OAK RIDGE National Laboratory

# TRITON-NEWT model development

## Best practices, tips, and tricks

SCALE 6.3

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

# Learning goals

- Obtain modeling recommendations for TRITON-NEWT
  - Cross section library and cross section processing
  - NEWT geometry and neutron transport
  - TRITON depletion

OAK RIDGE National Laboratory

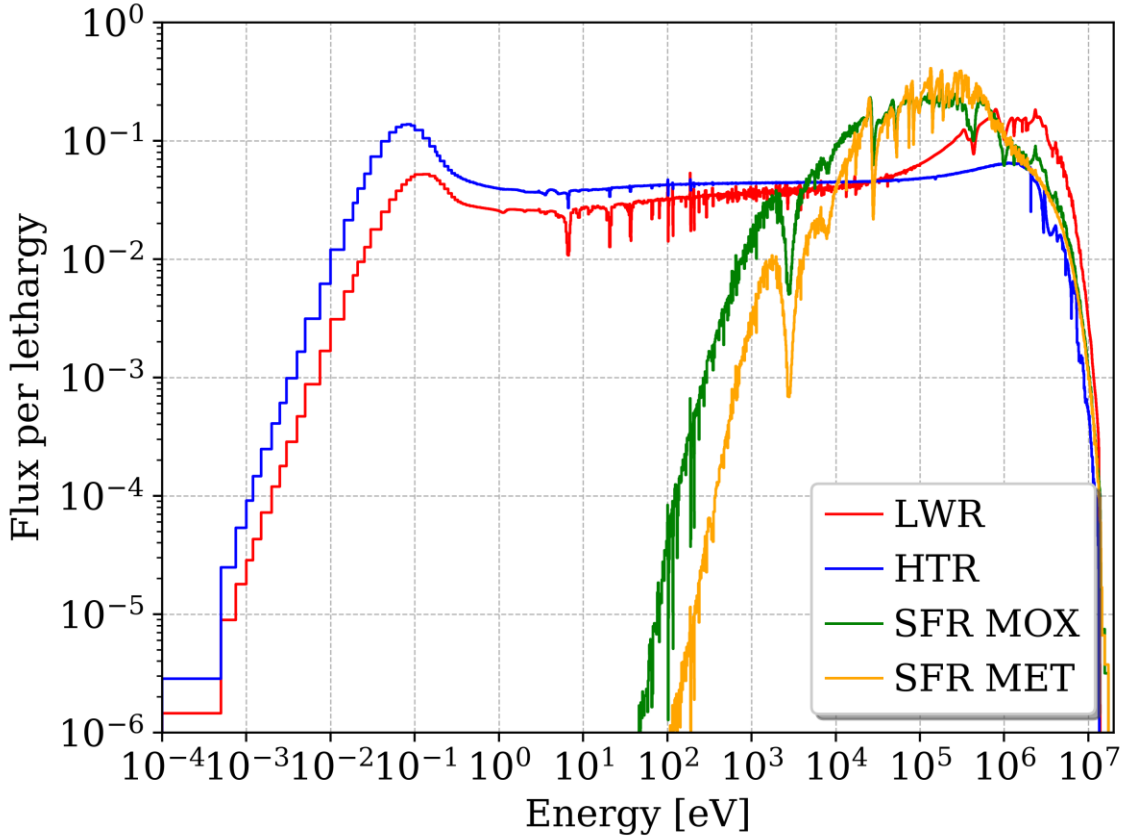
# Cross section libraries and cross section processing



# Cross section libraries

**Which multigroup cross section library should I use?**

- Depends on:
  - The spectral conditions of the model (thermal vs. fast)
  - The desired solution accuracy
  - The available simulation time
- For thermal systems and with limited applicability for other spectral conditions:
  - If time allows, use the 252-group ENDF/B-VII.1 (**v7-252**) library
  - If a quick answer is needed, use the 56-group ENDF/B-VII.1(**v7-56**) library
- Additional multigroup libraries:
  - Very fine group library with **1597 groups**
  - **302-group** library optimized for sodium-cooled fast reactor systems




Always compare MG calculations with CE to assess the bias due to cross section processing and choice of the multigroup cross section library

# Cross section processing

## Which cross section processing options should I use?

- The most accurate solution is obtained using CENTRM and a fine-group cross section library
- **Latticecell**: sufficient for most unit cells
  - Assumes that the fuel pin lies in an infinite lattice of other fuel pins – a good assumption in many cases
- **Multiregion**: to be used if unit cell needs to be divided in multiple rings or more complex scenarios
  - Example: Gadolinium-bearing fuel pins; gadolinium is a strong absorber of neutrons, so the flux shape across the fuel pin changes significantly
  - Example: Ag-In-Cd control rods
  - Example: Complex geometry not easily modeled using Latticecell
- **Doublehet**: to be used for double heterogeneous systems with fuel particles dispersed in a fuel component
  - Allowed fuel components are annular and solid rod, pebble, and slab
- All options allow for different lattice specifications: square, triangular (hexagonal), slab cell
- If approximation of infinite lattice of fuel pins is insufficient, modify the **Dancoff factor** for that fuel pin
  - Example: Edge and corner fuel pins in high-void BWR lattices
- If a unit cell geometry is widely different from repeated fuel lattices, try to create a multiregion with representative radii and perform thorough comparisons to continuous-energy calculations (see last slide)

# Cross section processing

## What are common input errors?

- Check that all necessary cell parameters are included: all radii, pitches, and heights
  - Remove optional ones that you don't need!
- Check that boundary conditions are correct (e.g., **`right_bdy=vacuum`** – most likely incorrect)
- Do not confuse pitches (pitch) and half pitches (**h**pitch) or radii (fuel**r**) and diameter (fuel**d**)
- For doublehet cells:
  - Be aware that coatings can be specified via radius, diameter, or thickness: coat**r**, coat**d**, coat**t**
  - Specify either the volume fraction of fuel particles (vf) **OR** the number of particles (numpar)
  - You need to define a doublehet mixture ID → don't confuse this ID with the fuel mixture ID from a possible CE model
- Triple-check material IDs in cell specifications, especially after a copy/paste from an old input
- "more data" and "centrm data" only apply to the unit cell specified immediately above the line

OAK RIDGE National Laboratory

# NEWT model development tips and tricks



# NEWT geometry construction

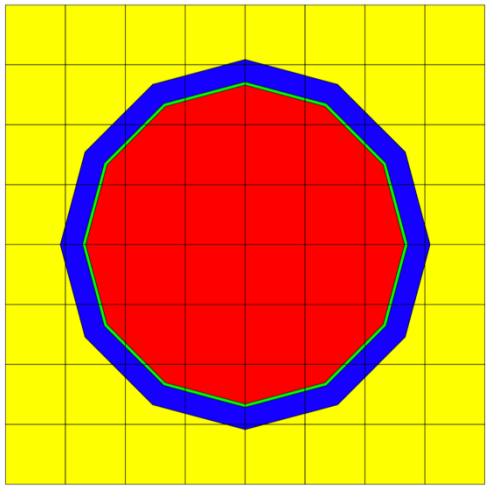
**Are there recommendations on how to set up my geometry?**

- **Use units**
  - You *could* specify everything you need in one global unit, but this can be very confusing
  - For each logical building block in the physical model, build a unit
  - Use arrays and holes to place all the different units into the global unit
  - Example: BWR control blades – if they are constructed as different units, you can simply comment them in or out of the global unit to insert or remove them
- We recommend giving the unit containing the **fuel unit cell** the same ID as the **fuel material** in that cell
  - Helps eliminate confusion
  - Makes lattice array more visibly intuitive in the input file

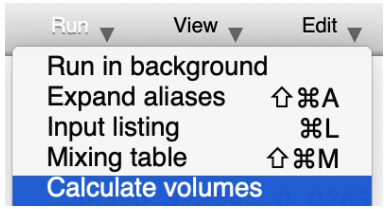
# NEWT geometry check

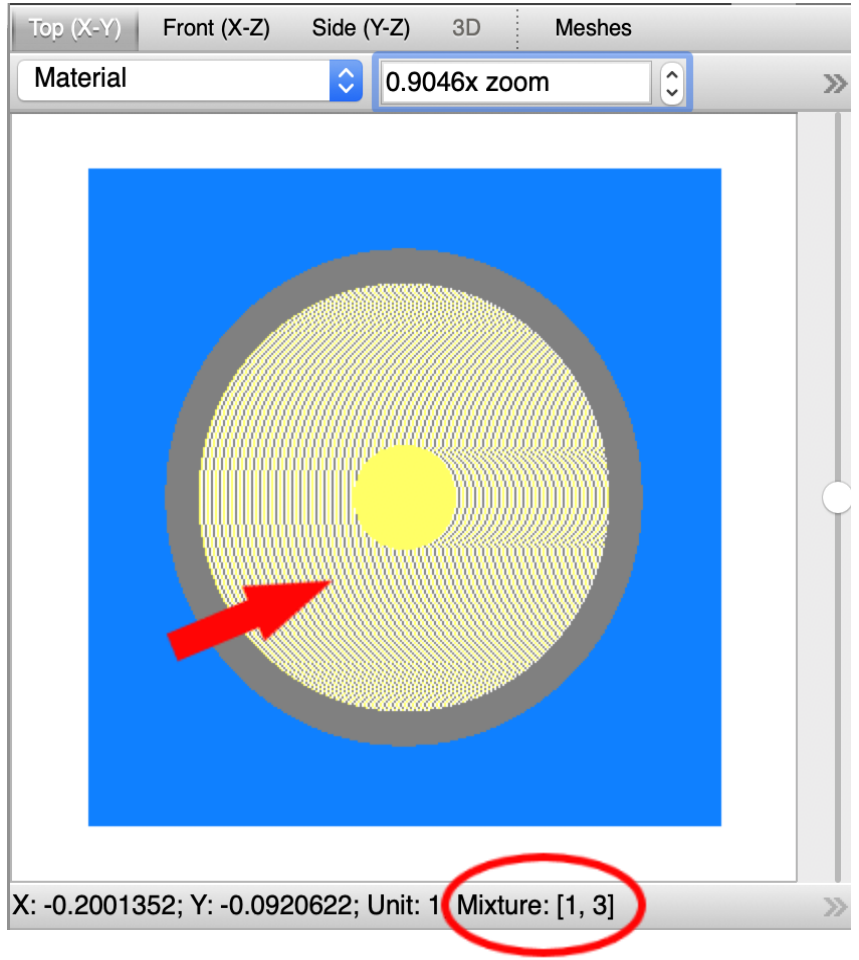
**How do I check that my geometry is correct?**

1. Use Fulcrum to check for input validation errors
2. Use Fulcrum to view model
3. Run TRITON-NEWT with *check* parameter:
   - **`=t-newt parm=(check)`**
   - This checks for so far undetected input errors
   - Can be used to plot the geometry in conjunction with **`drawit=yes`** in the NEWT parameter block
4. Also helpful: Use Fulcrum to calculate volumes via "*Run*"--> "*Calculate volumes*"




**NEWT geometry plot:**
${BASENAME}.newtmatl.ps




**Fulcrum view:**
Typical geometry error due to incorrect media placement, undetectable with the validation check

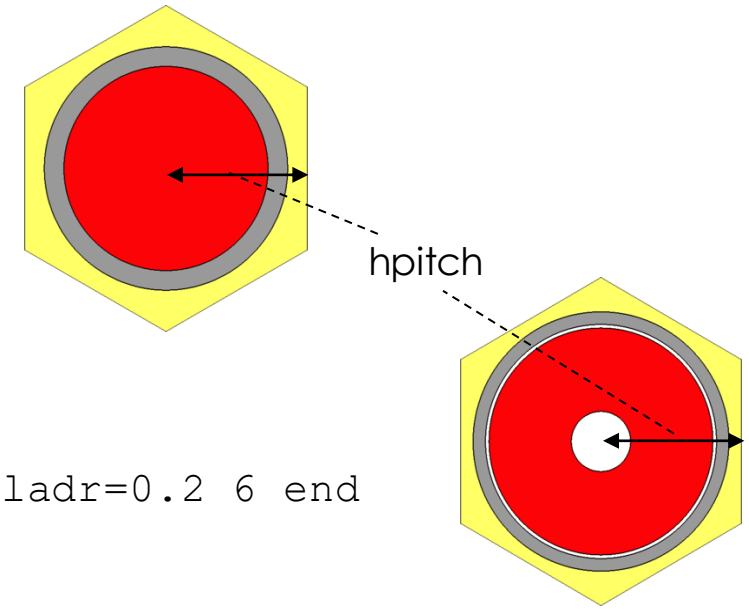
# Special case: NEWT hexagonal geometries

- Hexagonal geometries are used for VVERS as well as many advanced reactor concepts
- Unit cell level is straightforward:
  1. Define appropriate cell for self-shielding:

```
latticecell triangpitch hpitch=1.5 1
        fuelr=1.0 2 gapr=1.2 3 cladr=1.4 4 end

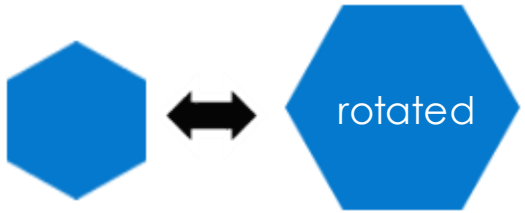
latticecell atriangpitch hpitch=1.5 1
        fuelr=1.0 2 gapr=1.2 3 cladr=1.4 4 igapr=0.1 5 icladr=0.2 6 end

doublehet: rod/pebble + (a)triangpitch
```




  2. Define appropriate geometry with consistent materials:

```
                hpitch
                  ↓
hexprism  1 1.0
rhexprism 2 2.0
```

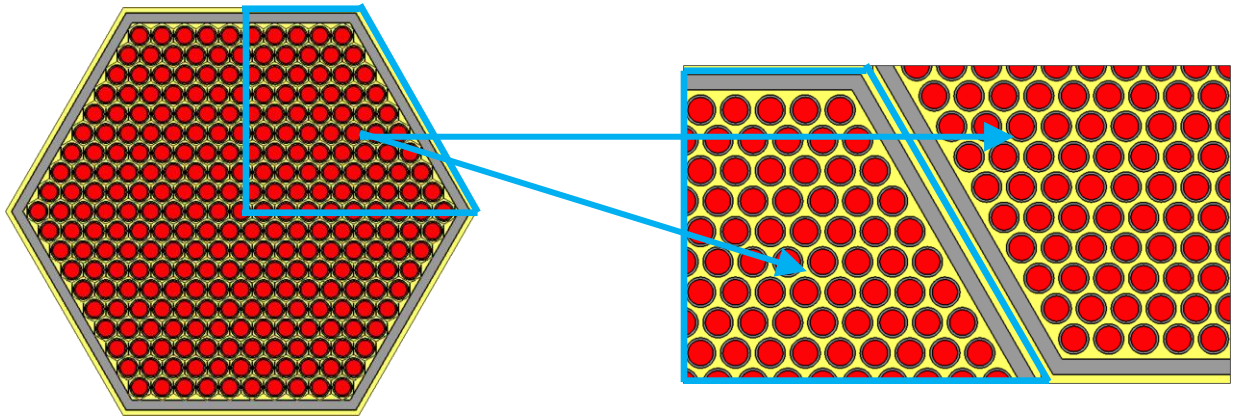


# Special case: NEWT hexagonal geometries

- Fuel assembly can be trickier, especially if there is an outer wrapper
- Definition of hexagonal array often leads to geometry errors
  → Issues visible in NEWT geometry plot
- Recommendation: explicitly place fuel pin units as **holes** in assembly
- Simplify geometry if possible: reflected assembly can be changed to a rectangular geometry

```
unit 1
cylinder 1 0.3236
cylinder 2 0.3857
media 1 1 1
media 2 1 -1 2
boundary 2
```

```
unit 10
rhexprism 111 7.50955
hole 1 origin x=0    y=0
hole 2 origin x=0.89 y=0
hole 1 origin x=0.44 y=0.77
hole 2 origin x=1.34 y=0.77
hole 1 origin x=0    y=1.55
hole 2 origin x=0.89 y=1.55
media 3 1 111
boundary 111 24 24
```

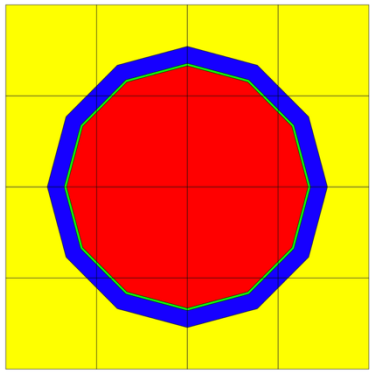


- Further recommendations for this type of geometry:
  - Only global grid
  - `cmfd=rect`
  - `grid_tol=1.0e-5`
  - `cell_tol=1.0e-8`
  - **Check NEWT plot of geometry**
  - Play with grid, npolar, nazim, etc. parameters to check whether your solution is converged

# NEWT transport settings

**What options should I use in my NEWT transport solution?**

- Currently we use and recommend the basic defaults for NEWT, which are a compromise between accuracy and runtime
- At least $S_6$ should be used for NEWT calculations
- $P_0$ in the gap; $P_1$ in fuel, clad, and structural materials; and $P_2$ in moderators
- Spatial mesh:
  - Spatial mesh of at least 4x4 in fuel cells
  - The global grid is sized such that each grid cell is approximately the size of one unit cell
  - Or global grid resulting in fine unit cell grid
  - Compare runtime: (global grid + unit cell grid + cmfd) vs. (global grid + cmfd)
- CMFD is performed on the global grid
- Epsilon = 1e-5



**Unit cell grid 4x4**

# NEWT model verification with Monte Carlo

**I'm doing lattice physics, why do I need Monte Carlo? ➔ To validate your assumptions!**

- **NEWT compared to KENO-MG or Shift-MG: Reveals the bias due to geometry approximations**

  ➔ Modify the number of grid cells, angular quadrature, number of sides on cylinders, etc.

- **KENO-MG/Shift-MG compared to KENO-CE/Shift-CE: Reveals the bias due to cross section processing and choice of the multigroup cross section library**

  ➔ Modify the cross-section library or cross-section processing options (Dancoff factors, CENTRM options)

- The model conversion is relatively easy because the NEWT and KENO-VI geometries are very similar

  ➔ For hexagonal assemblies, you can easily model a hexagonal array in KENO (i.e., without holes)

- Shift-CE allows for the generation of nodal data in t16 files

  ➔ This provides an additional great opportunity for verification with TRITON-NEWT

A KENO or Shift calculation can offer tremendous help in troubleshooting NEWT models and in verifying your results
→ We strongly recommend performing comparisons with KENO or Shift!

OAK RIDGE National Laboratory

# TRITON model development tips and tricks



# Tracking of nuclides on the neutron transport side

- Use an adequate addnux level: **parm=(addnux=N) with N=1,2,3,4**.
- Default addnux=2 is adequate for most cases
- Higher addnux values mean increased accuracy, but longer run time
- addnux=4 is desired for best-estimate benchmark models and when calculating a model different from LWRs (it's always good to start with addnux=4 and to then determine whether one can decrease the level)

# ALIAS block

- Aliases make the input file shorter and much easier to read; you can read a descriptive name for a material rather than just a number
- Aliases are especially useful when defining multiple fuel components
  - Example: multiple fuel pins that are supposed to be tracked individually during depletion
- Fulcrum offers an alias expansion, and expanded input is also printed in the output
- Note that CSAS does not permit aliases; only TRITON does
  - When converting from TRITON to CSAS, you can use the fully expanded input from Fulcrum or from the output file

# BURNDATA block

- The specific power density can be given in a few ways
  - The power density can be calculated from the total reactor power, the number of assemblies, and the mass of heavy metal in each assembly
  - It can also be calculated from the linear heat rate, fuel pin dimensions, and fuel density
- In TRITON, the power density must be provided in megawatts per metric ton of initial heavy metal (MW/MTIHM)
- For TRITON 3D calculations: If a full-core model is depleted, the total heavy metal mass (printed in TRITON output) and the specific power can be used to confirm the total reactor power

# BURNDATA block – Depletion steps

- Depletion steps determine the point in time when neutron transport calculations are run to update the 1-group cross sections used by ORIGEN
  - Users can use Fulcrum to plot 1-group fission and loss cross sections from F33 file
- To determine appropriate depletion steps and step lengths, a good approach is to double/triple/quadruple/… the number of depletion steps and to compare quantities of interest
  - Fine time steps at the beginning of depletion (~100 hours) are required to set xenon and samarium equilibrium concentrations
- Remember that "burn=0" can be used for transport-only calculations
- The TRITON "nlib" option can be used as shortcut to add additional depletion steps (neutron transport calculations and therefore 1-group cross section updates) per "burn" period

# Time stamp of results

- TRITON provides different results for different points in time due to the predictor–corrector approach:
  - **Nuclide inventory** as calculated by ORIGEN is available **at the end** of each depletion and decay step
  - **Neutron flux, $k_{eff}$,** and other quantities as calculated by the neutron transport solver are available **in the middle** of each depletion step
  - If neutron transport output is needed at the end of a depletion step or at the end of the irradiation history, users can add very small depletion intervals as needed (e.g., “burn=1e-5”).

# What are some common mistakes or issues?

- Typos in shell blocks lead to injection files not being found or files not being copied back from TMPDIR (e.g., for your own *arpdata.txt* file)
- Remember the data on ORIGEN concentration (F71) files are normalized to 1 MTIHM
  - Example: All fuel of a model is depleted – total (case=0) heavy metal mass on F71 is 1 MTIHM
  - Example: Depletion of only a few out of many fuel pebbles causes the total (case=0) heavy metal mass on the F71 file to be lower than 1 MTIHM because not all fuel mixtures end up on the file
- Common error when converting input from CSAS to TRITON: missing "read model" and "end model" block
- For TRITON 3D calculations: Incorrect volumes lead to incorrect mixture powers*
  - Calculate and explicitly define all possible volumes
  - Use volume block with tight settings for automatic volume calculations
  - Specific case: volumes applied in a unit refer to the total volume of that unit in a model

**With TRITON-NEWT, volumes are calculated analytically. The user doesn't provide volumes.*

OAK RIDGE National Laboratory

# Questions?



