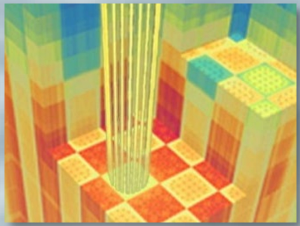
VERA Training:
Single Assembly Examples

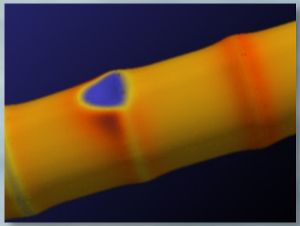
VERA Training – Core Simulator

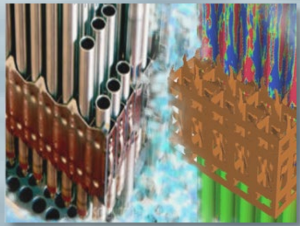
February 13, 2019

VERA Users Group Meeting

Oak Ridge National Laboratory







The Consortium for Advanced Simulation of LWRs
A DOE Energy Innovation Hub

U.S. DEPARTMENT OF ENERGY

Training Objectives

- Learn location of documentation and sample problems
- Work through several single-assembly examples in detail
  - 2D Single Assembly
  - IFBA, inserts, depletion
- Learn how to use job submission queue
- Learn about other features
  - Depletion
  - Restart files

Code Licenses

- VERA is export controlled software
- Everybody with access to the VERA code (source and/or executables) must have a valid software license
- You are responsible to not allow anybody else to get access to the codes

Has everybody signed the VERA User Agreement?

Lemhi

- Lehmi is the HPC cluster at INL where we will be running jobs
  - hpclogin.inl.gov is the login node (must log in here first)
  - lemhi1.hpc.inl.gov
  - lemhi2.hpc.inl.gov

- This training assumes that users already have an account on Lemhi and can log in using ssh or some other remote access client
  - Instructions to run a sample problem were sent out before training started

Location of Documentation

- For simplification, set the following Unix variable to the VERA source location:

```
export VERAHOME=
/projects/vera-users-grp/VERA_builds_lemhi/builds/VERA_4.0RC3
```

- General documentation: `$VERAHOME/share`
- MPACT documentation: `$VERAHOME/share/MPACT`
- CTF documentation: `$VERAHOME/share/COBRA-TF`
- VERAIn documentation: `$VERAHOME/share/VERAIn`
- Sample Problems:
  `$VERAHOME/share/VERAIn/Progression_Problems`

- VERAIn Input Manual
  `$VERAHOME/share/VERAIn/doc/verain_UM.pdf`
  (can also be downloaded from Github verain_UM.pdf)

Background: CASL Progression Problems

- #1 2D HZP Pin Cell
- #2 2D HZP Lattice
- #3 3D HZP Assembly
- #4 HZP 3x3 Assembly CRD Worth
- **#5 Physical Reactor Zero Power Physics Tests (ZPPT)**
- #6 HFP BOL Assembly
- #7 HFP BOC Physical Reactor
- **#8 Physical Reactor Startup Flux Maps**
- **#9 Physical Reactor Depletion**
- #10 Physical Reactor Refueling

- During CASL development, we used a set of "Progression Problems" to guide development.
- The nomenclature is still used in a lot of CASL documents
- For example, the case "P9" refers to "Problem 9", a full-core depletion
- "P2" is a 2D single assembly

- All progression problems are included in the VERAIn repository

More information on the progression problems can be found in the report:
http://www.casl.gov/sites/default/files/docs/CASL-U-2012-0131-004.pdf

VERA Input

- We’ve already covered the basics of the VERA input in the last training module.
- The VERA input is ASCII input arranged in “blocks”.
- Copy the 2D example file from the sample directory into a local directory:

```
mkdir run-vera
cd run-vera
cp $VERAHOME/share/VERAIn/Progression_Problems/2a.inp .
```

VERARUN

- New Program developed in VERA 4.0 to run jobs:

```
verarun [job]
```

- No longer need to write your own PBS scripts
- For list of advanced options, type “verarun” with no command line options
- Optional: set environment variable to turn on e-mail notifications

```
export PBS_EMAIL=USER@ornl.gov
```

- Optional: set run-time limits in new [RUN] block

```
[RUN]
  walltime 0.1    ! hours
```

2D Assembly Input (1)

```
[CASEID]
  title 'CASL Problem 2A'

[CORE]
  size    1           ! Single assembly
  apitch 21.50        ! Assembly pitch
  height 1.0          ! Core height
  rated  0.04832 0.01 ! Rated power and flow

  core_shape          ! Shape of core
    1

  assm_map            ! Map of assemblies
    ASSY

  bc_rad reflecting  ! Boundary conditions
  bc_top reflecting
  bc_bot reflecting
```

- CASEID is a header block that contains a title
- The core is defined as a single assembly named "ASSY"
- The rated power and flow are arbitrary since there is no feedback

Single assembly problem is set up as a core with a one assembly

2D Assembly Input (2)

```
[ASSEMBLY]
  npin   17        ! Number of rods across
  ppitch 1.26      ! Pin pitch

 ! 3.1% enriched fuel
  fuel U31 10.257 94.5 / 3.1

 ! Default materials for "he" and "zirc4"

  cell 1 0.4096 0.418 0.475 / U31 he zirc4
  cell 2        0.561 0.602 /     mod zirc4

  lattice LAT
    2
    1 1
    1 1 1
    2 1 1 2
    1 1 1 1 1
    1 1 1 1 1 2
    2 1 1 2 1 1 1
    1 1 1 1 1 1 1 1
    1 1 1 1 1 1 1 1 1

  axial ASSY 0.0 LAT 1.0
```

- Single 2D lattice
- "axial" card defines the assembly, but it is only one lattice 1 cm tall
- Cell names are "1" and "2"
- Lattice name is "LAT"
- Assembly name is "ASSY"

2D Assembly Input (3)

```
[STATE]
  power     0.0          ! Percent power %
  tinlet    565 K
  boron     1300         ! ppm
  sym       qtr

! Following parameters are only used
! With no feedback

  tfuel     565 K
  modden    0.743        ! g/cc
  feedback off
```

- STATE block sets the current reactor conditions
  - power (%)
  - Inlet temperature
  - Boron
  - Symmetry
- With no feedback, you also need to specify
  - Fuel temperature
  - Moderator density

2D Assembly Input (4)

```
[MPACT]

[SHIFT]
  num_cycles              1100
  num_inactive_cycles     100
  Np                      1000000
```

- Uses default MPACT values
- SHIFT options (and other code blocks) can be included in input, but they are ignored

Run Case

1. Copy input file and script into local directory

```
cp $VERAHOME/share/VERAIn/Progression_Problems/2a.inp .
```

2. Run

```
verarun 2a
```

3. Look for “k-eff” in summary file

Create Input, Run Case

VERA Output

VERA creates the following output files:

- $CASE.xml – the VERAIn parsed input to the codes
- $CASE.log – a record of the MPACT information echoed to the screen during the run
- $CASE.out – MPACT output file
- $CASE.sum – Statepoint summary
- $CASE.h5 – a standard hdf5 formatted file containing results for VERAView (binary file)
- other files used for scripting

Examine output file and confirm we get the correct answers!

2D Assembly Summary File

```
*************************************************************************
****************************** STATE_0001 *******************************
*************************************************************************

     State Summary
     -----------------------
     Core Exposure              0.00 MWD/MTHM
     Relative Power             0.00 %
     Thermal Power              0.00 MWt
     Relative Flow            100.00 %
     Absolute Flow              0.31 kg/s
     Inlet Temperature        291.85 C
     Boron Conc.             1300.00 ppm
     k-eff                      1.18228
```

Runs in about 30 seconds

Results for Vera 4.0
Results in previous versions may have included resonance upscatter models

Class Exercise 1: Add Gad

- Start with the input deck "2a.inp"
- Gadolinia is an integral burnable absorber that is part of the fuel
- Convert input deck to include gad rods
  - Include 12 gad rods as shown in map below (location X)
  - Gad fuel is enriched to 1.8% U-235 and 5% gadolinia

```
fuel g18 10.111 94.5 / 1.8 / gad=5.0
```

U-235 enrichment (points to 1.8)

Gad material and enrichment (integral component of fuel) (points to gad=5.0)

```
2
1 1
1 1 1
2 1 1 2
1 1 X 1 1
1 1 1 1 1 2
2 1 1 2 1 1 X
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
```

Create Input, Run Parser

Gad Input Solution

```
[ASSEMBLY]
  fuel u31 10.257 94.5 / 3.1
  fuel g18 10.111 94.5 / 1.8 / gad=5.0

  cell 1 0.4096 0.418 0.475 / U31  he zirc4
  cell X 0.4096 0.418 0.475 / g18  he zirc4
  cell O        0.561 0.602 /     mod zirc4

  lattice LAT
    2
    1 1
    1 1 1
    2 1 1 2
    1 1 X 1 1
    1 1 1 1 1 2
    2 1 1 2 1 1 X
    1 1 1 1 1 1 1 1
    1 1 1 1 1 1 1 1 1
```

Needed:

- “gad” is a material name that is already defined on library
- New fuel name “g18” with gad
- New rod definition with gad fuel
- New lattice map showing gad rods

Need to define a second fuel for gad pins and a second cell type that uses the gad fuel

Gad Output in Summary File

```
********************************************************************************
******************************* STATE_0001 *************************************
********************************************************************************

     State Summary
     -----------------------
     Core Exposure            0.00 MWD/MTHM
     Relative Power           0.00 %
     Thermal Power            0.00 MWt
     Relative Flow          100.00 %
     Absolute Flow            0.31 kg/s
     Inlet Temperature      291.85 C
     Boron Conc.           1300.00 ppm
     k-eff                    1.05140
```

Eigenvalue much lower than case 2A (1.18228)
Gad rods have very low power (in VERAView)
Runs in about 40 seconds

Results should be very similar to Progression Problem 2o

Notes on Gad depletion

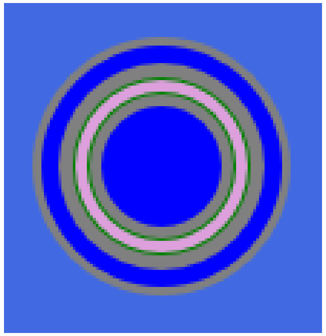
- Gadolinia is a very black absorber and tends to burn like an “onion skin”.
- Therefore, you need to use 10 depletion rings to get good results with gad pins instead of the default 3 rings.
- The code will automatically use 10 depletion rings in fuel rods with gadolinia

```
[MPACT]
 mesh cell_gad 10 1 1 / 10*8 8 8 8

! Number of rings and azimuthal
! divisions in each region
! One additional azimuthal for coolant
```

In older versions of VERA, the user had to manually increase the number of rings in the gad fuel

Class Exercise 2: Add WABA



- Start with the base input deck “2a.inp”
- WABA are discrete burnable absorber rods that are placed in guide tubes (like control rods)
- Convert input deck to include 12 WABA rods
  - Need to add INSERT block and core map
  - WABA materials are mod/zirc4/he/waba/he/zirc4
  - WABA radii are 0.286 0.339 0.353 0.404 0.418 0.484
  - 12 WABA rods (need map)

```
mat waba    3.65    b4c 0.0949
                  al2o3 0.9051
```

```
-
- -
- - -
1 - - -
- - - - -
- - - - - -
- - - 1 - - -
- - - - - - - -
- - - - - - - - -
```

Create Input, Run Parser

WABA Input Solution

```
[INSERT]
  npin 17

  mat waba    3.65    b4c 0.0949
                    al2o3 0.9051

  cell 1  0.286 0.339 0.353 0.404 0.418 0.484 /
          mod zirc4 he waba he zirc4

  rodmap  LAT12
     -
     - -
     - - -
     1 - - -
     - - - - -
     - - - - - -
     - - - 1 - - -
     - - - - - - - -
     - - - - - - - - -

  axial 12WABA  0.0 LAT12 1.0
```

```
[CORE]
  assm_map
    ASSY

  insert_map
    12WABA
```

Need:

- Create a new INSERT block for the WABA inserts
- Follow format of ASSEMBLY block
- The WABA rods must fit inside of the guide tubes
- Add WABA inserts in a core map

WABA Output

```
******************************************************************************
****************************** STATE_0001 ************************************
******************************************************************************

     State Summary
     -----------------------
     Core Exposure             0.00 MWD/MTHM
     Relative Power            0.00 %
     Thermal Power             0.00 MWt
     Relative Flow           100.00 %
     Absolute Flow             0.31 kg/s
     Inlet Temperature       291.85 C
     Boron Conc.            1300.00 ppm
     k-eff                     1.07518
```

Eigenvalue lower than case 2a, but higher than 2a-gad
Pin powers are lower around WABA rods (VERAView)
Runs in about 38 seconds

Class Exercise 3: Add IFBA

- Start with the base input deck "2a.inp"
- IFBA is a very layer of ZrB on outside of fuel rods
- Convert input deck to include 80 IFBA rods
  - Need IFBA composition
  - Need IFBA thickness and density (need to preserve total loading)
  - Recommend thickness 0.001 cm, define density to preserve loading
  - 80 IFBA rods (need map)

```
mat ifba 3.85  zr-90   0.412271
               zr-91   0.090907
               zr-92   0.140481
               zr-94   0.145466
               zr-96   0.023935
               b-10    0.09347
               b-11    0.09347
```

```
2
X 1
1 1 X
2 X 1 2
X 1 1 X 1
1 1 1 X X 2
2 X 1 2 X 1 X
X 1 1 X 1 1 1 1
1 1 1 1 1 1 1 1 X
```

Create Input, Run Parser

IFBA Input Solution

```
[ASSEMBLY]
  npin   17
  ppitch 1.26

  mat ifba 3.85  zr-90   0.412271
                 zr-91   0.090907
                 zr-92   0.140481
                 zr-94   0.145466
                 zr-96   0.023935
                 b-10    0.09347
                 b-11    0.09347

  fuel U31 10.257 94.5 / 3.1 u-234=0.026347

  cell 1 0.4096        0.418 0.475 / U31      he zirc4
  cell X 0.4096 0.4106 0.418 0.475 / U31 ifba he zirc4
  cell 2               0.561 0.602 /         mod zirc4

  lattice LAT
    2
    X 1
    1 1 X
    2 X 1 2
    X 1 1 X 1
    1 1 1 X X 2
    2 X 1 2 X 1 X
    X 1 1 X 1 1 1 1
    1 1 1 1 1 1 1 1 X

  axial ASSY 0.0 LAT 1.0
```

Need:

- Add IFBA material
- Add new cell card with IFBA layer on outside of fuel
- Add IFBA rod locations to lattice map

IFBA Output

```
********************************************************************************
******************************* STATE_0001 *************************************
********************************************************************************

     State Summary
     -----------------------
     Core Exposure             0.00 MWD/MTHM
     Relative Power            0.00 %
     Thermal Power             0.00 MWt
     Relative Flow           100.00 %
     Absolute Flow             0.31 kg/s
     Inlet Temperature       291.85 C
     Boron Conc.            1300.00 ppm
     k-eff                     1.02225
```

Eigenvalue lower than case 2a and 2a-gad

Use VERAView to see low power rods

Runs in about 30 seconds

Notes on IFBA Ray Spacing

- IFBA is a very small geometric ring, so it has been found that you need a smaller ray spacing to get accurate results

```
[MPACT]
  ray_spacing     0.005
```

- May cause significant increase in run-time
- Makes a big difference for single assembly problems, but results in a full-core are not as sensitive. User should use their own judgement.

```
Default       0.050  k-eff    1.02225    0:31 min
ray_spacing 0.005  k-eff    1.02068    2:50 min
ray_spacing 0.001  k-eff    1.02062   13:10 min
```

Class Exercise 4: Depletion

- For depletion, we need to set the correct power level (you can't deplete at zero power)
- In CORE block, confirm that the rated power should be 0.04832 MW for a single assembly 1 cm tall
- In STATE block, set the instantaneous power to 100%
- A more realistic fuel temperature is 900 K
- Depletion is performed with a single input

```
deplete [unit] [step(s)]
```

- Units can be EFPD, GWDMT, or hours

```
[STATE]
  power   100.0    ! %
  deplete EFPD 0 10 20 40 60
  tfuel 900 K

[CORE]
  rated   0.04832 0.01   ! MW
```

Create Input, Run Parser

Depletion Output

```
==================================
      Statepoint Summary
==================================
   N   exposure  exposure  eigenvalue   2PIN      3EXP
   1    0.0000       0.00   1.172147  1.0515    0.0000
   2    0.3841      10.00   1.128630  1.0525    0.4040
   3    0.7682      20.00   1.124366  1.0527    0.8083
   4    1.5363      40.00   1.118837  1.0527    1.6169
   5    2.3045      60.00   1.112637  1.0526    2.4255
```

Output from a post processing code reading HDF file

Runs in about 2.5 min

Depletion Step Shortcut

- Long depletion cards can be entered with a shortcut list generator

<n..mxd>

- where
  - "n" is the beginning step
  - "m" is the ending step
  - "d" is the step size
- Examples:

```
deplete GWDMT 0 0.1 0.5 <1..20x1>
deplete GWDMT <20..60x5>
deplete EFPD  <0..300x10> 305 308
```

- List expansion is done in the input parser, so the XML file will include all depletion steps

Restart Files

- Restart file can be used to save statepoint data and then restart the case later
- Important for long-running jobs and core shuffles

- Write a restart file:

```
restart_write FILE LABEL
```

- Read a restart file:

```
restart_read FILE LABEL
```

- FILE is the filename of a binary restart file
- LABEL is an arbitrary user-defined string

Restart Examples

- Write a restart file in one input deck:

```
[STATE]
  deplete GWDMT 0 0.1 0.5 1 2 4 6 8
  restart_write cycle4.res EXP8
[STATE]
  power 99.0
  deplete GWDMT 10 12 14 16 18 20
  restart_write cycle4.res EXP20
```

- Read restart file in another input deck:

```
[STATE]
  restart_read cycle4.res EXP8
  power 25.0
```

You still need all the other geometry input (CORE, ASSEMBLY, INSERT, etc.) when reading restart files

Restart File Rules

- Restart files are HDF files, but not VERAout files (you can't view them in VERAView)
- You can have multiple statepoints per restart file, they just need to have unique labels
- Restart file ONLY contains isotopic data and [STATE] data
  - User needs to include all other blocks (ASSEMBLY, CORE, etc.) in the input deck that reads the restart file
  - This behavior may be removed in future code versions
- Restart file is written for the last exposure if a "deplete" card is use with multiple exposures
- Restart file can be used to expand/contract from full-core to qtr-core symmetry

Class Exercise 5: Restart Files

- Start with the depletion test case
- Write a restart file at the last exposure step

Create Files and Submit Job

- Create another input file to read the restart file
  - Remember that you still need to include CORE and ASSEMBLY data

Create Files and Submit Job

- Do the answers agree?

Thermal Expansion (TE)

- TE can have significant effect on corner pin powers due to increased assembly gap size
- TE is performed once at beginning of calculation
  - We do not perform coupled thermo-mechanics
  - This is a small approximation because largest TE effect is from hot to cold, local temperature effects are smaller.
  - TE is assumed to be unconstrained
- TE temperatures must be specified beforehand. Set to average values
- Run script automatically performs TE

```
[STATE]
  thexp        on
  thexp_tmod   585 K
  thexp_tclad  600 K
  thexp_tfuel  900 K
```

Thermal Expansion (TE) Caveats

- 3D TE is available, but not recommended
  - Most of the TE effect is 2D from the core plate expansion
- User can input different TE temperatures for HZP and HFP
- User can not change TE temperatures AND do axial re-mesh at the same time
  - more details in full-core discussion

Thermal Expansion Demonstration

- Problem 2A with fuel temperature increased to 900K
- Pin Power differences (%) (TE on – TE off)

2.29	1.50	1.12	0.95	0.80	0.74	0.72	0.78	0.87	0.78	0.72	0.74	0.80	0.95	1.12	1.50	2.29
1.50	0.52	0.44	0.36	0.20	0.02	-0.03	0.06	-0.11	0.06	-0.03	0.02	0.20	0.36	0.44	0.52	1.50
1.12	0.44	0.04	-0.17	-0.33		-0.28	-0.34		-0.34	-0.28		-0.33	-0.17	0.04	0.44	1.12
0.95	0.36	-0.17		-0.46	-0.47	-0.59	-0.53	-0.61	-0.53	-0.59	-0.47	-0.46		-0.17	0.36	0.95
0.80	0.20	-0.33	-0.46	-0.59	-0.57	-0.58	-0.61	-0.64	-0.61	-0.58	-0.57	-0.59	-0.46	-0.33	0.20	0.80
0.74	0.02		-0.47	-0.57		-0.66	-0.78		-0.78	-0.66		-0.57	-0.47		0.02	0.74
0.72	-0.03	-0.28	-0.59	-0.58	-0.66	-0.78	-0.75	-0.72	-0.75	-0.78	-0.66	-0.58	-0.59	-0.28	-0.03	0.72
0.78	0.06	-0.34	-0.53	-0.61	-0.78	-0.75	-0.74	-0.61	-0.74	-0.75	-0.78	-0.61	-0.53	-0.34	0.06	0.78
0.87	-0.11		-0.61	-0.64		-0.72	-0.61		-0.61	-0.72		-0.64	-0.61		-0.11	0.87
0.78	0.06	-0.34	-0.53	-0.61	-0.78	-0.75	-0.74	-0.61	-0.74	-0.75	-0.78	-0.61	-0.53	-0.34	0.06	0.78
0.72	-0.03	-0.28	-0.59	-0.58	-0.66	-0.78	-0.75	-0.72	-0.75	-0.78	-0.66	-0.58	-0.59	-0.28	-0.03	0.72
0.74	0.02		-0.47	-0.57		-0.66	-0.78		-0.78	-0.66		-0.57	-0.47		0.02	0.74
0.80	0.20	-0.33	-0.46	-0.59	-0.57	-0.58	-0.61	-0.64	-0.61	-0.58	-0.57	-0.59	-0.46	-0.33	0.20	0.80
0.95	0.36	-0.17		-0.46	-0.47	-0.59	-0.53	-0.61	-0.53	-0.59	-0.47	-0.46		-0.17	0.36	0.95
1.12	0.44	0.04	-0.17	-0.33		-0.28	-0.34		-0.34	-0.28		-0.33	-0.17	0.04	0.44	1.12
1.50	0.52	0.44	0.36	0.20	0.02	-0.03	0.06	-0.11	0.06	-0.03	0.02	0.20	0.36	0.44	0.52	1.50
2.29	1.50	1.12	0.95	0.80	0.74	0.72	0.78	0.87	0.78	0.72	0.74	0.80	0.95	1.12	1.50	2.29

- Reactivity worth of TE = -49 pcm

Next

- In the next training section we will set up and run 3D Problems

Questions?

A DOE Energy Innovation Hub

www.casl.gov