UNCLASSIFIED

AD NUMBER

AD815897

NEW LIMITATION CHANGE

TO

Approved for public release, distribution unlimited

FROM

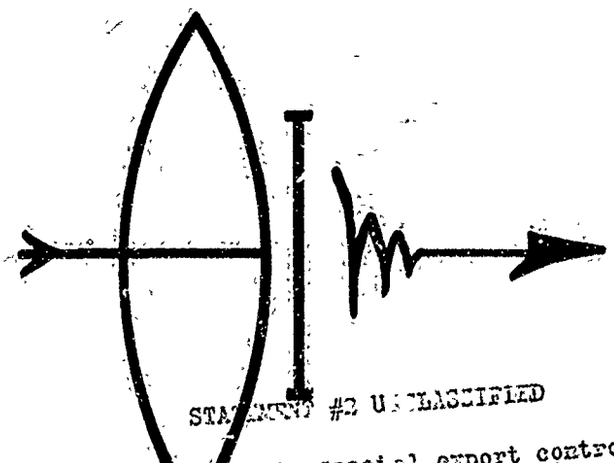
Distribution authorized to U.S. Gov't. agencies and their contractors; Critical Technology; NOV 1966. Other requests shall be referred to Air Force Systems Command, HQ, SCLA. Andrews AFB, DC 20334.

AUTHORITY

afsc ltr, 21 dec 1971

THIS PAGE IS UNCLASSIFIED

AD815897



STATEMENT #2 UNCLASSIFIED

This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of AF WESC (SCAS) Andrews AFB Wash. D.C. 20331

DDC
JUN 28 1967
C

ACADEMY FOR INTERSCIENCE METHODOLOGY

CHICAGO, ILLINOIS

DISCLAIMER NOTICE

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

ACADEMY FOR INTERSCIENCE METHODOLOGY

A nonprofit corporation devoted to scientific and educational purposes

MUSEUM OF SCIENCE AND INDUSTRY
CHICAGO, ILLINOIS 60637

Area Code 312
Phone: 667-0240

BEACON II

Description and Programs

by

Norman H. Painter

AIM 67-T-10

November 1966

Copy 53

Contract AF 18(600)-2793

CONTENTS

Introduction

Section I

ATLAS 1-1

STRIKE 1-6

MOTION 1-11

DEFEND 1-14

ASSESS 1-19

SHELL 1-32

SURVEY 1-35

COMAND 1-36

Utilization of the Programs 1-37

Section II. BEACON Main Routines for the IBM 7094

Section III. Supporting Programs

Functions 3-3

Subroutines 3-10

INTRODUCTION

BEACON denotes a set of programs organized to form a computing tool to study defense of the continental United States against strategic nuclear attack. The work has been predominately supported by the USAF, Headquarters Systems Command under contract AF 18(600)-2793. The purpose is not to evaluate current force structures (although this is not an unreasonable application), but rather to obtain a feel for the possibilities of newer technological concepts. Wild fresh ideas may form the other end of the application spectrum, though it is usually advisable to trim them down first by traditional study phases.

Thus, BEACON has not been designed toward the study of a particular weapon system or even type of systems, but (in intent at least) toward embracing the whole set of potentially interesting interacting systems of the near future. Since it is difficult for a small set of people to be good visionaries over such a vast realm, the method of procedure has been to concentrate on fundamental logical or physical processes of attack and defense on a global scale and to keep the structure modular, replaceable, and open-ended. It has not been easy. A distinction must be made between the general procedure that may be termed BEACON and any particular representation of it that may exist at a particular time such as BEACON I or II or III, etc. It is inevitable that representations be incomplete or unsuitable for the "next"

application. Success is often a combination of existence and adaptability. In this light the representation included here can only be regarded as a transitory example of the general procedure.

BEACON is not a model, at least not according to the usual thoughts that the word model evokes. The germanic concept that disinguishes a model is the occurrence of a central or main program that controls the processing flow according to some built-in logic which imitates or captures the essence of some real life activity. A key word lies in built-in. The fact that most models permit selective control or parameter variation through inputs at run time, cannot destroy the essential feature of a definable pre-understood logical path. By contrast BEACON substitutes for the central program a monitor which controls during execution other semantic component programs for the user in the way the user specifies. The entity takes the form of a computation by request, in the order of request, and to the extent requested. It is not farfetched to say that BEACON passes on to the user the joy (and the task) of "writing" his own main program at each execution. The method is simple and flexible enough to suit most user type needs without the elaborate procedure a programmer uses of actually writing and compiling a central program. The user pays an initial price in understanding and organizing his problem, but gains a degree of freedom and partica-pation. Although the subject matter obviously prohibits classifying BEACON as a general purpose software, within its scope this is its spirit.

SECTION I

Since BEACON does not have a prescribed logic, one cannot draw a very informative flow diagram. It seems best in this circumstance to discuss each of the main programs and then to illustrate some ways they may be exercised on defense problems. The function of the monitor BEAM is only to aid in effecting the latter and will be discussed later.

ATLAS

Almost always ATLAS will be the first program a user will wish to perform. The job which is accomplished by ATLAS is to initially set up in core memory what is called a facility record. The information content of the facility record is to be a description of the status of the geographical and physical characteristics of the globally fixed elements essential to the battle. Included in the record will be space for data from some of the other programs. The elements are grouped into sets with each set given a name by which it is referred. What facilities are necessary is solely dependent upon the problem being studied and the programs to be used subsequently. Typically there will be sets to define the targets of the attacker. These may be counterforce, countervalue, or defense sites. US offense bases are often included for this reason. If there is to be an active defense there will be one or more interceptor and/or sensor sets to affect the various battles. Still a third reason

for inclusion of a set may be for damage assessment comparisons. ATLAS decks for the 213 US and 26 Canadian urban areas have been constructed since they are so commonly used for damage assessment.

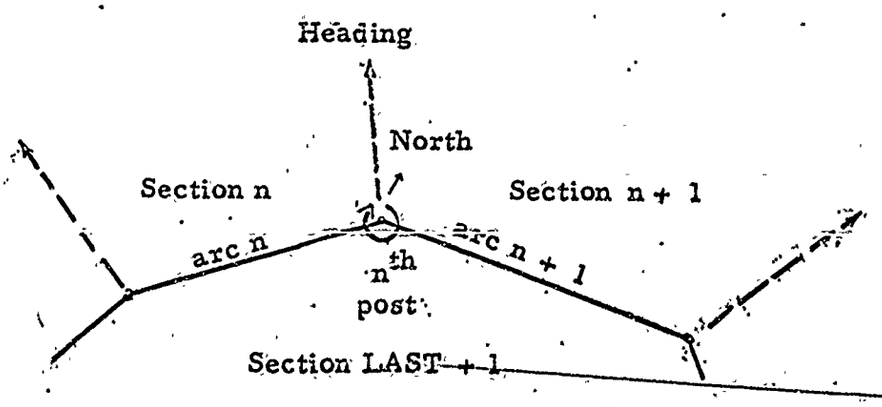
Thus, the facility record is an open-ended user defined concept. The ATLAS data deck required to supply the detailed information is by far the largest part of the typical input data. The common way of generating such a data deck is to add, delete, or otherwise shuffle a previous deck. With this in mind BEACON is so constructed that it is not necessary for the user to preserve or remember any special order nor to supply counts either of the sites within a set or of the sets themselves. It is incumbent upon the user to remember the names he has used. Another feature which makes it easier to generate an ATLAS deck from cannibalizing existing decks is the personal order and format specification each facility set is permitted. Source data may be presented in a wide variety of forms and more often than not the formats are mixed in any given run. It is then possible to input data for a set of airfields as : name, longitude, latitude, while for urban areas the list may read: population, name, latitude, longitude, area, . . .

The specific information required about each element depends in part upon the nature of the set. Currently five distinct kinds of

facilities are recognized in BEACON II: sites, blocks, places, barriers, and meshes. Sites are simple homogeneous elements which differ only in location and value, such as may suffice for many types of military bases. The sites of a given set are assumed to have the same physical radius and hardness for targeting and damage assessment purposes. Blocks may be used conveniently when a cluster of point targets are close enough together to be designated as a single squadron or farm and yet far enough apart to require separate targeting, as for example hardened missile silos. Each block is specified as to location, size and unit value. When required each block of a given set will be automatically deployed into a squarish array of the specified number of points around the center of the squadron "headquarters". The hardness and distance between points are set characteristics. Places are used to designate those elements, notably urban areas, which cannot be reasonably regarded as the same size. The individual information in this case includes not only location and value (population), but specific area in square nautical miles. For the three kinds of facilities so far sites, blocks, and places it is optional but recommended that an alphanumberic designation be also stored to identify later printouts.

Barriers and meshes are imaginary facilities; that is they are not targetable nor assessable. Their purpose is to allow the user to introduce a topology into the battle in a form suitable to his problem.

A barrier is formed by simply giving a list of fence post locations, i.e. latitudes and longitudes. In this case the order of the list is meaningful. ATLAS will automatically construct the barrier by connecting great circle arcs between successive posts and will close the ring by connecting the last post back to the first. As the barrier is traversed in ascending order, the arcs are numbered the same as the posts being approached and outward direction is taken to be to the left. Barriers often consist of a dozen or so arcs and may represent such things as probable detection lines, extent of fighter coverage, coastlines, etc. A mesh is an extension of the barrier concept by adding to each post a directional heading. Headings are specified as degrees clockwise from north and normally they are positioned to divide the outward sectors formed by the adjoining arcs. The exterior of the ring is thus subdivided into sections which are numbered the same as the corresponding arc (see illustration). ATLAS constructs great circle arcs so as to be tangent to the headings, and they



form the global extensions of the headings. (A heading of due EAST does not delineate the latitude circle at that point but rather the great circle tangent to the latitude circle.) The interior is also taken to be a meaningful section designated as LAST + 1. Meshes may take a variety of interesting forms from simple pie-shaped sectors with no interiors (posts all at the same location), or a representation of the NORAD defense sectors, or even an inward spiral forming several layers in an ingenious state-like map. The information about barriers and meshes is converted and stored in vector component form for more rapid utilization by other programs.

With proper key indications ATLAS will write the constructed facility record on tape one (as the next logical record). One may also have ATLAS retrieve a previous written facility record from tape one with or without further card extensions. As many as a thousand facilities in up to thirty sets may be placed in one record.

STRIKE

The STRIKE program can perform a number of interesting tasks related to targeting. The principal task is to make a reasonably coordinated pre-war allocation of various portions of a mixed stockpile of offensive warheads against selected sets of targets under various restrictions that the attacker might wish to impose. The end result of this consideration is the creation of a warhead record. This is a listing for each of the warheads used of their generated aimpoints with provision for later insertion (by other programs) of origin of launch and probability of kill before reaching target. The record is segregated by warhead types and contains its own keying information so that it may be read later if stored.

Warheads may be grouped into different types because of actual differences in warhead characteristics such as yield, CEP, etc. or because the attacker would prefer to use them in different waves (times) or for different objectives or delivered by different carriers. Up to two thousand warheads split into as many as twelve different types may be considered in any one allocation and placed in the same record. The attack, or at least this portion of it, is described by specifying the total number of warheads of each type, the order in which the types are to be allocated (not necessarily the order of arrival), their expected probability of abort and CEP.

STRIKE presumes that there is an appropriate facility record in memory containing the sets desired to be targeted. The chosen sets are requested in any desired order and for each target set - warhead type combination the attacker's desired constraints are stated. The main restriction is the most number of warheads of this type that will be allowed to any one facility of the set. A blank or zero here means no interest in this combination and the program proceeds on to those combinations where positive allocations are permitted. Where and within the stated limit allocation is possible, two basic choices are open for determining the relative allocation amongst the contending targets - requirement targeting or value targeting.

If the requirement option is chosen, the attacker is permitted to specify the least number of warheads he would like allocated to any target of the set. If least is the same as most, the user has completely determined the allocation for this combination regardless of other factors except the stockpile size. If a tolerance exists between the least and most bounds, a third number - a required probability of kill for each facility in the set - resolves the allocation to within fractional warhead round off (which may also be controlled). The probability of kill level is however the weakest of the constraints and will not be held to when it lies outside the stated bounds.

When targeting by value options are permitted, these are considered only after the particular warhead type has fulfilled its requirement obligations. The remaining stockpile is then available for distribution by value. The object is to maximize the total value destroyed. The allocations of previous warhead types leave the target complex with a reduced set of (expected) values. Targets within the currently allowed sets compete with each other, whether of the same set or not, based both upon their remaining values and their respective softness to the current warhead type. Softness is computed for each potential target as the average single shot probability of damage, based upon a specified matrix of weapon radii for each warhead type - facility set combination, and previous specifications for warhead CEP, probability of abort, and target sizes. If desired for certain warhead types, the attacker's estimates of the relative penetrability of the defense to specific targets can be introduced as an additional modifier of the softness. The value returned then is the product of the target's remaining value and its softness. In maximizing the value returned for the stockpile available only integral distributions of warheads are permitted. The mathematics underlying the program for finding the allocation has been given elsewhere[1].

---

[1] Academy for Interscience Methodology, AIM 65-T-18, Targeting Program Revised, by N. H. Painter, December 1964, pp. 5-10.

Once the allocation of the warheads is completed the list of aimpoints is constructed. For targets small in area the aimpoints coincide with the target coordinates and are taken from the facility record. For each missile block the appropriate array is generated from the ATLAS parameters and the aimpoints spread out over the array in sequential fashion as far as the allocation allows. For large area targets aimpoint patt ns for each warhead type are computed to "cover" the target as uniformly and multiply as the allocation permits.

For some purposes it may be desirable to preserve the warhead record as it now exists in the form of unaborted aimpoints. This may be done by using blanks for the final two STRIKE data cards and setting the keys so STRIKE will write the record on tape 3. An alternative would be to proceed and read in a new set of probability of aborts and CEP's, which may be the same or different from the previous estimated ones. In this case STRIKE will convert the aimpoints to impact points with the prescribed randomization and will initialize each warhead's probability of kill as starting out with the specified probability of abort. If desired STRIKE will immediately Monte Carlo this set of probabilities into random go-no-go choices. The warhead record in its completed form may then be preserved on tape. In either case whether the record is put on tape or not it remains in memory.

STRIKE might also be called in order to retrieve one or more warhead records from tape as needed thus obviating the need for reinputting and recomputing attacks which can be used multiple times. When this kind of call is made the option exists of either accepting the record as is or setting up a new abort and/or impact pattern. The latter would be a particularly convenient manner of performing Monte Carlo sensitivity analyses.

Finally one may call STRIKE again in the same run after a certain amount of damage assessment has been recorded on a facility list and request additional targeting be generated. The updated values of the targets selected will be used (whether the damage was of the direct or colocation nature) instead of the pre-war values. This presumes of course that the attacker has gained knowledge of this damage sufficiently well and in time to incorporate it in this targeting round. Since a new warhead record is hereby generated, the offense force structure involved may be more of the old force or contain new types or both. Thus as one progresses from one warhead record to the next, references to warheads of type one and two for example may retain their meaning while types three and four have changed.

MOTION

The MOTION program relates sets of facilities and warheads to one another in various ways. In doing so it makes extensive use of the barriers and meshes which have been set up by ATLAS. Most of the global geometric relationships are explored by means of the MOTION program. Besides generating information for its intrinsic value certain specific information may be useful in DEFEND, particularly for area defense systems. Four distinct functions are currently included within the program, each explained in the following paragraphs. None are restricted by frequency or order of utilization.

By selecting a particular mesh and naming other facility sets, the sites within each named facility are located according to the sector containing their latitude and longitude. A count of the sector distribution of sites is made and printed along with a listing of the sector number for each site. The sector designation for the site in the facility record is hereby tagged with the latest determination. Facilities which are not found in the facility record currently in memory are ignored. If the mesh is not in record, return is made to BEAM.

By selecting a particular mesh and naming certain facilities which are suitable for launch sites for selected warhead types, each warhead in the sets chosen will be tagged in the warhead record with an

appropriate launch site. The method for determining the launch site starts by locating the sector number of the named mesh that contains the aimpoint or impact point of the warhead. The sector number is then used to find a corresponding site in the launch set. If, for example, a warhead fell in the third sector, the third launch site of the specified set would be presumed to be the desired and appropriate site. There should be atleast as many sites in the launch sets as there are sectors; however, a given site may be listed more than once, and a given set may be used for more than one warhead type. This method is particularly attractive when range is not the dominate routing criterion and the user wishes to do his own broad brush positioning of launch or penetration points. If the warheads are to be defended against by sector oriented defenses, this should be the method for sector tagging the warheads using the same mesh as the defense. Otherwise, the mesh need not correspond and could be specially designed for the routing topology.

Without using a mesh but naming launch facilities for other warhead types together with maximum range constraints for each type, a different kind of selection of launch sites will be made. In this method the launch site which is closest to the impact point from among the designated facilities is found by computing the great circle distances. If the range is within the allowed max this association is accepted. If not, the particular warhead is given an artificial "out-of-range" kill and will

not be seen by later defenses or cause any damage assessment. The out-of-range feature can under certain circumstances be a planned phenomenon for achieving a modified targeting plan. Short range weapons used for hold down or defense roll up might well be targeted this way. Counts of the out-of-range kills as well as the loads placed upon the penetration or launch points in order to meet the in range targets are printed out.

By naming appropriate barriers for selected warhead types, for whom launch sites have been previously designated, information relative to the attack crossing these barriers can be obtained. For each warhead the position where the great circle arc from launch to target crosses the selected barrier is found. Not every warhead may cross. For some the target may lie outside the barrier, i.e. Honolulu, while others may be launched interior to the barrier. Other paths may cross more than once in which case the first computed crossing (lowest numbered arc) is used. For each arc of the barrier the sum of the probabilities of survival of the penetrators crossing that arc (surviving penetrators if Monte Carlo has been made) is printed out as well as the weighted average of time to go from arc to target. This information is not however currently retained in memory.

DEFEND

The DEFEND program is called for the single minded purpose of increasing the attrition on the attack. The program may be called at various times and in each call selected sets of defense facilities may be requested to do battle in arbitary order or repetition. A variety of local area, missile, and aircraft defenses may be included. In each defense battle the defense facility is faced with the presence of a possibly mixed selection of offense types in a common time span.

Several geographical mechanisms are available for confining the general commitment of each defense site or sector. Within the confined environment the efficacy of the site's normal attrition capability becomes limited with increased rapidity of events as the total raid is compressed in time. The actual locale of the battles are not sought nor are detailed intercept assignments or geometry computed. Given reasonable computation of forces which the defense can mount and the segments of the offense they are directed at it is not unreasonable to manage the intercept variations as a random application of kill potential to moving targets.

Each defense site listed in the atlas facility takes its turn at doing battle in order. The simplest commitment mechanism is that appropriate to local terminal defenses. The sites have a circular radius of coverage around their physical coordinates. This coverage may vary

with penetrator types including shrinking to nothing at all. Those warheads which are scheduled to impact within the coverage and which are still alive at the time the site is ready to commit constitute its threat. Note the coordination which is thereby achieved by overlapping sites since a first site kill will not be a second site target but a first site failure will.

A second commitment mechanism is more appropriate to area missile defense as such sites are not likely to be deployed near targets they defend. In this an effective circular intercept coverage around the sites are assumed as before but this time at some altitude. The footprint or ground projection of the equivalent coverage depends upon the direction and angle of reentry of the missile attack. With such information for the various missile types the down range displacements are inserted. A similar type coordination occurs when footprints overlap.

Either of these geometrical partitioning may of course be applied to aircraft defenses, particularly of the Hercules or Bomarc type. For long range fighters the facility "sites" might be taken as centers of NORAD sectors of responsibility or air cap positions displaced up range to the center of battle coverage, although the "reentry" is awkward in the latter case. A third procedure is to request defense by sector. This entails that a mesh appropriate to the sectorizing be included in ATLAS, the defense facilities located by sector by MOTION, and the defendable penetrators sector tagged by MOTION with the same mesh. Each site

then battles those penetrators with the corresponding sector designation. A fourth procedure is to use a simultaneous constraint of radius of coverage and sector coincidence, but an application is not apparent.

Whatever the commitment partitioning procedure is, an actual count of surviving penetrators is made. For the defense environment point of view this count is modified. If, for the span of time allocated to this raid, a group of penetrators can be presumed to be all in the battle area and remain there throughout each then has the same dwell time as the raid time. If a second group of fast moving penetrators all pass through the battle area during the raid interval it is assumed that they enter and leave at random times. Each has a dwell time that is only a fraction of the raid time and the number present at any moment will be on the average this same fraction. A third group of penetrators may enter and leave the battle area at random moments, not because of their speed, but because their entries are straggled out over a long time relative to the raid interval. Their dwell time should also be shortened to reflect the lower density present during the raid. However, when handled this way the defense only receives justice if repeated raids are called on the stragglers. Generally little is lost computationally by time collapsing a straggled raid into one separate simultaneous raid if saturation and inventory ready allowances are made.

The count of penetrators present may be further reduced by a visibility factor or increased by the portion of the cloud of false bogies

accompanying the visible penetrators which cannot be successfully discriminated against by the particular defense. This total density of bogies persist throughout the raid time although for fast penetrators the identities may be changing. The number of tracks a site or sector is able to maintain generally has an upper limit so that the probability a given bogy is a track decreases as the number of bogies approach the saturation limit. This probability of track is taken of the form

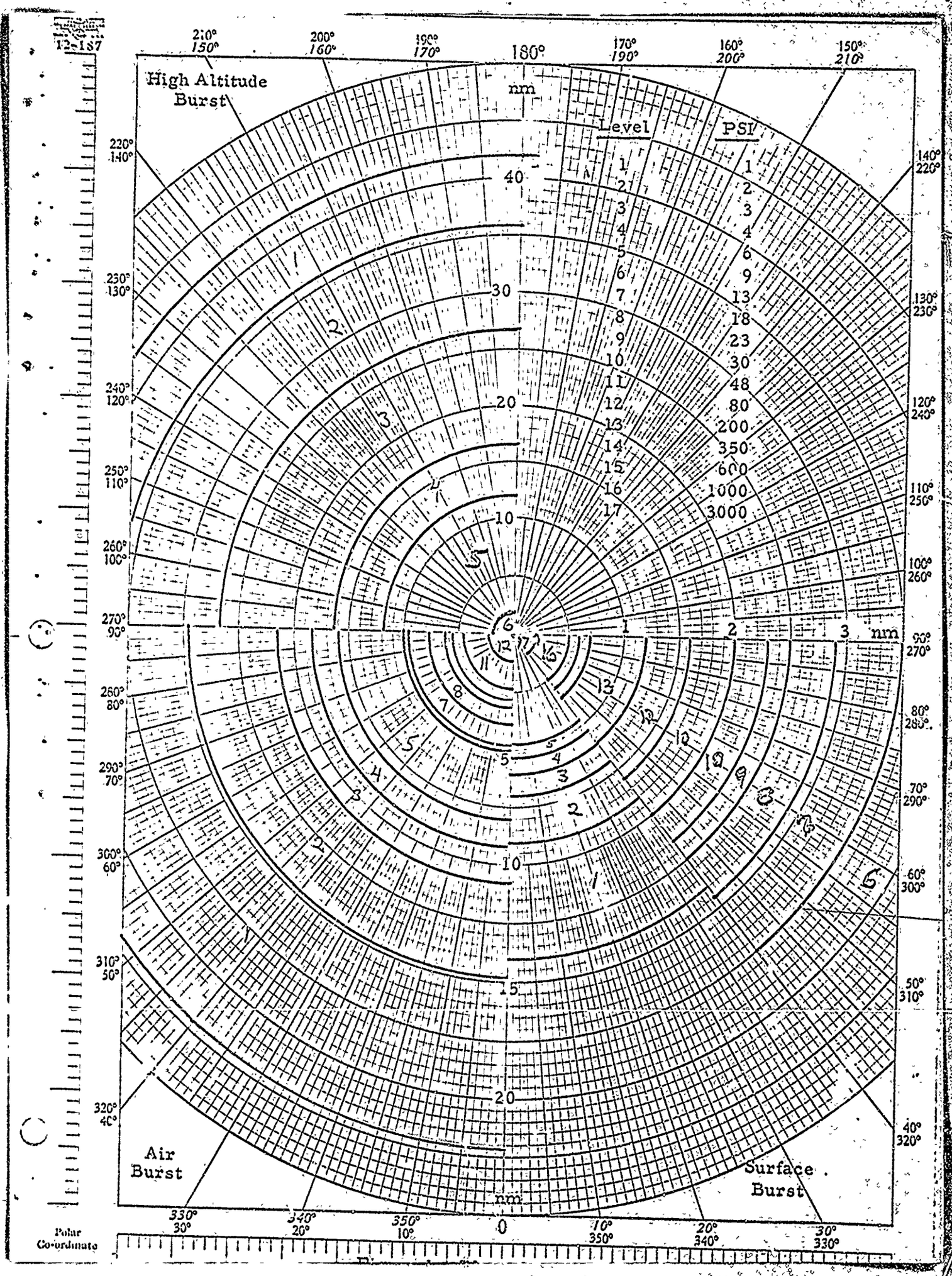
$$PT = e^{-\frac{1}{2}\left(\frac{\text{BOGIES}}{\text{SATURATION}}\right)^2}$$

If the bogies greatly exceed the limit the number of actual tracks may in fact decrease. Each intercept made on a track requires a ground data link or communication channel, an interceptor salvo and a certain amount of time. The total number of intercepts made during the raid may be limited by channel duty rates, interceptor ready inventories or failing either of these a command limitation when a satisfactory number of intercepts per distinct bogy is attained. (If the average dwell time is half the raid time, the number of distinct bogies will be twice the number that are on the average present.) The interceptor inventory may or may not be reques' d to be counted down as intercepts are made. There are rational applications for both missile and fighter type defenses for either request. The number of intercepts per track per unit time is computed.

The probability of survival of every warhead within the coverage decreases in relation to the probability of kill of the penetrator type per intercept, the number of intercepts per dwell time, and the probability of being tracked. If desired the expected probabilities of kill may be left to accumulate through several battles or they may be Monte Carloed each time. It is generally desirable to have each warhead Monte Carloed at least the last time before entering the damage assessment program. The expected number of defense kills and opportunities for each site and penetrator type is printed out for each battle.

ASSESS

The ASSESS program manages all the damage assessments desired on any or all real facilities whenever warheads have passed through the defense. Facilities do not have to have been targets and colocation damage is accounted for. When it is called into memory by the BEAM monitor, the next four input cards are read. These cards establish the desired yields (in megatons), the thermal efficiency adjustments (normally unity), the fission-fusion fractions, and the altitude type of burst (surface, air or high altitude) for all of the various warhead types. The program contains numerical data which define blast contours, thermal contours, and fallout contours. This basic data is for a 10 MT warhead with 50-50 fission-fusion. For blast these contours are concentric rings of lbs per square inch overpressure at various nautical mile distances from ground zero (Figure 1) For thermal they are rings of incident calories per square cm at various nautical mile distances (Figure 2). For fallout the contours are a nested set of oblong rectangles skewed toward the down-wind (easterly) direction, for total roentgens received in 48 hours with no protection (Figure 3). The blast and thermal contours change for each of the three different altitude type bursts, while the early fallout is assumed to be appreciable only for surface bursts. For a particular warhead


12-187
High Altitude Burst
Air Burst
Surface Burst
Level
PSI
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
1 2 3 4 6 9 13 18 23 30 48 80 200 350 600 1000 3000
nm
Polar Co-ordinate

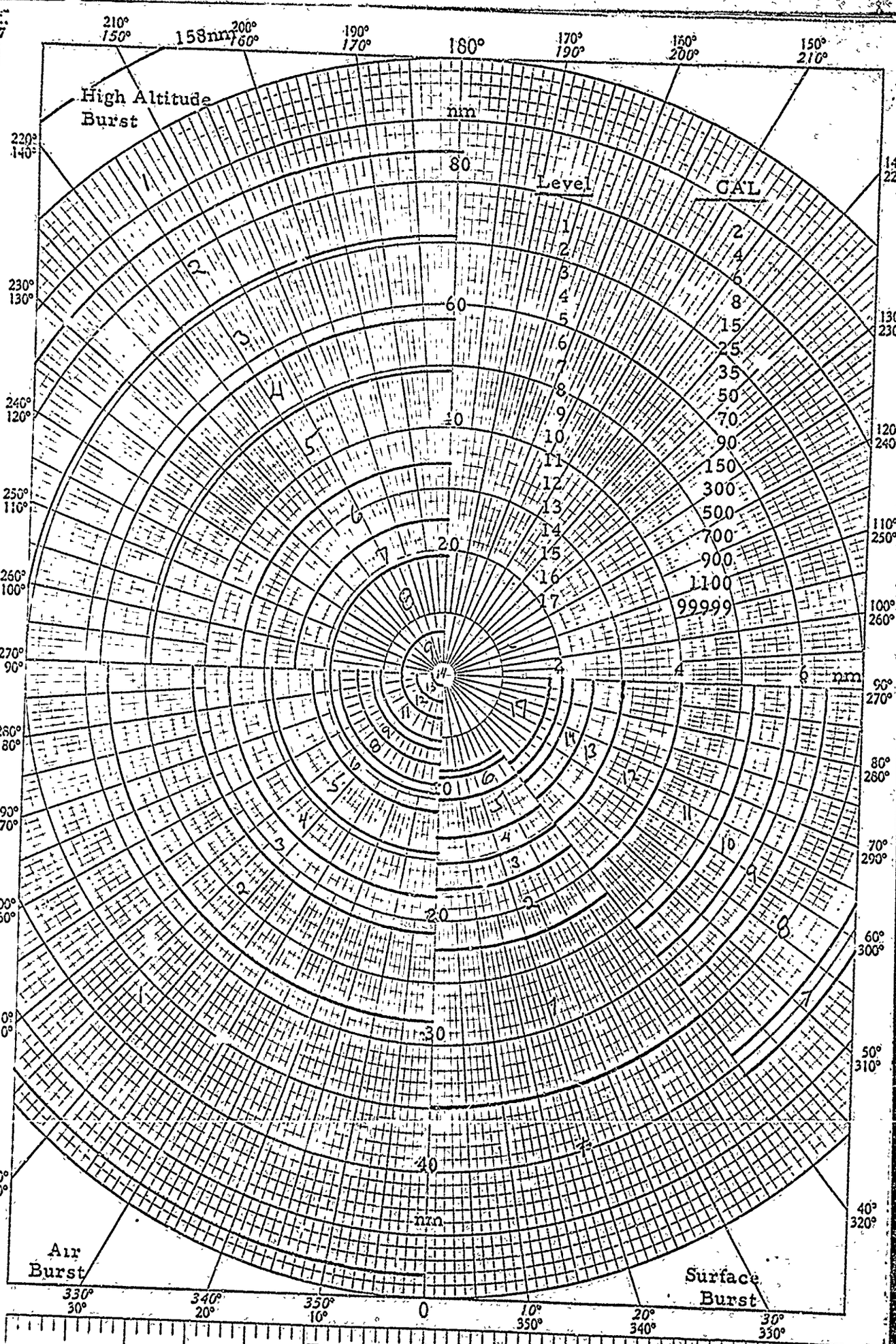
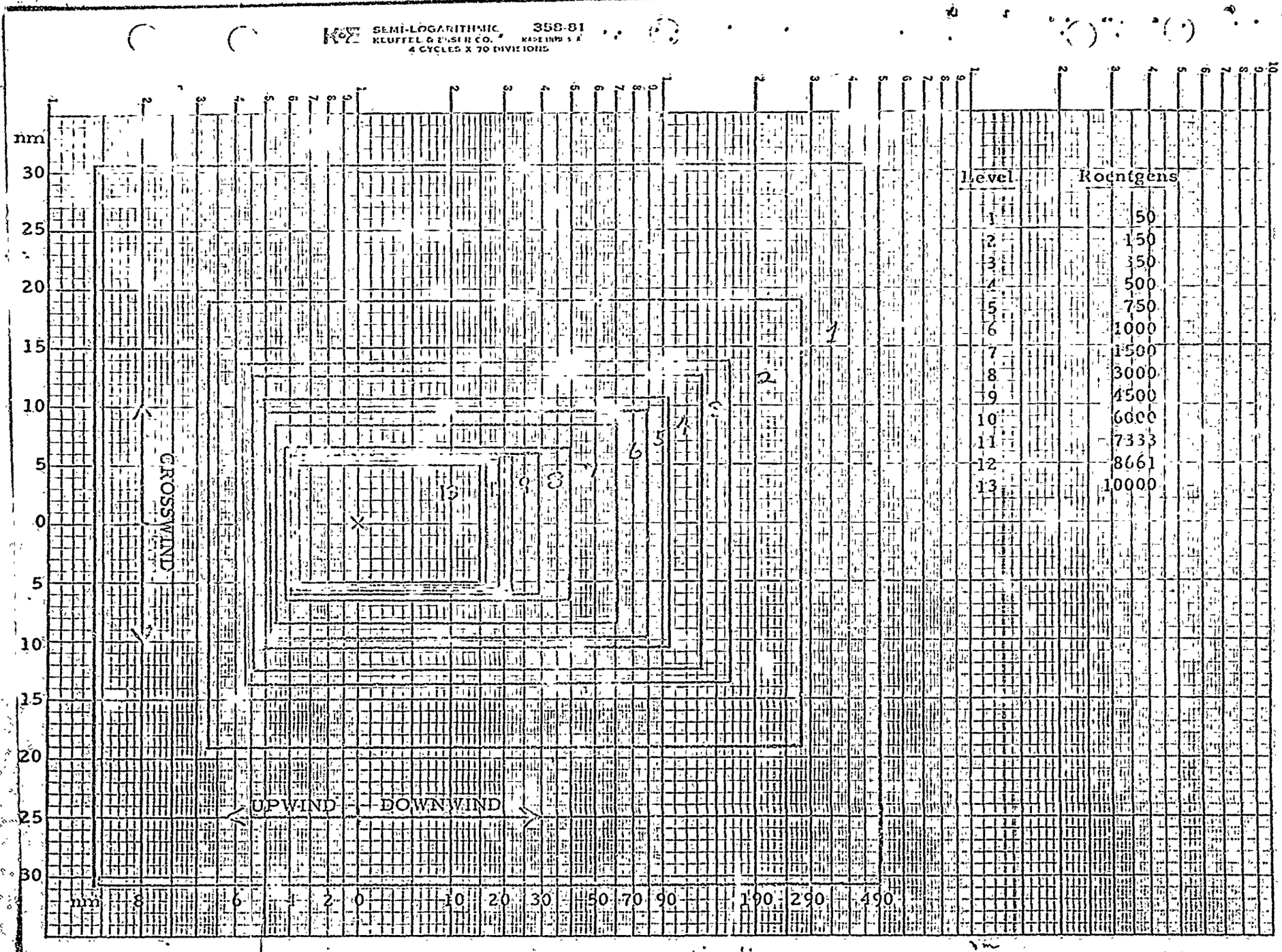


Figure 2 Thermal C...


K&E SEMI-LOGARITHMIC 358-81
KEUFFEL & ESSER CO. MADE IN U.S.A.
4 CYCLES X 70 DIVISIONS
nm
30
25
20
15
10
5
0
5
10
15
20
25
30
CROSSWIND
UPWIND
DOWNWIND
nm 8 6 4 2 0 10 20 30 50 70 90 190 290 490
Level Roentgens
1 50
2 150
3 350
4 500
5 750
6 1000
7 1500
8 3000
9 4500
10 6000
11 7333
12 8661
13 10000

type the appropriate altitude contours are scaled to the correct yield; for blast and fallout the scaling is by yield to the 1/3 power, for thermal by yield to the .42 power.

A particular set of warheads are withheld from consideration during the current assessment if their altitude types are set to a negative one. This permits different offensive components of the same strike to be assessed by different assessment calls. One advantage of this procedure occurs when the strike contains some time delayed elements and the defense survival and effectiveness can be studied and altered between waves. An alternate procedure which is some times useful is to construct one "strike kit" such that different combinations of warhead types represent different threat levels or targeting objectives. Each assessment then represent the effectiveness of one of the threats. In one application, the low threat was obtained by taking only warhead types 1 - 5, the medium threat was obtained by augmenting the low threat with those warheads called type 6, and the high threat by the further augmentation of warhead type 7. Warheads of type 5, 6, and 7 were all physically alike, the distinction being only in the extent of targets hit.

When the warhead initializations are complete, requests to assess damage to classes of facilities are processed. Each subsequent input card requests one class in the facility atlas be assessed and

supplies the levels of blast, thermal, and fallout stress (or protection) considered to be acceptable for that class. On the basis of the class name supplied (and previous ATLAS information) the subroutine distinguishes the need for five different procedures, depending on whether the class represents 1) bases or sites with similarity in area and function, 2) strategic missile squadrons or "blocks", 4) urban areas with shelter data (on tape prepared by the SHELL subroutine), or 5) "rural population" or continental map of values. Most military installations, with the exception of missile blocks, are usually considered to be treatable by the first procedure, and this is the one discussed in detail. The area of the base (πR^2) times the density of evaluation points per square nautical mile (both inputs) determine how many evaluation points are to be placed around each base. This number is at least one but no more than 400. Whatever the number, a squarish pattern of that many points is constructed, centered on each base and scaled to cover the equivalent area of the base (Figure 4).

For each base in the facility class, the list of warhead impact points is examined. Warheads which have aborted, or been successfully defended against, or disallowed (by the altitude ban mentioned above) are skipped. Those impacts which are too remote to affect the particular bases are also skipped. For each remaining

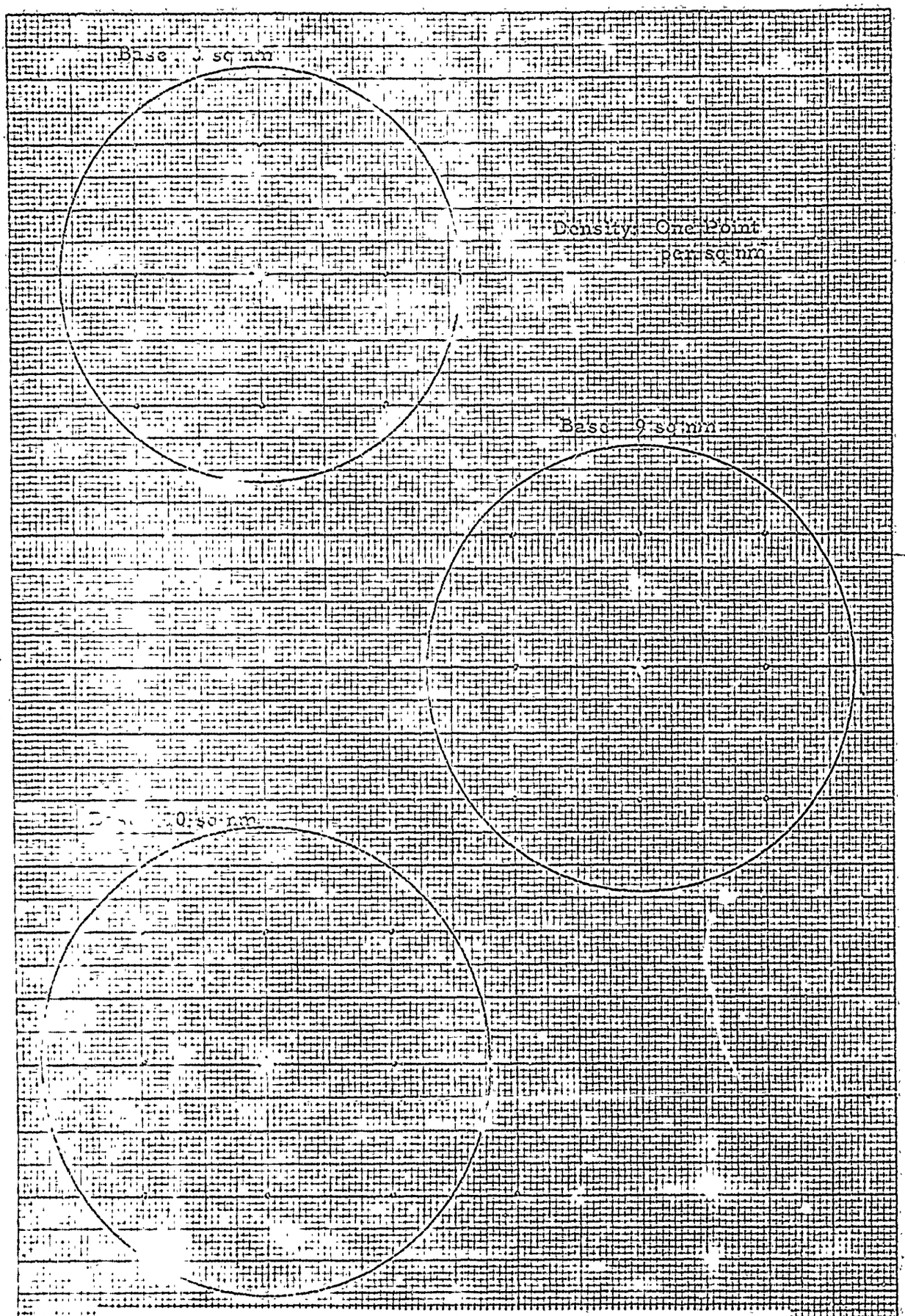


Figure 4. Pattern of Evaluation Points as Base Size Increases.

warhead the appropriate effects contours are placed around the impact point. The location of each of the base evaluation points is found with respect to the three sets of contours and the respective PSI, calorie, and roentgen doses that would be received from each detonation at the evaluation point are weighed. For blast and thermal doses the point is charged with receiving the maximum levels found among all the warheads detonating while for fallout the dose is the cumulative of each warhead's contribution. In order to present to the analyst an indication of the effects experienced on each facility of this type, the three types of dosages are averaged over all the target's evaluation points and these averages are pointed for blast, thermal and fallout before target vulnerability to the "absorbed" effects are considered.

When the doses have been computed for all the evaluation points around the base, the base survival is considered. The probability a particular evaluation point, considered to represent a portion of the base, survives is determined from functions of the effective doses received, which are the actual doses relative to the (input) stress limits for the facility class. For blast and thermal the functional relation is e^{-x^2} (Figure 5), while for radiation it follows a fatality table (Figure 6). The fraction of the base as a whole which is killed by blast is the number of evaluation points which do not survive the

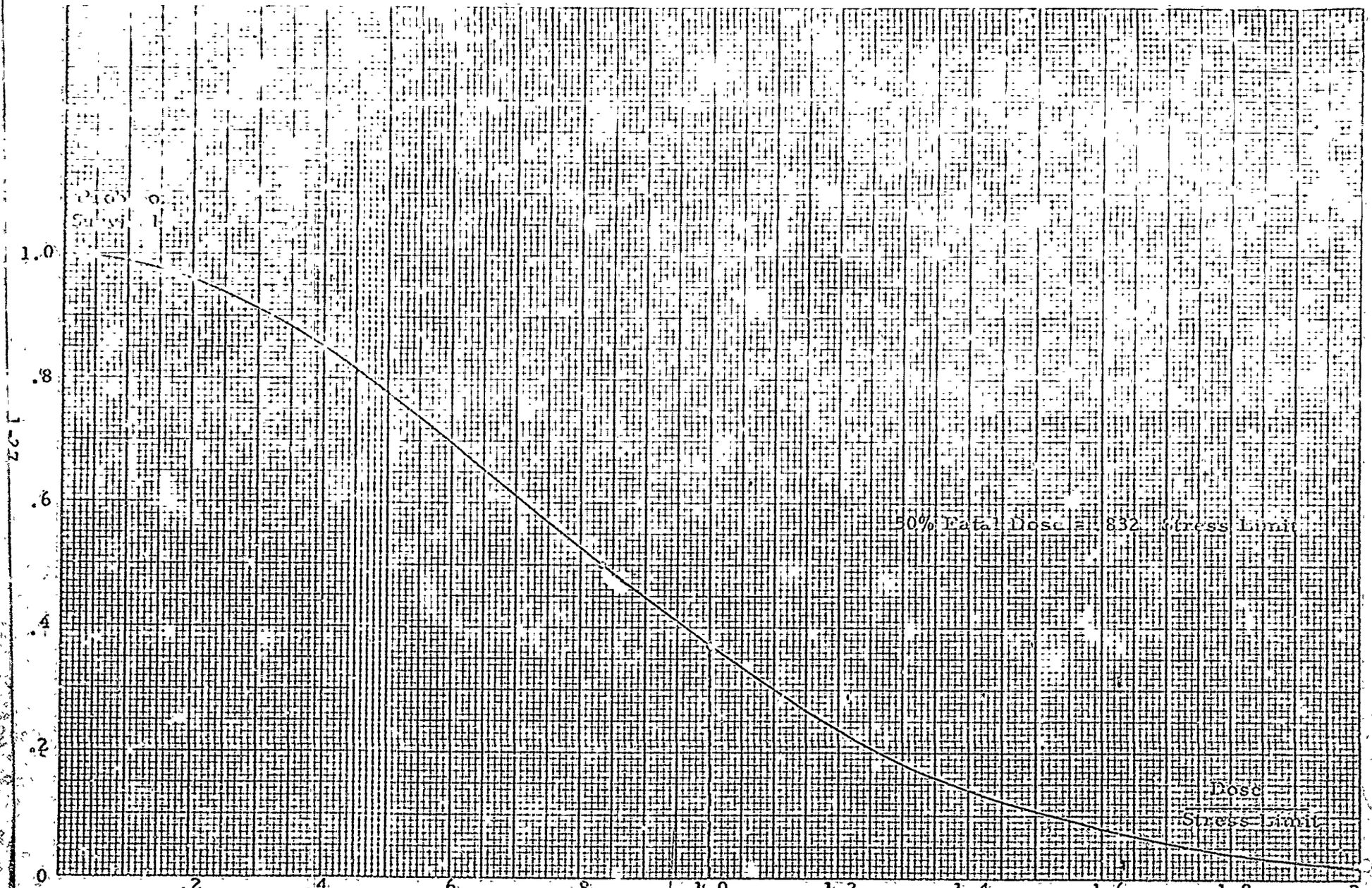


Figure 5. Fatality Curve for Blast or Thermal.

DRAWING PAPER NO. 1230-101
TRACING PAPER NO. 1227-101
CROSS SECTION-10X10 TO 1 INCH
AQUALEE
MADE IN USA

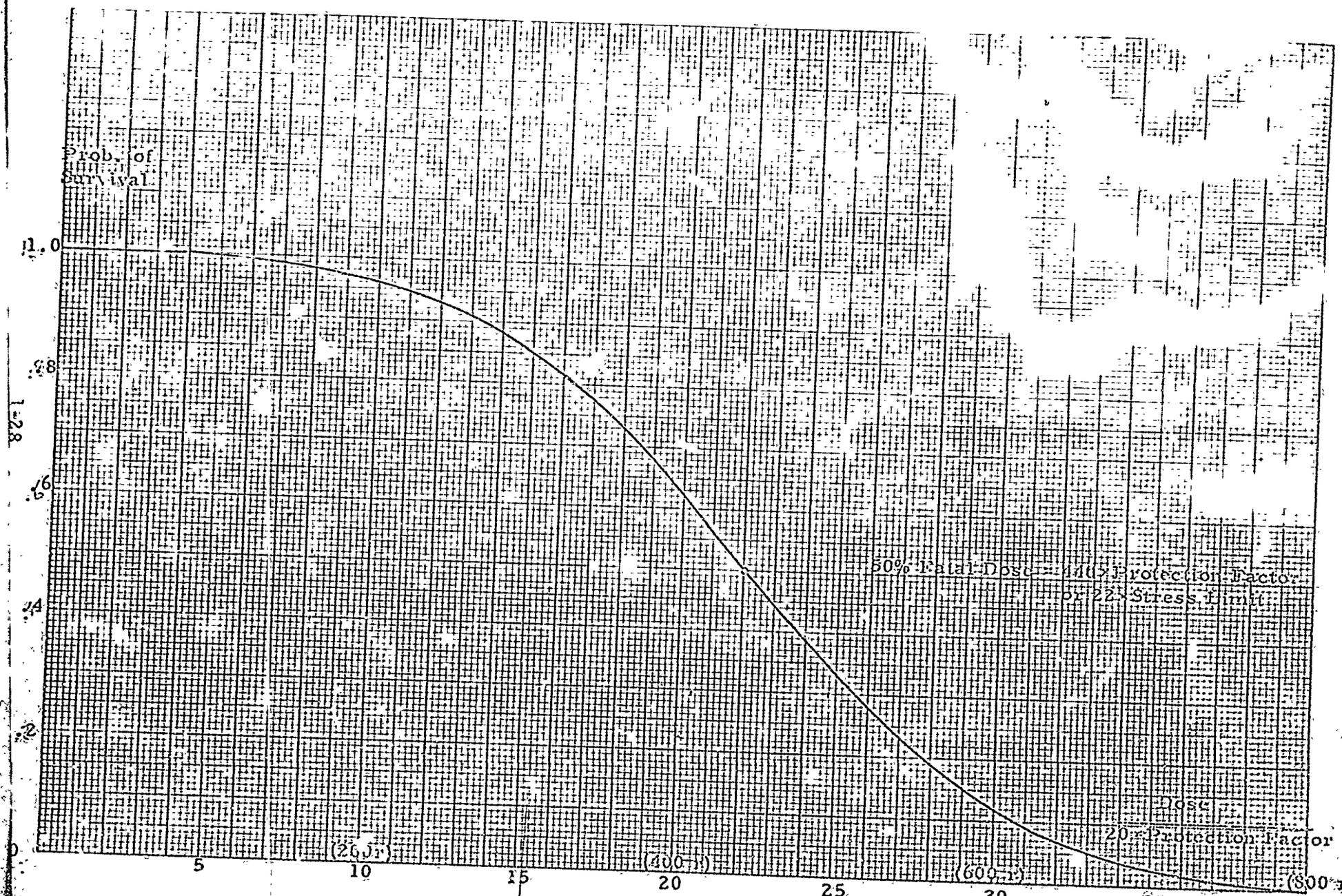


Figure 6. Fatality Curve for Radiation.

the blast alone. The fraction of the base killed by thermal is the count of points which survived the blast wave but did not survive the fire or thermal effects. The portion of kill attributed to radiation would be those which escaped the first two effects yet received a fatal roentgen dose in spite of the protection factor. To find what effectiveness fallout alone would have an assessment call could be made where the other effects are "turned off". The value of the base is assumed to be uniformly distributed over its area or equivalently its points and the value surviving that is printed out is that portion which survives all three effects. If, as sometimes occur, multiple independent assessments are desired, the option of leaving the base value in memory unchanged may be specified. Normally, the value decreases with succe. sive assessments. The total values surviving or killed by each of the effects is summed and printed out for each facility class at the end of its processing. A blank card ends each assessment call and returns control to the BEAM monitor.

The rural population does not occur as a facility in ATLAS. It or any other regular geographical array of values, such as rural MVA, may be assessed by directly reading in a density "map". Each 1° by 1° latitude-longitude square corresponds to a base or urban area in the normal procedure and has its own value. A specified square density of evaluation points are arranged symmetrically to cover

every cell (max 10 by 10) and the nation has a regular pattern of evaluation points except for the adjustment which is made for the cosine of the latitude. The value of a cell is represented as an alphanumeric character on a scale 0-50 with an arbitrary scale factor for the map as a whole. Each input card gives the latitude string of characters for a given latitude band, and the set of cards are in latitude order north to south. Blanks cells are not assessed. The output is represented in the same format as the input with maps for the survivors, blast kill, and fallout kill distribution. Cumulative nationwide totals are also printed out.

```
.RURPOP      -6.01017.200      49.00126.00    24                          5.0    999999.0
THIS IS INPUT MAP                                                                    10.0
      472111111111110311121312221111
    330(221431121111000111112215322341
    576642230122110100001223222244343233522                    14
    7(311219001211110000000001223489853536  6                 2153
    7931011200001011.10021011234580(=79+  633          65'94++4
    4311000811034311110012011335669877 7K 6=CC     + (97(8AE
   24420000031320001111111001135079777DL  .EL'    HG+=BE)V
    011000000014610000234111125(6598(=(.)HDAJ 8LD9(CAGP LH
    322100000701270000074321236455445=9=9(F.EDNHS,CBR VY
    629320000000022011133101122356'56789((DEMECCEBE.K-+
     D'3.1.  .001000311230011248488675G08=98A(A'648AHB
     (193000001600011221211224747(5635=('9=+8BH'96((
      7666011100000211100011226688433 7((8+8=BF=+G=='
       655222100011113211032346684445'+96(9+9EJ)BFG5
        =D722111111111100223144653359 7(7((+A'=FA+A+9
         776322211000022341236854566'697+'H'7('
          4 '222111522222432 36(5A.6787767(89658
                22  300111022744547678446968 66
                          000101254777CB=(55569645
                          0  02654+4  782     178
                                3234            7+
                                127             D88
                                 66              54
                                                 7
THIS IS MAP 3 OF FALOUT

    0300000000000000000000000000000000
   0011000000000000000000000000000000000
    000000000000000000000000000000000000000000000                00
    0100000000000000000000000000000000100000  0                 0000
    000000000000000000000000000000000000120000  000          00000000
    000000000000000000000000000000000000000000 0002    0 00000000
   000000000000000000000000000000000000000001  3136    31010226
    000000000000000000000000000000000000000007(725  6+2000104G57
    000000000000000000000000000000000000000000000000304+2047DQ
    000000000000000000000000000000000000000000254200000577
    0100000000000000000000000000000000000015000110010000061
    2+200000000000000000000000000000000000000000000000000
     000000000000000000000000000000000000000000000000000
      000000000000000000000000000000000000000000000000000
       022100000000000000000000000000000000000000000000
        1220000000000000000000000000000000000010010100
         0 000000000000000000000011000000000000000
                00  000000000000000000000000000
                     000000000000000000000000000
                     0  0000000  000      000
                          0000              00
                          000               010
                           00                00          VALUE SURVIVING   75003.
                                              0

KILLED BY BLAST   2381.4 , THERM    .  6.0 , RAD   5416.4
```

SHELL

This program is concerned with providing data on the blast and fallout protection the population might have in various circumstances. The output can be made available to ASSESS which enables the latter to perform a more sophisticated evaluation than it would otherwise. Three distinct functions can be performed using SHELL.

A basic set of data is required to describe the physical number and nature of the shelter spaces that exist at each facility site. The natural facility is of course the urban areas and this is the one for which a data bank has been accumulated and used. There is no reason why similar data could not be applied to military or other facilities that contain special groups of people. The national shelter survey was the source for creating the urban shell deck. Since the origin form of the data was not accumulated in a manner to be applicable to urban areas, selected component data was punched on cards for various parts of the urban complex. This consists of the number of spaces in the standard PF and PV categories defined by OCD. The first function and use of SHELL is to tabulate and order the totals to match the site list in the facility record, in this case

the 213 U.S. urban areas arranged by rank (identical to the 1966 Census Bureau ranking). This is one instance where order in the facility record is important. The tabulation is usually made once since survey data generally do not change often and a permanent shell tape written. Thereafter, the master tape is used to generate the operating tapes for SHELL and ASSESS. Both assume that the urban deck matches in number and order of places.

The second function that SHELL may do is to construct and add additional shelter spaces as directed. The new spaces may be constructed at a specified PF or PV category or both. The number added to any given place will depend on if and how much deficient that place falls below a requested criterion. The implication in terms of the national total spaces constructed is printed out.

The third function is to distribute at each place its population value in some manner into the various shelter spaces it has available. One choice, which might be suitable when long strategic warning occurs, would be to take maximum protection. In this case the hardest spaces are filled until either there are no more people or no more spaces. An opposite or no warning case would leave a percentage of the population with minimum protection. The remainder fill up the spaces in a pessimistic fashion, first next poorest and on

up perhaps leaving the best spaces empty. The most realistic option assumes that all shelters regardless of type are filled to a certain percentage of capacity, the percentage utilization specified. Any remaining population has minimum protection.

Any number or mixture of utilization can be stacked on a temporary or permanent tape for concurrent or later ASSESS requests.

SURVEY

The SURVEY program has been brought under the operating command of the BEACON monitor and is exercisable. It is currently more autonomous and thereby less in the battle stream than the other programs. While Task 41 study settles upon an MCI attrition model, it potentially may relate strongly to SURVEY and together integrate more firmly. At present SURVEY may be used to explore questions concerning missile and satellite trajectories, their visibility by one or a group of sensors, and potential warning or intercept aspects.

A number of objects may be launched from arbitrary places and directions around the globe with simple or elaborate descriptions of the boost stages and thrust forces. The gravitation attraction forces by the oblate earth are accounted for and the drag forces change with altitude and cross section in a density-temperature varying atmosphere. The spatial and rotating earth trajectories are integrated from the force equations. A trade-off between speed (of computation) and accuracy may be made.

A group of sensors may be deployed any where. These sensors which contain the trajectory in their horizon are indicated as a function of time as well as the object position in sensor coordinates.

COMAND

The COMAND program is a utility aid in providing visual plots of various geographical relationships between either sets in the facility records or in the warhead records or both simultaneously. A separate map is produced with each call. With a 1° cell resolution the map frame can be as large as 50° in latitude by 120° in longitude. Larger or smaller areas of the globe may be framed with a corresponding loss or gain in resolution. The frame may be positioned anywhere. The U.S. may be nicely framed by a $25^{\circ} \times 60^{\circ}$ map with $1/2^{\circ}$ cell sizes.

By naming a record, facility or warhead list, a facility set or type, and supplying a symbol, the program will print that symbol in every cell containg such a site or impact. Several requests may be made for the same map when one wishes to view the correlation between them. Clutter however should be avoided as multiple entries to the same cell result in converting the symbols to asterisks.

UTILIZATION OF THE PROGRAMS

The BEACON Monitor is initially given program control. The first card of the data deck is a date card which supplies the date, a name or number, and eight words for identification of the run. This information is printed out at the very beginning as part of the heading for the run. The monitor is able to read cards in any format; it passes them by until it comes to a BEAM card, which is identified by the word "BEAM* in the first six columns of the card (the quote sign " signifying a blank character). The next word (col 7-12) is a subroutine name and must be spelled precisely as one of the available routines ATLAS", STRIKE, DEFEND, ASSESS, SHELL", MOTION, SURVEY, COMAND or be completely blank. Any other combination of characters, such as END, acts as an end of run signal to the monitor. After the monitor has processed the remaining information on the BEAM card, control is then passed to the appropriate routine requested. As each routine completes the work for which it was called, control is returned to the monitor which resumes scanning the data deck for more BEAM* cards. It is in this manner that the user achieves his participation in the battle. By dividing the problem into appropriate segments, arranging their order and inserting BEAM* cards as definition of this control, the user literally commands the computer to perform the detailed calculations for a battle which he outlines.

Examples of some hypothetical deck set ups may be helpful. Simple preliminary jobs are given first.

1) Form a raw shelter tape for permanent use from the survey data, no construction, no utilization.

Date card

BEAM* SHELL" keys { 2=2, 5=0, 6=0 }

[PF and PV decks]

Blank card

BEAM* END

2) Run a missile trajectory past BMEWS.

Date card

BEAM* SURVEY

[Sensor list]

[Booster stage data]

[Launch coord data]

Blank card

BEAM* END

3) Write the main part of an atlas on tape.

Date card

BEAM* ATLAS Key 2=2 (write tape)

[Urban areas]

[SAC bases]

[Nike X sites]

[Fighter bases]

[Sage sites]

[AWACS Barrier]

[NORAD Mesh]

. . .

Blank card

BEAM* END

4) Read previous atlas from tape, add a set of launch sites, target and put aimpoints on tape, find dwell times for fast and slow bombers from AWACS barrier to target per NORAD sectors.

Date card

BEAM* ATLAS Key 2=1 (read tape)

[Launch sites]

Blank card

BEAM* STRIKE Key 2 = 2

[Warhead characteristics]

[Target requests]

. . .

[estimated PABORT, CEP]

2 blank cards (aimpoints only, no actual aborts)

BEAM* MOTION

NORAD3			(sector tag aircraft defense)
	(fast type)	(low type)	
NORAD2	LAUNCH	LAUNCH	(sector tag warhead
NORADØ	AWACS	AWACS	(find crossings and times)

Blank card

BEAM* END

5) Run three stage defense battle and assessments.

Date card

BEAM* ATLAS — Key 2 = 1 (read atlas tape)

Blank card

BEAM* STRIKE — Key 2 = 1 (read warhead tape)

PABT and CEP — Monte Carlo impacts

BEAM* DEFEND

[Missile attack only]

Nike X - parameters

Blank card

BEAM* ASSESS

[Missile attack only yields]

SAGE - hardness data

BOMARC - hardness data

Blank card

BEAM* DEFEND

(fast bomber attack)

SAGE

BOMARC

Blank card

BEAM* ASSESS

(SAGE and BOMARC for fast bomber attack)

BEAM* DEFEND

(SAGE and BOMARC and HERCULES for slow bomber attack)

BEAM* ASSESS

all warheads

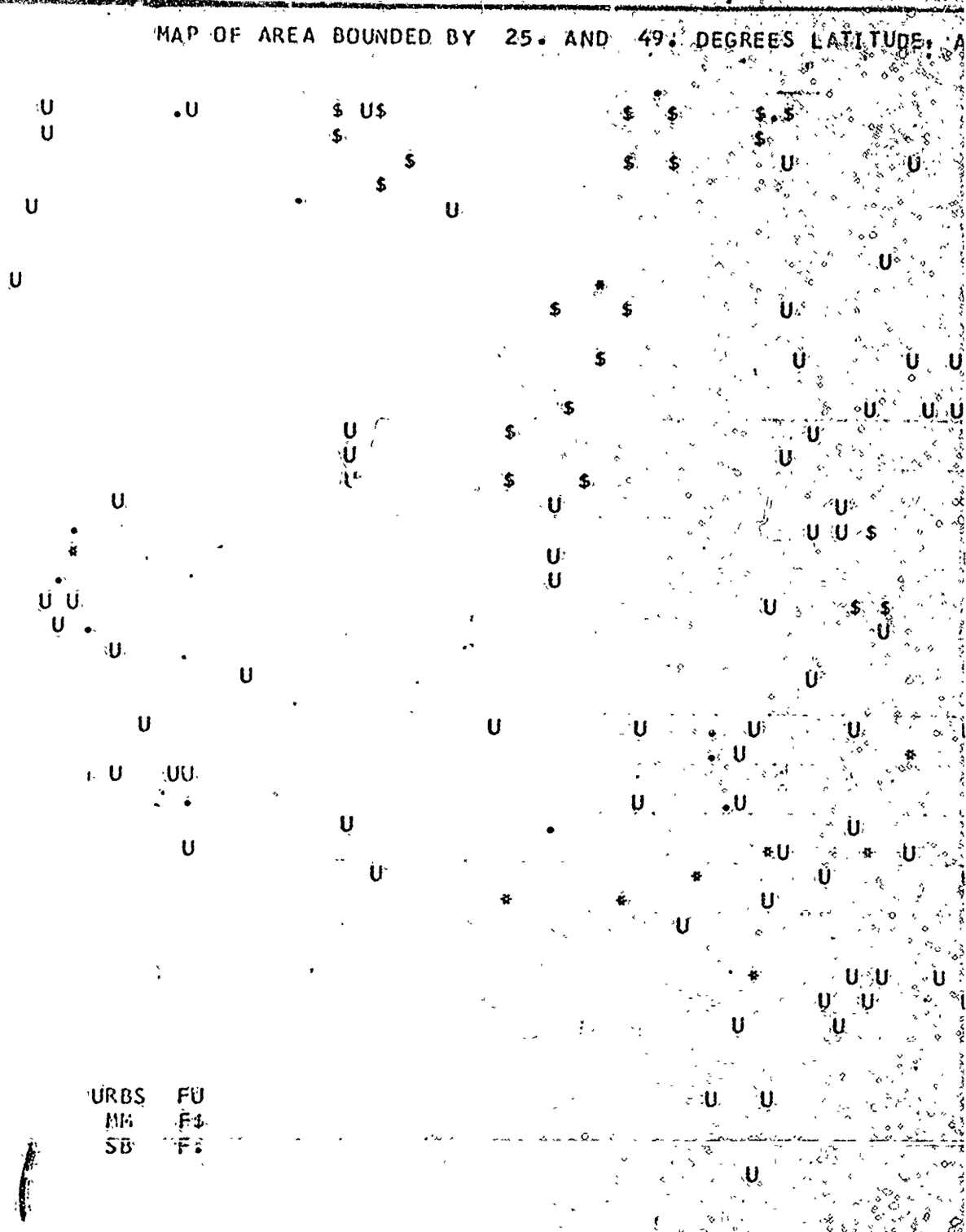
Urban areas

Rural Map

SAC bases

Blank card

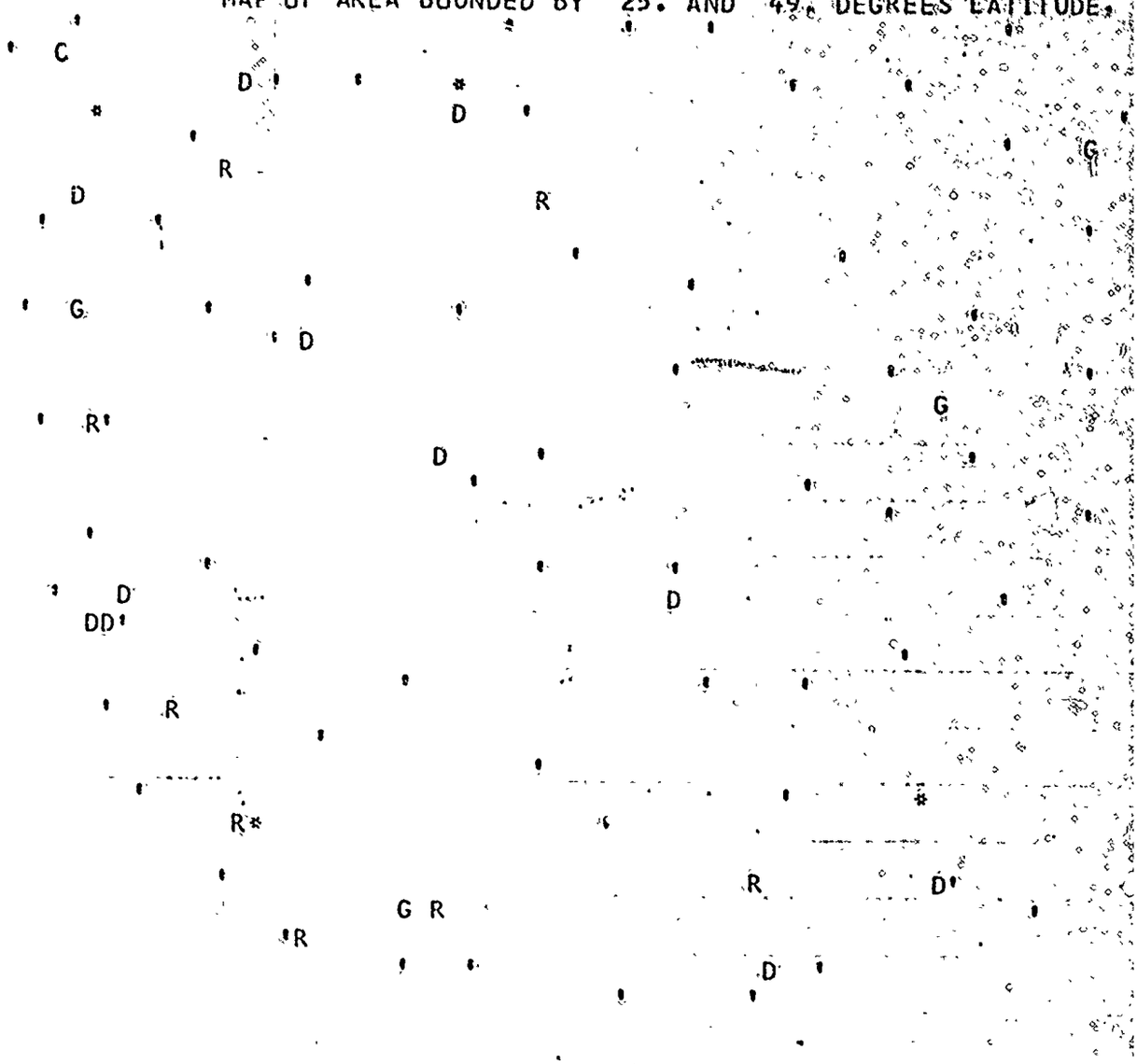
BEAM* END



Sample of map output showing typical ports of facility sets.

BY 25. AND 49. DEGREES LATITUDE, AND 67. AND 126. DEGREES LONGITUDE.

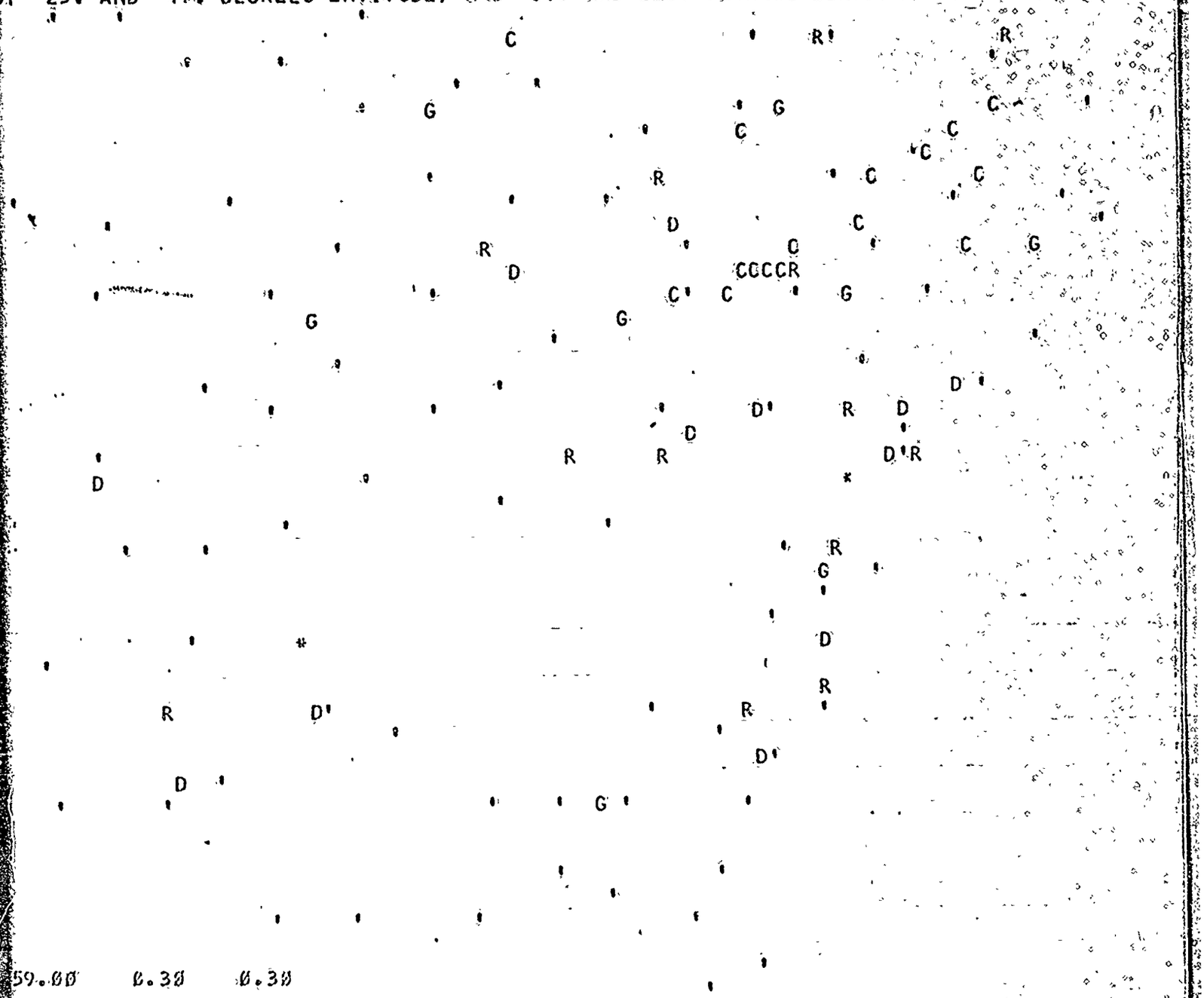
MAP OF AREA BOUNDED BY 25. AND 49. DEGREES LATITUDE.



THIS SECTION IS COMAND

MAP	US	49.00	126.00	24.00	59.00	0.30	0.30
RD	F						
SG	FG						
DB	FD						
RB	FR						
CANURB	FC						

BY 25. AND 49. DEGREES LATITUDE, AND 67. AND 126. DEGREES LONGITUDE.



SECTION II

BEACON MAIN ROUTINES for the IBM 7094

1. BEAM
2. ATLAS
3. STRIKE
4. MOTION
5. DEFEND
6. ASSESS
7. SHELL
8. SURVEY
9. COMAND

```
$IBFTC BEA7DK  DECK
CARDS FOR DEACO  MONITOR PROGRAM
      COMMON /MON/  IDENT(8),KEY(12),LBL(4),NUM
      COMMON  PAR/ LA,LB,RADIUS,PIE,RANDA,RANDB,
     + CONV,SCALAT,SCALON,POLAT,POLON,
     + ROUND,EPSI,DILATE,CPK1,CPK2,CPK3
      DIMENSION NEXT(9),TAPUN(4)
      DATA ICHECK/6H BEAM*/,KEYHOL/6H100000/,
     + TAPUN(1)/24H01..A3++++++03..A404..B4/,
     +NEXT(1)/54HATLAS STRIKEDEFENDASSESSSHELL MOTIONSURVEYCOMAND
      NAMELIST /NUPAR/ RANDA,RANDB,LA,LB,
     + CONV,SCALAT,SCALON,POLAT,POLON,
     + ROUND,EPSI,DILATE,CPK1,CPK2,CPK3,
COMMENCE
      READ  (LA,410) DATE, JOB, IDENT
      WRITE (LB,400) IDENT, DATE, JOB
      DO 107 L = 1,4
      IF( L.NE.2 )  REWIND L
  107    LBL(L) = 0
  101 READ  (LA,410) MATE, NAME, IDENT, KEY
      IF(ICHECK.NE.MATE) GO TO 101
      DO 102 J = 1,9
      IF( NAME.EQ.NEXT(J) ) GO TO 103
  102 CONTINUE
      STOP
  103 DO 104 K = 1,12
         KEY(K) = KEY(K)/KEYHOL
      IF( KEY(K).LT.0 )  KEY(K) = 0
  104 CONTINUE
         NUM = 1000*KEY(9) + 100*KEY(10) + 10*KEY(11) + KEY(12)
      IF( KEY(2).LT.7 )  GO TO 105
      READ  (LA,NUPAR)
      WRITE (LB,NUPAR)
         KEY(2) = KEY(2) - 7
  105    LAPSE = 0
  106 WRITE (LB,401) NAME, IDENT, KEY
      DO 90 L = 1,4
      IF( L.EQ.2 )  GO TO 90
         K = KEY(L)
      IF( K.EQ.0 )  GO TO 90
      GO TO (91,92,93,94,95,96,97,98,99),K
   91 PRINT 41, IDENT(L),TAPUN(L),IDENT(2)
         LBL(L) = 0
         LAPSE = L
      GO TO 90
   92 READ (L) LBL(L)
      GO TO 90
   93 BACKSPACE L
         LBL(L) = MAX0(LBL(L)-1,0)
      GO TO 90
   95 REWIND L
         LBL(L) = 0
   94    N = NUM
      IF( N.EQ.0 )  N = LBL(2)
      IF( KEY(8).EQ.1 )  N = -N
      IF( KEY(8).EQ.2 )  N = N - LBL(L)
      IF(N) 84,90,82
   82 DO 83 I = 1,N
   83 READ  (L)  LBL(L)
      GO TO 90
```

```
 84     N = MINO(-N,LBL(L))
        DO 85 I = 1,N
 85 BACKSPACE L
        LBL(L) = LBL(L)-N
        GO TO 90
 96 END FILE L
        GO TO 90
 98     LAPSE = -1
 99 PRINT 49, TAPUN(L), IDENT(L)
 97 REWIND L
        LBL(L) = 0
 90 CONTINUE
        IF( LAPSE.EQ.0 )  GO TO 108
        PRINT 45
        PAUSE
        IF( LAPSE.GT.0 )  REWIND LAPSE
108 GO TO (1,2,3,4,5,6,7,8,101),J
  1 CALL ATLAS
        GO TO 109
  2 CALL STRIKE
        GO TO 109
  3 CALL DEFEND
        GO TO 109
  4 CALL ASSESS
        GO TO 109
  5 CALL SHELL
        GO TO 109
  6 CALL MOTION
        GO TO 109
  7 CALL SURVEY
        GO TO 109
  8 CALL COMAND
109 WRITE (LB,409) NAME, DATE, JOB, LBL
        GO TO 101
410 FORMAT(10A6,12A1)
400 FORMAT(1H1//////////
     +          14X 39H HEADQUARTERS AIR FORCE SYSTEMS COMMAND
     +          15X 37H ACADEMY FOR INTERSCIENCE METHODOLOGY
     +//        22X 23H ANDREWS AIR FORCE BASE
     +          26X 31H MUSEUM OF SCIENCE AND INDUSTRY
     +//        22X 23H WASHINGTON, D.C. 20331
     +          30X 25H CHICAGO ILLINOIS 60637
     +////// 52X 15H BEACON PROGRAM
     +//  32X 55H BATTLE EFFECTIVENESS ASSESSMENT OF CONTINENTAL DEFENSE
     +////// 36X 8A6 // 54X 2A6 /1H1/)
401 FORMAT(17H THIS SECTION IS ,A6,3X,8A6,10H WITH KEYS,3(4X,4I2)//)
409 FORMAT(8HOEND OF ,A6,2A6,30X,19H, RECORD POSITIONS ,4I6/1H1/)
 41 FORMAT(15H MOUNT REEL NO,A6,11H , ON UNIT A6,13H , WITH RING A6)
 45 FORMAT(23H PRESS START WHEN READY)
 49 FORMAT(19H SAVE TAPE ON UNIT A6,12H WITH LABEL A6)
        END
$IBFTC BLK3DK  DECK
        BLOCK DATA
        COMMON /PAR/ LA,LB,RADIUS,PIE,RANDA,RANDB,
     + CONV,SCALAT,SCALON,POLAT,POLON,
     + ROUND,EPSI,DILATE,CPK1,CPK2,CPK3,
        DATA LA,LB,RADIUS,PIE,RANDA,RANDB /5,6,3437.7468,3.1415927,
     + .05746502,.74359130/,
     + CONV,SCALAT,SCALON,POLAT,POLON /.4,100.,-100.,280./,
     + ROUND,EPSI,DILATE,CPK1,CPK2,CPK3 /2*,5,341.,.5/,
        END
```

```
$IBFTC ATL70K  DECK
      SUBROUTINE ATLAS
      COMMON /MON/ IDENT(8),KEY(12),LBL(4),NUM
      COMMON /PAR/ LA,LB,RADIUS,DAT(3),CONV,SCALAT,SCALON,POLAT,POLON
      COMMON /FAC/ NF,NAMTYP(30),MFZ(30),VALT(30),RADT(30),
     + WLAT(1000),WLON(1000),VALU(1000),SIZE(1000),SECT(1000),QUAN(1000)
      DIMENSION FACILE(6121),INFMT(8),INOUT(8),Z(7),VAR(24)
      EQUIVALENCE (NF,FACILE)
      DATA NONE /6H      /, DEGREE /57.3  /
COMMENCE
      IF( MOD(KEY(2),2).EQ.0 )  GO TO 10
      READ  (1) LBL(1),FACILE
   10 IF( NUM.LT.100 )  NF = NUM
         I = 0
      IF( NF.NE.0 )  I = MFZ(NF)
   11 READ  (LA,41) NAME,NV,NS,INFMT,INOUT
      IF( NAME.EQ.NONE )  GO TO 20
      WRITE (LB,40)
      WRITE (LB,41) NAME,NV,NS,INFMT,INOUT
         NF = NF+1
         NAMTYP(NF) = NAME
      READ  (LA,42) VALT(NF),RADT(NF),Z
      WRITE (LB,43) VALT(NF),RADT(NF),Z
         NB = I+1
         JN = NV*NS
   12 READ  (LA,INFMT) (VAR(J),J=1,JN),KC
      IF( KC.NE.0 )  GO TO 15
      IF( KEY(6).EQ.0 )  WRITE (LB,INFMT) (VAR(J),J=1,JN)
         J = 0
      DO 13 JS = 1,NS
         I = I+1
      DO 14 JI = 1,NV
         JO = INOUT(JI)
         J = J+1
   14    Z(JO) = VAR(J)
         WLAT(I) = Z(1)
         WLON(I) = Z(2)
         VALU(I) = Z(3)
         SIZE(I) = Z(4)
         SECT(I) = Z(5)
   13    QUAN(I) = Z(6)
      GO TO 12
   15    I = I+1-KC
         NE = I
         MFZ(NF) = I
         VALUE = 0.
      DO 16 I = NB,NE
         WLAT(I) = SCALAT*(WLAT(I)-CONV*AINT(WLAT(I))) + POLAT
         WLON(I) = SCALON*(WLON(I)-CONV*AINT(WLON(I))) + POLON
      IF( RADT(NF).LT.0. )  VALU(I) = VALU(I)*SIZE(I)
```

```
16      VALUF = VALUF + VALU(I)
      IF( VALT(NF).EQ.0.)  GO TO 18
      WRITE (LB,403)
        A = WLAT(NE)/RADIUS
        B = WLON(NE)/RADIUS
        THED = COS(A)
        CHED = COS(B)*THED
        SHED = SIN(B)*THED
        THED = SIN(A)
      DO 17 I = NB,NE
        A = WLAT(I)/RADIUS
        B = WLON(I)/RADIUS
        CLAT = COS(A)
        SLAT = SIN(A)
        CLON = COS(B)
        SLON = SIN(B)
        A = CLON*CLAT
        B = SLON*CLAT
        SIZE(I) = SHED*SLAT - THED*B
        SECT(I) = THED*A - CHED*SLAT
        QUAN(I) = CHED*B - SHED*A
      IF( VALT(NF).GT.0.)  GO TO 19
        THED = VALU(I)/DEGREE
        CHED = COS(THED)
        SHED = SIN(THED)
        THED = SHED*SLAT
        WLAT(I) = CHED*SLON - CLON*THED
        WLON(I) = -CHED*CLON - SLON*THED
        VALU(I) = SHED*CLAT
19      CHED = A
        SHED = B
        THED = SLAT
17 WRITE (LB,402) WLAT(I),WLON(I),VALU(I),SIZE(I),SECT(I),QUAN(I)
      GO TO 11
18      VALT(NF) = VALUE
      IF( KEY(6).EQ.2 )  GO TO 11
      IF( INOUT(8).EQ.0 )  WRITE (LB,400) (SECT(I),QUAN(I),WLAT(I),
     + WLON(I),VALU(I),SIZE(I),I=NB,NE)
      IF( INOUT(8).EQ.1 )  WRITE (LB,401) (WLAT(I),WLON(I),I=NB,NE)
      GO TO 11
20 IF( KEY(2).LT.2 )  RETURN
        LBL(1) = LBL(1)+1
      WRITE (1) LBL(1),FACILE
      RETURN
40 FORMAT(1H1)
41 FORMAT(1XA6,1X,2I2,8A6,8I1)
42 FORMAT(9F8.2)
43 FORMAT(/8X,2F12.2,8X,7F10.2///)
400 FORMAT(/52X 17H ATLAS PARAMETERS //(1XA3,A6,3F7.0,F7.1,1XA3,A6,
     + 3F7.0,F7.1,1XA3,A6,3F7.0,F7.1))
401 FORMAT(//26X 34H LOGIC COORDINATES FOR THESE SITES//(8X,10F7.0))
402 FORMAT(12X,3F1..4,12X,3F12.4)
403 FORMAT(/51X 18H VECTOR COMPONENTS //)
      END
```

```
$IBFTC STP7DK  DECK
      SUBROUTINE STRIKE
      COMMON /NON/  IDENT(3),XT,YT,RT,RW,BYPASS,KEY(12),LBL(4)
      COMMON /PAR/ LA,LB,RADIUS,PIE,DAT(7),ROUND,FPSI
      COMMON /FAC/ NF,NAMTYP(30),MFZ(30),VALT(30),RADT(30),
     + Y(1000),X(1000),VAL(1000),APEA(1000),SECT(1000),QUAN(1000)
      COMMON /WHD/  W,MW7(12),YWH (2000),XWHD(2000),PWK(2000),IPW(2000)
      DIMENSION VAPK(1000),PSP(1000),ANP(1000),ANT(1000),VA(1000)
      DIMENSION INDEX(12),NORDER(12),MWIT(12),VLG(15),KSIT(6),PWHDKL(6),
     + RELY(12,15          12,15),WRAD(12,15),CODE(13,15),MWTGT(12,15)
      DIMENSION            ,CEP(12),VALIN(15),NRT(15),WEAPON(8013)
      EQUIVALENCE          ,MWIT),(VALIN,REQ),(NW,WEAPON)
      EQUIVALENCE (XWHD,VAPK),(YWHD,PSP,ANP),(REQ,MWTGT),(PWK,ANT)
COMMENCE
      IF( MOD(KEY(2),2).EQ.0 )  GO TO 900
      READ  (3) LBL(3), WEAPON
      IF( KEY(7).EQ.0 )  RETURN
      GO TO 999
  900 READ  (LA,400) INDEX
      WRITE (LB,400) INDEX
      READ  (LA,400) NORDER
      WRITE (LB,400) NORDER
         NT = 0
  910 READ  (LA,403) NAME,KG,PABT
      WRITE (LB,403) NAME,KG,PABT
      DO 911 KF = 1,NF
      IF( NAME.EQ.NAMTYP(KF) )  GO TO 912
  911 CONTINUE
      GO TO 914
  912    NT = NT + 1
         NRT(NT) = KF
      DO 913 KW = 1,12
         RFQ(KW,NT) = 0.
  913    WRAD(KW,NT) = PABT(KW)
      READ  (LA,403) NAME,KG,(RELY(KW,NT),KW=1,12)
      WRITE (LB,403) NAME,KG,(RELY(KW,NT),KW=1,12)
      IF( KG.LE.0 )  GO TO 910
      READ  (LA,403) NAME,KG,(RFQ  (KW,NT),KW=1,12)
      WRITE (LB,403) NAME,KG,(REQ  (KW,NT),KW=1,12)
      GO TO 910
  914    SIZE = 512**3
      READ  (LA,401) CEP
      WRITE (LB,401) CEP
      DO 923 KT = 1,NT
         CODE(1,KT) = 1.
      DO 924 KW = 1,12
         VSUM = ABS(RELY(KW,KT))
      IF( VSUM.LT.1. )  REQ(KW,KT) = -1.
         BIG = 1.
      DO 925 J = 1,10
      IF(BIG.GT.VSUM) GO TO 924
  925    BIG = BIG * 2.
  924    CODE(KW+1,KT) = CODE(KW,KT) * BIG
      IF( CODE(13,KT).LE.SIZE )  GO TO 923
      WRITE(LB,820) KT
      STOP
```

```
923 CONTINUE
        NEND = MFZ(NF)
919 DO 920 IT = 1,NEND
        VA(IT) = VAL(IT)
920     ANI(IT) = 0.0
        NW = 0
300 DO 399 LW = 1,12
        KW = NORDER (LW)
    IF( KW.LE.0 )  GO TO 940
        NW = MAX0(NW,KW)
        S7 = 4.5323*CEP(KW)**2
        MANY = INDEX(KW)
    IF( MANY.EQ.0 )  GO TO 399
    DO 329 KT = 1, NT
        KF = NRT(KT)
        NBEG = 1
    IF( KF.GT.1 )  NBEG = MFZ(KF-1) + 1
        NEND = MFZ(KF)
        PRQ = REQ(KW,KT)
    IF (PRQ) 329,313,312
312     AN = 1+NEND-NBEG
        VALUE = ABS(VALT(KF)) * (1.+AINT(PRQ)-PRQ) / AN
        VLG (KT) = ALOG (VALUE)
    GO TO 314
313     VALUE = 0.0
314     AW = PIE * WRAD(KW,KT)**2
    DO 309 IT = NBEG,NEND
309     VAPK (IT) = 0.0
    IF(RELY(KW,KT).GE.0.) GO TO 316
960     KF = 0
    READ  (LA,402) KC,(KSIT(J),PWHDKL(J),J=1,6)
    WRITE (LB,402) KC,(KSIT(J),PWHDKL(J),J=1,6)
    DO 964 J = 1,6
    IF(KSIT(J)) 962,964,961
961     KE = NBEG-1+KSIT(J)
        VAPK(KE) = PWHDKL(J)
    GO TO 964
962     KB = KE+1
        = NBEG-1-KSIT(J)
    DO 963 K = KB,KE
963     VAPK(K) = PWHDKL(J)
964 CONTINUE
    IF(KC.EQ.0) GO TO960
        RELY(KW,KT) = -RELY(KW,KT)
316     PX = RELY(KW,KT)-AINT(RELY(KW,KT))
        PX = AMAX1(PX,1.-PABT(KW))
    IF( RADT(KF) ) 317,320,315
317     PK = PX - PX*EXP(-AW/SZ)
    GO TO 320
315     AT = PIE * RADT(KF)**2
        PK = PX * PKAY(AT,AW,SZ)
320 DO 328 IT = NBEG,NEND
    IF(VA(IT).GT.VALUE) GO TO 322
        VAPK(IT) = 0.
    GO TO 328
322 IF(RADT(KF).NE.0.) GO TO 326
321     AT = AREA(IT)
        PK = PX * PKAY(AT,AW,SZ)
326     PSP(IT) = 1.0-PK*(1.-VAPK(IT))
        VAPK(IT) = VA(IT)*(1.-PSP(IT))
```

```
      IF(PA  (KF).GE.0.) GO TO 328
         PSP(IT) = PSP(IT)**(1./AREA(IT))
         VAPK(IT) = VAPK(IT)/AREA(IT)
  328 CONTINUE
  329 CONTINUE
  330    VPMAX = 0.0
      DO 339 KT = 1,NT
         KF = NRT(KT)
         NBEG = 1
      IF( KF.GT.1 )  NBEG = MFZ(KF-1) + 1
         NEND = MFZ(KF)
      IF(REQ(KW,KT)) 339,339,334
  334 DO 338 IT = NBEG,NEND
      IF(VPMAX-VAPK(IT)) 336,338,338
  336    VPMAX = VAPK(IT)
         JT = IT
         LT = KT
  338 CONTINUE
  339 CONTINUE
      IF(VPMAX) 350,350,340
  340    VAPK(JT) = 0.0
         AKNP = ALOG(VA(JT))-VLG(LT)
  342    PS = PSP(JT)
         BN = AKNP/AMAX1(.001,-ALOG(PS))
      IF(KFY(5).GT.0) ROUND = RANDOM(ROUND)
         N = INT(BN+ROUND)
         SHIFT = CODE(KW,LT)
         LEAST = INT(REQ(KW,LT))
         N = MAX0(N,LEAST)
         MOST = INT(RELY(KW,LT))
      IF( N.LE.MOST )  GO TO 347
         KF = NRT(LT)
      WRITE (LB,835) NAMTYP(KF),QUAN(JT),N,KW
         N = MOST
  347    N = MIN0 (N,MANY)
         VA(JT) = VA(JT)*PS**N
         ANI(JT) = ANI(JT) + FLOAT (N)* SHIFT
         MANY = MANY-N
      IF(MANY) 399,399,330
  350    ZPRO = MANY
         ZANI = 0.0
         NP = 0
  351    VPMAX = 0.0
      DO 359 KT = 1,NT
         KF = NRT(KT)
         NBEG = 1
      IF( KF.GT.1 )  NBEG = MFZ(KF-1) + 1
         NEND = MFZ(KF)
      IF(REQ(KW,KT)) 359,354,353
  353    REQ(KW,KT) = -1.
      GO TO 359
  354 DO 358 IT = NBEG,NEND
      IF(VPMAX-VAPK(IT)) 356,358,358
  356    VPMAX = VAPK(IT)
         JT = IT
  358 CONTINUE
  359 CONTINUE
      IF(.PMAX) 370,370,360
  360    AKNP = -ALOG(VPMAX)
      IF(AKNP * ZANI -ZPRO) 361,371,371
```

```
361     NP = NP + 1
        VAPK(JT) = -AKNP-20.0
365     AN = 1.0/AMAX1(.001,-ALOG(PSP(JT)))
        ANP(JT) = AN
366     ZANI = ZANI + AN
        ZPRO = ZPRO + AN* AKNP-PPST
      IF (AKNP * ZANI -ZPRO) 361,372,372
370 IF (NP) 390,390,371
371     AKNP = ZPRO/ZANI
372     VPMAX = -40.0
      DO 379 KT = 1,NT                                                  ALT
        KF = NRT(KT)
        NBEG = 1
      IF( KF.GT.1 )  NBEG = MFZ(KF-1) + 1
        NEND = MFZ(KF)
      IF( EQ(KW,KT).NE.0.)  GO TO 369
      DO 379 IT = NBEG,NEND
      IF( VAPK(IT).GE.0. )  GO TO 379
      IF( VPMAX.GE.VAPK(IT) )  GO TO 379
        JT = IT
        LT = KT
        VPMAX = VAPK(IT)
379   CONTINUE
369 CONTINUE
      IF( VPMAX.EQ.-40.0 )  GO TO 380
        BN = (AKNP+VAPK(JT)+20.) * ANP(JT)
        VAPK(JT) = 0.0
        N = INT(BN+1.0)
        MOST = INT(RELY(KW,LT))
      IF( N.LE.MOST )  GO TO 366
        KF = NRT(LT)
      WRITE (LB,835) NAMTYP(KF),QUAN(JT),N,KW
        N = MOST
366     N = MIN0 (N,MANY)
        AN = N
        SHIFT = CODE(KW,LT)
        VA(JT) = VA(JT) * EXP(-AN/ANP(JT))
        ANI(JT) = ANI(JT) + AN*SHIFT
        MANY = MANY-N
      IF(MANY) 399,399,372
380 WRITE(LB,838)MANY,KW
      GO TO 350
390 WRITE (LB,839)MANY,KW
399 CONTINUE
940 WRITE (LB,830)
        VALUE = 0.
        VSUM = 0.0
      DO 941 KT = 1, NT
        KF = NRT(KT)
        NBEG = 1
      IF( KF.GT.1 )  NBEG = MFZ(KF-1) + 1
        NEND = MFZ(KF)
      WRITE (LB,842) NAMTYP(KF)
      WRITE (LB,840) (VA(IT),IT=NBEG,NEND)
943     VLG(KT) = 0.0
        VALIN(KT) = 0.
      DO 944 IT = NBEG,NEND
        VALIN(KT) = VALIN(KT) + VAL(IT)
944     VLG(KT) = VLG(KT) + VA(IT)
        VALUE = VALUE + VALIN(KT)
```

```
941     VSUM = [illegible] + VLG(KT)
        WRITE [illegible] (VLG([illegible]),KT=1,NT)
        WRITE [illegible] VALIN(KT),KT=1,NT)
        WRITE [illegible]
        [illegible] = 0
        DO [illegible] NW
        [illegible] + INDEX(KW)
501     [illegible](KW) = MANY - INDEX(KW)
        DO [illegible] = 1,12
502     [illegible] = 0
        DO [illegible] KT = 1,NT
        [illegible] = NAT(KT)
        [illegible]
        IF( [illegible] NBEG - MFZ(KF-1) [illegible]
        [illegible] = [illegible](KF)
        [illegible] = 0.
        DO [illegible] KW = 1,NW
[illegible] [illegible](KW,[illegible]) = 0
511 WRITE (LB,852) NAMTYP(KF)
        [illegible] = 0.0
        RT = RADT(KF)
512 DO [illegible] IT = NBEG,NEND
        M=0
        DO 520 LW = 1,NW
        KW = NW+1-LW
        SHIFT = CODE(KW,KT)
        [illegible](KW) = INT(ANI(IT)/SHIFT)
        [illegible] = ANI(IT)-FLOAT(MWIT(KW))*SHIFT
520     M = M+[illegible](KW)
530 IF([illegible]) 569,569,531
531 DO 532 KW = 1,NW
532     [illegible](KW,KT) = MWIT(KW,KT)+MWIT(KW)
        JT = IT - NBEG + 1
        [illegible] (LB,[illegible]) QUAN(IT),JT,(MWIT(KW),KW=1,NW)
        IF( [illegible] ) GO TO 537
        DEV = -RADT(KF)
        LY = SQRT(AREA(IT)) + .5
        CDEV = LY-1
        CDEV = .5*CDEV*DEV
        EX = COS(Y(IT)/RADIUS)
        XDEV = DEV/EX
        KW = 1
        IWB = INDEX(1)
        IWE = INDEX(1) + MWIT(1)
540     WITE = 1.0
        XT = X(IT) - CDEV/EX
549     YT = Y(IT) - CDEV
        DO 540 J = 1,LY
542 IF( IWB.LT.IWE )  GO TO 547
        IF( MWIT(KW).LE.0 )  GO TO 545
        IF(KEY(6).GT.0) GO TO 566
        IWB = INDEX(KW) + 1
        WRITE (LB,861)(YWHD(IW),XWHD(IW),IW=IWB,IWE)
[illegible]     INDEX(KW) = IWE
545 IF( KW.GE.NW )  GO TO 569
        KW = KW+1
        IWB = INDEX(KW)
        IWE = IWB + MWIT(KW)
        GO TO 542
547     IWB = IWB + 1
```

```
            XWHD(IWB) = XT
            YWHD(IWB) = YT
        IF(SITE.GE.AREA(IT)) GO TO 540
            SITE = SITE + 1.0
548         YT = YT+DEV
            XT = XT+XDEV
        GO TO 549
537         XT = X(IT)
            YT = Y(IT)
        IF(PADT(KF).EQ.0.)RT = SQRT(AREA(IT)/PIE)
534     DO 536 KW = 1,NW
        IF(MWIT(KW)) 536,536,535
535         RW= WRAD(KW,KT)
        CALL COORD(MWIT(KW),INDEX(KW))
536     CONTINUE
589     CONTINUE
        WRITE (LB,858)(MWTGT(KW,KT),KW=1,NW)
        DO 590 KW = 1,NW
590         MWZ(KW) = MWZ(KW) + MWTGT(KW,KT)
599     CONTINUE
        WRITE (LB,897)
        DO 997 KT = 1,NT
            KF = NRT(KT)
997     WRITE (LB,853) NAMTYP(KF),KF,(MWTGT(KW,KT),KW=1,NW)
        WRITE (LB,859) MWZ
            NEND = 0
        DO 993 KW = 1,NW
            NEND = NEND + MWZ(KW)
993         MWZ(KW) = NEND
998     READ  (LA,401) PABT
        WRITE (LB,401) PABT
        READ  (LA,401) CEP
        WRITE (LB,401) CEP
            I = 0
            NEND = 0
        DO 995 KW = 1,NW
            NBEG = NEND + 1
            NEND = MWZ(KW)
        IF(NEND.LT.NBEG) GO TO 995
        IF( KEY(7).EQ.0 )  I = INDEX(KW)-NEND
            SZ = 0.84933*CEP(KW)
            ZABT = 0.
        DO 994 J = NBEG,NEND
        CALL RANORM(DX,DY)
            JI = J+I
            EX = COS(YWHD(JI)/RADIUS)
            XWHD(J) = XWHD(JI) + SZ*DX/EX
            YWHD(J) = YWHD(JI) + SZ*DY
            PWK(J) = PABT(KW)
        IF(KEY(5).GE.2) PWK(J) = AINT(PWK(J)+RANDOM(ROUND))
994         ZABT = ZABT + PWK(J)
        IF(KEY(6).GT.1) GO TO 995
        WRITE (LB,890) KW,ZABT,(YWHD(J),XWHD(J),PWK(J),J=NBEG,NEND)
995     CONTINUE
        IF( KEY(2).LT.2 )  RETURN
            LBL(3) = LBL(3)+1
        WRITE (3) LBL(3), WEAPON
        RETURN
400     FORMAT(8H CARDLBL,12I6)
401     FORMAT(8H CARDLBL,12F6.2)
```

```
402 FORMAT(1H ,I1,6HCARD#,6(I6,F6.3))
403 FORMAT(1H ,A6,A1,12F6.2)
820 FORMAT(//32H CODING OVERFLOW ON TARGET TYPE I3)
835 FORMAT(2XA6,8H TARGET.,4XA6,10H REQUESTEDI4,8H WHDTYPEI3)
838 FORMAT(11HOTHERE WERE I6,3X16HWARHEADS OF TYPEI3,3X8HLEFT ..//)
839 FORMAT(11HOTHERE  ARE I6,3X16HWARHEADS OF TYPEI3,3X8HLEFT .  //)
830 FORMAT( 1H1 39X 40H THE EXPECTED VALUE OF TARGETS LEFT ARE /)
842 FORMAT(//2XA6,8H TARGETS /)
840 FORMAT(( X,10(F10.2)))
841 FORMAT(//5H  SUM F11.2,8H AFTER ,37X10H PER TYPE /(16X,10(F10.2)))
843 FORMAT(//5H  SUM F11.2,8H BEFORE,37X10H PER TYPE /(16X,10(F10.2)))
850 FORMAT(1H1 37X 44HTHE RESULTS OF THE TARGETING ARE AS FOLLOWS /)
852 FORMAT(//2XA6,8H TARGETS,42X25HNUMBER OF WEAPONS OF TYPE / 28X,
   +  1H17X1H27X1H37X1H47X1H57X1H67X1H77X1H87X1H96X2H106X2H116X2H12//)
853 FORMAT(2XA6,2XI4,7X12I8)
858 FORMAT(//22H TOTAL WEAPONS TO TYPE I7,11I8/)
897 FORMAT(1H135X50H NUMBER OF WARHEADS BY TARGET TYPE AND WEAPON TYPE
   +////21H TARGET / WEAPON TYPE
   +7X1H17X1H27X1H37X1H47X1H57X1H67X1H77X1H87X1H96X2H106X2H116X2H12//)
859 FORMAT(//24H THE GRAND TOTAL WEAPONS I5,11I8/1H1//)
861 FORMAT(5(6X,2F9.2))
890 FORMAT(/41X 38HWARHEAD AIM POINTS FOR WEAPONS OF TYPE I3,/
   + 53X F6.1, 8H ABORTED  // (4(6X,2F9.2,F6.2)))
    END
```

```
$IBFTC MOVEDK  DECK
      SUBROUTINE MOTION
      COMMON /PAR/ LA,LB,RADIUS,PIE
      COMMON /FAC/ NF,NAMTYP(30),MFZ(30),VALT(30),RADT(30),
     + Y(1000),X(1000),VAL(1000),A(1000),B(1000),C(1000)
      COMMON /WHD/ NW,MWZ(12),YW(2000),XW(2000),PWK(2000),IPW(2000)
      COMMON /MAD/ LPZ(12)
      DIMENSION NRP(12),DSLIM(12),MANY(30),DWELT(30),L(1000)
      EQUIVALENCE (L,B)
COMMENCE
    1 READ  (LA,41) NAMESH,KC,NRP
      DO 2 KM = 1,NF
      IF( NAMTYP(KM).EQ.NAMESH )  GO TO 3
    2 CONTINUE
      RETURN
    3 WRITE (LB,41) NAMESH,KC,NRP
      IF( KC.LT.3 )  GO TO 9
CLASSIFY FACILITIES BY SECTOR
    4 DO 5 JW = 1,12
      DO 6 KF= 1,NF
      IF( NAMTYP(KF).EQ.NRP(JW) )  GO TO 7
    6 CONTINUE
      GO TO 5
    7    IFB = 1
      IF(KF.GT.1) IFB = MFZ(KF-1)+1
         IFE = MFZ(KF)
      DO 31 I = 1,30
   31    MANY(I) = 0
      DO 30 I = IFB,IFE
         L(I) = LOCATE(Y(I),X(I),KM)
         IL = L(I)
   30    MANY(IL) = MANY(IL) + 1
      WRITE (LB,404) NAMTYP(KF),NAMESH,MANY
      WRITE (LB,408) (C(I),L(I),I=IFB,IFE)
    5 CONTINUE
      GO TO 1
    9 IF( KC.EQ.2 )  GO TO 8
      READ  (LA,42) DSLIM
      WRITE (LB,42) DSLIM
    8    IWE = 0
      DO 10 KW = 1,NW
         IWB = IWE + 1
         IWE = MWZ(KW)
      IF( IWE.LT.IWB )  GO TO 10
      DO 71 KP = 1,NF
      IF( NAMTYP(KP).EQ.NRP(KW) )  GO TO 72
```

```
   71 CONTINUE
      GO TO 10
   72    IPB = 1
      IF( KP.GT.1 )  IPB = MFZ(KP-1) + 1
         IPE = MFZ(KP)
      DO 18 IP = 1,30
         DWELT(IP) = 0.
   18    MANY(IP) = 0
      IF( KC.EQ.0 )  GO TO 16
COMPUTE LAUNCH POSITIONS
   17 DO 11 IW = IWB,IWE
      IF( KC.EQ.2 ) GO TO 13
      IF( DSLIM(KW).EQ.0.)  DSLIM(KW) = 10800.
         DSMIN = 10800.
      DO 12 IP = IPB,IPE
         DS = DIST(YW(IW),XW(IW),Y(IP),X(IP))
      IF( DS.GE.DSMIN )  GO TO 12
         DSMIN = DS
         IPW(IW) = IP - IPB + 1
   12 CONTINUE
      IF( DSMIN.GT.DSLIM(KW) )  PWK(IW) = PWK(IW) + 10.
      GO TO 15
   13    IPW(IW) = LOCATE(YW(IW),XW(IW),KM)
   15 IF( PWK(IW).GT.0.99 )  IPW(IW) = 0
         IP = IPW(IW) + 1
   11    MANY(IP) = MANY(IP) + 1
         LPZ(KW) = IPB - 1
      WRITE (LB,401) NAMTYP(KP),KW,MANY
      WRITE (LB,409) IPB,(IPW(IW),IW=IWB,IWE)
      GO TO 10
CROSSINGS OF BARRIER
   16 DO 21 IARC = IPB,IPE
   21    VAL(IARC) = 0.
      DO 25 IW = IWB,IWE
      IF( PWK(IW).GT.0.99 )  GO TO 25
         JW = IPW(IW) + LPZ(KW)
         X1 = COS(X(JW)/RADIUS)*COS(Y(JW)/RADIUS)
         Y1 = SIN(X(JW)/RADIUS)*COS(Y(JW)/RADIUS)
         Z1 = SIN(Y(JW)/RADIUS)
         X2 = COS(XW(IW)/RADIUS)*COS(YW(IW)/RADIUS)
         Y2 = SIN(XW(IW)/RADIUS)*COS(YW(IW)/RADIUS)
         Z2 = SIN(YW(IW)/RADIUS)
         AA = Y1*Z2-Y2*Z1
         BB = Z1*X2-Z2*X1
         CC = X1*Y2-X2*Y1
      JARC=IPE
      AMILES=DIST(Y(JW),X(JW),YW(IW),XW(IW))
      DO20IARC=IPB,IPE
      DMILES=DIST(Y(JARC),X(JARC),Y(IARC),X(IARC))
         D1 = BB*C(IARC)-CC*B(IARC)
         D2 = CC*A(IARC)-AA*C(IARC)
         D3 = AA*B(IARC)-BB*A(IARC)
         D0 = SQRT(D1*D1+D2*D2)
         ALAT = RADIUS*ATAN(D3/SQRT(D1*D1+D2*D2))
         ALONG= 2.*ATAN(D2/(D0+D1))*RADIUS
```

```
      DO 304 J = 1,2
      TEMP=DIST(ALAT,ALONG,Y(IARC),X(IARC))
      TEMP1=DIST(ALAT,ALONG,Y(JARC),X(JARC))
      IF(TEMP+TEMP1.GT.1.+BMILES)GOTO303
      TEMP=DIST(ALAT,ALONG,Y(JW),X(JW))
      TEMP1=DIST(ALAT,ALONG,YW(IW),XW(IW))
      IF(TEMP+TEMP1.GT.1.+AMILES)GOTO303
      CROSS=TEMP1/DSLTM(KW)
      XLAT=AINT(ALAT/60.)+(.01)*AMOD(ALAT,60.)
      XLONG=AINT(ALONG/60.)+(.01)*AMOD(ALONG,60.)
      IWW=IW-IWB+1
      IAR=IARC-IPB+1
      WRITE(LB,402)KW,IWW,IAR,CROSS,XLAT,XLONG
      MANY(IAR)=MANY(IAR)+1
         DWELT(IAR) = DWELT(IAR) + CROSS*(1.-PWK(IW))
         VAL(IARC) = VAL(IARC) + 1. - PWK(IW)
      GO TO 25
  303 ALAT=-ALAT
      ALONG=-SIGN(10800.-ABS(ALONG),ALONG)
  304 CONTINUE
      JARC=IARC
   20 CONTINUE
   25 CONTINUE
      IPBE=IPE-IPB+1
      DO 22 IAR = 1,IPBE
         IARC = IAR + IPB - 1
   22    DWELT(IAR) = DWELT(IAR)/VAL(IARC)
      WRITE(LB,403)NAMTYP(KP),KW,DWELT
      WRITE (LB,409) IPBE,MANY
      WRITE (LB,407) (VAL(IARC),IARC=IPB,IPE)
   10 CONTINUE
      GO TO 1
   41 FORMAT(1H ,A6,I1 ,12A6)
   42 FORMAT(8X,12F6.0)
  404 FORMAT(11HONUMBER OF A6,9H IN MESH A6 /(3(10X5I6)))
  401 FORMAT(17HOLAUNCHS FROM SETA6,12H,WEAPON TYPEI4,75X16/(3(10X5I6)))
  402 FORMAT(8H WHDTYPEI3,5H PEN I3,12H CROSSED ARCI3,10H AT MINUS F6.1,
     +10H MINS, AT 2F8.2)
  403 FORMAT(25HODWELL TIMES FOR BARRIER A6,17H WEAPONS OF TYPE I4/
     +(2(10X,5F10.1)))
  407 FORMAT(/(3(10X5F6.1)))
  408 FORMAT(/(10(2XA6,I4)))
  409 FORMAT(114XI6/(3(10X5I6)))
      END
```

```
      SUBROUTINE DEFEND
      COMMON /MON/  IDENT(8),KEY(12),LBL(4)
      COMMON /PAR/ LA,LB,RADIUS,
      COMMON /FAC/ NF,NAMTYP(30),MFZ(30),VALT(30),RADT(30),
     + Y(1000),X(1000),F(1000),AREA(1000),SECT(1000),QUAN(1000)
      COMMON /WHD/ NW,MWZ(12),YW(2000),XW(2000),PWK(2000),IPW(2000)
      DIMENSION RENTRY(12),AZ(12),FBOGY(12),DWELT(12),PDET(12),PKQH(12),
     + COVER(12),DRIFTX(12),DRIFTY(12),WIND(12),ZWIND(12),WK(12),ZWK(12)
     +,WEAPON(8013),LSEC(1000)
      EQUIVALENCE (NW,WEAPON),(LSEC,SECT)
COMMENCE
      READ  (LA,43) RAIDT
      WRITE (LB,43) RAIDT
      READ  (LA,42) RENTRY
      WRITE (LB,42) RENTRY
      READ  (LA,42) AZ
      WRITE (LB,42) AZ
      READ  (LA,42) FBOGY
      WRITE (LB,42) FBOGY
      DO  5 KW = 1,NW
         DRIFT = 1./TAN(RENTRY(KW)/57.3)
         DRIFTX(KW) =-DRIFT*SIN(AZ(KW)/57.3)
    5    DRIFTY(KW) =-DRIFT*COS(AZ(KW)/57.3)
   10 READ  (LA,41)NAME,S,DEGRAD,SATUR,CHANS,AVAIL,TRKT,HLIM,DISCRM,HIGH
      DO 11 KD = 1,NF
   11 IF( NAME.EQ.NAMTYP(KD) )  GO TO 12
      GO TO 90
   12 WRITE (LB,41)NAME,S,DEGRAD,SATUR,CHANS,AVAIL,TRKT,HLIM,DISCRM,HIGH
      READ  (LA,42) COVER
      WRITE (LB,42) COVER
      READ  (LA,43) PKQH
      WRITE (LB,43) PKQH
      READ  (LA,43) PDET
      WRITE (LB,43) PDET
      READ  (LA,43) DWELT
      WRITE (LB,43) DWELT
      DO 15 KW = 1,12
         COVER(KW) = COVER(KW)**2
         WK(KW) = 0.
         WIND(KW) = 0.
         ZWIND(KW) = 0.
   15    ZWK(KW) = 0.
          IDB = 1
      IF(KD.GT.1) IDB = MFZ(KD-1) + 1
         IDE = MFZ(KD)
      DO 20 ID = IDB,IDE
      IF(F(ID).LE.0.5) GO TO 20
         WVIS = 0.
         BOGIES = 0.
         IWE = 0
         EX = COS(Y(ID)/RADIUS)
         AVDWLT = 0.
      DO 30 KW = 1,NW
         WIND(KW) = 0.
         IWB = IWE+1
```

```
      IWE = MWZ(KW)
   IF(IWE.LT.IWB) GO TO 30
   IF(COVER(KW).LE.0.) GO TO 30
      XD = X(ID) + DRIFTX(KW)/EX*HIGH
      YD = Y(ID) + DRIFTY(KW)*HIGH
   DO 50 IW = IWB,IWE
   IF(PWK(IW).GE.1.) GO TO 50
      RANGE = ((XW(IW)-XD)*EX)**2 + (YW(IW)-YD)**2
   IF(RANGE.GT.COVER(KW)) GO TO 50
   IF( S.LE.0.)  GO TO 51
   IF( IPW(IW).NE.LSEC(ID) )  GO TO 50
51    WIND(KW) = WIND(KW) + 1. - PWK(IW)
      PWK(IW) = PWK(IW) - 1.
50 CONTINUE
   IF( WIND(KW).EQ.0.)  GO TO 30
      ZWIND(KW) = ZWIND(KW) + WIND(KW)
      PCPRES = DWELT(KW)/RAIDT
      WPRES = WIND(KW)*PCPRES
      WOBS = WPRES*PDET(KW)
      WVIS = WVIS + WOBS
      AVDWLT = AVDWLT + WOBS*DWELT(KW)
      BOGIES = BOGIES + WOBS + FBOGY(KW)*(1.-DISCRM)*WOBS
30 CONTINUE
      HRQT = 0.
   IF(BOGIES.EQ.0.) GO TO 25
      PTQD = EXP(-0.5*(BOGIES/SATUR)**2)
      BT = BOGIES*PTQD
      AVDWLT = AVDWLT/WVIS
      HRQT = HLIM*TRKT/AVDWLT
      ZGCI = AMIN1(CHANS,BT*HRQT)*RAIDT/TRKT
      ZFH = ABS(AVAIL)*F(ID)
      ZH = AMIN1(ZGCI,ZFH)
   IF( AVAIL.LT.0.)  F(ID) = AMAX1(0.,F(ID)-ZH)
      HRQT = ZH*TRKT/(RAIDT*BT)
25    IWE = 0
   DO 60 KW = 1,NW
      WK(KW) = 0.
      IWB = IWE +1
      IWE = MWZ(KW)
   IF( WIND(KW).EQ.0.)  GO TO 60
      XD = X(ID) + DRIFTX(KW)/EX*HIGH
      YD = Y(ID) + DRIFTY(KW)*HIGH
      PT = PTQD*PDET(KW)
      HQT = HRQT*DWELT(KW)/TRKT
   DO 70 IW = IWB,IWE
   IF(PWK(IW).GE.0.) GO TO 70
      RANGE = ((XW(IW)-XD)*EX)**2 + (YW(IW)-YD)**2
      PS = 1. - PKQH(KW)*(1.-DEGRAD*RANGE/COVER(KW))
      WKB = 1. + PWK(IW)
      PWK(IW) = 1. + PWK(IW)*(1.-PT*(1.-PS**HQT))
   IF(KEY(5).EQ.1) PWK(IW) = AINT(PWK(IW)+RANCOM(IW))
      WK(KW) = WK(KW) + PWK(IW) - WKB
70 CONTINUE
      ZWK(KW) = ZWK(KW) + WK(KW)
60 CONTINUE
      JD = ID - IDB + 1
```

```
      IF( WVIS.GT.0.) WRITE (LB,401) QUAN(ID),JD,WK,F(ID),WIND
   20 CONTINUE
      WRITE (LB,402) NAMTYP(KD),ZWK,ZWIND
      GO TO 10
   90 IF( KEY(2).LT.2 )  RETURN
         LBL(3) = LBL(3)+1
      WRITE (3) LBL(3), WEAPON
      RETURN
   41 FORMAT(1XA6,A1,8F6.2)
   42 FORMAT(8H CARDLBL,12F6.1)
   43 FORMAT(8H CARDLBL,12F6.2)
  401 FORMAT(/2XA6,I4,12F8.1/ 4X,13F8.1)
  402 FORMAT(//1H ,A6,5H SUM ,12F8.1/12X,12F8.1/1H1/)
      END
```

```
$IBFTC ASSADK   DECK
      SUBROUTINE ASSESS
      COMMON /MON/  IDENT(8),KEY(12),LBL(4)
      COMMON /PAR/ LA,LB,RADIUS,PIE
      COMMON /FAC/ NF,NAMTYP(30),MFZ(30),VALT(30),RADT(30),
     + Y(1000),X(1000),VA(1000),AREA(1000),SECT(1000),QUAN(1000)
      COMMON /WHD/ NW,MWZ(12),YW(2000),XW(2000),PWK(2000),IPW(2000)
      DIMENSION PSI(18),CAL(18),REN(14),PSIR(18,3),CALR(18,3),UPW(13),
     + DNW(13),CSW(13),PSID(18,12),CALD(18,12),RENX(26,12),RENY(26,12)
      DIMENSION AVEDOS(3),PK(3),ZVK(3),PSFALL(40),SHIELD(8),RATIO(9)
      DIMENSION DOSPST( 400),DOSCAL( 400),DOSREN( 400),FACILE(6121)
      DIMENSION STRING(10),KLD(6,3),MAP(1200),PF(9),PV(9),TITLE(3)
      DIMENSION JALT(12),YIELD(12),DIRTY(12),HEAT(12),WORD(6),LETTER(6)
      EQUIVALENCE (NF,FACILE),(WORD,LETTER)
      EQUIVALENCE (DOSPST,MAP),(DOSCAL,MAP(401)),(DOSREN,MAP(801))
      DATA TITLE(1)/18HSURVIVRLAST FALOUT/
      DATA PST/0.,1.,2.,3.,4.,5.,9.,13.,18.,23.,30.,48.,80.,200.,350.,
     + 600.,1000.,3000./, CAL/0.,2.,4.,6.,8.,15.,25.,35.,50.,70.,90.,
     + 150.,300.,500.,700.,900.,1100.,99999./
      DATA REN/0.,50.,150.,350.,500.,750.,1000.,1500.,3000.,4500.,6000.,
     + 7333.,8667.,10000./
      DATA PSIR/14.,9.,7.,6.1,5.3,4.8,3.5,2.9,2.5,2.3,2.,1.6,1.2,.7,.6,
     + .5,.2,0.,22.1,14.8,10.7,9.2,8.,5.,4.8,2.9,3.1,2.8,2.3,1.1,6*0.,
     + 42.,36.,26.8,16.6,12.,2.,12*0./
      DATA CALR/35.,22.4,17.5,15.3,13.3,10.,8.2,7.8,6.4,5.8,5.2,4.,3.,
     + 2.5,2.2,2.,1.8,0.,48.,28.,21.1,17.9,14.5,11.,9.5,8.7,7.,6.,5.3,
     + 3.5,2.1,.8,4*0.,158.,85.,71.,58.,49.,34.,25.5,19.,7.,9*0./
      DATA UPW/8.60,6.72,5.75,5.42,5.08,4.88,4.61,4.21,3.99,3.87,3.80,
     + 3.72,3.66/, DNW/492.0,265.5,155.6,123.0,95.0,77.9,60.5,38.9,27.7,
     + 21.9,18.7,16.6,15.1/, CSW/30.7,19.0,13.8,12.6,10.6,9.69,8.35,6.48
     + ,6.03,5.73,5.54,5.25,5.04/
      DATA PSFALL/4*1.,.998,.992,.987,.978,.969,.959,.944,.931,.913,.884
     + ,.850,.812,.771,.735,.678,.622,.564,.496,.438,.380,.328,.276,.227
     + ,.180,.150,.117,.092,.070,.050,.030,.028,.020,.013,.008,.004,0.0/
      DATA SHIELD /20.,1400.,2000.,3000.,5000.,10000.,20000.,60000./
      DATA RATIO /4.,9.,25.,49.,100.,225.,400.,625.,6400./
      PSFUNC(G) = (.5+SIGN(.5,1.+G))*EXP(-AMIN1(20.,AMAX1(G,0.)**2))
COMMENCE
      READ  (LA,41) YIELD
      WRITE (LB,41) YIELD
      READ  (LA,41) HEAT
      WRITE (LB,41) HEAT
      READ  (LA,41) DIRTY
      WRITE (LB,41) DIRTY
      READ  (LA,42) JALT
      WRITE (LB,42) JALT
      DO 10 KW = 1,NW
         J = JALT(KW)+1
      IF(J.LE.0) GO TO 10
         YIE = YIELD(KW)/10.
         BY = YIE**0.333
         TY = YIE**0.420
         DIRT = 0.
      IF(J.EQ.1)  DIRT = (2.*YIE*DIRTY(KW))**0.333
         RAIN = DIRT**1.5
      DO 11 L = 1,18
   11    PSID(L,KW) = (BY*PSIR(L,J))**2
```

```
      DO 12 L = 1,18
12       CALR(L,KW) = (TY*CALR(L,J))**2
      DO 13 L = 1,13
         J = 27-L
         RENX(L,KW) = -UPW(L)*DTRT
         RENY(L,KW) = -CSW(L)*RAIN
         RENX(J,KW) =  RENX(L,KW) + DNW(L)*RAIN
13       RENY(J,KW) = -RENY(L,KW)
10 CONTINUE
20 READ  (LA,43) NAME,S,DENS,FACTOR,YTO,XTO,LAST,PSIMKL,CALMKL,RENMKL
      IF( DENS.LT.0. )  GO TO 24
      DO 25 KT = 1,NF
      IF( NAME.EQ.NAMTYP(KT) )  GO TO 24
25 CONTINUE
      GO TO 90
24 WRITE (LB,44) NAME,S,DENS,FACTOR,YTO,XTO,LAST,PSIMKL,CALMKL,RENMKL
         RENMKL = 20.*RENMKL
         VSUM = 0.
      DO 21 J = 1,3
21       VK(J) = 0.
         NP = MIN1(DENS*DENS+.1,100.1)
         AT = 3600.
         NTB = 1
      IF( DENS.LT.0. )  GO TO 26
      IF(KT.GT.1) NTB = MFZ(KT-1) + 1
         NTE = MFZ(KT)
      WRITE (LB,401) NAMTYP(KT)
         DEV = -RADT(KT)
      IF(RADT(KT).LE.0.) GO TO 23
22       AT = PIE*RADT(KT)**2
         NP = MIN1(DENS*AT+.1,400.1)
26       NP = MAX0(1,NP)
         AN = NP
         DEV = SQRT(AT/AN)
         NPY = SQRT(AN) + .5
         NPX = SQRT(AN) + .99
         WX = NPX-1
         WY = NPY-1
         WY = WY*DEV
         WX = WX*DEV
         CDEV = .5*WY
      IF( DENS.GE.0. )  GO TO 23
      WRITE (LB,47)
         YT = 60.*YTO -30. -CDEV
         XTO = -60.*XTO -30. -CDEV
         LAST = MIN0(10*LAST,300)
         LR = 0
         NTE = 6
 7       YT = YT - 60.
         XT = XTO
         EX = COS(YT/RADIUS)
         WX = WY*EX
         DEVX = DEV*EX
      READ  (LA,45) STRING
      WRITE (LB,45) STRING
         JR = 0
 8       JR = JR + 1
         LR = LR + 1
         WORD(1) = STRING(JR)
      DO 29 I = 1,5
```

```
      K = 7-I
      WORD(K) = AND(WORD(I),63)
29    LETTER(I) = LETTER(I)/64
   IF( WORD(I).LT.0.)  WORD(I) = OR(-WORD(I),32)
   DO 2 I = 1,6
   IF( LETTER(I).EQ.48 ) LETTER(I) = -1
   IF( LETTER(I).EQ.59 ) LETTER(I) = 42
   IF( LETTER(I).EQ.60 ) LETTER(I) = 10
   IF( LETTER(I).GT.60 ) LETTER(I) = LETTER(I)-3
   IF( LETTER(I).GT.47 ) LETTER(I) = LETTER(I)-4
   IF( LETTER(I).GT.31 ) LETTER(I) = LETTER(I)-3
   IF( LETTER(I).GT.15 ) LETTER(I) = LETTER(I)-3
 2 CONTINUE
23 DO 30 IT = NTB,NTE
      XT = XT + 60.
   DO 34 J = 1,3
   IF(DENS.LT.0.0) KLD(IT,J)=LETTER(IT)
      AVEDOS(J) = 0.
34    DK(J) = 0.
   IF( S.GT.0.)  READ (4) LBL(4),(PF(K),PV(K),K=1,9)
      VALUE = LETTER(IT)
   IF( DENS.LT.0. )  GO TO 27
      VALUE = VA(IT)
   IF(RADT(KT).GT.0.) GO TO 32
      NP = MIN1(AREA(IT)+.1,400.1)
   IF( RADT(KT).EQ.0.)  NP = MIN1(DENS*AREA(IT)+.1,400.1)
      NP = MAX0(1,NP)
      AN = NP
   IF( RADT(KT).EQ.0.)  DEV = SQRT(AREA(IT)/AN)
      NPY = SQRT(AN) + .5
      NPX = SQRT(AN) + .99
      WX = NPX-1
      WY = NPY-1
      WY = WY*DEV
      WX = WX*DEV
      CDEV = .5*WY
32    EX = COS(Y(IT)/RADIUS)
      YT = Y(IT) - CDEV
      XT = X(IT) - CDEV/EX
      DEVX = DEV
27 IF( VALUE.LE.0. )  GO TO 30
   DO 33 IP = 1,NP
      DOSPSI(IP) = 0.
      DOSCAL(IP) = 0.
33    DOSREN(IP) = 0.
      NWE = 0
   DO 50 KW = 1,NW
      NWB = NWE+1
      NWE = MWZ(KW)
   IF(JALT(KW).LT.0) GO TO 50
      DIST = SQRT(AMAX1(CALD(1,KW),PSID(1,KW)))
      EAST = AMAX1(DIST,-RENX(1,KW))
      SOUT = AMAX1(DIST,-RENY(1,KW))
      WEST = AMAX1(DIST,RENX(26,KW))
      YNORT = YT + WY + SOUT
      XEAST = XT + (WX+EAST)/EX
      YSOUT = YT - SOUT
      XWEST = XT - WEST/EX
   DO 60 IW = NWB,NWE
   IF( PWK(IW).GE.0.5 ) GO TO 60
```

```
      IF(YW(IW).LT.YSOUT) GO TO 60
      IF(YW(IW).GT.YNORT) GO TO 60
      IF(XW(IW).LT.XWEST) GO TO 60
      IF(XW(IW).GT.XEAST) GO TO 60
         IP = 0
         PX = (XT-XW(IW))*EX
         LX = 1
      DO 70 JX = 1,NPX
         PY = YT-YW(IW)
         LY = 27
   71 IF(LX.GT.26) GO TO 73
      IF(PX.LT.RENX(LX,KW)) GO TO 72
         LX = LX+1
         GO TO 71
   72    LY = 1
   73 DO 80 JY = 1,NPY
         IP = IP+1
   81 IF(LY.GT.26) GO TO 83
      IF(PY.LT.RENY(LY,KW)) GO TO 82
         LY = LY+1
         GO TO 81
   82    LXY = MIN0(LX,28-LX,LY,28-LY)
         DOSREN(IP) = DOSREN(IP) + REN(LXY)
   83    DIST = PX**2 + PY**2
      DO 84 L = 1,18
      IF(DIST.LT.CALD(L,KW)) GO TO 84
         DOSCAL(IP) = AMAX1(DOSCAL(IP),CAL(L)*HEAT(KW))
         GO TO 85
   84 CONTINUE
   85 DO 86 L = 1,18
      IF(DIST.LT.PSID(L,KW)) GO TO 86
         DOSPSI(IP) = AMAX1(DOSPSI(IP),PSI(L))
         GO TO 87
   86 CONTINUE
   87 IF(IP.EQ.NP) GO TO 60
   80    PY = PY + DEV
   70    PX = PX + DEVX
   60 CONTINUE
   50 CONTINUE
      DO 35 IP = 1,NP
         AVEDOS(1) = AVEDOS(1) + DOSPSI(IP)
         AVEDOS(2) = AVEDOS(2) + DOSCAL(IP)
         AVEDOS(3) = AVEDOS(3) + DOSREN(IP)
         SURVAL = PSFUNC(DOSPSI(IP)/PSIMKL)
      IF( S.LE.0.)  GO TO 52
        SURVAL = 0.
      DO 51 K = 1,9
   51    SURVAL = SURVAL + PSFUNC(DOSPSI(IP)/RATIO(K))*PV(K)/VALUE
   52   PK(1) = PK(1) + 1. - SURVAL
         PK(2) = PK(2) + SURVAL
         SURVAL = SURVAL*PSFUNC(DOSCAL(IP)/CALMKL)
         PK(2) = PK(2) - SURVAL
         PK(3) = PK(3) + SURVAL
        SURVPF = SURVAL - 1.
        J = MIN1(39.,DOSREN(IP)/RENMKL)+1
         SURVAL = SURVAL*PSFALL(J)
      IF( S.LE.0.)  GO TO 54
        SURVAL = 0.
      DO 53 K = 1,8
        SURVPF = SURVPF + PF(K)/VALUE
```

```
      IF( SURVPF.LE.0.)  GO TO 53
        J = MIN1(39.,DOSREN(IP)/SHIELD(K))+1
        SURVAL = SURVAL + SURVPF*PSFALL(J)
        SURVPF = 0.
   53 CONTINUE
   54    PK(3) = PK(3) - SURVAL
   35 CONTINUE
      DO 38 J = 1,3
         AVEDOS(J) = AVEDOS(J)/AN
         PK(J) = PK(J)/AN
      IF( DENS.GE.0. )  GO TO 38
         KLD(IT,2) = PK(1)*VALUE + .5
         KLD(IT,3) = PK(3)*VALUE + .5
   38    ZVK(J) = ZVK(J) + PK(J)*VALUE
         VALUE = VALUE * (1.-PK(1)-PK(2)-PK(3))
         VSUM = VSUM + VALUE
      IF( DENS.GE.0. )  GO TO 39
         KLD(IT,1) = VALUE + .5
      GO TO 30
   39    JT = IT-NTB+1
      WRITE (LB,402) QUAN(IT),VALUE,PK,AVEDOS,JT
      IF( LAST.EQ.0 )  VA(IT) = VALUE
   30 CONTINUE
      IF( DENS.GE.0.)   GO TO 91
         IR = LR - 300
      DO 3 J = 1,3
         IR = IR + 400
    4 DO 5 I = 1,6
      IF( KLD(I,J).GT.47 )  KLD(I,J) = 47
      IF( KLD(I,J).GT.12 )  KLD(I,J) = KLD(I,J)+3
      IF( KLD(I,J).GT.28 )  KLD(I,J) = KLD(I,J)+3
      IF( KLD(I,J).GT.44 )  KLD(I,J) = KLD(I,J)+4
      IF( KLD(I,J).EQ.10 )  KLD(I,J) = 60
      IF( KLD(I,J).EQ.42 )  KLD(I,J) = 59
      IF( KLD(I,J).LT.0  )  KLD(I,J) = 48
    5 CONTINUE
         MAP(IR) = KLD(I,J)
      IF( MAP(IR).GT.31 )  MAP(IR) = MAP(IR) - 32
      DO 6 I = 2,6
    6    MAP(IR) = MAP(IR)*64 + KLD(I,J)
      IF( KLD(1,J).GT.31 )  MAP(IR) = -MAP(IR)
    3 CONTINUE
      IF( JR.LT.10 )  GO TO 8
      IF( LR.LT.LAST )  GO TO 7
         NTB = -299
      DO 9 J = 1,3
         NTB = NTB + 400
         NTE = NTB + LAST - 1
         ZVK(J) = ZVK(J) * FACTOR
    9 WRITE (LB,46) J,TITLE(J),(MAP(IR),IR=NTB,NTE)
         VSUM = VSUM * FACTOR
   91 WRITE (LB,403) VSUM,ZVK
      GO TO 20
   90 IF( KEY(2).LT.2 )  RETURN
         LBL(1) = LBL(1)+1
      WRITE (1) LBL(1),FACILE
      RETURN
   41 FORMAT(8H CARDLBL,12F6.2)
   42 FORMAT(8H CARDLBL,12I6)
   43 FORMAT(1XA6,A1,2F6.3,2F6.2,I6,3F11.1)
```

```
   44 FORMAT(/1XA6,A1,5X2F6.3,5X2F6.2,I6,11X3F11.1)
   45 FORMAT(4H      ,10A6)
   46 FORMAT(14H THIS IS MAP ,I1,4H OF ,A6//(30X10A6))
   47 FORMAT(/18H THIS IS INPUT MAP //)
  401 FORMAT(2H0 A6,8X5HVALUE14X15HFRACTION KILLED21X12HAVERAGE DOSE/3X4
     +HSITF7X9HSURVIVING11X3HPKB4X3HPKT4X3HPKR14X3HPSI8X3HCAL8X3HREN///)
  402 FORMAT(2XA6,7XF7.1,9X,3(3XF4.2),7X,3(2XF9.1),8XI4)
  403 FORMAT(//17H VALUE SURVIVING F8.0,19H , KILLED BY BLAST F8.1,
     + 9H , THERM F8.1,7H , RAD F8.1/1H1/)
      END
```

```
$IBFTC SHLADK  DECK
      SUBROUTINE SHELL
      COMMON /PAR/ LA,LB
      COMMON /MON/  IDENT(8),KEY(12),LBL(4)
      COMMON /FAC/ NF,NAMTYP(30),MFZ(30),VALT(30),RADT(30),
     + WLAT(1000),WLON(1000),VAL(1000),SIZE(1000),SECT(1000),QUAN(1000)
      COMMON /SHEL/ ZPF(9,300),ZPV(9,300)
      DIMENSION PF(9),PV(9),SHL(9,300,2),NCAT(2),WORD(2)
      EQUIVALENCE (SHL,ZPF)
      DATA NCAT,WORD(1) /8,9,12HFALOUT BLAST/
COMMENCE
      DO 50 KT = 1,NF
      IF( IDENT(5).EQ.NAMTYP(KT) )  GO TO 51
   50 CONTINUE
   51    NUM = MFZ(KT)
      IF(KT.GT.1) NUM = NUM-MFZ(KT-1)
      IF( MOD(KEY(2),2).EQ.0 )  GO TO 60
      DO 52  J = 1,NUM
   52 READ  (3) LBL(3), (ZPF(K,J),ZPV(K,J),K=1,9)
      GO TO 70
   60 DO 61 J = 1,NUM
      DO 62 K = 1,9
         ZPF(K,J) = 0.
   62    ZPV(K,J) = 0.
   61 CONTINUE
   63 READ  (LA,41) KC,J,(PF(K),K=1,9),NAME,LOC
      WRITE (LB,41) KC,J,(PF(K),K=1,9),NAME,LOC
      DO 64 K = 1,9
   64    ZPF(K,J) = ZPF(K,J) + PF(K)
      IF(KC.LE.0) GO TO 63
   65 READ  (LA,41) KC,J,(PV(K),K=1,9),NAME,LOC
      WRITE (LB,41) KC,J,(PV(K),K=1,9),NAME,LOC
      DO 66 K = 1,9
   66    ZPV(K,J) = ZPV(K,J) + PV(K)
      IF(KC.LE.0) GO TO 65
   70 READ  (LA,42) UTIL,FRACF,KFM,KFC,FRACB,KBM,KBC
      WRITE (LB,42) UTIL,FRACF,KFM,KFC,FRACB,KBM,KBC
      IF( KEY(5).LE.0 )  GO TO 79
      DO 67 K = 1,9
         PF(K) = 0.
   67    PV(K) = 0.
         ZSHCN = 0.
      WRITE (LB,44)
      DO 71 J = 1,NUM
      IF( KEY(5).EQ.1 )  GO TO 78
         I = MFZ(KT)-NUM+J
         ZPF(9,J) = VAL(I)
   78 DO 72 K = 1,9
         PF(K) = PF(K) + ZPF(K,J)
   72    PV(K) = PV(K) + ZPV(K,J)
         SHCN = 0.
      IF(KFM.EQ.0) GO TO 74
      DO 73 K = KFM,8
   73    SHCN = SHCN + ZPF(K,J)
         SHCN = AMAX1(FRACF*ZPF(9,J)-SHCN,0.)
   74 IF(KBM.EQ.0) GO TO 76
         SUM = 0.
      DO 75 K = KBM,9
```

```
75      SUM = SUM + ZPV(K,J)
        SHCN = AMAX1(FRACB*ZPF(9,J)-SUM,SHCN)
76      ZSHCN = ZSHCN + SHCN
        ZPF(KFC,J) = ZPF(KFC,J) + SHCN
        ZPV(KBC,J) = ZPV(KBC,J) + SHCN
71 WRITE (LB,45) J,(ZPF(K,J),K=1,9),SHCN
   WRITE (LB,46)
   DO 77 J = 1,NUM
77 WRITE (LB,45) J,(ZPV(K,J),K=1,9)
   WRITE (LB,47) PF(9),(PF(K),K=1,8),ZSHCN,PV,KFC,KBC
79 IF( KEY(6).LE.0 )  GO TO 90
   DO 81 M = 1,2
      N = NCAT(M)
   WRITE (LB,48) UTIL, WORD(M)
   DO 82 J = 1,NUM
      I = MFZ(KT)-NUM+J
      V = VAL(I)
   IF(UTIL) 83,85,87
83    A = -UTIL*V
      V = V - A
   DO 84 K = 2,N
      SHL(K,J,M) = AMIN1(V,SHL(K,J,M))
84    V = V - SHL(K,J,M)
      V = V + A
   GO TO 80
85 DO 86 L = 2,N
      K = N+2-L
      SHL(K,J,M) = AMIN1(V,SHL(K,J,M))
86    V = V - SHL(K,J,M)
   GO TO 80
87    A = 0.
   DO 88 K = 2,N
88    A = A + SHL(K,J,M)
      A = AMIN1(1.,V*UTIL/A)
   DO 89 K = 2,N
      SHL(K,J,M) = A*SHL(K,J,M)
89    V = V - SHL(K,J,M)
80    SHL(1,J,M) = V
82 WRITE (LB,45) J,(SHL(K,J,M),K=1,N)
81 CONTINUE
90 IF( KEY(2).LT.2 ) RETURN
      LBL(2) = LBL(4)
   DO 91 J = 1,NUM
      LBL(4) = LBL(4) + 1
91 WRITE (4) LBL(4), (ZPF(K,J),ZPV(K,J),K=1,9)
   RETURN
41 FORMAT(1H ,I1,I4,9F7.1,A6,A4)
42 FORMAT(8H CARDLBL,F6.3,2(6X,F6.3,I6,I6))
44 FORMAT(40H1 FALLOUT SHELTERS BY AREA , CAT 9 = POP///10H  CATEGORY
  + 5X1H19X1H29X1H39X1H49X1H59X1H69X1H79X1H89X1H99X11HCONSTRUCTED// )
45 FORMAT(2X,I4,4X,9F10.1,5X,F10.1)
46 FORMAT(26H1  BLAST  SHELTERS BY AREA ///10H  CATEGORY
  + 5X1H19X1H29X1H39X1H49X1H59X1H69X1H79X1H89X1H9 //)
47 FORMAT(24H1 SHELTER SUMMARY TOTALS,25X,4HPOP F10.1///10H  CATEGORY
  + 5X1H19X1H29X1H39X1H49X1H59X1H69X1H79X1H89X1H99X11HCONSTRUCTED//
  + 10H  FALLOUT 8F10.1,15XF10.1//10H   BLAST  9F10.1,5X10HPLUS-ABOVE
  +//38H  CONSTRUCTS TO BE ADDED TO CATEGORIES,2I3,2X12HRESPECTIVELY)
48 FORMAT(30H1 SHELTER UTILIZATION ,PARAM =F6.3,35XA6///10H  CATEGORY
  + 5X1H19X1H29X1H39X1H49X1H59X1H69X1H79X1H89X1H9 //)
   END
```

```
      SUBROUTINE SURVEY
      DIMENSION S(12),Q(12),Q1(11),Q3(11),Q5(11),P(11),BACHN(11),BB(25),
     + TBSTG(25),TESTG(25),THR1(25),THR2(25),AZITH(25),ELETH(25),NN(25),
     + ANOZ(25),RATM(25),CD(25),DC(11,25),WT(25),TCROSS(3)
      DIMENSION RADAR(8),XS(8),YS(8),ZS(8),
     + D11(8),D12(8),D21(8),D22(8),D23(8),D31(8),D32(8),D33(8)
C
      RADF(X) = RA1 + RA2*COS (X+X) + RA3*COS ((X+X)+(X+X))
      GEOF(X) = X + F1*SIN(X+X) + F2*SIN((X+X)+(X+X))
CONSTANTS
      DATA BACHN/0.0,0.8,1.0,1.2,1.5,2.0,3.0,4.0,6.0,8.0,10.0/
      DATA S /11.63,2*9.5595,2*10.9189,7*8.3585/
      DATA Q/-5.,11.,25.,47.,52.,79.,90.,105.,160.,170.,200.,700./
      DATA  Q1  / .65807881,-1.0105494,-3.2178026,-6.5120337,-7.2373653,
     +-10.759466,-13.028863,-15.977971,-20.773277,-21.094464,-21.878067/
      DATA  Q3  /-.02026289,-.00001318,.013846579,.000023584,-.01593154,
     +.000032939,.024145841,.088628908,.007543412,.003507148,.002221291/
      DATA  Q5  /-4.2581444,-11956.974, 12.388264, 3126.0905,-6.5856931,
     + 6264.4021, 9.5411985, 2.7082397, 4.4164793, 7.8329587, 10.761370/
      DATA P/3709.631,474.088,51.7632,2.64295,1.29325,.026114,
     +.28265E-2,.2049698E-3,.839132E-5,.6585E-5,.3402E-5/
      DATA PI,DEGREE,OMEGA,AG/3.1415927,57.29578,.0043752695,19.061428/,
     + G,AE,AJ,AK,H /225508040.,3443.939,.32469E-2,.54E-4,.1075E-4/,
     + RA1,RA2,RA3,C8 /3438.1533,5.7978506,-.1222254E-1,23.27/,
     + R1,R2,PHO,SPO /6356.766,3432.3791,2116.2,1.48121/,
     + HING,BELOW,HFTH,HIGHT /-1234.,-3.,29.624,52./,
     + LA,LB,LC,JPRINT,TINT,TPRT,TSTOP /5,6,4,3,.01,.1,100./,
     + LOCI,IC,IPAGE,JSAT,IBIN,SHIFT /4*0,1,0./
COMMENCE
      F1 = -.33727E-2
      F2 = .567E-5
      REWIND LC
      RADIAN = 1. / DEGREE
  100 READ (LA,45)
      LAUNCH = 0
   91    IPAGE = IPAGE + 1
         LINE = 0
      WRITE (LB,45) IPAGE
CARD INPUTS
    1 READ (LA,48) K,A,B,C,D,J
      IF( K.EQ.0 )  RETURN
         K = K-6
      GO TO (80,81,82,83,84,85,86,87,88,89),K
```

```
   86 CONTINUE
   89 RETURN
CARD 0 BROUGHT IN RADAR LOCATIONS
   80 IF( J.EQ.9 )  LOCI = 0
         LOCI = LOCI + 1
      GLAT = A * RADIAN
      FLON = B * RADIAN
         UP = COS(GLAT)
         UZ = SIN(GLAT)
         CFLON = COS(FLON)
         SFLON = SIN(FLON)
      FLAT = GEOF(GLAT)
      RE = RADF(GLAT)
         HOR = RE*COS(FLAT) + C*UP
CARTESIAN COORDINATES OF SITES
         ZS(LOCI) = RE*SIN(FLAT) + C*UZ
         XS(LOCI) = HOR*CFLON
         YS(LOCI) = HOR*SFLON
      RADAR(LOCI) = D
COORDS EAST,NORTH,UP AT SITE
      D11(LOCI) = -SFLON
      D12(LOCI) =  CFLON
      D21(LOCI) = -UZ*CFLON
      D22(LOCI) = -UZ*SFLON
      D23(LOCI) =  UP
      D31(LOCI) =  UP*CFLON
      D32(LOCI) =  UP*SFLON
      D33(LOCI) =  UZ
      WRITE (LB,42) LOCI,A,B,C,D
      GO TO 1
   81    JSAT = J
         TINT = A
         TPRT = B
         TSTOP = C
         NORBIT = D
         IBIN = 1 - JSAT
      GO TO 1
   82 IF( JPRINT.GT.3 )  GO TO 21
CARD 3 BROUGHT IN INITIAL AND FINAL VALUE OF THRUST AND TIME FOR SUBSTAGE L
   83 IF( J.EQ.9 )  IC = 0
         IC = IC + 1
      THR1(IC) = A
      THR2(IC) = B
      TBSTG(IC) = C
      TESTG(IC) = D
      WRITE (LB,910)IC,A,B,C,D
      GO TO 1
```

```
CARD 4 BROUGHT IN SPECIFIC IMPULSE AND THRUST ORIENTATION FOR EACH SUBSTAGE
   84 NN(IC) = J
      RATM(IC) = A
      AZITH(IC) = B*RADIAN
      ELETH(IC) = C*RADIAN
      ANOZ(IC) = D
      WRITE (LB,911) A,D,J,C,B
      GO TO 1
CARD 6 ENTERS CD, AREA AND WEIGHT FOR EACH SUBSTAGE
   85    CD(IC) = A
      BB(IC) = B
         WT(IC) = C
      WRITE (LB,912) A,B,C,J
      IF( CD(IC).GE.0.)  GO TO 850
CARDS 5 ENTER DRAG COEFFICIENT TABLE
      READ  (LA,43) (DC(I,IC),I=1,11)
      WRITE (LB,913) BACHN,(DC(I,IC),I=1,11)
  850 IF( J.EQ.0 )  GO TO 1
        ICC = IC + J - 1
      DO 330 JJ=IC,ICC
         CD(JJ+1) = CD(JJ)
      BB(JJ+1)=BB(JJ)
      IF( CD(IC).GE.0.)  GO TO 330
      DO 852 I = 1,11
  852    DC(I,JJ+1) = DC(I,JJ)
  330 CONTINUE
      GO TO 1
CARD 7 BROUGHT INITIAL LATITUDE, LONGITUDE, HEIGHT, AND TIME
   87 T=D
      IF( J.NE.2 )  GO TO 870
      WRITE (LB,601)A,B,C,D
      X = A
      Y = B
      Z = C
      RSQ = X*X + Y*Y + Z*Z
      R = SQRT (RSQ)
      HOR = SQRT (X*X+Y*Y)
      FLON = 2.*ATAN(Y/(HOR+X))
      GLAT = ATAN ((1. + C8/R)*Z/HOR)
      GO TO 872
  870 WRITE (LB,901) A,B,C,D,J
      GLAT = A * RADIAN
      FLON = B * RADIAN
      IF( J.LE.0 )  FLON = FLON + T*OMEGA
  872 FLAT = GEOF(GLAT)
      RE = RADF(GLAT)
         UP = COS(GLAT)
         UZ = SIN(GLAT)
         CFLON = COS(FLON)
         SFLON = SIN(FLON)
      IF( J.EQ.2 )  GO TO 1
         HOR = RE*COS(FLAT) + C*UP
         Z = RE*SIN(FLAT) + C*UZ
         X = HOR*CFLON
         Y = HOR*SFLON
```

```
      GO TO 1
CARD 8 BROUGHT INITIAL VELOCITY (HEADING, INCLINE, AND MAGNITUDE)
   88 IF( J.NE.2 )  GO TO 880
      WRITE (LB,602)A,B,C
      VX = A
      VY = B
      VZ = C
      GO TO 102
  880 WRITE (LB,902) A,B,C,J
      A=A*RADIAN
      B=B*RADIAN
         U3 = COS(B)*C
         U1 = U3*SIN(A)
         U2 = U3*COS(A)
         U3 = SIN(B)*C
         VZ = UZ*U3 + UP*U2
         U2 = UP*U3 - UZ*U2
         VX = U2*CFLON - U1*SFLON
         VY = U2*SFLON + U1*CFLON
      IF( J.EQ.1 )  GO TO 102
      VX = VX - Y * OMEGA
      VY = VY + X * OMEGA
C BEGIN LAUNCH
  102 CALL SLITE(1)
         LINE = 60
         LAUNCH = LAUNCH + 1
         LEFT = 0
      ILIST = 0
         IH = 11
      ASSIGN 705 TO MM
      L = 1
      NNN = NN(L)
         W = WT(1)
      TL = T
      TB = T
      TREC = TL
      TINY = TINT/10.
CALCULATIONS FOR HEIGHT AND ACCELERATION
   60 RSQ = X*X + Y*Y + Z*Z
      R =  SQRT (RSQ)
      SIND = Z/R
      SINSQD = Z*Z / RSQ
      A TO R SQ = AE*AE/RSQ
      GR3 = G/(R*RSQ)
      COEFG = -GR3*ATORSQ*H*(1.6-3.*SINSQD)*AE
      COEFZ = -GR3*2.*ATORSQ*(AJ + AK*ATORSQ*(1.-SINSQD*7./3.))
      COEFR = -GR3*(1. + AJ*ATORSQ*(1.-5.*SINSQD) + AK*ATORSQ**2*(1.5-7.*
     +        SINSQD+10.5*SINSQD**2) - H*ATORSQ*SIND*(3.-7.*SINSQD)*AE/R)
      HOR = SQRT (X*X+Y*Y)
      FLON = 2.*ATAN(Y/(HOR+X))
         CFLON = COS(FLON)
         SFLON = SIN(FLON)
      GLAT = ATAN ((1. + C0/R)*Z/HOR)
         UP = COS(GLAT)
```

```
      UX = UP * CFLON
      UY = UP * SFLON
      UZ = SIN(GLAT)
      FLAT = GEDF(GLAT)
   RE = RADF(GLAT)
      HIGH = UZ*(Z-RE*SIN(FLAT)) + UP*(HOR-RE*COS(FLAT))
      XDOT = VX + Y*OMEGA
      YDOT = VY - X*OMEGA
   SPEED = SQRT (XDOT*XDOT + YDOT*YDOT + VZ*VZ)
      U3 = UX*XDOT + UY*YDOT + UZ*VZ
      U1 = UX*YDOT -UY*XDOT
      U2 = VZ - UZ*U3
      U3 = U3/SPEED
      HEADIN = 2.*ATAN(U1/(U2+SQRT(U1**2+U2**2)))
      ATTACK = 2.*ATAN(U3/(1.+SQRT(1.-U3**2)))
   IF( HIGH.LE.0. ) GO TO 71
   ATCH = R1*HIGH/(R2+HIGH)
   IF( ATCH.LE.Q(12) )  GO TO 54
   PH = 0.
   COEFV = 0.
      GO TO  70
53    IH = IH - 1
54 IF( ATCH.LE.Q(IH) )  GO TO 53
   IF( ATCH.LE.Q(IH+1) )  GO TO 57
      IH = IH + 1
   GO TO 54
57    PH = ALOG(1.+Q3(IH)*(ATCH-Q(IH)))
      DENSE = EXP(Q1(IH)-Q5(IH)*PH)
      PH = P(IH)*EXP(PH*(1.-Q5(IH)))
      SPSOUN = S(IH)+(S(IH+1)-S(IH))*(ATCH-Q(IH))/(Q(IH+1)-Q(IH))
   BACH = SPEED/SPSOUN
      DRAG = CD(L)
   IF( DRAG.GE.0.)  GO TO 59
   DO 56 K = 1,11
56 IF( BACH.LE.BACHN(K) )  GO TO 58
      DRAG = -DRAG
   GO TO 59
58    BACH=(BACH-BACHN(K-1))/(BACHN(K)-BACHN(K-1))
      DRAG = DC(K-1,L) + (DC(K,L)-DC(K-1,L))*BACH
59    DRAG = DRAG*BB(L)/W
      COEFV = -1.3170816*DRAG*DENSE*SPEED
      GO TO  70
C       SUBSTAGE L IS OVER
736 ASSIGN 705 TO MM
      W = W -(TESTG(L)-TB+TL)*THRUST/RATM(L)
   L = L+1
   IF( WT(L).GT.0. )  W = WT(L)
   NNN = NN(L)
70 ATHR = 0.
   IF(L.GT.IC) GO TO 701
   IF(T-TL.LT.TBSTG(L)) GO TO 701
   IF(T-TL.GT.TESTG(L)) GO TO 736
   GO TO MM,(705,732)
C       CHOICE OF ORIENTATION AT BEGINNING OF SUBSTAGE
705 ASSIGN 732 TO MM
   GO TO(710,712,708),NNN
710 IF( SPEED.LE.SPO )  GO TO 707
731 IF( HIGH.GT.HFTH )  GO TO 708
707    THX = XDOT/SPEED
      THY = YDOT/SPEED
```

```
            THZ = VZ/SPEED
         GO TO 708
  712       U3 = COS(ELETH(L))
            U1 = U3 * SIN(HEADIN+AZITH(L))
            U2 = U3 * COS(HEADIN+AZITH(L))
            U3 = SIN(ELETH(L))
            THX = U3*UX - U2*UZ*CFLON - U1*SFLON
            THY = U3*UY - U2*UZ*SFLON + U1*CFLON
            THZ = U3*UZ + U2*UP
         GO TO 708
C          ORIENTATION DURING SUBSTAGE
  732 GO TO(733,708,708),NNN
  733 IF( SPEED.GT.SPO )  GO TO 731
  708 THRUST = THR1(L) + ((THR2(L) - THR1(L))/(TESTG(L) - TBSTG(L)))*(T-
     +            TL - TBSTG(L))
         IF(THRUST.LE.0.) GO TO 701
            ATHR = (1.+(PHO-PH)*ANOZ(L)/AMAX1(THR1(L),THR2(L)))*AG*THRUST/W
            W = W - THRUST*AMIN1(TINT,T-TL-TBSTG(L))/RATM(L)
  701       AX = ATHR*THX + COEFV*XDOT + X*COEFR
            AY = ATHR*THY + COEFV*YDOT + Y*COEFR
            AZ = ATHR*THZ + COEFV* VZ  + Z*(COEFR+COEFZ) + COEFG
CHECKS FOR PRINTING AND STOPPING
         GLAT = GLAT*DEGREE
         GLON = FLON - T * OMEGA
         IF(GLON.LT.-PI) GLON = PI + AMOD(GLON,PI)
         GLON = GLON*DEGREE
            HEADIN = HEADIN*DEGREE
            ATTACK = ATTACK*DEGREE
         IF(T+TINY.LT.TREC) GO TO 64
C             WRITE BINARY TAPE ONLY
         IF( IBIN )  62,60,61
   62 WRITE (LC) T,GLAT,GLON,HING,X,Y,Z
         IBIN=0
         GO TO 60
   61 WRITE (LC) T,GLAT,GLON,HIGH,X,Y,Z
   60 IF( MOD(JPRINT,2).EQ.0)  GO TO 67
C             PRINT TRAJECTORY DATA
            ATOT = SQRT(AX*AX+AY*AY+AZ*AZ)/AG
            ATHRG = ATHR/AG
            DRAG = -COEFV*SPEED/AG
         IF(LINE.LT.54) GO TO 66
         IPAGE = IPAGE + 1
         LINE = 0
         WRITE (LB,45) IPAGE
         WRITE (LB,44)
   66 WRITE (LB,47) T,GLAT,GLON,HIGH,HEADIN,ATTACK,SPEED,ATOT,ATHRG,DRAG
            LINE = LINE + 1
   67 IF( JPRINT.GT.1 )  GO TO 29
  230       TREC = T + TPRT
   64 IF(HIGH.GT.HIGHT) GO TO 19
         IF(LEFT.EQ.0) GO TO 206
         IF( JSAT.EQ.1 )  GO TO 71
         IF( HIGH.GT.0. )  GO TO 206
C  THE END
   71 WRITE (LB,49)  LAUNCH,GLAT,GLON,T
         GO TO 1
   19 LEFT = 1
  206 IF( T.GE.TSTOP )  GO TO 90
C  INTEGRATION
            TB = T
```

```
      ZB = Z
      T = VIND(T,TINT)
      X = DPNV ( X, VX)
      Y = DPNV ( Y, VY)
      Z = DPNV ( Z, VZ)
      VX= DPNV ( VX,AX)
      VY= DPNV ( VY,AY)
      VZ= DPNV ( VZ,AZ)
      IF( JSAT.EQ.0 )  GO TO 68
      IF( Z*ZB.GT.0. )  GO TO 68
C     EQUATOR CROSSING
      ILIST=ILIST+1
      TCROSS(ILIST) = T - (VZ/AZ)*(1.-SQRT(VZ*VZ-2.*Z*AZ))
      IBIN=1
      IF(ILIST-3)69,1012,2012
 1012 IBIN=-1
      GOTO68
 2012 IBIN=0
      GO TO 68
C OVERTIME
   90 IF( IBIN.NE.0 )  WRITE (LC) T,GLAT,GLON,HING,X,Y,Z
      INDEX=((ILIST-1)/2)*2+1
      IF( INDEX.GT.2 )  GO TO 1014
      WRITE (LB,1015)
      GOTO499
 1014 PERIOD = (TCROSS(INDEX)-TCROSS(1))/(.5*(FLOAT (INDEX)-1.0))
      WRITE (LC) PERIOD
      DO4000JLIST=1,ILIST
 4000 WRITE (LB,4001)JLIST,TCROSS(JLIST)
      WRITE (LB,2016) PERIOD
  499 WRITE (LB,606)T
      GO TO 1
C     TAPE INPUTS
   21 READ  (LC) T,GLAT,GLON,HIGH,X,Y,Z
      T = T + SHIFT
C     EARTH (ROT)ATION  TO PRESENT TIME
   29 ROT=T*OMEGA
      COSOM=COS (ROT)
      SINOM=SIN (ROT)
      XP=X*COSOM + Y*SINOM
      YP=-X*SINOM+ Y*COSOM
C     TRANSLATE
      DO 25 I = 1,LOCI
      XV=XP-XS(I)
      YV=YP-YS(I)
      ZV=Z -ZS(I)
      PZ = XV*D31(I)+YV*D32(I)+ZV*D33(I)
      IF(PZ.LT.-BELOW) GO TO 25
C     HEIGHT ABOVE RADAR PLANE
      PX = XV* D11(I)+ YV*D12(I)
      VER2 = PX*PX + PY*PY
      PY = XV*D21(I)+YV*D22(I)+ZV*D23(I)
      VER  = SQRT (VER2)
      RHO2 = VER2 + PZ*PZ
C   RADAR SPHERICAL COORDINATES
      RHO  = SQRT (RHO2)
      ELEV = ATAN(PZ/VER) * DEGREE
      AZI = 2.*ATAN(PX/(VER+PY)) * DEGREE
      IF(LINE.LT.54) GO TO 24
      IPAGE = IPAGE + 1
```

```
      LINE = 0
      WRITE (LB,45) IPAGE
      WRITE (LB,44)
   24 WRITE (LB,46) RHO,AZI,ELEV,RADAR(I)
         LINE = LINE + 1
   25 CONTINUE
      IF( JPRINT.LT.4 )  GO TO 230
      IF(HIGH.GT.0.) GO TO 21
         NORBIT = NORBIT - 1
      IF( NORBIT ) 22,91,21
   22    NORBIT = NORBIT + 2
      IF( NORBIT.EQ.0 )  GO TO 91
      READ  (LC) PERIOD
      REWIND LC
      SHIFT = SHIFT + PERIOD
      GO TO 21
   48 FORMAT(I5,4F15.8,I5)
   43 FORMAT(12F6.3)
   45 FORMAT(72H
     +                    ,40X,6H PAGE ,I2)
   42 FORMAT(I6,4F9.4)
  910 FORMAT(13HOSUBSTAGE NO.I2,17H THRUST,INITIAL =F10.2,14H LBS,  FINA
     1L =F10.2,16H LBS,   FROM T =F7.3,11H MIN TO T =F7.3,4H MIN)
  911 FORMAT(20HO SPECIFIC IMPULSE =F7.4,14H MIN, NOZZLE =F6.3,12H SQ.FT
     +., G =I2,15H, THRUST ELEV =F6.2,17H DEG., AZ.CORR. =F7.2,6H CLOCK)
  912 FORMAT(18HO COEF. OF DRAG = F5.2,24H, AREA OF CROSSECTION = F8.2,2
     +6H SQ.IN., INITIAL WEIGHT = F9.2,21H LBS, EXTRA SUBSTAGES I2)
  913 FORMAT(8HOMACH NO,11F10.1/10H   CD      ,11F10.3)
  601 FORMAT(22HOINITIAL POSITION X = F9.3, 8H ,  Y = F9.3, 8H ,  Z = F9
     1.3,20H NM,  INITIAL TIME =F8.3,4H MIN)
  602 FORMAT(24HOINITIAL VELOCITY, VX = F14.7,9H ,  VY = F14.7,9H ,  VZ
     1= F14.7,7H NM/MIN)
  901 FORMAT(30HOINITIAL GEODETIC COORDINATES F7.3,8H NORTH, F8.3,7H EAS
     +T, F7.3,27H NM HIGH,    INITIAL TIME =F8.3,16H MIN, AXES FIXEDI2)
  902 FORMAT(29HOINITIAL AIRSPEED, DIR.ANGLE F7.2,13H,INCLINATION F6.2,
     + 11H MAGNITUDE F9.4,7H NM/MIN ,26X10HAXES FIXEDI2)
   44 FORMAT(120H    TIME        GLAT        GLON       HEIGHT     HEADING
     +  INCLINE     SPEED       RANGE     AZIMUTH       ELEV     SENSOR//)
   46 FORMAT(75X,3F11.3,4XF6.0)
   47 FORMAT(F9.3,9F11.3,4XF6.0)
   49 FORMAT(/ 9H MISSILE ,I4,11H LANDED AT ,F7.3,13H DEG. NORTH, F8.3,
     + 17H DEG. EAST AFTER F7.3,16H MIN. OF FLIGHT.  )
 1015 FORMAT(23HODID NOT COMPLETE ORBIT)
 4001 FORMAT(1H ,9HCROSSING I5,6X,5HTIME=E12.6,6X,5HLONG=F8.3)
 2016 FORMAT(15HONODAL PERIOD= E12.6)
  606 FORMAT(27HOINTEGRATION STOPPED AT T =F9.3,4H MIN)
      END
```

```
$IBFTC CMD1DK  DECK
      SUBROUTINE COMAND
      COMMON/PAR/LA,LB,DAT(4),CONV,SCALAT,SCALON,POLAT,POLON
      COMMON/FAC/NF,NAMTYP(30),MFZ(30),DATT(60),Y(1000),X(1000)
      COMMON/WHD/NW,MWZ(12),YW(2000),XW(2000)
      COMMON/MAP/NAME,CHAR,YN,YS,YSC,XV,XE,XSC,AMAP(50,120)
      DIMENSION NWTYP(12)
      DATA NWTYP(1)/72H     1     2     3     4     5     6     7     8
     1    9    10    11    12/, BLNK/6H     /,          KW/2000/,KF/10
     200/,MY/50/,MX/120/,W/1HW/,F/1HF/
      READ (LA,40) YLAT,XLON,DLAT,DLON,SLAT,SLON
      WRITE(LB,40) YLAT,XLON,DLAT,DLON,SLAT,SLON
      YN = SCALAT*(1.-CONV)*AINT(YLAT) + POLAT
      XV = SCALON*(1.-CONV)*AINT(XLON) + POLON
      YD = SCALAT*(1.-CONV)*AINT(DLAT)
      XD = SCALON*(1.-CONV)*AINT(DLON)
      YSC = SCALAT*(SLAT-CONV*AINT(SLAT))
      XSC = SCALON*(SLON-CONV*AINT(SLON))
      YS = YN-YD
      XE = XV-XD
      NY = (YD/YSC)+1.
      NX = (XD/XSC) +1.
      IF(NY.GT.MY.OR.NX.GT.MX) RETURN
      DO 11 IY=1,NY
      DO 11 IX = 1,MX
   11 AMAP(IY,IX) = BLNK
   12 READ (LA,41) NAME,TY,CHAR
      WRITE(LB,41) NAME,TY,CHAR
      IF(TY.NE.W) GO TO 13
      CALL COMPUT(NW,MWZ,NWTYP,KW,YW,XW)
      GO TO 12
   13 IF(TY.NE.F) GO TO 14
      CALL COMPUT(NF,MFZ,NAMTYP,KF,Y,X)
      GO TO 12
   14 YMN = AINT(YLAT)
      YMS = YMN-AINT(DLAT)
      XMW = AINT(XLON)
      XME = XMW-AINT(DLON)
      WRITE(LB,42) YMS,YMN,XME,XMW
      WRITE(LB,43) ((AMAP(IY,IX),IX=1,MX),IY=1,NY)
      RETURN
   40 FORMAT(8H CARDLBL 6F8.2)
   41 FORMAT(1X6,1X2A1)
   42 FORMAT(1H115X23HMAP OF AREA BOUNDED BY F4.0,5H AND F4.0,23H DEGREE
     1S LATITUDE, AND F4.0,5H AND F4.0,19H DEGREES LONGITUDE.)
   43 FORMAT(1H 120A1)
      END
```

SECTION III

SUPPORTING PROGRAMS

FUNCTIONS

1. RANDOM
2. PKAY
3. DIST
4. LOCATE
5. VIND/DPNV

SUBROUTINES

1. RANORM
2. COORD
3. COMPUT
4. DUMMY

FUNCTIONS

```
$IBFTC RND3DK   DECK
      FUNCTION RANDOM(X)
      COMMON /PAR/ DAT(4),RANDA,RANDB
      DO 12 J = 1,2
         RANDB = RANDB + RANDA
         RANDA = RANDB - RANDA
      IF(RANDB.GE.1.0) RANDB = RANDB - 1.0
   12 CONTINUE
      RANDOM = RANDA
      RETURN
      END
```

The purpose of the function RANDOM(X) is to generate a new random number from a uniform distribution on the interval 0 to 1. The procedure used is a Fabonachi type in which the last two numbers generated, RANDA and RANDB, are remembered and used to form the next one. Two non-commensurated numbers are supplied from the compiled parameter list to start the process.

```
$IBFTC PKA2DK  DECK
      FUNCTION PKAY(AT,AW,SZ)
      COMMON /PAR/ DAT(14),CPK1,CPK2,CPK3
      DIMENSION NC(50)
      DATA(NC(J),J=1,50)/12*61,55,43,37,2*31,4*19,2*13,11*7,8*4,5*3,3*2,18M
         ARG = AW/AT
         ANC = SZ/AT + CPK3
         ANC = AMAX1(1.,ARG)/ANC
         PS = EXP(-ANC)
         ANC = 1.
         K = 100.*SQRT(ARG)
      IF(K.LT.50) ANC = NC(K+1)
         PS = 1. - (1.-PS)*AMIN1(1.,ARG*CPK1*ANC**CPK2)
      PKAY = 1. - PS**(1./ANC)
      RETURN
      END
```

The purpose of this function PKAY(AT,AW,SZ) is to compute an estimate of the average fraction of remaining value destroyed by each weapon of lethal area AW and variance SZ that may be aimed at a target of area AT. The shape factors, CPK1,2,3 are normally set so that for targets of area less than the weapon (aimpoints all at center of target) the function reduces to the usual single shot probability. Targets of large area and unusual shape might be approximated by other shape factors.

```
$IBFTC DISIDK  DECK
      FUNCTION DIST(YA,XA,YB,XB)
      COMMON /PAR/ LA,LB,RADIUS
         YAR = YA/RADIUS
         YBR = YB/RADIUS
         XR = (XA-XB)/RADIUS
         COSANG = SIN(YAR)*SIN(YBR) + COS(YAR)*COS(YBR)*COS(XR)
      DIST = 2.*RADIUS*ATAN(SQRT((1.-COSANG)/(1.+COSANG)))
      RETURN
      END
```

The function DIST(YA, XA, YB, XB) computes the great circle distance in nautical miles between any two points A and B on the surface of the globe. The latitude and longitude of each of the points, expressed also in nautical miles (normal machine) form, are the arguments. RADIUS refers to the radius of the earth.

```
$IBFTC LOCIDK  DECK
      FUNCTION LOCATE(Y,X,K)
      COMMON /PAR/ LA,LB,RADIUS
      COMMON /FAC/ NF,NAMTYP(30),MFZ(30),VALT(30),RADT(30),
     + U(1000),V(1000),W(1000),R(1000),S(1000),T(1000)
        KB = 1
      IF( K.GT.1 )  KB = MFZ(K-1) + 1
        KE = MFZ(K)
        A = Y/RADIUS
        B = X/RADIUS
        THED = COS(A)
        CHED = COS(B)*THED
        SHED = SIN(B)*THED
        THED = SIN(A)
        DPL = CHED*U(KE) + SHED*V(KE) + THED*W(KE)
      DO 10 J = KB,KE
        DPR = CHED*U(J) + SHED*V(J) + THED*W(J)
      IF( DPL.GE.0.)  GO TO 10
      IF( DPR.LT.0.)  GO TO 10
        DPM = CHED*R(J) + SHED*S(J) + THED*T(J)
      IF( DPM.GE.0.)  GO TO 20
   10   DPL = DPR
        J = KE + 1
   20 LOCATE = J - KB + 1
      RETURN
      END
```

The function LOCATE(Y,X,K) determines in which section of a mesh a particular point lies. The arguments are the latitude and longitude of the point and the facility type number K of the desired mesh (usually preselected by the calling program). The procedure tests all sectors in order from one to last until a positive association is determined. If no association is found, the point is automatically classified as last + one. A point is in a sector when it lies to the right of the left hand heading looking outward, on or to the left of the right hand heading, and on or exterior of the section arc. Exterior means to the left as the sector posts are traversed in increasing order.

```
      FUNCTION VIND(X,XD)
      DIMENSION B(16),C(4),DPR(2),XX(50),XD1(50),XD2(50),XD3(50)
      DOUBLE PRECISION DP
      EQUIVALENCE (DPR,DP)
      DATA B/1.,.333333333,.5,1.5,2.291666666,-2.458333333,1.541666667,
     + -.375,1.125,-1.125,.375,1.916666666,-1.333333338,.416666661,
     + -.833333333,.4166666667/
      IF( XD.NE.0.)  GO TO 70
         H = 0
         M = 0
         TT = 0
      DO 1 J = 1,50
    1    XX(J) = 0
      GO TO 120
   70 IF( XD.EQ.H .OR. M.GT.4 )  GO TO (74,61,68,67,65,65,63,62),M
         H = XD
         M = 6
         C(1) = B(1)
         C(2) = 0
         C(3) = 0
         C(4) = 0
      GO TO 73
   65    M = M-1
         C(1) = B(M-2)
         C(2) =-B(M-2)
      GO TO 120
   67    M = 7
         C(1) =  B(4)
         C(2) = -B(3)
         DO 3 J = 1,50
    3    XD1(J) = XD3(J)
      GO TO 74
   63    M = 3
      DO 4 J = 1,3
    4    C(J) = B(J+13)
      GO TO 145
   68    M = 8
      DO 5 J = 1,3
    5    C(J) = B(J+11)
      GO TO 73
   62    M = 2
      DO 6 J = 1,4
    6    C(J) = -B(J+7)
      GO TO 145
   61    M = 1
      DO 7 J = 1,4
    7    C(J) = B(J+4)
   73    K = 1
   74    DPR(1) = X
         DPR(2) = TT
         DP = H +  P
         TT = DPR(2)
      GO TO 103
  145    K = 0
  120    DPR(1) = X
```


AFSC HQ FORM MAY 65 0-88-1

```
103     I = 1
      GO TO 54
      ENTRY DPNV(X,XD)
      IF( I.GT.50 )  STOP
         R = C(4)*XD3(I)
      IF( K.EQ.1 )  XD3(I) = XD2(I)
         R = R + C(3)*XD2(I)
      IF( K.EQ.1 )  XD2(I) = XD1(I)
         R = R + C(2)*XD1(I)
      IF( K.EQ.1 )  XD1(I) = XD
         R = R + C(1)*XD
         DPR(1) = X
         DPR(2) = XX(I)
         DP = (R*H) + DP
         XX(I) = DPR(2)
         I = I+1
54       VIND = DPR(1)
      RETURN
      END
```

The function VIND(X, XD) with alternate entry point DPNV(X, XD) is a general purpose routine to integrate up to fifty simultaneous first order differential equations. It is used by SURVEY to integrate the three second order (equivalently six first order) differential equations governing a missile or satellite orbit. While using only single precision instructions and arguments, it achieves double precision accuracy by itself saving, then retreiving next call the second word worth of digits for each variable. A call to VIND is used to increase the independent variable where X is the current time and XD the increment; a call to DPNV is made for each dependent

variable to be integrated where X, XD are the current values and first derivative respectively.

This program was originally obtained as a binary deck in 704 machine language from Lincoln Laboratories for use with Fortran II programs. It was rewritten by the author in MAP language to be compatible with the Fortran IV computer. It is this form or its corresponding binary deck which matches the SURVEY program included in this report. However MAP programs are not easy to understand. Since the MAP version has been rewritten in Fortran IV under another contract (to make it useable on a CDC 3600 machine) this more readable version is used here. It has been tested on the 7094 with SURVEY. (A few card modifications are required in the latter.)

SUBROUTINES

```
$IBFTC RAN1DK  DECK
      SUBROUTINE RANORM(X,Y)
         A = 6.2831853 * RANDOM(X)
         R = SQRT(-2.0 * ALOG(RANDOM(Y)))
         X = R * COS(A)
         Y = R * SIN(A)
      RETURN
      END
```

The routine RANORM(X Y) computes two random rectangular displacements based upon a circular normal distribution with mean zero and standard deviation unity. The procedure is to combine a random radius, computed as the inverse transformation of a uniformly sampled probability supplied by the RANDOM function, with as uniformly sampled angle from 0 to 2π. The results usually need to be scaled to the proper deviation and centered on the aimpoint to form an impact point.

```
$IBFTC CRD3DK  DECK
      SUBROUTINE COORD(MW,K)
      COMMON /MON/  IDENT(3),X,Y,T,R,BYPASS,KEY(12),LBL(4)
      COMMON /PAR/ LA,LB,RADIUS,DAT(10),DILATE
      COMMON /WHD/ NW,MWZ(12),YWHD(2000),XWHD(2000),PWK(2000)
      DIMENSION AIMX(61),AIMY(61),ARAD(8),CORK(24),LIST(61)                 COD
      DATA (ARAD(J),J=1,8)/2.0,3.464,4.0,5.292,6.0,6.928,7.211,8.0/,        IBM
     1(LIST(J1),J1=1,61)/0,6*1,6*2,6*3,12*4,6*5,6*6,12*7,6*8/,              IBM
     2(AIMX(J2),J2=1,61)/0.0,-1.732,1.732,-1.732,1.732,2*0.0,-3.464,        IBM
     33.464,-1.732,1.732,-1.732,1.732,-3.464,3.464,-3.464,3.464,2*0.0,      IBM
     4-5.196,5.196,-1.732,2*1.732,-1.732,-5.196,5.196,-3.464,3.464,         IBM
     5-3.464,3.464,-5.196,5.196,-5.196,5.196,2*0.0,-6.928,6.928,-3.464,     IBM
     63.464,-3.464,3.464,-6.928,6.928,-1.732,1.732,-5.196,5.196,6.928,      IBM
     7-6.928,1.732,-1.732,-5.196,5.196,2*0.0,6.928,2*-6.928,6.928/,         IBM
     8(AIMY(J3),J3=1,61)/0.0,1.0,2*-1.0,1.0,2.0,-2.0,2*0.0,3.0,2*-3.0,      IBM
     93.0,2.0,2*-2.0,2.0,4.0,-4.0,1.0,-1.0,-5.0,5.0,-5.0,5.0,-1.0,1.0,      IBM
     1-4.0,2*4.0,-4.0,3.0,2*-3.0,3.0,6.0,-6.0,2*0.0,-6.0,2*6.0,-6.0,2.0,IBM
     2-2.0,7.0,-7.0,-5.0,5.0,2.0,-2.0,7.0,-7.0,5.0,-5.0,8.0,-8.0,4.0,       IBM
     3-4.0,4.0,-4.0/,                                                       IBM
     4(CORK(J4),J4=1,24)/3*0.0,1.0,0.0,-1.0,0.0,-1.0,.866,.5,-.866,.5,      IBM
     50.0,1.0,-1.0,2*0.0,-1.0,1.0,0.0,2*1.0,1.155,1.414/                    IBM
COMMENCE
            EX = COS(Y/RADIUS)
            I = K+1
            RT = DILATE*(T-R)
      IF(RT)2603,2603,2610
 2603       K = K+MW
      DO 2604 J = I,K
            XWHD(J)=X
 2604       YWHD(J)=Y
      IF(KEY(6).LE.0) WRITE (LB,865) Y,X,MW
 2606 RETURN
 2610       MUCH = MINO(MW,61)
            MW = K+MW
 2620       M=MUCH
 2621 IF(M-4)2640,2640,2622
 2622       IR=LIST(M)
      IF(T-R-R*ARAD(IR))2623,2630,2630
 2623       M=M-1
      GO TO 2621
 2630       RR=AMIN1(RT/ARAD(IR),2.*R)
      DO 2631 J=1,M
            K = K+1
      IF(BYPASS) 2632,2632,2633
 2633       XWHD(K)=X+BYPASS*(RR*AIMY(J))**2
      GO TO 2631
 2632       XWHD(K)=X+RR*AIMX(J)/EX
 2631       YWHD(K)=Y+RR*AIMY(J)
      GO TO 2651
```

```
2640      RR=RT
          S = DILATE*R*CORK(M+20)
      IF(RT-S)2642,2642,2641
2641      RR=(RT+S)*.5
2642 DO 2643 J=1,M
          K = K+1
          L=(M-1)*M+2*J
      IF(BYPASS) 2644,2644,2645
2645      XWHD(K)=X+BYPASS*(RR*CORK(L))**E
      GO TO 2643
2644      XWHD(K)=X+RR*CORK(L-1)/EX
2643      YWHD(K)=Y+RR*CORK(L)
2651     MUCH=MUCH-M
      IF(MUCH.GT.0) GO TO 2620
      IF(KEY(6).LE.0) WRITE (LB,861)(YWHD(J),XWHD(J),J=1,K)
      GO TO 2650
2661      I = K+1
          K = MIN0(MW,K+61)
      DO 2662 J = I,K
          XWHD(J) = XWHD(J-61)
2662      YWHD(J) = YWHD(J-61)
2650 IF(MW-K) 2660,2660,2661
2660 RETURN
 865 FORMAT(8X,2F8.2,I8,6H TIMES)
 861 FORMAT(5(8X,2F8.2))
     END
```

The COORD(MW,K) subroutine is used to find aimpoints for the warheads of each type assigned by STRIKE to any targets (except missile blocks which are deployed differently by STRIKE itself). The target coordinates, radius, and warhead radius is passed to COORD indirectly by STRIKE having stored them in the common block /MON/. Depending upon the relative size of the target to warhead radii aimpoints are chosen either at the target coordinates or in a patterned array around them. The array will also be influenced by the number of weapons MW to be put down and may if MW is large enough give

multiple coverage. Normally the patterns are not scaled but they may be expanded or contracted by changing the parameter DILATE. The impact coordinates are stored, segregated by warhead type, by means of the index argument K.

```
$IBFTC CMPIDY   DECK
      SUBROUTINE COMPUT(J,JZ,JTYP,JI,YJ,XJ)
      COMMON/MAP/NAME,CHAR,YN,YS,YSC,XV,XE,XSC,AMAP(50,120)
      DIMENSION JZ(J),JTYP(J),YJ(JI),XJ(JI)
      DATA BLNK/6H      /,ASTRK/1H*/
      DO 20 IT = 1,J
      IF(NAME.EQ.JTYP(IT)) GO TO 21
   20 CONTINUE
      RETURN
   21 IB = 1
      IF(IT.GT.1) IB = JZ(IT-1)+1
      IE = JZ(IT)
      DO 22 IN = IB,IE
      IF(YJ(IN).GT.YN.OR.YJ(IN).LT.YS.OR.XJ(IN).LT.XV.OR.XJ(IN).GT.XE)
     +   GO TO 22
      KY = ABS((YN-YJ(IN))/YSC)+1.
      KX = ABS((XV-XJ(IN))/XSC)+1.
      IF(AMAP(KY,KX).EQ.BLNK) GO TO 23
      AMAP(KY,KX) = ASTRK
      GO TO 22
   23 AMAP(KY,KX) = CHAR
   22 CONTINUE
      RETURN
      END
```

The COMPUT subroutine looks up the coordinates of all the sites of a specific type and checks them with the boundaries of the map being constructed. Those outside are ignored. For the interior points the appropriate symbol is stored in the memory square corresponding to the rounded off coordinates. If more than two items occupy the same square on the same map they are replaced by an asterik.

```
$IBFTC SRVDMK
      SUBROUTINE SURVEY
      COMMON X
         X = 2.
      RETURN
      END
```

It is sometimes desirable to substitute a "DUMMY" subroutine in place of one or more of the main routines in runs where it is known that such routines will not be exercised. This is particularly true when submitting the program in card form, either source or binary decks as considerable weight and space may be saved. This program is a simple DUMMY which has been so used in this instance to replace the SURVEY routine.

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R&D

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

1. ORIGINATING ACTIVITY (Corporate author) Academy for Interscience Methodology	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP N/A
3. REPORT TITLE BEACON II: Description and Programs	
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Final	
5. AUTHOR(S) (Last name, first name, initial) Painter, Norman H.	
6. REPORT DATE November 1966	7a. TOTAL NO. OF PAGES 24 / 7b. NO. OF REFS 1
8a. CONTRACT OR GRANT NO. AF 18(600)-2793 b. PROJECT NO. 5401 c. 15 d.	9a. ORIGINATOR'S REPORT NUMBER(S) AIM 67-T-10 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) N/A
10. AVAILABILITY/LIMITATION NOTICES Qualified requesters may obtain copies of this report from DDC.	
11. SUPPLEMENTARY NOTES N/A	12. SPONSORING MILITARY ACTIVITY Hq AFSC (SCLA) Andrews AFB, Washington, D.C. 20331

13. ABSTRACT

BEACON denotes a set of programs organized to form a computing tool to study defense of the continental United States against strategic nuclear attack. The purpose is to obtain a feel for the possibilities of newer technological concepts useable in continental defense. Thus, BEACON has been designed toward embracing any set of potentially interesting interacting systems of the near future. The method of procedure has been to concentrate on fundamental logical or physical processes of attack and defense on a global scale and to keep the structure modular, replaceable, and open-ended.

DD FORM 1 JAN 64 1473

UNCLASSIFIED

Security Classification

Figure 1

AFSCR 80-20 UNCLASSIFIED

Security Classification

14. KEY WORDS	LINK A ROLE	LINK A WT	LINK B ROLE	LINK B WT	LINK C ROLE	LINK C WT
Defense of Continental United States (CONUS)						
Computing tool						
Open-ended						
Model						

INSTRUCTIONS

1. ORIGINATING ACTIVITY. Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through _____."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through _____."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through _____."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS), (S), (C), or (U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.

UNCLASSIFIED

Security Classification

Figure 1