Tutorial: Nodal Data with TRITON-Shift

B. Ade, F. Bostelmann
SCALE Users Group Workshop
April 27–29, 2022
Oak Ridge National Laboratory
Introduction

• Nodal data generation for use in nodal diffusion codes was so far enabled in SCALE through 2D multigroup calculations with Polaris or TRITON-NEWT

• New in SCALE 6.3: Capability to generate nodal data based on continuous-energy calculations of 3D geometries with the Monte Carlo code Shift

• Capability available with TRITON-Shift (T6-DEPL-Shift) using regular TRITON-KENO input with a new nodal data block (fgxs)

• Nodal data
  – Include few-group cross sections, scattering matrix, betas, lambdas, ADFs, etc.
  – Provided in arbitrary energy group structure
  – Enabled for individual cells and global meshes
  – Provided in SCALE’s standard t16 text files
Tutorial Goals and Structure

• Structure:
  – Background on nodal data
  – Methodology for nodal data in TRITON-Shift
  – Input format
  – Hands-on examples
  – Summary and outlook

• Goals:
  – Learn how to set up fgxs input
  – Understand output in out-file and t16-files
  – Understand limitations of current capability
Background
Nodal Data Basics: Lattice Physics and Nodal Simulator

Lattice Physics
- Fine-group pin-resolved simulations
- Isotopic depletion
- Pin power data

Point data
10,000s of energy groups

ENDF data

Heterogeneous cross sections
Homogenized cross sections

DOI: 10.5772/intechopen.79723

Nodal Simulator
- Full core simulation
- Pin power reconstruction
- Thermal feedback
- Steady-state and transient simulation

DOI: 10.5772/intechopen.79723

Functionalized over:
- Depletion states
- Temperatures
- Densities
- Control states
- Additional variables based on reactor being modeled

DOI: 10.5772/intechopen.79723
Nodal Data using SCALE

**ENDF data**
- Point data
- 10,000s of energy groups

**Cross section library generation**
- AMPX

**Resonance processing**
- XSProc

**Calculational libraries:**
- Continuous (point) data, multigroup: 10–100s of groups

**Data generation**
- Lattice code transport and depletion:
  - TRITON/Newt
  - Polaris
  - TRITON/Shift

**Other nodal simulators can be used, but formatting the nodal data is up to the user**

**CAMP**
- Advanced core simulator
- Neutron flux solver and depletion

**T/H code**
- TRACE

**Cross section library**
- PMAXS

**GenPMAXS**
- Few (2–8) group cross section database, parametric parameters (e.g., fuel/mod temp, mod dens)
Monte Carlo in Reactor Physics

- Compared to criticality safety and radiation shielding, Monte Carlo methods in reactor physics have special challenges.

- Need lots of tallies:
  - Reaction rates for many isotopes, many regions
  - Scalar flux and power everywhere
  - Uncertainty in transport solutions propagates into isotopes
  - Additional tally types and tally methods are still being developed (such as nodal data)

- Time-dependent calculation with branches (many state points) so potentially very expensive.

- Users need to ensure calculation convergence:
  - Sufficient number of active, inactive generations, total number of particles
  - Vary random seed to initialize random number generator
  - Analyze all outputs of interest for convergence
Nodal Tally Methodology
Path-length-based Nodal Tallies

- Shift path-length tallies are used to compute the following reaction rates in the user-energy grid and spatial regions:
  - non-fission absorption, i.e. capture
  - fission, nu-fission
  - non-fission transfer _1n, _2n, _3n, _4n
  - kappa-fission
  - flux

\[
RR^g_{\omega} = \int_D dV \int_{4\pi} d\Omega \int_{E^\omega} E^{\omega-1} dE \psi(E, \Omega) \Sigma^g_E(E)
\]

\[
(\phi^g V) = \int_D dV \int_{4\pi} d\Omega \int_{E^\omega} E^{\omega-1} dE \psi(E, \Omega)
\]

- TRITON converts the reaction rates to cross sections:
  \[
  \Sigma^g_\omega = \frac{RR^g_{\omega}}{\phi^g V}
  \]

- TRITON computes total, transfer, and total-transfer:

  \[
  \Sigma^g_t = \Sigma^g_c + \Sigma^g_f + \Sigma^g_{*_1n,0} + \Sigma^g_{*_2n,0} + \Sigma^g_{*_3n,0} + \Sigma^g_{*_4n,0}
  \]

  \[
  \Sigma^g_{X,0} = \Sigma^g_{*_1n,0} + 2\Sigma^g_{*_2n,0} + 3\Sigma^g_{*_3n,0} + 4\Sigma^g_{*_4n,0}
  \]

  \[
  \Sigma^g_t - \Sigma^g_{X,0} = \Sigma^g_c + \Sigma^g_f - \Sigma^g_{*_2n,0} - 2\Sigma^g_{*_3n,0} - 3\Sigma^g_{*_4n,0}
  \]
**T16 file**

- Path length tallies in red box
Collision-based Transfer Tallies

• Shift collision tallies are used to compute the transfer rates from one energy group to another

\[ R_{X,i}^{g \rightarrow g'} = \int dV \int_{4\pi} d\Omega \int_{4\pi} d\Omega' \int_{E^g}^{E^{g'-1}} dE \int_{E^{g'-1}}^{E^{g'-1}} dE' \, P_t(\bar{\Omega} \cdot \bar{\Omega}') \psi(E, \bar{\Omega}, \bar{r}) \sum_i [\sigma_{i,X}(E \rightarrow E', \bar{\Omega} \rightarrow \bar{\Omega}', T(\bar{r})) N_i(\bar{r})] \]

• (Note collision tallies are slower to converge than path-length tallies)

• TRITON computes relative transfer P0 matrix from the Shift collision tallies

\[ NRR_{X,0}^{g \rightarrow g'} = \frac{RR_{X,0}^{g \rightarrow g'}}{\sum_g RR_{X,0}^{g \rightarrow g'}} \]

• TRITON then multiplies the path-length transfer tally by the relative matrix to get the final P0 transfer matrix

\[ \Sigma_{X,0}^{g \rightarrow g'} = \Sigma_{X,0}^g NRR_{X,0}^{g \rightarrow g'} \]
T16 file

- Path length tallies in red box
- Transfer matrix tallies in the blue box
Outscatter Transport XS and Diffusion Coefficient

- The outscatter transport XS is defined as
  \[ \Sigma_{tr}^g = \Sigma_t^g - \bar{\mu}^g \Sigma_{X,0}^g \]
- “total minus mu-bar times transfer”
- Total and transfer are Shift path-length tallies
- Mu-bar, the mean transfer angle, can be derived from Shift collision tallies:
  \[ \bar{\mu}^g = \frac{\sum_{g'} R R_{X,1}^{g \rightarrow g'}}{\sum_{g'} R R_{X,0}^{g \rightarrow g'}} \]
- Diffusion Coefficient:
  \[ D^g = \frac{1}{3 \Sigma_{tr}^g} \]
- Verification tests show that this approach is poor for coarse group structures
- Solution: Compute \( D^g \) over fine group structure and collapse
- TRITON-Shift currently uses 1000 internally defined fine groups (equal-lethargy widths)
T16 file

- Path length tallies in red box
- Transfer matrix tallies in the blue box
- Diffusion tallies in green box
T16 file

- Path length tallies in red box
- Transfer matrix tallies in the blue box
- Diffusion tallies in green box
- Chi in orange box
- The energy distribution is tallied for all the fission neutrons born in the tally region
Nodal data generation with TRITON-Shift

• Prerequisites
  – Sequence name: =t6-depl-shift
  – Global unit boundary is **cuboid** or **rhex prism** (rotated hex prism)

```
global unit 100
  com="LWR assembly"
  cuboid 1 5 6 4 7 1 6
  cuboid 2 6 6 5 6 1 6
  media 4 1 2 1
  array 1 1 place 1 9 1 0.0 0.0 0.0
  boundary 2
```

```
global unit 100
  com="SFR assembly"
  rhex prism 110 7.50955 10 0
  rhex prism 120 7.90615 10 0
  rhex prism 130 8.12355 10 0
  array 1 110 place 13 13 1 0.0 0.0 0.0
  media 4 1 120 -110 vol=211.7905523
  media 3 1 130 -120 vol=120.71898
  boundary 130
```

Nodal data calculation in TRITON-Shift enabled through new FGXS block
The FGXS Block defines nodal data tallies

- FGXS block can be anywhere in the MODEL block
- Only one FGXS block is allowed
- Nodal tallies are organized through an integer id
- Multiple tallies are allowed (e.g., id=10, id=20, ...)

4 basic cards to define a nodal tally
  - **shape**: tally region shape
  - **energy**: tally energy grid
  - **tallyset**: tally type (only t16 supported)
  - **mesh**: tally over a mesh (optional)

```plaintext
read fgxs
  shape <type> id=10 <options>
  energy    id=10 1.0E-5 0.625 20E6 end
  tallyset t16 id=10
  ...
end fgxs
```
FGXS Block (shape)

read fgxs
    energy        id=1 1e-5 0.625 20E6 end
    tallyset t16 id=1
    shape cuboid id=1 5 -5 5 -5 2.5 0
end fgxs

read fgxs
    energy        id=2 1e-5 0.625 20E6 end
    tallyset t16 id=2
    shape rhex prism id=2 1.25 2.0 -2.0 origin x=0 y=0 z=0
end fgxs

• Calculation of nodal data for a single region
• syntax:
  shape <type> id=<ID> <options>
• <ID> must be a positive integer

• <type> must one of the following:
  – cuboid
  – rhex prism (rotated hexagonal prism)
  – global (used if mesh will be specified)
FGXS Block (mesh)

- Calculation of nodal data in mesh
- Syntax:
  ```
  mesh square|hexagonal id=<ID>
  hpitch=<P> origin x=<X> y=<Y>
  dz <Z1> ... <ZN> end
  ```
- `square` for square-pitched lattice, `hexagonal` for triangular-pitched lattice (note that this is a mesh of rotated hexagonal prisms)
- `<ID>`: must be a positive integer
- `<P>`: lattice half pitch (cm) used to define the mesh elements in X and Y
- `<X>` `<Y>`: coordinates in the global unit coordinate system in which to place the center of a lattice element. Lattice elements are repeated in the negative-x, negative-y, positive-x, and positive-y directions to fill the global unit.
- `<Z_i>`: relative axial mesh widths. (Bottom to top) The values must sum to 1. N entries will create N+1 mesh boundaries. The axial mesh boundaries is determined by the relative mesh widths and the bottom and top axial boundaries of the global unit.

```plaintext
read fgxs
  shape global id=10
  energy id=10 1E-5 0.625 20E6 end
  tallyset t16 id=10
  mesh square id=10 hpitch=0.63 origin x=0 y=0 dz 0.5 0.5 end
end fgxs
```
FGXS Block (mesh) – Example

... 
global unit 3
  cuboid 1 4p1.89 2p5
  array 1 1 place 2 2 1 3r0
  boundary 1

... 
read fgxs
  shape global id=10
  energy       id=10 1E-5 0.625 20E6 end
  tallyset t16 id=10
  mesh square id=10 hpitch=0.63 origin x=0 y=0 dz 0.3 0.7 end
end fgxs

- Cuboid 1 defines the global coordinate system with origin (0,0,0)
FGXS Block (mesh) – Example

... global unit 3
cuboid 1 4p1.89 2p5
array 1 1 place 2 2 1 3r0
boundary 1
...
read fgxs
shape global id=10
energy id=10 1E-5 0.625 20E6 end
tallyset t16 id=10
mesh square id=10 hpitch=0.63 origin x=0 y=0 dz 0.3 0.7 end
end fgxs

- Cuboid 1 defines the global coordinate system with origin $(0,0,0)$
- A mesh element is centered at $x=0$ $y=0$ with $hpitch=0.63$
FGXS Block (mesh) – Example

```
... global unit 3
  cuboid 1 4p1.89 2p5
  array 1 1 place 2 2 1 3r0
  boundary 1
...
read fgxs
  shape global id=10
  energy id=10 1E-5 0.625 20E6 end
  tallyset t16 id=10
  mesh square id=10 hpitch=0.63 origin x=0 y=0 dz 0.3 0.7 end
end fgxs

• Cuboid 1 defines the global coordinate system with origin (0,0,0)
• A mesh element is centered at x=0 y=0 with hpitch=0.63
• Additional mesh elements are added until reaching the global boundary
```
Nodal Tally Data Output

• A summary of the nodal tally user input is provided in regular out-file

• Nodal data is provided in t16 files:
  – Global nodal tally: ${BASENAME}.id10.t16
  – Mesh-based nodal tally: ${BASENAME}.id10-<I>-<J>-<K>.t16
    • I – goes from 1 to # of tallies in the x-direction
    • J – goes from 1 to # of tallies in the y-direction
    • K – goes from 1 to # of tallies in the z-direction
    • Number of mesh cells in each direction is computed from global boundary shape dimensions and mesh card input
    • Detailed description of I/J/K locations for hexagonal meshes printed in the output

• TRITON-Shift t16 files are in the same directory as the output file

• No =shell command is required in SCALE 6.3 to copy t16 files back
FGXS Block (mesh) – Example

```
... global unit 3  
cuboid 1 4p1.89 2p5  
array 1 1 place 2 2 1 3r0  
boundary 1  
... read fgxs  
  shape global id=10  
  energy id=10 1E-5 0.625 20E6 end  
  tallyset t16 id=10  
  mesh square id=10 hpitch=0.63 origin x=0 y=0 dz 0.3 0.7 end  
end fgxs  

• t16 file name id10-<I>-<J>-<K>  
• I and J indices shown on XY view  
• K index will be 1 for bottom 30% and 2 for top 70% of the global unit
```
FGXS Block (mesh) – Example

```plaintext
... global unit 3
cuboid 1 4p1.89 2p5
array 1 1 place 2 2 1 3r0
boundary 1
...
read fgxs
shape global id=10
energy id=10 1E-5 0.625 20E6 end
tallyset t16 id=10
mesh square id=10 hpitch=0.63 origin x=0 y=0 dz 0.3 0.7 end
end fgxs
```

- t16 file name id10-<I>-<J>-<K>
- I and J indices shown on XY view
- K index will be 1 for bottom 30% and 2 for top 70% of the global unit
Nodal Tally summary in the output file

read fgxs

shape global id=10
energy       id=10 1E-5 0.625 20E6 end
tallyset t16 id=10
end fgxs
Global Energy and Tally Settings

• Multiple different regions can be tallies in the same FGXS block using special “id=0”
  – Available for energy bins and tallysets

• Useful when nodal needs to be tallied over a nonuniform mesh

• Example: radial reflector

```plaintext
read fgxs
energy id=0 1e-5 0.625 20E6 end

tallyset t16 id=0
shape cuboid id=10 20.0 0.0 10.0 -10.0 1.0 0.0
shape cuboid id=20 39.0 20.0 10.0 -10.0 1.0 0.0
end fgxs
```
Comments and Recommendations
Comments and Recommendations

• Runtime can be long when requesting nodal data
  – Use Fulcrum validation when setting up input
  – Run short TRITON-Shift calculation without fgxs block to check input
  – Note that inactive generations can run fast, but active generations take longer

• Observe memory footprint, especially when requesting nodal data on a mesh for a complex geometry

• Statistical uncertainty of nodal data can be determined by running multiple calculations with different random seeds \( \text{rnd}=<\text{seed}> \)

• Volumes in all TRITON calculations:
  – If no volumes are specified and no volume block is found, a volume calculation with many rays is performed which can take a significant amount of time
  – Providing volumes via volume block or by specifying volumes for each media record \( \text{vol}=<\text{val}> \) saves runtime for each transport calculation

• Use “\text{burn=0}” in \text{burndata} block to request transport-only calculation (in that case, consider if you need addnux nuclides)

• Note that ADFs are provided as heterogeneous flux divided by homogeneous flux (NEWT ADF option 3)
Hands-on Exercises

1. Cuboid Shape
2. Square Mesh
3. Multiple shapes in one block
Exercise 1: Assembly-averaged nodal data

- Model: 17x17 PWR assembly
- Task: get assembly-averaged nodal data
- Starter file: `pwr_w17x17_starter.inp`
- Add FGXS block:
  - Tally ID
  - 2-group structure (1e-5 eV, 0.625 eV, 20 MeV)
  - Geometry: single cuboid shape
  - Tallyset t16
- Examine output:
  - Nodal data print in the out-file
  - How many t16 files were generated?

1/4th PWR assembly

Pin half pitch: 0.63 cm
Assembly half pitch: 10.75 cm
Exercise 1: Results

- FGXS block with individual shape **cuboid**
- Nodal data information in out-file starting with “Nodal FGXS Tally Summary”
- Generation of one t16 file `pwr_w17x17.id10.t16`
  - 4 burnup steps
  - 2 energy groups
- Examine xs, for example decreasing fission xs with depletion

```plaintext
read fgxs
  shape cuboid id=10 10.75 0.0 0.0 -10.75 1 0
  energy id=10 1E-5 0.625 20E6 end
  tallyset t16 id=10
end fgxs
```
Exercise 2: Pin-wise nodal data

• Model: 17x17 PWR assembly
• Task: get nodal data for each pin in the assembly only for the fresh fuel state (no depletion)
• Starter file: `pwr_w17x17_starter.inp`
• Add FGXS block:
  - Tally ID
  - 2-group structure (1e-5 eV, 0.625 eV, 20 MeV)
  - Geometry: square mesh (hpitch, origin), 1 axial zone
  - Tallyset t16
• Examine output:
  - Nodal data print in the out-file
  - How many t16 files were generated?
  - Use fission cross section to find non-fuel pins

---

Pin half pitch: 0.63 cm
Assembly half pitch: 10.75 cm
Exercise 2: Results

read fgxs
shape global id=10
energy id=10 1E-5 0.625 20E6 end
tallyset t16 id=10
mesh square id=10 hpitch=0.63 origin x=0 y=0 dz 1.0 end
end fgxs

• FGXS block with square mesh; origin in any pin center
• “Nodal FGXS Tally Summary” in out-file:
  volume mesh planes x:
  0,0.63,1.89,3.15,4.41,5.67,6.93,8.19,9.45,10.71,10.75
  volume mesh planes y:
  -10.75,-10.71,-9.45,-8.19,-6.93,-5.67,-4.41,-3.15,-1.89,-0.63,0
  volume mesh planes z: 0,1

• Generation of 100 t16 files
  pwr_w17x17_mesh.id10-<x>-<y>-<z>.t16
• Confirm fission xs is zero in water channels:
  (1,4,1), (4,4,1), (6,5,1), (7,7,1)

Extra mesh cells added because of outer gap!
Exercise 3: Nodal data from non-uniform assembly

- Model: 10x10 BWR assembly
- Task: nodal data for each quarter of the assembly
- Starter file: `bwr_assembly.inp`
- Add FGXS block:
  - Tally IDs
  - 2-group structure (1e-5 eV, 0.625 eV, 20 MeV)
  - Tallyset t16
  - Geometry: 4 cuboid shapes
  - Use “id=0” where applicable
- Examine output:
  - Nodal data print in the out-file
  - How many t16 files were generated?
  - Compare SW and NE: What would you expect?
- Further questions:
  - What is the best way to determine nodal data for each individual pin/channel in the assembly?
Exercise 3 Results

• Fission, transport, absorption, kappa*fission, and flux data post-processed from the t16 files. Quadrant power calculated using flux*kappa*fission

• Data provided as a ratio from the global XS data

• NE and SW corner compare well indicated reasonable statistics achieved. Your fast-running simulation results may differ due to statistics! This is especially true for tallies that use the collision estimator as they converge more slowly.

• Some quadrants different significantly from the assembly average data

• NW and SE quadrant yield different results:
  – Fission XS surprisingly higher in the SE corner due to Gd-rod layout
  – Thermal flux much lower in the SE compared to NW due to the non-uniform channel gap, which drive the quadrant power
  – Absorption much higher in the SE corner due to number of Gd rods
Summary and Outlook
Highlights of TRITON-Shift nodal data capability

• Nodal data generation with TRITON-Shift in continuous-energy mode for complex 3D geometries in SCALE 6.3

• “T6-DEPL-SHIFT” input consistent with T6-DEPL (TRITON/KENO) input, plus new nodal data “fgxs” block

• Nodal data generated for:
  – Individual cell: cuboid and rotated hexprism
  – Global mesh: Square or rotated hexagonal
  – Multiple cells/meshes in one input

• Tallies:
  – Arbitrary energy group structure
  – Calculation of typical nodal data: few-group cross sections, scattering matrix, kappa-fission, betas, lambdas, micro Xe/Sm absorption
  – ADFs for cartesian geometries

Questions:
scalehelp@ornl.gov
Outlook to Nodal Data in SCALE 7.0

- Nodal data in HDF5 format (or t16 text files)
- Statistical uncertainty of nodal data
- ADFs for both cartesian and hexagonal geometries
- Mesh types: cartesian, hexagonal, cylindrical
- Nodal data generation with Shift in both continuous-energy and multigroup mode
- Runtime performance improvements
- Nodal data within "tallies" block
  - Flexible control of energy grid, geometry, time array
  - Tallies block already supports flux and fission rate meshes in SCALE 6.3
  - Addition of infinite meshes “NET” and single shapes to definitions block