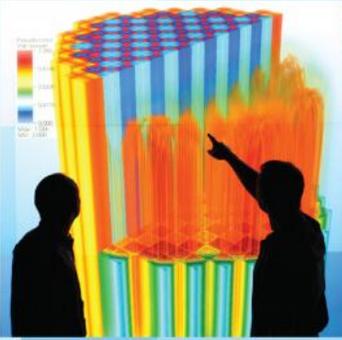


Power uprates  
and plant life extension



Engineering design  
and analysis



# Coupled Single Assembly Solution with VERA (Problem 6)

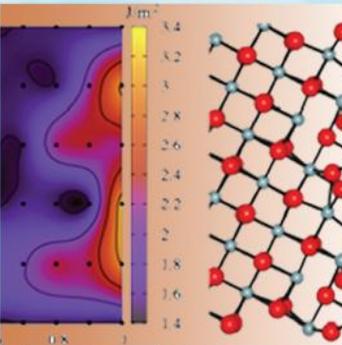
July 25, 2013

Science-enabling  
high performance  
computing

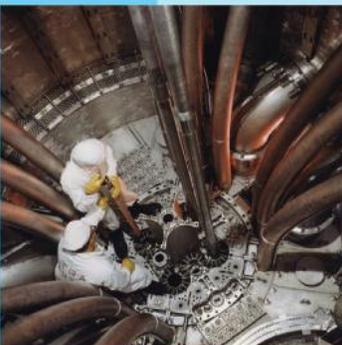


Scott Palmtag

Fundamental science



Plant operational data



U.S. DEPARTMENT OF  
**ENERGY**

**Nuclear Energy**



## Oak Ridge National Laboratory

### *in partnership with*

- Electric Power Research Institute
- Idaho National Laboratory
- Los Alamos National Laboratory
- Massachusetts Institute of Technology
- North Carolina State University
- Sandia National Laboratories
- Tennessee Valley Authority
- University of Michigan
- Westinghouse Electric Company

### *and individual contributions from*

- |                             |                                       |
|-----------------------------|---------------------------------------|
| Anatech Corporation         | Pacific Northwest National Laboratory |
| ASCOMP GmbH                 | Pennsylvania State University         |
| CD-adapco, Inc              | Rensselaer Polytechnic Institute      |
| Core Physics, Inc.          | Southern States Energy Board          |
| City University of New York | Texas A&M University                  |
| Florida State University    | University of Florida                 |
| Notre Dame University       | University of Tennessee               |
| Imperial College London     | University of Wisconsin               |

**REVISION LOG**

<b>Revision</b>	<b>Date</b>	<b>Affected Sections</b>	<b>Revision Description</b>
0	07/25/2013	All	Initial version





## EXECUTIVE SUMMARY

# CONTENTS

EXECUTIVE SUMMARY .....	iii
ACRONYMS .....	v
LIST OF FIGURES.....	vi
LIST OF TABLES.....	vii
1. Introduction.....	2
Acknowledgements .....	3
2. Physics Code Descriptions .....	4
2.1 VERA Common Input (VERAIn) .....	4
2.2 CTF .....	5
2.3 INSILICO .....	5
3. Code Coupling .....	6
3.1 Introduction.....	6
3.2 Building a Single Executable .....	6
3.3 LIME.....	7
3.4 Data Transfer Kit (DTK) .....	7
3.5 Coupling Strategy .....	8
4. Problem Description.....	11
5. Test Problem Results.....	14
5.1 Modeling Options.....	14
5.2 Nominal Results.....	14
5.3 Boron Perturbations .....	18
5.4 Power Perturbations .....	21
6. Conclusion and Recommendations.....	22
6.1 Recommendations.....	22
7. References.....	24
Appendix A – Property Averaging .....	25
A.1 Average coolant properties.....	25
A.2 Average fuel rod properties .....	26
Appendix B – Input File .....	28

## ACRONYMS

AMA	Advanced Modeling Applications
CASL	Consortium for Advanced Simulation of Light Water Reactors
CTF	COBRA-TF
DOE	U.S. Department of Energy
DOE-NE	U.S. Department of Energy Office of Nuclear Energy
DTK	Data Transfer Kit
FA	Focus Area
HFP	Hot Full Power
HPC	high-performance computing
HZP	Hot Zero Power
INL	Idaho National Laboratory
LANL	Los Alamos National Laboratory
LWR	light water reactor
MPO	Materials Performance and Optimization
ORNL	Oak Ridge National Laboratory
PCI	pellet-cladding interaction
PCM	percent mille ( $10^{-5}$ )
PNNL	Pacific Northwest National Laboratory
PPM	parts per million (usually boron)
PSU	Pennsylvania State University
PWR	pressurized water reactor
RTM	Radiation Transport Methods
SNL	Sandia National Laboratories
T-H	thermal-hydraulics
TPL	third-party library
V&V	verification and validation
VERA	Virtual Environment for Reactor Applications
VRI	Virtual Reactor Integration

## LIST OF FIGURES

<b>Figure</b>	<b>Page</b>
Figure 2-1 Key components of a coupled Insilico-CTF application created to solve the example problem.....	8
Figure 2-2 Simplified flow chart illustrating the coupled code “Seidel” fixed point algorithm .....	10
Figure 4-1 Fuel Rod Diagram.....	12
Figure 4-2 Assembly Layout Showing Guide Tubes (GT) and Instrument Tube (IT) placement. ....	13
Figure 5-1 Normalized Radial Fission Rate Distribution at 600 ppm and 100% power .....	15
Figure 5-2 Radial Power Distribution at 600 ppm and 100% power .....	16
Figure 5-3 Axial Distributions at 600 ppm and 100% power (with fuel temperature shown).....	17
Figure 5-4 Axial Distributions at 600 ppm and 100% power (without fuel temperature).....	18
Figure 5-5 Axial Plot of Fission Rates at Different Boron Concentrations .....	19
Figure 5-6 Graphical Output of the Fission Rate, Coolant Density, and Fuel Temperatures at 0 and 1300 ppm Boron.....	20
Figure 5-7 Axial Plot of Fission Rates at Different Power Levels.....	21
Figure A-1 Illustration of how a typical fuel rod is surrounded by four subchannels in a PWR square lattice fuel assembly. ....	25
Figure A-2 Illustrative radial discretization of a fuel rod modeled by CTF.....	26

## LIST OF TABLES

<b>Table</b>	<b>Page</b>
Table 1-1 AMA Progression Benchmark Problems .....	2
Table 4-1 Fuel Rod and Guide Tube Descriptions .....	12
Table 4-2 Assembly Specification.....	13
Table 4-3 Nominal Thermal-Hydraulic Conditions.....	13
Table 5-1 Iteration Summary at 600 ppm boron and 100% power .....	15
Table 5-2 Iteration Summary for Boron Cases.....	18
Table 5-3 Iteration Summary for Power Cases .....	21

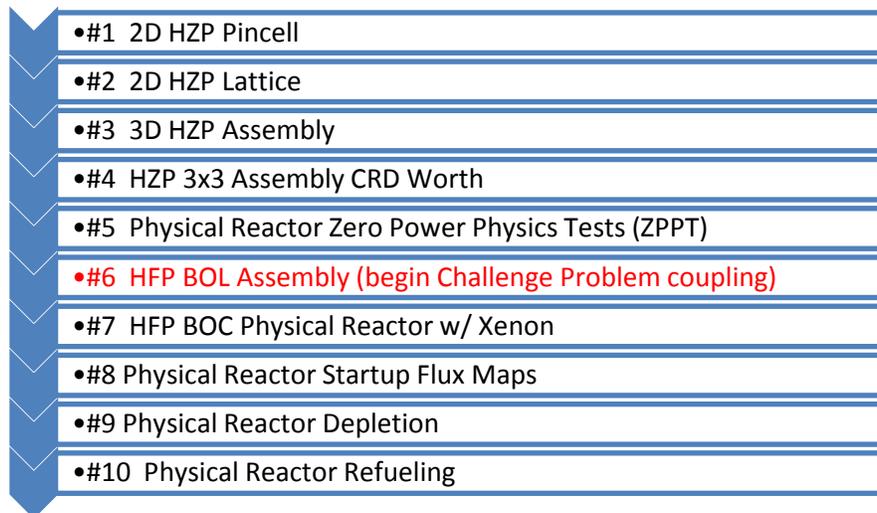
# 1. Introduction

This report documents the completion of Milestone L3:AMA.VDT.P6.03– Coupled Single Assembly Solution with VERA (Problem 6). The purpose of this milestone is to evaluate the modeling of a single PWR assembly with a multiphysics coupling of neutronics (including cross section and neutron transport) and thermal-hydraulics. The neutronics solution is provided by the “Insilico” code from ORNL, and the thermal-hydraulic solution is provided by the COBRA-TF (CTF) code from Penn State University (PSU). The neutronics and T/H are coupled with the LIME and DTK toolkits.

“Problem 6” is one of the AMA Progression Benchmark Problems [1]. The Progression Problems were defined to help drive development of the VERA Core Simulator capability. The problems serve several different functions. The first function is to help developers determine schedule and priorities for implementing features into the core simulator. This function is achieved by building up from a simple geometry capability (a single pincell) to an advanced geometry capability (full-core with depletion), and provide a useful capability at each step. The second function is to let users know when capabilities will be available and provide specific deliveries where users can start verification and validation studies of completed components. The third function, which was not envisioned initially, was to provide measurable metrics that management can use to gauge progress. The VERA Core Simulator is a multi-year development project and it is important to provide management with a long-term schedule and metrics to determine how well progress is being made.

The ten AMA Progression Benchmark Problems are listed in Table 1-1. The purpose of this Milestone report is to document the completion of Problem 6.

**Table 1-1 AMA Progression Benchmark Problems**



•#1 2D HZP Pincell
•#2 2D HZP Lattice
•#3 3D HZP Assembly
•#4 HZP 3x3 Assembly CRD Worth
•#5 Physical Reactor Zero Power Physics Tests (ZPPT)
•#6 HFP BOL Assembly (begin Challenge Problem coupling)
•#7 HFP BOC Physical Reactor w/ Xenon
•#8 Physical Reactor Startup Flux Maps
•#9 Physical Reactor Depletion
•#10 Physical Reactor Refueling

Note that progression problems 1-5 are at hot zero power (HZP) conditions. At HZP, there is no heat generation, and therefore, no thermal-hydraulic feedback. Problem 6 is the first progression problem that includes heat generation and requires coupled thermal-hydraulic feedback. Problem 6 is important because it provides the framework for coupling larger problems (Progression Problem 7) and for coupling other codes needed to address the CASL Challenge Problems (i.e. the PEREGRINE fuel performance code and the MAMBA crud chemistry code).

In Section 2 of this report, a description of the computer codes used in the coupling is given. Section 3 provides information on how the code coupling is performed. Section 4 contains a description of the test problem used in this Milestone. Section 5 provides results for the test problems. Finally, Section 6 contains a conclusion, discussion of future work, and AMA recommendations.

## Acknowledgements

This Milestone was a large project and involved the hard work of many people, including the following people and organizations (listed in alphabetical order):

- Roscoe Bartlett, ORNL
- Kenneth (Noel) Belcourt, SNL
- Kevin Clarno, ORNL
- Greg Davidson, ORNL
- Tom Evans, ORNL
- Andrew Godfrey, ORNL
- Russell Hooper, SNL
- Scott Palmtag, Core Physics Inc.
- Roger Pawlowski, SNL
- Robert Salko, PSU
- Rod Schmidt, SNL
- John Turner, ORNL

## 2. Physics Code Descriptions

This section includes descriptions of the individual physics codes (CTF and Insilico) and the VERA Common Input module.

### 2.1 VERA Common Input (VERAIIn)

The VERA Common Input (VERAIIn) is a single common input used to drive all of the physics codes in the VERA Core Simulator (VERA-CS). Early in the development of the core simulator, it was recognized that it would be unreasonable to require users to generate input decks for each of the individual physics codes. This is especially true if the core simulator allows multiple codes to solve each physics problem (i.e. multiple subchannel codes, multiple neutronics solvers). In addition to the ease-of-use aspects, it is critical in multiphysics applications that all of the different code systems have consistent input. Having a single common input simplifies the user experience and helps ensure that all of the physics applications are solving a consistent geometry.

The common input is based on a single ASCII input file. The input file uses a free-form input format that is based on keyword inputs. The format of the input file was designed by engineers with broad experience with current industry core design tools, so the format of the input file will be easy for industry users to understand. The ASCII input file provides several advantages to the users:

- Allows users to easily transfer input and output between different computer systems.
- Allows users ability to easily edit the file on remote computers.
- Provides a format that users can readily read and understand.
- ASCII input files are an approved archive format recognized by the NRC (ASCII, PDF, or TIFF).
- Allows users to “diff” input files on a variety of remote computers
- Allows users to archive inputs in standard source code repositories and/or directories with read-only permissions.

The input file contains a description of the physical reactor geometry, including: fuel assemblies, removable poison assemblies, control rods, non-fuel structures, detectors, baffle, etc. The input file also contains a description of the current reactor statepoint including: power, flow, depletion, search options, etc.

In order to translate the user input to input needed for the individual code packages, a multistep process is used. First, an input parser reads the text input file and converts it into an XML file. Some physics codes, such as Insilico and MPACT, can read the XML file directly using readily-available XML libraries. Other codes, such as CTF and Peregrine, require an intermediate step that converts the XML file into the native code input. This process allows the common input file to be used for existing physics codes where we do not want to make extensive modifications to the input.

Currently, the following physics codes can interface with the VERA common input:

- Insilico
- MPACT
- CTF
- Peregrine

An example of a VERA common input file is shown in Appendix B – Input File.

It should be noted that there is one class of input that cannot be readily generated by the VERA common input. Some physics codes, such as CFD, require a detailed mesh that is usually generated from a CAD file. For these codes, it is expected that the user will still have to attach an externally generated mesh file and make sure that the mesh file is consistent with the common input.

## 2.2 CTF

COBRA-TF (CTF) is a thermal-hydraulic simulation code designed for Light Water Reactor (LWR) analysis) [2]. CTF has a long lineage that goes back to the original COBRA computer developed in 1980 by Pacific Northwest Laboratory under sponsorship of the Nuclear Regulatory Commission (NRC). The original COBRA began as a thermal-hydraulic rod-bundle analysis code, but subsequent versions of the code have been continually updated and expanded over the past several decades to cover almost all of the steady-state and transient analysis of both PWR's and BWR's. CTF is being developed and maintained by the Reactor Dynamics and Fuel Management Group (RDFMG) at the Pennsylvania State University (PSU).

CTF includes a wide range of thermal-hydraulic models important to LWR safety analysis including flow regime dependent two-phase wall heat transfer, inter-phase heat transfer and drag, droplet breakup, and quench-front tracking. CTF also includes several internal models to help facilitate the simulation of actual fuel assemblies. These models include spacer grid models, a fuel rod conduction model, and built-in material properties for both the structural materials and the coolant (i.e. steam tables).

CTF uses a two-fluid, three-field representation of the two-phase flow. The equations and fields solved are:

- Continuous vapor (mass, momentum and energy)
- Continuous liquid (mass, momentum and energy)
- Entrained liquid drops (mass and momentum)
- Non-condensable gas mixture (mass)

Some of the reasons for selecting CTF as the primary T/H solver in the VERA core simulator are the reasonable run-times compared to CFD (although CFD will be available as an option), the fact that it is being actively developed and supported by PSU, and for the ability to support future applications of VERA such as transient safety analysis and BWR and SMR applications.

## 2.3 INSILICO

Insilico is one of the neutronics solvers in the VERA Core Simulator (along with MPACT) and is part of the Exnihilo transport suite being developed by ORNL. Insilico is the reactor toolkit package of Exnihilo and includes the reactor toolkit used for meshing of PWR geometry, and the cross section generation package based on XSPROC. Insilico uses the Denovo module [3][4] to solve for the flux and eigenvalue solutions of the 3D problem using either the discrete ordinates ( $S_N$ ) solver or the Simplified Legendre ( $SP_N$ ) solver. Exnihilo also includes the SHIFT Monte Carlo package, but SHIFT is not used in this study.

Multigroup cross sections are generated in Insilico using the SCALE code XSPROC. XSPROC performs resonance self-shielding with full range Bondarenko factors using either the narrow resonance approximation or the

intermediate resonance approximation. The fine energy group structure of the resonance self-shielding calculation can optionally be collapsed to a coarse group structure through a one-dimensional (1D) discrete ordinates transport calculation internal to XSPROC. For all of the calculations in this study, the fine energy group structure is collapsed to a 23-group coarse group structure to be used in the Denovo transport solver.

The cross section library used in this study is the SCALE 6.2 252 group ENDF/B-VII.0 neutron cross section data library. This library contains data for 417 nuclides and 19 thermal-scattering moderators.

For coupled calculations, both the  $S_N$  and  $SP_N$  solvers have been used and tested. However, all of the results in this report were generated with the  $S_N$  solver.

Reference [15] contains a more detailed description of the methods used in Insilico as part as the VERA Core Simulator.

## 3. Code Coupling

### 3.1 Introduction

This Milestone is a demonstration of coupling two physics codes together to calculate the temperature, fission rate, and neutron flux distribution within a single PWR 17x17 fuel assembly. All neutronics aspects of the problem (cross-sections, neutron transport, and power release) are solved using Insilico and all thermal-hydraulic aspects (including fuel rod conduction) are solved using CTF. The coupling of these codes to create a single-executable multiphysics coupled-code application is done using the VERA infrastructure tools LIME [6] and DTK [7].

### 3.2 Building a Single Executable

To couple the physics codes CTF and Insilico together, both programs must be combined and compiled in a single executable. This is done by creating a top level LIME problem manager and refactoring CTF and Insilico so they are subroutines instead of standalone programs. The LIME problem manager serves as the “main” program, controls the iteration strategy, calls the CTF and Insilico subroutines as needed, and transfers data between the codes using LIME model evaluators and DTK (See Figure 2-1 below).

The single coupled executable is named “VRIPSScobra\_denovo\_coupled.exe” and contains all of the coupling codes. This program is located in the VERA GIT repository “PSSDrivers”:

```
PSSDriversExt/VRIPSS/drivers/cobra_denovo/VRIPSScobra_denovo_coupled.exe
```

Compiling different physics codes together can be a complicated task, especially when the packages are large and rely on additional third-party libraries (TPL’s). To overcome these complications, the TriBITS build system is used. In addition to providing the build system, TriBITS also provides an integrated testing platform to help automated developer testing.

TriBITS stands for the “Tribal Build, Integrate, and Test System” and was originally developed for Trilinos, but was later extended for VERA, SCALE and other projects. TriBITS is based on the well-known Kitware open-source toolset CMake, CTest, and Cdash. Some additional features of TriBITS include the following:

- Built-in CMake-based package architecture support for partitioning a project into “Packages” with carefully regulated dependencies with numerous features including:
  - Automatic enabling of upstream and downstream packages (critical for large projects like Trilinos, SCALE, and CASL)
  - Integrated MPI and CUDA support
  - Integrated TPL support (coordinate common TPLs across unrelated packages, common behavior for user configuration, etc.)
  - Removal of a lot of boiler-plate CMake code for creating libraries, executables, copying files, etc.
- Powerful TRIBITS\_ADD\_[ADVANCED]\_TEST(...) wrapper CMake functions to create advanced tests
- Integrated support for add-on repositories with add-on packages.
- TribitsCTestDriver.cmake testing driver:
  - Partitioned package-by-package output to CDash and reporting on a package-by-package basis
  - Failed packages don’t propagate errors to downstream packages
  - Integrated coverage and memory testing (showing up on CDash)
  - Nightly and continuous integration (CI) test driver.
- Pre-push synchronous CI testing with the Python checkin-test.py script
- In addition: TribitsDashboardDriver system, download-cmake.py and numerous other tools

TriBITS is an open-source project and is available for download from the internet [8].

### 3.3 LIME

The Lightweight Integrating Multiphysics Environment for coupling codes (LIME) is used to integrate the two physics codes [6][9]. LIME is designed to integrate separate computer codes, which may be written in different languages, into a single package to solve multiphysics problems. LIME provides high-level routines to create a “Problem Manager” to control the overall-iterations and perform communication through “Model Evaluators” for each of the separate physics codes.

A description of how LIME is used to couple CTF and Insilico is provided in Section 3.5.

LIME is an open-source project and is available for download from the internet [10].

### 3.4 Data Transfer Kit (DTK)

The Data Transfer Kit (DTK) library is used to transfer data between the two physics codes. DTK is based on the Rendezvous algorithm [7] and facilitates the transfer of data between multiple codes with different meshes partitioned on different parallel processors. From the DTK website:

*“The Data Transfer Kit (DTK) is a software component designed to provide parallel services for mesh and geometry searching and data transfer for arbitrary physics components. In many physics applications, the concept of mesh and geometry is used to subdivide the physical domain into a discrete representation to facilitate the solution of the model problems that describe it. Additionally, the concept of the field is used to apply degrees of freedom to the mesh or geometry as a means of function discretization. With the increased development efforts in multiphysics simulation, adaptive mesh simulations, and other multiple mesh/geometry problems, generating parallel topology maps for*

*transferring fields and other data between meshes is a common operation. DTK is being developed to provide a suite of concrete algorithm implementations for these services.”*

DTK is an open-source project and is available for download from the internet [11].

### 3.5 Coupling Strategy

A challenging aspect of coupling neutronics and thermal-hydraulics is that the different physics associated with these two codes are strongly coupled and nonlinear. By strongly coupled we mean that the quantities calculated in each physics code and passed to the other have a significant impact on the solution of the other physics code. By nonlinear we mean that a change in values calculated in one code do NOT result in a “linearly-proportional” change to values in the other.

Figure 2-1 helps illustrate key aspects of the single-executable coupled-code (Insilico-CTF) simulation tool created within VERA to solve this problem.

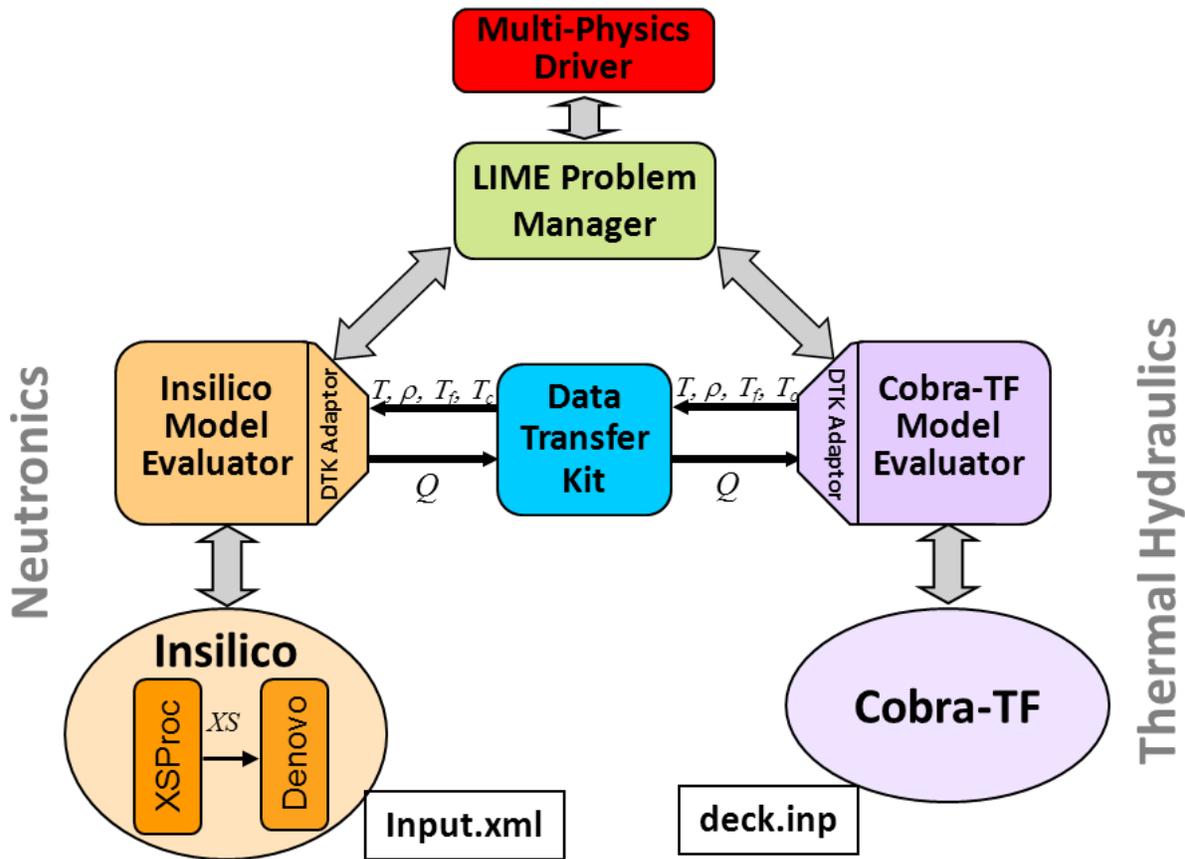


Figure 2-1 Key components of a coupled Insilico-CTF application created to solve the example problem.

To solve the neutronics part of the overall problem, Insilico must be provided with values for the following quantities associated with each rod at each axial level:

1. average fuel temperature,  $T_f$
2. average clad temperature,  $T_c$
3. average coolant temperature surrounding the rod,  $T$
4. average coolant density surrounding the rod,  $\rho$

These quantities are calculated in the CTF code and stored in the following two-dimensional arrays in the “transfer\_io” module.

cool_avg_den( $n,jh$ )	Average Coolant Density
cool_avg_tmp( $n,jh$ )	Average Coolant Temperature
clad_avg_tmp( $n,jh$ )	Average Clad Temperature
fuel_avg_tmp( $n,jh$ )	Average Fuel Temperature

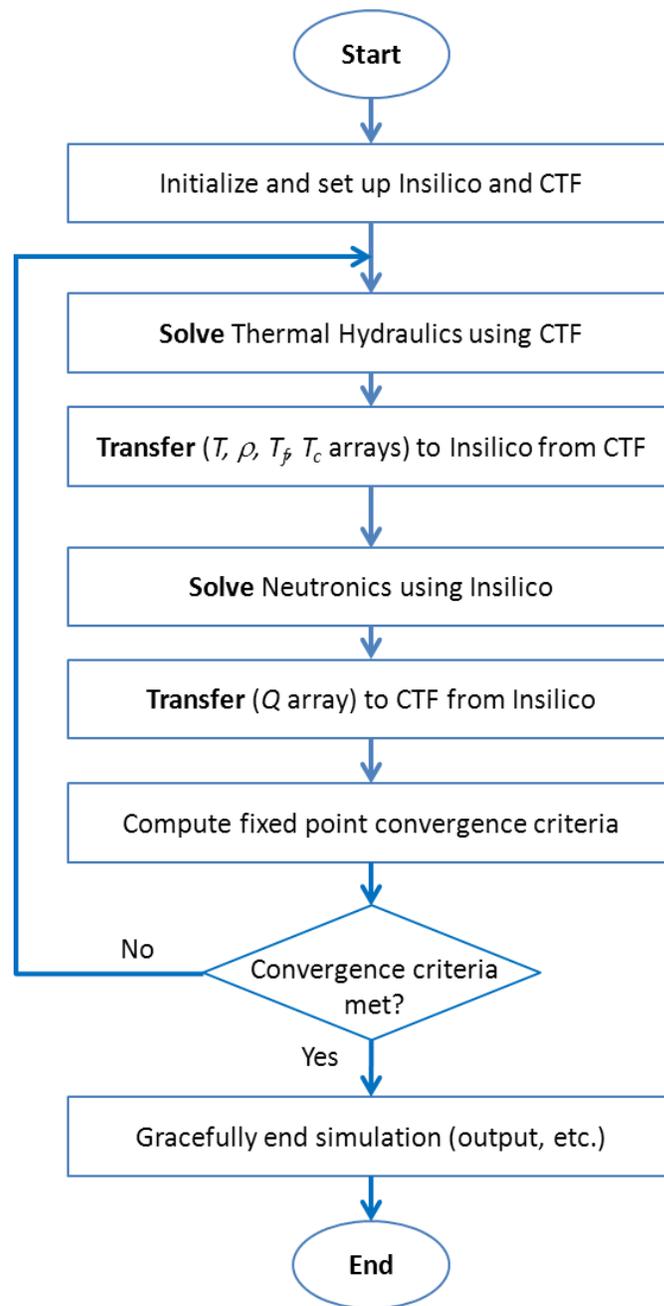
Here  $n$  denotes the fuel rod and  $jh$  the axial heat transfer level. Values in these arrays are transferred to Insilico at designated times during the overall solution procedure. Of note is that Insilico is itself solving a multiphysics neutronics problem that involves calculating cross sections, doing neutron transport, and computing energy release.

To solve the thermal hydraulics part of the problem, CTF needs the energy release rate  $Q$  in each fuel rod at each axial level. These values are computed by Insilico and transferred to CTF. Note that CTF also solves several coupled-physics equation sets internally, i.e. conservation of mass, momentum and energy in the fluid together with heat transfer to fuel rods where energy is being released and conducted within the rods.

The transfer of data between Insilico and CTF is enabled and directed by several additional software components represented in Figure 2-1 (e.g. Insilico and CTF Model Evaluators and DTK adaptors). These small components leverage LIME and DTK and provide the additional functionality needed to create the overall coupled-code simulation capability. In particular, they address the details of how and where the transfer data is stored in each code, and how to correctly transfer that data in the form required by both the “source” and the “target” during each transfer operation.

As described in references [6] and [9], LIME supports several different types of nonlinear solution strategies (i.e. Newton, JFNK, fixed point) depending on the capabilities available from the physics codes being coupled. In this case, we solve the overall coupled nonlinear system using a simple “Seidel” mode fixed point algorithm. This is an iterative method where each physics code is sequentially solved independently within a global iteration loop, and updated transfer-data is passed between physics codes immediately after each physics code solution. In addition, the change in transferred values between iterations can be “relaxed” so as to improve the convergence speed of the approach.

The following simplified execution diagram illustrates the basics of how the “Seidel” mode fixed point algorithm is executed by the LIME problem manager for our example problem



**Figure 2-2 Simplified flow chart illustrating the coupled code “Seidel” fixed point algorithm**

The first time CTF is asked to perform a solve the power release is internally specified based on a typical power profile that has the correct overall power. Thereafter, the power is specified by the transfer-data received from the most recent Insilico calculation.

Because neither Insilico nor CTF can currently provide a residual vector to LIME, the convergence criteria used here is based on checking that key global metrics associated with the solutions in each code have reached a steady invariant condition within a user-specified tolerance. Currently the following parameters are checked for convergence:

1. Eigenvalue
2. Maximum change in local power
3. Maximum change in local fuel temperature
4. Maximum change in local clad temperature
5. Maximum change in local coolant temperature

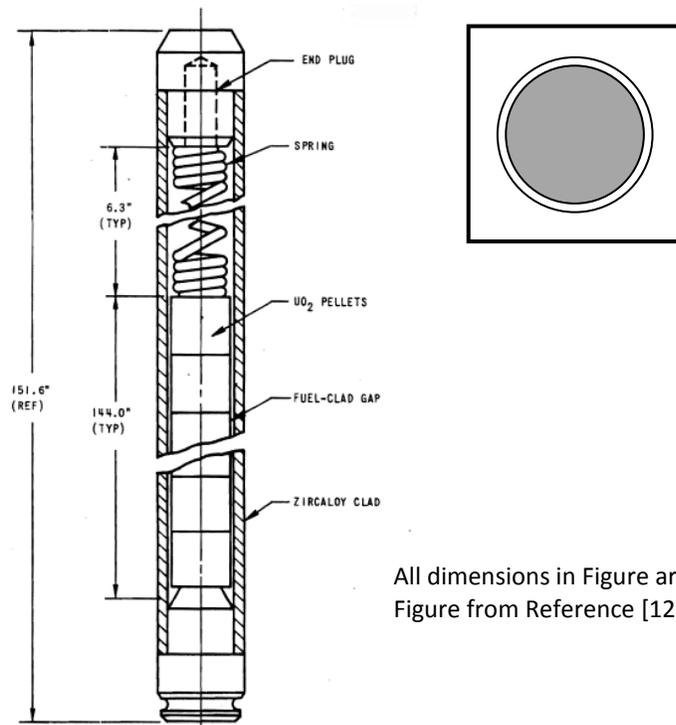
## 4. Problem Description

The example problem used in this Milestone is a PWR single assembly based on the dimensions and state conditions of Watts Bar Unit 1 Cycle 1. The dimensions for the assembly are identical to AMA Progression Benchmarks [1] “Problem 3” and “Problem 6”. Problems 3 and 6 are identical, except that Problem 3 is at Hot Zero Power (HZP) and has no T/H feedback, and Problem 6 is at Hot Full Power (HFP) and includes T/H feedback. (All dimensions are non-proprietary and are derived from the publically available Watts Bar Unit 1 FSAR [12].)

In addition to the boron and power levels specified in the benchmark specifications, additional test cases were run at different boron concentrations (0, 600, and 1300 ppm) and power levels (70, 100, 130, and 150% power).

The assembly is a standard 17x17 Westinghouse fuel design with uniform fuel enrichment. There are no axial blankets or enrichment zones. The assembly has 264 fuel rods, 24 guide tubes, and a single instrument tube in the center. There are no control rods or removable burnable absorber assemblies in this problem.

The primary geometry specifications of the fuel rod and guide tube materials are given in Figure 4-1 and Table 4-1. The geometry specification for the assembly is given in Figure 4-2 and Table 4-2. For a complete description of the geometry, including spacer grid and nozzle specifications, refer to Reference [1]. The complete input listing for this problem is shown in Appendix B – Input File



All dimensions in Figure are in inches  
Figure from Reference [12], Figure 4.2-3

**Figure 4-1 Fuel Rod Diagram**

**Table 4-1 Fuel Rod and Guide Tube Descriptions**

Parameter	Value	Units
Fuel Pellet Radius	0.4096	cm
Fuel Rod Clad Inner Radius	0.418	cm
Fuel Rod Clad Outer Radius	0.475	cm
Guide Tube Inner Radius	0.561	cm
Guide Tube Outer Radius	0.602	cm
Instrument Tube Inner Radius	0.559	cm
Instrument Tube Outer Radius	0.605	cm
Outside Rod Height	385.10	cm
Fuel Stack Height (active fuel)	365.76	cm
Plenum Height	16.00	cm
End Plug Heights (x2)	1.67	cm
Pellet Material	UO <sub>2</sub>	
Clad / Caps / Guide Tube Material	Zircaloy-4	

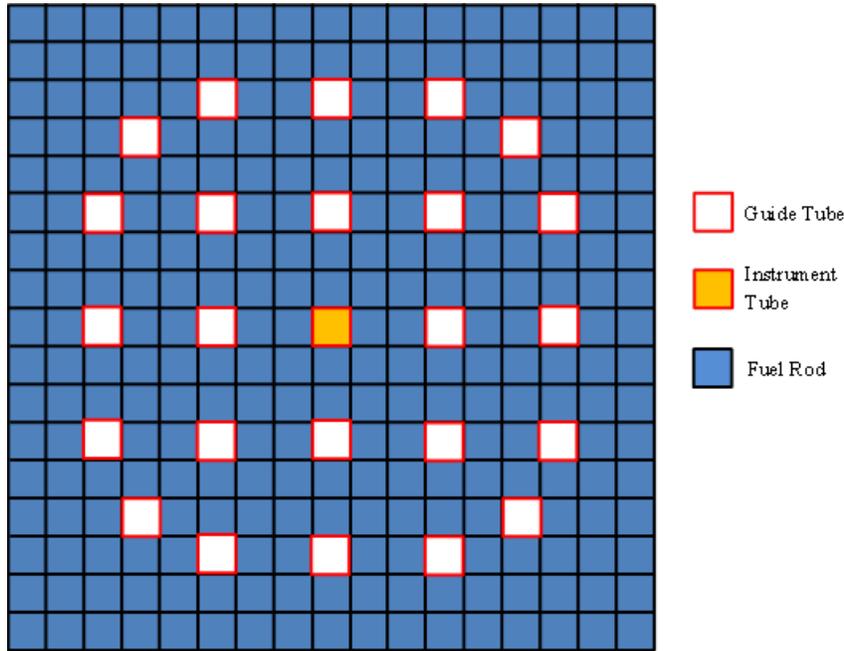


Figure 4-2 Assembly Layout Showing Guide Tubes (GT) and Instrument Tube (IT) placement.

Table 4-2 Assembly Specification

Parameter	Value	Units
Rod Pitch	1.26	cm
Assembly Pitch	21.5	cm
Inter-Assembly Half Gaps	0.04	cm
Geometry	17x17	
Number of Fuel Rods	264	
Number of Guide Tubes (GT)	24	
Number of Instrument Tubes (IT)	1	

The thermal-hydraulic specifications for this problem are shown in Table 4-3. The thermal-hydraulic conditions and feedback are what differentiate Progression Problems 3 and 6.

Table 4-3 Nominal Thermal-Hydraulic Conditions

Parameter	Value	Units
Inlet Temperature	559	degrees F
System Pressure	2250	psia
Rated Flow (100% flow)	0.6824	Mlb/hr
Rated Power (100% power)	17.67	MWt

## 5. Test Problem Results

### 5.1 Modeling Options

All of the results in this report were run on Titan using the VERA build from 6/28/2013. Each case was run with 1156 cores (17x17 rods x 4 energy sets).

In the neutronics solution, the Insilico  $S_N$  solver is used with the “qr” quadrature set (4 azimuthal angles and 4 polar angles per octant). The pincell calculation used 252 energy groups and is collapsed to 23 energy groups for the 3D transport solution. Insilico uses a 4x4 mesh in each fuel rod and a maximum 2.54 cm in the axial direction. Axial boundaries are positioned at each material and edit interfaces. The neutron flux is calculated from below the lower core plate to above the upper core plate in order to capture the axial leakage effects.

In the T/H solution, CTF has 49 axial levels over the active fuel region. The axial levels are defined to explicitly include the spacer grid heights, and to use uniform mesh spacing between the spacer grids. The maximum axial mesh is approximately 7 cm. The exact axial levels used in CTF are listed in the [EDITS] block of the sample input file. The CTF fuel rod heat conduction model uses 3 radial rings in each fuel rod.

Data transfer between Insilico and CTF occurs at each fuel rod on the 49 axial level mesh.

Note that these problems are not fully converged spatially in the DENOVO transport solver. The purpose of this Milestone is to demonstrate the coupling between neutronics and T/H, so a somewhat coarse spatial discretization is used in DENOVO to reduce problem run-times. One recommendation for future work is to determine the set of input parameters needed to converge the transport problem in spatial mesh, angle, scattering angle, and energy. (See further recommendations in Section 6.1.)

### 5.2 Nominal Results

A typical iteration summary for the single-assembly at 600 ppm boron is shown in Table 5-1.

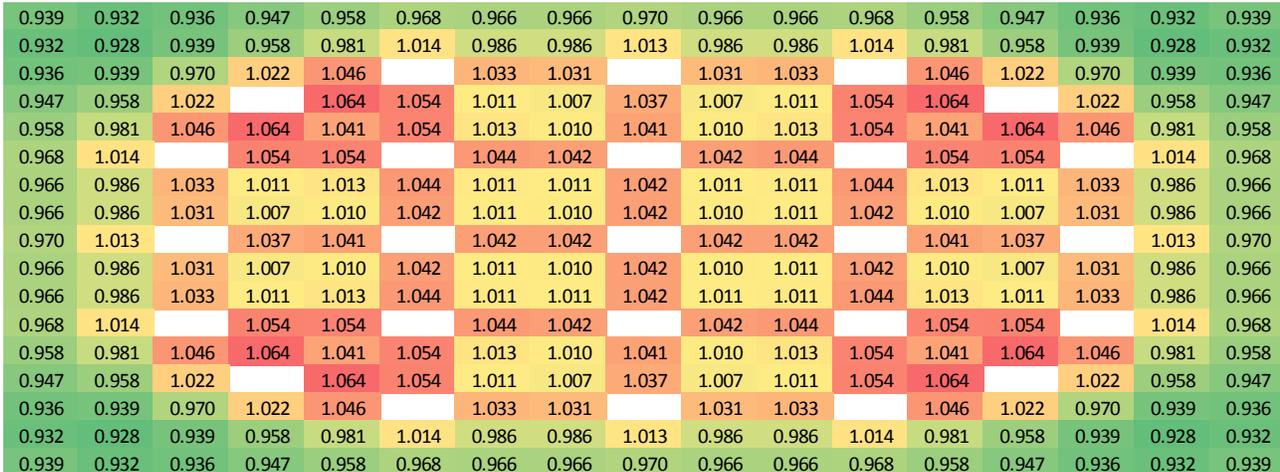
- The first column shows the coupled iteration count.
- The second column (its) shows the number of DENOVO iterations taken per coupled iteration.
- The third column (xkeff) shows the reactor eigenvalue.
- The fourth column (keff\_dif) shows the change in eigenvalue between coupled iterations.
- The fifth column (powr\_dif) shows the change in local power between coupled iterations.
- The sixth column shows the maximum coolant temperature (degrees F, which are native CTF units) averaged over a single subchannel mesh and axial mesh.
- The seventh and eighth columns show the maximum clad and fuel temperatures (degrees F, which are native CTF units) averaged over a single fuel rod and axial mesh.
- The ninth through eleventh columns show the change in peak temperatures between iterations for the coolant, clad, and fuel respectively.

**Table 5-1 Iteration Summary at 600 ppm boron and 100% power**

	its	xkeff	keff_dif	powr_dif	cool temp	clad temp	fuel temp	cool delta	clad delta	fuel delta	
	1	27	1.23376105	18947.0	1.000000	623.60	705.40	2109.00	64.55	128.00	1218.00
	2	22	1.23302139	60.0	0.222356	623.58	698.42	2038.82	0.03	-7.00	-69.94
	3	19	1.23336920	28.2	0.040185	623.70	697.68	2003.28	0.12	-0.74	-35.53
	4	15	1.23339591	2.2	0.004610	623.61	698.01	2013.56	-0.09	0.33	10.28
	5	9	1.23341182	1.3	0.000644	623.57	698.32	2018.20	-0.04	0.31	4.64
	6	5	1.23343045	1.5	0.000166	623.69	698.45	2021.05	0.12	0.12	2.85
	7	2	1.23344063	0.8	0.000015	623.68	698.54	2022.51	-0.02	0.10	1.46
	8	1	1.23343920	0.1	0.000000	623.67	698.59	2023.22	-0.01	0.05	0.71
	9	1	1.23343781	0.1	0.000000	623.66	698.62	2023.59	-0.01	0.03	0.37
	10	1	1.23343668	0.1	0.000000	623.66	698.63	2023.77	-0.00	0.01	0.18
	11	1	1.23343567	0.1	0.000000	623.66	698.64	2023.87	-0.00	0.01	0.10

Note that the convergence criterion for this problem is very tight. The eigenvalue convergence is set to 5 pcm ( $5 \times 10^{-5}$  delta-k). The most limiting convergence criteria is the maximum change in peak fuel temperature, which has a convergence criteria of 0.1 degrees F. All of the coupled iteration parameters are set in the [COUPLED] block of the input file.

The normalized radial fission rate distribution integrated over the axial direction is shown in Figure 5-1. Note that the results are octant symmetric and there is no power in the guide tubes or instrument tubes.



**Figure 5-1 Normalized Radial Fission Rate Distribution at 600 ppm and 100% power**



A map of the coolant density in the top axial elevation of the core is shown in Figure 5-2. The coolant density shown in this map is the average density surrounding each rod, not the density in each CTF channel. The calculation for this density is described in Appendix A – Property Averaging. Note that the exit density is lower in the center of the assembly, corresponding to the higher fuel rod powers shown in Figure 5-1.

0.659	0.658	0.657	0.656	0.656	0.655	0.655	0.655	0.655	0.655	0.655	0.655	0.656	0.656	0.657	0.658	0.659	
0.658	0.657	0.657	0.656	0.655	0.655	0.654	0.654	0.654	0.654	0.654	0.655	0.655	0.656	0.657	0.657	0.658	
0.657	0.657	0.656	0.655	0.654	0.654	0.653	0.653	0.653	0.653	0.653	0.654	0.654	0.655	0.656	0.657	0.657	
0.656	0.656	0.655	0.654	0.653	0.653	0.652	0.652	0.652	0.652	0.652	0.652	0.653	0.653	0.654	0.655	0.656	0.656
0.656	0.655	0.654	0.653	0.653	0.652	0.652	0.652	0.652	0.652	0.652	0.652	0.652	0.653	0.653	0.654	0.655	0.656
0.655	0.655	0.654	0.653	0.652	0.652	0.651	0.651	0.651	0.651	0.651	0.651	0.652	0.652	0.653	0.654	0.655	0.655
0.655	0.654	0.653	0.652	0.652	0.651	0.651	0.651	0.651	0.651	0.651	0.651	0.651	0.652	0.652	0.653	0.654	0.655
0.655	0.654	0.653	0.652	0.652	0.651	0.651	0.651	0.651	0.651	0.651	0.651	0.651	0.652	0.652	0.653	0.654	0.655
0.655	0.654	0.653	0.652	0.652	0.651	0.651	0.651	0.651	0.651	0.651	0.651	0.651	0.652	0.652	0.653	0.654	0.655
0.655	0.654	0.653	0.652	0.652	0.651	0.651	0.651	0.651	0.651	0.651	0.651	0.651	0.652	0.652	0.653	0.654	0.655
0.655	0.655	0.654	0.653	0.652	0.652	0.651	0.651	0.651	0.651	0.651	0.651	0.652	0.652	0.653	0.654	0.655	0.655
0.656	0.655	0.654	0.653	0.653	0.652	0.652	0.652	0.652	0.652	0.652	0.652	0.652	0.653	0.653	0.654	0.655	0.656
0.656	0.656	0.655	0.654	0.653	0.653	0.652	0.652	0.652	0.652	0.652	0.652	0.653	0.653	0.654	0.655	0.656	0.656
0.657	0.657	0.656	0.655	0.654	0.654	0.653	0.653	0.653	0.653	0.653	0.653	0.654	0.654	0.655	0.656	0.657	0.657
0.658	0.657	0.657	0.656	0.655	0.655	0.654	0.654	0.654	0.654	0.654	0.654	0.655	0.655	0.656	0.657	0.657	0.658
0.659	0.658	0.657	0.656	0.656	0.655	0.655	0.655	0.655	0.655	0.655	0.655	0.656	0.656	0.657	0.658	0.659	0.659

**Figure 5-2 Exit Coolant Density (g/cc) at 600 ppm and 100% power**

Average axial distributions for this problem are shown in Figures 5-3 and 5-4. Figure 5-3 includes the fuel temperature and Figure 5-4 does not include the fuel temperature so the coolant and clad temperature profiles are easier to see.

Note the small “dips” in the axial fission rate and fuel temperature profiles. These dips are due to the presence of spacer grids. The spacer grids displace moderator in the coolant channels and decrease the neutron moderation around the grids. The decreased moderation causes a local depression in the flux and power.

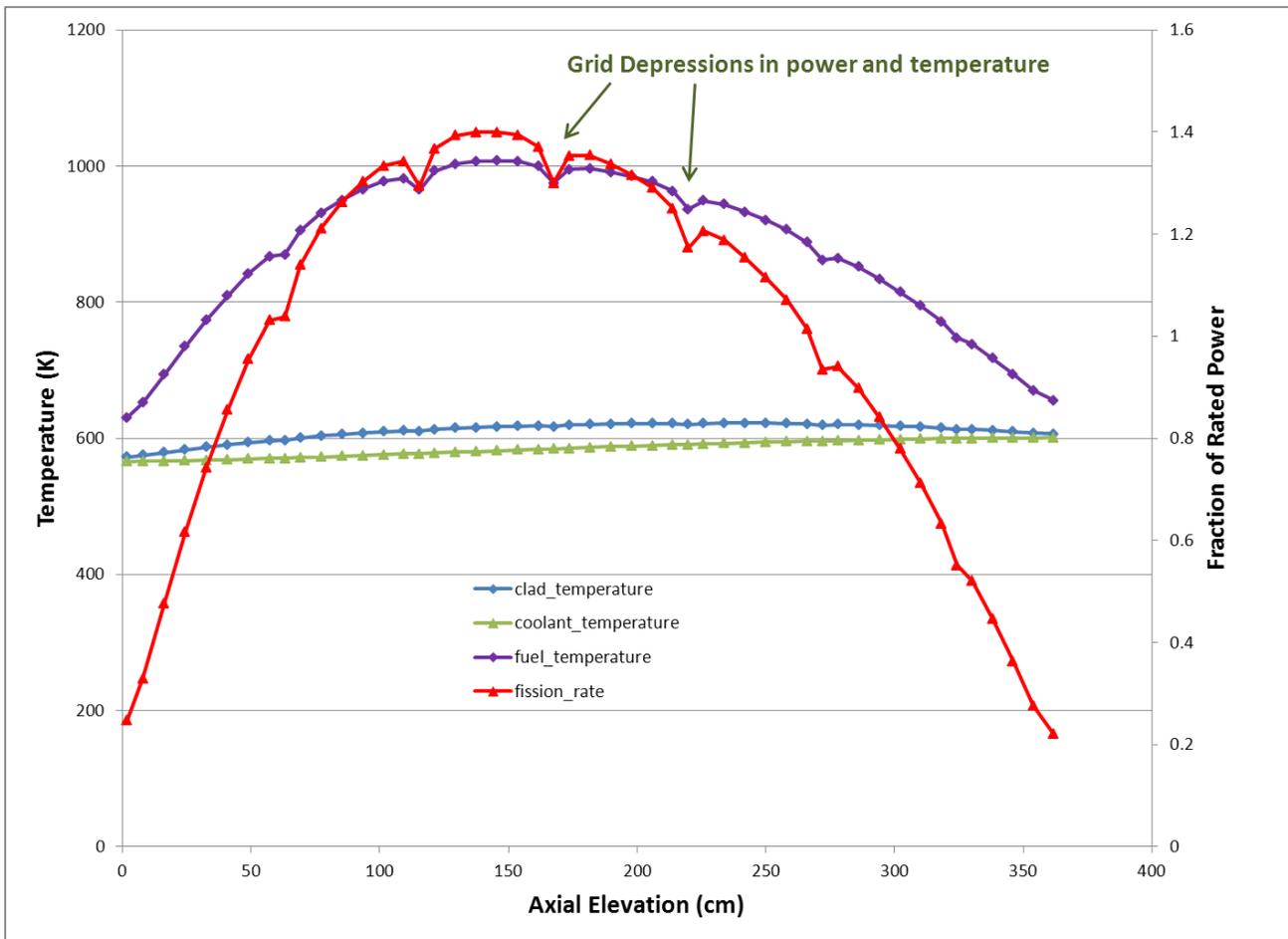


Figure 5-3 Axial Distributions at 600 ppm and 100% power (with fuel temperature shown)

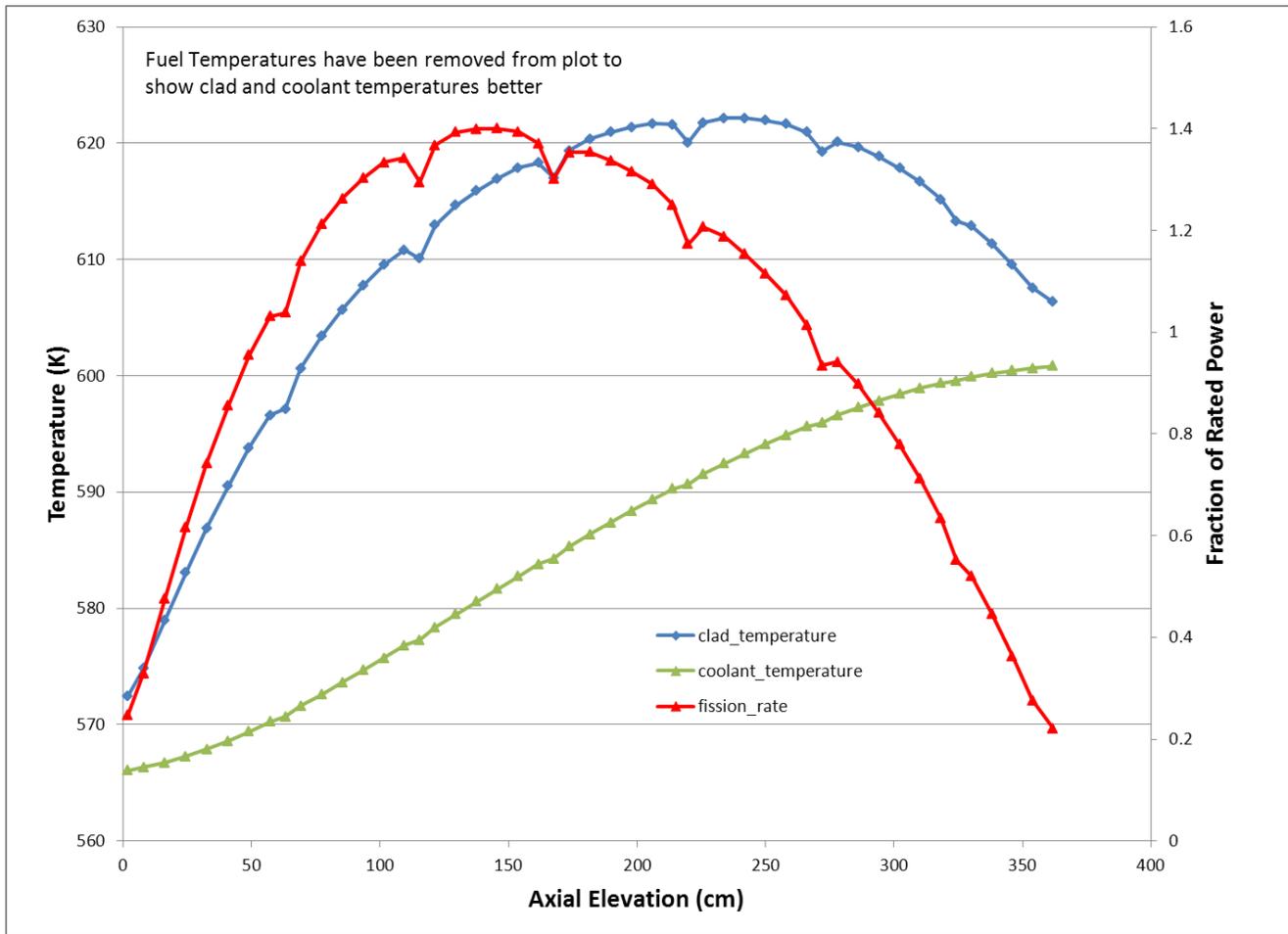


Figure 5-4 Axial Distributions at 600 ppm and 100% power (without fuel temperature)

### 5.3 Boron Perturbations

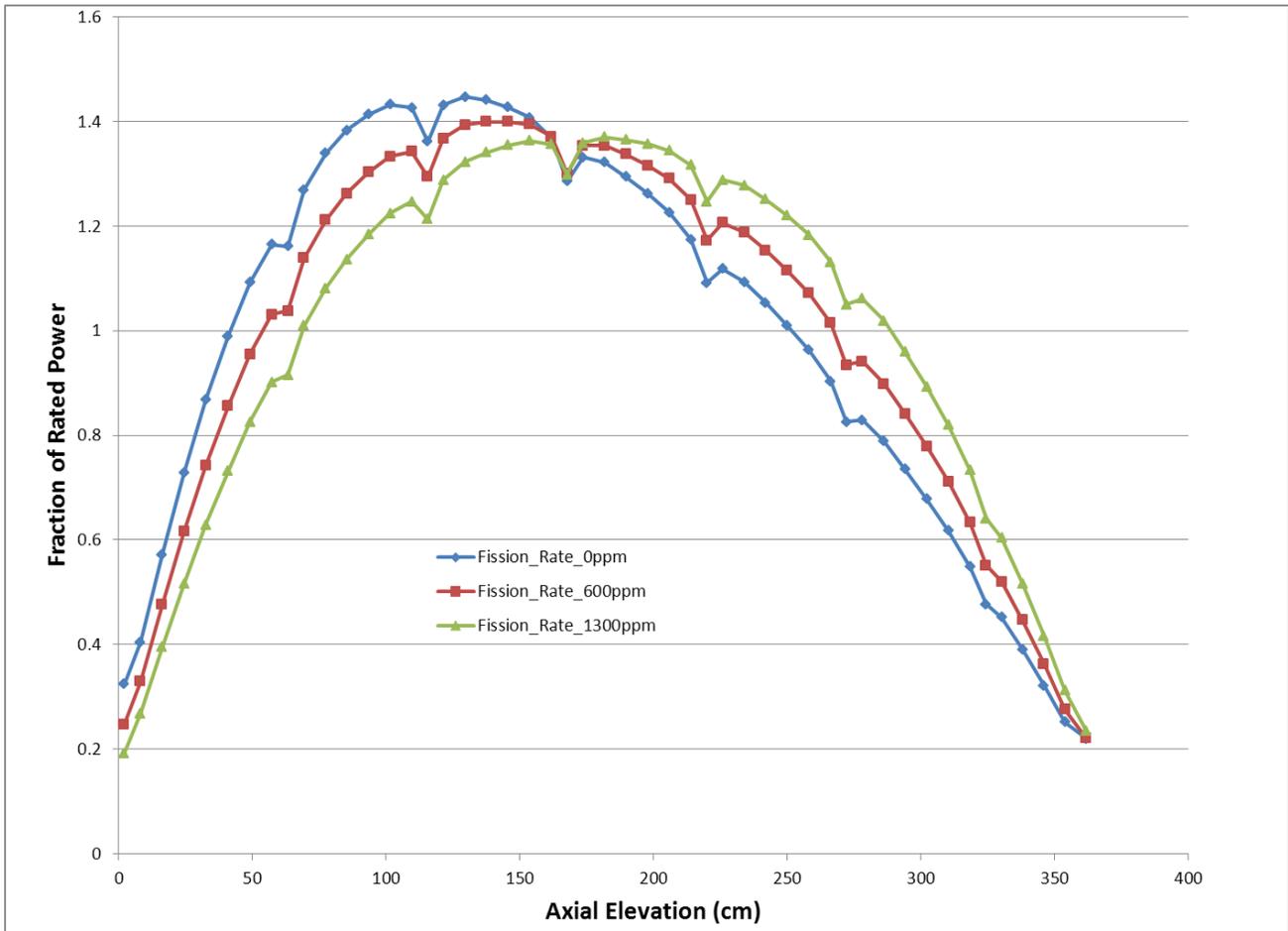
In order to see the effects of different boron concentrations on the results, the single-assembly case was run at three different boron concentrations – 0, 600, and 1300 ppm boron. The eigenvalues and wall-clock times for these cases are listed in Table 5-2.

Table 5-2 Iteration Summary for Boron Cases

Boron Concentration	Eigenvalue	Wall Time (HH:MM:SS)	Coupled Iterations
0 ppm	1.31286	5:11:54	11
600 ppm	1.23344	4:27:23	11
1300 ppm	1.15336	4:06:21	11

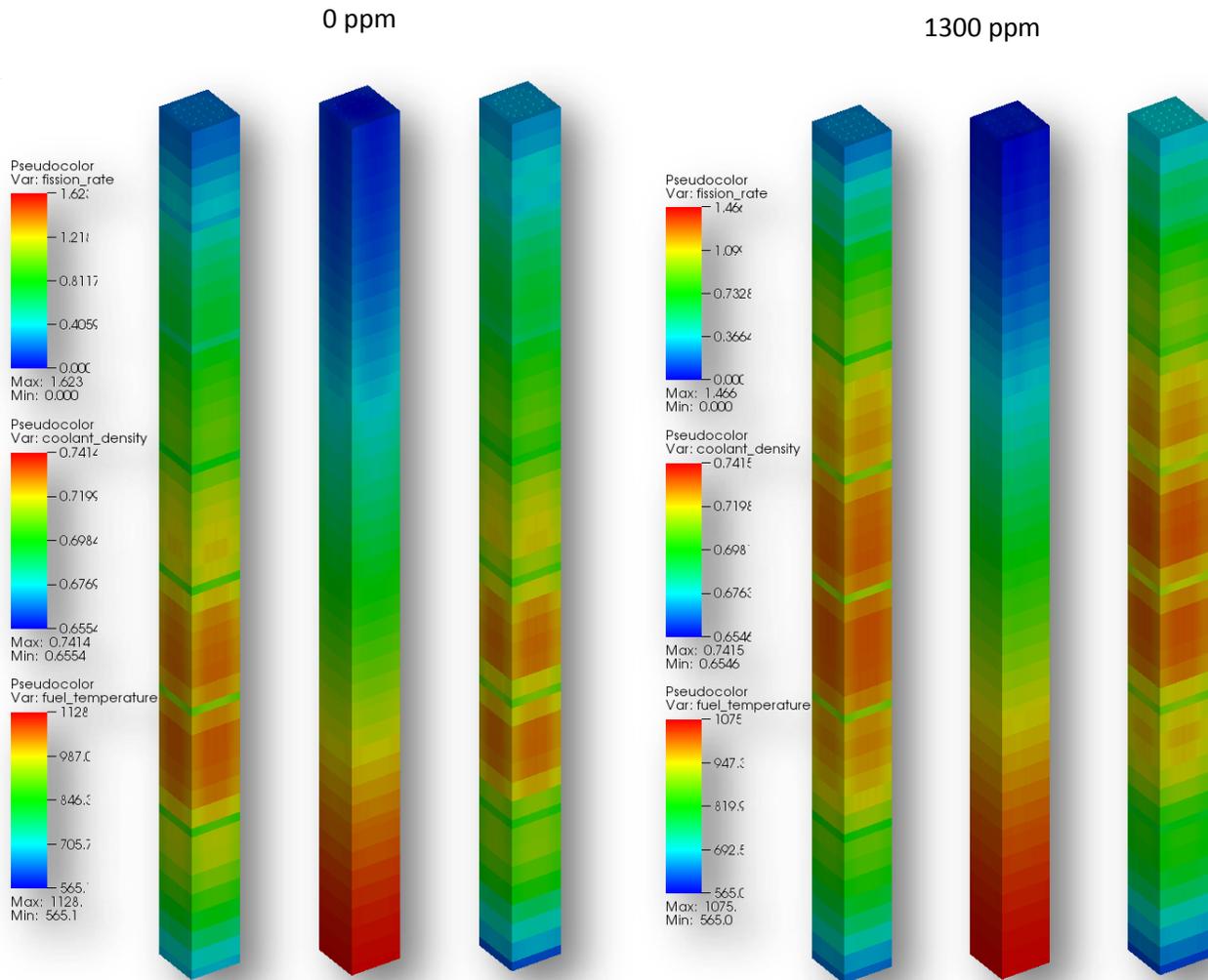
The fission rate and fuel temperature profiles for three different boron concentrations are shown in Figure 5-5. With T/H feedback, the fission rate shape is shifted lower in the core from the normal cosine-shaped distribution you would see with no T/H feedback. The reason for this downward shift in the fission rate is that the coolant density is higher at the bottom of the core, and the higher coolant density increases the neutron moderation.

As more boron is added, the additional neutron absorption counters the higher moderation, and less power shift towards the bottom of the core is observed.



**Figure 5-5 Axial Plot of Fission Rates at Different Boron Concentrations**

In order to look at 3D distributions, the coupled code also produces output in the form of SILO files. These files can be used by visualization tools, such as VISIT or PARAVIEW, to look at 3D plots of the output. Figure 5-6 shows an example of 3D distributions of the fission rate, coolant density, and fuel temperature for cases at 0 ppm and 1300 ppm boron. (Note that the results in this figure were generated with different code options and are not consistent with the results in the other figures.)



**Figure 5-6 Graphical Output of the Fission Rate, Coolant Density, and Fuel Temperatures at 0 and 1300 ppm Boron**

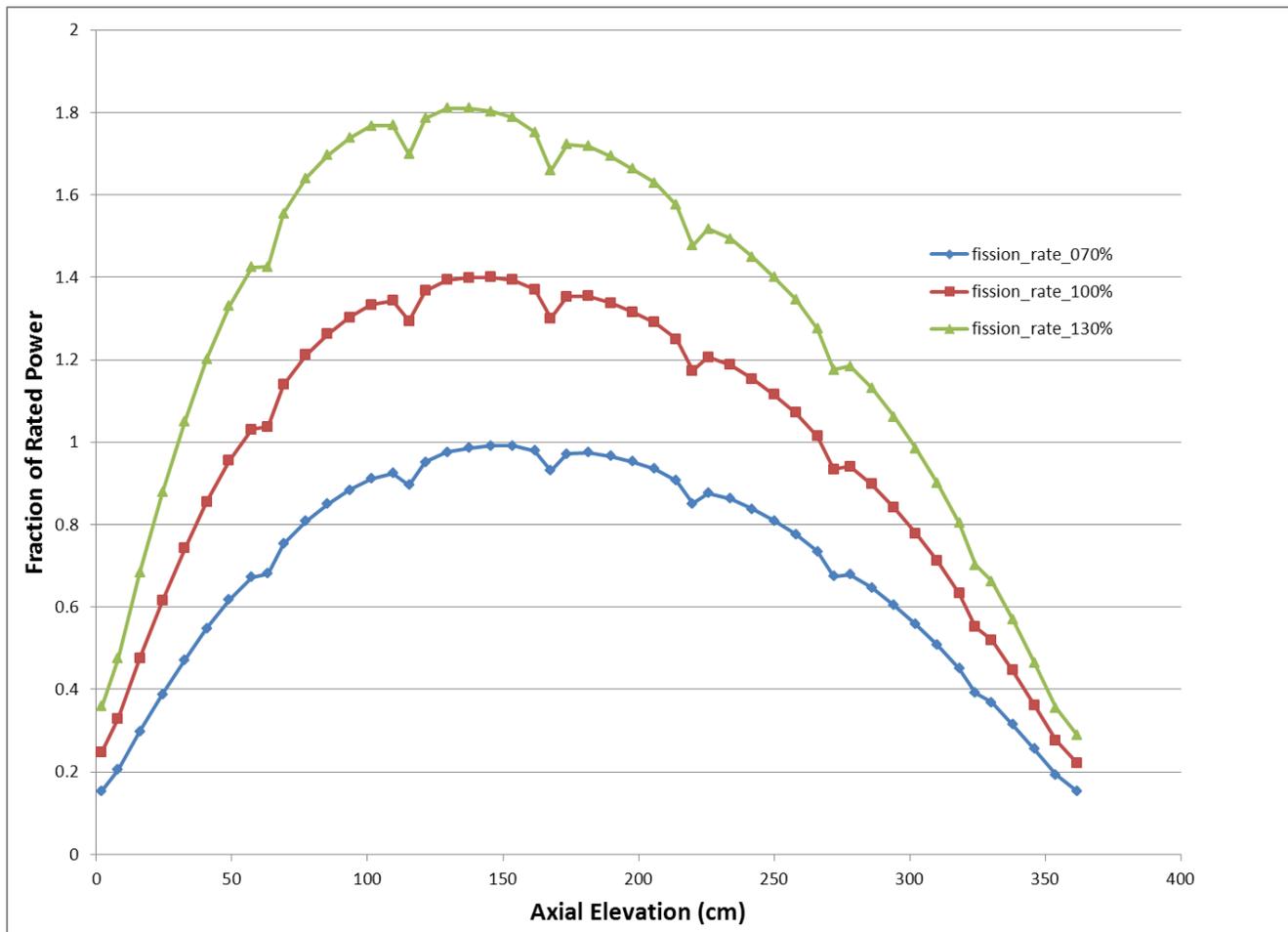
## 5.4 Power Perturbations

In order to see the effects of different power levels on the results, the nominal single-assembly case was run at four different power levels – 70, 100, and 130% power. The eigenvalues and wall-clock times for these cases are shown in Table 5-3.

**Table 5-3 Iteration Summary for Power Cases**

Power Level	Eigenvalue	Wall Time (HH:MM:SS)	Coupled Iterations
70%	1.24012	4:39:20	11
100%	1.23344	4:27:23	11
130%	1.22643	6:22:01	15

The fission rate profiles for the four power cases are shown in Figure 5-7. At higher power levels, the fission rate shape is shifted lower in the core from the normal cosine-shaped distribution you would see with no T/H feedback.



**Figure 5-7 Axial Plot of Fission Rates at Different Power Levels**

## 6. Conclusion and Recommendations

This Milestone demonstrates the successful multiphysics coupling of the Insilico neutronics code to the CTF thermal hydraulics code. The codes were coupled using the LIME and parallel data transfer was performed with the DTK library.

Several cases were run to demonstrate that the coupling was working properly. Cases were run at three different boron concentrations and four different power levels. All results show that the coupling is working as expected.

The successful coupling of Insilico and CTF provides a solid framework to pursue additional multiphysics coupling. One planned project is to add the Peregrine Fuel performance code [13] [14] to the existing Insilico and CTF coupling to create a 3-code coupling. Peregrine is an advanced fuel performance code developed on the MOOSE framework and will replace the simple fuel performance models currently used in CTF. The Peregrine coupling will be documented in the DOE-Reportable Milestone L2:MPO.P7.01 due on 7/31/2013.

Another project planned is to extend the capability of the Insilico and CTF coupling from a single-assembly problem to a PWR full-core problem (referred to as "Problem 7"). Problem 7 will combine the full-core capabilities developed in Problem 5 [15] with the thermal-hydraulic feedback capabilities developed in Problem 6. The result will be the capability to model a full-core problem with thermal-hydraulic feedback at the beginning of life (BOL). This Problem 7 capability will be documented in the DOE-Reportable Milestone L2:AMA.P7.02 due on 9/30/2013.

### 6.1 Recommendations

While this Milestone successfully meets all the objectives of coupling multiphysics codes together, there are still some areas that need to be studied further and/or improved upon. The AMA focus area makes the following recommendations for further work and improvements.

1. The SPN solver in Insilico needs to be evaluated further to determine if it is going to deliver the accuracy required to solve the CASL Challenge Problems. Determine if the method is adequate, and if not, can it be improved with better cross section and/or homogenization methods? If it is not adequate, what is the recommended neutronics solver?
2. For the SN solver in Insilico, a convergence study needs to be performed to determine what the recommended code options should be. This includes values for pincell mesh, axial mesh, spatial order (SC or LD), quadrature, energy group structure, scattering order, and convergence criteria. The effects on run-time should be evaluated along with the recommended code options.
3. A convergence study should also be repeated using the SPN solver in Insilico to determine if the SPN model gives satisfactory results. Compare run-times between the SPN models and the SN models.
4. Insilico currently has several eigenvalue solvers available (Arnoldi, Davidson, and Power iterations). Determine the recommended eigenvalue solver for coupled iterations.
5. A convergence study needs to be performed with CTF to determine what the recommended code options should be. This includes values for the number of rings in the fuel rod conduction model and the number of axial mesh.

6. The fuel temperatures predicted by CTF should be compared to the fuel temperatures predicted by the Peregrine fuel performance code. These comparisons will give confidence in whether the fuel rod conduction model and/or material properties are adequate in CTF.
7. The current iteration strategy using in this code coupling should be studied and improved upon, if possible. Some suggestions include:
  - a. Implement dynamic convergence criteria in the neutronics solution that changes with iteration number. Current industry nodal codes have found success using dynamic criteria that sets the neutronics convergence at each coupled iteration to 0.01 times the current convergence observed in the coupled solution. These criteria may reduce the number of neutronics iterations needed at the beginning of the calculation.
  - b. Implement and evaluate Anderson acceleration to the current fixed point convergence.
8. Implement boron searches so the user can search on the critical boron concentration.
9. Determine the optimal number of cores to run problems on. Is one core per fuel rod the most efficient strategy?
10. The current code coupling is only coupling with rod-averaged values at each axial elevation. Additional work should be performed to extend this to intra-rod distributions.

## 7. References

- [1] A. T. Godfrey, “VERA Core Physics Benchmark Progression Problem Specifications”, CASL-U-2012-0131-002, Rev. 2, March 29, 2013.
- [2] M.N. Avramova. CTF: A Thermal Hydraulic Sub-Channel Code for LWR Transient Analyses, User’s Manual, Pennsylvania State University, February 2009.
- [3] T. Evans, A. Stafford, R. Slaybaugh, and K. Clarno, “DENOVO: A New Three-Dimensional Parallel Discrete Ordinates Code in SCALE,” *Nuclear Technology*, **171**, 171–200 (2010).
- [4] G.G. Davidson, T.M. Evans, R.N. Slaybaugh, and C.G. Baker, “Massively Parallel Solutions to the k-Eigenvalue Problem,” *Trans. Am. Nucl. Soc.*, **103** (2010).
- [5] J. C. Gehin, et al., “Operational Reactor Model Demonstration with VERA: Watts Bar Unit 1 Cycle 1 Zero Power Physics Tests,” CASL-U-2013-0105-000, June 7, 2013.
- [6] R. Schmidt, N. Belcourt, R. Hooper, R. Pawlowski, An Introduction to LIME 1.0 and it’s Use in Coupling Codes for Multiphysics Simulation, Sandia National Laboratories, SAND2011-8524, November 2011.
- [7] S.R. Slattery, P.P.H. Wilson and R.P. Pawlowski, “The Data Transfer Kit: A Geometric Rendezvous-Based Tool for MultiPhysics Data Transfer,” *International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering (M&C 2013)*, Sun Valley Idaho, May 5-9, 2013.
- [8] TriBITS website: <https://code.google.com/p/tribits/>
- [9] R. Pawlowski, R. Bartlett, N. Belcourt, R. Hooper, R. Schmidt, A Theory Manual for Multiphysics Code Coupling in LIME Version 1.0, Sandia National Laboratories, SAND2011-2195, March 2011.
- [10] LIME website: <http://sourceforge.net/projects/lime1/>
- [11] Data Transfer Kit (DTK) website: <https://github.com/CNERG/DataTransferKit>
- [12] Watts Bar Unit 2 Final Safety Analysis Report (FSAR), Amendment 93, Section 4, ML091400651, April 30, 2009. <http://adamswebsearch2.nrc.gov/idmws/ViewDocByAccession.asp?AccessionNumber=ML091400651>
- [13] R. Montgomery, D. Sunderland, and R. Dunham, “Structural Mechanics Framework for VRI PCI Challenge Problem,” CASL Milestone Report L2.MPO.Y1.03, Anatech ANA-R-11-0844 Rev. 1, July 2011.
- [14] R. Montgomery, et al., “Peregrine: Validation and Benchmark Evaluation of Integrated Fuel Performance Modeling Using Test Reactor Data and Falcon,” CASL Milestone Report L1.CASL.P7.02, July 2013.
- [15] J. Gehin, et al., “Operational Reactor Model Demonstration with VERA: Watts Bar Unit 1 Cycle 1 Zero Power Physics Tests,” CASL-U-2013-0105-000, June 7, 2013.

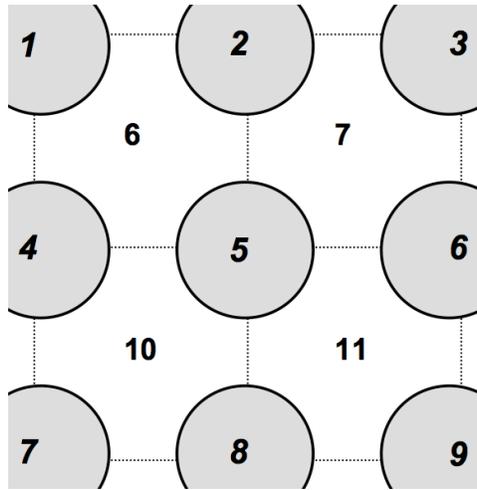
## Appendix A – Property Averaging

This appendix describes how to computing average values in CTF for transfer to Insilico.

### A.1 Average coolant properties

Insilico requires an average coolant density and average coolant temperature for each rod. Thus before values can be passed to Insilico, a weighted average of the four surrounding subchannel values must be calculated. These average values are computed in the `avg_rod_props` subroutine in CTF.

Figure A-1 illustrates how each rod within a PWR assembly surrounded by four subchannels.



**Figure A-1 Illustration of how a typical fuel rod is surrounded by four subchannels in a PWR square lattice fuel assembly.**

The average coolant density surrounding fuel rod “n” at axial level “j” is calculated as an area-weighted average of the liquid and vapor densities in the surrounding subchannels (i=1,4) per the following equation.

$$\rho_n = \sum_{i=1}^4 \omega_i (1 - \alpha_i) \rho_{liq, i, j} + \alpha_i \rho_{vap, i, j} \quad (1)$$

where  $\alpha_i$  is the volume fraction of the vapor phase, and the normalized weighting factor  $\omega_i$  is defined in terms of the four surrounding cross sectional areas  $A_i$  as

$$\omega_i = \frac{A_i}{\sum_{i=1}^4 A_j} \quad (2)$$

The average coolant temperature surrounding fuel rod “n” at axial level “j” is calculated as a mass-weighted average of the liquid phase temperatures in the surrounding subchannels (i=1,4) per the following equation.

$$\bar{T}_{n, j} = \sum_{i=1}^4 \psi_i T_{i, j} \quad (3)$$

where the normalized mass-weighting factor  $\Psi_i$  is defined in terms of the mass of liquid in the four surrounding subchannels ( $i=1,4$ )

$$\Psi_i = \frac{A_i \rho_{liq,i}}{\sum_{j=1}^4 A_j \rho_{liq,j}} \quad (4)$$

Note that the vapor-phase contribution is currently neglected because of the very large density difference between vapor and liquid and because of the relative insensitivity of cross sections to small changes in  $T$ .

## A.2 Average fuel rod properties

Insilico requires an average fuel temperature and average clad temperature for each rod at each axial level. These values must be computed (in the `avg_rod_props` subroutine) in CTF before they can be passed to Insilico.

CTF resolves the radial variation of temperature within a fuel rod using a finite difference grid as illustrated in Figure A-2. The number of nodes is problem dependent and defined by the user. In Figure A-2 there are five internal nodes together with the requisite fuel pellet surface node and the inner and outer cladding surface nodes (for a total of eight). CTF also has the option to model azimuthal variations with a coarse azimuthal grid that corresponds to the number of subchannels that surround the fuel rod (four in a square lattice).

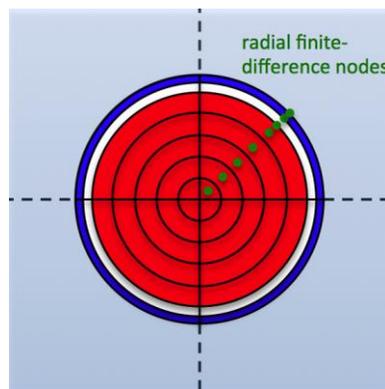


Figure A-2 Illustrative radial discretization of a fuel rod modeled by CTF

The average fuel temperature of fuel rod “ $n$ ” at axial level “ $j$ ” is calculated as an area-weighted average of the conduction node temperatures computed by the finite difference heat transfer solution within the fuel rod. This can be expressed as

$$\bar{T}_{f(n,j)} = \sum_{i=1}^{ic-2} \left( \frac{Ar_i}{Af_n} \right) \left[ \frac{1}{kla} \sum_{k=1}^{kla} T_{n,j,k} \right] \quad (5)$$

where

$Ar_i$  denotes the cross sectional area of radial finite difference node  $i$ ,

$Af_n$  denotes the total cross sectional area of the fuel,

$kla$  denotes the number of azimuthal sections being modeled, and

$ic$  denotes the total number of radial nodes in the fuel rod heat transfer model.

The average clad temperature of fuel rod “n” at axial level “j” is calculated in a similar fashion:

$$\bar{T}_{c(n,j)} = \sum_{i=ic-1}^{ic} \left( \frac{Ar_i}{Ac_n} \right) \left[ \frac{1}{kla} \sum_{k=1}^{kla} T_{n,j,k} \right] \quad (6)$$

where

$Ac_n$  denotes the total cross sectional area of the clad,

and all other terms are as previously defined above.

## Appendix B – Input File

This appendix contains the input listing for a PWR assembly. All of the input for CTF, Insilico, and the coupled code is created through the VERA Common Input.

### Input Listing

#### [CASEID]

```

title 'CASL Problem 6a'
=====
! Sample input for Problem 6 (Single-assembly with T/H feedback)
=====

```

#### [STATE]

```

power 100.0      ! %
tinlet 559.0     ! F
boron 600        ! ppmB
pressure 2250    ! psia

tfuel 900.0      ! K - 600K Not used with T/H feedback! set to 900K with feedback
modden 0.743     ! g/cc Not used with T/H feedback!

feedback on
sym full

```

#### [CORE]

```

size 1          ! 1x1 single-assembly
rated 17.67 0.6824 ! MW, Mlbs/hr
apitch 21.50
height 406.328

core_shape
1

assm_map
A1

lower_plate ss 5.0 0.5 ! mat, thickness, vol frac
upper_plate ss 7.6 0.5 ! mat, thickness, vol frac
lower_ref mod 26.0 1.0
upper_ref mod 25.0 1.0

bc_rad reflecting

mat he 0.000176
mat inc 8.19
mat ss 8.0
mat zirc 6.56
mat aic 10.20
mat pyrex 2.23
mat b4c 6.56

```

**[ASSEMBLY]**

```

title "Westinghouse 17x17"
npin 17
ppitch 1.260

fuel U31 10.257 95.0 / 3.1

cell 1      0.4096 0.418 0.475 / U31 he zirc
cell 100    0.561 0.602 / mod   zirc      ! guide tube
cell 200    0.561 0.602 / mod   zirc      ! instrument tube
cell 7      0.418 0.475 / mod   mod       ! empty location
cell 8      0.418 0.475 /      he zirc    ! plenum
cell 9      0.475 /            zirc       ! pincap
    
```

```

lattice FUEL1
200
  1 1
  1 1 1
100 1 1 100
  1 1 1 1 1
  1 1 1 1 1 100
100 1 1 100 1 1 1
  1 1 1 1 1 1 1 1
  1 1 1 1 1 1 1 1 1
    
```

```

lattice LGAP1
200
  7 7
  7 7 7
100 7 7 100
  7 7 7 7 7
  7 7 7 7 7 100
100 7 7 100 7 7 7
  7 7 7 7 7 7 7 7
  7 7 7 7 7 7 7 7 7
    
```

```

lattice PLEN1
200
  8 8
  8 8 8
100 8 8 100
  8 8 8 8 8
  8 8 8 8 8 100
100 8 8 100 8 8 8
  8 8 8 8 8 8 8 8
  8 8 8 8 8 8 8 8 8
    
```

```

lattice PCAP1
200
  9 9
  9 9 9
100 9 9 100
  9 9 9 9 9
  9 9 9 9 9 100
100 9 9 100 9 9 9
  9 9 9 9 9 9 9 9
  9 9 9 9 9 9 9 9 9
    
```

```

axial A1      6.050
  LGAP1      10.281
  PCAP1      11.951
  FUEL1      377.711
  PLEN1      393.711
  PCAP1      395.381
  LGAP1      397.501
    
```

```
grid END inc 1017 3.866
grid MID zirc 875 3.810
```

```
grid_axial
  END 13.884
  MID 75.2
  MID 127.4
  MID 179.6
  MID 231.8
  MID 284.0
  MID 336.2
  END 388.2
```

```
lower_nozzle ss 6.05 6250.0 ! mat, height, mass (g)
upper_nozzle ss 8.827 6250.0 ! mat, height, mass (g)
```

**[EDITS]**

! approximately 3in intervals in active fuel

```
axial_edit_bounds
  11.951
  15.817
  24.028
  32.239
  40.45
  48.662
  56.873
  65.084
  73.295
  77.105
  85.17
  93.235
  101.3
  109.365
  117.43
  125.495
  129.305
  137.37
  145.435
  153.5
  161.565
  169.63
  177.695
  181.505
  189.57
  197.635
  205.7
  213.765
  221.83
  229.895
  233.705
  241.77
  249.835
  257.9
  265.965
  274.03
  282.095
  285.905
  293.97
  302.035
  310.1
  318.165
  326.23
  334.295
```

338.105  
346.0262  
353.9474  
361.8686  
369.7898  
377.711

**[INSILICO]**

```
mat_library casl_comp.sh5
xs_library lib252_hetbondoneabs-noabssigg

max_delta_z 2.54
num_blocks_i 17
num_blocks_j 17
num_z_blocks 12
num_groups 23
num_sets 4
pin_partitioning true

mesh 4
dimension 3
eq_set sc
eigen_solver arnoldi
tolerance 1e-6
Pn_order 0

! eigenvalue_db:
  k_tolerance 1e-5
  L2_tolerance 1e-5
  energy_dep_ev true

! quadrature_db:
  quad_type qr
  azimuthals_octant 4
  polars_octant 4

! silo_db:
  silo_output p6
! silo_out_power true

! upscatter_db:
  upscatter_tolerance 1e-5

new_grp_bounds
8.2085e+05
1.1109e+05
5.5308e+03
1.8644e+02
3.7612e+01
3.5379e+01
2.7697e+01
2.1684e+01
2.0397e+01
1.5968e+01
7.1500e+00
6.7000e+00
6.3000e+00
1.0970e+00
1.0450e+00
9.5000e-01
3.5000e-01
2.0600e-01
1.0700e-01
5.8000e-02
2.5000e-02
```

1.0000e-02  
1.0000e-05

**[COBRATF]**

```
nfuel  3          ! number of fuel rings in conduction model
nc      1          ! conduction option - radial conduction
irfc    2          ! friction factor correlation default=2
dhfrac  0.02      ! fraction of power deposited directly into coolant
hgap    5678.3    ! gap conductance
eps0    0.001
oitmax  5
iitmax  40
gridloss END 0.9070 ! spacer grid loss coefficient
gridloss MID 0.9065 ! spacer grid loss coefficient
dtmin   0.000001
dtmax   0.1
tend    0.1
rtwfp   1000.0
maxits  10000
courant 0.8
```

**[COUPLING]**

```
epsk    5.0      ! pcm
rlx_power 0.5    ! power relaxation factor between coupled iterations
rlx_tfuel 1.0    ! fuel temperature relaxation factor
rlx_den   1.0    ! coolant density relaxation factor
maxiter  100
```