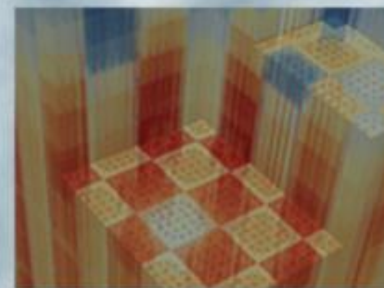
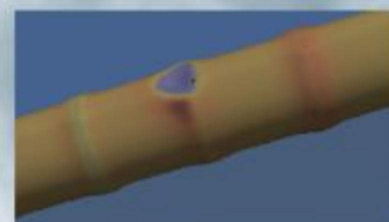
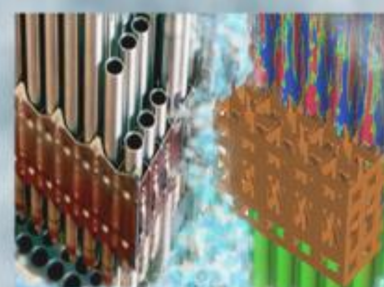
CASL-U-2017-1311-000







Excore Radiation Transport Modeling with VERA

Tara Pandya, ORNL
Thomas Evans, ORNL
Kevin Clarno, ORNL
Ben Collins, ORNL

April 28, 2017

Approved for Public Release

REVISION LOG

Revision	Date	Affected Pages	Revision Description
0	5/18/2017	All	Initial Public Release - Unlimited

Document pages that are:

Export Controlled ____________________None___________________________________

IP/Proprietary/NDA Controlled_______None____________________________________

Sensitive Controlled__________________None___________________________________

Requested Distribution:

To:

Copy:

EXECUTIVE SUMMARY

It is important to be able to accurately predict the neutron flux outside the immediate reactor core for a variety of safety and material analyses. Monte Carlo radiation transport calculations are required to produce the high fidelity excore responses. Under this milestone VERA (specifically the VERAShift package) has been extended to perform excore calculations by running radiation transport calculations with Shift. This package couples VERA-CS with Shift to perform excore tallies for multiple state points concurrently, with each component capable of parallel execution on independent domains. Specifically, this package performs fluence calculations in the core barrel and vessel, or, performs the requested tallies in any user-defined excore regions.

VERAShift takes advantage of the general geometry package in Shift. This gives VERAShift the flexibility to explicitly model features outside the core barrel, including detailed vessel models, detectors, and power plant details. A very limited set of experimental and numerical benchmarks is available for excore simulation comparison. The Consortium for the Advanced Simulation of Light Water Reactors (CASL) has developed a set of excore benchmark problems to include as part of the VERA-CS verification and validation (V&V) problems. The excore capability in VERAShift has been tested on small representative assembly problems, multiassembly problems, and quarter-core problems. VERAView has also been extended to visualize these vessel fluence results from VERAShift.

Preliminary vessel fluence results for quarter-core multistate calculations look very promising. Further development is needed to determine the details relevant to excore simulations. Validation of VERA for fluence and excore detectors still needs to be performed against experimental and numerical results.

MILESTONE CONTRIBUTION TEAM

Shift Team

Tom Evans	ORNL
Tara Pandya	ORNL
Seth Johnson	ORNL
Greg Davidson	ORNL

MPACT Team

Ben Collins	ORNL
Shane Stimpson	ORNL

PHI Team

Kevin Clarno	ORNL
Ron Lee	ORNL
Stuart Slattery	ORNL

CONTENTS

FIGURES

TABLES

LISTINGS

ALGORITHMS

ACRONYMS

AMA	Advanced Modeling and Applications
API	application programming interface
BOC	beginning of cycle
BWR	boiling water reactor
CASL	Consortium for Advanced Simulation of Light Water Reactors
CE	continuous energy
CTF	COBRA-TF
DTK	Data Transfer Kit
EOC	end of cycle
EFPD	effective full power days
FOM	figure of merit
FW-CADIS	Forward-Weighted Consistent Adjoint Driven Importance Sampling
GG	Exnihilo General Geometry
HDF5	Hierarchical Data Format 5
HPC	high performance computing
LWR	light water reactor
MC	Monte Carlo
MG	multigroup
MOC	middle of cycle
MSOD	Multiple-set, overlapping domain
NUMA	nonuniform memory access
OLCF	Oak Ridge Leadership Computing Facility
ORNL	Oak Ridge National Laboratory
PHI	physics integration
PWR	pressurized water reactor
RTK	Reactor ToolKit
SCALE	Standardized Computer Analyses for Licensing Evaluation
VERA	Virtual Environment for Reactor Applications
VERA-CS	Virtual Environment for Reactor Applications Core Simulator
V&V	verification and validation

1. INTRODUCTION

This report describes the coupling of VERA-CS [1] to Shift [2]in the VERAShift package, with an emphasis on excore capabilties. The objective is to provide an automated mechanism through which independent Shift fixed-source MC calculations are launched at user-specified state points during a standard VERA-CS calculation. For default vessel tally fluence calculations, very little user intervention is required to produce the vessel tally output. The capability to insert the detailed core model is also included, with all associated isotopics and temperatures, at each state point within a broader, user-defined model. For example, the detailed core model can be inserted within a containment geometry that includes external detector locations and other features.

Frequently requested analyses by reactor physicists are the ability to perform excore calculations such as vessel fluence and detector response. These calculations are needed for a variety of analyses including, safety, lifetime extension, and material durability and performance. Unfortunately the publicly available experimental results available for use as benchmarks for excore simulations are limited. MC radiation transport calculations are required to produce high fidelity excore responses. This milestone is defined to develop this capability in the VERAShift package. The VERAShift package couples VERA-CS with Shift. The initial VERAShift implementation supported running eigenvalue calculations under the L3:RTM.MCH.P13.03 Consortium for Advanced Simulation of Light Water Reactors (CASL) [3] milestone. In this work, VERAShift has been extended to perform forward calculations with excore tallies at multiple state points.

The basic coupling strategy involves transferring the detailed isotopics, temperatures, and fission source generated by VERA-CS to Shift. Shift uses this information to automatically generate a detailed geometric core model for that state point. Shift samples the fission source provided by VERA-CS and runs a fixed-source calculation to determine the fluence in the vessel and excore features. The fixed-source transport can be run in n- or n-γ mode. For coupled (n, γ) problems, the secondary gamma source is sampled from neutron interactions within the core elements during neutron transport. Shift does not currently support prompt fission gammas and delayed fission product gammas as a source in a forward calculation. Also, the current VERAShift implementation does not have the capability to sample from a user input gamma source.

The following sections present the details of the work performed to enable excore modeling in VERAShift. The capabilities of the VERAShift package are reviewed and summarized in § 2. Section 3 explains the methods and code-coupling algorithms used in VERAShift for performing excore calculations. Section 4 gives details on adding and integrating the excore geometry with VERA-CS. Section 5 shows preliminary results using VERAShift, including vessel fluence. The work performed for this milestone is summarized in § 6. The report concludes with a discussion of continuing and future work for excore calculations with VERAShift in § 7.

2. VERASHIFT PACKAGE OVERVIEW

The purpose of VERAShift is to allow VERA users to set up and execute independent MC calculations at user-defined state points to *validate* and *extend* the modeling capabilties in VERA-CS. Validation calculations typically consist of executing MC k-eigenvalue calculations at the current reactor state. The MC transport package used in VERA-CS is provided by Shift. Validation of Shift for CASL use is described in Pandya 2016b [4]. For this work, VERAShift was extended to enable fixed-source calculations to calculate vessel fluence and perform other relevant excore analyses.

Shift provides novel transport algorithms and hybrid methods tailored for leadership-class computing platforms, in addition to multiple geometries, tallies, and physics [2, 5]. However, the principal advantage provided by VERAShift is the ability to automatically set up a three-dimensional full-core model specified by the VERA input specification. Thus, almost no additional user-intervention is required to execute a Shift calculation through VERAShift, as demonstrated in § 3.2. This capability is essential because it removes any sources of error that often result from inconsistencies between the input models required by different codes.

Accordingly, VERAShift can operate in two problem modes: eigenvalue and forward (fixed-source). In both modes, data are transferred from VERA-CS that describe the isotopic state of all materials in the core, the temperatures of fuel, clad, and moderator, and the neutron fission source. In the current implementation, the core geometry out to the vessel is defined by the VERA input, as described in § 4.

There are eight main areas to consider when discussing the capabilities of MC codes:

1. energy treatment,
2. geometric representation,
3. particle types,
4. material isotopic treatment,
5. temperature treatment,
6. types of parallelism,
7. types of tallies, and
8. types of sources.

The energy treatment used by VERAShift is continuous-energy (CE) physics for both problem modes. In eigenvalue mode, VERAShift uses the RTK geometry with a simple vessel model; in forward mode VERAShift uses the GG package with more detailed modeling capability. Only neutrons are transported for eigenvalue calculations whereas VERAShift can follow neutrons or coupled neutron-photon transport for forward calculations. For both modes, VERAShift uses the depleted isotopics in the fuel pins. Also, VERAShift currently replicates the problem domain on all processors requested. Different quantities of interest are tallied in VERAShift based upon the problem mode. For eigenvalue calculations, the energy-integrated fission rate on a rectilinear mesh covering the pincells is calculated. For forward calculations, VERAShift calculates the energy-binned flux in a cylindrical mesh covering the barrel and vessel.

Finally, in a fixed-source problem, the VERA-CS fission source is used as the neutron source for n or n-γ transport. In eigenvalue problems, the VERA-CS fission source is used as the initial source (eigenvector) for power iteration. This technique can significantly reduce the number of inactive iterations required for convergence, as shown in Biondo et al. 2017 [6]. The current capabilities of VERAShift are summarized in Table 1.

Shift has additional capabilities not currently interfaced to VERA through VERAShift (e.g., multigroup energy treatment); a complete detailing of Shift's capabilities is available in Pandya et al. 2016a [2] and Johnson et al. 2017 [7]. Additionally, many modeling features are treated identically in both modes. For example, the temperatures are currently updated in all fuel pins for both eigenvalue and forward problem modes. However, in the future an option will be provided to homogenize temperatures in the inner part of the core for fixed-source problems to reduce the memory footprint in full core, three-dimensional problems.

Table 1. Capabilities of VERAShift.

	Problem mode	
Capability	Eigenvalue	Forward
Energy	CE	CE
Geometry[a]	Reactor Toolkit (RTK); only simple vessel model supported	Exnihilo General Geometry (GG); core automated through VERA out to vessel, translation and additional geometric features enabled through supplemental input
Particles	n	n or n-γ
Isotopics	depleted isotopics in fuel via VERA-CS; inserts, boron concentration, and control rods not yet supported	depleted isotopics in fuel via VERA-CS; inserts, boron concentration, and control rods not yet supported
Temperature	fuel, cladding, and moderator temperatures via VERA-CS	fuel, cladding, and moderator temperatures via VERA-CS
Parallelism	domain replicated	domain replicated
Tallies	rectilinear mesh (pincell) fission, energy-integrated	cylindrical mesh (vessel) flux with energy bins (through VERA input); user-defined tally definitions enabled through supplemental input (see [7])
Source	initial fission source via VERA-CS	spatial fission source via VERA-CS; energy is sampled from a ^{235}U Watt spectrum

[a]See § 4 for detailed geometry descriptions.

3. METHODOLOGY

The VERAShift package was developed under a previous CASL milestone (L3:RTM.MCH.P13.03) to validate VERA-CS calculations. This development included adding the ability to validate eigenvalues and pin powers for multistate calculations. Under this milestone, the same framework previously established was developed further to allow for excore calculations; this entails running Shift in the forward calculation mode. This section presents details on how the VERAShift excore calculations are performed, including communication, an explanation of what is transferred between VERA-CS and Shift, inputs, running, and details on what the simulation produces.

3.1. COUPLING ALGORITHM

The basic algorithm for solving forward calculations with VERAShift is shown in Alg. 1. First, a global communicator is established between VERA-CS and Shift. Shift and VERA-CS run on their own sets of processors. Next, the VERA-CS calculation is set up, along with the maps needed for the data transferral between VERA-CS and VERAShift. These maps use the Data Transfer Kit (DTK) [8] to handle the communication and data structures passed between VERA-CS and Shift. Next, VERA-CS launches a state point calculation. During this calculation, VERA-CS runs MPACT for the deterministic eigenvalue calculation, and then it runs CTF for the thermal hydraulics calculation.

Once a state point is complete in VERA-CS, the Shift forward calculation is launched. VERA-CS and Shift operate on independent domains. Each component has a unique message passing interface (MPI) communicator. Thus, VERA-CS and Shift can execute different state points concurrently. Shift must wait on VERA-CS to finish an eigenvalue calculation for a state point before it can launch the forward calculation of that state point. There is no feedback into VERA-CS from the Shift calculation.

The Shift calculation of a state point during the VERAShift solve is an MC forward calculation. Shift uses the fission source transferred from MPACT as the source for the fixed source solve. It also uses the short isotopics list from MPACT for the pins and inserts of interest. The short isotopics list is given in § 3.1.2. VERAShift does not currently use the full isotopics list due to the large memory requirement of tracking all of these nuclides. Shift then runs the requested number of particle histories for the given state point and records the vessel flux tally (or excore detector tally) in the HDF5 output file. The full cycle simulation is complete once MPACT completes the last statepoint and Shift completes the last requested forward calculation.

3.1.1 Code implementation

VERA-CS and Shift run different state points concurrently. A detailed diagram of the VERAShift driver is shown in Fig. 1. The driver first builds the `Multiphysics Distributor`. This class builds the communicators for VERA-CS and Shift and determines the domains on which each is defined. The driver then builds the VERA-CS and Shift evaluators denoted by `VeraCS_InlinePike` and `ShiftModelEvaluator`. The VERAShift driver controls when the transfer of data occurs and when the solvers in the evaluators are called. The solution procedure continues until the stop criteria (denoted by *wasLastState*) is communicated to the Shift evaluator. The last step is the driver calls the finalize methods in each evaluator to perform cleanup operations.

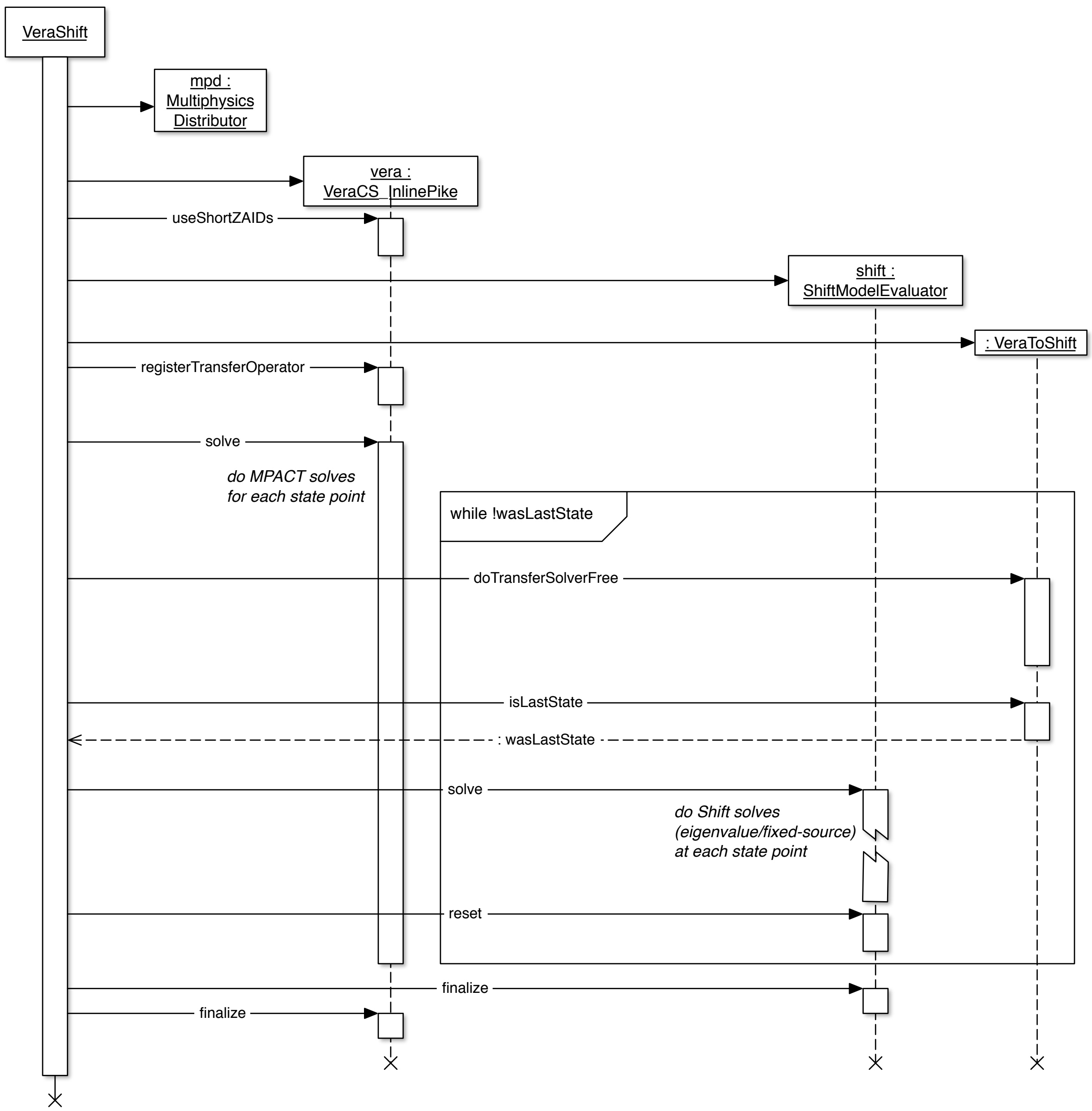


Figure 1. Sequence diagram showing VERAShift driver implementation.

Algorithm 1 Simplified VERAShift simulation flow.

```
Parse VERA-CS input
Set up MPI communicators
Construct VERA-CS and Shift model evaluators
Set up MPACT to Shift and CTF to Shift transfers
for each state do
    if processor running VERA-CS then
        Run a coupled VERA-CS single statepoint calculation or a predictor-corrector depletion calculation
        Send isotopics, temperatures, densities, fission source, and stop criteria
        Continue to next state
    else if processor running Shift then
        Receive isotopics, temperatures, densities, fission source, and stop criteria
        while !stop criteria do
            Run Shift forward calculation and tally flux in vessel
        end while
    end if
end for
```

The Shift solution procedure, executed by the `ShiftModelEvaluator::solve()` function, is illustrated in Fig. 2. The `ShiftModelEvaluator` is a class inside of VERAShift, and it provides the interface into Insilico's Shift driver implementation. The data transferred from MPACT are used to update the Shift model each cycle, and these operations are performed inside the `DTK_Adapter` class. As shown in Fig. 2, setup of the Shift manager and `DTK_Adapter` is done before the solve method is called. The solve method first updates the compositions by using the isotopics from the `DTK_Adapter`. Then the fission source is built using the `DTK_Adapter`. Finally, the solve method calls the problem function. This functions calls the Shift manager in Insilico to initialize, solve, and output the MC calculation for a state point.

The transfer operations between VERA-CS and Shift are performed using DTK. The `VeraToShift` class implements the DTK transfer operations, and the sequence of operations is illustrated in Fig. 3. This figure only explicitly shows the transfer with MPACT, but the transfer of temperatures and densities with CTF is similar. The data transfer fields and DTK field evaluators are accessed through the VERA-CS and Shift model evaluators. These field evalutators are passed the DTK volume source maps that perform the transfer operations. The sequence diagrams in Figs. 1-3 show the actual implementation of Alg. 1.

3.1.2 Transfer Data

As mentioned above, the fission source and isotopics for each state point are transferred between MPACT and Shift. VERAShift also couples temperatures and densities with CTF for each state point. The quantities specified below can be transferred between VERA-CS and Shift:

- in-core fission source from MPACT eigenvalue calculation;
- isotopics for pincells and inserts using the short list of tracked nuclides: ^{234}U, ^{235}U, ^{236}U,

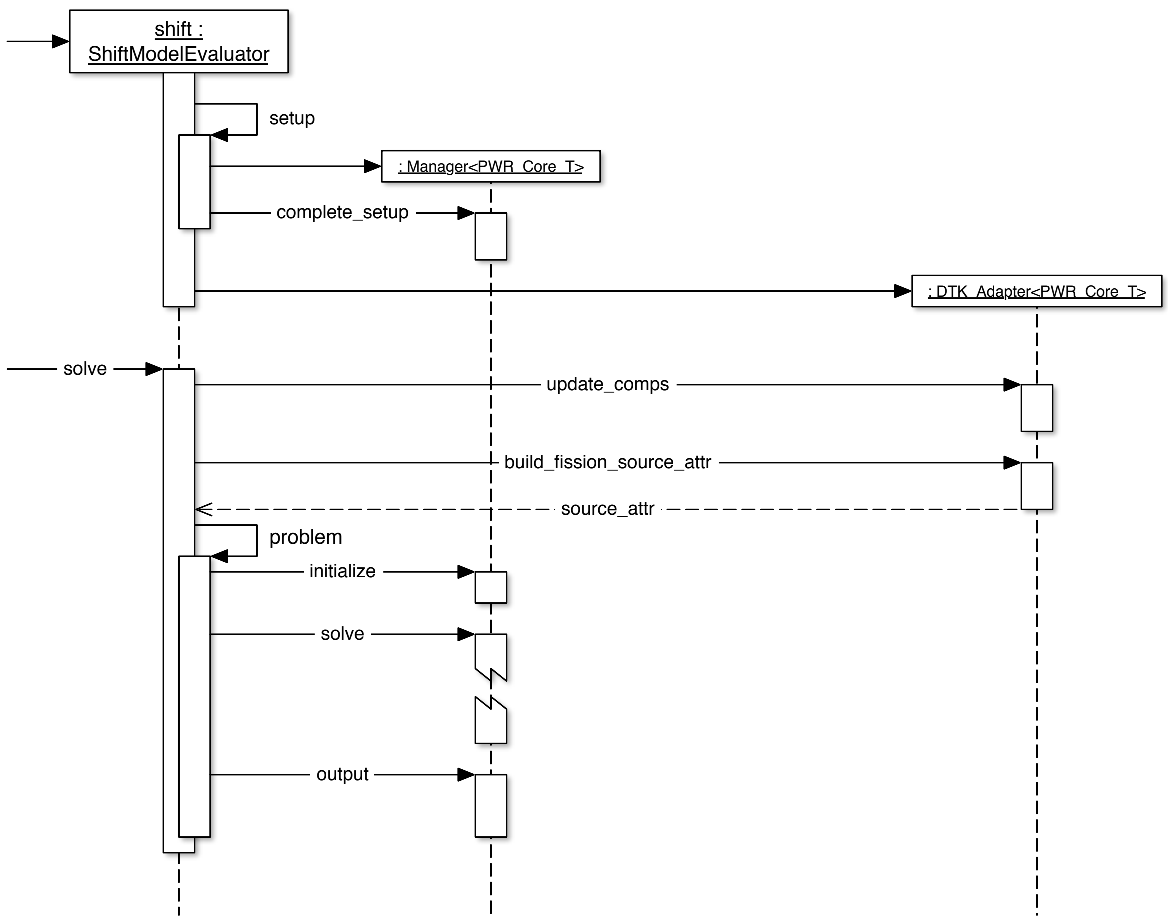


Figure 2. Sequence diagram showing the solution procedure in the `ShiftModelEvaluator`.

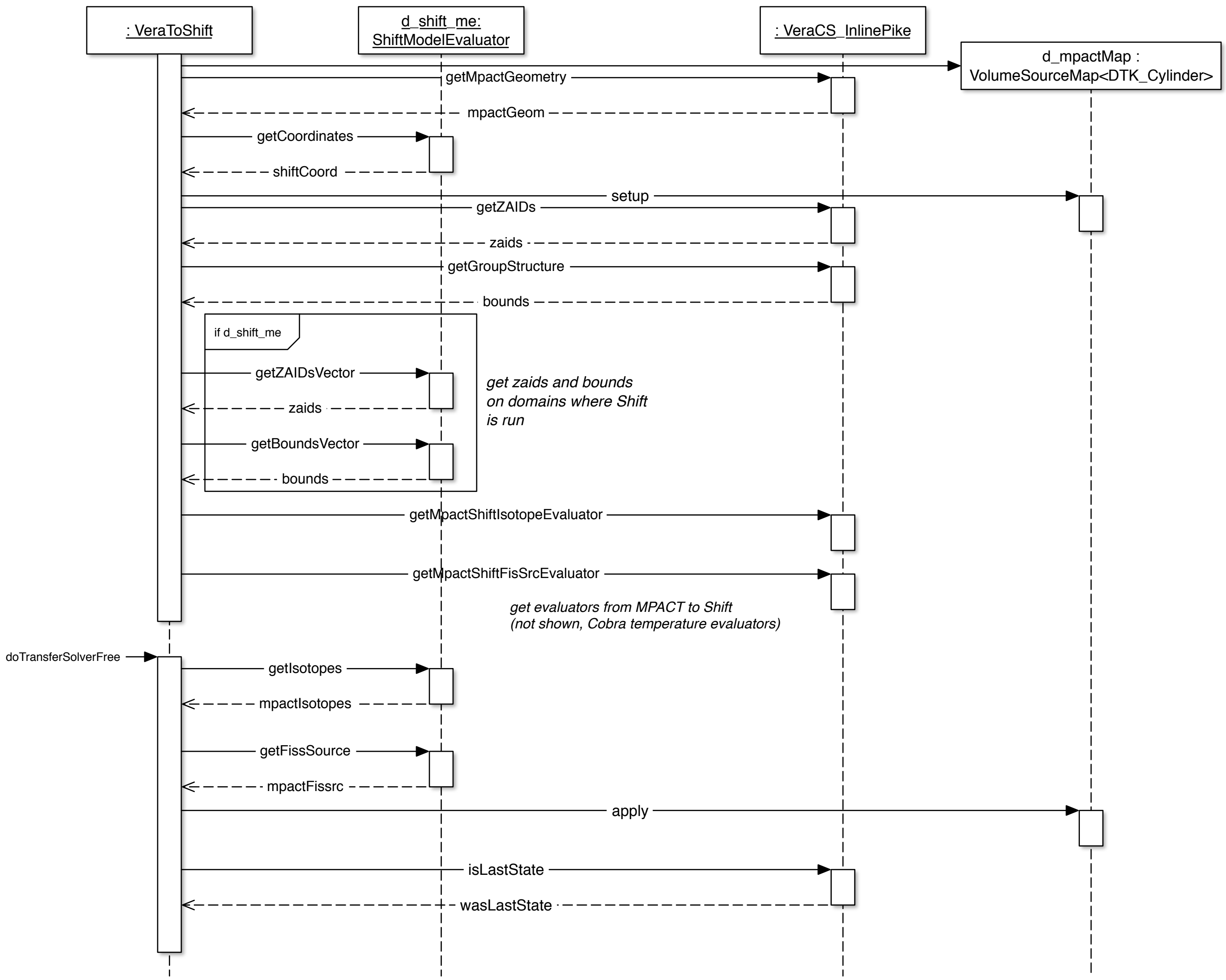


Figure 3. Sequence diagram showing the use of DTK to perform data transfer operations between VERA and Shift.

^{238}U, ^{16}O, ^{135}Xe, and ^{10}B;

- temperatures in fuel, clad, coolant of pincells and inserts; and
- density of coolant in pincells and inserts.

Data are transferred at pincell-centered points. Shift determines the cell-centroids by interrogating an internal reactor metadata model (see § 4) and calculating the center point in each fuel pincell. Inserts and control rods are not currently treated, but the mechanisms for handling this transfer are implemented in the code. This list of points is registered in DTK, which queries MPACT through VERA-CS.

On the MPACT side, the isotopics in the fuel pin identified at each point are *homogenized* over the containing fuel pin volume, and a single set of zaids is registered in DTK for transfer back to Shift because the same set of zaids is tracked in each fuel pin. The fission source and temperature transfers are handled in an analagous manner. DTK handles all the VERA-CS-to-Shift domain transfer mechanics.

For the fission source transfer, VERAShift transfers the spatial and energy information from MPACT. However, a ^{235}U Watt energy spectrum is currently used for the initial fission source for eigenvalue and forward calculations in Shift. The full energy distribution of the fission source from MPACT is not used yet because it is memory intensive and may not be necessary. Previous research has shown that the assumption of a Watt energy spectrum is reasonable for eigenvalue calculations [6]. For forward calculations, Shift can sample nonseparable sources. VERAShift could take advantage of the energy information transferred by setting up a nonseparable source, but it remains to be determined if this is advantageous or needed for excore calculations.

3.2. INPUT

For VERAShift forward calculations, the Shift parameters are defined in the common VERA input [9] in the SHIFT block. All of the usual inputs for the STATE, CORE, ASSEMBLY, INSERT, CONTROL, DETECTOR, EDITS, MPACT [10], and COBRATF blocks apply when running VERAShift. Table 2 gives the parameters regulating the forward calculations in Shift that can be used in the SHIFT block. An example of the Shift block in the VERA input with default vessel tally is shown in Listing 1.

Table 2. SHIFT block inputs through VERA.

Parameter	Description	Default	Type
celib_file	name of SCALE CE data file	ce_v7.1_endf.xml	string
create_unique_pins	make all pincells unique compositions	true	bool
do_transport	perform MC transport	true	bool
global_log	level of log information	info	{debug, diagnostic, status, info, warning, error, critical}
mode	type of particles to transport	n (eig), np (forward)	{n, np}
Np	number of particles to transport	1000	double
neutron_bounds	neutron energy bounds for tallies	none	array of decreasing doubles
num_axial	number of axial levels for tallies	1	integer
num_theta	number of theta divisions for tallies in [0, 2*pi*]	1	integer
output_fission_source	output the initial fission source for each state	false	bool
photon_bounds	photon energy bounds for tallies	none	array of decreasing doubles
problem_mode	run mode	eigenvalue	{eigenvalue, forward}
radial_mesh	radii for tally	vessel radii	array of doubles
temp_correction	type of doppler broadening for temperature	all	{none, oned, all}
temp_transfer	which temperatures to couple with CTF	all	{none, pin, all}
thermal_energy_cutoff	cutoff for treatment of thermal neutrons	10.0 eV	double
transport	type of physics	ce	{ce, mg}

Listing 1. Example SHIFT block with vessel tally in VERA common input.

```
[SHIFT]
  problem_mode            forward
  Np                      1000000
  output_fission_source   true
  transport               ce
  create_unique_pins      true

! OPTIONAL VESSEL TALLY PARAMETERS
  num_axial       10
  num_theta       8
  radial_mesh     65 68 70 72 75
  neutron_bounds  1000000 10000 1000 1000 1 0.1 0.01
```

3.3. RUNNING

Running the VERAShift executable is similar to running MPACT. The name of the executable is *vera_to_shift*. As mentioned earlier in this section, VERA-CS and Shift run on their own sets of processors, so at least 2 processors are always required. An example run command using a VERA input file called *sample.xml* is:

```
mpirun -np 16 ./vera_to_shift --case=sample
```

This command will run *vera_to_shift* on 16 processors using the given input file. The determinitation of the number of processors used for Shift is the leftover number of processors after the requested number for VERA-CS. The MPACT VERA user input manual provides details on the VERA-CS parallel decomposition [10]. A few important notes when running VERAShift are provided below.

- Unique pins are always turned on, so be aware of memory usage when running full-core calculations.
- To use temperature interpolation in SCALE, the XML form of the data must be used, because the compressed HDF5 data available with Shift are loaded through a different mechanism that does not currently support this feature. Note that the HDF5 form of the data is not currently available in VERAData, but it will soon be included.

3.4. OUTPUT

During the excore calculation, VERAShift writes the requested tallies to an HDF5 output file in output version 1 for each state point. Upon completion of the excore calculation, the user can postprocess and plot the tallies over all states. An example listing of the vessel tally output for two state points, generated via `h5dump -n`, is shown in Listing 2. The details of each tally data field in the HDF5 file `vessel_tally` group are described in Table 3.

This vessel tally can be plotted in VERAView [11] using the VERA-CS HDF5 output file and the Shift HDF5 output file. VERAView can currently produce 2D slices of the power distribution

Listing 2. Data in VERAShift HDF5 output file.

```
group      /STATE_0001
group      /STATE_0001/vessel_tally
dataset    /STATE_0001/vessel_tally/binned
dataset    /STATE_0001/vessel_tally/description
dataset    /STATE_0001/vessel_tally/group_bounds_n
dataset    /STATE_0001/vessel_tally/max_encountered_bins
dataset    /STATE_0001/vessel_tally/mesh_r
dataset    /STATE_0001/vessel_tally/mesh_stat
dataset    /STATE_0001/vessel_tally/mesh_theta
dataset    /STATE_0001/vessel_tally/mesh_z
dataset    /STATE_0001/vessel_tally/multiplier_descs
dataset    /STATE_0001/vessel_tally/multiplier_names
dataset    /STATE_0001/vessel_tally/normalization
dataset    /STATE_0001/vessel_tally/num_histories
dataset    /STATE_0001/vessel_tally/total
dataset    /STATE_0001/vessel_tally/translation
dataset    /STATE_0001/vessel_tally/volumes
group      /STATE_0002
group      /STATE_0002/vessel_tally
dataset    /STATE_0002/vessel_tally/binned
dataset    /STATE_0002/vessel_tally/description
dataset    /STATE_0002/vessel_tally/group_bounds_n
dataset    /STATE_0002/vessel_tally/max_encountered_bins
dataset    /STATE_0002/vessel_tally/mesh_r
dataset    /STATE_0002/vessel_tally/mesh_stat
dataset    /STATE_0002/vessel_tally/mesh_theta
dataset    /STATE_0002/vessel_tally/mesh_z
dataset    /STATE_0002/vessel_tally/multiplier_descs
dataset    /STATE_0002/vessel_tally/multiplier_names
dataset    /STATE_0002/vessel_tally/normalization
dataset    /STATE_0002/vessel_tally/num_histories
dataset    /STATE_0002/vessel_tally/total
dataset    /STATE_0002/vessel_tally/translation
dataset    /STATE_0002/vessel_tally/volumes
```

Table 3. Vessel tally data fields in the HDF5 output file. All of the data fields are stored in the `/STATE_nnnn/vessel_tally` group.

Data name	Type	Description
binned	array of doubles	energy-binned vessel flux mean and variance [g][z][theta][r][multiplier][stat]
description	string	description of tally
group_bounds_n	array of doubles	energy bin boundaries in decreasing order
max_encountered_bins	int	max number of tally bins encountered by one particle history
mesh_r	array of doubles	tally mesh radii
mesh_stat	array of strings	*mean* and *var*
mesh_theta	array of doubles	tally mesh azimuthal angles around full core
mesh_z	array of doubles	tally mesh axial edges
multiplier_descs	array of strings	description of flux
multiplier_names	array of strings	*flux*
normalization	double	normalization factor
num_histories	double	number of particle histories tallied
total	array of doubles	total vessel flux mean and variance [z][theta][r][multiplier][stat]
translation	array of doubles	cartesian translation of mesh [x][y][z]
volumes	array of doubles	volumes of tally mesh [z][theta][r]

and the vessel fluence for each state point as well as an animation of the change over time. Examples of the images that can be produced are shown in § 5.

4. GEOMETRY

The Shift transport engine supports multiple geometric implementations [2, 7]. VERAShift supports two geometric packages:

RTK (Reactor ToolKit): The RTK geometry is a highly optimized, simple geometric model consisting of nested boxes and cylinders. It is very effective for modeling incore pressurized water reactor (PWR) geometries, including the baffle. However, it has significant limitations treating detailed vessel models. Also, RTK cannot be extended to model detailed boiling water reactor (BWR) cores due to lack of support for curved channel boxes.

GG (Exnihilo General Geometry): GG is a new general purpose geometric modeling implementation in Shift [12]. It supports arbitrarily defined, nested universes, arrays, and holes. Furthermore, many complex geometric shapes and quadratic surfaces are supported. GG can support detailed vessel models and user-defined excore features. It also supports all of the incore elements that RTK handles. Finally, the GG package can easily support any future VERAShift extensions for BWR modeling.

Even though RTK is optimized for PWR geometries, the GG package performs favorably on CE problems as illustrated in Fig. 4. In multigroup problems, where geometric operations account for the majority of runtime, RTK is $\sim 2\times$ faster. However, in CE problems, where physics operations consume the majority of runtime, GG is only 17% slower than RTK.

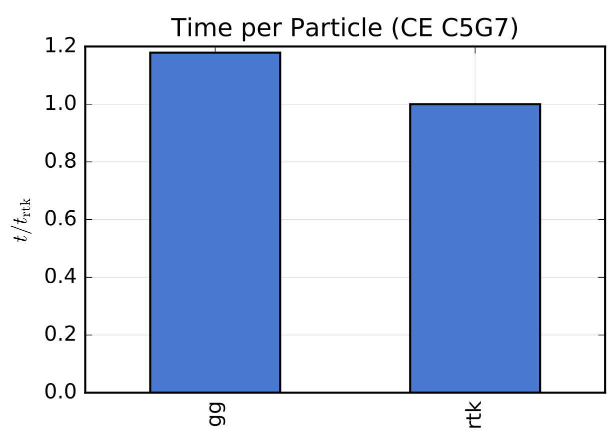


Figure 4. Performance of GG compared to RTK on reactor geometries with CE physics.

As discussed in § 2, the current implementation of VERAShift uses RTK for eigenvalue problems and GG for forward problems. There are plans to extend GG for use in eigenvalue problems, which will enable the use of detailed vessel models in eigenvalue problems. There are no geometric impediments to using GG for eigenvalue problems; the current limitations pertain to setting up specific tallies and output information for fission tallies in the GG model builder from the VERA input.

The control flow to build the geometry in VERAShift is roughly equivalent for both models. The full incore model is built from the VERA input specification, as shown in Fig. 5. A summary of the steps is given below:

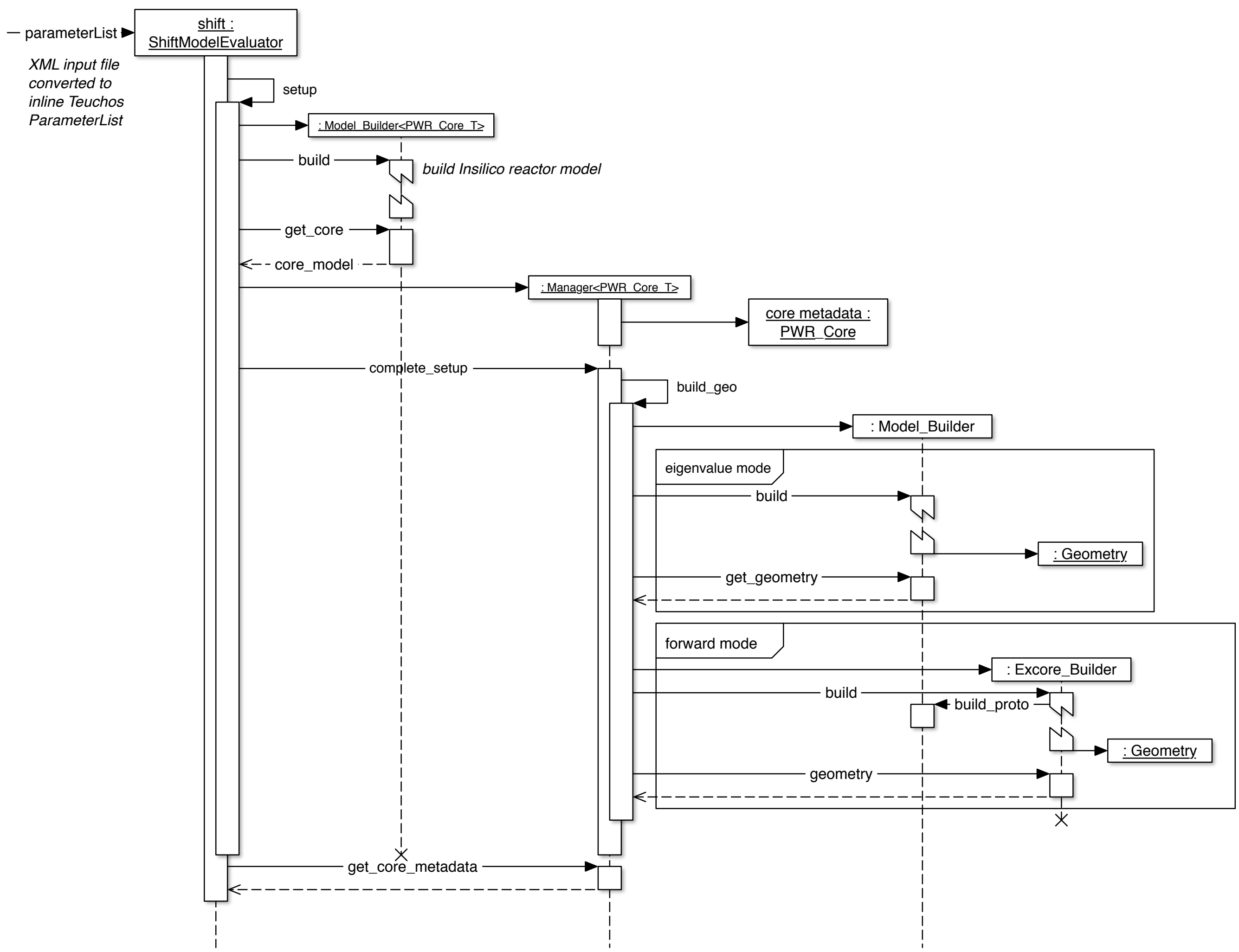


Figure 5. Control flow to construct reactor model geometries from VERA input.

1. Convert XML input to a `Teuchos::ParameterList`.
2. Build the core data model that is a reactor-specific data model defining pins, inserts, compositions, and other LWR-specific features.
3. Using the lightweight core model, build persistent core metadata that stores the reactor state.
4. Build the geometric model builder for either RTK or GG, depending on the problem mode.
5. In eigenvalue mode, use the geometric model builder to construct a complete incore reactor geometry for transport.
6. In forward mode, an excore builder is constructed that uses the geometric model builder to construct a complete incore model out to the inner barrel radius. The excore builder then adds the VERA vessel model, or alternatively, uses a user-supplied supplemental input model to define all geometric features outside of the core barrel.
7. The problem decomposition is constructed by the model builder.
8. The problem tallies are constructed by the model builder.

As described in step 3 above, VERAShift uses a metadata representation for the pincells and other core properties that is persistent throughout the simulation. The data transfer operations that were discussed in § 3.1 update the metadata after each VERA-CS state point calculation. On subsequent state points, the process for building the geometric model repeats steps 5-8.

In forward problems, the geometry can be constructed two ways: (1) as a full core model out to the outer vessel as defined in the standard VERA input, or (2) as a full core model out to the inner barrel radius with excore model details provided by a supplemental input file. In the latter case, the detailed core model can be translated to any position in space, which is a feature provided by the GG geometry. Using this mechanism, a detailed core model can be dropped into a larger scene that could include external core plant features not specified in the VERA input. The user must define tallies when using option 2. For option 1, all tallies and geometric model options are specified by the standard VERA input.

During work on this milestone, a useful feature was added to dump axial slices of the geometry (from a raytrace in Shift) to an HDF5 file. This file can then be loaded into Python and visualized using processing utilities in Shift. Figure 6 shows an example plot of a testing minicore with simple coloring. The compositions can also be included to define better coloring schemes as shown in Fig. 7 for a testing minicore based on AMA problem 5. An example geometric plot of using option 2 to define excore model details is shown in Fig. 8, which shows a minicore with an excore detector region defined. Note that in this model, everything outside of the inner barrel radius is defined in the supplemental input file.

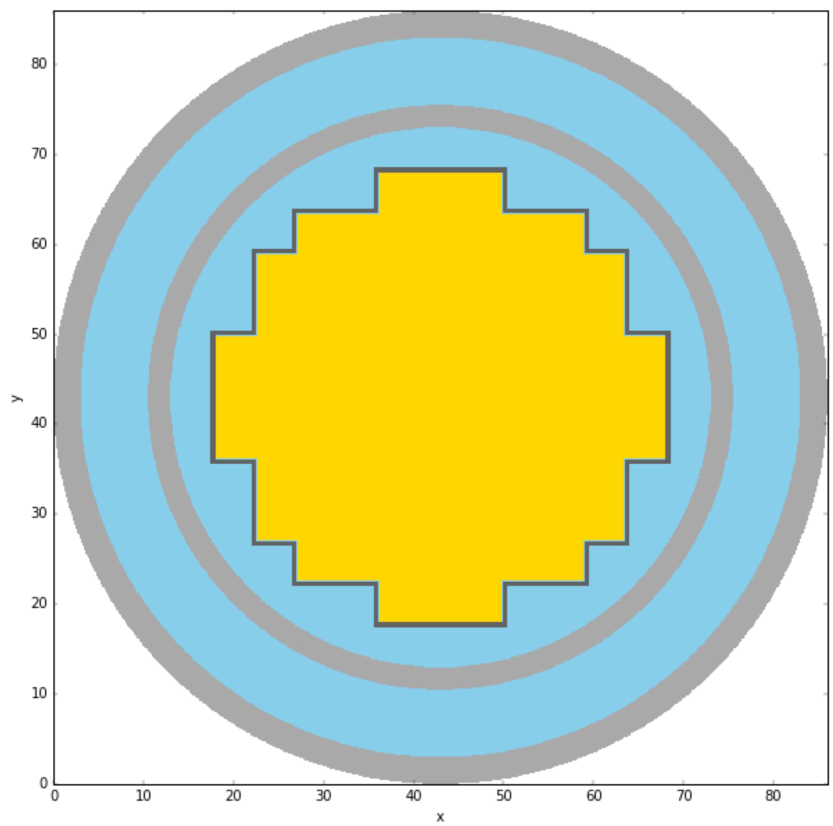


Figure 6. Axial slice of testing minicore with hand-tuned colors showing core, baffle, barrel, and vessel.

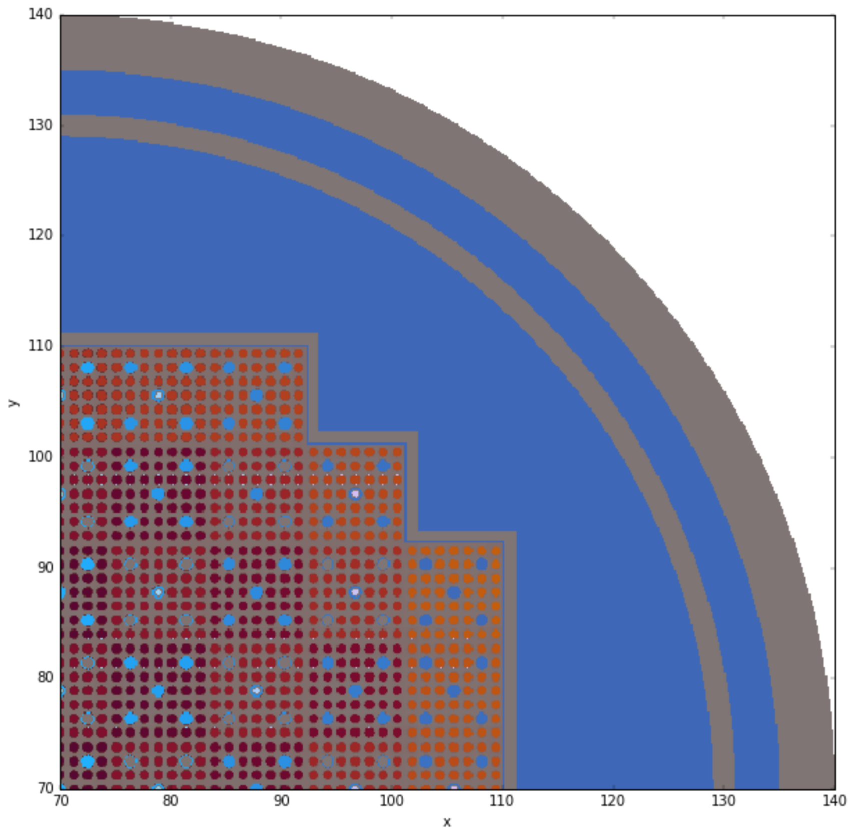


Figure 7. Axial slice of testing minicore NE quarter with composition coloring showing core, baffle, liner, and vessel.

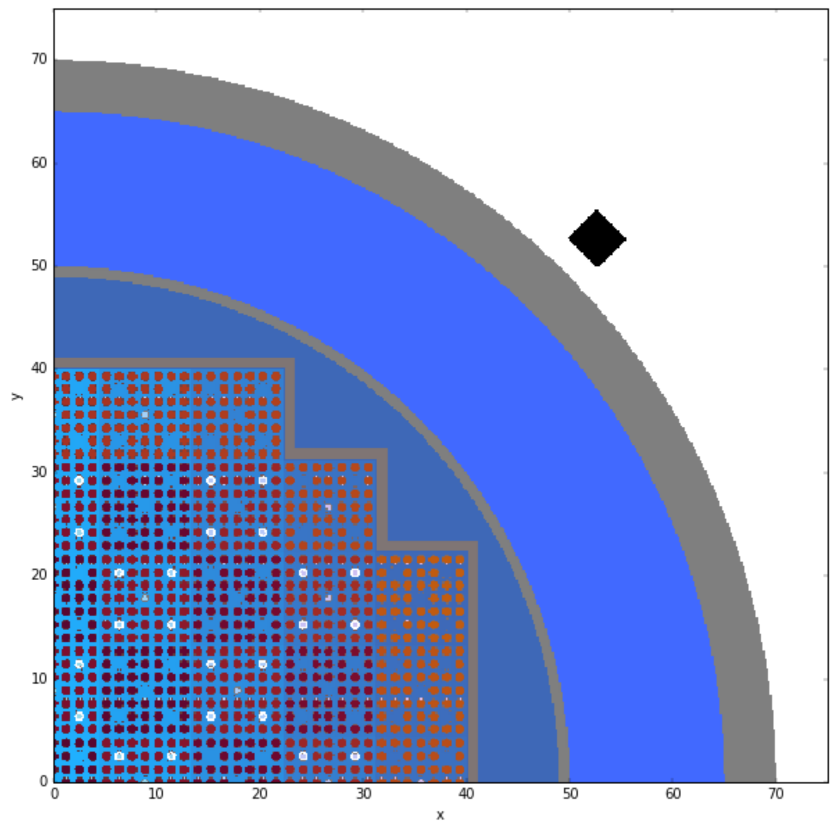


Figure 8. Axial slice of testing minicore NE quarter with excore detector region shown in black.

5. RESULTS

The purpose of this milestone was to implement and demonstrate the capability to produce vessel fluence calculated with VERA-CS. This section shows some of the results that were obtained for quarter-core simulations.

The fluence plots shown in this section are calculated and produced in VERAView. Fluence is defined as the time integration of the flux as given by

$$\phi(\boldsymbol{r}, E) = \int \phi(\boldsymbol{r}, E, t)\mathrm{d}t \,. \tag{1}$$

In practice, the fluence at a specific time step becomes

$$\phi_M^g(\boldsymbol{r}) = \sum_{m=0}^{M} \phi_m^g(\boldsymbol{r})\Delta t_m \,, \tag{2}$$

where M is the current time step and g is the energy group.

The flux in Eq. (2) comes from the energy-binned vessel tally for each state in the Shift HDF5 output file. The end time for a given state is calculated using the effective full power days (EFPD) from the MPACT HDF5 output file using Eq. (3).

$$t_m = 86400 \times \mathrm{EFPD}_m \tag{3}$$

Section 5.1 presents a detailed list of the tests performed on this capability within the Shift and VERAShift framework. Section 5.2 presents results running a two-dimensional version of AMA Problem 5 [13]. Studies investigating the performance and results on three-dimensional, quarter- and full-core problems are ongoing and are discussed in § 6.

5.1. TESTING

The V&V of excore modeling in VERAShift is an ongoing process. The first step is to verify the implemented methodology with unit and regression tests. These tests reside in VERAShift and Insilico, which is the frontend for Shift in VERA-CS. Table 4 gives the name and description of each unit test or set of unit tests associated with this milestone. They cover the full breadth of the features needed to complete this milestone.

Table 4. Relevant excore modeling unit/regression tests.

Test	Description
Insilico	
generators/geometry/test/tstAsbly_Array_GG_PWR	multiple tests of the ability to make the GG assembly array
generators/geometry/test/tstGG_Reactor	multiple tests of GG constructs for reactor core elements
generators/geometry/test/tstExcore_Builder	excore detector geometry setup using GG input file
generators/geometry/test/tstModel_Builder_GG_PWR	test building the GG model from the RTK model
generators/geometry/test/tstReflector_Builder_GG_PWR	test building the reflector region of the excore GG model
neutronics/test/tstManager_MC_FixedSource	test the full Shift excore solution procedure given VERA input
rtk_md/rtk_metadata/test/tstCore_Vessel	test setup of core vessel metadata including barrel and vessel
rtk_md/rtk_model/test/tstCore_Vessel_Model	test setup of core vessel model including barrel and vessel
rtk_md/rtk_model/test/tstPWR_Core_Model	test setup of vessel model as part of full PWR core
VERAShift	
tstDTK_Adapter	test mapping and fission source on assemblies and small core
tstDTK_Adapter_Full_Core	test mapping and fission grid using mini full core
2a_forward	runs AMA_2a through *vera_to_shift*; check VERA input and comps
2e_forward	runs AMA_2e through *vera_to_shift*; check VERA input and comps
3_mini_forward	runs minicore3 (single assembly) through *vera_to_shift*; check VERA input and comps
3a_forward	runs AMA_3a through *vera_to_shift*; check VERA input
4_mini_forward	runs minicore4 (multiassembly) through *vera_to_shift*; check VERA input
5_mini_forward	runs minicore5 (qtr core with vessel) through *vera_to_shift*; check VERA input
multistate_noTH	runs multistate problem through *vera_to_shift*; check VERA input
multistate_internalCTF	runs coupled multistate problem through *vera_to_shift*; check VERA input
VeraAPImpact	
inline_tests/tstCoupledMpact	check transfer and coupling of MPACT and Shift is working correctly
inline_tests/tstInline	check MPACT solution of case3962 and transfer through multiphysics distributor does not fail
inline_tests/tstMPACT_Geometry	check setup and MPACT geometry exists on proper processors for 4-mini with mirror and rotational symmetry

5.2. AMA PROBLEM 5A

The first quarter core to be addressed was AMA Progression Problem 5a in two dimensions [13]. A modified version of this problem was used that included depletion steps.[1] The VERA input is attached in the appendix. The core pads cannot currently be modeled in Shift, so these are shown in the following results. This problem was run on an institutional cluster at ORNL called *Panacea*. Shift simulated 1×10^{11} particle histories in its forward calculation to achieve low variance of the tallied flux in the vessel.

The following results show the normalized pin powers and vessel fluence. Normalized pin powers means the powers are normalized to the core-averaged linear heat rate. The normalized pin powers and the fluence are shown with different color scales and units (it not unitless).

Figure 9 shows the normalized pin powers and vessel fluence (with E > 0.1 MeV) at the beginning of the cycle produced in VERAView using the results from the VERAShift simulation. As expected, the vessel fluence is zero at the initial calculation. Figure 10 shows the normalized pin powers and vessel fluence (with E > 0.1 MeV) at the middle of the cycle. Finally, Fig. 11 shows the normalized pin powers and vessel fluence at the end of the cycle. We note that the highest fluence is along the diagonal of the quarter core, where the fuel is nearest to the vessel.

We see zero thermal flux in the vessel. The thermal range for neutrons is defined as energies below 0.5 eV. This is consistent with calculations performed by others [14]. Once FW-CADIS is integrated with the VERAShift interface, this zero thermal flux presence could be investigated further.

These vessel fluence results have relative statistical errors mostly less than 0.0004, as shown for step 7 and the end of cycle (EOC) in Figs. 12 and 13, respectively. The highest relative error occurs in one of the tally cells shown in Fig. 12. Note that the pin powers are shown as actual pin powers in these figures, with only the vessel fluence relative error shown according to the color scale.

Based upon these tests, Shift can transport one particle in approximately 5.2×10^{-4} seconds when running as part of VERAShift. Previous studies have shown the excellent scaling Shift can attain up to large numbers of processors in a weak and strong sense [2, 4]. Therefore, a user can determine an estimate of the time needed for the Shift calculation as part of VERAShift when considering the number of particles transported and number of processors allotted to Shift.

[1]Input provided by AMA courtesy of Andrew Godfrey.

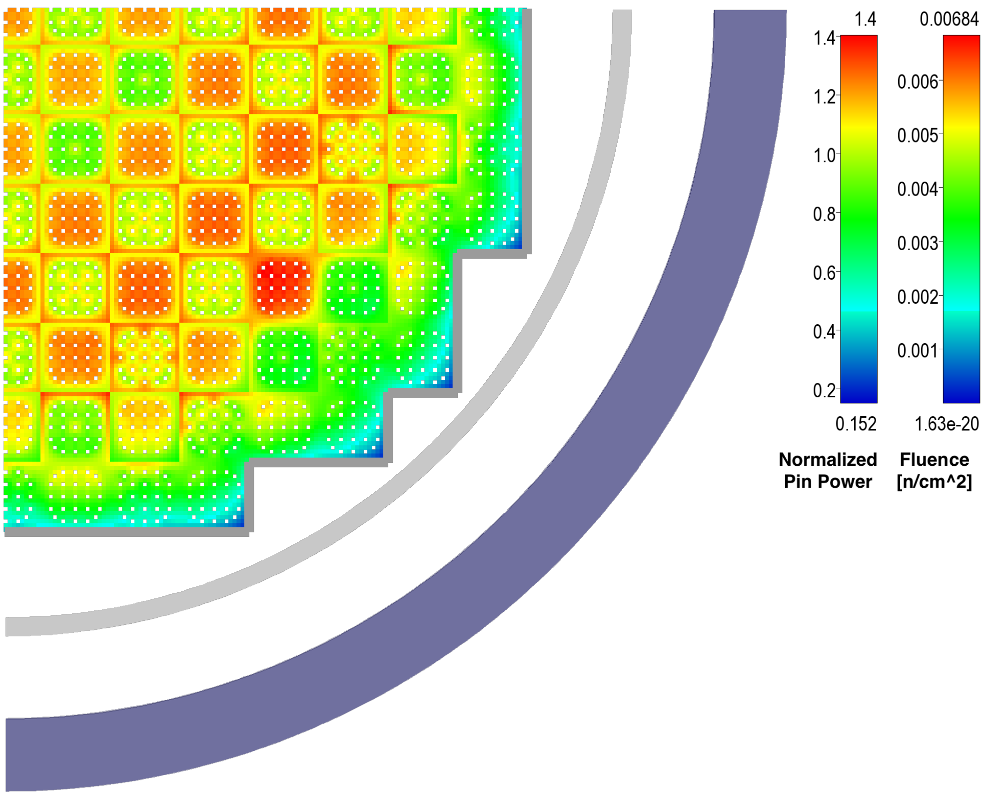


Figure 9. AMA Problem 5a-2D normalized pin powers and vessel fluence at initial step.

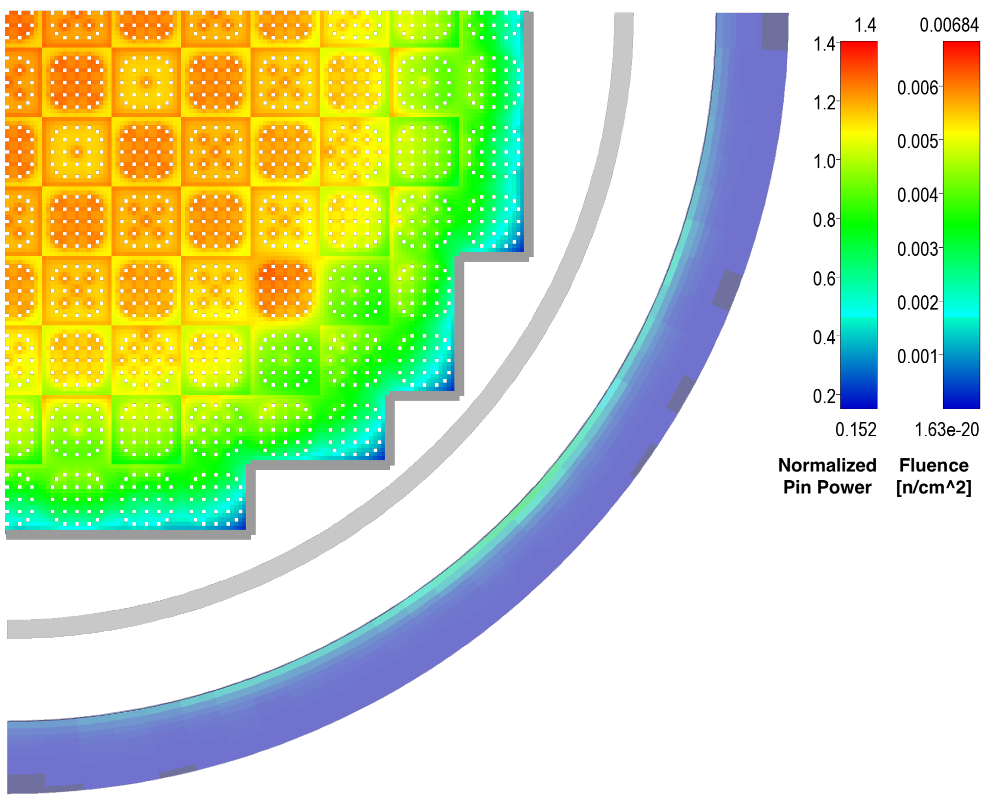


Figure 10. AMA Problem 5a-2D normalized pin powers and vessel fluence at middle of cycle.

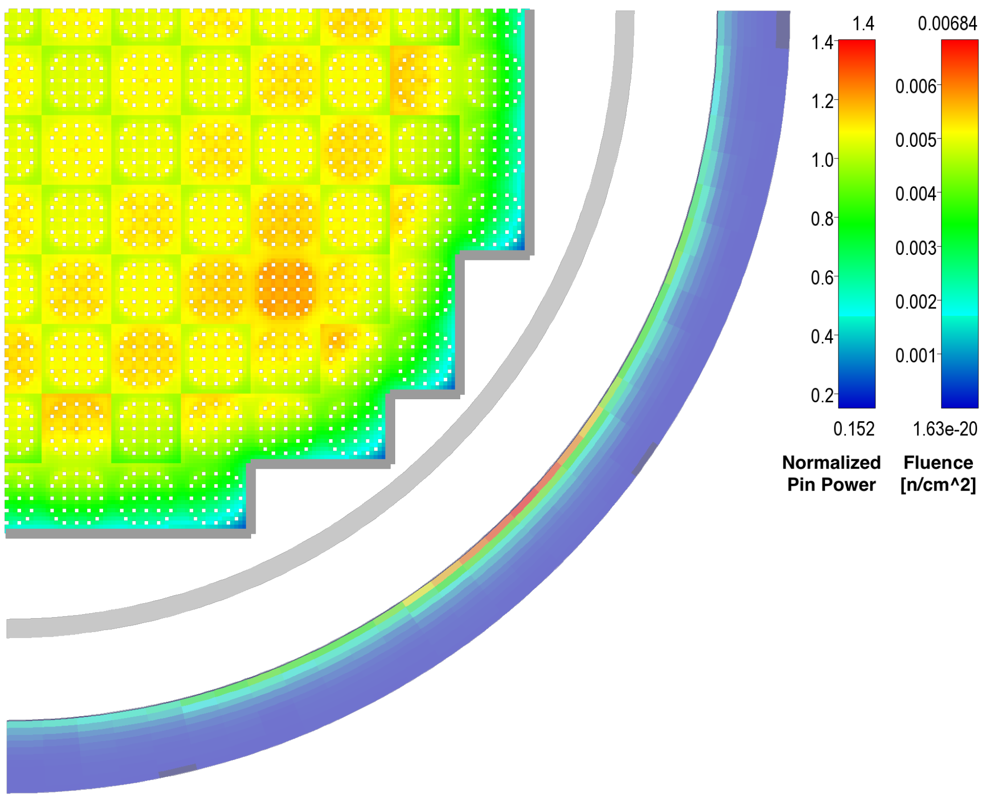


Figure 11. AMA Problem 5a-2D normalized pin powers and vessel fluence at end of cycle.

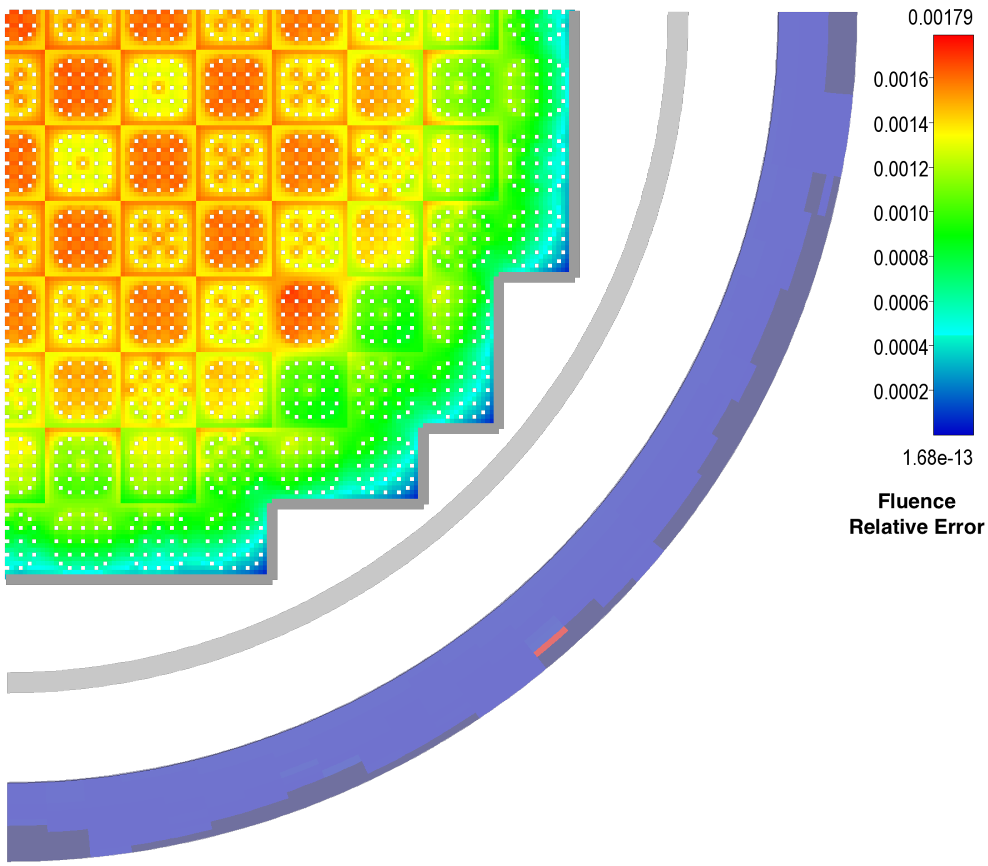


Figure 12. AMA Problem 5a-2D normalized pin powers and vessel fluence relative error at step 7.

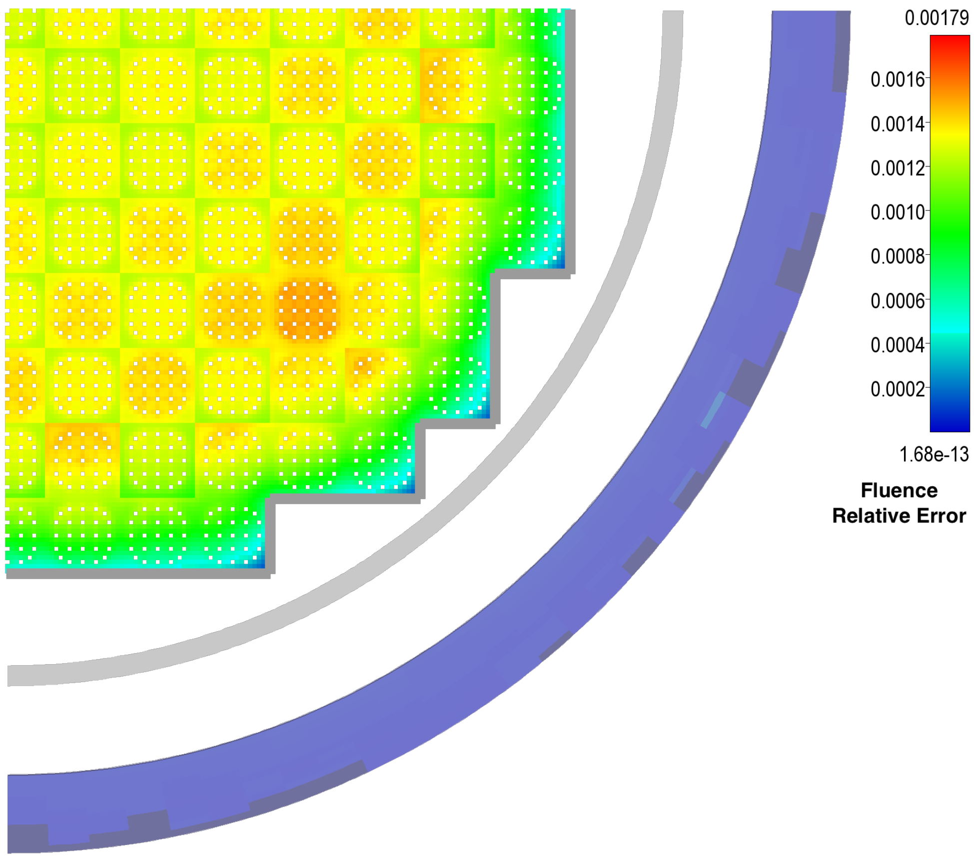


Figure 13. AMA Problem 5a-2D normalized pin powers and vessel fluence relative error at end of cycle.

6. SUMMARY

Under this milestone, excore calculation capability has been enabled with the VERAShift package as part of VERA-CS. This package couples VERA-CS and Shift in parallel to perform these calculations. VERA-CS performs an eigenvalue calculation for each state point and Shift performs either (a) an eigenvalue calculation for state point pin-power validation or (b) a fixed-source calculation to calculate vessel fluence and/or other excore quantities. In both cases the complete state of the core is transferred from VERA-CS to Shift.

This milestone effort has primarily focused on the excore capabilities in VERAShift; however, the coupling mechanics that have been implemented and improved are beneficial to both problem execution modes. VERAShift currently includes the following capabilities:

- node-based parallelism using domain replication,
- multistate,
- fission source coupling,
- temperature (fuel, clad, coolant) and coolant density coupling,
- flux tallying in core barrel, vessel liner, and vessel wall,
- supplemental geometric and tally input for excore model features and tallies, and
- output in VERA output format 1.0.

This milestone has demonstrated the ability to perform a vessel fluence calculation on a two-dimensional representative quarter core with reflecting symmetry.

As discussed in the following section, significant features still require testing and optimization. In particular, full core, three-dimensional simulations with full temperature and isotopic coupling are extremely memory intensive using domain replicated parallelism. This stems from the fact that each pin in the core has *unique* compositions. For example, a three-dimensional, full core AP1000® Shift model has $\geq 3 \times 10^6$ unique pin descriptions resident in memory on each domain. This is in addition to the temperature-interpolated CE cross section data for each pin. A single temperature point for ^{235}U requires $\sim$ 19 MB of memory. Thus, storing unique temperatures for all pin compositions replicated on each domain will limit the full core scaling of the current implementation.

Shift has several methods to address these memory limitations that can be integrated into VERAShift. This integration means to incorporate the use of the following methods into the managers that make up the VERAShift API to Shift. These methods include:

- Multiple-set, overlapping domain (MSOD) parallelism [2] that enables decomposition of the geometric model across nodes;
- Use of multithreaded concurrency for transport, which allows full scaling on a node without increasing the number of parallel domains per node. (However, it is not clear that even using a full NUMA node's memory on Titan will be sufficient to handle the full core cases.)

Alternatively, memory optimization methods for storing the unique composition data can be investigated as discussed in § 7.

7. FUTURE WORK

Enhancements to VERAShift fall in two basic categories: features and optimization. Furthermore, a strategy for validating the capability on excore problems is needed. The MC transport in Shift has undergone significant validation [4, 12], but intensive testing on excore problems using VERAShift has not been performed.

The following is a list of features to enhance and improve the excore modeling capabilities in VERAShift:

- allow flux tallying in core pads (mostly completed),
- fully enable general excore tallies (mostly completed),
- allow for multiple rings in fuel/depletable pins,
- allow for changing geometry between states to account for control rod movement,
- transfer isotopics for inserts and control rods,
- transfer and update the boron concentration in the moderator between states, and
- enable running the Shift calculation with FW-CADIS to improve the tally figure-of-merit (FOM) to reduce runtime requirements.

The next steps for optimizing VERAShift include the following:

- address memory issues for three-dimensional, quarter- and full-core problems,
- run AMA_Problem_7 and AMA_Problem_9,
- enable Westinghouse to develop inputs and test excore calculation capabilities, including detector region tallies (other tallies in addition to vessel fluence),
- run and analyze results for a range of representative reactor cores for different depletion cycles,
- investigate the non-separable space-energy fission source for excore problems, and
- develop a set of benchmarks for excore calculations.

8. ACKNOWLEDGMENTS

We would like to thank the efforts of Andrew Godfrey and Mark Baird for their valuable support in completing this milestone.

REFERENCES

[1] R. Montgomery. VERA tools and workflows. Technical Report CASL-U-2014-0054-001, Oak Ridge National Laboratory, March 2014.

[2] T. M. Pandya, S. R. Johnson, T. M. Evans, G. G. Davidson, S. P. Hamilton, and A. T. Godfrey. Implementation, capabilities, and benchmarking of Shift, a massively parallel Monte Carlo radiation transport code. *Journal of Computational Physics*, 308:239–272, 2016.

[3] CASL: Improving reactor performance with predictive science-based simulation technology that harnesses world-class computational power. Technical Report CASL-U-2014-0089-000, Oak Ridge National Laboratory, 2014.

[4] T. M. Pandya, T. M. Evans, G. G. Davidson, S. R. Johnson, and A. T. Godfrey. Shift verification and validation. Technical Report CASL-U-2016-1186-000, Oak Ridge National Laboratory, 2016.

[5] T. M. Pandya, S. R. Johnson, T. M. Evans, G. G. Davidson, and S. P. Hamilton. A Massively Parallel Monte Carlo Radiation Transport Package. In *Joint International Conference on Mathematics an Computation, Supercomputing in Nuclear Applications and the Monte Carlo Method*, April 2015.

[6] E. Biondo, G. Davidson, and T. Evans. Monte Carlo Fission Source Convergence Acceleration with Deterministically Estimated Fission Source Distributions. Technical Report ORNL/SR-2017/101, Oak Ridge National Laboratory, February 2017.

[7] S. R. Johnson, T. M. Evans, G. G. Davidson, S. P. Hamilton, and T. M. Pandya. Exnihilo transport code manual. Technical Report Release 6.2.0 (Dev), Oak Ridge National Laboratory, 2017.

[8] S. R. Slattery, P. P. H. Wilson, and R. P. Pawlowski. The Data Transfer Kit: A Geometric Rendezvous-Based Tool for Multiphysics Data Transfer. In *M&C 2013*. American Nuclear Society, May 5 - 9 2013.

[9] S. Palmtag and A. T. Godfrey. VERA Common Input User Manual. Technical Report CASL-U-2014-0014-002, CASL, February 2015.

[10] B. Collins et. al. MPACT VERA Input User's Manual Version 2.2.0. Technical Report CASL-U-2016-1109-000, CASL, June 2016.

[11] A. Godfrey and R. Lee. VERAView User's Guide. Technical Report CASL-U-2016-1058-000, Oak Ridge National Laboratory, March 2016.

[12] T. M. Pandya, S. R. Johnson, B. Langley, R. A. Lefebvre, and B. J. Marshall. Verification and validation of new SCALE sequence: CsasShift. Technical Report ORNL/SR-2016/401, Oak Ridge National Laboratory, 2016.

[13] A. T. Godfrey. VERA Core Physics Benchmark Progression Problem Specifications. Technical Report CASL-U-2012-0131-004, CASL, August 2014.

[14] F. A. Garner, L. R. Greenwood, and P. Roy. An assessment of potential gamma ray enhancement of embrittlement in ABWR pressure vessel walls. In *STP1325-EB*, number DOI: 10.1520/STP13857S. ASTM, 1999.

9. APPENDIX

This appendix includes snapshots of the VERA inputs used to produce the results shown from VERAShift in § 5.

AMA Progression Problem 5a in 2D

```
[CASEID]
  title 'CASL AMA Problem 5a-2d - Watts Bar Unit 1 Cycle 1 - Public'

[STATE]
  power    100.0
  tinlet   585 K
  tfuel    850 K
  boron    650            ! ppmB
  modden   0.7            ! g/cc
  rodbank  D 1            ! 1 = fully withdrawn
  sym      qtr
  feedback off
  xenon    equil
  search   boron
  kcrit    1.004
  deplete  GWDMT 0.0 0.1 0.5 <1..16> 16.939


[CORE]
  size   15               ! assemblies across core
  apitch 21.5
  rated  9.3258 131.68      ! MW, Mlbs/hr
  height 1.0

  core_shape
    0 0 0 0 1 1 1 1 1 1 1 0 0 0 0
    0 0 1 1 1 1 1 1 1 1 1 1 1 0 0
    0 1 1 1 1 1 1 1 1 1 1 1 1 1 0
    0 1 1 1 1 1 1 1 1 1 1 1 1 1 0
    1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
    1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
    1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
    1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
    1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
    1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
    1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
    0 1 1 1 1 1 1 1 1 1 1 1 1 1 0
    0 1 1 1 1 1 1 1 1 1 1 1 1 1 0
    0 0 1 1 1 1 1 1 1 1 1 1 1 0 0
    0 0 0 0 1 1 1 1 1 1 1 0 0 0 0

  assm_map
    1
    2 1
    1 2 1
    2 1 2 1
    1 2 1 2 2
```

```
    2 1 2 1 2 3
    1 3 1 3 3 3
    3 3 3 3

  insert_map
     -
    20  -
     - 24  -
    20  - 20  -
     - 20  - 20  -
    20  - 16  - 24 12
     - 24  - 16  -  -
    12  -  8  -

  crd_map
    A
    - -
    - - -
    - - - -
    A - - - A
    - - - - - -
    - - - - - -
    - - - -

  crd_bank
    D
    - -
    - - -
    - - - -
    D - - - D
    - - - - - -
    - - - - - -
    - - - -

  baffle ss 0.19 2.85

  vessel  mod 187.96        ! barrel IR (cm)
           ss 193.68        ! barrel OR (cm)
          mod 219.15        ! vessel liner IR (cm)
           ss 219.71        ! vessel liner OR / vessel IR (cm)
           cs 241.70        ! vessel OR (cm)

 ! neutron pad ID,OD arc lenth (degrees), angular pos (deg)
  pad ss  194.64 201.63 32 45 135 225 315

  xlabel  R P N M L K J H G  F  E  D  C  B  A
  ylabel  1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

  bc_top reflecting
  bc_bot reflecting

[ASSEMBLY]
  title "Westinghouse 17x17"
  npin 17
```

```
  ppitch 1.26

  fuel U21 10.257 94.5 / 2.110 u-234 0.017364
  fuel U26 10.257 94.5 / 2.619 u-234 0.021947
  fuel U31 10.257 94.5 / 3.100 u-234 0.026347

  cell 1     0.4096 0.418 0.475 / U21 he zirc4
  cell 2     0.4096 0.418 0.475 / U26 he zirc4
  cell 3     0.4096 0.418 0.475 / U31 he zirc4
  cell 4            0.561 0.602 / mod    zirc4      ! guide/instrument tube

  lattice LAT21
       4
       1 1
       1 1 1
       4 1 1 4
       1 1 1 1 1
       1 1 1 1 1 4
       4 1 1 4 1 1 1
       1 1 1 1 1 1 1 1
       1 1 1 1 1 1 1 1 1

  lattice LAT26
       4
       2 2
       2 2 2
       4 2 2 4
       2 2 2 2 2
       2 2 2 2 2 4
       4 2 2 4 2 2 2
       2 2 2 2 2 2 2 2
       2 2 2 2 2 2 2 2 2

  lattice LAT31
       4
       3 3
       3 3 3
       4 3 3 4
       3 3 3 3 3
       3 3 3 3 3 4
       4 3 3 4 3 3 3
       3 3 3 3 3 3 3 3
       3 3 3 3 3 3 3 3 3

  axial  1  0.0 LAT21 1.0
  axial  2  0.0 LAT26 1.0
  axial  3  0.0 LAT31 1.0

[INSERT]
  title "Pyrex"
  npin 17

  cell 1  0.214 0.231 0.241 0.427 0.437 0.484 / he ss he pyrex-vera he ss
```

```
rodmap  PY8
   -
   - -
   - - -
   1 - - -
   - - - - -
   - - - - - 1
   - - - - - - -
   - - - - - - - -
   - - - - - - - - -

rodmap  PY12
   -
   - -
   - - -
   1 - - -
   - - - - -
   - - - - - -
   - - - 1 - - -
   - - - - - - - -
   - - - - - - - - -

rodmap  PY16
   -
   - -
   - - -
   1 - - -
   - - - - -
   - - - - - 1
   - - - 1 - - -
   - - - - - - - -
   - - - - - - - - -

rodmap  PY20
   -
   - -
   - - -
   1 - - -
   - - - - -
   - - - - - 1
   1 - - 1 - - -
   - - - - - - - -
   - - - - - - - - -

rodmap  PY24
   -
   - -
   - - -
   1 - - 1
   - - - - -
   - - - - - 1
   1 - - 1 - - -
   - - - - - - - -
   - - - - - - - - -
```

```
  axial   8  0.0  PY8   1.0
  axial  12  0.0  PY12  1.0
  axial  16  0.0  PY16  1.0
  axial  20  0.0  PY20  1.0
  axial  24  0.0  PY24  1.0

[CONTROL]
  title "B4C with AIC tips"
  npin 17
  stroke  1.0 1

  cell 1  0.382 0.386 0.484 / aic he ss
  cell 2  0.373 0.386 0.484 / b4c he ss

  rodmap AIC
     -
     - -
     - - -
     1 - - 1
     - - - - -
     - - - - - 1
     1 - - 1 - - -
     - - - - - - - -
     - - - - - - - - -

  rodmap B4C
     -
     - -
     - - -
     2 - - 2
     - - - - -
     - - - - - 2
     2 - - 2 - - -
     - - - - - - - -
     - - - - - - - - -

  axial  A  0.0  AIC 1.0
  axial  B  0.0  B4C 1.0

[MPACT]
  num_space      73 ! 16
  par_method     ASSEMBLY ! EXPLICITRADIAL
  par_file       part_16r_qtr.txt

[SHIFT]
  transport             ce
  problem_mode          forward
  create_unique_pins    true
  Np                    100000000000
  output_fission_source true
  output_geometry       true
! tally_db
  neutron_bounds        2.0e7 1.0e6
```

```
num_theta            128 ! 32 per quadrant
radial_mesh          0 187.959999 189 190 191 192 193.680001 219.149999
                     219.7100001 221.71 223.71 225.71 227.71 229.71 231.71
                     233.71 235.71 237.71 239.71 241.70001
```