

AD-779 754

USER'S MANUAL FOR QUANTO-A WEAPON  
ALLOCATION CODE

Karl T. Benson, et al

Air Force Weapons Laboratory  
Kirtland Air Force Base, New Mexico

April 1974

DISTRIBUTED BY:

**NTIS**

National Technical Information Service  
U. S. DEPARTMENT OF COMMERCE  
5285 Port Royal Road, Springfield Va. 22151

UNCLASSIFIED

Security Classification

AD 779 754

DOCUMENT CONTROL DATA - R & D

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

1. ORIGINATING ACTIVITY (Corporate author) Air Force Weapons Laboratory (SAB) Kirtland Air Force Base, New Mexico 87117		2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	
		2b. GROUP	
3. REPORT TITLE USER'S MANUAL FOR QUANTO - A WEAPON ALLOCATION CODE			
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Final report; 1 September 1971-1 October 1973			
5. AUTHOR(S) (First name, middle initial, last name) Karl T. Benson, Capt, USAF; Craig E. Miller, Capt, USAF			
6. REPORT DATE April 1974		7a. TOTAL NO. OF PAGES 130	7b. NO. OF REFS None
8a. CONTRACT OR GRANT NO.		8b. ORIGINATOR'S REPORT NUMBER(S) AFWL-TR-74-20	
b. PROJECT NO. 8809			
c. Task No. 09	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)		
d.			
10. DISTRIBUTION STATEMENT Approved for public release; distribution unlimited.			
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY AFWL (SAB) Kirtland AFB, NM 87117	
13. ABSTRACT (Distribution Limitation Statement A) This report provides instructions for using the QUANTO computer code, a code developed within the Analysis Division of the Air Force Weapons Laboratory (AFWL) to study the vulnerability of aircraft flushing from a nuclear attack from a force of sea-launched ballistic missiles (SLBMs). The structure of the input deck, the array dimensions of concern to the user, the special significance of selected input parameters, and the required job control language are described. The most current version of the QUANTO code is attached as Appendix I. Additional information concerning QUANTO is contained in the AFWL Technical Report AFWL-TR-73-242.			

Reproduced by  
NATIONAL TECHNICAL  
INFORMATION SERVICE  
U S Department of Commerce  
Springfield VA 22151

129

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
QUANTO Allocation Flush Input data Job control Optimization Survivability User's manual Utilization						

ii.

AFWL-TR-74-20

USER'S MANUAL FOR QUANTO - A  
WEAPON ALLOCATION CODE

Karl T. Benson  
Captain USAF

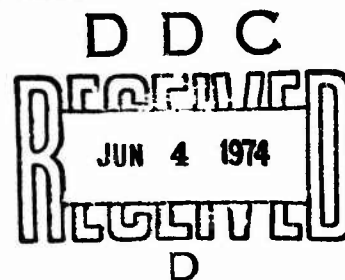
Craig E. Miller  
Captain USAF

Final Report for Period 1 September 1971 through 1 October 1973

TECHNICAL REPORT NO. AFWL-TR-74-20

Approved for public release; distribution unlimited.

iii.



FOREWORD

The research was performed under Program Element 62601F, Project 8809, Task 09.


Inclusive dates of research were 1 September 1971 through 1 October 1973. The report was submitted 18 January 1974 by the Air Force Weapons Laboratory Project Officer, Major Arthur R. Geldbach (SAB).

In the development of the user's manual for the QUANTO code, the experience and advice of Mr. William Peay and Mr. Eugene Omoda have been invaluable in deciding what information to include, in clarifying the user instructions, and in making the code easy to use efficiently.

This technical report has been reviewed and is approved.



ARTHUR R. GELDBACH  
Major, USAF  
Project Officer



GEORGE H. DIMON, JR.  
Colonel, USAF  
Chief, Battle Environments  
Branch



CHARLES C. HYRE, JR.  
Colonel, USAF  
Chief, Analysis Division

## CONTENTS

<u>Section</u>		<u>Page</u>
I	INTRODUCTION	1
II	USING THE PROGRAM	2
	Structure of the Input Deck	2
	Specification of Array Dimensions	17
	Sample QUANTO Run Considerations	18
III	TIPS ON THE UTILIZATION OF QUANTO	27
	Information Flow	27
	Job Control Variations	29
IV	SUMMARY	33
	APPENDIXES	
	I. Current Listing of QUANTO Source Code	35
	II. Sample Input Deck	118
	DISTRIBUTION	122

## TABLES

<u>Table</u>		<u>Page</u>
1	Initial Data Card Format	3
2	Mode Options	4
3	Target Identification Card(s) Format	6
4	Take-Off Sequence Card(s) Format	6
5	Aircraft Card Format	7
6	Take-Off Intervals Card Format	7
7	Submarine Location Card Format	8
8	SLBM Parameters Card Format	9
9	Missile Trajectory Card(s) Format	10
10	Altitudes Card Format	11
11	Aircraft Flight Profile Data Card Format	12
12	Control Variables Card Format	14
13	Nuclear Effects Parameters Card Format	15
14	Overpressure Card Format	15
15	Thermal Card Format	16
16	Convergence Parameters Format	17
17	Arrays of QUANTO with Problem-Dependent Dimensions	21
18	Problem-Dependent Dimensions in QUANTO	26

SECTION I  
INTRODUCTION

The QUANTO computer code and its mathematical model were developed within the Analysis Division of 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. Using the technique of Lagrangian multiplier optimization, a near-optimal allocation of SLBMs to targets is produced. In addition, the submarine positions or the aircraft beddown are optimized if requested. The code has considerable flexibility in the type of input data it will permit through (1) its automatic consideration of specific values of aircraft hardnesses, flyout profiles, level-off altitudes, and kill values and (2) its treatment of an attack by multiple types of SLBMs, which are described by differing missile trajectories, yields, launch intervals, reliabilities, and numbers of missiles per submarine.

The report contains information which enables the user of QUANTO to construct a data deck and make a successful run. In addition, the limits on the numbers of data array elements currently permitted in a QUANTO run are described. However, the user is provided instructions for redimensioning arrays to accommodate larger problems or to reduce core requirements for smaller problems. The most recent version of the program is listed in appendix I. Sample control language and comments concerning several of the input parameters are also included.

This document is intended as a supplement to the AFWL Technical Report AFWL-TR-73-242, "QUANTO--A Code to Optimize Weapon Allocations," which describes the mathematical model and methods of optimization used by QUANTO. An example of an input deck, its output, and brief descriptions of the major subroutines appear in the referenced report.

## SECTION II

### USING THE PROGRAM

The user of QUANTO must construct a data deck and verify that the dimensions of arrays in DIMENSION and COMMON statements are sufficient for the problems described. The format of the input data will first be described. Then a discussion will be given of the DIMENSION and COMMON statements of concern when problem size forces enlargement or adjustment of array sizes.

#### 1. STRUCTURE OF INPUT DECK

The input deck for a single problem is arranged in sections in the following order:

- a. Initial data card
- b. Beddown data
- c. Aircraft parameters
- d. Submarine positions
- e. Missile data
- f. Aircraft profile and nuclear effects parameters
- g. Convergence parameters
- h. Initial allocation

When multiple problems are to be run in a single job submission, the data decks for all problems may be stacked and input in a single stream.

#### a. Initial Data Card

The first card of any input problem contains basic data necessary to describe the problem. This card is described in table 1. Table 2 shows options.

#### b. Beddown Data

The aircraft beddown data is input as a set of cards for each base capable of having alert aircraft of any kind. The first card(s) of the set identifies the base. Each target (base) identification is followed by a take-off sequence list for alert aircraft on that base. The formats of these cards are described in tables 3 and 4.

Table 1  
INITIAL DATA CARD FORMAT  
(FORMAT (14I5))

<u>Columns</u>	<u>Program variable name</u>	<u>Description</u>
1-5	NTGTS	Number of bases in the list of aircraft beddown.
6-10	NSUBS	Number of candidate positions for submarines in the list of submarine positions.
11-15	NTYPES	Number of types of aircraft.
16-20	MXRWAY	Maximum number of runways on any one base.
21-25	MTYPES	Number of types of missiles.
26-30	IOUT	A control variable to limit output.  If IOUT=2, intermediate multiplier and allocation output, useful for debugging, will be output during the convergence to the optimal laydowns.  If IOUT=1, the intermediate output mentioned above will be suppressed.
31-35	ISOPT	A control variable to control whether or not optimization of submarine positioning among the candidate positions is performed.  If ISOPT=1, submarine positioning will NOT be optimized among the input submarine locations.  If ISOPT=2, submarine positioning will be optimized.
36-40	IVOPT	A control variable to control whether or not the beddown of aircraft will be optimized among the given bases.  If IVOPT=1, the beddown will NOT be optimized.  If IVOPT=2, the beddown will be optimized.  <u>NOTE:</u> If both ISOPT and IVOPT are set equal to 2, the submarine positioning is optimized first, then fixed in these locations for the beddown optimization.
41-45	NCASE	An arbitrary "case number" which the user may use to identify his problem.
46-50	MODE	A multipurpose control variable which is used to specify one of several operational modes. Table 2 describes the mode options.

Table 2

MODE OPTIONS

<u>Mode</u>	<u>Purpose</u>	<u>Input</u>	<u>Execution and termination</u>
0	The user may edit the major portion of his input to see if it is read properly by QUANTO without risking the computer time required for optimizations. At the same time, the QUANTO tasks of constructing the aircraft profiles and computing lethal areas may be checked for validity and exceptional conditions which may not have been foreseen.	Problem description data, without the convergence parameters and initial allocation, is read from data cards.	Execution of problem terminates after processing of aircraft profile and nuclear effects information. No restart tape is written.
1	The reading of the entire input deck and the computations of survivabilities and kill from the initial laydown may be tested without risking the computer time required for the iterative optimizations. If all operations are satisfactory, the job may be continued using the results of these computations by reading the created restart tape with a MODE=2 job.	Complete input deck is read from data cards.	Execution terminates after computation of survivabilities and expected kill from initial allocation. A restart tape (with all necessary data for continued operation with MODE=2) is written.
2	A MODE=1, 2, or 3 job may be continued using the results of all computations completed when the previous job terminated. A restart tape from the previous job supplies the intermediate results.	Intermediate problem description data is read from restart tape. Only initial data card is input in card format.	Execution terminates following the writing of a restart tape after any laydown convergence which terminates within 30 seconds of the job's time limit.

Table 2 (cont'd)

<u>Mode</u>	<u>Purpose</u>	<u>Input</u>	<u>Execution and termination</u>
3	The entire problem, from input of the data deck through optimization, may be attempted. Computer time may be wasted in the optimization if input errors exist. If the time limit is reached, all intermediate computational results will be preserved on a restart tape.	Complete input deck is read from data cards.	Execution terminates as for MODE=2.

Table 3\*

TARGET IDENTIFICATION CARD(S) FORMAT  
(FORMAT (3F10.4, I5, 5X, 3F10.4/(7F10.4)))

<u>Columns</u>	<u>Program array name</u>	<u>Description</u>
1-10	TGTLAT	Latitude coordinates of the base's brake release point (start of take-off roll), in degrees (northern hemisphere is assumed).
11-20	TGTLNG	Longitude coordinates of the base's brake release point (start of take-off roll), in degrees (west of Greenwich mean line is assumed).
21-30	DTCENT	Distance from brake release point to the centroid of the dispersing aircraft, in nautical miles (NM).
31-35	NRWAYS	Number of runways, used to determine take-off intervals between aircraft on this base.
41-50 51-60 61-70	VAL	Number of aircraft of Type 1, then Types 2, 3, ... , until the numbers of all types of aircraft on this base have been listed.
Following card(s) if necessary in 7F10.4 format.		

\*Note that if only three or fewer types of aircraft are included in the problem and no base has more than 70 alert aircraft, then only two cards will be necessary for each base, so the beddown data will consist of consecutive pairs of cards, one pair per target.

Table 4\*

TAKE-OFF SEQUENCE CARD(S) FORMAT  
(FORMAT (70I1))

<u>Columns</u>	<u>Program array name</u>	<u>Description</u>
1,2,3,... 70 1,2,...	ISEQ	The type numbers of the alert aircraft on the base, in the order of take-off, for as many cards as needed.

\*Note that if only three or fewer types of aircraft are included in the problem and no base has more than 70 alert aircraft, then only two cards will be necessary for each base, so the beddown data will consist of consecutive pairs of cards, one pair per target.

## c. Aircraft Parameters

The point values assigned to each aircraft, brake release times (for each type of aircraft if first to take off), aircraft hardinesses, and take-off intervals follow the beddown data in the input deck. First, one card per aircraft is input, in order by aircraft type number, giving the first three of these items; this card will be called the "aircraft card." Then the take-off intervals are listed in a specific order on the subsequent card(s). The formats of these cards are described in tables 5 and 6.

Table 5

AIRCRAFT CARD FORMAT  
(FORMAT (3F10.4, I5))

<u>Columns</u>	<u>Program array name</u>	<u>Description</u>
1-10	RELVAL	The number of value points each aircraft of the given type is worth to the enemy when killed.
11-20	BRTIME	The time between launch of the first SLBMs and the brake release time (start of take-off roll) of the given type aircraft, if that type aircraft is the first to take off at any base, in minutes.
21-30	PSI	The number of psi (pounds per square inch) of over-pressure which kills the aircraft.
31-35	ICAL	The number of cal/cm <sup>2</sup> of incident free-field thermal energy which kills the aircraft.

Table 6

TAKE-OFF INTERVALS CARD FORMAT  
(FORMAT (7F10.4))

<u>Columns</u>	<u>Program array name</u>	<u>Description</u>
1-10 11-20 21-30 etc., through 70; then repeat on subsequent cards.	DELTAC	The take-off intervals between each ordered pair of types of aircraft, for bases with each possible number of runways, are listed in specific order. First, for single runways, the intervals between the aircraft-type pairs (1,1),(1,2),(1,3),..., (1,NTYPES),(2,1),(2,2),..., (2,NTYPES),(3,1),..., (NTYPES,NTYPES), are input in that order. A set of intervals in this order is then input for dual runways, then triple, etc., until all numbers of runways which occur on input bases have been covered.

## d. Submarine Positions

One data card is input per candidate submarine location. A submarine location is characterized not only by the type of missile (or submarine) which may be located there, but by the number of missiles on that type of submarine. Type numbers are applied to submarine locations to indicate the type of submarine which may be located there. In the automated relocation of submarines, shifts of submarines may occur only between locations which may have like types of submarines. So the numbering of submarine types may be constructed so as to prevent submarines from moving to certain points or jumping from one ocean to another. If more than one type of submarine can be at the same location, that point must be input once for each candidate type of submarine. The format of the submarine location card is described in table 7.

Table 7  
SUBMARINE LOCATION CARD FORMAT  
(FORMAT (2F10.4, 3I5))

<u>Columns</u>	<u>Program array name</u>	<u>Description</u>
1-10	SUBLAT	Latitude coordinates of the candidate submarine location in degrees (north hemisphere is assumed).
11-20	SUBLNG	Longitude coordinates of the candidate submarine location in degrees (west of Greenwich mean line is assumed).
21-25	ISUBS	Number of submarines stationed at the submarine location, prior to submarine relocation (if requested). This may be zero.
26-30	NMPS	Number of missiles on all submarines at the location which will be dedicated to aircraft kills.
31-35	MTYPE	Type-number of the submarines (or SLRMs) which may be located at the point.

## e. Missile Data

The type numbers appearing in the submarine location input refer to the type of submarine. Since each submarine can carry, at most, one type of missile, the submarine type is equivalent to the missile type. For each type submarine, parameters describing the SLBM must be input. Sets of cards describing each

SLBM are input sequentially by submarine type number. Each set of cards for a single SLBM contains

- (1) The SLBM parameters card
- (2) The missile trajectory card(s)

If two different submarine type numbers are used for the same type SLBM merely to prevent submarines from shifting to certain points, the set of cards describing that SLBM must be input twice. The formats of these cards are described in tables 8 and 9.

Table 8  
SLBM PARAMETERS CARD FORMAT  
(FORMAT (7F10.4))

<u>Columns</u>	<u>Program array name</u>	<u>Description</u>
1-10	DELTM	The time interval between salvo launches for successive missiles of the given type, in minutes.
11-20	RELML	The launch reliability of the missile (i.e., the percent which launch successfully on the average).
21-30	RELMF	The in-flight reliability of the missile (i.e., the percent which successfully reach the target area on the average).
31-40	RELMWH	The warhead reliability of the missile (i.e., the percent which successfully detonate upon reaching the target on the average).
41-50	RNGMIN	The minimum range of the missile, in NM.
51-60	RNGMAX	The maximum range of the missile, in NM.
61-70	YIELD	The assumed yield of the missile warhead, in KT.

The tabular time/distance pairs of table 9 are used to compute missile arrival times on each target, with four-point Lagrangian interpolation supplying times for distances not appearing in the table.

Table 9

MISSILE TRAJECTORY CARD(S) FORMAT  
(FORMAT (15, 5X, (6F10.4)))

<u>Columns</u>	<u>Program variable or array name</u>	<u>Description</u>
1-5	MPR	Number of time/distance pairs which describe the times associated with the missile trajectories to various distances. These pairs immediately follow MPR.
11-20	FMTIME	The flight time, in minutes, which the missile requires to reach a target a given number of NM from the launch point, where this distance is given immediately following this time on the input card.
21-30	FMRNG	The distance, in NM, corresponding to the preceding missile flight time.
31-40	FMTIME	Succeeding time/distance pairs, in ascending order by time, which describe the missile flight times.
41-50	FMRNG	
51-60	FMTIME	
61-70	FMRNG	

Succeeding cards in (6F10.4) format until all MPR pairs have been input.

f. Aircraft Profile and Nuclear Effects Parameters

A set of cards is input for each aircraft type, with the sets in order by aircraft-type number. Each set consists of the following:

- (1) Altitudes card
- (2) Count card
- (3) Aircraft flight profile data cards
- (4) Control variables card
- (5) Nuclear effects parameters card
- (6) Lethal radii cards

The formats of these cards are described in the following tables.

(1) Altitudes Card

This card indicates the final level-off altitude of the aircraft of the given type and the height of burst (HOB) of the SLBMs. Both of these altitudes are relative to the ground. Thus, if each base has only one type of

aircraft stationed there, different HOBs may be selected for each type aircraft. However, this will cause misleading results if a base has more than one type of aircraft, since the lethal area computations for one SLBM against two different types of aircraft will be based on two different HOBs for the single weapon. The format of this card appears in table 10.

Table 10  
ALTIMITUDES CARD FORMAT  
(FORMAT (6F10.4))

<u>Columns</u>	<u>Program variable name</u>	<u>Description</u>
1-10	AZ	The final level-off altitude of the aircraft in its flyout profile, in feet above the ground.
11-20	HBL	The detonation height of SLBMs used in computing lethal areas for the aircraft, in feet above the ground.

### (2) Count Card

The count card indicates only the number of cards following it which contain the aircraft profile information. Columns 1 through 5 of this card should contain this integral number of cards, and the remaining columns may be used for comments.

### (3) Aircraft Flight Profile Data

The aircraft flight profile data is input in a set of data cards which describe the aircraft movement in time and space. The flight profile for a given type of aircraft consists of values for distance, time, altitude, and the horizontal components of velocity and level-off acceleration (as a function of altitude). These values are specified for a nonturning aircraft.

Distance and time values are measured relative to brake release, the point at which the aircraft begins moving on the runway for take-off. The values for altitude are measured relative to ground level and must be non-decreasing. The level-off altitude may be specified as any altitude less than or equal to the maximum (i.e., 1as ) altitude value of the aircraft flight profile. The values for velocity, distance, time, and acceleration of the aircraft when it levels off are obtained through interpolation among the

altitude values of the profile based on the input level-off altitude. The parameters associated with level-off are used in the standard equations of motion to accelerate the aircraft to its final escape velocity. The final escape velocity is assumed to be the maximum velocity, expressed as Mach number, which occurs in the input aircraft flight profile data, although it is input separately as FMACH in the control variables card to be described next. An exception to this rule is made when XATI=0.0 (in the control variables card-- see next paragraph), in which no acceleration takes place after level-off. The user is required to input values for altitude which are nondecreasing, and these values are automatically adjusted, if necessary, to ensure strict monotonicity for purposes of interpolation. The aircraft flight profile description should consist of approximately the maximum of 50 cards currently permitted by array dimensions in QUANTO. The format and content of each card are described in table 11.

Table 11  
AIRCRAFT FLIGHT PROFILE DATA CARD FORMAT  
(FORMAT (3F15.8, 2F10.6))

<u>Columns</u>	<u>Program array name</u>	<u>Description</u>
1-15	F	The ground range from the brake release point, in feet, indicating the distance traveled in the following time.
16-30	G	The time from brake release, in seconds, at which the aircraft reaches the preceding distance.
31-45	A	The altitude of the aircraft, in feet above ground elevation, at the preceding time.
46-55	VEL	The Mach number describing the aircraft's ground speed at the time indicated in this card.
55-65	ACCEL	The acceleration, in feet per second per second at which the aircraft would accelerate to its final escape velocity if it leveled off at a lesser velocity at the altitude on this card.

#### (4) Control Variables Card

This card supplies the values of variables which control the operation of generating a completed aircraft flight profile for use in the

program. The format of this card is described in table 12, and additional information concerning the variables is in the following narrative.

The variables, FALTCM and FMACH, are determined by the input aircraft flight profile. The aircraft is assumed to follow the input flight profile until the level-off altitude is reached. If  $XATI > 0.0$ , the aircraft is then accelerated to its final escape velocity which is assumed to be the maximum Mach number which occurs in the input profile and is specified separately as FMACH. The lowest altitude in the input profile data at which the aircraft reaches this final escape velocity is specified by FALTCM. If the level-off altitude is less than FALTCM, then the velocity upon leveling off is less than FMACH. The standard equations of motion are then used to accelerate the aircraft to its final escape velocity, if the variable XATI is greater than zero.

The variable XATI specifies the number of points which QUANTO generates in accelerating the aircraft to its final escape velocity. If XATI is equal to 0.0, then the velocity upon leveling off is used as the final escape velocity, and no acceleration takes place. The variable XATI should be chosen such that a sufficient number of points of distance and time are generated for the acceleration phase to permit reasonably accurate interpolation.

The distance associated with the point that the aircraft achieves its final escape velocity is usually not great enough for the lethal area determination routines to use in computing lethal area of the most distant SLBM weapon detonations. The variable  $TI(I)$  is used to extend the profile to the greatest distance needed. This is the greatest distance from the centroid that is necessary to compute lethal area for the placement of the last potential weapon arriving on that target.

The choice of  $TI(I)$  and XATI must be made so that the complete aircraft profile is generated with sufficient spacing in the data to accommodate four-point Lagrangian (cubic) interpolation. An upper limit of 99 total data points in the complete aircraft flight profile is permitted by the present array dimensions in QUANTO. Currently used values for  $TI(I)$  and XATI are 60 seconds and 10 points, respectively.

Table 12

CONTROL VARIABLES CARD FORMAT  
(FORMAT (3F15.8, 2F10.6))

<u>Columns</u>	<u>Program variable name</u>	<u>Description</u>
1-15	FALTCM	The altitude, in feet above ground elevation, from the aircraft flight profile, at which the aircraft first reaches its final escape velocity.
16-30	FMACH	The final escape velocity of the aircraft, expressed as Mach number, from the aircraft flight profile. The aircraft will not exceed this velocity during flush.
31-45	TI(I)	The time interval, in seconds, at which data points for distance will be generated after the aircraft (of type I) reaches its final escape velocity.
46-55	XATI	The number of equally spaced points in time for which distances will be generated to accelerate the aircraft from its velocity at its level-off altitude to its final escape velocity.

## (5) Nuclear Effects Parameters Card

This card contains parameters which indicate the environment in which the nuclear detonations take place. These parameters are arguments required for the nuclear effects subroutines SABERCM and SNAPTCM. Table 13 describes the format of this card.

## (6) Lethal Radii Cards

If the lethal overpressure radius and time of shock arrival and/or the lethal thermal radius are already known for the hardnesses of some or all aircraft types against some or all missile types for the case in which the aircraft has reached its terminal altitude by the time it encounters the lethal region, some of the computations of SABERCM and SNAPTCM may be bypassed by inputting the necessary values in cards following the nuclear effects parameters card. A pair of cards, the "overpressure card" and the "thermal card," must be input for each combination of aircraft type and missile type. These pairs are input in ascending order by aircraft type and by missile type within aircraft type. These cards must be input even if the values are not known, but the cards may be left blank. The card formats are in tables 14 and 15.

Table 13

NUCLEAR EFFECTS PARAMETERS CARD FORMAT  
(FORMAT (6F10.8))

<u>Columns</u>	<u>Program variable name</u>	<u>Description</u>
1-10	HTE	The assumed ground elevation, in feet above sea level, of the target air bases from which this aircraft flies. This currently must be zero since QUANTO's computations of lethal areas are not reaccomplished for each base.
11-20	BETIND	The "beta ID," an indicator which the subroutine SNAPTCM requires to compute the effect of thermal energy on the aircraft by considering the aircraft as a horizontal panel (when BETIND=0.0) or as a panel oriented for maximum perpendicular incident thermal energy (when BETIND=1.0). QUANTO uses BETIND=1.0, assuming some panel of the aircraft is in the worst orientation.
21-30	RHO	The albedo factor, which is the fraction 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.
31-40	VIS	The visibility in miles; 10 for a SAC Normal Day.
41-50	PZ	The water vapor in the air, in millimeters; 5 for a SAC Normal Day.
51-60	HSL	The altitude in feet above sea level of the top of the haze layer; 10,000 for a SAC Normal Day.

Table 14

OVERPRESSURE CARD FORMAT  
(FORMAT (I5, 2F15.8))

<u>Columns</u>	<u>Program variable name</u>	<u>Description</u>
1-5	ISABER	An indicator of whether the following two fields of this card are to be used. If ISABER=1, the values are to be used; otherwise, input ISABER=0 and QUANTO will compute the values.
6-20	HORF	The lethal overpressure radius, in feet.
21-35	TSA	The time of shock arrival, in seconds.

Table 15  
THERMAL CARD FORMAT  
(FORMAT (I5, 2F15.8))

<u>Columns</u>	<u>Program variable name</u>	<u>Description</u>
1-5	ISNAPT	An indicator of whether the following field on this card is to be used. If ISNAPT=1, the value is to be used; otherwise, input ISNAPT=0, and QUANTO will compute the value.
6-20	SZ	The lethal thermal radius, in feet.

#### g. Convergence Parameters

Several parameters may be input to control the termination of QUANTO's iterative procedure for converging the  $\lambda_{ij}$ 's in computing the optimal laydown. As described in AFWL-TR-73-242, the procedure terminates (1) when the  $\lambda_{ij}$  matrix is converged to some tolerance  $\epsilon$ , (2) when the expected kill does not significantly increase for a given number of iterations, or (3) when the number of iterations reaches a maximum specified by the user. The convergence parameters are input by the user in the format described in table 16 to indicate the final tolerance  $\epsilon$ , the number of kills used for testing for a "significant increase," the interval of iterations over which the kill increase is measured, and the maximum number of iterations. Current values of the parameters listed in table 16 are 0.01, 20, 100, and 0.0001, in the order of the table.

#### h. Initial Allocation

A missile laydown must be input to QUANTO to start the iterative procedure for improving the laydown. The initial allocation of missiles to targets is input submarine by submarine. The submarines are in the order of the submarine locations, with all submarines at one location preceding the first submarine at the next location. One or more cards, in the format (14I5), are input for each submarine where the numbers on the cards are the target numbers to which the SLBMs from that submarine (in ascending order by salvo or launch time) are allocated. One card per submarine is sufficient if each submarine has 14 or fewer missiles; otherwise, multiple cards per submarine are needed with a new card started for each new submarine.

Table 16

CONVERGENCE PARAMETERS FORMAT  
(FORMAT (F10.4, 2I5, F10.4))

<u>Columns</u>	<u>Program variable name</u>	<u>Description</u>
1-10	CHGKIL	The number of kills used to distinguish between a significant kill increase (in ITCUT2 iterations--see below) and an insignificant increase.
11-15	ITCUT1	The maximum number of groups of ITCUT2 iterations (see below) which will be accomplished by QUANTO in any lay-down optimization procedure before the iteration procedure is terminated by QUANTO.
16-20	ITCUT2	The number of iterations in each group, where the increase in expected kill resulting from each group of iterations is tested against the amount CHGKIL to determine if the increase was large enough to warrant continuing the iterations through another group.
21-30	EPSCUT	A tolerance quantity used to test for convergence of the $\lambda_{ij}$ matrix. QUANTO actually converges the $\lambda_{ij}$ matrix to tolerances of $\epsilon=.1$ , then 0.01, then 0.001, etc., until convergence to a tolerance $\epsilon \leq 1.5$ times EPSCUT is achieved.

2. SPECIFICATION OF ARRAY DIMENSIONS

Since array dimensions as currently set in QUANTO may not be sufficient for some types of problems, the arrays which have dimensions of interest to the user are listed in table 17 with the problem-dependent dimensions indicated as program variable names described in table 13. Only two of these program variables, MXTGT and MXWPNS, actually appear in the QUANTO code and must be set in the QUANTO main program (in a DATA statement). The others cross-reference the two tables and represent constants which may need to be changed (in the routines indicated) for problems which require more capacity than the constants of table 18 permit. In the execution of QUANTO, the maximum index of elements accessed in the arrays is controlled by program variables, so the constants specified in the DIMENSION and COMMON statements must be at least as large as (but may be larger than) the controlling program variables input through (or computed from inputs in) the data deck. Complete descriptions of those variables directly input may be found in the description of the input deck.

To conserve core, the user may want to reduce the array dimensions to the minimum required to run his problems. Furthermore, if only one type of aircraft occurs in his problems, the arrays FLAMB and FLAMBI may be equivalenced in the main program QUANTO.

### 3. SAMPLE QUANTO RUN CONSIDERATIONS

The listing of QUANTO in appendix I is the most current one as of the time this report was produced. For convenience, references to the lines of the program will use the alphanumeric identifiers on the right side of each line. Thus, the first line of the program is line QUA 10. This section will present the reader with the required considerations for several typical problems.

#### a. Considerations Concerning the Number of Aircraft Types

As the program is listed in appendix I, up to 25 bases, 168 weapon groups, one type of aircraft, etc. (in accordance with the figures of table 18) are permitted in the problem(s) to be run. Note that only one aircraft type is permitted since the arrays FLAMB and FLAMBI are equivalenced in line QUA 160. In view of this fact, core requirements could be reduced (when the problem has only one aircraft type) by changing the dimensions in lines QUA 30, QUA 50, QUA 70, and QUA 110 to the following, respectively:

```

1 SURV(35,168,1),FLAMBI(35,168,1),
3 VAL(35,1),PROD(35,1),VKILL(35,1),
5 RELVAL(1),BRTIME(1),PSI(1),ICAL(1),PKCIR(1),PKAN(1),
9 QPTS(30),QAREAL(30,1,4),QRLMX(30,1,4),DELTAC(2,1,1),JFLAC(2,1)

```

The user may quickly determine all of the occurrences of dimensions permitting two types of aircraft by searching table 17 for all occurrences of the dimension MXNTYP. In the QUANTO version in appendix I, MXNTYP has the value 2 wherever it occurs. Besides the above four lines, MXNTYP also occurs in the COMMON's labeled DISTIME, ACFTS, and NUCLER. Unfortunately, changing the dimensions in these labeled COMMON statements requires changing several lines since the COMMON statements appear in multiple routines, as listed in table 17. The user will probably want to weigh the inconveniences of changing QUANTO program statements against the core savings to be gained. For instance, the just described changes of lines QUA 30, QUA 50, QUA 70, and QUA 110 account for core savings of 11760, 105, 6, and 247 locations, respectively. The dimensions which appear in the QUANTO version in appendix I are intended to be adequate for most problems. If all these dimensions are adequate for the user's

problems, but the problem has two types of aircraft, the user need only delete the EQUIVALENCE statement (line QUA 160) for QUANTO to run the problem.

b. Changing the Numbers of Targets and/or Weapon Groups

By studying the first 16 lines of the QUANTO listing in appendix I, the user may determine that core requirements are most critically dependent on the numbers of targets, weapon groups, and types of aircraft. In particular, changes in lines QUA 20, QUA 30, QUA 150, and QUA 160 will usually bring the greatest payoff in core savings when the problems to be run have fewer than 35 targets, 168 weapon groups, and/or two types of aircraft. The considerations concerning the numbers of types of aircraft have already been discussed in paragraph a above.

When the problem has fewer than 35 targets, core savings will result from substituting the number of targets for 35 wherever 35 occurs in lines QUA 20, QUA 30, and QUA 150. In addition, if more than one type of aircraft is used, then the dimensions of the arrays VAL, PROD, and VKILL in line QUA 50 must be made consistent with the dimension changes in lines QUA 20, QUA 30, and QUA 150. Of course, the dimension MXTGT occurs elsewhere, as may be seen in table 17, but relatively small core savings result from changes elsewhere.

Likewise, the number of weapon groups (which is the sum of salvos from separately input submarine locations) may be reduced from 168 by changes in lines QUA 20, QUA 30, and QUA 150. These are all the lines in which the dimension MXWPNS (of table 18) occurs.

Changes in the dimensions must be consistent with other changes in the input deck. Dimensions must be at least as large as required by the input, or results will be unpredictable. A change in the number of targets will be reflected in the initial data card and the beddown data. A change in the number of weapon groups will be reflected in the initial data card, the number of submarine positions or salvos from those positions, and the initial allocation of SLBMs.

c. Output Options

The variable IOUT input in the initial data card indicates when an expanded output, sometimes useful for debugging, is desired. This output should never be requested for production runs and is only occasionally useful to the programmer analyzing program operations. Consequently, IOUT=1 should

always be input for production runs. Even then, voluminous output will be produced if beddown optimization is requested (i.e., IVOPT=2), and considerable output is produced by many other types of problems. Therefore, microfilm output is recommended until the volume of output is well known.

d. Distance to Centroid Computations

A side computation is required before the distances to the centroids may be estimated and input to QUANTO. The centroids of the areas of dispersing aircraft are dependent on the flyout profile and the geometry of turning. A rough approximation of the centroid location, for aircraft departing a single runway, may be obtained by computing (1) the location of an aircraft which turns 180 degrees following take-off at the time at which the aircraft reaches its terminal velocity, and (2) the location of that same aircraft after the same period of time, assuming it had not turned. The midpoint of these two locations would then be the approximate location of the centroid. A program is under development which will compute the centroid location giving equal weight to each possible aircraft location for turns of from 0 to 180 degrees in either direction. As the user might guess, considerable information about the flight characteristics of the aircraft is necessary for this computation. If multiple types of aircraft are used, the user must select one type as the one determining the centroid location.

Table 17

## ARRAYS OF QUANTO WITH PROBLEM-DEPENDENT DIMENSIONS

Array name and dimensions	Array contents (and routines which have constant dimensions for array in COMMON or DIMENSION statements)
A. <u>Arrays not in COMMON</u>	
1. ALOC (MXTGT, MXWPNS)	The $n_{ij}$ allocation matrix (in routine QUANTO).
2. FLAMB (MXTGT, MXWPNS)	The $\lambda_{ij}$ matrix, whose definition depends on the number of aircraft types (in QUANTO).
3. SURV (MXTGT, MXWPNS, MXNTYP)	The $S_{ij}$ survivability matrix for all aircraft types (in QUANTO).
4. FLAMBI (MXTGT, MXWPNS, MXNTYP)	The $\lambda_{ij}$ matrices for all aircraft types (in QUANTO).
5. TGTLAT (MXTGT)	The latitudes of the targets (in QUANTO).
6. TGT LNG (MXTGT)	The longitudes of the targets (in QUANTO).
7. DTCENT (MXTGT)	The distances to the centroids of the areas of dispersed aircraft leaving the targets (in QUANTO).
8. NRWAYS (MXTGT)	The number of runways at each target (in QUANTO).
9. NVALOC (MXTGT)	The number of weapons allocated to each target (in QUANTO).
10. VAL (MXTGT, MXNTYP)	The value of the aircraft of each type on each target* (in QUANTO).
11. PROD (MXTGT, MXNTYP)	The survivability products $\prod_{j=1}^{NWPNS} \left( S_{ij}^{n_{ij}} \right)$ for each target and each type aircraft (in QUANTO).
12. VKILL (MXTGT, MXNTYP)	The value of the aircraft of each type killed for each target (in QUANTO).
13. ISEQ (MXTGT, MXAC)	The take-off sequence on each base, as indicated by an ordered list of aircraft type numbers (in QUANTO).

Table 17 (cont'd)

Array name and dimensions	Array contents (and routines which have constant dimensions for array in COMMON or DIMENSION statements)
14. BRT (MXAC)	A working area used to compute the brake release times for each aircraft on a base (in QUANTO).
15. RELVAL (MXNTYP)	The point value assigned to a single aircraft of each type (in QUANTO).
16. BRTIME (MXNTYP)	The brake release times for each type aircraft, if that type takes off first from any base (in QUANTO).
17. PSI (MXNTYP)	The hardness of each aircraft type to overpressure; used to distinguish between kill and nonkill (in QUANTO).
18. ICAL (MXNTYP)	The hardness of each aircraft type to incident thermal energy; used to distinguish between kill and nonkill (in QUANTO).
19. PKCIR (MXNTYP)	A working area used to store the computed $P_k$ for each aircraft type when the aircraft are assumed to be uniformly distributed within a circle (in QUANTO).
20. KAN (MXNTYP)	A working area used to store the computed $P_k$ for each aircraft type when the aircraft are assumed to be uniformly distributed within an annulus (in QUANTO).
21. SUBLAT (MXSUB)	The latitudes of the submarine locations (in QUANTO).
22. SUBLNG (MXSUB)	The longitudes of the submarine locations (in QUANTO).
23. ISUBS (MXSUB)	The number of submarines at each submarine location (in QUANTO).
24. NMPS (MXSUB)	The number of missiles on each submarine at each submarine location (in QUANTO).
25. MTYPE (MXSUB)	The type of missile on each submarine at each submarine location (in QUANTO).
26. ITGTNO (MXM)	A working area used to store the target numbers to which the missiles from a single submarine are initially allocated (in QUANTO).

Table 17 (cont'd)

<u>Array name and dimensions</u>	<u>Array contents (and routines which have constant dimensions for array in COMMON or DIMENSION statements)</u>
27. DELTM (MXMTYP)	The launch interval of each missile type (in QUANTO).
28. RELML (MXMTYP)	The launch reliability of each missile type (in QUANTO).
29. RELMF (MXMTYP)	The in-flight reliability of each missile type (in QUANTO).
30. RELMWH (MXMTYP)	The warhead reliability of each missile type (in QUANTO).
31. RNGMAX (MXMTYP)	The maximum range of each missile type (in QUANTO).
32. RNGMIN (MXMTYP)	The minimum range of each missile type (in QUANTO).
33. YIELD (MXMTYP)	The yield of each missile type (in QUANTO).
34. FMTIME (MXTRAJ, MXMTYP)	The times that are input in describing the trajectories of each type missile (in QUANTO).
35. FMRNG (MXTRAJ, MXMTYP)	The ground ranges that are input in describing the trajectories of each type missile (in QUANTO).
36. MPROF (MXMTYP)	The number of time/distance pairs describing the trajectories of each missile type (in QUANTO).
37. QAREAL (30, MXNTYP, MXMTYP)	A table of lethal areas for 30 distances of the detonation from the centroid for each aircraft and missile combination (in QUANTO).
38. QRLMX (30, MXNTYP, MXMTYP)	A table of distances from detonation point, describing the farthest reach of the lethal area in a direction away from the centroid, for 30 distances of the detonation from the centroid for each aircraft and missile combination (in QUANTO).
39. DELTAC (MXRWAY, MXNTYP, MXNTYP)	The take-off intervals between each pair of aircraft types from bases with each possible number of runways (in QUANTO).
40. JFLAC (2, MXNTYP)	A working area used to store pointers to the first and last aircraft (of each type) departing a base (in QUANTO).
41. ALHOLD (MXTGT)	A working area used to integerize the weapon lay-down (in ALINT).

Table 17 (cont'd)

<u>Array name and dimensions</u>	<u>Array contents (and routines which have constant dimensions for array in COMMON or DIMENSION statements)</u>
42. VHOLD (MXTGT)	A working area used to integerize the beddown (in VINT).
43. SUMAC (5)	A working area used to compute the number of aircraft of each type on a single target for up to 5 aircraft types (in TGTKIL).
44. SUMACK (5)	A working area used to compute the number of aircraft killed of each type on a single target for up to 5 aircraft types (in TGTKIL).
45. LOHOLD (16) MXHOLD (16) LOTEMP (16) MXTEMP (16) LOH (16) MXH (16)	Working areas used in relocating submarines and assigning their SLBMs to new targets, for up to 16 missiles on a submarine (in SUBADJ).
B. <u>In COMMON named DISTIME</u> (arrays appear in routines QUANTO, PROCESS, DETAREA, BAKUP, TIMERAD, DATAGEN, and TIMEGEN with constant dimensions)	
1. FOTIME (MXPROF, MXNTYP) (also called S)	The times that are input and/or generated in describing the flight profile of each type aircraft.
2. FORNG (MXPROF, MXNTYP) (also called D)	The ground ranges that are input and/or generated in describing the flight profile of each type aircraft.
3. NPROF (MXNTYP) (also called NUMDATA)	The number of points of time and distance input and/or generated for the flight profile of each type aircraft.
4. CV (MXNTYP)	The final velocity reached by each aircraft type.
5. TI (MXNTYP)	The time interval used in generating profile points for each aircraft type.
C. <u>In COMMON named ACFTS</u> (arrays appear in routines PROCESS and DETAREA with constant dimensions)	
1. A (MXPIN, MXNTYP)	The altitude values input for each aircraft profile.
2. F (MXPIN, MXNTYP)	The ground range values input for each aircraft profile.
3. G (MXPIN, MXNTYP)	The time values input for each aircraft profile.

Table 17 (cont'd)

Array name and dimensions	Array contents (and routines which have constant dimensions for array in COMMON or DIMENSION statements)
4. VS (MXPIN, MXNTYP)	The velocities of sound for altitude values input for each aircraft profile.
5. VEL (MXPIN, MXNTYP)	The velocities input for each aircraft profile.
6. ACCEL (MXPIN, MXNTYP)	The level-off accelerations input for each aircraft profile.
7. NDATA (MXNTYP)	The number of input sets describing each aircraft profile.
D. <u>In COMMON named NUCLER</u> (arrays appear in routines PROCESS and DETAREA with constant dimensions)	
1. FLRP (MXNTYP, MXMTYP)	The computed lethal radii for overpressure for each missile type against each aircraft type.
2. FTSA (MXNTYP, MXMTYP)	The computed times of overpressure shock arrival for each missile type against each aircraft type.
3. FLRT (MXNTYP, MXMTYP)	The computed lethal thermal radii for each missile type against each aircraft type.
4. BURST (MXNTYP)	The heights of burst for each type aircraft.
5. DISMIN (MXNTYP)	The computed horizontal ground distances traveled by each type aircraft before it reaches its terminal altitude.
6. VPSI (MXNTYP)	The overpressure hardness of each type aircraft.
7. VCAL (MXNTYP)	The thermal hardness of each type aircraft.
8. VYIELD (MXMTYP)	The yield of each type missile.

Table 18  
PROBLEM-DEPENDENT DIMENSIONS IN QUANTO

<u>Variable dimension</u>	<u>Description</u>	<u>Current QUANTO dimension</u>
MXTGT	The maximum number of targets.	35
MXWPNS	The maximum number of weapon groups. A weapon group includes all potential missiles of the same type launched from the same point at the same time. The total number of weapon groups in a given problem may be computed easily by summing the numbers of missiles per submarine (NMPS of table 7) over all separately input potential submarine locations.	168
MXNTYP	The maximum number of types of aircraft.	2*
MXAC	The maximum number of aircraft on any base.	34
MXRWAY	The maximum number of runways at a base.	2
MXSUB	The maximum number of submarine locations.	28
MXM	The maximum number of missiles on a single submarine.	16
MXMTYP	The maximum number of missile types.	4
MXTRAJ	The maximum number of time/distance pairs input in describing missile trajectories.	15
MXPROF	The maximum number of time/distance pairs input and/or generated in describing aircraft flyout profiles.	99
MXPIN	The maximum number of input sets describing an aircraft profile.	50

---

\*Although two types of aircraft are permitted by the dimension statements in QUANTO, the "EQUIVALENCE (FLAMB, FLAMBI)" statement in QUANTO must be removed before running a problem having two types of aircraft.

## SECTION III

## TIPS ON THE UTILIZATION OF QUANTO

This section describes the flow of information to and from the program QUANTO and presents the user with the job control language (in SCOPE Version 3.2) required to execute QUANTO on the CDC 6600 computer system at the Air Force Weapons Laboratory (AFWL), Kirtland AFB, New Mexico. No attempt is made to describe the job control language functions in detail, but the reader who is familiar with some computer operating system should be able to generate similar operations by referring to his own job control language manuals and the discussion and examples presented here.

## 1. INFORMATION FLOW

The term QUANTO has been used to refer to the aggregation of all routines required to make optimization runs. The user deals with the program (making modifications, changing dimensions, etc.) in its source code form (i.e., the FORTRAN statements). This source code must be converted into machine-usable instructions called object code. If many sets of data are to be run independently, so that no run is dependent on the successful completion of a previous run, the source code should be compiled once into the object code, which is called QUANTOC, and recompilation avoided for consecutive runs. In figure 1 the information flow to and from this object code is shown.

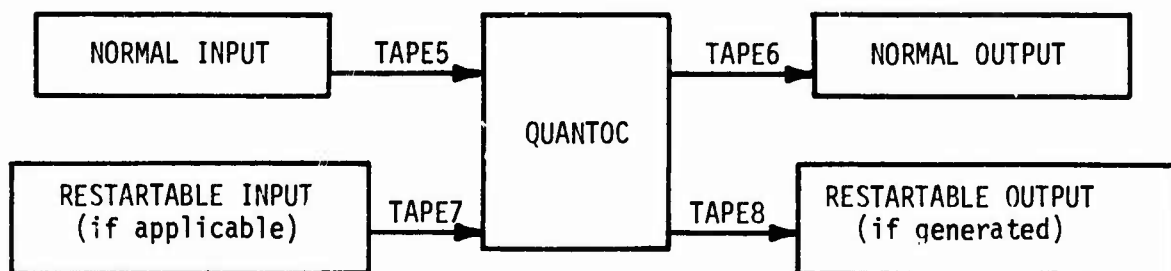


Figure 1. General Flow of Information

The identifiers TAPE5, TAPE6, TAPE7, and TAPE8 shown in figure 1 refer to input/output (I/O) devices which may be referenced by I/O unit numbers in the FORTRAN READ and WRITE statements in QUANTO. These four identifiers are listed in the first statement of QUANTO, the program header card, shown here:

```
PROGRAM QUANTO(INPUT,OUTPUT,TAPE7,TAPE8,TAPE5=INPUT,TAPE6=OUTPUT)
```

The "normal input" (figure 1) is data cards or "card images" from a data file. This card input is read from the TAPE5 input device. The amount of input required from this device depends on the MODE option to be used, as described in table 2.

"Normal output" (figure 1) is that produced by QUANTO during the processing of a job and is continually transmitted to an output buffer (disk storage), identified as TAPE6. This output is subsequently disposed to a printer or a microfilm output device.

"Restartable output" (figure 1) may be generated by certain jobs to permit a subsequent job to continue processing using the results of the first job as input. As described in table 2, all information necessary for the continuation of processing may be requested after the initial computations (survivabilities and expected kill) by inputting MODE=1 or may be obtained only on the condition that a job is within 30 seconds of its time limit (when MODE=2 or 3). The latter option prohibits jobs from terminating at the time limit without saving the results of the last computation cycle in a convenient form. The restartable output is transmitted to the output device indicated as TAPE8 and ultimately is saved on a physical magnetic tape. Jobs which create a restart tape may be continued by reading the created restart tape as input with a MODE=2 job. The restartable input (figure 1) is read from the magnetic tape input device identified as TAPE7.

The creation of restartable I/O has several purposes: (1) it prevents jobs from terminating at the time limit, due to an insufficient user time estimate, without saving the results of its computations; (2) it eliminates the necessity to reinitiate the problem with an increased time limit; and (3) it allows the user to analyze intermediate results to assure himself that the job is progressing properly before he risks a greater expenditure of computer time.

## 2. JOB CONTROL VARIATIONS

Job control language provides a means to exercise control over the manner in which a particular job is processed by a computer system. Information sources are directly (or indirectly by default) specified through control cards. The following examples offer various schemes for using the QUANTO program on the AFWL computer system. Bracketed upper-case Latin letters are used in line with the control cards to indicate information in punched cards, e.g., the deck containing the QUANTO source code. Each example is followed by a brief description of the overall process.

## a. Example I

```

JOBID,P2, T1000,CM240000.
TASK(MILLER,00000000-OXX,ORGANIZATION,TELEPHONE)
REQUEST TAPE7. XX000 RING OUT
REWIND(TAPE7)
REQUEST TAPE8. XX001 RING IN
REWIND(TAPE8)
RUN(A,,,INPUT,OUTPUT,QUANTOC,377777)
PRESET.
QUANTOC(LC=377777,INPUT,OUTPUT,TAPE7,TAPE8)
REWIND(TAPE7)
RETURN(TAPE7)
REWIND(TAPE8)
RETURN(TAPE8)
EXIT.
DMP(240000)
7/8/9
[QUANTO SOURCE CODE PUNCHED CARD DECK]
7/8/9
[DATA PUNCHED CARD DECK]
6/7/8/9

```

Example I illustrates job control language which runs QUANTO in batch processing on the AFWL computer system. In the job card a core requirement of 240,000 (octal) words and a time limit of 1000 (octal) seconds are specified. These limits will be sufficient for most individual problems run by QUANTO. In particular, the core is sufficient for the array sizes indicated in table 18. As each tape request card is processed, a flashing message appears on the computer operator's console. The operator then mounts the tapes identified by XX000 and XX001. This example assumes the presence of restartable input (TAPE7) and the possibility of generating restartable output (TAPE8) which is to be saved. If either of these conditions is absent, all references to the corresponding tape should be removed from the job control language.

The RUN card requests compilation of the QUANTO source code into the object code file called QUANTOC. The LC=377777 in the line requesting execution of the QUANTOC code removes the line count limit on output. This is frequently required since QUANTO currently generates considerable output for many types of problems and values of the input parameters. A compiled listing of the program statements and the program would appear on paper using the job control language of example I. The DMP card requests a dump of the entire core segment if an abnormal termination of the job occurs. The notations "7/8/9" and "6/7/8/9" indicate cards having multiple numeric punches in column one which translate to "end of record" and "end of file," respectively, on magnetic tape.

A major problem with the previous example is that it ties up two tape-drive units for the duration of the program run. Example II affords a means of transferring information to and from QUANTO via disk storage, again assuming that both tapes (TAPE7, TAPE8) are necessary and contain information.

b. Example II

```

JOBID,P2,T1000,CM60000.
TASK(NAME,00000000-OXX,ORGANIZATION,TELEPHONE)
RFL(10000) REDUCE TO 10K FOR TAPE HANDLING
REQUEST TAPE7. XX000 RING OUT
REWIND(TAPE7)
COPY(TAPE7,DISK7)
REWIND(TAPE7)
RETURN(TAPE7)
RFL(100000) FOR COMPILE
RUN(A,,,INPUT,OUTPUT,QUANTOC,377777)
RFL(240000) FOR LOAD
PRESET.
REWIND(DISK7)
QUANTOC(LC=377777,INPUT,FILMPR,DISK7,DISK8)
RFL(10000)
REQUEST TAPE8. XX000 RING IN
REWIND(TAPE8)
REWIND(DISK8)
COPY(DISK8,TAPE8)
REWIND(TAPE8)
RETURN(TAPE8)
EXIT.
DMP(240000)
7/8/9
[QUANTO SOURCE CODE PUNCHED CARD DECK]
7/8/9
[DATA PUNCHED CARD DECK]
6/7/8/9

```

Note that the job card requests 60,000 (octal) words of core memory to conform with the AFWL priority system but that the needed core is adjusted throughout the job control language by RFL cards. The output from the execution of QUANTOC is placed on microfilm due to the FILMPR entry on the QUANTOC execute card.

A major drawback of the first two examples is the punched card handling, since the QUANTO source deck contains approximately 4500 cards. A method for handling large jobs is available under the AFWL computer system and is shown in the next example.

c. Example III

```

JOBID,P2,T1000,CM60000.
TASK(NAME,00000000-OXX,ORG,TEL)
RFL(10000) REDUCE TO 10K FOR TAPE HANDLING
REQUEST TAPE7. XX000 RING OUT
REWIND(TAPE7)
COPY(TAPE7,DISK7)
REWIND(TAPE7)
RETURN(TAPE7)
COMMON(SABCEM)
RFL(40000) FOR UPDATE
UPDATE(Q,P=SABCEM,D,L=A1)
RETURN(SABCEM)
RFL(100000) FOR COMPILE
RUN(A,,,COMPILE,OUTPUT,QUANTOC,377777)
RFL(240000) FOR LOAD
PRESET.
REWIND(DISK7)
QUANTOC(LC=377777,INPUT,FILMPR,DISK7,DISK8)
RFL(10000)
REQUEST TAPE8. XX001 RING IN
REWIND(TAPE8)
REWIND(DISK8)
COPY(DISK8,TAPE8)
REWIND(TAPE8)
RETURN(TAPE8)
7/8/9
*COMPILE QUANTOT
7/8/9
[DATA PUNCHED CARD DECK]
6/7/8/9

```

For this example, the QUANTO source code is assumed to be previously built as an update file called QUANTOT and placed in the COMMON file, SABCEM. The AFWL update system could create temporary changes to the QUANTOT file if these changes appeared after the \*COMPILE QUANTOT card; for instance, array sizes could be changed by replacing the appropriate DIMENSION statements. Such

changes, if any, are made when the UPDATE card is encountered in the job control language. The resultant updated program, in source code form, is placed in a file named COMPILE by the UPDATE step. Thus, COMPILE appears in the RUN card instead of the usual INPUT.

For production purposes the object code QUANTOC could be saved on disk as shown in example IV. Then multiple executions of QUANTOC could be requested for multiple sets of data without requiring multiple compilations of the source code.

d. Example IV

```
JOBID,P5,T177,CM40000.
TASK(NAME,00000000-OXX,ORGANIZATION,TELEPHONE)
COMMON(SABCEM)
UPDATE(Q,P=SABCEM,L=A1,D)
RETURN(SABCEM)
RUN(A,,COMPILE,OUTPUT,QUANTOC,377777)
COMMON(QUANTOC)
RETURN(QUANTOC)
7/8/9
*COMPILE QUANTOC
7/8/9
6/7/8/9
```

Obviously many variations of job control language are possible, but the preceding examples should suffice to demonstrate the basic means of running QUANTO.

## SECTION IV

## SUMMARY

This report has described the input format, array dimensions, and job control language of concern to the user of the program QUANTO. It should be expected that occasional problems with the program will occur as the user attempts runs with new combinations of parameter values. Such difficulties should be brought to the attention of the authors, who will advise and instruct the user of QUANTO, or the Air Force Weapons Laboratory. As QUANTO is used, modifications and improvements are inevitable. The most current version of QUANTO and the documentation may be obtained from the Air Force Weapons Laboratory.

The mathematical model used by QUANTO would be appropriate for applications other than SLBM attacks on flushing aircraft, with slight modifications. For instance, if the survivability of any target from any weapon can be quantified, a weapon allocation may be obtained from the optimization model. A complete description of the mathematical model and its assumptions may be found in AFWL Technical Report AFWL-TR-73-242.

APPENDIX I

CURRENT LISTING OF QUANTO SOURCE CODE

```

PROGRAM QUANTO (INPUT,OUTPUT,TAPE7,TAPE6,TAPE5=INPUT,TAPE6=OUTPUT) QUA 10
DIMENSION ALOC(35,168), FLAMB(35,168),
1 SURV(35,168,2), FLAMBI(35,168,2), QUA 30
2 TGLAT(35), TGLNG(35), DTCENT(35), NRWAYS(35), NHALOC(35), QUA 40
3 VAL(35,2), PROD(35,2), VKILL(35,2), QUA 50
4 ISEQ(35,34), BRT(34), QUA 60
5 RELVAL(2), BRTIME(2), PSI(2), ICAL(2), PKCIR(2), PKAN(2), QUA 70
6 SUBLAT(28), SUBLNG(28), ISUBS(28), NMPS(28), MTYPE(28), ITGTNO(10), QUA 80
7 DELTH(4), RELML(4), RELMF(4), RELMWH(4), RNGMAX(4), RNGMIN(4), QUA 90
8 YIELD(4), FMTIME(15,4), FHRNG(15,4), NPROF(4), QUA 100
9 QPTS(30), QAREAL(30,2,4), QRLMX(3,2,4), DELTAC(2,2,2), JFLAC(2,2) QUA 110
COMMON /DISTIME/ FDTIME(99,2), FORNG(99,2), NPROF(2), CV(2), TI(2), JTYP QUA 120
1PE, NTPES, MISTYP, MYPES, RLMAX QUA 130
DATA RDOPEG, VERYHI, PI/.0174532925, 1.0E+300, 3.14159265/ QUA 140
DATA MXTGT, MXWPNS/35,168/ QUA 150
EQUIVALENCE (FLAMB, FLAMBI) QUA 160
C IOUT=1 SUPPRESSES INTERMEDIATE MULTIPLIER AND ALLOCATION OUTPUT. QUA 170
C OTHERWISE SET IOUT=2. QUA 180
C IF ISOFT=1, SUB LOCATIONS WILL NOT BE OPTIMIZED AMONG INPUT SUB QUA 190
C LOCATIONS. QUA 200
C IF ISOPT=2, SUB LOCATIONS WILL BE OPTIMIZED. QUA 210
C IF IVOPT = 1, BEDDOWN WILL NOT BE OPTIMIZED. QUA 220
C IF IVOPT = 2, BEDDOWN WILL BE OPTIMIZED. QUA 230
19 READ 890, NTGTS, NSUBS, NTPES, MXRWAY, MYPES, IOUT, ISOPT, IVOPT, NCASE, QUA 240
1MODE QUA 250
IF (ENDFILE 5) 800,20 QUA 260
20 CALL DATE (IDATE) QUA 270
CALL CLOCK (ITIME) QUA 280
CALL PAGE (0) QUA 290
PRINT 900, NCASE, MODE, IDATE, ITIME QUA 300
PRINT 910, NTGTS, NSUBS, NTPES, MYPES QUA 310
CALL PAGE (2) QUA 320
IF (MODE.NE.2) GO TO 30 QUA 330
READ (7) NWPNS, NCOL, ITER, ITCUT1, ITCUT2, MOVES, MOVEV, MOVEST, EPS, CPSC QUA 340
1UT, OBHOLD, DELOBJ, SUBOBJ, CHGKIL QUA 350
READ (7) ((SURV(I,J,K), I=1, NTGTS), J=1, NWPNS), K=1, NTPES) QUA 360
READ (7) ((ALOC(I,J), I=1, NTGTS), J=1, NWPNS) QUA 370
READ (7) ((VAL(I,J), I=1, NTGTS), J=1, NTPES), (RELVAL(I), I=1, NTPES), QUA 380
1 (ISUBS(I), NMPS(I), MTYPE(I), I=1, NSUBS) QUA 390
PRINT 920 QUA 400
CALL PAGE (1) QUA 410
GO TO 610 QUA 420
30 DO 50 I=1, NTGTS QUA 430
READ 930, (TGLAT(I), TGLNG(I), DTCENT(I), NRWAYS(I), (VAL(I, JTYPE), J QUA 440
1TYPE=1, NTPES)) QUA 450
NAC=0 QUA 460
DO 40 JTYPE=1, NTPES QUA 470
40 NAC=NAC+VAL(I, JTYPE)+.01 QUA 480
50 READ 940, (ISEQ(I, IAC), IAC=1, NAC) QUA 490
READ 950, (RELVAL(JTYPE), BRTIME(JTYPE), PSI(JTYPE), ICAL(JTYPE), JTYP QUA 500
1E=1, NTPES) QUA 510

```

```

READ 960, (((DELTAC(IRW,JTYPE,KTYPE),KTYPE=1,NTYPES),JTYPE=1,NTYPEQUA 520
1S),IRW=1,MXRWAY) QUA 530
PRINT 970 QUA 540
CALL PAGE (3) QUA 550
DO 70 I=1,NTGTS QUA 560
NAC=0 QUA 570
DO 60 JTYPE=1,NTYPES QUA 580
60 NAC=NAC+VAL(I,JTYPE)+.01 QUA 590
FNAC=NAC QUA 600
NLINES=(FNAC-1.)/30. QUA 610
CALL PAGE (-MAX0(2,NTYPES)-NLINES-2) QUA 620
PRINT 980, I,TGTLAT(I),TGTLNG(I),NRWAYS(I),DTCENT(I),(JTYPE,VAL(I,QUA 630
1JTYPE),JTYPE=1,NTYPES) QUA 640
70 PRINT 990, (ISEQ(I,IAC),IAC=1,NAC) QUA 650
CALL PAGE (-NTYPES-3) QUA 660
PRINT 1000, (JTYPE,RELVAL(JTYPE),BRTIME(JTYPE),PSI(JTYPE),ICAL(JTYQUA 670
1PE),JTYPE=1,NTYPES) QUA 680
CALL PAGE (-MXRWAY*NTYPES*NTYPES-3) QUA 690
PRINT 1010, (((IRW,JTYPE,KTYPE,DELTAC(IRW,JTYPE,KTYPE),KTYPE=1,NTYQUA 700
1PES),JTYPE=1,NTYPES),IRW=1,MXRWAY) QUA 710
NWPNS=0 QUA 720
NCOL=0 QUA 730
DO 80 I=1,NSUBS QUA 740
READ 1020, SUBLAT(I),SUBLNG(I),ISUBS(I),NMPS(I),MTYPE(I) QUA 750
NWPNS=NWPNS+NMPS(I) QUA 760
80 IF (ISUBS(I).NE.0) NCOL=NCOL+NMPS(I) QUA 770
CALL PAGE (-NSUBS-2) QUA 780
PRINT 1030, (I,SUBLAT(I),SUBLNG(I),ISUBS(I),NMPS(I),MTYPE(I),I=1,NQUA 790
1SUBS) QUA 800
CALL PAGE (-MTYPES*2) QUA 810
DO 100 J=1,MTYPES QUA 820
NSUBT=0 QUA 830
DO 90 I=1,NSUBS QUA 840
90 IF (MTYPE(I).EQ.J) NSUBT=NSUBT+ISUBS(I) QUA 850
100 PRINT 1040, J,NSUBT QUA 860
DO 110 I=1,MTYPES QUA 870
READ 960, DELTM(I),RELML(I),RELMF(I),RELMWH(I),RNGMIN(I),RNGMAX(I)QUA 880
1,YIELD(I) QUA 890
READ 1050, MPR,(FMTIME(J,I),FMRNG(J,I),J=1,MPR) QUA 900
110 MPROF(I)=MPR QUA 910
CALL PAGE (-MTYPES-3) QUA 920
PRINT 1060, (JTYPE,DELTM(JTYPE),RELML(JTYPE),RELMF(JTYPE),RELMWH(JQUA 930
1TYPE),RNGMIN(JTYPE),RNGMAX(JTYPE),YIELD(JTYPE),JTYPE=1,MTYPES) QUA 940
DO 120 JTYPE=1,MTYPES QUA 950
MPR=MPROF(JTYPE) QUA 960
CALL PAGE (-MPR-2) QUA 970
120 PRINT 1070, JTYPE,(FMTIME(I,JTYPE),FMRNG(I,JTYPE),I=1,MPR) QUA 980
CALL PROCESS (PSI,ICAL,YIELD,NCASE,MODE) QUA 990
IF (MODE.EQ.0) GO TO 10 QUA1000
C QUA1010
C QUA1020

```

C	COMPUTE SURVIVABILITIES.	QUA1030
C	CONVERT LOCATION DEGREES TO RADIANS.	QUA1040
	DO 130 I=1,NTGTS	QUA1050
	TGTLAT(I)=TGTLAT(I)*RDPDEG	QUA1060
130	TGTLNG(I)=TGTLNG(I)*RDPDEG	QUA1070
	DO 140 I=1,NSUBS	QUA1080
	SUBLAT(I)=SUBLAT(I)*RDPDEG	QUA1090
140	SUBLNG(I)=SUBLNG(I)*RDPDEG	QUA1100
C	SET-UP FOR BUILDING TABLES OF LETHAL AREAS.	QUA1110
	DO 150 IQ=1,30	QUA1120
	IF (IQ.LE.11) QPTS(IQ)=IQ-1.	QUA1130
150	IF (IQ.GT.11) QPTS(IQ)=(IQ-9.)*5.	QUA1140
	DTCOLD=-1.	QUA1150
	DO 530 I=1,NTGTS	QUA1160
	CPKTIM=VERYHI	QUA1170
	DTCNEW=DTCENT(I)	QUA1180
	IF (DTCNEW.EQ.DTCOLD) GO TO 180	QUA1190
	DTCOLD=DTCNEW	QUA1200
C	GENERATE TABLE OF LETHAL AREAS FOR NEW DISTANCE TO CENTROID.	QUA1210
	DO 160 IQ=1,30	QUA1220
	DO 160 JTYPE=1,NTYPES	QUA1230
	DO 160 MISTYP=1,MTYPES	QUA1240
	QAREAL(IQ,JTYPE,MISTYP)=DETAREA(QPTS(IQ),DTCNEW)	QUA1250
160	QRLMX(IQ,JTYPE,MISTYP)=RLMAX	QUA1260
	CALL PAGE (0)	QUA1270
	PRINT 1080, DTCNEW	QUA1280
	CALL PAGE (1)	QUA1290
	DO 170 JTYPE=1,NTYPES	QUA1300
	DO 170 MISTYP=1,MTYPES	QUA1320
	CALL PAGE(-33)	
170	PRINT 1030, JTYPE,MISTYP,(QPTS(IQ),QAREAL(IQ,JTYPE,MISTYP),QRLMX(IQ,QUA1330	
	1Q,JTYPE,MISTYP),IQ=1,30)	QUA1340
	CALL PAGE (0)	QUA1350
180	NAC=0	QUA1360
	DO 190 JTYPE=1,NTYPES	QUA1370
190	NAC=NAC+VAL(I,JTYPE)+.01	QUA1380
C	COMPUTE BRAKE RELEASE TIMES OF EACH AIRCRAFT.	QUA1390
	ITYPE=ISEQ(I,1)	QUA1400
	BRT(1)=BRTIME(ITYPE)	QUA1410
	IF (NAC.LT.2) GO TO 210	QUA1420
	DO 200 IAC=2,NAC	QUA1430
	NWY=NRWAYS(I)	QUA1440
	ITYP1=ISEQ(I,IAC-1)	QUA1450
	ITYPE=ISEQ(I,IAC)	QUA1460
200	BRT(IAC)=BRT(IAC-1)+DELTAC(NWY,ITYP1,ITYPE)	QUA1470
	CALL PAGE (-NAC-2)	QUA1480
210	PRINT 1100, I,(ISEQ(I,IAC),BRT(IAC),IAC=1,NAC)	QUA1490
C	LOCATE FIRST AND LAST AIRCRAFT OF EACH TYPE.	QUA1500
	DO 220 IAC=1,2	QUA1510
	DO 220 JTYPE=1,NTYPES	QUA1520
220	JFLAC(IAC,JTYPE)=0	QUA1530

	DO 270 JTYPE=1,NTYPES	QUA1540
	DO 230 IAC=1,NAC	QUA1550
	JAC=IAC	QUA1560
	IF (ISEQ(I,JAC).EQ.JTYPE) GO TO 240	QUA1570
230	CONTINUE	QUA1580
C	NO AIRCRAFT OF THIS TYPE FOUND.	QUA1590
	GO TO 270	QUA1600
240	JFLAC(1,JTYPE)=JAC	QUA1610
	DO 250 IAC=1,NAC	QUA1620
	JAC=NAC+1-IAC	QUA1630
	IF (ISEQ(I,JAC).EQ.JTYPE) GO TO 260	QUA1640
250	CONTINUE	QUA1650
C	WE WILL BE ABLE TO FIND ONE OF THIS TYPE IF WE GET TO THIS POINT,	QUA1660
C	BUT ANYWAY...	QUA1670
	GO TO 270	QUA1680
260	JFLAC(2,JTYPE)=JAC	QUA1690
270	CONTINUE	QUA1700
C	COMPUTE TIMES OF FLIGHT AND ARRIVALS IN MINUTES. IF DISTANCE IS	QUA1710
C	OUT OF RANGE, MAKE FLIGHT TIME VERY LARGE.	QUA1720
	K=0	QUA1730
	DO 520 KSUB=1,NSUBS	QUA1740
	DISTT=DIST(TGTLAT(I),TGTLNG(I),SUBLAT(KSUB),SUBLNG(KSUB))	QUA1750
	MISTYP=MTYPE(KSUB)	QUA1760
	RELH=RELML(MISTYP)*RELMF(MISTYP)*RELMHH(MISTYP)	QUA1770
	IF (DISTT.LT.RNGMIN(MISTYP).OR.DISTT.GT.RNGMAX(MISTYP)) GO TO 280	QUA1780
	TFLT=ALAG(DISTT,FMRNG(1,MISTYP),FMTIME(1,MISTYP),MPROF(MISTYP))	QUA1790
	GO TO 290	QUA1800
280	TFLT=VERYHI	QUA1810
290	PRINT 1110, I, KSUB, DISTT, MISTYP, TFLT	QUA1820
	CALL PAGE (2)	QUA1830
	NMIS=NMPS(KSUB)	QUA1840
	DO 520 KSAL=1, NMIS	QUA1850
	K=K+1	QUA1860
	TOAT=TFLT+(KSAL-1)*DELTM(MISTYP)	QUA1870
	IF (TOAT.LT.VERYHI) GO TO 310	QUA1880
	DO 300 JTYPE=1,NTYPES	QUA1890
300	SURV(I,K,JTYPE)=1.	QUA1900
	GO TO 520	QUA1910
310	RFC1=0.	QUA1920
	RFC1=9999.	QUA1930
C	COMPUTE RANGE OF FIRST AIRCRAFT AND LAST IF PK SWITCH INDICATES	QUA1940
C	ANNULAR PK COMPUTATION MAY STILL BE NECESSARY. (TOAT.LE.CPKTIM)	QUA1950
	DO 320 JTYPE=1,NTYPES	QUA1960
	JAC=JFLAC(1,JTYPE)	QUA1970
	IF (JAC.EQ.0) GO TO 320	QUA1980
	RTOA=TOAT-BRT(JAC)	QUA1990
	RFC=PROFLU(RTOA,FOTIME(1,JTYPE),FORNG(1,JTYPE),NPROF(JTYPE))-DTCEN	QUA2000
	1T(I)	QUA2010
	IF (RFC.GT.RFC1) RFC1=RFC	QUA2020
C	COMPUTE ONLY RADIUS OF FIRST AIRCRAFT IF SWITCH FOR CIRCULAR PK	QUA2030
C	COMPUTATION IS SET.	QUA2040

	IF (TOAT.GE.CPKTIM) GO TO 320	QUA2050
	JAC=JFLAC(2,JTYPE)	QUA2060
	RTDA=TOAT-BRT(JAC)	QUA2070
	RFC=PROFLU(RTOA,FOTIME(1,JTYPE),FORNG(1,JTYPE),NPROF(JTYPE))-DTGENQUA2080	
	1T(I)	QUA2090
	IF (RFC.LT.RFCL) RFCL=RFC	QUA2100
320	CONTINUE	QUA2110
	IF (RFCL.GE.RFC1) RFCL=RFC1-.01	QUA2120
	IF (RFCL.LE.0.) RFCL=0.	QUA2130
	IF (KSAL.NE.1) GO TO 330	QUA2140
	PRINT 1120, K,I,RFCL	QUA2150
	CALL PAGE (1)	QUA2160
330	RFCLSQ=RFC1*RFCL	QUA2170
	RFCLSQ=RFCL*RFCL	QUA2180
	ISJM=0	QUA2190
	IF (TOAT.GE.CPKTIM) GO TO 400	QUA2200
C	COMPUTE LARGEST LETHAL RADIUS FOR CENTERED WEAPON.	QUA2210
	CELRMX=0.	QUA2220
	DO 340 JTYPE=1,NTYPES	QUA2230
	CELR=SQRT(QAREAL(1,JTYPE,MISTYP)/PI)	QUA2240
340	IF (CELR.GT.CELRMX) CELRMX=CELR	QUA2250
	IF (RFC1.GT.CELRMX) GO TO 380	QUA2260
	IF (RFC1.GT.J.) GO TO 360	QUA2270
	DO 350 JTYPE=1,NTYPES	QUA2280
350	SURV(I,K,JTYPE)=1.-RELM	QUA2290
	GO TO 520	QUA2300
360	DO 370 JTYPE=1,NTYPES	QUA2310
	PK=(QAREAL(1,JTYPE,MISTYP)-PI*RFCLSQ)/(PI*(RFC1SQ-RFCLSQ))	QUA2320
	IF (PK.GT.1.) PK=1.	QUA2330
	IF (PK.LT.0.) PK=0.	
370	SURV(I,K,JTYPE)=1.-PK*RELM	QUA2340
	GO TO 520	QUA2350
C	COMPUTE FARTHEST REACH OF LETHAL REGION FOR WEAPON PLACED AT	QUA2360
C	RFC1/SQRT(2) TO SEE IF IT PROTRUDES BEYOND RFC1.	QUA2370
380	RLMX=0.	QUA2380
	DO 390 JTYPE=1,NTYPES	QUA2390
	RL=ALAG(RFC1/1.414213562,QPTS,QRLMX(1,JTYPE,MISTYP),30)	QUA2400
390	IF (RL.GT.RLMX) RLMX=RL	QUA2410
	IF (RFC1/1.414213562+RLMX.GT.RFC1) GO TO 560	QUA2420
	ISUM=ISUM+1	QUA2430
C	COMPUTE CIRCULAR PKS.	QUA2440
400	DO 410 JTYPE=1,NTYPES	QUA2450
	AREAL=ALAG(RFC1/1.414213562,QPTS,QAREAL(1,JTYPE,MISTYP),30)	QUA2460
	IF (AREAL.LT.0.) AREAL=0.	QUA2470
410	PKCIR(JTYPE)=AREAL/(PI*RFC1SQ)	QUA2480
	IF (TOAT.GE.CPKTIM) GO TO 480	QUA2490
C	COMPUTE ANNULAR PKS UNTIL A LARGER ONE IS FOUND.	QUA2500
420	DO 450 JTYPE=1,NTYPES	QUA2510
	AREAL=ALAG((RFC1+RFCL)/2.,QPTS,QAREAL(1,JTYPE,MISTYP),30)	QUA2520
	IF (AREAL.LT.0.) AREAL=0.	QUA2530
	ELR=SQRT(AREAL/PI)	QUA2540

	IF (ELR.LT..0001) GO TO 430	QUA255J
	PKAN(JTYPE)=XAREA(ELR, (RFC1+RFCL)/2., RFCL, RFC1) / (PI*(RFC1SQ-RFCLSQ	QUA256J
	1))	QUA257J
	GO TO 440	QUA258J
430	PKAN(JTYPE)=0.	QUA259J
44J	PRINT 1130, PKAN(JTYPE), PKCIR(JTYPE), K, JTYPE, RFC1, RFCL	QUA260J
	IF (PKAN(JTYPE).LT.0.) STOP	QUA261J
	CALL PAGE (1)	QUA262J
	IF (PKCIR(JTYPE).LE.PKAN(JTYPE)) GO TO 470	QUA263J
45J	CONTINUE	QUA264J
C	USE ANNULAR PKs.	QUA265J
	DO 450 JTYPE=1, NTYPES	QUA266J
460	SURV(I, K, JTYPE)=1.-PKAN(JTYPE)*RELM	QUA267J
	GO TO 520	QUA268J
470	ISUM=ISUM+1	QUA269J
48J	DO 490 JTYPE=1, NTYPES	QUA270J
	PK=AMIN1(1.0, PKCIR(JTYPE))	QUA271J
49J	SURV(I, K, JTYPE)=1.-PK*RELM	QUA272J
	IF (ISUM.EQ.2.AND.TOAT.LT.CPKTIM) CPKTIM=TOAT	QUA273J
	GO TO 520	QUA274J
50J	DO 510 JTYPE=1, NTYPES	QUA275J
	QPOS=AMAX1(0., RFC1-RLMX)	QUA276J
	AREAL=ALAG(QPOS, QPTS, QAREAL(1, JTYPE, MISTYP), 3J)	QUA277J
	IF (AREAL.LT.0.) AREAL=0.	QUA278J
51J	PKCIR(JTYPE)=AREAL/(PI*RFC1SQ)	QUA279J
	GO TO 42J	QUA280J
52J	CONTINUE	QUA281J
53J	CONTINUE	QUA282J
	DO 540 JTYPE=1, NTYPES	QUA283J
	CALL PAGE (0)	QUA284J
	CALL PAGE (2)	QUA285J
	PRINT 1140, JTYPE	QUA286J
	DO 540 I=1, NTGTS	QUA287J
	CALL PAGE (-NSUBS-2)	QUA288J
	PRINT 1150, I	QUA289J
	JLJ=1	QUA290J
	DO 540 JJ=1, NSUBS	QUA291J
	JHI=JLO+NMPS(JJ)-1	QUA292J
	PRINT 1160, (SURV(I, J, JTYPE), J=JLO, JHI)	QUA293J
54J	JLO=JHI+1	QUA294J
	SUBOBJ=0.	QUA295J
	MOVES=0	QUA296J
	MOVEST=0	QUA297J
	MOVEV=0	QUA298J
	READ 1170, CHGKIL, ITCUT1, ITCUT2, EPS CUT	QUA299J
	EPS=.1	QUA300J
	PRINT 1180, EPS	QUA301J
	ITER=0	QUA302J
	CALL PAGE (3)	QUA303J
C	COMPUTE VALUES OF AIRCRAFT.	QUA304J
	DO 550 I=1, NTGTS	QUA305J

	DO 550 JTYPE=1,NTYPES	QUA3360
550	VAL(I,JTYPE)=VAL(I,JTYPE)*RELVAL(JTYPE)	QUA3370
	DO 560 I=1,NTGTS	QUA3380
	DO 560 J=1,NWPNS	QUA3390
560	ALOC(I,J)=0.	QUA3400
C	INPUT FIRST ALLOCATION.	QUA3410
	MWPN=0	QUA3420
	DO 600 J=1,NSUBS	QUA3430
	JLIM=ISUBS(J)	QUA3440
	ILIM=NMPS(J)	QUA3450
	IF (JLIM.EQ.0) GO TO 590	QUA3460
	DO 580 JSUB=1,JLIM	QUA3470
	READ 890, (ITGTNO(I),I=1,ILIM)	QUA3480
	DO 570 I=1,ILIM	QUA3490
	MWPN=MWPN+1	QUA3500
	II=ITGTNO(I)	QUA3510
	IF (II.EQ.0) GO TO 570	QUA3520
	ALOC(II,MWPN)=ALOC(II,MWPN)+1.0	QUA3530
570	CONTINUE	QUA3540
	IF (JSUB.LT.JLIM) MWPN=MWPN-ILIM	QUA3550
580	CONTINUE	QUA3560
	GO TO 600	QUA3570
590	MWPN=MWPN+ILIM	QUA3580
600	CONTINUE	QUA3590
C		QUA3600
C		QUA3610
C	LOOP ON ALLOCATIONS FOLLOWS.	QUA3620
610	IBROUT=0	QUA3630
	CALL TGTKIL (NTGTS,NWPNS,NTYPES,OBJSUM,ALOC,FLAMB,FLAMBI,SURV,VAL,	QUA3640
	1PRD,VKILL,NWALOC,RELVAL,MXTGT,MXWPN)	QUA3650
	PRINT 1190, OBJSUM	QUA3660
	CALL PAGE (2)	QUA3670
	IF (MODE.EQ.1) GO TO 620	QUA3680
	CALL TINTGO (TLEFT)	QUA3690
	IF (TLEFT.GT.30.) GO TO 630	QUA3700
620	WRITE (8) NWPNS,NCOL,ITER,ITCUT1,ITCUT2,MOVES,MOVEV,MOVESL,EPS,EPS	QUA3710
	1CUT,OBHOLD,DELOBJ,SUBOBJ,CHGKIL	QUA3720
	WRITE (8) (((SURV(I,J,K),I=1,NTGTS),J=1,NWPNS),K=1,NTYPES)	QUA3730
	WRITE (8) ((ALOC(I,J),I=1,NTGTS),J=1,NWPNS)	QUA3740
	WRITE (8) ((VAL(I,J),I=1,NTGTS),J=1,NTYPES),(RELVAL(I),I=1,NTYPES)	QUA3750
	1,(ISUBS(I),NMPS(I),MTYPE(I),I=1,NSUBS)	QUA3760
	PRINT 1200	QUA3770
	CALL PAGE (1)	QUA3780
	IF (MODE.EQ.1) GO TO 10	QUA3790
	STOP	QUA3800
630	OBHOLD=0.	QUA3810
C	MAX NUMBER OF ITERATIONS IS THE PRODUCT OF UPPER LIMITS OF	QUA3820
C	ITER1 AND ITER2.	QUA3830
	IF (ITCUT1.EQ.0) GO TO 730	QUA3840
	DO 720 ITER1=1,ITCUT1	QUA3850
	DO 700 ITER2=1,ITCUT2	QUA3860

	ITER=ITER+1	QUA3570
	CALL ADJLAM (NTGTS, NSUBS, NTYPES, NCOL, IBROUT, MLOW, MHI, JCOL, DELTA, EPS,	QUA3580
	IS, ALOC, FLAMB, FLAMBI, SURV, NHALOC, ISUBS, NMPS, MXTGT, MXWPNS)	QUA3590
	IF (IBROUT.NE.1) GO TO 660	QUA3600
	IF (EPS.GT.1.5*EPSCUT) GO TO 640	QUA3610
	PRINT 1210, EPSCUT	QUA3620
	CALL PAGE (2)	QUA3630
	GO TO 730	QUA3640
640	EPS=EPS/10.	QUA3650
	PRINT 1180, EPS	QUA3660
	CALL PAGE (3)	QUA3670
	IBROUT=0	QUA3680
	IF (IOUT.EQ.1) GO TO 700	QUA3690
C	DEBUG OUTPUT.	QUA3700
	CALL ALROUT (ITER, NTGTS, NSUBS, NWPNS, ALOC, ISUBS, NMPS, MXTGT)	QUA3710
	CALL PAGE (0)	QUA3720
	CALL PAGE (3)	QUA3730
	PRINT 1220	QUA3740
	DO 650 I=1, NTGTS	QUA3750
	CALL PAGE (-NSUBS-2)	QUA3760
	PRINT 1150, I	QUA3770
	JLJ=1	QUA3780
	DO 650 JJ=1, NSUBS	QUA3790
	JHI=JLO+NMPS(JJ)-1	QUA3800
	PRINT 1160, (FLAMB(I, J), J=JLO, JHI)	QUA3810
650	JLJ=JHI+1	QUA3820
	GO TO 700	QUA3830
C	CHANGE THE PRODUCT MATRIX, VALUE KILLED ON EACH TARGET, AND	QUA3840
C	MULTIPLIER MATRIX AS CHANGED BY CHANGED ALLOCATION.	QUA3850
660	DO 670 JTYPE=1, NTYPES	QUA3860
	SNEG=SURV(MLOW, JCOL, JTYPE)**(-DELTA)	QUA3870
	SPOS=SURV(MHI, JCOL, JTYPE)**DELTA	QUA3880
	PROD(MLOW, JTYPE)=PROD(MLOW, JTYPE)*SNEG	QUA3890
	PROD(MHI, JTYPE)=PROD(MHI, JTYPE)*SPOS	QUA3900
	OBJSUM=OBJSUM-VKILL(MLOW, JTYPE)-VKILL(MHI, JTYPE)	QUA3910
	VKILL(MLOW, JTYPE)=VAL(MLOW, JTYPE)*(1-PROD(MLOW, JTYPE))	QUA3920
	VKILL(MHI, JTYPE)=VAL(MHI, JTYPE)*(1-PROD(MHI, JTYPE))	QUA3930
	OBJSUM=OBJSUM+VKILL(MLOW, JTYPE)+VKILL(MHI, JTYPE)	QUA3940
	DO 670 J=1, NWPNS	QUA3950
	FLAMBI(MLOW, J, JTYPE)=FLAMBI(MLOW, J, JTYPE)*SNEG	QUA3960
670	FLAMBI(MHI, J, JTYPE)=FLAMBI(MHI, J, JTYPE)*SPOS	QUA3970
	DO 690 J=1, NWPNS	QUA3980
	SUM1=0.	QUA3990
	SUM2=0.	QUA4000
	DO 680 JTYPE=1, NTYPES	QUA4010
	SUM1=SJM1+FLAMBI(MLOW, J, JTYPE)	QUA4020
680	SUM2=SJM2+FLAMBI(MHI, J, JTYPE)	QUA4030
	FLAMB(MLOW, J)=SUM1	QUA4040
690	FLAMB(MHI, J)=SUM2	QUA4050
	IF (IOJT.EQ.1) GO TO 700	QUA4060
	CALL PAGE (-7)	QUA4070

	PRINT 1230, ITER	QUA4080
	PRINT 1240, JCOL, MLOW, MHI, DELTA	QUA4090
	PRINT 1190, OBJSUM	QUA4100
700	CONTINUE	QUA4110
	IF (IOUT.NE.1) GO TO 710	QUA4120
	CALL PAGE (-5)	QUA4130
	PRINT 1230, ITER	QUA4140
	PRINT 1190, OBJSUM	QUA4150
710	DELOBJ=OBJSUM-OBHOLD	QUA4160
	OBHOLD=OBJSUM	QUA4170
	IF (DELOBJ.GE.CHGKIL) GO TO 720	QUA4180
	PRINT 1250, DELOBJ, CHGKIL, ITCUT2	QUA4190
	CALL PAGE (2)	QUA4200
	GO TO 730	QUA4210
720	CONTINUE	QUA4220
C	BRANCH OUT IF NO MORE ITERATIONS ARE POSSIBLE.	QUA4230
	PRINT 1260	( 14240
	CALL PAGE (2)	( 1250
	GO TO 730	( 1200
730	PRINT 1270, EPS	QUA4270
	CALL PAGE (2)	QUA4280
	CALL ALOUT (ITER, NTGTS, NSUBS, NWPNS, ALOC, ISUBS, NMPS, MXTGT)	QUA4290
	CALL KILOUT (NTGTS, NWPNS, NTPES, JBJSUM, ALOC, FLAMB, FLAMBI, SURV, VAL,	QUA4300
	1PRJD, VKILL, NHALOC, RELVAL, MXTGT, MXWPNS)	QUA4310
	PRINT 1190, OBJSUM	QUA4320
	CALL PAGE (2)	QUA4330
	IF (ISOPT.NE.1) GO TO 810	QUA4340
C		QUA4350
C		QUA4360
C	POST-PROCESSING TO RELOCATE AIRCRAFT.	QUA4370
740	IF (IVOPT.EQ.1) GO TO 780	QUA4380
	PSUM=0.	QUA4390
	DO 770 JTYPE=1, NTPES	QUA4400
	PMIN=2.0	QUA4410
	PMAX=-1.0	QUA4420
	CALL PAGE (0)	QUA4430
	CALL PAGE (-NTGTS)	QUA4440
	DO 760 I=1, NTGTS	QUA4450
	PTEST=PROD(I, JTYPE)	QUA4460
	PRINT 1280, I, JTYPE, PTEST	QUA4470
	IF (PTEST.GE.PMIN.OR.VAL(I, JTYPE).LE..0001) GO TO 750	QUA4480
	PMIN=PTEST	QUA4490
	IPMIN=I	QUA4500
750	IF (PTEST.LE.PMAX) GO TO 760	QUA4510
	PMAX=PTEST	QUA4520
	IPMAX=I	QUA4530
760	CONTINUE	QUA4540
	VMIN=VAL(I, PMIN, JTYPE)	QUA4550
	PDELTA=VMIN*(PMAX-PMIN)	QUA4560
	IF (VMIN.LT..1.OR.PDELTA.GT.VMIN) PDELTA=VMIN	QUA4570
	PSUM=PSUM+PDELTA	QUA4580

	VAL (IPMAX, JTYPE)=VAL (IPMAX, JTYPE)+PDEL	QUA459J
	VAL (IPMIN, JTYPE)=VMIN-PDEL	QUA460J
	PDEL=PDEL/RELVAL (JTYPE)	QUA461J
770	PRINT 1290, PDEL, JTYPE, IPMIN, IPMAX	QUA462J
	MOVEV=MOVEV+1	QUA463J
	PRINT 1300, MOVEV	QUA464J
	CALL PAGE (2)	QUA465J
	IF (PSUM.LT..05) GO TO 780	QUA466J
	EPS=.1	QUA467J
	PRINT 1180, EPS	QUA468J
	CALL PAGE (3)	QUA469J
	GO TO 610	QUA470J
C		QUA471J
C		QUA472J
C	WRAP-UP INTEGERIZATION AND OUTPUT.	QUA473J
780	CALL ALINT (NTGTS, NSUBS, ALOC, ISUBS, NMPS, MXTGT)	QUA474J
	PRINT 1310	QUA475J
	CALL PAGE (1)	QUA476J
	IF (IVOPT.NE.2) GO TO 790	QUA477J
	CALL VINT (NTGTS, NTYPES, VAL, RELVAL, MXTGT)	QUA478J
	PRINT 1320	QUA479J
	CALL PAGE (1)	QUA480J
790	CALL ALOUT (ITER, NTGTS, NSUBS, NMPNS, ALOC, ISUBS, NMPS, MXTGT)	QUA481J
	CALL TGTKIL (NTGTS, NMPNS, NTYPES, OBJSUM, ALOC, FLAMB, FLAMBI, SURV, VAL,	QUA482J
	1PROD, VKILL, NHALOC, RELVAL, MXTGT, MXMPNS)	QUA483J
	PRINT 1190, OBJSUM	QUA484J
	CALL PAGE (2)	QUA485J
	GO TO 13	QUA486J
800	PRINT 1330	QUA487J
	STOP	QUA488J
C		QUA489J
C		QUA490J
C	POST-PROCESSING TO RE-LOCATE SUBS.	QUA491J
810	IF (OBJSUM.GE.SUBOBJ) GO TO 830	QUA492J
	MOVES=MOVES+1	QUA493J
	IF (MOVES.LT.1) GO TO 840	QUA494J
820	ISOPT=1	QUA495J
	GO TO 740	QUA496J
830	MOVES=0	QUA497J
840	SUBOBJ=OBJSUM	QUA498J
	IF (IOUT.EQ.1) GO TO 860	QUA499J
	CALL PAGE (0)	QUA500J
	CALL PAGE (3)	QUA501J
	PRINT 1220	QUA502J
	DO 850 I=1, NTGTS	QUA503J
	CALL PAGE (-NSUBS-2)	QUA504J
	PRINT 1150, I	QUA505J
	JLJ=1	QUA506J
	DO 850 JJ=1, NSUBS	QUA507J
	JMI=JLJ+NMPS (JJ)-1	QUA508J
	PRINT 1160, (FLAMB (I, J), J=JLJ, JMI)	QUA509J

853	JLJ=JHI+1	QUA5100
850	IBOUT=0	QUA5110
	CALL SUBADJ (NTGTS,NSUBS,MTYPE,S,EPS,IBOUT,ALOC,FLAMB,ISUBS,NMPS,MT	QUA5120
	TYPE,MXTGT)	QUA5130
	IF (IBOUT.EQ.1) GO TO 820	QUA5140
	MOVEST=MOVEST+1	QUA5150
	PRINT 1340, MOVEST	QUA5160
	CALL PAGE (1)	QUA5170
	IF (IOUT.EQ.1) GO TO 870	QUA5180
	CALL ALOUT (I,NTGTS,NSUBS,NWPNS,ALOC,ISUBS,NMPS,MXTGT)	QUA5190
870	NCOL=0	QUA5200
	DO 880 I=1,NSUBS	QUA5210
880	IF (ISUBS(I).GT.0) NCOL=NCOL+NMPS(I)	QUA5220
	CALL PAGE (-NSUBS-2)	QUA5230
	PRINT 1350, (I,ISUBS(I),MTYPE(I),I=1,NSUBS)	QUA5240
	EPS=.1	QUA5250
	PRINT 1130, EPS	QUA5260
	CALL PAGE (3)	QUA5270
	GO TO 613	QUA5280
	C	QUA5290
890	FORMAT (14I5)	QUA5300
900	FORMAT (6H CASE ,I5,10X,5HMODE ,I5,11X,5HDATE ,A10,10X,5HTIME ,A10,	QUA5310
	1)	QUA5320
910	FORMAT (17H NEW PROBLEM WITH,I5,3H TARGETS,,I5,15H SUB LOCATIONS,,	QUA5330
	1I5,23H TYPES OF AIRCRAFT, AND,I5,19H TYPES OF MISSILES.)	QUA5340
920	FORMAT (26H RESTART INFORMATION READ.)	QUA5350
930	FORMAT (3F10.4,I5,5X,3F10.4/(7F10.4))	QUA5360
940	FORMAT (70I1)	QUA5370
950	FORMAT (3F10.4,I5)	QUA5380
960	FORMAT (7F10.4)	QUA5390
970	FORMAT (/27H TARGET LOCATIONS (DEGREES),3X,7HRUNWAYS,2X,8HCENTROID	QUA5400
	1,2X,8HAIRCRAFT/39X,6HDISTANCE,6X,15HTYPE AND NUMBER)	QUA5410
980	FORMAT (/I7,2F10.4,I10,F10.4,I10,F11.4/(47X,I10,F11.4))	QUA5420
990	FORMAT (32X,25HTAKE-OFF SEQUENCE BY TYPE,5X,30I2/(62X,30I2))	QUA5430
1000	FORMAT (/2X,8HAIRCRAFT,2X,8HRELATIVE,5X,5HBRAKE,/6X,4HTYPE,5X,5HVAA	QUA5440
	1LUE,3X,7HRELEASE,7X,3HPSI,7X,3HCAL,/(I10,3F10.4,I10))	QUA5450
1010	FORMAT (/28H AIRCRAFT TAKE-OFF INTERVALS/8X,7HRUNWAYS,5X,5HTYPE,1,5	QUA5460
	1X,5HTYPE2,3X,7HMINUTES/(I15,2I10,F10.4))	QUA5470
1020	FORMAT (2F10.4,3I5)	QUA5480
1030	FORMAT (/2X,23H SUB LOCATIONS (DEGREES),6X,4H SUBS,3X,13H MISSILES AN	QUA5490
	10 ,4HTYPE/(I5,2F10.4,I10,I11,I8))	QUA5500
1040	FORMAT (/23H NUMBER OF SUBS OF TYPE,I5,2H =,I5)	QUA5510
1050	FORMAT (I5,5X,6F10.4/(6F10.4))	QUA5520
1060	FORMAT (/3X,7H MISSILE,4X,6H LAUNCH,8X,13H RELIABILITIES,16X,3H MIN,7X	QUA5530
	1,3H MAX,5X/6X,4HTYPE,2X,8H INTERVAL,5X,6H LAUNCH,2X,6H FLIGHT,1X,7H WAR	QUA5540
	2HEAD,3X,2(5X,5HRANGE),5X,5HYIELD/(I11,F10.4,3X,3F8.4,3X,3F10.4))	QUA5550
1070	FORMAT (/13H MISSILE TYPE,11X,4HTIME,5X,5HRANGE/I13,5X,2F10.4/(18X	QUA5560
	1,2F10.4))	QUA5570
1080	FORMAT (16H FOR DISTANCE OF,F6.2,30H NM TO CENTROID FROM START OF	QUA5580
	1,13HTAKE-OFF ROLL)	QUA5590
1090	FORMAT (/14H AIRCRAFT TYPE,I5,20H VERSUS MISSILE TYPE,I5//19H WHE	QUA5600

```

1N DETONATION IS,F7.2,32H NM FROM CENTROID, LETHAL AREA =,F8.2,22H QUA5610
2SQUARE NM AND EXTENDS,F6.2,26H NM FARTHER FROM CENTROID.) QUA5620
1100 FORMAT (/7H TARGET,IS,20H BRAKE RELEASE TIMES/(16H AIRCRAFT TYPE =QUA5630
1,IS,10H STARTS AT,F7.2,9H MINUTES.)) QUA5640
1110 FORMAT (/19H DISTANCE TO TARGET,I4,13H FROM SUB LOCATION,I4,3H IS,QUA5650
1F10.2,30H NM. FLIGHT TIME (MISSILE TYPE,I3,3H) =,E10.8,5H MIN.) QUA5660
1120 FORMAT (7H WEAPON,IS,18H ARRIVES ON TARGET,IS,12H WHEN FIRST ,11HQUA5670
1AIRCRAFT IS,F10.2,32H NAUTICAL MILES BEYOND CENTROID.) QUA5680
1130 FORMAT (13H ANNULUS PK =,F7.4,15H, CIRCULAR PK =,F7.4,9H, WEAPON,QUA5690
1IS,18H VS. AIRCRAFT TYPE,IS,20H. ANNULAR RADII ARE,F10.4,4H AND,FQUA5700
210.4) QUA5710
1140 FORMAT (36H SURVIVABILITY MATRIX, AIRCRAFT TYPE,IS//) QUA5720
1150 FORMAT (/7H TARGET,IS) QUA5730
1160 FORMAT (1X,16F7.4) QUA5740
1170 FORMAT (F10.4,2I5,F10.4) QUA5750
1180 FORMAT (/46H LAGRANGE MULTIPLIERS MUST DIFFER BY AT LEAST ,F13.10,QUA5760
120H TO CAUSE ITERATION./) QUA5770
1190 FORMAT (/24H EXPECTED VALUE KILLED =,F10.4) QUA5780
1200 FORMAT (33H RESTART INFORMATION WRITTEN OUT.) QUA5790
1210 FORMAT (/48H MULTIPLIER MATRIX CONVERGED WITHIN TOLERANCE OF,F13.1QUA5800
10) QUA5810
1220 FORMAT (/18H MULTIPLIER MATRIX//) QUA5820
1230 FORMAT (//17H ITERATION NUMBER,I10) QUA5830
1240 FORMAT (/18H DELTA N IN COLUMN,I4,9H FROM ROW,I4,7H TO ROW,I4,3H IQUA5840
1S,F10.3) QUA5850
1250 FORMAT (/21H KILL CHANGED BY ONLY,F7.4,11H (LESS THAN,F7.4,9H) IN QUA5860
1PAST,IS,19H ITERATIONS. QUIT.) QUA5870
1260 FORMAT (/42H ITERATION CUT-OFF LIMIT HAS BEEN REACHED.) QUA5880
1270 FORMAT (/24H CURRENT DELTA LAMBDA IS,F13.10) QUA5890
1280 FORMAT (33H SURVIVABILITY PRODUCT FOR TARGET,IS,15H, AIRCRAFT TYPEQUA5900
1,IS,2H =,F10.4) QUA5910
1290 FORMAT (1X,F10.4,17H AIRCRAFT OF TYPE,IS,18H MOVED FROM TARGET,IS,QUA5920
110H TO TARGET,IS) QUA5930
1300 FORMAT (27H THIS IS VALUE SHIFT NUMBER,IS) QUA5940
1310 FORMAT (23H ALLOCATION INTEGERIZED) QUA5950
1320 FORMAT (20H BEDDOWN INTEGERIZED) QUA5960
1330 FORMAT (///4H EOJ//) QUA5970
1340 FORMAT (20H THIS IS MOVE NUMBER,IS,13H OF A SUB.) QUA5980
1350 FORMAT (/17H SUB POINT NUMBER,5X,14HNUMBER OF SUBS,5X,8HSUB TYPE/(QUA5990
1I10,I19,I17)) QUA6000
END QUA6010-
SUBROUTINE ALOUT (ITER,NTGTS,NSUBS,NWPNS,ALOC,ISUBS,NMPS,MXTGT) ALO 10
DIMENSION ALOC(MXTGT,1), ISUBS(1), NMPS(1) ALO 20
CALL PAGE (0) ALO 30
CALL PAGE (2) ALO 40
PRINT 40, ITER ALO 50
DO 10 I=1,NTGTS ALO 60
CALL PAGE (-3) ALO 70
PRINT 50, I ALO 80
J=0 ALO 90
DO 10 ISUB=1,NSUBS ALO 100

```

	ISLIM=NMPS(ISUB)	ALO 116
	DO 10 ISALVO=1,ISLIM	ALO 120
	J=J+1	ALO 130
	ALOCN=ALOC(I,J)	ALO 140
	IF (ALOCN.LT..0001) GO TO 10	ALO 150
	PRINT 60, ALOCN,ISUBS(ISUB),ISUB,ISALVO,J	ALO 160
	CALL PAGE (1)	ALO 170
10	CONTINJE	ALO 180
	MWPN=0	ALO 190
	DO 35 I=1,NSUBS	ALO 200
	CALL PAGE (-3)	ALO 210
	PRINT 70, I	ALO 220
	JSLIM=NMPS(I)	ALO 230
	IF(ISUBS(I).NE.0) GO TO 15	ALO 234
	MWPN=MWPN+JSLIM	ALO 238
	GO TO 35	ALO 242
15	DO 30 J=1,JSLIM	ALO 246
	MWPN=MWPN+1	ALO 250
	DO 20 K=1,NTGTS	ALO 260
	ALOCN=ALOC(K,MWPN)	ALO 270
	IF (ALOCN.LT..0001) GO TO 20	ALO 280
	PRINT 80, ALOCN,J,K	ALO 290
	CALL PAGE (1)	ALO 300
20	CONTINJE	ALO 310
30	CONTINUE	ALO 320
35	CONTINJE	ALO 325
	RETURN	ALO 330
C		ALO 340
40	FORMAT (12H ALLOCATION,,5X,16H ITERATION NUMBER,I10//)	ALO 350
50	FORMAT (/7H TARGET,I5)	ALO 360
60	FORMAT (3X,F9.4,14H MISSILES FROM,I4,21H SUBS AT SUB LOCATION,14,7A	ALO 370
	1H, SALVO,I4,2X,7H(WEAPON,I5,2H).)	ALO 380
70	FORMAT (/13H SUB LOCATION,I5)	ALO 390
80	FORMAT (3X,F9.4,20H MISSILES FROM SALVO,I4,10H TO TARGET,I4)	ALO 400
	END	ALO 410-
	SUBROUTINE ALINT (NTGTS,NSUBS,ALOC,ISUBS,NMPS,MXTGT)	ALI 10
	DIMENSION ALOC(MXTGT,1), ISUBS(1), NMPS(1), ALHOLD(35)	ALI 20
C	INTEGERIZING THE ALLOCATION MATRIX, COLUMN BY COLUMN.	ALI 30
	JWPN=0	ALI 40
	DO 80 J=1,NSUBS	ALI 50
	MLIM=NMPS(J)	ALI 60
	IF (ISUBS(J).NE.0) GO TO 10	ALI 70
	JWPN=JWPN+MLIM	ALI 80
	GO TO 80	ALI 90
10	DO 70 M=1,MLIM	ALI 100
	JWPN=JWPN+1	ALI 110
	DO 20 I=1,NTGTS	ALI 120
20	ALHOLD(I)=0.	ALI 130
	SUMW=0.	ALI 140
	DO 30 I=1,NTGTS	ALI 150
	SUMW=SUMW+ALOC(I,JWPN)	ALI 160

33	CONTINUE	ALI 170
	NMSLS=SUMH+.01	ALI 180
	IF (NMSLS.EQ.0) GO TO 70	ALI 190
	DO 50 JSUB=1,NMSLS	ALI 200
	HOLD=0.	ALI 210
	DO 40 I=1,NTGTS	ALI 220
	TEST=ALOC(I,JWPN)	ALI 230
	IF (TEST.LE.HOLD) GO TO 40	ALI 240
	HOLD=TEST	ALI 250
	IMAX=I	ALI 260
40	CONTINUE	ALI 270
	ALHOLD(IMAX)=ALHOLD(IMAX)+1.	ALI 280
	ALJC(IMAX,JWPN)=ALOC(IMAX,JWPN)-1.	ALI 290
	IF (ALOC(IMAX,JWPN).LT.0.) ALOC(IMAX,JWPN)=0.	ALI 300
50	CONTINUE	ALI 310
	DO 60 I=1,NTGTS	ALI 320
60	ALJC(I,JWPN)=ALHOLD(I)	ALI 330
70	CONTINUE	ALI 340
80	CONTINUE	ALI 350
	RETURN	ALI 360
	END	ALI 370-
	SUBROUTINE VINT (NTGTS,NTYPES,VAL,RELVAL,MXTGT)	VIN 10
	DIMENSION VAL(MXTGT,1), RELVAL(1), VHOLD(35)	VIN 20
	DO 60 JTYPE=1,NTYPES	VIN 30
	RV=RELVAL(JTYPE)	VIN 40
	DO 10 I=1,NTGTS	VIN 50
10	VHOLD(I)=0.	VIN 60
	SUMV=0.	VIN 70
	DO 20 I=1,NTGTS	VIN 80
20	SUMV=SUMV+VAL(I,JTYPE)/RV	VIN 90
	NVAL=SUMV+.01	VIN 100
	DO 40 IVAL=1,NVAL	VIN 110
	HOLD=C.	VIN 120
	DO 30 I=1,NTGTS	VIN 130
	TEST=VAL(I,JTYPE)	VIN 140
	IF (TEST.LE.HOLD) GO TO 30	VIN 150
	HOLD=TEST	VIN 160
	IMAX=I	VIN 170
30	CONTINUE	VIN 180
	VHOLD(IMAX)=VHOLD(IMAX)+RV	VIN 190
	VAL(IMAX,JTYPE)=VAL(IMAX,JTYPE)-RV	VIN 200
	IF (VAL(IMAX,JTYPE).LT.0.) VAL(IMAX,JTYPE)=0.	VIN 210
40	CONTINUE	VIN 220
	DO 50 I=1,NTGTS	VIN 230
50	VAL(I,JTYPE)=VHOLD(I)	VIN 240
60	CONTINUE	VIN 250
	RETURN	VIN 260
	END	VIN 270-
	SUBROUTINE TGTKIL (NTGTS,NWPNS,NTYPES,OBJSUM,ALOC,FLAMB,FLAMBI,SURTKL	TKL 10
	1V,VAL,PROD,VKILL,NWALOC,RELVAL,MXTGT,MXWPNS)	TKL 20
	DIMENSION ALOC(MXTGT,1), FLAMB(MXTGT,1), SURV(MXTGT,MXWPNS,1), FLATKL	TKL 30

```

1MBI(MXTGT,MXWPNS,1), VAL(MXTGT,1), PROD(MXTGT,1), VKILL(MXTGT,1), TKL 40
2NHALOC(1), RELVAL(1), SUMAC(5), SUMACK(5) TKL 50
C COMPUTE FROM SCRATCH, THE PRODUCTS OF EACH TARGET'S SURVIVABILITY TKL 60
C AND VALUE. COUNT WEAPONS ALLOCATED (BY ROUNDING OFF). TKL 70
DO 20 I=1,NTGTS TKL 80
NHALOC(I)=0. TKL 90
DO 10 JTYPE=1,NTYPES TKL 100
13 PRJD(I,JTYPE)=1.0 TKL 110
DO 20 J=1,NWPNS TKL 120
ALOCN=ALOC(I,J) TKL 130
IPART=ALOCN+.5 TKL 140
NHALOC(I)=NHALOC(I)+IPART TKL 150
DO 20 JTYPE=1,NTYPES TKL 160
20 SURVN=SURV(I,J,JTYPE) TKL 170
PROD(I,JTYPE)=PROD(I,JTYPE)*SURVN**ALOCN TKL 180
C COMPUTE FROM SCRATCH, THE VALUE KILLED ON EACH TARGET AND THE TKL 190
C MULTIPLIER MATRIX. TKL 200
OBJSUM=0. TKL 210
DO 50 I=1,NTGTS TKL 220
DO 30 JTYPE=1,NTYPES TKL 230
VKILL(I,JTYPE)=VAL(I,JTYPE)*(1-PROD(I,JTYPE)) TKL 240
33 OBJSUM=OBJSUM+VKILL(I,JTYPE) TKL 250
DO 50 J=1,NWPNS TKL 260
SUM=0. TKL 270
DO 40 JTYPE=1,NTYPES TKL 280
ADDEND=-PROD(I,JTYPE)*ALOG(SURV(I,J,JTYPE))*VAL(I,JTYPE) TKL 290
FLAMBI(I,J,JTYPE)=ADDEND TKL 300
43 SJM=SUM+ADDEND TKL 310
50 FLAMB(I,J)=SUM TKL 320
C TKL 330
ENTRY KILOUT TKL 340
C TKL 350
DO 60 JTYPE=1,NTYPES TKL 360
SUMAC(JTYPE)=0. TKL 370
63 SUMACK(JTYPE)=0. TKL 380
CALL PAGE (2) TKL 390
CALL PAGE (3) TKL 400
PRINT 80 TKL 410
DO 70 I=1,NTGTS TKL 420
CALL PAGE (-NTYPES-1) TKL 430
PRINT 90, I,NHALOC(I) TKL 440
DO 70 JTYPE=1,NTYPES TKL 450
V=VAL(I,JTYPE) TKL 460
VK=VKILL(I,JTYPE) TKL 470
RV=RELVAL(JTYPE) TKL 480
VN=V/RV TKL 490
VKN=VK/RV TKL 500
PRINT 100, JTYPE,V,VK,VN,VKN TKL 510
SUMAC(JTYPE)=SUMAC(JTYPE)+VN TKL 520
70 SUMACK(JTYPE)=SUMACK(JTYPE)+VKN TKL 530
PRINT 110, (JTYPE,SUMAC(JTYPE),SUMACK(JTYPE),JTYPE=1,NTYPES) TKL 540

```

	CALL PAGE (NTYPES+1)	TKL 550
	RETURN	TKL 560
C		TKL 570
80	FORMAT (10X,6HTARGET,9X,7HWEAPONS,8X,8HAIRCRAFT,2(11X,5HTOTAL,10X,TKL 580	
	16HKILLED)/10X,6HNUMBER,7X,9HALLOCATED,12X,4HTYPE,2(11X,5HVALUE),2(TKL 590	
	28X,8HAIRCRAFT)/)	TKL 600
90	FORMAT (2I16)	TKL 610
100	FORMAT (32X,I16,4F16.4)	TKL 620
110	FORMAT (7H TOTALS/(32X,I16,32X,2F16.4))	TKL 630
	END	TKL 640
	SUBROUTINE ADJLAM (NTGTS,NSUBS,NTYPES,NCOL,IBROUT,LOWPT,MAXPT,JCOLDL 10	
	1,DELTA,EPS,ALOG,FLAMB,FLAMBI,SURV,NWALOG,ISUBS,NMPS,MXTGT,MXWPN) ADL 20	
	1 DIMENSION ALOC(MXTGT,1), FLAMB(MXTGT,1), SURV(MXTGT,MXWPN,1), NWAADL 30	
	1 LOC(1), ISUBS(1), NMPS(1), FLAMBI(MXTGT,MXWPN,1)	ADL 40
	DATA J,ISUB,ISALVO/0,1,0/	ADL 50
	DO 60 JCNT=1,NCOL	ADL 60
	J=J+1	ADL 70
	ISALVO=ISALVO+1	ADL 80
	IF (ISALVO.LE.NMPS(ISUB)) GO TO 20	ADL 90
	ISALVO=1	ADL 100
10	ISUB=ISUB+1	ADL 110
	IF (ISUB.LE.NSUBS) GO TO 20	ADL 120
	ISUB=1	ADL 130
	J=1	ADL 140
20	IF (ISUBS(ISUB).NE.0) GO TO 30	ADL 150
	J=J+NMPS(ISUB)	ADL 160
	GO TO 10	ADL 170
30	JCOL=J	ADL 180
	FMAX=-1.	ADL 190
	MAXPT=1	ADL 200
	FMIN=9999.	ADL 210
	DO 50 I=1,NTGTS	ADL 220
	FTESTL=FLAMB(I,J)	ADL 230
	IF (FTESTL.LE.FMAX) GO TO 40	ADL 240
	FMAX=FTESTL	ADL 250
	MAXPT=I	ADL 260
40	IF (ALOG(I,J).LT..0001.OR.FTESTL.GE.FMIN) GO TO 50	ADL 270
	FMIN=FTESTL	ADL 280
	LOWPT=I	ADL 290
50	CONTINUE	ADL 300
	IF (FMIN.LT.FMAX-EPS) GO TO 70	ADL 310
60	CONTINUE	ADL 320
C	NO EXCHANGES OF ALLOCATION POSSIBLE.	ADL 330
	IBROUT=1	ADL 340
	RETURN	ADL 350
C	COMPUTE INCREMENT IN ALLOCATION.	ADL 360
70	ALOH=ALOG(LOWPT,JCOL)	ADL 370
	IF (NTYPES.NE.1) GO TO 10J	ADL 380
	IF (FMIN/FMAX.LT..0001) GO TO 80	ADL 390
	DELTA=ALOG(FMIN/FMAX)/ALOG(SURV(LOWPT,JCOL,1)*SURV(MAXPT,JCOL,1))	ADL 400
	GO TO 90	ADL 410

80	DELTA=9339.	ADL 420
90	IF (DELTA.GT.AL0W) DELTA=AL0W	ADL 430
	GO TO 110	ADL 440
100	DELTA=XNEW(J.,AL0W,EPS,LOWPT,MAXPT,JC0L,NTYPES,SURV,FLAMBI,MXTGT,ADL 450	
	1,MXHPNS)	ADL 460
110	AMAX=AL0C(MAXPT,JC0L)	ADL 470
	IPART1=AMAX+.5	ADL 480
	ANEW=AMAX+DELTA	ADL 490
	IPART=ANEW+.5	ADL 500
	NWAL0C(MAXPT)=NWAL0C(MAXPT)-IPART1+IPART	ADL 510
	AL0C(MAXPT,JC0L)=ANEW	ADL 520
	IPART1=AL0W+.5	ADL 530
	ANEW=AL0W-DELTA	ADL 540
	IPART=ANEW+.5	ADL 550
	NWAL0C(LOWPT)=NWAL0C(LOWPT)-IPART1+IPART	ADL 560
	AL0C(LOWPT,JC0L)=ANEW	ADL 570
	RETURN	ADL 580
	END	ADL 590-
	FUNCTION XNEW (XMIN,XMAX,EPS,LOWPT,MAXPT,JC0L,NTYPES,SURV,FLAMBI,XNT 10	
	1,MXTGT,MXHPNS)	XNT 20
	DIMENSION SURV(MXTGT,MXHPNS,1), FLAMBI(MXTGT,MXHPNS,1)	XNT 30
	DATA XBND/200./	XNT 40
	CALL FOFP (XMAX,LOWPT,MAXPT,JC0L,NTYPES,FMAX,DIV,SURV,FLAMBI,MXTGT,XNT 50	
	1,MXHPNS)	XNT 60
	IF (FMAX.LT.0.0) GO TO 60	XNT 70
	IST=0	XNT 80
	XNEW=XMAX-DIV	XNT 90
10	IF (ABS(XNEW).LT.XBND) GO TO 50	XNT 100
	IST=IST+1	XNT 110
	GO TO (20,30,40), IST	XNT 120
20	CALL FOFP (XMIN,LOWPT,MAXPT,JC0L,NTYPES,DUMMY,DIV,SURV,FLAMBI,MXTGT,XNT 130	
	1T,MXHPNS)	XNT 140
	XNEW=XMIN-DIV	XNT 150
	GO TO 10	XNT 160
30	XMID=(XMAX+XMIN)/2.	XNT 170
	CALL FOFP (XMID,LOWPT,MAXPT,JC0L,NTYPES,DUMMY,DIV,SURV,FLAMBI,MXTGT,XNT 180	
	1T,MXHPNS)	XNT 190
	XNEW=XMID-DIV	XNT 200
	GO TO 10	XNT 210
40	PRINT 70, IST	XNT 220
	XNEW=XMID	XNT 230
	RETURN	XNT 240
50	XOLD=XNEW	XNT 250
	CALL FOFP (XOLD,LOWPT,MAXPT,JC0L,NTYPES,DUMMY,DIV,SURV,FLAMBI,MXTGT,XNT 260	
	1T,MXHPNS)	XNT 270
	XNEW=XOLD-DIV	XNT 280
	IF (ABS(XNEW-XOLD).GT.EPS) GO TO 10	XNT 290
	XNEW=XNEW	XNT 300
	RETURN	XNT 310
60	XNEW=XMAX	XNT 320
	RETURN	XNT 330

C		XNT 340
70	FORMAT (16H NEWTON OIVERGED,I5,27H TIMES. XMID USEO AS ROOT.)	XNT 350
	END	XNT 360-
	SUBROUTINE FOFP (X,LOWPT,MAXPT,JCOL,NTYPES,SUM,DIV,SURV,FLAMBI,MXTGT,MAXPNS)	FOP 10
	DIMENSION SURV(MXTGT,MAXPNS,1), FLAMBI(MXTGT,MAXPNS,1)	FOP 20
	SUM=0.	FOP 30
	PRIME=0.	FOP 40
	DO 10 JTYPE=1,NTYPES	FOP 50
	FLOW=FLAMBI(LOWPT,JCOL,JTYPE)	FOP 60
	FLHI=FLAMBI(MAXPT,JCOL,JTYPE)	FOP 70
	SLOW=SURV(LOWPT,JCOL,JTYPE)	FOP 80
	SHI=SURV(MAXPT,JCOL,JTYPE)	FOP 90
	PLOW=FLOW*SLOW**(-X)	FOP 100
	PHI=FLHI*SHI**X	FOP 110
	SUM=SUM+PLOW-PHI	FOP 120
10	PRIME=PRIME-PLOW*ALOG(SLOW)-PHI*ALOG(SHI)	FOP 130
	DIV=SUM/PRIME	FOP 140
	RETURN	FOP 150
	END	FOP 160
	SUBROUTINE SUBADJ (NTGTS,NSUBS,MTYPES,EPS,IBOUT,ALOC,FLAMB,ISUBS,NSUBS,MTYPE,MXTGT)	FOP 170-
	DIMENSION LOMHOLD(16), MXHOLD(16), LOTEMP(16), MXTEMP(16), LOM(16),	SUB 10
	1 MXH(16), ALOC(MXTGT,1), FLAMB(MXTGT,1), ISUBS(1), NMPS(1), MTYPE(	SUB 20
	21)	SUB 30
	BIGDIF=-1.	SUB 40
	CALL PAGE (0)	SUB 50
	DO 110 JTYPE=1,MTYPES	SUB 60
	HOLDLO=1.0E+300	SUB 70
	HOLDHI=-1.	SUB 80
	JMPN=0	SUB 90
	DO 90 J=1,NSUBS	SUB 100
	NMIS=NMPS(J)	SUB 110
	IF (MTYPE(J).NE.JTYPE) GO TO 80	SUB 120
	IBRSW=1	SUB 130
	IF (ISUBS(J).EQ.0) IBRSW=2	SUB 140
	SUMLO=0.	SUB 150
	SUMHI=0.	SUB 160
	NM=NMIS	SUB 170
	DO 40 M=1,NMIS	SUB 180
	JMPN=JMPN+1	SUB 190
	HLO=1.0E+300	SUB 200
	HMI=-1.	SUB 210
	DO 30 I=1,NTGTS	SUB 220
	TEST=FLAMB(I,JMPN)	SUB 230
	IF (TEST.LE.HMI) GO TO 10	SUB 240
	HMI=TEST	SUB 250
	MXTEMP(M)=I	SUB 260
10	GO TO (20,30), IBRSW	SUB 270
20	IF (TEST.GE.HLO.OR.ALOC(I,JMPN).LT..0001) GO TO 30	SUB 280
	HLO=TEST	SUB 290
		SUB 300
		SUB 310

	LOTEMP(M)=I	SUB 320
30	CONTINUE	SUB 330
	SUMLO=SUMLO+HLO	SUB 340
40	SUMHI=SUMMI+HMI	SUB 350
	IF (SUMMI.LE.HOLDMI) GO TO 60	SUB 360
	HOLDMI=SUMMI	SUB 370
	MAXS=J	SUB 380
	MAXWS=JWPN-NMIS	SUB 390
	DO 50 M=1,NMIS	SUB 400
50	MXH(M)=MXTEMP(M)	SUB 410
60	IF (SUMLO.GE.MOLDLO) GO TO 90	SUB 420
	MOLOLO=SUMLO	SUB 430
	LOWS=J	SUB 440
	LOWWS=JWPN-NMIS	SUB 450
	DO 70 M=1,NMIS	SUB 460
70	LOH(M)=LOTEMP(M)	SUB 470
	GO TO 90	SUB 480
80	JWPN=JWPN+NMIS	SUB 490
90	CONTINUE	SUB 500
	CALL PAGE (-3)	SUB 510
	PRINT 180, JTYPE, MOLOLO, HOLOHI	SUB 520
	AVDIF=(HOLDHI-HOLDLO)/NM	SUB 530
	IF (AVDIF.LE.BIGDIF) GO TO 110	SUB 540
	BIGDIF=AVDIF	SUB 550
	MAXSUB=MAXS	SUB 560
	MAXW=MAXWS	SUB 570
	LOWSUB=LOWS	SUB 580
	LOWW=LOWWS	SUB 590
	DO 100 M=1,NM	SUB 600
	MXHOLD(M)=MXH(M)	SUB 610
100	LOHOLD(M)=LOH(M)	SUB 620
110	CONTINUE	SUB 630
	IF (BIGDIF.GE.EPS) GO TO 120	SUB 640
	IBOUT=1	SUB 650
	PRINT 190	SUB 660
	RETURN	SUB 670
C	FINO MAX LAMBDA IN COLUMN OF WEAPON TO BE ADDED.	SUB 680
120	MLIM=NMPS(MAXSUB)	SUB 690
	DO 170 M=1,MLIM	SUB 700
	MAXWPN=MAXW+M	SUB 710
	LAMAX=MXHOLD(M)	SUB 720
C	ADD A WEAPON IN APPROPRIATE SPOT.	SUB 730
	ALOC(LAMAX,MAXWPN)=ALOC(LAMAX,MAXWPN)+1.0	SUB 740
	LOWWPN=LOWW+M	SUB 750
	LAMLOW=LOHOLD(M)	SUB 760
C	MOVE WEAPON FRACTIONS UNTIL ONE WEAPON MOVED.	SUB 770
	ALFRAC=0.	SUB 780
	GO TO 150	SUB 790
130	FMIN=1.0E+300	SUB 800
	DO 140 I=1,NTGTS	SUB 810
	TEST=FLAMB(I,LOWWPN)	SUB 820

	IF (TEST.GE.FMIN.OR.ALGC(I,LOWHPN).LT..001) GO TO 140	SUB 830
	FMIN=TEST	SUB 840
	LAMLOW=I	SUB 850
140	CONTINUE	SUB 860
150	AL=ALOC(LAMLOW,LOWHPN)	SUB 870
	IF (AL.GT..999-ALFRAC) GO TO 160	SUB 880
	ALFRAC=ALFRAC+AL	SUB 890
	ALJC(LAMLOW,LOWHPN)=0.	SUB 900
	GO TO 130	SUB 910
160	ALJC(LAMLOW,LOWHPN)=ALOC(LAMLOW,LOWHPN)-(1.0-ALFRAC)	SUB 920
170	CONTINUE	SUB 930
	ISUBS(MAXSUB)=ISUBS(MAXSUB)+1	SUB 940
	ISUBS(LOWSUB)=ISUBS(LOWSUB)-1	SUB 950
	PRINT 200, LOWSUB,MAXSUB	SUB 960
	CALL PAGE (1)	SUB 970
	RETURN	SUB 980
C		SUB 990
180	FORMAT (/17H FOR MISSILE TYPE,I5/28H SUM OF LOWEST LAMBDAS WITH ,9SUB100-	
	1HWEAPONS =,F10.4/25H SUM OF HIGHEST LAMBDAS =,F10.4)	SUB1010
190	FORMAT (/15H SUBS CONVERGED)	SUB1020
200	FORMAT (24H SUB MOVED FROM LOCATION,I5,12H TO LOCATION,I5)	SUB1030
	END	SUB1040-
	FUNCTION PROFLU (XLU,X,Y,NPTS)	PRD 10
	DIMENSION X(NPTS), Y(NPTS)	PRD 20
	XHI=X(NPTS)	PRD 30
	XLO=X(1)	PRD 40
	YLO=Y(1)	PRD 50
	SLOPE=(Y(NPTS)-Y(NPTS-1))/(XHI-X(NPTS-1))	PRD 60
	IF (XLU.LE.XHI.AND.XLU.GE.XLO) GO TO 10	PRD 70
	IF (XLU.GT.XHI) PROFLU=Y(NPTS)+SLOPE*(XLU-XHI)	PRD 80
	IF (XLU.LT.XLO) PROFLU=YLO	PRD 90
	RETURN	PRD 100
10	PROFLU=ALAG(XLU,X,Y,NPTS)	PRD 110
	IF (PROFLU.LE.YLO) PROFLU=YLO	PRD 120
	RETURN	PRD 130
	END	PRD 140-
	FUNCTION XAREA (XL,XD,AL,AH)	XAR 10
	DIMENSION NUMBER(17), NUMBER(8)	XAR 20
	COMMON /LIST1/ PI	XAR 30
	DATA (NUMBER(J),J=1,17)/2121,2221,2321,2331,3121,3212,3221,3222,33	XAR 40
	112,3321,3322,3331,3332,4221,4321,4331,5321/	XAR 50
	DATA (NUMBER(J),J=1,8)/512,611,612,711,712,722,812,912/	XAR 60
		XAR 70
		XAR 80
C	XAREA SUBROUTINE RETURNS THE INTERSECTION OF THE CIRCULAR LETHAL	XAR 90
C	AREA OF A WEAPON WITH AN ANNULUS FORMED BY THE MAXIMUM AND	XAR 100
C	MINIMUM AIRCRAFT FLYOUT DISTANCES FROM A CENTROID.	XAR 110
C		XAR 120
C		XAR 130
C	INPUT VARIABLE DESCRIPTION-	XAR 140
C		XAR 150



	RETURN	XAR 670
30	XAREA=0.0	XAR 680
	RETURN	XAR 690
40	XAREA=SUB02 (XL, XD, AL, AH)	XAR 700
	RETURN	XAR 710
50	XAREA=SUB03 (XL, XD, AL, AH)	XAR 720
	RETURN	XAR 730
60	XAREA=PI*XL**2	XAR 740
	RETURN	XAR 750
70	XAREA=SUB05 (XL, XD, AL, AH)	XAR 760
	RETURN	XAR 770
80	XAREA=SUB06 (XL, XD, AL, AH)	XAR 780
	RETURN	XAR 790
90	XAREA=0.0	XAR 800
	RETURN	XAR 810
100	XAREA=PI*(XL**2-AL**2)	XAR 820
	RETURN	XAR 830
110	XAREA=SUB09 (XL, XD, AL, AH)	XAR 840
	RETURN	XAR 850
120	XAREA=SUB11 (XL, XD, AL, AH)	XAR 860
	RETURN	XAR 870
130	XAREA=SUB12 (XL, XD, AL, AH)	XAR 880
	RETURN	XAR 890
140	XAREA=SUB13 (XL, XD, AL, AH)	XAR 900
	RETURN	XAR 910
150	XAREA=SUB05 (XL, XD, AL, AH)-PI*AL**2	XAR 920
	RETURN	XAR 930
160	XAREA=SUB06 (XL, XD, AL, AH)-PI*AL**2	XAR 940
	RETURN	XAR 950
170	XAREA=SUB16 (XL, XD, AL, AH)	XAR 960
	RETURN	XAR 970
180	XAREA=SUB17 (XL, XD, AL, AH)	XAR 980
	RETURN	XAR 990
190	A=XD-XL	XAR1000
	IF (A.LE.(-AH)) I4=500	XAR1010
	IF (A.GT.(-AH).AND.A.LE.(-AL)) I4=600	XAR1020
	IF (A.GT.(-AL).AND.A.LE.(AL)) I4=700	XAR1030
	IF (A.GT.AL.AND.A.LE.AH) I4=800	XAR1040
	IF (A.GT.AH) I4=900	XAR1050
	IF (I4.EQ.700) XMIN=(XD**2+XL**2-AL**2)/(2.0*XD)	XAR1060
	IF (I4.GE.600.AND.I4.LE.800) XMAX=(XD**2+XL**2-AH**2)/(2.0*XD)	XAR1070
	IF (XMIN.LE.0.0) I3=10	XAR1080
	IF (XMIN.GT.0.0) I3=20	XAR1090
	IF (XMAX.LE.0.0) I2=1	XAR1100
	IF (XMAX.GT.0.0) I2=2	XAR1110
	NUM=I4+I3+I2	XAR1120
	ITST=1	XAR1130
	DO 200 J=1,8	XAR1140
	IF (NUM.GE.NUMBER(J)) ITST=ITST+1	XAR1150
200	CONTINUE	XAR1160
	GO TO (210,220,230,240,250,260,270,80,90), ITST	XAR1170

210	XAREA=-1.0	XAR1180
	RETURN	XAR1190
220	XAREA=PI*(AH**2-AL**2)	XAR1200
	RETURN	XAR1210
230	XAREA=SUB19(XL,XD,AL,AH)	XAR1220
	RETURN	XAR1230
240	XAREA=SUB06(XL,XD,AL,AH)-PI*AL**2	XAR1240
	RETURN	XAR1250
250	XAREA=SUB21(XL,XD,AL,AH)	XAR1260
	RETURN	XAR1270
260	XAREA=SUB22(XL,XD,AL,AH)	XAR1280
	RETURN	XAR1290
270	XAREA=SUB23(XL,XD,AL,AH)	XAR1300
	RETURN	XAR1310
	END	XAR1320-
	FUNCTION SUB02 (XL,XD,AL,AH)	SU2 10
	ALANG=(ACOS((AL**2+XD**2-XL**2)/(2.0*AL*XD)))*2.0	SU2 20
	OPANG=ACOS((AL**2+XL**2-XD**2)/(2.0*AL*XL))	SU2 30
	XLANG=(ALANG/2.0+OPANG)*2.0	SU2 40
	ASEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0	SU2 50
	BSEGMT=((AL**2)*(ALANG-SIN(ALANG)))/2.0	SU2 60
	SUB02=ASEGMT-BSEGMT	SU2 70
	RETURN	SU2 80
	END	SU2 90-
	FUNCTION SUB03 (XL,XD,AL,AH)	SU3 10
	COMMON /LIST1/ PI	SU3 20
	ALANG=(ACOS((AL**2+XD**2-XL**2)/(2.0*AL*XD)))*2.0	SU3 30
	XLANG=(ACOS((XL**2+XD**2-AL**2)/(2.0*XL*XD)))*2.0	SU3 40
	ASEGMT=(XL**2)*(PI-(XLANG-SIN(XLANG)))/2.0	SU3 50
	BSEGMT=((AL**2)*(ALANG-SIN(ALANG)))/2.0	SU3 60
	SUB03=ASEGMT-BSEGMT	SU3 70
	RETURN	SU3 80
	END	SU3 90-
	FUNCTION SUB05 (XL,XD,AL,AH)	SU5 10
	COMMON /LIST1/ PI	SU5 20
	AHANG=(ACOS((AH**2+XD**2-XL**2)/(2.0*AH*XD)))*2.0	SU5 30
	OPANG=ACOS((AH**2+XL**2-XD**2)/(2.0*AH*XL))	SU5 40
	XLANG=(AHANG/2.0+OPANG)*2.0	SU5 50
	ASEGMT=(XL**2)*(PI-(XLANG-SIN(XLANG)))/2.0	SU5 60
	BSEGMT=((AH**2)*(AHANG-SIN(AHANG)))/2.0	SU5 70
	SUB05=ASEGMT+BSEGMT	SU5 80
	RETURN	SU5 90
	END	SU5 100-
	FUNCTION SUB06 (XL,XD,AL,AH)	SU6 10
	AHANG=(ACOS((AH**2+XD**2-XL**2)/(2.0*AH*XD)))*2.0	SU6 20
	XLANG=(ACOS((XL**2+XD**2-AH**2)/(2.0*XL*XD)))*2.0	SU6 30
	ASEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0	SU6 40
	BSEGMT=((AH**2)*(AHANG-SIN(AHANG)))/2.0	SU6 50
	SUB06=ASEGMT+BSEGMT	SU6 60
	RETURN	SU6 70
	END	SU6 80-

FUNCTION SUB09 (XL,XD,AL,AH)	S09 10
COMMON /LIST1/ PI	S09 20
XLANG=(ACOS((XL**2+XD**2-AL**2)/(2.0*XL*XD)))*2.0	S09 30
DPANG=ACOS((XL**2+AL**2-XD**2)/(2.0*XL*AL))	S09 40
ALANG=(XLANG/2.0+DPANG)*2.0	S09 50
ASEGMT=((AL**2)*(ALANG-SIN(ALANG)))/2.0	S09 60
BSEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0	S09 70
SUB09=>I*(XL**2-AL**2)+ASEGMT-BSEGMT	S09 80
RETURN	S09 90
END	S09 100-
FUNCTION SUB11 (XL,XD,AL,AH)	S11 10
ALANG=(ACDS((AL**2+XD**2-XL**2)/(2.0*AL*XD)))*2.0	S11 20
OPANG=ACDS((AL**2+XL**2-XD**2)/(2.0*AL*XL))	S11 30
XLANG=(ALANG/2.0+OPANG)*2.0	S11 40
ASEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0	S11 50
BSEGMT=((AL**2)*(ALANG-SIN(ALANG)))/2.0	S11 60
AHANG=(ACDS((AH**2+XD**2-XL**2)/(2.0*AH*XD)))*2.0	S11 70
DPANG=ACDS((AH**2+XL**2-XD**2)/(2.0*AH*XL))	S11 80
XLANG=(AHANG/2.0+DPANG)*2.0	S11 90
CSEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0	S11 100
DSEGMT=((AH**2)*(AHANG-SIN(AHANG)))/2.0	S11 110
SUB11=ASEGMT+DSEGMT-BSEGMT-CSEGMT	S11 120
RETURN	S11 130
END	S11 140-
FUNCTION SUB12 (XL,XD,AL,AH)	S12 10
COMMON /LIST1/ PI	S12 20
ALANG=(ACOS((AL**2+XD**2-XL**2)/(2.0*AL*XD)))*2.0	S12 30
XLANG=(ACOS((XL**2+XD**2-AL**2)/(2.0*XL*XD)))*2.0	S12 40
ASEGMT=((XL**2)*(PI-(XLANG-SIN(XLANG)))/2.0)	S12 50
BSEGMT=((AL**2)*(ALANG-SIN(ALANG)))/2.0	S12 60
AHANG=(ACDS((AH**2+XD**2-XL**2)/(2.0*AH*XD)))*2.0	S12 70
OPANG=ACOS((AH**2+XL**2-XD**2)/(2.0*AH*XL))	S12 80
XLANG=(AHANG/2.0+OPANG)*2.0	S12 90
CSEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0	S12 100
DSEGMT=((AH**2)*(AHANG-SIN(AHANG)))/2.0	S12 110
SUB12=ASEGMT+DSEGMT-BSEGMT-CSEGMT	S12 120
RETURN	S12 130
END	S12 140-
FUNCTION SUB13 (XL,XD,AL,AH)	S13 10
ALANG=(ACDS((AL**2+XD**2-XL**2)/(2.0*AL*XD)))*2.0	S13 20
XLANG=(ACDS((XL**2+XD**2-AL**2)/(2.0*XL*XD)))*2.0	S13 30
ASEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0	S13 40
BSEGMT=((AL**2)*(ALANG-SIN(ALANG)))/2.0	S13 50
AHANG=(ACDS((AH**2+XD**2-XL**2)/(2.0*AH*XD)))*2.0	S13 60
XLANG=(ACDS((XL**2+XD**2-AH**2)/(2.0*XL*XD)))*2.0	S13 70
CSEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0	S13 80
DSEGMT=((AH**2)*(AHANG-SIN(AHANG)))/2.0	S13 90
SUB13=CSEGMT+DSEGMT-ASEGMT-BSEGMT	S13 100
RETURN	S13 110
END	S13 120-
FUNCTION SUB16 (XL,XD,AL,AH)	S16 10

COMMON /LIST1/ PI	S16 20
XLANG=(ACOS((XL**2+XD**2-AL**2)/(2.0*XL*XD)))*2.0	S16 30
OPANG=ACOS((XL**2+AL**2-XD**2)/(2.0*XL*AL))	S16 40
ALANG=(XLANG/2.0+OPANG)*2.0	S16 50
ASEGMT=((AL**2)*(ALANG-SIN(ALANG)))/2.0	S16 60
BSEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0	S16 70
AHANG=(ACOS((AH**2+XD**2-XL**2)/(2.0*AH*XD)))*2.0	S16 80
OPANG=ACOS((AH**2+XL**2-XD**2)/(2.0*AH*XL))	S16 90
XLANG=(AHANG/2.0+OPANG)*2.0	S16 100
CSEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0	S16 110
DSEGMT=((AH**2)*(AHANG-SIN(AHANG)))/2.0	S16 120
SUB16=PI*(XL**2-AL**2)+ASEGMT+DSEGMT-BSEGMT-CSEGMT	S16 130
RETURN	S16 140
END	S16 150-
FUNCTION SUB17 (XL, XD, AL, AH)	S17 10
COMMON /LIST1/ PI	S17 20
XLANG=(ACOS((XL**2+XD**2-AL**2)/(2.0*XL*XD)))*2.0	S17 30
OPANG=ACOS((XL**2+AL**2-XD**2)/(2.0*XL*AL))	S17 40
ALANG=(XLANG/2.0+OPANG)*2.0	S17 50
ASEGMT=((AL**2)*(ALANG-SIN(ALANG)))/2.0	S17 60
BSEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0	S17 70
AHANG=(ACOS((AH**2+XD**2-XL**2)/(2.0*AH*XD)))*2.0	S17 80
XLANG=(ACOS((XL**2+XD**2-AH**2)/(2.0*XL*XD)))*2.0	S17 90
CSEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0	S17 100
DSEGMT=((AH**2)*(AHANG-SIN(AHANG)))/2.0	S17 110
SUB17=ASEGMT+CSEGMT+DSEGMT-BSEGMT-PI*AL**2	S17 120
RETURN	S17 130
END	S17 140-
FUNCTION SUB19 (XL, XD, AL, AH)	S19 10
COMMON /LIST1/ PI	S19 20
AHANG=(ACOS((AH**2+XD**2-XL**2)/(2.0*AH*XD)))*2.0	S19 30
OPANG=ACOS((AH**2+XL**2-XD**2)/(2.0*AH*XL))	S19 40
XLANG=(AHANG/2.0+OPANG)*2.0	S19 50
ASEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0	S19 60
BSEGMT=((AH**2)*(AHANG-SIN(AHANG)))/2.0	S19 70
SUB19=PI*(AH**2-AL**2)-ASEGMT+BSEGMT	S19 80
RETURN	S19 90
END	S19 100-
FUNCTION SUB21 (XL, XD, AL, AH)	S21 10
COMMON /LIST1/ PI	S21 20
AHANG=(ACOS((AH**2+XD**2-XL**2)/(2.0*AH*XD)))*2.0	S21 30
OPANG=ACOS((AH**2+XL**2-XD**2)/(2.0*AH*XL))	S21 40
XLANG=(AHANG/2.0+OPANG)*2.0	S21 50
ASEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0	S21 60
BSEGMT=((AH**2)*(AHANG-SIN(AHANG)))/2.0	S21 70
XLANG=(ACOS((XL**2+XD**2-AL**2)/(2.0*XL*XD)))*2.0	S21 80
OPANG=ACOS((XL**2+AL**2-XD**2)/(2.0*XL*AL))	S21 90
ALANG=(XLANG/2.0+OPANG)*2.0	S21 100
CSEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0	S21 110
DSEGMT=((AL**2)*(ALANG-SIN(ALANG)))/2.0	S21 120
SUB21=PI*(AH**2-AL**2)+ASEGMT+DSEGMT-BSEGMT-CSEGMT	S21 130

RETURN	S21 140
END	S21 150-
FUNCTION SUB22 (XL,XD,AL,AH)	S22 10
COMMON /LIST1/ PI	S22 20
AHANG=(ACOS((AH**2+XD**2-XL**2)/(2.0*AH*XD)))*2.0	S22 30
XLANG=(ACOS((XL**2+XD**2-AH**2)/(2.0*XL*XD)))*2.0	S22 40
ASEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0	S22 50
BSEGMT=((AH**2)*(AHANG-SIN(AHANG)))/2.0	S22 60
XLANG=(ACOS((XL**2+XD**2-AL**2)/(2.0*XL*XD)))*2.0	S22 70
OPANG=ACOS((XL**2+AL**2-XD**2)/(2.0*XL*AL))	S22 80
ALANG=(XLANG/2.0+OPANG)*2.0	S22 90
CSEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0	S22 100
DSEGMT=((AL**2)*(ALANG-SIN(ALANG)))/2.0	S22 110
SUB22=ASEGMT+BSEGMT+DSEGMT-CSEGMT-PI*AL**2	S22 120
RETURN	S22 130
END	S22 140-
FUNCTION SUB23 (XL,XD,AL,AH)	S23 10
AHANG=(ACOS((AH**2+XD**2-XL**2)/(2.0*AH*XD)))*2.0	S23 20
XLANG=(ACOS((XL**2+XD**2-AH**2)/(2.0*XL*XD)))*2.0	S23 30
ASEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0	S23 40
BSEGMT=((AH**2)*(AHANG-SIN(AHANG)))/2.0	S23 50
XLANG=(ACOS((XL**2+XD**2-AL**2)/(2.0*XL*XD)))*2.0	S23 60
ALANG=(ACOS((AL**2+XD**2-XL**2)/(2.0*AL*XD)))*2.0	S23 70
CSEGMT=((AL**2)*(ALANG-SIN(ALANG)))/2.0	S23 80
DSEGMT=((XL**2)*(XLANG-SIN(XLANG)))/2.0	S23 90
SUB23=ASEGMT+BSEGMT-CSEGMT-DSEGMT	S23 100
RETURN	S23 110
END	S23 120-
FUNCTION DIST (XLAT,XLNG,YLAT,YLNG)	DIS 10
THIS FUNCTION COMPUTES THE GREAT CIRCLE DISTANCE (IN N.M.) BETWEEN	DIS 20
POINTS X AND Y WITH LATITUDES AND LONGITUDES GIVEN IN TERMS OF	DIS 30
RADIANS (NORTH AND WEST ARE POSITIVE).	DIS 40
DATA PI2/1.57079632/	DIS 50
A=ABS(XLNG-YLNG)	DIS 60
B=PI2-XLAT	DIS 70
C=PI2-YLAT	DIS 80
OIST=3442.2*ACOS(COS(B)*COS(C)+SIN(B)*SIN(C)*COS(A))	DIS 90
RETURN	DIS 100
END	DIS 110-
SUBROUTINE PAGE (N)	PGE 10
DATA LINE/1/	PGE 20
IF (N) 20,50,90	PGE 30
IF (LINE-N-50) 30,40,40	PGE 40
LINE=LINE-N	PGE 50
RETURN	PGE 60
IF (N) 20,50,90	PGE 70
IF (LINE-N-50) 30,40,40	PGE 80
LINE=LINE-N	PGE 90
RETURN	PGE 100
IF (N) 20,50,90	PGE 110
IF (LINE-N-50) 30,40,40	PGE 120
LINE=LINE-N	PGE 130
RETURN	PGE 140

```

5J  IF (LINE) 6J,80,60          PGE 130
60  LINE=0                      PGE 140
7J  PRINT 100                   PGE 150
80  RETURN                       PGE 160
C                                     PGE 170
90  LINE=LINE+N                 PGE 180
    IF (LINE-50) 80,60,60      PGE 190
C                                     PGE 200
C                                     PGE 210
100 FORMAT (1H1)               PGE 220
    END                          PGE 230-
    SUBROUTINE PROCESS (PSIN,ICALN,YIELDH,NCASE,MODE)  PRO 10
    DIMENSION PSIN(1), ICALN(1), YIELDM(1)          PRO 20
    COMMON /DISTIME/ S(99,2),D(99,2),NUMDATA(2),CV(2),TI(2),ITYPE,NTYPEPRO 30
1E,JTYPE,MTYPE,RADMAX          PRO 40
    COMMON /ACFTS/ A(50,2),F(50,2),G(50,2),VS(50,2),VEL(50,2),ACCEL(50PRO 50
1,2),NDATA(2)                  PRO 60
    COMMON /NUCLER/ FLRP(2,4),FTSA(2,4),FLRT(2,4),BURST(2),DISMIN(2),VPRO 70
1PSI(2),VCAL(2),VYIELD(4)      PRO 80
C                                     PRO 90
C THIS SUBROUTINE IS USED TO INPUT DATA FOR AIRCRAFT AND IS USED  PRO 100
C TO GENERATE NUCLEAR EFFECTS GEOMETRY FOR SPECIFIED HARDNESS  PRO 110
C LEVELS VIA SUBROUTINES SABERCM AND SNAPTCH          PRO 120
C                                                     PRO 130
C COMMON /PASS/ SZ,AZ,DELA,AMAX,BETIND,BETA,W,RHO,VIS,PZ,HSL,HTE,HBLPRO 140
1,VLL,ALPHL,TMPL,WUAL,CPL,XLEL,CRAFT,DBURST,MBURST,ATM,CAL,TeffI  PRO 150
C COMMON /PROBLEM/ NEWPROB          PRO 160
C LOGICAL NEWPROB                    PRO 170
C DATA TOL,PERTB1,PERTB2,CMTF,CFTM/0.001,0.001,1.0001,3.2808,0.3048/PRO 180
C DO 210 I=1,NTYPE                  PRO 190
C CALL PAGE (0)                      PRO 200
C WRITE (6,220) NCASE,MODE          PRO 210
C READ 310, AZ,HBL                  PRO 220
C BURST(I)=HBL                      PRO 230
C AZORIG=AZ                          PRO 240
C READ 230, N                        PRO 250
C NDATA(I)=N                        PRO 260
C WRITE (6,240) I,NTYPE             PRO 270
C CAL=FLOAT(ICALN(I))               PRO 280
C DELP=PSIN(I)                      PRO 290
C VPSI(I)=DELP                      PRO 300
C VCAL(I)=CAL                       PRO 310
C WRITE (6,250) I                    PRO 320
C WRITE (6,260) DELP,ICALN(I)       PRO 330
C WRITE (6,270) I                    PRO 340
C WRITE (6,280)                      PRO 350
C CALL PAGE (23)                    PRO 360
C MPTS=0                            PRO 380
C                                     PRO 390
C THE FOLLOWING SECTION READS AIRCRAFT INPUT DATA AND ADJUSTS THE  PRO 400
C ALTITUDE (IF NECESSARY) FOR USE BY ALAG WHICH REQUIRES IT TO BE  PRO 410

```

C	STRICTLY MONOTONICALLY INCREASING. THE ORIGINAL ALTITUDE INPUT	PRO 420
C	DATA IS ASSUMED TO BE MONOTONICALLY INCREASING.	PRO 430
C	AN AUTOMATIC LOOKUP ON THE VELOCITY OF SOUND GIVEN ALTITUDE IS	PRO 440
C	PERFORMED VIA FUNCTION SUBROUTINE SSPF.	PRO 450
C		PRO 460
	READ 290, F(1,I),G(1,I),A(1,I),VEL(1,I),ACCEL(1,I)	PRO 470
	ALT=CFTM*A(1,I)	PRO 480
	VS(1,I)=CMTF*SSPF(ALT)	PRO 490
	J=1	PRO 500
	WRITE (6,300) J,F(1,I),G(1,I),A(1,I),VS(1,I),VEL(1,I),ACCEL(1,I)	PRO 510
	DO 40 J=2,N	PRO 520
	READ 290, F(J,I),G(J,I),A(J,I),VEL(J,I),ACCEL(J,I)	PRO 530
10	IF (A(J,I).GT.A(J-1,I)) GO TO 30	PRO 540
	A(J,I)=PERTB2*A(J-1,I)	PRO 550
20	IF (AZ.EQ.A(J-1,I)) AZ=A(J,I)	PRO 560
	IF (A(J,I).GT.0.0) GO TO 10	PRO 570
	A(J,I)=A(J-1,I)+PERTB1	PRO 580
	GO TO 20	PRO 590
30	ALT=CFTM*A(J,I)	PRO 600
	VS(J,I)=CMTF*SSPF(ALT)	PRO 610
	WRITE (6,300) J,F(J,I),G(J,I),A(J,I),VS(J,I),VEL(J,I),ACCEL(J,I)	PRO 620
	CALL PAGE (1)	PRO 630
40	CONTINUE	PRO 640
	READ 290, FALTCM,FMACH,TI(I),XATI	PRO 670
	READ 310, HTE,BETIND,RHO,VIS,PZ,HSL	PRO 680
	FALT=AZ	PRO 690
	VS1=ALAG(AZ,A(1,I),VS(1,I),N)	PRO 695
	VEL1=ALAG(AZ,A(1,I),VEL(1,I),N)	PRO 700
	IF ((ABS(VEL1-FMACH)/(VEL1+FMACH)/2.0)).LE.0.01) VEL1=FMACH	PRO 720
	V0E=VS1*VEL1	PRO 730
	ACC=ALAG(AZ,A(1,I),ACCEL(1,I),N)	PRO 740
	CALL PAGE (-5)	PRO 750
	WRITE (6,320) FALTCM	PRO 760
	CALL PAGE (0)	PRO 770
	WRITE (6,330) I,AZ	PRO 780
	WRITE (6,340) VS1	PRO 790
	WRITE (6,350) ACC	PRO 800
	IF ((VEL1.LT.FMACH).AND.(XATI.EQ.0.0)) WRITE (6,360) VEL1,VEL1	PRO 810
	IF ((VEL1.GE.FMACH).OR.((VEL1.LT.FMACH).AND.(XATI.GT.0.0))) WRITE	PRO 820
	1(6,360) VEL1,FMACH	PRO 830
	VF=VS1*FMACH	PRO 840
	IF ((VEL1.LT.FMACH).AND.(XATI.EQ.0.0)) WRITE (6,370) V0E,V0E	PRO 850
	IF ((VEL1.GE.FMACH).OR.((VEL1.LT.FMACH).AND.(XATI.GT.0.0))) WRITE	PRO 860
	1(6,370) V0E,VF	PRO 870
	IF (VEL1.LE.FMACH) GO TO 50	PRO 880
	WRITE (6,380)	PRO 890
	STOP	PRO 900
50	CONTINUE	PRO 910
C		PRO 920
C		PRO 930
C	THE VARIABLE DISMIN CONTAINS THE MINIMUM DISTANCE FROM BRAKE	PRO 940

C	RELEASE AT WHICH THE LEVEL-OFF ALTITUDE IS ATTAINED.	PRO 950
C		PRO 960
	DISHIN(I)=ALAG(AZORIG,A(1,I),F(1,I),N)	PRO 970
C		PRO 980
C		PRO 990
C	EQUATIONS OF MOTION	PRO1000
C		PRO1010
	ALT=FALT	PRO1020
	V0=V0E	PRO1030
	L=NCDATA(I)	PRO1040
	ALTCM=FALTCM	PRO1050
	ACCI02=ACC/2.0	PRO1060
	XJJMP=0.0	PRO1070
	IF (VEL1.EQ.FMACH) XJUMP=1.0	PRO1080
	IF ((VEL1.LT.FMACH).AND.(XATI.EQ.0.0)) XJUMP=1.0	PRO1090
	IF (ALT.GE.ALTCM) XJUMP=1.0	PRO1100
	SJ=ALAG(ALT,A(1,I),F(1,I),L)	PRO1110
	T0=ALAG(ALT,A(1,I),G(1,I),L)	PRO1120
	IF (XJUMP.EQ.0.0) GO TO 60	PRO1130
	CV(I)=V0	PRO1140
	GO TO 80	PRO1150
60	T0SQ=T0*T0	PRO1160
	C1=ACC*T0	PRO1170
	V SOUND=ALAG(ALT,A(1,I),VS(1,I),L)	PRO1180
	CV(I)=FMACH*V SOUND	PRO1190
	TF=T0+((CV(I)-V0)/ACC)	PRO1200
	IF (TF.GT.T0) GO TO 70	PRO1210
	WRITE (5,390) T0,TF,CV(I),V0	PRO1220
	STOP	PRO1230
70	DT=(TF-T0)/XATI	PRO1240
80	DO 90 J=1,L	PRO1250
	K=J	PRO1260
	NUMDATA(I)=K	PRO1270
	IF ((F(J,I).GE.S0).OR.(G(J,I).GE.T0)) GO TO 100	PRO1280
	D(J,I)=F(J,I)	PRO1290
	S(J,I)=G(J,I)	PRO1300
90	CONTINUE	PRO1310
100	D(K,I)=S	PRO1320
	S(K,I)=T	PRO1330
	IF (XJJMP.NE.0.0) GO TO 140	PRO1340
110	K=K+1	PRO1350
	S(K,I)=S(K-1,I)+DT	PRO1360
	IF (S(K,I).GE.TF) GO TO 120	PRO1370
	D(K,I)=S0+(V0*(S(K,I)-T0))+(ACCI02*(S(K,I)**2+T0SQ))-(C1*S(K,I))	PRO1380
	GO TO 110	PRO1390
120	S(K,I)=TF	PRO1400
	D(K,I)=S0+(V0*(S(K,I)-T0))+(ACCI02*(S(K,I)**2+T0SQ))-(C1*S(K,I))	PRO1410
	IF (ABS(S(K,I)-S(K-1,I)).LT.TOL) GO TO 130	PRO1420
	NUMDATA(I)=K	PRO1430
	GO TO 140	PRO1440
130	NUMDATA(I)=K-1	PRO1450

140	NUMDATA(I)=NUMDATA(I)+1	PRO1460
	INDEX=NUMDATA(I)	PRO1470
	S(INDEX,I)=S(INDEX-1,I)+TI(I)	PRO1480
	D(INDEX,I)=(CV(I)*TI(I))+D(INDEX-1,I)	PRO1490
	IF (NUMDATA(I).LT.80) GO TO 140	PRO1500
C		PRO1510
	WRITE (6,400)	PRO1520
	CALL PAGE (27)	PRO1530
C	FOLLOWING CONVERTS FEET/SECONDS TO MM/MINUTES	PRO1540
	INDEX=NUMDATA(I)	PRO1550
	DO 150 KK=1,INDEX	PRO1560
	TEMP1=S(KK,I)	PRO1570
	TEMP2=D(KK,I)	PRO1580
	S(KK,I)=S(KK,I)/60.0	PRO1590
	D(KK,I)=D(KK,I)/60.0	PRO1600
	WRITE (6,410) TEMP2,TEMP1,D(KK,I),S(KK,I)	PRO1610
	CALL PAGE (1)	PRO1620
150	CONTINUE	PRO1630
	CV(I)=CV(I)*60.0/6080.0	PRO1640
	TI(I)=TI(I)/60.0	PRO1650
	DISMIN(I)=DISMIN(I)/6080.0	PRO1660
C		PRO1670
	DO 200 J=1,MTYPE	PRO1680
	W=YIELDM(J)	PRO1690
	VYIELD(J)=W	PRO1700
	WRITE (6,420) J,MTYPE,I	PRO1710
	WRITE (6,430) J,W	PRO1720
	READ 440, ISABER,HORF,TSA	PRO1730
	WRITE (6,450)	PRO1740
	WRITE (6,460)	PRO1750
	WRITE (6,470) DELP	PRO1760
	WRITE (6,480) W	PRO1770
	WRITE (6,490) HTE	PRO1780
	WRITE (6,500) HPL	PRO1790
	WRITE (6,510) AZ	PRO1800
	WRITE (6,520)	PRO1810
	IF (ISABER.NE.1) GO TO 160	PRO1820
	WRITE (6,530)	PRO1830
	GO TO 170	PRO1840
150	IPROB1=0	PRO1850
	HORF=TSA=0.0	PRO1860
	CALL SABERCM (SR,AZ,HTE,HBL,W,DELP,HORF,TSA,NCASE,IPROB1)	PRO1870
	IF (IPROB1.NE.1) GO TO 170	PRO1880
	WRITE (6,540)	PRO1890
	HORF=TSA=0.0	PRO1900
170	FLRP(I,J)=HORF/6080.0	PRO1910
	FTSA(I,J)=TSA/60.0	PRO1920
	WRITE (6,550) FLRP(I,J),HORF,FTSA(I,J),TSA	PRO1930
	READ 440, ISNAPT,SZ	PRO1940
	WRITE (6,560)	PRO1950
	WRITE (6,570)	PRO1960

	WRITE (6,580) CAL	PRO1970
	WRITE (6,590) W	PRO1980
	WRITE (6,600) HTE	PRO1990
	WRITE (6,610) HBL	PRO2000
	WRITE (6,620) AZ	PRO2010
	WRITE (6,630) HSL	PRO2020
	WRITE (6,640) PZ	PRO2030
	WRITE (6,650) VIS	PRO2040
	WRITE (6,660) RHO	PRO2050
	WRITE (6,670) BETIND	PRO2060
	WRITE (6,680)	PRO2070
	IF (ISNAPT.NE.1) GO TO 180	PRO2080
	WRITE (6,690)	PRO2090
	GO TO 190	PRO2100
180	IPROB2=0	PRO2110
	SZ=L.0	PRO2120
	CALL SNAPTCH (IPROB2)	PRO2130
	IF ((IPROB2.NE.1).AND.(.NOT.NEWPROB)) GO TO 190	PRO2140
	IF (IPROB2.EQ.1) WRITE (6,710)	PRO2150
	IF (NEWPROB) WRITE (6,710)	PRO2160
	SZ=C.0	PRO2170
190	FLRT(I,J)=SZ/6680.0	PRO2180
	WRITE (6,720) FLRT(I,J),SZ	PRO2190
200	CONTINJE	PRO2200
210	CONTINUE	PRO2210
	RETURN	PRO2220
	C	PRO2230
220	FORMAT (28H SUBROUTINE PROCESS, CASE ,I3,9H, MODE ,I2////)	PRO2240
230	FORMAT (I5)	PRO2250
240	FORMAT (1H ,5X,14HAIRCRAFT TYPE ,I2,5H OF ,I2,9H TYPE(S),///)	PRO2260
250	FORMAT (1H ,10X,41HVULNERABILITY CRITERIA FOR AIRCRAFT TYPE ,I2//)	PRO2270
250	FORMAT (1H ,15X,F5.2,5H PSI/16X,I5,3H GAL/CM,3H**2///)	PRO2280
270	FORMAT (1H ,10X,29HDATA INPUT FOR AIRCRAFT TYPE ,I2//)	PRO2290
280	FORMAT (1H ,16X,4HCARD,10X,6HGROUND,15X,6HFLIGHT,14X,8HAIRCRAFT,13X,8HVELOCITY,11X,4HMACH,5X,9HLEVEL-OFF/16X,6HNUMBER,9X,5HRANGE,17X,2,4HTIME,15X,8HALTITUDE,13X,8HOF SOUND,10X,6HNUMBER,3X,12HACCELERATION/3ION/)	PRO2300
290	FORMAT (3F15.8,2F10.6)	PRO2310
300	FORMAT (1H ,16X,I3,4(5X,E16.8),2(6X,F6.3))	PRO2320
310	FORMAT (6F10.8)	PRO2330
320	FORMAT (2(/,1X),15X,46HINITIAL ALTITUDE (IN CLIMBING) OF MAXIMUM MACH//21X,F7.0,6H FEET)	PRO2340
330	FORMAT (11X,32HDATA COMPUTED FOR AIRCRAFT TYPE ,I2,15H WITH RESP//21X,F7.0,6H FEET//)	PRO2350
340	FORMAT (1H ,15X,17HVELOCITY OF SOUND//21X,F8.2,13H FEET/SECOND//)	PRO2360
350	FORMAT (1H ,15X,22HACCELERATION COMPONENT//21X,F6.3,20H FEET/SECOND//)	PRO2370
360	FORMAT (1H ,15X,12HMACH NUMBERS//21X,9H INITIAL ,F6.3/21X,9HTERMINAL ,F6.3//)	PRO2380
370	FORMAT (1H ,15X,8HVELOCITY//21X,9H INITIAL ,F7.1,13H FEET/SECOND//)	PRO2390
	121X,9HTERMINAL ,F7.1,13H FEET/SECOND//)	PRO2400
		PRO2410
		PRO2420
		PRO2430
		PRO2440
		PRO2450
		PRO2460
		PRO2470

```

380  FORMAT (77H STOP ISSUED - INITIAL VELOCITY (VEL1) GREATER THAN TERPRO2480
MINAL VELOCITY (FHACH)) PRO2490
390  FORMAT (1H1,4HTG= ,E16.8,10X,4HTF= ,E16.8,10X,3HCV=,E16.8,10X,3HV,PRO2500
1=,E16.8//35H DATA INPUT INCORRECT---(CV.LE.V0)) PRO2510
400  FORMAT (17X,39HGROUND RANGE (FEET) FLIGHT TIME (SEC),18X,37HGROUPRO2520
1ND RANGE (NH) FLIGHT TIME (MIN)//) PRO2530
410  FORMAT (20X,E16.8,4X,E16.8,19X,E16.8,4X,E16.8) PRO2540
420  FORMAT (1H1,11X,13HMISSILE TYPE ,I2,5H OF ,I2,32H TYPE(S) AGAINSPRO2550
1T AIRCRAFT TYPE ,I2//) PRO2560
430  FORMAT (1H ,15X,22HYIELD OF MISSILE TYPE ,I2//20X,F6.0,4H KT//) PRO2570
440  FORMAT (I5,2F15.8) PRO2580
450  FORMAT (14(/,1X),14X,16HSUBROUTINE SABERCM/) PRO2590
460  FORMAT (1H ,20X,21HDATA INPUT TO SABERCM) PRO2600
470  FORMAT (1H ,26X,F5.2,26H PSI BLAST OVERPRESSURE) PRO2610
480  FORMAT (1H ,25X,F6.0,15H KT YIELD) PRO2620
490  FORMAT (1H ,25X,F6.0,24H FEET TERRAIN HEIGHT) PRO2630
500  FORMAT (1H ,25X,F6.0,22H FEET BURST HEIGHT) PRO2640
510  FORMAT (1H ,25X,F6.0,27H FEET AIRCRAFT ALTITUDE/) PRO2650
520  FORMAT (1H ,15X,22HSABERCM OUTPUT FOLLOWS) PRO2660
530  FORMAT (39H ***** SABERCM OVERRIDE EFFECTED ***** ,11H ISABER = 1)PRO2670
540  FORMAT (39H ***** SABERCM PROBLEMS DETECTED ***** ,11H IPR0B1 = 1)PRO2680
550  FORMAT (1H-,18X,91H A SUMMARY OF DATA OUTPUT FROM SABERCM USED BYPRO2690
1 SUBROUTINE DETAREA IN COMPUTING LETHAL AREA/26X,29HLETHAL OVERPRE2700
2SSURE RADIUS = ,E16.8,16H NAUTICAL MILES/53X,2H= ,E16.8,6H FEET/PRO2710
331X,24HTIME OF SHOCK ARRIVAL = ,E16.8,9H MINUTES/53X,2H= ,E16.8,9PRO2720
4H SECONDS//) PRO2730
560  FORMAT (1H1,15X,18HSUBROUTINE SNAPTCH/) PRO2740
570  FORMAT (1H ,20X,21HDATA INPUT TO SNAPTCH) PRO2750
580  FORMAT (1H ,26X,F5.0,8H CAL/CM,3H**2,16H THERMAL ENERGY) PRO2760
590  FORMAT (1H ,25X,F6.0,18H KT YIELD) PRO2770
600  FORMAT (1H ,25X,F6.0,27H FEET TERRAIN HEIGHT) PRO2780
610  FORMAT (1H ,25X,F6.0,25H FEET BURST HEIGHT) PRO2790
620  FORMAT (1H ,25X,F6.0,30H FEET AIRCRAFT ALTITUDE) PRO2800
630  FORMAT (1H ,25X,F6.0,36H FEET HAZE LAYER HEIGHT) PRO2810
640  FORMAT (1H ,25X,F6.1,33H MM HG WATER VAPOR PRESSURE) PRO2820
650  FORMAT (1H ,25X,F6.1,44H MILES VISIBILITY(U.S., STATUTE MILEPRO2830
1S)) PRO2840
660  FORMAT (1H ,25X,F6.2,31H (ALBEDO) GROUND REFLECTANCE) PRO2850
670  FORMAT (1H ,25X,F6.2,26H (BETIND) SHOULD BE 1.0/) PRO2860
680  FORMAT (1H ,15X,22HSNAPTCH OUTPUT FOLLOWS) PRO2870
690  FORMAT (39H ***** SNAPTCH OVERRIDE EFFECTED ***** ,11H ISNAPT = 1)PRO2880
700  FORMAT (39H ***** SNAPTCH PROBLEMS DETECTED ***** ,11H IPR0B2 = 1)PRO2890
710  FORMAT (39H ***** SNAPTCH PROBLEMS DETECTED ***** ,17HNEWPROB = .TPRO2900
1RU E.) PRO2910
720  FORMAT (1H-,18X,91H A SUMMARY OF DATA OUTPUT FROM SNAPTCH USED BYPRO2920
1 SUBROUTINE DETAREA IN COMPUTING LETHAL AREA/26X,24HLETHAL THERMALPRO2930
2 RADIUS = ,E16.8,16H NAUTICAL MILES/48X,2H= ,E16.8,6H FEET//) PRO2940
END PRO2950-
FUNCTION SSPF (HM) SSP 10
C SSP 20
C ROUTINE TO COMPUTE SOUND SPEED IN METERS/SECOND AS A FUNCTION OF SSP 30

```

```

C      GEOMETRIC ALTITUDE IN METERS.                      SSP  40
C      BASED ON DATA PRESENTED IN //U.S. STANDARD ATMOSPHERE, 1962//.  SSP  50
C      ROUTINE BY HARRY M. MURPHY, JR., 29APR71, CORRECTED 30CT72 (HMM) SSP  60
C      DIMENSION ALT(10), TM(10), DTDZ(10)                SSP  70
C      DATA ALT/-4996.0,0.0,11019.0,20033.0,32162.0,47350.0,52423.0,61991.0,71991.0,81991.0/
C      DATA TM/320.65,288.15,216.65,216.65,226.65,270.65,270.65,252.65,180.65,180.65/
C      DATA DTDZ/-6.5052E-3,-6.4888E-3,0.0,0.9182E-4,2.7653E-3,0.0,-1.9971E-3,-3.9124E-3,0.0,0.0/
C      Z=HM
C      I=1
C      IF (Z+5000.0) 50,50,20
C      IF (90000.0-Z) 60,60,30
C      DO 40 I=1,9
C      IF (ALT(I+1)-Z) 40,40,50
C      CONTINUE
C      I=10
C      SSPF=20.046796*SQRT(TM(I)+DZ(I)*(Z-ALT(I)))
C      RETURN
C      SSPF=269.44
C      RETURN
C      END
C      FUNCTION DETAREA (Q,DSPT)
C      DIMENSION X1(200), Y1(200), X2(200), Y2(200), X(400), Y(400)
C      COMMON /OISTIME/ S(99,2),D(99,2),NUMDATA(2),CV(2),TI(2),ITYPE,NTYPE
C      1E,JTYPE,MTYPE,RADMAX
C      COMMON /ACFTS/ A(50,2),F(50,2),G(50,2),VS(50,2),VEL(50,2),ACCEL(50,2),
C      1,2),NDATA(2)
C      COMMON /NUCLER/ FLRP(2,4),FTSA(2,4),FLRT(2,4),BURST(2),DISMIN(2),VDET
C      1PSI(2),VCAL(2),VYIELD(4)
C      COMMON /PASS/ SZ,AZ,OELA,AMAX,BETIND,BETA,W,RHO,VIS,PZ,HSL,HTC,HBLDET
C      1,VLL,ALPHL,TMPL,HUAL,CPL,XLEL,CRAFT,JBURST,MBURST,ATM,CAL,TEFFI
C      COMMON /PROBLEM/ NEWPROB
C      LOGICAL NEWPROB
C      DATA PI/3.141592653589793/
C      *****
C      THE LATTER PART OF THIS SUBROUTINE COMPUTES LETHAL AREA FOR A
C      BURST LOCATED Q NAUTICAL MILES FROM THE CENTROID
C      *****

```

C	IF (Q.LT.0.0) STOP	DET 190
	QDNM=Q+DSPT	DET 200
	IF (QDNM.GE.DISHIN(IATYPE)) GO TO 30	DET 210
C	QDFT=608J.0*QDNM	DET 220
	AZ=ALAG(QDFT,F(1,IATYPE),A(1,IATYPE),NDATA(IATYPE))	DET 230
	IF (AZ.LT.HTE) AZ=HTE	DET 240
	MBL=BURST(IATYPE)	DET 250
	WRITE (6,140) Q,DSPT,QDNM,DISHIN(IATYPE)	DET 260
	CAL=VCAL(IATYPE)	DET 270
	DELP=VPSI(IATYPE)	DET 280
	W=VYIELD(JATYPE)	DET 290
C	WRITE (6,150) JATYPE,MTYPE,IATYPE	DET 300
	WRITE (6,160) JATYPE,W	DET 310
	WRITE (6,170)	DET 320
	WRITE (6,180)	DET 330
	WRITE (6,190) DELP	DET 340
	WRITE (6,200) W	DET 350
	WRITE (6,210) HTE	DET 360
	WRITE (6,220) MBL	DET 370
	WRITE (6,230) AZ	DET 380
	WRITE (6,240)	DET 390
	IPROB1=J	DET 400
	HORF=TSA=J.0	DET 410
	CALL SABERCM (SR,AZ,HTE,MBL,W,DELP,HORF,TSA,NCASE,IPROB1)	DET 420
	IF (IPROB1.NE.1) GO TO 10	DET 430
	WRITE (6,250)	DET 440
	HORF=TSA=0.0	DET 450
10	FLRPSI=HORF/6080.0	DET 460
	FTSASI=TSA/60.0	DET 470
	WRITE (6,260) FLRPSI,HORF,FTSASI,TSA	DET 480
	WRITE (6,270)	DET 490
	WRITE (6,280)	DET 500
	WRITE (6,290) CAL	DET 510
	WRITE (6,300) W	DET 520
	WRITE (6,310) HTE	DET 530
	WRITE (6,320) MBL	DET 540
	WRITE (6,330) AZ	DET 550
	WRITE (6,340) MSL	DET 560
	WRITE (6,350) PZ	DET 570
	WRITE (6,360) VIS	DET 580
	WRITE (6,370) RHO	DET 590
	WRITE (6,380) BETIND	DET 600
	WRITE (6,390)	DET 610
	IPROB2=0	DET 620
	SZ=0.0	DET 630
	CALL SNAPTCH (IPROB2)	DET 640
	IF ((IPROB2.NE.1).AND.(.NOT.NENPROB)) GO TO 20	DET 650
	IF (IPROB2.EQ.1) WRITE (6,400)	DET 660
		DET 670
		DET 680
		DET 690

	IF (NEWPROB) WRITE (6,410)	DET 700
	SZ=C.0	DET 710
20	FLRTHM=SZ/6080.0	DET 720
	WRITE (6,420) FLRTHM,SZ	DET 730
C		DET 740
	R=HORF/6080.0	DET 750
	T=TSA/60.0	DET 760
	RHERM2=SZ/6080.0	DET 770
	GO TO 40	DET 780
C		DET 790
C		DET 800
30	R=FLRP(ITYPE,JTYPE)	DET 810
	RHERM2=FLRT(ITYPE,JTYPE)	DET 820
	T=FTSA(ITYPE,JTYPE)	DET 830
40	IF ((Q.EQ.0.0).OR.(R.LE.0.0).OR.(T.LE.0.0)) GO TO 110	DET 840
	INDEX=NUMDATA (ITYPE)	DET 850
	TSPT=ALAG(OSPT,D(1,ITYPE),S(1,ITYPE),INDEX)	DET 860
	IF (Q.GT.R) GO TO 90	DET 870
C		DET 880
C	AT THIS POINT THE VARIABLES R,T,RHERM2,OSPT, AND TSPT ARE KNOWN.	DET 890
C	ALSO IT HAS BEEN ESTABLISHED THAT (Q.GT.0.0),(R.GT.0.0),(T.GT.0.0)	DET 900
C	AND (Q.LE.R)	DET 910
C	NEED TO COMPUTE ACIR, THE DISTANCE FROM THE CENTROID THAT THE	DET 920
C	AIRCRAFT CAN TRAVEL IN TIME T, ASSUMING THAT THE AIRCRAFT WAS	DET 930
C	LOCATED AT THE CENTROID AT THE ONSET OF THE BURST.	DET 940
C		DET 950
	T2=TSPT+T	DET 960
	ACIR=ALAG(T2,S(1,ITYPE),D(1,ITYPE),INDEX)-OSPT	DET 970
	IF (ACIR.GT.0.0) GO TO 50	DET 980
	WRITE (6,430) ACIR	DET 990
	IF (ACIR.LT.0.0) STOP	DET1000
50	ACIRSQ=ACIR*ACIR	DET1010
	IF ((Q+R).LE.ACIR) GO TO 120	DET1020
C		DET1030
C		DET1040
	IF (Q.LT.R) GO TO 70	DET1050
C		DET1060
C		DET1070
C	AT THIS POINT Q.EQ.R	DET1080
C	IF (ACIR.EQ.0.0) GO TO 60	DET1090
C		DET1100
C	AT THIS POINT ACIR.GT.0.0	DET1110
C	ALSO ACIR.LT.(Q+R)=(2.C*R)	DET1120
	XAR=ACIRSQ/(2.0*R)	DET1130
	YAR=SQRT(ACIRSQ-(XAR*XAR))	DET1140
	ALPHA=ATAN(YAR/XAR)	DET1150
	GO TO 100	DET1160
C		DET1170
C		DET1180
60	ALPHA=PI/2.0	DET1190
	GO TO 100	DET1200

C		DET1210
C		DET1220
C	AT THIS POINT Q.LT.R	DET1230
70	IF ((R-Q).GE.ACIR) GO TO 80	DET1240
C		DET1250
C		DET1260
C	AT THIS POINT (R-Q).LT.ACIR.LT.(Q+R)	DET1270
	XAR=((Q*Q)-(R*R)+ACIRSQ)/(2.0*Q)	DET1280
	YAR=SQRT(ACIRSQ-(XAR*XAR))	DET1290
	IF (XAR.GT.0.0) ALPHA=ATAN(YAR/XAR)	DET1300
	IF (XAR.EQ.0.0) ALPHA=PI/2.0	DET1310
	IF (XAR.LT.0.0) ALPHA=PI-ATAN(YAR/(-XAR))	DET1320
	GO TO 100	DET1330
C		DET1340
C		DET1350
C	AT THIS POINT (R-Q).GE.ACIR	DET1360
80	ALPHA=PI	DET1370
	GO TO 100	DET1380
C		DET1390
C		DET1400
C	AT THIS POINT Q.GT.R	DET1410
90	ALPHA=ASIN(R/Q)	DET1420
100	BETA=0.0	DET1430
	RADMAX=R THERM2	DET1440
	CALL EXTRACT (X1,Y1,X2,Y2,IALPHA,R,Q,TSPT,X,Y,NPTS,T,ALPHA,R THERM2	DET1450
	1,BETA)	DET1460
	IF (BETA.EQ.0.0) SECTOR=0.0	DET1470
	IF (BETA.GT.0.0) SECTOR=BETA*R THERM2*R THERM2	DET1480
	IF (BETA.LT.0.0) STOP	DET1490
	IF (NPTS.EQ.0) GO TO 120	DET1500
	IF (NPTS.LT.0) RADMAX=X(-NPTS)-Q	DET1510
	IF (NPTS.LT.0) NPTS=-NPTS	DET1520
	DETAREA=TRISYM(X,Y,NPTS)+SECTOR	DET1530
	RETURN	DET1540
C		DET1550
110	IF (R.GT.R THERM2) GO TO 130	DET1560
120	RADMAX=R THERM2	DET1570
	DETAREA=PI*R THERM2*R THERM2	DET1580
	RETURN	DET1590
C		DET1600
C	AT THIS POINT (R.GT.R THERM2)	DET1610
130	INDEX=NUMDATA(ITYPE)	DET1620
	TSPT=ALAG(DSPT,D(1,ITYPE),S(1,ITYPE),INDEX)	DET1630
	RBOP=BAKUP(R,T,TSPT)	DET1640
	RMAX=AMAX1(R THERM2,RBOP)	DET1650
	RADMAX=RMAX	DET1660
	DETAREA=PI*RMAX*RMAX	DET1670
	RETURN	DET1680
C		DET1690
140	FORMAT (1H1,48H SUBROUTINE DETAREA NUCLEAR LOOKUP - (Q+DSPT) = (,F6.2,3H	DET1700
	1.2,3H + ,F6.2,4H) = ,F6.2,20H NM, WHERE DISHIN = ,F6.2,3H NM/)	DET1710

```

150  FORMAT (1H ,11X,13HMISSILE TYPE ,I2,5H OF ,I2,32H TYPE(S) AGAINSD
1T AIRCRAFT TYPE ,I2/)
160  FORMAT (1H ,15X,22HYIELD OF MISSILE TYPE ,I2/,20X,F6.0,4H KT/)
170  FORMAT (14(/,1X),14X,16HSUBROUTINE SABERCM/)
180  FORMAT (1H ,20X,21HDATA INPUT TO SABERCM)
190  FORMAT (1H ,26X,F5.2,26H PSI      BLAST OVERPRESSURE)
200  FORMAT (1H ,25X,F6.0,15H KT      YIELD)
210  FORMAT (1H ,25X,F6.0,24H FEET    TERRAIN HEIGHT)
220  FORMAT (1H ,25X,F6.0,22H FEET    BURST HEIGHT)
230  FORMAT (1H ,25X,F6.0,27H FEET    AIRCRAFT ALTITUDE/)
240  FORMAT (1H ,15X,22HSABERCM OUTPUT FOLLOWS)
250  FORMAT (39H ***** SABERCM PROBLEMS DETECTED ***** ,11H IPROB1 = 1)
250  FORMAT (1H-,18X,91H A SUMMARY OF DATA OUTPUT FROM SABERCM USED BY
1 SUBROUTINE DETAREA IN COMPUTING LETHAL AREA/26X,29HLETHAL OVERPR
2SSURE RADIUS = ,E16.8,16H NAUTICAL MILES/53X,2H= ,E16.8,6H FEET/
331X,24HTIME OF SHOCK ARRIVAL = ,E16.8,9H MINUTES/53X,2H= ,E16.8,9
4H SECONDS//)
270  FORMAT (1H,15X,18HSUBROUTINE SNAPTGM/)
280  FORMAT (1H ,20X,21HDATA INPUT TO SNAPTGM)
290  FORMAT (1H ,26X,F5.0,8H CAL/CM,3H**2,16H THERMAL ENERGY)
300  FORMAT (1H ,25X,F6.0,18H KT      YIELD)
310  FORMAT (1H ,25X,F6.0,27H FEET    TERRAIN HEIGHT)
320  FORMAT (1H ,25X,F6.0,25H FEET    BURST HEIGHT)
330  FORMAT (1H ,25X,F6.0,30H FEET    AIRCRAFT ALTITUDE)
340  FORMAT (1H ,25X,F6.0,30H FEET    HAZE LAYER HEIGHT)
350  FORMAT (1H ,25X,F6.1,33H MM HG    WATER VAPOR PRESSURE)
360  FORMAT (1H ,25X,F6.1,44H MILES    VISIBILITY(U.S., STATUTE MILE
1S))
370  FORMAT (1H ,25X,F6.2,31H (ALBEDO)  GROUND REFLECTANCE)
380  FORMAT (1H ,25X,F6.2,26H (BETIND)  SHOULD BE 1.0/)
390  FORMAT (1H ,15X,22HSNAPTGM OUTPUT FOLLOWS)
400  FORMAT (39H ***** SNAPTGM PROBLEMS DETECTED ***** ,11H IPROB2 = 1)
410  FORMAT (39H ***** SNAPTGM PROBLEMS DETECTED ***** ,17HNEWPRUB = .T
1RU E.)
420  FORMAT (1H-,18X,91H A SUMMARY OF DATA OUTPUT FROM SNAPTGM USED BY
1 SUBROUTINE DETAREA IN COMPUTING LETHAL AREA/26X,24HLETHAL THERMAL
2 RADIUS = ,E16.8,16H NAUTICAL MILES/48X,2H= ,E16.8,5H FEET//)
430  FORMAT (1H,130(1H*),//,7H ACIR =,E16.8,//,1X,130(1H*))
END
FUNCTION TRITSYM (X,Y,N,
DIMENSION X(1), Y(1)
TRITSYM=0.0
DO 10 J=2,N
TRITSYM=TRITSYM+(X(J-1)*Y(J)) - (X(J)*Y(J-1))
10 CONTINUE
TRITSYM=ABS (TRITSYM)
RETURN
END
SUBROUTINE EXTRACT (X1,Y1,X2,Y2,IALPHA,R,Q,TSPT,X,Y,NPTS,T,ALPHA,REXT
1THERM2,BETA)
DIMENSION RADSQ(4)

```

	DIMENSION X(1), Y(1), X1(1), Y1(1), X2(1), Y2(1)	EXT 40
	DATA PT/3.141592653589793/	EXT 50
	I=:	EXT 60
	K=0	EXT 70
	RSQ=R**2	EXT 80
	QSQ=Q**2	EXT 90
	THETA=0.3	EXT 100
	DELTA=ALPHA/150.0	EXT 110
	RTHERSQ=RTHERM2*RTHERM2	EXT 120
	IF (Q.EQ.R) GO TO 190	EXT 130
	IF (Q.LT.R) GO TO 220	EXT 140
10	I=I+1	EXT 150
	STHET=SIN(THETA)	EXT 160
	CTHET=COS(THETA)	EXT 170
	C1=Q*CTHET	EXT 180
	STSQ=STHET**2	EXT 190
	C1A=RSQ-QSQ*STSQ	EXT 200
	IF (C1A.LT.0.0) GO TO 30	EXT 210
	C2=SQRT(C1A)	EXT 220
	R1=C1+C2	EXT 230
	R2=C1-C2	EXT 240
20	IF (T.GT.0.0) R1C=BAKUP(R1,T,TSPT)	EXT 250
	IF (T.GT.0.0) R2C=BAKUP(R2,T,TSPT)	EXT 260
	IF (T.EQ.0.0) R1C=R1	EXT 270
	IF (T.EQ.0.0) R2C=R2	EXT 280
	X2(I)=R1C*CTHET	EXT 290
	Y2(I)=R1C*STHET	EXT 300
	X1(I)=R2C*CTHET	EXT 310
	Y1(I)=R2C*STHET	EXT 320
	IF (K.EQ.1) GO TO 40	EXT 330
	THETA=THETA+DELTA	EXT 340
	IF (THETA.LT.ALPHA) GO TO 10	EXT 350
	I=I+1	EXT 360
30	K=1	EXT 370
	STHET=SIN(ALPHA)	EXT 380
	CTHET=COS(ALPHA)	EXT 390
	R1=Q*CTHET	EXT 400
	R2=R1	EXT 410
	GO TO 20	EXT 420
40	DO 50 J=1,I	EXT 430
	X(J)=X1(J)	EXT 440
	Y(J)=Y1(J)	EXT 450
	X(I+J)=X2(I-J+1)	EXT 460
	Y(I+J)=Y2(I-J+1)	EXT 470
50	CONTINUE	EXT 480
	M=2*I	EXT 490
60	DO 70 J=1,M	EXT 500
	IF (((X(J)-Q)**2)+(Y(J)**2)).GE.RTHERSQ) GO TO 70	EXT 510
	NPTS=J-1	EXT 520
	IF (NPTS.GE.1) GO TO 100	EXT 530
C	AT THIS POINT (NPTS.EQ.0)	EXT 540

C	THERMAL CIRCLE CONTAINS PETAL LOCUS - RETURN TO DETAREA WITH	EXT 520
C	NPTS, EQ. 0 AND USE THERMAL RADIUS	EXT 560
	RETURN	EXT 570
70	CONTINUE	EXT 580
C		EXT 590
C	AT THIS POINT THERE IS NOT A SINGLE POINT OF THE OVERPRESSURE	EXT 600
C	LOCUS WHICH LIES EITHER ON OR INSIDE OF THE THERMAL CIRCLE.	EXT 610
C	TWO POSSIBILITIES EXIST,	EXT 620
C	(1) THE OVERPRESSURE LOCUS CONTAINS THE THERMAL CIRCLE	EXT 630
C	(2) THE OVERPRESSURE LOCUS DOES NOT CONTAIN, RATHER LIES	EXT 640
C	OUTSIDE OF THE THERMAL CIRCLE	EXT 650
C		EXT 660
C	IF X(M).LE.(Q-RTHERM2) THEN (2) HOLDS	EXT 670
C	IF X(M).GE.(Q+RTHERM2) THEN (1) HOLDS	EXT 680
C		EXT 690
C	IF (X(M).GE.(Q+RTHERM2)) GO TO 90	EXT 700
C		EXT 710
C	AT THIS POINT (2) HOLDS	EXT 720
80	BETA=PI	EXT 730
	NPTS=M	EXT 740
	WRITE (6,390)	EXT 750
	RETURN	EXT 760
C		EXT 770
C		EXT 780
90	BETA=0.0	EXT 790
	WRITE (6,400)	EXT 800
	NPTS=-M	EXT 810
	RETURN	EXT 820
C		EXT 830
100	NR=4	EXT 840
C		EXT 850
C	AT THIS POINT THE OVERPRESSURE LOCUS INTERSECTS THE THERMAL CIRCLE	EXT 860
C	THE POINT GIVEN BY X(NPTS),Y(NPTS) LIES OUTSIDE OF THE THERMAL	EXT 870
C	CIRCLE AND THE POINT GIVEN BY X(NPTS+1),Y(NPTS+1) LIES INSIDE OF	EXT 880
C	THE THERMAL CIRCLE. THE REMAINING TASK IS TO FIND BETA.	EXT 890
C	THE CONDITION HOLDS THAT (1.LE.NPTS.LE.(M-1))	EXT 900
C	MUST ESTABLISH WHETHER OR NOT ((NPTS+3).GT.M)	EXT 910
C		EXT 920
C	IF ((NPTS+3).GT.M) GO TO 140	EXT 930
	RADSQ(4)=((X(NPTS)-Q)**2)+(Y(NPTS)**2)	EXT 940
	RADSQ(3)=((X(NPTS+1)-Q)**2)+(Y(NPTS+1)**2)	EXT 950
	RADSQ(2)=((X(NPTS+2)-Q)**2)+(Y(NPTS+2)**2)	EXT 960
	RADSQ(1)=((X(NPTS+3)-Q)**2)+(Y(NPTS+3)**2)	EXT 970
C		EXT 980
C	IF ((RADSQ(1).EQ.RADSQ(2)).AND.(RADSQ(2).EQ.RADSQ(3))) STOP	EXT 990
	IF (RADSQ(1).EQ.RADSQ(2)) GO TO 110	EXT 1000
	IF (RADSQ(2).EQ.RADSQ(3)) GO TO 130	EXT 1010
C		EXT 1020
	M*1=M+1	EXT 1030
	X(M*1)=X(NPTS+3)	EXT 1040
	Y(M*1)=Y(NPTS+3)	EXT 1050

	X(MP1+1)=X(NPTS+2)	EXT1060
	Y(MP1+1)=Y(NPTS+2)	EXT1070
	X(MP1+2)=X(NPTS+1)	EXT1080
	Y(MP1+2)=Y(NPTS+1)	EXT1090
	X(MP1+3)=X(NPTS)	EXT1100
	Y(MP1+3)=Y(NPTS)	EXT1110
	XINT=ALAG(RTHERSQ,RADSQ,X(MP1),NR)	EXT1120
	YINT=ALAG(RTHERSQ,RADSQ,Y(MP1),NR)	EXT1130
	GO TO 150	EXT1140
C		EXT1150
C	AT THIS POINT RADSQ(1) EQUALS RADSQ(2)	EXT1160
110	X(M+2)=X(NPTS+2)	EXT1170
	Y(M+2)=Y(NPTS+2)	EXT1180
120	X(M+4)=X(NPTS)	EXT1190
	Y(M+4)=Y(NPTS)	EXT1200
	X(M+3)=X(NPTS+1)	EXT1210
	Y(M+3)=Y(NPTS+1)	EXT1220
	X(M+1)=X(NPTS+4)	EXT1230
	Y(M+1)=Y(NPTS+4)	EXT1240
	MP1=M+1	EXT1250
	RADSQ(1)=((X(MP1)-Q)**2)+(Y(MP1)**2)	EXT1260
	XINT=ALAG(RTHERSQ,RADSQ,X(MP1),NR)	EXT1270
	YINT=ALAG(RTHERSQ,RADSQ,Y(MP1),NR)	EXT1280
	GO TO 150	EXT1290
C		EXT1300
C	AT THIS POINT RADSQ(2) EQUALS RADSQ(3)	EXT1310
130	X(M+2)=X(NPTS+3)	EXT1320
	Y(M+2)=Y(NPTS+3)	EXT1330
	RADSQ(2)=RADSQ(1)	EXT1340
	GO TO 120	EXT1350
C		EXT1360
C		EXT1370
140	RADSQ(1)=((X(M)-Q)**2)+(Y(M)**2)	EXT1380
	RADSQ(2)=((X(M-1)-Q)**2)+(Y(M-1)**2)	EXT1390
	RADSQ(3)=((X(M-2)-Q)**2)+(Y(M-2)**2)	EXT1400
	RADSQ(4)=((X(M-3)-Q)**2)+(Y(M-3)**2)	EXT1410
	X(M+1)=X(M-1)	EXT1420
	Y(M+1)=Y(M-1)	EXT1430
	X(M+2)=X(M-2)	EXT1440
	Y(M+2)=Y(M-2)	EXT1450
	X(M+3)=X(M-3)	EXT1460
	Y(M+3)=Y(M-3)	EXT1470
	XINT=ALAG(RTHERSQ,RADSQ,X(M),NR)	EXT1480
	YINT=ALAG(RTHERSQ,RADSQ,Y(M),NR)	EXT1490
C		EXT1500
C	MUST ESTABLISH WHERE THE POINT GIVEN BY XINT,YINT LIES WITH	EXT1510
C	RESPECT TO THE THE POINT GIVEN BY Q,J.0	EXT1520
C		EXT1530
150	IF (I.EQ.1000) GO TO 340	EXT1540
	IF (XINT.LT.Q) GO TO 170	EXT1550
	IF (XINT.GT.Q) GO TO 160	EXT1560

C		EXT1570
C	WE HAVE (XINT.EQ.Q)	EXT1580
	BETA=PI/2.0	EXT1590
	GO TO 183	EXT1600
C		EXT1610
C	WE HAVE (XINT.GT.Q)	EXT1620
160	BETA=ATAN(YINT/(XINT-Q))	EXT1630
	GO TO 183	EXT1640
		EXT1650
C		EXT1660
C	WE HAVE (XINT.LT.Q)	EXT1670
170	BETA=PI-ATAN(YINT/(Q-XINT))	EXT1680
C		EXT1690
C		EXT1700
180	NPTS=NPTS+1	EXT1710
	X(NPTS)=XINT	EXT1720
	Y(NPTS)=YINT	EXT1730
	NPTS=NPTS+1	EXT1740
	X(NPTS)=Q	EXT1750
	Y(NPTS)=0.0	EXT1760
	RETURN	EXT1770
C		EXT1780
C		EXT1790
C	AT THIS POINT Q.EQ.R	EXT1800
C	THE Q.EQ.R ALGORITHM FOLLOWS	EXT1810
C		EXT1820
190	TWOR=2.0*R	EXT1830
200	I=I+1	EXT1840
	STHET=SIN(THETA)	EXT1850
	CTHET=COS(THETA)	EXT1860
	R1=TWOR*CTHET	EXT1870
210	IF (T.GT.0.0) R1C=BAKUP(R1,T,TSPT)	EXT1880
	IF (T.EQ.0.0) R1C=R1	EXT1890
	X1(I)=R1C*CTHET	EXT1900
	Y1(I)=R1C*STHET	EXT1910
C		EXT1920
	IF (K.EQ.1) GO TO 250	EXT1930
	THETA=THETA+OELTA	EXT1940
	IF (THETA.LT.ALPHA) GO TO 230	EXT1950
	I=I+1	EXT1960
	K=1	EXT1970
	STHET=SIN(ALPHA)	EXT1980
	CTHET=COS(ALPHA)	EXT1990
	R1=TWOR*CTHET	EXT2000
	GO TO 210	EXT2010
C		EXT2020
C		EXT2030
C	AT THIS POINT Q.LT.R	EXT2040
C	THE Q.LT.R ALGORITHM FOLLOWS	EXT2050
C		EXT2060
220	I=I+1	EXT2070
	STHET=SIN(THETA)	

	CTHET=COS(THETA)	EXT2080
	C1=Q*CTHET	EXT2090
	C1A=RSQ-(QSQ*STHET*STHET)	EXT2100
	IF (C1A.LT.0.0) GO TO 240	EXT2110
	C2=SQRT(C1A)	EXT2120
	R1=C1+C2	EXT2130
230	IF (T.GT.0.0) R1C=BAKUP(R1,T,TSPT)	EXT2140
	IF (T.EQ.0.0) R1C=R1	EXT2150
	X1(I)=R1C*CTHET	EXT2160
	Y1(I)=R1C*STHET	EXT2170
C		EXT2180
	IF (K.EQ.1) GO TO 250	EXT2190
	THETA=THETA+DELTA	EXT2200
	IF (THETA.LT.ALPHA) GO TO 220	EXT2210
	I=I+1	EXT2220
240	K=1	EXT2230
	STHET=SIN(ALPHA)	EXT2240
	CTHET=COS(ALPHA)	EXT2250
	R1=(Q*CTHET)+SQRT(RSQ-(QSQ*STHET*STHET))	EXT2260
	GO TO 230	EXT2270
C		EXT2280
C		EXT2290
250	K=0	EXT2300
	KM=0	EXT2310
	KP=0	EXT2320
	DO 260 J=1,I	EXT2330
	X(J)=X1(I-J+1)	EXT2340
	Y(J)=Y1(I-J+1)	EXT2350
	DISQ=((X(J)-Q)**2)+(Y(J)**2)	EXT2360
	IF (DISQ.LE.RTHERSQ) X2(J)=-1.0	EXT2370
	IF (DISQ.LE.RTHERSQ) KM=KM+1	EXT2380
	IF (DISQ.GT.RTHERSQ) X2(J)=1.0	EXT2390
	IF (DISQ.GT.RTHERSQ) KP=KP+1	EXT2400
	IF (J.EQ.1) GO TO 260	EXT2410
	IF (X2(J).NE.X2(J-1)) K=K+1	EXT2420
260	CONTINUE	EXT2430
	M=I	EXT2440
C		EXT2450
C		EXT2460
C	THE VARIABLE K CONTAINS THE NUMBER OF TIMES THAT THE OVERPRESSURE	EXT2470
C	LOCUS CROSSES THE THERMAL LOCUS.	EXT2480
C		EXT2490
C	IF KP = (+I) THEN ALL POINTS OF THE OVERPRESSURE LOCUS LIE	EXT2500
C	OUTSIDE OF THE THERMAL CIRCLE (K=0)	EXT2510
C	IF KM = (+I) THEN ALL POINTS OF THE OVERPRESSURE LOCUS LIE	EXT2520
C	EITHER ON OR INSIDE OF THE THERMAL CIRCLE (K=J)	EXT2530
C		EXT2540
C		EXT2550
	IF (RTHERM2.GT.0.0) GO TO 270	EXT2560
C		EXT2570
C	AT THIS POINT RTHERM2=0.0 WHEN ENTERED FROM ABOVE	EXT2580

	NPTS=-I	EXT2590
	BETA=0.0	EXT2600
	RETURN	EXT2610
C		EXT2620
C		EXT2630
C	AT THIS POINT RTHERM2.GT.J.0	EXT2640
270	IF (K.GT.0) GO TO 300	EXT2650
	IF (KM.NE.I) GO TO 290	EXT2660
C		EXT2670
C	AT THIS POINT KM=(+I) WHEN ENTERED FROM ABOVE	EXT2680
C	THE OVERPRESSURE LOCUS CONTRIBUTES LITTLE OR NOTHING	EXT2690
280	NPTS=0	EXT2700
	RETURN	EXT2710
C		EXT2720
C		EXT2730
C	AT THIS POINT KP=(+I) WHEN ENTERED FROM ABOVE	EXT2740
C	THE THERMAL CIRCLE IS CONTAINED BY OR LIES OUTSIDE OF AND DOES NOT	EXT2750
C	CONTAIN THE OVERPRESSURE LOCUS	EXT2760
290	M=I	EXT2770
	IF (X(I).GT.(Q+RTHERM2)) GO TO 9)	EXT2780
	GO TO 80	EXT2790
C		EXT2800
C		EXT2810
300	IF (K.GT.1) GO TO 370	EXT2820
C		EXT2830
C	AT THIS POINT THE MOST TRIVIAL CASES INVOLVING NON-INTERSECTION	EXT2840
C	OF THE LETHAL LOCI HAVE BEEN DETECTED AND ISOLATED.	EXT2850
C	THE NEXT MOST SIMPLE CASE IS THAT OF FIRST ORDER INTERSECTION.	EXT2860
C	FIRST ORDER INTERSECTION OCCURS WHEN THE POSITIVE HALF OF THE	EXT2870
C	OVERPRESSURE LOCUS INTERSECTS THE THERMAL CIRCLE ONLY ONCE.	EXT2880
C	THE POSITIVE HALF OF THE OVERPRESSURE LOCUS HAS BEEN GENERATED	EXT2890
C	ABOVE BY THE Q.LT.R OR THE Q.EQ.R ALGORITHM.	EXT2900
C	THE COORDINATES OF THE OVERPRESSURE LOCUS ARE CONTAINED IN THE	EXT2910
C	ARRAYS X AND Y. THE POINT GIVEN BY COORDINATES X(1),Y(1) IS	EXT2920
C	NEAREST THE CENTROID - THE POINT GIVEN BY X(I),Y(I) IS FARTHEST	EXT2930
C	FROM THE CENTROID.	EXT2940
C		EXT2950
C	FIRST ORDER INTERSECTION OCCURS WHEN K=1	EXT2960
C	FIRST ORDER INTERSECTION MAY BE CONSIDERED BY SENSING THE VALUES	EXT2970
C	CONTAINED IN THE ARRAY X2(J) FOR J=1,2,...,I. THE CONTENTS OF	EXT2980
C	THE ARRAY X2(J) SHOULD RESEMBLE (FOR FIRST ORDER INTERSECTION)	EXT2990
C	ONE OF THE FOLLOWING CASES,	EXT3000
C	CASE (1) +1,+1,...,+1,-1,-1,...,-1	EXT3010
C	CASE (2) -1,-1,...,-1,+1,+1,...,+1	EXT3020
C	CASE (3) +1,-1,-1,...,-1	EXT3030
C	CASE (4) -1,-1,...,-1,+1	EXT3040
C	CASE (5) +1,+1,...,+1,-1	EXT3050
C	CASE (6) -1,+1,+1,...,+1	EXT3060
C		EXT3070
C		EXT3080
C	CASES (3) AND (4), ALL OF OVERPRESSURE LOCUS (EXCEPT FOR ONE POINT)	EXT3090

C	IS EITHER ON OR WITHIN THE THERMAL CIRCLE	EXT3100
	IF (KM.EQ.(I-1)) GO TO 280	EXT3110
C		EXT3120
C		EXT3130
C	CASES (5) AND (6), ALL OF OVERPRESSURE LOCUS (EXCEPT FOR ONE POINT)	EXT3140
C	IS OUTSIDE OF THE THERMAL CIRCLE	EXT3150
	IF (KP.NE.(I-1)) GO TO 310	EXT3160
C		EXT3170
C	THE FOLLOWING IF IS TRUE FOR CASE (6)	EXT3180
	IF ((X2(1).EQ.(-1.0)).AND.(X2(I).EQ.(+1.0))) GO TO 290	EXT3190
C		EXT3200
C	IT IS ESTABLISHED INDIRECTLY THAT WE HAVE CASE (5)	EXT3210
	M=I	EXT3220
	IF (X(I).GE.Q) GO TO 90	EXT3230
	GO TO 80	EXT3240
C		EXT3250
C		EXT3260
C	CASES (1) AND (2) REMAIN TO BE DETECTED FOR K=1	EXT3270
C		EXT3280
C		EXT3290
C	THE FOLLOWING IS TRUE FOR CASE (1)	EXT3300
310	IF ((X2(1).EQ.(+1.0)).AND.(X2(I).EQ.(-1.0))) GO TO 60	EXT3310
C		EXT3320
C		EXT3330
C	AT THIS POINT CASE (2) REMAINS	EXT3340
C	CASE (2) IS ALMOST IDENTICAL TO CASE (1) AFTER INTERCHANGING	EXT3350
C	X(J) WITH X(I-J+1) AND Y(J) WITH Y(I-J+1) FOR J=1,2,...	EXT3360
C	THE COMPUTATION OF BETA IS DIFFERENT THOUGH.	EXT3370
C	VARIABLE I IS SET EQUAL TO 1000 AS A FLAG SO THAT THE BETA	EXT3380
C	COMPUTATION IS PERFORMED AT STATEMENT 390 RATHER THAN AT 9.	EXT3390
	M=I	EXT3400
	DO 320 J=1,I	EXT3410
	IF ((I-J+1).LE.J) GO TO 330	EXT3420
	TEMP=X(J)	EXT3430
	X(J)=X(I-J+1)	EXT3440
	X(I-J+1)=TEMP	EXT3450
	TEMP=Y(J)	EXT3460
	Y(J)=Y(I-J+1)	EXT3470
	Y(I-J+1)=TEMP	EXT3480
320	CONTINUE	EXT3490
330	I=1000	EXT3500
	GO TO 60	EXT3510
C		EXT3520
C		EXT3530
340	IF (XINT.LT.Q) GO TO 360	EXT3540
	IF (XINT.GT.Q) GO TO 350	EXT3550
C		EXT3560
C	WE HAVE (XINT.EQ.0.0)	EXT3570
	BETA=PI/2.0	EXT3580
	GO TO 180	EXT3590
C		EXT3600

C	WE HAVE (XINT.GT.Q)	EXT3610
350	BETA=PI-ATAN(YINT/(XINT-Q))	EXT3620
	GO TO 180	EXT3630
C		EXT3640
C	WE HAVE (XINT.LT.Q)	EXT3650
360	BETA=ATAN(YINT/(Q-XINT))	EXT3660
	GO TO 180	EXT3670
C		EXT3680
C		EXT3690
C	MULTIPLE INTERSECTION	EXT3700
370	WRITE (6,410) K,Q,R,T,RTHERM2	EXT3710
	DO 380 J=1,I	EXT3720
	WRITE (6,420) X(J),Y(J),X2(J)	EXT3730
380	CONTINUE	EXT3740
C		EXT3750
C	THE FOLLOWING IS A TEMPORARY APPROXIMATION FOR MULTIPLE INTERSECT.	EXT3760
	NPTS=C	EXT3770
	RETURN	EXT3780
C		EXT3790
390	FORMAT (75H OVERPRESSURE LOCUS LIES OUTSIDE OF AND DOES NOT CONTAIN	EXT3800
	1N THE THERMAL CIRCLE)	EXT3810
400	FORMAT (70H FROM SUBROUTINE EXTRACT, OVERPRESSURE LOCUS CONTAINS	EXT3820
	1 THERMAL LOCUS :	EXT3830
410	FORMAT (1H-,33HMULTIPLE INTERSECTION WHERE K = ,I3/1H ,4(210.8,5X	EXT3840
	1))	EXT3850
420	FORMAT (1H ,E16.8,5X,E16.8,5X,F5.1)	EXT3860
	END	EXT3870-
	FUNCTION BAKUP (R,T,TSPT)	BAK 10
	COMMON /DISTIME/ S(99,2),D(99,2),NUMDATA(2),CV(2),TI(2),ITYPE,NTYP	BAK 20
	1E,JTYPE,MTYPE,RADMAX	BAK 30
	INDEX=NUMDATA(ITYPE)	BAK 40
	IF (TSPT.LE.S(INDEX,ITYPE)) GO TO 10	BAK 50
C	WRITE(6,5) TSPT,NUMDATA(ITYPE)	BAK 60
	CALL TIMEGEN (TSPT)	BAK 70
10	INDEX=NUMDATA(ITYPE)	BAK 80
	DSPT=ALAG(TSPT,S(1,ITYPE),D(1,ITYPE),INDEX)	BAK 90
	RMAX=DSPT+R	BAK 100
	IF (RMAX.LE.D(INDEX,ITYPE)) GO TO 20	BAK 110
C	WRITE(6,15) RMAX,NUMDATA(ITYPE)	BAK 120
	CALL DATAGEN (RMAX)	BAK 130
20	INDEX=NUMDATA(ITYPE)	BAK 140
	TMAX=ALAG(RMAX,D(1,ITYPE),S(1,ITYPE),INDEX)	BAK 150
	T2=TMAX-T	BAK 160
	IF (T2.GE.TSPT) GO TO 30	BAK 170
C	WRITE(6,25) T2,TSPT	BAK 180
	BAKUP=0.0	BAK 190
	RETURN	BAK 200
30	INDEX=NUMDATA(ITYPE)	BAK 210
	BAKUP=ALAG(T2,S(1,ITYPE),D(1,ITYPE),INDEX)-DSPT	BAK 220
	IF (BAKUP.LT.0.0) BAKUP=0.0	BAK 230
	RETURN	BAK 240

C	END	BAK 250
	SUBROUTINE DATAGEN (R)	BAK 260-
	COMMON /DISTIME/ S(99,2),D(99,2),NUMDATA(2),CV(2),TI(2),ITYPE,NTYPE	DAT 10
	1E,JTYPE,MTYPE,RADMAX	DAT 20
10	NUMDATA(ITYPE)=NUMDATA(ITYPE)+1	DAT 30
	INDEX=NUMDATA(ITYPE)	DAT 40
	S(INDEX,ITYPE)=S(INDEX-1,ITYPE)+TI(ITYPE)	DAT 50
	D(INDEX,ITYPE)=(CV(ITYPE)*TI(ITYPE))+D(INDEX-1,ITYPE)	DAT 60
	IF (R.GT.D(INDEX,ITYPE)) GO TO 10	DAT 70
	RETURN	DAT 80
	END	DAT 90
	SUBROUTINE TIMEGEN (TIME)	DAT 100-
	COMMON /DISTIME/ S(99,2),D(99,2),NUMDATA(2),CV(2),TI(2),ITYPE,NTYPE	TIG 10
	1E,JTYPE,MTYPE,RADMAX	TIG 20
10	NUMDATA(ITYPE)=NUMDATA(ITYPE)+1	TIG 30
	INDEX=NUMDATA(ITYPE)	TIG 40
	S(INDEX,ITYPE)=S(INDEX-1,ITYPE)+TI(ITYPE)	TIG 50
	D(INDEX,ITYPE)=(CV(ITYPE)*TI(ITYPE))+D(INDEX-1,ITYPE)	TIG 60
	IF (TIME.GT.S(INDEX,ITYPE)) GO TO 10	TIG 70
	RETURN	TIG 80
	END	TIG 90
	FUNCTION ALAG (XV,X,Y,NXY)	TIG 100-
C		ALG 10
C	GENERAL PURPOSE FOUR-POINT LAGRANGIAN INTERPOLATION FUNCTION.	ALG 20
C	GIVES RESULT OF INTERPOLATION OF TABLE OF Y AS A FUNCTION OF X	ALG 30
C	AT ENTRY POINT XO.	ALG 40
C	NOTE - X ARRAY MUST BE IN ASCENDING ORDER.	ALG 50
C	FUNCTION BY HARRY M. MURPHY, JR., 30 NOVEMBER 1966.	ALG 60
C		ALG 70
C	DIMENSION X(NXY), Y(NXY)	ALG 80
C		ALG 90
10	L=2	ALG 100
	M=NXY-1	ALG 110
	XO=XV	ALG 120
	IF (2-M) 20,80,80	ALG 130
C		ALG 140
20	I=(L+M)/2	ALG 150
	IF (L-I) 30,70,20	ALG 160
30	IF (X(I)-XO) 50,70,40	ALG 170
40	M=I	ALG 180
	GO TO 20	ALG 190
50	IF (X(I+1)-XO) 60,70,70	ALG 200
60	L=I	ALG 210
	GO TO 20	ALG 220
C		ALG 230
70	ALAG=Y(I-1)*(XO-X(I))*(XO-X(I+1))*(XO-X(I+2))/((X(I-1)-X(I))*(X(I-1)-X(I+1))*(XO-X(I+2))	ALG 240
	+Y(I)*(XO-X(I-1))*(XO-X(I+1))*(XO-X(I+2)))/((X(I)-X(I-1))*(X(I)-X(I+1))*(XO-X(I+2))	ALG 250
	+Y(I+1)*(XO-X(I-1))*(XO-X(I+1))*(XO-X(I+2)))/((X(I+1)-X(I-1))*(X(I+1)-X(I))*(XO-X(I+2))	ALG 260
	+Y(I+2)*(XO-X(I-1))*(XO-X(I))*(XO-X(I+1))/((X(I+2)-X(I-1))*(X(I+2)-X(I+1))*(XO-X(I+2))	ALG 270
	+Y(I+2)*(XO-X(I-1))*(XO-X(I))*(XO-X(I+1))/((X(I+2)-X(I-1))*(X(I+2)-X(I+1))*(XO-X(I+2))	ALG 280
	+Y(I+2)*(XO-X(I-1))*(XO-X(I))*(XO-X(I+1))/((X(I+2)-X(I-1))*(X(I+2)-X(I+1))*(XO-X(I+2))	ALG 290

```

5+2)-X(I))*(X(I+2)-X(I+1)))          ALG 300
RETURN                                  ALG 310
C                                       ALG 320
80  WRITE (6,90) NXY,XO                 ALG 330
    ALAG=0.0                             ALG 340
    RETURN                                 ALG 350
C                                       ALG 360
C                                       ALG 370
90  FORMAT (26H0FUNCTION ALAG ERROR.  N =,I3,7H, ARG =,E12.4/1X)  ALG 380
    END                                    ALG 390-
    SUBROUTINE SNAPTCH (IPROB2)           SNP 10
C                                       SNP 20
C   THE ORIGINAL VERSION, SNAPT, MENTIONED BELOW HAS BEEN MODIFIED BY SNP 30
C   CRAIG E MILLER, CAPT, AFHL (SAB), KIRTLAND AFB, N. MEX., 2471711  SNP 40
C   EXT 2051 TO DETERMINE ONLY THE THERMAL RANGE SOLUTION WHICH IS ONE SNP 50
C   SPECIFIC USE OF THE ORIGINAL MULTIPURPOSE PROGRAM.                SNP 60
C                                       SNP 70
C   PROGRAM SNAPT (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,FILMPR,PUNCH) SNP 80
C                                       SNP 90
C                                       SNP 100
C   SNAPT WEAPON EFFECTS PROGRAM FOR THERMAL ENERGY                   SNP 110
C                                       SNP 120
COMMON /PASS/ SZ,AZ,DELA,AMAX,BETIND,BETA,W,RHO,VIS,PZ,HSL,HTE,HBL  SNP 120
1,VEL,ALPHL,TMPL,WUAL,CPL,XLEL,CRAFT,DBURST,MBURST,ATM,CAL,TEFFI  SNP 130
COMMON /TABLES/ XX(18),YY(5),ZZ(5,18),XXX(12),YYY(10),ZZZ(10,10),XSNP 140
1XXX(19),YYYY(20),ZZZZ(20,19),CX1(13),CX2(8),B(8,13)              SNP 150
COMMON /INFO/ RS,WSA,CANGP,WSH,TD,ANG,FIR,FV,FH,DELU,DELL,THSTU,TASNP 160
1TD,FHBM,SLBR,SAVE,TEFF                                             SNP 170
COMMON /INDCTR/ IND,KO1                                             SNP 180
REAL MBURST                                                         SNP 190
C   *****                                                         SNP 200
C   KOUNT=1                                                         SNP 210
C   1 CALL READ (KOUNT)                                             SNP 220
C   SR=((W/CAL)**0.5)*5280.0                                         SNP 230
C                                       SNP 240
C   FOLLOWING RELATIVIZES A/C ALTITUDE TO SEA LEVEL BY ADDING GROUND  SNP 250
C   HEIGHT                                                         SNP 260
C   AZ=AZ+HTE                                                       SNP 270
C                                       SNP 280
C   DIFF=ABS(AZ-(HBL+HTE))                                          SNP 290
C   IF (SR.LT.DIFF) SR=1.01*DIFF                                    SNP 300
C   SZ=SQRT(SR**2-DIFF**2)                                          SNP 310
C   DELA=AMAX=BETA=0.0                                             SNP 320
C   VEL=ALPHL=TMPL=0.0                                             SNP 330
C   WUAL=CPL=XLEL=0.0                                              SNP 340
C   CRAFT=DBURST=0.0                                               SNP 350
C   MBURST=0                                                        SNP 360
C   ATM=TEFFI=0.0                                                  SNP 370
C   *****                                                         SNP 380
C   IF (TEFFI.EQ.0.) GO TO 10                                       SNP 390
C   TEFF=TEFFI                                                       SNP 400
C   GO TO 20                                                         SNP 410

```

```

13      TEFF=TEFU(W,HBL+HTE)                                SNP 420
20      KO1=0                                                SNP 430
      IND=0                                                  SNP 440
      WRITE (6,30)                                          SNP 450
      WRITE (6,40) CRAFT,WUAL,VEL,AMAX,ALPHL,CPL,SZ,BETA,TMPL,XLEL,AZ,DELTA,SNP 460
1TIND,DELA,CAL                                            SNP 470
      WRITE (6,50) W,ATM,VIS,HBL,DAY,PZ,HTE,RHO,HSL,TEFF,DBURST,MBURST SNP 480
      WRITE (6,60)                                          SNP 490
C      * * * * *                                          SNP 500
C      CALL THERML                                          SNP 510
      CALL THERML (IPROB2)                                  SNP 520
C      * * * * *                                          SNP 530
C      FOLLOWING RELATIVIZES A/C ALTITUDE TO GROUND        SNP 540
      AZ=AZ-HTE                                            SNP 550
C      * * * * *                                          SNP 560
      RETURN                                               SNP 570
C      KOJNT=KOJNT+1                                       SNP 580
C      GO TO 1                                              SNP 590
C
C
C      * * * * *                                          SNP 600
C      * * * * *                                          SNP 610
C      * * * * *                                          SNP 620
C      * * * * *                                          SNP 630
C      * * * * *                                          SNP 640
C      * * * * *                                          SNP 650
C      * * * * *                                          SNP 660
C      * * * * *                                          SNP 670
33      * * * * *                                          SNP 680
40      FORMAT (6X,13HPANEL DATA ,50X,19HRECEIVER PARAMETERS,/,6X,8HCRASNP 690
1FT =,E14.7,10X,8HMTL =,E14.7,9X,8HFTSEC =,E14.7,10X,8HMAXALT SNP 700
2=,E14.7,/,6X,8HALPHAL =,E14.7,10X,8HBSTUL =,E14.7,9X,8H1STHR =,E14.7,10X,8H1SNP 710
314.7,10X,8HTILT =,E14.7,/,6X,8HTMPL =,E14.7,10X,8HXLLEL =,E14.7,10X,8H1SNP 720
4.7,9X,8H1STALT =,E14.7,10X,8HBETAID =,E14.7,/,6X,8HMDALT =,E14.7,10X,8H1SNP 730
510X,8HICAL =,E14.7,/,6X,8HMDALT =,E14.7,/,6X,8HMDALT =,E14.7,10X,8H1SNP 740
59      * * * * *                                          SNP 750
      FORMAT (6X,17HSOURCE PARAMETERS,46X,22HATMOSPHERIC PARAMETERS,/,6X,8HCRASNP 750
1X,8HYIELD =,E14.7,41X,8HATM =,E14.7,10X,8HVISBL =,E14.7,/,6X,8HCRASNP 760
28HMBURST =,E14.7,41X,8HDAY =,E14.7,10X,8HVAPOR =,E14.7,/,6X,8HCRASNP 770
3TARGET =,E14.7,41X,8HALBEDO =,E14.7,10X,8HHAZE =,E14.7,/,6X,8HCRASNP 780
4EFF =,E14.7,6X,8HDBURST =,E14.7,6X,8HMBURST =,E14.7,/,6X,8HCRASNP 790
63      * * * * *                                          SNP 800
      END                                                  SNP 810
      SUBROUTINE THERML (IPROB2)                            THE 10
C      SUBROUTINE THERML                                    THE 20
      COMMON /PASS/ SZ,AZ,DELA,AMAX,BETIND,BETA,W,RHO,VIS,PZ,HSL,HTE,HBLTHE 30
1,VEL,ALPHL,TMPL,WUAL,CPL,XLEL,CRAFT,DBURST,MBURST,ATM,CAL,TEFFI THE 40
      COMMON /NEEDED/ TP6,TP8,TP3,TP7,TP4,TP9,TP5,CON,PSL,TP6 THE 50
      COMMON /INFO/ RS,WSA,CANGP,WSH,TD,ANG,FIR,FV,FW,DELU,DELL,THSTU,TATHE 60
1TG,FHWH,SLBR,SAVE,TEFF THE 70
      COMMON /INDCTR/ IND,KO1 THE 80
      COMMON /CAN/ CON2,SZ1,MOM THE 90
      COMMON /PROBLM/ NEWPROB THE 100
      LOGICAL NEWPROB THE 110

```

```

REAL MBURST
REAL X
DATA TP10,TP9,TP8,TP7,TP6,TP5,TP4,TP3,TP2,TP1,T0,TM1,TM2,TM3,TM4,TM5,TM6,TM7,TM8,TM9,TM10/1.0E+10,1.0E+9,1.0E+8,1.0E+7,1.0E+6,1.0E+5,1.0E+4,1.0E+3,1.0E+2,1.0E+1,1.0E+0,1.0E-1,1.0E-2,1.0E-3,1.0E-4,1.0E-5,1.0E-6,1.0E-7,1.0E-8,1.0E-9,1.0E-10/
WSUBS(X)=2.3*PZ*(1.-10.**(-6.1*X*TM5))
NEWPROB=.FALSE.
C *****
KJNT=0
C *****
CON=-4.57*TM5
SZ1=SZ
DB=ABS(DBURST)
10 IF (AZ.LT.HTE) GO TO 520
TST=HBL/W**(1.0/3.0)
RS=238.1*W**(1.0/3.0)
IF (TST-177.0) 20,30,30
20 THST0=1.0
TAT0=0.0
GO TO 60
30 IF (TST-369.0) 40,40,50
40 THST0=6.21*TP6*((1.0/TST)**3.0-2.0*TM8)
TAT0=1.0-THST0
GO TO 60
50 THST0=0.0
TAT0=1.0
60 IF (TST-183.0) 70,70,80
70 FHBW=3.41*TM3*TST+1.0
GO TO 90
80 FHBW=1.63
90 IF (W-500.0) 100,100,110
100 DELL=1.0
DELU=0.0
GO TO 120
110 DELL=.9312-.2287*ALOG(W*TM3)*.43+29
DELU=1.0-DELL
120 WSH=WSUBS(HTE)
FW=(3./2.)*(1.32/TEFF)
130 IF (DBURST.GT.0.) AZ=HBL
WSA=WSUBS(AZ)
MOM=1
IF (TST-244.0) 140,140,150
140 TD=3000.0+12.3*TST
GO TO 160
150 TD=6000.0
160 FV=1.296*ALOG(TD)*.43429-4.426
FIR=1.322-1.338*TM4*TD-1.170*TM9*TD**2
QA=ABS(CAL)
170 HB=HBL
SLBR=SZ

```

```

THE 120
THE 130
THE 140
THE 150
THE 160
THE 170
THE 180
THE 190
THE 200
THE 210
THE 220
THE 230
THE 240
THE 250
THE 260
THE 270
THE 280
THE 290
THE 300
THE 310
THE 320
THE 330
THE 340
THE 350
THE 360
THE 370
THE 380
THE 390
THE 400
THE 410
THE 420
THE 430
THE 440
THE 450
THE 460
THE 470
THE 480
THE 490
THE 500
THE 510
THE 520
THE 530
THE 540
THE 550
THE 560
THE 570
THE 580
THE 590
THE 600
THE 610
THE 620

```

	CALL UPLOW (HB,SLBR,AL,BL,CBETL,CTHT,QLN)	THE 630
	IF (NEWPROB) RETURN	THE 640
	IF (KO1.EQ.1) GO TO 440	THE 650
	THT=ANG	THE 660
	IF (DELU) 180,190,180	THE 670
180	HB=HBL+RS	THE 680
	CALL UPLOW (HB,SLBR,AU,BU,CBETU,CALP,QNU)	THE 690
	IF (NEWPROB) RETURN	THE 700
	IF (KO1.EQ.1) GO TO 440	THE 710
	ALP=ANG	THE 720
190	CU=QNU*OZLU	THE 730
	CL=QLN*DELL	THE 740
	IF (BETIND) 240,310,200	THE 750
200	IF (DELU.EQ.0.0) GO TO 210	THE 760
	BMU=ATAN(SQRT(1.-CBETU**2)/CBETU)	THE 770
210	BML=ATAN(SQRT(1.-CBETL**2)/CBETL)	THE 780
	IF ((ALP-BMU).LT.1.570796326) GO TO 220	THE 790
	BU=0.	THE 800
	IF ((THT-BML).LT.1.570796326) GO TO 220	THE 810
	BL=0.	THE 820
220	IF (RMO.EQ.0..ANO.AZ.EQ.(HBL+HTE)) GO TO 230	THE 830
	BETA=ATAN((AL*SQRT(1.0-CTHT**2)+BL*SQRT(1.0-CBETL**2)+CU/CL*(AU*SITHE	THE 840
	1N(ALP)+BU*SQRT(1.0-CBETU**2)))/(AL*CTHT+BL*CBETL+CU/CL*(AU*CALP+BU	THE 850
	2*CBETU))	THE 860
	IF (BETA.LT.0.) BETA=BETA+3.14159	THE 870
	GO TO 310	THE 880
230	BETA=1.57295	THE 890
	GO TO 310	THE 900
C	MODIFIED INCIDENT ANGLE PACKAGE	THE 910
C		THE 920
240	ANGLE=60.0/57.3	THE 930
	SIDE=1.0	THE 940
	IF (BETA-1.5708) 260,260,250	THE 950
250	SIDE=2.0	THE 960
260	IF (ALP-ANGLE) 280,280,270	THE 970
270	ALP=(ALP+SIDE*ANGLE)/2.0	THE 980
280	IF (THT-ANGLE) 300,300,290	THE 990
290	THT=(THT+SIDE*ANGLE)/2.0	THE1000
300	CONTINUE	THE1010
C		THE1020
310	IF (OELL.EQ.1.0) GO TO 320	THE1030
	BMU=ATAN(SQRT(1.0-CBETU**2)/CBETU)	THE1040
320	BML=ATAN(SQRT(1.0-CBETL**2)/CBETL)	THE1050
	VERTU=COS(ALP-BETA)	THE1060
	IF (VERTU) 330,330,340	THE1070
330	VERTU=0.0	THE1080
340	VERTL=COS(THT-BETA)	THE1090
	IF (VERTL) 350,350,360	THE1100
350	VERTL=0.0	THE1110
360	QEUD=CU*AU*VERTU	THE1120
	QEUR=CU*BU*(ABS(COS(BMU-BETA)))	THE1130

	QELU=CL*AL*VERTL	THE1140
	QELR=CL*BL*(ABS(COS(BML-BETA)))	THE1150
	QEU=QEUD+QEUR	THE1160
	QEL=QELD+QELR	THE1170
	QE=QEL+QEU	THE1180
	AZ<=AZ/3281.	THE1190
	HBK=HBL/3281.	THE1200
	SZK=SZ/3281.	THE1210
	SZN=SZ*.000164	THE1220
	IF (CAL) 430,430,370	THE1230
370	IF (ABS(QA-QE)-0.0001*QA) 380,480,480	THE1240
380	IF (WUAL.LE.0.) GO TO 400	THE1250
	IF (AZ.GT.(HBL-HTE+RS)) GO TO 390	THE1260
	CALL TEMP (QE,TMP)	THE1270
	GO TO 410	THE1280
390	CALL TEMP (QE,TMP)	THE1290
	GO TO 410	THE1300
400	WRITE (6,530) AZ,HBL,AZK,HBK,SZ,SZK,BETA,SZN,CL,CU,QEL,QELD,QELR,QEU,1	THE1310
	EU,QEUD,QEUR,QA,QE	THE1320
	GO TO 420	THE1330
410	WRITE (6,540) AZ,CL,SZ,CU,QEL,QELD,QELR,QEU,QEUD,QEUR,QA,QE,BETA,1	THE1340
	MP	THE1350
420	MOM=1	THE1360
	GO TO 440	THE1370
430	WRITE (6,550) AZ,CL,SZ,CU,QEL,QELD,QELR,QEU,QEUD,QEUR,QE,BETA	THE1380
	MOM=1	THE1390
440	AZ=AZ+DELA	THE1400
	IF (AZ.GT.AMAX.OR.AZ.EQ.0.) GO TO 450	THE1410
	GO TO 130	THE1420
450	HBL=HBL+DB	THE1430
	IF (HBL.GT.MBURST.OR.HBL.EQ.0.) GO TO 470	THE1440
	IF (TEFFI.GT.0.) GO TO 460	THE1450
	TEFF=TEFU(W,HBL+HTE)	THE1460
460	SZ=SZ1	THE1470
	GO TO 10	THE1480
470	RETURN	THE1490
480	E=.3	THE1500
	SSZ=(SZ*SZ*(QE/QA)**E+(AZ-HBL)*(AZ-HBL)*((QE/QA)**E-1.0))	THE1510
	A=(QE/QA)**E	THE1520
	IF (SSZ.LT.0.) GO TO 490	THE1530
	SZ=SQRT(SSZ)	THE1540
	SZN=SZ*.000164	THE1550
	SZK=SZ/3281.	THE1560
	MOM=MOM+1	THE1570
	IF (MOM.GT.100) GO TO 510	THE1580
	GO TO 170	THE1590
490	IF (AZ.GE.(HBL+HTE)) GO TO 500	THE1600
	WRITE (6,560) AZ	THE1610
C	*****	THE1620
C	SZ=SZ1	THE1630
C	GO TO 42	THE1640

```

IPROB2=1 THE1650
RETURN THE1660
C *** THE1670
500 WRITE (6,570) AZ THE1680
C *** THE1690
C IF (DBURST.LT.0.) GO TO 43 THE1700
IPROB2=1 THE1710
C *** THE1720
RETURN THE1730
510 WRITE (6,580) THE1740
RETURN THE1750
520 WRITE (6,590) AZ THE1760
C *** THE1770
KJNT=KOUNT+1 THE1780
IF (KOUNT.GE.10) STOP THE1790
C *** THE1800
C *** THE1810
AZ=AZ+HTE THE1820
GO TO 13 THE1830
C THE1840
C THE1850
C *** THE1860
C $Y AT CRITICAL PANEL (CAL/CM**2) QL =,E12.5//,1X,135(1H*),THE1870
C $/) THE1880
C *** THE1890
C THE1900
530 FFORMAT (6X,37HRECEIVER ALTITUDE (FT) AZ =,E12.5,5X,27HBTHE1910
1RST ALTITUDE (FT) HBL =,E12.5/34X,9H(KM) AZ =,E12.5,21X,11H(KM) THL1920
2 HBL =,E12.5/6X,37HCALCULATED HORIZONTAL RANGE (FT) SZ =,E12.5,5X,THE1930
330HANGLE BETWEEN LOCAL HORIZONTAL/34X,9H(KM) SZ =,E12.5,5X,35HAND THE1940
4CRITICAL PANEL (RADIAN) BETA =,E12.5/34X,9H(NM) SZ =,E12.5//,6X,5THE1950
52HUNATTENUATED ENERGY IN LOWER PHASE (CAL/CM**2) CL =,E12.5/6X,52THE1960
6HUNATTENUATED ENERGY IN UPPER PHASE (CAL/CM**2) CU =,E12.5//,6X,1THE1970
755HATTENUATED ENERGY IN LOWER PHASE (CAL/CM**2) QLTHE1980
8 =,E12.5,4X,6HQELD =,E12.5,3X,6HQELR =,E12.5//,6X,65HATTENUATED ENERTHE1990
9RGY IN UPPER PHASE (CAL/CM**2) QU =,E12.5,4X,6HQEUTHE2000
10 =,E12.5,3X,6HQEUR =,E12.5//,6X,64HTOTAL FREE FIELD ENERGY ATCRITITHE2010
11CAL PANEL (CAL/CM**2) QA =,E12.5/6X,64HTOTAL ITERATED ENERGTHE2020
12Y AT CRITICAL PANEL (CAL/CM**2) QE =,E12.5/) THE2030
540 FFORMAT (6X,37HAIRCRAFT ALTITUDE (FT) AZ =,E12.5,5X,52HUNTHE2040
1ATTENUATED ENERGY IN LOWER PHASE (CAL/CM**2) CL =,E12.5//,6X,37HCTHE2050
2ALCULATED HORIZONTAL RANGE (FT) SZ =,E12.5,5X,52HUNATTENUATED ENERTHE2060
3GY IN UPPER PHASE (CAL/CM**2) CU =,E12.5//,6X,65HATTENUATED ENERTHE2070
4GY IN LOWER PHASE (CAL/CM**2) QL =,E12.5,4X,6HQELDTHE2080
5 =,E12.5,3X,6HQELR =,E12.5//,6X,65HATTENUATED ENERGY IN UPPER PHASTHE2090
6E (CAL/CM**2) QU =,E12.5,4X,6HQEUD =,E12.5,3X,6HQETHE2100
7UR =,E12.5//,6X,64HTOTAL FREE FIELD ENERGY AT CRITICAL PANEL (CALTHE2110
8/CM**2) QA =,E12.5//,6X,64HTOTAL ITERATED ENERGY AT CRITICALTHE2120
9 PANEL (CAL/CM**2) QE =,E12.5//,6X,66HANGLE BETWEEN LOCALTHE2130
10 HORIZONTAL AND CRITICAL PANEL (RADIAN) BETA =,E12.5//,6X,47H(USETHE2140
11D FOR CALCULATION OF HORIZONTAL RANGE ONLY),//,66HCALCULATED TEMPE2150

```

```

TEMP =,E12.5,/,36H(ASSTHE215.
$UMING HORIZONTAL RECEIVER AT SZ),//,1X,135(1H/),//) THE217L
550 FORMAT (6X,37HAIRCRAFT ALTITUDE (FT) AZ =,E12.5,5X,52HUNTHE2180
1ATTENUATED ENERGY IN LOWER PHASE (CAL/CM**2) CL =,E12.5,/,0X,37HHTHE2190
2ORIZONTAL RANGE (FT) SZ =,E12.5,5X,52HUMATTENUATED ENERTHE220.
3GY IN JPPER PHASE (CAL/CM**2) CJ =,E12.5,//,6X,65HATTENUATED ENERTHE2210
4GY IN LOWER PHASE (CAL/CM**2) QEL =,E12.5,4X,6HQELOTHE2220
5 =,E12.5,3X,6HQELR =,E12.5,/,6X,05HATTENUATED ENERGY IN UPPER PHASTHE223.
6E (CAL/CM**2) QEU =,E12.5,4X,6HQEUD =,E12.5,3X,6HQLTHE2240
7UR =,E12.5,//,6X,64HTOTAL FREE FIELD ENERGY AT CRITICAL PANEL (CALTHE2250
8/CM**2) QE =,E12.5,//,6X,66HANGLE BETWEEN LOCAL HORIZONTAL ATHE2260
9ND CRITICAL PANEL (RADIAN) BETA =,E12.5,//,1X,135(1H/),//) THE227.
560 FORMAT (46H RECEIVER BELOW LOWER LIMIT OF ENVELOPE AT AZ=,F10.0,5HTHE2280
1 FT.) THE2290
570 FORMAT (46H RECEIVER ABOVE UPPER LIMIT OF ENVELOPE AT AZ=,F10.0,5HTHE2300
1 FT.) THE2310
580 FORMAT (21H SZ ITERATIONS GT 100) THE2320
590 FORMAT (26H RECEIVER BELOW TERRAIN AT,F10.0,26H FT,ALTITUDE INCRITHE2330
1MENTED.) THE2340
END THE2350-
SUBROUTINE UPLOW (HB,SBR,A,B,CBET,CANG,QN) UPL 10
COMMON /PASS/ SZ,AZ,DELA,AMAX,BETIND,BETA,W,RHO,VIS,PZ,HSL,HTE,HBLUPL 20
1,VEL,ALPHL,TMPL,WUAL,CPL,XLEL,CRAFT,JBURST,MBURST,ATM,CAL,TEFFI UPL 30
COMMON /INFO/ RS,WSA,CANGP,WSH,TD,ANG,FIR,FV,FW,DELU,DELL,THSTL,TAUPL 40
1TD,FHBN,SLBR,SAVE,TEFF UPL 50
COMMON /NEEDED/ TP6,TM8,TM3,TP7,TM4,TM9,TM5,CON,PSL,TM6 UPL 60
COMMON /CAN/ CON2,SZ1,MOM UPL 70
COMMON /INDCTR/ IND,K01 UPL 80
COMMON /PROBLEM/ NEWPROB UPL 90
LOGICAL NEWPROB UPL 100
REAL MBURST UPL 110
REAL X,Y,Z UPL 120
DSUBL (X)=SLBR**2+(AZ-X)**2 UPL 130
COSAT (Y)=(AZ-Y)/SQRT(DSUBL (Y)) UPL 140
WSUBS (Z)=2.3*PZ*(1.-10.**(-0.1*Z*TM5)) UPL 150
TST=W/W**(1.0/3.0) UPL 160
CANGP=COSAT (HTE) UPL 170
CANG=COSAT (HTE+HB) UPL 180
ANG=ACOS (CANG) UPL 190
IF (ANG) 10,20,20 UPL 200
10 ANG=ANG+3.1416 UPL 210
20 CONTINUE UPL 220
IF (TST-278.0) 30,40,40 UPL 230
30 K=3 UPL 240
GO TO 50 UPL 250
40 K=2 UPL 260
50 QN=QNFN(DSUBL (HTE+HB),CANG,K) UPL 270
TLUP=HTE+HB+.5*RS*CANG UPL 280
CALL PHASE (TH,TV,TPH,TPV,TLUP,CANG) UPL 290
IF (ABS (CANG).LT..001) GO TO 60 UPL 300
WD=(WSA-WSUBS (TLUP))/CANG UPL 310

```

```

GO TO 70
50 MD=1.615*TM4*PZ*(SLBR-RS)*(10.**(-6.1*TM5*TLUP)+10.**(-6.1*TM5*AZ)) UPL 320
1) UPL 330
70 CALL TBLKUP (MD,TD,TM,1) UPL 340
IF (NEWPROB) RETURN UPL 350
IF (ABS(CANG).LT..001) GO TO 80 UPL 360
IF (ABS(CANGP).LT..001) GO TO 100 UPL 370
MD=(WSA-WSH)/CANGP+WSUBS(HB)-WSH UPL 380
GO TO 110 UPL 400
80 IF (ABS(CANGP).LT..001) GO TO 90 UPL 410
MDP=(WSA-WSH)/CANGP+WSUBS(HB)-WSH UPL 420
GO TO 110 UPL 430
90 MDP=1.615*TM4*PZ*((HB*(10.**(-6.1*TM5*HB)+10.**(-6.1*TM5*HTE)))+(SUP UPL 440
1QRT(SLBR*SLBR+(AZ-HTE)*(AZ-HTE))*(10.**(-6.1*TM5*AZ)+10.**(-6.1*TM UPL 450
25*HTE))) UPL 460
GO TO 110 UPL 470
100 MDP=1.615*TM4*PZ*SQRT(SLBR*SLBR+(AZ-HTE)*(AZ-HTE))*(10.**(-6.1*TM5 UPL 480
1*AZ)+10.**(-6.1*TM5*HTE))+WSUBS(HB)-WSH UPL 490
110 CALL TBLKUP (MDP,TD,TMP,1) UPL 500
IF (NEWPROB) RETURN UPL 510
IF (TST-278.0) 120,150,150 UPL 520
120 K=2 UPL 530
CON1=57.295*ANG UPL 540
IF (CON1-90.0) 140,130,130 UPL 550
130 CON1=90.0 UPL 560
140 CON2=SQRT(DSUBL(HTE+HB))/RS UPL 570
IF (CON2.LT.2.) GO TO 200 UPL 580
GO TO 160 UPL 590
150 K=3 UPL 600
CON1=(AZ-HTE)/HB UPL 610
CON2=SLBR/HB UPL 620
160 CALL TBLKUP (CON2,CON1,GAM,K) UPL 630
IF (NEWPROB) RETURN UPL 640
IF (TST-278.0) 180,170,170 UPL 650
170 CBET=COSAT(HTE-HB) UPL 660
GO TO 190 UPL 670
180 CBET=COSAT(HTE-3.0*RS) UPL 680
190 A=FIR*TH*TM+FV*TV UPL 690
B=(FIR*TPH*TMP+FV*TPV)*RHO*GAM/CBET UPL 700
RETURN UPL 710
200 WRITE (6,210) UPL 720
SZ=SZ1 UPL 730
K01=1 UPL 740
RETURN UPL 750
C UPL 760
C UPL 770
C UPL 780
210 FORMAT (20H RANGE LESS THAN 2RS) UPL 790
END UPL 800-
SUBROUTINE PHASE (TH,TV,TPH,TPV,TLUP,CANG) PHA 10
COMMON /PASS/ SZ,AZ,DELA,AMAX,BETIND,BETA,W,RHO,VIS,PZ,HSL,HTE,HBLPHA 20

```

	1,VEL,ALPHL,TMPL,MUAL,CPL,XLEL,CRAFT,DBURST,MBURST,ATM,CAL,TEFFI	PHA 30
	COMMON /INFO/ RS,WSA,CANGP,WSH,TJ,ANG,FIR,FV,FW,DELU,DELL,THSTU,TAPHA	PHA 40
	1T0,FHBM,SLBR,SAVE,TEFF	PHA 50
	COMMON /NEEDED/ TP6,TM8,TM3,TP7,TM4,TM9,TM5,CON,PSL,TM6	PHA 60
	REAL MBURST	PHA 70
	REAL X,Y,Z,ZZ	PHA 80
	EXPON(X,Y,Z,ZZ)=EXP(X*(EXP(CON*Y)-EXP(CON*Z))/ZZ)	PHA 90
	EXPON1(X,Y,Z,ZZ)=EXP(X*(EXP(CON*Y)+EXP(CON*Z))/ZZ)	PHA 100
	IF (ABS(CANG).LE..001) GO TO 80	PHA 110
	IF (T'UP,LT,HSL) GO TO 40	PHA 120
	IF (AZ,LT,TLUP) GO TO 10	PHA 130
	THAZE=1.	PHA 140
	TVIS=EXPON(-.0875,TLUP,AZ,CANG)	PHA 150
	TPHAZE=EXPON(-16.4/VIS,HTE,HSL,CANGP)	PHA 160
	TPVIS=EXPON(-.0875,HSL,AZ,CANGP)	PHA 170
	GO TO 120	PHA 180
10	IF (AZ,LT,HSL) GO TO 20	PHA 190
	THAZE=1.	PHA 200
	TVIS=EXPON(.0875,AZ,TLUP,CANG)	PHA 210
	TPHAZE=EXPON(-16.4/VIS,HTE,HSL,CANGP)	PHA 220
	TPVIS=EXPON(-.0875,HSL,AZ,CANGP)	PHA 230
	GO TO 120	PHA 240
20	IF (ABS(CANGP).LE..001) GO TO 30	PHA 250
	THAZE=EXPON(16.4/VIS,AZ,HSL,CANG)	PHA 260
	TVIS=EXPON(.0875,HSL,TLUP,CANG)	PHA 270
	TPHAZE=EXPON(-16.4/VIS,HTE,AZ,CANGP)	PHA 280
	TPVIS=1.	PHA 290
	GO TO 120	PHA 300
30	THAZE=EXPON(16.4/VIS,AZ,HSL,CANG)	PHA 310
	TVIS=EXPON(.0875,HSL,TLUP,CANG)	PHA 320
	TPHAZE=EXPON1(-3.75*TM4*SLBR/VIS,HTE,AZ,1.)	PHA 330
	TPVIS=1.	PHA 340
	GO TO 120	PHA 350
40	IF (AZ,LT,TLUP) GO TO 60	PHA 360
	IF (AZ,GT,HSL) GO TO 50	PHA 370
	THAZE=EXPON(-16.4/VIS,TLUP,AZ,CANG)	PHA 380
	TVIS=1.	PHA 390
	TPHAZE=EXPON(-16.4/VIS,HTE,AZ,CANGP)	PHA 400
	TPVIS=1.	PHA 410
	GO TO 120	PHA 420
50	THAZE=EXPON(-16.4/VIS,TLUP,HSL,CANG)	PHA 430
	TVIS=EXPON(-.0875,HSL,AZ,CANG)	PHA 440
	TPHAZE=EXPON(-16.4/VIS,HTE,HSL,CANGP)	PHA 450
	TPVIS=EXPON(-.0875,HSL,AZ,CANGP)	PHA 460
	GO TO 120	PHA 470
60	IF (ABS(CANGP).LE..001) GO TO 70	PHA 480
	THAZE=EXPON(16.4/VIS,AZ,TLUP,CANG)	PHA 490
	TVIS=1.	PHA 500
	TPHAZE=EXPON(-16.4/VIS,HTE,AZ,CANGP)	PHA 510
	TPVIS=1.	PHA 520
	GO TO 120	PHA 530

70	THAZE=EXPON(16.4/VIS,AZ,TLUP,CANG)	FHA 540
	TVIS=1.	PHA 550
	TPHAZE=EXPON1(-3.75*TM4*SLBR/VIS,AZ,HTE,1.)	PHA 560
	TPVIS=1.	PHA 570
	GO TO 120	PHA 580
80	IF (ABS(CANGP).LE..001) GO TO 100	PHA 590
	IF ((AZ+TLUP)/2..LT.HSL) GO TO 90	PHA 600
	THAZE=1.	PHA 610
	TVIS=EXPON1(-2.*TM6*(SLBR-RS),AZ,TLUP,1.)	PHA 620
	TPHAZE=EXPON(-16.4/VIS,HTE,HSL,CANGP)	PHA 630
	TPVIS=EXPON(-.0875,HSL,AZ,CANGP)	PHA 640
	GO TO 120	PHA 650
90	THAZE=EXPON1(-3.75*TM4*(SLBR-RS)/VIS,TLUP,AZ,1.)	PHA 660
	TVIS=1.0	PHA 670
	TPHAZE=EXPON(-16.4/VIS,HTE,AZ,CANGP)	PHA 680
	TPVIS=1.0	PHA 690
	GO TO 120	PHA 700
100	IF ((AZ+TLUP)/2..LT.HSL) GO TO 110	PHA 710
	THAZE=1.	PHA 720
	TVIS=EXPON1(-2.*TM6*(SLBR-RS),AZ,TLUP,1.)	PHA 730
	TPHAZE=1.	PHA 740
	TPVIS=EXPON1(-2.*TM6*SLBR,AZ,TLUP,1.)	PHA 750
	GO TO 120	PHA 760
110	THAZE=EXPON1(-3.75*TM4*(SLBR-RS)/VIS,TLUP,AZ,1.)	PHA 770
	TVIS=1.	PHA 780
	TPHAZE=EXPON1(-3.75*TM4*SLBR/VIS,HTE,AZ,1.)	PHA 790
	TPVIS=1.	PHA 800
120	TH=THAZE*.3+.7	PHA 810
	TV=.3*THAZE*TVIS+.7	PHA 820
	TPH=.3*TPHAZE+.7	PHA 830
	TPV=.3*TPHAZE*TPVIS+.7	PHA 840
	RETJRN	PHA 850
	END	PHA 860-
	FUNCTION QNFN (X,Y,K)	QNE 10
	COMMON /PASS/ SZ,AZ,DELA,AMAX,BETIND,BETA,W,RHO,VIS,PZ,HSL,HTE,HBLQNE	QNE 20
	1,VEL,ALPHL,TMPL,WUAL,CPL,XLEL,CRAFT,JBURST,MBURST,ATM,CAL,TEFFI	QNE 30
	COMMON /NEEDED/ TP6,TM6,TH3,TP7,TM4,TM9,TM5,CON,PSL,TM6	QNE 40
	COMMON /INFO/ RS,WSA,CANGP,WSH,TD,ANG,FIR,FV,FW,DELU,DELL,THST,TAQNE	QNE 50
	1TD,FHBW,SLBR,SAVE,TEFF	QNE 60
	REAL MBURST	QNE 70
	TEMP=8.569992*TEFF*W*TP7.*X	QNE 80
	IF (K-2) 20,20,10	QNE 90
10	CUN=2.0/3.0*FW	QNE 100
	GO TO 30	QNE 110
20	CUN=1.0	QNE 120
30	QNFN=TEMP*(TATG+CUN*THST0*FHBW*COS(2.0/3.0*ANG))	QNE 130
	RETURN	QNE 140
	END	QNE 150-
	SUBROUTINE TBLKUP (X,Y,Z,K)	TBL 10
	COMMON /PASS/ SZ,AZ,DELA,AMAX,BETIND,BETA,W,RHO,VIS,PZ,HSL,HTE,HBLTBL	TBL 20
	1,VEL,ALPHL,TMPL,WUAL,CPL,XLEL,CRAFT,JBURST,MBURST,ATM,CAL,TEFFI	TBL 30

```

COMMON /INFO/ RS,WSA,CANGP,WSH,TJ,ANG,FIR,FV,FH,DELU,DELL,THSTG,TATBL 40
1T0,FHBW,SLBR,SAVE,TEFF TBL 50
COMMON /TABLES/ XX(18),YY(5),ZZ(5,18),XXX(12),YYY(10),ZZZ(10,12),XTBL 60
1XXX(19),YYYY(20),ZZZZ(20,19),CX1(13),CX2(8),B(8,13) TBL 70
COMMON /NEEDED/ TP6,TM8,TN3,TP7,TM4,TM9,TM5,CON,PSL,TM6 TBL 80
DIMENSION X1(2) TBL 90
REAL MBURST TBL 100
10 GO TO (20,30,40,50,60), K TBL 110
20 X1(1)=X TBL 120
X1(2)=Y TBL 130
CALL DINT (XX,YY,ZZ,5,18,X1(1),X1(2),Z) TBL 140
GO TO 70 TBL 150
30 X1(1)=X TBL 160
X1(2)=Y TBL 170
CALL DINT (XXX,YYY,ZZZ,10,12,X1(1),X1(2),Z) TBL 180
GO TO 70 TBL 190
40 X1(1)=X TBL 200
X1(2)=Y TBL 210
CALL DINT (XXXX,YYYY,ZZZZ,20,19,X1(1),X1(2),Z) TBL 220
GO TO 70 TBL 230
50 X1(1)=X TBL 240
X1(2)=Y TBL 250
CALL DINT (CX1,CX2,B,8,13,X1(1),X1(2),Z) TBL 260
GO TO 70 TBL 270
60 X1(1)=X TBL 280
X1(2)=Y TBL 290
CALL DINT (XX1,XY1,XZ1,19,2,X1(1),X1(2),Z) TBL 300
70 RETURN TBL 310
END TBL 320-
FUNCTION TEFU (YIELD,HOB) TEF 10
C THIS FUNCTION PROVIDES THE UNATTENUATED THERMAL EFFICIENCY AT THE TEF 20
C SPECIFIC YIELD AND BURST ALTITUDE TEF 30
H=HOB*3.348E-4 TEF 40
TEFU=EXP((-3.5797123034E-1-8.8040573590E-3*H+7.1368010068E-4*H*H-1 TEF 50
1.2548009745E-5*H*H*H+6.4232050535E-8*H*H*H*H)*2.302585093) TEF 60
RETURN TEF 70
END TEF 80-
SUBROUTINE ATMOSP (AZ,TA,PA,CA,SSG,ROSL,ATM) ATP 10
ROSL=.002378 ATP 20
IF (AZ-36089.0) 20,20,30 ATP 30
TA=518.688-.00350616*AZ ATP 40
PA=29.9213*(TA/518.688)**5.2561393 ATP 50
CA=49.040895*SQRT(TA) ATP 60
SSG=(1.0-.00000689*AZ)**(-2.128) ATP 70
GO TO 40 ATP 80
30 TA=389.988 ATP 90
PA=6.8322*EXP(-4.8063618*(10.0**(-5))*(AZ-36089.0)) ATP 100
CA=968.405 ATP 110
SSG=1.8146*EXP(2.4032*(AZ-36089.0)*10.0**(-5.0)) ATP 120
40 CONTINUE ATP 130
RETURN ATP 140

```

	END	ATP 150-
	SUBROUTINE TEMP (Q,TIP)	TEM 10
C	CALCULATES MAXIMUM TEMPERATURE FOR A HORIZONTAL PANEL SUBJECTED TO	TEM 20
C	A TOTAL FLUX OF Q CAL/CM**2 IMPINGING WITH ANGLE BETA	TEM 30
	COMMON /PASS/ SZ,AZ,DELA,AMAX,BETIND,BETA,W,RHO,VIS,PZ,HSL,HTE,MBL	TEM 40
	1,VEL,ALPHL,TMPL,WUAL,CPL,XLEL,CRAFT,JBURST,MBURST,ATH,CAL,TEFFI	TEM 50
	COMMON /NEEDED/ TP6,TH8,TH3,TP7,TH4,TH9,TH5,CON,PSL,TH6	TEM 60
	REAL MBURST	TEM 70
	IF (BETA.GT.1.57) GO TO 40	TEM 80
	CALL ATMOSP (AZ,TA,PA,CA,SSGA,ROSL,ATH)	TEM 90
	SMLM=ABS(VEL/CA)	TEM 100
	TBL=TA*(1+.18*SMLM**2)	TEM 110
	TF=(TBL+TA)/2.0	TEM 120
	HA=5.46667*TH3*(SMLM*PA)**.8*TA**.4/(XLEL**.2*TF**.545)	TEM 130
	TAJ=.248*HA*SQRT(W)/(CPL*WUAL)	TEM 140
	IF (TAU-.2) 10,10,20	TEM 150
10	TPMAX=1.0134-.7147*TAU	TEM 160
	GO TO 30	TEM 170
20	TPMAX=.627-.362*ALOG(TAU)*.43429	TEM 180
30	QE=Q*COS(BETA)	TEM 190
	QAA=QE*ALPHL	TEM 200
	TH=TBL+QAA*TPMAX/(.27115*WUAL*CPL)	TEM 210
	RETURN	TEM 220
40	WRITE (6,53)	TEM 230
	RETURN	TEM 240
C		TEM 250
C		TEM 260
C		TEM 270
50	FORMAT (22H RECEIVER BELOW SOURCE)	TEM 280
	END	TEM 290-
	SUBROUTINE DINT (XX,YY,ZZ,M,N,X,Y,Z)	DIN 10
C	LINEAR 2 DIMENSIONAL INTERPOLATION	DIN 20
C	EXAMPLE-GIVEN THE FOLLOWING TABLE,AND X AND Y VALUES,WE CAN	DIN 30
C	1INTERPOLATE FOR APPROPRIATE Z VALUE	DIN 40
C	XX AND YY MUST BE IN INCREASING ORDER	DIN 50
C	X1  X2  X3	DIN 60
C	1Y1 Z11 Z12 Z13	DIN 70
C	2Y2 Z21 Z22 Z23	DIN 80
C	3Y3 Z31 Z32 Z33	DIN 90
	COMMON /PROBLEM/ NEMPROB	DIN 100
	DIMENSION XX(N), YY(M), ZZ(M,N)	DIN 110
	LOGICAL NEMPROB	DIN 120
	IF (((X.LT.XX(1)).OR.(X.GT.XX(N))).OR.((Y.LT.YY(1)).OR.(Y.GT.YY(M)	DIN 130
	1))) GO TO 70	DIN 140
	MM=M-1	DIN 150
	NN=N-1	DIN 160
	DO 20 I=1,MM	DIN 170
	IF ((YY(I).LE.Y).AND.(YY(I+1).GE.Y)) GO TO 10	DIN 180
	GO TO 20	DIN 190
10	C1=(Y-YY(I))/(YY(I+1)-YY(I))	DIN 200
	I1=I	DIN 210

```

I2=I+1
GO TO 30
20 CONTINUE
30 DO 50 J=1,NN
IF ((XX(J).LE.X).AND.(XX(J+1).GE.X)) GO TO 40
GO TO 50
40 C2=(X-XX(J))/(XX(J+1)-XX(J))
J1=J
J2=J+1
GO TO 60
50 CONTINUE
60 CONTINUE
Z1=ZZ(I1,J1)+C1*(ZZ(I2,J1)-ZZ(I1,J1))
Z2=ZZ(I1,J2)+C1*(ZZ(I2,J2)-ZZ(I1,J2))
Z=Z1+C2*(Z2-Z1)
REIJRN
70 WRITE (6,80) X,Y
NEWPROB=.TRUE.
RETURN

C
C
C
80 FORMAT (1H1,26HPARAMETER OUTSIDE RANGE,X=,E15.8,2X,2HY=,E15.6)
END
BLOCK DATA
COMMON /TABLES/ XX(18),YY(5),ZZ(5,18),XXX(12),YYY(10),ZZZ(10,12),XBD1
1XXX(19),YYYY(20),ZZZZ(20,19),CX1(13),CX2(8),B(8,13)
DATA XX/0.,2.00000E+00,4.00000E+00,6.00000E+00,8.00000E+00,1.00000E+01,
1E+01,1.50000E+01,2.00000E+01,2.50000E+01,3.00000E+01,4.00000E+01,5B01
2.00000E+01,7.00000E+01,9.00000E+01,1.10000E+02,1.50000E+02,3.00000E+02,
3E+02,1.00000E+03/
DATA YY/2.00000E+03,3.00000E+03,4.00000E+03,5.00000E+03 6.00000E+03
13/
DATA ZZ/8.45000E-01,9.45000E-01,9.43000E-01,9.70000E-01,9.75000E-01,10.
11,7.40000E-01,8.40000E-01,8.80000E-01,8.82000E-01,9.10000E-01,6.83801
2000E-01,7.84000E-01,8.17000E-01,8.43000E-01,8.62000E-01,8.93000E-01,
31,7.59000E-01,8.03000E-01,8.20000E-01,8.39000E-01,8.30000E-01,7.38801
4000E-01,7.81000E-01,7.98000E-01,8.13000E-01,6.10000E-01,7.18000E-01,
51,7.60000E-01,7.80000E-01,8.00000E-01,5.75000E-01,6.80000E-01,7.22801
6000E-01,7.45000E-01,7.60000E-01,5.51000E-01,6.50000E-01,6.98000E-01,
71,7.17000E-01,7.33000E-01,5.35000E-01,6.28000E-01,6.74000E-01,6.98801
8000E-01,7.15000E-01,5.18000E-01,6.11000E-01,6.48000E-01,6.78000E-01,
91,6.98000E-01,4.94000E-01,5.81000E-01,6.23000E-01,6.48000E-01,6.62801
0000E-01,4.72000E-01,5.59000E-01,6.01000E-01,6.21000E-01,6.38000E-01,
$1,4.28000E-01,5.19000E-01,5.58000E-01,5.79000E-01,5.92000E-01,4.06801
0000E-01,4.82000E-01,5.20000E-01,5.41000E-01,5.55000E-01,3.80000E-01,
$1,4.52000E-01,4.90000E-01,5.10000E-01,5.21000E-01,3.40000E-01,4.02801
0000E-01,4.40000E-01,4.55000E-01,4.65000E-01,2.40000E-01,2.60000E-01,
$1,2.90000E-01,2.50000E-01,2.70000E-01,1.00000E-01,1.00000E-01,1.00801
0000E-01,1.00000E-01,1.00000E-01/
DATA XXX/2.00000E+00,3.00000E+00,5.00000E+00,7.50000E+00,1.00000E+01,

```



\$-01,2.49000E-01,2.70000E-01,2.96000E-01,3.29000E-01,3.90000E-01,4.800 790  
\$62500E-01,5.84800E-01,7.03700E-01,9.06200E-01,1.08500E+00,1.19100E+00 800  
\$+00,1.32600E+00,1.41000E+00,1.54200E+00/ 801 810  
DATA (ZZZZ(I),I=96,190)/1.64000E+00,1.68600E+00,1.73700E+00,1.78000E+00 820  
10E+00,1.81000E+00,1.96200E-01,2.00000E-01,2.17700E-01,2.36000E-01,801 830  
22.67200E-01,3.26500E-01,3.95700E-01,4.26300E-01,6.35000E-01,8.60000E+00 840  
30E-01,1.03800E+00,1.18000E+00,1.29200E+00,1.38100E+00,1.52800E+00,801 850  
41.61000E+00,1.67200E+00,1.72200E+00,1.77000E+00,1.79500E+00,1.64400E+00 860  
50E-01,2.71000E-01,1.80000E-01,2.00000E-01,2.20100E-01,2.73700E-01,801 870  
63.34400E-01,4.60500E-01,5.72200E-01,8.00000E-01,9.81100E-01,1.12700E+00 880  
70E+00,1.24100E+00,1.34100E+00,1.48600E+00,1.57900E+00,1.64900E+00,801 890  
81.70600E+00,1.75300E+00,1.78400E+00,1.24000E-01,1.29000E-01,1.34000E+00 900  
90E-01,1.46000E-01,1.59900E-01,1.96500E-01,2.42900E-01,3.44000E-01,801 910  
\$4.55000E-01,6.67400E-01,8.52700E-01,1.00600E+00,1.13300E+00,1.24900E+00 920  
\$0E+00,1.49900E+00,1.52000E+00,1.60000E+00,1.66300E+00,1.72100E+00,801 930  
\$1.76000E+00,9.95000E-02,1.00000E-01,1.06500E-01,1.15000E-01,1.23700E+00 940  
\$0E-01,1.48500E-01,1.82000E-01,2.62000E-01,3.53600E-01,5.45700E-01,801 950  
\$7.26900E-01,8.84100E-01,1.02200E+00,1.14000E+00,1.32100E+00,1.45100E+00 960  
\$0E+00,1.54700E+00,1.62100E+00,1.69000E+00,1.73500E+00,8.30000E-02,801 970  
\$8.50000E-02,8.90000E-02,9.10000E-02,1.00000E-01,1.17000E-01,1.43000E+00 980  
\$0E-01,2.05000E-01,2.78000E-01,4.41000E-01/ 801 990  
DATA (ZZZZ(I),I=191,285)/6.10000E-01,7.68000E-01,9.08000E-01,1.02800E+00 1000  
100E+00,1.22800E+00,1.37000E+00,1.47800E+00,1.56200E+00,1.64900E+00,801 1010  
2,1.70100E+00,7.12000E-02,7.20000E-02,7.56000E-02,8.00000E-02,8.46000E+00 1020  
300E-02,9.82000E-02,1.16600E-01,1.64700E-01,2.24900E-01,3.66400E-01,801 1030  
4,5.17300E-01,6.65400E-01,8.00000E-01,9.24400E-01,1.13100E+00,1.24800E+00 1040  
500E+00,1.40100E+00,1.50300E+00,1.60900E+00,1.66300E+00,1.74000E+00,801 1050  
6,6.30000E-02,6.50000E-02,6.90000E-02,7.20000E-02,8.40000E-02,9.80000E+00 1060  
700E-02,1.38000E-01,1.85000E-01,3.07000E-01,4.39000E-01,5.71000E-01,801 1070  
8,7.06000E-01,8.30000E-01,1.04000E+00,1.20300E+00,1.32900E+00,1.43200E+00 1080  
900E+00,1.54300E+00,1.61700E+00,5.55000E-02,5.60000E-02,5.67000E-02,801 1090  
\$,6.00000E-02,6.43000E-02,7.23000E-02,8.38000E-02,1.15300E-01,1.35000E+00 1100  
\$00E-01,2.60000E-01,3.74500E-01,4.98000E-01,6.20000E-01,7.38300E-01,801 1110  
\$,9.46700E-01,1.12000E+00,1.25200E+00,1.37000E+00,1.48100E+00,1.56700E+00 1120  
\$00E+00,4.98000E-02,5.00000E-02,5.05000E-02,5.20000E-02,5.65000E-02,801 1130  
\$,6.90000E-02,7.25000E-02,9.90000E-02,1.31000E-01,2.19000E-01,3.21000E+00 1140  
\$00E-01,4.32000E-01,5.50000E-01,6.56000E-01,8.61000E-01,1.04000E+00,801 1150  
\$,1.18100E+00,1.30000E+00,1.43700E+00,1.51000E+00,4.54000E-02,4.54000E+00 1160  
\$00E-02,4.54000E-02,4.90000E-02,5.08000E-02/ 801 1170  
DATA (ZZZZ(I),I=286,380)/5.05000E-02,6.46000E-02,8.62000E-02,1.14800E+00 1000  
100E-01,1.89500E-01,2.79900E-01,3.80200E-01,4.88000E-01,5.89800E-01,801 1010  
2,7.87400E-01,9.61400E-01,1.11800E+00,1.23400E+00,1.35900E+00,1.46300E+00 1020  
300E+00,3.64000E-02,3.90000E-02,4.00000E-02,4.10000E-02,4.30000E-02,801 1030  
4,4.60000E-02,5.60000E-02,6.70000E-02,8.90000E-02,1.42000E-01,2.18000E+00 1040  
500E-01,2.99000E-01,3.88000E-01,4.82000E-01,6.52000E-01,8.26000E-01,801 1050  
6,9.83000E-01,1.10900E+00,1.23800E+00,1.35000E+00,3.33000E-02,3.30000E+00 1060  
700E-02,3.30000E-02,3.40000E-02,3.37000E-02,3.93000E-02,4.37000E-02,801 1070  
8,5.56000E-02,7.17000E-02,1.15200E-01,1.71400E-01,2.37200E-01,3.08000E+00 1080  
900E-01,3.88100E-01,5.46900E-01,7.04200E-01,8.52000E-01,9.81500E-01,801 1090  
\$,1.12300E+00,1.24100E+00,2.86000E-02,2.80000E-02,2.80000E-02,2.80000E+00 1100  
\$00E-02,2.90000E-02,3.30000E-02,3.60000E-02,4.10000E-02,5.80000E-02,801 1120



```

*****SAB 210
COMMON /TRI/ PZ,PG,PB,HZ,HG,HB,PZR,PGR,PBR,W,SR,FR,SRE,ALP1ER,ALP1SAB 210
1E,R90,PBPZ,R,PBRW,INCOMP SAB 220
COMMON /SENSE/ CFS SAB 230
COMMON /COLO/ CF11(2),CF22(2),CF33(2),CF44(2),CF55(2),CFR(6),CF1(7)SAB 240
1),CF2(7),CF3(7) SAB 250
COMMON /TAB/ TAB1I(69),TAB1D(69),TAB2I(62),TAB2D(62),TAB3I(69),TAB3D(69),TAB4I(18),TAB4D(18),TAB5I(26),TAB5D(26),TAB6I(69),TAB6D(69)SAB 260
COMMON /CON/ UL2(7),UL3(8),UL4(8),UL5(5),C4(7),C5(7),C6(7),C7(6),CSAB 280
18(8),C9(8),C10(8),C11(8),C12(8),P2(8),P3(8),A1E(41),ACF(13),CF(13)SAB 290
COMMON /PNT/ RA,STO,PHIR,SHB,ST,XKK,ALTSRG,HT,XKKX SAB 300
COMMON /OVP/ DELP,DELPD,DELPR,CONT,NIT,XITER SAB 310
COMMON /TBLKUP/ L1,LF,NA(16),XL(13),NNEX SAB 320
DIMENSION P(3), H(3), PR(3), TEMP(3), TH(3), RHO(3), SS(3), SUBS(15)SAB 330
12) SAB 340
DIMENSION IHED1(5,3), IHED2(3,2), ID(28), STP(130), HTP(130) SAB 350
DIMENSION ID1(2) SAB 360
EQUIVALENCE (P,PZ), (H,HZ), (PR,PZR), (RHOZ,RHO(1)), (RHOG,RHO(2))SAB 370
1, (RHOB,RHO(3)), (SSZ,SS(1)), (SSG,SS(2)), (SSB,SS(3)) SAB 380
DATA (IHED1(J),J=1,5)/50H OVERPRESSURE SOLUTION SAB 390
1 / SAB 400
DATA (IHED1(J),J=6,10)/50H TRIPLE POINT PATH SOLUTION SAB 410
1 / SAB 420
DATA (IHED1(J),J=11,15)/50H RANGE SOLUTION SAB 430
1 / SAB 440
DATA (IHED2(J),J=1,3)/30H SAB 450
DATA (IHED2(J),J=4,6)/30H (WITH TEMPERATURE CHANGE) / SAB 460
DATA (ID(J),J=1,28)/2HSR,2MHZ,2HHG,2HHB,1HW,3HTSA,2HFR,3HSFV,4HUELSAB 470
1P,3HPMV,3HPOD,5HSDLP,4HRBAR,1HR,4HALFA,3HSRE,5HALP1E,3HSTO,2HRA,2SAB 480
2HST,2HHT,5HPOGOP,4HPDMV,4HPDOD,1HQ,4HRHOZ,3HSSZ,3HCF / SAB 490
DATA (IU1(J),J=1,2)/4HHORF,4HHORN/ SAB 500
DATA CF11/.599829717,.59990265538/ SAB 510
DATA CF22/-.813121069,-.83454567392/ SAB 520
DATA CF33/.0537960684,.20153883184/ SAB 530
DATA CF44/.178602593,-.042235077752/ SAB 540
DATA CF55/.137932981,.038268822887/ SAB 550
DATA CFS/1.85/ SAB 560
DATA CFR/.5,.8,1.2,1.6,3.0,10./ SAB 570
DATA CF1/0.,-.6444,-.1625,-.05,-.007,.000111,0./ SAB 580
DATA CF2/.6499,1.2943,.4725,.2051,.0424,-.003466,0./ SAB 590
DATA CF3/.007076,-.154024,.195,.354,.50408,.57704,.591/ SAB 600
*****SAB 610
DATA IDOIT/0/ SAB 620
*****SAB 630
INTEGER SAVCASE SAB 640
*****SAB 650
SR=SRCH SAB 660
HZ=HZCM+HGCM SAB 670
HG=HGCM SAB 680
HB=HBCH+HGCM SAB 690
W=WCM SAB 700

```

```

C   DELP=DELPCH                                *SAB 710
* * * * *                                     *SAB 720
NPROB=0                                         *SAB 730
R10=10.0/57.296                               *SAB 740
R90=90.0/57.296                               *SAB 750
R130=130.0/57.296                             *SAB 760
C   * * * * *                                     *SAB 770
IF (IDJIT.NE.0) GO TO 20                       *SAB 780
C   * * * * *                                     *SAB 790
DO 10 J=1,18                                   *SAB 800
TAB4I(J)=TAB4I(J)/57.296                       *SAB 810
10  TAB4D(J)=TAB4D(J)/57.296                   *SAB 820
C   * * * * *                                     *SAB 830
20  IDJIT=1                                     *SAB 840
C   * * * * *                                     *SAB 850
NN=70                                           *SAB 860
DO 30 J=1,69                                   *SAB 870
N=NN-J                                          *SAB 880
30  TAB3I(J)=TAB1D(N)                          *SAB 890
40  TAB3D(J)=TAB1I(N)                          *SAB 900
C   PMVD=0.                                     *SAB 910
C   PMV=C.                                     *SAB 920
C   NPROB=NPROB+1                              *SAB 930
C   * * * * *                                     *SAB 940
C   READ (5,1190) XIHB, XMHB, XIHZ, XMHZ       *SAB 950
C   IF (ENDFILE 5) 1130,40                     *SAB 960
C   IF (NPROB.GE.2) GO TO 1130                 *SAB 970
C   XIHB=X4HB=XIHZ=XMHZ=0.0                   *SAB 980
C   40 IF (NCASE.GT.0) WRITE (6,1140) NCASE    *SAB 990
C   * * * * *                                     *SAB1000
CHK=1.                                          *SAB1010
PSL=2116.217/144.                             *SAB1020
50  N=0                                         *SAB1030
C   XPRNT=0.                                   *SAB1040
C   NIT=0.                                     *SAB1050
C   XITER=0.                                  *SAB1060
C   XKKX=0.                                   *SAB1070
C   OVR=0.                                    *SAB1080
C   CONT=0.                                   *SAB1090
C   DELP=0.                                   *SAB1100
C   INCOMP=0                                  *SAB1110
C   NPT=0.                                    *SAB1120
C   JTP=0.                                    *SAB1130
C   LPL=0.                                    *SAB1140
C   SRE=0.0                                   *SAB1150
C   ALP1E=0.0                                 *SAB1160
C   STO=0.0                                   *SAB1170
C   RA=0.0                                    *SAB1180
C   ST=0.0                                    *SAB1190
C   HT=0.0                                    *SAB1200
C   * * * * *                                     *SAB1210

```



	IF (KCASE-3) 200,200,160	SAB1730
180	PRINT 1170	SAB1740
	GO TO 50	SAB1750
C	*****	SAB1760
190	WRITE (6,1160)	SAB1770
C	190 READ (5,1190) PMV	SAB1780
	GO TO 1130	SAB1790
230	K=KTEMP+1	SAB1800
	IF ((XIH8.GT.0..OR.XIHZ.GT.0.).AND.K.GT.1) GO TO 210	SAB1810
	WRITE (6,1200) (IHED1(J,KCASE),J=1,5),(IHED2(J,K),J=1,3)	SAB1820
	IF (KCASE.NE.2) GO TO 210	SAB1830
	IF (KDELT.EQ.0) GO TO 210	SAB1840
	IF (KTEMP.EQ.0) GO TO 210	SAB1850
	WRITE (6,1210)	SAB1860
	KTEMP=0	SAB1870
213	IF (HB.LT.250000.) GO TO 220	SAB1880
	WRITE (6,1260)	SAB1890
	WRITE (6,1270) HB	SAB1900
	IF (KCASE.EQ.2) GO TO 1000	SAB1910
	GO TO 50	SAB1920
220	IF (CONT.GT.0.) GO TO 250	SAB1930
	IF (KCASE-1) 230,230,250	SAB1940
230	ALTM=ABS(HZ-HB)	SAB1950
	IF (SRSV-ALTM) 240,250,250	SAB1960
240	WRITE (6,1380) HB,HZ,SR	SAB1970
	IF (XIH8.GT.0..OR.XIHZ.GT.0.) GO TO 50	SAB1980
	GO TO 40	SAB1990
250	IF (XITER) 260,260,390	SAB2000
250	IF (CHK) 270,270,280	SAB2010
270	IF (OVPR-1.) 310,340,390	SAB2020
280	SR=SR/1000.	SAB2030
	DO 290 J=1,3	SAB2040
	CALL ATMOS (H(J),TEMP(J),DEN,RHO(J),TR,PR(J),SS(J),VC,KER)	SAB2050
	IF (KER.NE.1) GO TO 300	SAB2060
290	P(J)=PR(J)*PSL	SAB2070
	GO TO 350	SAB2080
300	WRITE (6,1250) J,KER	SAB2090
	GO TO 1060	SAB2100
310	IF (XIH8) 330,330,320	SAB2110
320	CALL ATMOS (HB,TEMP(3),DEN,RHO8,TR,P8R,SS8,VC,KER)	SAB2120
	IF (KER.NE.1) GO TO 300	SAB2130
	P8=P8R*PSL	SAB2140
330	IF (XIHZ) 350,350,340	SAB2150
340	CALL ATMOS (HZ,TEMP(1),DEN,RHOZ,TR,PZR,SSZ,VC,KER)	SAB2160
	IF (KER.NE.1) GO TO 300	SAB2170
	PZ=PZR*PSL	SAB2180
350	IF (KTEMP.EQ.0) GO TO 390	SAB2190
	PSL=14.696*TEMPS/518.67	SAB2200
	IF (OVPR) 360,360,380	SAB2210
360	DO 370 J=1,3	SAB2220
	P(J)=P(J)*TEMP(J)/TH(J)	SAB2230

	PR(J)=P(J)/PSL	SAB2240
	RHO(J)=RHO(J)*TM(J)/TEMP(J)	SAB2250
370	SS(J)=49.02*TM(J)**.5	SAB2260
	GO TO 390	SAB2270
380	SS(1)=49.02*TM(1)**.5	SAB2280
390	IF (PMVD.EQ.0.) GO TO 430	SAB2290
	IF (XPRNT-1.) 400,430,410	SAB2300
400	IF (XITER-1.) 420,410,430	SAB2310
410	WRITE (6,1280) PMVD	SAB2320
	GO TO 430	SAB2330
420	WRITE (6,1280) PMVD	SAB2340
	IF (KOPT.EQ.0) KOPT=1	SAB2350
	WCZ1=PMV/SSZ	SAB2360
	WCZ2=WCZ1**2.	SAB2370
	RADIC=SQRT(.36*WCZ2+1.)	SAB2380
	DELP=PZ*(.84*WCZ2+1.4*WCZ1*RADIC)	SAB2390
430	CHK=J.	SAB2400
	OVPR=0.	SAB2410
	PBPZ=PZ/PB	SAB2420
	SSZR=SSZ/1116.4437	SAB2430
	PBRW=(PBR/W)**.333333	SAB2440
	PZRW=(PZR/W)**.333333	SAB2450
440	GO TO (470,450,480), KCASE	SAB2460
450	IF (HF.EQ.0..AND.N.GT.0) GO TO 50	SAB2470
	RF=145.*W**.4	SAB2480
	ALTF=HB-HG	SAB2490
	IF (RF-ALTF) 490,460,460	SAB2500
460	WRITE (6,1300) W,HB	SAB2510
	GO TO 1060	SAB2520
470	IF (SR.NE.0.0) GO TO 490	SAB2530
	PRINT 1180	SAB2540
	GO TO 50	SAB2550
480	IF (W.NE.0.0) GO TO 890	SAB2560
	PRINT 1190	SAB2570
	GO TO 50	SAB2580
490	CALL TRIPNT (KCASE)	SAB2590
	IF (INCOMP.EQ.1) GO TO 1060	SAB2600
	GO TO (550,500), KCASE	SAB2610
500	STJ=SRE*SIN(ALP1ER)	SAB2620
	ALPHIR=0.0	SAB2630
510	CALL SETUP (TAB4I,1,2,18,0,0,0,0)	SAB2640
	CALL MACURE (TAB4D,ALPHIR,0,0,0,0,LER,PHIR)	SAB2650
	ALPHI=ALPHIR+ALP1ER	SAB2660
	RA=SRE*(SIN(R90+ALP1ER-PHIR)/SIN(R90-ALPHI+PHIR))	SAB2670
	RA=ABS(RA)	SAB2680
	JTP=JTP+1	SAB2690
	ST=RA*COS(R90-ALPHI)	SAB2700
	STP(JTP)=ST	SAB2710
	IF (NPT.GT.0) GO TO 520	SAB2720
	IF (JTP.EQ.1) GO TO 520	SAB2730
	IF (STP(JTP).LE.STP(JTP-1)) NPT=JTP-1	SAB2740

520	HT=(ST-ST0)*(SIN(PHIR)/COS(PHIR))	SAB2750
	HTP(JTP)=HT	SAB2760
	RBAR=RA*PBRW	SAB2770
	CALL SETUP (TAB6I,1,2,69,0,0,0,0,0)	SAB2780
	CALL MACURE (TAB6D,RBAR,0,0,0,0,0,0,LER,TSACAP)	SAB2790
	TSA=TSACAP/(SSZR*PZRW)	SAB2800
	IF (N.EQ.0) GO TO 530	SAB2810
	IF (ALPHIR.EQ.0.0) WRITE (6,123J) ID(4),HB, ID(5),W, ID(16),SRE, ID(1	SAB2820
	17),ALP1E, ID(18),STO, ID(3),HG	SAB2830
	WRITE (6,1240) ID(19),RA, ID(20),ST, ID(21),HT, ID(6),TSA	SAB2840
	GO TO 540	SAB2850
530	IF (ALPHIR.EQ.0.0) WRITE (6,1220) ID(4),HB, ID(5),W, ID(16),SRE, ID(1	SAB2860
	17),ALP1E, ID(18),STC, ID(3),HG	SAB2870
	WRITE (6,1240) ID(19),RA, ID(20),ST, ID(21),HT, ID(6),TSA	SAB2880
540	ALPHIR=ALPHIR+R10	SAB2890
	IF (R130-ALPHIR) 1(2),510,510	SAB2900
550	PBRWFR=(PBR/(W*FR))**.333333	SAB2910
	PZRWFR=(PZR/(W*FR))**.333333	SAB2920
	RBAR=SR*PBRWFR	SAB2930
	IF (KOPT.EQ.1) GO TO 570	SAB2940
	CALL SETUP (TAB1I,1,2,69,0,0,0,0,0)	SAB2950
	CALL MACURE (TAB1D,RBAR,0,0,0,0,0,0,LER,SDELP)	SAB2960
	IF (HZ.NE.HB) GO TO 560	SAB2970
	DELP=SDELP*PBR	SAB2980
	ALFA=1.0	SAB2990
	GO TO 570	SAB3000
560	CALL SETUP (TAB2I,1,2,62,0,0,0,0,0)	SAB3010
	CALL MACURE (TAB2D,RBAR,0,0,0,0,0,0,LER,ALFA)	SAB3020
	DELP=SDELP*PBR*PBPZ**ALFA	SAB3030
570	EPSILO=DELP/PZ	SAB3040
	DO 580 J=1,7	SAB3050
	IF (RBAR.LT.UL2(J)) GO TO 590	SAB3060
580	CONTINUE	SAB3070
	TDPZ=.252609+(1.)/11.21)*ALOG(RBAR)	SAB3080
	GO TO 600	SAB3090
590	TDPZ=C4(J)*RBAR**2+C5(J)*RBAR+C6(J)	SAB3100
600	DO 610 J=1,8	SAB3110
	IF (RBAR.LT.UL3(J)) GO TO 620	SAB3120
610	CONTINUE	SAB3130
	J=8	SAB3140
620	TMV=C7(J)*RBAR**P2(J)+C8(J)*RBAR+C9(J)	SAB3150
	DO 630 J=1,8	SAB3160
	IF (RBAR.LT.UL4(J)) GO TO 640	SAB3170
630	CONTINUE	SAB3180
	J=8	SAB3190
640	TDRZ=C10(J)*RBAR**P3(J)+C11(J)*RBAR+C12(J)	SAB3200
	PDOOP=TDPZ/(SSZR*PZRWFR)	SAB3210
	PMV=5.0*EPSILO*SSZ/(7.C*(1.0+6.0*EPSILO/7.0)**.5)	SAB3220
	PDMV=TMV*PDOOP	SAB3230
	POD=RHOZ*(7.J+6.J*EPSILO)/(7.+EPSILO)	SAB3240
	PODD=TDRZ*PDOOP	SAB3250

	SFV=SSZ*(1.0+6.C*EPSILO/7.0)**.5	SAB3260
	Q=2.5*DELPA**2.0/17.0*PZ+DELP	SAB3270
	CALL SETUP (TAB6I,1,2,69,0,0,0,0,0,0)	SAB3280
	CALL MACURE (TAB6D,RBAR,0,0,0,0,0,0,0,LER,TSACAP)	SAB3290
	TSA=TSACAP/(SSZR*PZRWFR)	SAB3300
	IF (KOFT.EQ.1) GO TO 830	SAB3310
	HORF=SQRT((SR*10J0.)**2-((ABS(HZ-HB))**2))	SAB3320
	HORN=HORF*.000164	SAB3330
	IF (CONT.EQ.0.) GO TO 730	SAB3340
	NIT=NIT+1	SAB3350
	IF (NIT-1) 730,730,650	SAB3360
650	IF (PMVD) 670,670,660	SAB3370
660	PVPO=ABS(PMV-PMVD)	SAB3380
	IF (PVPO-.01) 730,730,690	SAB3390
670	PODD=ABS(DELP-DELPD)	SAB3400
	IF (PODD-.01) 730,730,680	SAB3410
680	DELPR=DELP	SAB3420
	SRS=ABS(HZ-HB)	SAB3430
	SRS=SRS*(DELPR/DELPD)**.33	SAB3440
	GO TO 700	SAB3450
690	SRS=ABS(HZ-HB)	SAB3460
	SRS=SRS*(PMV/PMVD)**.33	SAB3470
700	IF (HB.GT.HZ) GO TO 710	SAB3480
	HZ=HB+SRS	SAB3490
	GO TO 720	SAB3500
710	HZ=HB-SRS	SAB3510
720	OVPR=1.0	SAB3520
	IF (NIT.GT.1) XPRNT=1.	SAB3530
	GO TO 970	SAB3540
730	WRITE (6,1220) IO(1),SR,IO(2),HZ,IO(3),HG,IO(4),HB,IO(5),W	SAB3550
	WRITE (6,1240) IO(6),TSA,IO(7),FR,IO(8),SFV,IO(25),Q	SAB3560
	WRITE (6,1240) IO(9),DELP,IO(10),PMV,IO(11),POD,IO(26),RHOZ	SAB3570
	WRITE (6,1240) IO(12),PDDOP,IO(23),PDMV,IO(24),PDD,IO(27),SSZ	SAB3580
	WRITE (6,1240) IO(12),SDELP,IO(13),RBAR,IO(14),R,IO(15),ALFA	SAB3590
	WRITE (6,1240) IO1(1),HORF,IO1(2),HORN	SAB3600
	IF (HB.LT.25000.) GO TO 740	SAB3610
	WRITE (6,1240) IO(28),CFF	SAB3620
	WRITE (6,1290) MOR	SAB3630
	GO TO 750	SAB3640
740	WRITE (6,1300)	SAB3650
750	IF (CCNT.EQ.0.) GO TO 1060	SAB3660
	IF (NIT-1) 650,650,760	SAB3670
760	IF (PMVD) 770,770,800	SAB3680
770	WRITE (6,1370)	SAB3690
	WRITE (6,1330)	SAB3700
	WRITE (6,1360) NIT	SAB3710
	WRITE (6,1370)	SAB3720
	IF (XIMB.GT.0..OR.XIHZ.GT.0.) GO TO 780	SAB3730
	IF (KOELT.GT.0) GO TO 790	SAB3740
	GO TO 1060	SAB3750
780	KCASE=SAVCASE	SAB3760

	HZ=SAVALT	SAB3770
	HB=SAVBURS	SAB3780
	DELP=SAVDELP	SAB3790
790	GO TO 1060	SAB3800
	KCASE=SAVCASE	SAB3810
	HZ=SAVALT	SAB3820
	DELP=SAVDELP	SAB3830
	GO TO 1060	SAB3840
800	WRITE (6,1370)	SAB3850
	WRITE (6,1320)	SAB3860
	WRITE (6,1360) NIT	SAB3870
	WRITE (6,1370)	SAB3880
	IF (XIMB.GT.0..OR.XIMZ.GT.0.) GO TO 810	SAB3890
	IF (KDEL.T.GT.0) GO TO 820	SAB3900
	GO TO 1060	SAB3910
810	KCASE=SAVCASE	SAB3920
	HZ=SAVALT	SAB3930
	HB=SAVBURS	SAB3940
	DELP=SAVDELP	SAB3950
	GO TO 1060	SAB3960
820	KCASE=SAVCASE	SAB3970
	HZ=SAVALT	SAB3980
	DELP=SAVDELP	SAB3990
	GO TO 1060	SAB4000
830	IF (PMVD) 860,860,840	SAB4010
840	PMVR=PMV	SAB4020
	NIT=NIT+1	SAB4030
	XITER=XITER+1.	SAB4040
	IF (ABS(PMVR-PMVD)-.01) 860,860,850	SAB4050
850	DELP=DELP*(PMVD/PMVR)**.33	SAB4060
	IF (NIT.GT.1) GO TO 210	SAB4070
	GO TO 200	SAB4080
860	WRITE (6,1240) ID(6),TSA, ID(8),SFV,IJ(25),Q, ID(10),PMV	SAB4090
	WRITE (6,1240) ID(11),POD, ID(26),RHDZ, ID(22),PDDOP, ID(23),PMHV	SAB4100
	WRITE (6,1240) ID(24),PDDO, ID(27),SSZ, ID(14),R, ID(15),ALFA	SAB4110
	IF (HB.LT.25000.) GO TO 870	SAB4120
	WRITE (6,1240) ID(28),CFF	SAB4130
	WRITE (6,1290) WOR	SAB4140
	GO TO 880	SAB4150
870	WRITE (6,1300)	SAB4160
880	IF (XITER.EQ.0.) GO TO 1060	SAB4170
	WRITE (6,1370)	SAB4180
	WRITE (6,1320)	SAB4190
	WRITE (6,1360) NIT	SAB4200
	WRITE (6,1370)	SAB4210
	GO TO 1060	SAB4220
890	ALFA=0.0	SAB4230
	SDELP=DELP/(PBR*PBPZ**ALFA)	SAB4240
	DOELP=-DELP*ALOG(PBPZ)/(PBR*PBPZ**ALFA)	SAB4250
900	J=1	SAB4260
	IF (DOELP-CFS) 920,910,910	SAB4270

```

91J   J=2                                     SAB4280
92G   PPP=CF11(J)+CF22(J)*ALOG1J(SOELP)+CF33(J)*(ALOG1U(SOELP)**2)+CF44(SAB429L
1J)* (ALOG10(SOELP)**3)+CF55(J)*(ALOG1J(SOELP)**4)   SAB430U
      RBAR=1J.**PPP                                     SAB431U
      IF (HB.EQ.HZ) GO TO 960                         SAB4320
      AA=ALOG10(2.71828)/SOELP                       SAB433U
      BB=2.*ALOG10(SOELP)*AA                          SAB434U
      CC=3.*(ALOG10(SOELP)**2)*AA                   SAB435U
      DD=4.*(ALOG10(SOELP)**3)*AA                   SAB4360
      OPOX=CF22(J)*AA+CF33(J)*BB+CF44(J)*CC+CF55(J)*DD SAB437U
      ORBAR=RBAR*ALOG(10.)*OPOX                      SAB438U
      CF4=J.                                          SAB439U
      OO 930 II=1,6                                    SAB4400
      IF (RBAR-CFR(II)) 94J,93U,93G                 SAB441U
93J   CONTINUE                                     SAB442U
      II=7                                           SAB443U
      C=4=-.J38                                       SAB444U
94G   SMF=(CF1(II)*RBAR+CF2(II))*RBAR+CF3(II)+CF4*ALOG10(RBAR) SAB445U
      CAPQ=2.*CF1(II)*RBAR+CF2(II)+CF4*ALOG10(2.71828)/RBAR SAB446U
      CAPQ=CAPQ*DRBAR*OOELP                          SAB447U
      ALFO=ALFA                                       SAB448U
      ALFA=(SMF-ALFO*CAPQ)/(1.-CAPQ)                 SAB449U
      IF (ABS(ALFA-ALFO)-.001) 96U,960,95J         SAB450U
95G   SOELP=OELP/(PBR*PBPZ**ALFA)                  SAB451U
      OOELP=-OELP*ALOG(PBPZ)/(PBR*PBPZ**ALFA)      SAB452U
      GO TO 90J                                       SAB453U
95J   SR=RBAR*(W/PBR)**.333333                      SAB454U
      CALL TRIPNT(KCASE)                              SAB455U
      IF (XKK.LE.1.0) GO TO 98U                      SAB456U
C     *****                                       SAB457U
      IPRJBI=1                                        SAB458U
      IF (TSA.GT.0.0) TSACH=TSA                      SAB4590
      GO TO 113U                                       SAB460U
97G   HABS=ABS(HZ-HB)                                 SAB461U
      SR=SQRT(HABS**2+(HABS*.J0001)**2)              SAB462U
      SR=SR/1000.                                     SAB4630
      KCASE=1                                         SAB464U
      KOPT=0                                          SAB465U
      IF (NIT.GT.1) GO TO 21C                        SAB466U
      GO TO 20J                                       SAB467U
99G   IF (FR.NE.1.0) SR=RBAR*(FR*W/PBR)**.333333   SAB468U
      IF (XKK=1.) 990,1000,1000                     SAB4690
99G   HORF=SQRT((SR*1000.)**2-((ABS(HZ-HB))**2))    SAB470U
      HORN=HORF*.000164                              SAB471U
      GO TO 101U                                       SAB472U
100C  HORF=0.                                         SAB4730
      HORN=0.                                         SAB474U
101C  WRITE (6,1220) IO(5),W,ID(9),OELP,IO(2),HZ,IO(3),HG,ID(4),HB SAB475U
      WRITE (6,1240) ID(1),SR,IO(13),RBAR,IO(12),SOELP,IO(7),FR  SAB476U
      WRITE (6,1240) ID1(1),HORF,ID1(2),HORN        SAB477U
      IF (KOPT.EQ.1) GO TO 55G                       SAB478U

```

1020	IF (HB.LT.25000.) GO TO 1030	SAB4790
	WRITE (6,1240) ID(28),CFF	SAB4800
	WRITE (6,1290) WOR	SAB4810
	GO TO 1040	SAB4820
1030	WRITE (6,1300)	SAB4830
1040	IF (JTP.EQ.0.) GO TO 1060	SAB4840
	LPL=LPL+100	SAB4850
	WRITE (6,1310) JTP,NPT	SAB4860
	DUMY1=AMAX1(STP(NPT),HTP(NPT))-0.2	SAB4870
	CALL GRAPH (3,1,10,10,9HHOR RANGE,8HALTITUDE,LPL,DUMY1,DUMY1,9.,9.,9.	SAB4880
	1.,1.,1.)	SAB4890
	CALL GRAPH (5,NPT,10,10,9HHOR RANGE,8HALTITUDE,LPL,STP,HTP,9.,9.,1	SAB4900
	1.,1.)	SAB4910
	CALL GRAPH (6)	SAB4920
	DO 1050 I=1,JTP	SAB4930
	STP(JTP)=0.	SAB4940
1050	HTP(JTP)=0.	SAB4950
1050	IF (XIMB.GT.0..OR.XIMZ.GT.0.) GO TO 50	SAB4960
	CHK=2.	SAB4970
	IF (KDELT.LE.0) GO TO 50	SAB4980
	INCOMP=0	SAB4990
	N=N+1	SAB5000
	IF (N.GT.12) GO TO 50	SAB5010
	GO TO (1100,1070,1110), KCASE	SAB5020
1070	HB=SUBS(N)	SAB5030
	IF (HB.LT.25000..OR.HB.GE.25000.) GO TO 1080	SAB5040
	CALL SETUP (ACF,1,2,13,0,0,0,0,0)	SAB5050
	CALL MACURE (CF,HB,0,0,0,0,0,LER,CFF)	SAB5060
	WOR=W	SAB5070
	W=CFF*W	SAB5080
	GO TO 1090	SAB5090
1080	IF (HB.LT.25000.) GO TO 1090	SAB5100
	WRITE (6,1260)	SAB5110
	WRITE (6,1270) HB	SAB5120
	GO TO 1060	SAB5130
1090	JTP=0	SAB5140
	NPT=0	SAB5150
	CALL ATMOS (HB,TEMP(3),DEN,RHOB,TR,PBR,SSB,VC,KER)	SAB5160
	IF (KER.NE.1) GO TO 300	SAB5170
	PB=PBR*PSL	SAB5180
	NIT=0	SAB5190
	GO TO 430	SAB5200
1100	SR=SUBS(N)/1000.0	SAB5210
	NIT=0	SAB5220
	GO TO 440	SAB5230
1110	IF (HB.LT.25000.) GO TO 1120	SAB5240
	WOR=SUBS(N)	SAB5250
	W=SUBS(N)*CFF	SAB5260
	NIT=0	SAB5270
	IF (W.GT.0.) GO TO 430	SAB5280
	GO TO 440	SAB5290

```

1120 W=SUBS(N)                                SAB5300
      NIT=0                                    SAB5310
      IF (W.GT.0.) GO TO 430                   SAB5320
      GO TO 440                                SAB5330
C     *****                                *SAB5340
1130 SRCH=SR                                    SAB5350
      HZCH=HZ                                    SAB5360
      HGCH=HG                                    SAB5370
      HBCM=HB                                    SAB5380
      WCY=W                                       SAB5390
      DELPCM=OELP                                SAB5400
      TSA=TSACM                                SAB5410
      RETURN                                     SAB5420
C     *****                                *SAB5430
C     *****                                *SAB5440
C     *****                                *SAB5450
C     *****                                *SAB5460
C     1220 FORMAT (1H1,45X,5A10,/,55X,3A10,///) SAB5470
C     *****                                *SAB5480
C     *****                                *SAB5490
1140 FORMAT (///17H STATEMENT 140 ///)        SAB5500
1150 FORMAT (///17H STATEMENT 160 ///)        SAB5510
1160 FORMAT (///17H STATEMENT 190 ///)        SAB5520
1170 FORMAT (1H1,////,10X,65HCASE INPUT ERROR - KCASE GREATER THAN 3 - SAB5530
      1PROCEEDING TO NEXT CASE)                SAB5540
1180 FORMAT (1H,////,10X,80HSLANT RANGE FOUND EQUAL TO ZERO IN OVERPRESAB5550
      1SSURE SOLUTION PROCEEDING TO NEXT CASE) SAB5550
1190 FORMAT (1H,////,10X,67HYIELD FOUND EQUAL TO ZERO IN RANGE SOLUTIOSAB5570
      1N PROCEEDING TO NEXT CASE)            SAB5580
1200 FORMAT (1H,45X,5A10,/,55X,3A10,///)     SAB5590
1210 FORMAT (1M0,20X,47HTHIS CASE MUST BE RUN WITH STANDARD TEMPERATURESAB5600
      1)                                       SAB5610
1220 FORMAT (1H0,6(4X,A6,1PE12.5))            SAB5620
1230 FORMAT (1H1,////,6(4X,A6,1PE12.5))      SAB5630
1240 FORMAT (1H0,22X,4(4X,A6,1PE12.5))       SAB5640
1250 FORMAT (1H0,8HATMOS ER,I2,2H =,I3)      SAB5650
1260 FORMAT (1H0,///,48X,37HBLAST EFFECTS ARE ESSENTIALLY REDUCED,///,48SAB5660
      1X,24HTO ZERO AT THIS ALTITUDE)        SAB5670
1270 FORMAT (1H0,47X,5HMB = .1PE12.5)        SAB5680
1280 FORMAT (56X,12H(GUST INPUT),//,56X,5HMPV =,E12.5,/) SAB5690
1290 FORMAT (1H0,26X,36HORIGINAL WEAPON YIELD AS INPUTED(KT),1PE12.9) SAB5700
1300 FORMAT (1H0,22X,39HYIELD CORRECTION FACTOR IS EQUAL TO ONE)    SAB5710
1310 FORMAT (1H0,6HJTP = ,I6,//,1X,6HNPT = ,I6) SAB5720
1320 FORMAT (1H0,9X,55HTHE GUST VELOCITY RECEIVED IS WITHIN ONE PERCENTSAB5730
      1 OF THE,///,10X,49HDESIRED VALUE - NO FURTHER ITERATION IS NECESSARSAB5740
      2)                                       SAB5750
1330 FORMAT (1H0,9X,54HTHE OVERPRESSURE RECEIVED IS WITHIN ONE PERCENT SAB5760
      1 OF THE,///,10X,49HDESIRED VALUE - NO FURTHER ITERATION IS NECESSARYSAB5770
      2)                                       SAB5780
1340 FORMAT (1H1,5/,1X,134(1H*),3/,10X,51HTHIS PROBLEM MUST BE RUN WITHSAB5790
      1 A STANOARD ATMOSPHERE,///,10X,46HIF NON STANDARD ATMOSPHERE IS STISAB5800

```

```

2LL DESIRED AN, //, 10X, 55H INCREMENTAL HEIGHT OF BURST OR RECEIVER MASAB5810
3Y NOT BE USED, 3/, 10X, 4LHE.G. SET XIHB AND XIHZ EQUAL TO ZERO AND, /SAB5820
4/, 10X, 31HRUN EACH CONDITION INDIVIDUALLY, 3/, 11X, 53H THE FOLLOWING OSAB5830
5UTPUT IS VALID FOR STANDOARD ATMOSPHERE, 3/, 1X, 134(1H*) ) SAB5840
135C FORMAT (1H1, 5/, 1X, 134(1H*), 3/, 10X, 36H THE KDEL T = 1 OPTION MAY NOT SAB5850
1BE USED, //, 10X, 37H WITH AN INCREMENTAL BURST OR RECEIVER, 3/, 10X, 41HSAB5860
2E.G. IF KOELT = 1 - XIHB AND XIHZ MUST BE, //, 10X, 13HEQUAL TO ZERO, SAB5870
33/, 10X, 33H IN THE FOLLOWING OUTPUT KDEL T = 0, 3/, 1X, 134(1H*) ) SAB5880
1360 FORMAT (1H0, 9X, 23H NUMBER OF ITERATIONS = , I2) SAB5890
137C FORMAT (1H0, 134(1H*)) SAB5900
138C FORMAT (10/, 45X, 40(1H*), 3/, 45X, 40H* AN OVERPRESSURE SOLUTION CASAB5910
1N NOT *, //, 45X, 40H* BE OBTAINED WITH THE GIVEN *, //, 45X, 4SAB5920
20H* INPUT GEOMETRY *, //, 60X, 4HMB = , E12.5, //, 6SAB5930
30X, 4HMB = , E12.5, //, 60X, 4HSR = , E12.5, //, 45X, 40H* THE PROGRAM WILL SAB5940
4 PROCEED WITH *, //, 45X, 40H* THE NEXT CASE SAB5950
5 *, 3/, 45X, 40(1H*)) SAB5960
1390 FORMAT (10/, 45X, 40(1H*), 3/, 45X, 40H* NO TRIPLE POINT CALCULATIO SAB5970
1N *, //, 45X, 40H* REQUIRED - BURST AND YIELD *, //, 45X, 4SAB5980
20H* COMBINATION REQUIRES GROUND *, //, 45X, 40H* BURST CR SAB5990
3ITERIA *, //, 60X, 5HW = , F10.2, //, 60X, 5HMB = , F10.2, SAB6000
42, //, 60X, 15HFR = 1.60, //, 45X, 40H* ALL RECEIVERS FALL WITH SAB6010
5IN *, //, 45X, 40H* TRIPLE POINT PATH *, //, 4SAB6020
65X, 40H* THE PROGRAM WILL PROCEED WITH *, //, 40X, 40H* THE SA SAB6030
7NEXT CASE *, 3/, 45X, 40(1H*)) SAB6040
END SAB6050-
SUBROUTINE TRIPNT (KCASE) TPN 10
*****TPN 20
SUBROUTINE TRIPNT CALCULATES LIMITING ANGLE FOR REGULAR REFLECTION TPN 30
AND PREDICTS WHETHER RECEIVER IS IN OUT OF THE FUSED SHOCK TPN 40
REGION TPN 50
TPN 60
ROUTINE REQUIREMENTS- TPN 70
NUMEROUS PARAMETERS FORM MAIN ROUTINE THROUGH COMMON TPN 80
CALLS SUBROUTINES SETUP AND MACURE TPN 90
TPN 100
CALLING SEQUENCE TPN 110
WHERE- TPN 120
KCASE=1 FOR OVERPRESSURE SOLUTION TPN 130
2 FOR TRIPLE POINT PATH SOLUTION TPN 140
3 FOR RANGE SOLUTION TPN 150
*****TPN 160
CALL TRIPNT(KCASE) TPN 170
COMMON /TRI/ PZ, PG, PB, HZ, HG, HB, PZR, PGR, PBR, W, SR, FR, SRE, ALP1ER, ALP1 TPN 180
1E, R90, PBPZ, R, PBRW, INCOMP TPN 190
COMMON /SENSE/ CFS TPN 200
COMMON /COLO/ CF11(2), CF22(2), CF33(2), CF44(2), CF55(2), CFR(6), CF1(7 TPN 210
1), CF2(7), CF3(7) TPN 220
COMMON /TAB/ TAB1I(69), TAB10(69), TAB2I(62), TAB20(62), TAB3I(69), TAB TPN 230
130(69), TAB4I(18), TAB40(18), TAB5I(26), TAB50(26), TAB6I(69), TAB6U(69) TPN 240
COMMON /CON/ UL2(7), UL3(8), UL4(8), UL5(5), C4(7), C5(7), C6(7), C7(8), CTPN 250
18(8), C9(8), C10(8), C11(8), C12(8), P2(8), P3(8), A1E(41), ACF(13), CF(13) TPN 260

```

	COMMON /PNT/ RA,STO,PHIR,SHB,ST,XKK,ALTSRG,HT,XKKX	TPN 270
	COMMON /QVP/ DELP,DELPD,DELPR,CONT,NIT,XITER	TPN 280
	DIMENSION ALPHA(41), IHED1(5,3), ID(28)	TPN 290
	DATA (IHED1(J),J=6,10)/50H TRIPLE POINT PATH SOLUTION	TPN 300
1	/	TPN 310
	DATA (ID(J),J=1,28)/2HSR,2HMZ,2HMG,2HMB,1HW,3HTSA,2HFR,3HSFV,4HDEL	TPN 320
	1P,3HPMV,3HPDD,5HSOELP,4HRBAR,1HR,4HALFA,3HSRE,5HALPE,3HSTJ,2HRA,2TPN	TPN 330
	2HST,2HHT,5HPDOOP,4HPDMV,4HPDOD,1HQ,4HRHOZ,3HSSZ,3HCFF/	TPN 340
	PGPB=PG/PB	TPN 350
	ALT=(HB-HG)/1000.0	TPN 360
	R=145.0*W**0.4/1000.0	TPN 370
	SHB=ALT*PBKM	TPN 380
	SHBB=ALT/W**.3333	TPN 390
	IF (SHBB.GT.2.5) GO TO 110	TPN 400
	IF (R.LT.ALT) GO TO 30	TPN 410
	FR=1.6	TPN 420
	IF (SR.NE.0.) GO TO 10	TPN 430
	XK=0.	TPN 440
	PRINT 220	TPN 450
	GO TO 2J	TPN 460
10	XK=ABS(HZ-HB)/(SR*1000.0)	TPN 470
20	IF ((ABS(XK-1.)).LE..002) XK=1.	TPN 480
	IF (XK.LE.1.0) GO TO 200	TPN 490
	GO TO 170	TPN 500
30	XI=C.0	TPN 510
	RBAR=ALT*PFRW	TPN 520
	CALL SETUP (TAB2I,1,2,62,0,0,0,0,0)	TPN 530
	CALL MACURE (TAB2D,RBAR,0,0,0,0,C),LER,ALFA)	TPN 540
	OO 100 J=1,41	TPN 550
	ALFA=0.	TPN 560
	XI=XI+0.025	TPN 570
	IF (XI.GT.1.0) XI=1.0	TPN 580
	DELPG=PG/XI-PG	TPN 590
40	SDELP=DELPG/(PBR*PGPB**ALFA)	TPN 600
	DDELP=-DELPG*ALOG(PGPB)/(PBR*PGPB**ALFA)	TPN 610
	K=1	TPN 620
	IF (SOELP-CFS) 60,50,50	TPN 630
50	K=2	TPN 640
60	PPP=CF11(K)+CF22(K)*ALOG10(SDELP)+CF33(K)*(ALOG10(SDELP)**2)+CF44(K)	TPN 650
	1K*(ALOG10(SDELP)**3)+CF55(K)*(ALOG10(SDELP)**4)	TPN 660
	RBAR=10.**PPP	TPN 670
	AA=ALOG10(2.71828)/SDELP	TPN 680
	BB=2.*ALOG10(SDELP)*AA	TPN 690
	CC=3.*(ALOG10(SDELP)**2)*AA	TPN 700
	DD=4.*(ALOG10(SDELP)**3)*AA	TPN 710
	DPX=CF22(K)*AA+CF33(K)*BB+CF44(K)*CC+CF55(K)*DD	TPN 720
	DRBAR=RBAR*ALOG(10.)*OPDX	TPN 730
	CF4=0.	TPN 740
	DO 70 II=1,6	TPN 750
	IF (RBAR-CFR(II)) 80,70,70	TPN 760
70	CONTINUE	TPN 770

	II=7	TPN 780
	CF4=-.038	TPN 790
80	SMF=(CF1(II)*RBAR+CF2(II))*RBAR+CF3(II)+CF4*ALOG1J(RBAR)	TPN 800
	CAPQ=2.*CF1(II)*RBAR+CF2(II)+CF4*ALOG1G(2.71828)/RBAR	TPN 810
	CAPQ=CAPQ*DRBAR*DELP	TPN 820
	ALFO=ALFA	TPN 830
	ALFA=(SMF-ALFO*CAPQ)/(1.-CAPQ)	TPN 840
	IF (ABS(ALFA-ALFO)-.001) 90,90,4J	TPN 850
90	ALTSRG=ALT/(RBAR*(W/PBR)**.333333)	TPN 860
	IF (ALTSRG.GT.1.00) GO TO 100	TPN 870
	ALPHA(J)=ACOS(ALTSRG)*57.296	TPN 880
	IF (ALPHA(J).GT.A1E(J)) GO TO 14J	TPN 890
130	CONTINUE	TPN 900
110	IF (KCASE.NE.2) GO TO 120	TPN 910
	INCOMP=1	TPN 920
	WRITE (6,230) (IHED1(J,KCASE),J=1,5)	TPN 930
	WRITE (6,240)	TPN 940
	WRITE (6,250) HB	TPN 950
	RETURN	TPN 960
120	FR=1.0	TPN 970
	IF (XITER.GT.1..OR.NIT.GT.1) GO TO 130	TPN 980
	PRINT 260	TPN 990
130	XK=ABS(HZ-HB)/(SR*1000.0)	TPN1000
	IF ((ABS(XK-1.)).LE..002) XK=1.	TPN1010
	IF (XK.LE.1.0) GO TO 200	TPN1020
	GO TO 170	TPN1030
140	IF (J.NE.1) GO TO 150	TPN1040
	ALP1E=A1E(1)	TPN1050
	GO TO 160	TPN1060
150	ALP1E=((A1E(J)-A1E(J-1))*(ALPHA(J-1)-A1E(J-1))/(A1E(J)-A1E(J-1))-AL	TPN1070
	1PHA(J)+ALPHA(J-1))+A1E(J))	TPN1080
160	ALP1ER=ALP1E/57.296	TPN1090
	SRE=ALT/COS(ALP1ER)	TPN1100
	IF (KCASE.EQ.2) GO TO 200	TPN1110
	ALPHI=R90	TPN1120
	IF (HZ.EQ.HB) GO TO 190	TPN1130
	XK=ABS(HZ-HB)/(SR*1000.0)	TPN1140
	IF ((ABS(XK-1.)).LE..002) XK=1.	TPN1150
	IF (XK.LE.1.0) GO TO 180	TPN1160
170	WRITE (6,270) XK	TPN1170
	WRITE (6,280) ID(5),W,ID(9),DELP	TPN1180
	WRITE (6,280) ID(2),HZ,ID(3),HG	TPN1190
	WRITE (6,280) ID(4),HB,ID(1),SR	TPN1200
	WRITE (6,290)	TPN1210
	DELPD=DELP	TPN1220
	CONT=CONT+1.	TPN1230
	WRITE (6,300)	TPN1240
	XKK=XK	TPN1250
	XKKX=XK	TPN1260
	RETURN	TPN1270
180	XKK=XK	TPN1280

```

XK=ASIN(XK)
IF (HZ.LT.HB) ALPHI=ALPHI-XK
IF (HZ.GT.HB) ALPHI=ALPHI+XK
190 ALPHIR=ALPHI-ALP1ER
CALL SETUP (TAB4I,1,2,18,0,J,0,0,0)
CALL MACURE (TAB4D,ALPHIR,0,0,0,0,LER,PHIR)
RA=SRE*COS(ALP1ER-PHIR)/COS(ALPHI-PHIR)
RA=ABS(RA)
FR=1.0
IF (XKKX.GT.0.) GO TO 210
IF (RA.GT.SR) GO TO 210
CALL SETUP (TAB5I,1,2,26,0,J,0,0,0)
CALL MACURE (TAB5D,SHB,0,0,0,0,LER,FR)
IF (SHB.LT.1.54) GO TO 210
IF ((SR-RA).LT.0.1) GO TO 210
FR=2.33-0.025*RBAR
200 XKK=XK
210 RETJRN
C
C
220 FORMAT (29H SR=0.0 IN SUBROUTINE TRIPNT )
230 FORMAT (1H1,45X,5A10,///)
240 FORMAT (1H0,///,48X,35HINPUT PARAMETERS ARE NOT COMPATIBLE,///,48X,
134HFOR THE TRIPLE POINT PATH SOLUTION)
250 FORMAT (1H0,47X,5HMB = ,1PE12.5)
260 FORMAT (3/,1X,134(1H*),3/,10X,42HSCALED HEIGHT OF BURST IS GREATER
1 THAN 2.5,///,10X,51HGROUND EFFECT AMPLIFICATION FACTOR SET EQUAL
20 1.0,3/,1X,134(1H*),3/)
270 FORMAT (38H ***ARG OF ASIN (X) OUT OF RANGE. X=,E16.8,///)
280 FORMAT (1H0,22X,2(4X,A6,1PE12.5))
290 FORMAT (1X,5/)
300 FORMAT (10X,66HTHE INPUT GIVEN IS INCOMPATIBLE WITH A POSSIBLE PHY
1SICAL CONDITION,///,10X,39HTWO ALTERNATE SETS OF OUTPUT ARE GIVEN-,
2/,14X,61H1-RECEIVER DIRECTLY ABOVE OR BELOW THE BURST DEPENDING ON
3 THE,/,16X,57HINITIAL ORIENTATION OF RECEIVER WITH RESPECT TO THE
4BURST,///,14X,64H2-THE ALTITUDE AT WHICH THE DESIRED GUST OR OVERPRT
5ESSURE OCCURRS:///,34X,1H*,/,34X,1H*,/,34X,1H*,/,34X,1H*,/,32X,7H**
6****,/,32X,5H****,/,33X,3H***,/,34X,1H*)
END
BLOCK DATA
*****
C BLOCK DATA CONTAINS TABULATED VALUES USED IN THE MAIN PROGRAM
C AND IN SUBROUTINE TRIPNT
C *****
COMMON /TRI/ PZ,PG,PB,HZ,HG,HS,PZR,PGR,PBR,W,SR,FR,SRE,ALP1ER,ALP1B
1E,R90,PBPZ,R,PERW
COMMON /TAB/ TAB1I(69),TAB1O(69),TAB2I(62),TAB2O(62),TAB3I(69),TABB
130(69),TAB4I(18),TAB4O(18),TAB5I(26),TAB5O(26),TAB6I(69),TAB6O(69)
COMMON /CON/ UL2(7),UL3(8),UL4(8),UL5(5),C4(7),C5(7),C6(7),C7(8),C9D
18(8),C9(8),C10(8),C11(8),C12(8),P2(8),P3(8),A1E(41),ACF(13),CF(13)

```

DATA (TAB11(J),J=1,69)/.050,.0625,.075,.0875,.100,.125,.150,.175,.802 126  
 120J,.225,.250,.275,.300,.325,.350,.375,.400,.450,.500,.550,.600,.6802 136  
 250,.700,.750,.800,.850,.900,.950,1.000,1.105,1.221,1.350,1.492,1.6802 140  
 349,1.822,2.014,2.226,2.460,2.718,3.004,3.320,3.669,4.055,4.482,4.9802 150  
 453,5.474,6.050,6.686,7.389,8.166,9.025,9.974,11.023,12.182,13.464,802 160  
 514.880,16.445,18.174,20.086,22.198,24.532,27.113,29.964,33.115,36.802 170  
 6598,40.447,44.701,50.000,100.00/ 802 180  
 DATA (TAB10(J),J=1,69)/17200.,8000.,4240.,2490.,1660.,900.,545.,36802 190  
 10.,246.,186.,144.,107.,88.8,73.5,62.7,54.5,46.0,36.4,29.3,24.1,20.802 200  
 22,17.2,14.8,12.9,11.3,9.9,8.8,7.9,7.1,6.2,5.2,4.45,3.83,3.3,2.85,802 210  
 32.48,2.10,1.85,1.62,1.43,1.27,1.12,.98,.87,.77,.68,.60,.53,.47,.41802 220  
 4.,36.,32.,28.,252.,222.,198.,174.,156.,139.,124.,113.,100.,891.,08802 230  
 53.,0755.,069.,063.,0574.,0333/ 802 240  
 DATA TAB21/0.,.1.,.2.,.30.,.35.,.40.,.45.,.50.,.55.,.60.,.65.,.70.,.75.,.80.,.8802 250  
 15.,90.,95,1.0,1.5,2.0,2.5,3.0,3.5,4.0,4.5,5.0,5.5,6.0,6.5,7.0,7.5,802 260  
 28.0,8.5,9.0,9.5,10.,15.,20.,25.,30.,35.,40.,45.,50.,55.,60.,65.,70.802 270  
 3.,75.,80.,85.,90.,95.,100.,150.,200.,250.,300.,350.,400.,45.,500.802 280  
 4/ 802 290  
 DATA TAB20/.020,.0625,.125,.1800,.2150,.2400,.2930,.3320,.3650,.39802 300  
 110.,4150.,4350.,4530.,4690.,4800.,4900.,4990.,5060.,5490.,5620.,56802 310  
 260.,5670.,5660.,5650.,5630.,5620.,5610.,5605.,5590.,5580.,5575.,55802 320  
 370.,5560.,5540.,5535.,5530.,5475.,5420.,5390.,5360.,5330.,5310.,52802 330  
 490.,5275.,5260.,5230.,5220.,5210.,5200.,5190.,5180.,5170.,5160.,51802 340  
 550.,5080.,5020.,4980.,4940.,4900.,4830.,4870.,4840/ 802 350  
 DATA (TAB41(J),J=1,18),(TAB40(J),J=1,18)/0.0,5.0,10.0,15.0,20.0,25802 360  
 1.0,30.0,35.0,40.0,45.0,50.0,60.0,70.0,80.0,90.0,100.0,110.0,120.0,802 370  
 20.0,3.0,0.25,0.50,1.0,2.0,3.5,5.0,7.0,9.5,12.5,20.0,28.0,36.5,45.3302 380  
 3,54.5,63.3,72.0/ 802 390  
 DATA (TAB51(J),J=1,26),(TAB50(J),J=1,26)/0.0,0.1,0.2,0.3,0.4,0.5,0802 400  
 1.6,0.7,0.8,0.9,1.0,1.1,1.2,1.3,1.4,1.5,1.6,1.7,1.8,1.9,2.0,2.1,2.2802 410  
 2,2.3,2.4,2.5,1.60,1.62,1.75,2.02,2.39,2.90,3.53,4.30,5.00,5.46,5.5802 420  
 36,5.33,4.45,3.51,2.88,2.45,2.12,1.86,1.65,1.50,1.36,1.25,1.16,1.08802 430  
 4,1.03,1.00/ 802 440  
 DATA (TAB61(J),J=1,59)/.050,.0625,.075,.0875,.100,.125,.150,.175,.802 450  
 1200.,225,.250,.275,.300,.325,.350,.375,.400,.450,.500,.550,.600,.6802 460  
 250,.700,.750,.800,.850,.900,.950,1.000,1.105,1.221,1.350,1.492,1.6802 470  
 349,1.822,2.014,2.226,2.460,2.718,3.004,3.320,3.669,4.055,4.482,4.9802 480  
 453,5.474,6.050,6.686,7.389,8.166,9.025,9.974,11.023,12.182,13.464,802 490  
 514.880,16.445,18.174,20.086,22.198,24.532,27.113,29.964,33.115,36.802 500  
 6598,40.447,44.701,50.000,100.00/ 802 510  
 DATA (TAB60(J),J=1,69)/.00062,.00106,.00170,.00255,.00363,.00638,.802 520  
 100994.,01432,.01954,.02555,.03235,.04032,.04903,.05844,.06853,.079802 530  
 224.,09055,.11481,.14105,.16905,.19860,.22955,.26173,.29501,.32929,802 540  
 3.36445,.40041,.43710,.47444,.555,.646,.749,.804,.994,1.14,1.30,1.4802 550  
 48,1.68,1.90,2.15,2.43,2.73,3.07,3.45,3.86,4.32,4.83,5.40,6.02,6.71802 560  
 5,7.47,8.32,9.25,10.29,11.43,12.63,14.09,15.64,17.34,19.23,21.32,23802 570  
 6.62,26.17,28.99,32.11,35.55,39.36,44.10,90.00/ 802 580  
 DATA (UL2(J),J=1,7)/0.3,.35,.45,.6,1.7,3.8,10./,(UL3(J),J=1,8)/.3802 590  
 135.,5,1.0,2.5,7.0,10.0,17.0,0.0/,(UL4(J),J=1,8)/.28,.5,.85,1.12,1802 600  
 2.5,4.2,10.0,0.0/,(UL5(J),J=1,5)/.265,.43,1.68,5.867375,0.0/ 802 610  
 DATA (C4(J),J=1,7)/-.724616,-.08,-.02,.416667,-.071254,-.007659,-.802 620

```

1001097/
DATA (C5(J),J=1,7), (C6(J),J=1,7), (C7(J),J=1,8), (C8(J),J=1,8)/.0804802 640
198,-.110,.137,-.220833,.277330,.180934,.029527,.167056,.1602,.0724802 650
2,.1450,.021749,.171847,.272444,.160875,-.606061,1.44,.097778,.0085802 660
392,0.0,-.0000083,76.202,1.813869,-3.554546,-3.16,-.542223,-.114293802 670
4,-0.00167,0.0,0.0/
DATA (C9(J),J=1,8), (C10(J),J=1,8), (C11(J),J=1,8), (C12(J),J=1,8)/2.802 690
10443,3.928788,3.22,1.944445,1.432133,1.064659,1.056289,1.0,1.15384802 700
26,-6.2972,-1.57143,-3.05556,.618+21,.0050067,-.00108276,-3,-.0038802 710
346,7.98454,3.03571,6.01944,-1.80976,-.0654667,.0224009,.0,0.14651802 720
44,-1.637971,-.345,-1.80889,2.35118,1.1148,.854207,1.0/
DATA (P2(J),J=1,8), (P3(J),J=1,8)/5*2,1,3,-3,7*2,-2/
DATA (A1E(J),J=1,41)/40,39.6,39.4,39.2,39.0,39.0,39.0,39.0,39.0,39.1,802 750
139.3,39.5,39.6,39.9,40.0,40.4,40.7,41.0,41.3,41.6,42.0,42.4,42.7,4802 760
23.0,43.5,44.0,44.5,45.3,46.0,47.0,48.0,49.5,50.8,52.3,54.0,56.0,58802 770
3.0,60.5,63.5,67.5,74.0,90.0/
DATA (CF(J),J=1,13)/1.0,.98,.96,.91,.83,.66,.47,.343,.265,.211,.1802 790
175,.143,0./
DATA (ACF(J),J=1,10)/25000.,50000.,62500.,75000.,87500.,100000.,11802 810
12500.,125000.,137500.,150000.,162500.,175000.,250000./
END
SUBROUTINE ATMOS (Z, TM, SIGMA, RHO, THETA, DELTA, CA, AMU, K)
*****
CALLING SEQUENCE
CALL ATMOS (Z, TM, SIGMA, RHO, THETA, DELTA, CA, AMU, K)
Z = GEOMETRIC ALTITUDE (FT)
TM = MOLECULAR SCALE TEMPERATURE (DEGREES RANKIN)
SIGMA = RATIO OF DENSITY TO THAT AT SEA LEVEL
RHO = DENSITY LB-SEC**2-FT**(-4) OR SLUGS-FT**3
THETA = RATIO OF TEMPERATURE TO THAT AT SEA LEVEL
DELTA = RATIO OF PRESSURE TO THAT AT SEA LEVEL
CA = SPEED OF SOUND (FT/SEC)
AMU = VISCOSITY COEFFICIENT (LB-SEC-FT**2)
K = 1 NORMAL,
= 2 ALTITUDE GREATER THAN 30000. FT.,
= 3 ALTITUDE NEGATIVE,
= 4 FLOATING POINT OVERFLOW,
= 5 ALTITUDE GREATER THAN 30000. FT. AND FLOATING POINT OVERFL.
*****
DIMENSION HPRIMB(11), TMB(11), SIGMAB(11), ALM(11)
DATA (HPRIMB(I), TMB(I), SIGMAB(I), ALM(I), I=1, 11)/0.,518.688,1.000000,
100E0,-0.00356616,36089.239,389.988,2.9706958E-01,0.,82020.997,389,
2.988,3.2665751E-02,0.00164592,15+199.480,508.788,1.2117870E-03,0.,
3173884.510,508.788,5.8677311E-04,-0.00246888,259186.350,298.188,5.
48677311E-04,0.,295275.590,298.188,1.7928595E-06,0.00219456,344488.
5190,406.188,9.3921519E-08,0.01097280,524934.380,2386.188,7.7658593
6E-10,0.01548640,557742.780,2566.188,5.6324677E-10,0.00274320,65616
77.980,2836.188,2.5726771E-10,0.0192124/

```

```

DATA Q/0.018744176/,RE/2.0955531597/,S/198.72/,PZ/2116.2/,AMUZ/3.7ATM 310
1372998E-J7/,RHOZ/0.0023769/,TMZ/518.688/ ATM 320
K=1 ATM 330
IF (Z) 13,30,20 ATM 340
10 K=3 ATM 350
GO TO 111 ATM 360
20 IF (Z.GT.300000.) K=K+1 ATM 370
30 HPRIM=(RE/(RE+Z))*Z ATM 380
DO 40 M=1,11 ATM 390
IF (HPRIM-HPRIMB(M)) 50,60,40 ATM 400
40 CONTINUE ATM 410
M=12 ATM 420
50 M=M-1 ATM 430
60 IF (ALM(M)) 70,80,70 ATM 440
70 TM=TMB(M)+ALM(M)*(HPRIM-HPRIMB(M)) ATM 450
SIGMA=EXP((1.0+(Q/ALM(M)))*(ALOG(TMB(M)/TM)))*SIGMAB(M) ATM 460
GO TO 90 ATM 470
80 TM=TMB(M) ATM 480
SIGMA=SIGMAB(M)*EXP(-(Q*(HPRIM-HPRIMB(M)))/TMB(M)) ATM 490
90 RHO=RHOZ*SIGMA ATM 500
THETA=TM/TMZ ATM 510
DELTA=SIGMA*THETA ATM 520
CA=49.02177*SQRT(TM) ATM 530
AMU=AMUZ*SQRT(THETA**3)*((TMZ+S)/(TM+S)) ATM 540
CALL OVERFL (J) ATM 550
GO TO (120,110), J ATM 560
100 K=K+3 ATM 570
110 RETJRN ATM 580
END ATM 590-
SUBROUTINE SETUP (X,NEXTR,ND,NA1,NA2,NA3,NA4,NA5,NA6) SEP 10
***** SEP 20
SUBROUTINE SETUP SETS UP ARRAYS FOR TABLE LOOK UP SEP 30
CALLING SEQUENCE- SEP 40
SEP 50
SEP 60
CALL SETUP(X,NEXTR,ND,NA1,NA2,NA3,NA4,NA5,NA6) SEP 70
WHERE SEP 80
X = TABLE OF INDEPENDENT VARIABLES SEP 90
NEXTR= 0 NO EXTRAPOLATION SEP 100
= 1 EXTRAPOLATION IS DESIRED SEP 110
ND = NUMBER OF DIMENSIONS (WHEN Z=F(X,Y), ND=3) SEP 120
NA1 = NO. OF VALUES FOR FIRST INDEPENDENT VARIABLE SEP 130
NA2 = NO. OF VALUES FOR SECOND INDEPENDENT VARIABLE SEP 140
NA3 = NO. OF VALUES FOR THIRD INDEPENDENT VARIABLE SEP 150
NA4 = NO. OF VALUES FOR FOURTH INDEPENDENT VARIABLE SEP 160
NA5 = NO. OF VALUES FOR FIFTH INDEPENDENT VARIABLE SEP 170
NA6 = NO. OF VALUES FOR SIXTH INDEPENDENT VARIABLE SEP 180
SEP 190
***** SEP 200
COMMON /TBLKUP/ L1,LF,NA(6),XL(100),NEX SEP 210
DIMENSION X(1),XA(6),NS(5),WJ(32),RATIO(5),NGROUP(5),ITOT(5) SEP 220

```

```

DO 10 I=1,NA1                                SEP 230
13 XL(I)=X(I)                                  SEP 240
NNEX=NEXTR                                    SEP 250
NA(1)=XA1                                     SEP 260
NA(2)=XA2                                     SEP 270
NA(3)=XA3                                     SEP 280
NA(4)=XA4                                     SEP 290
NA(5)=XA5                                     SEP 300
NA(6)=XA6                                     SEP 310
L1=2                                          SEP 320
LF=ND-1                                       SEP 330
RETURN                                        SEP 340
END                                            SEP 350-
SUBROUTINE MACURE (Z,XA1,XA2,XA3,XA4,XA5,XA6,IE,ZR)  MAC 10
*****MAC 20
SUBROUTINE MACURE EXECUTES AN N DIMENSIONAL TABLE LOOK UP  MAC 30
WITH EXTRAPOLATION IF DESIRED  MAC 40
CALLING SEQUENCE-  MAC 60
CALL MACURE(Z,XA1,XA2,XA3,XA4,XA5,XA6,IE,ZR)  MAC 80
WHERE  MAC 90
      IE = ERROR CODE  MAC 100
           0 INTERPOLATION SUCCESSFUL  MAC 110
           1 INDEPENDENT VARIABLES NOT IN ASCENDING ORDER  MAC 120
           2 FOR I=0, ARGUMENT EXCEEDS LIMITS OF TABLE  MAC 130
      Z( 1)= F(X1,Y1,Z1)      Z(13)= F(X3,Y1,Z1)  MAC 140
      Z( 2)= F(X1,Y1,Z2)      Z(14)= F(X3,Y1,Z2)  MAC 150
      Z( 3)= F(X1,Y2,Z1)      Z(15)= F(X3,Y2,Z1)  MAC 160
      Z( 4)= F(X1,Y2,Z2)      Z(16)= F(X3,Y2,Z2)  MAC 170
      Z( 5)= F(X1,Y3,Z1)      Z(17)= F(X3,Y3,Z1)  MAC 180
      Z( 6)= F(X1,Y3,Z2)      Z(18)= F(X3,Y3,Z2)  MAC 190
      Z( 7)= F(X2,Y1,Z1)      Z(19)= F(X4,Y1,Z1)  MAC 200
      Z( 8)= F(X2,Y1,Z2)      Z(20)= F(X4,Y1,Z2)  MAC 210
      Z( 9)= F(X2,Y2,Z1)      Z(21)= F(X4,Y2,Z1)  MAC 220
      Z(10)= F(X2,Y2,Z2)      Z(22)= F(X4,Y2,Z2)  MAC 230
      Z(11)= F(X2,Y3,Z1)      Z(23)= F(X4,Y3,Z1)  MAC 240
      Z(12)= F(X2,Y3,Z2)      Z(24)= F(X4,Y3,Z2)  MAC 250
*****MAC 270
COMMON/TBLKUP/L1,LF,NA(6),X(100),NEXTR  MAC 280
DIMENSION Z(1),XA(6),NS(5),WJ(32),RATIO(5),NGROUP(5),ITOT(5)  MAC 290
IE=0  MAC 300
XA(1)=XA1  MAC 310
XA(2)=XA2  MAC 320
XA(3)=XA3  MAC 330
XA(4)=XA4  MAC 340
XA(5)=XA5  MAC 350
XA(6)=XA6  MAC 360
DO 100 I=1,LF  MAC 370
L2=L1+NA(I)-2  MAC 380

```

	FOUND=0.	MAC 390
	DO 50 J=L1,L2	MAC 400
	IF (X(J).GT.X(J-1)) GO TO 10	MAC 410
	IE=2	MAC 420
	RRETURN	MAC 430
10	IF (FOUND.NE.0.) GO TO 50	MAC 440
	IF (XA(I)-X(J-1)) 20,5L,50	MAC 450
20	IF (J.GT.L1) GO TO 40	MAC 460
	IF (NEXTR.EQ.0) GO TO 30	MAC 470
	FOUND=1.	MAC 480
	NS(I)=L1-1	MAC 490
	GO TO 50	MAC 500
30	IE=-1	MAC 510
	RETURN	MAC 520
40	FOUND=1.	MAC 530
	NS(I)=J-2	MAC 540
50	CONTINJE	MAC 550
	IF (FOUND) 90,60,90	MAC 560
60	IF (XA(I)-X(L2)) 80,80,70	MAC 570
70	IF (NEXTR.NE.0) GO TO 60	MAC 580
	IE=1	MAC 590
	RETURN	MAC 600
80	NS(I)=L2-1	MAC 610
90	L1=L2+2	MAC 620
100	CONTINUE	MAC 630
C	IN NS(I) IS THE SUBSCRIPT IN THE ARRAY X SUCH THAT	MAC 640
C	X(NS(I)) IS LESS THAN THE ITH ARGUMENT	MAC 650
	DO 110 I=1,LF	MAC 660
	K=NS(I)	MAC 670
	RATIO(I)=(XA(I)-X(K))/(X(K+1)-X(K))	MAC 680
C	IN RATIO(I) IS THE RATIO OF X ARG, RATIO(2)=RATIO OF Y ETC.	MAC 690
110	CONTINUE	MAC 700
	NGROUP(1)=NS(1)	MAC 710
	NSUM=NA(1)	MAC 720
	DO 120 I=2,LF	MAC 730
	NGROUP(I)=NS(I)-NSUM	MAC 740
	NSUM=NSUM+NA(I)	MAC 750
120	CONTINUE	MAC 760
C	IN NGROUP(I) IS THE SUBSCRIPT OF THE ITH VARIABLE SUCH	MAC 770
C	THAT THE TABLE VALUE IS LESS THAN THE CORRESPONDING ARGUMENT	MAC 780
C	THIS IS IN TERMS OF THIS VARIABLE ONLY	MAC 790
C	FOR A FUNCTION OF DEGREE ND WE NEED 2**(ND-1) VALUES	MAC 800
C	FROM THE Z ARRAY	MAC 810
	ITOT(LF)=1	MAC 820
	DO 130 I=2,LF	MAC 830
	J=LF-I+1	MAC 840
	ITOT(J)=ITOT(J+1)*NA(J+1)	MAC 850
130	CONTINUE	MAC 860
C	IN ITOT(J) IS THE NUMBER OF LOCATIONS IN THE Z ARRAY NEEDED TO CHANGE	MAC 870
C	THE JTH SUBSCRIPT	MAC 880
	KF=2**LF	MAC 890

	MW=-2	MAC 900
	DO 170 I=1,KF,2	MAC 910
	IFIRST=1	MAC 920
	MW=MW+2	MAC 930
	DO 160 J=1,LF	MAC 940
	MM=2*(J-1)	MAC 950
	IF (MOD(MW/MM,2).EQ.0) GO TO 140	MAC 960
	IMON=NGROUP(J)+1	MAC 970
	GO TO 150	MAC 980
140	IMON=NGROUP(J)	MAC 990
150	IFIRST=IFIRST+(IMON-1)*ITOT(J)	MAC1000
160	CONTINUE	MAC1010
	ISEC=IFIRST+ITOT(1)	MAC1020
	WJ(I)=Z(IFIRST)	MAC1030
	WJ(I+1)=Z(ISEC)	MAC1040
170	CONTINUE	MAC1050
	DO 180 I=1,LF	MAC1060
	KF=KF/2	MAC1070
	DO 180 J=1,KF	MAC1080
180	WJ(J)=WJ(2*J-1)+(WJ(2*J)-WJ(2*J-1))*RATIO(I)	MAC1090
	ZR=WJ(1)	MAC1100
	RETURN	MAC1110
	END	MAC1120-
	SUBROUTINE GRAPH (IA, JM, IU, JC, IB, IL, IV, BR, AA, AP, BC, BS, AB)	GRA 10
C		GRA 20
C	THIS IS A DUMMY	GRA 30
C		GRA 40
	DIMENSION BR(1), AA(1)	GRA 50
	WRITE (6,10)	GRA 60
	RETURN	GRA 70
C		GRA 80
10	FORMAT (1H ,29H SUBROUTINE GRAPH WAS ENTERED)	GRA 90
	END	GRA 100-

APPENDIX II  
SAMPLE INPUT DECK

This appendix illustrates a sample input deck in figure 2. The reader may compare the data with the descriptions in tables 1 through 16. Output generated by QUANTO for this input appears in appendix III of AFWL-TR-73-242, "QUANTO--A Code to Optimize Weapon Allocations." The remarks which follow will merely point out several of the salient features of the input deck. Parenthetical line numbers on the left of each data card are used for reference.

The first two numbers on the first input data card indicate, respectively, that there are four airbases or targets and three candidate positions at which submarines may be located. The locations of the bases and their centroids and their numbers of runways and aircraft appear on cards 2, 4, 6, and 8. Strings of ones indicate the take-off sequences, since only one type of aircraft is included in this problem. Cards 12 through 14 give the candidate submarine positions, the number of submarines initially located at each point, the number of missiles per submarine, and the type of missile (or submarine) which is (or is permitted) at each point. By adding the numbers of salvos possible from all possible submarine locations, the user may compute the number of weapon groups as six in this example. Adjusting the array dimensions downward to accommodate only four targets and six weapon groups could result in an enormous core reduction.

Program options specified on the first data card include the optimizations of both the submarine positions and the aircraft beddown. The extended form of the output has been requested for illustration purposes only, and the user should never have occasion to request this voluminous output. The final number input on the first data card is the MODE option, which in this case requests that all QUANTO computations be attempted throughout all the optimizations, but if the user-specified time limit for this job is within 30 seconds, then all computational results are to be saved on an output magnetic tape for later use in continuing the job, if desired.



The data on cards 10 and 11 is self-explanatory. If there were more than one type of aircraft, there would be a card of the form of card 10 for each type of aircraft. Furthermore, since there are dual runway bases, two types of aircraft would cause the need to input eight take-off intervals, since there would then be four ordered pairs of aircraft-type numbers for single runway bases and four for dual runway bases. The eight take-off intervals would require two input cards, since each value occupies 10 columns but only 70 columns are read from each card.

Data cards 15 through 20 indicate data describing the SLBM and its flight time. Fourteen time/range pairs for the missile appear on cards 16 through 20 preceded by the pair count. The user has indicated a maximum range of 2000 NM in card 15 even though the time/range pairs extend up to a range of 3290 NM. In QUANTO, flight time for an individual missile will be computed for all targets falling within the range limits on card 15, using cubic interpolation and extrapolation, if necessary, on the time/range pairs, and targets outside the range limits will not be attacked by that missile. If there were more than one missile type, a set of cards like cards 15 through 20 would have to be input for each type missile.

The aircraft profile and nuclear effects parameters are input on cards 21 through 46. The user has labeled the profile in an unused field in the "count card," card 22, which also indicates that 20 range/time/altitude/velocity/acceleration sets are to follow. In these cards, the user has adjusted the altitude values to be strictly increasing, although this was unnecessary. Also, since card 21 indicates a level-off altitude of 5000 feet, the profile data contains an acceleration value only on the card having 5000 feet altitude. The aircraft will be accelerated from Mach 0.7312 to Mach 0.849 after leveling off according to the parameters on cards 42 and 43. The nuclear effects parameters appear on card 44 followed by two blank cards indicating that the lethal radii and the time of shock arrival are not known from previous runs of similar problems. If there were two types of aircraft for this problem, a set of cards similar to the set of cards 21 through 46 would have to be input for each aircraft type. If there were more than one missile type, a pair of cards like cards 45 and 46 would have to be input for each missile type, even if the lethal radii are unknown.

The convergence parameters and the initial missile allocation which starts the iterative procedure are input on cards 47 through 52. In the initial allocation, the two numbers on a single card correspond to the two salvos from a single submarine, and the five submarines are in the order indicated by cards 12 through 14. The initial laydown, for example, has the second submarine at the second submarine location firing its two salvos at targets one and four in the order of launching.

While this example does not exhaustively illustrate all input options, the user can use figure 2 as a guide to constructing the input deck.