LA-UR-22-26339

Title:	Monte Carlo Transport
Author(s):	Trahan, Travis John
Intended for:	Computational Physics Summer Workshop Lecture Series
Issued:	2022-07-01

Los Alamos
NATIONAL LABORATORY
EST. 1943

Delivering science and technology
to protect our nation
and promote world stability

Monte Carlo Transport

Computational Physics Summer Workshop

Travis Trahan, XCP-3

June 22, 2022

Legal Notice

Outline

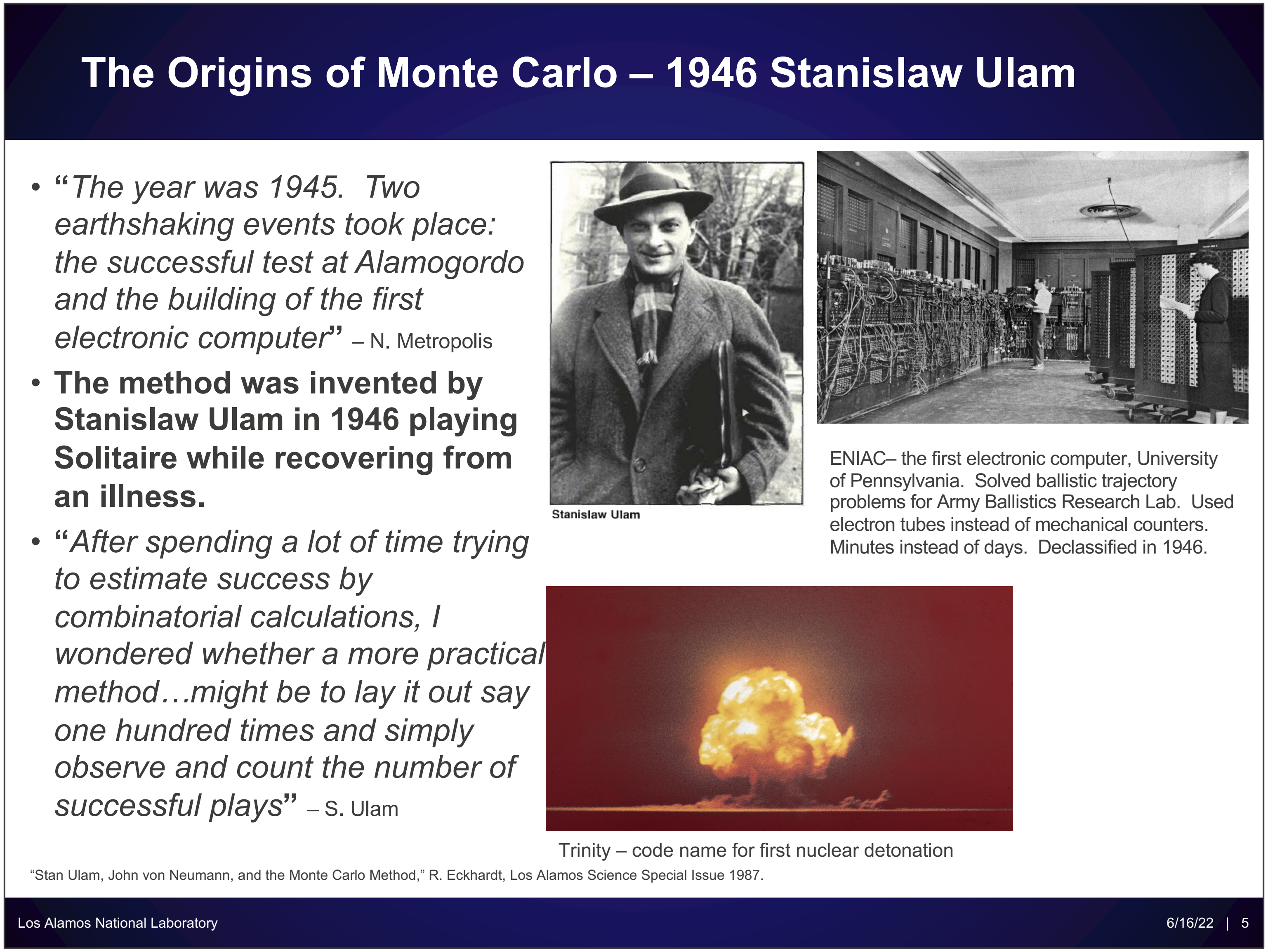
- History and Applications of the Monte Carlo Method
- The Random Walk
- Eigenvalue and Time-Dependent Algorithms
- Advantages and Disadvantages of Monte Carlo Methods
- Tally Basics
- Types of Tallies
- Using Variance Reduction to Improve Efficiency
- Exercise

History and Applications of the Monte Carlo Method

What is Monte Carlo Radiation Transport?

- **In general, Monte Carlo methods simulate large numbers of random trials in order to observe numerical behavior of systems described by probabilistic behavior**
  - Buffon’s needle experiment to calculate pi
  - Average and standard deviation of long term stock market returns
- **In radiation transport, pseudo-random number generators are used to randomly sample individual particle lives**
  - Source position, direction, and energy
  - Distance to collision
  - Collision type
  - Outgoing particle states
  - Repeat for many particle histories until each history ends (e.g., by leakage)
- **Information about the particles are tallied**
  - Reaction rates
  - Energy deposition
  - Multiplication

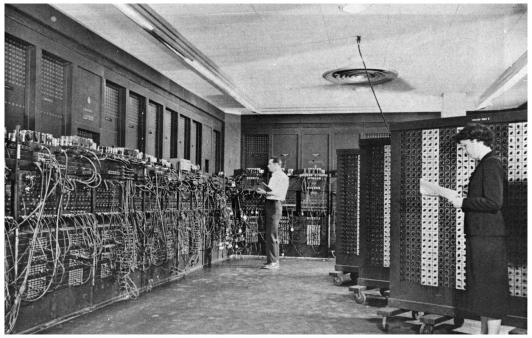
A. Sood, "The Monte Carlo Method and MCNP - A Brief Review of Our 40 Year History", LA-UR-17-25633.

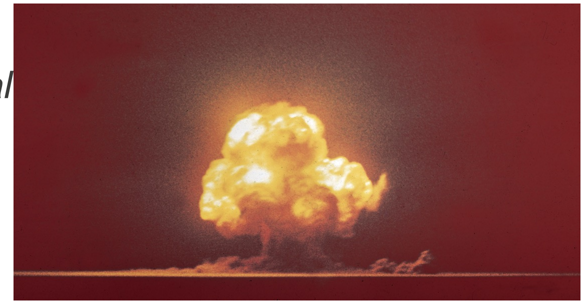


The Origins of Monte Carlo – 1946 Stanislaw Ulam

- "*The year was 1945. Two earthshaking events took place: the successful test at Alamogordo and the building of the first electronic computer*" – N. Metropolis
- **The method was invented by Stanislaw Ulam in 1946 playing Solitaire while recovering from an illness.**
- "*After spending a lot of time trying to estimate success by combinatorial calculations, I wondered whether a more practical method…might be to lay it out say one hundred times and simply observe and count the number of successful plays*" – S. Ulam

Stanislaw Ulam


ENIAC– the first electronic computer, University of Pennsylvania. Solved ballistic trajectory problems for Army Ballistics Research Lab. Used electron tubes instead of mechanical counters. Minutes instead of days. Declassified in 1946.


Trinity – code name for first nuclear detonation

"Stan Ulam, John von Neumann, and the Monte Carlo Method," R. Eckhardt, Los Alamos Science Special Issue 1987.

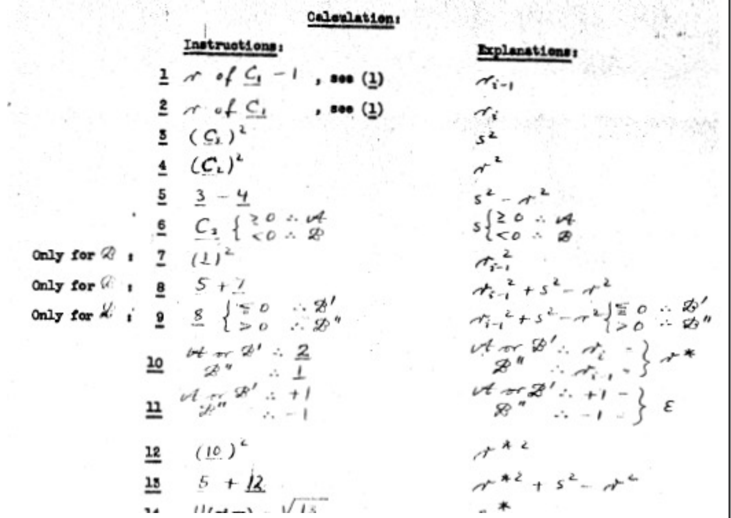
Los Alamos National Laboratory 6/16/22 | 5

A. Sood, “The Monte Carlo Method and MCNP - A Brief Review of Our 40 Year History”, LA-UR-17-25633.

The first Monte Carlo (pseudo) Code - 1947

- **Von Neumann’s Assumptions:**
  - Time-dependent, continuous energy, spherical but radially-varying, 1 fissionable material, isotropic scattering and fission production, fission multiplicities of 2,3, or 4
- **Suggested 100 neutrons each to be run for 100 collisions**
  - Thought these were too much
- **Estimated time: 5 hrs on ENIAC**
- **Richtmyer’s response:**
  - Very interested in idea and proposed suggestions
    - Allow for multiple fissionable materials, no fission spectrum energy dependence, single neutron multiplicity, run for computer time not collisions
- **ENIAC: first calculations run April/May 1948**
  - Code finalized in **December 1947**;
  - Continuous energy neutrons, fission spectra and XS tabulated at interval mid-points, histogram energy-dependence of XS, pseudo-RN.

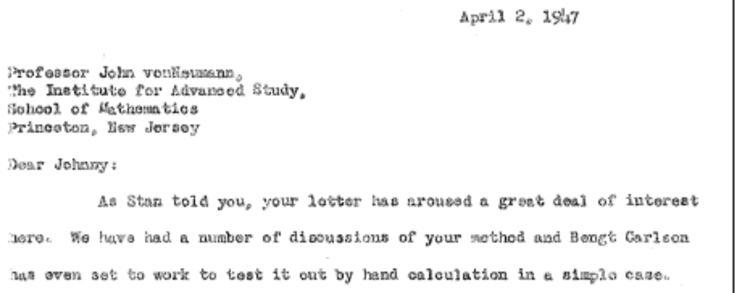
Thomas Haight, et al., “Los Alamos Bets on ENIAC: Nuclear Monte Carlo Simulations, 1947-1948, IEEE Ann. Of History of Comp July-Sept 2014



Calculation:

		Instructions:	Explanations:
	1	r of $C_1 - 1$, see (1)	r_{i-1}
	2	r of C_1, see (1)	r_i
	3	$(C_2)^2$	s^2
	4	$(C_2)^2$	r^2
	5	$3 - 4$	$s^2 - r^2$
	6	$C_3 \begin{cases} \geq 0 \therefore \mathcal{A} \\ < 0 \therefore \mathcal{B} \end{cases}$	$s \begin{cases} \geq 0 \therefore \mathcal{A} \\ < 0 \therefore \mathcal{B} \end{cases}$
Only for $\mathcal{B}$:	7	$(1)^2$	r_{i-1}^2
Only for $\mathcal{B}$:	8	$5 + 7$	$r_{i-1}^2 + s^2 - r^2$
Only for $\mathcal{B}$:	9	$8 \begin{cases} \leq 0 \therefore \mathcal{B}' \\ > 0 \therefore \mathcal{B}'' \end{cases}$	$r_{i-1}^2 + s^2 - r^2 \begin{cases} \leq 0 \therefore \mathcal{B}' \\ > 0 \therefore \mathcal{B}'' \end{cases}$
	10	$\mathcal{A}$ or $\mathcal{B}' \therefore 2$; $\mathcal{B}'' \therefore 1$	$\left.\begin{matrix} \mathcal{A} \text{ or } \mathcal{B}' \therefore r_i - \\ \mathcal{B}'' \therefore r_{i-1} - \end{matrix}\right\} r^*$
	11	$\mathcal{A}$ or $\mathcal{B}' \therefore +1$; $\mathcal{B}'' \therefore -1$	$\left.\begin{matrix} \mathcal{A} \text{ or } \mathcal{B}' \therefore +1 - \\ \mathcal{B}'' \therefore -1 - \end{matrix}\right\} \varepsilon$
	12	$(10)^2$	r^{*2}
	13	$5 + 12$	$r^{*2} + s^2 - r^2$

R.D Richtmyer and J. von Neumann “Statistical Methods in Neutron Diffusion”, Los Alamos (LAMS-557) April 9, 1947.



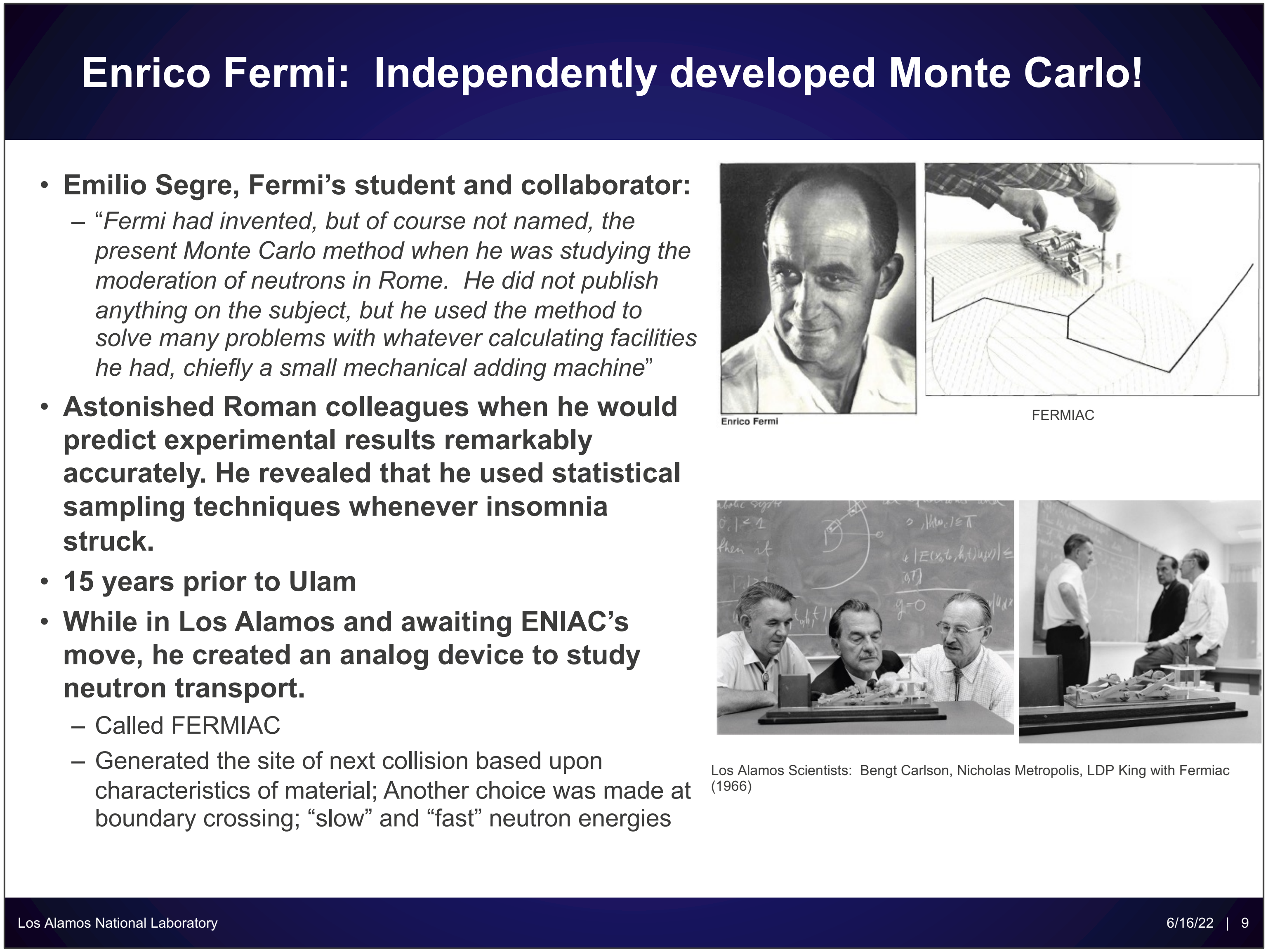
April 2, 1947

Professor John vonNeumann,
The Institute for Advanced Study,
School of Mathematics
Princeton, New Jersey

Dear Johnny:

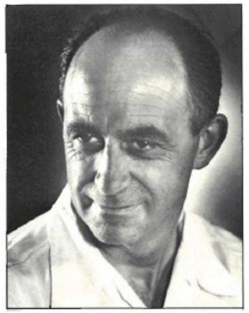
As Stan told you, your letter has aroused a great deal of interest here. We have had a number of discussions of your method and Bengt Carlson has even set to work to test it out by hand calculation in a simple case.

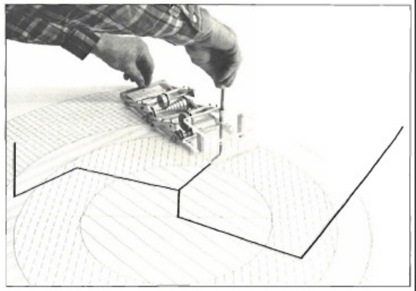
A. Sood, “The Monte Carlo Method and MCNP - A Brief Review of Our 40 Year History”, LA-UR-17-25633.

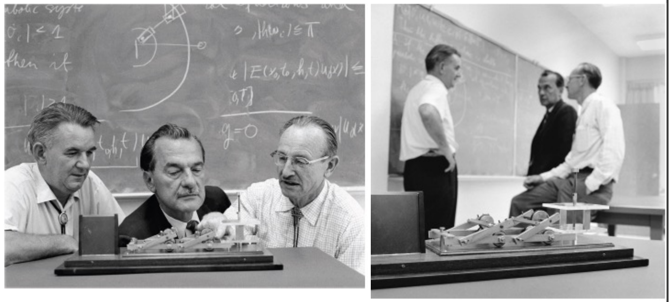


Enrico Fermi: Independently developed Monte Carlo!

- **Emilio Segre, Fermi’s student and collaborator:**
  - *“Fermi had invented, but of course not named, the present Monte Carlo method when he was studying the moderation of neutrons in Rome. He did not publish anything on the subject, but he used the method to solve many problems with whatever calculating facilities he had, chiefly a small mechanical adding machine”*
- **Astonished Roman colleagues when he would predict experimental results remarkably accurately. He revealed that he used statistical sampling techniques whenever insomnia struck.**
- **15 years prior to Ulam**
- **While in Los Alamos and awaiting ENIAC’s move, he created an analog device to study neutron transport.**
  - Called FERMIAC
  - Generated the site of next collision based upon characteristics of material; Another choice was made at boundary crossing; “slow” and “fast” neutron energies


Enrico Fermi


FERMIAC

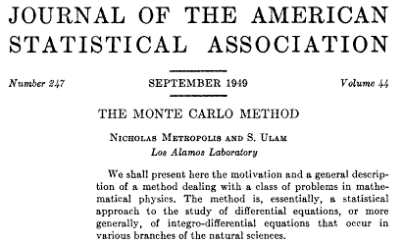

Los Alamos Scientists: Bengt Carlson, Nicholas Metropolis, LDP King with Fermiac (1966)

Los Alamos National Laboratory 6/16/22 | 9

A. Sood, “The Monte Carlo Method and MCNP - A Brief Review of Our 40 Year History”, LA-UR-17-25633.

MANIAC – Nicholas Metropolis

- **Post-war ENIAC started a revolution that continues today**
- **MANIAC – Mathematical and Numerical Integrator and Computer**
  – Was a product of Nicholas Metropolis at LANL; borrowed concepts from von Neumann’s IAS, operational in 1952;
  – MADCAP – high-level language and compiler
  – Rapid growth of computing: AVIDAC (Argonne) ORACLE (Oak Ridge), ILLIAC (U of I)
  – Special effort that helped bind Von Neumann, Fermi, Beta, Teller, Ulam, Feynman, etc in post-war efforts. MANIAC was a fascination.
  – First time “Monte Carlo” appears in publication:
    - Nicholas Metropolis and S. Ulam , “The Monte Carlo Method,” *Journal of the American Statistical Association* Vol. 44, No. 247 (Sep., 1949)
  – MC on MANIAC used for multiple problems other than radiation transport:



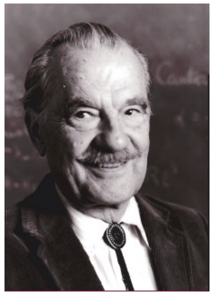
JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION

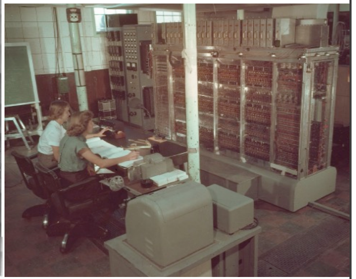
Number 247 SEPTEMBER 1949 *Volume 44*

THE MONTE CARLO METHOD

NICHOLAS METROPOLIS AND S. ULAM
Los Alamos Laboratory

We shall present here the motivation and a general description of a method dealing with a class of problems in mathematical physics. The method is, essentially, a statistical approach to the study of differential equations, or more generally, of integro-differential equations that occur in various branches of the natural sciences.





Pion-proton phase-shift analysis (Fermi, Metropolis; 1952)
Phase-shift analysis (Bethe, deHoffman, Metropolis; 1954)
Nonlinear coupled oscillators (Fermi, Pasta, Ulam; 1953)
Genetic code (Gamow, Metropolis; 1954)
Equation of state: Importance sampling (Metropolis, Teller; 1953)
Two-dimensional hydrodynamics (Metropolis, von Neumann; 1954)
Universalities of iterative functions (Metropolis, Stein, Stein; 1973)
Nuclear cascades using Monte Carlo (Metropolis, Turkevich; 1954)
Anti-clerical chess (Wells; 1956)
The lucky numbers (Metropolis, Ulam; 1956)

A. Sood, “The Monte Carlo Method and MCNP - A Brief Review of Our 40 Year History”, LA-UR-17-25633.

Monte Carlo & MCNP History

ENIAC – 1945

30 tons

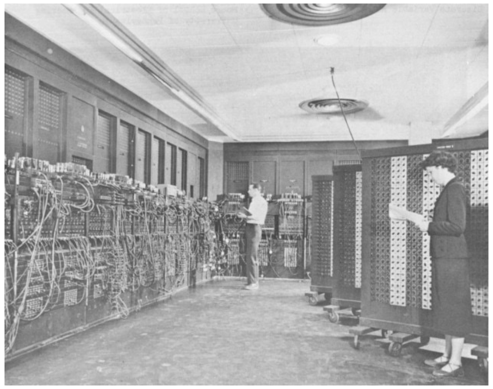
20 ft x 40 ft room

18,000 vacuum tubes

0.1 MHz

20 word memory

patchcords



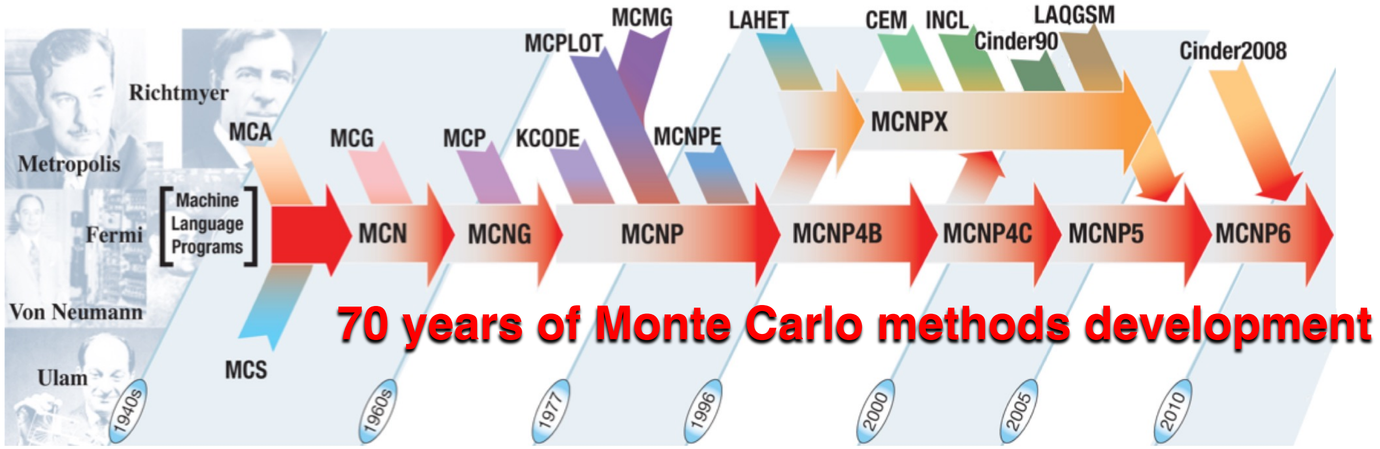
Manhattan Project – 1945...

Discussions on using ENIAC

Ulam suggested using the “method of statistical trials”

Metropolis suggested the name “Monte Carlo”

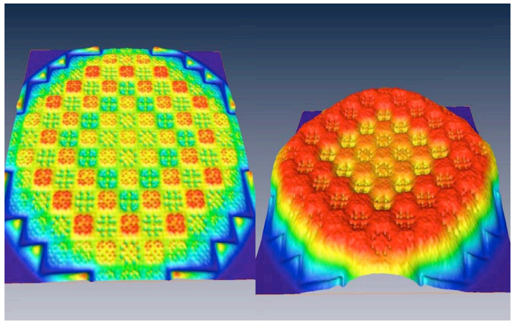
Von Neumann developed the first computer code



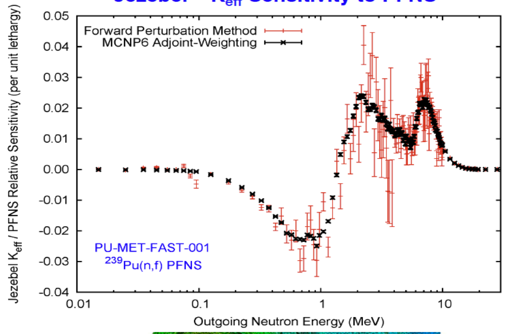
MCNP® Capabilities

- **Physics:**
  - Continuous energy particle transport
  - Neutron, photon, electron, and many more particle types
- **Algorithms:**
  - k-eigenvalue calculations
  - Fixed source calculations
- **Recently Implemented Features:**
  - Unstructured mesh transport
  - Electric and magnetic field transport
  - High-energy physics models
  - 33 additional particle types
  - Reactor fuel depletion and burnup
  - Radiation source and detection capabilities
  - Sensitivity and uncertainty analysis for nuclear criticality safety
- **Extensive Variance Reduction**
  - Weight Windows
  - DXTRAN

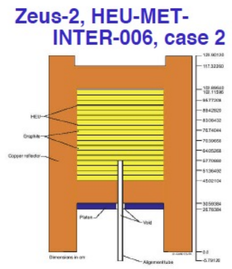
Whole-core Thermal & Total Flux

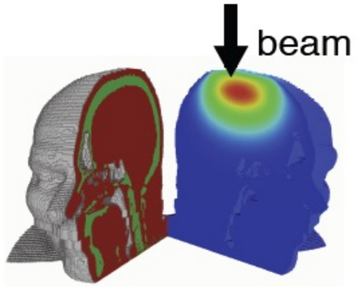


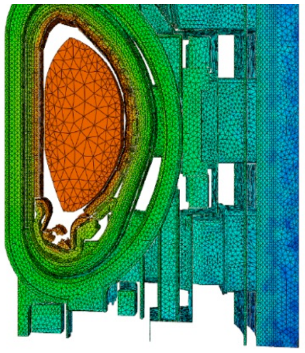
Jezebel – K_{eff} Sensitivity to PFNS



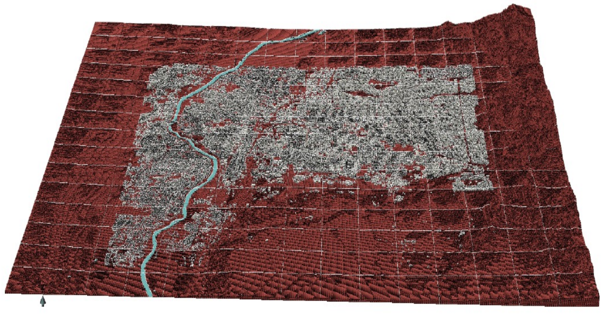
Experimental Benchmarks with Critical Assemblies







ITER Neutron Flux Calculations

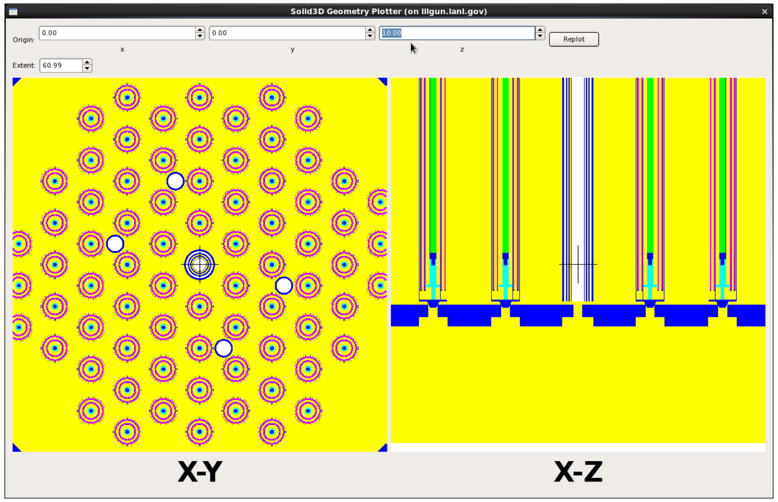


City model used to study nuclear weapon effects

MCATK

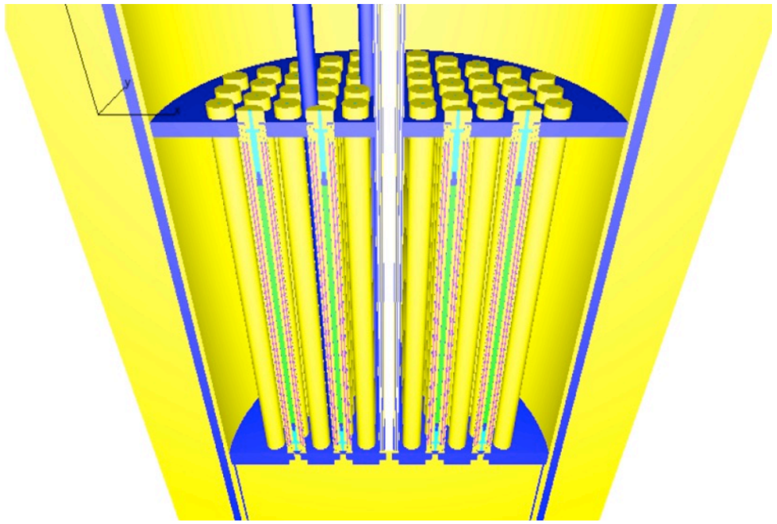
- **Physics:**
  - Continuous energy particle transport
  - Neutron-photon only (for now)
- **Algorithms:**
  - k- and α-eigenvalue calculations
  - Fixed source calculations
  - Time-dependent calculations
  - Fission chain analysis
- **Other Notable Features:**
  - Solid body and mesh geometries
  - Multiple parallel computing paradigms
  - Generic source definitions
  - Track-length and expected value tallies
  - Weight-windows
  - Multitemperature treatment
  - Visualization Tools
  - Python interface for problem setup

MCATK solid body representation of ICT2C3 Critical Benchmark



Origin: 0.00 (x) 0.00 (y) 10.00 (z)

Extent: 60.99



The Random Walk

Pseudo-Random Numbers

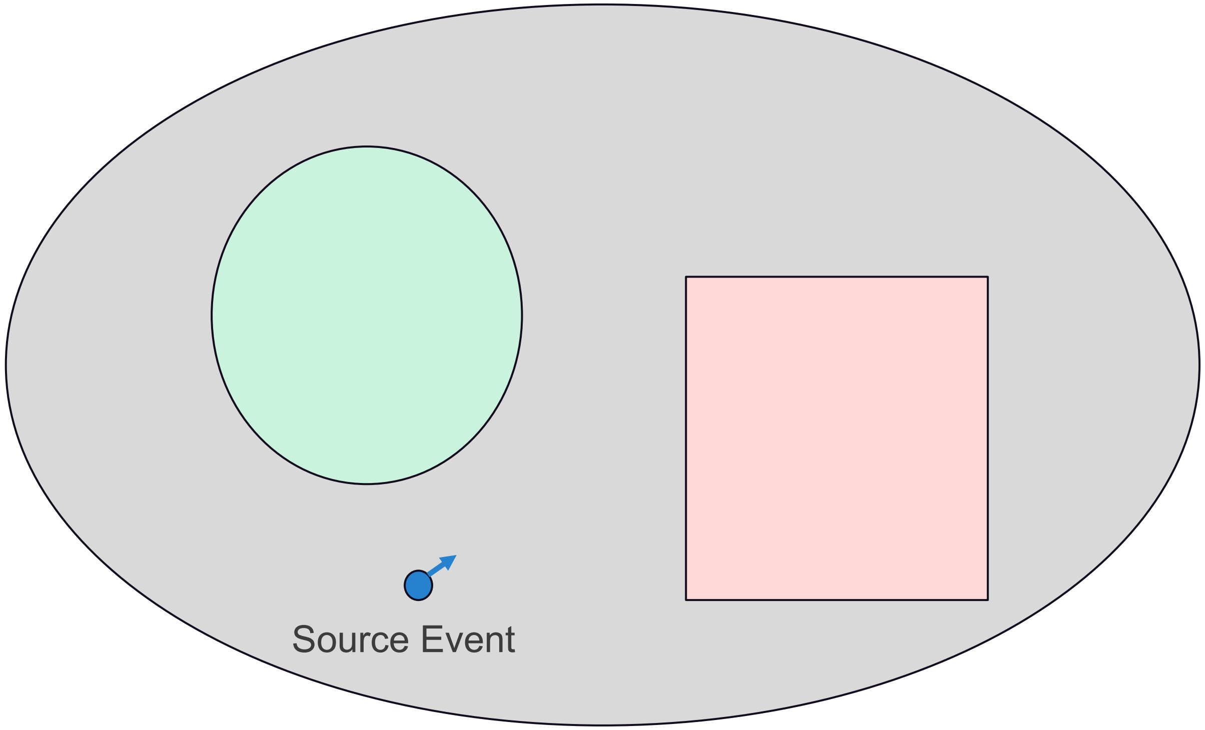
- **A pseudorandom number generator deterministically calculates a stream of numbers in $(0, 1]$ that appears random**
  - Must satisfy certain statistical tests
- **Truly random numbers are not desirable**
  - Simulations should be reproducible for testing and debugging
- **The starting point in the stream is determined by the seed**
  - Statistically independent results for the same calculation can be obtained by changing the random number seed

Particle Random Walk: Algorithm Overview

```
For each history:
    Sample source state
    While particle is alive:
        Calculate distance to next event
        Move particle
        If event is a collision:
            Sample isotope
            Sample collision type
            Sample number of outgoing particles (if req.)
            Sample outgoing particle states (if req.)
            Add all but the first outgoing particle to a bank (if req.)
        If particle is dead and particle bank is not empty:
            Grab a secondary particle from the bank
        Else if particle is dead and particle bank is empty:
            Start a new history (outermost loop)
```

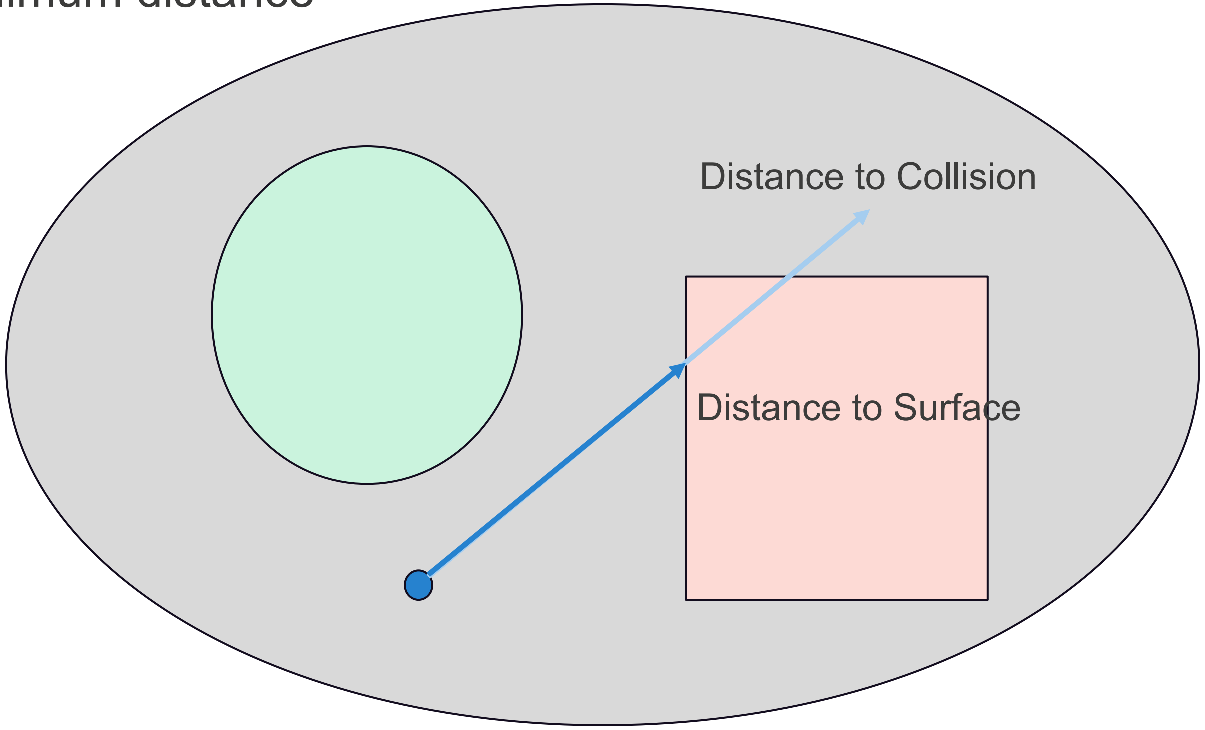
Particle Random Walk: Sample Source State

- Using random numbers, select initial:
  - Position
  - Energy
  - Angle
  - Birth Time



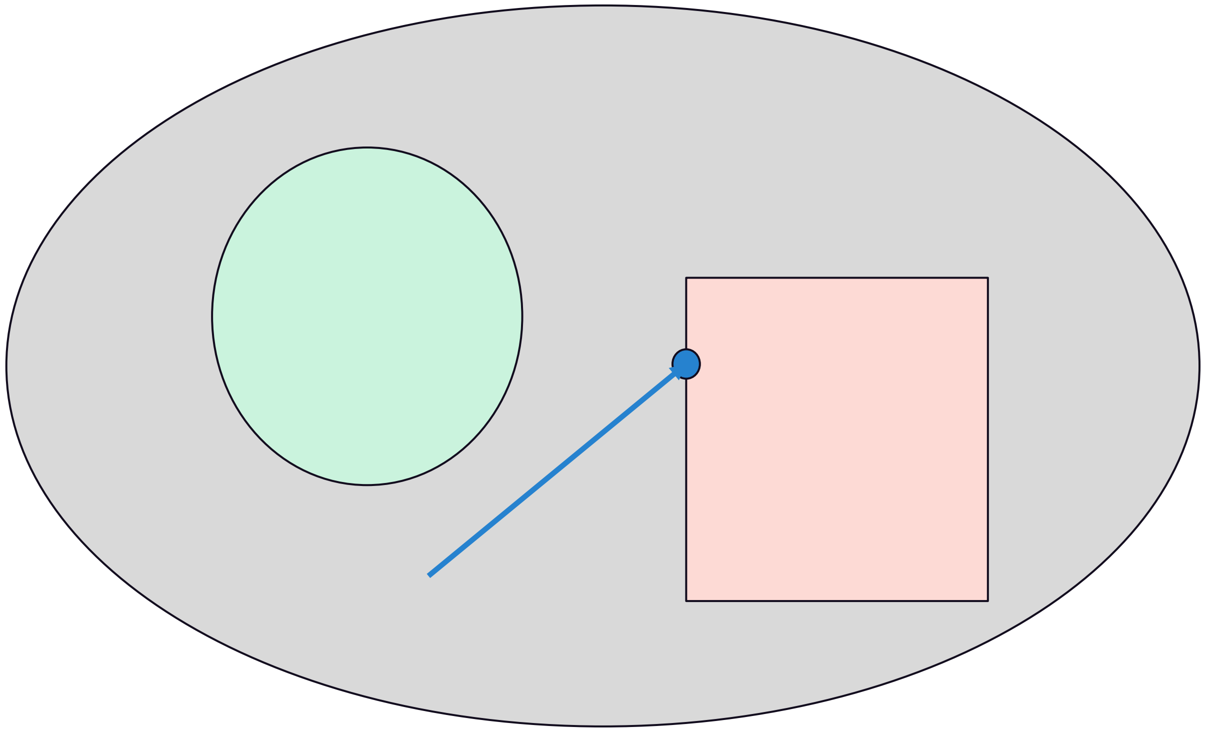
Particle Random Walk: Distance to Event

- Compute distance to:
  - Boundary crossing events
  - Collision events: $d = -\frac{1}{\Sigma_t}\ln\xi$
  - Time boundary (if time-dependent)
- Select event with minimum distance



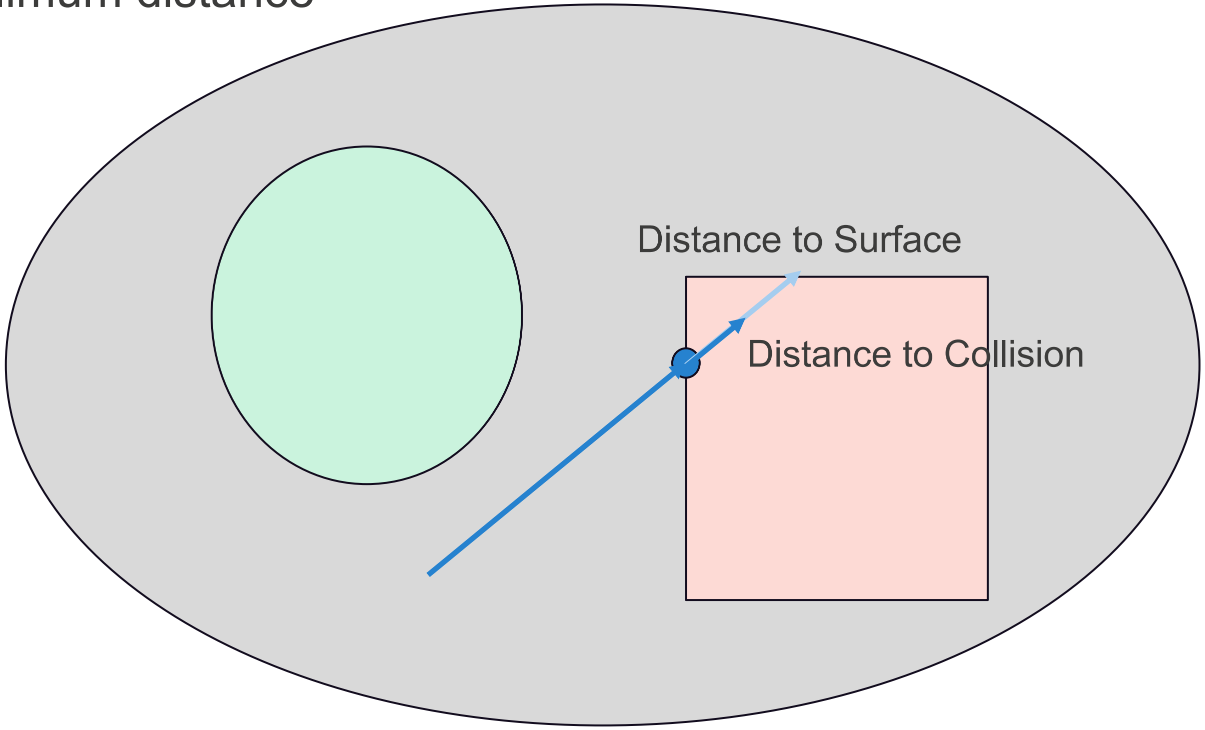
Particle Random Walk: Move Particle

– Move particle the selected distance on its current trajectory



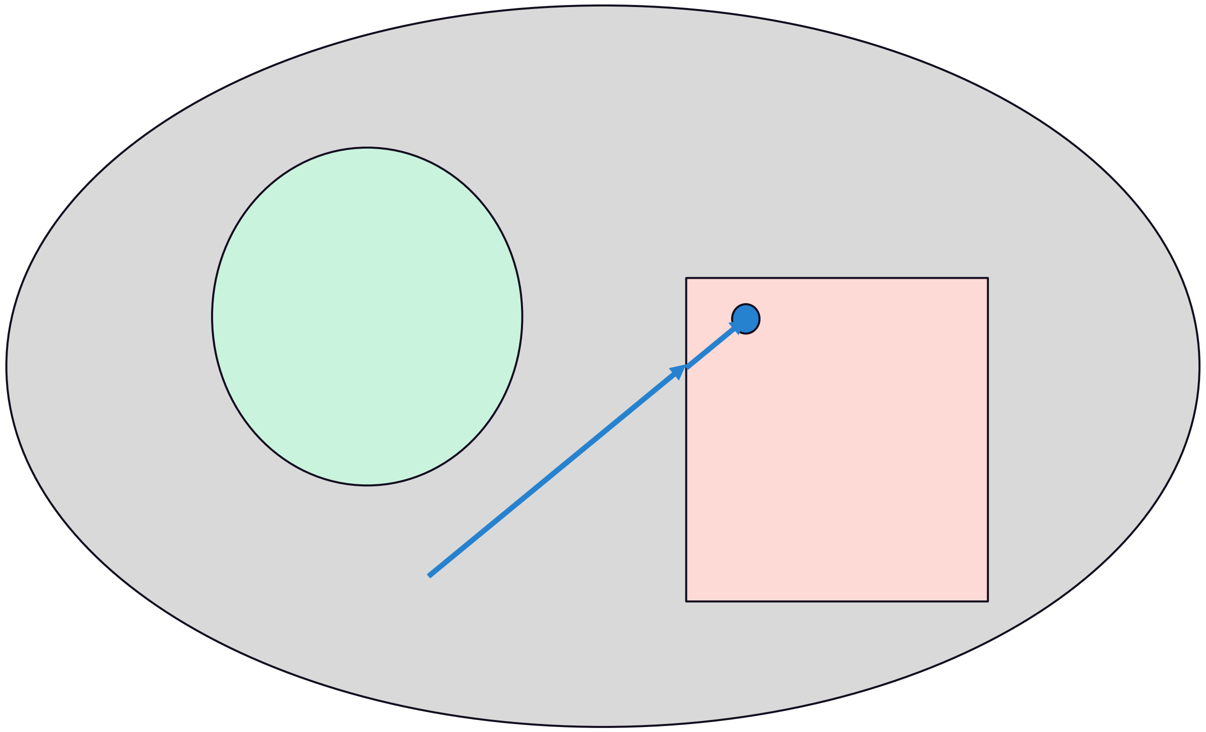
Particle Random Walk: Distance to Event

- Compute distance to:
  - Boundary crossing events
  - Collision events: $d = -\frac{1}{\Sigma_t} \ln \xi$
  - Time boundary (if time-dependent)
- Select event with minimum distance



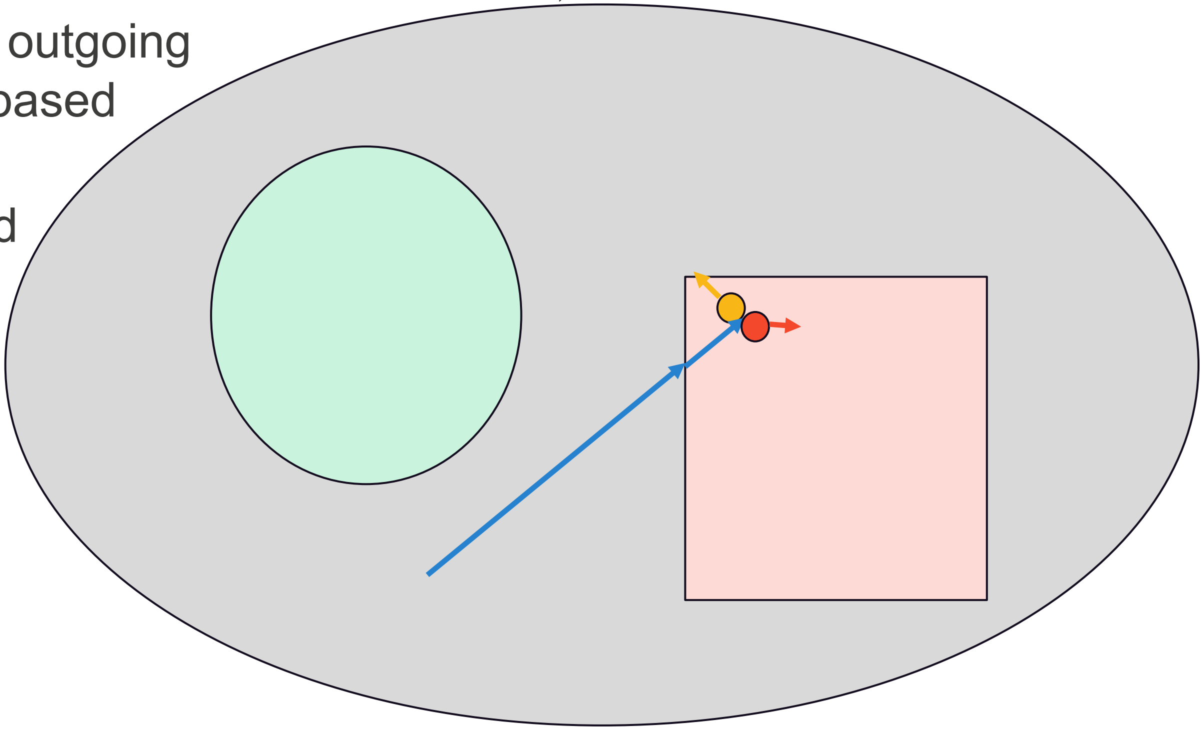
Particle Random Walk: Move Particle

– Move particle the selected distance on its current trajectory



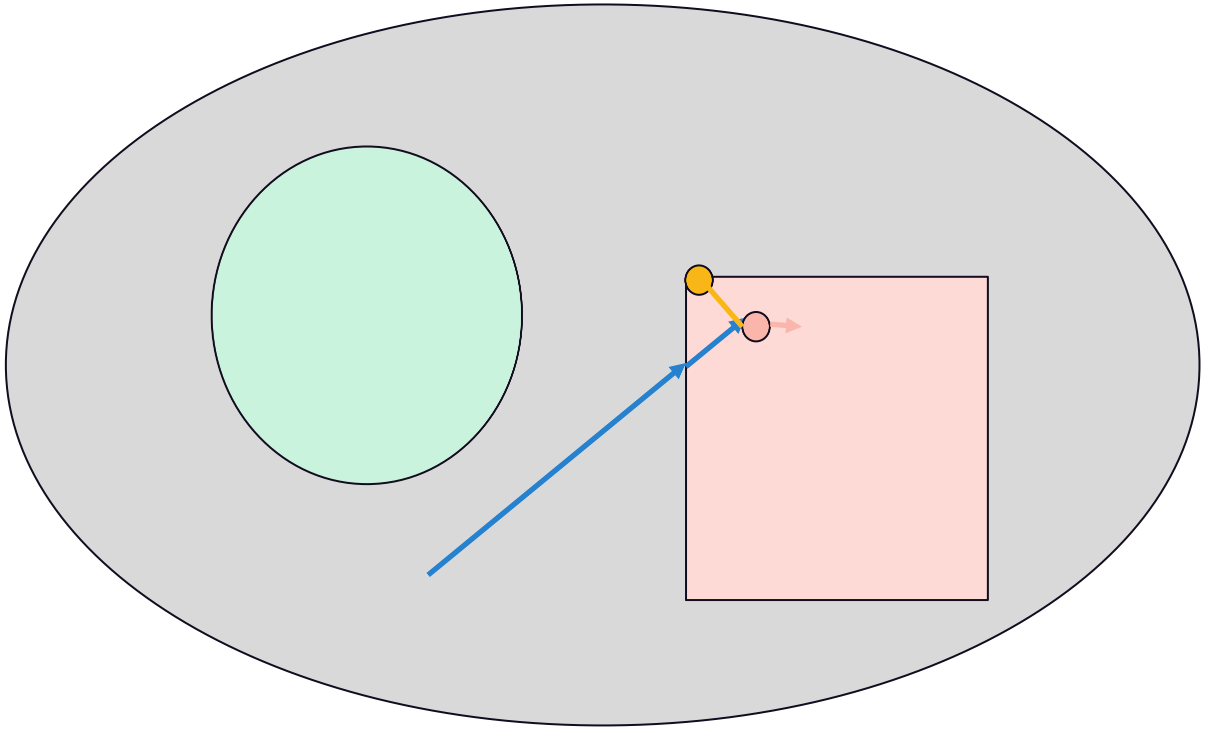
Particle Random Walk: Sample Collision

- Select the isotope within the material that the particle collides with:
  - The probability of colliding with each isotope i is: $p_i = \frac{\Sigma_{t,i}}{\Sigma_t}$
- Select the specific collision type:
  - The probability of reaction r in isotope i is: $p_{r,i} = \frac{\Sigma_{r,i}}{\Sigma_{t,i}}$
- Select the number of outgoing particles of all types based on nuclear data
- Select the energy and angle of each outgoing particle



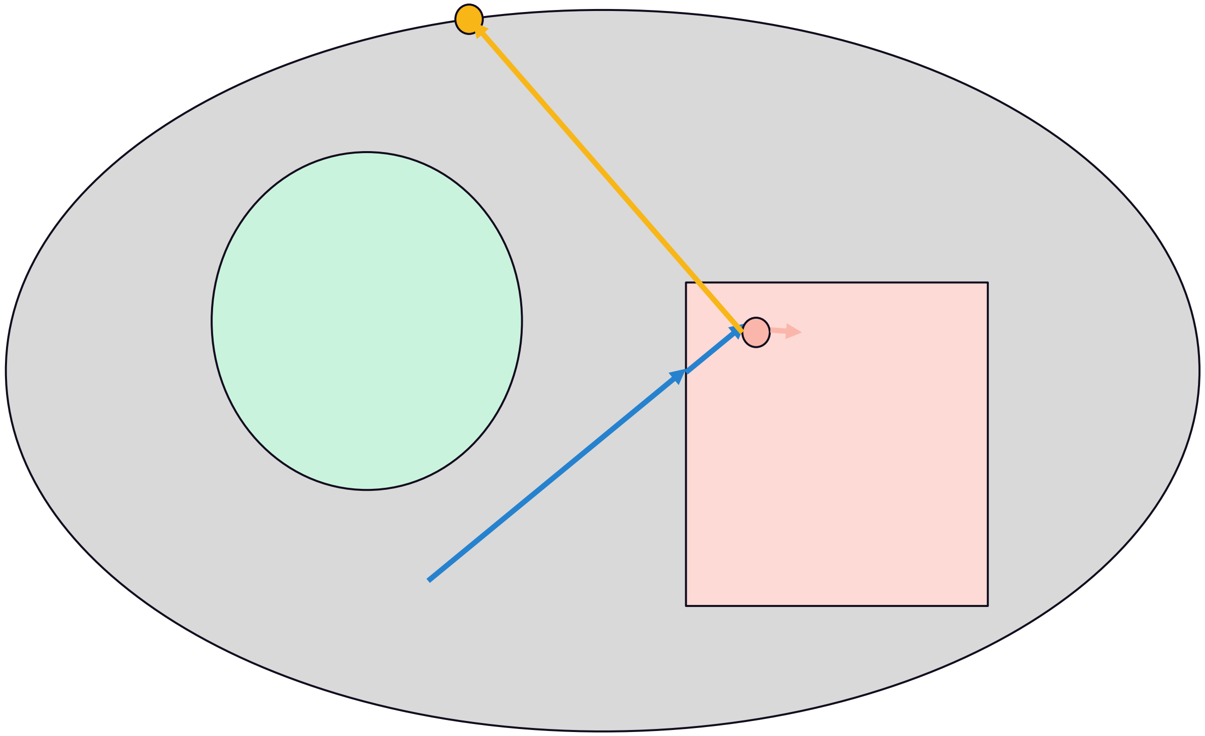
Particle Random Walk: Transport the First Outgoing Particle

- Transport one of the outgoing particles until it dies by leakage or absorption
- Add the rest to a bank



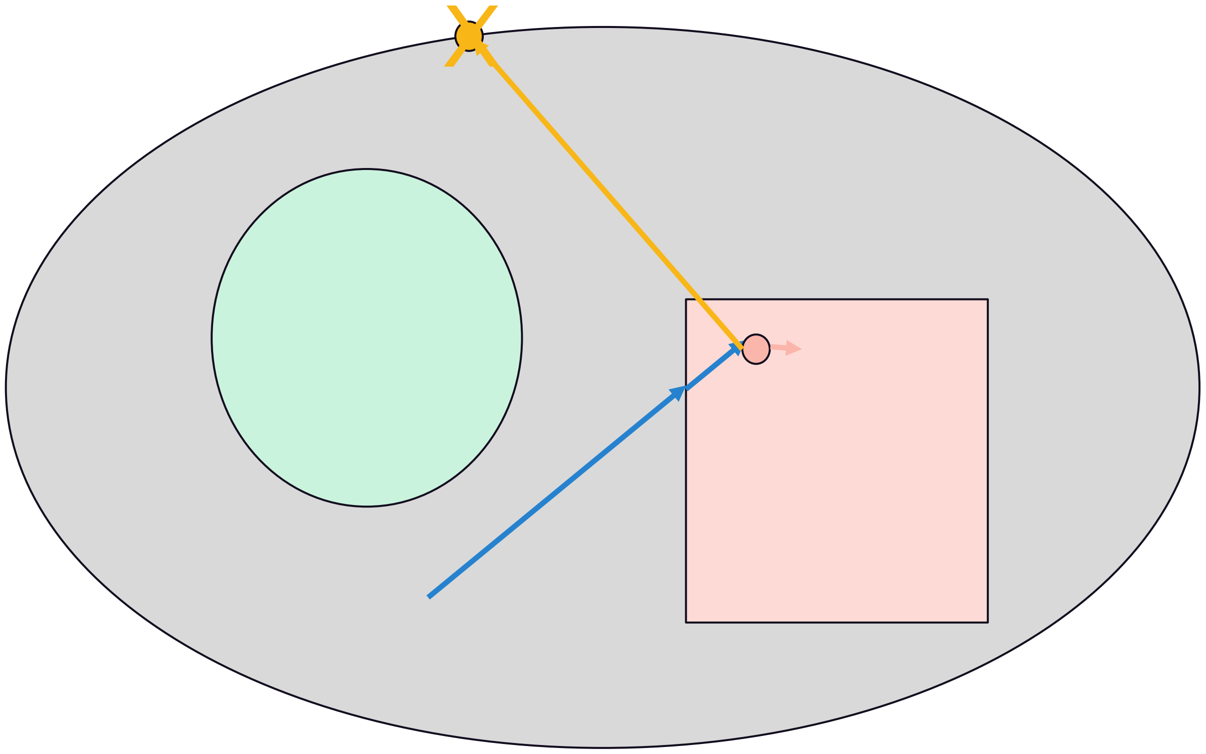
Particle Random Walk:
Transport the First Outgoing Particle

- Transport one of the outgoing particles until it dies by leakage or absorption
- Add the rest to a bank



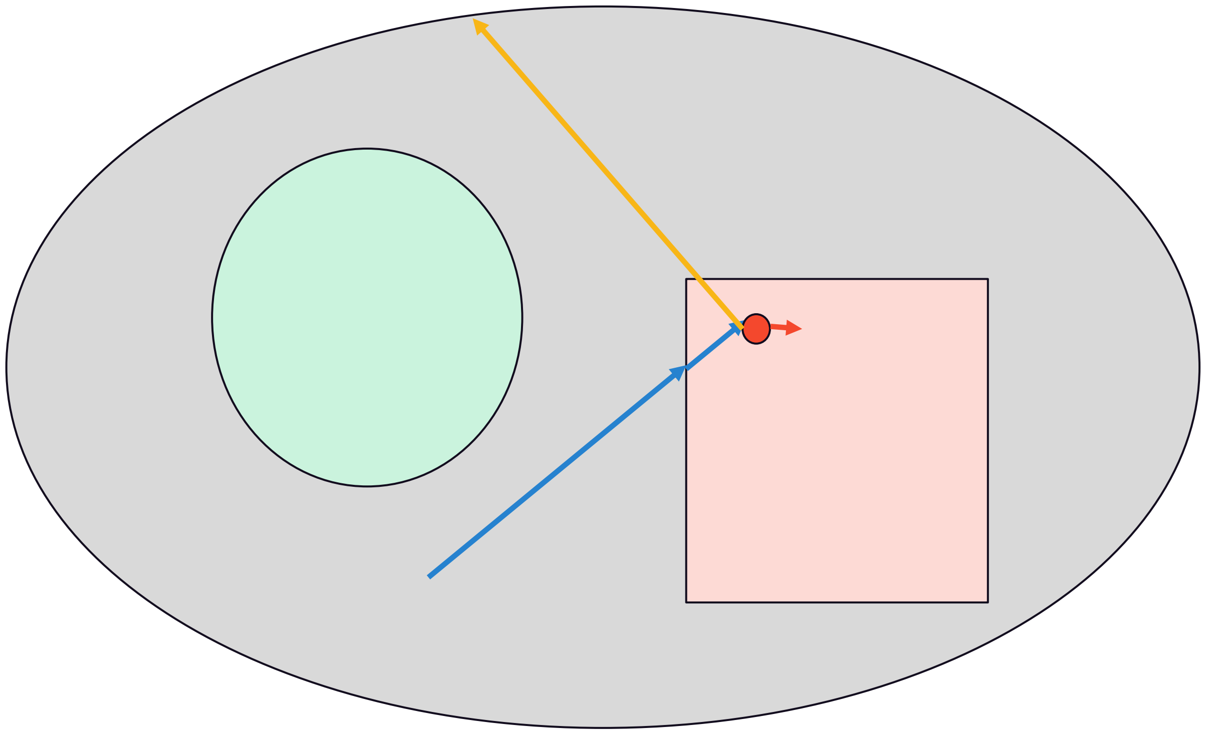
Particle Random Walk: Transport the First Outgoing Particle

– Transport one of the outgoing particles until it dies by leakage or absorption
– Add the rest to a bank



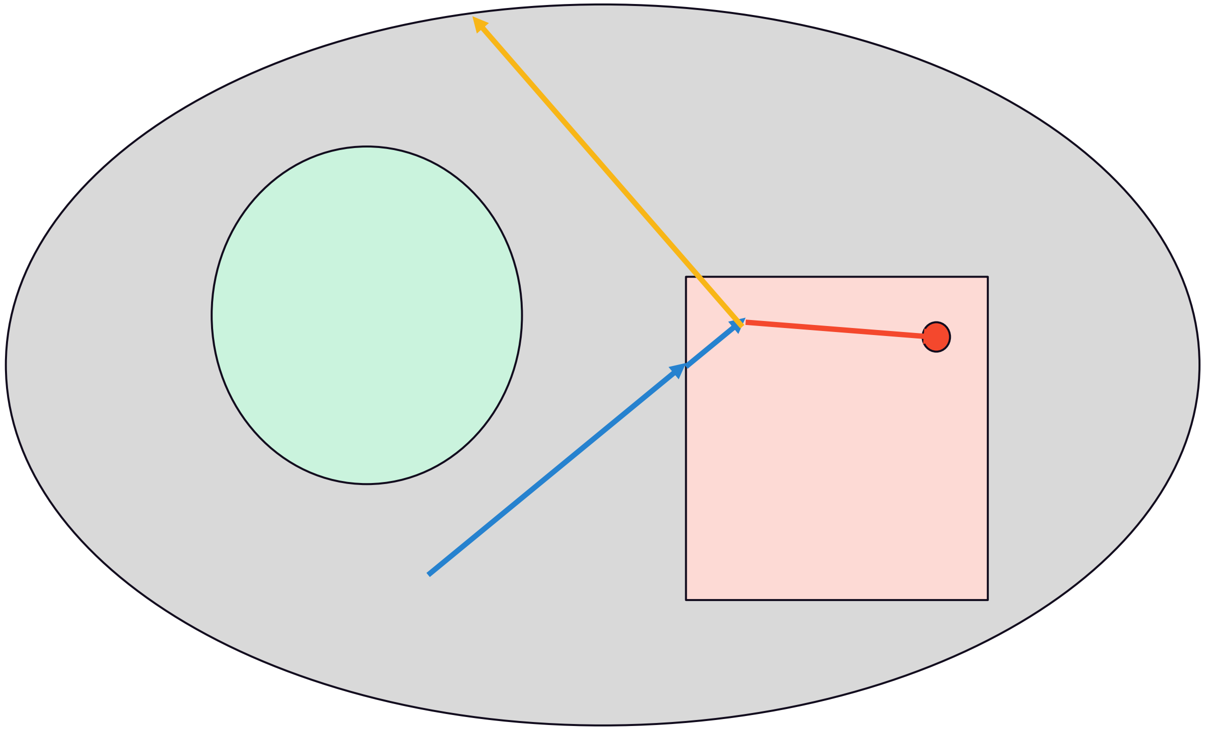
Particle Random Walk: Transport the Particles in the Bank

- Grab a particle from the bank
- Transport it until it dies by leakage or absorption
- Repeat as necessary



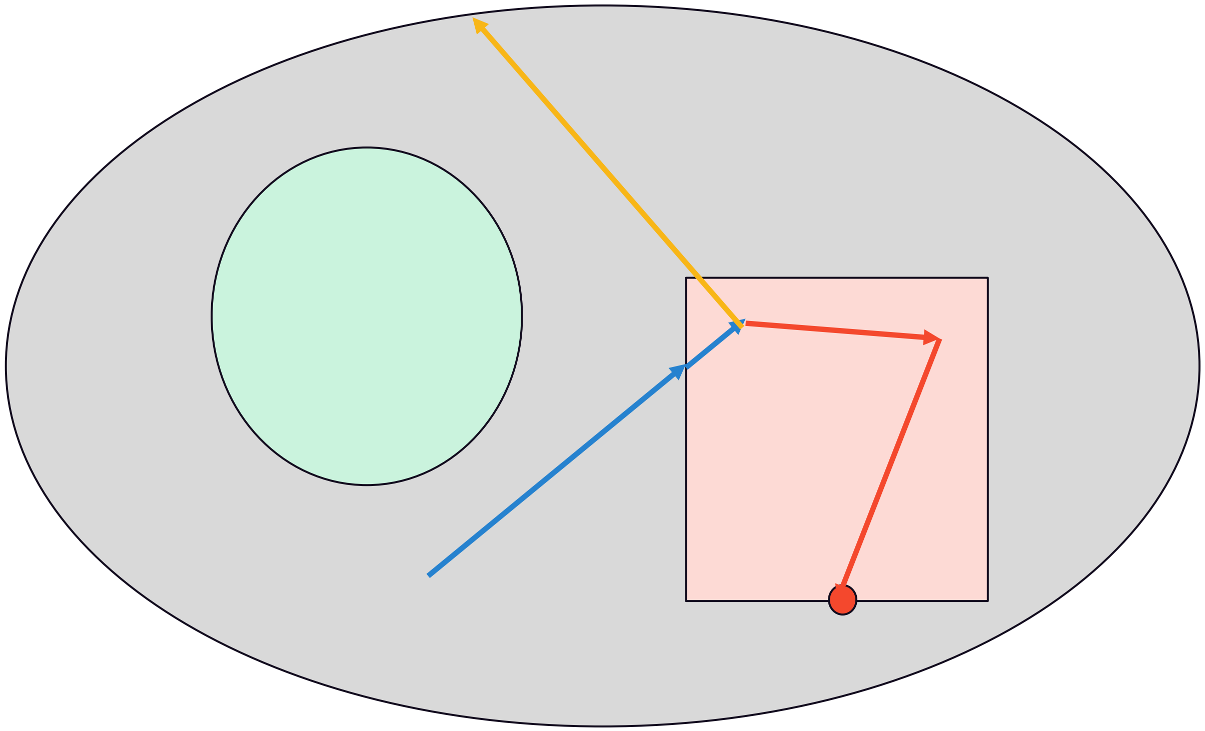
Particle Random Walk: Transport the Particles in the Bank

- Grab a particle from the bank
- Transport it until it dies by leakage or absorption
- Repeat as necessary



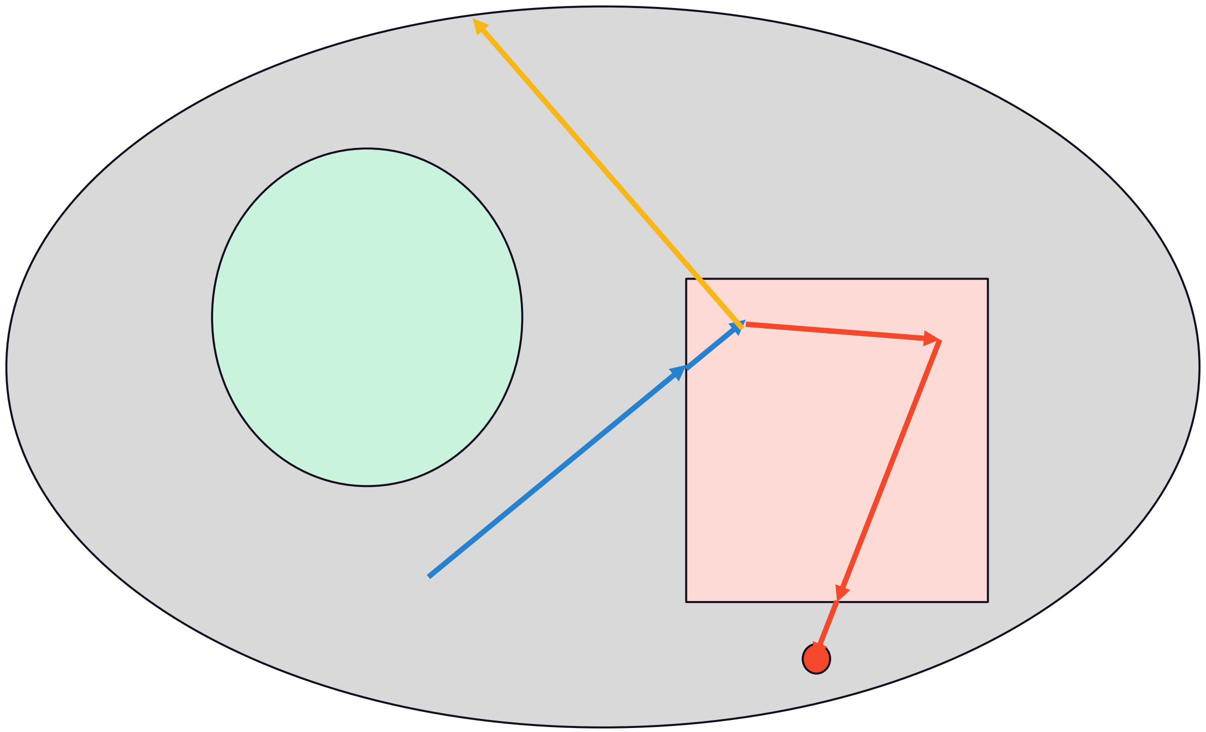
Particle Random Walk:
Transport the Particles in the Bank

- Grab a particle from the bank
- Transport it until it dies by leakage or absorption
- Repeat as necessary



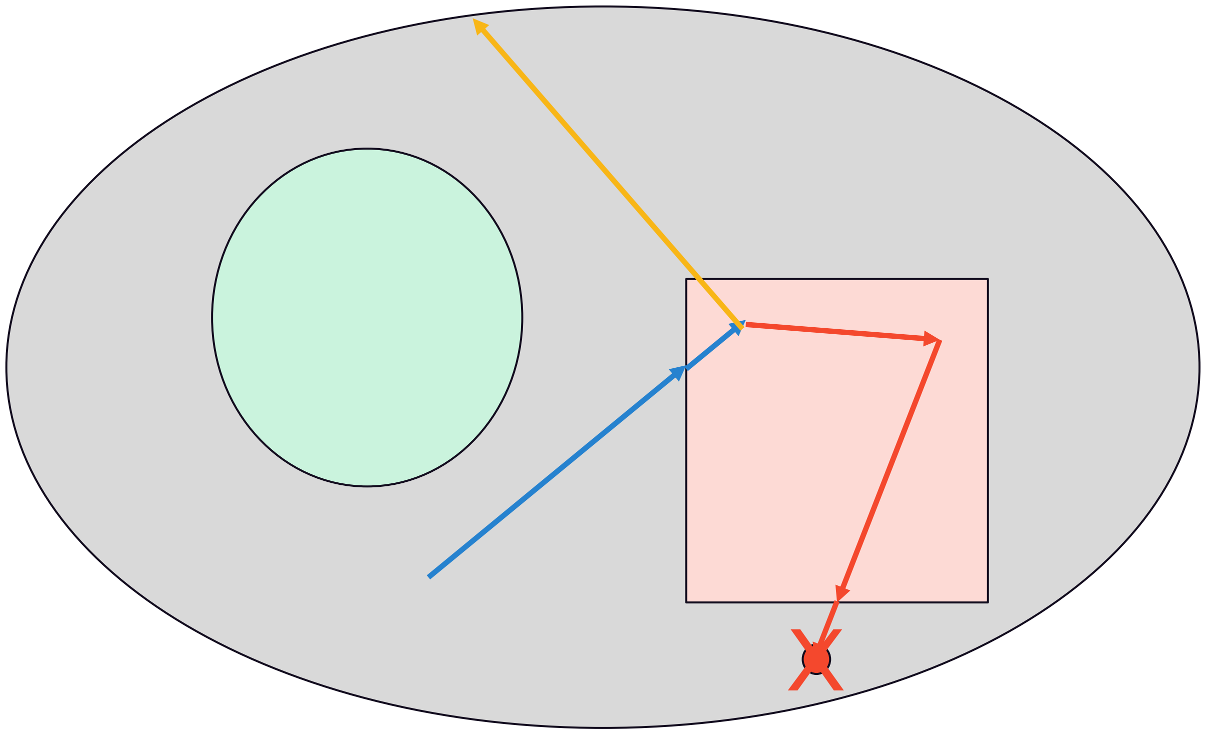
Particle Random Walk: Transport the Particles in the Bank

- Grab a particle from the bank
- Transport it until it dies by leakage or absorption
- Repeat as necessary



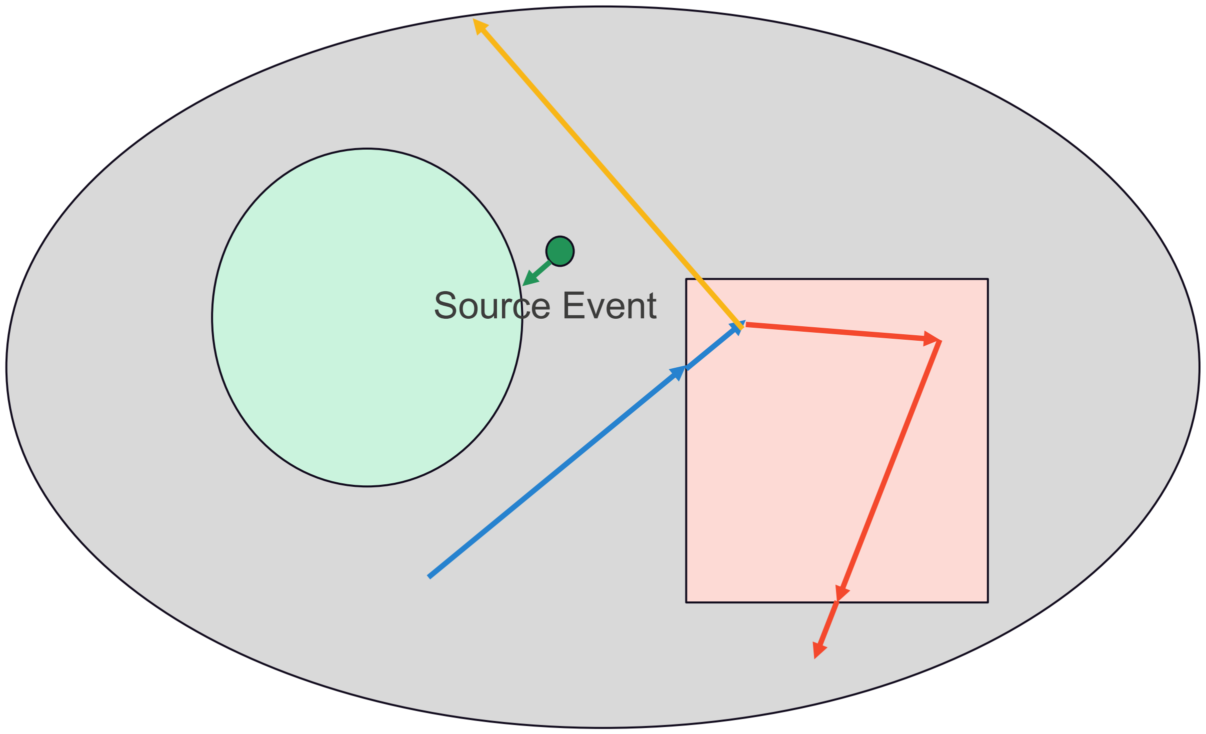
Particle Random Walk: Transport the Particles in the Bank

- Grab a particle from the bank
- Transport it until it dies by leakage or absorption
- Repeat as necessary



Particle Random Walk: Start a new history

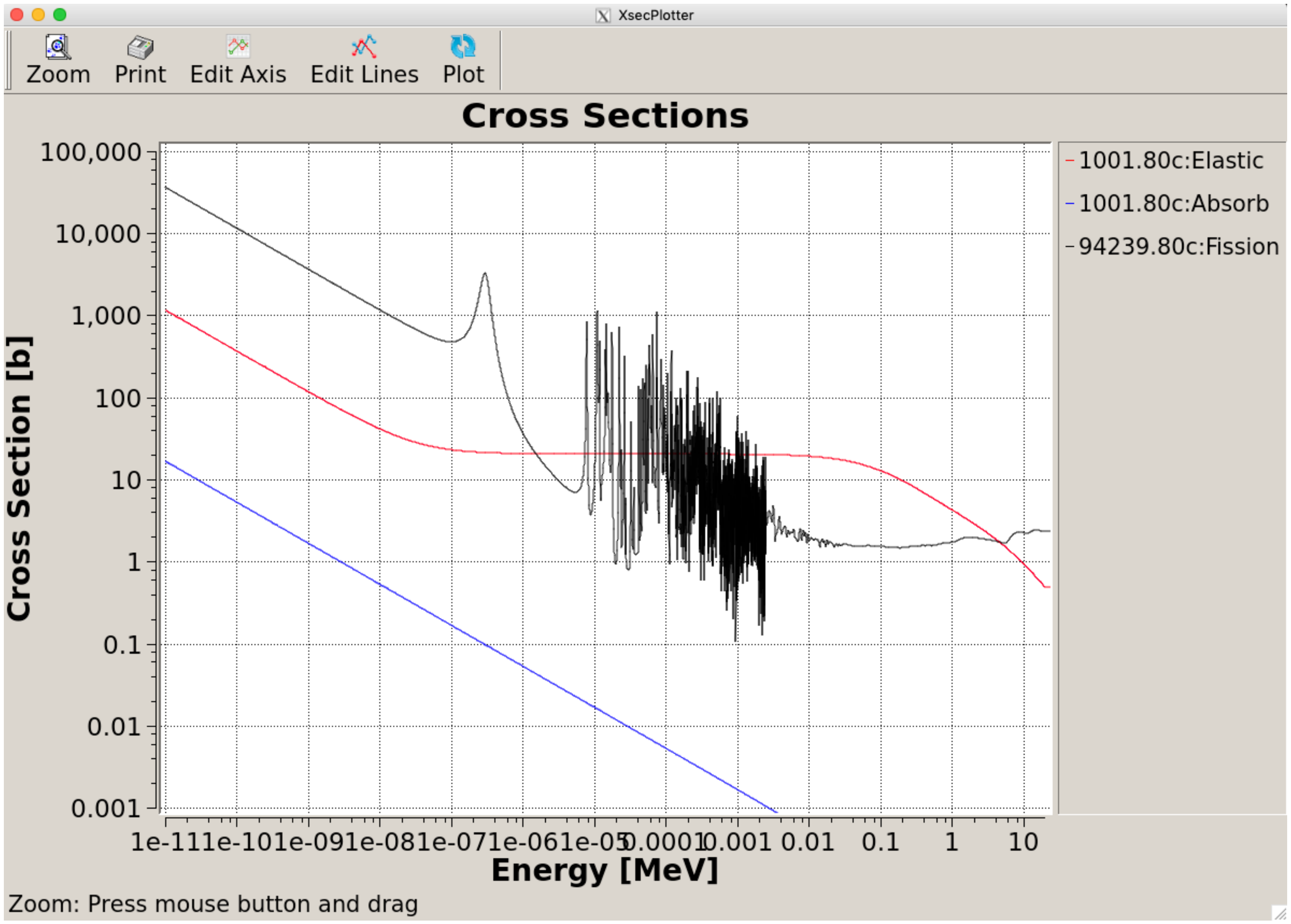
– Sample another source particle and transport it



Continuous Energy Nuclear Data

- Data comes from evaluated nuclear data files (ENDF) (other continuous data types do exist)
- Data is actually point-wise with continuous interpolation
- ENDF laws define outgoing energy and angle distributions for different reaction types and incident energies
- MCATK and MCNP read ACE (A Compact ENDF) formatted files

Continuous Energy Nuclear Data



ACE Format Overview

- A Compact ENDF Format: https://github.com/NuclearData/ACEFormat
- Can be ASCII (Type 1) or binary (Type 2)
- Generated from ENDF nuclear data using the NJOY code
- The nuclear data team in XCP-5 is responsible for creating ACE data files
- ACE data files are intimately tied to MCNP

- Data Types:
  - Neutron
  - Photoatomic
  - Photonuclear
  - Thermal scattering for molecular effects at low incident neutron energies

ACE Format Overview Continued

- "Continuous" or "pointwise" energy data
  - Actually pointwise with up to tens of thousands of energy points for major cross sections depending on the isotope
  - Continuous interpolation of cross sections between energy points
- Individual reaction cross sections and outgoing particle angle and energy information is tabulated at discrete incident particle energies
  - Data typically defined at coarser energy intervals
  - Information must be interpolated for incident particle energies
- ACE files contain additional information:
  - Fission multiplicities
  - Photon production
  - Delayed neutron precursor group information
  - Data covariances (used for sensitivity calculations)
  - More

Types of Neutron Reactions

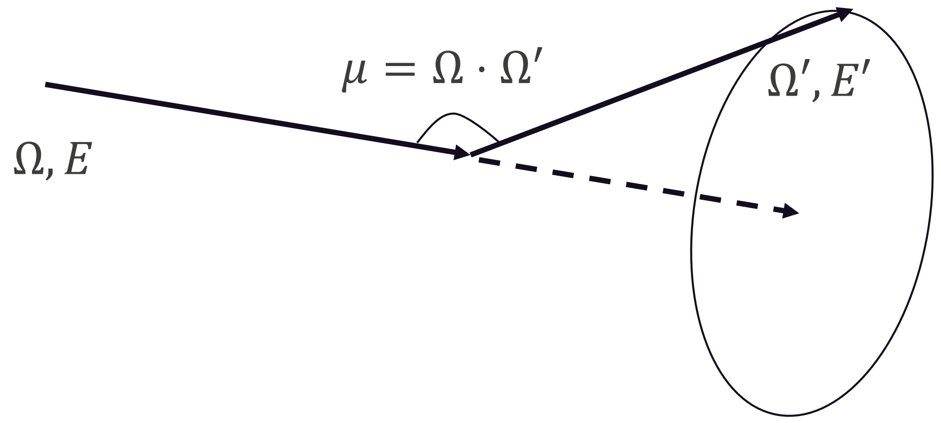
- Capture (n,γ)
- Fission
- Elastic scattering
- Inelastic scattering (n,n’)
- (n,anything)

Reaction Selection

- Select the isotope within the material that the particle collides with:
  - The probability of colliding with each isotope i is: $p_i = \frac{\Sigma_{t,i}}{\Sigma_t}$
- Select the specific collision type:
  - The probability of reaction r in isotope i is: $p_{r,i} = \frac{\Sigma_{r,i}}{\Sigma_{t,i}}$

Outgoing Angle Distributions

- For all collision types with outgoing particles, outgoing angle distributions are defined in terms of the cosine with the incident particle direction
- The final outgoing angle is then uniformly sampled in (0,2π) about this cosine



Capture

- If implicit capture is on, capture events don’t occur (see section on Variance Reduction)
- If implicit capture is off, capture events simply terminate the particle.

Fission

- Use random number to sample fission multiplicity
- Each outgoing neutron’s angle and energy are sampled independently
  - Particles are not correlated
  - Individual reactions are not modeled correctly: can get multiple 10 MeV neutrons if “unlucky” enough
  - Correct average behavior if enough histories are simulated
- Outgoing angular distribution is isotropic

Elastic Scatter

- Elastic cross sections are temperature-dependent
  - They are computed at the material temperature even if no other multi-temperature treatments are used
- If the free gas treatment is on, sample the target nucleus velocity based on the material temperature
- Sample scattering cosine from nuclear data
- Calculate outgoing energy by performing kinematics calculation of the collision between the neutron and nucleus in the center of mass frame

Other Reactions: Inelastic Scatter or (n,anything)

- Nuclear data defines angle and energy distributions of outgoing particles
  - Data can be defined in different ways, known as ENDF laws
  - Example: equiprobable cosine bins between -1 and 1
- Use random numbers to sample the outgoing energy and angle distribution
  - These distributions may or may not be correlated

Eigenvalue and Time-Dependent Algorithms

Boltzmann Transport Equation

$$\frac{1}{v}\frac{\partial\psi}{\partial t}(\boldsymbol{x},\boldsymbol{\Omega},E,t)+\boldsymbol{\Omega}\cdot\boldsymbol{\nabla}\psi(\boldsymbol{x},\boldsymbol{\Omega},E,t)+\Sigma_t(\boldsymbol{x},E,t)\psi(\boldsymbol{x},\boldsymbol{\Omega},E,t)$$
$$=F\psi+S\psi+Q(\boldsymbol{x},\boldsymbol{\Omega},E,t)$$

$$\begin{aligned}\psi(\boldsymbol{x},\boldsymbol{\Omega},E,t)&=f(\boldsymbol{x},\boldsymbol{\Omega},E,t), && x\in\partial V\\ \psi(\boldsymbol{x},\boldsymbol{\Omega},E,0)&=g(\boldsymbol{x},\boldsymbol{\Omega},E), && x\in V\end{aligned}$$

$\boldsymbol{\Omega}\cdot\boldsymbol{\nabla}\psi(\boldsymbol{x},\boldsymbol{\Omega},E,t)$: Streaming

$\Sigma_t(\boldsymbol{x},E,t)\psi(\boldsymbol{x},\boldsymbol{\Omega},E,t)$: Loss due to collisions

$F\psi$: Fission Production

$S\psi$: Inscattering Production

$Q(\boldsymbol{x},\boldsymbol{\Omega},E,t)$: Source

k-Eigenvalue Equation

$$\boldsymbol{\Omega} \cdot \boldsymbol{\nabla}\psi(\boldsymbol{x}, \boldsymbol{\Omega}, E) + \Sigma_t(\boldsymbol{x}, E)\psi(\boldsymbol{x}, \boldsymbol{\Omega}, E) = \frac{1}{k}F\psi + S\psi$$

$$\psi(\boldsymbol{x}, \boldsymbol{\Omega}, E) = f(\boldsymbol{x}, \boldsymbol{\Omega}, E), \qquad x \in \partial V$$

- There are multiple eigenvalues and eigenmodes
  - After some number of fission generations, the largest eigenvalue dominates
  - This is the fundamental eigenvalue and eigenmode
- k is the relative size of one fission generation relative the generation that immediately preceded it

How much does the fission source need to be scaled to maintain a constant population without a source?

k-Eigenvalue Equation

$$\boldsymbol{\Omega} \cdot \boldsymbol{\nabla}\psi(\boldsymbol{x}, \boldsymbol{\Omega}, E) + \Sigma_t(\boldsymbol{x}, E)\psi(\boldsymbol{x}, \boldsymbol{\Omega}, E) = \frac{1}{k}F\psi + S\psi$$

$$\psi(\boldsymbol{x}, \boldsymbol{\Omega}, E) = f(\boldsymbol{x}, \boldsymbol{\Omega}, E), \qquad x \in \partial V$$

- k=1: Critical
- k>1: Supercritical
- k<1: Subcritical

How much does the fission source need to be scaled to maintain a constant population without a source?

α-Eigenvalue Equation

- Assume that the time dependence is separable:

$$\psi(\boldsymbol{x}, \boldsymbol{\Omega}, E, t) = \psi(\boldsymbol{x}, \boldsymbol{\Omega}, E)e^{\alpha t}$$

$$\boldsymbol{\Omega} \cdot \boldsymbol{\nabla}\psi(\boldsymbol{x}, \boldsymbol{\Omega}, E) + \left(\Sigma_t(\boldsymbol{x}, E) + \frac{\alpha}{v}\right)\psi(\boldsymbol{x}, \boldsymbol{\Omega}, E) \;= F\psi + S\psi$$

$$\psi(\boldsymbol{x}, \boldsymbol{\Omega}, E) = f(\boldsymbol{x}, \boldsymbol{\Omega}, E), \qquad x \in \partial V$$

- There are multiple eigenvalues and eigenmodes, some of which are complex
  - After some amount of time, the right-most (always real) eigenvalue dominates and defines the exponential change in the flux over time
  - This is the fundamental eigenvalue and eigenmode

What is the asymptotic exponential rate of growth of the flux?

α-Eigenvalue Equation

$$\psi(\boldsymbol{x}, \boldsymbol{\Omega}, E, t) = \psi(\boldsymbol{x}, \boldsymbol{\Omega}, E) e^{\alpha t}$$

$$\boldsymbol{\Omega} \cdot \boldsymbol{\nabla} \psi(\boldsymbol{x}, \boldsymbol{\Omega}, E) + \left(\Sigma_t(\boldsymbol{x}, E) + \frac{\alpha}{v} \right) \psi(\boldsymbol{x}, \boldsymbol{\Omega}, E) = F\psi + S\psi$$

$$\psi(\boldsymbol{x}, \boldsymbol{\Omega}, E) = f(\boldsymbol{x}, \boldsymbol{\Omega}, E), \qquad x \in \partial V$$

- $\alpha = 0$: Critical
- $\alpha > 0$: Supercritical
- $\alpha < 0$: Subcritical

What is the asymptotic exponential rate of growth of the flux?

α-Eigenvalue Equation

$$\psi(\boldsymbol{x},\boldsymbol{\Omega},E,t) = \psi(\boldsymbol{x},\boldsymbol{\Omega},E)e^{\alpha t}$$

$$\boldsymbol{\Omega}\cdot\boldsymbol{\nabla}\psi(\boldsymbol{x},\boldsymbol{\Omega},E) + \left(\Sigma_t(\boldsymbol{x},E) + \frac{\alpha}{v}\right)\psi(\boldsymbol{x},\boldsymbol{\Omega},E) \;= F\psi + S\psi$$

$$\psi(\boldsymbol{x},\boldsymbol{\Omega},E) = f(\boldsymbol{x},\boldsymbol{\Omega},E), \qquad x \in \partial V$$

- $\alpha > 0$: $\frac{\alpha}{v}$ is known as time absorption
  - Low energy neutrons are “absorbed” at a greater rate
  - Represents the lag between when neutrons are born at fission energies and how long it takes them to reach low energies through scattering
  - Effectively, neutrons are created at fission energies at a higher rate than they are reaching thermal energies
- $\alpha < 0$: $\frac{\alpha}{v}$ is known as time source
  - Low energy neutrons are “time sourced” at a greater rate
  - Effectively, neutrons reaching thermal energies at a higher rate than they are being created at fission energies

Comparison of Forms of the Transport Equation

$$\frac{1}{v}\frac{\partial\psi}{\partial t}(\boldsymbol{x},\boldsymbol{\Omega},E,t) + \boldsymbol{\Omega}\cdot\boldsymbol{\nabla}\psi(\boldsymbol{x},\boldsymbol{\Omega},E,t) + \Sigma_t(\boldsymbol{x},E,t)\psi(\boldsymbol{x},\boldsymbol{\Omega},E,t) \ = F\psi + S\psi$$

$$\boldsymbol{\Omega}\cdot\boldsymbol{\nabla}\psi(\boldsymbol{x},\boldsymbol{\Omega},E) + \Sigma_t(\boldsymbol{x},E)\psi(\boldsymbol{x},\boldsymbol{\Omega},E) \ = \frac{1}{k}F\psi + S\psi$$

$$\boldsymbol{\Omega}\cdot\boldsymbol{\nabla}\psi(\boldsymbol{x},\boldsymbol{\Omega},E) + \left(\Sigma_t(\boldsymbol{x},E) + \frac{\alpha}{v}\right)\psi(\boldsymbol{x},\boldsymbol{\Omega},E) \ = F\psi + S\psi$$

- All forms of the equations are the same for a perfectly critical system
- The energy spectra of the two eigenvalue solutions differ
  - The α-eigenvalue equation modifies the energy spectrum in a physically meaningful way, i.e., it matches the true asymptotic solution of the time-dependent equation
  - The k-eigenvalue equation does not, and so is wrong far from criticality

k-Eigenvalue Algorithm

$$\boldsymbol{\Omega} \cdot \boldsymbol{\nabla}\psi(\boldsymbol{x}, \boldsymbol{\Omega}, E) + \Sigma_t(\boldsymbol{x}, E)\psi(\boldsymbol{x}, \boldsymbol{\Omega}, E) = \frac{1}{k}F\psi + S\psi$$

- Effectively a power iteration
- Simulate one fission generation at a time
  - Sample a user-specified particles from the fission source
  - Track particles until they are killed or undergo fission
  - Store the fission sites for the next cycle/generation
  - Repeat
- Every cycle, compute k:

$$k = \frac{\langle F\psi \rangle_{i+1}}{\langle F\psi \rangle_i}$$

- Average k over some number of cycles

k-Eigenvalue Convergence

$$\boldsymbol{\Omega} \cdot \boldsymbol{\nabla}\psi(\boldsymbol{x}, \boldsymbol{\Omega}, E) + \Sigma_t(\boldsymbol{x}, E)\psi(\boldsymbol{x}, \boldsymbol{\Omega}, E) = \frac{1}{k}F\psi + S\psi$$

- Start with an initial guess of the fission source
- For the early generations, the fission source and eigenvalue are wrong
  - Allow some number of “inactive” cycles for the fission source to settle into the fundamental mode
  - Do not include cycle estimates of k in the average
- The eigenvalue typically converges faster than the eigenmode
  - Looking only at the eigenvalue can lead to false convergence or bad tallies
  - MCATK computes a “radius of gyration”
    - Average distance between all particles and the centroid of their location
    - Convergence of the radius of gyration indicates that the fission source has spatially converged

α-Eigenvalue Algorithm

$$\boldsymbol{\Omega} \cdot \boldsymbol{\nabla}\psi(\boldsymbol{x}, \boldsymbol{\Omega}, E) + \left(\Sigma_t(\boldsymbol{x}, E) + \frac{\alpha}{v}\right)\psi(\boldsymbol{x}, \boldsymbol{\Omega}, E) = \frac{1}{k}F\psi + S\psi$$

- $k - \alpha$ iteration
- Write the transport equation with both eigenvalues
- Make an initial guess of α (often 0)
- Hold α fixed (i.e., modify the total cross section) and perform a standard k calculation (with only a few cycles)
- Update the guess of α and repeat
- When $k = 1$, we have arrived at the α eigenvalue

α-Eigenvalue Convergence

$$\mathbf{\Omega} \cdot \boldsymbol{\nabla}\psi(\boldsymbol{x}, \mathbf{\Omega}, E) + \left(\Sigma_t(\boldsymbol{x}, E) + \frac{\alpha}{v}\right)\psi(\boldsymbol{x}, \mathbf{\Omega}, E) = \frac{1}{k}F\psi + S\psi$$

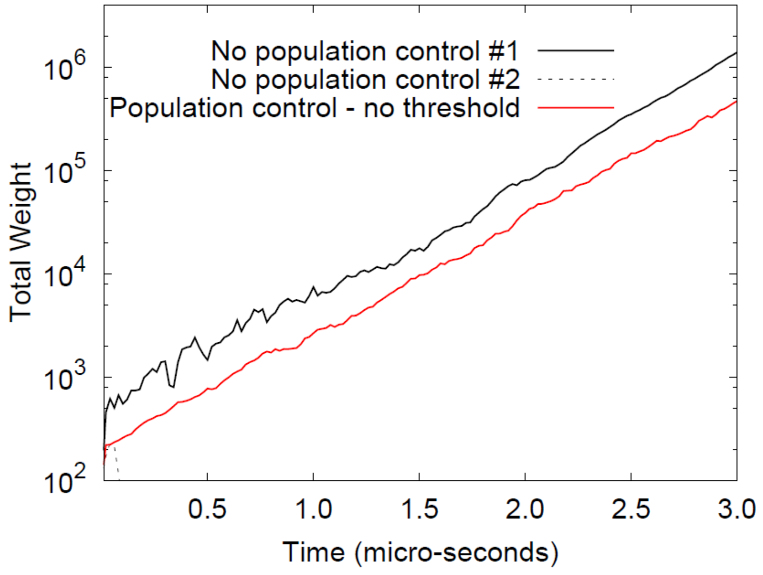
- Perform an initial k-eigenvalue calculation to get a reasonable fission source and initial k to update α with
- After this, only need a few k iterations per α iteration
- Use inactive and active α iterations, only average over active

Time-Dependent Algorithm

- Problems are driven by a user-defined, possibly time-dependent source
- Simulations are discretized in time
- Geometry can potentially change between time steps
  - Mesh is constant during a time step

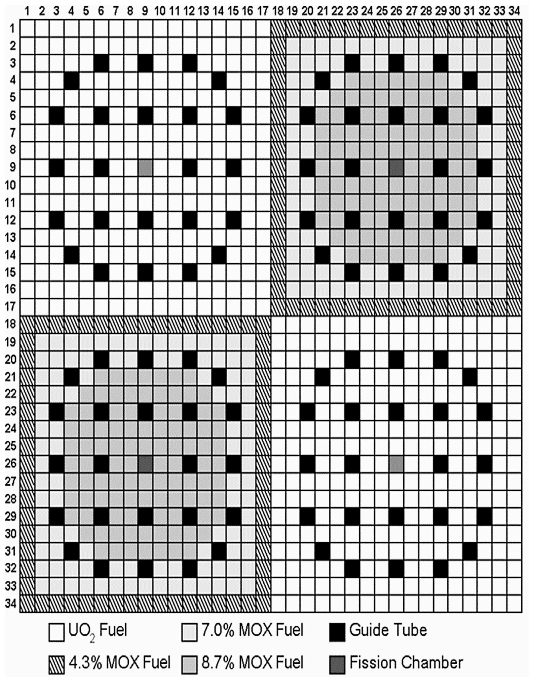
Population Control Algorithm

- When particles reach the end of a time step, they are added to a bank called the "census"
- Population control is applied between time steps
  - Maintains approximately the same number of computational particles in every time step
  - Weight of the particles is adjusted to keep the correct physical number of particles
  - Enables modeling of subcritical problems by keeping simulated particle population from dying out
  - Enables modeling of supercritical problems by keeping simulated particle population from exceeding memory limitations
- Some of the codes that use this algorithm are: TART, Serpent, McCARD, Mercury, MCATK

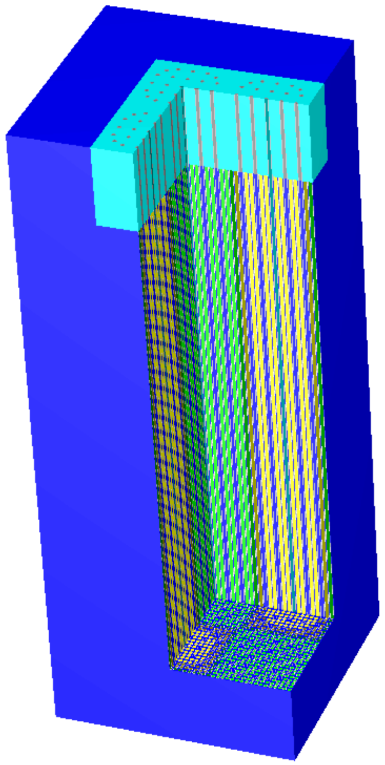


C5G7 Transient Benchmark

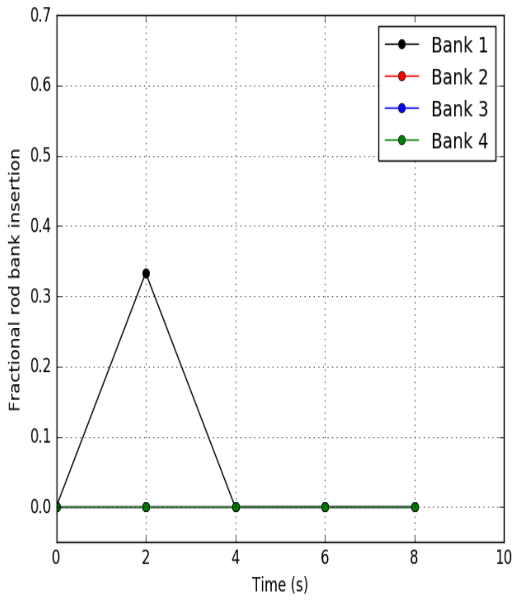
- Four assembly reactor (2 MOX, 2 UO2)
- Control rod banks inserted from the top
- Using the continuous energy version of the benchmark EXCEPT using the multigroup delayed neutron emission spectra



Assembly Compositions



3D Configuration

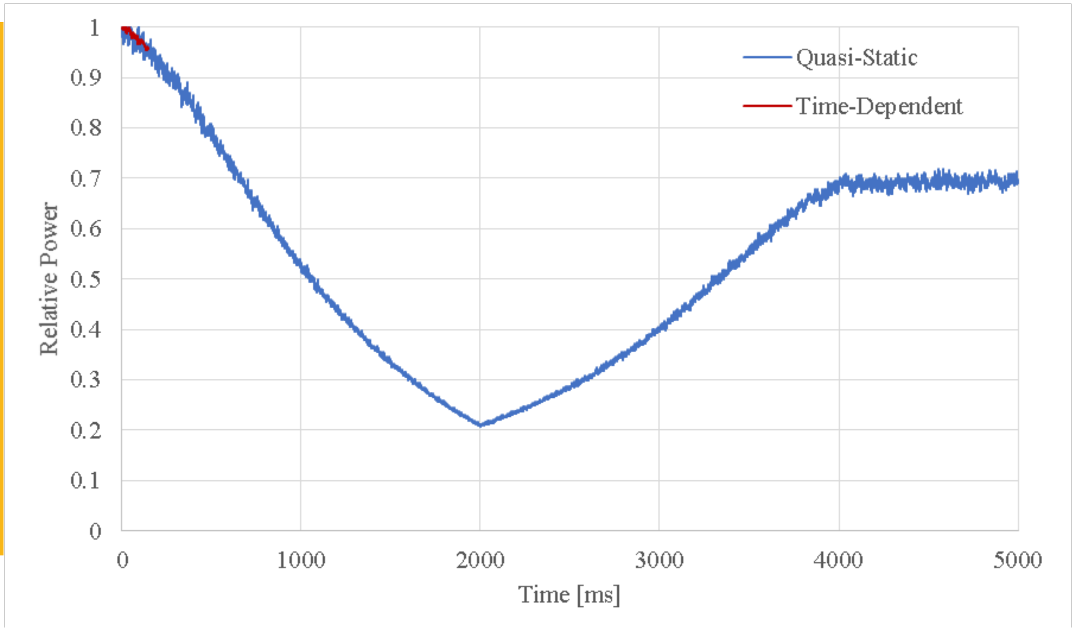


Transient Problem TD4-1

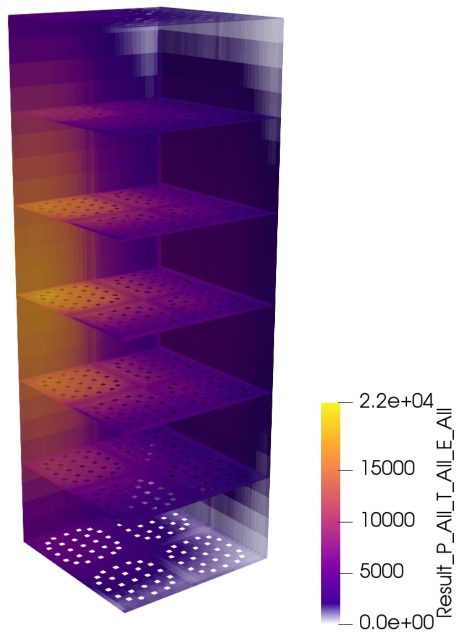
C5G7 Transient Benchmark Results

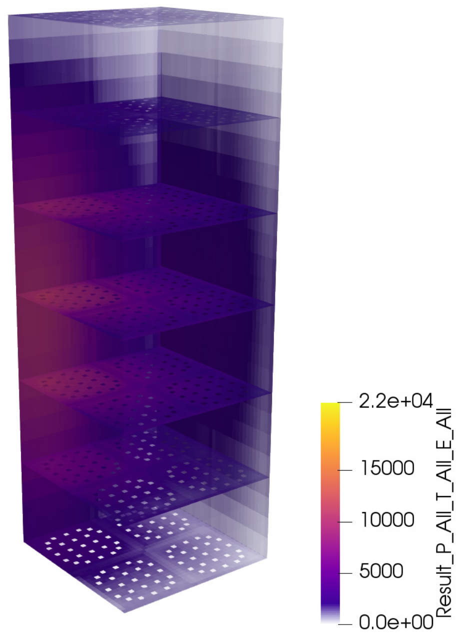
- Run quasi-static: 10 μs micro time steps, 1 micro step per macro step, 100 μs macro time steps
- Total powers are tallied every micro step
- 3-D Powers are tallied separately from the main calculation for 100 x 10 μs time steps
- NOTE: Power should rise slowly after 4s; the cause of the incorrect flat behavior is known but not yet fixed

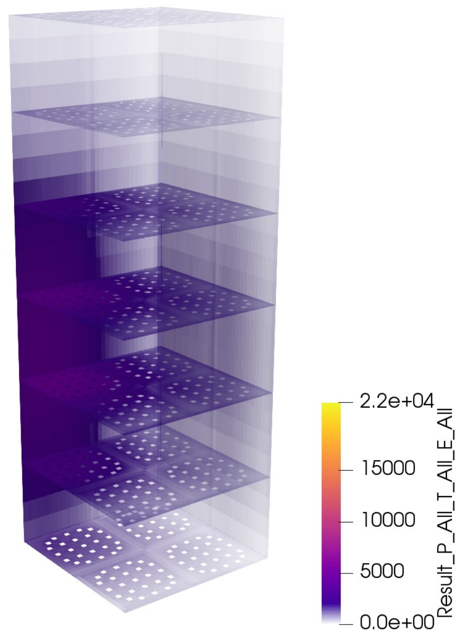
Need to rerun this with some changes: no prompt neutron census, longer time steps, separate population control for each DNP group

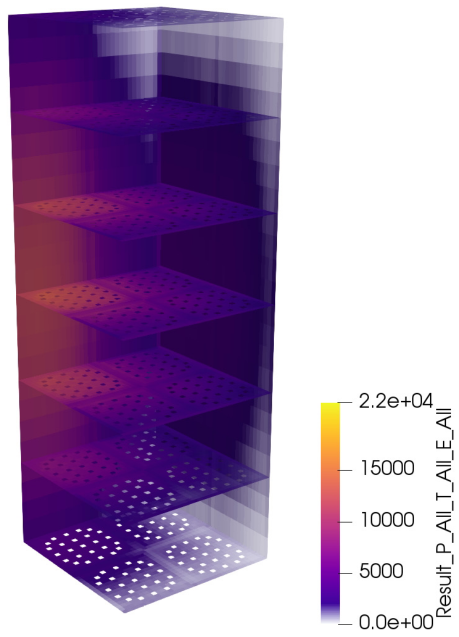


C5G7 Transient Benchmark Results









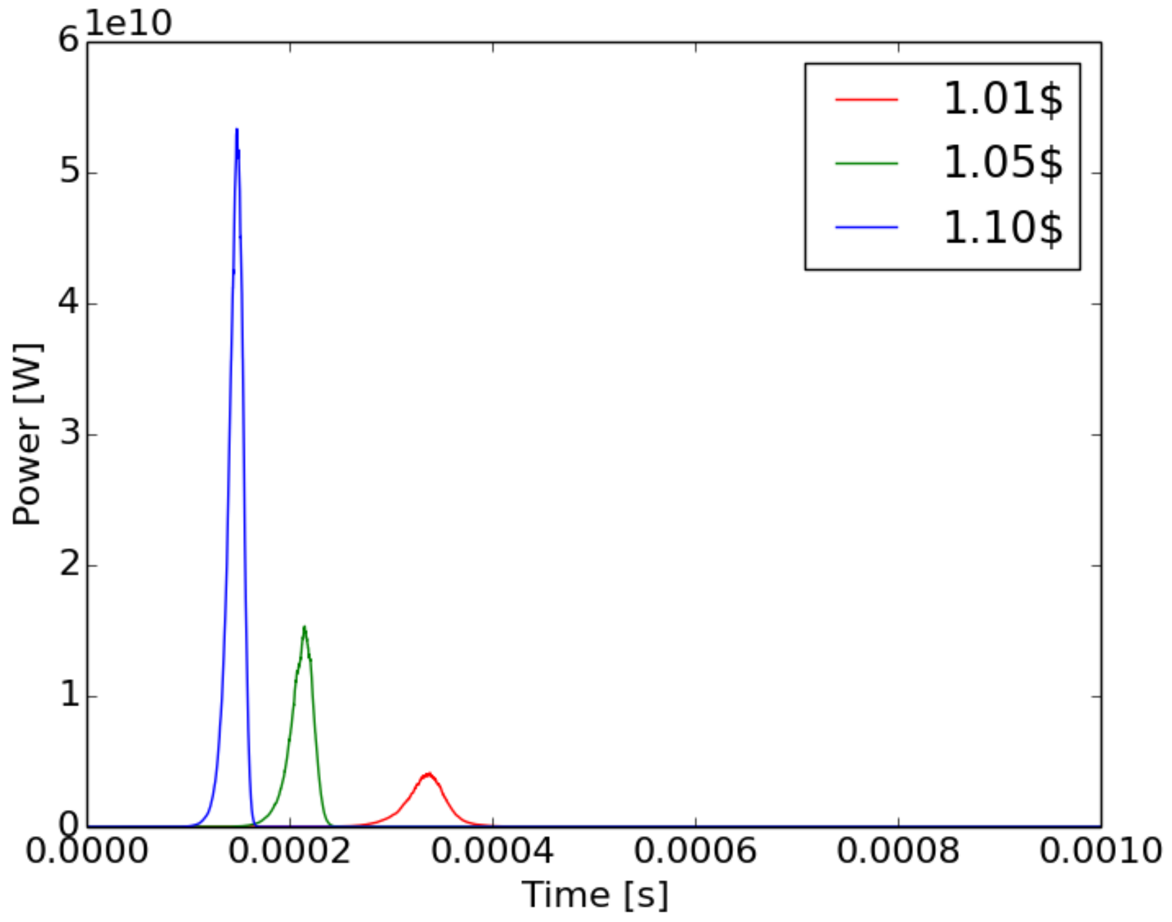
0 s 1 s 2 s 4 s

Multiphysics Simulations of Lady Godiva

- **MCATK has been coupled to a simple 1-D thermomechanics solver**
- **Godiva is simulated using a pseudo-1-D domain.**
  - Neutron transport is done in 3-D.
  - Multiphysics is done over a 1-D, spherically symmetric domain.
- **Simulations are shown for $1.01, $1.05, and $1.10 insertion reactivities.**
  - Temperature and density are initially uniform.
  - Mesh is initially undeformed.

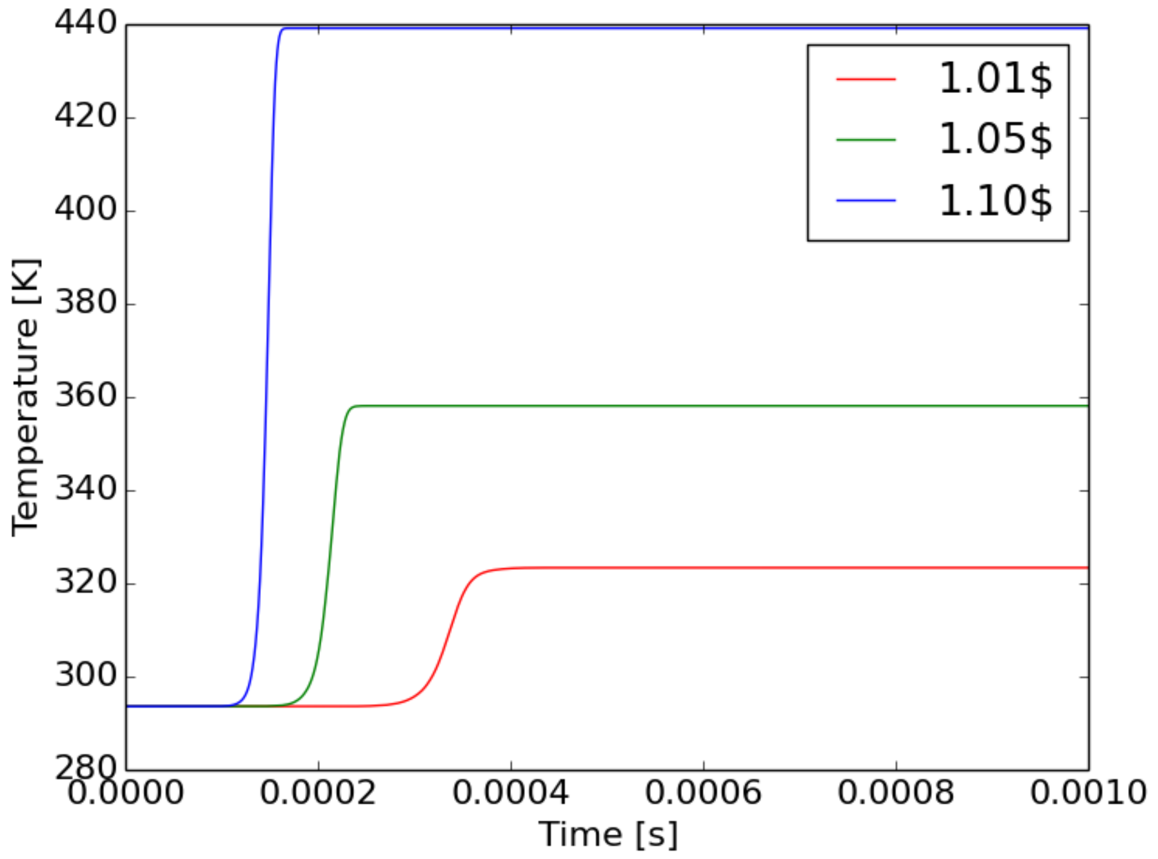
Results

- Prompt super-critical pulses elevate power output for a brief period.



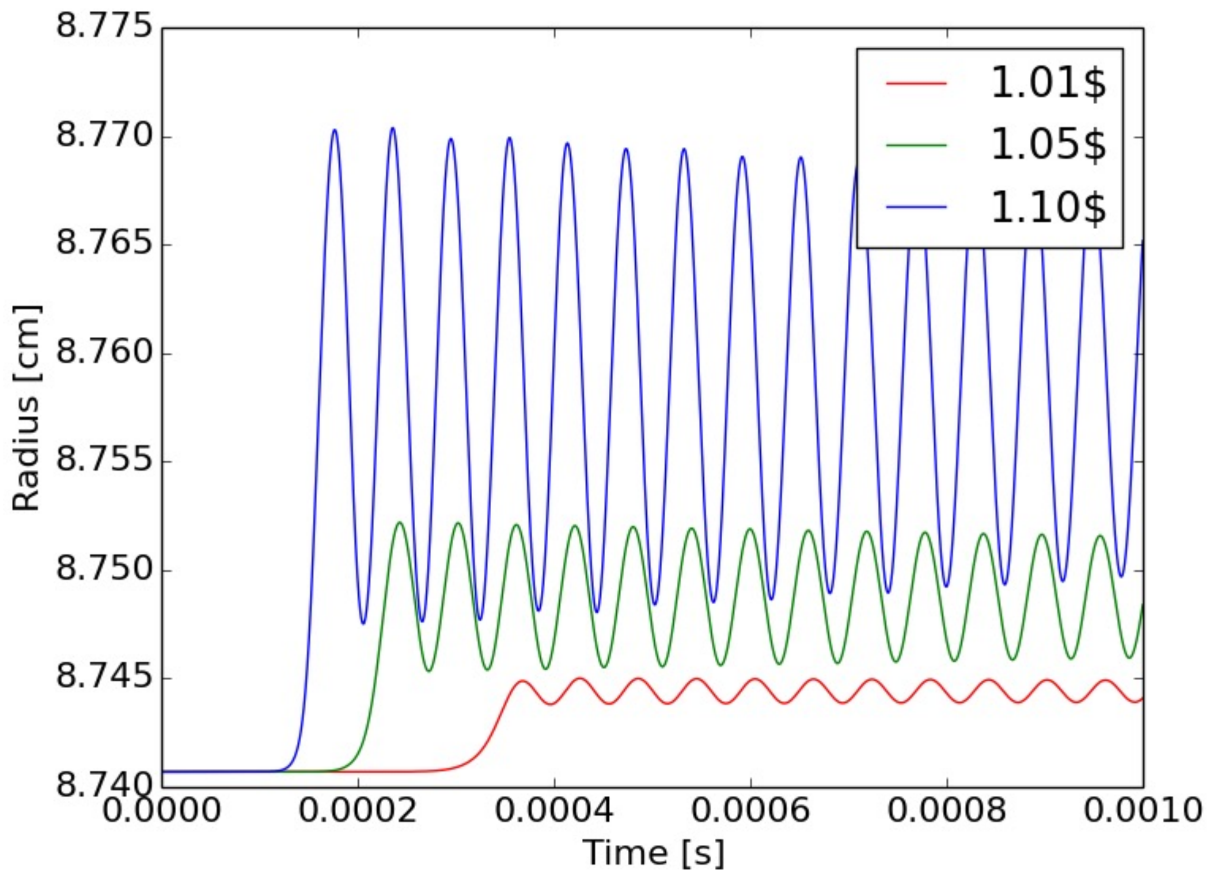
Results

- Average temperature rises due to neutron thermalization and remains constant due to assumption of no external radiation.



Results

- Once expansion due to rising temperatures has ceased, radial oscillations occur due to the assumption of a purely elastic medium.



Advantages and Disadvantages of Monte Carlo Methods

Differences Between Monte Carlo and Deterministic Methods

Monte Carlo

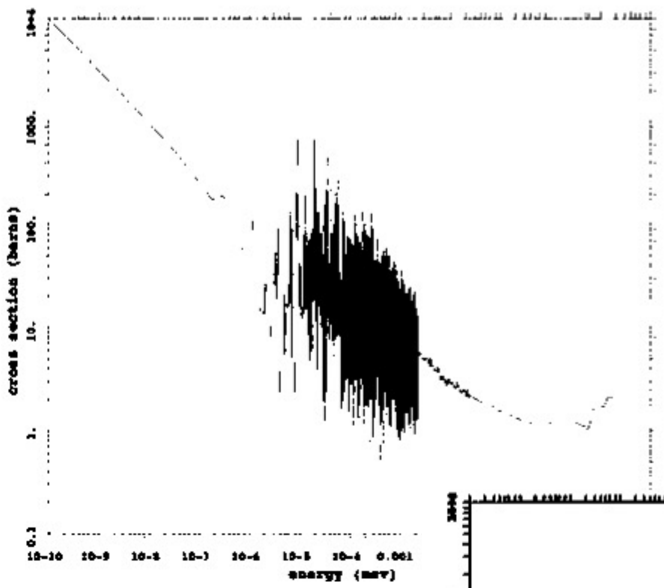
- **Continuous energy data**
- **Continuous in angle**
- **Supports constructive solid geometries, structured and unstructured meshes**
- **Statistical errors**
  - Reduced by variance reduction and/or running more particles
- **Typically more computationally expensive**

Deterministic

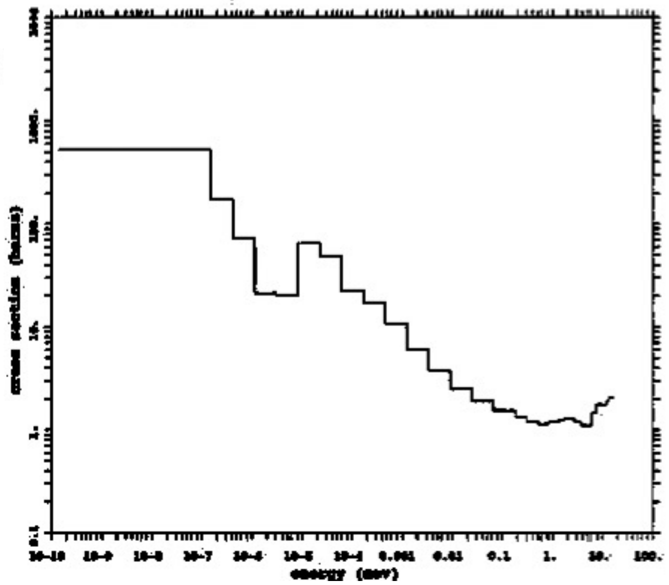
- **Multigroup Data**
- **Discretized in angle**
- **Structured and unstructured mesh geometries (except for MOC, which can use more general surfaces)**
- **Discretization errors**
  - Reduced by using more energy groups, angles
- **Typically less computationally expensive**

Why Use Monte Carlo?

- **No ray effects**
- **The multigroup approximation is a BIG one**
  - Multigroup cross sections are only truly correct if the weight function they are generated with is the actual flux solution – THIS IS PROBLEM DEPENDENT
  - Multigroup cross sections obscure resonance and self-shielding effects, though approximations exists to try and account for these



Continuous Energy 92235.80c Fission

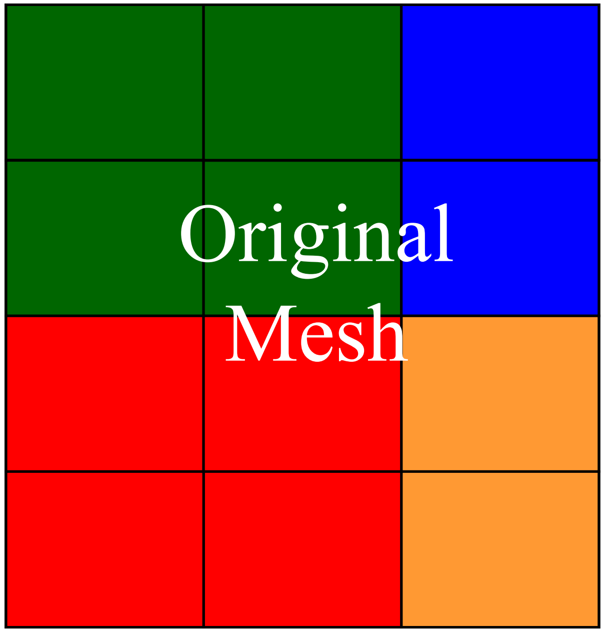


Multigroup Energy 92235.50m Fission

Monte Carlo Parallelism

- Monte Carlo is sometimes referred to as "embarrassingly parallel"
  - Can simply perform different histories on different compute units
  - However, large meshes and tallies may still require domain decomposition
  - Domain decomposition is problematic because particles can bounce back and forth across boundaries, and load balancing is poor

- Types of parallelism:
  - Domain-replicated (particle-decomposed only)
  - Domain-decomposed
  - Hybrid (decompose problem in space, but replicate some or all domains)
  - Shared-memory (compatible with any of the above, share some information, e.g., mesh definition, across all processes on the same compute node)



GPU Tallies: Why are Advanced Architectures Challenging for Monte Carlo?

- Advanced architectures often use Graphics Processing Units (GPUs)
- GPUs are especially common
  - GPUs have thousands of threads organized into groups called “warps”
  - Every thread in a warp must execute the same commands; “thread divergence” within a warp significantly decreases performance
  - Data must be transferred back-and-forth between CPU and GPU
  - GPUs have significantly lower memory per core
  - Code must be compiled specifically for the GPU and often requires using special languages (e.g., CUDA)
- Monte Carlo algorithms are not well suited for these architectures:
  - Random behavior means thread divergence is ubiquitous; event-based/vector Monte Carlo algorithms are available but require significant rewriting of code
  - Continuous energy data and tallying are very memory intensive

Tallies

Monte Carlo Tallies

- Tallies are how we obtain “the answer”
- Can record an almost limitless information from each particle history
- Tallies are an accumulation of information over all particle histories (or batches of particle histories)
  - The accumulation is computing an integral
  - The accumulation must be made over volumes in phase space, not discrete points

Users shouldn’t limit their thinking to what can currently be computed, but what would they like to compute - it may be possible

Example Tally Definition: Flux

- A tally is an integral quantity. For example, a flux tally over all phase space is of the form:

$$Tally = \sum_{particle\_type} \int_0^{\infty} \int_0^{\infty} \int_0^{2\pi} \int_{-1}^{1} \int_V F_p(\boldsymbol{x}, \boldsymbol{\Omega}, E, t)\psi_p(\boldsymbol{x}, \boldsymbol{\Omega}, E, t) dV\, d\mu\, d\theta\, dE\, dt$$

- ψ is the angular flux and is a measure of track-length traversed per unit time and unit volume; it can be measured by summing the track-lengths of the particles
- Flux is an important quantity because many quantities of interest are proportional to it
  - Reaction rates are flux times macroscopic cross section

Filters and Scores

- $F_p(\boldsymbol{x}, \boldsymbol{\Omega}, E, t)$ is a multiplier function or "score".
- For example, if $F_p(\boldsymbol{x}, \boldsymbol{\Omega}, E, t) = \Sigma_r(\boldsymbol{x}, \boldsymbol{\Omega}, E, t)$, i.e., a cross section, then the tally is a reaction rate.

- The tally phase space is typically divided into regions, i.e., tally bins.
- A "filter" is a function that takes a particle from anywhere in phase space and determines what tally bin, if any, the particle should score to.

- A tally for a particular particle type p, in spatial bin i, angular bin n, energy group bin g, and time bin j is expressed as:

$$Tally_{p,i,n,g,j} = \int_{t_{j-1}}^{t_j} \int_{E_{g-1}}^{E_g} \int_0^{2\pi} \int_{\mu_{n-1}}^{\mu_{n-1}} \int_{V_i} F_p(\boldsymbol{x}, \boldsymbol{\Omega}, E, t) \psi_p(\boldsymbol{x}, \boldsymbol{\Omega}, E, t) dV\, d\mu\, d\theta\, dE\, dt$$

Statistical Noise of Monte Carlo Tallies

- If we simulate enough particles, the tally result will converge to the correct average behavior
- Every tally will have a variance, σ^2, associated with it
- The standard deviation, σ, decreases slowly as the number or histories, N, is increased:

$$\sigma \propto \frac{1}{\sqrt{N}}$$

- Variance reduction techniques can be used to reduce the statistical noise for the same number of histories

Tally Figure of Merit

- A standard figure of merit for Monte Carlo tallies is:

$$FOM = \frac{1}{\sigma^2 T}$$

- This FOM should not change as number of histories, N, changes; the time, T, scales linearly with N, while the variance is inversely proportional to N.
- The FOM can be improved by:
  - Tallying the same quantity with a different tally type (e.g., track-length vs. event tally),
  - Using variance reduction to decrease variance faster than run-time increases.

The figure of merit is a measure of the efficiency of a calculation that is invariant with respect to the number of histories.

Types of Tallies

Event Tallies

- Record detailed information about the events in a particle’s history:
  - Surface crossing events
  - Collision events
    - Incident particle states
    - Outgoing particle states
    - Multiplication
  - Source events
- Event tallies are generally the noisiest
  - Not every event happens during every particle track
  - Rare events are hard to sample

Track-Length Tallies

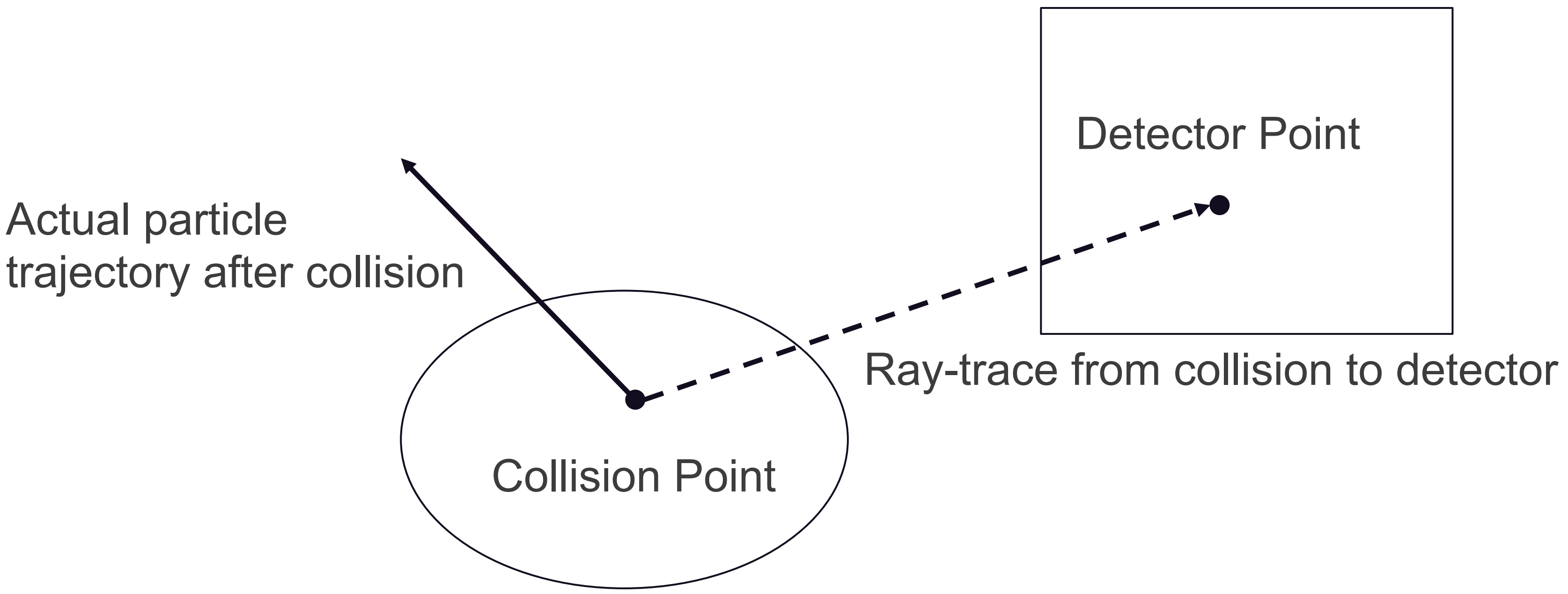
- Record quantities that are proportional to the distance traveled by every particle:
  - Flux
  - Reaction rate
  - Energy deposition
- Track-length tallies are generally less noisy
  - Every step in a particle history generates path length
  - Can tally reaction rates along every track segment no matter how rare the reaction is
  - Parts of phase space can still be hard to sample (e.g., certain energy ranges or well-shielded regions)

Expected Value Tallies

- Calculate the probability of a track or event occurring from a source or collision point
  - Often involves ray-tracing from a source or collision point
  - Example: Can calculate the probability of an outgoing particle reaching a point or surface by computing the probability of traveling in the direction of the target and then ray-tracing to the point with attenuation
- Expected-value tallies are generally the least noisy
  - Can calculate probabilities from any source or collision point no matter how rare the tallied outcome is (e.g., can always calculate the probability of penetrating a shield even though few particle histories actually do so)
  - Must still have adequate sampling of the source and collision distributions
- Expected value tallies can be very expensive

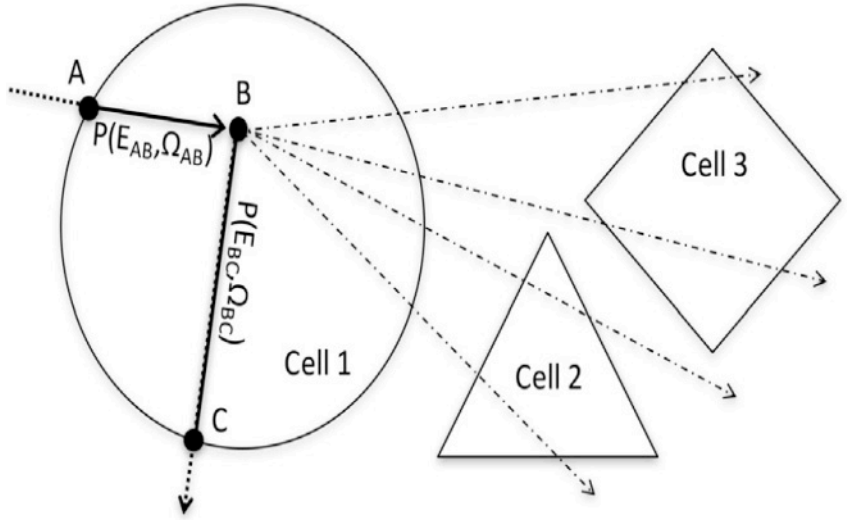
Expected Value Tallies: Next Event Estimators

- Also known as point-detector tallies.
- Estimates the flux at a point:
  - At each source and collision event, calculate the probability of scattering precisely in the direction of the point,
  - Ray-trace through geometry to calculate the probability of reaching the point without a collision.


Detector Point
Actual particle trajectory after collision
Ray-trace from collision to detector
Collision Point

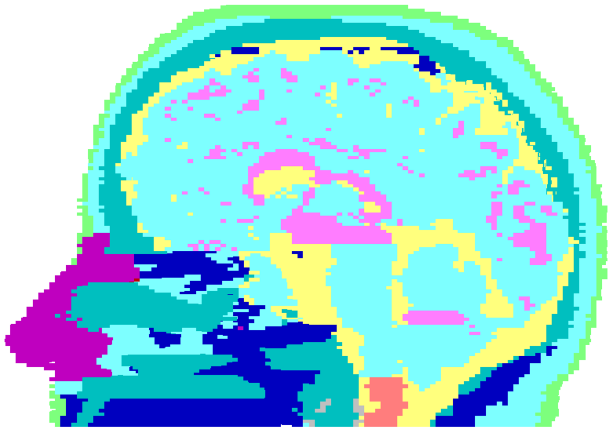
Expected Value Tallies: Volumetric Ray-Casting Estimators

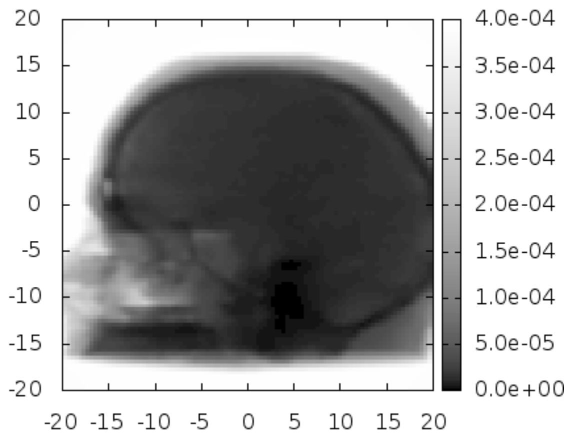
- Also referred to as expected-path-length tallies.
- Sample multiple pseudo-rays at each source and collision event and perform a ray trace to generate expected track-length contributions to volumetric tallies.
- These rays are transferred to the GPU (many rays are buffered and sent together)
- GPU then traces the rays through the geometry and scores to the tally
- Provides lower variance solution than standard track-length tally.



MonteRay Tally Library

- Supports performing tallies on GPUs asynchronously with transport on CPUs
- Takes a list of rays from a Monte Carlo transport code and performs one of two tallies based on ray tracing:
  - Next event estimators (point detectors)
  - Volumetric-ray-casting estimator
- Open-source distribution



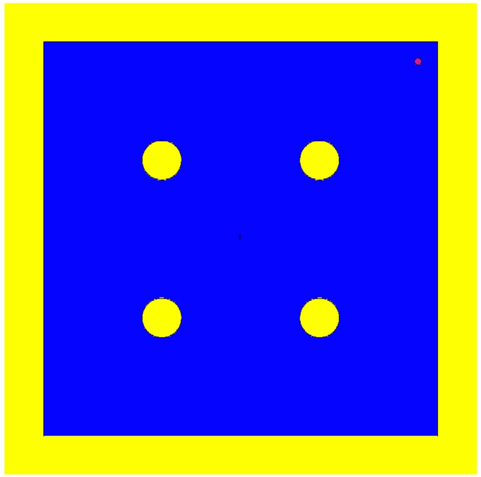


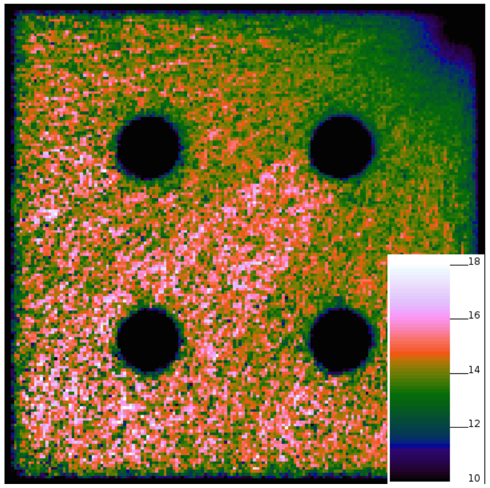
Simulated Radiograph of Human Head (15x performance)

Volumetric Ray-Casting Estimator Results

Performance of volumetric-ray-casting estimator for a room-sized shielding problem.

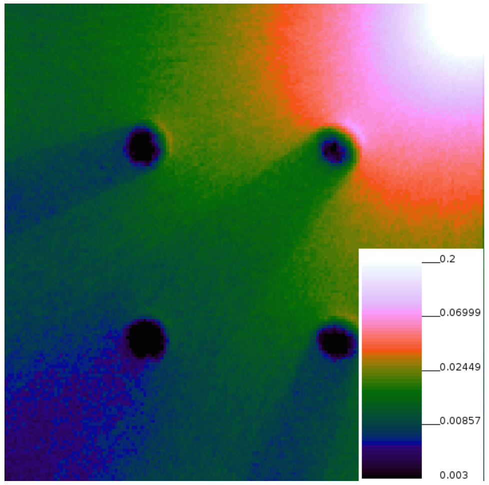
Geometry: U Sphere in a Concrete Room

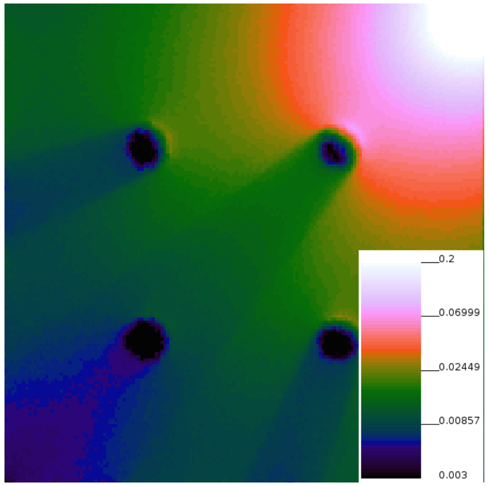




Relative Figure of Merit of GPU vs. 8 CPU Cores

Standard Track-Length Flux Tally





Ray-Casting Flux Tally

Using Variance Reduction to Improve Efficiency

Variance Reduction Motivation

- Monte Carlo tallies have a statistical noise measured by variance, σ^2
- A standard figure of merit for Monte Carlo tallies is:

$$FOM = \frac{1}{\sigma^2 T}$$

- This FOM should not change as number of histories, N, changes; the time, T, scales linearly with N, while the variance is inversely proportional to N.
- Variance reduction attempts to improve the FOM by increasing the number of score events to a tally
  - For the same number of histories, runtime may or may not increase
  - Variance should decrease faster

The figure of merit cannot be improved by increasing the number of histories, but can be improved using variance reduction.

Variance Reduction Basics

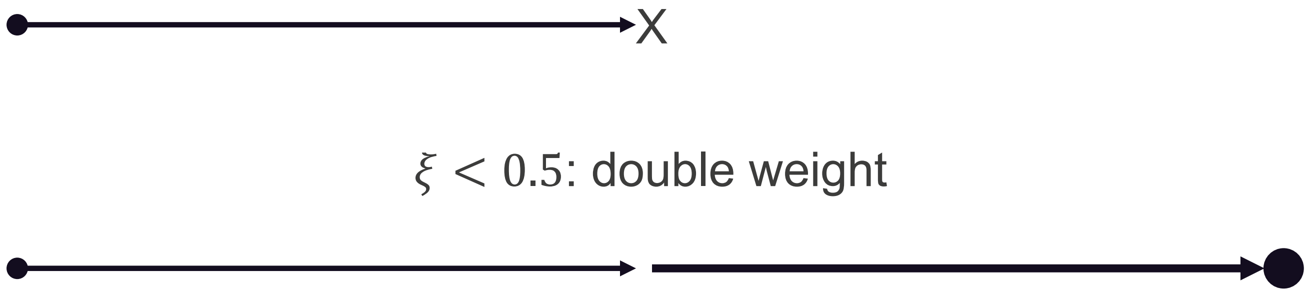
- Give each simulated particle a weight
- Bias aspects of each history to increase the probability of scoring to a tally
  - Bias the source
  - Bias collision sampling
  - Bias phase-space during transport
- Adjust the weight of the particles so that the tally converges to the same result

Russian Roulette

- Used to reduce the number of sample particles
- Use random numbers to determine of particle is killed or not
  - If the particle loses the roulette game, it is terminated
  - If the particle survives the roulette game, increase its weight
  - On average, preserve the weight in the problem

- Example: survival probability of 0.5

$\xi > 0.5$: kill particle

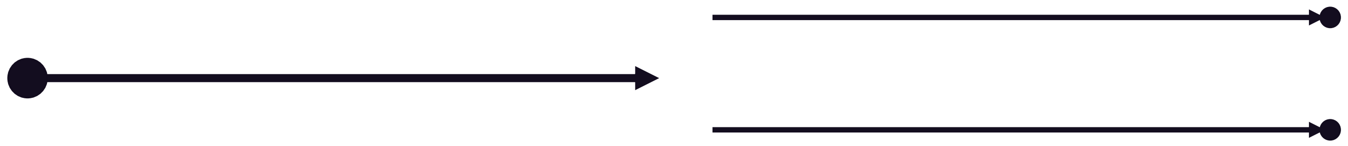


$\xi < 0.5$: double weight

Splitting

- Used to increase the number of sample particles
- Split a single simulated particle into multiple particles
- Divide the weight among the particles (typically equally)

- Example: Split by factor of two, halve the weight



Implicit Capture

- Do not let particle histories be terminated by capture events
- Instead, decrease the weight of the particle by the probability that a capture event would have occurred:

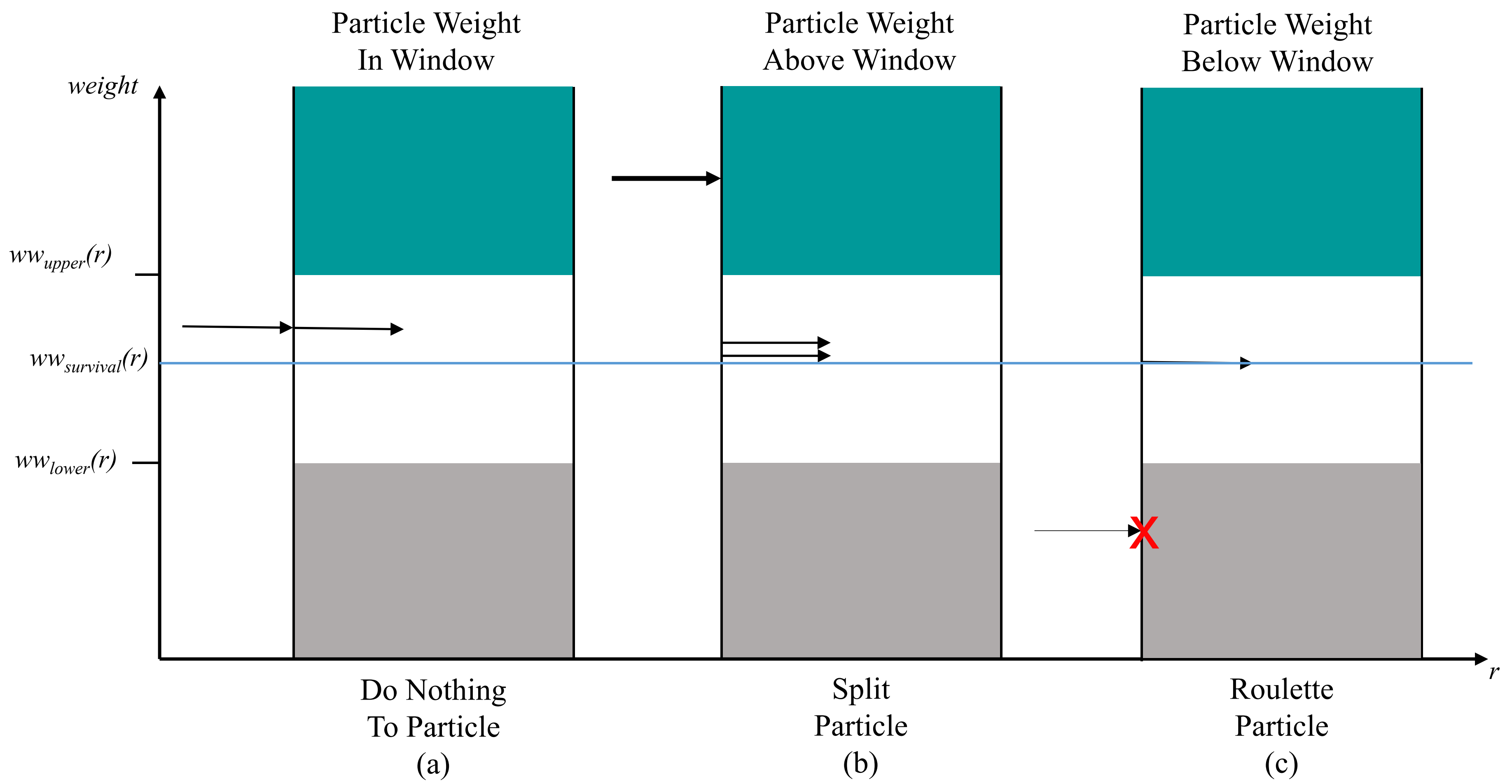
$$wt_{exit} = wt_{incident}\left(1 - \frac{\sigma_{capture}}{\sigma_{total}}\right)$$

- Extends each history and increase the number of tracks/collisions per history
- Play Russian roulette when particle weight drops below a user-specified cutoff

Weight Windows

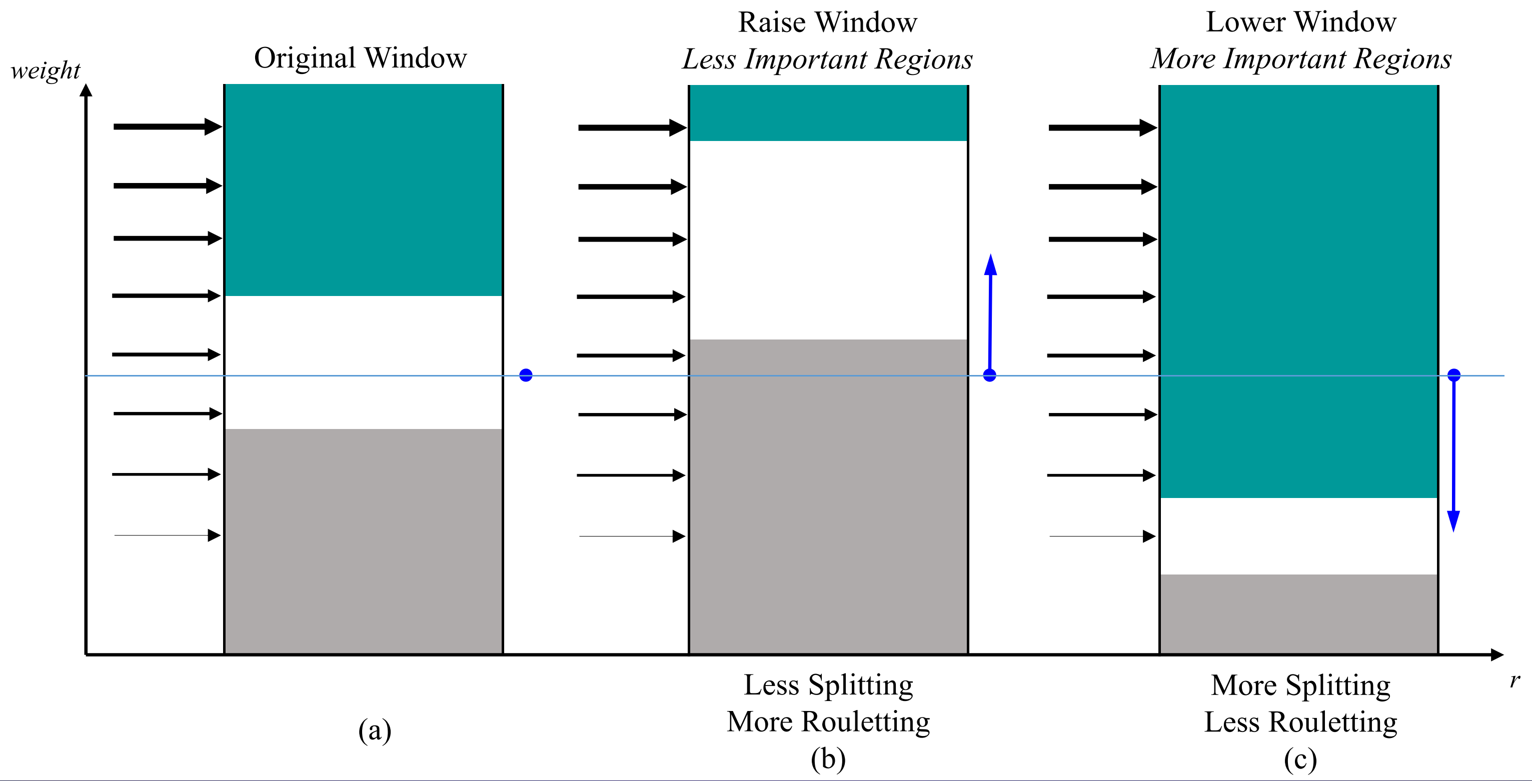
- Assign a lower and upper bound on particle weights for different regions of phase space
  - Particles with weights in the window continue without modification
  - Particles with weights above the window are split and have their weights reduced
  - Particles with weights below the window undergo Russian roulette; surviving particles have their weights increased

Weight Windows


Particle Weight In Window
Particle Weight Above Window
Particle Weight Below Window
weight
$ww_{upper}(r)$
$ww_{survival}(r)$
$ww_{lower}(r)$
r
X
Do Nothing To Particle (a)
Split Particle (b)
Roulette Particle (c)

Weight Windows

- Weight windows can increase the number of simulated particles in regions of interest while decreasing the number in unimportant regions
- Weight window bounds should be inversely related to the importance of a region to get more, low–weight particles in important regions



Automated Weight Window Generation

- Weight windows can be set manually:
  - Requires experience and trial and error to obtain good weight windows for any given problem
  - Essentially impossible to generate the optimal weight windows to get the maximum achievable figure of merit
- Automated weight window generation is preferred
- The solution of the adjoint transport equation gives the likelihood of a particle at any point in phase space contributing to a desired response (i.e., tally)
- In other words, the adjoint flux gives the importance of each region of phase space to a tally

Automated Weight Window Generation: CADIS and FW-CADIS Methods

- Perform cheap deterministic calculations (e.g., few angles, few groups) to get a rough importance map
  - Does not need to be highly accurate because the importance doesn't affect the solution, just the weight window values
- Use the importances to bias the Monte Carlo source and generate optimal weight windows for the desired tally

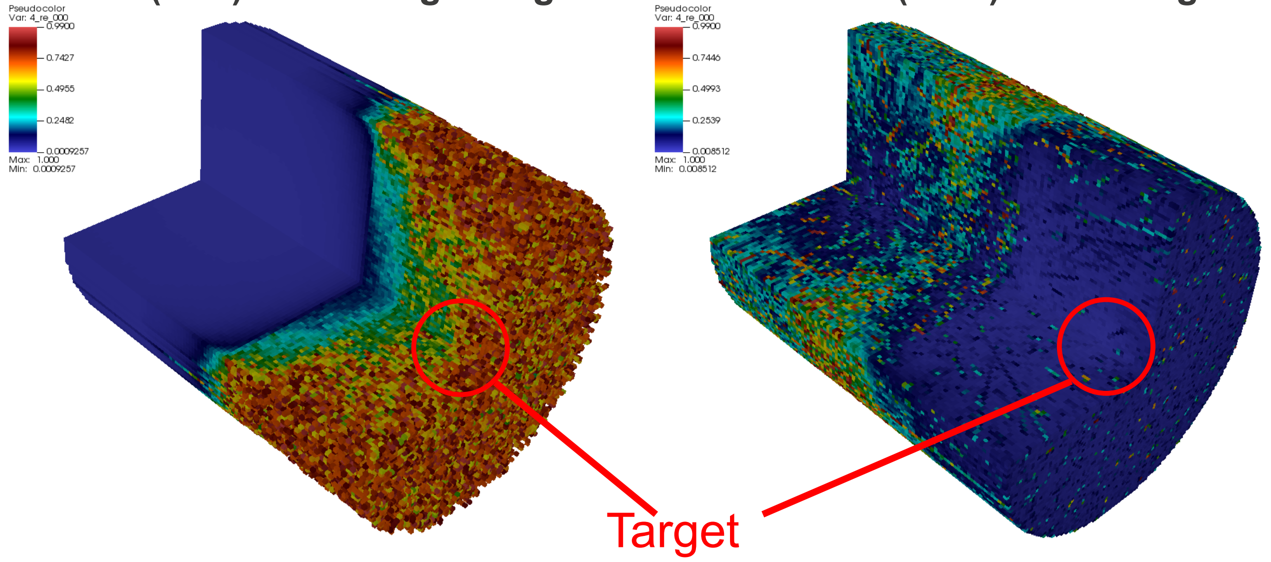
- MCATK does this by coupling to PARTISN via a tool called PAVR (PARTISN-driven Adjoint Variance Reduction)

Automated Weight Window Generation: Example

- Small source and target regions separated by polyethylene and lead shield layers.

MCNP
High Variance (Red) in the Target Region

MCNP + PARTISN
Low Variance (Blue) in the Target Region



Exercise

Exercise

- Assume a beam of monoenergetic neutrons incident on a 1.0 cm slab.
- The slab is a pure absorber with a total macroscopic cross section of 1.0 cm^{-1}.
- Compute the probability that neutrons will be transmitted through the slab.

Exercise

- For each history:
  - Calculate the distance to collision using: $d = -\frac{1}{\Sigma_t} \ln \xi$
  - If the distance to collision is larger than the width of the slab, score a leakage event:
    - Accumulate the sum and sum squared of the particle weights that leak (since all weights are 1.0, these sums are actually the same)
- After all histories are complete, compute the mean and standard deviation (σ) of the leakage tally:

$$mean = \frac{\sum_{i=1}^{N}(x_i)}{N}$$

$$variance = \sigma^2 = \frac{1}{N-1}\left(\sum_{i=1}^{N}\left(x_i^2\right)/N - \left(\frac{\sum_{i=1}^{N}(x_i)}{N}\right)^2\right)$$

Exercise

- Compute the probability of leakage and variance of the tally using 1000 histories.
- Repeat the calculation with 4000 histories. How does the standard deviation change?
- Repeat the calculation with a slab with of 2 cm. How does the mean change?

- Bonus exercises:
  - Divide the slab into 10 spatial tally bins and compute a track length tally of the flux in each bin.
  - Let the slab material include scattering: $\Sigma_t = 1.0\ cm^{-1}$, $\Sigma_s = 0.5\ cm^{-1}$, $\Sigma_a = 0.5\ cm^{-1}$, probability of scattering left or right are both 50%. Compute the probability of leakage out of both sides of the slab.

Thank you!

Contact Info:
Travis Trahan
tjtrahan@lanl.gov