Interim Report

Arnika Chidambaram   Caroline Hughes   Mitch Negus

November 15, 2016

1 Introduction

(Arnika)

The purpose of this project is to study variance reduction methods in Monte Carlo using deterministic methods. Deterministic methods provide fast solutions to the transport equation, and require relatively simple inputs. Currently there is high demand for a reliable and equally efficient transport method that can be used to solve complex fixed source problems [1]. The two variance reduction methods that will be explicitly explored for the the purposes of this project are Consistent Adjoint Driven Importance Sampling (CADIS-Ω) and Forward Weighted-Consistent Adjoint Driven Importance Sampling (FW-CADIS). CADIS-Ω optimizes specific, local response function for flux, where as FW-CADIS optimizes global response function for flux everywhere in the phase space and accelerates global flux solution calculations such that global fixed source calculations can be determined using Monte Carlo.

The goal is to create an importance map for the Monte Carlo game and selecting parameters is an important step in deterministic calculations. Ideally, accurate parameters are selected, which will result in a map that accelerates the Monte Carlo Calculation while simultaneously lowering deterministic cost [1].

2 Mathematics

2.1 Variance Reduction

(Mitch)

Monte Carlo simulations have the unique advantage of being continuous and entirely free from the effects of discretization choices that are present in deterministic simulations, however they tend to be much more computationally expensive. In analog Monte Carlo simulations, enormous numbers of particle histories must often be tracked to produce ample statistics for meaningful results.

To meet this criteria and be considered meaningful, data must sufficiently cover the solution space and have a low statistical uncertainty, represented by a small variance. The variance (the square of the standard deviation, σ) is defined as the squared sum of the deviation of each datapoint from the mean, $\bar{x}$, divided by the number of degrees of freedom in the problem (one less than the totlal number of datapoints) [2]:

$$\sigma^2 = \frac{\sum_i^N (x_i - \bar{x})^2}{N-1},$$

where N is the number of particle histories included in the calculation. Since the particle histories (and therefore x_i) are independent, the central limit theorem applies. It follows that the standard deviation of the mean, $S_{\bar{x}}$, decreases proportionally to $1/\sqrt{N}$. For comparative purposes, this measure is usually represented by the relative error, R, where

$$R \equiv \frac{S_{\bar{x}}}{\bar{x}} = \frac{\sigma^2}{\bar{x}\sqrt{N}}.$$

More particles lead directly to a reduced uncertainty.

While simulating more particles is beneficial for reducing uncertainty in a simulation, it is unsurprisingly detrimental to simulation efficiency. In general, the efficiency of a simulation is given by its figure of merit (FOM), defined as

$$\text{FOM} \equiv \frac{1}{R^2 T}.$$

R is the relative error and T is the computer time taken for the simulation [3]. Assuming independent particles, the run time T is directly proportional to the number of particles: $T \propto N$. When the dependencies of both R and T on N are considered, the FOM becomes constant and no substantial gains are made in terms of performance for more particles ($R^2 T \propto (1/\sqrt{N})^2 N = const.$).

Specific variance reduction methods attempt to circumvent this dilemma using a variety of clever techniques. Often, these methods emphasize simulations in regions of the problem space that have the strongest contributions to the final solution, gleaning deeper insights without wasting as much computation as pure analog Monte Carlo simulations. Example methods include truncation methods, population control methods, modified sampling methods, and partially-deterministic methods. For maximal effect, several of these methods may be employed together in a given simulation.

Truncation methods are undoubtedly the most common variance reduction technique, whereby the problem space is restricted in some dimension (technically, truncation methods are employed in almost every Monte Carlo simulation to some degree, as finite boundaries must exist on space, particle energy, time, etc.). Truncation methods speed up calculations by avoiding simulation of parts of a problem that will not contribute to the solution. While easy to incorporate into a simulation, care must be taken to ensure that these parts of the problem space actually *will not*, as opposed to just *should not*, contribute to the solution.

Population control methods allow more flexibility in sampling areas of importance in the phase space, and can be separated into two categories: splitting and rouletting. In both cases, particles are assigned weights which are adjusted depending on that particle's importance to the final solution. Splitting takes place when a particle enters an area of higher importance, since it is most important that the variance be low in this region. By dividing the particle into multiple copies, the total number of particle histories in the simulation, N, can be artificially inflated, and the relative error reduced. In order to preserve the initial particle's weight w on the final solution, the new n split copies are scaled appropriately by a factor of w/n.

$$\sum_i^n \frac{w}{n} = n\left(\frac{w}{n}\right) = w$$

Conversely, if a particle travels into a region of lower importance, the particle will undergo a "game" of Russian roulette, with a survival probability of $1/n$. In this way, we reduce the number of particles that must be tracked, thereby reducing the time T of the simulation. The relative error of the desired solution will experience less impact since these rouletted particles are in areas of low importance and less likely to contribute to that solution. Any particle that survives will have it's initial weight, w, increased by a factor of n, so that on average the total weights are conserved. In addition to the splitting and rouletting routines, weight window restrictions are often placed on the simulation so that weights do not become too large or too small. Particles with large weights will be split to prevent any one particle from having too much effect on the final solution, while particles with small weights will be rouletted to prevent the simulation wasting time tracking essentially meaningless histories.

Population control methods influence the number of particles based on where they travel, weighting them accordingly. Similarly, modified sampling methods use adjusted probability distributions to control the number of particles generated with a given set of initial conditions and weight those particles accordingly. If a given situation ought to occur with probability P, a modified sample would generate that same situation with probability P', but with weight $w_{\text{new}} = P/P'$. The overall probability is preserved:

$$P = w_{new} \cdot P' = \frac{P}{P'}P' \tag{1}$$

but the number of particles is adjusted to more effectively contribute to the solution and reduce the variance.

Finally, partially deterministic methods are perhaps the most difficult methods to use, and require preliminary deterministic calculations to find the simulation regions most likely to contribute to the final solution. While these prior understandings of the system can be developed using intuition, experience, and theory, procedures have been developed to perform these calculations automatically. Algorithms such as CADIS and FW-CADIS accomplish this task, establishing foundations for the implementation of further variance reduction techniques.

2.2 FW-CADIS

(Caroline)

The Forward-Weighted Consistent Adjoint Driven Importance Sampling (FW-CADIS) method employs the discrete ordinates S_N adjoint function for “automatic variance reduction of Monte Carlo calculations [4]. In general, the adjoint $A^\dagger$ of an operator, function, or matrix A is the complex conjugate of the transpose of A [5]. It has the property:

$$\langle \psi^\dagger A \psi \rangle = \langle \psi A^\dagger \psi^\dagger \rangle \tag{2}$$

where $\langle \cdot \rangle$ represents “integration over all independent variables,” which, in phase space, is given by [6]:

$$\langle \cdot \rangle = \int d^3\boldsymbol{r} \int dE \int d\hat{\boldsymbol{\Omega}}$$

The adjoint of the Boltzmann Neutron Transport equation (TE) represents the importance of the contribution of a particle at a particular location, time, energy, and angle, to an objective function such as detector source [4]. For the transport operator H defined in Eq. (3a), the adjoint transport operator $H^\dagger$ can be derived using the property given in Eq. (2) and is defined in Eq. (3b).

$$H = \hat{\boldsymbol{\Omega}} \cdot \nabla + \Sigma_t(\boldsymbol{r}, E) - \int_0^\infty dE' \int_{4\pi} d\hat{\boldsymbol{\Omega}}' \Sigma_s(\boldsymbol{r}, E' \to E, \hat{\boldsymbol{\Omega}}' \cdot \hat{\boldsymbol{\Omega}}) \tag{3a}$$

$$H^\dagger = -\hat{\boldsymbol{\Omega}} \cdot \nabla + \Sigma_t(\boldsymbol{r}, E) - \int_0^\infty dE' \int_{4\pi} d\hat{\boldsymbol{\Omega}}' \Sigma_s(\boldsymbol{r}, E \to E', \hat{\boldsymbol{\Omega}} \cdot \hat{\boldsymbol{\Omega}}') \tag{3b}$$

FW-CADIS is “particularly useful” for problems with a fixed source q [1]. For a detector response at $(\boldsymbol{r}_0, E, \hat{\boldsymbol{\Omega}}_0)$ the TE becomes [7],

$$\begin{aligned} H\psi &= q = \delta(\boldsymbol{r} - \boldsymbol{r}_0)\delta(E - E_0)\delta(\hat{\boldsymbol{\Omega}} - \hat{\boldsymbol{\Omega}}_0) \\ H^\dagger\psi^\dagger &= q^\dagger \end{aligned}$$

Detector response R is given by:

$$\begin{aligned} R &= \int d^3\boldsymbol{r} \int dE \int d\hat{\boldsymbol{\Omega}}\psi(\boldsymbol{r}, E, \hat{\boldsymbol{\Omega}})q^\dagger = \int d^3\boldsymbol{r} \int dE\phi(\boldsymbol{r}, E)q^\dagger \\ &= \int d^3\boldsymbol{r} \int dE \int d\hat{\boldsymbol{\Omega}}\psi^\dagger(\boldsymbol{r}, E, \hat{\boldsymbol{\Omega}})\rho(\boldsymbol{r}, E) = \int d^3\boldsymbol{r} \int dE\phi^\dagger(\boldsymbol{r}, E)\rho(\boldsymbol{r}, E) \end{aligned}$$

where $q^\dagger$ is the adjoint source, which will be used as the objective function, and ρ is a probability distribution function.

The goal of FW-CADIS is to construct reduce variance parameters for Monte Carlo simulations, as described in Section 2.1. This requires finding window weights ww, birth weight w_0, and a biased pdf $\hat{\rho}$. These are given in Eqs. (4) to (6).

The FW-CADIS method relies on both source and transport biasing, two of three significant categories of biasing schemes, with the third being collision biasing. The FW-CADIS method requires two major steps – explained in graphic detail in Fig. 1 – to implement both biasing schemes. First, the source biasing parameters are calculated using a space- and energy-dependent adjoint function $\phi^\dagger$. The biased (also refered to as “nonanalog”) probability distribution is given in Eq. (4) [4].

$$\hat{\rho}(\boldsymbol{r}, E) = \frac{\rho(\boldsymbol{r}, E)\phi(\boldsymbol{r}, E)}{R} \tag{4}$$

This represents the relative contribution of a particle to the total detector response.

Transport biasing is used to generate the weight window, which is used for Monte Carlo space- and energy-dependent applications of splitting and rouletting. The weight window ww is given by Eq. (5) and the weight window lower bounds w_0 are given by Eq. (6) [4, 8].

$$ww = \frac{\phi^\dagger(\boldsymbol{r}, E)}{R} \tag{5}$$

$$w_0 = \frac{\rho(\boldsymbol{r}, E)}{\hat{\rho}(\boldsymbol{r}, E)} \tag{6}$$

This is consistent with Eq. (1), which describes how probability is preserved even as weights are adjusted.

It is important to note that FW-CADIS works for problems dependent on space and energy *only.* Wagner and Haghighat describe that angular-dependent problems would require significantly more memory and be significantly less accurate as a result of the limitations of using discrete ordinates in three dimensions [4]. Developing a method that will consider angular dependence as well is the focus of work on CADIS-Ω. The algorithms described in Section 3 have been written in support of work by Rachel Slaybaugh, Madicken Munk, et. al. on the CADIS-Ω method, intending to serve as a point of comparison between the novel CADIS-Ω method and popular FW-CADIS method.

3 Algorithms

(Caroline)

The FW-CADIS method involves two major steps: a forward deterministic calculation to find the adjoint source, $q^\dagger$, and an adjoint deterministic calculation to create an importance map for Monte Carlo calculations [1]. This map requires the weight window ww, particle birth weight, and biased source probability, $\hat{\mathbf{rho}}$. This can be accomplished with the AutomateD VAriaNce reducTion Generator (ADVANTG), which performs the deterministic calculations using Denovo, a "3-D, block parallel discrete ordinates transport code" from ORNL [9]. The complete process is described graphically in Fig. 1 [9].

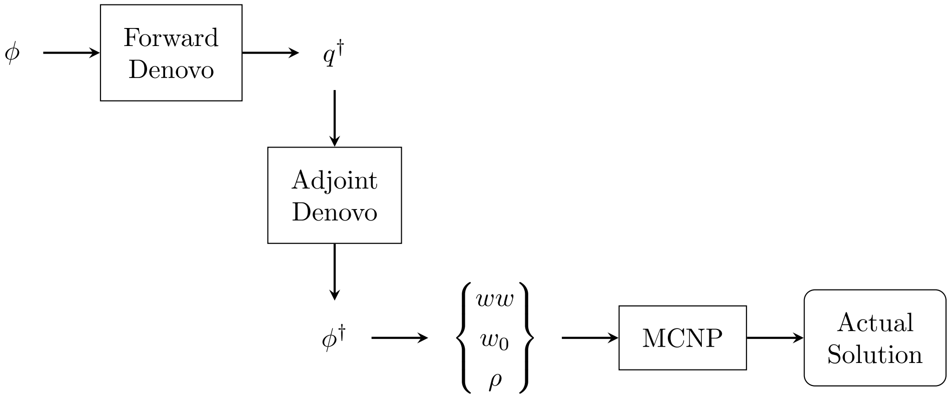

Figure 1: Flow chart of process required to use the FW-CADIS method for variance reduction

The algorithm described includes steps required to automate each step of the process described in Fig. 1.

3.1 ADVANTG Input

(Mitch)

Procedurally, in order to generate solutions with optimal efficiency yet no more than 5% uncertainty, we intend to test a variety of inputs using ADVANTG [9]. The parameter assignments to be tested are listed in the table below. The assignments were picked to give a spread of each parameter within the ADVANTG-imposed limits on that parameter.

From Table 1, it can be shown that the study will span a total of 30 distinct tests (2 triangular quadrature sets × 3 quadrature orders × 3 Legendre Scattering Expansion orders + 1 product quarature set × 4 quadrature angle numbers × 3 Legendre Scattering Expansion orders).

Test cases will each be generated in the form of an ADVANTG input file and cumulatively processed using the software. The process of generating ADVANTG files for varied parameters will be automated.

Table 1: List of assignments to be tested for each parameter. Optimization will seek the combination of parameter assignments which gives the lowest uncertainties as well as maximal efficiency.

Parameter	Parameter Assignment	Notes
Quadrature Type	Quadruple Range (tri.), Level Symmetric (tri.), Gauss Legendre (prod.)	
Quadrature Order *Triangular only*	$10, 15, 20$	default 10; max. 32 for Q.R. and max 24. for L.S.
Quadrature Angle Number *Product only* Ordered Pair: (Azimuthal, Polar)	$(2, 2), (4, 4), (10, 10), (16, 16)$	default $(4, 4)$; max. $(37, 16)$
Legendre Scatter Expansion Order	$2, 3, 5$	default 2

3.2 ADVANTG Output and MCNP Input

(Caroline, Mitch)

ADVANTG's main output is a weight window input (WWINP) MCNP file based on the input MCNP inp file. The WWINP file appends two types of cards to the original inp file: replacement biased source (`SB`) cards, which contain importance-weighted biased probabilities $\mathbf{r\hat{h}o}$, and weight window parameter (`WWP`) cards [9]. Inspecting the Silo file and visualizing the data with VisIt will be automated.

ADVANTG also outputs Silo files, which can be inspected using the VisIt visualization toolkit [10] for quality assurance. Upon passing inspection, the corresponding MCNP file will run to calculate the dose measured for the given problem. The process of running MCNP input files in the MCNP command window will be automated. When optimizing the computational time and result accuracy for various input parameters, it will be important to know the Denovo run times so that these can be contributed to the total run time of the process and studied individually. This process will also be automated.

3.3 Read MCNP Output

(Caroline)

Once MCNP runs the WWINP file created by ADVANTG, it will create an output file containing all of the information it collected about the test problem and while running the test problem. The user is able to select whether they want MCNP to return the solution to first- or second-order accuracy [11]. Reading these error terms will be beneficial in determining MCNP's confidence in the final result.

One of the major benefits in using a deterministic method to compute the variance reduction is to reduce computational time, since deterministic methods are generally faster than Monte Carlo methods. Monitoring the run time of both MCNP and Denovo will reveal when the deterministic method is improving run time and by how much.

4 Code Uses

4.1 Test Cases

(Arnika)

As specified by the work of Madicken Munk, there are nine test cases total that will be run using ADVANTG:

1. A five legged labyrinth running across a Lithium-doped polyethylene shield[12].
2. A three legged labyrinth running across a 100 centimeter-thick concrete shield[12].
3. A 100 centimeter-thick block of polyethylene with a steel bar going through it, acting as a streaming channel[12].
4. A labyrinth with a source on its left and a detector on it's right[12].
5. A polyethylene shielding wall thick shielding with a source on its left and a detector on its right and steel support structures running through the shield[12].
6. A beam source of neutrons directed towards a section of NaI[12].
7. A reactor containment building model designed off of an Advanced Boiling Water Reactor (ABWR)[12].
8. A fission source and a block of water sectioned off by a steel block[12].
9. A therapy room based off of a radiation therapy vault[12].

5 Plans for Completion

(Caroline, Mitch)

After careful study of the mathematical background required for our project, the remainder of the project can be divided into two major components: automation and optimization. The updated timeline divided by subject area with consideration of time required to write the final report and presentation is as follows:

Automation

Tu 11/22	Read and store Denovo run times Read and store MCNP run times and results
F 11/25	Write ADVANTG files. Inspect Silo file and visualize using VisIt Run MCNP input file in MCNP command window
Su 11/27	Complete automation.

Characterization Tests

F 12/02 Finish running Maze1, Maze2 and Prob-1 test cases:

Su 12/04 Finish running Prob-2, Prob-4 and Beam test cases:

Tu 12/06 Finish running Reactor, WS and Therapy Room test cases:

Th 12/08 For each test case, study uncertainty in results and Denovo and MCNP run times for the input parameters listed in Table 1. Plot relationships to analyze.

Reports and Presentations

Sa 12/10 Complete preliminary drafts of the report and presentation.

M 12/12 Complete final drafts of the report and presentation

W 12/14 Submit final report and present

References

[1] R. N. Slaybaugh and S. C. Wilson. "Deterministic Parameter Study for Fixed-Source Calculations Using FW-CADIS". *Proceedings of the 2013 ANS Annual Meeting*. Vol. 108. Atlanta, GA, June 2013. DOI: `10.13140/2.1.2255.1688`.

[2] I. G. Hughes and T. P. A Hase. *Measurements and their Uncertainties*. Oxford University Press, 2010. ISBN: 978-0-19-956633-4.

[3] T. E. Booth et al. "MCNP A General Monte Carlo N-Particle Transport Code, Version 5". *LA-UR-03-1987* (2008).

[4] J. C. Wagner and A. Haghighat. "Automated Variance Reduction of Monte Carlo Shielding Calculations Using the Discrete Ordinates Adjoint Function". *Nucl Sci Eng* 128.2 (Feb. 1998), pp. 186–208. DOI: `10.13182/NSE98-2`.

[5] M. L. Boas. *Mathematical Methods in the Physical Sciences*. 3rd. John Wiley & Sons, 2003. ISBN: 978-0-471-19826-8.

[6] E. E. Lewis and W. F. Miller. *Computational Methods of Neutron Transport*. John Wiley & Sons, 1984. ISBN: 0-471-09245-2.

[7] R. N. Slaybaugh. "Lecture Notes from NE 250: Nuclear Reactor Theory on Oct 19". University of California, Berkeley, 2015.

[8] R. N. Slaybaugh. "Lecture Notes from NE 250: Nuclear Reactor Theory on Oct 28". University of California, Berkeley, 2015.

[9] S. Mosher and etal. "ADVANTG–An Automated Variance Reduction Parameter Generator" (2013).

[10] Hank Childs et al. "VisIt: An End-User Tool For Visualizing and Analyzing Very Large Data". *High Performance Visualization–Enabling Extreme-Scale Scientific Insight*. Oct. 2012, pp. 357–372.

[11] T. Goorley. "MCNP6.1.1-Beta Release Notes". *LA-UR-14-24680* (2014).

[12] M. Munk. *caskmodels*. 2016. URL: `https://github.com/munkm/caskmodels/tree/master/characterization_tests`.