Development of COBRA-TF for Modeling Full-core, Reactor Operating Cycles

Robert Salko & Travis Lange
Oak Ridge National Laboratory

Vefa Kucukboyaci & Yixing Sung
Westinghouse Electric Co. LLC

Scott Palmtag, Core Physics

Jess Gehin
Oak Ridge National Laboratory

Maria Avramova
The Pennsylvania State University

March 26, 2015
DEVELOPMENT OF COBRA-TF FOR MODELING FULL-CORE,
REACTOR OPERATING CYCLES

Robert K. Salko and Travis Lange
Oak Ridge National Laboratory
salkork@ornl.gov and tlange@vols.utk.edu
Vefa Kucukboyaci
Westinghouse Electric Company
kucukbvn@westinghouse.com
Yixing Sung
Westinghouse Electric Company
sungy@westinghouse.com
Scott Palmtag
Core Physics
scott.palmtag@corephysics.com
Jess Gehin
Oak Ridge National Laboratory
gehinjc@ornl.gov
Maria Avramova
The Pennsylvania State University
mna109@psu.edu

Key Words: COBRA-TF, CTF, CASL, DNB, CRUD

ABSTRACT

CTF, the Pennsylvania State University version of COBRA-TF, has been adopted as the subchannel thermal hydraulic (T/H) capability in the core simulator being developed by the Consortium for Advanced Simulation of Light Water Reactors (CASL). This has resulted in significant development efforts to expand the applicability of CTF to performing high-fidelity, full-core, multi-physics simulations. These efforts have focused on addressing CASL challenge problems for pressurized water reactors (PWRs), which include modeling of departure from nucleate boiling and CRUD induced power shift. Developments specific to full-core modeling capabilities include creation of a preprocessor utility for the user-friendly, rapid generation of pin-cell-resolved PWR models and implementation of a domain-decomposition parallelization of the code solution algorithm. In preparation for modeling CRUD growth phenomena, a coupling interface has been developed for CTF and the code has been incorporated into a multistate driver, which allows for modeling entire reactor operating cycles (i.e. years of operation). A simple CRUD modeling capability has been coupled to the code through this driver for capturing CRUD growth over these long operational periods. This paper presents an overview of these new features and shows results of a full-core, pin-cell resolved simulation of a Westinghouse 4-loop PWR core during a loss-of-flow transient as well as an initial coupled T/H-CRUD simulation of a $17 \times 17$ assembly during a 15-month reactor operation cycle.
1. INTRODUCTION

CTF is the shortened name given to the version of COBRA-TF being jointly developed by Pennsylvania State University (PSU) and Oak Ridge National Laboratory under sponsorship of the US Department of Energy Consortium for Advanced Simulation of Light Water Reactors (CASL). CTF is a thermal hydraulic (T/H) simulation code designed for light water reactor (LWR) analysis. It uses a two-fluid, three-field modeling approach to model the independent behavior of liquid, droplets, and vapor. The code was adopted by the CASL program in 2012 for use as the T/H component in the Virtual Environment for Reactor Applications-Core Simulator (VERA-CS). Since its inclusion in this program, the code has undergone a number of software quality assurance improvements including:

- implementation of a source version control system;
- source code optimization for reduction in runtime and memory usage;
- integration into an automated build and testing system using TriBITS; and
- application of unit, integral, and validation testing.

In addition to these improvements, a series of new features have been added to the code in support of CASL modeling and simulation activities. The following major features were implemented into the code:

- creation of a PWR preprocessor utility capable of converting the VERA common input file into a native CTF input deck;
- implementation of a spatial decomposition parallelization of the solution algorithm using the Message-Passing Interface (MPI) for improving computational performance;
- addition of Visualization ToolKit (VTK) and Hierarchal Data Format (HDF5) file output features for aiding in visualization and post-processing of large data sets;
- development of a coupling interface and multistate driver for performing long depletion simulations; and
- coupling to neutronics and fuel performance codes.

Items 1–3 are discussed in Section 2 and Item 5 is discussed in Section 3. A demonstration of these capabilities is presented in Section 4.
2 DEVELOPMENTS FOR FULL-CORE ANALYSIS

2.1 PWR Preprocessor

Generating a CTF input deck requires the user to explicitly provide information for each rod and channel in the model (e.g. channel flow area, wetted perimeter, rod dimensions, etc.). The user must also specify how channels connect to one another and the rods. The input was designed this way so that CTF would be robust and capable of modeling a wide range of facilities (e.g. square and hexagonal fuel lattices and large models with lumped subchannels and rods). The drawback, however, is that the deck generation process becomes laborious and error prone with increasing model size.

If certain assumptions can be made about the model geometry, the amount of information required to generate a model becomes smaller. Rather than changing the CTF input interface directly, a separate program was written that builds the standard CTF input deck assuming that the model is for pressurized water reactor (PWR) geometry (i.e. square rod lattice). Furthermore, this PWR Preprocessor was designed to take model geometry and operating conditions in a map form, making it more intuitive.

![Diagram of PWR Preprocessor](image)

**Fig. 1.** Design of the PWR Preprocessor, used for generating large-scale CTF PWR models from reduced user input.

Rather than specifying individual channel and rod information, the user supplies a core map; general information about assemblies in the core (e.g. bundle pitch, assembly pitch, guide tube placement); and modeling options and operating conditions. The PWR Preprocessor will generate a CTF input deck at the pin-resolved level, meaning that each physical coolant channel in the bundle will be modeled by a separate subchannel in the CTF model. This results in full-core models containing roughly 60,000 subchannels.
As an example of the reduced input requirements, a quarter-core CTF model containing 200,000 lines of input was built using a PWR Preprocessor input deck of only 150 lines.

This utility was designed with two front-ends: one that reads from a set of plain text files; and an eXtensible Markup Language (XML) parser designed to work with the VERAIn common input file. Having the ability to build CTF models directly from the VERAIn input file, which provides a common interface to all VERA-CS codes, allows CTF to be seamlessly integrated into VERA-CS for doing coupled multi-physics simulations. Fig. 1 provides a flowchart representing the PWR Preprocessor algorithm.

2.2 Parallelization

The original application for CTF was model sizes in the hundreds of subchannel range and, therefore, the data structures and source were not optimized for large-scale problems. The high-resolution models intended to be run by CTF were originally too expensive for a full-core model. Therefore, it was necessary to speed up the code after it was adopted into the CASL program. The first stage of this two stage process involved executing a set of source code optimizations aimed at improving the serial performance of the code. Examples include using more efficient data storage techniques, performing loop-nesting optimization, and setting better initial conditions in the simulation. This serial optimization resulted in a nine times reduction in wall clock time for a model of a $17 \times 17$ assembly. These improvements allowed for running a quarter-core model of Watts Bar Unit 1 (460,000 computational cells), a Westinghouse 4-loop PWR containing 193 $17 \times 17$ fuel assemblies, in roughly 2 hours on one of the PSU development machines using a single Intel Xeon X5675 3.06 GHz processor to run the simulation.

The second stage of enhancements involved parallelizing the code. A domain decomposition approach targeting distributed memory machines and using the MPI was chosen for doing the parallelization. This approach required two main tasks: (1) implementing an algorithm that divides the model up into smaller pieces to be solved on separate processors and, (2) adding instructions for the separate processors to communicate with one another in order to maintain the integrity of the solution.

The domain decomposition step is performed external to CTF using the PWR Preprocessor. Users indicate that they would like to generate a model for running CTF in parallel, and the PWR Preprocessor produces an individual deck for each domain. The domain decomposition is based on assemblies, so each assembly in the core becomes a solution domain. This approach resulted in a more straightforward decomposition algorithm; however, it has the drawback that one processor must be used for each assembly in the core. For example, a 193 assembly model must be run with 193 processors. An example of how a 5-assembly core of $3 \times 3$ rod bundles would be decomposed is shown in Fig. 2, where rod and channel ownership is denoted with five different colors (e.g., Solution Domain 1 owns the blue rods and channels, Solution Domain 2 owns the red rods and channels, etc.).

To facilitate the communication of the separate solution domains, a perimeter of “ghost” channels is built around the boundary of each solution domain. These channels are referred to as “ghosts” since they are merely duplicates of a real channel that is owned by
Fig. 2. Example of domain decomposition for a model of five $3 \times 3$ rod bundles.

another solution domain. A limited solution is performed in the ghost channels with the main solution variables being obtained from the solution domain that owns the channel. Therefore, the ghost channels act as containers for data that the solution domain may pull from when it needs information about what is happening on the boundaries of the local model. This concept also applies for rods and gaps; there are real, owned rods and gaps and ghost rods and gaps in different solution domains.

Most of the solution can be done in parallel in this way, with data on the boundaries of each domain being periodically updated. However, the solution of the system of pressure correction equations requires special treatment of inter-domain communication. The Portable Extensible Toolkit for Scientific Applications (PETSc)$^{12}$ library was chosen for the task of solving the pressure matrix since it provides a robust collection of parallel iterative matrix solvers with good support for Fortran codes. Using this library, the pressure matrix is built and solved in parallel using the bi-conjugate gradient stabilized solver with Jacobian preconditioner.

A verification study was performed to ensure that the solution remained consistent between serial and parallel solves of different sized and shaped PWR models. Additionally, a scaling study done on the Oak Ridge Leadership Computational Facility Titan cluster demonstrated the parallel capability for a full-core model of Watts Bar Unit 1 modeled at pin-cell resolution. While this simulation takes about 24 hours when run in serial, it can now be been completed in under 20 minutes using the parallel version of the code.$^{3}$

2.3 Post-processing Capabilities

Originally, CTF only produced a set of text files giving rod and channel data in tabular form. Processing the data for analysis was a laborious task for models of significant size. To improve the user experience and make the data better organized and more accessible, two features were added to the code: (1) production of a set of VTK files...
and (2) production of a HDF5 file.

The VTK file is a plain text file that includes information about the mesh and simulation data. This file can be directly opened in a visualization tool such as ParaView.\textsuperscript{13} Because CTF did not originally have a concept of location of mesh cells (mesh cells are allowed to arbitrarily connect to one another), it was necessary to implement this information for a specific core design. The PWR Preprocessor generates the mesh location data and passes this to CTF for PWR models. CTF then passes this through to the VTK file. Currently, CTF writes both fluid and rod VTK files. The fluid file includes fluid pressure, enthalpy, temperature, void, velocity, density, vapor generation rate, and equilibrium quality. The rod VTK file includes clad outside temperature, clad inside temperature, fuel surface temperature, fuel centerline temperature, clad heat flux, and CRUD thickness. It is possible to write the VTK files at the end of the simulation or, if running a transient, periodically throughout the transient. Figs. 5, 6, and 8 were generated using the VTK capability implemented in CTF.

While the VTK file allows for visualization of the model, the HDF5 file allows for looking at actual data in a straightforward manner using the HDF5 viewer or custom-built parsing scripts. Data is identified by fuel rod indices and axal level in the assembly, making it possible to target specific computational cells in the model. Two benefits of the HDF5 file are that it is a binary file, which makes it more manageable than the larger plain text output files, and it allows for hierarchal organization of the data, making finding and processing simulation data easier. This capability was extended to store and read in simple restart data (power distribution) from the simulation. If a neutronics code were set up to write the power distribution using this standard VERA-CS data organization approach, CTF could directly read the detailed power distribution from an external code.

3 DEVELOPMENTS FOR CRUD MODELING

The CRUD challenges during PWR operations are related to thick CRUD deposits that have led to unexpected changes in core reactivity and power distribution (referred to as CRUD-induced power shift) and also to increased corrosion of the fuel cladding, which in some cases has led to fuel failures (referred to as CRUD-induced localized corrosion).\textsuperscript{14} In support of the CASL CRUD challenge problem, there was a need to incorporate CRUD modeling capabilities into VERA-CS. Capturing the complex CRUD deposition patterns caused by localized boiling and erosion effects requires multi-physics and high-fidelity modeling and simulation of coupled thermal hydraulics and coolant chemistry.\textsuperscript{15} By adding a CRUD modeling capability to a lower-resolution, faster-running T/H tool like CTF, it will be possible to target areas in the core with significant amounts of CRUD deposits in relatively short amounts of time. The more detailed and coupled T/H and coolant chemistry model can then be later used to “zoom-in” on these problem areas for higher-fidelity modeling.

3.1 T/H and CRUD Coupling

The goals of adding a CRUD modeling feature to CTF were to:
• capture CRUD growth and erosion on the fuel rods over long operation cycles and

• capture the thermal-feedback of the CRUD deposits on the fuel rod solution.

CTF already does a transient solution, but the physics it was developed to solve necessitate a highly resolved time scale (e.g. \(1 \times 10^{-6}\) to 0.1 second time steps). This makes doing long time scale simulations (months and years) impractical from a computational standpoint. Therefore, it was necessary to develop a capability in CTF for modeling long transients with very coarse time step sizes. This task was achieved by modeling the operational cycle as a string of pseudo-steady-state solves, with each solve using the results of the previous core state as the starting guess for the next state. The specific steps taken in the T/H-CRUD coupling are: (1) update CTF operating conditions for the current state, (2) do a pseudo-steady-state solve that defines the T/H behavior for the current operation state, (3) do a CRUD solve that models CRUD growth over the entire duration of that operation state, and (4) continue to the next operation state or exit if the simulation is finished. This approach required modification of the high-level CTF solution algorithm since the code did not originally consider different operational states. An external program, called the CTF multistate driver (discussed in Section 3.2), was created to implement this coupled CTF/CRUD modeling algorithm.

To capture the thermal feedback of the CRUD deposits, the CRUD layer was added to the CTF fuel rod conduction equation. This step primarily involved adding an additional node (thermal resistance) to the fuel rod radial conduction equation. Assumptions made in modeling CRUD thermal resistance included:

1. The CRUD has a constant thermal conductivity of 2 W/m-K, which is similar to that of uranium dioxide.\(^{16}\)

2. There is no axial or azimuthal conduction in the CRUD layer.

3. There is no heat generation in the CRUD layer.

4. The CRUD layer has no impact on clad oxidation or clad-water reaction heat generation.

5. There is no thermal contact resistance between the clad and the CRUD layer.

To model the CRUD behavior, a simple “surrogate” CRUD modeling tool (discussed in Section 3.3) was developed and implemented.

### 3.2 CTF Multistate Driver and Coupling Interface

The CTF multistate driver is a program, external to CTF, that reads operation period input data (e.g. boundary conditions and power distribution at different states) and activates steps in the CTF solution algorithm (i.e. initialization, setting boundary conditions, performing a pseudo-steady-state solve, and writing edits). The addition of this
capability made it possible to model months and years of reactor operation as a string of smaller pseudo-steady-state CTF solves.

To accomplish the task of driving the CTF solution from an external program in a clean and straightforward manner, a coupling interface was developed for CTF that can be used as a single point of communication for externally coupled codes. The multistate driver links to CTF through this interface exclusively; no internal CTF data or procedures are accessed directly. This interface consists of a collection of procedures that exist in one Fortran module and that have been thoroughly documented. This allows driver programs written in both C and Fortran languages to activate various segments of the CTF solution algorithm (e.g. initialization, steady-state solve, application of boundary conditions, writing of edits) independent from one another. The coupling interface additionally provides appropriate external access to internal CTF solution data (e.g. fuel rod temperatures, power, coolant temperature, and density). The coupling interface requires limited knowledge of the internal CTF solution implementation details, making coupling to CTF easier, while simultaneously protecting internal CTF data from external misuse. This coupling interface has additionally been used to couple CTF to neutronics codes in CASL.\textsuperscript{8}

The design of the multistate driver and coupling interface is shown in Fig. 3. In the future, the multi-physics coupling will be expanded to three codes—neutronics, T/H, and CRUD—by using the MPACT\textsuperscript{17} neutronics code to directly drive CTF and its CRUD capability in place of the multistate driver.

![Flowchart of the multistate driver created to drive CTF multistate simulations.](image-url)

**Fig. 3.** Flowchart of the multistate driver created to drive CTF multistate simulations.
3.3 CRUD Surrogate

A simple surrogate was developed to capture basic CRUD growth behavior and to facilitate setting up the multistate infrastructure outlined in Section 3.2. It is intended that the sub-grid version of the advanced Mamba CRUD chemistry code will be implemented as an optional alternative to this surrogate tool in the future.

The surrogate was designed to take reactor operation time, model size, and the core boiling distribution as inputs and return the local rod surface CRUD thickness and thermal resistance. The utility makes simplified assumptions about coolant chemistry, CRUD composition, and material properties in order to determine where CRUD distributes. This surrogate was based on the work of Zou et al. Major assumptions in the surrogate include:

1. All corrosion production released during an operational period deposit on the fuel rods at the end of the period.
2. Coolant impurity content can be estimated from Westinghouse plant data available in the Electric Power Research Institute report by Sawochka.
3. The steaming rate is the only determining factor for where CRUD will deposit.
4. CRUD density is 50% that of iron (3,500 kg/m$^3$).
5. CRUD erosion is negligible.

Future tasks for refining the CRUD surrogate include consideration for CRUD erosion effects and tracking CRUD composition (boron, lithium, and nickel content), which will be important when coupled to neutronics. The Mamba tool will already have consideration for these effects.

4 RESULTS

Results are presented in this section to demonstrate CTF’s capability to perform large-scale, high-resolution simulations of PWRs. Emphasis is placed on predictive capabilities related to the CASL departure from nucleate boiling (DNB) and CRUD challenge problems. Results of a $17 \times 17$ fuel assembly simulation, modeled over a long operational period, are shown to demonstrate the CTF/CRUD coupling. Finally, a brief discussion is provided describing how CTF has been incorporated in a coupled neutronics and T/H capability in VERA-CS.

4.1 DNB Modeling

CTF was assessed for its applicability in modeling the CASL DNB challenge problem in a recent study performed by Westinghouse. DNB is one of the safety-related challenge problems being addressed by CASL. DNB occurs when the fuel rod clad surface is dried out and overheated due to formation of a local vapor layer, causing dramatic
reduction in heat transfer capability and leading to possible fuel rod failure and release of fission products. Studies performed include:

- simulation of the loss-of-flow (LOF) transient in a 4-loop Westinghouse core,
- steady-state simulation of the most DNB-limited point in the main steam-line break (MSLB) transient in a 3-loop Westinghouse core, and
- simulation of a rod cluster control assembly (RCCA) bank ejection at hot-full power in a 3-loop Westinghouse core.

Each model was created at the fuel rod subchannel resolution level, leading to the 4-loop model having 8.5 million mesh cells and the 3-loop model having 6.9 million mesh cells. Models were created with use of the PWR Preprocessor. An example of the 4-loop radial mesh is presented in Fig. 4. The model consisted of 193 $17 \times 17$ Westinghouse fuel assemblies. Results of the LOF transient are summarized here to demonstrate CTF’s ability to model large-scale transients.

The power distribution was obtained from an external neutronics calculation and input to CTF through the PWR Preprocessor. The radial power distribution was assigned on a per-assembly and per-pin basis. The same assembly pin power distribution was applied to each assembly in the core; an assembly peaking factor was applied to each assembly. Since completion of this modeling effort, a coupling between CTF and the MPACT neutronics code has been achieved in VERA-CS, which is discussed further in Section 4.3. This capability can be used in the future for on-line capturing of multi-physics feedback effects.

The simulation was first run for 2 seconds to allow the flow-field to come to steady state. A transient forcing function was applied to the inlet flow boundary condition to simulate the coast down of the reactor coolant system (RCS) pumps, which started at 2 seconds into the transient. Similarly, a forcing function was applied to the core power to simulate the reactor trip, which occurred a few seconds after the RCS pumps tripped. Results show that minimum departure from nucleate boiling ratio (MDNBR) dropped to a minimum of about 1.6 before core power was reduced. The critical heat flux used to calculate the MDNBR was obtained using the W-3 correlation. The transient was run for a total of 12 seconds.

Transient visualizations of liquid velocity (Fig. 5) and coolant temperature (Fig. 6) were produced from this simulation. In Figs. 5 and 6, the time denotes the elapsed time since the RCS pumps trip. At $t=0.0$ seconds, the liquid velocity is higher than 4.2 m/s throughout the core, which explains its uniform appearance in the first figure. As the transient progresses, flow reduces in the bottom of the core first and then in the top of the core near the end of the transient due to fluid inertia. The core temperature response exhibits the opposite behavior, with liquid temperature increasing in the upper portion of the core and migrating downwards throughout the transient. The coolant temperature is capped at the saturation temperature in the upper region of the core early on in the transient.
Fig. 4. Radial mesh of the 4-loop Westinghouse PWR model used for LOF transient simulation.

Fig. 5. Liquid velocity distribution during LOF transient.
The LOF transient was run using 193 cores on one of Westinghouse’s computational facilities. The total wall time to complete the transient was roughly 8 hours. This simulation time was longer than the previously discussed parallelization results for three reasons:

1. A more refined axial mesh was employed for this case than previous scaling studies ($\sim 3 \times$ refinement).
2. A longer running transient was modeled instead of a pseudo-steady-state simulation ($\sim 10 \times$ iterations).
3. VTK files were produced throughout the transient instead of once at the end, which is a costly serial process ($\sim 4 \times$ VTK writes).

As a result of this study, the parallelization strategy will be further investigated in the future with intention of making the approach more robust (allowing more flexible core configurations) as well as making the code more computationally efficient for finer axial meshes and I/O-dominated simulations. A more detailed analysis of the results for the LOF, MSLB, and RCCA bank ejection reactivity insertion accident (RIA) transients is available in the Westinghouse technical report.\textsuperscript{21}

4.2 CRUD Modeling

For demonstration of the CRUD modeling feature that has been implemented into CTF, a single Westinghouse-style $17 \times 17$ assembly was modeled over a cycle of 15 months,
broken into 3 month states. The power distribution was obtained from a coupled CTF-neutronics simulation of the problem, run to steady state for the initial set of boundary conditions at State 1; the coupled simulation was not run at each state in the cycle. The boundary conditions and power distribution were kept at their nominal values throughout the entire duration of the cycle simulation. Boundary conditions included an inlet mass flow rate of 85.98 kg/s, inlet temperature of 300 °C, and outlet pressure of 15.513 MPa. Since the assembly specification was for a fresh, non-peaked-power assembly, the inlet temperature in the CTF model was increased over the nominal value in order to cause subcooled boiling and initiate CRUD growth.

Fig. 7 shows a set of three isometric visualizations of the rod bundle and selected simulation data. A box cut out was made in the top corner of the assembly to provide a view of the inside of the assembly. The left-most image shows the rod heat flux distribution; the middle image shows the rod steaming rate; and the right-most image shows the CRUD distribution at the end of the 15-month cycle. The neutronics core power distribution was cosine shaped, peaking in the center of the bundle. Inspection of the figure reveals local power depressions due to the presence of the spacer grids. The bundle steaming occurs slightly downstream of the bundle center due to the combination of high heat flux and low coolant subcooling. From the right-most image, it is evident that the CRUD deposits exactly where nucleate boiling occurs.

Because the final CRUD thickness was very small, there was a negligible impact on rod temperature for this case. The simulation was run for a longer period of time in order to grow CRUD to a thickness of 30 µm. This resulted in a 3.7 °C increase in fuel centerline temperature in fuel regions experiencing maximum deposits. Additional verification studies have been done to ensure correct behavior of the surrogate and the integration of the CRUD layer into the CTF conduction equation.7
4.3 Coupled Full-Core Modeling

The goal of developing VERA-CS is to create a high-fidelity LWR simulation tool capable of modeling multi-physics phenomena, including neutronics, thermal hydraulics, fuel performance, and CRUD growth dynamics. In support of this goal, several neutronics-T/H coupling strategies have been investigated and employed, including indirect couplings between CTF-Insilico and CTF-MPACT using the Tiamat driver being developed in CASL\(^9\) and a direct coupling of the MPACT neutronics code and CTF. The latter coupling has been demonstrated by performing a full-core simulation of Watts Bar Nuclear Plant, Unit 1.\(^8\)

The model included the region between lower and upper assembly nozzles and was divided into 58 axial planes. The CTF fluid mesh was created at subchannel resolution. Volumetric averaged pin power and temperature were passed between the two codes for each axial level of the rod. Coolant temperature and density were also passed from CTF to MPACT, with a remapping of the data being performed to convert the CTF coolant-channel-centered data to pin-centered mesh data needed by MPACT. Fig. 8 is a visualization of fuel temperatures in the core after a steady-state simulation performed at beginning of cycle (BOC).

![Fig. 8. Calculated fuel centerline temperatures at BOC in Watts Bar Unit 1 predicted with coupled CTF/MPACT simulation.](image)

5 CONCLUSIONS

A large body of development work has been done on the subchannel code, CTF, as part of the CASL program. This work has focused on improving code quality through implementation of a change control process and automated, continuous unit, regression, and validation testing. Work has been done to develop the code for performing pin-cell-level modeling of full PWR cores. This required developing a PWR Preprocessor utility and performing a serial optimization and full parallelization using MPI to expand the
applicability of CTF to full-core, pin-cell-resolved models. Results of a full-core DNB study have demonstrated these new capabilities. This study targeted costly bottlenecks in the parallelization algorithm that will be investigated in the future. In support of the CASL CRUD challenge problem, an infrastructure has been designed for coupling CTF to CRUD simulation tools. Specifically, a coupling interface was created for CTF to allow external manipulation of the CTF solution algorithm and accessing of internal CTF data. A multistate driver was created to drive CTF over long operational periods as a string of pseudo-steady-state solves through this coupling interface. For capturing CRUD growth behavior, a simple CRUD modeling tool was created and coupled to CTF. In the future, this CRUD modeling tool will be improved, and the more advanced Mamba CRUD code will be used as an optional alternative. This multistate driver has laid the groundwork for the future coupling of MPACT (neutronics), CTF (T/H), and Mamba (CRUD).

NOTICE

This research was supported by the Consortium for Advanced Simulation of Light Water Reactors (www.casl.gov), an Energy Innovation Hub (http://www.energy.gov/hubs) for Modeling and Simulation of Nuclear Reactors under U.S. Department of Energy Contract No. DE-AC05-00OR22725. This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

Westinghouse Electric Company LLC assumes no liability with respect to the translation or use of or for damages resulting from the translation or use of the information contained herein. Further, Westinghouse Electric Company LLC makes no warranty or representation, express or implied, with respect to the accuracy or completeness of the translation or the usefulness of the information contained herein.

ACRONYMS

BOC beginning of cycle
CASL Consortium for Advanced Simulation of Light Water Reactors
COBRA-TF Coolant Boiling in Rod Arrays—Two Fluids
CTF Pennsylvania State University version of COBRA-TF
DNB departure from nucleate boiling
HDF5 Hierarchal Data Format
LOF loss-of-flow
LWR light water reactor
MDNBR minimum departure from nucleate boiling ratio
MPI Message-Passing Interface
MSLB main steam-line break
PETSc Portable Extensible Toolkit for Scientific Applications
PSU Pennsylvania State University
PWR pressurized water reactor
RCCA rod cluster control assembly
RCS reactor coolant system
RIA reactivity insertion accident
T/H thermal hydraulic
VERA Virtual Environment for Reactor Applications
VERA-CS Virtual Environment for Reactor Applications-Core Simulator
VTK Visualization ToolKit
XML eXtensible Markup Language

REFERENCES


