(1)

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GNE/ENP/91M-1

1

GREEN'S FUNCTION APPROACH TO THE ATMOSPHERIC ALBEDO NEUTRON TRANSPORT PROBLEM

THESIS

Donald R. Culp Jr, Captain, USAF

AFIT/GNE/ENP/91M-1

AFIT/GNE/ENP/91M-1

GREEN'S FUNCTION APPROACH TO THE ATMOSPHERIC ALBEDO NEUTRON TRANSPORT PROBLEM

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Nuclear Engineering

Donald R. Culp Jr., B.N.E.

Captain, USAF

March 1991

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Preface

The purpose of this study was to investigate the atmospheric albedo neutron phenomena with the goal of generating a quick-running computer algorithm for the estimation of the albedo free field flux at any point above the atmosphere. This study engaged both analytic development and computational Monte Carlo simulation in the construction of the computer algorithm.

The Green's Function approach to modeling the neutron transport process involved approximating each energy bin of the source spectrum as a Dirac pulse in energy and then summing the contribution from each source bin via the principle of superposition. The computer algorithm spawned from this approach seems to give promising results in terms of run time and the behavior or trends in the data. The accuracy of the computer algorithm is greatly affected by the fineness of the energy bins involved. Recommendations are provided for some improvements in the algorithm.

In the accomplishment of this thesis I have had a great deal of help from my faculty advisor, Lieutenant Commander Kirk A. Mathews, who provided insightful assistance and direction to keep this project on track. In addition, I received help from classmates David Monti and Thomas Lutton during the course of the project. I also thank my wife Denise for her patience and support during this thesis effort as well as during the entire AFIT program.

Donald R. Culp, Jr.

Table of Contents

Table of Figures

Figure Page

Abstract

This study investigated the reflection of neutron radiation off of the earth's upper atmosphere, with the goal of generating a quick-running computer algorithm for the estimation of the albedo free field flux at any point above the atmosphere. This thesis involved analytic development in the construction of the computer algorithm and employed computational Monte Carlo simulation for generating data to be used by the algorithm.

The Green's function approach to modeling the neutron transport process involved approximating each energy bin of the source spectrum as a Dirac pulse in energy and then summing the contribution from each source bin via the principle of superposition. Monte Carlo simulation was used to produce the energy and angle distributions of the albedo flux. The computer program integrates over the surface of the atmosphere and uses the Monte Carlo data along with vacuous transport and a time/energy distribution transformation. The code employs a simplified handling of the neutron/atmosphere interface and uses a predictor-corrector sequence to establish the altitude of reflection. A generic source spectrum was used to evaluate this approach.

The computer algorithm spawned from this approach seems to give promising results in terms of run time and the behavior or trends in the data. Run time was a maximum of six minutes for a flux calculation, with minimum run times of one minute, but a gain on the order of one thousand should be achieved on mainframe computer systems. The albedo flux from an instantaneous point source rises quickly to a maximum, usually within a tenth of a second, and then falls off over time, frequently ten seconds or more. Albedo neutron fluxes as much as 10^{-16} (neutrons/square cm sec/source neutron) have been calculated.

The accuracy of the computer algorithm is greatly affected by the fineness of the energy bins involved. An energy spectrum for the albedo flux which is based upon uniform velocity bins would be best. Recommendations are provided for some improvements in the algorithm, especially in accounting for the buildup of albedo flux in each albedo energy bin over time.

GREEN'S FUNCTION APPROACH TO THE ATMOSPHERIC ALBEDO NEUTRON TRANSPORT PROBLEM

1 Introduction

Most exoatmospheric probes and satellites are equipped with sensitive sensor systems which can be damaged or degraded by nuclear radiation of excessive magnitude. Radiation sources can be either natural or man-made, such as an orbiting nuclear reactor. Radiation shielding for the sensors can incur significant weight to the satellite and therefore must be kept to a minimum.

Knowledge of the radiation environment to which the sensor system will be subjected is critical for the design of the sensor and its required radiation shielding. Given a known source and location, it is straight forward to calculate the direct-flight radiation flux incident on the satellite. However, there may be a significant flux of radiation reflected off of the earth's upper atmosphere. This type of reflected radiation is known as atmospheric albedo radiation.

The goal of this thesis was to investigate the atmospheric albedo phenomena in such a way as to generate a quick running computer routine which estimates the free-field neutron albedo flux at a location above the atmosphere from a known source, source location and time. This computer code implements a Green's function methodology in hope of minimizing run time, which is important as this routine may be integrated with other transport codes to produce a comprehensive computer model of the total space nuclear environment.

Section Two summarizes the neutron transport problem. Section Three discusses various aspects to the approach to this problem. Section Four presents the Monte Carlo simulation accomplished for this thesis.

Section Five introduces the computer program generated for this thesis and traces its development. Section Six presents the albedo neutron flux calculations from the albedo code.

2 Problem Description

The geometry of the transport problem, as presented in Figure 2-1, consists of a neutron source above the "sensible" atmosphere, above which interaction probabilities are negligible. Assuming isotropic emission, neutrons will travel via vacuous transport to the atmosphere and impact at all points on the "surface" illuminated by the source (i.e. within the points of tangency). The neutrons will interact with the atmospheric constituents until a fraction of the incident neutrons emerge from the top of the atmosphere. These reflected neutrons then travel by vacuous transport from the atmosphere, some of which strike the satellite or detector in orbit.

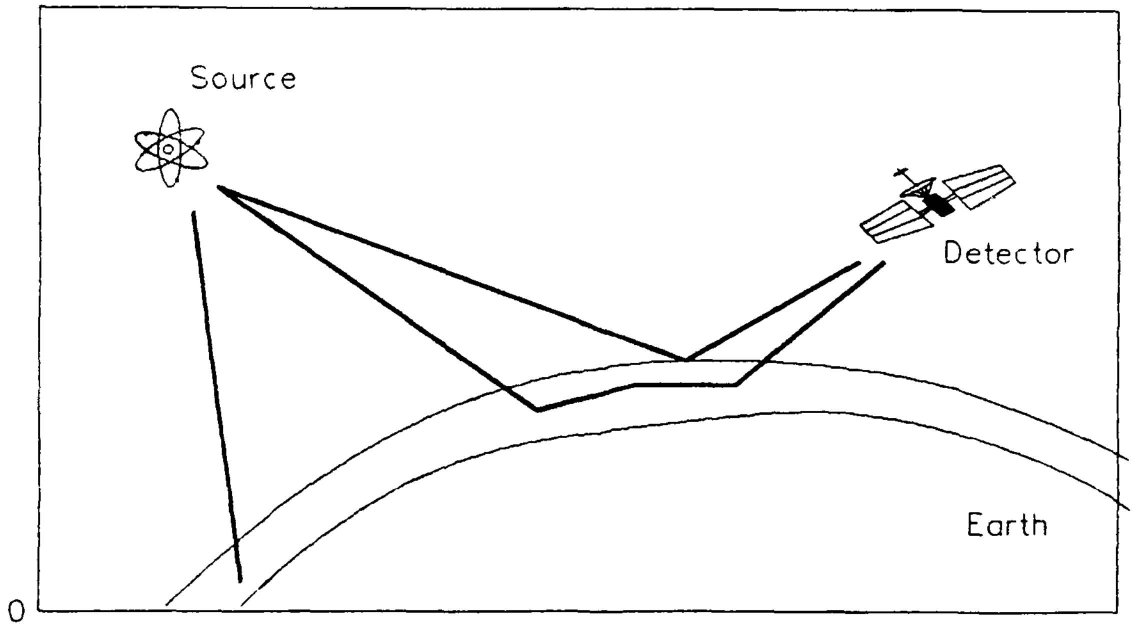


Figure 2-1 Albedo Transport Problem

The depth to which the incident neutrons penetrate the atmosphere depends upon their energy and angle of incidence; high energy neutrons at near normal angles of incidence penetrate deeply compared to low energy neutrons at glancing angles of incidence. The deeper the neutrons penetrate the atmosphere, the more dense the air and the greater number of scattering interactions per neutron, which results in greater absorption probabilities and less likelihood of migrating back to the upper surface. Those deeply penetrating neutrons that do make it back out should be fairly Lambertian in their angular distribution, while the shallowly incident neutrons should exhibit a more biased distribution in both angle and energy.

3 Approach

3.1 Green's Function Methodology

A Green's function is a solution to a differential equation in which the forcing function is a Dirac delta function. A Dirac delta function is a symbolic function which has "zero" duration or width and an amplitude such that its integral is unity. Applied to radiation transport, the Dirac delta function can represent a unit of neutron fluence, while the differential equation represents the physical transport of the neutron fluence through space and time. By regarding any distributed source of neutrons as a series of Dirac pulses, the general transport solution would be the superposition of the Green's Function solution from each Dirac pulse.

Mathematically, let $\mathcal{L}$ denote a differential operator and s a continuous function (1:165-166). Consider the problem of solving for the function f which satisfies the inhomogeneous differential equation

$$\mathcal{L}f(x) = s(x) \qquad 1.$$

and specified boundary conditions. If there exists a unique solution f for each s, there must exist an inverse operator such that for all s the formal solution to equation (1) is

$$f(x) = \mathcal{L}^{-1}s(x) \qquad 2.$$

By definition, the solution of equation (1) corresponding to $s(x)=\delta(x-x')$ is the Green's function, $G(x,x')$, for the operator and given boundary conditions. Therefore, the Green's function satisfies the equation

$$LG(x,x')=\delta(x-x') \qquad 3.$$

or

$$G(x,x')=L^{-1}\delta(x-x') \qquad 4.$$

Equation (2) can be rewritten as

$$f(x)=\int L^{-1}\delta(x-x')s(x')dx' \qquad 5.$$

Employing the definition of the Green's function, one obtains

$$f(x)=\int G(x,x')s(x')dx' \qquad 6.$$

Hence, knowledge of the Green's function for every s allows for the calculation of the solution to equation (1) for every s .

With respect to the neutron transport problem, the above equation takes the form

$$\Phi(\vec{r},E,t)=\int G(\vec{r},\vec{r}\,',E,E',t,t')S(\vec{r}\,',E',t')d\vec{r}\,',dE',dt' \qquad 7.$$

where

Φ – Neutron flux at the detector location $\vec{r}$ at time t

S – Neutron source at location $\vec{r}\,'$ of energy E' at time t'

G – Green's function relating the source to the detector

This study deals only with a single source location and a single detector location, so that the basic phase space parameters which apply to this problem are energy and time. Most sources are distributed in both time and energy, and are usually described by a multiple-bin energy spectrum. In Green's function methodology, each source energy bin may be represented by a Dirac pulse. The source is the sum of a series of Dirac pulses in both time and energy, such as

$$Source(E',t')dE'dt'=\sum S(E')_i\delta(E'-\bar{E}_i)\delta(t'-t) \qquad 8.$$

where

$S(E')$ – fraction of source neutrons within energy bin i

$\delta(E'-\bar{E}_i)$ – energy pulse about mean of bin i

$\delta(t'-t)$ – time pulse centered about emission

Next the source emission is defined to occur near instantaneously, so that the general solution will be the result of a single source pulse in time yet distributed in energy. Thus one can first sum a series of Dirac pulses in energy to get a solution in energy or time since emission, and then sum each of these solutions for each source pulse in time. This thesis concentrates only upon the instantaneous-source problem, leaving the time-distributed source problem as just an extension of this work. The above equation now becomes

$$Source(E',t')dE'dt' = \delta(t'-0)\sum S(E')_i\delta(E'-\overline{E}_i) \qquad 9.$$

This describes a series of pulses of neutrons, each defined by the mean energy of a source energy bin, all originating at time "zero". All time references in this problem are then just the time since emission. Equation (7) now becomes

$$\Phi(\vec{r},E,t) = \int G(\vec{r},\vec{r}',E,E',t)S(\vec{r}',E')d\vec{r}'dE' \qquad 10.$$

$$\Phi(\vec{r},E,t) = \sum_i G(\vec{r},\vec{r}',t,E;S(E')_i)S(\vec{r}',E'_i) \qquad 11.$$

A goal of this thesis is to develop a computer program that generates the above Green's function for every source energy pulse of a known source spectrum, given a source and detector location. If a time-since-emission is specified, the computer program should be able to calculate the albedo flux at the detector for a particular neutron bin energy.

3.2 Transport Processes Modeling

The overall transport process can be represented by

$$\psi(t) = \sum H_i S(E'_i) \qquad 12.$$

where

$\psi(t)$ = flux on detector at time t after emission

H = transport operator on source neutrons

$S(E'_i)$ = source pulse with Dirac symbols suppressed

A large portion of this thesis is devoted to deriving the characteristics of the transport operator so that the computer routine could then be constructed. Instead of starting from the Boltzmann Transport Equation, the approach is to apply "empirical" data from Monte Carlo simulation within a framework of analytic relationships and theory. Individual facets of the neutron transport problem will be separately analyzed and modeled until all of the processes have been addressed. Toward this goal, the parameters and variables applicable to the transport process must be identified.

The transport operator is a function of

$$H = F(Z_s, Z_d, \theta_o, t, E, \theta_i, E', \theta, \phi, \vec{r}) \qquad 13.$$

where

Z_s = altitude of source from the center of the earth

Z_d = altitude of detector from the earth's center

θ_o = earth centered angle of separation

t = time since emission

E' = energy of initial source neutron

θ_i = angle of incidence with the atmosphere

E = energy of the emergent albedo neutron

θ = local elevation angle of albedo neutron

ϕ = local azimuthal angle of albedo neutron

$\vec{r}$ = location along surface of atmosphere

Note that Zs, Zd and θ_o are parameters as they are known values. In addition, the time since emission will be assumed to be a given value. This is because many of the sensitive sensor systems aboard satellites need to operate only at specific points in time, so the albedo flux is only of interest at these times since emission. Figure 3-1 crudely depicts the geometry of the problem.

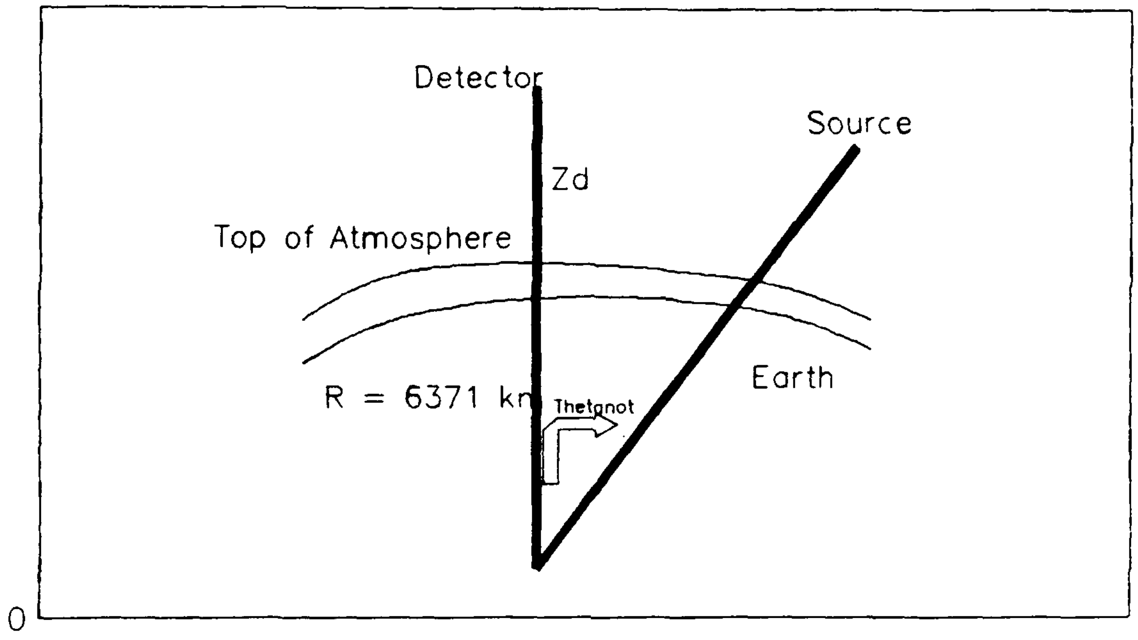


Figure 3-1 Problem Geometry

This problem can be further refined by assuming that there is a definite boundary to the upper atmosphere and that the horizontal distance a neutron travels after entering the atmosphere is small compared to the curvature of the earth. This allows for the treatment of the radiation/earth interface on a localized level and in planer coordinates rather then spherical or polar. This approximation will fail for the case of shallow or glancing angles of incidence because the curvature of the atmosphere means less air is available for interaction than for the planer geometry, but these cases equate to sources and detectors which are far apart and the radiation fluxes are probably attenuated by spherical divergence as to be noncritical.

By a similar argument, the only portion of the atmosphere from which the reflected radiation is important is the atmosphere below the section of upper surface which has been illuminated by the source and is directly visible from the detector. Graphically, this surface area is the common area of the two arced discs cut out from the earth's outer-sphere as defined by the points of tangency from the source and from the detector. The lesser quantity of albedo flux emanating from outside this bounded surface can be attributed to the bounded surface area without significant error.

For the purposes of simplifying the Monte Carlo simulation exercise, a separation of variables is undertaken between the albedo energy and angle dependencies. This takes the current definition

$$H = F(Z_s, Z_d, \theta_o, t, E' \rightarrow E, \theta_i \rightarrow \{\theta, \phi\}) \qquad 14.$$

and divides it into two contributions

$$H = f(\theta_i; E' \rightarrow E) g(E'; \theta_i \rightarrow \{\theta, \phi\}) \qquad 15.$$

The physical location parameters and time are still important but have been suppressed in the above equation. Monte Carlo simulation can now be employed to generate **f** by calculating the energy bin to energy bin probabilities for a given angle of incidence, and can generate **g** by calculating the angle bin distributions for a given angle of incidence and initial energy.

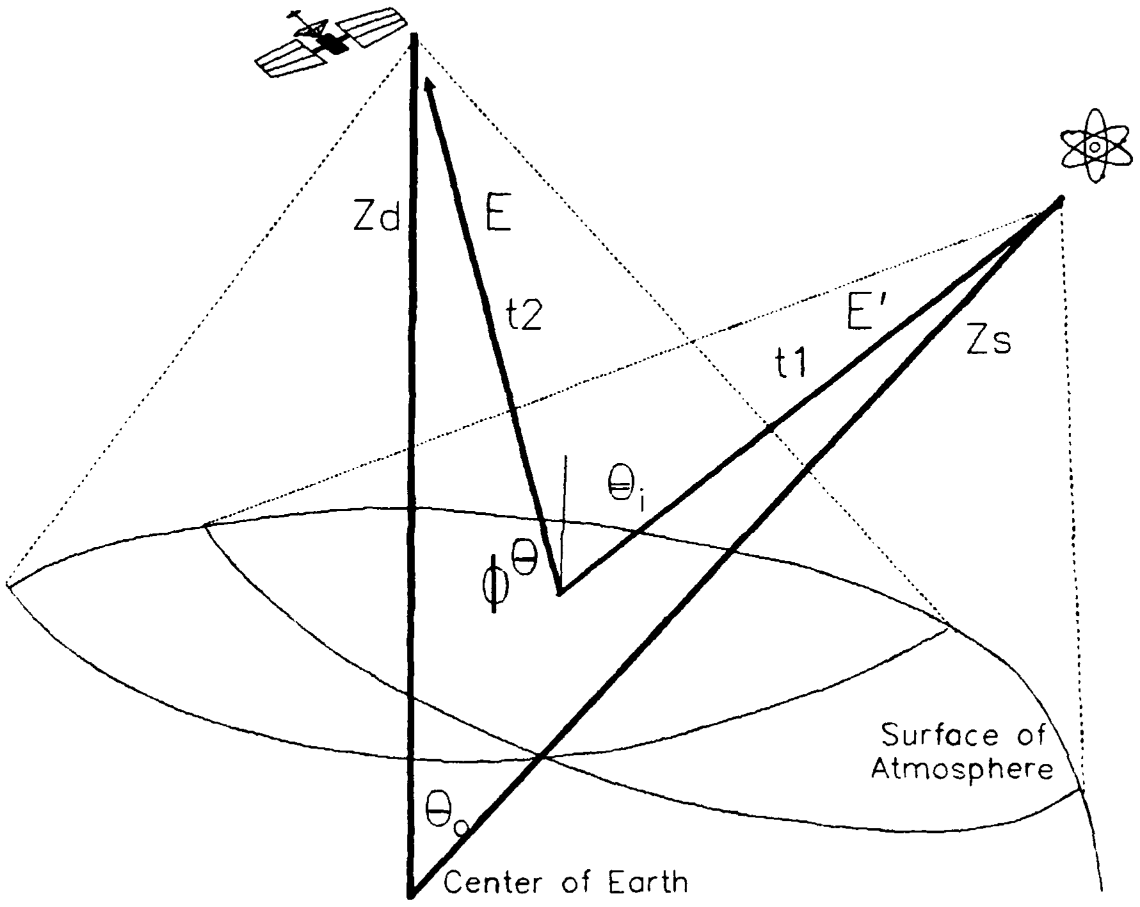


Figure 3-2 Refined Transport Process

Figure 3-2 displays the transport process described thus far. Yet the transport operator is not yet fully defined. The operator must also include the vacuous transport from the source to the atmosphere and then up to the detector; this is accounted for by spherical divergence. The operator must also take into account the energy-time transformation as a source which is distributed only in energy gets spread out over time as well.

3.3 Energy-Time Transform

For the development of this transform, we must take a step back from the Green's Function representation and consider a general case of a source distribution in energy but still instantaneous in time. Let this distribution be defined by a normalized bin spectrum as below.

$$\int_0^\infty S(E')dE' = 1 \qquad 16.$$

The simplest case is of direct vacuous transport from any source location to any another point. Just because the source is only distributed in energy does not mean that the spectrum at the receiving point is only distributed in energy.

Employing classical physics

$$E = \frac{1}{2}mv^2 \tag{17.}$$

The particle distribution in energy is also a distribution in velocity, so the various particles will travel at different rates through the vacuum. This results in a received distribution in both energy and time. What is needed is that transform between energy and time. The relationship between the two distributions is as

$$|S(E)dE| = |S(t)dt| \tag{18.}$$

or

$$S(t) = S(E)\left|\frac{dE}{dt}\right| \tag{19.}$$

This will be accomplished in two parts; first Equation (17) will be used to transform from energy to velocity, and then velocity will be converted to time. The first step is easy.

$$S(v) = S(E)|dE/dv| = S(E)mv \tag{20.}$$

The second step requires a fixed path length and a constant speed along that path, which translates to this case of direct vacuous transport. Then

$$v = r/t \qquad 21.$$

$$S(t) = S(v)\left|dv/dt\right| \qquad 22.$$

$$S(t) = \frac{r}{t^2}S(v) \qquad 23.$$

Combining the two steps yields

$$S(t) = \frac{mvr}{t^2}S(E) \qquad 24.$$

$$S(t) = \frac{mv^2}{t}S(E) = \frac{mr^2}{t^3}S(E) \qquad 25.$$

Unfortunately, the albedo problem is not a direct flight process and the above relation is only applicable for describing the neutron spectrum arriving upon the atmosphere from the source. Refer again to Figure 3-2. The reflected distribution from the atmosphere is now spread over both energy **(E)** and time **(t1)**. For this reflected radiation, it now undergoes vacuous transport and is spread out over yet another time shift **(t1 + t2)**. If the reflection time t_a is comparitively small, then the overall time is **t = t1 + t2**, and

$$|S(E,t)dE\,dt| = |S(E,t_1)dE\,dt_1| \qquad 26.$$

The required relationship is between **t** and **t1**. Since the transport from the atmosphere is direct and vacuous, then

$$t_1 = t - r_2/v_2 = t - r_2\left(\frac{m}{2E}\right)^{1/2} \qquad 27.$$

Therefore

$$\begin{aligned}\frac{dt_1}{dt} &= 1 - r_2\left(\frac{m}{2}\right)^{1/2}\frac{d(1/\sqrt{E})}{dt} \qquad 28.\\ &= 1 - r_2\left(\frac{m}{2}\right)^{1/2}\left(-\frac{1}{2}\right)(E)^{-3/2}\frac{dE}{dt}\\ &= 1 + r_2\sqrt{\frac{m}{8E^3}}\frac{dE}{dt}\end{aligned}$$

Using the chain rule .

$$\frac{dE}{dt} = \frac{dE}{dv_2}\frac{dv_2}{dt} = mv_2\frac{dv_2}{dt} = mv_2\left(-\frac{r_2}{t_2^2}\right)\left(\frac{dt_2}{dt}\right) \qquad 29.$$

Which leads to

$$\begin{aligned}\frac{dt_1}{dt} &= 1 + mr_2v_2\left(-\frac{r_2}{t_2^2}\right)\frac{dt_2}{dt}\sqrt{\frac{m}{8E^3}} \qquad 30.\\ &= 1 - mv_2^3\frac{dt_2}{dt}\sqrt{\frac{m}{8E^3}}\\ &= 1 - v_2\frac{dt_2}{dt}\sqrt{\frac{m}{2E}} = 1 - \frac{dt_2}{dt}\end{aligned}$$

Alas, a purely analytical derivation was not achieved. The changes in transit times t_1 and t_2 with respect to a change in time since emission t are accounted for within the framework of the computer algorithm, and are addressed in Section 5.2.

Combining the two steps, before and after the scatter, with the results from Section 3.2 yields

$$S(E,t)dE\,dt = S(E,t_1)\left(\frac{dt_1}{dt}\right)dE\,dt \qquad 31.$$

or more specifically

$$S(E,t)dE\,dt = S(E')\left(\frac{mr_1^2}{t_1^3}\right)\left(\frac{dt_1}{dt}\right)\times \qquad 32.$$

$$f(E \rightarrow E)g(\theta_i \rightarrow \{\theta,\phi\})dE\,dt$$

Note that $f(E' \rightarrow E)$ is a distribution function with respect to the scattered energy E. The above equation is the basis for the computer routine discussed in Section Five.

3.4 Source Spectrum

The Green's Function methodology and approach to the problem described thus far are applicable for any source spectrum. For the purpose of demonstrating the range and utility of this approach, the source spectrum to be used is a general-purpose nonuniform spectrum which runs from high energy (14.9 MeV) to low energy (10 eV) neutrons. This spectrum is depicted in Figures 3-3 and 3-4 and is comprised of 35 energy bins. Appendix A contains the spectral bin limits and values.

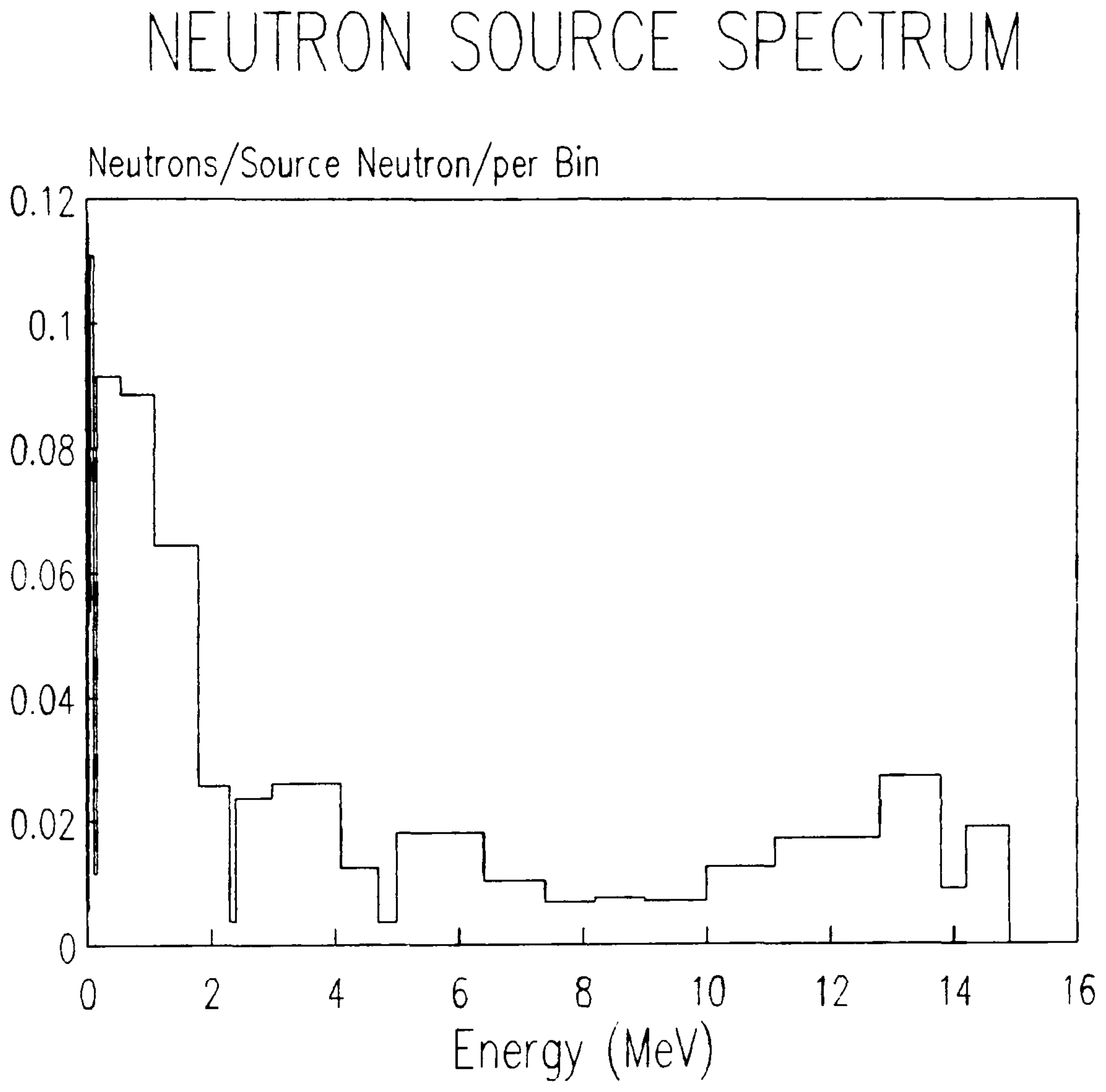


Figure 3-3 Source Spectrum

High Energy Representation

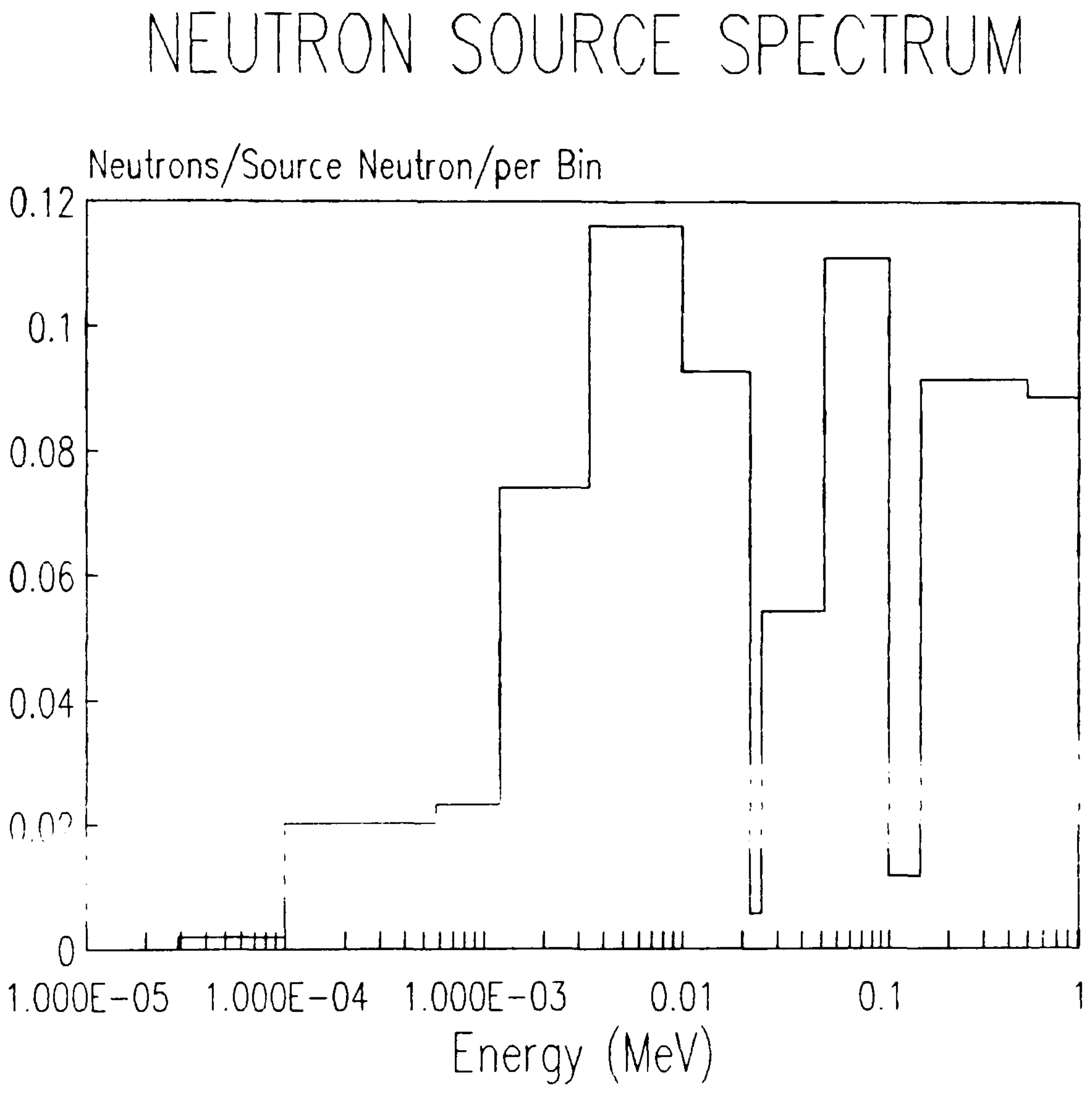


Figure 3-4 Source Spectrum

Low Energy Representation

4 Monte Carlo Simulation and Results

As stated before, Monte Carlo simulation was used to generate the energy bin-to-bin probabilities and albedo angle dependencies which comprise the functions "**f**" and "**g**". Prior to starting this exercise, a few working limits and parameters need to be established.

4.1 MCNP

The Monte Carlo program used for this project is called MCNP, which stands for *Monte Carlo Neutron Photon* and is a product of Los Alamos National Laboratory (2). MCNP is a general-purpose, generalized-geometry, time-dependent, coupled neutron/photon Monte Carlo transport production code. Only a fraction of the capabilities of this code were used as this thesis involves neutron transport calculations only. This code is presently operating at AFIT, though its availability is less than noteworthy due to computer reliability problems.

4.2 Defining the Atmosphere

4.2.1 Top of the Atmosphere

Vacuous transport of neutrons is assumed above the atmosphere. The density of the earth's atmosphere falls off roughly exponentially with altitude, but there is no physical or mathematical boundary between air and space. Defining such a boundary can be somewhat subjective, but a parametric analysis has been performed (6) to determine an effective cut-off value with altitude. This was done by determining the integral mass density between a given altitude and 2000 km, which is essentially infinity. The top of the atmosphere was found by relating the integral mass densities to neutron mean free paths. The albedo surface was set at 70 km because the mean free path for most energies is about equal to the integral mass density from 70 km to 2000 km. In other words, there is one mean free path or less of atmosphere above 70 kilometers. This boundary is used as the starting point for many thesis calculations with the view that the MCNP results (reflected energy and angle dependencies) should not be significantly affected by the small uncertainty in this artificial boundary.

4.2.2 Atmospheric Makeup

The primary atmospheric constituents below 70 kilometers altitude are $N_2, O_2, Ar, H_2O, CO_2,$ and CO. But since the last three are only significant below 20 km and few incident neutrons get that deep and return, the air was modeled only with the first three elements. The contributions from each (by volume) are about 78% nitrogen, 21% oxygen and 1% argon. Cross sections used were from ENDF/B data which had been processed by Los Alamos into "discrete reaction" cross sections, which are histographic, highly accurate representations of the actual variations of cross section with energy.

The other aspect to the atmosphere that must be approximated is its density variation with altitude. Since MCNP does not allow for density variations within its geometric blocks or cells (as most programs do not), the approach was to rely upon the common technique of modeling the atmosphere with multiple layers of air, each with its own density. Another common practice for approximating the density profile of the atmosphere is to match these artificial layers of air with regions of the atmosphere where the density falls off with a constant scaling factor. Based upon the U.S. Standard Atmosphere (1976), eight layers of air with eight separate scale heights were used to model the atmosphere. The table below contains the quantitative description of the model;

Table 4.1 Atmospheric Parameters for Constant Scale Heights

Region	ALTlow	ALThigh	Density	Hs
1	10 km	15 km	4.205E-4	6.56 km
2	15 km	20 km	1.942E-4	6.37 km
3	20 km	25 km	8.903E-5	6.28 km
4	25 km	30 km	4.043E-5	6.37 km
5	30 km	40 km	1.875E-5	6.55 km
6	40 km	50 km	3.917E-6	7.33 km
7	50 km	60 km	7.994E-7	8.39 km
8	60 km	70 km	3.114E-7	7.58 km

The densities given in the table are in grams per cubic centimeter and are derived by the equation;

$$\rho = \rho_0 \exp(-\{z - z_0\}/Hs) \qquad 33.$$

4.2.3 Geometric Configuration

The overall configuration for the MCNP runs is displayed in Figure 4-1. It was constructed in cartesian coordinates in accordance with the planar approximation described earlier. The dimensions of the "sides" were also based on trial and error; of all the neutrons departing this volume, at least 95% of them escaped through the upper surface.

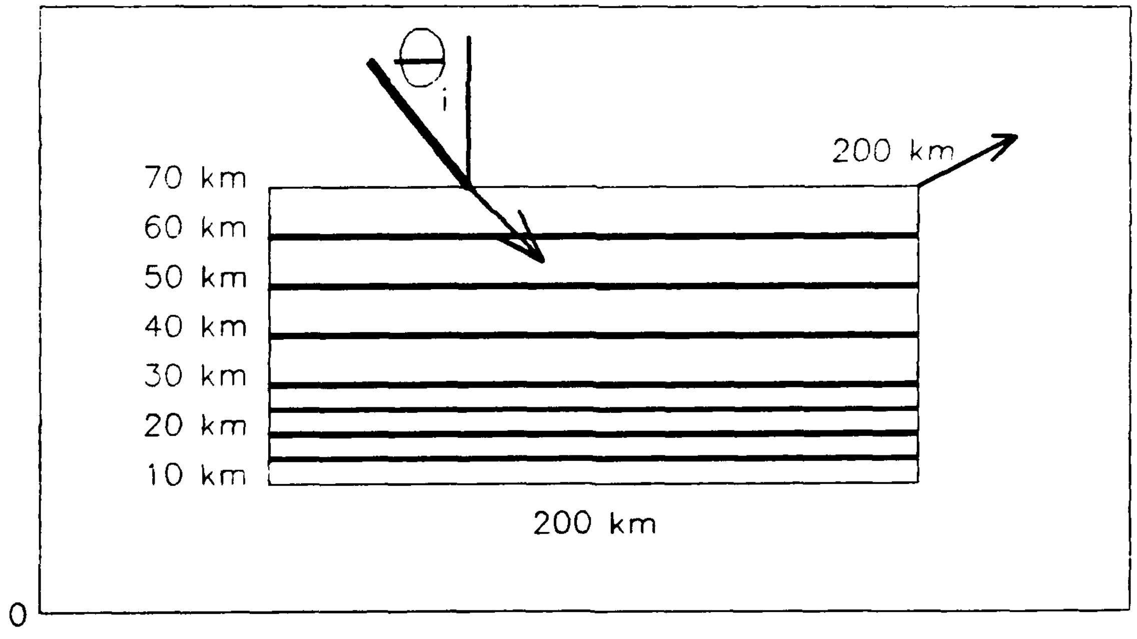


Figure 4-1 Geometry Used in MCNP Simulation

4.2.4 <u>MCNP Results</u>

The method used to generate the data was to take one of the source bins from the general spectrum in Section 3.4 as the neutron source and vary its angle of incidence onto the 70 km surface, each time letting MCNP track the albedo flux from the upper surface. The particles were started at the surface but their velocity was biased as monodirectional and into the atmosphere. MCNP sampled from the source bin and scored the surface current into energy and angle bins. First MCNP was used to get the energy dependence of the emergent flux as a function of angle of incidence, then later to get the angle dependencies as a function of incident and emergent energy.

As expected, the fraction of source neutrons which escaped from the top of the atmosphere is a function of both initial energy and angle of incidence. Figure 4-2 shows the total surface current out of the atmosphere for various incident energies at various angles of incidence. Generally, the lower the initial energy and more shallow the angle, the greater the leakage fraction.

The values for Leakage Fraction were not used in the algorithm directly, but they are inherent to the energy and angle data which were used.

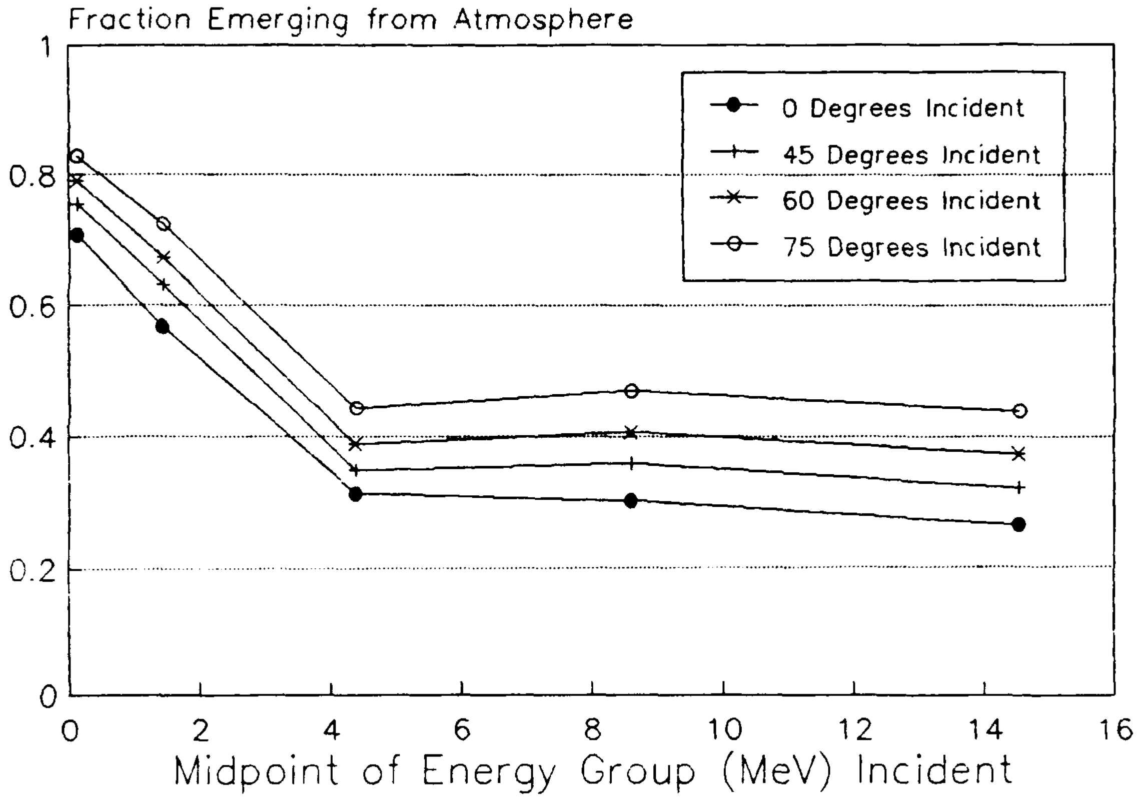


Figure 4-2 Fraction of Incident Neutrons Escaping Out of the Top

To calculate the data for $f(E' \rightarrow E)$, energy bins were set up to score the albedo flux over the same energy limits as the energy bins defining the source spectrum. This way the MCNP output is directly related to the values for $f(E' \rightarrow E)$. In order to reduce the fractional standard deviation below 0.05 for most bins, a batch size of 100,000 particles was run for each combination of incident energy bin and angle. The following figure is the results for various energy groups incident at 45 degrees. Except for the few narrower energy bins, uncertainties in the data are small, therefore error bars are not displayed.

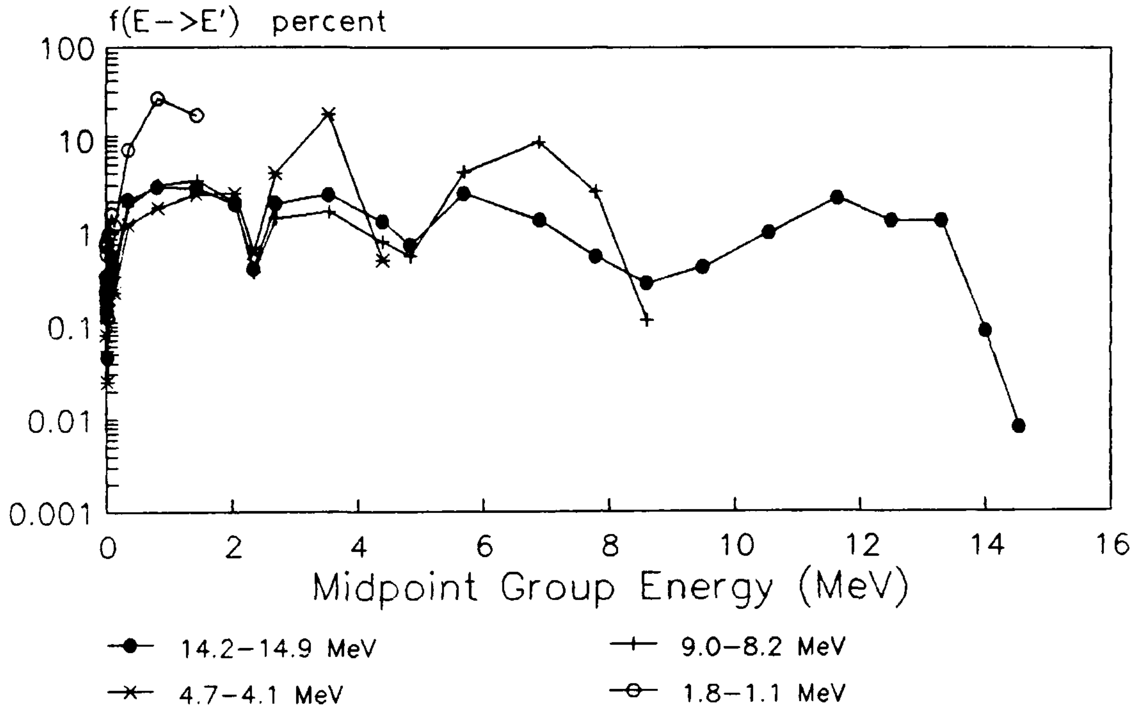


Figure 4-3 Albedo Spectrum

from 45 Degrees Incidence

The albedo bin values were multiplied by 100 to make the graph easier to read, but each energy bin needs to be divided by its width in MeV in order to get the true picture of the reflected spectrum.

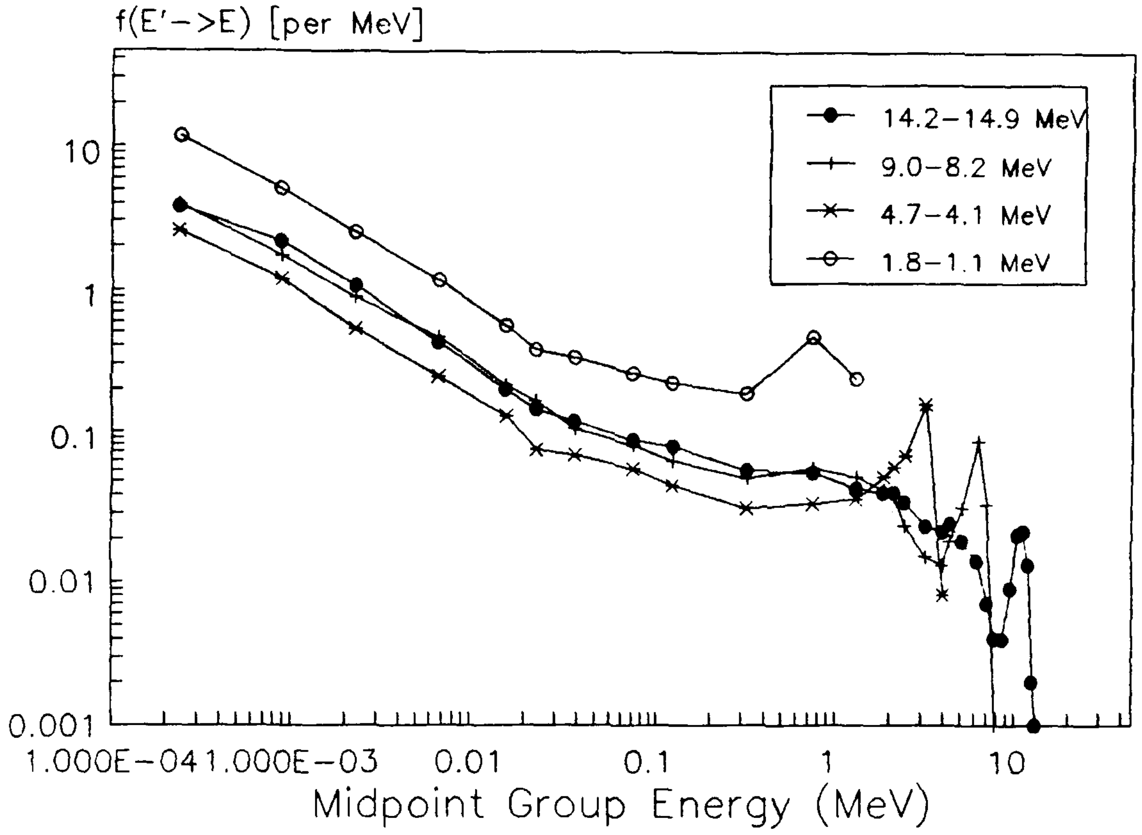


Figure 4-4 Albedo Spectrum per MeV

from 45 Degrees Incidence

The next figure shows the same thing except for an incident angle of zero degrees, or "straight down" into the atmosphere.

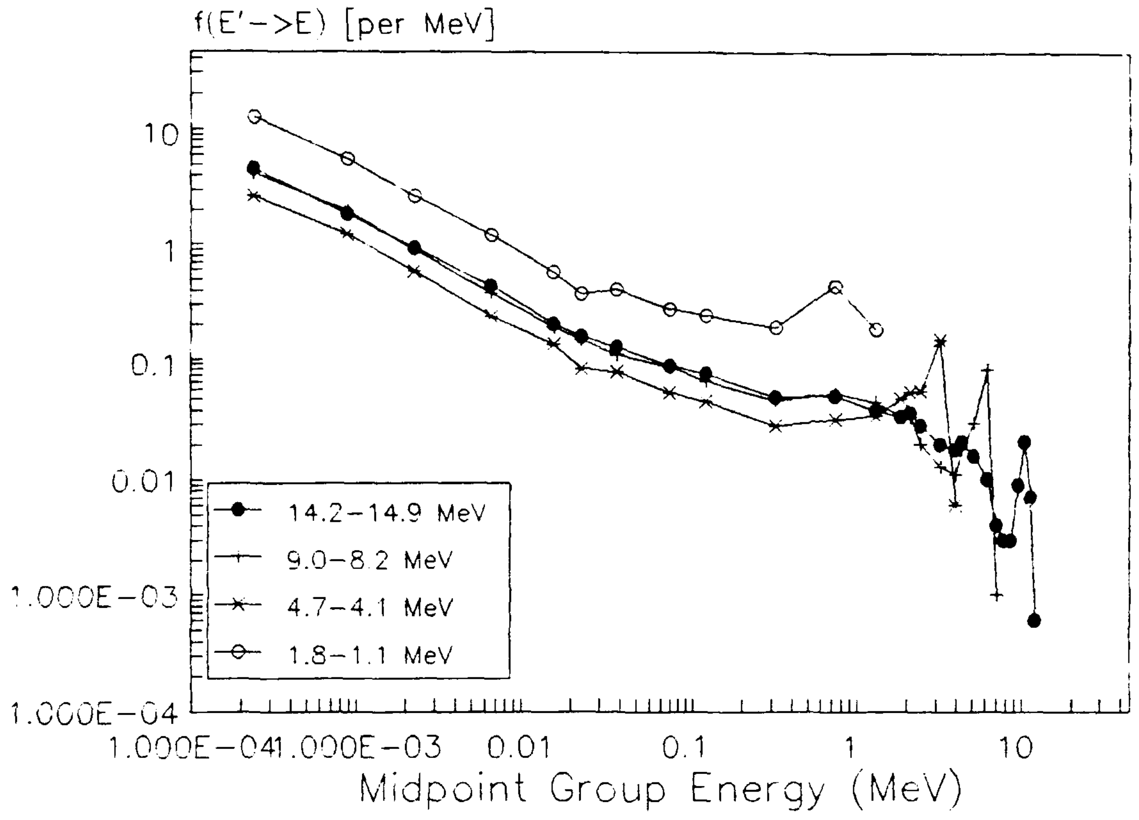


Figure 4-5 Albedo Spectrum from

Zero Degrees Incidence

At first glance there does not seem to be much difference between the zero and 45 degree cases, but there is a slight shift in distribution from the "humps" and to the rest of the spectrum for the zero degree case. Since these "humps" represent the first few scatters in the air and out, this means that there are more higher-energy albedo neutrons from the 45 degrees case then from the zero degree case because the initial neutrons penetrate less and suffer less energy loss. The bulk of the spectrum shift is found in the very-high-energy neutrons (with respect to the initial energy); there are few of these that come out for the deeply penetrating case of zero angle of incidence compared to the 45 degrees case. The difference is on the order of 30 percent. The trend continues for even greater incident angles, but graphically it is still not evident.

To get the data for $g(\theta_0 \rightarrow \theta, \phi)$ also required setting up bins for collecting the angle dependencies. This would have to be done within each outbound energy bin because of the coupled nature of this problem. The coordinate system used is centered upon the point of emanation from the albedo surface, wherever the emergent neutron escapes from. Its angle of elevation from the surface is then scored in the appropriate angle bin and its azimuthal angle is scored in its corresponding angle bin. Zero degrees for both equates to "straight ahead" along the same direction as from the source. Both the elevation angle and azimuthal angle distribution (over the full 180 degrees) were divided up into twenty equal angle bins; each bin width was equal in cosine value of one-tenth. Batches of 200,000 particles were necessary for the angle dependency data in order to keep the fractional standard deviation below 0.05 for most bins.

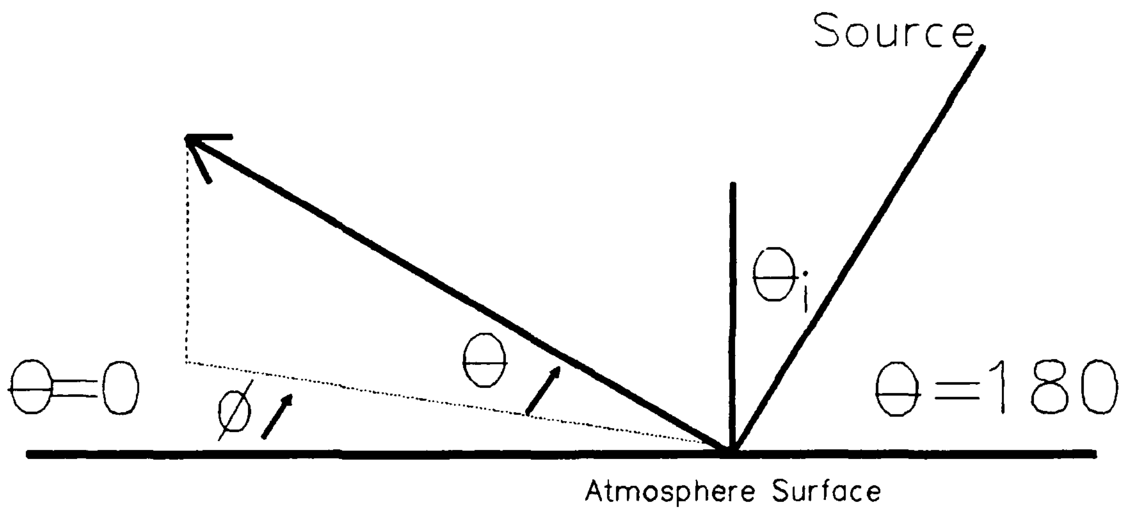


Figure 4-6 Angle Definitions

of Albedo Neutrons

Instead of generating the albedo angle distributions from every input combination of energy and angle and over every emergent energy bin, it was assumed that the albedo flux is Lambertian for all conditions, except for those albedo neutrons in the "hump" of the energy distribution. These were the only ones which should exhibit any angle bias because of the few collisions they suffered before they escaped.

The distribution of the albedo flux in elevation angle depends upon the initial energy but even more strongly upon the incident angle. If source neutrons enter at zero degrees or "straight down" then the albedo distribution is fairly Lambertian or unbiased towards any direction because the initial neutrons have undergone numerous scatters

before finally leaking out. Due to the $\Psi(\Omega)\hat{\Omega}\cdot\hat{n}$ nature of surface current, those neutrons of the Lambertian flux with directions differing the most from the surface normal contribute the least to the surface current, resulting in a cosine shaped distribution over the range -90 to 90 degrees with respect to the surface normal (over 0-180 degrees as defined by the elevation angle). On the other hand, for shallow angles of incidence, there can be a significant fraction of once or twice scattered neutrons which escape. These albedo particles have preserved some of the forward directionality of their initial momentum and are therefore forwardly biased in their distribution from the surface.
The forward bias of the higher energy albedo neutrons is evident in Figure 4-7. However, even within the $f(E'\rightarrow E)$ "hump", the lower energy albedo neutrons exhibit the cosine shaped distribution indicating a Lambertian flux. The 12.2-11.1 MeV group is about in the center of the hump. In fact, for all cases, only those emergent neutrons with energies within about 20% of the initial energy demonstrate any angle bias; all others have suffered enough collisions to cause their behavior to be diffusive-like.

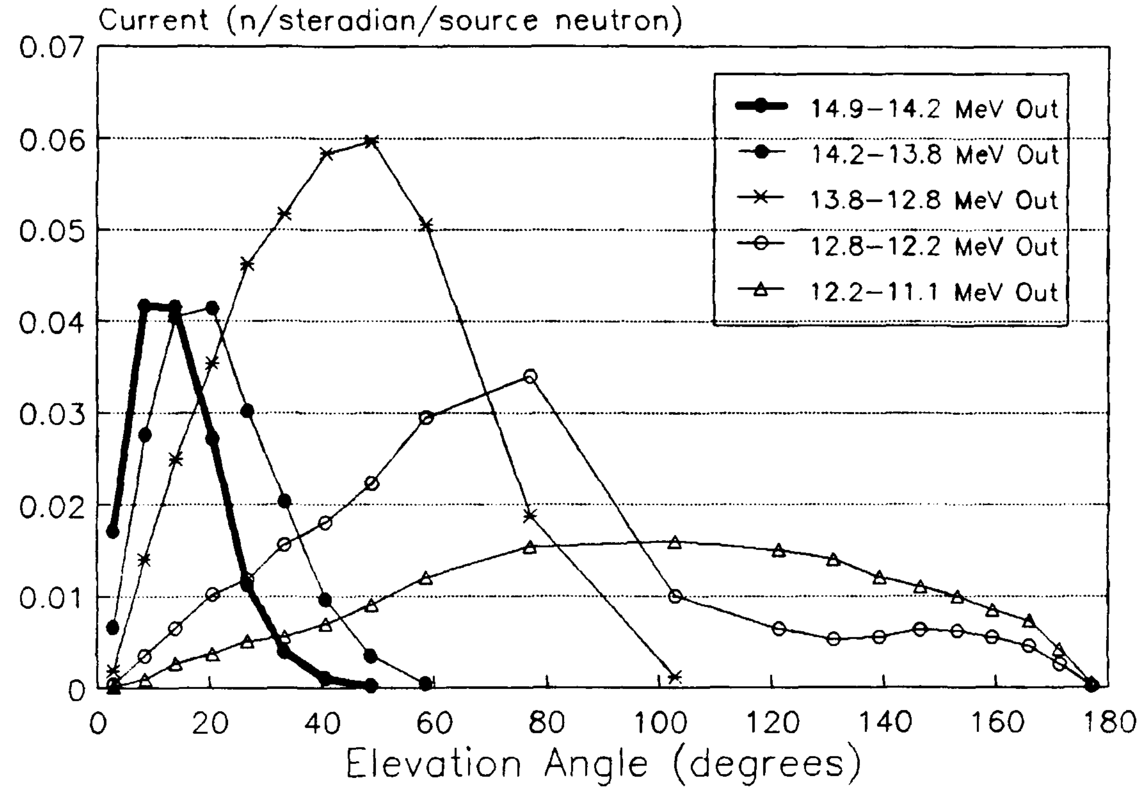


Figure 4-7 Distribution in Elevation Angle for 14.9-14.2 MeV Neutrons Incident at 75 Degrees

As a product of the bin nature of the approach to this transport problem, in many instances the angle bias of the albedo neutrons is masked by the energy bins themselves. Figure 4-8 demonstrates this characteristic. The reason for this is that the 20 percent "threshold" falls within the same energy bin as that of the initial energy. If this energy bin is sufficiently wide, the bias in angle is combined with the cosine contributions from other albedo energies, thereby hiding much of the biased distribution. This problem dominates most middle and lower energy bins of the spectrum because the relative widths of the bins increase.

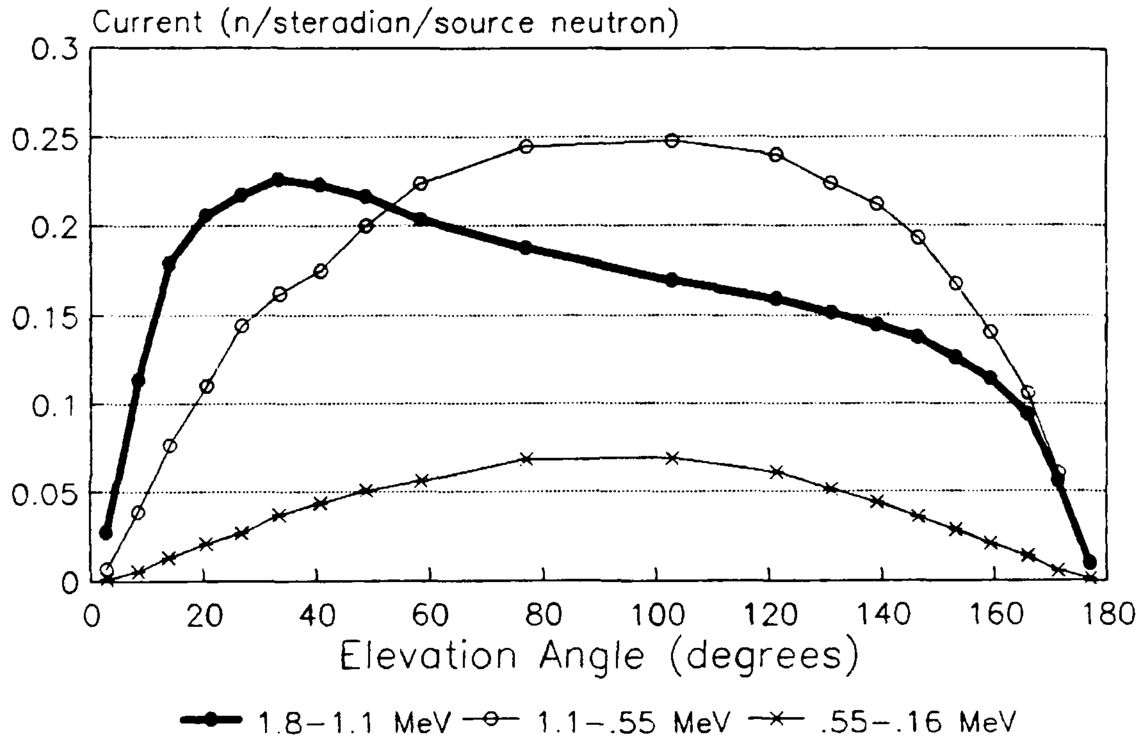


Figure 4-8 Elevation Angle Distribution

for 1.8-1.1 MeV Neutrons Incident at 75 Degrees

As expected, the angle distribution for the sharper angles of incidence are less biased due to the deeper penetration into the atmosphere. Figure 4-9 demonstrates this point.

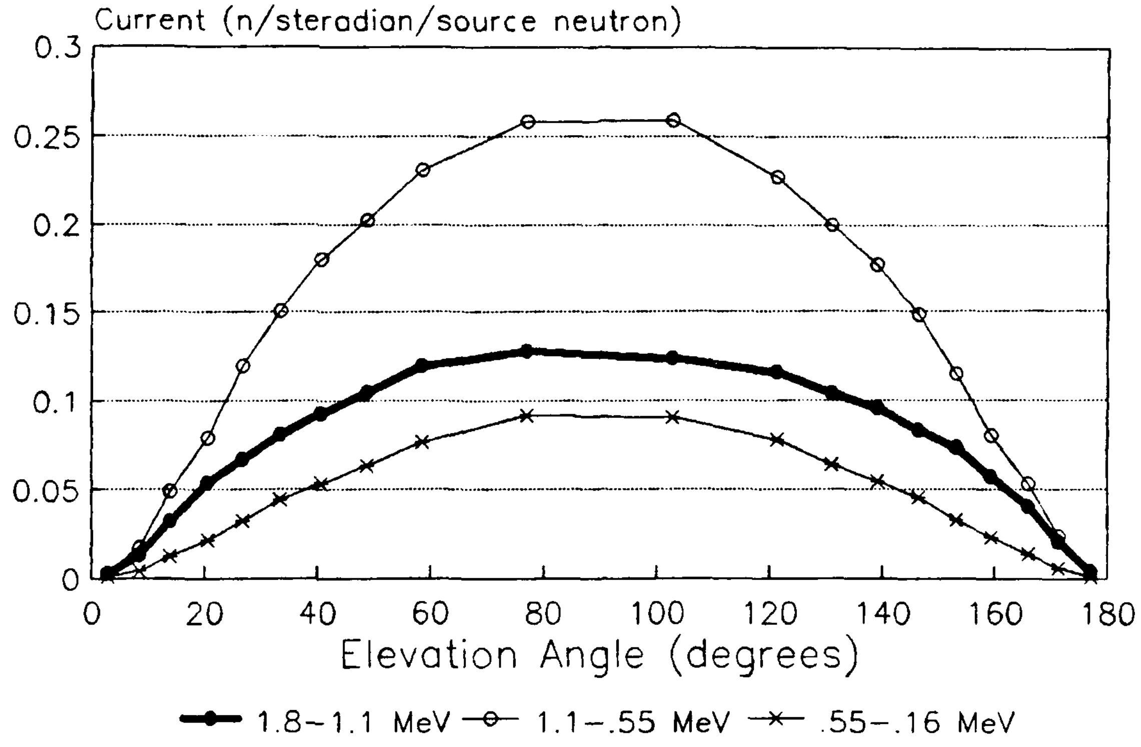


Figure 4-9 Distribution in Elevation Angle

for 1.8-1.1 MeV at 15 Degrees Incidence

Figure 4-10 displays the azimuthal distribution of the albedo neutrons about the point of escape, in which zero degrees pertains to the direction along which the neutrons travelled from the source.

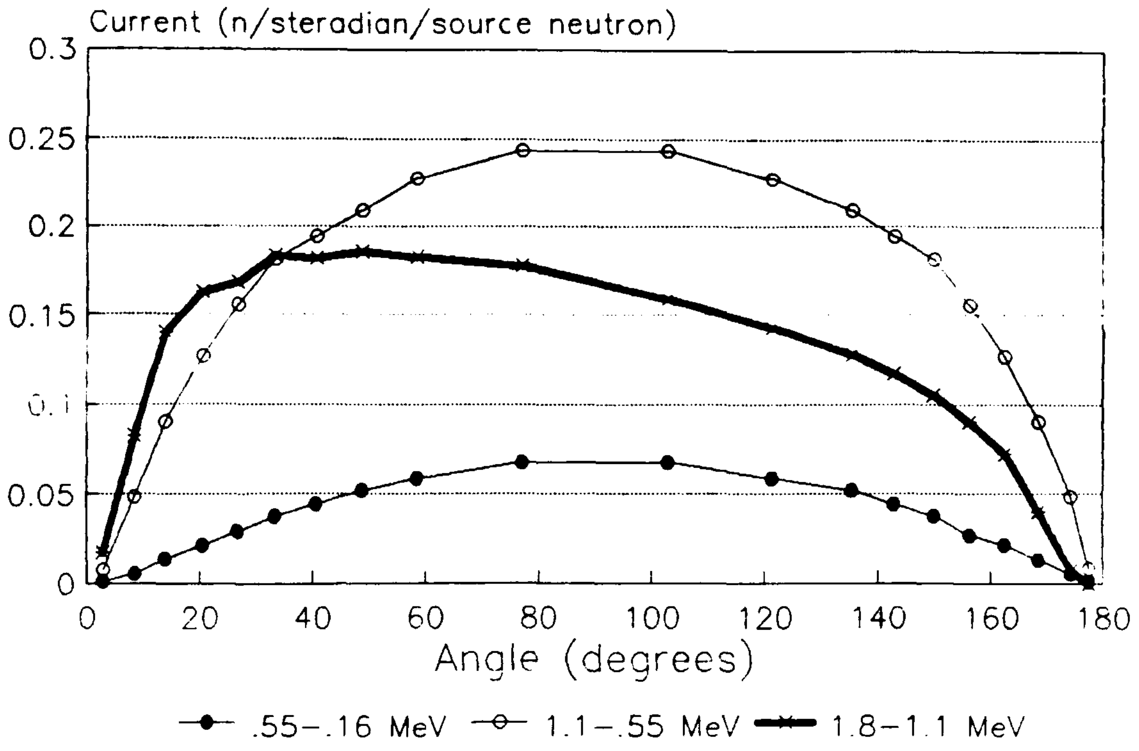


Figure 4-10 Azimuthal Distribution

from 1.8-1.1 MeV at 75 Degrees Incidence

5 Albedo Code Development

The creation of a quick-running computer algorithm for the calculation of the free field albedo neutron flux at a point above the atmosphere can now be initiated. The final form of this code will likely be written in Fortran and run on a mainframe computer, but for practical purposes the personal computer at home was better for working on the coding, debugging and calculations. The computer used was a 16 MHz IBM compatible PC with a 80386SX processor but without an additional math co-processor. The gain in speed from this PC to a Mainframe should ultimately be on the order of one thousand. The programming was in QuickBasic 4.5 because of familiarity with this programming language.

5.1 Reflective Surface Treatment

Thus far the energy and angle distributions of the albedo neutron flux have been attained via Monte Carlo simulation. All that remains is to model the neutron interaction and reflection event during the transport process to account for the time dependencies involved. The fastest working method was to assume that the incident neutrons reflect or "bounce" off of an unyielding layer of air, analogous to difusive scatter. Time-of-flight can be approximately conserved if this reflective surface is set about to the altitude at which many of the incident neutrons would have penetrated. The time it takes for the neutrons to travel from the 70 km top of the atmosphere to the reflective surface and back up to the top approximates the time the neutrons spend bouncing around within the atmosphere before they escape. Figure 5-1 demonstrates this concept.

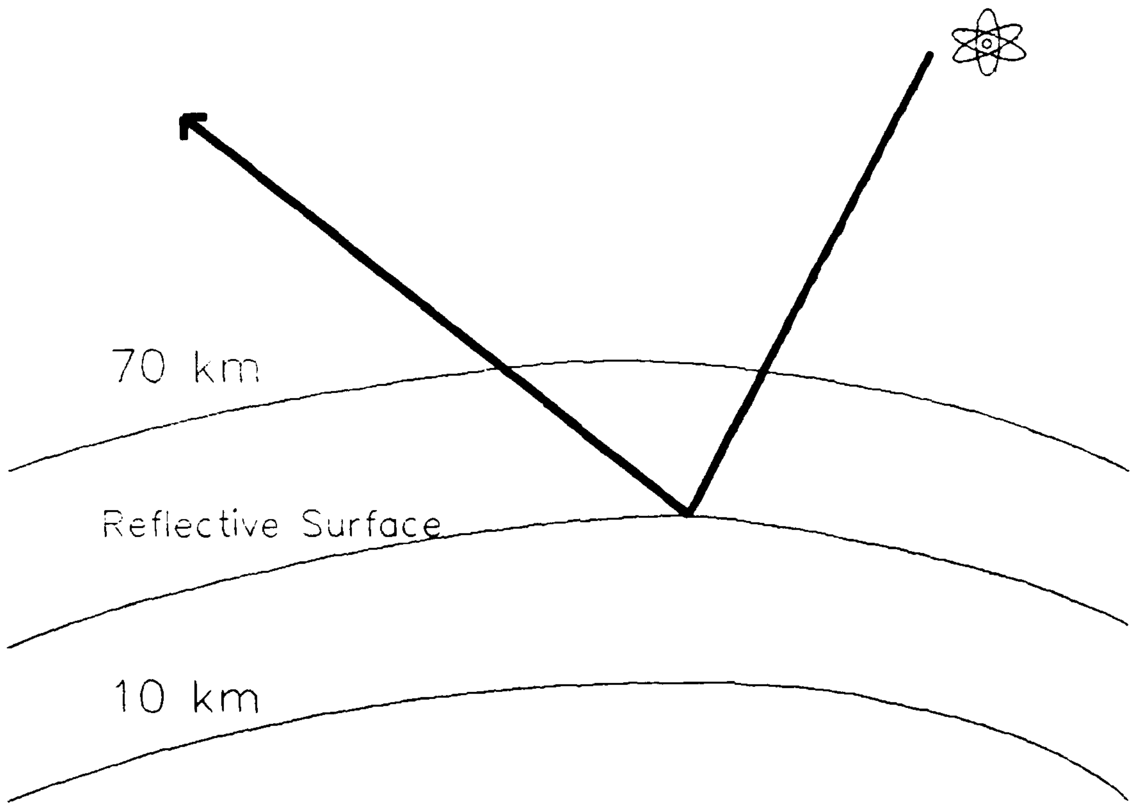


Figure 5-1 Reflective Surface Concept

For the travel times to be roughly equivalent, the altitude for this reflective surface must be linked to mean free path or optical depth of the upper atmosphere for the incident neutrons. The greater the initial energy or sharper the incident angle the lower the reflective surface.

The distance which the incident neutrons travel is related to the mass integral of air. The fraction of neutrons transmitted through a certain mass integral is given by the relation;

$$\text{Fraction Transmitted} = \exp\left(-MI*\frac{Na}{AW}*\sigma(E)\right) \qquad 34.$$

where

MI = Mass Integral of air in grams per square centimeter

Na = Avagadros Number

AW = Atomic Weight of air (14.6 grams per mol)

$\sigma(E)$ = Cross Section of air as a function of energy

For the fraction transmitted the value of one-half is used, which equates to the distance at which half of the incident neutrons have scattered. This will hopefully average out the short time in the atmosphere from the earlier scattered neutrons with the longer time in the atmosphere from the later scattered neutrons. If the cross section is known, then one could calculate the effective mass integral of air needed to attenuate half of the initial neutrons for a given energy. A detailed fit to the air cross section should not be necessary and would also be computationally more expensive. By taking specific values for cross section of nitrogen, oxygen and argon from the Barn Book (3) and ignoring resonances, an approximate dependence upon energy was pieced together and then a simple fit to this data was generated.

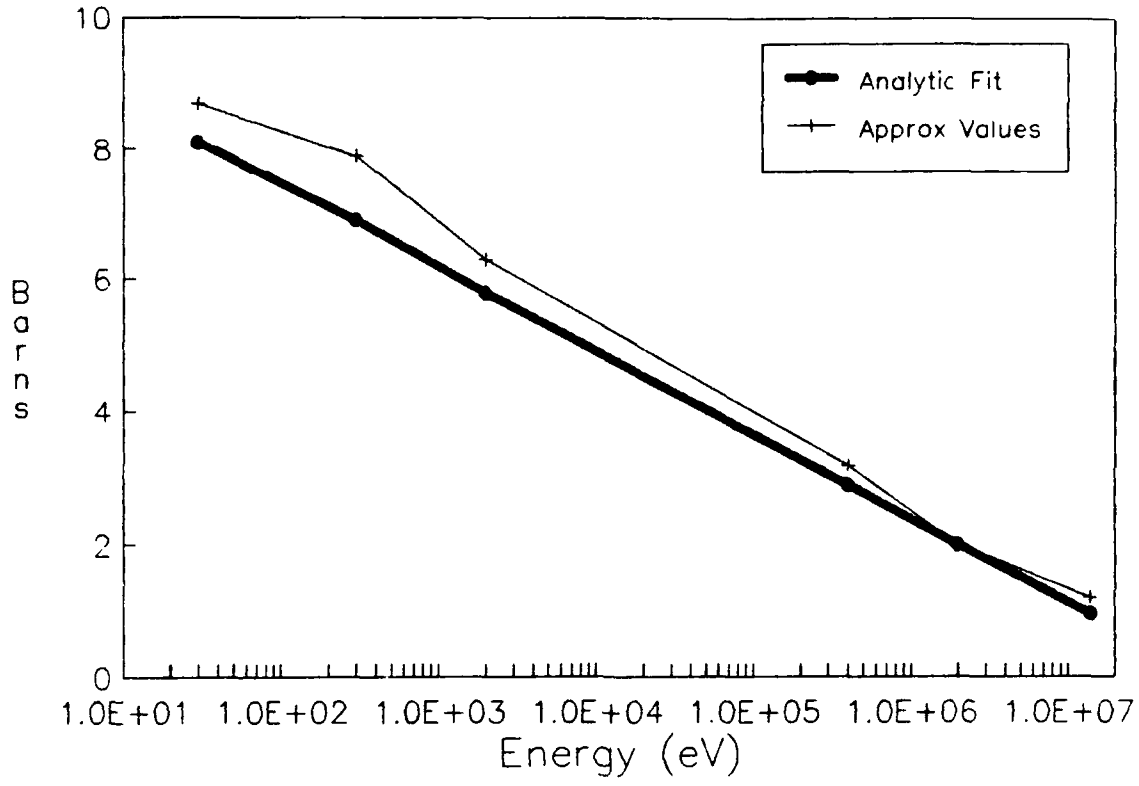


Figure 5-2 Approximation of Air Cross Sections

With a value for the mass integral calculated, one can relate this to the geopotential air pressure and subsequently to a value for the altitude for the reflective surface. A defining relation for the mass integral is given by the following equation (6).

$$MI = \frac{1}{\cos(\theta_i)}\left[\frac{Press(70km)}{g} - \frac{Press(Zeff)}{g}\right] \qquad 35.$$

This equation relates the pressure differential between two altitudes and the angle of travel to come up with the mass integral. Relying upon empirical fits (4) to the U.S. Standard Atmosphere (1976) over regions of constant scale height, one can solve for the altitude corresponding to the pressure which satisfies the above equation.

This capability to determine the altitude for the reflective surface as a function of energy and angle of incidence has been built into the albedo code. The behavior of the reflective surface versus energy (in terms of cross section) and angle of incidence is presented in Figure 5-3.

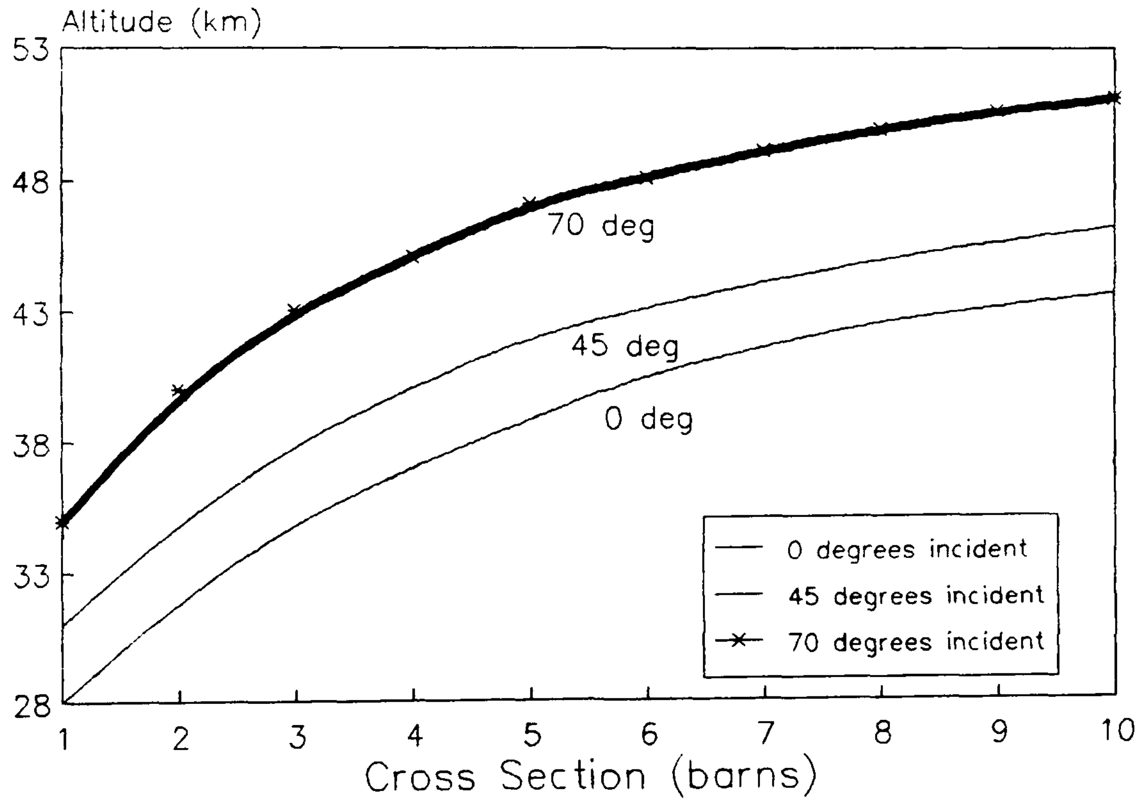


Figure 5-3 Altitudes for the Reflective Surface

5.2 Predictor-Corrector Scheme

The albedo code determines the common surface area on top of the atmosphere which is both illuminated by the source radiation and visible to the detector. The code then divides this common area into smaller wedges of equal area. With the coordinate system centered about the detector's pole, the range of polar angle (along the plane containing the source and detector) is divided into increments; the same with the azimuthal angle. The contribution of albedo flux from each surface wedge is then summed to get the flux onto the detector. The code takes advantage of azimuthal symmetry about the detector pole by calculating one-half and doubling. Figure 5-4 displays an example of this process, where **S** and **D** stand for the points on top of the atmosphere directly beneath the Source and Detector.

In Figure 5-4, theta and omega can be thought of as colatitude and longitude, where **D** is taken as the north pole and **S** is placed on the prime meridian (zero longitude). The arcs of constant theta are concentric circles centered on **D**.

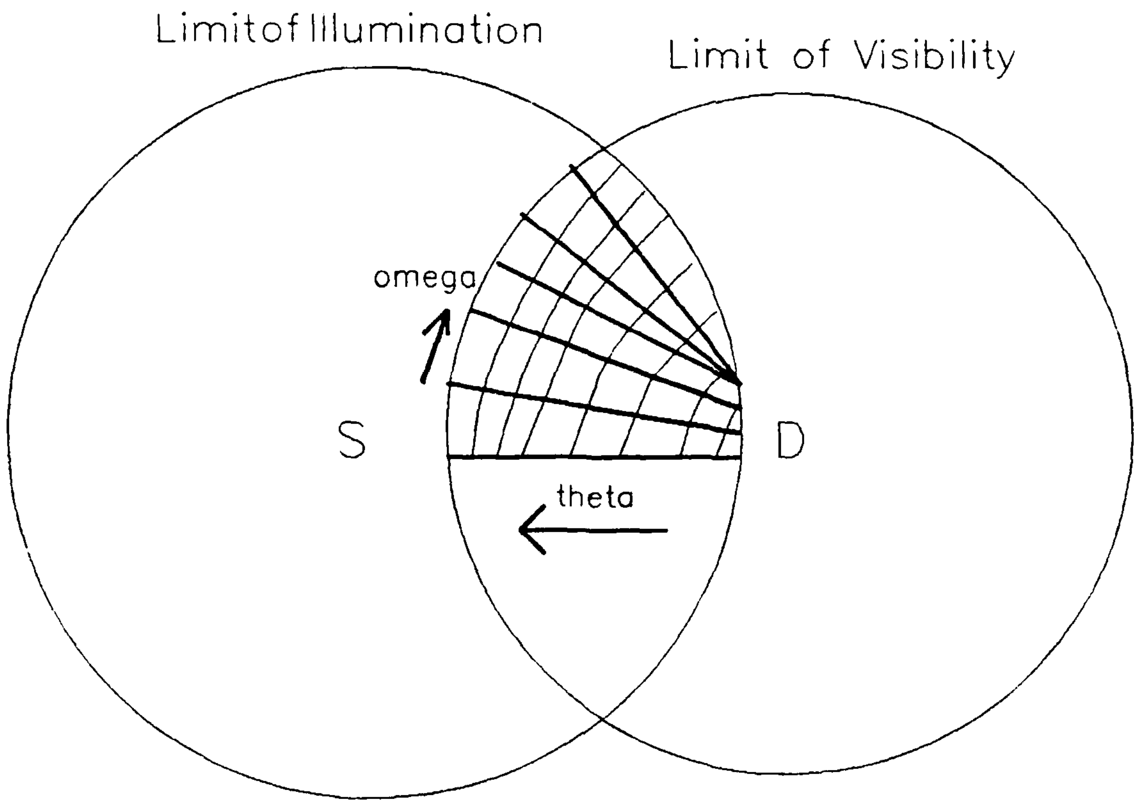


Figure 5-4 Example of Surface Mesh

An input value to the albedo code is the time since emission, which is the same as the time of interest for the flux on the detector. Each source bin is treated as a packet of neutron radiation with a velocity defined by its midpoint energy. If the transport sequence is started from the source and with one of the source energy bins, the energy and angle of incidence onto every area wedge of the common surface can be calculated, as well as the travel time to the surface. The time remaining from the time since emission, along with the pathlength to the detector from the point on the surface, defines the reflected velocity (and energy) of the albedo neutrons.

However, the reflective surface varies in altitude and its location is unknown when this sequence is started. Therefore, it starts with a layer altitude of 70 kilometers (the maximum) to set up the surface mesh of equal areas, then computes the angle of incidence for each one as needed, then resets the reflective surface to its approximate altitde based upon this angle and initial energy. The pathlengths, angle of incidence and travel times are then recomputed to determine the energy of the albedo flux. This procedure is called a predictor-corrector sequence.

The overall process just outlined follows (in an analog manner) the actual transport process. Unfortunately, the coding for this approach was cluttered due to the necessity to keep an active account of the albedo flux in each of its energy bins as it adds bit by bit. It was much easier to attack the transport problem "backwards".

To do this the algorithm starts with an energy bin at the detector and works backwards to find which source bin contributes to its albedo flux, just as before for every unit area and for a certain time since emission. The predictor - corrector scheme is still required because the reflective surface must be started at 70 kilometers. Once it calculates an estimate for the initial energy, the effective height for the reflective surface is determined just as described previously and the backwards transport is initiated again. Only one pass through the predictor-corrector sequence proved necessary as the initial energy values and surface height values changed little after additional passes.

Note that in this "backwards" process the transit time between a surface mesh point and the detector (t_2) is fixed by the detector energy bin value. Therefore, changes in the overall time-since-emission will affect only the calculated transit time from the source to that surface point (t_1). With respect to the final equation in Section 3.3

$$\frac{dt_2}{dt} = 0$$

so that

$$\frac{dt_1}{dt} = 1$$

5.3 <u>Basic Algorithm</u> - <u>Initial Version</u>

The first step of the albedo code is to read in the parameters defining the source and detector locations, the time of interest, and the source spectrum. The code then sets up the problem geometry.

The code constructs a coordinate system centered about the pole defined by the line segment from the earth's center to the detector. The code uses spherical coordinates in constructing the geometry but frequently reverts to cartesian for calculating distances between points. The intersection of the line segment with the top of the atmosphere is given "x" and "y" values of zero.

The next step is to determine the common surface area defined by the intersection of the area illuminated by the source and that which is visible to the detector. The illuminated and visible areas are bounded by the points of tangency on the upper atmosphere. The geometry or shape of the common area has been divided into multiple categories, each one bounded in dimension in a different manner. Appendix B contains diagrams of each of the categories.

This common area is described by maxima and minima in theta and omega (see Figure 5-4). Values for omega are calculated using the cosine rule for spherical triangles. Once the common area is determined, it is separated into increments of theta and omega so as to divide the total area into small wedges of uniform area. Theta is incremented in units of constant $\cos(\theta)$ while omega is incremented in units of constant angle.

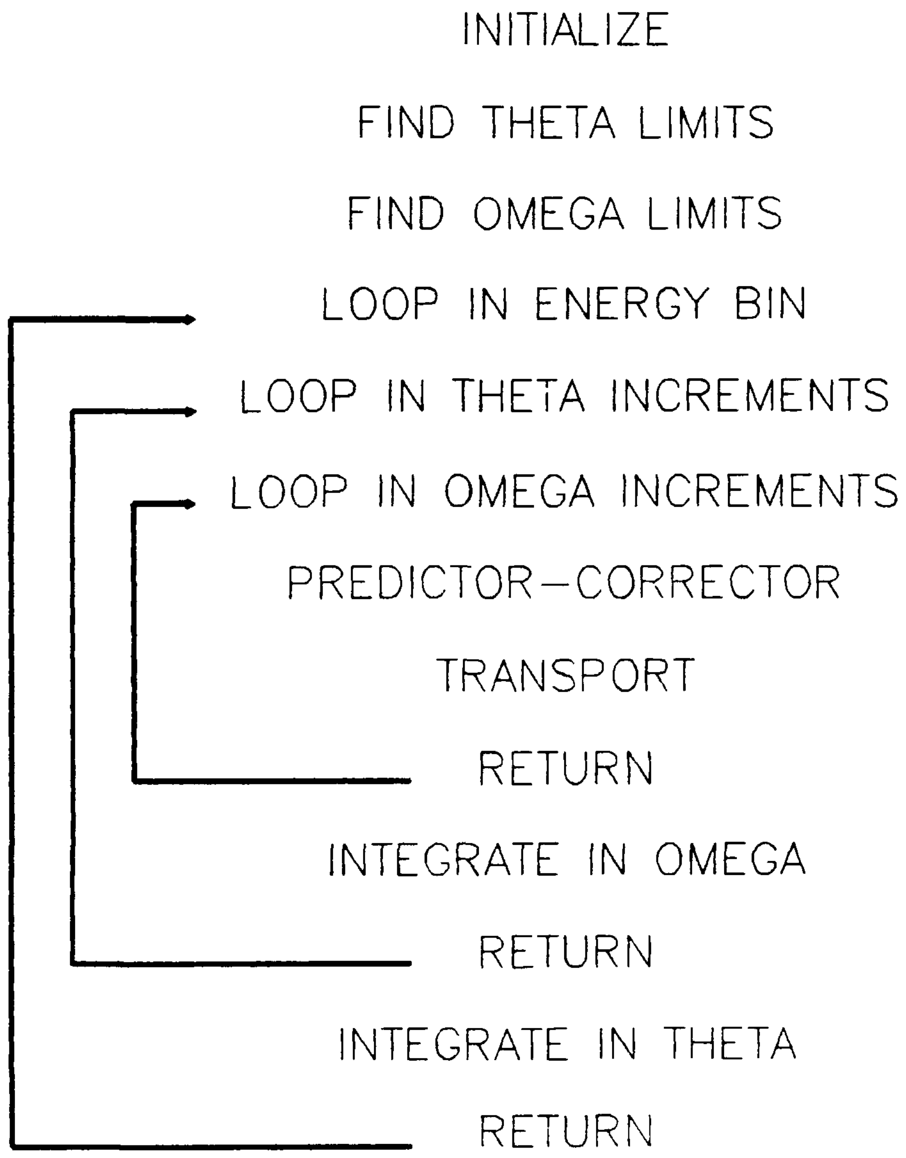


Figure 5-5 Albedo Code Algorithm

The albedo code next steps through nested loops for each energy bin at the detector and for each surface area mesh location. For a specific albedo energy bin and mesh point, the predictor-corrector scheme is exercised to determine the source energy bin, reflective surface location and angle of incidence. The angles defining the direction of transport from the surface to the detector are also calculated.

The albedo flux at the detector is then calculated via the transport process previously described, including the energy-time transform from Section 3. This is done for each mesh point in omega for a given theta value, followed by an integration across the omega mesh points to get a single value for albedo flux at each given theta increment. The code then performs an integration along the line of theta values to get the total albedo flux from the surface onto the detector for the given albedo energy bin. The code then loops to the next albedo energy bin and the transport sequence is engaged again.

The last step for the albedo code is to sum the contributions from each energy bin to get the total albedo flux at the detector for a set time since emission.

5.4 Test Case - Initial Code and Results

Determining a test case is subjective, and the configuration was chosen to produce significant albedo flux by keeping the source and detector fairly close together. The source is set at 150 km above the earth's surface and the detector at 250 km. The separation angle is 0.08 radians, which correlates to a distance of about 500 km over the earth's surface.

The initial albedo code assumed that all reflected neutrons emerged with a Lambertian distribution. This is because the first version of the code was completed prior to the accumulation of the necessary Monte Carlo angle distribution data.

In the case of an isotropic scatter event, the probability of the particle scattering into any one particular direction (per steradian) would be $1/(4\pi)$. In this case, one could view the reflecting surface as limiting the particle's new direction to half of the isotropic case, that being only the hemisphere above the surface. Now the probability of scattering into one particular direction (per steradian) above the surface is twice than before, or $1/(2\pi)$.

The initial code employed a low order integration scheme, composite midpoint numerical quaderature. This was in anticipation that the albedo flux off the surface will exhibit complex behavior in energy and angle, yet the coarseness of the energy and angle bin approximations will cause the calculated albedo flux not to be smoothly behaved. This means that high order integration schemes could produce less accurate answers.

The mesh size should play an important role in the accuracy of the albedo code. The finer the mesh the better the results but the longer the code runs. A mesh of 49 points (seven panels in theta and omega) was demonstrated to be sufficient for convergence.

The results from running this test case using this simple version of the code is shown in Figure 5-6. Note that the general behavior of the albedo flux with time is to rise quickly to a peak by one-tenth of a second after emission and then to fall off at a roughly exponential rate.

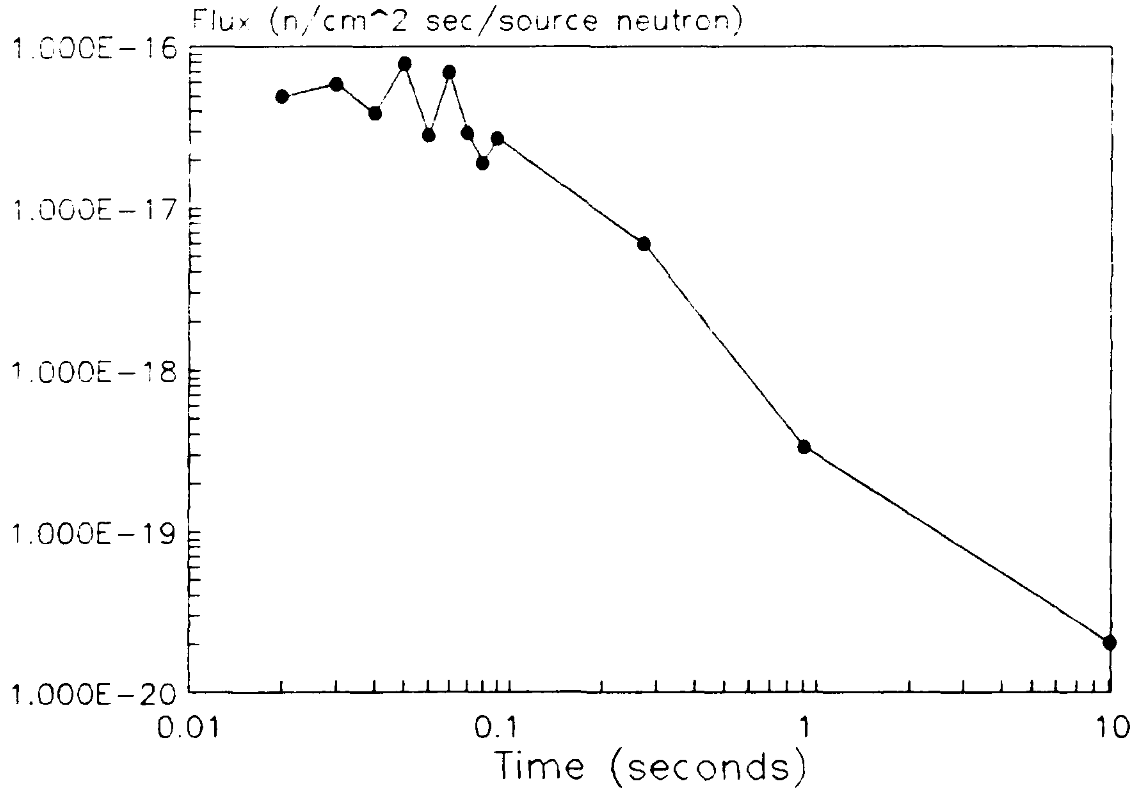


Figure 5-6 Initial Test Case Results

Note also the erratic nature of the results for time less than 0.1 seconds. This behavior could be a product of the mesh size and integration scheme or due to the coarseness of the bin approach to the problem, or a combination of these factors. Unfortunately, varying the mesh size and integration scheme did not eliminate the raggedness of the results.

A large part of the problem is that only a fraction of the common surface area may contribute albedo flux to the detector. The sections of the common surface which do not contribute will then only degrade the accuracy of the integration scheme by including zero values to the integrand. There are two additional constraints which limit the size of the surface area to be integrated;

1. For a given point on the surface and time since emission, source neutrons might not have enough time to get there and reflect up to the detector. If the maximum velocity source neutrons cannot travel to this location, scatter into the specified lower energy and transport to the detector because of time limitations, then this location on the surface should be eliminated early in the code sequence, at least for each specified detector energy bin loop.

2. Source neutrons cannot scatter into higher energies. In the predictor-corrector sequence, the calculated initial energy may be lower than the albedo energy. If this occurs, then the location contributing to this should also be eliminated prior to the transport portion of the albedo code.

5.5 Revised Albedo Algorithm

Both types of the above constraints are dependent upon the albedo energy group. At the higher albedo energies, the upscatter constraint could be dominant. This is because the fast albedo neutrons would leave considerable time for the transport of the source neutrons, which equates to low initial energies.

The reverse could be true for low energy albedo neutrons, in which they may consume all the time available in their transport from the surface (in the predictor-corrector approach), leaving no time to define the initial energy of the source neutrons. Or there may be so little time remaining that the initial energy calculated is greater then the highest energy available in the source spectrum.

These two constraints are both characterized by

$$t = \frac{r_1}{v_1} + \frac{r_2}{v_2} \qquad 36.$$

where T represents the overall time or time since emission.

The albedo code uses the above equation to test surface locations to see if the point is forbidden from consideration prior to setting up the surface mesh. In the case of the upscatter threshold, both velocities are the same and are the velocity associated with the albedo energy. Then if the pathlengths associated with a particular surface point cause the time of flight to be less than the overall time allowed, then that point is outside the range of applicability and is eliminated from the problem.

On the other hand, the source velocity (v_1) for the first constraint is that associated with the greatest energy in the source spectrum.

Note that the above equation defines an ellipse in space with the foci being the source and detector locations. The intersection of the upscatter ellipse with the upper surface of the atmosphere forms an ellipse as well. The intersection of the ellipsoid defined by the first constraint forms an ellipsoid on the surface which is skewed due to the different velocity components.

Figure 5-7 displays an example of how these two time-of-flight constraints can limit the surface area of integration. Here the first constraint is labeled TOFmax and the upscatter constraint is labeled TOFmin. The surface area which remains for transport is the area between the two ellipses. As the albedo code steps down in energy bins, the surface area boundaries will migrate in towards the source-detector focal line.

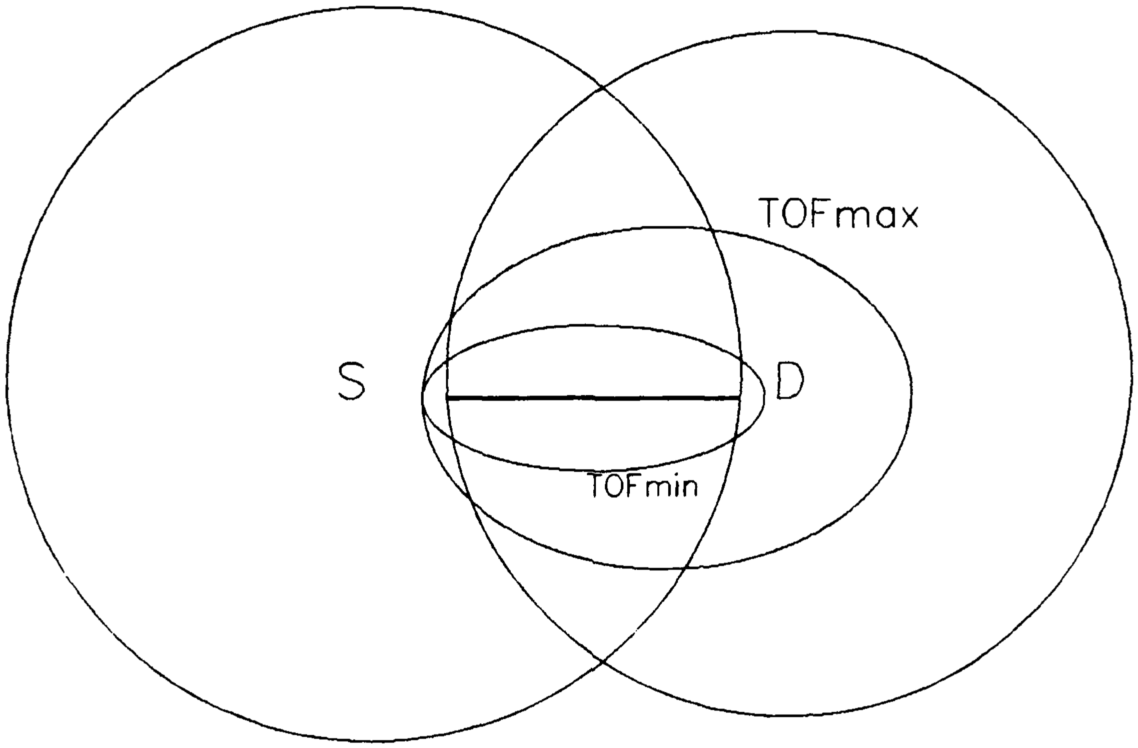


Figure 5-7 Time-of-Flight Constraints

The elimination of the out of range surface regions also eliminates zero values from the integrand of the albedo code, thereby increasing its accuracy. Since the time-of-flight tests rely upon the albedo energies, the section of the albedo code which determines the dimensions to the applicable surface area must be moved within the albedo energy bin loop. The shape of the surface area changes with each albedo energy bin.

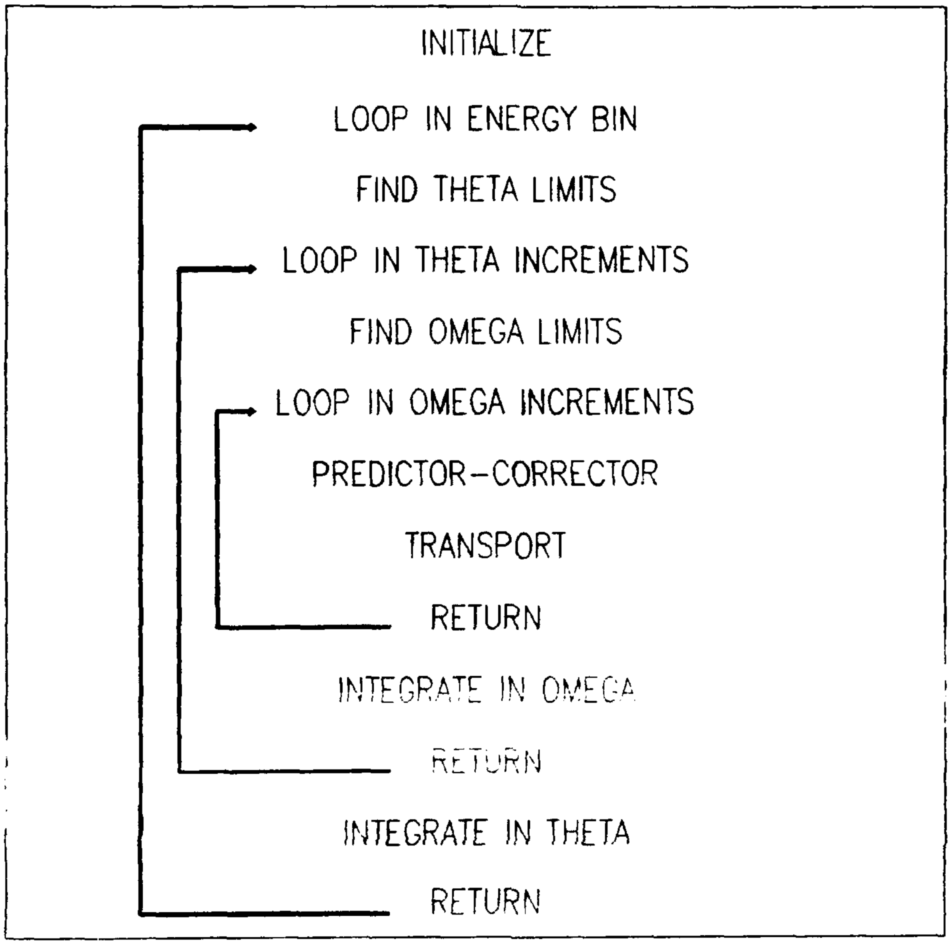


Figure 5-8 Revised Algorithm

The remaining surface area can be unusually shaped and dividing it into a uniform mesh may be impossible. In fact, the refinement of the algorithm to the point where none of the mesh points produced zero values was never achieved. Frequently the outer one or sometimes more mesh points along omega still contribute zero values. On the whole, however, the algorithm is much more efficient as the majority of the integrated surface contributes albedo flux to the detector.

Determining the new surface boundaries due to the time-of-flight constraints is accomplished by testing specific points upon the surface within bisection sorting routines. The routine steps along the surface until it finds the point which falls outside the ellipsoids of Figure 5-7. This point is then a boundary for the surface area to be integrated over. The routine also incorporates a predictor-corrector scheme in order to account for height variations to the reflecting surface; this slows down the code but is necessary due to the coupled nature of this problem. This routine is used to find the maximum and minimum values for theta and omega.

It was also realized that the inclusion of the actual albedo angle distributions will add more zero values to the problem because of the forward bias of the reflected neutrons with energies which are within about 20% of the initial energy. For the purpose of reducing the jaggedness in the results, all of the albedo angle distributions are modeled as that of the 1.8-1.1 MeV incident neutron case (see Section 4.2.4). This distribution is void of null contributions at any reflected angle, compared to that of the 14.2-14.9 MeV case for example. The additional error introduced by this decision should be secondary in most cases because the bias in the angle distributions, when present, is frequently masked by the energy bins anyway.

5.6 Test Case - Revised Code Results and Convergence

The revised albedo code generated the results for the test case as shown in Figure 5-9. Appendix C contains the albedo source code. Various mesh sizes were used to evaluate the convergence of the results.

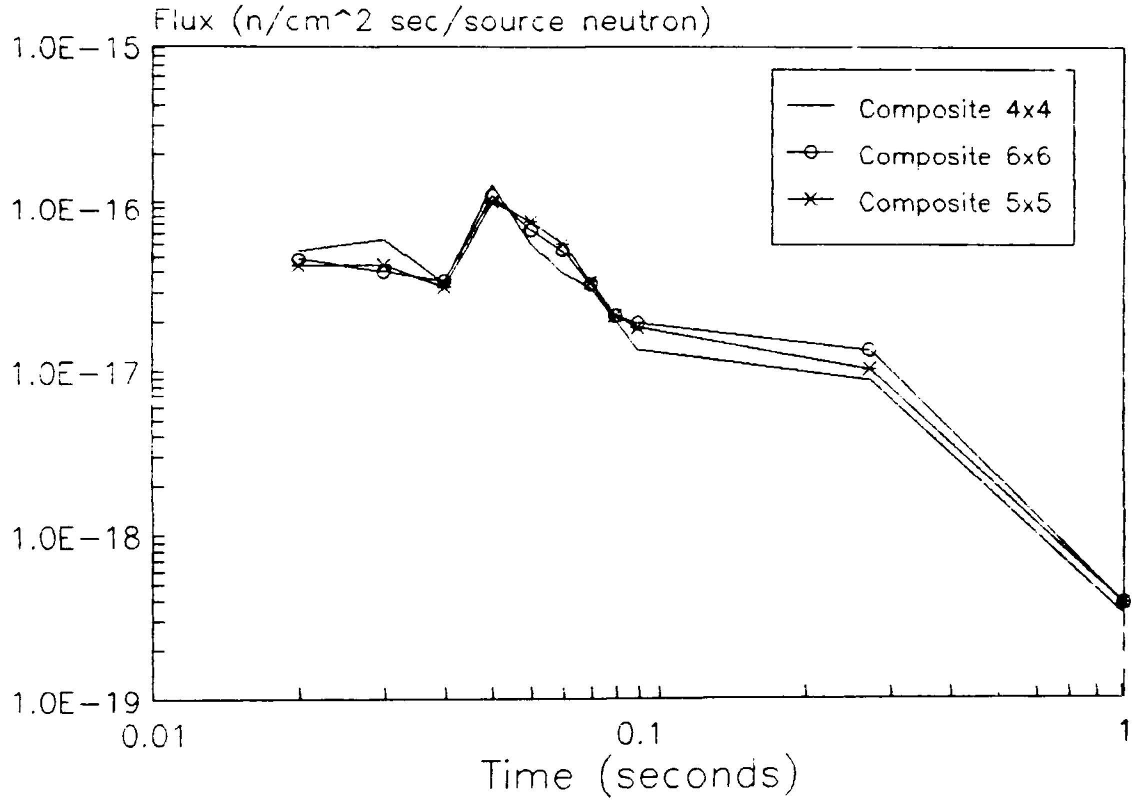


Figure 5-9 Albedo Spectrum vs Mesh Size

Test Case Results using Composite Midpoint

As before the albedo flux rises to a peak then falls off over time, except the peak is a little better defined then before. The results seem to be converged as there is little difference between the five-by-five versus the six-by-six mesh.

The computer run time for the 4-by-4 mesh was a maximum of four minutes for data points at about the flux peak and about one minute by 0.1 seconds and beyond. The times for the 5-by-5 mesh are 5.5 minutes at the peak and two minutes minimum, while the 6-by-6 mesh took 7.5 minutes and 3.5 minutes accordingly. Since there is little difference in results between the latter two mesh sizes, the 5-by-5 mesh seems to be the best choice when using the composite midpoint rule.

The effect of the integration scheme on convergence and accuracy was also evaluated by using a higher order algorithm. Simpson's 3/8 Rule was tried, a *fourth order scheme*, but there appears not to be any improvement in the results when comparing like mesh sizes. Figure 5-10 shows the results for various mesh sizes; note that Simpson's 3/8 Rule requires an even number of panels or equal area increments, which equates to one additional mesh point. Composite midpoint 7x7 and Simpson's Rule 8x8 both use seven panels, but Simpson's Rule relies upon values from the edges of each panel, while composite midpoint uses values at the center or midpoint of each panel.

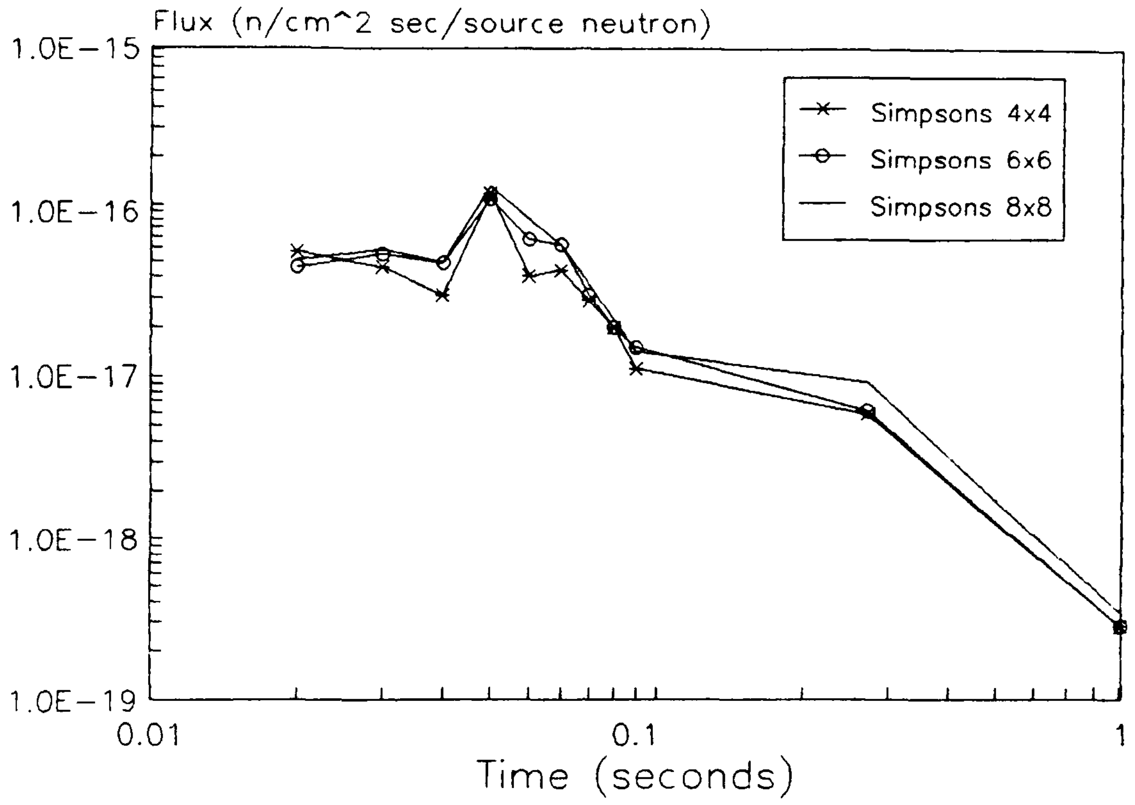


Figure 5-10 Albedo Flux vs Mesh Size

Test Case Results using Simpson's Rule

The Simpson's Rule takes a bit longer than composite midpoint for the same number of panels because of the additional mesh point. Simpson's Rule does not appear to offer any advantage over the composite midpoint scheme as it does not converge until the 8-by-8 mesh, which takes up to 15 minutes to complete a data point in time. This poor performance is probably due to the fact that Simpson's Rule uses mesh points from the edge of the applicable surface area, which are susceptible to zero values, while composite midpoint quadrature does not.

A comparison between the Composite Midpoint 5x5 and Simpson's Rule 8x8 in Figure 5-11 shows a slightly more well behaved result from the 8x8 mesh and Simpson's Rule, but the increase in run time of a factor of three removes the higher order integration schemes from consideration.

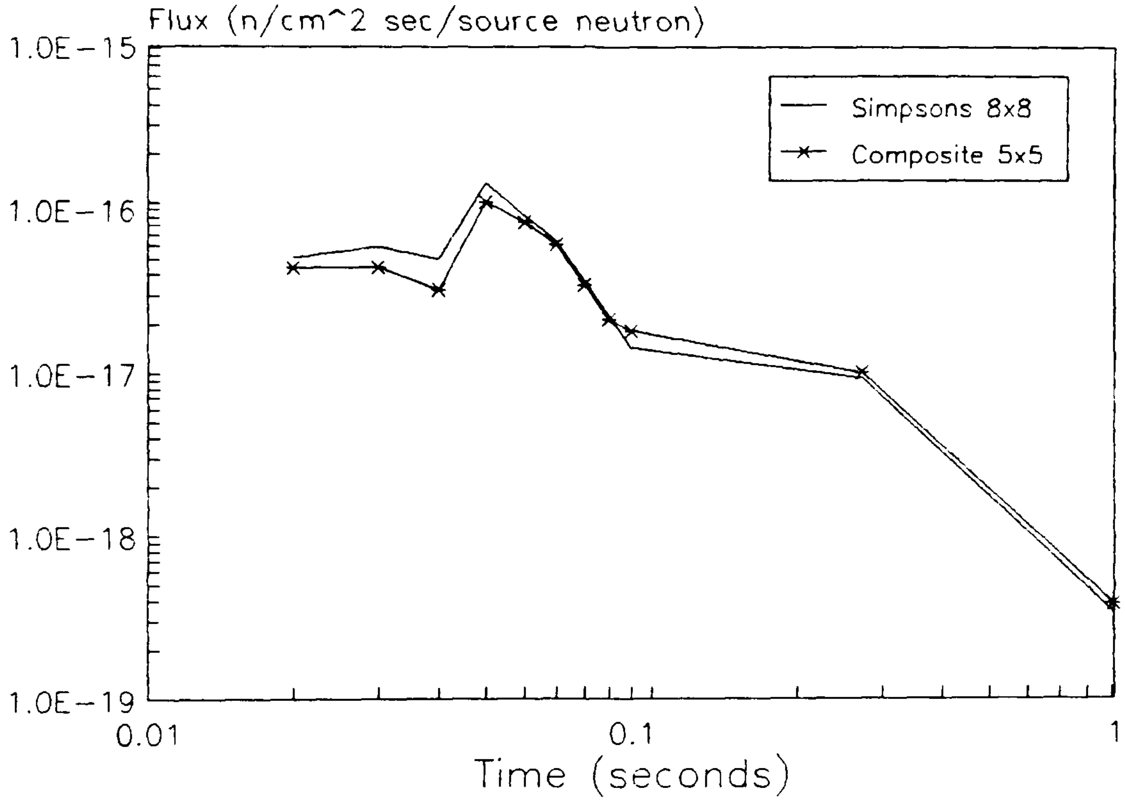


Figure 5-11 Comparison of Composite Midpoint 5x5

Versus Simpson's Rule 8x8

In the comparisons of the results from different mesh sizes and integration schemes, one observes that there are some unstable or jagged features which are common to all permutations. For example, the albedo flux at the time since emission of 0.4 seconds is always depressed relative to its surrounding values. There is no physical reason for this to occur; rather one expects the flux at this time should be greater so that the rise in the flux to its peak value exhibits a smoother behavior. These occasional depressions or bumps are attributable to the energy bin approach to this problem. Since the source and albedo spectra being used are made up of bins of nonuniform width, then the jumps in velocity of the particles from one bin to the next are also erratic. Figure 5-12 displays a typical energy spectrum of the albedo flux which results from the albedo code. The albedo flux in each energy bin is graphed versus the midpoint of each energy bin.

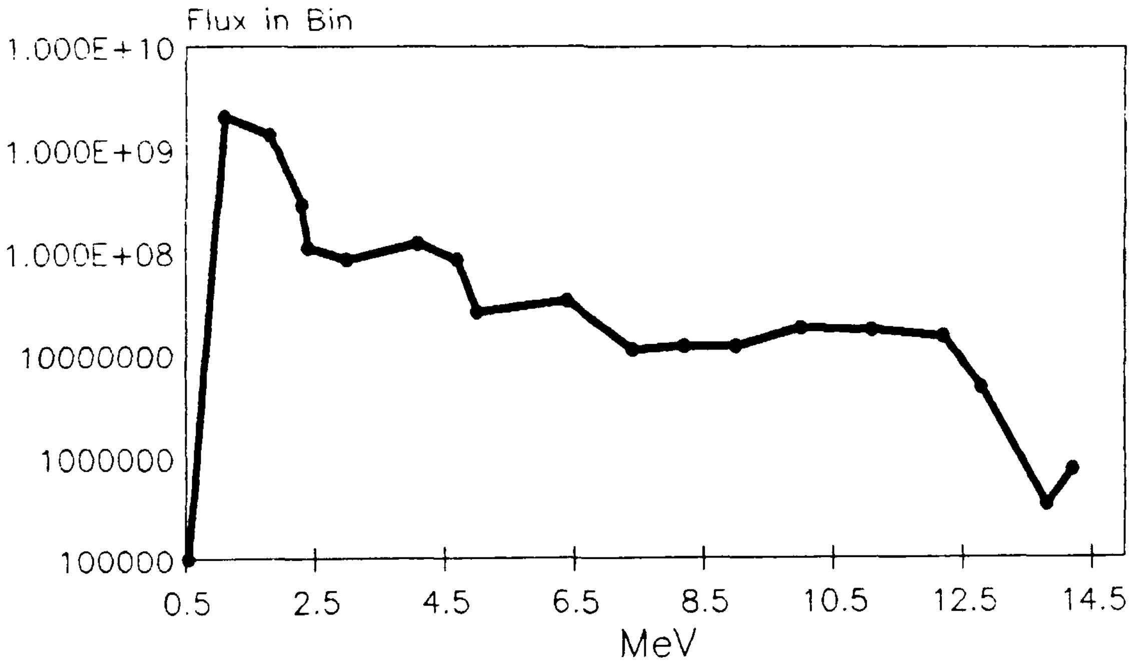


Figure 5-12 Typical Albedo Energy Spectrum at the Detector

As the albedo code steps down towards the lower albedo energy bins, the relative jumps in neutron velocities grow larger, which cause the first time-of-flight constraint to lurch over greater distances of the surface. In fact, it can jump suddenly to the point where it eliminates all of the surface area, which shuts down the program.

When this happens, the albedo spectrum in energy can quite suddenly go to zero, instead of a more gradual tapering off. So depending upon how the energy bin midpoint velocities match up against the actual velocity profile of the albedo flux, the flux can exhibit local depressions and or rises in the calculated answers. These variations are unpredictable in their occurrence anywhere along the spectrum, but are most evident at the early times about the peak when the flux is rapidly changing.

It appears that the jumpiness from the bin nature of the approach to this problem contributes more uncertainty to the results than the fineness of the surface mesh or the type of integration scheme used. Therefore, the Composite Midpoint 5x5 Mesh combination will be used for calculating the results in the next section.

6 Albedo Code Results

6.1 Variation in Source or Detector Altitude

As a continuation of the test case, the altitudes of the source and detector will be varied in order to observe the effects upon the albedo flux. The ground range will be kept constant at about 500 kilometers (0.08 radians). The composite midpoint 5x5 mesh is used due to its quick run time and adequate convergence.

Initially the detector altitude will be fixed at 250 kilometers above the earth's surface, while the altitude of the source is varied from the test case of 150 kilometers up to 1000 kilometers. The albedo code will calculate the free field flux at the detector at numerous times since emission. The time increments were varied so that the majority of the data points fall over the time period when the albedo flux is changing most, which is about its peak.

The results from the albedo code are presented in Figure 6-1. The albedo flux at the detector is given for source altitudes of 150 km, 600 km, and 1000 km. In each case the albedo flux rises quickly to a maximum and then falls off in magnitude over time. As the source rises in altitude the magnitude of the albedo flux decreases and the peak shifts to the right.

These trends were anticipated and make sense. The albedo flux drops in magnitude as the source rises due to the increase in pathlength and hence spherical divergence. The peak shifts in time for the same reason. The flux is spread out over a long period of time because the source spectrum is spread out over many orders of magnitude in energy and velocity.

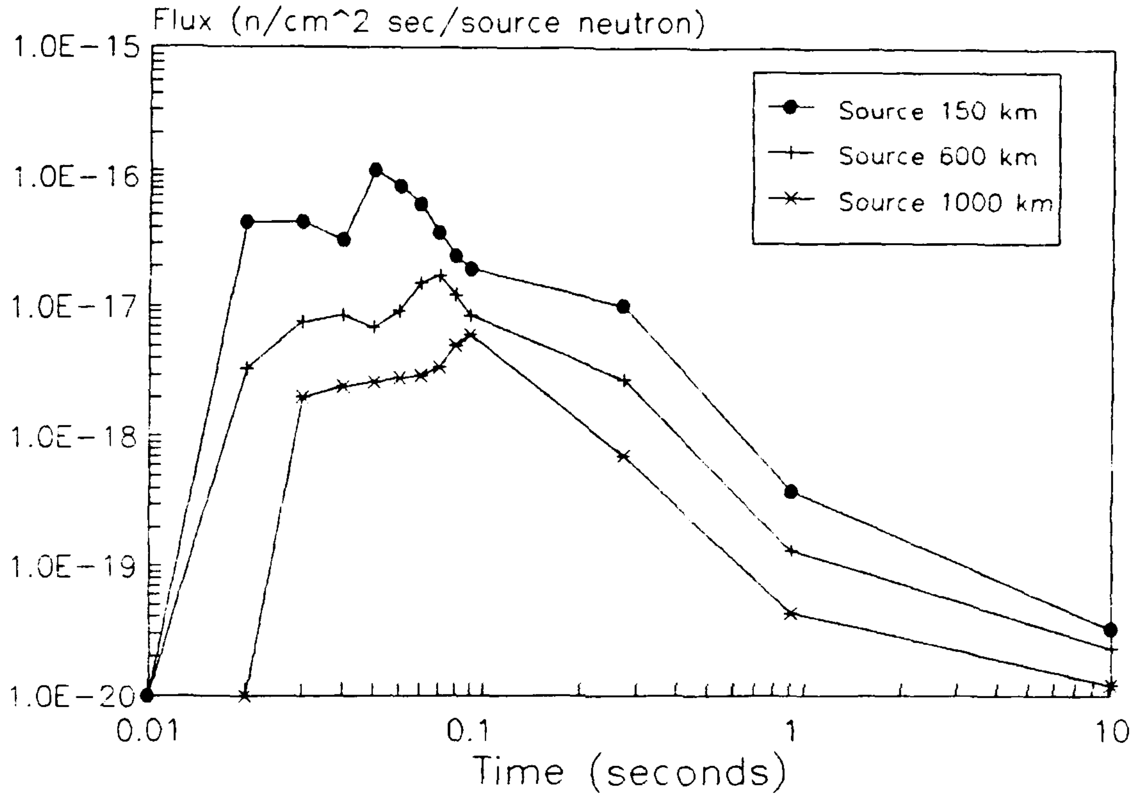


Figure 6-1 Albedo Flux with Detector at 250 Kilometers

With the source at 150 km, the peak flux occurs around 0.05 seconds. With the source at 600 km, the peak flux occurs about 0.07 seconds. With the source at 1000 km, the peak is around 0.09 seconds. Note that the rate at which the flux falls off over time can be attributed to the "1/t" behavior from the time-energy transform. Two examples of the albedo flux for a detector at 1000 kilometers are shown in Figure 6-2.

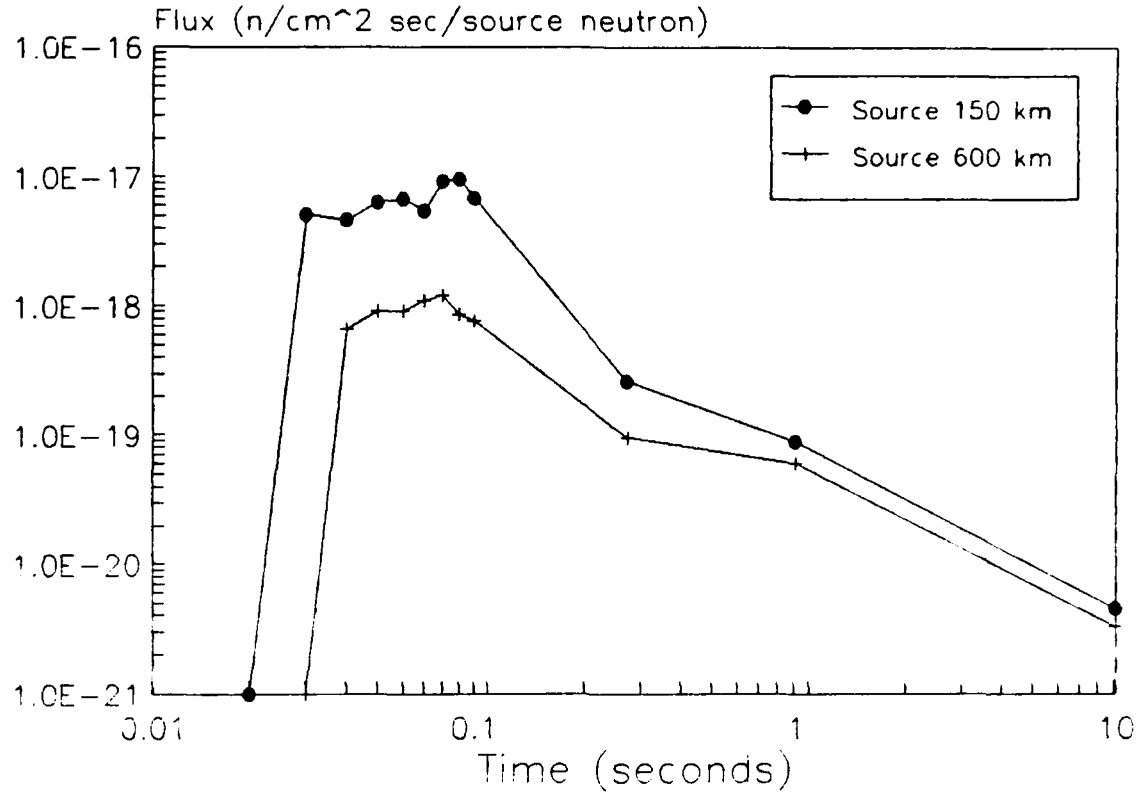


Figure 6-2 Albedo Flux with Detector at 1000 Kilometers

Here too the magnitude of the flux drops with increasing pathlength and the spectrum shifts to the right. One might expect the Detector250/Source1000 case to be very similar to the Detector1000/Source150 case due to the nearly equal pathlengths involved; in fact the Detector250/Source1000 case displays less albedo flux. This is due to the fact that a greater fraction of source radiation interacts with the atmosphere with the source at 150 km.

6.2 Variation in Ground Range

Next the affects of keeping the source and detector altitudes constant but varying the distance between them will be investigated. To keep the cases as simple as possible, both the source and detector are at identical altitudes of 600 kilometers. By starting the separation at 0.08 radians and incrementing upwards, the geometric cases will begin as Case-3a2 and shift into Case-3b1 prior to Case-2.

Figure 6-3 presents the total albedo flux versus angle of separation.

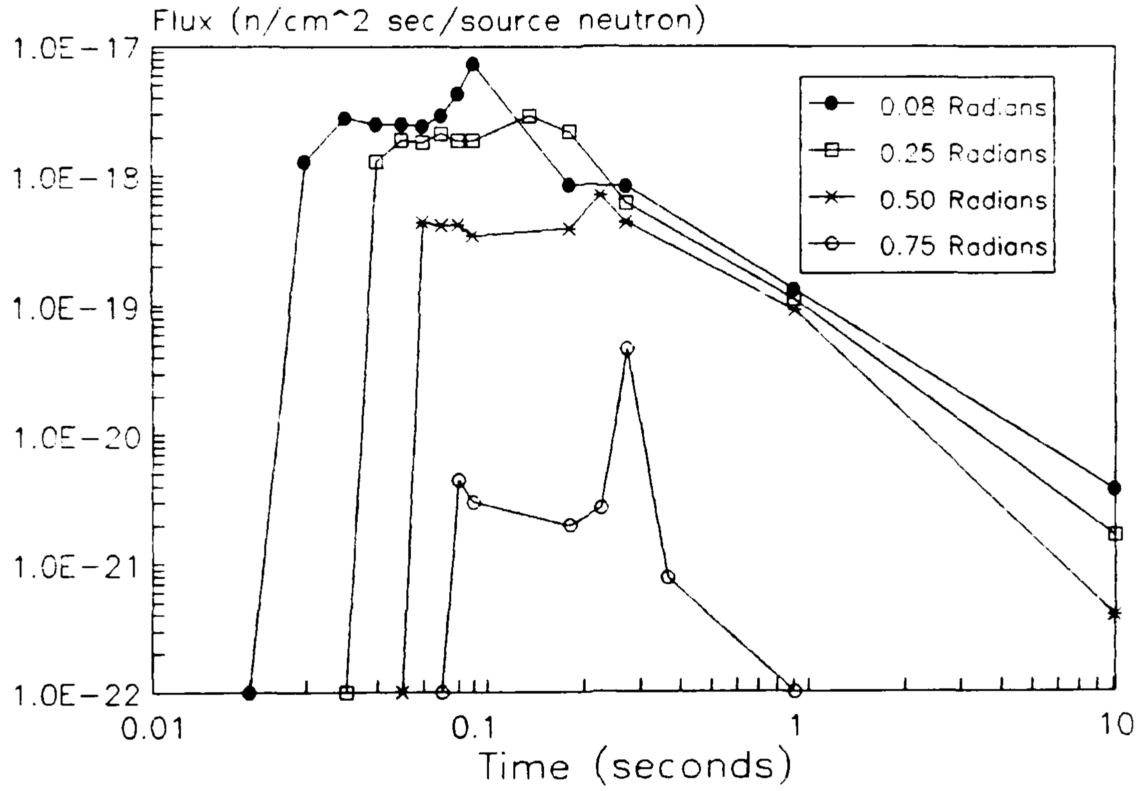


Figure 6-3 Albedo Flux vs Separation Angle

As expected the albedo flux generally decreases with increasing distance between the source and detector, and the flux peaks at later times. As this distance or ground range increases, the earlier approximations using local planar geometry grow weaker and weaker, but do not appear to affect the behavior of the results until the separation approaches 0.78 radians, which is the maximum angle before the surface areas cut by the source and detector's points of tangency no longer intersect. The results become quite ragged near this limit, but this behavior can also be attributed to numerical instabilities from modeling a very small and complex surface area.

7 Conclusions and Recommendations

The Green's function methodology, incorporating source and energy bin structure into solving a complex radiation transport problem, indeed offers a viable technique for approaching difficult problems such as this. One can generate a quick running computer algorithm or code built upon this approach; the albedo code assembled in QuickBasic can be optimized, rewritten and integrated with any comprehensive main-frame exoatmospheric simulation code. The run time of such an end product should be under one second (per data point for a given configuration and time since emission), but probably much less.

For a complex and intense source spectrum, a significant albedo neutron flux may be generated, depending upon the distances involved. The magnitude of this free field flux has been calculated as large as 10^{-16} (neutrons/sq cm sec/source neutron); possibly enough to impact any shielding designs.

The albedo flux from a source defined by a Dirac pulse in time will rise quickly to a peak value and then fall off over a long period of time; at least ten seconds. The results from this single source pulse can also be applied through the principle of superposition to generate an albedo flux from a source made up of multiple pulses in time. In most practical cases, the detector and source are moving and therefore the "time of exposure" is finite and short.

Unfortunately, the behavior of the results from the albedo code are a bit erratic over some regions in time. Additional refinement of the albedo code and approach might be worthwhile. The following recommendations for improvements or investigations are offered.

1. The limitations in accuracy of this bin-dependent approach lies in the behavior within the energy bin structure itself. If this problem had been approached with a source and detector spectrum built with energy bins spaced in equal velocity increments, then the albedo flux would have exhibited a smoother behavior.

Unfortunately, few if any spectra are produced with equal velocity increments. A source spectrum could be generated with this type of problem in mind, but the Monte Carlo bin-to-bin data would have to be recalculated.

With the insight gained only at the end of a project, the reflected energy bin structure should not have been linked to the source energy bin structure during the Monte Carlo bin-to-bin calculations. This would have allowed for the construction of an albedo bin structure up front which consisted of equal velocity increments. I recommend that this approach be investigated.

2. There is some suspicion of the technique implemented in the albedo code in which only one source bin is considered to contribute to the albedo flux from a mesh point. In other words, the predictor-corrector scheme identifies only a single source bin per mesh point for transport

into a albedo energy bin. It seems that by "later" times there should be a buildup within each albedo energy bin per mesh point due to the diffusive down-scatter from multiple source bins. The time since emission for a high energy source neutron to rattle around for a while and into an particular albedo bin could be the same for a lower energy source neutron to scatter but a few times and into the same energy bin. The additional albedo flux from this mechanism may only be significant for such long times after emission that its contribution is not important. In any case, an investigation of this process should be conducted, perhaps in union with the above Monte C lo investigation. The end result could be in the form of some kind of Build-Up Factors, where the BUFs are functions of energies and time. Another idea is to alter the predictor-corrector scheme to add multiple source bins to each transport event per mesh point.

3. Further work could also be done on the albedo code's attempts to define the restricted surface area prior to its division into the mesh of equal areas. This effort was not totally successful in eliminating all of the out of range locations atop the atmosphere. Considerable time was spent toward this end, and while it did improve the results, in hindsight more time should have been dedicated to one of the other topics mentioned above. The elimination of all the zero values in the albedo flux from the surface will affect the results, though to a lesser extent than other improvements could.

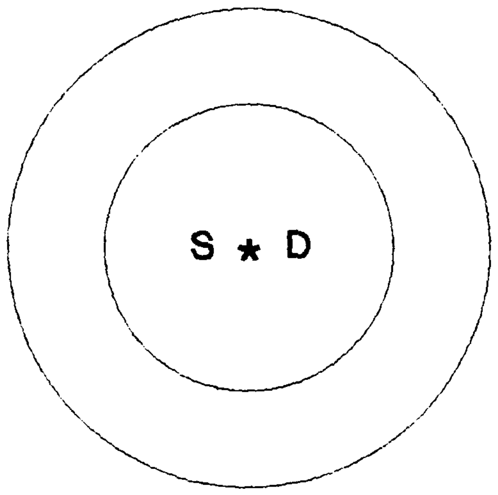
Appendix A: General Source Spectrum

Normalized Neutron Spectrum

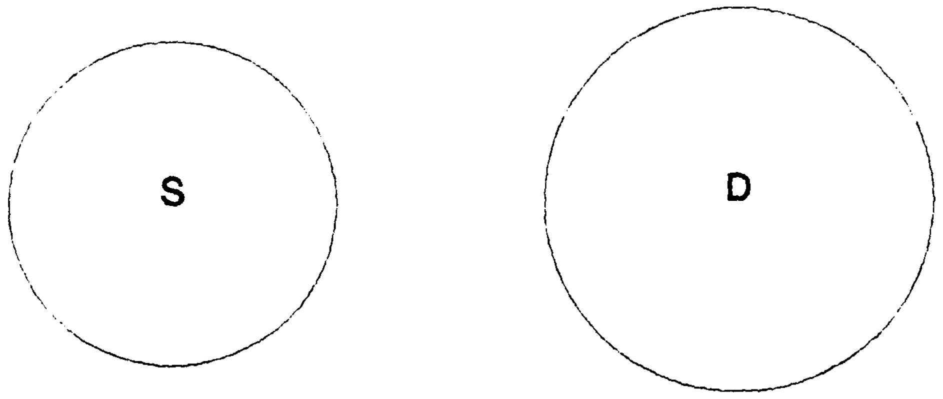
Group	Upper Energy Boundary (MeV)	Neutrons per Source Neutron
1	14.9	1.89-02
2	14.2	9.34-03
3	13.8	2.66-02
4	12.8	1.67-02
5	12.2	1.69-02
6	11.1	1.24-02
7	10.0	7.48-03
8	9.0	6.82-03
9	8.2	6.78-03
10	7.4	1.03-02
11	6.4	1.81-02
12	5.0	3.62-03
13	4.7	1.24-02
14	4.1	2.60-02
15	3.0	2.37-02
16	2.4	3.75-03
17	2.3	2.56-02
18	1.8	6.44-02
19	1.1	8.85-02
20	5.5-01	9.14-02
21	1.6-01	1.16-02
22	1.1-01	1.11-01
23	5.2-02	5.40-02
24	2.5-02	5.68-03
25	2.2-02	9.26-02
26	1.0-02	1.16-01
27	3.4-03	7.38-02
28	1.2-03	2.32-02
29	5.8-04	2.03-02
30	1.0-04	1.90-03
31	2.9-05	0.0
32	1.1-05	0.0
33	3.1-06	0.0
34	1.1-06	0.0
35	4.1-07	0.0
Lower Bound	1.0-11	
Total		1.0

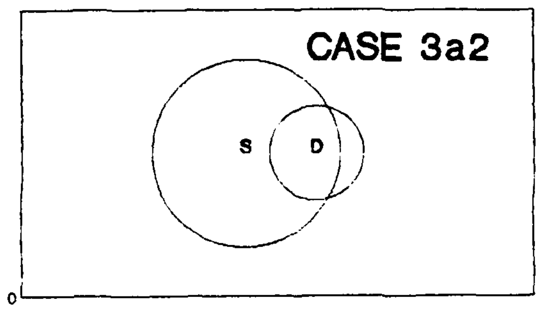
Appendix B: Source vs Detector Surface Geometries

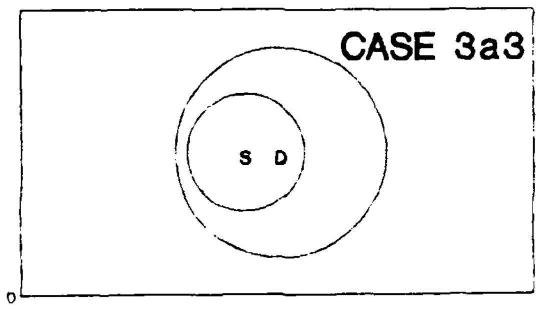
CASE 1

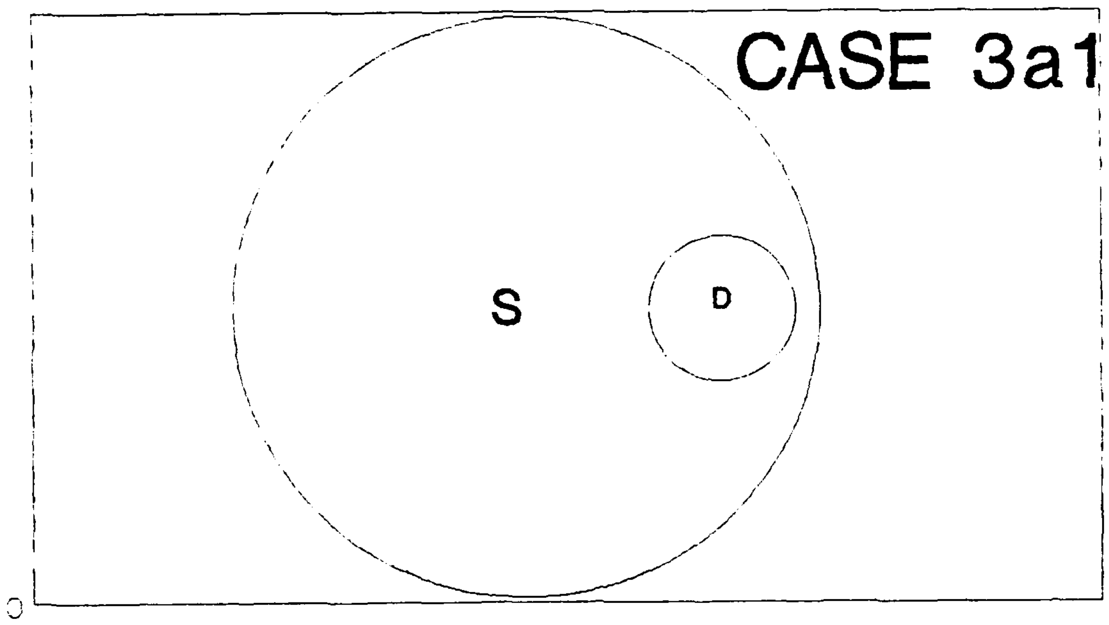


CASE 2

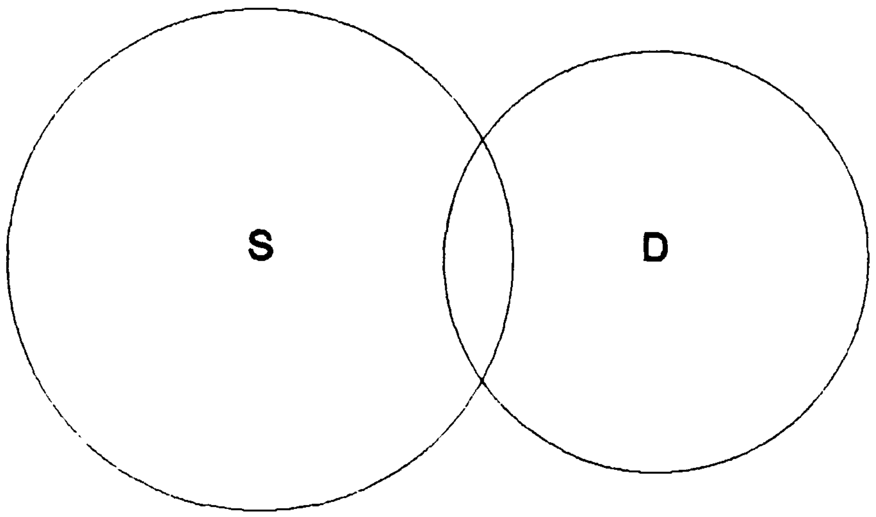



CASE 3a2
S
D
0

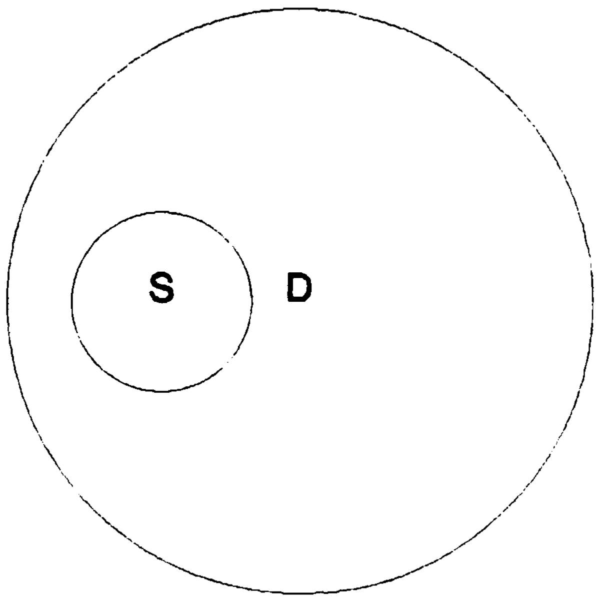

CASE 3a3
S
D
0


CASE 3a1
S
D
0

CASE3B1



CASE 3b2



Appendix C: Computer Code

This appendix contains the final version of the albedo code generated for this thesis. The code presented here is not a finished product but is the result of numerous rewrites, changes and additions during the course of the thesis effort. Various aspects to this code may seem odd; for example, the angle distribution of the albedo flux is initially taken as Lambertian, and then elsewhere the actual angle distribution data is normalized against a Lambertian distribution as a way of measuring the variance from the Lambertian. The code must still be optimized in QuickBasic 4.5 or in any other language.

```
DEFINT I-K
DEFDBL A-H, L-Z

DECLARE SUB Initialize ()
DECLARE SUB Atmoseff ()
DECLARE SUB Predict ()
DECLARE FUNCTION Prob (Eg, Egprime, Thetainc, thetat, phi)
DECLARE SUB Integrate (f#(), dist#, nangle#, G#)
DECLARE FUNCTION Xsection (E)
DECLARE SUB Transport (Eg, Egprime, W)
DECLARE SUB results ()
DECLARE SUB Bin (E, Eg)
DECLARE SUB Anglemu ()
DECLARE SUB Angleomega ()
DECLARE FUNCTION Thetai (Zeff, rs)

REDIM source(ngroup), Energy(ngroup), emid(ngroup)
REDIM mu(nmu), omega(nomega), FofT(ngroup)
REDIM A14(35, 6), A9(35, 6), A4(35, 6), A2(35, 6), A1(35, 6)
REDIM D75(20, 3), D45(20, 3), D15(20, 3), G75(10, 3), G45(10, 3), G15(10, 3)
pi = 3.1415928#

ver$ = "Program ALBEDO.BAS Version 1.0 dated 31 December 1990"


'What the Subroutines and Functions do;

'     Initialize - reads input data and sets up bin midpoint values
'                - determines cartesian source and detector location
'        Anglemu - determines polar angle associated with the limits
'                  to the integrating surface area and breaks angle
'                  down into equal increments in cosine
'     Angleomega - calculates the azimuthal angle associated with
'                  the limits to the integrating surface for a
'                  given polar angle and increments the angle
'        Predict - for a given neutron energy incident on detector and
'                  a given surface area location, this predicts the
'                  initial neutron energy from the source based upon
'                  distance traveled and time allowed
'       Atmoseff - adjusts the approximate effective top of the
'                  atmosphere (which is used as a reflective plane)
'                  based upon neutron energy and angle of incidence
'            Bin - determines the energy bin of a given energy value
'      Transport - calculates the attenuation by spherical divergence
'                  from source to the top of the atmosphere and then
'                  to the detector; assumes lambertain distribution but
'                  calls subroutine Prob for the actual angle and energy
'                probabilities, includes the time/energy transform
'         Prob - computes the probabilities for the initial neutron to
'                emerge from the atmosphere with the energy of
'                interest and with the direction to the target
'                (functions f and g)
'    Integrate - uses Composite Midpoint to calculate the albedo flux
'                over one dimension of the applicable surface area
'                at a time; i.e. this routine is called twice
'      Results - prints the results into an output file
'     Xsection - approximates the scatter cross section in air as
'                a function of energy
'       Thetai - determines the angle of incidence

'What the arrays mean;

'       Energy - the upper energy bounds for each energy bin
'         Emid - the midpoint energy of each energy bin
'       Source - the fraction of source neutrons in each bin
'           Mu - the polar angle increments
'        Omega - the azimuthal angle increments
'         FofT - the albedo neutron flux in each detector energy bin
'            A - contains the bin-to-bin energy
'                probabilities (function f)
'            D - contains the elevation angle distribution data
'            G - contains the azimuthal angle distribution data

CLS

CALL Initialize

OPEN "ALBEDO.OUT" FOR APPEND AS 2

REDIM FofT(ngroup)
Ftotal = 0

FOR i = 1 TO ngroup                 ' Loop in Energy Bin at Detector

Egprime = i

 CALL Anglemu

 IF thetat = 0 OR timefarlo <= T THEN GOTO 500

 REDIM G(nmu)
```

```
FOR j = 1 TO nmu - 1     ' Loop in Surface Area Polar Angle Incre-
ments

  CALL Angleomega

  IF omegat = 0 THEN GOTO 500

  REDIM f(nomega)

  FOR K = 1 TO nomega - 1  'Loop in Surface Area Azimuthal Increments
   E = 0

   Zeff = 6371000 + 40000

   CALL Predict               ' First Estimate

   CALL Atmoseff

   CALL Predict               ' Revised Estimate

   IF E > 0 THEN
    CALL Bin(E, Eg)
     IF Eg <= Egprime THEN
      CALL Transport(Eg, Egprime, W)
      f(K) = W
     END IF
   ELSE f(K) = 0
   END IF
  NEXT K

  dist = omegat * mu(j) * (6371000 + 70000)

  CALL Integrate(f(), dist, nomega - 1, q)

 G(j) = q

 NEXT j

 n = .33 * 6.023E+23 * Yield

 dist = thetat * (6371000 + 70000)
 CALL Integrate(G(), dist, nmu - 1, qq)

 FofT(i) = 2 * qq * .0001 * n          '  2 for azimuthal symmetry
                                       ' .0001 converts m^2 to cm^2

500 Ftotal = Ftotal + FofT(i)
```

```
PRINT "Just completed bin"; i; "with FofI ="; FofT(i)

NEXT i

PRINT "Just completed time of"; T; "with FofT ="; Ftotal

CALL results

END

'                              Subroutine Anglemu
SUB Anglemu STATIC
SHARED Zd, Zs, Zsi, Zdi, Xs, thetat, thetasource, thetadet, mu(), nmu, i
SHARED thetanot, emid(), T, thetamax, thetamin, vmax, vi, Zeff, timefarlo
CONST pi = 3.141592654#

Zeff = 6441000
vi = SQR((2 * emid(i) * 1.602E-13) / 1.67482E-27)  ' velocity of energy
index
vmax = 5.2759E+07                               ' velocity of maximum energy

' CASE 1

IF thetanot = 0 THEN
  IF Zd < Zs THEN thetat = 2 * ATN(SQR(Zd ^ 2 - Zeff ^ 2) / Zeff)
  IF Zs < Zd THEN thetat = 2 * ATN(SQR(Zs ^ 2 - Zeff ^ 2) / Zeff)

' First to test if the TOF limits constrain the geometric surface area

  timemax = (Zsi - 70000) / vmax + (Zdi - 70000) / vi    ' for TOFmax
  timemin = (Zsi - 70000) / vi + (Zdi - 70000) / vi      ' for TOFmin

  IF timemax >= T THEN thetat = 0
  IF timemax < T AND timemin >= T THEN thetamin = 0

  ' Bisection sequence to determine the maximum angle in theta
  IF timemax < T THEN
   t1 = thetat
   t2 = 0
   DO
    t3 = .5 * (t1 + t2)
    rs = SQR(Zs ^ 2 + Zeff ^ 2 - 2 * Zeff * Zs * COS(t3))
    rd = SQR(Zd ^ 2 + Zeff ^ 2 - 2 * Zeff * Zd * COS(t3))
    Thetainc = Thetai(Zeff, rs)
    E = emid(1)
    CALL Atmoseff
    rs = SQR(Zs ^ 2 + Zeff ^ 2 - 2 * Zeff * Zs * COS(t3))
    rd = SQR(Zd ^ 2 + Zeff ^ 2 - 2 * Zeff * Zd * COS(t3))
    Tprime = rs / vmax + rd / vi
    delta = Tprime - T
    IF delta > .005 THEN t1 = t3
    IF delta < -.005 THEN t2 = t3
   LOOP UNTIL ABS(delta) < .005
  END IF

  IF t3 >= thetat THEN thetamax = thetat
  IF t3 < thetat THEN thetamax = t3

  ' Bisection sequence to determine the mimimum angle in theta
  IF timemax < T AND timemin > T THEN
  t1 = thetat
  t2 = 0
   DO
    t3 = .5 * (t1 + t2)
    rs = SQR(Zs ^ 2 + Zeff ^ 2 - 2 * Zeff * Zs * COS(t3))
    rd = SQR(Zd ^ 2 + Zeff ^ 2 - 2 * Zeff * Zd * COS(t3))
    Thetainc = Thetai(Zeff, rs)
    E = emid(i)
    CALL Atmoseff
    rs = SQR(Zs ^ 2 + Zeff ^ 2 - 2 * Zeff * Zs * COS(t3))
    rd = SQR(Zd ^ 2 + Zeff ^ 2 - 2 * Zeff * Zd * COS(t3))
    Tprime = rs / vi + rd / vi
    delta = Tprime - T
    IF delta > .005 THEN t1 = t3
    IF delta < -.005 THEN t2 = t3
   LOOP UNTIL ABS(delta) < .005
   IF t3 < thetamax THEN thetamin = t3
   IF t3 >= thetamax THEN thetat = 0
  END IF
 IF thetat = 0 GOTO 200
END IF

' This determines the angles defined by points of tangency about the
poles
thetadet = ATN(SQR(Zd ^ 2 - Zeff ^ 2) / Zeff)
thetasource = ATN(SQR((Zsi + 6371000) ^ 2 - Zeff ^ 2) / Zeff)

' CASE 3a1

IF thetanot > 0 AND thetasource >= thetanot + thetadet THEN

  thetat = thetadet
' First to test if the TOF limits constrain the geometric surface area
```

```
timemax = (SQR((Zs - Zeff) ^ 2 + Xs ^ 2)) / vmax + (Zdi - 70000) / vi
 timemin = (SQR((Zs - Zeff) ^ 2 + Xs ^ 2)) / vi + (Zdi - 70000) / vi
 IF timemax >= T THEN thetat = 0
 IF timemax < T AND timemin >= T THEN thetamin = 0

 IF thetat = 0 THEN GOTO 200

 timefarhi = SQR((Zs - Zeff * COS(thetadet)) ^ 2 + (Xs + Zeff * SIN(the-
tadet)) ^ 2) / vmax + Zd * SIN(thetadet) / vi
 timefarlo = SQR((Zs - Zeff * COS(thetadet)) ^ 2 + (Xs + Zeff *
SIN(thetadet)) ^ 2) / vi + Zd * SIN(thetadet) / vi
 timecheck = SQR((Zs - Zeff * COS(thetadet)) ^ 2 + (Xs - Zeff *
SIN(thetadet)) ^ 2) / vmax + Zd * SIN(thetadet) / vi

 IF timefarlo <= T THEN GOTO 200  ' upscatter limit outside of area

 IF timefarhi <= T THEN     ' geometric maximum allowable by TOFmax
   thetamax = thetat
 END IF

 IF timecheck <= T THEN     ' geometric maximum allowable by TOFmax
   thetamax = thetat
 END IF

 ' Bisection sequence to determine the maximum angle in theta
 IF timecheck > T THEN
   t1 = thetat
   t2 = 0
  DO
   t3 = .5 * (t1 + t2)
   rs = SQR(Zs ^ 2 + Zeff ^ 2 - 2 * Zeff * Zs * COS(t3))
   rd = SQR(Zd ^ 2 + Zeff ^ 2 - 2 * Zeff * Zd * COS(t3))
   Thetainc = Thetai(Zeff, rs)
   E = emid(1)
   CALL Atmoseff
   rs = SQR(Zs ^ 2 + Zeff ^ 2 - 2 * Zeff * Zs * COS(t3))
   rd = SQR(Zd ^ 2 + Zeff ^ 2 - 2 * Zeff * Zd * COS(t3))
   Tprime = rs / vmax + rd / vi
   delta = Tprime - T
   IF delta > .005 THEN t1 = t3
   IF delta < -.005 THEN t2 = t3
  LOOP UNTIL ABS(delta) < .005 OR t3 < .01 OR t3 > .99 * thetat
   thetamax = t3
 END IF
```

```
  IF t3 <= .01 THEN
   thetat = 0
   GOTO 200
  END IF

  ' Bisection sequence to determine the mimimum angle in theta
  thetamin = 0
  IF timemin < T THEN
   t1 = thetasource
   t2 = 0
   DO
    t3 = .5 * (t1 + t2)
    Zeff = 6441000
    rs = SQR((Zs - Zeff * COS(t3)) ^ 2 + (Xs + Zeff * SIN(t3)) ^ 2)
    rd = SQR(Zd ^ 2 + Zeff ^ 2 - 2 * Zeff * Zd * COS(t3))
    Thetainc = Thetai(Zeff, rs)
    E = emid(i)
    CALL Atmoseff
    rs = SQR((Zs - Zeff * COS(t3)) ^ 2 + (Xs + Zeff * SIN(t3)) ^ 2)
    rd = SQR(Zd ^ 2 + Zeff ^ 2 - 2 * Zeff * Zd * COS(t3))
    Tprime = rs / vi + rd / vi
    delta = Tprime - T
    IF delta > .001 THEN t1 = t3
    IF delta < -.001 THEN t2 = t3
   LOOP UNTIL ABS(delta) < .001 OR t3 < .01 OR t3 > .95 * thetasource
   IF t3 < thetamax THEN thetamin = t3
   IF t3 >= thetamax THEN thetat = 0
  END IF

END IF

' CASES 3a2 and 3a3

IF thetanot > 0 AND thetasource <= thetanot + thetadet THEN
 IF thetasource + thetanot > thetadet THEN thetat = thetadet  'Case3a2
 IF thetasource + thetanot < thetadet THEN thetat = .98 * thetasource +
thetanot

' First to test if the TOF limits constrain the geometric surface area
  timemin = SQR((Zs - Zeff) ^ 2 + Xs ^ 2) / vi + (Zd - Zeff) / vi
  timemax = SQR((Zs - Zeff) ^ 2 + Xs ^ 2) / vmax + (Zd - Zeff) / vi

  IF timemax >= T THEN thetat = 0  'Fastest neutrons cannot reach
detector
  IF timemax < T AND timemin >= T THEN thetamin = 0

  IF thetat = 0 THEN GOTO 200
  IF thetasource + thetanot < thetadet THEN
   timefarlo = SQR((Zs - Zeff * COS(thetat)) ^ 2 + (Xs - Zeff * SIN(the-
tat)) ^ 2) / vi + SQR((Zd - Zeff * COS(thetat)) ^ 2 + (Zeff * SIN(thetat))
^ 2) / vi
   IF timefarlo <= T THEN GOTO 200       ' upscatter limits surface area
   rs = SQR((Zs - Zeff * COS(thetat)) ^ 2 + (Xs - Zeff * SIN(thetat)) ^ 2)
   E = emid(1)
   CALL Atmoseff
   timecheck = SQR((Zs - Zeff * COS(thetat)) ^ 2 + (Xs - Zeff * SIN(the-
tat)) ^ 2) / vmax + SQR((Zd - Zeff * COS(thetat)) ^ 2 + (Zeff *
SIN(thetat)) ^ 2) / vi
  END IF

  IF thetat = thetadet THEN
   timefarlo = (Zsi + 6371000) * SIN(thetasource) / vi + Zd * SIN(theta-
det) / vi
   IF timefarlo <= T THEN GOTO 200
   rs = SQR((Zs - Zeff * COS(thetadet)) ^ 2 + (Xs - Zeff * SIN(thetadet))
^ 2)
   E = emid(1)
   CALL Atmoseff
   timecheck = SQR((Zs - Zeff * COS(thetadet)) ^ 2 + (Xs - Zeff *
SIN(thetadet)) ^ 2) / vmax + Zd * SIN(thetadet) / vi
  END IF

  IF timecheck < T THEN thetamax = thetat
  ' Bisection sequence to determine the maximum angle in theta
  IF timecheck >= T THEN
    t1 = thetat
    t2 = 0
   DO
    t3 = .5 * (t1 + t2)
    Zeff = 6441000
    rs = SQR((Zs - Zeff * COS(t3)) ^ 2 + (Xs - Zeff * SIN(t3)) ^ 2)
    rd = SQR(Zd ^ 2 + Zeff ^ 2 - 2 * Zeff * Zd * COS(t3))
    Thetainc = Thetai(Zeff, rs)
    E = emid(1)
    CALL Atmoseff
    rs = SQR((Zs - Zeff * COS(t3)) ^ 2 + (Xs - Zeff * SIN(t3)) ^ 2)
    rd = SQR(Zd ^ 2 + Zeff ^ 2 - 2 * Zeff * Zd * COS(t3))
    Tprime = rs / vmax + rd / vi
    delta = Tprime - T
    IF delta > .001 THEN t1 = t3
    IF delta < -.001 THEN t2 = t3
   LOOP UNTIL ABS(delta) < .001
    thetamax = .95 * t3
  END IF
  timingmin = ((Zdi + 6371000) * SIN(thetasource)) / vi + SQR((Zd - Zeff
* COS(thetasource - thetanot)) ^ 2 + (Zeff * SIN(thetasource - thetanot))
^ 2) / vi

  IF timemin < T AND timingmin < T THEN
   Zeff = 6441000
   rs = (Zdi + 6371000) * SIN(thetasource)
   rd = vi * T - rs
   costhetamin = (6441000 ^ 2 + Zd ^ 2 - rd ^ 2) / (2 * 6441000 * Zd)
   IF ABS(costhetamin) < 1 THEN
    thetamin = (pi / 2) - ATN(costhetamin / SQR(1 - costhetamin ^ 2))
   ELSE thetamin = thetasource - thetanot
   END IF
  ELSEIF timemin < T AND timingmin > T THEN     ' Bisection for
mimimum theta
   t1 = thetasource - thetanot
   t2 = 0
   DO
    t3 = .5 * (t1 + t2)
    Zeff = 6441000
    rs = SQR((Zs - Zeff * COS(t3)) ^ 2 + (Xs + Zeff * SIN(t3)) ^ 2)
    rd = SQR(Zd ^ 2 + Zeff ^ 2 - 2 * Zeff * Zd * COS(t3))
    Thetainc = Thetai(Zeff, rs)
    E = emid(i)
    CALL Atmoseff
    rs = SQR((Zs - Zeff * COS(t3)) ^ 2 + (Xs + Zeff * SIN(t3)) ^ 2)
    rd = SQR(Zd ^ 2 + Zeff ^ 2 - 2 * Zeff * Zd * COS(t3))
    Tprime = rs / vi + rd / vi
    delta = Tprime - T
    IF delta > .001 THEN t1 = t3
    IF delta < -.001 THEN t2 = t3
   LOOP UNTIL ABS(delta) < .001 OR t3 < .01 OR t3 > .95 * (thetasource
- thetanot)
   IF t3 < thetamax THEN thetamin = t3
   IF t3 >= thetamax THEN thetat = 0
  END IF
END IF

' CASE 3b

IF thetanot > 0 AND thetanot > thetasource THEN
 thetamin = thetanot - thetasource
END IF


' CASE 2
```

```
IF thetasource + thetadet < thetanot THEN
PRINT "Detector too far away - try again"
STOP
END IF

thetat = thetamax - thetamin

nmu = 6

REDIM m(nmu)
REDIM mu(nmu - 1)

conmu = (COS(thetamin) - COS(thetamax)) / (nmu - 1)      '  constant
cos(theta) values

' Determines six theta values defining five panels
m(1) = thetamax
 FOR j = 2 TO nmu - 1
   var = COS(thetamax) + (j - 1) * conmu
   m(j) = (pi / 2) - ATN(var / SQR(1 - var ^ 2))
   NEXT j
m(nmu) = thetamin

' Determines the midpoint values for each panel
FOR index = 1 TO nmu - 1
 mu(index) = .5 * (m(index) + m(index + 1))
NEXT index

200 END SUB

'                                Subroutine Angleomega
SUB Angleomega STATIC
SHARED Zs, Zsi, Zd, Xs, nomega, omegat, omega(), j, mu(), thetamin, vi
SHARED T, emid(), thetanot, thetasource, thetadet, thetamax, vmax, i
CONST pi = 3.141592653#


' CASE 1

IF thetanot = 0 THEN
 omegat - pi
omegamax = pi
 omegamin = 0
END IF

IF thetanot > 0 THEN
```

```
 ' CASE 3

Zeff = 6441000

' First to calculate the angle in theta between the detector pole and the
'   furthest allowable distance from the source pole ("behind the
detector")
'   as this is the limit for using the cosine law for spherical triangles
'   to solve for the maximum omega angle

IF thetasource - thetanot < thetadet THEN thetaback = ABS(thetasource -
thetanot)
IF thetasource - thetanot >= thetadet THEN thetaback = thetadet

IF mu(j) > thetaback THEN
 ' Law of Cosines for spherical triangles
 cosomega = ((Zeff / (Zsi + 6371000)) - COS(thetanot) * COS(mu(j))) /
(SIN(thetanot) * SIN(mu(j)))
 omegamax = (pi / 2) - ATN(cosomega / SQR(1 - cosomega ^ 2))
ELSE omegamax = pi
END IF
IF omegamax > pi THEN omegamax = pi

 ' Check against time-of-flight constraints
rs = SQR((Zs - Zeff * COS(mu(j))) ^ 2 + (Xs - Zeff * SIN(mu(j)) * COS(o-
megamax)) ^ 2 + (Zeff * SIN(mu(j)) * SIN(omegamax)) ^ 2)
rd = SQR((Zd - Zeff * COS(mu(j))) ^ 2 + (Zeff * SIN(mu(j)) *
COS(omegamax)) ^ 2 + (Zeff * SIN(mu(j)) * SIN(omegamax)) ^ 2)
T1prime = rs / vmax + rd / vi

' Bisection sequence to solve for the maximum angle in omega
IF T1prime >= T THEN
t1 = omegamax
t2 = 0
 DO
  t3 = .5 * (t1 + t2)
  Zeff = 6441000
  rs = SQR((Zs - Zeff * COS(mu(j))) ^ 2 + (Xs - Zeff * SIN(mu(j)) *
COS(t3)) ^ 2 + (Zeff * SIN(mu(j)) * SIN(t3)) ^ 2)
  rd = SQR((Zd - Zeff * COS(mu(j))) ^ 2 + (Zeff * SIN(mu(j)) * COS(t3))
^ 2 + (Zeff * SIN(mu(j)) * SIN(t3)) ^ 2)
  Thetainc = Thetai(Zeff, rs)
  E = emid(i)
  CALL Atmoseff
  rs = SQR((Zs - Zeff * COS(mu(j))) ^ 2 + (Xs - Zeff * SIN(mu(j)) *
COS(t3)) ^ 2 + (Zeff * SIN(mu(j)) * SIN(t3)) ^ 2)
  rd = SQR((Zd - Zeff * COS(mu(j))) ^ 2 + (Zeff * SIN(mu(j)) * COS(t3))
^ 2 + (Zeff * SIN(mu(j)) * SIN(t3)) ^ 2)
  Tprime = rs / vmax + rd / vi
  delta = T - Tprime
   IF delta > .001 THEN t2 = t3
   IF delta < -.001 THEN t1 = t3
 LOOP UNTIL ABS(delta) < .001 OR t3 < .01 OR t3 > .98 * omegamax
 omegamax = t3
END IF

Zeff = 6441000
timecheck = SQR((Zs - Zeff * COS(mu(j))) ^ 2 + (Xs - Zeff * SIN(mu(j)))
^ 2) / vi + SQR((Zd - Zeff * COS(mu(j))) ^ 2 + (Zeff * SIN(mu(j))) ^ 2) /
vi
IF timecheck >= T THEN omegamin = 0

' Bisection sequence to solve for the minimum angle in omega
IF timecheck < T THEN
 t1 = omegamax
 t2 = 0
 DO
  t3 = .5 * (t1 + t2)
'  Zeff = 6441000
  rs = SQR((Zs - Zeff * COS(mu(j))) ^ 2 + (Xs - Zeff * SIN(mu(j)) *
COS(t3)) ^ 2 + (Zeff * SIN(mu(j)) * SIN(t3)) ^ 2)
  rd = SQR((Zd - Zeff * COS(mu(j))) ^ 2 + (Zeff * SIN(mu(j)) * COS(t3))
^ 2 + (Zeff * SIN(mu(j)) * SIN(t3)) ^ 2)
  Thetainc = Thetai(Zeff, rs)
  E = emid(i)
  CALL Atmoseff
  rs = SQR((Zs - Zeff * COS(mu(j))) ^ 2 + (Xs - Zeff * SIN(mu(j)) *
COS(t3)) ^ 2 + (Zeff * SIN(mu(j)) * SIN(t3)) ^ 2)
  rd = SQR((Zd - Zeff * COS(mu(j))) ^ 2 + (Zeff * SIN(mu(j)) * COS(t3))
^ 2 + (Zeff * SIN(mu(j)) * SIN(t3)) ^ 2)
  Tprime = rs / vi + rd / vi
  delta = T - Tprime
   IF delta > .001 THEN t2 = t3
   IF delta < -.001 THEN t1 = t3
 LOOP UNTIL ABS(delta) < .001 OR t3 < .01 OR t3 > .98 * omegamax
 omegamin = t3
END IF

END IF

omegat = omegamax - omegamin

nomega = 6
```

```
REDIM o(nomega)
REDIM omega(nomega - 1)

' Now to set up increments of omega

conomega = omegat / (nomega - 1)
o(1) = omegamax

FOR kk = 2 TO nomega - 1
 o(kk) = o(kk - 1) - conomega
NEXT kk

o(nomega) = omegamin

FOR index = 1 TO nomega - 1
 omega(index) = .5 * (o(index) + o(index + 1))      ' midpoint values
NEXT index

100 END SUB

'                              Subroutine Atmoseff
SUB Atmoseff STATIC
SHARED Zeff, E, Thetainc

'  Thetainc is the angle of incidence
'  Zeff is the altitude of the reflective surface
'  grav is the acceleration due to gravity
'  MI is the mass integral
'  Press is the atmospheric pressure
'  Temp is the atmospheric temperature

grav = 970
MI = -LOG(.5) / (.04125 * Xsection(E))
P = grav * (COS(Thetainc) * MI) + 49.44
Press = P * .1

SELECT CASE Press

 CASE IS < 66.94

    Temp = 270.65 * (Press / 66.94) ^ (.0028 / .034163195#)
    Zeff = 6371000 + ((270.65 - Temp) / .0028) + 51413

 CASE 66.94 TO 110.9

    Temp = 270.65
    Zeff = 6371000 - LOG(Press / 110.9) * (Temp / .034163195#) + 47350
 CASE 110.9 TO 868.01

    Temp = 228.65 * (Press / 868.01) ^ (-.0028 / .034163195#)
    Zeff = 6371000 + ((Temp - 228.65) / .0028) + 32162

 CASE 868.01 TO 5474.8

    Temp = 216.65 * (Press / 5474.8) ^ (-.001 / .034163195#)
    Zeff = 6371000 + ((Temp - 216.65) / .001) + 20063

 CASE ELSE

    Zeff = 20063 + 6371000

END SELECT

END SUB

'                                 Subroutine Bin
SUB Bin (E, Eg) STATIC
SHARED Energy(), ngroup

FOR ii = 1 TO ngroup - 1
  IF E > Energy(ii + 1) AND E <= Energy(ii) THEN
  Eg = ii
  END IF
NEXT ii
  IF E < Energy(ngroup) THEN Eg = ngroup
  IF E > Energy(1) THEN Eg = 0

END SUB

'                            Subroutine Initialize
SUB Initialize STATIC
SHARED Zs, Zd, Zsi, Zdi, thetanot, ngroup, Energy(), source(), emid(),
Yield
SHARED A14(), A9(), A4(), A2(), A1(), D75(), D45(), D15(), G75(), G45(),
G15()


' get name of input file
 INPUT "Please type the name of the input file (Drive:Name):"; aa$
 OPEN "i", 1, aa$
```

```
' Zs is source altitude above earth's surface
' Zd is detector altitude above earth's surface
' thetanot is earth centered angle of separation
' T is time since emission
' Yield is multiplication factor for source intensity
' ngroup is number of bins in the spectrum

 INPUT #1, Zsi, Zdi, thetanot, T, Yield
 INPUT #1, ngroup, lowbound

 Xd = 0
 Yd = 0
 Zd = Zdi + 6371000
 Xs = Zs * SIN(thetanot)
 Ys = 0
 Zs = (Zsi + 6371000) * COS(thetanot)

REDIM Energy(ngroup), source(ngroup), emid(ngroup)
REDIM A14(35, 6), A9(35, 6), A4(35, 6), A2(35, 6), A1(35, 6)
REDIM D75(20, 3), D45(20, 3), D15(20, 3), G75(10, 3), G45(10, 3), G15(10, 3)

 FOR i = 1 TO ngroup
  INPUT #1, Energy(i), source(i)
 NEXT i

 FOR i = 2 TO ngroup
  emid(i - 1) = Energy(i - 1) - (Energy(i - 1) - Energy(i)) / 2
 NEXT i
  emid(ngroup) = (Energy(ngroup) - lowbound) / 2

 INPUT #1, a$
  FOR i = 35 TO 1 STEP -1
  FOR j = 1 TO 6
    INPUT #1, A14(i, j)
  NEXT j
  NEXT i
 INPUT #1, b$
  FOR i = 35 TO 8 STEP -1
  FOR j = 1 TO 6
    INPUT #1, A9(i, j)
  NEXT j
  NEXT i
 INPUT #1, c$
  FOR i = 35 TO 13 STEP -1
  FOR j = 1 TO 6
    INPUT #1, A4(i, j)
  NEXT j
 NEXT i
INPUT #1, D$
 FOR i = 35 TO 18 STEP -1
 FOR j = 1 TO 6
   INPUT #1, A2(i, j)
 NEXT j
 NEXT i
INPUT #1, E$
 FOR i = 35 TO 21 STEP -1
 FOR j = 1 TO 6
   INPUT #1, A1(i, j)
 NEXT j
 NEXT i
INPUT #1, f$
 FOR i = 1 TO 20
 FOR j = 1 TO 3
  INPUT #1, D75(i, j)
 NEXT j
 NEXT i
INPUT #1, G$
 FOR i = 1 TO 20
 FOR j = 1 TO 3
  INPUT #1, D45(i, j)
 NEXT j
 NEXT i
INPUT #1, H$
 FOR i = 1 TO 20
 FOR j = 1 TO 3
  INPUT #1, D15(i, j)
 NEXT j
 NEXT i
INPUT #1, i$
 FOR i = 1 TO 10
 FOR j = 1 TO 3
  INPUT #1, G75(i, j)
 NEXT j
 NEXT i
INPUT #1, j$
 FOR i = 1 TO 10
 FOR j = 1 TO 3
  INPUT #1, G45(i, j)
 NEXT j
 NEXT i
INPUT #1, K$
 FOR i = 1 TO 10
 FOR j = 1 TO 3
```

```
   INPUT #1, G15(i, j)
  NEXT j
  NEXT i

CLOSE 1

END SUB

'                                Function Integrate
SUB Integrate (ff(), dist, nangle, gg) STATIC

sum = 0

ds = dist / nangle

FOR kk = 1 TO nangle
 sum = sum + ff(kk)
NEXT kk

gg = sum * ds

END SUB

'                                 Subroutine Predict
SUB Predict STATIC
SHARED emid(), mu(), omega(), Thetainc, Zeff, j, K, E, i, thetanot
SHARED Xd, Yd, Zd, Xs, Ys, Zs, Zsi, T, rd, rs, theta, dEdT, phi
CONST pi = 3.141592654#

' The grid location on the surface of the atmosphere is

x = Zeff * SIN(mu(j)) * COS(omega(K))
y = Zeff * SIN(mu(j)) * SIN(omega(K))
z = Zeff * COS(mu(j))

' The distances to this grid point are

rd = SQR((Xd - x) ^ 2 + (Yd - y) ^ 2 + (Zd - z) ^ 2)
rs = SQR((Xs - x) ^ 2 + (Ys - y) ^ 2 + (Zs - z) ^ 2)

' The time remaining from the atmosphere to the source is

ts = T - rd / SQR((2 * emid(i) * 1.602E-13) / 1.67482E-27)

IF ts <= 0 THEN
  E = 0
END IF
```

```
IF ts > 0 THEN
 vs = rs / ts
 E = (.5 * 1.67482E-27 * vs ^ 2) / 1.602E-13       ' the corresponding
energy

 ' Next to determine the time - energy transform

 dEdT = (1.67482E-27 * vs ^ 2) / (1.602E-13 * ts)

 ' Next to determine the angle of incidence

 u = ((Zsi + 6371000) ^ 2 - rs ^ 2 - Zeff ^ 2) / (-2 * rs * Zeff)

 IF ABS(u) < 1 THEN
 Thetainc = ABS(ATN(SQR(1 - u ^ 2) / u))
 ELSE Thetainc = 0
 END IF

 ' As well as the azimuthal angle to the target

 cosB = (rs ^ 2 - Zeff ^ 2 - (Zsi + 6371000) ^ 2) / (-2 * Zeff * (Zsi +
6371000))
 IF cosB >= 1 THEN cosB = .99999
 sinB = SIN(pi / 2 - ATN(cosB / SQR(1 - cosB ^ 2)))
   ' Law of Cosines for spherical triangles
 IF mu(j) > 0 THEN
  cosphi = (COS(thetanot) - cosB * COS(mu(j))) / (sinB * SIN(mu(j)))
  IF ABS(cosphi) < 1 THEN phi = pi / 2 + ATN(cosphi / SQR(1 - cosphi ^
2))
  IF ABS(cosphi) >= 1 THEN phi = pi / 2
 ELSE
  phi = pi / 4
 END IF
 IF x = 0 THEN phi = pi / 2
 IF y = 0 AND x > Xs THEN phi = pi
 IF y = 0 AND x < Xs THEN phi = 0
 IF y = 0 AND x < 0 THEN phi = pi

 ' As well as the elevation angle to the target

 yy = (Zd ^ 2 - rd ^ 2 - Zeff ^ 2) / (-2 * rd * Zeff)

 IF ABS(yy) < 1 THEN
 theta = (pi / 2) - ABS(ATN(yy / SQR(1 - yy ^ 2)))
 ELSE theta = pi / 2
 END IF
 IF x = 0 THEN theta = pi / 2
 IF phi >= pi / 2 THEN theta = pi - theta

END IF

END SUB

'                                        Function Prob
FUNCTION Prob (Eg, Egprime, Thetainc, theta, phi) STATIC
SHARED ngroup, A14(), A9(), A4(), A2(), A1(), emid(), Energy()
SHARED D75(), D45(), D15(), G75(), G45(), G15()
pi = 3.141592654#

'  Probe - the energy bin-to-bin probability
'  Probt - the elevation angle probability
'  Probp - the azimuthal angle probaility


IF Eg <= Egprime AND Eg > 0 THEN

 ' This determines whether the albedo energy is within the hump

 IF Eg <= 13 THEN Ebreak = emid(Eg) * (.6 + .025 * (emid(Eg) - 14.55))
 IF Eg > 13 THEN Ebreak = emid(Eg) * (.25 - .023 * emid(Eg))
 CALL Bin(Ebreak, Egbreak)

 ' If the albedo energy is after the hump, then the cosine distribution
 '   of the third bin is used as the angle distribution
 IF Egprime > Egbreak THEN

    SELECT CASE Thetainc

     CASE IS >= 1.047

      IF COS(theta) <= -.995 THEN Probt = D75(1, 3)
      IF COS(theta) <= -.98 AND COS(theta) > -.995 THEN Probt = D75(2,
3)
      IF COS(theta) <= -.95 AND COS(theta) > -.98 THEN Probt = D75(3,
3)
      IF COS(theta) <= -.916 AND COS(theta) > -.95 THEN Probt = D75(4,
3)
      IF COS(theta) <= -.866 AND COS(theta) > -.916 THEN Probt = D75(5,
3)
      IF COS(theta) <= -.8 AND COS(theta) > -.866 THEN Probt = D75(6,
3)
      IF COS(theta) <= -.714 AND COS(theta) > -.8 THEN Probt = D75(7,
3)
        IF COS(theta) <= -.52 AND COS(theta) > -.714 THEN Probt = D75(8,
3)
        IF COS(theta) <= -.223 AND COS(theta) > -.52 THEN Probt = D75(9,
3)
        IF COS(theta) <= 0 AND COS(theta) > -.223 THEN Probt = D75(10, 3)
        IF COS(theta) <= .223 AND COS(theta) > 0 THEN Probt = D75(11, 3)
        IF COS(theta) <= .52 AND COS(theta) > .223 THEN Probt = D75(12,
3)
        IF COS(theta) <= .714 AND COS(theta) > .52 THEN Probt = D75(13,
3)
        IF COS(theta) <= .8 AND COS(theta) > .714 THEN Probt = D75(14, 3)
        IF COS(theta) <= .866 AND COS(theta) > .8 THEN Probt = D75(15, 3)
        IF COS(theta) <= .916 AND COS(theta) > .866 THEN Probt = D75(16,
3)
        IF COS(theta) <= .95 AND COS(theta) > .916 THEN Probt = D75(17,
3)
        IF COS(theta) <= .98 AND COS(theta) > .95 THEN Probt = D75(18, 3)
        IF COS(theta) <= .995 AND COS(theta) > .98 THEN Probt = D75(19,
3)
        IF COS(theta) <= 1 AND COS(theta) > .995 THEN Probt = D75(20, 3)
        IF ABS(COS(phi)) >= .995 THEN Probp = G75(1, 3)
        IF ABS(COS(phi)) >= .98 AND ABS(COS(phi)) < .995 THEN Probp =
G75(2, 3)
        IF ABS(COS(phi)) >= .95 AND ABS(COS(phi)) < .98 THEN Probp =
G75(3, 3)
        IF ABS(COS(phi)) >= .916 AND ABS(COS(phi)) < .95 THEN Probp =
G75(4, 3)
        IF ABS(COS(phi)) >= .866 AND ABS(COS(phi)) < .916 THEN Probp =
G75(5, 3)
        IF ABS(COS(phi)) >= .8 AND ABS(COS(phi)) < .866 THEN Probp =
G75(6, 3)
        IF ABS(COS(phi)) >= .714 AND ABS(COS(phi)) < .8 THEN Probp =
G75(7, 3)
        IF ABS(COS(phi)) >= .52 AND ABS(COS(phi)) < .714 THEN Probp =
G75(8, 3)
        IF ABS(COS(phi)) >= .233 AND ABS(COS(phi)) < .52 THEN Probp =
G75(9, 3)
        IF ABS(COS(phi)) >= 0 AND ABS(COS(phi)) < .233 THEN Probp =
G75(10, 3)

       CASE .5236 TO 1.047

        IF COS(theta) <= -.995 THEN Probt = D45(1, 3)
        IF COS(theta) <= -.98 AND COS(theta) > -.995 THEN Probt = D45(2,
3)
        IF COS(theta) <= -.95 AND COS(theta) > -.98 THEN Probt = D45(3,
3)
        IF COS(theta) <= -.916 AND COS(theta) > -.95 THEN Probt = D45(4,
3)
        IF COS(theta) <= -.866 AND COS(theta) > -.916 THEN Probt = D45(5,
3)
        IF COS(theta) <= -.8 AND COS(theta) > -.866 THEN Probt = D45(6,
3)
        IF COS(theta) <= -.714 AND COS(theta) > -.8 THEN Probt = D45(7,
3)
        IF COS(theta) <= -.52 AND COS(theta) > -.714 THEN Probt = D45(8,
3)
        IF COS(theta) <= -.223 AND COS(theta) > -.52 THEN Probt = D45(9,
3)
        IF COS(theta) <= 0 AND COS(theta) > -.223 THEN Probt = D45(10, 3)
        IF COS(theta) <= .223 AND COS(theta) > 0 THEN Probt = D45(11, 3)
        IF COS(theta) <= .52 AND COS(theta) > .223 THEN Probt = D45(12,
3)
        IF COS(theta) <= .714 AND COS(theta) > .52 THEN Probt = D45(13,
3)
        IF COS(theta) <= .8 AND COS(theta) > .714 THEN Probt = D45(14, 3)
        IF COS(theta) <= .866 AND COS(theta) > .8 THEN Probt = D45(15, 3)
        IF COS(theta) <= .916 AND COS(theta) > .866 THEN Probt = D45(16,
3)
        IF COS(theta) <= .95 AND COS(theta) > .916 THEN Probt = D45(17,
3)
        IF COS(theta) <= .98 AND COS(theta) > .95 THEN Probt = D45(18, 3)
        IF COS(theta) <= .995 AND COS(theta) > .98 THEN Probt = D45(19,
3)
        IF COS(theta) <= 1 AND COS(theta) > .995 THEN Probt = D45(20, 3)
        IF ABS(COS(phi)) >= .995 THEN Probp = G45(1, 3)
        IF ABS(COS(phi)) >= .98 AND ABS(COS(phi)) < .995 THEN Probp =
G45(2, 3)
        IF ABS(COS(phi)) >= .95 AND ABS(COS(phi)) < .98 THEN Probp =
G45(3, 3)
        IF ABS(COS(phi)) >= .916 AND ABS(COS(phi)) < .95 THEN Probp =
G45(4, 3)
        IF ABS(COS(phi)) >= .866 AND ABS(COS(phi)) < .916 THEN Probp =
G45(5, 3)
        IF ABS(COS(phi)) >= .8 AND ABS(COS(phi)) < .866 THEN Probp =
G45(6, 3)
        IF ABS(COS(phi)) >= .714 AND ABS(COS(phi)) < .8 THEN Probp =
G45(7, 3)
        IF ABS(COS(phi)) >= .52 AND ABS(COS(phi)) < .714 THEN Probp =
G45(8, 3)
        IF ABS(COS(phi)) >= .233 AND ABS(COS(phi)) < .52 THEN Probp =
G45(9, 3)
      IF ABS(COS(phi)) >= 0 AND ABS(COS(phi)) < .233 THEN Probp =
G45(10, 3)


     CASE 0 TO .5236

      IF COS(theta) <= -.995 THEN Probt = D15(1, 3)
      IF COS(theta) <= -.98 AND COS(theta) > -.995 THEN Probt = D15(2,
3)
      IF COS(theta) <= -.95 AND COS(theta) > -.98 THEN Probt = D15(3,
3)
      IF COS(theta) <= -.916 AND COS(theta) > -.95 THEN Probt = D15(4,
3)
      IF COS(theta) <= -.866 AND COS(theta) > -.916 THEN Probt = D15(5,
3)
      IF COS(theta) <= -.8 AND COS(theta) > -.866 THEN Probt = D15(6,
3)
      IF COS(theta) <= -.714 AND COS(theta) > -.8 THEN Probt = D15(7,
3)
      IF COS(theta) <= -.52 AND COS(theta) > -.714 THEN Probt = D15(8,
3)
      IF COS(theta) <= -.223 AND COS(theta) > -.52 THEN Probt = D15(9,
3)
      IF COS(theta) <= 0 AND COS(theta) > -.223 THEN Probt = D15(10, 3)
      IF COS(theta) <= .223 AND COS(theta) > 0 THEN Probt = D15(11, 3)
      IF COS(theta) <= .52 AND COS(theta) > .223 THEN Probt = D15(12,
3)
      IF COS(theta) <= .714 AND COS(theta) > .52 THEN Probt = D15(13,
3)
      IF COS(theta) <= .8 AND COS(theta) > .714 THEN Probt = D15(14, 3)
      IF COS(theta) <= .866 AND COS(theta) > .8 THEN Probt = D15(15, 3)
      IF COS(theta) <= .916 AND COS(theta) > .866 THEN Probt = D15(16,
3)
      IF COS(theta) <= .95 AND COS(theta) > .916 THEN Probt = D15(17,
3)
      IF COS(theta) <= .98 AND COS(theta) > .95 THEN Probt = D15(18, 3)
      IF COS(theta) <= .995 AND COS(theta) > .98 THEN Probt = D15(19,
3)
      IF COS(theta) <= 1 AND COS(theta) > .995 THEN Probt = D15(20, 3)
      IF ABS(COS(phi)) >= .995 THEN Probp = G15(1, 3)
      IF ABS(COS(phi)) >= .98 AND ABS(COS(phi)) < .995 THEN Probp =
G15(2, 3)
      IF ABS(COS(phi)) >= .95 AND ABS(COS(phi)) < .98 THEN Probp =
G15(3, 3)
      IF ABS(COS(phi)) >= .916 AND ABS(COS(phi)) < .95 THEN Probp =
G15(4, 3)
```

```
      IF ABS(COS(phi)) >= .866 AND ABS(COS(phi)) < .916 THEN Probp =
G15(5, 3)
      IF ABS(COS(phi)) >= .8 AND ABS(COS(phi)) < .866 THEN Probp =
G15(6, 3)
      IF ABS(COS(phi)) >= .714 AND ABS(COS(phi)) < .8 THEN Probp =
G15(7, 3)
      IF ABS(COS(phi)) >= .52 AND ABS(COS(phi)) < .714 THEN Probp =
G15(8, 3)
      IF ABS(COS(phi)) >= .233 AND ABS(COS(phi)) < .52 THEN Probp =
G15(9, 3)
      IF ABS(COS(phi)) >= 0 AND ABS(COS(phi)) < .233 THEN Probp =
G15(10, 3)


     END SELECT

     Probt = Probt * 20 * 4 * pi       ' normalized by dividing by 1/(4
pi)
     Probp = Probp * 20 * 4 * pi       ' normalized by dividing by 1/(4
pi)

 END IF

 ' If the albedo angle is within the hump, then;
 IF Egprime <= Egbreak THEN

 ' This determines which of the three data bins to use from the 1.8-1.1
MeV
 '    angle distribution from MCNP

  here = 1.8 * (emid(Egprime) - Energy(Egbreak + 1)) / (emid(Eg) -
Energy(Egbreak + 1))
  IF here > .16 THEN CALL Bin(here, Egwhere)
  IF here <= .16 THEN Egwhere = 20
  Eghere = Egwhere - 17

    SELECT CASE Thetainc

     CASE IS >= 1.047

      IF COS(theta) <= -.995 THEN Probt = D75(1, Eghere)
      IF COS(theta) <= -.98 AND COS(theta) > -.995 THEN Probt = D75(2,
Eghere)
      IF COS(theta) <= -.95 AND COS(theta) > -.98 THEN Probt = D75(3,
Eghere)
      IF COS(theta) <= -.916 AND COS(theta) > -.95 THEN Probt = D75(4,
Eghere)
```

```
      IF COS(theta) <= -.866 AND COS(theta) > -.916 THEN Probt = D75(5,
Eghere)
      IF COS(theta) <= -.8 AND COS(theta) > -.866 THEN Probt = D75(6,
Eghere)
      IF COS(theta) <= -.714 AND COS(theta) > -.8 THEN Probt = D75(7,
Eghere)
      IF COS(theta) <= -.52 AND COS(theta) > -.714 THEN Probt = D75(8,
Eghere)
      IF COS(theta) <= -.223 AND COS(theta) > -.52 THEN Probt = D75(9,
Eghere)
      IF COS(theta) <= 0 AND COS(theta) > -.223 THEN Probt = D75(10,
Eghere)
      IF COS(theta) <= .223 AND COS(theta) > 0 THEN Probt = D75(11,
Eghere)
      IF COS(theta) <= .52 AND COS(theta) > .223 THEN Probt = D75(12,
Eghere)
      IF COS(theta) <= .714 AND COS(theta) > .52 THEN Probt = D75(13,
Eghere)
      IF COS(theta) <= .8 AND COS(theta) > .714 THEN Probt = D75(14,
Eghere)
      IF COS(theta) <= .866 AND COS(theta) >  8 THEN Probt = D75(15,
Eghere)
      IF COS(theta) <= .916 AND COS(theta) > .866 THEN Probt = D75(16,
Eghere)
      IF COS(theta) <= .95 AND COS(theta) > .916 THEN Probt = D75(17,
Eghere)
      IF COS(theta) <= .98 AND COS(theta) > .95 THEN Probt = D75(18,
Eghere)
      IF COS(theta) <= .995 AND COS(theta) > .98 THEN Probt = D75(19,
Eghere)
      IF COS(theta) <= 1 AND COS(theta) > .995 THEN Probt = D75(20,
Eghere)
      IF ABS(COS(phi)) >= .995 THEN Probp = G75(1, Eghere)
      IF ABS(COS(phi)) >= .98 AND ABS(COS(phi)) < .995 THEN Probp =
G75(2, Eghere)
      IF ABS(COS(phi)) >= .95 AND ABS(COS(phi)) < .98 THEN Probp =
G75(3, Eghere)
      IF ABS(COS(phi)) >= .916 AND ABS(COS(phi)) < .95 THEN Probp =
G75(4, Eghere)
      IF ABS(COS(phi)) >= .866 AND ABS(COS(phi)) < .916 THEN Probp =
G75(5, Eghere)
      IF ABS(COS(phi)) >= .8 AND ABS(COS(phi)) < .866 THEN Probp =
G75(6, Eghere)
      IF ABS(COS(phi)) >= .714 AND ABS(COS(phi)) < .8 THEN Probp =
G75(7, Eghere)
      IF ABS(COS(phi)) >= .52 AND ABS(COS(phi)) < .714 THEN Probp =
G75(8, Eghere)
```

```
        IF ABS(COS(phi)) >= .233 AND ABS(COS(phi)) < .52 THEN Probp =
G75(9, Eghere)
        IF ABS(COS(phi)) >= 0 AND ABS(COS(phi)) < .233 THEN Probp =
G75(10, Eghere)

       CASE .5236 TO 1.047

        IF COS(theta) <= -.995 THEN Probt = D45(1, Eghere)
        IF COS(theta) <= -.98 AND COS(theta) > -.995 THEN Probt = D45(2,
Eghere)
        IF COS(theta) <= -.95 AND COS(theta) > -.98 THEN Probt = D45(3,
Eghere)
        IF COS(theta) <= -.916 AND COS(theta) > -.95 THEN Probt = D45(4,
Eghere)
        IF COS(theta) <= -.866 AND COS(theta) > -.916 THEN Probt = D45(5,
Eghere)
        IF COS(theta) <= -.8 AND COS(theta) > -.866 THEN Probt = D45(6,
Eghere)
        IF COS(theta) <= -.714 AND COS(theta) > -.8 THEN Probt = D45(7,
Eghere)
        IF COS(theta) <= -.52 AND COS(theta) > -.714 THEN Probt = D45(8,
Eghere)
        IF COS(theta) <= -.223 AND COS(theta) > -.52 THEN Probt = D45(9,
Eghere)
        IF COS(theta) <= 0 AND COS(theta) > -.223 THEN Probt = D45(10,
Eghere)
        IF COS(theta) <= .223 AND COS(theta) > 0 THEN Probt = D45(11,
Eghere)
        IF COS(theta) <= .52 AND COS(theta) > .223 THEN Probt = D45(12,
Eghere)
        IF COS(theta) <= .714 AND COS(theta) > .52 THEN Probt = D45(13,
Eghere)
        IF COS(theta) <= .8 AND COS(theta) > .714 THEN Probt = D45(14,
Eghere)
        IF COS(theta) <= .866 AND COS(theta) > .8 THEN Probt = D45(15,
Eghere)
        IF COS(theta) <= .916 AND COS(theta) > .866 THEN Probt = D45(16,
Eghere)
        IF COS(theta) <= .95 AND COS(theta) > .916 THEN Probt = D45(17,
Eghere)
        IF COS(theta) <= .98 AND COS(theta) > .95 THEN Probt = D45(18,
Eghere)
        IF COS(theta) <= .995 AND COS(theta) > .98 THEN Probt = D45(19,
Eghere)
        IF COS(theta) <= 1 AND COS(theta) > .995 THEN Probt = D45(20,
Eghere)
        IF ABS(COS(phi)) >= .995 THEN Probp = G45(1, Eghere)
        IF ABS(COS(phi)) >= .98 AND ABS(COS(phi)) < .995 THEN Probp =
G45(2, Eghere)
        IF ABS(COS(phi)) >= .95 AND ABS(COS(phi)) < .98 THEN Probp =
G45(3, Eghere)
        IF ABS(COS(phi)) >= .916 AND ABS(COS(phi)) < .95 THEN Probp =
G45(4, Eghere)
        IF ABS(COS(phi)) >= .866 AND ABS(COS(phi)) < .916 THEN Probp =
G45(5, Eghere)
        IF ABS(COS(phi)) >= .8 AND ABS(COS(phi)) < .866 THEN Probp =
G45(6, Eghere)
        IF ABS(COS(phi)) >= .714 AND ABS(COS(phi)) < .8 THEN Probp =
G45(7, Eghere)
        IF ABS(COS(phi)) >= .52 AND ABS(COS(phi)) < .714 THEN Probp =
G45(8, Eghere)
        IF ABS(COS(phi)) >= .233 AND ABS(COS(phi)) < .52 THEN Probp =
G45(9, Eghere)
        IF ABS(COS(phi)) >= 0 AND ABS(COS(phi)) < .233 THEN Probp =
G45(10, Eghere)


       CASE 0 TO .5236

        IF COS(theta) <= -.995 THEN Probt = D15(1, Eghere)
        IF COS(theta) <= -.98 AND COS(theta) > -.995 THEN Probt = D15(2,
Eghere)
        IF COS(theta) <= -.95 AND COS(theta) > -.98 THEN Probt = D15(3,
Eghere)
        IF COS(theta) <= -.916 AND COS(theta) > -.95 THEN Probt = D15(4,
Eghere)
        IF COS(theta) <= -.866 AND COS(theta) > -.916 THEN Probt = D15(5,
Eghere)
        IF COS(theta) <= -.8 AND COS(theta) > -.866 THEN Probt = D15(6,
Eghere)
        IF COS(theta) <= -.714 AND COS(theta) > -.8 THEN Probt = D15(7,
Eghere)
        IF COS(theta) <= -.52 AND COS(theta) > -.714 THEN Probt = D15(8,
Eghere)
        IF COS(theta) <= -.223 AND COS(theta) > -.52 THEN Probt = D15(9,
Eghere)
        IF COS(theta) <= 0 AND COS(theta) > -.223 THEN Probt = D15(10,
Eghere)
        IF COS(theta) <= .223 AND COS(theta) > 0 THEN Probt = D15(11,
Eghere)
        IF COS(theta) <= .52 AND COS(theta) > .223 THEN Probt = D15(12,
Eghere)
        IF COS(theta) <= .714 AND COS(theta) > .52 THEN Probt = D15(13,
Eghere)
```

```
      IF COS(theta) <= .8 AND COS(theta) > .714 THEN Probt = D15(14,
Eghere)
      IF COS(theta) <= .866 AND COS(theta) > .8 THEN Probt = D15(15,
Eghere)
      IF COS(theta) <= .916 AND COS(theta) > .866 THEN Probt = D15(16,
Eghere)
      IF COS(theta) <= .95 AND COS(theta) > .916 THEN Probt = D15(17,
Eghere)
      IF COS(theta) <= .98 AND COS(theta) > .95 THEN Probt = D15(18,
Eghere)
      IF COS(theta) <= .995 AND COS(theta) > .98 THEN Probt = D15(19,
Eghere)
      IF COS(theta) <= 1 AND COS(theta) > .995 THEN Probt = D15(20,
Eghere)
      IF ABS(COS(phi)) >= .995 THEN Probp = G15(1, Eghere)
      IF ABS(COS(phi)) >= .98 AND ABS(COS(phi)) < .995 THEN Probp =
G15(2, Eghere)
      IF ABS(COS(phi)) >= .95 AND ABS(COS(phi)) < .98 THEN Probp =
G15(3, Eghere)
      IF ABS(COS(phi)) >= .916 AND ABS(COS(phi)) < .95 THEN Probp =
G15(4, Eghere)
      IF ABS(COS(phi)) >= .866 AND ABS(COS(phi)) < .916 THEN Probp =
G15(5, Eghere)
      IF ABS(COS(phi)) >= .8 AND ABS(COS(phi)) < .866 THEN Probp =
G15(6, Eghere)
      IF ABS(COS(phi)) >= .714 AND ABS(COS(phi)) < .8 THEN Probp =
G15(7, Eghere)
      IF ABS(COS(phi)) >= .52 AND ABS(COS(phi)) < .714 THEN Probp =
G15(8, Eghere)
      IF ABS(COS(phi)) >= .233 AND ABS(COS(phi)) < .52 THEN Probp =
G15(9, Eghere)
      IF ABS(COS(phi)) >= 0 AND ABS(COS(phi)) < .233 THEN Probp =
G15(10, Eghere)


     END SELECT

     Probt = Probt * 20 * 4 * pi       ' normalized by dividing by 1/(4
pi)
     Probp = Probp * 20 * 4 * pi       ' normalized by dividing by 1/(4
pi)

  END IF

 ' Now to determine the energy bin probability;
 SELECT CASE Eg
```

```
CASE IS <= 7

 IF Thetainc >= 1.22 THEN Probe = A14(Egprime, 6)
 IF Thetainc < 1.22 AND Thetainc >= .96 THEN Probe = A14(Egprime, 5)
 IF Thetainc < .96 AND Thetainc >= .61 THEN Probe = A14(Egprime, 4)
 IF Thetainc < .61 AND Thetainc >= .35 THEN Probe = A14(Egprime, 3)
 IF Thetainc < .35 AND Thetainc >= .087 THEN Probe = A14(Egprime, 2)
 IF Thetainc < .087 THEN Probe = A14(Egprime, 1)

CASE 8 TO 12

 IF Thetainc >= 1.22 THEN Probe = A9(Egprime, 6)
 IF Thetainc < 1.22 AND Thetainc >= .96 THEN Probe = A9(Egprime, 5)
 IF Thetainc < .96 AND Thetainc >= .61 THEN Probe = A9(Egprime, 4)
 IF Thetainc < .61 AND Thetainc >= .35 THEN Probe = A9(Egprime, 3)
 IF Thetainc < .35 AND Thetainc >= .087 THEN Probe = A9(Egprime, 2)
 IF Thetainc < .087 THEN Probe = A9(Egprime, 1)

CASE 13 TO 17

 IF Thetainc >= 1.22 THEN Probe = A4(Egprime, 6)
 IF Thetainc < 1.22 AND Thetainc >= .96 THEN Probe = A4(Egprime, 5)
 IF Thetainc < .96 AND Thetainc >= .61 THEN Probe = A4(Egprime, 4)
 IF Thetainc < .61 AND Thetainc >= .35 THEN Probe = A4(Egprime, 3)
 IF Thetainc < .35 AND Thetainc >= .087 THEN Probe = A4(Egprime, 2)
 IF Thetainc < .087 THEN Probe = A4(Egprime, 1)

CASE 18 TO 20

 IF Thetainc >= 1.22 THEN Probe = A2(Egprime, 6)
 IF Thetainc < 1.22 AND Thetainc >= .96 THEN Probe = A2(Egprime, 5)
 IF Thetainc < .96 AND Thetainc >= .61 THEN Probe = A2(Egprime, 4)
 IF Thetainc < .61 AND Thetainc >= .35 THEN Probe = A2(Egprime, 3)
 IF Thetainc < .35 AND Thetainc >= .087 THEN Probe = A2(Egprime, 2)
 IF Thetainc < .087 THEN Probe = A2(Egprime, 1)

CASE ELSE

 IF Thetainc >= 1.22 THEN Probe = A1(Egprime, 6)
 IF Thetainc < 1.22 AND Thetainc >= .96 THEN Probe = A1(Egprime, 5)
 IF Thetainc < .96 AND Thetainc >= .61 THEN Probe = A1(Egprime, 4)
 IF Thetainc < .61 AND Thetainc >= .35 THEN Probe = A1(Egprime, 3)
 IF Thetainc < .35 AND Thetainc >= .087 THEN Probe = A1(Egprime, 2)
 IF Thetainc < .087 THEN Probe = A1(Egprime, 1)
```

```
  END SELECT

  Prob = Probe * Probt * Probp

 END IF

 IF Eg = 0 OR Eg > Egprime THEN Prob = 0

END FUNCTION

'                               Subroutine results
SUB results

SHARED FofT(), T, Zsi, Zdi, thetanot, Energy(), ngroup, ver$, Ftotal

'OPEN "ALBEDO.OUT" FOR APPEND AS 2
PRINT #2, ver$, DATE$, TIME$
PRINT #2,
PRINT #2, "The parameters of this case were;"
PRINT #2,
PRINT #2, "    Source altitude of"; Zsi; "meters"
PRINT #2, "    Detector altitude of"; Zdi; "meters"
PRINT #2, "    Separation angle of"; thetanot; "radians"
PRINT #2,
PRINT #2, "The free field albedo flux at"; T; "seconds is as follows;"
PRINT #2,
PRINT #2, " Upper Energy  Flux "
 FOR i = 1 TO ngroup
  PRINT #2, Energy(i), FofT(i)
 NEXT i
'PRINT #2,
PRINT #2, "The total flux is"; Ftotal
PRINT #2,

END SUB

FUNCTION Thetai (Zeff, rs)
SHARED Zsi

 u = ((Zsi + 6371000) ^ 2 - rs ^ 2 - Zeff ^ 2) / (-2 * rs * Zeff)

 IF ABS(u) < 1 THEN
 Thetai = ABS(ATN(SQR(1 - u ^ 2) / u))
 ELSE Thetai = 0
 END IF

END FUNCTION
```

```
'                                   Subroutine Transport
SUB Transport (Eg, Egprime, W) STATIC
SHARED source(), Energy(), Thetainc, theta, phi
SHARED rd, rs, dEdT
CONST pi = 3.141592654#

'  mainequation refers to the last equation in Section 3.3 of the thesis
'  the 2 / (4 pi) term accounts for a lambertain reflection over the
surface
'  Prob modifies the lambertain distribution and includes f(E->E)

mainequation = (source(Eg) * dEdT) / (16 * pi ^ 2 * rs ^ 2 * rd ^ 2 *
(Energy(Eg) - Energy(Eg + 1)))

W = (2 * mainequation * Prob(Eg, Egprime, Thetainc, theta, phi)) / (4 *
pi)

END SUB

'                                   Function Xsection
FUNCTION Xsection (E) STATIC

IF E <= 18 AND E > 0 THEN
Xsection = 10 - .55 * LOG(E * 1000000)
ELSE
IF E > 18 THEN Xsection = .1
IF E = 0 THEN Xsection = 10
END IF

END FUNCTION
```

INPUT FILE

```
150000 250000 .08 .2 500
35 0.00000000001
14.9          .0189
14.2          .00934
13.8          .0266
12.8          .0167
12.2          .0169
11.1          .0124
10.0          .00748
9.0           .00682
8.2           .00678
7.4           .0103
6.4           .0181
5.0           .00362
4.7           .0124
4.1           .0260
3.0           .0237
2.4           .00375
2.3           .0256
1.8           .0644
1.1           .0885
.55           .0914
.16           .0116
.11           .111
.052          .054
.025          .00568
.022          .0926
.01           .116
.0034         .0738
.0012         .0232
.00058        .0203
.0001         .0019
.000029       0
.000011       0
.0000031      0
.0000011      0
.00000041     0
F14
.00682  .00660  .00671  .00633  .00580  .00453
.00682  .00660  .00671  .00633  .00580  .00453
.00682  .00660  .00671  .00633  .00580  .00453
.00682  .00660  .00671  .00633  .00580  .00453
.00682  .00660  .00671  .00633  .00580  .00453
```

.00682	.00660	.00671	.00633	.00580	.00453
.00291	.00292	.00290	.00242	.00248	.00192
.00125	.00138	.00143	.00147	.00116	.00103
.00224	.00218	.00244	.00256	.00215	.00166
.00314	.00291	.00303	.00309	.00306	.00244
.00265	.00266	.00277	.00265	.00264	.00223
.00051	.00052	.00046	.00047	.00059	.00056
.00373	.00380	.00388	.00353	.00351	.00303
.00544	.00567	.00595	.00562	.00566	.00494
.00404	.00393	.00416	.00437	.00428	.00388
.01987	.02089	.02182	.02290	.02358	.02337
.02839	.02917	.03088	.03120	.03448	.03469
.02747	.02741	.02914	.03068	.03416	.03545
.01698	.01775	.01882	.02049	.02288	.02451
.00372	.00362	.00415	.00414	.00431	.00473
.01721	.01789	.01876	.02110	.02343	.02490
.02147	.02175	.02289	.02592	.02942	.03274
.01098	.01141	.01228	.01339	.01580	.01800
.00633	.00682	.00704	.00752	.00898	.01003
.02305	.02347	.02447	.02636	.03043	.03453
.01016	.01042	.01168	.01362	.01607	.01913
.00353	.00366	.00438	.00566	.00710	.00978
.00210	.00226	.00242	.00291	.00396	.00537
.00337	.00339	.00355	.00431	.00484	.00570
.00989	.00998	.00991	.00989	.00992	.00993
.02293	.02255	.02234	.02328	.02533	.02695
.00415	.00641	.01017	.01325	.01699	.02153
.00061	.00159	.00533	.01329	.02406	.03800
0	0	.00009	.00087	.00484	.01851
0	0	0	.00008	.00133	.01424
F9					
.00676	.00647	.00664	.00604	.00563	.00437
.00676	.00647	.00664	.00604	.00563	.00437
.00676	.00647	.00664	.00604	.00563	.00437
.00676	.00647	.00664	.00604	.00563	.00437
.00676	.00647	.00664	.00604	.00563	.00437
.00676	.00647	.00664	.00604	.00563	.00437
.00265	.00275	.00249	.00250	.00233	.00191
.00132	.00132	.00127	.00117	.00125	.00091
.00214	.00218	.00204	.00214	.00210	.00169
.00271	.00268	.00287	.00336	.00261	.00242
.00247	.00249	.00270	.00281	.00244	.00214
.00048	.00044	.00047	.00055	.00041	.00037
.00322	.00333	.00295	.00314	.00318	.00270
.00538	.00514	.00515	.00517	.00513	.00447
.00356	.00353	.00347	.00343	.00322	.00307
.01864	.01898	.01996	.02007	.02005	.01896

```
.03025  .03116  .03189  .03320  .03526  .03537
.03193  .03305  .03463  .03688  .04214  .04497
.01686  .01710  .01830  .02131  .02502  .02902
.00339  .00332  .00358  .00386  .00446  .00540
.01209  .01230  .01314  .01451  .01710  .02078
.01402  .01446  .01535  .01694  .01956  .02243
.00688  .00726  .00726  .00806  .00914  .00978
.00525  .00526  .00559  .00579  .00636  .00605
.04175  .04173  .04280  .04485  .04781  .04916
.08664  .08768  .09050  .09519  .09994  .10995
.00448  .00809  .01564  .02718  .04474  .06900
0       0        .00012  .00114  .00614  .02509
F4
.00454  .00443  .00447  .00427  .00358  .00283
.00454  .00443  .00447  .00427  .00358  .00283
.00454  .00443  .00447  .00427  .00358  .00283
.00454  .00443  .00447  .00427  .00358  .00283
.00454  .00443  .00447  .00427  .00358  .00283
.00454  .00443  .00447  .00427  .00358  .00283
.00166  .00176  .00173  .00164  .00149  .00119
.00084  .00082  .00089  .00080  .00062  .00053
.00142  .00148  .00140  .00127  .00119  .00103
.00171  .00180  .00180  .00177  .00168  .00113
.00179  .00177  .00162  .00170  .00137  .00126
.00027  .00033  .00032  .00025  .00028  .00032
.00229  .00208  .00238  .00204  .00193  .00158
.00326  .00350  .00332  .00354  .00302  .00274
.00237  .00232  .00234  .00232  .00231  .00217
.01136  .01189  .01222  .01228  .01180  .01059
.01812  .01882  .01897  .01893  .01814  .01481
.02436  .02460  .02575  .02614  .02687  .02363
.02469  .02502  .02605  .02647  .02770  .02535
.00555  .00575  .00584  .00623  .00684  .00665
.03438  .03558  .03878  .04416  .05127  .05700
.17479  .17544  .18106  .19053  .20992  .23912
.00014  .00043  .00141  .00509  .01708  .05196
F2
.01810  .01777  .01812  .01693  .01503  .01127
.01810  .01777  .01812  .01693  .01503  .01127
.01810  .01777  .01812  .01693  .01503  .01127
.01810  .01777  .01812  .01693  .01503  .01127
.01810  .01777  .01812  .01693  .01503  .01127
.01810  .01777  .01812  .01693  .01503  .01127
.00810  .00805  .00763  .00751  .00673  .00530
.00376  .00374  .00348  .00347  .00305  .00249
.00639  .00624  .00675  .00608  .00530  .00427
.00873  .00866  .00844  .00839  .00730  .00595
```

```
.00757  .00791  .00777  .00733  .00687  .00572
.00121  .00151  .00126  .00124  .00126  .00096
.01178  .01060  .01056  .00991  .00895  .00741
.01705  .01713  .01697  .01630  .01483  .01230
.01269  .01252  .01244  .01209  .01044  .00903
.07968  .08153  .08134  .07971  .07664  .06490
.25643  .25857  .26632  .27925  .29059  .29976
.13673  .14304  .15691  .18233  .22521  .29506
F1
.04388  .04348  .04192  .03804  .03409  .02746
.04388  .04348  .04192  .03804  .03409  .02746
.04388  .04348  .04192  .03804  .03409  .02746
.04388  .04348  .04192  .03804  .03409  .02746
.04388  .04348  .04192  .03804  .03409  .02746
.04388  .04348  .04192  .03804  .03409  .02746
.02469  .02409  .02292  .02159  .01915  .01498
.01431  .01380  .01287  .01274  .01076  .00849
.02550  .02513  .02399  .02240  .02037  .01665
.03823  .03907  .03867  .03654  .03241  .02717
.04385  .04518  .04400  .04239  .03910  .03328
.01084  .00990  .00994  .00983  .00879  .00775
.08789  .08850  .08905  .08894  .08461  .07468
.32253  .32726  .33521  .34763  .37244  .39459
.09574  .09876  .11136  .13340  .16814  .22524
D75
.000916  .000976  .000099
.005598  .006035  .000521
.009388  .010562  .001350
.011411  .014042  .002080
.012584  .016717  .002861
.013738  .019229  .003597
.014496  .021197  .004415
.015165  .022397  .005156
.015960  .023970  .006081
.016376  .024766  .006846
.018743  .024443  .006796
.020346  .022352  .005582
.021633  .019997  .005063
.021999  .017473  .004336
.022842  .016221  .003685
.021739  .014403  .002690
.020573  .011025  .002080
.017874  .007643  .001285
.011328  .003901  .000545
.002759  .000609  .000126
D45
.000369  .000398  .000106
```

.002073	.002568	.000512
.004819	.006705	.001434
.007023	.011123	.002280
.008706	.013974	.003446
.009903	.017133	.004362
.011044	.020220	.005422
.011976	.022140	.006461
.012774	.024778	.007204
.013783	.026379	.008782
.013600	.024685	.008653
.013935	.022244	.007630
.013819	.019815	.006369
.013109	.017275	.005192
.011960	.014883	.004222
.010671	.012166	.003294
.009446	.009305	.002386
.007850	.006883	.001459
.004806	.003318	.000660
.001097	.000603	.000137
D15		
.000257	.000290	.000105
.001291	.001698	.000425
.003209	.004871	.001251
.005369	.007876	.002129
.006694	.011965	.003236
.008034	.015073	.004413
.009253	.017985	.005327
.010457	.020214	.006313
.011975	.023099	.00764[illegible]
.012763	.025844	.009160
.012055	.025897	.008546
.011236	.022736	.007689
.010435	.019499	.006519
.009742	.017092	.005809
.008179	.014055	.004480
.007819	.011215	.003258
.006311	.008542	.002494
.004950	.005809	.001588
.002837	.002941	.000665
.000601	.000558	.000158
G75		
.001714	.000756	.000120
.008264	.004835	.000533
.014017	.009011	.001383
.016242	.012697	.002129
.016816	.015522	.002695
.018279	.018122	.003775

.018184	.019376	.004435
.018564	.020860	.005188
.018253	.022755	.005847
.017777	.024360	.006735
G45		
.000810	.000453	.000132
.003410	.003046	.000612
.006398	.006909	.001389
.008235	.009992	.002208
.009853	.012970	.003222
.010913	.015677	.004374
.012205	.018753	.005560
.012960	.020708	.006346
.013358	.023293	.007456
.013568	.025760	.008786
G15		
.000390	.000361	.000126
.002089	.002395	.000589
.004048	.005307	.001334
.005708	.007999	.002291
.007363	.011467	.003255
.007888	.014843	.004519
.009571	.017700	.005421
.010380	.020045	.006385
.011580	.022713	.007740
.012383	.025900	.009027

Bibliography

1. Gerald Goertzel and Nunzio Tralli. "Some Mathematical Methods of Physics", *McGraw-Hill Book Company, Inc.*, 1960

2. RSIC Computer Code Collection CCC-200, "MCNP Version 3B, Monte Carlo Neutron and Photon Transport Code System", March 1989. Contributed by *Los Alamos National Laboratory*. See Los Alamos Technical Publication LA-7396-M.

3. Donald J. Hughes and Robert B. Schartz. "Neutron Cross Sections" Second Edition, *Brookhaven National Laboratories 325*, July 1958

4. David Monti. QuickBasic program "Atmosphere", Based upon U.S. Standard Atmosphere 1976, *Air Force Institute of Technology* 1990

5. William H. Beyer, Ph.D. "Standard Mathematical Tables", 26th Edition, *CRC Press*, 1981

6. Dr. N. Ricky Byrn, *Nichols Research Corporation*, discussions on 6 June 1990.

Vita

Captain Donald R. Culp Jr. was born on 7 August 1962 in Livingston, New Jersey. His family moved to South Florida in 1964 and he graduated from Cooper City High School in 1980. He enrolled in Air Force ROTC while attending the Georgia Institute of Technology in Atlanta, Georgia, and graduated in 1984 with a Bachelor of Nuclear Engineering. Upon graduation, he received a commission in the USAF and was assigned to Headquarters/Electronic Systems Division at Hanscom AFB, Bedford, Massachusetts where he served as a Test Director for the Physical Security Systems Directorate, responsible for test and evaluation of electronic systems which augment existing security forces worldwide in the protection of high value DOD resources. In 1987 he was also assigned to the Technical On-Site Inspection (TOSI) program, which was responsible for the development and integration of verification equipment and facilities for the implementation of the American/Soviet INF Treaty. Captain Culp was a member of the first U.S. inspection team to go to Votkinsk, Union of Soviet Socialist Republics, in July 1988. Captain Culp was assigned to the School of Engineering, Air Force Institute of Technology, in August 1989.

[redacted]

[redacted]

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE 1 MARCH 1991	3. REPORT TYPE AND DATES COVERED MASTER'S THESIS

4. TITLE AND SUBTITLE GREEN"S FUNCTION APPROACH TO THE ATMOSPHERIC ALBEDO NEUTRON TRANSPORT PROBLEM	5. FUNDING NUMBERS
6. AUTHOR(S) DONALD R. CULP JR., CAPTAIN USAF	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) AIR FORCE INSTITUTE OF TECHNOLOGY WPAFB OH 45433-6583	8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GNE/ENP/91M-1
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) HQ DEFENSE NUCLEAR AGENCY/RAAE 6801 TELEGRAPH ROAD, ALEXANDRIA, VA 22310	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED	12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This study investigated the reflection of neutron radiation off of the earth's upper atmosphere, with the goal of generating a quick-running computer algorithm to estimate the albedo free field flux at any point above the atmosphere. This thesis involved analytic development in the construction of the algorithm and employed Monte Carlo simulation for generating the energy and angle distributions of the reflected radiation. The Green's function approach to modeling the neutron transport process involved approximating each energy bin of the source spectrum as a Dirac pulse in energy and summing the contribution from each source bin. The computer program integrates over the surface of the atmosphere and uses the Monte Carlo data to calculate the albedo flux at any specified time and location. Run time was maximum of six minutes for a flux calculation, but a gain on the order of one thousand should be achieved on mainframe computer systems. The albedo flux from an instantaneous point source raises quickly to a maximum and then falls off over time. Albedo fluxes as much as 10(-16) (neutrons/square cm sec/source neutron) were calculated. The accuracy of the algorithm is greatly affected by the fineness of the energy bins involved.

14. SUBJECT TERMS EXOATMOSPHERIC, NEUTRON, REFLECTED, ALBEDO, FLUX, GREEN'S FUNCTION, MONTE CARLO, ALGORITHM	15. NUMBER OF PAGES 135
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT UL