# A$^3$MCNP: Automatic Adjoint Accelerated MCNP - USER'S MANUAL (Version 1.0i)

**J.C. Wagner**
Oak Ridge National Laboratory
P.O. Box 2008
Oak Ridge, TN 37831-6370
Phone: 865 241-3570
FAX: 865 576-3513
Email: wagnerjc@ornl.gov
URL: http://www.ornl.gov/~6cw

**A. Haghighat**
The Pennsylvania State University
Department of Nuclear & Mechanical Engineering
231 Sackett Building
University Park, PA 16802
Phone: 814 865-1341
FAX: 814 865-8499
Email: haghighat@psu.edu

**Date: May 2000**

# Contents

# List of Tables

# List of Figures

## Chapter 1

# INTRODUCTION

For numerous reasons, the Monte Carlo (MC) method has become the method of choice for performing complex shielding analyses, particularly where very accurate results are necessary. The merits of using a computer code based on the MC method, such as accurate modeling of complex geometries and the utilization of continuous-energy cross-section data, are well known. However, there are also significant difficulties associated with the MC method, such as the computational time required to achieve statistically meaningful results and the time and effort associated with the implementation of variance reduction techniques.

The use of variance reduction techniques is further complicated by the fact that improper use can lead to incorrect results, which, depending on the user's experience, may or may not be apparent to the user. To overcome these difficulties, the A$^3$MCNP - Automatic Adjoint Accelerated *MCNP* code[1, 2, 3, 4] has been developed. A$^3$MCNP is an enhanced version of the widely-used, general-purpose MCNP code[5] that has been modified to automatically prepare and utilize parameters for source and transport biasing based on the $S_N$ adjoint function.

A$^3$MCNP utilizes a new methodology, CADIS (Consistent Adjoint Driven Importance Sampling)[1, 3], for using the $S_N$ adjoint function for automatic variance reduction of MC calculations through source biasing and consistent transport biasing with the weight window technique. Additionally, A$^3$MCNP prepares the necessary input files for performing multi-group, three-dimensional adjoint $S_N$ calculations using TORT[6]. For this task, A$^3$MCNP prepares a mesh distribution and the corresponding mixtures and their identification numbers and densities. Where possible, the information is extracted from the normal MCNP input; requiring relatively few additional input cards. Upon completion of the adjoint $S_N$ calculation, A$^3$MCNP (1) reads the adjoint function, variable spatial mesh, and energy group boundaries from the standard TORT output file, (2) superimposes the variable spatial mesh and energy grid onto the MCNP problem, (3) couples the original source distributions with the adjoint function to generate dependent source biasing parameters and weight window lower bounds, and (4) performs the transport calculation using the superimposed grids and calculated parameters. The grids facilitate the use of the detailed space- and

energy-dependent importance function and do not impose any limitations on the transport of particles.

$A^3$MCNP has been used for the simulation of a few-real life problems, including a PWR pressure vessel and cavity dosimeter[7, 8], a BWR core shroud[4], and spent fuel shipping and storage casks[1, 9].

This manual is organized as follows: the development and implementation of the CADIS methodology is described in Chapter 2. Chapter 3 presents methodologies for the automatic generation of input files for $S_N$ adjoint calculations from the MCNP problem description, including generation of a deterministic spatial mesh, processing of multigroup material cross sections, and the appropriate definition of the remaining required parameters. Chapter 4 describes the $A^3$MCNP input and execution procedure. An example application of $A^3$MCNP is described in Chapter 5. Finally, Chapter 6 briefly lists some of the enhancements that are planned for the near future.

## Chapter 2

# VARIANCE REDUCTION VIA THE ADJOINT FUNCTION

### 2.1 Theory

Problems that can be solved by the MC method are essentially equivalent to integrations[10]. For example, the goal of most MC particle transport problems is to calculate the response (i.e., flux, dose, reaction rate, etc.) at some location. This is equivalent to solving the following integral

$$R = \int_P \Psi(P)\sigma_d(P)dP, \tag{2.1}$$

where $\Psi$ is the particle flux and $\sigma_d$ is some objective function in phase space $(\underline{r}, E, \hat{\Omega}) \in P$.

From the adjoint identity,[11]

$$\langle \Psi^\dagger L\Psi \rangle = \langle \Psi L^\dagger \Psi^\dagger \rangle, \tag{2.2}$$

where $(\dagger)$ denotes adjoint, it can be shown that the response $R$ at some location is also given by

$$R = \int_P \Psi^\dagger(P)q(P)dP, \tag{2.3}$$

where $\Psi^\dagger$ and $q$ are the adjoint function and source density, respectively, and Eqs. 2.1 and 2.3 are equivalent expressions for $R$. The function $\Psi^\dagger(P)$ has physical meaning as the expected contribution to the response $R$ from a particle in phase space $P$, or in other words, the importance of a particle to the response.

To solve this integral with the MC method the independent variables are sampled from $q(P)$, which is not necessarily the best probability density function (*pdf*) from which to sample. An alternative *pdf*, $\hat{q}(P)$ can be introduced into the integral as follows:

$$R = \int_P \left[\frac{\Psi^\dagger(P)q(P)}{\hat{q}(P)}\right]\hat{q}(P)dP, \tag{2.4}$$

where $\hat{q}(P) \geq 0$ and $\int_P \hat{q}(P)dP = 1$.

From importance sampling[12, 13], the alternative *pdf* $\hat{q}(P)$ that will minimize the variance for $R$ is then given by

$$\hat{q}(P) \;=\; \frac{\Psi^{\dagger}(P)q(P)}{\int_P \Psi^{\dagger}(P)q(P)dP}. \tag{2.5}$$

If the final result $R$ is known, then the MC integration will return $R$ with zero variance. However, in practice, the adjoint function is not known exactly, $R$ cannot be solved by direct integration, and thus, it is necessary to simulate the particle transport. For this process it is desirable to use the biased source distribution in Eq. 2.5 that, in the limit of an exact adjoint, leads to a zero variance solution.

Examining Eq. 2.5 reveals that the numerator is the detector response from phase-space point $P$, and the denominator is the total detector response $R$. Therefore, the ratio is a measure of the contribution from phase-space $P$ to the detector response. Intuitively, it is useful to bias the sampling of source particles by the ratio of their contribution to the detector response, and therefore, this expression could also be derived from physical arguments.

Since the source variables are sampled from a biased *pdf*, the statistical weight of the source particles must be adjusted using the following "conservation" formula:

$$w\ (biased\ pdf) \;=\; w_o\ (unbiased\ pdf) \tag{2.6}$$

where $w_o$ is the weight before the variance reduction technique is applied, such that

$$W(P)\ \hat{q}(P) \;=\; W_o\ q(P), \tag{2.7}$$

where $W_o$ is the unbiased particle starting weight, which is set equal to 1. Substituting Eq. 2.5 into Eq. 2.7 and rearranging, we obtain the following expression for the statistical weight of the particles

$$W(P) \;=\; \frac{\int_P \Psi^{\dagger}(P)q(P)dP}{\Psi^{\dagger}(P)} \;=\; \frac{R}{\Psi^{\dagger}(P)}. \tag{2.8}$$

This equation shows an inverse relationship between the adjoint (importance) function and the statistical weight. Previous work[14] in this area assumed this relationship and showed it to be near optimal, and others have verified this relationship through computational analysis[15, 16]. However, in this work, beginning with importance sampling, this relationship has been derived.

To consider the transport process, we examine the integral Boltzmann transport equation for particle density in the phase space $P$, given by

$$\Psi(P) \;=\; \int K(P' \rightarrow P)\Psi(P')dP' \;+\; q(P), \tag{2.9}$$

where $K(P' \rightarrow P)dP$ is the expected number of particles emerging in $dP$ about $P$ from an event in $P'$ and $q(P)$ is the source density. To transform Eq. 2.9 to be in terms of the biased source distribution $\hat{q}(P)$, we multiply it by

$$\frac{\Psi^\dagger(P)}{\int \Psi^\dagger(P)q(P)dP}, \tag{2.10}$$

and define

$$\hat{\Psi}(P) \;=\; \frac{\Psi(P)\Psi^\dagger(P)}{\int \Psi^\dagger(P)q(P)dP}, \tag{2.11}$$

to yield the following transformed equation

$$\hat{\Psi}(P) \;=\; \int K(P' \rightarrow P)\Psi(P')dP' \frac{\Psi^\dagger(P)}{\int \Psi^\dagger(P)q(P)dP} \;+\; \hat{q}(P), \tag{2.12}$$

or

$$\hat{\Psi}(P) \;=\; \int K(P' \rightarrow P)\hat{\Psi}(P')(\frac{\Psi^\dagger(P)}{\Psi^\dagger(P')})dP' \;+\; \hat{q}(P). \tag{2.13}$$

This transformed equation can be written as

$$\hat{\Psi}(P) \;=\; \int \hat{K}(P' \rightarrow P)\hat{\Psi}(P')dP' \;+\; \hat{q}(P), \tag{2.14}$$

where

$$\hat{K}(P' \rightarrow P) = \; K(P' \rightarrow P)(\frac{\Psi^\dagger(P)}{\Psi^\dagger(P')}). \tag{2.15}$$

In this transformed equation, the number of particles emerging in $P$ from an event in $P'$ is being altered by the ratio $\frac{\Psi^\dagger(P)}{\Psi^\dagger(P')}$, which is the ratio of importances. This adjustment to the transfer kernel can be accomplished through particle creation and termination, such that:

$$for\ \frac{\Psi^\dagger(P)}{\Psi^\dagger(P')} \;>\; 1 \quad particles\ are\ created\ (splitting), \tag{2.16}$$

and

$$for\ \frac{\Psi^\dagger(P)}{\Psi^\dagger(P')} \;<\; 1 \quad particles\ are\ destroyed\ (roulette). \tag{2.17}$$

Since we are altering the number of particles emerging from an event, the statistical weight of the particles must be corrected according to the conservation relation of Eq. 2.6, such that

$$W(P)K(P' \rightarrow P)(\frac{\Psi^\dagger(P)}{\Psi^\dagger(P')}) \;=\; W(P')K(P' \rightarrow P) \tag{2.18}$$

or

$$W(P) \;=\; W(P')\ \frac{\Psi^\dagger(P')}{\Psi^\dagger(P)}. \tag{2.19}$$

While the development of the equations is based on the concept of zero variance, a zero variance cannot be attained with estimation at particle events (e.g., collision, boundary

crossings, etc.) because the number of events is itself a random variable and contributes to the variance of the final result. However, minimum variance (even zero-variance solutions in the limit) can be achieved when every sampling (source and transport) is made proportional to its importance.

To administer the splitting and rouletting of particles, the weight window facilities that are available within MCNP, which deal with particle weights, are used. We have related these weights to particle importance via Eqs. 2.8 and 2.19. Since these relationships for the particle statistical weights, which are used in source sampling and the particle transport process, were derived from importance sampling in a consistent manner, the use of the relations is referred to as Consistent Adjoint Driven Importance Sampling (CADIS).

## 2.2 Implementation into MCNP

In the previous section, expressions for source biasing parameters and particle statistical weights were defined based on the adjoint (importance) function. In this section, we describe how this information is used within MCNP and the related difficulties and issues.

### 2.2.1 Calculation of Variance Reduction Parameters

To calculate source biasing parameters over the phase-space (space, energy, and angle) the source from the forward calculation is coupled with the adjoint function as shown in Eq. 2.5. Further, the particle transport is biased via Eqs. 2.8 and 2.19.

The space, energy, and angular dependent adjoint function may require a significant amount of storage, particularly for large 3-D problems. For example, the adjoint function for a 3-D problem with 100×100×100 spatial meshes, 50 energy groups, and 80 directions ($S_8$) is 4E+09 values that, for double precision, require 32 gigabytes of storage. The $S_N$ method can determine the angular independent (or scalar) adjoint accurately, but not necessarily the angular dependent adjoint because of the limited number of directions. Therefore, because of the memory requirements and inaccuracies of the angular dependent adjoint, we use the space and energy dependent (scalar) adjoint function

$$\phi^{\dagger}(r,E) = \int_{4\pi} \Psi^{\dagger}(r,E,\hat{\Omega}) d\hat{\Omega} \tag{2.20}$$

for calculating space and energy dependent source biasing and weight window parameters. It should be noted, however, that the use of a less accurate adjoint (importance) function may reduce the efficiency (with respect to that from a very accurate adjoint function), but does not impact the accuracy of the MC result.

### Source biasing

Source biasing allows the simulation of a larger number of source particles, with appropriately reduced weights, in the more important regions of each variable (e.g., space, energy, and angle). This technique consists of sampling the source from a biased (non-analog) probability distribution rather than from the true (analog) probability distribution, and then correcting the weight of the source particles by the ratio of the actual probability divided by the biased probability according to Eq. 2.6. Thus, the total weight of particles started in any given interval is conserved, and an unbiased estimate is preserved.

To accelerate the MC calculation the source energy and position are sampled from the biased source distribution $\hat{q}(r, E)$:

$$\hat{q}(r, E) \;=\; \frac{\phi^{\dagger}(r, E) q(r, E)}{\int_V \int_E \phi^{\dagger}(r, E) q(r, E) dr dE} \;=\; \frac{\phi^{\dagger}(r, E) q(r, E)}{R}. \tag{2.21}$$

Physically, the numerator is the detector response from space-energy element $(dr, dE)$, and the denominator is the total detector response $R$. Therefore, the ratio is a measure of the relative contribution to the detector response.

In order to calculate the source biasing parameters, it is necessary to couple the $S_N$ adjoint function and the forward Monte Carlo problem description. The main difficulty lies in the fact that MCNP offers a great deal of flexibility in the way the source spatial distribution can be defined. The present version of $A^3$MCNP is capable of calculating biased source distributions and weight window lower bounds properly for point, surface, and volume sources defined by points.

### Transport biasing

As mentioned, the weight window technique, as implemented in the MCNP code, is a space- and energy-dependent facility by which splitting/roulette are applied. The weight window technique provides an alternative to geometric splitting/roulette and energy splitting/roulette for assigning space- and energy-dependent importances. To use the weight window facility within MCNP, we need to calculate weight window lower bounds $W_l$ such that the statistical weights defined in (Eq. 2.8) are at the center of the weight windows (intervals). The width of the interval is controlled by the parameter $C_u$, which is the ratio of upper and lower weight window values ($C_u = \frac{W_u}{W_l}$). Therefore, the space and energy dependent weight window lower bounds $W_l$ are given by

$$W_l(r, E) \;=\; \frac{W}{(\frac{C_u+1}{2})} \;=\; \frac{R}{\phi^{\dagger}(r, E)} \frac{1}{(\frac{C_u+1}{2})}, \tag{2.22}$$

and during the transport process the weight window technique performs splitting or roulette according to Eq. 2.19. In MCNP, the default value for $C_u$ is 5.

It is important to note that because the source biasing parameters and weight window lower bounds are consistent, the statistical weights of the source particles ($W(r,E) = \frac{q(r,E)}{\hat{q}(r,E)}$) are within the weight windows as desired. Moreover, if the statistical weights of the source particles are not within the weight windows, the particles will immediately be split or rouletted in an effort to bring their weights into the weight windows[5]. This will result in unnecessary splitting/rouletting and a corresponding degradation in computational efficiency. For problems in which the adjoint function varies significantly within the source region (space and/or energy), this coupling between source and transport biasing is critical.

### 2.2.2 Integration of Importance Function into MCNP

The general version of MCNP provides facilities for energy and cell dependent weight windows. This means that in order to use a fine spatial weight window grid (which is necessary in optically thick regions) with the standard version of MCNP, the user must subdivide the MCNP cell based geometry such that the ratio of importances between adjacent geometric cells is not too large. Because the importance ratios are not apriori known, this geometric discretization is not straightforward and typically requires iterations of manual adjustments. Further, the subdivision of the geometry into a very large number of cells is time consuming and can actually degrade the efficiency of the calculation. For these reasons, we use the deterministic $S_N$ spatial mesh description to construct a separate, but related, geometric grid to facilitate the use of the adjoint distribution. A$^3$MCNP is able to read the binary flux file from the standard $S_N$ TORT[6] code (which contains the adjoint function and the spatial mesh and energy group information) and superimpose the variable spatial mesh and energy grid onto the standard MC problem in a manner transparent to the user. This grid enables the use of the spatial and energy dependent importance function, and does not directly affect the transport of particles. At various events in a particle history (e.g., collisions, surface crossings, and/or increments of mean free path), the grid is searched to determine the importance of the phase-space within which the particle resides. The importance is then compared to the statistical weight of the particle and the appropriate action is taken (e.g., splitting, Russian roulette, or no action).

Currently, the level of detail of the energy-dependent importance function is dictated by the multigroup library used for the $S_N$ adjoint calculation (i.e., all groups are used).

### 2.2.3 Weight Checking

Various concepts for minimizing the amount of computational *overhead* associated with this process have been examined. The first issue of concern is the determination of the appropriate occasion (or event) to check the particle's statistical weight. Because the MCNP geometry does not need to be manually subdivided to assign the spatial importances, the

presently available weight checks (i.e., at collisions and surface crossings) are no longer sufficient to control particle weight, and thus, large differences in the weight scored by individual particles are possible. Additional (more frequent) weight checking has two opposing effects: (1) there is a computational cost or penalty each time the weight is checked, and this penalty is the time required by the searching routines to determine the importance of the phase-space within which the particle resides and (2) more frequent checking leads to more reliable results with well-behaved statistical convergence. Therefore, it is clear that a criterion for an optimum or near-optimum compromise for checking particle statistical weights is needed. Moreover, it is desirable that this criterion be problem independent.

In deterministic methods the spatial domain of the problem is discretized into relatively fine spatial meshes to enable the approximation of spatial derivatives with finite differences. Thus, the spatial meshes must be small enough to allow this approximation (i.e., the particle density must not vary significantly within a mesh cell). Because the particle density is directly related to the material cross sections, and the corresponding mean free path, mfp (the average distance a particle travels between collisions), it is common practice to use mesh sizes on the order of one mfp to ensure that large variations do not occur and that the aforementioned approximation is valid.

Analogously, in MC methods the particle statistical weight has been related to the adjoint function (Eq. 2.8), which is also directly related to the material cross sections or mfp. Further, since the mfp is, by definition, the average distance a particle travels between collisions, it is a logical, problem independent parameter by which particle statistical weight can be controlled. Therefore, during particle transport, the distance to collision is determined as before, but this distance is now compared to the mfp. If the distance to collision exceeds the user specified mfp increment, the particle is transported the distance of that increment and the statistical weight is compared to the weight window boundaries for that region. Parametric studies[1] analyzing the effect of the increment of mfp on problem efficiency and reliability support the A$^3$MCNP default value of one mfp for weight checking.

The second issue of concern is the amount of time associated with checking the particle's statistical weight. The computational penalty in the MC calculation for using larger numbers of spatial meshes or energy groups is related to the search routine. For the binary search (which is currently being used), the average number of comparisons in a successful search, assuming that each of the N intervals is equally likely, is a slowly increasing function (i.e. $\propto log_2$) of the number of intervals.[17]

## Chapter 3

# AUTOMATION OF ADJOINT $S_N$ CALCULATIONS

In this chapter the strategies for generating input files for $S_N$ calculations directly from the MCNP problem description are briefly described. The *automation* of the generation of $S_N$ input files is intended to eliminate the tedious process of manually generating these files and require very little experience and effort on the part of the user.

### 3.1 Available Codes and Data

To calculate an adjoint function, the following codes and data are necessary: an $S_N$ transport code, a cross-section mixing/processing code, and an appropriate multigroup cross-section library. Because there are a number of publicly available codes and cross-section libraries that are acceptable for this work, new codes were not created. The current version of $A^3$MCNP uses the three-dimensional $S_N$ TORT code[6] for the adjoint transport calculation and the GIP code[18] to mix/process the multigroup cross-section data into macroscopic cross-section mixtures prior to performing the transport calculation.

In this work, the adjoint function is used for variance reduction of MC calculations (from which the final (correct) answer is sought), and thus it is not necessary to solve the adjoint problem with a very high degree of accuracy. Consequently, the choice of the multigroup library is not as critical as it is for a direct (forward) calculation. Studies[3] have shown that a relatively few group collapsed adjoint ($\sim$2-5 groups) is capable of increasing the calculational efficiency to approximately half of the observed maximum. Therefore, a multigroup library should be selected (by the user) based on the following criteria: problem applicability, accuracy, memory/disk space requirements, and CPU time. In the current version of $A^3$MCNP, it is assumed that the user chooses an appropriate multigroup cross-section library (i.e., the user has the flexibility to use various multigroup cross-section libraries).

## 3.2 Automatic Input Generation for $S_N$ Calculations

Automatic variance reduction of a MC calculation with an $S_N$ adjoint function requires the generation of an input file for a $S_N$ adjoint calculation and the generation of an input file for a cross-section mixing code. Hence, in this section strategies are described for the automation of these tasks.

A MC (MCNP) model or input file describes a particular problem in terms of combinatorial geometry and continuous energy, while a deterministic method requires discretization of the geometry and energy. Therefore, while the MC input file contains most of the information required to generate a corresponding deterministic input file, further processing beyond simple translation is necessary. Specifically, the MC geometry description must be appropriately discretized, a suitable energy group structure must be specified, the material cross sections must be processed, and various $S_N$ input parameters must be defined.

The task of $S_N$ input generation can be subdivided into the following four subtasks: (1) spatial mesh generation, (2) adjoint source specification, (3) appropriate assignment of remaining required input parameters, and (4) material cross-section preparation.

### 3.2.1 Mesh Generation/Projection

The MC geometry description must be discretized into a spatial mesh that is fine enough to adequately describe the material boundaries and enable an accurate deterministic calculation, while not being refined to the extent that the computational expense and/or disk space requirements for the deterministic calculation becomes prohibitive. It is not the intention of the adjoint calculation to solve the problem exactly, thus a compromise between accuracy and efficiency is required. Consequently, some approximations in the mesh generation/projection are acceptable.

The mesh generation approach in $A^3$MCNP[1] involves initially overlaying the entire problem with a fine spatial mesh, and then employ routines that currently exist in MCNP to assign materials to meshes based on mesh center coordinates. Based on the fine mesh material boundaries and the material mfps (which can be approximately determined with a short initial MCNP calculation) the fine meshes may be *back-thinned* (combined) into a coarser mesh description. The *back-thinning* approach takes full advantage of the discontinuous mesh feature of TORT.

This original approach to mesh generation takes full advantage of capabilities that currently exist in MCNP and, since meshes are not being explicitly fitted to geometric bodies, does not suffer from the limitations of current mesh generators. This mesh generator is applicable to any geometry that can be described within MCNP. However, geometries described by repeated structures are not currently allowed.

The current implementation is for three dimensional Cartesian geometry in a format

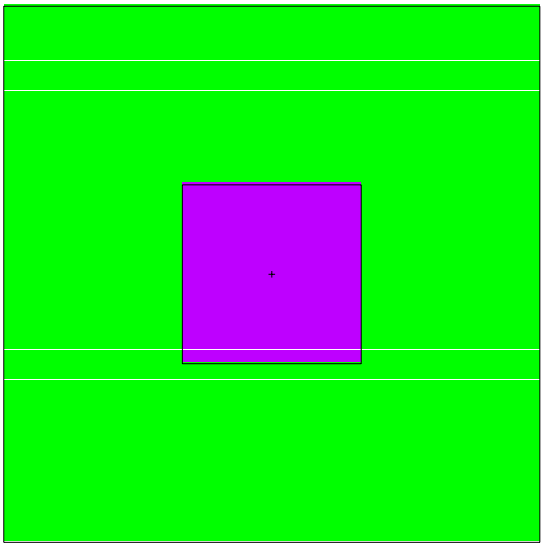

Figure 3.1: Mesh Generation Example 1: Box-In-A-Box

suitable to the 3-D $S_N$ code TORT[6]. The mesh generation technique is perhaps best understood with the assistance of a couple of simple examples.

**Mesh generation example 1: box-in-a-box**

The first example is a 4 cm cube centered within a 12 cm cube, and is depicted in Fig. 3.1. The mesh generation routine requires the user to supply nine parameters for the initial fine mesh, these include the bounding x, y, and z boundaries (6 values) and an x, y and z mesh thickness (3 values). For this example problem, the x, y and z upper and lower boundaries are all 0.0 and 12.0, respectively, and a mesh thickness of 1 cm is specified. From these values, a uniform fine mesh is generated over the range specified. Note that this approach to defining the initial fine mesh (i.e., requiring only nine values) was adopted because of its simplicity.

In the MC method, particles are tracked or followed through a problem. At each collision, it is necessary to determine where the particle is located in order to calculate the distance to the next collision and the distance to the next boundary. As a result, routines exist within MCNP that check the sense of a position with respect to the surfaces and associate the position to a cell. Each cell has an associated material. These existing routines are employed to assign materials to meshes based on mesh center coordinates. Therefore, a uniform mesh and material composition is specified for the entire problem.

For verification of the mesh generation capability and for assistance in checking the

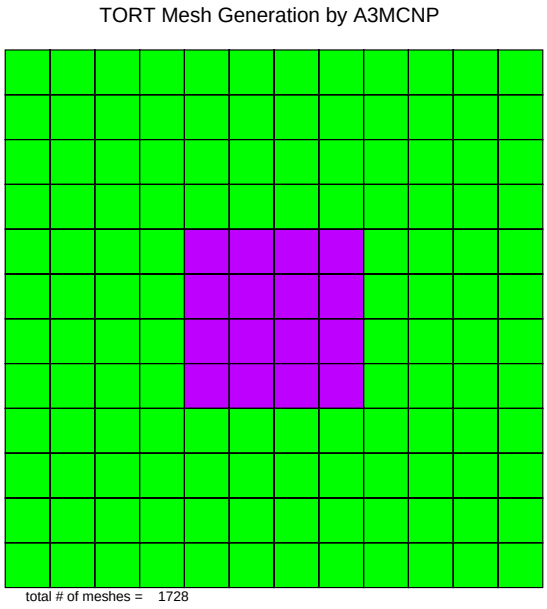


Figure 3.2: Uniform Generated Mesh for Example 1

quality of a generated mesh, it is necessary to be able to view the mesh. The mesh generation routine will generate a postscript file containing a 2-D view of the mesh for any requested axial (z) plane (or optionally, all axial z-planes). Figure 3.2 shows the mesh and material specification through the center of the cube as generated directly by the mesh generation routine. Note that the total number of meshes are given immediately below the geometry.

Although we have a fully specified geometry, in terms of mesh and materials, it may contain an unnecessarily large number of meshes. Because the number of spatial meshes is directly related to computer time and memory/disk space requirements for performing the $S_N$ calculation, it is desirable to minimize the number of meshes to whatever extent possible. The mesh boundaries are dictated by material composition (or mfp) and by material boundaries. These two criteria are used to remove unnecessary meshes via a process referred to as *back-thinning*. The user supplies a back-thinning parameter for each material, where this parameter is the maximum allowed thickness of any mesh within that material (e.g. a reasonable choice for this parameter is the material mfp). Then based on this parameter and the material boundaries, the fine meshes are combined (or thinned) where appropriate. To demonstrate this process with the example problem, assume that the mfp of the center material is 1 cm and the mfp of the outer material is 2 cm. Figure 3.3 shows the resulting mesh distribution. Note the original number of meshes and the number

of meshes after the back-thinning process. For this simple example, the number of meshes is reduced by 70%. The input file used to generate the mesh in Fig. 3.3 is provided in Fig. A.1 of Appendix A.

**Mesh generation example 2: Spheres-In-A-Box**

The second example is intended to illustrate the mesh generation capabilities for a combination of rectangular and curved bodies. The problem is characterized by two concentric spheres of radii 4 cm and 8 cm centered within a 20 cm cube, and is depicted in Fig. 3.4. For this example problem, the x,y, and z upper and lower boundaries are all -10.0 cm and 10.0 cm, respectively, and a mesh thickness of 0.5 cm is specified over the x and y ranges, and 1.0 cm over the z range. From these values, a uniform fine mesh is generated over the range specified, and is shown in Fig. 3.5. The fine mesh thickness (0.5 cm) is used to generate a mesh that very accurately represents the curved surfaces of the spheres.

To reduce the number of meshes, we once again invoke the back-thinning process. For this problem, assume that the mfp of the inner sphere, outer sphere, and outer material are 1, 1.5, and 2 cm, respectively. Fig. 3.6 shows the resulting mesh distribution. The 32,000 original meshes are reduced to 5,260 meshes, a reduction of nearly 84%.

It should be apparent that the quality of the mesh representation and number of final meshes is dependent on the selection of the initial fine mesh definition. If one employs a very fine initial mesh, the material boundaries will be represented very well, but the total number of final meshes, particularly for problems containing curved surfaces, may be large. On the other hand, if one employs a coarser fine initial mesh, the material meshes will not be represented as well, but the total number of final meshes will be less. In an attempt to demonstrate this behavior, we generate a new mesh for this example problem based on an initial fine mesh thickness of 1 cm for the x, y and z ranges. This mesh is shown in Fig. 3.7. Note that the previous uniform mesh, which is shown in Fig. 3.5, has a factor of 4 more meshes. Now, with the same back-thinning parameters the mesh is thinned to that shown in Fig. 3.8. While some accuracy in terms of the representation of material boundaries is lost, the final number of meshes (4316) is approximately 22% lower than that of our previously back thinned mesh (5260, see Fig. 3.6).

Thus, the generation of an optimum mesh (in terms of material representation and minimization of number of meshes) may require some experience with the mesh generator. However for our purposes, a reasonable, not optimum, mesh is sufficient, and can be generated quickly and easily.

### 3.2.2 Generation of $S_N$ Input Files

The remaining two subtasks necessary for automating the generation of $S_N$ input files are the specification of the adjoint source and the specification of the remaining required $S_N$ input parameters.

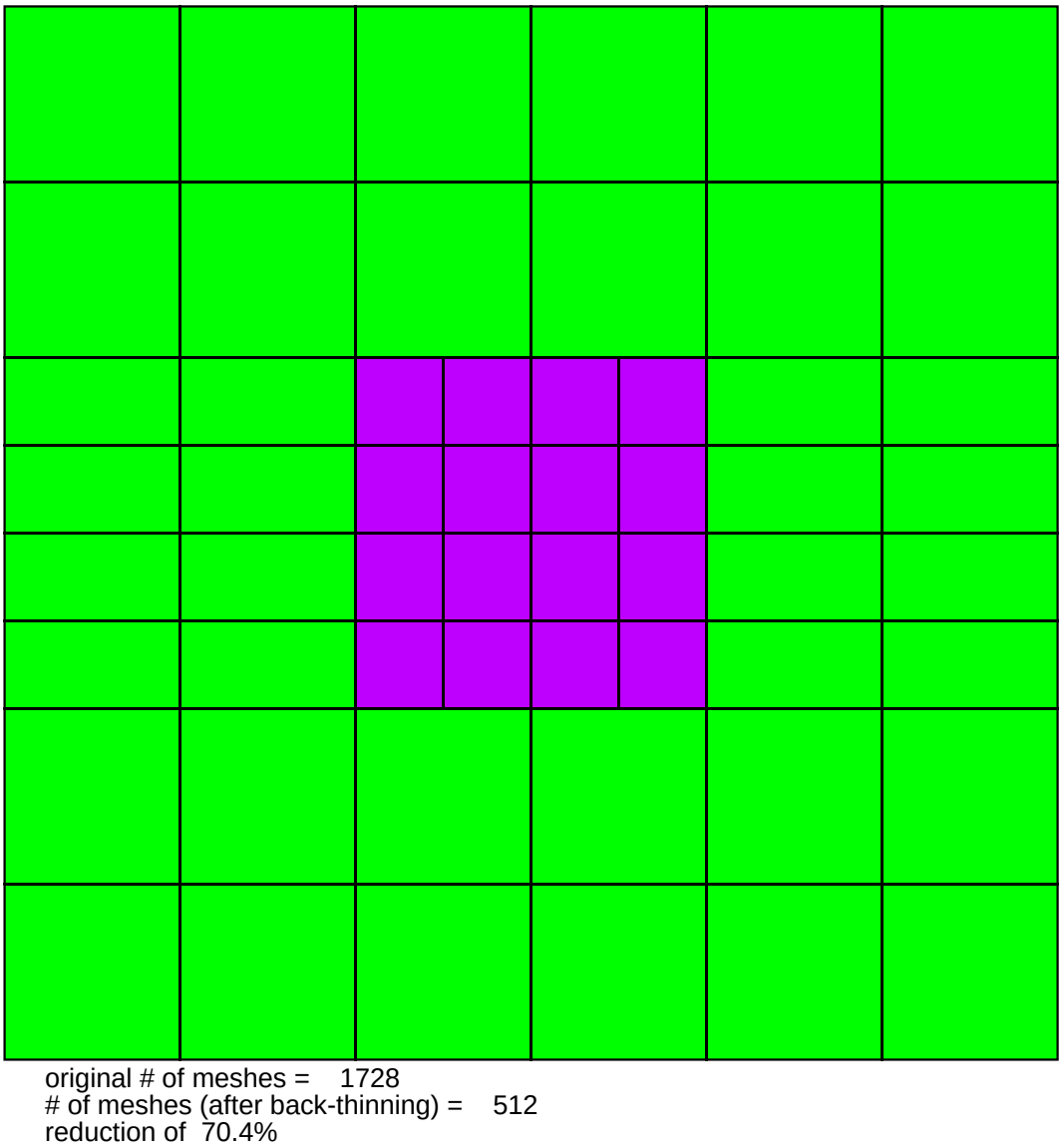


Figure 3.3: Back-Thinned Mesh for Example 1

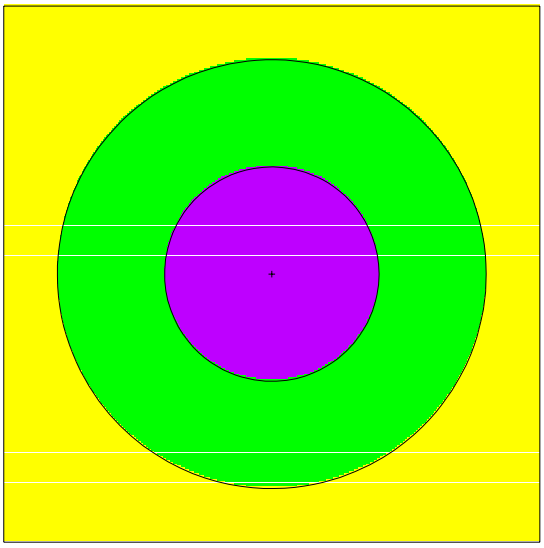

Figure 3.4: Mesh Generation Example 2: Spheres-In-A-Box

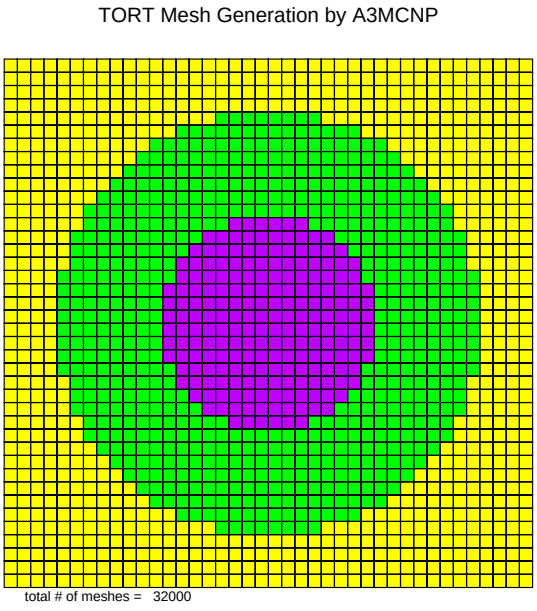


Figure 3.5: Uniform Generated Mesh for Example 2

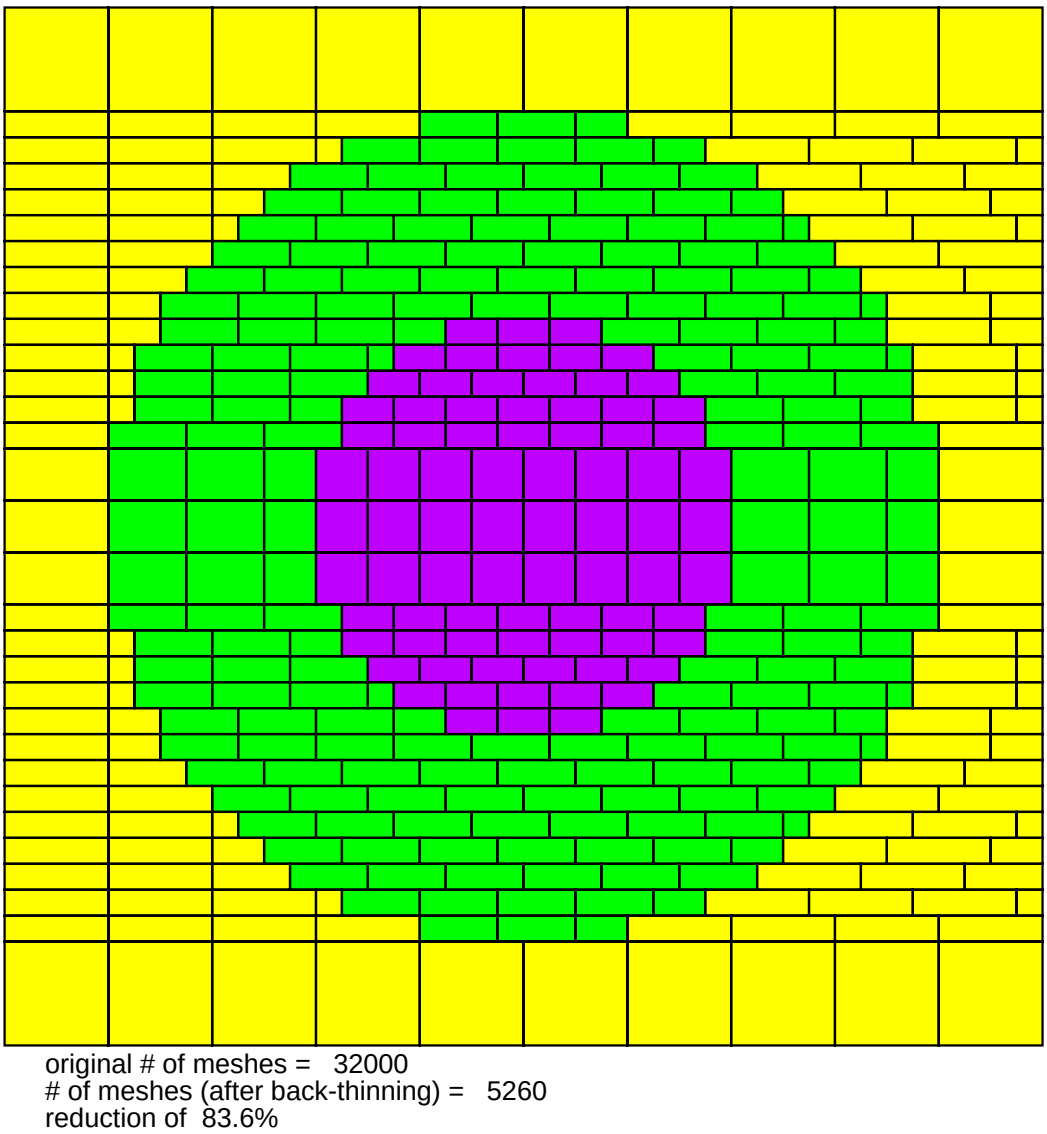


Figure 3.6: Back-Thinned Mesh for Example 2

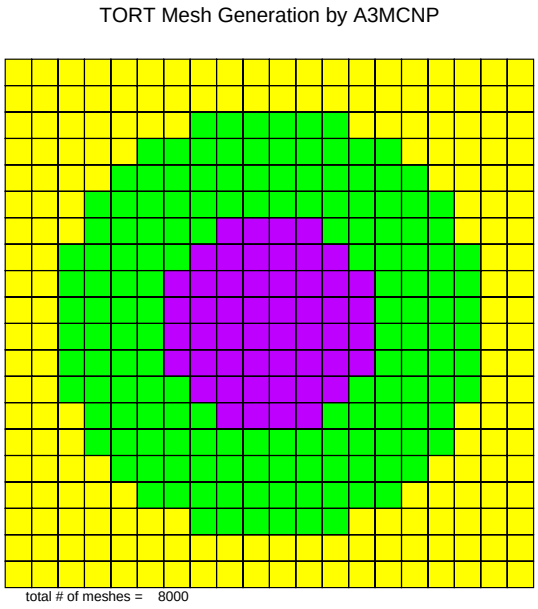


Figure 3.7: Alternative Uniform Generated Mesh for Example 2

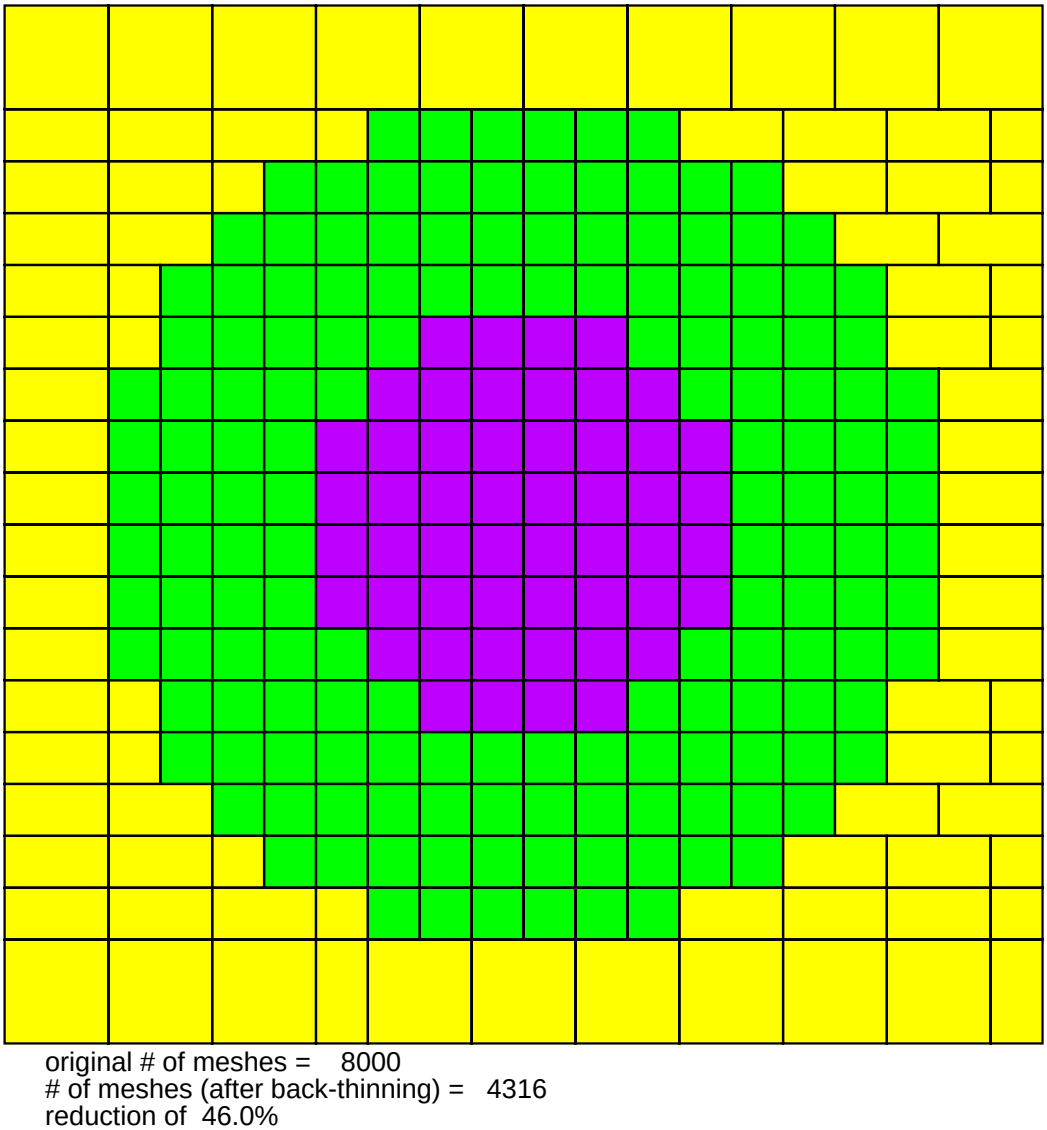


Figure 3.8: Alternative Back-Thinned Mesh for Example 2

### 3.2.3 Adjoint Source

The source in the adjoint problem is equivalent to the detector in the direct (forward) problem. The user must define a cell in the MCNP input file corresponding to the detector or region of interest. Then, by the same means by which materials are assigned to meshes, a uniform source is assigned over the spatial meshes that correspond to the user specified cell (region of interest). For numerical reasons, the user must define a reasonably fine mesh in the detector region. This can be accomplished by reducing the back-thinning parameter for the region/material of interest. The adjoint group boundaries and source spectrum or response function are taken directly from the MCNP input file, through the use of new input keywords. These values are entered in the normal MCNP convention (i.e., values are entered from low energies to high energies and energy boundaries are in units of MeV).

### 3.2.4 Other $S_N$ Input Parameters

In addition to the variables associated with the geometry and source, there are a few TORT parameters that must be specified. In the current version of A$^3$MCNP, these following parameters are set to specific values:

**ntscl**=2; scalar flux output is written to logical unit 2

**iadj**=1; adjoint solution

**mode**=1; $\theta$-weighted differencing scheme

**theta**=0.3; $\theta$ value for $\theta$-weighted differencing scheme[19, 20]

**locobj**=#; initial memory allocation is calculated based on the number of spatial meshes and energy groups

**ingeom**=0; x-y-z geometry indicator

**mm**=96; maximum number of directions ($S_8$)

Because many shielding problems are characterized by a small detector in a region containing low density material (e.g., air), quadrature order can be important in the adjoint problem, in which the detector is replaced by the adjoint source. $S_8$ is adequate for the majority of shielding type adjoint problems.

Notable values for which the default values are changed include:

**nfxmx**=40; maximum number of flux iterations (default=20)

**epp**=0.005; pointwise flux convergence criteria (default=0.001).

The maximum number of flux iterations is increased with respect to the default in an attempt to converge the first few groups (low energy groups) which can often be difficult or slowly converging. The pointwise flux convergence is relaxed slightly with respect to the default because we are not interested in high accuracy.

If any of the above values are not suitable for a given problem, the user is free to change them. Users are referred to the TORT manual[6] for detailed information on input parameters.

### 3.2.5 Generation of Input Files for Cross-Section Mixing

One of the distinctions between MC and deterministic methods, is that MC methods are capable of performing simulations with continuous energy or point-wise cross-section data. Thus when translating a MC input file into a corresponding deterministic input file, the energy dependence of the material cross sections must be discretized into energy groups. The selection of these groups must be based on the material cross sections. As with the spatial discretization, a compromise between accuracy and efficiency must be made.

Another distinction between the two methods is the specification of materials. In MC codes such as MCNP, a material is defined by isotope identifiers, referred to as ZAIDs, their corresponding weight or atomic fraction, and a total atom or mass density. For $S_N$ transport codes, on the other hand, it is necessary to mix/process the multigroup cross-section data into macroscopic cross-section mixtures prior to performing the transport calculation.

To automate the TORT calculation, it is necessary to automate the cross-section mixing/processing, which requires the generation of an input file for the GIP code. The typical GIP input file consists of five sections containing the following information: (1) basic parameters describing the multigroup library (e.g., number of groups, position of total cross section, etc.), (2) material (mixture) numbers, (3) component (isotope) numbers, (4) nuclide identifiers, and (5) atom densities. The information required in the first four of these sections is specific to the multigroup library, thus requiring additional user input.

The basic parameters describing the multigroup library are entered via a new MCNP input card. The atom densities and material mixture specifications, in terms of ZAIDs (isotope identifiers), are taken directly from the MCNP input. For the specification of material mixtures in GIP, the MCNP ZAIDs must be associated with ("translated into") the multigroup library specific component numbers (isotope identifiers). The major difficulty with automating the generation of GIP input file lies in this association. Because there is no consistency in component numbers between multigroup libraries, this association is different for each multigroup library, and thus, cannot be hard-coded. To solve this problem, the user is required to generate an additional input file (named *zaid.in*). The first section of this file contains two columns of numbers; the first column lists the MCNP isotope ZAIDs and the second column lists the associated library specific component numbers.

Thus, with the use of this information the multigroup component numbers are matched with the appropriate ZAIDs in the material description to produce material definitions in terms of the multigroup component numbers. The second (last) section of this file contains the multigroup library specific nuclide identifiers required by GIP. Once the *zaid.in* file is generated for a particular multigroup library, the generation of GIP input files from MCNP material descriptions is completely automated. Further, the generation of the *zaid.in* file is completely straightforward. An example of the *zaid.in* file for the CASK[21] multigroup cross-section library is provided in Fig. A.2 of Appendix A. *zaid.in* files are available for a number of widley-used multigroup cross-section libraries.

## Chapter 4

# USAGE - INPUT CARDS & EXECUTION

The methodologies described in the previous two chapters have been implemented into the standard MCNP code. The modified version of MCNP, designated A$^3$MCNP - Automatic Adjoint Accelerated MCNP, can automatically: (1) generate input files for $S_N$ adjoint calculations and (2) calculate and utilize variance reduction parameters from $S_N$ adjoint functions.

Figure 4.1 shows the automated process for variance reduction of MC calculations with A$^3$MCNP. The A$^3$MCNP input file consists of a standard MCNP input file with the following additional information:

- boundary conditions for the TORT calculation (6 values for 3-D)
- definition of initial fine mesh (9 values)
- coarse meshes for PCR in TORT
- a control parameter for activating the automatic mesh generation for uniform and/or back-thinned mesh
- MCNP cell number corresponding to the region of interest
- multigroup library parameters
- multigroup energy group boundaries
- response function.

With this input file, and the material processing input file (*zaid.in*), A$^3$MCNP generates input files for TORT and GIP. After the execution of these two codes, A$^3$MCNP reads the 3-D scalar adjoint function from the standard TORT binary VARSCL (VARiable SCaLar) output file and couples the original source distributions with the adjoint function to generate the source biasing and weight window parameters. These parameters are then used by A$^3$MCNP to perform the transport calculation. With the use of script files, this process is automated.

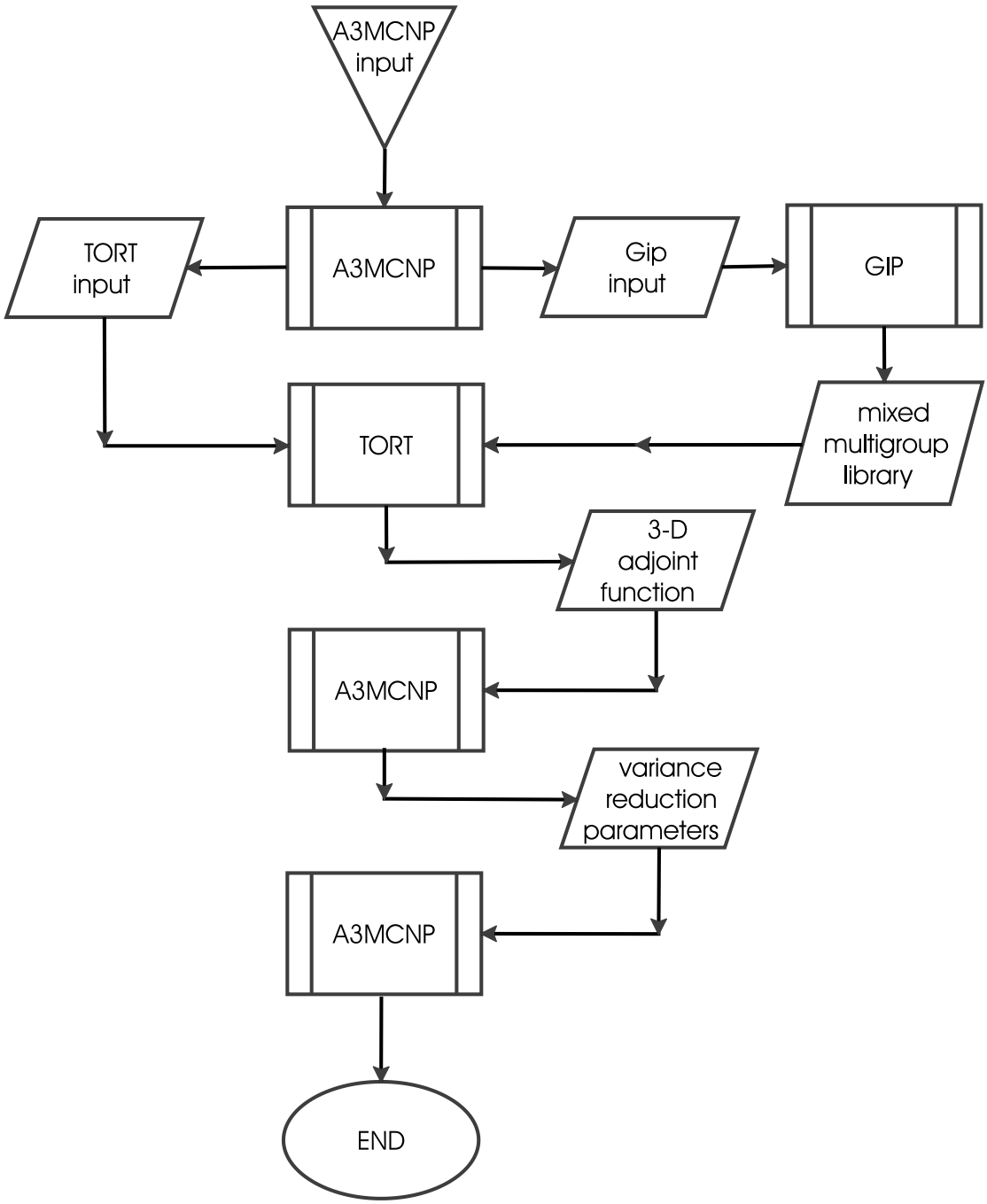


Figure 4.1: Automated Process for Variance Reduction with A$^3$MCNP

## 4.1 A$^3$MCNP Input Cards

As stated, the process has three distinct steps, which include: (1) generation of input and execution of the $S_N$ adjoint calculation, (2) processing of the adjoint function into VR parameters, and (3) performing the actual transport calculation. The input cards are described below according to their associated step in the overall process.

### 4.1.1 STEP 1: Sn Input Preparation Cards

The following cards have to do with the first step in the process (i.e., automatic generation of TORT and GIP input files from the A$^3$MCNP input file).

**SNGP - Sn General Input Parameters [REQUIRED INPUT CARD]**

*Form: SNGP ISN ISRC IGM NSCTM IHT IHM IUPS NEUT*

| | |
|---|---|
| ISN | = TORT input generation |
| | ISN = 0/1/2 = no mesh generation/discontinuous mesh/uniform mesh |
| ISRC | = MCNP cell for the adjoint source |
| IGM | = total number of energy groups for TORT calculation |
| NSCTM | = maxium order of scattering expansion for TORT calculation |
| IHT | = position of total cross section in cross section table |
| IHM | = length of cross section table for each group |
| IUPS | = number of upscatter cross sections per group |
| NEUT | = last neutron group |

Default: ISN=0, TORT input is not generated; ISRC=0, no default value; IGM=40; NSCTM=3; IHT=3; IHM=43; IUPS=0, no upscattering; NEUT=22. Default values for IGM, NSCTM, IHT, IHM, IUPS, and NEUT correspond to the CASK library, which is currently the default multigroup library.

**SNMSH - Sn Spatial Mesh Input Preparation [REQUIRED INPUT CARD]**

*Form: SNMSH XL XU YL YU ZL ZU DX DY DZ KPRN*

| | |
|---|---|
| XL | = lower x boundary |
| XU | = upper x boundary |
| YL | = lower y boundary |

YU = upper y boundary
ZL = lower z boundary
ZU = upper z boundary
DX = initial thickness of x mesh
DY = initial thickness of y mesh
DZ = initial thickness of z mesh
KPRN = z-plane of TORT mesh for plot of mesh distribution; negative entry results in generation of a plot for each z-plane (plot(s) are written to mesh.ps file)

Usage: The x,y,z boundaries are used to define the boundaries of the TORT problem, while the DX, DY, and DZ entries are used to define a uniform mesh throughout the problem. The x,y,z boundaries must correspond to the MCNP problem boundaries.
Default: none

**SNBC - Sn Boundary Conditions**

*Form: SNBC IBL IBR IBI IBO IBB IBT*

IBL = left boundary condition (lower x)
IBR = right boundary condition (upper x)
IBI = inside boundary condition (lower y)
IBO = outside boundary condition (upper y)
IBB = bottom boundary condition (lower z)
IBT = top boundary condition (upper z)

Usage: 0/1/2 = vacuum/reflective/periodic
Default: vacuum boundary conditions for all sides

**SNSI - Sn Energy Group Boundaries [REQUIRED INPUT CARD]**

*Form: SNSI* $E_{g1}$, $E_{g2}$, ... $E_{IGM-1}$, $E_{IGM}$

E = upper energy group boundaries for Sn adjoint calculation

Usage: The upper energy group boundaries are entered from lowest energy group to highest energy group in units of MeV. A total of IGM values are expected. In the case of a coupled neutron-gamma calculation, the energy group boundaries for the gamma groups should be entered first, followed by the group boundaries for the neutrons.
Default: none

**SNSP - Sn Source Energy Spectrum [REQUIRED INPUT CARD]**

*Form: SNSP* $SP_{g1}$, $SP_{g2}$, ... $SP_{IGM-1}$, $SP_{IGM}$

SP = group source for Sn adjoint calculation, which should correspond to the response function for the forward MCNP calculation

Usage: The energy spectrum should be entered from lowest energy group to highest energy group. A total of IGM values are expected. In the case of a coupled neutron-gamma calculation, the gamma spectrum should be entered first, followed by the neutron spectrum.
Default: none

**SNTHN - Sn Mesh Back-Thinning Parameters [OPTIONAL INPUT CARD]**

*Form: SNTHN* $BTP_1$, $BTP_2$, ... $BTP_{NM-1}$, $BTP_{NM}$

BTP = back-thinning parameter for each unique material. The value is used to establish the maximum mesh thickness (cm) for each material.

Default: zero - back-thinning is not performed

**SNACC - Course-Mesh Values for Sn Acceleration [OPTIONAL INPUT CARD]**

*Form: SNACC NXCR, NYCR,* $XCR_1$, $XCR_2$, ... $XCR_{NXCR-1}$, $XCR_{NXCR}$, $YCR_1$, $YCR_2$, ... $YCR_{NYCR-1}$, $YCR_{NYCR}$

NXCR = number of course meshes in x-direction
NYCR = number of course meshes in y-direction
XCR = actual x course meshes
YCR = actual y course meshes

### 4.1.2 STEPS 2 & 3: VR Parameter Calculation and Transport Card

The following card is used in the steps following the execution of GIP and TORT to tell A$^3$MCNP to either (1) read the 3-D adjoint function from the TORT binary VARSCL (VARiable SCaLar) output file and couple the original source distributions with the adjoint function to generate the source biasing and weight window parameters or (2) perform the transport calculation using the calculated VR parameters.

**WWA - Adjoint Weight Window Parameters [REQUIRED INPUT CARD]**

*Form: WWA:n WIGM AAAON IMFP*

| | |
|---|---|
| n | = N for neutrons, P for photons |
| WIGM | = total number of weight window energy intervals (energy groups) |
| AAAON | = 0 = calculate weight window and source biasing parameters |
| | = 1 = use calculated weight window and source biasing parameters |
| IMFP | = increment for mfp weight checkin, defaul=1 |

Usage: AAAON equals 0 for step 2 and 1 for step 3.
Default: none

## Chapter 5

# EXAMPLE APPLICATION OF A$^3$MCNP

A shipping cask problem is a multi-region shielding problem that features both rectangular and cylindrical geometries, and thus, is useful for demonstrating A$^3$MCNP, particularly the mesh generation capabilities. Therefore, in this section A$^3$MCNP is applied to the Four-Assembly PWR Depleted Uranium Shipping Cask problem that is clearly described in the CASK multigroup library documentation (Ref. [21]). The problem consists of four rectangular fuel assemblies centered within a large cylindrical cask composed of steel and depleted uranium The objective is to calculate dose external to the cask. Figure 5.1 shows a radial slice through one quarter of the problem as model in MCNP. The neutron source is uniformly distributed throughout the fuel assembly with a Cf-252 fission spectrum, as defined in [21]. The circular region in the upper right-hand corner of the figure represents the dose location. Figure 5.2 shows an axial slice through the problem at the azimuthal coordinate of 45°, as prepared by MCNP. The dose location is shown in the bottom right-hand corner of this figure.

The purpose of this problem is to demonstrate the automatic variance reduction capabilities of A$^3$MCNP in general, and to examine the mesh generation in particular. By adjusting the mesh generation parameters, different mesh descriptions for the $S_N$ adjoint calculation will be produced. Therefore, we examine the effect of the mesh on total computational time. The CASK 22-group $P_3$ neutron cross-section library[21] was employed for the $S_N$ adjoint calculations. It should be noted that this is not a deep-penetration problem, and thus, is not very difficult from a computational standpoint. As a result, this problem is not well suited for demonstrating the significant speedups[3, 4] that are possible with A$^3$MCNP. This problem was selected for demonstration due to its relative simplicity and general similarity to other more challenging problems (e.g., reactor pressure vessel, BWR core shroud, rail-type spent fuel shipping cask, concrete spent fuel storage cask, etc.)

For the calculation of dose at the detector position, the adequacy of the radial mesh description is clearly more important than that of the axial mesh description. As mentioned, the goal of the adjoint calculation is not to calculate the correct answer, but rather to calculate a function with approximately the correct shape. To accomplish this, it is necessary

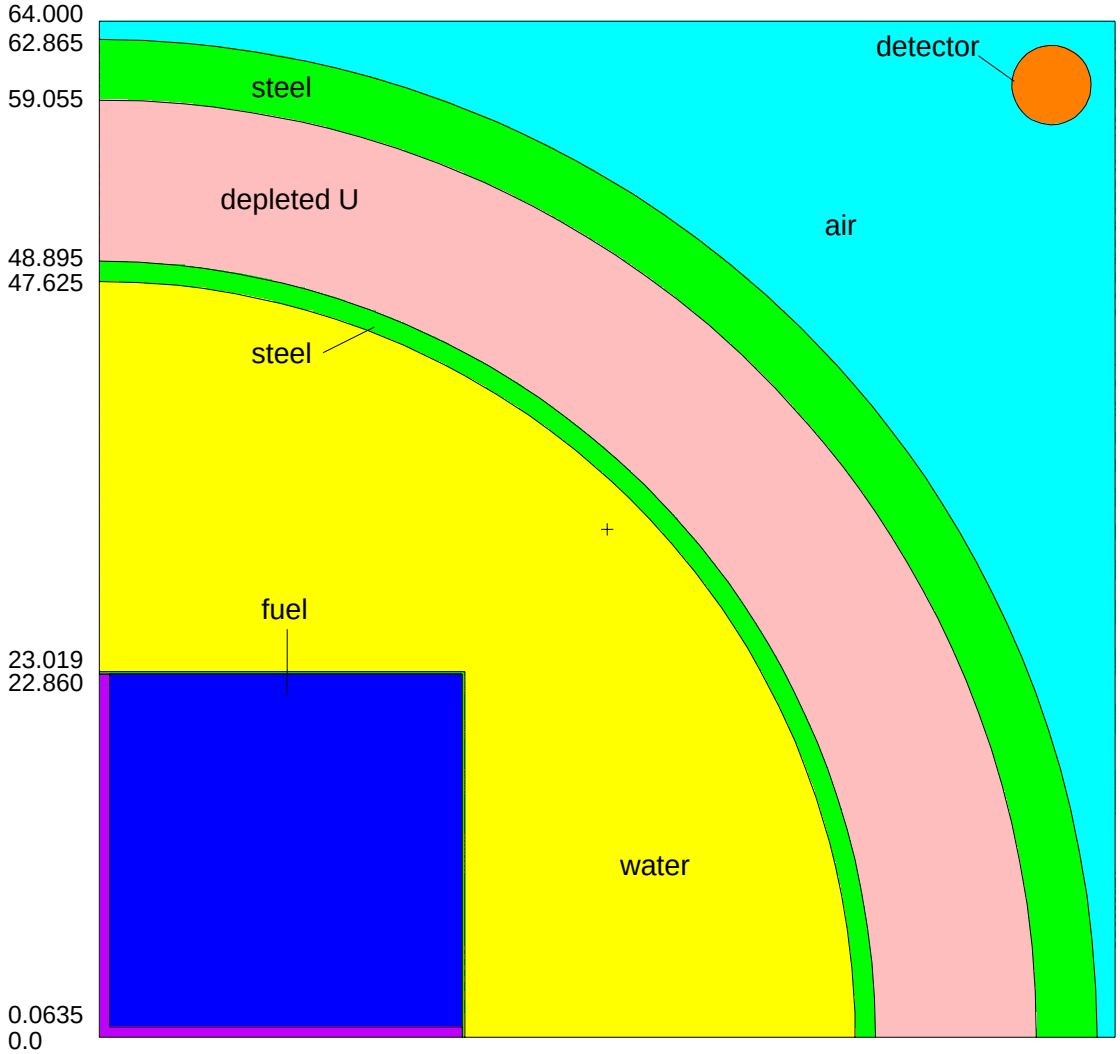


Figure 5.1: Radial Slice Through One Quarter of the Four-Assembly PWR Depleted Uranium Shipping Cask

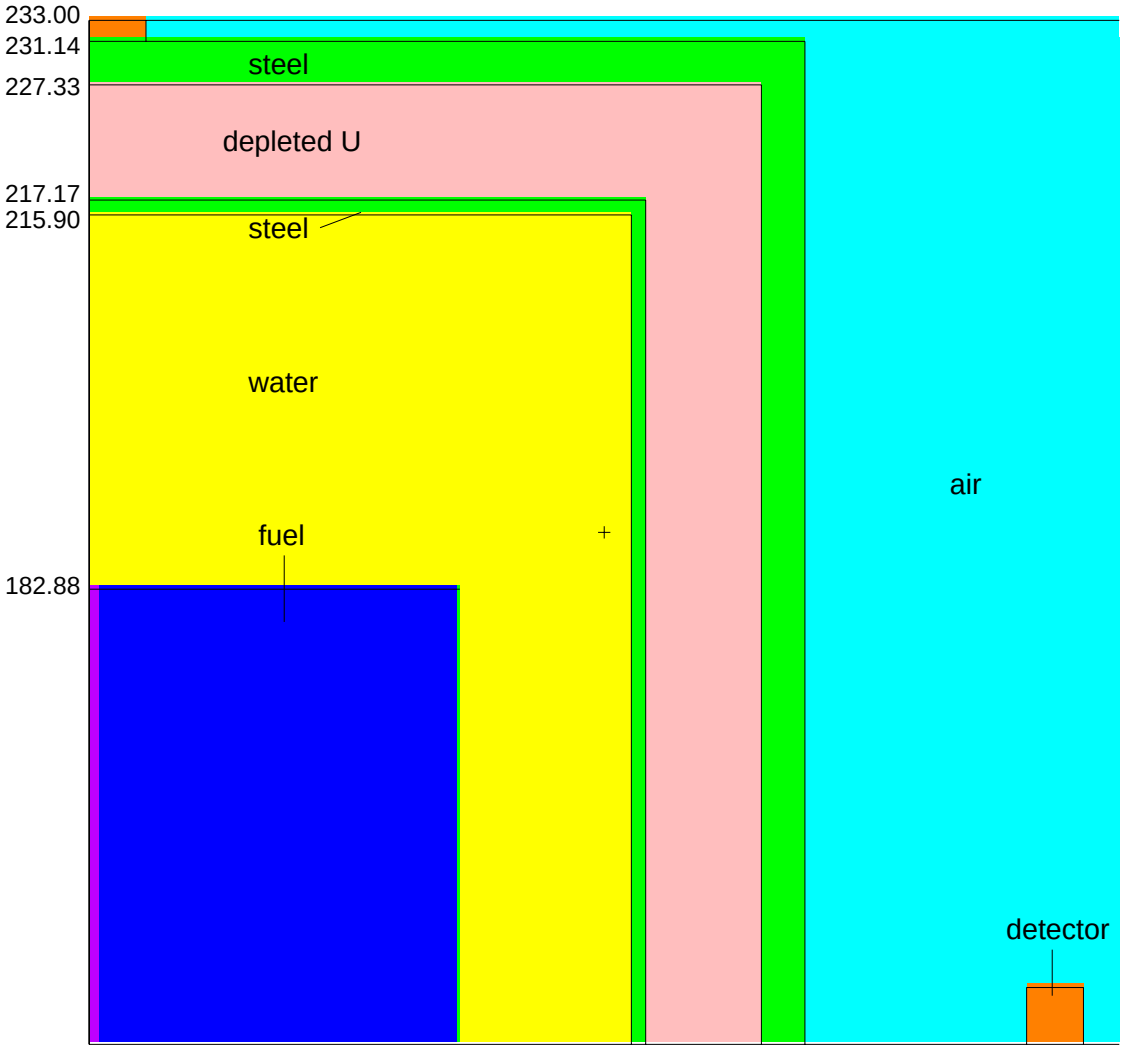


Figure 5.2: Axial Slice Through One Quarter of the Four-Assembly PWR Depleted Uranium Shipping Cask

Table 5.1: Average Material mfps and Back-Thinning Parameters for the Shipping Cask Problem

| material | mfp (cm) | back-thinning parameter (cm) |
|---|---|---|
| steel-boron | 3.1 | 3.1 |
| fuel | 2.0 | 2.0 |
| water | 1.0 | 1.0 |
| steel | 3.0 | 3.0 |
| depleted U | 2.3 | 2.3 |
| air | 3000 | 5.0 |

to have a mesh description that preserves the total thicknesses of the various shielding materials. However, accurate representation of material boundaries requires a large number of meshes, and subsequently, greater computer CPU time. This is particularly true when representing cylindrical boundaries with rectangular meshes as is the case for this problem.

In order to investigate the effect of the mesh on the adjoint accuracy, through its effectiveness for variance reduction of the MC calculation, the following cases are considered:

**Case 1:** a 1×1×2 cm $(x, y, z)$ fixed mesh over the entire problem (as shown in Fig. 5.3)

**Case 2:** a 2×2×2 cm $(x, y, z)$ fixed mesh over the entire problem (as shown in Fig. 5.4)

**Case 3:** a 2×2×3 cm $(x, y, z)$ fixed mesh over the entire problem (same as shown in Fig. 5.4)

**Case 4:** a discontinuous mesh derived from an initial fine mesh of 1×1×2 with back-thinning parameters based on mfps (as shown in Fig. 5.5)

**Case 5:** a discontinuous mesh derived from an initial fine mesh of 1.5×1.5×2 with back-thinning parameters based on mfps (as shown in Fig. 5.6).

As shown in Fig. 5.3, the Case 1 mesh description represents the material boundaries/thicknesses fairly well. However, the mesh sizes are smaller than the material mfps require. The Case 2 mesh description, as shown in Fig. 5.4, does not represent the material thicknesses nearly as well as the Case 1 mesh, but in general, the mesh sizes are acceptable when compared to the material mfps. Cases 4 and 5 are representative of a compromise between minimizing the number of meshes and preserving material boundaries. Table 5.1 lists the average material mfps as calculated by MCNP, and the back-thinning parameters used for creating the mesh descriptions for Cases 4 and 5.

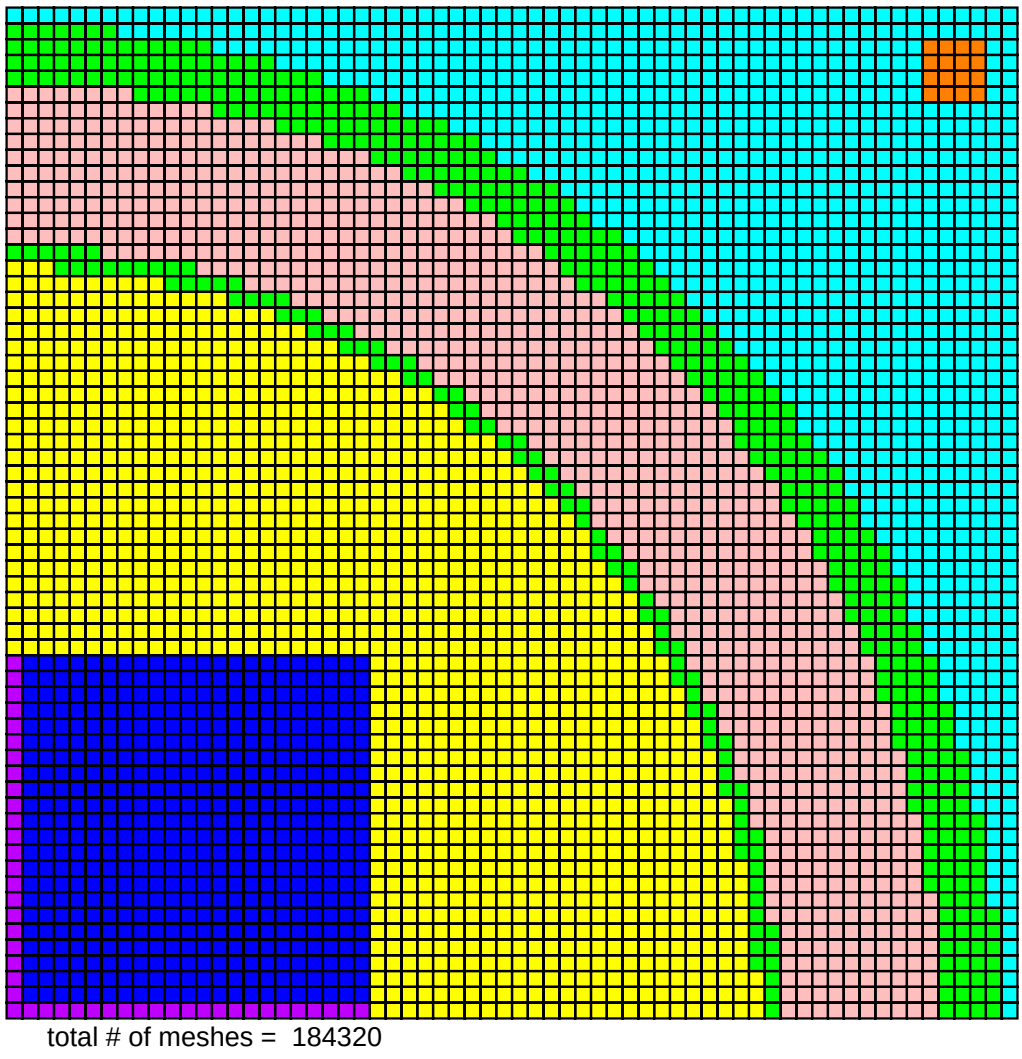


Figure 5.3: Case 1: 1×1×2 cm fixed mesh over the entire problem

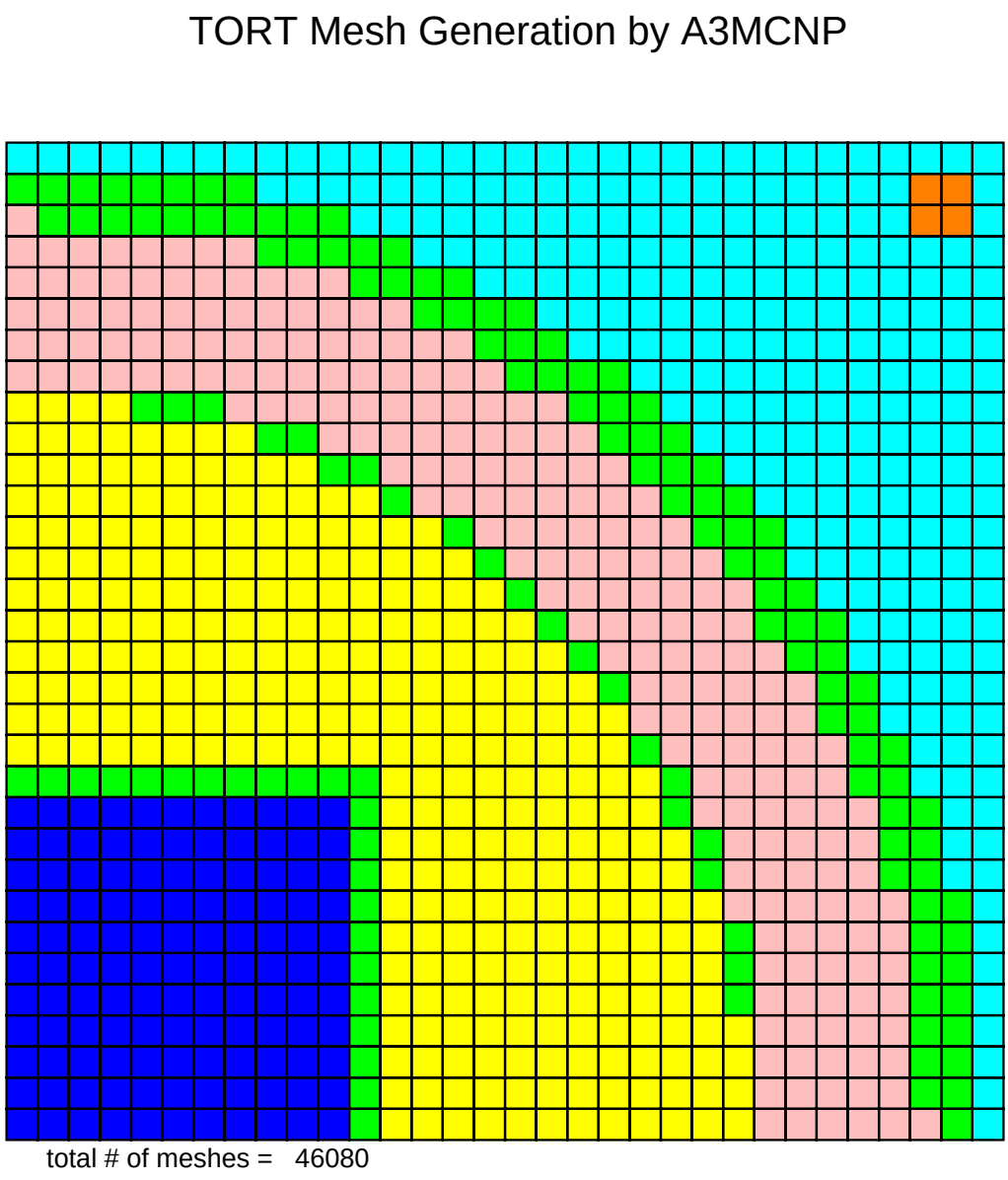


Figure 5.4: Case 2: 2×2×2 cm fixed mesh over the entire problem

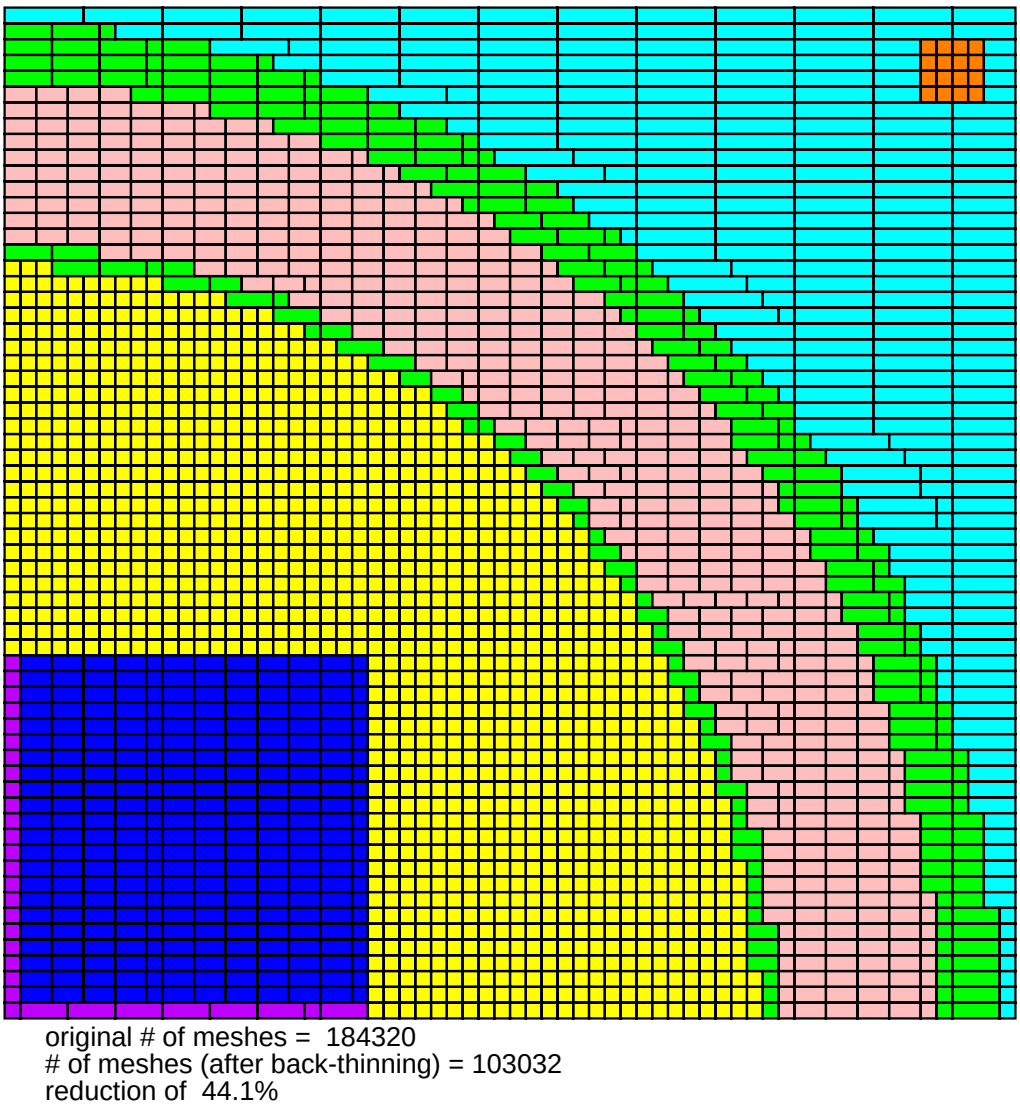


Figure 5.5: Case 4: discontinuous mesh derived from initial fine mesh of 1×1×2

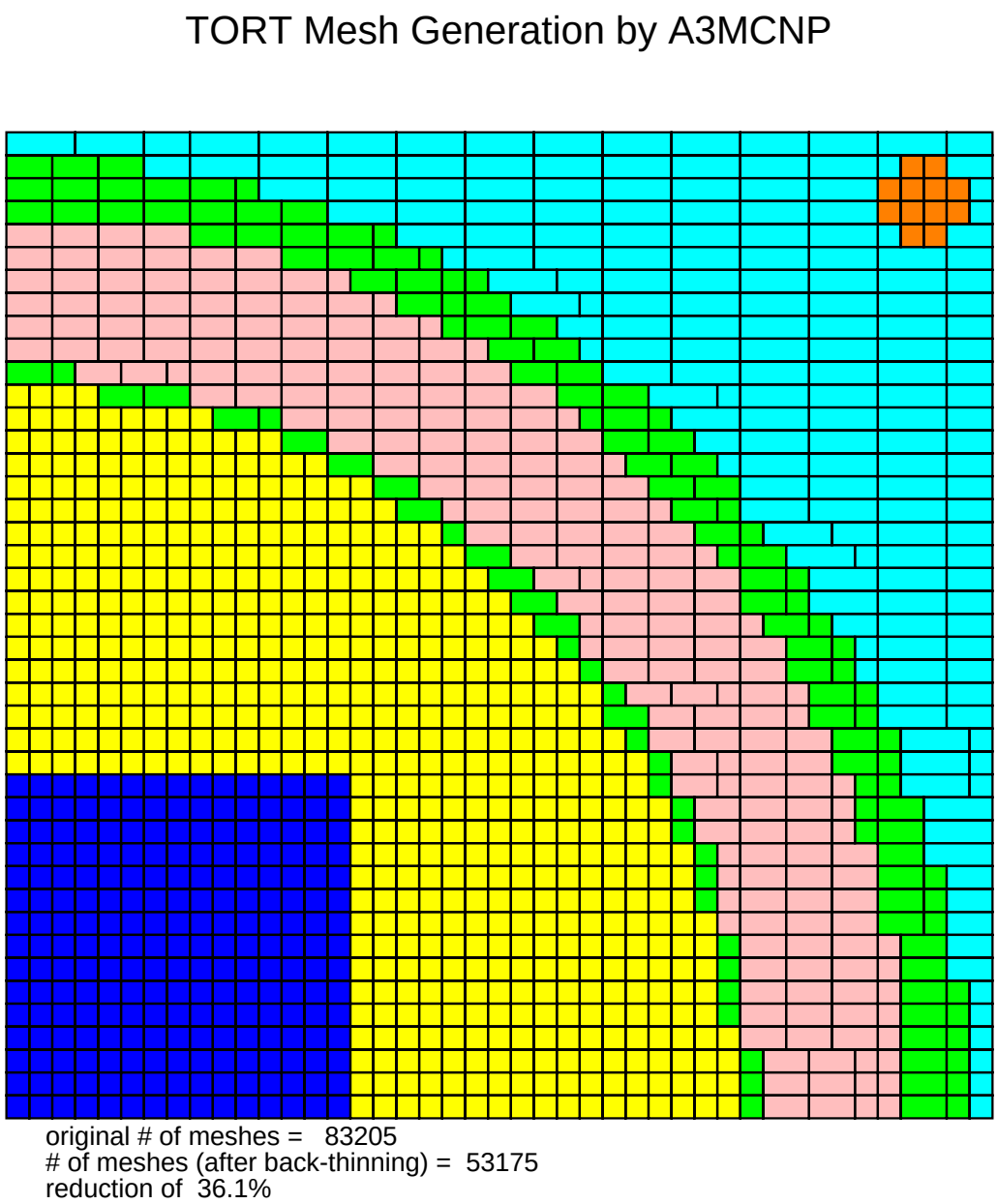


Figure 5.6: Case 5: discontinuous mesh derived from initial fine mesh of 1.5×1.5×2

Table 5.2: CASK Neutron Energy Group Boundaries

| Group | Upper Energy Group Boundaries (MeV) | Group | Upper Energy Group Boundaries (MeV) |
|---|---|---|---|
| 1 | 1.50E+01 | 12 | 1.11E+00 |
| 2 | 1.22E+01 | 13 | 5.50E-01 |
| 3 | 1.00E+01 | 14 | 1.11E-01 |
| 4 | 8.18E+00 | 15 | 3.35E-03 |
| 5 | 6.36E+00 | 16 | 5.83E-04 |
| 6 | 4.96E+00 | 17 | 1.01E-04 |
| 7 | 4.06E+00 | 18 | 2.90E-05 |
| 8 | 3.01E+00 | 19 | 1.07E-05 |
| 9 | 2.46E+00 | 20 | 3.06E-06 |
| 10 | 2.35E+00 | 21 | 1.12E-06 |
| 11 | 1.83E+00 | 22[a] | 4.14E-07 |

[a] Lower energy of group 22 is 1.00E-10 MeV

Figures 5.7 and 5.8 show the adjoint distributions for energy group 10 over an entire plane for Cases 1 and 2, respectively. These figures nicely demonstrate the behavior of the adjoint (importance) function throughout the problem for the two cases.

The variation of the adjoint function with energy is shown in Fig. 5.9, which plots the Case 1 radial distributions (through the azimuthal coordinate of 45°) for several energy groups. The energy groups to which the adjoint functions are referenced, correspond to the CASK library. The neutron energy group boundaries for the CASK library are provided in Table 5.2. To examine the effect of the mesh descriptions on accuracy, Figs. 5.10 through 5.12 compare the calculated adjoint functions from Cases 1 and 2 for energy groups 3, 10, and 20, respectively. The adjoint functions corresponding to Case 2 follow the adjoints from Case 1 near the detector (adjoint source), but begin to deviate in the cask. The differences between the two cases are larger for the lower energy groups, reaching nearly six orders of magnitude at the cask center for group 20. However, the general shapes remain similar

The effectiveness of the various cases for the variance reduction of MC calculations, is demonstrated in Table 5.3, which compares unbiased MCNP results to those calculated with $A^3$MCNP. While the $A^3$MCNP results were each generated with 15 minutes of CPU time, a non-zero estimate could not be obtained by the unbiased case in the same amount of CPU time. To obtain a non-zero estimate and a value for the FOM, the unbiased case was allowed to run for 120 minutes of CPU time. It is immediately clear from the FOM values, that $A^3$MCNP is capable of increasing the efficiency of this calculation by a factor of $\sim$400, regardless of the mesh description considered. In terms of the MC calculation alone, Case

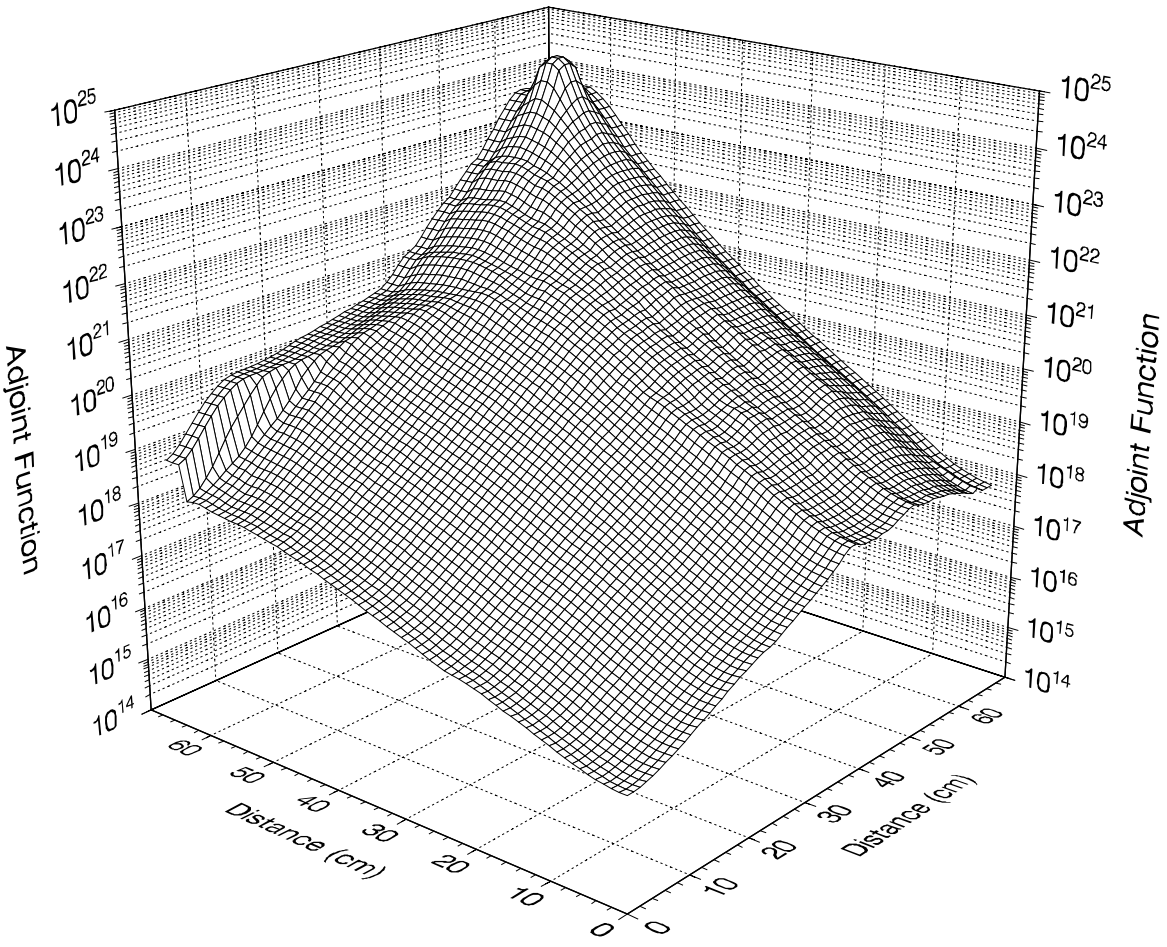


Figure 5.7: Case 1 Adjoint Function Distribution for Energy Group 10 (1.8-2.4 MeV)

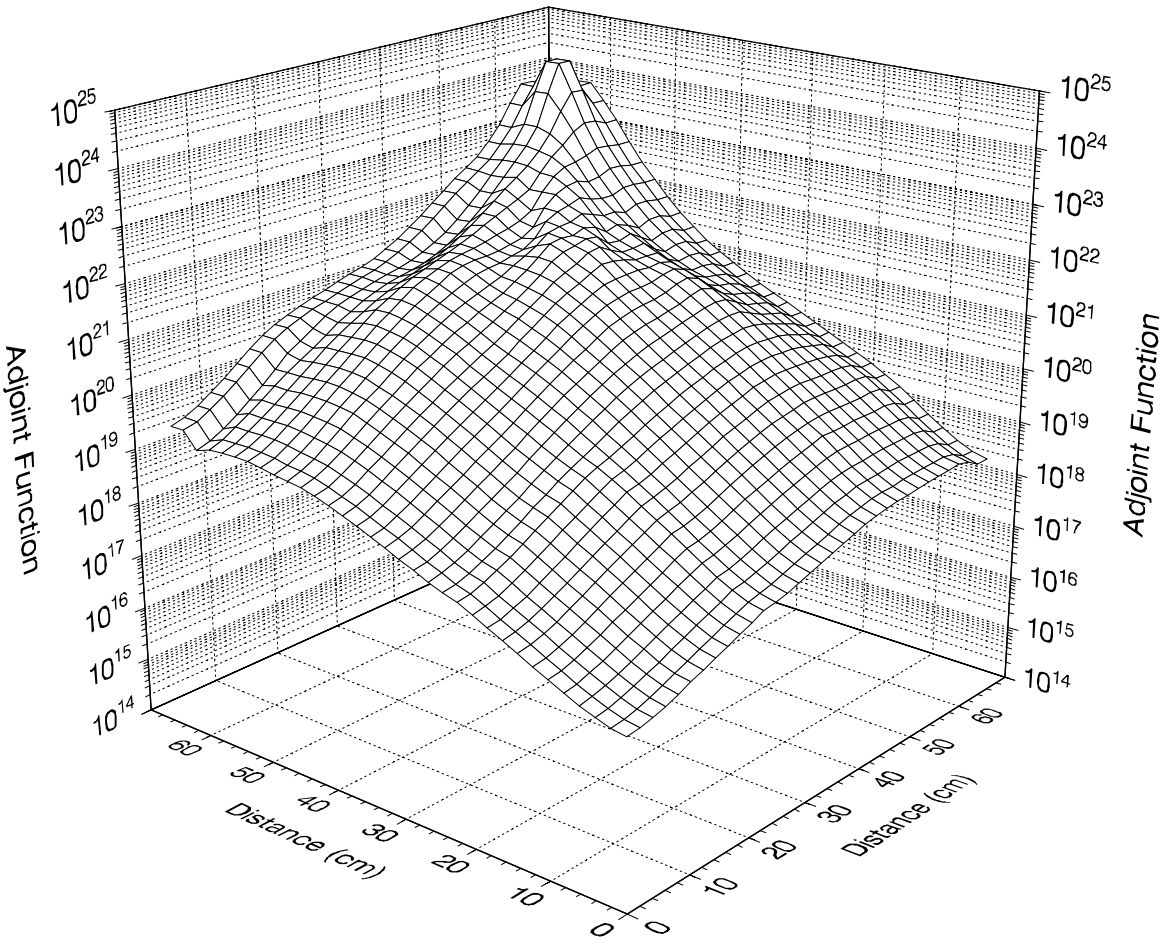


Figure 5.8: Case 2 Adjoint Function Distribution for Energy Group 10 (1.8-2.4 MeV)

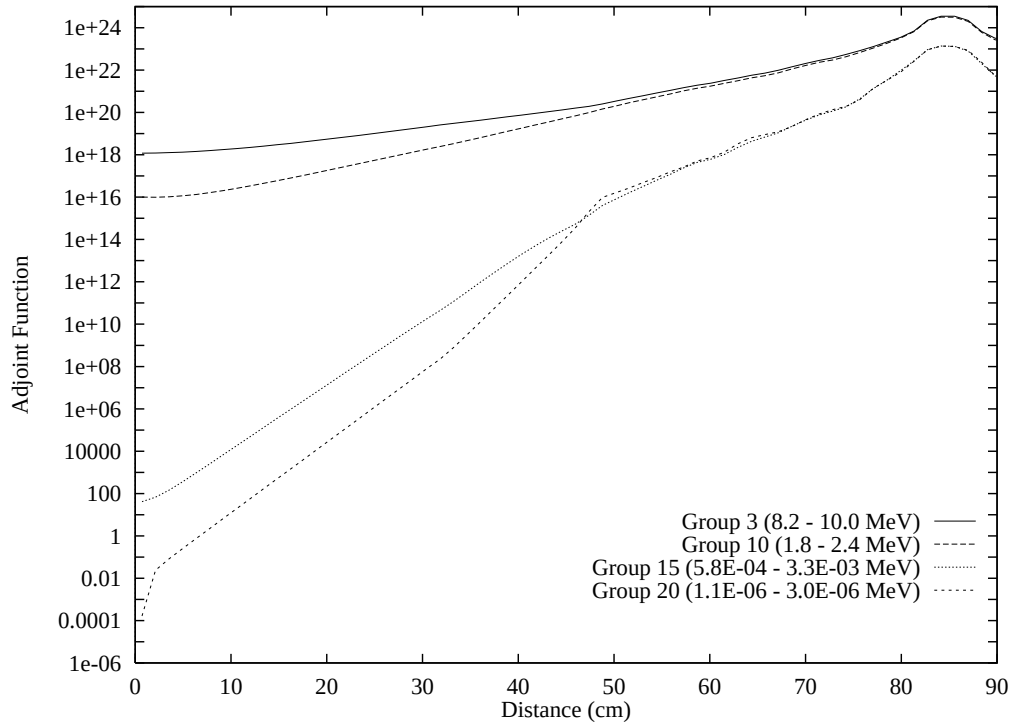


Figure 5.9: Case 1 Adjoint Distributions for Various Groups

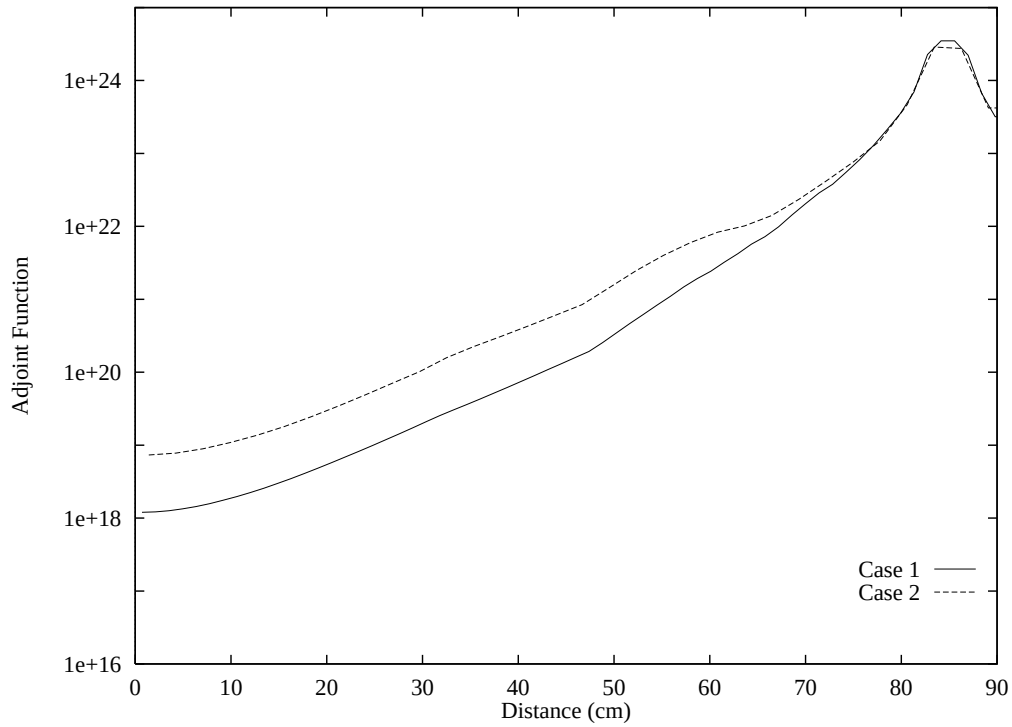


Figure 5.10: Comparison of Case 1 and 2 Adjoint Distributions for Group 3

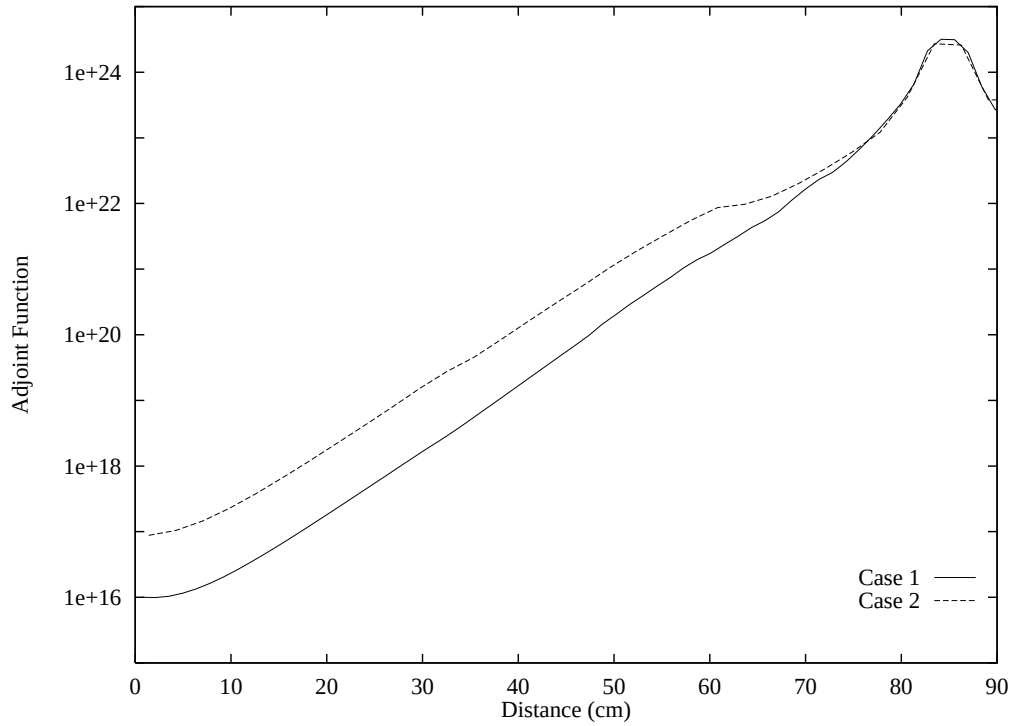


Figure 5.11: Comparison of Case 1 and 2 Adjoint Distributions for Group 10

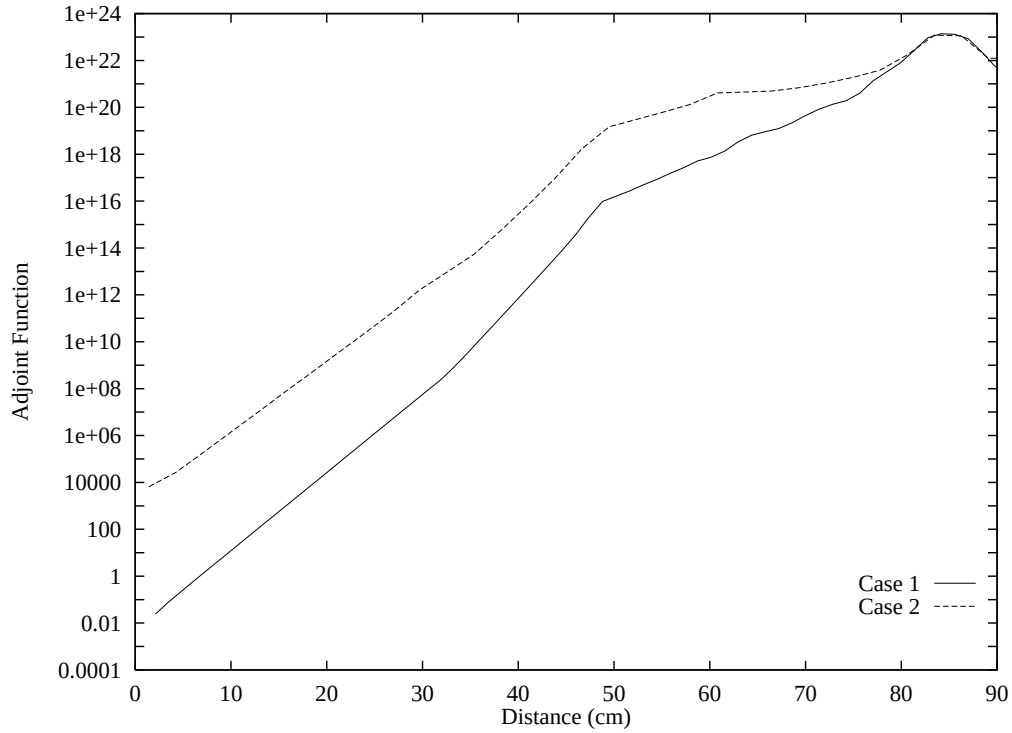


Figure 5.12: Comparison of Case 1 and 2 Adjoint Distributions for Group 20

Table 5.3: Effect of $S_N$ Adjoint on Dose Calculation at the Radial Detector Location

| Case | dose(mrem/hr) | R | VOV | FOM |
|---|---|---|---|---|
| unbiased | 52.93 | 0.40 | 0.201 | 0.052 |
| 1 | 34.02 | 0.06 | 0.063 | 20 |
| 2 | 32.80 | 0.05 | 0.039 | 25 |
| 3 | 33.61 | 0.06 | 0.175 | 18 |
| 4 | 31.37 | 0.05 | 0.020 | 29 |
| 5 | 33.82 | 0.05 | 0.034 | 30 |

5 is the most efficient, exhibiting a FOM nearly 600 times larger than that of the unbiased case. It is interesting to note that the Case 2 mesh description, which does not represent the material boundaries particularly well and was shown to generate adjoint functions that vary significantly from those calculated with the more accurate mesh description of Case 1, is quite effective for this problem. This demonstrates the effectiveness of an *approximate* adjoint function.

The 3-D $S_N$ calculation can require a significant amount of CPU time. In general, the greater the number of meshes, the greater the CPU time. Therefore, it is appropriate to compare the efficiency of the various cases in terms of total CPU time. In order to make such a comparison, it is necessary to qualify the MCNP CPU time to a particular precision. Table 5.4 lists the number of meshes and subsequent time required by the TORT calculations, the time required by the MCNP calculations to achieve a $1\sigma$ uncertainty of 2%, and the total CPU time for the various cases. While the time required by the TORT calculation can actually exceed that of the A$^3$MCNP calculation (e.g., for Cases 1, 4, and 5), the total time is still reduced by a factor of $\sim$100. The amount of time that TORT requires to calculate the adjoint is less than that required by an analyst to develop comparable variance reduction parameters. Further, computer time is unquestionably less expensive than an analyst's time. Comparing the speedups in the last column of Table 5.4, A$^3$MCNP is a factor of $\sim$200 more efficient than the unbiased case.

Table 5.4: Effect of $S_N$ Mesh on Total Computational Time for the Radial Detector Location

| Case | TORT | | MCNP | Total | Speedup |
|---|---|---|---|---|---|
| | # of meshes | CPU time (minutes) | CPU time (minutes) | CPU time (minutes) | (unbiased total CPU/ total CPU) |
| unbiased | —– | — | 48,077 | 48,077 | 1 |
| 1 | 184320 | 404 | 115 | 529 | 91 |
| 2 | 46080 | 87 | 100 | 187 | 257 |
| 3 | 30720 | 45 | 139 | 184 | 261 |
| 4 | 103032 | 364 | 86 | 450 | 107 |
| 5 | 53175 | 144 | 83 | 227 | 211 |

# Chapter 6

# FUTURE WORK

A$^3$MCNP continues to be upgraded and enhanced to enable greater user flexibility and to add new features. A brief list of the planned enhancements (in no particular order) is provided below.

## 6.1 Planned Enhancements for A$^3$MCNP

- Upgrade modifications to MCNP version 4C
- Add capability to couple to PENTRAN[22]
- Incorporate source biasing for surface and volume sources that are not specified with points
- Allow user to explicitly specify Z-mesh dimensions
- Add ability to back-thin in Z-dimension
- Enable automatic mesh generation with MCNP repeated structures
- Add capability to automatically project the source spectrum and response function onto the group structure of the multigroup library being used.
- Incorporate source biasing for use with the surface source restart file
- Add capability to generate multigroup cross sections generate multigroup cross sections

# Bibliography


[1] J.C. WAGNER, "Acceleration of Monte Carlo Shielding Calculations with an Automated Variance Reduction Technique and Parallel Processing," *Ph.D. Thesis*, The Pennsylvania State University, Nuclear Engineering Dept. (1997).

[2] J.C. WAGNER and A. HAGHIGHAT, "Automatic Variance Reduction for Monte Carlo Shielding Calculations with the Discrete Ordinates Adjoint Function," *Proc. Joint. Int. Conf. on Mathematical Methods and Supercomputing in Nuclear Applications*, Saratoga Springs, New York, October 6-10, 1997, Vol. 1, p. 671, American Nuclear Society.

[3] J.C. WAGNER and A. HAGHIGHAT, "Automated Variance Reduction of Monte Carlo Shielding Calculations Using the Discrete Ordinates Adjoint Function," *Nucl. Sci. Eng.*, **128**, 186 (1998).

[4] A. HAGHIGHAT, H. HIRUTA, B. PETROVIC, and J.C. WAGNER, "Performance of the Automated Adjoint Accelerated MCNP ($A^3$MCNP) for Simulation of a BWR Core Shroud Problem," *Proceedings of the International Conference on Mathematics and Computation, Reactor Physics, and Environmental Analysis in Nuclear Applications*, Madrid, Spain, September 27-30, 1999.

[5] J.F. BRIESMEISTER, Editor, "MCNP – A General Monte Carlo N-Particle Transport Code, Version 4A," LA-12625, Los Alamos National Laboratory (1993).

[6] W.A. RHOADES and R.L. CHILDS, "TORT - Two- and Three-Dimensional Discrete Ordinates Transport, Version 1.515" CCC-543, ORNL-RSICC, Oak Ridge, TN (1992).

[7] J.C. WAGNER and A. HAGHIGHAT, "Application of the Discrete Ordinates Adjoint Function to Accelerating Monte Carlo Reactor Cavity Dosimetry Calculations," *1996 Radiation Protection & Shielding Topical Meeting*, No. Falmouth, MA **1**, 345 (April 21-25, 1996).

[8] J.C. WAGNER and A. HAGHIGHAT, "Acceleration of Monte Carlo Reactor Cavity Dosimetry Calculations with the Discrete Ordinates Adjoint Function," *Ninth Int. Symposium on Reactor Dosimetry*, Prague, Czech Republic, September 2-6, pp. 754-761, World Scientific, 1998.

[9] A. HAGHIGHAT, J.C. WAGNER, E.L. REDMOND II, and S. ANTONE, " Performance of $A^3$MCNP for Simulation of a Storage Cask," Submitted for presentation at the Radiation and Protection Conference, 2000, Spokane, WA.

[10] I. LUX and L. KOBLINGER, Monte Carlo Particle Transport Methods: Neutron and Photon Calculations, CRC Press, Boca Raton, FL (1990).

[11] E. E. LEWIS and W. F. MILLER, Computational Methods of Neutron Transport, John Wiley & Sons, New York (1984).

[12] M.H. KALOS, "Importance Sampling in Monte Carlo Shielding Calculations," *Nucl. Sci. Eng.*, **16**, 227 (1963).

[13] M.H. KALOS and P.A. WHITLOCK, Monte Carlo Methods - Volume 1: Basics, John Wiley & Sons, New York, (1986).

[14] R.R. COVEYOU, V.R. CAIN, and K.J. YOST, "Adjoint and Importance in Monte Carlo Application," *Nucl. Sci. Eng.*, **27**, 219 (1967).

[15] J.S. TANG and T.J. HOFFMAN, "Monte Carlo Shielding Analyses Using an Automated Biasing Procedure," *Nucl. Sci. Eng.*, **99**, 329-342 (1988).

[16] K. NOACK, "Efficiency and Reliability in Deep-Penetration Monte Carlo Calculations," *Ann. Nucl. Energy*, **18**, 309-316 (1991).

[17] D.E. KNUTH, "The Art of Computer Programming, 2nd Edition," Addison-Wesley Pub. Co., Reading, Mass. (1973).

[18] W.A. RHOADES and M.B. EMMETT, "DOS: The Discrete Ordinates Code System," ORNL/TM-8362, Oak Ridge National Laboratory (1982).

[19] B.G. PETROVIC and A. HAGHIGHAT, "Effects of $S_N$ Method Numerics on Pressure Vessel Neutron Fluence Calculations," *Nucl. Sci. Eng.*, **122**, 167-193 (1996).

[20] B.G. PETROVIC and A. HAGHIGHAT, "Impact of Numerical Options in the DORT Code on PWR Pressure Vessel Fluence Calculations," *Proc. Int. Conf. Reactor Physics*, Knoxville, TN, 219 (April 11-15, 1994).

[21] "CASK-81: 22 Neutron, 18 Gamma-Ray Group, $P_3$ Cross Sections for Shipping Cask Analysis," DLC-23, ORNL-RSICC, Oak Ridge, TN (1987).

[22] G.E. SJODEN and A. HAGHIGHAT, "PENTRAN: A Code Package for Parallel Environment Neutral Particle Transport in 3-D Cartesian Geometry, Users Guide to version 4.36$\beta$," Nuclear Engineering Dept., Penn State University, Internal Report (1996).

# Appendix A

# Sample A$^3$MCNP Files for Geometry Example 1

## A.1 A$^3$MCNP Input File

```
Mesh generation example 1: box-in-a-box
c
c    CELL CARDS
1    1 -1.0   11 -12 21 -22 31 -32          imp:n 1  $ inner box
2    2 -2.0   10 -13 20 -23 30 -33  #1      imp:n 1  $ outer box
3    0      -100 (-10:13:-20:23:-30:33)     imp:n 0
4    0       100                            imp:n 0

c
c    SURFACE CARDS
10   px     0.0
11   px     4.0
12   px     8.0
13   px    12.0
20   py     0.0
21   py     4.0
22   py     8.0
23   py    12.0
30   pz     0.0
31   pz     4.0
32   pz     8.0
33   pz    12.0
100  so   200.0

c
c    DATA CARDS
mode n
c
c    source cards
sdef erg=d2 pos=d1
si1 l   .1 .1 .1
sp1 d    1
c    CASK 22-group neutron energy structure
si2 h 0.0         4.14000E-07 1.12000E-06 3.06000E-06 1.07000E-05
      2.90000E-05 1.01000E-04 5.83000E-04 3.35000E-03 1.11000E-01
      5.50000E-01 1.11000E+00 1.83000E+00 2.35000E+00 2.46000E+00
      3.01000E+00 4.06000E+00 4.96000E+00 6.36000E+00 8.18000E+00
      1.00000E+01 1.22000E+01 1.50000E+01
sp2 d 0 1.0 21r
```

```
c
c    material cards
m1    13027.60c  1.000
m2     8016.60c  0.333   1001.60c  0.667
c
c
c
c   ---------------------------------------------------------------------
c   end of standard MCNP input / beginning of A3MCNP input
c   ---------------------------------------------------------------------
c
c      sn general parameters
c      isn isrc  igm nsctm iht ihm iups neut
sngp     1   1    22   3    3   43   0    22
c
c      definition of initial sn fine mesh
c      xl  xu    yl  yu    zl  zu     dx  dy  dz  kprn
snmsh   0. 12.    0. 12.    0. 12.    1. 1. 1.     6
c
c    sn boundary conditions - snbc()=(ibl,ibr,ibi,ibo,ibb,ibt)
c      ibl ibr   ibi ibo   ibb ibt
snbc    0   0     0   0     0   0
c
c    sn energy group structure
c    CASK 22-group neutron energy structure
snsi 4.14000E-07 1.12000E-06 3.06000E-06 1.07000E-05 2.90000E-05
     1.01000E-04 5.83000E-04 3.35000E-03 1.11000E-01 5.50000E-01
     1.11000E+00 1.83000E+00 2.35000E+00 2.46000E+00 3.01000E+00
     4.06000E+00 4.96000E+00 6.36000E+00 8.18000E+00 1.00000E+01
     1.22000E+01 1.50000E+01
c
c    sn adjoint energy spectrum (response function)
c    CASK 22-group neutron flux-to-dose conversion factors
snsp 3.780E-03 3.960E-03 4.140E-03 4.320E-03 4.500E-03 4.680E-03
     4.680E-03 4.320E-03 6.480E-03 5.400E-02 1.188E-01 1.332E-01
     1.296E-01 1.260E-01 1.260E-01 1.296E-01 1.332E-01 1.404E-01
     1.476E-01 1.476E-01 1.656E-01 2.088E-01
c
c       btp1   btp2
snthn   1.0    2.0
c
c      nxcr  nycr  xcr1  xcr2  ycr1  ycr2
snacc   2     2    4.0   8.02  4.0   8.0
c
end-of-file
```

## A.2 Additional Input File for A$^3$MCNP (*zaid.in*)

```
  116        = mtp
  1001    1   This file corresponds to the CASK library DLC-23
  2004    5
  4009    9
  5010   13
  6000   17
  6012   17
  7014   21
  8016   25
 11023   29
 12000   33
 13027   37
 14000   41
 19000   45
 20000   49
 22000   53
 24000   57
 25055   61
 26000   65
 28000   69
 29000   73
 40000   77
 42000   81
 50000   85
 73181   89
 74000   93
 82000   97
 92235  101
 92238  105
 94239  109
 94240  113
t                          terminator
 13
  1000 1001 1002 1003 2000 2001 2002 2003 4000 4001 4002 4003
  5000 5001 5002 5003 6000 6001 6002 6003 7000 7001 7002 7003
  8000 8001 8002 8003 11000 11001 11002 11003 12000 12001 12002
  12003 13000 13001 13002 13003 14000 14001 14002 14003
  19000 19001 19002 19003 20000 20001 20002 20003
  22000 22001 22002 22003 24000 24001 24002 24003
  25000 25001 25002 25003 26000 26001 26002 26003
  28000 28001 28002 28003 29000 29001 29002 29003
  40000 40001 40002 40003 42000 42001 42002 42003
  50000 50001 50002 50003 73000 73001 73002 73003
  74000 74001 74002 74003 82000 82001 82002 82003
  92350 92351 92352 92353 92380 92381 92382 92383
  94390 94391 94392 94393 94400 94401 94402 94403
```