

Improvements to CTF for modeling of Boiling Water Reactor Geometry and Operating Conditions



**Approved for public release.
Distribution is unlimited.**

R. Salko
B. Hizoum
B. Collins
M. Asgari

September 30, 2020

DOCUMENT AVAILABILITY

Reports produced after January 1, 1996, are generally available free via US Department of Energy (DOE) SciTech Connect.

Website: www.osti.gov/

Reports produced before January 1, 1996, may be purchased by members of the public from the following source:

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
Telephone: 703-605-6000 (1-800-553-6847)
TDD: 703-487-4639
Fax: 703-605-6900
E-mail: info@ntis.gov
Website: <http://classic.ntis.gov/>

Reports are available to DOE employees, DOE contractors, Energy Technology Data Exchange representatives, and International Nuclear Information System representatives from the following source:

Office of Scientific and Technical Information
PO Box 62
Oak Ridge, TN 37831
Telephone: 865-576-8401
Fax: 865-576-5728
E-mail: report@osti.gov
Website: <http://www.osti.gov/contact.html>

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Reactor and Nuclear Systems Division

Improvements to CTF for modeling of Boiling Water Reactor Geometry and Operating Conditions

R. Salko
B. Hizoum
B. Collins
M. Asgari

Date Published: September 30, 2020

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, TN 37831-6283
managed by
UT-Battelle, LLC
for the
US DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

CONTENTS

LIST OF FIGURES	v
LIST OF TABLES	vii
ACRONYMS	ix
ABSTRACT	1
1. INTRODUCTION	3
2. NUMERICAL IMPROVEMENTS	5
2.1 Description of Hatch Model	5
2.2 Implementation of Outer Iteration Loop	5
2.3 Improvements to Pressure Matrix Solver	8
2.4 Pressure Balance Iteration Loop	9
3. GEOMETRY MODELING IMPROVEMENTS	13
3.1 Inlet Orifice Map	13
3.2 Axially Varying Water Rod Geometry	16
3.3 Large Radius Channel Box Corner Rounds	19
3.4 Addition of Support for Quarter Symmetry Models	20
4. SUMMARY	21
5. REFERENCES	23

LIST OF FIGURES

1	Flowchart of the original governing equation solution.	7
2	CTF Outer iteration flowchart.	8
3	Flowchart of how pressure balance iteration loop is implemented in CTF for BWR models. .	11
4	Improvement in the performance of the pressure balance loop after implementing improvements to the algorithm.	12
5	Total simulation time for the first statepoint of the full core Hatch model after implementing pressure balance loop improvements.	13
6	Hatch results with no inlet orifices.	15
7	Hatch results with inlet orifices applied.	15
8	Example of large water rod variation in CTF subchannel mesh.	17
9	Schematic of rod-bundle region used to test axially varying water rod modeling approach. .	17
10	Axial liquid velocity profile in subchannel 40 of a BWR bundle model for three different approaches for modeling axial water rod diameter variation.	18
11	Axial vapor void profile in subchannel 40 of a BWR bundle model for three different approaches for modeling axial water rod diameter variation.	18
12	Illustration of the impact of a large channel box corner round radius on corner and side subchannel geometry.	19

LIST OF TABLES

ACRONYMS

BiCGStab biconjugate gradient stabilized
BWR boiling water reactor
CASL Consortium for Advanced Simulation of Light Water Reactors
DOE US Department of Energy
FOA funding opportunity announcement
GE General Electric
HPC high-performance computing
INL Idaho National Laboratory
MPACT Michigan Parallel Characteristics Transport Code
NCSU North Carolina State University
PETSc Portable, Extensible Toolkit for Scientific Computation
PWR pressurized water reactor
T/H thermal hydraulics
VERA Virtual Environment for Reactor Applications

ABSTRACT

This report documents work done to improve CTF modeling of boiling water reactor (BWR) designs during FY 2020. Several improvements were made to the geometry modeling capabilities, including adding support for modeling of inlet flow orifices, axially varying large water rods, channel boxes with large corner round radii, and quarter symmetry models. Work was also done to improve on the numerical stability and run-time performance of CTF in Virtual Environment for Reactor Applications (VERA) for core-scale BWR problems. An outer iteration loop was implemented that ensures mass and energy are conserved before moving on to the next time step in the solution. The ability to use a direct solver for the pressure matrix was also added, which reduces the matrix solution error and improves code stability and run-time performance. Improvements to the pressure balance iteration loop were needed to allow CTF to be used in a coupled VERA solution of a model of Hatch that was developed. Finally, various improvements have led to a 2× reduction in the total VERA walltime for the Hatch model.

1. INTRODUCTION

CTF is the thermal hydraulics (T/H) feedback code used in the Virtual Environment for Reactor Applications (VERA) core simulator capability that was developed by the Consortium for Advanced Simulation of Light Water Reactors (CASL). As part of a 2 year US Department of Energy (DOE) funding opportunity announcement (FOA), VERA is being expanded to support the modeling and simulation of boiling water reactor (BWR)s. A significant portion of this effort will require further CTF development to ensure that geometry and operation conditions of common BWRs can be successfully modeled. Although some features, such as the pressure balance iteration loop and support for generic BWR geometry, were added during CASL [4], these capabilities needed to be expanded and improved to better support actual models that will be considered in this project.

This report provides a summary of the activities completed during the first year of this FOA project in relation to CTF, addressing some of the needs that were identified in an earlier milestone report [3]. This report summarizes the geometry- and numerical-related needs for modeling core-scale, pin-resolved BWR models. Improvements to code two-phase modeling accuracy were completed as part of a different project and are documented in Salko, Kumar, and Hizoum [5]. Another important aspect to consider is bypass flow modeling. An activity was undertaken by North Carolina State University (NCSU) to test options for bypass flow modeling between assemblies and through the large water rods, which was documented in a previous milestone report [1]. This report is broken into two sections: one that discusses improvements to the numerical solution that improves the stability and performance of CTF in VERA and one that discusses the geometry-modeling improvements, which add features needed to model BWR-specific geometry previously unavailable in CTF.

2. NUMERICAL IMPROVEMENTS

2.1 Description of Hatch Model

The Hatch core was used for core-scale testing of VERA and CTF during this work. The Hatch model is a quarter symmetry model that contains 140 solved assemblies. Each assembly is a General Electric (GE) 9 design, which is an 8×8 rod lattice. When performing a VERA simulation, a single statepoint of Cycle 1 is being modeled (one coupled solution of CTF and Michigan Parallel Characteristics Transport Code (MPACT)) [2]. The rated power is 2,436 MW thermal and the rated flow is 1,763 kg s⁻¹.

2.2 Implementation of Outer Iteration Loop

The objective of the outer iteration is to remove the constraint on time step size so that the solution of the linearized equations can be sufficiently representative of the nonlinear equations. This will provide an acceptable level of accuracy, providing that the momentum, mass, and energy equations are discretized in semi-implicit form for their own dependent variables. In the current linearization, the momentum equation is only partially implicit for terms that involve pressure propagation, such as interfacial and wall shear. This type of discretization is better than the explicit discretization because it eliminates the sound speed from the Courant stability limit, leaving what is commonly referred to as the *material Courant stability*; however, the time step must still be controlled, and the system can be unstable due to the explicit discretization of source terms and—more importantly—the advection flux. The goal of the CTF numerical improvements is to enhance the semi-implicit method by treating the convection term in the momentum equation implicitly and to use the outer iteration to ensure that the velocity field is satisfying the mass equation before marching in time. This will allow the code to violate the Courant limit and help to reduce the overall run time for a core simulation with slow transient. In this phase of work, only the outer implementation will be discussed; the improvement to the semi-implicit form will be part of future work. In the original CTF implementation, the outer iteration was omitted; only one iteration is allowed (i.e., ICE method), as shown in Figure 1. To avoid excessive discretization errors, the code examines the rate of the change ($x^{n+1} - x^n$) for the primary dependent variables from one time step to another and adjusts the time step accordingly to maximize the accuracy (CFL limit) and minimize the computer time. The obtained updated velocity fields will satisfy the mass and energy equations if time step is kept sufficiently small. To minimize the computer run time and improve stability, the authors reimplemented the outer iteration. In its implementation, some routines were rearranged. These are indicated in red in Figure 2. The green is the check of the rate of change for the dependent variables, which is very important for a fast transient to control the time step. This option is not currently implemented and is scheduled for future work. The acceptance criteria of the outer iteration that determine when the pressure satisfies the mass equation or when convergence is established are defined as:

$$\max \left(\frac{P_{i,j}^{n+1,m} - P_{i,j}^{n+1,m-1}}{P_{i,j}^{n+1,m-1}} \right) < \epsilon_P, \quad (1)$$

$$\max \left(\alpha_{v,i,j}^{n+1,m} - \alpha_{v,i,j}^{n+1,m-1} \right) < \epsilon_{\alpha_v}, \quad (2)$$

$$\max(\alpha_{e,i,j}^{n+1,m} - \alpha_{e,i,j}^{n+1,m-1}) < \epsilon_{\alpha_e}, \quad (3)$$

$$\max((\alpha_v h_v)_{i,j}^{n+1,m} - (\alpha_v h_v)_{i,j}^{n+1,m-1}) < \epsilon_{\alpha_v h_v}, \quad (4)$$

$$\max((\alpha_e h_l)_{i,j}^{n+1,m} - (\alpha_e h_l)_{i,j}^{n+1,m-1}) < \epsilon_{\alpha_e h_l}, \quad (5)$$

where i, j designates the subchannel location, n is the time iteration counter, and m is the outer iteration counter. The primary dependent variables, h_v , h_l , α_v , α_e , and p are vapor enthalpy, liquid enthalpy, vapor void, droplet void, and pressure, respectively. The acceptance criteria are given as $\epsilon_p = 10^{-03}$ to 10^{-05} , $\epsilon_{\alpha_e} = \epsilon_{\alpha_v} = 10^{-02}$, $\epsilon_{\alpha_e h_l} = \epsilon_{\alpha_v h_v} = 2Btu/lbm$ which corresponds to 1 K error in temperature. As shown in Figure 2, the code will step into the outer iteration loop with a guessed velocity field and then solve for the change in the pressure equations. Then, it will back substitute to obtain the change in the other dependent variables and evaluate the dependent variable at each outer iteration step (m) to be used for the next iteration. Within each outer iteration, the code will check for convergence in accordance with criteria expressed by Eqs. 1–5. If all convergence criteria are met, then the code will update the top and bottom boundary conditions and update old time dependent variables. If not, then it will back up solutions to the old time one, reduce the time step, and repeat the iteration.

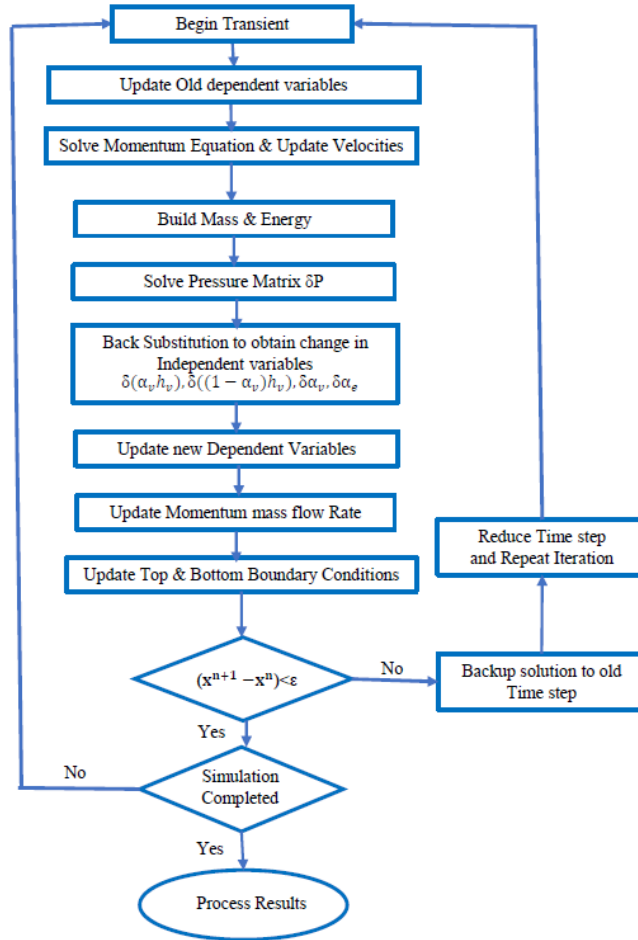


Figure 1. Flowchart of the original governing equation solution.

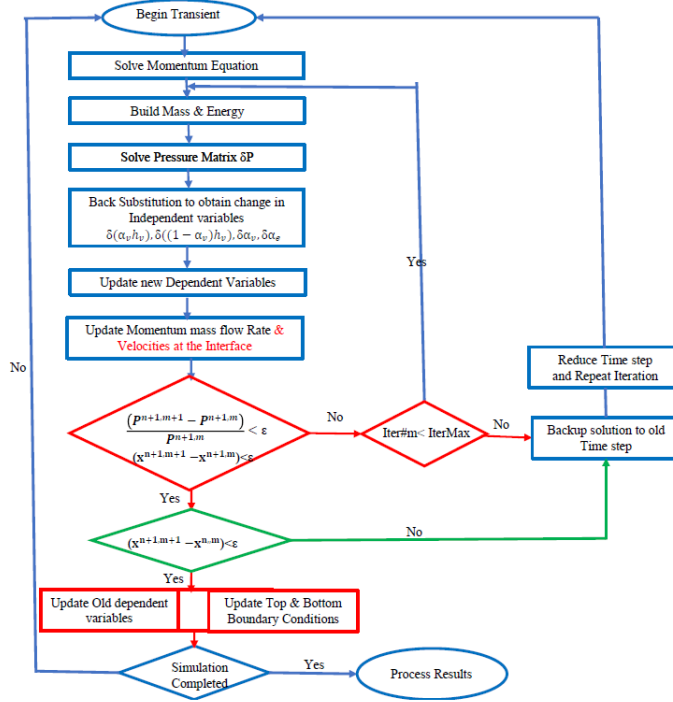


Figure 2. CTF Outer iteration flowchart.

2.3 Improvements to Pressure Matrix Solver

CTF uses an operator split approach to solve the governing equations in which the momentum equations are solved in the first step, and then the mass and energy equations are solved in the second step. The mass and energy equations are linked to the momentum equations through the solution of a single pressure correction matrix. The pressure in each cell in the model is corrected so that the new velocity distribution will conserve mass and energy. Because adjacent cells will communicate through the cell connections, all the pressure correction equations must be solved simultaneously, which results in a large matrix solve. Because of its size, the pressure matrix solve can consume a large percentage of the CTF walltime.

To counter this issue, CTF links to the advanced matrix solvers available in the external packages: Futility and Portable, Extensible Toolkit for Scientific Computation (PETSc). These solvers are optimized to perform direct or iterative solutions of large matrices by using multiple processors that use a wide variety of techniques. The default solver used in CTF is the biconjugate gradient stabilized (BiCGStab) solver, which is a Krylov subspace method that iteratively solves the matrix. This is an optimal choice for pressurized water reactor (PWR) models. However, initial studies have demonstrated that the small amounts of solver errors that result from performing an iterative solution can lead to numerically unstable solutions when highly voided flows are modeled. This requires solver tolerances to be significantly reduced, which further increases code walltime.

Studies on smaller, single-assembly problems have revealed that using a direct solver, which has no solver error, consistently leads to more stable numerical solutions and tends to outperform the iterative solver

when many iterations must be taken by the iterative approach. However, as the problem size increases to the size of a full core, the direct solver performance degrades rapidly. The SuperLU solver offered by PETSc can run in parallel, but it still lags in performance compared with the BiCGStab iterative solver for full core models.

One unique advantage of how BWR models are set up in CTF is that each assembly is nearly independent from other assemblies in the model. Assemblies only communicate through a pressure balance outer iteration loop used to balance inlet flow, which is discussed in Section 2.4. Taking advantage of this feature, a separate pressure matrix can be solved for each independent assembly because there are no connections that would allow the cell pressure in one assembly to affect a cell in another assembly. By setting up the solution in this way, it is possible to use the SuperLU direct solver for the pressure matrix solution, which will have the same performance as if it were running on a single assembly model.

To test performance, the Hatch model was run on the Idaho National Laboratory (INL) Falcon high-performance computing (HPC) system. The CTF model uses one processor per assembly. The total time spent in the pressure matrix solve when using BiCGStab is 163 seconds. When using the SuperLU solver, total pressure matrix time is 94 seconds. Not only is the pressure matrix solve time reduced by a factor of 1.7 but the direct solver approach also has the benefit of being less prone to numerical oscillation. To look at the scaling of the solve time, one assembly from the Hatch core was run by using the SuperLU solver. The total number of code iterations were checked to ensure that they matched the total number of iterations for the full core case. The solution was also compared for the single assembly case and a single assembly from a full core case. The single assembly total pressure matrix solve time was 65 seconds, indicating that about 30 seconds more time is spent in the pressure matrix solve when scaled up to a full core problem. These losses can likely be attributed to different HPC nodes that perform differently from others. Regardless, the lost 30 seconds represents a small portion of the total CTF solve time of 850 seconds. It is an even smaller fraction of the total VERA walltime.

2.4 Pressure Balance Iteration Loop

As discussed in Section 2.3, CTF models BWR cores as a collection of separate assemblies that are nearly independent from one another. In reality, these assemblies connect at the lower and upper plenum regions of the core, but these are not explicitly modeled by CTF. Therefore, to ensure that the pressure is equal in all assemblies, an outer iteration loop is used to adjust inlet flows to assemblies to equalize pressure drop [4]. An iteration of the pressure balance loop is a CTF solve. A linear relationship is formed between the assembly pressure drop and inlet mass flow rate to the assembly to estimate what inlet mass flow rate must be in every assembly so that all assemblies experience an equal pressure drop.

The inlet assembly flow rate is predicted by using the following:

$$\dot{m}_{k+1} = A + B\Delta P_{k+1}. \quad (6)$$

Here, \dot{m} is the bundle inlet mass flow rate, k is the pressure balance loop iteration number (k being the most recently completed iteration and $k + 1$ being the next iteration to be performed), ΔP is the bundle pressure drop, and B is the ratio of change in the pressure drop to change in inlet mass flow rate, or:

$$B = \frac{\dot{m}_{k-1} - \dot{m}_k}{\Delta P_{k-1} - \Delta P_k}. \quad (7)$$

The A coefficient is calculated by using the last calculated pressure drop and inlet mass flow rate substituted into Eq. 6:

$$A = \dot{m}_k - B\Delta P_k. \quad (8)$$

One equation will be created for each assembly in the core. The ΔP_{k+1} should be the same for every bundle in the core. This is the objective of the pressure balance iteration loop. This will lead to a total number of $N + 1$ unknowns, where N is the number of assemblies in the model. There is one \dot{m} for every assembly and the final core ΔP . The final equation needed to close the system is the mass conservation equation, which can be stated:

$$\dot{m}_{\text{tot}} = \sum_1^N (\dot{m}_{k+1}). \quad (9)$$

The total core flow rate, \dot{m}_{tot} , is known from user input, N is the total number of assemblies, and the \dot{m}_{k+1} values are the new assembly inlet mass flow rates being solved for in the system of equations.

While performing full core testing on the Hatch model, two issues were discovered. First, CTF sometimes crashed because the pressure drop/inlet flow relationship was predicting negative inlet mass flow rates for some conditions. Second, CTF was taking many pressure balance loop iterations for one coupled solution.

The cause of the first issue was the linear relationship between the pressure drop and mass flow rate becoming unrealistic for certain conditions. The two most recent pressure loop outer iterations are used to calculate the mass flow rate /pressure drop relationship. No iterations before the latest two are included in the relationship. In a coupled solution, the flow change can actually be zero if the two consecutive iterations fall over a power change. When coupled to MPACT, the pressure balance loop will perform several CTF solves to obtain a converged T/H solution, which is then returned to MPACT. MPACT will solve a power distribution for this new T/H distribution, which is then provided to CTF. The first CTF solve will result in a pressure drop change due to the power change, but the flow will be the same as it was in the previous solve of CTF before the power changed. This can lead to divide-by-zeros or very large values of the B coefficient, which can force inlet flow to unrealistic values that must be corrected with additional CTF solves. In some cases, the inlet flows can even become negative, which can crash the solution.

To solve this issue, four improvements were implemented into the algorithm.

1. The value of the inlet mass flow rate change is bounded to a 20 % in one iteration. This helps to make the solution more stable.
2. The mass flow rate /pressure drop relationship is not updated when a power change occurs due to MPACT passing in a new power distribution. This provides no information about how mass flow rate and pressure drop are related, so it should not be used to update the relationship.
3. The inlet mass flow rate distribution is left to its last value when a new CTF solve is performed. Previously, the inlet mass flow distribution was being reset to uniform flow every time CTF was called. The inlet mass flow rate distribution does not change by a great deal after the first few

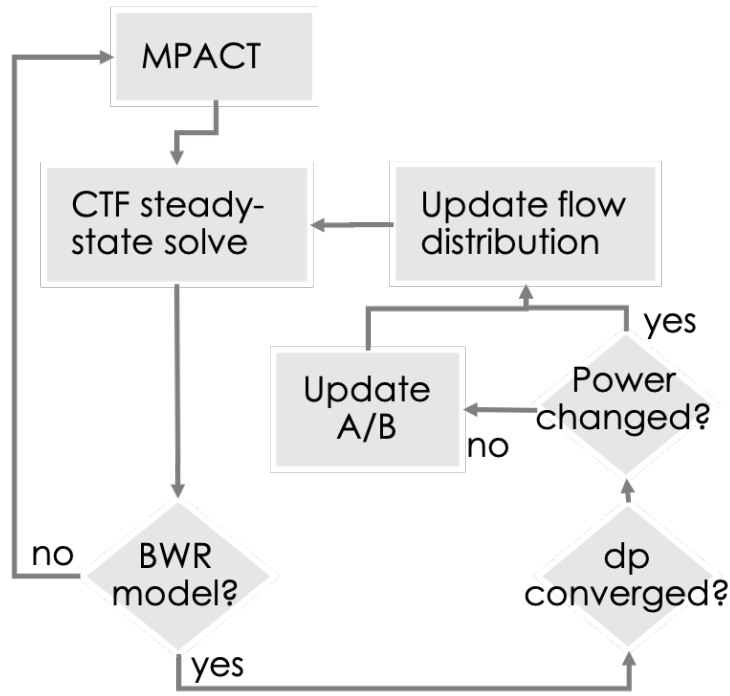


Figure 3. Flowchart of how pressure balance iteration loop is implemented in CTF for BWR models.

depletion steps. Resetting the distribution to uniform leads to extra pressure balance loop iterations being needed every CTF solve.

4. Force CTF to run for a minimum of 1.5 sec transient time when doing an iteration in the pressure balance loop. After implementing Fix 3, CTF could satisfy the steady-state stopping criteria very quickly during a pressure balance loop iteration, which lead to a bad prediction of the mass flow rate /pressure drop relationship. This can lead to oscillatory solutions. A minimum transient run of 1.5 sec was chosen because at nominal BWR operating conditions, this will allow the inlet mass flow rate change to propagate to the assembly outlet.

The new algorithm can be visualized with the flowchart shown in Figure 3.

The algorithm was improved by only updating the A and B coefficients when a flow change actually occurred. When a power change causes the ΔP change, the A/B values from the previous update are reused to predict the required flow change.

The second change was to add a limiter on the total allowable mass flow rate change. After \dot{m}_{k+1} is calculated, the relative change requested for each assembly is:

$$\Delta \dot{m} = \frac{\dot{m}_{k+1} - \dot{m}_k}{\dot{m}_k}. \quad (10)$$

The maximum $\Delta \dot{m}$ in the core is checked against the threshold change of 20 %. If it is greater than this threshold, then a scaling factor is calculated as follows:

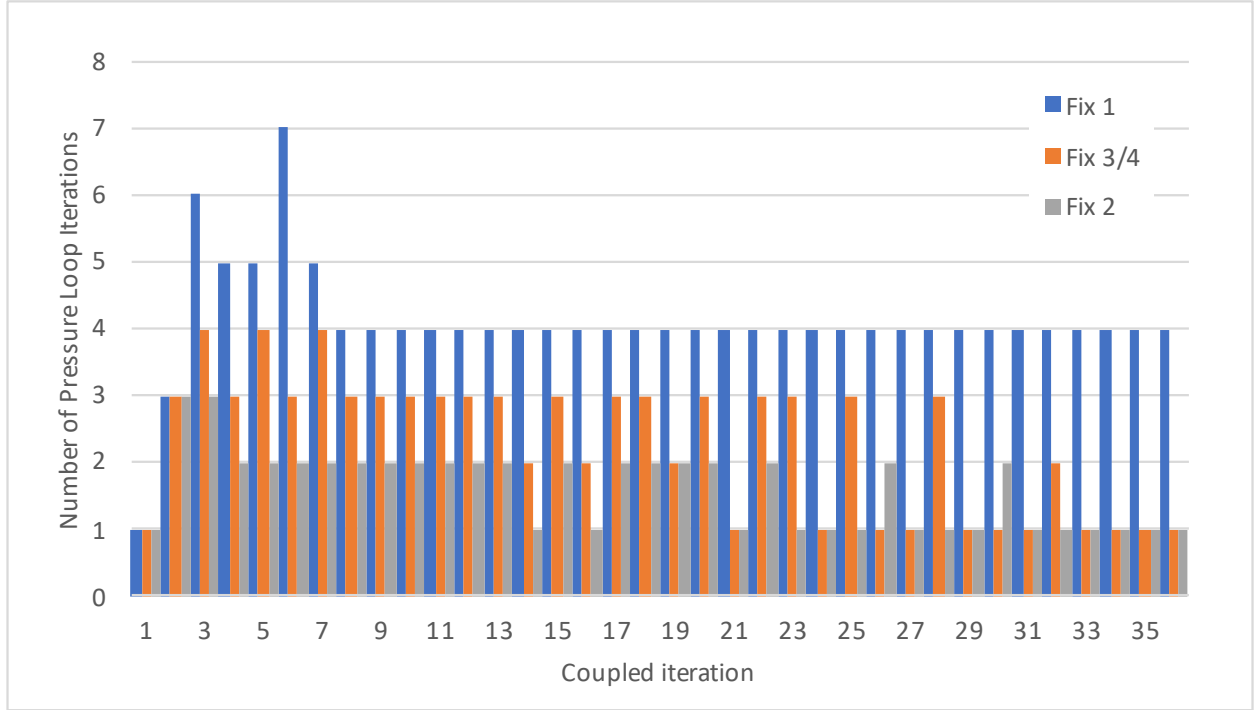


Figure 4. Improvement in the performance of the pressure balance loop after implementing improvements to the algorithm.

$$F = \frac{1}{\max(|\Delta \dot{m}|)/0.2}. \quad (11)$$

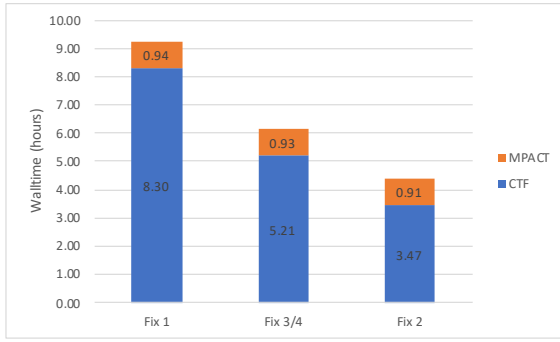
If the maximum $\Delta \dot{m}$ is less than the threshold, then F will be set to one. The following is the relationship to update the individual assembly mass flow rates:

$$\dot{m}_{k+1} = \dot{m}_k + F \Delta \dot{m}. \quad (12)$$

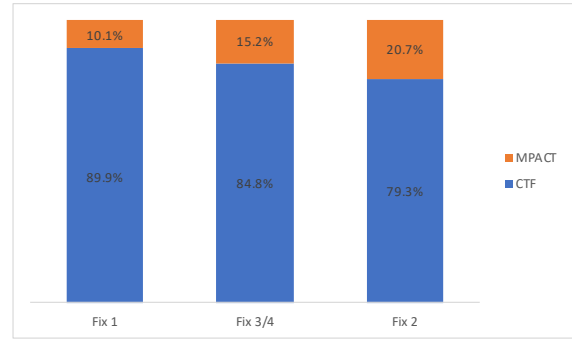
By using the same scaling factor for all assembly mass flow rate updates, the total core mass flow rate will be preserved. If the mass flow update was limited, then another pressure balance loop iteration will likely be required to meet the pressure balance tolerance.

Before implementing Fix 1, the coupled VERA simulation would crash during the Hatch model simulation. This fix allowed baseline results to be produced in which the VERA Hatch simulation ran in 9.24 hours on the INL Sawtooth HPC system. Fixes 3 and 4 were implemented together and led to a walltime of 6.1 hours. Fix 2 led to a further reduction in walltime to 4.4 hours. The primary source of performance gain is from the reduction in the number of CTF solves required. Figure 4 shows the number of pressure loop iterations taken for each coupled iteration being done in the Hatch solution.

Another way to look at the improvement in performance is by looking at the breakdown between the time spent in CTF vs. MPACT. Figure 5 shows how the walltime decreases with each improvement to the pressure balance loop. It also shows that CTF is dominating the walltime currently but that the pressure



(a) Absolute walltime spent in VERA components.



(b) Percentage of time spent in VERA components.

Figure 5. Total simulation time for the first statepoint of the full core Hatch model after implementing pressure balance loop improvements.

balance improvements have dropped the time spent in CTF from 90 to 80 %. After making the performance improvements, the total VERA walltime was decreased by a factor of 2.

3. GEOMETRY MODELING IMPROVEMENTS

3.1 Inlet Orifice Map

An inlet orifice map is used to specify loss coefficients at the inlet of each assembly in the model. Because the center of the core will be more voided than the periphery of the core, the pressure drop and flow resistance will be higher in the center. This will cause the water entering the core to prefer the periphery assemblies and will choke flow to the high-power regions. To counter this effect, the core plate can include assembly inlet orifices that are sized differently to better balance flow delivery.

These losses are assumed to be part of the core plate and are fixed in location. The losses will not move throughout a depletion or during a refuel, so these losses do not include effects of fuel assembly geometry. To calculate the impact of the loss on pressure loss, a form loss coefficient and an associated flow area are required. By considering these design requirements, a new set of cards were added to the CORE block of the VERAIn input file. An example of this input is shown as follows.

```
inlet_orifice_map
  B2 B2
  B2 B1 B1 B2
  B2 B1 B1 B2
  B2 B2

inlet_orifice_loss B1 0.5 1e-2
                  B2 1.0 1e-2
```

The `inlet_orifice_map` input is a 2D map that is the same shape as the `core_map`, with each entry (i.e., assembly) pointing to a labeled `inlet_orifice_loss`. The `inlet_orifice_loss` entry provides the

necessary form loss coefficient and flow area (in units of cm^2) of each inlet loss. Xml2ctf will read this information and pass it through to the CTF input file as a map of inlet form losses and orifice flow areas.

The most straightforward way to include the effect of the inlet orifices in the CTF solution was to add an extra loss term while performing the assembly pressure balancing done in the pressure balance outer iteration loop. In this stage, the total pressure drop of each assembly is calculated and used to predict the correct inlet flow distribution that leads to a balanced pressure drop in all assemblies. After summing the pressure drop within the assembly, the inlet form loss is added before performing the inlet flow distribution calculation. The inlet pressure drop is calculated as:

$$\Delta P = 1/2 k \dot{m}^2 / (\rho A^2). \quad (13)$$

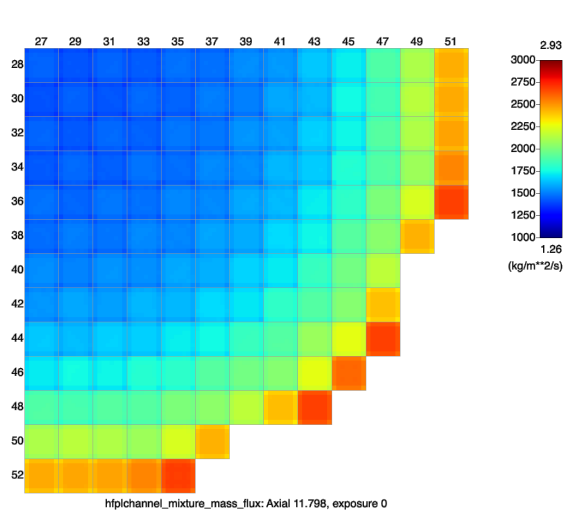
The k and A values are the user-supplied form loss coefficient and orifice flow area, respectively; \dot{m} is the CTF calculated inlet mass flow rate adjusted to balance pressure; and ρ is the inlet density.

After implementing this feature, it was tested for the Hatch core. The Hatch specification states that orifice diameters are 1.488 in. and 2.222 in. for outer and inner assembly types, respectively. The Borda-Carnot equation was used to estimate the form losses for the two orifice types:

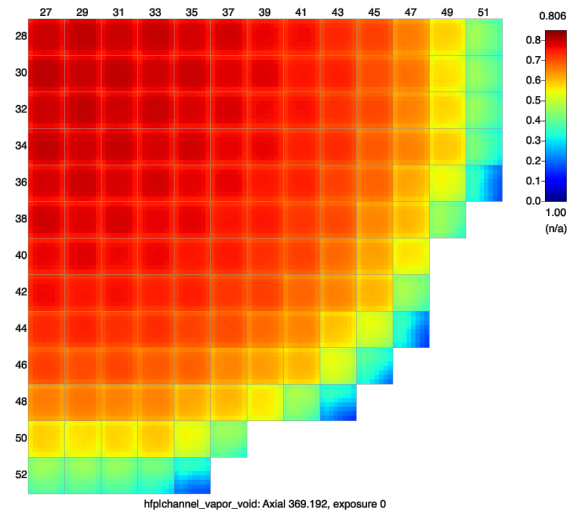
$$k = (1 - A_o/A_a). \quad (14)$$

In the equation, k is the form loss coefficient, A_o is the area of the orifice calculated from the hole diameter and assuming the orifice is a single round hole, and A_a is the assembly flow area. Loss coefficients were calculated to be 0.888 for the outer assembly and 0.75 for the inner assembly. The simulation was run in a coupled VERA simulation with and without inlet orifice loss coefficients applied. Figure 6 shows the inlet mass flux and outlet void distribution without the inlet orifice map applied. Figure 7 shows the same results when the inlet orifice map is applied.

Without the inlet orifice map in place, the assemblies on the periphery experience a low pressure drop, which leads to a high inlet flow. Because flow is high and power is low in the periphery assemblies, the void stays low in these assemblies. With the application of the inlet orifices, the pressure drop of the periphery assemblies is much higher, which leads to more flow going to the interior assemblies. As Figure 7 indicates, the outlet void distribution is more uniform with the inlet orifice losses applied.

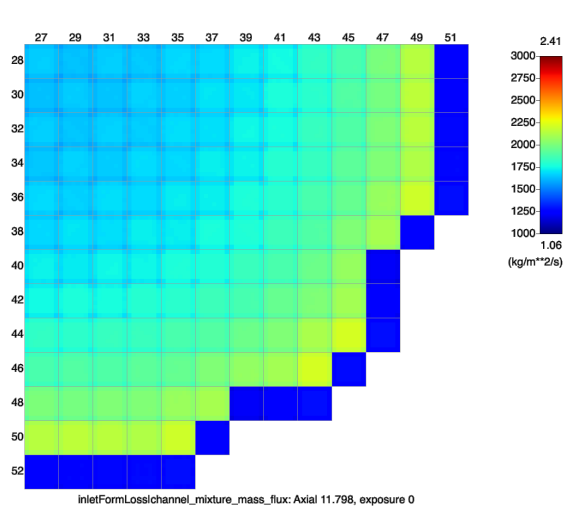


(a) Inlet mass flux distribution

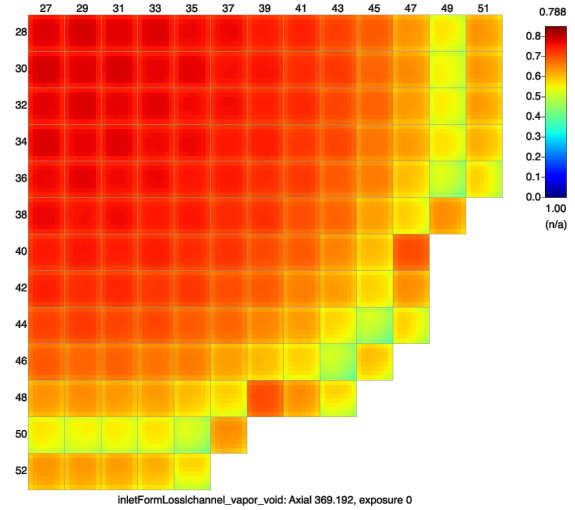


(b) Outlet void distribution

Figure 6. Hatch results with no inlet orifices.



(a) Inlet mass flux distribution



(b) Outlet void distribution.

Figure 7. Hatch results with inlet orifices applied.

3.2 Axially Varying Water Rod Geometry

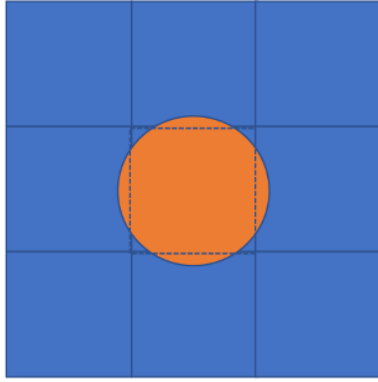
In BWR fuel assemblies, it is common for the large water rod size to change in the axial direction. The rod will start at a small diameter in the lower portion of the bundle, then increase in diameter for the majority of the assembly and reduce in size again at the top of the bundle. Xml2ctf did not support this geometry originally and instead tried to pick the longest axial lattice that contains fuel. If a complex arrangement of lattices is specified in the VERAIn file, the Xml2ctf might choose the wrong lattice, which can lead to Xml2ctf crashing or the wrong large water rod geometry being selected. At the least, Xml2ctf needed to be updated to account for this geometry, and a sensitivity study for assuming the large water rod has a constant diameter throughout the model should ideally be performed.

The axial variation in the water rod geometry can be achieved by using the axial geometry variation tables in CTF, which is the same feature that was used to model part length rods. However, there is an issue with this approach because when the water rod has a small diameter, it will actually be smaller than the channel it displaces due to the way the subchannel mesh is set up for large water rod models. An example of this issue is shown in Figure 8. Rather than connecting the pin centers, a rectangular meshing approach is used so that subchannel boundaries are always orthogonal to one another. Figure 8a shows that as the diameter of the large water rod is reduced, at some point the diameter is actually less than the diagonal of the center subchannel that is displaced by the presence of the rod.

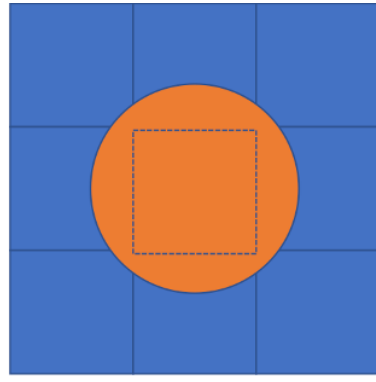
There are two ways to deal with the additional area in this phantom channel: (1) the area can be ignored if it is small without any significant impact on the solution or (2) it can be absorbed into the other three subchannels that are solved by CTF. Both approaches were tested for a rod bundle geometry depicted by Figure 9.

For the second approach, the area of the missing “notches” is added to the north/east/west/south subchannels. The notch between subchannels 39, 40, and 50 is divided equally between subchannels 40 and 50. The wetted perimeter in contact with this notch is also divided and added to the wetted perimeter of subchannels 40 and 50. In the second approach, the gap width between channels 39 and 40 or 39 and 50 is not modified to account for the additional distance between the southeast point of subchannel 39 and the large water rod surface. Figure 10 shows the liquid velocity profile for the two different approaches and assumes that the large water rod diameter is constant over the entire axial length of the bundle. In the figure, the assumption of constant diameter is labeled “uniform,” the first approach that involves ignoring the missed area is labeled “vary,” and the second approach in which the missed area is absorbed into adjacent subchannels is labeled “correct.” The axial void profile for this case is shown in Figure 11.

As shown in the results, there is no discernable difference as to whether the notches are ignored or absorbed into the adjacent subchannels. However, simply ignoring the reduced water rod area in the lower and upper regions can have a nonnegligible impact on the thermal hydraulic solution. Based on this finding, Xml2ctf was modified to account for the axially variation of large water rods by using axial variation tables. If the large water rod becomes smaller than the phantom subchannel it displaces, then the exposed areas are ignored due to negligible impact on the solution.



(a) Large water rod in lower/upper regions of bundle where its diameter is small compared with subchannel mesh.



(b) Large water rod in middle of bundle where its diameter is large compared with subchannel mesh.

Figure 8. Example of large water rod variation in CTF subchannel mesh.

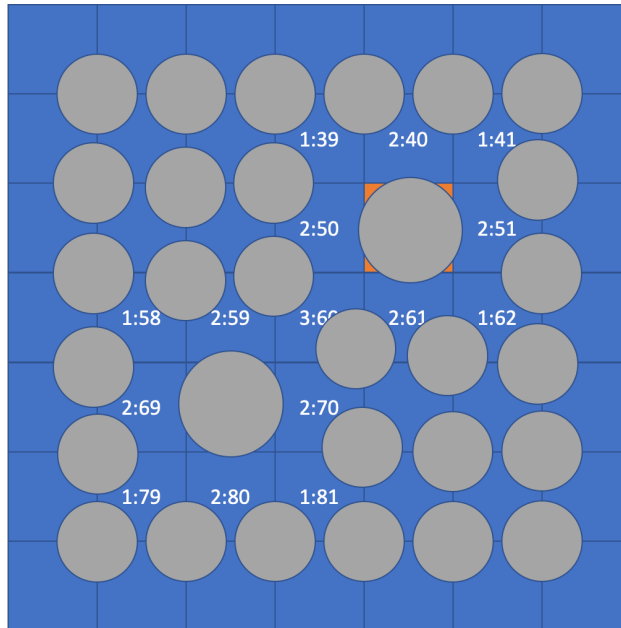


Figure 9. Schematic of rod-bundle region used to test axially varying water rod modeling approach.

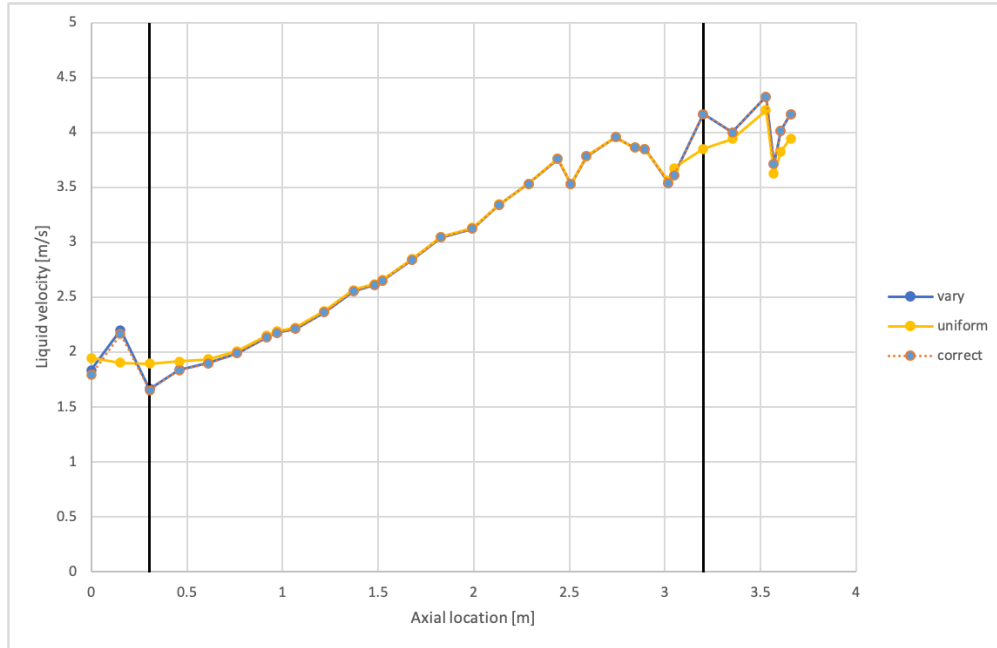


Figure 10. Axial liquid velocity profile in subchannel 40 of a BWR bundle model for three different approaches for modeling axial water rod diameter variation.

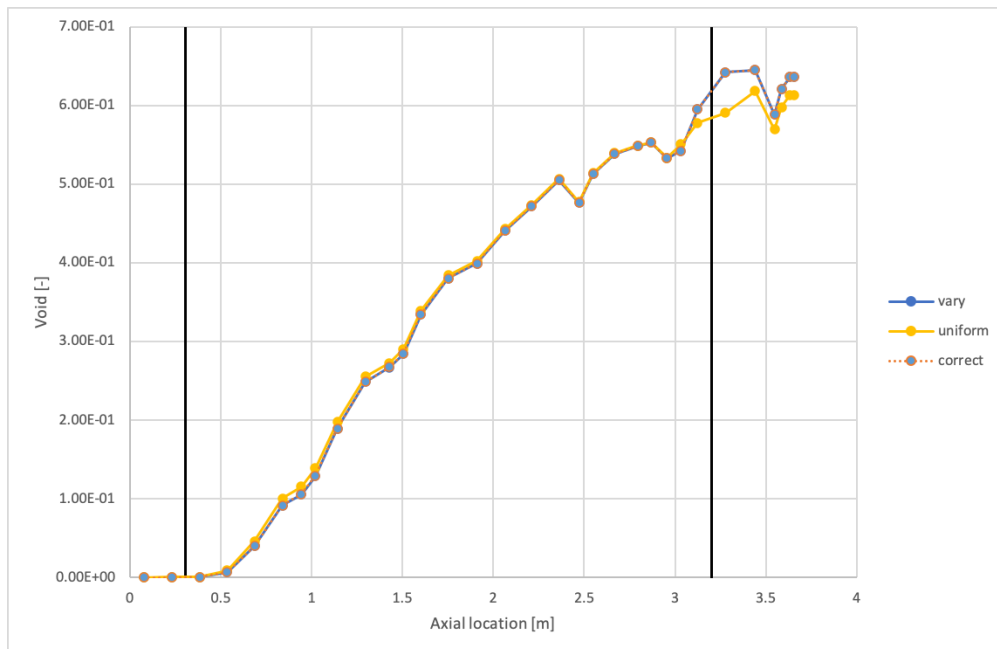


Figure 11. Axial vapor void profile in subchannel 40 of a BWR bundle model for three different approaches for modeling axial water rod diameter variation.

3.3 Large Radius Channel Box Corner Rounds

BWR channel boxes include rounded corners (when looked at from above), which must be correctly modeled by CTF and MPACT. The reduced area of the corner subchannels will impact void drift and moderation in the boundaries of the fuel assembly. The actual modeling of the channel box corner rounds is performed by the CTF preprocessor utility, Xml2ctf. When Xml2ctf was originally modified to support channel box corner rounds, a simplifying assumption was made that the radius of the corner round would not exceed the total width of the corner channel. With this assumption, the rounded corner would not affect the adjacent side-type subchannels that connect to the corner subchannel. However, although this assumption is valid for older fuel designs, it is common for newer designs to use a larger radius that will remove some flow area from adjacent side subchannels. Figure 12 illustrates how side subchannels will be impacted by larger radii.

The highlighted purple regions show the area removed from side subchannel types when the corner round radius is larger than the corner subchannel width. Not only does this impact the flow area and wetted perimeter of the side subchannels but the subchannel centroids, which are used to calculate gap lengths, and gap width are also impacted.

Xml2ctf uses a unit-tested set of classes to calculate the area, wetted perimeter, and centroid of various shapes and combinations of shapes. This set of classes is used to build the complex shape of actual subchannels to calculate the geometry that CTF needs. To add this feature, modifications must be made to the calculation of corner- and side subchannel-type geometries to account for the additional area removal. After adding the feature, a unit test was set up that creates a single BWR assembly. The flow area and centroid of both corner and side types are determined by defining subchannel boundaries as functions in a Cartesian plane, which can then be integrated appropriately to determine the correct values. These values are compared with each of the four and eight corner and side subchannels, respectively, that are affected by the large corner round radii to ensure that the calculations being performed by the shape classes are correct.

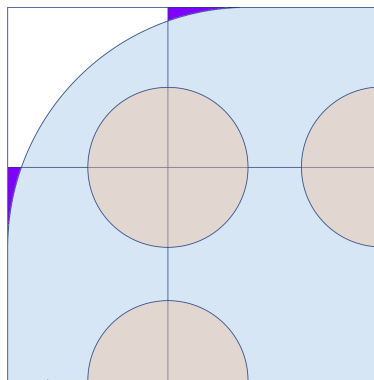


Figure 12. Illustration of the impact of a large channel box corner round radius on corner and side subchannel geometry.

3.4 Addition of Support for Quarter Symmetry Models

When Xml2ctf was initially developed to support BWR models, no support was added to model quarter symmetry. This capability was added as part of this work so that a full core model defined in the VERAIn input file could easily be modeled as quarter symmetry by simply changing the input symmetry flag.

4. SUMMARY

This report summarizes improvements in CTF related to the modeling of BWR-specific geometry and operating conditions. Several new features were added, including support for inlet orifice maps, large channel box corner rounds, axially varying large water rod diameter, and quarter symmetry modeling of BWR cores. In addition to this, improvements were made to the numerical solution approach to CTF, which included support for using a direct solver for the pressure matrix solution, the addition of a mass and energy conservation iteration loop, and improvement to the pressure balance outer iteration loop algorithm. The numerical improvements were needed to improve the stability of CTF (i.e., its ability to reach steady-state for two-phase flow) and reduce its walltime, which was observed to be a significant portion of the VERA total walltime for core-scale BWR problems.

The pressure balance iteration loop improvements have led to a 2× reduction in the total VERA walltime for the Hatch model that was used for testing and dropped the time spent in CTF from 90 to 80 %. Despite these significant improvements, there is still more work to be done in the second year of the project. In the current approach, CTF is only using a single processor per assembly. In PWR models, the default is to use nine processors per assembly. Since PWR assemblies are about four times larger, this means that the BWR models are using roughly two times fewer processors per assembly. Based on this, increasing the processors used in the parallelization of CTF will likely lead to substantial further performance improvements. Additionally, the current pressure balance algorithm only considers the impact of bundle flow on bundle pressure drop, but there will also be an impact caused by bundle power changing during the coupled simulation. The most current solution shows that CTF is still taking two pressure loop iterations for many of the coupled iterations. Including the power change in the relationship might help further reduce the number of iterations needed in CTF. Finally, the solution of the momentum equations is being reviewed to determine whether the solution can be made more implicit, which would allow for larger time steps to be taken and further improve the stability of the solution.

In addition to further performance improvements, the assessment of the two-phase pressure drop experienced due to spacer grids must be reviewed and improved because it has been observed that CTF predicts a significant void dip in the vicinity of the spacer grid. Further support will also likely be needed for previously unmodeled cores, such as mixed fuel designs, and bypass work being performed by NCSU must be integrated into VERA. Finally, work will be done to expand the CTF modeling region into the upper plenum, which will allow for the better determination of outlet flow quality and distribution that is delivered to steam separators.

5. REFERENCES

- [1] A. Abarca, D. Ziyad, and M. Avramova. Initial bwr bypass-flow modeling and assessment report for ctf. Technical Report M2EX-19OR04010118, North Carolina State University, 2020.
- [2] B. Crawford and R. Whitesel. Core design and operating data for cycle 1 of hatch 1. Technical Report EPRI NP-562, Electric Power Research Institute, 1979.
- [3] R. Salko and V. Kumar. Assessment of ctf needs for modeling of boiling water reactors. Technical Report ORNL/TM-2020-3, Oak Ridge National Laboratory, 2020.
- [4] R. Salko, A. Wysocki, B. Collins, A. Godfrey, C. Gosdin, and M. Avramova. Development and Testing of CTF to Support Modeling of BWR Operating Conditions. Technical Report CASL-U-2016-1030-000, The Reactor Dynamics Fuel Management Group at the Pennsylvania State University, 2016.
- [5] R. Salko, V. Kumar, and B. Hizoum. Improvements to ctf closure models for modeling of two-phase flow. Technical Report ORNL/TM-2020/3, Oak Ridge National Laboratory, 2020.