold period. Beginning near the end of the second cycle power ramp (segment $DE$), the trends separate as Falcon, starting at a lower temperature, predicts a larger increase in temperature as a function of time until approximately day 473. This can be more clearly observed by Figure 65, which plots the difference in the fuel outer surface temperatures calculated by Falcon and Peregrine. When compared to Figure 62, the dependence on the magnitude of the temperature, or power, is not as strongly pronounced. It would seem that the influence of heat transferred from the cladding to the coolant has a large competing effect on determining the temperature response on the outside fuel surface at these lower temperatures. Additional thermal results illustrating the cladding and coolant temperature comparisons are available in Appendix D.6. Note that similar to progression problems 2 through 5, agreement is observed between the cladding and coolant temperature results between the two codes.

Figure 66 plots the temperatures and displacements for the fuel outer surface and cladding inner surface during the ramp down and subsequent restart. This result shows good agreement between the two codes during the down power and second cycle restart, but also shows that Falcon estimates increasing fuel and cladding displacements throughout the final hold period while Peregrine does not estimate swelling of the fuel rod during this final hold period. This observed difference in the mechanical solution is likely the driver behind the difference in the thermal trends for this time period. It is important to note that this difference in trends does not necessarily suggest an inconsistency between the codes. Falcon is predicting temperatures that are approximately 140 K higher during this hold period, and this difference in magnitude could be responsible for variation in this and other various physical phenomena calculated by the two codes.

For progression problem 6, the inside cladding hoop stress is plotted in Figure 67. The notable observations from Figure 67 are that Falcon predicts more stress relaxation during the down power and during the first part of the second cycle restart while Peregrine shows more stress relaxation during the final hold period after the second cycle ramp is complete. This differs from progression problem 5 observations where the power was not reduced to zero before the second cycle ramp, as shown in Figure 58. Additionally, in progression problem 6, Peregrine predicts less stress buildup during the first ramp (segment $BC$) before the short hold, and more stress buildup during the second ramp (segment $DE$) to terminal power. Despite the various thermal and mechanical differences, the peak inside cladding hoop stress predicted by Peregrine is approximately 0.4% less than that predicted by Falcon.
Figure 61. Fuel Centerline Temperature for the First Cycle (left) and the Down Power and Subsequent Second Cycle Restart (right) Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 6

Figure 62. Fuel Centerline Temperature Difference between Falcon and Peregrine for Progression Problem 6
Figure 63. Fuel Mid-Plane Temperature for the First Cycle (left) and the Down Power and Subsequent Second Cycle Restart (right) Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 6

Figure 64. Fuel Outer Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 6
Figure 65. Fuel Outer Surface Temperature Difference between Falcon and Peregrine for Progression Problem 6

Figure 66. Temperatures (left) and Displacements (right) for the Fuel Outer Surface and Cladding Inner Surface Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 6
Figure 67. Cladding Inside Surface Hoop Stress Taken at a Point along the Axial Centerline of the Active Fuel Region for Progression Problem 6

4.7 Progression Problem 7: Watts Bar - Multi-Cycle Analysis

Data from the Tennessee Valley Authority’s (TVA) Watts Bar Nuclear Plant [32] were used as input for progression problem 7. Watts Bar Nuclear Unit 1 (WBN1) is a Westinghouse-designed 4-loop pressurized water reactor (PWR) that uses 17×17 fuel assemblies [33]. A horizontal cross-section of a Westinghouse 17×17 fuel assembly is provided in Figure 68.

The power history for progression problem 7 is presented in Figure 69. This progression problem is similar to progression problem 6 in that the simulation includes a ramp up to steady-state operation followed by a period of down power and a second-cycle ramp up to steady-state. The differences are that the period of zero power spans a few days in this progression problem and that the power ramps are not monotonic as they were in progression problem 6.
Figure 68. Horizontal Cross Section of a Westinghouse 17×17 Fuel Assembly (Guide/Instrument Tubes are Shown in Blue)

Figure 69. Power History for Progression Problem 7 [16], [17], [18], [19]
The fuel centerline temperature provided in Figure 70 shows the same trends observed in the previous progression problems; a hotter temperature estimate from Falcon with better agreement between the two codes during the up and down ramps. A notable point from this and all of the figures to follow for progression problem 7 is that the hold at zero power causes some stability issues within the Peregrine solver, and thus the problem was not able to run to completion following this extended period of down power. Figure 71 illustrates the fuel outer surface temperature and is provided to illustrate that the temperature profile on the outside of the fuel also exhibits similar behavior in comparison to the other progression problems.

Figure 72 and Figure 73 illustrate the gap thickness and hoop stress results for progression problem 7. Similar to the previous progression problems, gap closure is occurring later in the Peregrine simulation relative to the Falcon simulation, and the hoop stress results reflect this difference.

Figure 70. Fuel Centerline Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 7
Figure 71. Fuel Outer Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 7

Figure 72. Gap Thickness at the Axial Mid-Plane of the Active Fuel Region for Progression Problem 7
Figure 73. Hoop Stress on the Cladding Inside Surface Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 7
CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

The Peregrine code and its underlying framework provide a platform for coupling the thermal, mechanical, and chemical physics models required to perform fuel performance calculations. This framework has a vast potential for solving multi-physics simulations in 2D and 3D space and shows few limitations on what it can do with respect to finite element simulations. Consequently, the input file for Peregrine includes many modeling options that require user iteration and testing to determine which set of inputs is optimal for a particular model and operating history. The input file structure and capabilities need to be well understood and documented in order to be integrated into the VERA common input. Furthermore, the underlying models need to be documented and well understood before they are integrated with other VERA components.

EPRI has evaluated the capabilities of the Peregrine fuel performance code using a series of progression problems developed specifically for this Test Stand application and modeled in two-dimensional R-Z axisymmetric space. This evaluation was performed by making comparisons between Peregrine and EPRI's fuel performance code Falcon. Analysis of these progression problems yielded the following conclusions:

- The thermal behavior within the fuel is different during the initial startup between the two codes. It appears that more energy in the form of heat exists in the fuel during Falcon simulations relative to the Peregrine simulations considering all temperature histories and axial temperature distributions examined by this Test Stand. This could be the result of numerous differences that exist between the two codes. Two areas identified as having a higher likelihood of exposing the reasons behind the observed differences in the results include (1) the volumetric heat generation calculation and (2) the $\text{UO}_2$ thermal conductivity model.

- The thermal and mechanical response trends observed by both codes are quite similar, although they vary in numerical result. This is important because the Falcon model has been validated by several sets of experimental data, and Peregrine should produce near equivalent results.

- There are no outstanding large differences in Peregrine results when compared to Falcon, suggesting that Peregrine is ready for a more thorough verification program.

It is important to note that multi-physics code comparisons that involving dozens of different models and parameters are difficult. Significant effort went into making the Falcon and Peregrine models comparable; however, this cannot be rigorously enforced to ensure a true “like for like” comparison given the differences in the solution methodology, finite element formulation, and the treatment of the various thermal, mechanical, and chemical phenomena. It is important to recognize that definitive assessments and conclusions should not be derived using only code-to-code comparisons. Data validation of the Peregrine results is needed to accomplish this task.

An important conclusion derived from this work regarding Test Stands is that user support is necessary for the success of Test Stand applications, especially when the software application is under significant development during the Test Stand period. The various technical and administrative meetings as well as the travel opportunities that allowed members of the Test
Stand team to interact directly with developers and support staff were key factors in the success of this Test Stand. Having a good support system where users and developers can talk candidly and frequently is key to maximizing Test Stand success and efficiency.

Note that additional technical and administrative feedback is provided in Appendix B.

5.2 Recommendations for Future Work

There are several technical and programmatic recommendations listed throughout this document with regards to the development of both VERA and Peregrine. Select technical recommendations are emphasized below to motivate continued study comparing Peregrine with Falcon and other fuel performance models.

- Review the integrated power between codes—Overall the results were in good agreement between Falcon and Peregrine; however, a detailed comparison of the integrated power may help diagnose the causes for fuel centerline temperature discrepancies.

- Compare differences between smeared crack and relocation model—The Peregrine smeared crack model is still in development and not recommended for use during the Test Stand, while Falcon analyses do employ the Falcon smeared cracking model. After the smeared crack model is completed in Peregrine, the results should be compared again to check equivalency.

- Relocation model analysis and comparison—The relocation model analysis presented in Appendix C.1 should be revisited in conjunction with Peregrine’s smeared cracking model to ensure that reasonable model behavior is observed when both models are active.

- Material property comparisons—Peregrine uses different models for heat capacity and other properties. It is recommended to compare the effects of the material properties on the overall results.

- Steady state vs. transient heat conduction—Peregrine by default applies the transient heat conduction equation where Falcon assumes steady-state. It is believed the transient effects are small, but the option should be explored.

It is also recommended that extensive validation against measured data be performed using Peregrine to validate the various physics models both independently and collectively. Prior to this Test Stand, fuel centerline temperatures and select mechanical behavior were validated using several sets of test rod experiment data [13]. Given the significant development performed over the course of this Test Stand on fission gas release, plane strain, and pellet-cladding contact models, these validation activities should be repeated. A continuous validation process should be developed in order to ensure that VERA maintains a validated version of Peregrine that meets the VERA quality assurance (QA) plan. These validation cases should include full-length rods operated under typical LWR conditions with multiple changes in power. Because measured data are difficult to obtain, it is recommended that validation cases be included in a test suite to ensure that Peregrine can run to completion using models that are fairly representative of models that would be constructed to perform PCI analyses. If measured data are unavailable, these cases could instead be analyzed from a base physics standpoint to ensure that various physical phenomena are behaving as designed. One such example is verifying that all of the energy generated via the applied power history is conserved, and mass continuity is maintained.
REFERENCES


Appendix A: Falcon

The Falcon Fuel Rod Performance Code, Version 1.2, is a combined steady-state and transient thermal/mechanical finite element (FE) code for analyzing light water reactor fuel behavior [24]. The modeling approach employed by Falcon can analyze both normal operation and accident conditions for fuel rod average burn-up levels approaching 80 gigawatt days per metric ton of uranium (GWd/tU).

The nuclear industry uses fuel within the applicable regulatory limits, that is, the specified acceptable fuel design limits. Core designs and operational conditions are analyzed with fuel performance codes to verify that these limits are observed. Running these codes requires significant expertise in structural analysis, neutronics, fluid flow, heat transfer, numerical analysis, and material mechanics and chemistry in nuclear environments. Falcon is a state-of-the-art, best-estimate fuel performance analysis code that is designed to address these technical requirements and broaden the accessibility of its computational capabilities to an audience interested in fuel performance. Falcon objectives are:

- To provide a robust and vendor-independent best-estimate code that is validated to high burn-ups to support fuel performance analysis, reload designs, and licensing support activities
- To make typical fuel performance indicators for licensing and operating margins more accessible to users with varying levels of expertise—from beginners interested in simply learning fundamental fuel performance basics to advanced users who want to perform core-wide evaluations
- To provide a tool to quickly generate R-θ pellet cladding mechanical interaction (PCMI) local-effects cases for evaluating duty-related loading margins

Falcon is a two-dimensional FE analysis code for modeling fuel rods as an axisymmetric structure in R-Z space or as a cross-sectional slice in R-θ coordinates space. A fully coupled thermal and mechanical solution is used to solve both steady-state and transient analyses. The robust mechanical solution allows Falcon to compute large cladding deformations such as wall thinning and ballooning as seen in loss of coolant accidents (LOCAs). The ability to utilize R-θ coordinate-based models also allows Falcon to perform very detailed local-effects analyses for essential cladding failure determinations.

Falcon computes a large number of parameters to support licensing and fuel performance evaluations. Detailed information is available that comprehensively describes the thermal, mechanical, and chemical condition of the fuel rod at any time during the analysis. Falcon also has the capability to calculate parameters that can be used to evaluate the potential for cladding failure caused by a combination of thermal and mechanical forces.

For pellet cladding interaction and LOCA conditions, cladding failure potential can be assessed based on cladding stresses or a time-temperature-stress failure approach using the cumulative damage concept. For mechanical fracturing by PCMI, Falcon provides a critical limit state approach based on the cladding strain energy density for cladding failure evaluation.

Falcon features a graphical user interface (GUI) pre-processor that leverages an efficient, layered input file scheme to reduce problem setup complexity for analyses of typical pressurized water

CASL-U-2014-0121-000-a
reactor and boiling water reactor fuel rod designs, materials, and boundary conditions. Solver results for a large number of parameters are written to an industry standard HDF5 database file for each calculated time step. A GUI post-processor can read these results and offer simple ways to review these time and spatially dependent results in tabular and plotted forms. The post-processor offers the capability to automatically transfer results and design data from an R-Z model with power changes that may affect PCMI stress to R-0 local-effects models where implicit radial cracks or missing pellet surface conditions can be analyzed.

The ability to obtain the detailed thermal and mechanical response of commercial fuel rods in a nuclear environment is desirable to researchers, vendors, and nuclear utility fuel managers. The proven ability of Falcon to calculate best-estimate steady-state and transient fuel performance results is already well regarded by the industry, and the latest revision of the software leverages a modern interface and back-end design to realize a configurable framework.
Appendix B: Test Stand Feedback

This appendix includes detailed feedback on the installation and testing of VERA as well as the input specification, output processing, documentation, and the process of running Peregrine.

B.1 Feedback on Installation and the Installation Guide

Installation of the VERA DEV 3.3 software was performed on Phoebe in October 2013 using the installation guide provided by CASL, which was based on the previous Test Stand launched by Westinghouse in June 2013 [34]. Many of the general administrative tasks such as accessing the code repositories on casl-dev.ornl.gov and setting up the VERA directory structure and build environment were greatly simplified by this installation guide. The scripts provided for installing the Linux-based build tools and third-party libraries were also helpful in expediting the overall build process.

Phoebe was a new cluster prior to the installation of VERA. The version of the initial Red Hat distribution (RHEL 6.4) did not contain all of the required packages for performing the VERA installation. The following additional software packages were required because they were not part of the RHEL 6.4 general installation:

- GNU C development libraries (glibc),
- GNU C development libraries for GTK+ (glib2),
- SVN, and
- X11 development libraries.

It is advised that the required packages that are not a part of the general installation for Red Hat be listed explicitly in the installation guide. This strategy can also be extended to accommodate additional Linux distributions and/or operating systems as they become supported.

Additional minor issues were encountered when installing the SILO and QT libraries using the provided build scripts as these installations initially failed. In the case of the SILO installation, the make process was searching for the path to szlib even though the `--without-szlib` flag was used in the `silo.sh` file. The problem was corrected by first using an arbitrary directory `--with-szlib=//some/directory` in the initial configuration and then running configure a second time without specifying the `--with-szlib` or `--without-szlib` flags. In the case of the QT installation, adding `--no-webkit` to the QT_COMMON_OPTIONS environment variable was required in order to allow the installation to complete successfully.

It is also possible that there may be a conflict between the system QT libraries and the QT libraries being built for VERA. In the case of Red Hat, QT3 is a part of the base distribution, and the environment variables QTDIR, QTLIB, and QTINC are predefined in the root environment. Therefore one may consider the suggested addition provided in Figure B-1 to the VERA environment. Furthermore, noting that situations may arise where certain TPLs other than QT are
part of a given distribution, additional logic may need to be added to the provided build scripts in order to account for conflicting system library installations\textsuperscript{1}.

\begin{verbatim}
unset QTDIR
unset QTLIB
unset QTINC
\end{verbatim}

Figure B-1. Suggested Addition to the VERA Environment Scripts to Account for the Presence of a System QT Installation

Another issue was encountered during the VERA installation that, while specific to Phoebe, may come up in future installations. Because Peregrine requires an in-source build, the source tree must be copied to the build tree to avoid contaminating the source tree for additional compilations. This increased the effort required for configuration on Phoebe because compilation needed to be performed on a compute node while the source code was located on a storage node. Thus the process of copying the source tree over to the build tree when running an interactive job via the Torque queuing system required performing a synchronization over the network, which can burden the network and often timed out. This requires the process to be started over from the beginning since no restart capability is available on account of the mixed build system. Advice on how to deal with these issues should be provided in the installation guide.

For a standalone (modular) Peregrine installation, an additional build script is required to place the `CASL_MOOSE_Peregrine` executable, `libcasl` library, and header files in the VERA install path. It is recommended that this process should be included as an automated part of the installation process that is already in place.

B.2 Feedback on Testing

The unit and verification tests used by the Peregrine development team, as well as the tests for the underlying framework, are not integrated into the automated VERA test suite. It is recommended that these tests be integrated into the VERA test suite so that a verified version of VERA is maintained that meets the VERA quality assurance (QA) plan.

Continuous validation testing is also needed to ensure that Peregrine performs as expected during the ongoing development process. This includes developing a continuous testing suite that executes a series of validation cases to check the output parameters important to fuel performance modeling, such as those associated with PIE measurements. Examples of validation parameters include temperature, gap thickness, diametral strain, fission gas release, oxidation, hydrogen content, and axial elongation.

It is suggested that Peregrine utilize development tools to keep track of unit test statistics such as code coverage. A mechanism to include EPRI Test Stand cases should also be considered so that the models and features exercised by these cases can be easily used and retested in future versions of Peregrine. However, this would require scripts to modify the input decks as they are under development.

\textsuperscript{1} For example, if HDF5 is already installed on a given system via the package manager, then building MOAB will result in libraries that are linked with the system HDF5 libraries even though HDF5 support through MOAB is not required for the VERA installation, and this may result in conflicts with the version of HDF5 built for and supported by VERA.
B.3 Feedback on Input and Pre-Processing

This subsection details notable observations and recommendations regarding the Peregrine input file structure.

B.3.1 Unit System Convention

Many units used by Peregrine are not representative of units typically used by the nuclear utility industry. Select examples of the differences in units used by Peregrine and those used by industry are provided in Table B-1. It is recommended that a unit conversion system be developed and provided for running standalone Peregrine calculations. Furthermore, local and rod averaged burnup should be available in units of MWd/kgU or MWd/tU since burnup is commonly used as an independent variable for simulation analysis and since all of the information required to calculate burnup using either of these unit conventions is available within Peregrine.

Table B-1. Unit Comparison between Peregrine and Typical Industry Units

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Industry</th>
<th>Peregrine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>in, ft, cm, micron</td>
<td>m</td>
</tr>
<tr>
<td>Time</td>
<td>hr, day</td>
<td>s</td>
</tr>
<tr>
<td>Pressure</td>
<td>psi, MPa</td>
<td>Pa</td>
</tr>
<tr>
<td>Flux</td>
<td>n/cm²-s</td>
<td>n/m²-s</td>
</tr>
<tr>
<td>Burnup</td>
<td>MWd/tU, MWd/kgU</td>
<td>FIMA²</td>
</tr>
</tbody>
</table>

B.3.2 Pressure Ramp

Peregrine is typically initialized with a zero pressure that is then ramped up to system pressure at the start of the simulation. It is recommended that the initial (cold) pressure should instead be set equal to the initial rod internal pressure. Although this is set by the user in the input file, it will be relevant for integrating Peregrine into VERA, and thus a consistent initialization process should be developed.

B.3.3 Burnup Models

A pair of auxiliary kernels, \textit{BurnupAux} and \textit{FissionRateAux}, are available as alternatives to the TUBRNP model, which is specified by the inclusion of a \textit{Burnup} block. The TUBRNP model has been historically verified and therefore the \textit{Burnup} block is recommended for fuel performance modeling with Peregrine.

B.3.4 Pellet-Cladding Contact

During this Test Stand, the constraint-based contact system was identified, in collaboration with the INL and Peregrine development teams, as the current way to best model mechanical contact. The following modifications need to be made in the input file to make this system work [35]:

\footnote{Note that burnup-related input parameter are in units of MWd/kgU, not FIMA.}
1. The \textit{Mesh/patch size} must be scaled according to the number of axial elements in the fuel and cladding, however additional research needs to be performed to determine the proper scaling fraction.

2. Set \textit{Contact/system = constraint}

3. Set \textit{ThermalContact/quadrature = true}

4. Remove \textit{Executioner/petsc_options}

5. Set \textit{Executioner/petsc_options_iname = ‘-pc_type –sub_pc_type –pc_asm_overlap – ksp_gmres_restart’}

6. Set \textit{Executioner/petsc_options_value = ‘asm lu 20 101’}

7. Set \textit{Postprocessors/ave_temp_interior/execute_on = residual}

8. Set \textit{Postprocessors/gas_volume/execute_on = residual}

This list of modifications should be formalized and made available in the user documentation for Peregrine.

\subsection*{B.3.5 Coolant Channel Model}

A simple single channel enthalpy rise model is used to establish an estimate for the coolant boundary conditions. The hydraulic diameter is internally determined from the user supplied value for rod pitch using the \textit{CoolantChannel/rod_pitch} input variable. The derivation of the hydraulic diameter was not documented and was assumed to be based on a cluster configuration. Because Falcon input cases used the hydraulic diameter and not the rod pitch, the rod pitch was determined using the cluster formula defined in Falcon documentation. It is recommended that the hydraulic diameter definition be included in the documentation.

\subsection*{B.3.6 Fission Gas Release}

Peregrine contains a detailed fission gas release model called the Simple Integrated Fission Gas Release and Swelling (SIFGRS) model. This model uses multiple different physics models and material characteristics to increase the accuracy of the fission gas release estimates. However, the model is complicated and the user documentation and use cases that were available for this Test Stand were not sufficient for testing. It is recommended that references comparing SIFGRS to other fission gas release models be provided with enough diversity such that guidance on usage of the SIFGRS model can be derived from these reference comparisons.

For example, the SIFGRS model uses 6 additional kernels: fission rate, burnup, porosity, hydrostatic stress, grain radius, and pellet ID. Also, a “pellet brittle zone” \textit{(pbz)} user object must be created. A list of the additional input information that is needed when using the SIFGRS model includes:

- \textit{AuxVariables/fission_rate}
- \textit{AuxVariables/burnup}
- \textit{AuxVariables/porosity}
- \textit{AuxVariables/hydrostatic\_stress}
- \textit{AuxVariables/grain\_radius}
- \textit{AuxVariables/pellet\_id}
• AuxKernels/fission_rate
• AuxKernels/burnup
• AuxKernels/fuel_porosity
• AuxKernels/hydrostatic_stress
• AuxKernels/grain_radius
• AuxKernels/pelletid
• UserObjects/pbz
• Postprocesors/average_grain_radius

The following parameters are available and specific for the SIFGRS fission gas release model:

• Diffusion coefficient
• Burnup at which densification is complete
• HBS gas release threshold
• Initial porosity
• a hydrostatic stress constant
• Bubble surface tension
• Rod average linear power
• Saturation coverage
• Name of the scalar swelling material property
• Final densification fraction
• Transient model option
• A scaling factor for the grain boundary diffusion coefficient
• A scaling factor for the grain radius
• A scaling factor for the intragranular diffusion coefficient
• A scaling factor for the temperature

It is understood that the SIFGRS model is complex and remains under development, and that these options will be helpful to tune the model for specific case conditions when PIE data for those conditions are available. However, the default values for baseline cases should be explicitly defined and included in the distribution package.

B.3.7 Communicating Developer Modifications

Because the MOOSE system is under constant development, input file structure and content often change to accommodate code modifications. However, existing input case files will not necessarily work with updated versions of MOOSE that contain these changes. Information about all the input file changes are not systematically provided at the time of an upgrade. This occurs during development and is understandable, but improved documentation of input file changes during the development cycle are recommended in order to provide a location for users to find this information.

This inoperability has also been observed during modeling changes. A given set of input driving functions and boundary conditions may work with one modeling configuration and will not when a different modeling configuration has prompted developer-advertised modeling improvements. This can be attributed to active Peregrine development occurring during Test Stand activities. It also points to a need for developers to validate Peregrine with a variety of use cases that
represent more realistic scenarios. This collection should become part of a set of validation tests that are executed during the development process.

Another important aspect of the Peregrine development process is that it is dependent on changes dictated by MOOSE development cycles. MOOSE is collaboratively developed by INL in conjunction with partners and an open source community, MOOSE Framework (http://mooseframework.com). Because changes required by a number of collaborators can modify the master MOOSE branch, primary developers ensure that code and input files changes are properly managed for registered projects. This means that all input files associated with the registered projects are automatically updated to work with the latest master branch version. It is recommended that Peregrine be included in this set of registered projects so it may benefit from this input file updating process.

B.4 Feedback on Output and Post-Processing

B.4.1 Finite Element Meshing

Finite element model mesh construction can be the most difficult aspect in setting up a Peregrine input case. A user can generate a mesh by either providing a pre-generated mesh file in a format supported by libMesh or by using the built-in mesh generator that supports construction using lines, rectangles, and rectangular prisms [36]. These two methods are discussed in the following subsections.

B.4.1.1 Built-In Mesh Generation

The input specification for a built-in mesh generation option is available within Peregrine for R-Z axisymmetric models. The geometry information required to generate this FE mesh for Peregrine is non-trivial and features some built-in simplifications that constrain the type of model it can produce. To illustrate the first point, a list of input parameters and their understood definitions is provided in Table B-2 below.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>nx_p</td>
<td>number of radial divisions in the fuel</td>
</tr>
<tr>
<td>ny_p</td>
<td>number of axial divisions in the fuel</td>
</tr>
<tr>
<td>nx_c</td>
<td>number of radial divisions in the cladding</td>
</tr>
<tr>
<td>ny_c</td>
<td>number of axial divisions in the cladding</td>
</tr>
<tr>
<td>ny_cu</td>
<td>number of axial divisions in the upper end plug</td>
</tr>
<tr>
<td>ny_cl</td>
<td>number of axial divisions in the lower end plug</td>
</tr>
<tr>
<td>clad_gap_width</td>
<td>width of gap between the fuel and cladding</td>
</tr>
<tr>
<td>clad_bot_gap_height</td>
<td>lower plenum length</td>
</tr>
<tr>
<td>top_bot_clad_height</td>
<td>average end plug thickness</td>
</tr>
</tbody>
</table>

Note that the short parameter names require documentation to clarify their definitions. In the case of the end plugs, these are not treated separately but as an average thickness of both plugs.
based on the assumption that both plugs have the same height. Additionally, Peregrine does not explicitly model (mesh) the upper and lower end plugs individually. Although this will have a minimal if not negligible effect on most LWR fuel rod simulations, it is recommended that modifications be made such that both end plugs can be modeled explicitly by adding two keywords for end plug mesh specification; one for the upper end plug and one for the lower end plug. It is recommended that the required input parameter keywords used by this built-in mesh generator be consistent with what is typically used to describe the geometry of a fuel rod in the nuclear industry. For example, *clad_bot_gap_height* should be changed to *lower_plenum_length*.

An important parameter required to define the upper plenum length must be calculated externally from the Peregrine input. This parameter is called the *plenum_fuel_ratio* and is defined as

\[ \lambda = \frac{\ell_{\text{lower}} + \ell_{\text{upper}}}{H} \left( \frac{r_{\text{clad}}}{r_{\text{fuel}}} \right) + \frac{G}{r_{\text{fuel}}} \]  

Equation B-1

where

- \( H \) fuel stack length,
- \( \ell_{\text{lower}} \) lower plenum length,
- \( \ell_{\text{upper}} \) upper plenum length,
- \( r_{\text{clad}} \) cladding inner radius,
- \( r_{\text{fuel}} \) fuel outer radius,
- \( G \) radial gap thickness.

Note consistent length units need to be applied for Equation B-1. One final parameter that exists within the *Mesh* input block is the *patch_size*. This variable is of particular relevance when utilizing the constraint-based contact system. This variable is responsible for determining how many neighboring nodes on the slave surface should be checked for contact. As such, the number of axial elements in the fuel and cladding are used to determine the appropriate patch size. It is recommended that some additional studies be performed on the *patch_size* in order to develop a correlation between the *patch_size* and the number of axial elements so that a default value can be provided for this parameter.

### B.4.1.2 File-Based Mesh Generation

Although Peregrine accepts a variety of mesh input file formats such as Exodus II, Abaqus, and Tecplot file formats, none of the supported file formats are common as a standard industry file format. For fuel rod geometries other than 2D axisymmetric models (i.e. R-\( \theta \) or 3D geometries), an external mesh generation tool is required. The tool most often used by MOOSE users is CUBIT, a Geometry and Mesh Generation Toolkit developed by Computational Simulation Software, LLC (CSimSoft) for Sandia National Laboratories (SNL) [37], [38]. CUBIT requires a license before it can be downloaded and used. A U.S. government-use license has made CUBIT readily available to MOOSE and CASL users. Commercial licensing is available through CSimSoft. If built in mesh generators are not available to simplify the generation of R-\( \theta \) or 3D geometries, an external tool such as CUBIT is required.

\(^{3}\) On July 15, 2014, the MOOSE team identified this complexity as an issue and refactored the *plenum_fuel_ratio* to be equal the ratio of plenum height (both top and bottom) to the fuel height.
models, guidance to help users identify a CASL-endorsed and tested procedure for constructing these mesh models with non-VERA tools should be considered when packaging VERA for distribution.

B.4.1.3 Finite Element Fidelity

The fidelity of the finite element calculation performed by Peregrine is highly dependent on the fidelity of the mesh. The three configurations examined during the Test Stand were QUAD4 linear elements, QUAD8 second-order elements, and QUAD9 second-order elements. First-order QUAD4 elements represent the element type recommended by some developers while second-order QUAD8 elements represent the element type recommended by other developers. Consensus among developers must be identified in order to provide confidence in the established user recommendations.

Plots of the radial temperature distribution were obtained from Falcon using QUAD9 second-order elements with 24 axial regions, Peregrine using QUAD4 first-order elements with 70 axial regions, Peregrine using QUAD8 second-order elements with 70 axial regions, and Peregrine using QUAD9 second-order elements with 70 axial regions. Both the Peregrine QUAD8 and QUAD9 results agree with each other. The fuel centerline temperature difference between Peregrine and Falcon observed during this Test Stand is further complicated because this difference is reduced when QUAD4 elements are used.

![Figure B-2. Radial Temperature Profile Estimates using Different Finite Element Quadratures](image)

B.4.1.4 Time Stepping

The selection of a time stepping strategy has to consider computational time and whether Peregrine can obtain an acceptable solution in that time period. For example, the
IterationAdaptiveDT time stepper has the capability to force a time step at every point in a given function. This provides the user the option to base their time stepping strategy on the shape of the power profile. However, the case arises where the underlying time stepping algorithm (which increases or decreases the time step size based on the convergence of the previous step) conflicts with this strategy. It was found that the FunctionDT time stepper had to be used in these situations.

A procedure is also required to provide guidance on bringing a Peregrine case from cold zero power to hot zero power. Initial guidance was to utilize the first 5000 seconds to perform this ramp up to hot zero power. Additional guidance was later provided which suggested using negative time for this ramp. A standard procedure needs to be formalized for ramping from cold to hot conditions in Peregrine both as guidance to users and for integration into VERA.

B.4.1.5 Parallel Processing

The scalability and reproducibility of the Peregrine solver, with respect to the number of processors used for a given simulation, was investigated. One example involves the PCI Example case and Peregrine repository listed in Figure B-8. In this example, attempts were made to use larger time steps during the steady state portion of the simulation. This resulted in different error messages being raised and the time when each case failure occurred to also be different when the number of processors being applied to the case was modified. Then, the exact same case was run with 1, 4, 8, and 16 processors \(^4\). The simulation time before failure, the number of steps executed, and the error message received upon failure are provided in Table B-3.

---

\(^4\) These reference case numbers for these runs are 123 through 127, respectively.
Table B-3. Results from Cases 123 through 127

<table>
<thead>
<tr>
<th>Run ID</th>
<th>Processors</th>
<th>Time to Failure (days)</th>
<th># Steps Executed</th>
<th>Error Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>1</td>
<td>444</td>
<td>34</td>
<td>* SIFGRS: Negative temperature in element: 511</td>
</tr>
<tr>
<td>125</td>
<td>8</td>
<td>361</td>
<td>33</td>
<td>SIFGRS: Negative temperature: (in foo/src/materials/SIFGRS.C, line 598)</td>
</tr>
<tr>
<td>126</td>
<td>16</td>
<td>465</td>
<td>230</td>
<td>Solve failed and timestep already at or below dtmin, cannot continue! (in moose/framework/src/timesteppers/TimeStepper.C, line 219)</td>
</tr>
</tbody>
</table>

* Note that this error message was not properly printed to stderr, it was printed to stdout.

Another example involving the PCI Example case involved use of the adaptive time stepper (AdaptiveDT) and the Peregrine repository listed in Figure B-8 illustrates the difference in the solution when using a different number of processors. When running with 4 processors, the following error message is provided approximately 118 days (time step 67) into the simulation time:

```
[0]PETSC ERROR: --------------------------- Error Message ----------------------------------
[0]PETSC ERROR: Floating point exception!
[0]PETSC ERROR: Infinite or not-a-number generated in norm!
[0]PETSC ERROR: ________________________________
[0]PETSC ERROR: See docs/changes/index.html for recent updates.
[0]PETSC ERROR: See docs/faq.html for hints about trouble shooting.
[0]PETSC ERROR: -----------------------------------------------------------------------------
[0]PETSC ERROR: Libraries linked from /apps/vera/gcc-4.6.1/tpls/opt/common/petsc-3.3-p4/lib
[0]PETSC ERROR: Configure run at Fri Oct 18 17:03:02 2013
[0]PETSC ERROR: VecNorm() line 169 in /apps/vera/gcc-4.6.1/casl_tpls.svn/petsc/3.3-p4_build/src/vec/vec/interface/rvector.c
[0]PETSC ERROR: SNESLineSearchApply_BT() line 154 in /apps/vera/gcc-4.6.1/casl_tpls.svn/petsc/3.3-p4_build/src/snes/linesearch/impls/bt/linesearchbt.c
[0]PETSC ERROR: SNESLineSearchApply() line 503 in /apps/vera/gcc-4.6.1/casl_tpls.svn/petsc/3.3-p4_build/src/snes/linesearch/interface/linesearch.c
```
PETSC ERROR: SNESLSolve() line 222 in /apps/vera/gcc-4.6.1/casl_tpls.svn/petsc/3.3-p4_build/src/snls/impls/ls/ls.c
PETSC ERROR: SNESLSolve() line 3536 in /apps/vera/gcc-4.6.1/casl_tpls.svn/petsc/3.3-p4_build/src/snls/interface/snls.c
PETSC ERROR: solve() line 538 in "unknowndirectory"/home/pbme002/vera/build/app_opt/MOOSEExt/MOOSE/moose/libmesh/src/solvers/petsc_nonlinear_solver.c
-------------------------------------------------------------------------

Figure B-3. Peregrine Error #1

However, when using 16 processors, a different error message is received 465 days (time step 347) into the simulation time:

Solve failed and timestep already at dtmin, cannot continue!


Figure B-4. Peregrine Error #2

Investigation into the root cause of why the same results are not obtained when the number of processors is varied is recommended.

B.4.2 Additional Notes

Below are some miscellaneous items discovered during this Test Stand:

- The native units for burnup within Peregrine are fissions per initial metal atom (FIMA). The conversion for FIMA to MWd/kgU used by the development team is 950, however it is recommended that the conversion factor be calculated on a case-by-case basis since it is dependent on the fuel composition being modeled.
- The active and alive times output in the final summary banner for Peregrine do not take into account parallel processing [35].
- Pellet-to-pellet heat transfer can be modeled using the same gap heat transfer model as is used for pellet-to-clad head transfer [39].
- All final rod diameter measurements do not include the oxide outer surface layer [35].
- If using the SIFGRS fission gas release model, the fuel solid mechanics swelling model cannot be used [35].
- The initial grain radius should be input as the 3D grain radius which is given by the following equation [35]

\[
GR_{3D} = 1.56 \ast GR_{2D}
\]  

Equation B-2

- Until smeared cracking is completed, the fuel is treated as an elastic material [35].

B.4.3 Paraview Post-Processing

The default and most often used output format for Peregrine is an Exodus II-formatted output file. Data contained in this file format can then be visualized using Paraview [29]. However, there are several noteworthy precautions that became evident when using Paraview to post-process Peregrine output files. The first involves oscillating behavior when viewing 2D...
plots for cases which used QUAD8 elements. Figure B-5 illustrates this issue by comparing a contour plot of the radial power factor superimposed on a fuel element mesh using QUAD4, QUAD8, and QUAD9 element types.

Figure B-5. Plots of the Radial Power Factor for Cases Which Used QUAD4 (left), QUAD8 (center) and QUAD9 (right) Elements

From Figure B-5, a truncated saw tooth pattern is observed for the radial power factor in the QUAD8 case only. This was found to be an artifact of the way Paraview displays this data because the actual data solution does not yield this saw tooth pattern.

The second precaution arises when using the *plot over line* feature in Paraview. When dealing with full-length fuel rods which have a significantly large aspect ratio, one typically has to stretch the radial dimension of the model in order to visualize the results. The *plot over line* feature available within Paraview does not operate on this scaled representation of the model. Therefore, the line displayed by Paraview does not represent the line in which data are extracted for. This feature only works when the default scaling of 1:1:1 is used. This precaution must be recognized and notice provided to end users of the code (at least until it can be corrected by the Paraview developers).

### B.5 Feedback on Running Peregrine

#### B.5.1 Repository Listings

Four versions of the VERA software suite were officially configured, built, installed, and tested by EPRI over the course of the Test Stand. Although several repositories make up VERA, only two of these repositories, the MOOSEExt and the MOOSEExt/MOOSE repositories, are relevant to the Test Stand work performed in this report. The MOOSEExt repository holds CASL-specific configuration information required to compile MOOSE within the VERA build system, while the
MOOSEExt/MOOSE repository is the CASL version of INL’s development repository, which also contains the Peregrine repository. A text based Git extension versioning system is used to identify the set of repositories that define a specific version of VERA. The repository versions for MOOSEExt and MOOSEExt/MOOSE are used to specify the version of Peregrine delivered by CASL. There is no tractability between the INL MOOSE/BISON and Peregrine Subversion repository used by CASL to realize the version of Peregrine being distributed by CASL. It is therefore recommended that the Subversion repository information be included in the Git commit message for both MOOSE and Peregrine so that coordination with the development team at INL can be achieved.

The first version of Peregrine, identified using the repository listing provided in Figure B-6, was not used extensively and was quickly abandoned once discrepancies between the available documentation and the version of VERA in use were identified during initial testing. This version of Peregrine was replaced with the repositories listed in Figure B-7.

Official Peregrine training that was conducted in February 2014 identified issues that required a source code update in order to obtain some additional capabilities. This third revision was delivered to EPRI in April 2014 and identified using the repository listing in Figure B-8. This version was used for Progression Problem 1 presented in Section 4.1.

The fourth version of Peregrine was delivered on July 1\textsuperscript{st}, 2014 to provide access to a new fuel-cladding contact modeling system. This version of Peregrine, identified using the repository listing provided in Figure B-9, was used within the final three weeks of the Test Stand to generate results for Progression Problems 2 through 7. Much of the Test Stand efforts were part of the iterative development process of Peregrine, and holding this Test Stand under these conditions was challenging. However, the experience also proved to be beneficial to CASL and the Peregrine development team as they were able to quickly identify development gaps and issues that would need to be solved for EPRI and other Test Stand users to have a positive experience.
Figure B-9. Repository Listing for VERA Installation #4 Installed in July 2014

The logistics and processes associated with integrating the MOOSE and Peregrine repositories into the VERA repository needs improvement. An indicator of this is the 2 to 4 weeks experienced between when an update was requested and when it was received by the Test Stand team at EPRI. As a result, most of the actual engineering analyses described within this report were performed within the last month of the Test Stand. The last month of the project schedule, which was approved by CASL leadership, was scheduled for developing conclusions and recommendations and for writing the final report, and this schedule was approved by CASL leadership. Delays in obtaining code updates prevented the EPRI Test Stand team from executing the Test Stand schedule. The EPRI Test Stand team developed automated scripts during the time of the unscheduled delay. These automation tools became invaluable for comparing results associated with each of the progression problems. Even with these tools, many of the differences observed in the calculation results described in Section 4 of this report could not be fully investigated within the remaining time allotted for this project timeframe (either by the Peregrine developers or the EPRI Test Stand team).

B.5.2 Progression Problem Runs

The work involved in debugging cases and performing parametric studies required a number of cases to be run for each progression problem before satisfactory results were obtained. This included 67 Super-Ramp cases (i.e. progression problem 1) and 272 PCI Example cases (i.e. progression problems 2 through 6). Progression problem 4 required an increase in the quadrature order from second-order to seventh-order in order to get convergence. This translated to an increase in run time of approximately 25 to 30%. Progression problem 5 had significant difficulties converging and multiple different strategies were required in order to obtain reasonable results for this case. The most important observation when debugging this case was that the solution was unstable when run on more than one processor. Additionally, for progression problems 5 and 6, the majority of the information being output as element-, node-, and point-based post-processors needed to be removed from the input file. Otherwise the cases would freeze before executing the first step but never time out or produce an error. These cases had to be terminated manually by the user. In order to get these cases to run to completion, they had to be run on a single processor. However, it was not fully determined whether or not the additional output post-processors or the usage of more than one processor was causing the simulations to freeze. This issues needs to be investigated further by the development team(s).

B.5.3 Scaling

A brief scaling study was performed using progression problems 3 and 6. These two progression problems were run using a number of processors ranging from 1 to 256. Because the trends were similar, the results were averaged to provide a strong scaling result for Peregrine for the PCI Example case and mesh size used. The results of this scaling study along with a trend line representing linear scaling are provided in Figure B-10. From these results, it can be concluded
that, for cases similar to the PCI Example case, using more than 16 processors does not increase the speed of the solution. Further scaling studies should be performed including a weak scaling study to determine how Peregrine scales when the complexity of the problem mesh increases.

![Figure B-10. Strong Scaling for Peregrine using Progression Problems 3 and 6](image)

**B.6 Feedback on Documentation**

EPRI’s experience throughout the Test Stand process was limited because the user and developer documentation was insufficient for providing instruction and guidance for any applications that deviated from the example problems provided. It is recommended that input file descriptions, modeling options, and guidance be provided along with the code repositories to facilitate usage of Peregrine outside the developer community. For example, the documentation for Peregrine provided to the Test Stand team included a 22-page document that covered installation, execution, details for some input file parameters, and descriptions of the sample cases for a previous version of Peregrine. It is recommended more extensive documentation be developed and that the distribution procedure require a verification step before submission.

Several important modeling options and their input specifications need documentation to provide users with specific information. For example, the SIFGRS fission gas release model, described in Section 3.2.3 and discussed further in Appendix B.3.6, includes many modeling options and parameters with no clarification on their usage or their default values. Another example is the contact system model and formulation described in Section 3.2.4 and discussed in Appendix B.3.4. Several different options and parameters exist for specifying how pellet-cladding contact operates. A thorough description for common fuel performance analysis models should be provided to ensure that users apply the correct models and options when using Peregrine. These are two examples, however there are several other models usable by Peregrine that require
additional documentation. It is recommended that the modeling options provided to Peregrine users are reviewed for documentation clarity so that the LWR analysts are able to select default options more readily and advanced options are listed for expert applications.

The developer documentation within Peregrine also needs improvement. The comment-to-code ratio in Peregrine is below 10% and does not appear to follow code development or documentation standards. For this software to be maintainable in the long term (i.e. by individuals other than the original developers), a documentation standard should be developed and implemented. One possible documentation standard is the Doxygen documentation style applied by several VERA software development teams.
Appendix C: Individual Model Analyses

This appendix contains a more in-depth analysis of the relocation and smeared cracking models and how they affect the simulations presented in this report.

C.1 Relocation Model

Relocation, the process by which some of the initial fuel-cladding gap is redistributed within the fuel pellet in the form of cracks, has an effect on the thermal behavior of the fuel pellet region over time. Relocation is a function of the fuel rod geometry and operating conditions such as power level and burnup [1]. Both Peregrine and Falcon are utilizing the ESCORE relocation model [1][13].

This particular result highlights the importance of the relocation threshold value chosen for the relocation model. Figure C-1 plots the fuel centerline temperature at a point on the axial mid-plane of the fuel as a function of time for Falcon and Peregrine. In each case, the codes are using their predefined default value for the relocation threshold.

![Figure C-1. Fuel Centerline Temperature at the Axial Mid-Plane with Both Codes Using their Respective Default for the Relocation Threshold](image)

The reason temperature profiles are different. Specifically, a decrease in fuel centerline temperature is observed in the Peregrine case at approximately 200 days because the default relocation threshold in Peregrine is 6 kW/ft whereas the default in Falcon is 5 kW/ft\(^1\). To verify

\(^1\) Note that although the average linear heat generation rate does not cross this 6 kW/ft threshold at the 200 day mark, the local linear heat generation rate surrounding the data point being evaluated is, and this is the parameter used to trigger activation of the relocation model.
the behavior of the models in each code, the relocation threshold is modified and both sets of results are illustrated in Figure C-2.

![Graph showing fuel centerline temperature at the axial mid-plane plotted for both codes using a relocation threshold of 5 kW/ft and 6 kW/ft.]

**Figure C-2. Fuel Centerline Temperature at the Axial Mid-Plane Plotted for Both Codes Using a Relocation Threshold of 5 kW/ft and 6 kW/ft**

Figure C-2 illustrates that when a relocation threshold of 6 kW/ft is used in either code, the relocation model does not activate until local LHGR exceeds this threshold, which appears to occur at approximately 200 days into the first cycle for elements at the axial mid-plane. Furthermore, a relocation threshold of 5 kW/ft results in comparable results between the two codes. Both of these observations are as expected considering that both codes are using the ESCORE relocation model.

Note that the relocation threshold of 6 kW/ft was first recommended to Falcon users. This value is now recommended to be 5 kW/ft after recent evaluations show that it better represents relocation when evaluating a number of Falcon validation cases.

### C.2 Smeared Cracking Model

One of the more significant differences between the Falcon and Peregrine simulations performed during this Test Stand lies with the use of the smeared cracking model. The decision to turn off the smeared cracking model in Peregrine was made per the recommendation by the Peregrine development team. Falcon’s smeared cracking model remained on in order to facilitate the purpose of the code comparison; a comparison between Peregrine and an industry standard code. Because this represents a significant difference in the physics being modeled between the two codes, a brief analysis of the effects of the smeared cracking model in Falcon is performed using the PCI Example case.
Figure C-3 plots the gap thickness for Falcon, both with and without smeared cracking enabled, as well as for Peregrine with smeared cracking disabled. The impact of the smeared cracking model is minimal using the result presented in Figure C-3 as only minor differences in the gap thickness exists between the two Falcon runs. With smeared cracking turned off, the gap closes approximately 6 days later than it closes when smeared cracking is turned on in Falcon, yet this is still more than 50 days before gap closure is predicted by Peregrine.

![Figure C-3. Comparison of Gap Thickness both with and without the Smeared Cracking Model Enabled in Falcon for a Point along the Axial Mid-Plane of the Active Fuel Region for the PCI Example Case](image)

The fuel outer surface temperature and radial displacement are plotted in Figure C-4 for the first 50 days of the simulated operating history. By looking at the first 50 days the effect of the smeared cracking model becomes more distinguishable. This result also suggests that the relevant effect of the smeared cracking model for this particular simulation is confined to the first 10 days of the simulation. After that time period the effect appears negligible for the remainder of the simulation.
The influence of the smeared cracking model has on the predicted gap thickness during the down power and second cycle restart are evaluated in Figure C-5. This result illustrates significant difference in the result with smeared cracking turned off. Note however that Peregrine, despite not having an active smeared cracking model for this simulation, is predicting the trend in gap thickness to be closer to the trend predicted by Falcon with the smeared cracking model enabled. Figure C-6 shows significant differences in the magnitude of the radial displacement for the Falcon case with smeared cracking disabled, but little difference in the magnitude of the fuel outer surface temperature. The final result for this analysis is the hoop stress comparison presented in Figure C-7. Similar to the observed differences in the radial displacement from Figure C-6, the magnitude of the hoop stress result when smeared cracking is disabled is significantly different than the result predicted by Falcon when smeared cracking is enabled as well as the result predicted by Peregrine. In conclusion, although the smeared cracking model does have a significant effect on the results from the Falcon simulations, it appears that Peregrine is able to make up for some of these differences via relocation and other physics models.
Figure C-5. Comparison of Gap Thickness both with and without the Smeared Cracking Model Enabled in Falcon for a Point along the Axial Mid-Plane of the Active Fuel Region during the Down Power and Second Cycle Restart for the PCI Example Case

Figure C-6. Comparison of Fuel Outer Surface Temperatures (left) and Radial Displacements (right) both with and without Smeared Cracking Enabled in Falcon for the Down Power and Second Cycle Restart for a Point Along the Axial Mid-Plane of the Active Fuel Region for the PCI Example Case
Figure C-7. Comparison of the Hoop Stress both with and without Smeared Cracking Enabled in Falcon for the Down Power and Second Cycle Restart for a Point along the Axial Mid-Plane of the Active Fuel Region for the PCI Example Case
Appendix D: Additional Progression Problem Results

This appendix contains additional plots for the progression problems. Note that many of these plots are not mentioned in the body of the report in Section 4 because no new or significant conclusions could be drawn from the results. They are included in this Appendix as a supplement to the progression problem results provided in Section 4.

D.1 Progression Problem 1

![Axial Temperature Profile along the Cladding Inner Surface at 1126 days for Progression Problem 1](image)

*Figure D-1. Axial Temperature Profile along the Cladding Inner Surface at 1126 days for Progression Problem 1*
Figure D-2. Axial Temperature Profile along the Cladding Outer Surface at 1126 days for Progression Problem 1
D.2 Progression Problem 2

Figure D-3. Axial Temperature Profile along the Cladding Inner Surface at 460 days for Progression Problem 2

Figure D-4. Axial Temperature Profile along the Cladding Inner Surface at 460 days for Progression Problem 2
D.3 Progression Problem 3

Figure D-5. Fuel Centerline Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 3

Figure D-6. Fuel Mid-Plane Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 3
Figure D-7. Fuel Outer Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 3

Figure D-8. Cladding Inner Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 3
Figure D-9. Cladding Outer Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 3

Figure D-10. Coolant Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 3
Figure D-11. Hoop Stress on the Cladding inside Surface Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 3
D.4 Progression Problem 4

Figure D-12. Fuel Mid-Plane Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 3

Figure D-13. Cladding Inner Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 3
Figure D-14. Cladding Outer Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 3
D.5 Progression Problem 5

Figure D-15. Fuel Mid-Plane Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 5

Figure D-16. Cladding Inner Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region
Figure D-17. Cladding Outer Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region

Figure D-18. Temperature as a Function of Radial Position Taken at Points along the Axial Mid-Plane of the Active Fuel Region
Figure D-19. Axial Temperature Profile along the Fuel Centerline

Figure D-20. Axial Temperature Profile along the Fuel Outer Surface
D.6 Progression Problem 6

Figure D-21. Cladding Inner Surface Temperature for the First Cycle (left) and the Down Power and Subsequent Second Cycle Restart (right) Taken at a Point along the Axial Mid-Plane of the Active Fuel Region

Figure D-22. Cladding Outer Surface Temperature for the First Cycle (left) and the Down Power and Subsequent Second Cycle Restart (right) Taken at a Point along the Axial Mid-Plane of the Active Fuel Region
Figure D-23. Coolant Temperature for the First Cycle (left) and the Down Power and Subsequent Second Cycle Restart (right) Taken at a Point along the Axial Mid-Plane of the Active Fuel Region

Figure D-24. Axial Temperature Profile along the Cladding Inside Surface for Progression Problem 6
Figure D-25. Axial Temperature Profile along the Cladding Outside Surface for Progression Problem 6
D.7 Progression Problem 7

Figure D-26. Fuel Outer Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 7

Figure D-27. Cladding Inner Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 7
Figure D-28. Cladding Outer Surface Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region for Progression Problem 7

Figure D-29. Coolant Temperature Taken at a Point along the Axial Mid-Plane of the Active Fuel Region at 470 days for Progression Problem 7
Figure D-30. Temperature as a Function of Radial Position Taken at Points along the Axial Mid-Plane of the Active Fuel Region at 470 days for Progression Problem 7

Figure D-31. Axial Temperature Profile along the Fuel Centerline at 470 days for Progression Problem 7
Figure D-32. Axial Temperature Profile along the Fuel Outer Surface at 470 days for Progression Problem 7