MCNP Variance Reduction Techniques: What to Use When, and How

James Laird
University of Michigan
Darby S. Kimball
Bechtel National, Inc.

Introduction

- VRTs can help your code converge faster
- Staggering array of VRT options in MCNP
- Limited guidance for the new analyst
- Also limited time to learn

How VRTs Work

- Convergence is measured by statistical uncertainty (σ)
- Also by some built-in statistical tests
- Uncertainty goes down when:
  - More particles reach detector (increase ratio)
  - Contribution from each particle is about the same (reduce variance)

VRTs in MCNP: Truncation Methods

- Increase ratio of particles reaching detector
- by discarding low-contribution particles
- Energy cutoff, time cutoff
- Also: geometry cutoff

VRTs in MCNP: Population Control Methods

- Reduce variance in particle contributions
- by controlling number of tracks in interesting and uninteresting areas
- or adjusting weight of important and unimportant particles
- Geometry splitting and Russian roulette, energy splitting, time splitting, weight cutoff, weight windows

VRTs in MCNP: Modified Sampling Methods

- Increase ratio of particles reaching detector
- by altering statistical sampling to favor important particles
- Exponential transform, implicit capture, forced collisions, source biasing

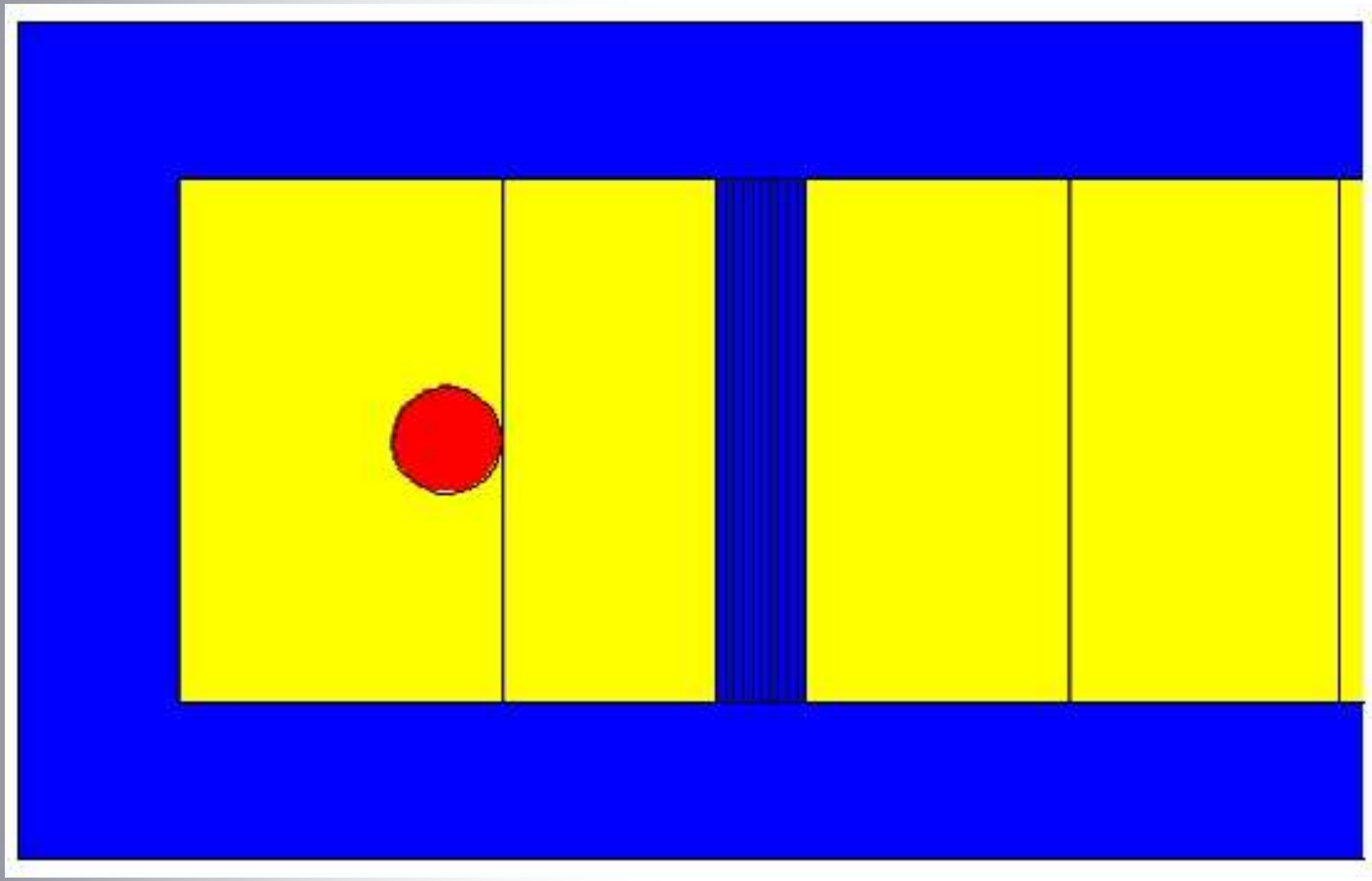
VRTs in MCNP: Partially Deterministic Methods

- Increase ratio of particles reaching detector
- AND reduce variance in particle contributions
- by circumventing the normal random walk
- Point detectors, DXTRAN spheres

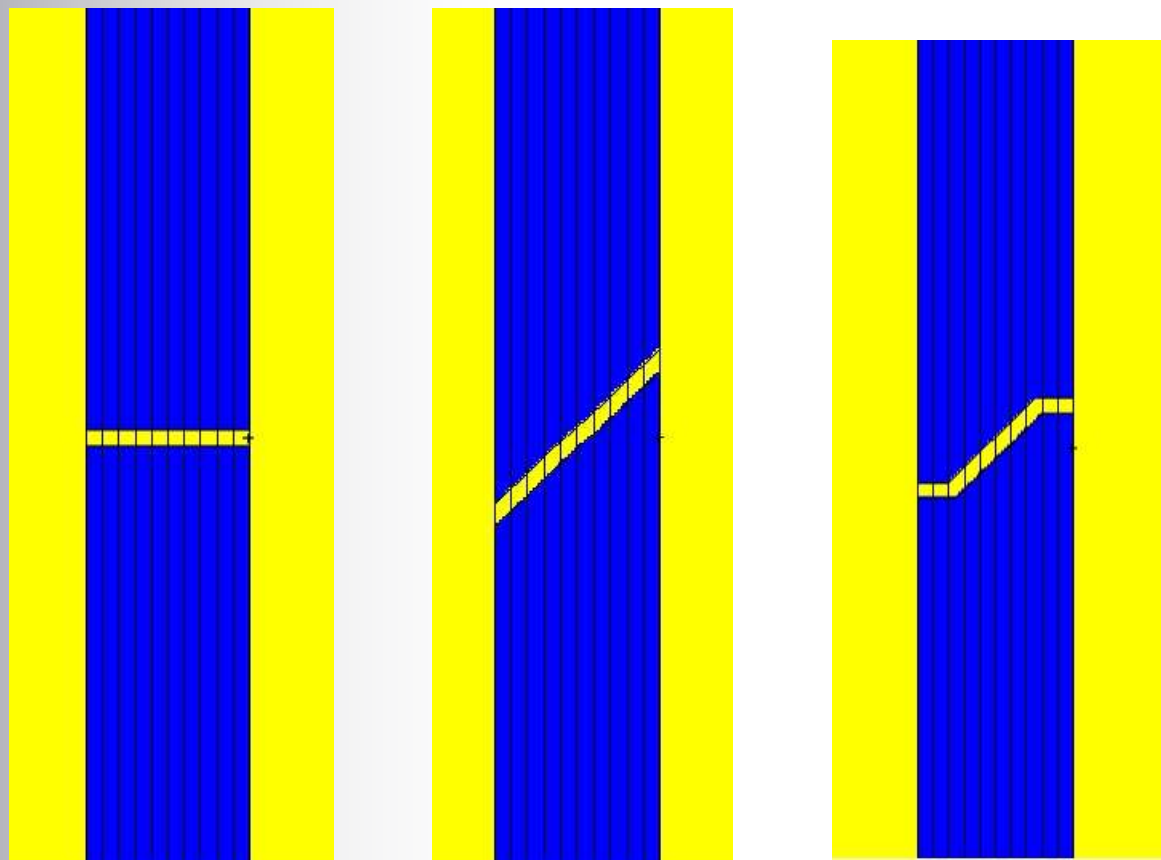
Comparison Approach

- Comparison basis:
  - Figure of Merit $$FOM \equiv 1/\left(R^2 T\right)$$
  - 10 statistical tests
- VRTs: 35 combinations of techniques and options

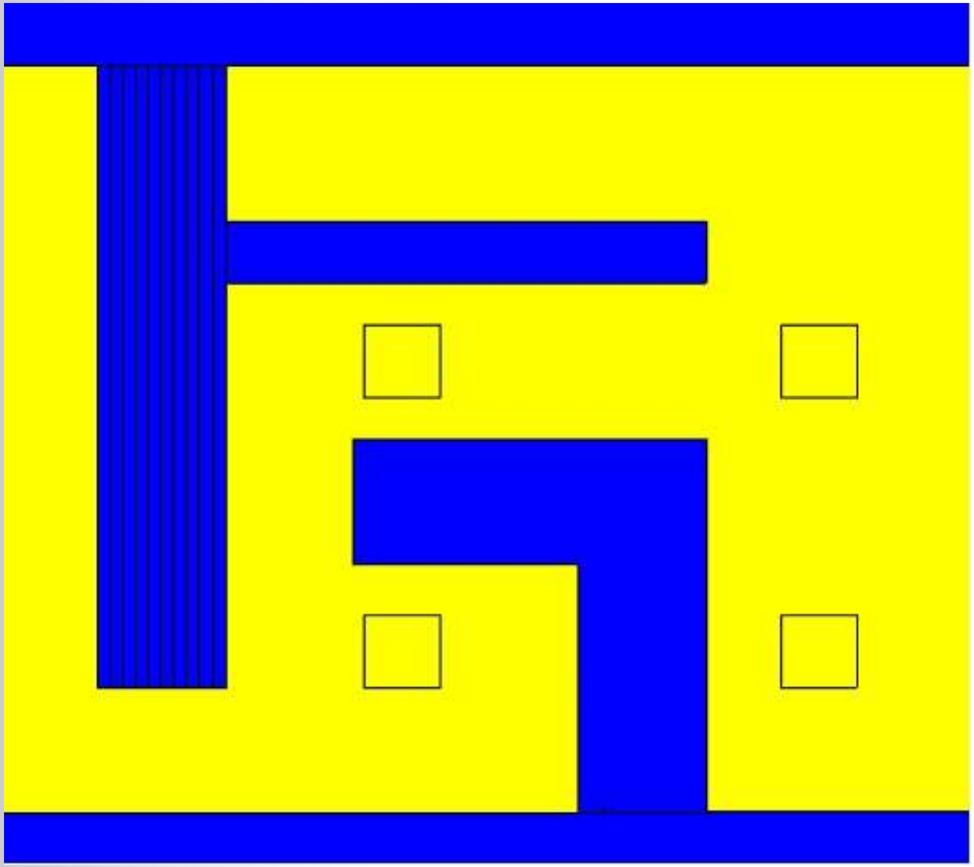
Bulk Shield Wall Model



Penetrations Models



Labyrinth Model



What to Use When Computer Time is Important

Highest FOM / most tests passed:

- Bulk shield:
  - Weight windows by energy and cell plus exponential transform (FOM 73, 10 checks passed)
  - Weight windows by energy and cell (FOM 45, 9 checks)
  - Importance splitting plus exponential transform (FOM 41, 9 checks)
  - Weight windows by cell only plus exponential transform (FOM 38, 10 checks)

What to Use
When Computer Time is Important

Highest FOM / most tests passed:

- Penetrations:
  - Weight windows by energy and cell plus DXTRAN (FOM 11-161, 10 checks passed)
  - Everything else had FOM about 1/10 of that case – generally combinations involving weight windows, DXTRAN performed well

What to Use
When Computer Time is Important

Highest FOM / most tests passed:

- Labyrinth:
  - Weight windows by energy and cell plus DXTRAN (FOM 76, 9 checks passed)
  - Weight windows by cell only plus DXTRAN (FOM 44, 9 checks passed)
  - Weight windows by energy and cell (FOM 38, 9 checks passed)

Selected VRTs by Time to Implement

Implicit capture, weight cutoff, time cutoff, energy cutoff	2 minutes
Source biasing by direction	15 minutes
Forced collisions	25 minutes
Importance splitting	45-60 minutes
Weight windows	45-180 minutes
DXTRAN	90 minutes
Exponential transform	90-210 minutes

What to Use
When Human Time is Important

- Source biasing by direction or energy
- Simple importance splitting
- Point detectors

Selected VRTs by Time to Learn

Anything with cut card	5 minutes
Importance splitting	15 minutes
Source biasing	30 minutes
Forced collisions	30 minutes
Energy splitting	60 minutes
Point detectors	2 hours
Weight windows	1 day
Weight windows by energy	1 day
Exponential transform	2 days
DXTRAN spheres	4 days

What to Learn First

- Importance splitting
- Weight windows
- Exponential transform
- Point detectors / DXTRAN

Sanity Check

- Always check: did it help or hurt?
- Watch out for improvements that aren't
- Think like a malicious particle
- Are you accidentally ignoring particles?
- Always balance the time you are spending with the efficiency gain

Tips & Tricks

- Single biggest help is properly creating problem
- Both importance splitting and weight windows recommended for nearly every problem – default settings are good, and one or two iterations generally enough
- Weight windows are faster to implement when importance splitting also used

Tips & Tricks

- Leave default VRTs on, generally (weight cutoff and implicit capture)
- However, do not use weight cutoff with forced collisions
- Definitely use weight cutoff with DXTRAN spheres to reduce cross-talk

Conclusions

- Learn how to use weight windows
- For bulk shield, use weight windows, then add exponential transform for hard problems
- For penetrations, use DXTRAN spheres, then add weight windows and/or forced collisions
- For labyrinths, use weight windows, then add point detectors or DXTRAN spheres

Questions?