Development of variance reduction capability in Serpent 2

6th International Serpent UGM, Milan, Italy, Sept. 26-29, 2016
Jaakko Leppänen
VTT Technical Research Center of Finland

Outline

Overview of variance reduction techniques:

- Analog and implicit Monte Carlo simulation
- Weight windows
- Adjoint problem and importance

Methodology implemented in Serpent:

- MCNP format weight window mesh
- Built-in response matrix based solver

Brief introduction to response matrix method

Demo

Overview of variance reduction techniques

Variance reduction techniques in Monte Carlo simulation can be understood by considering the following concepts:

Physical reality – Infinite number of possible particle histories

Analog simulation – Randomly selected sample from the infinite number of possible particle histories

Implicit simulation – Sample from the infinite number of possible particle histories, selected in such way that the histories contribute to a specific result

In implicit simulation, each particle is assigned with a statistical weight, which is used as a multiplier for all tallies

Variance reduction techniques work by "cheating" in the analog game, and correcting the results by adjusting the particle weight

The idea is that the transport of particles may be manipulated, but the simulation remains unbiased as long as the transport of the statistical weight is preserved

Overview of variance reduction techniques

Favoring histories that contribute to a specific result is done by assigning importances to various events (collisions, boundary crossings, etc)

Particles can be encouraged to migrate towards higher importance by using a weight window mesh:[1]

- The geometry is covered by a mesh (Cartesian, cylindrical, unstructured, etc.)
- Each mesh cell is assigned with a minimum and maximum weight: W_min, W_max
- The bounding weights are inversely proportional to importance

When the particle enters the mesh cell, its weight w is compared to the boundaries:

- If $w < W_\mathrm{min}$ – Russian roulette: particle is either killed or its weight is increased
- If $w > W_\mathrm{max}$ – Splitting: the history is divided into multiple parts

After the operation all particles inside the mesh cell have weights between W_min and W_max

[1] The mesh can also include energy dimension.

Overview of variance reduction techniques

Forward and adjoint transport problems:

Forward problem – Calculate responses (physical reaction rates) induced by particles originating from a given source

Adjoint problem – Calculate contributions of various events (reactions, boundary crossings, etc.) to a given response

Solution of the adjoint problem provides the importances needed for forming the weight window mesh for variance reduction

Physical interpretation of importance: The average contribution of a particle at position (x, y, z), traveling in direction (u, v, w) with energy E to given response f

Overview of variance reduction techniques

Solving the adjoint problem using the Monte Carlo method essentially implies running the transport simulation backwards (or back-tracking stored events)

For variance reduction purposes the solution typically relies on deterministic transport methods

Serpent provides two options:

1) Read MCNP format weight window mesh generated using some deterministic tool

2) Apply a built-in response-matrix based solver

The use of SCALE/MAVRIC for producing weight windows for Serpent was tested by a summer student from Aalto University (the report is not completed yet)

This presentation is focused on the built-in solver

Built-in response-matrix based solver

Response matrix method in a nutshell:

- The geometry is constructed from cells that are connected to their neighbors via coupling coefficients
- The coefficients form a matrix that defines which fraction ($\alpha_{i \to j} < 1$) of incoming particle current[2] from neighbor i passes into neighbor j
- Source term for each cell is formed by a vector that defines the fraction of source particles passing into neighbor j
- Response term for each cell is formed by a vector that defines the contribution of incoming current from neighbor i into the response
- Forward problem is solved by starting from the source term and recursively passing the currents into cell neighbors until the contribution becomes negligible
- Adjoint problem is solved by starting from the responses and iterating backwards

[2] The partial currents can be divided into energy groups.

Built-in response-matrix based solver

Solver in Serpent:

- The geometry is covered by a Cartesian or cyclindrical mesh
- Responses are linked to detectors
- Forward transport simulation provides the coupling coefficients, source and response vectors for each mesh cell
- The response matrix solver is run after the simulation is completed, the resulting importances are written in a file that can be used in a second calculation for forming the weight window mesh

The methodology is best demonstrated via examples!

Limitations and topics for future work

The basic weight window methodology still requires work:

- Support for cylindrical and multi-level mesh (MCNP format) and other mesh formats
- More efficient routines for source biasing
- Testing in complex geometries (fusion?)

Development of the response matrix solver continues:

- Calculation of importances needs to be verified
- Implementation of energy groups
- Extending from regular Cartesian and cylindrical to irregular meshes (adaptive Cartesian N-tree or even unstructured polyhedral)

The built-in methodology is most likely best suited as a light-weight solver for relatively simple problems, challenging variance reduction task require more elaborate methods