

AFWL-TR-79-193

AD-E200607

① LEVEL III

AFWL-TR-
79-193

AD A092379



DDC FILE COPY

QUANTA - USERS MANUAL

1st Lt M. E. Badessa

July 1980

Final Report

Approved for public release; distribution unlimited.

DTIC
ELECTE
DEC 3 1980
S B D

THIS DOCUMENT IS UNCLASSIFIED.
THE COPY IS NOT TO BE REPRODUCED OR
SIGNIFICANTLY MODIFIED OR USED FOR
REPRODUCTION PURPOSES.

AIR FORCE WEAPONS LABORATORY
Air Force Systems Command
Kirtland Air Force Base, NM 87117

80 12 02 007

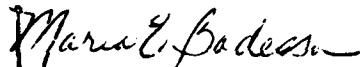
This final report was prepared by the Air Force Weapons Laboratory, Kirtland Air Force Base, New Mexico, under Job Order 88092220. First Lieutenant Maria Badessa (NTYV) was the Laboratory Project Officer-in-Charge.

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been authored by an employee of the United States Government. Accordingly, the United States Government retains a nonexclusive, royalty-free license to publish or reproduce the material contained herein, or allow others to do so, for the United States Government purposes.

This report has been reviewed by the Public Affairs Office and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.



MARIA E. BADESSA
1st Lt, USAF
Project Officer

FOR THE DIRECTOR



TERRY A. SCHMIDT
Major, USAF
Chief, Environment & Effects Branch



THOMAS W. CIAMBRONE
Colonel, USAF
Chief, Applied Physics Division

DO NOT RETURN THIS COPY. RETAIN OR DESTROY.

DISCLAIMER NOTICE

**THIS DOCUMENT IS BEST QUALITY
PRACTICABLE. THE COPY FURNISHED
TO DTIC CONTAINED A SIGNIFICANT
NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.**

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFWL-TR-79-193	2. GOVT ACCESSION NO. AD-A092 379	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) QUANTA - USERS MANUAL		5. TYPE OF REPORT & PERIOD COVERED Final Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) 1st Lt Maria E. Badessa		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Weapons Laboratory (NTYV) Kirtland Air Force Base, NM 87117		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62601F/88092220
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Weapons Laboratory (NTYV) Kirtland Air Force Base, NM 87117		12. REPORT DATE July 1980
		13. NUMBER OF PAGES 178
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Survivability Optimization Base Escape Nuclear Weapon Allocation Flush		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report provides instructions for using the QUANTA computer code, a code developed within the Applied Physics Division of the Air Force Weapons Laboratory to study the survivability of a flushing aircraft force from a nuclear attack of sea launched ballistic missiles. Included topics are structure of the input deck, description of the restart capabilities, details of array sizes, and a list of the error messages that might be encountered. The most current version of the QUANTA code is attached as Appendix A.		

DD FORM 1473
1 JAN 73

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

SUMMARY

The basic purpose of this report is to assist the QUANTA user in running the code efficiently. Included topics are structure of the input deck, description of the restart capabilities, details of array sizes, and a list of the error messages that may be encountered. It should be expected that occasional problems with the program will occur as the user attempts runs with new combinations of parameter values. If a problem is detected which cannot be solved through this documentation, notify the author or the Air Force Weapons Laboratory. Appendix A is a listing of the current version of QUANTA and Appendix B is the output of a run using the sample input data described in the text.

Accession For	
NTIS AD-81	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	23 CW

CONTENTS

INTRODUCTION	5
INPUT DATA	6
SAMPLE INPUT DATA	12
RESTART OPTION IN QUANTA	17
ARRAY DIMENSIONS	21
ERROR MESSAGES	24
APPENDIX A - QUANTA LISTING	29
APPENDIX B - QUANTA OUTPUT	162

INTRODUCTION

QUANTA is a computer model developed at the Air Force Weapons Laboratory (AFWL) to study the effects of a sea-launched ballistic missile (SLBM) nuclear attack on targets consisting of a flushing aircraft force. The program QUANTO (Ref. 1) was used as a basis for its mathematical development. Using the technique of Lagrangian multiplier optimization, a near-optimal allocation of SLBMs to targets is calculated. There are three main differences between these two codes. The first and most dramatic difference is that QUANTA simulates the base escape survivability in greater detail. QUANTO essentially attacks the circle about the turn point (not brake release point) where dispersed aircraft may possibly be at the time of weapon arrival. If aircraft are still on the runway or sitting at the brake release point, they still are considered to be evenly distributed within this circle. However, QUANTA attacks the target in two distinct positions: the runway and the circle centered at the turn point, thus giving a more accurate description of the probability of survival for flushing aircraft. The second difference is that QUANTA allows the attacker the option to allocate weapons to the nonalert aircraft at a base by assigning a value to that target. The third is that QUANTA does not allocate weapons which do not attain a certain value of aircraft killed. This value is user specified and will be discussed later in greater detail in the description of input.

This document contains the information needed to construct the input data for QUANTA and make a successful run. In addition, procedures for running with a restart capability are discussed. Notes on array dimensions and problem sizes are also included.

-
1. QUANTO--A CODE TO OPTIMIZE WEAPON ALLOCATIONS, AFWL-TR-73-242, Kirtland Air Force Base, New Mex., 1974.

INPUT DATA

The input is read by the subroutines STARTR, READER, PROCES, MARVIN, and ALOCIN in that order. A brief description of the variables that are read and their format will follow. The structure of the data deck is divided into the following categories:

- Initial data card
- Beddown data
- Aircraft parameters
- Submarine data
- Missile parameters and profile
- Aircraft profile
- Cutoff values
- Convergence parameters
- Initial allocation

STARTR will read the initial data card. Variables which determine the sizes of the arrays in QUANTA are on this card.

- Initial data card

Format: 815

Variables:

- NTGTS - number of targets/air bases
- NSUBS - number of submarine locations
- NTYPES - number of types of aircraft
- MXRWAY - maximum number of runways found on a given base
- MTYPES - number of types of missiles
- IOUT - If it equals 1, standard printout. Additional reallocation information will be printed, if it equals 2.
- NACTT - total number of aircraft in scenario
- NMARV - number of cutoff values. If all weapons in the scenario are to be allocated, NMARV must equal 0.

READER will read the beddown data, aircraft parameters, submarine data, and missile parameters and profile. The aircraft beddown data is input as a set of cards for each base. The first card(s) of the set identifies the base, with the number of alert aircraft at that target and is followed by a takeoff sequence list for the alert aircraft. The number of these sets must equal the value of

NTGTS, read by STARTR. The aircraft parameters such as hardness and brake release time will be read from one card per aircraft type. However, one parameter, the distance to the turn point, will be input for all aircraft types preceding aircraft data. The time intervals for takeoff sequences consist of a set of cards containing NTYPES variables. Submarine data will be read from one card per submarine location. The total number of cards must equal NSUBS (which was read from initial data card). The missile parameters and profile is input as a set of cards for each missile type. On the first card of the set will be the missile parameters such as yield and reliability. Following this card will be the missile profile.

- Beddown data

Format: 3F10.4, I5, 5X, 3F10.4/(7F10.4)

Variables:

TGTLAT - target latitude in decimal degrees

TGTLNG - target longitude in degrees

BASVAL - value of the base which also could include a value for nonalert aircraft

NRWAYS - number of runways on base

VALj - number of aircraft of type j on base, j=1 to NTYPES

Format: 7011

Variables:

ISEQ - takeoff sequence of aircraft identified by type number

- Aircraft parameters

Format: 7F10.6

Variables:

DTCENTj - the minimum distance from brake release (start of takeoff roll) to the turn point of aircraft type j, j=1 to NTYPES.

Format: 4F10.4

Variables:

RELVAL - relative value of aircraft

BRTIME - the time, in minutes, between launch of the first SLBMs and the start of takeoff roll of the aircraft. (Assume it is the first to take off.)

PSI - overpressure hardness in lb/in²

CAL - vulnerability level to free-field thermal energy in cal/cm²

Format: 7F10.4

Variables:

DELTA_{i,j,k} - takeoff time intervals in minutes between each ordered pair of aircraft. Type k will follow type j. After all possible combinations are read for single runways, then input dual, triple, etc. i=1 to MXRWAY, j=1 to NTPES, k=1 to NTPES. The last integer increments most often; e.g., (1,1,1), (1,1,2), ..., (1,1,NTPES), (1,2,1), ..., (1,2,NTPES), etc.

- Submarine data

Format: 2F10.4, 3I5

Variables:

SUBLAT - submarine latitude in degrees

SUBLNG - submarine longitude in degrees

ISUBS - number of submarines at this location

NMPS - number of missiles per submarine. (This number will automatically be the same for each sub at this location.)

MTYPE - missile type by number

- Missile parameters and profile

Format: 7F10.4

Variables:

DELT_M - launch interval between salvos in minutes

REL - overall reliability of the missile. This includes launch, in-flight, and detonation.

RNGMIN - minimum range in nautical miles

RNGMAX - maximum range in nautical miles

YIELD - warhead yield in kilotons

BURST - height of burst in feet aboveground

RVS - number of reentry vehicles per missile

Format: I5, 5X, 6F10.4/(6F10.4)

Variables:

M_{PROF} - number of data points in missile profile

F_{TIME}, F_{M_{RNG}} - time in minutes, distance in nautical miles of missile trajectory

PROCES will read the aircraft profile. A set of cards for each aircraft type consists of the following:

- Altitude
- Count

- Flight profile data - measured relative to the brake release for a nonturning aircraft.
- Additional parameters - needed to construct an extended flight profile in ground range and time.
- Environments - must be the same for all types of aircraft.
- Aircraft profile
 - Altitude
 - Format: F10.8
 - Variables:
 - AZ - final level-off altitude in feet above the ground. It must be less than or equal to the maximum (last) altitude in the flight profile.
 - Count
 - Format: I5
 - Variables:
 - NDATA - number of data points in aircraft profile
 - Flight profile data
 - Format: 3F15.8, 2F10.6
 - Variables:
 - F - ground range in feet
 - G - time in seconds
 - A - altitude in feet which is nondecreasing
 - VEL - Mach number describing ground speed
 - ACCEL - level-off acceleration in feet per second squared if the aircraft were to level-off at the altitude specified on this card.

NOTE: Accurate values for distance, time and altitude are important because of the method QUANTA uses to compute kill regions for an aircraft. At the time of weapon arrival, the altitude and distance from brake release of the aircraft are used to calculate the probability of kill. However, the values for VEL and ACCEL are used to accelerate the aircraft from its velocity at level-off altitude to its final escape velocity. The standard equations of motion use these parameters associated with level-off only to determine time-distance pairs for the flight profile. Therefore, it would suffice if every number in VEL and ACCEL arrays were equal to the velocity and acceleration of the aircraft at level-off altitude.

- Additional parameters
 - Format: 3F15.8, F10.6
 - Variables:
 - ALTFM - minimum altitude in feet at which the aircraft attains final Mach

FMACH - final (maximum) Mach number

TINT - time interval, in seconds, used to extend the profile after the aircraft has reached its final escape velocity (60 s is presently used).

XATI - number of data points (in time and distance) that are calculated for the flight profile from the level-off velocity to final escape velocity (10 is presently used).

NOTE: The aircraft is assumed to follow the input flight profile until the level-off altitude is reached. QUANTA then accelerates the aircraft to its final escape velocity which is assumed to be the maximum velocity and is specified as FMACH. However, this cannot occur if XATI is equal to zero or if ALTFM is less than the level-off altitude. If XATI equals zero (i.e., there are no data points between level-off and escape velocity), no acceleration takes place after level-off and the velocity at level-off will define the final escape velocity. If ALTFM is less than the level-off altitude, then there is no acceleration since the aircraft should have reached its final mach in the input profile. After the aircraft has reached final escape, TINT and FMACH are used to determine time-distance pairs. The total number of data points in the extended profile is NXAP.

- Environments

Format: 4F10.8

Variables:

HTE - terrain height

RHO - the albedo factor, which is the fraction (between 0.0 and 1.0) of incident thermal energy which reflects off the ground. For a SAC normal day, the albedo is 0.3. A worst case abnormal day has a higher albedo.

VIS - the visibility in miles; 10 for a SAC normal day

PZ - the water vapor in the air, in millimeters; 5 for a SAC normal day

MARVIN will read the cutoff values. SLBMs which arrive at a target sufficiently late may not produce adequate kill to justify allocation. These SLBMs may be the last (or next to the last) weapons launched. This is a result of the aircraft having enough time to disperse into a larger area. The program will check if a weapon can attain a kill at least as large as the cutoff value. If it does not, that weapon will not be allocated to any real targets but to a dummy target. After an optimum allocation has been reached using this cutoff value, the program will use it as an initial allocation for the next run with a different cutoff value. For this reason, the cutoff values should be in descending order; i.e., highest value read first. This enables the program to be run more efficiently saving the user cost and time. If the cutoff is as low as 0.1,

almost all weapons will be allocated. If it is near 10.0, as little as one-tenth of the weapons will be allocated depending on the scenario.

- Cutoff values

Format: 7F10.5

Variables:

CUT_i - the slope of the curve defining the relationship between the number of weapons allocated and the probability of kill, $i=1$ to NMARV.

ALOCIN will read the convergence parameters and the initial allocation. There are three methods by which the iteration process will stop: (1) if the increase in kill becomes insignificant, (2) if the limit for the total number of iterations has been reached, or (3) if the difference in the Lagrangian multipliers becomes relatively small. These criteria are user specified. The initial allocation of missiles to targets is input submarine by submarine. The subs are in the order of the submarine locations with all subs at one location preceding the first sub at the next location.

- Convergence parameters

Format: F10.4, 2I5, F10.4

Variables:

CHGKIL - the value of aircraft killed used as a measure for a significant increase in kill in ITCUT2 iterations, i.e., if in the last ITCUT2 iterations the value killed has not increased by at least CHGKIL, the iterative process will stop.

ITCUT1 - the total number of ITCUT2 iterations in the optimization process, i.e., QUANTA, if not stopped by some other means, will continue through ITCUT1 x ITCUT2 number of iterations.

ITCUT2 - after this number of iterations, QUANTA checks for a significant increase in value killed.

EPSCUT - for an optimum, the Lagrangian multipliers must all be equal. Since this is impossible, a test for convergence is if they are within a tolerance of EPSCUT.

- Initial allocation

Format: 14I5

Variables:

ITGINO - the target number to which the SLBM is allocated. The SLBMs from a submarine will appear in ascending order by salvo. If there are more than 14 salvos on a sub, then continue the allocation on the next card. Begin reading the allocation of the next sub with a new card. If the cutoff value is large, the initial allocation should read - 1 which represents the dummy target.

SAMPLE INPUT DATA

To assist the reader in developing an input deck, the following pages detail an example of the input data for QUANTA. The first row of numbers on each page of Table 1 are to aid the reader in locating the columns for the variables, while the first number on every line indicates the card number. The first card of input tells that there are four targets, three sub locations, two aircraft types, a maximum of two runways, two missile types, standard printout, 20 aircraft, and one marval run.

The next eight cards identify the four targets. The first base, located somewhere in the Pacific, has a value of 15 for either its importance as a target or for nonalert aircraft. This is to insure that weapons will be drawn down. Aircraft type one precedes aircraft type two in the takeoff sequence on every base. The only target that has two runways is target three.

The following cards describe the two aircraft types. According to the tenth card, the first aircraft has the capability to make its flyout turn 3.5 nmi from brake release. The following two cards identify the aircraft types with their relative value, brake-release time, and overpressure and thermal hardnesses. According to cards 13 and 14, aircraft type one following type one has a take-off interval of 0.25 min. If dual runways exist, the time interval is cut in half.

The following group of nine cards describes submarine and missile parameters. Cards 15 through 17 state the three submarine locations along with the number of missiles on each sub at that location. There are three subs at location two, while locations one and three have only two subs. These cards also show a total of 36 missiles of type one and 56 missiles of type two in this scenario. However, there are 40 weapon groups (NWPNS described later). A weapon group consists of weapons that have the same characteristics, i.e., they are launched at the same time, from the same location, and have the same parameters. The next six cards identify missile types one and two with weapon reliability, yield, height-of-burst and number of RVs. A total of four data points are used to describe the trajectory for each missile.

The next group of 20 cards details the flight profile for aircraft type one. Card number 24 says that this aircraft reaches level-off altitude of 10,000 ft., but the flight profile indicates that it reaches a maximum Mach of 0.9 at a

TABLE 1. SAMPLE DATA DECK

	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890
1.	4	3	2	2	2	1	20 1
2.	13.5833	144.9167	15.0		1	2.0	1.0
3.	112						
4.	44.8	68.8333	2.0		1	1.0	3.0
5.	1222						
6.	33.9167	84.5167			2	4.0	5.0
7.	111122222						
8.	33.4333	112.1667	10.0		1	4.0	
9.	1111						
10.	3.5	5.5					
11.	2.5	3.5	1.0		96.0		
12.	1.0	4.2	1.5		120.0		
13.	0.25	0.5	0.5		0.5	0.125	0.25 0.25
14.	0.25						
15.	36.75	74.0	2	18	1		
16.	28.05	93.45	3	12	2		
17.	45.65	126.05	2	10	2		
18.	0.2	0.9	250.0	2250.0	1000.0	7000.0	2.0
19.	4	3.5	250.0	5.5	950.0	7.5	1600.0
20.	9.5	2250.0					
21.	0.25	0.85	650.0	2000.0	800.0	6000.0	3.0
22.	4	5.0	650.0	7.5	1100.0	10.0	1550.0
23.	12.5	2000.0					
24.	10000.0						
25.	16	PROFILE 1					
26.		0.0	0.0		0.0	0.0	
27.		1500.0	7.5		5.0	0.13	
28.		3700.0	19.0		7.0	0.17	
29.		5700.0	30.0		16.0	0.20	
30.		9000.0	48.0		75.0	0.25	
31.		13000.0	63.0		230.0	0.34	
32.		20000.0	79.5		250.0	0.45	
33.		33000.0	105.0		500.0	0.60	

AFWL-TR-79-193

TABLE 1. CONCLUDED.

	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u>	<u>0</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u>	<u>0</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u>	<u>0</u>	
67.		4	4	4	3	3	3	2	2	2	1	1	1																		
68.		1	0	2	0	3	0	4	0	-1	-1																				
69.		1	0	2	0	3	0	4	0	-1	-1																				

lower altitude of 2,800 ft. This is stated on cards 36 and 42. For this reason, the acceleration is zero on all profile cards and on card 42 XATI is also zero. QUANTA will simply extend the profile using FMACH and TINT (card 42) to find time-distance pairs. Cards 44 through 58 describe the flight profile for aircraft type two. The first card of this group shows a level-off altitude of 5,000 ft. After it reaches this altitude it will accelerate at 6.2 ft/s^2 (card number 56) until it reaches a final mach of 0.849 at an altitude of 7,600 ft (card number 57). A total of 10 data points will identify this interval in the flight profile (XATI on card 57). At 60 s intervals, distance will be computed by QUANTA to complete the profile. Please note that cards 43 and 58 have the same values for the environments.

For this one marval run, the cutoff value on card 59 is 0.5. The next card, which is the convergence parameters card, states the three conditions which may terminate the iterative process: (1) after every 100 iterations, QUANTA will check for an increase of kill of at least 0.1 value points, (2) there will be no more than 1,000 iterations, and (3) if the difference between the largest and smallest Lagrangian multiplier is less than 0.01, an optimum solution has been reached.

The last group of cards which describes the initial allocation matrix consists of a total of nine cards. The first seven missiles from each sub at location one will be allocated to real targets (cards 61 and 63). The rest of the weapons will go to the dummy target since the user feels they will not attain enough kill. All of the missiles at sub location two will be allocated while some at the third sub location will not. This is shown by the zeros appearing on the last two cards. This is a mathematical representation of missiles being used for other time urgent targets such as communications.

RESTART OPTION IN QUANTA

The restart option, pertinent for CDC systems, permits the QUANTA user to resume calculations for a particular scenario without the loss of an excessive amount of time. For example, if the computer should crash after the program has been executing for 20 minutes, all is not lost. With a restart tape, the user can recover the computations close to the point of interruption.

There are three methods by which QUANTA writes a restart tape. First, if sense switch one is turned on (controlled by the operator), the program will write the essential variables onto the restart file. In the example mentioned above, the operator would turn the switch on before the computer crashed, thereby enabling QUANTA to write the tape and stop execution gracefully. The second method will also terminate the problem for insufficient time. After ITCUT2 iterations have been completed in the optimization process, QUANTA verifies that there is ample time for another ITCUT2 iteration. Third, during each run, the restart file will be automatically written after every fifth ITCUT1 iteration (i.e., 5 x ITCUT2, 10 x ITCUT2, etc.). This is to insure that the user can still obtain a tape even if something disastrous should happen. If there is any doubt the program will not be completed in the time allotted, the user should save the restart tape.

If the user chooses to save the file, it must be specified in the control language during the initial run. The local file name of the tape is TAPE13 and should be referenced as such throughout the initial run and all subsequent runs. The QUANTA user has the option of storing the restart file on magnetic tape or disk. When starting with a restart tape that is on permanent file, sense switch two must be turned on. If the sense switch is on and the file does not exist, QUANTA will abort. A sequence of commands appearing in the control cards for obtaining a permanent file on an initial run is:

```
REQUEST (TAPE13, *PF)
COPY (ZILCH, TAPE13)
FTN (I = COMPILE, S, S = PFMTEXT, . . .)
.
.
.
CATALOG (TAPE13, FILENAME, ID = . . .)
```

and for that restart run,

```
ATTACH (TAPE13, FILENAME, ID = . . .)
SWITCH(2)
FTN (I = COMPILE, S, S = PFMTEXT, . . .)
.
.
.
CATALOG (TAPE13, FILENAME, ID = . . .)
```

The first two cards in the above restart run replace the first two cards of the initial run. During all QUANTA runs, the FTN card requires the S, S = PFMTEXT parameters for compilation. The user may choose to catalogue TAPE13 on all subsequent runs.

At this point it becomes very apparent that the restart capability in QUANTA is machine dependent in its language. In particular, the ALTER and EXTEND routines are in job control language for the CDC computers. Before attempting to use the program on a computer other than the CDC, consult computer manuals for equivalent commands.

On a restart run, data are read just as in the initial run. QUANTA will use these data in the beginning of the program to set up arrays and compute the survivability matrix which for storage purposes is too large to be put onto the tape. When QUANTA begins the reallocation process (for finding the optimal laydown), it will check a restart file, if one exists. The information from that file will be used as a closer approximation to the optimal solution. Figure 1 demonstrates the above procedures.

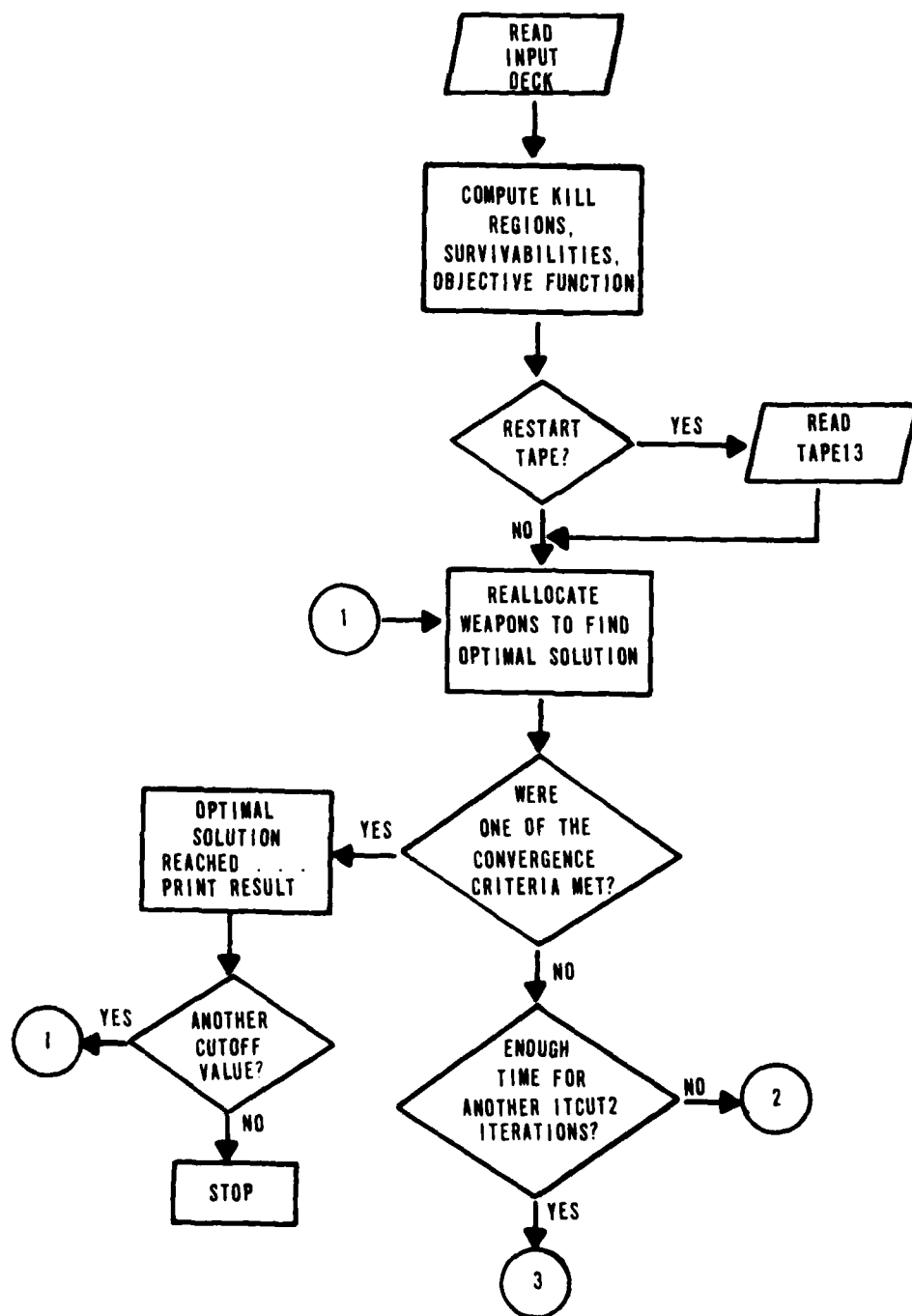


Figure 1. Flow chart.

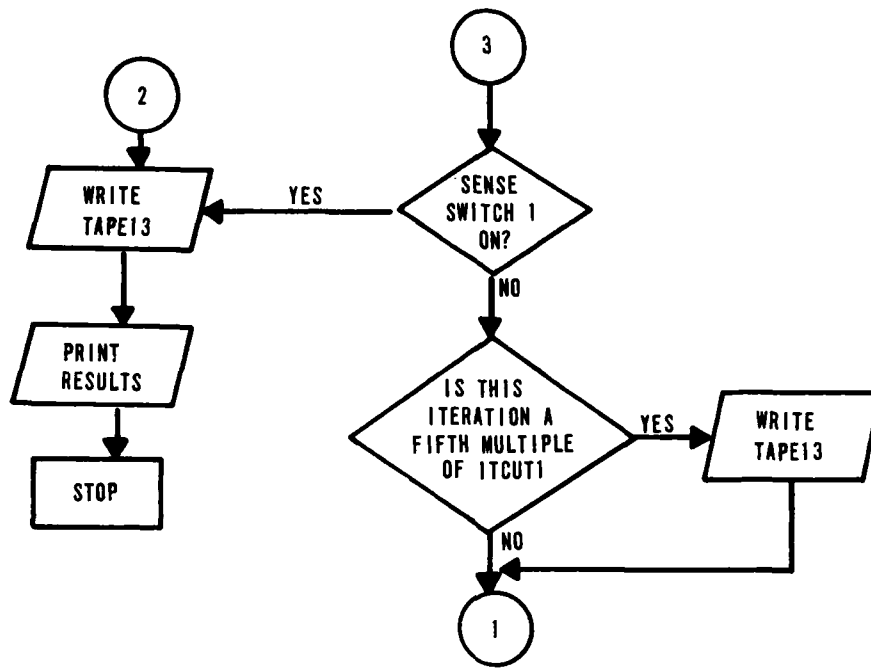


Figure 1. Concluded.

ARRAY DIMENSIONS

This section is to give the reader a better understanding of the program itself and to delineate the variables within the code that must be changed with problem size. Internally, QUANTA sets up four large arrays. In the main routine, they are named WORK1, WORK2, ALOC, and FLAMB.

The WORK1 array comprises a number of small arrays which have been read from input and will be utilized throughout the execution of the program. One example is the array containing the relative value of the aircraft. This variable is essential in the calculation of the total probability of kill which is continuously recomputed during the iterative process.* There are 12 such arrays in the WORK1 array and its size is determined by the subroutines WORK1 and REMW1. The formula for the size is given below.

After QUANTA reads and verifies all the input, it begins calculations for the survivability of aircraft at a target against all weapons. Since the size of this matrix can become extremely large, these computations are stored onto tape (TAPE4) for mass storage (machine dependent). After this is completed, most of the variables, such as target and submarine locations, needed for these calculations are no longer required. Therefore, in order to save core, the WORK2 array, originally storing such variables, is then used for computations which define the objective function. For this reason, two messages are printed onto output for the required WORK2 array size. The first message tells how much storage is needed for the input variables, while the second tells the amount of core needed to compute the probability of kill. The subroutines WORK2, REMW2, and SECW2 calculate the size and the formulas for both which are given below. The user should make the size of the WORK2 array at least as large as the maximum of the two.

The size is the same for both the ALOC and FLAMB arrays. The ALOC array stores the allocation matrix and the FLAMB array is used for the Lagrangian multiplier matrix. Their dimension should be set to $3 \times \text{NWPNS} \times \text{NTGTS}$, where NTGTS is the number of targets and NWPNS is the number of weapon groups, not total number of weapons. To compute NWPNS add the number of missiles (described as NMPS in input) at each submarine location (not each sub at the location).

*Through this reallocation process, QUANTA finds an optimal missile laydown by moving weapons from target to target.

If the user has not provided adequate core for all these arrays, an error message will be printed and execution stopped. The program does this check by comparing the values of NLCM1, NLCM2, and NLCM3, with the required dimension sizes. A data statement in the main program defining NLCM1, NLCM2, and NLCM3 should be set to the number of words in WORK1, WORK2, and ALOC, respectively. Stated below are the variables which determine the dimensions of the arrays. Most of them are read from the initial card of input.

NLCM1 = NTGTS (number of targets)
 + (NTGTS + 1) x NTYPES (types of aircraft)
 + 3 x NSUBS (number of sub locations)
 + MTYPES (types of missiles)
 + NACT (number of aircraft)
 + NMARV (number of marval runs)
 + NWPNS (total number of weapon groups)
 + 2 x NWPNS x MXNAPB (maximum number of aircraft found
 at a given target)

NLCM2 = 3 x NTGTS
 + 5 x NTYPES
 + NTYPES x NTYPES x MXRWAY (maximum runways on a given
 target)
 + 63 x NTYPES x MTYPES
 + 7 x MTYPES
 + 2 x NSUBS (number of sub locations)
 + 2 x NTYPES x NXAP (number of data points in the extended
 aircraft profile)
 + 6 x NTYPES x NPAP (maximum of data points in the aircraft
 profile)
 + 2 x MTYPES x MXMPR (maximum points in missile trajectory)
 + 2 x MXNAPB
 + MXNMPS (maximum of the number of missiles per sub - NMPS)
 + 30

NLCM2* = NACT
 + 2 x NTGTS
 + 3 x NTYPES
 + MTYPES
 + 4 x MXNAPB

*The value defined in the program should be the maximum of the two NLCM2 formulas.

$$NLCM3 = 3 \times NTGTS \times NWPNS$$

There is a data statement in the main routine which may affect the size of WORK2. The variables NPAP, NXAP, and MXMPR are in the equation for the first NLCM2 above. NPAP is the maximum of the number of data points (described in input as NDATA) of all the aircraft profiles. NXAP is the desired number of time-distance pairs in the extended profile for all aircraft types. MXMPR is the maximum of the number of pairs (MPROF in input) in the missile trajectories. These values should be appropriately defined in the DATA statement.

There is another array in the main program that must change with problem size. INDX sets up the index for reading and writing the large survivability matrix on mass storage. Its dimension must be at least $2 \times NTGTS + 1$. A data statement defining NDX must also be set to this value. As with the WORK arrays, if the size is incorrect, an error message will be printed indicating the problem.

Outside of the main program, in particular in four subroutines, there are arrays which must be dimensioned properly to insure complete execution. In subroutines SETS, OUTRNG, and ZERO an array BASENO must be the size of NTGTS. The data statement which checks its correct size is in subroutine ZERO. In subroutine BOSS arrays FN, ST, and STT must be at least as large as NMARV (the number of marval runs). The data statement initializing these arrays must also be changed.

With each new problem, the user must insure that the dimensions of all these arrays are adequate for storage. With each array there is an integer that must equal the value of its dimension. The purpose is to allow QUANTA's built-in check system to verify sufficient core for problem size.

ERROR MESSAGES

In order to minimize the number of errors that could occur in such a large code as QUANTA, a series of checks were developed for the input data. However, these checks do not enable the program to sense when a typographical error has been made by the user. For example, it cannot determine that the aircraft overpressure hardness was meant to read 1.0 when in actuality it reads 10. For this reason, the program will print the variables as they are read so that the user may verify the quantities. QUANTA also checks the variables when read but in a general sense. For example, it can detect if the aircraft vulnerability levels and weapon yields are positive values.

Table 2 lists all the error messages QUANTA could issue and the remedies for these errors. When an error is detected, the program will stop execution immediately following a message printed in the day file and at the end of output. The latter message will identify the subroutine along with an error number and a brief description of the problem.

TABLE 2. ERROR DESCRIPTION

<u>Subroutine/Error No.</u>	<u>Message</u>	<u>What To Do</u>
REMW1		
1	Size of WORK1 array exceeds dimension.	The dimension of the WORK1 array is too small for your problem size. Change its dimension to what the required size reads. Also change NLCM1 in the data statement.
SECW2		
1	Size of WORK2 array exceeds dimension.	The dimension of the WORK2 array is too small for your problem size. Change its dimension to what the required size reads. Also change NLCM2 in the data statement.
REMW2		
1	Size of WORK2 array exceeds dimension.	Same as SECW2-1.
STARTR		
1-5	Variable read must be greater than 0.	Change the first card of input to read nonnegative numbers.
6	IOUT must be either 1 or 2.	Change the sixth variable on the first card of input to comply with message.
7-8	Variable read must be greater than 0.	Change NACT or NMARV to comply.
READER		
1	Total number of aircraft is greater than original input.	The number of aircraft that was read in is less than the total number added from each base. Check the number of aircraft at each base. If that is correct, then change the first card (NACT). Otherwise, change the number at the base which is incorrect.

TABLE 2. CONTINUED.

Subroutine/Error No.	Message	What To Do
READER (cont.)		
2	Base latitude and longitude must be greater than 0.	Change the variables so that they are positive.
3	Number of runways on a base must be greater than 0 and not greater than the maximum number of runways.	Check to make sure that all bases have the correct number of runways. If it is not right, change that particular card. Otherwise, change the first card of input (MXRWAY) to reflect the maximum of the number of runways on a base.
4	Number of aircraft of a given type cannot be less than 0.	Change the variable VAL so that it is nonnegative.
5	An aircraft type on a base must be greater than 0 and not greater than the number of types.	Change the takeoff sequence ISEQ so that the values read will be from 1 to NTYPES (which is read on the first card).
6-13	Variables read must be greater than 0.	Change the variables so that they are positive.
14	The type of sub must be greater than 0 and not greater than the total.	Change the missile type of a sub location so that its value is between 1 and MTYPES (which is read on the first card of input); or, change MTYPES to comply.
15	The number of time/distance pairs cannot be greater than MXMPR.	MXMPR is a data statement in the main program which is the maximum of the number of pairs in the missile profile. Check that number against the profiles of each type. Change the appropriate value.
16	The time interval between salvo launches must be greater than 0.	Change DELTM so that it is positive.

TABLE 2. CONTINUED.

<u>Subroutine/Error No.</u>		<u>Message</u>	<u>What To Do</u>
READER (cont.)			
17		The reliability of a missile must be greater than 0 and less than 1.	Change the value of REL to comply with message.
18		The maximum range of the missile must be greater than its minimum range.	Change the maximum range RNGMAX or the minimum range RNGMIN to comply.
19-23		The variables read must be greater than 0.	Change the variables to comply with messages.
PROCES			
1		The level-off altitude of the aircraft must be greater than 0.	Change AZ to a positive value.
2		The number of data points in aircraft profile is larger than NPAP.	NPAP is in a data statement in the main program. It is the maximum of the number of points in the aircraft profile. Check it against all aircraft types. Change the appropriate value.
3-5		Variables read cannot be less than 0.	Change the first card in the profile so that the variables are nonnegative.
6-7		The ground range and time must be strictly increasing.	Check that each variable read on consecutive cards is increasing in value. Change those that are not.
8-11		Variables read must be positive.	Change the additional parameters card so that the variables comply with messages.
12-15		Variables read must be positive.	Change the variables on the environments card so that they meet the requirements specified in the error message.

TABLE 2. CONCLUDED.

<u>Subroutine/Error No.</u>	<u>Message</u>	<u>What To Do</u>
PROCES (cont.)		
16	Mach at level-off cannot be greater than final Mach.	Check the Mach at the level-off altitude and FMACH to insure that the values comply with error message.
MARVIN		
1	CUT cannot be less than 0.	Change the variable so that it is positive.
ALOCIN		
1-4	Variables read must be greater than 0.	Change the convergence parameters card so that the variables are positive.
5	The target number receiving this weapon must be between -1 and NTGTS.	Change the variables to comply with error message.
DTLINE		
1-8	Root cannot be found.	Check the vulnerability levels for the aircraft and weapon yield for validity.

AFWL-TR-79-193

APPENDIX A
QUANTA LISTING

```

1  PROGRAM  QUANTA,INPUT,OUTPUT,TAPE4,TAPE5,TAPE6=INPUT,TAPE6=OUTPUT
2  COMMON/LCMT/MURK1(806)
3  COMMON/LCMT/MURK2(656)
4  COMMON/LCMT/ALUC(408)
5  COMMON/LCMT/FLAMP(408)
6  COMMON/IRP/ISSW1,ISSW2
7  COMMON/IRP/ISSW3,ISSW4
8  COMMON/IRP/ISSW5,ISSW6
9  COMMON/IRP/ISSW7,ISSW8
10 COMMON/IRP/ISSW9,ISSW10
11 COMMON/IRP/ISSW11,ISSW12
12 COMMON/IRP/ISSW13,ISSW14
13 COMMON/IRP/ISSW15,ISSW16
14 COMMON/IRP/ISSW17,ISSW18
15 COMMON/IRP/ISSW19,ISSW20
16 COMMON/IRP/ISSW21,ISSW22
17 COMMON/IRP/ISSW23,ISSW24
18 COMMON/IRP/ISSW25,ISSW26
19 COMMON/IRP/ISSW27,ISSW28
20 COMMON/IRP/ISSW29,ISSW30
21 COMMON/IRP/ISSW31,ISSW32
22 COMMON/IRP/ISSW33,ISSW34
23 COMMON/IRP/ISSW35,ISSW36
24 COMMON/IRP/ISSW37,ISSW38
25 COMMON/IRP/ISSW39,ISSW40
26 COMMON/IRP/ISSW41,ISSW42
27 COMMON/IRP/ISSW43,ISSW44
28 COMMON/IRP/ISSW45,ISSW46
29 COMMON/IRP/ISSW47,ISSW48
30 COMMON/IRP/ISSW49,ISSW50
31 COMMON/IRP/ISSW51,ISSW52
32 COMMON/IRP/ISSW53,ISSW54
33 COMMON/IRP/ISSW55,ISSW56
34 COMMON/IRP/ISSW57,ISSW58
35 COMMON/IRP/ISSW59,ISSW60
36 COMMON/IRP/ISSW61,ISSW62
37 COMMON/IRP/ISSW63,ISSW64
38 COMMON/IRP/ISSW65,ISSW66
39 COMMON/IRP/ISSW67,ISSW68
40 COMMON/IRP/ISSW69,ISSW70
41 COMMON/IRP/ISSW71,ISSW72
42 COMMON/IRP/ISSW73,ISSW74
43 COMMON/IRP/ISSW75,ISSW76
44 COMMON/IRP/ISSW77,ISSW78
45 COMMON/IRP/ISSW79,ISSW80
46 COMMON/IRP/ISSW81,ISSW82
47 COMMON/IRP/ISSW83,ISSW84
48 COMMON/IRP/ISSW85,ISSW86
49 COMMON/IRP/ISSW87,ISSW88
50 COMMON/IRP/ISSW89,ISSW90
51 COMMON/IRP/ISSW91,ISSW92
52 COMMON/IRP/ISSW93,ISSW94
53 COMMON/IRP/ISSW95,ISSW96
54 COMMON/IRP/ISSW97,ISSW98
55 COMMON/IRP/ISSW99,ISSW100
56 COMMON/IRP/ISSW101,ISSW102
57 COMMON/IRP/ISSW103,ISSW104
58 COMMON/IRP/ISSW105,ISSW106
59 COMMON/IRP/ISSW107,ISSW108
60 COMMON/IRP/ISSW109,ISSW110
61 COMMON/IRP/ISSW111,ISSW112
62 COMMON/IRP/ISSW113,ISSW114
63 COMMON/IRP/ISSW115,ISSW116
64 COMMON/IRP/ISSW117,ISSW118
65 COMMON/IRP/ISSW119,ISSW120
66 COMMON/IRP/ISSW121,ISSW122
67 COMMON/IRP/ISSW123,ISSW124
68 COMMON/IRP/ISSW125,ISSW126
69 COMMON/IRP/ISSW127,ISSW128
70 COMMON/IRP/ISSW129,ISSW130
71 COMMON/IRP/ISSW131,ISSW132
72 COMMON/IRP/ISSW133,ISSW134
73 COMMON/IRP/ISSW135,ISSW136
74 COMMON/IRP/ISSW137,ISSW138
75 COMMON/IRP/ISSW139,ISSW140
76 COMMON/IRP/ISSW141,ISSW142
77 COMMON/IRP/ISSW143,ISSW144
78 COMMON/IRP/ISSW145,ISSW146
79 COMMON/IRP/ISSW147,ISSW148
80 COMMON/IRP/ISSW149,ISSW150
81 COMMON/IRP/ISSW151,ISSW152
82 COMMON/IRP/ISSW153,ISSW154
83 COMMON/IRP/ISSW155,ISSW156
84 COMMON/IRP/ISSW157,ISSW158
85 COMMON/IRP/ISSW159,ISSW160
86 COMMON/IRP/ISSW161,ISSW162
87 COMMON/IRP/ISSW163,ISSW164
88 COMMON/IRP/ISSW165,ISSW166
89 COMMON/IRP/ISSW167,ISSW168
90 COMMON/IRP/ISSW169,ISSW170
91 COMMON/IRP/ISSW171,ISSW172
92 COMMON/IRP/ISSW173,ISSW174
93 COMMON/IRP/ISSW175,ISSW176
94 COMMON/IRP/ISSW177,ISSW178
95 COMMON/IRP/ISSW179,ISSW180
96 COMMON/IRP/ISSW181,ISSW182
97 COMMON/IRP/ISSW183,ISSW184
98 COMMON/IRP/ISSW185,ISSW186
99 COMMON/IRP/ISSW187,ISSW188
100 COMMON/IRP/ISSW189,ISSW190
101 COMMON/IRP/ISSW191,ISSW192
102 COMMON/IRP/ISSW193,ISSW194
103 COMMON/IRP/ISSW195,ISSW196
104 COMMON/IRP/ISSW197,ISSW198
105 COMMON/IRP/ISSW199,ISSW200
106 COMMON/IRP/ISSW201,ISSW202
107 COMMON/IRP/ISSW203,ISSW204
108 COMMON/IRP/ISSW205,ISSW206
109 COMMON/IRP/ISSW207,ISSW208
110 COMMON/IRP/ISSW209,ISSW210
111 COMMON/IRP/ISSW211,ISSW212
112 COMMON/IRP/ISSW213,ISSW214
113 COMMON/IRP/ISSW215,ISSW216
114 COMMON/IRP/ISSW217,ISSW218
115 COMMON/IRP/ISSW219,ISSW220
116 COMMON/IRP/ISSW221,ISSW222
117 COMMON/IRP/ISSW223,ISSW224
118 COMMON/IRP/ISSW225,ISSW226
119 COMMON/IRP/ISSW227,ISSW228
120 COMMON/IRP/ISSW229,ISSW230
121 COMMON/IRP/ISSW231,ISSW232
122 COMMON/IRP/ISSW233,ISSW234
123 COMMON/IRP/ISSW235,ISSW236
124 COMMON/IRP/ISSW237,ISSW238
125 COMMON/IRP/ISSW239,ISSW240
126 COMMON/IRP/ISSW241,ISSW242
127 COMMON/IRP/ISSW243,ISSW244
128 COMMON/IRP/ISSW245,ISSW246
129 COMMON/IRP/ISSW247,ISSW248
130 COMMON/IRP/ISSW249,ISSW250
131 COMMON/IRP/ISSW251,ISSW252
132 COMMON/IRP/ISSW253,ISSW254
133 COMMON/IRP/ISSW255,ISSW256
134 COMMON/IRP/ISSW257,ISSW258
135 COMMON/IRP/ISSW259,ISSW260
136 COMMON/IRP/ISSW261,ISSW262
137 COMMON/IRP/ISSW263,ISSW264
138 COMMON/IRP/ISSW265,ISSW266
139 COMMON/IRP/ISSW267,ISSW268
140 COMMON/IRP/ISSW269,ISSW270
141 COMMON/IRP/ISSW271,ISSW272
142 COMMON/IRP/ISSW273,ISSW274
143 COMMON/IRP/ISSW275,ISSW276
144 COMMON/IRP/ISSW277,ISSW278
145 COMMON/IRP/ISSW279,ISSW280
146 COMMON/IRP/ISSW281,ISSW282
147 COMMON/IRP/ISSW283,ISSW284
148 COMMON/IRP/ISSW285,ISSW286
149 COMMON/IRP/ISSW287,ISSW288
150 COMMON/IRP/ISSW289,ISSW290
151 COMMON/IRP/ISSW291,ISSW292
152 COMMON/IRP/ISSW293,ISSW294
153 COMMON/IRP/ISSW295,ISSW296
154 COMMON/IRP/ISSW297,ISSW298
155 COMMON/IRP/ISSW299,ISSW300
156 COMMON/IRP/ISSW301,ISSW302
157 COMMON/IRP/ISSW303,ISSW304
158 COMMON/IRP/ISSW305,ISSW306
159 COMMON/IRP/ISSW307,ISSW308
160 COMMON/IRP/ISSW309,ISSW310
161 COMMON/IRP/ISSW311,ISSW312
162 COMMON/IRP/ISSW313,ISSW314
163 COMMON/IRP/ISSW315,ISSW316
164 COMMON/IRP/ISSW317,ISSW318
165 COMMON/IRP/ISSW319,ISSW320
166 COMMON/IRP/ISSW321,ISSW322
167 COMMON/IRP/ISSW323,ISSW324
168 COMMON/IRP/ISSW325,ISSW326
169 COMMON/IRP/ISSW327,ISSW328
170 COMMON/IRP/ISSW329,ISSW330
171 COMMON/IRP/ISSW331,ISSW332
172 COMMON/IRP/ISSW333,ISSW334
173 COMMON/IRP/ISSW335,ISSW336
174 COMMON/IRP/ISSW337,ISSW338
175 COMMON/IRP/ISSW339,ISSW340
176 COMMON/IRP/ISSW341,ISSW342
177 COMMON/IRP/ISSW343,ISSW344
178 COMMON/IRP/ISSW345,ISSW346
179 COMMON/IRP/ISSW347,ISSW348
180 COMMON/IRP/ISSW349,ISSW350
181 COMMON/IRP/ISSW351,ISSW352
182 COMMON/IRP/ISSW353,ISSW354
183 COMMON/IRP/ISSW355,ISSW356
184 COMMON/IRP/ISSW357,ISSW358
185 COMMON/IRP/ISSW359,ISSW360
186 COMMON/IRP/ISSW361,ISSW362
187 COMMON/IRP/ISSW363,ISSW364
188 COMMON/IRP/ISSW365,ISSW366
189 COMMON/IRP/ISSW367,ISSW368
190 COMMON/IRP/ISSW369,ISSW370
191 COMMON/IRP/ISSW371,ISSW372
192 COMMON/IRP/ISSW373,ISSW374
193 COMMON/IRP/ISSW375,ISSW376
194 COMMON/IRP/ISSW377,ISSW378
195 COMMON/IRP/ISSW379,ISSW380
196 COMMON/IRP/ISSW381,ISSW382
197 COMMON/IRP/ISSW383,ISSW384
198 COMMON/IRP/ISSW385,ISSW386
199 COMMON/IRP/ISSW387,ISSW388
200 COMMON/IRP/ISSW389,ISSW390
201 COMMON/IRP/ISSW391,ISSW392
202 COMMON/IRP/ISSW393,ISSW394
203 COMMON/IRP/ISSW395,ISSW396
204 COMMON/IRP/ISSW397,ISSW398
205 COMMON/IRP/ISSW399,ISSW400
206 COMMON/IRP/ISSW401,ISSW402
207 COMMON/IRP/ISSW403,ISSW404
208 COMMON/IRP/ISSW405,ISSW406
209 COMMON/IRP/ISSW407,ISSW408
210 COMMON/IRP/ISSW409,ISSW410
211 COMMON/IRP/ISSW411,ISSW412
212 COMMON/IRP/ISSW413,ISSW414
213 COMMON/IRP/ISSW415,ISSW416
214 COMMON/IRP/ISSW417,ISSW418
215 COMMON/IRP/ISSW419,ISSW420
216 COMMON/IRP/ISSW421,ISSW422
217 COMMON/IRP/ISSW423,ISSW424
218 COMMON/IRP/ISSW425,ISSW426
219 COMMON/IRP/ISSW427,ISSW428
220 COMMON/IRP/ISSW429,ISSW430
221 COMMON/IRP/ISSW431,ISSW432
222 COMMON/IRP/ISSW433,ISSW434
223 COMMON/IRP/ISSW435,ISSW436
224 COMMON/IRP/ISSW437,ISSW438
225 COMMON/IRP/ISSW439,ISSW440
226 COMMON/IRP/ISSW441,ISSW442
227 COMMON/IRP/ISSW443,ISSW444
228 COMMON/IRP/ISSW445,ISSW446
229 COMMON/IRP/ISSW447,ISSW448
230 COMMON/IRP/ISSW449,ISSW450
231 COMMON/IRP/ISSW451,ISSW452
232 COMMON/IRP/ISSW453,ISSW454
233 COMMON/IRP/ISSW455,ISSW456
234 COMMON/IRP/ISSW457,ISSW458
235 COMMON/IRP/ISSW459,ISSW460
236 COMMON/IRP/ISSW461,ISSW462
237 COMMON/IRP/ISSW463,ISSW464
238 COMMON/IRP/ISSW465,ISSW466
239 COMMON/IRP/ISSW467,ISSW468
240 COMMON/IRP/ISSW469,ISSW470
241 COMMON/IRP/ISSW471,ISSW472
242 COMMON/IRP/ISSW473,ISSW474
243 COMMON/IRP/ISSW475,ISSW476
244 COMMON/IRP/ISSW477,ISSW478
245 COMMON/IRP/ISSW479,ISSW480
246 COMMON/IRP/ISSW481,ISSW482
247 COMMON/IRP/ISSW483,ISSW484
248 COMMON/IRP/ISSW485,ISSW486
249 COMMON/IRP/ISSW487,ISSW488
250 COMMON/IRP/ISSW489,ISSW490
251 COMMON/IRP/ISSW491,ISSW492
252 COMMON/IRP/ISSW493,ISSW494
253 COMMON/IRP/ISSW495,ISSW496
254 COMMON/IRP/ISSW497,ISSW498
255 COMMON/IRP/ISSW499,ISSW500
256 COMMON/IRP/ISSW501,ISSW502
257 COMMON/IRP/ISSW503,ISSW504
258 COMMON/IRP/ISSW505,ISSW506
259 COMMON/IRP/ISSW507,ISSW508
260 COMMON/IRP/ISSW509,ISSW510
261 COMMON/IRP/ISSW511,ISSW512
262 COMMON/IRP/ISSW513,ISSW514
263 COMMON/IRP/ISSW515,ISSW516
264 COMMON/IRP/ISSW517,ISSW518
265 COMMON/IRP/ISSW519,ISSW520
266 COMMON/IRP/ISSW521,ISSW522
267 COMMON/IRP/ISSW523,ISSW524
268 COMMON/IRP/ISSW525,ISSW526
269 COMMON/IRP/ISSW527,ISSW528
270 COMMON/IRP/ISSW529,ISSW530
271 COMMON/IRP/ISSW531,ISSW532
272 COMMON/IRP/ISSW533,ISSW534
273 COMMON/IRP/ISSW535,ISSW536
274 COMMON/IRP/ISSW537,ISSW538
275 COMMON/IRP/ISSW539,ISSW540
276 COMMON/IRP/ISSW541,ISSW542
277 COMMON/IRP/ISSW543,ISSW544
278 COMMON/IRP/ISSW545,ISSW546
279 COMMON/IRP/ISSW547,ISSW548
280 COMMON/IRP/ISSW549,ISSW550
281 COMMON/IRP/ISSW551,ISSW552
282 COMMON/IRP/ISSW553,ISSW554
283 COMMON/IRP/ISSW555,ISSW556
284 COMMON/IRP/ISSW557,ISSW558
285 COMMON/IRP/ISSW559,ISSW560
286 COMMON/IRP/ISSW561,ISSW562
287 COMMON/IRP/ISSW563,ISSW564
288 COMMON/IRP/ISSW565,ISSW566
289 COMMON/IRP/ISSW567,ISSW568
290 COMMON/IRP/ISSW569,ISSW570
291 COMMON/IRP/ISSW571,ISSW572
292 COMMON/IRP/ISSW573,ISSW574
293 COMMON/IRP/ISSW575,ISSW576
294 COMMON/IRP/ISSW577,ISSW578
295 COMMON/IRP/ISSW579,ISSW580
296 COMMON/IRP/ISSW581,ISSW582
297 COMMON/IRP/ISSW583,ISSW584
298 COMMON/IRP/ISSW585,ISSW586
299 COMMON/IRP/ISSW587,ISSW588
300 COMMON/IRP/ISSW589,ISSW590
301 COMMON/IRP/ISSW591,ISSW592
302 COMMON/IRP/ISSW593,ISSW594
303 COMMON/IRP/ISSW595,ISSW596
304 COMMON/IRP/ISSW597,ISSW598
305 COMMON/IRP/ISSW599,ISSW600
306 COMMON/IRP/ISSW601,ISSW602
307 COMMON/IRP/ISSW603,ISSW604
308 COMMON/IRP/ISSW605,ISSW606
309 COMMON/IRP/ISSW607,ISSW608
310 COMMON/IRP/ISSW609,ISSW610
311 COMMON/IRP/ISSW611,ISSW612
312 COMMON/IRP/ISSW613,ISSW614
313 COMMON/IRP/ISSW615,ISSW616
314 COMMON/IRP/ISSW617,ISSW618
315 COMMON/IRP/ISSW619,ISSW620
316 COMMON/IRP/ISSW621,ISSW622
317 COMMON/IRP/ISSW623,ISSW624
318 COMMON/IRP/ISSW625,ISSW626
319 COMMON/IRP/ISSW627,ISSW628
320 COMMON/IRP/ISSW629,ISSW630
321 COMMON/IRP/ISSW631,ISSW632
322 COMMON/IRP/ISSW633,ISSW634
323 COMMON/IRP/ISSW635,ISSW636
324 COMMON/IRP/ISSW637,ISSW638
325 COMMON/IRP/ISSW639,ISSW640
326 COMMON/IRP/ISSW641,ISSW642
327 COMMON/IRP/ISSW643,ISSW644
328 COMMON/IRP/ISSW645,ISSW646
329 COMMON/IRP/ISSW647,ISSW648
330 COMMON/IRP/ISSW649,ISSW650
331 COMMON/IRP/ISSW651,ISSW652
332 COMMON/IRP/ISSW653,ISSW654
333 COMMON/IRP/ISSW655,ISSW656
334 COMMON/IRP/ISSW657,ISSW658
335 COMMON/IRP/ISSW659,ISSW660
336 COMMON/IRP/ISSW661,ISSW662
337 COMMON/IRP/ISSW663,ISSW664
338 COMMON/IRP/ISSW665,ISSW666
339 COMMON/IRP/ISSW667,ISSW668
340 COMMON/IRP/ISSW669,ISSW670
341 COMMON/IRP/ISSW671,ISSW672
342 COMMON/IRP/ISSW673,ISSW674
343 COMMON/IRP/ISSW675,ISSW676
344 COMMON/IRP/ISSW677,ISSW678
345 COMMON/IRP/ISSW679,ISSW680
346 COMMON/IRP/ISSW681,ISSW682
347 COMMON/IRP/ISSW683,ISSW684
348 COMMON/IRP/ISSW685,ISSW686
349 COMMON/IRP/ISSW687,ISSW688
350 COMMON/IRP/ISSW689,ISSW690
351 COMMON/IRP/ISSW691,ISSW692
352 COMMON/IRP/ISSW693,ISSW694
353 COMMON/IRP/ISSW695,ISSW696
354 COMMON/IRP/ISSW697,ISSW698
355 COMMON/IRP/ISSW699,ISSW700
356 COMMON/IRP/ISSW701,ISSW702
357 COMMON/IRP/ISSW703,ISSW704
358 COMMON/IRP/ISSW705,ISSW706
359 COMMON/IRP/ISSW707,ISSW708
360 COMMON/IRP/ISSW709,ISSW710
361 COMMON/IRP/ISSW711,ISSW712
362 COMMON/IRP/ISSW713,ISSW714
363 COMMON/IRP/ISSW715,ISSW716
364 COMMON/IRP/ISSW717,ISSW718
365 COMMON/IRP/ISSW719,ISSW720
366 COMMON/IRP/ISSW721,ISSW722
367 COMMON/IRP/ISSW723,ISSW724
368 COMMON/IRP/ISSW725,ISSW726
369 COMMON/IRP/ISSW727,ISSW728
370 COMMON/IRP/ISSW729,ISSW730
371 COMMON/IRP/ISSW731,ISSW732
372 COMMON/IRP/ISSW733,ISSW734
373 COMMON/IRP/ISSW735,ISSW736
374 COMMON/IRP/ISSW737,ISSW738
375 COMMON/IRP/ISSW739,ISSW740
376 COMMON/IRP/ISSW741,ISSW742
377 COMMON/IRP/ISSW743,ISSW744
378 COMMON/IRP/ISSW745,ISSW746
379 COMMON/IRP/ISSW747,ISSW748
380 COMMON/IRP/ISSW749,ISSW750
381 COMMON/IRP/ISSW751,ISSW752
382 COMMON/IRP/ISSW753,ISSW754
383 COMMON/IRP/ISSW755,ISSW756
384 COMMON/IRP/ISSW757,ISSW758
385 COMMON/IRP/ISSW759,ISSW760
386 COMMON/IRP/ISSW761,ISSW762
387 COMMON/IRP/ISSW763,ISSW764
388 COMMON/IRP/ISSW765,ISSW766
389 COMMON/IRP/ISSW767,ISSW768
390 COMMON/IRP/ISSW769,ISSW770
391 COMMON/IRP/ISSW771,ISSW772
392 COMMON/IRP/ISSW773,ISSW774
393 COMMON/IRP/ISSW775,ISSW776
394 COMMON/IRP/ISSW777,ISSW778
395 COMMON/IRP/ISSW779,ISSW780
396 COMMON/IRP/ISSW781,ISSW782
397 COMMON/IRP/ISSW783,ISSW784
398 COMMON/IRP/ISSW785,ISSW786
399 COMMON/IRP/ISSW787,ISSW788
400 COMMON/IRP/ISSW789,ISSW790
401 COMMON/IRP/ISSW791,ISSW792
402 COMMON/IRP/ISSW793,ISSW794
403 COMMON/IRP/ISSW795,ISSW796
404 COMMON/IRP/ISSW797,ISSW798
405 COMMON/IRP/ISSW799,ISSW800
406 COMMON/IRP/ISSW801,ISSW802
407 COMMON/IRP/ISSW803,ISSW804
408 COMMON/IRP/ISSW805,ISSW806
409 COMMON/IRP/ISSW807,ISSW808
410 COMMON/IRP/ISSW809,ISSW810
411 COMMON/IRP/ISSW811,ISSW812
412 COMMON/IRP/ISSW813,ISSW814
413 COMMON/IRP/ISSW815,ISSW816
414 COMMON/IRP/ISSW817,ISSW818
415 COMMON/IRP/ISSW819,ISSW820
416 COMMON/IRP/ISSW821,ISSW822
417 COMMON/IRP/ISSW823,ISSW824
418 COMMON/IRP/ISSW825,ISSW826
419 COMMON/IRP/ISSW827,ISSW828
420 COMMON/IRP/ISSW829,ISSW830
421 COMMON/IRP/ISSW831,ISSW832
422 COMMON/IRP/ISSW833,ISSW834
423 COMMON/IRP/ISSW835,ISSW836
424 COMMON/IRP/ISSW837,ISSW838
425 COMMON/IRP/ISSW839,ISSW840
426 COMMON/IRP/ISSW841,ISSW842
427 COMMON/IRP/ISSW843,ISSW844
428 COMMON/IRP/ISSW845,ISSW846
429 COMMON/IRP/ISSW847,ISSW848
430 COMMON/IRP/ISSW849,ISSW850
431 COMMON/IRP/ISSW851,ISSW852
432 COMMON/IRP/ISSW853,ISSW854
433 COMMON/IRP/ISSW855,ISSW856
434 COMMON/IRP/ISSW857,ISSW858
435 COMMON/IRP/ISSW859,ISSW860
436 COMMON/IRP/ISSW861,ISSW862
437 COMMON/IRP/ISSW863,ISSW864
438 COMMON/IRP/ISSW865,ISSW866
439 COMMON/IRP/ISSW867,ISSW868
440 COMMON/IRP/ISSW869,ISSW870
441 COMMON/IRP/ISSW871,ISSW872
442 COMMON/IRP/ISSW873,ISSW874
443 COMMON/IRP/ISSW875,ISSW876
444 COMMON/IRP/ISSW877,ISSW878
445 COMMON/IRP/ISSW879,ISSW880
446 COMMON/IRP/ISSW881,ISSW882
447 COMMON/IRP/ISSW883,ISSW884
448 COMMON/IRP/ISSW885,ISSW886
449 COMMON/IRP/ISSW887,ISSW888
450 COMMON/IRP/ISSW889,ISSW890
451 COMMON/IRP/ISSW891,ISSW892
452 COMMON/IRP/ISSW893,ISSW894
453 COMMON/IRP/ISSW895,ISSW896
454 COMMON/IRP/ISSW897,ISSW898
455 COMMON/IRP/ISSW899,ISSW900
456 COMMON/IRP/ISSW901,ISSW902
457 COMMON/IRP/ISSW903,ISSW904
458 COMMON/IRP/ISSW905,ISSW906
459 COMMON/IRP/ISSW907,ISSW908
460 COMMON/IRP/ISSW909,ISSW910
461 COMMON/IRP/ISSW911,ISSW912
462 COMMON/IRP/ISSW913,ISSW914
463 COMMON/IRP/ISSW915,ISSW916
464 COMMON/IRP/ISSW917,ISSW918
465 COMMON/IRP/ISSW919,ISSW920
466 COMMON/IRP/ISSW921,ISSW922
467 COMMON/IRP/ISSW923,ISSW924
468 COMMON/IRP/ISSW925,ISSW926
469 COMMON/IRP/ISSW927,ISSW928
470 COMMON/IRP/ISSW929,ISSW930
471 COMMON/IRP/ISSW931,ISSW932
472 COMMON/IRP/ISSW933,ISSW934
473 COMMON/IRP/ISSW935,ISSW936
474 COMMON/IRP/ISSW937,ISSW938
475 COMMON/IRP/ISSW939,ISSW940
476 COMMON/IRP/ISSW941,ISSW942
477 COMMON/IRP/ISSW943,ISSW944
478 COMMON/IRP/ISSW945,ISSW946
479 COMMON/IRP/ISSW947,ISSW948
480 COMMON/IRP/ISSW949,ISSW950
481 COMMON/IRP/ISSW951,ISSW952
482 COMMON/IRP/ISSW953,ISSW954
483 COMMON/IRP/ISSW955,ISSW956
484 COMMON/IRP/ISSW957,ISSW958
485 COMMON/IRP/ISSW959,ISSW960
486 COMMON/IRP/ISSW961,ISSW962
487 COMMON/IRP/ISSW963,ISSW964
488 COMMON/IRP/ISSW965,ISSW966
489 COMMON/IRP/ISSW967,ISSW968
490 COMMON/IRP/ISSW969,ISSW970
491 COMMON/IRP/ISSW971,ISSW972
492 COMMON/IRP/ISSW973,ISSW974
493 COMMON/IRP/ISSW975,ISSW976
494 COMMON/IRP/ISSW977,ISSW978
495 COMMON/IRP/ISSW979,ISSW980
496 COMMON/IRP/ISSW981,ISSW982
497 COMMON/IRP/ISSW983,ISSW984
498 COMMON/IRP/ISSW985,ISSW986
499 COMMON/IRP/ISSW987,ISSW988
500 COMMON/IRP/ISSW989,ISSW990
501 COMMON/IRP/ISSW991,ISSW992
502 COMMON/IRP/ISSW993,ISSW994
503 COMMON/IRP/ISSW995,ISSW996
504 COMMON/IRP/ISSW997,ISSW998
505 COMMON/IRP/ISSW999,ISSW1000
506 COMMON/IRP/ISSW1001,ISSW1002
507 COMMON/IRP/ISSW1003,ISSW1004
508 COMMON/IRP/ISSW1005,ISSW1006
509 COMMON/IRP/ISSW1007,ISSW1008
510 COMMON/IRP/ISSW1009,ISSW1010
511 COMMON/IRP/ISSW1011,ISSW1012
512 COMMON/IRP/ISSW1013,ISSW1014
513 COMMON/IRP/ISSW1015,ISSW1016
514 COMMON/IRP/ISSW1017,ISSW1018
515 COMMON/IRP/ISSW1019,ISSW1020
516 COMMON/IRP/ISSW1021,ISSW1022
517 COMMON/IRP/ISSW1023,ISSW1024
518 COMMON/IRP/ISSW1025,ISSW1026
519 COMMON/IRP/ISSW1027,ISSW1028
520 COMMON/IRP/ISSW1029,ISSW1030
521 COMMON/IRP/ISSW1031,ISSW1032
522 COMMON/IRP/ISSW1033,ISSW1034
523 COMMON/IRP/ISSW1035,ISSW1036
524 COMMON/IRP/ISSW1037,ISSW1038
525 COMMON/IRP/ISSW1039,ISSW1040
526 COMMON/IRP/ISSW1041,ISSW1042
527 COMMON/IRP/ISSW1043,ISSW1044
528 COMMON/IRP/ISSW1045,ISSW1046
529 COMMON/IRP/ISSW1047,ISSW1048
530 COMMON/IRP/ISSW1049,ISSW1050
531 COMMON/IRP/ISSW1051,ISSW1052
532 COMMON/IRP/ISSW1053,ISSW1054
533 COMMON/IRP/ISSW1055,ISSW1056
534 COMMON/IRP/ISSW1057,ISSW1058
535 COMMON/IRP/ISSW1059,ISSW1060
536 COMMON/IRP/ISSW1061,ISSW1062
537 COMMON/IRP/ISSW1063,ISSW1064
538 COMMON/IRP/ISSW1065,ISSW1066
539 COMMON/IRP/ISSW1067,ISSW1068
540 COMMON/IRP/ISSW1069,ISSW1070
541 COMMON/IRP/ISSW1071,ISSW1072
542 COMMON/IRP/ISSW1073,ISSW1074
543 COMMON/IRP/ISSW1075,ISSW1076
544 COMMON/IRP/ISSW1077,ISSW1078
545 COMMON/IRP/ISSW1079,ISSW1080
546 COMMON/IRP/ISSW1081,ISSW1082
547 COMMON/IRP/ISSW1083,ISSW1084
548 COMMON/IRP/ISSW1085,ISSW1086
549 COMMON/IRP/ISSW1087,ISSW1088
550 COMMON/IRP/ISSW1089,ISSW1090
551 COMMON/IRP/ISSW1091,ISSW1092
552 COMMON/IRP/ISSW1093,ISSW1094
553 COMMON/IRP/ISSW1095,ISSW1096
554 COMMON/IRP/ISSW1097,ISSW1098
555 COMMON/IRP/ISSW1099,ISSW1100
556 COMMON/IRP/ISSW1101,ISSW1102
557 COMMON/IRP/ISSW1103,ISSW1104
558 COMMON/IRP/ISSW1105,ISSW1106
559 COMMON/IRP/ISSW1107,ISSW1108
560 COMMON/IRP/ISSW1109,ISSW1110
561 COMMON/IRP/ISSW1111,ISSW1112
562 COMMON/IRP/ISSW1113,ISSW1114
563 COMMON/IRP/ISSW1115,ISSW1116
564 COMMON/IRP/ISSW1117,ISSW1118
565 COMMON/IRP/ISSW1119,ISSW1120
566 COMMON/IRP/ISSW1121,ISSW1122
567 COMMON/IRP/ISSW1123,ISSW1124
568 COMMON/IRP/ISSW1125,ISSW1126
569 COMMON/IRP/ISSW1127,ISSW1128
570 COMMON/IRP/ISSW1129,ISSW1130
571 COMMON/IRP/ISSW1131,ISSW1132
572 COMMON/IRP/ISSW1133,ISSW1134
573 COMMON/IRP/ISSW1135,ISSW1136
574 COMMON/IRP/ISSW1137,ISSW1138
575 COMMON/IRP/ISSW1139,ISSW1140
576 COMMON/IRP/ISSW1141,ISSW1142
577 COMMON/IRP/ISSW1143,ISSW1144
578 COMMON/IRP/ISSW1145,ISSW1146
579 COMMON/IRP/ISSW1147,ISSW1148
580 COMMON/IRP/ISSW1149,ISSW1150
581 COMMON/IRP/ISSW1151,ISSW1152
582 COMMON/IRP/ISSW1153,ISSW1154
583 COMMON/IRP/ISSW1155,ISSW1156
584 COMMON/IRP/ISSW1157,ISSW1158
585 COMMON/IRP/ISSW1159,ISSW1160
586 COMMON/IRP/ISSW1161,ISSW1162
587 COMMON/IRP/ISSW1163,ISSW1164
588 COMMON/IRP/ISSW1165,ISSW1166
589 COMMON/IRP/ISSW1167,ISSW1168
590 COMMON/IRP/ISSW1169,ISSW1170
591 COMMON/IRP/ISSW1171,ISSW1172
592 COMMON/IRP/ISSW1173,ISSW1174
593 COMMON/IRP/ISSW1175,ISSW1176
594 COMMON/IRP/ISSW1177,ISSW1178
595 COMMON/IRP/ISSW1179,ISSW1180
596 COMMON/IRP/ISSW1181,ISSW1182
597 COMMON/IRP/ISSW1183,ISSW1184
598 COMMON/IRP/ISSW1185,ISSW1186
599 COMMON/IRP/ISSW1187,ISSW1188
600 COMMON/IRP/ISSW1189,ISSW1190
601 COMMON/IRP/ISSW1191,ISSW1192
602 COMMON/IRP/ISSW1193,ISSW1194
603 COMMON/IRP/ISSW1195,ISSW1196
604 COMMON/IRP/ISSW1197,ISSW1198
605 COMMON/IRP/ISSW1199,ISSW1200
606 COMMON/IRP/ISSW1201,ISSW1202
607 COMMON/IRP/ISSW1203,ISSW1204
608 COMMON/IRP/ISSW1205,ISSW1206
609 COMMON/IRP/ISSW1207,ISSW1208
610 COMMON/IRP/ISSW1209,ISSW1210
611 COMMON/IRP/ISSW1211,ISSW1212
612 COMMON/IRP/ISSW1213,ISSW1214
613 COMMON/IRP/ISSW1215,ISSW1216
614 COMMON/IRP/ISSW1217,ISSW1218
615 COMMON/IRP/ISSW1219,ISSW1220
616
```

```

51 * #WKRZ(INFISA),#WKRZ(NFLMT),#WKRZ(NBUNUS))
52 CALL MARKV, (MARKV,#WKRZ(NCUT),#MARKV)
53 CALL ALLOC (INTGTS,NSUBS,NMPS,MANAP,ALOC,#WKRZ(NALUCU),
54 * #WKRZ(INISUBS),#WKRZ(NMPS),#WKRZ(NLUTG))
55 CALL LETHAL(#WKRZ(NMPS),#WKRZ(NUTEN),#NTYPES,#MTYPES,#MAP,#MAP,
56 * #WKRZ(NS),#WKRZ(NU),#WKRZ(NM),#WKRZ(NF),#WKRZ(INDATA),#WKRZ
57 * (NFLMP),#WKRZ(INFISA),#WKRZ(NFLMT),#WKRZ(NBUNUS),#WKRZ(NPS)),
58 * #WKRZ(INICAL),#WKRZ(NYTEL),#WKRZ(INAREAT),#WKRZ(NHMLA) )
59 ALL THE "KILL AREA" CALCULATIONS ARE COMPLETE
60 BEGIN COMPUTING SURVIVABILITIES
61 CALL SKYMAT(#WKRZ(NSUBV),#WKRZ(NSRV),#WKRZ(NBRLIM),#WKRZ(NBRL),
62 * #WKRZ(NMWAY),#WKRZ(NISEV),#WKRZ(NDELTA),#WKRZ(NUTG),#WKRZ
63 * (INTGTL),#WKRZ(NSUBLA),#WKRZ(NSUBLN),#WKRZ(NMTYPE),#WKRZ(NREL),
64 * #WKRZ(NHVS),#WKRZ(NKNGML),#WKRZ(NFNGMA),#WKRZ(NMPS),#WKRZ
65 * (NFLMTO),#WKRZ(NFLMT),#WKRZ(NMPS),#WKRZ(NDELTM),#WKRZ(NAREAT),
66 * #WKRZ(NHMLA),#WKRZ(NUTEN),#WKRZ(NS),#WKRZ(ND),#WKRZ(NHAW),
67 * #WKRZ(NUTS),#WKRZ(MANAP),#NTYPES,#MTYPES,NACT,MXPMAY,NSUBS,
68 * #MAP,#MAP,#WKRZ(NVAL))
69 COMPUTE THE OBJECTIVE FUNCTION
70 CALL TGTAL(INTGTS,NMPS,NTYPES,ALOC,#WKRZ(NSUBV),#WKRZ(NVAL),#WKRZ
71 * (NRELVA),#FLAMB,#WKRZ(NPRDU),#WKRZ(NKILL),#MANAP,NACT,#WKRZ
72 * (NISEV),#WKRZ(NSRV),#WKRZ(NBUNUS),#WKRZ(NBASVA),#WKRZ(NMTYPE),
73 * #WKRZ(NREL),#WKRZ(NMPS),NSUBS,NTYPES,#WKRZ(NPS))
74 CALL KILCU (MARKV,#WKRZ(NCU),#MAPV,NTYPES,#WKRZ(NSUBV),#WKRZ
75 * (NSUMVR),#WKRZ(NSUMAR),#INTGTS,ALOC,#WKRZ(NVAL),#WKRZ(NISEV),#WKRZ
76 * (NPRDU),#NACT,#WKRZ(NRELVA),#WKRZ(NKILL),#WKRZ(NBUNUS),
77 * #MAPV,#WKRZ(NBASVA),#WKRZ(NBUNUS))
78 CHECK FOR RESTART TAPE
79 CALL RSTAK (MARKV,#WKRZ(NALOC),#FLAMB,#WKRZ(NPRDU),#WKRZ(NKILL),
80 * #WKRZ(NALUCU),#INTGTS,NMPS,NACT)
81 FIND OPTIMAL ALLOCATION
82 CALL REALUC(INTGTS,NMPS,NSUBS,NTYPES,MANAP,MARKV,INTGTS,#WKRZ(NSUBV),
83 * #WKRZ(NVAL),#WKRZ(NRELVA),#WKRZ(NISEV),#WKRZ(NMPS),ALOC,#WKRZ
84 * (NALUCU),#WKRZ(NCU),#FLAMB,#WKRZ(NKILL),#WKRZ(NPRDU),#WKRZ(NBUNUS),
85 * #WKRZ(NISEV),#MANAP,NACT,#WKRZ(NBASVA),#WKRZ(NBUNUS),#WKRZ
86 * (NMTYPE),#WKRZ(NREL),#MTYPES,#WKRZ(NSRV),#WKRZ(NASHI),#WKRZ
87 * (NASLU),#WKRZ(NALHI),#WKRZ(NALU),#WKRZ(NPS))
88 OPTIMUM ATTAINED. INTERPRET AND PRINT RESULTS
89 CALL ALINT(INTGTS,NMPS,ALOC,#WKRZ(NALUCU))
90 CALL ALOUT(INTGTS,NSUBS,NMPS,ALOC,#WKRZ(NISEV),#WKRZ(NMPS))
91 CALL TGTAL(INTGTS,NMPS,NTYPES,ALOC,#WKRZ(NSUBV),#WKRZ(NVAL),#WKRZ
92 * (NRELVA),#FLAMB,#WKRZ(NPRDU),#WKRZ(NKILL),#MANAP,NACT,#WKRZ
93 * (NISEV),#WKRZ(NSRV),#WKRZ(NBUNUS),#WKRZ(NBASVA),#WKRZ(NMTYPE),
94 * #WKRZ(NREL),#WKRZ(NMPS),NSUBS,NTYPES,#WKRZ(NPS))
95 CALL KILCU (MARKV,#WKRZ(NCU),#MAPV,NTYPES,#WKRZ(NSUBV),#WKRZ
96 * (NSUMVR),#WKRZ(NSUMAR),#INTGTS,ALOC,#WKRZ(NVAL),#WKRZ(NISEV),#WKRZ
97 * (NPRDU),#NACT,#WKRZ(NRELVA),#WKRZ(NKILL),#WKRZ(NBUNUS),
98 * #MAPV,#WKRZ(NBASVA),#WKRZ(NBUNUS))
99 CHECK FOR ANOTHER MAXVAL
100 IF ((QUOT,EW,LS,UM,ISSW,EW,LS) CALL TERM (QUOT,ISSW))

```

52 QUANTA
 53 QUANTA
 54 QUANTA
 55 QUANTA
 56 QUANTA
 57 QUANTA
 58 QUANTA
 59 QUANTA
 60 QUANTA
 61 QUANTA
 62 QUANTA
 63 QUANTA
 64 QUANTA
 65 QUANTA
 66 QUANTA
 67 QUANTA
 68 QUANTA
 69 QUANTA
 70 QUANTA
 71 QUANTA
 72 QUANTA
 73 QUANTA
 74 QUANTA
 75 QUANTA
 76 QUANTA
 77 QUANTA
 78 QUANTA
 79 QUANTA
 80 QUANTA
 81 QUANTA
 82 QUANTA
 83 QUANTA
 84 QUANTA
 85 QUANTA
 86 QUANTA
 87 QUANTA
 88 QUANTA
 89 QUANTA
 90 QUANTA
 91 QUANTA
 92 QUANTA
 93 QUANTA
 94 QUANTA
 95 QUANTA
 96 QUANTA
 97 QUANTA
 98 QUANTA
 99 QUANTA
 100 QUANTA
 101 QUANTA

PAGE 3

```
101 C
102 IF (IMARKV.LL.IMARKV .AND. .IMARKV.GT.0) GO TO 80
103 STOP
104 802 FORMAT(IX," THE REQUIRED SIZE FOR THE ALUC AND FLAMB ARRAYS IS ",
105 " 17,75 THE SIZE YOU HAVE DESIGNATED IS ",I7)
106 801 FORMAT (IX," THE REQUIRED SIZE FOR THE INDX ARRAY IS ",I7)
107 " " THE SIZE YOU HAVE DESIGNATED IS ",I7)
108 END
109
```

QUANTA
QUANTA
QUANTA
QUANTA
QUANTA
QUANTA
QUANTA
QUANTA

102
103
104
105
106
107
108
109

```

109 SUBROUTINE WUKKRI(NUTS,NTYPES,NSUBS,NVAL,NKEL,VARIN,SUBS,INMPS,
110 *IMTYP,NWUKRI,NLCM),ILSECUACI(NKEL,MTYPES,IBASVA)
111
112 SUBROUTINE WUKKI CALCULATES SUBARRAY STAFFING
113 ADDRESSES IN THE WUKKI ARRAY
114
115 NVAL=I
116 NKELJA=NVAL*NTUTS*NTYPES
117 NLSUBS=NKELJA*NTYPES
118 INMPS=I*LSUBS*NSUBS
119 IMTYP=INMPS*NSUBS
120 NKEL=NMTP*NSUBS
121 NISEU=NKEL*MTYPES
122 NBSVA=INISEU*NACT
123 NWUKRI=NBASVA*NUTS
124 IF ((NWUKRI-1).GT.NLCM) WRITE(6,6000)NWUKRI,NLCM
125 RETURN
126
127 6000 FORMAT(1,'30M SUBROUTINE WUKKRI HAS DETECTED IMATP,*,
128 * 3M THE REQUIRED WUKKI ARRAY SIZE IS AT LEAST ,I10,*,
129 * 3M THE PROVIDED WUKKI ARRAY SIZE IS ,I10,*,
130 * 60M HENCE AN ERROR WILL BE DETECTED LATER IF WE GUNT BOMB FIRST)
131
132 END

```

PAGE 5

```
129 SUBROUTINE REMAI(NBARR,MMARRV,NSURV,NALUD,NOUT,NLCM,MMPI,MMANAPB) 130
130 * NSRV) 131
131 SUBROUTINE REMAI CALCULATES THE REMAINING SUBARRAY 132
132 STARTING ADDRESSES IN THE SUBARRAY 133
133 DATA MMPI/MMPI/ 134
134 NSURV=NBARR 135
135 NSURV=NBARR 136
136 NALUD=NSRV*MMPI*MMANAPB 137
137 NOUT=NALUD*MMPI 138
138 NBARR=NOUT*MMPI-1 139
139 WRITE (6,6000)NBARR,NLCM 140
140 IF (NBARR.NE.NLCM) CALL ABORT(MMPI*1,"SIZE OF SUBARRAY EXCEED 141
141 IS DIMENSION") 142
142 RETURN 143
143 6000 FORMAT(14H THE REQUIRED SUBARRAY SIZE IS *110*/, 144
144 * 34H THE PROVIDED SUBARRAY SIZE IS *110) 145
145 END 146
```

[illegible]

[illegible]

220	SUBROUTINE REMK2(NWORK,MANAP,MANAP,NDI,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15,N16,N17,N18,N19,N20,N21,N22,N23,N24,N25,N26,N27,N28,N29,N30,N31,N32,N33,N34,N35
221	QUANTA
222	QUANTA
223	QUANTA
224	QUANTA
225	QUANTA
226	QUANTA
227	QUANTA
228	QUANTA
229	QUANTA
230	QUANTA
231	QUANTA
232	QUANTA
233	QUANTA
234	QUANTA
235	QUANTA

```

SUBROUTINE REMK2(NWORK,MANAP,MANAP,NDI,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15,N16,N17,N18,N19,N20,N21,N22,N23,N24,N25,N26,N27,N28,N29,N30,N31,N32,N33,N34,N35
  SUBROUTINE REMK2 CALCULATES THE STARTING ADDRESSES OF
    THE REMAINING SUBARRAYS IN THE WORK2 ARRAY
    DATA WHERE/5000000/
    I=1,NWORK
    N1=1,NWORK
    N2=1,NWORK
    N3=1,NWORK
    N4=1,NWORK
    N5=1,NWORK
    N6=1,NWORK
    N7=1,NWORK
    N8=1,NWORK
    N9=1,NWORK
    N10=1,NWORK
    N11=1,NWORK
    N12=1,NWORK
    N13=1,NWORK
    N14=1,NWORK
    N15=1,NWORK
    N16=1,NWORK
    N17=1,NWORK
    N18=1,NWORK
    N19=1,NWORK
    N20=1,NWORK
    N21=1,NWORK
    N22=1,NWORK
    N23=1,NWORK
    N24=1,NWORK
    N25=1,NWORK
    N26=1,NWORK
    N27=1,NWORK
    N28=1,NWORK
    N29=1,NWORK
    N30=1,NWORK
    N31=1,NWORK
    N32=1,NWORK
    N33=1,NWORK
    N34=1,NWORK
    N35=1,NWORK
    IF (NWORK*5000000) CALL ADJUST(NDI,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15,N16,N17,N18,N19,N20,N21,N22,N23,N24,N25,N26,N27,N28,N29,N30,N31,N32,N33,N34,N35
    IS DIMENSION")
  RETURN
  FORMAT(10,34) THE REQUIRED WORK2 ARRAY SIZE IS 110000
  * 34 THE PROVIDED WORK2 ARRAY SIZE IS 1100
  END

```

235	1	SUBROUTINE ABOUT (WHERE,NUM,MSG)	QUANTA	235
236	0	DIMENSION MSG(12)	QUANTA	237
237	0		QUANTA	238
238	0	WRITE (6,3) NUM,WHERE	QUANTA	239
239	0	DO 1 L=1,12	QUANTA	240
240	0	IF (MSG(L).E9.0) GO TO 2	QUANTA	241
241	0	1 CONTINUE	QUANTA	242
242	0	L=13	QUANTA	243
243	0		QUANTA	244
244	0	2 L=L-1	QUANTA	245
245	0	IF (L.GT.0) WRITE (6,4) (MSG(I),I=1,L)	QUANTA	246
246	0		QUANTA	247
247	0	STOP "PROGRAM ABOUT "	QUANTA	248
248	0		QUANTA	249
249	0	3 FORMAT (1X,12I10,10X,12A10)	QUANTA	250
250	0	4 FORMAT (1X,5H-----,5A,12A10)	QUANTA	251
251	0	END	QUANTA	252
252	0		QUANTA	253

```

253 SUBROUTINE STARTK(NTUTS,NSUBS,NTYPES,MAXWAY,NTYPE,LOUT,INACT,INMAY 254
254 1) 255
255 DATA WHERE/BNSTART/ 256
256 READ(5,5000) NTUTS,NSUBS,NTYPES,MAXWAY,NTYPE,LOUT,INACT,INMAY 257
257 CALL PAGE(0) 258
258 *WRITE(6,0010) *MAP,NTUTS,NSUBS,NTYPES,INACT,NTYPES 259
259 IF(NTUTS.LE.0) CALL ABORT(WHERE,1,"NUMBER OF BASES MUST BE GREATER 260
260 1 THAN 0") 261
261 IF(NSUBS.LE.0) CALL ABORT(WHERE,2,"NUMBER OF SUBS MUST BE GREATER 262
262 1 THAN 0") 263
263 IF(NTYPES.LE.0) CALL ABORT(WHERE,3,"NUMBER OF AIRCRAFT TYPES MUST 264
264 1BE GREATER THAN 0") 265
265 IF(NTYPES.LE.0) CALL ABORT(WHERE,4,"NUMBER OF MISSILE TYPES MUST B 266
266 1 GREATER THAN 0") 267
267 IF(MAXWAY.LE.0) CALL ABORT(WHERE,5,"MAXIMUM NUMBER OF RUNWAYS ON A 268
268 1 BASE MUST BE GREATER THAN 0") 269
269 IF((LOUT.NE.1).AND.(LOUT.NE.2)) CALL ABORT(WHERE,6,"LOUT MUST BE C 270
270 1 EITHER 1 OR 2") 271
271 IF(INACT.LE.0) CALL ABORT(WHERE,7,"THE TOTAL NUMBER OF AIRCRAFT MUS 272
272 1 BE GREATER THAN 0") 273
273 IF(INMAY.LE.0) CALL ABORT(WHERE,8,"THE NUMBER OF MARVAL RUNS CANN 274
274 1 BE NEGATIVE") 275
275 CALL PAGE(2) 276
276 RETURN 277
277 5000 FORMAT(1X,0) 278
278 0010 FORMAT(1X,0) 279
279 15,0 TARGETS, 0.15,0 SUB LOCATIONS, 0.10,0.15,0 TYPES OF AIRCRAFT 280
280 1 (TOTAL OF 0.15,0 ), AND 0.15,0 TYPES OF MISSILES,0) 281
281 282
282 283

```



```

333      30 CONTINUE
334      FMAC=FLUAT(NAC-INAC1)*1.0
335      NLINES=(FMAC-1.)/30.
336      CALL PAGE(-MAX(12,INT(NLINES-2))
337      WRITE(6,6010)IBASVAL(1),TOTLAT(1),TOTLNG(1),NRWAYS(1),J,VAL(1,J)
338      *J=1,INT(NLINES)
339      IF((TOTLAT(1).LT.0.0).OR.(TOTLNG(1).LT.0.0)) CALL ABORT(WHERE*2,"
340      BASE LATITUDE AND LONGITUDE MUST BE GREATER THAN 0")
341      IF(NRWAYS(1).LE.0 .OR. NRWAYS(1).GT.MAXWAY) CALL ABORT(WHERE*3,"NO
342      *MBER OF RUNWAYS ON A BASE MUST BE GREATER THAN 0 AND NOT GREATER T
343      *HAN THE MAXIMUM NUMBER OF RUNWAYS")
344      DO 35 J=1,INT(NLINES)
345      IF(VAL(1,J).LT.0.0) CALL ABORT(WHERE*4,"NUMBER OF AIRCRAFT OF A GI
346      *VEN TYPE ON A GIVEN BASE CANNOT BE LESS THAN 0")
347      35 CONTINUE
348      WRITE(6,6020)(ISEV(IAC),IAC=INAC1,INAC)
349      DO 37 IAC=INAC1,INAC
350      IF((ISEV(IAC).LE.0).OR.((ISEV(IAC).GT.0,INT(NLINES))) CALL ABORT(WHERE*5,"
351      *ANY AIRCRAFT TYPE ON A BASE MUST BE GREATER THAN 0 AND NOT GREATER
352      *THAN THE NUMBER OF TYPES")
353      37 CONTINUE
354      40 CONTINUE
355      *WRITE AIRCRAFT TYPE PARAMETERS
356      CALL PAGE(-INT(NLINES-3))
357      WRITE(6,6030)(J,RELVAL(J),ORTIME(J),ULCENT(J),PSI(J),CAL(J),J=1,IN
358      *TYPES)
359      DO 45 J=1,INT(NLINES)
360      IF (PSI(J).LE.0.0) CALL ABORT(WHERE*6,"AIRCRAFT OVERPRESSURE MUST
361      *BE GREATER THAN 0")
362      IF (CAL(J).LE.0.0) CALL ABORT(WHERE*7,"THE VULNERABILITY LEVEL OF T
363      *HE AIRCRAFT TO FREE-FIELD THERMAL ENERGY CANNOT BE LESS THAN OR EQ
364      *UAL TO 0")
365      IF (ULCENT(J).LE.0.0) CALL ABORT(WHERE*8,"THE DISTANCE FROM ORIGIN
366      *RELEASE POINT TO THE TURN POINT MUST BE GREATER THAN 0")
367      IF (ORTIME(J).LE.0.0) CALL ABORT(WHERE*9,"THE BRAKE RELEASE TIME MU
368      *ST BE GREATER THAN 0")
369      45 CONTINUE
370      *WRITE TAKEOFF INTERVALS MAIN1X
371      CALL PAGE(-MAX(NLINES,INT(NLINES-3))
372      WRITE(6,6040)((I,IM,JLU,KFL,UELAC(I,IM,JLU,KFL)*KFL=1,INT(NLINES)).
373      * JLU=1,INT(NLINES),I,IM=1,MAXWAY)
374      DO 47 IM=1,MAXWAY
375      DO 47 JLU=1,INT(NLINES)
376      IF (UELAC(IM,JLU,KFL).LE.0.0) CALL ABORT(WHERE*10,"THE TAKE-OFF I
377      *NTERVALS BETWEEN AIRCRAFT TYPES MUST BE GREATER THAN 0")
378      47 CONTINUE
379      *END SUB LOCATION CARDS
380      NMPNS=0
381      MAXMPS=0
382
334 QUANTA
335 QUANTA
336 QUANTA
337 QUANTA
338 QUANTA
339 QUANTA
340 QUANTA
341 QUANTA
342 QUANTA
343 QUANTA
344 QUANTA
345 QUANTA
346 QUANTA
347 QUANTA
348 QUANTA
349 QUANTA
350 QUANTA
351 QUANTA
352 QUANTA
353 QUANTA
354 QUANTA
355 QUANTA
356 QUANTA
357 QUANTA
358 QUANTA
359 QUANTA
360 QUANTA
361 QUANTA
362 QUANTA
363 QUANTA
364 QUANTA
365 QUANTA
366 QUANTA
367 QUANTA
368 QUANTA
369 QUANTA
370 QUANTA
371 QUANTA
372 QUANTA
373 QUANTA
374 QUANTA
375 QUANTA
376 QUANTA
377 QUANTA
378 QUANTA
379 QUANTA
380 QUANTA
381 QUANTA
382 QUANTA
383 QUANTA

```

```

383      DO 50 I=1,NSUBS
384      READ(5,5000)SUBLAT(I),SUBLONG(I),ISUBS(I),NMPS(I),MYPE(I)
385      MAXMPS=MAX0(MAXMPS,NMPS(I))
386      NMPS=NMPS+NMPS(I)
387      50 CONTINUE
388      WRITE SUB LOCATION CARDS
389      CALL PAGE(-NSUBS-2)
390      WRITE(6,6000)(I,SUBLAT(I),SUBLONG(I),ISUBS(I),NMPS(I),MYPE(I),
391      *I=1,NSUBS)
392      DO 55 I=1,NSUBS
393      IF (SUBLAT(I).LT.0.0).OR.(SUBLONG(I).LT.0.0))CALL ABORT(WHERE,I,"I
394      THE SUB LATITUDE AND LONGITUDE MUST BE GREATER THAN 0")
395      IF (ISUBS(I).LE.0) CALL ABORT(WHERE,I,"THE NUMBER OF SUBS AT A GIV
396      IEN LOCATION MUST BE GREATER THAN 0")
397      IF (NMPS(I).LE.0) CALL ABORT(WHERE,I,"THE NUMBER OF MISSILES ON A
398      IL SUBS AT A GIVEN LOCATION MUST BE GREATER THAN 0")
399      IF (MYPE(I).LE.0).OR.(MYPE(I).GT.1)CALL ABORT(WHERE,I,"THE
400      IE TYPE OF SUB MUST BE GREATER THAN 0 AND NOT GREATER THAN THE TOTA
401      EL NUMBER OF SUB TYPES")
402      55 CONTINUE
403      CALL PAGE(-MYPES-2)
404      DO 70 J=1,MYPES
405      NSUBT=0
406      DO 60 I=1,NSUBS
407      IF (MYPE(I).EQ.J)NSUBT=NSUBT+ISUBS(I)
408      60 CONTINUE
409      WRITE(6,6000)J,NSUBT
410      70 CONTINUE
411      READ SUB TYPE PARAMETERS AND PROFILES
412      DO 80 I=1,MYPES
413      READ(5,5000)DELTM(I),REL(I),KNOMIN(I),KNOMAX(I),YIELD(I),BURST(I),
414      *MVS(I)
415      READ(5,5000)MPM,(PMTIME(J),J=1,MNOM(J),J=1,MPM)
416      IF (MPM.GT.MAXPM) CALL ABORT(WHERE,I,"NUMBER OF TIME/DISTANCE PAIR
417      IS CANNOT BE GREATER THAN MAXPM")
418      MPMOF(I)=MPM
419      80 CONTINUE
420      WRITE SUB TYPE PARAMETERS AND PROFILES
421      CALL PAGE(-MYPES-3)
422      WRITE(6,6000)(J,DELTM(J),REL(J),KNOMIN(J),KNOMAX(J),YIELD(J),BURST
423      *(J),MVS(J),J=1,MYPES)
424      DO 85 J=1,MYPES
425      IF (DELTM(J).LE.0.0) CALL ABORT(WHERE,I,"THE TIME INTERVAL BETWEEN
426      SALVO LAUNCHES MUST BE GREATER THAN 0")
427      IF ((REL(J).LE.0.0).OR.(REL(J).GE.1.0))CALL ABORT(WHERE,I,"THE REL
428      IABILITY OF A MISSILE MUST BE GREATER THAN 0 AND LESS THAN 1")
429      IF (KNOMAX(J).LE.KNOMIN(J)) CALL ABORT(WHERE,I,"THE MAXIMUM RANGE
430      OF THE MISSILE MUST BE GREATER THAN ITS MINIMUM RANGE")
431      IF (YIELD(J).LE.0.0) CALL ABORT(WHERE,I,"THE YIELD OF THE MISSILE
432

```

QUANTA 384
 QUANTA 385
 QUANTA 386
 QUANTA 387
 QUANTA 388
 QUANTA 389
 QUANTA 390
 QUANTA 391
 QUANTA 392
 QUANTA 393
 QUANTA 394
 QUANTA 395
 QUANTA 396
 QUANTA 397
 QUANTA 398
 QUANTA 399
 QUANTA 400
 QUANTA 401
 QUANTA 402
 QUANTA 403
 QUANTA 404
 QUANTA 405
 QUANTA 406
 QUANTA 407
 QUANTA 408
 QUANTA 409
 QUANTA 410
 QUANTA 411
 QUANTA 412
 QUANTA 413
 QUANTA 414
 QUANTA 415
 QUANTA 416
 QUANTA 417
 QUANTA 418
 QUANTA 419
 QUANTA 420
 QUANTA 421
 QUANTA 422
 QUANTA 423
 QUANTA 424
 QUANTA 425
 QUANTA 426
 QUANTA 427
 QUANTA 428
 QUANTA 429
 QUANTA 430
 QUANTA 431
 QUANTA 432
 QUANTA 433

PAGE 13

404	0110F10.4.3A.0B.4.9A.F10.3.F10.3.F10.4.4A.F10.2.10A.F4.1)	QUAN:IA	484
405	CUBU FURMAT (/13M MISSILE TYPE.11X.4MTIME.5A.CHANGE/113.5A.2F10.4/(10A	QUAN:IA	485
406	1.2F10.4))	QUAN:IA	486
407	END	QUAN:IA	487

```

487 SUBROUTINE PROCES(PST,CAL,YIELD,NTYPES,MTPES,NPAP,NXAP,SDU,
488 * AXP,VS,VEL,ACCEL,NUDATA,PLPP,FTSA,PLRT,BURST)
489 QUANTA
490 QUANTA
491 QUANTA
492 QUANTA
493 QUANTA
494 QUANTA
495 QUANTA
496 QUANTA
497 QUANTA
498 QUANTA
499 QUANTA
500 QUANTA
501 QUANTA
502 QUANTA
503 QUANTA
504 QUANTA
505 QUANTA
506 QUANTA
507 QUANTA
508 QUANTA
509 QUANTA
510 QUANTA
511 QUANTA
512 QUANTA
513 QUANTA
514 QUANTA
515 QUANTA
516 QUANTA
517 QUANTA
518 QUANTA
519 QUANTA
520 QUANTA
521 QUANTA
522 QUANTA
523 QUANTA
524 QUANTA
525 QUANTA
526 QUANTA
527 QUANTA
528 QUANTA
529 QUANTA
530 QUANTA
531 QUANTA
532 QUANTA
533 QUANTA
534 QUANTA
535 QUANTA
536 QUANTA
537 QUANTA

SUBROUTINE PROCES(PST,CAL,YIELD,NTYPES,MTPES,NPAP,NXAP,SDU,
* AXP,VS,VEL,ACCEL,NUDATA,PLPP,FTSA,PLRT,BURST)
DIMENSION PST(NTYPES),CAL(NTYPES),YIELD(NTYPES),S(NAP,NTYPES),
U(NAP,NTYPES),A(NAP,NTYPES),F(NAP,NTYPES),O(NAP,NTYPES),VS
(NAP,NTYPES),VEL(NAP,NTYPES),ACCEL(NAP,NTYPES),NUDATA(NTYPES),
PLPP(NTYPES,MTPES),FTSA(NTYPES,MTPES),PLRT(NTYPES,MTPES),
BURST(MTPES)
THIS SUBROUTINE IS USED TO INPUT DATA FOR AIRCRAFT AND IS USED
TO GENERATE NOISE EFFECTS GEOMETRICALLY FOR SPECIFIED HAPNESS
LEVELS VIA SUBROUTINES SPARK AND SNAKE
DIMENSION JUM(13)
COMMON/PASS/KHU,VIS,PZ,MIE
LOGICAL SNWE,VALU
LEVEL 25 PST,CAL,YIELD,U,S,A,F,O,VS,VEL,ACCEL,NUDATA,
PLPP,FTSA,PLRT,BURST
DATA UELT,CMIT,CFM /1.0001,3.2000,0.3000/
DATA WHERE/BNRPHGES/
DO 130 I=1,NTYPES
CALL PAGE (U)
HEAD (5,500)M2
HEAD (5,510)N
NUDATA(I)=M
IF (A2,LE,U,U) CALL ABOUT(WHERE,1,"THE LEVEL-OFF ALTITUDE OF THE AI
AIRCRAFT MUST BE GREATER THAN 0")
IF (N,O,(NPAP) CALL ABOUT(WHERE,2,"THE NUMBER OF DATA POINTS IN AIR
CRAFT PROFILE IS LARGER THAN NPAP")
ICAL=CAL(I)
WRITE (6,600),NTYPES
WRITE (6,610) PST(I),ICAL
WRITE (6,620)
CALL PAGE (17)
THE FOLLOWING READS AIRCRAFT PROFILE INPUT DATA AND ADJUSTS THE
ALTITUDE (IF NECESSARY) FOR USE BY AUSE WHICH REQUIRES IT TO BE
STRICTLY MONOTONICALLY INCREASING. A-ALTITUDE (FEET);
P-GROUND RANGE (FEET); J-TIME (SEC); VEL-MACH NUMBER;
ACCEL-ACCELERATION (FEET/SEC/SEC). THE VELOCITY OF SOUND (VS)
GIVEN ALTITUDE IS CALCULATED BY FUNCTION SSP.
HEAD (5,520) F(1),O(1),A(1),VEL(1),ACCEL(1)
IF (A(1),LT,0.0) CALL ABOUT(WHERE,3,"THE AIRCRAFT ALTITUDE CANNOT
BE LESS THAN 0")
IF (F(1),LT,0.0) CALL ABOUT(WHERE,4,"THE AIRCRAFT GROUND RANGE FM
THE BRAKE RELEASE POINT CANNOT BE LESS THAN 0")
IF (J(1),LT,0.0) CALL ABOUT(WHERE,5,"THE TIME FROM THE BRAKE RELEASE
POINT CANNOT BE LESS THAN 0")
AL=CFTMA(1,1)
VS(1)=CMIT*SSP(AL)

```


PAGE 18

```

507 WRITE (6,600) VMACH,FMACH
508 FMACH=VMACH/PMACH
509 PMACH=VSM/PMACH
510 WRITE (6,600) FMACH,FMACH
511 IF (VMACH.GT.FMACH) CALL ABORT('WHERE,16,"MACH AT LEVEL OFF CANNOT BE
512 GREATER THAN FINAL MACH"')
513
514 C
515 C EQUATIONS OF MOTION:
516 C
517 ACCZ=ACCAZ/2.0
518 MANAZ=AUSC(AZ,A(1,1),F(1,1),N)
519 TIMAZ=AUSC(AZ,A(1,1),G(1,1),N)
520 UU 50 J=1,N
521 INDEX=J
522 IF (F(J,1).GE.MANAZ .OR. G(J,1).GE.TIMAZ) GO TO 60
523 C THE FIRST PART OF THE S AND U ARRAYS IS SIMPLY THE FLIGHT
524 C PROFILE (P,U) READ IN
525 U(J,1)=F(J,1)
526 S(J,1)=G(J,1)
527 C
528 C
529 C
530 C
531 C
532 C
533 C
534 C
535 C
536 C
537 C
538 C
539 C
540 C
541 C
542 C
543 C
544 C
545 C
546 C
547 C
548 C
549 C
550 C
551 C
552 C
553 C
554 C
555 C
556 C
557 C
558 C
559 C
560 C
561 C
562 C
563 C
564 C
565 C
566 C
567 C
568 C
569 C
570 C
571 C
572 C
573 C
574 C
575 C
576 C
577 C
578 C
579 C
580 C
581 C
582 C
583 C
584 C
585 C
586 C
587 C
588 C
589 C
590 C
591 C
592 C
593 C
594 C
595 C
596 C
597 C
598 C
599 C
600 C
601 C
602 C
603 C
604 C
605 C
606 C
607 C
608 C
609 C
610 C
611 C
612 C
613 C
614 C
615 C
616 C
617 C
618 C
619 C
620 C
621 C
622 C
623 C
624 C
625 C
626 C
627 C
628 C
629 C
630 C
631 C
632 C
633 C
634 C
635 C
636 C
637 C

```

[illegible]

PAGE 20

```

007 030 FORMAT (12X, 13X, (6X, 15.0), 2(6X, 6.3))
008 040 FORMAT (//10X, INITIAL ALTITUDE OF MAXIMUM MACH *F8.1* FEET/)
009 050 FORMAT (//11X, JCHDATA COMPUTED FOR MACHAT TYPE *12.0* AT A TERMINAL
010 060 * ALTITUDE OF *F8.1* FEET/)
011 070 FORMAT (16X, VELOCITY OF SOUND *F8.2* FEET/SEC/)
012 080 FORMAT (11X, ACCELERATION COMPONENT *F8.3* FEET/SEC/SEC/)
013 090 FORMAT (22X, MACH NUMBER *F8.3* TERMINAL *F8.3/)
014 095 FORMAT (25X, VELOCITY *F8.2* FEET/SEC FINAL *F8.2*
015 096 *F8.2* FEET/SEC/)
016 097 FORMAT (17X, JCHMACHMACH RANGE (FEET) FLIGHT TIME (SEC), 10X, JCHMACHMACH
017 098 INU RANGE (NM) FLIGHT TIME (MIN)/)
019 710 FORMAT (20X, 16.3, 4X, 16.8, 19X, 16.0, 4X, 16.0)
020 END

```

007
008
009
010
011
012
013
014
015
016
017
018
019
020

P 02 22

750	5020 FORMALIN METH. DETECTION, IS 5020.100 100 FROM DETECTION. LINEAL WITH	5020.100	751
751	• = 5020.100 5020.100	5020.100	752
752	5030 FORMALIN (//)	5030.100	753
753	5040 FORMALIN DETECTION, IS 5040.100 100 FROM 5040.100 5040.100	5040.100	754
754	5050.100 PLUS 5050.100 5050.100	5050.100	755
755	END	5050.100	756

[illegible]

PAGE 20

```

097 SUBROUTINE MEAFALS(FN,XA,BUM,BUM,NTUL,XTUL,XPTUL,XPLAGS,PAA)
098
099 THIS SUBROUTINE FINDS THE REAL ROOTS OF A FUNCTION GIVEN AS
100 AN EXTERNAL. THE USER MUST GIVE BOUNDARY POINTS THAT LIE ON
101 EITHER SIDE OF THE ROOT. THESE POINTS ARE P1 AND P2.
102
103 GIVEN VARIABLES ARE:
104
105 FN - THE FUNCTION GIVEN BY THE USER
106 XA - THE LEFTMOST BOUND ON THE ROOT
107 BUM - THE RIGHTMOST BOUND ON THE ROOT
108 NTUL - THE MAXIMUM NUMBER OF ITERATIONS ALLOWED BY USER
109 XTUL - THE TOLERANCE SPECIFIED BY USER FOR THE
110 DIFFERENCE BETWEEN TWO ITERATIONS
111 PTUL - THE MAXIMUM ERROR IN THE FUNCTION AT THE ROOT
112 P - THE VALUE OF THE ROOT
113 IFLAG - A FLAG TO TELL THE USER HOW THE PROGRAM RAN
114 0. THE ROOT WAS FOUND TO WITHIN THE USERS
115 TOLERANCE.
116 1. THE ENDPOINTS DO NOT BOUND A REAL ROOT
117 2. THE ROUTINE DID NOT CONVERGE TO WITHIN
118 THE USERS TOLERANCE.
119
120 LEVEL 2, PAA
121
122 A=0.0
123 DEB=0
124 P=FN(XA,PAA)
125 J=FN(BUM,PAA)
126
127 CHECK TO SEE IF THE ENDPOINTS BOUND A ROOT.
128
129 SP=SIGN(1.0,P)
130 IF ((SP*J).LE.0.0001) GO TO 10
131 IFLAG=1
132 GO TO 50
133 IFLAG=0
134
135 W=A
136 F=0
137 DO 30 N=1,NTUL
138
139 CHECK FOR SMALL ENOUGH INTERVAL BETWEEN ROOTS
140 IF ((ABS((B-A)/C).LE.XTUL) GO TO 50
141
142 CHECK FOR SMALL ENOUGH FUNCTION VALUES
143 IF ((ABS(FM).LE.XPTUL) GO TO 50
144 W=((F*B)-(J*A))/(F-G)
145 SP=SIGN(1.0,FM)
146 F=FN(W,PAA)
147 IF ((SP*F).LT.0.0) GO TO 20
148
149 CALCULATE NEW A, P, AND CUT G IN HALF.
150 A=W
151 P=F
152 IF ((5*W*F).GT.0.0) G=0.5
153 GO TO 30
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

PAGE 21

```
946 C
947 C
948 C CALCULATE NEW D, G, AND CUT F IN HALF.
949 C D=D*
950 C IF ((S#*F#).GT.0.) F=F/2.
951 C CONTINUE
952 C
953 C SET IFLAG TO 2 IF THE PROGRAM DIDN'T CONVERGE WITHIN LIMITS
954 C IFLAG=2
955 C RETURN
956 C END
```

```
QUANTA
QUANTA
QUANTA
QUANTA
QUANTA
QUANTA
QUANTA
QUANTA
QUANTA
QUANTA
QUANTA
QUANTA
```

```
947
948
949
950
951
952
953
954
955
956
957
```

PAGE 28

```
957 C FUNCTION TPPI(X,F,A)
958 C TRIPLE POINT PATH INTERCEPT.....USED IN OUTLINE
959 COMMON /STUFF/
960 COMMON /FLICK/UL(5),GU,CTP
961 COMMON /CSTIME/UC(4),NAF
962 LEVEL 2,F,A
963 M1=GU*CTP*(ABS(Q-X)/60-1.)*.01
964 NC=ABS(C(A,F,A,NAF))
965 TPPI=M1-M2
966 RETURN
967 END
```

```
QUANTA
QUANTA
QUANTA
QUANTA
QUANTA
QUANTA
QUANTA
QUANTA
QUANTA
QUANTA
QUANTA
```

```
958
959
960
961
962
963
964
965
966
967
968
```

PAGE 29

```

908      REAL FUNCTION MS(X*0.5)
909      C      MACH SHUCK.....USED IN ULINE
910      COMMON /STUFF/J*HTE
911      COMMON /JUSTIME/MS*HAKUIM*HAKUUP*HAF
912      LEVEL 0*0.5
913      HAF=AUSC(X*0.5*HAF)*HTE
914      KORD=1
915      CALL BLINT(MS*HAF*HTE*HTE*HTE)
916      MS=SUMT(U-A)*0.2*MAC(0.2)
917      MS=TPPRUP(S*HTE*HTE*HTE)-HAKUUP
918      RETURN
919      END

```

```

QUANTA 909
QUANTA 910
QUANTA 911
QUANTA 912
QUANTA 913
QUANTA 914
QUANTA 915
QUANTA 916
QUANTA 917
QUANTA 918
QUANTA 919
QUANTA 920

```

```

980 FUNCTION FAS(A,F,A)
981   FREE AIM SHUC.....USED IN UTLIN
982   COMMON /STUFF/WHITE
983   COMMON /CSTIME/HB,NAMUTH,MAHUU,P,MINAT
984   LEVEL C,F,A
985   MAC=AUSC(A,F,A,NAT)-NTE
986   KUU=1
987   CALL BLINK(MB,MAC,HT,NT,KUU)
988   SH=SQRT((J-A)*C*(MAC-HB)*C)
989   FAS=MF*CUUP(SH)-MAHUU
990   RETURN
991   END

```

QUA1A	961
QUA1A	962
QUA1A	963
QUA1A	964
QUA1A	965
QUA1A	966
QUA1A	967
QUA1A	968
QUA1A	969
QUA1A	970
QUA1A	971
QUA1A	972

```

992      FUNCTION TRMML(XAP,A)
993      C  THERMAL EFFECTS.....USED IN ULINE
994      COMMON /STUFF/UMTE
995      COMMON /CSTIME/HO,HAUTH,MAHLOP,MANAP
996      COMMON /PASS/MU,VIS,PZ
997      LEVEL 2,P,A
998      LOGICAL SNEW,VALID
999      MCL=ACSL(XAP,A,MANAP)-MTE
1000      CALL SIGNEZ(ABS(XAP-A),MA,A,U,U,0.0,UMH,MTE,MU,VIS,PZ,U,U,0.0,U,0.0)
1001      1    CAL,UMUUM,-1.0,UM,VALID,SNEW)
1002      TRMML=UCL-HAUTH
1003      RETURN
1004      END

```


PAGE 33

1055 40
1056
1057
1058
1059
1060
1061
1062

ITB=PROFLU(XBMSI,U(1,ITYPE),S(1,ITYPE),NSU)
I=ITO-TSA
I=AMAX1(U,IT)
UIS=XBMSI-PROFLU(T,S(1,ITYPE),U(1,ITYPE),NSU)
NOW FIND THE TOTAL LETHAL AREA
UIA=EA=THUPA(KUP,UNM,KTH,UIS,THEIA)
RETURN
END

QUANTA 1056
QUANTA 1057
QUANTA 1058
QUANTA 1059
QUANTA 1060
QUANTA 1061
QUANTA 1062
QUANTA 1063

```

1063 FUNCTION THUPA(RUP,QUANT,THETA)
1064 LEVEL 200
1065
1066 PURPOSE
1067 TO COMPUTE THE AREA OF THE FLOURE FORMED BY THE UNION OF THE
1068 OVER PRESSURE "SQUISH" AND THE THERMAL CIRCLE INTERSECTED WITH
1069 THE SPACE TO THE SIDE OF THE LINE OF FLIGHT WITH DECREASING
1070 THETA.
1071
1072 LOGICAL EXIST,CEXIST
1073 DATA NJINC /500/
1074 DATA EPSLON/-1.0E-8/
1075 INITIALIZE.
1076
1077 THUPA=0.0
1078 THETA=ABS(THETA)
1079 RUPSU=RUP*PRUP
1080 RTHSU=RTH*PRTH
1081 SI=0.
1082 KULU=AMMAL(U*PRUP-PRUJ,U*PRTH)
1083 KULU=AMMAL(U*PRUP-PRUJ,U*PRTH)
1084 KULU=U*PRUP-PRUJ
1085 KULU=U*PRUP-PRUJ
1086 KULU=U*PRTH
1087 KULU=U*PRTH
1088 KULU=U*PRTH
1089
1090 IF (U*GE*PRTH).AND.(U*GE*PRUP) THEN TA=AMIN(THETA,AMAX(ASIN(RUP/U),
1091 1 ASIN(RTH/U)))
1092 U*THETA=THETA /FLUAT(NJINC)
1093
1094 STARTING AT ZERO INCREMENT SI UP TO THETA COMPUTING AREAS AT
1095 EACH INCREMENT
1096
1097 DO 40 J=1,NJINC
1098 SET UP THE ANGLE.
1099
1100 SJ=FLUAT(J)*U*THETA
1101 CUSI=COS(SI)*U
1102 SINI=SIN(SI)*U
1103 SINIU=SIN(SI)*SINI
1104 COMPUTE THE POINTS OF INTERSECTION.
1105 A=1.0E295
1106 B=-1.0E295
1107 EXIST=.FALSE.
1108
1109 DESC=RUPSU-SINIU
1110 IF (DESC.GT.EPSLON.AND.DESC.LT.0.0) DESC = 0.0
1111 IF (DESC.LT.0.0) DO 10
1112 DESC=SUM(DESC)

```

QUANTA 1064
 QUANTA 1065
 QUANTA 1066
 QUANTA 1067
 QUANTA 1068
 QUANTA 1069
 QUANTA 1070
 QUANTA 1071
 QUANTA 1072
 QUANTA 1073
 QUANTA 1074
 QUANTA 1075
 QUANTA 1076
 QUANTA 1077
 QUANTA 1078
 QUANTA 1079
 QUANTA 1080
 QUANTA 1081
 QUANTA 1082
 QUANTA 1083
 QUANTA 1084
 QUANTA 1085
 QUANTA 1086
 QUANTA 1087
 QUANTA 1088
 QUANTA 1089
 QUANTA 1090
 QUANTA 1091
 QUANTA 1092
 QUANTA 1093
 QUANTA 1094
 QUANTA 1095
 QUANTA 1096
 QUANTA 1097
 QUANTA 1098
 QUANTA 1099
 QUANTA 1100
 QUANTA 1101
 QUANTA 1102
 QUANTA 1103
 QUANTA 1104
 QUANTA 1105
 QUANTA 1106
 QUANTA 1107
 QUANTA 1108
 QUANTA 1109
 QUANTA 1110
 QUANTA 1111
 QUANTA 1112

Page 35

```

1113      W=CROSSI*PMUJ-UESC
1114      B=CROSSI*PPUJ+UESC
1115      W=AMAX1(W,U)
1116      B=AMAX1(B,U)
1117      BEASTI=BEAST
1118
1119      IU=C
1120      U=1.0E2Y5
1121      UESTI=PALEDE
1122
1123      UESC=HMSU-SINSU
1124      IF(UESC.GT.0.5) UESC=0.5
1125      IF(UESC.LT.0.5) UESC=0.5
1126      UESC=SUMT(UESC)
1127      W=CROSSI*SUMT(UESC)
1128      B=AMAX1(W,U)
1129      B=AMAX1(B,U)
1130
1131      C
1132      C
1133      C
1134      C
1135      C
1136      C
1137      C
1138      C
1139      C
1140      C
1141      C
1142      C
1143      C
1144      C
1145      C
1146      C
1147      C
1148      C
1149      C
1150      C
1151      C
1152      C
1153      C
1154      C

```

PAGE 36

```

1154 SUBROUTINE ZERO (NTOTS,NSUBS)
1155 C
1156 C
1157 C
1158 C
1159 C
1160 C
1161 C
1162 C
1163 C
1164 C
1165 C
1166 C
1167 C
1168 C
1169 C
1170 C
1171 C
1172 C

```

INITIALIZE PLAS TO ZERO
 COMMON/MNGFLU/BASENU(4/),NOLTS
 DATA NUBASE/4/
 DATA NOLTS/50/
 CHECK FOR TOO MANY BASES
 IF (NTOTS.GT.NUBASE) STOP "IN ZERO"
 CHECK FOR TOO MANY SUBS
 IF (NSUBS.GT.NOLTS) STOP "IN ZERO"
 DO 10 I=1,NUBASE
 BASENU(I)=0.
 CONTINUE
 RETURN
 END

QUANTA 1155
 QUANTA 1156
 QUANTA 1157
 QUANTA 1158
 QUANTA 1159
 QUANTA 1160
 QUANTA 1161
 QUANTA 1162
 QUANTA 1163
 QUANTA 1164
 QUANTA 1165
 QUANTA 1166
 QUANTA 1167
 QUANTA 1168
 QUANTA 1169
 QUANTA 1170
 QUANTA 1171
 QUANTA 1172

```

1172 SUBROUTINE SKVMAT(SKVLN,SKVPC,BRTIME,BRT,NRWAYS,ISEU,DELTA,
1173   * TOTLAT,TOTLNG,SUBLAT,SUBLNG,MTYPE,MEL,KVS,RNGMIN,RNGMAX,MPHUF,
1174   * FAKG,FMIME,NMPS,DELTA,AREAL,DIAM,UTCENT,ISU,RA,UGTS,NMPS,
1175   * MANAP,NTYPES,MTPES,NTGTS,NACT,MANWAT,NSUBS,MANPR,NSU,VAL,
1176   * DIMENSION SKVLN(NMPS,MANAP),SKVPC(NMPS,MANAP),BRTIME(NTYPES),
1177   * BRT(MANAP),NRWAYS(NTGTS),ISEU(NACT),DELTA(NRWAY,NTYPES,NTYPES),
1178   * TOTLAT(NTGTS),TOTLNG(NTGTS),SUBLAT(NSUBS),SUBLNG(NSUBS),
1179   * MTYPE(NSUBS),MEL(NTYPES),MVS(NTYPES),RNGMIN(NTYPES),RNGMAX
1180   * (NTYPES),MPHUF(NTYPES),NMPS(NRWAY,MANPR,NTYPES),FMIME(MANPR,NTYPES),
1181   * NMPS(NSUBS),DELTA(MTYPE),AREAL(UG,NTYPES,NTYPES),
1182   * DIAM(UG,NTYPES,NTYPES),UTCENT(NTYPES),UGTS(UG),RA(NACT),
1183   * SINSU(NTYPES),UINSU(NTYPES),VAL(NTGTS,NMPS),
1184   * LEVEL(UG,SKVLN,SKVPC,BRTIME,BRT,NRWAYS,ISEU,DELTA,TOTLAT,
1185   * TOTLNG,SUBLAT,SUBLNG,MTYPE,MEL,KVS,RNGMAX,RNGMIN,MPHUF,FMIME,
1186   * FMIME,NMPS,DELTA,AREAL,DIAM,UTCENT,ISU,UGTS,VAL,RA,
1187   * DATA P1/3,1-159205/
1188   CALL PAGE (U)
1189   LAST=0
1190   DO 140 I=1,NTGTS
1191     C
1192     C COMPUTE SURVIVABILITY OF EACH AIRCRAFT ON BASE FROM ALL WEAPONS
1193     C
1194     C
1195     IFST=LAST+1
1196     DO 10 JT=1,NTYPES
1197       LAST=LAST+INT(VAL(I,JT)+.1)
1198       NAC=LAST-IFST+1
1199       C COMPUTE THE BRAKE RELEASE TIMES(BRT) OF ALL AIRCRAFT
1200       C ON BASE I.
1201       NMV=NRWAYS(I)
1202       ITYPE=ISEU(IFST)
1203       BRT(I)=BRTIME(ITYPE)
1204       C IF THERE IS ONLY ONE PLANE ON BASE, DUNT GO THRU LOOP
1205       IF (NAC.EQ.1) GO TO 30
1206       IAC=IFST
1207       DO 20 NI=2,NAC
1208         IAC=IAC+1
1209         ITP1=ITYPE
1210         ITP2=ISLU(IAC)
1211         BRT(NI)=BRT(NI-1)+DELTA(NMV,ITYP1,ITYPE)
1212       CONTINUE
1213       C COMPUTE THE AVERAGE DISTANCE TO THE TURN POINT
1214       SUM=0.
1215       DO 35 IAC=IFST,LAST
1216         JT=ISEU(IAC)
1217         SUM=SUM+UTCENT(JT)
1218       CONTINUE
1219       AVUTC=SUM/FLOAT(NAC)
1220     C
1221     C BEGIN THE OUTER LOOP FOR SUBMARINE LOCATIONS
1222

```

PAGE 38

```

1222 C      ISALVU=U
1223      UU 120 KSUB=1:NSUBS
1224      DIST=U1ST(TOTLAT(1),TOTLNG(1),SUBLAT(KSUB),SUBLNG(KSUB))
1225      MISTYPE=MIST(KSUB)
1226      NMIS=NMPS(KSUB)
1227      CALL PAGE (-C)
1228      WHITE (6,010) 1:KSUB,DIST,AVDTC
1229      IF THE DISTANCE BETWEEN SUB KSUB AND BASE 1) IS OUT OF
1230      RANGE, MAKE THE SURVIVABILITY OF THE AIRCRAFT 1.
1231      IF (DIST-GE-MINGMIN(MISTYP),AND,DIST-LF-MNGMAX(MISTYP)) GO TO 50
1232      CALL SET(1:KSUB)
1233      LST=ISALVU*NMIS
1234      ISALVU=ISALVU*1
1235      UU 40 NI=1:MAC
1236      UU 40 N=ISALVU,LST
1237      SHVPC(N,N1)=.
1238      SHVLN(N,N1)=1.
1239      CONTINUE
1240      ISALVU=LST
1241      CONTINUE TO THE NEXT SUB LOCATION
1242      GO TO 130
1243
1244 C      COMPUTE THE FIRST WEAPON'S TIME OF ARRIVAL (TOAR) IN MINUTES
1245      MPROFAX=MPROF(MISTYP)
1246      TOAR = AOSC(DIST,FMRNG(1,MISTYP),FMTIME(1,MISTYP),MPROFAX)
1247
1248 C      START THE INNER LOOP FOR MISSILE SALVUS AT KSUB SUB LOC.
1249
1250 C      KVSMEV(S,MISTYP)
1251      HELMEREL(MISTYP)
1252      UU 120 KSALE=1:NMIS
1253      ISALVU=ISALVU*1
1254      TOA=TOAR+PLAT(KSALE-1)*DELTIM(MISTYP)
1255      COMPUTE THE GROUND RANGE, FROM OFARE RELEASE, OF EACH
1256      AIRCRAFT (KAR) AT THE TIME OF WEAPON ARRIVAL
1257      NI=0
1258      MFA=0.
1259      UU 60 IAC=IPKST*LAST
1260      NI=NI+1
1261      JT=ISEU(IAC)
1262      MTUA=TOA-BRT(IAC)
1263      KA = PROFLU(MTUA,S(1,JT),U(1,JT),NSU)
1264      KAR(NI)=KA
1265      MFA=MAA1(MFA,KFA)
1266      CONTINUE
1267      RANGE OF THE FARTHEST AIRCRAFT IF RELATIVE TO THE TURN PT
1268      MFA=MFA-AVDTC
1269      IF (KSALE=1) GO TO 65
1270      WHITE(6,020)TUA,ISALVU,(KAR(IAC),IAC=1:NI)
1271

```

QUANTA 1223
 QUANTA 1224
 QUANTA 1225
 QUANTA 1226
 QUANTA 1227
 QUANTA 1228
 QUANTA 1229
 QUANTA 1230
 QUANTA 1231
 QUANTA 1232
 QUANTA 1233
 QUANTA 1234
 QUANTA 1235
 QUANTA 1236
 QUANTA 1237
 QUANTA 1238
 QUANTA 1239
 QUANTA 1240
 QUANTA 1241
 QUANTA 1242
 QUANTA 1243
 QUANTA 1244
 QUANTA 1245
 QUANTA 1246
 QUANTA 1247
 QUANTA 1248
 QUANTA 1249
 QUANTA 1250
 QUANTA 1251
 QUANTA 1252
 QUANTA 1253
 QUANTA 1254
 QUANTA 1255
 QUANTA 1256
 QUANTA 1257
 QUANTA 1258
 QUANTA 1259
 QUANTA 1260
 QUANTA 1261
 QUANTA 1262
 QUANTA 1263
 QUANTA 1264
 QUANTA 1265
 QUANTA 1266
 QUANTA 1267
 QUANTA 1268
 QUANTA 1269
 QUANTA 1270
 QUANTA 1271
 QUANTA 1272

```

1272 CALL PADE (2*(NAC-1)/10)
1273 CONTINUE
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000

```

CALL PADE (2*(NAC-1)/10)
 CONTINUE

 COMPUTE THE SURVIVABILITY FOR EACH AIRCRAFT ON BASE 1.
 THE SURVIVABILITY MATRICES FOR A WEAPON PLACED:
 CASE 1-- ON THE RUNWAY (SKVLEN)-ALL MVS WILL BE EVENLY DISTRIBUTED ON THE RUNWAY (FROM THE BRAKE RELEASE POINT TO THE TURN POINT).
 CASE 2-- PAST THE CENTROID (SKVPC)-MVS WILL BE UNIFORMLY DISTRIBUTED IN THE AREA OF THE CIRCLE SWEEPED OUT BY THE FARTHEST AIRCRAFT MINUS THE RUNWAY.

 JLEN=-1
 JTPC=-1
 NI=0
 DO 120 IAC=1,MST,LAST
 NI=NI+1
 JT=ISCU(IAC)
 COMPUTE RADIUS OF A 180 TURN FOR THIS TYPE OF AC
 TP=PROFLC(AVUTC,UI(JT),S(1,JT),NSU)+1.
 JTP=PROFLC(TTP,S(1,JT),U(1,JT),NSU)-AVUTC
 TPN=2.*JTP/PI
 IF (KAGINI).GT.AVUTC+TMN) GO TO 90
 IF (JT.LE.JLEN) GO TO 110
 JLEN=JT
 THIS PLANE IS VULNERABLE TO A LINE ATTACK.
 CONSIDER A "TOTAL" LETHAL DIAMETER FROM FWH. TO AVUTC
 SUM=0.
 AINT=AVUTC/MVSM
 NINT=KASM
 A=XINT+.5
 TPE=-AVUTC
 DO 80 IN=1,NINT
 A=X*INT
 BM = AUSC(X,PTS,DIAM(1+JT,MISTYP),13)
 TP = AUSC(X,PTS,DIAM(13+JT,MISTYP),13)
 SUM=SUM+BM*TP
 SUM=SUM-AMAA1(0.,TPE+BM-XINT)
 TPE=TP
 CONTINUE
 SUM=SUM-AMAA1(0.,X*TP-AVUTC-TMN)
 FOR CASE 1- THE SURV. DEPENDS ON THE RELIABILITY OF THE MISSILE.
 THIS TOTAL LETHAL DIAMETER AND THE LENGTH OF THE "RUNWAY".
 FOR CASE 2- SURV=1.
 PR=SUM/(AVUTC+TMN)
 PR=AMINI(1.,PR)
 SKVLEN((SALVUINI)=1.-PR*OMELM
 SHVPC((SALVUINI)=1.
 CONTINUE TO THE NEXT AIRCRAFT
 GO TO 120

```

1322 C
1323 C
1324 C
1325 C
1326 C
1327 C
1328 C
1329 C
1330 C
1331 C
1332 C
1333 C
1334 C
1335 C
1336 C
1337 C
1338 C
1339 C
1340 C
1341 C
1342 C
1343 C
1344 C
1345 C
1346 C
1347 C
1348 C
1349 C
1350 C
1351 C
1352 C
1353 C
1354 C
1355 C
1356 C
1357 C
1358 C
1359 C
1360 C
1361 C
1362 C
1363 C
1364 C
1365 C
1366 C
1367 C
1368 C
1369 C
1370 C
1371 C
1372 C

IF (JT*EU*JTPC) GO TO 110
JTPC=JT
C THIS PLANE IS VULNERABLE TO A DAMAGE ATTACK.
C SINCE THE LETHAL AREAS DIMINISH AS THEY MOVE OUT FROM THE
C CENTROID, A WEIGHTED AVERAGE AREA WILL BE USED.
IN=2
SUM=0.0
DELTA=AREAAL(IN-1,JT,MISTYP)-AREAAL(IN,JT,MISTYP)
SUM=SUM+DELTA*(UPTS(IN)*UPTS(IN-1))**2)
IN=IN+1
IF (IN.LE.30 .AND. UPTS(IN).LT.RFA) GO TO 100
RHU=(RFA*UPTS(IN-1))**2
T1=AREAAL(IN-1,JT,MISTYP)*RHU
AK=AOSC(RFA*UPTS(IN-1)*RHU)
T2=AREAAL(4.0*(RFA**2)-RHU)
SUM=(SUM+T1+T2)/(4.0*RFA**2)
RHU=SUM*(SUM/PI)
BP=(RFA*TRN)*0.5
UTC=AVUTC
WAREA=CIRINT(RFA*RHU*BP) - CIRINT(TRN*RHU*BP)
IF (RFA.LE.UTC) TA=(RFA**2)*PI-ASIN(TRN/RFA) - PI*(TRN**2)/2.
* - TRN*(SQRT(RFA**2-TRN**2))
IF (RFA.GT.UTC .AND. RFA.LT.UTC*TRN) TA=RFA**2*PI - 2.0*TRN*UTC
* - CIRINT(RFA*TRN*UTC)
IF (RFA.GE.UTC*TRN) TA=PI*(RFA**2-TRN**2) - 2.0*TRN*UTC
FOR CASE 1- SURV=1
FOR CASE 2- THE SURV. DEPENDS ON THE RELIABILITY OF THE MISSILE,
THE WEIGHTED AREA, AND THE AREA THAT THE PLANE IS IN.
SKVLN(ISALVU,N1)=1.
PK=WAREA*WVSM/TA
PK=AMIN(1.0,PK)
SKVPC(ISALVU,N1)=1.0 - PK*WELM
SKVPC(ISALVU,N1)=SKVPC(ISALVU,N1-1)
GO TO 120
C PLANES OF THE SAME TYPE, HAVING THE SAME POSITION (FALL UNDER
THE SAME ATTACK POSITIONS) HAVE THE SAME SURVIVAL.
CONTINUE
SKVLN(ISALVU,N1)=SKVLN(ISALVU,N1-1)
SKVPC(ISALVU,N1)=SKVPC(ISALVU,N1-1)
CONTINUE
CONTINUE
IN=2*(I-1)
CALL WHITMS(4,SKVLN(I,1),N*PNS*MXNAPB,IN*1)
CALL WHITMS(4,SKVPC(I,1),N*PNS*MXNAPB,IN*2)
CONTINUE
140 RETURN
610 FORMAT (/'% THE DISTANCE TO TARGET %.13% FROM SUB %.12% IS %,
* F10.4%DA, THE AVERAGE DISTANCE TO THE TURN POINT IS%.7.2% NM%)

```

2024

1372
1373
1374
1375

020 FURMAT (O ATTEM 0010.00 MINUTES, WHEN ALAPUN O, 14,0 ARRIVES, THE
O MANGES(NM) OF AIRCRAFT ARE O/(2X01010.4)
END
QUANTA
QUANTA
QUANTA

```

1375 FUNCTION CIRCLEINTERSECT
1376 DATA PI/3.141592654
1377 CIRCLEINT COMPUTES THE INTERSECTION OF TWO CIRCLES WITH RADIUS
1378 MI AND MC. U IS THE DISTANCE BETWEEN THEM.
1379 IF (PI*U*U*U*U*U*MC*MC*U*U) GO TO 1
1380 IF (PI*MC*MC*U*U) GO TO 1
1381 IF (PI*U*U*MC*MC) GO TO 2
1382 IF (PI*U*U*MC*MC) GO TO 3
1383 IF A IS THE DISTANCE BETWEEN THE CENTER OF THE CIRCLE AND THE
1384 CORD LINE CORD IS THE LINE SEGMENT THAT CONNECTS THE
1385 POINTS OF INTERSECTION OF THE TWO CIRCLES. THEN THE TA
1386 HALF THE ANGLE THAT SWEEPS OUT THE SECTOR. IS THE
1387 APCOS(X/R) WHERE R IS THE RADIUS.
1388 THEREFORE THE INTERSECTION IS THE SUM OF THE TWO SECTORS
1389 (R*PI*THETA1) MINUS THE RECTANGLE FORMED BY THE SECTORS.
1390 COST1=0.5*(MI*MI+MC*MC*U*U)/(D*MI)
1391 THETA1=ACOS(COST1)
1392 COST2=0.5*(MC*MC+MI*MI*U*U)/(D*MC)
1393 THETA2=ACOS(COST2)
1394 CIRCLEINT=MI*THETA1+MC*THETA2-D*MI*PI*THETA1
1395 RETURN
1396 TWO CIRCLES DO NOT INTERSECT
1397 CIRCLEINT=0.
1398 RETURN
1399 MC CIRCLE ENCOMPASSES MI CIRCLE
1400 CIRCLEINT=PI*MI*MI
1401 RETURN
1402 MI CIRCLE ENCOMPASSES MC CIRCLE
1403 CIRCLEINT=PI*MC*MC
1404 RETURN
1405 END

```

QUANTA 1376
 QUANTA 1377
 QUANTA 1378
 QUANTA 1379
 QUANTA 1380
 QUANTA 1381
 QUANTA 1382
 QUANTA 1383
 QUANTA 1384
 QUANTA 1385
 QUANTA 1386
 QUANTA 1387
 QUANTA 1388
 QUANTA 1389
 QUANTA 1390
 QUANTA 1391
 QUANTA 1392
 QUANTA 1393
 QUANTA 1394
 QUANTA 1395
 QUANTA 1396
 QUANTA 1397
 QUANTA 1398
 QUANTA 1399
 QUANTA 1400
 QUANTA 1401
 QUANTA 1402
 QUANTA 1403
 QUANTA 1404
 QUANTA 1405

PL02 43

```

1406 C
1407 C
1408 C
1409 C
1410 C
1411 C
1412 C
1413 C
1414 C
1415 C
1416 C
1417 C
1418 C
1419 C
1420 C
1421 C

SUBROUTINE SET (I,KSUB)
SETS BIT TO SPECIFY PARTICULAR SUB IS OUT OF RANGE OF A PARTICULAR
BASE.
EACH WORD OF BASENO REPRESENTS A BASE
EACH BIT OF BASENO(I) REPRESENTS A SUBS RANGE FROM 1-TH BASE
EACH BIT (I. E. SUB) IS COUNTED FROM THE LEFT OF WORD BASENO(I)
COMMON/MINOFLO/BASENO(47),NBITS
TEMP=SHIFT(BASENO(I),KSUB).OR.1B
BASENO(I)=SHIFT(TEMP,NBITS-KSUB)
RETURN
END

```

QUANTA 1407
 QUANTA 1408
 QUANTA 1409
 QUANTA 1410
 QUANTA 1411
 QUANTA 1412
 QUANTA 1413
 QUANTA 1414
 QUANTA 1415
 QUANTA 1416
 QUANTA 1417
 QUANTA 1418
 QUANTA 1419
 QUANTA 1420
 QUANTA 1421

```

1421 SUBROUTINE MARKV(NMARKV,CUT,IMARKV)
1422 DIMENSION CUT(NMARKV)
1423 LEVEL = 0
1424 DATA WHERE/5,NMARKV/IN/
1425 IMARKV=0
1426 IF (NMARKV.LE.0) RETURN
1427 READ(5,5000) (CUT(I),I=1,NMARKV)
1428 DO 10 I=1,NMARKV
1429 IF (CUT(I).LT.0.0) CALL ABORT(WHERE,I,"CUT CANNOT BE LESS THAN 0")
1430 CONTINUE
1431 IMARKV=1
1432 WRITE (6,6000) NMARKV,NMARKV,(CUT(I),I=1,NMARKV)
1433 CALL PAGE(3)
1434 RETURN
1435 5000 FORMAT(7F10.5)
1436 6000 FORMAT(//14H THERE WILL BE 16.30H MARKVAL RUN(S) FOR THIS CASE./,
1437 *24H THE CUT-OFF VALUES ARE,=(2X,F10.5))
1438 END

```

```

QUANTA 1422
QUANTA 1423
QUANTA 1424
QUANTA 1425
QUANTA 1426
QUANTA 1427
QUANTA 1428
QUANTA 1429
QUANTA 1430
QUANTA 1431
QUANTA 1432
QUANTA 1433
QUANTA 1434
QUANTA 1435
QUANTA 1436
QUANTA 1437
QUANTA 1438
QUANTA 1439

```



```

1404 SUBROUTINE TOTAL (INTS, NMPS, NTYPES, ALUC, SHVLN, VAL, RELVAL, FLAMB,
1405 * PROD, VILL, MANAP, NACT, ISE, SKVPC, BONUS, BASVAL, MTYPE, MEL,
1406 * NMPS, NSUBS, NTYPES, SUB)
1407 DIMENSION ALUC (INTS, NMPS, SUB), SHVLN (NMPS, MANAP, RELVAL (NTYPES)),
1408 * SKVPC (NMPS, MANAP), VAL (INTS, NTYPES), FLAMB (INTS, NMPS, SUB),
1409 * PROD (NACT), VILL (INTS), ISE (NACT), BONUS (INTS), BASVAL (INTS),
1410 * MTYPE (NSUBS), REL (NTYPES), NMPS (NSUBS), PSOS (NTYPES),
1411 * LEVEL, ALUC, SKVPC, SHVLN, VAL, RELVAL, FLAMB, PROD, VILL, ISE,
1412 * BONUS, BASVAL, MTYPE, REL, NMPS, PSOS
1413 C
1414 C TOTAL CALCULATES THE APRAIS WHICH DEFINE THE OBJECTIVE
1415 C FUNCTION (PROD, BONUS, VILL) AND THE LAZARUSIAN MULTIPLIER
1416 C MATRIX (FLAMB)
1417 C
1418 C
1419 C
1420 C
1421 C
1422 C
1423 C
1424 C
1425 C
1426 C
1427 C
1428 C
1429 C
1430 C
1431 C
1432 C
1433 C
1434 C
1435 C
1436 C
1437 C
1438 C
1439 C
1440 C
1441 C
1442 C
1443 C
1444 C
1445 C
1446 C
1447 C
1448 C
1449 C
1450 C
1451 C
1452 C
1453 C
1454 C
1455 C
1456 C
1457 C
1458 C
1459 C
1460 C
1461 C
1462 C
1463 C
1464 C
1465 C
1466 C
1467 C
1468 C
1469 C
1470 C
1471 C
1472 C
1473 C
1474 C
1475 C
1476 C
1477 C
1478 C
1479 C
1480 C
1481 C
1482 C
1483 C
1484 C
1485 C
1486 C
1487 C
1488 C
1489 C
1490 C
1491 C
1492 C
1493 C
1494 C
1495 C
1496 C
1497 C
1498 C
1499 C
1500 C
1501 C
1502 C
1503 C
1504 C
1505 C
1506 C
1507 C
1508 C
1509 C
1510 C
1511 C
1512 C
1513 C
1514 C
1515 C
1516 C
1517 C
1518 C
1519 C
1520 C
1521 C
1522 C
1523 C
1524 C
1525 C
1526 C
1527 C
1528 C
1529 C
1530 C
1531 C
1532 C
1533 C
1534 C

```

PAGE 47

```

1534 VAL1(I)=VAL(I)+BASVAL(I)*(-BONUS(I))
1535      COMPUTE THE MULTIPLIER MATRIA (FLAMB)
1536      UU 70 J=1,NMIPS
1537      SUMC=0.
1538      SUML=0.
1539      ISUMV=0
1540      UU 60 IAC=NAL,NAL
1541      ISUMV=ISUMV+1
1542      JI=ISEL(IAC)
1543      FLAMB(I)=PMU(IAC)*ALUUS(MVPL(J)+ISUMV))*RELVAL(JT)
1544      SUMC=SUMC+FLAMB(I)
1545      FLAMB(I)=PMU(IAC)*ALUUS(MVPL(J)+ISUMV))*RELVAL(JT)
1546      SUML=SUML+FLAMB(I)
1547      CONTINUE
1548      FLAMB(I,J)=SUML
1549      FLAMB(I,J)=SUMC
1550      JFINE THE ARRAY PSB (LN OF THE SURVIV. OF THE BASE)
1551      UU 80 MI=1,MIPES
1552      PSB(MI)=ALUUS(I)-REL(MI)
1553      CONTINUE
1554      COMPUTE ADDITIONAL LAMBUA FOR BASE ATTACK
1555      J=0
1556      UU 90 ISUB=1,NSUBS
1557      MI=MI*PSB(ISUB)
1558      NMIS=NIPS(ISUB)
1559      UU 90 K=1,NMIS
1560      J=J+1
1561      FLAMB(I,J)=(-BASVAL(I)*BONUS(I))*PSB(MI)
1562      CONTINUE TO NEXT BASE
1563      RETURN
1564      END
1565
1535 QUANTA 1535
1536 QUANTA 1536
1537 QUANTA 1537
1538 QUANTA 1538
1539 QUANTA 1539
1540 QUANTA 1540
1541 QUANTA 1541
1542 QUANTA 1542
1543 QUANTA 1543
1544 QUANTA 1544
1545 QUANTA 1545
1546 QUANTA 1546
1547 QUANTA 1547
1548 QUANTA 1548
1549 QUANTA 1549
1550 QUANTA 1550
1551 QUANTA 1551
1552 QUANTA 1552
1553 QUANTA 1553
1554 QUANTA 1554
1555 QUANTA 1555
1556 QUANTA 1556
1557 QUANTA 1557
1558 QUANTA 1558
1559 QUANTA 1559
1560 QUANTA 1560
1561 QUANTA 1561
1562 QUANTA 1562
1563 QUANTA 1563
1564 QUANTA 1564
1565 QUANTA 1565
1566 QUANTA 1566

```


100	100	100	100
101	101	101	101
102	102	102	102
103	103	103	103
104	104	104	104
105	105	105	105
106	106	106	106
107	107	107	107
108	108	108	108
109	109	109	109
110	110	110	110
111	111	111	111
112	112	112	112
113	113	113	113
114	114	114	114
115	115	115	115
116	116	116	116
117	117	117	117
118	118	118	118
119	119	119	119
120	120	120	120
121	121	121	121
122	122	122	122
123	123	123	123
124	124	124	124
125	125	125	125
126	126	126	126
127	127	127	127
128	128	128	128
129	129	129	129
130	130	130	130
131	131	131	131
132	132	132	132
133	133	133	133
134	134	134	134
135	135	135	135
136	136	136	136
137	137	137	137
138	138	138	138
139	139	139	139
140	140	140	140
141	141	141	141
142	142	142	142
143	143	143	143
144	144	144	144
145	145	145	145
146	146	146	146
147	147	147	147
148	148	148	148
149	149	149	149
150	150	150	150
151	151	151	151
152	152	152	152
153	153	153	153
154	154	154	154
155	155	155	155
156	156	156	156
157	157	157	157
158	158	158	158
159	159	159	159
160	160	160	160
161	161	161	161
162	162	162	162
163	163	163	163
164	164	164	164
165	165	165	165
166	166	166	166
167	167	167	167
168	168	168	168
169	169	169	169
170	170	170	170
171	171	171	171
172	172	172	172
173	173	173	173
174	174	174	174
175	175	175	175
176	176	176	176
177	177	177	177
178	178	178	178
179	179	179	179
180	180	180	180
181	181	181	181
182	182	182	182
183	183	183	183
184	184	184	184
185	185	185	185
186	186	186	186
187	187	187	187
188	188	188	188
189	189	189	189
190	190	190	190
191	191	191	191
192	192	192	192
193	193	193	193
194	194	194	194
195	195	195	195
196	196	196	196
197	197	197	197
198	198	198	198
199	199	199	199
200	200	200	200


```

1707 CALL SEARCH(PLAMM,MIJUL,II,ISAMP,SS,MI,MMAX,MMWV,CUT,ISAMP) GUANIA 1707
1708 * ALLOC(MM) GUANIA 1708
1709 * IF (MMAX.EQ.0) GUANIA 1709
1710 IF (PLAMM.EQ.0) GO TO 30 GUANIA 1710
1711 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1711
1712 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1712
1713 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1713
1714 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1714
1715 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1715
1716 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1716
1717 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1717
1718 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1718
1719 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1719
1720 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1720
1721 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1721
1722 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1722
1723 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1723
1724 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1724
1725 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1725
1726 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1726
1727 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1727
1728 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1728
1729 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1729
1730 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1730
1731 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1731
1732 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1732
1733 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1733
1734 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1734
1735 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1735
1736 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1736
1737 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1737
1738 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1738
1739 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1739
1740 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1740
1741 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1741
1742 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1742
1743 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1743
1744 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1744
1745 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1745
1746 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1746
1747 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1747
1748 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1748
1749 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1749
1750 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1750
1751 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1751
1752 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1752
1753 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1753
1754 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1754
1755 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1755
1756 * IF (PLAMM.EQ.0) GO TO 30 GUANIA 1756

```



```

1807 PLAMTLCUM*MAXITER=DEL
1808 CONTINUE
1809
1810 GO TO 120
1811
1812 SUM IS THE SUM OF LAMCET
1813
1814 ALLOCUTEL=ALOCUTEL+DELTA
1815 UNDSUM=UNDSUM+DELTA*DELTA
1816
1817 IF THE USER HAS REQUESTED (AOCITEL*2) INFORMATION
1818 WILL BE PRINTED EACH TIME A WEAPON IS MOVED
1819
1820 CALL PRNITEL(DELTA) GO TO 130
1821
1822 CALL PAGE(7)
1823
1824 AMITE(5,500) IFX
1825
1826 AMITE(5,500) UNDSUM
1827
1828 UNDSUM=UNDSUM+UNDSUM
1829
1830 CONTINUE
1831
1832 RECALC HAS COMPLETED ITCUT ITERATIONS
1833
1834 IF (AOCITEL*2) GO TO 140
1835
1836 CALL PAGE(7)
1837
1838 AMITE(5,500) IFX
1839
1840 AMITE(5,500) UNDSUM
1841
1842 UNDSUM=UNDSUM+UNDSUM
1843
1844 CALL PRNITEL(DELTA) GO TO 150
1845
1846 IF THE WEAPONIVE POSITION HAS NOT CHANGED BY AT LEAST
1847 ONE IN THE SPECIFIED, THE ITERATIVE PROCESS WILL STOP.
1848
1849 IF (UNDSUM*DELTA*DELTA) GO TO 150
1850
1851 AMITE(5,500) UNDSUM=UNDSUM+UNDSUM
1852
1853 CALL PAGE(7)
1854
1855 GO TO 160
1856
1857 CONTINUE
1858
1859 RECALC HAS DONE THROUGH THE MAXIMUM NUMBER OF ITERATIONS
1860
1861 ITCUT=ITCUT+2 (USER SPECIFIED)
1862
1863 AMITE(5,500) EPS
1864
1865 CALL PAGE(7)
1866
1867 CONTINUE
1868
1869 RETURN
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000

```

100 100

101 102

103 104

105 106

107 108

109 110

111 112

113 114

115 116

117 118

119 120

121 122

123 124

125 126

127 128

129 130

131 132

133 134

135 136

137 138

139 140

141 142

143 144

145 146

147 148

1955	1956	1957	1958	1959	1960	1961	1962	1963	1964	1965
1966	1967	1968	1969	1970	1971	1972	1973	1974	1975	1976

IDENT SUBNOVE
ILLU WRITES MESSAGE TO OPERATIONS SUBNOVE 0407.
COMMENT WRITES MESSAGE TO OPERATIONS SUBNOVE 0407.
EXIT SUBNOVE
EG 0407
MESSAGE AT01
EG SUBNOVE
END
ENDIN/CELL.

PAGE 56

1875	IDENT	PPALTEX	1876
1876			1877
1877			1878
1878			1879
1879			1880
1880			1881
1881			1882
1882			1883
1883			1884
1884			1885
1885			1886
1886			1887
1887			1888
1888			1889
1889			1890
1890			1891
1891			1892
1892			1893
1893			1894
1894			1895
1895			1896
1896			1897
1897			1898
1898			1899

COMMENT PERMANENT FILE ALTER/EXTEND ROUTINE. 18 MAY 1977.
 TITLE PERMANENT FILE ALTER/EXTEND ROUTINE. 18 MAY 1977.
 ENABLES ALTER/EXTEND CONTROL OF THE FOLLOWING PERMANENT FILE -
 (TAP13,QUEST,IDE=DMYJJD)
 ROUTE -- THIS ROUTINE REQUIRES //PPMTEAT//.
 ENTRY ALTER,EXTEND
 SUBROUTINE ALTER ENTRY/EXIT POINT.
 EU *1517
 ALTER ALPHA
 EU RETURN.
 SUBROUTINE EXTEND ENTRY/EXIT POINT.
 EU *1517
 EXTEND ALPHA
 EU RETURN.
 FDB TAP13,QUEST,IDE=DMYJJD
 END

```

1999 SUBROUTINE MSTART (ISTART,IMARV,UBSUM,ALUC,FLAMB,PROD,VKILL,ALUCU
1900 *INTOT,IMP,INACT)
1901 COMMON/ALUC/EP,JS,ISALV,MSUB,ISSW,UBHOLD
1902 DIMENSION ALUC(INTOT,IMPNS,3), FLAMB(INTOT,IMPNS,3), PROD(INACT),
1903 * VKILL(INTOT), ALUC(IMPNS)
1904 INTEGER TODAY,NUM
1905 LEVEL 2, ALUC,FLAMB,PROD,VKILL,ALUCU
1906
1907 REMIND 13
1908 IF (ISTART.EQ.2) GO TO 1
1909 READ MSTART TAPE 13
1910 READ (13) TODAY,NUM
1911 IF (EOF(13).NEV.0) GO TO 2
1912 READ(13) IMARV,EP,JS,ISALV,MSUB,UBHOLD,UBJSUM
1913 READ(13) ((ALUC(K,L,M),K=1,INTOT),L=1,IMPNS),M=1,3)
1914 READ (13) ((FLAMB(K,L,M),K=1,INTOT),L=1,IMPNS),M=1,3)
1915 READ (13) (PROD(K),K=1,INACT)
1916 READ (13) (VKILL(K),K=1,INTOT)
1917 READ (13) (ALUC(K),K=1,IMPNS)
1918 WRITE (13) TODAY,NUM
1919 GO TO 2
1920
1921 WRITE MSTART TAPE 13
1922
1923 1 CONTINUE
1924 IF (ISSW.EQ.1) CALL ALTER
1925 CALL DATE (TODAY)
1926 CALL TIME (NUM)
1927 WRITE (13) TODAY,NUM
1928 *WRITE (13) IMARV,EP,JS,ISALV,MSUB,UBHOLD,UBJSUM
1929 *WRITE (13) ((ALUC(K,L,M),K=1,INTOT),L=1,IMPNS),M=1,3)
1930 *WRITE (13) ((FLAMB(K,L,M),K=1,INTOT),L=1,IMPNS),M=1,3)
1931 *WRITE (13) (PROD(K),K=1,INACT)
1932 *WRITE (13) (VKILL(K),K=1,INTOT)
1933 *WRITE (13) (ALUC(K),K=1,IMPNS)
1934 *WRITE (13) TODAY,NUM
1935
1936 END FILE 13
1937 REMIND 13
1938 IF (ISSW.EQ.1) CALL EXTEND
1939
1940 2 RETURN
1941
1942 3 FORMAT (5X,20H*** MSTART TAPE READ. ***5X,2A10)
1943 4 FORMAT (5X,20H*** MSTART TAPE WRITTEN. ***5X,2A10)
1944
1945 END

```

1900
 1901
 1902
 1903
 1904
 1905
 1906
 1907
 1908
 1909
 1910
 1911
 1912
 1913
 1914
 1915
 1916
 1917
 1918
 1919
 1920
 1921
 1922
 1923
 1924
 1925
 1926
 1927
 1928
 1929
 1930
 1931
 1932
 1933
 1934
 1935
 1936
 1937
 1938
 1939
 1940
 1941
 1942
 1943

PAGE 58

1943	SUBROUTINE TERM (10011+ISSW1)	UOANIA	1944
1944	IF (10011.EQ.1) WRITE (6,1)	UOANIA	1945
1945	IF (ISSW1.EQ.1) WRITE (6,2)	UOANIA	1946
1946	IF (10011.EQ.1) CALL REMARK (330000 UOANIA TERMINATED BY TIME 000)	UOANIA	1947
1947	IF (ISSW1.EQ.1) CALL REMARK (330000 UOANIA TERMINATED BY SSW1 000)	UOANIA	1948
1948	STOP	UOANIA	1949
1949		UOANIA	1950
1950	1 FORMAT (20X,330000 UOANIA TERMINATED BY TIME 000)	UOANIA	1951
1951	2 FORMAT (20X,330000 UOANIA TERMINATED BY SSW1 000)	UOANIA	1952
1952	END	UOANIA	1953

PLAGE 00

```

1994 SUBROUTINE SEARCH(FLAMB,ALUC,IT,IS,INPINS,MMI,MMAX,MMIN,MMAX,CUT)
1995 * (MMAX,ALUC,MSUB)
1996 * DIMENSION FLAMB(IT,IS,INPINS),ALUC(IT,IS,INPINS),CUT(MMAX)
1997 * ALUC(MMINS)
1998 LEVEL=FLAMB*ALUC,CUT,ALUC
1999 COMMON/IFF/LUM,MAX,MMAX,MMAXL,LUMPL
2000 LOGICAL OUTING
2001 C SEARCH FOR MIN AND MAX LAMBDA FROM WEAPON GROUP J
2002 C MAX DEFINES THE BASE WITH THE LARGEST VALUE WHILE LUM
2003 C DEFINES THE BASE WITH THE SMALLEST VALUE
2004 C MAXL AND LUMPL ARE THE ATTACK POINTS ON BASE MAX AND LUM,
2005 C RESPECTIVELY
2006 MMAX=-1.
2007 MMIN=99.
2008 MAX=1
2009 LUM=1
2010 DO 20 IF=1,3
2011 DO 20 I=1,NTGT,
2012 TEST=FLAMB(I,J)*I
2013 IF (CUT*NTGT(MSUB).GT.TEST.LE.MMAX) GO TO 10
2014 MMAX=TEST
2015 MAX=I
2016 MAXPL=I
2017 IF THE NO. OF WEAPONS ON BASE IS NOT LARGE ENOUGH,
2018 CONTINUE TO THE NEXT BASE
2019 IF (ALUC(I,J)*I*NTGT(MSUB).GT.MMIN) GO TO 20
2020 MMIN=TEST
2021 LUM=I
2022 LUMPL=I
2023 CONTINUE
2024 IF (MMAX.LE.0) GO TO 40
2025 C SINCE THIS IS A MARVAL RUN, CHECK OUT THE DUMMY BASE
2026 CUTOFF=CUT(MMAX),
2027 IF (CUTOFF.LE.MMAX) GO TO 30
2028 MMAX=CUTOFF
2029 MMAX=-1
2030 IF (ALUC(I,J)*I*0.001*MMAX.CUTOFF.GE.MMIN) GO TO 40
2031 MMIN=CUTOFF
2032 LUM=-1
2033 CONTINUE
2034 RETURN
2035 END

```

1995 GUANTIA
 1996 GUANTIA
 1997 GUANTIA
 1998 GUANTIA
 1999 GUANTIA
 2000 GUANTIA
 2001 GUANTIA
 2002 GUANTIA
 2003 GUANTIA
 2004 GUANTIA
 2005 GUANTIA
 2006 GUANTIA
 2007 GUANTIA
 2008 GUANTIA
 2009 GUANTIA
 2010 GUANTIA
 2011 GUANTIA
 2012 GUANTIA
 2013 GUANTIA
 2014 GUANTIA
 2015 GUANTIA
 2016 GUANTIA
 2017 GUANTIA
 2018 GUANTIA
 2019 GUANTIA
 2020 GUANTIA
 2021 GUANTIA
 2022 GUANTIA
 2023 GUANTIA
 2024 GUANTIA
 2025 GUANTIA
 2026 GUANTIA
 2027 GUANTIA
 2028 GUANTIA
 2029 GUANTIA
 2030 GUANTIA
 2031 GUANTIA
 2032 GUANTIA
 2033 GUANTIA
 2034 GUANTIA
 2035 GUANTIA
 2036 GUANTIA
 2037 GUANTIA

PAGE 61

037 0
038 0
039 0
040 0
041 0
042 0
043 0
044 0
045 0
046 0
047 0
048 0

LOGICAL FUNCTION OUTRNG(I,KSUB)
TESTS SUB (I.E. BIT) TO SEE IF IT IS OUT OF RANGE.
COMMON/NGFLUX/DASENU(47),NSITS
TEMP=SHIFT(DASENU(I),KSUB).AND.15
OUTNG=.FALSE.
IF (TEMP.NE.6) OUTNG=.TRUE.
RETURN
END

049 0
050 0
051 0
052 0
053 0
054 0
055 0
056 0
057 0
058 0
059 0

[illegible]

```

0099 FUNCTION FIVE1(AMIN,AMAX,EMPS,SUMVMI,SUMVLU,XALMI,ASLU,ASHMI,ASLU,MANAMP,
0100 * MANAPS,CULOFF,RELIM,BUMHI,BUMHI,BUMLU,AMEL)
0101 LEVEL = SUMVMI,SUMVLU,XALMI,ASLU,ASHMI,ASLU
0102 DATA ASLU/5.0/
0103
0104
0105 CALL MOUT(AMAX,EMAX,UIV,SUMVMI,SUMVLU,XALMI,ASLU,ASHMI,ASLU,MANAMP,
0106 * MANAPS,CULOFF,RELIM,BUMHI,BUMHI,BUMLU,AMEL)
0107 AULU=EMAX
0108 IF THE MOUT LIES TO THE RIGHT OF AMAX, RETURN AMAX
0109
0110 IF (EMAX-LEI,0.0) GO TO 50
0111
0112 AULU=0
0113 IF (UIV-EM,0.0) GO TO 20
0114
0115 DIFF=EMAX/UIV
0116 ANEW=ASLU-DIFF
0117 AULU=ANEW*1
0118 IF (AULU-LEI,0.0) GO TO 30
0119
0120 IF (ASLU-ANEW) LE,0.0) GO TO 20
0121
0122 AULU=ANEW
0123 CALL MOUT(AULU,EMAX,UIV,SUMVMI,SUMVLU,XALMI,ASLU,ASHMI,ASLU,MANAMP,
0124 * MANAPS,CULOFF,RELIM,BUMHI,BUMHI,BUMLU,AMEL)
0125
0126 IF (UIV-EM,0.0) GO TO 20
0127
0128 DIFF=EMAX/UIV
0129 ANEW=ASLU-DIFF
0130 AULU=ANEW*1
0131 IF (AULU-LEI,0.0) GO TO 40
0132
0133 AULU=ANEW
0134 IF (UIV-EM,0.0) AULU=(AMAX+XMIN)/2.0
0135
0136 CALL MOUT(AULU,EMAX,UIV,SUMVMI,SUMVLU,XALMI,ASLU,ASHMI,ASLU,MANAMP,
0137 * MANAPS,CULOFF,RELIM,BUMHI,BUMHI,BUMLU,AMEL)
0138
0139 DIFF=EMAX/UIV
0140 GO TO 10
0141
0142
0143
0144
0145
0146
0147
0148
0149
0150
0151
0152
0153
0154
0155
0156
0157
0158
0159
0160
0161
0162
0163
0164
0165
0166
0167
0168
0169
0170
0171
0172
0173
0174
0175
0176
0177
0178
0179
0180
0181
0182
0183
0184
0185
0186
0187
0188
0189
0190
0191
0192
0193
0194
0195
0196
0197
0198
0199
0200
0201
0202
0203
0204
0205
0206
0207
0208
0209
0210
0211
0212
0213
0214
0215
0216
0217
0218
0219
0220
0221
0222
0223
0224
0225
0226
0227
0228
0229
0230
0231
0232
0233
0234
0235
0236
0237
0238
0239
0240
0241
0242
0243
0244
0245
0246
0247
0248
0249
0250
0251
0252
0253
0254
0255
0256
0257
0258
0259
0260
0261
0262
0263
0264
0265
0266
0267
0268
0269
0270
0271
0272
0273
0274
0275
0276
0277
0278
0279
0280
0281
0282
0283
0284
0285
0286
0287
0288
0289
0290
0291
0292
0293
0294
0295
0296
0297
0298
0299
0300
0301
0302
0303
0304
0305
0306
0307
0308
0309
0310
0311
0312
0313
0314
0315
0316
0317
0318
0319
0320
0321
0322
0323
0324
0325
0326
0327
0328
0329
0330
0331
0332
0333
0334
0335
0336
0337
0338
0339
0340
0341
0342
0343
0344
0345
0346
0347
0348
0349
0350
0351
0352
0353
0354
0355
0356
0357
0358
0359
0360
0361
0362
0363
0364
0365
0366
0367
0368
0369
0370
0371
0372
0373
0374
0375
0376
0377
0378
0379
0380
0381
0382
0383
0384
0385
0386
0387
0388
0389
0390
0391
0392
0393
0394
0395
0396
0397
0398
0399
0400
0401
0402
0403
0404
0405
0406
0407
0408
0409
0410
0411
0412
0413
0414
0415
0416
0417
0418
0419
0420
0421
0422
0423
0424
0425
0426
0427
0428
0429
0430
0431
0432
0433
0434
0435
0436
0437
0438
0439
0440
0441
0442
0443
0444
0445
0446
0447
0448
0449
0450
0451
0452
0453
0454
0455
0456
0457
0458
0459
0460
0461
0462
0463
0464
0465
0466
0467
0468
0469
0470
0471
0472
0473
0474
0475
0476
0477
0478
0479
0480
0481
0482
0483
0484
0485
0486
0487
0488
0489
0490
0491
0492
0493
0494
0495
0496
0497
0498
0499
0500
0501
0502
0503
0504
0505
0506
0507
0508
0509
0510
0511
0512
0513
0514
0515
0516
0517
0518
0519
0520
0521
0522
0523
0524
0525
0526
0527
0528
0529
0530
0531
0532
0533
0534
0535
0536
0537
0538
0539
0540
0541
0542
0543
0544
0545
0546
0547
0548
0549
0550
0551
0552
0553
0554
0555
0556
0557
0558
0559
0560
0561
0562
0563
0564
0565
0566
0567
0568
0569
0570
0571
0572
0573
0574
0575
0576
0577
0578
0579
0580
0581
0582
0583
0584
0585
0586
0587
0588
0589
0590
0591
0592
0593
0594
0595
0596
0597
0598
0599
0600
0601
0602
0603
0604
0605
0606
0607
0608
0609
0610
0611
0612
0613
0614
0615
0616
0617
0618
0619
0620
0621
0622
0623
0624
0625
0626
0627
0628
0629
0630
0631
0632
0633
0634
0635
0636
0637
0638
0639
0640
0641
0642
0643
0644
0645
0646
0647
0648
0649
0650
0651
0652
0653
0654
0655
0656
0657
0658
0659
0660
0661
0662
0663
0664
0665
0666
0667
0668
0669
0670
0671
0672
0673
0674
0675
0676
0677
0678
0679
0680
0681
0682
0683
0684
0685
0686
0687
0688
0689
0690
0691
0692
0693
0694
0695
0696
0697
0698
0699
0700
0701
0702
0703
0704
0705
0706
0707
0708
0709
0710
0711
0712
0713
0714
0715
0716
0717
0718
0719
0720
0721
0722
0723
0724
0725
0726
0727
0728
0729
0730
0731
0732
0733
0734
0735
0736
0737
0738
0739
0740
0741
0742
0743
0744
0745
0746
0747
0748
0749
0750
0751
0752
0753
0754
0755
0756
0757
0758
0759
0760
0761
0762
0763
0764
0765
0766
0767
0768
0769
0770
0771
0772
0773
0774
0775
0776
0777
0778
0779
0780
0781
0782
0783
0784
0785
0786
0787
0788
0789
0790
0791
0792
0793
0794
0795
0796
0797
0798
0799
0800
0801
0802
0803
0804
0805
0806
0807
0808
0809
0810
0811
0812
0813
0814
0815
0816
0817
0818
0819
0820
0
```

[illegible]

[illegible]

```

210 SUBROUTINE ADJLAM(NAL,NAL,NWPN,INTYPE,MANAPB,ISURV,PROU,INACT,
211 * SURV,RELVAL,FLAMB,INTGTS,IFLAG,IBASE)
212 * DIMENSION ISEU(NACT),PROU(NACT),SURV(NWPN,MANAPB),RELVAL(INTYPE),
213 * FLAMB(INTGTS,NWPN,3)
214 LEVEL=ISEU,PROU,SURV,RELVAL,FLAMB
215 ADJLAM RECALCULATES THE MULTIPLIER MATRIX ACCORDING TO THE
216 CHANGE IN THE PRODUCT MATRIX (WHICH RESULTED FROM THE CHANGE
217 IN ALLOCATION)
218 DO 20 J=1,NWPN
219 SUM=0.
220 ISK=0
221 DO 10 I=NAL,NAL
222 ISK=ISK+1
223 JT=ISEU(I)
224 X=PROU(I)*ALOG(SURV(J,ISK))*RELVAL(JT)
225 SUM=SUM+X
226 CONTINUE
227 FLAMB(IBASE,J,IFLAG)=SUM
228 CONTINUE
229 * HEAD IN THE SURVIVABILITY OF THE AIRCRAFT AGAINST WEAPONS
230 * PLACED AT THE OTHER POINT ON BASE.
231 IF (IFLAG.EQ.2) GO TO 30
232 IN=2*(IBASE-1)+1
233 CALL HEADMS(4,SURV(1,1),NWPN,MANAPB,IN)
234 I1=2
235 GO TO 50
236
237 IN=2*(IBASE-1)+2
238 CALL HEADMS(4,SURV(1,1),NWPN,MANAPB,IN)
239 I1=3
240 * CALCULATE THE REMAINDER OF THE MULTIPLIER MATRIX
241 DO 70 J=1,NWPN
242 SUM=0.
243 ISK=0
244 DO 60 I=NAL,NAL
245 ISK=ISK+1
246 JT=ISEU(I)
247 X=PROU(I)*ALOG(SURV(J,ISK))*RELVAL(JT)
248 SUM=SUM+X
249 CONTINUE
250 FLAMB(IBASE,J,I1)=SUM
251 CONTINUE
252 RETURN
253 END

```

2211 QUANTA
 2212 QUANTA
 2213 QUANTA
 2214 QUANTA
 2215 QUANTA
 2216 QUANTA
 2217 QUANTA
 2218 QUANTA
 2219 QUANTA
 2220 QUANTA
 2221 QUANTA
 2222 QUANTA
 2223 QUANTA
 2224 QUANTA
 2225 QUANTA
 2226 QUANTA
 2227 QUANTA
 2228 QUANTA
 2229 QUANTA
 2230 QUANTA
 2231 QUANTA
 2232 QUANTA
 2233 QUANTA
 2234 QUANTA
 2235 QUANTA
 2236 QUANTA
 2237 QUANTA
 2238 QUANTA
 2239 QUANTA
 2240 QUANTA
 2241 QUANTA
 2242 QUANTA
 2243 QUANTA
 2244 QUANTA
 2245 QUANTA
 2246 QUANTA
 2247 QUANTA
 2248 QUANTA
 2249 QUANTA
 2250 QUANTA
 2251 QUANTA
 2252 QUANTA
 2253 QUANTA
 2254 QUANTA

```

254 SUBROUTINE ALINT(INTGTS,NWPNS,ALUC,ALUCU)
255 DIMENSION ALUC(INTGTS,NWPNS,J),ALUCU(NWPNS)
256 LEVEL = ALUC,ALUCU
257 C INTEGERIZE THE ALLOCATION MATRIX BY WEAPON GROUPS.
258 C
259 C
260 C
261 C
262 C
263 C
264 C
265 C
266 C
267 C
268 C
269 C
270 C
271 C
272 C
273 C
274 C
275 C
276 C
277 C
278 C
279 C
280 C
281 C
282 C
283 C
284 C
285 C
286 C
287 C
288 C
289 C
290 C
291 C
292 C
293 C
294 C
295 C
296 C
297 C
298 C
299 C
300 C
301 C
302 C
303 C
304 C
305 C
306 C
307 C
308 C
309 C
310 C
311 C
312 C
313 C
314 C
315 C
316 C
317 C
318 C
319 C
320 C
321 C
322 C
323 C
324 C
325 C
326 C
327 C
328 C
329 C
330 C
331 C
332 C
333 C
334 C
335 C
336 C
337 C
338 C
339 C
340 C
341 C
342 C
343 C
344 C
345 C
346 C
347 C
348 C
349 C
350 C
351 C
352 C
353 C
354 C
355 C
356 C
357 C
358 C
359 C
360 C
361 C
362 C
363 C
364 C
365 C
366 C
367 C
368 C
369 C
370 C
371 C
372 C
373 C
374 C
375 C
376 C
377 C
378 C
379 C
380 C
381 C
382 C
383 C
384 C
385 C
386 C
387 C
388 C
389 C
390 C
391 C
392 C
393 C
394 C
395 C
396 C
397 C
398 C
399 C
400 C
401 C
402 C
403 C
404 C
405 C
406 C
407 C
408 C
409 C
410 C
411 C
412 C
413 C
414 C
415 C
416 C
417 C
418 C
419 C
420 C
421 C
422 C
423 C
424 C
425 C
426 C
427 C
428 C
429 C
430 C
431 C
432 C
433 C
434 C
435 C
436 C
437 C
438 C
439 C
440 C
441 C
442 C
443 C
444 C
445 C
446 C
447 C
448 C
449 C
450 C
451 C
452 C
453 C
454 C
455 C
456 C
457 C
458 C
459 C
460 C
461 C
462 C
463 C
464 C
465 C
466 C
467 C
468 C
469 C
470 C
471 C
472 C
473 C
474 C
475 C
476 C
477 C
478 C
479 C
480 C
481 C
482 C
483 C
484 C
485 C
486 C
487 C
488 C
489 C
490 C
491 C
492 C
493 C
494 C
495 C
496 C
497 C
498 C
499 C
500 C
501 C
502 C
503 C
504 C
505 C
506 C
507 C
508 C
509 C
510 C
511 C
512 C
513 C
514 C
515 C
516 C
517 C
518 C
519 C
520 C
521 C
522 C
523 C
524 C
525 C
526 C
527 C
528 C
529 C
530 C
531 C
532 C
533 C
534 C
535 C
536 C
537 C
538 C
539 C
540 C
541 C
542 C
543 C
544 C
545 C
546 C
547 C
548 C
549 C
550 C
551 C
552 C
553 C
554 C
555 C
556 C
557 C
558 C
559 C
560 C
561 C
562 C
563 C
564 C
565 C
566 C
567 C
568 C
569 C
570 C
571 C
572 C
573 C
574 C
575 C
576 C
577 C
578 C
579 C
580 C
581 C
582 C
583 C
584 C
585 C
586 C
587 C
588 C
589 C
590 C
591 C
592 C
593 C
594 C
595 C
596 C
597 C
598 C
599 C
600 C
601 C
602 C
603 C
604 C
605 C
606 C
607 C
608 C
609 C
610 C
611 C
612 C
613 C
614 C
615 C
616 C
617 C
618 C
619 C
620 C
621 C
622 C
623 C
624 C
625 C
626 C
627 C
628 C
629 C
630 C
631 C
632 C
633 C
634 C
635 C
636 C
637 C
638 C
639 C
640 C
641 C
642 C
643 C
644 C
645 C
646 C
647 C
648 C
649 C
650 C
651 C
652 C
653 C
654 C
655 C
656 C
657 C
658 C
659 C
660 C
661 C
662 C
663 C
664 C
665 C
666 C
667 C
668 C
669 C
670 C
671 C
672 C
673 C
674 C
675 C
676 C
677 C
678 C
679 C
680 C
681 C
682 C
683 C
684 C
685 C
686 C
687 C
688 C
689 C
690 C
691 C
692 C
693 C
694 C
695 C
696 C
697 C
698 C
699 C
700 C
701 C
702 C
703 C
704 C
705 C
706 C
707 C
708 C
709 C
710 C
711 C
712 C
713 C
714 C
715 C
716 C
717 C
718 C
719 C
720 C
721 C
722 C
723 C
724 C
725 C
726 C
727 C
728 C
729 C
730 C
731 C
732 C
733 C
734 C
735 C
736 C
737 C
738 C
739 C
740 C
741 C
742 C
743 C
744 C
745 C
746 C
747 C
748 C
749 C
750 C
751 C
752 C
753 C
754 C
755 C
756 C
757 C
758 C
759 C
760 C
761 C
762 C
763 C
764 C
765 C
766 C
767 C
768 C
769 C
770 C
771 C
772 C
773 C
774 C
775 C
776 C
777 C
778 C
779 C
780 C
781 C
782 C
783 C
784 C
785 C
786 C
787 C
788 C
789 C
790 C
791 C
792 C
793 C
794 C
795 C
796 C
797 C
798 C
799 C
800 C
801 C
802 C
803 C
804 C
805 C
806 C
807 C
808 C
809 C
810 C
811 C
812 C
813 C
814 C
815 C
816 C
817 C
818 C
819 C
820 C
821 C
822 C
823 C
824 C
825 C
826 C
827 C
828 C
829 C
830 C
831 C
832 C
833 C
834 C
835 C
836 C
837 C
838 C
839 C
840 C
841 C
842 C
843 C
844 C
845 C
846 C
847 C
848 C
849 C
850 C
851 C
852 C
853 C
854 C
855 C
856 C
857 C
858 C
859 C
860 C
861 C
862 C
863 C
864 C
865 C
866 C
867 C
868 C
869 C
870 C
871 C
872 C
873 C
874 C
875 C
876 C
877 C
878 C
879 C
880 C
881 C
882 C
883 C
884 C
885 C
886 C
887 C
888 C
889 C
890 C
891 C
892 C
893 C
894 C
895 C
896 C
897 C
898 C
899 C
900 C
901 C
902 C
903 C
904 C
905 C
906 C
907 C
908 C
909 C
910 C
911 C
912 C
913 C
914 C
915 C
916 C
917 C
918 C
919 C
920 C
921 C
922 C
923 C
924 C
925 C
926 C
927 C
928 C
929 C
930 C
931 C
932 C
933 C
934 C
935 C
936 C
937 C
938 C
939 C
940 C
941 C
942 C
943 C
944 C
945 C
946 C
947 C
948 C
949 C
950 C
951 C
952 C
953 C
954 C
955 C
956 C
957 C
958 C
959 C
960 C
961 C
962 C
963 C
964 C
965 C
966 C
967 C
968 C
969 C
970 C
971 C
972 C
973 C
974 C
975 C
976 C
977 C
978 C
979 C
980 C
981 C
982 C
983 C
984 C
985 C
986 C
987 C
988 C
989 C
990 C
991 C
992 C
993 C
994 C
995 C
996 C
997 C
998 C
999 C
1000 C

```

2298	SUBROUTINE ALLOCINTGTS,NSUBS,NMPS,ALUC,LSUBS,NMPS)	2298	QUANTA
2299	DIMENSION ALUC(NTGTS,NMPS,3),LSUBS(INSUBS),NMPS(NSUBS)	2299	QUANTA
2300	LEVEL = ALUC(LSUBS,NMPS	2300	QUANTA
2301	ALUC PRINTS OUT ALLOCATION INFORMATION.	2301	QUANTA
2302		2302	QUANTA
2303		2303	QUANTA
2304	CALL PAGE (0)	2304	QUANTA
2305	PRINT OUT BY TARGET LOCATION.	2305	QUANTA
2306	WRITE (0,600)	2306	QUANTA
2307	CALL PAGE (1)	2307	QUANTA
2308	DO 10 I=1,NTGTS	2308	QUANTA
2309	CALL PAGE (-2)	2309	QUANTA
2310	WRITE (0,610) ,	2310	QUANTA
2311	NBW=0	2311	QUANTA
2312	DO 10 J=1,NSUBS	2312	QUANTA
2313	NMIS=NMPS(J)	2313	QUANTA
2314	DO 10 K=1,NMIS	2314	QUANTA
2315	NBW=NBW+1	2315	QUANTA
2316	ALBW=ALUC(I,NBW,1)	2316	QUANTA
2317	ALLN=ALUC(I,NBW,2)	2317	QUANTA
2318	ALPC=ALUC(I,NBW,3)	2318	QUANTA
2319	IF (ALBW+ALLN+ALPC.LT.0.0001) GO TO 10	2319	QUANTA
2320	WRITE (0,620) J,NSUBS(J),ALBW,ALLN,ALPC,N,NBW	2320	QUANTA
2321	CALL PAGE (1)	2321	QUANTA
2322	CONTINUE	2322	QUANTA
2323	PRINT OUT BY SUBMARINE LOCATION.	2323	QUANTA
2324	NBW=0	2324	QUANTA
2325	DO 20 J=1,NSUBS	2325	QUANTA
2326	CALL PAGE (-2)	2326	QUANTA
2327	WRITE (0,630) J,LSUBS(J)	2327	QUANTA
2328	NMIS=NMPS(J)	2328	QUANTA
2329	DO 20 K=1,NMIS	2329	QUANTA
2330	NBW=NBW+1	2330	QUANTA
2331	ALBW=ALUC(I,NBW,1)	2331	QUANTA
2332	ALLN=ALUC(I,NBW,2)	2332	QUANTA
2333	ALPC=ALUC(I,NBW,3)	2333	QUANTA
2334	A=ALBW+ALLN+ALPC	2334	QUANTA
2335	IF (A.LT.0.0001) GO TO 20	2335	QUANTA
2336	WRITE (0,640) A,K,1,ALBW	2336	QUANTA
2337	CALL PAGE (1)	2337	QUANTA
2338	CONTINUE	2338	QUANTA
2339	RETURN	2339	QUANTA
2340		2340	QUANTA
2341		2341	QUANTA
2342		2342	QUANTA
2343		2343	QUANTA
2344		2344	QUANTA
2345		2345	QUANTA
2346		2346	QUANTA
2347		2347	QUANTA
2348		2348	QUANTA
2349		2349	QUANTA
2350		2350	QUANTA
2351		2351	QUANTA
2352		2352	QUANTA
2353		2353	QUANTA
2354		2354	QUANTA
2355		2355	QUANTA
2356		2356	QUANTA
2357		2357	QUANTA
2358		2358	QUANTA
2359		2359	QUANTA
2360		2360	QUANTA
2361		2361	QUANTA
2362		2362	QUANTA
2363		2363	QUANTA
2364		2364	QUANTA
2365		2365	QUANTA
2366		2366	QUANTA
2367		2367	QUANTA
2368		2368	QUANTA
2369		2369	QUANTA
2370		2370	QUANTA
2371		2371	QUANTA
2372		2372	QUANTA
2373		2373	QUANTA
2374		2374	QUANTA
2375		2375	QUANTA
2376		2376	QUANTA
2377		2377	QUANTA
2378		2378	QUANTA
2379		2379	QUANTA
2380		2380	QUANTA
2381		2381	QUANTA
2382		2382	QUANTA
2383		2383	QUANTA
2384		2384	QUANTA
2385		2385	QUANTA
2386		2386	QUANTA
2387		2387	QUANTA
2388		2388	QUANTA
2389		2389	QUANTA
2390		2390	QUANTA
2391		2391	QUANTA
2392		2392	QUANTA
2393		2393	QUANTA
2394		2394	QUANTA
2395		2395	QUANTA
2396		2396	QUANTA
2397		2397	QUANTA
2398		2398	QUANTA
2399		2399	QUANTA
2400		2400	QUANTA

PAGE 69

2348
2349
2350

0300 FORMAT(10X,F10.1,0 MISSILES FROM SALVO,13,0 TO TARGET,14,
0,0 (0015.1,0 TO BASE),*)
END

QUANTA
QUANTA
QUANTA

2349
2350
2351

PAGE 70

```
2351 FUNCTION PRUFLO (ALU,X,Y,NPTS)
2352 DIMENSION X(NPTS), Y(NPTS)
2353 LEVEL 2,X,Y
2354 IF (ALU.LE.X(NPTS)).AND.ALU.GE.X(1)) GO TO 10
2355 IF (ALU.GT.X(NPTS)) GO TO 20
2356 PRUFLO=Y(1)
2357 RETURN
2358
2359 10 N=NPTS
2360 PRUFLO=AUSC(ALU,X,Y,N)
2361 PRUFLO=AMAX1(PRUFLO,Y(1))
2362 RETURN
2363
2364 20 SLOPE=(Y(NPTS)-Y(NPTS-1))/(X(NPTS)-X(NPTS-1))
2365 PRUFLO=Y(NPTS)+SLOPE*(ALU-X(NPTS))
2366 RETURN
2367 ENDF
2368
```

QUANTA
QUANTA
QUANTA
QUANTA
QUANTA
QUANTA
QUANTA
QUANTA
QUANTA
QUANTA
QUANTA
QUANTA
QUANTA
QUANTA
QUANTA
QUANTA
QUANTA
QUANTA

2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368

PAGE 71

```

369  FUNCTION DIST (ALAT,ALNG,YLAT,YLNG)
370  LEVEL 2,ALAT,ALNG,YLAT,YLNG
371  THIS FUNCTION COMPUTES THE GREAT CIRCLE DISTANCE (IN N.M.) BETWEEN
372  POINTS X AND Y WITH LATITUDES AND LONGITUDES GIVEN IN TERMS OF
373  NAUTICALS (NORTH AND WEST ARE POSITIVE).
374  DATA PI/180,PI/90,PI/360
375  A=ABS(ALNG-YLNG)
376  B=PI2-ALAT
377  C=PI2-YLAT
378  CBPC=COS(B-C)
379  X=(.5*(CBPC+CBMC)) + (.5*(COS(A)*(CBMC-CBPC)))
380  DIST=3442.2*ALUS(X)
381  RETURN
382  END

```

UUANIA
 UUANIA
 UUANIA
 UUANIA
 UUANIA
 UUANIA
 UUANIA
 UUANIA
 UUANIA
 UUANIA
 UUANIA
 UUANIA
 UUANIA
 UUANIA
 UUANIA
 UUANIA

2369
 2370
 2371
 2372
 2373
 2374
 2375
 2376
 2377
 2378
 2379
 2380
 2381
 2382
 2383

PAGE 72

```
2383 SUBROUTINE PAGE (N)
2384 C
2385 C
2386 C
2387 C
2388 C
2389 C
2390 C
2391 C
2392 C
2393 C
2394 C
2395 C
2396 C
2397 C
```

DATA LINE=NP/0.50/

IF (LINE.GT.0) GO TO 1

LINE=LINE+ABS(N)

IF (LINE.LT.0) GO TO 2

NP=NP+1

WRITE(6,N) NP

LINE=0

IF (LINE.LT.0) LINE=-N

RETURN

FORMAT(1H1,3H PAGE,14/)

END

QUANTA 2384
QUANTA 2385
QUANTA 2386
QUANTA 2387
QUANTA 2388
QUANTA 2389
QUANTA 2390
QUANTA 2391
QUANTA 2392
QUANTA 2393
QUANTA 2394
QUANTA 2395
QUANTA 2396
QUANTA 2397

PAGE 73

```

2397 C
2398 C
2399 C
2400 C
2401 C
2402 C
2403 C
2404 C
2405 C
2406 C
2407 C
2408 C
2409 C
2410 C
2411 C
2412 C
2413 C
2414 C
2415 C
2416 C
2417 C
2418 C
2419 C
2420 C
2421 C
2422 C
2423 C
2424 C
2425 C
2426 C
2427 C
2428 C
2429 C
2430 C
2431 C
2432 C
2433 C

FUNCTION SSPT (MM)
ROUTINE TO COMPUTE SOUND SPEED IN METERS/SECOND AS A FUNCTION OF
GEOMETRIC ALTITUDE IN METERS.
BASED ON DATA PRESENTED IN //U.S. STANDARD ATMOSPHERE, 1962//.
ROUTINE BY HARRY M. MURPHY, JR., 29APR/1, CORRECTED 30C172 (HMM)
DIMENSION ALT(10), TM(10), UT0Z(10)
DATA ALT/-4996.0,0.0,11019.0,20063.0,32162.0,47350.0,52425.0,61591
1.0,79994.0,90000.0/
DATA TM/320.05,288.15,216.05,216.05,220.05,270.05,252.05,18
10.05,160.05/
DATA UT0Z/-6.90522E-3,-6.4444E-3,0.0,9.9162E-4,-2.7653E-3,0.0,-1.99/
14E-3,-3.9124E-3,0.0,0.0/
Z=MM
I=1
IF (Z>9000.0) GO,50,20
IF (9000.0-Z) GO,60,30
GO 40 I=1,9
IF (ALT(I+1)-Z) 40,40,50
CONTINUE
I=10
SSPT=20.046796*SQRT(TM(I)+UT0Z(I)* (Z-ALT(I)))
RETURN
SSPT=209.44
RETURN
END
QUANTA 2398
QUANTA 2399
QUANTA 2400
QUANTA 2401
QUANTA 2402
QUANTA 2403
QUANTA 2404
QUANTA 2405
QUANTA 2406
QUANTA 2407
QUANTA 2408
QUANTA 2409
QUANTA 2410
QUANTA 2411
QUANTA 2412
QUANTA 2413
QUANTA 2414
QUANTA 2415
QUANTA 2416
QUANTA 2417
QUANTA 2418
QUANTA 2419
QUANTA 2420
QUANTA 2421
QUANTA 2422
QUANTA 2423
QUANTA 2424
QUANTA 2425
QUANTA 2426
QUANTA 2427
QUANTA 2428
QUANTA 2429
QUANTA 2430
QUANTA 2431
QUANTA 2432
QUANTA 2433

```

```

2433 C FUNCTION AUSEC (AV,X,Y,NAY)
2434 C
2435 C INTERPOLATES A TABLE OF NAY X,Y POINTS AT ENTRY POINT, AV, USING
2436 C FIRST-ORDER HERMITE, OR OSCILLATORY, INTERPOLATION.
2437 C
2438 C ALGORITHM --- EVALUATE A POLYNOMIAL WHICH PASSES THROUGH A PAIR
2439 C OF POINTS, A2,Y2, AND A3,Y3, AND WHICH HAS A GIVEN FIRST DERIVA-
2440 C TIVE AT EACH POINT. IN THIS CASE, THE DERIVATIVE AT POINT 2 IS
2441 C THE MEAN OF THE SLOPES 1-2 AND 2-3, WHILE THE DERIVATIVE AT POINT
2442 C 3 IS THE MEAN OF THE SLOPES 2-3 AND 3-4.
2443 C
2444 C ROUTINE BASED ON //HERMITE INTERPOLATION//, CAM ALGORITHM 211,
2445 C BY G.M. SCHUBERT. (COMM. ACM, OCT 1963.)
2446 C
2447 C NOTE -- THE X-ARRAY MUST MONOTONICALLY INCREASE IN VALUE.
2448 C
2449 C ROUTINE WRITTEN BY HARRY M. MURPHY, AFML/DTM, 4 JAN 1977.
2450 C
2451 C DIMENSION A(NAY), Y(NAY)
2452 C LEVEL 2, X,Y
2453 C
2454 C LOGICAL ERROR
2455 C IF (AVALT.AINAY) GO TO 1
2456 C AUSEC(Y,NAY)
2457 C GO TO 3
2458 C
2459 C START. MAKE AV AND NAY LOCAL VARIABLES.
2460 C
2461 C
2462 C CHECK TABLE LENGTH, NAY.
2463 C
2464 C ERROR=EM.L1+2
2465 C IF (ERROR) WRITE (6,*)
2466 C IF (ERROR) GO TO 3
2467 C
2468 C L=(L+M)/2
2469 C IF (X0.GT.X(1)) L=1
2470 C IF (X0.LE.X(1)) M=1
2471 C L=(L+M)/2
2472 C IF (L.LT.1) GO TO 2
2473 C
2474 C WE HAVE THE ENTRY POINT. COMPUTE X2, A3, Y2, AND
2475 C Y3 AS LOCAL VARIABLES.
2476 C
2477 C A2=X(1)
2478 C A3=X(1+1)
2479 C Y2=Y(1)
2480 C Y3=Y(1+1)
2481 C
2482 C SC3=(Y3-Y2)/(A3-A2)
2483 C IF (SC3.LE. .01) GO TO 5
2484 C SIZE=SC3

```

PAGE 75

```

1403 IF (1.0/1.0) 512=(Y(1)-Y(1-1))/(X(1)-X(1-1))
1404 534=523
1405 IF (1.0/1.0) 534=(Y(1+2)-Y(1+1))/(X(1+2)-X(1+1))
1406 C DEFINE SLOPES AT POINTS 2 AND 3.
1407 52=(512+523)/2.0
1408 53=(523+534)/2.0
1409 C PERFORM HERMITE INTERPOLATION AT X0.
1410 AUSC=((X3-X0)*52*(X2-X0)+Y2*2.0)/(X2-X3)-521+Y21+(X2-X0)*52*(X3-
1411 X0)*Y3*2.0/(X3-X2)-531+Y31)/(X3-X2)*52
1412 C RETURN.
1413 3 RETURN
1414 5 AUSC=523*(X0-X2) + Y2
1415 C RETURN
1416 4 FORMAT (1X/40H FUNCTION AUSC ERROR. TABLE LENGTH LESS THAN 2.0/1X)
1417 C
1418 END

```

GUANIA 2404
 GUANIA 2405
 GUANIA 2406
 GUANIA 2407
 GUANIA 2408
 GUANIA 2409
 GUANIA 2410
 GUANIA 2411
 GUANIA 2412
 GUANIA 2413
 GUANIA 2414
 GUANIA 2415
 GUANIA 2416
 GUANIA 2417
 GUANIA 2418
 GUANIA 2419

PAGE 1

```
1 SUBROUTINE INUCERR(SUBERR,TEMPH)  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297  
1298  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449  
1450  
1451  
1452  
1453  
1454  
1455  
1456  
1457  
1458  
1459  
1460  
1461  
1462  
1463  
1464  
1465  
1466  
1467  
1468  
1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1560  
1561  
1562  
1563  
1564  
1565  
1566  
1567  
1568  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577  
1578  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619  
1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1660  
1661  
1662  
1663  
1664  
1665  
1666  
1667  
1668  
1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1678  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727  
1728  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1750  
1751  
1752  
1753  
1754  
1755  
1756  
1757  
1758  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1780  
1781  
1782  
1783  
1784  
1785  
1786  
1787  
1788  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1870  
1871  
1872  
1873  
1874  
1875  
1876  
1877  
1878  
1879  
1880  
1881  
1882  
1883  
1884  
1885  
1886  
1887  
1888  
1889  
1890  
1891  
1892  
1893  
1894  
1895  
1896  
1897  
1898  
1899  
1900  
1901  
1902  
1903  
1904  
1905  
1906  
1907  
1908  
1909  
1910  
1911  
1912  
1913  
1914  
1915  
1916  
1917  
1918  
1919  
1920  
1921  
1922  
1923  
1924  
1925  
1926  
1927  
1928  
1929  
1930  
1931  
1932  
1933  
1934  
1935  
1936  
1937  
1938  
1939  
1940  
1941  
1942  
1943  
1944  
1945  
1946  
1947  
1948  
1949  
1950  
1951  
1952  
1953  
1954  
1955  
1956  
1957  
1958  
1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032  
2033  
2034  
2035  
2036  
2037  
2038  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050  
2051  
2052  
2053  
2054  
2055  
2056  
2057  
2058  
2059  
2060  
2061  
2062  
2063  
2064  
2065  
2066  
2067  
2068  
2069  
2070  
2071  
2072  
2073  
2074  
2075  
2076  
2077  
2078  
2079  
2080  
2081  
2082  
2083  
2084  
2085  
2086  
2087  
2088  
2089  
2090  
2091  
2092  
2093  
2094  
2095  
2096  
2097  
2098  
2099  
2100  
2101  
2102  
2103  
2104  
2105  
2106  
2107  
2108  
2109  
2110  
2111  
2112  
2113  
2114  
2115  
2116  
2117  
2118  
2119  
2120  
2121  
2122  
2123  
2124  
2125  
2126  
2127  
2128  
2129  
2130  
2131  
2132  
2133  
2134  
2135  
2136  
2137  
2138  
2139  
2140  
2141  
2142  
2143  
2144  
2145  
2146  
2147  
2148  
2149  
2150  
2151  
2152  
2153  
2154  
2155  
2156  
2157  
2158  
2159  
2160  
2161  
2162  
2163  
2164  
2165  
2166  
2167  
2168  
2169  
2170  
2171  
2172  
2173  
2174  
2175  
2176  
2177  
2178  
2179  
2180  
2181  
2182  
2183  
2184  
2185  
2186  
2187  
2188  
2189  
2190  
2191  
2192  
2193  
2194  
2195  
2196  
2197  
2198  
2199  
2200  
2201  
2202  
2203  
2204  
2205  
2206  
2207  
2208  
2209  
2210  
2211  
2212  
2213  
2214  
2215  
2216  
2217  
2218  
2219  
2220  
2221  
2222  
2223  
2224  
2225  
2226  
2227  
2228  
2229  
2230  
2231  
2232  
2233  
2234  
2235  
2236
```

```

20      DOOR=THE ATMOSP(Z),TMSIGMA,RHU,THE TAUDELTA(CA,AMU,R)
21      CALLING SEQUENCE
22      CALL ATMOSIZ,TMSIGMA,RHU,THE TAUDELTA(CA,AMU,R)
23      Z = GEOMETRIC ALTITUDE (FT)
24      TM = MOLECULAR SCALE TEMPERATURE (DEGREES RANKINE)
25      SIGMA = RATIO OF DENSITY TO THAT AT SEA LEVEL
26      RHU = DENSITY (SLUGS/CU FT)
27      THETA = RATIO OF TEMPERATURE TO THAT AT SEA LEVEL
28      DELTA = RATIO OF PRESSURE TO THAT AT SEA LEVEL
29      CA = SPEED OF SOUND (FT/SEC)
30      AMU = VISCOSITY COEFFICIENT (LB-SEC/FT**2)
31      K = 1-NORMAL
32      N = 2-ALTITUDE GREATER THAN 300000 * FT**
33      = 3-ALTITUDE NEGATIVE
34      DIMENSION A(100),B(100),C(100),D(100),E(100),F(100),G(100),H(100),I(100),J(100),K(100),L(100),M(100),N(100),O(100),P(100),Q(100),R(100),S(100),T(100),U(100),V(100),W(100),X(100),Y(100),Z(100),AA(100),AB(100),AC(100),AD(100),AE(100),AF(100),AG(100),AH(100),AI(100),AJ(100),AK(100),AL(100),AM(100),AN(100),AO(100),AP(100),AQ(100),AR(100),AS(100),AT(100),AU(100),AV(100),AW(100),AX(100),AY(100),AZ(100),BA(100),BB(100),BC(100),BD(100),BE(100),BF(100),BG(100),BH(100),BI(100),BJ(100),BK(100),BL(100),BM(100),BN(100),BO(100),BP(100),BQ(100),BR(100),BS(100),BT(100),BU(100),BV(100),BW(100),BX(100),BY(100),BZ(100),CA(100),CB(100),CC(100),CD(100),CE(100),CF(100),CG(100),CH(100),CI(100),CJ(100),CK(100),CL(100),CM(100),CN(100),CO(100),CP(100),CQ(100),CR(100),CS(100),CT(100),CU(100),CV(100),CW(100),CX(100),CY(100),CZ(100),DA(100),DB(100),DC(100),DD(100),DE(100),DF(100),DG(100),DH(100),DI(100),DJ(100),DK(100),DL(100),DM(100),DN(100),DO(100),DP(100),DQ(100),DR(100),DS(100),DT(100),DU(100),DV(100),DW(100),DX(100),DY(100),DZ(100),EA(100),EB(100),EC(100),ED(100),EE(100),EF(100),EG(100),EH(100),EI(100),EJ(100),EK(100),EL(100),EM(100),EN(100),EO(100),EP(100),EQ(100),ER(100),ES(100),ET(100),EU(100),EV(100),EW(100),EX(100),EY(100),EZ(100),FA(100),FB(100),FC(100),FD(100),FE(100),FF(100),FG(100),FH(100),FI(100),FJ(100),FK(100),FL(100),FM(100),FN(100),FO(100),FP(100),FQ(100),FR(100),FS(100),FT(100),FU(100),FV(100),FW(100),FX(100),FY(100),FZ(100),GA(100),GB(100),GC(100),GD(100),GE(100),GF(100),GG(100),GH(100),GI(100),GJ(100),GK(100),GL(100),GM(100),GN(100),GO(100),GP(100),GQ(100),GR(100),GS(100),GT(100),GU(100),GV(100),GW(100),GX(100),GY(100),GZ(100),HA(100),HB(100),HC(100),HD(100),HE(100),HF(100),HG(100),HH(100),HI(100),HJ(100),HK(100),HL(100),HM(100),HN(100),HO(100),HP(100),HQ(100),HR(100),HS(100),HT(100),HU(100),HV(100),HW(100),HX(100),HY(100),HZ(100),IA(100),IB(100),IC(100),ID(100),IE(100),IF(100),IG(100),IH(100),II(100),IJ(100),IK(100),IL(100),IM(100),IN(100),IO(100),IP(100),IQ(100),IR(100),IS(100),IT(100),IU(100),IV(100),IW(100),IX(100),IY(100),IZ(100),JA(100),JB(100),JC(100),JD(100),JE(100),JF(100),JG(100),JH(100),JI(100),JJ(100),JK(100),JL(100),JM(100),JN(100),JO(100),JP(100),JQ(100),JR(100),JS(100),JT(100),JU(100),JV(100),JW(100),JX(100),JY(100),JZ(100),KA(100),KB(100),KC(100),KD(100),KE(100),KF(100),KG(100),KH(100),KI(100),KJ(100),KK(100),KL(100),KM(100),KN(100),KO(100),KP(100),KQ(100),KR(100),KS(100),KT(100),KU(100),KV(100),KW(100),KX(100),KY(100),KZ(100),LA(100),LB(100),LC(100),LD(100),LE(100),LF(100),LG(100),LH(100),LI(100),LJ(100),LK(100),LL(100),LM(100),LN(100),LO(100),LP(100),LQ(100),LR(100),LS(100),LT(100),LU(100),LV(100),LW(100),LX(100),LY(100),LZ(100),MA(100),MB(100),MC(100),MD(100),ME(100),MF(100),MG(100),MH(100),MI(100),MJ(100),MK(100),ML(100),MN(100),MO(100),MP(100),MQ(100),MR(100),MS(100),MT(100),MU(100),MV(100),MW(100),MX(100),MY(100),MZ(100),NA(100),NB(100),NC(100),ND(100),NE(100),NF(100),NG(100),NH(100),NI(100),NJ(100),NK(100),NL(100),NM(100),NO(100),NP(100),NQ(100),NR(100),NS(100),NT(100),NU(100),NV(100),NW(100),NX(100),NY(100),NZ(100),OA(100),OB(100),OC(100),OD(100),OE(100),OF(100),OG(100),OH(100),OI(100),OJ(100),OK(100),OL(100),OM(100),ON(100),OO(100),OP(100),OQ(100),OR(100),OS(100),OT(100),OU(100),OV(100),OW(100),OX(100),OY(100),OZ(100),PA(100),PB(100),PC(100),PD(100),PE(100),PF(100),PG(100),PH(100),PI(100),PJ(100),PK(100),PL(100),PM(100),PN(100),PO(100),PP(100),PQ(100),PR(100),PS(100),PT(100),PU(100),PV(100),PW(100),PX(100),PY(100),PZ(100),QA(100),QB(100),QC(100),QD(100),QE(100),QF(100),QG(100),QH(100),QI(100),QJ(100),QK(100),QL(100),QM(100),QN(100),QO(100),QP(100),QQ(100),QR(100),QS(100),QT(100),QU(100),QV(100),QW(100),QX(100),QY(100),QZ(100),RA(100),RB(100),RC(100),RD(100),RE(100),RF(100),RG(100),RH(100),RI(100),RJ(100),RK(100),RL(100),RM(100),RN(100),RO(100),RP(100),RQ(100),RR(100),RS(100),RT(100),RU(100),RV(100),RW(100),RX(100),RY(100),RZ(100),SA(100),SB(100),SC(100),SD(100),SE(100),SF(100),SG(100),SH(100),SI(100),SJ(100),SK(100),SL(100),SM(100),SN(100),SO(100),SP(100),SQ(100),SR(100),SS(100),ST(100),SU(100),SV(100),SW(100),SX(100),SY(100),SZ(100),TA(100),TB(100),TC(100),TD(100),TE(100),TF(100),TG(100),TH(100),TI(100),TJ(100),TK(100),TL(100),TM(100),TN(100),TO(100),TP(100),TQ(100),TR(100),TS(100),TT(100),TU(100),TV(100),TW(100),TX(100),TY(100),TZ(100),UA(100),UB(100),UC(100),UD(100),UE(100),UF(100),UG(100),UH(100),UI(100),UJ(100),UK(100),UL(100),UM(100),UN(100),UO(100),UP(100),UQ(100),UR(100),US(100),UT(100),UU(100),UV(100),UW(100),UX(100),UY(100),UZ(100),VA(100),VB(100),VC(100),VD(100),VE(100),VF(100),VG(100),VH(100),VI(100),VJ(100),VK(100),VL(100),VM(100),VN(100),VO(100),VP(100),VQ(100),VR(100),VS(100),VT(100),VU(100),VV(100),VW(100),VX(100),VY(100),VZ(100),WA(100),WB(100),WC(100),WD(100),WE(100),WF(100),WG(100),WH(100),WI(100),WJ(100),WK(100),WL(100),WM(100),WN(100),WO(100),WP(100),WQ(100),WR(100),WS(100),WT(100),WU(100),WV(100),WW(100),WX(100),WY(100),WZ(100),XA(100),XB(100),XC(100),XD(100),XE(1
```

69	C	SUBROUTINE INIT (X,A,AT,Y,M)	70
70	C	SUBROUTINE INIT IS A LINEAR INTERPOLATION ROUTINE.	71
71	C	GIVEN A VALUE X, INIT RETURNS THE CORRESPONDING VALUE Y.	72
72	C	NA = DIMENSION OF X-TABLE AND Y-TABLE IN CALLING PROGRAM.	73
73	C	AT = TABLE OF X-VALUES IN CALLING PROGRAM.	74
74	C	YI = TABLE OF Y-VALUES IN CALLING PROGRAM.	75
75	C	M = RETURN VALUE Y.	76
76	C	DIMENSION AT(NA),YI(NA)	77
77	C		78
78	C	IF (X-AT(1)) LEQ 0	79
79	C	MYI(1)	80
80	C	CALL NUCLEK(BMINT, 1)	81
81	C	IF (X-AT(NA)) GEQ 0	82
82	C	MYI(NA)	83
83	C	CALL NUCLEK(BMINT, 2)	84
84	C	DO 5 I=2,NA	85
85	C	IF (X-AT(I)) LEQ 0	86
86	C	CONTINUE	87
87	C	MYI(I)	88
88	C		89
89	C	MYI(1) = (X-AT(1-1)) * (YI(1)-YI(1-1)) / (AT(1)-AT(1-1))	90
90	C		91
91	C	RETURN	92
92	C	MYI(1)	93
93	C	RETURN	94
94	C	MYI(NA)	95
95	C	RETURN	96
96	C	END	97

[illegible]

4.52 5

131
132
133
134
135
136
137
138

FOURTYE EXP(SHE)
JWEASION, SHM11(6), EXP(6)
DATA SHM11/0.0025, 5.1, 5.1, 5.500, /
DATA EXP1/5625, 537, 5642, 575, 500, 500, /
CALL IN11(SHE, SHM11, EXP1, E)
EAT=8
ECLUAY
END

EXP(SHE)
EXP(SHE)
EXP(SHE)
EXP(SHE)
EXP(SHE)
EXP(SHE)
EXP(SHE)
EXP(SHE)

132
133
134
135
136
137
138
139

PAGE 6

```
139 SUBROUTINE SPLITC  
140 COMMON/PLTCLN/MT,SP (3),SRE,00,CTH,S (3),SOUTP,SALT,SHBT,SP T  
141 SAVE=SP (1)  
142 SP (1)=S (1)  
143 S (1)=SAVE  
144 SAVE=SP (2)  
145 SP (2)=S (2)  
146 S (2)=SAVE  
147 SAVE=SP (3)  
148 SP (3)=S (3)  
149 S (3)=SAVE  
150 RETURN  
151 END  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152
```

[illegible]

017
517
617
717
817
917
017
117
217
317
417
517
617
717
817
917

```

UP1=UPRUP(SM)
CALL SW1CN
SMU=SMU/S(1)*0.001
CALL UPRUP(SM)
S(1)=S(1)*SMU-1.0
S(2)=S(1)
CALL SW1CN
SMU=UPRUP(SM)
CALL SW1CN
UPU=UPM-UP1
UPRUP=UPRUP(SM)
UPRUP=UP1
SM1=SM
GO TO 20
END

```

205	206	207	208	209	210	211	212	213	214	215	216	217
205	206	207	208	209	210	211	212	213	214	215	216	217

```

217 SUBROUTINE CMFM(S,M,KFM)
218
219 DIMENSION KMK(6),SMT(6)
220
221
222 DATA FUM KFM
223
224 DATA FKKK/1.70,1.505,1.515,1.460,1.430,1.402033/
225 DATA SMT/0.0,0.9,1.2,1.6,2.0,2.4/
226 IF (SM,GE,2.4) GO TO 20
227 CALL INTJ(S,M,6,SMT,KMK,KFM)
228 GO TO 30
229 KFM=1.3+0.48/SM
230 RETURN
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255

CALCULATES THE WAVEFORM MAJUS AT ZERO OVERPRESSURE, (WZRO). (I.E., THE MAJUS SEPARATING THE NEGATIVE AND POSITIVE PHASE PORTIONS OF THE WAVEFORM, AT THE SPECIFIED TIME (T)).

1 - TIME (SEC)
M17K - RAULUS (CM)

CUMMINS/ELLER/MT,ST(3),SMB/GU,CTP,S(3),SGUJP,SALT,SMBT,ST 1
 DATA B/O.03217,C/-1.056/C/L33597.07,BZ/8490.0/
 T = 11/ST(2)

```

      WZ=0.0
      IF (T.LI.0.0) CALL NUCEMH(4,HMFZM,1)
      IF (T.UT.0.0) WZ=(1.0-D*UT*OC)*(LZ*TBZ)
      ENLUMIN
      ENL

```

231	NUCMMU
232	NUCMMU
233	NUCMMU
234	NUCMMU
235	NUCMMU
236	NUCMMU
237	NUCMMU
238	NUCMMU
239	NUCMMU
240	NUCMMU
241	NUCMMU
242	NUCMMU
243	NUCMMU
244	NUCMMU
245	NUCMMU
246	NUCMMU
247	NUCMMU
248	NUCMMU
249	NUCMMU
250	NUCMMU
251	NUCMMU
252	NUCMMU
253	NUCMMU
254	NUCMMU
255	NUCMMU
256	NUCMMU

P 02 11

```

256      FUNCTION WPPM (I, WPPM)
257      C
258      C
259      C
260      C
261      C
262      C
263      C
264      C
265      C
266      C
267      C
268      C
269      C
270      C
271      C
272      C
273      C
274      C
275      C
276      C
277      C
278      C
279      C
280      C
281      C
282      C
283      C
284      C
285      C
286      C
287      C
288      C
289      C
290      C

```

CALCULATES THE WAVEFORM MAGNITUDE AT PEAK OVERPRESSURE (WPPM). 1.0...
 THE MAGNITUDE OF THE SHOCK FRONT, AT THE SPECIFIED TIME (T).

T = TIME (SEC)
 WPPM = MAGNITUDE

THIS MAGNITUDE IS PART OF THE AFWL INST STANDARD BY NEEDHAM ET AL.

COMMON/PLICA/PA1, SF(13), SMO, SMOUL, TP, S(13), SMOUL, SALT, SMOUL, SFT
 DATA TASP, C0, C05, TPAW1/0.317, TPAW2/0.797

I = T/ST(2)
 IF (T.LT.0.) CALL NUCLEMR(WPPM, I)

IF (T.EQ.0.) GO TO 2
 CALCULATE MAGNITUDE EXPLICITLY.

EARLY TIME FORM.
 IF (T.LT.0.20) WPPM=C4210.07507*TPW1*(1.0*(1.2307+0.123)*
 (1.0-EXP(TASP*(T-0.1)))

IF (T.LT.0.1) GO TO 2
 LATE TIME FORM.

ALI=ALUS(1)
 ALF=ALF(1,3,39102)
 WPPM=C4210.07507*TPW1*(1.0*(1.2307+0.123)*ALF
 INTERPOLATE TIME INTERPOLATION.

IF (T.LT.0.20) WPPM=(WPPM*(1-0.1)+WPPM*(0.20-1))/0.10
 WPPM=WPPM

WPPM=TPW1*SF(1)/30.40
 RETURN
 END

```

290      FUNCTION APRMT(T1,M1,PK1,UP1,MZP)
291      C
292      C
293      C
294      C
295      C
296      C
297      C
298      C
299      C
300      C
301      C
302      C
303      C
304      C
305      C
306      C
307      C
308      C
309      C
310      C
311      C
312      C
313      C
314      C
315      C
316      C
317      C
318      C
319      C
320      C
321      C
322      C
323      C
324      C
325      C
326      C
327      C
328      C
329      C
330      C
331      C
332      C
333      C
334      C
335      C
336      C
337      C
338      C
339      C
340      C

```

THIS ROUTINE IS PART OF THE AFML-1-1 STANDARD BY NEEDHAM, ET AL.
 COMMON/FLICK/MT,SP(3),SME,OU,CTP,S(3),SUOTP,SALT,SHBT,SP1
 DATA TTULU/0.0/
 T=TI/SP(2)
 K=M1*30.48/SP(1)
 PMAD=PK1*30.48/SP(1)
 UPK=UP1/SP(3)/1.45038E-5
 MPR=MPAU
 IF (T.LT.0.1) GO TO 3
 MPR=MPK*1.E-5
 M=M*1.E-5
 IF (T.EU.TTULU) GO TO 2
 MZ=RZP
 MNV=RZ-Y.7163E3*1.E-12115
 MZ=KZ*1.E-5
 MNV=MNV*1.E-5
 UPMN=2.2E4/(1+SURT(T))-0.5*UPK
 MEG=HZ-RMN
 KPLS=PK-KZ
 MNP=RPK-RMN
 ALN=OPPK-RPLS
 OLN=OPPK-RPLS
 OPMNL=ALPH*PMN*BLN
 PMLT=ABS(OPMN/UPK)
 OPMNY=OPMN*PMLT*(OPMNLN-OPMN)
 ALPHA=THNP/(OPMNY-OPPK)*MPLS/OPPK/MNEU
 BETA=-MPLS*(ALPHA*1.0/UPK)
 FNUZ=RP/RPLS
 UNUM=ALPHA*KNP+BETA
 HCMN=1.0-OPMN/(MNP/UNUM*UPK)
 CMMNL=ALUO(BL*RMN)
 CGZL=(BETA*(1.0/BL*RMN)-1.0/BL*RMN)/((FNUZ*CHMNL*KNP*(FNUZ-1.0))*(1.0/BL*RMN*OPPK*OUM))
 CGZ=CAP(CGZL)
 BUZL=CMNL*CGZ*(MNP*FNUZ)
 BUZ=EXP(BUZL)
 MNK=MPK-M
 UN=1.0-BUZ*(CGZ*(MNP*(FNUZ))
 IF (M*GT.MZ) UN=(MNP-M)/MPLS*GM*(M-MZ)/MPLS
 MNK=MK/(ALPHA*KNP*BETA)*UPK
 UPK=UPK*HM
 MPR=MPK*1.E-5

342	NJCMMU
343	NJCMU
344	NJCMMU
345	NJCMMU
346	NJCMMU
347	NJCMMU
348	NJCMMU
349	NJCMMU
350	NJCMMU
351	NJCMMU
352	NJCMMU
353	NJCMMU
354	NJCMMU
355	NJCMMU
356	NJCMMU
357	NJCMMU
358	NJCMMU

```

R=0.1,0.5
IF (T,0.02,0.05) GO TO 5
IF (T,0.0,TTULL) GO TO 4
A=5.44000E+00*(-1.00Z)-7.13500E+01*(-1.00Z)
IF (T,LT,0.02) C=7.01E-5*F0*(-0.085)
IF (T,0.02,0.02) C=AL00*(-A)/(KZP-KPK)
D=(0.0P-K1)*0.00A*(-0.00PK)
#LT=A+0.000AP(0.0K)
IF (T,LT,0.01) 0PK=#FLT
IF (T,0.0,0.1) 0PK=(#FLT*(0.95-T)+0.00K*(T-0.1))/0.85
TTULL=7
0PKMT=0PK*1.00030E-5*SF(3)
KLTUKK,
NJU

```

```

SUBROUTINE SPARKS( RADARS, **, MO, ALL, NO,
**A= JABUS, WELP, TSA, HADU,
**POINTS, AFA, AMS, IPA, ITPP, TMS)
SUBROUTINE SPARK AND ITS ASSOCIATED SUBROUTINES HAVE BEEN ADAPTED
BY CAPT DONALD WARETT, AFML/DYWM, FROM PROGRAM SPARK FURNISHED BY
AFML/DYWS. LAST PROGRAM SPARK UPDATE DENT = JULY 1977.
PARAMETERS
THE FOLLOWING INPUT PARAMETERS APPLY TO ALL SPARK CALLS-
PCASE # INDICATES OVERPRESSURE AND TSA SOLUTION DESIRED
#2 THE WEAPON FIELD IN KILOTONS.
MO = THE HEIGHT OF BURST ABOVE THE GROUND IN FEET.
ALT = THE RECEIVER ALTITUDE ABOVE THE GROUND IN FEET.
MU = THE ALTITUDE OF THE GROUND IN FEET ABOVE SEA LEVEL.
THE FOLLOWING PARAMETERS APPLY ONLY WHEN PCASE=1
INPUT-
X = THE HORIZONTAL RANGE IN FEET FROM BURST TO RECEIVER
OUTPUT-
JABUS #0 INDICATES RECEIVER INTERCEPTED BY FREE AIR SHOCK.
#1 INDICATES RECEIVER INTERCEPTED BY MACH SHOCK
WELP = PEAK OVERPRESSURE AT THE RECEIVER IN PSI.
TSA = TIME OF SHOCK ARRIVAL AT THE RECEIVER IN SECONDS.
THE FOLLOWING PARAMETERS APPLY ONLY WHEN KCASE=2
INPUT-
HADU = THE PEAK OVERPRESSURE IN PSI OF INTEREST AT RECEIVER.
OUTPUT-
POINTSE = A FLAG INDICATING THE RADIUS OF AREAS IN THE
RECEIVER'S ALTITUDE PLANE WHERE THE PEAK OVERPRESSURE
EXCEEDS HADU
#1 INDICATES CIRCLE OF RADIUS AFA.
#2 INDICATES CIRCLE OF RADIUS AMS.
#3 INDICATES CIRCLE OF RADIUS AFA SURROUNDED BY AN
ANNULUS OF MAJOR RADIUS AMS AND MINOR RADIUS ATPP.
#0 INDICATES NO AREA.
AFA = HORIZONTAL RANGE IN FEET TO POINT AT WHICH FREE AIR
PEAK OVERPRESSURE EQUALS HADU.
ATPP = HORIZONTAL RANGE IN FEET TO INTERSECTION OF TRIPLE
POINT PATH
AMS = HORIZONTAL RANGE IN FEET TO POINT AT WHICH MACH
REGION PEAK OVERPRESSURE EQUALS HADU.
TFA = TIME OF FREE AIR SHOCK ARRIVAL AT AFA IN SECONDS.
THP = TIME OF TRIPLE POINT PATH SHOCK ARRIVAL AT ATPP (SEC)
TMS = TIME OF MACH SHOCK ARRIVAL AT AMS IN SECONDS.
LOGICAL VALID
INITIALIZE BLAST AND ATMOSPHERIC PARAMETERS
MOD=means
KGRU=1

```


1. The first of these is the fact that the

2. second is the fact that the

3. third is the fact that the

4. fourth is the fact that the

5. fifth is the fact that the

6. sixth is the fact that the

7. seventh is the fact that the

8. eighth is the fact that the

9. ninth is the fact that the

10. tenth is the fact that the

11. eleventh is the fact that the

12. twelfth is the fact that the

```

405 NUMP=0
406 SP(1) = 1.2059F(1)
407 SP(2) = 1.2059F(2)
408 C INITIALIZE TRIPLE POINT PARAMETERS IF NECESSARY
409 C IF (NUMP.EQ.0) GO TO 50
410 C SET HORIZONTAL RANGE TO TRIPLE POINT PATH (TTP) ORIGIN (00.)
411 IF (SMPL(2,0).EQ.0) GO TO 40
412 SOUTP=SMPL(4,2)+3
413 SOUTP=SMPL(4,2)+3
414 GO TO 50
415 C CALL INIT(SMPL(1,4),SMPL(3,SOUTP),SOUTP)
416 GO TO 50
417 C SET SCALE FOR "STEPPEDNESS" FACTOR (CTP)
418 IF (SMPL(1,10) .GT. 10) GO TO 60
419 CT=0.004
420 GO TO 70
421 C CALL INIT(SMPL(10),SMPL(4,CTP),CTP)
422 SOUTP=SMPL(3,SOUTP)+2+SMPL(4,2)
423 IF (SOUTP.GE.204) GO TO 80
424 CALL INIT(SMPL(9),SMPL(4,CTP),CTP)
425 GO TO 90
426 C LET PPM = 1.03*0.40/SOUTP
427 P = SMPL(4)/CI
428 S(1) = C*PPM*SP(1)
429 S(2) = C*PPM*SP(2)
430 PTP = *PPM*PTM
431 RETURN
432 END

```

[illegible]

20.4 81

```
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000
```


PAGE 20

561	SUBROUTINE GETIP(ALT,ME,X,SMGZ,I,UPR,MUDU)	NUCMUD	562
562	COMMON/ELICK/ARL,SP(3),SMB,DU,CLP,S(3),SUGIP,SALT,SMBT,SPT	NUCMUD	563
563	GET HORIZONTAL RANGE TO INTERSECTION OF IMPLE POINT PATH	NUCMUD	564
564	CELT(SMBT)	NUCMUD	565
565	X=DU*((ALT/CLP/50)**2+1.)	NUCMUD	566
566	GET TIME OF SHOCK ARRIVAL AT INTERSECTION	NUCMUD	567
567	SHESMT=((ALT-MB)**2*X**2)	NUCMUD	568
568	TEFTU(SM)	NUCMUD	569
569	GET FREE AIM SHOCK AT INTERSECTION	NUCMUD	570
570	UPID=APPROX(SM)	NUCMUD	571
571	GET MACH SHOCK AT INTERSECTION	NUCMUD	572
572	SMGZ=SUPT(ALT**2*X**2)	NUCMUD	573
573	UPRETPROX(SMGZ,SINT,HINT)	NUCMUD	574
574	RETURN	NUCMUD	575
575	END	NUCMUD	576

PAGE 21

```
576 SUBROUTINE DELTA(TIME,MARU,X,T,VALU)  
577 LOGICAL VALU  
578 Z=ALTIME  
579 SP=ABS(Z)*.01  
580 VALU=.TRUE.  
581 C  
582 C  
583 C  
584 C  
585 C  
586 C  
587 C  
588 C  
589 C  
590 C  
591 C  
592 C  
593 C  
594 C  
595 C  
596 C  
597 C
```

```
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598
```

PAGE 22

```

590 SUBROUTINE GETMS(CALTIME,MAMU,X,SKIP,PI)
591 CUMMU=PI*CUK/RT*SF (3)*SM*100*CUK*5(3)*SOUTP*SALT*SM*1*SF I
592 SMZ=SM*PI
593 IF (ABS(DELPH-MAMU).LT..01)*MAMU)100 TO 30
594 SMZ=SMZ*(DELPH/MAMU)**25
595 IF (SMZ*.LT*.SKIP)100 TO 20
596 GO TO 10
597 NO SOLUTION
598 20 CALL NUCLEK(CMU*1MS *1)
599 SOLUTION FOUND
600 A=SGRT(SMZ**2-ALT**2)
601 GET TSA
602 SM=SGRT(CU*1MS**2*(100-MIN(T)**2)
603 T=FT*PI*SM
604 RETURN
605 END

```

```

599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615

```

```

600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615

```

015	FUNCTION IPRRUP(SM02,UNIT,INT)	NUCM00	015
016	COMMON/CLIC/INT,ISF(3),SMT,SO,CLT,IS(3),SOUP,SALT,SMTT,SFI	NUCM00	017
017	DIMENSION SMT(5),SMTI(13),FORM(76)	NUCM00	018
018		NUCM00	019
019		NUCM00	020
020	DATA FOR CFORM	NUCM00	021
021		NUCM00	022
022	DATA SMTI/0.0000,0.2500,0.5000,0.7500,1.0000,1.2500,1.5000,1.7500,2.0000,2.2500,2.5000,2.7500,3.0000,3.2500,3.5000,3.7500,4.0000,4.2500,4.5000,4.7500,5.0000,5.2500,5.5000,5.7500,6.0000,6.2500,6.5000,6.7500,7.0000,7.2500,7.5000,7.7500,8.0000,8.2500,8.5000,8.7500,9.0000,9.2500,9.5000,9.7500,10.0000,10.2500,10.5000,10.7500,11.0000,11.2500,11.5000,11.7500,12.0000,12.2500,12.5000,12.7500,13.0000,13.2500,13.5000,13.7500,14.0000,14.2500,14.5000,14.7500,15.0000,15.2500,15.5000,15.7500,16.0000,16.2500,16.5000,16.7500,17.0000,17.2500,17.5000,17.7500,18.0000,18.2500,18.5000,18.7500,19.0000,19.2500,19.5000,19.7500,20.0000,20.2500,20.5000,20.7500,21.0000,21.2500,21.5000,21.7500,22.0000,22.2500,22.5000,22.7500,23.0000,23.2500,23.5000,23.7500,24.0000,24.2500,24.5000,24.7500,25.0000,25.2500,25.5000,25.7500,26.0000,26.2500,26.5000,26.7500,27.0000,27.2500,27.5000,27.7500,28.0000,28.2500,28.5000,28.7500,29.0000,29.2500,29.5000,29.7500,30.0000,30.2500,30.5000,30.7500,31.0000,31.2500,31.5000,31.7500,32.0000,32.2500,32.5000,32.7500,33.0000,33.2500,33.5000,33.7500,34.0000,34.2500,34.5000,34.7500,35.0000,35.2500,35.5000,35.7500,36.0000,36.2500,36.5000,36.7500,37.0000,37.2500,37.5000,37.7500,38.0000,38.2500,38.5000,38.7500,39.0000,39.2500,39.5000,39.7500,40.0000,40.2500,40.5000,40.7500,41.0000,41.2500,41.5000,41.7500,42.0000,42.2500,42.5000,42.7500,43.0000,43.2500,43.5000,43.7500,44.0000,44.2500,44.5000,44.7500,45.0000,45.2500,45.5000,45.7500,46.0000,46.2500,46.5000,46.7500,47.0000,47.2500,47.5000,47.7500,48.0000,48.2500,48.5000,48.7500,49.0000,49.2500,49.5000,49.7500,50.0000,50.2500,50.5000,50.7500,51.0000,51.2500,51.5000,51.7500,52.0000,52.2500,52.5000,52.7500,53.0000,53.2500,53.5000,53.7500,54.0000,54.2500,54.5000,54.7500,55.0000,55.2500,55.5000,55.7500,56.0000,56.2500,56.5000,56.7500,57.0000,57.2500,57.5000,57.7500,58.0000,58.2500,58.5000,58.7500,59.0000,59.2500,59.5000,59.7500,60.0000,60.2500,60.5000,60.7500,61.0000,61.2500,61.5000,61.7500,62.0000,62.2500,62.5000,62.7500,63.0000,63.2500,63.5000,63.7500,64.0000,64.2500,64.5000,64.7500,65.0000,65.2500,65.5000,65.7500,66.0000,66.2500,66.5000,66.7500,67.0000,67.2500,67.5000,67.7500,68.0000,68.2500,68.5000,68.7500,69.0000,69.2500,69.5000,69.7500,70.0000,70.2500,70.5000,70.7500,71.0000,71.2500,71.5000,71.7500,72.0000,72.2500,72.5000,72.7500,73.0000,73.2500,73.5000,73.7500,74.0000,74.2500,74.5000,74.7500,75.0000,75.2500,75.5000,75.7500,76.0000,76.2500,76.5000,76.7500,77.0000,77.2500,77.5000,77.7500,78.0000,78.2500,78.5000,78.7500,79.0000,79.2500,79.5000,79.7500,80.0000,80.2500,80.5000,80.7500,81.0000,81.2500,81.5000,81.7500,82.0000,82.2500,82.5000,82.7500,83.0000,83.2500,83.5000,83.7500,84.0000,84.2500,84.5000,84.7500,85.0000,85.2500,85.5000,85.7500,86.0000,86.2500,86.5000,86.7500,87.0000,87.2500,87.5000,87.7500,88.0000,88.2500,88.5000,88.7500,89.0000,89.2500,89.5000,89.7500,90.0000,90.2500,90.5000,90.7500,91.0000,91.2500,91.5000,91.7500,92.0000,92.2500,92.5000,92.7500,93.0000,93.2500,93.5000,93.7500,94.0000,94.2500,94.5000,94.7500,95.0000,95.2500,95.5000,95.7500,96.0000,96.2500,96.5000,96.7500,97.0000,97.2500,97.5000,97.7500,98.0000,98.2500,98.5000,98.7500,99.0000,99.2500,99.5000,99.7500,100.0000,100.2500,100.5000,100.7500,101.0000,101.2500,101.5000,101.7500,102.0000,102.2500,102.5000,102.7500,103.0000,103.2500,103.5000,103.7500,104.0000,104.2500,104.5000,104.7500,105.0000,105.2500,105.5000,105.7500,106.0000,106.2500,106.5000,106.7500,107.0000,107.2500,107.5000,107.7500,108.0000,108.2500,108.5000,108.7500,109.0000,109.2500,109.5000,109.7500,110.0000,110.2500,110.5000,110.7500,111.0000,111.2500,111.5000,111.7500,112.0000,112.2500,112.5000,112.7500,113.0000,113.2500,113.5000,113.7500,114.0000,114.2500,114.5000,114.7500,115.0000,115.2500,115.5000,115.7500,116.0000,116.2500,1		

PAGE 24

```

641 C
642 C
643 C
644 C
645 C
646 C
647 C
648 C
649 C
650 C
651 C
652 C
653 C
654 C
655 C
656 C
657 C
658 C
659 C
660 C
661 C

FUNCTION WFTP(WFT)
  R = MAJUS (FEET)
  WFTP = TIME AT PEAK (SEC)
  COMMON/FLICK/WAL,SP(3),SMB,GG,CLP,S(3),SOUTP,SALT,SMOT,SP1
  CONVERT FROM FEET TO METERS
  R=WFT*.3048/SP(1)
  IF (WALT.57.0)GO TO 10
  A=SQRT(R)
  Y=SQRT(X)
  WFTP=.0029551*R-.06348*Y-.50446/X+.28741/(R**(.2/.3))-263.5/(X**Y)
  GO TO 20
10 WFTP=X*(1.7455/E-4+R*(-2.0)16E-5+R*(1.3371E-6+R*(-3.6325E-8+R*
  *(4.5554E-10+R*(-2.0)1633E-12))))
20 CONTINUE
  WFTP=WFTP*SP(2)
  IF (WFTP.LT.0.)CALL NUCLERM(OM*WFTP ,1)
  RETURN
END

```

NUCMUU 642
 NUCMUU 643
 NUCMUU 644
 NUCMUU 645
 NUCMUU 646
 NUCMUU 647
 NUCMUU 648
 NUCMUU 649
 NUCMUU 650
 NUCMUU 651
 NUCMUU 652
 NUCMUU 653
 NUCMUU 654
 NUCMUU 655
 NUCMUU 656
 NUCMUU 657
 NUCMUU 658
 NUCMUU 659
 NUCMUU 660
 NUCMUU 661


```

686 C SUBROUTINE TPI(M,M1,U1)
687 C GIVEN M, M1, U1, SUBROUTINE TPI RETURNS M1 AND U1.
688 C M IS THE HEIGHT OF THE CENTER OF THE CIRCLE
689 C M1 IS THE HEIGHT OF THE CENTER OF THE CIRCLE
690 C M1 IS THE HEIGHT OF THE CENTER OF THE CIRCLE
691 C M1 IS THE HEIGHT OF THE CENTER OF THE CIRCLE
692 C M1 IS THE HEIGHT OF THE CENTER OF THE CIRCLE
693 C M1 IS THE HEIGHT OF THE CENTER OF THE CIRCLE
694 C M1 IS THE HEIGHT OF THE CENTER OF THE CIRCLE
695 C M1 IS THE HEIGHT OF THE CENTER OF THE CIRCLE
696 C M1 IS THE HEIGHT OF THE CENTER OF THE CIRCLE
697 C M1 IS THE HEIGHT OF THE CENTER OF THE CIRCLE
698 C M1 IS THE HEIGHT OF THE CENTER OF THE CIRCLE
699 C M1 IS THE HEIGHT OF THE CENTER OF THE CIRCLE
700 C M1 IS THE HEIGHT OF THE CENTER OF THE CIRCLE
701 C M1 IS THE HEIGHT OF THE CENTER OF THE CIRCLE
702 C M1 IS THE HEIGHT OF THE CENTER OF THE CIRCLE
703 C M1 IS THE HEIGHT OF THE CENTER OF THE CIRCLE
704 C M1 IS THE HEIGHT OF THE CENTER OF THE CIRCLE
705 C M1 IS THE HEIGHT OF THE CENTER OF THE CIRCLE
706 C M1 IS THE HEIGHT OF THE CENTER OF THE CIRCLE
707 C M1 IS THE HEIGHT OF THE CENTER OF THE CIRCLE
708 C M1 IS THE HEIGHT OF THE CENTER OF THE CIRCLE
709 C M1 IS THE HEIGHT OF THE CENTER OF THE CIRCLE
710 C M1 IS THE HEIGHT OF THE CENTER OF THE CIRCLE
711 C M1 IS THE HEIGHT OF THE CENTER OF THE CIRCLE
712 C M1 IS THE HEIGHT OF THE CENTER OF THE CIRCLE
713 C M1 IS THE HEIGHT OF THE CENTER OF THE CIRCLE
714 C M1 IS THE HEIGHT OF THE CENTER OF THE CIRCLE
715 C M1 IS THE HEIGHT OF THE CENTER OF THE CIRCLE

```

```

NUCMU 687
NUCMU 688
NUCMU 689
NUCMU 690
NUCMU 691
NUCMU 692
NUCMU 693
NUCMU 694
NUCMU 695
NUCMU 696
NUCMU 697
NUCMU 698
NUCMU 699
NUCMU 700
NUCMU 701
NUCMU 702
NUCMU 703
NUCMU 704
NUCMU 705
NUCMU 706
NUCMU 707
NUCMU 708
NUCMU 709
NUCMU 710
NUCMU 711
NUCMU 712
NUCMU 713
NUCMU 714
NUCMU 715

```


PAGE 20

```

705 C THE RECEIVER PANEL AT THE RECEIVER LOCATION SPECIFIED BY NUCMUU
706 C NR AND ACALIT IS DESIRED. NUCMUU
707 C = THE HORIZONTAL RANGE IN FEET TO THE POINT AT WHICH THERMAL NUCMUU
708 C RADIATION INCIDENT ON THE RECEIVER PANEL EQUALS CAL. NUCMUU
709 C VALU = A LOGICAL VARIABLE WHICH PLAYS HUR=0.0 WHEN FALSE. NUCMUU
710 C NUCMUU
711 C NUCMUU
712 C NUCMUU
713 C NUCMUU
714 C NUCMUU
715 C NUCMUU
716 C NUCMUU
717 C NUCMUU
718 C NUCMUU
719 C NUCMUU
720 C NUCMUU
721 C NUCMUU
722 C NUCMUU
723 C NUCMUU
724 C NUCMUU
725 C NUCMUU
726 C NUCMUU
727 C NUCMUU
728 C NUCMUU
729 C NUCMUU
730 C NUCMUU
731 C NUCMUU
732 C NUCMUU
733 C NUCMUU
734 C NUCMUU
735 C NUCMUU
736 C NUCMUU
737 C NUCMUU
738 C NUCMUU
739 C NUCMUU
740 C NUCMUU
741 C NUCMUU
742 C NUCMUU
743 C NUCMUU
744 C NUCMUU
745 C NUCMUU
746 C NUCMUU
747 C NUCMUU
748 C NUCMUU
749 C NUCMUU
750 C NUCMUU
751 C NUCMUU
752 C NUCMUU
753 C NUCMUU
754 C NUCMUU
755 C NUCMUU
756 C NUCMUU
757 C NUCMUU
758 C NUCMUU
759 C NUCMUU
760 C NUCMUU
761 C NUCMUU
762 C NUCMUU
763 C NUCMUU
764 C NUCMUU
765 C NUCMUU
766 C NUCMUU
767 C NUCMUU
768 C NUCMUU
769 C NUCMUU
770 C NUCMUU
771 C NUCMUU
772 C NUCMUU
773 C NUCMUU
774 C NUCMUU
775 C NUCMUU
776 C NUCMUU
777 C NUCMUU
778 C NUCMUU
779 C NUCMUU
780 C NUCMUU
781 C NUCMUU
782 C NUCMUU
783 C NUCMUU
784 C NUCMUU
785 C NUCMUU
786 C NUCMUU
787 C NUCMUU
788 C NUCMUU
789 C NUCMUU
790 C NUCMUU
791 C NUCMUU
792 C NUCMUU
793 C NUCMUU
794 C NUCMUU
795 C NUCMUU
796 C NUCMUU
797 C NUCMUU
798 C NUCMUU
799 C NUCMUU
800 C NUCMUU
801 C NUCMUU
802 C NUCMUU
803 C NUCMUU
804 C NUCMUU
805 C NUCMUU
806 C NUCMUU
807 C NUCMUU
808 C NUCMUU
809 C NUCMUU
810 C NUCMUU
811 C NUCMUU
812 C NUCMUU
813 C NUCMUU
814 C NUCMUU
815 C NUCMUU

```


Page 30

```

865 1* HUB=SQRT(KRES)*KREST-D*IFALT*D*IFALT)
866 1D BVANG=ATAN2((HUB*D*IFALT)
867 IF (BVANG.LE.U.) BVANG=B*VANG*D*PI
868 XFB=HUB*VCUS(BHANG)
869 YFB=HUB*VSIN(BHANG)
870 ZFB=D*IFALT
871 PHANG=BHANG+HANGLE*HRADEG
872 PVANG=B*VANG+VANGLE*HRADEG
873 XVA=VSIN(PVANG)*VCUS(PHANG)
874 YVA=VSIN(PVANG)*VSIN(PHANG)
875 ZVA=VCUS(PVANG)
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915

```

PAGE 31

```

915 C 25 CALL AMEAT(IMEAT,SNIP,POFLAG)
916 IF (PFLAG.EQ.1) GO TO 45
917 UU = 39.0577 * U
918 URATE = UU / UELTIM
919 UC = UU * UC
920 UV = UU / VF
921 UU = UU * UV * AA
922 UR = UR * UV * AB
923 IF (TIME.LT. IMAX) GO TO 20
924 IF (URATE .GE. 0.027) GO TO 20
925 CALL COMM(CAL,MMUR,REST,UC,UU,UR)
926
927 C IF LEVEL SOLUTION ONLY DESIRED, RETURN
928
929 C 30 IF (CAL.LT.0.0) RETURN
930
931 C SEE IF A NEAR CAL. IF NOT, CHANGE TARGET POSITION AND ITERATE
932
933 C IF (ABS(UC-CAL).LT.0.01) CAL=RETURN
934 IF (JUU.GT.10) GO TO 40
935 REST=GUESN(JUU,CAL,UC,REST)
936 IF (REST.GT.MFB).AND.(REST.GT.ABS(DIFALT)) GO TO 14
937 IF (PFLAG.EQ.1) GO TO 35
938 PHLAG=1.
939 REST=MAX1(MFB*.01,(ABS(DIFALT)).01))
940 GO TO 14
941 C ABNORMAL RETURNS
942 C
943 C 35 VALUE=FALSE.
944 MUR=0.0
945 RETURN
946
947 C LIVERGENCE ABORT
948 40 CALL NUCERR(EMSNARE *1)
949 C RECEIVED GUT IN THE FIREBALL
950 45 UC=10000.
951 UR=10000.
952 UR=0.0
953 IF (MMTFLG) WRITE(6,6000)
954 6000 FORMAT(" RECEIVER ENTERED FIREBALL")
955 RETURN
956 END

```

916
 917
 918
 919
 920
 921
 922
 923
 924
 925
 926
 927
 928
 929
 930
 931
 932
 933
 934
 935
 936
 937
 938
 939
 940
 941
 942
 943
 944
 945
 946
 947
 948
 949
 950
 951
 952
 953
 954
 955
 956
 957

PAGE 34

1057
1058

RETURN
END

NUCMOD 1058
NUCMOD 1059

```

1059 FUNCTION GUESN(JJJ,WANT,UC,REST)
1060 DIMENSION TRY(10),CALC(10)
1061 CALC(JJJ)=UC
1062 TRY(JJJ)=REST
1063 IF (JJJ.EQ.1) GO TO 4
1064 A1=0.
1065 A2=0.
1066 A3=0.
1067 A4=0.
1068 ZN=0.
1069 DO 2 I=1,JJJ
1070 Z=1./((WANT-CALC(I))*2
1071 A=ALOG(TRY(I))
1072 C=ALOG(CALC(I))
1073 ZN=ZN+Z
1074 A1=A1+A*Z
1075 AC=A2+A*Z
1076 A3=A3+A*Z
1077 A4=A4+C*Z
1078 YVAL=(AC*A3-A1*A4)/(A2*AC-A1*ZN)
1079 SLOPE=(A3-YVAL)/A1
1080 A=EXP(YVAL)
1081 B=SLOPE
1082 GUESN=(WANT/A)**(1./B)
1083 RETURN
1084 B=-Z.
1085 A=CALC(1)+TRY(1)*2
1086 GO TO 3
1087 END

```

```

NUCMU 1060
NUCMU 1061
NUCMU 1062
NUCMU 1063
NUCMU 1064
NUCMU 1065
NUCMU 1066
NUCMU 1067
NUCMU 1068
NUCMU 1069
NUCMU 1070
NUCMU 1071
NUCMU 1072
NUCMU 1073
NUCMU 1074
NUCMU 1075
NUCMU 1076
NUCMU 1077
NUCMU 1078
NUCMU 1079
NUCMU 1080
NUCMU 1081
NUCMU 1082
NUCMU 1083
NUCMU 1084
NUCMU 1085
NUCMU 1086
NUCMU 1087
NUCMU 1088

```

P. 62 36

```

1088 SUBROUTINE UMM(BURST,WM,HUMF,MEST,UC,UU,UR)
1089 IF (W.GT.500.0) GO TO 20
1090 SHUB=BURST/(W*(1.73.1))
1091 IF (SHUB.GT.369.0) GO TO 20
1092 IF (SHUB.GE.177.0.AND.SHUB.LE.369.0) THSU=6.21*1.0E+06*(W/BURST**3)
1093 I=(2.0E-08)
1094 IF (SHUB.LI.177.0) THSU=1.0
1095 IF (SHUB.GT.369.0) THSU=0.0
1096 TSU=1.0-THSU
1097 IF (SHUB.LE.183.0) FMW=(3.41E-3*SHUB)*1.0
1098 IF (SHUB.GT.183.0) FMW=1.63
1099 SINT=(HUMF/MEST)
1100 THETA1=(2.0/3.0)*ASIN(SINT)
1101 THETA=COS(THETA1)
1102 FACTOR=((TSU)*(THSU*FMW*THETA))
1103 UC=UC*FACTOR
1104 UR=UR*FACTOR
1105 UU=UU*FACTOR
1106 GO TO 21
1107 UC=UC
1108 UR=UR
1109 UU=UU
1110 CONTINUE
1111 RETURN
1112 END
1089 NUCMOD
1090 NUCMOD
1091 NUCMOD
1092 NUCMOD
1093 NUCMOD
1094 NUCMOD
1095 NUCMOD
1096 NUCMOD
1097 NUCMOD
1098 NUCMOD
1099 NUCMOD
1100 NUCMOD
1101 NUCMOD
1102 NUCMOD
1103 NUCMOD
1104 NUCMOD
1105 NUCMOD
1106 NUCMOD
1107 NUCMOD
1108 NUCMOD
1109 NUCMOD
1110 NUCMOD
1111 NUCMOD
1112 NUCMOD
1113 NUCMOD

```

```

1113 C SUBROUTINE AINT (ZF1,ZF2,WAVEL,V,IMUN,PAVE,WIN,SIGAVE,AUACAV,WCUZ)
1114 C
1115 C COMPUTES REQUIRED ALTITUDE INTEGRALS
1116 C ZF1 AND ZF2 ARE SOURCE AND RECEIVER ALTITUDES IN FEET.
1117 C WAVEL IS THE CENTER WAVE LENGTH IN MICRONS.
1118 C V IS THE SURFACE VISUAL RANGE IN MILES.
1119 C IMUN IS ATMOSPHERE CODE - 1 = WET ATMOSPHERE
1120 C 2 = DRY ATMOSPHERE
1121 C PAVE IS THE RETURNED AVERAGE PRESSURE IN MM-HG.
1122 C WIN IS THE RETURNED WATER INTEGRAL FOR EIGHT DIFFERENT
1123 C VALUES OF EXPONENTS OF PRESSURE.
1124 C
1125 C IF ZF1=ZF2, VALUE OF INTEGRAND IS RETURNED IN WIN.
1126 C SIGAVE IS THE RETURNED AVERAGE SCATTERING COEFFICIENT IN KM-1.
1127 C AUACAV IS THE RETURNED AVERAGE ATMOSPHERIC OZONE ABSORPTION
1128 C COEFFICIENT IN KM-1.
1129 C WCUZ IS THE RETURNED CARBON DIOXIDE CONCENTRATION IN ATMOS-CM/KM.
1130 C
1131 C DIMENSION WIN(8), EX(8), WAVEL(14), SIGAVE(14), AUACAV(4), AUA(4)
1132 C
1133 C DATA EX/0.0,0.43,0.52,0.54,0.60,0.61,0.62,0.64/
1134 C
1135 C IF (ABS(ZF1-ZF2).GT.33.0) GO TO 3
1136 C CALL ATMUS(ZF1,TM,XZX,MHU,THETA,PAVE,CA,AMU,K)
1137 C PAVE=760.0*PAVE
1138 C CALL WATMHU (ZF1,IMUN,RHUH)
1139 C
1140 C DO 1 I=1,8
1141 C WIN(I)=PAVE*EX(I)*MHU
1142 C
1143 C DO 2 J=AND=1,14
1144 C ALAM=WAVEL (IDAND)
1145 C CALL SCAT (ZF1,ALAM,V,SIG)
1146 C SIGAVE(IDAND)=SIG
1147 C CALL OZAL (ZF1,AUACAV)
1148 C CALL WCUZP (ZF1,WCUZ*U)
1149 C WCUZ=-WCUZ
1150 C RETURN
1151 C
1152 C UZ=ABS(ZF1-ZF2)/20.0
1153 C Z=AMIN1(ZF1,ZF2)-UZ/Z.0
1154 C PAVE=0.0
1155 C DO 4 I=1,8
1156 C WIN(I)=0.0
1157 C
1158 C DO 5 IDAND=1,14
1159 C SIGAVE(IDAND)=0.0
1160 C
1161 C DO 6 J=1,14
1162 C AUACAV(J)=0.0
1163 C
1164 C DO 10 J=1,20
1165 C Z=Z*U/
1166 C CALL ATMUS(Z,TM,XZX,MHU,THETA,P,CA,AMU,K)
1167 C P=760.0*P
1168 C PAVE=PAVE*P
1169 C
1170 C
1171 C
1172 C
1173 C
1174 C
1175 C
1176 C
1177 C
1178 C
1179 C
1180 C
1181 C
1182 C
1183 C
1184 C
1185 C
1186 C
1187 C
1188 C
1189 C
1190 C
1191 C
1192 C
1193 C
1194 C
1195 C
1196 C
1197 C
1198 C
1199 C
1200 C
1201 C
1202 C
1203 C
1204 C
1205 C
1206 C
1207 C
1208 C
1209 C
1210 C
1211 C
1212 C
1213 C
1214 C
1215 C
1216 C
1217 C
1218 C
1219 C
1220 C
1221 C
1222 C
1223 C
1224 C
1225 C
1226 C
1227 C
1228 C
1229 C
1230 C
1231 C
1232 C
1233 C
1234 C
1235 C
1236 C
1237 C
1238 C
1239 C
1240 C
1241 C
1242 C
1243 C
1244 C
1245 C
1246 C
1247 C
1248 C
1249 C
1250 C
1251 C
1252 C
1253 C
1254 C
1255 C
1256 C
1257 C
1258 C
1259 C
1260 C
1261 C
1262 C
1263 C
1264 C
1265 C
1266 C
1267 C
1268 C
1269 C
1270 C
1271 C
1272 C
1273 C
1274 C
1275 C
1276 C
1277 C
1278 C
1279 C
1280 C
1281 C
1282 C
1283 C
1284 C
1285 C
1286 C
1287 C
1288 C
1289 C
1290 C
1291 C
1292 C
1293 C
1294 C
1295 C
1296 C
1297 C
1298 C
1299 C
1300 C
1301 C
1302 C
1303 C
1304 C
1305 C
1306 C
1307 C
1308 C
1309 C
1310 C
1311 C
1312 C
1313 C
1314 C
1315 C
1316 C
1317 C
1318 C
1319 C
1320 C
1321 C
1322 C
1323 C
1324 C
1325 C
1326 C
1327 C
1328 C
1329 C
1330 C
1331 C
1332 C
1333 C
1334 C
1335 C
1336 C
1337 C
1338 C
1339 C
1340 C
1341 C
1342 C
1343 C
1344 C
1345 C
1346 C
1347 C
1348 C
1349 C
1350 C
1351 C
1352 C
1353 C
1354 C
1355 C
1356 C
1357 C
1358 C
1359 C
1360 C
1361 C
1362 C
1363 C
1364 C
1365 C
1366 C
1367 C
1368 C
1369 C
1370 C
1371 C
1372 C
1373 C
1374 C
1375 C
1376 C
1377 C
1378 C
1379 C
1380 C
1381 C
1382 C
1383 C
1384 C
1385 C
1386 C
1387 C
1388 C
1389 C
1390 C
1391 C
1392 C
1393 C
1394 C
1395 C
1396 C
1397 C
1398 C
1399 C
1400 C
1401 C
1402 C
1403 C
1404 C
1405 C
1406 C
1407 C
1408 C
1409 C
1410 C
1411 C
1412 C
1413 C
1414 C
1415 C
1416 C
1417 C
1418 C
1419 C
1420 C
1421 C
1422 C
1423 C
1424 C
1425 C
1426 C
1427 C
1428 C
1429 C
1430 C
1431 C
1432 C
1433 C
1434 C
1435 C
1436 C
1437 C
1438 C
1439 C
1440 C
1441 C
1442 C
1443 C
1444 C
1445 C
1446 C
1447 C
1448 C
1449 C
1450 C
1451 C
1452 C
1453 C
1454 C
1455 C
1456 C
1457 C
1458 C
1459 C
1460 C
1461 C
1462 C
1463 C
1464 C
1465 C
1466 C
1467 C
1468 C
1469 C
1470 C
1471 C
1472 C
1473 C
1474 C
1475 C
1476 C
1477 C
1478 C
1479 C
1480 C
1481 C
1482 C
1483 C
1484 C
1485 C
1486 C
1487 C
1488 C
1489 C
1490 C
1491 C
1492 C
1493 C
1494 C
1495 C
1496 C
1497 C
1498 C
1499 C
1500 C
1501 C
1502 C
1503 C
1504 C
1505 C
1506 C
1507 C
1508 C
1509 C
1510 C
1511 C
1512 C
1513 C
1514 C
1515 C
1516 C
1517 C
1518 C
1519 C
1520 C
1521 C
1522 C
1523 C
1524 C
1525 C
1526 C
1527 C
1528 C
1529 C
1530 C
1531 C
1532 C
1533 C
1534 C
1535 C
1536 C
1537 C
1538 C
1539 C
1540 C
1541 C
1542 C
1543 C
1544 C
1545 C
1546 C
1547 C
1548 C
1549 C
1550 C
1551 C
1552 C
1553 C
1554 C
1555 C
1556 C
1557 C
1558 C
1559 C
1560 C
1561 C
1562 C
1563 C
1564 C
1565 C
1566 C
1567 C
1568 C
1569 C
1570 C
1571 C
1572 C
1573 C
1574 C
1575 C
1576 C
1577 C
1578 C
1579 C
1580 C
1581 C
1582 C
1583 C
1584 C
1585 C
1586 C
1587 C
1588 C
1589 C
1590 C
1591 C
1592 C
1593 C
1594 C
1595 C
1596 C
1597 C
1598 C
1599 C
1600 C
1601 C
1602 C
1603 C
1604 C
1605 C
1606 C
1607 C
1608 C
1609 C
1610 C
1611 C
1612 C
1613 C
1614 C
1615 C
1616 C
1617 C
1618 C
1619 C
1620 C
1621 C
1622 C
1623 C
1624 C
1625 C
1626 C
1627 C
1628 C
1629 C
1630 C
1631 C
1632 C
1633 C
1634 C
1635 C
1636 C
1637 C
1638 C
1639 C
1640 C
1641 C
1642 C
1643 C
1644 C
1645 C
1646 C
1647 C
1648 C
1649 C
1650 C
1651 C
1652 C
1653 C
1654 C
1655 C
1656 C
1657 C
1658 C
1659 C
1660 C
1661 C
1662 C
1663 C
1664 C
1665 C
1666 C
1667 C
1668 C
1669 C
1670 C
1671 C
1672 C
1673 C
1674 C
1675 C
1676 C
1677 C
1678 C
1679 C
1680 C
1681 C
1682 C
1683 C
1684 C
1685 C
1686 C
1687 C
1688 C
1689 C
1690 C
1691 C
1692 C
1693 C
1694 C
1695 C
1696 C
1697 C
1698 C
1699 C
1700 C
1701 C
1702 C
1703 C
1704 C
1705 C
1706 C
1707 C
1708 C
1709 C
1710 C
1711 C
1712 C
1713 C
1714 C
1715 C
1716 C
1717 C
1718 C
1719 C
1720 C
1721 C
1722 C
1723 C
1724 C
1725 C
1726 C
1727 C
1728 C
1729 C
1730 C
1731 C
1732 C
1733 C
1734 C
1735 C
1736 C
1737 C
1738 C
1739 C
1740 C
1741 C
1742 C
1743 C
1744 C
1745 C
1746 C
1747 C
1748 C
1749 C
1750 C
1751 C
1752 C
1753 C
1754 C
1755 C
1756 C
1757 C
1758 C
1759 C
1760 C
1761 C
1762 C
1763 C
1764 C
1765 C
1766 C
1767 C
1768 C
1769 C
1770 C
1771 C
1772 C
1773 C
1774 C
1775 C
1776 C
1777 C
1778 C
1779 C
1780 C
1781 C
1782 C
1783 C
1784 C
1785 C
1786 C
1787 C
1788 C
1789 C
1790 C
1791 C
1792 C
1793 C
1794 C
1795 C
1796 C
1797 C
1798 C
1799 C
1800 C
1801 C
1802 C
1803 C
1804 C
1805 C
1806 C
1807 C
1808 C
1809 C
1810 C
1811 C
1812 C
1813 C
1814 C
1815 C
1816 C
1817 C
1818 C
1819 C
1820 C
1821 C
1822 C
1823 C
1824 C
1825 C
1826 C
1827 C
1828 C
1829 C
1830 C
1831 C
1832 C
1833 C
1834 C
1835 C
1836 C
1837 C
1838 C
1839 C
1840 C
1841 C
1842 C
1843 C
1844 C
1845 C
1846 C
1847 C
1848 C
1849 C
1850 C
1851 C
1852 C
1853 C
1854 C
1855 C
1856 C
1857 C
1858 C
1859 C
1860 C
1861 C
1862 C
1863 C
1864 C
1865 C
1866 C
1867 C
1868 C
1869 C
1870 C
1871 C
1872 C
1873 C
1874 C
1875 C
1876 C
1877 C
1878 C
1879 C
1880 C
1881 C
1882 C
1883 C
1884 C
1885 C
1886 C
1887 C
1888 C
1889 C
1890 C
1891 C
1892 C
1893 C
1894 C
1895 C
1896 C
1897 C
1898 C
1899 C
1900 C
1901 C
1902 C
1903 C
1904 C
1905 C
1906 C
1907 C
1908 C
1909 C
1910 C
1911 C
1912 C
1913 C
1914 C
1915 C
1916 C
1917 C
1918 C
1919 C
1920 C
1921 C
1922 C
1923 C
1924 C
1925 C
1926 C
1927 C
1928 C
1929 C
1930 C
1931 C
1932 C
1933 C
1934 C
1935 C
1936 C
1937 C
1938 C
1939 C
1940 C
1941 C
1942 C
1943 C
1944 C
1945 C
1946 C
1947 C
1948 C
1949 C
1950 C
1951 C
1952 C
1953 C
1954 C
1955 C
1956 C
1957 C
1958 C
1959 C
1960 C
1961 C
1962 C
1963 C
1964 C
1965 C
1966 C
1967 C
1968 C
1969 C
1970 C
1971 C
1972 C
1973 C
1974 C
1975 C
1976 C
1977 C
1978 C
1979 C
1980 C
1981 C
1982 C
1983 C
1984 C
1985 C
1986 C
1987 C
1988 C
1989 C
1990 C
1991 C
1992 C
1993 C
1994 C
1995 C
1996 C
1997 C
1998 C
1999 C
2000 C

```

1163	CALL WATRHU (Z, IWDN, RMUW)	NUCMUW	1164
1164	UU 7 I=1,8	NUCMUW	1165
1165	WIN(I)=WIN(I)+P*EX(I)*RMUW	NUCMUW	1166
1166	UU 8 IBAND=1,14	NUCMUW	1167
1167	ALAM=WAVE(I,IBAND)	NUCMUW	1168
1168	CALL SCAT (Z,ALAM,V,SIG)	NUCMUW	1169
1169	SIGAVE(IBAND)=SIGAVE(IBAND)*SIG	NUCMUW	1170
1170	CALL UZAL (Z,AUAC)	NUCMUW	1171
1171	UU 9 I=1,4	NUCMUW	1172
1172	AUACAV(I)=AUACAV(I)*AUAC(I)	NUCMUW	1173
1173	CONTINUE	NUCMUW	1174
1174	CALL WCO2P (ZF1,ARG1,1)	NUCMUW	1175
1175	CALL WCO2P (ZF2,ARG2,1)	NUCMUW	1176
1176	WCO2=(ARG2-ARG1)/((ZF1-ZF2)*0.0005048)	NUCMUW	1177
1177	PAVE=PAVE/Z0.0	NUCMUW	1178
1178	UU 11 I=1,8	NUCMUW	1179
1179	W1(I)=WIN(I)/Z0.0	NUCMUW	1180
1180	UU 12 IBAND=1,14	NUCMUW	1181
1181	SIGAVE(IBAND)=SIGAVE(IBAND)/Z0.0	NUCMUW	1182
1182	UU 13 I=1,4	NUCMUW	1183
1183	AUACAV(I)=AUACAV(I)/Z0.0	NUCMUW	1184
1184	RETURN	NUCMUW	1185
1185	END	NUCMUW	1186

```

1186 FUNCTION FLUA(TT,IMARK)
1187
1188 *****
1189 *****
1190 *****
1191 *****
1192 *****
1193 *****
1194 *****
1195 *****
1196 *****
1197 *****
1198 *****
1199 *****
1200 *****
1201 *****
1202 *****
1203 *****
1204 *****
1205 *****
1206 *****
1207 *****
1208 *****
1209 *****
1210 *****
1211 *****
1212 *****
1213 *****
1214 *****
1215 *****
1216 *****
1217 *****
1218 *****
1219 *****
1220 *****
1221 *****
1222 *****
1223 *****
1224 *****
1225 *****
1226 *****
1227 *****
1228 *****
1229 *****
1230 *****
1231 *****
1232 *****
1233 *****
1234 *****
1235 *****
1236 *****

```

AFEL INTERNAL POWER TIME HISTORY FUNCTION

CALLING SEQUENCE

FLUA(=SIGMA,TT,IMST)

#YIELD (KILOTONS)

SIGMA=DENSITY RATIO AT BURST ALTITUDE (NHUA/NHUSL)

TT=TIME WHERE POWER IS DESIRED (SEC)

FLUA=POWER (WATTS)

ROUTINE BY ALFRED L. SHARP - 7 NOVEMBER 1972.

COMMON /SNCM3/ UT, N, SIGMA, TMAX

IF (IMARK.GT.0) GO TO 30

IF (SIGMA.LT.0.0008) GO TO 90

S=1.75995*SIGMA**(-.0432)

PSCM=.17214*SIGMA**(-.6244)*#**53

SIGLOG=ALOG(SIGMA)

TSEM=(1.6828E-4+SIGLOG*(-7.62326E-6+SIGLOG*(-3.72132E-6-SIGLOG*
 1.51933E-7)))#**3123

BHM=2.*SIGMA**1008

BML=SIGMA**1109

TMAX=.036*#**44*SIGMA**(.36)

TMIN=.00256*#**39*SIGMA**(-.062)

UT=(TMAX/TMIN)**.01

IF (TMIN.GT.+.99*TMAX) TMIN=.99*TMAX

PMAX=1.49E13*#**59*SIGMA**(-.45)

PMIN=6.82E11*#**54*SIGMA**(-1.02)

IF (PMIN.GE.+.99*PMAX) PMIN=.99*PMAX

PSEM=PSCM/PMIN-1.

BHR=3.4071*SIGMA**17827

BHL=9.83382*SIGLOG*(-.45+SIGLOG*(-.207-SIGLOG*.007741))

BTIME=1.523*SIGMA**(-.0752)*TMAX

BHITE=1./(.8*SIGMA**3372)-1.

C=(ALOG(TMAX)-ALOG(TMIN))*TMAX/(TMAX-TMIN)

B=5*EXP(C)/C

TM=TMIN/TMAX

HINTMR=PMIN/(TM**(-5)*EXP(-B*EXP(-C*TR)))

PAGE 40

```

1236 MINITMA=PMAX/(EXP(-B*EXP(-C)))
1237 ALPH=-.0209***.3209+1.023***(-.0013979)
1238 C
1239 30 TSMAX=TT/TMAX
1240 IF (TT.EU.0.) GO TO 100
1241 IF (TT.LT.IMIN) GO TO 50
1242 IF (TSMAX-1.) 40+60*60
1243 TETA=(1.-ALOU(1./TSMAX))/ALOU(1./TM)
1244 TETAB=3.14159265359*TETA
1245 MINT=MINTMX*(MINTMN-MINTMX)*((1.-COS(TETAB))/2.)*ALPH
1246 GO TO 70
1247 50 FAC=2.*BHITC/((BTIME/TT)*BNL*(TT/BTIME)*BNK)
1248 PFAC=1./(FAC*1.)
1249 FACM=(2.*BSEM/((TSEM/TT)*BNL*(TT/TSEM)*BNK))*1.
1250 FLUX=PMIN*FAC*FACM
1251 GO TO 80
1252 60 MINT=MINTMX
1253 70 FAC=2.*BHITC/((BTIME/TT)*BNL*(TT/BTIME)*BNK)
1254 PFAC=1./(FAC*1.)
1255 FACM=(2.*BSEM/((TSEP/TT)*BNL*(TT/TSEM)*BNK))*1.
1256 FLUX=MINT*(TSMAX**(-5)*EXP(-B*EXP(-C*TSMAX)))*PFAC*FACM
1257 RETURN
1258 C
1259 90 CALL NUCENR(BFLUX *1)
1260 RETURN
1261 C
1262 100 FLUX=PMIN
1263 RETURN
1264 END

```

NUCMUU 1237
 NUCMUU 1238
 NUCMUU 1239
 NUCMUU 1240
 NUCMUU 1241
 NUCMUU 1242
 NUCMUU 1243
 NUCMUU 1244
 NUCMUU 1245
 NUCMUU 1246
 NUCMUU 1247
 NUCMUU 1248
 NUCMUU 1249
 NUCMUU 1250
 NUCMUU 1251
 NUCMUU 1252
 NUCMUU 1253
 NUCMUU 1254
 NUCMUU 1255
 NUCMUU 1256
 NUCMUU 1257
 NUCMUU 1258
 NUCMUU 1259
 NUCMUU 1260
 NUCMUU 1261
 NUCMUU 1262
 NUCMUU 1263
 NUCMUU 1264
 NUCMUU 1265

1265	C	SUBROUTINE UZAC (ZP, AUAC)	NUCMUO	1266
1266	C		NUCMUO	1267
1267	C	COMPUTES OZONE ABSORPTION COEFFICIENT IN KM-1	NUCMUO	1268
1268	C	ZP IS ALTITUDE IN FEET.	NUCMUO	1269
1269	C	AUAC IS THE RETURNED ATMOSPHERIC OZONE ABSORPTION COEFFICIENT.	NUCMUO	1270
1270	C		NUCMUO	1271
1271	C	UIMENSION AUAC(4), A(4), MU(26), UT(26)	NUCMUO	1272
1272	C		NUCMUO	1273
1273	C	DATA A/210.0,101.0,0.0,0.34,0.0,0.17,0.0,0.4,0.6,0.8,0.10,0.0,0.11,0.0,12.0,	NUCMUO	1274
1274	C	113.0,14.0,0.15,0.16,0.17,0.18,0.21,0.22,0.23,0.24,0.25,0.30,0.32,0.3	NUCMUO	1275
1275	C	24.0,0.38,0.42,0.44,0.47,0.50,0.0,0.07,0.0,0.35,0.0,0.0226,0.0,0.0226,0	NUCMUO	1276
1276	C	3.0,0.35,0.0,0.046,0.0,0.021,0.0,0.028,0.0,0.095,0.0,0.0994,0.0,0.103,0.0,0.122	NUCMUO	1277
1277	C	40.0,0.84,0.0,0.147,0.0,0.192,0.0,0.193,0.0,0.180,0.0,0.0943,0.0,0.0682,0.0,0.0253	NUCMUO	1278
1278	C	5.0,0.00126,0.0,0.0919,0.0,0.00198,0.0,0.000647	NUCMUO	1279
1279	C		NUCMUO	1280
1280	C	Z=ZF*0.00030*0	NUCMUO	1281
1281	C	CALL INT1 (Z,26,MU,UT,U)	NUCMUO	1282
1282	C	DO 1 I=1,4	NUCMUO	1283
1283	C	AUAC(I)=U*A(I)	NUCMUO	1284
1284	C	RETURN	NUCMUO	1285
1285	C	END	NUCMUO	1286

```

1286 1287
1287 1288
1288 1289
1289 1290
1290 1291
1291 1292
1292 1293
1293 1294
1294 1295
1295 1296
1296 1297
1297 1298
1298 1299
1299 1300
1300 1301
1301 1302
1302 1303
1303 1304
1304 1305
1305 1306
1306 1307
1307 1308
1308 1309
1309 1310
1310 1311
1311 1312
1312 1313
1313 1314
1314 1315
1315 1316
1316 1317
1317 1318
1318 1319
1319 1320
1320 1321
1321 1322
1322 1323
1323 1324
1324 1325
1325 1326
1326 1327
1327 1328

```

FUNCTION POWER(TIME,IMARR)
 COMMON /COMMON/ DT, SIGMA, TMAX
 DATA A/1.E-6/
 IF (IMARR.GT.0) GO TO 10
 MU = FLUX(A*DT)
 ALUT=ALUT/DT
 POWER=DE-BPMU
 POWER=0.
 I=1
 TSTEP=TIME/2.
 RETURN
 10
 IF (I-1) 1,1,2
 POWER=POWER
 PLAST=PLAST
 TSTART=TA
 I=2
 GO TO 3
 20
 POWER=0.
 TSTART=TSTART
 TSTOP=TIME+TSTEP
 TLOG=ALUT*(TSTOP-TSTART)
 T=LOG/ALUT
 IF (N*LT.1) N=1
 AMULT=EXP(TLOG/FLUX(M))
 TLAST=TSTART
 GO 4 J=1,N
 T=TLAST*AMULT
 W = FLUX(T*1)
 POWER=POWER+(P*PLAST)*(T-TLAST)
 TLAST=T
 PLAST=W
 4
 TAKES CARE OF THE FACTOR OF 2 FOR TRAPEZOIDAL INTEGRATION AND
 CONVERTS (WATT-SECONDS) TO (BTU)
 POWER=0.0004/413333*POWER
 RETURN
 END

1329	DATA AAAA/	1.00000E+00	2.00000E+00	3.00000E+00	4.00000E+00	NUCMUU	1329
1330	DATA AAAA/	5.00000E+00	6.00000E+00	7.00000E+00	8.00000E+00	NUCMUU	1330
1331	DATA AAAA/	1.00000E+01	1.50000E+01	2.00000E+01	2.50000E+01	NUCMUU	1331
1332	DATA AAAA/	3.00000E+01	3.50000E+01	4.00000E+01	4.50000E+01	NUCMUU	1332
1333	DATA AAAA/	6.00000E+01	6.50000E+01	7.00000E+01	7.50000E+01	NUCMUU	1333
1334	DATA AAAA/	1.00000E+02	1.50000E+02	2.00000E+02	2.50000E+02	NUCMUU	1334
1335	DATA AAAA/	3.00000E+02	3.50000E+02	4.00000E+02	4.50000E+02	NUCMUU	1335
1336	DATA AAAA/	6.00000E+02	6.50000E+02	7.00000E+02	7.50000E+02	NUCMUU	1336
1337	DATA AAAA/	1.00000E+03	1.50000E+03	2.00000E+03	2.50000E+03	NUCMUU	1337
1338	DATA AAAA/	3.00000E+03	3.50000E+03	4.00000E+03	4.50000E+03	NUCMUU	1338
1339	DATA AAAA/	6.00000E+03	6.50000E+03	7.00000E+03	7.50000E+03	NUCMUU	1339
1340	DATA AAAA/	1.00000E+04	1.50000E+04	2.00000E+04	2.50000E+04	NUCMUU	1340
1341	DATA AAAA/	3.00000E+04	3.50000E+04	4.00000E+04	4.50000E+04	NUCMUU	1341
1342	DATA AAAA/	6.00000E+04	6.50000E+04	7.00000E+04	7.50000E+04	NUCMUU	1342
1343	DATA AAAA/	1.00000E+05	1.50000E+05	2.00000E+05	2.50000E+05	NUCMUU	1343
1344	DATA AAAA/	3.00000E+05	3.50000E+05	4.00000E+05	4.50000E+05	NUCMUU	1344
1345	DATA AAAA/	6.00000E+05	6.50000E+05	7.00000E+05	7.50000E+05	NUCMUU	1345
1346	DATA AAAA/	1.00000E+06	1.50000E+06	2.00000E+06	2.50000E+06	NUCMUU	1346
1347	DATA AAAA/	3.00000E+06	3.50000E+06	4.00000E+06	4.50000E+06	NUCMUU	1347
1348	DATA AAAA/	6.00000E+06	6.50000E+06	7.00000E+06	7.50000E+06	NUCMUU	1348
1349	DATA AAAA/	1.00000E+07	1.50000E+07	2.00000E+07	2.50000E+07	NUCMUU	1349
1350	DATA AAAA/	3.00000E+07	3.50000E+07	4.00000E+07	4.50000E+07	NUCMUU	1350
1351	DATA AAAA/	6.00000E+07	6.50000E+07	7.00000E+07	7.50000E+07	NUCMUU	1351
1352	DATA AAAA/	1.00000E+08	1.50000E+08	2.00000E+08	2.50000E+08	NUCMUU	1352
1353	DATA AAAA/	3.00000E+08	3.50000E+08	4.00000E+08	4.50000E+08	NUCMUU	1353
1354	DATA AAAA/	6.00000E+08	6.50000E+08	7.00000E+08	7.50000E+08	NUCMUU	1354
1355	DATA AAAA/	1.00000E+09	1.50000E+09	2.00000E+09	2.50000E+09	NUCMUU	1355
1356	DATA AAAA/	3.00000E+09	3.50000E+09	4.00000E+09	4.50000E+09	NUCMUU	1356
1357	DATA AAAA/	6.00000E+09	6.50000E+09	7.00000E+09	7.50000E+09	NUCMUU	1357
1358	DATA AAAA/	1.00000E+10	1.50000E+10	2.00000E+10	2.50000E+10	NUCMUU	1358
1359	DATA AAAA/	3.00000E+10	3.50000E+10	4.00000E+10	4.50000E+10	NUCMUU	1359
1360	DATA AAAA/	6.00000E+10	6.50000E+10	7.00000E+10	7.50000E+10	NUCMUU	1360
1361	DATA AAAA/	1.00000E+11	1.50000E+11	2.00000E+11	2.50000E+11	NUCMUU	1361
1362	DATA AAAA/	3.00000E+11	3.50000E+11	4.00000E+11	4.50000E+11	NUCMUU	1362
1363	DATA AAAA/	6.00000E+11	6.50000E+11	7.00000E+11	7.50000E+11	NUCMUU	1363
1364	DATA AAAA/	1.00000E+12	1.50000E+12	2.00000E+12	2.50000E+12	NUCMUU	1364
1365	DATA AAAA/	3.00000E+12	3.50000E+12	4.00000E+12	4.50000E+12	NUCMUU	1365
1366	DATA AAAA/	6.00000E+12	6.50000E+12	7.00000E+12	7.50000E+12	NUCMUU	1366
1367	DATA AAAA/	1.00000E+13	1.50000E+13	2.00000E+13	2.50000E+13	NUCMUU	1367
1368	DATA AAAA/	3.00000E+13	3.50000E+13	4.00000E+13	4.50000E+13	NUCMUU	1368
1369	DATA AAAA/	6.00000E+13	6.50000E+13	7.00000E+13	7.50000E+13	NUCMUU	1369
1370	DATA AAAA/	1.00000E+14	1.50000E+14	2.00000E+14	2.50000E+14	NUCMUU	1370
1371	DATA AAAA/	3.00000E+14	3.50000E+14	4.00000E+14	4.50000E+14	NUCMUU	1371
1372	DATA AAAA/	6.00000E+14	6.50000E+14	7.00000E+14	7.50000E+14	NUCMUU	1372
1373	DATA AAAA/	1.00000E+15	1.50000E+15	2.00000E+15	2.50000E+15	NUCMUU	1373
1374	DATA AAAA/	3.00000E+15	3.50000E+15	4.00000E+15	4.50000E+15	NUCMUU	1374
1375	DATA AAAA/	6.00000E+15	6.50000E+15	7.00000E+15	7.50000E+15	NUCMUU	1375
1376	DATA AAAA/	1.00000E+16	1.50000E+16	2.00000E+16	2.50000E+16	NUCMUU	1376
1377	DATA AAAA/	3.00000E+16	3.50000E+16	4.00000E+16	4.50000E+16	NUCMUU	1377

1376	3	1.24900E-01	1.24900E-01	1.34900E-01	1.44600E-01	1.54900E-01	NUL
1379	3	1.76500E-01	2.4900E-01	3.44600E-01	4.55000E-01	5.67400E-01	NUL
1380	3	6.32770E-01	1.0660E+00	1.13300E+00	1.24400E+00	1.40900E+00	NUL
1381	DATA(1222(1),1)=163.2417						
1382	3	1.52000E+00	1.6000E+00	1.66300E+00	1.72100E+00	1.76000E+00	NUL
1383	3	1.76600E+00					NUL
1384	3	7.75000E-02	1.0000E-01	1.09500E-01	1.15000E-01	1.23700E-01	NUL
1385	3	1.44500E-01	1.82000E-01	2.62000E-01	3.53600E-01	5.45700E-01	NUL
1386	3	7.2790E-01	8.94100E-01	1.02200E+00	1.14000E+00	1.32100E+00	NUL
1387	3	1.44100E+00	1.54700E+00	1.62100E+00	1.69900E+00	1.73500E+00	NUL
1388	3	1.74500E+00					NUL
1389	3	8.30000E-02	8.9000E-02	8.9000E-02	1.00000E-01	1.00000E-01	NUL
1390	3	1.17000E-01	1.43000E-01	2.05000E-01	2.78000E-01	4.41000E-01	NUL
1391	3	8.10000E-01	7.84000E-01	7.08000E-01	1.62800E+00	1.22800E+00	NUL
1392	3	1.37600E+00	1.47800E+00	1.56200E+00	1.64900E+00	1.70100E+00	NUL
1393	3	7.12000E-02	7.20000E-02	7.56000E-02	8.00000E-02	8.46000E-02	NUL
1394	3	5.217300E-01	6.65400E-01	8.00000E-01	9.24400E-01	1.13100E+00	NUL
1395	3	1.24800E+00	1.46100E+00	1.50300E+00	1.60000E+00	1.66300E+00	NUL
1396	3	1.6640E+00					NUL
1397	3	6.24000E-02	7.86000E-02	9.50000E-02	1.18500E-01	1.420000E-02	NUL
1400	DATA(1222(1),1)=242.32017						
1401	3	4.39000E-01	5.71000E-01	7.08000E-01	8.53000E-01	1.04900E+00	NUL
1402	3	1.240300E+00	1.32900E+00	1.43200E+00	1.54300E+00	1.61700E+00	NUL
1404	3	5.56000E-02	6.36000E-02	7.67000E-02	8.90000E-02	1.04300E-02	NUL
1405	3	7.23000E-02	8.38000E-02	1.15300E-01	1.55800E-01	2.60000E-01	NUL
1406	3	3.74500E-01	4.98000E-01	5.20600E-01	7.38300E-01	9.46700E-01	NUL
1407	3	1.12000E+00	1.24200E+00	1.37000E+00	1.48100E+00	1.56700E+00	NUL
1409	3	4.75000E-02	7.25000E-02	9.00000E-02	1.13000E-01	2.19000E-01	NUL
1410	3	3.21000E-02	4.32000E-01	5.50000E-01	6.76000E-01	8.61000E-01	NUL
1411	3	1.04000E+00	1.18100E+00	1.30000E+00	1.43700E+00	1.51600E+00	NUL
1413	3	1.51600E-02	4.54000E-02	6.46000E-02	8.62000E-02	1.14800E-01	NUL
1414	3	5.65000E-02	6.46000E-02	8.62000E-02	1.14800E-01	1.89500E-01	NUL
1416	3	2.75900E-01	3.48000E-01	4.28000E-01	5.89800E-01	7.87400E-01	NUL
1417	3	1.11800E-01	1.23400E+00	1.35900E+00	1.46300E+00	1.46300E+00	NUL
1418	3	1.46300E+00					NUL
1419	3	3.90000E-02	4.06000E-02	4.10000E-02	4.30000E-02	4.30000E-02	NUL
1420	DATA(1222(1),1)=321.40317						
1421	3	4.60000E-02	5.60000E-02	6.70000E-02	8.90000E-02	1.42000E-01	NUL
1422	3	2.18000E-01	2.99000E-01	3.88000E-01	4.82000E-01	6.52000E-01	NUL
1423	3	8.26000E-01	9.53000E-01	1.10900E+00	1.23800E+00	1.32600E+00	NUL
1425	3	1.35200E+00					NUL
1426	3	3.33000E-02	4.37000E-02	5.56000E-02	7.17000E-02	1.15200E-01	NUL
1427	3	3.93000E-02					NUL

1459 46

```

1459 C SUBROUTINE DIM1 (XX,YY,ZZ,M,N,AA,Y,Z)
1460 C LINEAR 2 DIMENSIONAL INTERPOLATION
1461 C EXAMPLE-GIVEN THE FOLLOWING TABLE AND A AND Y VALUES,WE CAN
1462 C INTERPOLATE FOR APPROPRIATE Z VALUE
1463 C XX AND YY MUST BE IN INCREASING ORDER
1464 C X1 X2 X3
1465 C Y1 Z1 Z2 Z3
1466 C Y2 Z1 Z2 Z3
1467 C Y3 Z1 Z2 Z3
1468 C DIMENSION XX(N), YY(M), ZZ(M,N)
1469 C IF ((X-XX(1)).OR.(X-XX(N)).OR.((Y-LT.YY(1)).OR.(Y-UT.YY(M)
1470 C 1))) GO TO 70
1471 C WHEN=1
1472 C WHEN=1
1473 C GO TO 1=1,MM
1474 C IF ((YY(1).LE.Y).AND.(YY(1+1).GT.Y)) GO TO 10
1475 C GO TO 20
1476 C C1=(Y-YY(1))/(YY(1+1)-YY(1))
1477 C 11=1
1478 C 12=1+1
1479 C GO TO 30
1480 C CONTINUE
1481 C GO 50 J=1,NN
1482 C IF ((XX(J).LE.X).AND.(XX(J+1).GT.X)) GO TO 40
1483 C GO TO 50
1484 C C2=(X-XX(J))/(XX(J+1)-XX(J))
1485 C J1=J
1486 C J2=J+1
1487 C GO TO 60
1488 C CONTINUE
1489 C CONTINUE
1490 C Z1=ZZ(11+J1)*C1*(ZZ(12+J1)-ZZ(11+J1))
1491 C Z2=ZZ(11+J2)*C1*(ZZ(12+J2)-ZZ(11+J2))
1492 C Z=Z1+C2*(Z2-Z1)
1493 C RETURN
1494 C WRITE (6,Y0) A,Y
1495 C CALL NUCHEM(OMU,INT ,1)
1496 C RETURN
1497 C
1498 C
1499 C
1500 C FORMAT (1H1,COMPARAMETER OUTSIDE RANGE,X=E15.8,ZX,ZHY=E15.8)
1501 C ENU

```

1460 NUCMOD
 1461 NUCMOD
 1462 NUCMOD
 1463 NUCMOD
 1464 NUCMOD
 1465 NUCMOD
 1466 NUCMOD
 1467 NUCMOD
 1468 NUCMOD
 1469 NUCMOD
 1470 NUCMOD
 1471 NUCMOD
 1472 NUCMOD
 1473 NUCMOD
 1474 NUCMOD
 1475 NUCMOD
 1476 NUCMOD
 1477 NUCMOD
 1478 NUCMOD
 1479 NUCMOD
 1480 NUCMOD
 1481 NUCMOD
 1482 NUCMOD
 1483 NUCMOD
 1484 NUCMOD
 1485 NUCMOD
 1486 NUCMOD
 1487 NUCMOD
 1488 NUCMOD
 1489 NUCMOD
 1490 NUCMOD
 1491 NUCMOD
 1492 NUCMOD
 1493 NUCMOD
 1494 NUCMOD
 1495 NUCMOD
 1496 NUCMOD
 1497 NUCMOD
 1498 NUCMOD
 1499 NUCMOD
 1500 NUCMOD
 1501 NUCMOD
 1502 NUCMOD

```

1502 C SUBROUTINE SCAT (ZF,ALAM,V,SIG)
1503 C
1504 C SUBROUTINE SCAT COMPUTES THE SCATTERING COEFFICIENT, KM-1.
1505 C ZF IS THE ALTITUDE IN FEET.
1506 C ALAM IS THE WAVELENGTH IN MICRONS.
1507 C V IS THE SURFACE VISUAL RANGE IN MILES.
1508 C SIG IS THE RETURNED SCATTERING COEFFICIENT, KM-1.
1509 C
1510 C DIMENSION AS(4), AL(4), ES(4), EL(4), VIS(4)
1511 C
1512 C DATA AS/1.22*0.244*0.0486*0.0132/
1513 C DATA AL/0.168*0.0777*0.0301*0.0132/
1514 C DATA ES/-0.033*-0.324*-0.221*-0.0920/
1515 C DATA EL/-0.173*-0.147*-0.122*-0.0920/
1516 C DATA VIS/2.0*10.0*50.0*100.0/
1517 C
1518 C V=MAX1(V,2.0)
1519 C V=MIN1(V,100.0)
1520 C Z=ZF*0.003048
1521 C AM=0.11*Z*0.70
1522 C AMU=(0.55/ALAM)*AM
1523 C DO I=1,4
1524 C IF (V*EL*VIS(I)*1) GO TO 2
1525 C GO TO 3
1526 C I=3
1527 C IF (Z*3.0) GO TO 3
1528 C SIG1=AS(I)*AMU*EXP(ES(I)*Z)
1529 C SIG2=AS(I)*AMU*EXP(ES(I)*V*7)
1530 C GO TO 4
1531 C SIG1=AL(I)*AMU*EXP(EL(I)*Z)
1532 C SIG2=AL(I)*AMU*EXP(EL(I)*V*2)
1533 C SIG=SIG1*(SIG2-SIG1)*(V-VIS(I))/(VIS(I)-VIS(I))
1534 C RETURN
1535 C END

```

1503 NUCMOD
 1504 NUCMOD
 1505 NUCMOD
 1506 NUCMOD
 1507 NUCMOD
 1508 NUCMOD
 1509 NUCMOD
 1510 NUCMOD
 1511 NUCMOD
 1512 NUCMOD
 1513 NUCMOD
 1514 NUCMOD
 1515 NUCMOD
 1516 NUCMOD
 1517 NUCMOD
 1518 NUCMOD
 1519 NUCMOD
 1520 NUCMOD
 1521 NUCMOD
 1522 NUCMOD
 1523 NUCMOD
 1524 NUCMOD
 1525 NUCMOD
 1526 NUCMOD
 1527 NUCMOD
 1528 NUCMOD
 1529 NUCMOD
 1530 NUCMOD
 1531 NUCMOD
 1532 NUCMOD
 1533 NUCMOD
 1534 NUCMOD
 1535 NUCMOD
 1536 NUCMOD

PAGE 48

```

1536 SUBROUTINE TC02(FLAG)
1537
1538 COMMON /SNLM2/  AUCAV(14),  UKM,
1539                TCUM(14),  T0IF(14),  T0Z(14),
1540                TWATW(14),  WCU2,  WIN(B)
1541
1542 COMMON /SNLM4/  PAVE
1543
1544 DIMENSION      C(4),  XK(4),  ALIM(4),  CC(4),  DELNU(4),  XKX(4),
1545                J C1(4),  IUSE(4),  C2(4),  C3(4),  C4(4),  U(4)
1546
1547 DATA IUSE/9.10,11.13,CUNST/.43429440/
1548 DATA C/.050,0.063,0.492,3.15/
1549 DATA XK/.41,0.38,0.39,0.43/
1550 DATA ALIM/3.00,0.50,./
1551 DATA CC/.0,0,0,0.536,0.137,./
1552 DATA DELNU/1500,550,1100,1060,./
1553 DATA XKX/0.40,0.114,0.62,./
1554 DATA U/0.0,0.138,0.77,./
1555
1556 IF (IFLAG .GT. 0) GO TO 10
1557 PALUG=ALUG(PAVE)
1558 DO 1 I=1,4
1559   C1(I)=C(I)*EXP(PALUG*XK(I))
1560   C2(I)=C(I)*CUNST
1561   C3(I)=XKX(I)*CUNST*PALUG*CC(I)
1562   C4(I)=1./DELNU(I)
1563 CONTINUE
1564 RETURN
1565
1566 CONTINUE
1567 UC 0 I=1,4
1568   ***=WCU2*UKM
1569   WSK=SUMT(WXK)
1570   AIN=C1(I)*WSK
1571   IF (AIN.LE.ALIM(I)) GO TO 3
1572   IF (C1(I) .GT. 3.0)
1573     AIN=C2(I)*ALUG(WXK)*C3(I)
1574     AIN=AMAX1(AIN,ALIM(I))
1575     TCU=1.-AIN*C4(I)
1576     IF (TCU) 4,5,5
1577     TCU=0.
1578     J=IUSE(I)
1579     TCUM(J)=TCU
1580 CONTINUE
1581 RETURN
1582 END

```

NUCMUU 1537
 NUCMUU 1538
 NUCMUU 1539
 NUCMUU 1540
 NUCMUU 1541
 NUCMUU 1542
 NUCMUU 1543
 NUCMUU 1544
 NUCMUU 1545
 NUCMUU 1546
 NUCMUU 1547
 NUCMUU 1548
 NUCMUU 1549
 NUCMUU 1550
 NUCMUU 1551
 NUCMUU 1552
 NUCMUU 1553
 NUCMUU 1554
 NUCMUU 1555
 NUCMUU 1556
 NUCMUU 1557
 NUCMUU 1558
 NUCMUU 1559
 NUCMUU 1560
 NUCMUU 1561
 NUCMUU 1562
 NUCMUU 1563
 NUCMUU 1564
 NUCMUU 1565
 NUCMUU 1566
 NUCMUU 1567
 NUCMUU 1568
 NUCMUU 1569
 NUCMUU 1570
 NUCMUU 1571
 NUCMUU 1572
 NUCMUU 1573
 NUCMUU 1574
 NUCMUU 1575
 NUCMUU 1576
 NUCMUU 1577
 NUCMUU 1578
 NUCMUU 1579
 NUCMUU 1580
 NUCMUU 1581
 NUCMUU 1582

```

1583 SUBROUTINE TUIFFS(IFLAG)
1584
1585 C
1586 C
1587 C
1588 C
1589 C
1590 C
1591 C
1592 C
1593 C
1594 C
1595 C
1596 C
1597 C
1598 C
1599 C
1600 C
1601 C
1602 C
1603 C
1604 C
1605 C
1606 C
1607 C
1608 C
1609 C

```

SUBROUTINE TUIFFS(IFLAG)
 SUBROUTINE TUIFFS COMPUTES THE DIFFUSE SCATTERING FUNCTION, TUIFF.
 RF IS THE RANGE FROM THE SOURCE TO THE RECEIVER IN FEET.
 SIGAVE IS THE SCATTERING COEFFICIENT, KM-1.
 TUIF IS THE RETURNED DIFFUSE SCATTERING FUNCTION.

```

      COMMON /SNCM2/  AUCAV(14),  UKM,  SIGAVE(14),
      *  TCUM(14),  TUIF(14),  TUZ(14),  THWL(14),
      *  TMAIW(14),  WCU2,  WJN(8)
      DIMENSION SMCBRT(14),P6SURT(14)
      IF (IFLAG .GT. 0) GO TO 10
      DO 1 IBAND=1,14
      S=SIGAVE(IBAND)
      SMCBRT(IBAND)=S*S*.3333333333
      P6SURT(IBAND)=S*.6*SURT(S)
      RETURN
10  CONTINUE
      DO 2 IBAND=1,14
      TUIF(IBAND)=EXP(-SIGAVE(IBAND)*UKM)*(1.+P6SURT(IBAND)*UKM*EXP(SMCE
      *RT(IBAND)*UKM))
      RETURN
      END

```

1000 50

SUBROUTINE TUZONE

COMPUTES OZONE ABSORPTION FACTOR.

HF IS THE RANGE FROM THE SOURCE TO THE RECEIVER IN FEET.

AUACAV IS THE OZONE ABSORPTION COEFFICIENT IN KM-1.

TLZ IS THE RETURNED OZONE TRANSMITTANCE FACTOR.

COMMON /SRUMCZ/ AUACAV(14), UKM, SIGAVE(14),

* TCOM(14), TUIF(14), THWL(14),

* TATW(14), W(14)

DO 1 I=1,4

TUZ(I)=EXP(-AUACAV(I)*UKM)

RETURN

END

NUCMU 1610
 NUCMU 1611
 NUCMU 1612
 NUCMU 1613
 NUCMU 1614
 NUCMU 1615
 NUCMU 1616
 NUCMU 1617
 NUCMU 1618
 NUCMU 1619
 NUCMU 1620
 NUCMU 1621
 NUCMU 1622
 NUCMU 1623
 NUCMU 1624

PAGE 51

```

1624 C SUBROUTINE TRANS(UPB)
1625 C
1626 C SUBROUTINE TRANS CALCULATES THE ATMOSPHERIC
1627 C TRANSMISSION FACTOR AS A FUNCTION OF WAVELENGTH.
1628 C
1629 C COMMON /SNOM2/ AUACAV(14), URM, SIGAVE(14),
1630 C TCUM(14), TUIF(14), TOZ(14), TRWL(14),
1631 C TWATW(14), WCU2, *IN(8)
1632 C
1633 C DATA TOZ,TWATW,TCUM/42*1.0/
1634 C
1635 C URM=0.000304*UPB
1636 C CALL TUIFS(1)
1637 C CALL TWATER(1)
1638 C CALL TCUZ(1)
1639 C CALL TOZONE
1640 C DO 1 I=1,14
1641 C TRWL(I)=TUIF(I)*TWATW(I)*TCUM(I)*TOZ(I)
1642 C CONTINUE
1643 C RETURN
1644 C END

```

NUCMUU 1625
 NUCMUU 1626
 NUCMUU 1627
 NUCMUU 1628
 NUCMUU 1629
 NUCMUU 1630
 NUCMUU 1631
 NUCMUU 1632
 NUCMUU 1633
 NUCMUU 1634
 NUCMUU 1635
 NUCMUU 1636
 NUCMUU 1637
 NUCMUU 1638
 NUCMUU 1639
 NUCMUU 1640
 NUCMUU 1641
 NUCMUU 1642
 NUCMUU 1643
 NUCMUU 1644
 NUCMUU 1645

```

1645 SUBROUTINE TWATER(FLAG)
1646 C
1647 COMMON /SINCM2/ AUACAV(14), URM, SIGAVE(14),
1648 TCOM(14), TUIF(14), TRWL(14),
1649 TWTW(14), WCUT, WIN(8)
1650 C
1651 DIMENSION
1652 IELNU(7), U(7), C(7), CC(7), ALIM(7), WINW(7), WINS(7), U
1653
1654 DATA CONST/0.43429448/.IUSE/.6.7.9.11.13.14/
1655 DATA ALIM/0.200.200.350.275.200.500./
1656 DATA WINW/WINS/140./
1657 DATA DELNU/0.2240.1000.1500.1100.1000.540./
1658 DATA U/0.0.0.0.460.230.246.0./
1659 DATA C/0.0.0.0.242.152.316.40.2/
1660 DATA CC/0.0.0.0.202.127.337.0./
1661 C
1662 IF (FLAG .GT. 0) GO TO 10
1663 WINW(2)=WIN(4)
1664 WINW(3)=WIN(3)
1665 WINW(4)=WIN(2)
1666 WINW(5)=WIN(5)
1667 WINW(6)=WIN(6)
1668 WINW(7)=WIN(5)
1669 WINW(4)=WIN(2)
1670 WINS(5)=WIN(7)
1671 WINS(6)=WIN(6)
1672 C(1)=0.1*WIN(1)
1673 U(1)=2.7
1674 C(1)=C(1)*SUMT(0.1*WIN(1))
1675 C(1)=1./DELNU(1)
1676 U(2)=4.6
1677 C(2)=0.1*WINS(1)
1678 C(3)=U(1)*CONST
1679 RETURN
1680 CONTINUE
1681 USUMT=SQRT(URM)
1682 TWTW(4)=AMAX1(0.01,-0.02*W)
1683 U(5)=2.7
1684 AIN=C(1)*USUMT
1685 IF (AIN.LE.ALIM(1)) GO TO 4
1686 IF (U(1)) 3,4,3
1687 U=C(2(1)*URM)
1688 AIN=C(1)+C3(1)*ALOG(UR)
1689 AIN=AMAX1(ALIM(1),AIN)
1690 TWT=1.-AIN*U(4)
1691 TWT=AMAX1(0.,TWT)
1692 J=IUSE(1)
1693 TWTW(J)=AMIN1(1.0,TWT)
1694 C
1695

```

PAGE 53

1695
1696

RETURN
END

NUCMU 1696
NUCMU 1697

[illegible]

PAGE 55

```

1747 IF (MFB*LT*HFB) GO TO 3
1748 VFKU=0.5
1749 IF (U2*UC*0.0) GO TO 2
1750 ALFA=ATAN(SUMT(UA**2*UY**2)/((-U2))**57.29576
1751 UBAR=UFB/HFB
1752 CALL INIZ(UBAR*ALFA*NUB*NA*U*ALF*VFKI*VFKU*190)
1753 VFR=VFKU*(1.0-VFKU)*SIN(PIU2*HFB/HFB)
1754 IF (UFB*0.1*HFB) GO TO 4
1755 FPLAQ=1.
1756 RETURN
1757 SINB=MFB/UFB
1758 COST=UAXNA*UY*YNA*U2*ZNA
1759 IF (COST*LT*-SINB) GO TO 7
1760 IF (COST*LT*SINB) GO TO 5
1761 VF=COST
1762 GO TO 6
1763 COSB=SUMT(1.0-SINB**2)
1764 SINT=SUMT(1.0-COST**2)
1765 VU=COSB*COST/(SINB*SINT)
1766 VV=COSB/SINT
1767 VF=(COST*(PI-ACUS(VU)*VU*SUMT(1.0-VU**2))+(ACUS(VV)-VV*SUMT(1.0-VV
1768 *2)))/SINB**2/PI
1769 VF=VF*VFK
1770 CURTINUE
1771 VVVF=VF
1772 RETURN
1773 END

```

NUCMUU 1748
 NUCMUU 1749
 NUCMUU 1750
 NUCMUU 1751
 NUCMUU 1752
 NUCMUU 1753
 NUCMUU 1754
 NUCMUU 1755
 NUCMUU 1756
 NUCMUU 1757
 NUCMUU 1758
 NUCMUU 1759
 NUCMUU 1760
 NUCMUU 1761
 NUCMUU 1762
 NUCMUU 1763
 NUCMUU 1764
 NUCMUU 1765
 NUCMUU 1766
 NUCMUU 1767
 NUCMUU 1768
 NUCMUU 1769
 NUCMUU 1770
 NUCMUU 1771
 NUCMUU 1772
 NUCMUU 1773
 NUCMUU 1774

1774 56

```

1775 C SUBROUTINE WATMD (ZF, JMDN, KMDW)
1776 C SUBROUTINE WATMD COMPUTES WATER VAPOR DENSITY GM/M**3
1777 C ZF IS ALTITUDE IN FEET.
1778 C JMDN IS ATMOSPHERE CODE, 1 = WET ATMOSPHERE.
1779 C KMDW IS RETURNED WATER VAPOR DENSITY, GM/M**3
1780 C
1781 C
1782 ZF=0.0003046
1783 IF (JMDN.EQ.2) GO TO 3
1784 IF (Z.GT.4.0) GO TO 1
1785 KMDW=EXP(3.135-0.3277*Z)
1786 GO TO 6
1787 IF (Z.GT.16.7) GO TO 2
1788 KMDW=EXP(3.962-0.534*Z)
1789 GO TO 6
1790 KMDW=0.007
1791 GO TO 6
1792 IF (Z.GT.5.0) GO TO 4
1793 KMDW=EXP(-0.408*Z)
1794 GO TO 6
1795 IF (Z.GT.15.0) GO TO 5
1796 KMDW=EXP(0.7403-0.5561*Z)
1797 GO TO 6
1798 KMDW=0.0005
1799 RETURN
1800 END

```

NUCMUU 1775
 NUCMUU 1776
 NUCMUU 1777
 NUCMUU 1778
 NUCMUU 1779
 NUCMUU 1780
 NUCMUU 1781
 NUCMUU 1782
 NUCMUU 1783
 NUCMUU 1784
 NUCMUU 1785
 NUCMUU 1786
 NUCMUU 1787
 NUCMUU 1788
 NUCMUU 1789
 NUCMUU 1790
 NUCMUU 1791
 NUCMUU 1792
 NUCMUU 1793
 NUCMUU 1794
 NUCMUU 1795
 NUCMUU 1796
 NUCMUU 1797
 NUCMUU 1798
 NUCMUU 1799
 NUCMUU 1800

PAGE 51

```

1001 C
1002 C
1003 C
1004 C
1005 C
1006 C
1007 C
1008 C
1009 C
1010 C
1011 C
1012 C
1013 C
1014 C
1015 C
1016 C
1017 C
1018 C
1019 C
1020 C
1021 C
1022 C
1023 C
1024 C

SUBROUTINE WCO2P (ZP,ARG,IMM)
  SUBROUTINE WCO2P COMPUTES CO2 CONCENTRATION.
  ZP IS ALTITUDE IN FEET.
  ARG IS RETURNED VALUE.
  IMM IS CODE: 0, RETURN DM/02 IN ATMOS-CM/KM
               1, RETURN * IN ATMOS-CM

  DIMENSION A(4), B(4), ZC(3)
  DATA A/-0.1313,-0.3009,-0.1327,-0.1774/.0/5.520,7.460,5.285,6.112/
  DATA ZC/11.5,13.0,18.5/

  Z=ZF*0.0003048
  DO 1 I=1,3
    IF (Z.LE.ZC(I)) GO TO 2
    CONTINUE
    I=4
  1=4
  ARG=EXP(A(I)*Z*B(I))
  IF (IMM.EQ.1) RETURN
  ARG=A(I)*ARG
  RETURN
END

```

APPENDIX B
QUANTA OUTPUT

PAGE 1

THERE ARE 1 MARVAL MUNS FOR THIS PROBLEM WITH 4 TARGETS, 3 SUB LOCATIONS,
2 TYPES OF AIRCRAFT (TOTAL OF 20), AND 2 TYPES OF MISSILES.
THE REQUIRED SIZE FOR THE INDA ARRAY IS 5
THE SIZE YOU HAVE DESIGNATED IS 5

TARGET	VALUE	LOCATION (DEGREES)	HONWAYS	AIRCRAFT TYPE AND NUMBER
1	100000	13.5833	144.7167	1 20000 2 10000 TAKE-OFF SEQUENCE BY TYPE 1 1 2
2	00000	44.0000	68.0333	1 10000 2 30000 TAKE-OFF SEQUENCE BY TYPE 1 2 2
3	0000	33.7167	84.5167	1 40000 2 50000 TAKE-OFF SEQUENCE BY TYPE 1 1 1 2 2 2
4	100000	33.4333	112.1667	1 40000 2 00000 TAKE-OFF SEQUENCE BY TYPE 1 1 1 1

AIRCRAFT TYPE	RELATIVE VALUE	ORAKE CENTRIGU	PSI	CAL
1	20000	30000	10000	900000
2	100000	40000	15000	1200000

AIRCRAFT TYPE	TAKE-OFF INTERVALS	MINUTES
1	1	0.2500
1	1	0.5000
1	2	0.5000
1	2	0.5000
2	1	0.1250
2	1	0.2500
2	2	0.2500
2	2	0.2500

SUB LOCATIONS (DEGREES)	SUBS	MISSILES AND TYPE
1 30.7500	74.0000	1 18
2 20.0500	93.4500	3 12
3 45.0500	120.0500	2 10

NUMBER OF SUBS OF TYPE	1	2
1 <td>2</td> <td>5</td>	2	5

MISSILE TYPE	LAUNCH INTERVAL	RELIABILITY	MIN RANGE	MAX RANGE	YIELD	HEIGHT OF BURST	NUMBER MVS/MISSILE
1	0.2500	0.9000	250.000	2250.000	1000.0000	7000.00	2.0
2	0.2500	0.8500	650.000	2000.000	800.0000	6000.00	3.0

MISSILE TYPE	TIME	RANGE
1	3.5000	250.0000
2	5.5000	950.0000

7.5000 1600.0000
9.5000 2250.0000

PAGE 2

MISSILE TYPE
2

TIME RANGE
5.0000 650.0000
7.5000 1100.0000
10.0000 1550.0000
12.5000 2000.0000

THE REQUIRED WORK ARRAY SIZE IS 656
THE PROVIDED WORK ARRAY SIZE IS 656
THE REQUIRED WORK ARRAY SIZE IS 656
THE PROVIDED WORK ARRAY SIZE IS 656
THE REQUIRED SIZE FOR THE ALU AND PLAM ARRAYS IS 480
THE SIZE YOU HAVE DESIGNATED IS 480
THE REQUIRED WORK ARRAY SIZE IS 72
THE PROVIDED WORK ARRAY SIZE IS 656

SUBROUTINE PROCESS

AIRCRAFT TYPE 1 OF 2 TIME(S)

VULNERABILITY CRITERIA FOR AIRCRAFT TYPE

1.00 PSI 90 CAL/CM²

ATA INPUT FOR AIRCRAFT PROFILE

CARD NUMBER	GROUND RANGE	FLIGHT TIME	AIRCRAFT ALTITUDE	VELOCITY OF SOUND	MACH NUMBER	LEVEL-OFF ACCELERATION
1	0.00000000	0.00000000	0.00000000	1115.43553240	0.000	0.000
2	1500.00000000	7.50000000	5.00000000	1115.41737491	.130	0.000
3	3700.00000000	17.00000000	7.00000000	1115.40971103	.170	0.000
4	5700.00000000	30.00000000	16.00000000	1115.37522727	.200	0.000
5	7000.00000000	46.00000000	75.00000000	1115.14913559	.250	0.000
6	13000.00000000	63.00000000	230.00000000	1115.55494757	.340	0.000
7	20000.00000000	79.00000000	250.00000000	1115.47825509	.450	0.000
8	33000.00000000	105.00000000	500.00000000	1114.51915304	.600	0.000
9	50000.00000000	130.00000000	1250.00000000	1114.638805.3	.750	0.000
10	63000.00000000	150.00000000	2300.00000000	1107.58910112	.800	0.000
11	72000.00000000	160.00000000	2800.00000000	1105.65638593	.900	0.000
12	90000.00000000	180.00000000	4500.00000000	1097.05971248	.900	0.000
13	98000.00000000	195.00000000	5800.00000000	1093.90035753	.900	0.000
14	105000.00000000	200.00000000	7000.00000000	1085.28615092	.900	0.000
15	110000.00000000	205.00000000	9200.00000000	1080.51229753	.900	0.000
16	115000.00000000	210.00000000	10000.00000000	1077.44003639	.900	0.000

INITIAL ALTITUDE OF MAXIMUM MACH 28000 FEET

ATA COMPUTED FOR AIRCRAFT TYPE 1 AT A TERMINAL ALTITUDE OF 10000.0 FEET

VELOCITY OF SOUND: 1077.44 FEET/SECOND
ACCELERATION COMPONENTS: 0.000 FEET/SEC/SEC

MACH NUMBER: 0.900 TERMINAL 0.900

VELOCITY: 969.70 FEET/SEC FINAL 995.09 FEET/SEC

GROUND RANGE (FEET) FLIGHT TIME (SEC) GROUND RANGE (FT) FLIGHT TIME (MIN)

0.	0.	0.	0.
.150000000E+04	.750000000E+01	.24671053E+00	.125000000E+00
.370000000E+04	.190000000E+02	.6085283E+00	.31666667E+00
.570000000E+04	.300000000E+02	.737500000E+00	.500000000E+00
.700000000E+04	.480000000E+02	.14802832E+01	.800000000E+00
.130000000E+05	.630000000E+02	.21381574E+01	.105000000E+01
.200000000E+05	.795000000E+02	.32894737E+01	.132500000E+01
.330000000E+05	.105000000E+03	.54276316E+01	.175000000E+01
.500000000E+05	.130000000E+03	.82238842E+01	.21666667E+01
.630000000E+05	.150000000E+03	.10361842E+02	.250000000E+01
.720000000E+05	.170000000E+03	.11942105E+02	.26666667E+01

14802632E+02	16118421E+02	32500000E+01
16000000E+03	17207137E+02	3333333E+01
18000000E+03	18042105E+02	3416060E+01
20000000E+03	18914474E+02	35000000E+01
21000000E+03	2048350E+02	36000000E+01
23000000E+03	26053226E+02	37000000E+01
25000000E+03	27622802E+02	38000000E+01
2845454E+06	37191976E+02	39000000E+01
3471272E+06		40000000E+01
40600000E+05		41000000E+01
40500000E+06		42000000E+01
4160000E+06		43000000E+01
41500000E+06		44000000E+01
4731616E+06		45000000E+01
49136361E+06		46000000E+01
50000000E+06		47000000E+01
50000000E+06		48000000E+01
50000000E+06		49000000E+01
50000000E+06		50000000E+01
50000000E+06		51000000E+01
50000000E+06		52000000E+01
50000000E+06		53000000E+01
50000000E+06		54000000E+01
50000000E+06		55000000E+01
50000000E+06		56000000E+01
50000000E+06		57000000E+01
50000000E+06		58000000E+01
50000000E+06		59000000E+01
50000000E+06		60000000E+01
50000000E+06		61000000E+01
50000000E+06		62000000E+01
50000000E+06		63000000E+01
50000000E+06		64000000E+01
50000000E+06		65000000E+01
50000000E+06		66000000E+01
50000000E+06		67000000E+01
50000000E+06		68000000E+01
50000000E+06		69000000E+01
50000000E+06		70000000E+01
50000000E+06		71000000E+01
50000000E+06		72000000E+01
50000000E+06		73000000E+01
50000000E+06		74000000E+01
50000000E+06		75000000E+01
50000000E+06		76000000E+01
50000000E+06		77000000E+01
50000000E+06		78000000E+01
50000000E+06		79000000E+01
50000000E+06		80000000E+01
50000000E+06		81000000E+01
50000000E+06		82000000E+01
50000000E+06		83000000E+01
50000000E+06		84000000E+01
50000000E+06		85000000E+01
50000000E+06		86000000E+01
50000000E+06		87000000E+01
50000000E+06		88000000E+01
50000000E+06		89000000E+01
50000000E+06		90000000E+01
50000000E+06		91000000E+01
50000000E+06		92000000E+01
50000000E+06		93000000E+01
50000000E+06		94000000E+01
50000000E+06		95000000E+01
50000000E+06		96000000E+01
50000000E+06		97000000E+01
50000000E+06		98000000E+01
50000000E+06		99000000E+01
50000000E+06		100000000E+01

PAGE 4

SUBROUTINE PROCESS

AIRCRAFT TYPE 2 OF 2 TYPE(S)

VULNERABILITY CRITERIA FOR AIRCRAFT TYPE

1.50 PSI 120 CAL/CM²

DATA INPUT FOR AIRCRAFT PROFILE

CARD NUMBER	GROUND RANGE	FLIGHT TIME	AIRCRAFT ALTITUDE	VELOCITY OF SOUND	MACH NUMBER	LEVEL-OFF ACCELERATION
1	0.00000000	0.00000000	0.00000000	1116.43653240	0.000	0.200
2	2000.00000000	20.00000000	9.00000000	1116.40204669	.140	0.200
3	3900.00000000	30.00000000	30.00000000	1116.32158255	.140	0.200
4	6200.00000000	40.00000000	92.00000000	1116.06398203	.190	0.200
5	9500.00000000	50.00000000	360.00000000	1115.05635218	.250	0.200
6	13400.00000000	60.00000000	500.00000000	1114.51911546	.310	0.200
7	15600.00000000	66.00000000	500.01000000	1114.51873166	.350	0.200
8	55700.00000000	125.00000000	500.11000000	1114.51873166	.680	0.200
9	61000.00000000	132.00000000	1425.00000000	1110.56327981	.700	0.200
10	68000.00000000	140.00000000	3950.00000000	1101.19625459	.720	0.200
11	75500.00000000	146.00000000	5000.00000000	1097.11196555	.730	0.200

INITIAL ALTITUDE OF MAXIMUM MACH 7600.0 FEET

DATA COMPUTED FOR AIRCRAFT TYPE 2 AT A TERMINAL ALTITUDE OF 5000.0 FEET

VELOCITY OF SOUND: 1097.11 FEET/SECOND

ACCELERATION COMPONENTS: 0.200 FEET/SEC/SEC

MACH NUMBER: .730 TERMINAL .849

VELOCITY:	800.89 FEET/SEC	FINAL	422.80 FEET/SEC	GROUND RANGE (NM)	FLIGHT TIME (MIN)
0.	0.	0.	0.	0.	0.
.20000000E+04	.20000000E+02	.20000000E+02	.20000000E+02	.32894737E+00	.33333333E+00
.39000000E+04	.30000000E+02	.30000000E+02	.30000000E+02	.6414737E+00	.50000000E+00
.62000000E+04	.40000000E+02	.40000000E+02	.40000000E+02	.1019736E+01	.66666667E+00
.95000000E+04	.50000000E+02	.50000000E+02	.50000000E+02	.15625000E+01	.83333333E+00
.13400000E+05	.60000000E+02	.60000000E+02	.60000000E+02	.22039474E+01	.10000000E+01
.15600000E+05	.66000000E+02	.66000000E+02	.66000000E+02	.25657895E+01	.11000000E+01
.55900000E+05	.12500000E+03	.12500000E+03	.12500000E+03	.91940789E+01	.20833333E+01
.61000000E+05	.13200000E+03	.13200000E+03	.13200000E+03	.10032695E+02	.22000000E+01
.68000000E+05	.14000000E+03	.14000000E+03	.14000000E+03	.11164211E+02	.23333333E+01
.75500000E+05	.14800000E+03	.14800000E+03	.14800000E+03	.1241763E+02	.24666667E+01
.77066767E+05	.14996628E+03	.14996628E+03	.14996628E+03	.12678744E+02	.24994381E+01
.76675042E+05	.15153257E+03	.15153257E+03	.15153257E+03	.12943666E+02	.25322095E+01
.80332212E+05	.15369885E+03	.15369885E+03	.15369885E+03	.13212535E+02	.25649809E+01
.81990692E+05	.15586514E+03	.15586514E+03	.15586514E+03	.13485344E+02	.25977523E+01
.83673542E+05	.15763142E+03	.15763142E+03	.15763142E+03	.13762096E+02	.26305237E+01

•C66J2951E*01

•14042790E*02

•C6960665E*01
•C7943808E*01
•J/943808E*01

•14327427E*02
•15204993E*02
•C4311585E*02

•159/9771E*03

•161/6399E*03
•16/66285E*03
•22/66285E*03

•85380163E*05

•8/1110/53E*05
•92466350E*05
•14/8144E*06

PAGE 0

THERE WILL BE 1 MANUAL RUN(S) FOR THIS CASE.
THE CUT-OFF VALUES ARE .50000

***** AIRCRAFT TYPE 1 .VS. MISSILE TYPE 1*****

WHEN DETONATION IS	0.0 NM FROM CENTROID, LETHAL AREA =	177.43 SQUARE NM
WHEN DETONATION IS	1.0 NM FROM CENTROID, LETHAL AREA =	149.20 SQUARE NM
WHEN DETONATION IS	2.0 NM FROM CENTROID, LETHAL AREA =	120.61 SQUARE NM
WHEN DETONATION IS	3.0 NM FROM CENTROID, LETHAL AREA =	111.74 SQUARE NM
WHEN DETONATION IS	4.0 NM FROM CENTROID, LETHAL AREA =	103.06 SQUARE NM
WHEN DETONATION IS	5.0 NM FROM CENTROID, LETHAL AREA =	100.15 SQUARE NM
WHEN DETONATION IS	6.0 NM FROM CENTROID, LETHAL AREA =	106.17 SQUARE NM
WHEN DETONATION IS	7.0 NM FROM CENTROID, LETHAL AREA =	108.15 SQUARE NM
WHEN DETONATION IS	8.0 NM FROM CENTROID, LETHAL AREA =	108.75 SQUARE NM
WHEN DETONATION IS	9.0 NM FROM CENTROID, LETHAL AREA =	112.26 SQUARE NM
WHEN DETONATION IS	10.0 NM FROM CENTROID, LETHAL AREA =	114.96 SQUARE NM
WHEN DETONATION IS	11.0 NM FROM CENTROID, LETHAL AREA =	119.49 SQUARE NM
WHEN DETONATION IS	12.0 NM FROM CENTROID, LETHAL AREA =	124.11 SQUARE NM
WHEN DETONATION IS	13.0 NM FROM CENTROID, LETHAL AREA =	133.85 SQUARE NM
WHEN DETONATION IS	14.0 NM FROM CENTROID, LETHAL AREA =	140.41 SQUARE NM
WHEN DETONATION IS	15.0 NM FROM CENTROID, LETHAL AREA =	152.33 SQUARE NM
WHEN DETONATION IS	16.0 NM FROM CENTROID, LETHAL AREA =	159.66 SQUARE NM
WHEN DETONATION IS	17.0 NM FROM CENTROID, LETHAL AREA =	142.79 SQUARE NM
WHEN DETONATION IS	18.0 NM FROM CENTROID, LETHAL AREA =	145.13 SQUARE NM
WHEN DETONATION IS	19.0 NM FROM CENTROID, LETHAL AREA =	140.94 SQUARE NM
WHEN DETONATION IS	20.0 NM FROM CENTROID, LETHAL AREA =	140.39 SQUARE NM
WHEN DETONATION IS	21.0 NM FROM CENTROID, LETHAL AREA =	149.37 SQUARE NM
WHEN DETONATION IS	22.0 NM FROM CENTROID, LETHAL AREA =	150.56 SQUARE NM
WHEN DETONATION IS	23.0 NM FROM CENTROID, LETHAL AREA =	151.39 SQUARE NM
WHEN DETONATION IS	24.0 NM FROM CENTROID, LETHAL AREA =	156.10 SQUARE NM
WHEN DETONATION IS	25.0 NM FROM CENTROID, LETHAL AREA =	156.72 SQUARE NM
WHEN DETONATION IS	26.0 NM FROM CENTROID, LETHAL AREA =	153.26 SQUARE NM
WHEN DETONATION IS	27.0 NM FROM CENTROID, LETHAL AREA =	153.74 SQUARE NM
WHEN DETONATION IS	28.0 NM FROM CENTROID, LETHAL AREA =	154.16 SQUARE NM
WHEN DETONATION IS	29.0 NM FROM CENTROID, LETHAL AREA =	154.54 SQUARE NM

WHEN DETONATION IS	0.0 NM FROM BRAKE RELEASE, KILL LINE IS	0.0 PLUS	4.7 NM
WHEN DETONATION IS	1.0 NM FROM BRAKE RELEASE, KILL LINE IS	1.0 PLUS	4.4 NM
WHEN DETONATION IS	2.0 NM FROM BRAKE RELEASE, KILL LINE IS	2.0 PLUS	4.1 NM
WHEN DETONATION IS	3.0 NM FROM BRAKE RELEASE, KILL LINE IS	3.0 PLUS	3.8 NM
WHEN DETONATION IS	4.0 NM FROM BRAKE RELEASE, KILL LINE IS	4.0 PLUS	3.5 NM
WHEN DETONATION IS	5.0 NM FROM BRAKE RELEASE, KILL LINE IS	5.0 PLUS	3.3 NM
WHEN DETONATION IS	6.0 NM FROM BRAKE RELEASE, KILL LINE IS	6.0 PLUS	3.3 NM
WHEN DETONATION IS	7.0 NM FROM BRAKE RELEASE, KILL LINE IS	7.0 PLUS	3.2 NM
WHEN DETONATION IS	8.0 NM FROM BRAKE RELEASE, KILL LINE IS	8.0 PLUS	3.1 NM
WHEN DETONATION IS	9.0 NM FROM BRAKE RELEASE, KILL LINE IS	9.0 PLUS	2.5 NM
WHEN DETONATION IS	10.0 NM FROM BRAKE RELEASE, KILL LINE IS	10.0 PLUS	2.7 NM
WHEN DETONATION IS	11.0 NM FROM BRAKE RELEASE, KILL LINE IS	11.0 PLUS	2.7 NM
WHEN DETONATION IS	12.0 NM FROM BRAKE RELEASE, KILL LINE IS	11.6 PLUS	2.6 NM

***** AIRCRAFT TYPE 1 .VS. MISSILE TYPE *****

WHEN DETONATION IS	0.0 NM FROM CENTROID, LETHAL AREA =	142.39 SQUARE NM
WHEN DETONATION IS	1.0 NM FROM CENTROID, LETHAL AREA =	110.97 SQUARE NM
WHEN DETONATION IS	2.0 NM FROM CENTROID, LETHAL AREA =	101.54 SQUARE NM
WHEN DETONATION IS	3.0 NM FROM CENTROID, LETHAL AREA =	91.57 SQUARE NM
WHEN DETONATION IS	4.0 NM FROM CENTROID, LETHAL AREA =	85.60 SQUARE NM
WHEN DETONATION IS	5.0 NM FROM CENTROID, LETHAL AREA =	84.55 SQUARE NM
WHEN DETONATION IS	6.0 NM FROM CENTROID, LETHAL AREA =	80.04 SQUARE NM
WHEN DETONATION IS	7.0 NM FROM CENTROID, LETHAL AREA =	73.45 SQUARE NM
WHEN DETONATION IS	8.0 NM FROM CENTROID, LETHAL AREA =	77.02 SQUARE NM
WHEN DETONATION IS	9.0 NM FROM CENTROID, LETHAL AREA =	100.59 SQUARE NM
WHEN DETONATION IS	10.0 NM FROM CENTROID, LETHAL AREA =	102.75 SQUARE NM
WHEN DETONATION IS	11.0 NM FROM CENTROID, LETHAL AREA =	105.90 SQUARE NM
WHEN DETONATION IS	12.0 NM FROM CENTROID, LETHAL AREA =	115.19 SQUARE NM
WHEN DETONATION IS	13.0 NM FROM CENTROID, LETHAL AREA =	119.66 SQUARE NM
WHEN DETONATION IS	14.0 NM FROM CENTROID, LETHAL AREA =	110.19 SQUARE NM
WHEN DETONATION IS	15.0 NM FROM CENTROID, LETHAL AREA =	121.85 SQUARE NM
WHEN DETONATION IS	16.0 NM FROM CENTROID, LETHAL AREA =	125.41 SQUARE NM
WHEN DETONATION IS	17.0 NM FROM CENTROID, LETHAL AREA =	127.99 SQUARE NM
WHEN DETONATION IS	18.0 NM FROM CENTROID, LETHAL AREA =	129.92 SQUARE NM
WHEN DETONATION IS	19.0 NM FROM CENTROID, LETHAL AREA =	131.41 SQUARE NM
WHEN DETONATION IS	20.0 NM FROM CENTROID, LETHAL AREA =	132.61 SQUARE NM
WHEN DETONATION IS	21.0 NM FROM CENTROID, LETHAL AREA =	133.58 SQUARE NM
WHEN DETONATION IS	22.0 NM FROM CENTROID, LETHAL AREA =	134.39 SQUARE NM
WHEN DETONATION IS	23.0 NM FROM CENTROID, LETHAL AREA =	135.06 SQUARE NM
WHEN DETONATION IS	24.0 NM FROM CENTROID, LETHAL AREA =	135.67 SQUARE NM
WHEN DETONATION IS	25.0 NM FROM CENTROID, LETHAL AREA =	136.18 SQUARE NM
WHEN DETONATION IS	26.0 NM FROM CENTROID, LETHAL AREA =	136.62 SQUARE NM
WHEN DETONATION IS	27.0 NM FROM CENTROID, LETHAL AREA =	137.01 SQUARE NM
WHEN DETONATION IS	28.0 NM FROM CENTROID, LETHAL AREA =	137.36 SQUARE NM
WHEN DETONATION IS	29.0 NM FROM CENTROID, LETHAL AREA =	137.68 SQUARE NM

WHEN DETONATION IS	0.0 NM FROM BRAKE RELEASE, KILL LINE IS	0.0 PLUS	4.4 NM
WHEN DETONATION IS	1.0 NM FROM BRAKE RELEASE, KILL LINE IS	1.0 PLUS	4.2 NM
WHEN DETONATION IS	2.0 NM FROM BRAKE RELEASE, KILL LINE IS	2.0 PLUS	4.0 NM
WHEN DETONATION IS	3.0 NM FROM BRAKE RELEASE, KILL LINE IS	3.0 PLUS	3.7 NM
WHEN DETONATION IS	4.0 NM FROM BRAKE RELEASE, KILL LINE IS	4.0 PLUS	3.4 NM
WHEN DETONATION IS	5.0 NM FROM BRAKE RELEASE, KILL LINE IS	5.0 PLUS	3.1 NM
WHEN DETONATION IS	6.0 NM FROM BRAKE RELEASE, KILL LINE IS	6.0 PLUS	2.9 NM
WHEN DETONATION IS	7.0 NM FROM BRAKE RELEASE, KILL LINE IS	7.0 PLUS	2.9 NM
WHEN DETONATION IS	8.0 NM FROM BRAKE RELEASE, KILL LINE IS	8.0 PLUS	3.2 NM
WHEN DETONATION IS	9.0 NM FROM BRAKE RELEASE, KILL LINE IS	9.0 PLUS	2.8 NM
WHEN DETONATION IS	10.0 NM FROM BRAKE RELEASE, KILL LINE IS	10.0 PLUS	2.4 NM
WHEN DETONATION IS	11.0 NM FROM BRAKE RELEASE, KILL LINE IS	10.5 PLUS	2.7 NM
WHEN DETONATION IS	12.0 NM FROM BRAKE RELEASE, KILL LINE IS	11.0 PLUS	2.4 NM

***** AIRCRAFT TYPE C.VS. MISSILE TYPE *****

WHEN DETONATION IS	0.0 NM FROM CENTRIGU	LETHAL AREA =	59.54 SQUARE NM
WHEN DETONATION IS	1.0 NM FROM CENTRIGU	LETHAL AREA =	40.39 SQUARE NM
WHEN DETONATION IS	2.0 NM FROM CENTRIGU	LETHAL AREA =	44.55 SQUARE NM
WHEN DETONATION IS	3.0 NM FROM CENTRIGU	LETHAL AREA =	45.22 SQUARE NM
WHEN DETONATION IS	4.0 NM FROM CENTRIGU	LETHAL AREA =	50.53 SQUARE NM
WHEN DETONATION IS	5.0 NM FROM CENTRIGU	LETHAL AREA =	54.52 SQUARE NM
WHEN DETONATION IS	6.0 NM FROM CENTRIGU	LETHAL AREA =	50.61 SQUARE NM
WHEN DETONATION IS	7.0 NM FROM CENTRIGU	LETHAL AREA =	60.22 SQUARE NM
WHEN DETONATION IS	8.0 NM FROM CENTRIGU	LETHAL AREA =	66.12 SQUARE NM
WHEN DETONATION IS	9.0 NM FROM CENTRIGU	LETHAL AREA =	71.07 SQUARE NM
WHEN DETONATION IS	10.0 NM FROM CENTRIGU	LETHAL AREA =	77.38 SQUARE NM
WHEN DETONATION IS	11.0 NM FROM CENTRIGU	LETHAL AREA =	84.67 SQUARE NM
WHEN DETONATION IS	12.0 NM FROM CENTRIGU	LETHAL AREA =	91.23 SQUARE NM
WHEN DETONATION IS	13.0 NM FROM CENTRIGU	LETHAL AREA =	104.46 SQUARE NM
WHEN DETONATION IS	14.0 NM FROM CENTRIGU	LETHAL AREA =	116.72 SQUARE NM
WHEN DETONATION IS	15.0 NM FROM CENTRIGU	LETHAL AREA =	127.13 SQUARE NM
WHEN DETONATION IS	16.0 NM FROM CENTRIGU	LETHAL AREA =	136.62 SQUARE NM
WHEN DETONATION IS	17.0 NM FROM CENTRIGU	LETHAL AREA =	150.52 SQUARE NM
WHEN DETONATION IS	18.0 NM FROM CENTRIGU	LETHAL AREA =	139.42 SQUARE NM
WHEN DETONATION IS	19.0 NM FROM CENTRIGU	LETHAL AREA =	141.68 SQUARE NM
WHEN DETONATION IS	20.0 NM FROM CENTRIGU	LETHAL AREA =	143.48 SQUARE NM
WHEN DETONATION IS	25.0 NM FROM CENTRIGU	LETHAL AREA =	144.95 SQUARE NM
WHEN DETONATION IS	30.0 NM FROM CENTRIGU	LETHAL AREA =	146.17 SQUARE NM
WHEN DETONATION IS	35.0 NM FROM CENTRIGU	LETHAL AREA =	147.21 SQUARE NM
WHEN DETONATION IS	40.0 NM FROM CENTRIGU	LETHAL AREA =	148.09 SQUARE NM
WHEN DETONATION IS	45.0 NM FROM CENTRIGU	LETHAL AREA =	148.86 SQUARE NM
WHEN DETONATION IS	50.0 NM FROM CENTRIGU	LETHAL AREA =	149.53 SQUARE NM
WHEN DETONATION IS	55.0 NM FROM CENTRIGU	LETHAL AREA =	150.12 SQUARE NM
WHEN DETONATION IS	60.0 NM FROM CENTRIGU	LETHAL AREA =	150.65 SQUARE NM
WHEN DETONATION IS	65.0 NM FROM CENTRIGU	LETHAL AREA =	151.12 SQUARE NM

WHEN DETONATION IS	0.0 NM FROM SHAK	RELEASE, KILL LINE IS	0.0 PLUS	3.0 NM
WHEN DETONATION IS	1.0 NM FROM SHAK	RELEASE, KILL LINE IS	1.0 PLUS	3.4 NM
WHEN DETONATION IS	2.0 NM FROM SHAK	RELEASE, KILL LINE IS	2.0 PLUS	3.3 NM
WHEN DETONATION IS	3.0 NM FROM SHAK	RELEASE, KILL LINE IS	3.0 PLUS	3.2 NM
WHEN DETONATION IS	4.0 NM FROM SHAK	RELEASE, KILL LINE IS	4.0 PLUS	3.0 NM
WHEN DETONATION IS	5.0 NM FROM SHAK	RELEASE, KILL LINE IS	5.0 PLUS	2.7 NM
WHEN DETONATION IS	6.0 NM FROM SHAK	RELEASE, KILL LINE IS	6.0 PLUS	2.6 NM
WHEN DETONATION IS	7.0 NM FROM SHAK	RELEASE, KILL LINE IS	7.0 PLUS	2.4 NM
WHEN DETONATION IS	8.0 NM FROM SHAK	RELEASE, KILL LINE IS	8.0 PLUS	2.3 NM
WHEN DETONATION IS	9.0 NM FROM SHAK	RELEASE, KILL LINE IS	8.9 PLUS	2.4 NM
WHEN DETONATION IS	10.0 NM FROM SHAK	RELEASE, KILL LINE IS	9.5 PLUS	2.4 NM
WHEN DETONATION IS	11.0 NM FROM SHAK	RELEASE, KILL LINE IS	10.1 PLUS	2.4 NM
WHEN DETONATION IS	12.0 NM FROM SHAK	RELEASE, KILL LINE IS	10.5 PLUS	2.4 NM

***** AIRCRAFT TYPE C VS. MISSILE TYPE *****

WHEN DETONATION IS	0.0 NM FROM	CENTROID,	LETHAL AREA =	40.51	SQUARE NM
WHEN DETONATION IS	1.0 NM FROM	CENTROID,	LETHAL AREA =	39.71	SQUARE NM
WHEN DETONATION IS	2.0 NM FROM	CENTROID,	LETHAL AREA =	37.25	SQUARE NM
WHEN DETONATION IS	3.0 NM FROM	CENTROID,	LETHAL AREA =	36.88	SQUARE NM
WHEN DETONATION IS	4.0 NM FROM	CENTROID,	LETHAL AREA =	44.55	SQUARE NM
WHEN DETONATION IS	5.0 NM FROM	CENTROID,	LETHAL AREA =	48.01	SQUARE NM
WHEN DETONATION IS	6.0 NM FROM	CENTROID,	LETHAL AREA =	50.80	SQUARE NM
WHEN DETONATION IS	7.0 NM FROM	CENTROID,	LETHAL AREA =	54.09	SQUARE NM
WHEN DETONATION IS	8.0 NM FROM	CENTROID,	LETHAL AREA =	59.27	SQUARE NM
WHEN DETONATION IS	9.0 NM FROM	CENTROID,	LETHAL AREA =	64.19	SQUARE NM
WHEN DETONATION IS	10.0 NM FROM	CENTROID,	LETHAL AREA =	70.43	SQUARE NM
WHEN DETONATION IS	11.0 NM FROM	CENTROID,	LETHAL AREA =	76.80	SQUARE NM
WHEN DETONATION IS	12.0 NM FROM	CENTROID,	LETHAL AREA =	81.02	SQUARE NM
WHEN DETONATION IS	13.0 NM FROM	CENTROID,	LETHAL AREA =	91.02	SQUARE NM
WHEN DETONATION IS	14.0 NM FROM	CENTROID,	LETHAL AREA =	101.59	SQUARE NM
WHEN DETONATION IS	15.0 NM FROM	CENTROID,	LETHAL AREA =	107.84	SQUARE NM
WHEN DETONATION IS	16.0 NM FROM	CENTROID,	LETHAL AREA =	111.94	SQUARE NM
WHEN DETONATION IS	17.0 NM FROM	CENTROID,	LETHAL AREA =	114.05	SQUARE NM
WHEN DETONATION IS	18.0 NM FROM	CENTROID,	LETHAL AREA =	117.02	SQUARE NM
WHEN DETONATION IS	19.0 NM FROM	CENTROID,	LETHAL AREA =	118.71	SQUARE NM
WHEN DETONATION IS	20.0 NM FROM	CENTROID,	LETHAL AREA =	120.06	SQUARE NM
WHEN DETONATION IS	21.0 NM FROM	CENTROID,	LETHAL AREA =	121.16	SQUARE NM
WHEN DETONATION IS	22.0 NM FROM	CENTROID,	LETHAL AREA =	122.07	SQUARE NM
WHEN DETONATION IS	23.0 NM FROM	CENTROID,	LETHAL AREA =	122.85	SQUARE NM
WHEN DETONATION IS	24.0 NM FROM	CENTROID,	LETHAL AREA =	123.51	SQUARE NM
WHEN DETONATION IS	25.0 NM FROM	CENTROID,	LETHAL AREA =	124.08	SQUARE NM
WHEN DETONATION IS	26.0 NM FROM	CENTROID,	LETHAL AREA =	124.59	SQUARE NM
WHEN DETONATION IS	27.0 NM FROM	CENTROID,	LETHAL AREA =	125.03	SQUARE NM
WHEN DETONATION IS	28.0 NM FROM	CENTROID,	LETHAL AREA =	125.42	SQUARE NM
WHEN DETONATION IS	29.0 NM FROM	CENTROID,	LETHAL AREA =	125.76	SQUARE NM

WHEN DETONATION IS	0.0 NM FROM	BRAKE RELEASE,	KILL LINE IS	0.0 PLUS	3.3 NM
WHEN DETONATION IS	1.0 NM FROM	BRAKE RELEASE,	KILL LINE IS	1.0 PLUS	3.1 NM
WHEN DETONATION IS	2.0 NM FROM	BRAKE RELEASE,	KILL LINE IS	2.0 PLUS	3.0 NM
WHEN DETONATION IS	3.0 NM FROM	BRAKE RELEASE,	KILL LINE IS	3.0 PLUS	3.0 NM
WHEN DETONATION IS	4.0 NM FROM	BRAKE RELEASE,	KILL LINE IS	4.0 PLUS	2.9 NM
WHEN DETONATION IS	5.0 NM FROM	BRAKE RELEASE,	KILL LINE IS	5.0 PLUS	2.6 NM
WHEN DETONATION IS	6.0 NM FROM	BRAKE RELEASE,	KILL LINE IS	6.0 PLUS	2.4 NM
WHEN DETONATION IS	7.0 NM FROM	BRAKE RELEASE,	KILL LINE IS	7.0 PLUS	2.3 NM
WHEN DETONATION IS	8.0 NM FROM	BRAKE RELEASE,	KILL LINE IS	7.9 PLUS	2.1 NM
WHEN DETONATION IS	9.0 NM FROM	BRAKE RELEASE,	KILL LINE IS	8.6 PLUS	2.1 NM
WHEN DETONATION IS	10.0 NM FROM	BRAKE RELEASE,	KILL LINE IS	9.1 PLUS	2.2 NM
WHEN DETONATION IS	11.0 NM FROM	BRAKE RELEASE,	KILL LINE IS	9.5 PLUS	2.2 NM
WHEN DETONATION IS	12.0 NM FROM	BRAKE RELEASE,	KILL LINE IS	9.9 PLUS	2.2 NM

PAGE 11

THE DISTANCE TO TARGET 1 FROM SUB 1 IS 4000000000 THE AVERAGE DISTANCE TO THE TURN POINT IS 4000 NM
 THE DISTANCE TO TARGET 1 FROM SUB 2 IS 2999999999 THE AVERAGE DISTANCE TO THE TURN POINT IS 4000 NM
 THE DISTANCE TO TARGET 1 FROM SUB 3 IS 2151400000 THE AVERAGE DISTANCE TO THE TURN POINT IS 4000 NM
 THE DISTANCE TO TARGET 2 FROM SUB 1 IS 3374000000 THE AVERAGE DISTANCE TO THE TURN POINT IS 3000 NM
 AFTER 4000000000 MINUTES, WHEN REAPPEAR, 1 ARRIVES, THE RANGES(NM) OF AIRCRAFT ARE
 1000000000 000000
 THE DISTANCE TO TARGET 2 FROM SUB 2 IS 1547000000 THE AVERAGE DISTANCE TO THE TURN POINT IS 3000 NM
 AFTER 4000000000 MINUTES, WHEN REAPPEAR, 2 ARRIVES, THE RANGES(NM) OF AIRCRAFT ARE
 4740000000 3970000000 3510000000
 THE DISTANCE TO TARGET 2 FROM SUB 3 IS 2360000000 THE AVERAGE DISTANCE TO THE TURN POINT IS 3000 NM
 THE DISTANCE TO TARGET 3 FROM SUB 1 IS 3424000000 THE AVERAGE DISTANCE TO THE TURN POINT IS 4000 NM
 AFTER 4000000000 MINUTES, WHEN REAPPEAR, 1 ARRIVES, THE RANGES(NM) OF AIRCRAFT ARE
 1000000000 1000000000 000000 000000 000000
 THE DISTANCE TO TARGET 3 FROM SUB 2 IS 5794000000 THE AVERAGE DISTANCE TO THE TURN POINT IS 4000 NM
 THE DISTANCE TO TARGET 3 FROM SUB 3 IS 2015000000 THE AVERAGE DISTANCE TO THE TURN POINT IS 4000 NM
 THE DISTANCE TO TARGET 4 FROM SUB 1 IS 1874000000 THE AVERAGE DISTANCE TO THE TURN POINT IS 3000 NM
 AFTER 4000000000 MINUTES, WHEN REAPPEAR, 1 ARRIVES, THE RANGES(NM) OF AIRCRAFT ARE
 3100000000 2900000000 2400000000
 THE DISTANCE TO TARGET 4 FROM SUB 2 IS 1017000000 THE AVERAGE DISTANCE TO THE TURN POINT IS 3000 NM
 AFTER 4000000000 MINUTES, WHEN REAPPEAR, 2 ARRIVES, THE RANGES(NM) OF AIRCRAFT ARE
 1400000000 1500000000 1200000000
 THE DISTANCE TO TARGET 4 FROM SUB 3 IS 973000000 THE AVERAGE DISTANCE TO THE TURN POINT IS 3000 NM
 AFTER 4000000000 MINUTES, WHEN REAPPEAR, 3 ARRIVES, THE RANGES(NM) OF AIRCRAFT ARE
 1600000000 1500000000 1000000000

TARGET NUMBER	NUMBER OF MEMBERS ASSOCIATED TO THE SAME NUMBER	NO.	DATE PAID	NUMBER OF BILL	AMOUNT PAID	AMOUNT DUE	DATE PAID	NUMBER OF BILL
1	0.00	0.00	15.00	15.00000	0.00000			
		TARGET	1	14.50	TOTAL PAID OF	0.00000		
2	0.00	0.00	15.00	2.00000	0.00000			
		TARGET	2	14.50	TOTAL PAID OF	1.45555		
3	0.00	0.00	15.00	0.00000	0.00000			
		TARGET	3	14.50	TOTAL PAID OF	0.00000		
4	0.00	0.00	14.00	1.000000	0.00000			
		TARGET	4	14.50	TOTAL PAID OF	0.05555		
5000000.000	TOTALS	0000000000	50000					

$$2.4966 \times 10^{-11} \text{ m} = 2.4966 \text{ \AA}$$

$\frac{d}{dt} \left(\frac{1}{r^2} \right) = -\frac{2}{r^3} \frac{dr}{dt}$

PAGE 13

ITERATION 34
THE MULTIPLE MATRIX HAS CONVERGED USING EPSILON = .1000 THE MAXIMUM DELTA LAMBDA FOUND: .092207

ITERATION 35
THE MULTIPLE MATRIX HAS CONVERGED USING EPSILON = .0100 THE MAXIMUM DELTA LAMBDA FOUND: .006631

ITERATION 36
MULTIPLE MATRIX CONVERGED WITHIN TOLERANCE OF .010000

*****OVERVIEW***** OF THE OPTIMUM ALLOCATION OF MISSILES*****

TARGET	1	VS. SUB LOCATION	BASE	ROUTE	TURN PT	
TARGET 2	VS. SUB LOCATION	BASE	ROUTE	TURN PT		
	1 (2 SUBS)	1.0	1.0	0.0 FROM SALVO	4 (WEAPON GROUP	4)
	1 (2 SUBS)	0.0	1.0	0.0 FROM SALVO	7 (WEAPON GROUP	7)
TARGET 3	VS. SUB LOCATION	BASE	ROUTE	TURN PT		
	1 (2 SUBS)	0.0	1.0	0.0 FROM SALVO	2 (WEAPON GROUP	2)
	1 (2 SUBS)	0.0	1.0	0.0 FROM SALVO	5 (WEAPON GROUP	5)
	1 (2 SUBS)	0.0	1.0	0.0 FROM SALVO	6 (WEAPON GROUP	6)
	1 (2 SUBS)	0.0	1.0	0.0 FROM SALVO	7 (WEAPON GROUP	7)
TARGET 4	VS. SUB LOCATION	BASE	ROUTE	TURN PT		
	1 (2 SUBS)	1.0	0.0	0.0 FROM SALVO	1 (WEAPON GROUP	1)
	1 (2 SUBS)	1.0	0.0	0.0 FROM SALVO	2 (WEAPON GROUP	2)
	2 (3 SUBS)	0.0	0.0	0.0 FROM SALVO	1 (WEAPON GROUP	1)
	3 (2 SUBS)	0.0	0.0	0.0 FROM SALVO	1 (WEAPON GROUP	3)

SUB LOCATION 1 (2 SUBS)

1.0 MISSILES FROM SALVO	1 TO TARGET	4 (1.0 TO BASE)
1.0 MISSILES FROM SALVO	2 TO TARGET	3 (0.0 TO BASE)
1.0 MISSILES FROM SALVO	2 TO TARGET	4 (1.0 TO BASE)
2.0 MISSILES FROM SALVO	4 TO TARGET	2 (1.0 TO BASE)
1.0 MISSILES FROM SALVO	5 TO TARGET	3 (0.0 TO BASE)
1.0 MISSILES FROM SALVO	6 TO TARGET	3 (0.0 TO BASE)
1.0 MISSILES FROM SALVO	7 TO TARGET	2 (0.0 TO BASE)
1.0 MISSILES FROM SALVO	7 TO TARGET	3 (0.0 TO BASE)

SUB LOCATION 2 (3 SUBS)

2.0 MISSILES FROM SALVO	1 TO TARGET	4 (0.0 TO BASE)
-------------------------	-------------	------------------

SUB LOCATION 3 (2 SUBS)

2.0 MISSILES FROM SALVO	1 TO TARGET	4 (0.0 TO BASE)
-------------------------	-------------	------------------

