SANDIA REPORT
SAND2009-6227
Unlimited Release
Printed September 2009

Automated Monte Carlo Biasing for Photon-Generated Electrons Near Surfaces

Brian C. Franke, Ronald P. Kensek, Martin J. Crawford

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2009-6227
Unlimited Release
Printed September 2009

Automated Monte Carlo Biasing for Photon-Generated Electrons Near Surfaces

Brian C. Franke, Ronald P. Kensek, and Martin J. Crawford
Radiation Transport Department
Sandia National Laboratories
P.O. Box 5800, MS 1179
Albuquerque, NM 87185

Abstract

This report describes efforts to automate the biasing of coupled electron-photon Monte Carlo particle transport calculations. The approach was based on weight-windows biasing. Weight-window settings were determined using adjoint-flux Monte Carlo calculations. A variety of algorithms were investigated for adaptivity of the Monte Carlo tallies. Tree data structures were used to investigate spatial partitioning. Functional-expansion tallies were used to investigate higher-order spatial representations.

Acknowledgments

We wish to acknowledge the many valuable contributions made to this project. Tim Tautges was involved in formulating the plan for this project and provided helpful ideas and discussion throughout. Jason Kraftcheck implemented the tree capabilities and associated data-storage capabilities in the MOAB library and otherwise assisted in guiding our efforts. Tom Laub provided much helpful feedback and critique in reviewing the plans, coding, documentation, and papers produced during this project. Tom Booth generously made himself available for discussions and provided helpful critiques of our work. We would like to acknowledge Len Lorence for his support and guidance throughout this project.

This work was supported by Sandia's LDRD program.

Table of Contents

Table of Figures

This page is intentionally blank.

I. INTRODUCTION

Our goal was to produce an automated technique for the biasing of Monte Carlo coupled electron-photon calculations, even for problems where electrons have high importance only near surfaces. As computing capabilities have dramatically increased, the problems being analyzed with Monte Carlo calculations have greatly increased in complexity. This project was motivated by a desire to develop biasing methods that can effectively improve the efficiency of calculations without heavily relying on user intuition about the physics and geometry of the problem. We planned to accomplish this by implementing a weight-windows method that could accurately represent the adjoint importance map for an arbitrary problem. The primary challenge to be addressed was in obtaining accurate importance maps in problems where electron ranges are 4 or 5 orders of magnitude smaller than the system geometry. This required an adaptive, efficient data structure. To achieve such efficient data representation, we investigated a combination of tree data structures and functional expansion techniques. Tree structures can be used to partition phase-space. A functional expansion can represent the importance map within each partition. Allowing both the tree and expansion to adaptively refine is similar to hp-adaptivity in finite element methods. For our problems, the adaptation must be based on particle importance. We calculated importance as represented by fluxes in adjoint Monte Carlo simulations.

The greatest impact of this project will be for problems that require accurate determination of electron flux. For instance, a variety of problems require the calculation of electron escape or surface emission. Dose enhancement, a phenomenon arising from variation of the electron flux at material interfaces, can occur at many locations inside an electronic component. The method could also expedite satellite shielding analyses involving deep penetration of radiation.

Throughout this report, we will refer to several types of tree data structures. We have adopted a terminology which may not strictly conform to some definitions of these tree structures, so we will document our definitions here. The data structure that we refer to as an **octree** is a binary space-partitioning tree that resembles an octree. In other words, each level of the tree divides space into two portions (not eight, as is typical of octrees). Each division is axis-aligned and made at the center of the node. The orientation of the cuts is determined by the orientation of the preceding cut in the tree structure, such that three successive cuts are made in the three unique axial directions and result in the cuts consistent with a single level of a traditional octree. The data structure that we refer to as a **kd-tree** is a binary spatial-partitioning tree in which each division is axis-aligned but can be made at any location within a node. The cuts at succeeding levels can be made in any of the 3 axial directions. In some of our implementations, these cuts are always made at the center of the cell; this is similar to the octree algorithm, but since the cuts may be oriented in any of the three axes, we refer to these algorithms as kd-trees. The binary spatial-partitioning algorithm that we refer to as a **bsp-tree** allows each division to be arbitrarily oriented and arbitrarily located within a node. To constrain requirements on the complexity of the adaptivity algorithms and to facilitate functional-expansion representations, the bsp-tree structures that we have investigated were constrained to always contain hexahedral nodes.

Section II discusses biasing methods used in Monte Carlo radiation transport calculations. While this project has not developed any new biasing methods, efficient biasing of calculations is the motivation for the adaptivity algorithms that have been investigated, and this section provides

background on the biasing methods of interest. Section III discusses investigations of adaptivity algorithms designed to resolve electron-range maps. These algorithms were pursued with the intent of using electron-range data for biasing or to improve the importance-map adaptivity. This approach was ultimately abandoned, but we include some discussion of our investigations. Section IV reviews the adaptivity algorithms investigated in this project for resolving importance maps. Sections IV.A describes the h-adaptivity algorithms that were investigated. These are the methods that have proven most effective for generating weight-window settings. Section IV.B contains a discussion of a p-adaptivity algorithm that was presented at the ICRS-7/RPSD-2008 conference. An expanded and more detailed version of this section has been submitted for journal publication[HYPERLINK \l "Fra08" 1], and interested readers are encouraged to look for that article. Section IV.C contains a paper on hp-adaptivity presented at the M&C-2009 conference**Error! Reference source not found.**, with some additional results and discussion. Section V is based on another paper presented at the M&C-2009 conference**Error! Reference source not found.**, also including additional results and discussion. We offer some brief concluding remarks in Section VI. Finally, for completeness, we have included several Appendices. The original project proposal is included in Appendix A. Some discussion of practical issues in formulating an effective source of adjoint particles for simulating the detector response of electron emission into a cavity is included in Appendix B. In Appendix C, we provide our derivation of an equation for calculating Legendre-moment coefficients for functional expansions based on particle track-length tallies in a Monte Carlo code.

II. BIASING

In this section we offer a brief (and admittedly somewhat rambling) discussion of various topics related to biasing Monte Carlo particle transport.

Zero variance biasing methods have been investigated in the radiation transport community**Error! Reference source not found.Error! Reference source not found.Error! Reference source not found.Error! Reference source not found.Error! Reference source not found.Error! Reference source not found.**. While these methods may not apply to practical problems, they do offer many insights into the application of biasing methods.

Adaptive methods have been explored to provide exponential convergence of Monte Carlo calculations, rather than the standard $\frac{1}{\sqrt{N}}$ convergence. Such approaches were first demonstrated for discrete-state transport problems**Error! Reference source not found.Error! Reference source not found.** and then extended to continuous transport problems**Error! Reference source not found.**. Both sequential correlated sampling and adaptive importance sampling are methods that have been shown to produce exponential convergence of Monte Carlo calculations**Error! Reference source not found.Error! Reference source not found.**. Application of adaptive importance sampling methods to realistic problems have been investigated for particle transport by researchers at Claremont Graduate University and LANL**Error! Reference source not found.Error! Reference source not found.Error! Reference source not found.Error! Reference source not found.Error! Reference source not found.**. Ideally, this approach can converge toward a zero variance simulation.

Importance sampling is difficult and intrusive. Each algorithm for sampling physics within the code must be modified to accommodate a rigorous importance-based biased sampling. In some cases this could result in a very expensive sampling algorithm. It seems more reasonable to use other biasing methods, perhaps guided by importance data so that particle weights generally stay near the desired weight and migrate towards regions of higher importance, and rely on weight windows to make adjustments when weights deviate from the desired levels.

Booth's method of biasing the underlying random number sampling**Error! Reference source not found.** (rather than the physics that is being sampled) is an elegant solution to avoid rewriting many sampling algorithms, but it also has some drawbacks. While less intrusive, it is potentially less convenient: there may be no way to use an adjoint simulation to initialize the importance map, and there may be more limitations on modifying the simulation while still using the same importance map. There may also be larger memory requirements, since importance maps as a function of particle type and phase space would also need to be stored for each unique random-number usage within the code.

Our work has not attempted to address the issue of biasing for many detectors. Instead, all detectors were merged into a single detector for calculating the importance distribution, and the result was nearly uniform improvements in runtime figure-of-merit across all detectors. Others

have examined methods for using weight windows to simultaneously bias for multiple detectors**Error! Reference source not found.Error! Reference source not found.Error! Reference source not found.**. Their work suggests that the uniform improvements in FOM that we observed were the result of uniformly sampling from all detectors. To achieve uniform FOMs, one would need to sample adjoint particles from the detectors in inverse proportion to the expected forward result (but note that this requires having estimates for all of the forward results).

It is not clear to us how an adaptive importance sampling algorithm would perform if the importance map was based on a single cavity emission detector but was used for a problem where a set of differential detectors were of interest. For example, one can imagine that the importance map may only "learn" that particles need to reach a localized region in order to contribute (i.e., only one detector out of many), and could proceed to develop a highly skewed importance map that could fail to produce any tallies for many of the detectors, while becoming very efficient for one of the detectors of interest. An approach used by Ueki**Error! Reference source not found.** and others**Error! Reference source not found.** is to learn the importance map by performing a series of forward calculations in which the source is moved away from the detector while using weight-windows biasing from the previous calculation. It is possible that such an approach would address our concerns. We suspect that a few automation mechanisms may be required: (1) to acquire and maintain adequate importance information even in regions of low forward flux, and (2) to iteratively select a forward source that will efficiently improve the importance map (e.g., without biased sampling, a forward source of low energy electrons beyond the electron range from the detector will always have difficulty contributing any importance data). An alternative approach would be to maintain a separate importance map for each detector, but there are many complications in employing multiple importance maps simultaneously and the memory requirements for storing these maps would necessarily increase with the number of detectors. Such an approach for simultaneous zero-variance biasing was discussed by Booth**Error! Reference source not found.**.

There are number of biasing methods available in the latest version of the ITS code**Error! Reference source not found.**. These include collision forcing, biased sampling of relaxation radiation, scaling of bremsstrahlung and electron impact ionization radiation production, Russian roulette of electron production, electron trapping, and selective transport of particle types by zone and energy. The last two of these are not "fair games", as they will change the results to some degree; when used "properly", the objective is to decrease the computational run time without changing the results "too much".

Electron trapping is one of the most effective biasing methods employed across a wide range of applications. Because transport of electrons can easily dominate the expense of a calculation, minimizing that expense is generally the first goal in improving the run time of a coupled electron-photon problem. Electron trapping works by terminating the transport of electrons below some user-specified energy if they are incapable of escaping from their current geometry tally region (because of the finite range of electrons in a medium). This is similar to the concept of "range rejection", which terminates the transport of electrons incapable of reaching a detector tally of interest, but trapping requires no information about which specific tallies are of interest. Electron trapping is much easier to implement and less computationally expensive, but may be

less effective. As will be shown, weight-windows biasing is capable of achieving an effect similar to range rejection, but does so while maintaining a fair game. The disadvantage is that weight-windows biasing must be set-up to target specific detectors, and therefore will avoid tallies to other detectors. For example, if the weight-windows biasing is used to bias the calculation of electron emission, then one cannot expect to obtain good estimates of charge deposition; any electron approaching the cutoff energy and not within range of the emission surface will become decreasingly important and will be increasingly likely to be eliminated by Russian roulette.

Selective transport of particle types by zone and energy generally takes one of two forms: raised cutoff energies and suppression of all electron transport within a zone. Again, weight-windows biasing can provide a more informed and subtle approach. Instead of entirely suppressing transport, the probability of transporting a particle in a given region of phase-space is reduced, based on its importance. (Arguably, the escape zone serves a similar purpose of terminating all particle transport for all particle types, but there is no analogy with weight windows, and it is also used in conjunction with weight windows.)

All other biasing options in ITS are a form of biased sampling. For the expert user, these methods are powerful techniques, but they can easily be abused to produce misleading "overbiased" results by making the simulation of some important physics highly improbable. These methods can complement weight-windows biasing (except for Russian roulette, which is already included in weight-windows biasing, albeit in a "more informed" implementation). In theory, any of these biased sampling methods could be informed by the importance information underlying the weight-window settings. As such, they would be importance-sampling methods. In practice, this is a difficult undertaking, that involves trade-offs between the complexity and effectiveness of the sampling algorithm. Such trade-offs can be viewed as balancing computational time and the resulting variance. The optimum trade-off is likely to be problem dependent. At the same time, it is far too easy to implement an ineffective biased-sampling algorithm that will harm the efficiency of the calculation by simultaneously increasing computation time and variance. Although not documented in this report, one "importance biasing" mechanism was explored in this project: photon collision forcing, in conjunction with weight windows. This method was found to have little or no benefit for the test problems that were examined. However, it is important to note that, as implemented, only the sampling of whether or not a collision occurred was biased; the outcome of the interaction was not biased.

III. ADAPTIVE ELECTRON-RANGE MAPS

In this section we discuss our attempt to develop adaptive electron-range maps, which we explored but ultimately abandoned. The basis for this investigation was that, in many problems, electron range from a detector is the primary indicator of electron importance. If the contours of electron-range from a detector could be estimated, then they could be used for range-rejection biasing and/or could inform the initialization of an adaptive importance map. There appear to be many possible approaches to this problem. (For example, we considered the possibility of using a coarse approximation to electron transport, such as strictly correlating time, energy, and location by straight-ahead and continuous-slowing-down approximations in a wave-front calculation to determine the electron range as a function of distance from a detector.) We investigated two approaches: using tree data structures to resolve the range of electrons and constructing convex-hull contours of ranges from electron energy and location data. These two approaches were chosen primarily because the algorithms for investigating them were readily available.

A. Octrees from Electron Location Data

This method is aimed at identifying regions of the problem that are within an electron range of a detector that is sensitive to electron transport. This is a refinement of the electron trapping capability in ITS, and can be implemented by placing hooks into the current trapping routine. A trapping energy must be applied, such that trapping is only applied to electrons below that energy. This is because high-energy electrons can produce significant numbers of bremsstrahlung photons that can travel anywhere in the problem. This is not an optimal biasing approach. It essentially eliminates electron transport in most of the problem. However, it may be cheaper than the weight windows approach (and sufficiently accurate in some cases).

This approach will be based on performing an electron-only forward or adjoint calculation that begins with source particles at the detector (e.g., an emission surface). During the simulation, the code will report on electron locations. The tree will use this location information to build a map of regions where electrons are reported. The bounds of the final map should represent the range of electrons from the detector. A difficulty arises in the fact that the “practical range” is distinct from the theoretical range. The practical range is a distance beyond which electrons are unlikely to transport and contribute significant dose. A simulation cannot easily identify the practical range. A difficulty in identifying either type of range is illustrated by the depth-deposition curves in Figure 1. (The distinction between the various curves is discussed in the following paragraph.) With the transported particles, when angular scattering is included and energy-loss is not treated deterministically with a continuous-slowing-down approximation, there is no sharp distinction as to where the electron range is reached. The deposition falls off at an increasing exponential rate. The deposition level is correlated with the probability of an electron reaching a given depth, and from Figure 1 we can see that there are small probabilities of electrons reaching ever greater depths.

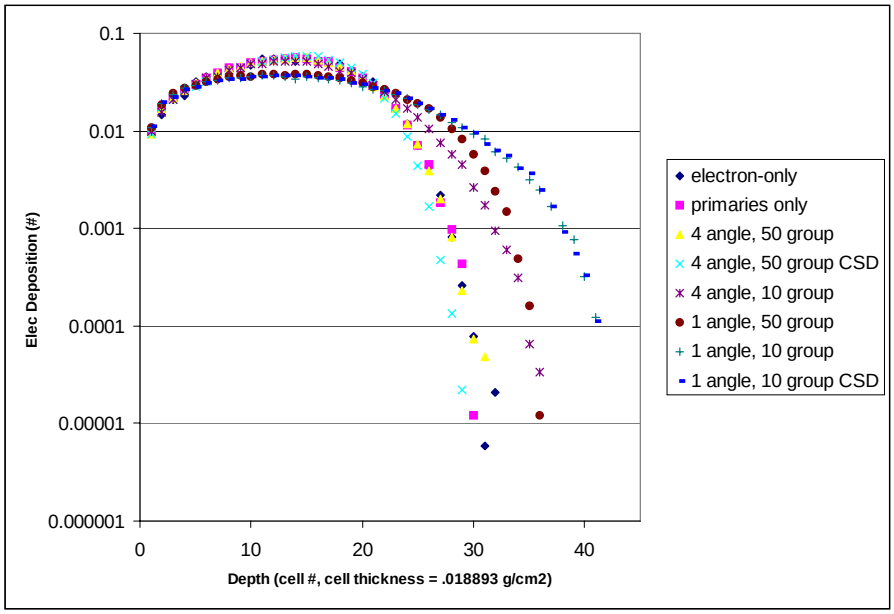

Figure 1. Depth-deposition curves for various approximations for electron transport.

Figure 1 also shows comparisons between various electron transport approximations using MITS (multigroup ITS). The most "faithful" simulation is the "electron-only" case, which neglects photon transport but uses the default code behavior in all other respects. The "primaries only" approximation is faster (by a factor >3) because it neglects transport of secondary electrons. Because the code defines the primary electron as the electron emerging from a knock-on collision with the higher energy, the primary electron is the one with greater range. Neglecting secondary electron transport affects deposition at shallow depths but not at large depths. The default transport in MITS uses 8 Boltzmann scattering angles, 1 Fokker-Planck scattering angle, and 50 cross section groups. The remaining cases omit Fokker-Planck scattering. Using 4 scattering angles slightly increases the computational cost but has no discernible effect on accuracy. Using either 10 groups or 1 scattering angle degrades the accuracy in comparable ways. Using both 10 groups and 1 scattering angle further degrades the accuracy. However, this degradation extends the range of the electrons, increasing the probability of finding electrons at (or beyond) the actual range. These transport approximations (when used to generate a biasing scheme based on accumulated range data) will be conservative by over-estimating the range, and may therefore reduce the runtime required to produce range estimates that are accurate enough for biasing purposes.

To test the accumulation of electron range data, logic was added to an octree to follow this algorithm:

1. A point is located within a leaf of the tree.
2. If the leaf is filled (has previously been identified as containing a point), exit.
3. If the leaf is not at the lowest allowed (most refined) level of the tree, then refine the tree about the point until this condition is met.
4. Mark the leaf as filled.
5. If the sibling leaf is also filled, merge the leaves (iteratively up the tree).

A set of point data was constructed by randomly sampling points on the surface of a sphere. The points were sequentially read into the algorithm described above. The points that were used and the resulting tree are illustrated in Figure 2.

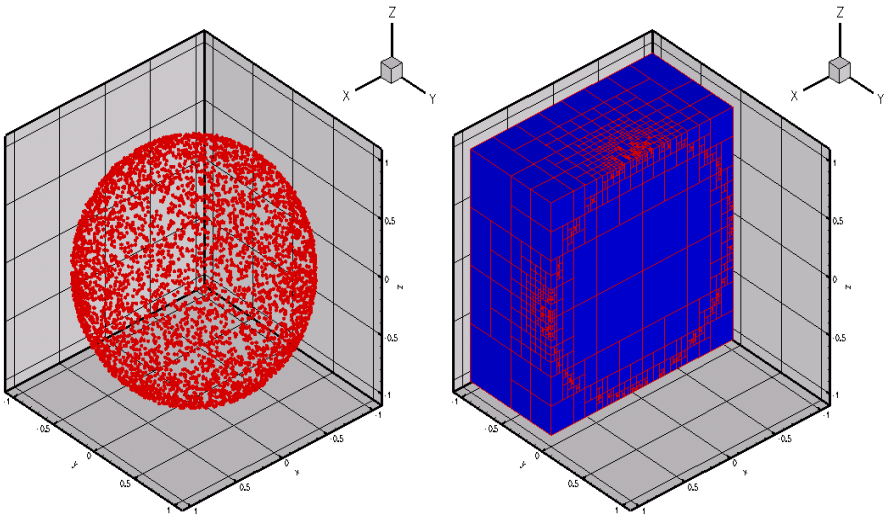

Figure 2. Points on the surface of a sphere and an octree refined about these points.

The tree logic was then refined to store energy-dependent data, so that energy-dependent range contours could be resolved. In the results shown in Figure 3, 6 million data points were inserted into trees of maximum depths 21, 15, 12, and 9. The final leaf counts were 134624, 3721, 875, and 244 and runtimes were 2034, 181, 140, and 130 seconds, respectively. In this problem a rectangular cavity was surrounded on the sides and below by a material in which electrons had a short range and bordered above by a material in which electrons had a long range.

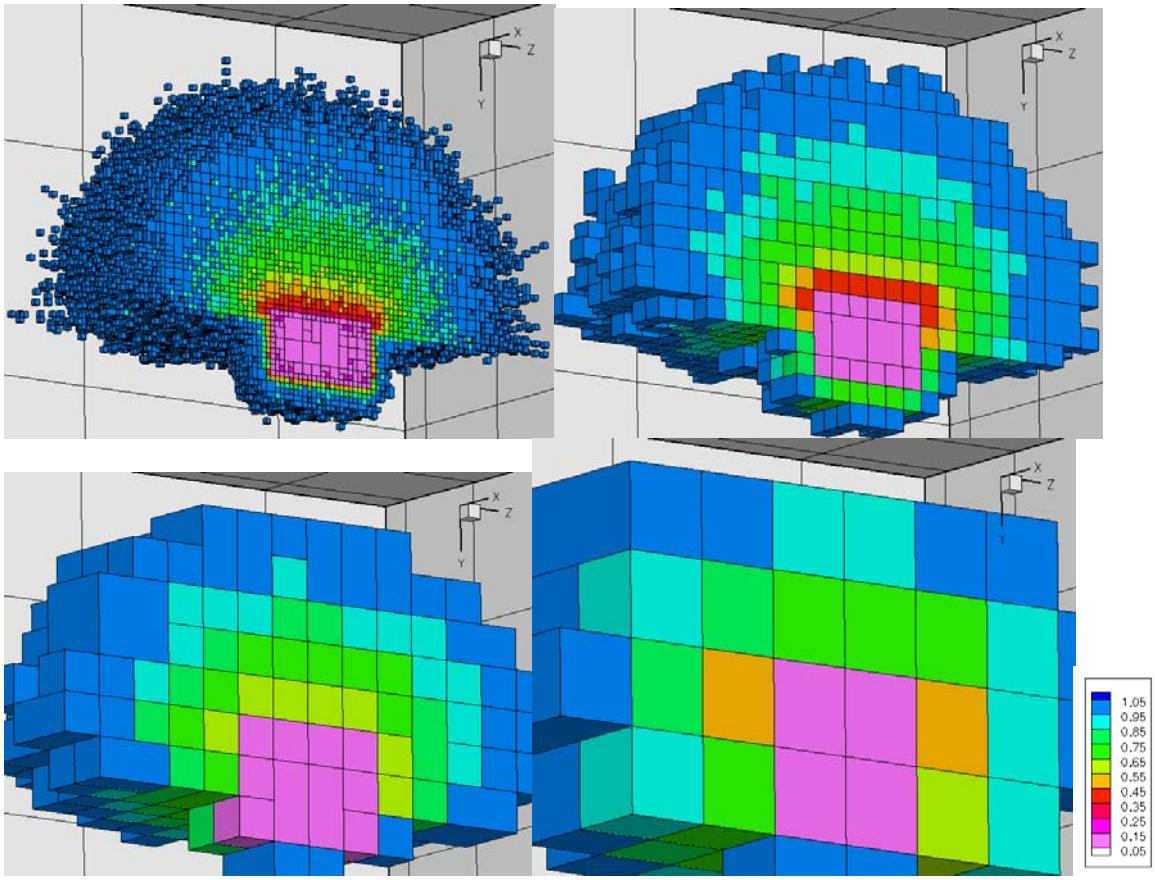

Figure 3. Cutaway views of energy-dependent electron-range data stored in an octree.

B. kd-trees from Electron Location Data

Since isolated electron location data is not sufficient to inform an arbitrary division of space, an accumulation of data is required. In the approach described here, we began with an octree as described in the previous section and used that to select the position and ordering of spatial divisions. This can be performed iteratively: (1) use the tree to store particles near boundaries of point clouds and (2) use those boundary locations to build a kd-tree or bsp-tree as an approximation of the boundary of the cloud. The selection of a proper cost function for building the kd-tree from the leaves of an octree was the main difficulty.

In Figure 4 we show the efficiency gained in going from an octree, with 79000 leaves, to a kd-tree, with 9500 leaves (limited to a comparable refinement level based on the smallest leaf volume allowed). However, in this example, even one octree leaf was substantially wider than an electron range. Even at a depth of 27, the octree (shown in Figure 5 with 485,000 cells) still has a leaf-size larger than the electron range. A kd-tree (limited by the same smallest leaf volume) resolves the electron range (to varying degrees) using 56,000 cells.

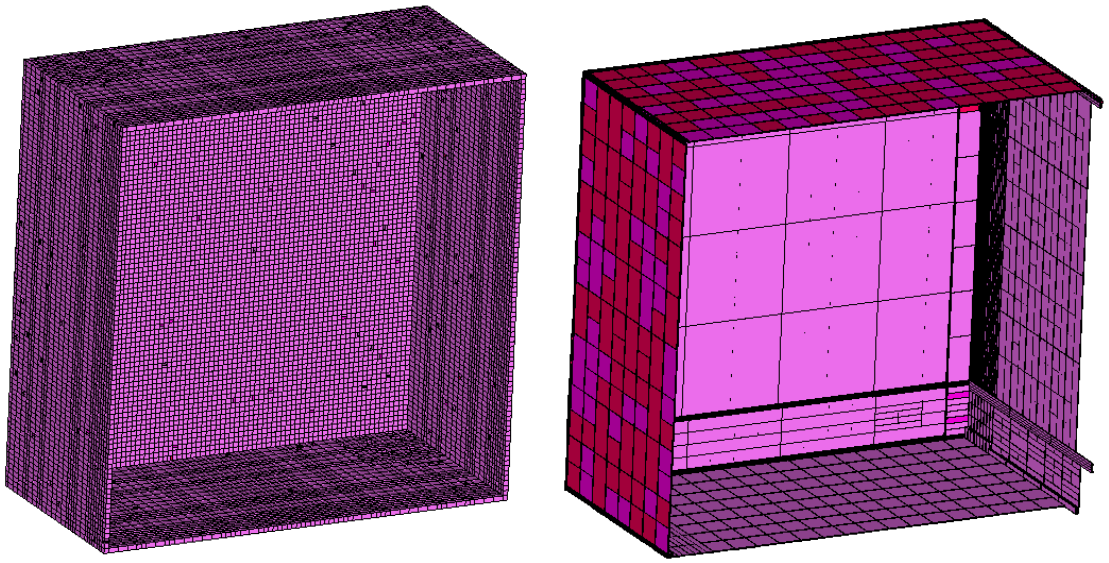

Figure 4. Octree and kd-tree with coarse representations of electron-range data.

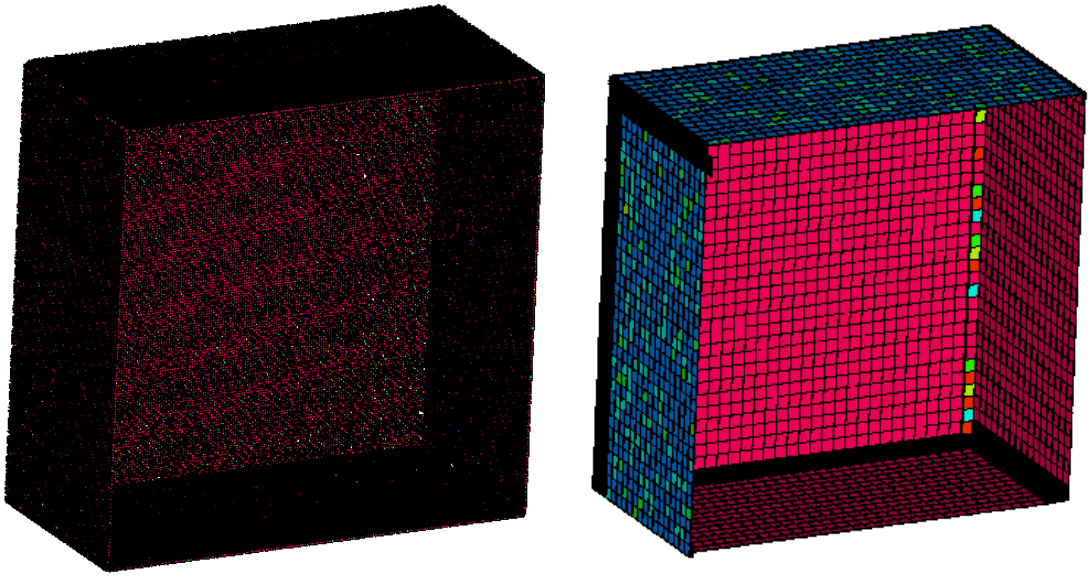

Figure 5. Octree and kd-tree representations of electron-range data.

We explored another approach to building a kd-tree representation of electron range data using binned electron location data within tree leaves. The bins were uniformly distributed along each axis of a leaf. Variations options for utilizing the electron location data within the bins were explored to determine when and where leaves would be divided. Four types of tallies within each bin were used: minimum energy, number of particles, energy of particles, and squared energy of particles. The desire to resolve electron range would suggest that the minimum energy should be the most effective. However, this tally may have been less effective because of greater sensitivity to statistical variation of the data. Also, this tally had difficulty resolving some regions. For example, if the minimum-energy distribution forms a corner, the minimum-energy distribution stored in the one-dimensional axial bins could be flat in all three axis-aligned dimensions. Results generated with this technique are shown in Figure 6. The plot on the left

was generated with the minimum-energy bin data and contained 1103 leaves. The plot on the right was generated with the energy-squared bin data and contained 393 leaves.

Figure 6. kd-tree representations of electron-range data generated with bin-based algorithms.

C. Convex Hulls from Electron Location Data

We investigated using convex-hull algorithms for building contours of electron range. These algorithms accepted the same point-location data as described in the preceding sections to build range trees. The benefit of the convex-hull approach is that they may use much less information to represent the resulting contour. The algorithm that was investigated had the benefit over other convex-hull algorithms of not requiring all data to be stored in memory simultaneously. Concavities in the range contour will never be captured. Figure 7 shows the convex hull for the same data used in the figure above (but rotated 90 degrees), shown growing as additional range data is incorporated.

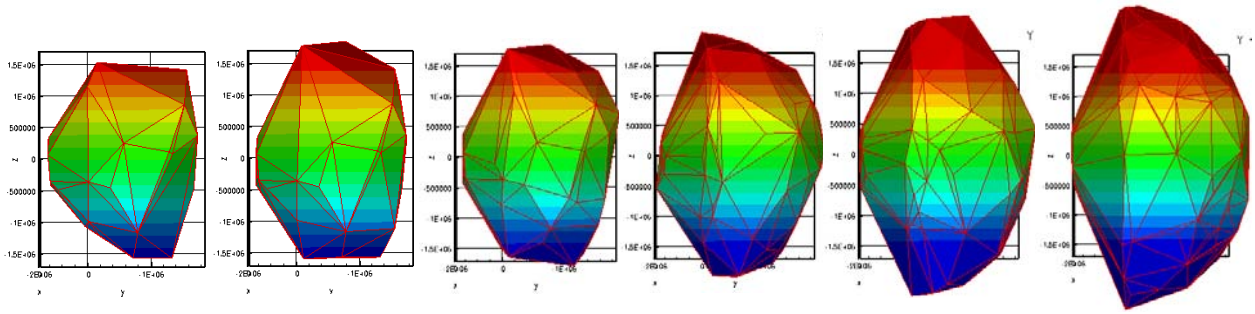

Figure 7. Electron-range data represented by a convex hull.

Figure 8 shows how multiple contours could be used to record the energy-dependent range data. The biggest drawback to this approach is that, as the name implies, the resulting hulls are always convex. We considering building separate convex hulls in partitions of space. In such an approach the ability of the hulls to capture concavities would be limited by the granularity of the spatial partitions. Especially for use in range rejection, it would be necessary to join the resulting convex hulls; that process might be made easier if the partitions of space were overlapping so that the resulting hulls could also overlap and simply be united in Boolean operations.

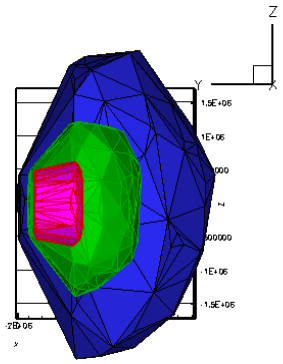

Figure 8. Energy-dependent electron-range data represented by multiple convex hulls.

IV. ADAPTIVE IMPORTANCE MAPS

In this section we discuss the methods that were explored for adaptively resolving particle importance maps. In separate sections, we discuss h-adaptivity (adaptivity of phase-space partitioning), p-adaptivity (adaptivity of functional-expansion representations), and hp-adaptivity (adaptivity of both partitioning and functional representation). We found that for the purposes of setting weight windows, which do not need to be resolved to better than about a factor of 3, partitioning was the best approach. If some type of importance sampling were to be used in biasing a calculation also, functional representations may prove beneficial. We would also note that while only relatively simple algorithms were used for spatial partitioning, the more sophisticated error metrics from the p-adaptivity approaches could be employed to make the h-adaptivity algorithms more robust.

A. h-Adaptivity

In this section we discuss the spatial-partitioning algorithms that were investigated in this project. We have divided the discussion into four parts: neighbor-comparison, functional-variation, bin-based, and grid-based algorithms. In practice, there was much overlap between the approaches, and in most cases they could be employed simultaneously. Neighbor-comparison was always used in some form. The other three are different approaches to representing the variation of importance within a tree leaf: using continuous functional forms, using multiple one-dimensional histogram representations, or using multi-dimensional histogram representations.

There are advantages and disadvantages to each of these approaches, and the best choice is likely to be problem-dependent. The simplest and least computationally expensive is the neighbor-comparison (using only leaf-averaged quantities), and this algorithm is the one that was used in producing all of the weight-window biasing results presented in Section V. The functional-variation approach described here (as examining only linear or quadratic variation) was not robust and was abandoned early in the project. Subsequent work on p- and hp-adaptivity (described in Sections IV.B and IV.C) demonstrated an error-metrics approach that could make functional-variation h-adaptivity more robust, though it would also make the algorithm significantly more expensive. One-dimensional histogram representations performed much better than neighbor-comparison algorithms in resolving one-dimensional variation of importance when the variation was aligned with the leaf and required much less memory than a grid-based algorithm. The grid-based algorithm is described here, as the proposed approach for efficiently resolving non-axis-aligned importance maps.

We began by exploring how electron and photon importance vary in space for a simple one-dimensional problem. The problem detector was electron escape from a surface. Adjoint flux was recorded as a function of distance from the surface in a slab of finite thickness composed of aluminum. Figure 9 shows electron and photon fluxes for a one-dimensional problem with uniform spatial bins. Energy bins were logarithmically distributed from 1 MeV to 1 keV, with each energy bin assigned a different color in the plot. Subzone indices 1-100 share the same spatial location as subzones 101-200, with the former providing the electron flux and the latter

providing the photon flux. The highest energy fluxes are shown in red representing an energy band from 658-1048 keV. The lowest energy fluxes are shown in black, representing an energy band from 10-16 keV.

From Figure 9 we observe the following trends. The electron importance decreases rapidly with distance from the surface until the distance is greater than an electron range. The flat (or flatter) low-magnitude region of the electron fluence reflects the transport of electrons that create bremsstrahlung photons that create electrons that are tallied by the escape detector; in this region, the magnitude of electron importance is lower than the magnitude of photon importance, but the spatial variation of the electron importance is similar to the spatial variation of the photon importance. The flux of photons is relatively flat compared to the electron flux (except at very low energies), but does show the exponential attenuation of photons as a function of distance from the surface. The drop in electron fluence near subzone 100 is because of the finite size of the problem leading to escape of electrons which suppresses the flux from the surface furthest from the detector.

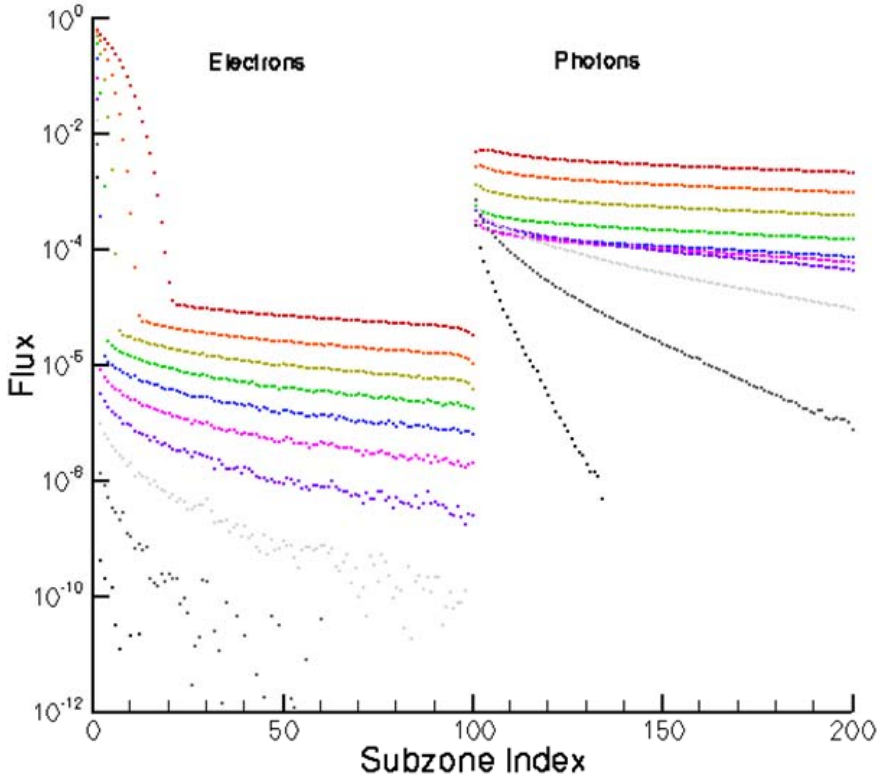

Figure 9. Electron and photon importance as a function of depth in a 1D problem.

i. Neighbor-Comparison Algorithms

The first adaptivity algorithm that we explored used an octree to manage importance-map data. The data was calculated from an analytical function (not real adjoint flux data). The octree stored a flux value on each leaf. (We used a binary spatial partitioning tree. A leaf is defined as a node of the tree that is not further divided. The tree is divided such that any point in the base

node of the tree is contained in one and only one leaf of the tree.) The tree was initialized with a uniform division into leaves specified by an initial depth of the tree. All of the leaves were populated with flux data, based on integrating the analytical function over the volume of the leaf.

Each leaf of the tree was then interrogated to determine whether the difference in flux levels between a leaf and its neighbors warranted dividing the leaf (if the difference in flux was greater than a specified factor). If the leaf was not to be divided, the sibling leaf was interrogated to determine whether the two leaves could be merged (if the difference in flux was less than a specified factor). We found that it was beneficial to have the merging factor slightly less than the square root of the splitting factor to avoid having nodes oscillating between splitting and merging. After each leaf had been interrogated for splitting or merging, the tree was populated with flux values recomputed for each leaf of the tree. This process was repeated until the tree had converged. "Sufficient" convergence was determined by the user and not always rigorously imposed. In simple problems, rigorous convergence was usually achieved. In more complex problems, the tree iteration was usually halted short of complete convergence.

There were many variations on the algorithm. The "octree" algorithm required that tree partition cuts be made by rotating sequentially through the axes, such that leaves in every third level of the tree were always cubes. The "kd-tree" algorithm allowed cuts to be placed in alignment with any of the three axes. For the kd-tree algorithm, a "cut-alignment" mechanism was found to be necessary. That is, when the faces of neighboring cells were not identical (because one had been divided more than another), it was necessary to make cuts perpendicular to the shared faces, so that the flux values across an identical face could be compared before making cuts parallel to the shared face to resolve the difference in flux across the face.

A simple analytic function was used to test the algorithms. The problem was defined on a 10×10×10 box with one corner at the origin. It had a flux value of 1 at radii less than 8. At radii greater than 8, the flux level was 0.01/(radius-7.99). The results are colored using a logarithmic scheme in flux level.

The octree algorithm appeared to function adequately. Figure 10 shows the converged octree results, initiated at a depth of 3 cuts (i.e., 8 leaves), with a splitting threshold at a factor of 10 and a merging threshold at a factor of 3. The tree converged with 5557 leaves after 17 iterations. Not surprisingly, initiating the octree at a greater depth was observed to produce faster convergence.

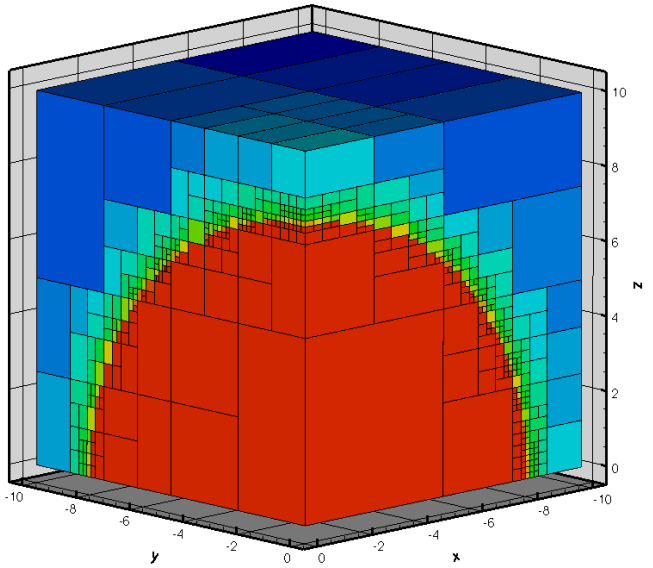

Figure 10. Octree flux representation for a simple test problem.

Results from the kd-tree algorithm are shown in Figure 11. Again, the tree was initiated at a depth of 3 cuts (i.e., 8 leaves), and the refinement algorithm used a splitting threshold at a factor of 10 and a merging threshold at a factor of 3. The tree converged with 1424 leaves after 20 iterations. The previously mentioned cut-alignment mechanism was used in producing these results. Without that mechanism, the tree refinement had a tendency to produce run-away splitting. The aligning splits helped to better resolve the map and isolate the regions of flux gradients. Initiating the kd-tree at a greater depth was observed to produce faster convergence, but the final tree would contain more leaves.

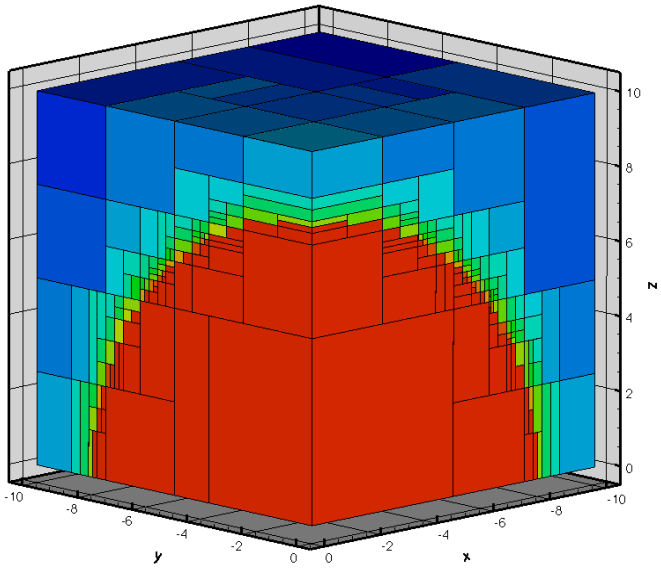

Figure 11. kd-tree flux representation for a simple test problem.

Next, we examined the behavior of the algorithm with statistical uncertainty in the results.[1] A result obtained with statistical uncertainty is shown in Figure 12. In this case, the tree was initiated at a depth of 12 cuts (i.e., 4096 leaves). The refinement algorithm used a splitting threshold at a factor of 6 and a merging threshold at a factor of 2.4, but also imposed the requirement that those thresholds had to be met by at least one-sigma in statistical uncertainty. The tree converged with 4995 leaves after 8 iterations (though the statistical uncertainty caused the tree to continue to change in apparently insignificant ways even after many more iterations). The count size was specified as 10^{-8}. In general, we observed that statistical uncertainty (when properly managed in the refinement algorithm) did not prevent the tree from converging correctly, but could restrict the tree from fully converging to the desired accuracy of representation. However, reducing the level of statistical uncertainty could lead to an adequately converged result. The one-sigma threshold requirement appeared to be sufficient for managing the refinement algorithm. Higher threshold requirements (two or three sigma) also converged correctly, but required lower statistical uncertainty levels to reach the same level of refinement. In the absence of an uncertainty requirement (or a zero-sigma requirement), the tree refinement appeared to be unstable. We also investigated an "uncertainty threshold", which required the relative statistical uncertainty to be lower than some level (such as 10%) before a split or merge could be applied to the leaf. By itself, this mechanism did not appear to be sufficient. In combination with the one-sigma threshold requirement, this feature appeared to be unnecessary.

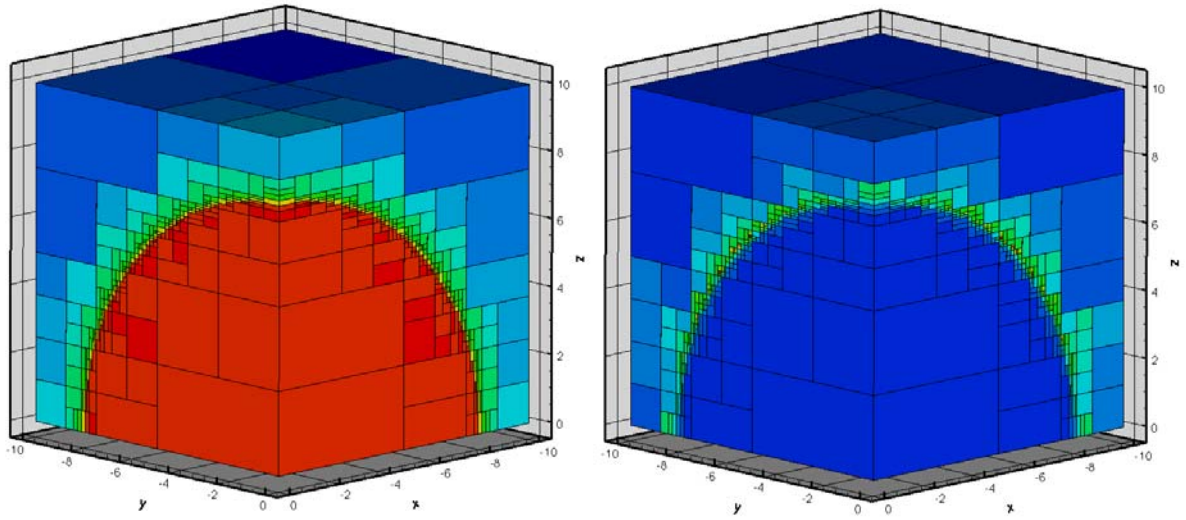

Figure 12. kd-tree flux and statistical uncertainty for a simple test problem.

We also exercised the tree adaptation logic on more realistic test problems. Instead of using a simple analytic test function, one-dimensional data was incorporated from a 20 group transport calculation, with 10 electron and 10 photon groups. Finely resolved one-dimensional spherical-

[1] Statistical uncertainty was introduced in to the quadrature integration of the analytic function using a counting statistics approach at each evaluation point. Using the quadrature weights, a volume was associated with each evaluation. The flux level and volume were equated to a number of counts using a user-specified count-size magnitude. The counting statistics were used to statistically perturb the flux level. For example, if the count size was 10^{-4}, the leaf volume was 0.2, and the flux level was 0.3, then the number of counts would be 600, the relative statistical uncertainty would be about 4%, and the flux variance would be 0.00015. The statistical uncertainty of the volume-integrated flux was estimated by quadrature integration of variance across the volume.

geometry adjoint-flux data was used in place of the test function. This data was similar to that shown in Figure 9 (but did not extend to 10 keV and achieved good statistics throughout). The data was quadrature-integrated over each leaf during the refinement iteration process. In the first test case, the data was interpreted as varying spherically as a function of radius. The resulting tree results are shown in Figure 13 for the lowest-energy electron group. In this case, more than 1E+6 leaves were needed to reduce flux ratios between neighboring cells to factors less than 1000 in all groups. The most difficult group is the lowest energy electrons, shown in the following results. These results use too many cells and are not adequately refined, showing the need for a better representation (such as non-axis-aligned partitioning) for a problem like this.

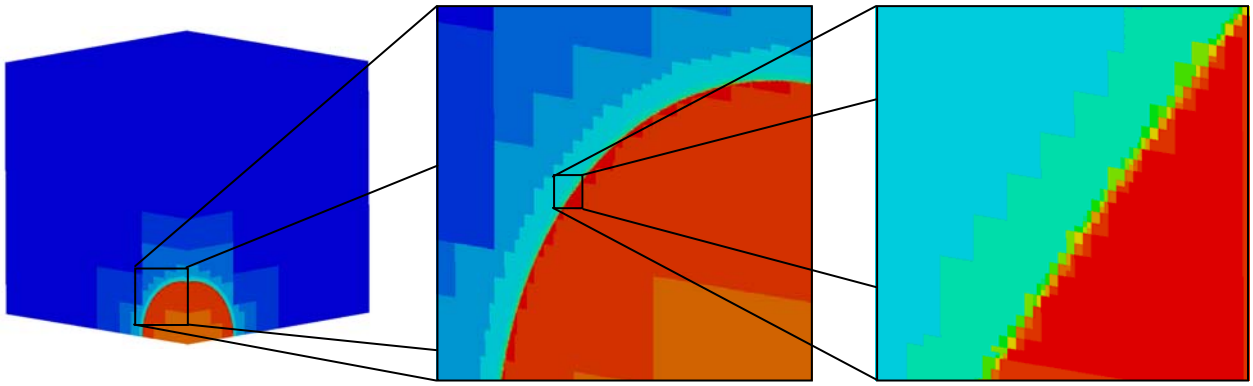

Figure 13. kd-tree adjoint-flux representation for a spherical-cavity test problem.

Using the same transport-derived adjoint-flux data, but interpreting radius as the maximum Cartesian coordinate so as to transform the problem into a rectangular cavity, we again generated kd-trees of the flux map. This tree, with results for the lowest energy electron group, are shown in Figure 14. In this case, the long aspect ratios of the kd-tree were adequate to represent the data. Even requiring flux ratios of less than factors of 3 in the 20 group problem resulted in a tree of just over 80,000 leaves.

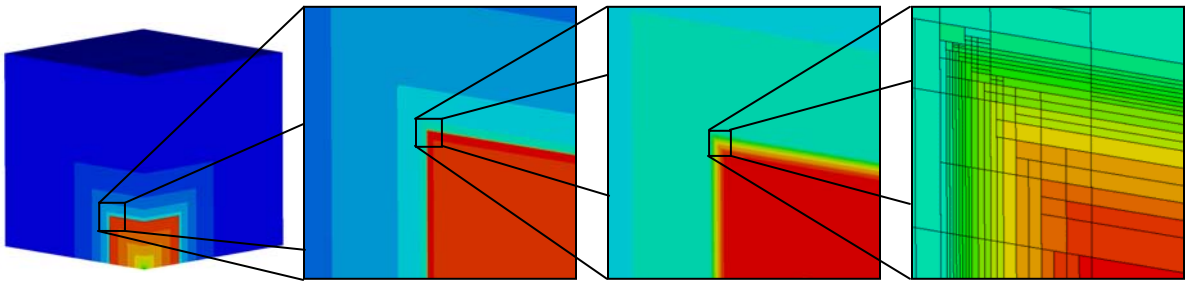

Figure 14. kd-tree adjoint-flux representation for a rectangular-cavity test problem.

From these results, one can conclude that the neighbor-comparison algorithm is converging to a reasonable representation of a flux map even in the presence of statistical uncertainty, but one cannot conclude that the neighbor-comparison algorithm is functioning adequately for our purposes. The important point for setting weight windows is that the particle weight should be maintained in inverse proportion to the particle importance. The relevant metric is a point-wise measure of the error in the importance map. The volume-averaging within tree leaves can easily mask important variations. Further, the neighbor-comparison algorithm has no mechanisms for identifying large variations that occur entirely within a leaf.

ii. Functional-Variation Algorithms

We made some preliminary investigations of using the functional variation of flux within leaves to determine when and where to impose space-partitioning cuts. We calculated low-order Legendre moments of flux (only up to 2^{nd} order) and did not calculate cross moments.

First, we examined using only the first moment as a replacement for the neighbor comparison. This slope information is indicative of how much flux varies across the cell. This approach was found to be less reliable than the neighbor comparison. If flux varies sharply (especially, downward) near the cell boundary, the volume-averaged slope is unlikely to indicate this, but a neighbor comparison is more likely to indicate this.

The moments (up to the second moment, and neglecting cross moments) were used to formulate a quadratic representation of the flux in each coordinate direction. The maximum and minimum of this representation within the cell could be used as a criterion of whether there was excessive flux variation within the cell, such that it needed to be cut. The quadratic representation could also be used to select a location at which to make the cut. The first implementation assumed that the cut should be made at the logarithmic average of the maximum and minimum flux values. Unfortunately, the quadratic representation is generally much less accurate for the minimum value than for the maximum value. Placing a cut at some level as fraction-of-flux down from the maximum seemed to be a more robust approach.

The functional form by itself did not always succeed in indicating when a cell needed to be split. This seemed to be especially problematic in a cell with a predominately high, uniform level of flux but which included a small region that in which the flux level decreased drastically. Due to the volume-averaging involved in calculating the moments, the functional form did not adequately represent the flux fall-off near the edge of the cell. We concluded that higher-order moments would be required. Indeed, subsequent investigations of p- and hp-adaptivity algorithms (described in Sections IV.B and IV.C) include not only higher-order representations, but also error metrics that can successfully indicate how well the functional-expansion representation is converging to the true distribution. Using such function-expansion data for a purely h-adaptive algorithm was not revisited.

iii. Bin-Based Algorithms

Binned (or histogram) data is an alternative to using a functional representation of the importance variation within a leaf. In our bin-based approach, a leaf was divided into a user-specified number of bins in each coordinate direction. Unlike a grid, each point within a leaf was contained within three bins, one in each coordinate direction. In other words, a bin in one coordinate axis extended fully across the leaf in the other two coordinate directions. This representation could accurately capture one-dimensional variation of importance. In favorable comparison to the functional representation, a stark decline in the flux level at the edge of the leaf could be detected (provided that a sufficient number of bins were used so that volume-averaging within the bin did not mask the variation). Due to the larger amount of information, multiple leaf divisions could be determined using the data contained in a single leaf. It was found necessary to use the neighbor comparison method in addition to this method, so that flux variations between leaves were detected also.

iv. Grid-Based Algorithms

Due to time and effort constraints on the project, the final milestone of the project, a grid-based algorithm bsp-tree algorithm, has not yet been successfully demonstrated for pure h-adaptivity (although Section IV.C contains a discussion of a grid-based approach to performing hp-adaptivity). This approach is very similar to the bin-based approach. The grid requires more data but is capable of resolving more complicated flux variations through the cell. The intent of pursuing the grid-based algorithm is that with additional information to draw upon, the algorithm may be able to guide the placement of cell divisions in non-axis-aligned positions. As such, the adaptivity algorithm will be able to build a more arbitrary bsp-tree (not restricted to the axis-aligned octrees and kd-trees previously discussed). In theory, the arbitrary bsp-tree will be able to more efficiently represent importance maps with non-axis-aligned variations, like the one shown in Figure 13.

B. p-Adaptivity

The advantages of Monte Carlo codes are in their ease-of-use and high degree of accuracy. However, they are limited by the accuracy of the histogram-style tally structures of the Monte Carlo code output. The alternative of functional expansion tallies (FET) has been proposed**Error! Reference source not found.Error! Reference source not found.**. For an FET, the Monte Carlo code estimates orthogonal-polynomial moments of the distribution of a quantity such as flux. It has been shown that unbiased estimators of these moments are possible using collision or track-length tallies**Error! Reference source not found.**. The accuracy of FET structures has been compared favorably against the accuracy of the histogram tally structure by Griesheimer et al**Error! Reference source not found.**. The FET error can be estimated for an existing tally structure and even predicted for an alternative structure. Thus, Monte Carlo codes can be made to not only provide users with an estimated error of representation for results but also iteratively refine the tally structure to achieve a user-specified error level. In this Section, we extend the FET error estimation of Griesheimer et al.**Error! Reference source not found.** to three-dimensional distributions and demonstrate such an adaptive approach.

i. Functional-Expansion Tallies

We use Legendre polynomials to form a three-dimensional functional expansion of a distribution:

$$f(x,y,z) = \sum_{n=0}^{\infty}\sum_{m=0}^{\infty}\sum_{l=0}^{\infty} b_{n,m,l} k_n k_m k_l P_n(x) P_m(y) P_l(z)\,, \tag{1}$$

where the normalization constants k_n, k_m, and k_l are defined as

$$k_n = \frac{2n+1}{2} \tag{2}$$

and the coefficients are defined as

$$b_{n,m,l} = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} f(x,y,z) P_n(x) P_m(y) P_l(z)\; dx\; dy\; dz\,. \tag{3}$$

In order to work with normalized functions, so that our error metrics are independent of the magnitude of the distribution, we define normalized coefficients: $a_{n,m,l} = b_{n,m,l} / b_{0,0,0}$. Also for convenience in our analysis, we organize the terms by their order and perform an equivalent pair of summations of the expansion coefficients:

$$f(x,y,z) = b_{0,0,0} \sum_{j=0}^{\infty} \sum_{\left\{ \substack{n,m,l \mid \\ n+m+l=j} \right\}} a_{n,m,l} k_n k_m k_l \, P_n(x) P_m(y) P_l(z). \tag{4}$$

In any practical application, we truncate the expansion to a finite order, *J*. We denote the truncated functional expansion as $\tilde{f}(x,y,z)$. We also estimate the expansion coefficients by estimating them in a Monte Carlo evaluation of Eq. (3). We denote the estimated coefficients as $\hat{a}_{n,m,l}$ with an estimated standard deviation of $\hat{\sigma}_{\hat{a}_{n,m,l}}$ yielding an approximate function $\hat{f}(x,y,z)$.

a. Functional-Expansion Tally Error

There are two sources of error in our approximation of the true distribution: truncation error in our expansion and statistical error in our Monte Carlo estimation of the coefficients. Using the functional expansions and the orthogonality of the Legendre polynomials, we find the 2-norm error of truncation is:

$$\begin{aligned} \left\| \tilde{E}_J \right\| &= \sqrt{\frac{1}{8} \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} \left(\frac{f(x,y,z)}{b_{0,0,0}/8} - \frac{\tilde{f}(x,y,z)}{b_{0,0,0}/8} \right)^2 dx\, dy\, dz} \\ &= \sqrt{8 \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} \left(\sum_{j=J+1}^{\infty} \sum_{\left\{ \substack{n,m,l \mid \\ n+m+l=j} \right\}} a_{n,m,l} k_n k_m k_l P_n(x) P_m(y) P_l(z) \right)^2 dx\, dy\, dz} \\ &= \sqrt{8 \sum_{j=J+1}^{\infty} \sum_{\left\{ \substack{n,m,l \mid \\ n+m+l=j} \right\}} a_{n,m,l}^2 k_n k_m k_l} \end{aligned} \tag{5}$$

Factors of 1/8 are due to volume averaging, and $b_{0,0,0}/8$ is the volume averaged value of the function $f(x,y,z)$. Similarly, we find the 2-norm error due to statistics is:

$$\begin{aligned} \left\| \hat{E}_J \right\| &= \sqrt{\frac{1}{8} \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} \left(\frac{\tilde{f}(x,y,z)}{b_{0,0,0}/8} - \frac{\hat{f}(x,y,z)}{\hat{b}_{0,0,0}/8} \right)^2 dx\, dy\, dz} \\ &= \sqrt{8 \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} \left(\sum_{j=0}^{J} \sum_{\left\{ \substack{n,m,l \mid \\ n+m+l=j} \right\}} \left(a_{n,m,l} - \hat{a}_{n,m,l} \right) k_n k_m k_l P_n(x) P_m(y) P_l(z) \right)^2 dx\, dy\, dz} \\ &= \sqrt{8 \sum_{j=0}^{J} \sum_{\left\{ \substack{n,m,l \mid \\ n+m+l=j} \right\}} \left(a_{n,m,l} - \hat{a}_{n,m,l} \right)^2 k_n k_m k_l} \end{aligned} \tag{6}$$

The test function used by Griesheimer et al. was

$$g(x) = \begin{cases} \cos(x)\exp(2x+1), & -1.0 \le x < -0.5 \\ \cos(x), & -0.5 < x < 0.5 \\ \cos(x)\exp\left[-(2x+1)/4\right], & 0.5 < x \le 1.0 \end{cases} . \quad (7)$$

For our test function, we simply extend this to a three-dimensional distribution as

$$f(x,y,z) = g(x)g(y)g(z) . \quad (8)$$

The distribution is C^0 continuous but contains planar surfaces of C^1 discontinuities (i.e., discontinuities in the derivative) at -0.5 and 0.5 in each of the coordinate axes. We used 100,000 samples from this distribution to perform Monte Carlo integration of the coefficients defined by Eq. (3). It has been shown that this provides an unbiased estimate of the coefficients**Error! Reference source not found.**. The true truncation and statistical errors as a function of expansion order are shown in Figure 15. The lines on the plot are power laws. We note the decrease in truncation error and increase in statistical error with increasing order appear well-characterized by these lines. This is expected for distributions with a finite algebraic index of convergence**Error! Reference source not found.Error! Reference source not found.**. Analytic distributions, which have infinite algebraic indices of convergence, are expected to converge exponentially fast.

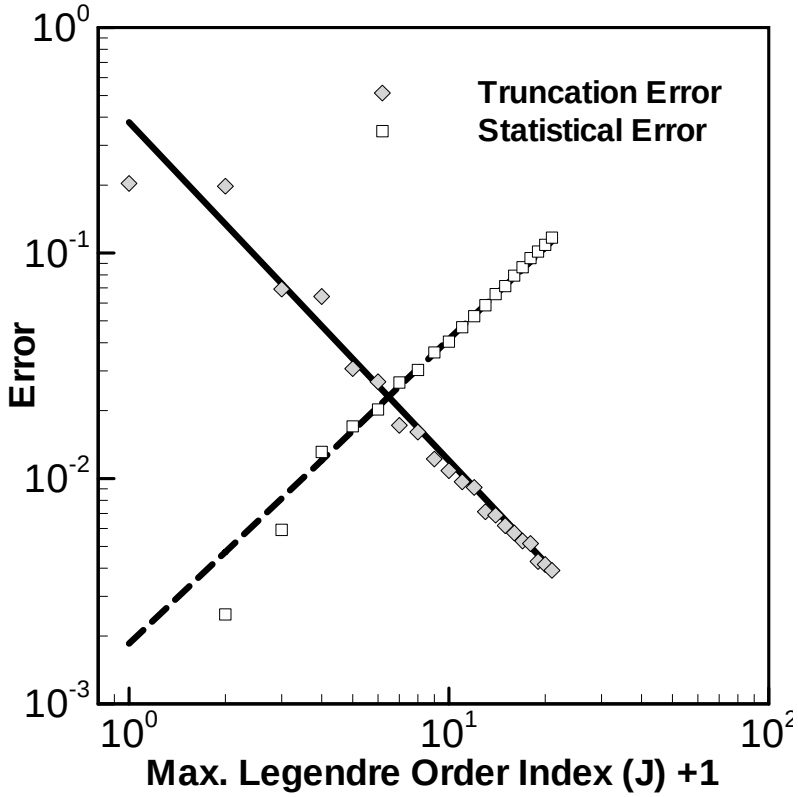

Figure 15. The true truncation and statistical error as a function of expansion order, using coefficients generated with 100,000 samples of the test function.

b. Functional-Expansion Tally Error Estimation

In practice, the true distribution is unknown, and the true coefficients, $a_{n,m,l}$, used in Eqs. (5) and (6) are unknown. Thus, we must estimate the truncation and statistical error. We do this by following Griesheimer et al.**Error! Reference source not found.** in approximating the above equations. Recognizing the square of the difference between the true and estimated coefficients as the variance and using our estimate of that quantity, we approximate Eq. (6) as:

$$\left\|\hat{E}_J\right\|^2 \approx 8\sum_{j=0}^{J}\sum_{\left\{\begin{smallmatrix} n,m,l \mid \\ n+m+l=j \end{smallmatrix}\right\}} \hat{\sigma}^2_{\hat{a}_{n,m,l}} k_n k_m k_l \,. \tag{9}$$

We approximate Eq. (5) by using the estimated coefficients in a finite expansion:

$$\left\|\tilde{E}_J\right\|^2 \approx 8\sum_{j=J+1}^{I}\sum_{\left\{\begin{smallmatrix} n,m,l \mid \\ n+m+l=j \end{smallmatrix}\right\}} \hat{a}^2_{n,m,l} k_n k_m k_l \,. \tag{10}$$

We want to limit this expansion not just to the number of estimated coefficients but to only the statistically valid coefficients. To do this, Griesheimer et al.**Error! Reference source not found.** proposed a cost-to-benefit ratio by recognizing the benefit of reducing truncation error at the cost of increasing statistical error. In our 3D expansion, the cost-to-benefit ratio is estimated as

$$R_j^2 = \frac{\sum_{\left\{\begin{smallmatrix} n,m,l \mid \\ n+m+l=j \end{smallmatrix}\right\}} \hat{\sigma}^2_{\hat{a}_{n,m,l}} k_n k_m k_l}{\sum_{\left\{\begin{smallmatrix} n,m,l \mid \\ n+m+l=j \end{smallmatrix}\right\}} \hat{a}^2_{n,m,l} k_n k_m k_l} \,. \tag{11}$$

This suggests that the expansion limit, I, for estimating truncation error is the highest order, j, for which $R_j < 1$. Since both $\hat{a}_{n,m,l}$ and $\hat{\sigma}_{\hat{a}_{n,m,l}}$ are estimated values and those estimations degrade for large uncertainty, we find that a slightly more stringent criteria must be applied. Our experience indicates that using $I=j$ for the largest j such that $R_j < 0.7$ generally provides an appropriate truncation order.

The values of R_j for our test function are shown in Figure 16. Using our threshold criteria of $R_j < 0.7$, we conclude that, with the existing statistical uncertainties, the distribution can be best approximated using a 6th order expansion. We use Eq. (10) with I=6 to estimate truncation error for orders J=0 to 5 and Eq. (9) to estimate statistical error at all orders for which coefficients were estimated. The estimated truncation and statistical errors are compared with the true errors in Figure 17. The truncation error tends to be underestimated due to the neglect of higher-order terms (greater than order I). Using the slope of the truncation error, an estimated correction is possible, but this appears to be a minor approximation in our method. Our estimate of truncation error is also contaminated with statistical error, which we have not attempted to correct.

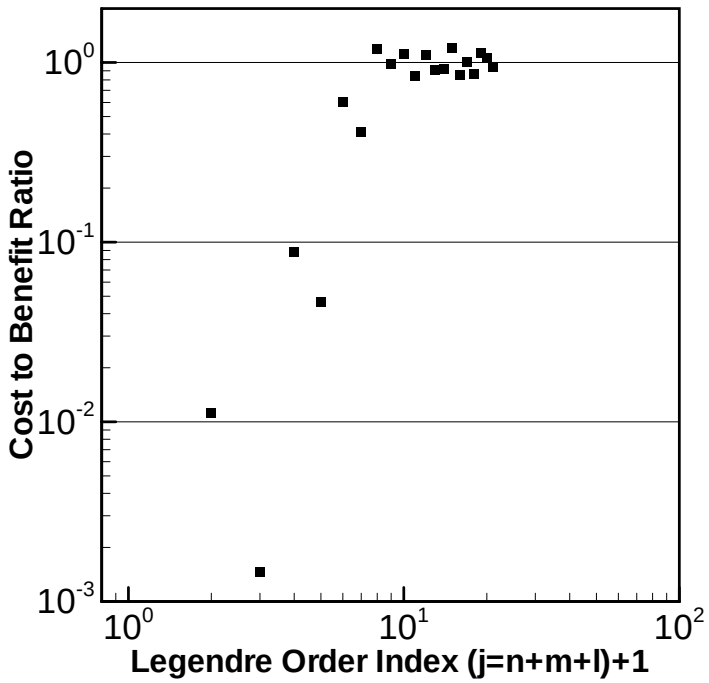

Figure 16. The cost-to-benefit ratio of expansion terms versus expansion order, using coefficients and variances generated with 100,000 samples of the test function.

ii. p-Adaptivity Algorithm

The fit to the estimated truncation error in Figure 17 illustrates that we can reasonably predict the truncation error for higher-order expansions. We can use that estimation to iteratively refine the tally structure in a Monte Carlo calculation. Some iteration is desirable, since the initial estimations may be based on less accurate low-order data with more marginal statistical validity. This iteration can proceed until we achieve an expansion order that sufficiently reduces truncation error. Likewise, since statistical error will decrease at a rate inversely proportional to the square root of Monte Carlo runtime, we can estimate the runtime required to sufficiently reduce the statistical error. The combination of truncation error and statistical error can be balanced to meet a user-specified total error threshold, but how these are balanced may depend to some degree upon the value of computation time and the memory required for the expansion coefficients. In practice, adaptively decreasing the order of representation is not strictly necessary, but it can reduce memory and runtime requirements by decreasing the number of tallies required of the Monte Carlo code.

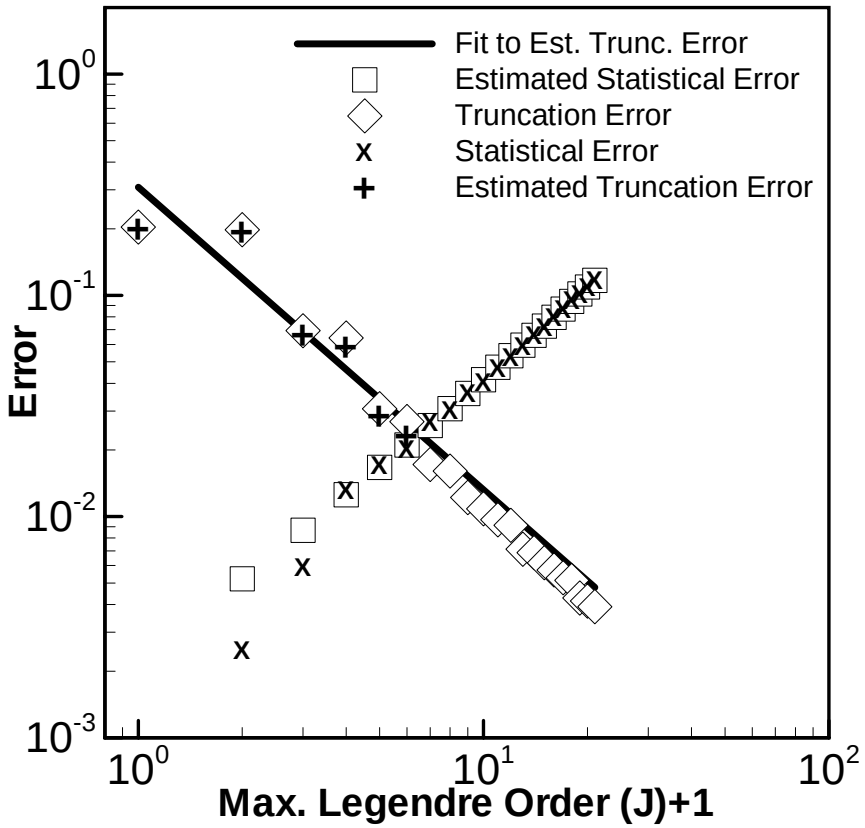

Figure 17. Comparison of the estimated truncation and statistical error with the true errors, using estimated coefficients generated with 100,000 samples of the test function.

iii. p-Adaptivity Results

To test the adaptivity algorithm, the sample problem was divided into seven cells in each axis, and the 4x4x4 set of cells in the corner of the problem with the minimum coordinates was analyzed. Each cell was initialized to examine a 6^{th} order representation of the function and to start with a number of Monte Carlo samples that turned out to be at least an order of magnitude less than required for converged results. The algorithm was requested to achieve a volume-averaged 2-norm error of 0.002. The order of the representation and the number of Monte Carlo samples were iterated upon until results were converged. The cells of this problem containing C^1 discontinuities require higher-order representations to resolve. In cells containing one of the discontinuities, the converged representation is 7^{th} or 8^{th} order. In cells containing two discontinuities, the representation is 10^{th} order. The cell containing the intersection of all 3 of these discontinuity surfaces required a 12^{th} order representation, the highest in the problem. Figure 18 shows the true distribution across a diagonal of these results, from the coordinate (-1,-1,-1) through (0,0,0). The functional-expansion distribution would be indistinguishable from the true distribution if shown on this same plot. Figure 18 also shows the relative error in the functional expansion along the same diagonal.

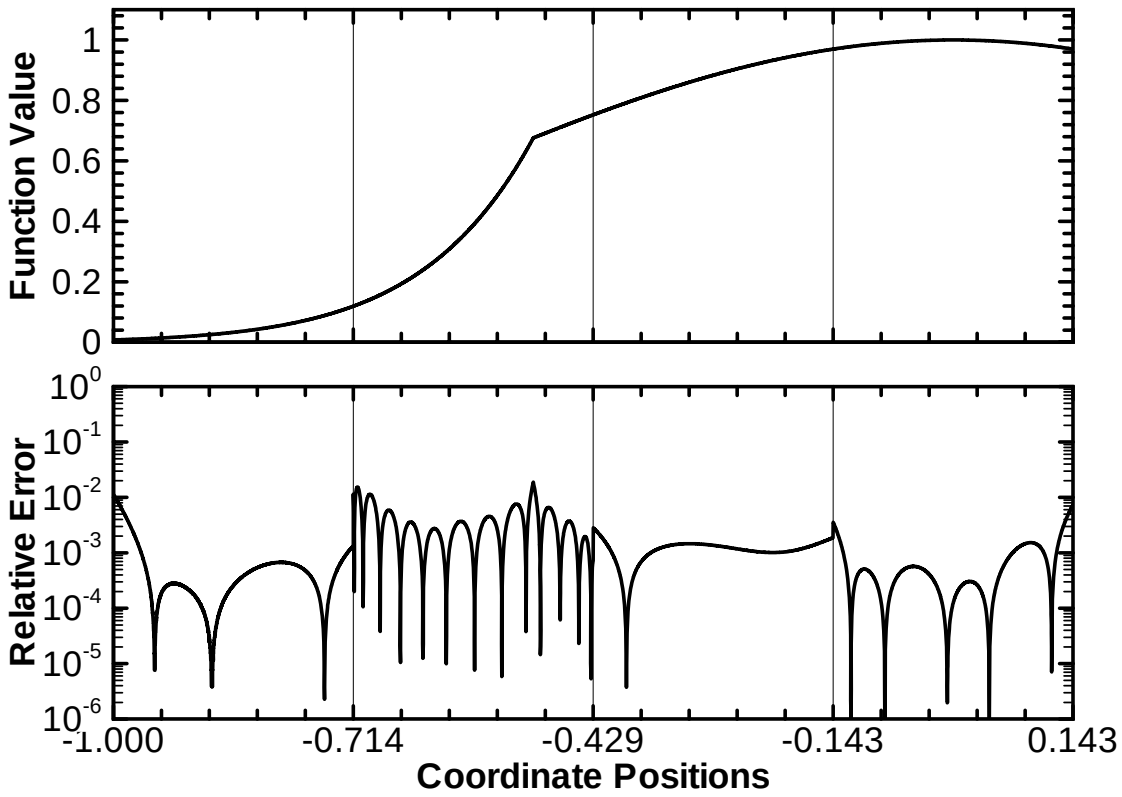

Figure 18. The upper portion of the plot shows the test function distribution along a diagonal of the test problem. The lower portion of the plot shows the relative error in the functional expansion, $\left|1-\hat{f}/f\right|$. The functional expansion was independently adapted in each cell via Monte Carlo estimations.

Assessing the true volume-averaged 2-norm error, the test problem results are generally converged to levels near or better than the requested error. In the low-order expansions, the error is lower than the requested error and often significantly so. In the absence of a discontinuity in the true distribution, the error in the functional expansion can be expected to decrease exponentially fast. Thus, the algorithm tends to over-predict the expected error level during refinement.

In the high-order expansions in this test problem, the true error is always slightly higher than the requested error. In cells containing a discontinuity of the true distribution (i.e., cells with a 7^{th} order expansion or higher), the errors range from 2.42E-03 to 3.09E-03, with the highest-order expansions producing errors at the lower end of the range. In this problem, the discontinuity in the test-function distribution results in a shape of the truncation error distribution for which our fit is not conservative for the selected expansion orders. Although these true errors are close to the desired error level and more conservative predictive algorithms are possible, this illustrates the potential pitfalls inherent in the approximations employed.

iv. p-Adaptivity Conclusions

We have described and demonstrated an algorithm for adaptively refining three-dimensional functional expansion tallies to meet a user-specified error-of-representation threshold. This capability builds on previous development by Griesheimer et al. of error estimation techniques for functional-expansion tallies. This technique may be employed to make Monte Carlo codes still more user-friendly.

Many additional developments seem possible and desirable. Our algorithm refines only according to our definition of expansion order, which imposes a particular constraint on the orders of the phase-space components. An improvement would be to determine a proper expansion order for each component, with a different constraint, possibly allowing more degrees of freedom while increasing complexity. If the solution can be adequately represented by a one- or two-dimensional expansion, the memory savings on a high-order expansion is significant. We have offered no guidance on an optimum number of cells for functional expansion adaptivity. The algorithm would perform more efficiently if the user placed cell divisions at material boundaries where solution C^1 discontinuities are expected in the flux. Ideally, an algorithm would adaptively determine the cell distribution as well as the functional expansion within each cell. We present such an hp-adaptivity algorithm in the following Section.

C. hp-Adaptivity

Adaptivity based on a posteriori error-estimation techniques has been extensively developed for finite-element methods, to automatically reduce solution error and improve code usability**Error! Reference source not found.**. Adaptivity of tallies in Monte Carlo codes could provide similar improvements. Because the Monte Carlo transport of particles does not depend on the result tallies, adaptivity may be more robust than in finite-element methods, in which the results of one element affect the solution of the adjacent elements. This may make it possible to be more aggressive in the adaptive methods. Aggressive adaptivity is also desirable because of the greater expense of Monte Carlo calculations. Our method adaptively coarsens or refines both the spatial binning and the functional-expansion tallies within each spatial bin. We spatially divide the problem domain using a binary space-partitioning tree using arbitrarily positioned axis-aligned cutting planes. We use functional-expansion tallies with Legendre polynomial basis functions.

Functional-expansion tallies (FET) were first demonstrated to be more effective than binned tallies for some problems in the mid-1970's**Error! Reference source not found.Error! Reference source not found.**. Recently, Griesheimer et al. explored FET in Monte Carlo calculations and compared the sources of error in FET and uniform binning refinement**Error! Reference source not found.**. In previous work, we have extended FET error estimation to three-dimensional tallies and demonstrated an adaptivity approach**Error! Reference source not found.**. Here, we investigate methods for including spatial adaptivity to achieve more efficient solution representations.

Adaptive Monte Carlo tallies were used by Booth in an effort to automate phase-space partitioning for arriving at a sufficiently detailed importance map**Error! Reference source not found.**. That work used no tally moments and was entirely "h-adaptive." Adaptive functional tallies were used by Lai and Spanier for importance sampling to demonstrate geometric convergence of biasing in Monte Carlo**Error! Reference source not found.**. Booth further demonstrated that adaptive importance sampling could achieve geometric convergence by using both functional-expansion and spatial-partitioning adaptivity**Error! Reference source not found.**. This work differs from those two previous efforts in several ways. The prior "adaptive" techniques calculated a fixed number of Monte Carlo tally moments and adapted the number of moments used to bias the forward calculation based on whether the moment tallies had achieved a low statistical uncertainty. We adapt the number of Monte Carlo tally moments calculated based on error estimation and aim to achieve a specified degree of accuracy everywhere in the global solution representation. Booth**Error! Reference source not found.** achieved spatial adaptivity by examining the effectiveness of the forward biasing and comparing point-wise estimates of importance against the functional-expansion representation. We achieve spatial adaptivity by determining regions of the problem where the functional expansion either converges too slowly or can be more efficient when spatial refinement is applied.

In Section IV.C.i, we present the error estimation technique that is used to drive the adaptivity algorithm. As discussed in Section IV.C.ii, the functional-expansion error estimation forms the basis for both the h-adaptivity and p-adaptivity aspects of the algorithm. That section presents

the details of our hp-adaptivity method and discusses what we perceive to be the key remaining deficiencies. In Section IV.C.iii, we present some detailed results from using the hp-adaptivity method on a test problem. In Section IV.C.iv, we compare the hp-adaptivity algorithm against several alternative approaches in terms of both the (reduced) memory usage and the (increased) computational expense. In Section IV.C.v, we offer a few concluding remarks on hp-adaptivity.

i. Function-Expansion-Tally (FET) Error Metrics

The error metrics presented here are an extension of the one-dimensional error metrics derived by Griesheimer et al.**Error! Reference source not found.**. We use Legendre polynomials to form a three-dimensional functional expansion of a distribution:

$$f(x,y,z) = \sum_{n=0}^{\infty}\sum_{m=0}^{\infty}\sum_{l=0}^{\infty} b_{n,m,l} k_n k_m k_l P_n(x) P_m(y) P_l(z)\,, \tag{12}$$

where the normalization constants k_n, k_m, and k_l are defined as

$$k_n = \frac{2n+1}{2} \tag{13}$$

and the coefficients are defined as

$$b_{n,m,l} = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} f(x,y,z) P_n(x) P_m(y) P_l(z)\; dx\; dy\; dz\,. \tag{14}$$

In order to work with normalized functions, so that our error metrics are independent of the magnitude of the distribution, we define normalized coefficients: $a_{n,m,l} = b_{n,m,l} / b_{0,0,0}$. Also for convenience in our analysis, we organize the terms by their order and perform an equivalent pair of summations of the expansion coefficients:

$$f(x,y,z) = b_{0,0,0}\sum_{j=0}^{\infty}\sum_{\left\{\begin{smallmatrix} n,m,l \mid \\ n+m+l=j \end{smallmatrix}\right\}} a_{n,m,l} k_n k_m k_l\; P_n(x) P_m(y) P_l(z)\,. \tag{15}$$

In any practical application, we truncate the expansion to a finite order, J. We denote the truncated functional expansion as $\tilde{f}(x,y,z)$. We also estimate the expansion coefficients in a Monte Carlo evaluation of Eq. (14). We denote the estimated coefficients as $\hat{a}_{n,m,l}$ with an estimated standard deviation of $\hat{\sigma}_{\hat{a}_{n,m,l}}$ yielding an approximate function $\hat{f}(x,y,z)$.

There are two sources of error in our approximation of the true distribution: truncation error in our expansion and statistical error in our Monte Carlo estimation of the coefficients. Using the functional expansions and the orthogonality of the Legendre polynomials, we find the volume-averaged 2-norm error of truncation is:

$$\left\|\tilde{E}_J\right\| = \sqrt{\frac{1}{8}\int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1}\left(\frac{f(x,y,z)}{b_{0,0,0}/8} - \frac{\tilde{f}(x,y,z)}{b_{0,0,0}/8}\right)^2 dx\,dy\,dz} = \sqrt{8\sum_{j=J+1}^{\infty}\sum_{\left\{\begin{smallmatrix} n,m,l \mid \\ n+m+l=j \end{smallmatrix}\right\}} a_{n,m,l}^2 k_n k_m k_l}\,, \tag{16}$$

where J is the order at which the expansion is truncated. Factors of 1/8 are due to volume averaging, and $b_{0,0,0}/8$ is the volume averaged value of the function $f(x,y,z)$. Similarly, we find the volume-averaged 2-norm error due to statistics is:

$$\left\|\hat{E}_J\right\| = \sqrt{\frac{1}{8}\int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1}\left(\frac{\tilde{f}(x,y,z)}{b_{0,0,0}/8} - \frac{\hat{f}(x,y,z)}{\hat{b}_{0,0,0}/8}\right)^2 dx\,dy\,dz} = \sqrt{8\sum_{j=0}^{J}\sum_{\left\{n,m,l \mid n+m+l=j\right\}}\left(a_{n,m,l} - \hat{a}_{n,m,l}\right)^2 k_n k_m k_l} \tag{17}$$

Following Griesheimer et al.**Error! Reference source not found.**, we approximate these as:

$$\left\|\tilde{\tilde{E}}_J\right\|^2 \approx 8\sum_{j=J+1}^{I}\sum_{\left\{n,m,l \mid n+m+l=j\right\}}\hat{a}_{n,m,l}^2 k_n k_m k_l\ , \tag{18}$$

where *I* is the maximum order for which data has been computed, and

$$\left\|\hat{E}_J\right\|^2 \approx 8\sum_{j=0}^{J}\sum_{\left\{n,m,l \mid n+m+l=j\right\}}\hat{\sigma}_{\hat{a}_{n,m,l}}^2 k_n k_m k_l\ . \tag{19}$$

The subject of using this data to determine the required expansion order and run time to achieve a specified error level is dealt with in greater detail in Ref. **Error! Reference source not found.** and Section IV.B.ii, but we will summarize the method here. If there is not sufficient statistically valid data, an estimate is made of the additional run time required. Provided enough statistically valid data, we can estimate the truncation error as a function of expansion order. For distributions with a finite algebraic index of convergence, we expect the truncation error to follow a power law. Thus, using a power-law fit to the low-order truncation error data, we can predict the order necessary to achieve a specified truncation error. Since statistical error increases in a predictable relation to expansion order (as the square root of order) and decreases as a function of Monte Carlo run time (inversely as the square root of the number of histories), we can estimate the run time required to a desired statistical error level at the necessary truncation order. In the next section we discuss how this estimate of a required expansion order (and run time) can be used to find an optimal spatial division to minimize the memory requirements for achieving the specified error level.

ii. FET-Based hp-Adaptivity

The h-adaptivity of our algorithm is built on a binary space-partitioning tree. The p-adaptivity of our algorithm is built on the functional-expansion moments. Adaptivity in the functional expansion is driven by the error metrics described in the previous section. It would be highly desirable to have similar error metrics to guide adaptivity in spatial partitioning. Such metrics allow predictive extrapolation and enable more aggressive adaptation. As far as we know such information has only been derived within the constraints of a uniform-binning approach**Error! Reference source not found.**. Instead, we have adopted a trial-and-error approach to comparing the spatial-partitioning error for limited numbers of possible tree cuts. This is less aggressive but more memory efficient.

For simplicity, we constrain the tree cut planes to be axis aligned, which results in a “kd-tree.” We require the root of the tree to be an axis-aligned box, and since all cut planes

are axis aligned, all nodes of the tree are also axis-aligned boxes. Because we use a binary tree, each node containing a cut plane is divided into two nodes. A node that is not cut is a "leaf" of the tree. Because we use a spatial-partitioning tree, each point of space within the root box is uniquely contained within a single leaf of the tree.

To obtain sufficient data on which to base our partitioning decisions, we calculate distribution data as discontinuous functional expansions on a regular grid within each tree leaf. Each regular grid is defined as an N×N×N division of the leaf's box into spatial "cells." A 4th order functional expansion (or higher order specified by the user) is calculated within each cell of the grid.

A Monte Carlo tally at any point in space is only performed for a single grid cell functional expansion. In the adaptivity algorithm, when we wish to evaluate the functional-expansion error-convergence rate for a prospective spatial division, we perform three-dimensional quadrature integration**Error! Reference source not found.** across the functional data stored on grid cells. Thus, these alternative spatial divsions may arbitrarily encompass any portion of one or more grid cells and may span one or more leaves.

The goal of our adaptivity algorithm is to minimize the number of variables required to achieve a user-specified level for a volume-averaged 2-norm error criterion. To achieve this we calculate Monte Carlo estimates of functional-expansion coefficients in each cell of the regular grid on each leaf of the tree. The tree may begin as a single leaf. Based on the grid data, there are four ways in which the tree may be adapted:

1. We will attempt to isolate any grid cell where the estimated order-of-representation required to meet the error criterion is greater than a maximum limit. This maximum limit is used due to the difficulty of accurately projecting error to very high orders (short of calculating moments to very high orders) and the dubious comparisons of such extrapolations that would result. The algorithm for doing this attempts to identify tree cutting planes along grid boundaries that will most effectively partition space to isolate regions requiring high-order representations.

2. We estimate the required order-of-representation for the unmodified leaf (by performing numerical integration across the Monte-Carlo-calculated grid-cell data and determining an error-convergence rate) and compare the number of degrees of freedom against the results for a selection of possible tree cuts (by similar numerical integrations and analysis). We choose the representation that yields the minimum degrees of freedom from this set of alternatives. If the order-of-representation required for the unmodified leaf is greater than the maximum limit, the leaf will be divided, regardless of the outcome of the degree-of-freedom comparisons.

3. All tree cuts are re-evaluated to determine whether eliminating cuts will reduce the degrees of freedom required to meet the user-specified error criterion (while not exceeding the maximum order-of-representation limit). This re-evaluation is done by performing numerical integration over the data stored in the tree leaves encompassed by each higher level node in the tree. The degrees of freedom

required for the child nodes are compared with the requirements of the parent node at each level of the tree to decide whether a tree cut should be eliminated.

4. If a tree leaf is not being modified, the functional-expansion order is adjusted to the estimated optimum.

Continuing to use the same grid-cell data, steps 1 and 2 of this process may be applied repeatedly to further partition space. When the process of refining and coarsening the spatial partitioning and functional expansions has completed, another Monte Carlo calculation is performed to repopulate the tree with data. Iteration between tree refinement and Monte Carlo calculation continues until the user is satisfied that the results have converged.

There are many improvements that could be applied to our implementation to reduce the cost of the Monte Carlo calculations and to make the adaptivity algorithm more robust. First, the last step of the adaptivity algorithm has not been integrated into the process. That is, p-adaptivity of tree leaves is only being performed when h-adaptivity has been halted; only the estimation of p-adaptivity error is currently being performed while spatial partitioning cuts are being made. As will be seen later, this is why we allow some leaves to adapt to representation orders beyond the "maximum limit." There have been a few cases where this has resulted in a leaf that needed an excessively high-order representation and the existing algorithm was clearly deficient. Second, we have not attempted to integrate the adaptivity algorithm into the Monte Carlo calculation. This could be done in a variety of ways, with differing degrees of implementation cost and potential efficiency-improvement payoffs. The simplest improvement would be to carry over Monte Carlo result data. When a tree leaf is not modified, the functional-expansion tallies remain valid. And increasing the order of representation does not invalidate the low-order tallies. There is no need to reset the tallies, as long as the number of histories contributing to each tally is properly recorded. A more tightly integrated approach could embed the adaptivity within the Monte Carlo calculation to assess whether cell refinement is warranted after some fixed number of incremental tallies. Third, the algorithm for determining the optimal tree cut is currently only evaluating one possible cut at a time. More complicated algorithms could determine an optimal series of cuts. Fourth, the p-adaptivity method is constrained to adapting to a single order of representation (see Eq. (15)). A more efficient representation might be achieved in some cases by having a high order representation for one or two axial components of the functional expansion and a low-order expansion for the other component(s).

iii. hp-Adaptivity Results

The test function used by Griesheimer et al.**Error! Reference source not found.** was

$$g(x)=\begin{cases}\cos(x)\exp(2x+1), & -1.0\le x<-0.5\\ \cos(x), & -0.5<x<0.5\\ \cos(x)\exp\left[-(2x+1)/4\right], & 0.5<x\le 1.0\end{cases}. \qquad (20)$$

For our test function, we simply extend this to a three-dimensional distribution as

$$f(x, y, z) = g(x)g(y)g(z) \,. \tag{21}$$

The distribution is C^0 continuous but contains planar surfaces of C^1 discontinuities (i.e., discontinuities in the derivative) at -0.5 and 0.5 in each of the coordinate axes. We use samples from this distribution to perform Monte Carlo integration of the coefficients defined by Eq. (3). We restrict evaluation to the range (-1.0, 0.9) in all three axes in order to prevent the algorithm from being capable of exactly determining that tree cuts should be placed on the C^1 discontinuities. A cut-away of the distribution is shown in Figure 19.

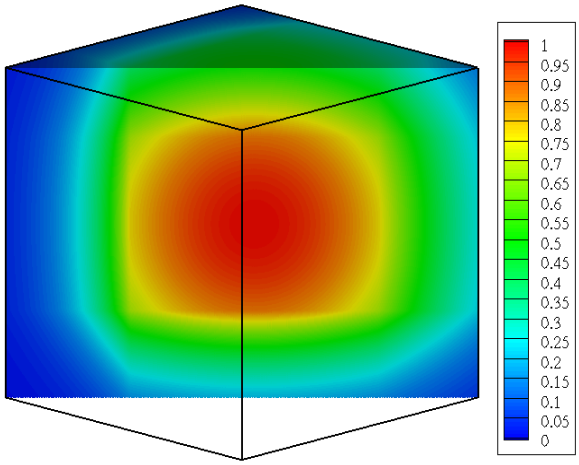

Figure 19. The test-function distribution.

Two sets of results were generated by allowing the adaptivity algorithm to iterate on our test function until converged to a 0.005 and 0.001 volume-averaged 2-norm error criterion. The tree-cut boundaries and the functional-expansion orders for these two cases are shown in Figure 20. The maximum-order limit was 10 in this test. As previously noted, functional-expansion orders are allowed to settle to an expansion order beyond what was desired. We note that tree cuts were placed very near C^1 discontinuities or were placed repeatedly on either side in an attempt to isolate them. Many more cuts are required for the more stringent error criterion.

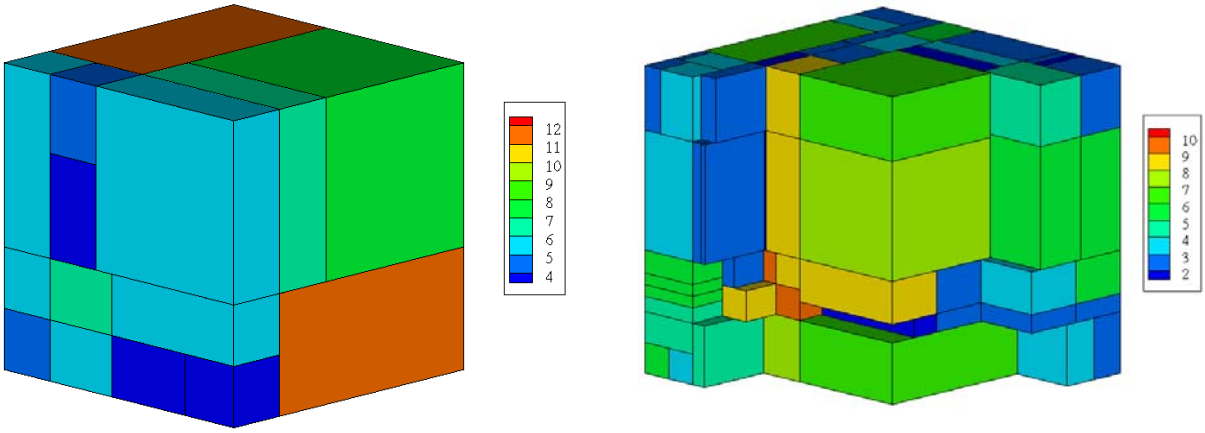

Figure 20. The spatial partitioning and functional-expansion order for converged results for a 0.005 volume-averaged 2-norm error criterion on the left and a 0.001 error criterion on the right.

Cut-aways of the point-wise relative error are shown in Figure 21. The error in the plot is measured point-wise, whereas the algorithm is based on a volume-averaged 2-norm error. The highest errors are near C^1 discontinuities and at corners of leaves. Throughout much of the problem the error is near the desired level.

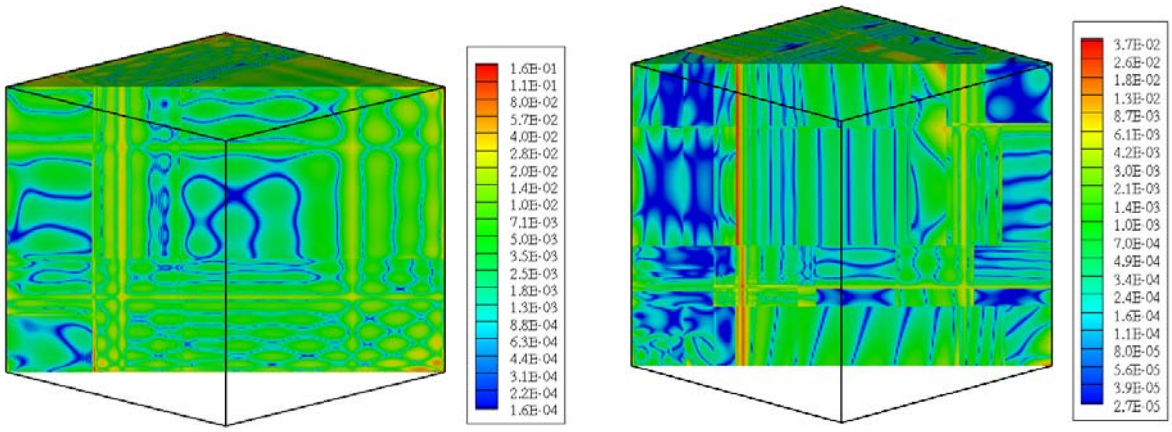

Figure 21. Absolute point-wise relative error of the converged functional distribution, $\left|1-\hat{f}/f\right|$, for a 0.005 volume-averaged 2-norm error criterion on the left and a 0.001 error criterion on the right.

iv. hp-Adaptivity Versus Alternative Approaches

The previous section presented some evidence that the hp-adaptivity method could be used to achieve a desired level of error. In this section we address how this approach compares with some alternatives. We present evidence that our approach generally uses memory as efficiently as (or more efficiently than) either h-adaptivity or p-adaptivity alone. Our objective was to decrease memory usage and improve ease-of-use, but

since this is accomplished by iterating on Monte Carlo calculations, we also present some estimates of the added computational expense. This may allow an assessment of whether the trade-off is acceptable for some applications. To limit the run-time requirements in performing the following studies to compare memory requirements, these results were acquired using deterministic methods (i.e., using three-dimensional quadrature integration) for all aspects of the analysis and neglecting the effects of statistical uncertainty that would be present with Monte Carlo calculations.

Here we consider three test functions. The first was described by Eqs. (20) and (21). The second test function is similar to the first, but with C^0 discontinuities included:

$$g(x) = \begin{cases} 9\cos(x)/10, & 0 \le |x| < 0.25 \\ \cos(x), & 0.25 \le |x| < 0.5 \\ \cos(x)\exp\left[-(2x+1)/4\right]/2, & 0.5 \le |x| \le 0.75 \\ 11\cos(x)\exp\left[-(2x+1)/4\right]/20, & 0.75 \le |x| \le 1.0 \end{cases}. \tag{22}$$

Again, we extend this to a three-dimensional distribution by Eq. (21). The third test function is:

$$g_1(x) = \begin{cases} 1+\cos(10\pi x)\exp(-2x-3.5), & abs(x) \ge 0.1 \\ 1+\cos(10\pi x)\exp(-2x-3.5) - 0.2\cos(5\pi x), & abs(x) < 0.1 \end{cases}$$
$$g_2(y) = 1+\cos(3\pi y)\exp(-2y-3.5), \tag{23}$$
$$g_3(z) = 0.75 + 0.25\cos(\pi z)$$
$$f(x,y,z) = g_1(x)g_2(y)g_3(z)$$

This contains only two C^1 discontinuities. These three test functions are illustrated in Figure 22.

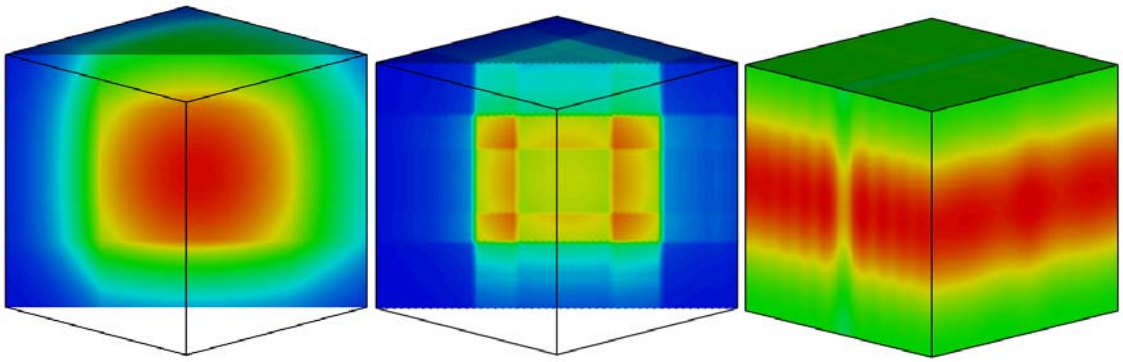

Figure 22. Test function distributions.

They were evaluated using four methods: the hp-adaptivity method previously described, uniform h-refinement, uniform p-refinement, and an h-adaptivity method. The h-adaptivity method was based on a simple algorithm of comparing neighboring leaves of a tree. If the ratio of the average values of two neighbors is greater than a threshold value, then both are divided in half parallel to the plane of their shared face. For each method, a parameter was varied that was expected to improve the accuracy, and the 2-norm of the relative error was recorded as function of the number of

distribution variables required for the representation. For uniform h-refinement, the parameter was the depth to which the binary tree was uniformly divided. For uniform p-refinement, the parameter was the functional-expansion order. For h-adaptivity, the parameter was the threshold value for determining whether neighboring leaves were to be divided. For hp-adaptivity, the parameter was the requested volume-averaged 2-norm error level. The plots of error versus degrees of freedom are shown in Figure 23, Figure 24, and Figure 25, for the first, second, and third test functions, respectively.

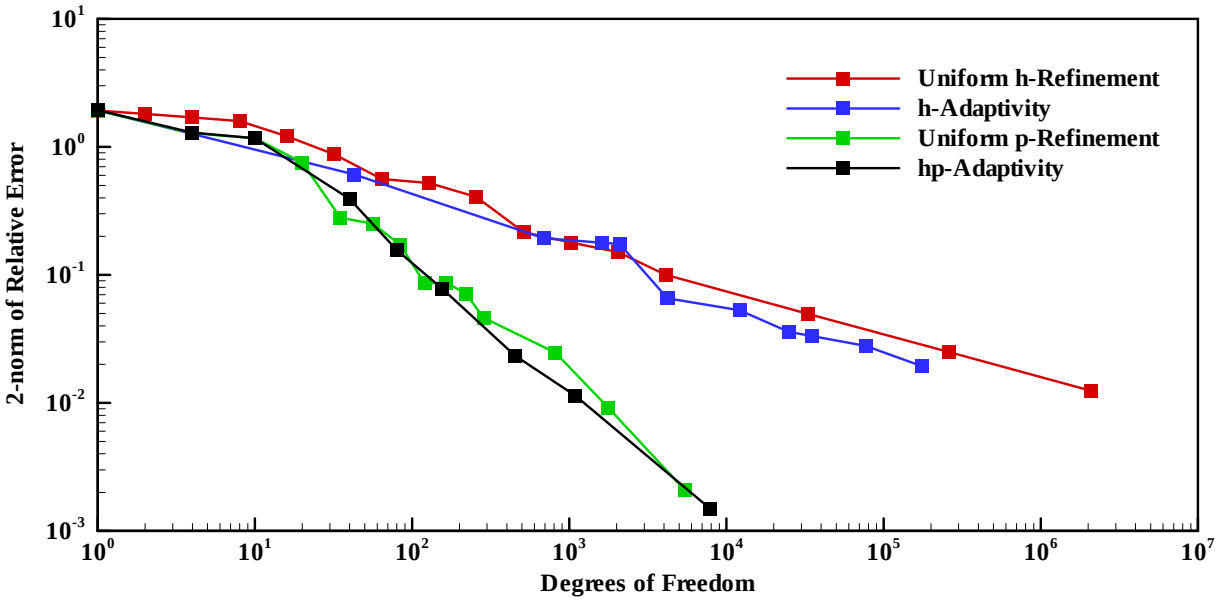

Figure 23. Error versus degrees of freedom for the first test function.

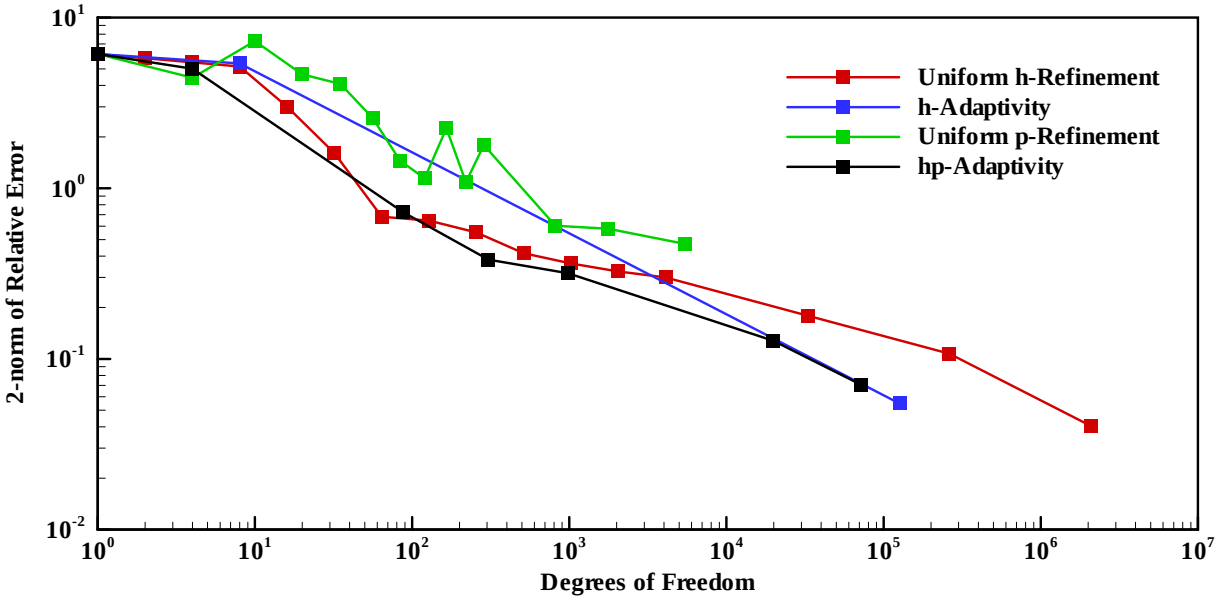

Figure 24. Error versus degrees of freedom for the second test function.

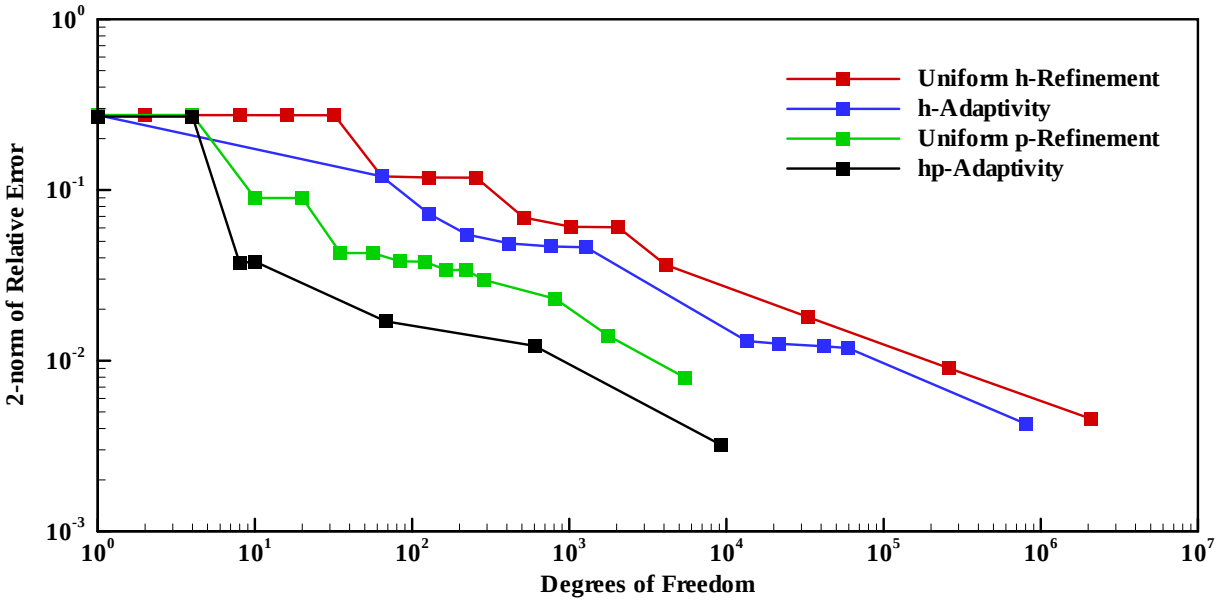

Figure 25. Error versus degrees of freedom for the third test function.

By these measures, hp-adaptivity is almost always performing as well or better than the best alternative. The h-adaptivity generally performs slightly better than the uniform h-refinement (but these test problems were not selected to highlight the difference between them). The h-adaptivity method employed was not appropriate for the second test problem, because it is not suited to problems with C^0 discontinuities; that is, when the discontinuity in the distribution exceeds the threshold for dividing neighboring leaves, the algorithm will make divisions that do not improve the representation and may attempt to make divisions interminably.

To estimate the run-time requirements for each method, we selected an error level and performed the Monte Carlo integrations to include the contributions of statistical error. To achieve a 2-norm error level of approximately 0.02 with the first test function, we found the run-time requirement for hp-adaptivity to be about 10,000 processor-seconds. This compared with an estimated run-time requirement for h-refinement of about 2900 processor-seconds and p-refinement of about 120 processor-seconds. For a 2-norm error level of approximately 0.5 with the second test function, we found the run-time requirement for hp-adaptivity to be about 440 processor-seconds. This compared with an estimated run-time requirement for h-refinement of about 0.4 processor-seconds and p-refinement of about 40 processor-seconds. For a 2-norm error level of approximately 0.02 with the third test function, we found the run-time requirement for hp-adaptivity to be about 6000 processor-seconds. This compared with an estimated run-time requirement for h-refinement of about 210 processor-seconds and p-refinement of about 60 processor-seconds.

The run times provided are coarse estimations, intended only to give the reader a sense of the cost, and there are several important caveats. The run times cited for the h-refinement and p-refinement are based only on running a calculation at the refinement level already known to give the desired error level; the expense of Monte Carlo

calculations and analysis for adaptively finding the refinement needed to yield a desired error level is not included. Our expectation was that the ratio of run time between a p-refinement calculation and an h-refinement calculation would be approximately proportional to the square of the ratio of the degrees of freedom; the h-refinement calculations consistently outperformed this. We suspect that the p-refinement calculation was less efficient than expected because the Monte Carlo calculation expense consists almost entirely of the tally expense and the moment tallies are more expensive than the spatially binned tallies (as implemented). In a realistic Monte Carlo calculation, where run time is dominated by the physics of the simulation, we would expect moment tallies to perform better than the results shown here. The time required for the hp-adaptivity to converge can vary depending upon the cuts that are chosen, which can be affected by the statistical variation of the data. In all cases, we aimed for having 20% of the error due to statistical uncertainty and 80% of the error due to truncation error, but we did not iterate exhaustively to achieve this precise balance at the precise error level. Other balances between the error terms could affect the relative run times for the different approaches.

v. hp-Adaptivity Conclusions

We have presented a method for adaptively refining Monte Carlo tallies in both spatial division and functional representation. Adaptive tallies offer the potential to make Monte Carlo codes still more user friendly, at the expense of increased computational cost. Other adaptivity goals (such as minimizing run time to achieve an error level) and error criteria (such as an infinity-norm) appear to be possible. Memory requirements could be further reduced by determining an optimal expansion order for each axial component; in regions where a distribution is primarily one-dimensional, significant memory savings could result.

V. WEIGHT-WINDOWS BIASING RESULTS

To facilitate biasing of electron-photon transport in the Integrated Tiger Series (ITS) codes**Error! Reference source not found.**, we have implemented a weight-window generator based on the adjoint Monte Carlo transport capability. Among our goals were: (1) to use the same geometry model in the generator as in the biased Monte Carlo calculation, (2) to avoid the need for geometry-model modification to facilitate biasing, (3) to avoid the need for user intervention in setting weight windows, and (4) to bias any electron-photon problem, including those with rapidly varying electron importance near surfaces. To accomplish these, we use: (1) adjoint Monte Carlo calculations, (2) an adjoint-flux/weight-windows tally structure distinct from the material geometry, and (3) an adaptive adjoint-flux tally structure. Tallies are performed on a tree data structure that has been implemented in the Mesh-Oriented datABase (MOAB)**Error! Reference source not found.**. While the adaptivity could encompass all seven transport dimensions, the current work adapts only three spatial dimensions.

Weight-windows generation and biasing have existed for some time**Error! Reference source not found.Error! Reference source not found.Error! Reference source not found.**, but have only recently been applied to coupled electron-photon problems**Error! Reference source not found.Error! Reference source not found.**. Dionne**Error! Reference source not found.** explored use of deterministic adjoint methods to bias three-dimensional forward Monte Carlo calculations. Ueki**Error! Reference source not found.** focused on using forward calculations to determine particle importance. Ueki demonstrated in one-dimensional calculations the effectiveness of using weight-windows biasing for the type of electron-emission calculations that we will be examining in this paper. Here we attempt to address the issue of adequately representing the weight-windows biasing map in a multi-dimensional problem.

Adaptive Monte Carlo tallies have been described by Booth**Error! Reference source not found.Error! Reference source not found.** and Lai and Spanier**Error! Reference source not found.**. Our method is limited to only refining the phase-space division, as in Ref.**Error! Reference source not found.**. But similar to Refs.**Error! Reference source not found.** and **Error! Reference source not found.**, our approach directly iteratively refines the importance map of a problem. Those works demonstrated that adaptivity of importance tallies could be used to converge toward zero-variance Monte Carlo calculations. Those approaches estimated importance from forward Monte Carlo calculations and used the resulting importance map to bias the next forward calculation, iteratively improving the importance map. Our approach would appear to be advantageous in the early portion of the importance map evolution when detector tallies are very difficult to achieve. We also expect our approach to be more reliable, if more expensive, because using unbiased adjoint calculations to build the importance map is unlikely to over-bias by omitting important low-probability physics paths. (Perhaps these concerns are alleviated when an initial deterministic global importance-map evaluation is used).

In Section V.A, we discuss our spatial-adaptivity algorithm used to refine the adjoint-flux tally distribution. In Section V.B, we discuss our implementation of the weight-windows

biasing method. In Section V.C, we present data on the efficiency of our method for a test problem. We offer some concluding remarks on our weight-windows biasing algorithm in Section V.D.

A. Spatial Adaptivity of Adjoint-Flux Tallies

We use "volume-source" tallies in adjoint calculations. These are equivalent to track-length adjoint-flux tallies but normalized to yield detector-response values for volume-distributed sources. The tally routines were modified to allow particle tracks to be apportioned into a tree data structure.

The spatial distribution of the tallies is stored in a binary space-partitioning tree data structure. Because it is a binary tree, each node of the tree can only be divided into two smaller nodes. For simplicity, we have chosen to constrain the tree cut planes to be axis aligned, which results in a "kd-tree." We require that the root node of the tree is an axis-aligned bounding box, so every node in the tree will also be an axis-aligned bounding box. This tree has been implemented in MOAB**Error! Reference source not found.**. Tallies are only made on the leaves of the tree, where a "leaf" is any node on the tree that is not further divided. Since this is a space-partitioning tree, each point in space within the root node is contained by one and only one tree leaf.

We begin our suite of adjoint calculations with an initial tree that partitions the space of the problem. After each adjoint calculation, neighboring tree leaves are compared to determine whether the ratio of fluxes between them meets a test criterion (e.g., is the ratio less than a factor of three?). If the criterion is not met, then the leaves are split. If the boundaries of the shared faces of the neighboring leaves do not coincide in extent, then cuts will first be made to make the faces coincident. Otherwise, the leaves are split in half in the axial direction corresponding to the shared face. This can be thought of as an adaptive refinement technique driven by error estimation. The error estimator is an infinity-norm assessed only on leaf boundaries using only leaf-average values. In these terms, there are clearly defects to the technique. The leaf-average value may mask important information about solution variation within the leaf, which cannot be assessed. Assessing the error only at leaf boundaries offers limited information. Clearly, we are not assured of achieving a solution within a factor of three of the true solution everywhere within the problem. However, this also suggests that, if necessary, a higher-order error estimator might be able to more rigorously enforce the desired infinity-norm error level.

B. Weight-Windows Implementation

Our implementation of weight windows follows common descriptions of the technique**Error! Reference source not found.**, with a few minor differences. The center weight of the window is stored on each leaf of the kd-tree on which the adjoint particle flux was calculated. The weight is the inverse of the adjoint flux, with all weights normalized by the same value. Ideally, the normalization factor will avoid splitting and

minimize Russian roulette of source particles. (We have not implemented source biasing, which can alleviate these concerns to some degree.) The extent of the window is a user-prescribed setting for the ratio of the top to the bottom of the window, logarithmically centered about each weight. A particle with weight greater than the top of the weight window is split into an integer number of particles sufficient to bring the weight below the center weight. A particle with weight lower than the bottom of the window is Russian rouletted to bring the weight of a surviving particle to the center weight.

Weight-windows biasing is applied to photons at each interaction point and each time they cross a geometry boundary. Weight-windows biasing is applied to electrons at the beginning of each ITS step, which includes each boundary crossing into a unique material. For each application of the weight-windows settings, the kd-tree is interrogated to determine the weight window for the current particle phase-space.

As stated above, the center weight of a window is determined as the inverse of the adjoint flux. However, two special cases must be dealt with: adjoint-flux values with poor statistics and regions of phase space that obtained no adjoint-flux tallies. A lack of tallies in the unbiased adjoint calculation can occur for two reasons: a low adjoint-flux level and/or a small-volume tally region. Our approach is to determine the largest value of the product of flux and volume (equivalent to total track-length tallied in the region) with relative statistical uncertainty greater than 20%. We reset all smaller flux-times-volume values to this level. This is a conservative approach that systematically over-estimates the importance of particles in regions of phase space that lack adequate information.

C. Weight-Windows Results

To test and demonstrate our method, we use a complicated real problem that conveniently can be analyzed with the current constraints of our algorithm. That is, most of the geometry boundaries are axis-aligned, so many of the particle-importance contours will be nearly axis-aligned and can be efficiently represented using a tree structure with axis-aligned cutting planes. This problem involves modeling a validation experiment performed on Sandia's Saturn electron-accelerator facility. The geometry of the Saturn bremsstrahlung converter and cavity device is shown in Figure 26.

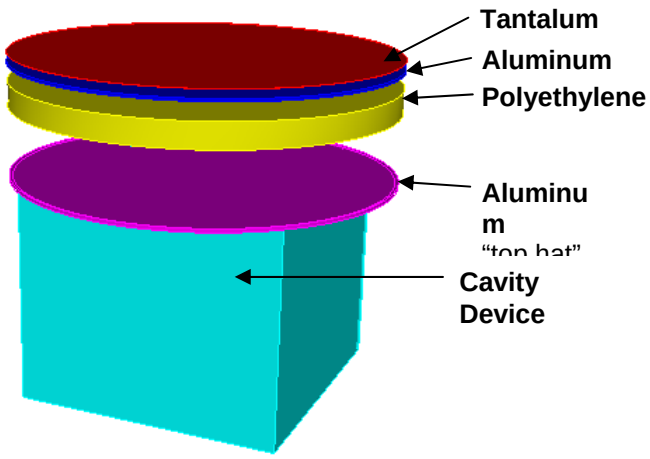

Figure 26. Geometry model with converter and cavity device.

The Saturn "converter" consists of tantalum, aluminum, and polyethylene layers to convert source electrons into bremsstrahlung photons, which impinge from above onto the cavity device. A source of electrons is modeled in three concentric rings. The source electrons are 1.6 MeV and normally incident on the tantalum. An aluminum "top hat" sits over the cavity device. The cavity device is simplified as a square shell of brass wrapping around the four sides of a hollow graphite box. A portion of the inside surface of the top of the graphite box is covered in a thin layer of gold. The goal of the ITS calculation is to calculate the distribution of electrons emitted into the cavity, differential in the five dimensions of surface area, angle, and energy. Most of the electrons entering the cavity are photo-electrons from the gold layer. However, electrons also enter the cavity due to photon interactions in the graphite. Plus, the graphite is sufficiently thin to allow electrons from the aluminum top hat and brass shell to transport into the cavity.

We begin by performing a calculation of the adjoint particle flux due to the detector response of electrons emitted into the cavity. Figure 27 and Figure 28 show the evolution of the importance map for the highest-energy electron flux for two separate sets of calculations. Despite our objective to avoid the need for user intervention in constructing the weight-windows settings, there are several "knobs" in the algorithm during the adjoint-flux adaptivity. First, the user must decide on an initial tree to begin the iteration process. In these figures, the first tree used a uniformly refined tree. A better choice is to begin with a tree that incorporates information about material boundaries, but this may be more labor intensive for the user. For each iteration of the tree, the user must choose the adjoint-flux tallies. Since our algorithm only partitions spatial dimensions, the user must choose the energy and angular binning of the flux tallies. In the example problems, five energy bins and one angle bin were used for both electrons and photons. Also for each iteration of the tree, the user must select the number of histories to be simulated. In the third flux map shown in Figure 27, corresponding to the 20^{th} iteration on the tree, the poorer statistical results are shown by the dark blue cells indicating zero flux. Another feature of the results worth noting is that the importance inside the cavity has been over-estimated compared to a true adjoint calculation of electron emission into the cavity. The adjoint source was

intentionally modified to accomplish this so as to prevent the tree-refinement algorithm from refining the importance map within the cavity, where the forward calculation will not track electrons.

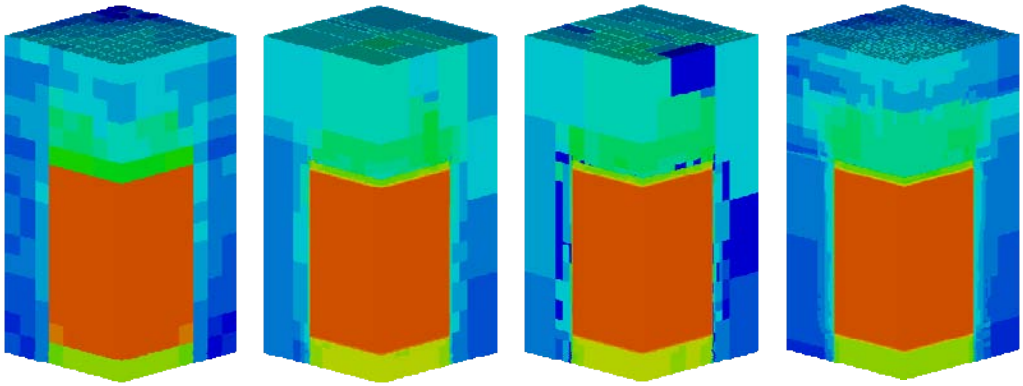

Figure 27. Adjoint-flux distribution for the highest-energy electrons after 1, 10, 20, and 30 Monte Carlo calculations.

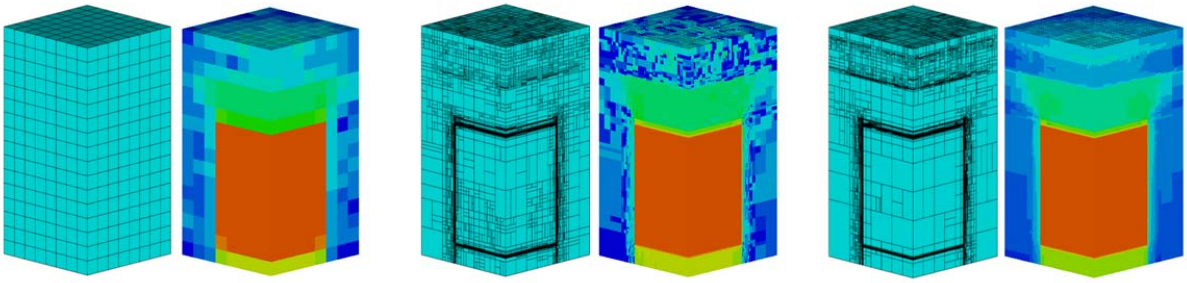

Figure 28. The tree-data-structures and the adjoint-flux distributions for the highest-energy electrons after 1, 10, and 26 Monte Carlo calculations.

In Figure 29, some metrics on the evolution of the tree are provided as a function of the adjoint Monte Carlo run time. Abrupt changes in the level of the metrics (especially the number of leaf merges) occurred twice at approximately 3E+5 and 1E+6 seconds due to changes in the number of energy groups in the adjoint importance data. Increases in the horizontal spacing between data points are due to increases in the number of histories in the adjoint calculation. The number of leaves suggests some convergence in the tree. Examination of the tree showed that much of continuing refinement was occurring in regions of the problem where importance contours were not axis aligned, such as near the edges of the cavity.

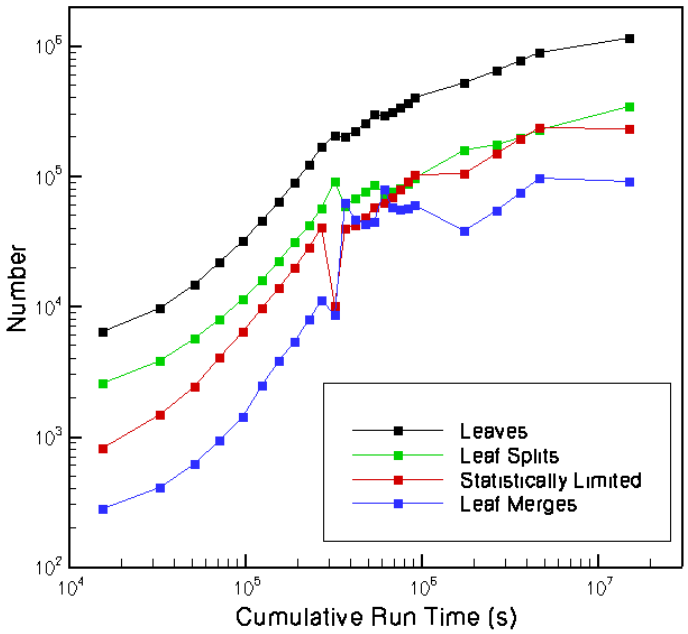

Figure 29. Tree evolution metrics.

In Figure 30 and Figure 31, line-outs of the converged electron and photon adjoint fluxes (i.e., unnormalized importances) are shown on log-log plots as a function of distance upward from the gold-cavity interface. These results do not include the "fix up" for poor statistical results, which are visible in the low-energy electron results. Indeed, the lowest-energy results for electrons and photons received no tallies beyond the gold material of the first 12.5 μm. However, the importance map has been resolved over almost six orders of magnitude. The map has resolved both the rapid variation of importance of the lowest-energy electrons over almost six orders of magnitude in importance across less than 1 μm and the slower variation of the highest-energy electrons over four orders-of-magnitude in importance across 10 cm. In the results that were "assisted" with material boundary information, we observe that above the 10^{-9} flux level in the line-out data, the tree has spatially resolved factors of three within each energy level (with one exception in the 10-50 keV photon flux in the converter region). However, deficiencies in the importance map remain. From the data in Figure 30 and Figure 31, we can observe that the coarse energy binning causes significant differences in importance from one energy bin to the next at some locations. This data does not indicate how much variation in importance might exist as a function of angle, which has been suppressed into a single angle bin. Also not shown in these figures, the tree has not fully resolved importance gradients where contours are not axis aligned, but this would require a very large number of leaves using axis-aligned cutting planes.

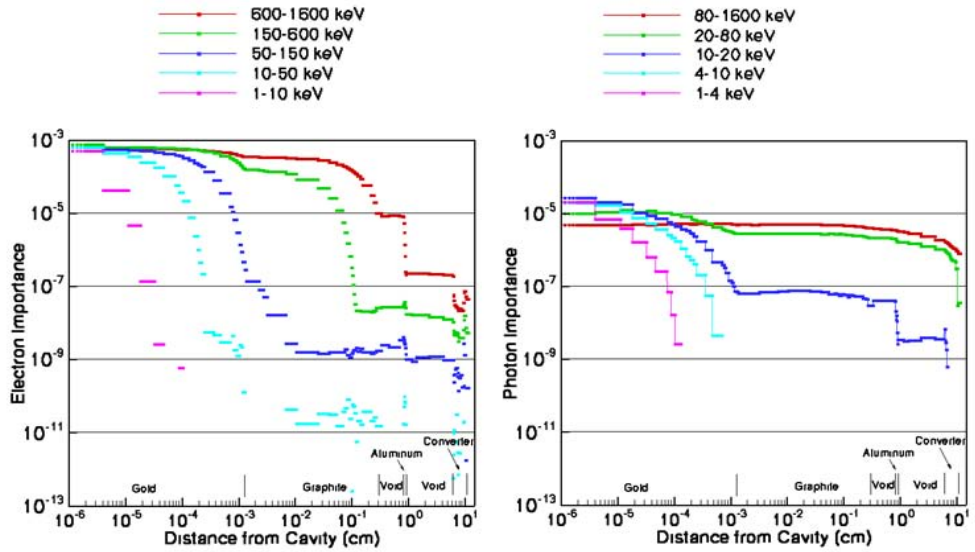

Figure 30. Line-outs of the adjoint-flux distributions for electrons (left) and photons (right).

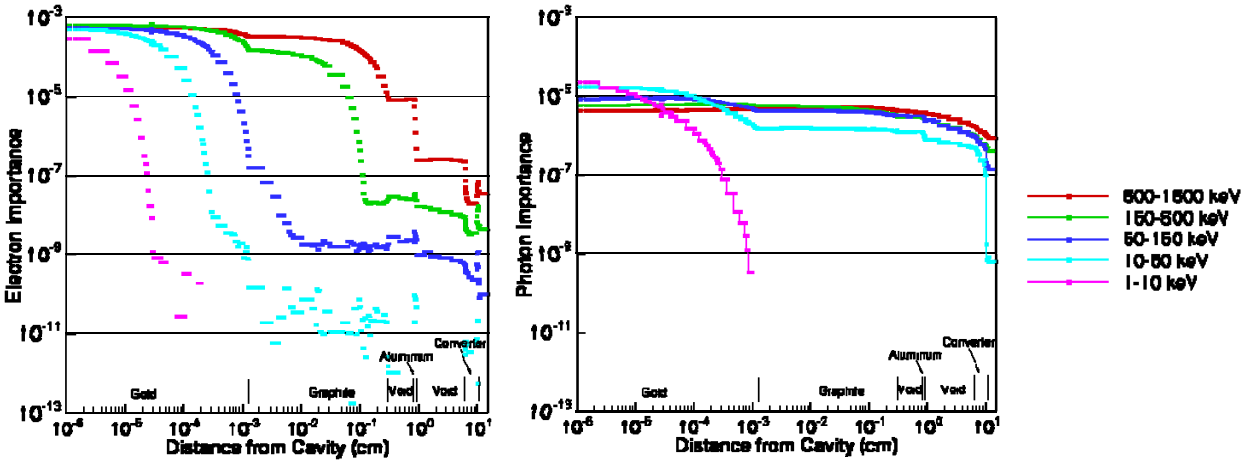

Figure 31. Line-outs of the adjoint-flux distributions for electrons (left) and photons (right), achieved using a tree initialized with some material boundary information.

The weight-window settings were determined from the adjoint fluxes, and a line-out of this data is shown in Figure 32. The inverse values of the adjoint flux were normalized by the inverse value of the adjoint flux at the forward source location (in this case, specified as the point at the center of the Saturn rings). Note that the electron weight is approximately one within the converter. In this case, a simple "capping" scheme was applied to the weight settings; the window setting was not allowed to be greater than the largest weight setting that had statistical uncertainty better than 20%. Unfortunately, such a scheme is not conservative in setting the weight window and tends to have a cap that is determined by a relatively large-volume tree leaf, where good statistics were easier to achieve. A scheme was implemented that attempted to account for the leaf volume, but it was found that relative statistical uncertainty depended too much on the problem physics and location in the problem. A more complicated scheme would be

required to automate the process of setting weight windows in regions where importance estimates failed to achieve good statistics. An alternative would be to allow a code user to manually study the effect of varying a weight cap for the problem.

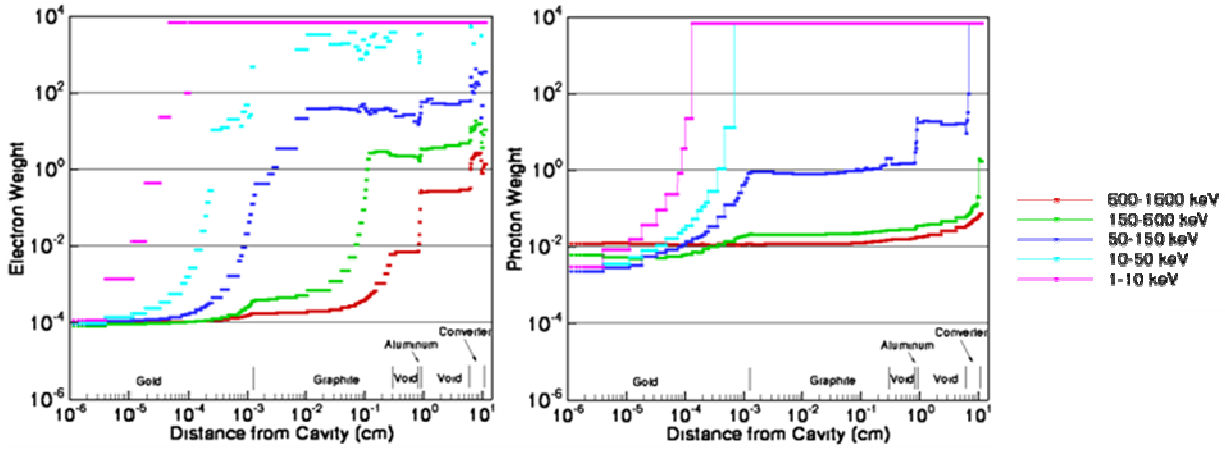

Figure 32. Line-outs of the weight-window settings for electrons (left) and photons (right).

The forward calculation was run to calculate electron emission. The emission tallies were differential in 7 surfaces, 128 angle bins, and 40 energy bins. The calculation was performed using both weight-windows biasing and manual biasing. The manual biasing used three techniques: (1) scaling the cross section to increase bremsstrahlung production in the converter, (2) raising electron-cutoff energies in some regions, and (3) trapping electrons to terminate electron transport when appropriate. (It may be worth mentioning that the manual biasing originally used with this problem included forcing of interactions near the cavity boundaries. Anomalies in the emission tallies indicated that the calculation was being over-biased, a common danger with manual biasing techniques that can lead to false confidence in erroneous results.) The electron emission results are shown in Figure 33. The “tally index” merely refers to the ordering of the data output from ITS. Tally indices between 1 and 20480 refer to electron emission from the sides of the cavity, between 20481 and 25600 refer to electron emission from the graphite on the top of the cavity, between 25601 and 30720 refer to electron emission from the gold on the top of the cavity, and between 30721 and 35840 refer to electron emission from the graphite on the bottom of the cavity.

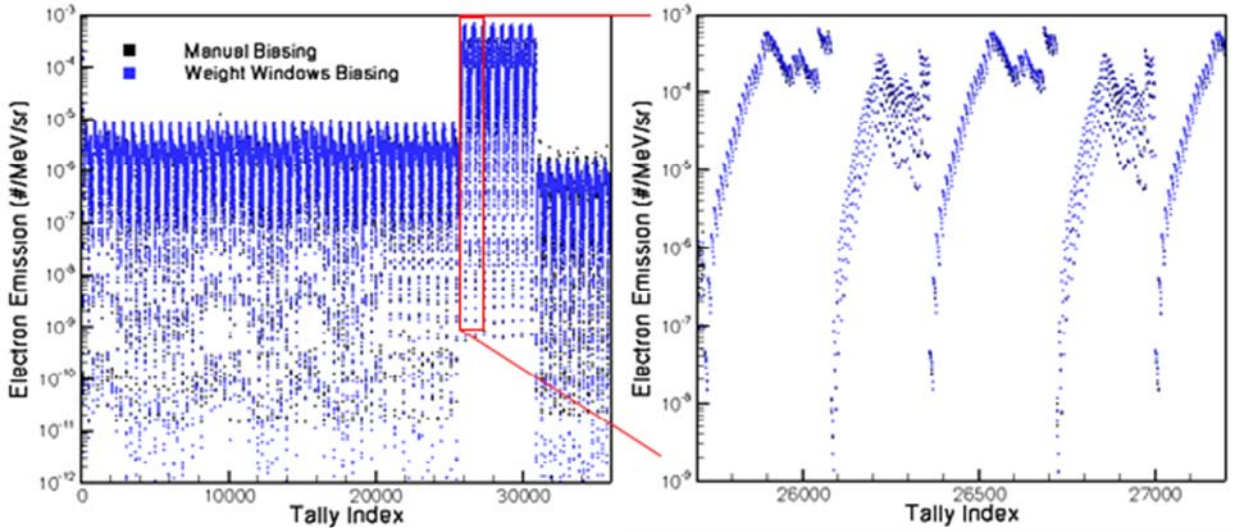

Figure 33. Cavity electron emission as a function of ITS tally index.

To determine whether differences between electron-emission results obtained with manual biasing and weight-windows biasing were statistically significant, t-statistic values were calculated as:

$$t = \frac{x_{manual} - x_{ww}}{\sqrt{\sigma^2_{manual} + \sigma^2_{ww}}}, \tag{24}$$

where σ^2 is the variance of the tally and x is the mean estimated emission value. Figure 34 shows the t-statistic comparison as a function of tally index. These results indicate that the electron emission from the gold shows statistically significant disagreement between the results. Based on closer examination of the data, we found that the disagreement occurs at low energies. We believe this is due to errors introduced by the manual biasing mechanisms of electron trapping and raised electron cutoff energies. This theory was supported by results generated using these manual biasing mechanisms in combination with the weight-windows biasing, which achieved results in agreement with the manual biasing results. The discrepancies show up in the gold emission results because of the high numbers of electrons emitted from the gold; presumably, longer calculations that achieved lower statistical uncertainties would reveal discrepancies in the low-energy electron emission from all surfaces.

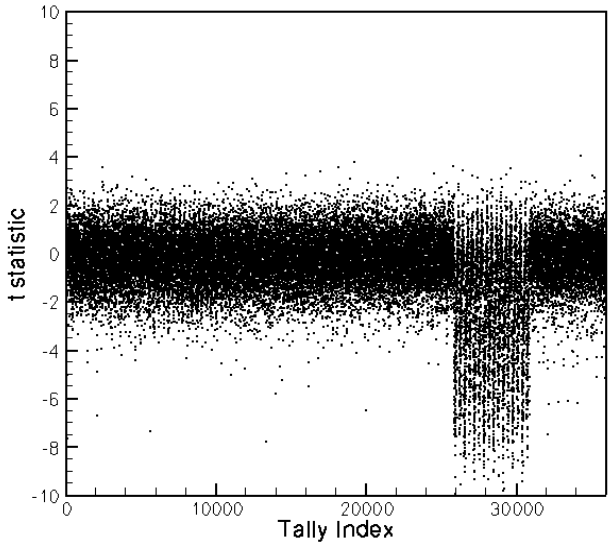

Figure 34. T-statistic comparison of electron emission results with manual biasing and weight-windows biasing.

In Figure 35, we show the "figure of merit ratio" as a function of the electron-emission tally. The figure of merit is defined as

$$FOM = \frac{1}{\sigma^2 t}, \tag{25}$$

where σ^2 is the variance of the tally and *t* is the computational time. The figure-of-merit ratios are given relative to the manually biased calculation. The "automatic" results used the algorithm previously described in this paper. The "assisted" results differ from the "automatic" results only in that they began with a tree that had been divided along material interfaces near the surface of the cavity. An "automatic calculation" was later performed using weight-window settings from a tree that had been refined through more steps (with different refinement options, numbers of histories, etc.). This later calculation achieved figures-of-merit, shown in Figure 36, that are comparable to the previous "assisted" results. From these results, we conclude that the weight-windows biasing scheme has achieved an efficiency improvement of approximately an order-of-magnitude over a manually biased calculation for this test problem. Some improvement in efficiency and in convergence of the tree structure was achieved when the adaptive tree was initialized with key information about geometry boundaries, but this was not necessary. As previously noted, calculations were performed using both weight-windows biasing and the manual biasing mechanisms of trapping and raised electron cutoff energies (and the manual biasing mechanisms were observed to affect the accuracy of the results). In these calculations, the FOM was generally about 50% better than without the addition of the manual biasing mechanisms.

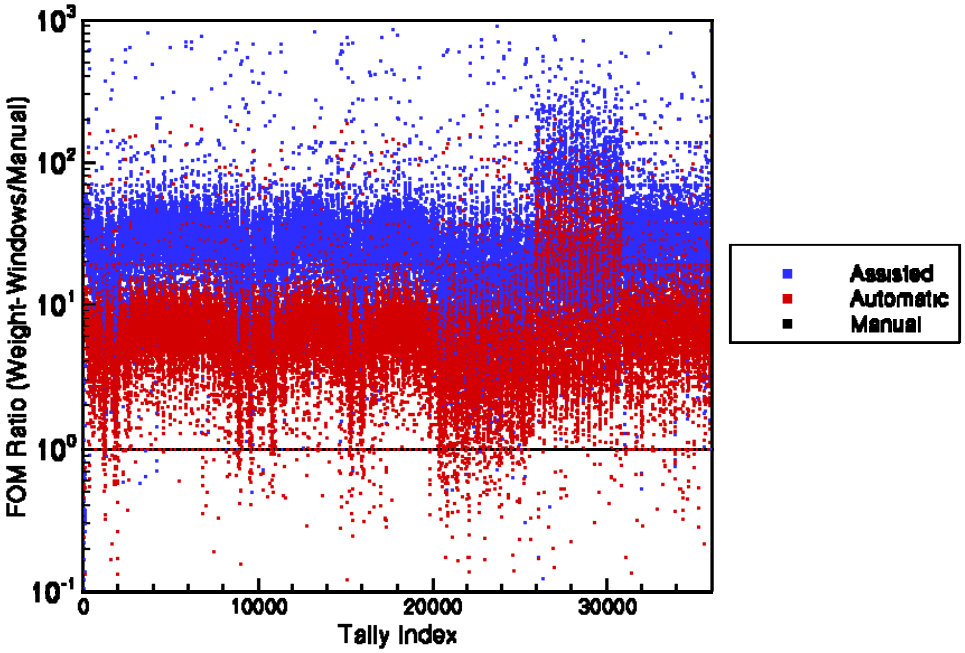

Figure 35. Figure-of-merit ratios for electron-emission tallies in three biased Monte Carlo calculations.

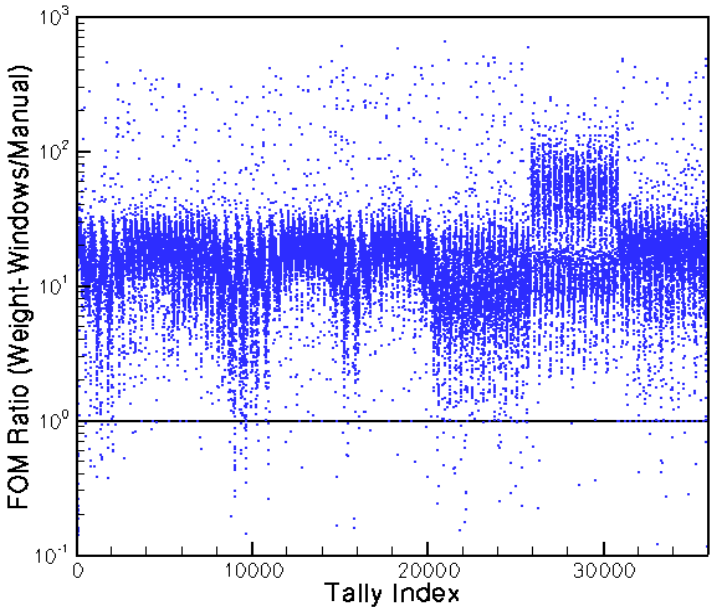

Figure 36. Figure-of-merit ratios for electron-emission tallies in biased Monte Carlo calculations.

In light of the effect of manual biasing on the accuracy of the results, one may wish to consider the efficiency of the weight-windows biasing as compared to an analog calculation. Based on calculations using coarse binning structures (five energy bins and a single angular bin), the manually biased calculation was estimated to be about four orders-of-magnitude more efficient than an analog (completely unbiased) calculation,

indicating that the weight-windows biased calculation is about five orders-of-magnitude more efficient than an analog calculation.

Our investigation of weight-windows biasing was not without cautionary notes. In one calculation a large number of low-energy tallies from the gold surface were observed to have significantly degraded FOMs. These degraded FOMs corresponded to tallies with increased mean values and increased statistical uncertainties. We have attributed this anomaly to a "zinger", a history that contributes large tallies and significantly disturbs the convergence of statistical results. This event was caused by a low-energy photon that travelled farther than was probable, starting with a high weight and moving to a region where it should have had a low weight, where it interacted to produce numerous low-weight electrons, many of which contributed to emission tallies. There were two "lessons learned" that were drawn from this event. (We could say that these were lessons of which we were theoretically aware, but of course, one appreciates a theoretical point much better when it is encountered in practice.)

The first "lesson learned" from the zinger event was that photons needed to be tracked on the weight-window tree so that the photon weight would be forced to stay within the weight-window as it was tracked. When this is done, the photon is split as it moves to a region where it is more important and must have a lower weight. This prevents a single high-weight photon from travelling an anomalously large distance. Among the split particles, the weight windows should exert an indirect form of population control. That is, the importance should be indicative of the probability of the photon travelling over some distance. If the weight windows are properly set, then as a photon is moved through a region where splitting is often being applied, *on average* a single photon should survive, resulting in a single low-weight photon, which can interact to produce a single low-weight electron that can contribute to a single detector tally. That is, through each region where weight windows causes a photon to be split into three photons, only one of the three photons should reach the next region where the photon(s) will again be split into three.

The second "lesson learned" from the zinger event was that weight windows in themselves cannot always prevent anomalous events. While continuously tracking photons on the tree may be sufficient for this problem and will significantly reduce the probability of zinger events, weight windows are not a perfect form of biasing. Weight windows do not affect the outcome of physical events, instead they make adjustments to the particle weight/population after events. Particle tracking "events" happen continuously, and as discussed in the previous paragraph, particle tracking can be interrupted to apply weight windows as needed. Most other events (such as scattering, photoelectric conversion, pair production, etc.) involve discontinuous changes in the particle phase space (i.e., abrupt changes in the direction of flight, energy, and/or particle type). In these cases, large variance can be introduced. Weight windows can only try to mitigate the situation after-the-fact. In some cases, weight windows may roulette the resulting particle, which may result in zero tallies for the particle history. Zero tallies mean that the calculation is less efficient than is otherwise possible, but these are not zinger events, and they are only of concern if they are happening too

often. Of greater concern are events that dramatically increase the importance of the particle, in which case weight windows can only split the particle into many replications. That split may be too late to prevent many or all of the split particles from contributing to a single tally, which is a “zinger”. Prohibiting such anomalies is a difficult undertaking. We have identified only two possible approaches that might prohibit “zingers” due to discontinuous changes in particle phase space. (Neither of these approaches has been tested.)

Importance sampling is the first approach that may prohibit zingers within weight-windows biased calculations. While theoretically enticing, importance sampling is difficult in practice. Most physical events involve sampling from a complicated distribution. Importance sampling requires the sampling of such complicated distributions to be continuously weighted by the importance of the outcome. On the positive side, importance sampling can be introduced piecemeal, and each addition may be valuable in improving efficiency and reducing the likelihood of zingers. For example, one can bias photoelectric-absorption production of electrons independent of the biasing of Compton-scattering production of electrons, and at low energies, photoelectric events may be the dominant means of electron production. Ultimately, a calculation with importance sampling of all events would not require weight-windows, as particles would always have the correct weight. This is the goal of adaptive importance sampling algorithms.

The second approach to prohibit zingers within weight-windows biased calculations is to attempt setting weight windows conservatively. That is, for each point in phase space, the target particle weight (i.e., weight-window setting) would be determined by the highest particle importance that can be reached discontinuously from that phase space. In many cases, this would include the phase-space of all particle types, angles of flight, and all lower energies at each spatial location. The resulting simulation would track too many particles in almost all regions of phase-space. Many of the tracked particles would eventually be eliminated by roulette. This would increase the expense and decrease the figure-of-merit for the simulation. Examining the importance of electrons and photons in Figure 30, we can observe that (1) electrons generally have higher importance than photons near the detector, (2) photons generally have higher importance than electrons far from the detector, and (3) the only place that we find low-energy particles with higher importance than high-energy particles in for photons near the detector. Using the electron importance to set photon weight windows near the detector was an approach used by Ueki**Error! Reference source not found.**, and this is probably a reasonable approach (due to the much higher computational expense of tracking an electron versus tracking a photon). However, using photon importance to set electron weight windows far from the detector appears unreasonable (due to the same relative tracking expenses), as this would result in the tracking of many more electrons that have very little importance. It appears that a preferable approach for the electrons would be to implement biased sampling of bremsstrahlung photon production and electron-interaction fluorescence photon production.

Based on this experience, several variations on the algorithm were tested. Tracking of photons on the tree (so that photon weights were always maintained within the window for any phase-space that was encountered) was found to decrease the efficiency of the calculation by about a factor of 4. Figure 30 shows that at some locations the importance changes much more than a factor of three between energy groups (i.e., the importance map has not been sufficiently resolved in energy). We made a cursory examination of the effect of refining the energy group structure to better resolve importance as a function of energy. Using 20 electron energy groups and 10 photon energy groups, we observed about a 50% improvement in the efficiency of the calculation.

D. Weight-Windows Conclusions

We have demonstrated the capability of our adaptive importance maps to achieve a sufficiently accurate importance map to result in an effective automated biasing method. For a realistic test problem, an efficiency improvement of approximately an order-of-magnitude was achieved over a manually biased calculation. Based on calculations using very coarse binning structures, the manually biased calculation was estimated to be about four orders-of-magnitude more efficient than a completely unbiased calculation. However, memory limitations encountered using the axis-aligned cutting planes of a kd-tree for particle importance maps severely limit the complexity and/or efficiency of problems that can be analyzed. To analyze more practical problems, a more general spatial-partitioning approach must be implemented.

VI. CONCLUSIONS

We have succeeded in demonstrating the principles on which the proposal was based. That is, we have demonstrated that adaptive refinement of the adjoint-flux map can be used with weight-windows biasing to significantly improve the computational efficiency of coupled electron-photon calculations in which electron transport is of localized importance. We have found adaptivity algorithms that function well for this purpose. We also found adaptivity algorithms that may function well for other similar purposes. Better methods for resolving the particle importance map may exist, and better biasing methods may exist, but the goal was to advance the state-of-the-art, develop an automated practical approach, and achieve more effective biasing for a class of problems.

There are two primary objectives (that were listed among the milestones in the original proposal) that this project has failed to achieve. Foremost, we have not yet succeeded in demonstrating this biasing approach for arbitrary importance maps using non-axis-aligned space-partitioning data structures. At this point, we believe that a suitable algorithm has been identified and may be demonstrated with additional effort. Secondly, a third-year milestone to demonstrate a "proximity tracking algorithm" was never attempted. The idea was to efficiently incorporate one-dimensional importance map data for particles that were "near" the detector, without the need for explicit non-axis-aligned spatial partitioning. In the end, we did not formulate an approach to implementing such a capability that we felt was promising enough to pursue.

There are many possible additional investigations that could follow from this project. We list a sampling of ideas that we believe are promising:

- The ability to cope with non-axis-aligned importance contours arises even in problems with purely axis-aligned geometry. Applying this capability to general problems requires a more flexible space partitioning scheme. The grid-based bsp-adaptivity algorithm seems very promising, but there are many possible approaches. For instance, one could require the user to provide an initial bsp-tree or mesh to initialize the adaptivity algorithm with sufficiently contour-aligned hexahedral elements.
- There are many sampled processes in the transport calculation. Using the importance information that underlies the weight-window settings, these processes could be converted to automatically use importance biasing. Some effort was expended late in the project to examine survival biasing in combination with scaled photoelectric interactions, but no conclusive results were reached. It would also be relatively simple to scale the production of bremsstrahlung photons based on importance data (though biasing the energy and angle sampling would be a much more difficult undertaking).
- Adaptivity has only been applied to the three spatial dimensions. It could be extended to all seven dimensions of the transport problem.

- Our efforts to cope with statistical uncertainty in setting weight windows met with mixed success. One would like to strike a balance, setting the weight window to highest importance that may occur within that region of phase-space but setting the importance no higher than absolutely necessary. For coupled electron-photon problems, it seems that one could generally apply some logic to examining the calculated importances at higher energies, since one expects importance to decrease with decreasing energy (with the exception of photons near the detector in the cavity electron-emission problem). For a well resolved importance map, one also expects importance to vary slowly as a function of phase-space, so nearby importances with good statistical estimates may indicate a conservative setting of the weight window.
- The adaptivity capability may allow an initial deterministic calculation to be much coarser. That is, a very crude deterministic calculation could be used to make an initial spatial-partitioning structure, and the adaptivity algorithm could make refinements from that.
- The adaptivity algorithm could be extended to allow forward importance tallies to further refine the importance map. Beginning with either deterministic or Monte Carlo adjoint results, the importance map could be further refined in forward calculations. Depending on the quality of this initial importance map, this approach may require some automation in the specification of forward sources. (As briefly discussed in Section II, one may need to iteratively move the source away from the detector while building the importance map.)
- The adaptivity algorithms must strike a balance between effectiveness in achieving the goal and computational expense. One can gain greater confidence that the converged adaptivity result will be sufficiently accurate by increasing the data-acquisition requirements (i.e., the number spatial partitions and functional-expansion coefficients) and performing more rigorous error-metric assessment, but these require increased computational memory and processing. More experience is needed to appreciate how to perform this trade-off.

APPENDIX A. PROJECT PROPOSAL

1. Overview of the Problem and Idea

A variety of analysis problems involving coupled electron-photon transport exist in which electron transport is primarily of interest near geometry boundaries. These problems arise especially in assessing radiation effects on electronics. Since electrons are computationally expensive to simulate, it is desirable to transport electrons only as necessary. Ideally, particles would be transported in proportion to their importance to the solution. Variance reduction techniques can accomplish this based on importance information. Importance is proportional to adjoint flux and can be calculated using adjoint transport methods. The difficulty is in determining and representing the importance map for a coupled electron-photon problem, in which electron importance can have sharp gradients near boundaries of interest. To record this information for diverse problems on arbitrarily complex CAD models, an adaptive approach is needed to provide sufficient accuracy while minimizing data requirements. Developing an effective technique could dramatically accelerate calculations for various types of radiation effects, including electron emission and escape, dose enhancement at material interfaces, and deep-penetration satellite shielding.

2. Proposed R&D

2.1 Technical Approach

Our approach is based on traditional weight windows techniques but is unique in developing novel methods to minimize memory usage. We will develop efficient data structures to store importance information and develop weight windows biasing mechanisms to utilize this information. There are many possible data structures, but they must meet the following requirements: They must be (1) capable of storing data with a high degree of spatial anisotropy and on length scales several orders of magnitude smaller than the characteristic size of the geometry model, (2) constructed automatically based on geometric information and/or importance data from adjoint transport calculations, and (3) computationally efficient for both point-based evaluation as well as straight-path integrals. We will investigate a variety of data structures capable of meeting these constraints. Two possible adaptations are in the partitioning of space and in the representation of the importance function within each partition. To partition space, we will use tree data structures. Within each partition, we will use functional expansion techniques.

For space partitioning, the octree, kd-tree, and binary splitting plane (BSP) tree data structures will be considered. We expect to find decreasing memory requirements and increasing computational expense in that progression. Since the complexity increases in going from octree to kd-tree to BSP-tree, we will attempt to enable the methods in that order. Although the BSP-tree can be expected to require greater computational expense to build than an octree, the improved efficiency in biasing expensive forward calculations is expected to make the algorithm effective.

We will have two methods to construct the trees. A coarser (possibly preliminary) method will be based on an adjoint calculation consisting only of electrons that would contribute detector tallies. Points along the path of each electron history will be examined to determine whether they extend into a region of the problem where electrons have not been detected before. Electrons will be stopped at a high energy cutoff, thus limiting the range of their transport. As the simulation proceeds, the tree structure will identify the maximum range of the electrons from the detector surface(s). A more refined tree building technique will be based on adjoint calculations that include the electron-photon coupling physics. A full adjoint calculation will be performed, with adjoint flux (or importance) tallied on a tree structure. The variation of fluxes across the leaves of the tree will be used to determine where to refine and where to coarsen the tree. Another adjoint calculation will be performed tallying

on the modified tree, and the process will be repeated until the variation of fluxes is deemed acceptable. A general guidance is that weight windows should apply splitting or Russian Roulette to change weights by a factor of 2 or 3. This guidance can be used to determine when a desirable tree has been obtained. This method may build on a preliminary tree obtained by the lower-order method.

In addition to the data structure used to partition space and represent the importance function, there is also the question of how to interpolate that function inside the space. This problem is similar on one level to evaluation of shape functions in the finite element method, except that in our case the spatial "elements" are axis-aligned hexahedra. Still, we may be able to borrow methods like p-adaptivity from finite elements. P-adaptivity has been shown able to interpolate smoothly-varying functions more efficiently than h-adaptivity. The combination of the two, referred to as hp-adaptivity, has been shown to achieve exponential convergence when used in the finite element method. However, these shape functions may be more expensive to evaluate. We will investigate the use of these methods for interpolating importance functions on an adapted grid data structure.

In the forward calculation, particles will be tracked on both the real geometry and an importance-map geometry. The latter may entail tracking directly on the tree data structure or tracking on a facetted geometry created from the tree. Using a facetted representation may provide memory savings for the forward calculation where memory will be more constrained than in the adjoint simulation. Variations in particle importance will be applied via weight window settings stored on the importance-map geometry. It is not certain how memory-efficient a tree can be for the most extreme problems: those with short electron ranges and many complex surfaces throughout a large geometry. In the most memory-constrained problems, the importance map may only convey the extent of electron range from surfaces of interest. Within that region, weight windows will be applied based on particle proximity to a surface of interest. This last approach will require a “proximity tracking algorithm” that determines how near to the surface a particle is as a function of its path while it is within an electron range of the surface. This algorithm must be used to bias photon-production of electrons and to apply weight window settings to the electrons.

2.2 Key R&D Goals and Project Milestones

Goal/Milestone	Completion Date
Demonstrate octree adaptivity for electron-range simulations	03/31/2007
Demonstrate tree adaptivity for importance data	03/31/2007
Demonstrate functional expansion adaptivity for importance data	09/30/2007
Demonstrate kd-tree adaptivity for electron-range simulations	09/30/2007
Demonstrate tree and functional expansion adaptivity for importance data	03/31/2008
Complete implementation of weight windows geometry tracking	03/31/2008
Demonstrate octree with functional expansion adaptivity	09/30/2008
Demonstrate kd-tree adaptivity for importance tallies	09/30/2008
Demonstrate BSP-tree adaptivity for importance tallies	03/31/2009
Demonstrate proximity tracking algorithm	03/31/2009
Demonstrate kd-tree and functional expansion adaptivity	06/30/2009
SAND report	09/30/2009

2.3 Risk and Likelihood of Success

- The efficiency of calculations will be dependent on our ability to resolve the contours of electron importance maps. For problems with intricate contours, compromises to the algorithm are possible:

    - Instead of resolving numerous contours, we can resolve only the major contour of electron range from the detector. Within this more important area, we may apply an approximate interpolated field of the importance map or a simplified importance map relative to the nearest surface.
    - The sensitivity of the efficiency of photon-electron calculations to the accuracy of importance map is unknown. A conservative approximation to the importance contour map will provide an accurate simulation, but with less-than-ideal efficiency. This may still provide a significant improvement over existing capabilities.
- When using arbitrarily placed cutting planes, it is not clear what algorithm should be used to move or add planes to adapt to new data. Known point-insertion techniques may not be effective for modifying trees for our purposes.
    - Tree data structures may not be tracked on (if converted to a facetted representation), so the algorithm may be less constrained than for trees intended for geometry interrogation acceleration.
- The constraints of the algorithms are not fully known for the proposed applications. Monte Carlo calculated values will include statistical uncertainty, which may be detrimental to adaptivity algorithms.
    - Several approaches will be explored. Some algorithms are expected to be more memory constrained and some will be less efficient. Some may be more robust in coping with statistical uncertainty.

3. Relationship to Other Work

Previous Work

Weight windows have been most extensively used in the MCNP code [1]. Most prior efforts to develop biasing of photon-electron coupling required the code user to manually define regions where electrons were of interest. Tree data structures have been mostly developed for computer graphics applications. V. Havran [2] has performed an extensive investigation of BSP-tree algorithms. Tree data structures have recently been applied to CAD and facet geometry interrogation algorithms in Monte Carlo transport codes. D. Griesheimer, et al. [3] have studied tallying and using Legendre moments of data within Monte Carlo calculations.

Relationship to Other Ongoing Work

B. Dionne and A. Haghighat [4] and T. Ueki [5] have examined using weight windows for biasing electron-photon transport. These efforts on focused on using data stored on the regular structures within the MCNP code.

Tree data structures are actively being researched and improved for geometry interrogation methods for computational geometry, computer graphics, and robotics. We are not aware of comparable efforts to use such structures for storage of information that is not directly tied to geometry.

The highly anisotropic importance map required for electron transport could be viewed as being similar to boundary layer modeling in fluid transport. In this context, the question arises as to why a boundary-layer unstructured mesh is not being proposed to represent importance functions. However, boundary layer mesh generation remains a user-intensive task, lacking the automation required of Monte Carlo analysis. It is also unlikely that evaluations on an unstructured mesh would be as efficient as on a grid-based data structure.

Appropriateness of Approach

Weight windows based on adjoint importance maps offer the most robust method for biasing Monte Carlo transport calculations with the lowest requirement for user intuition applied to a problem.

Biasing for electrons near geometry surfaces is a most extreme case of rapidly varying particle importance. An effective automated biasing method requires a sufficiently accurate representation of this importance map. Our approach brings together several innovative techniques (tree data structures, functional expansion techniques, and adaptivity algorithms) to attempt to maximize accuracy with minimal data. Alternative approaches have been used, but they are more problem-dependent (by making assumptions about the geometry near the surface), less automated (by requiring code users to customize biasing for specific problems), or less efficient (for example, by only coarsely representing the importance map).

While coupled electron-photon deterministic transport methods could provide the needed resolution, they cannot handle the geometrical complexity of CAD engineering models. While the Monte Carlo method can handle complex geometries, it requires extremely long computational time for applications in which information must be determined "everywhere", or over much of the solution space. The methods in this proposal are intended to greatly reduce that long computational time. If deterministic methods are developed that are capable of generating sufficiently accurate importance maps, that data can be used within the method being proposed to bias forward Monte Carlo calculations.

4. Resources

Key Research Team Members:

Name	Org	Role
Brian Franke	1341	PI
Tim Tautges	1421	Geometry Expertise
Ron Kensek	1341	Radiation Transport Expertise

Qualifications of the Team to Perform This Work:

Brian Franke and Ron Kensek are ITS code developers and users with expertise in coupled electron-photon transport algorithms and applications. Tim Tautges is a member of the Cubit code team with expertise in geometry modeling and interrogation techniques.

5. Importance

Relevance to Missions:

The techniques developed by this LDRD will enable efficient simulations with realistic system models. They will improve the efficiency and ease of assessing electron escape and dose enhancement in microelectronics. Both of these effects can contribute to the electrical response of a component. The techniques will also improve the efficiency of deep-penetration satellite shielding assessments.

Programmatic Benefit to Investment Area, If Successful:

This proposal responds to the Science of Extreme Environments Investment Area. It is intended to improve predictive capabilities for radiation effects in radiation environments. It builds on (1) existing transport capabilities within the Integrated TIGER Series (ITS) code, (2) knowledge of how to build tree data structures for geometry in ITS and CGM (the Common Geometry Module), and (3) weight windows technologies developed in other codes. For the purposes of representing particle importance maps, we will develop an efficient, adaptive method for representing information that will attempt to maximize accuracy while minimizing storage. Algorithms for efficiently and adaptively acquiring information from computational simulations offer the prospect of harnessing computational resources to reduce the burden on code users to apply intuition about physical problems or process

large amounts of data. For this LDRD, the importance map information is expected to greatly improve the efficiency of difficult calculations.

References

[1] X-5 Monte Carlo Team, "MCNP – A General N-Particle Transport Code, Version 5", LA-UR-03-1987, Los Alamos National Laboratory, Los Alamos, NM (April, 2003).

[2] V. Havran, *Heuristic Ray Shooting Algorithms,* Czech Technical University, Prague, Dissertation Thesis (2000).

[3] D. P. Griesheimer, W. R. Martin, and J. P. Holloway, "A Functional Expansion Method for Monte Carlo Eigenvalue Caclulations," in proceedings of *MC 2005*, Am. Nucl. Soc. Topical Meeting, Chattanooga, TN (2005).

[4] B. Dionne and A. Haghighat, "Development of the ADEIS Variance Reduction Methodology for Coupled Electron-Photon Transport," in proceedings of *MC 2005*, Am. Nucl. Soc. Topical Meeting, Chattanooga, TN (2005).

[5] T. Ueki, "Weight Window Generation for Photon Electron Coupled Transport with Photon Source and Electron Detection," *ANS Transactions*, **93** (2005).

APPENDIX B. ADJUNCTON SOURCES FOR CAVITY EMISSION

In principle, sampling is straightforward to obtain the distribution of adjoint electrons for a cavity emission detector. The adjunctons should be (1) uniformly distributed in energy (assuming that each electron emitted into the cavity produces an equal detector response regardless of the energy of the emitted electron), (2) uniformly distributed in area on the surface of the cavity (assuming that each electron emitted into the cavity produces an equal detector response regardless of the location of the emitted electron), and (3) cosine-law distributed in angle relative to the local surface normal. There are two practical difficulties in this sampling scheme: (1) the source imposes a discontinuity in the otherwise continuous adjoint flux distribution, and (2) it will be difficult to write a surface sampling algorithm that can cope with the many possible geometry constructions of a cavity. First, we will consider methods to diminish or eliminate the adjoint-flux discontinuity resulting from the source. Then, we will consider sampling algorithms that can produce uniform distributions of source particles on cavity surfaces without direct surface sampling.

The first difficulty (flux discontinuity) arises because the adaptivity algorithms that we have devised are especially sensitive to changes in the level of the flux. A large discontinuity (over many orders of magnitude) would require special exemption within the adaptivity algorithm. It is far easier to construct an alternative adjoint problem to accommodate the adaptivity. We must require that the alternative adjoint problem (1) does not fundamentally alter the particle importances that we wish to calculate and (2) will diminish the discontinuity sufficiently to prevent the adaptivity algorithm from attempting to resolve the flux gradient at the source. To accomplish this we can take advantage of the fact that the desired cavity emission calculations do not include electron transport within the cavity. Therefore, the electron importance within the cavity will not affect the biased forward simulation.

Since the electron importance with the cavity will not affect the biased calculation, we can assign whatever electron importance is required to remove the flux discontinuity at the cavity surface. Because the tally structure is adaptive and changing, the most automated way to accomplish this is to build the importance assignment into the particle flux tally (via the particle tracking) in the Monte Carlo calculation. To eliminate the discontinuity, the flux level must be tallied identically on both sides of the source. This could be accomplished by tracking particles (and tallying flux) going in both the sampled source direction and in the opposite direction. However, the pseudo-particle that is tracked in the opposite direction (i.e., into the cavity) is not consistent with the adjuncton source that we require and therefore must not contribute tallies outside of the cavity (where importances will affect the forward biasing scheme). Simply terminating these pseudo-particles when they reach the opposite side of the cavity would accomplish this but would introduce a flux discontinuity on the opposite side of the cavity. This

alternative flux discontinuity will be a smaller relative difference.[2] This may be sufficient for some adaptivity algorithms/goals. If the adaptivity algorithm/goals require that there be no flux discontinuity, then the pseudo-particle flux tally within the cavity must be modified, which could be accomplished by making the tally continuously decreases to a particle weight zero before reaching the opposite cavity wall, or by randomly terminating the pseudo-particle flux tally within the cavity to decrease the probability of a tally to zero before reaching the opposite cavity wall.

The second difficulty (arbitrary surface sampling) could be overcome by a variety of sampling algorithms and tricks. Given the desired source particle distribution, one algorithm appears to be clearly preferred as the most general alternative, relatively simple to write, and usually reasonably efficient. The desired distribution corresponds to a uniform isotropic flux distribution restricted to particles on the cavity surface and outwardly directed. A uniform isotropic flux of particles within a volume is easy to achieve: particles are sampled uniformly on the outer surface of the volume, the particle directions are sampled as a cosine-law relative to the local surface normal and uniform in azimuth, and particles are tracked without scattering through the volume. To achieve our desired source with a convex cavity, one need only (1) define a primitive volume surrounding the cavity, such as a bounding box or bounding sphere and (2) track the uniform-isotropic-flux particles until they intersect the cavity surface in an outward direction, at which point they have the desired characteristics in space and angle to be used as a source particle.

In the case of a concave cavity, a more complicated algorithm is required. There are many possible approaches to this problem. These are our thoughts on a few possible approaches, but we are not prepared to advocate one. With a concave cavity, the uniform-isotropic-flux particle may have more than one outward intersection with the cavity surfaces. Each source particle must be given the weight of one history in the adjoint Monte Carlo calculation, so it is not sufficient to simply initiate a source particle for each intersection. A simple solution would be to store each additional intersection to be used as the source particle in subsequent Monte Carlo histories, but the downside is that since these intersections were not sampled independently, undesirable correlations would be introduced between the Monte Carlo histories. One could devise a simple rejection algorithm to properly populate the surfaces of the cavity. Any line-of-intersection sample resulting in fewer than the maximum number of possible intersections would have the source particle accepted with a probably proportional to the number of intersections relative to the maximum number. For an accepted line-of-intersection, the source particle location would be chosen randomly (uniformly) among the outward intersection points. This rejection method could be very inefficient and

[2] This can be illustrated by considering a simple pure-void problem, where both the convex cavity and the surrounding geometry are all void – thus particles are not scattered. In such a case, the adjoint flux level inside the cavity is zero. Outside of the cavity, the flux decreases from some uniform level at the cavity boundary. In our alternative adjoint formulation, flux is tallied within the cavity, and the relative flux discontinuity at the cavity boundary must be decreased. For simple geometries, one can show that the resulting alternative discontinuity is a factor of 2. Since weight window settings should not be resolved by better than a factor of 2, this discontinuity is likely to be insignificant for an adaptivity scheme that is being used to initialize weight windows.

would require *a priori* knowledge of the maximum number of straight-line intersections possible with the cavity.

An alternative approach for processing the line-of-intersection information is to run all outward intersection points in a single history (or, equivalently, one of the points randomly chosen but with a source weight set at the number of outward intersections). In this case, adjustments must be made to the results normalization logic. For instance, (in ITS where batch statistics are used), batches could be made to have the same total starting history weight rather than the same number of histories, or the increased weights could count against the number of histories to be run in each batch. This should be a fair approach to obtaining mean particle importances. Such an approach would not be fair in analyzing results in a pulse-height-type tally (though ITS currently has no tallies of this kind in adjoint).

APPENDIX C. FUNCTIONAL-EXPANSION TRACK-LENGTH TALLIES

To make use of moment information for p-adaptivity or hp-adaptivity, it is necessary to tally the moments of distributions within the Monte Carlo code. For point estimators, such as interaction-based quantities, this is trivial. For track-length-based quantities, one could resort to randomly sampling tally points within the track. However, it is also possible to perform a more rigorous integration of the moment contributions of each track segment. Griesheimer has provided a derivation of equations for such tallies for 2D distributions in Ref. **Error! Reference source not found.** and for 3D distributions in his dissertation**Error! Reference source not found.**. Here we provide a derivation of the 3D equations needed to tally Legendre moments of flux based on the track-length data used for tallying zone-averaged flux in ITS. (We believe that the only differences in our final formulation may be explained by a few typographical errors in Ref. **Error! Reference source not found.**.)

The true spatial-dependent flux within a cell can be represented as a combination of infinite series Legendre polynomial expansions in the three Cartesian dimensions,

$$\psi(x,y,z)=\sum_{l=0}^{\infty}\sum_{m=0}^{\infty}\sum_{n=0}^{\infty}\bar{a}_{n,m,l}P_n(x)P_m(y)P_l(z)\,. \tag{26}$$

In a simulation, the contribution of each track segment is added into an estimate of the expansion coefficient,

$$\bar{a}_{n,m,l}=\frac{\frac{1}{J}\sum_{j=1}^{J}\sum_{k=1}^{K_j}w_k\int\int\int P_n(x)P_m(y)P_l(z)\;\; f_k(x,y,z)\;dx\;dy\;dz}{\int\int\int dx\;dy\;dz}, \tag{27}$$

where J is the number of histories, K_j is the number of track segments in the cell for history j, and w_k is the weight of the particle for track segment k. The $\bar{a}_{n,m,l}$ coefficient is averaged over many histories. For each track segment, the contribution to this coefficient is

$$a_{n,m,l}=\frac{\int\int\int P_n(x)P_m(y)P_l(z)\;\; f_k(x,y,z)\;dx\;dy\;dz}{\int\int\int dx\;dy\;dz}. \tag{28}$$

In ITS, we tally only the zero-th moment flux, $\bar{a}_{0,0,0}$, as

$$\bar{a}_{0,0,0}=\frac{1}{VJ}\sum_{j=1}^{J}\sum_{k=1}^{K_j}w_k\int\int\int P_0(x)P_0(y)P_0(z)\;\; f_k(x,y,z)\;dx\;dy\;dz, \tag{29}$$

where V is the volume of the cell. Since $P_0(x) = 1$, this simplifies to

$$\bar{a}_{0,0,0}=\frac{1}{VJ}\sum_{j=1}^{J}\sum_{k=1}^{K_j}w_k\int\int\int f_k(x,y,z)\;dx\;dy\;dz. \tag{30}$$

Since each track is finite and evenly-weighted along some segment of track length, T, within the cell, this simplifies to

$$\bar{a}_{0,0,0} = \frac{1}{VJ}\sum_{j=1}^{J}\sum_{k=1}^{K_j} w_k T_k = \frac{1}{J}\sum_{j=1}^{J}\sum_{k=1}^{K_j} w_k a_{0,0,0}. \quad (31)$$

We want to evaluate arbitrary Legendre polynomial moments in a three-dimensional problem for each track length tallied. Spanier discusses this for 1D tallies**Error! Reference source not found.**.

Let $f(\vec{r})$ be the functional form of the volume flux estimator. Its magnitude will be in units of number/cm^2, such that the 3D volume integral is equal to the track *T*. The track *T* is located within the cell bounded by $(x_{\min}, y_{\min}, z_{\min})$ and $(x_{\max}, y_{\max}, z_{\max})$, with a starting point at (x_0, y_0, z_0) and direction cosines (w_x, w_y, w_z). (This assumes that the tally cell is an axis-aligned box. It is possible to use this approach to tally the distribution in any three dimensional shape, provided it is mapped onto a three-dimensional axis-aligned cube.) Not all direction cosines can be zero; for now, we choose *x* as a direction with non-zero direction cosine w_x, but we will generalize the result, and we will choose to evaluate the integral based on the largest direction cosine. The track runs from (x_0, y_0, z_0) to $(x_0 + Tw_x, y_0 + Tw_y, z_0 + Tw_z)$.

$f(\vec{r})$ may be written as the product of two delta functions. Due to the coupling of arguments of the delta functions, each restricts the non-zero values to a plane, while the product restricts the non-zero values to lie on the line of their intersection.

$$f(\vec{r}) = w_x\delta((x-x_0)w_y - (y-y_0)w_x)\delta((x-x_0)w_z - (z-z_0)w_x),$$
$$\text{where } \vec{r} = (x,y,z), w_x \neq 0, x = x_0 + sw_x, y = y_0 + sw_y, z = z_0 + sw_z, 0 \leq s \leq T. \quad (32)$$

Note that the dimensions of a delta function are the inverse units of its argument, so indeed $f(\vec{r})$ has the correct dimensions. Also note that $f(\vec{r})$ always is the product of two delta functions, even when one or two of the direction cosines are zero. The above expression holds provided $w_x \neq 0$. The coefficient in front of the delta functions is to retrieve total track length *T* when integrated over volume. To verify the value of the integral:

$$\frac{\int_{x_{\min}}^{x_{\max}} dx \int_{y_{\min}}^{y_{\max}} dy \int_{z_{\min}}^{z_{\max}} dz\ f(\vec{r})}{\int_{x_{\min}}^{x_{\max}} dx \int_{y_{\min}}^{y_{\max}} dy \int_{z_{\min}}^{z_{\max}} dz} = \frac{w_x}{V} \int_{x_{\min}}^{x_{\max}} dx \int_{y_{\min}}^{y_{\max}} dy\, \delta((x-x_0)w_y - (y-y_0)w_x)$$

$$\int_{z_{\min}}^{z_{\max}} dz \delta((x-x_0)w_z - (z-z_0)w_x)$$

$$= \frac{w_x}{V} \int_{x_0}^{x_0+Tw_x} dx \int_{y_0}^{y_0+Tw_y} dy\, \delta((x-x_0)w_y - (y-y_0)w_x)$$

$$\int_{z_0}^{z_0+Tw_z} dz \delta((x-x_0)w_z - (z-z_0)w_x) \qquad (33)$$

$$= \frac{w_x}{V} \int_{x_0}^{x_0+Tw_x} dx \int_{y_0}^{y_0+Tw_y} dy\, \delta((x-x_0)w_y - (y-y_0)w_x)\, \frac{1}{w_x}$$

$$= \frac{w_x}{V} \int_{x_0}^{x_0+Tw_x} dx\, \frac{1}{w_x} \frac{1}{w_x}$$

$$= \frac{w_x}{V}\ Tw_x\ \frac{1}{w_x} \frac{1}{w_x}$$

$$= \frac{T}{V}$$

To define expansion coefficients for Legendre polynomials, which are only defined over [-1,1], we first map the lab frame cell to $[-1,1]^3$.

For each coordinate, define primed, dimensionless coordinates as

$$x' = \frac{\left(x - \left(x_{\max} + x_{\min}\right)/2\right)}{\left(x_{\max} - x_{\min}\right)/2} \equiv \frac{\left(x - av_x\right)}{d_x}. \qquad (34)$$

For convenience we have also defined

$$x'_0 = \frac{(x_0 - av_x)}{d_x}, \qquad (35)$$

which gives us

$$f(\vec{r}) = \delta((x'-x'_0)d_x w_y - (y'-y'_0)d_y w_x)\delta((x'-x'_0)d_x w_z - (z'-z'_0)d_z w_z) \equiv f(x', y', z'). \qquad (36)$$

Now we can define the expansion coefficients of the Legendre polynomials:

$$a_{n,m,l} = \int_{-1}^{1} dx' \int_{-1}^{1} dy' \int_{-1}^{1} dz'\, P_n(x') P_m(y') P_l(z') f\left(x', y', z'\right). \qquad (37)$$

To evaluate the integral over *dz*', we again use the property of delta functions $\delta(c \cdot x) = \frac{\delta(x)}{c}$ to obtain

$$\delta((x'-x'_0)d_x w_z-(z'-z'_0)d_z w_x)=\delta(d_z w_x\cdot(\frac{d_x w_z}{d_z w_x}(x'-x'_0)-(z'-z'_0)))$$
$$=\frac{1}{d_z w_x}\delta(z'-z'_0-\frac{d_x w_z}{d_z w_x}(x'-x'_0)) \quad (38)$$

This gives us

$$a_{n,m,l}=\frac{1}{d_z d_y w_x}\int_{x'_0}^{x'_0+Tw_x/d_x} dx' P_n(x')P_m(y'_0+\frac{d_x w_y}{d_y w_x}(x'-x'_0))P_l(z'_0+\frac{d_x w_z}{d_z w_x}(x'-x'_0)). \quad (39)$$

Note the integration limits over *dx*' are restricted over the length of track *T*.

To evaluate the above, we use Gaussian quadrature of order (*N*+*M*+*L*+1)/2 (where *N* is the largest value of *n*, etc.). First we map the limits of integration to [-1,1] through

$$x''=\frac{(x'-x'_0-Tw_x/2d_x)}{Tw_x/2d_x}, \quad (40)$$

which gives

$$a_{n,m,l}=\frac{T}{(2d_z d_y d_x)}\int_{-1}^{1} dx'' P_n(x'')P_m(y'')P_l(z''), \quad (41)$$

where

$$y''=y'_0+\frac{Tw_y}{2d_y}(x''+1)=\frac{1}{d_y}\left(y_0-y_{ave}+\frac{w_y}{w_x}(x-x_0)\right), \quad (42)$$

$$z''=z'_0+\frac{Tw_z}{2d_z}(x''+1)=\frac{1}{d_z}\left(z_0-z_{ave}+\frac{w_z}{w_x}(x-x_0)\right). \quad (43)$$

Note that this agrees with our current track-length flux estimator in ITS, since $a_{0,0,0}$ is tallied as T/V for each track segment. This more general method allows us to evaluate all of the $a_{n,m,l}$'s.

VII. BIBLIOGRAPHY

Error! Reference source not found.

ELECTRONIC DISTRIBUTION:

2 University of Wisconsin
Attn: Tim Tautges
Jason Kraftcheck
1500 Engineering Dr.
Madison, WI 53706

1 MS 1179 B. C. Franke, 1341
1 MS 1179 R. P. Kensek, 1341
1 MS 1179 M. J. Crawford, 1341
1 MS 1179 W. C. Fan, 1341
1 MS 1179 T. W. Laub, 1341
1 MS 1179 L. J. Lorence, 1341
1 MS 0899 Technical Library, 9536
1 MS 0188 D. Chavez, LDRD Office, 1011


Sandia National Laboratories