Serpent Research at Idaho National Laboratory

Leslie Kerby, Mark DeHart, Aaron Tumulak, Josh Rehak

International Serpent Users Group Meeting
September 26-30, 2016
Milan, Italy
INL-STIMS-

Goal:

Improve simulation of complex nuclear reactors, in particular the Transient Reactor Test Facility (TREAT) and the Advanced Test Reactor (ATR), both at Idaho National Laboratory (INL).

- Coupled Serpent—MOOSE/BISON/MAMMOTH
- Functional Expansion Tallies
- Weighted Delta Tracking

Idaho National Laboratory

Coupled Serpent—MOOSE

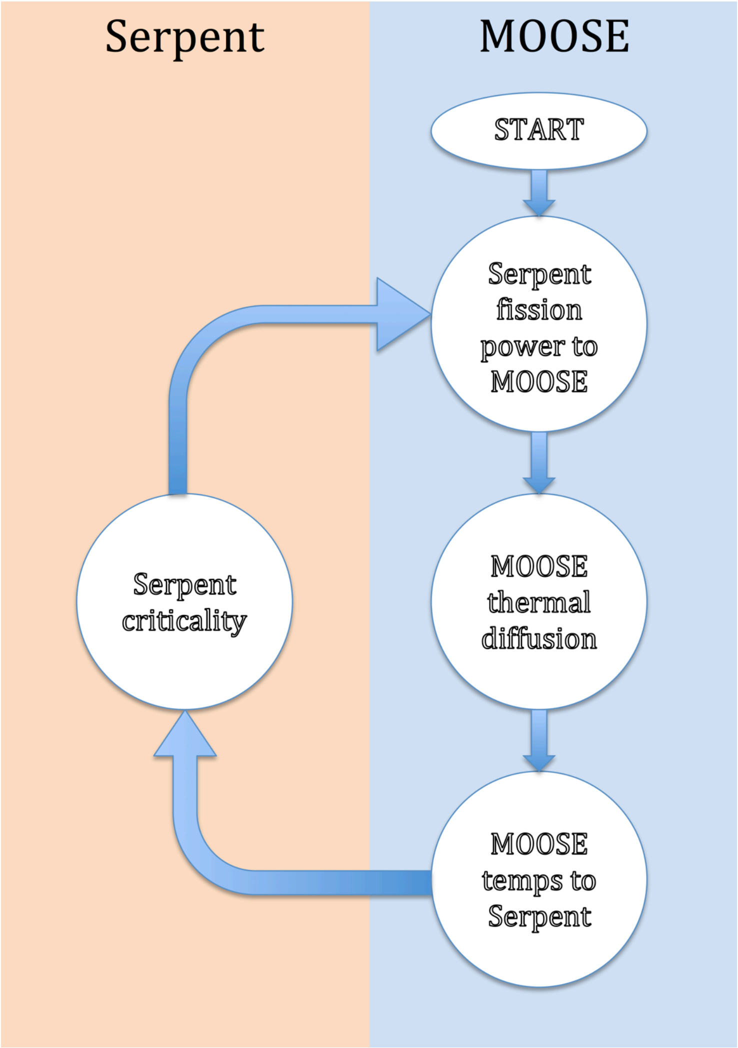
Built off of prototype provided by Ville Valtavirta

- HeatToMoose: transfers the Serpent fission heat generated, per element volume, to the MOOSE mesh. Simulation must start with initial "guess," which can be accomplished by running standalone Serpent.
- ElementTransfer: averages the MOOSE temperature solution field for each element, transforms this to an OpenFOAM mesh, and transfers this to the Serpent interface input file.
- RunSerpent: runs the Serpent calculation and updates the fission heat distributions.

Idaho National Laboratory

Coupled Serpent—MOOSE II

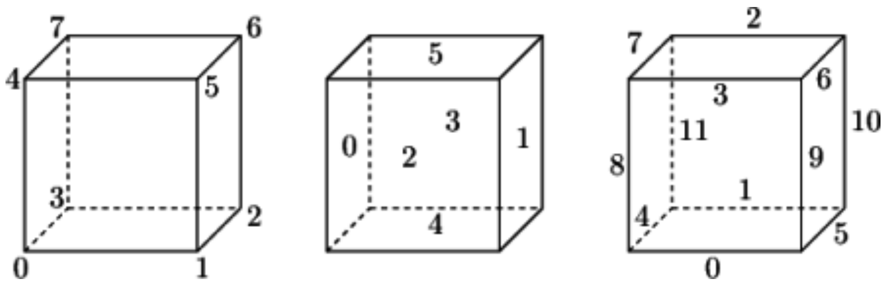
Process



Coupled Serpent—MOOSE III

Meshes

Serpent

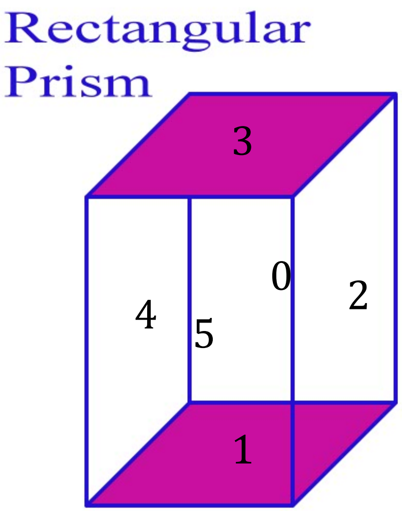


OpenFOAM hexahedron vertices, faces, and edge numbering

Four separate files:

- points
- faces
- owners
- neighbors

MOOSE



GeneratedMesh with dim=3 and elem_type=PRISM6, with sides numbered

Coupled Serpent—MOOSE IV

Code Modification

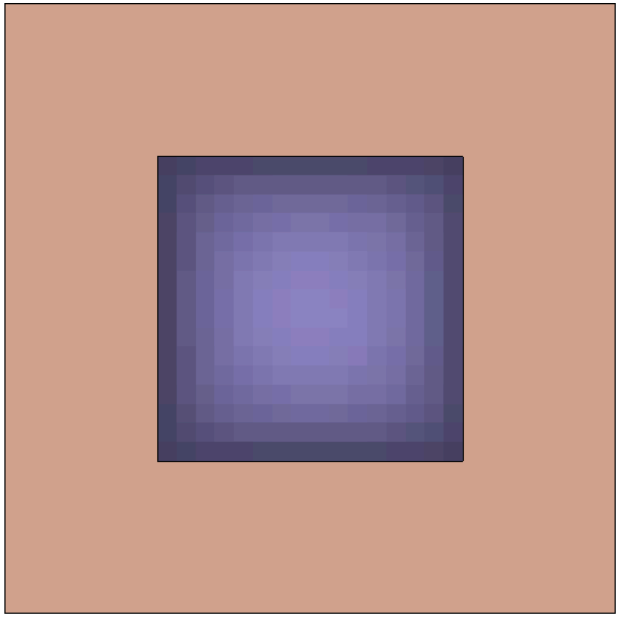
- Protect header files from multiple definitions;
- Extern global variables declared in header files;
- Create C file to define those global variables;
- Change main to cmain;
- #define OPEN MP and NO GFX MODE;
- Create DATA EXT MODE option;
- Add extern clause to all *.c and *.h files;
- Fix Newton's method bug in TMS.

Compiling Options

- Makefile
- In src/ directory

Coupled Serpent—MOOSE V

Single Fuel Pin



Fuel		Water	
Isotope	% mass	Isotope	%mass
^{235}U	2.9971	^{1}H	66.6667
^{238}U	85.153	^{16}O	33.3333
^{16}O	11.85		
5 cm^2 square		10 cm^2 square	

Idaho National Laboratory

Coupled Serpent—MOOSE VI

Single Fuel Pin Results

	Standalone Serpent 2.1.26	Coupled Serpent–MOOSE
k_{eff} (analog)	0.20592 +/- 0.00080	0.20535 +/- 0.00085
k_{eff} (implicit)	0.20598 +/- 0.00060	0.20542 +/- 0.00061
Transport comp. time	20.9	20.3

Functional Expansion Tallies

What are Functional Expansion Tallies (FETs)?

- Typically, Monte Carlo codes score tallies in spatial bins arranged in some type of regular mesh.
- Rather than score tallies in a finite number of bins, FETs score the contribution of each tally to a handful coefficients which correspond to continuous basis functions.

Functional Expansion Tallies II

Why implement FETs in Serpent?

- Structured to unstructured mesh conversion
- Only have to store a handful of coefficients
- Data transfer speed up
- Implemented successfully in OpenMC
- Continuous!

Functional Expansion Tallies III

Legendre Polynomials

The Legendre polynomials P_i are defined for integers $i \geq 0$ as

$$P_i(z) = \sqrt{\frac{2i+1}{2}} \sum_{k=0}^{i} \binom{i}{k} \binom{-i-1}{k} \left(\frac{1-z}{2}\right)^k \tag{1}$$

where the first two factors in parenthesis are binomial coefficients and the last factor is a real number.

Functional Expansion Tallies IV

Zernike Polynomials

The Zernike polynomials $Z_j = Z_n^m$ are implemented for even $n - m$ and $n \geq m$ as

$$Z_n^m(r,\theta) = \begin{cases} \sqrt{2(n+1)}R_n^m(r)\cos(m\theta) & \text{for } m > 0 \\ \sqrt{2(n+1)}R_n^{-m}(r)\sin(-m\theta) & \text{for } m < 0 \\ \sqrt{n+1}R_n^0(r) & \text{for } m = 0 \end{cases} \tag{2}$$

$$R_n^m(r) = \sum_{k=0}^{\frac{n-m}{2}} (-1)^k \binom{n-k}{k} \binom{n-2k}{\frac{n-m}{2}-k} r^{n-2k}$$

where the second and third factors in parenthesis are binomial coefficients.

Functional Expansion Tallies V

Calculating the coefficients

From the orthogonality of Zernike and Legendre polynomials, the product $Z_j(r,\theta)P_i(z)$ satisfies

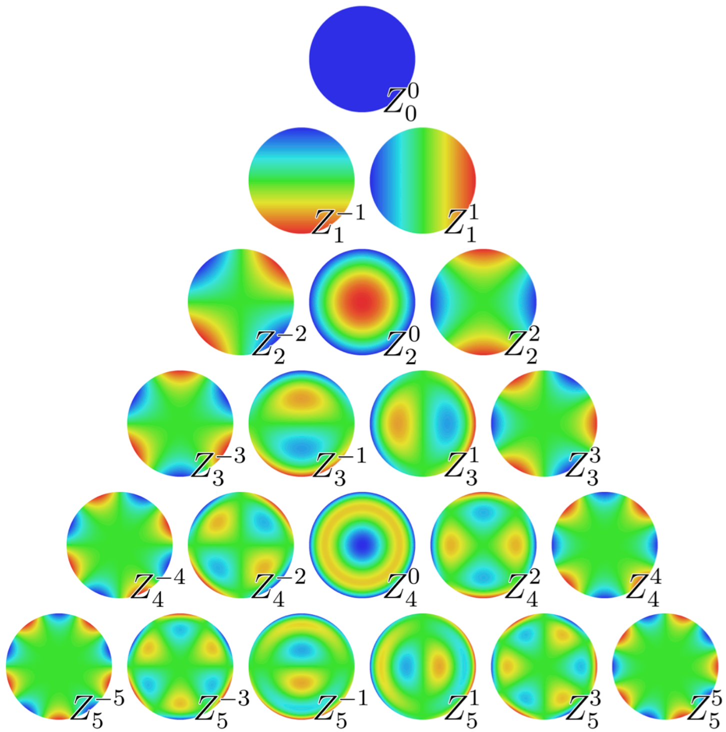
$$\int_{-1}^{1} dz \int_{0}^{1} dr \int_{0}^{2\pi} d\theta \Big(Z_j(r,\theta)P_i(z)\Big)\Big(Z_{j'}(r,\theta)P_{i'}(z)\Big) = \delta_{i,i'}\delta_{j,j'} \tag{3}$$

where δ is the Kronecker delta function. Note that the forms of the polynomials defined in Eqs. [2] and [1] are normalized so that their inner products are one. The constants c_{ij} can then be obtained from

$$c_{ij} = \int_{-1}^{1} dz \int_{0}^{1} dr \int_{0}^{2\pi} d\theta \; f(r,\theta,z)Z_j(r,\theta)P_i(z) \tag{4}$$

Functional Expansion Tallies VI

First 15 Zernike Polynomials



Functional Expansion Tallies VII

Multiphysics Interface

- New to Serpent 2
- Standardized input and output file formats for reading in temperature/data and writing fission power
- Support for regular mesh, tetrahedral mesh, unstructured mesh, and more
- Type 3 interface 'User Defined Functional Dependence', allows a user to pass in an arbitrary number of parameters and modify userifc.c, the source file that interprets the parameters

Idaho National Laboratory

Functional Expansion Tallies VIII

Reading temperature and density data

```
function UserIFC(*f, x, y, z, nz, nr, c_{0,1}, ..., c_ij, ..., c_{nz-1, nr})
    *f = 0
    i = 0
    for i < nz do
        j = 1
        for j ≤ nr do
            *f = *f + c_ij × Z_j(r(x, y), θ(x, y)) × P_i(z)
        end for
    end for
end function
```

Outline of UserIFC() function using *nz* Legendre polynomials and *nr* Zernike polynomials. Note that *f is a pointer to either temperature or density at location (x,y,z).

Functional Expansion Tallies IX

Writing fission power data

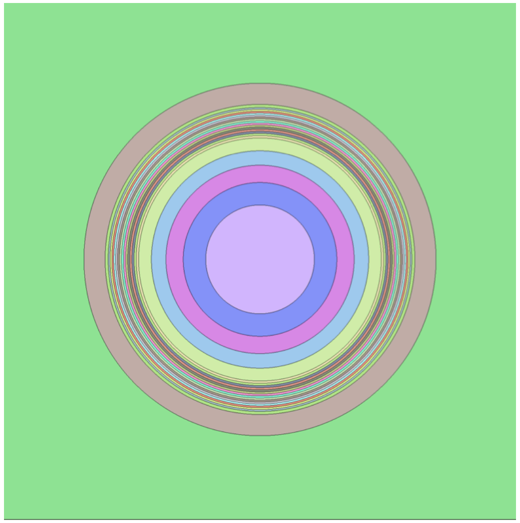
```
function ScoreInterfacePower(E, w, nz, nr, x, y, z)
    i = 0
    for i < nz do
        j = 1
        for j ≤ nr do
            E = E × Z_j(r(x, y), θ(x, y)) × P_i(z)
            AddScoreToBin(E, w, i, j)
        end for
    end for
end function
```

Modified ScoreInterfacePower() function using *nz* Legendre polynomials and *nr* Zernike polynomials. The coefficients are stored in nz × nr bins.

Functional Expansion Tallies X

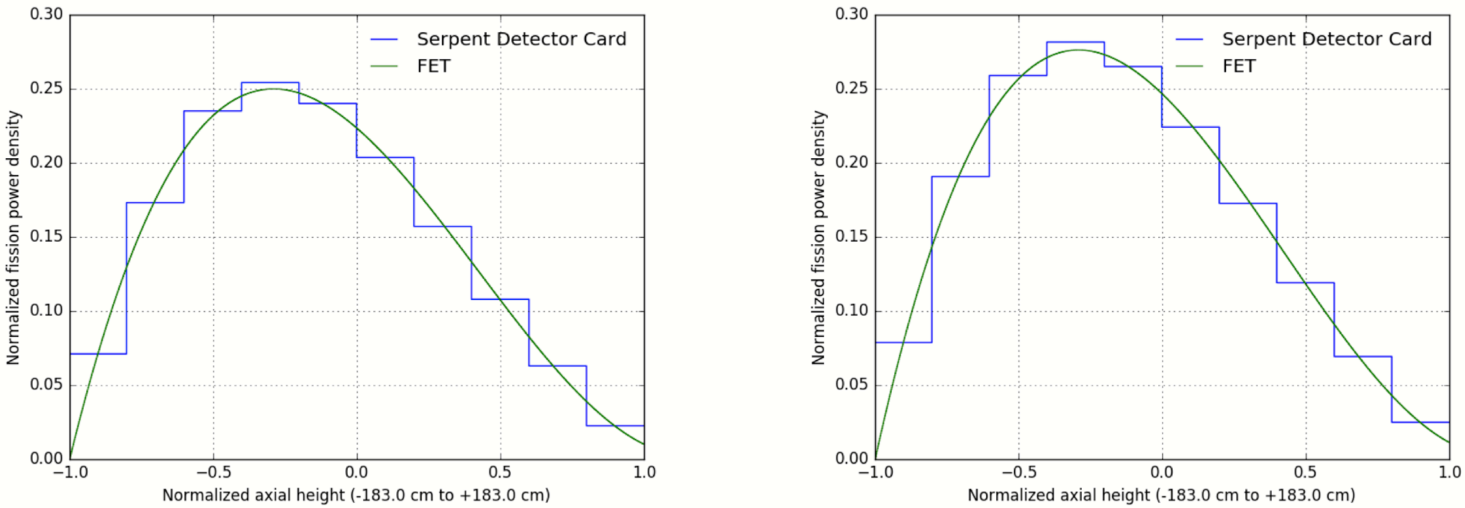
Serpent Detector Cards

The fuel pin was partitioned into 10 equal-width axial zones and 20 radial zones. The functional expansion tally was carried out to 5th order Legendre and 4th order Zernike polynomials.



Functional Expansion Tallies XI

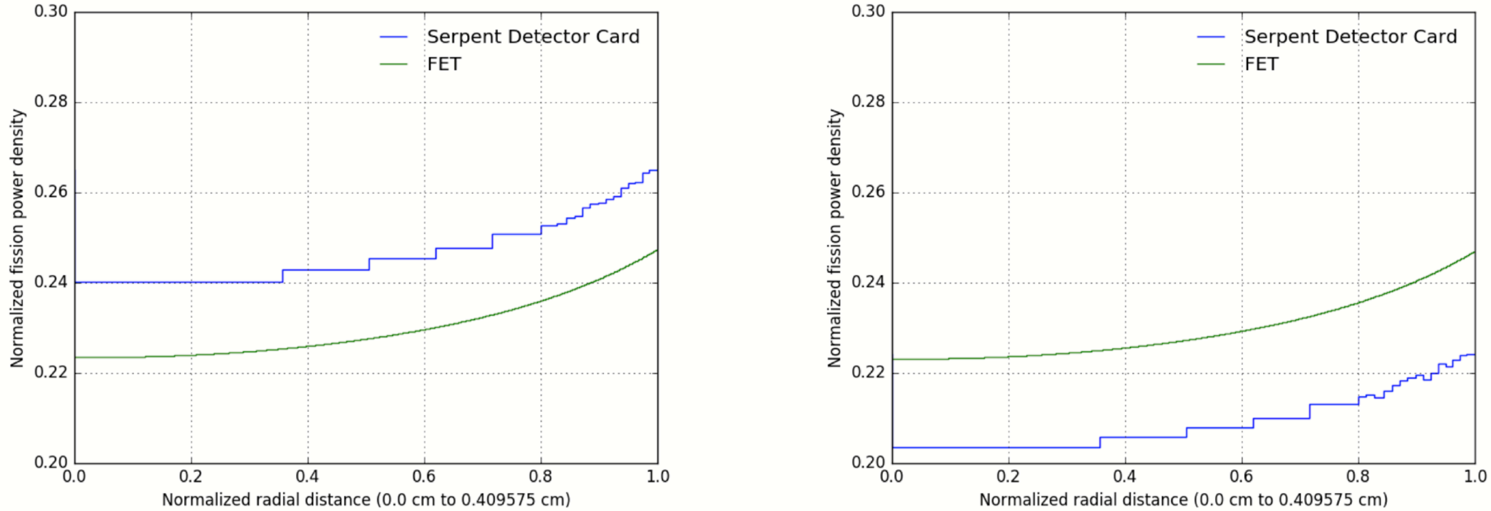
Axial Fission Power Distribution Comparison



(Left) Fission power density along centerline of fuel pin. (Right) Fission power density along outer edge of fuel pin.

Functional Expansion Tallies XII

Radial Fission Power Distribution Comparison



Fission power density at axial midpoint of fuel pin. (Left) Immediately before detector discontinuity. (Right) Immediately after detector discontinuity.

Weighted Delta Tracking

Woodcock Delta Tracking

Choose $g(x)$ using the **majorant cross-section**, the maximum cross-section for our region of interest $\mathbb{V}$.

$$\Sigma_{\mathrm{maj}} \equiv \max_{\mathbf{r}\in\mathbb{V}}\{\Sigma_t(\mathbf{r})\}$$

Sample from

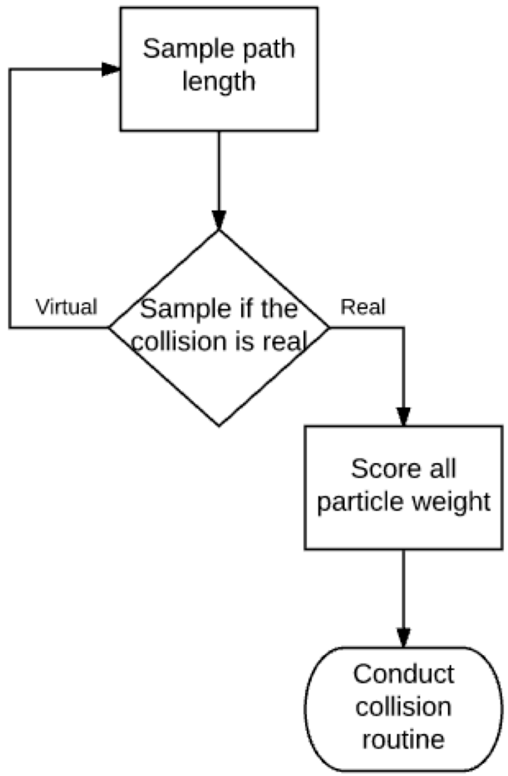
$$s(\xi) = F^{-1}(\xi) = -\frac{1}{\Sigma_{maj}} \ln \xi$$

And accept with probability

$$P_{\mathrm{real}} = \frac{f(x)}{Mg(x)} = \frac{\Sigma_t e^{-\Sigma_{\mathrm{maj}} s}}{\Sigma_{\mathrm{maj}} e^{-\Sigma_{\mathrm{maj}} s}} = \frac{\Sigma_t}{\Sigma_{\mathrm{maj}}}$$

Weighted Delta Tracking II

Woodcock Delta Tracking Process



Weighted Delta Tracking III

Woodcock Delta Tracking Advantages/Disadvantages

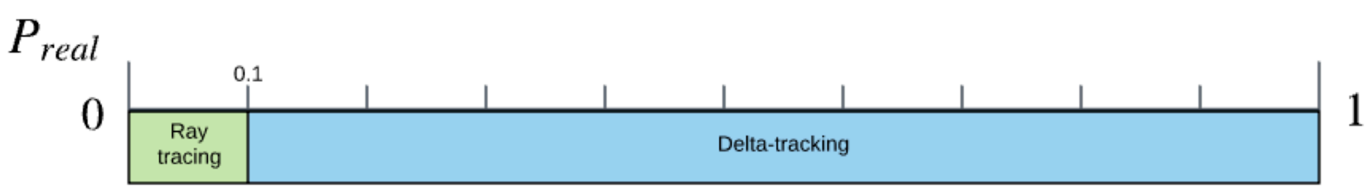
Benefits:

- Can sample across multiple geometric regions.
- No calculation of boundary distance required.

Downsides:

- Lots of virtual collisions when $\Sigma_t \ll \Sigma_{\text{maj}}$
- Heavy absorbers
- Can't use track length estimator for flux

Serpent 2 uses a combination of ray tracing and delta-tracking [2]



Weighted Delta Tracking IV

Implicit Events

Statistical processes can be replaced by using the expected value of the outcome:[1]

$$E[x] = x_1 p_1 + x_2 p_2 \ldots + x_n p_n$$

Implicit capture:

$$\begin{aligned} E[w_f] &= w_{f,\text{scattering}} P_{\text{scattering}} + w_{f,\text{absorption}} P_{\text{absorption}} \\ &= w_i P_{\text{scattering}} \end{aligned}$$

$$\begin{aligned} S_{\text{capture}} &= E[w_i - w_f] \\ &= E[w_i] - E[w_f] \\ &= w_i - w_i P_{\text{scattering}} \\ &= w_i (1 - P_{\text{scattering}}) \end{aligned}$$

Weighted Delta Tracking V

Weighted Delta Tracking

Weighted delta-tracking replaces the rejection sampling of delta-tracking with a weight reduction [3].

$$E[w_f] = w_{f,\mathrm{real}} P_{\mathrm{real}} + w_{f,\mathrm{virt}} P_{\mathrm{virt}}$$

For an absorption event:

$$\begin{aligned} E[w_f] &= w_{f,\mathrm{real}} P_{\mathrm{real}} + w_{f,\mathrm{virt}} P_{\mathrm{virt}} \\ &= 0 + w_i P_{\mathrm{virt}} \\ &= w_i(1 - P_{\mathrm{real}}) \\ &= w_i \left(1 - \frac{\Sigma_t}{\Sigma_{\mathrm{maj}}}\right) \end{aligned}$$

Weighted Delta Tracking VI

WDT Scattering

Application of the expectation value to scattering:

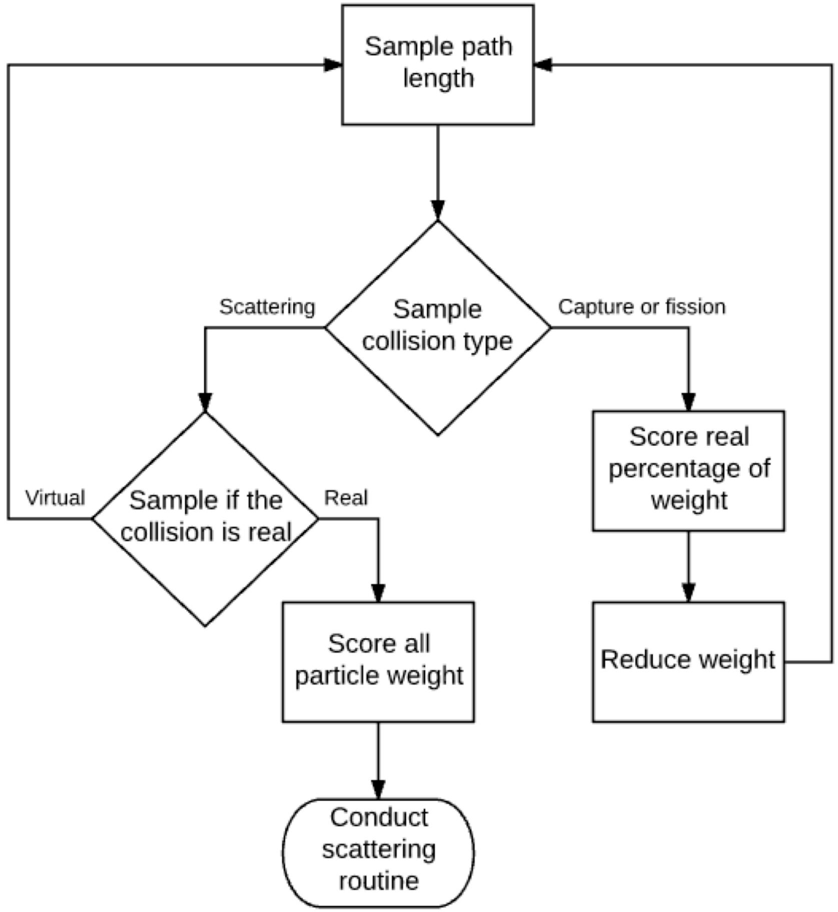
$$
\begin{aligned}
E[w_f] &= w_{f,\mathrm{real}} P_{\mathrm{real}} + w_{f,\mathrm{virt}} P_{\mathrm{virt}} \\
&= w_i P_{\mathrm{real}} + w_i P_{\mathrm{virt}} \\
&= w_i (P_{\mathrm{real}} + 1 - P_{\mathrm{real}}) \\
&= w_i
\end{aligned}
$$

This doesn't make any physical sense. In reality there are two portions of the weight: one that undergoes a real event and one that undergoes a virtual collision. Particle must be split at every scattering event.

Instead, reintroduce rejection sampling into the scattering routine to model virtual collisions explicitly.

Weighted Delta Tracking VII

WDT Process

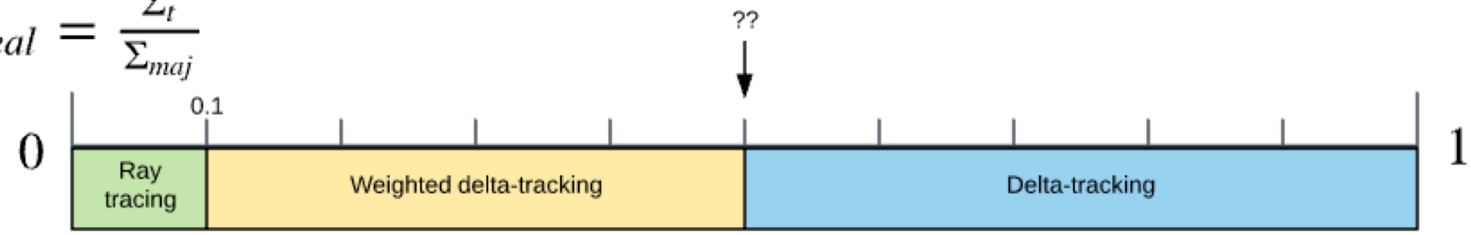


Weighted Delta Tracking VIII

WDT Scattering Summary

- Requires a rouletting routine.
- Moves collision scoring to after collision type sampling.
- Scattering collisions that are determined to be virtual cost more than normal delta-tracking virtual collisions.
- Weight reduction when $\Sigma_t \rightarrow \Sigma_{maj}$ generates many low weight particles.

$$P_{real} = \frac{\Sigma_t}{\Sigma_{maj}}$$



Idaho National Laboratory

Weighted Delta Tracking IX

Homogenous TREAT fuel cell

- Eleven energy groups.
- 100 000 source neutrons, 5 inactive and 100 active cycles
- Interested in cross-section FOM.
- Run with and without WDT, compare FOM.
- Looked at Σ_t, Σ_c, Σ_a, and Σ_{scatter} (P_0, P_1, and P_2).

Figure of merit:

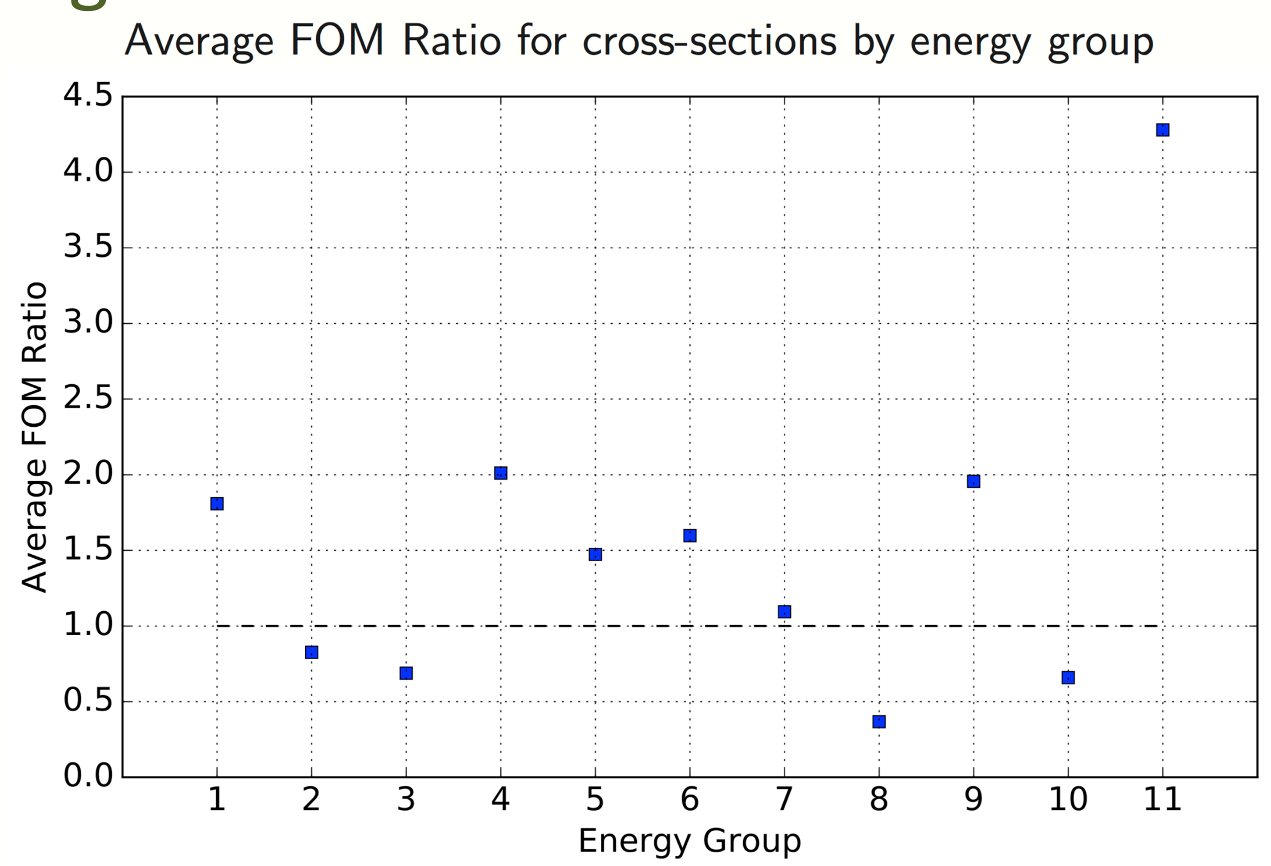
$$\text{FOM} = \frac{1}{\sigma(\hat{x})^2 T}$$

FOM ratio:

$$\text{Ratio} = \frac{\text{FOM}_{\text{WDT}}}{\text{FOM}_{\text{noWDT}}}$$

Weighted Delta Tracking X

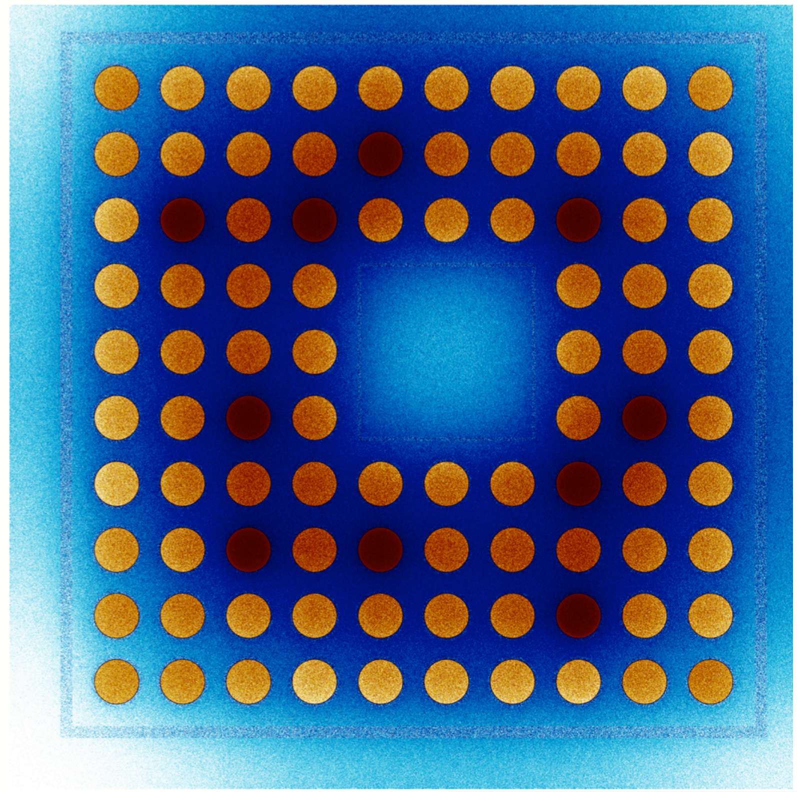
Homogenous TREAT fuel cell FOM



Weighted Delta Tracking XI

BWR

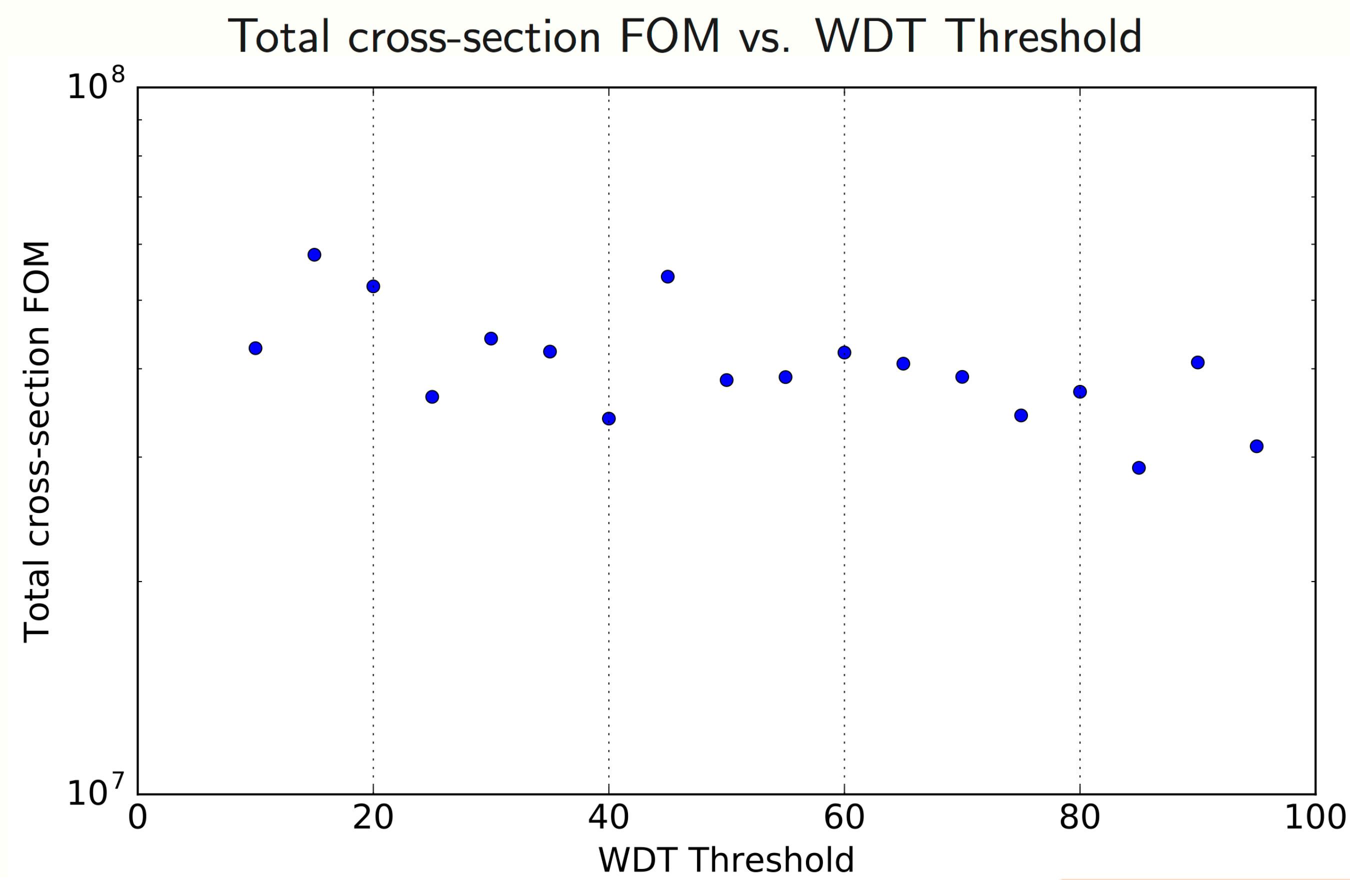
- Gd pins
- 1.295 cm pitch
- 25% void fraction
- 20 000 neutrons, 10 inactive, 500 active cycles



Idaho National Laboratory

Weighted Delta Tracking XII

BWR FOM



Future Work

Serpent—BISON coupling

- Analyze Serpent—MOOSE coupled fission, thermal
- FETs implemented, optimized in coupled Serpent
- Serpent FETs → BISON
- BISON functional expansions → Serpent
- Serpent—BISON coupling optimization
- V&V

Weighted Delta Tracking (WDT)

- Replace some surface tracking with WDT
- Test on TREAT

Idaho National Laboratory

Acknowledgements

Thank you to Jaakko Leppänen, Javier Ortensi, Ville Valtavirta, Yaqi Wang, Frederick Gleicher, Matthew Ellis, Tuomas Viitanen, Rachel Slaybaugh, Cody Permann, Andrew Slaughter, John Peterson, and Benoit Forget, all of whom provided expert direction and discussion.

This study was carried out under the auspices of the U.S. Department of Energy at Idaho National Laboratory under Contract No. DE- AC07-05ID14517.

References

- M. Ellis, B. Forget, and K. Smith, "Continuous temperature representation in coupled OpenMC/MOOSE simulations," PHYSOR 2016, Unifying Theory and Experiments in the 21st Century, Sun Valley, Idaho, May 1–5, 2016.
- M. Ellis, B. Forget, and K. Smith, "Preliminary coupling of the monte carlo code openmc and the multiphysics object-oriented simulation environment (MOOSE) for analyzing doppler feedback in monte carlo simulations," ANS M&C2015, Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method, Nashville, Tennessee, April 19–23, 2015.
- J. Leppänen, "Performance of Woodcock delta-tracking in lattice physics applications using the Serpent Monte Carlo reactor physics burnup calculation code," Annals of Nuclear Energy, vol. 37, pp. 715–722, 2010.
- J. Leppänen, V. Valtavirta, and T. Viitanen, "Unstructured mesh based multi-physics interface for CFD code coupling in the Serpent 2 Monte Carlo code," PHYSOR 2014, The Role of Reactor Physics toward a Sustainable Future, Kyoto, Japan, September 28 – October 3, 2014.
- L. Morgan and D. Kotlyar, "Weighted-delta-tracking for Monte Carlo particle transport," Annals of Nuclear Energy, vol. 85, pp. 1184–1188, 2015.