JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY

PASADENA, CALIFORNIA

RQ-50

RE-ORDER NO. 67-99

Document Number VOY-C2-TR8
13 February, 1967

This work was performed for the Jet Propulsion Laboratory, California Institute of Technology, sponsored by the National Aeronautics and Space Administration under Contract NAS7-100.

VOYAGER MARS PLANETARY QUARANTINE - BASIC MATH MODEL REPORT

Prepared by

G.E. Ingram
Technical Military Planning Operation
Santa Barbara, California

T. Green
Applied Mathematics
Engineering and Scientific Computer Applications
Valley Forge, Pennsylvania

M. Martin
Analysis and Techniques
Advanced Mission Requirements
Spacecraft Department
Valley Forge, Pennsylvania

E. Berger
Advanced Interplanetary Programs
Advanced Requirements Planning Operations
Pasadena, California

Approved by

R.P. Wolfson
R.P. Wolfson
Cognizant Engineer
Planetary Quarantine
Voyager System Project

Prepared For

Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, California
Under JPL Contract No. 951112

GENERAL ELECTRIC
MISSILE AND SPACE DIVISION
Valley Forge Space Technology Center
P.O. Box 8555 • Philadelphia 1, Penna.

TABLE OF CONTENTS

INTRODUCTION

The prime goal of the Planetary Quarantine Subtask under the Voyager Phase 1A Task C Study is to show the effect of the Planetary Quarantine requirements on the Voyager mission and its elements.

Figure I is a simplified work flow diagram showing the major Planetary Quarantine subtasks and their interrelationships. Activities performed on this contract are being documented in bimonthly progress reports and a separate series of technical reports and memos. The present report presents the activities to date under the Basic Math Model Development Subtask. Section 1 discusses the basic questions involved and the selected approach to be used. Section 2 describes the mechanization of the basic concepts. Two previous reports, VOY-C2-TR7 and VOY-C2-TR4, presented work done in the basic parametric analysis subtasks for Orbit Mechanics and Entry Analysis. Section 3 of the present report presents the work performed in taking the basic entry results and converting them to a form suitable for incorporation into the math model computer programs. Section 4 of this report gives an example of how the math model is used in performing sensitivity studies, the next major subtask in the Planetary Quarantine Study. Appendix A gives a complete listing of the programs developed for this task.

Table I lists the reports issued to date on the Planetary Quarantine Task.

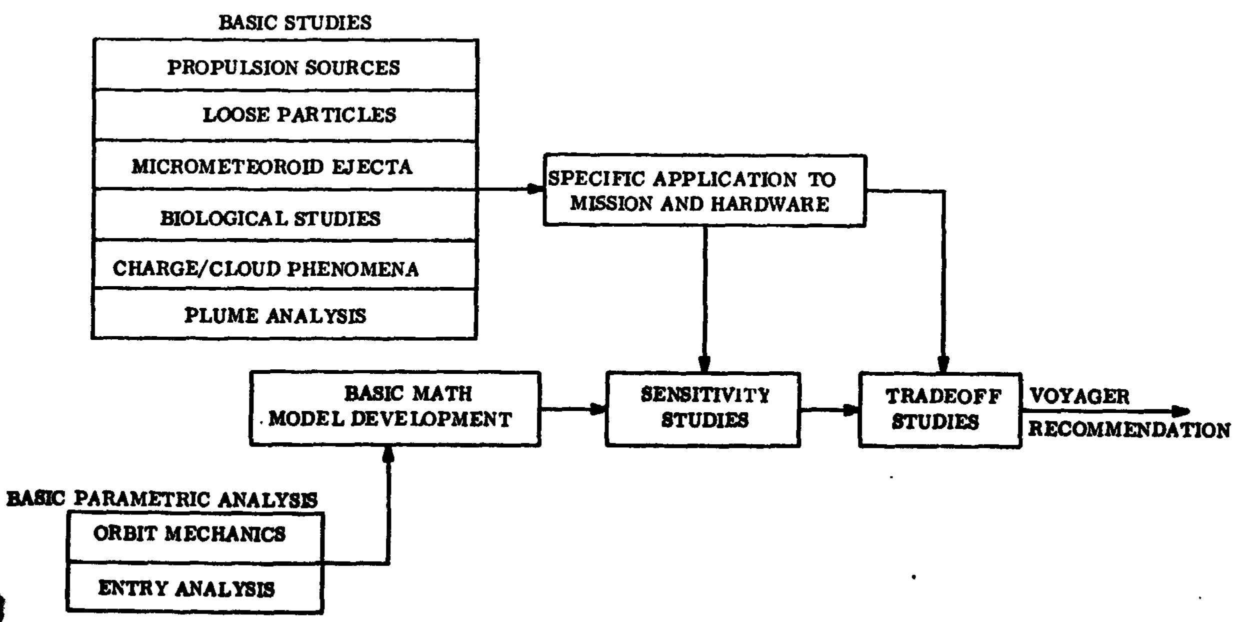


Figure I. Planetary Quarantine Task, Simplified Work Flow Diagram

Table I. Report Index for Voyager Planetary Quarantine Task C

VOY C2	Title	Author	Date
TR 1	On the Distribution of Density at Orbital Altitudes in the Martian Atmosphere.	Vachon	15 August 1966
TR 2	Prelim. Assessment of the Micrometeoroid Phenomena.	Good	July 1966
TR 3	Influence of Space Vehicle Charge and Plasma Field on the Quarantine of the Planet Mars.	McKee	1 September 1966
TR 4	Voyager Mars Planetary Quarantine Particle Burnup Study.	Parker Beerger Burrows	15 September 1966
TR 5	An Investigation into the Feasibility of Conducting an Experiment To Determine the Effects of Rocket Combustion on the Viability of Microorganisms.	Oberta	23 September 1966
TR 6	An Approximate Plume Analysis for the Voyager, Task C, Planetary Quarantine Study.	Hamel	30 September 1966
Tr 7	Voyager Mars Quarantine-Ejected Particle Trajectory Study.	Korenstein	30 November 1966
TM-1	Preliminary Combinatorial Probability Model for the Voyager Quarantine Problem (Phases 1, 2, 3).	T. Green	October 1966
TM-2	Voyager Mars P.Q. Thermal Kill of Bacteria during Mars Entry.	M. Martin	October 1966
TM-3	Loose Particle Investigation-An Evaluation of the Problem.	R. Waite	October 1966
TM-4	Micrometeoroid Effects - Analytical Studies Status Report- September 30, 1966.	R. Good	October 1966
TM-5	Voyager P.Q. Literature Search- Cold Gas Systems.	J. Mason	October 1966
TM-6	Preliminary Bio Burden Cata - Voyager P.Q.	M. Korsterer	October 1966
TM-7	Combustion Lethality Experiment - Status Report.	Oberta	November 1966
TM-8	Radiation Effects on the Viability of Microorganisms.	Peterson Koesterer	December 1966
TM-9	CLE-Integ. Dev. Test Plan	Oberta Gillis Landry	December 1966
TM-10	Micrometeoroid Studies - Status Report, 12/66	Good Behringer Nayor	December 1966
TM-11	Micrometeoroid Simulation Experimental Studies - Status Report, Jan. 1967.	Koesterer Behringer Semon Nayor	January 1967
TM-12	Cold Gas ACS Experimental Program-Status Report January 1967.	Mason	January 1967
TM-13	Loose Particle Investigation - Status Report, January 1967.	Jones Resta Nayor	January 1967

SECTION 1
APPROACH TO MATH MODEL

by
G. E. Ingram

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

SECTION 1

APPROACH TO MATH MODEL

1.1 RATIONALE

1.1.1 STATEMENT OF PROBLEM

The problem of determining the probability of contaminating Mars before time T, as a result of a Voyager mission, is essentially one of identifying all possible contamination sources associated with the "Voyager hardware" (launch vehicle, spacecraft, lander, etc.) and describing the various mechanisms that will cause viable organisms to reach the surface of Mars. The events of interest, therefore, can be described generically as follows---
One or more viable organisms launched from earth on Voyager hardware are placed on an impact trajectory to the surface of Mars using some mechanism and survive all potential kill mechanisms (e.g., U.V., atmospheric entry heating, etc.) and arrive, survive and spread on the surface of Mars before time T. The probability of the occurrence of all such events, then, is the probability of contaminating Mars before time T as a result of the missions.

Figure 1-1 illustrates the approach being used in matrix form. The rows of the matrix enumerate all possible sources of contamination while the columns of the matrix are descriptive of how particles find their way to the surface of Mars. For purposes of illustration, only four sources of contamination are listed.

ROUTE TO MARS →

SOURCE OF CONTAMINATION	1 INITIAL LOADING - V.C.	2 SURVIVE DURING TRIP	3 EJECTION PROCESS	4 TRANSPORT PROCESS	5 SURVIVE DIE-OFF	6 SURVIVE VACUUM	7 SURVIVE U.V.	8 SURVIVE OTHER SOLAR RADIATION	9 SURVIVE ENTRY HEATING	10 SURVIVE MARS ENVIRONMENT	11 NUMBER V.O.'S - MARS SURFACE PRIOR TO TIME T
ATTITUDE CONTROL GAS SYSTEM											
ORBIT INSERTION ENGINE											
LOOSE PARTICLES											
MICRO-METEOROID EJECTA											

Figure 1-1. Math Model Format

To determine how viable organisms might find their way to the surface of Mars, one must first consider the initial loading on the vehicle, second the transport process that the particles undergo in arriving at the surface of Mars, and third the potential kill mechanisms that the viable organisms are subjected to enroute to Mars. Column 1 calls for input data on the initial loading of viable organisms on the vehicle. Column 2 calls for data describing the probability that the viable organisms initially on the vehicle will survive during the trip before the time of ejection. Data describing the manner in which the particles are ejected from the vehicle is called for in Column 3. Column 4 describes the process by which the ejected particles find their way to Mars. Columns 5 through 10 call for data on the probability that the organisms will be killed enroute to Mars.

Finally after performing the operations indicated by Columns 1 through 10 on all of the sources of contamination indicated by the rows, the number of viable organisms that reach the surface of Mars and survive before time T is given in Column 11. Column 11 then is totalled for all possible sources of contamination giving the total number of viable organisms that reach the surface of Mars and survive before time T.

1.1.2 INPUT

Figure 1-2 illustrates the flow of information by considering some important elements of the problem,---the initial loading of viable organisms, the transport process, the Mars atmosphere entry heating kill mechanism, and finally the number of viable organisms arriving at the surface of Mars and surviving before time T. Consider first the initial loading of viable organisms on the vehicle. In what terms might the initial loading be given? It might be stated as an average number of the several measurements of the loading. A more conservative approach would be to state it as some upper or maximum value of several measurements of the loading. A realistic description would be to describe this number as several ranges of values, each with an associated probability. Figure 1-3 graphically shows this type description. Note from the figure that the initial loading could be small in number, that is near 0, or it could be very large, as high as say one million. The probabiliy however, that the number is very small, or very large, is a small probability. The most probable value is somewhere in the range of one thousand. The actual intervals shown here were arbitrarily selected. Any appropriate intervals may be used.

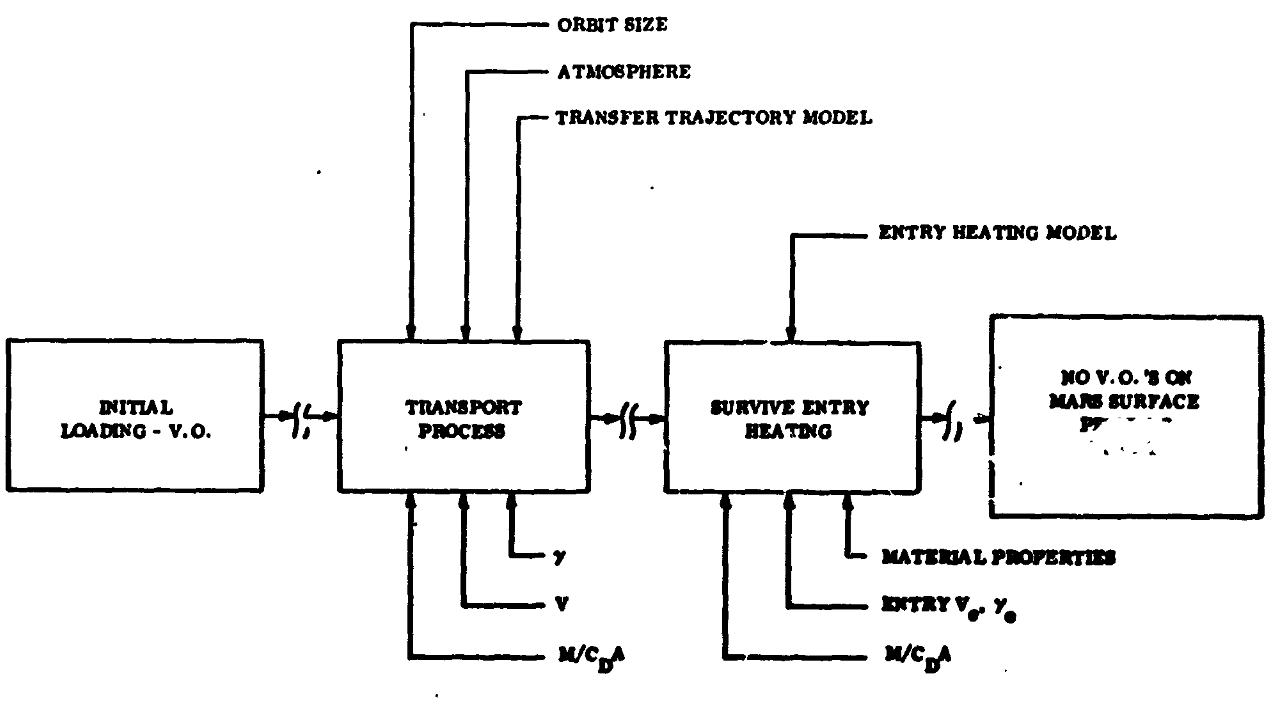


Figure 1-2. Computational Flow Diagram

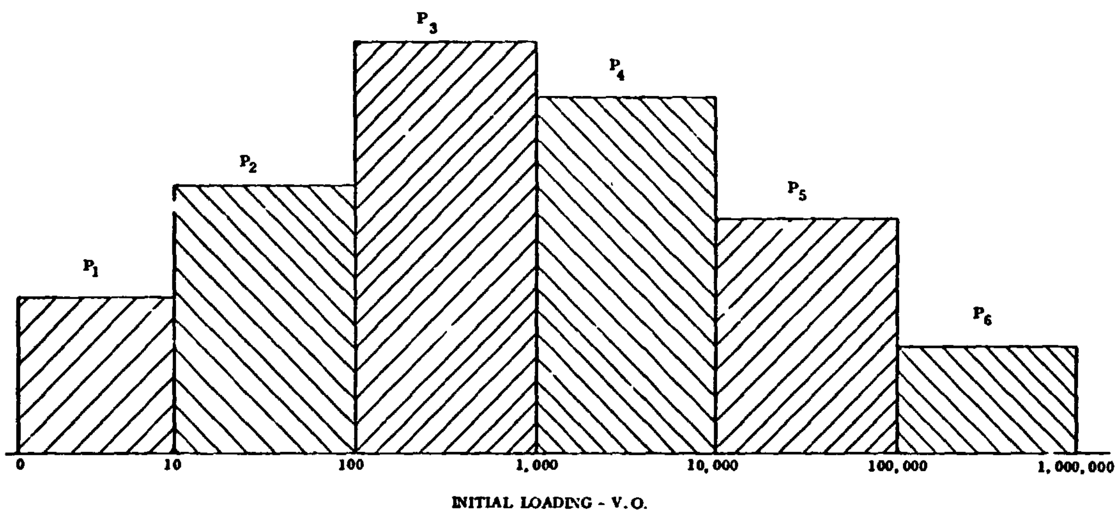


Figure 1-3. Input Data Format

Consider next the transport process. By this point in the problem, the manner in which particles leave the spacecraft has been considered. The trajectory the ejected particles take must be considered to determine if they can impact the surface of Mars by time T. An analysis must be made to determine if the particle will enter the atmosphere and eventually impact the surface of Mars. Some of the significant parameters associated with this analysis are the $\frac{M}{C_D A}$ of the particle, the velocity at which the particle leaves the spacecraft, and the angle at which it leaves the spacecraft. This analysis has been described in detail in document number VOY-C2-TR7.

Although the particle may take on a trajectory that will cause it to reach Mars, viable organisms carried by this particle may be killed enroute. As an example of one of the kill mechanisms, consider the atmosphere entry heating. The time-temperature history of the particle as it enters the Mars atmosphere and continues through the atmosphere to the surface of Mars must be considered to determine if the organisms will survive this kill mechanism. Parameters associated with this analysis are again the $\frac{M}{C_D A}$ of the particle, the velocity and angle at which the particle enters the atmosphere of Mars, and the material properties of the particle, such as emissivity. This analysis is discussed in Section 5 of this report.

Figure 1-4 illustrates the type data that might be applicable to the description of the parameters associated with the transport process and the atmosphere entry heating. The three types of density functions that we encounter are a) the smooth continuous probability density function, b) a continuous probability density function having a finite number of intervals with the density uniform over each interval, and c) a discrete type probability density function which takes on only certain values, such as integers, each value having some probability of occurrence. Each of these probability density functions may be approximated by either of the other two. Parameters such as $\frac{M}{C_D A}$, velocity, angle, emissivity, etc., are normally described by continuous type density functions. Such parameters as number of viable organisms, however, are best described by discrete type density functions. If it is sufficient to describe the number as lying within some range, however, then the second type of density function may be used to describe the number of viable organisms.

1.1.3 OUTPUT

Consider now the number of viable organisms that reach the Mars atmosphere or surface and survive before time T, which is the output of the contamination analysis for a given source. Figure 1-5 illustrates the type density function that might represent this output. A discrete type density function is used to represent the number of viable organisms in the region near zero. This allows one to look at the probability of getting one or more organisms, two or more organisms, three or more organisms, etc. For the larger numbers it may be sufficient merely to state the probability that the number lies within some range or ranges, such as between five and one hundred, or between one hundred and one million. Based on the hypothetical information in Figure 1-5, the probability that one or more viable organisms

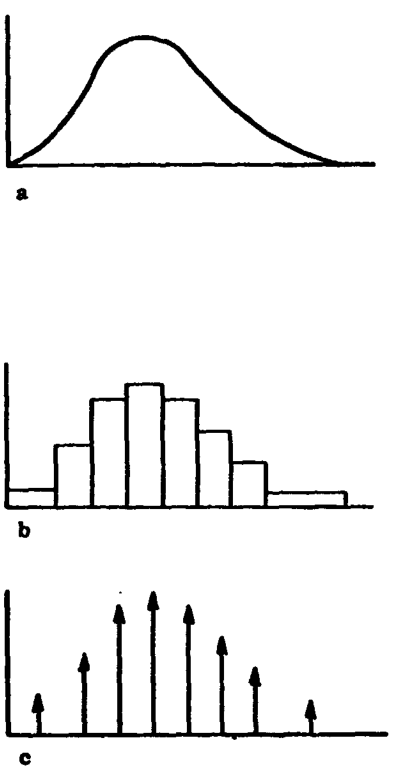


Figure 1-4. Alternate Data Formats

will reach Mars surface and continue to survive before time T is 10^{-3}. This is arrived at by either adding up the probabilities for 1, 2, 3, 4, 5, and on up, or by subtracting the probability of zero (which is 0.999) from 1. This hypothetical density function also shows that the probability of four or more viable organisms reaching Mars surface and continuing to survive before time T is 10^{-4}. If, for example, the criteria for contamination of Mars was one or more organisms reaching Mars and the constraint is that the probability of contaminating Mars must not exceed 10^{-4}, then the constraint would not have been met based on the information shown in the figure. If, on the other hand, as many as three viable organisms reaching Mars could be tolerated without representing contamination, then the constraint of 10^{-4} would have been met. If it were to cost a great deal more to meet the constraint of 10^{-4} using the criteria of one or more viable organisms representing contamination, then some reevaluation of the problem may be in order.

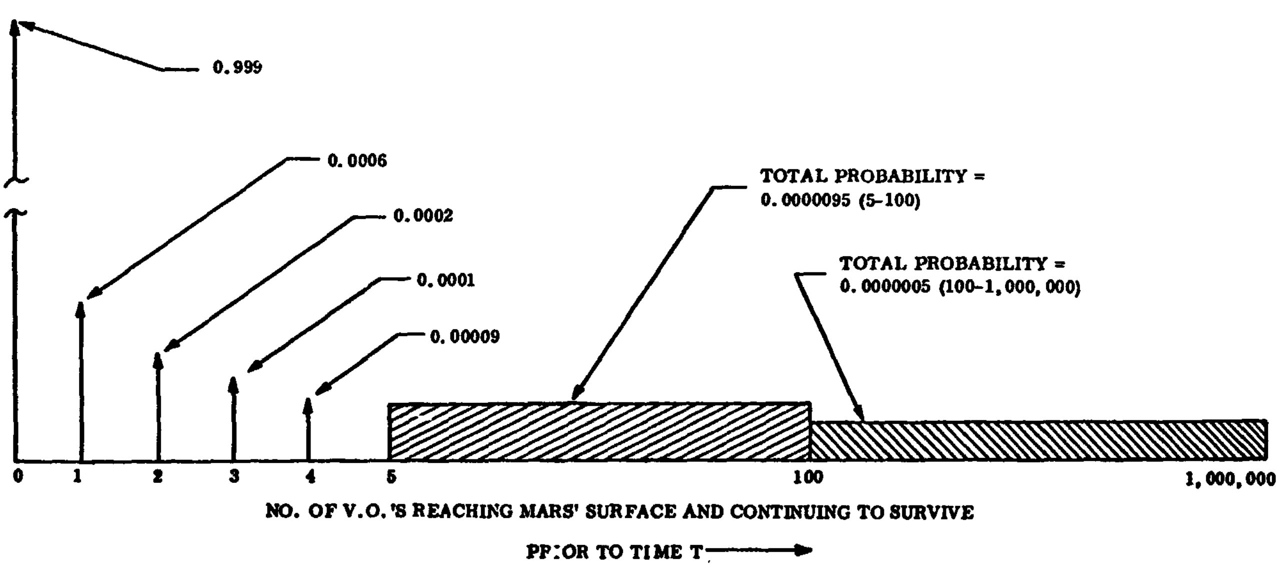


Figure 1-5. Typical Output Format

1.2 COMPUTATIONAL PROBLEMS

Primarily, there are three distinct kinds of computational problems associated with the determination of the probability of contaminating Mars before time T. The first is that of

operating on an initial probability density function by a set of conditional probability density functions and thus generating a marginal probability density function as the output. The second is that of summing probability density functions. The third is that of generating the probability density function on a random variable that has been defined as some function of one or more other random variables, each having its own a priori probability density function.

1.2.1 CONDITIONAL PROBABILITY DENSITY FUNCTIONS

This computation arises in situations such as:

a. Given a range of sizes of particles (S) (probability density function on size), what are the number of viable organisms (O) carried by these particles? This computation is represented graphically by Figure 1-6.

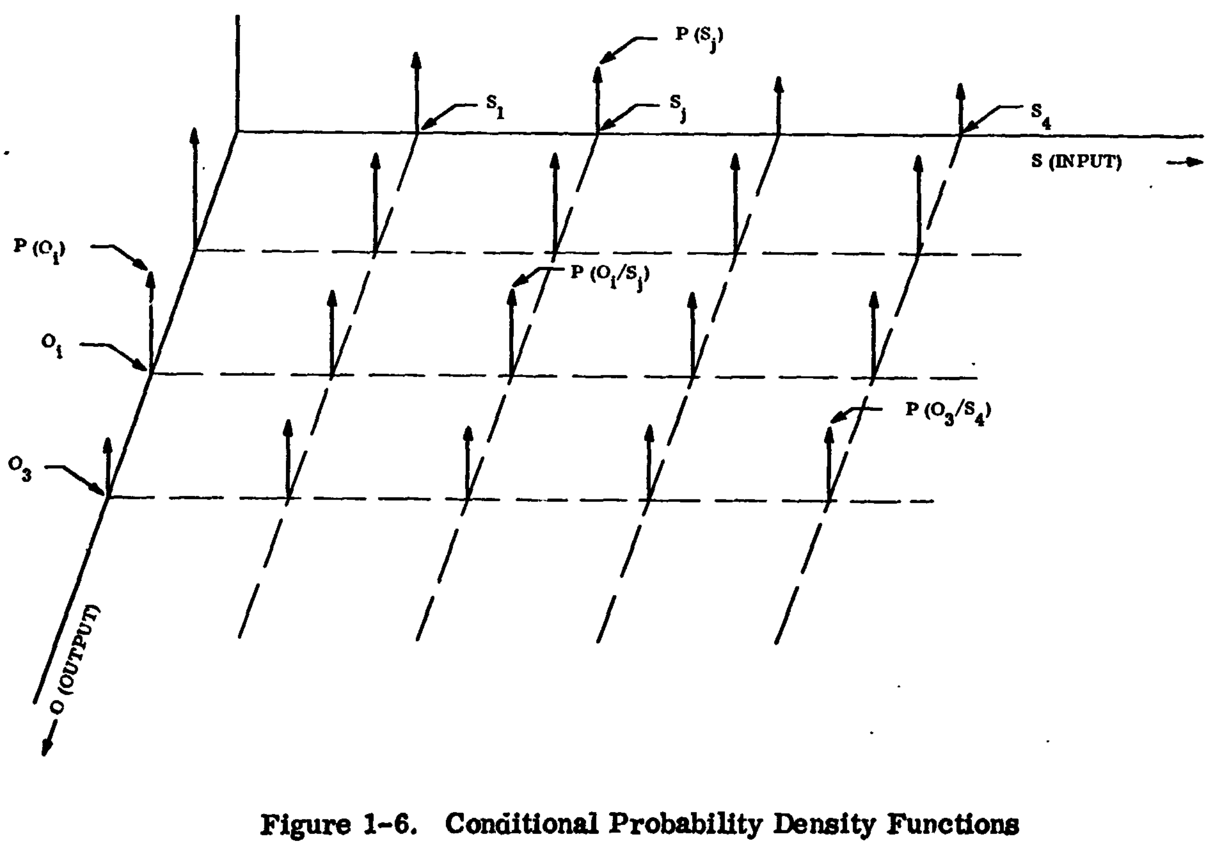


Figure 1-6. Conditional Probability Density Functions

13

The output (or marginal) distribution is computed as follows:

$$p(O_1) = p(O_1 | S_1) \cdot p(S_1) + p(O_1 | S_2) \cdot p(S_2) + p(O_1 | S_3) \cdot p(S_3) + p(O_1 | S_4) \cdot p(S_4)$$

$$p(O_2) = p(O_2 | S_1) \cdot p(S_1) + p(O_2 | S_2) \cdot p(S_2) + p(O_2 | S_3) \cdot p(S_3) + p(O_2 | S_4) \cdot p(S_4)$$

$$p(O_3) = p(O_3 | S_1) \cdot p(S_1) + p(O_3 | S_2) \cdot p(S_2) + p(O_3 | S_3) \cdot p(S_3) + p(O_3 | S_4) \cdot p(S_4)$$

In general notation, the above can be written in one statement as follows:

$$p(O_i) = \sum_{j=1}^{j=4} p(O_i | S_j) \cdot p(S_j) \quad ; \qquad i = 1, 2, 3$$

b. Given a range of number of viable organisms that are subjected to a certain kill mechanism (X) (prob. density function on number), what are the numbers of viable organisms (Y) that survive the kill mechanism? This computation is represented graphically by Figure 1-7.

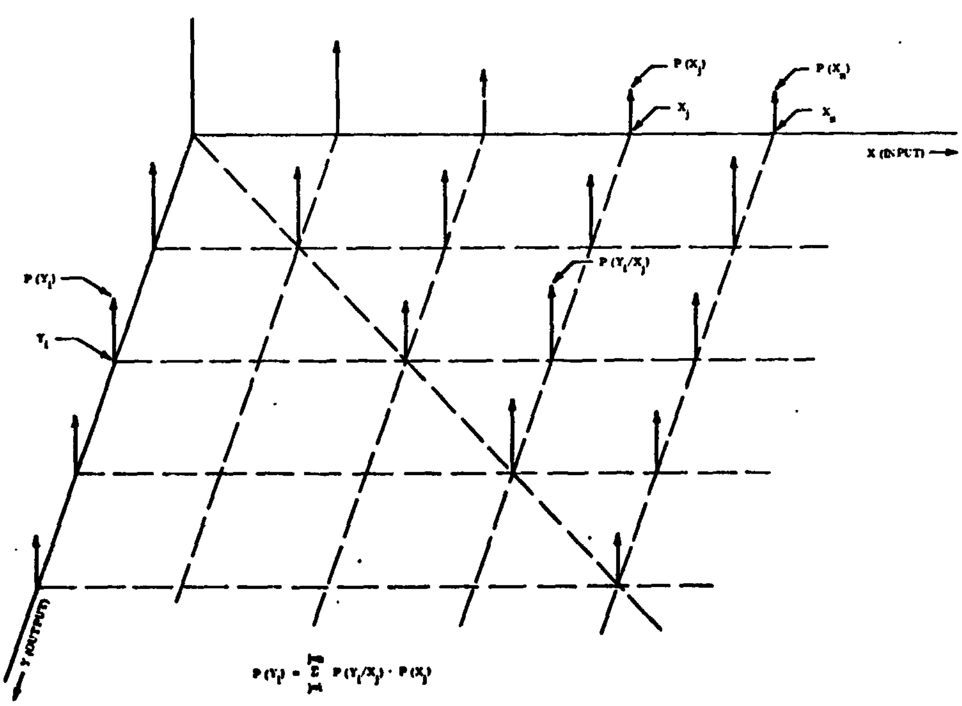


Figure 1-7. Conditional Probability Density Functions

14

1.2.2 SUMMATION OF PROBABILITY DENSITY FUNCTIONS

This computation arises, primarily, in arriving at the total number of viable organisms that will reach the surface of Mars as a result of all possible contamination sources. Conceptually, the computation consists of adding all possible combinations of numbers from the several probability density functions, computing the probability of each combination, establishing groups of common sums of numbers, and then computing the probability of each group of common sums. A very simplified illustration of this is given in Figure 1-8 and Tables 1 and 2. Two discrete probability density functions are considered for this purpose.

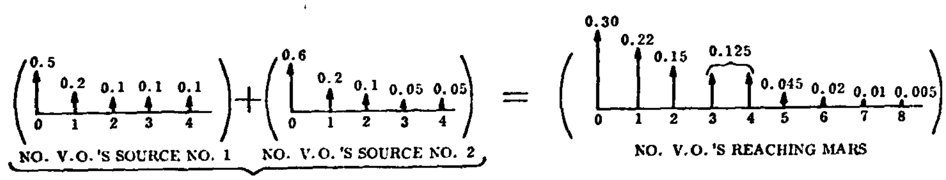


TABLE 1

ORIGINAL COMBINATIONS

$(V.O._1+V.O._2)$	$P(V.O._1+V.O._2)$
0+0=0	0.5·0.6 =0.30
0+1=1	0.5·0.2 =0.10
0+2=2	0.5·0.1 =0.05
0+3=3	0.5·0.05=0.025
0+4=4	0.5·0.05=0.025
1+0=1	0.2·0.6 =0.12
1+1=2	0.2·0.2 =0.04
1+2=3	0.2·0.1 =0.02
1+3=4	0.2·0.05=0.01
1+4=5	0.2·0.05=0.01
2+0=2	0.1·0.6 =0.06
2+1=3	0.1·0.2 =0.02
2+2=4	0.1·0.1 =0.01
2+3=5	0.1·0.05=0.005
2+4=6	0.1·0.05=0.005
3+0=3	0.1·0.6 =0.06
3+1=4	0.1·0.2 =0.02
3+2=5	0.1·0.1 =0.01
3+3=6	0.1·0.05=0.005
3+4=7	0.1·0.05=0.005
4+0=4	0.1·0.6 =0.06
4+1=5	0.1·0.2 =0.02
4+2=6	0.1·0.1 =0.01
4+3=7	0.1·0.05=0.005
4+4=8	0.1·0.05=0.005
	SUM = 1.000

TABLE 2

COMMON SUMS OF NUMBERS GROUPED

$V.O._1+V.O._2$	$P(V.O._1+V.O._2)$
0	0.30
1	0.22 = (0.1+0.12)
2	0.15 = (0.05+0.04+0.06)
3	0.125 = (0.025+0.02+0.02+0.06)
4	0.125 = (0.025+0.01+0.01+0.02+0.06
5	0.045 = (0.01+0.005+0.01+0.02)
6	0.02 = (0.005+0.005+0.01)
7	0.01 = (0.005+0.005)
8	0.005
SUM =	1.000

Figure 1-8. Summation of Probability Density Functions

1.2.3 COMBINATION OF PROBABILITY DENSITY FUNCTIONS ACCORDING TO SPECIFIED EQUATION

This type computation arises in the interplanetary trajectory analysis, orbit mechanics analysis, atmosphere entry heating analysis, etc. Conceptually, the computation is done in the same manner as for the summation of random variables. Most equations of interest, obviously, are much more complex than a simple summation, but the technique is essentially the same.

1.3 IMPLEMENTATION

There are four basic ways of implementing the computational process described in the previous section.

1.3.1 CLOSED FORM SOLUTION

This approach requires that the probability density functions be described in closed mathematical form. The data from which these density functions are derived rarely lend themselves to a closed form mathematical description. However, some density functions may be described in this form either directly or through curve fitting.

Even when all of the density functions of interest are known in closed mathematical form, it is generally not possible to perform the necessary mathematical integrations in order to arrive at a closed form solution of the problem of interest. This difficulty becomes greatly magnified when the density functions must be combined according to some complex equation.

1.3.2 MONTE CARLO SIMULATION

Conceptually, this is a very simple technique in terms of formulating the problem. It consists, essentially, of randomly selecting a value from each density function, operating on the set of values in the appropriate manner (i.e., summation, multiplication, division, or by some complex equation) and then repeating this process a sufficient number of times until the true density solution has been closely simulated.

The shortcoming of this approach is that, when a large number of random variables are involved, the number of samples required by the Monte Carlo approach to simulate the true answer is extremely large, thus requiring a large amount of high speed computer time. A further difficulty is that there is no technique available to determine in advance just how large the sample must be in order to approach the true answer with a given level of accuracy.

1.3.3 NUMERICAL APPROACH (DISCRETE VALUES)

This technique calls for combining all possible values of the parameters and computing the probability of each combination so that an output value is generated with an associated

16

probability. After all combinations have been generated, the output values are then grouped into common groupings and the probability of each group is computed, thus describing the probability density function for the output. The total number of combinations can be extremely large if there are a large number of parameters to be considered simultaneously. Quite often the approach becomes impractical for this reason.

1.3.4 NUMERICAL APPROACH (INTERVAL CONCEPT)

This approach is also a straightforward technique of using discrete approximations of continuous functions, considering all possible combination of values, computing the probability of each combination of values, operating on the set of values appropriately to generate the output value, grouping similar output values and then computing the probability of each group. Each density function is divided into intervals so that the combination of values referred to above are combinations of intervals of values and not combinations of discrete values. Furthermore, the density functions are combined pairwise in such a manner as to reduce the total number of combinations under consideration.

To illustrate the interval approach a transfer function ($V = W \cdot X + Y/Z$), such as shown in Figure 1-9, is assumed. W, X, Y and Z each have probability density functions as shown. The density functions are truncated at lower and upper values and the parameters W, X, Y, Z may take on any values within the range of their respective lower and upper values. The figures should be interpreted as follows: Looking first at the probability density function on W---"The probability that the value lies between 2 and 3 is 0.1; between 3 and 4 is 0.7; and between 4 and 5 is 0.2.

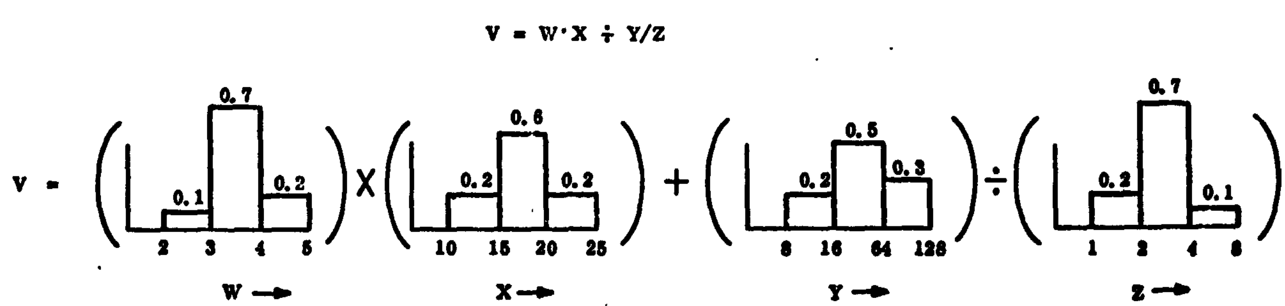
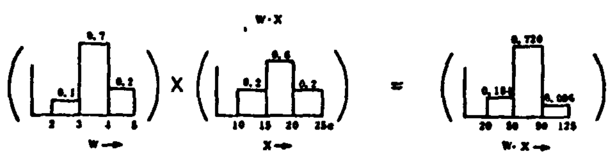


Figure 1-9. Intervals of Variables Problem

17

The probability density functions on X, Y, and Z are interpreted in the same way as those for W. However, there is one difference. The widths of the intervals on W and X are not uniform. As stated earlier, any appropriate widths of intervals may be used - they need not be uniform.

Figure 1-10 illustrates the combination of parameters by considering the first part of the transfer function V = W · X + Y/Z ---; that is, W · X. Consider first the interval 2 -3 on W and the interval 10 - 15 on X. All values of W · X resulting from these intervals will lie in a new interval having as a lower value 2 · 10 = 20, and an upper value 3 · 15 = 45. The probability associated with this new interval is 0.1 · 0.2 = 0.02. This is simply the probability that the value of W lies between 2 and 3 and the value of X lies between 10 and 15. The output of this combination of intervals is shown in Row 1 of the table. Columns 1 and 2 show the lower and upper limits of the new interval generated by combining the first interval of W with the first interval of X. Column 3 shows the probability that the new value W · X lies in the new interval. Columns 4, 5 and 6 are representative of the intervals into which the outputs of the transfer of W · X are to be grouped. The probability shown in Column 3 that is associated with the interval indicated by Columns 1 and 2 is appropriately prorated into Columns 4, 5 and 6. This process is repeated for each combination of intervals. Because W has three intervals and X has three intervals, there are nine combinations of intervals to be considered.



			20/50	50/90	90/125
2,0	45	0.02	0.020	—	—
30	60	0.06	0.040	0.020	—
40	75	0.02	0.006	0.014	—
30	60	0.14	0.047	0.093	—
45	90	0.42	0.060	0.360	—
60	150	0.14	—	0.105	0.035
40	75	0.04	0.011	0.029	—
60	100	0.12	—	0.90	0.030
80	125	0.04	—	0.009	0.031
		1.00	0.104	0.720	0.096

(W · X)

PROB. (W · X)

Figure 1-10. Intervals of Variables Solution

The intervals into which the outputs of the transfer of W· X are to be grouped have been arbitrarily designated as 20-50, 50-70 and 70-125.

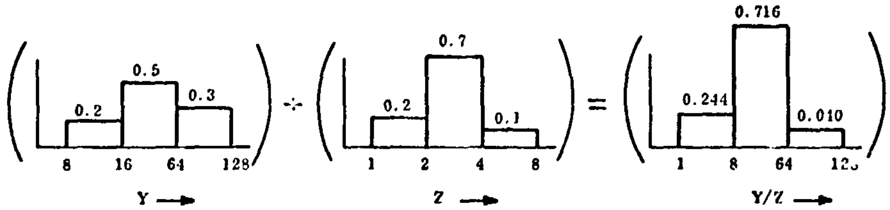
The new interval generated by the first combination of intervals is 20-45 with an associated probability of 0.02 (this is shown in the first three columns - Row 1 of the table). Because this interval is wholly contained by the first interval of the output density function, the entire 0.02 is put into the 20-50 interval.

Consider now the second combination of intervals (i.e., - W → 2-3 and X → 15-20). The new interval generated by this combination will have as its lower limit 2 · 15 = 30 and as its upper limit 3 · 20 = 60. The probability associated with this interval 30-60 is 0.1 · 0.6 = 0.06 (this is shown in the first three columns - Row 2 of the table). This probability is prorated to the 20-50 interval by the ratio 30-50/30-60, and to the 50-70 interval by the ratio 50-60/30-60 (i.e., $\frac{20}{30}$ x 0.06 = 0.04 is put into the 20-50 output interval and $\frac{10}{30}$ x 0.06 = 0.02 is put into the 50-70 output interval). The remainder of the nine combinations is done exactly in the same manner. The probabilities in each of the output intervals are now totaled thus defining the output probability density function. That is, the probability that the value of the output, W · X, lies in the interval.

- 20-50 is 0.184
- 50-70 is 0.720
- 70-125 is 0.096

The solution generated by this technique approaches the exact solution as we consider more and more intervlas on the input parameters (i.e., as the width of the intervals approach zero). This allows the analyst to test for convergence to "sufficient" accuracy as the number of intervals considered are increased.

The rest of the computations for generating the output probability density function on V, when V has been defined by V = W · X + Y/Z (shown in Figures 1-11 and 1-12), are accomplished in basically the same manner as we have shown here. Figure 1-13 graphically represents the total process. First operate on W and X according to the transfer function to produce the output W · X. Then operate on Y and Z according to the transfer function to produce Y/Z. Finally, operate on (W · X) and (Y/Z) according to the transfer function to produce (V = W · X + Y/Z).



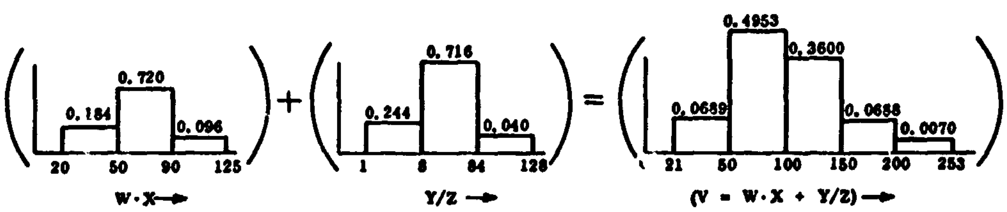
			1/8	8/64	64/128
4	16	0.04	0.013	0.027	–
2	8	0.14	0.140	–	–
1	4	0.02	0.020	–	–
8	64	0.10	–	0.100	–
4	32	0.35	0.050	0.300	–
2	16	0.05	0.021	0.029	–
32	128	0.06	–	0.020	0.040
16	64	0.21	–	0.210	–
8	32	0.03	–	0.030	–
		1.00	0.244	0.716	0.040

← (Y/Z)

← PROB. (Y/Z)

Figure 1-11. Intervals of Variables Solution

V = W·X + Y/Z



			21/50	50/100	100/150	100/200	200/253
21	58	0.0449	0.0352	0.0097	—	—	—
28	114	0.1317	0.0337	0.0766	0.0214	—	—
84	178	0.0074	—	0.0013	0.0039	0.0022	—
51	98	0.1757	—	0.1757	—	—	—
58	154	0.5156	—	0.2255	0.2686	0.0215	—
114	218	0.0268	—	—	0.0100	0.0138	0.0050
91	133	0.0234	—	0.0050	0.0184	—	—
88	189	0.0688	—	0.0015	0.0378	0.0295	—
154	253	0.0038	—	—	—	0.0018	0.0020
		1.0000	0.0689	0.4952	0.3600	0.0688	0.0070

← (V = W·X + Y/Z)

← PROB. (V = W·X + Y/Z)

Figure 1-12. Intervals of Variables Solution

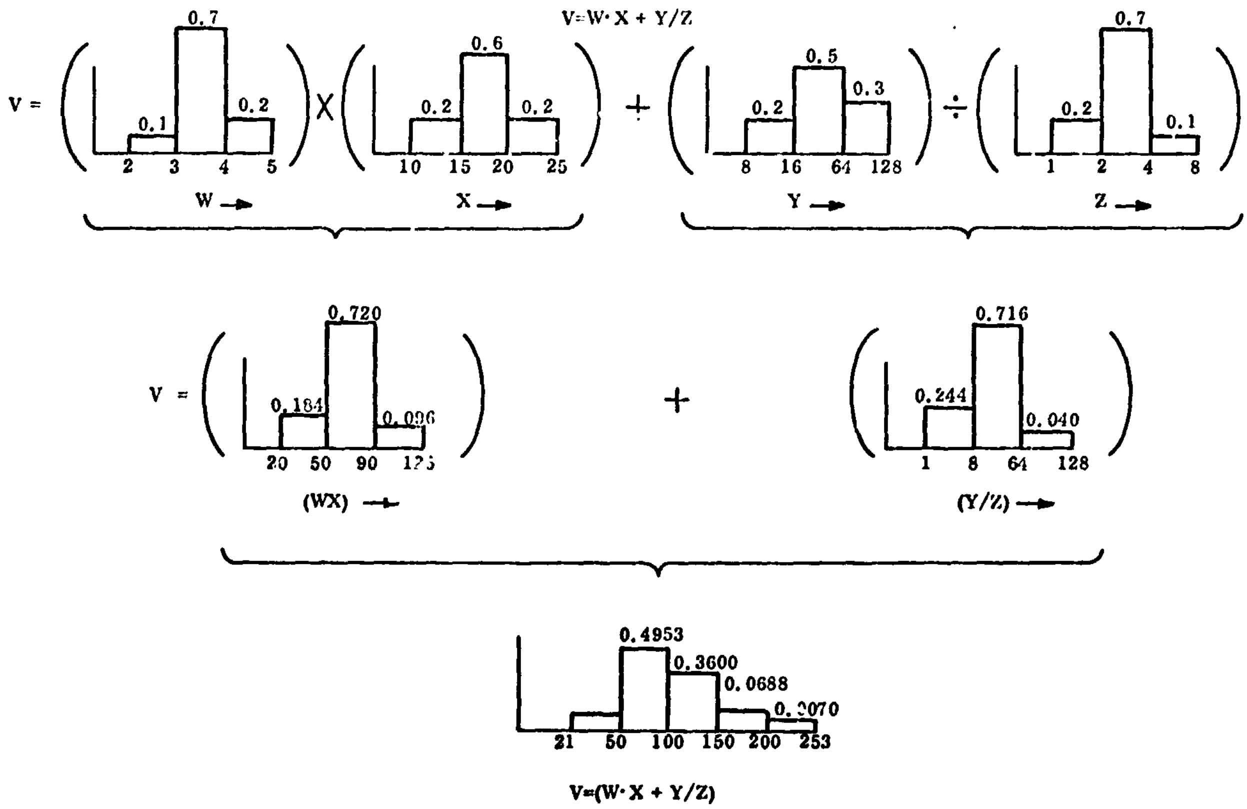


Figure 1-13. Intervals of Variables Solution

Not that the input and output density functions may have any number of intervals - they may vary in number from density function to density function. In the case of some parameters, a large number of intervals may be appropriate while in the case of other parameters a small number of intervals may be adequate.

The following is a brief summary of the computation methods discussed above.

a. An analytical solution - that is, arriving at the output density function as a mathematical expression in closed form - is generally not practical. The input functions usually cannot be expressed as closed form mathematical expressions. Even when it is possible to do so, the solution for arriving at the output (i.e., the multiple integrations that must be performed) generally becomes intractable.

b. A numerical solution using a Monte Carlo simulation technique is at best questionable. Because this is basically a sampling technique, the question is - how many samples must be taken in order to "adequately" simulate the output? A good method for arriving at an answer to this question is not available. Industry's experience has indicated that, in general, a very large number of samples are required, thus requiring a lot of computer time. The problem gets especially severe when there are a large number of input random variables to be considered.

c. A numerical approach whereby an input probability density function is represented by a set of discrete values each with an associated probability of occurrence---all possible combinations of values from the parameters are generated with its associated probability. The concept is very simple but there may also be time of computation drawbacks associated with this technique. For example, let us consider the problem of arriving at the total number of viable organisms reaching the surface of Mars. Assuming that we have done this for each possible source of contamination and further assuming that there are 100 sources, we must now sum up over all 100 sources. If we were to consider all possible combinations of sums, we would have a very large number to compute. If each density function were described by only 3 discrete values there would be 3^{100} sums to compute. (This is approximately equal to 10^{50}.) If each calculation were to take 10^{-5} seconds on a computer, the total number of seconds of computer time required would be $10^{50}/10^{5} \cong 10^{45}$ seconds or approximately 10^{38} years of continuous computer operation. If each density function was represented by 100 discrete values, the total number of sums to compute would be 100^{100} or 10^{200}. It is immediately obvious that this is not the method to use.

d. If the operation of summing was done pairwise, the total number of sums to be computed would be drastically reduced. Using the same example as before (i.e., 100 density function each having 3 discrete values) the technique is to:

- sum the first two density functions thus generating 3^2 or 9 sums
- group these values and approximate the resulting density function by 3 discrete values
- continue this process until the 100th density function has been added thus defining the final total.

The total number of sums that would have been generated is $(100-1) \cdot 3^2$ or approximately 900. It is true that three points are not very representative of most probability density functions. Therefore, consider 100 points on each density function. The pairwise approach would yield a total of $(100-1)\ 100^2$ or approximately 10^6 sums. If, as before, each calculation took 10^{-5} sec of computer time, the time required would be $10^6/10^5 = 10$ sec. It is obvious then, that this is well within practical limits.

Now if the "interval concept" is considered which considers density functions pairwise, we see that the number of calculations are even further reduced. For example, if we consider 20 intervals of values instead of 100 discrete values, we reduce the last number of calculations by a factor of $\left(\frac{20}{100}\right)^2 = \frac{1}{25}$. It should be remembered that the interval approach allows you to constantly know the upper and lower bounds of the true value of the output. This feature is required so that the analyst will have confidence in the accuracy of his solution, given a set of input data.

1.4 COMPUTER PROGRAM DEVELOPMENT

To implement the calculations that have been described, a set of programs have been developed that may be used on the remote access computer system. (also called the Desk Side Computer System). The computational difficulties are somewhat minimized when one considers that high speed computers are available for the task. The problem is further minimized when one considers that desk side computer systems are available with the capability of having many different programs in storage for immediate callup. This type of system enables the analyst to work in his own office (or work area). An advantage of this is that he has the appropriate information and material immediately (and conveniently) available to him and he can work at the computer as problems arise rather than storing up problems for one big computer run.

A group of subprograms, such as described in Section 2 of this report, enables the analyst to work on the total problem in pieces. This allows him to (1) become completely familiar with the inner workings of the program and (2) give him the ability to make sensitivity studies on parameters within confined areas of the total problem. A further advantage of having a group of subprograms, rather than one large integrated program, is that the analyst may wish to change the program in certain areas. It is very difficult to do this when the program has been developed as one integrated program.

SECTION 2

MECHANIZATION

By

T. Green

2

TABLE OF CONTENTS

SECTION 2

THE DESK SIDE VERSION OF THE VOYAGER PLANETARY QUARANTINE MATH MODEL

2.1 BASIC NUMERICAL PROCESS

The math model consists of a package of programs which help to provide insight into the various sensitivities of the effect of parameter values on random variables.

This section describes the key concepts involved in the development of the computer programs and contains a brief explanation of how to use each program.

The basic numerical process incorporated in many of the programs is the method of loading an interval probability onto a selected grid pattern. This technique is used for both discrete and continuous or interval probabilities. In the following runs the probability of 1. at 0., 0. to 1., 3., 0. to 3., 0. to4., .5 to 1.2, 1. to3.5, and 1.5 to 3, respectively, were loaded into the basic grid pattern:

0., 0., 1., 1., 2., 2., 3., 3., 4., 4.

Repeated numbers indicate that a discrete probability could be assigned. The logic is as follows:

If a discrete probability is given, it will either be loaded directly onto a discrete grid point or added into the appropriate interval. For an interval, no probability will be added onto discrete points but will be apportioned proportionately onto the grid pattern.

```
PROGRAM TO LOAD PROBABILITIES

NUMBER, OUTPUT GRID VALUES:=10,0.,0.,1.,1.,2.,2.,3.,3.,4.,4.

INTERVAL START, END, PROBABILITY:=0.,0.,1.


   RESULTING PROBABILITIES
 1.000000E+00  0.000000E-01  0.000000E-01  0.000000E-01  0.000000E-01
 0.000000E-01  0.000000E-01  0.000000E-01  0.000000E-01



INTERVAL START, END, PROBABILITY:=0.,1.,1.


   RESULTING PROBABILITIES
 0.000000E-01  1.000000E+00  0.000000E-01  0.000000E-01  0.000000E-01
 0.000000E-01  0.000000E-01  0.000000E-01  0.000000E-01



INTERVAL START, END, PROBABILITY:=3.,3.,1.


   RESULTING PROBABILITIES
 0.000000E-01  0.000000E-01  0.000000E-01  0.000000E-01  0.000000E-01
 0.000000E-01  1.000000E+00  0.000000E-01  0.000000E-01



INTERVAL START, END, PROBABILITY:=0.,3.,1.


   RESULTING PROBABILITIES
 0.000000E-01  3.333333E-01  0.000000E-01  3.333333E-01  0.000000E-01
 3.333333E-01  0.000000E-01  0.000000E-01  0.000000E-01
```

```
INTERVAL START, END, PROBABILITY:=0.,4.,1.

    RESULTING PROBABILITIES
 0.000000E-01  2.500000E-01  0.000000E-01  2.500000E-01  0.000000E-01
 2.500000E-01  0.000000E-01  2.500000E-01  0.000000E-01


INTERVAL START, END, PROBABILITY:=.5,1.2,1.

    RESULTING PROBABILITIES
 0.000000E-01  7.142857E-01  0.000000E-01  2.857143E-01  0.000000E-01
 0.000000E-01  0.000000E-01  0.000000E-01  0.000000E-01


INTERVAL START, END, PROBABILITY:=1.,3.5,1.

    RESULTING PROBABILITIES
 0.000000E-01  0.000000E-01  0.000000E-01  4.000000E-01  0.000000E-01
 4.000000E-01  0.000000E-01  2.000000E-01  0.000000E-01


INTERVAL START, END, PROBABILITY:=1.5,3.,1.

    RESULTING PROBABILITIES
 0.000000E-01  0.000000E-01  0.000000E-01  3.333333E-01  0.000000E-01
 6.666667E-01  0.000000E-01  0.000000E-01  0.000000E-01
```

2.2 BINOMIAL PROBABILITIES

A fundamental distribution which appears to be basic to much of our study is the binomial. If the probability that an organism survives an event is θ, then in repeated trials (n), the probability of survival X times is

$$f(x) = \binom{n}{x} \theta^{x} (1-\theta)^{n-x}$$

The mean of this distribution is $n\theta$, which gives an indication of its shape. For smaller θ the distribution piles up about $x = 0$ and becomes increasingly skewed to the right.

The numerical evaluation of this distribution becomes difficult if the above expression is used directly.

Since $$\frac{f(x+1)}{f(x)} = \left(\frac{n-x}{x+1}\right)\left(\frac{\theta}{1-\theta}\right) = R(x)$$

The recursion $f(x+1) = R(x)\, f(x)$

where $f(o) = (1-\theta)^{n}$ allows us to evaluate this distribution without the calculation of factorials or other combinatorial formulas.

A program called "BINOM" performs the above.

For large n, the recursion becomes less attractive for obvious reasons. In this case, if

$$\frac{1}{n+1} < \theta$$

The binomial can be approximated by the normal in a 3σ range about the mean. Denoting the cumulative values of the normal by $\Phi(x)$, the approximation is

$$\Pr\left\{a \le x \le b\right\} \triangleq \Phi\left(\frac{b+\frac{1}{2}-\mu}{\sigma}\right) - \Phi\left(\frac{a+\frac{1}{2}-\mu}{\sigma}\right)$$

$$\mu = n\theta, \quad \sigma = \sqrt{n\theta(1-\theta)}$$

A program called "BINOMX" performs the above calculations by utilizing a technique developed and described in Reference 3.

It is recommended that "BINOMX" be used for $n > 100$.

The two following sets of examples illustrate the output of BINOM and BINOMX.

In BINOMX, the equivalent upper and lower bounds for the given first and last values are in terms of the standard normal distribution.

Note that essentially all (99.99 percent approximately) of the probability lies between ± 3.

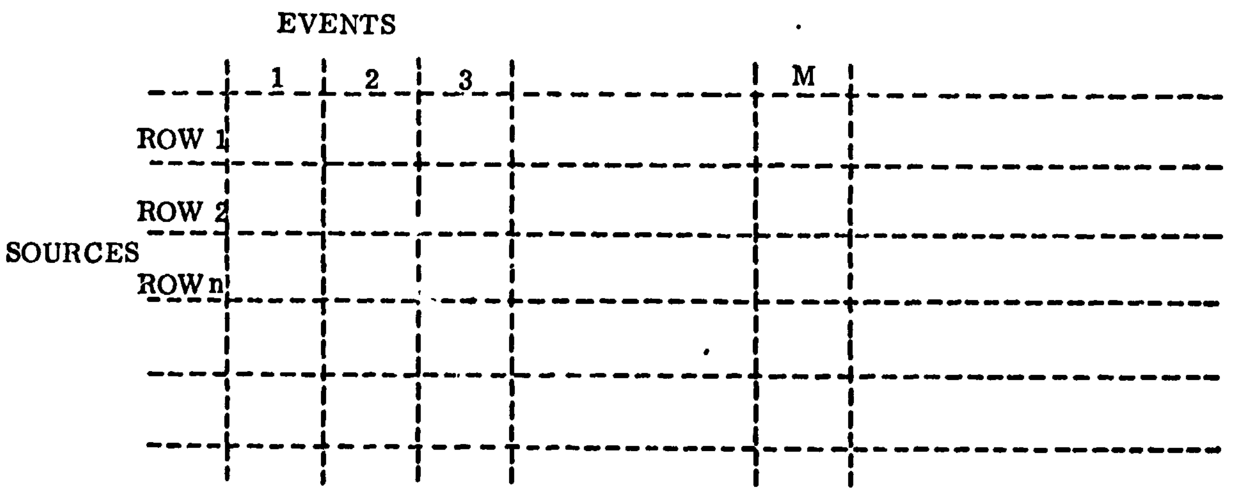
```
BINOMIAL PROBABILITIES

READ PROBABILITY AND NUMBER:=.1,0

     VALUE  PROBABILITY
         0  1.000000E+00

   SUM=        1.0000000

READ PROBABILITY AND NUMBER:=.1,3

     VALUE  PROBABILITY
         0  7.290000E-01
         1  2.430000E-01
         2  2.700000E-02
         3  1.000000E-03

   SUM=        1.0000000

READ PROBABILITY AND NUMBER:=.1,7

     VALUE  PROBABILITY
         0  4.782969E-01
         1  3.720087E-01.
         2  1.240029E-01
         3  2.296350E-02
         4  2.551500E-03
         5  1.701000E-04

         6  6.300000E-06
         7  1.000000E-07

   SUM=        1.0000000
```

```
READ PROBABILITY AND NUMBER:=.1,10

     VALUE  PROBABILITY
         0  3.486784E-01
         1  3.874205E-01
         2  1.937102E-01
         3  5.739563E-02
         4  1.116026E-02
         5  1.488035E-03
         6  1.377810E-04
         7  8.748000E-06
         8  3.645000E-07
         9  9.000000E-09
        10  1.000000E-10

   SUM=        1.0000000
```

```
 NORMAL APPROXIMATION TO BINOMIAL

READ FIRST,LAST VALUE, NUMBER, PROBABILITY:=10.,50.,550.,.1

 STANDARD NORMAL LIMITS***** -0.646709E+01 -0.639602E+00
 PROBABILITY=  0.261216E+00

READ FIRST,LAST VALUE, NUMBER, PROBABILITY:=10.,100.,550.,.1

 STANDARD NORMAL LIMITS***** -0.646709E+01  0.646709E+01
 PROBABILITY=  0.100000E+01

READ FIRST,LAST VALUE, NUMBER, PROBABILITY:=20.,30.,550.,.1

 STANDARD NORMAL LIMITS***** -0.504575E+01 -0.348228E+01
 PROBABILITY=  0.248358E-03

READ FIRST,LAST VALUE, NUMBER, PROBABILITY:=20.,25.,100.,.5

 STANDARD NORMAL LIMITS***** -0.610000E+01 -0.490000E+01
 PROBABILITY=  0.478653E-06

READ FIRST,LAST VALUE, NUMBER, PROBABILITY:=25.,75.,100.,.5

 STANDARD NORMAL LIMITS***** -0.510000E+01  0.510000E+01
 PROBABILITY=  0.100000E+01

READ FIRST,LAST VALUE, NUMBER, PROBABILITY:=30.,60.,100.,.5

 STANDARD NORMAL LIMITS***** -0.410000E+01  0.210000E+01
 PROBABILITY=  0.982115E+00

READ FIRST,LAST VALUE, NUMBER, PROBABILITY:=40.,60.,100.,.5

 STANDARD NORMAL LIMITS***** -0.210000E+01  0.210000E+01
 PROBABILITY=  0.964271E+00

READ FIRST,LAST VALUE, NUMBER, PROBABILITY:=45.,55.,100.,.5

 STANDARD NORMAL LIMITS***** -0.110000E+01  0.110000E+01
 PROBABILITY=  0.728668E+00

READ FIRST,LAST VALUE, NUMBER, PROBABILITY:=47.,53.,100.,.5

 STANDARD NORMAL LIMITS***** -0.700000E+00  0.700000E+00
 PROBABILITY=  0.516073E+00

READ FIRST,LAST VALUE, NUMBER, PROBABILITY:=$STOP
READY.
```

2.3 BASIC COMBINATORIAL MODEL

The basic module in the numerical math model is the two-segment program titled "Basic 1, Basic 2." The primary function of the program is to combine probabilities in the following form:



Viable organisms can be thought of as being available from various sources about the spacecraft and environs. The number available conceivably could be a random variable and thus can be represented as a "row" probability distribution. Each event (represented by a column) can be thought of as altering the row (source) probability distribution. The simplest way of representing the effect of an event is to say that an organism has probability of surviving the event. If it can be hypothesized that each organism has the same probability, then the conditional distribution of survival is of a binomial nature. That is, given n organisms, the probability of x survivors is

$$pr\,(x/n) = \binom{n}{x} \epsilon^{x} (1-\epsilon)^{n-x}$$

the resulting marginal probability

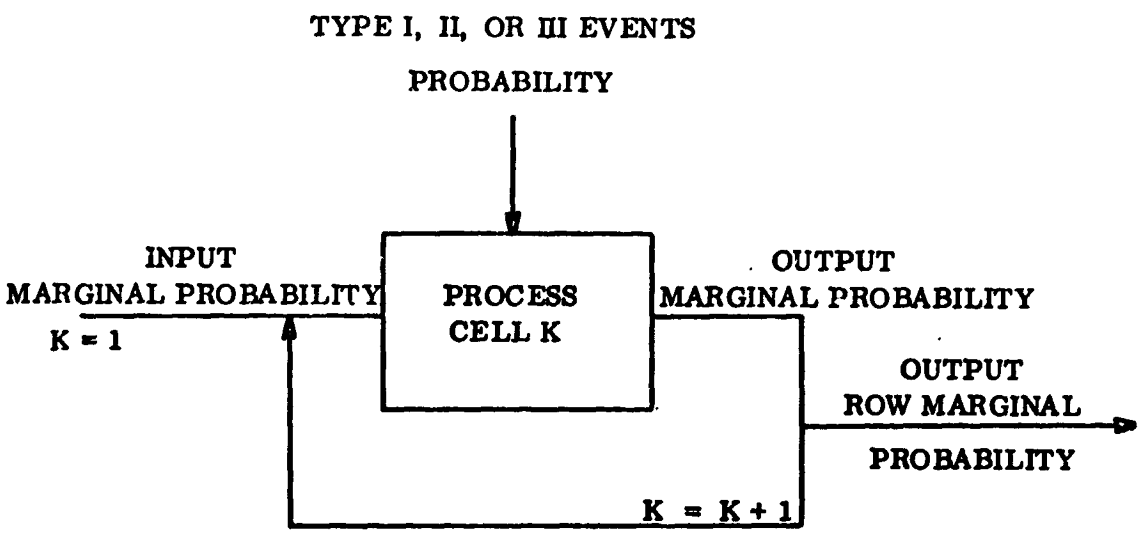
$$pr\,(x) = \sum_{n_i}^{n} pr\,(x) \binom{n_i}{x} \epsilon^{x} (1-\epsilon)^{n_i-x}$$

where N is the total number of organisms before the event. If the binomial hypothesis is not satisfactory any conditional distribution, can be used in place of pr (x/n). It is primarily the function of this program to calculate each succeeding marginal probability as one courses through the events. As a row is completed, the resulting random variables can be added by another program called "BUGS" which is described elsewhere.

1. Read in basic GRID pattern.

2. Read in Row probability source distribution along with its own grid pattern.

3. Adjust source probability distribution to standard grid pattern.



1. **Read in row probability end points.**

2. **Read in associated probabilities.**

3. **Read in the standard set of end points to be used during the course of row calculations.**

4. **Adjust the input probability distribution to the standard set of end points.**

5. Read in code word:

   1: Binomial probabilities

   2: Conditional distributions at arbitrary nature

   3: Proportion, probability

   4: Scale the standard end points

   5: Return to choose another row distribution and thence to process another row.

6. Process according to code word and print the input and output marginal distributions.

The option is described below:

OPTION CODE

1. Conditional probabilities are computed by assuming a binomial condition is satisfactory. Appropriately, the probability that an organism survives an event is input initially.

2. The conditional distributions are provided one at a time starting at the second to lowest grid value. The distributions are provided in the standard interval concept.

3. The conditional distributions are provided by inputting two numbers for each distribution (θ, ξ). Each conditional distribution has the end points σ, G(I)* θ, G(I) with associated probabilities ξ and $1. - \xi$.

4. A simple scale change is provided here. If the scale factor is q, then the new standard grid pattern is q * G(I) with the same probabilities.

5. This indicates that a new row probability is to be provided. Essentially this restarts the program.

The combinatorial technique is based on the fundamental concept of conditional and marginal probability. Reference 4 describes the approach.

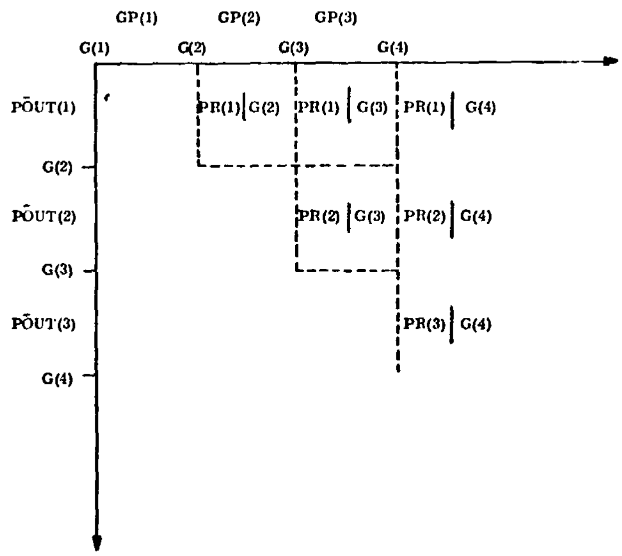
The combinatorial procedure is at best a computer approximation and requires some explanation of the method. Figure 2-1 illustrates the procedure.

GP(I) represents the input probabilities.

G(I) represents the standard grid points.

P$\bar{\text{O}}$UT(I) is the output (marginal) probability distribution.

PR(J)/G(I) describes the conditional probabilities.



The conditional distributions in option 1 are either generated by the basic binomial recursion formula or the normal approximation to the binomial (see Reference 33) if the given number is greater than 100.

Options 2 and 3 require that the conditional distributions, be input via the interval concept.

The marginal probability distribution is calculated by

$$P\bar{O}UT(J) = \sum_{I=J+1}^{NST} GP(I-1)\ (PR(J)/G(I))$$

The probabilities GP (I) are associated with G(I+1). This appears to be the most conservative approach.

It is recommended that the first value for the standard grid be zero. This is consistent with the concept of the binomial hypothesis.

BASIC COMBINATORIAL MODEL

NUMBER, ROW END POINTS:=4,0.,2.,20.,1000.

ROW PROBABILITIES:=.6,.3,.1

NUMBER, STANDARD END POINTS:=10,0.,1.,5.,10.,100.,1000.,2000.,3000.
:=4000.,5000.

READ IN CODE(1-5):=1

SIMPLE PROBABILITY:=.01

FIRST	LAST	INPUT	OUTPUT
0.000000E-01	1.000000E+00	3.000000E-01	8.614235E-01
1.000000E+00	5.000000E+00	3.500000E-01	5.175419E-02
5.000000E+00	1.000000E+01	8.333333E-02	4.088453E-02
1.000000E+01	1.000000E+02	1.748299E-01	4.593781E-02
1.000000E+02	1.000000E+03	9.183673E-02	0.000000E-01
1.000000E+03	2.000000E+03	0.000000E-01	0.000000E-01
2.000000E+03	3.000000E+03	0.000000E-01	0.000000E-01
3.000000E+03	4.000000E+03	0.000000E-01	0.000000E-01
4.000000E+03	5.000000E+03	0.000000E-01	0.000000E-01

READ IN CODE(1-5):=2

NUMBER, END POINTS FOR (1.00000E+00):=2,0.,1.

CONDITIONAL PROBABILITIES:=1.

NUMBER, END POINTS FOR (5.00000E+00):=4,0.,2.,3.,5.

CONDITIONAL PROBABILITIES:=.7,.2,.1

NUMBER, END POINTS FOR (1.00000E+01):=4,0.,1.,5.,10.

CONDITIONAL PROBABILITIES:=.8,.2,0.

NUMBER, END POINTS FOR (1.00000E+02):=3,0.,30.,100.

CONDITIONAL PROBABILITIES:=.9,.1

NUMBER, END POINTS FOR (1.00000E+03):=3,0.,100.,1000.

CONDITIONAL PROBABILITIES:=.95,.05

NUMBER, END POINTS FOR (2.00000E+03):=3,0.,1000.,2000.

CONDITIONAL PROBABILITIES:=.5,.5

NUMBER, END POINTS FOR (3.00000E+03):=3,0.,2000.,3000.

CONDITIONAL PROBABILITIES:=1.,0.

NUMBER, END POINTS FOR (4.00000E+03):=2,0.,4000.

CONDITIONAL PROBABILITIES:=1.

NUMBER, END POINTS FOR (5.00000E+03):=2,0.,5000.

CONDITIONAL PROBABILITIES:=1.

FIRST	LAST	INPUT	OUTPUT
0.000000E-01	1.000000E+00	8.614235E-01	9.136232E-01
1.000000E+00	5.000000E+00	5.175419E-02	4.732967E-02
5.000000E+00	1.000000E+01	4.088453E-02	6.890672E-03
1.000000E+01	1.000000E+02	4.593781E-02	3.215647E-02
1.000000E+02	1.000000E+03	0.000000E-01	0.000000E-01
1.000000E+03	2.000000E+03	0.000000E-01	0.000000E-01
2.000000E+03	3.000000E+03	0.000000E-01	0.000000E-01
3.000000E+03	4.000000E+03	0.000000E-01	0.000000E-01
4.000000E+03	5.000000E+03	0.000000E-01	0.000000E-01

READ IN CODE(1-5):=3

PROPORTION, PROBABILITY FOR (1.00000E+00):=.3,.8

PROPORTION, PROBABILITY FOR (5.00000E+00):=.5,.65

PROPORTION, PROBABILITY FOR (1.00000E+01):=.8,.7

PROPORTION, PROBABILITY FOR (1.00000E+02):=.25,.1

PROPORTION, PROBABILITY FOR (1.00000E+03):=.1,.001

PROPORTION, PROBABILITY FOR (2.00000E+03):=1.,1.

PROPORTION, PROBABILITY FOR (3.00000E+03):=1.,1.

PROPORTION, PRO BILITY FOR (4.00000E+03):=1.,1.

PROPORTION, PROBABILITY FOR (5.00000E+03):=1.,1.

FIRST	LAST	INPUT	OUTPUT
0.000000E-01	1.000000E+00	9.136232E-01	9.266605E-01
1.000000E+00	5.000000E+00	4.732967E-02	3.795019E-02
5.000000E+00	1.000000E+01	6.890672E-03	4.519132E-03
1.000000E+01	1.000000E+02	3.215647E-02	3.087021E-02
1.000000E+02	1.000000E+03	0.000000E-01	0.000000E-01
1.000000E+03	2.000000E+03	0.000000E-01	0.000000E-01
2.000000E+03	3.000000E+03	0.000000E-01	0.000000E-01
3.000000E+03	4.000000E+03	0.000000E-01	0.000000E-01
4.000000E+03	5.000000E+03	0.000000E-01	0.000000E-01

READ IN CODE(1-5):=4

SCALE FACTOR:=20.

FIRST	LAST	INPUT	OUTPUT
0.000000E-01	1.000000E+00	9.266605E-01	4.633302E-02
1.000000E+00	5.000000E+00	3.795019E-02	1.853321E-01
5.000000E+00	1.000000E+01	4.519132E-03	2.316651E-01
1.000000E+01	1.000000E+02	3.087021E-02	5.012804E-01
1.000000E+02	1.000000E+03	0.000000E-01	1.823923E-02
1.000000E+03	2.000000E+03	0.000000E-01	1.715012E-02
2.000000E+03	3.000000E+03	0.000000E-01	0.000000E-01
3.000000E+03	4.000000E+03	0.000000E-01	0.000000E-01
4.000000E+03	5.000000E+03	0.000000E-01	0.000000E-01

READ IN CODE(1-5):=5

NUMBER, ROW END POINTS:=$STOP
READY.

2.4 ADDITION OF RANDOM VARIABLES

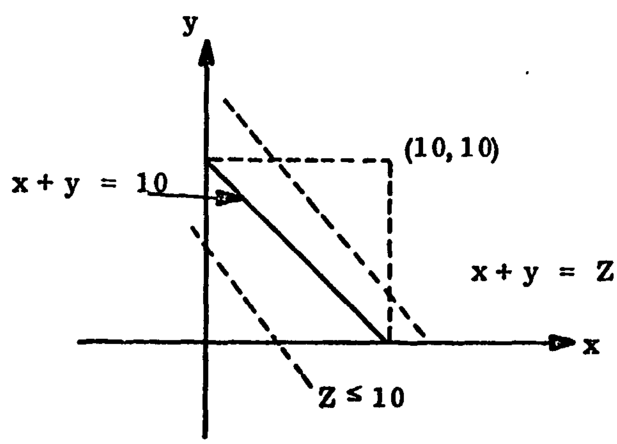
A common combinatorial problem is the computation of the distribution of the sum of two random variables. This occurs, for example, in the processing of combining row probabilities as described in Reference 4.

Consider the determination of the distribution of $Z = x + y$ where $0 \le x \le 10$, $0 \le y \le 10$, and it is assumed that the probability density functions of x and y are independent. (This is a basic assumption that will be made between rows or sources.)

The cumulative distribution

$$F(Z_o) = \Pr(Z \le Z_o) = \iint_R f(x)\, g(y)\, dx dy$$

where R is the space in the x-y plane such that $Z \le Z_o$.



The distribution can be explicitly written

$$\Pr(x+y \le Z_o) = \int_o^{Z_o} \int_o^{Z_o - x} f(x)\, g(y)\, dy\, dx \quad (o \le Z_o \le 10)$$

$$\Pr(x+y \le y \le Z_o) = 1 - \int_{Z_o-10}^{10} \int_{Z_o-x}^{10} f(x)\, g(y)\, dy\, dx \quad (10 \le Z_o \le 20)$$

and zero elsewhere.

For example, let $g(y) = \frac{1}{10}$; $f(x) = \frac{1}{10}$ for $0 \le x \le 10$ and $0 \le y \le 10$.

Then the resulting probabilities from ($0 \rightarrow 20$) could be tabulated as follows:

Interval	Probability
0 → 2	0.02
2 → 4	0.06
4 → 6	0.10
6 → 8	0.14
8 → 10	0.18
10 → 12	0.18
12 → 14	0.14
14 → 16	0.10
16 → 18	0.06
18 → 20	0.02

where $\Pr(Z_1 \le Z_o \le Z_2) = \dfrac{Z_2^2 - Z_1^2}{200} \quad (0 \le Z_o < 10)$

$$\Pr(Z_1 \le Z_o \le Z_2) = \left(\frac{-Z_2^2}{200} + \frac{Z_o Z_2}{100} - 1\right) - \left(\frac{-Z_1^2}{200} + \frac{20Z_1}{100} - 1\right)$$

$$(10 \le Z_o < 20)$$

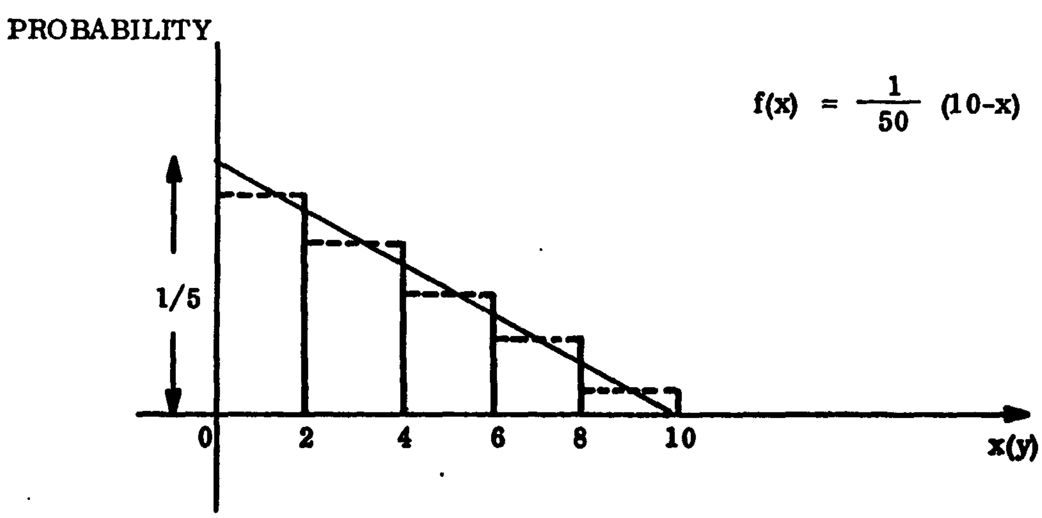
If the probabilities f(x) and g(x) where given in more complex form, say as a mixed function defined as constant over prescribed intervals, it is more advantageous from a computer standpoint to develop an algorithm to "lump" probabilities assigned over prescribed intervals into the standard grid pattern.

The above problem was checked by using the intervals given above.

The following illustrates the results of a probability "adder."

The first number is a sum for the resulting probabilities and the following are the probabilities of Z in the intervals 0 → 2, 2 → 4, etc.

```
LOAD LIMITS 06345 15505
:=6,0.,2.,4.,6.,8.,10.
:=6,0.,2.,4.,6.,8.,10.
:=11,0.,2.,4.,6.,8.,10.,12.,14.,16.,18.,20.
:=.2,.2,.2,.2,.2
:=.2,.2,.2,.2,.2
     SUM
  1.000000E+00  2.000000E-02  6.000000E-02  1.000000E-01  1.400000E-01
  1.800000E-01  1.800000E-01  1.400000E-01  1.000000E-01  6.000000E-02
  2.000000E-02



:=$STOP
READY.


  ELAPSED TIME IN HUNDREDTHS OF HOURS 003
```



Let the above triangular distribution be approximated by rectangles representing (areas) approximate probabilities of

$$\frac{18}{50},\ \frac{14}{50},\ \frac{10}{50},\ \frac{6}{50},\ \frac{2}{50}$$

The analytical probabiliti-s should be (if the above represents both x and y)

Interval	Probability
0 → 2	0.0648
2 → 4	0.1656
4 → 6	0.212
6 → 8	0.2104
8 → 10	0.1672
10 → 12	0.1032
12 → 14	0.0504
14 → 16	0.02
16 → 18	0.0056
18 → 20	0.0008

where for $(0 \le Z_o < 10)$

$$\Pr(Z_1 \le Z_o \le Z_2) = \int_0^{Z_o} \int_0^{Z_o - x} \left(\frac{1}{50}\right)^2 (10-x)(10-y)\, dy dx$$

and for $(10 \le Z_o \le 20)$

$$\Pr(Z_1 \le Z_o \le Z_2) = 1 - \int_{Z_o-10}^{10} \int_{Z_o-x}^{10} \frac{1}{50} (10-x)(10-y)\, dy dx.$$

$$\Pr(Z_1 \le Z_o \le Z_2) \doteq 1 - \int_{Z_o-10}^{10} \int_{Z_o-X}^{10} \frac{1}{50} (10-x)(10-y)\, dy dx.$$

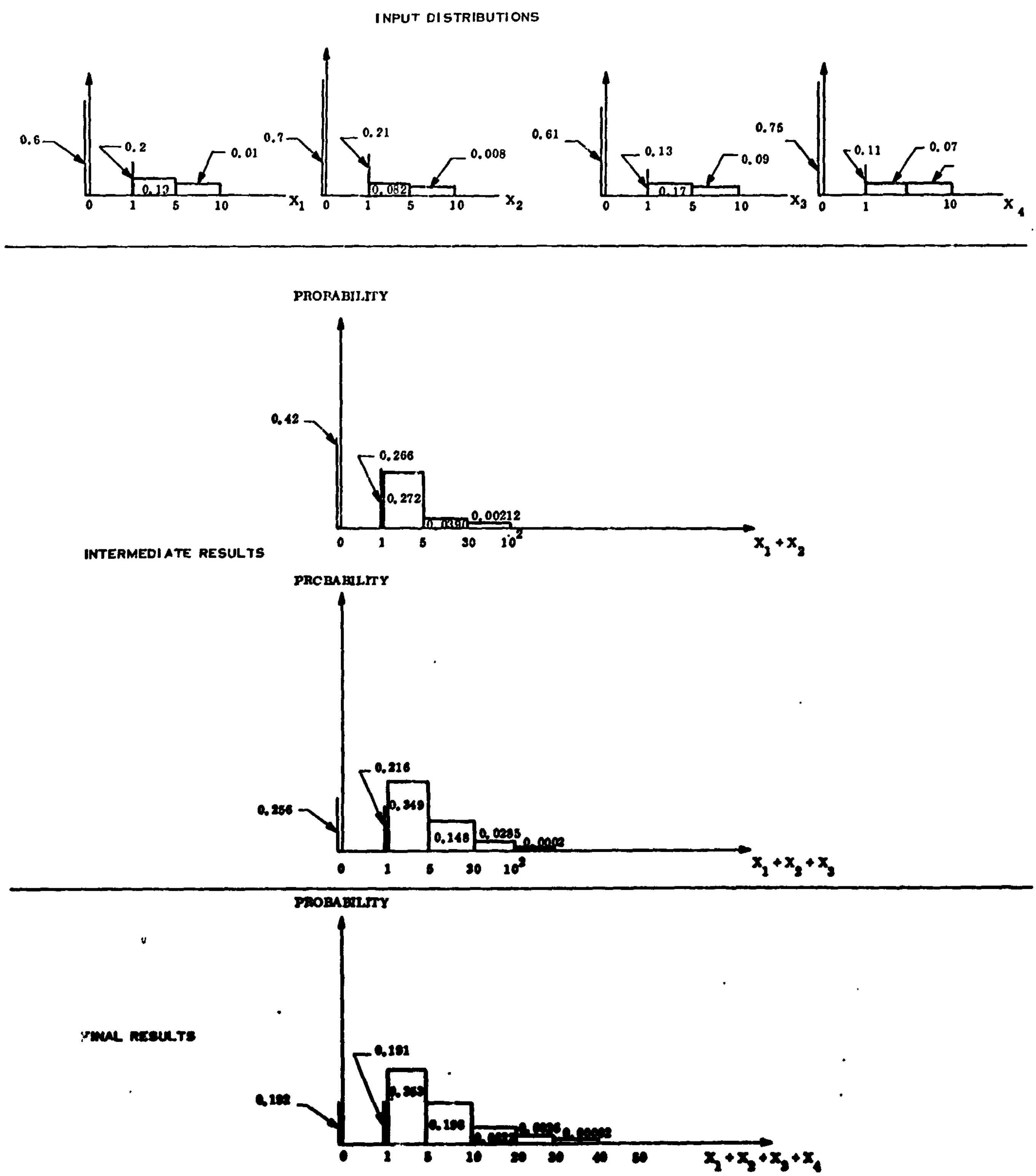
```
LOAD LIMITS 06345 15505
:=6,0.,2.,4.,6.,8.,10.
:=6,0.,2.,4.,6.,8.,10.
:=11,0.,2.,4.,6.,8.,10.,12.,14.,16.,18.,20.
:=.36,.28,.2,.12,.04
:=.36,.28,.2,.12,.04
      SUM
  1.000000E+00  6.480000E-02  1.656000E-01  2.120000E-01  2.104000E-01
  1.672000E-01  1.032000E-01  5.040000E-02  2.000000E-02  5.600000E-03
  8.000000E-04
```

The probabilities thus tend to "bunch" closer to the lower end of the scale.

For nonlinear frequency distributions, the above procedure is an approximation; however, the finer the grid, the closer the approximation to the true probability.

From these preliminary studies, a program called "BUGS" was written to handle the "mixed" (discrete plus continuous cases) probability addition problem. The program was set up to allow a recursive addition of sample spaces. An example follows which illustrates the addition of four distributions such that the probabilities are discretely defined at 0, 1 and continuously between 1-5 and 5-10. This illustrates the case when the probability of low discrete numbers is important enough to be preserved.

```
LOAD LIMITS 06622 15505

PROGRAM TO FIND PROBABILITY OF SUMS


 NUMBER,POINTS FOR FIRST DENSITY:=6,0.,0.,1.,1.,5.,10.

FIRST SET OF PROBABILITIES:=.6,0.,.2,.19,.01

NUMBER,POINTS FOR RESULTING DENSITY:=10,0.,0.,1.,1.,5.,10.,20.,30.
:=40.,50.


NUMBER,POINTS FOR NEXT DENSITY:=6,0.,0.,1.,1.,5.,10.

NEXT SET OF PROBABILITIES:=.7,0.,.21,.082,.008

RESULTING PROBABILITIES
  4.200000E-01  0.000000E-01  2.660000E-01  2.722675E-01  3.961250E-02
  2.120000E-03
CHECK SUM =  1.000000


NUMBER,POINTS FOR NEXT DENSITY:=6,0.,0.,1.,1.,5.,10.

NEXT SET OF PROBABILITIES:=.61,0.,.13,.17,.09

RESULTING PROBABILITIES
  2.562000E-01  0.000000E-01  2.168600E-01  3.498813E-01  1.482011E-01
  2.857413E-02  2.834743E-04
CHECK SUM =  1.000000


NUMBER,POINTS FOR NEXT DENSITY:=6,0.,0.,1.,1.,5.,10.

NEXT SET OF PROBABILITIES:=.75,0.,.11,.07,.07

RESULTING PROBABILITIES
  1.921500E-01  0.000000E-01  1.908270E-01  3.536343E-01  1.984907E-01
  6.225241E-02  2.622169E-03  2.343387E-05
CHECK SUM =  1.000000


NUMBER,POINTS FOR NEXT DENSITY:=$STOP
READY.
```


INPUT DISTRIBUTIONS
0.6
0.2
0.13
0.01
0.7
0.21
0.082
0.008
0.61
0.13
0.17
0.09
0.75
0.11
0.07
PROBABILITY
0.42
0.266
0.272
0.00212
INTERMEDIATE RESULTS
PROBABILITY
0.256
0.216
0.349
0.148
0.0285
PROBABILITY
FINAL RESULTS
0.191
0.363
0.198

2.5 DEPENDENT PROBABILITIES

In the calculation of the distribution of a function of other random variables in general, it is assumed that the independent variables are stochastically independent and the joint probability is simply the product.

However, if future needs require the combining of dependent or correlated variables the calculation could be modified.

A simple example can be described.

Suppose wé have 2 distributions given by

$$p(A) = \{0.6,\ 0.4\} \quad \text{at} \quad A = \{0,\ 1,\ 2\}$$
$$\text{and } p(B): \{0.1,\ 0.9\} \quad \text{at} \quad B = \{0,\ 1,\ 2\}$$

Then the distribution of the product of the variables is approximated by

$$p(0 \le Z < 1) = 0.06 + 0.54\left(\frac{1}{2}\right) + 0.04\left(\frac{1}{2}\right) = 0.35$$

$$p(1 \le Z \le 2) = 0.54\left(\frac{1}{2}\right) + 0.04\left(\frac{1}{2}\right) + 0.36\left(\frac{1}{3}\right) = 0.41$$

$$p(2 < Z \le 3) = 0.36\left(\frac{1}{3}\right) = 0.12$$

$$p(3 \le Z \le 4) = 0.36\left(\frac{1}{3}\right) = 0.12$$

Suppose, however, that the variable B depends on A. How can we describe this? One way is to consider that p(B) is not constant but "varies" with A.

That is, for example, for

$$(0 \leq A \leq 1) : \begin{cases} p\,(0 \leq B \leq 1) = 0.9 \\ p\,(1 \leq B \leq 2) = 0.1 \end{cases}$$

but for

$$(1 \leq A \leq 2) : \begin{cases} p\,(0 \leq B \leq 1) = 0.2 \\ p\,(1 \leq B \leq 2) = 0.8 \end{cases}$$

This indicates that when A is small, B is likely to be small also and vice versa. In this case the result becomes:

$$p\,(0 \leq Z \leq 1) = 0.54 + 0.06\left(\frac{1}{2}\right) + 0.08\left(\frac{1}{2}\right) = 0.61$$

$$p\,(1 \leq Z \leq 2) = 0.06\left(\frac{1}{2}\right) + 0.08\left(\frac{1}{2}\right) + 0.32\left(\frac{1}{3}\right) = 0.1766$$

$$p\,(2 \leq Z \leq 3) = 0.32\left(\frac{1}{3}\right) = 0.1066$$

$$p\,(3 \leq Z \leq 4) = 0.32\left(\frac{1}{3}\right) = 0.1066$$

Thus the independent case can be considered as a special case of dependency where p(B) is unchanged for all intervals of A.

It is anticipated that "DELP" for example, will require this treatment, since pitch down angle and the associated velocity increment conceivably have a correlation.

2.6 MARGINAL PROBABILITY CALCULATION

Probability input appears to be frequently in the form of the probability of surviving a specified event. Also the hypothesis that the probability of an organism surviving remains unchanged from one organism to another seems to be a reasonable assumption in many cases.

The marginal distribution of a random variable x can be written in density function form as

$$f_1(x) = \int f_2(y)\ f(x \mid y)\ dy$$

where y may be thought of as the given random variable and x as the new random variable after suffering the effects of an event.

Numerically this can be approximated by considering

y in interval form
with the following distribution:

$$Pr\ \{yo < y \le y1\} = Py1$$
$$Pr\ \{y1 < y \le y2\} = Py2$$

$$Pr\ \{y_{n-1} < y \le yn\} = Pyn$$

Under the binomial hypotheses, the conditional distributions $f(x/y)$ can be generated using the numerical techniques in "BINOM" and "BINOMX."

The resulting marginal distribution which approximates f1(x) is

$$Pr\{x_{j-1} < x \le y_j\} = \sum_{k=1}^{n} Pr\{y_{k-1} < y \le y_k\}\ Pr\left\{x_{j-1} < x \le x_j \;\middle|\; \begin{matrix}\text{given} \\ y_{x-1} < y \le y_k\end{matrix}\right\}$$

A program called "ICBMAR" was written, which computes $f_1(x)$ over integer values for a selected y values up to or less than 100.

47

```
:=55,4,.1,1.
:=.6,.2,.1,.1
:=0,3,7,55
```

0	7.939340E-01
1	8.766064E-02
2	2.337959E-02
3	1.344830E-02
4	1.607464E-02
5	1.794576E-02
6	1.660133E-02
7	1.291166E-02
8	8.607769E-03
9	4.994631E-03
10	2.552812E-03
11	1.160369E-03
12	4.727429E-04
13	1.737431E-04
14	5.791437E-05
15	1.758881E-05
16	4.885780E-06
17	1.245395E-06
18	2.921297E-07
19	6.320934E-08
20	1.264187E-08
21	2.341087E-09
22	4.020048E-10
23	6.408772E-11
24	9.494477E-12
25	1.308128E-12
26	1.677087E-13
27	2.001462E-14
28	2.223847E-15
29	2.300531E-16
30	2.215326E-17
31	1.985059E-18
32	1.654216E-19
33	1.281043E-20
34	9.210112E-22
35	6.140074E-23
36	3.790169E-24
37	2.162559E-25
38	1.138189E-26
39	5.512596E-28
40	2.450043E-29
41	9.959524E-31
42	3.688712E-32
43	1.239102E-33
44	3.754856E-35
45	1.019837E-36
46	2.463375E-38
47	5.2412[illegible]4E-40
48	9.705970E-42
49	1.540630E-43
50	2.054174E-45
51	2.237662E-47
52	1.912531E-49
53	1.202850E-51
54	4.950000E-54
55	[illegible]E-56

1.000000E+00

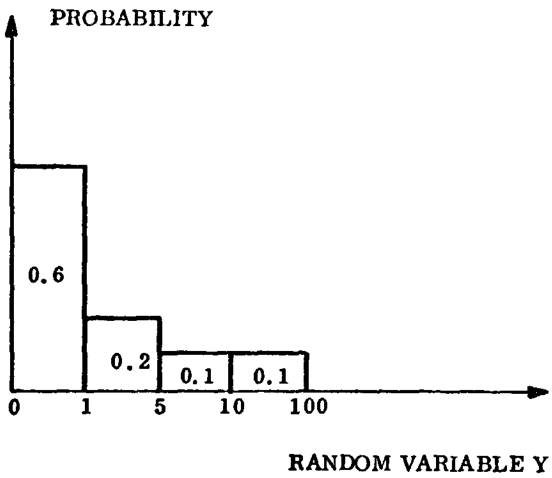
```
:=$STOP
READY.
```

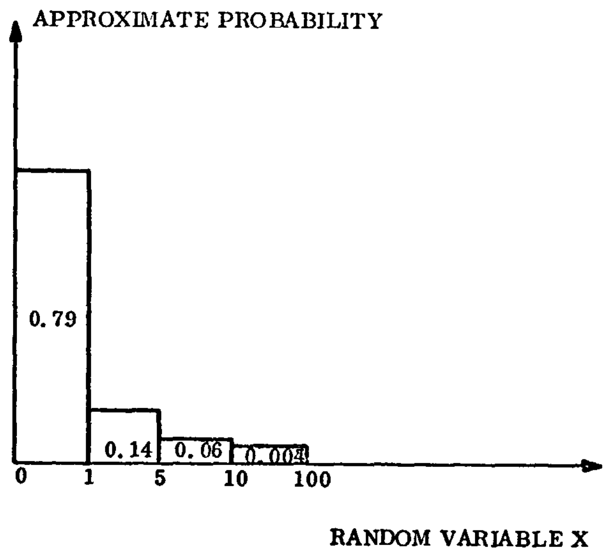

48

An example was run for y = $\{0, 3, 7, 55\}$ with associated probabilities $\{0.6, 0.2, 0.1, 0.1\}$ and $(0 \le x \le 55)$.

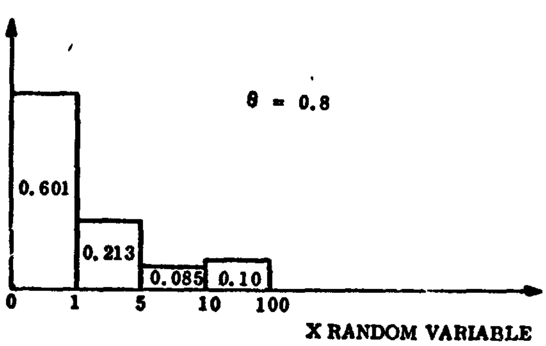
The results are shown on the previous page for a probability of survival of 0.1:

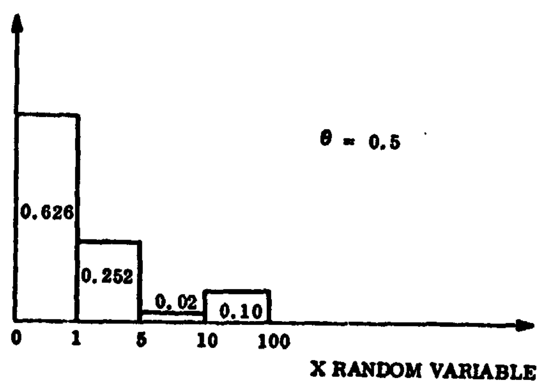
If the y values are considered to be mean values for the intervals (0, 1); (1, 5); (5, 10); (10, 100) then the relationship to interval probabilities can be established. If the resulting y probabilities are grouped into intervals then the input and output distributions can be pictured as follows:





As might be expected, the .1 probability has created a "piling up" effect about X=0. In the event the probability is large, say .8, the piling effect occurs in two different places and tends to create a bimodal distribution. For the same y distribution two runs were made for a probability of .8 and .5. The approximate results are shown below:





49

2.7 PERIAPSIS DISTRIBUTION DETERMINATION

Reference 5 contains a set of six curves which relate the function $\Delta p/\Delta V$ to λ where:

Δp = periapsis decrement (km)

ΔV = incremental velocity magnitude $(\frac{m}{s})$

λ = angle of attack (deg.)

These velocities are small (compared to orbital). This is concerned with ejecta that may enter the Martian atmosphere and thus violate the quarantine. The curves represent a varity variety of periapsis and apoapsis altitudes. The curves are cosine type and consequently were fitted with a finite Fourier series. The result of the curve fitting appears at the end of this section. The function fitted is of the form:

$$f(\partial p, \partial v) = \Delta p/_{\Delta V} \approx \frac{A_o}{z} \qquad \sum_{k=1}^{10} \left\{ A_k \cos\left(\frac{\pi k Z}{9}\right) + B_k \sin \frac{\pi k Z}{9} \right.$$

$$\text{where } Z = \frac{\lambda - 90}{10} \quad (90^\circ \leq \lambda \leq 270^\circ)$$

$f(\Delta p, \Delta V)$ has a double-valued, inverse; however, the point $\lambda = 180$ separates to curve into single valued branches. The interval concept appears to be a satisfactory technique in this case. A two-segment program called "DELP 1" and "DELP 2" was written to compute an approximation to the distribution of $\Delta p = \Delta V \times f(\Delta p, \Delta V)$, given the distributions of λ and ΔV. The program requires the number of the orbit (1 through 6) and proceeds to select the appropriate one. Otherwise, the nature of the input and output is similar to that of the other programs in the package.

A sample run follows. The curve-fitting results are as follows where the cosine terms refer to the A_k and the sine terms refer to the B_k which incidentally are zero due to the cosine nature of the curves.

```
LOAD LIMITS 07670 15361

 PERIAPSIS DISTRIBUTION PROGRAM

 ORBIT TYPE(1-6):=1

 NUMBER,ANGLE ATTACK VALUES:=7,90.,120.,150.,180.,210.,240.,270.

 ANGLE PROBABILITIES, 90. TO 270.:=.29,.2,.01,.01,.2,.29

 NUMBER,VALUES FOR PERIAPSIS DECREMENT:=11,0.,1000.,2000.,3000.
:=4000.,5000.,6000.,7000.,8000.,9000.,10000.

 NUMBER,VALUES FOR VELOCITY INCREMENT:=6,0.,200.,400.,600.,800.,1000.

 VELOCITY INCREMENT PROBABILITIES:=.2,.2,.2,.2,.2

    PERIAPSIS DECREMENT DISTRIBUTION
  1.467342E-01   1.037801E-01   1.229073E-01   1.253384E-01   1.261699E-01
  1.243205E-01   1.252317E-01   1.249776E-01   4.030254E-05
CHECK SUM =    1.000000
```

TYPE 1:

DESIRED NUMBER OF HARMONICS TO TRY:=GO

DO YOU WISH TO USE MAGNETIC TAPE, TYPE YES OR NO:=NO

TOTAL NUMBER OF DATA POINTS:=18

READ IN DATA POINTS:=.25,1.7,3.,4.2,5.4,6.4,7.1,7.7,8.,8.1,8.
:=7.7,7.1,6.4,5.4,4.2,3.,1.7

HARMONIC	COS TERMS	SINE TERMS	ERROR SS REMOVED
0	1.059444E+01	0.000000E-01	5.050901E+02
1	-3.348694E+00	3.311308E-07	1.009238E+02
2	-7.277126E-01	1.625647E-07	4.766090E+00
3	-2.944445E-01	1.039552E-07	7.802780E-01
4	-1.791110E-01	9.375554E-08	2.887267E-01
5	-1.401124E-01	8.947539E-08	1.766833E-01
6	-1.388889E-01	1.027006E-07	1.736112E-01
7	-9.452752E-02	8.567242E-08	8.041907E-02
8	-7.650987E-02	7.765082E-08	5.268384E-02
9	-9.444452E-02	0.000000E-01	2.006948E-02

TYPE 2:

FINITE FOURIER SERIES
DO YOU WISH TO USE MAGNETIC TAPE, TYPE YES OR NO:=NO

TOTAL NUMBER OF DATA POINTS:=18

READ IN DATA POINTS:=.25,1.6,2.7,3.9,5.,5.9,6.6,7.1,7.4,7.5,7.4
:=7.1,6.6,5.9,5.,3.9,2.7,1.6

HARMONIC	COS TERMS	SINE TERMS	ERROR SS REMOVED
0	9.794444E+00	0.000000E-01	4.316901E+02
1	-3.100466E+00	3.343161E-07	8.651601E+01
2	-6.714439E-01	1.440733E-07	4.057533E+00
3	-2.611111E-01	9.481432E-08	6.136113E-01
4	-1.491711E-01	7.948839E-08	2.002681E-01
5	-1.161407E-01	7.683878E-08	1.213980E-01
6	-1.055556E-01	8.155107E-08	1.002778E-01
7	-1.000602E-01	8.966337E-08	9.010841E-02
8	-9.605177E-02	9.180776E-08	8.303348E-02
9	-9.444452E-02	0.000000E-01	2.006948E-02

TYPE 3:

```
   DESIRED NUMBER OF HARMONICS TO TRY:=GO

DO YOU WISH TO USE MAGNETIC TAPE, TYPE YES OR NO:=NO

TOTAL NUMBER OF DATA POINTS:=18

READ IN DATA POINTS:=.25,1.4,2.5,3.7,4.7,5.6,6.2,6.6,7.,7.1,7.,
:=6.6,6.2,5.6,4.7,3.7,2.5,1.4
```

HARMONIC	COS TERMS	SINE TERMS	ERROR SS REMOVED
0	9.194444E+00	0.000000E-01	3.804201E+02
1	-2.940966E+00	3.112403E-07	7.784351E+01
2	-6.530806E-01	1.454036E-07	3.838628E+00
3	-2.722222E-01	9.947438E-08	6.669446E-01
4	-1.060668E-01	5.931057E-08	1.012514E-01
5	-1.198379E-01	7.928457E-08	1.292501E-01
6	-7.222224E-02	6.237313E-08	4.694446E-02
7	-5.586342E-02	5.906608E-08	2.808649E-02
8	-9.085276E-02	8.954265E-08	7.428801E-02
9	-7.222228E-02	0.000000E-01	1.173613E-02

TYPE 4:

```
   DESIRED NUMBER OF HARMONICS TO TRY:=GO

DO YOU WISH TO USE MAGNETIC TAPE, TYPE YES OR NO:=NO

TOTAL NUMBER OF DATA POINTS:=18

READ IN DATA POINTS:=.25,2.4,4.4,6.3,8.,9.5,10.7,11.4,11.8,12.
:=11.8,11.4,10.7,9.5,8.,6.3,4.4,2.4
```

HARMONIC	COS TERMS	SINE TERMS	ERROR SS REMOVED
0	1.569444E+01	0.000000E-01	1.108420E+03
1	-5.006866E+00	5.580665E-07	2.256183E+02
2	-1.155144E+00	2.570172E-07	1.200922E+01
3	-4.277778E-01	1.519897E-07	1.646945E+00
4	-3.001248E-01	1.479707E-07	8.106742E-01
5	-2.253168E-01	1.341112E-07	4.569089E-01
6	-1.388889E-01	1.043137E-07	1.736112E-01
7	-1.511513E-01	1.250498E-07	2.056203E-01
8	-1.280644E-01	1.155206E-07	1.476045E-01
9	-1.277779E-01	0.000000E-01	3.673617E-02

TYPE 5:

```
   FINITE FOURIER SERIES
DO YOU WISH TO USE MAGNETIC TAPE, TYPE YES OR NO:=NO

TOTAL NUMBER OF DATA POINTS:=18

READ IN DATA POINTS:=.25,2.,4.2,6.1,7.7,9.1,10.3,11.,11.4,11.5,11.4
:=11.,10.3,9.1,7.7,6.1,4.2,2.
```

HARMONIC	COS TERMS	SINE TERMS	ERROR SS REMOVED
0	1.503889E+01	0.000000E-01	1.017757E+03
1	-4.891183E+00	5.526868E-07	2.153130E+02
2	-1.157197E+00	2.532923E-07	1.205195E+01
3	-4.500000E-01	1.591590E-07	1.822500E+00
4	-3.137546E-01	1.549004E-07	8.859775E-01
5	-1.722799E-01	1.057807E-07	2.671232E-01
6	-9.444444E-02	7.599485E-08	8.027777E-02
7	-8.653781E-02	7.955296E-08	6.739914E-02
8	-7.904804E-02	7.977436E-08	5.623733E-02
9	-5.000005E-02	0.000000E-01	5.625011E-03

TYPE 6:

```
   DESIRED NUMBER OF HARMONICS TO TRY:=GO

DO YOU WISH TO USE MAGNETIC TAPE, TYPE YES OR NO:=NO

TOTAL NUMBER OF DATA POINTS:=18

READ IN DATA POINTS:=.25,1.3,3.9,5.7,7.6,8.8,9.9,10.7,11.1,11.3
:=11.1,10.7,9.9,8.8,7.6,5.7,3.9,1.3
```

HARMONIC	COS TERMS	SINE TERMS	ERROR SS REMOVED
0	1.439444E+01	0.000000E-01	9.324001E+02
1	-4.944771E+00	5.681179E-07	2.200569E+02
2	-1.200388E+00	2.695222E-07	1.296838E+01
3	-4.944444E-01	1.717053E-07	2.200278E+00
4	-2.284768E-01	1.161753E-07	4.698150E-01
5	-1.005758E-01	6.827849E-08	9.103944E-02
6	-7.222217E-02	6.183543E-08	4.694438E-02
7	-3.798681E-02	4.629503E-08	1.298698E-02
8	7.886461E-02	-3.737432E-08	5.597663E-02
9	1.055556E-01	0.000000E-01	2.506946E-02

2.8 MARS ORBIT TIME AND M/C_dA DISTRIBUTION

Reference 5 contains two curves relating the quantity $T/h_a\ (M/C_dA)$ to periapsis altitude (all distances in km). The terms are: T, orbital lifetime (years); b_a, apoapsis altitude for the six types of orbits in the periapsis section; and M/C_dA, drag parameter.

The independent variable $P = P_a - \Delta P$ where P_a is the periapsis altitude for the type of orbit under consideration and ΔP is the random variable whose distribution has been determined in the periapsis section. Each curve represents extremes in the VM-3 atmosphere variation. The curves were fit by fitting orthogonal polynomials to $\ln\ (T/h_a\ (M/C_dA))$ VS. P. That is, each curve was approximated by

$$f\ (T,\ h_a,\ M/C_dA) = T/h_a\ (M/C_dA) = e^{\left(\overset{4}{\Sigma}\ B_j\ \Phi_j\ (P)\right)}$$

where Φ_j (P) are orthogonal polynomials of degree j. The shape of the curves indicates that transforming to $\sqrt{P}$ would help the approximation, and so this will be attempted at a later date. The present results do look satisfactory, however. The curve-fitting results are shown at the end of this section.

Since the curves are monotonic, the interval technique should be effective. The programs titled "TIMF1" and "M/C_dA1" were written to provide the distributions of T and M/C_dA, respectively. Obviously, the former approximates the density of $T = h_a \times (M/C_dA) \times f\ (T, h_a, M/C_dA)$, while the latter approximates $T = h_a \times \dfrac{1}{f\ (T, h_a, M/C_dA)}$.

In "TIME" the orbit type (1 through 6) and atmosphere type (1 or 2, where 1 is the upper curve and 2 is the lower curve) are entered initially. The associated h_a and P_a are printed out as a check. Following this the grid and probabilities for P and (M/C_dA) are entered, including the output (T) grid in the usual way. Note that T, M/C_dA, and P are all considered as random variables in each case. A typical run follows.

```
READY.

$RUN
WAIT.

LOAD LIMITS 07440 15311

 TIME IN MARS ORBIT PROGRAM



  ORBIT TYPE(1-6),ATMOS TYPE(1 OR 2):=1,1

   APOAPSIS(KM)     PERIAPSIS(KM)
  1.000000E+04  1.000000E+03
 NUMBER, PERIAPSIS VALUES:=3,200.,600.,1000.

 PERIAPSIS PROBABILITIES:=.7,.3

 NUMBER, VALUES FOR TIME IN ORBIT:=6,0.,5.,10.,100.,1000.,2000.

 NUMBER, VALUES FOR DRAG PARAMETER:=3,1.E-5,1.E-4,1.E-3

 DRAG PARAMETER PROBABILITIES:=.4,.6

 TIME IN ORBIT DISTRIBUTION
  2.218431E-01  1.235655E-01  4.544181E-01  1.515289E-01  4.864454E-02
CHECK SUM =  1.000000

READY.

$RUN
WAIT.

LOAD LIMITS 07440 15311

 M/CDA DISTRIBUTION PROGRAM



  ORBIT TYPE(1-6),ATMOS TYPE(1 OR 2):=1,1

   APOAPSIS(KM)     PERIAPSIS(KM)
  1.000000E+04  1.000000E+03
 NUMBER, PERIAPSIS VALUES:=3,200.,600.,1000.

 PERIAPSIS PROBABILITIES:=.7,.3

 NUMBER,VALUES FOR M/CDA:=20,0.,10.,100.,1000.,1.E4,1.E5,1.E6
:=1.E7,1.E8,1.E9,1.E10,1.E20,1.E30,1.E40,1.E50,1.E60,1.E70
:=1.E71,1.E72,1.E72

 NUMBER,VALUES FOR TIME IN ORBIT:=6,0.,1.,2.,3.,5.,10.

 TIME IN ORBIT PROBABILITIES:=.2,.2,.2,.2,.2

 M/CDA DISTRIBUTION
  2.336721E-01  2.852110E-01  3.811170E-01
CHECK SUM =  1.000000
```

The curve-fitting results for the pair of curves is shown below:

```
ORTHOGONAL POLYNOMIAL CURVE FITTING
DO YOU WISH TO USE MAGNETIC TAPE, TYPE YES OR NO:=NO

TYPE NUMBER OF POINTS,MAXIMUM DEGREE:=9,8

TYPE IN DEPENDENT DATA:=-13.81551,-5.298317,-1.609438,.6931472
:=2.079442,2.995732,3.912023,4.382027,5.010635

TYPE IN INDEPENDENT DATA:=200.,300.,400.,500.,600.,700.,800.,900.,1000.

TYPE IN WEIGHTS:=1.,1.,1.,1.,1.,1.,1.,1.,1.

DEPENDENT DATA MEAN
 -1.833621E-01
  DEGREE        ALPHA           BETA           COEFF           SSR
       1    .300000E+02  0.000000E-01  1.961519E-02  2.308533E+02
       2   6.000000E+02  6.666667E+04 -4.065703E-05  5.091223E+01
       3   6.000000E+02  5.133333E+04  8.695401E-08  1.077896E+01
       4   6.000000E+02  4.628571E+04 -1.711093E-10  1.722574E+00
       5   6.000000E+02  4.126984E+04  3.727533E-13  2.890057E-01
       6   6.000000E+02  3.535354E+04 -6.375616E-16  2.394565E-02
       7   6.000000E+02  2.832168E+04  5.762255E-18  3.932047E-02
       8   6.000000E+02  2.010256E+04  4.914316E-21  3.053624E-04


DESIRED NUMBER OF POLYNOMIALS TO TRY:=4

WHICH ONES:=1,2,3,4

INPUT:=200.

 PREDICTED VALUE -1.369558E+01
INPUT:=300.

 PREDICTED VALUE -5.670175E+00
INPUT:=400.

 PREDICTED VALUE -1.343066E+00
INPUT:=500.

 PREDICTED VALUE  8.341239E-01
INPUT:=600.

 PREDICTED VALUE  1.999112E+00
INPUT:=700.

 PREDICTED VALUE  2.878955E+00
INPUT:=800.

 PREDICTED VALUE  3.790044E+00
INPUT:=900.

 PREDICTED VALUE  4.638110E+00
INPUT:=1000.

 PREDICTED VALUE  4.918221E+00
INPUT:=1.E75

DESIRED NUMBER OF POLYNOMIALS TO TRY:=0
```


57

```
DO YOU WISH TO USE MAGNETIC TAPE, TYPE YES OR NO:=NO

TYPE NUMBER OF POINTS,MAXIMUM DEGREE:=9,8

TYPE IN DEPENDENT DATA:=-13.81551,-8.111728,-5.298317,-3.912023
:=-2.65926,-1.609438,-.5108256,0.,.6931472

TYPE IN INDEPENDENT DATA:=200.,300.,400.,500.,600.,700.,800.,900.,1000.

TYPE IN WEIGHTS:=1.,1.,1.,1.,1.,1.,1.,1.,1.

DEPENDENT DATA MEAN
 -3.913773E+00
  DEGREE         ALPHA           BETA           COEFF              SSR
        1   6.000000E+02   0.000000E-01   1.570790E-02   1.480428E+02
        2   6.000000E+02   6.666667E+04  -2.496592E-05   1.919755E+01
        3   6.000000E+02   5.133333E+04   5.334886E-08   4.057402E+00
        4   6.000000E+02   4.628571E+04  -1.370223E-10   1.104620E+00
        5   6.000000E+02   4.126984E+04   2.781683E-13   1.609454E-01
        6   6.000000E+02   3.535354E+04   4.881966E-16   1.404015E-02
        7   6.000000E+02   2.832168E+04   6.220064E-19   4.581667E-04
        8   6.000000E+02   2.010256E+04   3.018112E-20   1.150623E-02



DESIRED NUMBER OF POLYNOMIALS TO TRY:=4

WHICH ONES:=1,2,3,4

INPUT:=200.

 PREDICTED VALUE -1.375220E+01
INPUT:=300.

 PREDICTED VALUE -8.267269E+00
INPUT:=400.

 PREDICTED VALUE -5.298967E+00
INPUT:=500.

 PREDICTED VALUE -3.705065E+00
INPUT:=600.

 PREDICTED VALUE -2.672190E+00
INPUT:=700.

 PREDICTED VALUE -1.715821E+00
INPUT:=800.

 PREDICTED VALUE -6.802930E-01
INPUT:=900.

 PREDICTED VALUE  2.612080E-01
INPUT:=1000.

 PREDICTED VALUE  6.066411E-01
INPUT:=$STOP
```

2.9 HELIOCENTRIC TRANSFER CASE

Reference 5 contains information on the various effects on the Mars impact miss distance during the transfer orbit phase. This report contains four curves which relate the four following quantities to time in days to intercept.

a. In-plane miss distance due to tangential component of ejection velocity.

$$T_1 / \Delta V_T = f_1(t) \quad (\text{km}/\frac{m}{s})$$

b. In-plane miss distance due to normal ejection velocity component.

$$T_2 / \Delta V_N = f_2(t) \quad (\text{km}/\frac{m}{s})$$

The results here were multiplied by 10^3 to obtain the necessary units.

c. Radiation pressure perturbation to transfer trajectories.

$$T_3\left(\frac{M}{C_d A}\right) = f_3(t) \quad \text{km}/\left(\frac{\text{slugs}}{\text{ft}^2}\right)$$

Two curves (Type I, 1973 and Type II, 1975)

d. Out-of-plane component of particle miss distance caused by out-of-plane component of ejection velocity.

$$R / \Delta V_R = f_4(t) \quad (\text{km}/\frac{m}{s})$$

Two curves (Type I, 1973 and Type II, 1975). The results here were multiplied by 10^3 to obtain the necessary units.

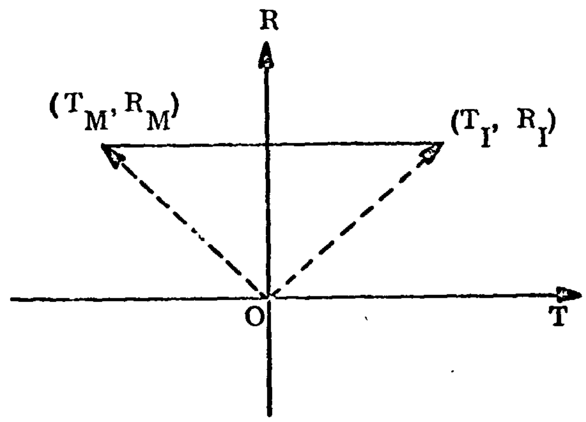
Thus the four random variables (ΔV_T, ΔV_N, M/C_dA, and DV_R) contribute to the Mars quarantine area miss distance, including a bias deliberately programmed into the guidance system.

Denoting R and T as the out-of-plane and in-plane components in the impact plane, the impact point (T_I, R_I) components are given by $T_I = T_1 + T_2 + T_3$

$R_I = R.$

The miss distance from Mars (T_m, R_m) is given by

$$d = \sqrt{(T_I - T_m)^2 + (R_I - R_m)^2} \text{ (kilometers)}$$



It is the job of the two segments "HELIO 1" and HELIO 2" to approximate the distribution of d, given the above-described random variables. The four (actually six) curves were fit in the following ways:

a. Orthogonal polynomials were used to fit $\ln f_1(t)$ VS $\sqrt{t}$.

b. Orthogonal polynomials were used to fit $10^3 \times f_2(t)$ VS. t

c. Orthogonal polynomials were used to fit $\ln f_3(t)$ VS. $\sqrt{t}$ (for both curves).

d. Orthogonal polynomials were used to fit $10^3 \times f_4(t)$ VS t (for both curves).

The results of the fits follow for all six curves. The procedure is fairly simple. The user inputs the four grids and associated probabilities and the output (d) grid. He also must provide: days to impact, orbit type (1 or 2-needed for $f_3(t)$, $f_4(t)$, and the Mars bias coordinates in the impact plane (in km.) The program then samples the appropriate curves and calculates intervals for:

a. $T_1 = \Delta V_T \times f_1(t)$

b. $T_2 = \Delta V_N \times 10^3 \times f_2(t)$

c. $T_3 = f_3(t) / M/C_d A$

d. $R = \Delta V_R \times 10^3 \times f_4(t)$

Thus for n_1, n_2, n_3, n_4 intervals, the program form $n_1 \times n_2 \times n_3 \times n_4$ intervals and computes $d\sqrt{(T_1 + T_2 + T_3 - T_m)^2 + (R-R_m)^2}$ for each interval. The probability associated with each interval is the product of the probability for each variable for the particular intervals concerned.

61

ORTHOGONAL POLYNOMIAL CURVE FITTING
DO YOU WISH TO USE MAGNETIC TAPE, TYPE YES OR NO:=NO $\ell n(\frac{T_1}{\Delta V_T}) \sim \frac{km}{m/s}$

TYPE NUMBER OF POINTS,MAXIMUM DEGREE:=13,12

TYPE IN DEPENDENT DATA:=10.81978,10.73204,10.63586,10.18867
:=9.21034,8.922658,6.907755,5.857933,4.434382,3.931826
:=2.944439,2.302585,0.

TYPE IN INDEPENDENT DATA:=18.70829,17.32051,15.81139
:=14.14214,12.24745,10.,7.071068,5.,3.535534,2.5,1.767767
:=1.251998,0.

TYPE IN WEIGHTS:=1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,10.

DEPENDENT DATA MEAN
3.949467E+00

DEGREE	ALPHA	BETA	COEFF	SSR
1	4.970734E+00	0.000000E-01	6.436589E-01	3.748817E+02
2	1.128421E+01	4.113012E+01	-3.821528E-02	2.392741E+01
3	9.779519E+00	1.810669E+01	1.998154E-03	1.639684E+00
4	9.675320E+00	2.506573E+01	-2.700023E-04	6.134089E-01
5	9.866702E+00	2.048864E+01	2.481163E-05	1.108942E-01
6	9.626673E+00	2.140835E+01	-5.069287E-06	8.658075E-02
7	9.838191E+00	1.870377E+01	4.697225E-07	1.422646E-02
8	9.183295E+00	1.913756E+01	2.604144E-07	7.278603E-02
9	9.748145E+00	1.664577E+01	2.248741E-08	8.901758E-03
10	8.043692E+00	1.640132E+01	-1.189293E-08	3.132336E-02
11	1.002555E+01	1.378526E+01	-6.717493E-09	[illegible].244618E-01
12	4.868961E+00	1.136606E+01	-5.608993E-09	6.941599E-02

ORTHOGONAL POLYNOMIAL CURVE FITTING
DO YOU WISH TO USE MAGNETIC TAPE, TYPE YES OR NO:=NO $10^3 x \frac{T_2}{\Delta V_N} \sim \frac{km}{m/s}$

TYPE NUMBER OF POINTS,MAXIMUM DEGREE:=8,7

TYPE IN DEPENDENT DATA:=-4.6,-.8,3.,6.6,8.2,7.3,4.2,0.

TYPE IN INDEPENDENT DATA:=350.,300.,250.,200.,150.,100.,50.,0.

TYPE IN WEIGHTS:=1.,1.,1.,1.,1.,1.,1.,1.

DEPENDENT DATA MEAN
2.987500E+00

DEGREE	ALPHA	BETA	COEFF	SSR
1	1.750000E+02	0.000000E-01	-1.707143E-02	3.060054E+01
2	1.750000E+02	1.312500E+04	-3.183333E-04	1.064029E+02
3	1.750000E+02	1.000000E+04	5.595960E-07	2.906402E+00
4	1.750000E+02	8.839286E+03	3.924243E-09	1.088977E+00
5	1.750000E+02	7.619048E+03	-1.466667E-11	9.363094E-02
6	1.750000E+02	6.155303E+03	-1.288889E-13	3.185605E-02
7	1.750000E+02	4.405594E+03	-9.904750E-16	4.431807E-03

```
ORTHOGONAL POLYNOMIAL CURVE FITTING
DO YOU WISH TO USE MAGNETIC TAPE, TYPE YES OR NO:=NO
```

$\ln(T_3(\frac{M}{C_d A}))$

Type I (1973)

```
TYPE NUMBER OF POINTS,MAXIMUM DEGREE:=7,6

TYPE IN DEPENDENT DATA:=8.373322,6.907755,6.39693
:=4.447346,2.928524,2.079442,0.

TYPE IN INDEPENDENT DATA:=14.14214,12.24745,10.,7.071068
:=5.,3.162278,0.

TYPE IN WEIGHTS:=1.,1.,1.,1.,1.,1.,100.

DEPENDENT DATA MEAN
  2.937106E-01
```

DEGREE	ALPHA	BETA	COEFF	SSP
1	4.870038E-01	0.000000E-01	5.972080E-01	1.818451E+02
2	1.111457E+01	4.809994E+00	-5.188701E-03	1.184448E-01
3	8.302233E+00	8.628762E+00	-2.828954E-04	3.424910E-03
4	8.578036E+00	9.727467E+00	3.301192E-04	4.112717E-02
5	7.382239E+00	8.833383E+00	3.147279E-04	2.759230E-01
6	8.432462E+00	7.368716E+00	9.506986E-06	1.278175E-02

```
ORTHOGONAL POLYNOMIAL CURVE FITTING
DO YOU WISH TO USE MAGNETIC TAPE, TYPE YES OR NO:=NO
```

$\ln(T_3(\frac{M}{C_d A}))$

Type II (1975)

```
TYPE NUMBER OF POINTS,MAXIMUM DEGREE:=10.,9

TYPE IN DEPENDENT DATA:=8.007947,7.839334,7.192934,6.320476
:=6.51947,5.011141,4.312141,2.995732,2.012895
:=0.

TYPE IN INDEPENDENT DATA:=18.70829,17.32051,15.81139,14.14214
:=12.24745,10.,7.071068,5.,3.162278,0.

TYPE IN WEIGHTS:=1.,1.,1.,1.,1.,1.,1.,1.,1.,100.

DEPENDENT DATA MEAN
  4.751153E-01
```

DEGREE	ALPHA	BETA	COEFF	SSP
1	9.492030E-01	0.000000E-01	4.797478E-01	3.076736E+02
2	1.439389E+01	1.226415E+01	-1.435841E-02	3.533358E+00
3	1.094751E+01	1.282067E+01	1.445612E-04	[illegible]
4	1.263706E+01	1.959507E+01	1.624449E-04	[illegible]
5	1.075831E+01	1.674908E+01	[illegible]	[illegible]
6	1.032730E+01	1.746049E+01	-6.352601E-06	[illegible]
7	1.095632E+01	1.544923E+01	-5.401351E-07	5.610216E-03
8	1.046540E+01	1.269522E+01	-1.775554E-07	[illegible]
9	1.039951E+01	1.322034E+01	[illegible]	[illegible]

63

ORTHOGONAL POLYNOMIAL CURVE FITTING
DO YOU WISH TO USE MAGNETIC TAPE, TYPE YES OR NO:=NO

TYPE NUMBER OF POINTS,MAXIMUM DEGREE:=5,4 $10^3 \times \frac{R}{\Delta}$

TYPE IN DEPENDENT DATA:=3.3,7.4,6.8,3.4,0.

Type I (1978)

TYPE IN INDEPENDENT DATA:=200.,150.,100.,50.,0.

TYPE IN WEIGHTS:=1.,1.,1.,1.,50.

DEPENDENT DATA MEAN
3.870370E-01

DEGREE	ALPHA	BETA	COEFF	SSR
1	9.259259E+00	0.000000E-01	3.448158E-02	8.366891E+01
2	1.585039E+02	1.303155E+03	-4.781210E-04	2.921414E+01
3	1.227774E+02	1.816049E+03	-3.141853E-06	2.466142E+00
4	1.080396E+02	1.954920E+03	6.000009E-09	1.173576E-02

DO YOU WISH TO USE MAGNETIC TAPE, TYPE YES OR NO:=NO

$10^3 \times \frac{R}{\Delta}$

TYPE NUMBER OF POINTS,MAXIMUM DEGREE:=8,7

Type II (1975)

TYPE IN DEPENDENT DATA:=-3.,3.3,6.9,7.7,6.6,4.9,2.7,0.

TYPE IN INDEPENDENT DATA:=350.,300.,250.,200.,150.,100.,50.,0.

TYPE IN WEIGHTS:=1.,1.,1.,1.,1.,1.,1.,1.

DEPENDENT DATA MEAN
3.637500E+00

DEGREE	ALPHA	BETA	COEFF	SSR
1	1.750000E+02	0.000000E-01	-2.595238E-03	7.079024E-[illegible]
2	1.750000E+02	1.312500E+04	-2.902381E-04	8.845006E+01
3	1.750000E+02	1.000000E+04	-8.343434E-07	6.463947E+00
4	1.750000E+02	8.839286E+03	-8.636362E-10	5.274349E-02
5	1.750000E+02	7.619048E+03	1.610256E-11	1.128617E-01
6	1.750000E+02	6.155303E+03	7.555557E-14	1.094697E-02
7	1.750000E+02	4.405594E+03	-9.396823E-16	3.988926E-03

```
$LOAD HEL101,HEL102

LOAD LIMITS 11643 13611

 HELIOCENTRIC ORBIT PROBABILITY PROGRAM



 DAYS TO IMPACT, ORBIT TYPE,T,R MARS :=3,1,0.,0.

 NUMBER, TANGENTIAL VELOCITY VALUES:=3,0.,5.,10.

 TANGENTIAL PROBABILITIES:=.8,.2

 NUMBER, NORMAL VELOCITY VALUES:=3,0.,5.,10.

 NORMAL PROBABILITIES:=.7,.3

 NUMBER, M/CDA VALUES:=4,1.E-1,1.E-2,1.E-3,1.E-4

 M/CDA PROBABILITIES:=.7,.2,.1

 NUMBER, O-O-P VELOCITY VALUES:=3,0.,5.,10.

 O-O-L PROBABILITIES:=.6,.4

 NUMBER, MISS DISTANCE VALUES:=10,0.,100.,1000.,1.E4,1.E5,1.E6
:=1.E7,1.E8,1.E9,1.E10


  T1/DVT  T2/DVN  T3(M/CDA)  R/DVR   C-F
  1.593031E+01  2.415908E+02  2.528376E+00  1.690686E+02
 IN-PLANE MISS DISTANCE PROBABILITIES
  1.014702E-02  1.921751E-01  7.269995E-01  7.067839E-02
CHECK SUM =  1.000000




 DAYS TO IMPACT, ORBIT TYPE,T,R MARS :=$STOP
READY.
```

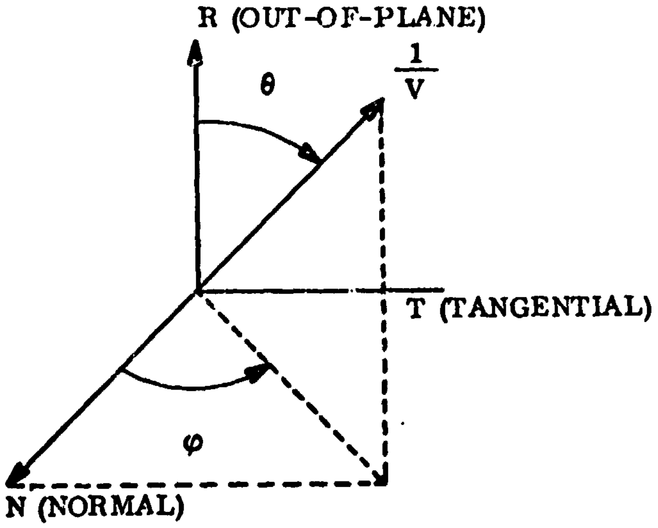
ADDENDUM TO HELIO1, HELIO2

The segments HELIO1, HELIO2 perform as described in PIR 5540-41.

A new program, to be loaded as HELIO3, HELIO2, was written to allow the user to input ejection velocity magnitude and two angles along with drag parameter. These four quantities are considered to be stochastically independent.

Define a local axis system as N, T, R where N is the local normal of the velocity vector, T is the local tangent of the velocity vector, and R is the out of (transfer) plane component.

It is along thse three axes that the "old" program HELIO1, HELIO2 considered as its basic input.



Define

- V, magnitude of velocity $(\frac{m}{s})$
- θ, polar angle (deg)
- φ, N-T plane angle (deg)
- β, drag term $(\frac{slugs}{ft^2})$ (M/C_dA)

The miss distance from the center of Mars is calculated as

$$d = \sqrt{(T-T_M)^2 + (R-R_M)^2} \text{ (km)}$$

where
$$T = T_1 + T_2 + T_3$$
$$= C_1 V \sin\theta \sin\varphi + C_2 V \sin\theta \cos\varphi + C_3/\beta$$
$$R = C_4 V \cos\theta$$

and

$$C_1 = T_1/\Delta V_T$$

$$C_2 = T_2/\Delta V_N$$

$$C_3 = T_3\beta$$

$$C_4 = R/\Delta V_R$$

which are curve-fitted results of curves supplied by D. A. Korenstein given as a function of "TIME" (days to intercept).

T_M and R_M are the coordinates of the center of Mars in the impact plane. (T, transfer plane direction and R, out of transfer plane direction.)

Note that the T_1, T_2, T_3 components of velocity are not independent and involve a complete different numerical process as performed in HELIO1, HELIO2.

The numerical technique involves calculating all 16 possible d's for each combination of random variable values (V, θ, φ, β), choosing the minimum and maximum, and loading the associated probabilities by the technique described in the writeup of "PLOAD."

Thus, for n1 values of V, n2 values of θ, n3 values of φ, and n4 values of β the program must calculate d a total of 2^4x(n1-1)x(n2-1)x(n3-1)x(n4-1) times. Also (n1-1)x(n2-1)x(n3-1)x(n4-1) intervals are loaded onto the "d" grid as in the usual manner.

67

SAMPLE PROBLEM

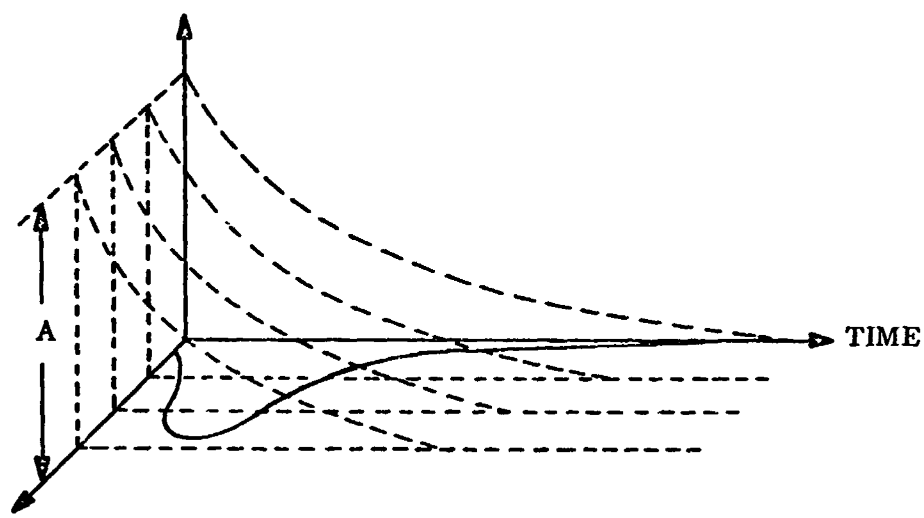
A time of 5 days to impact was chosen for a TYPE1 orbit and Mars coordinates of (-10., -10).

CHECK CASE
12/15/66

```
 $STOP
READY.

$LOAD HELIO3,HELIO2

LOAD LIMITS 11521 13611
 HELIOCENTRIC ORBIT PROBABILITY PROGRAM


 DAYS TO IMPACT, ORBIT TYPE,T,R MARS :=5,1,-10.,-10.
 NUMBER, VELOCITY MAG VALUES(M/S):=3,0.,10.,20.
 VELOCITY MAG PROBABILITIES:=.8,.2
 NUMBER,POLAR ANGLE VALUES(DEG):=2,5.,15.
 POLAR ANGLE PROBABILITIES:=1.
 NUMBER, N-T PLANE ANGLE VALUES(DEG):=2,5.,15.
 N-T PLANE ANGLE PROBABILITIES:=1.
 NUMBER, M/CDA VALUES(SLUGS/FT*FT):=2,1.E-3,1.E-2
 M/CDA PROBABILITIES:=1.
 NUMBER, MISS DISTANCE VALUES (KM):=10,0.,1000.,2000.,3000.,4000.
:=5000.,10000.,1.E5,1.E6,1.E7

    T1/DVT       T2/DVN      T3(M/CDA)      R/DVR                             C-F
  2.991216E+01  4.263029E+02  ?.496503E+00  2.860507E+02
 ***** MISS DISTANCE PROBABILITIES *****
  1.019843E-01  1.592980E-01  1.612878E-01  1.992267E-01  1.992267E-01
  1.789764E-01
CHECK SUM =  1.000000
```

2.10 ENTRY SURVIVAL PROBABILITY

Reference (1) contains a description of the parameterization of the estimated effect of heat-time on viable organisms entering the Martian atmosphere.



The above diagram illustrates the process. If we have "A" organisms to start with, the die off will proceed (negative exponential) as is shown in the dotted curves at constant temperatures. The history of an entering particle may suffer a heat-time curve in the heat-time plane as illustrated.

Reference 1 develops a justification for computing a particular index of the particle history called the lethality integral (I_L). Once I_L is computed, the probability of an organism surviving (to some indicated percentage) is

$$\frac{1}{\left(A^{I_L}\right)}$$

Thus if I_L is a random variable itself, the probability of survival can be estimated by

$$\text{pr (survival)} = \sum_j \text{pr}\left(I_L = I_{L_j}\right)\left(\frac{1}{\left(A^{I_{L_j}}\right)}\right)$$

69

In Reference 1, I_L is considered to be a function of several parameters. In particular, four seem to be the most important:

ϵ = emissivity

ν = initial entry velocity

γ = initial entry angle

Z = drag parameter

M. A. Martin has demonstrated that the relationship:

$$\rho\, n_{1\epsilon}\, I_L = 3.34036 - 5.34036 \left(\frac{\xi}{\xi_2}\right)$$

where

$$\xi_2 = k_1 + k_2 \bar{Z} + k_3 \bar{\gamma} + k_4 \bar{\nu} + k_5 \bar{Z}\,\bar{\gamma} + k_6 \bar{\gamma}\,\bar{\nu} + k_7 \bar{\nu}\,\bar{Z} + k_8 \bar{Z}\,\bar{\gamma}\,\bar{\nu}$$

where

$$\bar{Z} = Z \times 10^4$$

$$\gamma = \left(\frac{90-\gamma_2}{100}\right)$$

$$\bar{\nu} = \left(\frac{\nu}{10^4}\right)^3$$

is a satisfactory form in his preliminary studies from available data. Appropriately, the program "LID" was written to compute the probability distribution of I_L. The input is by the same method of providing grid intervals and probabilities used in other programs.

```
NUMBER, BALLISTIC COEFFICIENTS
:=3,4.E-5,22.E-5,4.E-4

BALLISTIC PROBABILITIES
:=.8,.2

NUMBER, INITIAL ENTRY ANGLES
:=3,5.,10.,25.

ENTRY ANGLES PROBABILITIES
:=.5,.5

NUMBER, INITIAL PARTICLE VELOCITY
:=3,12000.,13000.,14000.

PARTICLE VELOCITY PROBABILITIES
:=.5,.5

NUMBER, EMISSIVITIES
:=3,.2,.3,.4

EMISSIVITY PROBABILITIES
:=.7,.3

NUMBER, LETHALITY INTEGRAL
:=10,1.E-4,1.E-3,1.E-2,1.E-1,1.,2.,5.,10.,1.E5←←←←100.,1.E5



***LETHALITY PROBABILITIES

  2.934343E-04  2.648213E-03  2.649075E-02  1.338197E-01  4.328347E-02
  1.298504E-01  1.236249E-01  3.835066E-01  1.564824E-01


07704 EOT
SUM =   1.000000E+00


NUMBER, BALLISTIC COEFFICIENTS
:=$STOP
READY.
```

2.11 M/CDA SURVIVAL PROBABILITIES

According to Reference 7, one method of determining the distribution of M/CDA that enters the atmosphere is to generate a distribution of upper limits on M/CDA entering the atmosphere (M/CDA program).

This distribution is then merged with the given M/CDA distribution to determine the surviving distribution of M/CDA that enters the atmosphere.

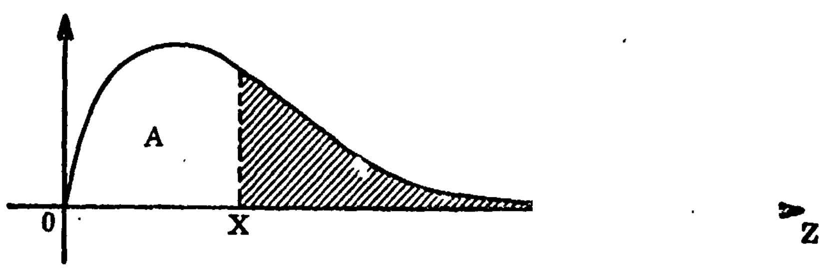
Define

Z = original a priori random variable

Z_{ul} = upper limit random variable

Z_a = resulting "a posteriori" random variable

When a value for $Z_{ul} = \alpha$ is given, the conditional distribution for Z



can be found by dividing the modified area A. That is the density of Z is modified to form

$$f\left(Z \mid Z_{ul} = \alpha\right) = \frac{f(Z)}{A} .$$

The final distribution becomes:

$$f(Z_A) = \sum_{Z_{ul}} g(Z_{ul})\ f(Z \mid Z_{ul})$$

73

The numerical procedure consists of reading the density for Z and Z_{ul} in interval probability form and generating the distributing of Z_A in the same form by calculating the summation described above. A program now exists on the DSCS to perform this calculation.

The program is known as "LIMIT".

To use the program, provide first the upper limit points and related probabilities; next the M/C_DA points and probabilities and finally the desired put points for the resulting marginal distribution.

```
PROGRAM TO COMBINE UPPER LIMIT AND M/CDA VARIABLES.

NUMBER,UPPER LIMIT POINTS:=5,0.,1.,2.,3.,4.

UPPER LIMIT PROBABILITIES:=.25,.25,.25,.25

NUMBER,M/CDA POINTS:=5,0.,1.,2.,3.,4.

M/CDA PROBABILITIES:=.6,.2,.1,.1

NUMBER,OUTPUT POINTS:=10,0.,1.,2.,3.,4.,5.,10.,20.,30.,50.

   FIRST           LAST          PROBABILITY
 0.000000E-01   1.000000E+00   7.541667E-01
 1.000000E+00   2.000000E+00   1.680556E-01
 2.000000E+00   3.000000E+00   5.277778E-02
 3.000000E+00   4.000000E+00   2.500000E-02
 4.000000E+00   5.000000E+00   0.000000E-01
 5.000000E+00   1.000000E+01   0.000000E-01
 1.000000E+01   2.000000E+01   0.000000E-01
 2.000000E+01   3.000000E+01   0.000000E-01
 3.000000E+01   5.000000E+01   0.000000E-01

NUMBER,UPPER LIMIT POINTS:=5,0.,1.,2.,3.,4.

UPPER LIMIT PROBABILITIES:=.25,.25,.25,.25

NUMBER,M/CDA POINTS:=10,0.,1.,2.,3.,4.,5.,6.,7.,8.,9.

M/CDA PROBABILITIES:=.6,.2,.1,.1,0.,0.,0.,0.,0.

NUMBER,OUTPUT POINTS:=10,0.,1.,2.,3.,4.,5.,10.,20.,30.,50.

   FIRST           LAST          PROBABILITY
 0.000000E-01   1.000000E+00   7.541667E-01
 1.000000E+00   2.000000E+00   1.680556E-01
 2.000000E+00   3.000000E+00   5.277778E-02
 3.000000E+00   4.000000E+00   2.500000E-02
 4.000000E+00   5.000000E+00   0.000000E-01
 5.000000E+00   1.000000E+01   0.000000E-01
 1.000000E+01   2.000000E+01   0.000000E-01
 2.000000E+01   3.000000E+01   0.000000E-01
 3.000000E+01   5.000000E+01   0.000000E-01

NUMBER,UPPER LIMIT POINTS:=2,0.,5.

UPPER LIMIT PROBABILITIES:=1.

NUMBER,M/CDA POINTS:=3,0.,8.,10.

M/CDA PROBABILITIES:=.7,.3

NUMBER,OUTPUT POINTS:=10,0.,1.,2.,3.,4.,5.,10.,11.,12.,20.

   FIRST           LAST          PROBABILITY
 0.000000E-01   1.000000E+00   4.387499E-01
 1.000000E+00   2.000000E+00   4.387499E-01
 2.000000E+00   3.000000E+00   4.415385E-02
 3.000000E+00   4.000000E+00   4.415385E-02
 4.000000E+00   5.000000E+00   4.415385E-02
 5.000000E+00   1.000000E+01   0.000000E-01
 1.000000E+01   1.100000E+01   0.000000E-01
 1.100000E+01   1.200000E+01   0.000000E-01
 1.200000E+01   2.000000E+01   0.000000E-01

NUMBER,UPPER LIMIT POINTS:=$STOP
READY.
```


73

2.12 SCALE PROBABILITIES

A program called "HEX" was written, utilizing the interval concept, to compute an estimate of probabilities of a series of scale-related quantities. These quantities are all associated with the geometry and mass of a homogeneous spherical particle.

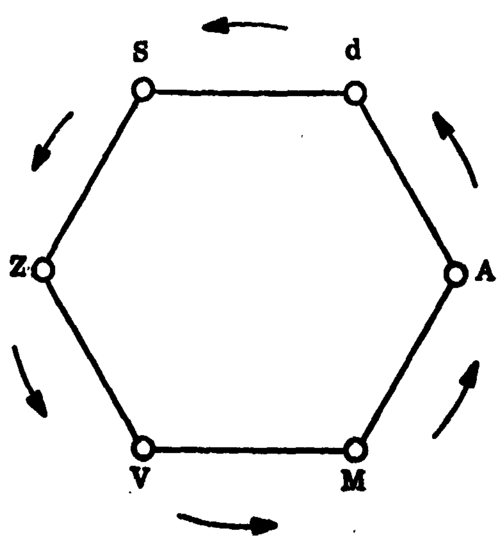
The interval concept is valid when in computing the distribution of a function of several variables, say $\xi = f(x_1, x_2, ---, x_p)$, the intervals (or input grid) are chosen small enough so that $\frac{\partial \xi}{\partial x_j}$ $(j = 1, \ldots, p)$ do not change sign in the given p-dimensional regions.

The quantities under consideration are:

- d, diameter
- s, surface area
- z, ballistic parameter (M/CdA)
- V, volume
- M, mass
- A, cross - sectional area

They are all related in such a way that the above restriction is satisfied. In fact, the functions (30 in all) are all one-to-one for any given interval.

The program HEX allows one to compute the probability distribution of function 1 through 6 given any other. The 30 relations, however, are avoided and reduced to 6 by the suggestion of E. Berger. Instead, the relations are computed recursively in what may be thought of as a counterclock wise direction around the rim of a hexagon.



The six functions are given by:

1. Given d: $S = \pi d^2$
2. Given S: $Z = 2\sigma \sqrt{\frac{S}{\pi}} / 3\, C_D$
3. Given Z: $V = \frac{9\pi}{16} \frac{(C_D Z)^3}{\delta}$
4. Given V: $M = V\delta$
5. Given M: $A = \frac{\pi}{4}\left(\frac{(\delta M)}{\pi\sigma}\right)^{2/3}$
6. Given A: $d = \sqrt{\frac{4A}{\pi}}$

Where σ is the density of the particle and C_D is the (unitless) drag coefficient.

Note that the units cancel out appropriately so that it is necessary only that the given function be consistent in units with δ

For example if we start with d = cm, then S is cm^2, Z is gm/cm^2 (if δ is gm/cm^3), V is cm^3, M is gm, A is cm^2, and d is cm.

The question may arise concerning the loss of significance encountered in "going around the horn."

This turns out to be no serious problem in the test runs encountered so far. To illustrate this, a program was written which, initially calls for δ and C_D.

Following this, it calls for the code and related functional value. The program then computes "around the horn" to the given function and prints this on the following line.

In the following runs, no round off was observed out to seven digits.

```
LOAD LIMITS 06031 17766
?=1.E-3,1.E-2
?=1,.0001

  1.000000E-04
```

That is $\delta = .001$, $C_D = .01$
and $d = S = Z + V = M = A = .0001$
initially

```
?=2,.0001

  1.000000E-04

?=3,.0001

  1.000000E-04

?=4,.0001

  1.000000E-04

?=5,.0001

  1.000000E-04

?=6,.0001

  1.000000E-04

?=1,.1234567

  1.234567E-01

?=2,.1234567

  1.234567E-01
```

Here, all the initial quantity values are chosen to be = .1234567

```
?=3,.1234567

  1.234567E-01

?=4,.1234567

  1.234567E-01

?=5,.1234567

  1.234567E-01

?=6,.1234567

  1.234567E-01

?=$STOP
READY.
```

The program usage is simple and is described briefly in the following.

The program will print the title and code initially.

The first input will be the number (integer) and functional values of the given quantity (in the standard grid format).

Following this is the set of probabilities in the standard interval concept (one less than number of end points).

Next the program calls for three A quantities: given function code (integer), density, and drag coefficient (both floating).

The function code of the desired quantity is then called for (integer).

Finally the number (integer) and functional values of the desired quantity (that is, the output grid) are called for.

The resulting probabilities (out to the last non-zero value) are printed.

The program will treat this as the input distribution for further calculations. Appropriately the code and then the related number and functional values are called for.

The recursion can be halted by giving a function code ≥7. The program will then call for a new input distribution.

A sample series of runs is shown below:

The first and second test the ability to restart over for a called function code ≥ 7.

The third and fourth demonstrate the recursion and the fifth demonstrates the ability to "recreate" the input distribution.

```
LOAD LIMITS 07273 16325

   PROGRAM TO COMPUTE SCALE PROBABILITIES

  FUNCTION CODE
 1, DIAMETER
 2, SURFACE AREA
 3, DRAG PARAMETER
 4, VOLUME
 5, MASS
 6, CROSS-SECTIONAL AREA



 NUMBER,END POINT VALUES:=3,1.,2.,3.

 PROBABILITIES:=.1,.9

 GIVEN FUNCTION CODE, DENSITY, DRAG:=1,2.,3.

 READ NEXT FUNCTION CODE:=2

 NUMBER, POINTS FOR NEXT DENSITY:=5,0.,1.,5.,10.,50.,100.

RESULTING PROBABILITIES
  0.000000E-01  1.971831E-02  5.305165E-02  9.272300E-01
CHECK SUM =   1.000000



 READ NEXT FUNCTION CODE:=7



 NUMBER,END POINT VALUES:=3,1.,2.,3.

 PROBABILITIES:=.1,.9

 FIVEN FUNCTION CODE, DENSITY, DRAG:=1,2.,3.

 READ NEXT FUNCTION CODE:=2

 NUMBER, POINTS FOR NEXT DENSITY:=5,0.,1.,5.,10.,50.

RESULTING PROBABILITIES
  0.000000E-01  1.970831E-02  5.305165E-02  9.272300E-01
CHECK SUM =   1.000000
```

```
READ NEXT FUNCTION CODE:=8



NUMBER,END POINT VALUES:=5,0.,1.,5.,10.,50.

PROBABILITIES:=.1705102,.8294898,0.,0.

GIVEN FUNCTION CODE, DENSITY, DRAG:=3,2.,3.

READ NEXT FUNCTION CODE:=6

NUMBER, POINTS FOR NEXT DENSITY:=10,0.,1.,5.,10.,50.,100.,200.
:=300.,400.,500.

RESULTING PROBABILITIES
  4.283402E-02  1.365266E-01  4.346252E-02  3.477002E-01  4.294267E-01
CHECK SUM =   1.000000



READ NEXT FUNCTION CODE:=$STOP
READY.
```

An associated program with "HEX" is the program known as "SPHERE".

The input is similar to that of "HEX", but the probabilities are not required. Only the function code, density, drag parameter and end points are needed.

The program will provide a spectrum of end points values for all the functions associated with the given end points. Of course, whatever probability is required will hold for all the end point values across each function.

```
PROGRAM TO COMPUTE PARTICLE PARAMETERS

FUNCTION CODE
1, DIAMETER
2, SURFACE AREA
3, DRAG PARAMETER
4, VOLUME
5, MASS
6, CROSS-SECTIONAL AREA


GIVEN FUNCTION CODE, DENSITY, DRAG:=1,2.,4.

NUMBER, POINTS:=5,0.,1.,2.,3.,4.
```

DIAMETER	SURFACE	DRAG	VOLUME	MASS	CROSS-SEC
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01
1.0000E+00	3.1416E+00	3.3333E-01	5.2360E-01	1.0472E+00	7.8540E-01
2.0000E+00	1.2566E+01	6.6667E-01	4.1888E+00	8.3776E+00	3.1416E+00
3.0000E+00	2.8274E+01	1.0000E+00	1.4137E+01	2.8274E+01	7.0686E+00
4.0000E+00	5.0265E+01	1.3333E+00	3.3510E+01	6.7021E+01	1.2566E+01

```
GIVEN FUNCTION CODE, DENSITY, DRAG:=2,2.,4.

NUMBER, POINTS:=5,0.,3.1416,12.566,28.274,50.265
```

DIAMETER	SURFACE	DRAG	VOLUME	MASS	CROSS-SEC
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01
1.0000E+00	3.1416E+00	3.3333E-01	5.2360E-01	1.0472E+00	7.8540E-01
2.0000E+00	1.2566E+01	6.6666E-01	4.1886E+00	8.3772E+00	3.1415E+00
3.0000E+00	2.8274E+01	9.9999E-01	1.4137E+01	2.8274E+01	7.0685E+00
4.0000E+00	5.0265E+01	1.3333E+00	3.3510E+01	6.7020E+01	1.2566E+01

```
GIVEN FUNCTION CODE, DENSITY, DRAG:=3,2.,4.

NUMBER, POINTS:=5,0.,.33333,.66667,1.,1.3333
```

DIAMETER	SURFACE	DRAG	VOLUME	MASS	CROSS-SEC
0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01	0.0000E-01
9.9999E-01	3.1415E+00	3.3333E-01	5.2358E-01	1.0472E+00	7.8539E-01
2.0000E+00	1.2566E+01	6.6667E-01	4.1888E+00	8.3777E+00	3.1416E+00
3.0000E+00	2.8274E+01	1.0000E+00	1.4137E+01	2.8274E+01	7.0685E+00
3.9999E+00	5.0262E+01	1.3333E+00	3.3508E+01	6.7016E+01	1.2566E+01

```
GIVEN FUNCTION CODE, DENSITY, DRAG:=4,2.,4.

NUMBER, POINTS:=5,0.,.52358,4.1889,14.137,33.508

 DIAMETER       SURFACE        DRAG         VOLUME        MASS         CROSS-SEC
 0.0000E-01    0.0000E-01    0.0000E-01    0.0000E-01    0.0000E-01    0.0000E-01
 9.9999E-01    3.1415E+00    3.3333E-01    5.2358E-01    1.0472E+00    7.8538E-01
 2.0000E+00    1.2567E+01    6.6667E-01    4.1889E+00    8.3778E+00    3.1416E+00
 3.0000E+00    2.8274E+01    1.0000E+00    1.4137E+01    2.8274E+01    7.0685E+00
 3.9999E+00    5.0263E+01    1.3333E+00    3.3508E+01    6.7016E+01    1.2566E+01


GIVEN FUNCTION CODE, DENSITY, DRAG:=5,2.,4.

NUMBER, POINTS:=5,0.,1.0472,8.3776,28.274,67.021

 DIAMETER       SURFACE        DRAG         VOLUME        MASS         CROSS-SEC
 0.0000E-01    0.0000E-01    0.0000E-01    0.0000E-01    0.0000E-01    0.0000E-01
 1.0000E+00    3.1416E+00    3.3333E-01    5.2360E-01    1.0472E+00    7.8540E-01
 2.0000E+00    1.2566E+01    6.6667E-01    4.1888E+00    8.3776E+00    3.1416E+00
 3.0000E+00    2.8274E+01    1.0000E+00    1.4137E+01    2.8274E+01    7.0685E+00
 4.0000E+00    5.0266E+01    1.3333E+00    3.3511E+01    6.7021E+01    1.2566E+01


GIVEN FUNCTION CODE, DENSITY, DRAG:=6,2.,4.

NUMBER, POINTS:=5,0.,.78540,3.1416,7.0686,12.566

 DIAMETER       SURFACE        DRAG         VOLUME        MASS         CROSS-SEC
 0.0000E-01    0.0000E-01    0.0000E-01    0.0000E-01    0.0000E-01    0.0000E-01
 1.0000E+00    3.1416E+00    3.3333E-01    5.2360E-01    1.0472E+00    7.8540E-01
 2.0000E+00    1.2566E+01    6.6667E-01    4.1888E+00    8.3776E+00    3.1416E+00
 3.0000E+00    2.8274E+01    1.0000E+00    1.4137E+01    2.8274E+01    7.0686E+00
 3.9999E+00    5.0264E+01    1.3333E+00    3.3509E+01    6.7018E+01    1.2566E+01


GIVEN FUNCTION CODE, DENSITY, DRAG:=$STOP
READY.
```


81

2.13 GENERAL COMBINING OF RANDOM VARIABLES

As an aid for general engineering analysis of the probability of combinations of variates, a program called "CŌMBI" was written to accommodate such problems.

This program is a generalization of "BUGS", in that it now allows for not only addition but subtraction, multiplication and division of random variates.

The user's instructions are similar to that of "BUGS", with the only extra requirement that the binary operation code be entered.

```
 PROGRAM TO COMBINE RANDOM VARIATES

 OPERATION CODE
 ADD, 1
 SUBTRACT, 2
 MULTIPLY, 3
 DIVIDE, 4
 RESTART, 5 OR GREATER



 NUMBER,POINTS FOR FIRST DENSITY:=2,1.,2.,5.

FIRST SET OF PROBABILITIES:=.4,.6

NUMBER,POINTS FOR RESULTING DENSITY:=10,0.,1.,2.,3.,4.,5.,10.
:=20.,50.,100.

READ OPERATION CODE(1-4):=1

NUMBER,POINTS FOR NEXT DENSITY:=2,1.,2.

NEXT SET OF PROBABILITIES:=1.

  ******RESULTING PROBABILITIES******
  0.000000E-01  0.000000E-01  2.000000E-01  3.500000E-01  1.500000E-01
  3.000000E-01
CHECK SUM =  1.000000



READ OPERATION CODE(1-4):=5



 NUMBER,POINTS FOR FIRST DENSITY:=2,1.,2.,5.

FIRST SET OF PROBABILITIES:=.4,.6

NUMBER,POINTS FOR RESULTING DENSITY:=10,0.,1.,2.,3.,4.,5.,10.
:=20.,50.,100.

READ OPERATION CODE(1-4):=2

NUMBER,POINTS FOR NEXT DENSITY:=2,-2.,-1.

NEXT SET OF PROBABILITIES:=1.

  ******RESULTING PROBABILITIES******
  0.000000E-01  0.000000E-01  2.000000E-01  3.500000E-01  1.500000E-01
  3.000000E-01
CHECK SUM =  1.000000
```

```
READ OPERATION CODE(1-4):=5


NUMBER,POINTS FOR FIRST DENSITY:=3,1.,2.,5.

FIRST SET OF PROBABILITIES:=.4,.6

NUMBER,POINTS FOR RESULTING DENSITY:=10,0.,1.,2.,3.,4.,5.,10.
:=20.,50.,100.

READ OPERATION CODE(1-4):=3

NUMBER,POINTS FO.. NEXT DENSITY:=2,1.,2.

NEXT SET OF PROBABILITIES:=1.

  ******RESULTING PROBABILITIES******
  0.000000E-01  1.333333E-01  2.083333E-01  2.083333E-01  7.500000E-02
  3.750000E-01
CHECK SUM =  1.000000


READ OPERATION CODE(1-4):=5


NUMBER,POINTS FOR FIRST DENSITY:=3,1.,2.,5.

FIRST SET OF PROBABILITIES:=.4,.6

NUMBER,POINTS FOR RESULTING DENSITY:=10,0.,1.,2.,3.,4.,5.,10.
:=20.,50.,100.

READ OPERATION CODE(1-4):=4

NUMBER,POINTS FOR NEXT DENSITY:=2,1.,2.

NEXT SET OF PROBABILITIES:=1.

  ******RESULTING PROBABILITIES******
  1.333333E-01  4.166667E-01  1.500000E-01  1.500000E-01  1.500000E-01
CHECK SUM =  1.000000


READ OPERATION CODE(1-4):=$STOP
READY.
```

2.14 REFERENCES FOR SECTION 2

1. M.A. Martin, "Voyager Mars Planetary Quarantine: Thermal Kill of Bacteria During Mars Entry," Section 3 of this report.

2. A. Hald, Statistical Theory with Engineering Applications, John Wiley & Sons, 1952.

3. G.M. Roe, A Computer Algorithm for the Error Integral, TIS 66-C-050, March 1966.

4. T.F. Green, "Preliminary Combinatorial Probability Model for the Voyager Quarantine Problem," VOY-C2-TM1

5. D.A. Korenstein, "Voyager Mars Quarantine Ejected Particle Trajectory Study." VOY-C2-TR7

6. K. Maddock, "Compro-Combinatorial Probability Program." PIR 4T24-068, November 11, 1966.

7. E. Berger, Unpublished notes on the Math Model.

8. T.F. Green, Some Curve Fitting Techniques and Related Applications on the GE Desk-Side Computer System, TIS 66SD314, September 13, 1966.

2.15 ACKNOWLEDGEMENTS

This is to acknowledge the assistance of Mrs. K. Maddock in the preparation of several of the programs described in this section.

SECTION 3

AN APPROACH TO THE EVALUATION OF THE THERMAL INACTIVATION OF MICROORGANISMS DURING MARS ENTRY

By

M. A. Martin

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

86

SECTION 3
ENTRY

3.1 INTRODUCTION AND SUMMARY

An approach to evaluate the potential thermal kill of singular bacteria or small aggregates or clumps during Mars entry has been outlined in Reference 3-1. The concept of a lethality integral (I_L) permits the calculation of the survival ratio (f) under varied conditions.

In the preliminary investigation of the method reported here, the analysis of the effects of four parameters included: the ballistic coefficient, β; the entry angle, γE; the entry velocity, V_E; and the particle emissivity, ϵ. Within the range of values used for these four parameters, it has been possible to determine an algorithm to compute the lethality integral and, hence, the survival ratio (f) for the effects of any combination of these four parameters.

The development of this algorithm is explained, and the results of the functionalization of I_L are evaluated.

Changes made in the choice and the values of the parameters for further investigation of this approach, now in progress, are mentioned.

Results of calculations made on individual or singular living microbial cells or spores carried by nonviable particles are given.

3.2 SURVIVAL RATIO (f) AND LETHALITY INTEGRAL I_L

The kinetics of thermal death of microorganisms can be defined by the differential equation

$$\frac{dN}{N\,dt} = -K(t') \qquad (3\text{-}1)$$

If N is the number of living organisms at time t, Equation 3-1, expresses that the relative rate of change of the number of living organisms is a constant depending upon the temperature t'.

Integration of Equation 3-1 yields

$$f = \frac{N}{N_0} = e^{-\int_{t_0}^{t} K(t')\, dt} \qquad (3\text{-}2)$$

which gives the survival ratio f (ratio of the number N of living organisms at time t to the number N_0 of living organisms at time t_0) as a function of the time history t'(t) of the temperature.

If the temperature, t', is constant, Equation 3-2 reduces to (if $t_0 = 0$)

$$f = \frac{N}{N_0}\ e^{-K(t')\ .\ t} \quad (t' = \text{constant}) \qquad (3\text{-}3)$$

In particular, the time $\tau(t')$ necessary to produce a specified survival ratio $f\tau$ at temperature t' is given by

$$f\tau = e^{-K(t')\ \tau(t')} \quad (t' = \text{constant}) \qquad (3\text{-}4)$$

If the time τ is known for a specified $f\tau$, inversion of Equation 3-4 yields

$$K(t') = -\frac{\ln f\tau}{\tau(t')} \qquad (3\text{-}5)$$

In all our equations, the symbol ln represents a natural logarithm, and the symbol log represents a decimal logarithm.

Replacing, in Equation 3-2, K(t') by its value from Equation 3-5 yields

$$f = \frac{N}{N_0} = e^{\ln f\tau \int_{t_0}^{t} \frac{dt}{\tau(t')}} = 10^{\log f\tau \int_{t_0}^{t} \frac{dt}{\tau(t')}} \qquad (3\text{-}6)$$

We can then define a lethality integral I_L by

$$I_L = \int_{t_o}^{t} \frac{dt}{\tau(t')} \qquad (3\text{-}7)$$

The lethality integral, I_L, is the classical "sterility" considered in the food industry (Reference 3-2).

In our investigation, we have used:

$$f\tau = 10^{-12} \qquad (3\text{-}8)$$

Hence, in our case, Equation 3-6 can be written

$$f = \frac{N}{N_o} = 10^{-12}\, I_L \qquad (3\text{-}9)$$

For the purpose of this investigation, we have assumed that any survival ratio smaller than 10^{-4} is considered as meeting the planctary quarantine requirement. Consequently, the range of values of interest for I_L is from 0 to 1/3.

3.3 DECIMAL REDUCTION TIME (D) AND THERMAL DEATH TIME (F)

Equation 3-3 can be written

$$f = \frac{N}{N_o} = 10^{-K'(t')\,t} \quad (t' = \text{constant}) \qquad (3\text{-}10)$$

with

$$K'(t') = (\log e) \,.\, K(t') = 0.43429\, K(t') \qquad (3\text{-}11)$$

The constant K'(t') is determined experimentally from the measurement of the survival ratio, f, for a known time, t, at the specified temperature, t'. Instead of K'(t'), the biologists use its reciprocal D(t'), hence Equation 3-10 can be written

$$f = \frac{N}{N_o} = 10^{-\frac{t}{D(t')}} \qquad (t' = \text{constant}) \qquad (3\text{-}12)$$

If the time, t, is equal to D(t'), the survival ratio, f, is 1/10; that is, D(t') represents the time to reduce the number of viable organisms in a population to one tenth of its initial value, hence the term Decimal reduction time given to D.

When D(t') is known, the time τ necessary to produce a specified survival ratio $f\tau$ is given by

$$f_\tau = 10^{-\frac{\tau}{D(t')}} \qquad (t' = \text{constant}) \qquad (3\text{-}13)$$

hence

$$\tau = D(t') \cdot \left| -\log f\tau \right| \qquad (3\text{-}14)$$

If N_o is the initial population and N_τ the population at time τ, we have

$$-\log f\tau = -\log \frac{N_\tau}{N_o} = \log N_o - \log N\tau \qquad (3\text{-}15)$$

Hence

$$\tau = D(t') \left[\log N_o - \log N\tau \right] \qquad (3\text{-}16)$$

$N\tau$ may be interpreted as the probability of having, at time τ, one living organism out of an initial population N_o maintained at constant temperature t.

The F value (time to sterilize, more commonly referred to as thermal death time) familiar to the biologists, is derived from the D value by the equation of Schmidt (Reference 3-3), which can be written, with our notations,

$$F = D(t') \left[\log N_o + 1 \right] \qquad (3\text{-}17)$$

or by Hobby's modification of Schmidt's Equation (Reference 4)

$$F = D(t') \left| \log N_0 + a\tau - \log N\tau \right| \tag{3-18}$$

Equation 3-18 differs from Equation 3-16 only by the term a_τ. Hobby takes a_τ equal to 2; Koesterer used the value 1 for a_τ. Actually when N_τ is specified, the value 0 should be used for this term. Adding it is equivalent to replacing N_τ by

$$N'_\tau = 10^{-a\tau} \cdot N_\tau \tag{3-19}$$

3.4 CURVE τ(t') FOR THERMAL RESISTANCE OF DRY SPORES

From D values obtained at temperatures of 80°C, 100°C to 150°C by 5°C increments, and 160°C, (Reference 3-5), and from D values obtained from Decker's work for higher temperature (Reference 3-6), M. Koesterer established a curve (Figure 3-1) of τ as a function of the temperature t' for $N_0 = 10^8$ and $N_\tau = 10^{-4}$, that is, for $f_\tau = 10^{-12}$ as mentioned in Equation 3-8.

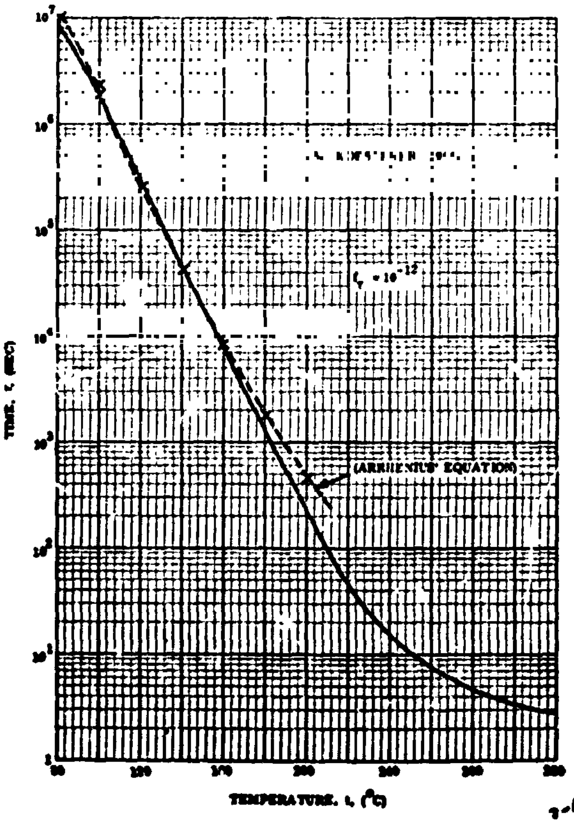


Figure 3-1. Thermal Resistance of Dry Spores

Figure 3-1 is actually a curve of F values. Koesterer used the value 1 for a_τ; hence, the curve really corresponds to a value 10^{-13} for f_τ.

This fact was discovered only recently, and since the purpose of this preliminary investigation was to develop a method for the functionalization of the lethality integral, I_L, the value 12 has been retained for this report. Correction would involve replacing 12 by 13 in Equation 3-9 and in all the calculations which convert I_L into values of survival ratios. The use of 12 rather than 13 provides conservative estimates for the survival ratio.

Arrhenius-Van t'Hoff's theory provides a theoretical expression for D(t'):

$$D(t') = A\ e^{\frac{E_{dh}}{RT}} \qquad (3\text{-}20)$$

In Equation 3-20, A is a constant referred to as frequency factor, R is the gas constant, E_{dh} is the thermal inactivation energy, and T is the absolute temperature (Reference 3-7).

We can transform Equation 3-20 into

$$\log D(t') = \frac{a_D}{T} + b_D \qquad (3\text{-}21)$$

The coefficients a_D and b_D were determined to match Koesterer's curve for the temperatures 100°C and 160°C. The resulting equation was

$$\log D(t') = \frac{6355.1}{T} - 11.826 \qquad (3\text{-}22)$$

Since, in our case, τ is equal to 12 times the D value, values of τ were computed with Equation 3-22 for values of t' from 80°C to 210°C. The results are shown in Figure 3-1 (dotted line).

t can be seen that for high temperatures, the time required to produce a specified survival ratio is less than that predicted by the kinetic theory.

3.5 FUNCTIONALIZATION OF τ (t')

For the purpose of our investigation, Koesterer's curve (Figure 3-1) was assumed to represent actual values.

Figure 3-1 represents log τ as a function of the temperature t'. For t' between 80°C and 210°C, the curve is a straight line. The portion of the curve for t' between 210°C and 320°C was approximated by a portion of a rectangular hyperbola.

Specifically, the following functions have been used:

$$\tau = e^{(-0.089810\,t' + 23.443)} \qquad \text{for } t' \le 210^{\circ}C \qquad (3\text{-}23)$$

$$\tau = e^{\frac{-0.797776\,t' + 416.679}{t' - 155.656}} \qquad \text{for } t' > 210^{\circ}C \qquad (3\text{-}24)$$

In our calculations, Equation 3-24 was also applied to temperatures higher than the maximum temperature (320°C) for which experimental data exist. The reason is that Equation 3-24 still provides reasonable extrapolated values for these higher temperatures. Furthermore, at 320°C, an exposure time of 0.89 second is sufficient to produce a survival ratio of 10^{-4}; hence, when that temperature is reached or exceeded during Mars entry, the thermal kill is almost instantaneous. The exact value of n in the survival ratio 10^{-n} cannot be computed accurately, but is not pertinent when n is larger than 4.

Table 3-1 provides a synopsis of the quality of the functionalization of τ. Between 100°C and 210°C, the % error should be theoretically zero, since Koesterer's curve is a straight line and Equation 3-23 represents also a straight line when τ is plotted in logarithmic scale; the small errors are due to errors in interpreting the curve.

Table 3-1 shows that for the range of experimental temperatures, the value of τ functionalized by Equation 3-23 or 3-24 does not differ from the observed value by more than a few percent.

Table 3-1. Functionalization of τ

t' (°C)	τ (seconds) Read	τ (seconds) Calculated	Percentage of Error	t' (°C)	τ (seconds) Read	τ (seconds) Calculated	Percentage of Error
100	1 920 000.	1 910 000.	-0.5	260	7.5	7.43	-0.9
120	313 000.	317 000.	1.3	270	6.0	5.81	-3.2
140	51 000.	52 500.	2.9	280	4.8	4.73	-1.5
160	8 4[illegible]0.	8 [illegible]20.	3.8	290	4.0	3.97	-0.7
180	1 390.	1.450.	4.3	300	3.4	3.42	0.6
200	235.	240.	2.1	310	3.0	3.00	0.0
210	100.	98.0	-2.0	320	2.66	2.67	0.4
220	43.5	42.4	-2.5	340		2.20	
230	23.0	23.0	0.0	360		1.88	
240	14.3	14.4	0.7	380		1.66	
250	10.0	10.0	0.0	400		1.49	

3.6 COMPUTER PROGRAMS FOR DETERMINING I_L

The method of determination of a particle temperature during an entry trajectory is described in Reference 3-8.

A trajectory program is first run with selected initial conditions to provide values of the free molecular heat flux $\dot{q}_{FM}$ (in Btu/ft^2-sec) as functions of time. At each instant, $\dot{q}_{FM}$ is proportional to the atmospheric density ρ_a at the particle position and to the cube of the particle velocity

$$q_{FM} \therefore \rho_a V^3 \tag{3-25}$$

The heat balance equation can be written (for a sphere)

$$\dot{q}_{FM} + \alpha \bar{S} = K_1 \epsilon \sigma T^4 + K_2 \ r\rho \ C_P \ \frac{dT}{dt} \tag{3-26}$$

In Equation 3-26,

- α is the solar absorptivity of the particle
- $\bar{S}$ is the solar constant for Mars (a value of 0.0653 Btu/ft^2-sec was used)
- ϵ is the particle emissivity
- σ is the Stefan-Boltzmann constant (4.76 x 10^{-13} Btu/ft-sec-^oR)
- r is the particle radius (ft)
- ρ is the particle density (lb/ft^3)
- C_P is the specific heat capacity of the particle (Btu/lb-^oR)
- T is the absolute temperature of the particle (^oR)
- K_1 has value 4 for a spherical particle, and π for a cylindrical particle
- K_2 has value 4/3 for a spherical particle, and $\pi/2$ for a cylindrical particle

The input to the Thermodynamics program includes the values of $\dot{q}_{FM}$ for all the times needed by the computer program and the value T_0 of T at the initial time t_0. At each time t_n, the derivative $\frac{dT}{dt}$ is calculated by Equation 3-26, and the value T for the next time is obtained by integration of this derivative.

A subroutine has been added to the thermodynamics program to compute I_L as follows: at each time t_n, the absolute temperature T_n of the particle (in ^oR) is converted to a value t'_n in ^oC from which $\frac{1}{\tau_n}$ is computed from Equation 3-23 or 3-24, by simple change of sign in the exponent. If $I_{L(n-1)}$ is the value of the integral I_L up to the preceding time t_{n-1}, the corresponding value of τ_n, the value I_{Ln} of I_L at time t_n is computed, according to the trapezoidal rule of integration, by

$$I_{Ln} = I_{L(n-1)} + \left(\frac{t_n - t_{n-1}}{2}\right)\left(\frac{1}{\tau_n} + \frac{1}{\tau_{n-1}}\right) \qquad (3\text{-}27)$$

The value of I_{Ln} for the last value of t_n processed by the thermodynamics program represents the lethality integral I_L.

3.7 PARAMETERS AFFECTING I_L

In the preliminary investigation reported here, the particles were assumed to be spherical. It is shown in Reference 3-8 that the temperatures obtained with cylindrical particles, with the end effects neglected, are slightly higher than those obtained with spherical particles and hence produce slightly higher lethality integrals. Consideration of spherical particles is then favorable to survival of the particle.

The particles have been assumed to have a temperature of 500^oR at the start of the entry trajectory, the altitude h_o of which has been maintained at the constant value of 721,000 feet (entry altitude).

The VM3 atmosphere has been selected because it is less dense than the VM8 atmosphere and hence provides conservative estimates for particle survival.

The solar absorptivity α has been maintained to 1 (that is, the particle has been assumed to be in daytime entry and to absorb all the solar energy it receives).

The particle specific heat capacity C_P has been maintained equal to 0.2 Btu/lb-°R. A constant drag coefficient c_D equal to 2 has been used.

The particles have been assumed to have a constant density equal to 68.6 lb/ft^3 or 2.132 slugs/ft^3.

We have varied only four parameters:

a. The ballistic coefficient

$$\beta = \frac{M}{c_D A} \tag{3-28}$$

where M is the particle mass (in slugs) and A the area (in ft^2) of the particle section. The ballistic coefficient β is related to the particle radius (in feet) by

$$\beta = \frac{M}{c_D A} = \frac{\frac{4}{3}\pi r^3 \rho}{c_D \cdot \pi r^2} = \frac{4}{3} \frac{\rho}{c_D} r \tag{3-29}$$

hence

$$r = \frac{3}{4} \frac{c_D}{\rho} \beta \tag{3-30}$$

Since c_D and ρ have been maintained constant, r was determined by the value of β. Specifically, the three sets of values of β and r we have used are:

β	4.0×10^{-5}	2.2×10^{-4}	4.0×10^{-4}	slugs/ft^2
r	2.81×10^{-5}	1.55×10^{-4}	2.81×10^{-4}	feet

These values are in agreement with those of Table 4-1 of Reference 3-8.

b. Five values of the entry angle γ_E (angle of the trajectory with the local horizontal at entry, counted positive downward) have been used:

5 degrees, 10 degrees, 20 degrees, 45 degrees, 90 degrees (downward vertical)

c. Five values of the entry velocity, V_E have been used:

11306, 15000, 19000, 22500, 26000 ft/sec

d. Nine values of the particle emissivity ϵ have been used:

0.1 to 0.9 by 0.1 increment

The lethality integral I_L is a monotonic function of some of these parameters. Specifically:

I_L increases when the entry temperature T_0	increases
I_L increases when the solar absorptivity α	increases
I_L increases when the ballistic coefficient β	increases
I_L increases when the entry velocity, V_E	increases
I_L decreases when the emissivity ϵ	increases
I_L decreases when the specific heat capacity (C_P)	increases

The variation of I_L with the entry angle γ_E could not be predicted. Effectively, as shown in Figure 3-2, when γ_E increases, the maximum temperature increases, but the duration of the temperature history which significantly contributes to I_L decreases.

3.8 COMPUTER RUNS FOR DETERMINING I_L

A total of 75 trajectories was reouired to represent all the possible combinations of three values of β, five values of γ_E, and five values of V_E. If all the nine possible values of emissivity, ϵ, had been used for each trajectory, 675 computer runs would have been necessary. The resulting computer time and manpower necessary to prepare all the input for the computer runs would have been prohibitively high. Furthermore, a large number of values of I_L would have been outside the range of interest.

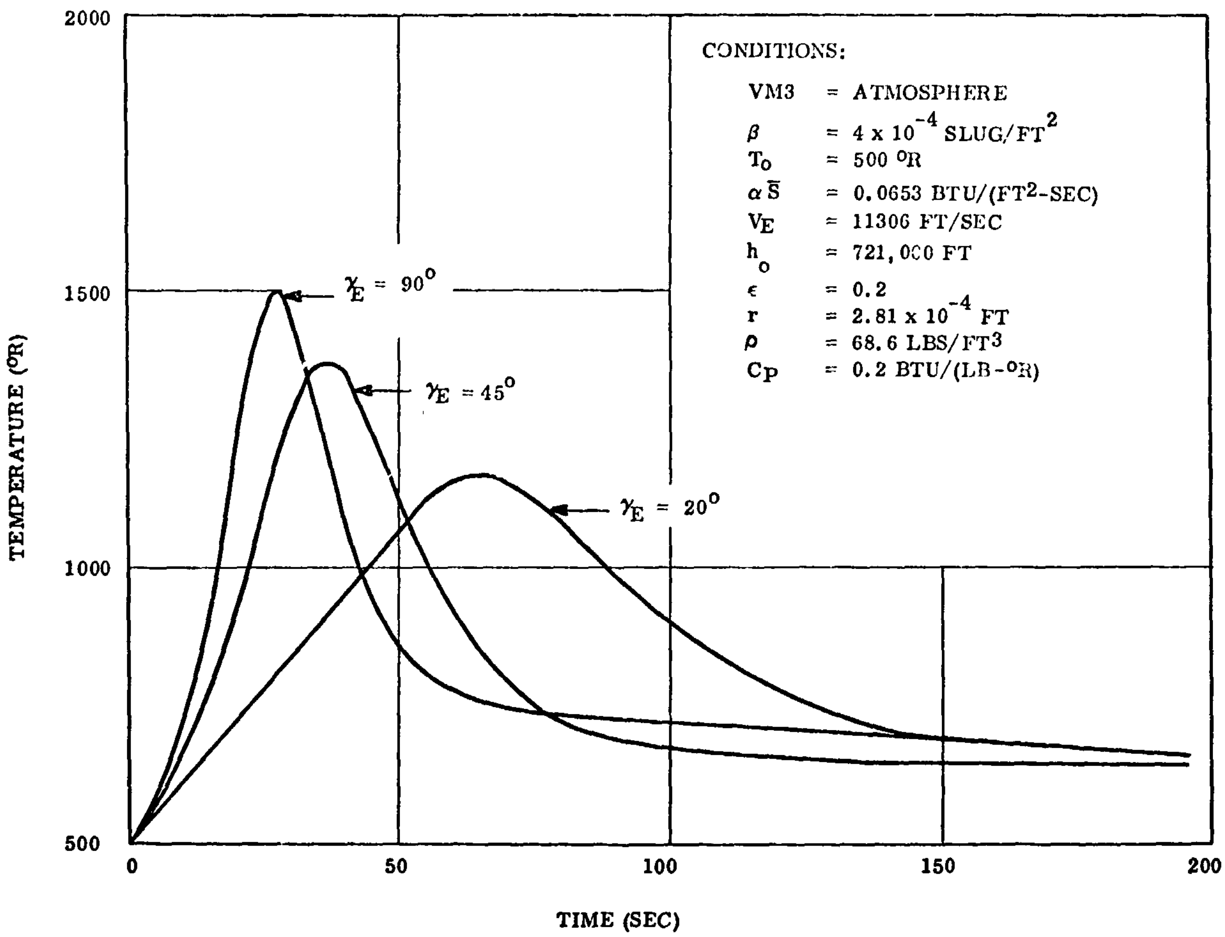


Figure 3-2. Temperature Histories

Consequently, computer runs were specified by small batches of 10 to 16. No batch was specified until the results of the preceding batch had been obtained and analyzed. In that manner, more educated guesses could be made as additional results became available.

Table 3-2 shows a synopsis of all the computer runs which were made and for which the lethality integral I_L was calculated.

A total of less than 150 computer runs was made, that is, about one-fifth of the number 675 of possible combinations of the four variables.

Table 3-2. Lethality Integral I_L for Various Combinations of γ_E, β, V_E, ϵ

		γ_E = 5 Degrees			γ_E = 10 Degrees			γ_E = 20 Degrees			γ_E = 45 Degrees			γ_E = 90 Degrees		
V_E	ϵ \ β	4.0×10^{-5}	2.2×10^{-4}	4.0×10^{-4}	4.0×10^{-5}	2.2×10^{-4}	4.0×10^{-4}	4.0×10^{-5}	2.2×10^{-4}	4.0×10^{-4}	4.0×10^{-5}	2.2×10^{-4}	4.0×10^{-4}	4.0×10^{-5}	2.2×10^{-4}	4.0×10^{-4}
	0.1	0.175	19.659	53.738	0.439	29.703		1.579			4.282			5.441		
	0.2	0.	0.074	3.497	0.000	1.689	14.786	0.002	7.58	23.38	0.041	12.84		0.290		
	0.3			0.065		0.030	2.478		1.56					0.004		
11300	0.4			0.004			0.179		0.139							
ft/	0.5						0.018		0.014	1.64		1.34			2.99	
sec	0.6									0.431		0.470			1.686	
	0.7						0.001			0.087		0.124			0.883	
	0.8											0.030			0.417	
	0.9						0.			0.007		0.009	1.10		0.172	2.71
	0.1	1.547			4.417											
	0.2	0.002	0.688		0.008			0.175								
	0.3		0.011	0.134							0.244					
15000	0.4			0.008		0.273								0.162		
ft/	0.5															
sec	0.6						0.594									
	0.7						0.116		0.361							
	0.8															
	0.9									2.467		2.129			3.508	
	0.1	9.047	60.703	No	12.978											
	0.2	0.046	6.823	entry	0.524	38.33		4.42			6.65			7.08		
	0.3		0.209		0.008			0.504								
19000	0.4		0.011		0.001			0.027			1.170					
ft/	0.5		0.001			3.14					0.375			1.65		
sec	0.6					0.934					0.086			0.554		
	0.7					0.202		0.000	6.17		0.017			0.227		
	0.8					0.050								0.075		
	0.9					0.015	1.51	0.	2.84	12.05	0.001	7.27	13.65	0.022	7.66	
	0.1			No												
	0.2	0.883		entry												
	0.3	0.014	3.043		0.349											
22500	0.4		0.214		0.017			0.939								
ft/	0.5							0.175								
sec	0.6															
	0.7										0.639					
	0.8					1.904					0.289			0.963		
	0.9					0.827								0.616		
	0.1		No	No												
	0.2	4.87	entry	entry	11.55			13.75								
	0.3	0.418			3.530											
36000	0.4	0.021			0.641											
ft/	0.5	0.002			0.068			1.844								
sec	0.6				0.011			0.739								
	0.7				0.002			0.233								
	0.8							0.059								
	0.9				0.000	6.58	30.817	0.017	14.16		1.08			1.44		17.04

3.9 VARIATION OF I_L WITH EMISSIVITY ϵ. INTERMEDIATE VARIABLE $\epsilon 2$

As it can be shown in Table 3-2, I_L varies quite nonlinearly with ϵ. It was then natural to plot log I_L as function of I_L.

Figure 3-3 shows a few of the curves which were plotted for $\beta = 2.2 \times 10^{-4}$ slug/ft^2. It can be seen that for I_L between 0.01 and 0.3 (range close to the range of interest 0 to 0.333), the various curves can be approximated by straight lines.

We could then define these straight lines by two parameters. We selected the value ϵ_1 of the emissivity for I_L equal to 0.3 and the value ϵ_2 of the emissivity for I_L equal to 0.01.

In order to determine the envelope of these straight lines, ϵ_2 was plotted as function of ϵ_1. Figure 3-4 shows the result: a straight line passing through the origin.

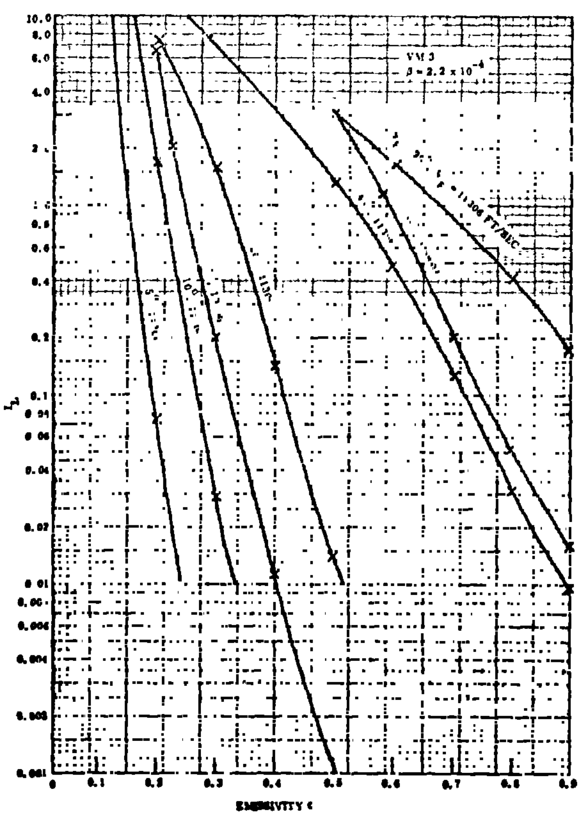


Figure 3-3. Variation of I_L with ϵ

This indicated that all the straight lines passed through a point of the log I_L axis, that is, having for coordinates in Figure 3-3.

$$\epsilon = 0 \qquad \log I_{LO} = c \qquad (3\text{-}31)$$

Effectively, the equation of any of the straight lines is

$$\log I_L = a\epsilon + c \quad (3\text{-}32)$$

Expressing that the line goes through the points (ϵ_1, log 0.3) and (ϵ_2, log 0.01) yields the equations

$$-0.52288 = \log 0.3 = a\,\epsilon_1 + c \quad (3\text{-}33)$$

$$-2. \quad = \log 0.01 = a\,\epsilon_2 + c \quad (3\text{-}34)$$

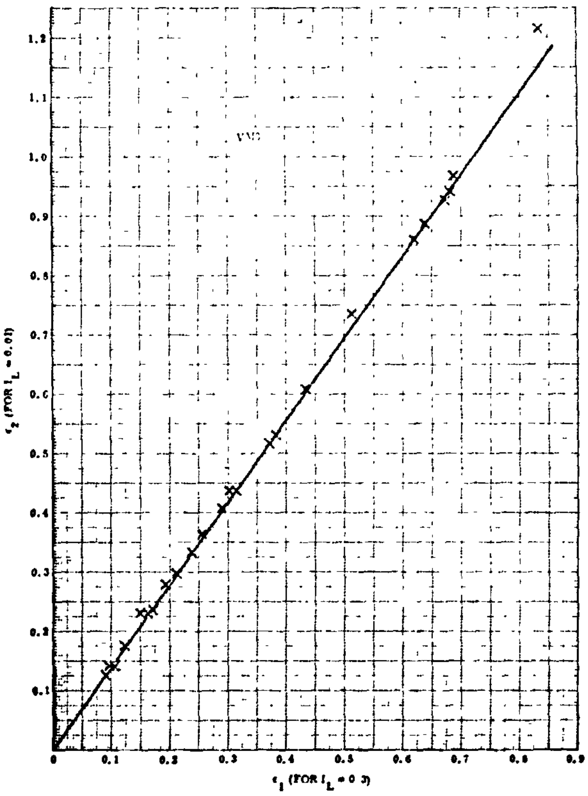


Figure 3-4. Variation of ϵ_2 with ϵ_1

From Figure 3-4, we obtained

$$\epsilon_2 = 1.38235\,\epsilon_1 \quad (3\text{-}35)$$

Hence

$$a\,\epsilon_1 = -0.52288 - c \quad (3\text{-}36)$$

$$a\,\epsilon_2 = -2 \quad - c \quad (3\text{-}37)$$

and

$$1.38235 = \frac{\epsilon_2}{\epsilon_1} = \frac{a\,\epsilon_2}{a\,\epsilon_1} = \frac{-2 - c}{-0.52288 - c} \quad (3\text{-}38)$$

Solution of Equation 3-38 yields

$$c = 3.340 = \text{constant} \tag{3-39}$$

Since all the straight lines of Figure 3-4 passed through the same point defined by Equations 3-31 and 3-39, each straight line could be defined by a single parameter. We selected ϵ_2.

Elimination of a between Equations 3-32 and 3-34 yields

$$\log I_L = c - \frac{(c + 2)\, \epsilon}{\epsilon_2} \tag{3-40}$$

or

$$\epsilon_2 = \frac{(c + 2)\, \epsilon}{c - \log I_L} \tag{3-41}$$

For each data point (combination β, γ_E, V_E, ϵ, and corresponding I_L obtained by computer run) it is possible to compute ϵ_2 by Equation 3-41.

The problem was then to express ϵ_2 as function of the three remaining variables β, γ_E and V_E.

The maximum value of ϵ_2 of interest is obtained for $\epsilon = 0.9$ and $I_L = 1/3$ and is 1.244 with the value of Equation 3-39 used for c.

3.10 VARIATION OF ϵ_2 WITH BALLISTIC COEFFICIENT β

Figure 3-5 shows a few of the plots of ϵ_2 as function of β. (Figure 3-5 is for $V_E = 11306$ ft/sec only). It was found that every time we had data for the three values of β and a value of ϵ_2 not exceeding 1.244, the three points belonging to the case combination (γ_E, V_E) were on a straight line.

102

In other words, within the range of values for β that we have used, ϵ_2 was a linear function of β.

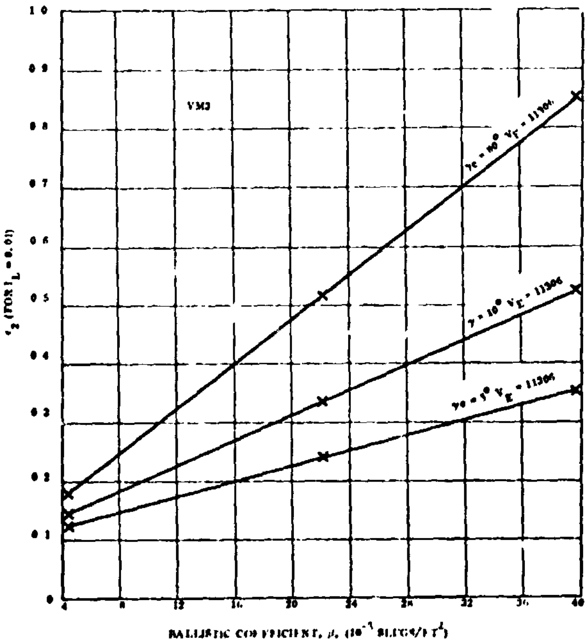


Figure 3-5. Variation of ϵ_2 with β

3.11 VARIATION OF ϵ_2 WITH ENTRY ANGLE (γ_E)

It was found that ϵ_2 was not a linear function of γ_E, but was a linear function of $(90^{\circ} - \gamma_E)^2$. This can be shown for instance, in Figure 3-6, which represents curves obtained with $\beta = 4 \times 10^{-5}$ slug/ft^2.

This quadratic form of the function, with symmetry with respect to 90 degrees, can easily be understood: the entry angles $90 - \alpha$ and $90 + \alpha$ define two trajectories symmetrical with respect to the downward vertical.

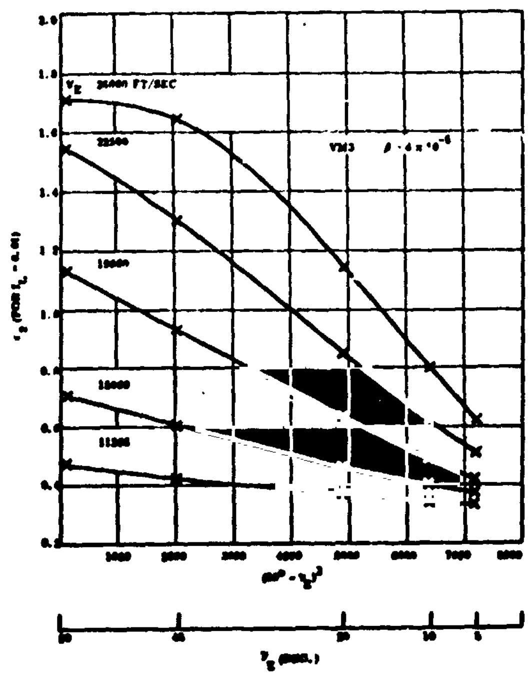


Figure 3-6. Variation of ϵ_2 with γ_E

3.12 VARIATION OF ϵ_2 WITH ENTRY VELOCITY (V_E)

It was found that ϵ_2 could be represented by a linear function of the cube of the entry velocity V_E. This can be shown, for instance, in Figure 3-7, which represents curves obtained with $\beta = 4 \times 10^{-5}$ slug/ft^2.

It seems interesting to compare this dependence of I_L on V_E^3 with the dependence of $\dot{q}_{FM}$ on V^3 as shown in Equation 3-25.

3.13 FUNCTIONALIZATION OF I_L

For convenience, the functions β, $(90 - \gamma_E)^2$, and V_E^3 have been scaled and replaced by

$$\bar{\beta} = 10^4 \beta \qquad \bar{\gamma} = \left(\frac{90 - \gamma_E}{100}\right)^2$$

$$\bar{V} = \left(\frac{V_E}{10^4}\right)^{33} \qquad (3\text{-}42)$$

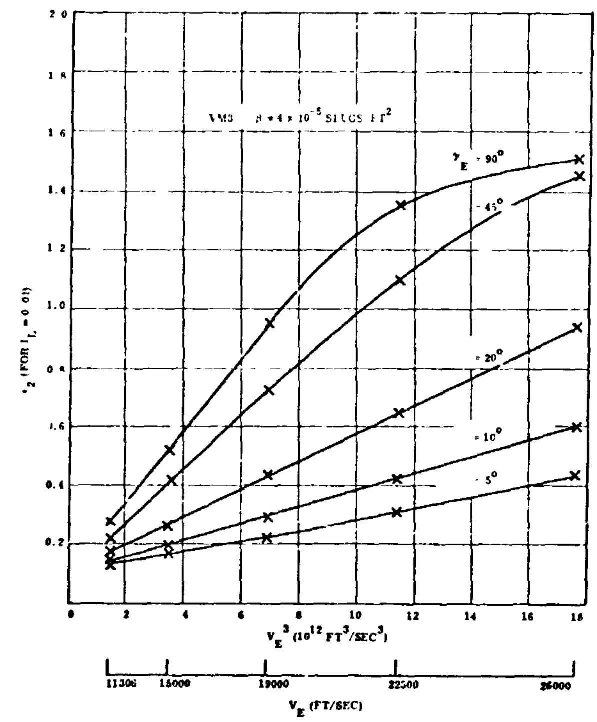


Figure 3-7. Variation of ϵ_2 With V_E

Since ϵ_2 is a linear function of $\bar{\beta}$, $\bar{\gamma}$, $\bar{V}$, it can be represented by a sum of terms of the form

$$\bar{\beta}^{\,i} \; \bar{\gamma}^{\,j} \; \bar{V}^{\,k}$$

in which the exponents i, j, k can take values 0 or 1. Hence ϵ_2 can be approximated by a sum ϵ_{2c} of eight such terms. Specifically

$$\epsilon_2 \cong \epsilon_{2c} = a_1 + a_2 \bar{\beta} + a_3 \bar{\gamma} + a_4 \bar{V} + a_5 \bar{\beta}\bar{\gamma}$$

$$+ a_6 \bar{\gamma}\bar{V} + a_7 \bar{V}\bar{\beta} + a_8 \bar{\beta}\bar{\gamma}\bar{V} \qquad (3\text{-}43)$$

For each data point $\bar{\beta}$, $\bar{\gamma}$, $\bar{V}$ can be calculated by Equation 3-42 and ϵ_2 by Equation 3-41. We can then obtain the coefficients of the linear combination (3-43) of known functions, for instance, by least square fit.

We have assigned a weight (1 or 0) to each data point. This has permitted us to make basically all the same calculations on all data points, but to eliminate from the least square fit the data points which did not agree closely enough with the calculated fit. In that manner, we processed 119 data points, but considered only 63 data points for determining the least square fit. The other 56 points had values of I_L larger than 0.333 and hence corresponded to complete kill (survival ratios f smaller than 10^{-4}). For these values of f smaller than 10^{-4}, the exact value of f was of no interest, as long as it was smaller than 10^{-4}. We gave a weight of zero to only 3 of the basic 63 data points.

An iterative process, operating on the value c, was used to improve the accuracy of the fit. The procedure can be explained as follows:

For a given value of c (the first value was 3.340), ϵ_2 can be computed for each data point by Equation 3-41 and the values $\bar{\beta}$, $\bar{\gamma}$, $\bar{V}$ by (3-42). After all the data points have been processed, the coefficients a_1 to a_8 of Equation 3-43 are obtained by the classical weighted least square method. Then for each data point a computed value ϵ_{2c} is obtained by Equation 3-43, a corresponding computed lethality integral I_{LC} is obtained, according to Equation 3-40 by

$$\log I_{LC} = c - \frac{(c+2)\,\epsilon}{\epsilon_{2c}} \tag{3-44}$$

then

$$I_{LC} = 10^{\log I_{LC}} \tag{3-45}$$

The corresponding computed survival ratio f_c is obtained according to Equation 3-9 by

$$f_c = 10^{-12\, I_{LC}} \tag{3-46}$$

For each data point, the residual

$$\Delta I_L = I_L - I_{LC} \tag{3-47}$$

difference between the true value I_L and the computed value I_{LC} is calculated. The ratio between the true survival ratio (f) and the computed survival ratio (f_c) is related to this residual by

$$\frac{f}{f_c} = 10^{-12\,\Delta I_L} \qquad (3\text{-}48)$$

A weighted root mean square value of the residual ΔI_L is obtained for all the N data points by

$$RMS = \sqrt{\frac{\sum_{n=1}^{N} W_n \, \Delta I_L^{\,2}}{\sum_{n=1}^{N} W_n}} \qquad (3\text{-}49)$$

W_n being the weight assigned to the data point of sequential order (n).

The process is repeated for various values of (c) until, by trial and error, a practical minimum value is obtained for RMS. Figure 3-8 summarizes the results.

The minimum RMS was obtained with

$$c = 2.87 \qquad (3\text{-}50)$$

and had for value

$$RMS = 0.02047$$

corresponding to a "root mean square" ratio $\frac{f}{f_c}$ equal to 1.75.

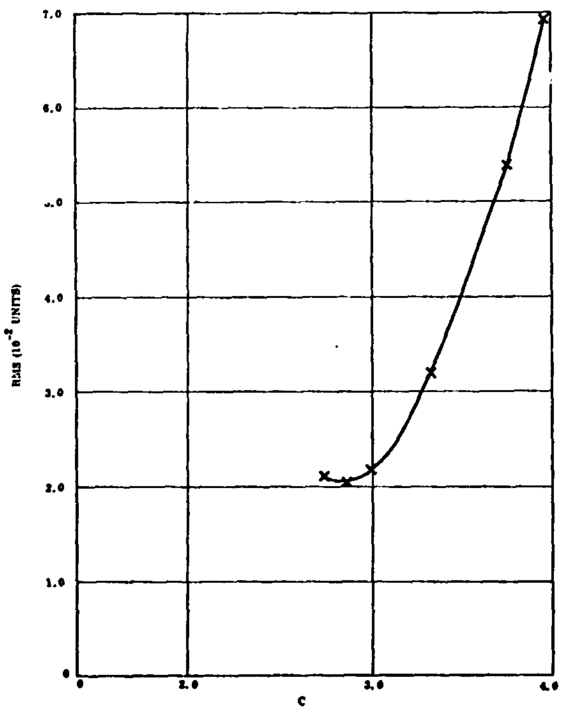


Figure 3-8. Variation of RMS With c

Table 3-3 shows the results of the calculations on each of the 119 data points with the value 2.87 for c. In that table the first eight numbers are the coefficients a_1 to a_8 of Equation 3-43. The last line shows the numerator and denominator of the fraction in Equation 3-49 and the RMS.

Table 3-3. Functionalization of I_L for c = 2.87

2.870000E+00 1.465678E-01 -1.630667E-02 -1.016700E-01 -1.899623E-02
9.516288E-02 4.684823E-02 3.397403E-01 -4.598621E-0

I	W	β	γ_E	V_E	ϵ	ϵ_L	ϵ_{2c}	I_L	I_{LC}	ΔI_L
1	0.	0.00004	5.00	11306.	0.10	0.1343	0.1199	0.1750	0.0642	0.1108
2	1.	0.00004	5.00	15000.	0.20	0.1711	0.1543	0.0675	0.0664	0.0011
3	1.	0.00004	5.00	19000.	0.20	0.2282	0.2165	0.0400	0.0235	0.0165
4	1.	0.00004	5.00	22500.	0.30	0.3091	0.2974	0.0139	0.0091	0.0048
5	1.	0.00004	5.00	26000.	0.40	0.4263	0.4078	0.0210	0.0124	0.0086
6	1.	0.00004	5.00	26000.	0.50	0.4372	0.4076	0.0020	0.0008	0.0012
7	1.	0.00004	10.00	11306.	0.20	0.1238	0.1415	0.0000	0.0001	-0.0001
8	1.	0.00004	10.00	15000.	0.20	0.1961	0.1977	0.0080	0.0088	-0.0008
9	1.	0.00004	10.00	19000.	0.30	0.2941	0.2993	0.0078	0.0096	-0.0018
10	1.	0.00004	10.00	19000.	0.40	0.3319	0.2993	0.0010	0.0002	0.0008
11	1.	0.00004	10.00	22500.	0.30	0.4391	0.4315	0.3490	0.3047	0.0443
12	1.	0.00004	10.00	22500.	0.40	0.4201	0.4315	0.0171	0.0227	-0.0056
13	1.	0.00004	10.00	26000.	0.50	0.6031	0.6115	0.0680	0.0776	-0.0096
14	1.	0.00004	10.00	26000.	0.60	0.6051	0.6118	0.0110	0.0124	-0.0014
15	1.	0.00004	10.00	26000.	0.70	0.6121	0.6118	0.0020	0.0020	0.0000
16	1.	0.00004	20.00	11306.	0.20	0.1735	0.1507	0.0018	0.0030	-0.0012
17	1.	0.00004	20.00	15000.	0.20	0.2665	0.2767	0.1750	0.2238	-0.0488
18	1.	0.00004	20.00	19000.	0.40	0.4389	0.4499	0.0270	0.0347	-0.0077
19	1.	0.00004	20.00	19000.	0.70	0.4332	0.4499	0.0000	0.0060	-0.0060
20	1.	0.00004	20.00	22500.	0.50	0.6714	0.6752	0.1750	0.1836	-0.0086
21	1.	0.00004	20.00	26000.	0.70	0.9733	0.9828	0.2333	0.2528	-0.0196
22	1.	0.00004	20.00	26000.	0.80	0.9504	0.9828	0.0590	0.0805	-0.0215
23	1.	0.00004	20.00	26000.	0.90	0.9447	0.9828	0.0170	0.0257	-0.0087
24	1.	0.00004	45.00	11306.	0.20	0.2288	0.2560	0.0410	0.1162	-0.0752
25	1.	0.00004	45.00	15000.	0.30	0.4195	0.4280	0.2440	0.2861	-0.0421
26	1.	0.00004	45.00	19000.	0.60	0.74[illegible]	0.7386	0.0860	0.0820	0.0040
27	1.	0.00004	45.00	19000.	0.70	0.7348	0.7386	0.0176	0.0186	-0.0010
28	1.	0.00004	45.00	19000.	0.90	0.7467	0.7386	0.0010	0.0009	0.0001
29	1.	0.00004	45.00	22500.	0.80	1.1428	1.1425	0.2870	0.2864	0.0006
30	0.	0.00004	90.00	11306.	0.20	0.2858	0.3090	0.2900	0.5221	-0.2321
31	1.	0.00004	90.00	11306.	0.30	0.2773	0.3090	0.0040	0.0139	-0.0099
32	1.	0.00004	90.00	15000.	0.40	0.5322	0.5340	0.1620	0.1683	-0.0063
33	1.	0.00004	90.00	19000.	0.70	0.9701	0.9419	0.2250	0.1761	0.0489
34	1.	0.00004	90.00	19000.	0.80	0.9752	0.9419	0.0750	0.0541	0.0209
35	1.	0.00004	90.00	19000.	0.90	0.9661	0.9419	0.0220	0.0165	0.0055
36	1.	0.00022	5.00	11306.	0.20	0.2435	0.2338	0.0740	0.0505	0.0235
37	1.	0.00022	5.00	15000.	0.30	0.3033	0.2942	0.0113	0.0080	0.0033
38	1.	0.00022	5.00	19000.	0.30	0.4116	0.4034	0.2090	0.1771	0.0319
39	1.	0.00022	5.00	19000.	0.40	0.4034	0.4034	0.0110	0.0110	0.0000
40	1.	0.00022	5.00	19000.	0.50	0.4145	0.4034	0.0010	0.0007	0.0003
41	1.	0.00022	5.00	22500.	[illegible]	0.5563	0.5454	0.2140	0.1986	0.0154
42	1.	0.00022	10.00	11306.	0.30	0.3326	0.3329	0.0200	0.0373	-0.0173
43	1.	0.00022	10.00	15000.	0.40	0.5673	0.5540	0.2730	0.2256	0.0474
44	1.	0.00022	10.00	19000.	0.70	0.9563	0.9405	0.2020	0.1759	0.0261
45	1.	0.00022	10.00	19000.	[illegible]	0.9341	0.9405	0.0500	0.0534	-0.0034
46	1.	0.00022	10.00	19000.	[illegible]	0.9338	0.9405	0.0150	0.0162	-0.0012
47	1.	0.00022	20.00	11306.	0.40	0.5227	0.5329	0.1390	0.1640	-0.0250
48	1.	0.00022	20.00	15000.	0.70	1.0291	1.0263	0.3610	0.3535	0.0075
49	1.	0.00022	20.00	11306.	0.50	0.5155	0.5329	0.0140	0.0200	-0.0060
50	1.	0.00022	45.00	11306.	0.70	0.9027	0.9029	0.1040	0.1042	-0.0002
51	1.	0.00022	45.00	11306.	0.80	0.9069	0.9029	0.0300	0.0359	-0.0059
52	1.	0.00022	45.00	11306.	0.90	0.8916	0.9029	0.0090	0.0104	-0.0014
53	1.	0.00022	90.00	11306.	0.90	1.[illegible]	1.1634	0.1720	0.1267	0.0453
54	1.	0.00040	5.00	11306.	0.30	0.3641	0.3477	0.0650	0.0465	0.0185
55	1.	0.00040	5.00	11306.	0.40	0.3698	0.3477	0.0040	0.0018	0.0022
56	0.	0.00040	5.00	15000.	0.30	[illegible]	0.4541	0.1340	0.3197	-0.1857
57	1.	0.00040	5.00	15000.	0.40	0.391[illegible]	0.4541	0.0079	0.0242	-0.0163
58	1.	0.00040	10.00	11306.	0.40	0.5305	0.5364	0.1790	0.1785	0.0005
59	1.	0.00040	10.00	11306.	0.50	0.5277	0.5364	0.0180	0.0222	-0.0042
60	1.	0.00040	10.00	11306.	0.70	[illegible]	0.5364	0.0010	0.0003	0.0007
61	1.	0.00040	10.00	15000.	0.70	[illegible]	0.9103	0.1160	0.1333	-0.0173
62	1.	0.00040	20.00	11306.	0.70	0.2673	0.8851	0.0870	0.1044	-0.0174

FOLDOUT FRAME

54	1.	0.00040	5.00	11306.	0.30	0.3691	0.3477	0.0650	0.0465	0.0185
55	1.	0.00040	5.00	11306.	0.40	0.3698	0.3477	0.0040	0.0018	0.0022
56	1.	0.00040	5.00	15000.	0.50	[illegible]	0.4341	0.1340	0.3197	-0.1[illegible]
57	1.	0.00040	5.00	15000.	[illegible]	0.391[illegible]	0.4341	0.0079	0.0242	-0.0163
58	1.	0.00040	10.00	11306.	0.40	0.5365	0.5364	0.1790	0.1785	[illegible]005
59	1.	0.00040	10.00	11306.	0.50	0.5277	0.5364	0.0180	0.0222	-0.0042
60	1.	0.00040	10.00	11306.	0.70	0.5707	0.5364	0.0010	0.0003	0.0007
61	1.	0.00040	10.00	15000.	0.70	0.9250	0.9103	0.1160	0.1333	-0.0173
62	1.	0.00040	20.00	11306.	0.70	0.7673	0.6851	0.0870	0.1044	-0.0174
63	1.	0.00040	20.00	11306.	0.90	0.[illegible]723	0.6851	0.0070	0.0083	-0.0013
64	0.	0.00004	5.00	26000.	0.30	0.4497	0.4075	0.4160	0.1937	0.2243
65	0.	0.00004	10.00	11306.	0.10	0.1509	0.1415	0.4390	0.2678	0.1712
66	0.	0.00004	20.00	19000.	0.30	0.4612	0.4299	0.5640	0.4196	0.1444
67	0.	0.00004	45.00	19000.	0.50	0.7358	0.7306	0.3750	0.3742	0.0008
68	0.	0.00022	45.00	11306.	0.60	0.9127	0.9029	0.4700	0.4302	0.0398
69	0.	0.00022	90.00	11306.	0.80	1.1980	1.1634	0.4170	0.3321	0.0849
70	0.	0.00040	20.00	11306.	0.60	0.7031	0.6851	0.4310	0.3705	0.0605
71	0.	0.00004	5.00	22500.	0.20	0.3331	0.2974	0.8830	0.3934	0.4896
72	0.	0.00004	10.00	19000.	0.20	0.3691	0.2993	0.5300	0.4131	0.1169
73	0.	0.00004	10.00	26000.	0.40	0.6357	0.6116	0.6410	0.4854	0.1556
74	0.	0.00004	20.00	22500.	0.40	0.6723	0.6752	0.9390	0.9664	-0.0274
75	0.	0.00004	20.00	26000.	0.60	0.9736	0.9626	0.7390	0.7886	-0.0496
76	0.	0.00004	45.00	22500.	0.70	1.1124	1.1425	0.6390	0.7695	-0.1305
77	0.	0.00004	90.00	19000.	0.60	0.9346	0.9419	0.5540	0.5856	-0.0316
78	0.	0.00004	90.00	22500.	0.80	1.3496	1.4716	0.9630	1.6694	-0.7064
79	0.	0.00004	90.00	22500.	0.90	1.7229	1.4716	0.6160	0.7792	-0.1632
80	0.	0.00022	5.00	15000.	0.20	0.3212	0.2942	0.6880	0.3628	0.3252
81	0.	0.00022	10.00	19000.	0.60	1.0077	0.9405	0.9340	0.5795	0.3545
82	0.	0.00022	10.00	22500.	0.90	1.4645	1.4432	0.8270	0.6807	0.1463
83	0.	0.00022	90.00	11306.	0.70	1.1659	1.1634	0.6830	0.6707	0.0123
84	0.	0.00040	10.00	15000.	0.60	0.9431	0.9103	0.5940	0.4570	0.1370
85	0.	0.00004	5.00	15000.	0.10	0.1317	0.1543	1.5470	0.5180	1.0290
86	0.	0.00004	5.00	19000.	0.10	0.2545	0.2165	9.0470	4.1750	4.8720
87	0.	0.00004	5.00	26000.	0.20	0.4463	0.4075	4.6700	3.0301	1.6399
88	0.	0.00004	10.00	15000.	0.10	0.2182	0.1977	4.4170	2.5542	1.8628
89	0.	0.00004	10.00	19000.	0.1[illegible]	0.2772	0.2993	12.9780	17.5005	-4.5225
[illegible]	0.	0.0000[illegible]	10.00	26000.	[illegible]	[illegible]	[illegible]	3.55[illegible]	3.0[illegible]	0.4938
[illegible]	0.	0.00004	[illegible]	11306.	0.1	[illegible]	0.1007	1.979	1.4[illegible]	0.5011
[illegible]	0.	0.00004	2[illegible]	19000.	[illegible]	[illegible]	0.4499	4.4[illegible]	[illegible]722	-0.6522
93	0.	0.00004	[illegible]	26000.	[illegible]	0.9350	0.9[illegible]	1.64[illegible]	2.4633	-0.0245
94	0.	0.00004	[illegible]5.00	11306.	0.1	0.2176	0.25[illegible]	4.2620	9.2[illegible]	-4.9932
95	0.	0.00004	45.00	1[illegible]00.	0.40	0.6953	0.7306	1.1730	1.7[illegible]76	-0.5570
96	0.	0.00004	45.00	26000.	0.90	1.5452	1.6939	1.0800	1.2160	[illegible]
97	0.	0.00004	90.00	11306.	0.10	0.2262	0.3090	5.4415	19.6739	-14.2329
98	0.	0.00004	90.00	19000.	0.50	0.8685	0.9419	1.1650	1.9262	-0.7612
99	0.	0.00004	90.00	26000.	0.93	1.6164	2.1947	1.4400	7.4627	-6.0227
100	0.	0.00022	5.00	11306.	0.10	0.3052	0.2338	19.6590	6.1867	13.5363
101	0.	0.00022	5.00	19000.	0.20	0.4764	0.4034	6.6230	2.6539	3.9691
102	0.	0.00022	5.00	22500.	0.30	0.6121	0.5454	3.0430	1.5525	1.4905
103	0.	0.00022	10.00	11306.	0.20	0.3666	0.3399	1.6590	1.0109	0.6731
104	0.	0.00022	10.00	19000.	0.50	1.0261	0.9405	3.1400	1.9094	1.2306
105	0.	0.00022	10.00	22500.	0.80	1.5041	1.4432	1.9040	1.4804	0.4236
106	0.	0.00022	10.00	26000.	0.90	2.1360	2.1293	6.5600	6.4602	0.0998
107	0.	0.00022	20.00	11306.	0.30	0.5454	0.5329	1.5600	1.3445	0.2155
108	0.	0.00022	20.00	19000.	0.90	1.8136	1.9170	2.8400	3.8338	-0.9938
109	0.	0.00022	45.00	11306.	0.50	0.9377	0.9029	1.3400	1.4894	-0.1494
110	0.	0.00022	45.00	15000.	0.90	1.7244	1.9315	2.1290	3.9887	-1.8597
111	0.	0.00022	90.00	11306.	0.60	1.1855	1.1634	1.6860	2.2827	-0.5967
112	0.	0.00022	90.00	15000.	0.90	1.5852	2.5692	3.5080	14.5878	-11.0798
113	0.	0.00040	5.00	11306.	0.20	0.4187	0.3477	3.4970	1.1708	2.3262
114	0.	0.00040	10.00	11306.	0.30	0.5901	0.5384	2.4780	1.4332	1.0448
115	0.	0.00040	10.00	19000.	0.[illegible]	1.6287	1.5816	1.5100	1.2555	0.2545
116	0.	0.00040	20.00	11306.	0.50	0.9171	0.8851	1.6400	1.3152	0.3248
117	0.	0.00040	20.00	15000.	0.90	1.7689	1.7759	2.4670	2.5233	-0.0563
118	0.	0.00040	45.00	11306.	0.90	1.5495	1.5497	1.1000	1.1010	-0.0010
119	0.	0.00040	90.00	11306.	0.90	1.7985	2.0179	2.7100	4.9876	-2.2776

2.514681E-02 6.000000E+01 2.047226E-02

Figure 3-9 permits a quick evaluation of the quality of the functionalization of I_L. The abscissa scales represent the computed lethality integral, I_{LC}, (on logarithmic scale), and its corresponding computed survival ratio f_c given by Equation 3-46. The ordinate scales represent the residuals ΔI_L given by Equation 3-47, in linear scale, and the corresponding ratio $\frac{f}{fc}$ of true survival ratio to computed survival ratio given by Equation 3-48.

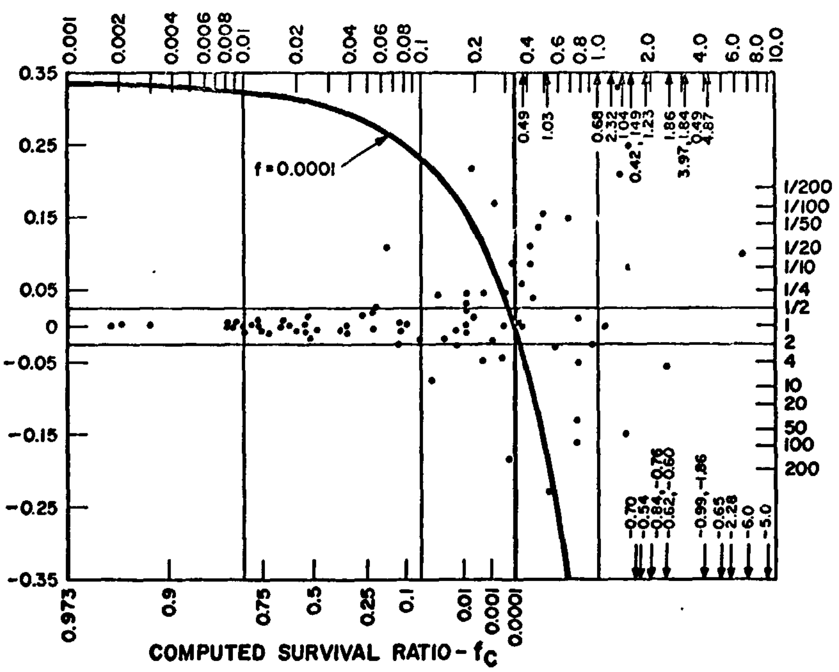


Figure 3-9. Evaluation of Functionalization

The locus of the points corresponding to f_c equal to 0.0001 is the vertical straight line. The locus of the points corresponding to f equal to 0.0001 is the curved line. For all the points which are simultaneously to the right of these two lines, the true kill as well as the computed kill are complete.

The two horizontal lines correspond to ratio $\frac{f}{f_c}$ equal to $\frac{1}{2}$ and 2. These lines define the bounds for all the points for which the true survival ratio f is within a factor 2 of the computed survival ratio f_c.

It can be readily seen from Figure 3-9 that most of the data points having survival ratio larger than 10^{-4} are within this band, that is, within this factor of 2 of the computed survival ratio. The few points which are not within the band would, however, almost all fall within a band having a factor of 4 from the computed survival ratio (f_c).

Since this was only a preliminary investigation, no special investigation was made of the few data points which were not within this band defined by a maximum factor of 4 between f_c and f.

3.14 CHANGES IN PARAMETERS

This preliminary investigation was conducted mainly to determine a procedure for functionalizing the lethality integral I_L. Since then, the following changes in the values or ranges of some parameters have appeared desirable:

a. The range of ballistic coefficient, β, should cover from 4×10^{-6} to 4×10^{-3} slug/ft^2.

b. The entry altitude, h_o, should be much higher than the 721,000 feet we have used. A value of 2,000,000 feet seems to be satisfactory for the range of ballistic coefficients considered in item a, above.

c. The range of entry angles, γ_E, should be extended. Viability of particles entering Mars atmosphere at a very small angle is of concern for the quarantine study.

d. The solar absorptivity, α, should be varied to simulate nighttime as well as daytime entry into Mars atmosphere and to take into account experimental values of α which are much smaller than 1.

e. While an initial (equilibrium) temperature of 500°R is reasonable for daytime entry, a lower initial temperature (183°R, for instance) should be used for nighttime entry.

The influence of these parameters and of these changes in parameter ranges is being investigated.

3.15 ORGANISMS CARRIED BY NONVIABLE PARTICLES

If we assume that, during all the entry trajectory, the microorganism attains the same temperature as the nonviable particle which carries it, the integral (I_L) can be computed for the particulate carrier (with its own solar absorptivity and its own emissivity) and inferences drawn or implied as to its effect on the microorganism.

Results are summarized in Table 3-4.

Table 3-4. I_L for Nonviable Particles in Full Sun

Material	β → / ↓ V_E	$\gamma_E = 5^o$		$\gamma_E = 45^o$		$\gamma_E = 90^o$	
		4×10^{-5}	4×10^{-4}	4×10^{-5}	4×10^{-4}	4×10^{-5}	4×10^{-4}
Aluminum	11 306	0.0006	14.11	0.0111	29.80	0.0438	26.28
	15 000	0.0012		2.034		2.856	
Fuzed Silica	11 306	0.0000	36.94	0.0315	42.16	0.1824	47.67
	15 000	0.0014		4.409		5.448	
Haynes-25	11 306	0.0000	9.49	0.0986	27.14	0.5332	26.9
	15 000	0.0010		3.128		4.047	
Magnesium	11 306	0.0001	28.54	0.0147	36.82	0.0518	38.89
	15 000	0.0022		2.791		3.651	
Epoxy Glass	11 306	5.46	125.20	5.71	68.41	4.66	75.86
	15 000						

For comparing Table 3-4 and Table 3-2, we can consider an equivalent emissivity, ϵQ, giving in Table 3-2 the same I_L as in Table 3-4 for the same combination (β, γ_E, V_E).

It can be seen that an equivalent emissivity, ϵQ, (between 0.1 and 0.3) can be defined for each of the first four materials. The equivalent emissivity increases with the entry angle γ_E: the materials seem to be less sensitive to the change in entry angle than the microorganism.

The conclusion would then be that any organism carried by the epoxy glass would receive a heat treatment which would kill it.

110

3.16 REFERENCES

3-1 Martin, M. A., "Voyager Mars Planetary Quarantine: Thermal Kill of Bacteria During Mars Entry," GE (MSD) Report VOY-C2-TM2, October 1966.

3-2 Ball, C. O., and Olson, F. C. W., "Sterilization in Food Industry, "McGraw-Hill Book Company, New York, N. Y., 1957, page 292.

3-3 Bruch, C. W., Koesterer, M. G., and Bruch, M. K., "Dry-Heat Sterilization: Its Development and Application to Components of Exobiological Space Probes," Developments in Industrial Microbiology, American Institute of Biological Sciences, Washington, D. C., 1963, Volume 4, pages 334-342.

3-4 Hobby, G. L., "Review of the NASA-JPL Spacecraft Sterilization Program," Appendix III of Chapter 10, In a Review of Space Research, Publication No. 1079 of the National Academy of Sciences, National Research Council, Washington, D.C., pages 10-34.

3-5 Koesterer, M. G., "Thermal Death Studies on Microbial Spores and Some Considerations for the Sterilization of Spacecraft Components, "Developments in Industrial Microbiology, American Institute of Biological Sciences, Washington, D.C., 1964, Volume 6, pages 268-276.

3-6 Decker, H. M., Citek, F. J., Harstad, J. B., Gross, N. H., and Piper, F. J., 1954, "Time Temperature Studies of Spore Penetration Through an Electric Air Sterilizer," Applied Microbiology. 2(1):33-36.

3-7 Koesterer, M., "Temperature Effects," Section 3, Task 2 - Planetary Quarantine, Second Bi-Monthly Report, pages 3-14.

3-8 Parker, J., Beerger, N., and Burrows, G., Voyager Mars Quarantine - Particle Burnup, GE (MSD) Report VOY-C2-TR4, 15 September 1966.

3.17 ACKNOWLEDGEMENTS

It is indeed a pleasure to acknowledge the contributions of the following persons to this investigation: J. Parker, who, with D. Florence, provided background information on his previous work, and by his knowledge helped considerably in the selection of the parameters, and, with N. Beerger, provided all the necessary computer runs as well as data on nonviable particles; C. Bursey, who made all the trajectory runs needed and, with D. Korenstein, pointed out the aerodynamic considerations which had to be taken into account; and M. Koesterer, who established the curve of thermal resistance of dry spores and explained how the curve was established. My thanks go also to Miss P. McManus who computed the lethality integral on the Desk Side Computer before P. Friend added an ad hoc subroutine to his thermodynamics computer program.

SECTION 4

SENSITIVITY STUDIES

by

E. Berger
R. Wolfson

112

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

SECTION 4

SENSITIVITY STUDIES

4.1 INTRODUCTION

The sensitivity studies involve the exercising of the basic math model with inputs from the basic studies as specifically applied to the Voyager mission and hardware. The initial cases shown are primarily designed to illustrate the process for working through the analysis and showing how the sensitivity of the contamination probability varies with different input parameters. In many cases the input parameters used have been essentially educated guesses. The continuing work in this area will use better and better input data, but these early studies serve to illustrate the areas which are important and require more careful analysis as the study proceeds.

Figure 4-1 is a simplified work flow diagram and illustrates how the sensitivity studies are related to the Quarantine Task.

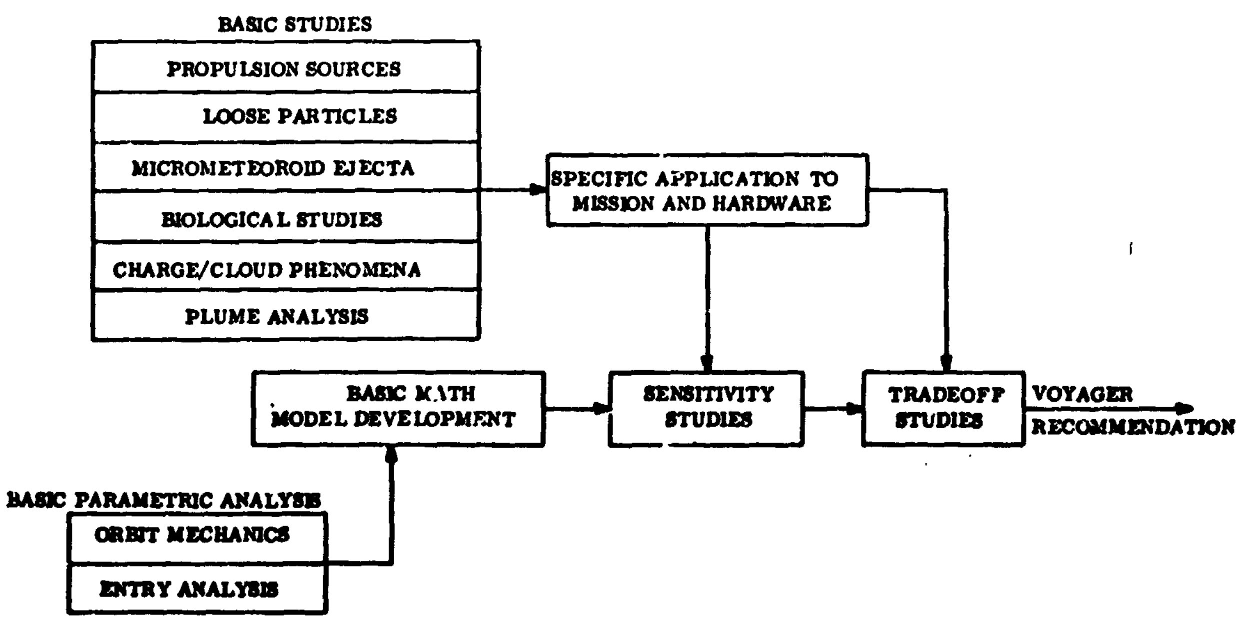


Figure 4-1. Planetary Quarantine Task Simplified Work Flow Diagram

The math model format, as illustrated in Figure 4-1, shows how the various sources of contamination are to be analyzed. The basic kill mechanisms are associated with columns. Each column either requires the development of input or operation on a particular portion of the math model.

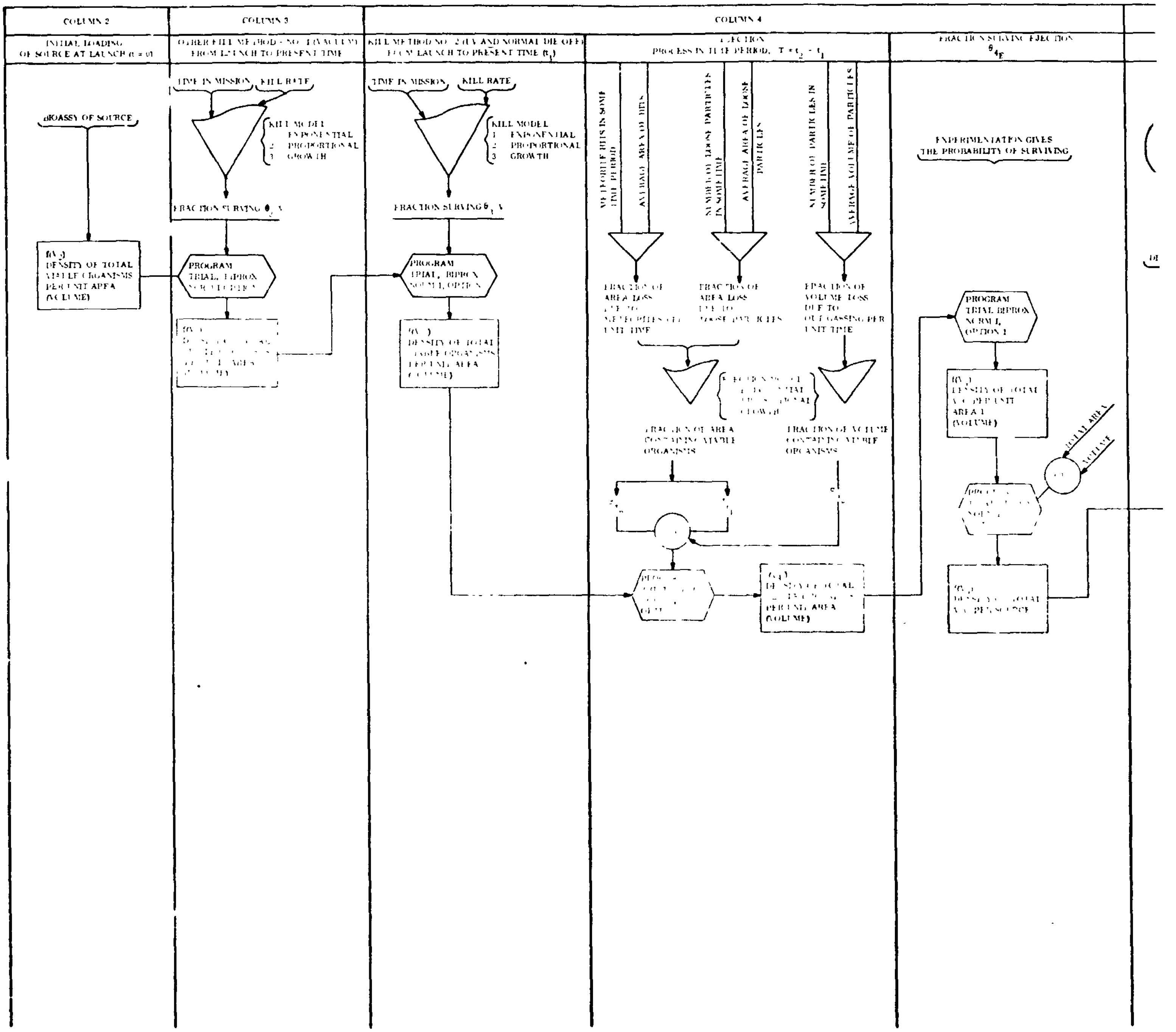
Figure 4-2 is a detailed computational flow diagram. This flow is for loose particles, micrometeoroid ejecta, and gaseous emissions, either cold or hot. In this figure rectangles give the output from or input to each column, and six-sided boxes tell what computer program is to be used.

Figure 4-3 illustrates how each of the contamination sources will be combined to give the probabilities of viable organisms reaching and growing on the planet.

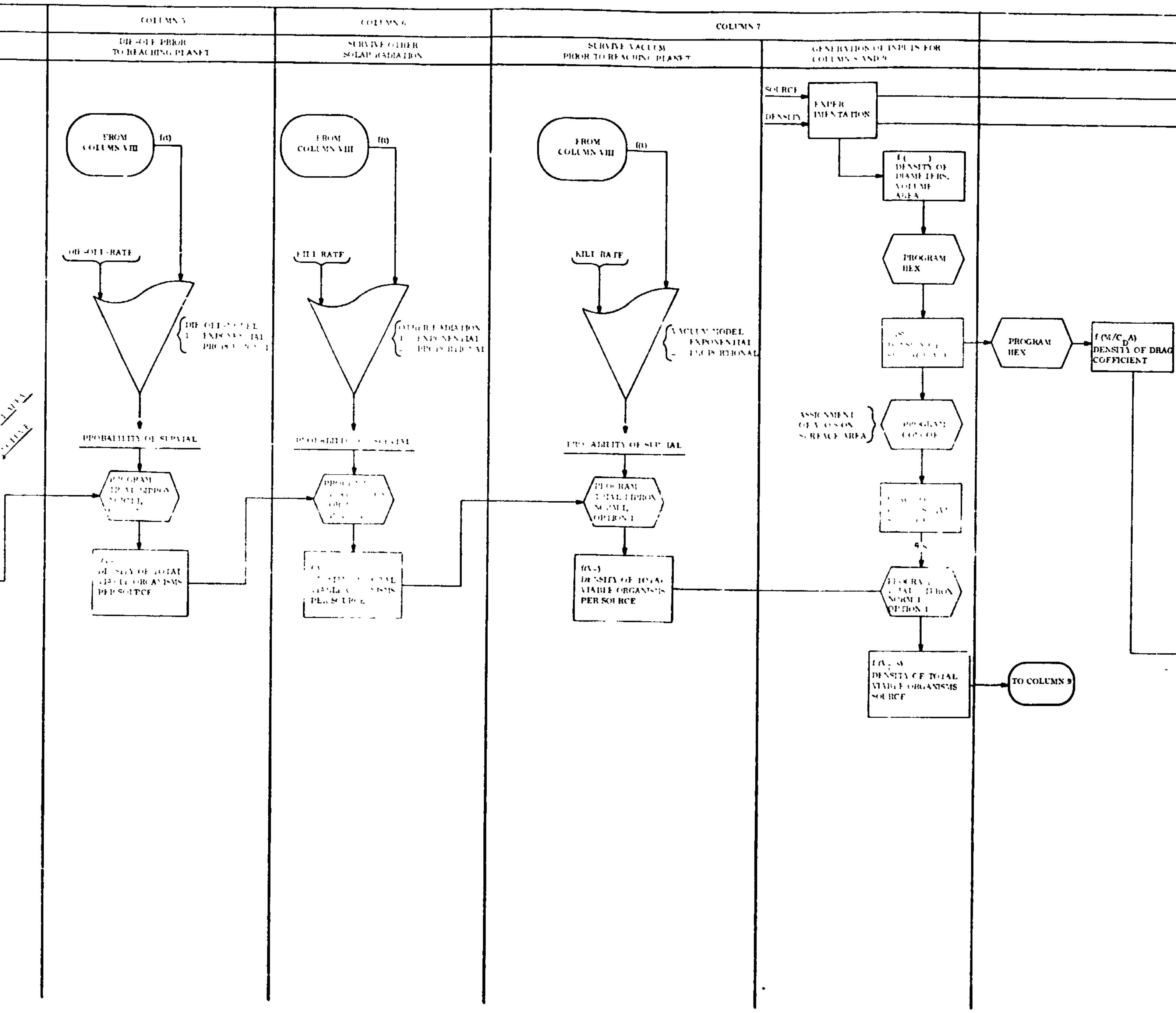
A few sources have been partially evaluated in a preliminary manner. None of the cases studied include all of the kill methods. The assumptions used will be stated with the results. Caution should be used in generalizing from the results presented. The assumptions must always be kept in mind.

4.2 LOOSE PARTICLES

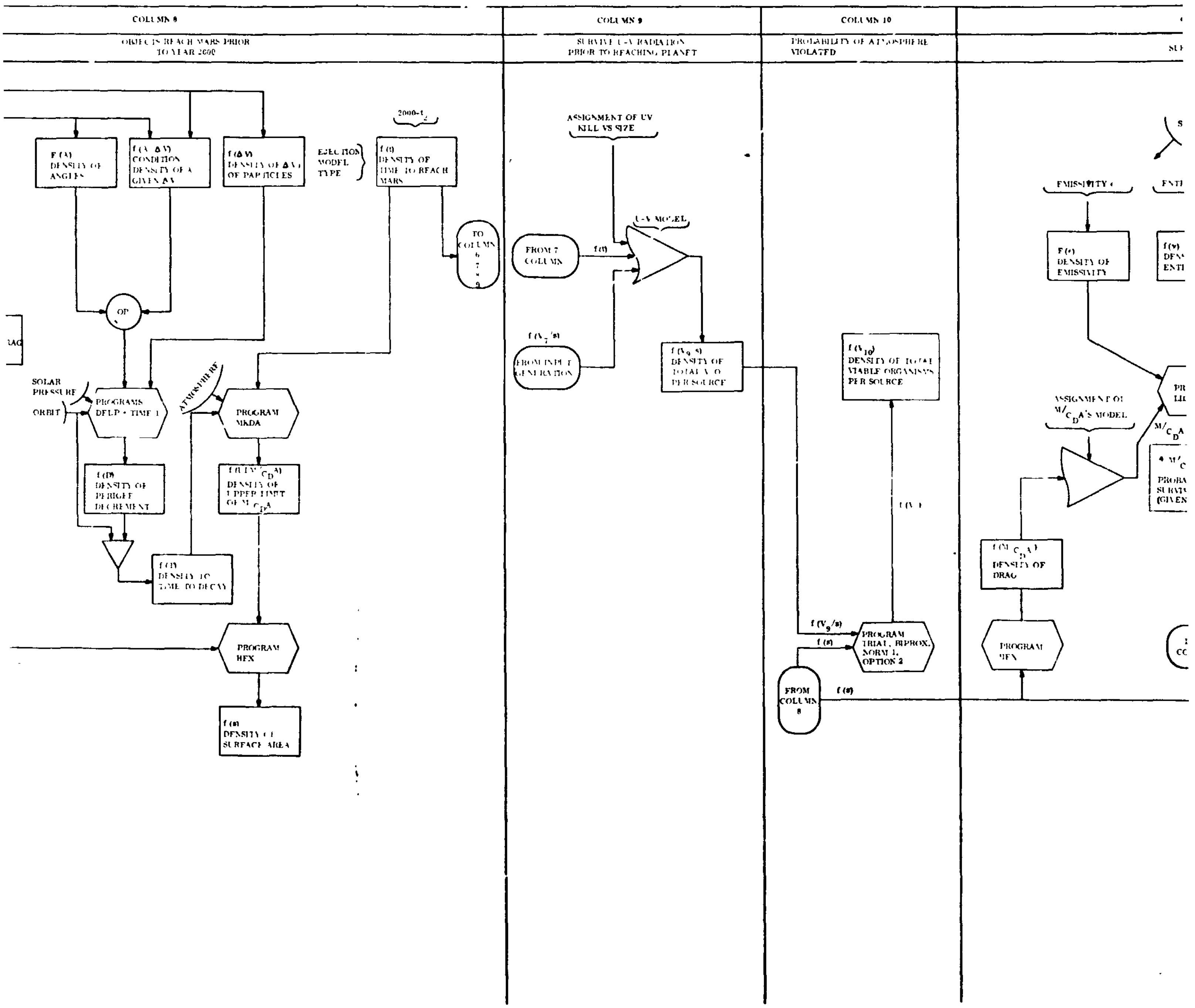
Preliminary results on loose particles are given in this section. The initial loading on the spacecraft was assumed as shown as curve (A) in Figure 4-4. Figure 4-4 shows the cumulative probability distribution function, whereas most of the other figures in this section are probability density functions. From basic data, VOY-C2-TM3, the total number of loose particles was estimated; then, with distribution of sizes, obtained by the same experimental investigation and modified to account for the lack of data below 150 microns, an estimate was made of the fraction of the total viable organisms on the particles and was found to be 0.001. This then gave the distribution labled (B) in Figure 4-4 viable organisms on loose particles.



Foldout Frame 1



Fold out FRAME 2



Foldout Frame 3

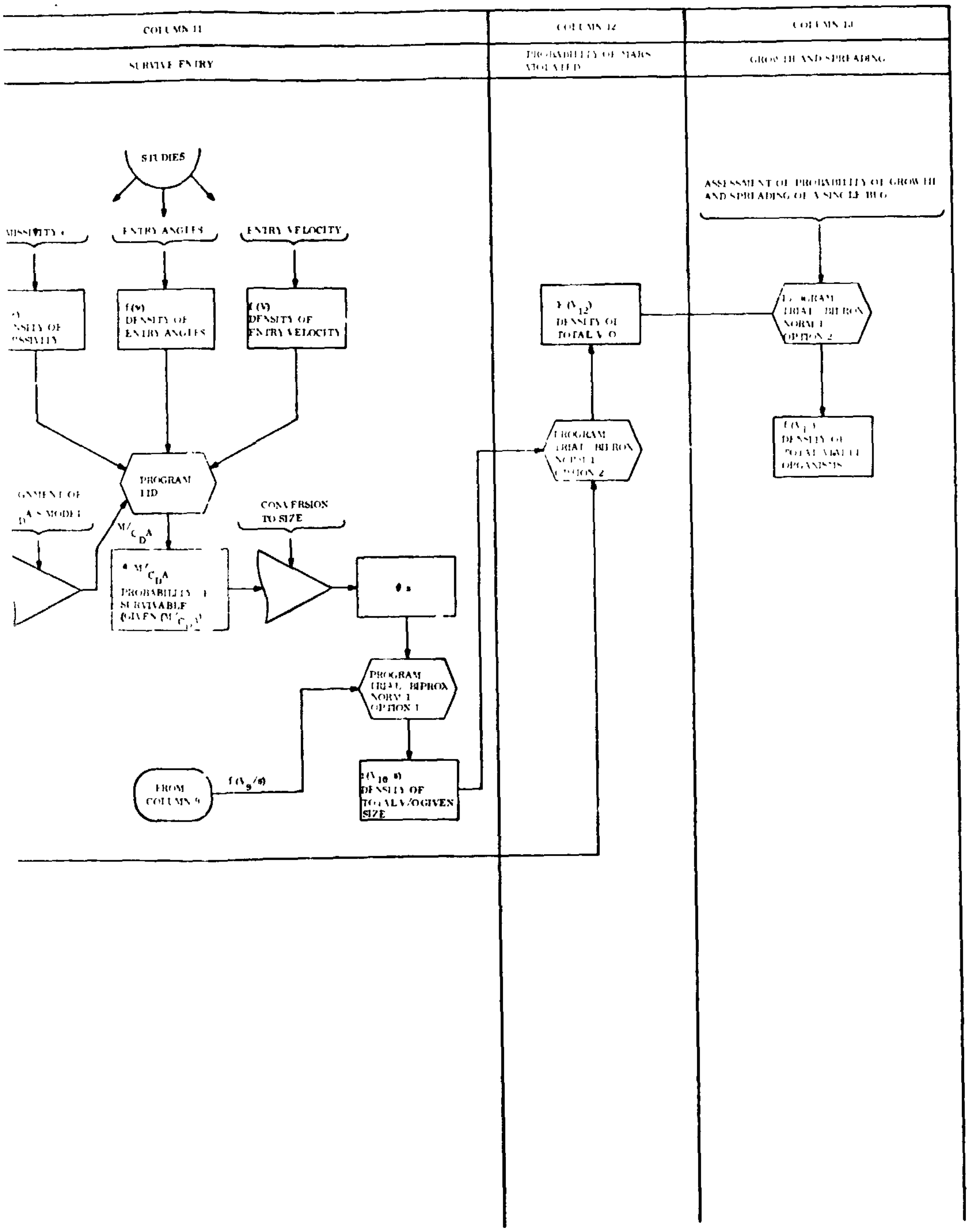


Figure 4-2. Detailed Computational Flow Diagram

Foldout FRAME 4

Figure 4-5 gives the distribution of particle diameters used in the loose particle investigation. An average weight per cubic foot of 68.6 was then assigned for the loose particles. Then, assuming a spherical relationship between diameter and surface area, the probability density

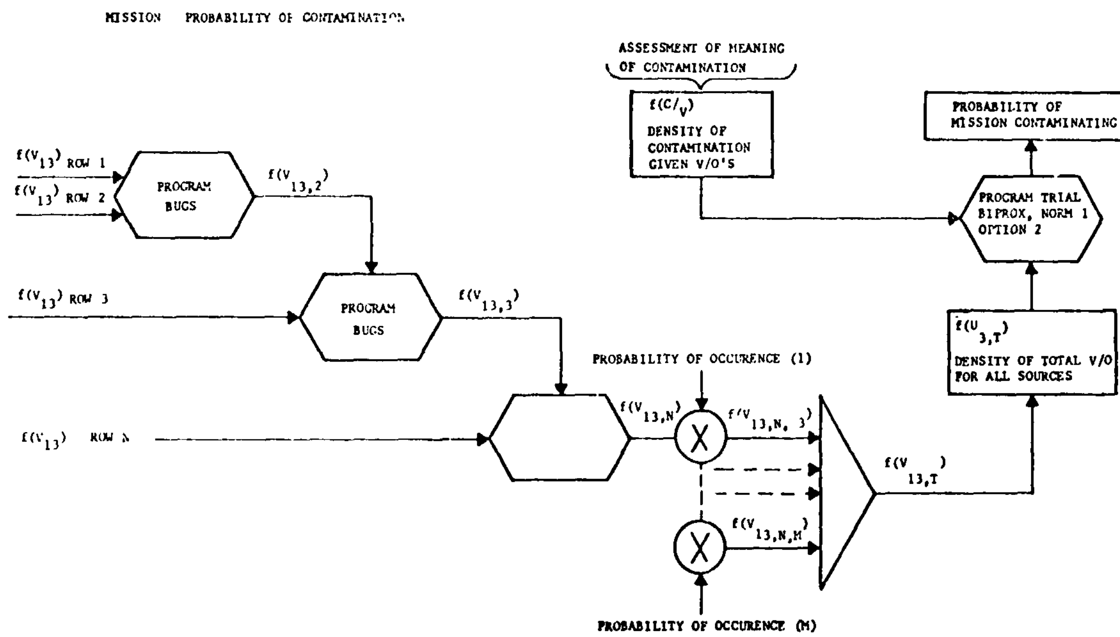


Figure 4-3. Combination of Contamination Sources

function of surface area was obtained by the use of program "HEX." Then, assuming that viable organisms were distributed in direct proportion to the surface area of the particle and that a loose particle below π square microns would not carry an organism (since particles below this size approach the size of a microorganism), we obtained the fraction of viable organisms on each range of surface area. This is an important step because the ballistic characteristic of each range of surface area is different and, to obtain a good estimate of the viable organisms surviving, these ballistic effects must be considered.

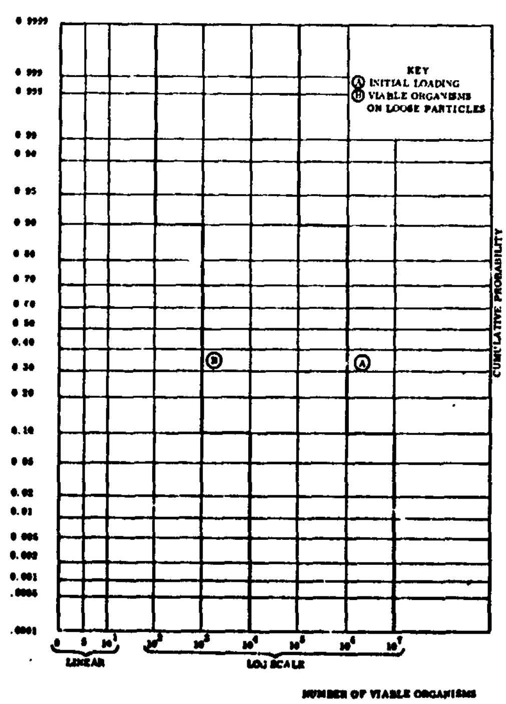


Figure 4-4. Cumulative Probability of Viable Organisms

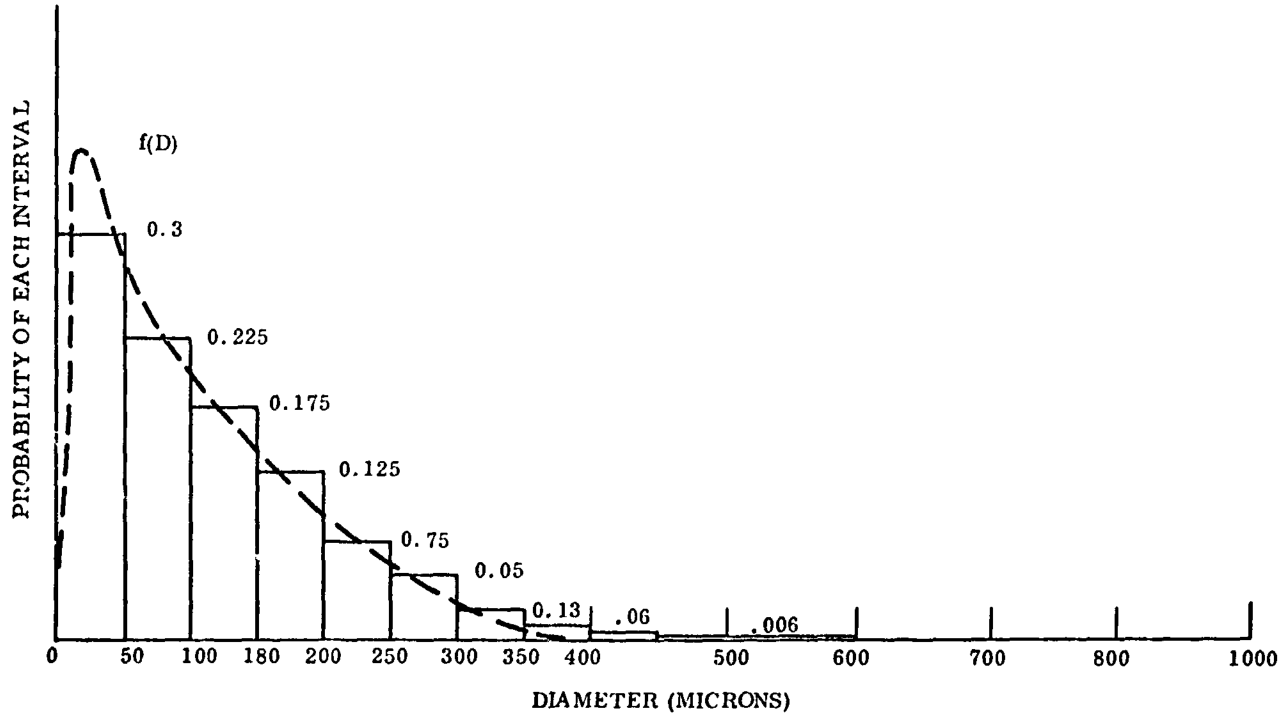


Figure 4-5. Probability Density Function of Loose Particles

Figure 4-6 gives the fraction of viable organisms on each surface area range considered. This assignment is accomplished with a program called "CONCOF." With the fraction of viable organisms assigned to each range of surface area and using the distribution of total viable organisms assigned to loose particles, a distribution of total viable organisms on each size is obtained.

The next step in the analysis of loose particles is to investigate both ejection prior to orbit insertion and ejection during the Mars orbiting phase. The analysis is only shown for those particles ejected during orbit. The number of loose particles carrying viable organisms has not been decreased by those leaving prior to reaching orbit insertion; this effect is subject to the micrometeoroid environment and will be investigated later. For now it is assumed that all locse particles come off in orbit and at the first apoapsis after insertion so that they will have a favorable time to decay from orbit to the planet.

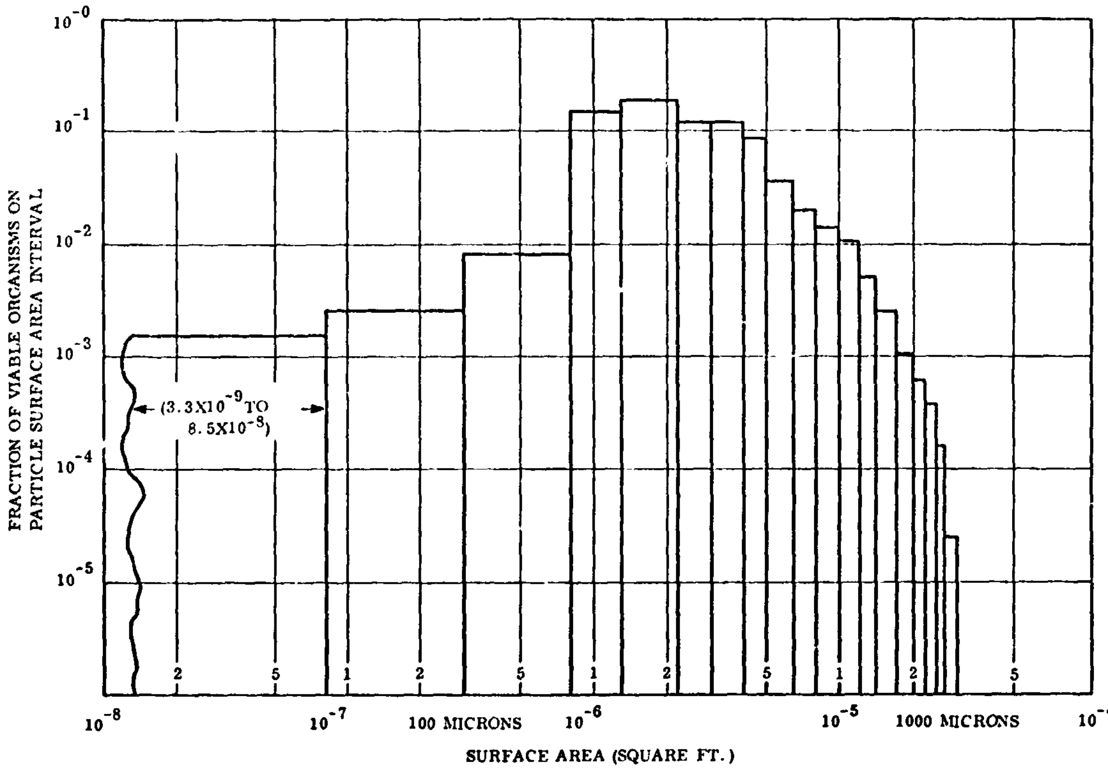


Figure 4-6. Fraction of Viable Organisms on Particles of Different Sizes

Under these assumptions, which are conservative, the analysis of orbit mechanics was performed for loose particles. Six orbits and two atmospheres were investigated.

Table 4-1 is a summary of the orbits and atmospheres investigated. All of the loose particles are assumed ejected at apoapsis which gives the largest decrease in periapsis altitude. The atmospheres are indicative of the expected variation in the VM-3 atmosphere as a function of solar heating of the planet. Atmosphere 1 is the night model; atmosphere 2 is the daylight model. Both models are less dense than the recent atmosphere adopted by JPL; consequently, more conservative from the quarantine viewpoint.

Table 4-1. Orbits and Atmospheres Investigated

Orbit		Atmosphere
1. 1000 x 10, 000 km		VM-3 Atmos. Extended by Vachon
2. 500 x 10, 000 km	1	for 0400 hours (min density)
3. 200 x 10, 000 km	2	for 1400 hours (max density)
4. 1000 x 20, 000 km		
5. 500 x 20, 000 km		
6. 200 x 10, 000 km		

Various ejection velocities and angles of leaving the spacecraft were assumed. The angles were assumed uniform over 4 π steradians, and the velocity increment was that shown in Figure 4-7. These velocities are to be representative of loose particles drifting off of the spacecraft.

Programs DELP1, DELP2, and TIME1 were run to obtain the distribution of time to entry. Figure 4-8 gives the distribution for the six orbits and two atmospheres. Since atmosphere 1 is more conservative than the official JPL atmosphere, it is the one which was used in the investigation. Notice, however, that both orbit and atmosphere have significant effects. Notice also that only one-half of the loose particles have a chance of gettin; to the planet no matter what the orbit or atmosphere. This effect is due to the assumption of unifrom angles of ejection; that is, one half of the particles leave at the wrong angles. This effect may be removed when solar pressure is included in the analysis. From Figure 4-8a it is seen that the probability of a particle reaching the planet prior to 30 years is 0.00121, for orbit type 1 and atmosphere 1.

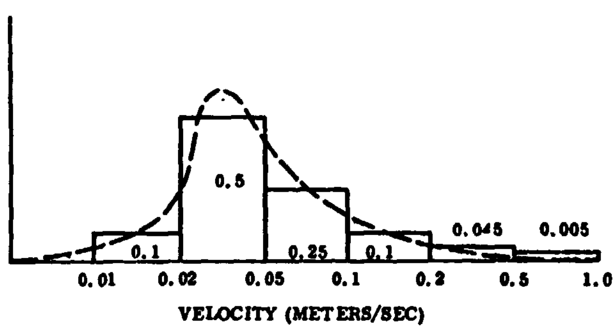


Figure 4-7. Ejection Velocities

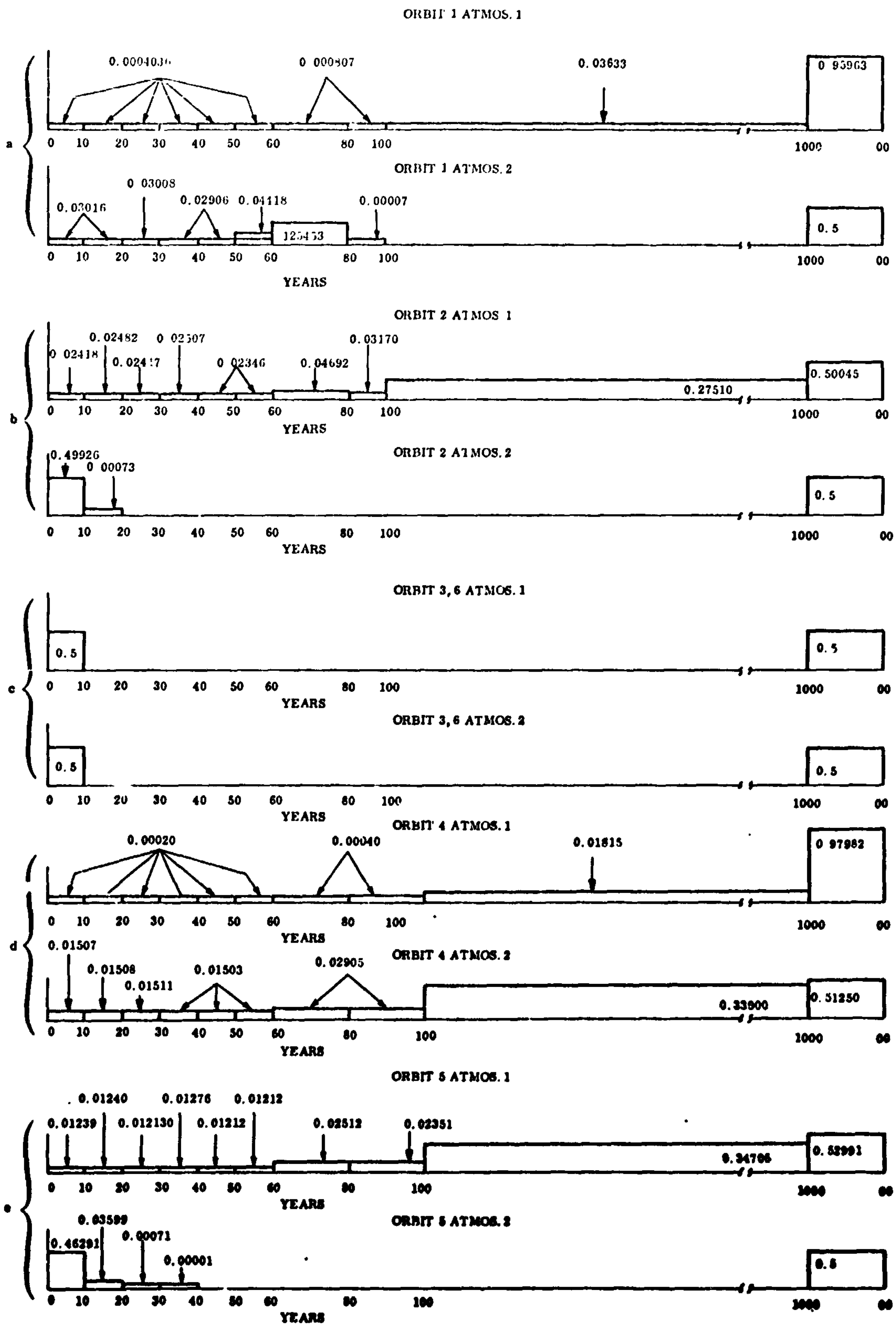


Figure 4-8. Probability of Entry for Various Time Intervals

121

Based on orbit 1 and atmosphere 1, programs M/C_DA and MCED were used to obtain the distribution of surviving sizes which enter within 30 years; this distribution was then used as an input to Program LID, which calculates the probability of surviving entry heating. The probability calculated was 0.1722.

The probabilities of surviving entry heating and entering prior to 30 years are then used on the distribution of total viable organisms on each size. The effect of U.V. kill and die-off are also assigned. The next result is then obtained by combining each size with its probability of occurrence. Figure 4-9, shown as a cumulative distribution function as was Figure 4-4, gives the preliminary estimate of this process using a probability of entering 0.00121, a probability of surviving heating of 0.1722, a probability of surviving U.V. of 0.1, and a probability of growth and spreading of 0.01.

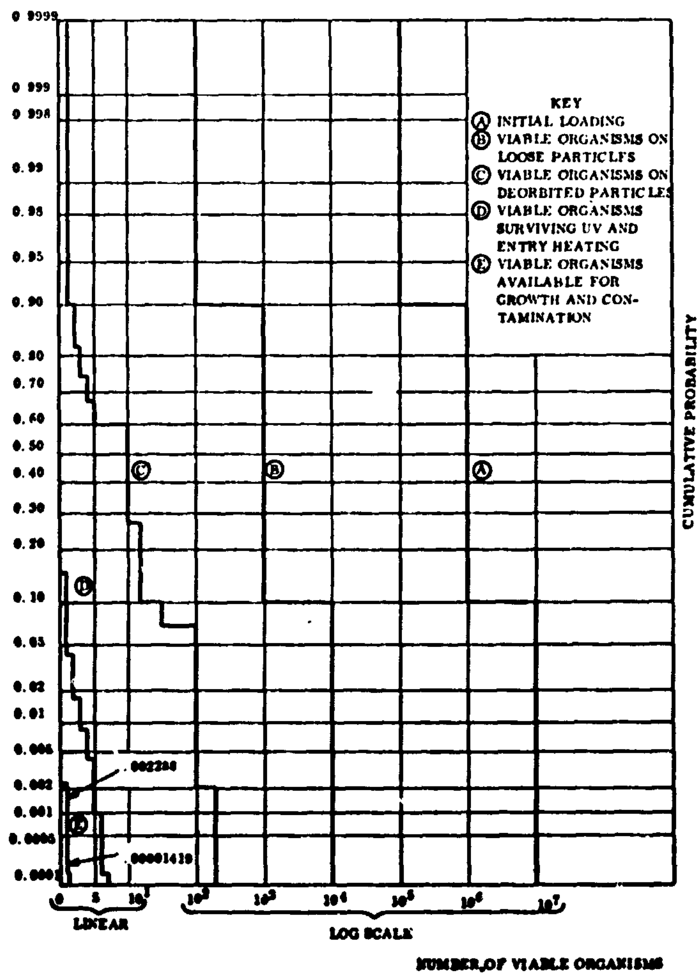


Figure 4-9. Cumulative Probability of Viable Organisms

When the final analysis is conducted, U.V. kill will be entered as a function of particle size If the particle is 10 microns or smaller, kill will be assigned so that the viable organisms will have a high probability of being killed. The estimate of 0.1 is a "guesstimate" of the effects of U.V. The 0.01 growth and spreading probability is based on a recommendation presented to the last COSPAR meeting by Dr. C.W. Craven and J.O. Light of JPL.

The work in this section should be considered only as an example of the computational procedures and the way in which the quarantine problem can be studied. The actual data should not be used since the input data in many areas were guesstimates, and, in particular, the range of M/C_DA values under consideration is currently being revised.

4.3 ATTITUDE CONTROL GAS

Figure 4-10 gives the initial number of viable organisms in the attitude control gas system. This initial number of viable organisms is assumed ejected in proportion to the usage rate of the attitude control gas as illustrated in Figure 4-11.

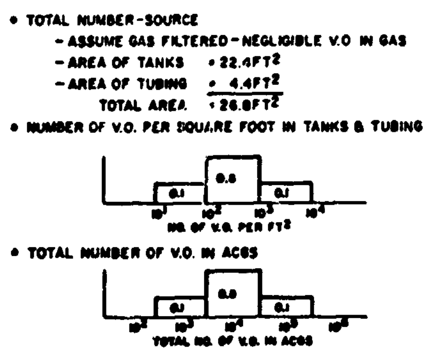


Figure 4-10. Initial Number of Viable Organisms in Attitude Control Gas System

Figures 4-12 and 4-13 give the parameters associated with the size distribution, drag parameter (M/C_DA), velocity, and ejection angles.

An analysis was conducted for a period of 1 day, 5 days before heliocentric encounter. An aim point based on the GE Task B study and a type I trajectory was assumed. The resulting probability of being on an impact trajectory was obtained as 0.00153. Figures 4-14 and 4-15 illustrate the results of applying first the fraction ejected and then the probability of being on an impact trajectory.

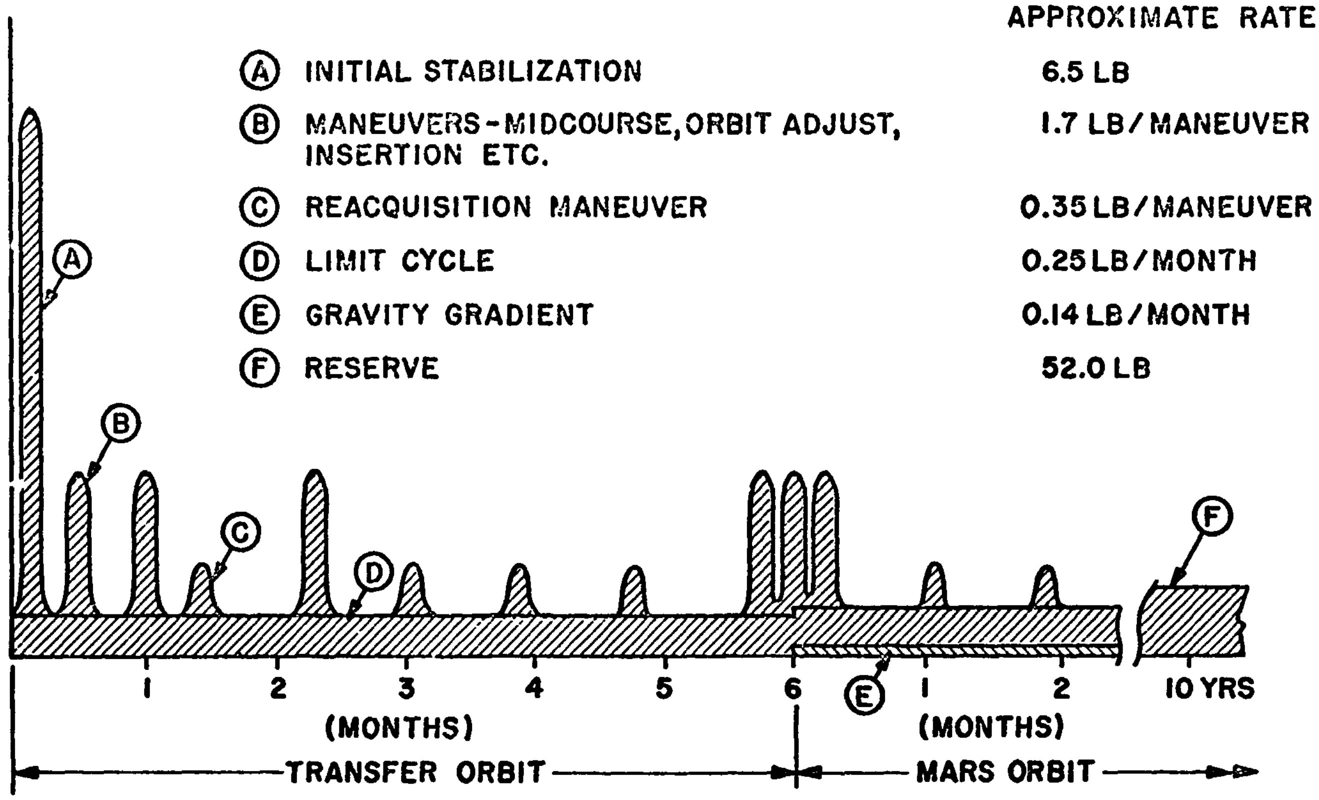


Figure 4-11. Attitude Control Gas Use Profile

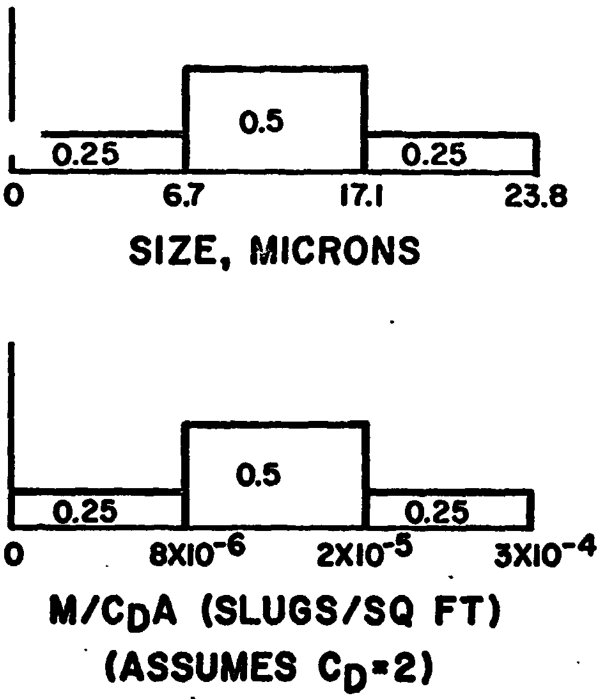


Figure 4-12. Size Distribution

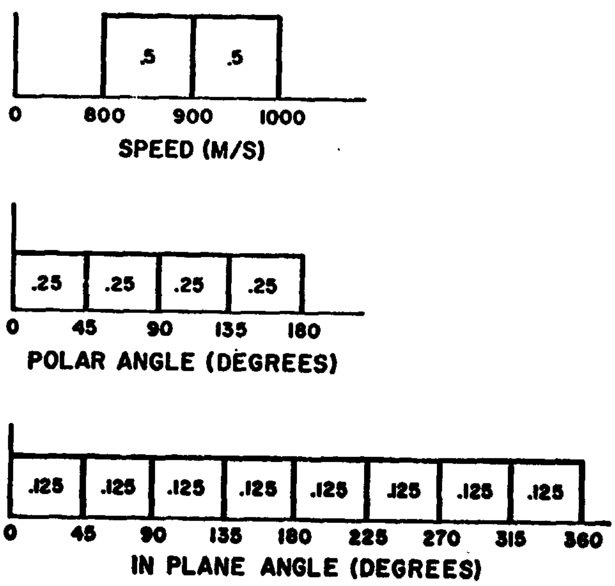


Figure 4-13. Velocity Distribution

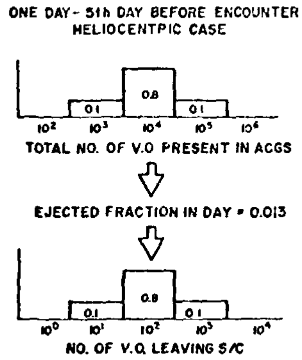


Figure 4-14. Start Analysis

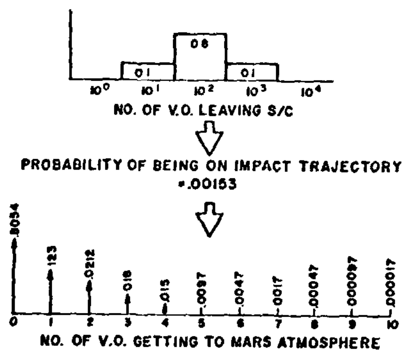


Figure 4-15. Analysis Continued

Figure 4-16 and 4-17 illustrate the results of the viable organisms surviving entry, U.V. kill and die-off (assumed to be 0.1), and growth and contamination (assumed to be 0.01).

Newly revised estimates of the M/C_DA range required use of the entry heating program outside its original design range for this study. The range of accuracy of this program is currently being extended.

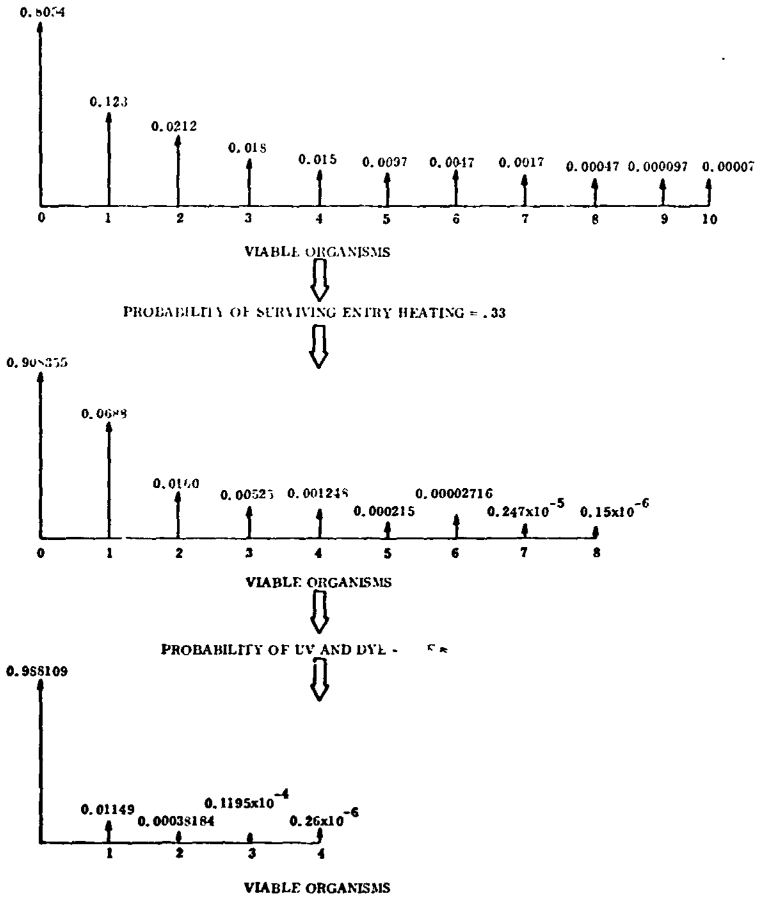


Figure 4-16. Analysis Continued

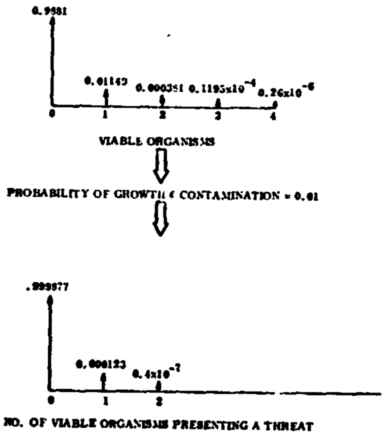


Figure 4-17. Analysis Continued

4.4 ACKNOWLEDGEMENTS

This is to acknowledge the assistance of Mr. D. Higley in operating the Desk Side Computer System and in reducing the computer output data for these sensitivity studies.

APPENDIX A

PROGRAM SOURCE LISTING

WAIT.

READY.

$LIST

```
C    *****BASIC COMBINATORIAL MODEL FOR VOYAGER********
        COMMON GIN(100),PRO(100),G(100),GP(100),POUT(100),GTP(100)
  1000 FORMAT("1   BASIC COMBINATORIAL MODEL")
  1020 FORMAT("0 NUMBER, ROW END POINTS")
  1040 FORMAT(" ROW PROBABILITIES")
  1060 FORMAT(" NUMBER, STANDARD END POINTS")
  1080 FORMAT("1  READ IN CODE(1-5)")
  2000 FORMAT("0          FIRST          LAST          INPUT"
      +"       OUTPUT")
  2020 FORMAT(" SIMPLE PROBABILITY")
  2040 FORMAT(" SCALE FACTOR")
  2060 FORMAT(" NUMBER, END POINTS FOR (",1PE14.5,")")
  2080 FORMAT(" CONDITIONAL PROBABILITIES")
  3000 FORMAT(" PROPORTION, PROBABILITY FOR (",1PE14.5,")")

        PRINT 1000
      1 PRINT 1020
        READ:N1,(GIN(I),I=1,N1)
        N1M1=N1-1
        PRINT 1040
        READ:(PRO(I),I=1,N1M1)
        PRINT 1060
        READ:NST,(G(I),I=1,NST)
        NTM1=NST-1
        DO 20 I=1,NTM1
     20 GP(I)=0.
        DO 30 I=1,N1M1
        SP=GIN(I+1)-GIN(I)
     30 CALL LOAD(SP,PRO(I),GIN(I),GIN(I+1),NST,G,GP)
        TEST=100.
     60 DO 70 I=1,NTM1
     70 POUT(I)=0.
        PRINT 1080
        READ:ICODE
        GO TO (100,200,300,400,1),ICODE
    100 PRINT 2020
        READ:TR
        DO 150 I=2,NST
        IF(TEST-G(I))145,140,140
    140 NS=G(I)
        CALL BIACCU(NS,TR,G,GP(I-1),POUT,NST)
        GO TO 150
    145 NG=I
        CALL BCONT(TR,G,POUT,NG,GP(I-1))
    150 CONTINUE
        GO TO 600
    200 DO 250 I=2,NST
        DO 210 J=1,NTM1
      1 GTP(I)...
```

```
      CALL BCONT(TR,G,POUT,NG,GP(I-1))
  150 CONTINUE
      GO TO 600
  200 DO 250 I=2,NST
      DO 210 J=1,NTM1
  210 GTP(J)=0.
      PRINT 2060,G(I)
      READ:N2,(GIN(IZ),IZ=1,N2)
      N2M1=N2-1
      PRINT 2080
      READ:(PRO(IZ),IZ=1,N2M1)
      DO 220 K=1,N2M1
      SP=GIN(K+1)-GIN(K)
  220 CALL LOAD(SP,PRO(K),GIN(K),GIN(K+1),NST,G,GTP)
      DO 230 L=1,NTM1
  230 POUT(L)=POUT(L)+GTP(L)*GP(I-1)
  250 CONTINUE
      GO TO 600
  300 DO 350 I=2,NST
      DO 310 J=1,NTM1
  310 GTP(J)=0.
      PRINT 3000,G(I)
      READ:THET,PSI
      N2=3
      GIN(1)=G(1)
      GIN(2)=G(I)*THET
      GIN(3)=G(I)
      PRO(1)=PSI
      PRO(2)=1.-PSI
      DO 320 K=1,2
      SP=GIN(K+1)-GIN(K)
  320 CALL LOAD(SP,PRO(K),GIN(K),GIN(K+1),NST,G,GTP)
      DO 330 L=1,NTM1
  330 POUT(L)=POUT(L)+GTP(L)*GP(I-1)
  350 CONTINUE
      GO TO 600
  400 PRINT 2040
      READ:SCALE
      DO 410 I=1,NST
      G(I)=SCALE*G(I)
  410 CONTINUE
      DO 500 I=1,NTM1
      POUT(I)=GP(I)
  500 GP(I)=0.
      GO TO 600
  600 PRINT 2000
      DO 620 I=1,NTM1
      PRINT:G(I),G(I+1),GP(I),POUT(I)
  620 GP(I)=POUT(I)
      GO TO 60
      END
```

$LIST

```
C     ******SEGMENT   BASIC2***********
      SUBROUTINE BIACCU(NS,TR,G,PROB,GQ,NST)
      DIMENSION G(1),GQ(1)
      FN=NS
      X=0.
      RAT=TR/(1.-TR)
      PR=(1.-TR)**NS
      GO TO 18
    5 IF(X-FN)10,30,30
   10 PR=PR*(FN-X)*RAT/(X+1.)
      X=X+1.
   18 DO 25 I=2,NST
      IF(G(I)-X)25,19,19
   19 IF(G(I-1)-X)20,20,25
   20 GQ(I-1)=GQ(I-1)+PR*PROB
      GO TO 5
   25 CONTINUE
      GO TO 5
   30 RETURN
      END
      SUBROUTINE LOAD(SP,TPR,ZL,ZH,M,Z,RP)
      DIMENSION Z(1),RP(1)
      IF(SP)100,80,20
   20 DO 30 JL=2,M
      IF(Z(JL)-ZL)30,30,25
   25 ZLOW=ZL
      JTEMP=JL
      GO TO 40
   30 CONTINUE
   40 CONTINUE
      DO 60 J=JTEMP,M
      IF(Z(J)-ZH)50,45,45
   45 JSAVE=J
      ZHIGH=ZH
      GO TO 70
   50 ZHIGH=Z(J)
      RP(J-1)=RP(J-1)+((ZHIGH-ZLOW)/SP)*TPR
      ZLOW=Z(J)
   60 CONTINUE
   70 RP(JSAVE-1)=RP(JSAVE-1)+((ZHIGH-ZLOW)/SP)*TPR
      GO TO 100
   80 DO 90 J=2,M
      IF(ZL-Z(J-1))85,82,85
   82 RP(J-1)=RP(J-1)+TPR
      JSAVE=J
      GO TO 100
   85 IF(ZL-Z(J))82,90,90
   90 CONTINUE
      RP(M-1)=RP(M-1)+TPR
      JSAVE=M
  100 CONTINUE
      RETURN
      END
      SUBROUTINE BCONT(TR,G,GQ,NG,PROB)
      DIMENSION G(1),GQ(1),ANS(100)
      XMU=TR*G(NG)
      SIGMA=SQRTF(XMU*(1.-TR))
      IF(NG-2)50,50,130
   50 G1=(G(1)-XMU-.5)/SIGMA
      G2=(G(2)-XMU+.5)/SIGMA
      ANS(1)=HAD(G1,G2)
```

```
      END
      SUBROUTINE BCONT(TR,G,G0,NG,PROB)
      DIMENSION G(1),G0(1),ANS(100)
      XMU=TR*G(NG)
      SIGMA=SQRTF(XMU*(1.-TR))
      IF(NG-2)50,50,130
   50 G1=(G(1)-XMU-.5)/SIGMA
      G2=(G(2)-XMU+.5)/SIGMA
      ANS(1)=HAD(G1,G2)
      GO TO 190
  130 DO 180 I=2,NG
      IF(I-2)150,150,155
  150 G1=(G(I-1)-XMU-.5)/SIGMA
      G2=(G(I)-XMU)/SIGMA
      GO TO 170
  155 IF(I-NG)165,160,160
  160 G1=(G(I-1)-XMU)/SIGMA
      G2=(G(I)-XMU+.5)/SIGMA
      GO TO 170
  165 G1=(G(I-1)-XMU)/SIGMA
      G2=(G(I)-XMU)/SIGMA
  170 ANS(I-1)=HAD(G1,G2)
  180 CONTINUE
  190 TMP=0.
      NGM1=NG-1
      DO 200 I=1,NGM1
  200 TMP=TMP+ANS(I)
      DO 220 I=1,NGM1
  220 G0(I)=G0(I)+PROB*ANS(I)/TMP
      RETURN
      END
      FUNCTION HAD(X1,X2)
      IF(ABSF(X1)-13.) 20,20,10
   10 X1 = SIGNF(13.0,X1)
   20 IF(ABSF(X2)-13.) 40,40,30
   30 X2 = SIGNF(13.0,X2)
   40 TEMP = 1.4142136
      U1 = X1/TEMP
      U2 = X2/TEMP
      IF(U1) 60,70,50
   50 IF(U2) 70,70,80
   60 IF(U2) 80,70,70
   70 HAD = .5*(ERR(ABSF(U2))*SIGNF(1.,U2)-ERR(ABSF(U1))
     1 *SIGNF(1.,U1))
      RETURN
   80 HAD = .5*(ERR(-ABSF(U1))*SIGNF(1.,U1)-ERR(-ABSF(U2))*
     1 SIGNF(1.,U2))
      RETURN
      END
C234567
      FUNCTION ERR(W)
      DIMENSION A(25),B(30)
      M=24
      A(1)=164431522242714E-13
      A(2)=-904976049754 8E-13
      A(3)= 0643570883797E-13
      A(4)= 0196418177368E-13
      A(5)=-0001244215694E-13
      A(6)=-0009101941905E-13
      A(7)=-0001796219835E-13
      A(8)= 0000139836786E-13
      A(9)=      164789417E-13
      A(10)=      39009267E-13
      A(11)=-   00893145E-13
      A(12)=-   03747896E-13
      A(13)=-   01298818E-13
```

```
      A(1)=1644315224271 4F-13
      A(2)=-9049760497548E-13
      A(3)= 0643570883797E-13
      A(4)= 019641817736RE-13
      A(5)=-0001244215694E-13
      A(6)=-0009101941905E-13
      A(7)=-0001796219835E-13
      A(8)= 0000139836786E-13
      A(9)=     164789417E-13
      A(10)=     39009267E-13
      A(11)=-   00893145E-13
      A(12)=-   03747896E-13
      A(13)=-   01298818E-13
      A(14)=-   00136773E-13
      A(15)=     00077107E-13
      A(16)=     00046810E-13
      A(17)=     00011844E-13
      A(18)=-        0005E-13
      A(19)=-        1384E-13
      A(20)=-        0652E-13
      A(21)=          0145E-13
      A(22)=          0010E-13
      A(23)=          0024E-13
      A(24)=          0011E-13
      A(25)=          0002E-13
      X=ABSF(W)
      IF(ABSF(X)-.01) 100,110,110
  100 XERR = 2.0/(3.0*1.77245385)*X*(3.0-X**2)
      GO TO 140
  110 ? = (X-1.0)/(X+1.0)
      DO 120 I=1,30
      B(I)=0.
  120 CONTINUE
      DO 130 I=1,M
      M1=(M+1)-I
      B(M1)=2.0*Z*B(M1+1)-B(M1+2)+A(M1+1)
  130 CONTINUE
      F=-3(2)+Z*B(1)+.5*A(1)
      XERR=1.-(1./1.77245385)*(EXPF(-(X**2)))*F
      IF(ABSF(X)-.01) 140,150,150
  140 CERR = 1.0-XERR
      GO TO 160
  150 CERR = (1.0/1.77245385)*(EXPF(-(X**2)))*F
  160 IF(W) 180,170,170
  170 ERR = XERR
      GO TO 200
  180 ERR = CERR
  200 RETURN
      END
```

FOLDOUT FRAME 3

SLIST

```
      COMMON X(100),Z(100),P(100),RP(100)
      PRINT 1000
      PRINT 1004
 1004 FORMAT("0  FUNCTION CODE")
      PRINT 1005
 1005 FORMAT(" 1, DIAMETER"/" 2, SURFACE AREA")
      PRINT 1006
 1006 FORMAT(" 3, DRAG PARAMETER"/" 4, VOLUME"/" 5, MASS")
      PRINT 1007
 1007 FORMAT(" 6, CROSS-SECTIONAL AREA")
 1000 FORMAT("0   PROGRAM TO COMPUTE SCALE PROBABILITIES")
    1 PRINT 1010
 1010 FORMAT("1 NUMBER,END POINT VALUES")
      READ:N1,(X(I),I=1,N1)
      N1M1=N1-1
      PRINT 1015
 1015 FORMAT(" PROBABILITIES")
      READ:(P(I),I=1,N1M1)
      PRINT 1025
 1025 FORMAT(" GIVEN FUNCTION CODE, DENSITY, DRAG")
      READ:I1,D,C
    2 PRINT 1030
 1030 FORMAT(" READ NEXT FUNCTION CODE")
      READ:I2
      IF(I2-7)4,1,1
    4 PRINT 1040
 1040 FORMAT(" NUMBER, POINTS FOR NEXT DENSITY")
      READ:M,(Z(I),I=1,M)
      CALL ADDER(N1M1,M,D,C,I1,I2)
      MT=M
    5 MT=MT-1
      IF(RP(MT))6,6,7
    6 IF(MT-1)7,7,5
    7 JM1=MT
      T=0.
      DO 10 I=1,JM1
      T=T+RP(I)
   10 P(I)=RP(I)
      PRINT 1060
 1060 FORMAT("RESULTING PROBABILITIES")
      PRINT:(RP(I),I=1,JM1)
      PRINT 1070,T
 1070 FORMAT("CHECK SUM =",F10.6)
      N1M1=JM1
      DO 20 I=1,M
   20 X(I)=Z(I)
      PRINT 1001
 1001 FORMAT(1H1)
      I1=I2
      GO TO 2
      END
      SUBROUTINE ADDER(N1M1,M,D,C,I1,I2)
      COMMON X(100),Z(100),P(100),RP(100)
      DO 10 I=2,101
   10 RP(I-1)=0.
      DO  120 I=1,N1M1
      F1=HORN(I1,I2,X(I),D,C)
      F2=HORN(I1,I2,X(I+1),D,C)
      ZH=MAX1F(F1,F2)
      ZL=MIN1F(F1,F2)
      SP=ZH-ZL
      TPR=P(I)
      IF(SP)15,80,20
   15 PRINT 1
```

```
    1 RP(I-1)=0.
      DO  120 I=1,NIM1
      F1=HORN(I1,I2,X(I),D,C)
      F2=HORN(I1,I2,X(I+1),D,C)
      ZH=MAX1F(F1,F2)
      ZL=MIN1F(F1,F2)
      SP=ZH-ZL
      TPR=P(I)
      IF(SP)15,80,20
   15 PRINT 1500
 1500 FORMAT("INTERVAL NEGATIVE")
      GO TO  200
   20 DO 30 JL=2,M
      IF(Z(JL)-ZL)30,30,25
   25 ZLOW=ZL
      JTEMP=JL
      GO TO 40
   30 CONTINUE
   40 CONTINUE
      DO 60 J=JTEMP,M
      IF(Z(J)-ZH)50,45,45
   45 JSAVE=J
      ZHIGH=ZH
      GO TO 70
   50 ZHIGH=Z(J)
      RP(J-1)=RP(J-1)+((ZHIGH-ZLOW)/SP)*TPR
      ZLOW=Z(J)
   60 CONTINUE
   70 RP(JSAVE-1)=RP(JSAVE-1)+((ZHIGH-ZLOW)/SP)*TPR
      GO TO 120
   80 DO 90 J=2,M
      IF(ZL-Z(J-1))85,82,85
   82 RP(J-1)=RP(J-1)+TPR
      JSAVE=J
      GO TO 120
   85 IF(ZL-Z(J))82,90,90
   90 CONTINUE
      RP(M-1)=RP(M-1)+TPR
      JSAVE=M
  120 CONTINUE
  200 RETURN
      END
      FUNCTION HORN(INUTS,JNUTS,   D,C)
      IX=INUTS
      JX=JNUTS
      PI=3.14159265
      X=Y
    1 GO TO (10,20,30,40,50,60),IX
   10 X=PI*X*X
   11 IX=IX+1
      IF(IX-7)18,15,18
   15 IX=1
   18 IF(IX-JX)1,80,1
   20 X=(2.*D*SQRTF(X/PI))/(3.*C)
      GO TO 11
   30 X=(9.*PI/16.)*(C*X/D)**3
      GO TO 11
   40 X=X*D
      GO TO 11
   50 X=(PI/4.)*(6.*X/(PI*D))**(2./3.)
      GO TO 11
   60 X=SQRTF(4.*X/PI)
      GO TO 11
   80 HORN=X
      RETURN
      END
```

```
      COMMON X(100),Y(100),Z(100),P(100),PP(100),RP(100)
      PRINT 1000
 1000 FORMAT("PROGRAM TO FIND PROBABILITY OF SUMS")
      PRINT 1010
 1010 FORMAT("1 NUMBER,POINTS FOR FIRST DENSITY")
      READ:N1,(X(I),I=1,N1)
      N1M1=N1-1
      PRINT 1015
 1015 FORMAT("FIRST SET OF PROBABILITIES")
      READ:(P(I),I=1,N1M1)
      PRINT 1020
 1020 FORMAT("NUMBER,POINTS FOR RESULTING DENSITY")
      READ:M,(Z(I),I=1,M)
    2 PRINT 1030
 1030 FORMAT("NUMBER,POINTS FOR NEXT DENSITY")
      READ:N2,(Y(I),I=1,N2)
      PRINT 1040
 1040 FORMAT("NEXT SET OF PROBABILITIES")
      N2M1=N2-1
      READ:(PP(I),I=1,N2M1)
      CALL ADDER(N1M1,N2M1,M,JSAVE)
      JM1=JSAVE-1
      T=0.
      DO 10 I=1,JM1
      T=T+RP(I)
   10 P(I)=RP(I)
      PRINT 1060
 1060 FORMAT("RESULTING PROBABILITIES")
      PRINT:(RP(I),I=1,JM1)
      PRINT 1070,T
 1070 FORMAT("CHECK SUM =",F10.6)
      N1M1=JM1
      DO 20 I=1,M
   20 X(I)=Z(I)
      PRINT 1001
 1001 FORMAT(1H1)
      GO TO 2
      END
      SUBROUTINE ADDER(N1M1,N2M1,M,JSAVE)
      COMMON X(100),Y(100),Z(100),P(100),PP(100),RP(100)
      DO 10 I=2,M
   10 RP(I-1)=0.
      DO 120 K=1,N1M1
      DO 100 I=1,N2M1
```

```
      END
      SUBROUTINE ADDER(N1M1,N2M1,M,JSAVE)
      COMMON X(100),Y(100),Z(100),P(100),PP(100),RP(100)
      DO 10 I=2,M
   10 RP(I-1)=0.
      DO 120 K=1,N1M1
      DO 100 I=1,N2M1
      ZL=X(K)+Y(I)
      ZH=X(K+1)+Y(I+1)
      SP=ZH-ZL
      TPR=P(K)*PP(I)
      IF(SP)15,80,20
   15 PRINT 1500
 1500 FORMAT("INTERVAL NEGATIVE")
      GO TO  200
   20 DO 30 JL=2,M
      IF(Z(JL)-ZL)30,30,25
   25 ZLOW=ZL
      JTEMP=JL
      GO TO 40
   30 CONTINUE
   40 CONTINUE
      DO 60 J=JTEMP,M
      IF(Z(J)-ZH)50,45,45
   45 JSAVE=J
      ZHIGH=ZH
      GO TO 70
   50 ZHIGH=Z(J)
      RP(J-1)=RP(J-1)+((ZHIGH-ZLOW)/SP)*TPR
      ZLOW=Z(J)
   60 CONTINUE
   70 RP(JSAVE-1)=RP(JSAVE-1)+((ZHIGH-ZLOW)/SP)*TPR
      GO TO 100
   80 DO 90 J=2,M
      IF(ZL-Z(J-1))85,82,85
   82 RP(J-1)=RP(J-1)+TPR
      JSAVE=J
      GO TO 100
   85 IF(ZL-Z(J))82,90,90
   90 CONTINUE
      RP(M-1)=RP(M-1)+TPR
      JSAVE=M
  100 CONTINUE
  120 CONTINUE
  200 RETURN
      END
```

```
C      BINOMX-----NORMAL APPROXIMATION TO NORMAL
     1 PRINT 2000
  2000 FORMAT("1  NORMAL APPROXIMATION TO BINOMIAL")
     2 PRINT 2020
  2020 FORMAT("0READ FIRST,LAST VALUE, NUMBER, PROBABILITY")
       READ:X1,X2,XMAX,THETA
       XMU=XMAX*THETA
       SIGMA=SQRTF(XMU*(1.-THETA))
       X1=(X1-.5-XMU)/SIGMA
       X2=(X2+.5-XMU)/SIGMA
       ANS=HAD(X1,X2)
       PRINT 2025,X1,X2
  2025 FORMAT(" STANDARD NORMAL LIMITS*****",2E14.6)
       PRINT 2030,ANS
  2030 FORMAT(" PROBABILITY=",E14.6)
       GO TO 2
       END
       FUNCTION HAD(X1,X2)
       TEMP=1.4142136
       U1=X1/TEMP
       U2=X2/TEMP
       IF(U1)300,200,100
   100 IF(U2)200,200,900
   300 IF(U2)900,200,200
   200 HAD=.5*(ERR(ABSF(U2))*SIGNF(1.,U2)-ERR(ABSF(U1))
      1 *SIGNF(1.,U1))
       RETURN
   900 HAD=.5*(ERR(-ABSF(U1))*SIGNF(1.,U1)-ERR(-ABSF(U2))*
      1 SIGNF(1.,U2))
       RETURN
       END
       FUNCTION ERR(W)
       DIMENSION A(25),B(30)
       M=24
       A(1)=1644315224271 4E-13
       A(2)=-9049760497548E-13
       A(3)='0643570883797E-13
       A(4)= 0196418177368E-13
       A(5)=-0001244215694E-13
       A(6)=-0009101941905E-13
       A(7)=-0001796219835E-13
       A(8)= 0000139836786E-13
       A(9)=     [illegible]
```

```
      A(4)= 019641817736RE-13
      A(5)=-0001244215694E-13
      A(6)=-0009101941905E-13
      A(7)=-0001796219835E-13
      A(8)= 0000139836786E-13
      A(9)=      164789417E-13
      A(10)=      39009267E-13
      A(11)=-   00893145E-13
      A(12)=-   03747896E-13
      A(13)=-   01298818E-13
      A(14)=-   00136773E-13
      A(15)=      00077107E-13
      A(16)=      00046810E-13
      A(17)=      00011844E-13
      A(18)=-        0005E-13
      A(19)=-        1384E-13
      A(20)=-        0652E-13
      A(21)=           0145E-13
      A(22)=           0010E-13
      A(23)=           0024E-13
      A(24)=           0011E-13
      A(25)=           0002E-13
      X=ABSF(W)
      IF(ABSF(X)-.01)1,2,2
    1 XERR=2.0/(3.0*1.77245385)*X*(3.0-X**2)
      GO TO 6
    2 Z=(X-1.0)/(X+1.0)
      DO 3 I=1,30
      B(I)=0.
    3 CONTINUE
      DO 4 I=1,M
      M1=(M+1)-I
      B(M1)=2.0*Z*B(M1+1)-B(M1+2)+A(M1+1)
    4 CONTINUE
      F=-B(2)+Z*B(1)+.5*A(1)
      XERR=1.0-(1.0/1.77245385)*(EXPF(-(X**2)))*F
      IF(ABSF(X)-.01)6,7,7
    6 CERR=1.0-XERR
      GO TO 5
    7 CERR=(1.0/1.77245385)*(EXPF(-(X**2)))*F
    5 IF(W)9,8,8
    8 ERR=XERR
      GO TO 13
    9 ERR=CERR
   13 RETURN
      END
```

FOLDOUT FRAME 2

```
      COMMON X(100),Y(100),Z(100),RP(100),M
      PRINT 2000
2000  FORMAT("1   PROGRAM TO LOAD PROBABILITIES")
      PRINT 2020
2020  FORMAT("0NUMBER, OUTPUT GRID VALUES")
   1  READ:M,(Z(I),I=1,M)
   2  PRINT 2040
2040  FORMAT(" INTERVAL START, END, PROBABILITY")
      READ:ZL,ZH,TPR
      CALL ADDER(ZL,ZH,TPR)
      PRINT 2060
2060  FORMAT("0     RESULTING PROBABILITIES")
      MM1=M-1
      PRINT:(RP(I),I=1,MM1)
      PRINT 1000
1000  FORMAT (1H1)
      GO TO 2
      END
      SUBROUTINE ADDER(ZL,ZH,TPR)
      COMMON X(100),Y(100),Z(100),RP(100),M
      DO 10 I=2,M
  10  RP(I-1)=0.
      SP=ZH-ZL
      IF(SP)15,80,20
  15  PRINT 1500
1500  FORMAT("0 INTERVAL NEGATIVE")
      GO TO 200
  20  DO 30 JL=2,M
      IF(Z(JL)-ZL)30,30,25
  25  ZLOW=ZL
      JTEMP=JL
      GO TO 40
  30  CONTINUE
  40  CONTINUE
      DO 60 J=JTEMP,M
      IF(Z(J)-ZH)50,45,45
  45  JSAVE=J
      ZHIGH=ZH
      GO TO 70
  50  ZHIGH=Z(J)
      RP(J-1)=RP(J-1)+((ZHIGH-ZLOW)/SP)*TPR
      ZLOW=Z(J)
  60  CONTINUE
  70  RP(JSAVE-1)=RP(JSAVE-1)+((ZHIGH-ZLOW)/SP)*TPR
      GO TO 100
  80  DO 90 J=2,M
      IF(ZL-Z(J-1))85,82,85
  82  RP(J-1)=RP(J-1)+TPR
      GO TO 100
  85  IF(ZL-Z(J))82,90,90
  90  CONTINUE
      RP(M-1)=RP(M-1)+TPR
 100  CONTINUE
 120  CONTINUE
 200  RETURN
      END
```

```
      COMMON X(100),Y(100),Z(100),P(100),PP(100),RP(100)
      COMMON A(20),B(20),C(20),NN,YMEAN
      PRINT 1000
      PRINT 999
  999 FORMAT("1  ORBIT TYPE(1-6),ATMOS TYPE(1 OR 2)")
 1000 FORMAT(" TIME IN MARS ORBIT PROGRAM")
      NN=4
      READ:IT,ITT
      A(1)=A(2)=A(3)=A(4)=600.
      B(1)=0.
      B(2)=66666.67
      B(3)=51333.33
      B(4)=46285.71
      GO TO (60,80),ITT
   60 C(1)=.01961519
      C(2)=-4.065703E-5
      C(3)=8.695401E-8
      C(4)=-1.711093E-10
      YMEAN=-.1833621
      GO TO 90
   80 C(1)=.0157079
      C(2)=-2.496592E-5
      C(3)=5.334866E-8
      C(4)=-1.370223E-10
      YMEAN=-3.913773
   90 GO TO (91,92,93,94,95,96),IT
   91 HA=10000.
      PA=1000.
      GO TO 100
   92 HA=10000.
      PA=500.
      GO TO 100
   93 HA=10000.
      PA=200.
      GO TO 100
   94 HA=20000.
      PA=1000.
      GO TO 100
   95 HA=20000.
      PA=500.
      GO TO 100
   96 HA=20000.
      PA=200.
  100 PRINT 998
  998 FORMAT("   APOAPSIS(KM)      PERIAPSIS(KM)")
      PRINT:HA,PA
    2 PRINT 1010
 1010 FORMAT(" NUMBER, PERIAPSIS VALUES")
      READ:N1,(X(I),I=1,N1)
      N1M1=N1-1
      PRINT 1015
 1015 FORMAT(" PERIAPSIS PROBABILITIES")
      READ:(P(I),I=1,N1M1)
      PRINT 1020
 1020 FORMAT(" NUMBER, VALUES FOR TIME IN ORBIT")
      READ:M,(Z(I),I=1,M)
      PRINT 1030
 1030 FORMAT(" NUMBER, VALUES FOR DRAG PARAMETER")
      READ:N2,(Y(I),I=1,N2)
```

```
1015 FORMAT(" PERIAPSIS PROBABILITIES")
     READ:(P(I),I=1,N1M1)
     PRINT 1020
1020 FORMAT(" NUMBER, VALUES FOR TIME IN ORBIT")
     READ:M,(Z(I),I=1,M)
     PRINT 1030
1030 FORMAT(" NUMBER, VALUES FOR DRAG PARAMETER")
     READ:N2,(Y(I),I=1,N2)
     PRINT 1040
1040 FORMAT(" DRAG PARAMETER PROBABILITIES")
     N2M1=N2-1
     READ:(PP(I),I=1,N2M1)
     CALL MULPY(N1M1,N2M1,M,JSAVE,HA)
     MT=M
   5 MT=MT-1
     IF(RP(MT))6,6,7
   6 IF(MT-1)7,7,5
   7 JM1=MT
     T=0.
     DO 10 I=1,JM1
  10 T=T+RP(I)
     PRINT 1060
1060 FORMAT(" TIME IN ORBIT DISTRIBUTION")
     PRINT:(RP(I),I=1,JM1)
     PRINT 1070,T
1070 FORMAT("CHECK SUM =",F10.6)
     PRINT 1001
1001 FORMAT(1H1)
     GO TO 2
     END
     SUBROUTINE MULPY(N1M1,N2M1,M,JSAVE,HA)
     COMMON X(100),Y(100),Z(100),P(100),PP(100),RP(100)
     COMMON A(20),B(20),C(20),NN,YMEAN
     DO 10 I=2,M
  10 RP(I-1)=0.
     DO 120 K=1,N1M1
     DO 100 I=1,N2M1
     T1=TOPH(X(K))
     T2=TOPH(X(K+1))
     U1=T1*HA*Y(I)
     U2=T1*HA*Y(I+1)
     U3=T2*HA*Y(I)
     U4=T2*HA*Y(I+1)
     ZL=MIN1F(U1,U2,U3,U4)
     ZH=MAX1F(U1,U2,U3,U4)
     SP=ZH-ZL
     TPR=P(K)*PP(I)
     IF(SP)15,80,20
  15 PRINT 1500
1500 FORMAT("INTERVAL NEGATIVE")
     GO TO 200
  20 DO 30 JL=2,M
     IF(Z(JL)-ZL)30,30,25
  25 ZLOW=ZL
     JTEMP=JL
     GO TO 40
  30 CONTINUE
  40 CONTINUE
     DO 60 J=JTEMP,M
     IF(Z(J)-ZH)50,45,45
  45 JSAVE=J
     ZHIGH=ZH
     GO TO 70
  50 ZHIGH=Z(J)
     RP(J-1)=RP(J-1)+((ZHIGH-ZLOW)/SP)*TPR
     ZLOW=Z(J)
```

```
 20 DO 30 JL=2,M
    IF(Z(JL)-ZL)30,30,25
 25 ZLOW=ZL
    JTEMP=JL
    GO TO 40
 30 CONTINUE
 40 CONTINUE
    DO 60 J=JTEMP,M
    IF(Z(J)-ZH)50,45,45
 45 JSAVE=J
    ZHIGH=ZH
    GO TO 70
 50 ZHIGH=Z(J)
    RP(J-1)=RP(J-1)+((ZHIGH-ZLOW)/SP)*TPR
    ZLOW=Z(J)
 60 CONTINUE
 70 RP(JSAVE-1)=RP(JSAVE-1)+((ZHIGH-ZLOW)/SP)*TPR
    GO TO 100
 80 DO 90 J=2,M
    IF(ZL-Z(J-1))85,82,85
 82 RP(J-1)=RP(J-1)+TPR
    JSAVE=J
    GO TO 100
 85 IF(ZL-Z(J))82,90,90
 90 CONTINUE
    RP(M-1)=RP(M-1)+TPR
    JSAVE=M
100 CONTINUE
120 CONTINUE
200 RETURN
    END
    FUNCTION TOPH(ZZ)
    COMMON X(100),Y(100),Z(100),P(100),PP(100),RP(100)
    COMMON A(20),B(20),C(20),NN,YMEAN
    DIMENSION POL(20)
    POL(1)=1.
    POL(2)=ZZ-A(1)
    MM=NN+1
    DO 20 I=3,MM
 20 POL(I)=ZZ*POL(I-1)-A(I-1)*POL(I-1)-B(I-1)*POL(I-2)
    TOPH=YMEAN
    DO 25 K=1,NN
 25 TOPH=TOPH+C(K)*POL(K+1)
    TOPH=EXPF(TOPH)
    RETURN
    END
```

FOLDOUT FRAME 3

```
      SUBROUTINE CUFIT(T,NTYPE,T1D,T2D,T3D,T4D)
      COMMON A1(4),B1(4),C1(4),A2(4),B2(4),C2(4)
      COMMON A3(4),B3(4),C3(4),A4(4),B4(4),C4(4)
      COMMON A5(4),B5(4),C5(4),A6(4),B6(4),C6(4)
      COMMON YM(6)
      COMMON DUMMY(1000)
      CALL COEFF
      MM=5
      NN=4
      TS=SQRTF(T)
      T1D=EXPF(OPOLY(TS,A1,B1,C1,MM,NN,YM(1)))
      T2D=OPOLY(T,A2,B2,C2,MM,NN,YM(2))*1.E3
      GO TO(10,15),NTYPE
   10 T3D=EXPF(OPOLY(TS,A3,B3,C3,MM,NN,YM(3)))
      GO TO 20
   15 T3D=EXPF(OPOLY(TS,A4,B4,C4,MM,NN,YM(4)))
   20 GO TO (30,35),NTYPE
   30 T4D=OPOLY(T,A5,B5,C5,MM,NN,YM(5))*1.E3
      GO TO 40
   35 T4D=OPOLY(T,A6,B6,C6,MM,NN,YM(6))*1.E3
   40 RETURN
      END
      SUBROUTINE COEFF
      COMMON A1(4),B1(4),C1(4),A2(4),B2(4),C2(4)
      COMMON A3(4),B3(4),C3(4),A4(4),B4(4),C4(4)
      COMMON A5(4),B5(4),C5(4),A6(4),B6(4),C6(4)
      COMMON YM(6)
      COMMON DUMMY(1000)
      A1(1)=4.978734
      A1(2)=11.28421
      A1(3)=9.779519
      A1(4)=9.67532
      B1(1)=B2(1)=B3(1)=B4(1)=B5(1)=B6(1)=0.
      B1(2)=41.13912
      B1(3)=18.10669
      B1(4)=25.86573
      C1(1)=.6436589
      C1(2)=-.03881528
      C1(3)=.001998154
      C1(4)=-.2709923E-3
      A2(1)=A2(2)=A2(3)=A2(4)=175.
      B2(2)=13125.
      B2(3)=10000.
      B2(4)=8839.286
      C2(1)=-.01707143
      C2(2)=-.3183333E-3
      C2(3)=5.59596E-7
      C2(4)=3.924243E-9
      A3(1)=.4870088
      A3(2)=11.11457
      A3(3)=8.392233
      A3(4)=8.578036
      B3(2)=4.899994
      B3(3)=8.629762
      B3(4)=9.727467
      C3(1)=.597208
      C3(2)=-5.188791E-3
      C3(3)=-2.828954E-4
      C3(4)=3.301192E-4
      A4(1)=.949203
      A4(2)=14.39388
      A4(3)=10.94751
      A4(4)=10.63796
      B4(2)=12.26415
      B4(3)=12.82967
      B4(4)=19.59597
      C4(1)=.4797487
      C4(2)=-1.435841E-2
      C4(3)=1.445613E-4
      C4(4)=1.624449E-4
```

FOLDOUT FRAME 1

```
      A4(2)=1[illegible]
      A4(3)=19.94751
      A4(4)=19.63796
      B4(2)=12.26415
      B4(3)=12.82967
      B4(4)=19.59597
      C4(1)=.4797487
      C4(2)=-1.435841E-2
      C4(3)=1.445613E-4
      C4(4)=1.624449E-4
      A5(1)=9.259259
      A5(2)=158.5939
      A5(3)=122.7774
      A5(4)=B5(4)=C5(4)=A6(4)=B6(4)=C6(4)=0.
      B5(2)=1393.155
      B5(3)=1816.049
      C5(1)=.03448159
      C5(2)=-4.78121E-4
      C5(3)=-3.141853E-6
      A6(1)=A6(2)=A6(3)=175.
      B6(2)=13125.
      B6(3)=19999.
      C6(1)=-2.595238E-3
      C6(2)=-2.902381E-4
      C6(3)=-8.343434E-7
      YM(1)=3.949467
      YM(2)=2.9875
      YM(3)=.2937196
      YM(4)=.4751153
      YM(5)=.387937
      YM(6)=3.6375
      RETURN
      END
      FUNCTION OPOLY(Z,A,B,C,MM,NN,YMEAN)
      DIMENSION A(1),B(1),C(1),POL(50)
      POL(1)=1.
      POL(2)=Z-A(1)
      DO 20 I=3,MM
   20 POL(I)=Z*POL(I-1)-A(I-1)*POL(I-1)-B(I-1)*POL(I-2)
      OPOLY=YMEAN
      DO 25 K=1,NN
   25 OPOLY=OPOLY+C(K)*POL(K+1)
      RETURN
      END
      SUBROUTINE DDDD(SP,TPR,ZL,ZH,JSAVE,M)
      COMMON DUMMY(78)
      COMMON X1(100),X2(100),X3(100),X4(100),Z(100)
      COMMON P1(100),P2(100),P3(100),P4(100),RP(100)
      IF(SP)100,80,20
   20 DO 30 JL=2,M
      IF (Z(JL)-ZL)30,30,25
   25 ZLOW=ZL
      JTEMP=JL
      GO TO 40
   30 CONTINUE
   40 CONTINUE
      DO 60 J=JTEMP,M
      IF(Z(J)-ZH)50,45,45
   45 JSAVE=J
      ZHIGH=ZH
      GO TO 70
   50 ZHIGH=Z(J)
      RP(J-1)=RP(J-1)+((ZHIGH-ZLOW)/SP)*TPR
      ZLOW=Z(J)
   60 CONTINUE
   70 RP(JSAVE-1)=RP(JSAVE-1)+((ZHIGH-ZLOW)/SP)+TPR
      GO TO 100
   80 DO 90 J=2,M
      IF(ZL-Z(J-1))85,82,85
   82 RP(J-1)=RP(J-1)+TPR
      JSAVE=J
      GO TO 100
   85 IF(ZL-Z(J))82,90,90
   90 CONTINUE
      RP(M-1)=RP(M-1)+TPR
      JSAVE=M
  100 CONTINUE
      RETURN
      END
```

```
      COMMON PMARG(100),PTOP(100)
      COMMON INT(100)
    1 PRINT 1000
      READ:NMAX,MMAX,T,SUM
      READ:(PTOP(I),I=1,MMAX)
      READ:(INT(I),I=1,MMAX)
      NMAXP=NMAX+1
      DO 15 I=1,MMAX
   15 PTOP(I)=PTOP(I)/SUM
      DO 20 I=1,NMAXP
   20 PMARG(I)=0.
      RAT=T/(1.-T)
      DO 40 K=1,MMAX
      N=INT(K)
      FN=N
      X=0.
      PR=(1.-T)**N
      PMARG(1)=PMARG(1)+PR*PTOP(K)
   25 IF(X-FN)30,40,30
   30 PR=PR*(FN-X)*RAT/(X+1.)
      X=X+1.
      NX=X+1
      PMARG(NX)=PMARG(NX)+PR*PTOP(K)
      GO TO 25
   40 CONTINUE
      PRINT 1000
 1000 FORMAT(1H1)
      SS=0.
      DO 50 I=1,NMAXP
      NI=I-1
      PRINT:NI,PMARG(I)
      SS=SS+PMARG(I)
   50 CONTINUE
      PRINT 1000
      PRINT:SS
      GO TO 1
      END
```

```
$LIST

      COMMON X(100),Y(100),Z(100),RP(100),M
    1 READ:M,(Z(I),I=1,M)
    2 READ:ZL,ZH,TPR
      CALL ADDER(ZL,ZH,TPR)
      PRINT:(RP(I),I=1,M)
      PRINT 1000
 1000 FORMAT(1H1)
      GO TO 2
      END
      SUBROUTINE ADDER(ZL,ZH,TPR)
      COMMON X(100),Y(100),Z(100),RP(100),M
      DO 10 I=2,M
   10 RP(I-1)=0.
      SP=ZH-ZL
      IF(SP)15,80,20
   15 PRINT 1500
 1500 FORMAT("INTERVAL NEGATIVE")
      GO TO 200
   20 DO 30 JL=2,M
      IF(Z(JL)-ZL)30,30,25
   25 ZLOW=ZL
      JTEMP=JL
      GO TO 40
   30 CONTINUE
   40 CONTINUE
      DO 60 J=JTEMP,M
      IF(Z(J)-ZH)50,45,45
   45 JSAVE=J
      ZHIGH=ZH
      GO TO 70
   50 ZHIGH=Z(J)
      RP(J-1)=RP(J-1)+((ZHIGH-ZLOW)/SP)*TPR
      ZLOW=Z(J)
   60 CONTINUE
   70 RP(JSAVE-1)=RP(JSAVE-1)+((ZHIGH-ZLOW)/SP)*TPR
      GO TO 100
   80 DO 90 J=2,M
      IF(ZL-Z(J-1))85,82,85
   82 RP(J-1)=RP(J-1)+TPR
      GO TO 100
   85 IF(ZL-Z(J))82,90,90
   90 CONTINUE
      RP(M-1)=RP(M-1)+TPR
  100 CONTINUE
  120 CONTINUE
  200 RETURN
      END
```

OLD PROGRAM NAME--HELIO3
WAIT.

READY.

HELIO 3
1/16/67

*LIST

```
      COMMON  DUMMY(78)
      COMMON X1(100),X2(100),X3(100),X4(100),Z(100)
      COMMON P1(100),P2(100),P3(100),P4(100),RP(100)
      COMMON XC2(100),XC3(100)
      PRINT 1000
    2 PRINT 999
      RAD=57.2957795
  999 FORMAT("1 DAYS TO IMPACT, ORBIT TYPE,T,R MARS ")
 1000 FORMAT(" HELIOCENTRIC ORBIT PROBABILITY PROGRAM")
      READ:TIME,NTYPE,TP,RRP
      CALL CUFIT(TIME,NTYPE,C1,C2,C3,C4)
      PRINT 998
  998 FORMAT(" NUMBER, VELOCITY MAG VALUES(M/S)")
      READ:N1,(X1(I),I=1,N1)
      N1M1=N1-1
      PRINT 997
  997 FORMAT(" VELOCITY MAG PROBABILITIES")
      READ:(P1(I),I=1,N1M1)
      PRINT 996
  996 FORMAT(" NUMBER,POLAR ANGLE VALUES(DEG)")
      READ:N2,(X2(I),I=1,N2)
      PRINT 1010
 1010 FORMAT(" POLAR ANGLE PROBABILITIES")
      N2M1=N2-1
      READ:(P2(I),I=1,N2M1)
      PRINT 1015
 1015 FORMAT(" NUMBER, N-T PLANE ANGLE VALUES(DEG)")
      READ:N3,(X3(I),I=1,N3)
      N3M1=N3-1
      PRINT 1020
 1020 FORMAT(" N-T PLANE ANGLE PROBABILITIES")
      READ:(P3(I),I=1,N3M1)
      PRINT 1025
 1025 FORMAT(" NUMBER, M/CDA VALUES(SLUGS/FT*FT)")
      READ:N4,(X4(I),I=1,N4)
      N4M1=N4-1
      PRINT 1027
 1027 FORMAT(" M/CDA PROBABILITIES")
      READ:(P4(I),I=1,N4M1)
      DO 5000 I=1,N2
      QQ=X2(I)/RAD
      X2(I)=SINF(QQ)
 5000 XC2(I)=COSF(QQ)
      DO 5010 I=1,N3
      QQ=X3(I)/RAD
      X3(I)=SINF(QQ)
 5010 XC3(I)=COSF(QQ)
      PRINT 1030
 1030 FORMAT(" NUMBER, MISS DISTANCE VALUES (KM)")
      READ:M,(Z(I),I=1,M)
      PRINT 1080
 1080 FORMAT("0      T1/DVT      T2/DVN      T3(M/CDA)      R/DVR
     +     C-F")
      PRINT:C1,C2,C3,C4
      CALL GROSS(N1M1,N2M1,N3M1,N4M1,C1,C2,C3,C4,TP,RRP,
     +JSAVE,M)
      MT=M
    5 MT=MT-1
```

FOLDOUT FRAME 1

```
      [illegible]
      PRINT 1080
 1080 FORMAT("0     T1/DVT      T2/DVN      T3(M/CDA)      R/DVR
     +    C-F")
      PRINT:C1,C2,C3,C4
      CALL GROSS(N1M1,N2M1,N3M1,N4M1,C1,C2,C3,C4,TP,RRP,
     +JSAVE,M)
      MT=M
    5 MT=MT-1
      IF(RP(MT))6,6,7
    6 IF(MT-1)7,7,5
    7 JM1=MT
      T=0.
      DO 10 I=1,JM1
   10 T=T+RP(I)
      PRINT 1060
 1060 FORMAT("0  ***** MISS DISTANCE PROBABILITIES *****")
      PRINT:(RP(I),I=1,JM1)
      PRINT 1070,T
 1070 FORMAT("CHECK SUM =",F10.6)
      PRINT 1001
 1001 FORMAT(1H1)
      GO TO 2
      END
      SUBROUTINE GROSS(N1M1,N2M1,N3M1,N4M1,C1,C2,C3,C4,
     +TP,RRP,JSAVE,M)
      COMMON DUMMY(78)
      COMMON X1(100),X2(100),X3(100),X4(100),Z(130)
      COMMON P1(100),P2(100),P3(100),P4(100),RP(100)
      COMMON XC2(100),XC3(100)
      DIMENSION Q(20)
      DO 10 I=2,M
   10 RP(I-1)=0.
      DO 130 I1=1,N1M1
      DO 120 I2=1,N2M1
      DO 110 I3=1,N3M1
      DO 100 I4=1,N4M1
      TPR=P1(I1)*P2(I2)*P3(I3)*P4(I4)
      JZ=0
      DO 20 I01=1,2
      IZ1=I1+I01-1
      DO 30 I02=1,2
      IZ2=I2+I02-1
      DO 40 I03=1,2
      IZ3=I03+I3-1
      DO 50 I04=1,2
      IZ4=I4+I04-1
      JZ=JZ+1
      Q(JZ)=ZZZ(C1,C2,C3,C4,X1(IZ1),X2(IZ2),X3(IZ3),X4(IZ4)
     +,XC2(IZ2),XC3(IZ3),TP,RRP)
   50 CONTINUE
   40 CONTINUE
   30 CONTINUE
   20 CONTINUE
      ZL=MIN1F(Q(1),Q(2),Q(3),Q(4),Q(5),Q(6),Q(7),Q(8),Q(9),
     +Q(10),Q(11),Q(12),Q(13),Q(14),Q(15),Q(16))
      ZH=MAX1F(Q(1),Q(2),Q(3),Q(4),Q(5),Q(6),Q(7),Q(8),Q(9),
     +Q(10),Q(11),Q(12),Q(13),Q(14),Q(15),Q(16))
      SP=ZH-ZL
      CALL DDDD(SP,TPR,ZL,ZH,JSAVE,M)
  100 CONTINUE
  110 CONTINUE
  120 CONTINUE
  130 CONTINUE
      RETURN
      END
      FUNCTION ZZZ(D1,D2,D3,D4,VM,STM,SPM,RT,CTM,CPM,TP,RRP)
      T=VM*STM*(D1*SPM+D2*CPM)+D3/RT
      R=D4*VM*CTM
      ZZZ=SQRTF((T-TP)**2+(R-RRP)**2)
      RETURN
      END
```

FOLDOUT FRAME 2

ЗAD.

$LIST

```
      COMMON X(100),Y(100),Z(100),P(100),PP(100),RP(100)
      COMMON A(20),B(20),NN,C
      PRINT 1000
      PRINT 999
  999 FORMAT("1 ORBIT TYPE(1-6)")
 1000 FORMAT(" PERIAPSIS DISTRIBUTION PROGRAM")
      READ:I1
      CALL OTYPE(I1,MM)
      A(1)=A(1)/2.
      A(NN)=A(NN)/2.
      FFM=MM
      C=3.14159265/FFM
    2 PRINT 1010
 1010 FORMAT("1 NUMBER,ANGLE ATTACK VALUES")
      READ:N1,(X(I),I=1,N1)
      N1M1=N1-1
      PRINT 1015
 1015 FORMAT(" ANGLE PROBABILITIES, 90. TO 270.")
      READ:(P(I),I=1,N1M1)
      PRINT 1020
 1020 FORMAT(" NUMBER,VALUES FOR PERIAPSIS DECREMENT")
      READ:M,(Z(I),I=1,M)
      PRINT 1030
 1030 FORMAT(" NUMBER,VALUES FOR VELOCITY INCREMENT")
      READ:N2,(Y(I),I=1,N2)
      PRINT 1040
 1040 FORMAT(" VELOCITY INCREMENT PROBABILITIES")
      N2M1=N2-1
      READ:(PP(I),I=1,N2M1)
      CALL ADDER(N1M1,N2M1,M,JSAVE)
      MT=M
    5 MT=MT-1
      IF(RP(MT))6,6,7
    6 IF(MT-1)7,7,5
    7 JM1=MT
      T=0.
      DO 10 I=1,JM1
   10 T=T+RP(I)
      PRINT 1060
 1060 FORMAT("      PERIAPSIS DECREMENT DISTRIBUTION")
      PRINT:(RP(I),I=1,JM1)
      PRINT 1070,T
 1070 FORMAT("CHECK SUM =",F10.6)
      PRINT 1001
 1001 FORMAT(1H1)
      GO TO 2
      END
      SUBROUTINE ADDER(N1M1,N2M1,M,JSAVE)
      COMMON X(100),Y(100),Z(100),P(100),PP(100),RP(100)
      COMMON A(20),B(20),NN,C
      DO 10 I=2,M
   10 RP(I-1)=0.
      DO 120 K=1,N1M1
      DO 100 I=1,N2M1
      T1=PDV(X(K))
      T2=PDV(X(K+1))
      U1=T1*Y(I)
      U2=T1*Y(I+1)
```

```
      DO 120 K=1,N1M1
      DO 100 I=1,N2M1
      T1=PDV(X(K))
      T2=PDV(X(K+1))
      U1=T1*Y(I)
      U2=T1*Y(I+1)
      U3=T2*Y(I)
      U4=T2*Y(I+1)
      ZL=MIN1F(U1,U2,U3,U4)
      ZH=MAX1F(U1,U2,U3,U4)
      SP=ZH-ZL
      TPR=P(K)*PP(I)
      IF(SP)15,80,20
   15 PRINT 1500
 1500 FORMAT("INTERVAL NEGATIVE")
      GO TO  200
   20 DO 30 JL=2,M
      IF(Z(JL)-ZL)30,30,25
   25 ZLOW=ZL
      JTEMP=JL
      GO TO 40
   30 CONTINUE
   40 CONTINUE
      DO 60 J=JTEMP,M
      IF(Z(J)-ZH)50,45,45
   45 JSAVE=J
      ZHIGH=ZH
      GO TO 70
   50 ZHIGH=Z(J)
      RP(J-1)=RP(J-1)+((ZHIGH-ZLOW)/SP)*TPR
      ZLOW=Z(J)
   60 CONTINUE
   70 RP(JSAVE-1)=RP(JSAVE-1)+((ZHIGH-ZLOW)/SP)*TPR
      GO TO 100
   80 DO 90 J=2,M
      IF(ZL-Z(J-1))85,82,85
   82 RP(J-1)=RP(J-1)+TPR
      JSAVE=J
      GO TO 100
   85 IF(ZL-Z(J))82,90,90
   90 CONTINUE
      RP(M-1)=RP(M-1)+TPR
      JSAVE=M
  100 CONTINUE
  120 CONTINUE
  200 RETURN
      END
      FUNCTION PDV(ZZ)
      COMMON X(100),Y(100),Z(100),P(100),PP(100),RP(100)
      COMMON A(20),B(20),NN,C
      ZZ=(ZZ-90.)/10.
      PDV=0.
      DO 15 J=1,NN
      FJ=J-1
      G=C*FJ*ZZ
   15 PDV=PDV+A(J)*COSF(G)+B(J)*SINF(G)
      RETURN
      END
```

FOLD-IN 2

WAIT.

READY.

DELP2
1/16/67

$LIST

```
C     CURVE FITTING PROGRAM FOR DELP1
      SUBROUTINE OTYPE(IT,MM)
      COMMON DUMMY(600)
      COMMON A(20),B(20),NN,C
      NN=10
      MM=9
      DO 20 I=1,20
   20 B(I)=0.
      GO TO (100,200,300,400,500,600),IT
  100 A(1)=10.59444
      A(2)=-3.348694
      A(3)=-.7277126
      A(4)=-.2944445
      A(5)=-.179111
      A(6)=-.1401124
      A(7)=-.1388889
      A(8)=-.09452752
      A(9)=-.07650987
      A(10)=-.09444452
      RETURN
  200 A(1)=9.794444
      A(2)=-3.100466
      A(3)=-.6714439
      A(4)=-.2611111
      A(5)=-.1491711
      A(6)=-.1161407
      A(7)=-.1055556
      A(8)=-.1000602
      A(9)=-.09605177
      A(10)=-.09444452
      RETURN
  300 A(1)=9.194444
      A(2)=-2.9040966
      A(3)=-.6530806
      A(4)=-.2722222
      A(5)=-.1060668
      A(6)=-.1198379
```

```
      A(9)=-.09605177
      A(10)=-.09444452
      RETURN
  300 A(1)=9.194444
      A(2)=-2.9040966
      A(3)=-.6530806
      A(4)=-.2722222
      A(5)=-.1060668
      A(6)=-.1198379
      A(7)=-.07222224
      A(8)=-.05586342
      A(9)=-.09085276
      A(10)=-.07222228
      RETURN
  400 A(1)=15.69444
      A(2)=-5.006866
      A(3)=-1.155144
      A(4)=-.4277778
      A(5)=-.3001248
      A(6)=-.2253168
      A(7)=-.1388889
      A(8)=-.1511513
      A(9)=-.1280644
      A(10)=-.1277779
      RETURN
  500 A(1)=15.03889
      A(2)=-4.891183
      A(3)=-1.157197
      A(4)=-.45
      A(5)=-.3137546
      A(6)=-.1722799
      A(7)=-.09444444
      A(8)=-.08653781
      A(9)=-.07904804
      A(10)=-.05000005
      RETURN
  600 A(1)=14.39444
      A(2)=-4.944771
      A(3)=-1.200388
      A(4)=-.4944444
      A(5)=-.2284768
      A(6)=-.1005758
      A(7)=-.07222217
      A(8)=-.03798681
      A(9)=-.07886461
      A(10)=.1055556
      RETURN
      END
```

FOLDOUT FRAME 2

```
READY.

                                                                    HELIO1
$LIST                                                               1/16/67

      COMMON  DUMMY(78)
      COMMON X1(100),X2(100),X3(100),X4(100),Z(100)
      COMMON P1(100),P2(100),P3(100),P4(100),RP(100)
      PRINT 1000
    2 PRINT 999
  999 FORMAT("1 DAYS TO IMPACT, ORBIT TYPE,T,R MARS ")
 1000 FORMAT(" HELIOCENTRIC ORBIT PROBABILITY PROGRAM")
      READ:TIME,NTYPE,TP,RRP
      CALL CUFIT(TIME,NTYPE,C1,C2,C3,C4)
      PRINT 998
  998 FORMAT(" NUMBER, TANGENTIAL VELOCITY VALUES")
      READ:N1,(X1(I),I=1,N1)

      N1M1=N1-1
      PRINT 997
  997 FORMAT(" TANGENTIAL PROBABILITIES")
      READ:(P1(I),I=1,N1M1)
      PRINT 996
 996  FORMAT(" NUMBER, NORMAL VELOCITY VALUES")
      READ:N2,(X2(I),I=1,N2)
      PRINT 1010
 1010 FORMAT(" NORMAL PROBABILITIES")
      N2M1=N2-1
      READ:(P2(I),I=1,N2M1)
      PRINT 1015
 1015 FORMAT(" NUMBER, M/CDA VALUES")
      READ:N3,(X3(I),I=1,N3)
      N3M1=N3-1
      PRINT 1020
 1020 FORMAT(" M/CDA PROBABILITIES")
      READ:(P3(I),I=1,N3M1)
      PRINT 1025
 1025 FORMAT(" NUMBER, O-O-P VELOCITY VALUES")
      READ:N4,(X4(I),I=1,N4)
      N4M1=N4-1
      PRINT 1027
 1027 FORMAT(" O-O-L PROBABILITIES")
      READ:(P4(I),I=1,N4M1)
      PRINT 1030
 1030 FORMAT(" NUMBER, MISS DISTANCE VALUES")
      READ:M,(Z(I),I=1,M)
      PRINT 1080
 1080 FORMAT("0  T1/DVT  T2/DVN  T3(M/CDA)  R/DVR   C-F")
      PRINT:C1,C2,C3,C4
      CALL GROSS(N1M1,N2M1,N3M1,N4M1,C1,C2,C3,C4,TP,RRP,
     +JSAVE,M)
      MT=M
    5 MT=MT-1
      IF(RP(MT))6,6,7
    6 IF(MT-1)7,7,5
    7 JM1=MT
      T=0.
      DO 10 I=1,JM1
   10 T=T+RP(I)
      PRINT 1060
 1060 FORMAT(" IN-PLANE MISS DISTANCE PROBABILITIES")
      PRINT:(RP(I),I=1,' 1)
```

```
    5 MT=MT-1
      IF(RP(MT))6,6,7
    6 IF(MT-1)7,7,5
    7 JM1=MT
      T=0.
      DO 10 I=1,JM1
   10 T=T+RP(I)
      PRINT 1060
 1060 FORMAT(" IN-PLANE MISS DISTANCE PROBABILITIES")
      PRINT:(RP(I),I=1,JM1)
      PRINT 1070,T
 1070 FORMAT("CHECK SUM =",F10.6)
      PRINT 1001
 1001 FORMAT(1H1)
      GO TO 2
      END
      SUBROUTINE GROSS(N1M1,N2M1,N3M1,N4M1,C1,C2,C3,C4,
     +TP,RRP,JSAVE,M)
      COMMON DUMMY(78)
      COMMON X1(100),X2(100),X3(100),X4(100),Z(100)
      COMMON P1(100),P2(100),P3(100),P4(100),RP(100)
      DIMENSION Q(20)
      DO 10 I=2,M
   10 RP(I-1)=0.
      DO 130 I1=1,N1M1
      DO 120 I2=1,N2M1
      DO 110 I3=1,N3M1
      DO 100 I4=1,N4M1
      TPR=P1(I1)*P2(I2)*P3(I3)*P4(I4)
      JZ=0
      DO 20 IQ1=1,2
      IZ1=I1+IQ1-1
      DO 30 IQ2=1,2
      IZ2=I2+IQ2-1
      DO 40 IQ3=1,2
      IZ3=I3+IQ3-1
      DO 50 IQ4=1,2
      IZ4=I4+IQ4-1
      JZ=JZ+1
      Q(JZ)=ZZZ(C1,C2,C3,C4,X1(IZ1),X2(IZ2),X3(IZ3),X4(IZ4)
     +,TP,RRP)
   50 CONTINUE
   40 CONTINUE
   30 CONTINUE
   20 CONTINUE
      ZL=MIN1F(Q(1),Q(2),Q(3),Q(4),Q(5),Q(6),Q(7),Q(8),Q(9),
     +Q(10),Q(11),Q(12),Q(13),Q(14),Q(15),Q(16))
      ZH=MAX1F(Q(1),Q(2),Q(3),Q(4),Q(5),Q(6),Q(7),Q(8),Q(9),
     +Q(10),Q(11),Q(12),Q(13),Q(14),Q(15),Q(16))
      SP=ZH-ZL
   80 CALL DDDD(SP,TPR,ZL,ZH,JSAVE,M)
  100 CONTINUE
  110 CONTINUE
  120 CONTINUE
  130 CONTINUE
      RETURN
      END
      FUNCTION ZZZ(D1,D2,D3,D4,G1,G2,G3,G4,TP,RRP)
      T=G1*D1+G2*D2+G3/D3
      R=G4*D4
      ZZZ=SQRTF((T-TP)**2+(R-RRP)**2)
      RETURN
      END
```

FOLDOUT FRAME 2

SLIS.

```
      COMMON X(100),Y(100),Z(100),P(100),PP(100),RP(100)
      COMMON A(20),B(20),C(20),NN,YMEAN
      PRINT 1000
      PRINT 999
  999 FORMAT("1  ORBIT TYPE(1-6),ATMOS TYPE(1 OR 2)")
 1000 FORMAT(" M/CDA DISTRIBUTION PROGRAM")
      NN=4
      READ:IT,ITT
      A(1)=A(2)=A(3)=A(4)=600.
      B(1)=0.
      B(2)=66666.67
      B(3)=51333.33
      B(4)=46285.71
      GO TO (60,80),ITT
   60 C(1)=.01961519
      C(2)=-4.065703E-5
      C(3)=8.695401E-8
      C(4)=-1.711093E-10
      YMEAN=-.1833621
      GO TO 90
   80 C(1)=.0157079
      C(2)=-2.496592E-5
      C(3)=5.334886E-8
      C(4)=-1.370223E-10
      YMEAN=-3.913773
   90 GO TO (91,92,93,94,95,96),IT
   91 HA=10000.
      PA=1000.
      GO TO 100
   92 HA=10000.
      PA=500.
      GO TO 100
   93 HA=10000.
      PA=200.
      GO TO 100
   94 HA=20000.
      PA=1000.
      GO TO 100

   95 HA=20000.
      PA=500.
      GO TO 100
   96 HA=20000.
      PA=200.
  100 PRINT 998
  998 FORMAT("   APOAPSIS(KM)    PERIAPSIS(KM)")
      PRINT:HA,PA
    2 PRINT 1010
 1010 FORMAT(" NUMBER, PERIAPSIS VALUES")
      READ:N1,(X(I),I=1,N1)
      N1M1=N1-1
      PRINT 1015
 1015 FORMAT(" PERIAPSIS PROBABILITIES")
      READ:(P(I),I=1,N1M1)
      PRINT 1020
 1020 FORMAT(" NUMBER,VALUES FOR M/CDA")
      READ:M,(Z(I),I=1,M)
      PRINT 1030
 1030 FORMAT(" NUMBER,VALUES FOR TIME IN ORBIT")
      READ:N2,(Y(I),I=1,N2)
      PRINT 1040
 1040 FORMAT(" TIME IN ORBIT PROBABILITIES")
      N2M1=N2-1
```

FOLDOUT FRAME 1

```
1030 FORMAT(" NUMBER,VALUES FOR TIME IN ORBIT")
     READ:N2,(Y(I),I=1,N2)
     PRINT 1040
1040 FORMAT(" TIME IN ORBIT PROBABILITIES")
     N2M1=N2-1
     READ:(PP(I),I=1,N2M1)
     CALL MULPY(N1M1,N2M1,M,JSAVE,HA)
     MT=M
   5 MT=MT-1
     IF(RP(MT))6,6,7
   6 IF(MT-1)7,7,5
   7 JM1=MT
     T=0.
     DO 10 I=1,JM1

  10 T=T+RP(I)
     PRINT 1060
1060 FORMAT(" M/CDA DISTRIBUTION")
     PRINT:(RP(I),I=1,JM1)
     PRINT 1070,T
1070 FORMAT("CHECK SUM =",F10.6)
     PRINT 1001
1001 FORMAT(1H1)
     GO TO 2
     END
     SUBROUTINE MULPY(N1M1,N2M1,M,JSAVE,HA)
     COMMON X(100),Y(100),Z(100),P(100),PP(100),RP(100)
     COMMON A(20),B(20),C(20),NN,YMEAN
     DO 10 I=2,M
  10 RP(I-1)=0.
     DO 120 K=1,N1M1
     DO 100 I=1,N2M1
     T1=TOPH(X(K))
     T2=TOPH(X(K+1))
     Q1=Y(I)/(HA*T1)
     Q2=Y(I+1)/(HA*T1)
     Q3=Y(I)/(HA*T2)
     Q4=Y(I+1)/(HA*T2)
     ZL=MIN1F(Q1,Q2,Q3,Q4)
     ZH=MAX1F(Q1,Q2,Q3,Q4)
     SP=ZH-ZL
     TPR=P(K)*PP(I)
     IF(SP)15,80,20

  15 PRIN1 1500

1500 FORMAT("INTERVAL NEGATIVE")
     GO TO  200
  20 DO 30 JL=2,M
     IF(Z(JL)-ZL)30,30,25
  25 ZLOW=ZL
     JTEMP=JL
     GO TO 40
  30 CONTINUE
  40 CONTINUE
     DO 60 J=JTEMP,M
     IF(Z(J)-ZH)50,45,45
  45 JSAVE=J
     ZHIGH=ZH
     GO TO 70
  50 ZHIGH=Z(J)
     RP(J-1)=RP(J-1)+((ZHIGH-ZLOW)/SP)*TPR
     ZLOW=Z(J)
  60 CONTINUE
  70 RP(JSAVE-1)=RP(JSAVE-1)+((ZHIGH-ZLOW)/SP)*TPR
     GO TO 100
```

FOLDOUT FRAME 2

```
   25 ZLOW=ZL
      JTEMP=JL
      GO TO 40
   30 CONTINUE
   40 CONTINUE
      DO 60 J=JTEMP,M
      IF(Z(J)-ZH)50,45,45
   45 JSAVE=J
      ZHIGH=ZH
      GO TO 70
   50 ZHIGH=Z(J)
      RP(J-1)=RP(J-1)+((ZHIGH-ZLOW)/SP)*TPR
      ZLOW=Z(J)
   60 CONTINUE
   70 RP(JSAVE-1)=RP(JSAVE-1)+((ZHIGH-ZLOW)/SP)*TPR
      GO TO 100
   80 DO 90 J=2,M
      IF(ZL-Z(J-1))85,82,85
   82 RP(J-1)=RP(J-1)+TPR
      JSAVE=J
      GO TO 100
   85 IF(ZL-Z(J))82,90,90
   90 CONTINUE
      RP(M-1)=RP(M-1)+TPR
      JSAVE=M
  100 CONTINUE
  120 CONTINUE
  200 RETURN
      END
      FUNCTION TOPH(ZZ)
      COMMON X(100),Y(100),Z(100),P(100),PP(100),RP(100)
      COMMON A(20),B(20),C(20),NN,YMEAN
      DIMENSION POL(20)
      POL(1)=1.
      POL(2)=ZZ-A(1)
      MM=NN+1
      DO 20 I=3,MM

   20 POL(I)=ZZ*POL(I-1)-A(I-1)*POL(I-1)-B(I-1)*POL(I-2)
      TOPH=YMEAN
      DO 25 K=1,NN
   25 TOPH=TOPH+C(K)*POL(K+1)
      TOPH=EXPF(TOPH)
      RETURN
      END
```

REA

AGRID
1/16/67

$LIST

```
C234567
C      GRID - ROUTINE TO CALCULATE NEW PROB. TO FIT NORMAL GRID
       FUNCTION GRID(M,PROB,Q,A,B)
       DIMENSION Q(1),PROB(1)
        GRID = 0.
       DO 10 I=1,M
       IF(Q(I)-A) 10,10,5
     5 IS = I
       GO TO 20
    10 CONTINUE
    20 ATEMP = A
       DO 50 K=IS,M
       IF(Q(K)-B) 25,25,23
    23 KT = K
       GO TO 100
    25 GRID = GRID+((Q(K)-ATEMP)/(Q(K)-Q(K-1)))*PROB(K-1)
       ATEMP = Q(K)
    50 CONTINUE
    60 GO TO 500
   100 GRID = GRID+((B-ATEMP)/(Q(KT)-Q(KT-1)))*PROB(KT-1)
   500 RETURN
       END
C234567
C      MAR - ROUTINE TO CALCULATE MARGINAL PROBABILITIES
       SUBROUTINE MAR(MMAX,T,PTOP,INT,PMARG)
       DIMENSION PTOP(1),INT(1),PMARG(1)
       NMAXP = 101
       DO 20 I=1,NMAXP
    20 PMARG(I) = 0.0
       RAT = T/(1.-T)
       DO 40 K=1,MMAX
       N = INT(K)
       FN = N
       X = 0.0
       PR = (1.-T)**N
       PMARG(1) = PMARG(1)+PR*PTOP(K)
    25 IF(X-FN) 30,40,30
    30 PR = PR*(FN-X)*RAT/(X+1.)
       X = X+1.
       NX = X+1
       PMARG(NX) = PMARG(NX)+PR*PTOP(K)
       GO TO 25
    40 CONTINUE
       RETURN
       END
```

```
C  *****UPPER LIMIT PROGRAM********
      COMMON ALP(100),ULP(100),Z(100),P(100),ZOUT(100)
      COMMON POUT(100),GTP(100),PP(100),ZZ(100)
C FORMAT STATEMENTS*******
 1000 FORMAT("1 PROGRAM TO COMBINE UPPER LIMIT AND"
     +" M/CDA VARIABLES")
 1020 FORMAT("0 NUMBER,UPPER LIMIT POINTS")
 1040 FORMAT(" UPPER LIMIT PROBABILITIES")
 1060 FORMAT(" NUMBER,M/CDA POINTS")
 1080 FORMAT(" M/CDA PROBABILITIES")
 2000 FORMAT(" NUMBER,OUTPUT POINTS")
 2020 FORMAT("0      FIRST            LAST          PROBABILITY")
C*****************
      PRINT 1000
    1 CONTINUE
      PRINT 1020
      READ:NUL,(ALP(K),K=1,NUL)
      NLM1=NUL-1
      PRINT 1040
      READ:(ULP(K),K=1,NLM1)
      PRINT 1060
      READ:N,(Z(I),I=1,N)
      NM1=N-1
      PRINT 1080
      READ:(P(I),I=1,NM1)
      PRINT 2000
      READ:NOUT,(ZOUT(I),I=1,NOUT)
      NOM1=NOUT-1
      DO 3 I=1,NOM1
    3 POUT(I)=0.
C******************
      DO 200 K=2,NUL
      SUM=0.
      ZZ(1)=Z(1)
      IF(Z(1)-ALP(K))8,200,200
    8 DO 50 J=2,N
      IF(Z(J)-ALP(K))20,10,10
   10 ZZ(J)=ALP(K)
      DEN=Z(J)-Z(J-1)
      IF(DEN)15,12,15
   12 PP(J-1)=P(J-1)
      GO TO 18
   15 PP(J-1)=P(J-1)*((ALP(K)-Z(J-1))/DEN)
   18 NT=J
      NTM1=NT-1
      SUM=SUM+PP(J-1)
      GO TO 60
   20 PP(J-1)=P(J-1)
      ZZ(J)=Z(J)
      SUM=SUM+PP(J-1)
   50 CONTINUE
      ZZ(N+1)=ALP(K)
      NT=N+1
      NTM1=N
      PP(N)=0.
C*****************
   60 DO 70 I=1,NTM1
   70 PP(I)=PP(I)/
```

```
      SUM=SUM+PP(J-1)
   50 CONTINUE
      ZZ(N+1)=ALP(K)
      NT=N+1
      NTM1=N
      PP(N)=0.
C****************
   60 DO 70 I=1,NTM1
   70 PP(I)=PP(I)/SUM
      DO 80 I=1,NOM1
   80 GTP(I)=0.
      DO 90 I=1,NTM1
      SP=ZZ(I+1)-ZZ(I)
   90 CALL LOAD(SP,PP(I),ZZ(I),ZZ(I+1),NOUT,ZOUT,GTP)
      DO 100 I=1,NOM1
  100 POUT(I)=POUT(I)+GTP(I)*CLP(K-1)
  200 CONTINUE
C****************
      PRINT 2020
      DO 230 I=1,NOM1
  230 PRINT:ZOUT(I),ZOUT(I+1),POUT(I)
      GO TO 1
      END
C*****************
      SUBROUTINE LOAD(SP,TPR,ZL,ZH,M,Y,RP)
      DIMENSION Y(1),RP(1)
      IF(SP)100,80,20
   20 DO 30 JL=2,M
      IF(Y(JL)-ZL)30,30,25
   25 ZLOW=ZL
      JTEMP=JL
      GO TO 40
   30 CONTINUE
   40 CONTINUE
      DO 60 J=JTEMP,M
      IF(Y(J)-ZH)50,45,45
   45 JSAVE=J
      ZHIGH=ZH
      GO TO 70
   50 ZHIGH=Y(J)
      RP(J-1)=RP(J-1)+((ZHIGH-ZLOW)/SP)*TPR
      ZLOW=Y(J)
   60 CONTINUE
   70 RP(JSAVE-1)=RP(JSAVE-1)+((ZHIGH-ZLOW)/SP)*TPR
      GO TO 100
   80 DO 90 J=2,M
      IF(ZL-Y(J-1))85,82,85
   82 RP(J-1)=RP(J-1)+TPR
      JSAVE=J
      GO TO 100
   85 IF(ZL-Y(J))82,90,90
   90 CONTINUE
      RP(M-1)=RP(M-1)+TPR
      JSAVE=M
  100 CONTINUE
      RETURN
      END
```

FOLDOUT FRAME 2

READY.

TIME
1/16/67

$LIST

```
      COMMON X(100),Y(100),Z(100),P(100),PP(100),RP(100)
      COMMON A(20),B(20),C(20),NN,YMEAN
      PRINT 1000
      PRINT 999
  999 FORMAT("1 DEGREE, APOAPSIS, YMEAN")
 1000 FORMAT(" TIME IN MARS ORBIT PROGRAM")
      READ:NN,HA,YMEAN
      PRINT 998
  998 FORMAT(" ALPHA COEFFICIENTS")
      READ:(A(I),I=1,NN)
      PRINT 997
  997 FORMAT(" BETA COEFFICIENTS")
      READ:(B(I),I=1,NN)
      PRINT 996
  996 FORMAT(" LEAST SQUARE COEFFICIENTS")
      READ:(C(I),I=1,NN)
    2 PRINT 1010
 1010 FORMAT(" NUMBER, PERIAPSIS VALUES")
      READ:N1,(X(I),I=1,N1)
      N1M1=N1-1
      PRINT 1015
 1015 FORMAT(" PERIAPSIS PROBABILITIES")
      READ:(P(I),I=1,N1M1)
      PRINT 1020
 1020 FORMAT(" NUMBER, VALUES FOR TIME IN ORBIT")
      READ:M,(Z(I),I=1,M)
      PRINT 1030
 1030 FORMAT(" NUMBER, VALUES FOR DRAG PARAMETER")
      READ:N2,(Y(I),I=1,N2)
      PRINT 1040
 1040 FORMAT(" DRAG PARAMETER PROBABILITIES")
      N2M1=N2-1
      READ:(PP(I),I=1,N2M1)
      CALL MULPY(N1M1,N2M1,M,JSAVE,HA)
      MT=M
    5 MT=MT-1
      IF(RP(MT))6,6,7
    6 IF(MT-1)7,7,5
    7 JM1=MT
      T=0.
      DO 10 I=1,JM1
   10 T=T+RP(I)
      PRINT 1060
 1060 FORMAT(" TIME IN ORBIT DISTRIBUTION")
      PRINT:(RP(I),I=1,JM1)
      PRINT 1070,T
 1070 FORMAT("CHECK SUM =",F10.6)
      PRINT 1001
 1001 FORMAT(1H1)
      GO TO 2
      END
      SUBROUTINE MULPY(N1M1,N2M1,M,JSAVE,HA)
      COMMON X(100),Y(100),Z(100),P(100),PP(100),RP(100)
      COMMON A(20),B(20),C(20),NN,YMEAN
      DO 10 I=2,M
   10 RP(I-1)=0.
      DO 120 K=1,N1M1
      DO 100 I=1,N2M1
[illegible]
```

```
      SUBROUTINE ULF(N1M1,N2M1,M,JSAVE,HA)
      COMMON X(100),Y(100),Z(100),P(100),PP(100),RP(100)
      COMMON A(20),B(20),C(20),NN,YMEAN
      DO 10 I=2,M
   10 RP(I-1)=0.
      DO 120 K=1,N1M1
      DO 100 I=1,N2M1
      T1=TOPH(X(K))
      T2=TOPH(X(K+1))
      Q1=T1*HA*Y(I)
      Q2=T1*HA*Y(I+1)
      Q3=T2*HA*Y(I)
      Q4=T2*HA*Y(I+1)
      ZL=MIN1F(Q1,Q2,Q3,Q4)
      ZH=MAX1F(Q1,Q2,Q3,Q4)
      SP=ZH-ZL
      TPR=P(K)*PP(I)
      IF(SP)15,80,20
   15 PRINT 1500
 1500 FORMAT("INTERVAL NEGATIVE")
      GO TO 200
   20 DO 30 JL=2,M
      IF(Z(JL)-ZL)30,30,25
   25 ZLOW=ZL
      JTEMP=JL
      GO TO 40
   30 CONTINUE
   40 CONTINUE
      DO 60 J=JTEMP,M
      IF(Z(J)-ZH)50,45,45
   45 JSAVE=J
      ZHIGH=ZH
      GO TO 70
   50 ZHIGH=Z(J)
      RP(J-1)=RP(J-1)+((ZHIGH-ZLOW)/SP)*TPR
      ZLOW=Z(J)
   60 CONTINUE
   70 RP(JSAVE-1)=RP(JSAVE-1)+((ZHIGH-ZLOW)/SP)*TPR
      GO TO 100
   80 DO 90 J=2,M
      IF(ZL-Z(J-1))85,82,85
   82 RP(J-1)=RP(J-1)+TPR
      JSAVE=J
      GO TO 100
   85 IF(ZL-Z(J))82,90,90
   90 CONTINUE
      RP(M-1)=RP(M-1)+TPR
      JSAVE=M
  100 CONTINUE
  120 CONTINUE
  200 RETURN
      END
      FUNCTION TOPH(ZZ)
      COMMON X(100),Y(100),Z(100),P(100),PP(100),RP(100)
      COMMON A(20),B(20),C(20),NN,YMEAN
      DIMENSION POL(20)
      POL(1)=1.
      POL(2)=ZZ-A(1)
      MM=NN+1
      DO 20 I=3,MM
   20 POL(I)=ZZ*POL(I-1)-A(I-1)*POL(I-1)-B(I-1)*POL(I-2)
      TOPH=YMEAN
      DO 25 K=1,NN
   25 TOPH=TOPH+C(K)*POL(K+1)
      TOPH=EXPF(TOPH)
      RETURN
      END
```

FOLDOUT FRAME 2

```
      COMMON X(100),Y(100),Z(100),P(100),PP(100),RP(100)
      COMMON A(20),B(20),C(20),NN,YMEAN
      PRINT 1000
      PRINT 999
  999 FORMAT("1 DEGREE, APOAPSIS, YMEAN")
 1000 FORMAT(" M/CDA DISTRIBUTION PROGRAM")
      READ:NN,HA,YMEAN
      PRINT 998
  998 FORMAT(" ALPHA COEFFICIENTS")
      READ:(A(I),I=1,NN)
      PRINT 997
  997 FORMAT(" BETA COEFFICIENTS")
      READ:(B(I),I=1,NN)
      PRINT 996
  996 FORMAT(" LEAST SQUARE COEFFICIENTS")
      READ:(C(I),I=1,NN)
    2 PRINT 1010
 1010 FORMAT(" NUMBER, PERIAPSIS VALUES")
      READ:N1,(X(I),I=1,N1)
      N1M1=N1-1
      PRINT 1015
 1015 FORMAT(" PERIAPSIS PROBABILITIES")
      READ:(P(I),I=1,N1M1)
      PRINT 1020
 1020 FORMAT(" NUMBER,VALUES FOR M/CDA")
      READ:M,(Z(I),I=1,M)
      PRINT 1030

 1030 FORMAT(" NUMBER,VALUES FOR TIME IN ORBIT")
      READ:N2,(Y(I),I=1,N2)
      PRINT 1040
 1040 FORMAT(" TIME IN ORBIT PROBABILITIES")
      N2M1=N2-1
      READ:(PP(I),I=1,N2M1)
      CALL MULPY(N1M1,N2M1,M,JSAVE,HA)
      MT=M
    5 MT=MT-1
      IF(RP(MT))6,6,7
    6 IF(MT-1)7,7,5
    7 JM1=MT
      T=0.
      DO 10 I=1,JM1
   10 T=T+RP(I)
      PRINT 1060
 1060 FORMAT(" M/CDA DISTRIBUTION")
      PRINT:(RP(I),I=1,JM1)
      PRINT 1070,T
 1070 FORMAT("CHECK SUM =",F10.6)
      PRINT 1001
 1001 FORMAT(1H1)
      GO TO 2
      END
      SUBROUTINE MULPY(N1M1,N2M1,M,JSAVE,HA)
      COMMON X(100),Y(100),Z(100),P(100),PP(100),RP(100)
      COMMON A(20),B(20),C(20),NN,YMEAN
      DO 10 I=2,M
   10 RP(I-1)=0.
      DO 120 K=1,N1M1
      DO 100 I=1,N2M1

      T1=TOPH(X(K))
      T2=TOPH(X(K+1))
      Q1=Y(I)/(HA*T1)
      Q2=Y(I+1)/(HA*T1)
      Q3=Y(I)/(HA*T2)
      Q4=Y(I+1)/(HA*T2)
      ZL=MIN1F(Q1,Q2,Q3,Q4)
```

```
      T1=TOPH(X(K))
      T2=TOPH(X(K+1))
      Q1=Y(I)/(HA*T1)
      Q2=Y(I+1)/(HA*T1)
      Q3=Y(I)/(HA*T2)
      Q4=Y(I+1)/(HA*T2)
      ZL=MIN1F(Q1,Q2,Q3,Q4)
      ZH=MAX1F(Q1,Q2,Q3,Q4)
      SP=ZH-ZL
      TPR=P(K)*PP(I)
      IF(SP)15,80,20
   15 PRINT 1500
 1500 FORMAT("INTERVAL NEGATIVE")
      GO TO  200
   20 DO 30 JL=2,M
      IF(Z(JL)-ZL)30,30,25
   25 ZLOW=ZL
      JTEMP=JL
      GO TO 40
   30 CONTINUE
   40 CONTINUE
      DO 60 J=JTEMP,M
      IF(Z(J)-ZH)50,45,45
   45 JSAVE=J
      ZHIGH=ZH
      GO TO 70
   50 ZHIGH=Z(J)
      RP(J-1)=RP(J-1)+((ZHIGH-ZLOW)/SP)*TPR
      ZLOW=Z(J)
   60 CONTINUE
   70 RP(JSAVE-1)=RP(JSAVE-1)+((ZHIGH-ZLOW)/SP)*TPR
      GO TO 100
   80 DO 90 J=2,M
      IF(ZL-Z(J-1))85,82,85
   82 RP(J-1)=RP(J-1)+TPR
      JSAVE=J
      GO TO 100
   85 IF(ZL-Z(J))82,90,90
   90 CONTINUE
      RP(M-1)=RP(M-1)+TPR
      JSAVE=M
  100 CONTINUE
  120 CONTINUE
  200 RETURN
      END
      FUNCTION TOPH(ZZ)
      COMMON X(100),Y(100),Z(100),P(100),PP(100),RP(100)
      COMMON A(20),B(20),C(20),NN,YMEAN
      DIMENSION POL(20)
      POL(1)=1.
      POL(2)=ZZ-A(1)
      MM=NN+1
      DO 20 I=3,MM

   20 POL(I)=ZZ*POL(I-1)-A(I-1)*POL(I-1)-B(I-1)*POL(I-2)
      TOPH=YMEAN
      DO 25 K=1,NN

   25 TOPH=TOPH+C(K)*POL(K+1)
      TOPH=EXPF(TOPH)
      RETURN
      END
```

BOLDFACE FRAME 2

146

LI:

```
      COMMON X(100),Y(100),Z(100),P(100),PP(100),RP(100)
      COMMON A(20),B(20),NN,C
      PRINT 1000
      PRINT 999
  999 FORMAT("1 ORBIT TYPE(1-6)")
 1000 FORMAT(" PERIAPSIS DISTRIBUTION PROGRAM")
      READ:IT
      CALL OTYPE(IT)
      A(1)=A(1)/2.
      A(NN)=A(NN)/2.
      FFM=MM
      C=3.14159265/FFM
    2 PRINT 1010
 1010 FORMAT("1 NUMBER,ANGLE ATTACK VALUES")
      READ:N1,(X(I),I=1,N1)
      N1M1=N1-1
      PRINT 1015
 1015 FORMAT(" ANGLE PROBABILITIES, 90. TO 270.")
      READ:(P(I),I=1,N1M1)
      PRINT 1020
 1020 FORMAT(" NUMBER,VALUES FOR PERIAPSIS DECREMENT")
      READ:M,(Z(I),I=1,M)
      PRINT 1030
 1030 FORMAT(" NUMBER,VALUES FOR VELOCITY INCREMENT")
      READ:N2,(Y(I),I=1,N2)
      PRINT 1040
 1040 FORMAT(" VELOCITY INCREMENT PROBABILITIES")
      N2M1=N2-1
      READ:(PP(I),I=1,N2M1)
      CALL ADDER(N1M1,N2M1,M,JSAVE)
      MT=M
    5 MT=MT-1
      IF(RP(MT))6,6,7
    6 IF(MT-1)7,7,5
    7 JM1=MT
      T=0.
      DO 10 I=1,JM1
   10 T=T+RP(I)
      PRINT 1060
 1060 FORMAT("     PERIAPSIS DECREMENT DISTRIBUTION")
      PRINT:(RP(I),I=1,JM1)
      PRINT 1070,T
 1070 FORMAT("CHECK SUM =",F10.6)
      PRINT 1001
 1001 FORMAT(1H1)
      GO TO 2
      END
      SUBROUTINE ADDER(N1M1,N2M1,M,JSAVE)
      COMMON X(100),Y(100),Z(100),P(100),PP(100),RP(100)
      COMMON A(20),B(20),NN,C
      DO 10 I=2,M
   10 RP(I-1)=0.
      DO 120 K=1,N1M1
      DO 100 I=1,N2M1
      T1=PDV(X(K))
      T2=PDV(X(K+1))
```

FOLDOUT FRAME 1

```
   10 RP(I-1)=0.
      DO 120 K=1,N1M1
      DO 100 I=1,N2M1
      T1=PDV(X(K))
      T2=PDV(X(K+1))
      Q1=T1*Y(I)
      Q2=T1*Y(I+1)
      Q3=T2*Y(I)
      Q4=T2*Y(I+1)
      ZL=MIN1F(Q1,Q2,Q3,Q4)
      ZH=MAX1F(Q1,Q2,Q3,Q4)
      SP=ZH-ZL
      TPR=P(K)*PP(I)
      IF(SP)15,80,20
   15 PRINT 1500
 1500 FORMAT("INTERVAL NEGATIVE")
      GO TO 200
   20 DO 30 JL=2,M
      IF(Z(JL)-ZL)30,30,25
   25 ZLOW=ZL
      JTEMP=JL
      GO TO 40
   30 CONTINUE
   40 CONTINUE
      DO 60 J=JTEMP,M
      IF(Z(J)-ZH)50,45,45
   45 JSAVE=J
      ZHIGH=ZH
      GO TO 70
   50 ZHIGH=Z(J)
      RP(J-1)=RP(J-1)+((ZHIGH-ZLOW)/SP)*TPR
      ZLOW=Z(J)
   60 CONTINUE
   70 RP(JSAVE-1)=RP(JSAVE-1)+((ZHIGH-ZLOW)/SP)*TPR
      GO TO 100
   80 DO 90 J=2,M
      IF(ZL-Z(J-1))85,82,85
   82 RP(J-1)=RP(J-1)+TPR
      JSAVE=J
      GO TO 100
   85 IF(ZL-Z(J))82,90,90
   90 CONTINUE
      RP(M-1)=RP(M-1)+TPR
      JSAVE=M
  100 CONTINUE
  120 CONTINUE
  200 RETURN
      END
      FUNCTION PDV(ZZ)
      COMMON X(100),Y(100),Z(100),P(100),PP(100),RP(100)
      COMMON A(20),B(20),NN,C
      ZZ=(ZZ-90.)/10.
      PDV=0.
      DO 15 J=1,NN
      FJ=J-1
      G=C*FJ*ZZ
   15 PDV=PDV+A(J)*COSF(G)+B(J)*SINF(G)
      RETURN
      END
```

```
C     *****-PROGRAM TO FIND PROBABILITY OF COMBINATIONS*******
      COMMON X(100),Y(100),Z(100),P(100),PP(100),RP(100)
      PRINT 1000
      PRINT 1002
      PRINT 1004
 1000 FORMAT("1  PROGRAM TO COMBINE RANDOM VARIATES")
 1002 FORMAT("0 OPERATION CODE")
 1004 FORMAT(" ADD, 1"/" SUBTRACT, 2"/" MULTIPLY, 3"/
     +" DIVIDE, 4"/" RESTART, 5 OR GREATER")
 1006 FORMAT("0READ OPERATION CODE(1-4)")
    1 PRINT 1010
 1010 FORMAT("1 NUMBER,POINTS FOR FIRST DENSITY")
      READ:N1,(X(I),I=1,N1)
      N1M1=N1-1
      PRINT 1015
 1015 FORMAT("FIRST SET OF PROBABILITIES")
      READ:(P(I),I=1,N1M1)
      PRINT 1020
 1020 FORMAT("NUMBER,POINTS FOR RESULTING DENSITY")
      READ:M,(Z(I),I=1,M)
    2 PRINT 1006
      READ:ICODE
      IF(ICODE-5)3,1,1
    3 PRINT 1030
 1030 FORMAT("NUMBER,POINTS FOR NEXT DENSITY")
      READ:N2,(Y(I),I=1,N2)
      PRINT 1040
 1040 FORMAT("NEXT SET OF PROBABILITIES")
      N2M1=N2-1
      READ:(PP(I),I=1,N2M1)
      CALL ADDER(N1M1,N2M1,M,JSAVE,ICODE)
      MT=M
    5 MT=MT-1
      IF(RP(MT))6,6,7

    6 IF(MT-1)7,7,5
    7 JM1=MT
      T=0.
      DO 10 I=1,JM1
      T=T+RP(I)
   10 P(I)=RP(I)
      PRINT 1060
 1060 FORMAT("0  ******RESULTING PROBABILITIES******")
      PRINT:(RP(I),I=1,JM1)
      PRINT 1070,T
 1070 FORMAT("CHECK SUM =",F10.6)
      N1M1=JM1
      DO 20 I=1,M
   20 X(I)=Z(I)
      PRINT 1001
 1001 FORMAT(1H1)
      GO TO 2
      END
      SUBROUTINE ADDER(N1M1,N2M1,M,JSAVE,ICODE)
      COMMON X(100),Y(100),Z(100),P(100),PP(100),RP(100)
      DO 10 I=2,M
   10 RP(I-1)=0.
      DO 120 K=1,N1M1
```

```
      SUBROUTINE ADDER(N1M1,N2M1,M,JSAVE,ICODE)
      COMMON X(100),Y(100),Z(100),P(100),PP(100),RP(100)
      DO 10 I=2,M
   10 RP(I-1)=0.
      DO 120 K=1,N1M1
      DO 100 I=1,N2M1
      GO TO (11,12,13,14),ICODE
   11 Q1=X(K)+Y(I)
      Q2=X(K+1)+Y(I+1)
      Q3=X(K)+Y(I+1)
      Q4=X(K+1)+Y(I)

      GO TO 15
   12 Q1=X(K)-Y(I)

      Q2=X(K+1)-Y(I+1)
      Q3=X(K)-Y(I+1)
      Q4=X(K+1)-Y(I)

      GO TO 15
   13 Q1=X(K)*Y(I)
      Q2=X(K+1)*Y(I+1)
      Q3=X(K)*Y(I+1)
      Q4=X(K+1)*Y(I)
      GO TO 15
   14 Q1=X(K)/Y(I)
      Q2=X(K+1)/Y(I+1)
      Q3=X(K)/Y(I+1)
      Q4=X(K+1)/Y(I)
   15 ZL=MIN1F(Q1,Q2,Q3,Q4)
      ZH=MAX1F(Q1,Q2,Q3,Q4)
      SP=ZH-ZL
      TPR=P(K)*PP(I)
   20 DO 30 JL=2,M

      IF(Z(JL)-ZL)30,30,25
   25 ZLOW=ZL
      JTEMP=JL
      GO TO 40
   30 CONTINUE
   40 CONTINUE
      DO 60 J=JTEMP,M
      IF(Z(J)-ZH)50,45,45
   45 JSAVE=J
      ZHIGH=ZH
      GO TO 70
   50 ZHIGH=Z(J)
      RP(J-1)=RP(J-1)+((ZHIGH-ZLOW)/SP)*TPR
      ZLOW=Z(J)
   60 CONTINUE
   70 RP(JSAVE-1)=RP(JSAVE-1)+((ZHIGH-ZLOW)/SP)*TPR
      GO TO 100
   80 DO 90 J=2,M
      IF(ZL-Z(J-1))85,82,85
   82 RP(J-1)=RP(J-1)+TPR
      JSAVE=J
      GO TO 100
   85 IF(ZL-Z(J))82,90,90
   90 CONTINUE
      RP(M-1)=RP(M-1)+TPR
      JSAVE=M
  100 CONTINUE
  120 CONTINUE
  200 RETURN
      END
```

2

```
$LIST

C**********SPHERICAL PARTICLE PROGRAM************
      COMMON Z(6),X(100)
      PRINT 1000
      PRINT 1004
 1004 FORMAT("0  FUNCTION CODE")
      PRINT 1005
 1005 FORMAT(" 1, DIAMETER"/" 2, SURFACE AREA")
      PRINT 1006
 1006 FORMAT(" 3, DRAG PARAMETER"/" 4, VOLUME"/" 5, MASS")
      PRINT 1007
 1007 FORMAT(" 6, CROSS-SECTIONAL AREA")
 1000 FORMAT("1  PROGRAM TO COMPUTE PARTICLE PARAMETERS")
    1 CONTINUE
 1010 FORMAT(" NUMBER, POINTS")
      PRINT 1025
 1025 FORMAT("1 GIVEN FUNCTION CODE, DENSITY, DRAG")
      READ:I1,D,C
      PRINT 1010
      READ:NG,(X(I),I=1,NG)
   26 X(I)=Z(I)
      PRINT 1040
 1040 FORMAT("0  DIAMETER",5X,"SURFACE",7X,"DRAG",6X,"VOLUME"
     +,7X,"MASS",7X,"CROSS-SEC")
      DO 100 I=1,NG
      CALL HORNY(I1,I1,X(I),D,C,Z)
      PRINT 1020,(Z(J),J=1,6)
 1020 FORMAT(6(1PE12.4))
  100 CONTINUE
      GO TO 1
      END
      SUBROUTINE HORNY(INUTS,JNUTS,Y,D,C,Z)
```

```
 1010 FORMAT("0  DIAMETER",5X,"SURFACE",7X,"DRAG",6X,"VOLUME"
     +,7X,"MASS",7X,"CROSS-SEC")
      DO 100 I=1,NC
      CALL HORNY(I',I1,X(I),D,C,Z)
      PRINT 1020,(Z(J),J=1,6)
 1020 FORMAT(6(1PE12.4))
  100 CONTINUE
      GO TO 1
      END
      SUBROUTINE HORNY(INUTS,JNUTS,Y,D,C,Z)
      DIMENSION Z(6)
      IX=INUTS
      JX=JNUTS
      PI=3.14159265
      X=Y
    1 GO TO (10,20,30,40,50,60),IX
   10 X=PI*X*X
      Z(2)=X
   11 IX=IX+1
      IF(IX-7)18,15,18
   15 IX=1
   18 IF(IX-JX)1,80,1
   20 X=(2.*D*SQRTF(X/PI))/(3.*C)
      Z(3)=X
      GO TO 11
   30 X=(9.*PI/16.)*(C*X/D)**3
      Z(4)=X
      GO TO 11
   40 X=X*D
      Z(5)=X
      GO TO 11
   50 X=(PI/4.)*(6.*X/(PI*D))**(2./3.)
      Z(6)=X
      GO TO 11
   60 X=SQRTF(4.*X/PI)
      Z(1)=X
      GO TO 11
   80 CONTINUE
      RETURN
      END
```