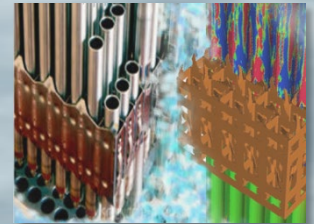
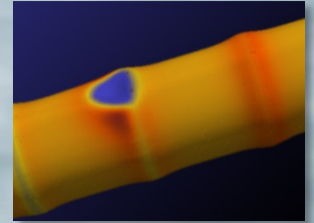
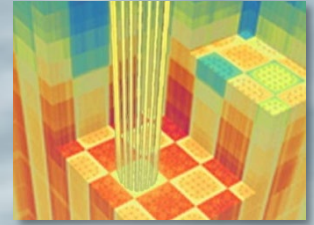


VERA Training: Input Description

VERA Training – Core Simulator
February 13, 2019
VERA Users Group Meeting
Oak Ridge National Laboratory



Training Objectives

- Give short overview of PWR core components
- Show how the reactor components are connected to the input
- We will not cover every input card, just the important concepts

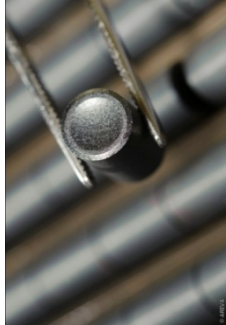
Hands-on training will be given in later sections

Agenda

- Introductions
- Short PWR Overview
 - Go over quickly for experienced users
- Detailed Review of Input
 - Geometry Concepts
 - Assemblies
 - Inserts
 - Control Rods
 - Statepoints
 - Code Options

PWR Core Components

PWR Fuel Rods

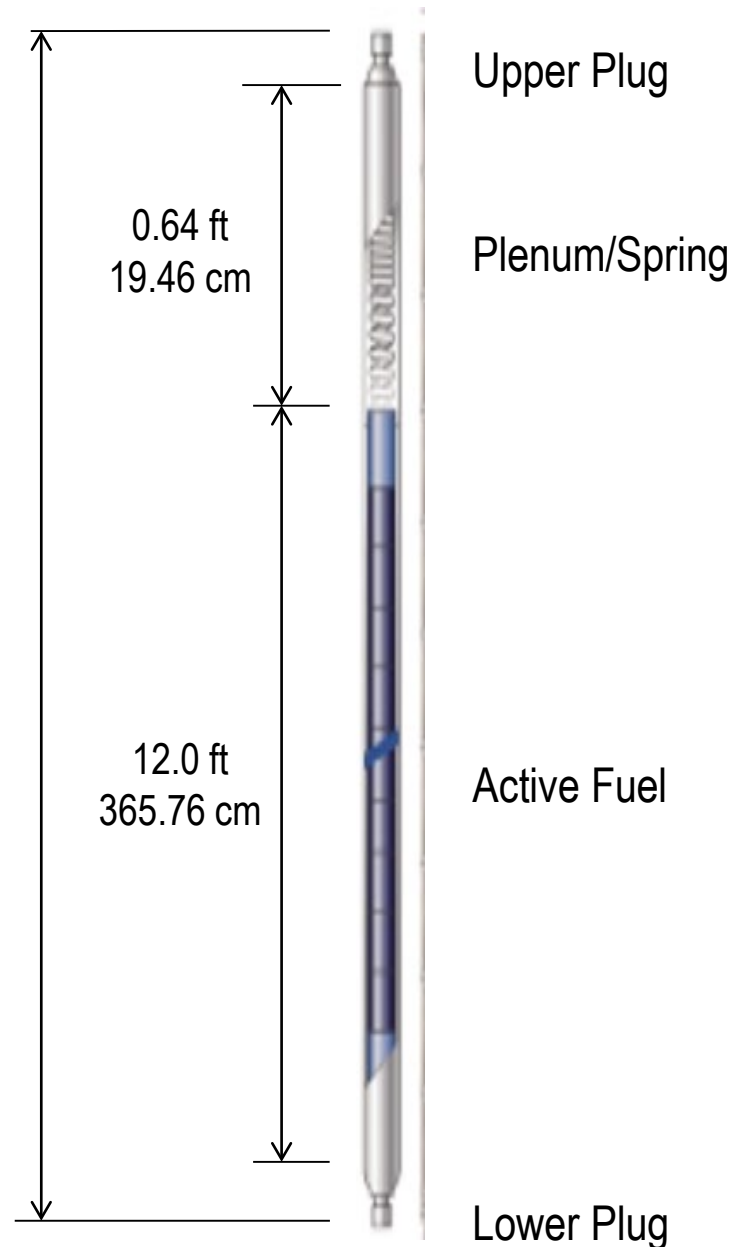


PWR Fuel Pellet
(0.82 cm OD)

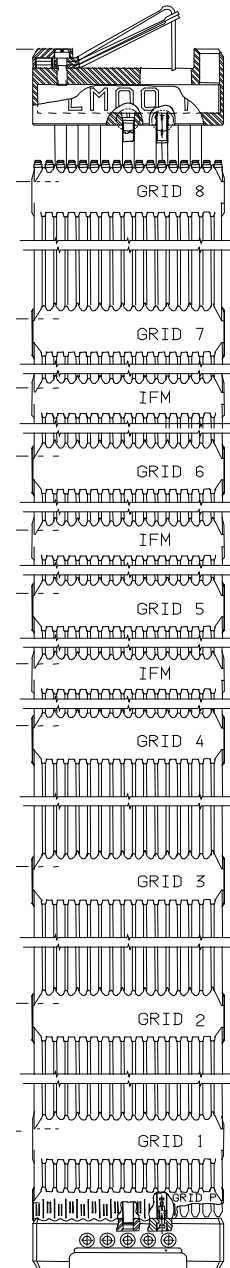
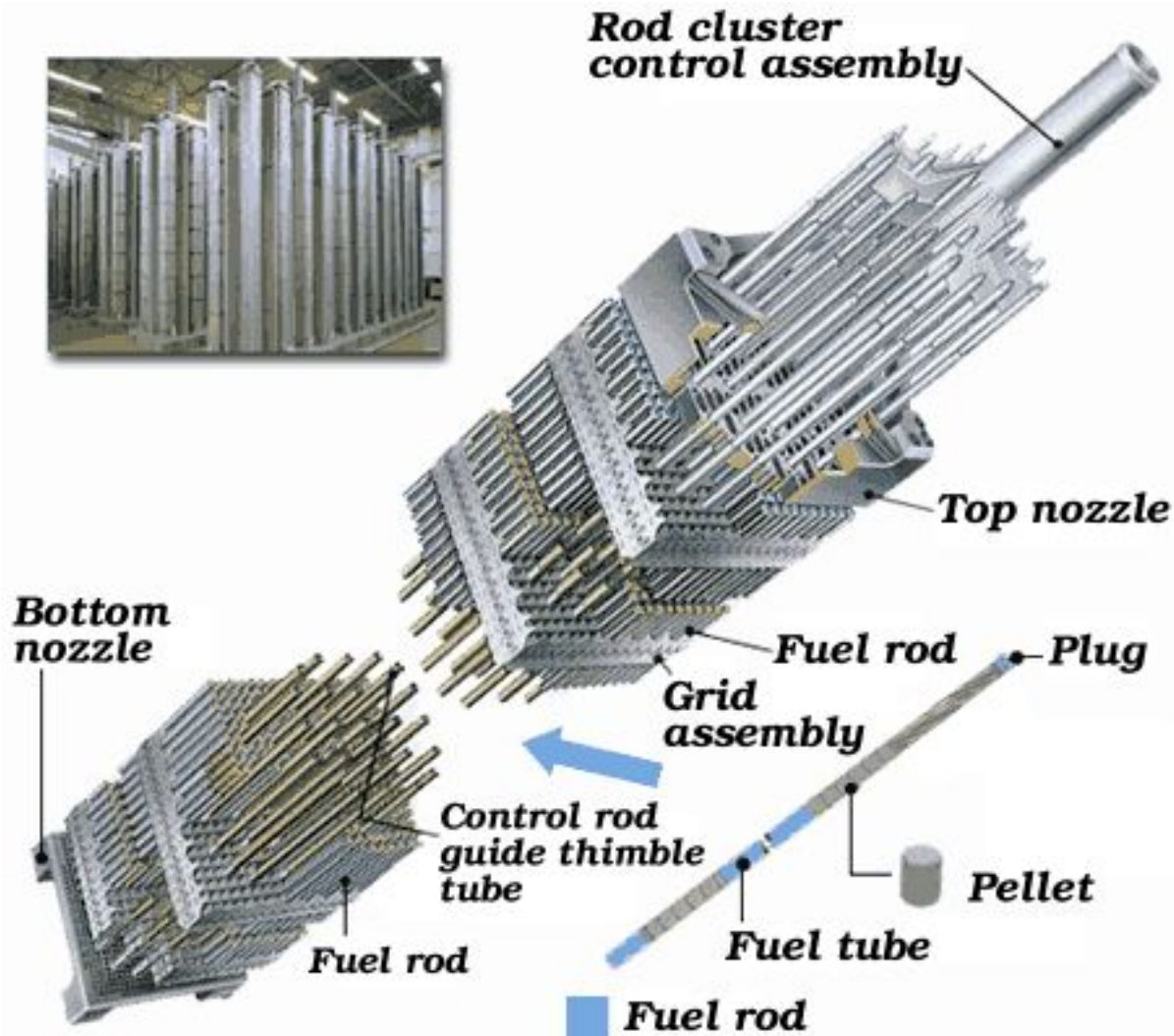
- UO_2 Pellets
- multiple initial U-235 enrichments
(2.1, 2.6, 3.1 w/o heavy metal)
- Density ~ 10.2 g/cc

*All dimensions in presentation
are typical of Watts Bar
(non-proprietary)*

12.73 ft
388.11 cm



Fuel Assembly

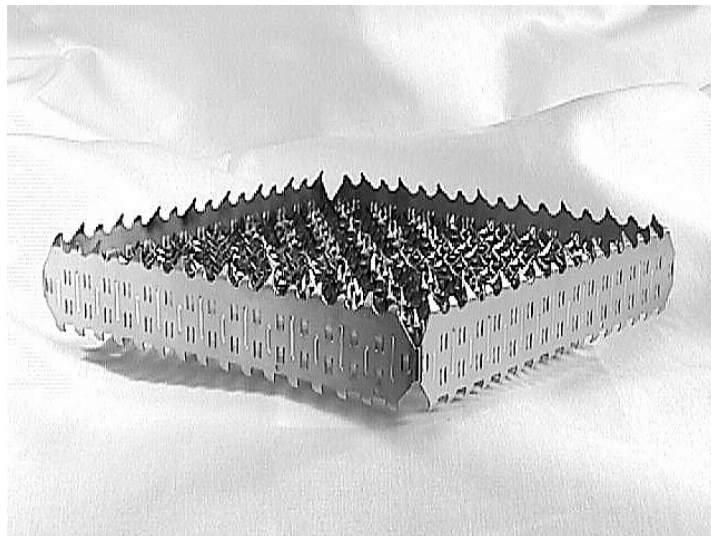
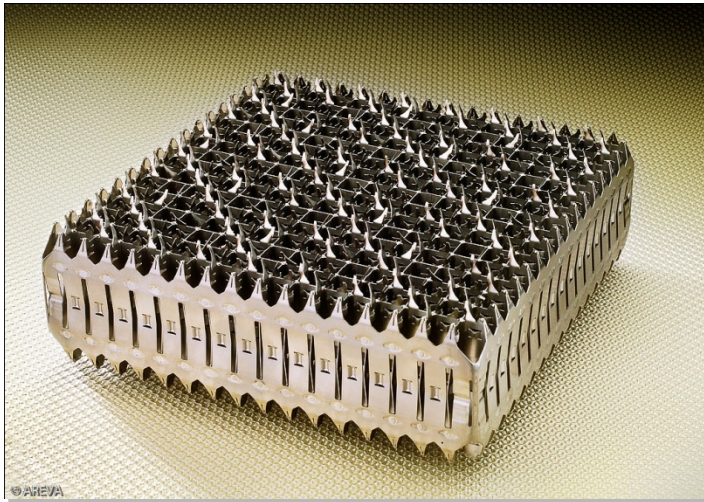


Fuel Assembly

Some idea of
relative size!



Spacer Grids



8 (or more) grids per assembly

Grid Materials:

- Inconel (top and bottom)
- Zirconium (middle)
- Combination for structure + springs

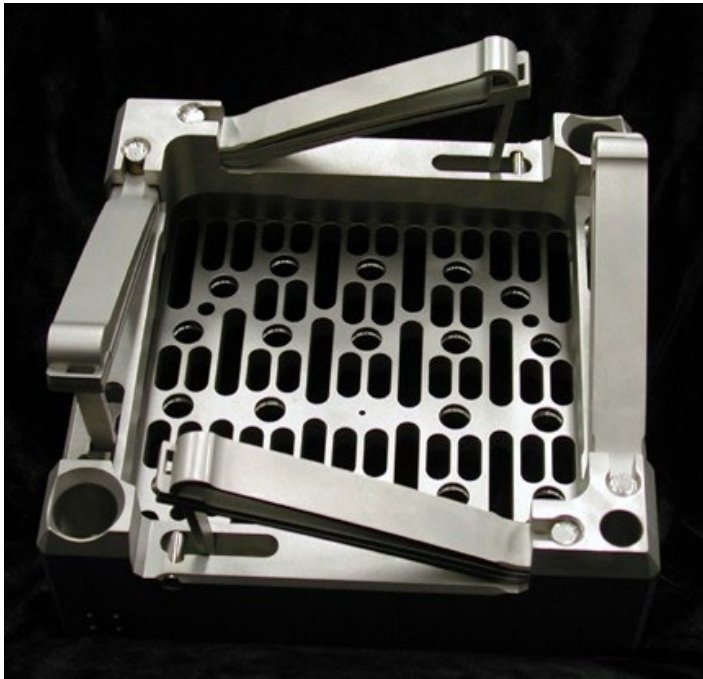
Axial Height 3.8 cm

Mass 875-1014 g

Notes on modeling grids:

- Neutron transport usually models as a smeared volume
- Subchannel T/H usually models with loss coefficients and mixing factors
- CFD can model explicitly using CAD drawings (or porous material?)

Top and Bottom Nozzle

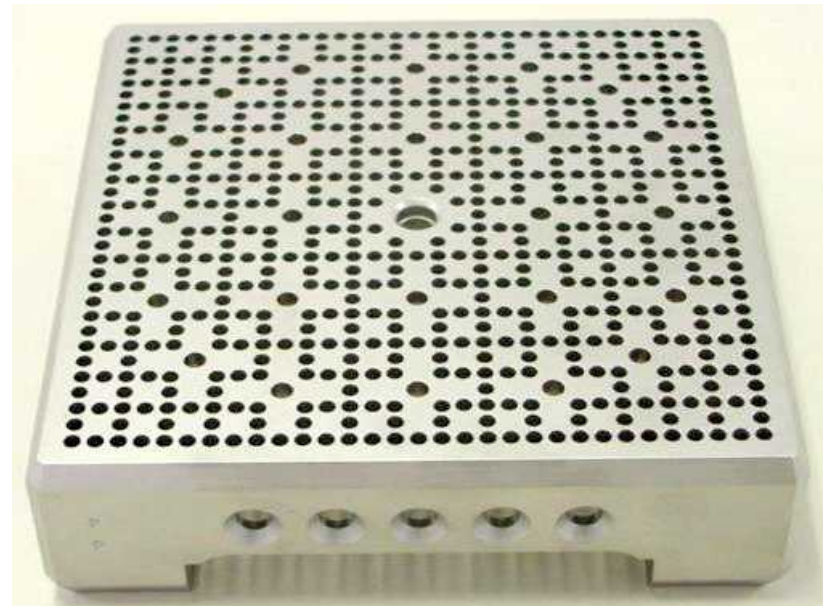


Materials: Stainless steel

Axial Height 6-9 cm

Mass 6250 g

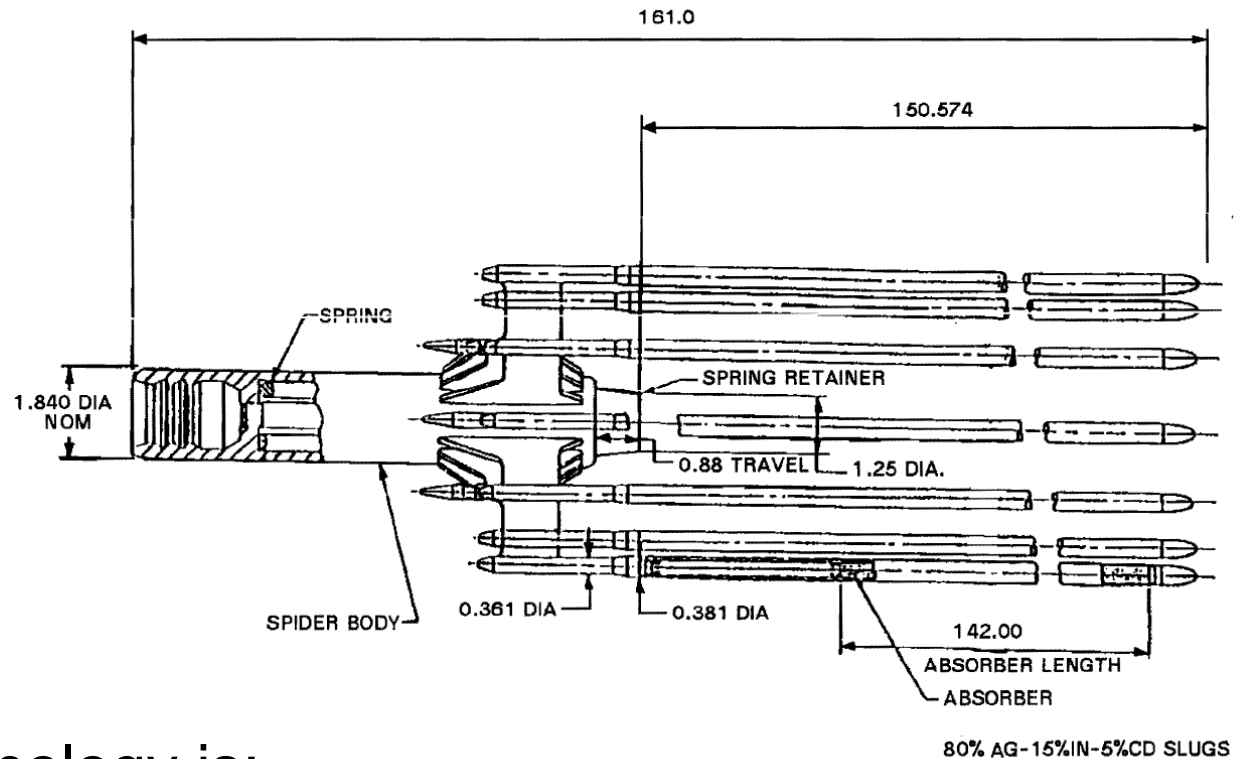
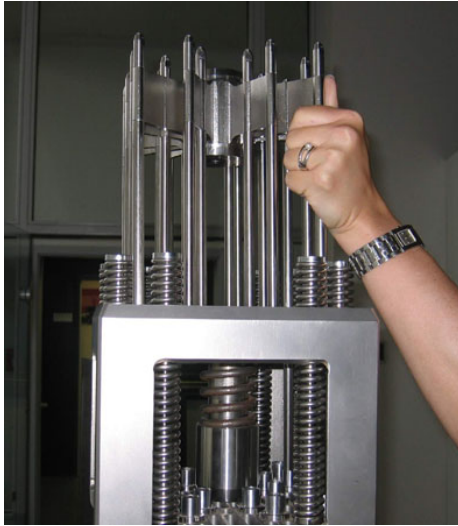
Modeled similar to spacer grids



Note on Reactivity Control

- As the core is operating, criticality must be maintained (fission source = absorptions + leakage)
- As the core depletes, the fission source term will decrease. Therefore, you must start out with “extra” source in the core at the beginning of life (BOL). This is “excess reactivity”.
- To counteract the excess reactivity at BOL, you must have extra absorptions at the BOL to match the excess reactivity and maintain criticality.
- As the source depletes, the extra absorptions must also deplete or be removed. This is done with one of the following:
 - Soluble boron in coolant
 - Control Rods (RCCA)
 - Integral Burnable Absorbers (IFBA, Gadolinia, Erbia)
 - Discrete Burnable Absorbers (Pyrex, WABA)

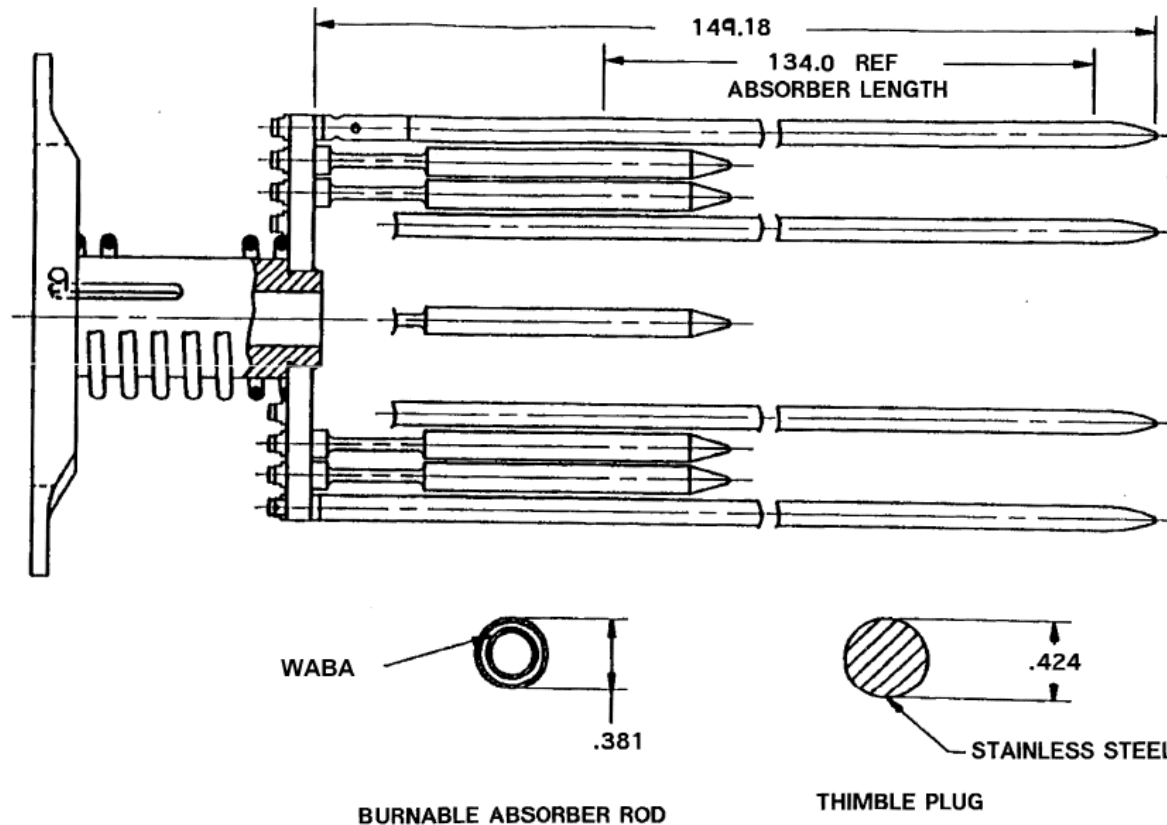
Control Rods



The correct terminology is:
Rod Cluster Control Assemblies (RCCA)

Control Rods move during operations

Discrete Burnable Absorber Assemblies



BA's can have different configurations of absorber rods (8, 12, 16, 20, 24)

BA's are usually Pyrex or WABA

BA's do not move during operation

Fuel Loading Pattern

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

Reg 1	2.10%
Reg 2	2.60%
Reg 3	3.10%

Watts Bar Unit 1 Cycle 1

- 3 enrichment zones
- No IFBA
- Pyrex

Burnable Absorber Loading Pattern

	R	P	N	M	L	K	J	H	G	F	E	D	C	B	A
1						8		12		8					
2					16		24		23s		16				
3			12	24		16		20		16		24	12		
4			24		20		20		20		20		24		
5		16		20		20		20		20		20		16	
6	8		16		20		24		24		20		16		8
7		24		20		24		20		24		20		24	
8	12		19s		20		20	+	20		20		19s		12
9		24		20		24		20		24		20		24	
10	8		16		20		24		24		20		16		8
11		16		20		20		20		20		20		16	
12			24		20		20		20		20		24		
13			12	24		16		20		16		24	12		
14					16		24		23s		16				
15						8		12		8					

Watts Bar Unit 1 Cycle 1

- 5 Pyrex Assembly Types
- +4 neutron sources

Note: large number of BA's
because all the fuel is fresh

RCCA Bank Positions

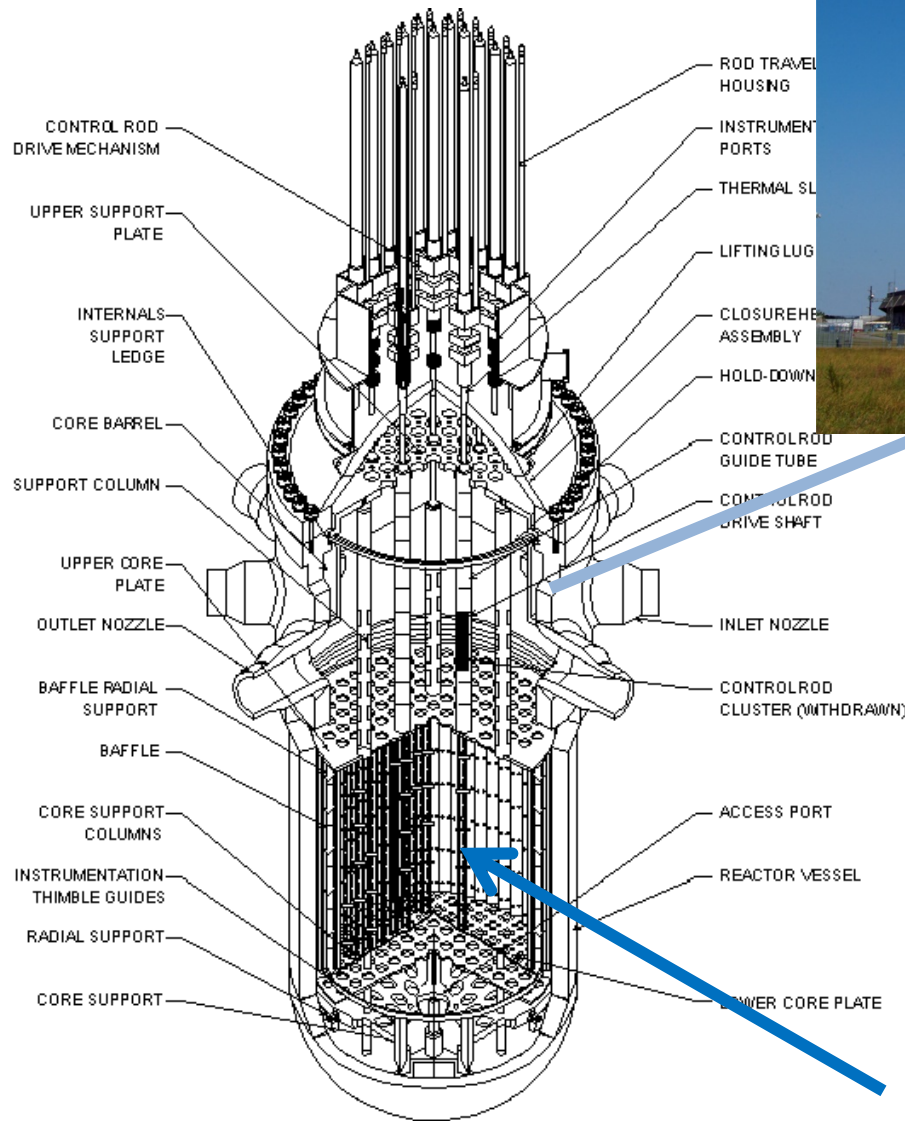
	R	P	N	M	L	K	J	H	G	F	E	D	C	B	A
1															
2				SA		B		C		B			SA		
3					SD		SB		SB		SC				
4		SA		D				D				D			SA
5			SC		A						A		SD		
6		B				C		A		C				B	
7			SB		SP								SB		
8		C		D		A		D		A		D		C	
9			SB										SB		
10		B				C		A		C				B	
11			SD		A						A		SC		
12		SA		D				D				D	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

Watts Bar Unit 1 Cycle 1

- Operational Banks A-D
- Shutdown Banks
- Operating Banks are symmetric

During normal operation, all rods are withdrawn and criticality maintained with soluble boron

PWR Vessel



TVA Watts Bar

Original CASL scope was the inside of the pressure vessel, but we do have ability to calculate detectors and fluence outside of core.

Reactor Core

VERA-Input

VERA-Input

Why do we have a common input?

- VERA is a “virtual environment” that is composed of many different computer codes, each with its own input
- It was recognized that users should not have to become familiar with the input of every code
- Another benefit of a common input is to reduce errors due to inconsistencies between code inputs

Virtual Environment for Reactor Applications (VERA)

Interoperability
with External
Components

ANC

STAR-CCM+

RELAP5

RELAP7

Others TBD

Geometry / Mesh /
Solution Transfer

DTK

libMesh

VERA

DAKOTA

MOOSE

Trilinos

PETSc

Solvers / Coupling /
SA / UQ

VeraIn/VeraOut

VERAView

Common Input/Output
And Visualization

ParaView

VisIt

Neutronics

MPACT

Shift

ORIGEN

Thermal-Hydraulics

CTF

Fuel Performance

BISON

Chemistry

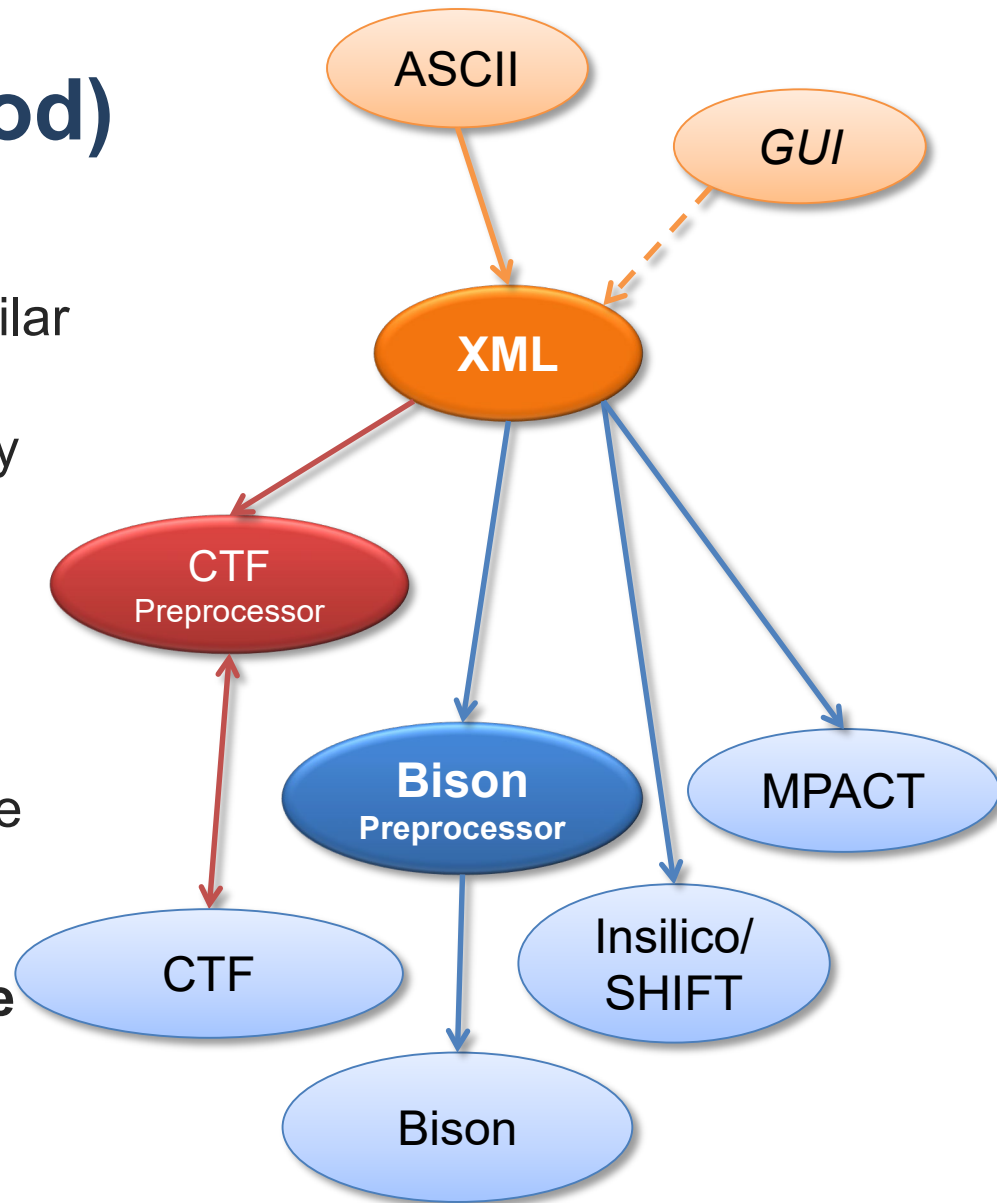
MAMBA

SCALE/
AMPX

Many different codes and inputs!

VERA-Input Internals (a peak under the hood)

- Input provides ability to create, archive, compare, and modify similar to current industry workflows
- Provide common reactor geometry for each physics components
 - assemblies, poisons, control rods, non-fuel structures, baffle, power, flow, depletion, etc.
- Reduce inconsistencies between coupled physics codes through the use of a common geometry description
- **Users only interact with a single ASCII input!**



Why ASCII?

- Provides simple, intuitive interface to build complex models
- Input is free format, uses minimum characters, and allows symmetry options
- Input is easy to edit on remote computers and move files back and forth between computer systems
- Provides archival format that can be used with version control and operating system access rules.

Students may want to open a copy of an input deck on their laptop to follow along ([p9.inp](#))

P9 input file also available on Github.com

https://github.com/CASL/VERAin/blob/master/verain/Progression_Problems/p9.inp

Input Manual

- The current VERA input manual (Rev 2) is include in the code distribution and can be downloaded from the CASL website.

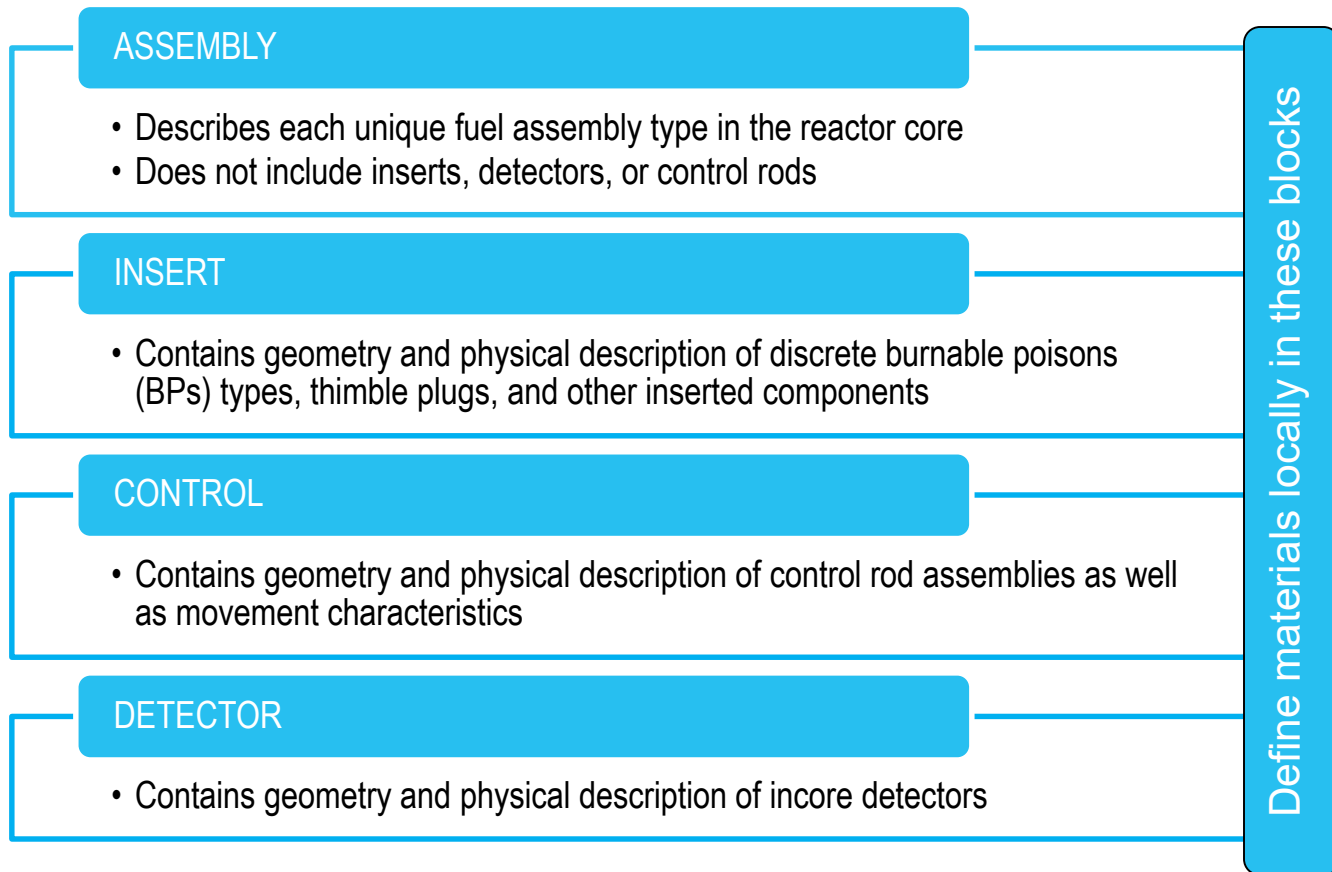
Palmtag, S., and A. Godfrey, “User Manual for the VERA Input Processor”, CASL Technical Report: CASL-U-2014-0014-002, February 28, 2015.

<https://www.casl.gov/sites/default/files/docs/CASL-U-2014-0014-002.pdf>

- The input manual mostly describes geometry options. Individual code options (i.e. MPACT options) are documented in the individual code manuals.

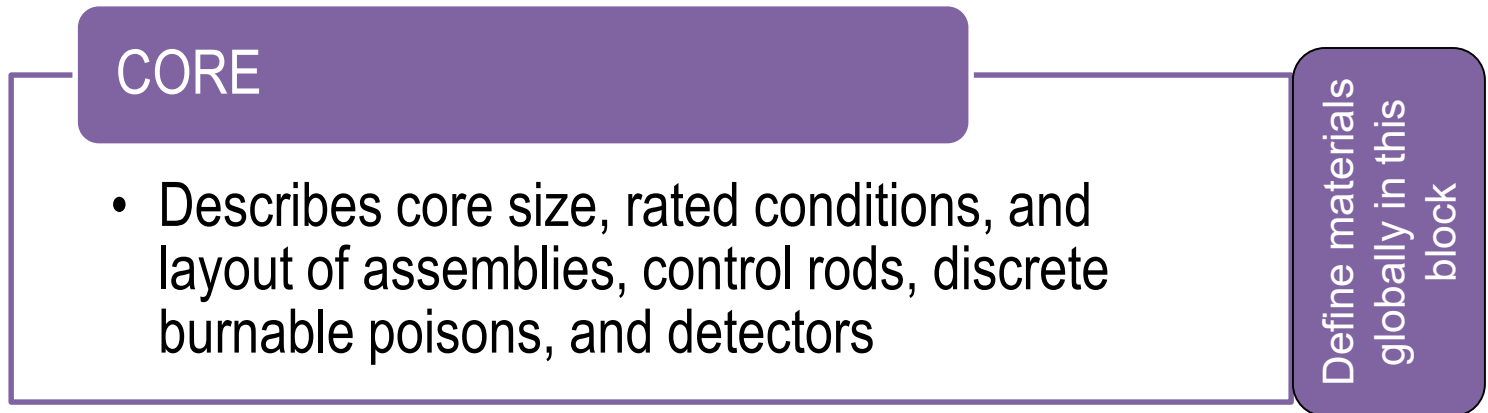
Input Blocks (Geometry Objects)

The VERA In Standard Input Deck is divided into several [BLOCKS] which align with reactor geometry objects



Input Blocks (Core)

- The [CORE] block lays out the geometry objects into the core (assemblies, detectors, inserts, etc.)
- The [CORE] block does not change during a cycle depletion



Input Blocks (Statepoints)

- [STATE] blocks define the current core conditions at a single point in time (power, rod position, inlet temperature, etc.)
- Multiple [STATE] blocks exist, one for each statepoint

STATE

- Describes core operating parameters for each statepoint in a simulation

Input Blocks (Code Options)

Each physics code may have a block to set code-specific options

EDITS

- Specifies axial levels at which power and/or neutronics-T/H coupling will be solved

MPACT

- Neutron transport solver options and other code specific options

COBRATF

- Subchannel thermal-hydraulics code options

What does this look like?

```
[ASSEMBLY]
..title "Westinghouse 17x17"
..npin 17
..ppitch 1.260
..
..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 zirc
..cell 2 ..... 0.4096 0.418 0.475 / U26 he zirc
..cell 3 ..... 0.4096 0.418 0.475 / U31 he zirc
..cell 4 ..... 0.561 0.602 / mod ..... zirc
..cell 5 ..... 0.418 0.475 / ..... he zirc
..
..lattice LAT21
.....4
.....1 1
.....1 1 1
.....4 1 1 4
.....1 1 1 1 1
.....1 1 1 1 1 4
.....4 1 1 4 1 1 1
.....1 1 1 1 1 1 1
.....1 1 1 1 1 1 1 1
```

```
[CORE]
..size 15
..rated 3411 131.68 .....
..apitch 21.5
..height 406.337
..
..assm_map
....1
....2 1
....1 2 1
....2 1 2 1
....1 2 1 2 2
....2 1 2 1 2 3
....1 3 1 3 3 3
....3 3 3 3
..
..crd_bank
....D....A....D....C...
....-...-...-SB...-
....A....C....-...B...
....-...-A...-SC....
....D....-...D...SA...
....-SB...SD....-
....C....B...SA...
....-...-...
```

```
[CASEID]
..title 'WBN1 Cycle 1'

[STATE]
..power 0.0 .....! %
..tinlet 557.33 .....! F - 565K
..tfuel 565.0 .....! K
..boron 1285 .....! ppmB
..modden 0.743 .....! g/cc
..feedback off .....
..sym qtr...

..rodbank SA 230
.....SB 230
.....SC 230
.....SD 230
.....A 230
.....B 230
.....C 230
.....D 167
```

- White space is ignored
- Comments start with exclamation point (!)
- Long input cards can be split over multiple lines

Geometry Concepts

We build up the assembly geometry from smallest to largest

1. Define material
2. Define cell
3. Define 2D lattice/segment
4. Define 3D assembly
5. Add grids and nozzle
6. Place assemblies in core

1. Define Materials

Structural Material

mat [user-name] [density (g/cc)] {[libname_i] [fraction_i], i=1, N}

```
mat he      0.000176  he-4
mat inc     8.19
mat gmat    8.0   zirc4 0.5   ss 0.5
mat zirc    6.56   zirc4 1.0
mat aic     10.20
mat pyrex   2.23
mat b4c     6.56
```

Fuel Material

fuel [user-name] [density] [th-den] / [U-235 enrichment]

```
fuel  U21 10.257 95.0 / 2.11
fuel  U26 10.257 95.0 / 2.60
fuel  U31 10.257 95.0 / 3.10
```

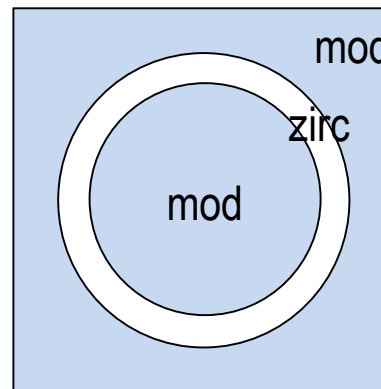
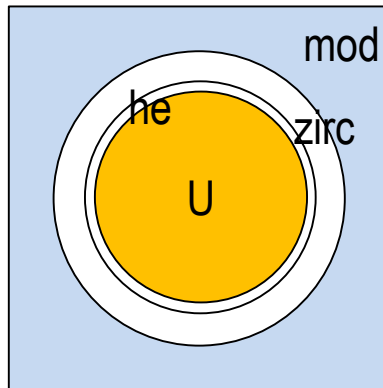
Additional options on fuel input to define gad and other additives

2. Define Cells

```
cell 1 0.4096 0.418 0.475 / U21 he zirc ! low enriched pin cell
cell 2 0.4096 0.418 0.475 / U26 he zirc ! med enriched pin cell
cell 3 0.4096 0.418 0.475 / U31 he zirc ! hi enriched pin cell

cell x      0.561 0.602 / mod zirc ! guide tube
cell 0      0.559 0.605 / mod zirc ! instrument tube

cell 8      0.418 0.475 / he zirc ! plenum pin
cell 9      0.418 0.475 / zirc zirc ! end plug
```



Outer material
defaults to “mod”,
which is determined
by T/H

Note that cell names are character strings!
“mod” has special significance for T/H coupling

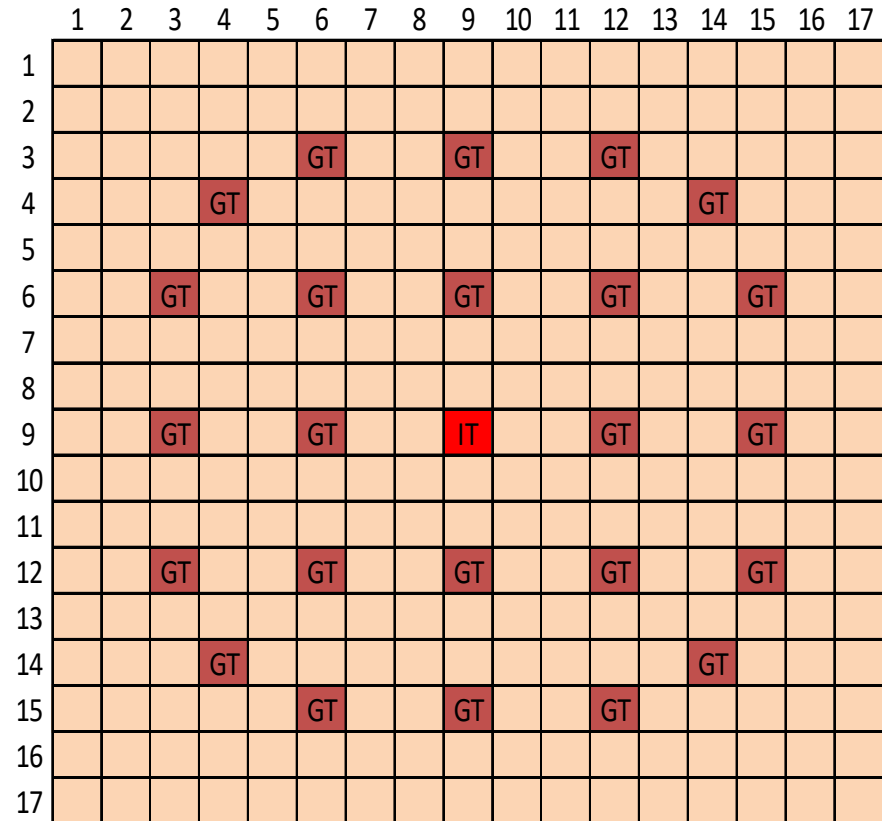
3. Define 2D Lattice/Segment

```
lattice FUEL1
O
1 1
1 1 1
X 1 1 X
1 1 1 1 1
1 1 1 1 1 X
X 1 1 X 1 1 1
1 1 1 1 1 1 1
1 1 1 1 1 1 1 1

! Numbers/labels in map
! refer to cell labels
```

Only need to define
octant (1/8) maps due
to symmetry

Define multiple lattices for each unique 2D slice



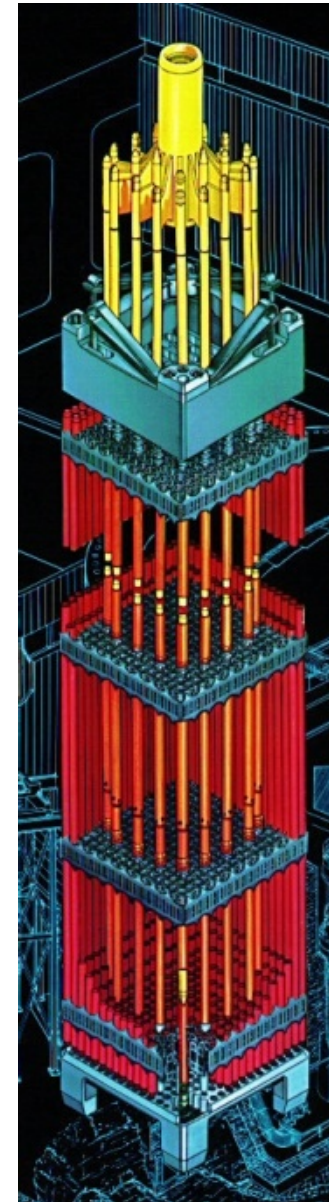
4. Define 3D Assembly

Stack up 2D lattices to form a 3D assembly

```
axial A1                ! Assembly label
                        6.050 ! Bottom elevation
      LGAP1  10.281      ! Lattice label, elevation
      PCAP1  11.951      ! etc.
      FUEL1  377.711
      PLEN1  393.711
      PCAP1  395.381
      LGAP1  397.501

! Lattice labels are defined on lattice maps
! Assembly label used in core maps
```

Assembly name “A1” will be used
in the core maps

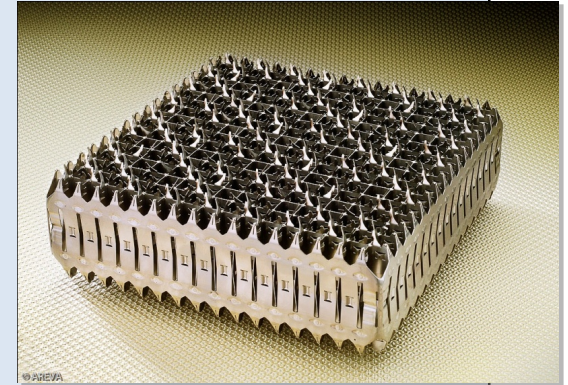


5. Add Grids and Nozzles

```
grid END inc      3.866 1017
grid MID zirc4    3.810 875      ! height (cm), mass (g)

grid_axial        ! Axial grid locations
  END 13.884      ! Grid midpoints (cm)
  MID 75.2
  MID 127.4
  MID 179.6
  MID 231.8
  MID 284.0
  MID 336.2
  END 388.2

lower_nozzle ss 6.05 6250.0 ! material, height (cm), mass (g)
upper_nozzle ss 8.827 6250.0 ! material, height (cm), mass (g)
```



* Format of grid card changed in VERA 3.6 and 3.7

6. Place Assemblies in Core

```
[CORE]
  assm_map
      A3 A3 A3 A3 A3 A3 A3
    A3 A3 A3 A1 A3 A1 A3 A1 A3 A3 A3
  A3 A3 A2 A1 A2 A1 A2 A1 A2 A1 A2 A3 A3
    A3 A2 A2 A2 A1 A2 A1 A2 A1 A2 A2 A2 A3
A3 A3 A1 A2 A1 A2 A1 A2 A1 A2 A1 A2 A1 A3 A3
A3 A1 A2 A1 A2 A1 A2 A1 A2 A1 A2 A1 A2 A1 A3
A3 A3 A1 A2 A1 A2 A1 A2 A1 A2 A1 A2 A1 A3 A3
A3 A1 A2 A1 A2 A1 A2 A1 A2 A1 A2 A1 A2 A1 A3
A3 A3 A1 A2 A1 A2 A1 A2 A1 A2 A1 A2 A1 A3 A3
A3 A1 A2 A1 A2 A1 A2 A1 A2 A1 A2 A1 A2 A1 A3
    A3 A2 A2 A2 A1 A2 A1 A2 A1 A2 A2 A2 A3
    A3 A3 A2 A1 A2 A1 A2 A1 A2 A1 A2 A3 A3
      A3 A3 A3 A1 A3 A1 A3 A1 A3 A3 A3
        A3 A3 A3 A3 A3 A3 A3
```

Additional core maps for RCCA's, RCCA Banks,
Inserts, and Detectors

[INSERTS] and [DETECTOR] Concepts

- INSERT and DETECTOR geometry is defined the same as assembly geometry
cells → 2D segments → 3D axial description → core map

```
[INSERT]
title "Pyrex"
npin 17
cell X 0.214 0.231 0.241 0.427 0.437 0.484 / he ss he pyrex-vera he ss

rodmap PY24
-
- -
- - -
X - - X
- - - - -
- - - - - X
X - - X - - -
- - - - - - -
- - - - - - - - - ! Dashes represent "empty" locations
```

Dashes are significant – represent “empty” location

[INSERT] locations

- Once Inserts are defined, they are loaded into the core like the assemblies were loaded

```
[CORE]
insert_map
-
20 TP
- 24 -
20 TP 20 -
- 20 TP 20 -
20 - 16 - 24 12
- 24 - 16 - TP
12 TP 8 TP
```

- User can use octant maps instead of full-core maps
- Insert names are user-defined strings that have been defined in [INSERT] blocks
- Dashes represent “empty” locations
- User defined insert names:
 - TP (Thimble plug)
 - 16 (16 pyrex rod insert)
 - 20 (20 pyrex rod insert)
 - 24 (24 pyrex rod insert)
 - Etc.

[CONTROL] Concepts

- RCCS geometry is defined same as assembly geometry
 - cells → 2D segments → 3D axial description → core map
- RCCS geometry is defined as “fully inserted”
- Total “stroke” is defined as distance from “fully inserted” to “fully withdrawn”
- Total number of notches is number of steps from fully inserted to fully withdrawn

Example: Stroke 360 cm, 228 total notches

- 228 notches is fully withdrawn
- 114 notches is inserted half-way
- 0 notches is fully inserted

[CONTROL] locations

- Once control rods are defined, we can load them into the core like assemblies and inserts
- Control rods need both a location and a bank name

```
[CORE]
crd_map
  1
  - -
  1 - 1
  - - - 1
  1 - - - 1
  - 1 - 1 - -
  1 - 1 - 1 -
  - - - -
crd_bank
  D - A - D - C -
  - - - - - SB - -
  A - C - - - B -
  - - - A - SC - -
  D - - - D - SA
  - SB - SD - - -
  C - B - SA -
  - - - -
```

- User can use octant maps and/or qtr-core maps
- Control names are user-defined strings that have been defined in [CONTROL] blocks (“1” in this case)
- Dashes represent “empty” locations
- User defined bank names

[STATE]

- STATE blocks define individual reactor “states” at a point in time
- Define variables that change during a depletion: power, inlet temperature, bank position, depletion step, etc.

```
[STATE]
  power 65.7
  tinlet 557.6 F
  rodbank D 192
  deplete EFPD 9.0
```

```
[STATE]
  power 99.7
  tinlet 558.1 F
  rodbank D 219
  deplete EFPD 32.0
```

- Can have as many [STATE] blocks as necessary
- Semicolons can be used to put more than one value on the same line – see P9 example (useful if copying input from Excel)

[MPACT/COBRATF] Code Option Blocks

- Each individual code has a “code block” to define code specific options.

```
[MPACT]
  scattering      TCP0
  ray_spacing     0.1
  azimuthals_octant 12
  k_tolerance     1e-5
  flux_tolerance  1e-4

  num_space      408
```

```
[COBRATF]
  beta_sp        0.005
  parallel        1
  proc_per_assem  4
```

- MPACT options are defined in MPACT User Manual (not VERA input manual)
- Recommend using default parameters for most values

Materials

- There are special rules when defining materials
- Materials can be defined in the CORE block
 - Materials have global “scope” throughout the input deck
- Materials can be defined in ASSEMBLY/INSERT/DETECTOR/CONTROL blocks
 - Materials have local “scope”, only for that particular block
 - Useful for cases where you might have two assemblies from different vendors, and each one has a slightly different “zirc” composition

Materials

- VERA-CS includes a large number of default materials for LWRs, mostly based on SCALE 6.2 compositions
 - air, aic, al2o3, b4c, boron, cs, gad, he, inc, pyrex, ss, water, zirc4. etc.
 - Defined in \$VERAHOME/bin/Init/CORE.ini
- Customers can define their own proprietary materials and bring them in as include files
 - One workflow is to define and verify material definitions in a separate file with read-only access. Individual users can then “include” the verified material definitions in their input

Include Files

- VERAIn allows include files to be used.

Examples:

```
include ctf_options.inc  
include /home/projects/Reactor/cycle12/assembly2.inc
```

- One workflow option is to:
 - Create input decks for individual assembly types (or control, etc.)
 - Verify inputs
 - Make the verified files read-only files on the system
 - Allow users to “include” the verified files in their inputs

Look at Example Input

Link to [Full Core Input file](#)

Link to [Full Core XML file](#) (optional)

Don't worry about all of the details now.
We will go through examples in the next
training sections!

Running Parser (Optional)

- VERA Input parser is in the VERA installation directory:

```
$VERAHOME/bin/react2xml.pl
```

- Run parser with command line:

```
react2xml.pl [file.inp] [file.xml]
```

- You can view the resulting xml file, but it is meant to be read by individual physics codes, not humans

Run Script will automatically run parser

Next

- The next training sessions will go through several examples in details
- Start with single-assembly input and move up to a full-core input

Questions?

