

ANALYSIS OF TITAN HYBRID ALGORITHMS FOR SPECT SIMULATION

By

KATHERINE ELIZABETH KELLER ROYSTON

A THESIS PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

UNIVERSITY OF FLORIDA

2010

© 2010 Katherine Elizabeth Keller Royston

To my parents for all their encouragement and my husband for his support

ACKNOWLEDGMENTS

I would like to express my gratitude to Dr. Alireza Haghghat, my advisor, for his patience and guidance in this work. His advice enabled me to successfully overcome the many obstacles I encountered. I thank committee member Dr. Ce Yi for taking the time to work with me and provide vital assistance in this research. I also thank Dr. David Gilland for his time as a committee member and his suggestions for the completion of this thesis. I would like to thank my fellow graduate students, Mike Wenner and Will Walters, for providing discussion of problems as they arose. Mike especially provided support on many occasions for which I am very grateful.

My family deserves acknowledgement for their endless support and encouragement of my academic studies. My parents have taught me to always do my best and provided me with the skills to succeed. I would like to thank my husband Sean for always having in faith in me.

TABLE OF CONTENTS

	<u>page</u>
ACKNOWLEDGMENTS.....	4
LIST OF TABLES.....	7
LIST OF FIGURES.....	8
ABSTRACT	10
CHAPTER	
1 INTRODUCTION	12
Motivation	12
SPECT Background.....	12
NCAT Code	13
2 THEORY.....	17
TITAN	17
Deterministic Transport Code.....	17
Differencing Schemes	19
Adjoint Transport Equation.....	20
Monte Carlo	21
Multigroup Approximation	23
3 METHODOLOGY	25
General Simulation Methods.....	25
TITAN Methods.....	26
SIMIND Methods	27
MCNP5 Methods	28
Adjoint Methods.....	30
4 RESULTS	37
TITAN Analysis	38
SIMIND Analysis and Comparison.....	39
Investigation into SIMIND and TITAN Differences	42
Adjoint Analysis and MCNP5 Comparison.....	43
5 CONCLUSIONS AND FUTURE WORK	60

APPENDIX

A	INPUT FILES.....	62
B	DATA PROCESSING CODES.....	77
	LIST OF REFERENCES	93
	BIOGRAPHICAL SKETCH.....	95

LIST OF TABLES

<u>Table</u>		<u>page</u>
4-1	Comparison between myocardium pixels in group 1 anterior projection images with increasing order of quadrature for different number of voxels (meshes)	48
4-2	Maximum absolute relative difference between TITAN flux distributions in the myocardium of the 64 x 64 x 64 model relative to the 128 x 128 x 128 model ...	49
4-3	Infinity-norm and 2-norm for TITAN S ₆ -DDZ and S ₆ -DTW compared to SIMIND for 131 million particles per projection.....	50
4-4	Three group computation time comparison for four projection angles	50
4-5	Comparison of relative differences of TITAN S ₆ -DDZ and S ₆ -DTW compared to SIMIND for 131 million particles per projection including the infinity-norm, the next largest difference after the infinity-norm, and the percentage of pixels with absolute relative differences greater than 10%	51
4-6	Single-hole collimator responses for TITAN compared with MCNP5.....	56
4-7	Single-hole collimator responses for TITAN with P _N -T _N ordinate splitting compared with MCNP5.....	56
4-8	Four-hole collimator responses for TITAN compared with MCNP5	57
4-9	Nine-hole collimator responses for TITAN compared with MCNP5	57
4-10	Ideal and fitted direction weights for each collimator case and the resulting recommended order of quadrature	59
4-11	Calculation times for 16 processors in MCNP5 and TITAN for each collimator..	59

LIST OF FIGURES

<u>Figure</u>	<u>page</u>
1-1 Geometry of a SPECT study	15
1-2 Cut-away view of NCAT material distribution.....	15
1-3 Cut-away view of NCAT source distribution.....	16
3-1 Example of pixel numbering in projection images.....	33
3-2 Schematic of a fictitious quadrature set with circular ordinate splitting	33
3-3 YZ planar view of MCNP5 simulation geometry with a single collimated detector voxel of air located over the heart.....	34
3-4 MCNP5 simulation collimators.....	34
3-5 Depiction of a photon passing through the minimum path length w from one collimator hole to another and being detected as a function of hole length l , hole diameter d , and septal thickness t	35
3-6 TITAN geometry for adjoint simulation using S_N and CM regions.....	35
3-7 Source directions chosen in four octants for S_{62} used in adjoint calculations to approximate the acceptance angle shown by the circle	36
4-1 Normalized TITAN projection images at four angles around the patient	48
4-2 Infinity-norm and 2-norm relative to 131 million photons per projection in SIMIND.....	49
4-3 SIMIND three energy group anterior projection images	49
4-4 TITAN three energy group anterior projection images.....	50
4-5 SIMIND and TITAN normalized projection images of the heart.	50
4-6 Plot of normalized flux along a single profile of anterior projection images created by the SIMIND and TITAN code along with the relative difference through A) column 44 and B) row 38	52
4-7 Plot of normalized flux along a single profile through row 38 of SIMIND and TITAN projection images and the relative difference	53
4-8 Group 1 adjoint flux through the heart from the TITAN model of a single-hole collimator	55

4-9	Single-hole collimator relative difference in TITAN response compared to MCNP5 as a function of direction weight (order of quadrature).....	56
4-10	Four-hole collimator relative error in TITAN response as a function of direction weight (order of quadrature).....	57
4-11	Nine-hole collimator relative error in TITAN response as a function of direction weight (order of quadrature).....	58
4-12	Optimal TITAN direction weight versus the cosine of the collimator acceptance angle in MCNP5	58
A-1	NCAT input file	63
A-2	CEPXS input file.....	65
A-3	TITAN input file for SPECT simulation.....	66
A-4	SIMIND parameter file.....	70
A-5	MCNP5 input file for the four-hole case.....	71
A-6	TITAN input file for adjoint methodology.....	74
B-1	Code to compare binary projection images from the TITAN and SIMIND codes.....	78
B-2	Program to calculate the detector response using the adjoint flux produced from an adjoint calculation in the TITAN code	81
B-3	Program to convert NCAT material distribution into material numbers for MCNP5 input.....	86
B-4	Linear regression program	89
B-5	Program to compare flux distributions of two different mesh sizes	91

Abstract of Thesis Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Master of Science

ANALYSIS OF TITAN HYBRID ALGORITHMS FOR SPECT SIMULATION

By

Katherine Elizabeth Keller Royston

May 2010

Chair: Alireza Haghighat
Major: Nuclear Engineering Sciences

Traditionally, Monte Carlo methods have been used in the simulation of Single Photon Emission Computed Tomography (SPECT), however computer advances have lead deterministic methodologies to become more feasible than previously. The TITAN code is a hybrid deterministic transport code that solves the linear Boltzmann equation using several approaches: the discrete ordinates method (S_N), the characteristics method (CM), and ray-tracing. While the S_N method is the most commonly used and generally more efficient, the MOC is best in low scatter regions. By allowing the use of these two methods in different regions of a problem, the TITAN code is more efficiently able to solve problems like SPECT simulation than a code allowing only one of the methods.

In this research, benchmarking of the TITAN code was performed on its application to a SPECT simulation of a myocardial perfusion study. The TITAN results were compared with two Monte Carlo codes: SIMIND and MCNP5. The SIMIND code was used as an established Monte Carlo SPECT simulation code for comparison with projection images created using TITAN. In the heart region, general agreement was found between the SIMIND and TITAN projection images with differences attributed

largely to the differences in cross sections used in the two codes. Examination of calculation times showed that for this model the TITAN code is 1.5 to 48 times faster than the SIMIND code, depending on the particular simulation parameters. The TITAN projection images were also compared for two types of differencing schemes and mesh sizes and several orders of quadrature. Neither of the two differencing schemes stood out as better than the other and the larger $0.625 \times 0.625 \times 0.625 \text{ cm}^3$ mesh size was adequate.

The adjoint methodology of TITAN was tested for its ability to represent a collimator by comparing detector response with the MCNP5 code. Several collimator hole sizes were used to examine potential limitations of the TITAN code. In each case, a quadrature that gave less than a 1% relative difference was found. Also, a linear relationship was demonstrated between the TITAN adjoint source direction weight and the response. A limitation of this adjoint methodology is the increased calculation time for the high orders of quadrature needed to represent increasingly small acceptance angles. However, the TITAN adjoint calculation was still demonstrated to be more than a factor of 3 faster than the MCNP5 response calculation time.

CHAPTER 1 INTRODUCTION

Motivation

Traditionally, Single Photon Emission Computed Tomography (SPECT) simulations have been conducted using Monte Carlo methods.¹ However Monte Carlo methods have the drawback of requiring large amounts of calculation time in general compared to deterministic methods.² Recently, the hybrid deterministic transport code TITAN has been applied to the simulation of SPECT³ with the hope that it can be used to develop methods of improving image quality and reducing patient dose. The immediate goal of this research is the benchmarking of the TITAN code for the simulation of a myocardial perfusion study. The results of the TITAN code will be compared with those of the SIMIND and MCNP5 codes. The effects of variations in mesh size, quadrature order, and differencing scheme on the results of the TITAN code will also be investigated.

SPECT Background

Nuclear medicine consists of two main imaging modalities: Single Photon Emission Computed Tomography (SPECT) and Positron Emission Tomography (PET). The SPECT images use radionuclides that decay by gamma ray emission, whereas PET uses positron emitting radionuclides. A general nuclear medicine study involves injecting a radiopharmaceutical into the patient. The radiopharmaceutical consists of a radionuclide tagged to a pharmaceutical that is designed to be preferentially absorbed by the region of interest. The radiation emitted from the patient can then be used to provide a functional image. Projection images of the patient are obtained using gamma cameras positioned at various angles around the patient, as shown in Figure 1-1. If

they have been obtained at enough angles, these projection images can then be used to reconstruct a 3-dimensional image of the patient.

The acquisition of a projection image in SPECT requires the use of a collimator in front of the gamma camera to provide spatial resolution. A collimator consists of a slab of a high z material, typically lead, with holes that are usually parallel. The collimator restricts photons that are counted to photons from the phantom along the detector normal from each collimator hole, within an acceptance angle. Rejection of photons outside of an energy window then reduces scatter counts. Unfortunately, these techniques also greatly reduce the number of counts at the detector and therefore increase the length of time needed to acquire an image of adequate quality.

Nuclear medicine images are obtained for many diagnostic studies including the determination of coronary artery disease, neurological disorders, and cancers. The radionuclide most commonly used in SPECT is ^{99m}Tc , which is frequently applied to measuring myocardial perfusion through the radiopharmaceutical Sestamibi. Myocardial perfusion studies determine if the myocardium, or heart wall, is receiving enough blood supply and can be used to diagnose coronary artery disease based upon how much of the radionuclide is absorbed into the myocardium. In this research, a myocardial perfusion study is simulated by placing a uniform source in the heart wall of the phantom.

NCAT Code

To provide a phantom of a patient for the SPECT simulations, the NURBS-based cardiac-torso (NCAT) code⁴ was used. The NCAT code uses non-uniform rational B-splines (NURBS) to generate the phantom organs based upon data from the Visible Human Project. An input file is used to specify all the details of the desired phantom

and the NCAT code then outputs two distributions: a material distribution and a source distribution. The material distribution contains the linear attenuation coefficient of each of the materials in the phantom and the source distribution contains the relative intensity of each voxel.

For the simulations completed in this work, a $40 \times 40 \times 40 \text{ cm}^3$ phantom simulating an upper torso was generated using the NCAT code with default inputs. Two voxel sizes were used: $0.625 \times 0.625 \times 0.625 \text{ cm}^3$, which results in $64 \times 64 \times 64$ voxels in the phantom, and $0.3125 \times 0.3125 \times 0.3125 \text{ cm}^3$ for a $128 \times 128 \times 128$ voxel phantom. Thirteen different materials were represented including: air, water (body), intestine, lung, rib bone, liver, kidney, spine bone, muscle, cartilage, spleen, heart, and blood. Figure 1-2 is a visual representation of the phantom with the top cut away so that the internal organs are visible. In the source distribution, a 140.5 keV source was placed in the heart wall to simulate uptake of ^{99m}Tc with an intensity of 75 specified in the NCAT input file. A far less intense source with a specified intensity of 2 was placed in the body, muscle, and blood to account for some radiopharmaceutical uptake of these materials as shown in Figure 1-3.

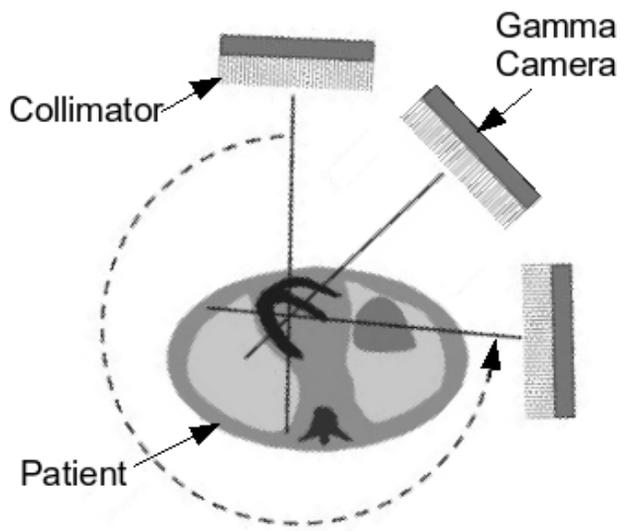


Figure 1-1. Geometry of a SPECT study

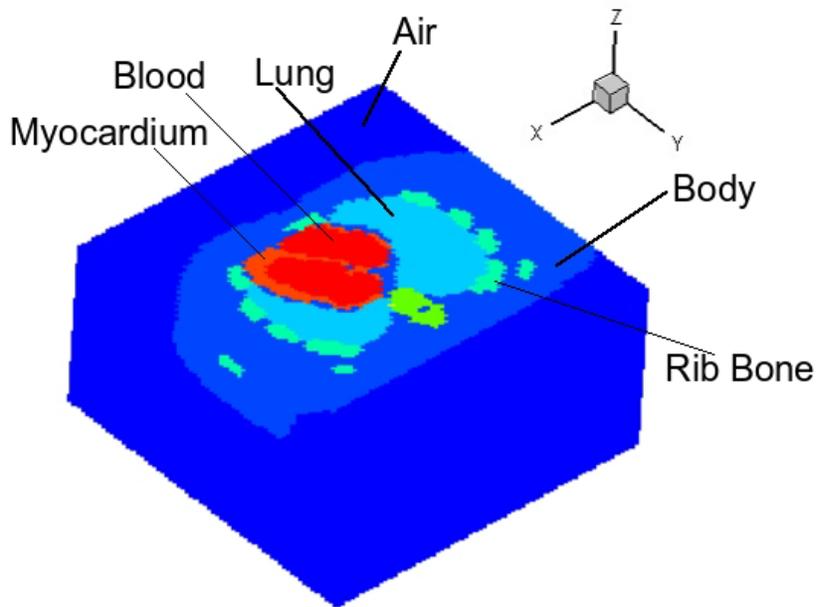


Figure 1-2. Cut-away view of NCAT material distribution

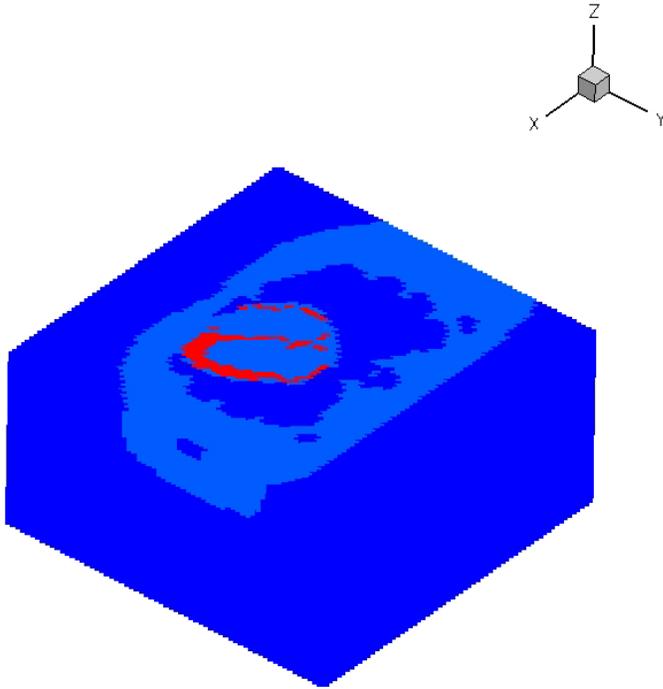


Figure 1-3. Cut-away view of NCAT source distribution. The red is the intense source in the myocardium and the light blue is a less intense source.

CHAPTER 2 THEORY

TITAN

The TITAN code is a hybrid deterministic transport code that solves the linear Boltzmann equation (LBE).⁵ The code is a hybrid because it allows the use of different solvers of the LBE in different regions (or coarse meshes) of a problem: a Discrete Ordinates (S_N)² solver, a Characteristics Method (CM)⁶ solver, and a ray-tracing solver.⁶ The TITAN code is especially applicable in problems with low scattering regions, as is found in the SPECT problem, where the CM and ray-tracing solver can be used to speed up the calculations.

Deterministic Transport Code

Deterministic codes seek to numerically solve the LBE for neutral particles, shown in time-independent integro-differential form in Equation 2-1 for a non-multiplying medium.⁷ The LBE is derived from a balance of particles in a phase space as given by $d^3r dE d\Omega$.

$$H\psi = S \text{ in } V \quad (2-1)$$

where,

$$H = \underbrace{\sigma_t}_{\text{loss}}(\mathbf{r}, E) + \hat{\Omega} \cdot \nabla - \int_0^\infty dE' \int_{4\pi} d\hat{\Omega}' \underbrace{\sigma_s}_{\text{gain}}(\mathbf{r}, E' \rightarrow E, \hat{\Omega}' \rightarrow \hat{\Omega})$$

In Equation 2-1, ψ is the angular flux, σ_t is the total macroscopic cross-section, $\sigma_s(\mathbf{r}, E' \rightarrow E, \hat{\Omega}' \rightarrow \hat{\Omega})$ is the differential macroscopic scattering cross-section and S is the independent source. The first two terms on the left side of Equation 2-1 are the loss from the phase space and called the collision and streaming terms, respectively. The third term on the left side and the right-hand side of Equation 2-1 are the gains to the

phase space: the scattering and independent source terms, respectively. To solve this equation, the phase space can be discretized. Energy is generally discretized using the multigroup approximation⁷ in deterministic methods and the angular domain is discretized into discrete ordinates or S_N .² In the TITAN code, the spatial domain is discretized using either meshes for the Discrete Ordinates method (S_N), or arbitrarily shaped regions for the Characteristics Method (CM) and ray-tracing. The differential scattering cross-section is also expanded in terms of Legendre polynomials $P_l(\mu_o)$, where $\mu_o = \hat{\Omega} \cdot \hat{\Omega}'$, as shown in Equation 2-2.

$$\sigma_s(\vec{r}, E' \rightarrow E, \mu_o) = \sum_{l=0}^{\infty} \frac{2l+1}{4\pi} \sigma_{sl}(\vec{r}, E' \rightarrow E) P_l(\mu_o) \quad (2-2)$$

Equation 2-2 is then truncated at some order L according to the degree of anisotropy in the particular problem of interest.

In regions defined by the user, the TITAN code enables the use of the S_N method to solve the LBE. The S_N method solves Equation 2-1 for the angular flux in a discrete set of directions.² N is called the quadrature order and in a 3-D geometry there are $N(N+2)/8$ directions per octant. Each direction is weighted to preserve the moments of angular flux (e.g. flux and current), and the combination of the directions and their weights is called the quadrature set. Differencing schemes must then be used to deal with the streaming term and solve the LBE for the quadrature set.

The CM is especially useful in low scattering regions. The same angular quadrature method used to discretize the angle in the S_N method is usually used, however instead of using spatial meshes, the LBE is solved along parallel directions called characteristic rays.⁵ The integral transport equation can then be used to calculate the angular flux along a characteristic ray. A uniform fine mesh is built on the

coarse mesh boundaries and characteristic rays are drawn along the quadrature directions from the centers of the fine meshes.⁶ The CM does not require a differencing scheme to solve the transport equation, but can be very memory intensive.

Differencing Schemes

If the S_N method is being used to solve the LBE, a spatial approximation is required and the problem geometry is divided up into cell volume elements. The TITAN code allows the use of two spatial differencing schemes: diamond differencing with zero fix-up (DDZ)⁸ and directional theta-weighted (DTW).⁹ The diamond differencing method (DD) simply uses a linear average of two opposite boundary angular fluxes to determine the cell-average angular flux. Equation 2-3 gives an example of this formulation for 3D Cartesian coordinates along the x-axis where ϕ_{gnijk} is the cell average angular flux in group g , direction n , and in spatial mesh ijk .

$$\phi_{gnijk} = \frac{1}{2} \left(\phi_{gn,i-1/2,jk} + \phi_{gn,i+1/2,jk} \right) \quad (2-3)$$

In Equation 2-3, $\phi_{gn,i-1/2,jk}$ and $\phi_{gn,i+1/2,jk}$ are the boundary angular fluxes (or incoming and outgoing fluxes) on either side of the cell-average angular flux. Similar equations are used along the y-axis and z-axis.

In a typical problem, the incoming flux will be known from boundary conditions so that the outgoing and cell-average angular flux can be solved for. As can be seen from Equation 2-3, it is possible to obtain negative outgoing angular flux values from the DD scheme, especially in regions with steep flux gradients. One method of dealing with this is DDZ and entails setting the negative flux value to zero and recalculating the cell-average angular flux before continuing with the calculation. This method is expensive

and leads to numerical difficulties.⁸ Hence, Petrovic and Haghghat developed a positive method called DTW (directional theta-weighted).

The DTW scheme does not equally weight the incoming and outgoing boundary angular fluxes, as the DDZ scheme does. Instead the fluxes are weighted as a function of space, energy, and angle. Equation 2-4 gives the weighting scheme for 3D Cartesian coordinates along the x-axis for a cell, where the weighting coefficient (for the DTW scheme), a , is calculated according to Equation 2-5.

$$\phi = a \phi_{out\ x} + (1 - a) \phi_{in\ x} \quad (2-4)$$

$$a = 1 - \frac{\frac{|\mu_n|}{\Delta x} \phi_{in,x} + \theta(\mu_n) \left(\frac{|\eta_n|}{\Delta y} \phi_{in,y} + \frac{|\xi_n|}{\Delta z} \phi_{in,z} \right) + q_n}{(\sigma + 2 \frac{|\eta_n|}{\Delta y} + 2 \frac{|\xi_n|}{\Delta z}) \phi_{in,x}} \quad (2-5)$$

$$\theta(\mu_n) = \mu_n^2 \quad (2-6)$$

Similar weights of b and c are calculated along the y-axis and z-axis and all three weights depend on a θ parameter, given in Equation 2-6 for the x-axis. In Equation 2-5, μ_n , η_n and ξ_n are the direction cosines for direction index n and q_n contains the scattering and independent source terms.

Adjoint Transport Equation

The adjoint equation is derived using the adjoint property⁷ given in Equation 2-7, where \dagger symbolizes adjoint and $\langle \rangle$ symbolizes integration over all independent variables.

$$\langle \psi^\dagger H \psi \rangle = \langle \psi H^\dagger \psi^\dagger \rangle \quad (2-7)$$

In Equation 2-7, H is the transport operator, H^\dagger is the adjoint transport operator, and ψ^\dagger is the adjoint flux. The adjoint identity requires the boundary condition that the flux of incoming particles and the adjoint flux of outgoing particles are both zero (vacuum

boundary condition). The adjoint form of the transport equation for the time-independent and non-multiplying case is given by Equation 2-8.⁷

$$H^\dagger \psi^\dagger = S^\dagger \text{ in } V \quad (2-8)$$

where,

$$H^\dagger = \sigma_t(r, E) - \hat{\Omega} \cdot \nabla - \int_0^\infty dE' \int_{4\pi} d\hat{\Omega}' \sigma_s(r, E \rightarrow E', \hat{\Omega} \rightarrow \hat{\Omega}')$$

In the adjoint equation $\sigma_s(r, E \rightarrow E', \hat{\Omega} \rightarrow \hat{\Omega}')$ is the scattering cross-section from energy E to E' and direction $\hat{\Omega}$ to $\hat{\Omega}'$, $S^\dagger(r, E, \hat{\Omega})$ is the adjoint source, and the rest of the terms are defined as in the LBE. If the adjoint source is defined as a detector cross-section, the adjoint flux will be a measure of the contribution, or importance, to that detector response.

Monte Carlo

Monte Carlo is a method that involves simulating each particle history separately and calculating some average behavior usually using tallies, in contrast with deterministic methods that seek to solve the transport equation for an average value. The method is highly dependent on having a good random number generator to sample the probability density functions (PDFs) of the particle's interaction types, path length between interactions, scattering angle, etc. In general, Monte Carlo codes may require less computer memory than deterministic codes but are far more computationally expensive to complete calculations with an acceptable relative error.² Monte Carlo methods allow for continuous treatment of the position, energy, and angle variables and are generally more accurate. As a result of having continuous energy, continuous (or point-wise) energy cross sections can be used, rather than the approximated multigroup cross sections used in deterministic codes. Monte Carlo methods also have the

advantage of being naturally parallel since each particle transport is an independent simulation.

The term “analog Monte Carlo” refers to methods that do not use any variance reduction techniques. In “non-analog” Monte Carlo, variance reduction is used to speed up the problem by biasing the PDF to increase the probability that a particle contributes to the desired quantity. The outcome is then preserved by applying a weight (w) to the biased result as shown in Equation 2-9.

$$w_{biased} \times pdf_{biased} = w_{unbiased} \times pdf_{unbiased} \quad (2-9)$$

Non-analog Monte Carlo is frequently necessary to obtain a precise result in a reasonable amount of time. As long as the particle weights are conserved, as in Equation 2-9, non-analog Monte Carlo can produce accurate results, however it must be used with caution to avoid unintentional biasing of results.

In Monte Carlo methods, the statistical error corresponding to the final result is of the utmost importance to know how precise the result is.

$$S^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2 \quad (2-10)$$

The sample variance is calculated according to Equation 2-10, where \bar{x} is the sample mean and N is the number of histories.¹ The sample variance of the mean is then given by the Central Limit Theorem¹ to be Equation 2-11.

$$S_{\bar{x}}^2 = \frac{S^2}{N} \quad (2-11)$$

The relative error can then be estimated using Equation 2-12.

$$R = \frac{S_{\bar{x}}}{\bar{x}} \quad (2-12)$$

Multigroup Approximation

In order to solve the LBE, deterministic methods use the multigroup approximation to discretize the energy variable into energy groups ($g=1, G$). Within an energy group, the multigroup approximation assumes that all particles interact with the same probability. Equation 2-13 gives the multigroup transport equation.⁷

$$\hat{\Omega} \cdot \nabla \psi_g(\vec{r}, \hat{\Omega}) + \sigma_g(\vec{r}) \psi_g(\vec{r}, \hat{\Omega}) = \sum_{g'=1}^g \int_{4\pi} d\Omega' \sigma_{sg' \rightarrow g}(\vec{r}, \hat{\Omega} \cdot \hat{\Omega}') \psi_{g'}(\vec{r}, \hat{\Omega}') + S_g(\vec{r}, \hat{\Omega}) \quad (2-13)$$

In Equation 2-13, the subscript g denotes the energy group and the angular flux for group g is given by Equation 2-14.

$$\psi_g(\vec{r}, \hat{\Omega}) = \int_{E_g}^{E_{g-1}} dE \psi(\vec{r}, \hat{\Omega}, E) \quad (2-14)$$

The multigroup cross sections are defined in Equations 2-15 and 2-16.

$$\sigma_g(\vec{r}) = \frac{1}{\psi_g(\vec{r}, \hat{\Omega})} \int_{E_g}^{E_{g-1}} dE \sigma_t(\vec{r}, E) \psi(\vec{r}, E, \hat{\Omega}) \quad (2-15)$$

$$\sigma_{sg' \rightarrow g}(\vec{r}, \hat{\Omega} \cdot \hat{\Omega}') = \frac{1}{\psi_g(\vec{r}, \hat{\Omega})} \int_{E_g}^{E_{g-1}} dE \int_{E_{g'}}^{E_{g'-1}} dE' \sigma_s(\vec{r}, E' \rightarrow E, \hat{\Omega}' \cdot \hat{\Omega}) \psi(\vec{r}, E', \hat{\Omega}') \quad (2-16)$$

Finally, the multigroup source is defined in Equation 2-17.

$$S_g(\vec{r}, \hat{\Omega}) = \int_{E_g}^{E_{g-1}} dE S(\vec{r}, \hat{\Omega}, E) \quad (2-17)$$

By convention, the highest energy group is group 1 and the lower groups follow down to group G .

The differential scattering cross-section $\sigma_{sg' \rightarrow g}(\vec{r}, \hat{\Omega}' \cdot \hat{\Omega})$ is expanded in the orthogonal Legendre polynomials $P_l(\mu_o)^2$, where $\mu_o = \hat{\Omega}' \cdot \hat{\Omega}$, in Equation 2-18.

$$\sigma_{s_{lg'} \rightarrow g}(\bar{r}, \mu_0) = \sum_{l=0}^{\infty} \frac{2l+1}{4\pi} \sigma_{s_{lg'} \rightarrow g}(\bar{r}) P_l(\mu_0) \quad (2-18)$$

Using the orthogonal properties of the Legendre polynomials, Equation 2-19 can then be written.

$$\sigma_{s_{lg'} \rightarrow g}(\bar{r}) = \frac{\int_{E_g}^{E_{g-1}} dE \int_{E_{g'}}^{E_{g'-1}} dE' \sigma_{sl}(E' \rightarrow E) \varphi(E')}{\int_{E_{g'}}^{E_{g'-1}} dE' \varphi(E')} \quad (2-19)$$

In Equation 2-19, φ is the scalar flux and weights the differential scattering cross-section.

To generate the multigroup cross sections, the CEPXS code¹⁰ was used. CEPXS was designed to create multigroup-Legendre cross sections for use in coupled electron-photon transport calculations with the ONEDANT code. CEPXS calculates the cross sections of arbitrary materials using the cross sections of the constituent elements according to Equation 2-20.

$$\sigma = \sum_i N_i \frac{A_i}{A} \sigma_i = \sum_i w_i \sigma_i \quad (2-20)$$

In Equation 2-20, N_i is the number fraction of element i , A_i is the atomic weight of element i , w_i is the weight fraction of element i , and σ_i is the macroscopic cross-section of element i . The CEPXS code uses a zero order weighting function in Equation 2-19, because it does not have any knowledge of the scalar flux. Equation 2-19 is then written as Equation 2-21.

$$\sigma_{s_{lg'} \rightarrow g}(\bar{r}) = \frac{\int_{E_g}^{E_{g-1}} dE \int_{E_{g'}}^{E_{g'-1}} dE' \sigma_{sl}(E' \rightarrow E)}{E_{g'-1} - E_{g'}} \quad (2-21)$$

CHAPTER 3 METHODOLOGY

In order to simulate a myocardial perfusion study in each of the codes, some code specific modifications had to be made. The following sections describe the methods used to perform a variety of SPECT simulations in the TITAN, SIMIND, and MCNP5 codes.

General Simulation Methods

Each single photon emission computed tomography (SPECT) simulation produces projection images like those that would be obtained by a gamma camera placed at several angles around a patient injected with ^{99m}Tc . Four projection angles were simulated in both the SIMIND and the TITAN codes: anterior, left lateral, posterior, and right lateral. In MCNP5, only the anterior view was simulated due to time limitations. Projection images plot pixel (1,1) in the upper left corner as depicted in Figure 3-1. The multigroup calculations used three energy bins and cross sections generated using the CEPXS code with ICRP Publication 89¹¹ data on elemental composition of body tissues. Technetium-99m emits a 140.5 keV gamma ray and SPECT systems typically have an energy resolution of approximately 10%¹², so the group 1 bounds were chosen to be 10% above and below the source energy (126.45-154.55 keV). The second group range was chosen to have the same width as the first (98.35-126.45 keV) and the third group then extended to the cutoff energy of 10 keV.

Projection images were normalized to the highest pixel value. In the projection images, it is clear that the peripheral pixels will receive few counts. These low count pixels are in unimportant regions far from the source and will necessarily have large errors. Also, in a myocardial perfusion study only accuracy in the heart is sought and

other regions of the body can be disregarded. For these reasons, numerical comparison between projections was confined to pixels with normalized values greater than 0.65 in the heart region. This cutoff value was chosen to restrict the analysis to the high-count pixels in the myocardium. Quantitative comparisons between different projection images at the same angle were made using the *infinity-norm* and *2-norm* given by Equations 3-1 and 3-2.

$$\text{Infinity - norm} = \max[(P'_{ij} - P_{ij})/P_{ij}] \quad (3-1)$$

$$2\text{-norm} = \left[\sum_{i,j} \left(\frac{P'_{ij} - P_{ij}}{P_{ij}} \right)^2 \right]^{1/2} \quad (3-2)$$

In Equations 3-1 and 3-2, P_{ij} is the pixel value in position (i,j) of the projection image.

TITAN Methods

The TITAN code allowed for direct use of the NCAT code's material and source distributions as well as the multigroup cross sections generated using the CEPXS code. The flux distributions of all energy groups are solved for in one calculation and a projection image is output for each energy group at each specified projection angle. In TITAN SPECT simulations, the S_N method was used to solve the LBE within the NCAT phantom and the ray-tracing method was used to transport photons from the phantom surface to the detector. For each projection angle, TITAN uses a fictitious quadrature technique to calculate the boundary angular flux along the projection angle.³

The detector is not directly simulated in the TITAN code, however the effects of a collimator are incorporated using an ordinate splitting (OS) technique.¹³ The OS technique involves further splitting a chosen direction in a quadrature set. Ordinate splitting can be applied when a highly peaked angular dependent flux or source is

present in a problem. To simulate a SPECT collimator, the TITAN code implements the circular ordinate splitting technique³ in which the split directions are located on the edge of a circle centered on the original projection direction as depicted in Figure 3-2. The SPECT collimator acceptance angle determines the radius of the circle. The angular flux reaching the detector location is then averaged over the angular fluxes of the original and split directions. In this way, collimator blur is accounted for in the simulated projection image. A limitation of this method is that interactions with the collimator septa are not simulated.

The sensitivity of the results of the TITAN code to quadrature order, differencing scheme, and mesh size was investigated with the 64 x 64 x 64 and 128 x 128 x 128 voxel NCAT phantoms. The TITAN code allows the user to choose either the DDZ or DTW differencing scheme. These differencing schemes were used in conjunction with different quadrature orders of S_4 , S_6 , S_8 , S_{10} , and S_{12} , which correspond to 3, 6, 10, 15, and 21 directions per octant, respectively. A brief comparison of the flux distributions in the 64 x 64 x 64 and 128 x 128 x 128 phantoms is given in Chapter 4.

SIMIND Methods

The SIMIND code is a Monte Carlo code designed specifically for SPECT simulation.¹⁴ The code is able to directly use the NCAT material and source distributions for the phantom, however it does not use any sort of provided cross-section file. An input file allows the specification of many parameters, including collimator details. To match the TITAN simulation as closely as possible, the modeling of a collimator was turned off in the SIMIND code and an acceptance angle equal to that used in TITAN (0.66°) was specified. To obtain results for each energy group, the code was executed separately with an energy window set to contain the energy group

desired. As output, the SIMIND code produces a projection image for each angle that has been simulated.

A major drawback of the SIMIND code was the fact that it does not provide any statistical output. Because of this, the first step with using this code was to determine the appropriate number of simulated source particles needed to converge on a solution. To achieve this, calculations with increasing numbers of particles were performed and compared using an infinity-norm and a 2-norm. Once it was determined that the code was converging on a solution, the number of particles needed to obtain an appropriate level of precision could be found. The projection images produced by the SIMIND code were then compared with those created in TITAN as described at the beginning of this Chapter.

MCNP5 Methods

The Monte Carlo N-Particle, Version 5 (MCNP5) code is a general-purpose Monte Carlo code for neutron, photon, and electron transport developed by Los Alamos National Laboratory.¹⁵ The MCNP5 model used a lattice of universes containing different materials, ultimately resulting in the same 40 x 40 x 40 cm³ phantom that the SIMIND and TITAN codes read in. Only the 64 x 64 x 64 voxel phantom was used in the MCNP5 simulations. A single 0.625 x 0.625 x 0.625 cm³ detector voxel of air was positioned over the heart and 10 cm from the phantom surface with a collimator placed in front of it. Figure 3-3 shows the MCNP5 geometry.

In the MCNP5 code, the collimator was modeled as a lead, parallel hole collimator with circular holes and a depth of 5.7 cm. Three collimator hole configurations were used and these are displayed in Figure 3-4. There were single-hole, four-hole, and nine-hole collimators with acceptance angles of 3.14°, 1.42°, and 0.98°, respectively, in

the 0.625 x 0.625 cm² face of the lead. The acceptance angle θ was defined as $\tan^{-1}(r/l)$, in which r is the collimator hole radius and l is the collimator hole length.

To effectively eliminate any stray photons from reaching the single voxel detector by traveling through the collimator walls or outside of the collimator, a 0.625 cm thickness of lead was added around the collimator and detector. The minimum thickness between adjacent holes was calculated to limit septal penetration to less than 5%.¹² Figure 3-5 depicts a photon passing through a collimator septum with the shortest path length possible to reach the detector, w . Using this figure, an approximate relationship can be made between the septal thickness t , the collimator hole diameter d , and the collimator hole length l .

$$\frac{t}{w} \approx \frac{2d}{l-w} \quad (3-3)$$

To limit the septal penetration to 5%, exponential attenuation was used to obtain Equation 3-4.

$$w \geq 3/\mu \quad (3-4)$$

In Equation 3, μ is the linear attenuation coefficient of the septal material. Placing Equation 3-4 into Equation 3-3 and rearranging then gives Equation 3-5.

$$t \geq 6d/(\mu l - 3) \quad (3-5)$$

The detector response R (counts/s) of a detector with cross-section σ_d was calculated from the flux distribution according to Equation 3-6.

$$R = \iiint_{\text{Detector}} dE dV \varphi(\vec{r}, E) S^{\dagger}(\vec{r}, E) = \sum_{\substack{g=1,G \\ \Delta V_i \in V_d}} \varphi_{g,i} \sigma_{d,g,i} \Delta V_i \quad (3-6)$$

In Equation 3-6, φ is the scalar flux, S^{\dagger} is the adjoint source density, V_d is the detector volume, and ΔV_i is the volume of cell i . The adjoint source density was chosen to be 1

in each energy group for simplicity, because the purpose of this study is to compare the responses not obtain the exact value. A flux tally was used in the detector volume to obtain the scalar flux for each energy group.

Adjoint Methods

Adjoint calculations were used to compare the detector response from the phantom in the TITAN code simulations to the response determined from the MCNP5 code described in the previous section. These comparisons were made for a single voxel detector, which was the adjoint source. Figure 3-6 displays this configuration with the S_N and CM regions used in the TITAN code labeled. The purpose of this comparison was to develop and validate an adjoint method of representing the collimator in the TITAN code.

The deterministic TITAN code only solves the LBE for a limited number of directions. However, since the detector response is limited to particles approaching within the acceptance angle of the simulated collimator, the adjoint source needed to be restricted to within this same angle. To achieve this, angles much smaller than those found in S_{12} were needed and therefore high orders of quadrature were used. Some experimentation was done to determine the best quadrature choice. In each case, a quadrature was used that contained only one direction within the acceptance angle along the +y-axis. The adjoint source was then defined along that direction in the four octants that face the +y-axis. Figure 3-7 demonstrates how these four directions approximate the collimator acceptance angle. The ordinate splitting technique was also used to determine if increasing the number of directions within the acceptance angle improved the accuracy of the response.

The TITAN adjoint model of collimation is fundamentally different from the MCNP5 collimator model. However, it is interesting to consider the area subtended by the solid angle $\Delta\Omega$ for comparison with the area of the opening in the MCNP5 collimator holes. The quadrature direction weight, w , is directly related to $\Delta\Omega$ and the area, A' , as shown in Equations 3-7 through 3-9.

$$w = \frac{\Delta\Omega}{\pi/2} \quad (3-7)$$

$$\Delta\Omega = \frac{A'}{r^2} \quad (3-8)$$

$$w = \left(\frac{2}{\pi}\right) \frac{A'}{r^2} \quad (3-9)$$

In Equations 3-7 through 3-9, r is the collimator thickness and the $\pi/2$ is a normalization factor for the solid angle of an octant. Since the source will be specified in four octants, the total subtended area can finally be found according to Equation 3-10.

$$A = 4A' = 4\left(\frac{\pi}{2}\right)r^2w \quad (3-10)$$

By using this linear relationship between area and direction weight, the relative difference in the response of the TITAN for different orders of quadrature can be viewed as a function of area.

The adjoint calculation in the TITAN code results in one adjoint flux distribution for each energy group. The response of a detector placed at the location of the adjoint source can then be calculated according to Equation 3-11.

$$R = \iiint dE dV \phi^\dagger(\vec{r}, E) S(\vec{r}, E) = \sum_{\substack{g=1,G \\ \Delta V_i \in V_s}} \phi_{d,g,i}^\dagger S_{g,i} \Delta V_i \quad (3-11)$$

In Equation 3-11, ϕ^\dagger is the adjoint flux and S is the source density. The MCNP5 response with collimation will have an entrance surface area into the collimator that is

determined by the number and size of the collimator holes. However, the TITAN code's adjoint source will be emitted from the entire surface of the detector voxel. To account for this difference, a ratio of surface areas was multiplied by the TITAN response.

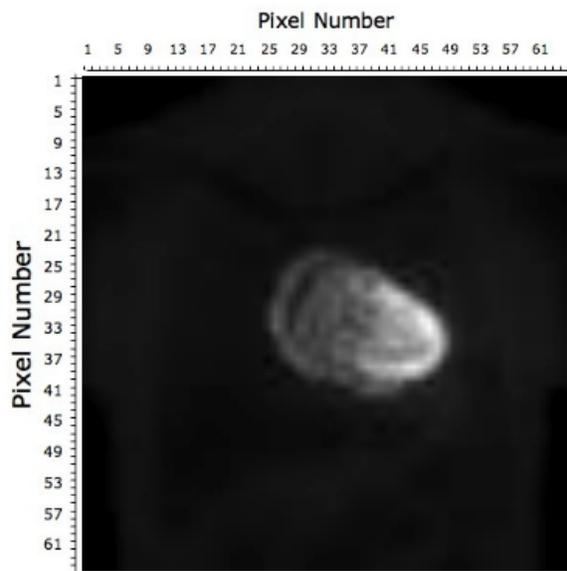


Figure 3-1. Example of pixel numbering in projection images

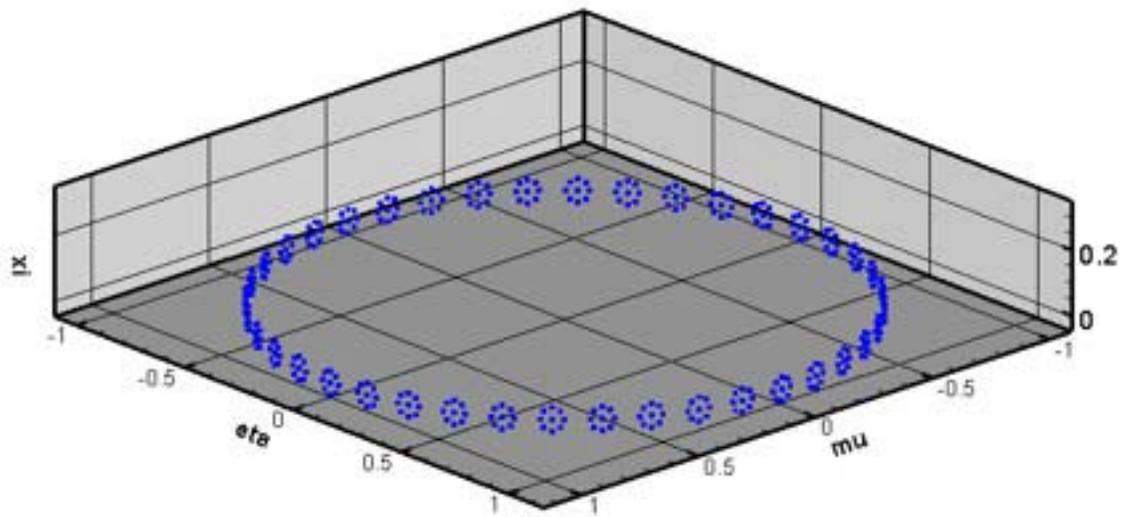


Figure 3-2. Schematic of a fictitious quadrature set with circular ordinate splitting

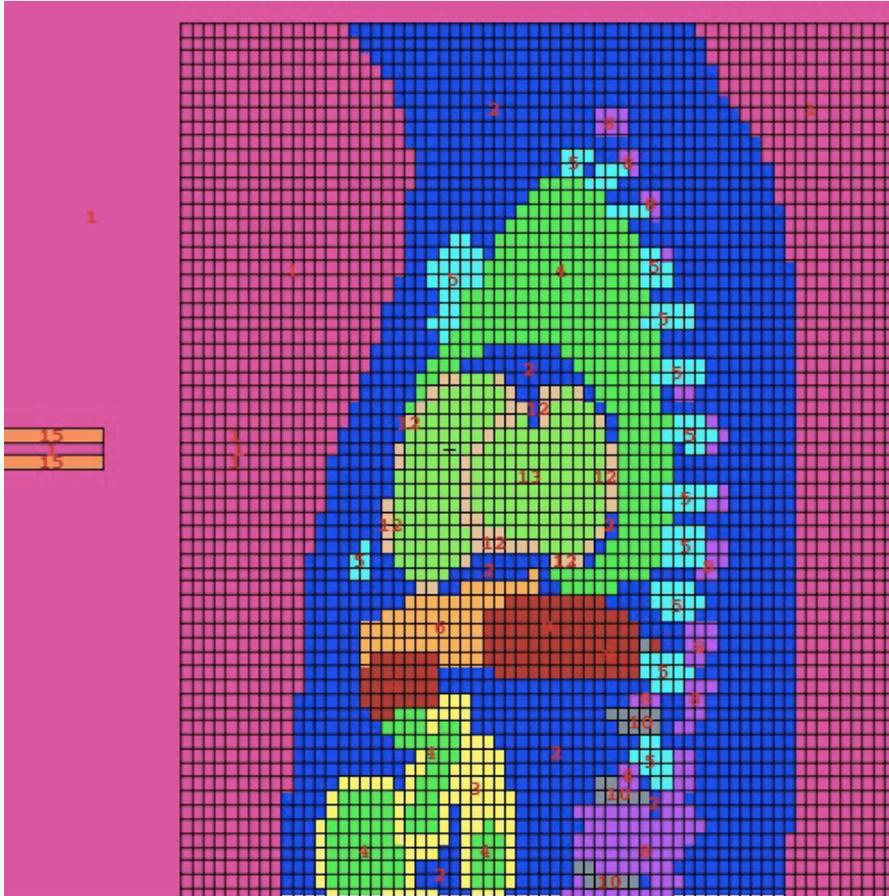


Figure 3-3. YZ planar view of MCNP5 simulation geometry with a single collimated detector voxel of air located over the heart

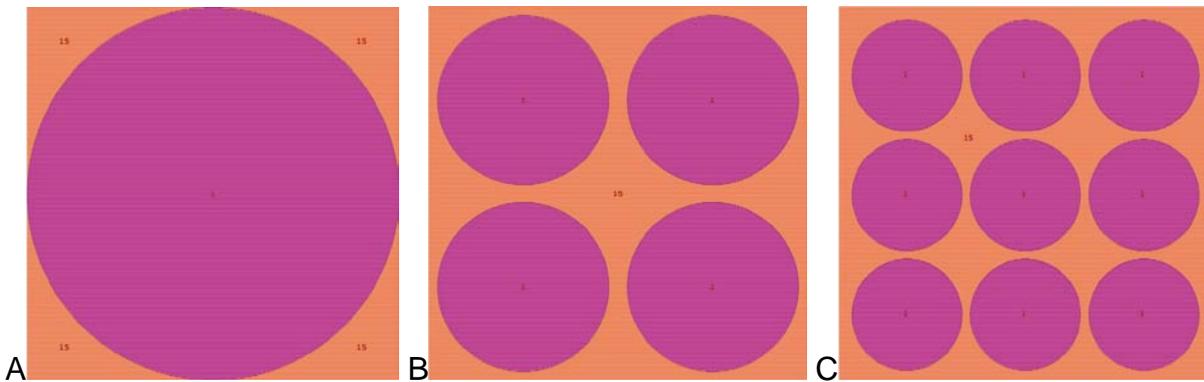


Figure 3-4. MCNP5 simulation collimators A) Single-hole collimator with 3.14° acceptance angle B) Four-hole collimator with 1.42° acceptance angle C) Nine-hole collimator with 0.98° acceptance angle

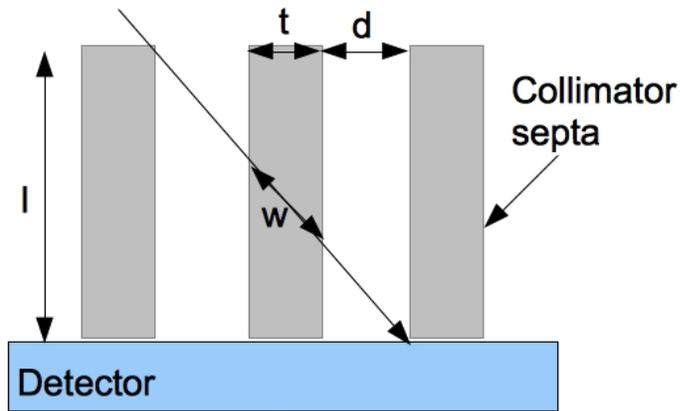


Figure 3-5. Depiction of a photon passing through the minimum path length w from one collimator hole to another and being detected as a function of hole length l , hole diameter d , and septal thickness t

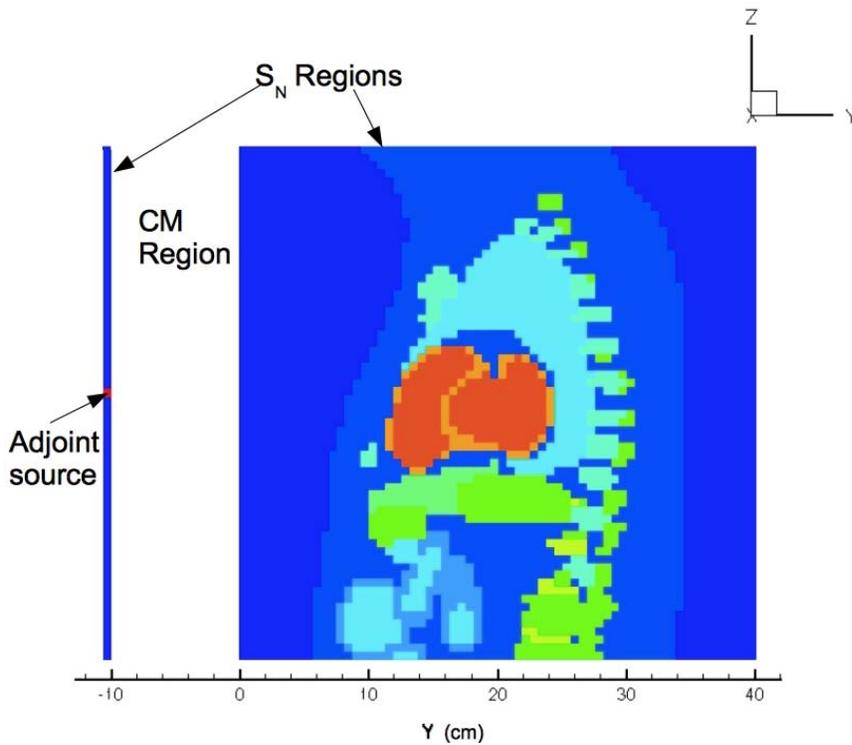


Figure 3-6. TITAN geometry for adjoint simulation using S_N and CM regions

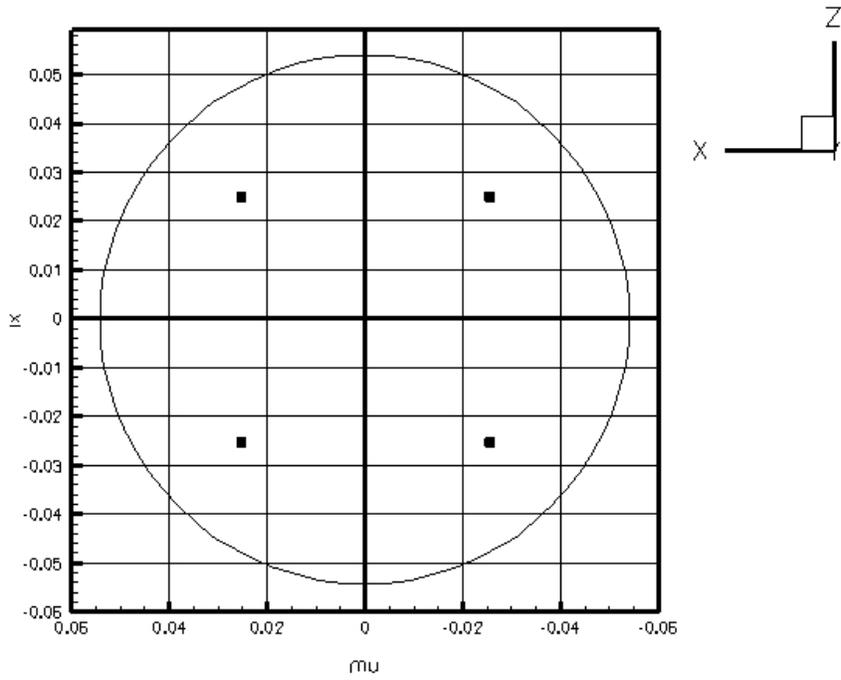


Figure 3-7. Source directions chosen in four octants for S_{62} used in adjoint calculations to approximate the acceptance angle shown by the circle

CHAPTER 4 RESULTS

In this section, the previously described methodologies are put into practice. The projection images produced by TITAN in a SPECT simulation are compared for increasing orders of quadrature for two differencing schemes and mesh cases. The SIMIND code is analyzed to verify that enough particles are being simulated and that the solution is converging. The projection images from the SIMIND and TITAN codes are compared both visually and quantitatively for four projection angles. Differences between images produced for each energy group are also examined. For several chosen cases, the computation times of the TITAN and SIMIND codes are compared. All of the above calculations were completed in serial using the PC-cluster designated Einstein. Einstein contains eight nodes with two processors per node. Each processor is a 2.4 GHz AMD Dual Opteron processor and each node contains 4 GB of DDR SDRAM.

For the adjoint methodology in TITAN, the detector responses from several different orders of quadrature are compared to the “forward” calculations performed by the MCNP5 code for each collimator case. This allows an optimal order of quadrature for a particular collimator simulation to be found for each case. An attempt is then made to develop a methodology to recommend the best order of quadrature to use in the TITAN code, for a given collimator acceptance angle from the MCNP5 collimator models. Finally, the calculation times for each collimator case are compared. These calculations were performed on the PC-cluster designated Dirac. Dirac contains 16 nodes with eight processors per node. Each processor is a 2 GHz Intel Xeon Quad Core processor with 4 GB of DDR2 SDRAM.

TITAN Analysis

Normalized group 1 projection images created by TITAN at four different angles around the patient are displayed in Figure 4-1. The sensitivity of TITAN's results to varying orders of quadrature for two differencing schemes and mesh sizes was examined. Anterior projection images created using the TITAN code were compared for the diamond differencing with zero fix-up (DDZ) and directional theta-weighted (DTW) differencing schemes and $64 \times 64 \times 64$ and $128 \times 128 \times 128$ mesh cases. Results are given in Table 4-1 for each projection image compared to the next higher order of quadrature. The infinity-norm and 2-norm values demonstrate that for either mesh size, the DTW differencing scheme converges to its solution at a lower order of quadrature than the DDZ method. It can be seen that both differencing schemes maintained similar infinity-norms between the $64 \times 64 \times 64$ and $128 \times 128 \times 128$ mesh cases. The infinity-norm values, regardless of the particular case, indicate that a low order of quadrature is sufficient for this simulation.

The differences in the flux distributions for the $64 \times 64 \times 64$ (large-mesh) and $128 \times 128 \times 128$ (small-mesh) phantoms in TITAN were examined. Since each voxel in the large-mesh phantom will correspond to 8 voxels in the small-mesh phantom, two steps were taken to compare these cases. First, only voxels with the heart wall material in them were compared. This means that for a single large-mesh voxel there could be anywhere from 0 to 8 small-mesh voxels compared to it. Second, the relative difference for a large-mesh voxel was defined as the average of the absolute relative difference between the large-mesh voxel and each corresponding small-mesh voxel meeting the first step. The maximum absolute differences are given in Table 4-2 for each differencing scheme and quadrature order used. No consistent pattern of increasing or

decreasing absolute relative differences is seen with increasing order of quadrature. The DDZ and DTW schemes also do not show any pattern to indicate that one of them has higher differences between the two voxel sizes. The differences are all small (<0.2%) and the same order of magnitude, which indicates good agreement between the two models regardless of differencing scheme and quadrature order.

SIMIND Analysis and Comparison

In order to determine the number of particles needed to converge on a solution with no statistical data from SIMIND, several calculations were performed with an increasing number of particles simulated. Figure 4-2 shows the infinity-norm and 2-norm relative to a run of 131 million photons per projection as a function of increasing number of photons per projection image. It is clear from the trend of the plot that the SIMIND code's solution is converging. The appropriate number of photons per projection needed will depend on the level of precision desired

Visual comparison of projection images produced by the SIMIND and TITAN codes revealed some notable differences in the lower energy groups. Figures 4-3 and 4-4 display the anterior projection images produced by the SIMIND and TITAN codes respectively for each of the three energy groups simulated. Due to photoelectric absorption, dense and high Z materials, like bone, will greatly reduce the number of photons reaching the detector, especially at low energies. In the TITAN projection images this effect is clearly visible, however it is not so apparent in the SIMIND code images. Figures 4-3B and 4-3C show the SIMIND projection images for energy groups 2 and 3 with the expected attenuation in bone not visible and the photons appearing to have spread out in the phantom. The group 3 SIMIND image shows a faint shadowing of the rib cage. The TITAN energy group 2 projection image, Figure 4-4B, shows a light

outline of the rib cage and Figure 4-4C shows the TITAN group 3 result with the clavicle, sternum and ribs clearly visible. This matches well with the drastic increase in attenuation that is expected with decreasing photon energy due to the photoelectric effect. As mentioned previously, the SIMIND input does not include any cross sections and so its calculations are based on its knowledge of the material linear attenuation coefficients. The limited information SIMIND works with appears to leave the code unable to accurately represent energy groups lower than the first group containing the source (i.e. groups made up of scattered particles). From these results it was determined that further analysis with the SIMIND data shall be performed for the first energy group only. Since a SPECT study desires images without scatter, the first energy group is the one of primary interest and this limitation is acceptable for this study.

The results of the SIMIND and TITAN codes were compared using the infinity-norm and 2-norm of the normalized projection images at four difference angles around the phantom. The most accurate SIMIND result (131 million particles per projection) and the TITAN code results for S_6 -DTW and S_6 -DDZ were chosen for comparison. The choice of S_6 for order of quadrature in the TITAN code had an infinity-norm of about 1%, as found in Table 4-1. Figure 4-5 displays the normalized projection images of the heart produced by each code for a visual comparison and the images are in good agreement. The numerical comparison can be found in Table 4-3 and shows that comparisons with S_6 -DDZ and S_6 -DTW were similar. The anterior projection images show the smallest infinity-norm of 15% and the right lateral images have the largest infinity-norm of 49.2%. The heart is closest to the anterior surface of the body and so this projection is expected

to be more accurate than other angles in SIMIND. Comparing the DDZ and DTW results reveals that neither method is better than the other for this model. A likely contributor to the significant differences between the TITAN and SIMIND images is the fact that SIMIND uses attenuation values provided in the NCAT phantom, while cross sections were generated separately for TITAN. Further, the large difference in the right lateral image could be attributed to a higher expected uncertainty for the SIMIND calculations at that particular angle.

The computation times for each code to create four projection images are compared in Table 4-4 for SIMIND with 131 million and 1 million photons per projection and for TITAN S_6 using DDZ and DTW. All calculations were performed in serial using the Einstein PC-cluster. A speedup factor is defined relative to the SIMIND case with 131 million photons per projection. Table 4-4 clearly shows that the TITAN code was able to complete the calculations in significantly less time than SIMIND; however, the time is highly dependent on the number of photons per projection simulated in SIMIND. Even for the relatively fast 1 million photons per projection SIMIND case, the TITAN code took approximately a third less time. The speedup of the TITAN code over the SIMIND code will further improve if the number of projection angles simulated is increased.³ This is because TITAN has the advantage of finding a solution for the angular flux distribution in the phantom once and then creating however many projection images are required. In contrast, the SIMIND code must repeat the simulation for each projection image. The DDZ scheme was 30 s faster than the DTW method, which could become a reason to use DDZ for larger problems since their accuracy appears to be equal for this case. However, this is for comparing the methods

using the same mesh size and, in principle, the DTW scheme yields accurate solutions for larger mesh sizes, for which the DDZ scheme is not accurate, and therefore requires a larger number of meshes and more computation time. For the present study, very small mesh sizes were needed to properly represent the heart wall in the phantom. For these mesh sizes the DDZ scheme is as accurate as DTW, and somewhat more efficient.

Investigation into SIMIND and TITAN Differences

The differences seen between the TITAN and SIMIND code results in the previous section were significant. Further analysis was completed to better understand these differences. First relative differences between normalized projection image pixels greater than 0.65 were considered in greater detail. The fraction of these pixels with absolute relative differences greater than 10% is given in Table 4-5. The second largest relative difference is also found in Table 4-5. The results show drops of 10% and 20.2% to the next largest relative difference in the right lateral projection images for the S_6 -DDZ and S_6 -DTW cases, respectively. This indicates that the differences between the two images are overstated by the infinity-norm. The majority of pixels have a relative difference of less than 10% for the anterior, left lateral, and right lateral projection images. The posterior projection images have differences greater than 10% for most pixels.

Single profiles of pixels, either columns or rows, were plotted from the SIMIND and TITAN S_6 -DDZ projection images. These plots display the behavior of the flux through the projection images. Profiles were chosen that passed through the heart and are plotted in Figures 4-6 and 4-7 along with the relative difference for each pixel. Figure 4-6 plots a column and a row through the heart for the anterior projection angle. It is clear

that the SIMIND and TITAN plots have the same shape; however there appears to be a shift between the two data sets in the column plot (Figure 4-6A). Attempts to discover a cause for this shift have not yet been successful, but it should be noted that the edges of the NCAT heart phantom line up exactly with the position of the edges in the TITAN anterior projection image. This would indicate that the SIMIND data is shifted. Further studies using the MCNP5 code are planned and should help determine if there is actually a shift.

Profile plots for a row through the heart of the left lateral, posterior, and right lateral projection images are found in Figure 4-7. Figure 4-7A shows good general agreement with a difference in plot shape seen around pixel 32. While the overall behavior is similar in the posterior image, Figure 4-7B, there appear to be differences in magnitude. The right lateral image, Figure 4-7C, again shows that the behavior of the two plots matches very well; however a shift between the data points is visible again.

Adjoint Analysis and MCNP5 Comparison

The single-hole collimator with a $0.625 \times 0.625 \times 0.625 \text{ cm}^3$ voxel detector located 10 cm from the phantom surface and centered over the heart had a response of $0.07493 \pm 2.13\%$ in MCNP5. Relative differences for the TITAN response calculations were then found relative to this value. MCNP5 stated uncertainties are all one standard deviation. The single-hole MCNP5 collimator had an acceptance angle of 3.14° and several orders of quadrature were used to calculate the response in TITAN to determine the best match to this acceptance angle. In each case the direction closest to the +y-axis is the only direction from the quadrature set within the acceptance angle, and the adjoint source is defined along that direction in each of the four octants along the +y-

axis. As was seen previously, a quadrature order of 6 is perfectly adequate for this simulation; however, in the following simulations much higher orders of quadrature will be used for the sole purpose of obtaining a particular direction in the quadrature set. The lowest quadrature order used here, S_{42} , has the source direction approximately equal to the acceptance angle, while the higher quadrature orders will have direction cosines smaller than the acceptance angle cosine.

The group 1 adjoint flux obtained from the TITAN model of a single-hole collimator using S_{62} is shown in Figure 4-8 for an axial slice and a sagittal slice through the adjoint source location. The adjoint flux is shown on an exponential scale due to its significant drop-off away from the adjoint source. At the far left of Figure 4-8 the peak adjoint flux value can be seen in the voxel containing the adjoint source. To the right of the figures and far from the adjoint source ray effects² (i.e. the effects of not having a quadrature direction actually along the y-axis) are visible. However this is seen in the air behind the torso phantom where there is no source and should therefore not contribute to the calculated response. Also, the majority of the detector response will come from the intense source in the heart, which is in the front of the torso where this effect is not visible.

Table 4-6 gives the responses for several different quadrature orders of the angular adjoint source and the percent difference relative to the MCNP5 solution. The direction weight and the area represented at the collimator entrance by this weight are also tabulated. The results in Table 4-6 show that the best choice is S_{62} and this order of quadrature places the adjoint source direction well within the acceptance angle. The solid angle range that this direction represents ($\Delta\Omega$) then covers the distribution of

acceptable angles. The solid angle range represented is not a direct match to the circular acceptance angle of the MCNP5 collimator; however, it seems that the TITAN methodology makes a good approximation. The relative difference in the response for S_{62} is -0.23% and this demonstrates that the adjoint methodology of simulating a collimator in TITAN can produce accurate results when the appropriate adjoint source direction can be obtained. In Figure 4-9 the relative difference in each of the four orders of quadrature is plotted versus the adjoint source direction weight, which is proportional to the area subtended by the adjoint source direction. A linear regression was used to fit a line to these points and is shown in Figure 4-9 with a correlation coefficient of 0.99998. This linear relationship means that the appropriate quadrature order to use could be determined from two test points.

In the single-hole case above, the adjoint source was represented using only four directions in TITAN. To investigate this as a potential source of error, the ordinate splitting technique was used to split the source direction in each octant into several directions whose weights summed to weight of the original direction. These cases are presented in Table 4-7 along with their non-split counterparts for comparison. The S_{60} , S_{62} , and S_{68} cases used P_N - T_N splitting of various orders. It is clear from the table that while there is a measurable change in the response with ordinate splitting, the difference is slight and one direction per octant is adequate. The remaining collimator cases will use just one direction per octant.

The four- and nine-hole cases were considered next with one adjoint source direction per octant. Tables 4-8 and 4-9 give the response and relative differences for various orders of quadrature compared to the MCNP5 response for the four- and nine-

hole cases, respectively. The most accurate quadrature is S_{134} for the four-hole collimator with a relative difference of 0.94% and S_{188} for the nine-hole collimator with a relative error of 0.12%. As with the single-hole case, it is clear that the adjoint methodology in TITAN can give accurate calculations of response with collimation. The area corresponding to the direction weight that is closest to the MCNP5 response is also consistently larger than the collimator hole area in MCNP5. The relative differences as a function of direction weight for the four- and nine-hole cases are plotted in Figure 4-10 and Figure 4-11, respectively. These plots also have lines from linear regression fits that show, as seen in the single-hole case, that the relationship between the relative difference in response and the direction weight is linear. This relationship makes sense when one considers that the direction weight is directly proportional to the solid angle represented by the direction.

Given the above described linear regression fits for the single, four, and nine-hole collimator cases, an attempt was made to create a model to predict the best order of quadrature for a given acceptance angle in MCNP5. The fitted lines for the collimator cases were used to find direction weights corresponding to relative differences of zero. These “ideal direction weights” were then plotted versus the cosines of the collimator acceptance angles. There appeared to be a linear relationship between the weight/area and the direction cosine, therefore, a line was fitted to this data using a linear regression and each of the acceptance angle cosines for the collimators was then substituted into the fitted equation to get a predicted direction weight. The predicted direction weights were then compared with the direction weights of different quadrature orders to obtain the closest match.

Initially, the best choice of quadrature order was correctly predicted for the single-hole case; however, wrong quadrature orders were obtained for the other two collimator cases. The predictions were close to the best answer so it was hypothesized that if the MCNP5 data had too large of a variance the fitted lines would not be accurate enough to predict the best direction weights. To this effect, the four-hole collimator MCNP5 response had a standard deviation of $\sim 4.3\%$ and was noticeably larger than the other two cases, which had standard deviations of $\sim 3\%$. Hence, for the four-hole collimator case, the number of histories was increased until a standard deviation of $\sim 3\%$ was achieved. Figure 4-12 gives the plot and the fitted line from a linear regression performed using the new ideal direction weight for the four-hole collimator. The data showed linear behavior with a correlation coefficient of 0.999998, resulting in accurate prediction of the quadrature orders that had given the smallest relative differences. These findings are summarized in Table 4-10. More analysis is needed with additional acceptance angles to verify that this method of predicting the best order of quadrature can work consistently, however these results are promising.

The time for calculation of the single, four, and nine-hole collimator cases for each code are tabulated in Table 4-11. All calculations were performed in parallel on the Dirac PC-cluster using 16 processors. For the MCNP5 code, these results represent the time taken to achieve the uncertainty given with the response values in previous tables. For the TITAN code, it is clear that time becomes an important consideration as the acceptance angles are reduced and higher orders of quadrature are required. For these cases, TITAN shows a significant speedup over the MCNP5 code.

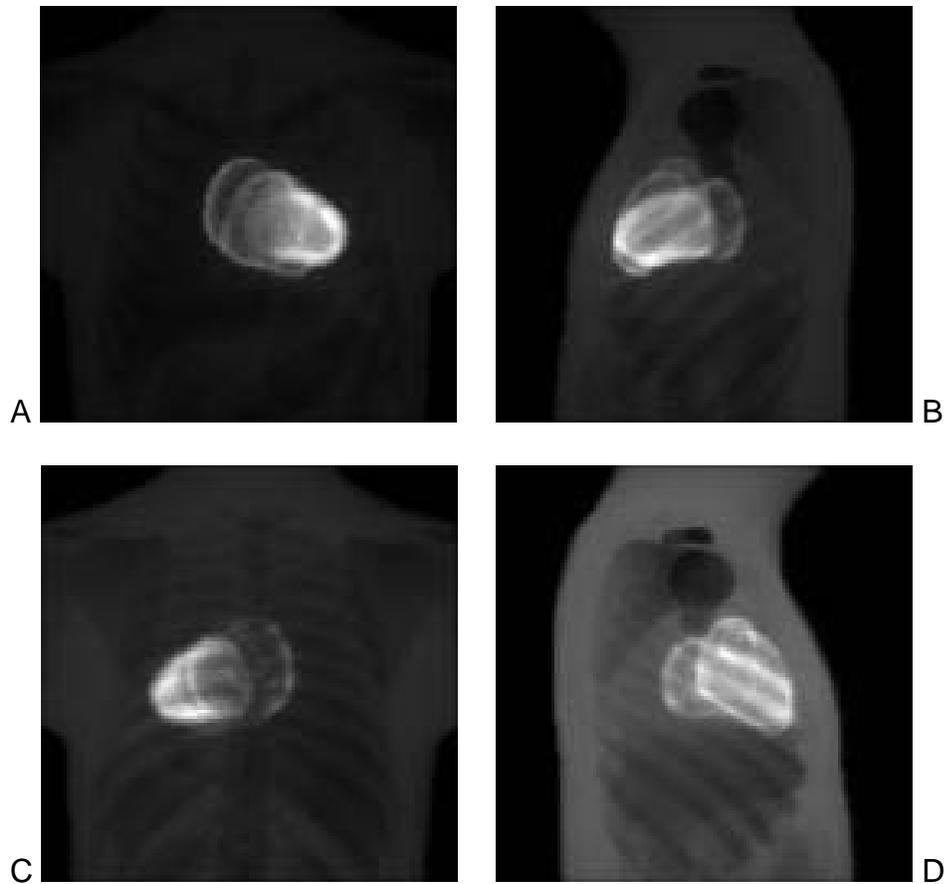


Figure 4-1. Normalized TITAN projection images at four angles around the patient A) Anterior B) Left Lateral C) Posterior D) Right Lateral

Table 4-1. Comparison between myocardium pixels in group 1 anterior projection images with increasing order of quadrature for different number of voxels (meshes)

Differencing scheme	Quadrature order	64 x 64 x 64		128 x 128 x 128	
		Infinity-norm	2-norm	Infinity-norm	2-norm
DDZ	S ₄	-2.36%	0.0536	-2.24%	0.1410
	S ₆	1.12%	0.0272	-1.15%	0.0621
	S ₈	-0.48%	0.015	-0.57%	0.0389
	S ₁₀	-0.34%	0.0104	-0.38%	0.0322
	S ₁₂	-	-	-	-
DTW	S ₄	-1.58%	0.0404	-1.71%	0.1038
	S ₆	-0.82%	0.0209	-0.87%	0.0439
	S ₈	-0.39%	0.0121	-0.38%	0.0268
	S ₁₀	-0.20%	0.0063	-0.21%	0.0156
	S ₁₂	-	-	-	-

Table 4-2. Maximum absolute relative difference between TITAN flux distributions in the myocardium of the 64 x 64 x 64 model relative to the 128 x 128 x 128 model

	S ₄	S ₆	S ₈	S ₁₀	S ₁₂
DDZ	0.127%	0.168%	0.159%	0.188%	0.187%
DTW	0.154%	0.191%	0.168%	0.173%	0.175%

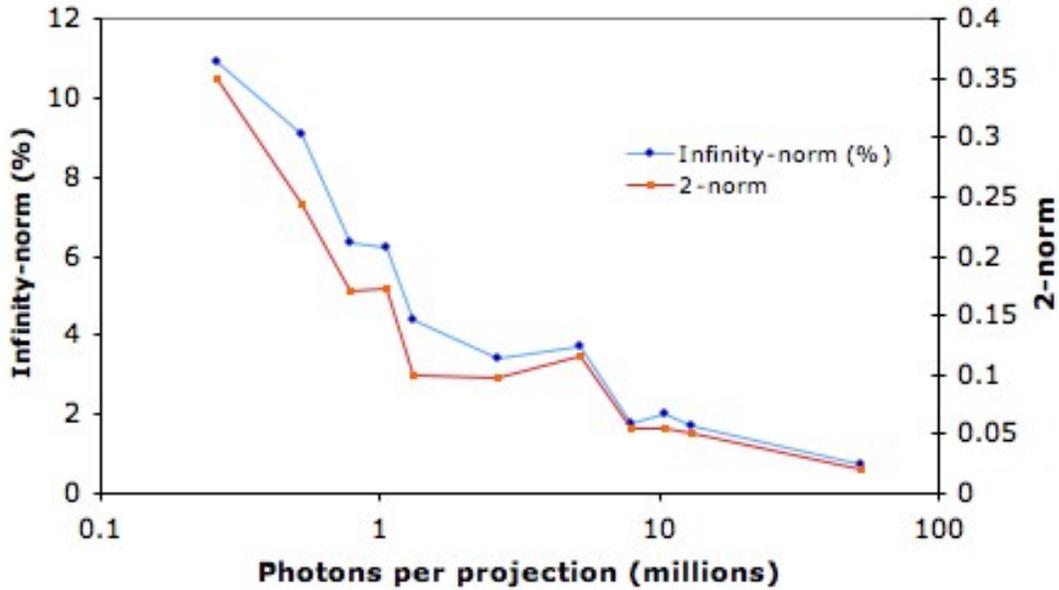


Figure 4-2. Infinity-norm and 2-norm relative to 131 million photons per projection in SIMIND

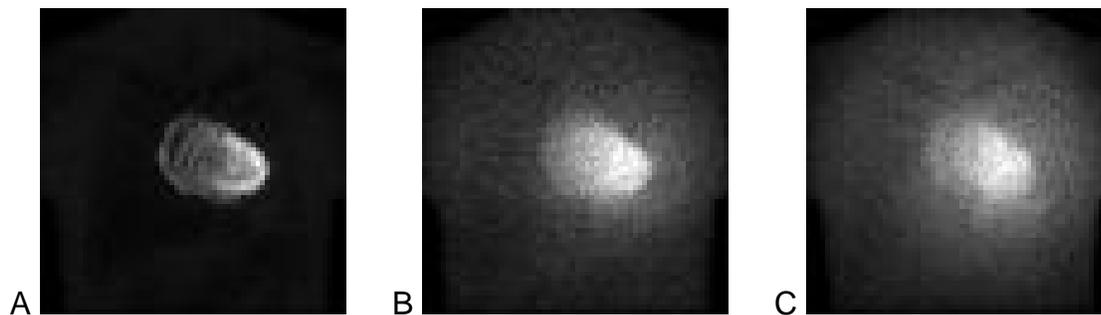


Figure 4-3. SIMIND three energy group anterior projection images A) Group 1 B) Group 2 C) Group 3

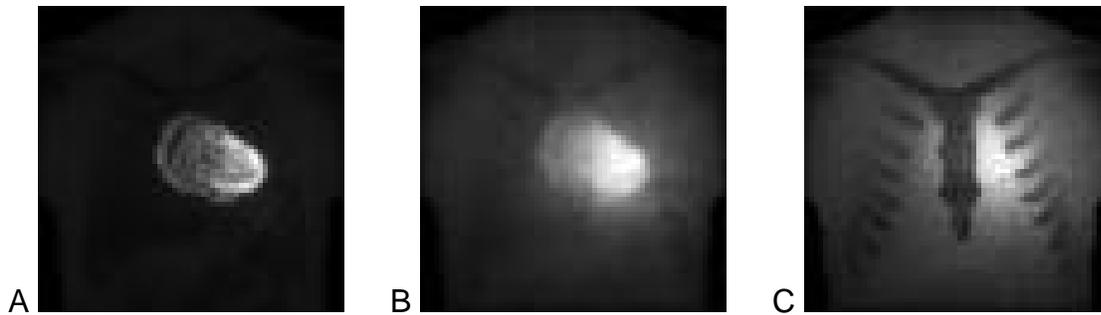


Figure 4-4. TITAN three energy group anterior projection images A) Group 1 B) Group 2 C) Group 3

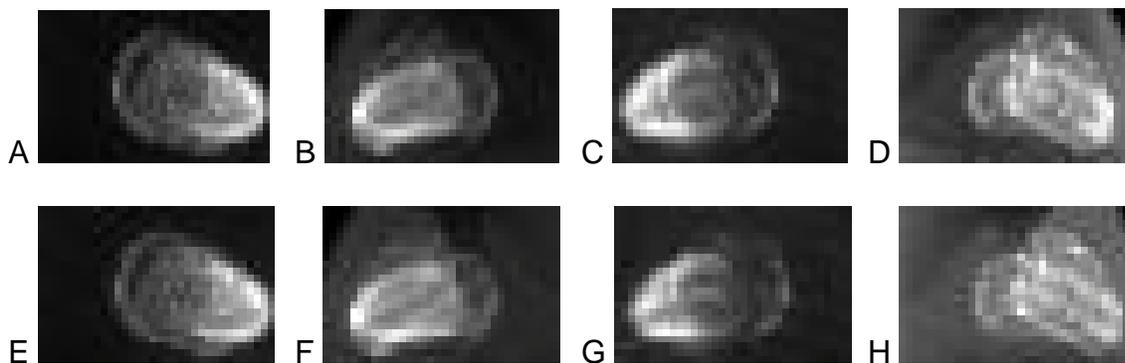


Figure 4-5. SIMIND and TITAN normalized projection images of the heart. A) SIMIND anterior. B) SIMIND left lateral. C) SIMIND posterior. D) SIMIND right lateral. E) TITAN anterior. F) TITAN left lateral. G) TITAN posterior. H) TITAN right lateral.

Table 4-3. Infinity-norm and 2-norm for TITAN S_6 -DDZ and S_6 -DTW compared to SIMIND for 131 million particles per projection

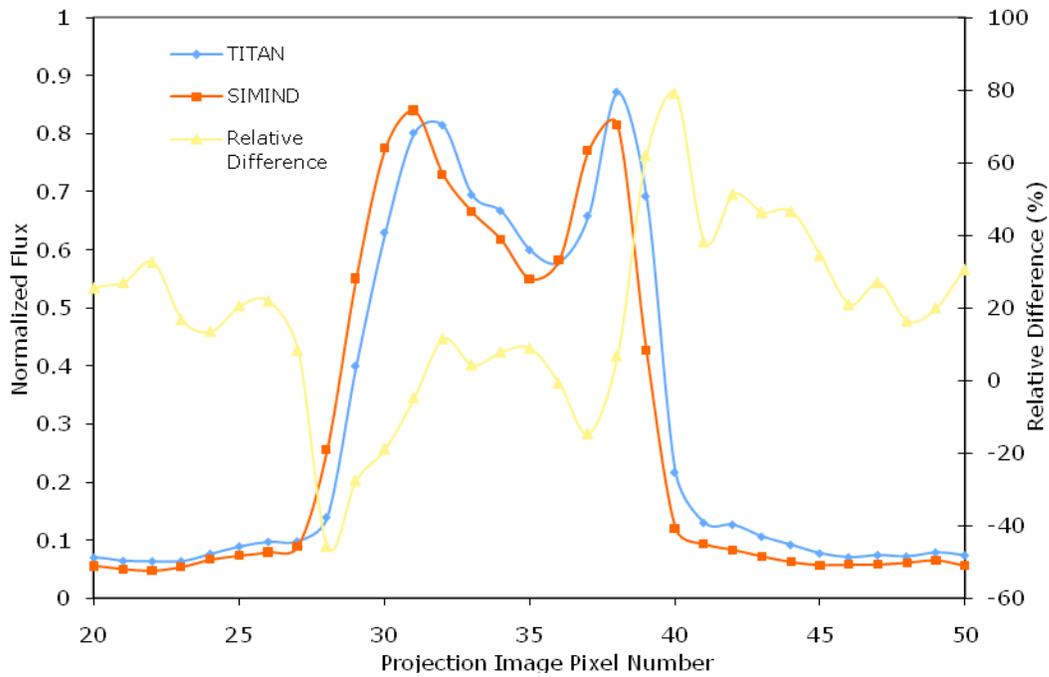
Projection angle		Anterior	Left lateral	Posterior	Right lateral
DDZ	Infinity-norm	15.0%	29.0%	-31.8%	42.7%
	2-norm	0.445	0.674	0.893	1.06
DTW	Infinity-norm	19.9%	30.1%	-27.6%	49.2%
	2-norm	0.545	0.679	0.868	1.08

Table 4-4. Three group computation time comparison for four projection angles

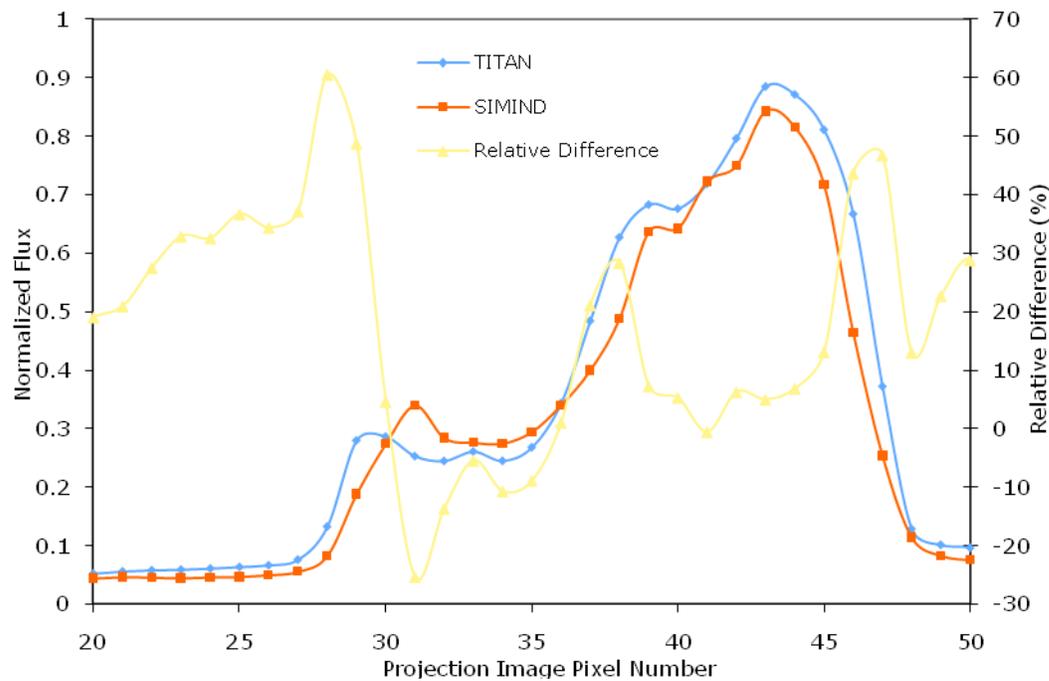
Code	Time (minutes)	Speedup factor
SIMIND (131 million particles)	377.0	1.0
SIMIND (1 million particles)	11.9	31.8
TITAN (S_6 -DDZ)	7.8	48.3
TITAN (S_6 -DTW)	8.3	45.4

Table 4-5. Comparison of relative differences of TITAN S₆-DDZ and S₆-DTW compared to SIMIND for 131 million particles per projection including the infinity-norm, the next largest difference after the infinity-norm, and the percentage of pixels with absolute relative differences greater than 10%

Projection angle		Anterior	Left lateral	Posterior	Right lateral
DDZ	Infinity-norm	15.0%	29.0%	-31.8%	42.7%
	Next largest difference	14.6%	28.7%	27.7%	32.7%
	Fraction > 10%	24.2%	38.2%	59.5%	36.3%
DTW	Infinity-norm	19.9%	30.1%	-27.6%	49.2%
	Next largest difference	17.1%	27.4%	26.3%	29.0%
	Fraction > 10%	27.3%	34.3%	60.6%	45.1%

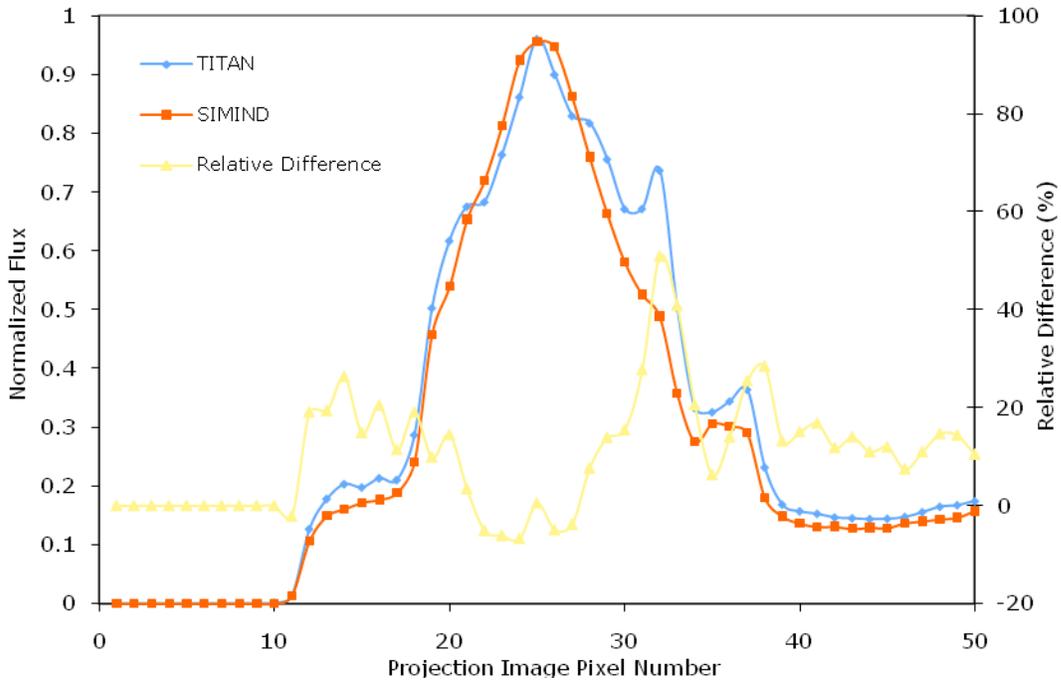


A

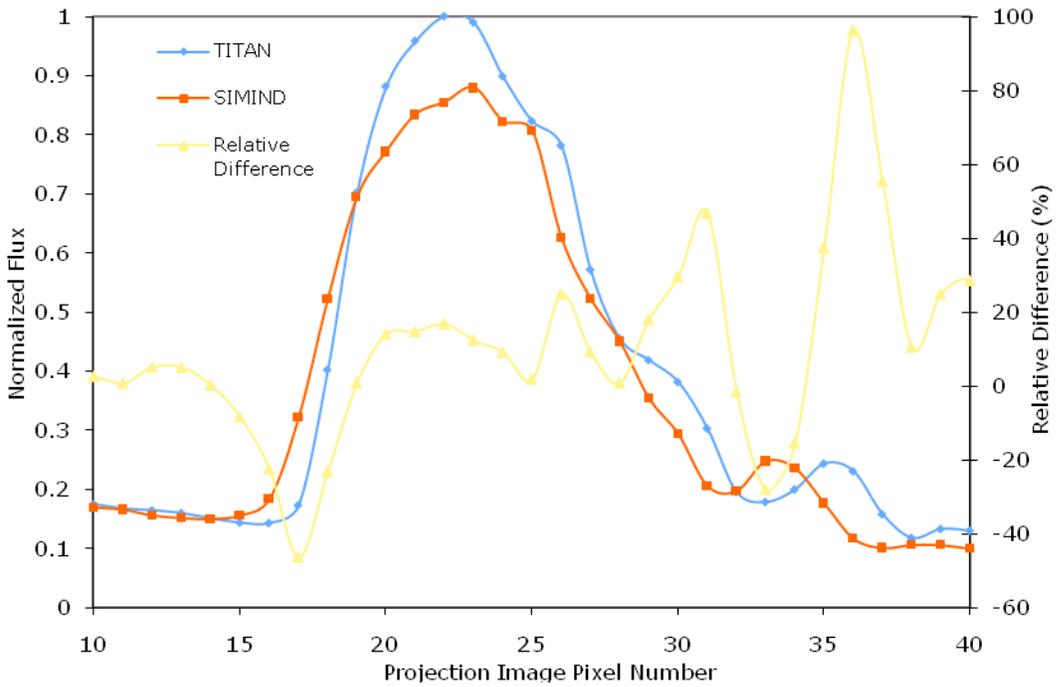


B

Figure 4-6. Plot of normalized flux along a single profile of anterior projection images created by the SIMIND and TITAN code along with the relative difference through A) column 44 and B) row 38

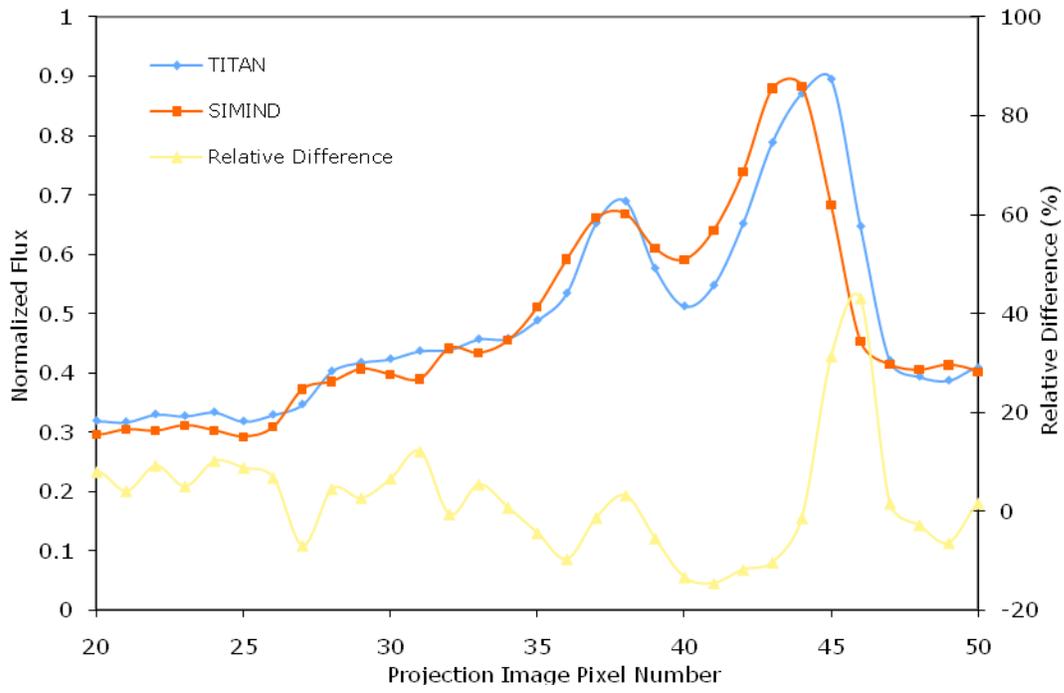


A



B

Figure 4-7. Plot of normalized flux along a single profile through row 38 of SIMIND and TITAN projection images and the relative difference for A) left lateral, B) posterior, and C) right lateral angles



C

Figure 4-7. Continued

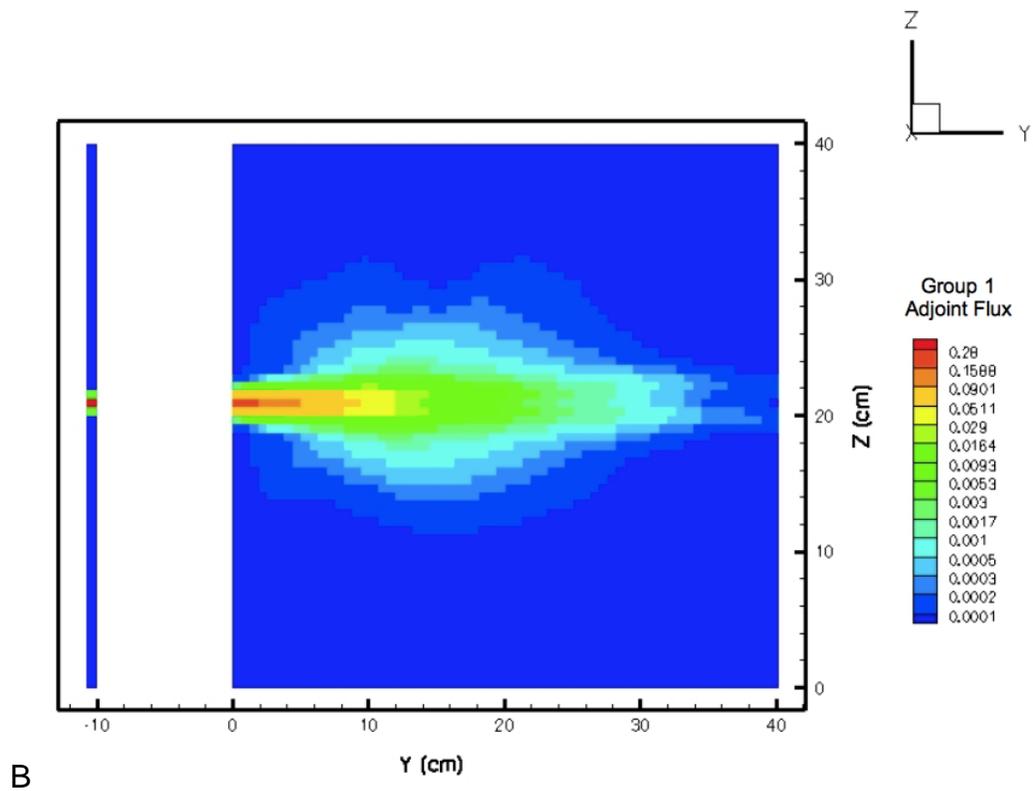
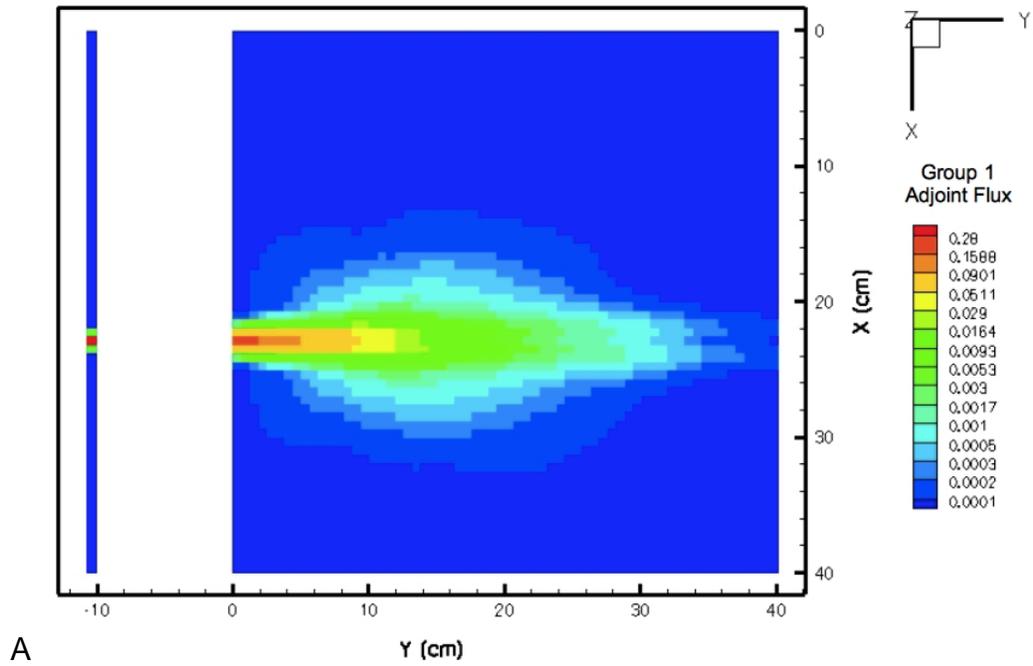


Figure 4-8. Group 1 adjoint flux through the heart from the TITAN model of a single-hole collimator for A) an axial slice and B) a sagittal slice

Table 4-6. Single-hole collimator responses for TITAN compared with MCNP5

	Response (counts/s)	Relative difference	Area (cm ²)	Direction weight
MCNP5	0.0749 ±2.13%	-	0.3068	-
S ₄₂	0.1676	123.71%	0.7180	3.517E-3
S ₆₀	0.0800	6.75%	0.3532	1.730E-3
S ₆₂	0.0748	-0.23%	0.3313	1.621E-3
S ₆₈	0.0618	-17.48%	0.2753	1.349E-3

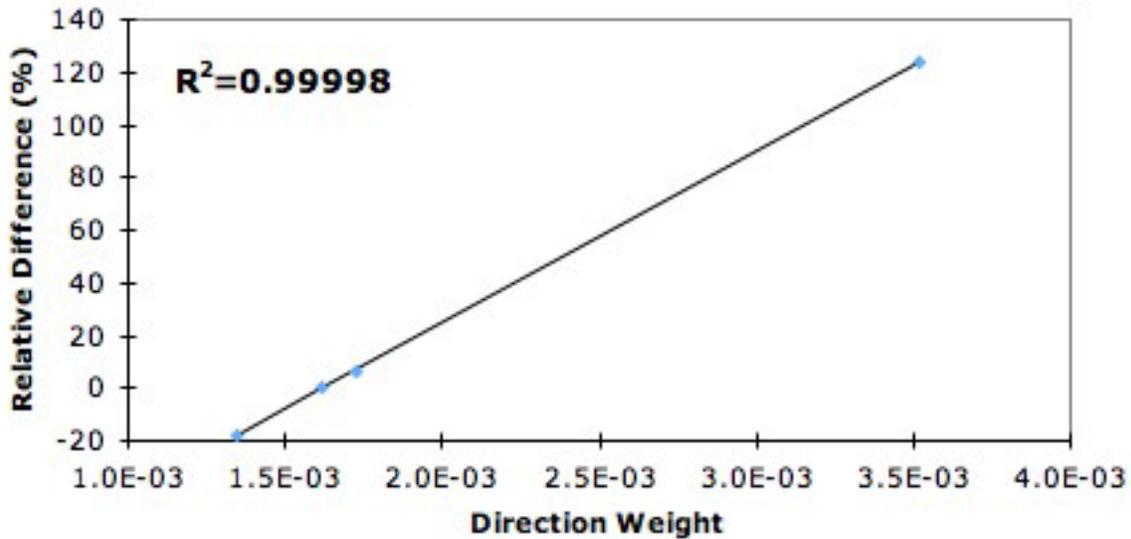


Figure 4-9. Single-hole collimator relative difference in TITAN response compared to MCNP5 as a function of direction weight (order of quadrature)

Table 4-7. Single-hole collimator responses for TITAN with P_N-T_N ordinate splitting compared with MCNP5

Code	Source directions per octant	Response (counts/s)	Relative difference
MCNP5	-	0.0763 ±2.13%	-
TITAN S ₆₀	No splitting	1	0.0800
	P _N -T _N Order 3	6	0.0800
	P _N -T _N Order 6	21	0.0801
TITAN S ₆₂	No splitting	1	0.0748
	P _N -T _N Order 3	6	0.0748
	P _N -T _N Order 5	15	0.0749
TITAN S ₆₈	No splitting	1	0.0618
	P _N -T _N Order 6	21	0.0620

Table 4-8. Four-hole collimator responses for TITAN compared with MCNP5

	Response (counts/s)	Relative difference	Area (cm ²)	Direction weight
MCNP5	0.0125 ±2.94%	-	0.0631	-
S ₉₀	0.0287	123.83%	0.1575	7.713E-4
S ₁₃₀	0.0138	7.20%	0.0756	3.703E-4
S ₁₃₄	0.0130	0.94%	0.0712	3.486E-4
S ₁₃₆	0.0126	-2.99%	0.0691	3.384E-4
S ₁₃₈	0.0122	-4.78%	0.0671	3.287E-4

Table 4-9. Nine-hole collimator responses for TITAN compared with MCNP5

	Response (counts/s)	Relative difference	Area (cm ²)	Direction weight
MCNP5	0.00704±3.58%	-	0.0297	-
S ₁₃₀	0.01459	107.38%	0.0756	3.703E-4
S ₁₈₀	0.00767	9.05%	0.0395	1.730E-4
S ₁₈₈	0.00704	0.12%	0.0362	1.621E-4
S ₁₉₀	0.00690	-1.94%	0.0354	1.349E-4

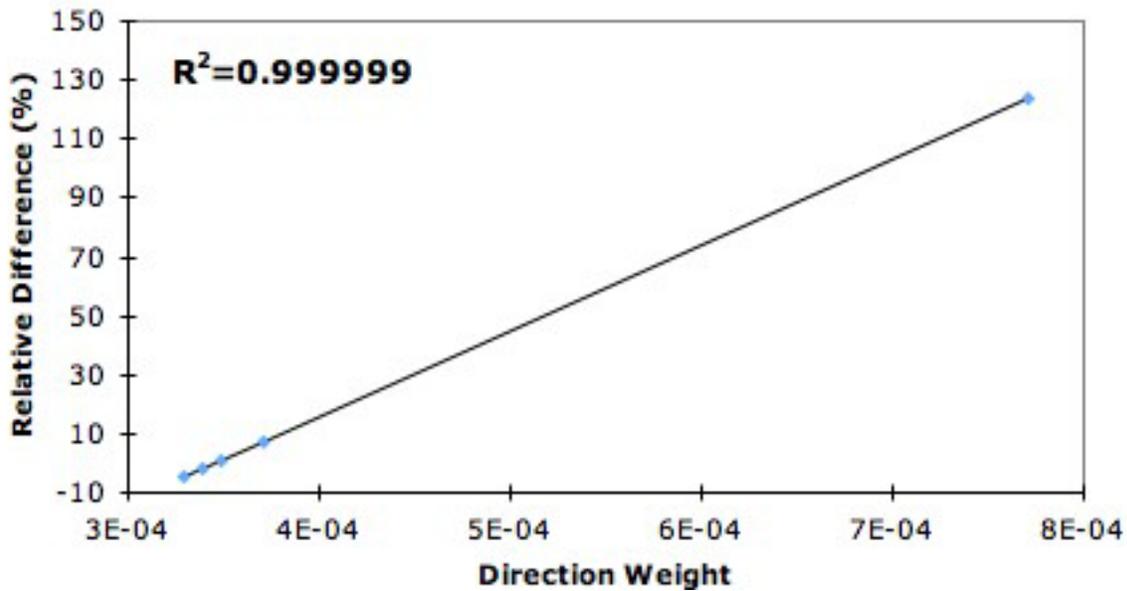


Figure 4-10. Four-hole collimator relative error in TITAN response as a function of direction weight (order of quadrature)

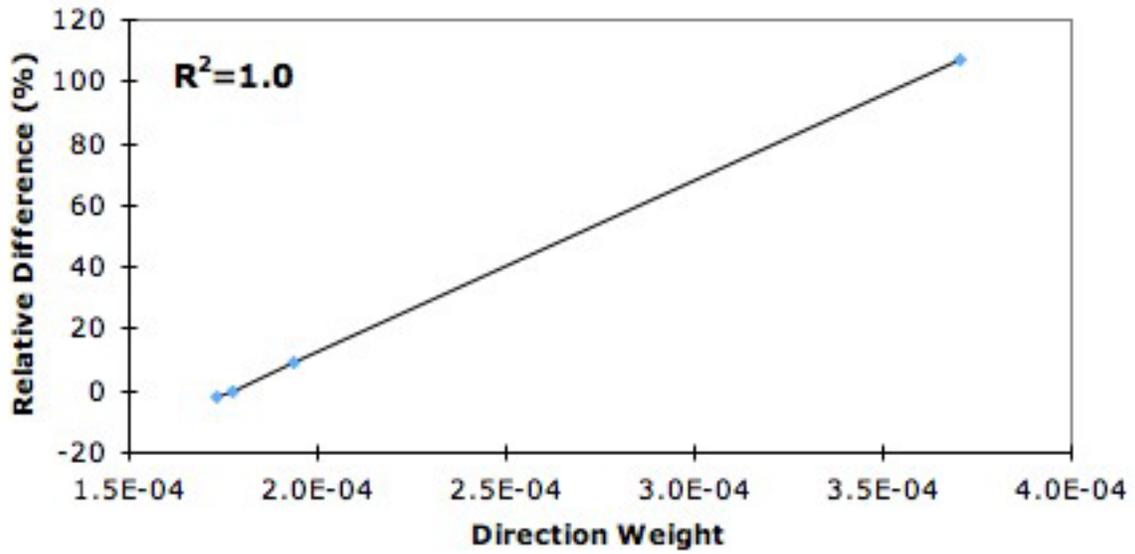


Figure 4-11. Nine-hole collimator relative error in TITAN response as a function of direction weight (order of quadrature)

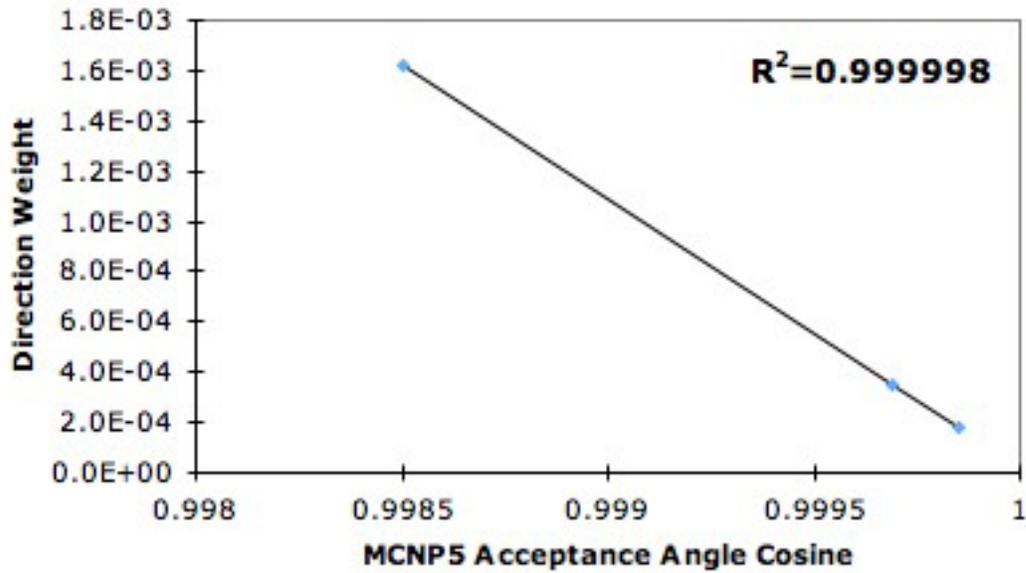


Figure 4-12. Optimal TITAN direction weight versus the cosine of the collimator acceptance angle in MCNP5

Table 4-10. Ideal and fitted direction weights for each collimator case and the resulting recommended order of quadrature

Collimator	Acceptance angle cosine	Ideal direction weight	Fitted direction weight	Recommended order of quadrature
Single-hole	0.99850	1.6222E-3	1.6221E-3	S ₆₂
Four-hole	0.99969	3.4534E-4	3.4646E-4	S ₁₃₄
Nine-hole	0.99985	1.7709E-4	1.7610E-4	S ₁₈₈

Table 4-11. Calculation times for 16 processors in MCNP5 and TITAN for each collimator

Collimator	Code	Time (min)	Speedup factor
Single-hole	MCNP5	366.5	-
	TITAN	39.3	9.3
Four-hole	MCNP5	1066.8	-
	TITAN	180.7	5.9
Nine-hole	MCNP5	1348.8	-
	TITAN	390.5	3.5

CHAPTER 5 CONCLUSIONS AND FUTURE WORK

Several aspects of single photon emission computed tomography (SPECT) simulation using the TITAN code were investigated. For this simulated myocardial perfusion study, low orders of quadrature were found to be adequate with increasing the order resulting in only about 2% improvement or less in projection images. The results using the diamond differencing with zero fix-up (DDZ) and directional theta-weighted (DTW) differencing schemes showed no significant advantages for either method. This is attributed to the fact that mesh sizes are small enough for DDZ to be accurate. In comparing with SIMIND, the projection images created using the TITAN code were found to be in good visual agreement for energy group 1. However, significant differences ranging from 15% to 49% were found when the projection images were compared numerically. These differences are believed to be primarily the result of TITAN using cross sections generated using the CEPXS code, whereas the SIMIND code does not allow the input of any cross sections. Comparison of the lower energy group projection images revealed discrepancies in attenuation in bone that are also thought to be the result of SIMIND not having cross-section information. It was determined that the SIMIND methodology could not accurately represent lower energy groups that do not contain the source.

A SPECT model was created in MCNP5 to provide a Monte Carlo comparison to the adjoint methodology of the TITAN code. The adjoint methodology used in TITAN to simulate collimation was found to provide highly accurate results for detector response with three different collimator angles obtaining responses within 1% of the MCNP5 result. A linear relationship was seen between the direction weight used to define the

adjoint source in TITAN and the relative difference of the response with the MCNP5 result. A line obtained via linear regression was able to predict the best order of quadrature for the TITAN code for simulating a given MCNP5 collimator acceptance angle. The TITAN code also proved to be at least three times as fast as the MCNP5 calculation for each acceptance angle simulated. However, TITAN calculation times did increase significantly as extremely high orders of quadrature were needed to represent smaller acceptance angles.

With future work, it is hoped that the TITAN code can be used to develop algorithms resulting in improved image quality. By studying and applying the results of SPECT simulations in the TITAN code, the inevitable scatter that reduces image quality in patient SPECT images can be characterized and predicted. If a strong understanding of the scatter in an image can be obtained, this source of image degradation can be reduced. With improved image quality, disease diagnosis can be more accurate, treatment can be more effective, and doses delivered to patients can be reduced without sacrificing image quality.

APPENDIX A INPUT FILES

The general input files used in this research are found below. The parameters adjusted for different cases are described. To conserve space most files had to be abbreviated however any important parameters should still be present.

- The NCAT input file consists mostly of default parameters. The parameters of note are the pixel and slice width, array size, Start_slice, and end_slice. These are given in Figure A-1 for the 64 x 64 x 64 voxel phantom and were adjusted to create the 128 x 128 x 128 voxel phantom.

```

1      : activity_phantom_each_frame (1=save phantom to file, 0=don't save)
1      : attenuation_coeff_phantom_each_frame (1=save phantom to file, 0=don't save)
0      : activity_phantom_average (1=save , 0=don't save) see NOTE 0
0      : attenuation_coeff_phantom_average (1=save, 0=don't save) see NOTE 0

0      : motion_option (0=beating heart only, 1=respiratory motion only, 2=both motions) see NOTE 1

1      : output_period (SECS) (if <= 0, then output_period=time_per_frame*output_frames)
1      : time_per_frame (SECS) (**IGNORED unless output_period<=0**)
8      : output_frames (# of output time frames )

1      : hrt_period (SECS) (length of beating heart cycle; normal = 1s) see NOTE 2
0.0    : hrt_start_phase_index (range=0 to 1; ED=0, ES=0.4) see NOTE 2
heart_base : basename for heart files
5      : resp_period (SECS) (length of respiratory cycle; normal breathing = 5s) see NOTE 2
0.0    : resp_start_phase_index (range=0 to 1, full exhale=0, full inhale=0.455) see NOTE 2
...
...
...
0.6250 : pixel width (cm); see NOTE 5
0.6250 : slice width (cm); see NOTE 5
64     : array size see NOTE 6
1      : subvoxel_index (=1,2,3,4 -> 1,8,27,64 subvoxels/voxel, respectively)
64     : Start_slice; see NOTE 7
127    : end_slice; see NOTE 7

-90    : zy_rotation (beta) in deg. (-90); see NOTE 8
-20.   : xz_rotation ( phi) in deg. (-20); see NOTE 8
-50.   : yx_rotation ( psi) in deg. (-50); see NOTE 8
0.0    : x translation in mm ; see NOTE 8
0.0    : y translation in mm ; see NOTE 8
20.0   : z translation in mm ; see NOTE 8
1      : apical_thinning (1 = present, otherwise not present) /*parameter is ignored*/
0.0    : valve thickness in cm (0= no valve); cannot be a negative value /*parameter is ignored*/
0.3    : av_step(cm): step width for smooth change between Atr & Ven (0=none) /*parameter is ignored*/
0      : total_rotation (deg); /*parameter is ignored*/

0      : activity units (1= scale by voxel volume; 0= don't scale) NOTE 9

SEE NOTE 9 FOR INFORMATION ON SETTING ORGAN ACTIVITIES - activities can be fixed or
determined by user defined time-activity curves
-----
0      : myoLV_time_curve (1= activity determined by a time-activity curve; 0= fixed activity)
75     : hrt_myoLV_act - fixed activity in left ventricle myocardium if above option is 0
sample_act.txt : myoLV_act_filename - name of file containing time-activity curve for LV myocardium
-----

```

Figure A-1. NCAAT input file

```

0          : myoRV_time_curve (1= activity determined by a time-activity curve; 0= fixed activity)
75         : hrt_myoRV_act - activity in right ventricle myocardium
sample_act.txt : myoRV_act_filename - name of file containing time-activity curve for RV myocardium
-----
0          : myoLA_time_curve (1= activity determined by a time-activity curve; 0= fixed activity)
75         : hrt_myoLA_act - activity in left atrium myocardium
sample_act.txt : myoLA_act_filename - name of file containing time-activity curve for LA myocardium
-----
0          : myoRA_time_curve (1= activity determined by a time-activity curve; 0= fixed activity)
75         : hrt_myoRA_act - activity in right atrium myocardium
sample_act.txt : myoRA_act_filename - name of file containing time-activity curve for RA myocardium
-----
0          : bldpLV_time_curve (1= activity determined by a time-activity curve; 0= fixed activity)
2          : hrt_bldpLV_act - activity in left ventricle chamber (blood pool)
sample_act.txt : bldpLV_act_filename - name of file containing time-activity curve for LV blood pool
-----
0          : bldpRV_time_curve (1= activity determined by a time-activity curve; 0= fixed activity)
2          : hrt_bldpRV_act - activity in right ventricle chamber (blood pool)
sample_act.txt : bldpRV_act_filename - name of file containing time-activity curve for RV blood pool
-----
0          : bldpLA_time_curve (1= activity determined by a time-activity curve; 0= fixed activity)
2          : hrt_bldpLA_act - activity in left atria chamber (blood pool)
sample_act.txt : bldpLA_act_filename - name of file containing time-activity curve for LA blood pool
-----
0          : bldpRA_time_curve (1= activity determined by a time-activity curve; 0= fixed activity)
2          : hrt_bldpRA_act - activity in right atria chamber (blood pool)
sample_act.txt : bldpRA_act_filename - name of file containing time-activity curve for RA blood pool
-----
0          : body_time_curve (1= activity determined by a time-activity curve; 0= fixed activity)
2          : body_activity (background activity) ;
sample_act.txt : body_act_filename - name of file containing time-activity curve for body
-----
0          : liver_time_curve (1= activity determined by a time-activity curve; 0= fixed activity)
0.0        : liver_activity;
sample_act.txt : liver_act_filename - name of file containing time-activity curve for liver
-----
...
...
...

```

Figure A-1. Continued

- The CEPXS input file to generate the multigroup cross sections is found in Figure A-2.

```

title
Heart Phantom Cross sections
first-order
energy 0.15455
cutoff 0.010
legendre 3
photon-source
no-coupling
photons
user 3
0.12645
0.09835
0.010000
electrons
linear 0
*air
material o .24 n .76
state gas
density 0.0012
*body (water)
material h 0.111 o 0.889
density 1.00
*intestine
material h 0.106 c 0.115 n 0.022 o 0.751 na
0.001 -
p 0.001 s 0.001 cl 0.002 k 0.001
density 1.03
*lung
material h 0.103 c 0.105 n 0.031 o 0.749 -
na 0.002 p 0.002 s 0.003 cl 0.003 k 0.002
density 0.3
*rib bone
material h 0.024 o 0.196 ca 0.78
density 1.92
*liver
material h .102 c .139 n .030 o .716 na .002 -
p .003 s .003 cl .002 k .003
density 1.06
*kidney
material h .103 c .132 n .03 o .724 na .002 -
p .002 s .002 cl 0.002 k .002 ca .001
density 1.05
*spine bone
material h 0.024 o 0.196 ca 0.78
density 1.42
*muscle
material h .102 c .143 n .034 o .710 -
na .001 p .002 s .003 cl 0.001 k .004
density 1.05
*cartilage
material h 0.024 o 0.196 ca 0.78
density 1.10
*spleen
material h .103 c .113 n .032 o .741 na .001 -
p .003 s .002 cl .002 k .003
density 1.06
*heart
material h .105 c .104 n .025 o .757 na .001 -
p .002 s .001 cl .002 k .003
density 1.05
*blood
material h .102 c .110 n .033 o .745 -
na .001 p .001 s .002 cl .003 k .002 fe .001
density 1.06
print
Legendre 3

```

Figure A-2. CEPXS input file

- The TITAN input file for SPECT simulation includes a section, Section 10, where the SPECT specific parameters are specified. Some parameters that were varied for different cases include: oquad, cmdiff, cmxfin, cmyfin, and cmzfin. Figure A-3 is the input file for the 64 x 64 x 64 voxel phantom with S₄ quadrature and the DDZ differencing scheme.

```

/* TITAN input file hrtk_titan.inp
/* generated by PENMSHXP version 2.0 (August 2008)
/* total number of fine meshes: 262144
/* total number of coarse meshes: 1
/* Number of zlevs: 1
/* Number of coarse mesh per z lev: 1
/******
#0 /general transport parameter quadrature
/acceptable keywords: nquad, tquad, oquad, splitq(multi), ncmesh, numsrc, nummat, numgrp
/ nquad: number of quadratures.
/ tquad: type of quadratures, NumOfEntry=nquad (0=LevelSym, 1=LegendreCheby)
/ oquad: order of quadratures, NumOfEntry=nquad (even number)
/ splitq: ordinate splitting setup
/ splitq=QuadId, #Splitting, SplitDirIDs, SplitOrders, SplitType, Topnums, alpha
/ QuadId=1, 2,..or nquad, #Splitting=num of splitting directions
/ #Splitting= total num of splitting directions in QuadID
/ SplitDirIDs=splitting direction IDs in the base quadrature,NumOfEntry=#Splitting
/ SplitOrder=splitting order for each splitting direction,NumOfEntry=#Splitting
/ SplitType: 0=rectangular, 1=Pn-Tn, 2=circular, NumOfEntry=#Splitting
/ Topnum: rectangular: unused; Pn-Tn: Num of dirs on top level; circular: # of circles.
NumOfEntry=#Splitting
/ alpha: rectangular/Pn-Tn: unused (angular range for future version); circular: derail angle.
NumOfEntry=#Splitting
/ ncmesh: # of coarse mesh along x, y ,z . NumOfEntry=3
/ numsrc: # of sources . NumOfEntry=1
/ nummat: # of materials . NumOfEntry=1
/ numgrp: # of groups . NumOfEntry=1
ncmesh=1 1 1
numsrc=1
nummat=13
numgrp=3
nquad=1
tquad=0
oquad=4
/******

```

Figure A-3. TITAN input file for SPECT simulation

```

#1 /Section 1 : Geometry setup
/acceptable keywords: dcpara, xcmbnd, ycmbnd, zcmbnd, cmxfin, cmyfin, cmzfin, cmsolv,cmdiff,cmquad
/ dcpara: spatial parallel factor along x, y, z (future version)
/ xcmbnd, ycmbnd, zcmbnd: x,y,z coarse mesh boundaries, NumOfEntry=ncmesh+1
/ cmxfin, cmyfin, cmzfin: fine mesh number along x,y,z for each coarse mesh, NumOfEntry=TotNumOfCM
/ cmsolv: solver ID for each corase mesh, 0=Sn, 1=Characteristics
/ cmdiff: Differencing Scheme ID for each corase mesh, 1=DD with fixup, 2=DTW
/ cmquad: Quadrature ID for each coarse mesh, available values=1,2,... or nquad
dcpara=1,1,1
xcmbnd= 0.00000E+00 4.00000E+01
ycmbnd= 0.00000E+00 4.00000E+01
zcmbnd= 0.00000E+00 4.00000E+01
cmxfin=64
cmyfin=64
cmzfin=64
cmsolv=0
cmdiff=1
cmquad=1
/*****

#2 /Section 2 : Fine mesh mat. number
/acceptable keywords: cmmatn (multi)
/ cmmatn=CM#, mat # for each fine mesh in this CM
cmmatn=1
602R1 16R2 43R1 26R2 36R1 30R2 31R1 35R2 27R1 16R2 6R3 16R2 25R1 12R2 4R3 6R4
2R3 16R2 23R1 10R2 2R3 11R4 3R3 16R2 21R1 10R2 3 13R4 4R3 2R2 2R3 12R2 19R1
9R2 2R3 14R4 8R3 2 3 2R4 2R3 7R2 18R1 8R2 3 15R4 3R3 4 2R3 2 2R3 2 3 3R4 2R3
...
...
...
39R1 26R2 38R1 26R2 37R1 27R2 37R1 28R2 36R1 28R2 36R1 28R2 36R1 28R2 36R1
28R2 36R1 28R2 36R1 29R2 35R1 29R2 35R1 28R2 37R1 27R2 37R1 27R2 37R1 27R2
38R1 25R2 40R1 23R2 42R1 21R2 44R1 18R2 48R1 14R2 1177R1
/*****

```

Figure A-3. Continued

```

#3 /Section 3 : src distribution
/acceptable keywords: srcloc, srcmag, srcspm, srcdis (multi), keffin
/ srcloc: source location (CM number where source located), NumOfEntry=numsrc
/ srcmag: source strength (srcmag*srcdis=src density in that fm, #/cm3-sec), NumOfEntry=numsrc
/ srcspm: source spectrum (srcmag*srcdis*srcspm=src density for a group), NumOfEntry=numsrc*numgrp
/ srcdis: source spatial distribution srcdis=Src#, FineMeshDist
/ keffin: initial Keff guess, NumOfEntry=1, (more entries reserved for future Keff accelaration
srcloc=1
srcmag=2.62901E+05
srcspm=1.000E+00 2R0.000E+00
srcdis=1
602R0.00000E+00 16R7.60743E-06 43R0.00000E+00 26R7.60743E-06 36R0.00000E+00
30R7.60743E-06 31R0.00000E+00 35R7.60743E-06 27R0.00000E+00 16R7.60743E-06
6R0.00000E+00 16R7.60743E-06 25R0.00000E+00 12R7.60743E-06 12R0.00000E+00
...
...
...
37R0.00000E+00 27R7.60743E-06 37R0.00000E+00 27R7.60743E-06 38R0.00000E+00
25R7.60743E-06 40R0.00000E+00 23R7.60743E-06 42R0.00000E+00 21R7.60743E-06
44R0.00000E+00 18R7.60743E-06 48R0.00000E+00 14R7.60743E-06 1177R0.00000E+00
/*****
#4 /Section 4 : xs data
/acceptable keywords: xsname, xstype, numcmt, xstihm, legord, legoxs,xstchi
/ xsname: cross section file name (charater entry)
/ xstype: 0=(2I-1) is not pre-multiplied; 1=(2I-1) pre-multiplied
/ numcmt: number of comment lines in between material xs block
/ xstihm: xs table total length, (SigmaTot@Column 3 always
/ xstihm=3+numgrp: SigmaSelfScatter@4 , downscattering only
/ xstihm=3+(2*numgrp-1): SigmaSelfScatter@(3+numgrp) , upperscattering
/ keffin: initial Keff guess, NumOfEntry=1, (more entries reserved for future Keff accelaration
xsname=hrtk.xs
legord=3 legoxs=3
xstype=0
xstihm=6
xstchi=1.0000E+00 2R0.0000E+00 12Q3
numcmt=1
/*****

```

Figure A-3. Continued

```

#5 /Section 5 : boundary cond. and tol.
/acceptable keywords: tolinn,tolout,maxinn,maxout,xminus,xpluss,yminus,ypluss,zminus,zpluss
/ tolinn: inner iteration (within-group) tolerance, negative value: adjustable for keff loop
/ tolout: outer iteration (keff loop) tolerance, negative value: adjustable for keff loop
/ maxinn: maxium inner iteration number, negative value: adjustable for keff loop
/ maxout: maxium outer iteration number, negative value: adjustable for keff loop
/ xminus,xpluss,yminus,ypluss,zminus,zpluss : Boundary conditions at -x,+x, -y,+y, -z,+z
/ =0: vaccum; =1 albedos for each group: reflective
xminus=0
xpluss=0
yminus=0
ypluss=0
zminus=0
zpluss=0
tolinn= 1.00000E-03
tolout= 1.00000E-05
maxout=50
maxinn=10
/*****
#10 /Section 10 (Optional) : SPECT
/acceptable keywords: numang,iniang,engang, vecaxs, posaxs, radius, sptcir,detsiz
/ numang: number of projection angles
/ iniang, engang: rotation starting and ending angles in degree
/ vecaxs: rotation axis vector, NumOfEntry=3
/ posaxs: position of rotation vector, NumOfEntry=3
/ radius: rotation radius
/ sptcir: circular splitting for collimators
/ =splitting order(num of dir on one circle), number of circles, collimator angle
/ detsiz: detector size along x and y, in cm and pixels
/ =x-size(cm), y-size(cm), #OfPix along x, #ofPix along y
numang=4
iniang=-9.00000E+01
engang= 2.70000E+02
vecaxs= 0.00000E+00 0.00000E+00 1.00000E+00
posaxs= 2.00000E+01 2.00000E+01 2.00000E+01
radius= 3.00000E+01
sptcir=3 1 1.15350E-02
detsiz= 4.00000E+01 4.00000E+01 64 64

```

Figure A-3. Continued

- The MCNP5 input file is in Figure A-5 for the four-hole case. To create the single and nine-hole cases the number, position, and size of the collimator hole surfaces have to be changed.

```

c SPECT Simulation of Heart Phantom
c CELLS
51 0 1 -2 3 -4 5 -6 fill=50 imp:p=1
52 1 1.0 (-1:2:-3:4:-5:6) (31) (29) (-21) imp:p=1
53 0 21 imp:p=0
54 0 -7 1 -8 3 -9 5 lat=1 u=50 imp:p=1 fill=0:63 0:63 0:63
  1 601R 2 15R 1 42R 2 25R 1 35R 2 29R
  1 30R 2 34R 1 26R 2 15R 3 5R 2 15R
  1 24R 2 11R 3 3R 4 5R 3 1R 2 15R
...
...
...
  1 36R 2 26R 1 36R 2 26R 1 36R 2 26R
  1 37R 2 24R 1 39R 2 22R 1 41R 2 20R
  1 43R 2 17R 1 47R 2 13R 1 1176R
  1 1 1.0 -7 1 -8 3 -9 5 u=1 imp:p=1 $ Air (-0.0012)
101 0 (7:-1:8:-3:9:-5) u=1 imp:p=0
  2 2 1.0 -22 u=2 imp:p=1 $ water(body) (-1.0)
102 0 22 u=2 imp:p=0
  3 3 1.0 -22 u=3 imp:p=1 $ intestine (-1.03)
103 0 22 u=3 imp:p=0
  4 4 1.0 -22 u=4 imp:p=1 $ lung (-0.3)
104 0 22 u=4 imp:p=0
  5 5 1.0 -22 u=5 imp:p=1 $ rib bone (-1.92)
105 0 22 u=5 imp:p=0
  6 6 1.0 -22 u=6 imp:p=1 $ liver (-1.06)
106 0 22 u=6 imp:p=0
  7 7 1.0 -22 u=7 imp:p=1 $ kidney (-1.05)
107 0 22 u=7 imp:p=0
  8 8 1.0 -22 u=8 imp:p=1 $ spine bone (-1.42)
108 0 22 u=8 imp:p=0
  9 9 1.0 -22 u=9 imp:p=1 $ muscle (-1.05)
109 0 22 u=9 imp:p=0
  10 10 1.0 -22 u=10 imp:p=1 $ cartilage (-1.10)
110 0 22 u=10 imp:p=0
  11 11 1.0 -22 u=11 imp:p=1 $ spleen (-1.06)
111 0 22 u=11 imp:p=0
  12 12 1.0 -22 u=12 imp:p=1 $ heart (-1.05)
112 0 22 u=12 imp:p=0
  13 13 1.0 -22 u=13 imp:p=1 $ blood (-1.06)
113 0 22 u=13 imp:p=0
  14 14 1.0 -22 u=14 imp:p=1 $ NaI(Tl) Detector (-3.67)
114 0 22 u=14 imp:p=0
  56 1 1.0 -7 1 -8 3 -9 5 u=56 imp:p=1
156 0 (7:-1:8:-3:9:-5) u=56 imp:p=0
  18 9 1.0 -22 u=18 imp:p=1 $ Muscle without source

```

Figure A-5. MCNP5 input file for the four-hole case

```

118 0 22 u=18 imp:p=0
63 1 1.0 -7 1 -8 3 -9 5 u=55 imp:p=1
64 0 (7:-1:8:-3:9:-5) u=55 imp:p=0
71 0 -29 fill=61 imp:p=1 $ collimator
73 15 1.0 30 -31 imp:p=1 $ Lead around detector voxel
74 1 1.0 -24 u=61 imp:p=1 $ Holes in collimator
75 1 1.0 -25 u=61 imp:p=1
76 1 1.0 -26 u=61 imp:p=1
77 1 1.0 -27 u=61 imp:p=1
79 15 1.0 24 25 26 27 u=61 imp:p=1
81 1 1.0 -30 imp:p=1 $ Detector voxel cell

```

c Surfaces

```

1 px -20.0
2 px 20.0
3 py -20.0
4 py 20.0
5 pz -20.0
6 pz 20.0
7 px -19.375
8 py -19.375
9 pz -19.375
21 so 100
22 so 101
24 c/y 2.65625 0.78125 0.1417
25 c/y 2.65625 1.09375 0.1417
26 c/y 2.96875 0.78125 0.1417
27 c/y 2.96875 1.09375 0.1417
29 rpp 1.875 3.75 -30 -24.3 0 1.875 $ Outer edge of lead
30 rpp 2.5 3.125 -30.625 -30.0 0.625 1.25
31 rpp 1.875 3.75 -30.625 -30.0 0 1.875

```

Figure A-5. Continued

```

mode p
m1 1000.22g 1.0
m2 2000.22g 1.0
m3 3000.22g 1.0
m4 4000.22g 1.0
m5 5000.22g 1.0
m6 6000.22g 1.0
m7 7000.22g 1.0
m8 8000.22g 1.0
m9 9000.22g 1.0
m10 10000.22g 1.0
m11 11000.22g 1.0
m12 12000.22g 1.0
m13 13000.22g 1.0
m14 14000.22g 1.0
m15 15000.22g 1.0
sdef erg=d1 par=2 pos=d2
si1 h 1.0E-02 9.835E-02 1.265E-01 1.546E-01
sp1 d 0 0 0 1
si2 L
-19.687500 -19.687500 -19.687500 -19.062500 -19.687500 -19.687500
-18.437500 -19.687500 -19.687500 -17.812500 -19.687500 -19.687500
-17.187500 -19.687500 -19.687500 -16.562500 -19.687500 -19.687500
...
...
...
16.562500 19.687500 19.687500 17.187500 19.687500 19.687500
17.812500 19.687500 19.687500 18.437500 19.687500 19.687500
19.062500 19.687500 19.687500 19.687500 19.687500 19.687500
sp2 d
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
...
...
...
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
CTME 3360
mgopt f 3
e0 9.835E-02 1.265E-01 1.546E-01
f4:p 81
print 110

```

Figure A-5. Continued

- The TITAN input file for the adjoint methodology follows the same form as the SPECT input file in Figure A-3 except that Section 10 is no longer used. In Figure A-6, the S_{62} quadrature is used with the DTW differencing scheme. The srcang parameter specifies the adjoint source in four octants along the quadrature direction closest to the +y-axis. The srcdis parameter indicates the single voxel that the adjoint source is in. The splitq parameter is commented out in this figure, but would be used for ordinate splitting.

```

/* TITAN input file hrtk_titan.inp
/* generated by PENMSHXP version 2.0 (August 2008)
/* total number of fine meshes: 262144
/* total number of coarse meshes: 1
/* Number of zlevs: 1
/* Number of coarse mesh per z lev: 1
/*****
#0 /general transport parameter quadrature
/acceptable keywords: nquad, tquad, oquad, splitq(multi), ncmesh, numsrc, nummat, numgrp
/ nquad: number of quadratures.
/ tquad: type of quadratures, NumOfEntry=nquad (0=LevelSym, 1=LegendreCheby)
/ oquad: order of quadratures, NumOfEntry=nquad (even number)
/ splitq: ordinate splitting setup
/ splitq=QuadId, #Splitting, SplitDirIDs, SplitOrders, SplitType, Topnums, alpha
/ QuadId=1, 2,..or nquad, #Splitting=num of splitting directions
/ #Splitting= total num of splitting directions in QuadID
/ SplitDirIDs=splitting direction IDs in the base quadrature,NumOfEntry=#Splitting
/ SplitOrder=splitting order for each splitting direction,NumOfEntry=#Splitting
/ SplitType: 0=rectangular, 1=Pn-Tn, 2=circular, NumOfEntry=#Splitting
/ Topnum: rectangular: unused; Pn-Tn: Num of dirs on top level; circular: # of circles.
NumOfEntry=#Splitting
/ alpha: rectangular/Pn-Tn: unused (angular range for future version); circular: derail angle.
NumOfEntry=#Splitting
/ ncmesh: # of coarse mesh along x, y ,z . NumOfEntry=3
/ numsrc: # of sources . NumOfEntry=1
/ nummat: # of materials . NumOfEntry=1
/ numgrp: # of groups . NumOfEntry=1
ncmesh=1 3 1
numsrc=1
nummat=13
numgrp=3
nquad=1
tquad=1
oquad=62
/splitq=1 1 496 6 1 1 1
/*****

```

Figure A-6. TITAN input file for adjoint methodology

```

#1 /Section 1 : Geometry setup
/acceptable keywords: dcpara, xcmbnd, ycmbnd, zcmbnd, cmxfn, cmyfn, cmzfn, cmsolv,cmdiff,cmquad
/ dcpara: spatial parallel factor along x, y, z (future version)
/ xcmbnd, ycmbnd, zcmbnd: x,y,z coarse mesh boundaries, NumOfEntry=ncmesh+1
/ cmxfn, cmyfn, cmzfn: fine mesh number along x,y,z for each coarse mesh, NumOfEntry=TotNumOfCM
/ cmsolv: solver ID for each corase mesh, 0=Sn, 1=Characteristics
/ cmdiff: Differencing Scheme ID for each corase mesh, 1=DD with fixup, 2=DTW
/ cmquad: Quadrature ID for each coarse mesh, available values=1,2,... or nquad
dcpara=1,1,1
xcmbnd= 0.00000E+00 4.00000E+01
ycmbnd= -1.0625E+01 -1.0E+01 0.00000E+00 4.00000E+01
zcmbnd= 0.00000E+00 4.00000E+01
cmxfn=64 64 64
cmyfn=1 1 64
cmzfn=64 64 64
cmsolv=0 1 0
cmdiff=2 2 2
cmquad=1 1 1
/.....
#2 /Section 2 : Fine mesh mat. number
/acceptable keywords: cmmatn (multi)
/ cmmatn=CM#, mat # for each fine mesh in this CM
cmmatn=1
4096R1
cmmatn=2
1
cmmatn=3
602R1 16R2 43R1 26R2 36R1 30R2 31R1 35R2 27R1 16R2 6R3 16R2 25R1 12R2 4R3 6R4
...
...
...
38R1 25R2 40R1 23R2 42R1 21R2 44R1 18R2 48R1 14R2 1177R1
/.....
#3 /Section 3 : src distribution
/acceptable keywords: srcloc, srcmag, srcspm, srodis (multi), keffin
/ srcloc: source location (CM number where source located), NumOfEntry=numsrc
/ srcmag: source strength (srcmag*srodis=src density in that fm, #/cm3-sec), NumOfEntry=numsrc
/ srcspm: source spectrum (srcmag*srodis*srcspm=src density for a group), NumOfEntry=numsrc*numgrp
/ srodis: source spatial distribution srodis=Src#, FineMeshDist
/ keffin: initial Keff guess, NumOfEntry=1, (more entries reserved for future Keff acelaration
srcloc=1
srcmag=1.000E+00
srcspm=3R1.000E+00
srodis=1
2148R0.00000E+00 1.000E+00 1947R0.00000E+00
srcang=1 1 495R0 1
srcang=1 3 495R0 1
srcang=1 6 495R0 1
srcang=1 8 495R0 1
/.....

```

Figure A-6. Continued

```

#4 /Section 4 : xs data
/acceptable keywords: xsname, xstype, numcmt, xstihm, legord, legoxs,xstchi
/ xsname: cross section file name (charater entry)
/ xstype: 0=(2l-1) is not pre-multiplied; 1=(2l-1) pre-multiplied
/ numcmt: number of comment lines in between material xs block
/ xstihm: xs table total length, (SigmaTot@Column 3 always
/ xstihm=3+numgrp: SigmaSelfScatter@4 , downscattering only
/ xstihm=3+(2*numgrp-1): SigmaSelfScatter@(3+numgrp) , upperscattering
/ keffin: initial Keff guess, NumOfEntry=1, (more entries reserved for future Keff accelaration
xsname=hrtk.xs
legord=3 legoxs=3
xstype=0
xstihm=6
xstchi=1.0000E+00 2R0.0000E+00 12Q3
numcmt=1
/*****
#5 /Section 5 : boundary cond. and tol.
/acceptable keywords: tolinn,tolout,maxinn,maxout,xminus,xpluss,yminus,ypluss,zminus,zpluss
/ tolinn: inner iteration (within-group) tolerance, negative value: adjustable for keff loop
/ tolout: outer iteration (keff loop) tolerance, negative value: adjustable for keff loop
/ maxinn: maxium inner iteration number, negative value: adjustable for keff loop
/ maxout: maxium outer iteration number, negative value: adjustable for keff loop
/ xminus,xpluss,yminus,ypluss,zminus,zpluss : Boundary conditions at -x,+x, -y,+y, -z,+z
/ =0: vaccum; =1 albedos for each group: reflective
xminus=0
xpluss=0
yminus=0
ypluss=0
zminus=0
zpluss=0
tolinn= 1.00000E-03
tolout= 1.00000E-05
maxout=50
maxinn=25

```

Figure A-6. Continued

APPENDIX B DATA PROCESSING CODES

The codes in this section were written to prepare input files, format data and analyze outputs.

- The imageCompare program was developed to compare the binary projection images created by the SIMIND and TITAN codes. The program prompts the user for two input files and gives the infinity-norm and 2-norm of the second file relative to the first.

```

PROGRAM imageCompare
  IMPLICIT NONE

  REAL(4), ALLOCATABLE :: A(:,:,:), B(:,:,:), normA(:,:,:), normB(:,:,:)
  INTEGER :: i, j, k, IOS, xDim, yDim, zDim, iMax, jMax, kMax, skipA, skipB, pixCnt
  REAL(4) :: relErr, maxErr, norm, maxA, maxB
  CHARACTER(25) :: fileA, fileB

  WRITE(*,*) 'Files to compare: '
  WRITE(*,*) 'File A: '
  READ(*,*) fileA
  WRITE(*,*) 'File B: '
  READ(*,*) fileB
  WRITE(*,*) 'Enter image dimensions: x y z '
  READ(*,*) xDim, yDim, zDim

  ALLOCATE(A(xDim, yDim, zDim), B(xDim, yDim, zDim))
  ALLOCATE(normA(xDim,yDim,zDim), normB(xDim,yDim,zDim))
  OPEN(1,file=fileA,FORM='binary',IOSTAT=IOS)
  WRITE(*,*) 'Status A: ', IOS
  OPEN(2,file=fileB,FORM='binary',IOSTAT=IOS)
  WRITE(*,*) 'Status B: ', IOS
  WRITE(*,*) 'dimensions: ', xDim, yDim, zDim
  READ(1) A
  CLOSE(1)
  READ(2) B
  CLOSE(2)
  OPEN(1,file='Anorm.out',FORM='binary',IOSTAT=IOS)
  OPEN(2,file='Bnorm.out',FORM='binary',IOSTAT=IOS)
  OPEN(3,file='Anorm_hrt.out',FORM='binary',IOSTAT=IOS)
  OPEN(4,file='Bnorm_hrt.out',FORM='binary',IOSTAT=IOS)

  DO k=1,zDim
    maxA=maxval(A(:, :,k))
    maxB=maxval(B(:, :,k))
    normA(:, :,k)=A(:, :,k)/maxA
    normB(:, :,k)=B(:, :,k)/maxB
  END DO
  WRITE(1) normA
  WRITE(2) normB
  CLOSE(1)
  CLOSE(2)
  OPEN(1,file='results.out',FORM='FORMATTED',IOSTAT=IOS)
  WRITE(1,'(a)') 'For entire projection: '
  norm=0
  maxErr=0

```

Figure B-1. Code to compare binary projection images from the TITAN and SIMIND codes

```

DO k=1,zDim
  norm=0
  maxErr=0
  DO j=1,yDim
    DO i=1, xDim
      relErr=0
      IF(normA(i,j,k)>0 .AND. normB(i,j,k)>0) relErr=ABS((normB(i,j,k)-normA(i,j,k))/normA(i,j,k))
      IF(relErr>abs(maxErr)) THEN
        maxErr=(normB(i,j,k)-normA(i,j,k))/normA(i,j,k)
        iMax=i
        jMax=j
        kMax=k
      END If
      norm=norm+(relErr*relErr)
    END DO
  END DO
  WRITE(1,'(a,i3,a,f,a,i3,a,i3,a,f)') 'k=', k, ' max error=', maxErr, ' at (', iMax, ',', jMax, ') 2-norm=',
SQRT(norm)
END DO
WRITE(1,'(a)') 'For heart only:'
norm=0
maxErr=0
DO k=1,zDim
  pixCnt=0
  norm=0
  maxErr=0
  DO j=20,42
    DO i=15, 48
      relErr=0
      IF(normA(i,j,k)>0 .AND. normB(i,j,k)>0) THEN
        relErr=ABS((normB(i,j,k)-normA(i,j,k))/normA(i,j,k))
      END IF
      IF(relErr>abs(maxErr)) THEN
        maxErr=(normB(i,j,k)-normA(i,j,k))/normA(i,j,k)
        iMax=i
        jMax=j
        kMax=k
      END If
      norm=norm+(relErr*relErr)
    END DO
  END DO
  WRITE(1,'(a3,i3,a14,f,a5,i3,a1,i3,a12,f,a9,i)') 'k=', k, ' max error=', maxErr, ' at (', iMax, ',', jMax, ')
2-norm=', SQRT(norm)
END DO
WRITE(1,'(a)') 'For heart pixels >=0.65 only:'
WRITE(3) normA(15:48,20:42,:)

```

Figure B-1. Continued

```

WRITE(4) normB(15:48,20:42,:)
CLOSE(3)
CLOSE(4)
pixCnt=0
norm=0
maxErr=0
DO k=1,zDim
  pixCnt=0
  norm=0
  maxErr=0
  DO j=20,42
    DO i=15, 48
      relErr=0
      IF(normA(i,j,k)<0.65) normA(i,j,k)=0.0
      IF(normB(i,j,k)<0.65) normB(i,j,k)=0.0
      IF(normA(i,j,k)>0 .AND. normB(i,j,k)>0) THEN
        relErr=ABS((normB(i,j,k)-normA(i,j,k))/normA(i,j,k))
        pixCnt=pixCnt+1
      END IF
      IF(relErr>abs(maxErr)) THEN
        maxErr=(normB(i,j,k)-normA(i,j,k))/normA(i,j,k)
        iMax=i
        jMax=j
        kMax=k
      END If
      norm=norm+(relErr*relErr)
    END DO
  END DO
  WRITE(1,'(a3,i3,a14,f,a5,i3,a1,i3,a12,f,a9,i)') 'k=', k, ' max error=', maxErr, ' at (', iMax, ', ', jMax, ')
  2-norm=', SQRT(norm), ' cnts: ', pixCnt
END DO
WRITE(3) normA(15:48,20:42,:)
WRITE(4) normB(15:48,20:42,:)
CLOSE(3)
CLOSE(4)
CLOSE(1)
DEALLOCATE(A,B,normA,normB)
END PROGRAM imageCompare

```

Figure B-1. Continued

- The response program in Figure B-2 reads in the flux distribution from an adjoint calculation in TITAN and returns the detector response. It requires the problem source distribution.

```

PROGRAM response
IMPLICIT NONE
REAL(4), ALLOCATABLE :: flux(:,:,:), flux1(:,:,:), flux2(:,:,:), flux3(:,:,:), src(:,:,:)
REAL(4) :: dummy
REAL(4) :: R1
Integer :: i, j, k, dInteger, MOC, dimMOC
CHARACTER(LEN=25) :: fluxFile, prbName, dChar
ALLOCATE(flux1(64,64,64))
ALLOCATE(flux2(64,64,64))
ALLOCATE(flux3(64,64,64))
ALLOCATE(flux(64,64,64))
ALLOCATE(src(64,64,64))

flux1(:,:)=0.0
flux2(:,:)=0.0
flux3(:,:)=0.0
src(:,:)=0.0
resp(:,:)=0.0
dChar='123456789'

WRITE(*,*) 'Please enter the problem name: '
READ(*,*) prbName
! Flip the adjoint flux groups back as they are read in.
fluxFile=trim(prbName)//'1.flx'
OPEN(1,file=fluxFile)
READ(1,*) dChar

WRITE(*,*) 'Is this a MOC problem? ( 1=yes 0=no )'
READ(*,*) MOC
IF(MOC==1) THEN
  WRITE(*,*) 'Is this a 64 by 64 mesh (enter 1), a 128 by 128 mesh (enter 2), or a 64 mesh MOC region (enter3) ?'
  READ(*,*) dimMOC
  ! Skip through first coarse mesh
  IF(dimMOC==1.OR.dimMOC==3) THEN
    DO i=1,4096
      READ(1,*) dChar
    END DO
  ! Skip through second coarse mesh
  IF(dimMOC==1) THEN
    DO i=1,4096
      READ(1,*) dChar
    END DO
  ELSE
    DO i=1,262144
      READ(1,*) dChar

```

Figure B-2. Program to calculate the detector response using the adjoint flux produced from an adjoint calculation in the TITAN code

```

    END DO
  END IF
ELSE IF(dimMOC==2) THEN
  DO i=1,16384
    READ(1,*) dChar
  END DO
! Skip through second coarse mesh
  DO i=1,32768
    READ(1,*) dChar
  END DO
END IF
END IF

DO k=1,64
  DO j=1,64
    DO i=1,64
      READ(1,*) dInteger, dInteger, dummy, dummy, dummy, flux3(i,j,k)
    END DO
  END DO
END DO
WRITE(*,*) 'File ', fluxFile, ' read in'
CLOSE(1)

fluxFile=trim(prbName)//'2.flx'
OPEN(1,file=fluxFile)
READ(1,*) dChar
IF(MOC==1) THEN
  IF(dimMOC==1.OR.dimMOC==3) THEN
! Skip through first coarse mesh
    DO i=1,4096
      READ(1,*) dChar
    END DO
! Skip through second coarse mesh
    IF(dimMOC==1) THEN
      DO i=1,4096
        READ(1,*) dChar
      END DO
    ELSE
      DO i=1,262144
        READ(1,*) dChar
      END DO
    END IF
  ELSE IF(dimMOC==2) THEN
    DO i=1,16384
      READ(1,*) dChar
    END DO
  
```

Figure B-2. Continued

```

! Skip through second coarse mesh
  DO i=1,32768
    READ(1,*) dChar
  END DO
END IF
END IF
DO k=1,64
  DO j=1,64
    DO i=1,64
      READ(1,*) dInteger, dInteger, dummy, dummy, dummy, flux2(i,j,k)
    END DO
  END DO
END DO
WRITE(*,*) 'File ', fluxFile, ' read in'
CLOSE(1)

fluxFile=trim(prbName)//'3.flx'
OPEN(1,file=fluxFile)
READ(1,*) dChar
IF(MOC==1) THEN
  IF(dimMOC==1.OR.dimMOC==3) THEN
! Skip through first coarse mesh
    DO i=1,4096
      READ(1,*) dChar
    END DO
! Skip through second coarse mesh
    IF(dimMOC==1) THEN
      DO i=1,4096
        READ(1,*) dChar
      END DO
    ELSE
      DO i=1,262144
        READ(1,*) dChar
      END DO
    END IF
  ELSE IF(dimMOC==2) THEN
    DO i=1,16384
      READ(1,*) dChar
    END DO
! Skip through second coarse mesh
    DO i=1,32768
      READ(1,*) dChar
    END DO
  END IF
END IF
DO k=1,64

```

Figure B-2. Continued

```

        DO j=1,64
            DO i=1,64
                READ(1,*) dInteger, dInteger, dummy, dummy, dummy, flux1(i,j,k)
            END DO
        END DO
    END DO
    WRITE(*,*) 'File ', fluxFile, ' read in'
    CLOSE(1)

    OPEN(1,file='hrtk_fix_act_1.bin',form='binary')
    READ(1) src
    WRITE(*,*) 'Source file read in.'
    CLOSE(1)
    WRITE(*,*) 'Source sum: ', sum(src)

    ! Write out binary flux files
    OPEN(1,file='flux1.bin',form='binary')
    WRITE(1) flux1(:,:,:)
    CLOSE(1)
    OPEN(1,file='flux2.bin',form='binary')
    WRITE(1) flux2(:,:,:)
    CLOSE(1)
    OPEN(1,file='flux3.bin',form='binary')
    WRITE(1) flux3(:,:,:)
    CLOSE(1)

    R1=0
    DO k=1,64
        DO j=1,64
            DO i=1,64
                R1=R1+(src(i,j,k)*flux1(i,j,k))
            END DO
        END DO
    END DO
    WRITE(*,*) 'Total detector response: ', R1

    DEALLOCATE(flux1,flux2,flux3)
    DEALLOCATE(src,flux)

    END PROGRAM response

```

Figure B-2. Continued

- The phanMat program in Figure B-3 turns the NCAT material distribution into an array of material numbers that can be directly placed into the MCNP5 input. In the MCNP5 input, these material numbers represent universes in a lattice. The program creates a simple distribution of material numbers and a more compact distribution using a shorthand notation readable by MCNP5.

```

PROGRAM phanMat
! Convert NCAT material distribution to material numbers for use in MCNP5
IMPLICIT NONE
  REAL(4) :: matCoef(18), matDens(18), atr(64,64,64), diff
  REAL(4) :: pixSize=0.625, materials(18)
  CHARACTER(LEN=13) :: matName(18)
  CHARACTER(LEN=80) matStr
  INTEGER :: i, j, k, m, matNum(64,64,64), num, nline, l, dimx, dimy, dimz, remain
  INTEGER :: mNum(18), numCur, current, strPos, strLen, entNum
! Material linear attenuation coefficients are input per pixel and below is per cm.
  matName(2)='Body (water)'
  matCoef(2)=0.149415776
  matDens(2)=1.0
  matName(9)='Muscle'
  matCoef(9)=0.1558294248
  matDens(9)=1.05
  matName(18)='Adipose (fat)'
  matCoef(18)=0.1384
  matDens(18)=0.92
  matName(4)='Lung'
  matCoef(4)=0.0430442688
  matDens(4)=0.3
  matName(8)='Spine Bone'
  matCoef(8)=0.181190944
  matDens(8)=1.42
  matName(5)='Rib Bone'
  matCoef(5)=0.219484608
  matDens(5)=1.92
  matName(13)='Blood'
  matCoef(13)=0.157330176
  matDens(13)=1.06
  matName(12)='Heart'
  matCoef(12)=0.1560583584
  matDens(12)=1.05
  matName(7)='Kidney'
  matCoef(7)=0.155943456
  matDens(7)=1.05
  matName(6)='Liver'
  matCoef(6)=0.157318352
  matDens(6)=1.06
  matName(16)='Lymph'
  matCoef(16)=0.1535
  matDens(16)=1.03
  matName(14)='Pancreas'
  matCoef(14)=0.1548
  matDens(14)=1.04
  matName(11)='Spleen'
  matCoef(11)=0.157420576

```

Figure B-3. Program to convert NCAT material distribution into material numbers for MCNP5 input

```

matDens(11)=1.06
matName(3)='Intestine'
matCoef(3)=0.153245376
matDens(3)=1.03
matName(15)='Skull'
matCoef(15)=0.2422
matDens(15)=1.61
matName(10)='Cartilage'
matCoef(10)=0.162762448
matDens(10)=1.10
matName(17)='Brain'
matCoef(17)=0.0969
matDens(17)=1.04
matName(1)='Air'
matCoef(1)=0.0
matDens(1)=0.0012
num=0
materials(:)=0

OPEN(1,file='hrtk_fix_atn_1.bin',form='binary')
READ(1) atn
WRITE(*,*) 'File hrtk_atn_1.bin read in'
CLOSE(1)

dimx=64
dimy=64
dimz=64

! Convert attenuation coefficients per pixel to per cm
DO k=1,dimz
  DO j=1,dimy
    DO i=1,dimx
      atn(i,j,k)=atn(i,j,k)/pixSize
    END DO
  END DO
END DO
DO k=1,dimz
  DO j=1,dimy
    DO i=1,dimx
      DO m=1,18
        IF(abs(matCoef(m)-atn(i,j,k))<0.00001) THEN
          matNum(i,j,k)=m
          mNum(m)=mNum(m)+1
          exit
        END IF
      END DO
    END DO
  END DO
END DO

```

Figure B-3. Continued

```

END DO
nline=(dimx/18) + 1
remain=mod(dimx,18)
OPEN(2,file='hrtk_mat_1.txt',form='formatted')
DO k=1,dimz
  DO j=1,dimy
    DO i=1,nline-1
      l=((i-1)*18)+1
      WRITE(2,'(a5,18i4)') ' ', matNum(l:l+17,j,k)
    END DO
    IF(remain>0) WRITE(2,'(a5,17i4)') ' ', matNum(l+18:j,k)
  END DO
END DO
CLOSE(2)
OPEN(2,file='hrtk_mat_1b.txt',form='formatted')
current=matNum(1,1,1)
numCur=-1
strPos=1
entNum=0
DO k=1, dimz
  DO j=1, dimy
    DO i=1, dimx
      IF(current==matNum(i,j,k)) THEN
        numCur=numCur+1
      ELSE
        entNum=entNum+1
        WRITE(matStr(strPos:strPos+10),'(" ",i2," ",i6,"R")) current, numCur
        strPos=strPos+11
        IF(entNum==6) THEN
          WRITE(2,'(" ",a)') matStr(1:strPos-1)
          entNum=0
          strPos=1
        END IF
        current=matNum(i,j,k)
        numCur=0
      END IF
    END DO
  END DO
END DO
WRITE(matStr(strPos:strPos+10),'(" ",i2," ",i6,"R")) current, numCur
strPos=strPos+11
WRITE(2,'(" ",a)') matStr(1:strPos-1)
CLOSE(2)
WRITE(*,*) 'mNum: ', mNum

END PROGRAM phanMat

```

Figure B-3. Continued

- The regression program in Figure B-4 was written to perform a linear regression on an input file containing the number of points followed by the x and y values.

```

#include <iostream>
#include <fstream>
#include <cstdlib>
#include <iomanip>
using namespace std;

int main (int argc, char * const argv[]) {

    int i;
    int n;
    char temp[100];
    char fileName[100];
    double cov, varx, vary, xavg, yavg, b, a, rsq, xint;

    cout << "Enter file name to open: " << endl;
    cin >> fileName;
    ifstream fileIn;
    fileIn.open (fileName);
    if(!fileIn) {
        cout << "Couldn't open file";
    }
    fileIn.getline(temp,100);
    n=atoi(temp);
    cout << "n: " << n << endl;

    double xi[n], yi[n];
    i=0;
    while(i<n) {
        fileIn.getline(temp, 100);
        xi[i]=atof(temp);
        i++;
    }
    i=0;
    while(i<n) {
        fileIn.getline(temp, 100);
        yi[i]=atof(temp);
        i++;
    }
    fileIn.close();
    cout << "File read in" << endl;

    for(i=0;i<n;i++) {cout << "xi: " << setw(15) << setprecision(10) << xi[i] << endl;}
    for(i=0;i<n;i++) {cout << "yi: " << setw(15) << setprecision(10) << yi[i] << endl;}

    xavg=0;
    yavg=0;

```

Figure B-4. Linear regression program

```

for(i=0;i<n;i++) {
    xavg+=xi[i];
    yavg+=yi[i];
}
xavg=xavg/n;
yavg=yavg/n;

//Covariance Calculation
cov=0;
for(i=0;i<n;i++) {
    cov+=(xi[i]*yi[i]);
}
cov=cov-(n*xavg*yavg);
cov=cov/n;

//Variance Calculation
varx=0;
vary=0;
for(i=0;i<n;i++) {
    varx+=(xi[i]*xi[i]);
    vary+=(yi[i]*yi[i]);
}
varx=varx-(n*xavg*xavg);
varx=varx/n;
vary=vary-(n*yavg*yavg);
vary=vary/n;

b=cov/varx;
a=yavg-(b*xavg);
rsq=(cov*cov)/(varx*vary);

cout << "a=" << setw(20) << setprecision(15) << a << endl;
cout << "b=" << setw(20) << setprecision(15) << b << endl;
cout << "r^2=" << setw(20) << setprecision(15) << rsq << endl;

xint=-a/b;
cout << "x intercept=" << setw(20) << setprecision(15) << xint << endl;
for(i=0;i<n;i++) {
    cout << "y(xi=" << xi[i] << ")=" << setw(20) << setprecision(15) << (a+(b*xi[i])) << endl;
}
return 0;
}

```

Figure B-4. Continued

- The program in Figure B-5 reads in a 64 x 64 x 64 flux distribution and a 128 x 128 x 128 flux distribution and calculates a maximum absolute relative difference.

```

PROGRAM compareFlux
  IMPLICIT NONE

  REAL(4) :: tal64(64,64,64), tal128(128,128,128)
  REAL(4) :: act64(64,64,64), act128(128,128,128)
  INTEGER :: i, j, k, IOS, l, m, n, cnt
  REAL(4) :: dummy, max64, max128, absDiff, maxDiff
  CHARACTER :: skipLine

  OPEN(1,file='hrtk1_64.flx',IOSTAT=IOS)
  WRITE(*,*) 'File status: ', IOS
  READ(1,*) skipLine

  DO k=1,64
    DO j=1,64
      DO i=1,64
        READ(1,*) dummy, dummy, dummy, dummy, dummy, tal64(i,j,k)
      END DO
    END DO
  END DO
  CLOSE(1)

  OPEN(1,file='hrtk1_128.flx', IOSTAT=IOS)
  WRITE(*,*) 'File status: ', IOS
  READ(1,*) skipLine
  DO k=1,128
    DO j=1,128
      DO i=1,128
        READ(1,*) dummy, dummy, dummy, dummy, dummy, tal128(i,j,k)
      END DO
    END DO
  END DO
  CLOSE(1)

  max64=maxval(tal64)
  max128=maxval(tal128)
  tal64(:,:,:) = tal64(:,::)/max64
  tal128(:,:,:) = tal128(:,::)/max128

  OPEN(1,file='hrtk_fix_act_1.bin',FORM='binary',IOSTAT=IOS)
  WRITE(*,*) 'Activity Status: ', IOS
  READ(1) act64
  CLOSE(1)

  OPEN(1,file='hrtk128_act_1.bin',FORM='binary',IOSTAT=IOS)
  WRITE(*,*) 'Activity Status: ', IOS
  READ(1) act128
  CLOSE(1)

```

Figure B-5. Program to compare flux distributions of two different mesh sizes

```

maxDiff=0.0
DO k=32,32 !1,64
  DO j=32,32 !1,64
    DO i=1,64
      cnt=0
      absDiff=0.0
      DO n=(2*k)-1,(2*k)
        DO m=(2*j)-1,(2*j)
          DO l=(2*i)-1,(2*i)
            IF(act64(i,j,k)==75.0.AND.act128(l,m,n)==75.00) THEN
              cnt=cnt+1
              absDiff=absDiff+abs((tal64(i,j,k)-tal128(l,m,n))/act128(l,m,n))
            END IF
          END DO
        END DO
      END DO
      IF(cnt>0) absDiff=absDiff/cnt
      IF(absDiff>maxDiff) THEN
        maxDiff=absDiff
      END IF
    END DO
  END DO
END DO

```

END PROGRAM compareFlux

Figure B-5. Continued

LIST OF REFERENCES

1. M. H. KALOS and P. A. WHITLOCK, *Monte Carlo Methods Volume 1: Basics*, John Wiley and Sons, New York (1986).
2. E. E. LEWIS and W. F. MILLER, *Computational Methods of Neutron Transport*, John Wiley & Sons, New York (1984).
3. C. YI and A. HAGHIGHAT, "Hybrid S_N and Ray-Tracing with Fictitious Quadrature for Simulation of SPECT," *Proc. Int. Mtg. Mathematical Methods for Nuclear Applications (M&C 2009)*, Saratoga Springs, New York, May 2009, American Nuclear Society (2009) (CD-ROM).
4. W. P. SEGARS, "Development and application of the new dynamic NURBS-based cardia-torso (NCAT) phantom," PhD Thesis, University of North Carolina (2001).
5. C. YI and A. HAGHIGHAT, "A Three-Dimensional Block-Oriented Hybrid Discrete Ordinates and Characteristics Method," *Nucl. Sci. Eng.*, **164**, 221 (2010).
6. C. YI, "Hybrid Discrete Ordinates and Characteristics Method for Solving the Linear Boltzmann Equation," PhD Thesis, University of Florida (2007).
7. G. I. BELL and S. GLASSTONE, *Nuclear Reactor Theory*, Robert E. Krieger Publishing, Malabar, FL (1985).
8. B. G. PETROVIC and A. HAGHIGHAT, "Analysis of Inherent Oscillations in Multidimensional S_N Solutions of the Neutron Transport Equation," *Nucl. Sci. Eng.*, **124**, 31 (1996).
9. B. G. PETROVIC and A. HAGHIGHAT, "A New Directional θ -Weighted S_N Differencing Scheme," *Trans. Am. Nucl. Soc.*, **73**, 195 (1995).
10. L. J. LORENCE, J. E. MOREL, and G. D. VALDEZ, "User's Guide to CEPXS/ONELD: A One-Dimensional Coupled Electron-Photon Discrete Ordinates Code Package," Sandia National Laboratory (1989).
11. ICRP, "Basic anatomical and physiological data for use in radiological protection: reference values. ICRP Publication 89," *Annals of the ICRP* (2001).
12. S. R. CHERRY, J. A. SORENSON, and M. E. PHELPS, *Physics in Nuclear Medicine*, Saunders, Philadelphia, PA (2003).
13. G. LONGONI and A. HAGHIGHAT, "Development of New Quadrature Sets with the Ordinate Splitting Technique," *Proc. Int. Mtg. Mathematical Methods for Nuclear Applications (M&C 2001)*, Salt Lake City, Utah, September 2001, American Nuclear Society (2001) (CD-ROM).

14. M. LJUNGBERG, S. STRAND, and M. A. KING, "The SIMIND Monte Carlo program: A Monte Carlo Calculation in Nuclear Medicine," *Applications in Diagnostic Imaging*, **11**, 145 (1998).
15. X-5 MONTE CARLO TEAM, "MCNP-A General Monte Carlo for Neutron and Photon Transport, Version 5," Los Alamos National Laboratory (2003).

BIOGRAPHICAL SKETCH

Katherine Royston was born in West Palm Beach, Florida in 1984. She graduated from Wellington High School in 2002 and started her undergraduate career at the University of Florida in the fall of that same year. In the summer of 2005, she completed the Research Experiences for Undergraduates Program at the University of Notre Dame. She completed her Bachelor of Science in physics with a minor in business administration at the University of Florida in May of 2006. She then worked for a University of Florida startup company, Viewray, until August of 2007 when she was accepted to graduate school at the University of Florida.