AD 746015

AIR FORCE INSTITUTE OF TECHNOLOGY

AF INSTITUTE OF TECHNOLOGY

AIR UNIVERSITY
UNITED STATES AIR FORCE

release; LAW AFR 190-17 SPONSORING MILITARY ACTIVITY

1st Lt., USAF

nformation

presents a code, developed by the auth
and gamma fluences (or doses), includi
in an exponential atmosphere for burst
0 kilometers. The problem was solved by
on, with separation of virgin and scatte
al coordinate system

SCHOOL OF ENGINEERING

WRIGHT–PATTERSON AIR FORCE BASE, OHIO

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Air Force Institute of Technology Wright-Patterson Air Force Base, Ohio 45433	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

A CODE FOR AIRCRAFT SURVIVABILITY ANALYSIS - GAMMA AND NEUTRON EFFECTS

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

AFIT Thesis

5. AUTHOR(S) (First name, middle initial, last name)

Robert D. McLaren
Captain USAF

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
June 1972	142	10

8a. CONTRACT OR GRANT NO	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO. N/A	GNE/PH/72-8
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES Approved for public release; IAW AFR 190-17 Keith A. Williams, 1st Lt., USAF Acting Director of Information	12. SPONSORING MILITARY ACTIVITY

13. ABSTRACT

This report presents a code, developed by the author, that predicts neutron and gamma fluences (or doses), including neutron induced gammas, in an exponential atmosphere for burst altitudes between 7 and 100 kilometers. The problem was solved by using the diffusion equation, with separation of virgin and scattered particles, in a nonorthogonal coordinate system. The diffusion equation in this coordinate system was approximated by a nine point difference equation and the resulting matrix equation was solved by use of a block tri-diagonal algorithm. The resulting computer code, in FORTRAN Extended, was written to calculate the survivability of up to 100 aircraft or space vehicles in addition to the calculation of fluences (doses). The code requires 20 minutes of central processor time and one hour of input/output time on the Wright-Patterson Air Force Base CDC 6600 computer. The results of this code for a burst at 25 km are compared to those obtained from a constant density atmospheric model and from charts based on SMAUG which employs mass integral scaling. Significant differences are noted for the case where the burst and receiver are at the same altitude, which casts some doubt on the validity of mass integral scaling at this altitude.

DD FORM 1 NOV 65 1473

Unclassified

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Nuclear effects						
Neutron fluence						
Gamma fluence						
Exponential atmosphere						
Aircraft survivability code						

A CODE FOR AIRCRAFT SURVIVABILITY

ANALYSIS - GAMMA AND NEUTRON EFFECTS

THESIS

GNE/PH/72-8

Robert D. McLaren
Captain USAF

GNE/PH/72-8

A CODE FOR AIRCRAFT SURVIVABILITY ANALYSIS - GAMMA AND NEUTRON EFFECTS

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

by

Robert D. McLaren, B.Ch.E.
Captain USAF
Graduate Nuclear Engineering

June 1972

Approved for public release; distribution unlimited.

Preface

This report presents a code that calculates the neutron and gamma fluences (or doses) that result from a nuclear weapon explosion between 7 and 100 kilometers altitude in an exponential atmosphere. The code also determines if aircraft located in the vicinity survive the neutron and gamma effects. The code is listed in Appendix B and the instructions for using the code are in Chapter III.

The purpose of developing this code was to provide a subroutine for gamma and neutron effects to a general nuclear effects survivability/vulnerability code. I was successful in developing such a code; however, the time required to run it does limit its applications. The time required is about 20 minutes central processor time and an hour of input/output time.

Many people, in the course of my efforts to complete this code, provided assistance. These people were from the Air Force Weapons Laboratory (AFWL), Oak Ridge National Laboratory, and the Air Force Institute of Technology (AFIT). To all of you I offer my thanks. I would like to single out a few for special thanks, because without their help, I could not have completed this task. First is Mr. Harry Murphy of AFWL for his talks with me on the code, SMAUG. I borrowed heavily from this code to develop mine. Mr. Robert Roussin of the Radiation Shielding Information Center of Oak Ridge furnished the cross section data I used in the code. Next are two members of the AFIT Mathematics Department, Major John Jones and Dr. Wilhelm Ericksen. Major Jones and I spent considerable time investigating methods of directly solving sparse matrix equations. Dr. Ericksen provided the basis for the nonorthogonal coordinate system I used. Dr. Donn Shankland of the AFIT Physics Department provided further assistance on this coordinate system and removed the last problem area standing in the way of success. I can not, of course, forget my advisor, Dr. Charles Bridgman of the AFIT Physics Department. I hope this report demonstrates that his encouragement and faith in me has been rewarded.

I would also like to thank Mrs. Marge Hockemeier, wife of my classmate John, for typing much of my draft. Finally, I would like to express my appreciation to my wife Bonnie for putting up with me during this hectic period.

Contents

List of Figures

List of Tables

Abstract

This report presents a code, developed by the author, that predicts neutron and gamma fluences (or doses), including neutron induced gammas, in an exponential atmosphere for burst altitudes between 7 and 100 kilometers. The problem was solved by using the diffusion equation, with separation of virgin and scattered particles, in a nonorthogonal coordinate system. The diffusion equation in this coordinate system was approximated by a nine point difference equation and the resulting matrix equation was solved by use of a block tri-diagonal algorithm. The resulting computer code, in FORTRAN Extended, was written to calculate the survivability of up to 100 aircraft or space vehicles in addition to the calculation of fluences (doses). The code requires 20 minutes of central processor time and one hour of input/output time on the Wright-Patterson Air Force Base CDC 6600 computer. The results of this code for a burst at 25 km are compared to those obtained from a constant density atmospheric model and from charts based on SMAUG which employs mass integral scaling. Significant differences are noted for the case where the burst and receiver are at the same altitude, which casts some doubt on the validity of mass integral scaling at this altitude.

A CODE FOR AIRCRAFT SURVIVABILITY ANALYSIS - GAMMA AND NEUTRON EFFECTS

I. Introduction

Numerous attempts have been made to determine the radiation field, neutrons and gammas, from nuclear weapons. Straker, in a report titled Status of Neutron Transport in The Atmosphere (Ref 5) has listed these attempts. He lists a total of 35 attempts of which 21 were based on a constant density, (no variations with altitude, no ground, no clouds) infinite air model. Five attempts were based on an exponential variation in air density and nine were concerned only with the air/ground interface. The method of solution for 26 of the 35 attempts was Monte Carlo computer codes. All of the exponential air models were solved by Monte Carlo codes. Monte Carlo codes normally require hours of computer run time and are therefore relatively expensive calculations.

Recently, two high speed computer codes have been developed to estimate the neutron and gamma radiation dose in the vicinity of an atmospheric nuclear detonation by interpolation of a library of precalculated results. These precalculated results include some of those listed by Straker and others by the code authors. Both codes were presented at the Radiation Transport in Air seminar held at Oak Ridge National Laboratory on 15-17 November 1971. One code, ATR (Ref 10), was developed by Science Applications, Inc. The final documentation of this code, however, has not yet been made available to the Radiation Shielding Information Center (RSIC) of Oak Ridge National Laboratory. The other code, SMAUG (Ref 4), was developed by Murphy of the Air Force Weapons Laboratory (AFWL), Kirtland Air Force Base, New Mexico. SMAUG calculates neutron and secondary gamma fluences through a series of equations that were curve fitted to the data of Straker and Gritzner (Ref 6). The calculation of primary gamma fluences is done through a series of equations that were curve fitted to the data of Bigoni (Ref 4:4). Both data sources were generated using a homogeneous constant - density, infinite air model.

Both SMAUG and ATR require only seconds of computer time (on a CDC 6600) and thus both offer inexpensive air transport calculations. However they both rely on a library of homogeneous air calculations which are modified by mass integral scaling to account for the true exponential variation of the air. This approach is probably valid only to burst altitudes of 20-25 km. The code ATR accounts for the air-ground interface by the first-last collision approximation (Ref 9).

Similar inexpensive calculations are needed for burst altitudes of 25 km and above, particularly for ABM war gaming and reentry vehicle survivability calculations. Such a technique is reported here. The author has sacrificed the accuracy of Monte Carlo and higher order Boltzmann equation solutions as used in the 35 previous attempts reported by

Stracker in favor of the Diffusion (P-1) approximation of the Boltzmann Equation. However some of the usual weaknesses of this approximation (diffusion) are avoided by a mathematical separation of scattered and unscattered radiation. The latter is calculated rigorously and the diffusion approximation is used only for the scattered radiation. The exponential variation of the atmosphere is treated with a unique coordinate system developed here by the author which also would permit the inclusion of layered clouds. This last feature is not included in the code and sample calculations presented here.

The resulting code is designed to operate as a subroutine of a nuclear survivability code being simultaneously developed at AFIT by DeRaad (Ref 2). As such, the input includes the spatial position of up to 100 target vehicles and their vulnerabilities to neutrons and gammas. The code compares the radiation field at each vehicle location to the vulnerability level for a survivability determination. Additionally the code will output iso-fluence lines for neutrons and gammas including neutron-induced gammas.

While the code does treat the exponential air exactly, the goal of an inexpensive calculation was not realized. A calculation requires 20 minutes of central processor (CP) time and 60 minutes of input-output time on the Wright-Patterson Air Force Base CDC 6600 computer. While this is less than a Monte Carlo calculation it is still excessive for repeated runs. (Perhaps $400 per run). However the results do offer an interesting test of the mass integral scaling approximation against a true exponential atmosphere and provide some insight to the radiation fields from high altitude bursts.

The mathematical development of the code is presented in Chapter II. The use of the code is illustrated with a sample problem in Chapter III. This chapter also serves as a complete users guide to the code. Chapter IV discusses the results obtained from this sample problem. Conclusions are drawn on the validity of these results and the successfulness of the code in Chapter V. Recommendations are also given in Chapter V. The code is presented in Appendix B.

II. Mathematical Development

The development of this computer code required a numerical approximation of the diffusion equation. The numerical approximation depends upon the atmospheric model selected and the coordinate system used. Therefore, the atmospheric model selected is discussed first. Next, the diffusion equation is presented with a discussion of the coordinate system selected. The actual numerical approximation used is then developed. This is followed by a discussion of the meshing, source terms, boundary conditions, and cross sections used. Finally, the actual method of solution is presented.

The Atmospheric Model

The atmospheric model chosen is based on four assumptions. The first assumption is that the composition of the atmosphere is 21% oxygen and 79% nitrogen. The second assumption is that the total particle density varies exponentially according to the relation

$$\rho = \rho_0 e^{-Z/H} \tag{1}$$

where

ρ = particle density at altitude z (particles/cm^3)

ρ_0 = particle density at sea level (particle/cm^3)

Z = altitude (km)

H = atmospheric scale height (km)

In order to evaluate the constants ρ_0 and H, this equation was changed to linear form by taking the natural logarithm of both sides to get

$$\ell n\rho = -Z/H + \ell n\rho_0 \qquad (2)$$

A curve fit was established using the values of particle densities from the U. S. Standard Atmosphere, 1962 (Ref 7: 2-19) for altitudes of 0 to 100 km. The scale height H was determined to be 7.0239 km and ρ_0 was determined to be 3.066×10^{19} particles/cm^3.

The third and fourth assumptions are that no atmosphere exists above 100 km, and that the earth is flat throughout the region of interest.

The Diffusion Equation

The multigroup, time independent diffusion equation for any energy group g is

$$\nabla \cdot D^g(\bar{r})\nabla F^g(\bar{r}) - \Sigma_R^g(\bar{r})F^g(\bar{r}) + S^g(\bar{r}) = 0 \qquad (3)$$

where

$D^g(\bar{r})$ = group diffusion coefficient (cm) as a function of spacial position $\bar{r}$

$F^g(\bar{r})$ = group neutron (or gamma) fluence (particles/cm^2) as a function of spacial position $\bar{r}$

Σ_R^g = group macroscopic removal cross-section (cm^{-1}) as a function of spacial position $\bar{r}$

S^g = group neutron (or gamma) source (particles/cm^3) as a function of spacial position $\bar{r}$

<u>The Diffusion Equation in a Cylindrical Coordinate System</u>. If the diffusion equation is expressed in a symmetric cylindrical (r,Z) coordinate system, Eq (3) can be written as a partial differential equation in r and Z (Ref 8:6). The equation is

$$D^g(Z)\,\frac{\partial^2 F^g(r,Z)}{\partial r^2} + \frac{D^g(Z)}{r}\,\frac{\partial F^g(r,Z)}{\partial r} + \frac{\partial D^g(Z)}{\partial Z}\,\frac{\partial F^g(r,Z)}{\partial Z}$$

$$+ D^g(Z)\,\frac{\partial^2 F^g(r,Z)}{\partial Z^2} - \Sigma_R^g(Z)F^g(r,Z) = -S^g(r,Z) \qquad (4)$$

When this equation is approximated by a difference equation, the spacing between adjacent radial points can vary; but, this radial spacing, once fixed, can not vary with altitude. In order that an accurate determination of fluence can be made, the radial spacing should have at least two or three mesh points per neutron (or gamma) mean free path. The horizontal mean free path, however, is dependent on the altitude. The exact dependence can be derived from the following:

Let us combine

$$\Sigma^g(Z) = \rho(Z)\sigma^g \qquad (5)$$

where

$\Sigma^g(Z)$ = group macroscopic cross section (cm^{-1}) at altitude Z

$\rho(Z)$ = particle density (particles/cm^3) at altitude Z

σ^g = group microscopic cross section (cm^{-2})

with Eq (1) to obtain

$$\Sigma^g(Z) = \Sigma^g(Z=0)e^{-Z/H} \tag{6}$$

where

$\Sigma^g(Z=0)$ = group macroscopic cross section (cm^{-1}) at sea level

The horizontal mean free path is dependent on the macroscopic total cross section and is given by the relation

$$\lambda^g = 1/\Sigma_t^g \tag{7}$$

where

λ^g = group mean free path (cm)

Σ_t^g = group macroscopic total cross section (cm^{-1})

The dependence on altitude Z can be shown by combining Eqs (7) and (6) to get

$$\lambda^g = \frac{e^{Z/H}}{\Sigma_t^g(Z=0)} \tag{8}$$

Equation (8) clearly illustrates that the horizontal mean free path increases exponentially with altitude. The required radial spacing is therefore controlled by the lowest altitude of interest. This requirement, however, means that the radial spacing at the highest altitude of interest will be a very small fraction of a mean free path.

Since a difference equation is written for each radial point on each altitude line, the number of points should be a minimum. However, at the higher altitudes, more radial points exist than required for a sufficiently accurate solution. This observation implies that another coordinate system may be advantageous. Since the solution requires about three points per mean free path, one coordinate logically should be mean free path.

A Non-Orthogonal Coordinate System. The author therefore proposes the following coordinate transforms:

$$y^1 = Hx^1 e^{x^3} \cos x^2 \tag{9}$$

$$y^2 = Hx^1 e^{x^3} \sin x^2 \tag{10}$$

$$y^3 = He^{x^3} \tag{11}$$

where

y^1 = the x coordinate in the Cartesian coordinate system

y^2 = the y coordinate in the Cartesian coordinate system

y^3 = the z coordinate in the Cartesian coordinate system

H = the scale height of the atmosphere.

x^1 = the first new coordinate

x^2 = the second new coordinate

x^3 = the third new coordinate

x^1, x^2, and x^3 are defined to be

$$x^1 = \frac{N}{H\Sigma_t^g(Z=0)} \tag{12}$$

$$x^2 = \theta \tag{13}$$

$$x^3 = \ell n(Z/H) \tag{14}$$

where

N = the radial distance in mean free paths from the x^3 axis

θ = the transverse angle about the x^3 axis

The diffusion equation now becomes

$$\nabla \cdot D(x^1,x^2,x^3)\nabla F(x^1,x^2,x^3) - \Sigma_R(x^1,x^2,x^3)F(x^1,x^2,x^3) + S(x^1,x^2,x^3) = 0 \tag{15}$$

in this new coordinate system. In tensor notation, the first term of Eq (15) can be written

$$\nabla \cdot D\nabla F = \frac{1}{\sqrt{g}} \sum_{K=1}^{3} \frac{\partial}{\partial x^K} \left(\sqrt{g}\ D \sum_{P=1}^{3} g^{KP} \frac{\partial F}{\partial x^P} \right) \tag{16}$$

where

g^{KP} are the contravariant metric tensors

g is the determinant formed by g_{KP}, the metric tensors

Therefore, both the metric and the contravariant metric tensors need to be determined to evaluate Eq (16).

The metric tensor is defined as

$$g_{KP} = \sum_{\alpha=1}^{3} \sum_{\beta=1}^{3} \frac{\partial y^{\alpha}}{\partial x^{K}} \frac{\partial y^{\beta}}{\partial x^{P}} \delta_{\alpha\beta} \tag{17}$$

where

$\delta_{\alpha\beta}$ = the Kronecker delta

When Eq (17) is solved, the following nine metric tensor components are obtained:

$$g_{11} = H^2 e^{2e^{x^3}} \tag{18}$$

$$g_{12} = 0 \quad (19$$

$$g_{13} = H^2 x^1 e^{x^3} e^{2e^{x^3}} \tag{20}$$

$$g_{21} = 0 \tag{21}$$

$$g_{22} = H^2 (x^1)^2 e^{2e^{x^3}} \tag{22}$$

$$g_{23} = 0 \tag{23}$$

$$g_{31} = H^2 x^1 e^{x^3} e^{2e^{x^3}} \tag{24}$$

$$g_{32} = 0 \tag{25}$$

$$g_{33} = H^2 e^{2x^3} \left[1 + (x^1)^2 e^{2e^{x^3}}\right] \tag{26}$$

Therefore, g is

$$g = \begin{vmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{vmatrix} = H^6 (x^1)^2 e^{2x^3} e^{4e^{x^3}} \tag{27}$$

The contravariant metric tensor g^{KP} is defined as

$$g^{KP} = \frac{G^{KP}}{g} \tag{28}$$

where

G^{KP} = the cofactor of the element g_{KP} in the determinant of Eq (27)

When Eq (28) is solved, the following nine contravariant metric components are obtained:

$$g^{11} = \frac{1 + (x^1)^2 \ e^{2e^{x^3}}}{H^2 e^{2e^{x^3}}} \tag{29}$$

$$g^{12} = 0 \tag{30}$$

$$g^{13} = \frac{-x^1}{H^2 e^{x^3}} \tag{31}$$

$$g^{21} = 0 \tag{32}$$

$$g^{22} = \frac{1}{H^2 (x^1)^2 e^{2e^{x^3}}} \tag{33}$$

$$g^{23} = 0 \tag{34}$$

$$g^{31} = \frac{-x^1}{H^2 e^{x^3}} \tag{35}$$

$$g^{32} = 0 \tag{36}$$

$$g^{33} = \frac{1}{H^2 e^{2x^3}} \tag{37}$$

If we assume fluence is symmetric with respect to x^2 and expand Eq (16), eliminating the zero values of g^{KP}, Eq (16) becomes

$$\nabla \cdot D \nabla F = \frac{1}{\sqrt{g}} \left[\frac{\partial}{\partial x^1} \sqrt{g}\, D \left(g^{11} \frac{\partial F}{\partial x^1} + g^{13} \frac{\partial F}{\partial x^3} \right) + \frac{\partial}{\partial x^3} \sqrt{g}\, D \left(g^{31} \frac{\partial F}{\partial x^1} + g^{33} \frac{\partial F}{\partial x^3} \right) \right] \tag{38}$$

In order to evaluate this expression, the functional dependence of D, the diffusion coefficient, must be determined. The diffusion constant is defined as

$$D = \frac{1}{3\Sigma_{TR}(Z)} \tag{39}$$

where

$\Sigma_{TR}(Z)$ = the macroscopic transport cross section (cm^{-1}) at altitude Z

The transport cross section is defined as $\Sigma_{TR} = \Sigma_t - \bar{\mu}\Sigma_S$ where $\bar{\mu}$ is the average cosine of the angle of anisotropic scatter and Σ_S is the macroscopic scatter cross section. Therefore, combining Eq (6) and Eq (39), D becomes

$$D = \frac{e^{Z/H}}{3\Sigma_{TR}(Z=0)} = D_0 e^{Z/H} \tag{40}$$

where

D_0 = the diffusion coefficient (cm) at sea level

When Eq (11), the Z coordinate transform equation, is used on Eq (40), D becomes a function only of x^3 and is given by

$$D = D_0 e^{e^{x^3}} \tag{41}$$

If the expressions for D, g, g^{11}, g^{13}, g^{31}, and g^{33} are substituted into Eq (38) and the indicated partial derivatives evaluated, Eq (38) becomes

$$\nabla\cdot D\nabla F = \frac{D_0 e^{e^{x^3}}}{H^2 e^{2x^3}} \left(\frac{e^{2x^3}}{x^1 e^{2e^{x^3}}} \frac{\partial F}{\partial x^1} + (e^{x^3} - 1) \frac{\partial^2 F}{\partial (x^1)^2} \right.$$

$$- 2x^1 e^{x^3} \frac{\partial^2 F}{\partial x^3 \partial x^1} + \frac{e^{2x^3} + (x^1)^2 e^{2x^3} e^{2e^{x^3}}}{e^{2e^{x^3}}} \frac{\partial^2 F}{\partial (x^1)^2}$$

$$\left. + \frac{\partial^2 F}{\partial (x^3)^2} \right) \tag{42}$$

Once Eq (42) is substituted into the diffusion equation, Eq (15), the diffusion equation becomes

$$\frac{D_0 e^{e^{x^3}}}{H^2 e^{2x^3}} \left(\frac{e^{2x^3}}{x^1 e^{2e^{x^3}}} \frac{\partial F}{\partial x^1} + (e^{x^3} - 1) \frac{\partial F}{\partial x^3} - 2x^1 e^{x^3} \frac{\partial^2 F}{\partial x^1 \partial x^3} \right.$$

$$\left. + \frac{e^{2x^3} + (x^1)^2 e^{2x^3} e^{2e^{x^3}}}{e^{2e^{x^3}}} \frac{\partial^2 F}{\partial (x^1)^2} + \frac{\partial^2 F}{\partial (x^3)^2} \right)$$

$$- \Sigma_R(Z=0) e^{e^{x^3}} F + S = 0 \tag{43}$$

Equation (43), when simplified, becomes

$$\frac{1}{x^1 e^{2e^{x^3}}} \frac{\partial F}{\partial x^1} + \frac{e^{x^3} - 1}{e^{2x^3}} \frac{\partial F}{\partial x^3} - \frac{2x^1}{e^{x^3}} \frac{\partial^2 F}{\partial x^1 \partial x^3}$$

$$+ \frac{1 + (x^1)^2 e^{2e^{x^3}}}{e^{2e^{x^3}}} \frac{\partial^2 F}{\partial (x^1)^2} + \frac{1}{e^{2x^3}} \frac{\partial^2 F}{\partial (x^3)^2}$$

$$- \frac{H^2}{D_0} \Sigma_R (Z=0) F = - \frac{H^2}{D_0 e^{e^{x^3}}} S \qquad (44)$$

The Difference Form of the Diffusion Equation

In order to express Eq (44) in difference form, the dependent variables F^g and S^g at any point (x^1_i, x^3_j) are defined as

$$F^g(x^1_i, x^3_j) = F^g_{i,j} \qquad (45)$$

$$S^g(x^1_i, x^3_j) = S^g_{i,j} \qquad (46)$$

The partial derivatives of Eq (44) can be expressed in difference form by the use of a central difference operator. The partial derivatives are therefore

$$\frac{\partial F^g_{i,j}}{\partial x^1} = \frac{F^g_{i+1,j} - F^g_{i-1,j}}{2\Delta x^1} \qquad (47)$$

$$\frac{\partial F^g_{i,j}}{\partial x^3} = \frac{F^g_{i,j+1} - F^g_{i,j-1}}{2\Delta x^3} \tag{48}$$

$$\frac{\partial^2 F^g_{i,j}}{\partial (x^1)^2} = \frac{F^g_{i+1,j} - 2F^g_{i,j} + F^g_{i-1,j}}{(\Delta x^1)^2} \tag{49}$$

$$\frac{\partial^2 F^g_{i,j}}{\partial (x^3)^2} = \frac{F^g_{i,j+1} - 2F^g_{i,j} + F^g_{i,j-1}}{(\Delta x^3)^2} \tag{50}$$

$$\frac{\partial^2 F^g}{\partial x^1 \partial x^3} = \frac{F^g_{i+1,j+1} - F^g_{i+1,j-1} - F^g_{i-1,j+1} + F^g_{i-1,j-1}}{4\Delta x^1 \Delta x^3} \tag{51}$$

The difference equation is then derived by substituting equations (45) through (51) into Eq (44) to get

$$- \frac{x^1}{2\Delta x^1 \Delta x^3 e^{x^3}} F^g_{i-1,j-1} + \left(\frac{1}{(\Delta x^3)^2 e^{2x^3}} - \frac{1}{2\Delta x^3 e^{x^3}} \right.$$

$$\left. + \frac{1}{2\Delta x^3 e^{2x^3}} \right) F^g_{i,j-1} + \frac{x^1}{2\Delta x^1 \Delta x^3 e^{x^3}} F^g_{i+1,j-1}$$

$$+ \left(\frac{1}{(\Delta x^1)^2 e^{2e^{x^3}}} + \left[\frac{x^1}{\Delta x^1} \right]^2 - \frac{1}{2\Delta x^1 x^1 e^{2e^{x^3}}} \right) F^g_{i-1,j}$$

$$- \left(\frac{2}{(\Delta x^1)^2 e^{2c^{x^3}}} + \frac{2(x^1)^2}{(\Delta x^1)^2} + \frac{2}{(\Delta x^3)^2 e^{2x^3}} + \frac{H^2 \Sigma^g_R (Z=0)}{D_0} \right) F^g_{i,j}$$

$$+\left(\frac{1}{(\Delta x^1)^2 e^{2e^{x^3}}} + \left[\frac{x^1}{\Delta x^1}\right]^2 + \frac{1}{2\Delta x^1 x^1 e^{2e^{x^3}}}\right) F^g_{i+1,j}$$

$$+ \frac{x^1}{2\Delta x^1 \Delta x^3 e^{x^3}} F^g_{i-1,j-1} + \left(\frac{1}{(\Delta x^3)^2 e^{2x^3}} - \frac{1}{2\Delta x^3 e^{x^3}} - \frac{1}{2\Delta x^3 e^{2x^3}}\right)$$

$$F^g_{i,j+1} - \frac{x^1}{2\Delta x^1 \Delta x^3 e^{x^3}} F_{i+1,j+1} = - \frac{H^2 S^g_{i,j}}{D^g_0 e^{e^{x^3}}} \tag{52}$$

The fluence coefficients of Eq (52) are defined as

$$G_{i,j} = \frac{x^1}{2\Delta x^1 \Delta x^3 e^{x^3}} \tag{53}$$

$$\overline{A}_j = \frac{1}{(\Delta x^3)^2 e^{2x^3}} - \frac{1}{2\Delta x^3 e^{x^3}} + \frac{1}{2\Delta x^3 e^{2x^3}} \tag{54}$$

$$\overline{B}_{i,j} = \frac{1}{(\Delta x^1)^2 e^{2e^{x^3}}} + \left(\frac{x^1}{\Delta x^1}\right)^2 - \frac{1}{2\Delta x^1 x^1 e^{2e^{x^3}}} \tag{55}$$

$$C_{i,j} = \frac{2}{(\Delta x^1)^2 e^{2e^{x^3}}} + \frac{2(x^1)^2}{(\Delta x^1)^2} + \frac{2}{(\Delta x^3)^2 e^{2x^3}} + \frac{H^2 \Sigma^g_R (Z=0)}{D^g_0} \tag{56}$$

$$B_{i,j} = \frac{1}{(\Delta x^1)^2 e^{2e^{x^3}}} + \left(\frac{x^1}{\Delta x^1}\right)^2 + \frac{1}{2\Delta x^1 x^1 e^{2e^{x^3}}} \tag{57}$$

$$A_j = \frac{1}{(\Delta x^3)^2 e^{2x^3}} + \frac{1}{2\Delta x^3 e^{x^3}} - \frac{1}{2\Delta x^3 e^{2x^3}} \quad (58)$$

When Eqs (53) through (58) are substituted into Eq (52), the difference equation finally becomes

$$-G_{i,j}F^g_{i-1,j-1} + \overline{A}_j F^g_{i,j-1} + G_{i,j}F^g_{i+1,j-1} + \overline{B}_{i,j}F^g_{i-1,j}$$

$$-C_{i,j}F^g_{i,j} + B_{i,j}F^g_{i+1,j} + G_{i,j}F^g_{i-1,j+1} + A_j F^g_{i,j+1}$$

$$-G_{i,j}F^g_{i+1,j+1} = -\frac{H^2 S^g_{i,j}}{D^g_0 e^{x^3}} \quad (59)$$

Mesh, Source, Boundary Conditions, and Cross Sections

In order to completely prepare Eq (59) for use in a computer four factors must be defined. First the mesh area and mesh interval (that is, the range of x^1 and x^3 and the values of Δx^1 and Δx^3) must be determined. Second, the source term $S^g_{i,j}$ must be defined. Third, boundary conditions around the edge of the mesh must be defined. Fourth, the cross section used must be defined.

The Mesh Area and Mesh Interval. The x^1 coordinate is a function of the sea level mean free path of some energy group of gammas or neutrons. This can be shown by combining equations (12) and (7) to get

$$x^1 = \frac{N\lambda^g(Z=0)}{H} \quad (60)$$

where

$\lambda^g(Z=0)$ = the sea level horizontal mean free of gammas or neutrons of an energy group g.

The minimum value of x^1 is zero (the x^3 axis). Rigorously, the maximum value of x^1 should be infinite because the neutrons and gammas are transported to great heights or radial distances. However, because a finite number of mesh points must be defined for a computer solution, these boundaries must be defined as a finite (and large) number of mean free paths, N, and the fluence at these boundaries will be forced to zero. In order to determine the effect of this approximation, the author wrote a homogeneous air, constant density, one energy group code in which the boundary was first taken as 10 mean free paths and then as 25 mean free paths. The fluences were identical up to seven mean free paths from the burst. Between seven and nine mean free paths only the third significant figure of the fluences were different. Only in the tenth mean free path order of magnitude differences appeared. Therefore the maximum value of x^1 was chosen to correspond to ten mean free paths.

The value of λ^g, the mean free path, could be selected from any energy group. However, in order to obtain reasonable accuracy the maximum mean free path was chosen.

Ideally, the mesh interval should be chosen such that the shortest mean free path has at least three mesh points. This requires, in the 40 group structure used, about 600 total mesh points over the x^1 axis. This number of points was impractical to use on the computer available. Therefore, 40 points were selected for the sample run of Chapter III. (However, the program was constructed so that up to 60 points could be used.) The description of the MESH subroutine in Appendix B describes how this option may be used.

The determination of the range of x^3 was also based on a maximum range of ten mean free paths from the burst. In this case the maximum value of x^3 would be equivalent to ten mean paths up from the burst and the minimum value would be equivalent to ten mean free paths down from the burst. The energy group with the longest mean free path was also chosen to evaluate the range of x^3. However, a simple relationship between x^3 and mean free path such as Eq (60) does not exist since the air density is changing over the path traveled by the neutron or gamma. A relationship can be derived between altitude Z and mean free path based on the assumption of the exponential variation in air density with height.

This relationship is based upon the definition of macroscopic cross section given in Eq (5):

$$\Sigma_t(Z) = \rho(Z)\sigma_T \qquad (5)$$

If this equation is integrated over the average mean free path at any altitude Z to an altitude Z + λ

$$\int_{Z}^{Z+\lambda} \Sigma_t dZ' = \int_{Z}^{Z+\lambda} \sigma_T \rho dZ' \qquad (61)$$

the following series of relationships can be obtained.

On the left hand side of Eq (60) we extract the average value of Σ_t over one mean free path ($\langle\Sigma_t\rangle$) and on the right hand side replace ρ by its definition in Eq (1) to get

$$\langle\Sigma_t\rangle \int_{Z}^{Z+\lambda} dZ' = \int_{Z}^{Z+\lambda} \sigma_T \rho_0 e^{-Z'/H} dZ' \qquad (62)$$

But the integral on the left hand side is λ and the expression $\rho_0 \sigma_T$ on the right hand side is $\Sigma_t(Z=0)$, a constant. Therefore

$$\langle\Sigma_t\rangle \lambda = \Sigma_t(Z=0) \int_{Z}^{Z+\lambda} e^{-Z'/H} dZ' \qquad (63)$$

Carrying out the integration on the right side and using the definition of mean free path as the reciprocal of macroscopic total cross section, Eq (63) becomes

$$1 = \Sigma_t(Z=0)e^{-Z/H}[1 - e^{-\lambda/H}] \qquad (64)$$

But, $\Sigma_t(Z=0)e^{-Z/H}$ is $\Sigma_t(Z)$. Therefore

$$1 = \Sigma_t[1 - e^{-\lambda/H}] \qquad (65)$$

Solving Eq (65) for λ

$$\lambda = -H \ln\left(1 - \frac{1}{H\Sigma_t}\right) \qquad (66)$$

Therefore, this is the relationship between λ and Z (since Σ_t is a function of Z) for the average mean free path up from any altitude Z.

In a similar manner, the relationship between λ and Z for the average mean free path down from any altitude Z can be found to be

$$\lambda = H \ln\left(1 + \frac{1}{H\Sigma_t}\right) \qquad (67)$$

Equations (66) and (67) were used to find the upper and lower values of altitude equivalent to ten mean free paths in either direction. The minimum and maximum values of x^3 were then obtained from its definition of

$$x^3 = \ln(Z/H) \qquad (14)$$

The minimum value of x^3, however. has an additional constraint. Z can not be equal to or less than zero. This constraint in turn causes a lower limit of seven kilometers for the burst height in order to keep a ten mean free path lower boundary without intersecting the flat earth. The adoption of an upper limit of 100 km for the model atmosphere places a restriction upon the upper value of x^3 to 100 km.

As in the case for the x^1 mesh interval, the mesh interval should be selected such that the shortest mean free path has at least three mesh points. This requires, in the

40 group structure used, about 1200 total mesh points over the x^3 axis. Again to limit the number of mesh points in the x^3 direction, 70 points were selected for the sample run of Chapter III. (The program, however, was constructed to allow up to 120 points in the x^3 direction.) The description of the MESH subroutine in Appendix B describes how this option may be used.

Definition of the Source Term. The diffusion equation implicitly assumes a linear variation in the cosine of the directional angle of the diffusing particles. This follows because the diffusion equation results from a Legendre expansion in direction of the more rigorous Boltzmann equation which retains only the first two terms. Near the source the virgin particles make up the largest fraction of the total fluence and are nearly monodirectional in the outbound direction. Thus diffusion theory is weakest close to the source which is the region of highest interest. This dilemma can be avoided by defining the fluence at any point to have two components.

The first component at any point consists of the virgin (unscattered) particles from the burst. The second component consists of the scattered particles. This second component at any point therefore contains those particles scattered down from a higher energy group or from virgin scatter which entered the energy group at some other point and have not yet downscattered to a lower energy group.

The virgin particles are solved rigorously while the diffusion equation is used to describe only the scattered particles. The source term in this diffusion equation consists of those particles scattering out of the virgin groups at that point plus those particles downscattering from higher energy scattered groups. The source term can therefore be expressed as

$$S^g_{i,j} = \sum_{g'=1}^{g} \Sigma^{g'}_{S_v} F^{g'}_{v_{i,j}} + \sum_{g'=1}^{g-1} \Sigma^{g'}_{S} F^{g'}_{i,j} \qquad (68)$$

where

$S^g_{i,j}$ = the source term for group g at spacial point i,j (particles/cm^3)

$\Sigma^{g'}_{S_v}$ = the macroscopic scatter cross section for virgin group g' (cm^{-1})

$F^{g'}_{v_{i,j}}$ = the virgin fluence in group g' at spacial location i,j (particles/cm^2)

$\Sigma^{g'}_{S}$ = the macroscopic scatter cross section for scattered group g' (cm^{-1})

$F^{g'}_{i,j}$ = the scattered fluence in group g' at spacial location i,j (particles/cm^2)

However, in order to evaluate Eq (68), the virgin fluence for any point i,j must be determined. The virgin fluence at any point is

$$F^g_{v_{i,j}} = \frac{S^g_0 \, e^{-\langle \Sigma_t \rangle R}}{4\pi R^2} \qquad (69)$$

where

S_0^g = the total number of particles in group g emitted from the nuclear weapon.

R = the distance from the burst point to the point i,j

$\langle \Sigma_t \rangle$ = the average macroscopic total cross section of the particle over the distance

The value of $\langle \Sigma_t \rangle$ can be found by integrating the point value of Σ_t (which is dependent upon the altitude Z) over the path length R. So

$$\int_0^R \Sigma_t \, dR' = \langle \Sigma_t \rangle R \tag{70}$$

However, since Σ_t is a function of altitude the left side should be expressed in terms of Z. Figure 1 illustrates the geometry.

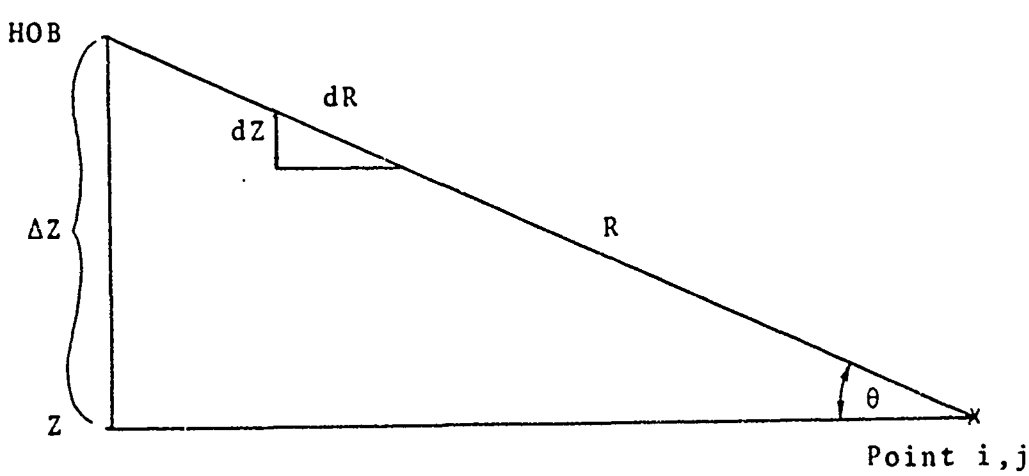

Fig. 1. Sample Geometry Relating Variables Needed in Calculation of Average Cross Section. (Note: HOB is Height of Burst.)

From Fig. 1 the following relationship is obvious:

$$dR = \csc \theta dZ \tag{71}$$

Therefore, from Eqs (71), (70), and (6)

$$\langle \Sigma_t \rangle R = \Sigma_t(Z=0) \csc \theta \int_{HOB}^{Z} e^{-Z'/H} dZ' \tag{72}$$

Once this integ--tion is performed, Eq (72) becomes

$$\langle \Sigma_t \rangle R = H\Sigma_t(Z=0) \csc \theta [e^{-Z/H} - e^{-HOB/H}] \tag{73}$$

and also

$$\csc \theta = \frac{R}{\Delta Z} \tag{74}$$

Therefore, from Eqs (69), (73), and (74), the virgin fluence at any point is given by

$$F^g_{v_{i,j}} = \frac{S^g_0}{4\pi R^2} e^{H\Sigma_t(Z=0) \frac{R}{\Delta Z} [e^{-Z/H} - e^{-HOB/H}]} \tag{75}$$

Using Eqs (75), (68), and (59), a difference equation can be written for any interior point i,j in the meshed area defined by the ranges of x^1 and x^3 and by the intervals Δx^1 and Δx^3. However, since Eq (59) is a nine-point difference equation involving fluences at points such as i-1, j-1, boundary conditions must be defined.

Boundary Conditions. The first boundary condition that must be defined is for the points on the x^3 axis, that is, when x^1 is zero. Two problems appear if the difference equation (59) is used. This first problem is that the coefficients $B_{i,j}$ and $\overline{B}_{i,j}$ contain the term $1/x^1$ which is infinite at x^1 equal to zero. The second problem is that at x^1 equal to zero, i is equal to one, and in the nine point difference equation three i - 1 (i = 0) terms appear.

The second problem is easily treated since, in the development of the diffusion equation in the x^1, x^2, x^3 coordinate system, symmetry of fluence with respect to the x^2 coordinate (the transverse angle) was assumed. Therefore

$$F_{0,j} = F_{2,j} \tag{76}$$

The first problem requires a special development of the difference equation. This requirement exists since the diffusion equation in the x^1, x^2, x^3 coordinate system, Eq (44), contained the term

$$\frac{1}{x^1 e^{2e^{x^3}}} \frac{\partial F}{\partial x^1}$$

Equation (76) is equivalent to stating

$$\left(\frac{\partial F}{\partial x^1} \right)_{x^1=0} = 0 \tag{77}$$

The term, therefore, at $x^1 = 0$ becomes 0/0 which is undefined. However, the use of L'Hospital's rule on this term results in

$$\lim_{x^1 \to 0} \frac{\partial F/\partial x^1}{x^1 e^{2e^{x^3}}} = \lim_{x^1 \to 0} \frac{\partial^2 F/\partial (x^1)^2}{e^{2e^{x^3}}} \quad (78)$$

Therefore, using the substitution indicated in Eq (78) into Eq (44), forming a new difference equation, and using the definition of Eq (76), the following special form of the difference equation for i = 1 is obtained:

$$\bar{A}F_{1,j-1} - CF_{1,j} + BF_{2,j} + AF_{1,j+1} = -\frac{H^2 S_{i,j}}{D_0 e^{e^{x^3}}} \quad (79)$$

where

$$\bar{A} = \frac{1}{(\Delta x^3)^2 e^{2x^3}} - \frac{1}{2\Delta x^3 e^{x^3}} + \frac{1}{2\Delta x^3 e^{2x^3}} \quad (80)$$

$$C = \frac{4}{(\Delta x^1)^2 e^{2e^{x^3}}} + \frac{2}{(\Delta x^3)^2 e^{2x^3}} + \frac{H^2 \Sigma_R (Z=0)}{D_0} \quad (81)$$

$$B = \frac{4}{(\Delta x^1)^2 e^{2e^{x^3}}} \quad (82)$$

$$A = \frac{1}{(\Delta x^3)^2 e^{2x^3}} + \frac{1}{2\Delta x^3 e^{x^3}} - \frac{1}{2\Delta x^3 e^{2x^3}} \quad (83)$$

The second boundary that must be defined is when x^3 is a minimum (the lower boundary of the mesh). This lower boundary was chosen far enough away from the burst (10 mean free paths of the group with the longest mean free path) so that the value of fluence is at least seven or eight orders of magnitude lower than the fluences existing around the burst. Therefore, along this boundary, fluence is assumed to be zero. Reviewing Eq (53) through (59) this condition implies that the coefficients of the $F_{i-1,j-1}$, $F_{i,j-1}$, and $F_{i+1,j-1}$ terms must be zero when j = 1.

The third boundary that must be defined is when x^1 is a maximum (the outer boundary of the mesh). This boundary was also chosen so that the fluences existing at this boundary were low compared to those around the burst. Therefore, along this boundary, fluences are assumed to be zero. This condition implies that the coefficients in Eq (59) of the $F_{i+1,j-1}$, $F_{i+1,j}$, and $F_{i+1,j+1}$ terms must be zero when x^1 is a maximum.

The final boundary condition that must be defined is when x^3 is a maximum (the upper boundary of the mesh). Two possible conditions can exist. First, this upper boundary is less than 100 km and second, this upper boundary is 100 km. If the first condition exists, the upper boundary is ten mean free paths, of the group with the longest mean free path, away from the burst. Therefore, by the same logic of the preceding two paragraphs, fluence along this boundary is zero and the coefficients of the $F_{i-1,j+1}$, $F_{i,j+1}$, and $F_{i+1,j+1}$ terms of Eq (59) must be zero.

The second condition occurs when, in the calculation for the upper boundary, altitudes of greater than 100 km are obtained. In this case, the upper boundary is set to 100 km. Therefore, the fluences at this boundary may not be very much lower than about the burst and assuming zero fluences for this boundary would be invalid. However, since a vacuum is assumed to exist above 100 km, a valid boundary condition is that the return current is zero. Or using Fick's Law it can be stated as

$$J_- = \frac{F}{4} + \frac{D}{2} \frac{\partial F}{\partial x^3} = 0 \tag{84}$$

where

J_- = return current (particles/cm^2)

D = diffusion coefficient (cm) at 100 km

F = fluence (particles/cm^2)

This equation can be differenced using a central difference operator and solved for $F_{i,j+1}$ to get

$$F_{i,j+1} = F_{i,j-1} - \frac{\Delta x^3 F_{i,j}}{D_0 e^{e^{x^3}}} \tag{85}$$

When this is substituted into Eq (59) the following special form of the difference equation is obtained:

$$\bar{A}F_{i,j-1} + \bar{B}F_{i-1,j} + CF_{i,j} + BF_{i+1,j} = -\frac{H^2 S_{i,j}}{D_0 e^{e^{x^3}}} \tag{86}$$

where

$$\overline{A} = \frac{2}{(\Delta x^3)^2 e^{2x^3}} \tag{87}$$

$$\overline{B} = \frac{1}{(\Delta x^1)^2 e^{2e^{x^3}}} + \left(\frac{x^1}{\Delta x^1}\right)^2 - \frac{1}{2\Delta x^1 x^1 e^{2e^{x^3}}}$$

$$- \frac{x^1}{2D_0 \Delta x^1 e^{x^3} e^{e^{x^3}}} \tag{88}$$

$$C = \frac{2}{(\Delta x^1)^2 e^{2e^{x^3}}} + \frac{2(x^1)^2}{(\Delta x^1)^2} + \frac{2}{(\Delta x^3)^2 e^{2x^3}} + \frac{H^2 \Sigma_R (Z=0)}{D_0}$$

$$+ \frac{1}{D_0 \Delta x^3 e^{2x^3} e^{e^{x^3}}} + \frac{1}{2D_0 e^{2e^{x^3}}} - \frac{1}{2D_0 e^{2x^3} e^{e^{x^3}}} \tag{89}$$

$$B = \frac{1}{(\Delta x^1)^2 e^{2e^{x^3}}} + \left(\frac{x^1}{\Delta x^1}\right)^2 + \frac{1}{2\Delta x^1 x^1 e^{2e^{x^3}}}$$

$$+ \frac{x^1}{2D_0 \Delta x^1 e^{x^3} e^{e^{x^3}}} \tag{90}$$

A difference equation now can be written for every point in the mesh with all terms defined except for the cross sections.

Cross Sections. The cross sections selected were obtained from the Radiation Shielding Information Center (RSIC) of Oak Ridge National Laboratory from the RSIC Data Library Collection DLC-14 (Ref 3). These cross sections are multigroup and consist of 22 neutron groups coupled with 18 gamma groups. Since the gamma groups were coupled with the neutron groups, calculations of secondary gamma fluences (caused by neutron scattering in air) are possible in addition to the calculation of primary neutrons and gamma fluences. These cross sections were selected since they were used for Straker and Gritzner's constant density, infinite air calculations (Ref 6). Therefore, these cross sections were indirectly used by SMAUG since SMAUG curve fits Straker and Gritzner's data. Use of these cross sections allows comparison between the code presented in this report and SMAUG.

These cross sections, however, are for an air density of 1.11 mg/cm^3 which is equivalent to about 1 km in the U.S. Standard Atmosphere, or to about 2 km in the curve fitted exponential atmosphere used in this report. Since the entire mathematical development of this report has been based on sea level cross sections, these cross sections were extrapolated to sea level using Eq (6). In addition, the cross sections are differential cross sections expanded in six terms of a Legendre expansion. Therefore, removal and transport cross sections had to be calculated. The energy ranges of the neutron groups are presented in Table I and the energy ranges

Table I

Neutron Group Energy Ranges

Neutron Group Number	Energy Range (MeV)
1	12.2 to 15.0
2	10.0 to 12.2
3	8.19 to 10.0
4	6.36 to 8.19
5	4.97 to 6.36
6	4.07 to 4.97
7	3.01 to 4.07
8	2.46 to 3.01
9	2.35 to 2.46
10	1.83 to 2.35
11	1.11 to 1.83
12	0.55 to 1.11
13	0.111 to 0.55
14	0.00335 to 0.111
15	5.83E-4 to 3.35E-3 (a)
16	1.01E-4 to 5.83E-4
17	2.90E-5 to 1.01E-4
18	1.07E-5 to 2.90E-5
19	3.06E-6 to 1.07E-5
20	1.12E-6 to 3.06E-6
21	4.14E-7 to 1.12E-6
22	0.0 to 4.14E-7

(a) 3.35E-3 reads as 3.35×10^{-3} (Ref 4:3) of the gamma groups are presented in Table II. The actual cross sections used are presented in Appendix C.

Method of Solution

The group scattered fluences were solved, one group at a time starting with the highest energy neutron group (group 1). A difference equation was written for each point in the mesh. The result, for any group, is a matrix equation of the form

$$\underline{A}\ \underline{F} = \underline{S} \tag{91}$$

where

$\underline{A}$ = the matrix consisting of the terms $\overline{A}$, $\overline{B}$, C, A, B, and G

$\underline{F}$ = the array of unknown scattered fluences

$\underline{S}$ = the array of source terms

The matrix $\underline{A}$ is a block tri-diagonal matrix. Appendix A defines a block tri-diagonal matrix and presents the algorithm used to solve matrix equation (91).

A problem arose during the computer solution using the coordinate system and meshing developed in this chapter. Negative fluences occurred in the upper portions of the meshed area. This was a result of the radial spacing between mesh spaces. In the x^1, x^2, x^3 coordinate system, Δx^1 is a constant; however, the actual distance between mesh points is increasing exponentially with increase in altitude. Therefore, in the upper portion of the mesh, the fluence was

Table II

Gamma Group Energy Ranges

Gamma Group Number	Energy Range (MeV)
1	8.0 to 10.0
2	6.5 to 8.0
3	5.0 to 6.5
4	4.0 to 5.0
5	3.0 to 4.0
6	2.5 to 3.0
7	2.0 to 2.5
8	1.66 to 2.0
9	1.33 to 1.66
10	1.0 to 1.33
11	0.8 to 1.0
12	0.6 to 0.8
13	0.4 to 0.6
14	0.3 to 0.4
15	0.2 to 0.3
16	0.1 to 0.2
17	0.05 to 0.1
18	0.02 to 0.05

(Ref 4:4)

changing rapidly between mesh points and causing the negative fluences. In order to correct this fault, x^1 was not allowed to exceed the equivalent of 200 km at the outer boundary.

When x^1 exceeds this value, a different coordinate system is used starting on the next value of x^3. This system was

$$y^1 = x^1 \cos x^2 \tag{92}$$

$$y^2 = x^1 \sin x^2 \tag{93}$$

$$y^3 = He^{x^3} \tag{94}$$

x^1 and x^2 are therefore the r and θ of a normal cylindrical coordinate system. The difference equation from using this coordinate system is

$$\bar{A}F_{i,j-1} + \bar{B}F_{i-1,j} - CF_{i,j} + BF_{i+1,j} + AF_{i,j+1} =$$

$$= - \frac{H^2 S_{i,j}}{D_0} \tag{95}$$

where

$$\bar{A} = \frac{1}{(\Delta x^3)^2 e^{2x^3}} - \frac{1}{2\Delta x^3 e^{x^3}} + \frac{1}{2\Delta x^3 e^{2x^3}} \tag{96}$$

$$\bar{B} = \frac{H^2}{(\Delta x^1)^2} - \frac{H^2}{2x^1 \Delta x^1} \tag{97}$$

$$C = \frac{2H^2}{(\Delta x^1)^2} + \frac{2}{(\Delta x^3)^2 e^{2x^3}} + \frac{H^2 \Sigma_R (Z=0)}{D_0} \tag{98}$$

$$B = \frac{H^2}{(\Delta x^1)^2} + \frac{H^2}{2x^1 \Delta x^1} \tag{99}$$

$$A = \frac{1}{(\Delta x^3)^2 e^{2x^3}} + \frac{1}{2\Delta x^3 e^{x^3}} - \frac{1}{2\Delta x^3 e^{2x^3}} \tag{100}$$

Since this difference equation is used only in the upper area of the mesh, only three boundary conditions need be defined. They are for the x^3 axis, the upper boundary, and the outer boundary. The boundary conditions are the same as defined before. Therefore, for the x^3 axis, $\bar{A}$ and A remain as defined by Eqs (96) and (100), $\bar{B}$ becomes zero, and

$$B = \frac{4H^2}{(\Delta x^1)^2} \tag{101}$$

$$C = \frac{4H^2}{(\Delta x^1)^2} + \frac{2}{(\Delta x^3)^2 e^{2x^3}} + \frac{H^2 \Sigma_R (Z=0)}{D_0} \tag{102}$$

For the upper boundary, the same two possibilities exist. The boundary can be less than 100 km or it can be 100 km. If the upper boundary is less than 100 km, $\bar{A}$, $\bar{B}$, C, and B remain defined as in Eqs (96) through (99) and A is equal to zero. If the upper boundary is 100 km, $\bar{B}$ and B remain as defined in Eqs (97) and (99), A is zero, and

$$\bar{A} = \frac{2}{(\Delta x^3)^2 e^{2x^3}} \tag{103}$$

$$C = \frac{2H^2}{(\Delta x^1)^2} + \frac{2}{(\Delta x^3)^2 e^{2x^3}} + \frac{H^2 \Sigma_R(Z=0)}{D_0} + \frac{1}{D_0 \Delta x^3 e^{2x^3} e^{e^{x^3}}}$$

$$+ \frac{1}{2D_0 e^{x^3} e^{e^{x^3}}} - \frac{1}{2D_0 e^{2x^3} e^{e^{x^3}}} \tag{104}$$

For the outer boundary, $\bar{A}$, $\bar{B}$, C, and A remain as defined as in Eqs (96), (97), (98), and (100). B is zero.

The mesh interval Δx^3 remains the same since x^3 has not been changed. The new Δx^1 (actually a Δr) is the radial equivalent to the last Δx^1 of the old coordinate scheme.

The computer code was written for the Wright-Patterson Air Force Base CDC 6600 computer using an on-line plotter. The language used was FORTRAN Extended.

III. A Users Guide to the Code

This chapter presents a guide to the user on how to use the code. The input for a sample problem is shown as an illustration. Three types of punched cards are required; control cards, data cards, and end of job cards.

Control Cards

Four control cards are required. These four cards preceed any other cards. Entries on all these cards start in column one. These cards are unique to the Wright-Patterson Air Force Base CDC 6600. Other computers will require different control cards.

Card No. 1:

This card contains the users initials, time to run the code, core size in octal, problem number assigned to the user by the computer center, name, telephone number, and class.

EXAMPLE: RDM,T3400,CM120000. T710675,MCLAREN,2533886,
GNE 72

The time required for running the code is 3400 seconds and a core size of 120000_8 is required.

Card No. 2:

This card selects the FORTRAN compiler.

EXAMPLE: FTN.

Card No. 3:

This card loads the program and causes it to go into execution.

EXAMPLE: LGO.

Card No. 4:

This card signals the end of the control cards.

EXAMPLE: A multipunch 7/8/9 in column one.

Data Cards

The computer code itself follows the control cards. After the code, a series of data cards are required. The data is input in either an I format or an E format. All numbers are right justified in the field specified by the format statement. Examples of correct and incorrect use are shown below. b represents a blank space. All entries start in column one.

Format	Correct	Incorrect
2I4	bb22bb18 bbb5-100	22bb18bb b5b-100b
1x,1PE11.4	bb1.2345E+04 b-1.2345E+04	1.2345E+04bb

Card No. 1:

This card contains two numbers in 2I4 format. The first number is the number of neutron energy groups and the second number is the number of gamma groups. This card is included as part of the input library and normally requires no preparation on the part of the user. However, if the user desires to use a different cross section data set with different number of groups, this card must be replaced with a card prepared by the user in the same format.

EXAMPLE: bb22bb18

Card No. 2:

This is actually a deck of cards containing the multigroup cross sections. The format for each card is 1x,1P7E11.4. This deck is included as part of the input library and normally requires no preparation on the part of the user. If the user desires to use different cross sections, this deck must be replaced with a deck written in the same format. The cross section data supplied by the user must be in the following order. All cross section data for the highest energy neutron group must be first. This is followed by the data for the rest of the neutron groups in the order of descending energy. Next, the highest energy gamma group data follows. The data for the gamma groups must also be arranged in order of descending energy. Each group G must contain a series of cross sections whose number must be three more than the total number of groups (neutron plus gammas). In any group G, the cross sections must be arranged in the following order:

1 is transport

2 is removal

3 is total

4 is scatter (G to G)

5 is scatter (G-1 to G)

6 is scatter (G-2 to G)

7 is scatter (G-3 to G)

etc.

Every card, except for the final card in this deck must have seven numbers in the specified format. In cases where the group to group scatter cross section is zero or do not exist (example: G-8 when G is 3) a zero must be entered on the card. The cross section data used in the input library is listed in Appendix C.

Card No. 3:

This is also a deck of cards and contains the response functions required to translate the multigroup neutron fluence to silicon dose in rads. The format for each card is 1X,1P7E11.4. This deck is supplied as part of the input library. If the user desires to use different response functions, he must replace this deck. The number of values in this deck must correspond to the number of neutron groups specified on card number one. The response functions should be arranged in order of decreasing energy groups. The response functions used in the input library are presented in Table III. These functions were obtained from the SMAUG code.

Card No. 4:

This is also a deck of cards and contains the response functions required to translate the multigroup gamma fluence to silicon dose in rads. The format is the same as for card number three. This deck is supplied as part of the input library and if the user wishes to use different response functions, he must replace this deck. The number of values in this deck must correspond to the number of gamma groups

Table III

Neutron Response Functions

Energy Group	For Silicon Dose in rads	For Tissue Dose in rads
1	9.35E-10 (a)	6.2545E-09
2	9.66E-10	5.7744E-09
3	8.74E-10	5.1525E-09
4	5.56E-10	4.9958E-09
5	2.15E-10	4.4861E-09
6	1.41E-10	4.1199E-09
7	1.02E-10	4.0662E-09
8	8.4E-11	3.4281E-09
9	7.7E-11	3.1542E-09
10	6.9E-11	3.0504E-09
11	5.5E-11	2.6187E-09
12	4.8E-11	2.0131E-09
13	3.4E-11	1.2918E-09
14	0	4.2594E-10
15	0	1.9581E-11
16	0	3.6590E-12
17	0	1.1533E-12
18	0	1.0841E-12
19	0	1.5460E-12
20	0	2.6671E-12
21	0	4.3388E-12
22	0	8.2591E-12

(a) read as 9.35×10^{-10}

specified on card number one. The response functions used in the input library are presented in Table IV. These functions were obtained from the SMAUG code.

Card No. 5:

This card is the same as card number three except the response functions are for tissue dose in rads. All instructions are the same as for card number three. The response functions used in the input library are presented in Table III. These functions were obtained from the SMAUG code.

Card No. 6:

This card is the same as card number four except the response functions are for tissue dose in rads. All instructions are the same as for card number four. The response functions used in the input library are presented in Table IV. The functions were obtained from the SMAUG code.

Card No. 7:

This card contains one number in I4 format. This number is the number of aircraft being evaluated and must be between one and 100. This card must be supplied by the user.

EXAMPLE: bbb4

Card No. 8:

This may be one or more cards and supplies the aircraft position. The positions are defined in an X,Y,Z coordinate system (in that order) and the units on the coordinates are kilometers. Each card contains the coordinates for two aircraft in 1X,1P6E11.4 format and will normally have six numbers. A card could have three numbers (one aircraft

Table IV

Gamma Response Functions

Gamma Group	For Silicon Dose in rads	For Tissue Dose in rads
1	2.8E-09 (a)	2.2576E-09
2	2.28E-09	1.9371E-09
3	1.83E-09	1.6436E-09
4	1.48E-09	1.3901E-09
5	1.2E-09	1.1812E-09
6	9.85E-10	1.0102E-09
7	8.4E-10	8.8378E-10
8	7.12E-10	7.6726E-10
9	6.1E-10	6.6473E-10
10	5.05E-10	5.5115E-10
11	4.1E-10	4.4620E-10
12	2.7E-10	3.5710E-10
13	2.37E-10	2.6009E-10
14	1.65E-10	1.7915E-10
15	1.17E-10	1.2257E-10
16	7.25E-11	6.2561E-11
17	9.75E-11	3.2027E-11
18	4.13E-10	4.7209E-11

(a) read as 2.8×10^{-9}

position) if only one aircraft is listed on card number seven or if the card is the last card for an odd number of aircraft. The number of aircraft positions must agree with the number of aircraft specified on card number seven. This card (or cards) must be supplied by the user.

EXAMPLE:

```
bb1.0000E+00b1.0000E+00b1.0000E+01b0.0000E+00b5.0000E-01b1.5000E+00
bb1.2500E-01b1.0000E+00b2.0000E+01b0.0000E+00b0.0000E+00b2.3000E+00
```

Card No. 9:

This card contains one or two numbers in 2I4 format. The number specifies the units of aircraft vulnerability and must be a number from one to eight. If the first number is five or eight no second number is required. The first number gives the units of neutron vulnerability and the second number gives the units of gamma vulnerability. The meaning of the numbers follow:

Number	Meaning
1	Total neutron fluence (neutrons/cm^2)
2	Total gamma fluence (gammas/cm^2)
3	Neutron tissue dose (rads)
4	Gamma tissue dose (rads)
5	Neutron+gamma tissue dose (rads)
6	Neutron silicon dose (rads)
7	Gamma silicon dose (rads)
8	Neutron+gamma silicon dose (rads)

This card must be supplied by the user.

EXAMPLE: bbb1bbb2

Card No. 10:

This card contains one or two numbers in 1X,1P2E11.4 format. The numbers specify the aircraft vulnerability. The first number is the aircraft neutron vulnerability and the second number is the aircraft gamma vulnerability. The units must be those specified on card number nine. If the first number on card nine is a five or eight only one number is required on this card. This card must be supplied by the user.

EXAMPLE: bb1.0000E+10b1.0000E+10

Card No. 11:

This card contains four numbers in 1X,1P4E11.4 format. The first number is the yield of the nuclear weapon in kilotons. The next three numbers give the position of the burst in X,Y,Z coordinates (in that order). The units of the coordinates are kilometers. This card must be supplied by the user.

EXAMPLE: bb1.0000E+01b1.2500E-01b1.0000E-01b2.5000E+01

Card No. 12:

This card may contain two numbers in 2I4 format and specifies the type of output desired. The first number may be blank (or zero), one, two, or three. This number specifies the type of plot output. A blank means no plots will be output. A one means the vulnerability isofluence or isodose line will be plotted and the aircraft will be located on the plot. A two means only isofluence or isodose lines will be plotted. A three means both the one and two options will be plotted. The second number may be blank (zero) or one. This number specifies the type of printed output. A blank is the normal mode and will result in a short printout giving the location of the burst, location of the aircraft, and if the aircraft survived with gamma and neutron levels experienced by the aircraft. A one will give the same output as the blank plus the detailed group by group fluence at every mesh point. The one option generates several hundred pages and should not be used unless the group fluences are needed. This card must be supplied by the user.

EXAMPLE: bbb3bbbb

Card No. 13:

This card may contain one to three numbers in 3I4 format. The first number must be a one or two and specifies the type of nuclear weapon. A one is a fission weapon and a two is a thermonuclear weapon. The second number may be a blank or a one and specifies the source of the weapon output spectrum for neutrons. If this number is blank, an unclassified default spectrum will be used. This default spectrum is contained within the code and will be a fission or thermonuclear spectrum depending upon the first number on this card. Table V lists the default neutron spectra. If this number is one, the default spectrum will not be used and a user supplied spectrum will be used. The third number may be a blank or a one and specifies the source of the weapon output spectrum for gammas. If this number is blank, an

Table V

Weapon Neutron Spectra

Neutron Group	Fission Spectrum (neutrons/kiloton)	Thermonuclear Spectrum (neutrons/kiloton)
1	3.92E+19 (a)	6.001E+22
2	2.233E+20	2.176E+22
3	8.7E+20	1.1985E+22
4	3.48E+21	1.2495E+22
5	8.705E+21	1.4875E+22
6	8.705E+21	1.4875E+22
7	1.4951E+22	1.4167E+22
8	1.4951E+22	1.4167E+22
9	1.4951E+22	3.825E+22
10	4.23E+22	3.825E+22
11	4.23E+22	3.825E+22
12	4.2325E+22	7.9475E+22
13	4.2325E+22	7.9475E+22
14	3.875E+21	3.1025E+23
15	0	0
16	0	0
17	0	0
18	0	0
19	0	0
20	0	0
21	0	0
22	0	0

(a) read as 3.92×10^{19}

unclassified default spectrum will be used. This default spectrum is contained within the code and will be a fission or thermonuclear spectrum, depending upon the first number on this card. Table VI lists the default gamma spectrum. If this number is one, the default spectrum will not be used and a user supplied spectrum will be used. The default spectra are from the SMAUG computer code. If both the second and third numbers are blank, this is the last card required. This card must be supplied by the user.

Card No. 14:

If the second number on card 13 was blank, omit this card. If that number was a one, this card (actually a small deck) will contain the neutron output spectrum of the weapon in 1X,1P7E11.4 format. The number of entries must agree with the number of neutron groups specified on card one. The group structure must be arranged in order of descending energy.

Card No. 15:

If the third number on card 13 was blank, omit this card. If that number was a one, this card (or several cards) will contain the gamma output spectrum of the weapon in 1X,1P7E11.4 format. The number of entries must agree with the number of gamma groups specified on card number one. The group structure must be arranged in order of descending energy.

Card 15 is the last data card that may be supplied by the user. As can be noted in the above description of the

Table VI

Weapon Gamma Spectra

Gamma Group	Fission Spectrum (gammas/kiloton)	Thermonuclear Spectrum (gammas/kiloton)
1	1.2648E+19 (a)	6.3239E+18
2	5.9019E+19	2.9508E+19
3	1.0539E+20	5.2693E+19
4	4.7555E+20	2.3778E+20
5	4.7556E+20	2.3778E+20
6	1.0718E+21	5.3595E+20
7	1.0719E+21	5.3595E+20
8	2.3562E+21	1.1781E+21
9	3.22E+21	1.615E+21
10	4.0838E+21	2.0419E+21
11	3.7741E+21	1.887E+21
12	4.512E+21	2.2559E+21
13	5.2499E+21	2.6249E+21
14	2.2981E+21	1.149E+21
15	2.2981E+21	1.149E+21
16	2.7062E+21	1.3531E+21
17	0	0
18	0	0

(a) read as 1.2648×10^{19}

cards, this code is flexible and will allow the input library and contained data to be replaced. However, if any of the options of replacing cross sections, response functions, or weapon spectrum are exercised, the user should note the following caution. These three selections of data are self-consistant in number of groups and the energy range in each group. If any one (or more) are replaced, this self-consistency must be retained. That is, the user is responsible for insuring the number of groups, and the energy ranges in each group, agree.

End of Job Cards

For the Wright-Patterson Air Force Base CDC 6600, the user must supply two more cards following the data card. The first is a multipunched 7/8/9 in column one. The second is a multipunched 6/7/8/9 in column one. This last card is orange. Other computers will require different end of job cards.

IV. Results

This chapter presents the results obtained from the sample problem illustrated in Chapter III. These results are compared with results obtained from a constant density air model with the same input data. The results are also compared to those given in the Quick-Look Radiation charts (Ref 1). These charts were derived from SMAUG; therefore, this comparison is effectively between the author's code and SMAUG.

The Problem

The problem presented in Chapter III (as sample entries) is reviewed in this paragraph.

<u>Weapon Parameters</u>

Position: X = 0.125 km, Y = 0.1 km, Z = 25 km

Yield: 10 kilotons

Type: Fission

Neutron spectrum: Input library (see Table V)

Gamma spectrum: Input library (see Table VI)

<u>Aircraft Parameters</u>

Number: Four

Positions: No. 1, X = 1 km, Y = 1 km, Z = 10 km

No. 2, X = 0 km, Y = 0.5 km, Z = 1.5 km

No. 3, X = 0.125 km, Y = 1 km, Z = 20 km

No. 4, X = 0 km, Y = 0 km, Z = 2.3 km

Neutron vulnerability: 10^{10} neutrons/cm^2

Gamma vulnerability: 10^{10} gammas/cm^2

Air Parameters

Cross sections: the 22 group neutron, 18 group coupled gamma library (see Appendix C)

Response functions: Input library (see Tables III and IV)

The Output from the Code

The output required was the short printed output which states what happened to the aircraft and four plots; two showing neutron and gamma isofluence lines, and two showing the neutron and the gamma isovulnerability line. The printed output stated that aircraft number one survived and experienced a neutron fluence of 4.8×10^6 neutrons/cm^2 and a gamma fluence of 5.4×10^6 gammas/cm^2. Aircraft numbers two and four survived and experienced zero neutron and gamma fluences. Aircraft number three was killed by neutrons with a fluence of 2.2×10^{12} neutrons/cm^2. The four plots are presented as Figs. 2 through 5.

Comparison with Results from a Constant Density Air Model

Another air model was selected for comparison purposes. This model was a constant density, homogeneous composition, infinite atmosphere model. The same input data was used. A program was written for this model with the same requirement on output. The printed output stated all four aircraft

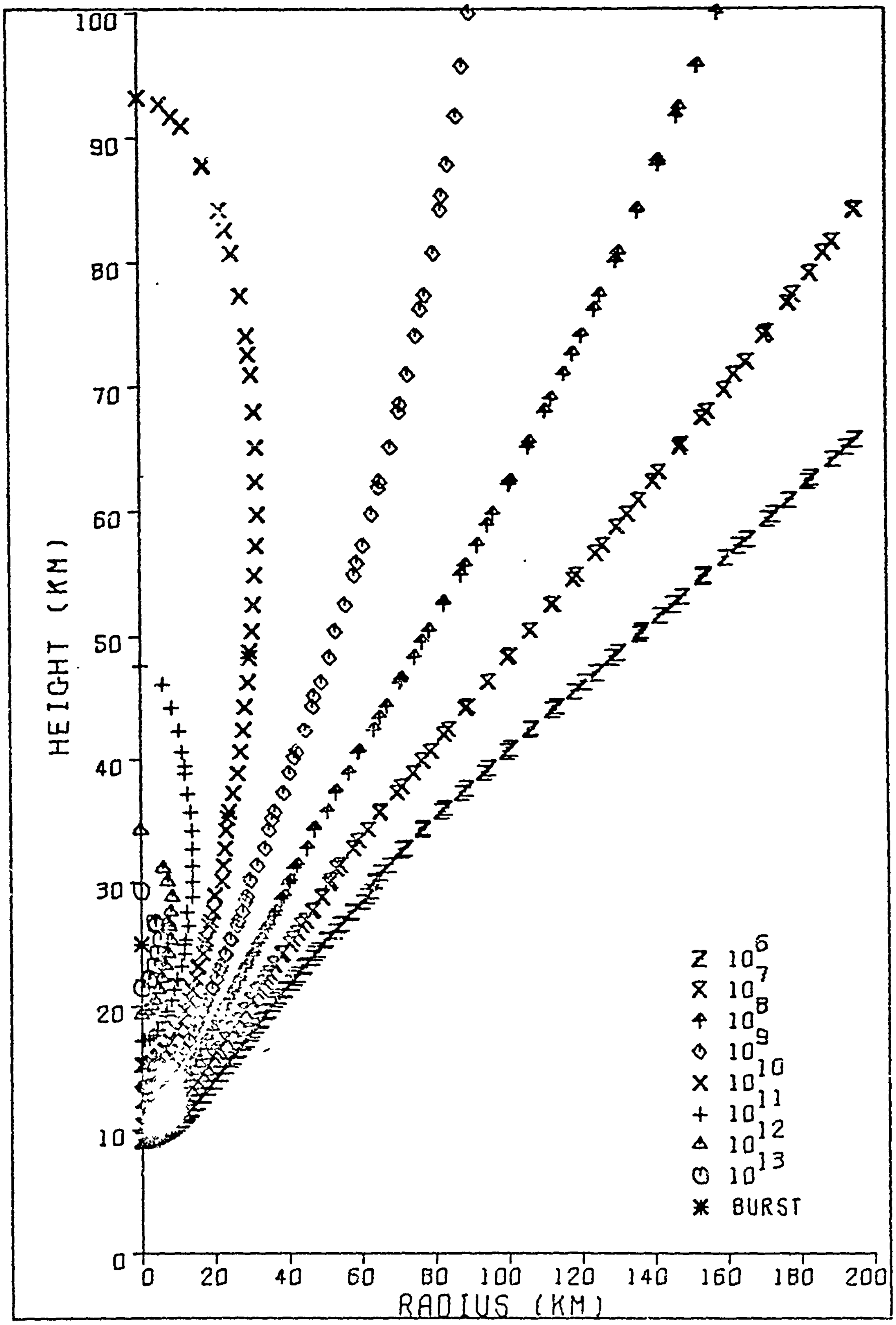

FIG. 2 NEUTRON ISOFLUENCE LINES (N/CM2)
EXPONENTIAL AIR

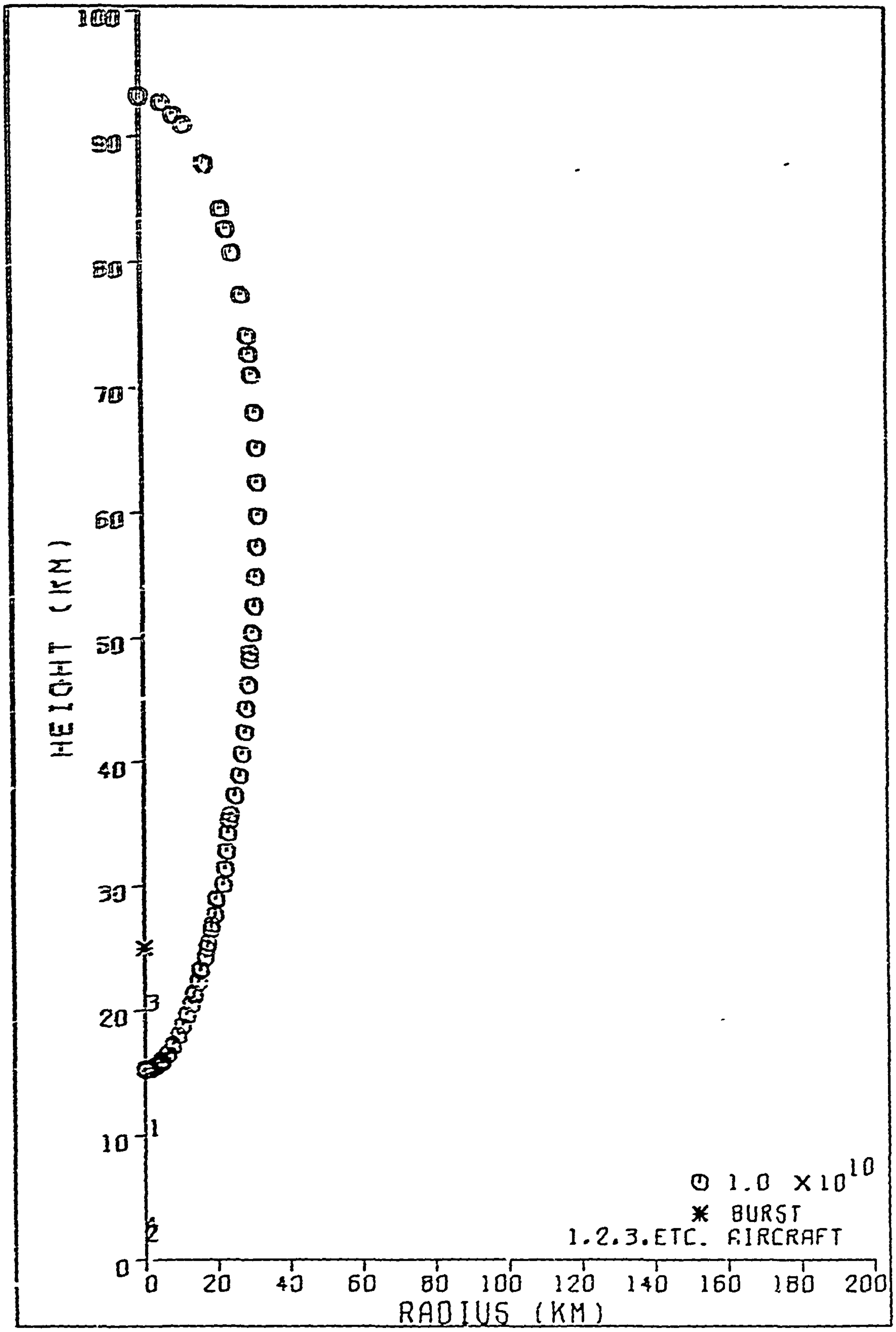

FIG. 3 NEUTRON FLUENCE VULNERABILITY LINE (N/CM2) EXPONENTIAL AIR

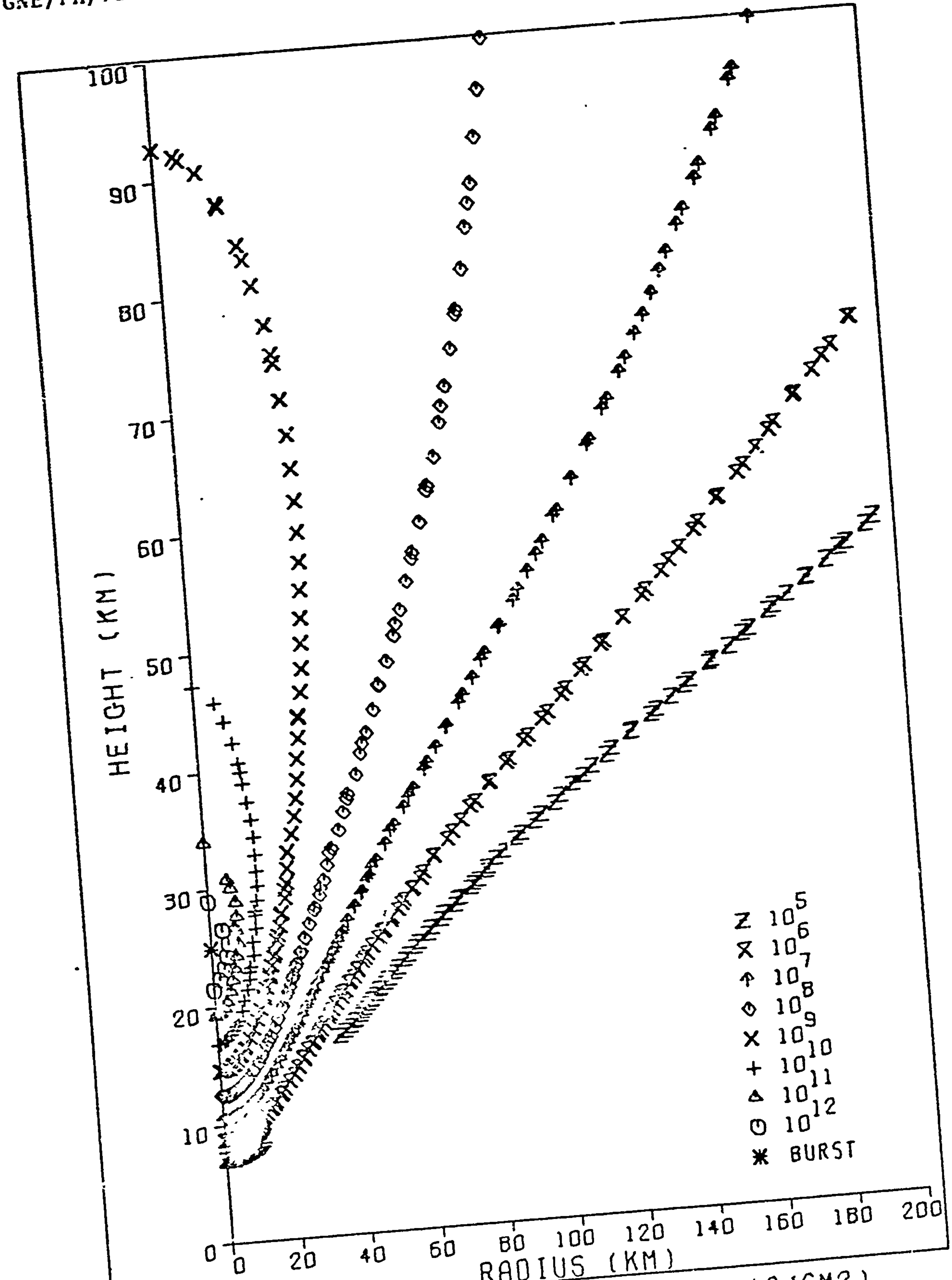

FIG. 4 GAMMA ISOFLUENCE LINES (G/CM2)
EXPONENTIAL AIR

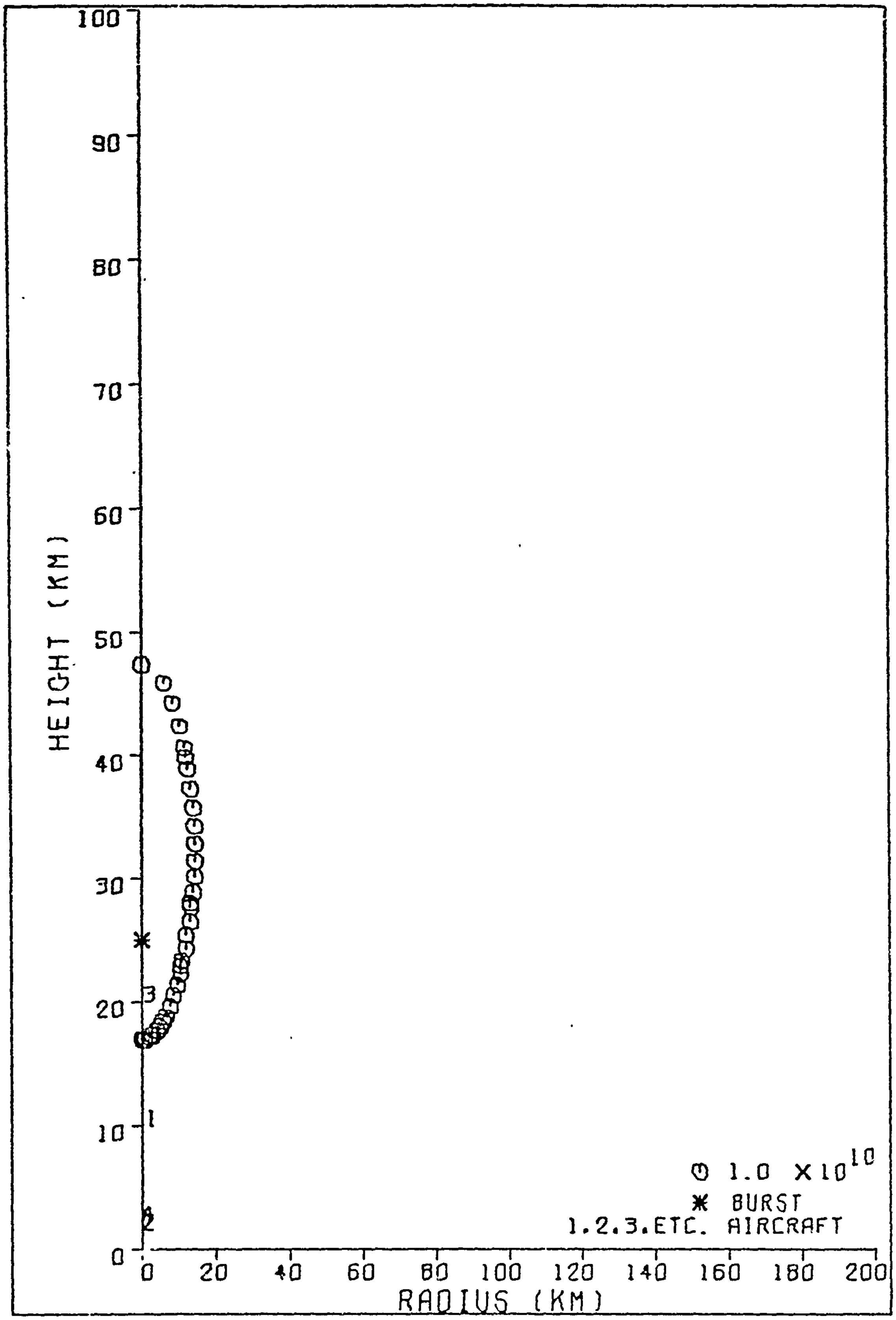

FIG. 5 GAMMA FLUENCE VULNERABILITY LINE (G/CM2) EXPONENTIAL AIR

were killed. Aircraft number one experienced a neutron fluence of 6.6×10^{11} neutrons/cm^2, aircraft number two experienced a neutron fluence of 1.9×10^{11} neutrons/cm^2, aircraft number three experienced a neutron fluence of 3.1×10^{12} neutrons/cm^2, and aircraft number four experienced a neutron fluence of 2.1×10^{11} neutrons/cm^2. The four plots are presented as Figs. 6 through 9.

The neutron fluences calculated at the four aircraft positions are greater for the constant density model. This difference, however, is expected since, in the actual atmosphere, air density is decreasing with altitude. Therefore, in the actual atmosphere, any fixed position below the burst should experience a lower fluence than that predicted for a constant density model. The exponential air model selected for the code is a reasonable approximation to the actual atmosphere, therefore, the fluences calculated with this code for fixed positions below the burst should be lower than those predicted for the constant air model.

Two observations can be made by examining the plots from the two codes. Figure 6, the neutron isofluence lines for the constant density model, and Fig. 2, the neutron isofluence lines for the exponential air model, show the effect of the atmospheric model on neutron fluence. The isofluence lines for the constant density model are circles. The isofluence lines for the exponential air model are not circles. These lines are close together in the region of greater air density and diverge as the air density decreases.

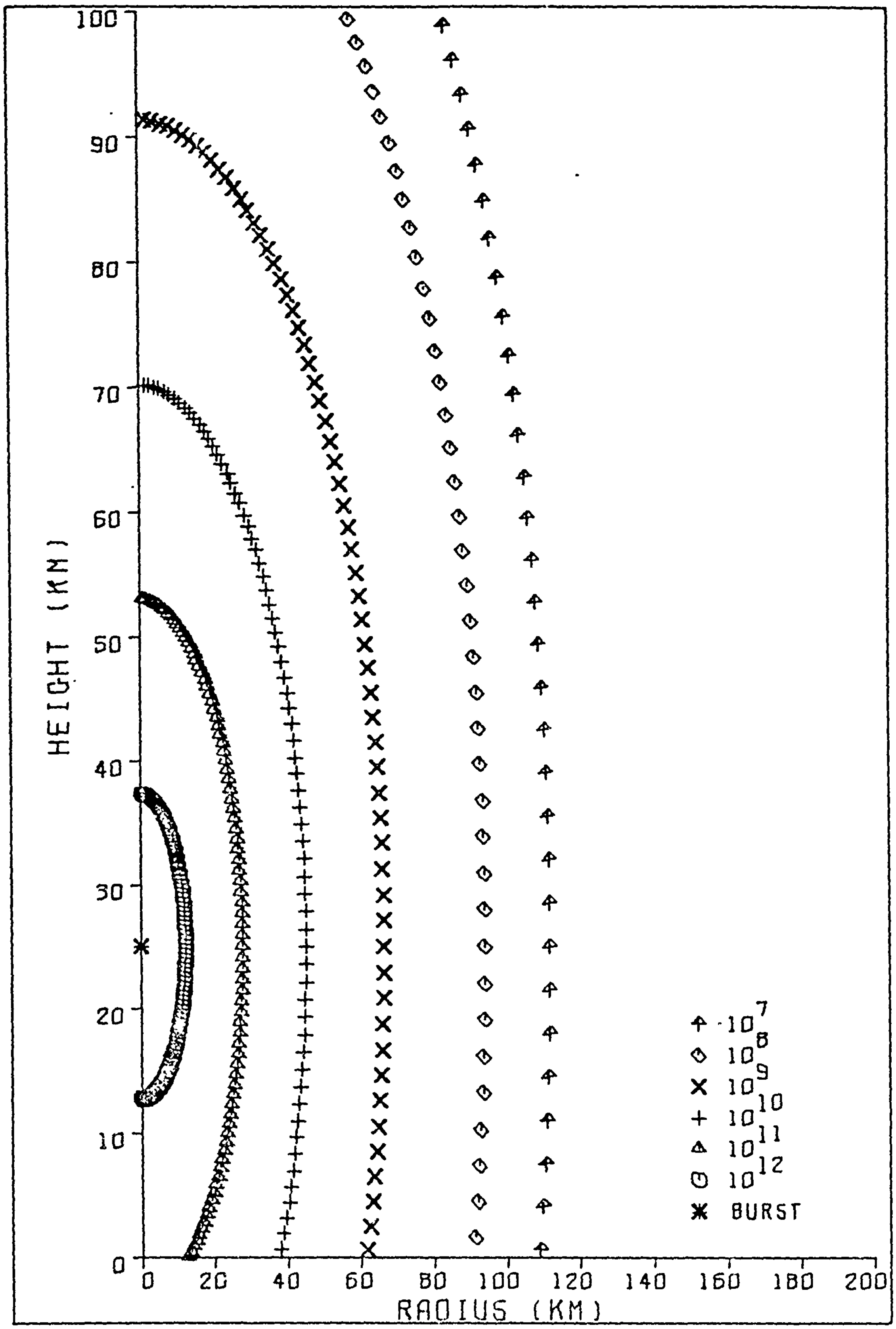

FIG. 6 NEUTRON ISOFLUENCE LINES (N/CM2)
CONSTANT DENSITY AIR

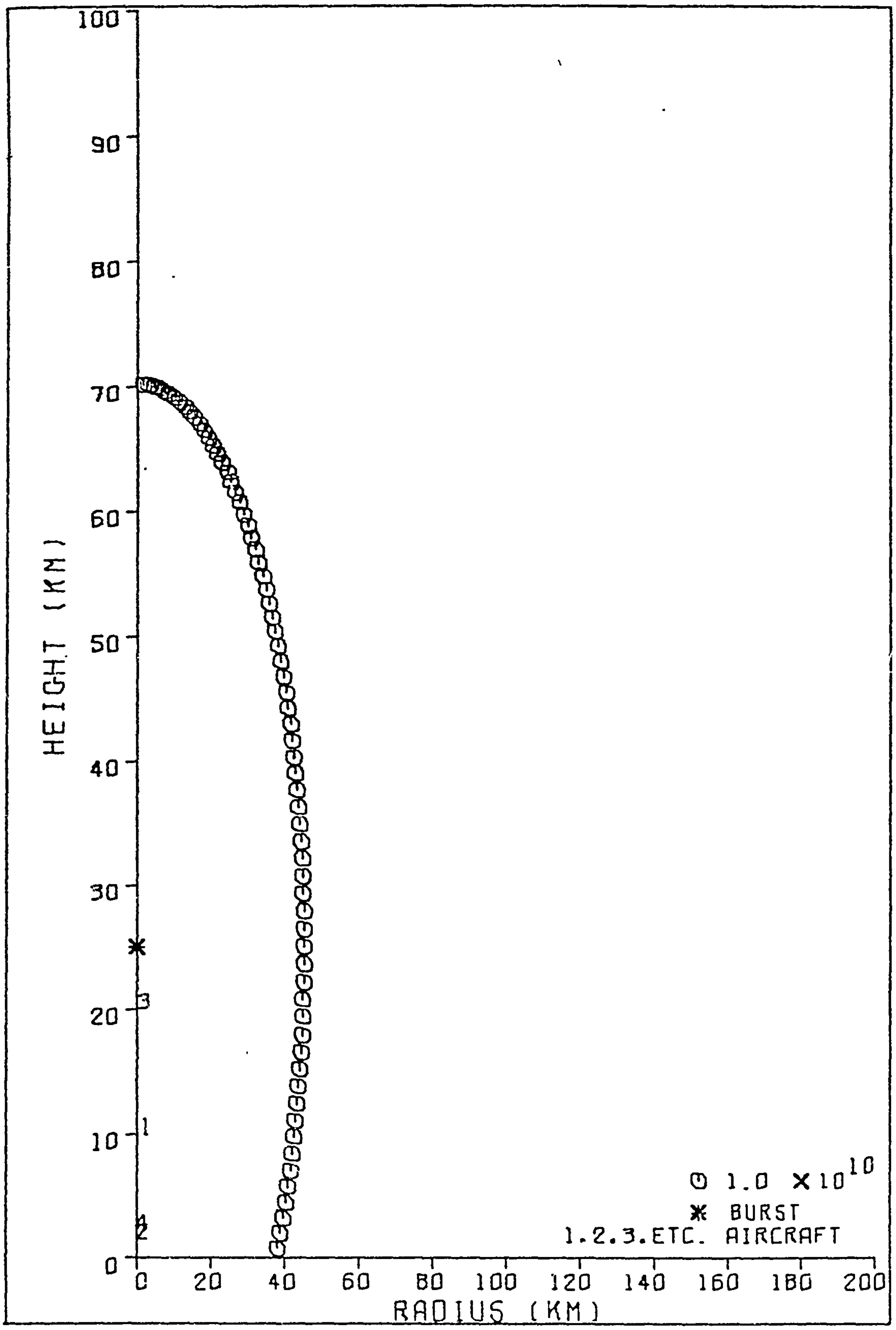

FIG. 7 NEUTRON FLUENCE VULNERABILITY LINE (N/CM2) CONSTANT DENSITY AIR

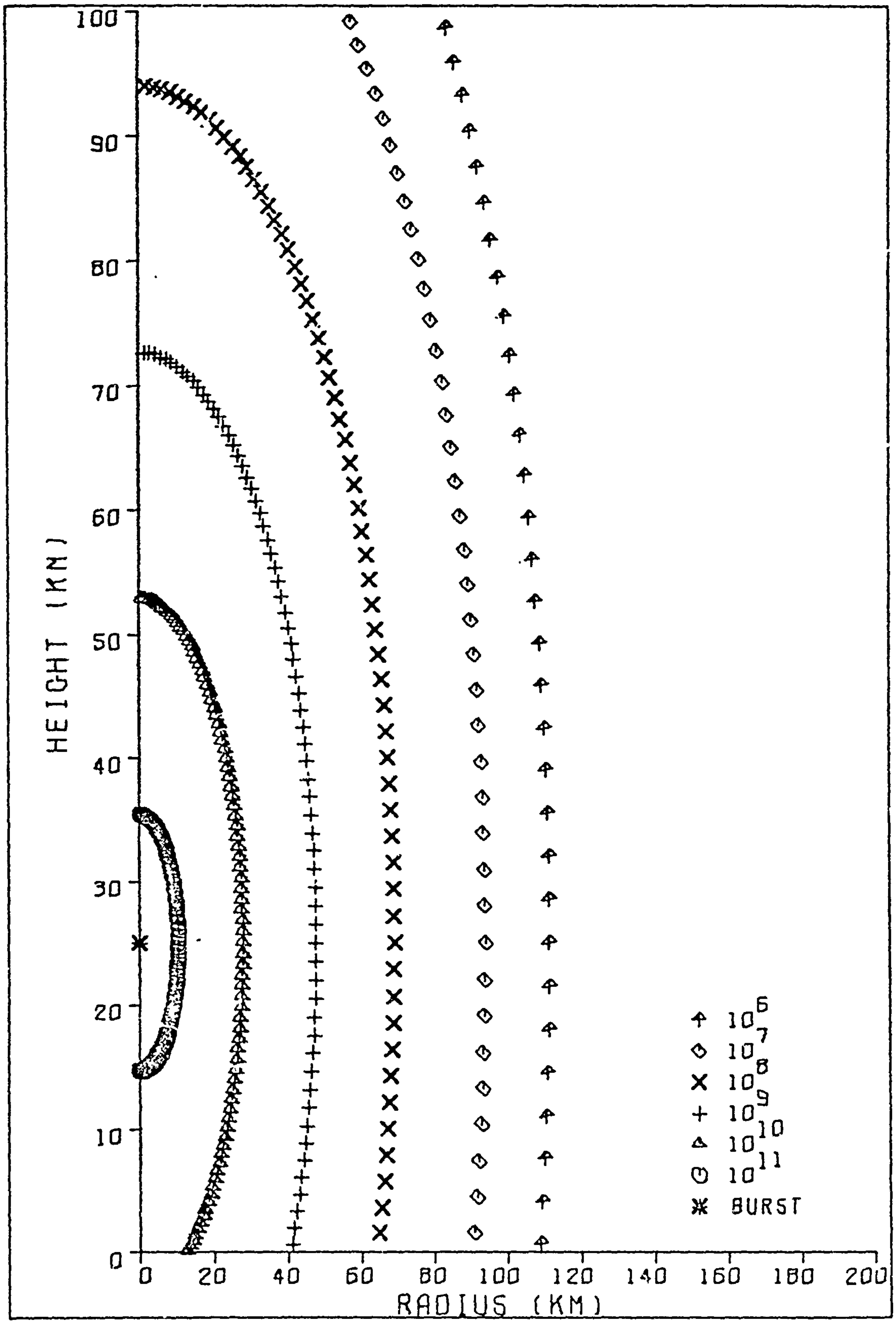

FIG. 8 GAMMA ISOFLUENCE LINES (G/CM2)
CONSTANT DENSITY AIR

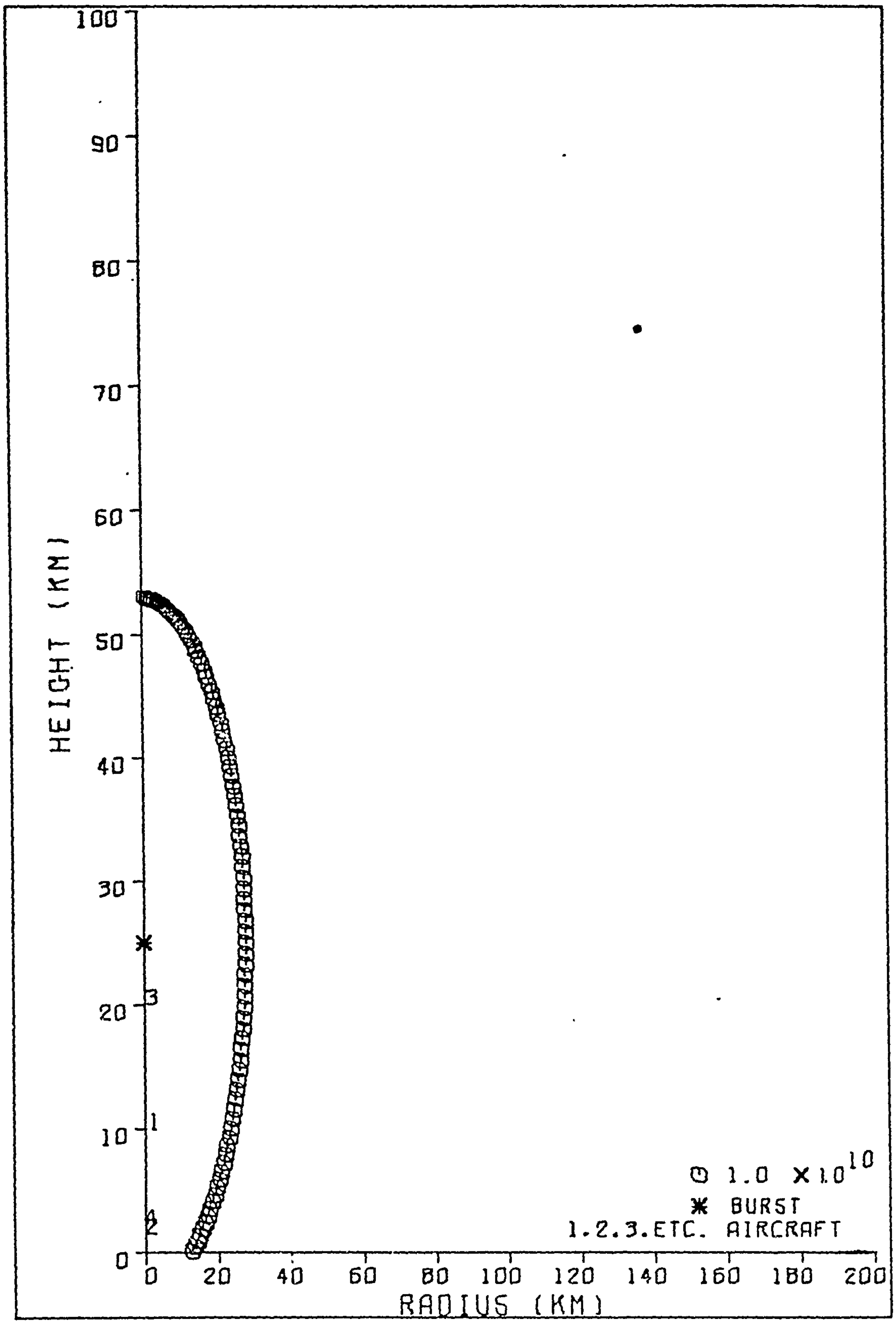

FIG. 9 GAMMA FLUENCE VULNERABILITY LINE (G/CM2) CONSTANT DENSITY AIR

This observation can also be made by comparing Figs. 4 and 8, the gamma isofluence lines.

The second observation that can be made is that the isofluence lines at a height coaltitude with the burst differ between the two models. This difference can be easily seen by comparing Figs. 3 and 7, the neutron fluence vulnerability line, or by comparing Figs. 5 and 9, the gamma fluence vulnerability line. From Fig. 3, the vulnerability level of 10^{10} neutrons/cm^2 is at about 18 km coaltitude for exponential air. From Fig. 7, the vulnerability level of 10^{10} neutrons/cm^2 is at about 44 km. Figures 5 and 9 show similar results for the gamma vulnerability line of 10^{10} gammas/cm^2. The coaltitude distance for exponential air is about 12 km but for constant density air is about 25 km. Therefore, in this example, the constant density air model overpredicts the coaltitude distance by a factor of two.

Comparison with the Quick-Look Radiation Charts

Neutron and gamma fluences at the aircraft positions were also calculated using the Quick-Look Radiation Charts (Ref 1). These charts were derived from SMAUG. The expected values from these charts should be between the exponential air model and the constant density air model because the SMAUG data base is from a constant density model. SMAUG compensates for the actual change in air density by mass integral scaling along the line of sight between the burst and the receiver.

The fluence values from these charts were 1.6×10^8 neutrons/cm^2 and 3.8×10^8 gammas/cm^2 for aircraft number one, 3×10^{12} neutrons/cm^2 and 3×10^{11} gammas/cm^2 for aircraft number three. Values for aircraft numbers two and four could not be obtained because they were out of the range of the charts. The results of all these calculations are summarized in Tables VII and VIII.

Table VII

Neutron Fluences at the Aircraft Positions as Calculated by the Three Models

Aircraft Number	Neutron Fluence (neutrons/cm^2)		
	Exponential Air	Quick-Look Charts	Constant Density Air
1	4.8×10^6	1.6×10^8	6.6×10^{11}
2	0	(a)	1.9×10^{11}
3	2.2×10^{12}	3×10^{12}	3.1×10^{12}
4	0	(a)	2.1×10^{11}

(a) Out of the range of the charts

The comparison between the three calculations do show the Quick-Look results between those of exponential air and constant density air.

Validity of Results

The general appearance of the results seem valid. The results show the expected effect of the decreasing air density with height. Furthermore, when these results are

Table VIII

Gamma Fluences at the Aircraft Positions

as Calculated by the Three Models

Aircraft Number	Gamma Fluence (gammas/cm^2)		
	Exponential Air	Quick-Look Charts	Constant Density Air
1	5.4×10^6	3.8×10^8	8×10^{10} (c)
2	0	(b)	3×10^{10} (c)
3	10^{11} (a)	3×10^{11}	$> 10^{11}$ (c)
4	0	(b)	3×10^{10} (c)

(a) Not calculated, estimated from Fig. 4
(b) Out of range of the charts
(c) Not calculated, estimated from Fig. 8

compared to the constant density air model and to the Quick-Look charts, they appear to be correctly predicting increased (or decreased) fluences where applicable. The actual validity of the numbers can not be determined since no experimental observations exist for these type of calculations. Some checks, other than comparison with experimental data, can be made to estimate the validity of the results. First, a check for conservation of particles can be made. The conservation relationship is that the number of particles entering a specified volume plus the number of particles gained inside the volume is equal to the number of particles leaving the volume plus the number of particles lost inside the volume. This check, once made, would establish that the difference equations used in this report do conserve particles and are therefore valid. Another check that can be made is to let the atmospheric model used approach a constant density model. This can be done by changing the scale height of the atmosphere to a very large number. If the same example used in this report is then run, the results should approach those obtained from a constant density model.

V. Conclusions and Recommendations

Conclusions

Definite conclusions can not be made since only one sample problem has been run on the computer code. One tentative conclusion, however, can be made subject to confirmation by more computer runs. The codes that consider coaltitude burst and receiver, if based on the constant density air assumption, overestimate the gamma and neutron fluence. The reason for the overestimation is most likely the implicit assumption that scatter from above is equal to scatter from below in the constant density air model. In actuality, due to the density variation, scatter from above would be less than scatter from below. Therefore, the exponential model correctly indicates lower coaltitude fluences.

Another conclusion that can be drawn is that the constant density air model poorly estimates the neutron and gamma fluences at any fixed point above or below the burst. This estimate becomes progressively poorer as the distance from the burst increases.

In conclusion, an alternate, and successful, approach to the calculation of neutron and gamma fluences in the atmosphere has been developed. This code also does determine if aircraft in the vicinity of the burst survive the neutron and gamma fluences. However, the requirement that a quick and therefore inexpensive code be developed has not been fully satisfied. The running time of the code, 20 minutes of central processor time plus one hour of input/output time, does not compete favorably with SMAUG's running time of several seconds, but it is much faster than the Monte Carlo codes mentioned in Straker's report (Ref 5).

Recommendations

Based on the above conclusions, the author has five recommendations. The first is that a conservation check be performed, using the geometry developed, to insure the validity of the difference equations. The second is that the code be rerun with an increased scale height to determine if the results do approach those obtained with a constant density air model. The third recommendation is that, if the above checks are successfully made, the code be rerun at heights of burst for which Monte Carlo data exists and a comparison made between the Monte Carlo results and those obtained from this code. The fourth recommendation is that, if the above three checks still show the code to be valid, the code be given to an experienced programmer to revise in order to decrease the run time. The final recommendation is to then run the revised code at a series of burst heights in order to generate a broad data base. This data base can then be used to write a SMAUG-like program which should fully satisfy the requirement for an inexpensive (quick running) code.

Bibliography

1. Brock, Terrence K. Quick-Look Radiation Charts. AFWL-TR-71-84. Kirtland Air Force Base, New Mexico: Air Force Weapons Laboratory, August 1971.

2. DeRaad, Robert G. CSSANE A Code for System Survivability Analysis - Nuclear Effects. Unpublished Thesis. Wright-Patterson Air Force Base, Ohio: Air Force Institute of Technology, June 1972.

3. DLC-14. Air: Sample Input to ANISN for Calculation of Neutron and Secondary Gamma-Ray Transport in Air. Oak Ridge, Tennessee: Oak Ridge National Laboratory, June 1971.

4. Murphy, Harry M., Jr. A Users Guide to the SMAUG Computer Code. Technical Note No. SAA-TN-71-2. Kirtland Air Force Base, New Mexico: Air Force Weapons Laboratory, January 1971.

5. Straker, E. A. Status of Neutron Transport in the Atmosphere. ORNL-TM-3065. Oak Ridge, Tennessee: Oak Ridge National Laboratory, July 1970.

6. Straker, E. A., and M. L. Gritzner. Neutron and Secondary Gamma-Ray Transport in Infinite Homogeneous Air. ORNL-4464. Oak Ridge, Tennessee: Oak Ridge National Laboratory, December 1969.

7. Valley, Shea L. Handbook of Geophysics and Space Environments. Air Force Cambridge Research Laboratories, Office of Aerospace Research, United States Air Force, 1965.

8. Winchester, Robert O. Calculation of Neutron and Neutron Induced Gamma Fluences from a Point Source in the Atmosphere. Unpublished Thesis. Wright-Patterson Air Force Base, Ohio: Air Force Institute of Technology, June 1969.

9. French, R. L. "A First-Last Collision Model of the Air/Ground Interface Effects on Fast-Neutron Distributions." Nuclear Science and Engineering, 19: 151-157 (1964).

10. Harris, R. J., Jr., J. A. Lonergan, and L. Huszar. Models of Radiation Transport in Air - The ATR Code. Document SAI-71-557-LJ. P. O. Box 2351, 125 Prospect St., La Jolla, Calif.: Science Applications, Inc., November 1971.

Appendix A

An Algorithm for the Solution of Matrix Equations with Block Tri-Diagonal Matrix of Coefficients

A matrix equation of the form

$$\underline{A}\underline{F} = \underline{S} \tag{1}$$

where $\underline{A}$ is an N x N matrix, $\underline{F}$ is an N x 1 matrix, and $\underline{S}$ is an N x 1 matrix, can be directly solved for $\underline{F}$ by finding $\underline{A}$ inverse and multiplying $\underline{S}$ by $\underline{A}$ inverse. The solution is therefore

$$\underline{F} = \underline{A}^{-1}\underline{S} \tag{2}$$

This is the normal method of solution used in most computer programs. However, when N is large, this method is impractical. Two reasons for the impracticality of this method are the large computer core and the long computer run time required. For example, if N is 1000, a computer core of 1,000,000 words will be required just to store the inverse. The core requirement can be reduced by using magnetic tapes for storage; however, the use of magnetic tapes increases the computer run time. Numerous algorithms have therefore been developed to avoid the problems of solution by direct inversion.

These algorithms depend on the composition of the coefficient matrix $\underline{A}$. If $\underline{A}$ is a block tri-diagonal matrix, an algorithm has been described by Winchester (Ref 8:57-61) that greatly reduces the computer core size and run time to solve Eq (1). This algorithm is described in the following paragraphs.

If $\underline{A}$ is block tri-diagonal, it can be partitioned such that

$$\underline{A} = \begin{pmatrix} DIA_1 & UP_1 & 0 & 0 & \cdot & \cdot & 0 \\ LOW_2 & DIA_2 & UP_2 & 0 & \cdot & \cdot & 0 \\ 0 & LOW_3 & DIA_3 & UP_3 & \cdot & \cdot & 0 \\ \vdots & & & & & & \vdots \\ 0 & \cdot & \cdot & \cdot & LOW_{N-1} & DIA_{N-1} & UP_{N-1} \\ 0 & \cdot & \cdot & \cdot & 0 & LOW_N & DIA_N \end{pmatrix} \qquad (3)$$

where DIA_i, UP_i and LOW_i are all square matrices. $\underline{F}$ and $\underline{S}$ can also be partitioned such that

$$\underline{F} = \begin{pmatrix} FM_1 \\ FM_2 \\ \vdots \\ FM_N \end{pmatrix} \qquad (4)$$

$$\underline{S} = \begin{pmatrix} SM_1 \\ SM_2 \\ \vdots \\ SM_N \end{pmatrix} \qquad (5)$$

where SM_i and FM_i are also matrices

Next, factor $\underline{A}$ into matrices WM and QM such that

$$WM\ QM = \underline{A} \tag{6}$$

where

$$WM = \begin{pmatrix} W_1 & 0 & 0 & 0 & . & 0 \\ LOW_2 & W_2 & 0 & 0 & . & 0 \\ 0 & LOW_3 & W_3 & 0 & . & 0 \\ \vdots & & & & & \vdots \\ 0 & . & 0 & LOW_{N-1} & W_{N-1} & 0 \\ 0 & . & 0 & 0 & LOW_N & W_N \end{pmatrix} \tag{7}$$

and

$$QM = \begin{pmatrix} I & q_1 & 0 & . & . & 0 \\ 0 & I & q_2 & . & . & 0 \\ \vdots & & & & & \vdots \\ 0 & . & . & 0 & I & q_{N-1} \\ 0 & . & . & 0 & 0 & I \end{pmatrix} \tag{8}$$

Carrying out the matrix multiplication of Eq (6) and equating components to corresponding diagonal components of $\underline{A}$ yields the following relations:

$$W_1 = DIA_1 \tag{9}$$

$$W_i = DIA_i - LOW_i\ q_{i-1} \tag{10}$$

By equating components with corresponding off-diagonal components of $\underline{A}$ the following relation is also obtained:

$$q_i = (W_i)^{-1} UP_i \tag{11}$$

Next, define a matrix GM

$$GM = \begin{pmatrix} G_1 \\ G_2 \\ \cdot \\ \cdot \\ \cdot \\ G_N \end{pmatrix} \tag{12}$$

such that

$$WM\ GM = \underline{S} \tag{13}$$

Therefore

$$GM = QM\ \underline{F} \tag{14}$$

Carrying out the matrix multiplication of Eq (13) and equating the components with corresponding components of $\underline{S}$ yields the following relations:

$$G_1 = (W_1)^{-1}\ SM_1 \tag{15}$$

$$G_i = (W_i)^{-1}(SM_i - LOW_i G_{i-1}) \quad 1 < i \leq N \tag{16}$$

The same operation with Eq (14) yields

$$FM_N = G_N \tag{17}$$

$$FM_i = G_i - q_i FM_{i+1} \quad 1 \leq i < N \tag{18}$$

Equations (14) and (15) are the equations defining the unknown matrix $\underline{F}$.

The entire algorithm to solve for $\underline{F}$ is listed below.

(1) Start with i = 1

(2) Solve for W_i

$W_1 = DIA_1$

$W_i = DIA_i - LOW_i q_{i-1} \quad 1 < i \leq N$

(3) Solve for q_i

$q_i = (W_i)^{-1} UP_i$

(4) Solve for G_i

$G_1 = (W_1)^{-1} SM_1$

$G_i = (W_i)^{-1} (SM_i - LOW_i G_{i-1}) \quad 1 < i \leq N$

(5) Return to Step (2) with next i and repeat until all components are calculated.

(6) After all the q_i matrices and the G_i matrices have been calculated, solve for the FM_i components of the unknown matrix $\underline{F}$. Start with i = N.

$FM_N = G_N$

$FM_i = G_i - q_i \, FM_{i+1} \quad 1 \leq i < N$

Appendix B

Description and Listing of the Code

Introduction

This appendix describes and presents the code. Each subroutine is described separately with a glossary and listing. Some program modifications that can be made with the exchange or addition of a few cards are also discussed. The code is written for the Wright-Patterson Air Force Base CDC 6600 in FORTRAN Extended. In addition, the code, as written, should only be run on a 1700 terminal of the CDC 6600 that has on-line plot capability. The program can easily be converted to run at the computer center, rather than at a terminal, with a few easy modifications. These modifications are discussed in subroutine MAP. If the code is run on another CDC computer, the library matrix and plot subroutines used by the code may be different and require modifications to the code.

Program MAIN

This is the main program and is a substitute for the master nuclear effects program developed by Capt DeRaad (Ref 2). The program is responsible for feeding the cross section data, response function data, aircraft data, weapon yield and position, and output options to subroutine GAMNEUT.

FORTRAN Glossary:

ACPOS	An array that stores the aircraft positions in X, Y, Z coordinates. Units are kilometers.
ARRAY	An array that specifies a live aircraft or one killed. A one means the aircraft is live while a zero means the aircraft has been killed.
BURST	An array that store the burst position in X, Y, Z coordinates. Units are kilometers.
LMAP	A number indicating the plot option.
LOUT	A number indicating the printed output option.
MODE	An array indicating the units of the figures given for aircraft vulnerability.
NGAM	The number of gamma groups.
NGROUP	The number of gamma plus neutron groups.
NNEUT	The number of neutron groups.
NUMBER	The starting number of aircraft.
STMAX	The maximum macroscopic total air cross section.
STMIN	The minimum macroscopic total air cross section.
VUL	An array that stores the aircraft neutron and gamma vulnerability.
YIELD	The weapon yield in kilotons.

```
      PROGRAM MAIN(INPUT,TAPE9,TAPE10,TAPE11,OUTPUT,PLOT)
      INTEGER ARRAY(100)
      DIMENSION ACPOS(3,100),BURST(3),VUL(2),MODE(2)
      COMMON I
C         READ IN THE NUMBER OF GAMMA AND NEUTRON
C         GROUPS.  READ IN THE GROUP CROSS SECTIONS.
C         READ IN NEUTRON AND GAMMA RESPONSE FUNCTIONS.
C
      CALL CROSS(STMAX,STMIN,NNEUT,NGAM,NGROUP)
      CALL DOSE(NNEUT,NGAM)
C
C         READ IN AIRCRAFT DATA, THE NUMBER OF AIRCRAFT,
C         POSITIONS, VULNERABILITY.  ALSO SET UP AN
C         ARRAY TO KEEP TRACK OF AIRCRAFT THAT ARE
C         KILLED.
C
      CALL AIRCR(ARRAY,ACPOS,VUL,MODE,NUMBER)
C
C         READ IN BURST LOCATION AND YIELD.
C
      READ 100,YIELD,(BURST(I),I=1,3)
C
C         READ IN PLOT AND OUTPUT OPTIONS.
C
C         LMAP=0 (BLANK) MEANS NO PLOT.
C         LMAP=1 MEANS A VULNERABILITY ISOFLUENCE OR
C              DOSE LINE WILL BE PLOTTED AND THE
C              AIRCRAFT LOCATED ON THE PLOT.
C         LMAP=2 MEANS THE ISOFLUENCE LINES WILL
C              BE PLOTTED.
C         LMAP=3 MEANS BOTH LMAP=1 AND LMAP=2 OPTIONS
C                WILL BE PLOTTED.
C
C         LOUT=0 (BLANK) MEANS SHORT (NORMAL) OUTPUT.
C         LOUT=1 MEANS A DETAILED OUTPUT.
      READ 101,LMAP,LOUT
C
C         CALCULATE THE NEUTRON AND GAMMA ENVIRONMENT.
C
      CALL GAMNEUT(NUMBER,ACPOS,ARRAY,BURST,VUL,MODE,NNEUT,NGAM,STMAX,ST
     AMIN,LMAP,LOUT,NGROUP,YIELD)
  100 FORMAT(1X,1P5E11.4)
  101 FORMAT(10I4)
      STOP
      END
```

Subroutine CROSS

This subroutine reads in the sea level macroscopic cross sections, determines the minimum and maximum total cross sections, and stores all cross sections on a disk file. This subroutine should be included in the master nuclear effects program.

FORTRAN Glossary:

NGAM	The number of gamma groups.
NGROUP	The number of gamma plus neutron groups.
NNEUT	The number of neutron groups.
STMAX	The maximum macroscopic total air cross section.
STMIN	The minimum macroscopic total air cross section.
XSECT	An array containing the neutron and gamma sea level air macroscopic cross sections.

```
      SUBROUTINE CROSS(STMAX,STMIN,NNEUT,NGAM,NGROUP)
C
C          THIS SUBROUTINE READS IN THE GAMMA AND
C          NEUTRON CROSS SECTIONS AND STORES THEM ON
C          TAPE 9.
C
C          NGROUP IS THE NUMBER OF GAMMA AND NEUTRON
C          GROUPS.
C
C          STMAX IS THE LARGEST TOTAL CROSS SECTION.
C
C          STMIN IS THE SMALLEST TOTAL CROSS SECTION.
C
C          M IS THE NUMBER OF CROSS SECTIONS IN ANY
C          ONE GROUP.
C
C          IN ANY GROUP G, THE CROSS SECTIONS ARE
C          ARRANGED IN THE FOLLOWING ORDER,
C              1 IS TRANSPORT
C              2 IS REMOVAL
C              3 IS TOTAL
C              4 IS SCATTER (G TO G)
C              5 IS SCATTER (G-1 TO G)
C              6 IS SCATTER (G-2 TO G)
C              7 IS SCATTER (G-3 TO G)
C              ETC.
C
C          NOTE.. IF OTHER THAN THE SUPPLIED DATA IS
C          USED, THE NUMBER OF GROUPS AND ENERGY BANDS
C          MUST AGREE WITH THAT SUPPLIED AS INPUT FOR
C          DOSE CALCULATIONS AND WEAPON SOURCE SPECTRUM.
C
      COMMON XSECT(43,40),I,J,M
      REWIND 9
      READ 4,NNEUT,NGAM
      NGROUP=NNEUT+NGAM
      M=NGROUP+3
      READ 1, ((XSECT(I,J),I=1,M),J=1,NGROUP)
      DO 2 J=1,NGROUP
 2    WRITE (9) (XSECT(I,J),I=1,M)
      STMAX=XSECT(3,1)
      STMIN=XSECT(3,1)
      DO 3 I=1,NGROUP
      STMAX=AMAX1(STMAX,XSECT(3,I))
 3    STMIN=AMIN1(STMIN,XSECT(3,I))
 1    FORMAT(1X,1P7E11.4)
 4    FORMAT(2I4)
      END
```

Subroutine DOSE

This subroutine reads in the response functions that are used to convert multigroup fluence to silicon dose in rads or tissue dose in rads. These response functions are then stored on a disk file. This subroutine should be included in the master nuclear effects program.

FORTRAN Glossary:

NGAM	The number of gamma groups.
NNEUT	The number of neutron groups.
SILGAM	An array containing the response functions to convert group gamma fluences to silicon dose.
SILNEUT	An array containing the response functions to convert group neutron fluences to silicon dose.
TISGAM	An array containing the response functions to convert group neutron fluences to tissue dose.
TISNEUT	An array containing the response functions to convert group neutron fluences to tissue dose.

```
      SUBROUTINE DOSE(NNEUT,NGAM)
C
C         THIS SUBROUTINE READS IN THE NEUTRON AND
C         GAMMA RESPONSE FUNCTIONS FOR CONVERTING
C         FLUENCE TO TISSUE DOSE OR SILICON DOSE IN
C         RADS.
C
C         NOTE.. IF OTHER THAN THE SUPPLIED DATA IS
C         USED, THE NUMBER OF GROUPS AND ENERGY BANDS
C         MUST AGREE WITH THAT SUPPLIED AS INPUT FOR
C         GROUP CROSS SECTIONS AND WEAPON SOURCE
C         SPECTRUM.
C
C         THE RESPONSE FUNCTIONS ARE STORED ON TAPE 9.
C
C         SILNEUT IS THE RESPONSE FUNCTION FOR CONVERTING
C         NEUTRON FLUENCE TO  SILICON DOSE IN RADS.
C         SILGAM IS THE RESPONSE FUNCTION FOR CONVERTING
C         GAMMA FLUENCE TO SILICON DOSE IN RADS.
C         TISNEUT IS THE RESPONSE FUNCTION FOR CONVERTING
C         NEUTRON FLUENCE TO TISSUE DOSE IN RADS.
C         TISGAM IS THE RESPONSE FUNCTION FOR CONVERTING
C         GAMMA FLUENCE TO TISSUE DOSE IN RADS.
C
      COMMON SILNEUT(22),SILGAM(18),TISNEUT(22),TISGAM(18),I
      READ 1,(SILNEUT(I),I=1,NNEUT)
      WRITE (9) (SILNEUT(I),I=1,NNEUT)
      READ 1, (SILGAM(I),I=1,NGAM)
      WRITE (9) (SILGAM(I),I=1,NGAM)
      READ 1, (TISNEUT(I),I=1,NNEUT)
      WRITE (9) (TISNEUT(I),I=1,NNEUT)
      READ 1, (TISGAM(I),I=1,NGAM)
      WRITE (9) (TISGAM(I),I=1,NGAM)
      REWIND 9
 1    FORMAT(1X,1P7E11.4)
      RETURN
      END
```

Subroutine AIRCR

This subroutine reads the aircraft data: positions, vulnerability, and vulnerability units. It also sets up the array that keeps track of whether the aircraft survives or is killed. This subroutine is a substitute for the data that would be obtained from the master nuclear effects program.

FORTRAN Glossary:

ACPOS	An array that stores the aircraft positions in X, Y, Z coordinates. Units are kilometers.
ARRAY	An array that specifies a live aircraft or one killed.
MODE	An array indicating the units of the figures given for aircraft vulnerability.
NUMBER	The starting number of aircraft.
VUL	An array that stores the aircraft neutron and gamma vulnerability.

```
      SUBROUTINE AIRCR(ARRAY,ACPOS,VUL,MODE,NUMBER)
      INTEGER ARRAY(100)
      DIMENSION ACPOS(3,100),VUL(2),MODE(2)
      READ 101,NUMBER
      DO 1 I=1,NUMBER
1     ARRAY(I)=1
      READ 100,((ACPOS(I,J),I=1,3),J=1,NUMBER)
      READ 101,(MODE(I),I=1,2)
      READ 100,(VUL(I),I=1,2)
100   FORMAT(1X,1P6E11.4)
101   FORMAT(10I4)
      END
```

Subroutine GAMNEUT

This subroutine is the heart of the code and is the subroutine designed for incorporation into the master nuclear effects program. This subroutine calculates the scattered neutron and gamma fluences. All the following subroutines are called by this subroutine and have functions of calculating the preliminary information needed for the fluence calculations or converting the data to a form suitable for data output.

Subroutine GAMNEUT Glossary:

A	A coefficient of the difference equation
ABAR	A coefficient of the difference equation
ACPOS	An array that stores the aircraft positions in X, Y, Z coordinates
B	A coefficient of the difference equation
BBAR	A coefficient of the difference equation
BURST	An array that stores the burst position in X, Y, Z coordinates
C	A coefficient of the difference equation
D	The diffusion coefficient
DELR	The radial mesh interval
DELRLOW	The value of DELR equivalent to the value of the x^1 interval in the highest row in x^3 using the nonorthogonal coordinate system
DELX1	The x^1 interval
DELX3	The x^3 interval
DIA	A packed matrix. This is one of the submatrices used in the block tri-diagonal algorithm

FLU	An array that stores the calculated group scattered fluences
G	One of the matrices used in the block tri-diagonal algorithm
GROUP	A counter. Its value is that of the group for which scattered fluence is being calculated
G1 G2 G3 G4	Coefficients of the difference equation
H	The scale height of the atmosphere
HOB	The height of burst
KCOL	An array used to unpack the LOWER matrix for the row connecting the two coordinate systems
LMAP	A number indicating the plot option
LOUT	A number indicating the printed output option
LOWER	A packed matrix. This is one of the submatrices used in the block tri-diagonal algorithm
MODE	An array indicating the units of the figures given for aircraft vulnerability
NGAM	The number of gamma groups
NGROUP	The number of gamma plus neutron groups
NHOR	The number of horizontal mesh points
NNEUT	The number of neutron groups
NUMBER	The starting number of aircraft
NUP	The number of vertical mesh points
POS	An array that stores the aircraft positions in r,z coordinates
Q	One of the submatrices used in the block tri-diagonal algorithm
S	One of the submatrices used in the block tri-diagonal algorithm

SIGR	The group air macroscopic removal cross section
SIGS	The group air macroscopic scatter cross section
SIGT	The group air macroscopic total cross section
SIGTR	The group air macroscopic transport cross section
SOURCE	The total number of neutrons or gammas output from the weapon for a group
SPECT	The number of neutrons or gammas per kiloton output from the weapon for a group
UPPER	A packed matrix. One of the submatrices used in the block tri-diagonal algorithm
X1	The value of the x^1 coordinate
X1MIN	The minimum value of the x^1 coordinate
X3	The value of the x^3 coordinate
X3HOB	The height of burst expressed in terms of x^3
X3MIN	The minimum value of x^3
X3SW	The value of x^3 when the coordinate systems are switched
YIELD	The weapon yield in kilotons

```
      SUBROUTINE GAMNEUT(NUMBEP,ACPOS,ARRAY,PURST,VUL,MODE,NNEUT,
     ASTMAX,STMIN,LMAP,LOUT,NGROUP,YIELD)
C
C         THIS SUBROUTINE DETERMINES THE MULTIGROUP
C         GAMMA AND NEUTRON FLUENCE
C
      REAL LOWER(60,3)
      INTEGER ARRAY(100),GROUP
      DIMENSION VUL(2),MODE(2),ACPOS(3,100),EURST(3),SPECT(40),FLU
     A0),POS(2,100)
      COMMON G(60,120),C(60,60),DIA(60,3),UPPER(60,3),KCOL(60),W(
     AS(60,120)
      EQUIVALENCE(S,FLU)
C
C         DETERMINE THE WEAPON OUTPUT SPECTRUM.
C
      CALL SELECT(SPECT,NNEUT,NGAM,NGROUP)
C
C         LOCATE HOB (IN UNITS OF CM.) AND SET UP
C         SCALE HEIGHT H AND PI.  IF HOB IS LESS THAN
C         7 KM. OR GREATER THAN 100 KM., NO CALCULATIONS
C         WILL BE MADE.
C
      HOB=BURST(3)*1.0E+05
      IF(HOB.LT.7.0E+05.OR.HOB.GE.100.0E+05)301,302
  301 PRINT 500
      PRINT 501
  500 FORMAT(////,4X,*HOB IS LESS THAN 7 KM. OR GREATER THAN 1.
  501 FORMAT(4X,*NO GAMMA OR NEUTRON CALCULATIONS WILL BE MADE*,//,
      GO TO 300
  302 H=7.0239E+05
      PI=3.1415927
      REWIND 11
C
C         DETERMINE THE MESH
C
      CALL MESH(H,STMIN,STMAX,HOB,X1MIN,DELX1,NHOR,L,X3MIN,DELX3,
     AX3HOB)
C
C         LOCATE THE AIRCRAFT IN THE MESH.  IF NO
C         AIRCRAFT ARE IN THE MESH AREA, ALL AIRCRAFT
C         ARE ASSUMED TO HAVE SURVIVED.  NO FURTHER
C         CALCULATIONS WILL BE MADE.
C
      CALL LOCATE(NUMBER,ACPOS,BURST,X1MIN,DELX1,NHOR,H,X3MIN,D
     AL,POS,NRET,ARRAY)
      IF(NRET.EQ.0)303,304
  303 PRINT 502
      PRINT 501
  502 FORMAT(////,4X,*ALL AIRCRAFT ARE OUTSIDE THE AREA OF GAMMAS
     AUTRONS*)
      GO TO 300
C
C         CALCULATE THE GROUP AND POSITION INDEPENDENT CONSTANTS.
C
```

```
 [illegible] M1=NHOR-1
      [illegible]
      T3=1./[illegible]
      T4=T2/2.
      T5=T4*T3
      T6=T2**2
      T7=2.*T6
      T8=2.*T7
      T9=T7**2
      T10=T9*2.
      T11=T3/2.
      T16=H**2
C
C          START THE CALCULATION OF THE GROUP FLUENCES.
C
      GROUP=0
 40   GROUP=GROUP+1
      REWIND 10
      IF(GROUP.GT.NGROUP)GO TO 50
C
C          CALCULATE THE GROUP VIRGIN FLUENCE AND
C          THE GROUP SOURCE (S) MATRIX.
C
C
      SOURCE=YIELD*SPECT(GROUP)
      CALL VIRGIN(SOURCE,PI,H,T16,X3HOB,X3MIN,X1MIN,NUP,NHOR,T,DELX1,
     ADELX3,D,SIGS,S,DELR,X3SW,SIGTR,SIGR,SIGT,GROUP,NGROUP)
C
C          SET UP THE SUBMATRICES DIA,UPPER, AND LOWER
C          IN PACKED FORM.
C
C          FIRST CALCULATE POSITION INDEPENDENT CONSTANTS
C
      T1=T16*SIGR/D
      T15=1./DELR
      T17=T16*T15/2.
      T18=T15**2
      T19=T18*T16
      T20=2.*T19
      T21=2.*T20
      T22=T1+T2[illegible]
      T12=1./(2.*D)
      T13=2.*T12*T3
      T14=T12*T2
 1    X3=X3MIN
      DO 40 J=1,NUP
C
C          CALCULATE THE HEIGHT DEPENDENT CONSTANTS
C
      Z1=EXP(X3)
      Z2=EXP(Z1)
      Z3=Z1**2
      Z4=Z2**2
      Z5=T5/Z1
      Z6=T7/Z4
```

```
      Z7=T10/Z3
      Z8=T6/Z4
      Z9=T4/Z4
      Z10=T8/Z4
      IF(J.EQ.NUP.AND.L.EQ.1)3,2
2     Z11=T9/Z3
      Z12=[illegible]
      Z13=T11/Z3
      GO TO 4
3     Z14=T13/(Z3*Z2)
      Z15=T12/(Z1*Z2)
      Z16=T12/(Z3*Z2)
      Z19=Z14+Z15-Z16
      Z17=T14/(Z1*Z2)
4     IF(X3.GE.X3SW)GO TO 100
      Z18=T1+Z6+Z7
      X1=X1MIN
C
C         CALCULATE DIA, LOWER, AND UPPER
C
      DO 14 I=1,NHOR
      X=X1/75
      C=Z18
      IF(I.EQ.1)GO TO 6
      G1=-X
      G3=X
      C=C+T7*(X1**2)
      BPAP=Z8+T6*(X1**2)-Z9/X1
      IF(I.EQ.NHOR)GO TO 5
      G2=X
      G4=-X
      B=Z8+T6*(X1**2)+Z9/X1
      GO TO 7
5     B=0
      G2=0.
      G4=0.
      GO TO 7
6     G1=0.
      G2=0.
      G3=0.
      G4=0.
      BPAR=0.
      B=Z10
      C=C+T6
7     IF(J.EQ.NUP.AND.L.EQ.1)GO TO 11
      IF(J.EQ.1)GO TO 8
      APAR=Z11-Z12+Z13
      IF(J.EQ.NUP)10,9
8     APAR=0
      G1=0.
      G2=0.
9     A=Z11+Z12-Z13
      GO TO 13
10    A=0.
      G3=0.
```

```
      G4=0.
      GO TO 13
11    A=0.
      G1=0.
      G2=0.
      G3=0.
      G4=0.
      ABAR=Z7
      C=C+Z18
      IF(I.EQ.1)GO TO 12
      BBAR=BBAR-(X1/Z17)
      IF(I.EQ.NHOR)GO TO 13
12    B=B+(
13    DIA(I,1)=BBAR
      DIA(I,2)=-C
      DIA(I,3)=B
      LOWER(I,1)=G1
      LOWER(I,2)=ABAR
      LOWER(I,3)=G2
      UPPER(I,1)=G3
      UPPER(I,2)=A
      UPPER(I,3)=G4
14    X1=X1+DELX1
      DFLRLOW=DELX1*H*Z2
      GO TO 120
100   IF(J.EQ.NUP.AND.L.EQ.1)GO TO 113
      IF(J.EQ.NUP)GO TO 111
      A=Z11+Z12-Z13
      IF(J.EQ.1)110,112
110   ABAR=0.
      GO TO 114
111   A=0.
112   ABAR=Z11-Z12+Z13
      GO TO 114
113   ABAR=Z7
      A=0.
114   Z18=Z7+T22
      X1=0
      DO 106 I=1,NHOR
      C=Z18
      IF(I.EQ.1)GO TO 102
      BBAR=T19-(T17/X1)
      IF(I.EQ.NHOR)GO TO 101
      B=T19+(T17/X1)
      GO TO 103
101   B=0.
      GO TO 103
102   C=C+T20
      BBAR=0.
      B=T21
103   IF(J.EQ.NUP.AND.L.EQ.1)104,105
104   C=C+Z19
105   DIA(I,1)=BBAR
      DIA(I,2)=-C
      DIA(I,3)=B
```

```
      IF(X3.NE.X3SW)GO TO 1?6
      KA=1
      DO 121 K=1,3
121   LOWER(I,K)=0.
      IF(I.NE.1)GO TO 122
      KCOL(I)=1
      LOWER(I,1)=ABAR
      GO TO 1?6
122   IF(KA.EQ.2.OR.KA.EQ.3)GO TO 106
      K=X1/DELRLCW
      KK=K+1
      IF(KK.EQ.NHOR)KA=2
      IF(KK.GT.NHOR)KA=3
      CHI=(X1-(K*DELRLCW))/DELRLOW
      LOWER(I,1)=ABAR-CHI*ABAR
      GO TO (123,124,124)KA
123   LOWER(I,2)=CHI*ABAR
124   KCOL(I)=KK
106   X1=X1+DELR
C
C          CALCULATE THE W SUBMATRIX
C
120   DO 15 I=1,NHOR
      DO 15 K=1,NHOR
15    W(K,I)=0.0
      IF(J.NE.1)GO TO 18
      W(1,1)=DIA(1,2)
      W(1,2)=DIA(1,3)
      W(NHOR,K1)=DIA(NHOR,1)
      W(NHOR,NHOR)=DIA(NHOR,2)
      K2=0
      DO 17 I=2,K1
      DO 16 K=1,3
16    W(I,K+K2)=DIA(I,K)
17    K2=K2+1
      GO TO 25
18    IF(X3-X3SW)99,130,140
99    DO 19 I=1,NHOR
      W(1,I)=-LOWER(1,2)*Q(1,I)
19    W(NHOR,I)=-LOWER(NHOR,1)*Q(K1,I    )-LOWER(NHOR,2)*Q(NHOR,I)
      K2=0
      DO 22 M=2,K1
      DO 21 I=1,NHOR
      SUM=0.
      DO 20 K=1,3
20    SUM=SUM+LOWER(M,K)*Q(K+K2,I)
21    W(M,I)=-SUM
22    K2=K2+1
      GO TO 131
130   KA=1
      DO 135 M=1,NHOR
      IF(KA.EQ.2.OR.KA.EQ.3)GO TO 180
      KK=KCOL(M)
      IF(KK.EQ.NHOR)KA=2
      IF(KK.GT.NHOR)KA=3
```

```
 18.  DO 135 I=1,NHOR
      SUM=0.
      GO TO (133,134,135)KA
 133  SUM=     LOWER(M,1)*Q(KK,I   )+LOWER(M,2)*Q(KK+1,I)
      GO TO 135
 134  SUM=     LOWER(M,1)*Q(KK,I)
 135  W(M,I)=-SUM
      GO TO 131
 140  DO 141 M=1,NHOR
      DO 141 I=1,NHOR
 141  W(M,I)=-ABAR*Q(M,I)
 131  W(1,1)=DIA(1,2)+W(1,1)
      W(1,2)=DIA(1,3)+W(1,2)
      W(NHOR,K1)=DIA(NHOR,1)+W(NHOR,K1)
      W(NHOR,NHOR)=DIA(NHOR,2)+W(NHOR,NHOR)
      K2=0
      DO 24 I=2,K1
      DO 23 K=1,3
 23   W(I,K+K2)=DIA(I,K)+W(I,K+K2)
 24   K2=K2+1
C
C          CALCULATE W INVERSE
C
 25   CALL MATRIX(10,NHOR,NHOR,C,W,60,ZZ)
C
C          CALCULATE THE Q SUBMATRIX
C
      IF(J.EQ.NUP)GO TO 31
      IF(X3.GE.X3SW)GO TO 142
      DO 26 M=1,NHOR
      Q(M,1)    =W(M,1)*UPPER(1,2)+W(M,2)*UPPER(2,1)
 26   Q(M,NHOR)    =W(M,K1)*UPPER(K1,3)+W(M,NHOR)*UPPER(NHOR,2)
      DO 29 M=1,NHOR
      K2=0
      DO 28 I=2,K1
      SUM=0.
      DO 27 K=1,3
 27   SUM=SUM+W(M,K+K2)*UPPER(K+K2,4-K)
      Q(M,I)    =SUM
 28   K2=K2+1
 29   CONTINUE
      GO TO 144
 142  DO 143 I=1,NHOR
      DO 143 K=1,NHOR
 143  Q(I,K)    =A*W(I,K)
C
C          STORE THE Q SUBMATRIX ON DISK.
C
 144  WRITE (10)   ((Q(I,K   ),I=1,NHOR),K=1,NHOR)
C
C          CALCULATE THE G SUBMATRIX
C
 31   IF(J.NE.1)GO TO 34
      DO 33 I=1,NHOR
      SUM=0.
```

```
      DO 32 K=1,NHOR
32    SUM=SUM+W(I,K)*S(K,J)
33    G(I,J)=SUM
      GO TO 40
34    JM=J-1
      IF(X3-X3SW)150,151,156
150   G(1,J)=LOWER(1,2)*G(1,JM)
      G(NHOR,J)=LOWER(NHOR,1)*G(K1,JM)+LOWER(NHOR,2)*G(NHOR,JM)
      K2=0
      DO 36 M=2,K1
      SUM=0.
      DO 35 K=1,3
35    SUM=SUM+LOWER(M,K)*G(K+K2,JM)
      G(M,J)=SUM
36    K2=K2+1
      GO TO 155
151   KA=1
      DO 154 M=1,NHOR
      IF(KA.EQ.2.OR.KA.EQ.3)GO TO 181
      KK=KCOL(M)
      IF(KK.EQ.NHOR)KA=2
      IF(KK.GT.NHOR)KA=3
181   SUM=0.
      GO TO (152,153,154)KA
152   SUM=LOWER(M,1)*G(KK,JM)+LOWER(M,2)*G(KK+1,JM)
      GO TO 154
153   SUM=LOWER(M,1)*G(KK,JM)
154   G(M,J)=SUM
      GO TO 155
156   DO 157 M=1,NHOR
157   G(M,J)=ABAR*G(M,JM)
155   DO 37 I=1,NHOR
37    DIA(I,1)=S(I,J)-G(I,J)
      DO 39 I=1,NHOR
      SUM=0.
      DO 38 K=1,NHOR
38    SUM=SUM+W(I,K)*DIA(K,1)
39    G(I,J)=SUM
40    X3=X3+DELX3
C
C          CALCULATE FLUENCE AT EACH POINT IN MESH
C
      JM=NUP+1
      DO 48 J=1,NUP
      JJ=JM-J
      IF(JJ.NE.NUP)GO TO 42
      DO 41 I=1,NHOR
41    FLU(I,JJ)=G(I,JJ)
      GO TO 46
42    DO 44 I=1,NHOR
      SUM=0.
      DO 43 K=1,NHOR
43    SUM=SUM+D(I,K    )*FLU(K,JJ+1)
44    FLU(I,JJ)=SUM
      DO 45 I=1,NHOR
```

```
 45   FLU(I,JJ)=G(I,JJ)-FLU(I,JJ)
C
C          READ A G SUBMATRIX FROM DISK.
C
 46   IF(J.EQ.1)GO TO 48
      BACKSPACE 10
      IF(J.EQ.2)GO TO 47
      BACKSPACE 10
 47   READ (10)   ((G(I,K  ),I=1,NHOR),K=1,NHOR)
 48   CONTINUE
C
C          COMPUTE THE TOTAL GROUP FLUENCE.
C
      CALL ADD(FLU,NUP,NHOR)
C
C          PRINT THE DETAILED GROUP OUTPUT IF DESIRED.
C
      IF(LOUT.EQ.0)GO TO 49
      CALL OUT(NUP,NHOR,X3MIN,X1MIN,DELX3,DELX1,H,T,DELR,X3SW,SIGT,GROUP
     A)
 49   GO TO 400
C
C          CONVERT THE MULTIGROUP FLUENCE TO A TOTAL
C          FLUENCE OR DOSE DEPENDING UPON MODE SPECIFIED.
C
 50   CALL CONVERT(NHOR,NUP,MODE,NGROUP,NNELT,NGAM)
C
C          DETERMINE IF AIRCRAFT SURVIVE.
C
      CALL CHECK(NHOR,NUP,MODE,VUL,POS,NUMBER,X3MIN,DELX3,X3SW,DELR,
     AX1MIN,DELX1,H,ARRAY,L,BURST,ACPOS)
C
C          DRAW PLOTS IF DESIRED.
C
      IF(LMAP.EQ.0)GO TO 51
      CALL MAP(LMAP,NUP,NHOR,POS,MODE,VUL,HOB,DELR,X1MIN,DELX1,X3SW,
     AX3MIN,DELX3,L,ARRAY,NUMBER,H)
 51   REWIND  9
      REWIND 10
      REWIND 11
 300  RETURN
      END
```

Subroutine SELECT

This subroutine determines what weapon output spectrum will be used. The user specifies the type of weapon and the source of the spectrum. He can load his own spectrum or use one of the unclassified default spectra stored in the labeled common SPECTRA. This subroutine is called by GAMNEUT

Subroutine SELECT Glossary:

GAMSO NEUTSO	A number that determines if the default spectra or a user supplied spectra is to be used
NGROUP	The number of neutron plus gamma groups
NNEUT	The number of neutron groups
SPECFG	The default fission gamma spectrum
SPECFN	The default fission neutron spectrum
SPECTNG	The default thermonuclear gamma spectrum
SPECTNN	The default thermonuclear neutron spectrum
TYPE	The weapon type

```
      SUBROUTINE SELECT(SPECT,NNEUT,NGAM,NGROUP)
C
C         THIS SUBROUTINE READS IN THE WEAPON TYPE,
C         SOURCE OF WEAPON SPECTRUM, AND SPECTRUM
C         (GAMMA AND NEUTRON) IF SUPPLIED BY USER.
C
C         WEAPON TYPE (TYPE)
C                - A ONE IS A FISSION WEAPON
C                - A TWO IS A THERMONUCLEAR WEAPON
C
C         SOURCE OF WEAPON (GAMSO FOR GAMMAS,
C         NEUTSO FOR NEUTRONS)
C                - A BLANK (ZERO) INDICATES THE DEFAULT
C                  SPECTRUM WILL BE USED
C                - A ONE INDICATES A USER SUPPLIED
C                  SPECTRUM WILL BE USED
C
C        .NOTE.. IF THE DEFAULT SPECTRUM IS NOT USED,
C         THE NUMBER OF GROUPS AND ENERGY BANDS MUST
C         AGREE WITH THAT SUPPLIED AS INPUT FOR
C         GROUP CROSS SECTIONS AND DOSE CALCULATIONS.
C
      DIMENSION SPECT(40)
      INTEGER GAMSO, TYPE
      COMMON GAMSO,TYPE,NEUTSO,I,M
      COMMON/SPECTRA/SPECTNN(22),SPECFN(22),SPECFG(18),SPECTNG(18)
      READ 1,TYPE,NEUTSO,GAMSO
C
C         DETERMINE SOURCE OF NEUTRON SPECTRUM.
C
      IF(NEUTSO.EQ.1)GO TO 6
C
C         DETERMINE WEAPON TYPE.
C
      IF(TYPE.EQ.2)GO TO 4
C
C         LOAD DEFAULT NEUTRON FISSION SPECTRUM.
C
      DO 3 I=1,22
    3 SPECT(I)=SPECFN(I)
      GO TO 7
C
C         LOAD DEFAULT NEUTRON TN SPECTRUM.
C
    4 DO 5 I=1,22
    5 SPECT(I)=SPECTNN(I)
      GO TO 7
C
C         LOAD USER SUPPLIED NEUTRON SPECTRUM
C
    6 READ 100,(SPECT(I),I=1,NNEUT)
C
C         DETERMINE SOURCE OF GAMMA SPECTRUM.
C
    7 IF (GAMSO.EQ.1)GO TO 11
```

```
C
C          DETERMINE WEAPON TYPE.
C
      IF(TYPE.EQ.2)GO TO 20
C
C          LOAD DEFAULT GAMMA FISSION SPECTRUM.
C
      DO 21 I=1,18
 21   SPECT(22+I)=SPECFG(I)
      GO TO 12
C
C          LOAD DEFAULT GAMMA TN SPECTRUM.
C
 20   DO 8 I=1,18
 8    SPECT(22+I)=SPECTNG(I)
      GO TO 12
C
C        . LOAD USER SUPPLIED GAMMA SPECTRUM,
C
 11   M=NNEUT+1
      READ 100,(SPECT(I),I=M,NGROUP)
 1    FORMAT(10I4)
 100  FORMAT(1X,1P7E11.4)
 12   RETURN
      END
```

Subroutine MESH

This subroutine calculates the number of vertical and horizontal mesh points and the spacing between mesh points. The maximum number of horizontal mesh points used is 40; however, the number can be changed. The card

```
IF(NHOR.GT.40)NHOR=40
```

limits the number to 40. If any other number of mesh points is desired, the user can change the two 40's to that number. This number can not be greater than 60.

The maximum number of vertical mesh points is 70, but this number can also be changed. The card

```
IF(M.GT.70)M=70
```

limits the number to 70. If any other number is desired, the user can change the two 70's to that number. This number can not be greater than 120.

If the number of mesh points are increased, the running time will also increase. At present, this option has not been exercised, so no estimate of the increased running time can be made.

Subroutine MESH Glossary:

X1MAX	The maximum value of x^1
X3MAX	The maximum value of x^3
X3TOTAL	The total length of x^3

See the GAMNEUT glossary for the remaining terms.

```
      SUBROUTINE MESH(H,STMIN,STMAX,HOB,X1MIN,DELX1,NHOR,L,X3MIN,DELX3,N
     AUP,X3HOB)
      COMMON Z,I,J,T1,K,M,R,N,X3TOTAL,X1MAX,X3MAX,TT,T
C
C         THIS SUBROUTINE CALCULATES THE NUMBER OF
C         VERTICAL AND HORIZONTAL MESH POINTS AND THE
C         SPACING BETWEEN MESH POINTS.  IT ALSO
C         DETERMINES THE MINIMUM AND MAXIMUM FOR EACH
C         COORDINATE.  CHECKS ARE MADE TO PREVENT THE
C         TOP OF THE MESH BEING GREATER THAN 100 KM
C         AND THE BOTTOM OF THE MESH BEING LESS
C         THEN  OR EQUAL TO 0 KM.  IN ADDITION A MESH
C         POINT IS NOT ALLOWED TO BE ON THE HOB.
C
C         X1MIN IS THE MINIMUM X1 COORDINATE.
C
C         X1MAX IS THE MAXIMUM X1 COORDINATE.
C
C         DELX1 IS THE X1 INTERVAL.
C
C         NHOR IS THE NUMBER OF MESH POINTS IN THE
C         X1 DIRECTION.
C
C         L IS ZERO UNLESS THE TOP OF THE MESH IS
C         100 KM. IF THE TOP OF THE MESH IS 100 KM
C         L IS SET TO 1.
C
C         X3MIN IS THE M INIMUM X3 COORDINATE.
C
C         X3MAX IS THE MAXIMUM X3 COORDINATE.
C
C         DELX3 IS THE X3 INTERVAL.
C
C         NUP IS THE NUMBER OF MESH POINTS IN THE
C         X3 DIRECTION.
C
C
C
C         X1MIN,X1MAX,NHOR, AND DELX1 ARE CALCULATED
C         FIRST.  X1MAX IS EQUIVALENT TO 10 MEAN
C         FREE PATHS OF THE GROUP WITH THE LONGEST
C         MEAN FREE PATH.
C
      T=H*STMIN
      TT=H*STMAX
      X1MIN=0.
      X1MAX=10./T
      NHOR=3*TT*X1MAX
      IF(NHOR.GT.40)NHOR=40
      DELX1=X1MAX/NHOR
C
C         NEXT X3MIN IS LOCATED 10 MEAN FREE PATHS
C         DOWN FROM HOB.  HOWEVER, Z IS NOT ALLOWED
C         TO BE EQUAL TO OR LESS THEN 0 KM.
C
      Z=HOB
```

```
      DO 1 I=1,10
      J=I
      T1=H*ALOG(1.+(EXP(Z/H)/T))
      Z=Z-T1
      IF(Z.GT.0)GO TO 1
      Z=Z+T1
      GO TO 2
 1    CONTINUE
 2    X3MIN=ALOG(Z/H)
C
C          NEXT, X3MAX IS LOCATED 10 MEAN FREE PATHS
C          UP FROM HOB.  HOWEVER, Z IS NOT ALLOWED
C          TO BE GREATER THEN 100 KM.  IN ADDITION,
C          A CHECK IS MADE FOR AN INFINITE MEAN FREE
C          PATH IN WHICH CASE Z IS SET TO 100 KM.
C          IF Z IS SET TO 100 KM, L IS SET TO 1.
C
      L=0
      Z=HOB
      DO 5 I=1,10
      K=I
      T1=EXP(Z/H)/T
      IF(T1.LT.1.)GO TO 4
 3    L=1
      Z=1.0E+07
      GO TO 6
 4    Z=Z-H*ALOG(1.-T1)
      IF(Z.GE.1.0E+07)GO TO 3
 5    CONTINUE
 6    X3MAX=ALOG(Z/H)
C
C          NEXT, DELX3 AND NUP ARE CALCULATED.  A
C          CHECK IS MADE TO INSURE THAT THE HOB IS
C          NOT ON A MESH POINT.
C
      J=J+K
      M=3.0*J*ALOG(1.+(EXP(HOB/H)/T))/ALOG(1.+(EXP(HOB/H)/TT))
      IF(M.GT. 70)M=70
      X3TOTAL=X3MAX-X3MIN
      X3HOB=ALOG(HOB/H)
      T1=X3HOB-X3MIN
 7    DELX3=X3TOTAL/M
      B=T1/DELX3
      N=B
      IF((B-N).NE.0)GO TO 8
      M=M-1
      GO TO 7
 8    NUP=M
      X3MIN=X3MIN+DELX3
      IF(L.EQ.1)GO TO 9
      NUP=NUP-1
 9    RETURN
      END
```

Subroutine LOCATE

This subroutine checks the aircraft positions against the mesh area set up in subroutine MESH. If all the aircraft are outside the meshed area, NRET will remain zero and cause subroutine GAMNEUT to terminate fluence calculations. If at least one aircraft is inside the meshed area, NRET is set equal to one and fluence calculations will continue.

Subroutine LOCATE Glossary:

A	The scale height of the atmosphere in kilometers
ACPOS	An array that stores aircraft positions in X, Y, Z coordinates
ARRAY	An array that specifies if the aircraft is still live
BURST	The coordinates of the weapon burst
DELX1	The x^1 interval
DELX3	The x^3 interval
H	The scale height of the atmosphere in centimeters
NHOR	The number of horizontal mesh points
NRET	A number that indicates if any aircraft are in the meshed area
NUMBER	The number of starting aircraft
NUP	The number of vertical mesh points
POS	An array that stores the aircraft positions in r,z coordinates
R	The radial distance
X1	The value of x^1
X1MIN	The minimum value of x^1

X3MIN	The minimum value of x^3
ZMAX	The altitude of the upper boundary of the meshed area
ZMIN	The altitude of the lower boundary of the meshed area

```
      SUBROUTINE LOCATE(NUMBER,ACPOS,BURST,X1MIN,DELX1,NHOR,H,X3MIN,DELX
     A3,NUP,L,POS,NRET,ARRAY)
C
C          THIS SUBROUTINE LOCATES THE AIRCRAFT IN
C          A R,Z GEOMETRY WITH RESPECT TO THE BURST.
C          THE AIRCRAFT POSITIONS ARE CHECKED AGAINST THE
C          OUTER BOUNDARIES OF THE MESH TO SEE IF ANY
C          AIRCRAFT ARE WITHIN THE MESH.  ONCE AN
C          AIRCRAFT IS LOCATED INSIDE THE MESHED AREA
C          THIS SUBROUTINE TERMINATES AND CONTROL
C          RETURNS TO GAMNEUT.  IF ALL THE AIRCRAFT
C          ARE OUTSIDE THE MESHED AREA, NRET REMAINS
C          ZERO.
C
      DIMENSION ACPOS(3,100),BURST(3),POS(2,100)
      INTEGER ARRAY(100)
      COMMON A,B,X1,ZMIN,ZMAX,R,I
      A=H*1.0E-05
      NRET=0
      DO 10 I=1,NUMBER
      IF(ARRAY(I).EQ.0)GO TO 10
      POS(2,I)=ACPOS(3,I)
      POS(1,I)=SQRT(((BURST(1)-ACPOS(1,I))**2)+((BURST(2)-ACPOS(2,I))**2
     A))
   10 CONTINUE
      ZMIN=A*EXP(X3MIN-DELX3)
      B=-DELX3
      IF(L.NE.1)B=0.
      ZMAX=A*EXP(X3MIN+(NUP*DELX3)+B)
      X1=NHOR*DELX1*A
      DO 11 I=1,NUMBER
      IF(ARRAY(I).EQ.0)GO TO 11
      IF(POS(2,I).GE.ZMAX.OR.POS(2,I).LE.ZMIN)12,13
   13 NRET=1
      GO TO 20
   12 R=X1*EXP(  POS(2,I)/A)
      IF(POS(1,I).GE.R.OR.POS(1,I).GE.400.)11,13
   11 CONTINUE
   20 RETURN
      END
```

Subroutine VIRGIN

This subroutine has three functions. First, the group virgin fluence is calculated at each point in the mesh. Second, a check is made to determine if the horizontal radius exceeds 200 kilometers. If the radius is greater than 200 km at altitude x^3, the value of x^3 is stored in X3SW, and the coordinate system is switched from the nonorthogonal system to the orthogonal system. The third function is the calculation of the S matrix for the matrix equation AF = S. The virgin fluence is stored on a disk file for later use.

Subroutine VIRGIN Glossary:

D	The diffusion coefficient
DELR	The actual radial spacing
DELX1	The x^1 interval
DELX3	The x^3 interval
DELZ	The distance between two vertical mesh points
DELZSQ	DELZ squared
F	Group total fluence
GROUP	A counter relating for what group the virgin fluence is being calculated
RHOSQ	The distance between the burst and a point, squared
VIRG	The group virgin fluence
XSECT	The group cross sections

```
      SUBROUTINE VIRGIN(SOURCE,PI,H,T0,X3HOE,X3MIN,X1MIN,NUP,NHOR,T,DELX
     A1,DFLX3,D,SIGS,S,DELR,X3SW,SIGTR,SIGR,SIGT,GROUP,NGROUP)
      DIMENSION S(60,120),F(60,120)
      COMMON T1,T2,T3,X3,J,T4,T5,T6,DELZ,DELZSQ,T9,X1,I,RHOSQ,VIRG(60,12
     A0),RA,K,T8,XSECT(43),L,LL,M
      INTEGER GROUP
      EQUIVALENCE (VIRG,F)
C
C         THIS SUBROUTINE CALCULATES THE VIRGIN
C         FLUENCE AT EACH MESH POINT AND STORES THE
C         RESULTING MATRIX ON TAPE.  THE S MATRIX
C         FOR THE MATRIX EQUATION AF=S IS CALCULATED
C         AND RETURNED TO THE MAIN PROGRAM.
C
      M=NGROUP+3
      READ (9) (XSECT(I),I=1,M)
      SIGTR=XSECT(1)
      SIGR=XSECT(2)
      SIGT=XSECT(3)
      SIGS=XSECT(4)
      D=1./(3.*SIGTR)
      T=H*SIGT
      IF(SOURCE.EQ.0.0)16,17
 16   DO 18 J=1,NUP
      DO 18 I=1,NHOR
 18   VIRG(I,J)=0.
      GO TO 19
 17   K=1
      T8=SOURCE/(4.*PI)
      T1=T8/T0
      T2=EXP(X3HOR)
      T3=1./EXP(T2)
      X3=X3MIN
      DO 4 J=1,NUP
      T4=EXP(X3)
      T5=EXP(T4)
      T6=1./T5
      T9=T6-T3
      DELZ=T4-T2
      IF(GROUP.EQ.1)7,8
 8    IF(X3.GE.X3SW)2,9
 7    IF(K.EQ.2)GO TO 2
 9    DELZSQ=DELZ**2
      X1=X1MIN
      DO 1 I=1,NHOR
      RHOSQ=DELZSQ+((X1*T5)**2)
      VIRG(I,J)=(T1/RHOSQ)*EXP((T*SQRT(RHOSQ)/DELZ)*T9)
 1    X1=X1+DELX1
      IF(GROUP.NE.1)GO TO 4
      X3SW=1.0E15
      RA=H*X1*T5
      IF(RA.LT.2.0E+07)GO TO 4
      DELR=RA/NHOR
      K=2
      X3SW=X3
```

```
      GO TO 4
2     DELZ=H*DELZ
      DELZSQ=DELZ**2
      RA=0.
      DO 3 I=1,NHOR
      RHOSQ=DELZSQ+(RA**2)
      VIRG(I,J)=(T3/RHOSQ)*EXP((T*SQRT(RHOSQ)/DELZ)*T9)
3     RA=RA+DELR
4     X3=X3+DELX3
19    T1=(-T0*STGS)/D
      WRITE (11) ((VIRG(I,J),I=1,NHOR),J=1,NUP)
      REWIND 11
      DO 5 J=1,NUP
      DO 5 I=1,NHOR
5     S(I,J)=T1*VIRG(I,J)
      IF(GROUP.EQ.1)GO TO 2.
      K=GROUP+4
      J=GROUP-1
      DO 10 I=1,J
      K=K-1
      IF(XSECT(K).EQ.0)11,12
11    READ (11)
      GO TO 10
12    T1=(-T0*XSECT(K))/D
      READ (11) ((F(L,LL),L=1,NHOR),LL=1,NUP)
      DO 13 LL=1,NUP
      DO 13 L=1,NHOR
13    S(L,LL)=S(L,LL)+T1*F(L,LL)
10    CONTINUE
20    RETURN
      END
```

Subroutine ADD

The function of this subroutine is to add the group virgin fluence to the group scattered fluence to produce the group total fluence. The group virgin fluence is read from a disk file and the group total fluence is written on the same record of this disk file.

Subroutine ADD Glossary:

FLU	At first, the group scattered fluence and later, the group total fluence
NHOR	The number of horizontal mesh points
NUP	The number of vertical mesh points
VIRG	The group virgin fluence

```
      SUBROUTINE ADD(FLU,NUP,NHCR)
C
C          THIS SUBROUTINE ADDS THE VIRGIN AND
C          SCATTERED FLUENCE FOR EACH GROUP TO GIVE
C          THE TOTAL GROUP FLUENCE.
C
      DIMENSION FLU(60,120)
      COMMON I,J,VIRG(60,120)
      READ (11) ((VIRG(I,J),I=1,NHCR),J=1,NUP)
      DO 1 J=1,NUP
      DO 1 I=1,NHCR
 1    FLU(I,J)=FLU(I,J)+VIRG(I,J)
      BACKSPACE 11
      WRITE (11) ((FLU(I,J),I=1,NHCR),J=1,NUP)
      END
```

Subroutine OUT

This subroutine prints a detailed output. A fluence for every group is printed for every mesh point. This subroutine is called from GAMNEUT only if LOUT is equal to one. Since the output will be several hundred pages, the use of this subroutine is not recommended unless the user wants to examine the group fluences.

Subroutine OUT Glossary:

A	The scale height of the atmosphere in km.
DELR	The actual radial mesh interval
DELX1	The x^1 mesh interval
DELX3	The x^3 mesh interval
FLU	The group total fluence
GROUP	The group number
R	The actual radial distance
X3SW	The value of x^3 when the change in coordinate systems occur
Z	The altitude

```
      SUBROUTINE OUT(NUP,NHOR,X3MIN,X1MIN,DELX3,DELX1,H,DELR,X3SW,GROU
      COMMON FLU(60,120),A,X3,I,J,B,Z,X1,R(80),C,K,M,JJ,G,RA
      INTEGER GROUP
C
C        THIS SUBROUTINE PRINTS THE DETAILED OUTPUT.
C        A FLUENCE IS PRINTED FOR EVERY MESH POINT
C        IN EVERY GROUP.
C
      A=H*1.0E-05
      X3=X3MIN
      BACKSPACE 11
      READ (11) ((FLU (I,J),I=1,NHOR),J=1,NUP)
      PRINT 500,GROUP
      DO 6 J=1,NUP
      B=EXP(X3)
      G=EXP(B)
      Z=A*B
      IF(X3.GE.X3SW)GO TO 2
      X1=X1MIN
      C=A*G
      DO 1 I=1,NHOR
      R(I)=C*X1
    1 X1=X1+DELX1
      GO TO 4
    2 RA=0.
      DO 3 I=1,NHOR
      R(I)=RA*1.0E-05
    3 RA=RA+DELR
    4 K=1
      M=8
      DO 5 I=1,30
      PRINT 501,Z
      PRINT 503,(R(JJ),JJ=K,M)
      PRINT 505,(FLU(JJ,J),JJ=K,M)
      K=K+8
      IF(K.GT.NHOR)GO TO 6
      M=M+8
      IF(M.GT.NHOR)M=NHOR
    5 CONTINUE
    6 X3=X3+DELX3
  500 FORMAT(///,3X,*THE GROUP IS *,I4,//)
  501 FORMAT(//,3X,*HEIGHT IS *,F10.5,* KM.*)
  503 FORMAT(3X,*RADIUS (KM)*,4X,8F14.3)
  505 FORMAT(3X,*GROUP   FLUENCE*,1P8E14.4)
      END
```

Subroutine CONVERT

This subroutine converts the multigroup fluences to the units specified by the MODE parameter. At the present time MODE can have a value from one to eight. The meanings of these values are described in the subroutine listing.

Subroutine CONVERT Glossary:

F	The group total fluence
FLU	The result of the conversion, either total fluence or dose
MODE	The units of FLU
SILGAM	The multigroup gamma response functions for silicon dose in rads
SILNEUT	The multigroup neutron response functions for silicon dose in rads
TISGAM	The multigroup gamma response functions for tissue dose in rads
TISNEUT	The multigroup neutron response functions for tissue dose in rads

```
      SUBROUTINE CONVERT(NHOR,NUP,MODE,NGRCLF,NNEUT,NGAM)
C
C          THIS SUBROUTINE CONVERTS THE MULTIGROUP
C          FLUENCE TO THE MODE SPECIFIED IN THE
C          VULNERABILITY.
C
C          MODE=1   TOTAL NEUTRON FLUENCE (N/CM2)
C          MODE=2   TOTAL GAMMA FLUENCE (G/CM2)
C          MODE=3   NEUTRON TISSUE DOSE (RADS)
C          MODE=4   GAMMA TISSUE DOSE (RADS)
C          MODE=5   NEUTRON + GAMMA TISSUE DOSE (RADS)
C          MODE=6   NEUTRON SILICON DOSE (RADS)
C          MODE=7   GAMMA SILICON DOSE (RADS)
C          MODE=8   NEUTRON + GAMMA TISSUE DOSE (RADS)
C
      DIMENSION MODE(2)
      COMMON SILNEUT(22),SILGAM(18),TISNEUT(22),TISGAM(18),K,F(60,120
     AGROUP,I,J,FLU(60,120),LIMIT
      INTEGER GROUP
      REWIND 10
      REWIND 11
      READ (9) (SILNEUT(I),I=1,NNEUT)
      READ (9) (SILGAM(I),I=1,NGAM)
      READ (9) (TISNEUT(I),I=1,NNEUT)
      READ (9) (TISGAM(I),I=1,NGAM)
      K=MODE(1)
 3    DO 4 J=1,NUP
      DO 4 I=1,NHOR
 4    FLU(I,J)=0.0
 10   GO TO (1,2,8,12,8,15,19,15)K
 1    GROUP=1
      LIMIT=NNEUT
      GO TO 5
 2    GROUP=1
      LIMIT=NGAM
 5    IF(GROUP.GT.LIMIT)GO TO 7
      READ (11) ((F(I,J),I=1,NHOR),J=1,NUP)
      DO 6 J=1,NUP
      DO 6 I=1,NHOR
 6    FLU(I,J)=FLU(I,J)+F(I,J)
      GROUP=GROUP+1
      GO TO 5
 7    WRITE (10)((FLU(I,J),I=1,NHOR),J=1,NUP)
      GO TO 50
 8    GROUP=1
      LIMIT=NNEUT
 9    IF(GROUP.GT.LIMIT)GO TO 11
      READ (11) ((F(I,J),I=1,NHOR),J=1,NUP)
      DO 30 J=1,NUP
      DO 30 I=1,NHOR
 30   FLU(I,J)=TISNEUT(GROUP)*F(I,J)+FLU(I,J)
      GROUP=GROUP+1
      GO TO 9
 11   IF(K.EQ.3)GO TO 7
 12   GROUP=1
```

```
      LIMIT=NGAM
14    IF(GROUP.GT.LIMIT)GO TO 7
      READ (11) ((F(I,J),I=1,NHOR),J=1,NUP)
      DO 13 J=1,NUP
      DO 13 I=1,NHOR
13    FLU(I,J)=TISGAM(GROUP)*F(I,J)+FLU(I,J)
      GROUP =GROUP+1
      GO TO 14
15    GROUP=1
      LIMIT=NNEUT
16    IF(GROUP.GT.LIMIT)GO TO 18
      READ (11) ((F(I,J),I=1,NHOR),J=1,NUP)
      DO 17 J=1,NUP
      DO 17 I=1,NHOR
17    FLU(I,J)=SILNEUT(GROUP)*F(I,J)+FLU(I,J)
      GROUP=GROUP+1
      GO TO 16
18    IF(K.EQ.6)GO TO 7
19    GROUP=1
      LIMIT=NGAM
20    IF(GROUP.GT.LIMIT)GO TO 7
      READ (11) ((F(I,J),I=1,NHOR),J=1,NUP)
      DO 21 J=1,NUP
      DO 21 I=1,NHOR
21    FLU(I,J)=SILGAM(GROUP)*F(I,J)+FLU(I,J)
      GROUP=GROUP+1
      GO TO 20
50    GO TO(51,60,51,60,60,51,60,60)K
51    K=MODE(2)
      GO TO 3
60    REWIND 10
      REWIND 11
      RETURN
      END
```

Subroutine CHECK

This subroutine places the aircraft in the meshed area and calculates the neutron and gamma fluence of dose the aircraft experienced. This calculation is compared to the aircraft vulnerability to determine if the aircraft survived. Subroutine NOTICE is then called to give the printed output concerning the aircraft.

Subroutine CHECK Glossary:

ACPOS	The aircraft positions in X, Y, Z coordinates
BURST	The burst coordinates
GLEVEL	The gamma level (fluence or dose) experienced by the aircraft
NLEVEL	The neutron level (fluence or dose) experienced by the aircraft
VUL	The aircraft gamma and neutron vulnerability

```
      SUBROUTINE CHECK(NHCR,NUP,MODE,VUL,FCS,NUMBER,X3MIN,DELX3,X3SW,
     ADELR,X1MIN,DELX1,H,ARRAY,L,BURST,ACPCS)
C
C          THIS SUBROUTINE LOCATES THE AIRCRAFT AND
C          DETERMINES IF GAMMAS OR NEUTRONS HAVE
C          KILLED THE AIRCRAFT
C
      REAL NLEVEL
      INTEGER ARRAY(100)
      DIMENSION MODE(2),VUL(2),POS(2,100),BURST(3),ACPOS(3,100)
      COMMON IL(5)      ,FG(60,120),K,ZMIN,ZMAX,N,ZA,S,NL,NH,B,DEL,Z1,Z2,
     AX1MAX,RMAX,R1MAX,R2MAX,NP,D,D1,NR1,FN1,FN2,FG1,FG2,NLEVEL,GLEVEL,
     BW,W1,W3,W4,FN(60,120)
      DEL=DELR*1.0E-05
      S=H*1.0E-05
      X1MAX=NHOR*DELX1
      RMAX=NHOR*DEL
      K=2
      IF(MODE(1).EQ.5.OR.MODE(1).EQ.8)K=1
      READ (10)  ((FN(I,J),I=1,NHOR),J=1,NUP)
      IF(K.EQ.1)GO TO 1
      READ (12)  ((FG(I,J),I=1,NHOR),J=1,NUP)
 1    ZMIN=X3MIN-DELX3
      ZMAX=(NUP*DELX3)+ZMIN
      IF(L.EQ.1)ZMAX=ZMAX+DELX3
      N=1
 2    IF(N.GT.NUMBER)GO TO 10
      IF(ARRAY(N).EQ.0)GO TO 9
      A=POS(2,N)
      ZA=ALOG(A/S)
      IF(ZA.GE.ZMAX.OR.ZA.LE.ZMIN)GO TO 8
      NL=(ZA-ZMIN)/DELX3
      NH=NL+1
      B=POS(1,N)
      Z1=ZMIN+(NL*DELX3)
      Z2=Z1+DELX3
      IF(Z1.GE.X3SW)GO TO 3
      R1MAX=S*X1MAX*EXP(EXP(Z1))
      IF(Z2.GE.X3SW)GO TO 4
      R2MAX=S*X1MAX*EXP(EXP(Z2))
      GO TO 5
 3    R1MAX=RMAX
 4    R2MAX=RMAX
 5    IF(R1MAX.LE.B)FN1=0.
      IF(R2MAX.LE.B)GO TO 8
      IF(FN1.EQ.0.)GO TO 6
      D=R1MAX/NHOR
      NR=(B/D)+1
      IF(NL.EQ.0)FN1=0.
      IF(FN1.EQ.0)GO TO 6
      IF(NR.EQ.NHOR)W1=0.
      IF(W1.EQ.0.)GO TO 20
      W1=FN(NR+1,NL)
 20   W=FN(NR,NL)
      FN1=W+((B-(NR*D))*(W1-W)/D)
```

```
6     D1=R2MAX/NHOR
      NR1=(P/D1)+1
      IF(NR1.EQ.NHOR)W4=0.
      IF(W4.EQ.0.)GO TO 21
      W4=FN(NR1+1,NH)
21    W3=FN(NR,NH)
      FN2=W3+((B-(NR1*D1))*(W4-W3)/D1)
      NLEVEL=FN1+(ZA-Z1)*(FN2-FN1)/DELX3
      IF(NLEVEL-VUL(1))23,22,22
22    ARRAY(N)=2
      GO TO 24
23    ARRAY(N)=4
24    IF(ARRAY(N).EQ.2.OR.K.EQ.1)GO TO 8
      IF(FN1.EQ.0.)FG1=0.
      IF(FG1.EQ.0.)GO TO 7
      IF(W1.EQ.0.)GO TO 25
      W1=FG(NR+1,NL)
25    W=FG(NR,NL)
      FG1=W+((B-(NR*D))*(W1-W)/D)
7     IF(W4.EQ.0.)GO TO 26
      W4=FG(NR1+1,NH)
26    W3=FG(NR1,NH)
      FG2=W3+((B-(NR1*D1))*(W4-W3)/D1)
      GLEVEL=FG1+((ZA-Z1)*(FG2-FG1)/DELX3)
      IF (GLEVEL-VUL(2))28,27,27
27    ARRAY(N)=3
      GO TO 8
28    ARRAY(N)=4
 8    CALL NOTICE(ARRAY,GLEVEL,NLEVEL,BURST,VUL,N,MODE,ACPOS,K)
9     N=N+1
      W1=1.
      GO TO 2
10    RETURN
      END
```

Subroutine NOTICE

This subroutine is called by subroutine CHECK and its only function is to print output detailing the neutron and gamma levels experienced by the aircraft. The printout also states if the aircraft survived the neutron and gamma levels.

Subroutine NOTICE Glossary:

ACPOS	The aircraft positions
BURST	The burst coordinates
GLEVEL	The gamma level experienced by the aircraft
MODE	The units of GLEVEL and NLEVEL
NLEVEL	The neutron level experienced by the aircraft
VUL	The aircraft neutron and gamma vulnerability

```
      SUBROUTINE NOTICE(ARRAY,GLEVEL,NLEVEL,BURST,VUL,N,MODE,ACPOS,
C
C        THIS SUBROUTINE WILL PRINT OUT THE RESULTS
C        OF THE GAMMA AND NEUTRON ENVIROMENT ON THE
C        AIRCRAFT.  IF THE AIRCRAFT HAS BEEN KILLED
C        IT WILL STATE AIRCRAFT KILLED AND GIVE
C        BURST AND AIRCRAFT LOCATION, VULNERABILITY
C        LEVEL, AND ACTUAL GAMMA OR NEUTRON LEVEL.
C        IF THE AIRCRAFT SURVIVES, BUT DID EXPERIENCE
C        SOME GAMMA AND NEUTRON LEVEL, THE PRINTOUT
C        WILL STATE THE AIRCRAFT SURVIVED, GIVE
C        AIRCRAFT AND BURST LOCATION, AND GIVE ACTUAL
C        GAMMA OR NEUTRON LEVEL.  IF THE AIRCRAFT
C        SURVIVED AND WAS OUTSIDE THE MESH AREA
C        (THAT IS, THE AIRCRAFT DID NOT EXPERIENCE
C        ANY GAMMA OR NEUTRON LEVEL), THE PRINTOUT
C        WILL STATE THE AIRCRAFT SURVIVED, GIVE THE
C        AIRCRAFT AND BURST LOCATION, AND STATE THAT
C        THE EFFECTIVE GAMMA AND NEUTRON LEVEL WAS
C        ZERO.
C
      REAL NLEVEL
      INTEGER ARRAY(100)
      DIMENSION BURST(3),VUL(2),MODE(2),ACPOS(3,100)
      COMMON I,M,L,J,KK
      M=MODE(1)
      L=MODE(2)
      J=ARRAY(N)
      KK=1
      PRINT 1,N,(ACPOS(I,N),I=1,3)
      GO TO (10,11,12,13)J
   10 PRINT 2,(BURST(I),I=1,3)
      PRINT 3
      GO TO 50
   11 GO TO (14,15)K
   14 PRINT 5,(BURST(I),I=1,3)
   32 IF(M.EQ.8)GO TO 16
   17 PRINT 6,VUL(1),NLEVEL
      GO TO 50
   16 PRINT 7,VUL(1),NLEVEL
      GO TO 50
   15 PRINT 4,(BURST(I),I=1,3)
   33 IF(M.EQ.3)GO TO 17
      IF(M.EQ.6)GO TO 16
   23 PRINT 8,VUL(1),NLEVEL
      GO TO 50
   12 PRINT 51,(BURST(I),I=1,3)
   31 IF(L.EQ.2)GO TO 19
      IF(L.EQ.4)GO TO 22
      PRINT 6,VUL(2),GLEVEL
      GO TO 50
   19 PRINT 9,VUL(2),GLEVEL
      GO TO 50
   22 PRINT 7,VUL(2),GLEVEL
      GO TO 50
```

```
13     GO TO (20,21)K
20     PPINT 52,(BURST(I),I=1,3)
       GO TO 32
21     PRINT 2,(BURST(I),...3)
       PRINT 53
       GO TO 3?
5]     IF(J.NE.4)GO TO 100
       IF(KK.EQ.2)GO TO 100
       KK=2
       PRINT 54
       GO TO 31
1      FORMAT(////,3X,*AIRCRAFT NUMBER *,I4,* LOCATED AT*,F8.3,* KM.,*,
      AF8.3,* KM.,*,F8.3,* KM.*)
2      FORMAT(4X,*AIRCRAFT SURVIVED GAMMAS AND NEUTRONS FROM BURST AT*,
      AF8.3,* KM.,*,F8.3,* KM.,*,F8.3,* KM.*)
3      FORMAT(4X,*EFFECTIVE GAMMA AND NEUTRON LEVEL WAS ZERO*)
4      FORMAT(4X,*KILLED BY NEUTRONS FROM BURST AT*,F8.3,* KM.,*,F8.3,*
      AM.,*,F8.3,* KM.*)
5      FORMAT(4X,*KILLED BY COMBINED GAMMA + NEUTRON DOSE FROM BURST AT*
      AF8.3,* KM.,*,F8.3,* KM.,*,F8.3,* KM.*)
6      FORMAT(4X,*VULNERABILITY LEVEL WAS*,1PE11.4,* RADS (SILICON DOSE)
      A ACTUAL LEVEL WAS*,1PE11.4,* RADS (SILICON DOSE)*)
7      FORMAT(4X,*VULNERABILITY LEVEL WAS*,1PE11.4,* RADS (TISSUE DOSE),
      AACTUAL LEVEL WAS*,1PE11.4,* RADS (TISSUE DOSE)*)
8      FORMAT(4X,*VULNERABILITY LEVEL WAS*,1PE11.4,* NEUTRONS/CM2, ACTUA
      A LEVEL WAS*,1PE11.4,* NEUTRONS/CM2*)
9      FORMAT(4X,*VULNERABILITY LEVEL WAS*,1PE11.4,* GAMMAS/CM2, ACTUAL
      AEVEL WAS*,1PE11.4,* GAMMAS/CM2*)
51     FORMAT(4X,*KILLED BY GAMMAS FROM BURST AT*,F8.3,* KM.,*,F8.3,* KM
      A,*,F8.3,* KM.*)
52     FORMAT(4X,*AIRCRAFT SURVIVED COMBINED GAMMA + NEUTRON DOSE FROM B
      ARST AT*,F8.3,* KM.,*,F8.3,* KM.,*,F8.3,* KM.*)
53     FORMAT(4X,*NEUTRONS*)
54     FORMAT(4X,*GAMMAS*)
100    RETURN
       END
```

Subroutine MAP

This subroutine controls the plot output. It calls subroutine CONTOUR to calculate the isofluence or isodose lines and subroutine SETUP to set up the plots. The actual plotting of the lines is done in this subroutine.

This subroutine, as written, is designed for use on an on-line plotter. With the addition of two cards, however, this subroutine, and therefore the entire code, can be used at the computer center where only an off-line plotter is available. The two cards are:

```
CALL PLOTS(WORKA,1024,7)
CALL PLOTE
```

The first card should be the first instruction in this subroutine and the second card should be the last instruction.

Using this option will require different control cards since a magnetic tape is necessary. Therefore, the user should check the local instructions before this option is exercised. Also, the first card of the MAIN program must be changed. The file called PLOT should be replaced with a file called TAPE7. Tape 7 is then the magnetic tape required for the off-line plotter.

Subroutine MAP Glossary:

ALT	The height of the burst in km
DATA	The isofluence or isodose data calculated by CONTOUR. This array stores the r,z coordinates for the isofluence or isodose points

SCALER	The x axis scale factor for the plot routines
SCALEZ	The y axis scale factor for the plot routines
WORKA	An array that is unused in this program. If the program is converted for use on an off line plotter, this array will be the work area needed for the plot calculations

```
      SUBROUTINE MAP(LMAP,NUP,NHOR,POS,MODE,VUL,HOB,DELR,X1MIN,DELX1,
     AX3SW,X3MIN,DELX3,L,ARRAY,NAC    ,H)
C
C         THIS SUBROUTINE PLOTS THE GAMMA AND NEUTRON
C         ENVIRONMENT.  THE TYPE OF PLOT IS DETERMINED
C         BY MODE AND LMAP.
C
      DIMENSION MODE(2),VUL(2),POS(2,100)
      INTEGER ARRAY(100)
      COMMON BL(15),DATA(300,16),N(8),F(60,120),RAD,SCALER,SCALEZ,K,M,
     ANPASS,KK,LIMIT,J,LCOUNT,I,NPO,ALT,X,Y,B,NB,BA,NSTOP,NCOUNT,HI,C,D
     ABB,WORKA(1024)
C
C         DETERMINE WHAT PLOTS WILL BE DONE.
C
      REWIND 10
      ALT=HOB*1.0E-05
      NPASS=1
      NSTOP=2
      K=MODE(NPASS)
      IF(K.EQ.5.OR.K.EQ.6)NSTOP=1
      S=H*1.0E-05
      IF(X3SW.EQ.1.0E16)80,81
 80   TT=X3MIN+(NUP*DELX3)
      IF(L.EQ.1)TT=TT-DELX3
      RAD=S*NHOR*DELX1*EXP(EXP(TT))
      GO TO 1
 81   RAD=NHOR*DELR*1.0E-05
 1    IF(NPASS.GT.NSTOP)GO TO 15
      K=MODE(NPASS)
      M=LMAP
      IF(LMAP.EQ.3)M=1
C
C         CONSTRUCT CONTOUR CURVES FROM POINT VALUE DATA.
C
      A=VUL(NPASS)
      KK=1
 2    CALL CONTOUR(M,K,KK,NHOR,NUP,A,S,DELX3,DELX1,X1MIN,X3MIN,X3SW,DELR
     A,NN,DATA,N,F)
C
C         SET UP THE PLOT.
C
      CALL SETUP(M,K,RAD,SCALER,SCALEZ,WORKA)
      HI=ALT*SCALEZ
C
C         DRAW THE PLOT
C
      LIMIT=2
      IF(M.EQ.2)LIMIT=15
      LCOUNT=0
      X=3.8
      Y=.3
      CALL SYMBOL(0.0,HI ,0.1,11,0.0,-1)
      CALL SYMBOL(X    ,Y    ,0.1,11,0.0,-1)
      CALL SYMBOL(999.,999.,0.1,6H BURST,0.0,6)
```

```
3     LCOUNT=LCOUNT+2
      IF(LCOUNT.GT.LIMIT)GO TO 5
      J=LCOUNT/2
      NPO=N(J)
      IF(NPO.EQ.1)GO TO 3
      DO 4 I=2,NPO
      C=DATA(I,LCOUNT-1)*SCALEP
      D=DATA(I,LCOUNT)*SCALEZ
4     CALL SYMBOL(C,D,0.1,J,0.0,-1)
10    Y=Y+.2
      B=DATA(1,LCOUNT-1)
      IF(M.NE.1)GO TO 12
      BP=ALOG10(B)
      NB=BP
      AP=10
      B=B/(BB**NB)
      GO TO 11
12    NN=NN-1
      BN=NN
      BA=10
11    CALL SYMBOL(X,Y,0.1,J,0.0,-1)
      IF(M.EQ.1)GO TO 20
      CALL NUMBER(X+.2,999.0,0.1,BA,0.0,-1)
      CALL NUMBER(999.,Y+0.05,0.1,BN,0.,-1)
      GO TO 3
20    CALL NUMBER(X+.2,999.0,0.1,B,0.0,+1)
      CALLSYMBOL(X+.7,Y    ,0.1,4,0.,-1)
      CALL NUMBER(999.,999.,0.1,AB,0.,-1)
      CALL NUMBER(999.,Y+0.05,0.1,BB,0.,-1)
      GO TO 3
5     IF(M.EQ.2)GO TO 28
      NCOUNT=0
25    NCOUNT=NCOUNT+1
      IF(NCOUNT.GT.NAC    )GO TO 27
      IF(ARRAY(NCOUNT).EQ.0)GO TO 25
      IF(POS(1,NCOUNT).GT.2.0)GO TO 25
      I=NCOUNT+54
      C=POS(1,NCOUNT)*SCALEP
      D=POS(2,NCOUNT)*SCALEZ
      CALL SYMBOL(C,D,0.1,I,0.0,-1)
      GO TO 25
27    CALL SYMBOL(2.9,0.1,0.1,19H1,2,3,ETC. AIRCRAFT,0.0,19)
28    CALL PLOT(12.0,-2.0,-3)
      IF(LMAP.EQ.3)6,7
6     IF(M.EQ.2)7,9
9     M=2
      KK=2
      GO TO 2
7     NPASS=NPASS+1
      GO TO 1
15    NCOUNT=0
16    NCOUNT=NCOUNT+1
      IF(NCOUNT.GT.NAC    )GO TO 50
      IF(ARRAY(NCOUNT).EQ.4)ARRAY(NCOUNT)=1
      IF(ARRAY(NCOUNT).EQ.2.OR.ARRAY(NCOUNT).EQ.3)ARRAY(NCOUNT)=0
```

```
      GO TO 16
50    REWIND 10
      RETURN
      END
```

Subroutine CONTOUR

This subroutine calculates points on isofluence or isodose lines. The subroutine is constructed so that the entire meshed area is scanned; however, if any points appear that will be out of the range of the plot, this point will not be stored.

Subroutine CONTOUR Glossary:

A	The fluence or dose value for which the line is being calculated
DATA	The results of the mesh scan. This array contains the r,z coordinates of the points
F	The total fluence or dose
NPO	The number of points defining an isofluence or isodose line

```
      SUBROUTINE CONTCUR(M,K,KK,NHCR,NUP,VUL,S,DELX3,DELX1,X1MIN,X3MI
     AX3SW,DELR,NN,DATA,N,F)
      COMMON A,LIMIT,B,II,J,NPO,K1,J1,I,D,X3,Z,R,DEL,X1
      DIMENSION DATA(300,16),N(8),F(60,120)
C
C          THIS SUBROUTINE SCANS THE MESHED AREA
C          AND DETERMINES ,FOR ANY GIVEN VALUE OF
C          FLUENCE OR DOSE ,WHERE THE VALUES EXIST IN
C          TERMS OF RADIUS AND HEIGHT
C
      IF(KK.NE.1)GO TO 1
      READ (10)((F(I,J),I=1,NHCR),J=1,NUP)
 1    GO TO (2,3)M
 2    A=VUL
      J=0
      LIMIT=2
      GO TO 8
 3    LIMIT=16
      B=F(1,1)
      DO 4 I=1,NUP
 4    B=AMAX1(B,F(1,I))
      NN=0
 5    B=B/10
      NN=NN+1
      IF(B.LT.1.)6,5
 6    NN=NN-2
      A=10.**NN
      J=0
 7    A=A/10
 8    NPO=1
      J=J+2
      IF(J.GT.LIMIT)GO TO 40
      K1=NUP-1
      DATA (NPO,J-1)=A
      NPO=NPO+1
      DO 9 II=1,NHCR
      DO 10 J1=2,K1
      IF(F(II,J1).LT.A)GO TO 10
      IF(F(II,J1-1).GT.A)GO TO 11
      DEL=DELX3*(A-F(II,J1-1))/(F(II,J1)-F(II,J1-1))
      X3=X3MIN+(DELX3*(J1-2))+DEL
      GO TO 12
 11   IF(F(II,J1+1).GT.A)GO TO 10
      DEL=DELX3*(A-F(II,J1))/(F(II,J1+1)-F(II,J1))
      X3=X3MIN+(DELX3*(J1-1))+DEL
 12   Z=S*EXP(X3)
      IF(X3.GE.X3SW)GO TO 13
      X1=X1MIN+(DELX1*(II-1))
      R=S*X1*EXP(EXP(X3))
      GO TO 14
 13   R=(X1MIN+((II-1)*DELR))*1.0E-05
 14   IF(R.GT.2.0.)GO TO 10
      DATA(NPO,J-1)=R
      DATA(NPO,J)=Z
      NPO=NPO+1
```

```
      IF(NPO.EQ.300)GO TO 21
10    CONTINUE
9     CONTINUE
      K1=NHOR-1
      DO 15 J1=1,NUP
      X3=X3MIN+(DELX3*(J1-1))
      D=DELX1
      IF(X3.GE.X3SW)D=DELR
      Z=S*EXP(X3)
      DO 16 I=2,K1
      IF(F(I,J1).LT.A)GO TO 16
      IF(F(I-1,J1).GT.A)GO TO 17
      DEL=D*(A-F(I-1,J1))/(F(I,J1)-F(I-1,J1))
      X1=X1MIN+(D*(I-2))+DEL
17    IF(F(I+1,J1).GT.A)GO TO 16
      DEL=D*(A-F(I,J1))/(F(I+1,J1)-F(I,J1))
      X1=X1MIN+(D*(I-1))+DEL
20    IF(X3.GE.X3SW)GO TO 18
      R=S*X1*EXP(EXP(X3))
      GO TO 19
18    R=X1*1.0E-35
19    IF(R.GT.200.)GO TO 16
      DATA(NPO,J-1)=R
      DATA(NPO,J)=Z
      NPO=NPO+1
      IF(NPO.EQ.300)GO TO 21
16    CONTINUE
15    CONTINUE
21    J1=J/2
      N(J1)=NPO-1
      GO TO 7
40    RETURN
      END
```

Subroutine SETUP

This subroutine completely sets up the plot. The axes are drawn and the correct axis labels are selected and drawn. In addition, this subroutine selects the plot title based on what type of plot is to be drawn. The scale factors required by MAP for plotting are also calculated.

```
      SUBROUTINE SETUP(M,MODE,RAD,SCALER,SCALEZ,WORKA)
      DIMENSION WORKA(1024)
      COMMON K,MM,L,N,Y,X,Q,CA,I,D,E,DD
      COMMON/TITLES/IA(11,3),T(5,22)
C
C          THIS SUBROUTINE DRAWS AND LABELS THE AXIS
C          AND TITLES THE FIGURE.
C
C
C          FIRST LOCATE THE ORIGIN.
C
      CALL PLOT(2.0,-7.0,-3)
      CALL PLOT(0.0,1.5,-3)
      CALL PLOT(6.0,0.0,2)
      CALL PLOT(6.0,8.45,2)
      CALL PLOT(0.0,8.45,2)
      CALL PLOT(0.0,0.0,2)
C
C          SET UP THE FIGURE TITLES.
C
      K=MODE
      GO TO (1,2)M
    1 GO TO(3,4,5,5,6,5,5,6)MODE
    3 MM=17
      GO TO 9
    4 MM=18
      GO TO 9
    5 MM=19
      GO TO 9
    6 MM=20
      GO TO 9
    2 K=MODE+8
      GO TO (7,7,7,7,8,7,7,8)MODE
    7 MM=21
      GO TO 9
    8 MM=22
C
C          SET UP THE AXIS LABELS.
C
    9 D=6HHEIGHT
      E=6HRADIUS
      DD=5H (KM)
C
C          DETERMINE THE RADIUS AND HEIGHT MAXIMUM
C          COORDINATES.  DETERMINE THE RADIUS AND
C          HEIGHT SCALE FACTORS.
C
      IF(RAD.GT.50.)GO TO 10
      N=5
      L=1
      GO TO 12
   10 IF(RAD.GT.100.)GO TO 11
      N=10
      L=2
      GO TO 12
```

```
 11   N=20
      L=3
 12   SCALEZ=8./100.
      SCALER=5./(N*10.)
C
C     DRAW THE AXIS.
C
      CALL PLOT(0.9,0.42,-3)
      CALL PLOT(0.0,8.0,2)
      CALL PLOT(0.0,0.0,3)
      CALL PLOT(5.0,0.0,2)
      CALL PLOT(0.0,0.0,3)
C
C         DRAW THE TIC MARKS.
C
      Q=8./10.
      Y=0.
      DO 13 I=1,11
      CALL PLOT(0.,Y,3)
      CALL PLOT(-0.07,Y,2)
 13   Y=Y+Q
      QA=5./10.
      X=0.
      DO 14 I=1,11
      CALL PLOT(X,0.,3)
      CALL PLOT(X,-0.07,2)
 14   X=X+QA
C
C         NUMBER THE AXIS.
C
      Y=-0.1
      DO 15 I=1,11
      CALL SYMBOL(-.4,Y,0.1,IA(I,2),0.,3)
 15   Y=Y+Q
      X=-.2
      DO 16 I=1,11
      CALL SYMBOL(X,-.2,0.1,IA(I,L),0.,3)
 16   X=X+QA
C
C         LABEL THE AXIS.
C
      CALL SYMBOL(1.75,-.4,0.13,E,0.  ,6)
      CALL SYMBOL(999.,999.,0.13,DD,0.0,5)
      CALL SYMBOL(-.5,3.25,0.13,D,90.0,6)
      CALL SYMBOL(999.,999.,0.13,DD,90.0,5)
C
C         LABEL THE FIGURE.
C
      CALL SYMBOL(-0.9,-0.65,0.13,T(1,K),0.0,46)
      CALL SYMBOL(-0.9,-0.95,0.13,T(1,MM),0.0,46)
      RETURN
      END
```

BLOCK DATA

This subroutine stores data into labeled common.

BLOCK DATA Glossary:

T	An array that contains the titles to all the plots that can be drawn by subroutine MAP
IA	An array containing three sets of coordinates for the plot axis
SPECFN	The fission weapon neutron output spectrum
SPECFG	The fission weapon gamma output spectrum
SPECTNN	The thermonuclear weapon neutron output spectrum
SPECTNG	The thermonuclear weapon gamma output spectrum

```
      BLOCK DATA
      COMMON/SPECTRA/SPECTNN(22),SPECFN(22),SPECFG(18),SPECTNG(18)
      COMMON/TITLES/IA(11,3),T(5,22)
C
      DATA ((T(I,J),I=1,5),J=1,10)/10H    FIG.  ,10H  NEUTRON ,
     U10HFLUENCE VU,10HLNERABILIT,
     A6HY      ,10H     FIG. ,10H   GAMMA F,10HLUENCE VUL,10HNERABILITY,
     B6H       ,10H  FIG.    ,10HNEUTRON TI,10HSSUE DOSE ,10HVULNERABIL,
     C6HITY    ,10H   FIG.   ,10H GAMMA TIS,10HSUE DOSE V,10HULNERABILI,
     D6HTY     ,10H     FIG. ,10H   NEUTRON,10H + GAMMA T,10HISSUE DOSE,
     E6H       ,10H  FIG.    ,10HNEUTRON SI,10HLICON DOSE,10H VULNERABI,
     F6HLITY   ,10H   FIG.   ,10H GAMMA SIL,10HICON DOSE ,10HVULNERABIL,
     G6HITY    ,10H     FIG. ,10H   NEUTRON,10H + GAMMA S,10HILICON DOS,
     H6HE      ,10H   FIG.   ,10H NEUTRON I,10HSOFLUENCE ,10HLINES (N/C,
     I6HM2)    ,10H    FIG.  ,10H  GAMMA IS,10HOFLUENCE L,10HINES (G/CM,
     J6H2)     /
      DATA ((T(I,J),I=1,5),J=11,22)/10H FIG.    N,10HEUTRON TIS,
     X10HSUE ISODOS,10HE LINES (R,
     K6HADS)   ,10H  FIG.    ,10HGAMMA TISS,10HUE ISODOSE,10H LINES (RA,
     L6HDS)    ,10H    FIG.  ,10H  NEUTRON ,10H+ GAMMA TI,10HSSUE ISODO,
     M6HSE     ,10H FIG.    N,10HEUTRON SIL,10HICON ISODO,10HSE LINES (,
     Y6HRADS)  ,10H  FIG.    ,10HGAMMA SILI,10HCON ISODOS,10HE LINES (R,
     W6HADS)   ,10H   FIG.   ,10H NEUTRON +,10H GAMMA SIL,10HICON ISODO,
     N6HSE     ,10H          L,10HINE (N/CM2,10H) EXPONENT,10HIAL AIR   ,
     O6H       ,10H          L,10HINE (G/CM2,10H) EXPONENT,10HIAL AIR   ,
     P6H       ,10H          L,10HINE (RADS),10H EXPONENTI,10HAL AIR    ,
     Q6H       ,10H  VULNERAB,10HILITY LINE,10H (RADS) EX,10HPONENTIAL ,
     R6HAIR    ,10H           ,10H     EXPON,10HENTIAL AIR,10H          ,
     S6H       ,10H          L,10HINES (RADS,10H) EXPONENT,10HIAL AIR   ,
     T6H       /
C
      DATA IA/3H  0,3H  5,3H 10,3H 15,3H 20,3H 25,3H 30,3H 35,3H 40,
     A3H 45,3H 50,3H  0,3H 10,3H 20,3H 30,3H 40,3H 50,3H 60,3H 70,3H 80
     B3H 90,3H100,3H  0,3H 20,3H 40,3H 60,3H 80,3H100,3H120,3H140,3H160
     C3H180,3H200/
C
      DATA SPECFN/3.925E+19,2.233E+20,8.7E+20,3.48E+21,8.705E+21,8.705E+
     A21,3*1.4951E+22,2*4.23E+22,2*4.2325E+22,3.875E+21,8*0.0/
C
      DATA SPECTNN/6.001E+22,2.176E+22,1.1985E+22,1.2495E+22,
     A2*1.4875E+22,3*1.4167E+22,2*3.825E+22,2*7.9475E+22,3.1025E+23,
     B8*0.0/
C
      DATA SPECFG/1.2648E+19,5.9019E+19,1.0539E+20,4.7555E+20,4.7555E+2
     A,1.0719E+21,1.0719E+21,2.3562E+21,3.22E+21,4.0830E+21,3.7741E+21,
     B4.512E+21,5.2499E+21,2*2.2981E+21,2.7062E+21,2*0.0/
C
      DATA SPECTNG/6.3279E+18,2.9508E+19,5.2693E+19,2*2.3776E+20,
     A2*5.3595E+20,1.1781E+21,1.615E+21,2.0419E+21,1.887E+21,2.2558E+21
     B2.6249E+21,2*1.149E+21,1.3531E+21,2*0.0/
      END
```

Appendix C

Sea Level Air Cross Sections

The following pages contain the 40 group sea level air cross sections used in the code. This page gives the instructions on how to read these pages.

The data is listed in 40 columns, one for each group. The first column is neutron group one, the second column is neutron group two, and so on. Since the neutron and gamma cross sections are coupled, the data can be regarded as one 40 group cross section set. Therefore, column 23, which is gamma group one, can be considered as group 23. Each column contains 43 values for group cross sections. The first number is the group transport cross section, the second number is the group removal cross section, the third number is the group total cross section, and the fourth number is the in group scatter cross section. The remaining numbers in the column are values for the group to group scatter cross sections. So, the fifth number is scatter to that group G from group G-1, the sixth number is scatter to that group G from group G-2, and so on. All the cross sections are macroscopic with units of cm^{-1}.

3.9972E-04	3.7487E-04	3.0201E-04	1.6247E-04	7.9725E-05	3.0211E-05	2.1526E-05	4.4377E-05
3.9444E-05	3.6785E-05	1.4118E-05	1.7814E-05	2.5462E-05	3.5899E-05	4.9537E-05	6.1127E-05
5.0792E-04	4.7665E-04	3.6490E-04	2.1168E-04	1.3504E-04	1.3415E-04	9.6579E-05	7.1174E-05
4.6847E-04	4.3986E-04	3.5078E-04	1.9387E-04	1.0958E-04	9.9270E-05	4.8932E-05	1.[illegible]49E-05
3.6417E-05	1.4036E-05	1.7486E-05	2.3670E-05	3.2238E-05	4.4534E-05	5.4799E-05	1.1565E-05
0.	0.	0.	0.	0.	1.0383E-06	2.9774E-05	3.5629E-06
0.	0.	0.	0.	0.	0.	9.0540E-07	5.2983E-09
0.	0.	0.	0.	0.	4.6484E-08	1.3157E-07	8.6331E-08
0.	0.	0.	9.4229E-08	1.6368E-07	7.4712E-07	3.9347E-07	1.7[illegible]E-07
0.	0.	3.1519E-08	1.2574E-07	2.2533E-07	5.1156E-07	7.6299E-07	4.7243E-07
0.	6.8630E-11	6.5213E-09	9.1952E-08	3.0559E-07	1.4673E-06	2.63[illegible]9E-06	4.9027E-07
3.3672E-12	7.8797E-12	4.6558E-09	1.2141E-07	1.1556E-06	1.5[illegible]E-06	2.17[illegible]9E-06	4.0[illegible]21E-07
4.9930E-13	5.8811E-12	6.3443E-09	4.1074E-07	9.3[illegible]9E-07	2.0022E-06	1.995[illegible]E-06	0.
2.3540E-13	7.7875E-12	1.3721E-08	3.1848E-07	1.4311E-06	2.3255E-06	0.	0.
3.0233E-13	1.3834E-11	1.5427E-08	5.3424E-07	1.2472E-06	0.	0.	0.
2.5572E-13	2.0569E-11	2.5679E-09	4.5635E-07	0.	0.	0.	0.
1.4412E-12	3.6302E-11	2.1739E-08	0.	0.	0.	0.	0.
2.7514E-12	3.1826E-11	0.	0.	0.	0.	0.	0.
2.1906E-12	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.

3.1473E-05	3.4609E-05	7.5861E-05	4.4473E-05	4.5947E-05	5.2539E-05	5.3619E-05	5.1254E-05
3.5098E-05	3.9722E-05	4.4094E-05	5.1732E-05	5.5963E-05	6.2907E-05	6.8287E-05	7.4200E-05
3.7458E-05	4.2300E-05	4.8295E-05	5.4319E-05	6.1097E-05	6.9224E-05	7.5903E-05	8.6398E-05
[illegible]	2.5976E-06	4.2018E-06	7.4867E-06	5.1337E-06	5.3179E-06	7.6158E-06	1.2197E-05
3.2793E-06	3.4301E-06	5.6749E-05	4.6956E-06	7.3590E-06	7.4414E-06	1.0756E-05	1.5861E-05
2.2594E-06	2.7213E-06	3.7152E-06	2.9597E-06	4.7851E-06	5.0735E-06	7.2571E-06	1.0725E-05
7.8418E-06	1.6756E-06	2.6155E-06	2.0537E-06	3.2071E-06	3.4699E-06	5.1871E-06	7.6914E-06
7.0491E-07	5.7930E-07	1.9585E-06	1.5069E-06	2.3140E-06	2.4386E-06	7.7127E-06	5.8008E-06
4.2715E-07	1.0701E-07	1.1593E-05	1.1061E-06	1.7477E-06	1.8195E-06	2.7143E-06	4.3018E-06
3.7697E-07	6.4474E-08	2.1402E-07	0.	1.3771E-06	1.4045E-06	2.0759E-06	3.3107E-06
1.3467E-07	7.5768E-08	1.2995E-07	0.	0.	1.1220E-06	1.6295E-06	2.5988E-06
7.2778E-08	2.0327E-08	7.1335E-08	0.	0.	0.	1.3147E-06	2.0597E-06
2.5480E-08	1.0979E-08	4.0654E-08	0.	0.	0.	0.	1.6733E-06
8.7492E-09	4.4510E-09	2.1359E-08	0.	0.	0.	0.	0.
3.9582E-10	1.3205E-09	3.9123E-09	0.	0.	0.	0.	0.
0.	5.0748E-11	2.2412E-09	0.	0.	0.	0.	0.
0.	0.	1.1941E-10	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
1.0155E-09	0.	0.	0.	0.	0.	0.	0.
8.6925E-07	4.2052E-09	1.1913E-08	3.6243E-08	0.	0.	0.	0.
5.2059E-06	5.6071E-07	4.7539E-09	1.7249E-07	3.6243E-08	0.	0.	0.
1.1180E-05	2.1352E-06	8.2944E-09	3.8992E-07	1.7249E-07	0.	0.	0.
1.2596E-05	7.4398E-06	4.4997E-06	6.7098E-07	3.8792E-07	2.3170E-10	0.	0.
1.1576E-05	5.1292E-06	2.7410E-05	1.3708E-06	1.4714E-06	1.3725E-07	2.7107E-10	0.
0.	6.5934E-06	4.1870E-06	3.7697E-06	2.6069E-06	4.9243E-07	1.6657E-07	0.
0.	0.	3.6798E-05	7.1689E-06	3.8868E-06	1.0242E-06	5.7608E-07	0.
0.	0.	0.	7.4446E-06	7.4330E-06	2.4522E-06	1.1982E-06	0.
0.	0.	0.	0.	1.9285E-06	4.7773E-06	2.8918E-06	1.6510E-08
0.	0.	0.	0.	0.	4.2272E-06	5.7913E-06	1.5710E-06
0.	0.	0.	0.	0.	0.	5.9591E-06	3.3511E-06
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.

Vita

Robert Douglas McLaren was born on 24 April 1938 in Rockville Center, New York. He was graduated from high school in Smithtown, New York, in 1954. He then studied at the Polytechnic Institute of Brooklyn, New York, from which he received the degree of Bachelor of Chemical Engineering in 1959. He enlisted in the Air Force in March 1962 and was commissioned in September 1962. His initial duty was as a communications officer at Carswell Air Force Base, Texas. In 1964 he was transferred to Minot Air Force Base, North Dakota, as an electronic engineer on the Minuteman I weapon system. He has subsequently worked as an electronic engineer on the Minuteman I and III weapon systems at both Minot and Patrick Air Force Base, Florida. He attended the Air Force Institute of Technology where he received the degree of Master of Science in Nuclear Engineering in 1972.

Permanent address: Gaynor Avenue
Nesconset, New York

This thesis was typed by Jane Manemann.