LA-UR- 09-02594

Approved for public release;
distribution is unlimited.

Title:	Adjoint-Weighting for Critical Systems with Continuous Energy Monte Carlo
Author(s):	Brian Kiedrowski & Forrest Brown
Intended for:	American Nuclear Society 2009 Nuclear Criticality Safety Topical Meeting, Richland, WA, 13-17 September 2009

Form 836 (7/06)

ADJOINT-WEIGHTING FOR CRITICAL SYSTEMS WITH CONTINUOUS ENERGY MONTE CARLO

Brian C. Kiedrowski[a], Forrest B. Brown[b]
[a] *University of Wisconsin-Madison (Dept. of Engineering Physics)*
[b] *Los Alamos National Laboratory (X-3 MCC)*
bckiedro@lanl.gov, fbrown@lanl.gov

Traditionally, Monte Carlo radiation transport codes have been unable to tally adjoint-weighted quantities in a robust and consistent way. A new framework based on the iterated fission probability interpretation of the adjoint flux is developed that solves this problem at minimal increase in CPU time. A derivation of adjoint-weighted tallies is presented along with a methodology for computing adjoint-weighted reactor kinetics parameters and first-order reactivity perturbations. Details are given for the implementation in the production MCNP code. The calculations are benchmarked to both experimental measurements and calculations with the discrete ordinates code, Partisn.

I. INTRODUCTION

Continuous energy Monte Carlo radiation transport has been unable to calculate many quantities used in reactor physics. Unfortunately, many of these quantities are important for criticality safety applications such as analyzing the effects of perturbations, kinetics analysis, and sensitivity studies. These take the form of the inner-product of some function $f(\mathbf{r}, \hat{\mathbf{\Omega}}, E)$ weighted by the adjoint flux $\psi^\dagger(\mathbf{r}, \hat{\mathbf{\Omega}}, E)$. Two examples are the neutron generation time Λ and the effective delayed neutron fraction β_{eff} in the classic point reactor kinetics[1] equation:

$$\frac{dn}{dt} = \frac{\rho - \beta}{\Lambda} n(t) + \sum_i \lambda_i c_i(t). \tag{1}$$

The neutron generation time Λ and the effective delayed neutron fraction β are given by

$$\Lambda = \frac{\left\langle \psi^\dagger \frac{1}{v} \psi \right\rangle}{\left\langle \psi^\dagger \mathbf{F} \psi \right\rangle}, \tag{2}$$

$$\beta = \frac{\left\langle \psi^\dagger \mathbf{B} \psi \right\rangle}{\left\langle \psi^\dagger \mathbf{F} \psi \right\rangle}. \tag{3}$$

Here ψ is the angular flux, $\psi^\dagger$ is the adjoint flux, v is the neutron speed, and $\mathbf{F}$ and $\mathbf{B}$ are the total and delayed fission operators respectively.

Another very common application to reactor safety is linear perturbation theory. Often an engineer would like to know the effect of a small change to a reactor configuration. This effect is measured as the change in the reactivity $\Delta\rho$. From linear perturbation theory, the change in reactivity is estimated by

$$\Delta\rho = -\frac{\left\langle \psi^\dagger \mathbf{P} \psi \right\rangle}{\left\langle \psi^\dagger \mathbf{F} \psi \right\rangle}. \tag{4}$$

$\mathbf{P}$ represents the operator for perturbation, which takes the form $\mathbf{P} = \Delta\Sigma_t - \Delta\mathbf{S} - \frac{1}{k}\Delta\mathbf{F}$. Each term multiplied by the flux represents (from left to right): the change in the total reaction rate, the change in the scattering source, and the change in the fission source divided by the system multiplicity k.

Each of the inner products in the equations above represents an integral quantity (called a tally) for Monte Carlo to compute during the random walk process. By taking ratios of these tallies, these parameters can be estimated.

The concept of these adjoint-weighted tallies will be developed with a theoretical basis. Explicit formulations for the tallies will be given along with details of the implementation in MCNP5.[2] The results will be validated to both a multigroup discrete ordinates code, Partisn,[3] and experimental data.

II. THEORETICAL DEVELOPMENT

The iterated fission probability[4] interpretation of the adjoint flux in a critical system at a point in phase space $(\mathbf{r}, \hat{\mathbf{\Omega}}, E)$ is the expected steady state population caused by a neutron introduced into the system at that point.[5, 6] For cases that are subcritical or supercritical, the adjoint flux is similar except that it is the expected total neutron population in the core caused by that source neutron in some fission generation after the asymptotic period is established. Since all of the quantities of interest consist of ratios and most iterative schemes for computing k-eigenvalue problems renormalize the population every generation, the subcritical, critical, and supercritical cases will yield consistent results.

II.A. Derivation of the Adjoint-Weighted Source

Suppose a reactor at critical and the phase space is subdivided into J arbitrary regions with index j and volume V_j. An arbitrary discrete source conforming to each region is introduced into the reactor such that the source intensity in region j is s_j. The source is normalized such that the sum of s_j is unity and can be considered a probability density function. The cumulative density function is therefore

$$F_j = \sum_{i=1}^{j} s_i.$$

$F_0 = 0$. Suppose that N random trials are made and a contribution function for trial n is defined such that

$$t_j^n = \begin{cases} \pi_j & F_{j-1} \leq \xi_n < F_j \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

ξ_n represents the nth uniform random variable sampled on domain $[0, 1]$. π_j is a sample from an arbitrary probability distribution representing the asymptotic population of a source neutron emitted in region j. This can be obtained by following a neutron for a large number of generations and counting the number of progeny in that distant generation. The form of the distribution π_j is unknown, but it has a mean of $C\psi_j^\dagger$ through the iterated fission probability interpretation of the adjoint flux.

Now define the tally in region j as

$$T_j = \mathbb{E}\left(t_j\right). \tag{6}$$

Here $\mathbb{E}$ denotes the expected value or mean. Now the distribution t_j must be analyzed. First consider the Bernoulli distribution:

$$B(p) = \begin{cases} 1 & \text{w.p. } p \\ 0 & \text{otherwise} \end{cases}$$

This distribution (Bernoulli with parameter p) is one with probability p and zero otherwise. The expected value of $B(p)$ is p.

The distribution of t_j is a product of a Bernoulli distribution with parameter s_j and distribution p_j. To see this, first sample the distribution π_j and then multiply it by the result of $B(s_j)$ which is zero or one. The result of this is exactly equivalent to one from t_j given in (5). Further, s_j and π_j are sampled independently: whether or not a neutron is emitted by the arbitrary external source does not depend on its asymptotic population. Through some manipulation this tally yields the adjoint-weighted source:

$$T_j = \mathbb{E}\left(t_j\right) = \mathbb{E}\left(\pi_j B(s_j)\right) = \mathbb{E}\left(\pi_j\right)\mathbb{E}\left(B(s_j)\right) = C\psi_j^\dagger s_j. \tag{7}$$

II.B. Adjoint-Weighted Flux Tally

Extending this to encompass the flux follows. The random walk sequence of a direct radiation transport simulation used in Monte Carlo provides a very straightforward way to sample the flux distribution. The flux tally in region j (for the analog case) is

$$\psi_j = \frac{1}{V_j}\mathbb{E}(d_j) = \frac{1}{V_j}\sum_{n=1}^{N} d_j^n.$$

Here d_j is a distribution that describes the total track length accumulated within region j for a single history. The total track length in region j for history n, d_j^n, is the sum of trajectories τ each with length ℓ_τ. Every history in the Monte Carlo simulation, this distribution is sampled and the average (divided by the region volume) represents the average flux in that region.

Suppose that region j is broken into numerous differential elements of phase space. If a trajectory interacts with the differential element, it has a differential contribution $d\ell$. Summing (integrating) all these contributions over region j and dividing by the volume and the number of histories will yield an estimate of the average flux integrated over the region. From the law of large numbers, this tally will converge to the true average flux in region j given a sufficient number of trials.

The adjoint-weighted flux tally is very similar. This time the trajectory will contribute $\pi^n d\ell$, where π^n represents an estimate of the asymptotic population for the entire history n and acts as a stochastic weighting factor. Each tally interaction to the differential region of phase space can be thought of as a source point and following the neutron to obtain its asymptotic population yields a result proportional to the adjoint flux at that point. Since the estimate is same for a single history, it will be influenced by each differential region the trajectory crosses. A trajectory that enters a region of low importance, such as one close to a vacuum boundary, will, in the limit of large trials, be weighted by a lower asymptotic population than a trajectory that only stayed in regions of high importance. This is because of the higher likelihood of leakage; trajectories that leak from the system, by definition, weight all their respective contributions by zero. On average, this will be lower than ones that stay in the center of the system.

To summarize, the expected magnitude of each contribution for each differential region within a history is proportional to the interaction probability (the flux) and the asymptotic population (the adjoint). Therefore, each differential contribution is an estimate of the adjoint-weighted flux. Integrating these weighted differential contributions over region j over a large number of histories will result in an adjoint-weighted flux averaged over this region. The tally over region j is therefore

$$T_j = \frac{1}{V_j}\frac{1}{N}\sum_{n=1}^{N}\iiint \pi^n d\ell^n \; dV_j \; d\Omega_j \; dE_j = \frac{C}{V_j}\iiint \psi^\dagger d\ell \; dV_j \; d\Omega_j \; dE_j = C\left(\psi^\dagger\psi\right)_j. \tag{8}$$

To make this more concrete within the framework of a Monte Carlo simulation, suppose each new history is assigned a progenitor index p. Any progeny will inherit this index. After a long time, the number of progeny with index p will reach a steady state value. The asymptotic population π for progenitor p can be estimated with a track length estimator:

$$\pi_p = \sum_{\tau\in p}\left(\bar{\nu}\Sigma_f\ell\right)_\tau \tag{9}$$

Here $\bar{\nu}$ denotes the average number of neutrons produced per fission for this trajectory, Σ_f is the macroscopic fission cross section, and ℓ is the length of this trajectory. The summation is carried over all trajectories that have a neutron with progenitor index p over this entire generation.

With this notation, the analog adjoint-weighted flux tally is

$$C\left(\psi^\dagger\psi\right)_j = \frac{1}{V_j}\frac{1}{N_p}\sum_p \pi_p \sum_{\tau\in p}\ell_\tau. \tag{10}$$

N_p is the number of progenitors, which may be different than the total number of histories.

Extending the tally to handle non-analog simulation (i.e. use of implicit capture, weight cutoffs, etc.) requires extra caution. There are two concerns: the appropriate weight to use in the tallies and handling branching issues with implicit capture.

In the power iteration method, each particle begins with an equivalent weight w_0 such that the sum of each weight is the total source weight W. Nominally, the flux tally would be modified such that each trajectory is multiplied by its current weight w. However, the question that arises is how each progenitor, effectively launched in each differential element of phase space, should be weighted. While the track has a weight in the flux, it turns out each progenitor

should be equal. This implies an additional factor of w_0/w for each trajectory. Tests on numerous cases show consistent answers for analog and non-analog cases. The tally in the non-analog case is therefore

$$C\left(\psi^\dagger\psi\right)_j = \frac{1}{V_j}\frac{1}{W_p}\sum_p \pi_p \sum_{\tau\in p} w_{0,p}\ell_\tau. \tag{11}$$

Another issue arises from branching events. Even in analog physics, n,xn reactions create this concern as well. However, it is especially problematic for implicit collisions. In a non-analog simulation, a neutron that collides with a fissile nucleus has its weight partitioned among all possible reactions. The partitioning for each reaction is determined by the ratio of the reaction's cross section to the total cross section at the collision site.

The process causes, in a simulated sense, the fission chains to branch. One time line spawns neutrons in the next generation, while the other continues in the current generation. Any events that occur after this branching event may not be able to cause the same sequence of neutron progeny. For illustration, consider a branching event of a fast neutron that thermalizes and has another fission event. Fast fission events will produce progeny of a different spectrum and they will behave differently than the thermal fission progeny.

It is therefore required that the tallies obey the notion of causality. Whenever a branching event occurs due to implicit capture or an n,xn reaction, a branch must be introduced. The asymptotic population from that branch is multiplied only by tracks that came before the branching event and none after. Tests of numerous problems demonstrate that this is required for results to come out consistently between analog and non-analog cases.

II.C. Kinetics Tallies

The kinetics parameters are evaluated in terms of ratios of integrals over the entire core of importance weighted functions. Each of these integrals must be computed. There are three tallies: the adjoint-weighted neutron density, the adjoint-weighted fission source, and the adjoint-weighted delayed fission source.

The tally for the adjoint-weighted neutron density is very similar to the adjoint-weighted flux except for a $\frac{1}{v}$ factor and the region of integration covers all of phase space. The tally is

$$C\left\langle \psi^\dagger \frac{1}{v}\psi \right\rangle = \frac{1}{W_p}\sum_p \pi_p \sum_{\tau\in p}\frac{1}{v_\tau} w_{0,p}\ell_\tau. \tag{12}$$

The other two tallies closely follow the derivation of the adjoint-weighted source. The fission source $\frac{1}{k}\mathbf{F}\psi$ (k denotes the fundamental k-eigenvalue for the system) is estimated every cycle in the power iteration method. By following each fission source neutron's progeny to its asymptotic population, it is possible to estimate the importance at each of those points. Thinking along the lines of breaking the problem into infinitesimal regions, the tally would be proportional to the fission source intensity and its importance. Integrating over the entire domain yields the adjoint-weighted fission source.

The total fission source uses all of the fission neutrons, whereas the delayed fission source only uses ones produced from delayed fission (denoted by β). Note that a factor of $\frac{1}{k}$ exists in each of these tallies because the fission source is renormalized to $k = 1$ every cycle.

$$\frac{C}{k}\left\langle \psi^\dagger \mathbf{F}\psi \right\rangle = \frac{1}{W_p}\sum_p \pi_p w_{0,p}, \tag{13}$$

$$\frac{C}{k}\left\langle \psi^\dagger \mathbf{B}\psi \right\rangle = \frac{1}{W_p}\sum_{p\in\beta} \pi_p w_{0,p}. \tag{14}$$

The constant multipliers C are identical for each tally so that when a ratio is taken, they divide to unity. From these tally definitions it is possible to estimate Λ and β.

There are two issues with calculating error. The first is doing the propagation correctly. Secondly, is very important to remember that these three tallies are correlated. As such, the computation of the covariances between the tallies is required.

The uncertainty of a ratio of stochastic quantities $z = \frac{x}{y}$ can be obtained from

$$\left(\frac{\sigma_z}{z}\right)^2 = \left(\frac{\sigma_x}{x}\right)^2 + \left(\frac{\sigma_y}{y}\right)^2 - 2\frac{c_{xy}}{xy}. \tag{15}$$

The covariance between two random variables can be found by

$$c_{xy} = \mathbb{E}(XY) - \mathbb{E}(X)\mathbb{E}(Y) = \left(\frac{1}{N}\sum_{i=1}^{N} x_i y_i\right) - \left(\frac{1}{N}\sum_{i=1}^{N} x_i\right)\left(\frac{1}{N}\sum_{i=1}^{N} y_i\right). \tag{16}$$

To do this requires accumulating the products of each tally score along with the the individual scores and the squares.

II.D. Perturbation Tallies

Many Monte Carlo codes such as MCNP use the differential operator method to calculate the change in tally response due to a small change in the configuration. While this method works well in some cases, it produces very suspect results for many types of perturbations in eigenvalue calculations. For these cases, a more accurate formulation for estimating reactivity changes is done with a ratio of adjoint-weighted quantities.

The numerator requires finding the importance weighting of the function $\mathbf{P}\psi$. The three components reflect the change in the total collision rate, the change in the source from scattering, and the change in the fission source. Each of these terms takes the following form:

$$C\left\langle \psi^\dagger \Delta\Sigma_t \psi \right\rangle = \frac{1}{W_p}\sum_p \pi_p \sum_{\tau\in p} \Delta\Sigma_t w_{0,p} \ell_\tau, \tag{17}$$

$$C\left\langle \psi^\dagger \Delta\mathbf{S} \psi \right\rangle = \frac{1}{W_p}\sum_p \pi_p w_{0,p} \frac{\Delta\Sigma_s}{\Sigma_s}, \tag{18}$$

$$\frac{C}{k}\left\langle \psi^\dagger \Delta\mathbf{F} \psi \right\rangle = \frac{1}{W_p}\sum_p \pi_p w_{0,p} \frac{\Delta\Sigma_f}{\Sigma_f}. \tag{19}$$

Σ defines the macroscopic cross section where the subscripts t, s, and f denote total, scattering, and fission respectively. $\Delta\Sigma$ represents the user defined perturbation of the cross section in question. For each score these macroscopic cross sections and their perturbations are evaluated at the current location in phase space.

The collision part of the perturbation is tallied every track, while the other two are tallied at each scattering or fission source event. It is possible to rearrange to tally all of the contribution in the original generation together and later weight by the asymptotic population several generations later. This removes the need to factor in correlation between each of these three sections of the perturbation tally. However, computing the uncertainty of $\Delta\rho$ still requires the correlation between the importance-weighted perturbation and fission source. This is done so in a similar way described for computing uncertainties of the kinetics parameters.

III. IMPLEMENTATION

A framework for computing these tallies needs to be implemented in a production Monte Carlo radiation transport code. The Monte Carlo N-Particle (MCNP) code is widely used and rigorously benchmarked, and is therefore the program of choice for implementation. The design should be modular, use the existing random walks, and be efficient with memory usage.

III.A. Calculation Flow

Each cycle in a criticality calculation represents a fission generation. The cycles are given one of the following attributes: original, latent, or asymptotic. In an original cycle, every source neutron is given a progenitor ID number at birth and at each branch point. During this cycle, the contributions for each progenitor are stored for every adjoint-weighted tally. At fission events, the progeny receive the progenitor ID. During latent generations, the neutrons transport normally; no additional contributions are collected, but the progenitor IDs pass on to any subsequent progeny. After a certain number of latent generations, an asymptotic cycle occurs. During the asymptotic cycle, the neutron production for each history within the cycle is estimated. The sum of these estimates represents the steady state population for the specific progenitor and is multiplied by the contributions obtained in the original generation. These products are summed to get an estimate of the particular tally of interest. After an asymptotic generation, all

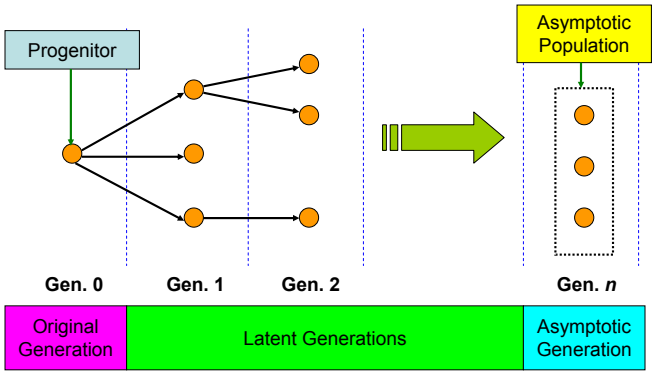

Figure 1: Illustration of generation attributes of cycles.

the progenitor IDs are cleared and the next cycle is an original generation. This process continues until the end of the calculation. A single progenitor set can be seen in figure 1.

It is possible to dispense with this cataloging of cycles and instead do everything simultaneously. This incurs a greater cost of memory, even though it would be more efficient from a CPU time point of view. Since the storage scales linearly with batch size, for large batch sizes (in excess of hundreds of thousands) such calculations may become memory prohibitive on modern machines. This is especially true when numerous branching events due to implicit capture can occur. For this reason, each cycle only has one generation category even though it is possible to run overlapping sets every cycle.

No additional random walks are required. The adjoint-weighting is done with the existing information in a forward calculation. This means the increase in CPU time is very small; however, there is an added cost of memory usage that scales linearly with batch size.

III.B. Modules & Data Structures

Modularity is a key concern because actual changes to the code itself should be limited because of verification concerns. Also, this framework should, in principle, be portable to other Monte Carlo codes. Note that MCNP is largely written in Fortran 90, and so are these routines.

The framework is accomplished by creating a module that contains all the high level routines, data structures, and variables for accomplishing the intergenerational accounting. This module, called *kadjoint*, sits on top of more application specific modules that compute quantities like kinetics parameters or changes in reactivity due to defined perturbations. The routines in *kadjoint* are called in key places throughout the normal operation of the code. Examples include every fission source, every collision, every track, and at the beginning and end of each cycle. The goal is to limit the amount of modifications to these function calls.

The one modification to the actual code itself must be done with passing the progenitor IDs between generation. The power iteration method necessitates some storage of the state (position, emission energy, etc.) of each progeny neutron. Another state variable, the progenitor ID, must be defined. Also, delayed neutron fraction calculations necessitate the storage of the delayed neutron precursor index.

III.B.1. Progenitor Data Structure

An efficient way to facilitate these calculations is to create an derived type that resembles a linked list. The elements of this derived type that are absolutely required are integer storage for the progenitor ID and a pointer to this data type to record any branch associations. Applications may necessitate more data be stored within this structure.

Every time a new progenitor ID is required, a factory function generates the ID and stores it within the next available element in the linked list. Further, if the event was from branching (n,xn or a continued fission path in an implicit collision) the pointer must associate with the corresponding parent that produced this branched progenitor (branch index denoted by $b = 0...B$). At the end of the history, all branches will link back to the element that was assigned when the neutron was emitted as part of the fission source. This is illustrated in the top row in figure 2.

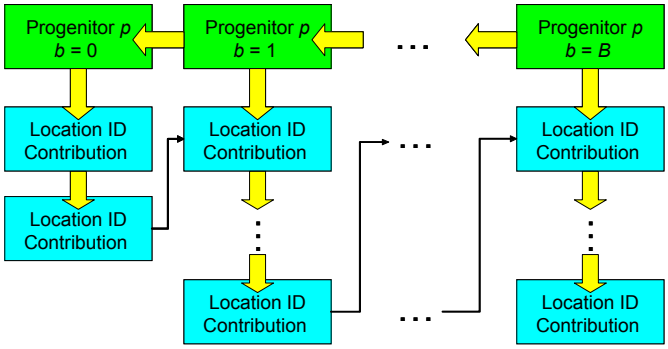

Figure 2: The two linked list data structures used for progenitor accounting. The thick arrows denote a pointer association and the thin black arrows denote connection in array memory.

III.B.2. Data Structure for Local Quantities

Most applications such as reactor kinetics parameters are inherently global in nature. However, the adjoint-weighted flux as a function of space and energy can help provide guidance to a reactor designer in determining which regions are most affected by perturbations. Such adjoint-weighted flux tallies are challenging from a memory management standpoint. The code must remember the contribution in each region for each progenitor. A large number of regions will probably require a prohibitive amount of storage.

Fortunately, this can be addressed because a large number of regions will receive no contributions and need not be recalled. Another linked list is appropriate for this application (bottom part of figure 2).

This new data structure is an array so that each element must remember a location ID (mesh element number) and the magnitude of its contribution in that element. Also, a pointer is required to link all the elements of the same progenitor together. The progenitor data structure must be modified so that each element contains a pointer to this new derived type.

IV. VALIDATION & VERIFICATION

To ensure the validity of this approach, the results are validated against either multigroup discrete ordinates results or experimental measurements. The three general classes of calculations performed are: local adjoint-weighted fluxes, point reactor kinetics parameters, and reactivity changes due to first order perturbations. All of the calculations in this section approximated the iterated fission probability with five latent generations. This was shown by way of experimental tests to be adequate for these problems.

IV.A. Adjoint-Weighted Fluxes

A discrete ordinates code can easily compute the flux and the adjoint flux as a function of various points in phase space for a multigroup problem. By combining these, the adjoint-weighted flux averaged over a domain can be computed. MCNP can easily operate in multigroup mode as well as the standard continuous energy one.

A three energy group 1-D reflected slab ($k = 1$) problem with artificial cross section data is run in both Partisn and MCNP. The center-line on the left is a reflecting boundary. To the right is a 10 cm core region followed by a 30 cm reflector region. The right boundary is a vacuum boundary. The cross sections are given in table I. The results are obtained and scaled by a multiplicative constant (the arbitrary factor C in the tallies). The adjoint-weighted flux from MCNP scales very well with the Partisn equivalent.

Also, a similar 1-D slab problem is run with continuous energy physics. The setup is similar to before with a 10 cm core region and a 5 cm reflector region. The core region is a 5 percent enriched uranium-water mixture (atomic fuel to moderator ratio of 0.05) with a mass density of 1.8 g/cc. The reflector region is pure water with a mass density of 1.0 g/cc. This shows physically expected behavior in the fast (> 100 keV), intermediate, and thermal (< 0.625 eV) energy regimes.

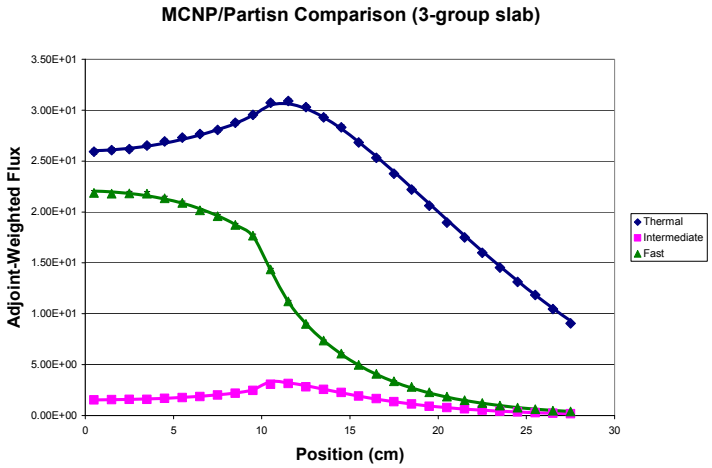

Figure 3: Flux and adjoint-weighted flux in the 3-group reflected slab reactor. The solid line is the reference Partisn case and the dots are the results from the MCNP tallies.

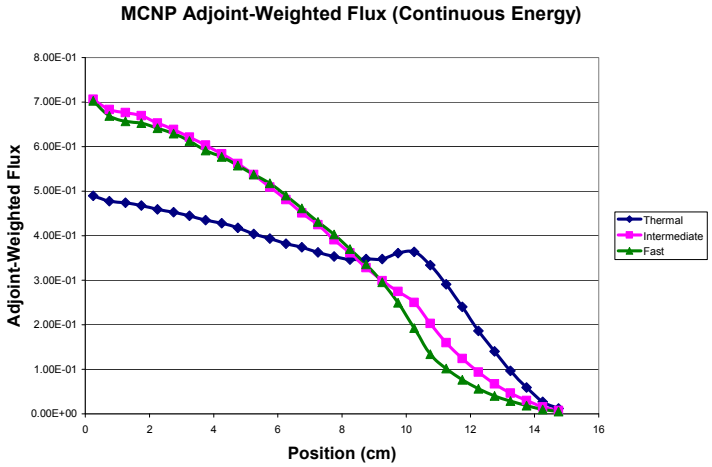

Figure 4: Flux and adjoint-weighted flux for a continuous energy reflected slab reactor.

IV.B. Point Reactor Kinetics Parameters

The results from Partisn can be used to compute the neutron generation time Λ for multigroup problems. Numerous 1-D multigroup test cases are run: slabs versus spheres, different group structures, different spectra, bare versus reflected, and varied criticality states. Table II summarizes the results of a few highlighted calculations and they are

Table I: Artificial cross section data for 1-D reflected slab problem.

	g	Σ_t	$\bar{\nu}\Sigma_f$	Σ_γ	χ	$\Sigma_{s,g\to1}$	$\Sigma_{s,g\to2}$	$\Sigma_{s,g\to3}$
Core	1	0.05	0	0	1	0.05	0	0
	2	0.15	0	0.01	0	0	0.14	0
	3	0.15	0.0238	0	0	0	0	0.14
Refl	1	0.2	0	0	-	0.15	0.05	0
	2	0.2	0	0	-	0	0.05	0.15
	3	0.2	0	0	-	0	0	0.199

Table II: MCNP Λ results compared to equivalent Partisn Calculations.

Case	Partisn Λ	MCNP Λ
Bare, Fast Slab (4-gp)	9.8100 ns	9.8099 +/- 0.0010 ns
Reflected, Thermal Slab (4-gp)	135.2222 μs	135.0876 +/- 0.2081 μs
Bare, Intermediate Slab (8-gp)	112.0086 ns	112.5003 +/- 0.4341 ns
Reflected, Fast Sphere (4-gp)	10.1895 ns	10.1969 +/- 0.0258 ns
Subcrit. Bare Slab (4-gp)	10.1715 ns	10.1714 +/- 0.0138 ns
Supercrit. Bare Slab (4-gp)	9.6725 ns	9.6752 +/- 0.0131 ns

in excellent agreement. Numerous other cases were run and show similar agreement.

It is also possible to compare the continuous energy aspect by comparing with criticality experiments. Experimental results of Rossi-α ($-\beta/\Lambda$) are compared with the results obtained in MCNP. The results (Table III) show fairly good (albeit not as perfect) agreement.

Table III: Rossi-α MCNP results compared to experimental measurements.[7]

Experiment	Measured α (ms^{-1})	MCNP α (ms^{-1})
Godiva	-1100 +/- 20	-1136 +/- 12
Jezebel	-640 +/- 10	-643 +/- 13
Flattop-23	-267 +/- 5	-296 +/- 5
BIG TEN	-117 +/- 1	-122 +/- 2.5
STACY-29	-0.122 +/- 0.004	-0.128 +/- 0.002
WINCO-5	-1.109 +/- 0.003	-1.153 +/- 0.037

IV.C. Perturbations

Using MCNP and Partisn, $\Delta\rho$ is computed for a few small perturbations to the cross section data of a four group, 1-D slab problem. All the perturbations are in the thermal energy group, or group 4. The Partisn reactivity change is exact: run two separate calculations and compare the k-eigenvalues of both. The MCNP calculation is a first-order perturbation using the tallies detailed previously. The results for the various perturbations are given in Table IV and show very good agreement.

However, these results are very preliminary and further work needs to be done for a continuous energy implementation. Also, there are some limits of applicability to this method because the existing random walk sequence is used. Namely, it is impossible to measure the effect of something that was completely absent from the original calculation; however, doing so is possible the other way around.

Table IV: $\Delta\rho$ for various perturbations for equivalent Partisn and MCNP Calculations.

$\Delta\sigma_{\gamma,4}$	$\Delta\sigma_{f,4}$	$\Delta\sigma_{s,4\rightarrow4}$	$\Delta\bar{\nu}_4$	Partisn $\Delta\rho$ (pcm)	MCNP $\Delta\rho$ (pcm)
+0.05 b	0.00 b	0.00 b	0.00	-32.839	-32.748 +/- 0.152
0.00 b	+0.05 b	0.00 b	0.00	+49.517	+49.506 +/- 0.308
+0.05 b	+0.05 b	0.00 b	0.00	+16.644	+16.758 +/- 0.406
0.00 b	0.00 b	0.00 b	+0.01	+33.068	+32.901 +/- 0.107
+0.05 b	+0.05 b	0.00 b	+0.01	+49.767	+49.988 +/- 0.484
0.00 b	0.00 b	+0.05 b	0.00	+2.166	+2.192 +/- 0.283

V. CONCLUSIONS & FUTURE WORK

The procedure for adjoint-weighting of continuous energy Monte Carlo tallies has been derived and applied to calculating quantities in reactor physics. Details of the implementation in the production Monte Carlo code, MCNP, have been given. The method and implementation have been tested against experimental data and discrete ordinates calculations. The results for adjoint-weighted fluxes, point reactor kinetics parameters, and a few changes in reactivity due to small perturbations are in good agreement.

Further testing and development, especially with the continuous energy perturbation techniques need to continue. Also, multidimensional cases need to be tested as well. The iterated fission probability is for an infinite number of generations, but has been approximated with a finite truncation. More work is required to quantify the impact of this truncation on various quantities.

A future application for the adjoint-weighting methodology is the calculation of sensitivities of nuclear data using continuous energy Monte Carlo.

REFERENCES

[1] G.R. KEEPIN, *Physics of Nuclear Kinetics*, Addison-Wesley (1965).

[2] X-5 MONTE CARLO TEAM, "MCNP - A General N-Particle Transport Code, Version 5, Volume I: Overview and Theory," Tech. Rep. LA-UR-03-1987, Los Alamos National Laboratory (2003).

[3] R.E. ALCOUFFE, R.S. BAKER, J.A. DAHL, and S.A. TURNER, "Partisn Manual," Tech. Rep. LA-UR-02-5633, Los Alamos National Laboratory (2002).

[4] H. HURWITZ, JR., "A Note on the Theory of Danger Coefficients," Tech. Rep. KAPL-98, Knolls Atomic Power Laboratory (1948).

[5] G. BELL and S. GLASSTONE, *Nuclear Reactor Theory*, Van Nostrand Reinhold (1970).

[6] A. RADKOWSKY, editor, *Naval Reactors Physics Handbook*, Naval Reactors, United States Atomic Energy Commission, vol. 1, pp. 864–869 (1964).

[7] J.B. BRIGGS, editor, *International Handbook of Evaluated Criticality Safety Experiments*, Nuclear Energy Agency (2004).