FEB 15 1965

LEGAL NOTICE

This report was prepared as an account of Government sponsored work under the United States, nor the Commission, nor any person acting on behalf of the Commission.

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights.

B. Assumes any liabilities with respect to the use of or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above person acting on behalf of the Commission includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

NOTICE

This report contains patentable, preliminary, unverified, or erroneous information. For one or more of these reasons the author or issuing institution and responsible office have limited its distribution to Governmental agencies and their contractors as authorized by AEC Manual Chapter 3202-062. A formal report will be published at a later date when the data is complete enough to warrant publication.

WANL-TME-896

JULY 31, 1964

MASTER

THE EFFECT OF ORBITAL DESTRUCT OF NUCLEAR ROCKETS ON EARTH CONTAMINATION

DISTRIBUTION OF THIS DOCUMENT UNLIMITED

FOR OFFICIAL RELEASE - OFFICIAL
DISTRIBUTION MAY BE MADE. OFFICIAL
REQUESTS MAY BE FILLED. REPORT
CONTAINS NOTHING OF PATENT INTEREST.
PROCEDURES ON FILE IN RECEIVING
SECTION.

**Astronuclear Laboratory
Westinghouse Electric Corporation**

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

July 31, 1964

NOTICE
This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

MASTER

THE EFFECT OF ORBITAL DESTRUCT OF NUCLEAR
ROCKETS ON EARTH CONTAMINATION

DISTRIBUTION OF THIS DOCUMENT UNLIMITED

PREPARED BY:

H. G. Hargrove

M. R. Trammell

APPROVED BY:

J. E. Faulkner, Manager
Safeguards Engineering

J. M. Bridges, Supervisor
Safeguards Engineering

INFORMATION CATEGORY

UNCLASSIFIED

8/11/64

ACKNOWLEDGMENT

The authors wish to express their grateful appreciation for the generosity and help extended to them by Dr. Joanne Bridges and Mr. C. P. Saalbach. Dr. Bridges prepared in part several sections of this report and performed the arduous task of editing the manuscript. Mr. Saalbach of the Physics and Mathematics Section contributed immeasurably with his advice during the preparation of the NEFØD and NURØDAC computer programs, particularly with "debugging" operations.

TABLE OF CONTENTS

		<u>Page</u>
	<u>SUMMARY</u>	vi
1.0	<u>INTRODUCTION</u>	1
2.0	<u>GENERAL APPROACH TO THE ANALYSES OF IMPORTANT VARIABLES</u>	4
2.1	<u>Problem Approach</u>	4
2.1.1	Evaluation of Concentration	4
2.1.2	Error Analysis	6
2.2	<u>Various Steps Involved in Setting Up the Problem</u>	7
3.0	<u>MATHEMATICAL ANALYSIS OF DESTRUCT PROCESS</u>	9
4.0	<u>PHASE TWO - CALCULATION OF EARTH RANGE</u>	17
4.1	<u>Earth Range Concept</u>	17
4.2	<u>Analysis to Determine Functional Relationship Between λ, Ψ, and V_R</u>	21
4.2.1	Importance of Knowing $\lambda = f(V_R, \Psi)$	21
4.2.2	Method of Least Squares to Determine $\lambda = f(V_R, \Psi)$	24
4.3	<u>Use of NEFØD</u>	26
4.3.1	Limits of Applicability	26
4.3.2	Use of NEFØD	27
4.3.3	Results	28
4.4	Exclusion Velocity Analysis	31
5.0	<u>THE DESTRUCT AREA AND ERROR CALCULATIONS</u>	34
5.1	<u>Calculation of Contamination Area</u>	34
5.2	<u>Method for Determining the Error in the Earth Area as Function of Error in Measuring Destruct Variables</u>	37

TABLE OF CONTENTS (CONT' D)

		<u>Page</u>
5.3	<u>Digital Computer Program for Determining the Earth Area Error Coefficients</u>	41
6.0	<u>TYPICAL EXAMPLES OF THE USE OF NURØDAC</u>	44
	<u>GLOSSARY</u>	61
	<u>BIBLIOGRAPHY</u>	64
Appendix I	<u>METHOD OF LEAST SQUARES FOR FITTING POLYNOMIALS</u>	I-1
Appendix II	<u>GENERAL OUTLINE OF NEFØD COMPUTER PROGRAM FOR POLYNOMIAL FITTING</u>	II-1
	II.1 <u>Contents</u>	II-1
	II.2 <u>Explanation of Program Segments</u>	II-1
	II.2.1 Main Program	II-1
	II.2.2 Subroutine FUNCT	II-2
	II.2.3 Subroutine SLSTQR	II-2
	II.2.4 Subroutine EVAL	II-3
	II.3 <u>Output Format and Information</u>	II-4
Appendix III	<u>DEVIATION OF BASIC SPHERICAL TRIGONOMETRY FORMULAS USED IN AREA CALCULATION</u>	III-1
Appendix IV	<u>DETERMINATION OF AREA</u>	IV-1
Appendix V	<u>EXPLANATION AND DESCRIPTION OF THE NUCLEAR ROCKET ORBITAL DESTRUCT AREA COMPUTATION CODE</u>	V-1
	V.1 <u>Main Program</u>	V-1
	V.2 <u>Subroutine FLIGHT</u>	V-2
	V.3 <u>Subroutine GNU</u>	V-2
	V.4 <u>Subroutine DVELØ</u>	V-2

TABLE OF CONTENTS (CONT' D)

	<u>Page</u>
V.5 <u>Subroutine ØUTPT 1</u>	V-3
V.6 <u>Subroutine DA LPHA</u>	V-3
V.7 <u>Subroutine LAMDA</u>	V-3
V.8 <u>Subroutine SUMUP</u>	V-3
V.9 <u>Subroutine TAREA</u>	V-3
V.10 <u>Subroutine CØEFF</u>	V-4

LIST OF FIGURES

<u>Number</u>	<u>Title</u>	<u>Page</u>
2.1	Schematic Description of Behavior of Destruct Debris	5
3.1	The Co-Ordinate System and Relationship of Variables in the Destruct Vector Analysis	10
3.2	Representation of Spatial Relationship Between Initial, Destruct, and Resultant Velocities	13
4.1	Representation of the Flight Path of a Re-Entering Particle	20
4.2	λ vs Ψ at Constant V_R for a 1000 Micron Diameter Particle	22
4.3	λ vs Ψ at Constant V_R for a 100 Micron Diameter Particle	23
4.4	Permissible Destruct Velocity as a Function of ζ and ϵ for Forward Half of Destruct Sphere	33
5.1	Concept of Destruct Sphere	36
5.2	Relationship of the Variables in the Earth Area and Error Coefficient Analyses	38
6.1	Illustration of Destruct Triangles	57
III-1	Vector Relationships for Spherical Trigonometry	III-2

LIST OF TABLES

<u>Number</u>	<u>Title</u>	<u>Page</u>
4.1	Polynomial Coefficients From NEFØD for 100 μ Particle	29
4.2	Polynomial Coefficients From NEFØD for 1000 μ Particle	30
6.1	Sample Results for Destruct Triangle Number 1	45
6.2	Sample Results for Destruct Triangle Number 2	47
6.3	Sample Results for Destruct Triangle Number 3	49
6.4	Sample Results for Destruct Triangle Number 4	51
6.5	Sample Results for Destruct Triangle Number 5	53
6.6	Sample Results for Destruct Triangle Number 6	55
6.7	Summary of Destruct Data for Typical Destruct Triangle Calculations	59

SUMMARY

This report describes an analytical approach for the determination of the magnitude of earth area contaminated by destruct debris. It also contains an analysis for the estimation of the errors in calculated concentration of debris due to experimental variations in the measurement of destruct parameters. The destruct parameters to be measured are identified. The tools developed include a 7094 digital computer program, NEFØD, which is used to calculate the fraction of an orbit traversed by a destruct particle before it impacts on earth. A second program was developed, NURØDAC, which calculates the contaminated earth area and sets of error coefficients which can be used to calculate variations in earth contamination area as a function of destruct parameters. Some examples of typical error calculations have been included. Complete descriptions of the two computer programs are also included.

1.0 INTRODUCTION

From the outset of the nuclear rocket development program, the inclusion of an explosive destruct system has figured prominently in the overall safety countermeasure considerations. In the original concepts, the explosive destruct system would be designed to pulverize the reactor (particularly the graphite-uranium core) into small fragments capable of being consumed by ablation and oxidation during subsequent atmospheric re-entry. It was thought that the portion of the original particles or residue of the ablated fragments which did survive the re-entry phase would be sufficiently dispersed by the atmosphere that they would not constitute a hazard upon impact with the surface of the earth. Recent studies^(1,2,3,4) have indicated, however, that no substantial degree of burn-up of NERVA fuel may be expected to occur during the re-entry process.

With the basic design principle eliminated, i.e. burnup, the case for an explosive destruct system would seem to have little justification. This is not necessarily so, and the following reasons help to indicate this. First, and most important, the explosive destruct scheme can accomplish dilution of the radioactivity contained in the unburned fragments by scattering the debris over large areas of the earth's surface, perhaps to the extent that statistically no human will receive an adverse amount of radiation. Secondly, the breakup of the reactor precludes the possibility of a water or earth impact nuclear excursion, a situation which might prove to be undesirable. Although anti-criticality devices need not be as drastic in action as the destruct system, the two could be combined to conserve non-payload weight. Third, with an eye to the future, there may arise a development

wherein the oxidation characteristics of the fuel, fuel bearing moderator, etc., may be enhanced by chemical means. Admittedly, such efforts to date have not been successful, but for more sophisticated, advanced designs of nuclear rocket propulsion systems this may not necessarily be so. Therefore, the inclusion of a destruct system warrants further consideration.

Because of the nature of the problem, just being able to fragment the reactor into small pieces is not enough. Such questions arise as: what happens to fragments produced by destruct? Do they hit the earth's surface or go into larger orbits (presuming destruct while the reactor is in orbit)? If they hit the earth, what area will be contaminated by the debris and what will the level of contamination be?

To design a destruct system certainly requires experimental information on the fragmentation characteristics of the nuclear components, the various types of high explosives which can be used, and methods for placing, sequencing, and delivering same. Both the system designer and the experimentalist must ask himself what variables and parameters are important. Once these are isolated, the question then arises as to how accurately these must be known in order to design a system that is properly integrated into the framework of the safety countermeasures program.

The purpose of this report is to analyze the action of destruct to identify the variables (other than those concerned directly with fragmentation) that will be of prime importance in the design of a high explosive destruct system. Once these variables have been isolated, attention will be directed to the task of how to determine analytically the degree of

accuracy to which these variables must be known to define accurately the area of the earth's surface contaminated by these fragments upon termination of re-entry. The analyses given herein are perfectly general, but for the present, attention will be directed to the consideration of destruct in a 100 nautical mile circular orbit, a typical situation as presently envisioned.

2.0 GENERAL APPROACH TO THE ANALYSES OF IMPORTANT VARIABLES

2.1 PROBLEM APPROACH

With the knowledge that nuclear debris will not burn-up during the re-entry process, it becomes necessary to perform a hazards analysis to determine the feasibility of utilizing destruct as a safety countermeasure. To perform such an analysis it is necessary to know (1) the operational history of the reactor, (2) fragment size distribution, (3) the re-entry times of fragments, and (4) the spatial distribution of these fragments on the earth's surface. One obvious means of performing such an analysis would be to study the history of each individual particle created by destruct action. Since it is obviously not possible to evaluate the destruct behavior of individual fragments nor is it feasible to utilize the computer for such a point-by-point analysis, it is necessary to develop some classification scheme which will serve as a guideline for performing destruct measurements and also for performing hazards analyses.

2.1.1 Evaluation of Concentration

Consider a reactor in orbit with a velocity vector $\vec{V}$ and located at position P. (See Figure 2.1) When destruct is employed, a multitude of small particles will be created each of which will be given a discrete destruct velocity, $\vec{V}_D$. This vector will perturb the original velocity of the fragment and will cause the particle to re-enter to some position on the earth's surface called P_1 . Another particle with the same ballistic parameter and the same scalar magnitude of destruct velocity but with different direction will land at a position P_2 on the earth's surface, and a third particle also having the same scalar

SCHEMATIC DESCRIPTION OF BEHAVIOR OF DESTRUCT DEBRIS

FIGURE 2.1

velocity of destruct but still a different direction will be located at position P_3 on the earth's surface. Obviously, if these three particles are not on a straight line, they will define a spherical triangle on an imaginary sphere having as an origin the position of the reactor at the time of destruct and as a radius the velocity of destruct. In addition, the position of these particles upon re-entry will define a spherical triangle on the earth's surface. More important, all particles with identical destruct velocities fluxing through this "destruct triangle" will lie within the confines of the spherical triangle located on the earth's surface. In addition, if this triangle is sufficiently small, all particles lying within it should require about the same times for re-entry, and hence should have similar radioactive inventories. Thus by knowing the number of fragments of a given particle size passing through a given destruct triangle, it becomes possible to evaluate the concentration of debris on the earth's surface and, hence, possible to perform a hazards analysis. The basic problems then become: (1) determining the area occupied by the debris on the earth's surface and (2) defining how destruct measurements must be made to provide sufficient information for calculation of this area.

2.1.2 Error Analysis

The concentration of debris on the earth's surface can be defined as:

$$C_i = \frac{N_i}{A} \quad (2-1)$$

where N_i = the number of particles of the i^{th} class passing through a particular destruct triangle which produces an area A on the earth's surface.

Differentiation of Equation (2-1) yields the following:

$$\frac{dC_i}{C_i} = \frac{dN_i}{N_i} - \frac{dA}{A} \quad (2-2)$$

For the purpose of error analysis Equation (2-2) may be rewritten:

$$\frac{\Delta C_i}{C_i} = \pm \frac{\Delta N_i}{N_i} \mp \frac{\Delta A}{A} \quad (2-3)$$

Thus, the percentage error in measuring concentration is equal to the sum of the errors in measuring the number of particles of a given class and in measuring area. The influence of errors in measuring the number of particles passing through a given destruct triangle is obvious. For example, a 10 per cent error in measuring the number of fragments passing through a given destruct triangle will produce a 10 per cent error in the evaluation of concentration on the earth's surface, providing one knows the correlation between the destruct triangle and the area of the triangle on the earth's surface. However, the definition of this area constitutes a major problem. The area is determined by both the direction and the magnitude of the destruct velocities defining the destruct triangle. The effect of errors in measuring these quantities is not obvious. It is the purpose of this report to evaluate the quantity $\Delta A/A$ in terms of the experimental measurements which can be performed during the testing of destruct techniques. This report is designed to serve as a guideline for the performance of these tests.

2.2 VARIOUS STEPS INVOLVED IN SETTING UP THE PROBLEM

For the purpose of logical or mathematical analysis, many physical problems can be

resolved into several, practically independent, phases each embodying more or less a single physical phenomenon. Such is the case here, for several events occur from the time of destruct until the debris settles to earth. The phases to be considered in this report are:

1. The destruct action itself,
2. Re-entry of fragments,
3. Impact, over some area, of the destruct fragments.

In the first phase, it is necessary to identify the measurable quantities that can be used to characterize the destruct action. This is done in Chapter 3. In the second phase, the parameters required to describe re-entry adequately, e.g., flight angle and resultant velocity, must be related mathematically to the destruct quantities to be measured. This analysis is also described in Chapter 3. Determination of the impact area of debris fragments requires the development of mathematical tools to relate this area to the re-entry and destruct variables. This analysis is shown in Chapters 4 and 5. Finally, for purposes of hazards analyses, it is necessary to find a means of describing the variation in area (and hence in concentration of debris) which is caused by uncertainties in the measurement of destruct variables. This analysis is described in Chapter 5.

3.0 MATHEMATICAL ANALYSIS OF DESTRUCT PROCESS

Consider a body moving in an orbit about the earth. Now it is obvious that any and all component parts, segments, etc. making up the body are all moving along the same path. Now isolate any portion of the original body from the main bulk (consider the new fragment to be produced by destruct). If no effect has been manifested other than splitting the fragment from its parent body, then the new particle will traverse the same flight path as it did when it was part of the large body. Now if it happens that this particle is subjected to a force that accelerates it away from its original position in the bulk, then it will no longer behave in the manner previously, but assume a new flight path independent of the original body. This is precisely what occurs during a high explosive destruct. Part of the energy of the high explosive goes into the production of particles from the parent bulk, and, except for inefficiencies, the remainder manifests itself as kinetic energy input to the particles, causing the fragments to be accelerated from their original flight path.

Let us consider a particle that has just been split from a parent body which is in orbit. Then let us impose an additional thrust to simulate the kinetic energy input to the body. Referring to Figure 3.1, place the particle of interest at the origin of three-dimensional coordinate system whose axes, x , y , and z are mutually orthogonal. Consider the particle's original line of flight to be in the " y - z " plane and at the moment of destruct the particle has a flight path angle, ϕ , with respect to a local horizontal, which is the " x - y " plane⁽⁵⁾. The particle has an initial orbital velocity, $\vec{V}$. Then apply the additional thrust at an arbitrary direction to the original flight plane. Let this additional thrust, which manifests itself as a

PICTORIAL OF THE CO-ORDINATE SYSTEM AND
RELATIONSHIP OF VARIABLES IN THE DESTRUCT VECTOR ANALYSIS

FIGURE 3.1

"destruct velocity", V_D , be located in space by two angles, ϵ , and ζ . The first angle ϵ , is measured in a counter-clockwise sense from the plane of the original flight path to the plane containing the destruct velocity. The second angle, ζ , is measured from the local horizontal to the destruct velocity in the plane of the destruct velocity. The angle ϵ may have any value from zero to 2π radians; ζ lies between minus $\pi/2$ and plus $\pi/2$ radians. Since the action of destruct by itself is unaffected by whether the body is in motion or not, the three quantities V_D , ϵ , and ζ , if measured in a static experiment, can thus be used to determine the new flight conditions of the orbiting particle.

We proceed as follows:*

$$(\vec{V}_D)_x = (V_D \cos \zeta \sin \epsilon) \hat{i} \quad (3-1a)$$

$$(\vec{V}_D)_y = (V_D \cos \zeta \cos \epsilon) \hat{j} \quad (3-1b)$$

$$(\vec{V}_D)_z = (V_D \sin \zeta) \hat{k} \quad (3-1c)$$

$$(\vec{V})_x = 0 \quad (3-2a)$$

$$(\vec{V})_y = (V \cos \phi) \hat{j} \quad (3-2b)$$

$$(\vec{V})_z = (V \sin \phi) \hat{k} \quad (3-2c)$$

Now the resultant velocity, $\vec{V}_R$, will have components in the i , j and k directions whose values are given below:

$$(\vec{V}_R)_x = (V_D \cos \zeta \sin \epsilon) \hat{i} \quad (3-3a)$$

* Nomenclature will be found in the Glossary.

$$(\vec{V}_R)_y = (V_D \cos \zeta \cos \epsilon + V \cos \phi) \hat{i} \quad (3-3b)$$

$$(\vec{V}_R)_z = (V_D \sin \zeta + V \sin \phi) \hat{k} \quad (3-3c)$$

The resultant velocity is now:

$$\vec{V}_R = (\vec{V}_R)_x + (\vec{V}_R)_y + (\vec{V}_R)_z, \quad (3-4a)$$

or

$$\begin{aligned} \vec{V}_R = & (V_D \cos \zeta \sin \epsilon) \hat{j} + (V_D \cos \zeta \cos \epsilon + V \cos \phi) \hat{i} \\ & + (V_D \sin \zeta + V \sin \phi) \hat{k} \end{aligned} \quad (3-4b)$$

Now by scalar multiplication of the vector, $\vec{V}_R$, we obtain:

$$V_R^2 = \vec{V}_R \cdot \vec{V}_R,$$

or

$$\begin{aligned} V_R^2 = & V^2 + V_D^2 + 2VV_D (\cos \zeta \cos \epsilon \cos \phi \\ & + \sin \zeta \sin \phi) \end{aligned}$$

(3-4c)

Equation (3-4c) now gives us an expression for the magnitude of the resultant velocity as a function of the three destruct quantities, V_D , ζ , ϵ and the two original flight parameters, V and ϕ .*

We now turn our attention to determining the direction the resultant velocity makes with respect to the co-ordinate system. Referring to Figure 3.2, note that the plane

* In this report, all calculations are carried out for $\phi = 0$ since only a circular orbit has been considered in this study.

PICTORIAL REPRESENTATION OF SPATIAL RELATIONSHIP BETWEEN
INITIAL, DESTRUCT, AND RESULTANT VELOCITY

FIGURE 3.2

containing the resultant velocity lies along the θ direction. The intersection of this plane with the "x-y" or horizontal plane, coincides with the projection of the resultant velocity on this plane. We shall term the projected resultant velocity on the "x-y" plane, $\vec{(V_R)_{xy}}$.

Then,

$$\vec{(V_R)_{xy}} = \vec{(V_R)_x} + \vec{(V_R)_y} \quad , \quad (3-5a)$$

and

$$(V^2_{R_{xy}}) = \vec{(V_R)_{xy}} \cdot \vec{(V_R)_{xy}} = [\vec{(V_R)_x} + \vec{(V_R)_y}] \cdot [\vec{(V_R)_x} + \vec{(V_R)_y}] \quad (3-5b)$$

Since $\vec{(V_R)_x}$ and $\vec{(V_R)_y}$ lie perpendicular to each other, scalar multiplication of Equation (3-5b) produces:

$$(V^2_{R_{xy}}) = (V_R)_x^2 + (V_R)_y^2 \quad (3-5c)$$

Hence, the angle, ν , between the "y-z" plane (the plane of the original flight path) and the plane containing the resultant velocity, V_R , can be expressed as:

$$\tan \nu = \frac{(V_R)_x}{(V_R)_y} \quad (3-5d)$$

If we now substitute the magnitude of the components of the resultant velocity along the x and y axes for $(V_R)_x$ and $(V_R)_y$, using Equations (3-3a) and (3-3b), we find that Equation (3-5d) becomes:

$$\tan \nu = \frac{V_D \cos \zeta \sin \epsilon}{V_D \cos \zeta \cos \epsilon + V \cos \phi}$$

(3-5e)

The angle ψ will be referred to throughout this report as the "orbit plane shift angle". It is the angle between the plane of the original flight path and the plane of the resultant velocity.

Now, one other direction angle must be specified, viz., the angle that the resultant velocity makes with the local horizontal.

Substituting Equation (3-5a) in Equation (3-4a), we see that:

$$\vec{V}_R = (\vec{V}_R)_{xy} + (\vec{V}_R)_z \quad (3-6a)$$

Again, by scalar multiplication and since $(V_R)_{xy}$ and $(V_R)_z$ lie perpendicular to each other, we obtain:

$$V_R^2 = (V_R)_{xy}^2 + (V_R)_z^2 \quad (3-6b)$$

Hence, the angle, ψ , that the resultant velocity makes with the local horizontal or "x-y" plane can be expressed as:

$$\tan \psi = \frac{(V_R)_z}{(V_R)_{xy}} \quad (3-6c)$$

Using the scalar magnitude of the component of the resultant velocity along the z axis as given in Equation (3-3c) and the result of Equation (3-5c) in Equation (3-6c) we obtain:

$$\tan \psi = \frac{V_D \sin \zeta + V \sin \phi}{\left[V_D^2 \cos^2 \zeta + 2VV_D \cos \zeta \cos \epsilon \cos \phi + V^2 \cos^2 \phi \right]^{1/2}} \quad (3-6d)$$

The angle Ψ is the new flight angle or re-entry angle of the resultant velocity and will be termed in this report the "resultant flight angle".

Thus, we have defined all necessary factors to determine the resultant velocity, and its direction with respect to the original flight path as a function of the three destruct quantities V_D , ζ , and ϵ and the two initial flight conditions, i.e. the original velocity, V , and original flight angle, ϕ . Hence, if we know values for V_D , ζ , and ϵ from destruct experiments we can determine resultant conditions for any flight trajectory.

The use of Equations (3-4c), (3-5e) and (3-6d) will be demonstrated in Section 5.

4.0 PHASE TWO - CALCULATION OF EARTH RANGE

4.1 EARTH RANGE CONCEPT

There are two factors which determine whether a body will re-enter and impact with the earth's surface on any given pass around the earth. These factors are: (1) the total energy at the beginning of the pass, and (2) the flight path. The former determines the energy available to combat drag losses that tend to dissipate the stability of the trajectory, and the latter determines (for a given initial location in space) the extent to which a body will dip into the earth's atmosphere and, hence, determines the drag losses incurred in any given pass. In general, when the condition arises that the rate of drag loss exceeds or equals the orbiting body's conversion rate of potential to kinetic energy at every point on the trajectory, then the body will re-enter and impact with the earth on that particular orbital passage.

Since we are limiting our concern to re-entry of explosive destruct debris from a 100 nautical mile circular orbit, in one pass or less around the earth from the point of destruct, our task is somewhat eased.

The term "total energy" refers to the sum of the kinetic and potential energy available at the beginning of the orbit or in our context right after destruct has occurred. This total energy is a direct function of particle mass and, hence, size.

Now the drag energy loss rate can be shown to be (starting from Newton's Drag Law)⁽⁵⁾:

$$\frac{d E_D}{dt} = 1/2 C_D \rho V_R^2 A_C \quad (4-1)$$

Equation (4-1) can be rearranged by multiplying and dividing the right-hand side by the body mass, m , into the following form:

$$\frac{d E_D}{dt} = \frac{1}{2} \frac{m \rho V_R^2}{(m/C_D A_C)} \quad (4-2)$$

The quantity, $m/C_D A$, is commonly referred to as the ballistic parameter; hence, the drag energy rate is an inverse function of the ballistic parameter. The drag coefficient, C_D , and the frontal area (drag area) are a function of, among other things, the body shape.

Thus, the fraction of one orbit that destruct debris will navigate before settling to earth is a function of: (1) the body shape and drag area, (2) the mass of the body, (3) the initial resultant velocity of the body and (4) the initial resultant flight angle of the trajectory.

At this juncture, one further simplification in our analysis will be made. Because of the complexity involved in handling tumbling, spinning and like gyrations of most basic particle shapes, only the spherical body geometry will be incorporated in our analysis. The mathematical and physical difficulties involved in the treatment of non-spherical bodies is formidable to say the least and, moreover, for the particle sizes of interest, nothing basic to the analyses to follow is lost.

Since for a sphere the mass is directly proportional to the cube of the diameter of the body and the drag area (frontal area) is a direct function of the square of the diameter, the fractional portion of one orbit that a spherical particle makes from the point of its production during destruct until it settles to earth can be reduced to a consideration of the resultant initial velocity and flight angle and the particle diameter.

Now, referring to Figure 4.1, note that as a destruct particle descends toward the earth along its flight path, we may follow its course using a projection of its flight path on the earth's surface. Then the angle produced by drawing two lines from the center of the earth, one to the point of destruct in orbit and another to the point of impact on the earth's surface, is then a direct measure of the re-entry range of the particle with respect to the earth's surface. This angle, λ , is then a function of the resultant velocity, V_R , and initial flight angle, ψ . Hence, we may write that for a given particle size and initial orbit altitude:

$$\lambda = f(V_R, \psi) \quad (4-3)$$

The problem has now evolved to the point where the trajectory portion of the WEREC-I code can be utilized^(1,5). A parametric study was undertaken to determine λ for spherical particles having ballistic parameters of 0.148 and 0.0755 lb/ft² which correspond to 1000 micron and 100 micron diameter particles. These particle sizes represent, in lieu of experimental information, what was considered to be typical destruct debris sizes and, more important, to represent the range of ingestible particle sizes, which is of critical interest from the standpoint of health physics⁽⁶⁾. The parametric study involved the determination of λ for the two sizes given above from 100 nautical mile altitude, using resultant velocities from 27,000 ft/sec to 24,000 ft/sec and corresponding re-entry angles of +3.5° to -3.5°.*

* The plus sign on re-entry angles corresponds to a positive (up) inclination to the horizontal; minus re-entry angles to a negative re-entry angle.

PICTORIAL REPRESENTATION OF THE FLIGHT PATH OF
A RE-ENTERING PARTICLE

FIGURE 4.1

Since the orbital velocity in a 100 nautical mile orbit is 25,584 ft/sec the velocity range quoted above corresponds to a maximum destruct velocity of approximately 1500ft/sec. The results of this study are given in Figures 4.2 and 4.3 where the earth range is plotted against the initial flight angle (re-entry angle) at constant resultant velocities on a semi-logarithmic grid. Note that when the earth range reaches the value of 360 degrees, there is a cut-off line which denotes the boundary limit of consideration in this study. Furthermore, it should be brought to attention that not all the curves are uniform, particularly those corresponding to higher velocities. As will be noted later, this condition forces a slight compromise in the study.

4.2 ANALYSIS TO DETERMINE FUNCTIONAL RELATIONSHIP BETWEEN λ , Ψ , AND V_R

4.2.1 Importance of Knowing $\lambda = f(V_R, \Psi)$

It was stated earlier, in Section 2.1, that a method would be established to determine the change in earth area contaminated by debris as a function of the change in the experimentally measured destruct parameters. It is obvious that, if the earth range determines an impact locus of a particle, then it must be intimately involved in this calculation. Since the earth range, λ , is a function of V_R and Ψ for a given particle size, then, from the results of Section 3.0, it is also a function of the three measurable destruct variables, V_D, ζ, ϵ . Now, any change in λ must be the result of a change in V_R and/or Ψ . Hence, we may write:

FIGURE 4.2

FIGURE 4.3

$$d\lambda = \left(\frac{\partial \lambda}{\partial V_R} \right)_{\Psi} dV_R + \left(\frac{\partial \lambda}{\partial \Psi} \right)_{V_R} d\Psi, \quad (4-4)$$

or since V_R and Ψ are both functions of V_D , ζ , and ϵ , we can transform (4-4) into:

$$\begin{aligned}
 d\lambda = & \left(\frac{\partial \lambda}{\partial V_R} \right)_{\Psi} \left(\frac{\partial V_R}{\partial V_D} \right) dV_D + \left(\frac{\partial \lambda}{\partial V_R} \right)_{\Psi} \left(\frac{\partial V_R}{\partial \zeta} \right) d\zeta + \left(\frac{\partial \lambda}{\partial V_R} \right)_{\Psi} \left(\frac{\partial V_R}{\partial \epsilon} \right) d\epsilon + \\
 & \left(\frac{\partial \lambda}{\partial \Psi} \right)_{V_R} \left(\frac{\partial \Psi}{\partial V_D} \right) dV_D + \left(\frac{\partial \lambda}{\partial \Psi} \right)_{V_R} \left(\frac{\partial \Psi}{\partial \zeta} \right) d\zeta + \left(\frac{\partial \lambda}{\partial \Psi} \right)_{V_R} \left(\frac{\partial \Psi}{\partial \epsilon} \right) d\epsilon
 \end{aligned} \quad (4-5)$$

Obviously, to even be able to perform the indicated operations shown in Equations (4-4) and (4-5) requires an explicit relationship between λ , V_R , and Ψ .

One has only to glance at the graphs of $\lambda = f(V_R, \Psi)$ to recognize that the development of this relationship is no mean task. One of the more satisfactory mathematical methods by which this may be done is the "method of least squares". Its application to the problem at hand is outlined in the following section.

4.2.2 Method of Least Squares to Determine $\lambda = f(V_R, \Psi)$

If a closed form solution to the functional relationship between the independent and dependent variables neither exists nor is known, then many times a power series can be found that will give satisfactory results throughout the domain of values for the dependent variable. However, the results must be more than just satisfactory, since in this study the polynomial must approximate a closed form solution to the extent that the mathematical operations indicated in the previous sections may be carried out without introducing any appreciable error.

The method of least squares is one of the most powerful tools for use in this type of work and was employed in this investigation. The theory involved in the least-squares polynomial fitting techniques will be deferred to Appendix I, due to the length of the analysis. In this appendix both the basic theory and the application to our particular problem is given in detail.

The wide variations in $\lambda = f(V_R, \Psi)$ curves and the need for near closed form solution accuracy, when compounded with the mathematical manipulations outlined in Equations (4-4) and (4-5) require that a polynomial be found that:

- (1) is not excessively long, and
- (2) provides sufficient accuracy.

The task of finding such a polynomial boils down to a trial and error procedure, and one has only to follow the derivation and use of the least squares technique to see that this is indeed quite a job, even in simple cases. Hence, a digital computer program was prepared to do the "fitting" of the polynomial by determining the coefficients of a polynomial using the method of least squares.

The actual details of this computer program are given in Appendix II, together with a complete listing of the contents of the program and the results from a typical case. For our purposes, at this point, it is sufficient for us to say that the job of empirically determining $\lambda = f(V_R, \Psi)$ has been reduced to reading in a number of data points (i.e., λ, V_R, Ψ) and the terms (functions) making up the polynomial. The computer program (given the name NEFØD) then determines the coefficients for each term of the polynomial and as

explained in Appendix II performs the necessary calculations to determine the degree of accuracy of the polynomial using several different criteria.

4.3 USE OF NEFØD

4.3.1 Limits of Applicability

For the initial studies, it was felt that more success would be obtained in the available time if a concentrated effort were made on fitting as much of the domain lying between $\lambda \leq 360^\circ$ and $-3.5 \leq \Psi \leq +3.5^\circ$ using polynomials in V_R and Ψ only. Unfortunately, certain complications arise here because it was quickly determined that those curves representing velocities in excess of 25,600 ft/sec were going to be difficult to fit using ordinary polynomials in V_R and Ψ only. This difficulty is due to the fact that the curves representing velocities in excess of 25,600 ft/sec are (for the most part) non-uniform throughout all values of V_R and Ψ , and the least squares technique requires uniform conditions. This turn of events then forced us to consider (for the time being) only resultant velocities 25,600 ft/sec or less in magnitude. Now 25,600 ft/sec is only 16 ft/sec larger than the circular orbit velocity at 100 nautical miles. This in turn means that there is a certain maximum destruct velocity for every forward destruct direction (i.e., destruct on the direction of the reactor flight path). This velocity is less than 1500 ft/sec and more than 16 ft/sec. It must not be exceeded at a given ζ for values of ϵ less than 90° and more than 270° , if a correct value of λ is to be determined using the polynomial fit determined by the NEFØD program. More will be said about this in Section 4.4.

4.3.2 Use of NEFØD

Because of the large values of velocity, V_R , used in the computer program there subsequently arose the condition whereby the computational limits of the computer were exceeded when higher degree polynomials were used. This condition necessitated a transformation of the data for the double purpose of achieving more sensitivity in the data and preventing the computational limits of the computer being exceeded.

Data transformation was found to be necessary only for the velocity. To achieve the proper sensitivity, the resultant velocity was transformed into a new resultant velocity, V_R^* , by the following relation:

$$V_R^* = \frac{V_R - 24000}{24000} \quad (4-6)$$

A re-iterative trial and error method was used to obtain a power series polynomial fit. This method consisted of first fitting a polynomial of degree 4 and by using the various error evaluation techniques given in Appendix I, the accuracy of the fit could be ascertained. If the fit was not sufficiently good, then an evaluation of each term for its significance could be carried out (using the "term evaluation" results) and those that were the least significant could be eliminated. Then the next higher power polynomial would be used incorporating all the significant terms left over from the previous calculation.

This procedure thus is a method to obtain a sufficiently accurate fit in a power series of V_R and ψ and at the same time serves to limit the polynomial to some reasonable size. For after all, we could arrive at the situation where we had as many terms in the polynomial as data points if some such devices as outlined above were not employed.

Consider the following illustrative case where a second power polynomial is chosen for a trial fit to a set of data:

$$Y = B_0 + B_1X^2 + B_2Z^2 + B_3ZX + B_4X + B_5Z \quad (4-7)$$

Now suppose that from the least squares fitting routine we find that the polynomial does not give sufficient accuracy and that the third, fourth and fifth terms prove least significant. Then using the next higher power polynomial (third power) and eliminating the terms noted above we have for our next trial polynomial:

$$Y = B_0 + B_1X^3 + B_2Z^3 + B_3X^2Z + B_4XZ^2 + B_5X^2 + B_6Z \quad (4-8)$$

Thus, we continue in this fashion until we obtain a satisfactory fit.

The criterion for significance of a term was arbitrarily chosen such that, if the squared difference between the actual value of λ and the computed value of λ using that one term alone was one thousand times larger than the lowest averaged squared difference obtained for all terms, then that term was insignificant.

The criterion for selection of a polynomial was based on two points:

- (1) The probable error not exceed $2-3^\circ$,
- (2) The average difference not exceed 3° .

Since the validity of this entire analysis of destruct depends, as will become evident, on the accuracy of fitting λ , this step is extremely critical.

4.3.3 Results

In Tables 4.1 and 4.2, the polynomial fits for 100 and 1000 μ particles are given.

TABLE 4.1
POLYNOMIAL COEFFICIENTS FROM NEFØD FOR 100 μ PARTICLE

Ballistic Parameter = 0.0755 lb/ft²

$$\lambda = \sum_{m=1}^M \sum_{n=1}^N \beta_{mn} (\Psi^*)^{m-1} (V_R^*)^{n-1}, \quad \text{where} \quad \begin{cases} \Psi^* = \Psi \\ V_R^* = \frac{V_R - 24,000}{24,000} \end{cases}$$

<u>m</u>	<u>n</u>	<u>β_{mn}</u>	<u>m</u>	<u>n</u>	<u>β_{mn}</u>
1	1	2.6218×10^1	15	1	-2.5777×10^{-4}
17	1	1.0984×10^{-5}	1	15	5.5728×10^{19}
1	17	9.6589×10^{19}	14	2	-8.9358×10^{-5}
16	2	2.8861×10^{-6}	2	14	2.9071×10^{16}
2	16	5.2117×10^{19}	4	12	4.9083×10^{14}
15	3	1.7626×10^{-2}	5	11	-1.2974×10^{12}
3	15	-2.7232×10^{17}	1	14	-2.1764×10^{18}
14	4	2.7331×10^{-1}	13	1	1.5498×10^{-3}
4	14	-1.4829×10^{17}	4	10	-1.7112×10^{12}
13	5	-1.4937×10^2	2	8	1.5250×10^{10}
5	13	-1.6293×10^{14}	1	7	1.1410×10^9
12	6	-4.2985×10^3	6	2	2.3659×10^{-1}
6	12	1.4546×10^{14}	6	1	1.2973×10^{-2}
11	7	6.9527×10^5	4	3	3.9582×10^2
7	11	1.5009×10^{12}	5	1	-1.2566
10	8	2.8572×10^7	4	2	-1.5697×10^1
8	10	-8.9695×10^{10}	4	1	-1.2069×10^{-1}
9	9	-1.5300×10^9	3	2	5.4163×10^1
16	1	-2.1688×10^{-8}	3	1	5.0416
1	16	-3.5660×10^{20}	2	2	2.1549×10^2
15	2	2.8535×10^{-4}	2	1	7.1563
2	15	-3.2175×10^{18}	1	2	2.8275×10^2

TABLE 4.2

POLYNOMIAL COEFFICIENTS FROM NEFØD FOR 1000 µ PARTICLE

Ballistic Parameter = 0.148 lb/ft²

$$\lambda = \sum_{m=1}^M \sum_{n=1}^N \beta_{mn} (\Psi^*)^{m-1} (V_R^*)^{n-1}, \quad \text{where} \quad \begin{cases} \Psi^* &= \Psi \\ V_R^* &= \frac{V_R - 24,000}{24,000} \end{cases}$$

<u>m</u>	<u>n</u>	<u>β_{mn}</u>	<u>m</u>	<u>n</u>	<u>β_{mn}</u>
1	1	2.7414×10^1	15	1	1.3304×10^{-4}
17	1	-6.7395×10^{-6}	1	15	1.4211×10^{19}
1	17	1.2545×10^{19}	14	2	-8.4467×10^{-4}
16	2	1.8971×10^{-4}	2	14	-1.4622×10^{17}
2	16	3.7029×10^{19}	4	12	5.8352×10^{14}
15	3	4.6975×10^{-2}	5	11	1.1604×10^{12}
3	15	-2.4210×10^{18}	1	14	-7.6380×10^{17}
14	4	-5.1082	13	1	-6.1110×10^{-4}
4	14	-1.7189×10^{17}	4	10	-2.2264×10^{12}
13	5	8.5550	2	8	1.7298×10^{10}
5	13	5.8410×10^{15}	1	7	1.0798×10^9
12	6	3.0460×10^4	6	2	1.1755×10^1
6	12	1.5356×10^{14}	6	1	-2.7291×10^{-1}
11	7	-1.3392×10^6	4	3	5.1682×10^2
7	11	-8.2185×10^{12}	5	1	-2.5548×10^{-1}
10	8	-5.3501×10^7	4	2	-9.5461×10^1
8	10	-2.2344×10^{10}	4	1	1.9909
9	9	5.2198×10^9	3	2	4.3997×10^1
16	1	5.0512×10^{-7}	3	1	2.4209
1	16	-2.4059×10^{19}	2	2	3.1400×10^2
15	2	-9.7197×10^{-4}	2	1	4.2848
2	15	7.0273×10^{17}	1	2	3.2123×10^2

4.4 EXCLUSION VELOCITY ANALYSIS

It was pointed out in the foregoing subsections that when the destruct velocity has a component in the forward direction along the flight path then the situation arises whereby the polynomial used to describe the relation between λ , V_R , and ψ is not valid.

It will be recalled that for several good reasons resultant velocities above 25,000 ft/sec were excluded from consideration. This in turn means certain magnitudes of destruct velocities are excluded from consideration.

Since resultant velocities are a function of the original velocity and flight angle and the three destruct quantities, V_D , ζ , and ϵ , we have a means of determining which destruct velocities are excluded as a function of ζ and ϵ , starting from Equation (3-4c) given in Section 3.0. Hence, rearranging Equation (3-4c) we obtain:

$$V^2 + V_D^2 + 2V_D V (\cos \zeta \cos \epsilon \cos \phi + \sin \zeta \sin \phi) - V_R^2 = 0 \quad (4-9)$$

Since we are concerned only with initially circular orbits, the initial flight angle is zero, its sine is zero, and its cosine must be one. Thus Equation (4-9) becomes:

$$V_D^2 + [2V \cos \zeta \cos \epsilon] V_D + C = 0 \quad (4-10)$$

where

$$C = V^2 - V_R^2$$

Now Equation (4-10) is a quadratic and can be solved using the quadratic formula.

We then obtain the solutions to V_D as follows:

$$V_D = \frac{-2V_D \cos\zeta \cos\epsilon \pm \sqrt{4V^2 \cos^2\zeta \cos^2\epsilon - 4(V^2 - V_R^2)}}{2}, \quad (4-11)$$

and factoring out and cancelling the denominator we finally get,

$$V_D = V \cos\zeta \cos\epsilon \pm \sqrt{V_R^2 - V^2 (1 - \cos^2\zeta \cos^2\epsilon)} \quad (4-12)$$

Using Equation (4-12) the value of V_D at any ζ and ϵ can be determined such that the resultant velocity does not exceed a prescribed resultant velocity, V_R , which in this case is 25,600 ft/sec.

A small computer program was written to evaluate the allowed magnitude of V_D as a function of ζ and ϵ with $V_R = 25,600$ and $V = 25,583.9$ ft/sec. The results of this study are shown in Figure 4.4. Note that the destruct velocity value obtained for a given ζ and ϵ cannot be exceeded or otherwise the resultant velocity will exceed the 25,600 ft/sec limit. As pointed out before, limitations on destruct velocity up to magnitudes of 1500 ft/sec have to be considered only for destruct velocities which have a component in the forward direction.

GRAPH OF PERMISSABLE DESTRUCT VELOCITY AS FUNCTION OF ζ, ϵ (FOR FORWARD HALF OF DESTRUCT SPHERE)

FIGURE 4.4

5.0 THE DESTRUCT AREA AND ERROR CALCULATIONS

5.1 CALCULATION OF CONTAMINATION AREA

It can be readily seen from the destruct vector and earth range studies that we can now follow a particle from its production at destruct until it impacts with earth and ascribe to it an arbitrary impact locus. (It must be remembered that we have made no attempt to base our analysis on locating the particle with respect to latitude and longitude on earth.) Moreover, it is also obvious that mathematically there are an infinite number of different loci that can be obtained for the same size particle, each locus differing from the others only because their destruct variables had different values.

It can be seen readily that we could describe a segment of the earth's surface area using a minimum of three impact loci as terminals between lines drawn along the surface of the earth to generate a spherical triangle. It is clear then that this surface area is a function of the destruct variables: ζ , ϵ , and V_D since the loci depend on the earth range, which in turn depends on the destruct variables as shown in Equation (4-5).

Hence, by arbitrarily selecting three different combinations of destruct variables (ζ, ϵ, V_D) we can generate a spherical triangle on the earth's surface. If the destruct explosive action is isotropic, then we can consider that the various destruct variables coincide with points on a "destruct sphere" generated by the debris. As long as all three arbitrarily selected points on the destruct sphere do not have the same ζ 's, ϵ 's, and V_D 's then the particles emanating from those points on the destruct sphere will re-enter along separate paths to form the points of the triangle on the earth's surface.

Now referring to Figure 5.1, we see a pictorial representation of a body undergoing isotropic destruct to produce a destruct pattern in the general shape of a sphere*. Now pick three arbitrary points on the "sphere" and let these points represent particles moving away from the center of the destruct along lines A, B, and C connecting them to the center of the destruct. Then using the same co-ordinate system employed in the destruct vector study to locate each point on the destruct sphere (or pattern), we can now describe three re-entry ranges, one for each point. Since each destruct point will produce a unique λ either with respect to shift from the original flight path or with respect to magnitude or both, they will generate a spherical triangle on the surface of the earth upon termination of re-entry.

Furthermore, because the λ 's are unique, any and all points, i.e., debris, emanating from the destruct sphere (pattern) that fall entirely within the "pyramid" of which the triangle of points on the destruct sphere is the base, will also impact on the earth inside the triangular area described by the impact loci of these same three base points on the destruct sphere provided that:

- (1) The particles are the same size (or have roughly the same ballistic parameter) and,
- (2) That the components of the destruct velocity of any particle both along the line of the original flight path and at right angles to the flight path, are neither greater nor less than those corresponding to any of the destruct velocities associated with the three base points.

* It should be pointed out that the analysis to follow is not restricted to a particular destruct geometry. It may be a cylinder, cube or any other arbitrary form.

ORIGINAL FLIGHT
PATH OF BODY

Now in Figure 5.2 there is shown a pictorial representation of the contamination area on the earth from the re-entry of the three "base points" particles on the destruct sphere. Note that each impact point is joined to the other two by arcs of great circles making the inscribed area a spherical triangle. The destruct re-entry ranges are labeled λ_a , λ_b and λ_c and correspond to impact loci subtended by angles λ_a , λ_b and λ_c , respectively. The arcs labeled a , b and c lie opposite to the angles A , B and C and are arcs of great circles which are subtended by the central earth angles a , b and c . The angles ν_a , ν_b and ν_c are the orbit plane shift angles described in Section 3.0, and correspond to λ_a , λ_b and λ_c , respectively. The shaded area shown between two of the projected re-entry paths lies on the surface of the earth and consequently, it too, is a spherical triangle. There are three such "projected" spherical triangles one corresponding to each side of the contamination area triangle shown in cross-hatching.

The area, $\bar{A}$, of the contaminated triangle is given by the expression:

$$\bar{A} = (A + B + C - \pi) R^2 \quad (5-1)$$

where A , B and C are the indicated angles and R is the earth radius. The method of determination of the angles A , B and C is given in detail in Appendices III and IV.

5.2 METHOD FOR DETERMINING THE ERROR IN THE EARTH AREA AS FUNCTION OF ERROR IN MEASURING DESTRUCT VARIABLES

From Equation (5-1) we can write the following relation for an incremental change in the contamination area:

PICTORIAL RELATIONSHIP OF VARIABLES IN THE EARTHS-SURFACE-AREA ERROR ANALYSIS

FIGURE 5.2

$$d\bar{A} = \left(\frac{\partial \bar{A}}{\partial A} \right) dA + \left(\frac{\partial \bar{A}}{\partial B} \right) dB + \left(\frac{\partial \bar{A}}{\partial C} \right) dC \quad (5-2)$$

Similarly, since the angles A , B and C depend on the angles a , b and c (see Appendix IV)

we can then express dA as a function of the angles a , b and c . Now, the angles a , b

and c are in turn functions of the λ and ν for each of the three destructure sphere points.

It will be recalled that λ is a function of V_R and Ψ from the polynomial fitting procedure,

and in turn V_R and Ψ are functions of ζ , ϵ , and V_D as given in Equations (3-4c) and

(3-6d). From Equation (3-5e) we see that ν is likewise a function of the destructure variables.

By following this chain of reasoning we can express the change in the area of contamination

on the earth's surface as a function of the deviation increment, or if you wish, the error in

measuring the destructure variables. Therefore, for one point on the destructure sphere, we may

write that the change in the earth area as function of change in the value of the destructure

variables at the point is:

$$d\bar{A}_1 = X_1 dV_{D1} + Y_1 d\epsilon_1 + Z_1 d\zeta_1 \quad (5-3)$$

where X_1 , Y_1 and Z_1 are:

$$\left(\frac{\partial \bar{A}}{\partial V_D} \right)_1, \quad \left(\frac{\partial \bar{A}}{\partial \epsilon} \right)_1, \quad \left(\frac{\partial \bar{A}}{\partial \zeta} \right)_1,$$

respectively. The three partial derivatives are actually mathematical combinations of the

functions of λ , ν , etc., as outlined above. The subscript 1, refers to the fact that

Equation (5-3) as written is valid for only one point on the destructure sphere. However, if we

make the rather reasonable assumption that the precision in measuring V_D , ϵ and ζ is

not a function of orientation, then for the three destruct points we may write:

$$\left. \begin{aligned} dV_{D1} &= dV_{D2} = dV_{D3} \\ d\zeta_1 &= d\zeta_2 = d\zeta_3 \\ d\epsilon_1 &= d\epsilon_2 = d\epsilon_3 \end{aligned} \right\} \quad (5-4)$$

Hence,

$$d\bar{A} = d\bar{A}_1 + d\bar{A}_2 + d\bar{A}_3 \quad (5-5)$$

and

$$dA = X dV_D + Y d\epsilon + Z d\zeta \quad , \quad (5-6)$$

where

$$\left. \begin{aligned} X &= X_1 + X_2 + X_3 \\ Y &= Y_1 + Y_2 + Y_3 \\ Z &= Z_1 + Z_2 + Z_3 \end{aligned} \right\} \quad (5-7)$$

The Equation (5-6) provides the expression which we sought at the beginning of this study. The equation is perfectly general and is not dependent on any particular explosive pattern although we have for purposes of illustration used the term "destruct sphere". But, it should be pointed out that the explosive pattern could be cylindrical, cubic, conical or any other shape; the analysis is still valid.

It is now obvious why the accuracy in the determination of the $\lambda = f(V_R, \Psi)$ polynomial is critical, since we mathematically manipulate the chosen polynomial just as if it

were a closed-form solution. Providing we have sufficient accuracy, this is perfectly allowable just as is the case of a power series solution to a differential equation (e.g., Bessel Functions) or a Taylor series expansion.

Because of the tediousness of the calculations, not to mention the time required to compute just one set of destruct area error coefficients (that is to say, X, Y and Z) it was necessary to prepare a program for use on the IBM-7094 digital computer. The general outline and salient features of this program are explained in the next subsection.

5.3 DIGITAL COMPUTER PROGRAM FOR DETERMINING THE EARTH AREA ERROR COEFFICIENTS

The digital computer program for determining the earth area error coefficients and the area contaminated by the debris consists of a MAIN Program and 10 Sub-programs. The entire program is given the name NURØDAC, for Nuclear Rocket Orbital Destruct Area Computation.

The original version of NURØDAC was written in Fortran II language, but after a most trying and difficult "debugging" operation it was determined that due to particular and unusual demands for accuracy that double-precision calculations would be required, especially in the earth area of contamination analysis.* This necessitated that the program be re-written, this time using the newer programming system Fortran IV, wherein advantage could be taken of the built-in double-precision capability of the machine. This transformation led, naturally, to a new set of difficulties in debugging the program. As a

* Single precision in the 7094 is eight significant digits; double precision is 16 significant digits.

consequence, a simultaneous effort was begun to program the analysis in Fortran II using double precision arithmetic. The difficulty here is that to incorporate double-precision type calculations is quite tedious.

However, it was felt that the time used and the effort required would be more than offset by the fact that one could obtain an independent check on the results in either system. As a result, two double precision programs were developed, one in Fortran II and the other in Fortran IV which are referred to as NURØDAC-II and NURØDAC-IV, respectively. But more important, each program gives the same results as the other. In Appendix V a listing is given for the Fortran II version of the program. Both versions have the same basic computational steps and have the same input-output routines; hence as far as the user is concerned the two are mathematically equivalent and differ only in internal workings of the computer and computer time required.*

The input required to the computer has been reduced to a minimum and consists of a Title Card, Cards designating coefficients and powers of V_R and Ψ for each term of the polynomial representing λ , a control card for use in incrementing data points (this will be explained in more detail in Appendix V) and the three destruct data variables for each of the three points on the destruct sphere.

The output consists of:

- (1) The three destruct points described by ζ , ϵ and V_D .

* The Fortran IV version requires less running time than the Fortran II Program.

- (2) The destruct area error coefficients X , Y and Z .
- (3) The earth range of debris from point of destruct until impact, in miles.
- (4) The orbit plane shift angle, ψ , in degrees.
- (5) The per cent error in the magnitude of the area of contamination due to one degree error in ζ and ϵ and a 100 ft/sec error in V_D .

6.0 TYPICAL EXAMPLES OF THE USE OF NURØDAC

With the use of the NURØDAC and NEFØD computer programs, six samples cases were examined. These cases, which consider destruct in a 100 NM orbit for particles having a 1000 μ diameter, are given in Tables 6.1 through 6.6, and the destruct triangles chosen for consideration are shown in Figure 6.1.

In order to understand the usage of these tables, one typical destruct triangle will be considered in detail, i.e., destruct triangle 1. For example, examine the following simple calculations:

Example 1

Choose $\Delta V_D = 100 \text{ ft/sec}$, $\Delta \epsilon = \Delta \zeta = 0 \text{ deg}$.

Then for $V_D = 100 \text{ ft/sec}$

$$\begin{aligned}
 \Delta \text{Area} &= (1.06720 \times 10^1) \frac{\text{sq mi}}{\text{ft/sec}} \times 100 \text{ ft/sec} \\
 &\quad + (1.67619 \times 10^1) \frac{\text{sq mi}}{\text{deg}} \times 0 \text{ deg} \\
 &\quad + (1.15513 \times 10^{-1}) \frac{\text{sq mi}}{\text{deg}} \times 0 \text{ deg}
 \end{aligned}$$

$$\Delta \text{Area} = 1.06720 \times 10^3 \text{ sq mi.}$$

Therefore, since the area of the exactly defined triangle = $4.74705 \times 10^2 \text{ sq mi}$, the percentage error introduced in that area by an error in measurement of the velocity of destruct of $\pm 100 \text{ ft/sec} \equiv$

$$\Delta E \equiv \frac{\Delta A}{A} = \frac{1.06720 \times 10^3}{474.705} \times 100\% = 225\%$$

TABLE 6.1

* 1ST. CASE RESULTS FOR VARIOUS DEST. VEL. IN DESTRUCT TRIANGLE NO. 1 *						

DESTRUCT VARIABLES VALUE			DESTRUCT AREA ERROR COEFFICIENTS			DESTRUCT AREA
*****			*****			*****
ZETA (DEG.)	EPSILON (DEG.)	DEST.VEL. (FT./SEC)	DEL DST VEL * SQ MI/(FT/SEC)*	DEL EPSILON * (SQ MI/DEG) *	DEL ZETA (SQ MI/DEG)	(SQ.MILES)
30.0000	120.0000	100.0000	*	*	*	*
45.0000	135.0000	100.0000	1.06720E 01*	1.67619E 01*	1.15513E-01*	4.74705E 02
60.0000	150.0000	100.0000	*	*	*	*

DESTRUCT RANGES * FLIGHT PATH SHIFT*						
(MILES)	*	(DEGREES)	PER CENT DEST			
1.04523E 04	*	1.68249E-01	AREA CHANGE = 2.28369E 02 (BASED ON ONE DEGREE ERROR			
1.04450E 04	*	1.12196E-01	IN ZETA AND EPSILON,			
1.09488E 04	*	5.60830E-02	100 FT/SEC IN DEST VEL.)			

DESTRUCT VARIABLES VALUE			DESTRUCT AREA ERROR COEFFICIENTS			DESTRUCT AREA
*****			*****			*****
ZETA (DEG.)	EPSILON (DEG.)	DEST.VEL. (FT./SEC)	DEL DST VEL * SQ MI/(FT/SEC)*	DEL EPSILON * (SQ MI/DEG) *	DEL ZETA (SQ MI/DEG)	(SQ.MILES)
30.0000	120.0000	400.0000	*	*	*	*
45.0000	135.0000	400.0000	2.11290E 01*	4.46905E 01*	1.02452E 02*	6.22162E 03
60.0000	150.0000	400.0000	*	*	*	*

DESTRUCT RANGES * FLIGHT PATH SHIFT*						
(MILES)	*	(DEGREES)	PER CENT DEST			
7.99750E 03	*	6.76437E-01	AREA CHANGE = 3.63256E 01 (BASED ON ONE DEGREE ERROR			
7.99960E 03	*	4.51434E-01	IN ZETA AND EPSILON,			
8.88625E 03	*	2.25479E-01	100 FT/SEC IN DEST VEL.)			

TABLE 6.1 (Cont'd)

```
*****
*
* 1ST. CASE RESULTS FOR VARIOUS DEST. VEL. IN DESTRUCT TRIANGLE NO. 1
*
*****
```

DESTRUCT VARIABLES VALUE			DESTRUCT AREA ERROR COEFFICIENTS			DESTRUCT AREA
ZETA (DEG.)	EPSILON (DEG.)	DEST.VEL. (FT./SEC)	DEL DST VEL * SQ MI/(FT/SEC)*	DEL EPSILON (SQ MI/DEG) *	DEL ZETA (SQ MI/DEG)	(SQ.MILES)
30.0000	120.0000	700.0000				
45.0000	135.0000	700.0000	2.31230E 01*	-8.02437E 00*	3.34773E 02*	1.28579E 04
60.0000	150.0000	700.0000				

```

*****
DESTRUCT RANGES * FLIGHT PATH SHIFT*
(MILES) * (DEGREES) * PER CENT DEST
6.87326E 03 * 1.18985E 00 * AREA CHANGE = 2.05247E 01 (BASED ON ONE DEGREE ERROR
7.02903E 03 * 7.94706E-01 * IN ZETA AND EPSILON,
8.11780E 03 * 3.96616E-01 * 100 FT/SEC IN DEST VEL.)
*****

```

DESTRUCT VARIABLES VALUE			DESTRUCT AREA ERROR COEFFICIENTS			DESTRUCT AREA	
ZETA (DEG.)	EPSILON (DEG.)	DEST.VEL. (FT./SEC)	DEL DST VEL * SQ MI/(FT/SEC)*	DEL EPSILON (SQ MI/DEG) *	DEL ZETA (SQ MI/DEG)	(SQ.MILES)	
30.0000	120.0000	1000.0000					
45.0000	135.0000	1000.0000	1.27743E 01	-1.94748E 02	7.20580E 02	1.88374E 04	
60.0000	150.0000	1000.0000					

```

*****
DESTRUCT RANGES * FLIGHT PATH SHIFT*
(MILES) * (DEGREES) * PER CENT DEST
6.28672E 03 * 1.70856E 00 * AREA CHANGE = 9.57280E 00 (BASED ON ONE DEGREE ERROR
6.58198E 03 * 1.14208E 00 * IN ZETA AND EPSILON,
7.80887E 03 * 5.69520E-01 * 100 FT/SEC IN DEST VEL.)
*****

```

TABLE 6.2

* 2ND CASE RESULTS FOR VARIOUS DEST. VEL. IN DESTRUCT TRIANGLE NO. 2 *						

DESTRUCT VARIABLES VALUE			DESTRUCT AREA ERROR COEFFICIENTS			DESTRUCT AREA
*****			*****			*****
ZETA	EPSILON	DEST. VEL.	DEL DST VEL	DEL EPSILON	DEL ZETA	(SQ. MILES)
(DEG.)	(DEG.)	(FT./SEC)	SQ MI/(FT/SEC)	(SQ MI/DEG)	(SQ MI/DEG)	
20.0000	210.0000	100.0000	*	*	*	*
35.0000	235.0000	100.0000	6.45759E 01	-1.88542E 02	-1.73159E 02	2.92371E 03
70.0000	250.0000	100.0000	*	*	*	*

DESTRUCT RANGES * FLIGHT PATH SHIFT*						
(MILES)		(DEGREES)	PER CENT DEST			
8.78035E 03	*	-1.05559E-01	AREA CHANGE = 2.08499E 02 (BASED ON ONE DEGREE ERROR			
1.03948E 04	*	-1.50551E-01	IN ZETA AND EPSILON,			
1.26817E 04	*	-7.20099E-02	100 FT/SEC IN DEST VEL.)			

DESTRUCT VARIABLES VALUE			DESTRUCT AREA ERROR COEFFICIENTS			DESTRUCT AREA
*****			*****			*****
ZETA	EPSILON	DEST. VEL.	DEL DST VEL	DEL EPSILON	DEL ZETA	(SQ. MILES)
(DEG.)	(DEG.)	(FT./SEC)	SQ MI/(FT/SEC)	(SQ MI/DEG)	(SQ MI/DEG)	
20.0000	210.0000	400.0000	*	*	*	*
35.0000	235.0000	400.0000	1.74379E 02	-2.16557E 03	-2.04761E 03	4.42533E 04
70.0000	250.0000	400.0000	*	*	*	*

DESTRUCT RANGES * FLIGHT PATH SHIFT*						
(MILES)		(DEGREES)	PER CENT DEST			
5.31528E 03	*	-4.26317E-01	AREA CHANGE = 2.98840E 01 (BASED ON ONE DEGREE ERROR			
7.89661E 03	*	-6.05546E-01	IN ZETA AND EPSILON,			
1.25749E 04	*	-2.88435E-01	100 FT/SEC IN DEST VEL.)

TABLE 6.2 (Cont'd)

* 2ND CASE RESULTS FOR VARIOUS DEST. VEL. IN DESTRUCT TRIANGLE NO. 2 *						

DESTRUCT	VARIABLES	VALUE	DESTRUCT AREA ERROR COEFFICIENTS			DESTRUCT AREA

ZETA	EPSILON	DEST. VEL.	DEL DST VEL	DEL EPSILON	DEL ZETA	(SQ. MILES)
(DEG.)	(DEG.)	(FT./SEC)	SQ MI/(FT/SEC)	(SQ MI/DEG)	(SQ MI/DEG)	
20.0000	210.0000	700.0000				
35.0000	235.0000	700.0000	1.77373E 02	-2.36859E 03	-2.55492E 03	9.69615E 04
70.0000	250.0000	700.0000				

DESTRUCT	RANGES	FLIGHT	PATH	SHIFT		
(MILES)		(DEGREES)				
4.13737E 03		-7.53337E-01			(BASED ON ONE DEGREE ERROR	
6.81941E 03		-1.06562E 00			IN ZETA AND EPSILON,	
1.14843E 04		-5.05456E-01			100 FT/SEC IN DEST VEL.)	

DESTRUCT	VARIABLES	VALUE	DESTRUCT AREA ERROR COEFFICIENTS			DESTRUCT AREA

ZETA	EPSILON	DEST. VEL.	DEL DST VEL	DEL EPSILON	DEL ZETA	(SQ. MILES)
(DEG.)	(DEG.)	(FT./SEC)	SQ MI/(FT/SEC)	(SQ MI/DEG)	(SQ MI/DEG)	
20.0000	210.0000	1000.0000				
35.0000	235.0000	1000.0000	1.66364E 02	-3.03221E 03	-3.59941E 03	1.50270E 05
70.0000	250.0000	1000.0000				

DESTRUCT	RANGES	FLIGHT	PATH	SHIFT		
(MILES)		(DEGREES)				
3.66242E 03		-1.08680E 00			(BASED ON ONE DEGREE ERROR	
6.27773E 03		-1.53086E 00			IN ZETA AND EPSILON,	
1.12464E 04		-7.23076E-01			100 FT/SEC IN DEST VEL.)	

TABLE 6.3

* 3RD CASE RESULTS FOR VARIOUS DEST. VEL. IN DESTRUCT TRIANGLE NO. 3 *						

DESTRUCT VARIABLES VALUE			DESTRUCT AREA ERROR COEFFICIENTS			DESTRUCT AREA
*****			*****			*****
ZETA (DEG.)	EPSILON (DEG.)	DEST.VEL. (FT./SEC)	DEL DST VEL * SQ MI/(FT/SEC) *	DEL EPSILON * (SQ MI/DEG) *	DEL ZETA * (SQ MI/DEG) *	(SQ.MILES)
-30.0000	120.0000	100.0000	*	*	*	*
-45.0000	135.0000	100.0000	* 1.85335E 01*	-1.46031E 01*	8.96413E 00*	9.40764E 02
-60.0000	150.0000	100.0000	*	*	*	*

DESTRUCT RANGES * FLIGHT PATH SHIFT*						
(MILES)	*	(DEGREES)	* PER CENT DEST			
8.84750E 03	*	1.68249E-01	* AREA CHANGE = 1.96405E 02 (BASED ON ONE DEGREE ERROR			
8.25225E 03	*	1.12196E-01	IN ZETA AND EPSILON,			
8.20439E 03	*	5.60830E-02	100 FT/SEC IN DEST VEL.)			

DESTRUCT VARIABLES VALUE			DESTRUCT AREA ERROR COEFFICIENTS			DESTRUCT AREA
*****			*****			*****
ZETA (DEG.)	EPSILON (DEG.)	DEST.VEL. (FT./SEC)	DEL DST VEL * SQ MI/(FT/SEC) *	DEL EPSILON * (SQ MI/DEG) *	DEL ZETA * (SQ MI/DEG) *	(SQ.MILES)
-30.0000	120.0000	400.0000	*	*	*	*
-45.0000	135.0000	400.0000	* -1.90276E 00*	-1.26567E 02*	1.91274E 01*	3.08391E 03
-60.0000	150.0000	400.0000	*	*	*	*

DESTRUCT RANGES * FLIGHT PATH SHIFT*						
(MILES)	*	(DEGREES)	* PER CENT DEST			
4.30121E 03	*	6.76437E-01	* AREA CHANGE = -9.65385E 00 (BASED ON ONE DEGREE ERROR			
3.47803E 03	*	4.51434E-01	IN ZETA AND EPSILON,			
3.24221E 03	*	2.25479E-01	100 FT/SEC IN DEST VEL.)			

TABLE 6.3 (Cont'd)

 *
 * 3RD CASE RESULTS FOR VARIOUS DEST. VEL. IN DESTRUCT TRIANGLE NO. 3 *
 *

DESTRUCT VARIABLES VALUE			DESTRUCT AREA ERROR COEFFICIENTS			DESTRUCT AREA
ZETA (DEG.)	EPSILON (DEG.)	DEST.VEL. (FT./SEC)	DEL DST VEL * SQ MI/(FT/SEC)*	DEL EPSILON (SQ MI/DEG)	DEL ZETA (SQ MI/DEG)	(SQ.MILES)
-30.0000	120.0000	700.0000	*	*	*	*
-45.0000	135.0000	700.0000	-2.43714E 00*	-9.67969E 01*	2.50708E 01*	2.55261E 03
-60.0000	150.0000	700.0000	*	*	*	*

DESTRUCT RANGES (MILES)	FLIGHT PATH SHIFT (DEGREES)	PER CENT DEST AREA CHANGE	DESTRUCT AREA ERROR COEFFICIENTS
2.78452E 03	1.18985E 00	-1.23576E 01	(BASED ON ONE DEGREE ERROR IN ZETA AND EPSILON, 100 FT/SEC IN DEST VEL.)
2.29550E 03	7.94706E-01	*	
2.18490E 03	3.96616E-01	*	

DESTRUCT VARIABLES VALUE			DESTRUCT AREA ERROR COEFFICIENTS			DESTRUCT AREA
ZETA (DEG.)	EPSILON (DEG.)	DEST.VEL. (FT./SEC)	DEL DST VEL * SQ MI/(FT/SEC)*	DEL EPSILON (SQ MI/DEG)	DEL ZETA (SQ MI/DEG)	(SQ.MILES)
-30.0000	120.0000	1000.0000	*	*	*	*
-45.0000	135.0000	1000.0000	-9.24727E 00*	-9.43493E 01*	-2.50312E 01*	1.65384E 03
-60.0000	150.0000	1000.0000	*	*	*	*

DESTRUCT RANGES (MILES)	FLIGHT PATH SHIFT (DEGREES)	PER CENT DEST AREA CHANGE	DESTRUCT AREA ERROR COEFFICIENTS
2.18912E 03	1.70856E 00	-6.31323E 01	(BASED ON ONE DEGREE ERROR IN ZETA AND EPSILON, 100 FT/SEC IN DEST VEL.)
1.85620E 03	1.14208E 00	*	
1.74396E 03	5.69520E-01	*	

TABLE 6.4

 *
 * 4TH CASE RESULTS FOR VARIOUS DEST. VEL. IN DESTRUCT TRIANGLE NO. 4 *
 *

DESTRUCT VARIABLES VALUE			DESTRUCT AREA ERROR COEFFICIENTS			DESTRUCT AREA
ZETA (DEG.)	EPSILON (DEG.)	DEST.VEL. (FT./SEC)	DEL DST VEL SQ MI/(FT/SEC)	DEL EPSILON (SQ MI/DEG)	DEL ZETA (SQ MI/DEG)	(SQ.MILES)
-20.0000	210.0000	100.0000				
-35.0000	235.0000	100.0000	4.41281E 01	-6.15195E 00	-3.40338E 01	2.49272E 03
-70.0000	250.0000	100.0000				

 DESTRUCT RANGES * FLIGHT PATH SHIFT *
 (MILES) * (DEGREES) * PER CENT DEST
 7.83863E 03 * -1.05559E-01 * AREA CHANGE = 1.75416E 02 (BASED ON ONE DEGREE ERROR
 8.58577E 03 * -1.50551E-01 * IN ZETA AND EPSILON,
 9.28368E 03 * -7.20099E-02 * 100 FT/SEC IN DEST VEL.)

DESTRUCT VARIABLES VALUE			DESTRUCT AREA ERROR COEFFICIENTS			DESTRUCT AREA
ZETA (DEG.)	EPSILON (DEG.)	DEST.VEL. (FT./SEC)	DEL DST VEL SQ MI/(FT/SEC)	DEL EPSILON (SQ MI/DEG)	DEL ZETA (SQ MI/DEG)	(SQ.MILES)
-20.0000	210.0000	400.0000				
-35.0000	235.0000	400.0000	-1.14176E 01	1.65608E 02	-2.06735E 02	3.30797E 03
-70.0000	250.0000	400.0000				

 DESTRUCT RANGES * FLIGHT PATH SHIFT *
 (MILES) * (DEGREES) * PER CENT DEST
 3.68527E 03 * -4.26317E-01 * AREA CHANGE = -3.57588E 01 (BASED ON ONE DEGREE ERROR
 3.93379E 03 * -6.05546E-01 * IN ZETA AND EPSILON,
 4.12534E 03 * -2.88435E-01 * 100 FT/SEC IN DEST VEL.)

TABLE 6.4 (Cont'd)

* 4TH CASE RESULTS FOR VARIOUS DEST. VEL. IN DESTRUCT TRIANGLE NO. 4 *						

DESTRUCT VARIABLES VALUE			DESTRUCT AREA ERROR COEFFICIENTS			DESTRUCT AREA
*****			*****			*****
ZETA (DEG.)	EPSILON (DEG.)	DEST.VEL. (FT./SEC)	DEL DST VEL SQ MI/(FT/SEC)	DEL EPSILON (SQ MI/DEG)	DEL ZETA (SQ MI/DEG)	(SQ.MILES)
-20.0000	210.0000	700.0000	*	*	*	*
-35.0000	235.0000	700.0000	1.27738E 01	1.97933E 02	-3.53314E 02	6.22987E 03
-70.0000	250.0000	700.0000	*	*	*	*

DESTRUCT RANGES * FLIGHT PATH SHIFT*						
(MILES)	*	(DEGREES)	* PER CENT DEST			
2.53059E 03	*	-7.53337E-01	* AREA CHANGE = 1.80100E 01 (BASED ON ONE DEGREE ERROR			
2.55671E 03	*	-1.06562E 00	IN ZETA AND EPSILON,			
3.45939E 03	*	-5.05456E-01	100 FT/SEC IN DEST VEL.)			

DESTRUCT VARIABLES VALUE			DESTRUCT AREA ERROR COEFFICIENTS			DESTRUCT AREA
*****			*****			*****
ZETA (DEG.)	EPSILON (DEG.)	DEST.VEL. (FT./SEC)	DEL DST VEL SQ MI/(FT/SEC)	DEL EPSILON (SQ MI/DEG)	DEL ZETA (SQ MI/DEG)	(SQ.MILES)
-20.0000	210.0000	1000.0000	*	*	*	*
-35.0000	235.0000	1000.0000	1.23713E-01	1.40467E 02	-2.68155E 02	4.92787E 03
-70.0000	250.0000	1000.0000	*	*	*	*

DESTRUCT RANGES * FLIGHT PATH SHIFT*						
(MILES)	*	(DEGREES)	* PER CENT DEST			
2.05129E 03	*	-1.08680E 00	* AREA CHANGE = -2.34009E 00 (BASED ON ONE DEGREE ERROR			
2.03675E 03	*	-1.53086E 00	IN ZETA AND EPSILON,			
2.71793E 03	*	-7.23076E-01	100 FT/SEC IN DEST VEL.)			

TABLE 6.5

```

*****
*
* 5TH CASE RESULTS FOR VARIOUS DEST. VEL. IN DESTRUCT TRIANGLE NO. 5
*
*****

```

```

*****
DESTRUCT VARIABLES VALUE      * DESTRUCT AREA ERROR COEFFICIENTS * DESTRUCT AREA
*****                      *                      * *****
*                               *                               *
* ZETA   EPSILON  DEST. VEL. * DEL DST VEL * DEL EPSILON * DEL ZETA * (SQ.MILES)
* (DEG.) (DEG.)  (FT./SEC) * SQ MI/(FT/SEC) * (SQ MI/DEG) * (SQ MI/DEG) *
*                               *                               *
10.0000  180.0000  100.0000 *           *           *           *
*                               *           *           *
-5.0000  170.0000  100.0000 * 1.54461E 01* 1.36915E-07* 2.86900E 01* 8.10056E 02
*                               *           *           *
-5.0000  190.0000  100.0000 *           *           *           *
*                               *           *           *
*****
DESTRUCT RANGES * FLIGHT PATH SHIFT*
(MILES)         * (DEGREES)         * PER CENT DEST
8.04230E 03    * 2.33041E-09    * AREA CHANGE = 1.94221E 02 (BASED ON ONE DEGREE ERROR
7.71807E 03    * 3.88901E-02    *                               IN ZETA AND EPSILON,
7.71807E 03    * -3.88901E-02   *                               100 FT/SEC IN DEST VEL.)
*****

```

```

*****
DESTRUCT VARIABLES VALUE      * DESTRUCT AREA ERROR COEFFICIENTS * DESTRUCT AREA
*****                      *                      * *****
*                               *                               *
* ZETA   EPSILON  DEST. VEL. * DEL DST VEL * DEL EPSILON * DEL ZETA * (SQ.MILES)
* (DEG.) (DEG.)  (FT./SEC) * SQ MI/(FT/SEC) * (SQ MI/DEG) * (SQ MI/DEG) *
*                               *                               *
10.0000  180.0000  400.0000 *           *           *           *
*                               *           *           *
-5.0000  170.0000  400.0000 * 7.50781E 00* -2.18032E-07* 1.45939E 02* 4.67145E 03
*                               *           *           *
-5.0000  190.0000  400.0000 *           *           *           *
*                               *           *           *
*****
DESTRUCT RANGES * FLIGHT PATH SHIFT*
(MILES)         * (DEGREES)         * PER CENT DEST
4.36922E 03    * 9.43097E-09    * AREA CHANGE = 1.91957E 01 (BASED ON ONE DEGREE ERROR
3.85009E 03    * 1.57378E-01    *                               IN ZETA AND EPSILON,
3.85009E 03    * -1.57378E-01   *                               100 FT/SEC IN DEST VEL.)
*****

```


TABLE 6.5 (Cont'd)

* 5TH CASE RESULTS FOR VARIOUS DEST. VEL. IN DESTRUCT TRIANGLE NO. 5 *						

DESTRUCT VARIABLES VALUE			DESTRUCT AREA ERROR COEFFICIENTS			DESTRUCT AREA
*****			*****			*****
ZETA	EPSILON	DEST.VEL.	DEL DST VEL	DEL EPSILON	DEL ZETA	(SQ.MILES)
(DEG.)	(DEG.)	(FT./SEC)	SQ MI/(FT/SEC)	(SQ MI/DEG)	(SQ MI/DEG)	
10.0000	180.0000	700.0000				
-5.0000	170.0000	700.0000	5.92497E 00	-3.62093E-07	2.37565E 02	6.46789E 03
-5.0000	190.0000	700.0000				

DESTRUCT RANGES * FLIGHT PATH SHIFT*						
(MILES)		(DEGREES)	* PER CENT DEST			
3.27945E 03	*	1.67001E-08	AREA CHANGE = 1.28336E 01 (BASED ON ONE DEGREE ERROR			
2.75650E 03	*	2.78667E-01	IN ZETA AND EPSILON,			
2.75650E 03	*	-2.78667E-01	100 FT/SEC IN DEST VEL.)			

DESTRUCT VARIABLES VALUE			DESTRUCT AREA ERROR COEFFICIENTS			DESTRUCT AREA
*****			*****			*****
ZETA	EPSILON	DEST.VEL.	DEL DST VEL	DEL EPSILON	DEL ZETA	(SQ.MILES)
(DEG.)	(DEG.)	(FT./SEC)	SQ MI/(FT/SEC)	(SQ MI/DEG)	(SQ MI/DEG)	
10.0000	180.0000	1000.0000				
-5.0000	170.0000	1000.0000	5.32241E 00	-5.00513E-07	3.69966E 02	8.27489E 03
-5.0000	190.0000	1000.0000				

DESTRUCT RANGES * FLIGHT PATH SHIFT*						
(MILES)		(DEGREES)	* PER CENT DEST			
2.84783E 03	*	2.41438E-08	AREA CHANGE = 1.09030E 01 (BASED ON ONE DEGREE ERROR			
2.30892E 03	*	4.02858E-01	IN ZETA AND EPSILON,			
2.30892E 03	*	-4.02858E-01	100 FT/SEC IN DEST VEL.)			

TABLE 6.6

 *
 * 6TH CASE RESULTS FOR VARIOUS DEST. VEL. IN DESTRUCT TRIANGLE NO. 6 *
 *

DESTRUCT VARIABLES VALUE			DESTRUCT AREA ERROR COEFFICIENTS			DESTRUCT AREA
ZETA (DEG.)	EPSILON (DEG.)	DEST.VEL. (FT./SEC)	DEL DST VEL * SQ MI/(FT/SEC)*	DEL EPSILON * (SQ MI/DEG) *	DEL ZETA * (SQ MI/DEG) *	(SQ.MILES)
60.0000	60.0000	10.0000	*	*	*	*
45.0000	55.0000	10.0000	1.52838E-01*	1.89645E-02*	-2.18606E-05*	1.41530E 00
55.0000	80.0000	10.0000	*	*	*	*

 DESTRUCT RANGES * FLIGHT PATH SHIFT*
 (MILES) * (DEGREES) * PER CENT DEST
 1.21064E 04 * 9.69648E-03 * AREA CHANGE = 1.08124E 03 (BASED ON ONE DEGREE ERROR
 1.21683E 04 * 1.29699E-02 * IN ZETA AND EPSILON,
 1.20087E 04 * 1.26497E-02 * 100 FT/SEC IN DEST VEL.)

DESTRUCT VARIABLES VALUE			DESTRUCT AREA ERROR COEFFICIENTS			DESTRUCT AREA
ZETA (DEG.)	EPSILON (DEG.)	DEST.VEL. (FT./SEC)	DEL DST VEL * SQ MI/(FT/SEC)*	DEL EPSILON * (SQ MI/DEG) *	DEL ZETA * (SQ MI/DEG) *	(SQ.MILES)
60.0000	60.0000	60.0000	*	*	*	*
45.0000	55.0000	60.0000	1.47019E 01*	-4.14444E 00*	-4.44164E 00*	2.40488E 02
55.0000	80.0000	60.0000	*	*	*	*

 DESTRUCT RANGES * FLIGHT PATH SHIFT*
 (MILES) * (DEGREES) * PER CENT DEST
 1.37176E 04 * 5.81505E-02 * AREA CHANGE = 6.07768E 02 (BASED ON ONE DEGREE ERROR
 1.42068E 04 * 7.77578E-02 * IN ZETA AND EPSILON,
 1.30682E 04 * 7.58837E-02 * 100 FT/SEC IN DEST VEL.)

TABLE 6.6 (Cont'd)

* 6TH CASE RESULTS FOR VARIOUS DEST. VEL. IN DESTRUCT TRIANGLE NO. 6 *							

DESTRUCT VARIABLES VALUE			DESTRUCT AREA ERROR COEFFICIENTS			DESTRUCT AREA	
*****			*****			*****	
ZETA (DEG.)	EPSILON (DEG.)	DEST.VEL. (FT./SEC)	DEL DST VEL SQ MI/(FT/SEC)	DEL EPSILON (SQ MI/DEG)	DEL ZETA (SQ MI/DEG)	(SQ.MILES)	
60.0000	60.0000	110.0000					
45.0000	55.0000	110.0000	6.07703E 01	-3.72128E 01	-4.01721E 01	2.00189E 03	
55.0000	80.0000	110.0000					

DESTRUCT RANGES * FLIGHT PATH SHIFT*							
(MILES)	*	(DEGREES)	PER CENT DEST				
1.53639E 04	*	1.06557E-01	AREA CHANGE =		2.99699E 02	(BASED ON ONE DEGREE ERROR	
1.65121E 04	*	1.42443E-01				IN ZETA AND EPSILON,	
1.40678E 04	*	1.39093E-01				100 FT/SEC IN DEST VEL.)	

DESTRUCT VARIABLES VALUE			DESTRUCT AREA ERROR COEFFICIENTS			DESTRUCT AREA	
*****			*****			*****	
ZETA (DEG.)	EPSILON (DEG.)	DEST.VEL. (FT./SEC)	DEL DST VEL SQ MI/(FT/SEC)	DEL EPSILON (SQ MI/DEG)	DEL ZETA (SQ MI/DEG)	(SQ.MILES)	
60.0000	60.0000	160.0000					
45.0000	55.0000	160.0000	1.26657E 02	-1.32051E 02	-1.40491E 02	6.66033E 03	
55.0000	80.0000	160.0000					

DESTRUCT RANGES * FLIGHT PATH SHIFT*							
(MILES)	*	(DEGREES)	PER CENT DEST				
1.69975E 04	*	1.54917E-01	AREA CHANGE =		1.86075E 02	(BASED ON ONE DEGREE ERROR	
1.90820E 04	*	2.07026E-01				IN ZETA AND EPSILON,	
1.49802E 04	*	2.02278E-01				100 FT/SEC IN DEST VEL.)

ILLUSTRATION OF DESTRUCT TRIANGLES

FIGURE 6.1

Example II

Choose $\Delta V_D = 10 \text{ ft/sec}$, $\Delta \epsilon = 10 \text{ deg}$, $\Delta \zeta = 20 \text{ deg}$

Then for $V_D = 700 \text{ ft/sec}$

$$\Delta(\text{Area}) = (2.31230 \times 10^1) \frac{\text{sq mi}}{\text{ft/sec}} \times (10) \text{ ft/sec} -$$

$$(8.02437) \frac{\text{sq mi}}{\text{deg}} \times 10 \text{ deg} +$$

$$(3.34773 \times 10^2) \frac{\text{sq mi}}{\text{deg}} \times 20 \text{ deg}$$

$$= 6846.41 \text{ sq mi.}$$

Since $\text{Area} = 1.28579 \times 10^4 \text{ sq mi.}$

Therefore:

$$\Delta E = \frac{6.84641 \times 10^3}{1.28579 \times 10^4} \times 100\% = \pm 53.2\%$$

Thus, data of the type generated in Table 6.1 may be utilized to establish a criterion for error limits in destruct measurements.

Table 6.7 is representative of those which could be prepared to analyze the variation in contaminated earth area as a function of variation in measurement of destruct parameters. It has been assumed in calculating the ΔE 's which appear in Table 6.3, that $\Delta V_D = 100 \text{ ft/sec}$ and that $\Delta \epsilon = \Delta \zeta = 0 \text{ deg}$.

Case 6 has been included in the set of sample cases to illustrate the concept of exclusion velocities and the use of Figure 4.4. For the set of points considered here,

TABLE 6.7

SUMMARY OF DESTRUCT DATA FOR TYPICAL DESTRUCT TRIANGLE CALCULATIONS

V_D (ft/sec)	Case 1		Case 2		Case 3		Case 4		Case 5	
	Dest. Area (sq mi)	ΔE (%)	Dest. Area (sq mi)	ΔE (%)	Dest. Area (sq mi)	ΔE (%)	Dest. Area (sq mi)	ΔE (%)	Dest. Area (sq mi)	ΔE (%)
100	474.705	225.0	2923.71	221.0	940.764	197.0	2492.72	177.0	810.056	197.0
400	6221.62	34.0	44253.3	39.4	3083.91	(-)6.17	3307.97	(-)34.5	4671.45	16.1
700	12857.9	18.0	96961.5	18.3	2552.61	(-)9.55	6229.87	20.5	6467.89	9.16
1000	18837.4	6.78	150270.0	11.1	1653.84	(-)55.9	4927.87	0.25	8274.89	6.43

Cases 1 and 2 are cases in which $\vec{V}_D$ is forward and to the side.

Cases 3 and 4 are cases in which $\vec{V}_D$ is backward and to the side.

Case 5 represents the direct backward kick, i.e., $\vec{V}_D$ directed opposite to $\vec{V}$.

It is assumed in the above table that $\Delta V_D = \pm 100$ ft/sec and that $\Delta \epsilon = \Delta \zeta = 0$ degree.

<u>Point</u>	<u>ζ</u>	<u>ϵ</u>
A	60	60
B	45	55
C	55	80

it can be ascertained from Figure 4.4 that all values of V_D greater than 65 ft/sec yield invalid results for Point A; all greater than 45 ft/sec yield invalid results for Point B; all greater than > 400 ft/sec yield invalid results for Point C. Therefore, since the first problem uses a V_D of 10 ft/sec, the area and error coefficient calculations are valid. However, for the three subsequent problems ($V_D = 60$ ft/sec, 110 ft/sec and 160 ft/sec, respectively) the calculated areas and error coefficients cannot be used to make legitimate error calculations since these combinations of V_D , ϵ , and ζ yield resultant particle velocities outside the domain of values for which the particle range, λ , calculations are valid.

GLOSSARY

$\vec{V}_R$	=	Resultant velocity, ft/sec
V_R	=	Scalar magnitude of resultant velocity, ft/sec
$\hat{i}, \hat{j}, \hat{k}$	=	Unit vectors along x, y, and z axes, respectively
$\vec{(V_R)_x}$	=	Component of resultant velocity along x-axis, ft/sec
$\vec{(V_R)_y}$	=	Component of resultant velocity along y-axis, ft/sec
$\vec{(V_R)_z}$	=	Component of resultant velocity along z-axis, ft/sec
$\vec{V}_D$	=	Destruct velocity, ft/sec
V_D	=	Scalar magnitude of destruct velocity, ft/sec
$\vec{(V_D)_x}$	=	Component of destruct velocity along x-axis, ft/sec
$\vec{(V_D)_y}$	=	Component of destruct velocity along y-axis, ft/sec
$\vec{(V_D)_z}$	=	Component of destruct velocity along z-axis, ft/sec
$\vec{V}$	=	Initial velocity, ft/sec
$\vec{(V)_x}$	=	Component of initial velocity along x-axis, ft/sec

GLOSSARY (CONTINUED)

ρ	=	Atmospheric density, lb/ft ³
V	=	Scalar magnitude of initial velocity, ft/sec
A_c	=	Cross-sectional area of re-entering particle, ft ²
m	=	Body mass of re-entering particle, lb.
λ	=	Re-entry range of particle, radians
$\bar{A}$	=	Triangular earth area, sq. mi.
R	=	Radius of earth, miles
X_1	=	Error coefficient for destruct velocity, $\left(\frac{\partial \bar{A}}{\partial V_D} \right), \frac{\text{sq mi}}{\text{ft/sec}}$
Y_1	=	Error coefficient for zeta, $\left(\frac{\partial \bar{A}}{\partial \zeta} \right), \frac{\text{sq. miles}}{\text{degree}}$
Z_1	=	Error coefficient for epsilon, $\left(\frac{\partial \bar{A}}{\partial \epsilon} \right), \frac{\text{sq. miles}}{\text{degree}}$

GLOSSARY (CONTINUED)

$\longrightarrow$		
$(V)_y$	=	Component of initial velocity along y-axis, ft/sec
$\longrightarrow$		
$(V)_z$	=	Component of initial velocity along z-axis, ft/sec
$\longrightarrow$		
$(V_R)_{xy}$	=	Component of resultant velocity in horizontal plane, ft/sec
$(V_R)_{xy}$	=	Scalar magnitude of $\overrightarrow{(V_R)_{xy}}$, ft/sec
ζ	=	Direction angle of destruct velocity, measured from plane of local horizontal to line of action of destruct velocity, radians.
ϵ	=	Direction angle of destruct velocity measured from plane of original flight path to plane containing destruct velocity vector, measured counter-clockwise, radians
$\emptyset$	=	Initial flight angle, radians
ν	=	Angle between plane of resultant velocity and plane of initial velocity, radians
Ψ	=	Flight angle of resultant velocity, angle between line of action of resultant velocity and local horizontal, radians
E_D	=	Drag energy, lb ft ² /sec ²
C_D	=	Drag coefficient

BIBLIOGRAPHY

1. J. M. Bridges, H. G. Hargrove, D. Hoecker, "Summary Report of the Theoretical, Analytical and Experimental Studies Performed in the Investigation of the Re-Entry Burn-Up of NERVA Fuel," WANL-TME-135 (September 1963).
2. H. G. Hargrove, "Aerodynamic Heating and Oxidation of Graphite Particles in the Free Molecule Flow Regime," WANL-TNR-067 (September 1962).
3. J. M. Bridges, "Experimental Study of the Re-Entry Oxidation of NERVA Fuel," WANL-TME-072 (October 1962).
4. M. J. Engel, et al, "The Effects of Atmospheric Re-Entry on Graphite Particles," DIH:214-L28F (January 1964) (U).
5. H. G. Hargrove, D. Hoecker, "Flight Mechanics of NERVA Re-Entry and Orbital Lifetime Studies," WANL-TME-635 (December 1963).
6. R. W. Hamming, Numerical Methods for Scientists and Engineers, McGraw-Hill Book Company, Inc. (1962).
7. Sokolnikoff, I. S. and E. S., Higher Mathematics for Engineers and Physicists, McGraw-Hill Book Company, Inc. (1941).
8. C. P. Saalbach, "A Generalized Least Squares Program," WANL-TME-756 (April 1964).
9. F. B. Hildebrand, Introduction to Numerical Analysis, McGraw-Hill Book Company, Inc. (1956).

APPENDIX I

METHOD OF LEAST SQUARES FOR FITTING POLYNOMIALS

The method of least squares operates on the principal that the best polynomial is the one that reduces the sum of the square of the differences between the actual values of the dependent variable and the calculated values (using the polynomial) to a minimum throughout the domain of interest. From the Gaussian Error Law we find that this criterion is tantamount to saying that the best empirical fit (in a least square sense) is the one that produces the least average error throughout the domain of values^(7,8). Thus, we seek a solution for the dependent variable, Y_c , in the form:

$$Y_c = f(X) \equiv \sum_{i=1}^N C_i [\phi_i(X)] \quad (I-1)$$

such that throughout the range of values of Y we obtain the following condition:

$$\sum_{i=1}^N (Y - Y_c)_i^2 = \text{minimum} \quad , \quad (I-2)$$

where Y is the actual data value.

Hence, our job is to determine the coefficients of a polynomial that has been arbitrarily selected by us, such that the requirement of Equation (I-2) can be met as closely as possible.

Thus, for a given set of data points $(Y_k, X_k, k = 1, R)$ and an arbitrary set of functions $(\phi_i(X), i = 1, N)$ where $R \geq N$ we seek to determine the C_i 's ($C_i, i = 1, N$) such

that:

$$Y_c = C_1 \phi_1(X) + C_2 \phi_2(X) + \dots + C_N \phi_N(X) \quad (1-3)$$

best fits the proposition outlined by Equation (1-2).

Since the $\phi_i(X)$ are arbitrarily selected by the investigator, then Y_c actually becomes a function of the C_i 's or:

$$Y_c = F(C_i) \quad (1-4)$$

Thus, Equation (1-2) may now be expressed as:

$$F(C_i) = \sum_{k=1}^R W_k \left[Y_k - Y_c(X_k) \right]^2 = \text{minimum} \quad (1-5)$$

for each i , where the subscript "k" refers to data points.

Equation (1-5) may be expanded with the aid of Equation (1-3) into the following:

$$\begin{aligned}
 F(C_i) &= \sum_{k=1}^R W_k \left[Y_k - C_1 \phi_1(X_k) - C_2 \phi_2(X_k) - \dots - C_N \phi_N(X_k) \right]^2 \\
 &= \text{minimum} \quad ,
 \end{aligned} \quad (1-6)$$

where W_k is a weighting function associated with each point.*

Then, for the criterion of Equation (1-6) to be valid we require that:

$$\frac{\partial [F(C_i)]}{\partial (C_i)} = 0 \quad (1-7)$$

for all C_i .

* The use of the weighting function will not be described here, but reference should be made to a standard text on numerical analysis, e.g., Reference (7).

Hence, differentiating Equation (I-7) with respect to each C_i and setting the result to zero, we obtain:

$$\begin{aligned}
 \frac{\partial F(C_i)}{\partial C_i} &= \sum_{k=1}^R 2W_k \phi_i(X_k) [Y_k - C_1 \phi_1(X_k) - C_2 \phi_2(X_k) - \dots - C_N \phi_N(X_k)] \\
 &= 0
 \end{aligned} \tag{I-8}$$

Rearranging Equation (I-8), we get:

$$\begin{aligned}
 \sum_{k=1}^R W_k \phi_i(X_k) Y_k &= \sum_{k=1}^R [W_k \phi_i(X_k)] C_1 \phi_1(X_k) + \sum_{k=1}^R [W_k \phi_i(X_k)] C_2 \phi_2(X_k) + \\
 &\dots + \sum_{k=1}^R [W_k \phi_i(X_k)] C_N \phi_N(X_k)
 \end{aligned} \tag{I-9}$$

We may put Equation (I-9) into the following notation:

$$\sum_{k=1}^R W_k \phi_i(X_k) Y_k = \sum_{k=1}^R W_k \phi_i(X_k) \left(\sum_{j=1}^N C_j \phi_j(X_k) \right) \tag{I-10}$$

Now, since Equation (I-10) can be written for all C_i 's from $i=1, N$ the result is N simultaneous linear (in C_i) equations which can then be solved for each C_i .

Such a task is obviously a laborious one and the entire job takes on mammoth proportions when one considers that the polynomial for which the coefficients are determined may not give a "fit" throughout the domain of interest to the desired accuracy. In short, we still have a trial and error procedure in selecting the functional forms of the polynomial,

in addition to the job of then determining the coefficient for each term. Hence, a digital computer program was prepared to solve (using the technique outlined above) for the coefficients of polynomials. However, instead of using Equation (I-10), the solution for the C_i 's is carried out with the aid of matrix algebra. Hence, if the following matrices are defined:

$$A \equiv \begin{pmatrix} \phi_1(x_1) & \phi_2(x_1) & \dots & \phi_N(x_1) \\ \phi_1(x_2) & \phi_2(x_2) & \dots & \phi_N(x_2) \\ \vdots & \vdots & & \vdots \\ \phi_1(x_k) & \phi_2(x_k) & \dots & \phi_N(x_k) \end{pmatrix},$$

$$W \equiv \begin{pmatrix} w_1 & 0 & 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & w_2 & & & & & & \\ 0 & 0 & w_3 & & & & & \\ 0 & 0 & 0 & w_4 & & & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & w_k \end{pmatrix},$$

$$C \equiv \begin{pmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ \cdot \\ \cdot \\ \cdot \\ C_N \end{pmatrix},$$

and

$$D \equiv \begin{pmatrix} \sum_{k=1}^R Y_k W_k \phi_1(X_k) \\ \sum_{k=1}^R Y_k W_k \phi_2(X_k) \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \sum_{k=1}^R Y_k W_k \phi_N(X_k) \end{pmatrix},$$

then it may be verified by matrix multiplication^(9,10) that the equations resulting from Equation (I-10) can be represented by:

$$(A W A^*) C = D \quad (I-11)$$

where A^* is the transpose of the matrix A . Hence, the element $d_{i,i}$ of the matrix $(A W A^*)$ is identical with the coefficient C_i in the i^{th} row of the set of equations developed from Equation (I-10).

Thus, the least squares digital computer program that was written for use in this task involved only three main segments, viz: (1) A Main Program to read in the data points and a proposed polynomial, (2) a subroutine named FUNCTION that generated the matrix "A" and (3) a second subroutine, SLSTQR, that generated the transpose of matrix "A", performed the matrix multiplication, and then, using a library subroutine for solving simultaneous equations called ISIMEQ, the coefficients for the polynomial were generated.

One note should be mentioned at this point in the discussion. In Equation (I-3) we proposed the polynomial could be a series of any functional forms of X . Hence, the form of the polynomial is not necessarily restricted to explicit forms of independent variables, i.e., X, X^2, X^4 , etc., but could be more involved expressions such as $X e^X \cos(\eta \bar{N} X)$. Finally, no restrictions were placed on the dimension (number) of independent variables upon which it depends. Thus, the use of "X" in Equations (I-1) through (I-10) is purely notational only, for any number of variables could be used, although rarely would the number exceed three, i.e., for the case where:

$$Y = f(X, Z, \theta) \quad .$$

In our particular situation, the latter two considerations have a most important bearing since we are trying to find an empirical fit for:

$$\lambda = f(V_R, \Psi) \quad ,$$

which must take the form of a two-dimensional polynomial, plus the fact that over the wide variation in λ as given in Figures 4-4 to 4-5 we do not know which functional forms might produce the most compact polynomial having the required accuracy in the domain.

Therefore, the least squares routine was written to handle two-dimensional polynomials, with the capability of specifying each term in the polynomial as a multiple of up to three different functional forms for each variable. With this tool at our disposal, we could solve for the coefficients in a relatively simple polynomial such as:

$$Y = B_0 + B_1X + B_2XZ + B_3XZ^2 \quad , \quad (I-12)$$

or for more complicated expressions like:

$$Y = B_0 + B_1X(e^X)(\cos(nZ)) + B_2X^2(\log X)(\sin X)(e^Z)(\cos Z) + B_3X^3Z^2e^{|z|} \quad (I-13)$$

Finally, one further computational aid was added to the program. This took the form of a subroutine named EVAL that compared the sum of the squared differences between the actual value and calculated value of Y , at all data points, using only one term at a time in the polynomial. This, then, served as a measure of importance the term commanded in contributing to the accuracy of the fit. For example, the smaller the sum of the squared differences, the more important the term. This tool proved to be an invaluable aid in building up a polynomial and evaluating it for use as a functional relationship between λ , Ψ , and V_R as explained in the following appendix.

The total combination of Main Program, and Subroutines FUNCTION, SLSQRS and EVAL make up the digital computer program NEFØD (NERVA FOOTPRINT from ORBITAL DESTRUCT).

Thus, the laborious task of fitting a given polynomial to a set of data has been reduced to reading data points into the computer, viz., λ , V_R , Ψ , the size of the polynomial (number of terms), and the functional forms for each term of the polynomial and the powers into which each functional form is raised in that term of the polynomial. The results obtained are the following:

(1) The coefficients of the polynomial

(2) Difference in calculated and actual value of each λ ; i.e., $\lambda_c - \lambda$

(3) Relative error at each point $\equiv \frac{\lambda_c - \lambda}{\lambda}$

(4) Average relative error $\equiv \frac{\left| \sum_{i=1}^N \frac{(Y_c - Y)_i}{Y_i} \right|}{N}$

(5) Average difference $\equiv \frac{\sum_{i=1}^N |(Y_c - Y)_i|}{N}$

(6) Average sum of squares $\equiv \frac{\sum_{i=1}^N (Y_c - Y)_i^2}{N}$

$$(7) \text{ Standard deviation} = \sigma \equiv \left(\frac{\sum_{i=1}^N (Y_c - Y)_i^2}{N} \right)^{1/2}$$

$$(8) \text{ Probable error} = 0.6745\sigma$$

These eight qualifying factors, particularly items 6, 7, and 8, served as measuring rods to determine the accuracy of the fit and together with the term evaluation by Sub-routine EVAL formed a rather complete analysis of the fit of any given polynomial to the data points.

APPENDIX II

GENERAL OUTLINE OF NEFØD COMPUTER PROGRAM FOR POLYNOMIAL FITTING

II.1 CONTENTS

As explained in Appendix I, the polynomial fitting routine is composed of a Main Program and three subroutines. In this appendix, a brief description of these four segments of the program is given and a list of the important input/output variable names is given. Finally a complete listing is provided for reference along with the results from a sample problem.

II.2 EXPLANATION OF PROGRAM SEGMENTS

II.2.1 Main Program

This part of the entire program serves as the vehicle for reading in:

- (1) the data points (values of independent and dependent variables),
- (2) the polynomial chosen for fitting to the data points, and
- (3) transformation factors for the independent variables.

The transformation factors permit the user to obtain a polynomial fit of $Z = f(x,y)$ in the form:

$$Z = f(X^*, Y^*) \quad (II-1)$$

where:

$$X^* = \left[(X - C_1)(C_2) \right] / C_3 \quad (II-2)$$

$$Y^* = \left[(Y - C_4)(C_5) \right] / C_6 \quad (II-3)$$

These transformations are extremely valuable and find extensive use in the particular program in transforming the resultant velocity, V_R . Hence, the polynomial fit of $\lambda = f(\Psi, V_R)$ was actually obtained in the form:

$$\lambda = f(\Psi, V_R^*) \quad (II-4)$$

where

$$V_R^* = \frac{(V_R - 24,000)}{(24,000)}$$

II.2.2 Subroutine FUNCT

This subroutine is called into use by the main program. Its purpose is to construct a matrix in which each row represents one of the functions in the polynomial and each column one of the data points. Hence, a given element of the array represents the quantity obtained by multiplying together the value of the independent variables at a given data point according to the relationship given by one of the terms comprising the polynomial.

II.2.3 Subroutine SLSTQR

Subroutine SLSTQR performs the bulk of the calculations in the NEFØD Program. This segment takes the matrix built-up by the FUNCT Subprogram and (1) generates the transpose of this matrix, (2) performs matrix multiplication of the original matrix and its transpose and (3) feeds the results into the ISIMEQ Routine (the library routine for solving simultaneous equations). The results are the values of the coefficients for each term of the polynomial.

In addition this subprogram uses the polynomial to calculate the value of the dependent variable, Y_c , at each data point and compares these with the actual values of Y from the standpoint of relative error and difference. Relative error is defined to be:

$$\frac{(Y_{ci} - Y_i)}{Y_i}, \text{ and}$$

difference is: $(Y_{ci} - Y_i)$, where Y_{ci} and Y_i are the calculated and actual values of the dependent variable, respectively, at the i^{th} data point. Subroutine SLSTQR also performs all calculations (other than the term evaluation) for analyzing the accuracy of the fit.

These calculations are:

- (1) Average relative error,
- (2) Average difference,
- (3) Average sum of squares,
- (4) Standard deviation,
- (5) Probable error,

as defined in Appendix I.

II.2.4 Subroutine EVAL

Subroutine EVAL is called by the Main Program. Its sole function is to evaluate the importance of each term in the polynomial. This evaluation is made by calculating λ using just one term of the polynomial and then subtracting that value from the actual value of λ and squaring this difference for every data point. This is done for every term of the polynomial.

Since the smaller the squared difference the more important the term, this method provides a simple, but effective means of evaluating the individual elements of the polynomial. This evaluation is carried out by comparing each term with the lowest squared difference obtained in all the terms. Those terms having a squared difference in excess of 10^3 times the lowest are dropped from further consideration.

II.3 OUTPUT FORMAT AND INFORMATION

The output from the NEFØD Program gives the coefficients for each term, the functional factor for each term, and the power to which it is raised. A sample output is given in Section II.5. To illustrate the use of this table, we will examine the line beginning with B(8):

$$\begin{array}{ccc}
 \text{B(8)} & 2.7331\text{E-01} & \frac{((1))'(13), ((-0))'(-0), ((-0))'(-0)}{\Psi \text{ factors}} \frac{((1))'(3), ((-0))'(-0), ((-0))'(-0)}{V_R^* \text{ factors}}
 \end{array}$$

This line gives the complete makeup of the eighth term of the polynomial. The first three functional factors correspond to functions of Ψ , the last three to V_R^* . The numbers enclosed in double parenthesis are identification numbers which correspond to a particular functional form for Ψ or V_R^* and the number enclosed in single parenthesis is the power to which this function is raised. (The functional form code is given at the end of the coefficient list.)

Hence, the line above is translated to read:

$$2.7331 \times 10^{-1} (\Psi^{13})(V_R^{*3})$$

This is the eighth term in the polynomial. As noted in Section II.2.1 of this Appendix, either one or both of the independent variables, Ψ and V_R , may have been transformed.

However, any transformation of either of the variables is the same for all terms in a given polynomial being fitted.

The remainder of the output is nearly self-explanatory and consists of: (1) a listing of each actual value of the dependent variable and the calculated value using the polynomial, (2) the relative error and difference at that point between the calculated and given values and, (3) the values of the various measuring devices used to determine the "goodness" of the fit. (These are explained in Appendix I.) Finally, the term evaluation list is given for use in comparing the terms as outlined in Section II.2.4 of the appendix.

Table II-1 gives a list of the most important variable names in NEFØD.

The polynomials which were found to fit acceptably the relation $\lambda = f(\Psi, V_R^*)$ are given in Tables 4.1 and 4.2 of the main text.

TABLE II-1

LIST OF MOST IMPORTANT VARIABLE NAMES IN NEFØD

A	- Matrix in which each row represents a function and each column a data point.
W	- Weighting function.
R	- Transformed value of first independent variable.
RØ	- Actual value of first independent variable.
T	- Transformed value of second independent variable.
TØ	- Actual value of second independent variable.
IVF	- Function factor in one term of polynomial.
IVFP	- Power to which function factor in one term of polynomial is raised.
NRØW	- Number of terms in polynomial.
NCØL	- Number of data points.
IADIM	- Used to transform A matrix from one to two dimensional form (optional use in program).
NUMBRA	- Used to set aside proper number of storage locations for matrices used to generate coefficients.
DATRAN	- Factors used to transform original values of independent variables.
D	- Name of array in which DATRAN values are stored in computer.
ITAG	- Identification number used with each DATRAN value that tells how value is to be used to transform independent variables.

II.4 NEFØD LISTINGS

```

*      XEQ
C      MAIN PROGRAM FOR LEAST MEAN SQUARE ROUTINE  HG HARGROVE
      COMMON A,NROW,NCOL,R,T,Y,W,IVF, IVFP, B, IADIM, NMBRA
      DIMENSION A(17250), Y(300), W(300), R(300), T(300), IVF(6,50),
      1IVFP(6,50), TITLE(14), RO(300), TO(300), TEVAL(50), B(50), D(6),
      2DATRAN(6), ITAG(6)
      1 RIPT 5, 40, TITLE
      40 FORMAT(13A6, A2)
      RIPT 5, 50, LTEST, NCOL, (RO(J), TO(J), Y(J), W(J),J=1,NCOL)
      50 FORMAT (2I5/(4F10.4))
      48 RIPT 5, 51, NROW, IADIM, NMBRA
      51 FORMAT(3I5)
      RIPT 5, 101, ((IVF(I,J), IVFP(I,J), I=1,6), J=1, NROW)
      101 FORMAT(6(I3,I2))
      CALL EPMDE(5)
C      **ORIGINAL DATRAN VALUES****
      D(1) = 0.0
      D(2) = 1.0
      D(3) = 1.0
      D(4) = 0.0
      D(5) = 1.0
      D(6) = 1.0
      RIPT 5, 80, (DATRAN(J), ITAG(J), J=1,6)
      80 FORMAT(6(F10.4,I2))
      DO 10 J=1,6
      N= ITAG(J)
      IF(N)11,11,10
      10 D(N) = DATRAN(J)
      11 DO 15 K=1, NCOL
      R(K) = (RO(K)+D(1))*D(2)/D(3)
      T(K) = (TO(K)+D(4))*D(5)/D(6)
      15 CONTINUE
      TIME = CLOCKF(GMONTH, DAY)*0.001
      JOUR = DAY
      MONTH = GMONTH
      WOT 6, 200, TITLE, MONTH, JOUR
      200 FORMAT(1H1,12X13A6,A2, 5X12, 1H/I2,///)
      WOT 6, 205, (N, D(N), N= 1,3)
      205 FORMAT(12X29H1ST VARIABLE TRANSFORM FACTORS//((15X2HD(I1,4H) = 1PE11
      1.4/))
      WOT 6, 206, (N, D(N), N= 4,6)

```

```
206 FORMAT(///12X29H2ND VARIABLE TRANSFORM FACTORS//(15X2HD(I1,4H) = 1P
1E11.4/))
      WOT 6, 210
210 FORMAT(10X14HCOEFFICIENTS    30X20HTERMS OF POLYNOMIAL //)
      CALL FUNCT
      CALL SLSTQR
      CALL EVAL
      LTEST = LTEST - 1
      IF(LTEST)48, 1, 48
      END
```



```

      GO TO 380
315  A(IA) = A(IA)* COSF(T(J))*IVFP(I,K)
      GO TO 380
320  IF (I-3) 321,321,325
321  A(IA) = A(IA)* SINR(R(J))*IVFP(I,K)
      GO TO 380
325  A(IA) = A(IA)* SINR(T(J))*IVFP(I,K)
      GO TO 380
330  IF (I-3) 331,331,335
331  A(IA) = A(IA)* EXPR(R(J))*IVFP(I,K)
      GO TO 380
335  A(IA) = A(IA)* EXPR(T(J))*IVFP(I,K)
      GO TO 380
340  IF (I-3) 341,341,345
341  A(IA) = A(IA)* EXPR(ABSR(R(J)))*IVFP(I,K)
      GO TO 380
345  A(IA) = A(IA)* EXPR(ABSR(T(J)))*IVFP(I,K)
      GO TO 380
350  IF (I-3) 351,351,355
351  A(IA) = A(IA)* LOGR(R(J))*IVFP(I,K)
      GO TO 380
355  A(IA) = A(IA)* LOGR(T(J))*IVFP(I,K)
      GO TO 380
360  IF (I-3) 361,361,365
361  A(IA) = A(IA)*(LOGR(R(J))/2.3026)*IVFP(I,K)
      GO TO 380
365  A(IA) = A(IA)*(LOGR(T(J))/2.3026)*IVFP(I,K)
      GO TO 380
370  IF(I-3) 371, 371, 373
371  A(IA) = A(IA)*(1.0/EXPR(ABSR(R(J))))*IVFP(I,K)
      GO TO 380
373  A(IA) = A(IA)*(1.0/EXPR(ABSR(T(J))))*IVFP(I,K)
      GO TO 380
375  IF(I - 3) 376, 376, 380
376  A(IA) = A(IA)*(1.0/(1.0 + (R(J)**2)))*IVFP(I,K)
380  I = I+1
      IF (I-6) 190,190,390
390  CONTINUE
400  CONTINUE
      RETURN
      END

```

```

SUBROUTINE SLSTQR
COMMON A,NROW,NCOL,R,T,Y,W,IVF, IVFP, B, IADIM, NMBRA
DIMENSION A(17250), Y(300), W(300), R(300), T(300), IVF(6,50),
1IVFP(6,50), TITLE(14), RO(300), TO(300), TEVAL(50), B(50), D(6),
2DATRAN(6), ITAG(6)
NCF1 = NROW
NPT1 = NCOL
IA1= IADIM
I1D= 0
IF(IA1) 12,13,12
12 I1D= IA1-NPT1
N1= NCF1*IA1
GO TO 14
13 N1= NCF1*NPT1
14 N2= N1+1
N3= N1+NCF1
N4= N1+NCF1*NCF1+1
N5= N4+NCF1-1
IF(NMBRA-N5) 1,2,2
1 WRITE OUTPUT TAPE 6,99
99 FORMAT(33X43HA MATRIX EXCEEDS DIMENSION STATEMENT SIZE )
CALL EXIT
2 DO 3 I=N2,N5
3 A(I)= 0.
I1= 1
DO 4 I2=1,NCF1
DO 5 I3=1,NPT1
Z= A(I1)*W(I3)
I4= I1
DO 6 I5=N2,N3
A(I5)= A(I5)+A(I4)*Z
IF ACCUMULATOR OVERFLOW 50, 6
50 WOT 6, 52
52 FORMAT(30X48HMATRIX TIMES TRANSPOSE EXCEEDS MAXIMUM NO. SIZE )
CALL EXIT
6 I4= I4+NPT1
5 I1= I1+1
I1= I1+I1D
I6= N2
DO 7 I7=N2,N3
A(I6)= A(I7)

```

```

7 I6= I6+NCF1
  N2= N2+NCF1+1
4 N3= N3+NCF1
  I1= 1
  DO 11 I2=N4,N5
  DO 8 I3=1,NPT1
    A(I2)= A(I2)+W(I3)*A(I1)*Y(I3)
    IF ACCUMULATOR OVERFLOW 55, 8
55 WOT 6, 56
56 FORMAT(26X58HFORMATION OF B MATRIX FOR ISIMEQ EXCEEDS MAXIMUM NO.
  1SIZE )
  CALL EXIT
  8 I1= I1+1
11 I1= I1+I1D
  N6= N1+1
  CALL ISIMEQ(A(N6),NCF1,NCF1,1)
  J= 0
  DO 10 N= N4, N5
    J=J + 1
10 B(J) = A(N)
  WOT 6, 100, (J, B(J), (IVF(I,J),IVFP(I,J),I=1,6), J=1,NROW)
100 FORMAT(5X2HB(I2,4H) = 1PE11.4, 5X2H((I2,4H))'(I2,5H), ((I2,4H))'(I
  12,5H), ((I2,4H))'(I2,5H), ((I2,4H))'(I2,5H), ((I2,4H))'(I2,5H), ((
  2I2,4H))'(I2,1H), // )
  WOT 6, 101
101 FORMAT(38X34HTRANSLATION OF TERMS IN POLYNOMIAL //,15X74H(FIRST TH
  1REE TERMS CORRESPOND TO 1ST VARIABLE, SECOND THREE TO 2ND VARIABLE)
  2//,20X10H((0))= 1 //,20X13H((1))= R OR T //,20X18H((2))= COS(R OR
  3 T)//,20X18H((3))= SIN(R OR T) //, 20X18H((4))= EXP(R OR T) //,20X
  433H((5))= EXP(ABSOLUTE VALUE R OR T) //,20X19H((6))= LOGN(R OR T)/
  5//,20X18H((7))= LOG(R OR T) //, 20X34H((8))= EXP(-ABSOLUTE VALUE R
  6OR T) //, 20X55H((9 TO 99))= OTHER FUNCTIONS OF R OR T PREPARED BY
  7USER ///)
C RELATIVE ERROR
  WRITE OUTPUT TAPE 6,102
102 FORMAT( 30X48HCOMPARISON OF ACTUAL WITH CALCULATED RANGES //
  119X12HDEP. VAR.(Y), 10X8HY(CALC) , 13X5HERROR, 13X10HDIFFERENCE )
  SUMS = 0.0
  DIFFSQ = 0.0
  SUMSQT = 0.0
  DO 15 I=1,NCOL

```



```

YC=0.0
DO 70 J=1,NROW
IA = I + (J-1)*NCOL
YC = YC + B(J)*A(IA)
GO TO 70
70 CONTINUE
DIFF = YC - Y(I)
SUMS = SUMS + ABSF(DIFF)
DIFFSQ = DIFFSQ + DIFF**2
DELTA = ABSF(DIFF/Y(I))
SUMSQT = SUMSQT + DELTA
WOT 6, 103, Y(I), YC, DELTA, DIFF
103 FORMAT(15X,4(8X,1PE12.4))
15 CONTINUE
C FINAL ERROR VALUES
XCOL = NCOL
SUMS = SUMS/XCOL
DIFFSQ = DIFFSQ/XCOL
SUMSQT = SUMSQT/XCOL
DEV = SQRTF(DIFFSQ)
PE = 0.6745 * DEV
WOT 6, 205
205 FORMAT(// 38X, 33HRELATIVE ERRORS--ACCURACY OF FIT // )
WOT 6, 210, SUMSQT, SUMS, DIFFSQ, DEV, PE
210 FORMAT (37X22HAVG. RELATIVE ERROR = 1PE11.4 //, 37X21HAVG. DIFF(DE
1GREES) = 1PE11.4 //, 37X22HAVG. SUM OF SQUARES = 1PE11.4 //, 37X
221HSTANDARD DEVIATION = 1PE11.4 //,37X17HPROBABLE ERROR = 1PE11.4)
RETURN
END

```

```

SUBROUTINE EVAL
COMMON A,NROW,NCOL,R,T,Y,W,IVF, IVFP, B, IADIM, NMBRA
DIMENSION A(17250), Y(300), W(300), R(300), T(300), IVF(6,50),
1IVFP(6,50), TITLE(14), RO(300), TO(300), TEVAL(50), B(50), D(6),
2DATRAN(6), ITAG(6)
IA = 1
DO 60 J= 1, NROW
TEVAL(J) = 0.0
DO 61 I= 1, NCOL
XT = B(J)*A(IA)
TEVAL(J) = TEVAL(J) + (Y(I) - XT)**2
IA = IA + 1
61 CONTINUE
XNCOL = NCOL
TEVAL(J) = TEVAL(J)/XNCOL
60 CONTINUE
WOT 6, 21
21 FORMAT(///47X15HTERM EVALUATION /)
WOT 6, 20,(J, TEVAL(J), J= 1, NROW)
20 FORMAT(20X5HTERM(I2, 4H) = 1PE11.4)
RETURN
END
DATA

```

II.5 EXAMPLE OF NEFØD OUTPUT

FOOTPRINT PROBLEM EARTH RANGE OF DEBRIS AS F(RE-ENTRY ANGLE, VELOCITY)BP= 0.07

7/20

1ST VARIABLE TRANSFORM FACTORS

D(1) = 0.

D(2) = 1.0000E 00

D(3) = 1.0000E 00

2ND VARIABLE TRANSFORM FACTORS

D(4) = -2.4000E 04

D(5) = 1.0000E 00

D(6) = 2.4000E 04

COEFFICIENTS

TERMS OF POLYNOMIAL

B(1) = 2.6218E 01	((0))'(-0), ((-0))'(-0), ((-0))'(-0), ((-0))'(-0), ((-0))'(-0), ((-0))'(-0)
B(2) = 1.0984E-05	((1))'(16), ((-0))'(-0), ((-0))'(-0), ((-0))'(-0), ((-0))'(-0), ((-0))'(-0)
B(3) = 9.6589E 19	((-0))'(-0), ((-0))'(-0), ((-0))'(-0), ((1))'(16), ((-0))'(-0), ((-0))'(-0)
B(4) = 2.8861E-06	((1))'(15), ((-0))'(-0), ((-0))'(-0), ((1))'(1), ((-0))'(-0), ((-0))'(-0)
B(5) = 5.2117E 19	((1))'(1), ((-0))'(-0), ((-0))'(-0), ((1))'(15), ((-0))'(-0), ((-0))'(-0)
B(6) = 1.7626E-02	((1))'(14), ((-0))'(-0), ((-0))'(-0), ((1))'(2), ((-0))'(-0), ((-0))'(-0)
B(7) = -2.7232E 17	((1))'(2), ((-0))'(-0), ((-0))'(-0), ((1))'(14), ((-0))'(-0), ((-0))'(-0)
B(8) = 2.7331E-01	((1))'(13), ((-0))'(-0), ((-0))'(-0), ((1))'(3), ((-0))'(-0), ((-0))'(-0)
B(9) = -1.4829E 17	((1))'(3), ((-0))'(-0), ((-0))'(-0), ((1))'(13), ((-0))'(-0), ((-0))'(-0)

B(10) = -1.4937E 02	((1))' (12), ((-0))' (-0), ((-0))' (-0), ((1))' (4), ((-0))' (-0), ((-0))' (-0)
B(11) = -1.6293E 14	((1))' (4), ((-0))' (-0), ((-0))' (-0), ((1))' (12), ((-0))' (-0), ((-0))' (-0)
B(12) = -4.2985E 03	((1))' (11), ((-0))' (-0), ((-0))' (-0), ((1))' (5), ((-0))' (-0), ((-0))' (-0)
B(13) = 1.4546E 14	((1))' (5), ((-0))' (-0), ((-0))' (-0), ((1))' (11), ((-0))' (-0), ((-0))' (-0)
B(14) = 6.9527E 05	((1))' (10), ((-0))' (-0), ((-0))' (-0), ((1))' (6), ((-0))' (-0), ((-0))' (-0)
B(15) = 1.5009E 12	((1))' (6), ((-0))' (-0), ((-0))' (-0), ((1))' (10), ((-0))' (-0), ((-0))' (-0)
B(16) = 2.8572E 07	((1))' (9), ((-0))' (-0), ((-0))' (-0), ((1))' (7), ((-0))' (-0), ((-0))' (-0)
B(17) = -8.9695E 10	((1))' (7), ((-0))' (-0), ((-0))' (-0), ((1))' (9), ((-0))' (-0), ((-0))' (-0)
B(18) = -1.5300E 09	((1))' (8), ((-0))' (-0), ((-0))' (-0), ((1))' (8), ((-0))' (-0), ((-0))' (-0)
B(19) = -2.1688E-08	((1))' (15), ((-0))' (-0), ((-0))' (-0), ((-0))' (-0), ((-0))' (-0), ((-0))' (-0)
B(20) = -3.5660E 20	((-0))' (-0), ((-0))' (-0), ((-0))' (-0), ((1))' (15), ((-0))' (-0), ((-0))' (-0)
B(21) = -2.8535E-04	((1))' (14), ((-0))' (-0), ((-0))' (-0), ((1))' (1), ((-0))' (-0), ((-0))' (-0)
B(22) = -3.2175E 18	((1))' (1), ((-0))' (-0), ((-0))' (-0), ((1))' (14), ((-0))' (-0), ((-0))' (-0)
B(23) = -2.5777E-04	((1))' (14), ((-0))' (-0), ((-0))' (-0), ((-0))' (-0), ((-0))' (-0), ((-0))' (-0)
B(24) = 5.5728E 19	((-0))' (-0), ((-0))' (-0), ((-0))' (-0), ((1))' (14), ((-0))' (-0), ((-0))' (-0)
B(25) = -8.9358E-05	((1))' (13), ((-0))' (-0), ((-0))' (-0), ((1))' (1), ((-0))' (-0), ((-0))' (-0)

B(26) = 2.9071E 16	((1))*(1), ((-0))*(-0), ((-0))*(-0), ((1))*{13}, ((-0))*(-0), ((-0))*(-0)
B(27) = 4.9083E 14	((1))*{ 3}, ((-0))*(-0), ((-0))*(-0), ((1))*{11}, ((-0))*(-0), ((-0))*(-0)
B(28) = -1.2974E 12	((1))*{ 4}, ((-0))*(-0), ((-0))*(-0), ((1))*{10}, ((-0))*(-0), ((-0))*(-0)
B(29) = -2.1764E 18	((-0))*(-0), ((-0))*(-0), ((-0))*(-0), ((1))*{13}, ((-0))*(-0), ((-0))*(-0)
B(30) = 1.5498E-03	((1))*{12}, ((-0))*(-0), ((-0))*(-0), ((-0))*(-0), ((-0))*(-0), ((-0))*(-0)
B(31) = -1.7112E 12	((1))*{ 3}, ((-0))*(-0), ((-0))*(-0), ((1))*{ 9}, ((-0))*(-0), ((-0))*(-0)
B(32) = 1.5250E 10	((1))*{ 1}, ((-0))*(-0), ((-0))*(-0), ((1))*{ 7}, ((-0))*(-0), ((-0))*(-0)
B(33) = 1.1410E 09	((-0))*(-0), ((-0))*(-0), ((-0))*(-0), ((1))*{ 6}, ((-0))*(-0), ((-0))*(-0)
B(34) = -2.3659E-01	((1))*{ 5}, ((-0))*(-0), ((-0))*(-0), ((1))*{ 1}, ((-0))*(-0), ((-0))*(-0)
B(35) = 1.2973E-02	((1))*{ 5}, ((-0))*(-0), ((-0))*(-0), ((-0))*(-0), ((-0))*(-0), ((-0))*(-0)
B(36) = 3.9582E 02	((1))*{ 3}, ((-0))*(-0), ((-0))*(-0), ((1))*{ 2}, ((-0))*(-0), ((-0))*(-0)
B(37) = -1.2566E 00	((1))*{ 4}, ((-0))*(-0), ((-0))*(-0), ((-0))*(-0), ((-0))*(-0), ((-0))*(-0)
B(38) = -1.5697E 01	((1))*{ 3}, ((-0))*(-0), ((-0))*(-0), ((1))*{ 1}, ((-0))*(-0), ((-0))*(-0)
B(39) = -1.2069E-01	((1))*{ 3}, ((-0))*(-0), ((-0))*(-0), ((-0))*(-0), ((-0))*(-0), ((-0))*(-0)
B(40) = 5.4163E 01	((1))*{ 2}, ((-0))*(-0), ((-0))*(-0), ((1))*{ 1}, ((-0))*(-0), ((-0))*(-0)
B(41) = 5.0416E 00	((1))*{ 2}, ((-0))*(-0), ((-0))*(-0), ((-0))*(-0), ((-0))*(-0), ((-0))*(-0)

B(42) = 2.1549E 02 ((1))^(1), ((-0))^(-0), ((-0))^(-0), ((1))^(1), ((-0))^(-0), ((-0))^(-0)

B(43) = 7.1563E 00 ((1))^(1), ((-0))^(-0), ((-0))^(-0), ((-0))^(-0), ((-0))^(-0), ((-0))^(-0)

B(44) = 2.8275E 02 ((-0))^(-0), ((-0))^(-0), ((-0))^(-0), ((1))^(1), ((-0))^(-0), ((-0))^(-0)

TRANSLATION OF TERMS IN POLYNOMIAL

(FIRST THREE TERMS CORRESPOND TO 1ST VARIABLE, SECOND THREE TO 2ND VARIABLE)

((0))= 1

((1))= R OR T

((2))= COS(R OR T)

((3))= SIN(R OR T)

((4))= EXP(R OR T)

((5))= EXP(ABSOLUTE VALUE R OR T)

((6))= LOGN(R OR T)

((7))= LOG(R OR T)

((8))= EXP(-ABSOLUTE VALUE R OR T)

((9 TO 99))= OTHER FUNCTIONS OF R OR T PREPARED BYUSER

COMPARISON OF ACTUAL WITH CALCULATED RANGES

DEP. VAR.(Y)	Y(CALC)	ERROR	DIFFERENCE
4.5170E 01	4.7134E 01	4.3479E-02	1.9639E 00
7.5050E 01	7.6023E 01	1.2963E-02	9.7291E-01
1.4110E 02	1.3630E 02	3.4047E-02	-4.8041E 00
2.0010E 02	1.9571E 02	2.1927E-02	-4.3875E 00
2.1260E 02	2.1871E 02	2.8749E-02	6.1121E 00
2.0900E 02	2.0997E 02	4.6499E-03	9.7182E-01
2.0350E 02	2.0150E 02	9.8371E-03	-2.0019E 00
1.9870E 02	1.9917E 02	2.3574E-03	4.6842E-01
1.9470E 02	1.9496E 02	1.3540E-03	2.6362E-01
1.5050E 01	1.6285E 01	8.2053E-02	1.2349E 00
1.7010E 01	1.7842E 01	4.8916E-02	8.3206E-01

1.9490E 01	1.8944E 01	2.8021E-02	-5.4613E-01
2.2670E 01	2.4351E 01	7.4139E-02	1.6807E 00
2.6730E 01	2.7496E 01	2.8645E-02	7.6567E-01
2.9190E 01	2.8845E 01	1.1833E-02	-3.4541E-01
3.5030E 01	3.2932E 01	5.9887E-02	-2.0978E 00
3.8450E 01	3.6095E 01	6.1236E-02	-2.3545E 00
5.0400E 01	5.0126E 01	5.4289E-03	-2.7362E-01
5.4780E 01	5.5651E 01	1.5898E-02	8.7087E-01
6.3720E 01	6.5674E 01	3.0669E-02	1.9542E 00
7.2440E 01	7.2444E 01	5.1278E-05	3.7146E-03
8.0540E 01	8.2052E 01	1.8773E-02	1.5120E 00
8.7790E 01	8.8574E 01	8.9344E-03	7.8435E-01
1.5830E 01	1.4349E 01	9.3582E-02	-1.4814E 00
1.8150E 01	1.8966E 01	4.4973E-02	8.1626E-01
2.1200E 01	1.9962E 01	5.8388E-02	-1.2378E 00
2.5330E 01	2.6526E 01	4.7217E-02	1.1960E 00
3.1040E 01	3.2035E 01	3.2063E-02	9.9523E-01
3.4680E 01	3.4966E 01	8.2432E-03	2.8587E-01
3.8990E 01	3.8519E 01	1.2076E-02	-4.7083E-01
4.4000E 01	4.2940E 01	2.4087E-02	-1.0598E 00
4.9730E 01	4.8329E 01	2.8172E-02	-1.4010E 00
7.0190E 01	6.9280E 01	1.2959E-02	-9.0962E-01
7.7400E 01	7.7003E 01	5.1346E-03	-3.9742E-01
9.1020E 01	9.1090E 01	7.7306E-04	7.0364E-02
1.0260E 02	1.0101E 02	1.5457E-02	-1.5859E 00
1.1190E 02	1.1281E 02	8.1379E-03	9.1064E-01
1.1910E 02	1.1773E 02	1.1489E-02	-1.3683E 00
1.6220E 01	1.7585E 01	8.4150E-02	1.3649E 00
1.8720E 01	1.9080E 01	1.9207E-02	3.5956E-01
2.2100E 01	1.9215E 01	1.3054E-01	-2.8849E 00
2.6830E 01	2.6522E 01	1.1466E-02	-3.0763E-01
3.3700E 01	3.4036E 01	9.9786E-03	3.3628E-01
3.8310E 01	3.8548E 01	6.2253E-03	2.3849E-01
4.3920E 01	4.4121E 01	4.5712E-03	2.0077E-01
5.0690E 01	5.0954E 01	5.2025E-03	2.6371E-01
5.8640E 01	5.8999E 01	6.1137E-03	3.5851E-01
8.6920E 01	8.7069E 01	1.7183E-03	1.4936E-01
9.6260E 01	9.6376E 01	1.2093E-03	1.1640E-01
1.1230E 02	1.1260E 02	2.6518E-03	2.9780E-01
1.2410E 02	1.2293E 02	9.4501E-03	-1.1728E 00
1.3230E 02	1.3227E 02	1.9575E-04	-2.5898E-02
1.3810E 02	1.3903E 02	6.7236E-03	9.2853E-01
3.5970E 01	3.5072E 01	2.4956E-02	-8.9767E-01
4.1540E 01	4.1061E 01	1.1531E-02	-4.7901E-01
5.7350E 01	5.8431E 01	1.8845E-02	1.0808E 00
6.7930E 01	6.9644E 01	2.5230E-02	1.7139E 00
1.0480E 02	1.0534E 02	5.1108E-03	5.3561E-01
1.1580E 02	1.1569E 02	9.6098E-04	-1.1128E-01

1.3240E 02	1.3251E 02	8.5082E-04	1.1265E-01
1.4270E 02	1.4239E 02	2.1556E-03	-3.0761E-01
1.4890E 02	1.4806E 02	5.6550E-03	-8.4203E-01
1.5280E 02	1.5397E 02	7.6874E-03	1.1746E 00
3.7310E 01	3.5721E 01	4.2578E-02	-1.5886E 00
4.3530E 01	4.2547E 01	2.2593E-02	-9.8347E-01
6.1860E 01	6.3322E 01	2.3634E-02	1.4620E 00
7.4450E 01	7.6819E 01	3.1825E-02	2.3694E 00
1.1720E 02	1.1758E 02	3.2080E-03	3.7598E-01
1.2880E 02	1.2825E 02	4.2489E-03	-5.4726E-01
1.4470E 02	1.4448E 02	1.5172E-03	-2.1954E-01
1.5340E 02	1.5365E 02	1.6507E-03	2.5322E-01
1.5810E 02	1.5711E 02	6.2376E-03	-9.8616E-01
1.6070E 02	1.6139E 02	4.2991E-03	6.9087E-01
3.8840E 01	3.6770E 01	5.3283E-02	-2.0695E 00
4.5880E 01	4.4457E 01	3.1014E-02	-1.4229E 00
6.7630E 01	6.9494E 01	2.7569E-02	1.8645E 00
8.3080E 01	8.5961E 01	3.4683E-02	2.8815E 00
1.0050E 02	1.0315E 02	2.6417E-02	2.6549E 00
1.3310E 02	1.3290E 02	1.5136E-03	-2.0146E-01
1.4480E 02	1.4358E 02	8.4190E-03	-1.2191E 00
1.5880E 02	1.5798E 02	5.1658E-03	-8.2034E-01
1.6500E 02	1.6576E 02	4.6322E-03	7.6431E-01
1.6780E 02	1.6707E 02	4.3619E-03	-7.3192E-01
1.6890E 02	1.6893E 02	1.5419E-04	2.6043E-02
5.0360E 01	4.8943E 01	2.8136E-02	-1.4169E 00
8.0350E 01	8.2640E 01	2.8495E-02	2.2896E 00
1.0300E 02	1.0544E 02	2.3688E-02	2.4398E 00
1.6570E 02	1.6450E 02	7.2681E-03	-1.2043E 00
1.7520E 02	1.7408E 02	6.4207E-03	-1.1249E 00
1.8270E 02	1.8155E 02	6.3206E-03	-1.1548E 00
1.8380E 02	1.8462E 02	4.4615E-03	8.2002E-01
1.8300E 02	1.8340E 02	2.1686E-03	3.9685E-01
1.8160E 02	1.8110E 02	2.7664E-03	-5.0237E-01
4.2680E 01	4.1859E 01	1.9228E-02	-8.2065E-01
5.2210E 01	5.1125E 01	2.0774E-02	-1.0846E 00
8.6410E 01	8.8539E 01	2.4638E-02	2.1290E 00
1.1280E 02	1.1415E 02	1.1962E-02	1.3494E 00
1.4100E 02	1.4005E 02	6.7295E-03	-9.4886E-01
1.7930E 02	1.7826E 02	5.7745E-03	-1.0354E 00
1.8710E 02	1.8701E 02	4.6125E-04	-8.6300E-02
1.9130E 02	1.9042E 02	4.5953E-03	-8.7909E-01
1.9030E 02	1.9070E 02	2.0796E-03	3.9576E-01
1.8820E 02	1.8894E 02	3.9266E-03	7.3898E-01
1.8600E 02	1.8553E 02	2.5116E-03	-4.6716E-01
4.3860E 01	4.4197E 01	7.6948E-03	3.3749E-01
5.4320E 01	5.3776E 01	1.0016E-02	-5.4407E-01
7.0190E 01	7.1030E 01	1.1966E-02	8.3987E-01

9.3990E 01	9.5457E 01	1.5606E-02	1.4668E 00
1.2520E 02	1.2434E 02	6.8543E-03	-8.5816E-01
1.9420E 02	1.9418E 02	8.5899E-05	-1.6682E-02
1.9960E 02	2.0178E 02	1.0918E-02	2.1793E 00
2.0010E 02	1.9988E 02	1.1040E-03	-2.2091E-01
1.9690E 02	1.9640E 02	2.5387E-03	-4.9988E-01
1.9350E 02	1.9428E 02	4.0414E-03	7.8201E-01
1.9030E 02	1.9017E 02	6.7515E-04	-1.2848E-01
5.9600E 01	5.9664E 01	1.0762E-03	6.4140E-02
5.4950E 01	5.4430E 01	9.4691E-03	-5.2033E-01
5.0160E 01	4.7818E 01	4.6695E-02	-2.3422E 00
4.5330E 01	4.2885E 01	5.3948E-02	-2.4455E 00
4.0560E 01	4.1759E 01	2.9552E-02	1.1986E 00
3.5980E 01	3.7053E 01	2.9820E-02	1.0729E 00
3.1700E 01	3.0963E 01	2.3235E-02	-7.3656E-01
2.7820E 01	2.6218E 01	5.7580E-02	-1.6019E 00
2.4410E 01	2.3837E 01	2.3494E-02	-5.7348E-01
2.1470E 01	2.2956E 01	6.9204E-02	1.4858E 00
1.8990E 01	2.0907E 01	1.0097E-01	1.9175E 00
1.6890E 01	1.5362E 01	9.0489E-02	-1.5284E 00
1.5140E 01	1.3315E 01	1.2056E-01	-1.8253E 00
1.3660E 01	1.2327E 01	9.7584E-02	-1.3330E 00
1.2420E 01	1.2577E 01	1.2636E-02	1.5694E-01
7.9180E 01	7.9047E 01	1.6795E-03	-1.3299E-01
7.3270E 01	7.4583E 01	1.7927E-02	1.3135E 00
6.6870E 01	6.8534E 01	2.4890E-02	1.6644E 00
6.0070E 01	6.0952E 01	1.4689E-02	8.8235E-01
5.3080E 01	5.6488E 01	6.4200E-02	3.4077E 00
4.6160E 01	4.8519E 01	5.1106E-02	2.3590E 00
3.9640E 01	3.9467E 01	4.3595E-03	-1.7281E-01
3.3770E 01	3.2202E 01	4.6437E-02	-1.5682E 00
2.8750E 01	2.7864E 01	3.0804E-02	-8.8562E-01
2.4580E 01	2.5714E 01	4.6130E-02	1.1339E 00
2.1220E 01	2.3221E 01	9.4315E-02	2.0014E 00
1.8500E 01	1.8205E 01	1.5961E-02	-2.9528E-01
1.6310E 01	1.7515E 01	7.3865E-02	1.2047E 00
1.4540E 01	1.6886E 01	1.6136E-01	2.3462E 00
1.3080E 01	1.2800E 01	2.1415E-02	-2.8011E-01
1.1460E 02	1.1468E 02	7.2632E-04	8.3236E-02
1.0850E 02	1.0623E 02	2.0959E-02	-2.2741E 00
1.0090E 02	1.0205E 02	1.1365E-02	1.1467E 00
9.1740E 01	9.0327E 01	1.5402E-02	-1.4130E 00
8.0880E 01	8.1112E 01	2.8685E-03	2.3201E-01
6.8850E 01	6.8388E 01	6.7128E-03	-4.6217E-01
5.6660E 01	5.4831E 01	3.2279E-02	-1.8289E 00
4.5590E 01	4.3438E 01	4.7211E-02	-2.1523E 00
3.6520E 01	3.5527E 01	2.7179E-02	-9.9256E-01
2.9590E 01	3.0524E 01	3.1559E-02	9.3383E-01

2.4460E 01	2.5979E 01	6.2087E-02	1.5186E 00
2.0670E 01	1.9801E 01	4.2060E-02	-8.6937E-01
1.7800E 01	1.8427E 01	3.5245E-02	6.2736E-01
1.5590E 01	1.3550E 01	1.3087E-01	-2.0403E 00
1.3850E 01	1.3959E 01	7.8999E-03	1.0941E-01
1.7730E 02	1.7687E 02	2.4243E-03	-4.2983E-01
1.7780E 02	1.7783E 02	1.8869E-04	3.3548E-02
1.7740E 02	1.7835E 02	5.3761E-03	9.5372E-01
1.7440E 02	1.7321E 02	6.8473E-03	-1.1942E 00
1.6420E 02	1.6268E 02	9.2763E-03	-1.5232E 00
1.3750E 02	1.3775E 02	1.8148E-03	2.4954E-01
9.5060E 01	9.7972E 01	3.0629E-02	2.9116E 00
5.9940E 01	6.0180E 01	4.0101E-03	2.4036E-01
4.0600E 01	3.8629E 01	4.8543E-02	-1.9709E 00
3.0170E 01	3.0441E 01	8.9715E-03	2.7067E-01
2.3920E 01	2.4164E 01	1.0207E-02	2.4416E-01
1.9810E 01	2.0556E 01	3.7655E-02	7.4595E-01
1.6910E 01	1.6539E 01	2.1960E-02	-3.7134E-01
1.4760E 01	1.4952E 01	1.3027E-02	1.9227E-01

RELATIVE ERRORS--ACCURACY OF FIT

AVG. RELATIVE ERROR = 2.3981E-02

AVG. DIFF(DEGREES) = 1.0770E 00

AVG. SUM OF SQUARES = 1.9872E 00

STANDARD DEVIATION = 1.4097E 00

PROBABLE ERROR = 9.5084E-01

TERM EVALUATION

TERM(1) = 7.2851E 03
 TERM(2) = 1.2545E 06
 TERM(3) = 1.0247E 04
 TERM(4) = 1.1026E 04
 TERM(5) = 6.7994E 03
 TERM(6) = 7.4665E 04
 TERM(7) = 1.3140E 04
 TERM(8) = 1.3738E 04
 TERM(9) = 1.5921E 05
 TERM(10) = 4.9342E 05
 TERM(11) = 1.3010E 04
 TERM(12) = 1.4997E 05

TERM(13) = 4.2760E 05
TERM(14) = 1.0833E 06
TERM(15) = 9.9740E 04
TERM(16) = 6.5470E 05
TERM(17) = 1.0836E 06
TERM(18) = 8.0995E 05
TERM(19) = 1.1108E 04
TERM(20) = 1.5683E 05
TERM(21) = 2.1297E 04
TERM(22) = 2.5046E 04
TERM(23) = 4.8460E 06
TERM(24) = 4.3983E 05
TERM(25) = 1.1398E 04
TERM(26) = 9.8948E 03
TERM(27) = 5.3622E 04
TERM(28) = 1.5329E 04
TERM(29) = 2.8359E 05
TERM(30) = 1.1567E 06
TERM(31) = 8.6460E 04
TERM(32) = 6.0356E 03
TERM(33) = 4.4882E 03
TERM(34) = 1.1210E 04
TERM(35) = 1.0975E 04
TERM(36) = 9.7099E 03
TERM(37) = 1.9078E 04
TERM(38) = 1.2257E 04
TERM(39) = 1.1248E 04
TERM(40) = 9.5460E 03
TERM(41) = 8.6901E 03
TERM(42) = 8.8506E 03
TERM(43) = 9.7230E 03
TERM(44) = 8.6351E 03

APPENDIX III

DERIVATION OF BASIC SPHERICAL TRIGONOMETRY FORMULAS USED IN AREA CALCULATION

Consider three unit vectors, $\hat{i}$, $\hat{j}$ and $\hat{k}$, not mutually perpendicular, that emanate from a point (see Figure III-1). Now let these cut a sphere of unit diameter to form a spherical triangle. Then we may write that:

$$(\hat{j} \times \hat{i}) = \sin \alpha \quad (III-1)$$

$$(\hat{i} \times \hat{k}) = \sin \beta \quad (III-2)$$

$$(\hat{j} \times \hat{i}) \cdot (\hat{i} \times \hat{k}) = \sin \alpha \sin \beta \cos C \quad (III-3)$$

Similarly, we note that:

$$\begin{aligned} (\hat{j} \times \hat{i}) \cdot (\hat{k} \times \hat{i}) &= \hat{j} \cdot [\hat{i} \times (\hat{k} \times \hat{i})] = \hat{j} \cdot [(\hat{i} \cdot \hat{i}) \hat{k} - (\hat{i} \cdot \hat{k}) \hat{i}] \\ \hat{j} \cdot [(\hat{i} \cdot \hat{i}) \hat{k} - (\hat{i} \cdot \hat{k}) \hat{i}] &= \hat{j} \cdot \hat{k} - (\hat{i} \cdot \hat{k}) (\hat{i} \cdot \hat{j}) \end{aligned} \quad (III-4)$$

Now, Equation (III-5) becomes:

$$\hat{j} \cdot \hat{k} - (\hat{i} \cdot \hat{k})(\hat{i} \cdot \hat{j}) = \cos \gamma - \cos \beta \cos \alpha \quad (III-6)$$

Equations (III-3) and (III-6) are equivalent, hence:

$$\cos \gamma - \cos \beta \cos \alpha = \sin \alpha \sin \beta \cos C \quad (III-7A)$$

Equation (III-7A) then becomes, on rearranging:

$$\cos \gamma = \cos \beta \cos \alpha + \sin \beta \sin \alpha \cos C \quad (III-7B)$$

which is the basic formula used in both the earth area and error measurement analyses.

PICTORIAL OF VECTOR RELATIONSHIPS FOR SPHERICAL TRIGONOMETRY

FIGURE III-1

APPENDIX IV DETERMINATION OF AREA

Consider the triangle formed on the surface of any sphere by choosing any three points P_1 , P_2 and P_3 and joining these points with arcs of great circles. For such a triangle, the following

relations hold (see Appendix III):

$$\left. \begin{aligned} \cos \alpha' &= \cos \beta' \cos \gamma' + \sin \beta' \sin \gamma' \cos \alpha \\ \cos \beta' &= \cos \alpha' \cos \gamma' + \sin \alpha' \sin \gamma' \cos \beta \\ \cos \gamma' &= \cos \alpha' \cos \beta' + \sin \alpha' \sin \beta' \cos \gamma \end{aligned} \right\} \quad (\text{IV-1})$$

$$\text{Area of the triangle} = (\alpha + \beta + \gamma - \pi) R^2 \quad (\text{IV-2})$$

where all angles are measured in radians and R is the radius of the sphere.

Now let P_1 , P_2 , and P_3 represent the points on the earth's surface at which particles of destruct debris with destruct parameters $(V_D, \epsilon, \zeta)_A$, $(V_D, \epsilon, \zeta)_B$, and $(V_D, \epsilon, \zeta)_C$, respectively, will land. Then the area of this triangle (formed by joining P_1 , P_2 and P_3 with arcs of great circles) represents the area, $AREA$, over which any destruct particles having destruct parameters that lie within the pyramid defined by $(V_D, \epsilon, \zeta)_A$, $(V_D, \epsilon, \zeta)_B$, and $(V_D, \epsilon, \zeta)_C$ in the "destruct sphere" will be spread.

Refer to Figures 2.1 and 5.2. Recognize that $P-P_1-P_3$, $P-P_1-P_2$, and $P-P_2-P_3$ also represent spherical triangles on the surface of the earth. Then $AREA$ can be determined mathematically. Using the set of relations (IV-1),

$$\left. \begin{aligned} \cos a &= \cos \lambda_B \cos \lambda_C + \sin \lambda_B \sin \lambda_C \cos (\nu_C - \nu_B) \\ \cos b &= \cos \lambda_A \cos \lambda_C + \sin \lambda_A \sin \lambda_C \cos (\nu_C - \nu_A) \\ \cos c &= \cos \lambda_A \cos \lambda_B + \sin \lambda_A \sin \lambda_B \cos (\nu_A - \nu_B) \end{aligned} \right\} \quad (\text{IV-3})$$

Then evaluate $\sin a$, $\sin b$, and $\sin c$ using the identity,

$$\sin \mu = (1 - \cos^2 \mu)^{1/2}$$

Again make use of (IV-1),

$$\left. \begin{aligned} \cos A &= \frac{\cos a - \cos b \cos c}{\sin b \sin c} \\ \cos B &= \frac{\cos b - \cos a \cos c}{\sin a \sin c} \end{aligned} \right\} \quad (\text{IV-4})$$

$$\cos C = \frac{\cos c - \cos a \cos b}{\sin a \sin b} \quad (\text{IV-4})$$

Then

$$\left. \begin{aligned} A &= \cos^{-1}(\cos A) \\ B &= \cos^{-1}(\cos B) \\ C &= \cos^{-1}(\cos C) \end{aligned} \right\} \quad (\text{IV-5})$$

And

$$\text{AREA} = (A + B + C - \pi) R^2 \quad (\text{IV-6})$$

DETERMINATION OF ERROR COEFFICIENTS

In the following derivation, the notation and variable names used correspond as closely to those appearing in the program listings (Appendix V) as is consistent with the text of this report. Deviations between listings and text will be noted.

$$AREA = (A + B + C - \pi) R^2$$

(In the listings, $A \equiv \text{ANGLE}(1)$, $B \equiv \text{ANGLE}(2)$, and $C \equiv \text{ANGLE}(3)$)

$$\therefore d(AREA) = d[(A + B + C - \pi) R^2] = R^2 [dA + dB + dC]$$

Since A , B , and C are functions of a , b , and c , it is possible to write

$$\left. \begin{aligned}
 dA &= \frac{\partial A}{\partial a} da + \frac{\partial A}{\partial b} db + \frac{\partial A}{\partial c} dc \\
 dB &= \frac{\partial B}{\partial a} da + \frac{\partial B}{\partial b} db + \frac{\partial B}{\partial c} dc \\
 dC &= \frac{\partial C}{\partial a} da + \frac{\partial C}{\partial b} db + \frac{\partial C}{\partial c} dc
 \end{aligned} \right\} \text{IV-7}$$

From which,

$$\begin{aligned}
 \frac{1}{R^2} d(AREA) &= \left(\frac{\partial A}{\partial a} + \frac{\partial B}{\partial a} + \frac{\partial C}{\partial a} \right) da + \left(\frac{\partial A}{\partial b} + \frac{\partial B}{\partial b} + \frac{\partial C}{\partial b} \right) db \\
 &\quad + \left(\frac{\partial A}{\partial c} + \frac{\partial B}{\partial c} + \frac{\partial C}{\partial c} \right) dc
 \end{aligned}$$

$$\begin{array}{lcl}
 \text{Now define} & GA(1) = \frac{\partial A}{\partial a} + \frac{\partial B}{\partial a} + \frac{\partial C}{\partial a} & \\
 & GA(2) = \frac{\partial A}{\partial b} + \frac{\partial B}{\partial b} + \frac{\partial C}{\partial b} & \\
 & GA(3) = \frac{\partial A}{\partial c} + \frac{\partial B}{\partial c} + \frac{\partial C}{\partial c} &
 \end{array} \quad \left. \vphantom{\begin{array}{l} GA(1) \\ GA(2) \\ GA(3) \end{array}} \right\} \text{IV-8}$$

$$\text{Then } \left(\frac{1}{R^2}\right) d(\text{AREA}) = GA(1) da + GA(2) db + GA(3) dc \quad \text{IV-9}$$

Recognizing that $a, b,$ and c are functions of $\lambda_A, \lambda_B, \lambda_C,$

$v_A, v_B,$ and $v_C,$ it is possible to write

$$\begin{array}{lcl}
 da = \frac{\partial a}{\partial \lambda_A} d\lambda_A + \frac{\partial a}{\partial \lambda_B} d\lambda_B + \frac{\partial a}{\partial \lambda_C} d\lambda_C + \frac{\partial a}{\partial v_A} dv_A + \frac{\partial a}{\partial v_B} dv_B + \frac{\partial a}{\partial v_C} dv_C & & \\
 db = \frac{\partial b}{\partial \lambda_A} d\lambda_A + \frac{\partial b}{\partial \lambda_B} d\lambda_B + \frac{\partial b}{\partial \lambda_C} d\lambda_C + \frac{\partial b}{\partial v_A} dv_A + \frac{\partial b}{\partial v_B} dv_B + \frac{\partial b}{\partial v_C} dv_C & & \\
 dc = \frac{\partial c}{\partial \lambda_A} d\lambda_A + \frac{\partial c}{\partial \lambda_B} d\lambda_B + \frac{\partial c}{\partial \lambda_C} d\lambda_C + \frac{\partial c}{\partial v_A} dv_A + \frac{\partial c}{\partial v_B} dv_B + \frac{\partial c}{\partial v_C} dv_C & &
 \end{array} \quad \left. \vphantom{\begin{array}{l} da \\ db \\ dc \end{array}} \right\} \text{IV-10}$$

Now define the W-array as follows:

$$\begin{array}{lll}
 W_{11} = \frac{\partial a}{\partial \lambda_A} & W_{21} = \frac{\partial b}{\partial \lambda_A} & W_{31} = \frac{\partial c}{\partial \lambda_A} \\
 W_{12} = \frac{\partial a}{\partial \lambda_B} & W_{22} = \frac{\partial b}{\partial \lambda_B} & W_{32} = \frac{\partial c}{\partial \lambda_B} \\
 W_{13} = \frac{\partial a}{\partial \lambda_C} & W_{23} = \frac{\partial b}{\partial \lambda_C} & W_{33} = \frac{\partial c}{\partial \lambda_C} \\
 W_{14} = \frac{\partial a}{\partial v_A} & W_{24} = \frac{\partial b}{\partial v_A} & W_{34} = \frac{\partial c}{\partial v_A} \\
 W_{15} = \frac{\partial a}{\partial v_B} & W_{25} = \frac{\partial b}{\partial v_B} & W_{35} = \frac{\partial c}{\partial v_B} \\
 W_{16} = \frac{\partial a}{\partial v_C} & W_{26} = \frac{\partial b}{\partial v_C} & W_{36} = \frac{\partial c}{\partial v_C}
 \end{array}$$

Using the set of equations IV-3,

$$a = \cos^{-1} [\cos \lambda_B \cos \lambda_C + \sin \lambda_B \sin \lambda_C \cos (v_C - v_B)]$$

$$b = \cos^{-1} [\cos \lambda_A \cos \lambda_c + \sin \lambda_A \sin \lambda_c \cos (\nu_c - \nu_A)]$$

$$c = \cos^{-1} [\cos \lambda_A \cos \lambda_B + \sin \lambda_A \sin \lambda_B \cos (\nu_B - \nu_A)]$$

from which the values in the W-array are calculated to be

$$W_{11} = 0$$

$$W_{12} = -\frac{1}{(1 - \cos^2 a)^{1/2}} \left\{ -\sin \lambda_B \cos \lambda_c + \cos \lambda_B \sin \lambda_c \cos (\nu_c - \nu_B) \right\}$$

$$W_{13} = -\frac{1}{(1 - \cos^2 a)^{1/2}} \left\{ -\cos \lambda_B \sin \lambda_c + \sin \lambda_B \cos \lambda_c \cos (\nu_c - \nu_B) \right\}$$

$$W_{14} = 0$$

$$W_{15} = -\frac{1}{(1 - \cos^2 a)^{1/2}} \left\{ \sin \lambda_B \sin \lambda_c \sin (\nu_c - \nu_B) \right\}$$

$$W_{16} = -W_{15}$$

Note that $(1 - \cos^2 a)^{1/2} = \sin a$. Similarly, $(1 - \cos^2 b)^{1/2} = \sin b$ and $(1 - \cos^2 c)^{1/2} = \sin c$.

$$W_{21} = -\frac{1}{\sin b} \left\{ -\cos \lambda_A \sin \lambda_c + \sin \lambda_A \cos \lambda_c \cos (\nu_c - \nu_A) \right\}$$

$$W_{22} = 0$$

$$W_{23} = -\frac{1}{\sin b} \left\{ -\cos \lambda_A \sin \lambda_c + \sin \lambda_A \cos \lambda_c \cos (\nu_c - \nu_A) \right\}$$

$$W_{24} = -\frac{1}{\sin b} \left\{ \sin \lambda_A \sin \lambda_c \sin (\nu_c - \nu_A) \right\}$$

$$W_{25} = 0$$

$$W_{26} = -W_{24}$$

$$W_{31} = -\frac{1}{\sin c} \left\{ -\sin \lambda_A \cos \lambda_B + \cos \lambda_A \sin \lambda_B \cos(\psi_A - \psi_B) \right\}$$

$$W_{32} = -\frac{1}{\sin c} \left\{ -\cos \lambda_A \sin \lambda_B + \sin \lambda_A \cos \lambda_B \cos(\psi_A - \psi_B) \right\}$$

$$W_{33} = 0$$

$$W_{34} = -\frac{1}{\sin c} \left\{ -\sin \lambda_A \sin \lambda_B \sin(\psi_A - \psi_B) \right\}$$

$$W_{35} = -W_{34}$$

$$W_{36} = 0$$

Using the set of equations IV-4, the GA's can be calculated as follows.

$$\begin{aligned}
 \text{First consider } \frac{1}{(1 - \cos^2 A)^{\frac{1}{2}}} &= \frac{1}{\sin A} \\
 &= \frac{\sin b \sin c}{(\sin^2 b \sin^2 c - \cos^2 a - \cos^2 b \cos^2 c + 2 \cos a \cos b \cos c)^{\frac{1}{2}}}
 \end{aligned}$$

Call the denominator of this fraction DENOM and set $\sin^2 c = 1 - \cos^2 c$. Then,

$$\begin{aligned}
 \text{DENOM} &= (\sin^2 b - \sin^2 b \cos^2 c - \cos^2 a - \cos^2 c \cos^2 b + 2 \cos a \cos b \cos c)^{\frac{1}{2}} \\
 &= (\sin^2 b - \cos^2 a - \cos^2 c + 2 \cos a \cos b \cos c)^{\frac{1}{2}}
 \end{aligned}$$

and

$$\frac{1}{\sin A} = \frac{\sin b \sin c}{\text{DENOM}}$$

It can be shown in similar fashion that

$$\frac{1}{\sin B} = \frac{\sin a \sin c}{\text{DENOM}} \quad \text{and} \quad \frac{1}{\sin C} = \frac{\sin a \sin b}{\text{DENOM}}$$

Using IV-4 ,

$$\begin{aligned} \frac{\partial A}{\partial a} &= - \frac{1}{\sin A} \left\{ - \frac{\sin a}{\sin b \sin c} \right\} = \left\{ \frac{\sin b \sin c}{\text{DENOM}} \right\} \left\{ \frac{\sin a}{\sin b \sin c} \right\} \\ &= \frac{\sin a}{\text{DENOM}} \end{aligned}$$

$$\begin{aligned} \frac{\partial B}{\partial a} &= - \frac{1}{\sin B} \left\{ \frac{(\sin a \sin c)(\sin a \cos c) - (\cos b - \cos a \cos c)(\cos a \sin c)}{\sin^2 a \sin^2 c} \right\} \\ &= \frac{\sin a \sin c}{\text{DENOM}} \left\{ \frac{\cos b \cos a - \cos^2 a \cos c - \sin^2 a \cos c}{\sin^2 a \sin c} \right\} \\ &= \frac{\cos b \cos a - \cos c}{(\sin a)(\text{DENOM})} \end{aligned}$$

$$\begin{aligned} \frac{\partial C}{\partial a} &= - \frac{1}{\sin C} \left\{ \frac{(\sin a \sin b)(\sin a \cos b) - (\cos c - \cos a \cos b)(\cos a \sin b)}{\sin^2 a \sin^2 b} \right\} \\ &= \frac{\cos a \cos c - \cos b}{(\sin a)(\text{DENOM})} \end{aligned}$$

Then

$$\begin{aligned} GA(1) &= \frac{\partial A}{\partial a} + \frac{\partial B}{\partial a} + \frac{\partial C}{\partial a} \\ &= \frac{1}{\text{DENOM}} \left\{ \sin a + \frac{\cos a \cos b - \cos c}{\sin a} + \frac{\cos a \cos c - \cos b}{\sin a} \right\} \end{aligned}$$

$$\therefore GA(1) = \frac{\sin^2 a - (1 - \cos a)(\cos b + \cos c)}{(\sin a)(\text{DENOM})} \quad \text{IV-11}$$

Similarly,

$$\left. \begin{aligned}
 GA(2) &= \frac{\sin^2 b - (1 - \cos b)(\cos a + \cos c)}{(\sin b)(\text{DENOM})} \\
 GA(3) &= \frac{\sin^2 c - (1 - \cos c)(\cos a + \cos b)}{(\sin c)(\text{DENOM})}
 \end{aligned} \right\} \begin{array}{l} \text{IV-11} \\ (\text{cont'd.}) \end{array}$$

Thus it is possible to compute the components of the GA-array and the W-array from the λ and v values for the three points which determine the (spherical triangular) contaminated area on the earth's surface.

To relate the error coefficients to the destruct parameters, V_0 , ζ , and ϵ , it is necessary to express $d\lambda$ and dv in terms of dV_0 , $d\zeta$, and $d\epsilon$.

For λ , a polynomial of the following form has been assumed:

$$\lambda = \sum_{m=1}^M \sum_{n=1}^N \beta_{mn} (\alpha^*)^{m-1} (v^*)^{n-1} \quad \text{IV-12}$$

where α^* is the transformed value of α
 v^* is the transformed value of v

and the coefficients β_{mn} are determined by NEFOD.

α represents the ^{resultant} pre-entry angle (referred to in the text as ψ) and v the resultant velocity (in the text, V_R) of the re-entering destruct particle.

Now define λ_{mn} such that $\lambda = \sum_{m=1}^M \sum_{n=1}^N \lambda_{mn}$, i.e.,

$$\lambda_{mn} = \beta_{mn} (\alpha^*)^{m-1} (v^*)^{n-1}$$

Then

$$d\lambda = d\left(\sum_{m=1}^M \sum_{n=1}^N \lambda_{mn}\right) = \sum_{m=1}^M \sum_{n=1}^N d\lambda_{mn}$$

$$d\lambda_{mn} = \left(\frac{\partial \lambda_{mn}}{\partial \alpha}\right) d\alpha + \left(\frac{\partial \lambda_{mn}}{\partial v}\right) dv$$

$$= \beta_{mn} \left\{ (m-1) (\alpha^*)^{m-2} (v^*)^{n-1} \left(\frac{\partial \alpha^*}{\partial \alpha}\right) d\alpha + (n-1) (\alpha^*)^{m-1} (v^*)^{n-2} \left(\frac{\partial v^*}{\partial v}\right) dv \right\}$$

$$= \beta_{mn} (\alpha^*)^{m-2} (v^*)^{n-2} \left\{ (m-1) (v^*) \left(\frac{\partial \alpha^*}{\partial \alpha}\right) d\alpha + (n-1) (\alpha^*) \left(\frac{\partial v^*}{\partial v}\right) dv \right\}$$

Now, both α and v are functions of V_D , ϵ , and ξ , so that $d\alpha$ and dv can be expressed as

$$d\alpha = \left(\frac{\partial \alpha}{\partial V_D}\right) dV_D + \left(\frac{\partial \alpha}{\partial \epsilon}\right) d\epsilon + \left(\frac{\partial \alpha}{\partial \xi}\right) d\xi$$

$$dv = \left(\frac{\partial v}{\partial V_D}\right) dV_D + \left(\frac{\partial v}{\partial \epsilon}\right) d\epsilon + \left(\frac{\partial v}{\partial \xi}\right) d\xi,$$

so that

$$d\lambda_{mn} = \beta_{mn} (\alpha^*)^{m-2} (v^*)^{n-2} \left\{ (m-1) (v^*) \left(\frac{\partial \alpha^*}{\partial \alpha}\right) \left(\frac{\partial \alpha}{\partial V_D} dV_D + \frac{\partial \alpha}{\partial \epsilon} d\epsilon + \frac{\partial \alpha}{\partial \xi} d\xi \right) + (n-1) (\alpha^*) \left(\frac{\partial v^*}{\partial v}\right) \left(\frac{\partial v}{\partial V_D} dV_D + \frac{\partial v}{\partial \epsilon} d\epsilon + \frac{\partial v}{\partial \xi} d\xi \right) \right\}$$

$$\therefore d\lambda_{mn} = \beta_{mn}(\alpha^*)^{m-2}(\nu^*)^{n-2} \left\{ \begin{aligned} &[(m-1)(\nu^*)\left(\frac{\partial \alpha^*}{\partial x}\right)\left(\frac{\partial \nu^*}{\partial v_0}\right) + (m-1)(\alpha^*)\left(\frac{\partial \nu^*}{\partial \nu}\right)\left(\frac{\partial \nu^*}{\partial v_0}\right)] dv_0 \\ &+ [(m-1)(\nu^*)\left(\frac{\partial \alpha^*}{\partial x}\right)\left(\frac{\partial \nu^*}{\partial \epsilon}\right) + (m-1)(\alpha^*)\left(\frac{\partial \nu^*}{\partial \nu}\right)\left(\frac{\partial \nu^*}{\partial \epsilon}\right)] d\epsilon \\ &+ [(m-1)(\nu^*)\left(\frac{\partial \alpha^*}{\partial x}\right)\left(\frac{\partial \nu^*}{\partial \zeta}\right) + (m-1)(\alpha^*)\left(\frac{\partial \nu^*}{\partial \nu}\right)\left(\frac{\partial \nu^*}{\partial \zeta}\right)] d\zeta \end{aligned} \right\}$$

Define:

$$\begin{aligned} F_{mn} &= \beta_{mn}(\alpha^*)^{m-2}(\nu^*)^{n-2} \left[(m-1)(\nu^*)\left(\frac{\partial \alpha^*}{\partial x}\right)\left(\frac{\partial \nu^*}{\partial v_0}\right) + (m-1)(\alpha^*)\left(\frac{\partial \nu^*}{\partial \nu}\right)\left(\frac{\partial \nu^*}{\partial v_0}\right) \right] \\ G_{mn} &= \beta_{mn}(\alpha^*)^{m-2}(\nu^*)^{n-2} \left[(m-1)(\nu^*)\left(\frac{\partial \alpha^*}{\partial x}\right)\left(\frac{\partial \nu^*}{\partial \epsilon}\right) + (m-1)(\alpha^*)\left(\frac{\partial \nu^*}{\partial \nu}\right)\left(\frac{\partial \nu^*}{\partial \epsilon}\right) \right] \\ H_{mn} &= \beta_{mn}(\alpha^*)^{m-2}(\nu^*)^{n-2} \left[(m-1)(\nu^*)\left(\frac{\partial \alpha^*}{\partial x}\right)\left(\frac{\partial \nu^*}{\partial \zeta}\right) + (m-1)(\alpha^*)\left(\frac{\partial \nu^*}{\partial \nu}\right)\left(\frac{\partial \nu^*}{\partial \zeta}\right) \right] \end{aligned}$$

Then $d\lambda_{mn} = F_{mn} dv_0 + G_{mn} d\epsilon + H_{mn} d\zeta$

And

$$d\lambda = \left[\sum_{m=1}^M \sum_{n=1}^N F_{mn} \right] dv_0 + \left[\sum_{m=1}^M \sum_{n=1}^N G_{mn} \right] d\epsilon + \left[\sum_{m=1}^M \sum_{n=1}^N H_{mn} \right] d\zeta \quad \text{IV-13}$$

Since v is known explicitly as a function of v_0 , ϵ , and ζ , it is possible to write:

$$dv = \left[\frac{\partial v}{\partial v_0} \right] dv_0 + \left[\frac{\partial v}{\partial \epsilon} \right] d\epsilon + \left[\frac{\partial v}{\partial \zeta} \right] d\zeta \quad \text{IV-14}$$

Thus for each of the points defining the spherical triangle, it is possible to compute values for each of the bracketed quantities in IV-13 and IV-14. In the remainder of the discussion, these quantities are identified with the desired point by a subscripted A, B, or C.

At this point it is convenient to define of another array, the X-array, as follows:

Using λ_A, ν_A	Using λ_B, ν_B	Using λ_C, ν_C
$X_{11} = \left[\sum_{m=1}^M \sum_{n=1}^N F_{mn} \right]_A$	$X_{21} = \left[\sum_{m=1}^M \sum_{n=1}^N F_{mn} \right]_B$	$X_{31} = \left[\sum_{m=1}^M \sum_{n=1}^N F_{mn} \right]_C$
$X_{12} = \left[\sum_{m=1}^M \sum_{n=1}^N G_{mn} \right]_A$	$X_{22} = \left[\sum_{m=1}^M \sum_{n=1}^N G_{mn} \right]_B$	$X_{32} = \left[\sum_{m=1}^M \sum_{n=1}^N G_{mn} \right]_C$
$X_{13} = \left[\sum_{m=1}^M \sum_{n=1}^N H_{mn} \right]_A$	$X_{23} = \left[\sum_{m=1}^M \sum_{n=1}^N H_{mn} \right]_B$	$X_{33} = \left[\sum_{m=1}^M \sum_{n=1}^N H_{mn} \right]_C$
$X_{41} = \frac{\partial \nu_A}{\partial \nu_D}$	$X_{51} = \frac{\partial \nu_B}{\partial \nu_D}$	$X_{61} = \frac{\partial \nu_C}{\partial \nu_D}$
$X_{42} = \frac{\partial \nu_A}{\partial \epsilon}$	$X_{52} = \frac{\partial \nu_B}{\partial \epsilon}$	$X_{62} = \frac{\partial \nu_C}{\partial \epsilon}$
$X_{43} = \frac{\partial \nu_A}{\partial \zeta}$	$X_{53} = \frac{\partial \nu_B}{\partial \zeta}$	$X_{63} = \frac{\partial \nu_C}{\partial \zeta}$

It is reasonable to assume that the error in the measurement of the destruct parameters will not differ for the three points (though the destruct parameters themselves must differ), so that

$$d\nu_D = (d\nu_D)_A = (d\nu_D)_B = (d\nu_D)_C$$

$$d\epsilon = (d\epsilon)_A = (d\epsilon)_B = (d\epsilon)_C$$

$$d\zeta = (d\zeta)_A = (d\zeta)_B = (d\zeta)_C$$

Now it is possible to draw together the elements of the analysis into the form actually used to compute the error coefficients.

Begin by inserting the values from the W-array into the set of equations IV-10.

$$\left. \begin{aligned}
 da &= W_{11}d\lambda_A + W_{12}d\lambda_B + W_{13}d\lambda_C + W_{14}dv_A + W_{15}dv_B + W_{16}dv_C \\
 db &= W_{21}d\lambda_A + W_{22}d\lambda_B + W_{23}d\lambda_C + W_{24}dv_A + W_{25}dv_B + W_{26}dv_C \\
 dc &= W_{31}d\lambda_A + W_{32}d\lambda_B + W_{33}d\lambda_C + W_{34}dv_A + W_{35}dv_B + W_{36}dv_C
 \end{aligned} \right\} \text{IV-15}$$

Now, making use of equations IV-15 and of the GA- and X-arrays, equation IV-9 becomes

$$\begin{aligned}
 \left(\frac{1}{R^2}\right)d(\text{AREA}) = & GA(1) \left\{ W_{11}(X_{11}dv_B + X_{12}d\epsilon + X_{13}d\zeta) + W_{12}(X_{21}dv_B + X_{22}d\epsilon + X_{23}d\zeta) \right. \\
 & + W_{13}(X_{31}dv_B + X_{32}d\epsilon + X_{33}d\zeta) + W_{14}(X_{41}dv_B + X_{42}d\epsilon + X_{43}d\zeta) \\
 & \left. + W_{15}(X_{51}dv_B + X_{52}d\epsilon + X_{53}d\zeta) + W_{16}(X_{61}dv_B + X_{62}d\epsilon + X_{63}d\zeta) \right\} \\
 & + GA(2) \left\{ W_{21}(X_{11}dv_B + X_{12}d\epsilon + X_{13}d\zeta) + W_{22}(X_{21}dv_B + X_{22}d\epsilon + X_{23}d\zeta) \right. \\
 & + W_{23}(X_{31}dv_B + X_{32}d\epsilon + X_{33}d\zeta) + W_{24}(X_{41}dv_B + X_{42}d\epsilon + X_{43}d\zeta) \\
 & + W_{25}(X_{51}dv_B + X_{52}d\epsilon + X_{53}d\zeta) + W_{26}(X_{61}dv_B + X_{62}d\epsilon + X_{63}d\zeta) \left. \right\} \\
 & + GA(3) \left\{ W_{31}(X_{11}dv_B + X_{12}d\epsilon + X_{13}d\zeta) + W_{32}(X_{21}dv_B + X_{22}d\epsilon + X_{23}d\zeta) \right. \\
 & + W_{33}(X_{31}dv_B + X_{32}d\epsilon + X_{33}d\zeta) + W_{34}(X_{41}dv_B + X_{42}d\epsilon + X_{43}d\zeta) \\
 & \left. + W_{35}(X_{51}dv_B + X_{52}d\epsilon + X_{53}d\zeta) + W_{36}(X_{61}dv_B + X_{62}d\epsilon + X_{63}d\zeta) \right\}
 \end{aligned}$$

which re-arranges to

$$\begin{aligned}
 \left(\frac{1}{R^2}\right)d(\text{AREA}) = & \left[X_{11} \sum_{i=1}^3 GA(i)W_{i1} + X_{21} \sum_{i=1}^3 GA(i)W_{i2} + X_{31} \sum_{i=1}^3 GA(i)W_{i3} \right. \\
 & \left. + X_{41} \sum_{i=1}^3 GA(i)W_{i4} + X_{51} \sum_{i=1}^3 GA(i)W_{i5} + X_{61} \sum_{i=1}^3 GA(i)W_{i6} \right] dv_B
 \end{aligned}$$

$$\begin{aligned}
 & + \left[X_{12} \sum_{i=1}^3 GA(i) W_{i1} + X_{22} \sum_{i=1}^3 GA(i) W_{i2} + X_{32} \sum_{i=1}^3 GA(i) W_{i3} \right. \\
 & \quad \left. + X_{42} \sum_{i=1}^3 GA(i) W_{i4} + X_{52} \sum_{i=1}^3 GA(i) W_{i5} + X_{62} \sum_{i=1}^3 GA(i) W_{i6} \right] d\epsilon \quad \text{IV-16} \\
 & + \left[X_{13} \sum_{i=1}^3 GA(i) W_{i1} + X_{23} \sum_{i=1}^3 GA(i) W_{i2} + X_{33} \sum_{i=1}^3 GA(i) W_{i3} \right. \\
 & \quad \left. + X_{43} \sum_{i=1}^3 GA(i) W_{i4} + X_{53} \sum_{i=1}^3 GA(i) W_{i5} + X_{63} \sum_{i=1}^3 GA(i) W_{i6} \right] d\zeta
 \end{aligned}$$

Equation IV-16 can be much more simply expressed if one more array, the γ -array, is defined such that

$$\gamma_j = \sum_{i=1}^3 GA(i) W_{ij} \quad , \quad j = 1, 2, \dots, 6.$$

Then IV-16 becomes

$$\begin{aligned}
 d(\text{AREA}) = & \left[R^2 \sum_{j=1}^6 (X_{j1} \gamma_j) \right] dV_0 + \left[R^2 \sum_{j=1}^6 (X_{j2} \gamma_j) \right] d\epsilon \\
 & + \left[R^2 \sum_{j=1}^6 (X_{j3} \gamma_j) \right] d\zeta . \quad \text{IV-17}
 \end{aligned}$$

The error coefficients X , γ , and Z (text notation) are assigned the names A_1 , A_2 , and A_3 , respectively, in the program listings. Their values are

$$\left. \begin{aligned}
 X &= R^2 \sum_{j=1}^6 (X_{j1} \gamma_j) \\
 \gamma &= R^2 \sum_{j=1}^6 (X_{j2} \gamma_j) \\
 Z &= R^2 \sum_{j=1}^6 (X_{j3} \gamma_j)
 \end{aligned} \right\} \quad \text{IV-18}$$

PARTIAL DERIVATIVES OF V_R , ψ , AND ν *

$$V_R = (V^2 + 2VV_D \cos \zeta \cos \epsilon + V_D^2)^{\frac{1}{2}}$$

$$\frac{\partial V_R}{\partial V_D} = \frac{1}{V_R} (V_D + V \cos \zeta \cos \epsilon)$$

$$\frac{\partial V_R}{\partial \epsilon} = -\frac{1}{V_R} (V V_D \cos \zeta \sin \epsilon)$$

$$\frac{\partial V_R}{\partial \zeta} = -\frac{1}{V_R} (V V_D \sin \zeta \cos \epsilon)$$

$$\psi = \tan^{-1} \left\{ \frac{V_D \sin \zeta}{(V_D^2 \cos^2 \zeta + 2VV_D \cos \zeta \cos \epsilon + V^2)^{\frac{1}{2}}} \right\}$$

$$\text{Let } Q = (V_D^2 \cos^2 \zeta + 2VV_D \cos \zeta \cos \epsilon + V^2)$$

$$\frac{\partial \psi}{\partial V_D} = \frac{\sin \zeta}{(1 + \tan^2 \psi) Q^{\frac{1}{2}}} \left\{ 1 - \frac{(V_D \cos \zeta)(V_D \cos \zeta + V \cos \epsilon)}{Q} \right\}$$

$$\frac{\partial \psi}{\partial \epsilon} = \frac{V V_D^2 \sin \zeta \cos \zeta \sin \epsilon}{(1 + \tan^2 \psi) Q^{\frac{3}{2}}}$$

$$\frac{\partial \psi}{\partial \zeta} = \frac{V_D}{(1 + \tan^2 \psi) Q^{\frac{1}{2}}} \left\{ \cos \zeta + \frac{(V_D \sin^2 \zeta)(V_D \cos \zeta + V \cos \epsilon)}{Q} \right\}$$

$$\nu = \tan^{-1} \left\{ \frac{V_D \cos \zeta \sin \epsilon}{V_D \cos \zeta \cos \epsilon + V} \right\}$$

$$\text{Let } Q_1 = (V_D \cos \zeta \cos \epsilon + V)^2$$

$$\frac{\partial \nu}{\partial V_D} = \frac{V \cos \zeta \sin \epsilon}{(1 + \tan^2 \nu) Q_1}$$

$$\frac{\partial \nu}{\partial \epsilon} = \frac{(V_D \cos \zeta)(V_D \cos \zeta + V \cos \epsilon)}{(1 + \tan^2 \nu) Q_1}$$

$$\frac{\partial \nu}{\partial \zeta} = -\frac{V V_D \sin \zeta \sin \epsilon}{(1 + \tan^2 \nu) Q_1}$$

* The angle ϕ has been assumed to be zero.

APPENDIX V

EXPLANATION AND DESCRIPTION OF THE NUCLEAR ROCKET ORBITAL DESTRUCT AREA COMPUTATION CODE

V.1 MAIN PROGRAM

The main program reads in all of the input data and data transformations, together with a number of control cards that give the user optional control over the flow of information in the computer. The data are read in the following sequence:

- a. Title Card and Ballistic Parameter.
- b. Data transformation (same transformation as used in NEFØD Program, c.f. Appendix II).
- c. Coefficients and powers in each term of the polynomial used to calculate earth range (λ).
- d. Number of sets of points, number of data increments, and number of destruct velocities to be considered at each data point.
- e. The zeta and epsilon values for each point (3 points per set) and the destruct velocity at each point.

The main program acts as a "clearing house" by first setting up the data in usable form, then calling the participating subroutines in the proper sequence to calculate the earth area contaminated by the re-entry of the destruct debris lying within the set of points and then to determine the area error coefficients X , Y and Z defined in Chapter 5.

There are several features built into the main program segment of the NURØDAC Code that permit a maximum amount of information returned for a minimum amount of data input. For example, at the end of each case, a new title and ballistic parameter card is

is read into the computer. If the ballistic parameter is the same as the one for the previous case, the computer will not require a new set of polynomial coefficients, since the earth range polynomial is the same.

Another useful feature permits one set of ζ and ϵ points (one destruct triangle) to be used over and over at different destruct velocities or one set of ζ 's and ϵ 's may be incremented several times to produce new destruct triangles. Besides condensing the input data requirements, the use of the incrementation aspect permits a parametric study of one particular region of the destruct sphere to be done in an efficient manner with the least possible input.

V.2 SUBROUTINE FLIGHT

This subprogram determines the resultant velocity, V_R , and the flight angle, ψ , corresponding to each of the three points on the destruct triangle. Further, if the resultant velocity and/or the flight angle must be transformed for use in determining earth range, λ , or derivatives of λ with respect to ζ , ϵ and V_D , it is done in this subprogram.

V.3 SUBROUTINE GNU

This subprogram calculates the orbit plane shift angles, ν , associated with the three destruct points, using the relationships shown in Chapter 3 of the text. The derivatives of ν with respect to ζ , ϵ , and V_D for each destruct point are also determined using the formulas developed in Appendix IV.

V.4 SUBROUTINE DVELØ

Subroutine DVELØ calculates the derivatives of the resultant velocity with respect

to ζ , ϵ , and V_D for each of the three destruct points. Each of the derivatives is transformed (using the data transformation described in the NEFØD Program - Appendices I and II).

V.5 SUBROUTINE OUTPT 1

This subroutine controls the printing of each case and at the completion of each problem it is called by the MAIN PROGRAM to set up the output in proper form.

V.6 SUBROUTINE DALPHA

Subroutine DALPHA performs the same operations on the flight angle, ψ , that Subroutine DVELØ does on resultant velocity.

V.7 SUBROUTINE LAMDA

This subprogram calculates the earth range, λ , associated with each of the three destruct points. To do this job, LAMDA uses the values of resultant velocity and flight angle as determined in Subroutine FLIGHT, and the polynomial coefficients read as input data.

V.8 SUBROUTINE SUMUP

This subroutine is one of the mainstays in the calculation of the area error coefficients. It is called by Subroutine CØEFF (see below) for the express purpose of calculating the coefficients of the derivatives of earth range (λ) with respect to ζ , ϵ and V_D for each of the three destruct points.

V.9 SUBROUTINE TAREA

This subprogram calculates the area on the earth generated by the impact loci of

the three points on the destructure sphere. The program employs the equations developed in Appendix IV, and uses the values of λ and ν calculated in Subroutines LAMDA and GNU, respectively. Subroutine TAREA uses as an integral part of its calculations the function subprogram, FUNCTION DACØS (Z). This latter subprogram calculates the angles subtended by the three sides on the earth area triangle.

V.10 SUBROUTINE CØEFF

Subroutine CØEFF determines the error coefficients X , Y and Z of Equation (5-6). This subprogram is the culmination of the entire program and as such draws on the results from practically all other subroutines noted above.

For instance, Subroutine CØEFF uses the λ 's determined in Subroutine LAMDA, the λ -derivative coefficients calculated in Subroutine SUMUP, and the ν -derivatives calculated in Subroutine GNU.

Table V-1 gives a list of the most important variable names in NURØDAC.

TABLE V-1

DESCRIPTION OF MOST IMPORTANT VARIABLE NAMES IN NURØDAC

HEDING	-	Temporary name assigned to Title Array when read in from data.
IBP	-	Ballistic Parameter (read in on HEDING data card).
PØLYCØ	-	Polynomial Coefficients as determined from NEFØD Program.
BETA	-	Two-dimensional form of PØLYCØ (single dimension to two dimension transformation made internally in program).
V	-	Original orbit velocity.
NSETPT	-	Number of sets - each set consisting of three destruct points that is to be considered during a given run on the computer.
NINC	-	Number of increments (of zeta, epsilon) to be considered for each set of points.
NDVP	-	Number of destruct velocities to be considered per set of points.
VD	-	Destruct velocity.
ZETAD	-	Zeta (degrees).
EPSD	-	Epsilon (degrees).
ZETA	-	Zeta (radians).
EPS	-	Epsilon (radians).
DATRAN	-	Data transformation for V_R, Ψ (see Appendix II).
ALPHA	-	Resultant flight angle.
VELØ	-	Resultant velocity.

TABLE V-1 (Cont'd)

ZNU	- Orbit plane shift angle.
GNUVD	- Derivative of V with respect to destruct velocity.
GNUZT	- Derivative of V with respect to zeta.
GNUPE	- Derivative of V with respect to epsilon.
VELØVD	- Derivative of resultant velocity with respect to destruct velocity.
VELØZT	- Derivative of resultant velocity with respect to zeta.
VELØPE	- Derivative of resultant velocity with respect to epsilon.
ALPHVD	- Derivative of resultant flight angle with respect to destruct velocity.
ALPHZT	- Derivative of resultant flight angle with respect to zeta.
ZLAM	- Earth range in radians.
ANGLE	- Included angles of earth area triangle.
AREA	- Earth area described by re-entry loci of destruct points.
A	- Array name of error coefficients for $\Delta\zeta$, $\Delta\epsilon$ and ΔV_D .
ALPHEP	- Derivative of resultant flight angle with respect to epsilon.

V.11 LISTINGS FOR NURØDAC - II

```

*      XEQ
C      MAIN PROGRAM
C      **** MAIN PROGRAM READS IN DATA AND INITIALIZES INPUT
C      TO ALL SUBROUTINES
C
C      **** COMMON (SINGLE PRECISION)
C      COMMON M, N, DATRAN, ITAG, TITLE, NCASE
C      **** COMMON (DOUBLE PRECISION)
C      COMMON A, AREA, ANGLE, COS1, SIN1, COS2, BETA, V, VD, ZETA, EPS,
1      ALPHA, VELO, ALPHVD, ALPHZT, ALPHEP, VELOVD, VELOEP, VELOZT, ZETAD
2      EPSD, ZLAM, ZNU, GNUVD, GNUZT, GNUMP, R, D
C      **** SINGLE DIMENSION VARIABLES DIMENSION STATEMENT
C      DIMENSION DATRAN(6), TITLE(12), ITAG(6)
C- **** DOUBLE PRECISION DIMENSION STATEMENT FOR VARIABLES
D      DIMENSION A(3), AREA(1), ANGLE(3), COS1(5), SIN1(5), COS2(5),
1      BETA(20,20), V(1), VD(3), ZETA(3), EPS(3), ALPHA(3), VELO(3),
2      ALPHVD(3), ALPHZT(3), ALPHEP(3), VELOVD(3), VELOZT(3), VELOEP(3),
3      ZETAD(3), EPSD(3), ZLAM(5), ZNU(5), GNUVD(3), GNUZT(3), GNUMP(3),
4      R(1), D(6)
D      DIMENSION RAD(1)
D      DIMENSION POLYCO(100), IPWRA(100), IPWRV(100), HEDING(12)
D      DIMENSION VDO(3), DELZ(3), DELEP(3), DELVD(3)
D      CALL EPMDE(5)
D      NTEST = 0
D      NCASE = 1
D      ITX = 0
25  RIPT 5, 100, HEDING , IBP
D      IF(IBP)26,26,30
26  BACKSPACE 5
D      GO TO 27
30  DO 35 I=1,12
35  TITLE(I) = HEDING(I)
D      NRT = ITX + 1
D      IF(NTEST - IBP)40,27,40
40  RIPT 5,110,(DATRAN(I), ITAG(I), I=1,6)
D      RIPT 5, 120, NTERMS, (IPWRA(I), IPWRV(I), POLYCO(I), I=1,NTERMS)
C      **** FOLLOWING STATEMENTS CONVERT POLYNOMIAL COEFFICIENTS TO TWO
C      DIMENSIONAL FORM
D      MMAX = 0
D      NMAX = 0
D      DO 250 N=1, NTERMS

```

```

        I = IPWRA(N) + 1
        J = IPWRV(N) + 1
        BETA(I,J) = POLYCO(N)
        IF(MMAX-I)210, 210, 220
210 MMAX = I
220 IF(NMAX -J)230, 230, 250
230 NMAX = J
250 CONTINUE
        R(1)= 3.9588825E03
        R(2)= 0.0E-08
        M= MMAX
        N= NMAX
        RAD(1)= 57.29578
        RAD(2)= 0.0E-08
        V(1)= 25583.9
        V(2)= 0.0E-08
27 RIPT 5, 160, NSETPT, NINC, NDVD
        IF(NINC - 1)60, 60, 65
60 NINC = 1
65 IF(NDVD - 1)68, 68, 50
68 NDVD = 1
50 RIPT 5, 180,(VD(I), ZETAD(I),EPSD(I), DELVD(I), DELZ(I), DELEP(I),
        1I=1,3)
        DO 55 I=1,3
55 VDO(I) = VD(I)
70 DO 650 K=1,NINC
        DO 500 I=1,3
D      ZETA(I) = ZETAD(I)/RAD
D 500 EPS(I) = EPSD(I)/RAD
        DO 625 J=1,NDVD
        CALL FLIGHT
        CALL LAMDA
        CALL DALPHA
        CALL DVELO
        CALL GNU
        CALL TAREA
        CALL COEFF
        CALL OUTPT1(NRT , ITX)
        DO 627 I=1,3
627 VD(I)= VD(I) + DELVD
625 CONTINUE

```

```

      DO 655 I=1,3
      ZETAD(I) = ZETAD(I) + DELZ(I)
      EPSD(I)= EPSD(I) + DELEP(I)
655  VD(I) = VDO(I)
650  CONTINUE
      IF(IBP)651, 652, 651
651  NTEST = IBP
652  NSETPT = NSETPT - 1
      IF(NSETPT)25, 25, 50
100  FORMAT(12A6, 15)
110  FORMAT(6(F10.4,12))
120  FORMAT( 15/(215,E20.8))
160  FORMAT(3I5)
180  FORMAT(6F10.4)
      END

```

MAIN0660

```

      SUBROUTINE FLIGHT
C ****      FLIGHT DETERMINES ALPHA AND RESULTANT VELOCITY FOR USE
C              IN DETERMINING LAMDA DERIVATIVES
C ****      COMMON (SINGLE PRECISION)
      COMMON M, N, DATRAN, ITAG, TITLE, NCASE
C ****      COMMON (DOUBLE PRECISION)
      COMMON A, AREA, ANGLE, COS1, SIN1, COS2, BETA, V, VD, ZETA, EPS,
      1 ALPHA, VELO, ALPHVD, ALPHZT, ALPHEP, VELOVD, VELOEP, VELOZT, ZETAD
      2, EPSD, ZLAM, ZNU, GNUVD, GNUZT, GNUMP, R, D
C ****      SINGLE DIMENSION VARIABLES DIMENSION STATEMENT
      DIMENSION DATRAN(6), TITLE(12), ITAG(6)
C- ****      DOUBLE PRECISION DIMENSION STATEMENT FOR VARIABLES
D      DIMENSION A(3), AREA(1), ANGLE(3), COS1(5), SIN1(5), COS2(5),
      1BETA(20,20), V(1), VD(3), ZETA(3), EPS(3), ALPHA(3), VELO(3),
      2ALPHVD(3), ALPHZT(3), ALPHEP(3), VELOVD(3), VELOZT(3), VELOEP(3),
      3ZETAD(3), EPSD(3), ZLAM(5), ZNU(5), GNUVD(3), GNUZT(3), GNUMP(3),
      4R(1), D(6)
      DO 100 I=1,3
      TERM1= 2.0 *V*VD(I)*COSF(ZETA(I))*COSF(EPS(I))
      VELO(I) = SQRTF(V**2 + VD(I)**2 + TERM1)
      X= VD(I)*SINF(ZETA(I))
      YSQ=(VD(I)**2)*(COSF(ZETA(I))**2)+V**2 + TERM1
      Y = SQRTF(YSQ)
      ALPHA(I) = ARTNF(X,Y)
      ALPHA(I) = ALPHA(I)*57.29578
100 CONTINUE
C ****      FOLLOWING LOOPS TRANSFORMS RESULTANT VELOCITY AND ALPHA
C              FOR USE IN LAMDA POLYNOMIAL
      D(1) = 0.0
      D(2) = 1.0
      D(3) = 1.0
      D(4) = 0.0
      D(5) = 1.0
      D(6) = 1.0
      DO 160 I=1,6
150 J= ITAG(I)
      IF(J)200, 200, 160
160 D(J) = DATRAN(I)
200 DO 250 I= 1,3
      ALPHA(I) = (ALPHA(I)+ D(1))*D(2) / D(3)
      VELO(I) = (VELO(I)+D(4))*D(5)/D(6)

```

250 CONTINUE
RETURN
END

- V-11 -

WANL-TME-896

```

      SUBROUTINE GNU
C ***** GNU DETERMINES PARTIAL DERIVATIVES OF NU WITH RESPECT TO
C          VD(GNUVD), ZETA(GNUZT), EPSILON(GNUEP)
C          AND THE VALUES OF ZNU
C ***** COMMON (SINGLE PRECISION)
      COMMON M, N, DATRAN, ITAG, TITLE, NCASE
C ***** COMMON (DOUBLE PRECISION)
      COMMON A, AREA, ANGLE, COS1, SIN1, COS2, BETA, V, VD, ZETA, EPS,
      1 ALPHA, VELO, ALPHVD, ALPHZT, ALPHEP, VELOVD, VELOEP, VELOZT, ZETAD
      2, EPSD, ZLAM, ZNU, GNUVD, GNUZT, GNUEP, R, D
C ***** SINGLE DIMENSION VARIABLES DIMENSION STATEMENT
      DIMENSION DATRAN(6), TITLE(12), ITAG(6)
C- ***** DOUBLE PRECISION DIMENSION STATEMENT FOR VARIABLES
D      DIMENSION A(3), AREA(1), ANGLE(3), COS1(5), SIN1(5), COS2(5),
      1BETA(20,20), V(1), VD(3), ZETA(3), EPS(3), ALPHA(3), VELO(3),
      2ALPHVD(3), ALPHZT(3), ALPHEP(3), VELOVD(3), VELOZT(3), VELOEP(3),
      3ZETAD(3), EPSD(3), ZLAM(5), ZNU(5), GNUVD(3), GNUZT(3), GNUEP(3),
      4R(1), D(6)
D      DIMENSION DENOM(1), DSQ(1)
      DO 100 K=1,3
D      DENOM = VD(K)*COSF(ZETA(K))*COSF(EPS(K)) + V
D      DSQ = DENOM*DENOM
D      ZNU(K) = VD(K)*COSF(ZETA(K))*SINF(EPS(K))/DENOM
D      GNUVD(K) = V*COSF(ZETA(K))*SINF(EPS(K))/DSQ
D      GNUZT(K) = (-1.0 * VD(K) * V * SINF(ZETA(K))*SINF(EPS(K)))/DSQ
D      GNUEP(K) = VD(K)*COSF(ZETA(K))*(VD(K)*COSF(ZETA(K))+V*COSF(EPS(K))
      1)/DSQ
100 CONTINUE
      RETURN
      END

```



```

      SUBROUTINE DVELO
C ***** DVELO OBTAINS PARTIAL DERIVATIVES OF RESULTANT VELOCITY WITH
C          RESPECT TO VD(VELOVD), ZETA(VELOZT), EPSILON(VELOEP) *****
C ***** COMMON (SINGLE PRECISION)
      COMMON M, N, DATRAN, ITAG, TITLE,      NCASE
C ***** COMMON (DOUBLE PRECISION)
      COMMON A, AREA, ANGLE, COS1, SIN1, COS2, BETA, V, VD, ZETA, EPS,
      1 ALPHA, VELO, ALPHVD, ALPHZT, ALPHEP, VELOVD, VELOEP,VELOZT, ZETAD
      2, EPSD, ZLAM, ZNU, GNUVD, GNUZT, GNUMP, R, D
C ***** SINGLE DIMENSION VARIABLES DIMENSION STATEMENT
      DIMENSION DATRAN(6), TITLE(12), ITAG(6)
C- ***** DOUBLE PRECISION DIMENSION STATEMENT FOR VARIABLES
D      DIMENSION A(3), AREA(1), ANGLE(3), COS1(5), SIN1(5), COS2(5),
      1BETA(20,20), V(1), VD(3), ZETA(3), EPS(3), ALPHA(3), VELO(3),
      2ALPHVD(3), ALPHZT(3), ALPHEP(3), VELOVD(3), VELOZT(3), VELOEP(3),
      3ZETAD(3), EPSD(3), ZLAM(5), ZNU(5), GNUVD(3), GNUZT(3), GNUMP(3),
      4R(1), D(6)
D      DIMENSION DNOMSQ(1), DENOM(1)
      DO 100 K=1,3
D      DNOMSQ = VD(K)**2 +(V**2)+ 2.0*VD(K)*V*COSF(ZETA(K))*COSF(EPS(K))
D      DENOM = SQRTF(DNOMSQ)
D      VELOVD(K)=(VD(K)+ V*COSF(ZETA(K))*COSF(EPS(K)))/DENOM
D      VELOZT(K)=(-1.0 *VD(K)*V* SIN F(ZETA(K))* COSF(EPS(K)))/DENOM
D      VELOEP(K)=(-1.0 *VD(K)*V* COSF(ZETA(K))* SIN F(EPS(K)))/DENOM
C      FOLLOWING STATEMENTS TRANSFORM VELO DERIVATIVES INTO SAME
C      TYPE AS USED FOR DETERMINING LAMDA
D      VELOVD(K) = VELOVD(K)*D(5)/D(6)
D      VELOZT(K) = VELOZT(K)*D(5)/D(6)
D      VELOEP(K) = VELOEP(K)*D(5)/D(6)
100 CONTINUE
      RETURN
      END

```

```

      SUBROUTINE OUTPT1(NRT, ITX)
C ***** OUTPT1 SETS UP AND PRINTS RESULTS IF DELTA AREA NOT CALC
C ***** COMMON (SINGLE PRECISION)
      COMMON M, N, DATRAN, ITAG, TITLE,      NCASE
C ***** COMMON (DOUBLE PRECISION)
      COMMON A, AREA, ANGLE, COS1, SIN1, COS2, BETA, V, VD, ZETA, EPS,
      1 ALPHA, VELO, ALPHVD, ALPHZT, ALPHEP, VELOVD, VELOEP, VELOZT, ZETAD
      2, EPSD, ZLAM, ZNU, GNUVD, GNUZT, GNUEP, R, D
C ***** SINGLE DIMENSION VARIABLES DIMENSION STATEMENT
      DIMENSION DATRAN(6), TITLE(12), ITAG(6)
C- ***** DOUBLE PRECISION DIMENSION STATEMENT FOR VARIABLES
D      DIMENSION A(3), AREA(1), ANGLE(3), COS1(5), SIN1(5), COS2(5),
      1BETA(20,20), V(1), VD(3), ZETA(3), EPS(3), ALPHA(3), VELO(3),
      2ALPHVD(3), ALPHZT(3), ALPHEP(3), VELOVD(3), VELOZT(3), VELOEP(3),
      3ZETAD(3), EPSD(3), ZLAM(5), ZNU(5), GNUVD(3), GNUZT(3), GNUEP(3),
      4R(1), D(6)
      DIMENSION ARC(3), ZNUDG(3)
      DO 700 I=1,3
      ARC(I) = ZLAM(I)*R
      700 ZNUDG(I) = ZNU(I)*57.29578
C ***** FOLLOWING STATEMENTS CALCULATES PER CENT AREA CHANGE
      TRM1 = A(1)*100.0
      TRM2 = A(2)*1.0
      TRM3 = A(3)*1.0
      DELA = TRM1 + TRM2 + TRM3
      DADIVA = (DELA/AREA)*100.0
      IF(NRT - ITX)75, 60, 75
      60 IF(NCASE - 2)80, 90, 75
      75 NCASE = 1
      ITX = NRT
      80 WOT 6, 100, TITLE
      90 WOT 6, 105
      WOT 6, 110
      WOT 6, 120, ZETAD(1), EPSD(1), VD(1)
      WOT 6, 140, ZETAD(2), EPSD(2), VD(2), A, AREA
      WOT 6, 120, ZETAD(3), EPSD(3), VD(3)
      WOT 6, 160
      WOT 6, 180, ARC(1), ZNUDG(1), DADIVA
      WOT 6, 200, ARC(2), ZNUDG(2)
      WOT 6, 210, ARC(3), ZNUDG(3)
      WOT 6, 220

```

```

      NCASE = NCASE + 1
100 FORMAT(1H1,17X74H*****
1 ***** /18X1H*72X1H*/18X1H*12A6,1H*/18X1H*72X
21H*/18X74H*****
3 ***** //)
105 FORMAT(110H*****
1 ***** )
110 FORMAT(          7X26HDESTRUCT VARIABLES VALUE 8X1H*5X35HDEST
1RUCT AREA ERROR COEFFICIENTS 7X1H*16H DESTRUCT AREA / 7X26H**
2 ***** 8X1H*5X35H*****
3 ***** 7X1H*2X14H*****/ 41X1H*47X1H*/7X4HZETA7X7HEPSILON4X
49HDEST.VEL.3X1H* 62H DEL DST VEL * DEL EPSILON * DEL ZETA
5 * (SQ.MILES) / 6X6H(DEG.)6X6H(DEG.)5X9H(FT./SEC)3X1H*48H SQ MI
6/(FT/SEC)* (SQ MI/DEG) * (SQ MI/DEG) * /41X1H*15X1H*15X1H*15X
71H*/ )
120 FORMAT(5X,F8.4,4X,F8.4,4X,F9.4,3X1H*,15X1H*,15X1H*15X1H*/41X1H*15X
11H*15X1H*15X1H* )
140 FORMAT(5X,F8.4,4X,F8.4,4XF9.4,3X1H*,3X1PE12.5,4H* E12.5,4H* ,
1E12.5,6H* E12.5,/ 41X1H*15X1H*15X1H*15X1H* )
160 FORMAT(110H*****
1 ***** /5X38HDEST
2RUCT RANGES * FLIGHT PATH SHIFT* / 8X7H(MILES)7X1H*5X 9H(DEGREES
3)5X 18H* PER CENT DEST )
180 FORMAT(7X1PE12.5,7H * E12.5,24H * AREA CHANGE = E12.5,
14X29H(BASED ON ONE DEGREE ERROR )
200 FORMAT(7X1PE12.5,7H * E12.5,5H * 39X23HIN ZETA AND EPSIL
1ON, )
210 FORMAT(7X1PE12.5,7H * E12.5,5H * 39X24H100 FT/SEC IN DEST V
1EL.) )
220 FORMAT(110H*****
1 ***** //)
      RETURN
      END

```

```

      SUBROUTINE DALPHA
C  **** DALPHA CALCULATES PARTIAL DERIVATIVES OF ALPHA WITH RESPECT
C          TO VD(ALPHVD), ZETA(ALPHZT) , EPSILON(ALPHEP)
C  **** COMMON (SINGLE PRECISION)
      COMMON M, N, DATRAN, ITAG, TITLE, NCASE
C  **** COMMON (DOUBLE PRECISION)
      COMMON A, AREA, ANGLE, COS1, SIN1, COS2, BETA, V, VD, ZETA, EPS,
      1 ALPHA, VELO, ALPHVD, ALPHZT, ALPHEP, VELOVD, VELOEP, VELOZT, ZETAD
      2, EPSD, ZLAM, ZNU, GNUVD, GNUZT, GNEP, R, D
C  **** SINGLE DIMENSION VARIABLES DIMENSION STATEMENT
      DIMENSION DATRAN(6), TITLE(12), ITAG(6)
C- **** DOUBLE PRECISION DIMENSION STATEMENT FOR VARIABLES
D      DIMENSION A(3), AREA(1), ANGLE(3), COS1(5), SIN1(5), COS2(5),
      1BETA(20,20), V(1), VD(3), ZETA(3), EPS(3), ALPHA(3), VELO(3),
      2ALPHVD(3), ALPHZT(3), ALPHEP(3), VELOVD(3), VELOZT(3), VELOEP(3),
      3ZETAD(3), EPSD(3), ZLAM(5), ZNU(5), GNUVD(3), GNUZT(3), GNEP(3),
      4R(1), D(6)
D      DIMENSION DENOM(1)
      DO 100 K=1, 3
D      DENOM = SQRTF((( VD(K)**2)*(COSF(ZETA(K))**2)+ 2.0*V*VD(K)*COSF(Z
      1ETA(K))*COSF(EPS(K))+ V**2)**3)
D      ALPHVD(K)= (V*VD(K)*COSF(ZETA(K))*COSF(EPS(K))+ V**2)*SINF(ZETA(K)
      1)/DENOM
D      ALPHZT(K) = (VD(K)*COSF(ZETA(K))*(VD(K)**2 +V**2)+VD(K)**2*V*COSF(
      1EPS(K))*(1.0 + COSF(ZETA(K))**2))/DENOM
D      ALPHEP(K) =(V*VD(K)**2)*SINF(ZETA(K))*COSF(ZETA(K))*SINF(EPS(K))/
      1DENOM
      100 CONTINUE
      DO 200 K = 1,3
C      ***** THE LAST TWO FACTORS , D(2) AND D(3) TRANSFORM ALPHA DERIV
C          ATIVES INTO SAME TRANSFORMATION USED FOR DETERMINING LAMDA
D      ALPHZT(K) = ALPHZT(K)*57.29578*D(2)/D(3)
D      ALPHVD(K) = ALPHVD(K)*57.29578*D(2)/D(3)
D 200 ALPHEP(K) = ALPHEP(K)*57.29578*D(2)/D(3)
      RETURN
      END

```

```

      SUBROUTINE LAMDA
C  ****  COMMON (SINGLE PRECISION)
      COMMON M, N, DATRAN, ITAG, TITLE,      NCASE
C  ****  COMMON (DOUBLE PRECISION)
      COMMON A, AREA, ANGLE, COS1, SIN1, COS2, BETA, V, VD, ZETA, EPS,
      1 ALPHA, VELO, ALPHVD, ALPHZT, ALPHEP, VELOVD, VELOEP,VELOZT, ZETAD
      2, EPSD, ZLAM, ZNU, GNUVD, GNUZT, GNUMP, R, D
C  ****  SINGLE DIMENSION VARIABLES DIMENSION STATEMENT
      DIMENSION DATRAN(6), TITLE(12), ITAG(6)
C- ****  DOUBLE PRECISION DIMENSION STATEMENT FOR VARIABLES
D  DIMENSION A(3), AREA(1), ANGLE(3), COS1(5), SIN1(5), COS2(5),
      1BETA(20,20), V(1), VD(3),  ZETA(3), EPS(3), ALPHA(3), VELO(3),
      2ALPHVD(3), ALPHZT(3), ALPHEP(3), VELOVD(3), VELOZT(3), VELOEP(3),
      3ZETAD(3), EPSD(3), ZLAM(5), ZNU(5), GNUVD(3), GNUZT(3), GNUMP(3),
      4R(1), D(6)
      DO 3 K=1,3
      ZLAM(K) = 0.
      IF(ALPHA(K))100,200,100
100  DO 1 I=1,M
      DO 1 J = 1,N
      1  ZLAM(K) = BETA(I,J)*ALPHA(K)**(I-1)*VELO(K)**(J-1) + ZLAM(K)
      GO TO 3
200  DO 2 J = 1,N
      2  ZLAM(K) = BETA(1,J)*VELO(K)**(J-1) + ZLAM(K)
      3  ZLAM(K) = ZLAM(K)/57.29578
      RETURN
      END

```

```

      SUBROUTINE SUMUP(L,LL,SUM)
C   ****  COMMON (SINGLE PRECISION)
      COMMON M, N, DATRAN, ITAG, TITLE,      NCASE
C   ****  COMMON (DOUBLE PRECISION)
      COMMON A, AREA, ANGLE, COS1, SIN1, COS2, BETA, V, VD, ZETA, EPS,
      1 ALPHA, VELO, ALPHVD, ALPHZT, ALPHEP, VELOVD, VELOEP,VELOZT, ZETAD
      2, EPSD, ZLAM, ZNU, GNUVD, GNUZT, GNUMP, R, D
C   ****  SINGLE DIMENSION VARIABLES DIMENSION STATEMENT
      DIMENSION DATRAN(6), TITLE(12), ITAG(6)
C-  ****  DOUBLE PRECISION DIMENSION STATEMENT FOR VARIABLES
D   DIMENSION A(3), AREA(1), ANGLE(3), COS1(5), SIN1(5), COS2(5),
      1BETA(20,20), V(1), VD(3), ZETA(3), EPS(3), ALPHA(3), VELO(3),
      2ALPHVD(3), ALPHZT(3), ALPHEP(3), VELOVD(3), VELOZT(3), VELOEP(3),
      3ZETAD(3), EPSD(3), ZLAM(5), ZNU(5), GNUVD(3), GNUZT(3), GNUMP(3),
      4R(1), D(6)
D   DIMENSION SUM(1) , SUM1(1), SUM2(1),XM(1), XN(1), DB(1), DV(1),
      1B(1), VB(1)
D   B = ALPHA(LL)
D   VB = VELO(LL)
      GO TO (1,2,3),L
D   1 DB = ALPHVD(LL)
D   DV = VELOVD(LL)
      GO TO 4
D   2 DB = ALPHEP(LL)
D   DV = VELOEP(LL)
      GO TO 4
D   3 DB = ALPHZT(LL)
D   DV = VELOZT(LL)
D   4 SUM = 0.
      IF(B)100,50,100
50   DO 60 J = 1,N
      XN = J
D   60 SUM = (XN-1.)*BETA(1,J)*DV*VB**(J-2)+BETA(2,J)*DB*VB**(J-1)+SUM
      GO TO 6
100  DO 5 I=1,M
      DO 5 J=1,N
      XM = I
      XN = J
D   SUM1 = BETA(I,J)*B**(I-2)*VB**(J-2)
D   SUM2 = (XM-1.)*VB*DB+(XN-1.)*B*DV
D   5 SUM = SUM+SUM1*SUM2

```

```
C **** STATEMENT 6 CONVERTS THE SUMMED PARTIALS OF LAMDA WITH RESPECT
C      TO VD, EPS, AND ZETA FROM DEGREES TO RADIANS
D      6 SUM = SUM/57.29578
      RETURN
      END
```

```

      SUBROUTINE TAREA
C  ****  COMMON (SINGLE PRECISION)
      COMMON M, N, DATRAN, ITAG, TITLE,      NCASE
C  ****  COMMON (DOUBLE PRECISION)
      COMMON A, AREA, ANGLE, COS1, SIN1, COS2, BETA, V, VD, ZETA, EPS,
      1 ALPHA, VELO, ALPHVD, ALPHZT, ALPHEP, VELOVD, VELOEP,VELOZT, ZETAD
      2, EPSD, ZLAM, ZNU, GNUVD, GNUZT, GNUMP, R, D
C  ****  SINGLE DIMENSION VARIABLES DIMENSION STATEMENT
      DIMENSION DATRAN(6), TITLE(12), ITAG(6)
C- ****  DOUBLE PRECISION DIMENSION STATEMENT FOR VARIABLES
D      DIMENSION A(3), AREA(1), ANGLE(3), COS1(5), SIN1(5), COS2(5),
      1BETA(20,20), V(1), VD(3), ZETA(3), EPS(3), ALPHA(3), VELO(3),
      2ALPHVD(3), ALPHZT(3), ALPHEP(3), VELOVD(3), VELOZT(3), VELOEP(3),
      3ZETAD(3), EPSD(3), ZLAM(5), ZNU(5), GNUVD(3), GNUZT(3), GNUMP(3),
      4R(1), D(6)
D      DIMENSION SUMANG(1), PI(1)
C  CALCULATE ZLAM(4), ZLAM(5), ZNU(4), AND ZNU(5)
C  STATEMENT 1 DELETED
      PI(1)= 3.1415926
      PI(2) = 0.536E-07
D      ZLAM(4) = ZLAM(1)
D      ZLAM(5) = ZLAM(2)
D      ZNU(4) = ZNU(1)
D      ZNU(5) = ZNU(2)
C  CALCULATE COS1(I)
      DO 2 I=1,3
D      COS1(I) = COSF(ZLAM(I+1))*COSF(ZLAM(I+2))
D      2 COS1(I) = COS1(I)+SINF(ZLAM(I+1))*SINF(ZLAM(I+2))*COSF(ZNU(I+1)-
      1ZNU(I+2))
D      COS1(4) = COS1(1)
D      COS1(5) = COS1(2)
C  CALCULATE SIN1(I)
      DO 3 I=1,3
D      3 SIN1(I) = SQRTF(1.-COS1(I)**2)
D      SIN1(4) = SIN1(1)
D      SIN1(5) = SIN1(2)
C  CALCULATE COS2(I)
      DO 4 I=1,3
D      4 COS2(I) = (COS1(I)-COS1(I+1)*COS1(I+2))/(SIN1(I+1)* SIN1(I+2))
D      COS2(4) = COS2(1)
D      COS2(5) = COS2(2)

```



```

C  CALCULATE ANGLE(I),SUMANG, AND AREA
D    SUMANG = 0.
      DO 5 I=1,3
        IF(COS2(I) - 1.0)800, 800, 810
      810 WOT6, 900
      900 FORMAT(18HCOS2 LARGER THAN 1  )
D  800 ANGLE(I) = DACOS(COS2(I))
D    5 SUMANG = SUMANG+ANGLE(I)
D    AREA = (SUMANG - PI)*R**2
      RETURN
      END

```

```

      FUNCTION DACOS(Z)
D     DIMENSION PI(1), Z(5)
      PI(1)= 3.1415926
      PI(2)= 0.536E-07
      IF(Z)1,2,3
D     1 Z = -Z
D     DACOS = PI      -ATANF(SQRTF(1.-Z**2)/Z)
      GO TO 4
D     2 DACOS= 1.570796
      GO TO 4
D     3 DACOS= ATANF(SQRTF(1.-Z**2)/Z)
      GO TO 4
D     4 RETURN
      END

```

```

      SUBROUTINE COEFF
C   **** COMMON (SINGLE PRECISION)
      COMMON M, N, DATRAN, ITAG, TITLE,      NCASE
C   **** COMMON (DOUBLE PRECISION)
      COMMON A, AREA, ANGLE, COS1, SIN1, COS2, BETA, V, VD, ZETA, EPS,
      1 ALPHA, VELO, ALPHVD, ALPHZT, ALPHEP, VELOVD, VELOEP, VELOZT, ZETAD
      2, EPSD, ZLAM, ZNU, GNUVD, GNUZT, GNUMP, R, D
C   **** SINGLE DIMENSION VARIABLES DIMENSION STATEMENT
      DIMENSION DATRAN(6), TITLE(12), ITAG(6)
C-  **** DOUBLE PRECISION DIMENSION STATEMENT FOR VARIABLES
D   DIMENSION A(3), AREA(1), ANGLE(3), COS1(5), SIN1(5), COS2(5),
      1BETA(20,20), V(1), VD(3), ZETA(3), EPS(3), ALPHA(3), VELO(3),
      2ALPHVD(3), ALPHZT(3), ALPHEP(3), VELOVD(3), VELOZT(3), VELOEP(3),
      3ZETAD(3), EPSD(3), ZLAM(5), ZNU(5), GNUVD(3), GNUZT(3), GNUMP(3),
      4R(1), D(6)
D   DIMENSION W(3,6), SLAM(3), CLAM(3), FACT(1), DENOM(1), TEMP(1),
      1GA(3), Y(6), X(6,3), CNU(5), SNU(3), SUM(1)
C   CALCULATE W(I,J)
      DO 6 I=1,3
D     SLAM(I) = SINF(ZLAM(I))
D     CLAM(I) = COSF(ZLAM(I))
D     6 CNU(I) = COSF(ZNU(I+1)-ZNU(I+2))
D     SNU(1) = SINF(ZNU(3)-ZNU(2))
D     SNU(2) = SINF(ZNU(3)-ZNU(1))
D     SNU(3) = SINF(ZNU(1)-ZNU(2))
D     FACT = -1./SQRTF(1.-COS1(1)**2)
D     W(1,2) = FACT*(-SLAM(2)*CLAM(3)+CLAM(2)*SLAM(3)*CNU(1))
D     W(1,3) = FACT*(-CLAM(2)*SLAM(3)+SLAM(2)*CLAM(3)*CNU(1))
D     W(1,5) = FACT*(SLAM(2)*SLAM(3)*SNU(1))
D     W(1,6) = -W(1,5)
D     FACT = -1./SQRTF(1.-COS1(2)**2)
D     W(2,1) = FACT*(-SLAM(1)*CLAM(3)+CLAM(1)*SLAM(3)*CNU(2))
D     W(2,3) = FACT*(-CLAM(1)*SLAM(3)+SLAM(1)*CLAM(3)*CNU(2))
D     W(2,4) = FACT*(SLAM(1)*SLAM(3)*SNU(2))
D     W(2,6) = -W(2,4)
D     FACT = -1./SQRTF(1.-COS1(3)**2)
D     W(3,1) = FACT*(-SLAM(1)*CLAM(2)+CLAM(1)*SLAM(2)*CNU(3))
D     W(3,2) = FACT*(-CLAM(1)*SLAM(2)+SLAM(1)*CLAM(2)*CNU(3))
D     W(3,4) = FACT*(-SLAM(1)*SLAM(2)*SNU(3))
D     W(3,5) = -W(3,4)
C     W(1,1),W(1,4),W(2,2),W(2,5),W(3,3), AND W(3,6) ARE ZERO

```

```

C  CALCULATE GA(I)
D    DENOM=SIN1(2)**2-COS1(1)**2-COS1(3)**2+2.*COS1(1)*COS1(2)*COS1(3)
D    DENOM = SQRTF(DENOM)
D    DO 7 I=1,3
D      TEMP = SIN1(I)**2+(COS1(I)-1.)*(COS1(I+1)+COS1(I+2))
D    7 GA(I) = TEMP/(DENOM*SIN1(I))
C  CALCULATE Y'S
D    DO 8 I=1,6
D      Y(I) = 0.
D      DO 8 J=1,3
D    8 Y(I) = GA(J)*W(J,I)+Y(I)
C  CALCULATE X(I,J)
D    DO 10 I=1,3
D      DO 9 J=1,3
D        CALL SUMUP(I,J,SUM)
D    9 X(J,I) = SUM
D      DO 10 J=4,6
D        K = J-3
D        GO TO (100,101,102),I
D 100 X(J,I) = GNUVD(K)
D      GO TO 10
D 101 X(J,I) = GNUEP(K)
D      GO TO 10
D 102 X(J,I) = GNUZT(K)
D 10  CONTINUE
C  CALCULATE COEFFICIENTS
D    DO 11 I=1,3
D      A(I) = 0.
D      DO 11 J=1,6
D    11 A(I) = A(I)+ Y(J)*X(J,I)
D      DO 12 I=1,3
D    12 A(I) = A(I)*R**2
D      A(2) = A(2)/57.29578
D      A(3) = A(3)/57.29578
D      RETURN
D      END

```