CASL-U-2019-1921-000

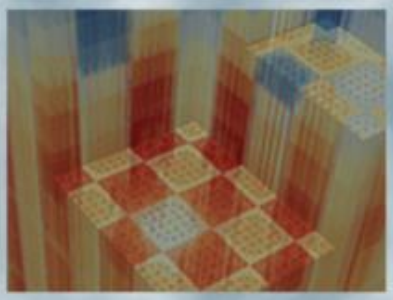
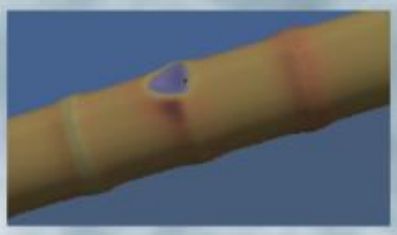
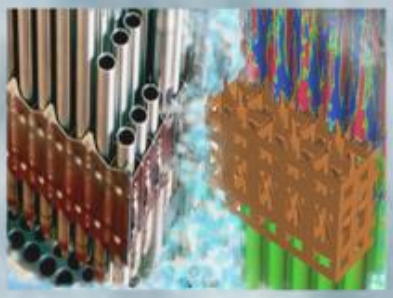
Consortium for Advanced Simulation of LWRs

VERAShift User's Manual

Tara Pandya[1], Katherine Royston[1], Thomas Evans[1], Benjamin Collins[1], Shane Stimpson[1], Shane Henderson[1], and Kevin Clarno[2]

[1]Oak Ridge National Laboratory
[2]University of Texas

1/14/2020

U.S. DEPARTMENT OF ENERGY | Nuclear Energy

Revision Log

Revision	Date	Affected Pages	Revision Description
0	1/14/2020	All	Initial Release

Document pages that are:

Export Controlled:	None
IP/Proprietary/NDA Controlled:	None
Sensitive Controlled:	None
Unlimited:	All

VERAShift User's Manual

Approvals:

Tara Pandya

Tara Pandya, VERAShift Product Software Manager

1-14-2020

Date

Steven Hamilton

Steven Hamilton, Independent Reviewer

1-14-2020

Date

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 these high–fidelity ex-core responses. The Virtual Environment for Reactor Applications (VERA) provides the automated capability to launch independent Shift [1] fixed-source and eigenvalue Monte Carlo (MC) calculations for user-specified state points during a standard Virtual Environment for Reactor Applications (VERA) calculation. VERA couples MPACT with COBRA-TF (CTF) to Shift to perform ex-core tallies for multiple state points concurrently, with each component capable of parallel execution on independent processor domains.

In these ex-core calculations, MPACT is coupled to CTF and performs the in-core depletion and heat transfer calculation, followed by a fixed-source Shift transport calculation including ex-core regions to produce ex-core responses. The fission source, fuel pin temperatures, moderator temperature and density, boron concentration, and fuel pin depleted isotopic compositions can be transferred to Shift from the MPACT calculation. Specifically, VERA performs fluence calculations in the core barrel outward to the end of the pressure vessel and detector response calculations in ex-core detectors. It also performs the requested tallies in any user-defined ex-core regions.

VERA takes advantage of the General Geometry (GG) package in Shift. This gives VERA 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 ex-core simulation comparison. The Consortium for Advanced Simulation of Light Water Reactors has developed a set of ex-core benchmark problems to include as part of the VERA verification and validation set of problems. The ex-core capability in VERA has been tested on small representative assembly problems, multi-assembly problems, as well as quarter-core and full-core problems. VERAView has also been extended to visualize these vessel fluence results from VERA.

This manual serves to present a guide to VERA users about the methodology behind ex-core calculations and the details of input, output, and analysis of results from these calculations. Details in this version of the manual are based on features in VERA 4.1.

Developer Team

The following people are contributors to the development of the specific parts of VERA relevant for ex-core calculations.

Shift Team

Elliott Biondo	ORNL
Gregory Davidson	ORNL
Thomas Evans	ORNL
Steven Hamilton	ORNL
Seth Johnson	ORNL
Tara Pandya	ORNL
Katherine Royston	ORNL
Jose Salcedo-Perez	ORNL
Casey Stocking	(formerly NCSU)

MPACT Team

Benjamin Collins (also PHI)	ORNL
Shane Stimpson	ORNL
Shane Henderson	ORNL

PHI Team

Kevin Clarno	University of Texas (formerly ORNL)
Stuart Slattery	ORNL

CTF Team

Robert Salko	ORNL

INF Team

Mark Baird	ORNL
Ron Lee	ORNL

Contents

Executive Summary iv

Developer Team v

Acronyms xiii

1 Introduction **1**

1.1 Purpose 2

1.2 Code functionality/limitations 2

1.3 Code installation 2

1.4 Training 2

1.5 Problem reporting methods 2

2 Input **4**

2.1 Parameters 4

2.2 Example Shift input blocks 11

3 Running **13**

3.1 Running with *verarun* 13

3.2 Running manually 14

4 Output **15**

4.1 General output 15

4.2 Eigenvalue calculation . . . 18
4.3 Simple vessel fluence calculation . . . 18
4.4 Detector response calculation . . . 20
4.5 Visualization with VERAView . . . 20

5 Examples **25**
5.1 Shift eigenvalue assembly example . . . 25
5.2 Shift forward minicore example . . . 28
5.3 CADIS vessel fluence minicore example . . . 35
5.4 Automated detectors example . . . 36
5.5 Ex-core file detector example . . . 39

6 Advanced and Experimental Features **49**
6.1 Core temperature homogenization . . . 49
6.2 Advanced postprocessing . . . 51

7 Summary **59**

References **59**

A Omnibus Materials Template File **62**

B Defaults **64**

C Geometry **66**

D Methodology **70**
D.1 Package overview . . . 70
D.2 Coupling algorithm . . . 73
D.2.1 Transfer data . . . 73

List of Figures

4.1 VERAView snapshot window of binned neutron flux for progression problem 9. 22

4.2 VERAView snapshot window of binned neutron fluence for progression problem 9. 23

4.3 VERAView snapshot window of pin adjoint for progression problem 9. 24

5.1 Ray trace of the progression problem 3a geometry at the axial midplane. 29

5.2 Ray trace of progression problem 9 geometry at axial midplane. 35

5.3 Ray trace of example geometry using the automated generation of the supplemental ex-core file with bioshield and detectors. 38

5.4 Ray trace of example at axial midplane with the ex-core regions defined manually with the supplemental ex-core input file. 48

6.1 Example of assembly-averaged temperature homogenization in the fuel of progression problem 7 minicore [2]. 50

6.2 Examples of ring-averaged temperature homogenization in the fuel of progression problem 7 minicore [2]. 50

6.3 Example of explicit peripheral pins with assembly-averaged temperature homogenization in the fuel of progression problem 7 minicore [2]. 51

6.4 Full–core material ray trace x–y slice at axial midplane. 52

6.5 Full–core material ray trace x–z slice at $\mathbf{y} = \mathbf{0}$. 53

6.6 Irradiation capsules cell ray trace x–y slice. 53

6.7 Source detector cell ray trace x–y slice. 54

6.8 Progression problem 9 state 2 fission source x–y slice. 55

6.9 Progression problem 9 adjoint source x–y slice. 56

6.10 Progression problem 9 slowest energy group adjoint flux x–y slice. 57

6.11 Progression problem 9 fastest energy group weight window lower bounds x–y slice. 58

C.1 Performance of GG compared with RTK on reactor geometries with CE physics. 67

C.2 Geometry origin relationships in VERA between Shift GG, RTK, and MPACT. In this figure, VERA refers to MPACT. 68

List of Tables

1 SHIFT block inputs related to transport and ex-core through VERA. 6

2 SHIFT block inputs related to hybrid MC/deterministic through VERA. 7

3 SHIFT block inputs related to hybrid MC/deterministic (continued) and output through VERA. 8

4 SHIFT block inputs related to eigenvalue, physics, and tallies through VERA. 9

5 Shift-related input parameters in the CORE block. 10

6 Experimental temperature homogenization SHIFT block input parameters. 10

7 Adjoint flux data fields in the Shift hierarchical data format 5 (HDF5) output file. 17

8 Fission source data fields in the HDF5 output file. 17

9 Vessel tally data fields in the Shift HDF5 output file. (All data fields are stored in the `/STATE_nnnn/tally/vessel_tally` group.) 20

10 Cell tally data fields in the Shift HDF5 output file. (All cell tallies are stored in the `/STATE_nnnn/tally` group.) 21

11 Ex-core capabilities of VERA. 72

12 Nuclide Watt energy spectra constants used in Shift. 75

List of Listings

2.1 Required STATE parameter. 4
2.2 Example SHIFT block with vessel tally in VERA common input. 11
2.3 Example SHIFT block with hybrid parameters and vessel tally in VERA common input. . . . 12
2.4 Example SHIFT block with ex-core parameters in VERA common input. 12
3.1 Example processor specification for CADIS ex-core calculation using 48 total processors. . . . 14
3.2 Example processor specification for forward ex-core calculation with *verarun* on 64 total processors. 14
4.1 General data in Shift HDF5 output file. 16
4.2 General data in Shift HDF5 output file (continued). 17
4.3 Data in Shift HDF5 output file from an eigenvalue calculation. 18
4.4 Data in Shift HDF5 output file from a vessel fluence calculation 19
5.1 VERA CASL progression problem 3a Shift eigenvalue input. 25
5.2 VERA Consortium for Advanced Simulation of Light Water Reactors (CASL) progression problem 9 minicore input. 29
5.3 VERA CASL progression problem 9 minicore CADIS input. 36
5.4 VERA CASL progression problem 9 minicore input with automated excore detectors. 37
5.5 VERA CASL progression problem 5 minicore VERA input using ex-core file. 39
5.6 VERA CASL progression problem 5 minicore ex-core file input. 46
A.1 Omnibus materials template file in VERA. 62
B.1 Default SHIFT block parameters. 64

List of Algorithms

1 Simplified VERAShift simulation flow. 74

Acronyms

BWR boiling water reactor

CADIS Consistent Adjoint Driven Importance Sampling

CASL Consortium for Advanced Simulation of Light Water Reactors

CE continuous energy

CTF COBRA-TF

DTK Data Transfer Kit

FW-CADIS Forward-Weighted CADIS

GG Exnihilo General Geometry

HDF5 hierarchical data format 5

MC Monte Carlo

MPI message passing interface

ORNL Oak Ridge National Laboratory

PBS portal batch system

PWR pressurized water reactor

RTK Reactor ToolKit

VERA Virtual Environment for Reactor Applications

1. Introduction

This manual describes the use of and some basic coupling in VERA [3] for ex-core capabilities. VERA provides the automated capability to launch independent Shift [1] fixed-source and eigenvalue MC calculations for user-specified state points during a standard VERA calculation. VERA couples MPACT [4] (including its coupling with COBRA-TF (CTF) [5]) to Shift to perform ex-core tallies for multiple state points concurrently, with each component capable of parallel execution on independent processor domains. In these ex-core calculations, MPACT is coupled to CTF and performs the in-core depletion and heat transfer calculation, followed by a fixed-source Shift transport calculation on the entire geometry including ex-core regions to produce ex-core responses. The fission source, fuel pin temperatures, moderator temperature and density, boron concentration, and fuel pin depleted isotopic compositions can be transferred to Shift from the MPACT calculation (if sufficient memory is available). For quarter core and full core problems typically only the fission source and boron concentration are transferred.

For default vessel fluence calculations, minimal user intervention outside the common VERA input is required [6]. This is also true for certain default ex-core detector response calculations in the pressure vessel and bioshield. For more detailed ex-core calculations, ex-core modeling in VERA provides users with the capability to insert a detailed core model into a broader user-defined ex-core model. For example, a detailed core model can be inserted into a containment geometry that includes nozzles and detector locations.

For ex-core calculations, Shift can be run in analog or Consistent Adjoint Driven Importance Sampling (CADIS) [7] mode. Forward-Weighted CADIS (FW-CADIS) [8] is not currently supported. Shift is coupled to MPACT in VERA for validation and ex-core calculations. Shift uses the information transferred from MPACT for a prior state point to automatically generate a detailed geometric core model for that state point. Shift then samples the fission source provided by MPACT and runs a fixed-source calculation to determine the fluence in the vessel and ex-core features. This fission source can include only the spatial distribution, the full multigroup energy spectrum, or a nuclide-dependent source distribution from MPACT. In the case where only the spatial distribution from MPACT is used, a ^{235}U Watt energy spectrum is used in the Shift calculation. For the case of a nuclide-dependent fission source, Shift samples a Watt energy spectrum based on four nuclides (^{235}U, ^{238}U, ^{239}Pu and ^{241}Pu) sampled in a spatial location.

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 fixed-source calculation. Also, the current implementation does not have the capability to sample from a user input gamma source. The user should note that the coupled n-γ calculation is not optimized and will run significantly slower than neutron-only calculations.

1.1 Purpose

This document provides the details for creating inputs and running with the VERAShift executable. The summary of all VERAShift documentation can be found in the VERAShift Software Management Plan document.

1.2 Code functionality/limitations

VERAShift was developed specifically for ex-core reactor calculations. The VERAShift Verification and Validation Manual will state the high–level requirements. The VERAShift Software Requirements, Test Plan, and Test Report document states low-level requirements and documents the tests that ensure it meets these requirements. The user should assume that any functionality not clearly stated in the high– or low–level requirements lists is not a supported function of VERAShift.

VERAShift includes more features than are necessary to support ex-core calculations. These features are considered research features and their quality/correctness is not guaranteed due to limited testing. Any features that are not stated in the aforementioned requirements lists should be considered to be experimental and their use is cautioned.

1.3 Code installation

Code installation instructions are covered in the VERA release notes that are provided for the version of VERAShift being distributed. It is suggested that access to the installed software be limited to trained and qualified users.

1.4 Training

The intended VERAShift user is one who is familiar with computational neutronics codes that include deterministic, MC, and hybrid methods. Prior to using VERAShift, the user should read through the VERAShift this document fully. The VERA Users Group does periodically hold training courses including the use of VERAShift for ex-core calculations. Contact pandyatm@ornl.gov for further information.

1.5 Problem reporting methods

The problem reporting method is described in VERA-QA-010: VERA Problem Reporting and Corrective Action. These reports can be found at https://www.casl.gov/corrective-actions.

The following sections present the details of running ex-core calculations through VERA. This revision of the manual presents capabilities available in VERA 4.1 which uses SCALE version 6.3beta3. Section 2 goes

over the input options specific to ex-core calculations with VERA and § 3 explains how to run the ex-core calculations. The output is discussed in § 4 and § 5 gives several examples of vessel fluence and ex-core detector calculations that can be run through VERA. Advanced and experimental capabilities are described in § 6 and a summary of ex-core capabilities in VERA is given in § 7. Finally, Appendix A gives the material template file used for the automated generation of a supplemental ex-core file, Appendix C gives details on adding and integrating the ex-core geometry with VERA, and Appendix D explains the methods and code-coupling algorithms used in VERA for performing these ex-core calculations.

2. Input

For VERA ex-core calculations, the Shift parameters are defined in the common VERA input [6] in the **SHIFT** block. All of the usual inputs for the **STATE**, **CORE**, **ASSEMBLY**, **INSERT**, **CONTROL**, **DETECTOR**, **EDITS**, **MPACT** [4], and **COBRATF** [5] blocks apply to their respective setups. To run any Shift calculation through VERA, the parameter in Listing 2.1 must be included in the first **STATE** block. Do not include this extra parameter in subsequent **STATE** blocks in the same input.

Listing 2.1: Required STATE parameter.

```
[STATE]
  excore_transport on     ! Needed so MPACT knows Shift will run
```

§ 2.1 outlines all of the parameters available in the **SHIFT** input block. § B presents the Shift defaults set for ex-core calculations with VERA. Finally, § 2.2 shows a few example **SHIFT** input blocks for typical ex-core problems.

2.1 Parameters

Tables 1 through 4 show the parameters regulating the Shift calculations used in the **SHIFT** block of the VERA common input. Table 5 gives parameters located in the **CORE** block of the VERA input used for Shift calculations, and Table 6 gives several experimental parameters. The user should be aware of several caveats when changing some of these advanced parameters.

1. If **use_pole_data** is *true*, the other physics broadening parameters do not apply **(broaden_xs, delta_t, energy_tol, temperature_tol, union_energy)**.
2. The **core_translate** should correspond to the translation used in the supplemental ex-core geometry.
3. When using the supplemental ex-core geometry in CADIS mode, **hybrid_tally_names** and **hybrid_multiplier_names** must be specified. The **hybrid_tally_names** must correspond to tally names in the supplemental ex-core geometry, and only *flux* for **hybrid_multiplier_names** is supported at this time.
4. **num_blocks_i** $\times$ **num_blocks_j** should equal the total number of Shift processors.
5. Shift does not model any ex-core geometry (core only) when the **problem_mode** is set to *eigenvalue*.

When using the automatic generation of the supplemental ex-core geometry by specifying the **bioshield** parameter, the user needs to keep in mind the following items:

1. Materials specified in the **bioshield** and **det** definitions must exist in the template file located at `scripts/Init/omnibus_template.txt` (see Appendix A).
2. The generated file will use the base of the VERA input filename and the `.omn` extension. The user must ensure that once this file is converted to XML, the name matches the filename specified next to the **excore_filename** parameter in the **SHIFT** block of the VERA input file.
3. In CADIS mode, if the VERA input file specifies **hybrid_tally_names** that do not match a detector ID, the parameter will not be changed. This means the user must know the name of the tally they wish to optimize. If the user-specified tally name does not exist, the calculation is killed.

Table 1: SHIFT block inputs related to transport and ex-core through VERA.

Parameter	Description	Type	Default
create_unique_pins	make all pincells unique compositions	bool	false
do_transport	perform MC transport	bool	true
fiss_src_spectrum	type of spectrum to use for fission source	string in {"mpact", "nuclide_watt", "u235_watt"}	"nuclide_watt"
lost_particle_error_tolerance	fraction of lost particles to tolerate before aborting	double in (0,1)	1e-6
mode	type of particles to transport	string in {"n", "np"}	"n"
Np	number of particles to transport	double	1e6
num_threads	number of threads per core for transport	integer	1
problem_mode	run mode	string in {"eigenvalue", "forward", "cadis"}	"cadis"
seed	initialize global random number generator object	integer	121434
temp_transfer	which temperatures to couple with CTF	string in {"all", "none", "pin"}	"all"
track_isotopes	which set of isotopes to transfer	string in {"full", "short"}	"short"
transfer	what to transfer from MPACT	string in {"all", "fiss_src", "isotopics", "temps"}	"fiss_src"
verbosity	how often to print about particles being transported	string in {"none", "low", "medium", "high"}	"low"
	Ex-core		
bc_bnd_mesh	boundary mesh boundary conditions on all sides	array of 6 strings in {"reflect", "vacuum"} (-X, +X, -Y, +Y, -Z, +Z)	{"vacuum", "vacuum", "vacuum", "vacuum", "vacuum", "vacuum"}
core_translate	position to translate center of core	array of 3 doubles	{0.0, 0.0, 0.0}
excore_filename	name of Omnibus XML file with ex-core features and tallies	string	none
hybrid_multiplier_names	ex-core tally multipliers to optimize for CADIS	array of strings	none
hybrid_tally_names	ex-core tally names to optimize for CADIS	array of strings	none
raytrace_resolution	resolution for geometry ray trace	integer	1024
vera_pressure_vessel	pull in outer pressure vessel radius from VERA input	bool	false
x_bnd_mesh	boundary mesh x-axis limits	array of 2 doubles	none
y_bnd_mesh	boundary mesh y-axis limits	array of 2 doubles	none
z_bnd_mesh	boundary mesh z-axis limits	array of 2 doubles	none

Table 2: SHIFT block inputs related to hybrid MC/deterministic through VERA.

Parameter	Description	Type	Default
	Hybrid MC/Deterministic		
adjoint	perform adjoint solve	bool	true
azimuthals_octant	number of azimuthal angles per octant for S_N	positive integer	4
cell_homogenize	homogenize material in cells	bool	true
downscatter	downscatter-only	bool	false
eq_set	solution method or spatial discretization	string in {"bld", "bld_2d", "ld", "sc"}	"sc"
extend_axial_mesh_size	adjoint axial mesh size in ex-core region	double	5.0
grp_collapse_src	source to do group collapse	array of doubles	depends on **xs_library**
iterate_downscatter	iterate over downscatter groups	bool	false
max_delta_z	maximum core mesh size in z	double	5.0
mesh	number of mesh cells per pincell	positive integer	1
new_grp_bounds	collapsed group lower boundaries	array of decreasing doubles	{6.0653e6, 3.6788e6, 2.3457e6, 1.6530e6, 8.2085e5, 2.4176e4, 1.0130e2, 1.0000e-5}
num_blocks_i	number of partitions (processors) in x	positive integer	10
num_blocks_j	number of partitions (processors) in y	positive integer	10
num_groups	number of energy groups	positive integer	8
num_sets	number of energy sets	positive integer	1
num_z_blocks	number of pipelining blocks in z	positive integer	depends on number of processors
output_adjoint	output adjoint flux and pin adjoint to Shift HDF5 file and adjoint source to separate HDF5 file	bool	false
partition_upscatter	partition energy over upscatter groups only	bool	false
pin_partitioning	partition mesh over pincells	bool	false
Pn_correction	use outscatter-corrected diffusion coefficient to reduce memory in solve	bool	false
Pn_order	order of moments	integer	0
polars_octant	number of polar angles per octant for S_N	positive integer	2
quad_type	type of S_N quadrature	string in {"galerkin", "glproduct", "ldfe", "levelsym", "qr"}	"qr"

Table 3: SHIFT block inputs related to hybrid MC/deterministic (continued) and output through VERA.

Parameter	Description	Type	Default
	Hybrid MC/Deterministic		
refl_mesh_size	radial mesh size in ex-core region	double	5.0
src_disc_l2_error	maximum L2 error for point-sampling discretization	double in (0,1)	1e-3
src_disc_max_samples	maximum number of discretization samples	integer	1e9
src_disc_samples_per_batch	number of samples per point-sampling batch	integer	1e7
store_fulcrum_string	save fulcrum string as file	bool	false
upscatter_max_itr	maximum number of iterations for deterministic solve	integer	1e3
upscatter_solver	upscatter solver	string in {"gauss_seidel", "gmres"}	"gauss_seidel"
upscatter_subspace_size	upscatter solver subspace size	integer	20
upscatter_tolerance	upscatter solver convergence tolerance	double	1e-4
upscatter_verbosity	log verbosity for upscatter solver	string in {"none", "low", "medium", "high"}	"high"
within_group_max_itr	maximum number of iterations for within group solver	integer	1e3
within_group_solver	within-group solver	string in {"gmres"}	"gmres"
within_group_subspace_size	max subspace size for within group solver	integer	20
within_group_tolerance	within group solver convergence tolerance	double in (0,1)	1e-4
within_group_verbosity	log verbosity for within group solver	string in {"none", "low", "medium", "high"}	"low"
ww_decomp	type of adjoint flux decomposition	string in {"full", "separable"}	"separable"
xs_library	name of SCALE MG data library file	string	"v7-56"
	Output		
do_output	do Shift output	bool	true
local_log	level of local node log information level	string in {"debug", "diagnostic", "status", "info", "warning", "error", "critical"}	"error"
global_log	level of global log information	string in {"debug", "diagnostic", "status", "info", "warning", "error", "critical"}	"info"
ray trace_levels	z levels to ray trace geometry and output	array of doubles	midpoint of active fuel or core axial levels
rtk_output_format	format of core RTK geometry output file	string in {"xml", "hdf5"}	"hdf5"
output_fission_source	output the initial fission source for each state	bool	false
output_geometry	output HDF5 files of ray traced geometry (initial) and compositions (each state)	bool	true
output_micro_tally	output micro reaction tallies	bool	false
output_ww	output the weight windows if used	bool	false

Table 4: SHIFT block inputs related to eigenvalue, physics, and tallies through VERA.

Parameter	Description	Type	Default
	Physics Treatment		
broaden_xs	apply cross–section doppler broadening	bool	true
ce_lib_path	path to alternate SCALE CE data library file not in VERAData	string	none
dbrc	apply doppler–broadening resonance correction	bool	false
delta_t	finite difference grid spacing for Leal-Hwang interpolation of cross sections	positive double	1.0
energy_tol	relative difference for considering two energy points equal	double in (0,1)	1e-10
temperature_tol	tolerance for reusing existing broadened cross sections	positive double	4.0
thermal_energy_cutoff	cutoff for treatment of thermal neutrons	double	10.0 eV
union_energy	unionize lower and upper library temperature energy grids	bool	true
use_pole_data	use pole CE data for on-the-fly doppler broadening	bool	false
	Tallies		
do_micro_tally	tally microscopic reaction rate in eigenvalue mode	bool	false
gamma_flux	tally the photon flux in each pincell	bool	false
micro_zaids	nuclides to tally micro reactions in eigenvalue mode	array of integers	{92235, 92238}
micro_rxns	MT of micro reactions to tally in eigenvalue mode	array of integers	{18, 102}
radial_mesh	radii for flux tally in vessel	array of doubles	vessel radii
n_bounds	neutron energy bounds for tallies	array of decreasing doubles	none
num_axial	number of axial levels for flux tallies in vessel	integer	1
num_theta	number of theta divisions for flux tallies in vessel in $[0, 2\pi]$	integer	1
p_bounds	photon energy bounds for tallies	array of decreasing doubles	none
	Eigenvalue Only		
num_cycles	total number of eigenvalue cycles	integer	50
num_inactive_cycles	number of inactive eigenvalue cycles	integer	10

Table 5: Shift-related input parameters in the CORE block.

Parameter	Description	Type	Default
bioshield	bioshield materials and radii	array of alternating strings and doubles	none
det	detector id, type, radii, materials, heights, response type, and well type	strings and doubles; type: string in {"PWR", "SRC"}, response type : string in {"b10", "u235"}, well type: string in {"none", "wedge"}	none
det_locations	detector id, radius, degree, and elevation	string and double	none

Table 6: Experimental temperature homogenization SHIFT block input parameters.

Parameter	Description	Type	Default
homog_type	if homogenizing, go by assembly or in rings	string in {"assem", "rings"}	none
homog_explicit_ring	radius to homogenize within and have explicit pins outside of	positive double	none
homog_pin_rings	homogenize according to pin locations, instead of assembly locations	bool	false
homog_ring_radii	radii for homogenization, if homog_type is rings	array of increasing doubles	none

2.2 Example Shift input blocks

An example of the **SHIFT** block in the VERA input with a default vessel tally is shown in Listing 2.2. This example would run an analog Shift calculation using 1×10^7 particle histories with the specified vessel fluence tally and the other default parameters listed in Listing B.

Another example of the **SHIFT** block that includes modified parameters for a CADIS run is shown in Listing 2.3. The modified S_N parameters control the deterministic adjoint solve. This example would run a CADIS Shift calculation on 25 processors with the specified vessel fluence tally. The adjoint calculation would use the collapsed group structure (collapsed from the default 56 group SCALE library).

Finally, an example of the **SHIFT** block using a supplemental ex-core geometry file and associated input parameters is shown in Listing 2.4. The details of the ex-core geometry and tallies are defined in a supplemental ex-core file (not shown here). This input would perform a CADIS Shift calculation to optimize the flux reaching several detectors and produce a detector response. A few things to note: first, the full pressure vessel is being pulled in from the VERA input to combine with a supplemental ex-core file geometry; second, the core is translated to correspond with the origin used by the supplemental ex-core geometry. Finally, the boundary mesh given is for quarter core symmetry and is input based on the supplemental geometry origin. More details on setting up and using a supplemental ex-core geometry file are given in § 5 and § C.

Section 5 presents full example VERA inputs for several different ex-core calculations. This includes an example of using the automated detector and bioshield setup. For further details on input, running, and output of the experimental homogenization methods (with inputs given in Table 6), please contact Tara Pandya (pandyatm@ornl.gov) at Oak Ridge National Laboratory (ORNL).

Listing 2.2: Example SHIFT block with vessel tally in VERA common input.

```
1 [SHIFT]
2   problem_mode          forward
3   Np                    10000000
4   output_fission_source true
5   num_axial             10
6   num_theta             8
7   radial_mesh           65 68 70 72 75
8   n_bounds              1000000 10000 1000 1000 1 0.1 0.01
```

Listing 2.3: Example SHIFT block with hybrid parameters and vessel tally in VERA common input.

```
[SHIFT]
  problem_mode          cadis
  Np                    10000000
  output_fission_source true
  output_adjoint        true
  num_blocks_i          5
  num_blocks_j          5
  num_groups            4
  new_grp_bounds        6.0653E+06 3.6788E+06 2.3457E+06 1.6530E+06
  max_delta_z           4.0
  num_axial             15
  num_theta             128
  radial_mesh           65 66 67 69 70 72 75
  n_bounds              2000000 100000
```

Listing 2.4: Example SHIFT block with ex-core parameters in VERA common input.

```
[SHIFT]
  output_fission_source   true
  vera_pressure_vessel    true
  excore_filename         dummy_geometry.excore.xml
  raytrace_levels         10.0 50.0 100.0 200.0 300.0
  hybrid_tally_names      det_tally src_det_tally
  hybrid_multiplier_names flux flux
  core_translate          -25.0 -25.0 0.0
  bc_bnd_mesh             reflect vacuum reflect vacuum vacuum
  x_bnd_mesh              0.0 26.0
  y_bnd_mesh              0.0 26.0
  z_bnd_mesh              0.0 400.0
```

3. Running

There are two ways to run ex-core calculations with VERA: with *verarun* and manually. Note that since MPACT and Shift run on their own sets of processors, at least two processors are always required to run ex-core calculations. When running a CADIS calculation, the number of processors that Shift can run on is limited by the decomposition chosen for the adjoint deterministic calculation (as mentioned in § 2.1).

3.1 Running with *verarun*

The simplest way to run an ex-core calculation is to use the *verarun* executable, assuming it is installed on the machine. If the VERA environment is set up on the machine, a simple run command using a VERA input file called *sample.inp* can be used, such as the following example.

```
verarun sample.inp
```

This command automates all of the file generation (detailed in § 3.2) and submits a job using whatever job scheduler is used by the machine (portal batch system (PBS) or Slurm). If running an eigenvalue or forward-mode calculation, Shift will only run on one processor unless the *nprocs* parameter is given in a RUN block in the VERA input to specify the total number of processors. If the *nprocs* parameter is given, Shift will use the leftover processors that remain after the requested number for MPACT is fulfilled. The MPACT VERA user input manual provides details on the MPACT parallel decomposition [4].

The *verarun* executable can also be run without automatically submitting your job to allow for editing the submission script, allowing customization of the processor layout and other advanced options. This command is the following:

```
verarun -x sample.inp
```

Listing 3.1 shows the relevant parameters in the VERA input to specify a CADIS calculation with 32 processors for MPACT and 16 processors for Shift. Another simple example showing the relevant parameters for running a Shift forward calculation with 32 processors for MPACT and 32 processors for Shift is given in Listing 3.2.

Listing 3.1: Example processor specification for CADIS ex-core calculation using 48 total processors.

```
[MPACT]
 num_space    32

[SHIFT]
 problem_mode  cadis
 num_blocks_i  4
 num_blocks_j  4
```

Listing 3.2: Example processor specification for forward ex-core calculation with *verarun* on 64 total processors.

```
[MPACT]
 num_space   32

[SHIFT]
 problem_mode forward

[RUN]
 nprocs      64
```

3.2 Running manually

All of the automation encapsulated in *verarun* can also be performed separately using the *vera_to_shift* executable. If using a supplemental ex-core file, the ASCII input must be converted to XML before running *vera_to_shift*. The step-by-step process for running an ex-core calculation is as follows:

1. Run the *react2xml* Perl script to convert the VERA input file to XML.
2. If using a supplementary ex-core geometry file, run *excore2xml* on the Omnibus input file.
3. Run *vera_to_shift*.

For example, for the VERA input file *sample.vera* and supplemental ex-core geometry file *sample.excore.omn*, the following series of commands will convert the inputs to XML and run the problem on 16 processors.

```
./react2xml sample.vera
```

```
./excore2xml sample.excore.omn -o sample.excore.xml
```

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

Finally, the ability to run Shift without coupling to MPACT using the VERA common input also exists. This executable will run a Shift eigenvalue calculation of the given single statepoint VERA input and produces k_{eff} and pin powers written to the Shift HDF5 output file. Using the same preceding example, the first two commands would be the same to convert the inputs to XML. To run the problem on 16 processors the command is the following.

```
mpirun -np 16 ./vera_shift --i=sample.xml
```

4. Output

From an ex-core calculation, the relevant output quantities are located in different output files depending on the type of ex-core calculation performed. The usual VERA HDF5 output file is produced along with the associated output files from MPACT and CTF. This output file can include some results from the Shift calculation, while the majority of the Shift output is found in the separate **.shift.h5* output file.

The following section presents an overview of the data available in output files and where the user can expect specific data to be located based on the type of calculation. Upon completion of the ex-core calculation, the user can postprocess and plot the tallies and other quantities of interest over all states from the output files. This section presents the plotting capability in VERAView [9]. Details on some of the advanced postprocessing that can be performed on the output can be found in § 6.

4.1 General output

Regardless of the type of ex-core calculation, the Shift HDF5 output file contains general output information. Listings 4.1 and 4.2 gives an example of output data produced from most calculations. This output was produced from a problem with a single state point where the fission source was output. The `CORE` and `metadata` groups are always present. A `STATE_####` group exists for each state point specified in the VERA common input. Each state point group contains the appropriate fission source groups and datasets if the user-specified **output_fission_source** is set to *true*.

If the adjoint is being calculated, the user can also choose to have the adjoint flux, pin adjoint, and adjoint source written with the **output_adjoint** parameter. The adjoint flux and pin adjoint are written to the first state point, since they are currently only calculated for the first state in a cycle, as shown in Listing 4.1. The adjoint source is written to an independent file named *adjoint_source.h5*. The details of each adjoint flux data field in the Shift HDF5 file `adjoint` group are described in Table 7. The details of each fission source data field in the HDF5 file `fission_source` group are described in Table 8.

If the geometry output is turned on (**output_geometry** set to *true*), composition output files are created corresponding to each state point (**.shift_state_#_compositions.h5*), as well as a ray trace file (**.shift_state_1_ray trace.h5*). Note that a ray trace file is only created for the first state point. The user can specify the levels to perform the ray trace with the **ray trace_levels** parameter; if no levels are specified, the core midplane is used. These files can be used for advanced geometry visualization with Python as discussed in § 6.

Listing 4.1: General data in Shift HDF5 output file. File contents generated via `h5dump -n`.

```
group      /CORE
dataset    /CORE/axial_mesh
dataset    /CORE/baffle_gap_inner
dataset    /CORE/baffle_inner_radius
dataset    /CORE/baffle_outer_radius
dataset    /CORE/barrel_inner_radius
dataset    /CORE/barrel_outer_radius
dataset    /CORE/core_map
dataset    /CORE/core_sym
dataset    /CORE/liner_inner_radius
dataset    /CORE/liner_outer_radius
dataset    /CORE/pad_angles
dataset    /CORE/pad_arc
dataset    /CORE/pad_inner_radius
dataset    /CORE/pad_nonuniform_arcs
dataset    /CORE/pad_outer_radius
dataset    /CORE/vessel_inner_radius
dataset    /CORE/vessel_outer_radius
group      /STATE_0001
group      /STATE_0001/adjoint
dataset    /STATE_0001/adjoint/flux
dataset    /STATE_0001/adjoint/group_bounds_n
dataset    /STATE_0001/adjoint/mesh_g
dataset    /STATE_0001/adjoint/mesh_x
dataset    /STATE_0001/adjoint/mesh_y
dataset    /STATE_0001/adjoint/mesh_z
dataset    /STATE_0001/peak_memory
dataset    /STATE_0001/pin_adjoint
group      /STATE_0001/fission_source_U235
dataset    /STATE_0001/fission_source_U235/acceptance_condition
dataset    /STATE_0001/fission_source_U235/mesh_x
dataset    /STATE_0001/fission_source_U235/mesh_y
dataset    /STATE_0001/fission_source_U235/mesh_z
dataset    /STATE_0001/fission_source_U235/num_rejection_samples
dataset    /STATE_0001/fission_source_U235/pdf
dataset    /STATE_0001/fission_source_U235/strength
dataset    /STATE_0001/fission_source_U235/watt_a
dataset    /STATE_0001/fission_source_U235/watt_b
group      /STATE_0001/fission_source_U238
dataset    /STATE_0001/fission_source_U238/acceptance_condition
dataset    /STATE_0001/fission_source_U238/mesh_x
dataset    /STATE_0001/fission_source_U238/mesh_y
dataset    /STATE_0001/fission_source_U238/mesh_z
dataset    /STATE_0001/fission_source_U238/num_rejection_samples
dataset    /STATE_0001/fission_source_U238/pdf
dataset    /STATE_0001/fission_source_U238/strength
dataset    /STATE_0001/fission_source_U238/watt_a
dataset    /STATE_0001/fission_source_U238/watt_b
```

Listing 4.2: General data in Shift HDF5 output file (continued).

```
group      /metadata
dataset    /metadata/datetime
dataset    /metadata/scale_rev
dataset    /metadata/scale_version
dataset    /metadata/system
dataset    /veraout_version
```

Table 7: Adjoint flux data fields in the Shift HDF5 output file.

Data name	Type	Description
flux	array of doubles	energy-dependent adjoint flux [g][z][y][x]
group_bounds_n	array of doubles	energy group boundaries in decreasing order [eV]
mesh_g	array of integers	group indices
mesh_x	array of doubles	adjoint flux mesh edges along x-axis
mesh_y	array of doubles	adjoint flux mesh edges along y-axis
mesh_z	array of doubles	adjoint flux mesh edges along z-axis

Table 8: Fission source data fields in the HDF5 output file.

Data name	Type	Description
mesh_x	array of doubles	fission source mesh edges along x-axis
mesh_y	array of doubles	fission source mesh edges along y-axis
mesh_z	array of doubles	fission source mesh edges along z-axis
mesh_g†	array of doubles	fission source energy group bounds
pdf	array of doubles	fission source strength *[z][y][x] or †[g][z][y][x]
strength	scalar	fission source strength
watt_a*	scalar	Watt spectrum constant (MeV)
watt_b*	scalar	Watt spectrum constant (1/MeV)

†If the **fiss_src_spectrum** type is *mpact*
*If the **fiss_src_spectrum** type is *nuclide_watt* or *u235_watt*

Listing 4.3: Data in Shift HDF5 output file from an eigenvalue calculation. File contents generated via `h5dump -n`.

```
group      /STATE_0001
dataset    /STATE_0001/active_cycles
dataset    /STATE_0001/entropy
dataset    /STATE_0001/entropy_active_deviance
dataset    /STATE_0001/entropy_check
dataset    /STATE_0001/entropy_crossover_cycle
dataset    /STATE_0001/first_active_cycle
dataset    /STATE_0001/keff
dataset    /STATE_0001/keff_sigma
dataset    /STATE_0001/peak_memory
dataset    /STATE_0001/pin_powers
dataset    /STATE_0001/pin_powers_sigma
dataset    /STATE_0001/total_cycles
group      /STATE_0002
dataset    /STATE_0002/active_cycles
dataset    /STATE_0002/entropy
dataset    /STATE_0002/entropy_active_deviance
dataset    /STATE_0002/entropy_check
dataset    /STATE_0002/entropy_crossover_cycle
dataset    /STATE_0002/first_active_cycle
dataset    /STATE_0002/keff
dataset    /STATE_0002/keff_sigma
dataset    /STATE_0002/peak_memory
dataset    /STATE_0002/pin_powers
dataset    /STATE_0002/pin_powers_sigma
dataset    /STATE_0002/total_cycles
```

4.2 Eigenvalue calculation

From an eigenvalue calculation, results such as Shannon entropy, k_{eff}, active cycles, and pin powers, are located in the group for each state point in the Shift HDF5 output file. An example of the structure of eigenvalue-specific results in the output file is given in Listing 4.3 for a problem with multiple state points.

4.3 Simple vessel fluence calculation

From a simple vessel fluence calculation in which only the VERA common input file is used (i.e., there is no supplemental ex-core geometry file), the vessel flux is automatically tallied even if the user does not specify the tally parameters in the **SHIFT** block. The default vessel flux tally has a radial bin for each radius given in the **vessel** card, one axial bin, one theta bin, and one energy bin. The vessel flux tally result is automatically passed from Shift to VERA at each state point and is output to both the VERA and Shift HDF5 output files. An example of the Shift output for two state points is shown in Listing 4.4. The details of each tally data field in the HDF5 file `vessel_tally` group are described in Table 9. Note that there can also be photon boundaries specified for the vessel tally if running an ex-core calculation with Shift in (n, γ) mode.

Listing 4.4: Data in Shift HDF5 output file from a vessel fluence calculation. File output was generated via `h5dump -n`

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

Table 9: Vessel tally data fields in the Shift HDF5 output file. (All data fields are stored in the `/STATE_nnnn/tally/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 (eV)
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 neutron and photon 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]

4.4 Detector response calculation

For a detector response calculation, a typical use case is to use a Shift cell tally in the detector regions of interest. This can be done automatically with the detector and bioshield inputs from the VERA common input or with the supplemental ex-core file as mentioned previously. Table 10 gives the detail of a detector cell tally output in the Shift HDF5 output file. Again, note that photon energy bounds could also be present if the Shift calculation was run in the (n, γ) mode.

4.5 Visualization with VERAView

As mentioned previously, vessel flux calculated by Shift is reported back to MPACT for each state point to accumulate fluence. Fluence, $\phi(\vec{r}, E)$, is defined as the time integration of the flux, $\phi(\vec{r}, E, t)$, as given by

$$\phi(\vec{r}, E) = \int \phi(\vec{r}, E, t) \mathrm{d}t \,. \tag{4.1}$$

In practice, the fluence, $\phi_M^g(\vec{r})$, at a specific time step, assuming the flux is linearly varying with time, becomes

$$\phi_M^g(\vec{r}) = \sum_{m=1}^{M} \frac{\Delta t_m}{2} (\phi_{m-1}^g(\vec{r}) + \phi_m^g(\vec{r})) \,, \tag{4.2}$$

Table 10: Cell tally data fields in the Shift HDF5 output file. (All cell tallies are stored in the `/STATE_nnnn/tally` group.)

Data name	Type	Description
binned	array of doubles	energy-binned flux mean and variance [g][cell][multiplier][stat]
description	string	description of tally
group_bounds_n	array of doubles	energy bin boundaries in decreasing order (eV)
max_encountered_bins	int	max number of tally bins encountered by one particle history
mesh_stat	array of strings	*mean* and *var*
multiplier_descs	array of strings	description of each response tallied
multiplier_names	array of strings	scalars and responses tallied
normalization	double	normalization factor
num_histories	double	number of particle histories tallied
total	array of doubles	total neutron and photon flux mean and variance [cell][multiplier][stat]
union_cellids	array of ints	internal cell indices in this tally
union_lengths	array of ints	number of cell unions making up this tally
volumes	array of doubles	volumes of tally cells

where M is the current time step and g is the energy group. The flux, $\phi_m^g(\vec{r})$, in Eq. (4.2) comes from the energy-binned vessel tally for each state passed from Shift to MPACT. The end time, t_m, for a given state is calculated using the effective full-power days (EFPD) (aka exposure) stored in MPACT converted to seconds:

$$t_m = 86400 \times \text{EFPD}_m \,. \tag{4.3}$$

Specifically, MPACT performs numerical integration via the trapezoidal rule to accumulate the fluence. MPACT also propagates the associated MC uncertainty on the fluence, and this is output to the VERA HDF5 file as well.

The vessel fluence output to the MPACT HDF5 file can be visualized and analyzed in VERAView along with other core parameters such as pin powers. Figures 4.1–4.2 show examples of plotting the vessel flux, vessel fluence, and pin powers in VERAView. These two figures show the location of the maximum vessel flux and maximum vessel fluence. VERAView can also show the pin–resolved adjoint from the first state (which is currently the only state that performs the adjoint calculation) that is output to the Shift HDF5 file, as shown in Fig. 4.3.

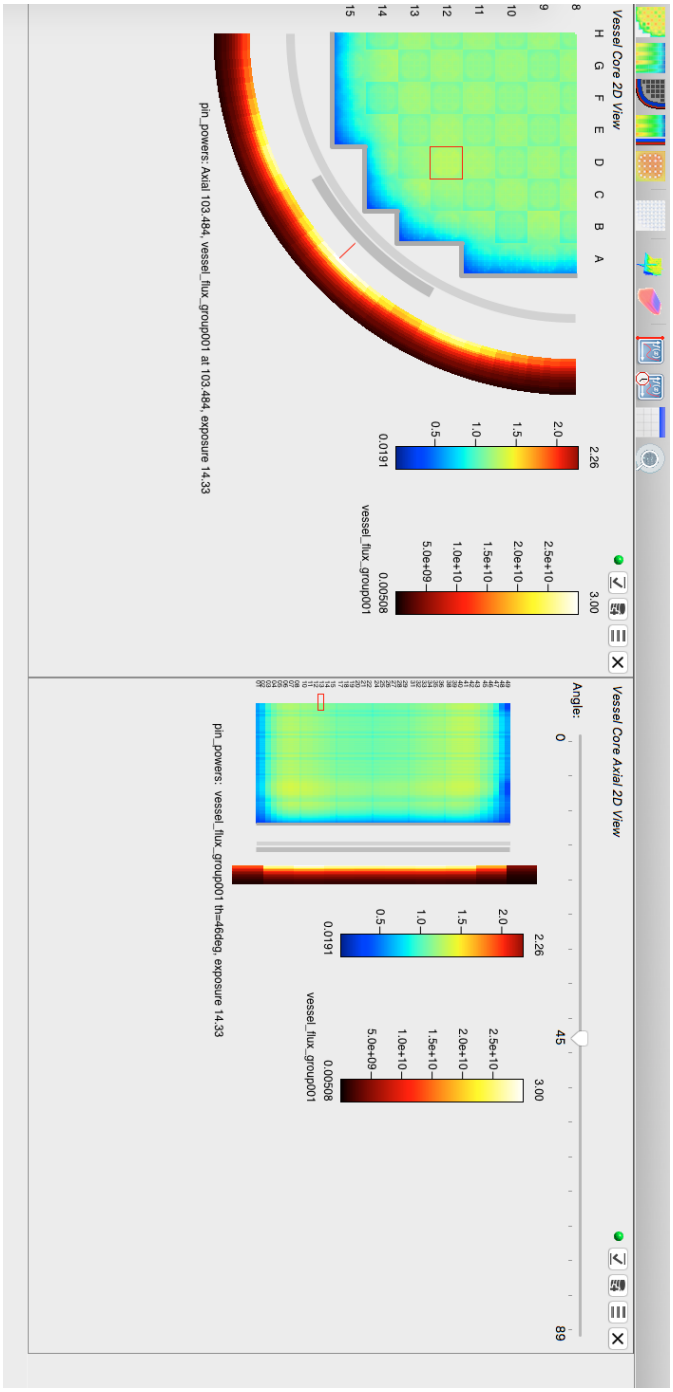


Figure 4.1: VERAView snapshot window of binned neutron flux for progression problem 9.

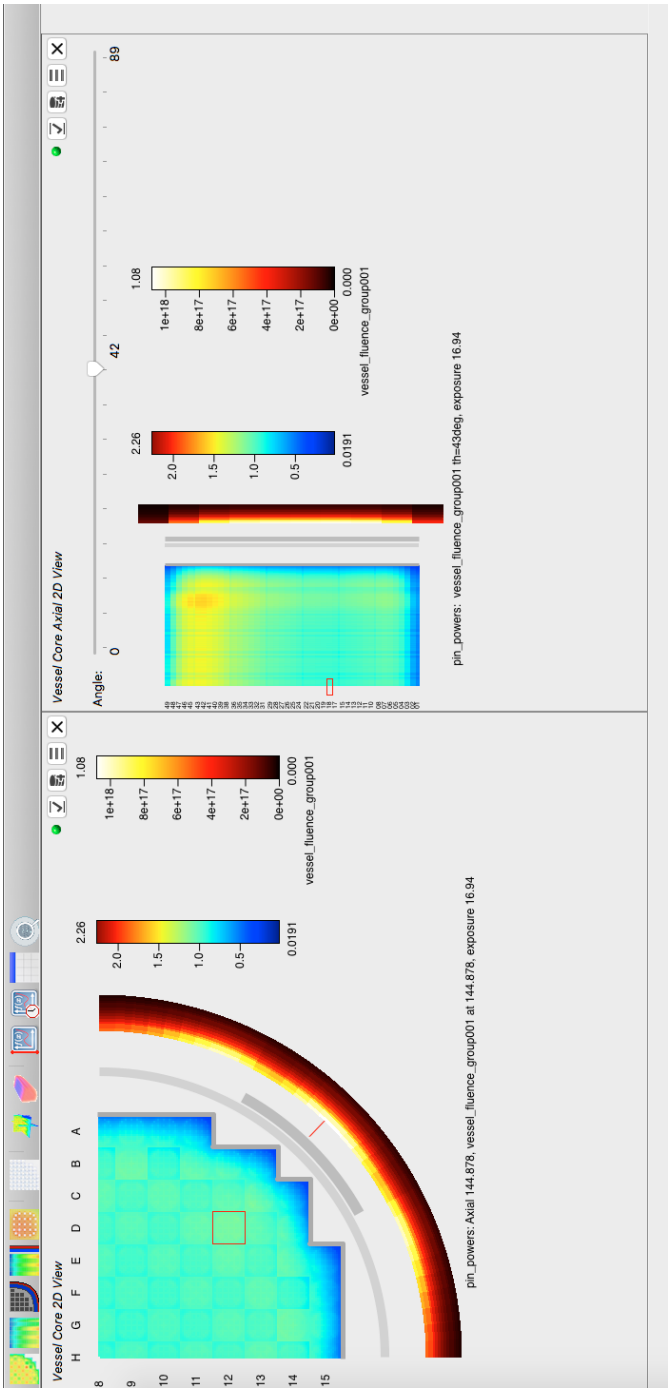


Figure 4.2: VERAView snapshot window of binned neutron fluence for progression problem 9.

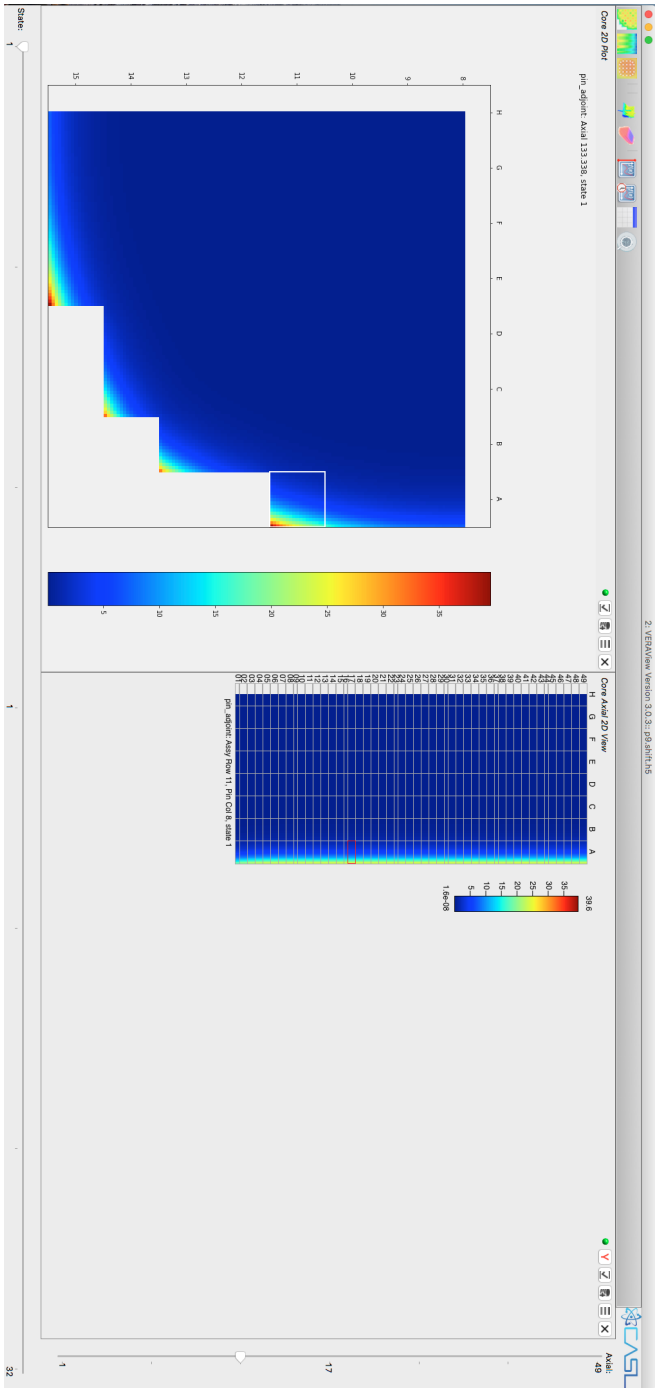


Figure 4.3: VERAView snapshot window of pin adjoint for progression problem 9.

5. Examples

This section presents inputs for several types of ex-core problems that can be run with VERA. Many variations and input parameters can be used for ex-core calculations; the following examples should be used as guidance for how a typical user might set up a problem.

5.1 Shift eigenvalue assembly example

This first example input is a modified version of CASL progression problem 3a [10]. The VERA common input for this problem is publicly available on Github [11]. It serves to show the input mechanics for running Shift in eigenvalue mode through VERA. Listing 5.1 shows the full VERA input for this example, and a ray trace through the geometry axial midplane is shown in Fig. 5.1. The **SHIFT** block is set up to perform an eigenvalue calculation with micro reaction rate tallies in the core. Note that to run this example problem, MPACT uses eight cores, whereas Shift can run on however many extra processors are requested.

Listing 5.1: VERA CASL progression problem 3a Shift eigenvalue input.

```
[CASEID]
  title 'CASL Progression Problem 3a - Single 17x17 Assembly - Public'

[STATE]
  power     0.0
  tinlet    600 K
  tfuel     600 K
  modden    0.743          ! g/cc
  sym       qtr
  search    boron
  feedback off
  excore_transport on      ! Needed so MPACT knows Shift will run

[CORE]
  size   1              ! one assembly
  rated  17.67 0.6823   ! MW, Mlbs/hr
  apitch 21.5
  height 406.337

  core_shape
    1

  assm_map
    ASSY
```

```
25
26    lower_plate ss  5.0 0.5   ! mat, thickness, vol frac
27    upper_plate ss  7.6 0.5
28
29    bc_rad reflecting
30
31  [ASSEMBLY]
32    title "Westinghouse 17x17"
33    npin 17
34    ppitch 1.26
35
36    fuel U31 10.257 94.5 / 3.1 u-234 0.026347
37
38    cell 1     0.4096 0.418 0.475 / U31 he zirc4
39    cell 3            0.561 0.602 / mod    zirc4       ! guide/instrument tube
40    cell 4            0.418 0.475 /     he zirc4       ! plenum
41    cell 5                  0.475 /        zirc4       ! end plug
42    cell 6                  0.475 /        mod         ! empty
43
44    lattice FUEL
45          3
46          1 1
47          1 1 1
48          3 1 1 3
49          1 1 1 1 1
50          1 1 1 1 1 3
51          3 1 1 3 1 1 1
52          1 1 1 1 1 1 1 1
53          1 1 1 1 1 1 1 1 1
54
55    lattice PLEN
56          3
57          4 4
58          4 4 4
59          3 4 4 3
60          4 4 4 4 4
61          4 4 4 4 4 3
62          3 4 4 3 4 4 4
63          4 4 4 4 4 4 4 4 4
64          4 4 4 4 4 4 4 4 4 4
65
66    lattice PLUG
67          3
68          5 5
69          5 5 5
70          3 5 5 3
71          5 5 5 5 5
72          5 5 5 5 5 3
73          3 5 5 3 5 5 5
74          5 5 5 5 5 5 5 5 5
75          5 5 5 5 5 5 5 5 5 5
76
77    lattice GAP
78          3
79          6 6
80          6 6 6
81          3 6 6 3
82          6 6 6 6 6
83          6 6 6 6 6 3
84          3 6 6 3 6 6 6
```

```
85        6 6 6 6 6 6 6 6
86        6 6 6 6 6 6 6 6 6
87
88    axial ASSY   6.053
89          GAP   10.281
90          PLUG  11.951
91          FUEL 377.711
92          PLEN 393.711
93          PLUG 395.381
94          GAP  397.51
95
96    grid END inc    3.866 1017 / loss=0.9070 ! grid height, mass, loss coef
97    grid MID zirc4  3.810 875  / loss=0.9065 ! grid height, mass, loss coef
98
99    grid_axial
100       END  13.884
101       MID  75.2
102       MID 127.4
103       MID 179.6
104       MID 231.8
105       MID 284.0
106       MID 336.2
107       END 388.2
108
109   lower_nozzle  ss 6.053 6250.0  ! mat, height, mass (g)
110   upper_nozzle  ss 8.827 6250.0  ! mat, height, mass (g)
111
112 [EDITS]
113   axial_edit_bounds
114       11.951
115       15.817
116       24.028
117       32.239
118       40.45
119       48.662
120       56.873
121       65.084
122       73.295
123       77.105
124       85.17
125       93.235
126       101.3
127       109.365
128       117.43
129       125.495
130       129.305
131       137.37
132       145.435
133       153.5
134       161.565
135       169.63
136       177.695
137       181.505
138       189.57
139       197.635
140       205.7
141       213.765
142       221.83
143       229.895
144       233.705
```

```
      241.77
      249.835
      257.9
      265.965
      274.03
      282.095
      285.905
      293.97
      302.035
      310.1
      318.165
      326.23
      334.295
      338.105
      346.0262
      353.9474
      361.8686
      369.7898
      377.711

[MPACT]
  num_space       8
  num_angle       1
  num_threads     1
  ray_spacing         0.08
  shield_ray_spacing 0.08
  azimuthals_octant 4
  shield_azimuthals_octant 4
  polars_octant     1
  ! This library is only for testing
  xs_filename       mpact8g_70s_v4.0m0_02232015.fmt
  res_up_scatter    false

[SHIFT]
  Np                    1000000
  num_cycles            1000
  num_inactive_cycles   300
  output_fission_source true
  output_geometry       true
  global_log            debug
  problem_mode          eigenvalue
  ! tally db
  n_bounds              2.0e7 1.0e6 1.0e4 1.0e2
  ! nuclides to tally reactions for in ZAID format
  micro_zaids           92238 92235 94239
  ! ENDF MT values
  micro_rxns            1 18 27
```

5.2 Shift forward minicore example

This next example is a minicore version of CASL progression problem 9 [10], with Shift running in forward mode to calculate vessel fluence. The number of depletion states has been reduced for this example from the original problem.

Listing 5.2 shows the explicit VERA input with Shift set to run in forward mode. In the **SHIFT** block, a

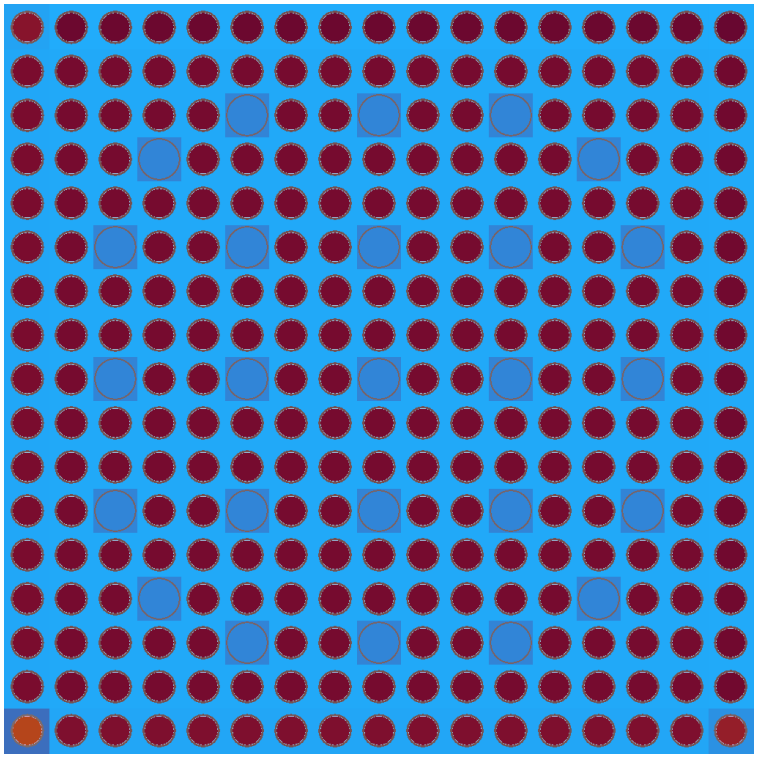


Figure 5.1: Ray trace of the progression problem 3a geometry at the axial midplane.

vessel fluence tally is set up in the barrel, pads, liner, and vessel. The neutron bounds specified are used in Shift to tally the energy-binned flux in these regions. Even though this problem has quarter core symmetry, **num_theta** refers to the number of bins over the full 360°. This number of angular bins must be divisible by four for quarter core symmetry problems. Note that although feedback is on, the temperatures and densities are not set to be transferred from MPACT to Shift because unique pins are turned off. This problem is set up to run MPACT on 32 cores, whereas Shift can use however many cores the user designates. A ray trace of the geometry midplane created by Shift is shown in Fig. 5.2.

Listing 5.2: VERA CASL progression problem 9 minicore input.

```
[CASEID]
  title 'Mini Core - using 7x7 assemblies'

[STATE]
  power     100.0
  tinlet    565 K
  tfuel     600 K
  modden    0.743          ! g/cc
  sym       qtr
  pressure  2250           ! psia
  feedback  on
  excore_transport on
  search    boron
  kcrit     0.89500
  xenon     equil
```

```
  rodbank  A 230
           B 230
           C 230
           D 215
  deplete EFPD 0 30 60

[CORE]
  size    9
  ! Note that rated power is lower than normal due to smaller core
  rated  88.3333  5.0865 ! MW, Mlbs/hr
  apitch 8.9
  height 245.9380

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

  assm_map
    1
    2 1
    1 2 1
    2 1 2 3
    3 3 3

  insert_map
    -
   04 TP
    - 04  -
   04 TP 04  -
    - TP TP

  crd_map
    1
    - -
    1 - 1
    - - - 1
    1 - -

  crd_bank
    D
    - -
    A - D
    - - - C
    B - -

 det_map
       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 -

  baffle ss 0.19 1.18

  vessel mod 50.0  ! barrel IR (cm)
         ss  52.0  ! barrel OR (cm)
         mod 60.25 ! vessel liner IR (cm)
         ss  60.5  ! vessel liner OR / vessel IR (cm)
         cs  70.5  ! vessel OR (cm)
  ! neutron pad ID,OD arc length (degrees), and angular positions (degrees)
  pad ss 52.5 55.0 32 45 135 225 315

  xlabel J H G F E D C B A
  ylabel 1 2 3 4 5 6 7 8 9

  lower_plate ss  5.0 0.5   ! mat, thickness, vol frac
  upper_plate ss  7.6 0.5

[ASSEMBLY]
  title "Mini 7x7"
  npin 7
  ppitch 1.26

  fuel U21 10.257 94.5 / 2.110
  fuel U26 10.257 94.5 / 2.619
  fuel U31 10.257 94.5 / 3.100

  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
  cell 5            0.418 0.475 /     he zirc4       ! plenum
  cell 6                  0.475 /        zirc4       ! plug
  cell 7                  0.475 /        mod         ! empty

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

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

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

  lattice PLEN
       4
       5 5
       5 5 4
       5 5 5 5
```

```
  lattice PLUG
       4
       6 6
       6 6 4
       6 6 6 6

  lattice GAP
       4
       7 7
       7 7 4
       7 7 7 7

  axial  1  10.281 PLUG 11.951 LAT21 221.111 PLEN 237.111
  axial  2  10.281 PLUG 11.951 LAT26 221.111 PLEN 237.111
  axial  3  10.281 PLUG 11.951 LAT31 221.111 PLEN 237.111

  grid END inc   3.866 172.43 / loss=0.9070
  grid MID zirc4 3.810 148.36 / loss=0.9065

  grid_axial
      END  13.884
      MID  75.2
      MID 127.4
      MID 179.6
      END 235.1

  lower_nozzle  ss 10.281 1059.69  ! mat, height, mass (g)
  upper_nozzle  ss 8.827  1059.69  ! mat, height, mass (g)

[INSERT]
  title "Pyrex"
  npin 7

  cell 1  0.214 0.231 0.241 0.427 0.437 0.484 / he ss he pyrex-vera he ss
  cell 2                          0.437 0.484 /                     he ss
  cell 3                                0.484 /                        ss
  cell 4                                0.538 /                        ss

  rodmap  PY04
     -
     - -
     - - 1
     - - - -

  rodmap  PLEN04
     -
     - -
     - - 2
     - - - -


  rodmap  PLUG04
     -
     - -
     - - 3
     - - - -

  rodmap  TPLUG
     -
     - -
```

```
     - - 4
     - - - -

  axial 04     13.221
        PLUG04 15.761
          PY04 219.841
        PLEN04 226.71
         TPLUG 242.04

  axial TP     226.71
        TPLUG  242.04

[CONTROL]
  title "B4C with AIC tips"
  npin    7
  stroke  208.525 230     ! approx for 1.5875 step sizes and 230 max stroke

  cell 1  0.382 0.386 0.484 / aic he ss
  cell 2  0.373 0.386 0.484 / b4c he ss
  cell 3        0.386 0.484 /     he ss !plenum
  cell 4              0.484 /        ss !plug

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

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

  rodmap PLEN
     -
     - -
     - - 3
     - - - -

  rodmap PLUG
     -
     - -
     - - 4
     - - - -

  axial 1     16.2875  ! adusted to position the rod at 257.9 cm
        PLUG  18.1875
         AIC 76.21185
         B4C 224.1740
        PLEN 230.2848

[DETECTOR]
  title "Incore instrument thimble"
  npin 7

  cell 1  0.258 0.382 / he ss

  rodmap  LAT
     1
```

```
      - -
      - - -
      - - - -

  axial 1  0.0 LAT 240.91

[EDITS]
  axial_edit_bounds
        11.9510
        15.8170
        24.0280
        32.2390
        40.4500
        48.6620
        56.8730
        65.0840
        73.2950
        77.1050
        85.1700
        93.2350
       101.3000
       109.3650
       117.4300
       125.4950
       129.3050
       137.3700
       145.4350
       153.5000
       161.5650
       169.6300
       177.6950
       181.5050
       189.4262
       197.3474
       205.2686
       213.1898
       221.1110

[MPACT]
  num_space           32
  num_angle           1
  num_threads         1
  ray_spacing         0.08
  shield_ray_spacing  0.08
  azimuthals_octant   16
  polars_octant       2
  res_up_scatter      false
  dep_substep         1
  k_tolerance         1e-5
  flux_tolerance      5e-5

[SHIFT]
  problem_mode           forward
  Np                     1e10
  output_geometry        true
  output_fission_source  true
  create_unique_pins     false
  transfer               fiss_src
  ! tally_db
  n_bounds    2.0e7 1.0e6
```

```
  num_theta    64
  num_axial    10
  radial_mesh  0 49.99999 50.25 50.5 50.75 51 51.25 51.5 51.75 52.00001
               60.249999 60.50001 61.5 62.5 63.5 64.5 65.5 66.5 67.5
               68.5 69.5 70.50001
```

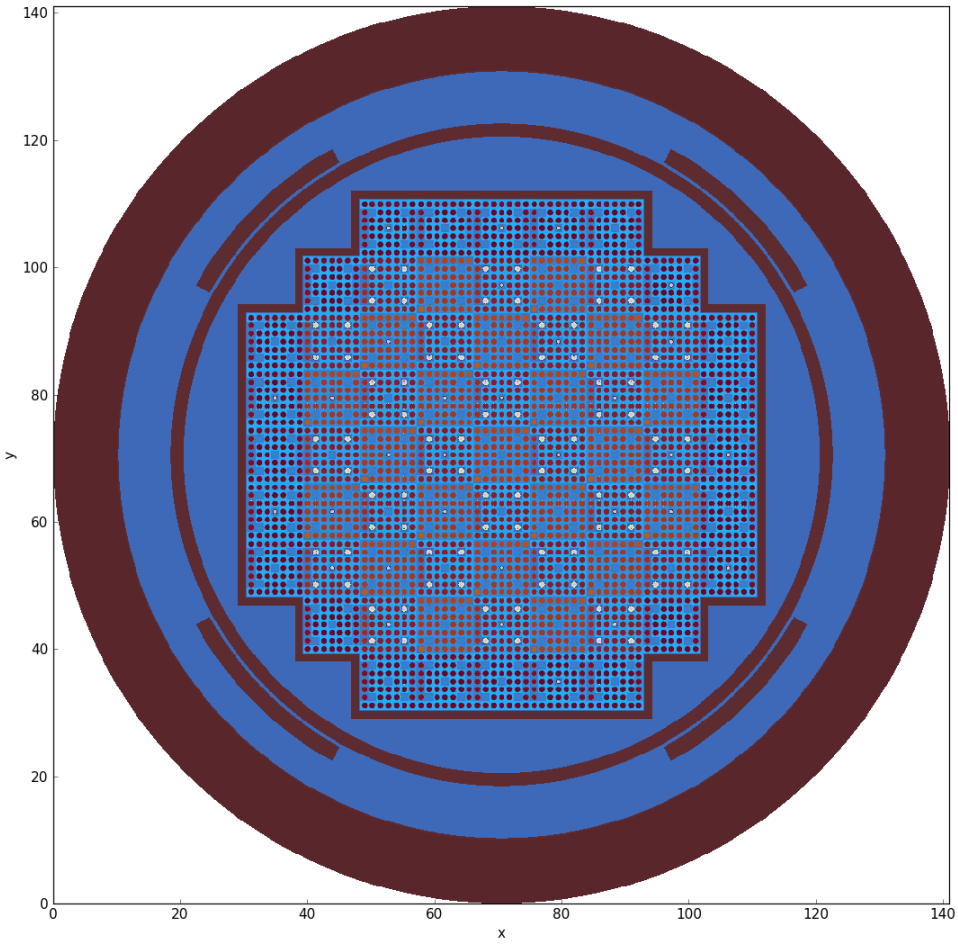


Figure 5.2: Ray trace of progression problem 9 geometry at axial midplane.

5.3 CADIS vessel fluence minicore example

This example is the same problem as the previous example, but Shift is set up to run in CADIS mode. Listing 5.3 only shows the **SHIFT** block since those are the parameters that change relative to the previous example. Several parameters should be noted. First, the decomposition specified by the number of blocks in i and j gives a total number of processors for the Shift calculation of 25. The **xs_library** parameter specifies the multigroup library used for the adjoint deterministic transport calculation to produce variance

reduction parameters. The **num_groups** and **new_grp_bounds** parameters indicate the energy resolution that the adjoint deterministic calculation uses. In this example, the multigroup library has 56 groups but is collapsed to 8 groups for efficiency, less memory use, and to target higher energy regimes for ex-core regions. The deterministic calculation is run with cell homogenized cross sections in each pin cell with flux–weighting when the **cell_homogenize** parameter is *true*. Otherwise, the cross sections are volume weighted in each computational cell, which is less time consuming. The **verbosity** parameter controls how much information from the deterministic adjoint solver is printed to the screen at each iteration.

Listing 5.3: VERA CASL progression problem 9 minicore CADIS input.

```
[SHIFT]
  problem_mode          cadis
  Np                    1e9
  output_geometry       true
  output_fission_source true
  create_unique_pins    false
  ! hybrid
  adjoint          true
  num_blocks_i     5
  num_blocks_j     5
  output_adjoint   true
  xs_library       v7-56
  num_groups       8
  new_grp_bounds   6.0653E+06 3.6788E+06 2.3457E+06 1.6530E+06
                   8.2085E+05 2.4176E+04 1.0130E+02 1.0000E-05
  cell_homogenize true
  verbosity        high
  ! tally_db
  n_bounds     2.0e7 1.0e6
  num_theta    64
  num_axial    10
  radial_mesh 0 49.99999 50.25 50.5 50.75 51 51.25 51.5 51.75 52.00001
              60.249999 60.50001 61.5 62.5 63.5 64.5 65.5 66.5 67.5
              68.5 69.5 70.50001
```

5.4 Automated detectors example

This example is based on the previous two examples and demonstrates use of the VERA input to automatically generate a supplemental ex-core file with detectors and bioshield. Listing 5.4 shows the relevant sections of the **CORE** and **SHIFT** blocks for this example. A ray trace through the core midplane of this geometry is shown in Fig. 5.3. The supplemental ex-core input file is created when the user runs the `react2xml` script with a VERA input file. This supplemental file takes the base of the VERA input filename with an extension of `.omn`. The user must then run `excore2xml` with the `*.omn` file to convert it to xml format (as is the case with any calculation using a supplemental ex-core file).

The user should note that this example does not specify the **num_blocks_i** and **num_blocks_j** parameters, so the default values of ten for each are used; therefore Shift would run on 100 processors. Also, if running this example through `verarun`, as described in § 3.1, the `excore2xml` script is run automatically and an error is thrown if the **excore_filename** parameter does not specify the correct filename.

Listing 5.4: VERA CASL progression problem 9 minicore input with automated excore detectors.

```
[CORE]
  vessel mod 50.0   ! barrel IR (cm)
         ss  52.0   ! barrel OR (cm)
         mod 60.25  ! vessel liner IR (cm)
         ss  60.5   ! vessel liner OR / vessel IR (cm)
         cs  70.5   ! vessel OR (cm)
  ! neutron pad ID,OD arc length (degrees), and angular positions (degrees)
  pad ss 52.5 55.0 32 45 135 225 315

  bioshield  void      90.0
             ss        91.0
             concrete 110.0

!      ID    type     radii    / mats    / heights   / response_type well_type
! ------------------------------------------------------------------------------
  det  PWR  power   1.5  2.5 / void cs / 52.4 52.4 / u235  wedge
  det  SRC  source  2.0  3.0 / void cs / 48.0      / b10   none

!               ID,   radius degree elevation
! ------------------------------------------
  det_locations  PWR   92.5    330    70.0
                 PWR   92.5    300    70.0
                 SRC   80.0      0   100.0


[SHIFT]
  problem_mode          cadis
  Np                    1e10
  output_geometry       true
  output_fission_source true
  excore_filename       prob9_mini.excore.xml
  core_translate        -70.5 -70.5 0.0
  vera_pressure_vessel  true
```

Several parameters shown in Listing 5.4 should be noted.

- The bioshield and detector materials specified after the **bioshield** and **det** keywords must be defined in the template file located in the VERA source code (`scripts/Init/omnibus_template.txt`) and given in Appendix A.
- For the **det** parameter, the **response_type** options are `b10` or `u235`, and the **well_type** options are `none` or `wedge` (as shown in Table 5).
- This example defines two different detector types with IDs of *PWR* and *SRC*, where the *PWR* detector uses the wedge-shaped well.
- The placement of the user-defined detector types in the ex-core region is given by the **det_locations** parameter. Detectors must be located outside of the vessel. The angle specified is specified as clockwise from the positive x-axis. For quarter core problems, this needs to be noted since Shift plots and models the geometry as the northeast quadrant.
- The user needs to ensure that the **excore_filename** parameter in the **SHIFT** block matches the actual name of the supplemental ex-core file after it is converted to XML format.

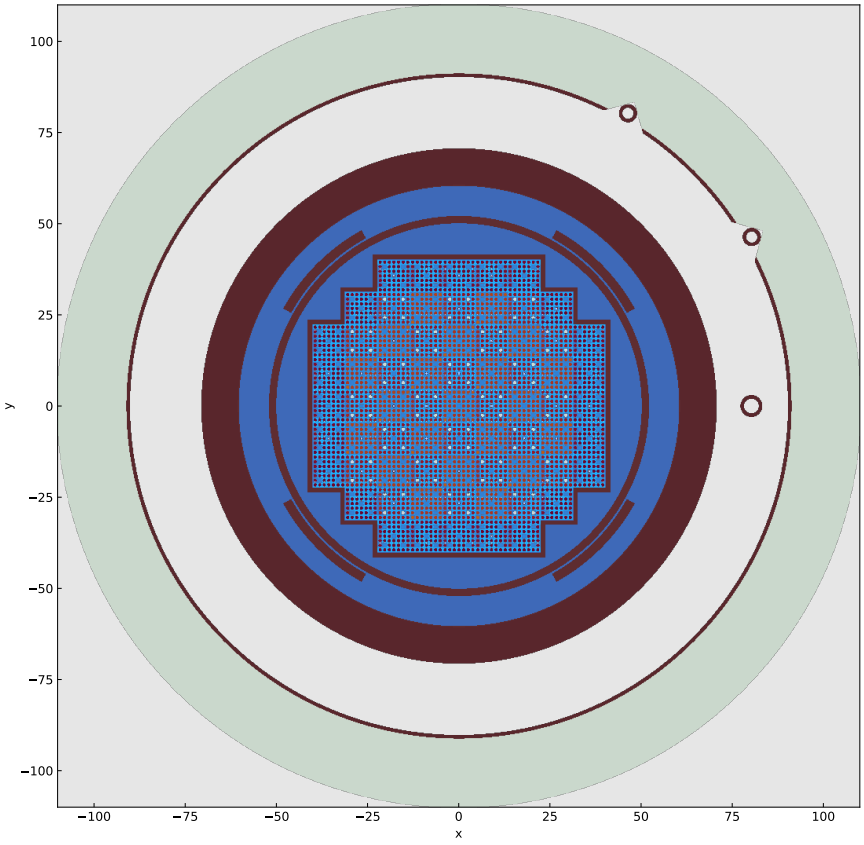


Figure 5.3: Ray trace of example geometry using the automated generation of the supplemental ex-core file with bioshield and detectors.

- In most cases, the user should have **vera_pressure_vessel** set to *true*. If this parameter is not set, the VERA geometry extends to the outer radius of the barrel and the supplemental ex-core file defines the geometry beyond it. When using the **bioshield** parameter, most users should use the full vessel geometry defined in the VERA input file by the **vessel** parameter.

In addition to the ex-core geometry, the generated supplemental ex-core file also includes a tally for the innermost volume of each specified detector. For this example, each *PWR* detector has two tallies, one for each axial extent. When running in CADIS mode, this can require the user to inspect the `*.omn` file to determine the name of the tally to optimize and add it to the **hybrid_tally_names** parameter. To reduce the burden on the user, there are several ways to automatically populate **hybrid_tally_names** in the XML version of the VERA input:

1. If no **hybrid_tally_names** parameter is given in the **SHIFT** block, the parameter will automatically be populated with all ex-core detector tallies.
2. If the user gives the name of a detector ID, such as *PWR* or *SRC* in Listing 5.4, for the **hy-**

brid_tally_names parameter, the generated VERA XML input will replace the detector ID with the appropriate tally names for all instances of that detector ID.

3. If **hybrid_tally_names** contains a tally name(s) that does not match a detector ID, the parameter will be left unchanged in the VERA XML file and the user must ensure that it matches the name of a tally defined in the supplemental ex-core file.

This capability is under active development, and additional features will be available in future releases of VERA.

5.5 Ex-core file detector example

Building on § 5.4, this example demonstrates the ability to provide a manually defined supplemental ex-core input file defining the external geometry and tallies for the Shift calculation. The example input shown here is a modified mini version of CASL progression problem 5a [10]. All of the geometry outside of the core barrel is manually defined in the supplemental ex-core input file. Figure 5.4 shows a ray trace of this example geometry at the axial midplane.

Listing 5.5 shows the explicit VERA input with Shift set to run in CADIS mode. In the **SHIFT** block, the name of the ex-core file is specified. Also, since the ex-core geometry is defined with the origin at the center of the core, the VERA core must be translated by the radius of the outer barrel radius (additional details in Appendix C). Note that if the parameter **vera_pressure_vessel** was set to *true*, the translation would correspond to the outermost vessel radius. The final item to note from this input block is the definition of the boundary mesh. As noted in the comments in the example, this boundary mesh serves to cut the full geometry, including the ex-core part, to the northeast corner with reflecting boundaries on the negative x and negative y boundaries (to simulate a quarter core). Also, the coordinates of the boundary mesh box are based on the ex-core geometry origin and not the VERA origin.

Listing 5.6 shows the explicit supplemental ex-core input defining the ex-core geometry, tallies, and compositions for the Shift calculation. For most of the entries in this ex-core file, details can be found in the Shift manual [12]. One entry is specific for the geometry section of the ex-core file: the core universe. This indicates the core geometry is taken from VERA and is inserted with the *fill* parameter in the hole universe that is defined in the ex-core file. The composition materials must also be defined to correspond to the materials in the ex-core geometry; this example shows dummy materials. Currently, the user cannot use any of the materials that were defined in the VERA input for ex-core geometry cells. Finally, the tally block shows a cell tally set up in the detector cell for the flux and fission reaction rate. Since there were no neutron or photon bins specified, this tally will give the user the total (summation of neutron plus photon) flux and fission from neutrons only. This total flux is not useful for analysis so the user should definitely include neutron and photon bins for a flux tally if it applies to their problem of interest.

Listing 5.5: VERA CASL progression problem 5 minicore VERA input using ex-core file.

```
1  [CASEID]
2    title 'Mini Core - using 7x7 assemblies'
3
4  [STATE]
5    power    0.0
6    tinlet   600 K
```

```
  tfuel     600 K
  boron     1300            ! ppmB
  modden    0.743           ! g/cc
  sym       qtr
  feedback off
  excore_transport on
  rodbank   A 230
            B 230
            C 230
            D 167

[CORE]
  size     9
  ! Note that rated power is lower than normal because high peaking factor
  ! in small core
  rated  88.3333  5.0865 ! MW, Mlbs/hr
  apitch 8.9
  height 245.9380

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

  assm_map
    1
    2 1
    1 2 1
    2 1 2 3
    3 3 3

  insert_map
    -
   04 TP
    - 04  -
   04 TP 04  -
    - TP TP

  crd_map
    1
    - -
    1 - 1
    - - - 1
    1 - -

  crd_bank
    D
    - -
    A - D
    - - - C
    B - -

 det_map
       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 -

  baffle ss 0.19 1.0
  vessel  mod 48.0         ! barrel IR (cm)
          ss  50.0         ! barrel OR (cm)
          mod 65.0         ! vessel liner IR (cm)
          ss  66.0         ! vessel liner OR / vessel IR (cm)
          ss  70.0         ! vessel OR (cm)
! neutron pad ID,OD arc lenth (degrees), and angular positions (degrees)
  pad ss  62.0 63.5 32 45 135 225 315

  xlabel J H G F E D C B A
  ylabel 1 2 3 4 5 6 7 8 9

[ASSEMBLY]
  title "Mini 7x7"
  npin 7
  ppitch 1.26

  fuel U21 10.257 94.5 / 2.110
  fuel U26 10.257 94.5 / 2.619
  fuel U31 10.257 94.5 / 3.100

  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
  cell 5            0.418 0.475 /     he zirc4       ! plenum
  cell 6                  0.475 /        zirc4       ! plug
  cell 7                  0.475 /        mod         ! empty

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

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

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

  lattice PLEN
       4
       5 5
       5 5 4
```

```
      5 5 5 5

  lattice PLUG
      4
      6 6
      6 6 4
      6 6 6 6

  lattice GAP
      4
      7 7
      7 7 4
      7 7 7 7

  axial  1  10.281 PLUG 11.951 LAT21 221.111 PLEN 237.111
  axial  2  10.281 PLUG 11.951 LAT26 221.111 PLEN 237.111
  axial  3  10.281 PLUG 11.951 LAT31 221.111 PLEN 237.111

  grid END inc   3.866 172.43 / loss=0.9070
  grid MID zirc4 3.810 148.36 / loss=0.9065

  grid_axial
     END  13.884
     MID  75.2
     MID 127.4
     MID 179.6
     END 235.1

  lower_nozzle  ss 10.281 1059.69  ! mat, height, mass (g)
  upper_nozzle  ss 8.827 1059.69  ! mat, height, mass (g)

[INSERT]
  title "Pyrex"
  npin 7

  cell 1 0.214 0.231 0.241 0.427 0.437 0.484 / he ss he pyrex-vera he ss
  cell 2                         0.437 0.484 / he ss  ! plenum
  cell 3                               0.484 / ss     ! plug
  cell 4                               0.538 / ss     ! thimble plug

  rodmap  PY04
     -
     - -
     - - 1
     - - - -

  rodmap  PLEN04
     -
     - -
     - - 2
     - - - -


  rodmap  PLUG04
     -
     - -
     - - 3
     - - - -

  rodmap  TPLUG
```

```
     -
     - -
     - - 4
     - - - -

  axial 04     13.221
        PLUG04 15.761
          PY04 219.841
        PLEN04 226.71
         TPLUG 242.04

  axial TP     226.71
         TPLUG 242.04

[CONTROL]
  title "B4C with AIC tips"
  npin 7
  stroke  208.525 230     ! approx for 1.5875 step sizes and 230 max stroke

  cell 1  0.382 0.386 0.484 / aic he ss
  cell 2  0.373 0.386 0.484 / b4c he ss
  cell 3        0.386 0.484 /     he ss !plenum
  cell 4              0.484 /        ss !plug

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

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

  rodmap PLEN
     -
     - -
     - - 3
     - - - -

  rodmap PLUG
     -
     - -
     - - 4
     - - - -

  axial 1     16.2875  ! adusted to position the rod at 257.9 cm
        PLUG  18.1875
         AIC 76.21185
         B4C 224.1740
        PLEN 230.2848

[DETECTOR]
  title "Incore instrument thimble"
  npin 7

  cell 1  0.258 0.382 / he ss
```

```
  rodmap  LAT
     1
     - -
     - - -
     - - - -

  axial 1  0.0 LAT 240.91

[EDITS]
  axial_edit_bounds
        11.9510
        15.8170
        24.0280
        32.2390
        40.4500
        48.6620
        56.8730
        65.0840
        73.2950
        77.1050
        85.1700
        93.2350
       101.3000
       109.3650
       117.4300
       125.4950
       129.3050
       137.3700
       145.4350
       153.5000
       161.5650
       169.6300
       177.6950
       181.5050
       189.4262
       197.3474
       205.2686
       213.1898
       221.1110

[MPACT]
  num_space           16
  num_angle           1
  num_threads         1
  ray_spacing         0.08
  shield_ray_spacing  0.08
  azimuthals_octant   4
  shield_azimuthals_octant 4
  polars_octant       1
  ! This library is just for testing
  xs_filename         mpact8g_70s_v4.0m0_02232015.fmt
  dep_substep         1
  axial_mesh   11.9510
               3.866
               8.211
               8.211
               8.211
               8.212
               8.211
               8.211
```

```
                8.211
                3.81
                8.065
                8.065
                8.065
                8.065
                8.065
                8.065
                3.81
                8.065
                8.065
                8.065
                8.065
                8.065
                8.065
                3.81
                7.9212
                7.9212
                7.9212
                7.9212
                7.9212
               12.056
                3.866
                8.905

[SHIFT]
  excore_filename       p5_mini_det.excore.xml
  ! Translation corresponds to the origin of the ex-core file geometry
  core_translate           -50.0 -50.0 0.0
  problem_mode             forward
  Np                       1e10
  output_geometry          true
  output_fission_source    true
  transport                ce
  do_transport             true
  create_unique_pins       false
  vera_pressure_vessel     false
  hybrid_tally_names       det_tally
  hybrid_tally_multipliers flux
  ! This boundary mesh cuts the full ex-core geometry to the NE quarter
  bc_bnd_mesh reflect vacuum reflect vacuum vacuum vacuum
  ! The origin of this boundary mesh is based on the ex-core file geometry
  x_bnd_mesh 0.0  85.0
  y_bnd_mesh 0.0  85.0
  z_bnd_mesh 0.0 260.0
```

Listing 5.6: VERA CASL progression problem 5 minicore ex-core file input.

```
! Mini Core with barrel and vessel and ex-core detectors
[GEOMETRY]
global "outer"

! This universe is where the core proto will be inserted
[UNIVERSE=core reactor]

[UNIVERSE=general outer]
boundary "bigbox"

!!!!!! SHAPES
[UNIVERSE][SHAPE=cuboid bigbox]
faces -85 85 -85 85 0 260

! Barrel
[UNIVERSE][SHAPE=cyl outer_barrel]
radius 50
extents 0 260
origin 0 0 0
axis z

! Vessel liner
[UNIVERSE][SHAPE=cyl inner_vessel_liner]
radius 65
extents 0 260
origin 0 0 0
axis z

[UNIVERSE][SHAPE=cyl outer_vessel_liner]
radius 66
extents 0 260
origin 0 0 0
axis z

! Vessel
[UNIVERSE][SHAPE=cyl outer_vessel]
radius 70
extents 0 260
origin 0 0 0
axis z

! Detector outside the vessel (rotate first then translate)
[UNIVERSE][SHAPE=cuboid det1]
faces -2 2 -2 2 -2 2
rotate 0.7071067811865476 -0.7071067811865476 0
       0.707106781186547   0.707106781186547  0
       0                   0                  1
translate 52.6794551983978 52.6794551983978 100

! The barrel has an outer radius of 50 cm
! The bottom left corner of the core is at 0,0,0 so it must be translated
[UNIVERSE][HOLE thecore]
fill reactor
translate -50.0 -50.0 0.0

!!!!!! CELLS
[UNIVERSE][CELL detector1]
comp dummy1
shapes -det1
```

```
60
61  [UNIVERSE][CELL barrel]
62  comp dummy2
63  shapes +thecore -outer_barrel
64
65  [UNIVERSE][CELL mod_space]
66  comp dummy3
67  shapes +outer_barrel -inner_vessel_liner
68
69  [UNIVERSE][CELL vessel_liner]
70  comp dummy2
71  shapes +inner_vessel_liner -outer_vessel_liner
72
73  [UNIVERSE][CELL vessel]
74  comp dummy2
75  shapes +outer_vessel_liner -outer_vessel
76
77  ! Boundary box cell
78  [UNIVERSE][CELL bbox]
79  comp dummy4
80  shapes -bigbox +outer_vessel +det1
81
82  !################################################
83
84  [COMP]
85
86  [COMP][MATERIAL dummy1]
87  matid 0
88  temperature 293
89  zaid 92235
90  nd 1.0
91
92  [COMP][MATERIAL dummy2]
93  matid 1
94  temperature 293
95  zaid 6012
96  nd 1.0
97
98  [COMP][MATERIAL dummy3]
99  matid 2
100 temperature 293
101 zaid 8016
102 nd 1.0
103
104 [COMP][MATERIAL dummy4]
105 matid 3
106 temperature 293
107 zaid 1001
108 nd 1.0
109
110 !################################################
111
112 [TALLY]
113
114 [TALLY][CELL det_tally]
115 reactions flux fission
116 cells detector1
```

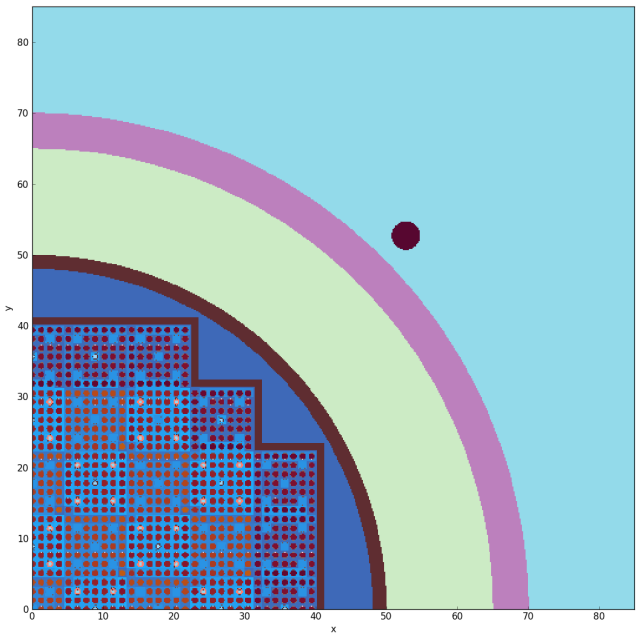


Figure 5.4: Ray trace of example at axial midplane with the ex-core regions defined manually with the supplemental ex-core input file.

6. Advanced and Experimental Features

This section details the advanced postprocessing capabilities using Python with output from an ex-core calculation and the experimental core temperature homogenization options in Shift.

6.1 Core temperature homogenization

An experimental capability in VERA using Shift is the capability to investigate the effects of homogenizing temperature and coolant density regions of the core on ex-core quantities. These homogenization methods apply an average temperature in the fuel, clad, and coolant regions of a pincell depending on the method used. The input parameters governing these methods are given in Table 6. Four core temperature and density homogenization methods exist in VERAShift:

1. *Assembly averaged*: This method applies a volume-averaged temperature and density over each assembly in the core (shown in Fig. 6.1).

2. *Assembly ring averaged*: This method applies a volume-averaged temperature and coolant density in full assemblies encompassed by user-specified radial rings around the center of the core (shown in Fig. 6.2a).

3. *Pin ring averaged*: This method applies a volume-averaged temperature and coolant density to pins encompassed by user-specified radial rings around the center of the core (shown in Fig. 6.2b).

4. *Explicit peripheral pins*: This method must be used in conjunction with methods (1) or (2). It uses explicit temperature and coolant density in each pincell outside of a user-specified radial ring from the center of the core (shown in Fig. 6.3).

Full details about this capability and the investigation of its effect on vessel fluence can be found in Stocking [2, 13].

To use these experimental methods for ex-core calculations with VERA, please contact Tara Pandya (pandyatm@ornl.gov) at ORNL.

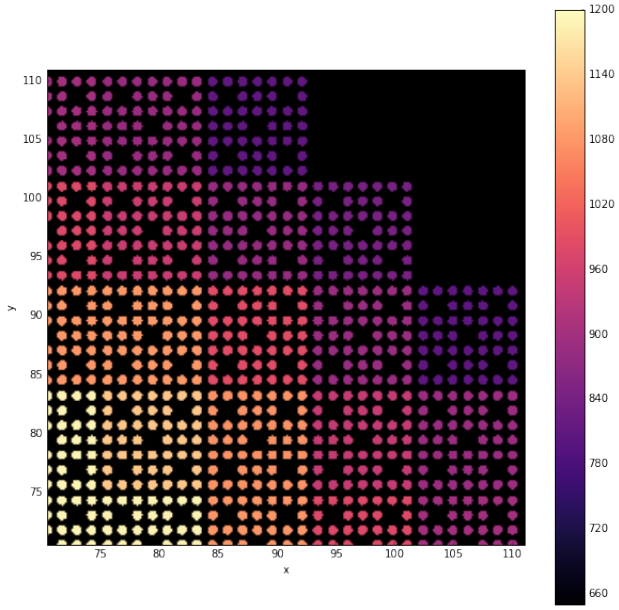
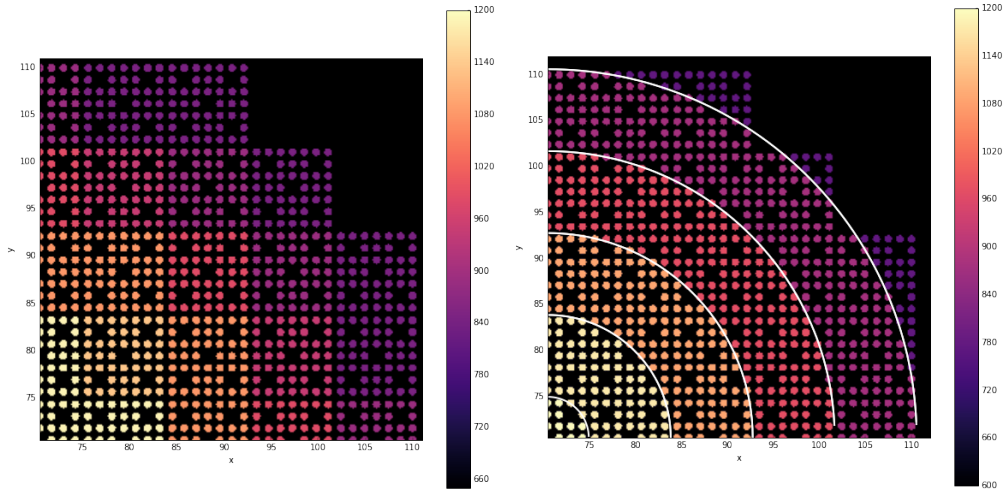


Figure 6.1: Example of assembly-averaged temperature homogenization in the fuel of progression problem 7 minicore [2].



(a) Assembly ring-averaged **(b)** Pin ring-averaged

Figure 6.2: Examples of ring-averaged temperature homogenization in the fuel of progression problem 7 minicore [2].

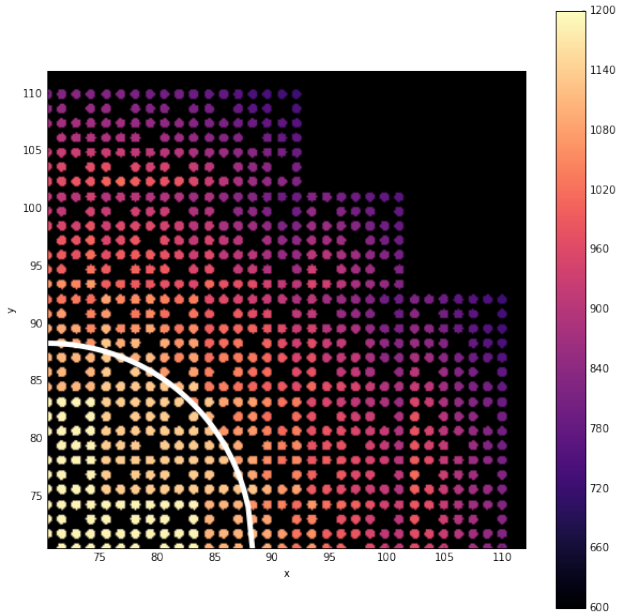


Figure 6.3: Example of explicit peripheral pins with assembly-averaged temperature homogenization in the fuel of progression problem 7 minicore [2].

6.2 Advanced postprocessing

Besides the capabilities of VERAView, there are a lot of advanced Python postprocessing capabilities in Exnihilo that are useful for ex-core analysis with VERA. The commands for performing postprocessing are not shown here because they are not typical user capabilities. The following advanced processing capabilities are shown so that an interested user can contact one of the Shift developers for further details.

The most useful capability is being able to visualize a ray trace of the geometry. As stated in § 4 and § 5, this ray trace is dumped to an HDF5 file if the **output_geometry** is set to *true*. Along with this ray trace, the compositions used in the Shift calculation for each state point are also dumped to separate HDF5 files. These compositions can be used to enhance the coloring of the Python visualization. The geometry figures in § 5 were produced using this Python visualization capability.

Figure 6.4 and Fig. 6.5 show examples of a plot by material using the ray trace and composition output files. Figure 6.6 and Fig. 6.7 show examples of using the ray trace by cell to interrogate different aspects of the ex-core geometry.

As mentioned previously, the fission source from MPACT used in the Shift calculation is dumped to the Shift HDF5 output file for every state. This source can also be visualized and interrogated with Shift postprocessing tools. Figure 6.8 shows an example of the spatial distribution of the fission source from progression problem 9 for the second state point plotted on top of the geometry.

Another postprocessing capability is visualization and inspection of the adjoint source and adjoint flux used in the CADIS calculation. Again, as mentioned in § 4, the adjoint source is output to a separate HDF5 file,

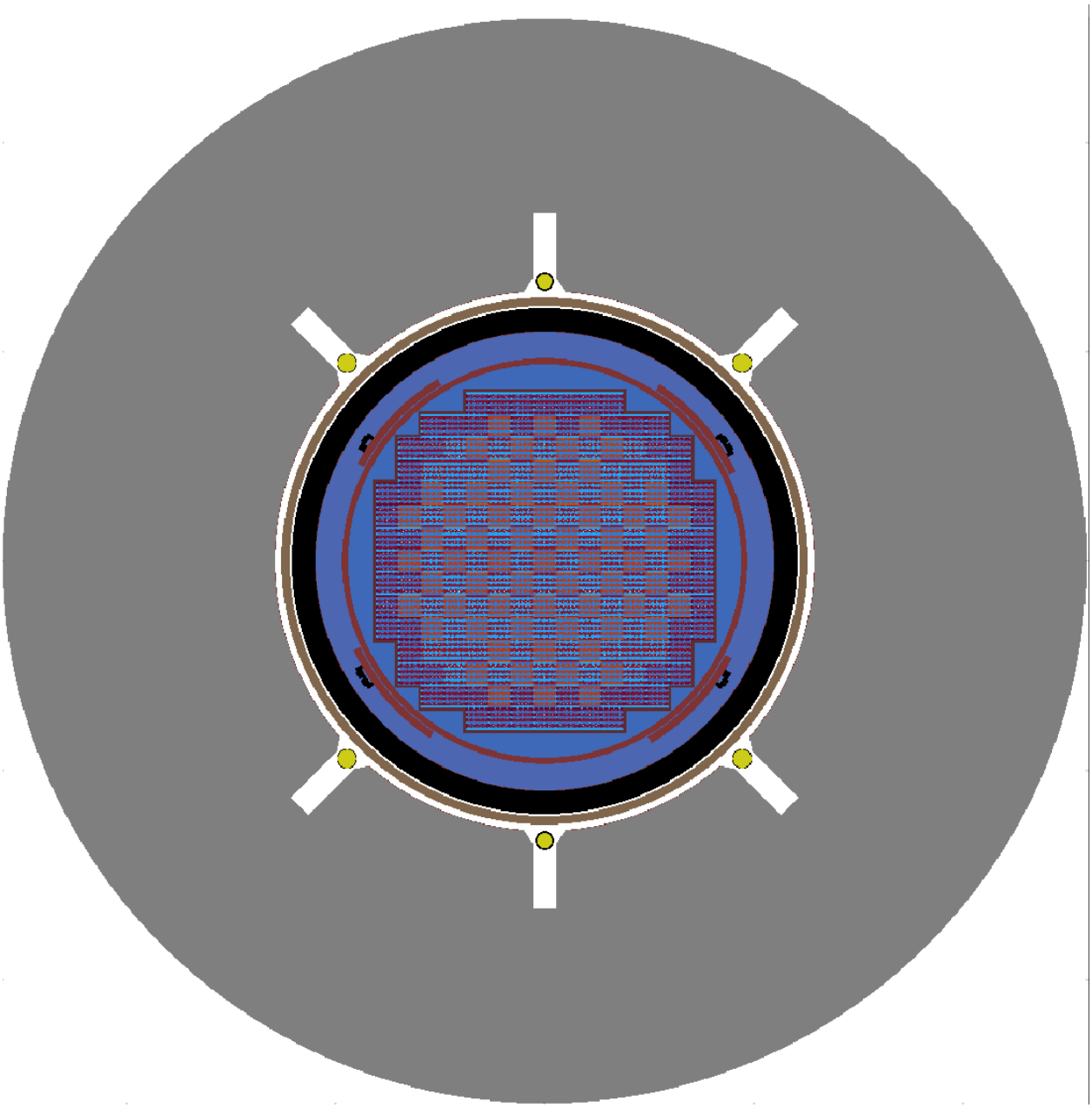


Figure 6.4: Full–core material ray trace x–y slice at axial midplane (courtesy of Eva Davidson). The colors represent different material compositions.

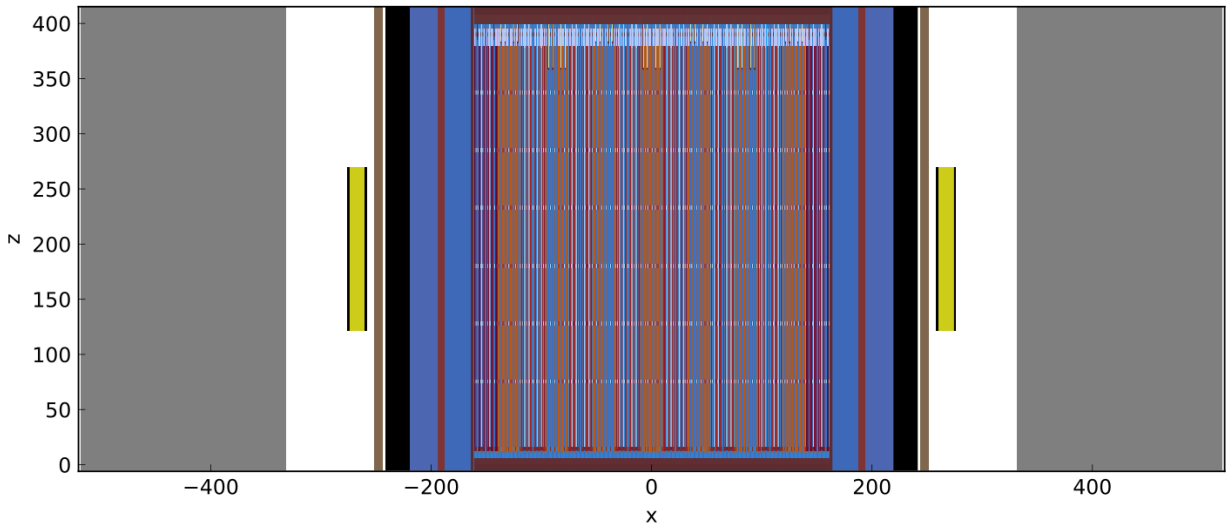


Figure 6.5: Full–core material ray trace x–z slice at $\mathbf{y} = \mathbf{0}$ (courtesy of Eva Davidson). The colors represent different material compositions.

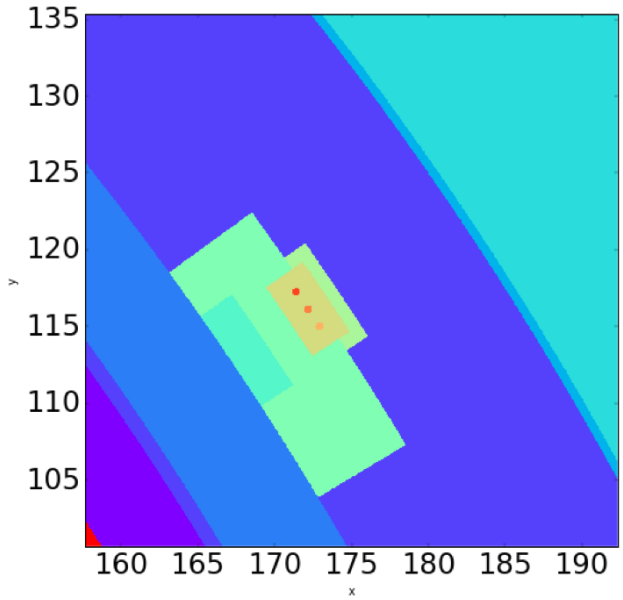


Figure 6.6: Irradiation capsules cell ray trace x–y slice (courtesy of Eva Davidson).

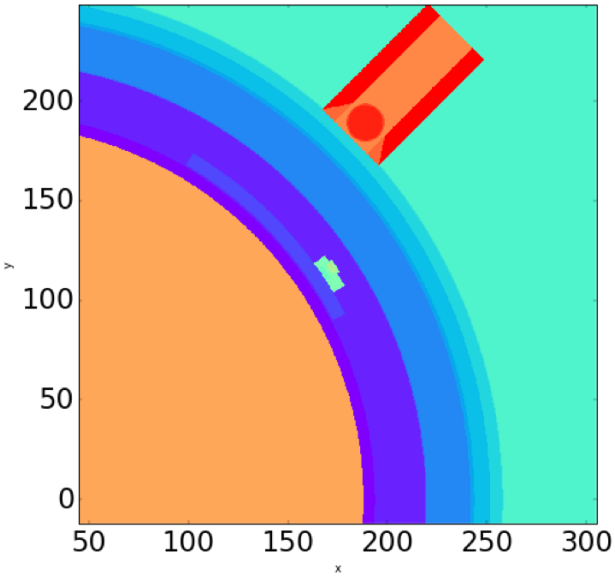


Figure 6.7: Source detector cell ray trace x–y slice (courtesy of Eva Davidson).

and the adjoint flux is output to the Shift HDF5 output file when running *vera_to_shift* if the **output_adjoint** parameter is set to *true*. Figure 6.9 shows an example of an axial slice of the adjoint source used in the CADIS calculation of progression problem 9. Figure 6.10 shows an axial slice of the calculated adjoint flux for a particular energy group from this same calculation.

Finally, another useful capability is the visualization of the weight windows used in the Shift calculation when using CADIS. These weight windows are output to the Shift HDF5 output file if the **output_ww** flag is set to *true*. Figure 6.11 shows an axial slice of the weight windows for a particular energy group for progression problem 9.

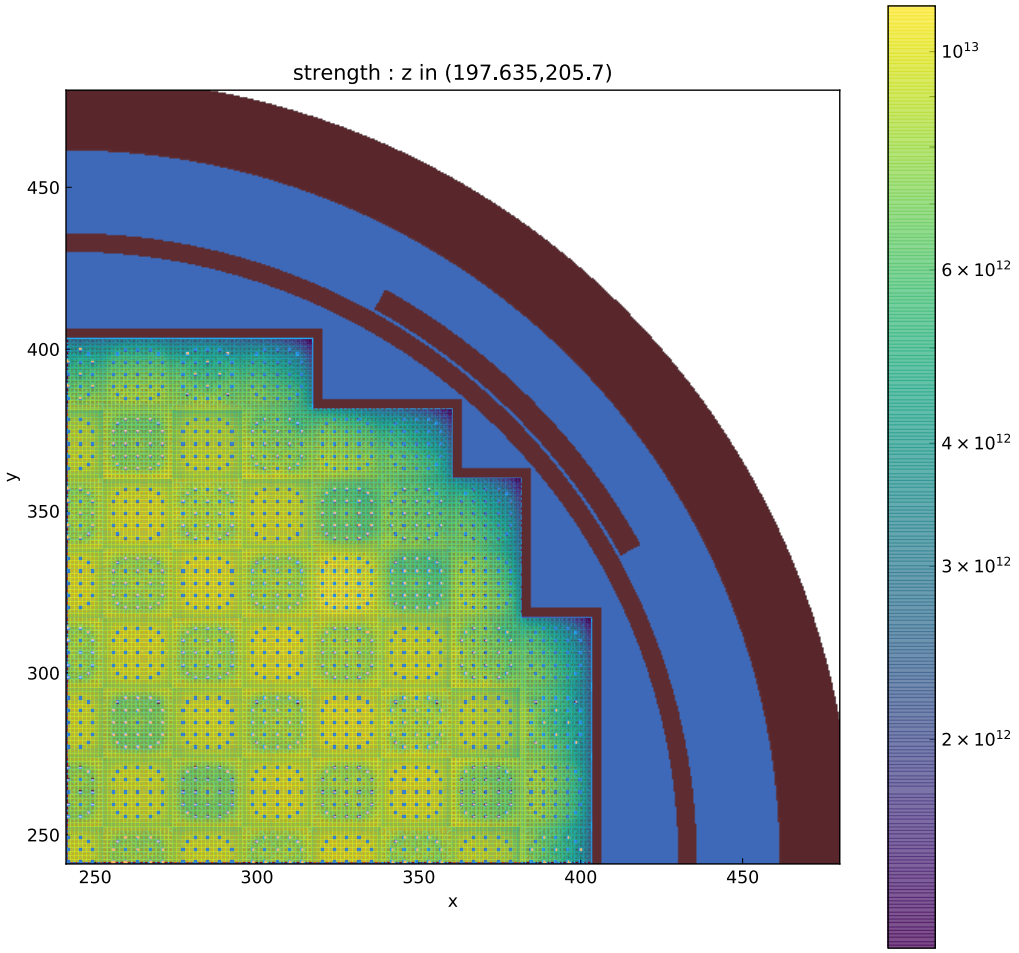


Figure 6.8: Progression problem 9 state 2 fission source x–y slice. Fission source color scale is only applied in the assemblies.

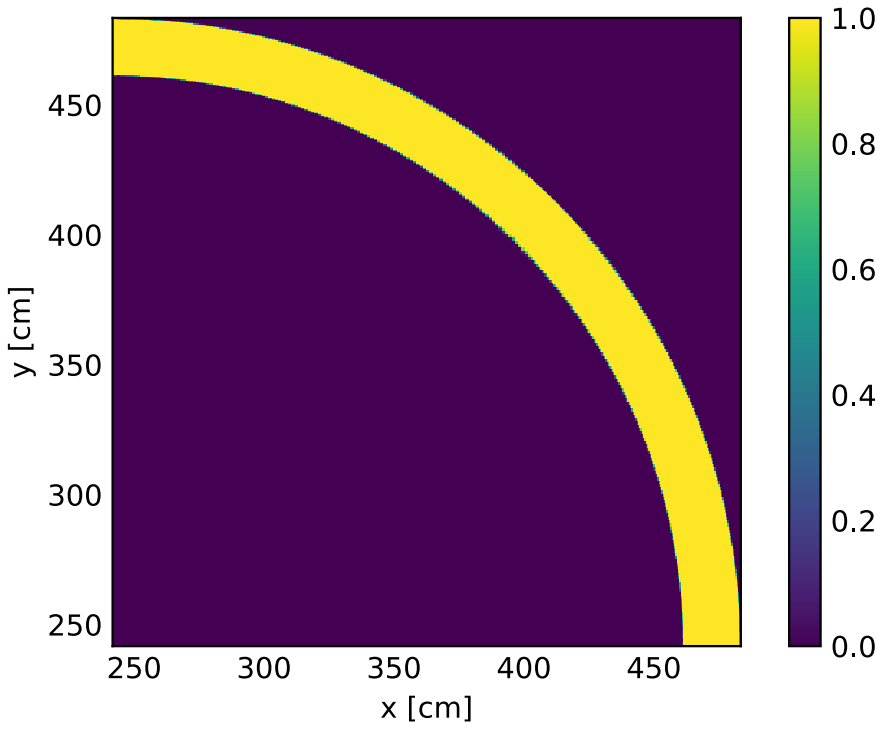


Figure 6.9: Progression problem 9 adjoint source x–y slice.

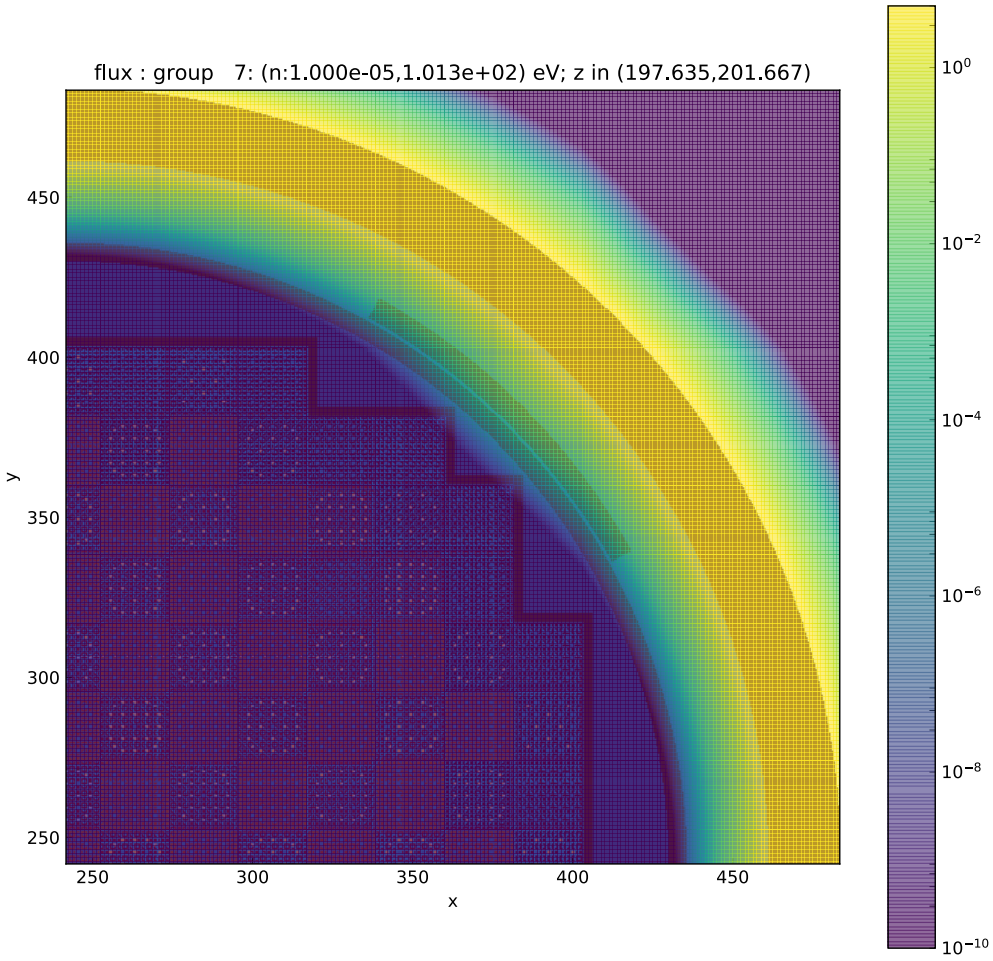


Figure 6.10: Progression problem 9 slowest energy group adjoint flux x–y slice. The flux is superimposed on a colored material plot of the geometry but the color scale shown applies only to the adjoint flux.

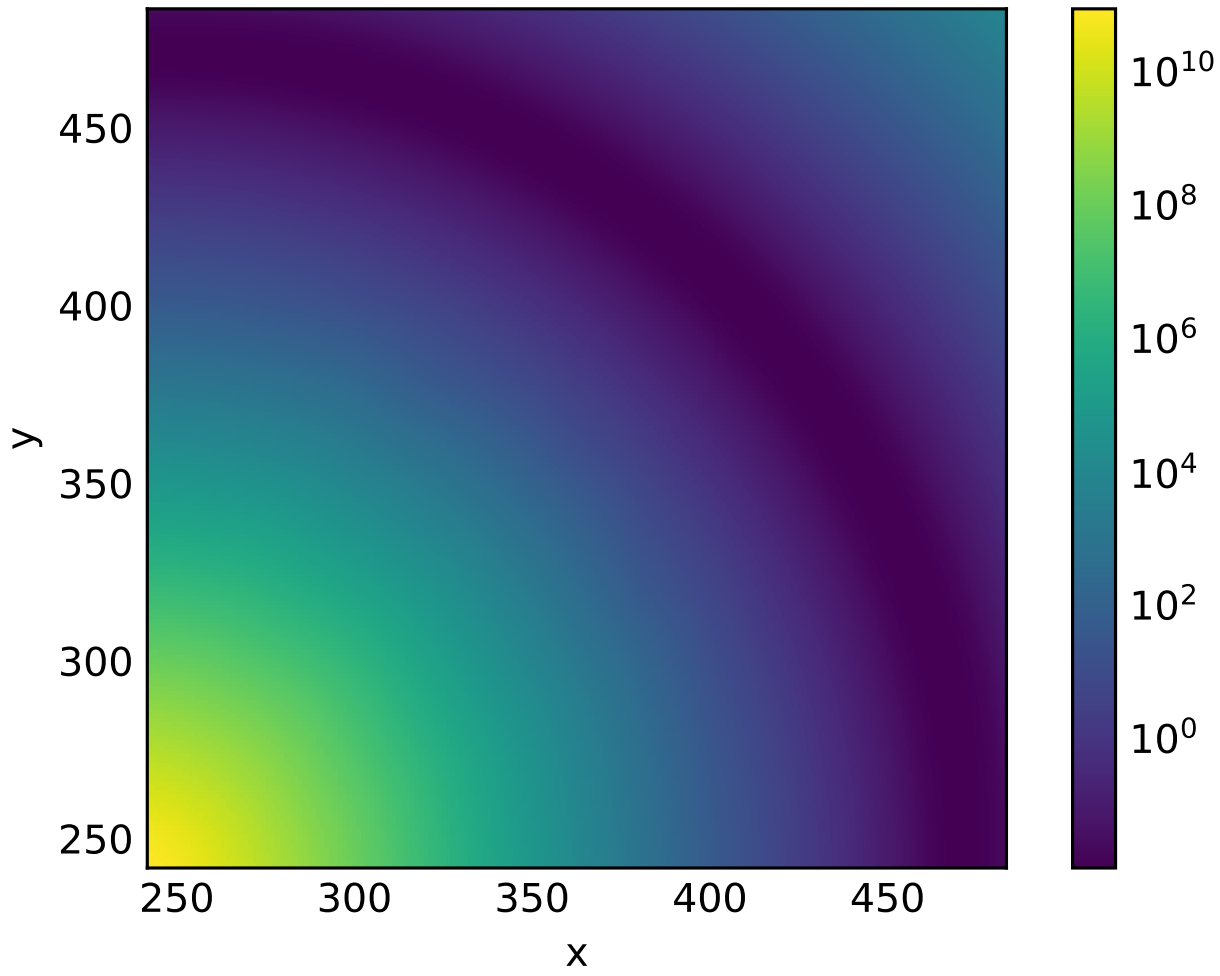


Figure 6.11: Progression problem 9 fastest energy group weight window lower bounds x–y slice.

7. Summary

This manual describes the ex-core calculation capability of VERA. Shift and MPACT are coupled in parallel to perform these calculations. MPACT 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 ex-core quantities. In both cases, the complete state of the core is transferred to Shift.

VERA includes the following capabilities for ex-core calculations:

- node-based parallelism using domain replication;
- multistate;
- fission source coupling;
- pincell isotopic coupling;
- temperature (fuel, clad, coolant) and coolant density coupling;
- fluence in core barrel, core pads, vessel liner, and vessel wall;
- supplemental geometric and tally input for ex-core model features and tallies;
- automatic generation of supplemental ex-core input for bioshield and detectors; and
- eigenvalue, forward, and CADIS modes.

Although all of the transfers listed below can be enabled, the full coupling of pincell isotopic compositions, temperatures, and densities is very memory intensive and will not run for some quarter-core and all full core problems.

Bibliography

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

[2] C. Stocking, An Investigation on the Impact of Temperature and Density Homogenization on Vessel Fluence Calculations, Thesis, North Carolina State University (2018).

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

[4] D. Jabaay, CASL MPACT User's Manual, Tech. Rep. CASL-U-2019-1873-000, CASL (July 2019).

[5] R. Salko, M. Avramova, A. Wysocki, J. Hu, A. Toptan, N. Porter, T. Blyth, C. Dances, A. Gomez, C. Jernigan, J. Kelly, CASL CTF User's Manual, Tech. Rep. CASL-U-2019-1885-000, CASL (March 2019).

[6] S. Palmtag, A. T. Godfrey, VERA Common Input User Manual, Tech. Rep. CASL-U-2014-0014-002, CASL (February 2015).

[7] J. C. Wagner, A. Haghighat, Automated variance reduction of Monte Carlo shielding calculations using the discrete ordinates adjoint function, Nuclear Science and Engineering 128 (2) (1998) 186–208.

[8] J. Wagner, D. Peplow, S. Mosher, FW-CADIS Method For Global And Regional Variance Reduction Of Monte Carlo Radiation Transport Calculations, Nuclear Science and Engineering 176 (1).

[9] A. Godfrey, R. Lee, VERAView User's Guide, Tech. Rep. CASL-U-2016-1058-000, Oak Ridge National Laboratory (March 2016).

[10] A. T. Godfrey, VERA Core Physics Benchmark Progression Problem Specifications, Tech. Rep. CASL-U-2012-0131-004, CASL (August 2014).

[11] CASL, CASL VERAIn public GIT Repository: VERA common input processor, https://github.com/CASL/VERAin, accessed October 2019 (October 2019).

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

[13] C. Stocking, T. Pandya, M. Avramova, Validation of MPACT Cross Section Library with Shift using VERA-CS, in: PHYSOR 2018: Reactor Physics paving the way towards more efficient systems, ANS PHYSOR Topical Meeting, American Nuclear Society, LaGrange Park, IL, 2018.

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

[15] T. M. Pandya, T. M. Evans, G. G. Davidson, S. R. Johnson, A. T. Godfrey, Shift verification and validation, Tech. Rep. CASL-U-2016-1186-000, Oak Ridge National Laboratory (2016).

[16] T. M. Pandya, S. R. Johnson, T. M. Evans, G. G. Davidson, 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, 2015.

[17] E. D. Biondo, G. G. Davidson, T. M. Pandya, S. P. Hamilton, T. M. Evans, Deterministically estimated fission source distributions for Monte Carlo k-eigenvalue problems, Annals of Nuclear Energy 119 (2018) 7–22.

[18] S. R. Slattery, P. P. H. Wilson, R. P. Pawlowski, The Data Transfer Kit: A Geometric Rendezvous-Based Tool for Multiphysics Data Transfer, in: M&C 2013, American Nuclear Society, 2013.

[19] X-5 Monte Carlo Team, MCNP–A General N-Particle Transport Code, Version 5, Volume II, Tech. Rep. LA-CP-03-0245, Los Alamos National Laboratory (2003).

[20] X-5 Monte Carlo Team, MCNP–A General N-Particle Transport Code, Version 5, Volume III, Tech. Rep. LA-CP-03-0284, Los Alamos National Laboratory (2003).

A. Omnibus Materials Template File

Listing A.1 shows the materials in the Omnibus template file included in VERA. These materials can be used with the bioshield and detector specifications in the VERA common input.

Listing A.1: Omnibus materials template file in VERA.

```
! ============================================================
!                    DEFINE THE COMPOSITIONS
! ============================================================
[COMP]

[COMP][MATERIAL void]
matid 0
temperature 293
zaid --
nd   --

! from page 114 of CASL progression problems (CASL-U-2012-0131-004)
[COMP][MATERIAL mod]
matid 1
temperature 293
zaid 1001 8016 5010 5011
nd   6.667279121238251E-02  3.333639560619125E-02 3.325516211867062E-05
     1.338562053118350E-04

! from page 114 of CASL progression problems (CASL-U-2012-0131-004)
[COMP][MATERIAL ss]
matid 2
temperature 293
zaid  6000 14028 14029 14030 15031 24050 24052 24053 24054 25055
     26054 26056 26057 26058 28058 28060 28061 28062 28064
nd   3.20895E-04 1.58197E-03 8.03653E-05 5.30394E-05 6.99938E-05
     7.64915E-04 1.47506E-02 1.67260E-03 4.16346E-04 1.75387E-03
     3.44776E-03 5.41225E-02 1.24992E-03 1.66342E-04 5.30854E-03
     2.04484E-03 8.88879E-05 2.83413E-04 7.21770E-05

! from page 114 of CASL progression problems (CASL-U-2012-0131-004)
[COMP][MATERIAL cs]
matid 3
temperature 293
zaid 6000 26054 26056 26057 26058
nd   3.93598E-03 4.89841E-03 7.68945E-02 1.77583E-03 2.36330E-04

!    Concrete, Oak Ridge (ORNL), 2.300000 g/cm3
!    material data from PNNL 15870 Rev 1
```

```
!    The above density is estimated to be accurate to 2 significant digits.
!    Uncertainties are not addressed. The following data were calculated
!    from the input weight and converted to number densities.
[COMP][MATERIAL concrete]
matid 4
temperature 293
zaid   1001   1002   6012   6013   8016   8017   11023 12024 12025 12026
       13027  14028  14029  14030  19039  19040  19041 20040 20042 20043
       20044  20048  26054  26056  26057  26058
nd     8.49996E-03 1.70019E-06 1.99790E-02 2.24257E-04 3.54247E-02
       8.52237E-05 1.63354E-05 1.46966E-03 1.86056E-04 2.04848E-04
       5.55961E-04 1.56827E-03 7.95785E-05 5.25422E-05 3.76029E-05
       4.03217E-09 2.71356E-06 1.07627E-02 7.21657E-05 1.55434E-05
       2.32041E-04 2.10946E-05 1.12950E-05 1.77148E-04  4.09321E-06
       5.40617E-07

[COMP][MATERIAL al]
matid 5
temperature 293
zaid 13027
nd   6.02626E-02
```

B. Defaults

The following defaults are initialized in the **SHIFT** block. All defaults were given in Tables 1– 4 but these are the values explicitly set by the VERA pre-processing script. Any of these parameters can be overridden in the VERA input file, but this is not recommended for typical users.

Listing B.1: Default SHIFT block parameters.

```
[SHIFT]
  problem_mode               cadis
  mode                       n
  verbosity                  low
  do_transport               true
  transfer                   fiss_src
  output_geometry            true
  output_fission_source      true
  output_external_source     false
  transport                  ce
  thermal_energy_cutoff      10.0
  Np                         1e6
  src_disc_l2_error          1e-3
  src_disc_samples_per_batch 1e7
  src_disc_max_samples       1e9
  fiss_src_spectrum          nuclide_watt
  use_fission_source         true
  use_external_source        false
  use_pole_data              false
  create_unique_pins         false
  global_log                 info
  eq_set                     sc
  Pn_order                   0
  mesh                       1
  adjoint                    true
  output_adjoint             false
  num_blocks_i               10
  num_blocks_j               10
  xs_library                 v7-56
  num_groups                 8
  new_grp_bounds             6.0653E+06 3.6788E+06 2.3457E+06
                             1.6530E+06 8.2085E+05 2.4176E+04
                             1.0130E+02 1.0000E-05
  cell_homogenize            true
  max_delta_z                5.0
  upscatter_verbosity        high
  upscatter_subspace_size    20
  refl_mesh_size             5.0
```

```
39   extend_axial_mesh_size      5.0
40   store_fulcrum_string        false
```

C. Geometry

The Shift transport engine supports multiple geometric implementations, [1, 12] and VERA supports two of these 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 its lack of support for curved channel boxes.

Exnihilo General Geometry (GG) (Exnihilo General Geometry): GG is a new general –purpose geometric modeling implementation in Shift [14]. 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 ex-core features. It also supports all of the incore elements that RTK handles. The GG package can easily support any future VERA extensions for reactor models.

Even though RTK is optimized for PWR geometries, the GG package performs favorably on CE problems, as illustrated in Fig. C.1. 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 run time, GG is only 17% slower than RTK.

As mentioned in § 2 and discussed in § D.1, the current implementation of VERA uses RTK for eigenvalue problems, GG for forward problems, and a combination of both for CADIS problems. There are no geometric impediments to using GG for eigenvalue problems; however, no current research interests exist to develop. The control flow to build the geometry in Shift through VERA is roughly equivalent for both models. The full incore model is built from the VERA input specification. A summary of the steps follows:

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 light water reactor–specific features.
3. Using the lightweight core model, build persistent core metadata to store the reactor state.
4. Build the geometric model builder for either RTK or GG, depending on the problem mode.

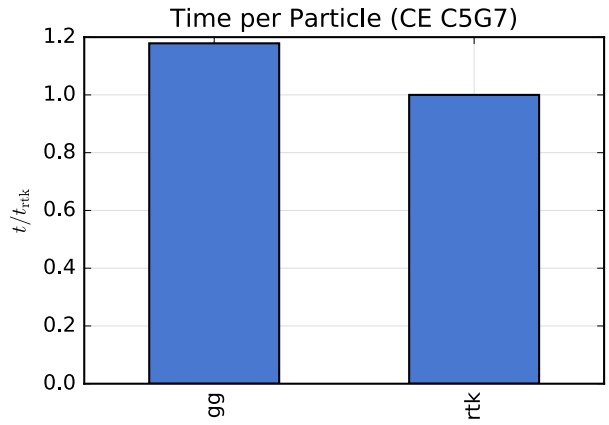


Figure C.1: Performance of GG compared with RTK on reactor geometries with CE physics.

5. In eigenvalue mode, use the geometric model builder to construct a complete incore reactor geometry for transport.
6. In forward mode, construct complete incore model out to the inner barrel or outer vessel radius. The ex-core builder then either adds the VERA vessel model, or it uses the supplemental ex-core model to define all geometric features outside the incore model.
7. Construct the problem decomposition using the model builder.
8. Construct the problem tallies using the model builder.

In CADIS mode, step 4 consists of meshing the RTK model for the deterministic adjoint calculation and building the GG geometry model for the MC calculation.

As described in step 3, VERA uses a metadata representation for the pincells and other core properties that is persistent throughout the simulation. The data transfer operations discussed in § D.2 update the metadata after each MPACT state point calculation. To allow for flexible updates from the VERA common input and to accommodate metadata updates, steps 2–8 are repeated for every state point.

In forward and CADIS problems, the geometry used for MC transport 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 or outer vessel radius, with ex-core model details provided by a supplemental input file. In the second case, the detailed core model can be translated to any position in space, which is a feature provided by the GG geometry. This mechanism can be used to drop a detailed core model 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 (these can be auto-populated if taking advantage of the automatic detector and bioshield option discussed in § 5.4). For option 1, all tallies and geometric model options are specified by the standard VERA input. Figure C.2 shows how the MPACT, Shift GG, and RTK geometry origins are defined by default in VERA.

A very useful feature through VERA is to dump axial slices of the geometry (from a ray trace in Shift) to an HDF5 file. This file can then be loaded into Python and visualized using processing utilities in Shift, as

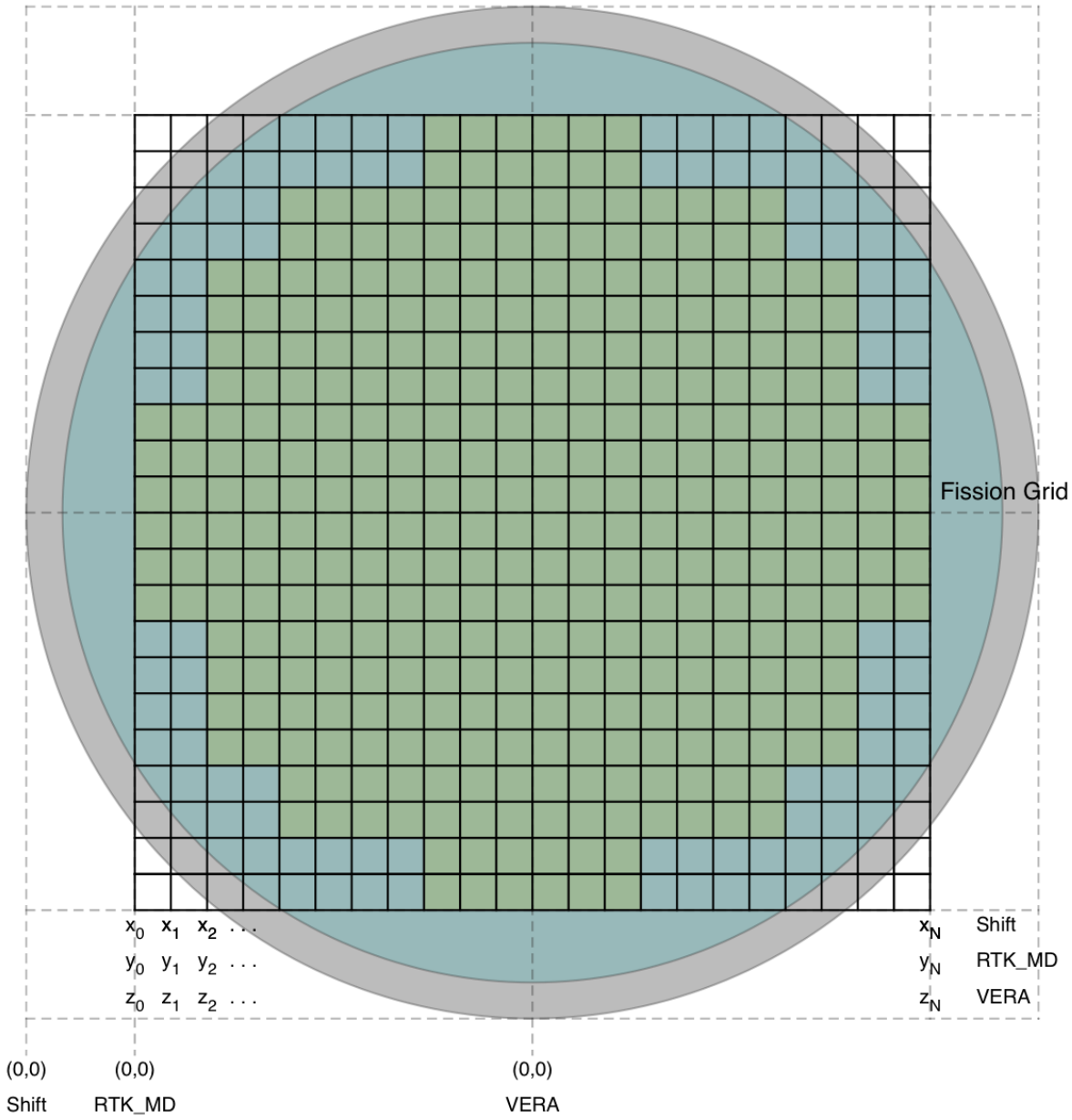


Figure C.2: Geometry origin relationships in VERA between Shift GG, RTK, and MPACT. In this figure, VERA refers to MPACT.

discussed in § 6.2. All of the input parameters supported for setting up a GG model can be found in the Shift manual for VERA [12]. An example of an ex-core GG model is given in § 5.

D. Methodology

This section presents details on how the VERA ex-core calculations are performed as implemented in the code itself, including methods of communication, and a more detailed explanation of what is transferred between MPACT and Shift.

D.1 Package overview

The purpose of the *vera_to_shift* executable 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. Validation calculations typically consist of executing MC k-eigenvalue calculations at the current reactor state. The MC transport package used in VERA is provided by Shift. Validation of Shift for CASL use is described in Pandya 2016b [15]. Ex-core analyses are performed with Shift through VERA by running fixed-source calculations.

Shift provides novel transport algorithms and hybrid methods tailored for leadership-class computing platforms, in addition to multiple geometries, tallies, and physics [1, 16]. For either vessel fluence or ex-core analyses, only one simulation is performed to calculate the quantities of interest by taking advantage of the coupling in VERA.

There are eight main areas to consider when discussing the capabilities of MC codes. A summary of these current ex-core capabilities in VERA are summarized in Table 11.

1. Energy treatment: The energy treatment used by Shift is CE physics for eigenvalue and forward problem modes. In CADIS mode, the deterministic S_N adjoint calculation uses multigroup physics, and the forward MC calculation uses CE physics.

2. Geometric representation: In CADIS mode, the adjoint calculation meshes the RTK geometry up to the core barrel inner radius and then meshes the ex-core geometry using the GG geometry representation. The multigroup cross sections are then generated on this combined mesh.

3. Particle types: Only neutrons are transported for eigenvalue calculations, whereas Shift can follow neutrons or coupled neutron-photon transport for forward and CADIS calculations.

4. Material isotopic treatment: For both modes, Shift can use the depleted isotopics in the fuel pins.

5. Temperature treatment, For both modes, Shift can use the fuel, clad, and moderator temperature from CTF for each pincell.

6. Types of parallelism: VERA replicates the problem domain on all Shift processors requested.

7. Types of tallies: Different quantities of interest are tallied based on the problem mode. For eigenvalue calculations, the energy-integrated fission rate on a rectilinear mesh covering the pincells is calculated along with the eigenvalue. For forward and CADIS calculations, VERA can calculate the following: (1) the energy-binned fluence in a cylindrical mesh covering the barrel and vessel by default; (2) ^{10}B or ^{235}U detector responses from the VERA input; or (3) any user-defined tally in the supplemental ex-core file.

8. Types of sources: In a fixed-source problem, the MPACT neutron fission source is used as the neutron source for n or n-γ transport. The fission source from prompt γ particles is not currently used from MPACT. When running a typical vessel fluence CADIS calculation, a detector response that is uniform in energy in the meshed vessel is used for the adjoint source, and the MPACT fission source is used for the MC forward calculation. For ex-core detector CADIS calculations, a user-defined detector response that is uniform in energy is meshed in the user-defined geometry cells for the adjoint source, and then the MPACT fission source is used for the MC forward calculation. As mentioned previously in § 2, the user can specify whether to only use the spatial distribution from MPACT with a default ^{235}U Watt energy spectrum, a nuclide-dependent Watt spectra, or pull in the full spatial and energy spectra from MPACT for the fission source. In eigenvalue problems, the spatial distribution of the MPACT 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 [17] under specific circumstances.

Shift has additional capabilities not interfaced to VERA (e.g., MC multigroup energy treatment); a complete detailing of Shift's capabilities is available in Pandya et al. 2016a [1] and Johnson et al. 2017 [12]. Most modeling features are treated identically in both modes. For example, the temperatures are updated in all fuel pins for eigenvalue, forward, and CADIS problem modes if requested.

Table 11: Ex-core capabilities of VERA.

	Problem mode		
Capability	Eigenvalue	Analog	CADIS
Energy	CE	CE	MG and CE
Geometry[a]	RTK; only simple vessel model supported	GG; core automated through VERA out through barrel or vessel, translation and additional geometric features enabled through supplemental input	meshed RTK for S_N adjoint and GG for MC
Particles	n	n or n-γ	n or n-γ
Isotopics	depleted isotopics in fuel, control rods, inserts, and boron concentration via MPACT	same as Eigenvalue	same as Forward
Temperature[b]	fuel, cladding, and moderator temperatures via CTF	same as Eigenvalue	same as Eigenvalue
Parallelism	domain replicated	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 [12])	same as Eigenvalue
Source	spatial fission source via MPACT; energy is either sampled from a ^{235}U Watt spectrum, nuclide-dependent Watt spectra, or MPACT fission spectrum	same as Eigenvalue	adjoint source is uniform energy response in vessel or user-defined detector response; MC source is same as Eigenvalue

[a] See § C for detailed geometry descriptions.

[b] See § 6.1 for experimental core temperature homogenization options.

D.2 Coupling algorithm

The basic algorithm for solving forward and CADIS calculations with VERA is shown in Alg. 1. First, a global message passing interface (MPI) communicator is established between MPACT and Shift. Shift and MPACT each run on their own sets of processors and on their own message passing interface (MPI) communicators. Next, the MPACT calculation is set up, along with the maps needed for the data transfer between MPACT and Shift. These maps use the Data Transfer Kit (DTK) [18] to handle the communication and data structures passed between MPACT and Shift. Next, MPACT launches a state point calculation. During this calculation, MPACT is run for the deterministic eigenvalue calculation, and CTF is run for the thermal hydraulics calculation.

Once a state point is complete in MPACT, the Shift eigenvalue, forward or CADIS calculation is launched. MPACT and Shift operate on independent domains. Thus, MPACT and Shift can execute different state points concurrently. Shift must wait on MPACT to finish an eigenvalue calculation for a state point before it can launch the forward or CADIS calculation of that state point. The vessel flux and some detector responses are communicated from Shift to MPACT. This communication is lagged by a state point.

The Shift calculation of a state point during the solve is either an MC forward calculation or a CADIS calculation, which performs an S_N adjoint calculation followed by an MC forward calculation. Shift uses the fission source transferred from MPACT as the source for the fixed source solve. The adjoint source for CADIS is either the detector response in the vessel, which is represented as uniform in energy, or the detector response in user-defined ex-core regions.

By default, Shift uses the short isotopic nuclide list from MPACT for the pins, control rods, and inserts of interest. The short nuclide list is given in § D.2.1. Shift can use the full isotopic nuclide list, but this introduces a large memory requirement for tracking all of the nuclides. Shift then runs the requested number of particle histories for the given state point and records the vessel flux tally (or ex-core detector tallies) in the HDF5 output file. The full cycle simulation is complete once MPACT completes the last state point and Shift completes the last requested forward calculation.

D.2.1 Transfer data

As mentioned previously, the fission source and depleted isotopic compositions for each state point can be transferred between MPACT and Shift. VERA also couples temperatures and coolant densities to Shift with CTF via MPACT for each state point.

The following specified quantities can be transferred between MPACT and Shift:

- in-core neutron fission source from MPACT eigenvalue calculation;
- depleted isotopic compositions for pincells and inserts using the short list of tracked nuclides: ^{234}U, ^{235}U, ^{236}U, ^{238}U, ^{16}O, ^{135}Xe, and ^{10}B;
- boron concentration in coolant;
- temperatures in fuel, clad, coolant of pincells and inserts; and
- density of coolant in pincells and inserts.

Algorithm 1 Simplified VERAShift simulation flow.

```
Parse VERA input
Set up MPI communicators
Construct MPACT and Shift model evaluators
Set up MPACT-to-Shift transfers
for each state do
    if processor running MPACT–CTF then
        Receive vessel flux and detector tallies from Shift to accumulate fluence for consolidated output
        Run coupled MPACT–CTF single state or predictor-corrector depletion calculation
        Send isotopic compositions, temperatures, densities, fission source, and stop criteria
        Continue to next state
    else if processor running Shift then
        Receive isotopic compositions, temperatures, densities, fission source, and stop criteria
        while !stop criteria do
            Run Shift forward/CADIS calculation and tally vessel flux and other ex-core regions
            Send vessel flux and detector tallies to MPACT
        end while
    end if
end for
```

Data are transferred at pincell-centered points, where the term *pincell* refers to a 2D slice of a fuel pin, insert, or control rod. Shift determines the cell-centroids by interrogating an internal reactor metadata model (see § C) and calculating the center point in each pincell axial location. This list of points is registered in DTK, which queries MPACT through the evaluator.

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

For the fission source transfer, VERAShift can transfer either a nuclide-dependent fission source or a space- and energy-dependent source from MPACT. The spatially dependent fission source from neutrons calculated by MPACT is given by Eq. D.1:

$$S_c = \sum_i \nu^i_{f,c} \Sigma^i_{f,c} \phi_c V_c \,, \tag{D.1}$$

where i is a fissionable nuclide, c is a pincell fuel region, $\nu^i_{f,c}$ is the number of neutrons produced from fission by nuclide i, $\Sigma^i_{f,c}$ is the macroscopic fission cross section of nuclide i, ϕ_c is the neutron flux in cell c where the energy dependence has been omitted, and V_c is the volume of cell c. The nuclide-dependent fission source is the default, and MPACT sends the spatial distributions (neutrons/s) for ^{235}U, ^{238}U, ^{239}Pu, and ^{241}Pu. Shift then samples the appropriate Watt energy spectrum based on the nuclide sampled for the fission source in a spatial location. The angular distribution of this fission source is always modeled as isotropic. The Watt energy spectrum is represented in Eq. D.2, where C is a normalization factor and the constants (a and b) used in Shift are given in Table 12.

$$W(E) = Ce^{-E/a} \sinh \sqrt{bE} \,. \tag{D.2}$$

With the space- and energy-dependent source from MPACT, Shift can either collapse the source in energy

Table 12: Nuclide Watt energy spectra constants used in Shift.

Nuclide	**a** (MeV)	**b** (1/MeV)	**Reference**
^{235}U	0.96500	2.2900000	MCNP5 manual (Cranberg spectrum) [19]
^{238}U	0.88111	3.4005000	MCNP5 manual [20]
^{239}Pu	0.96600	2.8420000	MCNP5 manual
^{241}Pu	1.33196	0.0929657	Manual fit to cross section data

and sample a ^{235}U Watt energy spectrum for all nuclides or use the full nonseparable source distribution on the energy groups used by MPACT. Note that using the full energy distribution from MPACT is memory intensive and the provided group structure may not be appropriate for ex-core calculations. Previous research has shown that the assumption of a Watt energy spectrum is most likely reasonable for eigenvalue calculations [17]. The transferred source is the initial fission source for eigenvalue calculations in Shift and the fixed-source for forward and CADIS calculations.

Although VERAShift can transfer all of the quantities mentioned in this section, for most realistic problems this is memory intensive and is not realistic to do on the current computing architectures, especially considering the replicated nature of the materials in Shift. For problems larger than a few assemblies, it is strongly recommended that depleted isotopic compositions, temperatures, and densities not be transferred. For most ex-core calculations, transferring only the fission source is sufficient for the responses of interest. The ability to take advantage of domain decomposition in Shift through VERAShift is being implemented and will allow for transferring of all of the quantities discussed in this section.