AN ANTI-RADIATION PROJECTILE (ARP) TERMINAL EFFECTS SIMULATION COMPUTER PROGRAM (ARPSIM)

R. D. WEBSTER

JUNE 1981

US ARMY ARMAMENT RESEARCH AND DEVELOPMENT COMMAND
LARGE CALIBER WEAPON SYSTEMS LABORATORY
DOVER, NEW JERSEY

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED.
The views, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

Destroy this report when no longer needed. Do not return it to the originator.
This report is the documentation for a computer code developed primarily to aid development engineers by providing estimates of the relative importance of components in terms of effectiveness.

The ARPSIM computer model was developed in support of a requirement to estimate the effectiveness of the various kill mechanisms (fragmentation, antenna blast, vehicle body blast, and direct hit) of an Anti-Radiation Projectile (ARP) against a typical air defense radar-emitting target. A Monte Carlo technique...
is used to generate estimates of the probability of kill for a single ARP fired against a single target. The influence of various fuzing schemes and guidance errors are considered in determining burst points.
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Program Flow</td>
<td>1</td>
</tr>
<tr>
<td>Terminal Effects</td>
<td>3</td>
</tr>
<tr>
<td>Coordinate System</td>
<td>3</td>
</tr>
<tr>
<td>Attack Angle</td>
<td>4</td>
</tr>
<tr>
<td>Guidance Errors</td>
<td>4</td>
</tr>
<tr>
<td>Fuzing</td>
<td>6</td>
</tr>
<tr>
<td>Target</td>
<td>8</td>
</tr>
<tr>
<td>Burst Point</td>
<td>8</td>
</tr>
<tr>
<td>Direct Hit</td>
<td>10</td>
</tr>
<tr>
<td>Blast</td>
<td>10</td>
</tr>
<tr>
<td>Fragmentation</td>
<td>10</td>
</tr>
<tr>
<td>Monte Carlo Estimates</td>
<td>13</td>
</tr>
<tr>
<td>Conclusions</td>
<td>13</td>
</tr>
<tr>
<td>Recommendations</td>
<td>14</td>
</tr>
<tr>
<td>References</td>
<td>15</td>
</tr>
<tr>
<td>Appendixes</td>
<td></td>
</tr>
<tr>
<td>A User Guide</td>
<td>17</td>
</tr>
<tr>
<td>B Example</td>
<td>27</td>
</tr>
<tr>
<td>C FORTRAN Listing</td>
<td>37</td>
</tr>
<tr>
<td>Distribution List</td>
<td>103</td>
</tr>
</tbody>
</table>

---

Accession For

- NTIS GRAI
- DTIC TAB
- Unannounced
- Justification

By Distribution/

Availability Codes

Dist | Avail and/or Special
--- | ---
A | Special
## FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Program flow</td>
<td>2</td>
</tr>
<tr>
<td>2. Coordinate system</td>
<td>4</td>
</tr>
<tr>
<td>3. Attack angle</td>
<td>4</td>
</tr>
<tr>
<td>4. Guidance errors</td>
<td>5</td>
</tr>
<tr>
<td>5. Fuzing angle</td>
<td>7</td>
</tr>
<tr>
<td>6. Linear fuzing</td>
<td>7</td>
</tr>
<tr>
<td>7. Target description</td>
<td>9</td>
</tr>
<tr>
<td>8. Blast radii vs height</td>
<td>11</td>
</tr>
<tr>
<td>9. Blast kill probability vs height</td>
<td>11</td>
</tr>
<tr>
<td>10. Radar blast function</td>
<td>11</td>
</tr>
<tr>
<td>11. Fragmentation grid interpolation</td>
<td>12</td>
</tr>
</tbody>
</table>
INTRODUCTION

ARPSIM is a computer program developed to provide estimates of the terminal effectiveness of an Anti-Radiation Projectile (ARP) fired against an air defense, radar-emitting target.

The primary objective of ARPSIM is to provide the user with a tool to parametrically ascertain the sensitivity of the ARP to warhead, guidance, and fuzing design changes.

The ARPSIM model simulates single round terminal conditions from the time when the ARP is flying a straight line trajectory at some fixed attack elevation in the vicinity of the target. Trajectories are determined from guidance errors distributed about a specified homing point. No further trajectory alterations are made. Fuzing points on the target are specified, and when fuzing conditions are satisfied, a burst point is established along the selected trajectory. The proximity of the burst point to the target determines the magnitude of kill probabilities for blast, direct hit, and fragmentation effects. Separate blast kills for both the target body and radar antenna can be estimated. Fragmentation effects are based upon terminal effectiveness estimates generated by the full spray material lethal area (MAE) computer code (refs 1 and 2).

The ARPSIM program is coded in FORTRAN for interactive use on the CDC 6500/6600 in the INTERCOM mode. The user is prompted for data entry. Also, at the option of the user, an input guide can be generated prior to each use. Fragmentation effects are estimated from data previously generated by the MAE program relative to conditional kill probabilities. Optionally, a function, $P_k(r)$, can be provided to specify fragmentation kill probability as a function of range. Comments are added throughout the FORTRAN code for better understanding and for development of future options for the code.

A user guide, an example of a computer run, and a FORTRAN code listing are presented as appendixes A, B, and C.

PROGRAM FLOW

For each Monte Carlo sample, a simulation of the terminal characteristics of the ARP is made beginning at a time prior to fuzing during the ARP flight after final corrections to the trajectory have been made and when the remaining trajectory is linear at a fixed attack angle. The sequence of events for each simulation is:

1. An attack angle is chosen which provides a straight line flight path with respect to a specified homing point.

2. A trajectory is chosen based upon the guidance errors with respect to the homing point.

3. A fuzing point along the chosen flight path is determined.
Input Data

Describe Target

Choose Attack Angle

Select Trajectory Using Guidance Errors

Select Fuzing Point On Trajectory

Locate Burst Point

Determine \( P_k \) 's For:
- Direct Hit
- Blast
- Radar Blast
- Fragmentation

Accumulate Monte Carlo Statistics

Compute And Display Results

More Cases?

Stop

Figure 1. Program flow
4. A burst point is established based on the type of fuze, direct hit potential and possible backup fuzing or ground impact prior to nominal fuzing.

The proximity of the burst point to the target yields estimates of kill probability for direct hit, target body blast, radar blast, and fragmentation effects. The overall kill probability for each simulation is determined from the individual kill mechanism effects. This process is repeated for each simulation to provide the desired estimates of ARP terminal effectiveness. The above-described program flow is illustrated in figure 1.

The following subsections briefly describe portions of the model in the approximate order in which they follow the program flow.

Terminal Effects

Terminal effects are measured in terms of direct hit, blast, and fragmentation. Knowledge of the ARP warhead characteristics as well as the target's vulnerability to each of these effects is essential. Consequently, a preliminary analysis is required of the vulnerability of the target to the ARP warhead. Fragmentation effects are provided in either of two distinct formats: a $P_k$ grid which yields conditional kill probability as a function of burst point proximity to the target, or a $P_k$ vs $R$ (range) function which provides the kill probability data as a function of range only; i.e., azimuth characteristics are averaged for each range from projectile burst to target. These functions are provided by the MAE program. Direct hit and blast effects are estimated from standard target vulnerability analysis.

The overall kill probability for each Monte Carlo sample is based upon these individual effects and is computed as:

$$P_k = 1 - (1 - P_{DH})(1 - P_{RDR})(1 - P_{BLST})(1 - P_F)$$

where

- $P_{DH}$ = direct hit kill probability,
- $P_{RDR}$ = radar blast kill probability,
- $P_{BLST}$ = vehicle blast kill probability,
- $P_F$ = fragmentation kill probability.

Coordinate System

The simulation uses a rectangular coordinate system whose origin is at ground zero of the target center of vulnerability. Target heading establishes the negative range direction (−R); positive deflection (D) is to the left (driver's side) of the target; height (H) is measured from the ground (positive up) (fig. 2).
The attack angle is the combination of both elevation and azimuth angles which define the direction of the incoming ARP with respect to the coordinate system for the target. Azimuth is measured from the negative range direction toward the positive deflection. The elevation angle, \( \omega \), is measured from the horizontal in the positive height direction (fig. 3). Azimuth can be either fixed or chosen randomly for each simulation. Elevation is chosen from a Gaussian distribution with a specified mean, \( \mu_\omega \), and standard deviation, \( \sigma_\omega \). The attack angle orients the direction of the ARP flight path (trajectory).

Guidance Errors

Guidance errors are Gaussian and are specified by either the standard deviations in deflection and height or CEP in deflection and height. These errors are defined in the plane normal to the ARP trajectory and passing through the homing point. The location of the guidance plane and the selection of a sample trajectory through the point \((GR, GD, GH)\) are illustrated in figure 4. The determination of the point \((GR, GD, GH)\) is as follows:
First, the homing point \((GMR, GMD, GMH)\), defined in the target coordinate system \((R_t, D_t, H_t)\), is rotated through the azimuth angle, \(\lambda\).

\[
\begin{align*}
GMR' &= GMR \cos(\lambda) - GMD \sin(\lambda) \\
GMD' &= GMD \cos(\lambda) + GMR \sin(\lambda)
\end{align*}
\]

Then \(GR, GD,\) and \(GH\) are defined based on the sampled errors about the rotated homing point. Then

where \(H, D\) are random normal deviates with \(\mu = 0, \sigma = 1,\)

\[
\begin{align*}
GR &= GMR' + H \sigma_h \sin(\omega) \\
GD &= GMD' + D \sigma_d \\
GH &= GMH + H \sigma_h \cos(\omega)
\end{align*}
\]

where \(GR, GD, GH\) are in the \(R, D, H\) (projectile) coordinate system and \(\sigma_h, \sigma_d\) are the standard deviations in height and deflection, respectively, of the guidance error in the guidance plane \((H_g, D_g)\).

Fuzing

Six options are available for primary fuzing; both point detonating (PD) and proximity (VT) backup fuzes can be considered. Each of the primary fuzes is described below:

Gaussian Fuzing Angle

Fuze glitter points are specified on the target and a single glitter point is selected as either the first glitter point encountered or, optionally, chosen randomly for each simulation. When the angle between the flight path and a line from the ARP to the glitter point is equal to the fuzing angle, \(\phi\), the point on the trajectory at the vertex of the angle is taken to be the fuzing point (fig. 5). The fuze angle for each simulation is selected from a Gaussian distribution as,

\[
\phi = \mu_{\phi} + \nu \sigma_{\phi}
\]

where \(\nu\) is a random normal deviate with \(\mu = 0\) and \(\sigma = 1\).
Uniform Fuzing Angle

Identical to the Gaussian fuzing angle except that $\phi$ is chosen as uniformly random between specified limits for each simulation.

Linear Fuzing

Fuzing occurs at some distance along the ARP flight path measured from the guidance plane. The distance along the flight path is chosen from a Gaussian distribution with a specified mean, $\mu_1$ and standard deviation, $\sigma_1$ (fig. 6). Given the ARP terminal velocity, linear fuzing can be used to represent a time fuze where time is measured from the guidance plane. If $\mu_t, \sigma_t$ represent the Gaussian parameters for a time fuze, then where $V_T$ is the ARP terminal velocity, $\mu_t = V_T \mu_1$ and $\sigma_t = V_T \sigma_1$. 

Guidance Plane Intercept

ARP Trajectory

Fuzing Point

Figure 5. Fuzing angle

Guidance Plane Intercept

ARP Trajectory

Fuzing Point

Figure 6. Linear fuzing

7
Height Fuzing

Fuzing occurs at a specific height above the ground. Height is chosen from a Gaussian distribution where the mean and standard deviation are specified. The point on the ARP flight path which corresponds to the selected height is the fuzing point.

VT Fuze

A VT fuze functioning distribution is considered by specifying the cumulative distribution function of fuzing height. A fuzing height is chosen according to sampling from that distribution and the fuzing point is the point on the ARP flight path which corresponds to the selected height.

PD Fuze

The intersection of the flight path with the ground establishes the PD fuzing point.

All of the above described primary fuze options can have either a PD or VT backup fuze. The backup fuze is used if a test for primary fuze functioning fails; otherwise, the primary fuze establishes the fuzing point unless a VT backup fuzing point occurs at a greater height than the height component of the primary fuze point.

Target

The physical dimensions of the target are represented by a group (up to 5) of rectangular parallelepipeds (fig. 7) with the center of target vulnerability located over the origin of the ARP terminal coordinate system $(R_t, D_t, H_t)$.

Burst Point

In all cases, once the fuzing point is found, a check is made to ascertain whether the target has been penetrated in order to reach that fuze point. If such penetration is found, the first penetration point is taken as the warhead functioning burst point (in this case, a direct hit burst point). Since the burst point is established in the rotated coordinate system (through the azimuth component of the attack angle), prior to determining kill effects, the burst point is rotated back into the target coordinate system.
Figure 7. Target description
Direct Hit

If the burst point of the ARP is found to be at the surface of a parallel-epiped representing a face of the target, a direct hit is deemed to have occurred.

Blast

Blast kills can be estimated for both the target vehicle and radar antenna.

Target Vehicle Blast

A table of blast radius versus burst height must be provided (fig. 8). If the burst point occurs within the radius specified for the determined height of burst, then a blast kill of the target vehicle is deemed to have occurred for that sample simulation with probability, p (fig. 9 and User Guide, app B).

Radar Blast

A function of the form illustrated in figure 10 must be provided for this option. This function defines radar blast kill probability as a function of range from the antenna to the burst point. For each simulation, radar blast kill is determined from the specified function.

Fragmentation

Fragmentation effects are determined from the results of preliminary MAE analysis of the fragmenting warhead. The MAE computer code is described in references 1 and 2. The MAE program computes conditional kill probabilities as a function of burst point proximity to target center, burst height, attack elevation angle, and projectile terminal velocity. With the MAE code for a given terminal scenario for each of several burst heights, a suitable representation of the fragmentation $P_k$ function can be described. For each burst height, a $P_k$ grid is computed which provides the basis for the construction of a $P_k$ box grid about the target center. It is then a simple matter of interpolating in the range, deflection and height directions as well as for elevation angle to estimate the fragmentation $P_k$ for the actual burst point (fig. 11). Fall-off $P_k$ along the edges of the $P_k$ box is assumed to be linear out to a specified limit; that is, a limit is specified in the range, deflection, and height directions at which the fragmentation $P_k$ drops to zero.

It is important to note that the fragmentation kill probabilities generated by the MAE program are based on vulnerability data averaged over all attack azimuths. Also, $P_k$'s are determined by the MAE code by computation of the proximity
Figure 8. Blast radii vs height

Figure 9. Blast kill probability vs height

Figure 10. Radar blast function
Estimation of Fragmentation $P_k$

By Triple Interpolation

Burst Point: (BR, BD, BH)

Figure 11. Fragmentation grid interpolation
of the burst point to the target center of vulnerability point. This point-to-
point relationship is deficient for narrow spray angle munitions in close prox-
imity to the target. Also, since ARPSIM assumes a particular attack azimuth, the
assumption is made that, for the purposes of the ARPSIM model, the average vul-
nerability of the target can be used to represent the vulnerability for any par-
ticular attack azimuth.

As an alternative to the \( P_k \) grid box, the MAE program can be used to gener-
ate a \( P_k \)-versus-range function, where the \( P_k \) is averaged over all target azi-
muths. ARPSIM can utilize this function to interpolate for \( P_k \) based up on the
range from the burst point to the target center. The \( P_k \)-versus-range function
can be generated for various burst height and elevation angle combinations. This
approach is not recommended with directional warheads.

When the MAE program is used, the blast option available with the MAE code
should not be used.

MONTE CARLO ESTIMATES

The program flow procedures are followed for each simulation to provide esti-
mates for direct hit, body blast, radar blast, and fragmentation effects in
the form of kill probabilities, \( P_k \). Estimates of these kill probabilities are
computed by using

\[
P_k(n) = \frac{1}{n} \sum_{i=1}^{n} P_k(i), \quad n = \text{sample size}
\]

for each of the kill mechanisms. The combined kill probability is computed for
each sample using

\[
P_k(1) = 1 - [1 - P_{DH}(1)] [1 - P_{RDR}(1)] [1 - P_{BLST}(1)] [1 - P_F(1)]
\]

These overall kill probabilities are averaged for each individual component
kill probability to give Monte Carlo estimates of the effectiveness of the indi-
vidual kill mechanisms as well as the overall probability of defeating the tar-
get.

CONCLUSIONS

The ARPSIM model can be used to provide both weapon designers and effective-
ness analysts with an assessment of the potential for the ARP system. As a
design tool, ARPSIM provides insight into the contributions of guidance and fuz-
ing policies to the overall performance of the ARP warhead. ARPSIM does not
simulate the guidance and control or radiation sensing mechanisms. ARPSIM does
provide a means to parametrically assess the relative importance of various per-
formance levels of the guidance, fuzing, and warhead functions. By providing
effectiveness information for a host of performance capabilities, ARPSIM is a useful tool to aid in exploiting those elements of the system which provide the greatest payoff in terms of system effectiveness. ARPSIM can also be utilized to provide data for systems analyses once performance criteria for guidance, fuzing, and warhead functioning have been firmly established by weapon design.

The following specific assumptions and limitations are imbedded within the ARPSIM model:

1. Target is engaged in open flat terrain.

2. ARP terminal trajectory is linear with the longitudinal axis of the projectile collinear with the trajectory.

3. The target configuration can be adequately represented by an aggregate of rectangular parallelepipeds.

4. Fragmentation effects can be estimated with the use of either a $P_k$ box or a $P_k$-versus-range function generated by the material lethal area program based upon vulnerability data averaged over all azimuths.

RECOMMENDATIONS

The computer code follows a sequence of steps for each sample simulation. Any of these steps can be treated as a separate functional module (fig. 1). The degree of simulation detail can be changed by developing more complex modules to either increase simulation accuracy or expand modular function. The consequences of either improving the model's resolution or expanding its scope are an increase in computer processing time and a resultant increase in the cost of analysis. These consequences must be weighed against the advantages to be gained from the refinement of the model.

Some refinements which might be of merit include the direct computation of fragmentation effects (rather than use the results of precomputations with the MAE code) and the capability to define a complex target array consisting of a multiplicity of target elements.
REFERENCES


This user guide is intended to aid those who have access to the ARRADCOM CDC 6500/6600 central computing facility via INTERCOM and BATCH mode processing. Others who may wish to use or modify ARPSIM for operation on a different computer system should also find this guide informative and helpful.

For assessment of fragmentation effects with ARPSIM, it is first necessary to generate files containing fragmentation $P_k$ data as determined by the materiel MAE code (ref 1). There are two alternate forms that the MAE-produced $P_k$ data may take for use by ARPSIM:

1. A $P_k$ grid where grids are defined for the ARP terminal elevation attack angle for up to four different burst heights.

2. A $P_k$ versus range table defined for these same terminal conditions.

For directional fragmentation patterns, the $P_k$ grid format provides a better estimate of the effects produced by the non-symmetry of the warhead effects pattern. $P_k$ functions produced by the MAE code are developed as follows:

$P_k$ Grid Function

Several options exist with the MAE code described in reference 1 which allow the user to define the bounds of the $P_k$ grid in a variety of ways. It is important to note that ARPSIM is limited to a grid size of no more than 20 cells in either range or deflection directions. It is quite possible that fragmentation effects for an ARP warhead might exist at ranges far in excess of the actual miss distance from the target being attacked. For this reason, the user is advised to analyze the guidance errors and fuzing scheme being considered in order to determine practical limits to the size of the $P_k$ grid. Input data for the MAE code are often in units of feet, whereas the $P_k$ grid boundaries which are output are metric. Also, ARPSIM can be used with any consistent set of units, although it is recommended that the metric system be used. It is advisable then to pre-determine the practical range for a $P_k$ grid and then use an option with the MAE code to define the limits of the $P_k$ grid.

When using the OVRLAY code described in reference 1 to make MAE calculations, the MTRX option should be called for but not actually used; that is, the MTRX input data should consist of a blank card. For the user who is not familiar with the OVRLAY system of computer codes, it is a system that was established to provide users with the capability to make single computer runs beginning with raw fragmentation data continuing through MAE computations and the development of $P_k$ grids, and culminating with estimates of artillery system effectiveness against certain target arrays. The overlay technique is used to combine a number of computer codes devoted to these analyses. The MTRX option signals the MAE code to produce a $P_k$ grid on a file named TAPE4 in formats which are compatible with both the MTRX and ARPSIM codes. For this reason, the user should call for the MTRX option when using the OVRLAY code, and then provide only a blank card as the input for the MTRX code. By doing this, the user will normally terminate the OVRLAY code and will have defined a TAPE4 file consisting of a string of $P_k$ grids, one for each burst height. It is advisable to save the TAPE4 file as a permanent file for future recall of the data, as necessary, when using the ARPSIM code.

19
If $P_k$ grids are being generated for several (up to three) different attack elevations for use by ARPSIM, each elevation angle data set should be generated by a separate MAE run. Then, when recalling the $P_k$ grid files, define the data on file TAPE2 for the lowest angle data, TAPE3 for the next lowest, and TAPE4 for the highest. Burst heights should always be computed in the order of lowest to highest.

For users who do not have access to the MAE code or who will use an alternate code to generate $P_k$ grids, the files TAPE2, TAPE3, and TAPE4 should contain, sequentially, a card image data record in format (2I3) indicating the number of grid coordinates in range and deflection.

Next, are two card sets in format (10F7.1) where the first set defines the range coordinates of the grid boundaries and the second set defines the deflection boundaries. Boundaries are defined from lowest to highest values. Following these data sets are the $P_k$'s associated with the grid in format (10F7.5) where $P_k$'s are given first for the first range cell (lowest grid bracket) for each of the deflection cells (again, beginning with the lowest bracket) and proceeding through all range brackets in the same manner. All $P_k$ grids are defined this way for each burst height in order of lowest to highest burst height.

$P_k$ Versus Range

An average $P_k$ versus range function (table) can be used if the number of ranges is no greater than 200. Format for data entry is (F8.3, F8.5) where the first item is range (usually in meters) followed by the corresponding average $P_k$. The MAE code can generate this table on a file named TAPE15. These files can be saved, like the grid files, and recalled when using the ARPSIM. These files when used other than with the MAE code or when recalling MAE-generated files, are defined like the grid files, i.e., lowest angle data on TAPE2, next lowest on TAPE3, and highest on TAPE4. Each burst height (up to four) has its own table defined beginning with the lowest burst height and stored sequentially on each file.

Following definition of the $P_k$ functions on TAPE2, TAPE3, and TAPE4 (as required), the ARPSIM can be exercised using a teletype (TTY). Preliminary steps required to run ARPSIM on the ARRADCOM computer in INTERCOM mode are as follows:

**INTERCOM Mode Setup**

The following sequence is required to access the ARPSIM code and begin its execution:

```
LOGIN.
...follow normal login procedures
COMMAND - ETL,500.
COMMAND - FETCH,ARP,BWEBSTER.
COMMAND - ATTACH,T,TAPE1FILE,ID=your id.
COMMAND - COPYBF,T,TAPE1.
COMMAND - RETURN,T.
```
The sequence from ATTACH,T... through RETURN,T is only required if a previously defined set of basic inputs is to be used as a basis for this run. Also, the sequence ATTACH,TAPE2,... through ATTACH,TAPE4,... is required only in accordance with the requirements to estimate fragmentation effects and the diversity of attack elevations required.

In response to the command ARP, the user will be given the opportunity to produce a summary input guide. Following that, the user will be asked whether a file named TAPE1 is to be used as the basis for input data. This option is provided as an aid to the user who expects to make several computer runs with the model using the same basic input data set. The ARPSIM code has a built-in input editing routine which continually redefines the file TAPE1 to be the current basic input data set. The user who wishes to make additional runs with a basic data set merely has to define the current data set and then, after ARPSIM has been run, the TAPE1 file is stored on a permanent file for later use as with the ATTACH,T... through RETURN,T sequence described above. If a basic data set is being used, then the initial input conditions are listed. Then, in all cases, the user is asked to ENTER DATA OR END. In response to this command the user begins to enter "word" type data to either initialize a data type or change a data type. Word type data which can be entered are defined according to general function in the section which follows. Formats are (A4,F10.4).

"Word" Type Data

This section is divided into functional areas as follows:

Guidance Data

NGER,n. NGER signifies the number of guidance error data sets to be input. The value of n equals the number of different guidance error sets to be analyzed.

NCEP,l. If guidance errors are input as standard deviations in both deflection and height, omit this set. If errors are input as CEP, then include this set. Note that in all cases errors are defined in a plane passing through a homing point and normal to the ARP flight path.

Fuzing Scheme

FZAM,n. FZAM signifies the use of the fuzing angle primary fuze where n is the mean value of the fuze half-vertex angle; i.e., n is
the mean angle from the ARP trajectory to the fuzing glitter point at which fuzing will occur. Units are degrees.

FZAS,n. FZAS signifies the standard deviation of fuzing angle associated with the mean value defined by FZAM, where the value n is the standard deviation. Units are degrees.

FZTM,n. FZTM signifies the use of a linear (or time) fuze where the sign of the value of n indicates whether the fuze operates in the vertical direction or along the trajectory. A negative n signifies the vertical option. The value of n is the mean distance from the guidance plane (or initial fuzing point if used in conjunction with the FZAM option) in the negative range direction where fuzing occurs. With the vertical option, the distance is measured from the ground. A time fuze operating along the ARP trajectory can be simulated by converting the values to distances by using the known AR1 terminal velocity.

FZTS,n. FZTS defines the standard deviation associated with the FZTM data in all modes.

PKPF,n. The value of n is the probability that the primary fuze (options described above) will function.

PDVT,n. Selects the backup fuze option. The value for n is 0 for a PD (ground burst) backup and is the number of entries in a height versus probability table (up to 5 values) to define the VT fuze functioning distribution.

GLTR,n. Specifies the glitter points used by the angular fuzing function option. If n is 0, the fuze functions relative to the point (0,0,TGTC) where TGTC is the center of target vulnerability. If n is non-zero, the fuze functions relative to one of the n input glitter points. A positive n signifies that the fuzing glitter point is selected randomly; a negative n signifies that the first glitter point encountered will cause fuzing.

Terminal Conditions

OMEG,n. The elevation angle measured from the ground is chosen from a normal distribution with mean value n.

OMGS,n. The standard deviation associated with OMEG is input as n.

TGTC,n. The center of target vulnerability is input as a height above the origin at (0,0,n). If direct hit effects are not being analyzed (direct hit boxes are not defined), then the vehicle blast effects are determined based on the range from the burst point to (0,0,TGTC).
DHAZ, n. The azimuth angle-of-attack is \( n \) and is measured from the negative range axis in the direction of the positive deflection axis. Units are degrees. To choose the azimuth uniformly random between 0 and 360 degrees, set \( n = -1 \).

DUDR, n. The dud rate of ARP projectiles is given as \( n \), where a 5\% dud rate corresponds to \( n = 0.05 \).

**General Conditions**

SAMP, n. The number of Monte Carlo samples is \( n \).

PRNT, 1. Specifies that only a final summary of results is to be output.

SRNG, n. Tables of average combined \( P_k \) can be output as a function of azimuth, elevation and range as well as averaged over non-zero results obtained in the angular bins for each range. The value for \( n \) is the upper limit (defaults to 100) for range information. The range scale is logarithmic and includes 10 bins, beginning with the minimum range obtainable (considering direct hit implications) and ending at \( n \).

**Fragmentation Effects**

PKNH, n. Specifies the number of heights, \( n \), at which fragmentation effects are provided (either as \( P_k \) grids or \( P_k \) versus range tables). Must not exceed 4.

PKNA, n. Specifies the number of elevations, \( n \), for which fragmentation effects are provided. Must not exceed 3.

For \( n = 1 \), effects are on TAPE2.
For \( n = 2 \), effects are on TAPE2 for lowest angle data and on TAPE3 for highest angle data.
For \( n = 3 \), effects are on TAPE2 for lowest angle data, TAPE3 for middle angle data, and TAPE4 for highest angle data.

FUNC, 1. Selects option to use \( P_k \) versus range tables for fragmentation effects in place of the \( P_k \) grids.

**Direct Hit Effects**

DHIT, n. Specifies the number of target boxes to be input to approximate the shape of the target for purposes of computing direct hit effects. Boxes are defined relative to (0,0,0) and the total number of boxes cannot exceed 5.
PKDH, n. Direct hit $P_k$ if a direct hit is achieved. If $n = 0$, $P_k$ is defaulted to one.

**Blast Effects**

PKBL, n. Specifies the blast $P_k$ if the burst point is within a range specified by the BLST data of the surface of any direct hit box. If direct hit boxes are not used, then range is calculated to the point $(0, 0, \text{TGTC})$.

BLST, n. Specifies the range from the direct hit surfaces or the point $(0, 0, \text{TGTC})$ within which the blast $P_k$ against the vehicle body is that given by the PKBL data. To enter a table of blast ranges versus burst height, enter a negative value for $n$ which corresponds to the number of entries in the blast range versus height table (may not exceed 5).

RADR, l. Include to compute blast effects against radar antenna separately from vehicle blast.

**End of Word Data**

**END** Must always be included at the end of the "word"-type data entries.

After all "word"-type data have been entered, the code will ask for certain data which are required by some of the options chosen by the "word" cards. These additional input requirements are discussed in the following section. All data are free-formatted.

**Guidance Data**

Either pairs of deflection and height standard deviations are entered or, if NCEP, I. data is entered in the "word" section, then the guidance errors are input as CEP's.

The homing point coordinates follow the guidance error inputs. The homing point is generally the coordinates of the center of the radar antenna.

**Direct Hit Boxes**

The limits of the dimensions of each direct hit box are input for range, deflection, and height, respectively. For example, for a direct hit box centered at the origin and having a length of 20 meters, a width of 10 meters, and a height of 5 meters, this data would be input as $-10, 10, -5, 5, 0, 5$. 

24
Radar Data

Radar antenna coordinates are entered for the purposes of radar blast $P_k$ computation.

Following the entry of the radar coordinates, values are entered for two ranges, $R_1$ and $R_2$, which define the radar blast $P_k$ function as being one out to $R_1$ and declining linearly to zero at $R_2$.

Fragmentation

Heights are entered beginning with the lowest value and corresponding to the burst heights used for the MAE computations. An additional height is input last and corresponds to that height at which all fragmentation $P_k$'s are zero.

Following the height data, two values are input corresponding to the distances beyond the edge of the $P_k$ grids where the fragmentation $P_k$ becomes zero in range and deflection, respectively.

Elevation angles are entered next, beginning with the lowest angle and corresponding to the angles for which the MAE code was run to produce the fragmentation $P_k$ data.

VT Backup Fuzing

A table of probability of fuze functioning at height less than or equal to height, $H$, is used to generate VT fuzing data. Up to five heights are input followed by probabilities corresponding to the probability of fuze functioning between the respective height and the next lower height. Ideally, probability values should sum to unity.

Glitter Points

Glitter point coordinates are entered for each glitter point. All coordinates are relative to (0,0,0) of the target.

Blast Data (Vehicle)

If the blast-distance-versus-burst-height option is chosen (negative n on BLST,n data), then n pairs of blast distance, height are entered.

This concludes the input requirements for using the ARPSIM model. Word type data can be changed or input in any order. Required additional data will be
prompted from the user by the code. The user is always given the option of list-
ing the current data set (with the exception of the fragmentation $P_k$ data) or
changing the data set prior to actual computations. When the computations are
completed for all cases, the user is given the opportunity to run additional
cases based on the current data sets.
APPENDIX B

EXAMPLE
The following example, provided as a supplement to the User Guide in Appendix A, denotes the type of material generated for a typical ARPSIM run:

```
* ANTI-RADIATION SIMULATION PROGRAM - 9/1/80 *

NOTE: ALL COORDINATES ARE DEFINED RELATIVE TO ORIGIN AT GROUND ZERO OF TARGET.
COORDINATE SYSTEM IS RECTANGULAR.
TARGET HEADING IS NEGATIVE RANGE.
DRIVER SIDE (L) IS POSITIVE DEFLECTION.
HEIGHT IS MEASURED FROM GROUND.

DATE - 09/27/80
TIME - 13.47.13.

DO YOU WANT A LISTING OF CODE NAMES? Y

* OMEG - MEAN ATTACK ANGLE
* OMEG - ATTACK ANGLE STD DEVIATION
* NOTE: FOLLOWING GUIDANCE ERROR PARAMETERS
* (SIGM,SIGH) ARE MEASURED IN PLANE NORMAL TO TRAJECTORY AND PASSING THROUGH HOMING POINT
* NGER - NUMBER OF GUIDANCE ERRORS TO CONSIDER
* ENTER HOMING POINT (R,Z,H), GUIDANCE ERRORS ARE DISTRIBUTED ABOUT HOMING PT.
* NCEP = 1., IF CEP IS INPUT FOR GUIDANCE ERROR SIGMAS
* FANC = 1., IF OPTION TO USE PK VS RANGE DAMAGE IN PLACE OF PK BOX FUNCTION IS SELECTED
* YOU MUST DEFINE PK VS R DATA FOR EACH HEIGHT LAYER SPECIFIED BY PKNH CARD AND EACH ANGLE SPECIFIED BY PKNA CARD
* FZAM,FZAS,FZTM,FZTS - FUZING ERROR OPTIONS
* FZAM - MEAN ANGLE AT WHICH FUZING OCCURS ON INTERCEPT
* FZAS - STD DEVIATION ASSOCIATED WITH FZAM
* NOTE: FOR UNIFORM FUZING ANGLE BETWEEN FZAM AND FZAS, ENTER A NEGATIVE VALUE FOR FZAM
* FUZE ANGLE WILL BE CHOSEN UNIFORMLY RANDOM BETWEEN POSITIVE FZAM AND FZAS
* NOTE: FUZING PLANE PASSES THROUGH FUZING GLITTER
* POINT NORMAL TO SAMPLE TRAJECTORY
* FZTM - MEAN DISTANCE FROM GUIDANCE PLANE AT WHICH FUZING WILL OCCUR ALONG TRAJECTORY
* NOTE: ENTER A NEGATIVE FZTM FOR HEIGHT FUZING
* WITH MEAN HEIGHT AT FZTM
```
FZTR - STD DEVIATION ASSOCIATED WITH FZTR
SAMP - SAMPLE SIZE
PKNH - NUMBER OF HEIGHTS AT WHICH FRAGMENTATION
      PK DATA WILL BE DEFINED
NOTE: PKNH < 5
PKNA - NUMBER OF ELEVATION ANGLES FRAGMENTATION
      PK DATA WILL BE DEFINED FOR
NOTE: PKNA < 4
PKPF - PROBABILITY OF PRIMARY FUZE FUNCTIONING
PDUT = 0, FOR PD BACKUP; NOT FOR UT BACKUP FUZE
WHERE HUT = NUMBER OF UT BURST HEIGHTS
GLTR = 0, IF PRIMARY FUZE FUNCTIONS RELATIVE TO
      CENTER OF TARGET, HULT IF PRIMARY FUZE
      FUNCTIONS RELATIVE TO ANY ONE OF HULT
      EQUALLY LIKELY GLITTER POINTS
S'NOTEI PKNN (PKA - NUMBER OF ELEVATION ANGLES
      FRAGMENTATION
PKDATA UILL
PKHA (PKA - PRIMARY FUZE FUNCTIONING
PUMP - . FOR PD BACKUP, NOT FOR UT BACKUP FUZE

* DBUG - TO PRINT SUMMARY ONLY, 0, OTHERWISE
* EBUG - 0, TO PRINTOUT PROGRAM DEBUGGING DATA
       DBUG = 1, GUIDANCE & FUZING DATA
       DBUG = 2, DIRECT HIT PENETRATION DATA
       DBUG = 3, HOMING ANGLE DATA
       DBUG = 4, PK BOX DATA
       DBUG = 5, PK GRIDS
       DBUG = 6, PK US A DATA
*TGTC - HEIGHT OF TARGET CENTER ABOVE GROUND
DUSR - DUSR RATE OF PROJECTILE, EXPRESSED AS A FRACTION
DHTT - DIRECT HIT OPTION, NUMBER OF TARGET BOXES
       IF DHTT IS OMITTED AND BUST IS INCLUDED,
       BUST IS RADIUS 'FROM (0,0,TGTC) WITHIN
       WHICH BUST IS IN'I
PKHN - DIRECT HIT PK (BUST PK = 1)
HUTO - 1, DEFINE FUNC FOR BUST KILL OF RADAR ONLY
       AND READ IN RADAR ANTENNA COORDINATES.
       TO DEFINE FUNC, SPECIFY R1 AND R2.
       WHERE BUST PK IS 1 OUT TO R1 AND
       DECLINES LINEARLY TO 0 AT R2.
DHAR - AZIMUTH ANGLE OF ATTACK UP FRONT OF TARGET
       TOWARD DRIVER SIDE, SET TO -11. FOR RADAR
BLST = BLAST RADIUS WITHIN WHICH VEHICLE PK=PBLT
       NOTE: TO ENTER BLAST RADIUS, BUST HEIGHT.
       ENTER NEGATIVE NUMBER OF BLAST,HOT HOURS
       IN PLACE OF VALUE OF BLST. PKHS OF
       BUST HOURS ARE ENTERED IN ASCENDING ORDER
       OF HEIGHT.
COORDINATE SYSTEM IS RECTANGULAR.
TARGET HEADING IS NEGATIVE RANGE.
DRIVER SIDE (LEFT) IS POSITIVE DEFLECTION.
HEIGHT IS MEASURED FROM GROUND.
ENTER DATA BY ENTERING CODE NAME
FOLLOWED BY A COMMA AND THE VALUE IN FLOATING
POINT FORMAT. TO END DATA ENTRY, ENTER
THE WORD END IN COLUMNS 1-3.
DO YOU WISH TO INITIALIZE DATA FROM SAVED?
DATA FILE (TAPE1) ?

INITIAL INPUTS
FZAM 78.000
PKNL 1.000
PKBL 1.000
FZAS 16.000
OMCL 18.000
NGER 3.000
MECP 1.000
FUNC 1.000
DHTT 2.000
SAMP 180.000
PKRN 4.000
PKNA 3.000
PDLT 5.000
PKPF 0.950
GLTR 3.000
SRNG 180.000
TGTC 10.000
DUSR 3.000
BLST 3.000
END
### Radar Data

**Enter Radar Antenna Coordinates (R, D, H) Relative to Target Ground Zero.**

- \( R \): \(-10\) to \(+10\)
- \( D \): \(-10\) to \(+10\)

Enter \( R_1, R_2 \), where Radar BLAST PK=1
diplomatic to \( R_1 \) and declines linearly to zero at \( R_2 = -10, 0, +10\).

**Do you want current data listed?** 'Y

**Do you want to change any data?** 'N

---

### Final Results

- **PK = 0.7814**
- **PKED = 0.0350**
- **HSLAP = 100**

---

**Do you want PK US R, ALPHA, BETA?** 'Y

<table>
<thead>
<tr>
<th>PK</th>
<th>( E )</th>
<th>ALPHA</th>
<th>BETA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0000</td>
<td>11.0</td>
<td>66.0 – 98.0</td>
<td>66.0 – 75.0</td>
</tr>
<tr>
<td>1.0000</td>
<td>11.8</td>
<td>120.0 – 150.0</td>
<td>15.0 – 36.0</td>
</tr>
<tr>
<td>1.0000</td>
<td>11.8</td>
<td>150.0 – 180.0</td>
<td>45.0 – 60.0</td>
</tr>
<tr>
<td>1.0000</td>
<td>11.8</td>
<td>180.0 – 200.0</td>
<td>60.0 – 75.0</td>
</tr>
<tr>
<td>1.0000</td>
<td>11.8</td>
<td>210.0 – 240.0</td>
<td>75.0 – 90.0</td>
</tr>
<tr>
<td>1.0000</td>
<td>11.8</td>
<td>240.0 – 270.0</td>
<td>90.0 – 100.0</td>
</tr>
<tr>
<td>1.0000</td>
<td>11.8</td>
<td>270.0 – 300.0</td>
<td>100.0 – 120.0</td>
</tr>
</tbody>
</table>

---

31
AVG PK VS. A,

S'r FINXK RESULTS
PIC .6976 PKSD .04169 NSAMP .6100

'YOU WANT PK VS R, ALPHA, BETA? *V'

<table>
<thead>
<tr>
<th>PK</th>
<th>R</th>
<th>ALPHA</th>
<th>BETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0000</td>
<td>11.8</td>
<td>128.0 - 150.0</td>
<td>15.0 - 150.0</td>
</tr>
<tr>
<td>1.0000</td>
<td>11.8</td>
<td>128.0 - 150.0</td>
<td>15.0 - 150.0</td>
</tr>
<tr>
<td>1.0000</td>
<td>11.8</td>
<td>128.0 - 150.0</td>
<td>15.0 - 150.0</td>
</tr>
<tr>
<td>1.0000</td>
<td>11.8</td>
<td>128.0 - 150.0</td>
<td>15.0 - 150.0</td>
</tr>
<tr>
<td>1.0000</td>
<td>11.8</td>
<td>128.0 - 150.0</td>
<td>15.0 - 150.0</td>
</tr>
</tbody>
</table>

32
### AUG وك US. N* 

<table>
<thead>
<tr>
<th>Alpha</th>
<th>Beta</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0000</td>
<td>11.8</td>
</tr>
<tr>
<td>1.0000</td>
<td>11.8</td>
</tr>
<tr>
<td>1.0000</td>
<td>11.8</td>
</tr>
<tr>
<td>1.0000</td>
<td>11.8</td>
</tr>
<tr>
<td>1.0000</td>
<td>11.8</td>
</tr>
<tr>
<td>1.0000</td>
<td>11.8</td>
</tr>
<tr>
<td>1.0000</td>
<td>11.8</td>
</tr>
<tr>
<td>1.0000</td>
<td>11.8</td>
</tr>
<tr>
<td>1.0000</td>
<td>11.8</td>
</tr>
<tr>
<td>1.0000</td>
<td>11.8</td>
</tr>
</tbody>
</table>

### FINAL RESULTS

PK = 0.6083 PK5D = 0.007 USAMP = 100

### PHABULOUS PK US K, ALPHABET* Y

<table>
<thead>
<tr>
<th>PK</th>
<th>R</th>
<th>Alpha</th>
<th>Beta</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0000</td>
<td>11.8</td>
<td>120.0 - 150.0</td>
<td>150.0 - 200.0</td>
</tr>
<tr>
<td>1.0000</td>
<td>11.8</td>
<td>120.0 - 150.0</td>
<td>150.0 - 200.0</td>
</tr>
<tr>
<td>1.0000</td>
<td>11.8</td>
<td>120.0 - 150.0</td>
<td>150.0 - 200.0</td>
</tr>
<tr>
<td>1.0000</td>
<td>11.8</td>
<td>120.0 - 150.0</td>
<td>150.0 - 200.0</td>
</tr>
<tr>
<td>1.0000</td>
<td>11.8</td>
<td>120.0 - 150.0</td>
<td>150.0 - 200.0</td>
</tr>
<tr>
<td>1.0000</td>
<td>11.8</td>
<td>120.0 - 150.0</td>
<td>150.0 - 200.0</td>
</tr>
<tr>
<td>1.0000</td>
<td>11.8</td>
<td>120.0 - 150.0</td>
<td>150.0 - 200.0</td>
</tr>
<tr>
<td>1.0000</td>
<td>11.8</td>
<td>120.0 - 150.0</td>
<td>150.0 - 200.0</td>
</tr>
<tr>
<td>1.0000</td>
<td>11.8</td>
<td>120.0 - 150.0</td>
<td>150.0 - 200.0</td>
</tr>
<tr>
<td>1.0000</td>
<td>11.8</td>
<td>120.0 - 150.0</td>
<td>150.0 - 200.0</td>
</tr>
</tbody>
</table>

33
RESULTS FOR FOLLOWING CONDITIONS -

<table>
<thead>
<tr>
<th>ITEM</th>
<th>MEAN</th>
<th>STD DEV</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELEVATION</td>
<td>10.000</td>
<td>0.000</td>
</tr>
<tr>
<td>FUZE ANGLE</td>
<td>70.000</td>
<td>10.000</td>
</tr>
<tr>
<td>LINEAR FUZE</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>AZIMUTH</td>
<td>0.000</td>
<td></td>
</tr>
</tbody>
</table>

HOMING POINT COORDINATES (R,D,H) = 0.0, 0.0, 10.0

ERROR DATA

<table>
<thead>
<tr>
<th>CEP</th>
<th>PK</th>
<th>PKFRA</th>
<th>PKFRA2</th>
<th>PKFRA3</th>
<th>PKFRA4</th>
<th>PKFRA5</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.0</td>
<td>.7516</td>
<td>.4173</td>
<td>.2403</td>
<td>.3798</td>
<td>.5400</td>
<td></td>
</tr>
<tr>
<td>6.0</td>
<td>.6573</td>
<td>.5097</td>
<td>.5517</td>
<td>.5800</td>
<td>.4000</td>
<td></td>
</tr>
<tr>
<td>9.0</td>
<td>.6583</td>
<td>.2613</td>
<td>.5725</td>
<td>.6100</td>
<td>.3200</td>
<td></td>
</tr>
</tbody>
</table>

'DO YOU WISH TO RUN ANOTHER CASE? *Y
A description of the material produced by this particular ARPSIM run follows:

Header information is printed, including the time and date of the run. The user is asked whether a listing of input code names is desired (as an aid to generating a proper set of inputs). In this example, the code names are printed. Next, the user is given the option of starting with a previously developed set of inputs which can be changed by a built-in input editing routine. That option is invoked for this example. Note that a file named TAPE1 must be defined which contains this data prior to running ARPSIM. A listing of initial data conditions is provided next. The user is then asked whether any data changes are required.

In this example the user desires to add the capability to estimate radar blast effects. Note that only changed data need be entered at this point. The code then asks for additional information required by the added data. Having fulfilled the data requirements, the user is given the option of listing the entire data set again. Following this, the user is given the option of making any additional changes or corrections to the data set. In this example no additional changes are requested.

Before proceeding with the discussion of the ARPSIM results for this case, a brief run-through is given of the input data set. The FZAM data specifies a fuze angle option with a mean value of 70 degrees for the fuze angle. The FZAS code specifies a 10-degree standard deviation for the fuze angle from simulation to simulation. The PKDH and PKBL data indicate direct hit and vehicle blast Pk's, respectively. Attack elevation of 10 degrees is specified by the OMEG card. NGER indicates three different sets of guidance errors will be analyzed, and NCEP indicates that guidance errors will be input as CEP. FUNC specifies that the fragmentation Pk's will be estimated from interpolations in a set of Pk versus range tables generated by the MAE code for a combination of burst height and elevation angles.

Up to three elevation angle sets can be provided on files TAPE2, TAPE3, and TAPE4. If only a single elevation angle data set is provided, then only TAPE2 is required. Two elevation angles require both TAPE2 and TAPE3. Each file contains Pk versus range for identical burst heights, beginning with the lowest burst height. That is, if four burst heights have been analyzed by the MAE code (the maximum allowable by ARPSIM), each file will contain four Pk versus range tables, one for each burst height beginning with the lowest height and progressing to the highest.

In this example, four burst heights were considered for each of three angles of fall (elevation angles) as specified by the PKNH and PKNA codes, respectively. SAMP provides the number of simulations to run for each case. PDVT specifies that a VT backup fuze is being considered where the height of burst distribution for the backup fuze will be typified at five burst heights. PKPF specifies that the probability that the primary fuze functions is 0.95. GLTR specifies that three glitter points for primary fuzing exist. SRNG gives the maximum range for a Pk versus range table to be generated based upon the results of the ARPSIM run. TGTC provides that the center of target vulnerability is located at 10 (in this case meters) above the target origin (0,0,0). DUDR specifies a projectile dud rate of 5%. BLST provides a blast radius from the TGTC point within which the Pk for vehicle blast effect is as stated on the PKBL data above.
The END code signifies the end of the word type data. The numbers 3, 6, and 9 specify the guidance error CEP's. Following this are the homing point coordinates (0,0,10), and the limits in range, deflection and height of the two direct hit target description boxes. Burst heights and angles of fall (elevations) utilized by the MAE code in generating the P_k versus range tables are specified next. Then the heights and probabilities associated with the backup fuzing function are listed. Finally, glitter point coordinates are specified.

Final results are given as the combined kill probability, the standard deviation of kill probability and the sample size upon which these numbers are based. The user is given the option of listing the generated hemispheric distribution of computed combined P_k's, where the angle alpha denotes azimuth and beta denotes elevation from the burst point to (0,0,0). The range specified is also the range from the burst point to the origin (0,0,0). These hemispheric data (only the positive elevation angles are considered since negative angles would imply a burst below ground) are averaged over all angular bins for which burst points were analyzed to provide a table of average P_k versus range.

The final results are repeated for each case and followed by a summary of the results for each type P_k considered together with the corresponding error data for that case.

After all results have been given for all cases specified, the user is given the opportunity to run additional cases, based upon the same data set. In all cases, the contents of the file TAPE1 are always the last data set considered. Consequently, if the user wishes to make additional runs with ARPSIM at a later time using the same basic data set, then after the current runs with ARPSIM are finished, the file TAPE1 can be saved as a starting point for future runs.

TAPE1 can be retained as a permanent file. However, for access at a later date, this TAPE1 must be attached with a different local file name. Then this local file name is copied to a new file named TAPE1. These steps are necessary because the ARPSIM code changes the contents of the file TAPE1.
APPENDIX C

FORTRAN LISTING

Note: The following FORTRAN listing is subject to changes as dictated by improvements or modifications to the ARPSIM model.
C

PROGRAM ARP 73/74 OPT=1 FTN 4.6+S08 03/13/81 08.28.23 PAGE 1

1

PROGRAM ARP (INPUT=220, OUTPUT=220, TAPE6=INPUT, TAPE6=OUTPUT,
CTAPE1=220, TAPE2=220, TAPE3=220, TAPE4=220, TAPE8=220)
005100

000110

000120

000130

000140

000150

000160

000170

000180

000190

000200

000210

000220

000230

000240

000250

000260

000270

000280

000290

000300

000310

000320

000330

000340

000350

000360

000370

000380

000390

000400

000410

000420

000430

000440

000450

000460

000470

000480

000490

000500

000510

000520

000530

000540

000550

000560

000570

000580

000590

000600

000610

000620

000630

000640

000650

000660
WRITE (6,*) "NOTE: FOLLOWING GUIDANCE ERROR PARAMETERS*" 000674C
WRITE (6,*) "(SIGD, SIGH) ARE MEASURED*" 000580H
WRITE (6,*) "IN PLANE NORMAL TO TRAJECTORY AND*" 000690C
WRITE (6,*) "PASSING THROUGH HOMING POINT*" 0007000
WRITE (6,*) "NGER - NUMBER OF GUIDANCE ERRORS TO CONSIDER*" 0007100
WRITE (6,*) "ENTER HOMING POINT (R,0,H). GUIDANCE*" 000720C
WRITE (6,*) "ARE DISTRIBUTED ABOUT HOMING PT.*" 0007500
WRITE (6,*) "NCEP - 1., IF CEP IS INPUT FOR GUIDANCE ERROR SIGMAs*" 000740C
WRITE (6,*) "FZAM,FZAS,FZTR,FZTS - FUZING ERROR OPTIONS*" 0007500
WRITE (6,*) "FZAM - MEAN ANGLE AT WHICH FUZING OCCURS ON*" 000760C
WRITE (6,*) "INTERCEPT*" 0007700
WRITE (6,*) "FZAS - STD DEV ASSOCIATED WITH FZAM*" 000780C
WRITE (6,*) "NOTE: FUZE ANGLE IS CONSTRAINED TO (0,P1)*" 000790C
WRITE (6,*) "FOR UNIFORM FUZING ANGLE BETWEEN FZAM*" 0008000
WRITE (6,*) "AND FZAS, ENTER A NEGATIVE VALUE FOR FZAM*" 000810C
WRITE (6,*) "FUZE ANGLE WILL BE CHOSEN UNIFORMLY RANDOM*" 000820C
WRITE (6,*) "BETWEEN POSITIVE FZAM AND FZAS*" 000830C
WRITE (6,*) "FOR TIME-TO-GO FUZE, ENTER NEGATIVE FZAS.*" 000840C
WRITE (6,*) "NOTE: FUZING PLANE PASSES THROUGH FUZING GLITTER*" 000850C
WRITE (6,*) "POINT NORMAL TO SAMPLE TRAJECTORY*" 000860C
WRITE (6,*) "FZTM - MEAN DISTANCE FROM GUIDANCE PLANE AT WHICH*" 000870C
WRITE (6,*) "FUZING WILL OCCUR ALONG TRAJECTORY*" 000880C
WRITE (6,*) "NOTE: ENTER A NEGATIVE FZTM FOR HEIGHT FUZING*" 000890C
WRITE (6,*) "WITH MEAN HEIGHT ABS(FZTM) *" 000900C
WRITE (6,*) "FZTS - STD DEV ASSOCIATED WITH FZTM*" 0009100
WRITE (6,*) "SAMPLE SIZE*" 000920C
WRITE (6,*) "PKH - NUMBER OF HEIGHTS AT WHICH FRAGMENTATION*" 000930C
WRITE (6,*) "PK DATA WILL BE DEFINED" 0009400
WRITE (6,*) "NOTE: PKH < 9" 0009500
WRITE (6,*) "PKPE - PROBABILITY OF PRIMARY FUZE FUNCTIONING*" 0009600
WRITE (6,*) "WHERE PK = PK BACKUP, NVT = FOR VT BACKUP FUZE*" 000970C
WRITE (6,*) "WHERE NVT = NUMBER OF VT BURST HEIGHTS*" 000980C
WRITE (6,*) "GLTR - 0. IF PRIMARY FUZE FUNCTIONS RELATIVE TO*" 000990C
WRITE (6,*) "CENTER OF TARGET, NGLT IF PRIMARY FUZE*" 001000C
WRITE (6,*) "FUNCTIONS RELATIVE TO ANY ONE OF NGLT*" 001010C
WRITE (6,*) "EQUALLY LIKELY GLITTER POINTS ,*" 001020C
WRITE (6,*) "SET NGLT NEGATIVE TO PICK FIRST*" 001030C
WRITE (6,*) "POINT ENCOUNTERED,*" 001040C
WRITE (6,*) "SRANG - MAXIMUM RANGE FOR COMPUTING PK VS RANGE*" 001050C
WRITE (6,*) "FRNT - 1. TO PRINT SUMMARY ONLY, 0. OTHERWISE*" 001060C
WRITE (6,*) "TO PRINTOUT PROGRAM DEBUGGING DATA*" 001070C
WRITE (6,*) "DBUG = 1, GUIDANCE & FUZING DATA*" 001080C
WRITE (6,*) "DBUG = 2, DIRECT HIT PENETRATION DATA*" 001090C
WRITE (6,*) "DBUG = 4, PK BOX DATA*" 001100C
WRITE (6,*) "DBUG = 5, PK GRIDS*" 001110C
WRITE (6,*) "DBUG = 6, PK VS R DATA*" 001120C
WRITE (6,*) "IGTC - HEIGHT OF TARGET CENTER ABOVE GROUND*" 001130C
WRITE (6,*) "DUSB - DUO RATE OF PROJECTILE, EXPRESSED AS A FRACTION*" 001140C
WRITE (6,*) "CHIT - DIRECT HIT OPTION, NUMBER OF TARGET BOXES*" 001150C
WRITE (6,*) IF CHIT IS CHITTED AND BLST IS INCLUDED,*" 001160C
WRITE (6,*) "BLST IS RADIUS FROM (0,0,IGTC) WITHIN*" 0011700
WRITE (6,*) "WHICH PKBLST = 1,*" 001180C
WRITE (6,*) "PKBLST = DIRECT HIT PK (0. = 1.)*" 001190C
WRITE (6,*) "PKBL = BLAST PK (0. = 1.)*" 001200C
WRITE (6,*) "RAEST 1., DEFINE FUNC FOR BLAST KILL OF RADAR ONLY*" 001210C
WRITE (6,*) "AND READ IN RADAR ANTENNA COORDINATES.*" 001220C
WRITE (6,*) "0.10
115 WRITE (6,*) " TO DEFINE FUNC, SPECIFY R1 AND R2, " 001240
WRITE (6,*) " WHERE BLAST PK IS 1 OUT TO R1 AND R2, " 001250
WRITE (6,*) " DECREASE LINEARLY TO 0 AT R2, " 001260
WRITE (6,*) " " 001270
WRITE (6,*) " " 001280
WRITE (6,*) " " 001290
WRITE (6,*) " " 001300
WRITE (6,*) " BLAST - AZIMUTH ANGLE OF ATTACK GFF FRONT OF TARGET " 001310
WRITE (6,*) " TOWARD DRIVER SIDE. SET TO " -1. FOR RANDOM " 001320
WRITE (6,*) " " 001330
WRITE (6,*) " " 001340
WRITE (6,*) " " 001350
WRITE (6,*) " " 001360
WRITE (6,*) " " 001370
WRITE (6,*) " " 001380
WRITE (6,*) " " 001390
54 NBRIT = 0 001400
ISIT = 0 001410
IITNE = 0 001420
CALL RDMDU(INIT) 001430
CALL RDMDU(INIT) 001440
IF(IDR.EQ.5) GO TO 88 001450
IF(NPRR.EQ.0) GO TO 80 001460
WRITE (6,*) " ENTER DATA BY ENTERING CODE NAME " 001470
WRITE (6,*) " " 001480
WRITE (6,*) " " 001490
WRITE (6,*) " " 001500
WRITE (6,*) " " 001510
WRITE (6,*) " " 001520
WRITE (6,*) " " 001530
WRITE (6,*) " " 001540
WRITE (6,*) " " 001550
WRITE (6,*) " " 001560
WRITE (6,*) " " 001570
READ (5,1001) IDR 001580
IF(ANS.EQ.5) IDR = 1 001590
80 REMIND 1 001600
REMIN 2 001610
REMIN 3 001620
REMIN 4 001630
P1 = ATAN2(0.,-1.) 001640
DO 51 I = 1,10 001650
DO 51 ID = 1,10 001660
DO 51 ID = 1,10 001670
DO 51 ID = 1,10 001680
DO 51 ID = 1,10 001690
DO 51 ID = 1,10 001700
DO 51 ID = 1,10 001710
DO 51 ID = 1,10 001720
DO 51 ID = 1,10 001730
DO 51 ID = 1,10 001740
DO 51 ID = 1,10 001750
DO 51 ID = 1,10 001760
DO 51 ID = 1,10 001770
DO 51 ID = 1,10 001780
DO 51 ID = 1,10 001790
DO 51 ID = 1,10 001800
PROGRAM ARP 73/74  OPT=1

53 CONTINUE
WRITE (6,2000) AAAA
GO TO 7

14 CALL READ (DATA,INEW,ANAM,IPD,1,RDH,D0H,HDH)

C
SET UP TAPE

C
9 REWIND 1
DO 81 I=1,50
IF(DATA(I).EQ.0) GO TO 81
WRITE (1,1000) ANAM(I),DATA(I)

81 CONTINUE
WRITE (1,1000) END
CALL WRITE (DATA,1,CEP,RDH,D0H,HDH)
REWIND 1
IF(ITIME.EQ.0) GO TO 12
WRITE (6,*) "DO YOU WANT CURRENT INPUT LISTED?"
READ (5,1001) ANS
IF(ANS.NE.YES) GO TO 23
IF(ITIME.GT.0) WRITE (6,*) "CURRENT DATA ="
12 IF(ITIME.EQ.0) WRITE (6,*) "INITIAL INPUTS ="

C
LIST DATA FILE (TAPE)

C
DO 8 I=1,50
REAC (1,1000) A,B
IF(A.EQ.END) GO TO 6
WRITE (5,1002) A,B

6 WRITE (6,1002) END
CALL WRITE (DATA,6,CEP,RDH,D0H,HDH)
REWIND 1
ITIME = ITIME + 1
IF(ITIME.EQ.0) GO TO 23
WRITE (6,*) "DO YOU WANT TO CHANGE ANY DATA? ="
READ (5,1001) ANS
IF(ANS.NE.YES) GO TO 82
89 ISET = 0

C
READ IN CHANGES

C
DO 13 I=1,50
13 INEW(I) = 0
DO 2 I=1,1000
WRITE (6,*) "ENTER DATA OR END ="
READ (5,1000) AAAA,VALUE
IF(AAAA.EQ.END) GO TO 3

1001 FORMAT (A1)
DO 4 J=1,50
IF(AAAA.NE.ANAM(J)) GO TO 4
DATA(J) = VALUE
INEW(J) = 1
GO TO 2

4 CONTINUE
WRITE (6,2000) AAAA

2000 FORMAT (1X,****** DO NOT RECOGNIZE *,A4,* ******)

2 CONTINUE
3 CALL READ (DATA, INEW, ANAM, 5.0, PDOH, DDH, HDH)
GO TO 9
230
82 DO 83 I = 1, 50
83 INEW(1) = 0

C
SET UP DATA

C
LOAD INPUT DATA INTO VARIABLE SET
AND CONVERT DEGREES TO RADIANS

C
FZAM = DATA(1)/57.29578
FZTM = ABS (DATA (2))
PKDHX = DATA (3)
PKBLX = DATA (4)
FZAS = DATA (5)/57.29578
ITIG = 0
IF (FZAS.LT.0.) ITIG = 1
FZAS = ABS (FZAS)
FZTS = DATA (6)
OMEG = DATA (7)/57.29578
NGER = DATA (8)
NGEP = DATA (9)
IFUN = 0
NDHT = DATA (11)
NSMP = DATA (14)
NRQ = DATA (15)
DHAZ = DATA (16)/57.29578
NH = DATA (17)
NA = 0
ONGS = 0.
PKPF = DATA (21)
NVT = DATA (20)
NGLT = DATA (22)
JGRT = 1
JGLT = ISIGN (JGLT, NGLT)
NGLT = IABS (NGLT)
SRNG = DATA (23)
NPRT = DATA (24)
NDBG = DATA (25)
TGTC = DATA (26)
DUDR = DATA (27)
BLST = DATA (28)
IF (BLST.LT.0.) GO TO 94
BBLST (1) = BLST
IBBLST (1) = 100000.

C
BLST = 1.

C
94 KBLST = ABS (BLST)

C
IHIF = 0
IF (DATA (2).LT.0.) IHIF = 1
IF (PKDHX .EQ. 0.) PKDHX = 1.
IF (PKBLX .EQ. 0.) PKBLX = 1.

C
NLOGP = NGER
IF (NDBG.GE.1) WRITE (6,*), *DBG OPTION *, NDBG
IF (DATA (2).NE.0.) IFUZ = 2
IF (DATA (1).NE.0.) IFUZ = 1
XRNG = 0.

C
285 IF (XHGT.EQ.0.) GO TO 115
PROGRAM ARP  73/74  OPT=1

DO 116 I=1,NHT
  RAND = 10000.
  IF(SIGN(1.,DDH(I,1)).EQ.SIGN(1.,DDH(I,2))) GO TO 118
  IF(SIGN(1.,RDM(I,1)).EQ.SIGN(1.,RDM(I,2))) GO TO 118
  DO 119 J=1,2
  RAND = AMIN1(RAND,DDH(I,J))
  119 RRNG = AMIN1(RRNG,RDM(I,J))
  XRNG = SORT(RRNG+2. + HDM(I,1)*2.)
      GO TO 116
  118 XRNG = HDM(I,1)
  116 CONTINUE
  XRNG = AMIN1(XRNG,HDM(NHT,2))
  DL = ALOG(RRNG-XRNG)/10.
  DO 111 I=1,10
       XI = I
  111 RANGE(I) = RRNG + EXP(DL*XI)
          RANGE(11) = 1000.
          IF(NVT.LE.1) GO TO 67
  300 UI = 15*I2,N VT
  305 PVT(I) = PVT(1) + PVT(I-1)
  67 IF(NGLT.GT.0) GO TO 59
  66 DO 60 I=1,3
       60 GLTR(I,1) = 0.
  59 IF(NA.EQ.0) GO TO 48
  310 DO 28 I=1,NA
       28 XDOG(I) = X2XG(I)/57.29578
  40 CONTINUE

C C C READ IN PK GRIDS FOR EACH ATTACK ANGLE, BURST HEIGHT
C C C COMBINATION
  IF(NH.EQ.0) GO TO 78
  CALL GRIDS (PK1,NH,2,RGRD,DGRD,NR,ND,NDEG)
C C C LOOP OVER SIMULATIONS FOR EACH GUIDANCE ERRDR SET
C 78 DO 60 IJLP=1,NLP
C C C INITIALIZE COUNTERS
  DO 70 I=1,50
       FMR(I) = 0.
       70 IF(R(I) = 0.0)
       DO 52 I=1,12
       52 PKS(I,J,K) = 0.
       DO 52 J=1,6
       52 DO 60 K=1,10
             52 IKS(I,J,K) = 0
       60 DO 70 I=1,50
             70 PKMR = 0.
             70 PKDHT = 0.
             70 PKBASE = 0.
             70 PKSLT = 0.
             70 PKTGT = 0.
             70 PKG = 0.
             70 PKDT2 = 0.
             70 RRBAR = 0.
             70 RRBAR2 = 0.

03/13/81  08.28.23  PAGE  6
PROGRAM ARP  73/74  CPT=1

BDBAR = 0.
BDBAR2 = 0.
BRBAR = 0.
BRGAR = 0.
BHBAR = 0.
BHBAR2 = 0.

IF(PKPF.EQ.0.) PKPF = 1.
IF(PKPF.LT.0.) PKPF = 0.
SIGO = SOD(ILUP).
SIGH = SUM(ILUP).
MCT = 0.

C
BEIN SIMULATIONS
C
DO 1 ISIM=1,NSMP
IF(DATA(16).LT.0.) DHAZ = RDM(1)*2.*PI
PKGAMP = 0.0
PKDH = 0.
PKBLST = C.
PKROR = 0.

C
CHECK FOR DUD
C
IF(RDM(1).LE.DUDR) GO TO 18

C
SAMPLE FROM ATTACK ANGLE DISTRIBUTION

CALL BOXNO (Z1,Z2)
OMEGA = Z1*DGS + OMEGA
SIND = SIN(OMEGA)
COSD = COS(OMEGA)
TAND = 1.

IF(COSO.HE.0.) TAND = SIND/COSD

C
ROTATE COORDINATES OF HOMING POINT ACCORDING
TO AZIMUTH COMPONENT OF ATTACK ANGLE.

C
A,L COMPUTATIONS TO DETERMINE FUZING POINT ARE IN
ROTATED COORDINATE SYSTEM.

C
GMAR = GNR
GMOR = GM0
CALL ROTATE (GMAR,GMOR,DHAZ,1.)

C
SAMPLE FROM GUIDANCE ERROR DISTRIBUTION

CALL BOXNO (D,M)
GAR = SRT((SIGO+H)**2. + (SIGD+D)**2.)
GR = GMAR + SIGO*SIND
GD = GMOR + SIGD*

C
(GR,GD,GH) IS INTERCEPT OF
TRAJECTORY WITH GUIDANCE PLANE
(RF,DF,HF) WILL BE FUZING POINT ON TRAJECTORY.
400   C
     RF = G3
     DF = G3
     HF = GH
405   C
     CHECK FOR PRIMARY FUZE FUNCTION
     IEKUP = 0
     IF (RDZ(1).GT.PKFF) GO TO 16
410   C
     CHECK FOR HEIGHT FUZING
     IF (INFZ.EQ.1) GO TO 74
     Q3 = 0.
415   C
     CHECK FOR APPROPRIATE FUZING
     CALL BOXNC (21,22)
     QD = IFUZ + 1
     IF (NOBG.EQ.1) WRITE (6,5003) IFUZ,IGO,GR,GO,CH
     GO TO (65,75,32,85),103
420   C
     CHOOSE GLITTER POINT FOR FUZING. ANGULAR FUZE ONLY
     75 IF (UCQT.LT.9.AND.NGLT.LT.1) GO TO 76
825   C
     XGLT = NGLT
     NGLT = (RDZ(1)-0.0001)*XGLT + 1.0
     IF (IGLTEL.0) IGLT = 1
     88 XGLT = GLTR(1,IGLT)
     DG.T = GLTR(2,IGLT)
    90 NGLT = GLTR(3,IGLT)
     100 = 1
     GO TO 77
77 100 = NGLT
     GRMAX = -10000.
     77 DD(64,IGL=1,100)
     IF (ICO.EQ.1) GO TO 21
     RGLT = GLTR(IGLT,11G)
     DG.T = GLTR(2,IGLT)
     HGLT = GLTR(3,IGLT)
     440 21 IF (NOBG.EQ.1) WRITE (6,+) *RGLT,DGLT,HGLT = *,RGLT,DGLT,HGLT
    445   C
     ROTATE GLITTER POINT INTO ARP COORDINATE SYSTEM
     CALL ROTATE (RGLT,DGLT,DMAX,1.)
     455   C
     IF (NOBG.EQ.1) WRITE (6,*) *ROTATED GLITTER POINT = *
     IF (NOBG.EQ.1) WRITE (6,*) *DMAX,RGLT,DGLT,HGLT = *,DMAX,RGLT,DGLT
     C
     5003 FORMAT (1X,*,IFDZ,IGO = *,2(12,*,1X),*,GR,GO,CH = *,3(F6.1,*,1X))
    457   C
850   C
     USE LAW OF SINES AND LAW OF COSINES TO FIND
     FUSING POINT ON TRAJECTORY. FIRST PICK A POINT
     ALONG TRAJECTORY TO COMPUTE EKET (ANGLE BETWEEN
     TRAJECTORY AND A LINE (AB) FROM GLITTER POINT
     (HGLT,DGLT,RGLT) TO GUIDANCE LANE INTERCEPT
     (85,GS,FK) = NOTE THAT EVERYTHING IS IN ROTATED
COORDINATE SYSTEM (THROUGH AZIMUTH ATTACK ANGLE COMPONENT). THEN, KNOWING BETAX AND FUZING ANGLE (ANG) COMPUTE ANGLE (GAMMA) WITH ITS VERTEX AT GLITTER POINT AND OPPOSITE TRAJECTORY SEGMENT BOUNDED BY GUIDANCE PLANE INTERCEPT AND FUZING POINT. FINALLY, KNOWING GAMMA, AB, AND ANG, COMPUTE DISTANCE FROM GUIDANCE PLANE INTERCEPT TO FUZING POINT (USING THE LAW OF SINES).

TANX = TAN
IF(SIND.GE.0.) TANX = 1.
CB = 10.
IF(SIND.NE.0.) CB = CB/SIND

GRL, GDL, GHL ARE COORDINATES OF A POINT ON THE TRAJECTORY USED TO COMPUTE BETAX.

GRL = GR - 10./TANX
GDL = GD
GHL = GH
IF(SIND.NE.0.) GHL = GH + 10.
AB = SQRT(AB2).

USE LAW OF COSINES TO COMPUTE BETAX, ANGLE WITH VERTEX AT GLITTER POINT AND OPPOSITE TRAJECTORY SEGMENT BOUNDED BY GUIDANCE PLANE INTERCEPT AND FUZING POINT.

BETAX = ACOS((AD2-BB2+CB*CB)/(2.*AB*CB))
IF(NDBG.EQ.1) WRITE (6,*) 'BETAX, GRL, GDL, GHL, AB, CB = *
C BETAX, GRL, GDL, GHL, AB, CB
FZASX = FZAS
IF(ITIG.EQ.1) FZASX = 0.

ANGULAR FUZING FUNCTION

ANG = Z2+FZASX + FZAM
IF(FZAM.LT.0.) ANG = FZAM + RDM(1)*(FZASX-FZAM)
IF(ANG.LT..01745) GO TO 16
IF(ANG.GT.PI) GO TO 16

D2 IS DISTANCE ALONG TRAJECTORY FROM GUIDANCE PLANE INTERCEPT TO FUZING POINT.

GAMMA = PI - BETAX - ANG
IF GAMMA.LT.ZERO, USE SUPPLEMENT OF ANG FOR FUZING.

IF(GAMMA.LT.PI.) ANG = PI - ANG
D2 = AB*ABSIN(GAMMA)/(ABSIN(ANG))
IF(NDBG.EQ.1) WRITE (6,*) 'D2, GAMMA, ANG = *
IF(100.EQ.1) GO TO 22
IF(D2.LT.GRMAX) GO TO 84
GRMAX = D2
IGLT = IGL
84 CONTINUE
GO TO 86

515 C C C
LINEAR FUZING FUNCTION (ALONG TRAJECTORY)
FUZING DIRECTION IS POSITIVE IN THE NEGATIVE
RANGE DIRECTION, I.E., A POSITIVE CHANGE IN
THE FUZING DISTANCE, O2, IS IN THE NEGATIVE
RANGE DIRECTION.

520 C C C
22 IF(ITG.EQ.1) F2S = DMIN*TAN(FZAS)
D2 = O2 + Z2+F2S + F2M
RF = G2 - O2+CD5O
HF = GH + Q2+SIGN
DF = GD
GO TO 85

525 C C
BACKUP FUZING

530 C C
16 HF = 0.
IBCUP = 1
IF(OMEGA.EQ.0.) GO TO 5
IF(NVT.EQ.0.) GO TO 17

535 C C
87 K = ROM(1)
DO 60 K=1,NVT
K = K
IF(KK.LE.PVT(K)) GO TO 66
65 CONTINUE
66 HFX = VINT(KK)
IF(HFX.LE.HF) GO TO 24
HF = HFX
17 RF = GR - (HF+GH)/TANO
DF = GD
GO TO 61

540 C C
5 WRITE (6,*) "NO BACKUP FUZING FOR OMEGA = 0."
WRITE (6,*) "TRAJECTORY CLOSEST POINT OF APPROACH TO TARGET"
WRITE (6,*) "CENTER IS USED"
RF = 0.
DF = GD
HF = GH
GO TO 61

545 C C
HEIGHT FUZING

550 C
74 IF(SINO.EQ.0.) STOP 74
CALL BOXNO (Z1,Z2)
HF = FZTM + Z1+FZTS
RF = (GH+HF)/TANO

555 C
85 IF(OMEGA.EQ.0.) GO TO 24
IF(NVT.LE.0.) GO TO 87

560 C C
CHECK FOR FUZING POINT BELOW GROUND

565 C C
24 IF(HF.GE.0.) GO TO 61
IF(OMEGA.EQ.0.) GO TO 61
RF = RF + HF/TANO
HF = 0.

570 C C
PUT BURST POINT IN TARGET COORDINATE SYSTEM FOR
C BUST ANP DIRECT HIT COMPUTATIONS.

C 61 CALL ROTATE (RF,DF,DHAIU.T-1.1)
      LN = RF
      BD = DF
      BH = HF
      IF(NDBQ.GE.1) WRITE (6,*); "BR, BD, BH AT STMT 61 = ",BR, BD, BH

C SET UP BLST VALUE FOR BLST VS. HGT

C 580 IF(NBLST.LE.6) GO TO 105
      DO 10 I=1,NBLST
      IF(MF.GT.HBLST(I)) GO TO 10
      BLST = BBLST(I)
      GO TO 105

      10 CONTINUE
      BLST = 0.
      WRITE (6,*); "HF EXCEEDS ALL HBLST, HF = ",HF
      GO TO 106

C Tests on Line 105

C DETERMINE DIRECT HIT PK

C USE 2 POINTS TO DEFINE TRAJECTORY, BURST POINT
      (BR, BD, BH) AND POINT AT ER+10 (RBS.DBS, HBS).
      IF AZIMUTH ATTACK ANGLE IS 90 DEGREES, SET
      RBS, DBS, HBS POINT AT ER+10.
      (RPN, DPN, HPN) WILL BE BURST POINT, WITH OR
      WITHOUT DIRECT HIT.

C IPN IS PENETRATION INDEX (0 = NO PENETRATION,
      N = BOX N PENTENETATED)

C RPN = BR
      DPN = BD
      HPN = BH
      IF(ABS(DATA(16)).EQ.0.) GO TO 95

C RBS = BR + 10.
      DBS = BD - 10.*TAN(DHAZ)
      HBS = BH - 10.*TANDOH/DS(DHAZ)
      GO TO 95

C Checks on Line 95

C RBS = BR
      DBS = BD + 10.
      HBS = BH + 10.*TAND

C IPN = 0

C CHECK EACH BOX FOR PENETRATION

C IF(NDBQ.EQ.1) WRITE (6,*); "OMEGA, RBS, DBS, HBS = ",OMEGA, RBS, DBS, HBS
C IF(NDBQ.EQ.1) WRITE (6,*); "RF, DF, HF = ",RF, DF, HF
C IF(NDBQ.EQ.1) WRITE (6,*); "GR, GD, GH = ",GR, GD, GH
      GO TO 92
      IF(DQ.EQ.1) WRITE (6,*); "OMEGA, RBS, DBS, HBS = ",OMEGA, RBS, DBS, HBS
      IF(DQ.EQ.1) WRITE (6,*); "RF, DF, HF = ",RF, DF, HF
      IF(DQ.EQ.1) WRITE (6,*); "GR, GD, GH = ",GR, GD, GH
      GO TO 92

C Tests on Line 92

C IF(BR.LT.ROH(1,1)) GO TO 92
      IF(DATAP(16).NE.6.) GO TO 105
      IF(BR.LT.ROH(1,1)) GO TO 92
      IF(DQ.EQ.1) WRITE (6,*); "OMEGA, RBS, DBS, HBS = ",OMEGA, RBS, DBS, HBS
      IF(DQ.EQ.1) WRITE (6,*); "RF, DF, HF = ",RF, DF, HF
      IF(DQ.EQ.1) WRITE (6,*); "GR, GD, GH = ",GR, GD, GH
      GO TO 92

C Tests on Line 92

C IF(BR.GT.ROH(1,2)) AND OMEGA.GE.0.) GO TO 92

C AT END OF PROCESSING WRITE "COMPLETE".
IF(BH,LT,HDM(1,1),AND.OMEGA.EQ.0.) GO TO 92
RDH1 = RDH(1,1)
RDH2 = RDH(1,2)
DDH1 = DDH(1,1)
DDH2 = DDH(1,2)
HDM1 = HDM(1,1)
HDM2 = HDM(1,2)

635  C
C      IPEN = NUMBER OF SIDES PENETRATED (MUST BE 0 OR 2)

C
IPEN = 0
IF(ABS(DATA(16)).EQ.90.) GO TO 102

640  C
C      CHECK RANGE SIDES

DO 97 K=1,2
RDH = RDH1
IF(K.EQ.2) RDH = RDH2
CALL SEARCH (1,1,RDH,DA,HA)
IF(NDBG.EQ.2) WRITE (6,*); IPEN,RDH,DA,HA = *,IPEN,
1 RDH,DA,HA
97 CONTINUE

650  C
C      CHECK DEFLECTION SIDES

DO 107 K=1,2
DDH = DDH1
IF(K.EQ.2) DDH = DDH2
CALL SEARCH (1,2,RA,DDH,HA)
IF(NDBG.EQ.2) WRITE (6,*); IPEN,RA,DDH,HA = *,IPEN,
1 RA,DDH,HA
IF(IPEN.EQ.2) GO TO 92
107 CONTINUE

108 IF(OMEGA.EQ.0.) GO TO 108

660  C
C      CHECK HEIGHT SIDES

DO 117 K=1,2
HDM = HDM1
IF(K.EQ.2) HDM = HDM2
CALL SEARCH (1,3,RA,DA,HDM)
IF(NDBG.EQ.2) WRITE (6,*); IPEN,RA,DA,HDM = *,IPEN,
1 RA,DA,HDM
IF(IPEN.EQ.2) GO TO 92
117 CONTINUE

675  C
C      SET UP BURST COORDINATES (BR,BD,BH) FROM DIRECT HIT.

101 IF(INTERFACE.EQ.1) STOP 117
92 CONTINUE

680  C
C      SET UP BURST COORDINATES (BR,BD,BH) FROM DIRECT HIT.

BR = RPN
BD = DPN
BH = HPN
685 106 IF(BH,GE,0.) GO TO 37
     IF(OMEGA,EQ,0.) STOP 106
     BR = BR + BH/TANQ
     BH = 0.
      C
   690 C      COMPUTE NEAR MISS BLAST KILL
   695 37 IF(NBLST,EQ,0) GO TO 90
     IF(HCMT,EQ,0) GO TO 103
     DO 104 I=1,HCMT
     IBLST = I
     CALL BLAST (IBLST, BR, BLST, RHMT(I))
     CALL BLAST (IBLST, BD, BLST, DGMT(I))
     CALL BLAST (IBLST, BR, BLST, HDMT(I))
     104 CONTINUE
     GO TO 90
  103 DIST = SORT(BR, BR + BD, BD + (BH-TGTC)*(BH-TGTC))
     IF(DIST,GT,BLST) GO TO 90
   700 11 PKBLST = PKBLST + PKBLX
     C
  705 C      COMPUTE RAGAR BLAST KILL
   710 90 IF(NDBG,EQ,2) WRITE (6,*)*IPN,RPN,DPN,HPN,BR,BD,BH = *
     C  IPN,RPN,DPN,HPN,BR,BD,BH
     IF(NDBG,EQ,0) GO TO 27
     BRD = BR-RDR(1)
     DRD = BD+RDR(2)
     RDR = BH-RDR(5)
     RDR = SORT(BRD+BD+BD+RDR+RDR+RDR+RDR)
     PKRDR = 1.0
     IF(RRDR,GT,RDR(4)) PKRDR = 1.0-(RDR-RDR(4))/(RDR(5)-RDR(4))
     IF(RRDR,GE,RDR(5)) PKRDR = 0.
  715 5004 FORMAT (1X,*BR,BD,BH = +,3(F6.1,+,*),1X))
  720 27 IBX = 0
  725 20 IROT = 0
     IF(NDBG,EQ,1) WRITE (6,5004) BR, BD, BH
     IF(NH,EQ,0) GO TO 50
      C
  725 C      COMPUTE PK DUE TO FRAGMENTATION (PKSAMP)
  730 C      INTERPOLATE IN RANGE, DEFLECTION, HEIGHT & ANGLE TO
  735 C      GET FRAGMENTATION PK FROM PK GRIDS.
  740 C      II = 1
     IF(BH,GT,HGTM(NH+1)) GO TO 50
      C
  745 C      ROTATE SURF PK FOR FRAGMENTATION PK
  750 C      INTERPOLATION INTO ARP COORDINATE SYSTEM.
  755 C      RECALL THAT PK GRIDS ARE IN PROJECTILE COORDINATE
  760 C      SYSTEM.
  770 C      CALL ROTATE (BR, BD, DHAZ, 1.)
     IROT = 1
  780 C      LOCATE HEIGHT BOUNDARIES

006940
006950
006960
006970
006980
007000
007010
007020
007030
007040
007050
007060
007070
007080
007090
007100
007110
007120
007130
007140
007150
007160
007170
007180
007190
007200
007210
007220
007230
007240
007250
007260
007270
007280
007290
007300
007310
007320
007330
007340
007350
007360
007370
007380
007390
007400
007410
007420
007430
007440
007450
007460
007470
007480
007490
007500
PROGRAM ARP   73/74 OPT=1

C
DO 20 I=1,NH
    IH2 = 1
20   CONTINUE
25   IH = IH2 - 1
    IF(IH.EQ.0) IH = 1
    IF(IH.LT.0) IH = NH
    IF(NDBG.EQ.4) WRITE (6,*) "IH,IH2,NR,ND,DU,DR,BR,BD,BH = ",
C IH,IH2,NR,ND,DU,DR,BR,BD,BH
31   CALL INTERP(BR,BD,BH,BR,DR,ED,M,GT,IH2,PK1,PKA,NR,ND,DU,BH,NO3D)
50   PKSAMP = PKA
GO TO 41
760   COMPUTE SPHERICAL COORDINATES TO BURST POINT (BR,BD,BH)
600   FROM GROUND ZERO (0,0,0)
765   SA1 = ANGLE OFF POSITIVE RANGE AXIS MEASURED
C CLOCKWISE
770   SA2 = ANGLE OFF R-D PLANE MEASURED TOWARD POSITIVE
C RANGE FROM BURST POINT TO (0,0,0)
775   H-AXIS IN VERTICLE PLANE
C
41   IF(NDBG.EQ.4) WRITE (6,*) "PK(FRAG) = ",PKSAMP
C
770   GET BURST POINT BACK INTO TARGET COORDINATE
C
C   SYSTEM IF IROT = !.
C
C
IF(IROT.EQ.1) CALL ROTATE (BR,BD,BH2,-1.)
BRR = BR+3D
BED = BD+3D
BHH = BH*BH
RRR = BRR + BED + BHH
RR = SQRT(RRR)
WRITE (6,*) BR,BD,BH,RR
BRBAR = BRBAR + BR
BRBAR2 = BRBAR2 + BRR
BDBAR = 3*BAR + BD
BDBAR2 = BDBAR2 + BDD
BDBAR = BDBAR + BH
BDBAR2 = BDBAR2 + BHH
RRBAR = RRBAR + RR
RRBAR2 = RRBAR2 + RRR
SA1 = PT/2.
SA2 = 0.
IF(IR.BEQ.0,.) GO TO 55
790   SA1 = ATAN2(BR,ED)
C IF(SA1.LT.0.) SA1 = 2.*PI + SA1
55   IF(BD.EQ.0. AND.BR.EQ.0.) GO TO 56
799   SA2 = ATAN(BH/SQRT(BR*BR+BD*BD))
56   SA1 = SA1+360./(2.*PI)
SA2 = SA2+360.//(2.*PI)
DO 57 I=1,12
57   I=I
I=I
I=I
IF(SA1.LT.ALPHA(I+1)) GO TO 58
070510
070520
070530
070540
070550
070560
070570
070580
070590
070600
070610
070620
070630
070640
070650
070660
070670
070680
070690
070700
070710
070720
070730
070740
070750
070760
070770
070780
070790
070800
070810
070820
070830
070840
070850
070860
070870
070880
070890
070900
070910
070920
070930
070940
070950
070960
070970
070980
070990
080000
080010
080020
080030
080040
080050
080060
080070
57 CONTINUE
803 DO 88 * = 1.6
88 IQA2 = I
IF(SA2.LT.BETA(I-1)) GO TO 99
99 CONTINUE
904 SR = SQRT(8R+BR + BD+BD + MH+BH)
ISR = 0
DO 100 I=1,10
10 I = I
IF(I.EQ.10) I = 11
ISR = ISR + 1
IF(SR.LT.RANGE(I)) GO TO 110
100 CONTINUE
110 IF(ND`G.EQ.6) WRITE (6,*,'*ISA1,ISA2.ISR = *,ISA1,ISA2.ISR
IF(NDG.EQ.6) WRITE (6,*,'*SA1,SA2,SR = *,SA1,SA2,SR

C STORE PK'S ACCORDING TO SPHERICAL COORDINATES

IKL(ISA1,ISA2,ISR) = IKS(ISA1,ISA2,ISR) + 1

C SUM PK'S OVER ALL SAMPLES

IF(NDG.EQ.6) WRITE (6,*,'*PKR,PKR,PKD,PKB = *,PKSAMP,PKRDER,PKOH
C.PXSMP
PKBASE = PKBASE + PKSAMP
PKRADR = PKRADR + PKRDER
PKDM = PKDM + PKOH
PKBLT = PKBLT + PKSAMP
PKSAMP = 1. - ((1.-PKSAMP)*(1.-PKRDER)*(1.-PKOH)*(1.-PKBLT))
PKS(ISA1,ISA2,ISR) = PKS(ISA1,ISA2,ISR) + PKSAMP
PKTOT = PKTOT + PKSAMP
PKPTOT = PKPTOT + PKSAMP
IF(NDGG.EQ.6) WRITE (6,*,'*PKSAMP

C FORMAT (5X, SAMPLE PK = *,F6.4)

3002 FORMAT (5X, SAMPLE PK = *,F6.4)
IF(NPRT.EQ.1) GO TO 1
IF(NDG.EQ.10).NE.0) GO TO 1
WRITE (6,*,'*NO. SIMULATIONS, PK = *,ISIM,PXKPRNT
GO TO 1
19 NCT = NCT + 1
CONTINUE

C DISPLAY FINAL RESULTS

IF(NPRT.EQ.0) GO TO 79
WRITE (6,2002) WRITE (6,*,'**FINAL RESULTS'
3000 FORMAT (/1X,*PK = *,F6.4,2X,*PKSD = *,F6.4,2X,*NSAMP = *,16.)
79 XSAMP = NSAMP
PKSD = PKTOT/XSAMP
PKBASE = PKBASE/XSAMP
PKRADR = PKRADR/XSAMP
PKDM = PKDM/XSAMP
PKBLT = PKBLT/XSAMP
PKILLUP = PKBASE
PKR(ILLUP) = PKBASE
PROGRAM ARP

PKD(ILUP) = PKOHIT
PKG(ILUP) = PKBAR
PKBL(ILUP) = PKBLT
XSAMP = NSMP = NCT

IF(NSMP.EQ.NCT) XSAMP = 1.
XSAMP = XSAMP+1.
IF(NSMP.EQ.0) XSAMP = 1.

BRG(ILUP) = BRBAR/XSAMP

BDSG(ILUP) = BDSAR/XSAMP

PRSG(ILUP) = SORT((BBAR2 - XSAMP*BDSG(ILUP)+BDSG(ILUP))/XSAMP)

READ (5,1001) ANS

WRITE (6,2001)
WRITE (6,*) "DO YOU WANT PK VS R, ALPHA, BETA?"

WRITE (6,2002)
WRITE (6,*) "DO KP VS R, ALPHA, BETA, WHERE ALPHA IS AZIMUTH ANGLE"
WRITE (6,*) "MEASURED FROM POSITIVE RANGE AXIS TOWARD POSITIVE"
WRITE (6,*) "DEFLECTION AXIS (0 TO 360), BETA IS ELEVATION ANGLE"
WRITE (6,*) "MEASURED FROM NEGATIVE HEIGHT AXIS TO POSITIVE"
WRITE (6,*) "HEIGHT AXIS (0 TO 90)."

3004 FORMAT (2X,F6.4,2X,F5.1,2(2X,F6.1,* -,F6.1))
DIMENSION ALPHAt(3),BETA(7),PVT(5)
DIMENSION PK(10),CEP(10),RANGE(11),RSUM(10)
DIMENSION PKl(10),PKR(10),PK(10),PK0(10)
DIMENSION RDH(5,2),DSH(5,2),DH(5,2)
DIMENSION RSG(10),RS(10),RS(10),RDSG(10),RDSG(10)
DIMENSION DHG(10),BHS(10)
COMMON /SRCH/IPEN,IPN,RSBS,CSBS,RS,BS,SD,OD,OMEGA,RPN
1,DPN,HPN,M0H,M0H2,MRH2,MRH2,MRH2
COMMON /ROWT/HTK,HTK,HTK,HTK,HTK,HTK,HTK,HTK
1,GAH,SDG,SGG,SGG,SGG,SGG,SGG,SGG,SGG,SGG,SGG
DATA ALPHA/0.0.,30.,60.,90.,120.,150.,180.,210.,240.,270.,300.0/
DATA C330./300.,0.0/...
WRITE (6,*) * NOTE: FOLLOWING GUIDANCE ERROR PARAMETERS* 000670
WRITE (6,*)  (S100, S120) ARE MEASURED* 000680
WRITE (6,*) * IN PLANE NORMAL TO TRAJECTORY AND* 000690
WRITE (6,*) * PASSING THROUGH HOMING POINT* 000700
WRITE (6,*) "NGER - NUMBER OF GUIDANCE ERRORS TO CONSIDER* 000710
WRITE (6,*) * ENTER HOMING POINT (R.D.M.) GUIDANCE* 000720
WRITE (6,*) * ERRORS ARE DISTRIBUTED ABOUT HOMING PT.* 000730
WRITE (6,*) "NCEP - 1.. IF CEP IS INPUT FOR GUIDANCE ERROR SIGNUS 000740
WRITE (6,*) "FZAM,FZAS,FZTM,FZTS - FUZING ERROR OPTIONS* 000750
WRITE (6,*) "FZAM - MEAN ANGLE AT WHICH FUZING OCCURS ON* 000760
WRITE (6,*) * INTERCEPT* 000770
WRITE (6,*) "FZAS - STD DEV ASSOCIATED WITH FZAM* 000780
WRITE (6,*) "NOTE: FUZING ANGLE IS CONstrained TO [0,PI)* 000790
WRITE (6,*) "NOTE: FOR UNIFORm FUZING ANGLE BETWEEN FZAM* 000800
WRITE (6,*) "AND FZAS, ENTER A NEGATIVE VALUE FOR FZAM* 000810
WRITE (6,*) "FUZING ANGLE WILL BE CHosen UNIFORmLY RANDOM* 000820
WRITE (6,*) * BETWEEN POSITIVE FZAM AND FZAS* 000830
WRITE (6,*) "FOR TIME-TO-GO FUZE, ENTER NEGATIVE FZAM.* 000840
WRITE (6,*) "NOTE: FUZING PLANE PASSES THROUGH FUZING GLITTER* 000850
WRITE (6,*) "FUZING WILL OCCUR ALONG TRAJECTORY* 000860
WRITE (6,*) "FZTM = MEAN DISTANCE FROM GUIDANCE PLANE AT WHICH* 000870
WRITE (6,*) "FUZING WILL OCCUR ALONG TRAJECTORY* 000880
WRITE (6,*) "NOTE: ENTER A NEGATIVE FZTM FOR HEIGHT FUZING * 000890
WRITE (6,*) "WITH MEAN HEIGHT ABS(FZTM)* 000900
WRITE (6,*) "FZTS = STD DEV ASSOCIATED WITH FZTM* 000910
WRITE (6,*) "SAMP - SAMPLE SIZE* 000920
WRITE (6,*) "FKHM - NUMBER OF HEIGHTS AT WHICH FRAGMENTATION* 000930
WRITE (6,*) "PK DATA WILL BE DEFINED* 000940
WRITE (6,*) "NOTE: FKHM < 9* 000950
WRITE (6,*) "PKPF = PROBABILITY OF PRIMARY FUZE FUNCTIONING* 000960
WRITE (6,*) "PDVF = 0. FOR PG BACKUP, NVT FOR VT BACKUP FUZE* 000970
WRITE (6,*) "HERE NVT = NUMBER OF VT BUSt HEIGHTS* 000980
WRITE (6,*) "GLTR = 0. IF PRIMARY FUZE FUNCTIONS RELATIVE TO* 000990
WRITE (6,*) "CENTER OF TARGET, NOLT IF PRIMARY FUZE* 001000
WRITE (6,*) "FUNCTIONS RELATIVE TO ANY ONE OF NOLT* 001010
WRITE (6,*) "EQUALLY LIKELY GLITTER POINTS .* 001020
WRITE (6,*) "SET NOLT NEGATIVE TO PICK FIRST* 001030
WRITE (6,*) * POINT ENCOUNTERED.* 001040
WRITE (6,*) "SRNG = MAXIMUM RANGE FOR COMPUTING PK VS RANGE* 001050
WRITE (6,*) "PRINT = 1. TO PRINT SUMMARY ONLY, 0. OTHERWISE" 001060
WRITE (6,*) "DBG - 6. TO PRINTOUT PROGRAM DEBUGGING DATA" 001070
WRITE (6,*) "DBG = 1. GUIDANCE & FUZING DATA* 001080
WRITE (6,*) "DBG = 2. DIRECT HIT PENETRATION DATA* 001090
WRITE (6,*) "DBG = 4. PK BOX DATA* 001100
WRITE (6,*) "DBG = 5. PK GRIDS* 001110
WRITE (6,*) "DBG = 6. PK VS R DATA* 001120
WRITE (6,*) "TGSC = HEIGHT OF TARGET CENTER ABOVE GROUND* 001130
WRITE (6,*) "DUDR = DLD RATE OF PROJECTILE. EXPRESSED AS A FRACTION* 001140
WRITE (6,*) * CHIT = DIRECT HIT OPTION, NUMBER OF TARGET BOXES* 001150
WRITE (6,*) * IF CHIT IS OMITTED AND BLST IS INCLUDED,* 001160
WRITE (6,*) "BLST IS RADIUS FROM (0.0, TGSC) WITHIN* 001170
WRITE (6,*) "WHICH PKBLS = 1.* 001180
WRITE (6,*) "PKOH = DIRECT HIT PK (0. = 1.)* 001190
WRITE (6,*) "PKBL = BLAST PK (0. = 1.)* 001200
WRITE (6,*) "RACK = 1., DEFINE FUNC FOR BLAST KILL OF RADAR ONLY* 001210
WRITE (6,*) "AND READ IN RADAR ANTENA COORDINATES." 001220
Program ARP

73/74  OPT=1

FTN 4,8+50G  03/13/81  08.29.0  PAGE 3

115 WRITE (6,*)  "TO DEFINE FUNC. SPECIFY R1 AND R2","  001240
WRITE (6,*)  "WHERE BLAST PK IS 1 OUT TO R1 AND","  001250
WRITE (6,*)  "DECLINES LINEARLY TO 0 AT R2","  001260
WRITE (6,*)  "HAZ = AZIMUTH ANGLE OF ATTACK OFF FRONT OF TARGET","  001270
WRITE (6,*)  "TOWARD DRIVER SIDE. SET TO -1. FOR RANDOM","  001280
WRITE (6,*)  "BLST = BLAST RADIUS WITHIN WHICH VEHICLE PK+PKBL","  001290
WRITE (6,*)  "NOTE: TO ENTER BLAST RADII VS. BURST HEIGHT","  001300
WRITE (6,*)  "ENTER NEGATIVE NUMBER OF BLAST,HGT PAIRS","  001310
WRITE (6,*)  "IN PLACE OF VALUE OF BLST. PAIRS OF","  001320
WRITE (6,*)  "BLAST,HGT ARE ENTERED IN ASCENDING ORDER","  001330
WRITE (6,*)  "OF HEIGHT","  001340
WRITE (6,*)  "COORDINATE SYSTEM IS RECTANGULAR.","  001350
WRITE (6,*)  "TARGET HEADING IS NEGATIVE RANGE.","  001360
WRITE (6,*)  "DRIVER SIDE (LEFT) IS POSITIVE DEFLECTION.","  001370
WRITE (6,*)  "HEIGHT IS MEASURED FROM GROUND","  001380

130 PRINT  = 0
ISET = 0
ITIME = 0
CALL ROMOUT(INIT)
15 CALL ROMIN(INIT)

135 IF (IRD.EQ.5) GO TO 180
IF (NPRT.GT.0) GO TO 180
WRITE (6,*)  "ENTER DATA BY ENTERING CODE NAME","  001470
WRITE (6,*)  "FOLLOWED BY A COMMA AND THE VALUE IN FLOATING","  001480
WRITE (6,*)  "FORMAT. TO END DATA ENTRY, ENTER","  001490
WRITE (6,*)  "THE WORD END IN COLUMNS 1-5","  001500

88 WRITE (6,*)  "DO YOU WISH TO INITIALIZE DATA FROM","  001550
WRITE (6,*)  "DATA FILE TAPE?","  001570
READ (5,1001) ANS
150 IRD = 5
IF (ANS.EQ.YES) IRD = 1
80 REWIND 1
REWIND 2
REWIND 3
REWIND 4
PI = ATAN2(0.,-1.)
DO 51 I = 1,10
51 PKG(I) = 0.

160 CALL INITIALIZ OR UPDATE DATA

165 FORMAT (A4,1X,F10.3)
1000 FORMAT (A4,1X,F10.3)
DO 55 J = 1,50
IF (AAAA.EQ.NE.ANAM(J)) GO TO 55
INKE(J) = 1
DO 7 I = 1,50
DATA(I) = VALUE
170 CALL TO
53 CONTINUE
  WRITE (6,2000) AAA
  GO TO 7

14 CALL READ (DATA,INAME,TRD,1,DRH,DDH,MDH)

    SET UP TAPE1

9  REIND 1
    DO 61 I=1,50
      IF(DATA(I).EQ.0) GO TO 61
      WRITE (1,1001) ANAM(I),DATA(I)
    CONTINUE

61  WRITE (1,1000) END
    CALL WRITE (DATA,1,CEP,DRH,DDH,MDH)
    REIND 1
    IF(ITEM.NE.0) GO TO 12
    WRITE (6,99) 'YOU WANT CURRENT INPUT LISTED?'
    READ (5,IV0) ANS
    IF(ANS.NE.YES) GO TO 12

   IF(ITEM.GT.0) WRITE (6,99) 'CURRENT DATA : '
   12 IF(ITEM.EQ.0) WRITE (6,99) 'INITIAL INPUTS : '

C

195  LIST DATA FILE (TAPE1)

DO 8 I=1,50
   READ (1,1000) A,B
   IF(A.EQ.END) GO TO 6
   WRITE (6,1002) A,B

8  FORMAT (1X,AA,1X,F10.3)

6  WRITE (6,1C2) END
    CALL WRITE (DATA,6,CEP,DRH,DDH,MDH)
    IFITEM = ITEM + 1
    IF(ISET.EQ.1) GO TO 60
    WRITE (6,99) 'DO YOU WANT TO CHANGE ANY DATA? - '

205  READ (5,1001) ANS
    IF(ANS.NE.YES) GO TO 82
    ISET = 0

89  ISET = 0

210  CONTINUE

C

READ IN CHANGES

C

DO 13 I=1,50
  13 INEX(I) = 0
  DO 2 I=1,1000
      WRITE (6,99) 'ENTER DATA OR END - '
      READ (5,1000) AAAA,VALUE
      IF(AAA.EQ.END) GO TO 2

200  FORMAT (A1)

220  DO 4 J=1,50
      IF(AAA.NE.ANAM(J)) GO TO 4
      DATA(J) = VALUE
      INEX(J) = 1
      GO TO 2

225  CONTINUE

4  WRITE (6,2000) AAA

2000  FORMAT (1X,'***** DO NOT RECOGNIZE "A"*****')

2 CONTINUE
3 CALL READ (DATA, INEW, ANAM, S, O, ROH, DQH, HDH)
GO TO 9
82 GO TO 83
83 INEW(1) = 0
C C C C
SET UP DATA
235 C C
LOAD INPUT DATA INTO VARIABLE SET
240 C C AND CONVERT DEGREES TO RADIANS
C C
FZAM = DATA(1)/57.29578
FZTM = ABS(DATA(2))
PKDH = DATA(3)
PBLX = DATA(4)
FZAS = DATA(5)*57.29578
ITG = 0
245 IF(FZAS.LT.0.) ITG = 1
FZAS = ABS(FZAS)
FZTS = DATA(6)
OMEG = DATA(7)/57.29578
NGE = DATA(8)
250 NCE = DATA(9)
IFUN = 0
NCHT = DATA(10)
NSHP = DATA(11)
NDR = DATA(12)
255 DHAZ = DATA(16)/57.29578
NH = DATA(17)
NA = 0
CMGS = 0.
PKPF = DATA(21)
NVT = DATA(20)
NGLT = DATA(22)
JGLT = 1
JGLT = ISIGN(JGLT,NGLT)
NGLT = IABS(NGLT)
260 SRNG = DATA(22)
NPRT = DATA(24)
NDSG = DATA(25)
TGTC = DATA(26)
265 DUDR = DATA(27)
BLST = DATA(29)
IF(BLST.LE.0.) GO TO 94
BBLST(1) = BLST
270 MBLST(1) = 060000.
BLST = 1.
BLST = 1.
94 NLBLST = ABS(BLST)
275 IFZ = 0
280 IF(DATA(2).LT.0.) IFZ = 1
IF(PKDHX.EQ.0.) PKDHX = 1.
IF(PKBLX.EQ.0.) PKBLX = 1.
285 NCP = NGE
IF(NLBSG.GE.1) WRITE (6,*), "DEUG OPTION",NLBSG
IF(DATA(2).NE.0.) IFUZ = 2
IF(DATA(1).NE.0.) IFUZ = 1
XNG = 0.
IF(NLHT.EQ.0) GO TO 115
00 DO 116 I=1,NDMT
02 RRNG = 10000.
03 IF(SIGN(1.,DDH(I,1)).EQ.SIGN(1.,DDH(I,2))) GO TO 118
04 IF(SIGN(1.,RDH(I,1)).EQ.SIGN(1.,RDH(I,2))) GO TO 116
290 DO 119 J=1,2
05 RRNG = AMIN1(RRNG,EDH(I,J))
06 119 RRNG = AMIN1(RRNG,RDH(I,J))
07 XRN0 = SQRT(RRN0**2. + MDH(I,1)**2.)
08 GO TO 116
295 118 XRN0 = MDH(I,1)
116 CONTINUE
115 XRNG = AMIN1(XRNG,MDH(NDMT,2))
110 IF(SRNG.EQ.0.) SRNG = 100.
DL = ALOG(SRNG-XRNG)/10.
300 DO 111 I=1,10
301 XI = I
305 111 RANGE(I) = XRNG + EXP(DL*XI)
304 RANGE(11) = 1000.
306 IF(NVT.LE.1) GO TO 67
307 DO 60 I=2,NVT
308 60 PTV(I) = PTV(I) + PTV(I-1)
309 67 IF(VGLT.GT.0) GO TO 59
310 DO 60 I=1,3
311 60 GLTH(I,1) = 0.
312 59 IF(MA.EQ.0) GO TO 48
313 DO 28 I=1,NA
314 28 XCMG(I) = XDG(I)/57.29578
48 CONTINUE
315 C C C READ IN PK GRIDS FOR EACH ATTACK ANGLE/BURST WEIGHT
316 C COMBINATION:
317 C
318 C IF(MA.EQ.0) GO TO 78
319 C CALL GRIDS (PK1,NH,2,RGRD,DGRD,MR,ND,NDBG)
320 C C LOOP OVER SIMULATIONS FOR EACH GUIDANCE ERROR SET
320 C
78 DO 69 ILOP=1,NLCP
325 C C C INITIALIZE COUNTERS
329 C
330 DO 70 I=1,50
331 PKM(I) = 0.
332 70 JKMK(I) = 0.
333 DO 52 J=1,12
334 DO 52 K=1,10
335 IKS(J,1,K) = 0.
336 52 PKS(J,1.K) = 0.
337 C C C PKRDR = 0.
338 C C C PKDHIT = 0.
339 C C C PKBASE = 0.
340 C C C PKBLT = 0.
341 C C C PKGT = 0.
342 540 PKGT2 = 0.
343 C C C RRBAR = 0.
344 C C C RRBAR2 = 0.
BEGIN SIMULATIONS

CALL BOXND (Z1, Z2)

GAMMA = 2 * COSG + CMED
SIN = SIN(GAMMA)
COS = COS(GAMMA)

TAN = 1,

IF(COSD.NE.0.) TAN = SIN/COS

ROTATE COORDINATES OF HOMING POINT ACCORDING TO AZIMUTH COMPONENT OF ATTACK ANGLE.

ALL COMPUTATIONS TO DETERMINE FUZZING POINT ARE IN ROTATED COORDINATE SYSTEM.

GRAV = GRAV
GMED = GMED

CALL ROTATE (CMED, GMED, DHAZ, 1.)

SAMPLE FROM GUIDANCE ERROR DISTRIBUTION RELATIVE TO HOMING POINT

CALL BOXND (D, H)

DMIN = SQRT((SIGH+H)**2 + (SIGH+D)**2)
GR = GRAV + SIGH+H*SINO
GD = GMED + SIGH+D
GH = GMED + SIGH+H*COS

(GR, GD, GH) IS INTERCEPT OF TRAJECTORY WITH GUIDANCE PLANE
(RF, DF, HF) WILL BE FUZZING POINT ON TRAJECTORY.
PROGRAM ARP  73/74  OPT=1

400 C
RF = GR
DF = GD
MF = GH

405 C
CHECK FOR PRIMARY FUZE FUNCTION
IBKUP = 0
IF(ROM(1).GT.PKPF) GO TO 16

410 C
CHECK FOR HEIGHT FUZING
IF(HFZ.EQ.1) GO TO 74
Q2 = 0.

415 C
CHECK FOR APPROPRIATE FUZING
CALL BOXNO (21,22)
IGO = IFUZ + 1
IF(KOOG.EQ.1) WRITE (6,5003) IFUZ,IGO,GR,GD,GH
GO TO (85,7.5,22,85),IGO

420 C
CHOOSE GLITTER POINT FOR FUZING. ANGULAR FUZE ONLY

425 C
75 IF(UGLT.LT.0.ANDF.UGRT.GT.1) GO TO 76
XGLT = NGLT
IGLT = 1.ROM(1)-0.0001)*XGLT + 1.0
IF(IGLT.EQ.0) IGLT = 1
66 RGLT = GLTR(1,IGLTS)
DGLT = GLTR(2,IGLTS)
HGLT = GLTR(3,IGLTS)
IGO = 1
GO TO 77
GO TO 76
76 IGO = NGLT
DAMAX = -1.00000.
77 GO EQ.1 = 1.0CC
IF(:.GO.EQ.1) GO TO 21
RGLT = GLTR(1,IGLTS)
DGLT = GLTR(2,IGLTS)
HGLT = GLTR(3,IGLTS)

440 C
21 IF(KOOG.EQ.1) WRITE (6,*) "RGLT,DGLT,HGLT = ",RGLT,DGLT,HGLT
C
ROTE GLITTER POINT INTO ARP COORDINATE SYSTEM
CALL ROTATE (RGLT,DGLT,DAH,1.0)

445 C
IF(KOOG.EQ.1) WRITE (6,*) "ROTE GLITTER POINT = ",
IF(KOOG.EQ.1) WRITE (6,*) "DAH,RGLT,DGLT,HGLT = ",DAH,RGLT,DGLT,HGLT
C
HGLT
5003 FORMAT (1X,IFUZ,IGO = *,2(12,*,*,1X),*,GR,GD,GH = *,3(F6.1,*,*,1X)02570)
C

450 C
USE LAW OF SINES AND LAW OF COSINES TO FIND
FUZING POINT ON TRAJECTORY. FIRST PICK A POINT
ALONG TRAJECTORY TO COMPUTE ETAK (ANGLE BETWEEN
TRAJECTORY AND A LINE (AB) FROM GLITTER POINT
(RGLT,DGLT,HGLT) TO GUIDANCE PLANE INTERCEPT
(GR,GD,GH) - NOTE THAT EVERYTHING IS IN ROTATED

455 C
COORDINATE SYSTEM (THROUGH AZIMUTH ATTACK ANGLE COMPONENT), THEN, KNOWING BETAX AND FUZING ANGLE (ANG) COMPARE ANGLE (GAMMA) WITH ITS VERTEX AT GLITTER POINT AND OPPOSITE TRAJECTORY SEGMENT BOUNDED BY GUIDANCE PLANE INTERCEPT AND FUZING POINT. FINALLY, KNOWING GAMMA, AB, AND ANG, COMPUTE O3. THE DISTANCE FROM GUIDANCE PLANE INTERCEPT TO FUZING POINT (USING THE LAW OF SINES).

TANX = TAN
IF(SING.EQ.0.) TANX = 1.
CB = 10.
IF(SING.NE.0.) CB = CB/SING

GRL, GDL, GHL ARE COORDINATES OF A POINT ON THE TRAJECTORY USED TO COMPUTE BETAX.

GRL = GR - 10./TANX
GDL = GD
GHL = GH
IF(SING.NE.0.) GHL = GH + 10.
AB = SQRT(AB2)

USE LAW OF COSINES TO COMPUTE BETAX, ANGLE WITH VERTEX AT GLITTER POINT AND OPPOSITE TRAJECTORY SEGMENT BOUNDED BY GUIDANCE PLANE INTERCEPT AND FUZING POINT.

BETAX = ACOS((AB2-BB2+CB**2)/(2.*AB*CB))
IF(NOBG.EQ.1) WRITE (6,*) *BETAX, GRL, GDL, GHL, AB, CB = *,
C BETAX, GRL, GDL, GHL, AB, CB
FZASX = FZAS
IF(ITTG.EQ.1) FZASX = 0.

ANGULAR FUZING FUNCTION

ANG = 22*FZASX + FZAM
IF(FZAM.LT.0.) ANG = FZAM + RDF(1)*(FZASX-FZAM)
IF(AANG.LT.0.1745) GO TO 16
IF(AANG.GT.1) GO TO 16

D2 IS DISTANCE ALONG TRAJECTORY FROM GUIDANCE PLANE INTERCEPT TO FUZING POINT.

GAMMA = PI - BETAX - ANG
IF GAMMA.LT.ZERO, USE SUPPLEMENT OF ANG FOR FUZING.

D2 = AB*(SIN(GAMMA)/SIN(ANG))
IF(NOBQ.EQ.1) WRITE (6,*) *D2, GAMMA, ANG = *, D2, GAMMA, ANG
IF(IDQ.EQ.1) GO TO 22
IF(O2.LT.GRMAX) GO TO 64
GRMAX = D2
ISGL = IGL
B4 CONTINUE
GO TO 66

515

**LINEAR FUZING FUNCTION (ALONG TRAJECTORY)**
FUZING DIRECTION IS POSITIVE IN THE NEGATIVE
RANGE DIRECTION, I.E., A POSITIVE CHANGE IN
THE FUZING DISTANCE, O2, IS IN THE NEGATIVE
RANGE DIRECTION.

22 IF(IITG.EQ.1) FZTS = DWLN/TAN(FZAS)
O2 = O2 + 2*FZTS + FZTM
RF = GR - O2*COS
HF = GH + O2*SINQ
DF = GD
GO TO 65

520

**BACKUP FUZING**

530

16 HF = 0.
IBKUP = 1
IF(OMEGA.EQ.0.) GO TO 5
IF(NVT.EQ.0.) GO TO 37

535

67 XK = RDM(1)
DO 65 K=1,NVT
KK = K
IF(XK.LE.VT(X)) GO TO 66
CONTINUE

540

66 HFX = VTHT(KK)
IF(HFX.LE.HF) GO TO 24

545

17 RF = GR - (HF-GM)/TAND
DF = GD
GO TO 61

550

5 WRITE (6,*) "NO BACKUP FUZING FOR OMEGA = 0."
WRITE (6,*) "TRAJECTORY CLOSEST POINT OF APPROACH TO TARGET"
WRITE (6,*) "CENTER IS USED"
RF = 0.
DF = GD
HF = GH
GO TO 61

555

**HEIGHT FUZING**

74 IF(SING.EQ.0.) SICP 74
CALL BOXNO (21,Z2)
HF = FZTM + Z1+FZTS
RF = RF + (GH-HF)/TAND

560

85 IF(OMEGA.EQ.0.) GO TO 24
IF(NVT.NE.0) GO TO 67

565

**CHECK FOR FUZING POINT BELOW GROUND**

24 IF(HF.GE.0.) GO TO 61
IF(OMEGA.EQ.0.) GO TO 61
RF = RF + HF/TAND
HF = 0.

570

**PUT BURST POINT IN TARGET COORDINATE SYSTEM FOR**
IF(BH.LT.HOM(1,1).AND.OMEGA.EQ.0.) GO TO 92
ROH1 = ROH(1,1)
ROH2 = ROH(1,2)
DHH = DHH(1,1)
DHH2 = DHH(1,2)
630
IPEN = NNUMER : SIMycled PENETRATED (MUST BE 0 OR 2)
IPEN = 0
IF(ABS(ABS(16)).EQ.90.) GO TO 102
640
DO 57 K=1,2
RDMX = RDMH
IF(K.EQ.2) RDMH = RDM2
CALL SEARCH (1,1,ROH,DA,HA)
IF(HESQ.EQ.2) WRITE (6,*) 'IPEN,RCMX,DA,HA = *,IPEN,
1 RDMX,DA,HA
57 CONTINUE
IF(IPEN.EQ.2) GO TO 92
102 IF(DATA(16).EQ.0..OR.DATA(16).EQ.180.) GO TO 108
645
DO 100 K=1,2
RDMX = DDMH
IF(K.EQ.2) DDMH = DDM2
CALL SEARCH (1,2,RA,DHH,HA)
IF(HESQ.EQ.2) WRITE (6,*) 'IPEN,RA,DDMX,HA = *,IPEN,
1 RA,DDMX,HA
100 CONTINUE
IF(IPEN.EQ.2) GO TO 92
108 IF(OMEGA.EQ.0.) GO TO 101
650
DO 117 K=1,2
DHHX = DHH1
IF(K.EQ.2) DHHX = DHH2
CALL SEARCH (1,2,RA,DA,DHHX)
IF(HESQ.EQ.2) WRITE (6,*) 'IPEN,RA,DA,DHHX = *,IPEN,
1 RA,DA,DHHX
117 CONTINUE
655
IF(IPEN.EQ.1) STOP 117
110 IF(IPEN.EQ.0) GO TO 92
660
IF(IPEN.EQ.0) GO TO 106
665
CONTINUE
670
117 CONTINUE
675
680
SET UP BURST COORDINATES (BR,BD,BH) FROM DIRECT HIT.
BR = RPN
BD = DPN
BH = PPN
0. co
4C4C
m me
I
tvv-1 qU
(Dt D (?)
or-
Ir;Ir;Ir;Ir;Ir;Ir;Ir;Ir;Ir;Ir;Ir;
1t t-t t
3.t-
II-
I-
0000000 o~
~ ~ ~
O O O O 00 0 0 0 0 0
000 00 00 00 00 00 00o oo oo oo oo oo oo oQOz
0QO
OO
451
00
451
00
101
287
5004
27
1Bx = 0
IROT = 0
IF(NSGE.4,0) WRITE (6,*);,IPN,RPN,DPN,HPN,PR,BD,BH = *, 007170
C IPN,RPN,DPN,HPN,PR,BD,BH 007180
IF(NEDR.EQ.0) GO TO 27
EDDR = BX-EDR(1)
EDDR = B3-EDDR(2)
EDDR = BH-EDDR(5)
RDRD = SORT(BADDR,BRDR+ORDR+CRDR+HRDR+HRRD)
PKDR = 1.0
IF(HDR.GT.RDR(4)) PKDR = 1. - (9DR-RDR(4))/(RDR(5)-RDR(4))
IF(HDR.GE.RDR(5)) PKDR = 0.
5004 FORMAT (1X,*EX,ED,BH = *,3(F6.1,*),1X)) 007270
27 IBX = 0 002880
IROT = 0 002890
IF(NSGE.1) WRITE (6,5004) BR,BD,BH 007300
IF(NM.EQ.0) GO TO 50 007310
C COMPUTE PK DUE TO FRAGMENTATION (PKSNAP) 007320
C COMPUTE PK DUE TO FRAGMENTATION (PKSNAP) 007330
C COMPUTE PK DUE TO FRAGMENTATION (PKSNAP) 007340
C COMPUTE PK DUE TO FRAGMENTATION (PKSNAP) 007350
C INTERPOLATE IN RANGE, DEFLECTION, HEIGHT & ANGLE TO GET FRAGMENTATION PK FROM PK GRIDS. 007360
C INTERPOLATE IN RANGE, DEFLECTION, HEIGHT & ANGLE TO GET FRAGMENTATION PK FROM PK GRID. 007370
C INTERPOLATE IN RANGE, DEFLECTION, HEIGHT & ANGLE TO GET FRAGMENTATION PK FROM PK GRID. 007380
C INTERPOLATE IN RANGE, DEFLECTION, HEIGHT & ANGLE TO GET FRAGMENTATION PK FROM PK GRID. 007390
C IF = 1 IF(BH.GT.HC MH+1)) GO TO 50 007400
C IF = 1 IF(BH.GT.HC MH+1)) GO TO 50 007410
C IF = 1 IF(BH.GT.HC MH+1)) GO TO 50 007420
C INTERPOLATE PT FOR FRAGMENTATION PK INTERPOLATION INTO ARP COORDINATE SYSTEM. 007430
C RECALL THAT PK GRIDS ARE IN PROJECTILE COORDINATE SYSTEM. 007440
C RECALL THAT PK GRIDS ARE IN PROJECTILE COORDINATE SYSTEM. 007450
C RECALL THAT PK GRIDS ARE IN PROJECTILE COORDINATE SYSTEM. 007460
C CALL ROTATE (BR,SD,CHAZ) 1.) IROT = 1 007470
C CALL ROTATE (BR,SD,CHAZ,1.) IROT = 1 007480
C CALL ROTATE (BR,SD,CHAZ,1.) IROT = 1 007490
C LOCATE HEIGHT BOUNDARIES 007500
C
DO 20 I=1,MH
   IM2 = I
   IF(BH.LE.HGT(I)) GO TO 25
   CONTINUE
   IM2 = 0
25 IM1 = IM2 - 1
   IF(IM1.EQ.0) IM1 = 1
   IF(IM1.LT.0) IM1 = NH
   IF(NDBG.EQ.4) WRITE (6,*),'IM1,IM2,NR,ND,RU,DU,BR,BD,BH=',IM1,IM2,NR,ND,RU,DU,BR,BD,BH
C IM1,IM2,NR,ND,RU,DU,BR,BD,BH
C NDBG)
      007590
50 PKSAMP = 0.
      007660
C
60 COMPUTE SPHERICAL COORDINATES TO BURST POINT (BR, BD, BH)
C SAI = ANGLE OFF POSITIVE RANGE AXIS MEASURED
C CLOCKWISE
C SAZ = ANGLE OFF R-O PLANE MEASURED TOWARD POSITIVE
C SR = RANGE FROM BURST POINT TO (0,0,0)
C H-AXIS IN VERTICAL PLANE
C
   007700
C
   007710
C
   007720
C
   007730
C
   007740
C
   007750
C
   007760
C
   007770
C
   007780
C
   007790
C
   007800
C
   007810
C
   007820
C
   007830
C
   007840
C
   007850
C
   007860
C
   007870
C
   007880
C
   007890
C
   007900
C
   007910
C
   007920
C
   007930
C
   007940
C
   007950
C
   007960
C
   007970
C
   007980
C
   007990
C
   008000
C
   008010
C
   008020
C
   008030
C
   008040
C
   008050
C
   008060
C
   008070
PROGRAM ARP 73/74 C:T=1 F8N 4.8+503 03/13/81 08.29.30 PAGE 15

57 CONTINUE
58 GO TO 98 I=1,6
ISA2 = 1
IF(ISA2.LT.BETA(I+1)) GO TO 99
98 CONTINUE
99 SR = SQRT(BR*BR + BD*BD + BM*BM)
ISR = 0
DO 100 I=1,10
II = I
IF(1.E-10) II = 11
ISR = ISR + 1
IF(SR.LT.RANGE(I)) GO TO 110
100 CONTINUE
110 IF(NDBG.EQ.6) WRITE (6,*) *ISA1,ISA2,ISR = *,ISA1,ISA2,ISR
IF(NDBG.EQ.6) WRITE (6,*) *ISA1,ISA2,SR = *,ISA1,ISA2,SR
C STORE PK'S ACCORDING TO SPHERICAL COORDINATES
C IKS(ISA1,ISA2,ISR) = IKS(ISA1,ISA2,ISR) + 1
C SUM PK'S OVER ALL SAMPLES
820 IF(NDBG.GT.0) WRITE (6,*) *PKR,PKR,PKD,PKS = *,PKSAMP,PKRAD,PKDH
C PKBLST
PKBASE = PKBASE + PKSAMP
PKRAD = PKRAD + PKD0
PKBLT = PKBLT + PKD0
PKSL = PKSL + PKSAMP
PKSAMP = 1. - (1.-PKSAMP)*(1.-PKRAD)*(1.-PKDH)*(1.-PKBLST)
PKS(ISA1,ISA2,ISR) = PKS(ISA1,ISA2,ISR) + PKSAMP
PKTOT = PKTOT + PKSAMP
PKTOT2 = PKTOT2 + PKSAMP
IF(NDBG.GE.1) WRITE (6,3003) PKSAMP
3003 FORMAT (5X,SAMPLE PK = *,F6.4)
IF(NFR1.EQ.1) GO TO 1
IF(MOD(ISIM,10).NE.0) GO TO 1
PKNRT = ISIM
PKRAD = PKRAD/PKSNRT
WRITE (6,*) "NO. SIMULATIONS, PK = *,ISIM,PKNRT"
GO TO 1
18 NCT = NCT + 1
C CONTINUE
C DISPLAY FINAL RESULTS
C IF(NFR1.GT.0) GO TO 79
C WRITE (6,2002)
B645 WRITE (6,*) "FINAL RESULTS"
3000 FORMAT (/1X,*PK = *,F6.4,2X,*PKSD = *,F6.4,2X,*NSAMP = *,F6.16/)
79 XSAMP = NSAMP
PKBAR = PKTOT/XSAMP
PKBASE = PKBASE/XSAMP
PKRAD = PKRAD/XSAMP
PKDHT = PKDHT/XSAMP
PKSL = PKSL/XSAMP
PK(ILL) = PKBASE
PK(ILL) = PKRAD
PROGRAM ARP 73/74 CPT=1

PKD(ILUP) = PKDHT
PKG(ILUP) = PKBAR
PKBL(ILUP) = PKBLT
XSAMP = NSMP - NCT

IF(NSMP.EQ.NCT) XSAMP = 1.
XSAMP = XSAMP-1.

BRG(ILUP) = BRBAR/XSAMP
BDG(ILUP) = BDAR/XSAMP
BHSG(ILUP) = BHSAR/XSAMP
RRG(ILUP) = RRRBAR/XSAMP
BSG(ILUP) = SQRT((BRBAR2 + XSAMP*BRG(ILUP)*ERG(ILUP))/XSMP)
EDSG(ILUP) = SQRT((BDAR2 + XSAMP*BDG(ILUP))/XSMP)
BHSG(ILUP) = SQRT((BHSAR2 + XSAMP)/XSMP)
RRG(ILUP) = SQRT((RRRBAR2 + XSAMP)/XSMP)

IF(NCT.NE.0) WRITE (6,2004) NCT,NSMP,CNP(ILUP)
2004 FORMAT (2X,*SIMULATION OR FUZING DUDS = *,I4,* OUT OF *,I4,
C = SIMULATION, 2X,*GUIDANCE CEP = *,F6.2)

C
C
C
C
C
C
C
C

C

PK VS R, ALPHA, BETA, WHERE ALPHA IS AZIMUTH ANGLE

MEASURED FROM POSITIVE RANGE AXIS TOWARD POSITIVE

DEFLECTION AXIS (0 TO 360), BETA IS ELEVATION ANGLE

MEASURED FROM NEGATIVE HEIGHT AXIS TO POSITIVE

HEIGHT AXIS (0 TO 90).

WRITE (6,2001)
WRITE (6,*) "PK VS R, ALPHA, BETA":

WRITE (6,*): "---" ALPHA BETA"

DO 49 I=1,10
DO 49 J=1,10
DO 49 K=1,10
IF(IKS(J,K,I).EQ.0) GO TO 49
X0KS = IKS(J,K,I)
PKS(J,K,I) = PKS(J,K,I)/X0KS
CONTINUE

DO 45 I=1,10
DO 45 J=1,12
DO 47 J=1,12
DO 47 K=1,12
RPK = PKS(J,K,I)
X0KS = IKS(J,K,I)
XI = XI + X0KS
RSUM(I) = RSUM(I) + X0KS
RSUM(I) = RSUM(I) + X0KS

IF(RPK.GT.0.0) WRITE (6,3004) RPK,RANG,ALPHA(J),ALPHA(J+1),BETA(K)
3004 FORMAT (1X,F6.4,2X,F5.1,2(2X,F6.1,* = *,F6.1))
PROGRAM ARP  73/74  DPT=1  FTN 4.6-50B  03/13/81  08:29:30  PAGE  17

47 CONTINUE  
IF(X1.EQ.0.) GO TO 45  
RSUM(I) = RSUM(I)/XI  
45 CONTINUE  
WRITE (6,2002)  
WRITE (6,*) "AVG PK VS. R"  
WRITE (6,*) "-------------"  
920  
DO 43 I=1,10  
R = RANGE(I)  
IF(RSUM(I).EQ.0.) GO TO 43  
WRITE (6,3001) RSUM(I),R  
43 CONTINUE  
925  
2001 FORMAT (1X,F6.4,4X,F5.1)  
WRITE (6,2002)  
C  
CHECK FOR ANOTHER CASE  
44 WRITE (6,2001)  
930  
69 CONTINUE  
C  
DISPLAY RESULTS FOR EACH GUIDANCE ERROR  
IF(NPRT.GT.0) WRITE (6,2002)  
935  
FZTM = DATA(2)  
OMEGD = DATA(7)  
OMGOD = DATA(19)  
FZASD = DATA(5)  
940  
WRITE (6,2006) OMEG0D,OMGOD,FZASD,FZTM,FZTS,DHAZ,NAMP  
2006 FORMAT (/5X,*RESULTS FOR FOLLOWING CONDITIONS *,/,
C10X*,ITEM*,13X*,MEAN*,4X*,STD DEVI*,/,,
C10X*,ELEVATION*,4X*,2F10.4*,/,,10X*,FUZE ANGLE*,3X*,2F10.4*,/,,
C10X*,LINEAR FUZE*,2X*,2F10.4*,/,,10X*,AZIMUTH*,,F10.4*,/,,
945  
C10X*,SAMPLE SIZE *,15X,/
WRITE (6,2003) GDR,GRD,GMH  
WRITE (6,2002)  
950  
2012 FORMAT (/5X,*ERROR DATA*,17X,*PK*,3X,  
C=PKFRA, PKMAD, PKMHT, PKBLST*)  
955  
2005 FORMAT (5X,*HOMEING POINT: COORDINATES (R,D,H) = *,
C4(F6.1,*),*,F6.1)  
DO 72 I=1,NLOOP  
IF(NCEP.EQ.0) WRITE (6,2007) SDD(I),SDH(I),PKG(I)  
C,PK(I),PKR(I),PKD(I),PKBL(I)  
72 CONTINUE  
WRITE (6,2002)  
WRITE (6,1003)  
DO 26 I=1,N LOOP  
26 WRITE (6,1004) CEP(I),RSG(I),RSG(I),RSG(I),RSG(I),RSG(I),RSG(I)09700  
C,BRG(I),BRG(I),BRG(I),BRG(I),BRG(I),BRG(I)09710  
1003 FORMAT (/5X,*BURST STATISTICS (MEAN, STD DEVIATION)*,,/1X*,CEP*,
C4X*,BURST RANGE*,7X*,RANGE*,8X*,DEFLECTION*,7X*,HEIGHT*)  
1004 FORMAT (1X,F4.1,4X,(2X,F6.2,1X,F6.2))  
2007 FORMAT (5X,*SD (0,H) = *,
C2(F4.1,*,*),1X,5F7.4)  
2008 FORMAT (5X,*CEP = *,F4.1,  
C14X,5F7.4)
## SYMBOLIC REFERENCE MAP (R=2)

<table>
<thead>
<tr>
<th>ENTRY POINTS</th>
<th>DEF LINE</th>
<th>REFERENCES</th>
</tr>
</thead>
<tbody>
<tr>
<td>3505 ARP</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VARIABLES</th>
<th>SN</th>
<th>TYPE</th>
<th>RELLOCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>10232 A</td>
<td></td>
<td>REAL</td>
<td>199 199 DEFINED 137</td>
</tr>
<tr>
<td>10227 AAAA</td>
<td></td>
<td>REAL</td>
<td>168 168</td>
</tr>
<tr>
<td>10373 AB</td>
<td></td>
<td>REAL</td>
<td>486 487 507 DEFINED 480</td>
</tr>
<tr>
<td>10371 AB2</td>
<td></td>
<td>REAL</td>
<td>480 495 506 DEFINED 494</td>
</tr>
<tr>
<td>31267 ALPHA</td>
<td></td>
<td>REAL ARRAY</td>
<td>7 790 2+910 DEFINED 17</td>
</tr>
<tr>
<td>10435 AKAM</td>
<td></td>
<td>REAL ARRAY</td>
<td>3 168 175 182 221 229</td>
</tr>
<tr>
<td>10376 ANG</td>
<td></td>
<td>REAL</td>
<td>456 497 502 506 507 508</td>
</tr>
<tr>
<td>10217 ANS</td>
<td></td>
<td>REAL</td>
<td>55 515 190 208 887 972</td>
</tr>
<tr>
<td>10233 B</td>
<td></td>
<td>REAL</td>
<td>460 495 506 DEFINED 494</td>
</tr>
<tr>
<td>132 BOLST</td>
<td></td>
<td>REAL ARRAY RDRT</td>
<td>15 584 DEFINED 272</td>
</tr>
<tr>
<td>10372 BB2</td>
<td></td>
<td>REAL</td>
<td>486 DEFINED 479</td>
</tr>
<tr>
<td>6 BD</td>
<td></td>
<td>REAL</td>
<td>13 577 605 610 614 2+826 697</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2+702 708 712 721 730 751 753 772</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2+774 776 781 790 792 2+792 2+804</td>
</tr>
<tr>
<td>10313 BDBAR</td>
<td></td>
<td>REAL</td>
<td>781 864 DEFINED 343 781</td>
</tr>
<tr>
<td>10314 BDBAR2</td>
<td></td>
<td>REAL</td>
<td>792 866 DEFINED 344 782</td>
</tr>
<tr>
<td>10426 BDD</td>
<td></td>
<td>RFAL</td>
<td>776 782 DEFINED 774</td>
</tr>
<tr>
<td>31542 BDG</td>
<td></td>
<td>REAL ARRAY</td>
<td>11 2+868 951 DEFINED 864</td>
</tr>
<tr>
<td>31564 BDG2</td>
<td></td>
<td>REAL ARRAY</td>
<td>11 561 DEFINED 868</td>
</tr>
<tr>
<td>31304 BETA</td>
<td></td>
<td>REAL ARRAY</td>
<td>7 602 2+810 DEFINED 19</td>
</tr>
<tr>
<td>10374 BETAX</td>
<td></td>
<td>REAL</td>
<td>487 502 DEFINED 486</td>
</tr>
<tr>
<td>7 BH</td>
<td></td>
<td>REAL</td>
<td>13 577 607 611 615 627 628</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>645 696 708 713 721 721 721 721</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>745 751 753 2+775 778 783 793 2+804</td>
</tr>
<tr>
<td>10317 BHBAR</td>
<td></td>
<td>REAL</td>
<td>783 865 DEFINED 347 783</td>
</tr>
<tr>
<td>10325 BHBAR2</td>
<td></td>
<td>REAL</td>
<td>784 869 DEFINED 348 784</td>
</tr>
<tr>
<td>31566 BHH</td>
<td></td>
<td>REAL ARRAY</td>
<td>12 2+869 951 DEFINED 865</td>
</tr>
<tr>
<td>10427 BHM</td>
<td></td>
<td>REAL</td>
<td>776 784 DEFINED 775</td>
</tr>
<tr>
<td>31600 BHS5G</td>
<td></td>
<td>REAL ARRAY</td>
<td>12 601 DEFINED 869</td>
</tr>
<tr>
<td>10286 BLST</td>
<td></td>
<td>REAL</td>
<td>271 272 275 695 697 698 703</td>
</tr>
<tr>
<td>5 BR</td>
<td></td>
<td>REAL</td>
<td>13 577 605 609 613 624 687</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>606 2+702 708 711 721 721 721 721</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>772 2+773 778 789 790 792 2+793</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2+804 DEFINED 574 682 687</td>
</tr>
<tr>
<td>VARIABLES</td>
<td>SN</td>
<td>TYPE</td>
<td>RELOCATION</td>
</tr>
<tr>
<td>-----------</td>
<td>----</td>
<td>------</td>
<td>------------</td>
</tr>
<tr>
<td>10315</td>
<td>B3R</td>
<td>REAL</td>
<td>REF</td>
</tr>
<tr>
<td>10316</td>
<td>B3BAR2</td>
<td>REAL</td>
<td>REF</td>
</tr>
<tr>
<td>10413</td>
<td>BDR</td>
<td>REAL</td>
<td>REF</td>
</tr>
<tr>
<td>31516</td>
<td>BRG</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>10425</td>
<td>BR</td>
<td>REAL</td>
<td>REF</td>
</tr>
<tr>
<td>10350</td>
<td>BRS</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>10365</td>
<td>CB</td>
<td>REAL</td>
<td>REF</td>
</tr>
<tr>
<td>31325</td>
<td>CEP</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>10334</td>
<td>CEG</td>
<td>REAL</td>
<td>REF</td>
</tr>
<tr>
<td>10404</td>
<td>DA</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>10537</td>
<td>DATA</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>3</td>
<td>DBS</td>
<td>REAL</td>
<td>SRCH</td>
</tr>
<tr>
<td>31446</td>
<td>DDH</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>10406</td>
<td>DOH</td>
<td>REAL</td>
<td>SRCH</td>
</tr>
<tr>
<td>16</td>
<td>DOH</td>
<td>REAL</td>
<td>SRCH</td>
</tr>
<tr>
<td>17</td>
<td>DOL</td>
<td>REAL</td>
<td>SRCH</td>
</tr>
<tr>
<td>10347</td>
<td>DF</td>
<td>REAL</td>
<td>REF</td>
</tr>
<tr>
<td>10357</td>
<td>DGLT</td>
<td>REAL</td>
<td>DEFINED</td>
</tr>
<tr>
<td>30725</td>
<td>DGRD</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>10252</td>
<td>DHAZ</td>
<td>REAL</td>
<td>REF</td>
</tr>
<tr>
<td>10412</td>
<td>DIST</td>
<td>REAL</td>
<td>REF</td>
</tr>
<tr>
<td>10275</td>
<td>DL</td>
<td>REAL</td>
<td>REF</td>
</tr>
<tr>
<td>10342</td>
<td>DMIN</td>
<td>REAL</td>
<td>REF</td>
</tr>
<tr>
<td>12</td>
<td>DPN</td>
<td>REAL</td>
<td>SRCH</td>
</tr>
<tr>
<td>10414</td>
<td>DRDR</td>
<td>REAL</td>
<td>REF</td>
</tr>
<tr>
<td>10215</td>
<td>DTE</td>
<td>REAL</td>
<td>REF</td>
</tr>
<tr>
<td>145</td>
<td>DU</td>
<td>REAL</td>
<td>RDWRT</td>
</tr>
<tr>
<td>10265</td>
<td>DUR</td>
<td>REAL</td>
<td>RDWRT</td>
</tr>
<tr>
<td>6522</td>
<td>END</td>
<td>REAL</td>
<td>REF</td>
</tr>
<tr>
<td>10234</td>
<td>FZAM</td>
<td>REAL</td>
<td>DEFINED</td>
</tr>
<tr>
<td>10453</td>
<td>FZAMD</td>
<td>REAL</td>
<td>REF</td>
</tr>
<tr>
<td>10240</td>
<td>FZAS</td>
<td>REAL</td>
<td>REF</td>
</tr>
<tr>
<td>10454</td>
<td>FZASD</td>
<td>REAL</td>
<td>REF</td>
</tr>
<tr>
<td>10375</td>
<td>FZASX</td>
<td>REAL</td>
<td>REF</td>
</tr>
<tr>
<td>10235</td>
<td>FZTM</td>
<td>REAL</td>
<td>REF</td>
</tr>
<tr>
<td>10242</td>
<td>FZTS</td>
<td>REAL</td>
<td>REF</td>
</tr>
<tr>
<td>10377</td>
<td>GAMMA</td>
<td>REAL</td>
<td>REF</td>
</tr>
<tr>
<td>10344</td>
<td>GD</td>
<td>REAL</td>
<td>DEFINED</td>
</tr>
<tr>
<td>10367</td>
<td>GOL</td>
<td>REAL</td>
<td>REF</td>
</tr>
<tr>
<td>10345</td>
<td>GH</td>
<td>REAL</td>
<td>REF</td>
</tr>
<tr>
<td>10370</td>
<td>GHL</td>
<td>REAL</td>
<td>REF</td>
</tr>
<tr>
<td>21</td>
<td>GLTR</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>65</td>
<td>GMO</td>
<td>REAL</td>
<td>RDWRT</td>
</tr>
<tr>
<td>VARIABLES</td>
<td>SN TYPE</td>
<td>RELOCATION</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>---------</td>
<td>------------</td>
<td></td>
</tr>
<tr>
<td>10337 GMDR</td>
<td>REAL</td>
<td>REFs 366</td>
<td>394</td>
</tr>
<tr>
<td>66 GMRH</td>
<td>REAL</td>
<td>REFs 355</td>
<td>395</td>
</tr>
<tr>
<td>64 GMR</td>
<td>REAL</td>
<td>REFs 355</td>
<td>395</td>
</tr>
<tr>
<td>10336 GMRR</td>
<td>REAL</td>
<td>REFs 386</td>
<td>353</td>
</tr>
<tr>
<td>10343 GR</td>
<td>REAL</td>
<td>REFs 401</td>
<td>419</td>
</tr>
<tr>
<td>10336 GRL</td>
<td>REAL</td>
<td>REFs 479</td>
<td>487</td>
</tr>
<tr>
<td>10332 GRMAX</td>
<td>REAL</td>
<td>REFs 510</td>
<td>DEFINED</td>
</tr>
<tr>
<td>10341 H</td>
<td>REAL</td>
<td>REFs 381</td>
<td>392</td>
</tr>
<tr>
<td>10405 HAI</td>
<td>ARRAY</td>
<td>REFs 15</td>
<td>662</td>
</tr>
<tr>
<td>137 HBLST</td>
<td>REAL</td>
<td>REFs 13</td>
<td>620</td>
</tr>
<tr>
<td>31450 HDM</td>
<td>REAL</td>
<td>REFs 10</td>
<td>175</td>
</tr>
<tr>
<td>10410 HMXH</td>
<td>REAL</td>
<td>REFs 670</td>
<td>671</td>
</tr>
<tr>
<td>20 HDDH</td>
<td>REAL</td>
<td>REFs 13</td>
<td>668</td>
</tr>
<tr>
<td>21 HDD2</td>
<td>REAL</td>
<td>REFs 13</td>
<td>668</td>
</tr>
<tr>
<td>10350 HF</td>
<td>REAL</td>
<td>REFs 541</td>
<td>543</td>
</tr>
<tr>
<td>10422 HFX</td>
<td>REAL</td>
<td>REFs 541</td>
<td>542</td>
</tr>
<tr>
<td>10350 HGTT</td>
<td>REAL</td>
<td>REFs 440</td>
<td>448</td>
</tr>
<tr>
<td>10 HGT</td>
<td>REAL</td>
<td>REFs 6</td>
<td>15</td>
</tr>
<tr>
<td>13 HPN</td>
<td>REAL</td>
<td>REFs 13</td>
<td>684</td>
</tr>
<tr>
<td>10415 HRDR</td>
<td>REAL</td>
<td>REFs 2*174</td>
<td>DEFINED</td>
</tr>
<tr>
<td>10286 I</td>
<td>INTEGER</td>
<td>REFs 155</td>
<td>121</td>
</tr>
<tr>
<td>291</td>
<td></td>
<td>292</td>
<td>293</td>
</tr>
<tr>
<td>2*312</td>
<td></td>
<td>326</td>
<td>329</td>
</tr>
<tr>
<td>2*626</td>
<td></td>
<td>627</td>
<td>629</td>
</tr>
<tr>
<td>634</td>
<td></td>
<td>646</td>
<td>656</td>
</tr>
<tr>
<td>745</td>
<td></td>
<td>759</td>
<td>798</td>
</tr>
<tr>
<td>857</td>
<td></td>
<td>2*898</td>
<td>902</td>
</tr>
<tr>
<td>921</td>
<td></td>
<td>922</td>
<td>923</td>
</tr>
<tr>
<td>DEFINED</td>
<td>157</td>
<td>180</td>
<td>196</td>
</tr>
<tr>
<td>300</td>
<td></td>
<td>305</td>
<td>308</td>
</tr>
<tr>
<td>694</td>
<td></td>
<td>743</td>
<td>795</td>
</tr>
<tr>
<td>952</td>
<td></td>
<td>960</td>
<td>962</td>
</tr>
<tr>
<td>10351 IBKUP</td>
<td>INTEGER</td>
<td>REFs 407</td>
<td>522</td>
</tr>
<tr>
<td>10411 IBLST</td>
<td>INTEGER</td>
<td>REFs 696</td>
<td>697</td>
</tr>
<tr>
<td>10417 IBK</td>
<td>INTEGER</td>
<td>REFs 719</td>
<td>720</td>
</tr>
<tr>
<td>113 IDAT</td>
<td>ARRAY</td>
<td>REFs 15</td>
<td>DEFINED</td>
</tr>
<tr>
<td>10361 IDQ</td>
<td>INTEGER</td>
<td>REFs 435</td>
<td>436</td>
</tr>
<tr>
<td>10246 IUF</td>
<td>INTEGER</td>
<td>REFs 251</td>
<td>DEFINED</td>
</tr>
<tr>
<td>10272 IFUZ</td>
<td>INTEGER</td>
<td>REFs 418</td>
<td>419</td>
</tr>
<tr>
<td>10363 IGL</td>
<td>INTEGER</td>
<td>REFs 437</td>
<td>438</td>
</tr>
<tr>
<td>10355 IGLT</td>
<td>INTEGER</td>
<td>REFs 427</td>
<td>428</td>
</tr>
<tr>
<td>512</td>
<td></td>
<td>419</td>
<td>420</td>
</tr>
<tr>
<td>10353 IGO</td>
<td>INTEGER</td>
<td>REFs 412</td>
<td>DEFINED</td>
</tr>
<tr>
<td>10270 IHPZ</td>
<td>INTEGER</td>
<td>REFs 748</td>
<td>759</td>
</tr>
<tr>
<td>10423 IH1</td>
<td>INTEGER</td>
<td>REFs 748</td>
<td>751</td>
</tr>
<tr>
<td>10422 IH2</td>
<td>INTEGER</td>
<td>REFs 810</td>
<td>730</td>
</tr>
<tr>
<td>30143 IMK</td>
<td>ARRAY</td>
<td>REFs 5</td>
<td>617</td>
</tr>
<tr>
<td>26623 IKS</td>
<td>ARRAY</td>
<td>REFs 5</td>
<td>617</td>
</tr>
<tr>
<td>DEFINED</td>
<td>333</td>
<td>352</td>
<td>654</td>
</tr>
<tr>
<td>PROGRAM ARB</td>
<td>73/74</td>
<td>OPT+1</td>
<td>FTN 4.8+50B</td>
</tr>
<tr>
<td>------------</td>
<td>-------</td>
<td>-------</td>
<td>------------</td>
</tr>
<tr>
<td>VARIABLES</td>
<td>SN</td>
<td>TYPE</td>
<td>RELLOCATION</td>
</tr>
<tr>
<td>10401</td>
<td>KK</td>
<td>INTEGER</td>
<td>663</td>
</tr>
<tr>
<td>10254</td>
<td>NA</td>
<td>INTEGER</td>
<td>667</td>
</tr>
<tr>
<td>10244</td>
<td>NGER</td>
<td>INTEGER</td>
<td>310</td>
</tr>
<tr>
<td>10253</td>
<td>NH</td>
<td>INTEGER</td>
<td>639</td>
</tr>
<tr>
<td>10251</td>
<td>NDRR</td>
<td>INTEGER</td>
<td>319</td>
</tr>
<tr>
<td>10257</td>
<td>NVT</td>
<td>INTEGER</td>
<td>253</td>
</tr>
<tr>
<td>10243</td>
<td>OMEG</td>
<td>REAL</td>
<td>371</td>
</tr>
<tr>
<td>10</td>
<td>OMEGA</td>
<td>REAL</td>
<td>375</td>
</tr>
<tr>
<td>10451</td>
<td>OMEGD</td>
<td>REAL</td>
<td>940</td>
</tr>
<tr>
<td>10255</td>
<td>OMGS</td>
<td>REAL</td>
<td>371</td>
</tr>
<tr>
<td>10452</td>
<td>OMGSD</td>
<td>REAL</td>
<td>940</td>
</tr>
<tr>
<td>VARIABLES</td>
<td>SN</td>
<td>TYPE</td>
<td>RELOCATION</td>
</tr>
<tr>
<td>-----------</td>
<td>----</td>
<td>------</td>
<td>------------</td>
</tr>
<tr>
<td>10225</td>
<td>P1</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>31364</td>
<td>PK</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>10424</td>
<td>PKA</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>10442</td>
<td>PKBAR</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>10305</td>
<td>PKASE</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>31422</td>
<td>PKPSMT</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>10306</td>
<td>PKBLSI</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>10226</td>
<td>PKDLX</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>31410</td>
<td>PKD</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>10326</td>
<td>PKDH</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>10304</td>
<td>PKDMTH</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>10226</td>
<td>PKDMLX</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>31313</td>
<td>PKG</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>28541</td>
<td>PKM</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>10250</td>
<td>PKMF</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>10420</td>
<td>PKMRNT</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>31376</td>
<td>PKR</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>10303</td>
<td>PKRARD</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>10300</td>
<td>PKROR</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>25221</td>
<td>PKS</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>609</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10325</td>
<td>PKSAMP</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>10444</td>
<td>PKSD</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>10307</td>
<td>PKTOK</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>10621</td>
<td>PK1</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>67</td>
<td>PVT</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>10495</td>
<td>R</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>10407</td>
<td>RA</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>10446</td>
<td>RANG</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>31337</td>
<td>RANGE</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>2</td>
<td>RBS</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>31434</td>
<td>RHDI</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>10403</td>
<td>RDHX</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>RHD1</td>
<td>REAL</td>
<td>SRCH</td>
</tr>
<tr>
<td>15</td>
<td>RHD2</td>
<td>REAL</td>
<td>SRCH</td>
</tr>
<tr>
<td>125</td>
<td>RDR</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>10346</td>
<td>RF</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>10356</td>
<td>RGLT</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>30225</td>
<td>RGDR</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>10447</td>
<td>RPK</td>
<td>REAL</td>
<td>SRCH</td>
</tr>
<tr>
<td>11</td>
<td>RN</td>
<td>REAL</td>
<td>SRCH</td>
</tr>
<tr>
<td>10431</td>
<td>RR</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>10311</td>
<td>RRBAR</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>10212</td>
<td>RRBAR2</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>10415</td>
<td>RR</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>31472</td>
<td>RRG</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>10274</td>
<td>RRG</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>VARIABLES</td>
<td>SN</td>
<td>TYPE</td>
<td>RELOCATION</td>
</tr>
<tr>
<td>-----------</td>
<td>----</td>
<td>------</td>
<td>------------</td>
</tr>
<tr>
<td>10430</td>
<td>RRR</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>31504</td>
<td>RRS</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>31352</td>
<td>RSM</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>144</td>
<td>RU</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>10432</td>
<td>SA1</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>10433</td>
<td>SA2</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>101</td>
<td>SDD</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>10321</td>
<td>SCD</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>10322</td>
<td>SIGH</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>10333</td>
<td>SINO</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>10436</td>
<td>SR</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>10262</td>
<td>SPR</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>10335</td>
<td>TANO</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>10364</td>
<td>TANOX</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>10254</td>
<td>TGTC</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>10216</td>
<td>TME</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>10230</td>
<td>VALUE</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>10354</td>
<td>XALT</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>10276</td>
<td>XI</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>10445</td>
<td>XIXS</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>10400</td>
<td>XK</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>XDMG</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>10273</td>
<td>XRNG</td>
<td>REAL</td>
<td>ARRAY</td>
</tr>
<tr>
<td>10441</td>
<td>XSAM</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>6521</td>
<td>YES</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>10331</td>
<td>Z1</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>10332</td>
<td>Z2</td>
<td>REAL</td>
<td></td>
</tr>
</tbody>
</table>

**FILE NAMES**
- **MODE**
  - 0 INPUT
  - 410 OUTPUT

**WRITEs and READs**
<table>
<thead>
<tr>
<th>TAPE</th>
<th>WRITEs</th>
<th>READs</th>
<th>MOTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1020</td>
<td>162</td>
<td>184</td>
<td>197</td>
</tr>
<tr>
<td>1430</td>
<td>186</td>
<td>203</td>
<td>153</td>
</tr>
<tr>
<td>2040</td>
<td>154</td>
<td>154</td>
<td>154</td>
</tr>
<tr>
<td>2450</td>
<td>182</td>
<td>182</td>
<td>182</td>
</tr>
<tr>
<td>0 TAPes</td>
<td>54</td>
<td>54</td>
<td>54</td>
</tr>
</tbody>
</table>

**WRITEs and READs**
<table>
<thead>
<tr>
<th>TAPE</th>
<th>WRITEs</th>
<th>READs</th>
<th>MOTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>410 TAPE</td>
<td>41</td>
<td>41</td>
<td>41</td>
</tr>
<tr>
<td>60</td>
<td>48</td>
<td>52</td>
<td>55</td>
</tr>
<tr>
<td>69</td>
<td>71</td>
<td>71</td>
<td>71</td>
</tr>
<tr>
<td>78</td>
<td>80</td>
<td>81</td>
<td>82</td>
</tr>
<tr>
<td>87</td>
<td>88</td>
<td>89</td>
<td>90</td>
</tr>
<tr>
<td>96</td>
<td>97</td>
<td>98</td>
<td>99</td>
</tr>
<tr>
<td>PROGRAM ARP</td>
<td>73/74</td>
<td>OPT=1</td>
<td>FTN 4.8*508</td>
</tr>
<tr>
<td>------------</td>
<td>-------</td>
<td>-------</td>
<td>------------</td>
</tr>
<tr>
<td>FILE NAMES</td>
<td>MODE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>105</td>
<td>107</td>
</tr>
<tr>
<td></td>
<td></td>
<td>115</td>
<td>116</td>
</tr>
<tr>
<td></td>
<td></td>
<td>124</td>
<td>125</td>
</tr>
<tr>
<td></td>
<td></td>
<td>141</td>
<td>147</td>
</tr>
<tr>
<td></td>
<td></td>
<td>201</td>
<td>206</td>
</tr>
<tr>
<td></td>
<td></td>
<td>407</td>
<td>508</td>
</tr>
<tr>
<td></td>
<td></td>
<td>622</td>
<td>647</td>
</tr>
<tr>
<td></td>
<td></td>
<td>813</td>
<td>821</td>
</tr>
<tr>
<td></td>
<td></td>
<td>880</td>
<td>690</td>
</tr>
<tr>
<td></td>
<td></td>
<td>926</td>
<td>929</td>
</tr>
<tr>
<td></td>
<td></td>
<td>959</td>
<td>961</td>
</tr>
</tbody>
</table>

3060 TAPED FREE WRITES 778 MOTION 29

VARIABLES USED AS FILE NAMES, SEE ABOVE

<table>
<thead>
<tr>
<th>EXTERNALS</th>
<th>TYPE</th>
<th>ARGS</th>
<th>REFERENCES</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACOS</td>
<td>REAL</td>
<td>1 LIBRARY</td>
<td>486</td>
</tr>
<tr>
<td>ALOG</td>
<td>REAL</td>
<td>1 LIBRARY</td>
<td>269</td>
</tr>
<tr>
<td>ATAN</td>
<td>REAL</td>
<td>1 LIBRARY</td>
<td>793</td>
</tr>
<tr>
<td>ATAN2</td>
<td>REAL</td>
<td>2 LIBRARY</td>
<td>790</td>
</tr>
<tr>
<td>BLAST</td>
<td>REAL</td>
<td>5</td>
<td>696 697</td>
</tr>
<tr>
<td>BOXNG</td>
<td>REAL</td>
<td>2</td>
<td>370 391 417 557</td>
</tr>
<tr>
<td>CONN</td>
<td>REAL</td>
<td>1</td>
<td>27</td>
</tr>
<tr>
<td>COS</td>
<td>REAL</td>
<td>1 LIBRARY</td>
<td>273</td>
</tr>
<tr>
<td>DATE</td>
<td>REAL</td>
<td>1</td>
<td>46</td>
</tr>
<tr>
<td>EXP</td>
<td>REAL</td>
<td>1 LIBRARY</td>
<td>302</td>
</tr>
<tr>
<td>GRIDS</td>
<td>REAL</td>
<td>8</td>
<td>319</td>
</tr>
<tr>
<td>INTERP</td>
<td>REAL</td>
<td>16</td>
<td>253</td>
</tr>
<tr>
<td>ROM</td>
<td>REAL</td>
<td>1</td>
<td>358 366 408 426 495 535</td>
</tr>
<tr>
<td>RMDIN</td>
<td>REAL</td>
<td>1</td>
<td>134</td>
</tr>
<tr>
<td>RMDOUT</td>
<td>REAL</td>
<td>1</td>
<td>132</td>
</tr>
<tr>
<td>READ</td>
<td>REAL</td>
<td>8</td>
<td>175 229</td>
</tr>
<tr>
<td>ROTATE</td>
<td>REAL</td>
<td>4</td>
<td>366 444 573 738 772</td>
</tr>
<tr>
<td>SEARCH</td>
<td>REAL</td>
<td>5</td>
<td>646 658 670</td>
</tr>
<tr>
<td>SIN</td>
<td>REAL</td>
<td>1 LIBRARY</td>
<td>372 2*507</td>
</tr>
<tr>
<td>SORT</td>
<td>REAL</td>
<td>1 LIBRARY</td>
<td>293 392 480 702 714 777 793 804 867</td>
</tr>
<tr>
<td>TAN</td>
<td>REAL</td>
<td>1 LIBRARY</td>
<td>522 610</td>
</tr>
<tr>
<td>TIME</td>
<td>REAL</td>
<td>1</td>
<td>47</td>
</tr>
<tr>
<td>WRITE</td>
<td>REAL</td>
<td>6</td>
<td>185 202</td>
</tr>
</tbody>
</table>

INLINE FUNCTIONS TYPE ARGS DEF LINE REFERENCES
| ABS        | REAL    | 1 INTRIN | 260 246 275 608 639 |
| AMIN       | REAL    | 0 INTRIN | 291 292 297 |
| IABS       | INTEGER | 1 INTRIN | 264       |
| ISIGN      | INTEGER | 2 INTRIN | 263       |
| MOD        | INTEGER | 2 INTRIN | 834       |
| SIGN       | REAL    | 2 INTRIN | 2*288 2*289 |

STATEMENT LABELS DEF LINE REFERENCES
<p>| 5725 | 1 | 840 837 833 834 838 |
| 4226 | 2 | 226 215 224 |
| 4231 | 3 | 229 218 |
| 4222 | 4 | 225 220 221 |
| 5103 | 5 | 546 533 |
| 4155 | 6 | 201 198 |
| 4087 | 7 | 163 171 174 |
| 0     | 8 | 199 196 |</p>
<table>
<thead>
<tr>
<th>PROGRAM ARP</th>
<th>73/74</th>
<th>CPT=1</th>
<th>FORN. 4.8=9PB</th>
<th>05/13/61</th>
<th>08.29.39</th>
<th>PAGE 26</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATEMENT LABELS</td>
<td>DEF LINE</td>
<td>REFERENCES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5531</td>
<td>64</td>
<td>513</td>
<td>435</td>
<td>516</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5127</td>
<td>65</td>
<td>531</td>
<td>2-420</td>
<td>527</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4682</td>
<td>66</td>
<td>424</td>
<td>514</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5237</td>
<td>67</td>
<td>533</td>
<td>561</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4233</td>
<td>68</td>
<td>147</td>
<td>158</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4173</td>
<td>69</td>
<td>230</td>
<td>255</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5423</td>
<td>70</td>
<td>732</td>
<td>692</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5354</td>
<td>71</td>
<td>676</td>
<td>623</td>
<td>224</td>
<td>226</td>
<td>627</td>
</tr>
<tr>
<td>5324</td>
<td>72</td>
<td>235</td>
<td>271</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5207</td>
<td>73</td>
<td>613</td>
<td>603</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5218</td>
<td>74</td>
<td>616</td>
<td>612</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>75</td>
<td>640</td>
<td>643</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>76</td>
<td>654</td>
<td>660</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5517</td>
<td>77</td>
<td>854</td>
<td>622</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>78</td>
<td>811</td>
<td>603</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5351</td>
<td>79</td>
<td>676</td>
<td>653</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5304</td>
<td>80</td>
<td>651</td>
<td>639</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5144</td>
<td>81</td>
<td>732</td>
<td>693</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>82</td>
<td>700</td>
<td>694</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5185</td>
<td>83</td>
<td>593</td>
<td>581</td>
<td>566</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5365</td>
<td>84</td>
<td>665</td>
<td>593</td>
<td>677</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>85</td>
<td>662</td>
<td>655</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5533</td>
<td>86</td>
<td>653</td>
<td>651</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5242</td>
<td>87</td>
<td>627</td>
<td>625</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5241</td>
<td>88</td>
<td>812</td>
<td>810</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>89</td>
<td>302</td>
<td>300</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4415</td>
<td>90</td>
<td>298</td>
<td>299</td>
<td>294</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4437</td>
<td>91</td>
<td>295</td>
<td>299</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>92</td>
<td>674</td>
<td>667</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4405</td>
<td>93</td>
<td>235</td>
<td>259</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4405</td>
<td>94</td>
<td>235</td>
<td>259</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>95</td>
<td>292</td>
<td>250</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7124</td>
<td>96</td>
<td>165</td>
<td>127</td>
<td>182</td>
<td>164</td>
<td>197</td>
</tr>
<tr>
<td>7130</td>
<td>97</td>
<td>219</td>
<td>54</td>
<td>149</td>
<td>189</td>
<td>207</td>
</tr>
<tr>
<td>7267</td>
<td>98</td>
<td>260</td>
<td>159</td>
<td></td>
<td>261</td>
<td></td>
</tr>
<tr>
<td>10152</td>
<td>99</td>
<td>263</td>
<td>265</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10142</td>
<td>100</td>
<td>955</td>
<td>961</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7328</td>
<td>101</td>
<td>237</td>
<td>173</td>
<td></td>
<td>226</td>
<td></td>
</tr>
<tr>
<td>6577</td>
<td>102</td>
<td>51</td>
<td>650</td>
<td></td>
<td>929</td>
<td></td>
</tr>
<tr>
<td>6577</td>
<td>103</td>
<td>6</td>
<td>30</td>
<td>45</td>
<td>52</td>
<td>56</td>
</tr>
<tr>
<td>10050</td>
<td>104</td>
<td>950</td>
<td>946</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7653</td>
<td>105</td>
<td>372</td>
<td>671</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6567</td>
<td>106</td>
<td>49</td>
<td>48</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10030</td>
<td>107</td>
<td>941</td>
<td>940</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10146</td>
<td>108</td>
<td>966</td>
<td>953</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10153</td>
<td>109</td>
<td>958</td>
<td>953</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10140</td>
<td>110</td>
<td>749</td>
<td>947</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7333</td>
<td>111</td>
<td>84</td>
<td>878</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7781</td>
<td>112</td>
<td>925</td>
<td>923</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7114</td>
<td>113</td>
<td>912</td>
<td>831</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7726</td>
<td>114</td>
<td>912</td>
<td>910</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7370</td>
<td>115</td>
<td>448</td>
<td>419</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7524</td>
<td>116</td>
<td>713</td>
<td>721</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOOPS LABEL INDEX FROM TO LENGTH PROPERTIES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4067</td>
<td>51</td>
<td>1</td>
<td>157 158</td>
<td>26</td>
<td>INSTACK</td>
<td></td>
</tr>
<tr>
<td>4070</td>
<td>53</td>
<td>1</td>
<td>147 172</td>
<td>115</td>
<td>CPT</td>
<td></td>
</tr>
<tr>
<td>EXITS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
SUBROUTINE ROTATE 73/74 CPE=1

SUBROUTINE ROTATE (R,D,PHI,SIGNX)

ROTATES COORDINATE SYSTEM FROM TARGET SYSTEM
TO PROJECTILE SYSTEM OR VICE VERSA, DEPENDING
ON THE VALUE OF SIGNX (±1 = TO PROJECTILE SYSTEM,
AND −1 = TO TARGET SYSTEM).

RT = R
R = R*COS(PHI) - SIGNX*D*SIN(PHI)
D = D*COS(PHI) + SIGNX*RT*SIN(PHI)
END

SYMBOLIC REFERENCE MAP (R=2)

ENTRY POINTS DEF LINE REFERENCES
2  ROTATE 1 1

VARIABLES SN TYPE RELLOCATION REFS REFERENCES
0  D  REAL F.P.  9  10 DEFINED 1 10
0  PHI  REAL  F.P.  2=9  2=10 DEFINED 1  9
0  R  REAL  F.P.  8  8 DEFINED 9  9
0  SIGNX  REAL  F.P.  10 DEFINED 1  1

EXTERNALS TYPE ARGS REFERENCES
COS  REAL  1 LIBRARY 9  10
SIN  REAL  1 LIBRARY 9  10

STATISTICS
PROGRAM LENGTH JIB 25
820093 CM USED
<table>
<thead>
<tr>
<th>SUBROUTINE READ</th>
<th>73/74</th>
<th>GPT=1</th>
<th>FTN 4.6+508</th>
<th>03/13/81 08:25:30</th>
<th>PAGE 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>C</td>
<td>C</td>
<td>READ IN SUPPLEMENTAL INPUTS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>C</td>
<td>C</td>
<td>DIMENSION X(50),H(5),O(3),V(5),G(3,10),PV(5),ANAM(50),INEW(50)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DIMENSION SRED(5,2),SH(5,2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>COMVRK,ROVRK,HDG,V,G,PV,GM2,GMD,MGH,SM2(SDF(10),SDF(10),IDAT(10))</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1, RS(5),BLST(5),RBLST(5),RJ,DU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td>DO 11: ID=1,10</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IDR = IDAT(ID)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IF(IDR.EQ.0.OR.(INVR(IDR).EQ.0.AND.IDAT.EQ.0)) GO TO 11</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IF(K(28).EQ.0.AND.ID.EQ.8) GO TO 11</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NN = 4SS(X(IDR))</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IF(NN.EQ.0) GO TO 11</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IF(IDR.EQ.5) WRITE (6,1009) ANAM(IDR)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GO TO (1,2,3,4,5,6,8,10)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 IF(X(9)) 21,20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td>IF(IDR.EQ.5) WRITE (6,1007) NN</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>READ (IRD,*), (SSD(I),SDF(I),I=1,NN)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GO TO 9</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>21 IF(IDR.EQ.5) WRITE (6,1008) NN</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>READ (IRD,*), (SSD(I),I=1,NN)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DD 12 I=1,NN</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SSD(I) = SSD(I)/1.1774</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>12 SSD(I) = SSD(I)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>9 IF(IDR.EQ.5) WRITE (6,1012)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>READ (IRD,*), GM3,GMD,GMH</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GO TO 11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
<td></td>
<td>2 IF(IDR.EQ.5) WRITE (6,1005) NN</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>READ (IRD,*), (SR(I,1),SR(I,2),SO(I,1),SO(I,2),SH(I,1),SH(I,2),I=1,10250)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C NN</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GO TO 11</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3 NN = NN + 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IF(IDR.EQ.5) WRITE (6,1000) NN</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>READ (IRD,*), (HH(I),I=1,NN)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IF(IDR.EQ.5) WRITE (6,1014)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>READ (IRD,*), R1,DU</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GO TO 11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td></td>
<td></td>
<td>4 IF(IDR.EQ.5) WRITE (6,1001) NN</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>READ (IRD,*), (G(I),I=1,NN)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GO TO 11</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5 IF(IDR.EQ.5) WRITE (6,1002) NN</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>READ (IRD,*), (VI(I),I=1,NN)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IF(IDR.EQ.5) WRITE (6,1004) NN</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>READ (IRD,*), (VP(I),I=1,NN)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GO TO 11</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5 IF(IDR.EQ.5) WRITE (6,1003) NN</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>READ (IRD,*), (GI(I,J),I=1,3,J=1,NN)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GO TO 11</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>7 IF(IDR.EQ.5) WRITE (6,1005)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>READ (IRD,*), (RDR(I),I=1,3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IF(IDR.EQ.5) WRITE (6,1013)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>READ (IRD,*), RDR(5)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GO TO 11</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6 IF(IDR.EQ.5) WRITE (6,1010) NN</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IF(IDR.EQ.5) WRITE (6,1011)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### SUBROUTINE READ 73/74 GPT1

**READ (10,*) (EBLIST(I),HELSST(I),I=1,NN)**

1 CONTINUE

100 FORMAT (1X,*ENTER *,I2,* HEIGHTS FOR FRAGMENTATION PK GRID,* ,/)

CIX,*LAST VALUE CORRESPONS TO HEIGHT*,/.

CIX,*WHERE PK GOES TO ZERO*,*)

1001 FORMAT (1X,*ENTER *,I2,* ELEVATION ANGLES ASSOCIATED WITH *,*

C/.*,I2*,FRAGMENTATION P DATA *=*

1002 FORMAT (1X,*ENTER *,I2,* VT FUZZING HEIGHTS *)

1003 FORMAT (1X,*ENTER *,I2,* SETS OF GLITTER POINT COORDINATES (R,D,H)010600

C *=

1004 FORMAT (1X,*ENTER *,I2,* PROB. VT DETONATION AT HEIGHT M = *)

1005 FORMAT (1X,*ENTER *,I2,* SETS OF BOUNDARIES FOR DIRECT HIT BOXES*=*

C/.* MIN HGT, MAX HGT = *

1006 FORMAT (1X,*ENTER RADAR ANTENNA COORDINATES (R,D,H) RELATIVE*

C/.* TARGET GROUND ZERO, *=

1007 FORMAT (1X,*ENTER *,I2,* SETS OF GUIDANCE ERRORS *=*

1008 FORMAT (1X,*ENTER *,I2,* SETS OF GUIDANCE ERRORS *=

1/3A,CEP *=. /**

1009 FORMAT (1X,6A1X,*DATA *=)

1010 FORMAT (1X,*ENTER *,I2,* SETS OF BLST, HGT DATA *)

1011 FORMAT (1X,*BEGINNING WITH LOWEST HEIGHT =~*)

1012 FORMAT (1X,*ENTER COORDINATES OF MOING POINT (R,D,H) *=*,/)

1013 FORMAT (1X,*ENTER R1,R2, WHERE RADAR BLAST PK *=,/*

CIX,*OUT TO RT AND DECLINES LINEARLY *=./**

CIX,*TO ZERO AT R2 *=

1014 FORMAT (1X,*ENTER RANGE AND DEFLCTION DISTANCES*=./**

CIX,*FROM EDGE OF GRID TO WHERE THE FR DETONATION *=./**

CIX,*PK GOES TO ZERO = *

END

---

### SYMBOLIC REFERENCE MAP (R=2)

**ENTRY POINTS**

- **3 READ**

**DEF LINE**

- **86**

**REFERENCES**

**VARIABLES**

- **ANAM**
- **EBLIST**
- **GMH**
- **GMR**
- **ID**
- **IDAT**

**SN TYPE**

- **REAL**
- **ARRAY**
- **REFERENCE**

**RELOCATION**

- **16 DEFINED**
- **58 DEFINED**
- **38 DEFINED**
- **49 DEFINED**
- **36 DEFINED**
- **10 DEFINED**

**REFS**

- **6**
- **8**
- **6**
- **6**
- **6**
- **8**

**VALUES**

- **2+20**
- **44**
- **24**
- **23**
- **45**
- **31**

**REFERENCES**

- **11 DEFINED**
- **13 DEFINED**
- **17 DEFINED**
- **20 DEFINED**
- **23 DEFINED**
- **46 DEFINED**
- **49 DEFINED**
- **52 DEFINED**
### Variable Declaration Table

<table>
<thead>
<tr>
<th>VARIABLES</th>
<th>SN</th>
<th>TYPE</th>
<th>RELLOCATION</th>
<th>REFS</th>
<th>2*12</th>
<th>14</th>
<th>16</th>
<th>DEFINED</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>707 IDR</td>
<td>0</td>
<td>INTEGER</td>
<td>ARRAY F.P.</td>
<td>REFS</td>
<td>6</td>
<td>12</td>
<td>DEFINED</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>707 INEW</td>
<td>0</td>
<td>INTEGER</td>
<td>ARRAY F.P.</td>
<td>REFS</td>
<td>16</td>
<td>19</td>
<td>22</td>
<td>27</td>
<td>30</td>
</tr>
<tr>
<td>707 I0PT</td>
<td>0</td>
<td>INTEGER</td>
<td>ARRAY F.P.</td>
<td>REFS</td>
<td>40</td>
<td>43</td>
<td>45</td>
<td>48</td>
<td>51</td>
</tr>
<tr>
<td>707 I0D</td>
<td>0</td>
<td>INTEGER</td>
<td>ARRAY F.P.</td>
<td>REFS</td>
<td>38</td>
<td>41</td>
<td>44</td>
<td>46</td>
<td>49</td>
</tr>
<tr>
<td>712 J</td>
<td>0</td>
<td>INTEGER</td>
<td>ARRAY F.P.</td>
<td>REFS</td>
<td>49</td>
<td>DEFINED</td>
<td>49</td>
<td></td>
<td></td>
</tr>
<tr>
<td>710 NN</td>
<td>0</td>
<td>INTEGER</td>
<td>ARRAY F.P.</td>
<td>REFS</td>
<td>15</td>
<td>19</td>
<td>20</td>
<td>22</td>
<td>23</td>
</tr>
<tr>
<td>11 D</td>
<td>0</td>
<td>REAL</td>
<td>ARRAY R0WRT</td>
<td>REFS</td>
<td>6</td>
<td>8</td>
<td>DEFINED</td>
<td>41</td>
<td></td>
</tr>
<tr>
<td>57 PV</td>
<td>0</td>
<td>REAL</td>
<td>ARRAY R0WRT</td>
<td>REFS</td>
<td>6</td>
<td>8</td>
<td>DEFINED</td>
<td>46</td>
<td></td>
</tr>
<tr>
<td>125 RDR</td>
<td>0</td>
<td>REAL</td>
<td>ARRAY R0WRT</td>
<td>REFS</td>
<td>8</td>
<td>DEFINED</td>
<td>52</td>
<td></td>
<td></td>
</tr>
<tr>
<td>144 RJ</td>
<td>0</td>
<td>REAL</td>
<td>ARRAY R0WRT</td>
<td>REFS</td>
<td>7</td>
<td>DEFINED</td>
<td>38</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 SD</td>
<td>0</td>
<td>REAL</td>
<td>ARRAY F.P.</td>
<td>REFS</td>
<td>8</td>
<td>25</td>
<td>26</td>
<td>DEFINED</td>
<td>20</td>
</tr>
<tr>
<td>67 SDD</td>
<td>0</td>
<td>REAL</td>
<td>ARRAY R0WRT</td>
<td>REFS</td>
<td>8</td>
<td>DEFINED</td>
<td>26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>101 SDH</td>
<td>0</td>
<td>REAL</td>
<td>ARRAY R0WRT</td>
<td>REFS</td>
<td>7</td>
<td>DEFINED</td>
<td>23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 SH</td>
<td>0</td>
<td>REAL</td>
<td>ARRAY F.P.</td>
<td>REFS</td>
<td>7</td>
<td>DEFINED</td>
<td>25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 SR</td>
<td>0</td>
<td>REAL</td>
<td>ARRAY F.P.</td>
<td>REFS</td>
<td>7</td>
<td>DEFINED</td>
<td>37</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16 V</td>
<td>0</td>
<td>REAL</td>
<td>ARRAY R0WRT</td>
<td>REFS</td>
<td>6</td>
<td>8</td>
<td>DEFINED</td>
<td>44</td>
<td></td>
</tr>
<tr>
<td>0 X</td>
<td>0</td>
<td>REAL</td>
<td>ARRAY F.P.</td>
<td>REFS</td>
<td>13</td>
<td>14</td>
<td>16</td>
<td>DEFINED</td>
<td>1</td>
</tr>
</tbody>
</table>

**FILE NAMES**

- TAPE6 FMT
- MODE WRITE

**Variables Used as File Names, See Above**

**Statement Labels**

<table>
<thead>
<tr>
<th>STATEMENT LABELS</th>
<th>DEF LINE</th>
<th>REFERENCES</th>
</tr>
</thead>
<tbody>
<tr>
<td>46 1</td>
<td>18</td>
<td>17</td>
</tr>
<tr>
<td>117 2</td>
<td>30</td>
<td>17</td>
</tr>
<tr>
<td>145 3</td>
<td>34</td>
<td>17</td>
</tr>
<tr>
<td>165 4</td>
<td>40</td>
<td>17</td>
</tr>
<tr>
<td>200 5</td>
<td>43</td>
<td>17</td>
</tr>
<tr>
<td>223 6</td>
<td>48</td>
<td>17</td>
</tr>
<tr>
<td>235 7</td>
<td>51</td>
<td>17</td>
</tr>
<tr>
<td>253 6</td>
<td>56</td>
<td>17</td>
</tr>
<tr>
<td>110 9</td>
<td>27</td>
<td>21</td>
</tr>
<tr>
<td>277 11</td>
<td>59</td>
<td>10</td>
</tr>
<tr>
<td>0 12</td>
<td>26</td>
<td>24</td>
</tr>
<tr>
<td>0 20</td>
<td>19</td>
<td>18</td>
</tr>
<tr>
<td>70 21</td>
<td>22</td>
<td>18</td>
</tr>
</tbody>
</table>

**Ass. REAL Intrin 1 Instrn 14**

**Statement Functions**

<table>
<thead>
<tr>
<th>STATEMENT LABELS</th>
<th>TYPE</th>
<th>ARGS</th>
<th>DEF LINE</th>
<th>REFERENCES</th>
</tr>
</thead>
<tbody>
<tr>
<td>476 1000</td>
<td>FMT</td>
<td>60</td>
<td>35</td>
<td>33</td>
</tr>
<tr>
<td>514 1001</td>
<td>FMT</td>
<td>63</td>
<td>40</td>
<td>39</td>
</tr>
<tr>
<td>526 1002</td>
<td>FMT</td>
<td>65</td>
<td>43</td>
<td>42</td>
</tr>
<tr>
<td>534 1003</td>
<td>FMT</td>
<td>66</td>
<td>46</td>
<td>47</td>
</tr>
<tr>
<td>544 1004</td>
<td>FMT</td>
<td>66</td>
<td>45</td>
<td>47</td>
</tr>
<tr>
<td>553 1005</td>
<td>FMT</td>
<td>69</td>
<td>30</td>
<td>47</td>
</tr>
<tr>
<td>574 1006</td>
<td>FMT</td>
<td>72</td>
<td>51</td>
<td>47</td>
</tr>
<tr>
<td>606 1007</td>
<td>FMT</td>
<td>74</td>
<td>19</td>
<td>47</td>
</tr>
<tr>
<td>617 1008</td>
<td>FMT</td>
<td>76</td>
<td>22</td>
<td>47</td>
</tr>
<tr>
<td>627 1009</td>
<td>FMT</td>
<td>78</td>
<td>16</td>
<td>47</td>
</tr>
<tr>
<td>SUBROUTINE READ</td>
<td>73/74</td>
<td>CPT=1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>-------</td>
<td>-------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STATEMENT LABELS</td>
<td>DEF LINE</td>
<td>REFERENCES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>632 1010 FMT</td>
<td>79</td>
<td>56</td>
<td></td>
<td></td>
</tr>
<tr>
<td>640 1011 FMT</td>
<td>80</td>
<td>57</td>
<td></td>
<td></td>
</tr>
<tr>
<td>645 1012 FMT</td>
<td>81</td>
<td>27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>654 1013 FMT</td>
<td>82</td>
<td>53</td>
<td></td>
<td></td>
</tr>
<tr>
<td>670 1014 FMT</td>
<td>85</td>
<td>37</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LOOPS</th>
<th>LABEL</th>
<th>INDEX</th>
<th>FROM-TO</th>
<th>LENGTH</th>
<th>PROPERTIES</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>11</td>
<td>I</td>
<td>10 59</td>
<td>2738</td>
<td>EXT REFS NOT INNER</td>
</tr>
<tr>
<td>56</td>
<td>I</td>
<td>20 20</td>
<td>102</td>
<td>EXT REFS</td>
<td></td>
</tr>
<tr>
<td>104</td>
<td>12</td>
<td>I</td>
<td>24 26</td>
<td>38</td>
<td>INSTACK</td>
</tr>
<tr>
<td>126</td>
<td>I</td>
<td>31 31</td>
<td>156</td>
<td>EXT REFS</td>
<td></td>
</tr>
<tr>
<td>268</td>
<td>I</td>
<td>58 58</td>
<td>106</td>
<td>EXT REFS</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COMMON BLOCKS LENGTH</th>
<th>RDWRT</th>
<th>102</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>STATISTICS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROGRAM LENGTH</td>
</tr>
<tr>
<td>7478</td>
</tr>
<tr>
<td>CM LABELED COMMON LENGTH</td>
</tr>
<tr>
<td>146B</td>
</tr>
</tbody>
</table>

520092 CM USED
SUBROUTINE WRITE 73/74 OPT=1 39 4.8+508 03/13/61 08.29.30 PAGE 1

1        SUBROUTINE WRITE (X, IWRT, CFP, SR, SD, SH)
2             WRITE LIST OF DATA (OUTPUT & TAPE1)
3
4      DIMENSION X(50), H(9), Q(3), V(5), G(3, 10), PV(5), CEP(10)
5      DIMENSION SR(5, 2), SD(5, 2), SH(10), SDH(10), IDAT(10)
6
7      1. PDR(5), BBLST(5), HBLST(5), RU, DU
8      DO 8 I=1, 50
9      NN = ABS(X(I))
10     IF(NN.EQ.0) GO TO 8
11     DU = J
12     IF(IDAT(JJ).EQ.0) GO TO 11
13     CONTINUE
14     GO TO 8
15
16     11 GO TO (1, 2, 7, 3, 4, 5, 6, 9), JJ
17     IF(A(9)) 21, 20
18
19     20 WRITE (IWRT,*) (SDD(K), SDH(K), K=1, NN)
20     GO TO 13
21     21 END 12 K=1, NN
22     12 CEP(K) = SDD(K)*.1774
23     WRITE (IWRT,*) (CEP(K), K=1, NN)
24     WRITE (IWRT,*) (SR(K, 1), SR(K, 2), SO(K, 1), SO(K, 2), SH(K, 1), SH(K, 2), K=1, NN)
25     GO TO 8
26
27     30 WRITE (IWRT,*) (H(K), K=1, NN)
28     WRITE (IWRT,*) RU, DU
29     GO TO 8
30
31     4 WRITE (IWRT,*) (C(K), K=1, NN)
32     GO TO 8
33
34     5 WRITE (IWRT,*) (V(K), K=1, NN)
35     WRITE (IWRT,*) (PV(K), K=1, NN)
36     GO TO 8
37
38     6 WRITE (IWRT,*) ((G(L, K), L=1, 3), K=1, NN)
39     GO TO 8
40
41     7 WRITE (IWRT,*) (PDR(J), J=1, 2)
42     WRITE (IWRT,*) KDR(4), KDR(5)
43     GO TO 8
44
45     9 IF(X(I).GE.0) GO TO 6
46     WRITE (IWRT,*) (BBLST(J), HBLST(J), J=1, NN)
47
48     CONTINUE
49
50     END

SYMBOLIC REFERENCE MAP (R=2)

ENTRY POINTS  DEF LINE  REFERENCES
3 WRITE

1 47
<table>
<thead>
<tr>
<th>LOOPS</th>
<th>LABEL</th>
<th>INDEX</th>
<th>FROM-TO</th>
<th>LENGTH</th>
<th>PROPERTIES</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>8</td>
<td>I</td>
<td>10-46</td>
<td>176B</td>
<td>EXT REFS NOT INNER</td>
</tr>
<tr>
<td>14</td>
<td>16</td>
<td>J</td>
<td>13-16</td>
<td>58</td>
<td>INSTACK EXITS</td>
</tr>
<tr>
<td>43</td>
<td></td>
<td>K</td>
<td>20-20</td>
<td>108</td>
<td>EXT REFS</td>
</tr>
<tr>
<td>60</td>
<td>12</td>
<td>K</td>
<td>22-23</td>
<td>38</td>
<td>INSTACK EXITS</td>
</tr>
<tr>
<td>77</td>
<td></td>
<td>K</td>
<td>27-27</td>
<td>158</td>
<td>EXT REFS</td>
</tr>
<tr>
<td>171</td>
<td></td>
<td>J</td>
<td>45-45</td>
<td>108</td>
<td>EXT REFS</td>
</tr>
</tbody>
</table>

**COMMON BLOCKS**

| LENGTH | RDWRT | 102  |

**STATISTICS**

<table>
<thead>
<tr>
<th>PROGRAM LENGTH</th>
<th>320B</th>
<th>208</th>
</tr>
</thead>
<tbody>
<tr>
<td>CM LABELED COMMON LENGTH</td>
<td>146B</td>
<td>102</td>
</tr>
</tbody>
</table>

52000B CM USED
SUBROUTINE GRIDS

READ IN FRAGMENTATION PK GRID REDEFINE AND ORIENT AXES TO CORRESPOND WITH GEOMETRY OF MODEL

GRIDS ARE IN ROTATED PROJECTILE COORDINATE SYSTEM.

DIMENSION PK(40,20,6),R(6,41),D(6,21)

DO 1 I=1,NH
READ (KK,1601) NR,NK
IF(NK NE.5) WRITE (6,2000) R(I,NK),NR,NK
READ (KK,1000) (R(I,NK-J+1),J=1,NR)
READ (KK,1000) (D(I,NK-J+1),J=1,NK)
NR = NR - 1
NK = NK - 1

DO 4 J=1,NR

DO 5 J=1,NK
5 D(I,J) = -(D(I,J) + D(I,J+1))/2.

IF(NK NE.5) WRITE (6,2000) (R(I,J),J=1,NR)
IF(NK NE.5) WRITE (6,2000) (D(I,J),J=1,NK)

DO 1 J=1,NK
1 READ (KK,1002) (PK(NR-J+1,NK-J+1),J=1,NK)
IF(NK NE.5) RETURN
END

SYMBOLIC REFERENCE MAP (R=2)

ENTRY POINTS  DEF LINE  REFERENCES
3 GRIDS  1  32  43

VARIABLES  SN  TYPE  ARRAY  F.P.  REFS  9  2=27  29  DEFINED  1  14  27
0  0  REAL  327  1  35  DEFINED  10  35
327  1  INTEGER  327  5  14  3=27  29  29  31

03/13/81  08:29:30  PAGE  1
### Subroutine Grids 73/74 OPE-1

<table>
<thead>
<tr>
<th>VARIABLES</th>
<th>TYPE</th>
<th>SN</th>
<th>RELLOCATION</th>
<th>REFS</th>
<th>DEFINED</th>
<th>REFERENCES</th>
<th>REV</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>310 J</td>
<td>INTEGER</td>
<td>310</td>
<td>REFS</td>
<td>13</td>
<td>35</td>
<td>3</td>
<td>28</td>
<td>31</td>
</tr>
<tr>
<td>311 K</td>
<td>INTEGER</td>
<td>311</td>
<td>REFS</td>
<td>31</td>
<td>35</td>
<td>3</td>
<td>28</td>
<td>31</td>
</tr>
<tr>
<td>0 KK</td>
<td>INTEGER</td>
<td>0</td>
<td>F.P.</td>
<td>1</td>
<td>I/O REFS</td>
<td>11</td>
<td>14</td>
<td>31</td>
</tr>
<tr>
<td>0 ND</td>
<td>INTEGER</td>
<td>0</td>
<td>F.P.</td>
<td>12</td>
<td>2*14</td>
<td>16</td>
<td>26</td>
<td>29</td>
</tr>
<tr>
<td>0 NN</td>
<td>INTEGER</td>
<td>0</td>
<td>F.P.</td>
<td>10</td>
<td>33</td>
<td>29</td>
<td>32</td>
<td>31</td>
</tr>
<tr>
<td>0 NR</td>
<td>INTEGER</td>
<td>0</td>
<td>F.P.</td>
<td>12</td>
<td>2*13</td>
<td>15</td>
<td>3</td>
<td>28</td>
</tr>
<tr>
<td>0 PK</td>
<td>REAL</td>
<td>0</td>
<td>ARRAY</td>
<td>9</td>
<td>35</td>
<td>1</td>
<td>31</td>
<td>31</td>
</tr>
<tr>
<td>0 R</td>
<td>REAL</td>
<td>0</td>
<td>ARRAY</td>
<td>9</td>
<td>2*25</td>
<td>28</td>
<td>34</td>
<td>31</td>
</tr>
</tbody>
</table>

### File Names MODE

<table>
<thead>
<tr>
<th>TAPE6</th>
<th>FMT</th>
<th>WRITES</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>12</td>
</tr>
</tbody>
</table>

### Statement Labels

<table>
<thead>
<tr>
<th>STATEMENT</th>
<th>LABEL</th>
<th>DEF</th>
<th>LINE</th>
<th>REFERENCES</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1</td>
<td>31</td>
<td>10</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>0 2</td>
<td>35</td>
<td>33</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>0 4</td>
<td>25</td>
<td>24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 5</td>
<td>27</td>
<td>26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>271 1000 FMT</td>
<td>37</td>
<td>13</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>273 1001 FMT</td>
<td>38</td>
<td></td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>275 1002 FMT</td>
<td>39</td>
<td></td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>277 2000 FMT</td>
<td>40</td>
<td>28</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td>301 2001 FMT</td>
<td>41</td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>303 2002 FMT</td>
<td>42</td>
<td>35</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Loops LABEL INDEX FROM-TO LENGTH PROPERTIES

<table>
<thead>
<tr>
<th>LOOPS</th>
<th>LABEL</th>
<th>INDEX</th>
<th>FROM-TO</th>
<th>LENGTH</th>
<th>PROPERTIES</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 1</td>
<td>J</td>
<td>10 21</td>
<td>1508</td>
<td></td>
<td>EXT REFS NOT INNER</td>
</tr>
<tr>
<td>20</td>
<td>J</td>
<td>13 13</td>
<td></td>
<td></td>
<td>EXT REFS</td>
</tr>
<tr>
<td>35</td>
<td>J</td>
<td>14 14</td>
<td>118</td>
<td></td>
<td>EXT REFS</td>
</tr>
<tr>
<td>56</td>
<td>J</td>
<td>24 25</td>
<td>46</td>
<td>INSTACK</td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>J</td>
<td>26 27</td>
<td>46</td>
<td>INSTACK</td>
<td></td>
</tr>
<tr>
<td>101</td>
<td>J</td>
<td>28 28</td>
<td>105</td>
<td></td>
<td>EXT REFS</td>
</tr>
<tr>
<td>117</td>
<td>J</td>
<td>29 29</td>
<td>108</td>
<td></td>
<td>EXT REFS</td>
</tr>
<tr>
<td>131 1</td>
<td>J</td>
<td>30 31</td>
<td>248</td>
<td></td>
<td>EXT REFS NOT INNER</td>
</tr>
<tr>
<td>134</td>
<td>K</td>
<td>31 31</td>
<td>158</td>
<td></td>
<td>EXT REFS</td>
</tr>
<tr>
<td>162 2</td>
<td>I</td>
<td>33 36</td>
<td>248</td>
<td></td>
<td>EXT REFS NOT INNER</td>
</tr>
<tr>
<td>163 2</td>
<td>J</td>
<td>34 36</td>
<td>216</td>
<td></td>
<td>EXT REFS NOT INNER</td>
</tr>
<tr>
<td>168</td>
<td>K</td>
<td>35 35</td>
<td>128</td>
<td></td>
<td>EXT REFS</td>
</tr>
</tbody>
</table>

### Statistics

<table>
<thead>
<tr>
<th>PROGRAM LENGTH</th>
<th>CM USED</th>
</tr>
</thead>
<tbody>
<tr>
<td>350B</td>
<td>232</td>
</tr>
</tbody>
</table>
SUBROUTINE SEARCH 73/74  OPT=1

SUBROUTINE SEARCH (II,JJ,R,D,H)
DETERMINES WHETHER INTERCEPT OF TRAJECTORY WITH DIRECT
HIT BOX PLANES FALLS WITHIN BOUNDARY OF DIRECT HIT BOX.
UPDATES PENETRATION (BOX INTERCEPT) COORDINATES (RPN,DPN,HPN)
IF HEIGHT COMPONENT INDICATES BOX PENETRATION OCCURS
PRIOR TO CURRENT HPN POINT ALONG TRAJECTORY. INITIAL VALUE
OF HPN IS BASED ON BURST POINT HEIGHT, BH.

COMMON /SRCH/, IPEN, DRN, RBS, DBS, HBS, BR, BD, BH, OMEGA, RPN, DPN
C, HPN, RH2, RH2, COM1, COM2, HDH1, HDH2, HDH1
DX(R1,R2, D1, D2, H1, H2) = (R2-R1)*(R2-R1)+(D2-D1)+(H2-H1)*H0
C2-H1)

CALL INTRAP (R,D,H,JJ)

ENTRY POINTS
3 SEARCH 1 15 16 17 19 20 25

SYMBOLIC REFERENCE MAP (R=2)

VARIABLES SN TYPE RELLOCATION
6 BD REAL SRCH REFS 10
7 BH REAL SRCH REFS 10
5 BR REAL SRCH REFS 10
0 D REAL F.P. REFS 10
3 DBS REAL SRCH REFS 10
16 DDH1 REAL SRCH REFS 10
17 DDH2 REAL SRCH REFS 10
12 DPN REAL SRCH REFS 10
0 IPN REAL F.P. REFS 10
4 HBS REAL SRCH REFS 10
20 HDH1 REAL SRCH REFS 10
21 HDH2 REAL SRCH REFS 10
13 HPN REAL SRCH REFS 10
0 II INTEGER F.P. REFS 10
0 IPEN INTEGER SRCH REFS 10
1 IPN INTEGER SRCH REFS 10
0 JJ INTEGER F.P. REFS 10
10 OMEGA REAL SRCH REFS 10
0 R REAL F.P. REFS 10
2 RBS REAL SRCH REFS 10
### Subroutine Search

<table>
<thead>
<tr>
<th>Variables</th>
<th>SN</th>
<th>Type</th>
<th>Relocation</th>
<th>Refs</th>
<th>Defined</th>
</tr>
</thead>
<tbody>
<tr>
<td>14 RHR1</td>
<td></td>
<td>REAL</td>
<td>SRCH</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>15 RHR2</td>
<td></td>
<td>REAL</td>
<td>SRCH</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>11 RPN</td>
<td></td>
<td>REAL</td>
<td>SRCH</td>
<td>10</td>
<td>DEFINED 21</td>
</tr>
</tbody>
</table>

### Externals

<table>
<thead>
<tr>
<th>Type</th>
<th>Args</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTCP</td>
<td>4</td>
<td>14</td>
</tr>
</tbody>
</table>

### Inline Functions

<table>
<thead>
<tr>
<th>Type</th>
<th>Args</th>
<th>Def Line</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>DX</td>
<td>REAL</td>
<td>SF</td>
<td>12</td>
</tr>
</tbody>
</table>

### Common Blocks

<table>
<thead>
<tr>
<th>Length</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SRCH</td>
<td>18</td>
</tr>
</tbody>
</table>

### Statistics

<table>
<thead>
<tr>
<th>Program Length</th>
<th>468</th>
<th>38</th>
</tr>
</thead>
<tbody>
<tr>
<td>CM Labeled Common Length</td>
<td>228</td>
<td>18</td>
</tr>
<tr>
<td>52000B CM Used</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
SUBROUTINE INTRCP (R,D,H,IGO)
COMPUTES INTERCEPT OF TRAJECTORY WITH SIDES OF TARGET BOXES USING TWO POINTS. (BR,BD,BH) AND (RBS,DBS,HBS).
INTERCEPT IS AT (R,D).
SEE MAIN ROUTINE BETWEEN STATEMENTS 105 AND 96.
COMMON /SRCH/,IPEN,IBS,RBS,DBS,HBS,BR,BD,BH,OMEGA,RPN,DPN,
C,HPN,RDH1,DDH1,DDH2,MDH1,MDH2
XOFY(GX,XA,Y,GY,YA) = GX + (XA-GX)*(YA-GY)/(YA-GY)
GO TO (1,2,3),IGO
GIVEN R, SOLVE FOR D,H
D = XOFY(BD,DBS,R,BR,RBS)
H = XOFY(BH,HBS,R,BR,RBS)
RETURN
GIVEN D, SOLVE FOR R,H
R = XOFY(BR,RBS,D,BD,DBS)
H = XOFY(BH,HBS,D,BD,DBS)
RETURN
GIVEN H, SOLVE FOR R,D
R = XOFY(BR,RBS,H,BH,HBS)
D = XOFY(BD,DBS,H,BH,HBS)
RETURN
END

CARD NR. SEVERITY DETAILS DIAGNOSIS OF PROBLEM
11 I
AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.

SYMBOLIC REFERENCE MAP (R=2)

<table>
<thead>
<tr>
<th>ENTRY</th>
<th>POINTS</th>
<th>DEF LINE</th>
<th>REFERENCES</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 INTRCP</td>
<td>1</td>
<td>17</td>
<td>23</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VARIABLES</th>
<th>SN</th>
<th>TYPE</th>
<th>RELLOCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 BD</td>
<td>REAL</td>
<td>SRCH</td>
<td>REFS 8</td>
</tr>
<tr>
<td>7 BH</td>
<td>REAL</td>
<td>SRCH</td>
<td>REFS 8</td>
</tr>
<tr>
<td>5 BR</td>
<td>REAL</td>
<td>SRCH</td>
<td>REFS 8</td>
</tr>
<tr>
<td>0 D</td>
<td>REAL</td>
<td>F.P.</td>
<td>REFS 21</td>
</tr>
<tr>
<td>3 DBS</td>
<td>REAL</td>
<td>SRCH</td>
<td>REFS 8</td>
</tr>
<tr>
<td>16 DDH1</td>
<td>REAL</td>
<td>SRCH</td>
<td>REFS 8</td>
</tr>
<tr>
<td>17 DDH2</td>
<td>REAL</td>
<td>SRCH</td>
<td>REFS 8</td>
</tr>
<tr>
<td>12 DPN</td>
<td>REAL</td>
<td>SRCH</td>
<td>REFS 8</td>
</tr>
<tr>
<td>0 H</td>
<td>REAL</td>
<td>F.P.</td>
<td>REFS 27</td>
</tr>
<tr>
<td>VARIABLES</td>
<td>SN</td>
<td>TYPE</td>
<td>RELOCATION</td>
</tr>
<tr>
<td>-----------</td>
<td>----</td>
<td>------</td>
<td>------------</td>
</tr>
<tr>
<td>4 HBS</td>
<td></td>
<td>REAL</td>
<td>SRCH</td>
</tr>
<tr>
<td>20 MDH1</td>
<td></td>
<td>REAL</td>
<td>SRCH</td>
</tr>
<tr>
<td>21 MDH2</td>
<td></td>
<td>REAL</td>
<td>SRCH</td>
</tr>
<tr>
<td>13 HPN</td>
<td></td>
<td>REAL</td>
<td>SRCH</td>
</tr>
<tr>
<td>1 IBS</td>
<td></td>
<td>INTEGER</td>
<td>F.P.</td>
</tr>
<tr>
<td>0 IGO</td>
<td></td>
<td>INTEGER</td>
<td>F.P.</td>
</tr>
<tr>
<td>0 NIGHA</td>
<td></td>
<td>REAL</td>
<td>SRCH</td>
</tr>
<tr>
<td>10 MEGA</td>
<td></td>
<td>REAL</td>
<td>SRCH</td>
</tr>
<tr>
<td>2 RBS</td>
<td></td>
<td>REAL</td>
<td>SRCH</td>
</tr>
<tr>
<td>14 ROH1</td>
<td></td>
<td>REAL</td>
<td>SRCH</td>
</tr>
<tr>
<td>15 ROH2</td>
<td></td>
<td>REAL</td>
<td>SRCH</td>
</tr>
<tr>
<td>11 RPN</td>
<td></td>
<td>REAL</td>
<td>SRCH</td>
</tr>
</tbody>
</table>

**INLINE FUNCTIONS**

<table>
<thead>
<tr>
<th>XOFY</th>
<th>REAL</th>
<th>DEF</th>
<th>LINE</th>
<th>REFERENCES</th>
<th>10</th>
<th>15</th>
<th>16</th>
<th>21</th>
<th>22</th>
<th>27</th>
<th>28</th>
</tr>
</thead>
</table>

**STATEMENT LABELS**

| 15 | 15 | 11 |
| 27 | 21 | 11 |
| 41 | 27 | 11 |

**COMMON BLOCKS**

| SRCH | LENGTH | 18 |

**STATISTICS**

| PROGRAM LENGTH | 53B | 43 |
| CM LABELED COMMON LENGTH | 22B | 18 |
SUBROUTINE BLAST 73/74  OPT=1  FTN 4.8+508  03/13/81  08.29.30  PAGE 1

1
C
C
C
C

SUBROUTINE BLAS1 (IB,P,B,X,I)

SET IB=0 IF BURST POINT IS OUT OF RANGE
OF NEAR MISS BLAST

5
DIMENSION X(5,2)
IF(P.LT.(X(I,1)-B)) IB = 0
IF(P.GT.(X(I,2)+B)) IB = 0
END

012280
012290
012300
012310
012320
012330
012340
012350
012360

SYMBOLIC REFERENCE MAP (R=2)

ENTRY POINTS   DEF LINE   REFERENCES
3 BLAST 1 9

VARIABLES  SN  TYPE  RELOCATION
0  B  REAL  F.P.  REFS  7  8  DEFINED  1
0  I  INTEGER  F.P.  REFS  7  8  DEFINED  1
0  IB  INTEGER  F.P.  DEFINED  1  7  8
0  P  REAL  F.P.  REFS  7  8  DEFINED  1
0  X  REAL  ARRAY  F.P.  REFS  6  7  8  DEFINED  1

STATISTICS
PROGRAM LENGTH  228  18
520000B CM USED
SUBROUTINE INTERP (BR,BD,BH,ROD,DDROD,HTG,IR1,IR2,PKS,PK,ND,NU)
C INTERPOLATES IN PK GRID TABLE.
C
DIMENSION PKS(40,20,8),ROD(8.41),DDROD(8,21),HTG(9)

XINT(A,B,C,D,E) = E + (D-E)*(B-C)/(B-A)
FOR EACH HEIGHT, FIND R,D BOUNDS WHICH BRACKET BURST
POINT.
P2 = -1.

INITIAL PASS FOR LOWER HEIGHT BOUND.
IH = IH1

CALL FIND (BR,ROD,ND,IH,IR1,IR2)
CALL FIND (BD,ROD,ND,IH,IR1,IR2)

SET UP INTERPOLATION PARAMETERS & INTERPOLATE
TO GET APPROXIMATE PK(FRAG).
R1 = -RU + ROD(IH,1)
IF(I1NE,0) R1 = ROD(IH,IR1)
R2 = RU + ROD(IH,IR1)
IF(I2NE,0) R2 = ROD(IH,IR2)
D1 = RU + ROD(IH,1)
IF(I1NE,0) D1 = ROD(IH,1)
D2 = RU + ROD(IH,ND)
IF(I2NE,0) D2 = ROD(IH,IC2)
IF(BR,LT,R1.OR,BR,GT,R2) GO TO 7
IF(BD,LT,0.6,OR,BD,GT,0.6) GO TO 7
IF(NDG.2.EQ.4) WRITE (6,*) "IR1,IR2,ID1,ID2,R1,R2,D1,D2 = *
C IR1,IR2,ID1,ID2,R1,R2,D1,D2

INTERPOLATE FOR BURST RANGE ALONG LOWER DEFINITION BOUND.
P1 = 0.
IF(I1EQ,0) GO TO 1
PR1 = 0.
IF(I1NE,0) PR1 = PKS(IR1,IR1,1)
PR2 = 0.
IF(I2NE,0) PR2 = PKS(IR2,IR1,1)
PD1 = XINT(R1,R2,BR,PR1,PR2)
IF(NDG.2.EQ.4) WRITE (6,*) "PR1,PR2,BR,PD1 = *
C

INTERPOLATE FOR BURST RANGE ALONG UPPER DEFINITION BOUND.
IF(I2EQ,0) GO TO 2
PR1 = 0.
IF(I1NE,0) PR1 = PKS(IR1,IR2,1)
PR2 = 0.
IF(I2NE,0) PR2 = PKS(IR2,IR2,1)
P1 = XINT(R1,R2,BR,PR1,PR2)
IF(NDG.2.EQ.4) WRITE (6,*) "PR1,PR2,BR,PD1 = *
C
SUBROUTINE INTERP 73/74 OPT=1

INTERPOLATE FOR BURST DEFLECTION ALONG BURST RANGE, LOWER
HEIGHT.

C

10 IF (IH.EQ.IH1) P1 = XINT(D1,D2,BD,PD1,PD2)
20 IF (HDBG.EQ.4) WRITE (6,*) "D1,D2,BD,P1 = ",D1,D2,BD,P1
30 IF (IH1.EQ.IH2) GO TO 5

65 INTERPOLATE FOR BURST DEFLECTION ALONG BURST RANGE, UPPER
HEIGHT.

C

70 IF (IH.EQ.IH2) P2 = XINT(D1,D2,BD,PD1,PD2)
80 IF (P2.NE.-1.) GO TO 3
90 IH = IH2
100 IF (IH2.EQ.0) GO TO 6

C REWRITE FOR UPPER HEIGHT BOUND.

C

120 P2 = 0.
130 IH2 = IH + 1

C INTERPOLATE FOR BURST HEIGHT.

C

150 PK = XINT(HGT(IH1),HGT(IH2),BD,P1,P2)
160 RETURN
170 PK = P1
180 RETURN
190 PK = 0.
200 END

SYMBOLIC REFERENCE MAP (R=2)

ENTRY POINTS

INTERP  DEF LINE REFERENCES

VARIABLES  SN  TYPE  RELOCATION  REFS  2=32  61  62  68

0  BD  REAL  F.P.  DEFINED  18
0  BH  REAL  F.P.  DEFINED  81
0  BR  REAL  F.P.  DEFINED  17
0  DGRD  REAL  ARRAY  F.P.  DEFINED  1
0  DU  REAL  F.P.  DEFINED  17
321  D1  REAL  F.P.  DEFINED  27
322  D2  REAL  F.P.  DEFINED  27
0  HGT  REAL  ARRAY  F.P.  DEFINED  6
315  ID1  INTEGER  REFS  18  3=28  33  39  41
316  ID2  INTEGER  REFS  18  2=30  33  50  51
312  IH  INTEGER  REFS  17  18  23  24  25
 28  29  30  41  43  52  54  61
<table>
<thead>
<tr>
<th>VARIABLES</th>
<th>SN</th>
<th>TYPE</th>
<th>RELOCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 IH1</td>
<td>1</td>
<td>INTEGER</td>
<td>F.P. REFS</td>
</tr>
<tr>
<td>0 IH2</td>
<td>2</td>
<td>INTEGER</td>
<td>F.P. REFS</td>
</tr>
<tr>
<td>313 IR1</td>
<td>3</td>
<td>INTEGER</td>
<td>REFS</td>
</tr>
<tr>
<td>314 IR2</td>
<td>4</td>
<td>INTEGER</td>
<td>REFS</td>
</tr>
<tr>
<td>0 ND</td>
<td>5</td>
<td>INTEGER</td>
<td>F.P. REFS</td>
</tr>
<tr>
<td>0 NDG</td>
<td>6</td>
<td>INTEGER</td>
<td>F.P. REFS</td>
</tr>
<tr>
<td>0 NM</td>
<td>7</td>
<td>INTEGER</td>
<td>F.P. REFS</td>
</tr>
<tr>
<td>0 NR</td>
<td>8</td>
<td>INTEGER</td>
<td>F.P. REFS</td>
</tr>
<tr>
<td>223 PD1</td>
<td>9</td>
<td>REAL</td>
<td>REFS</td>
</tr>
<tr>
<td>226 PD2</td>
<td>10</td>
<td>REAL</td>
<td>REFS</td>
</tr>
<tr>
<td>0 PK</td>
<td>11</td>
<td>REAL</td>
<td>F.P. REFS</td>
</tr>
<tr>
<td>0 PKS</td>
<td>12</td>
<td>REAL</td>
<td>ARRAY F.P. REFS</td>
</tr>
<tr>
<td>324 PR1</td>
<td>13</td>
<td>REAL</td>
<td>REFS</td>
</tr>
<tr>
<td>325 PR2</td>
<td>14</td>
<td>REAL</td>
<td>REFS</td>
</tr>
<tr>
<td>327 P1</td>
<td>15</td>
<td>REAL</td>
<td>REFS</td>
</tr>
<tr>
<td>311 P2</td>
<td>16</td>
<td>REAL</td>
<td>REFS</td>
</tr>
<tr>
<td>0 RGRD</td>
<td>17</td>
<td>REAL</td>
<td>ARRAY F.P. REFS</td>
</tr>
<tr>
<td>0 RU</td>
<td>18</td>
<td>REAL</td>
<td>F.P. REFS</td>
</tr>
<tr>
<td>317 R1</td>
<td>19</td>
<td>REAL</td>
<td>REFS</td>
</tr>
<tr>
<td>320 R2</td>
<td>20</td>
<td>REAL</td>
<td>REFS</td>
</tr>
</tbody>
</table>

FILE NAMES
- MODE TAPE6
- WRITE 33 45 56 62

EXTERNALS
- TYPE ARG S REFERENCES
  - FIND 6 17 18

INLINE FUNCTIONS
- TYPE ARG S DEF LINE REFERENCES
  - XINT REAL 5 SF 7 44 55 61 68 81

STATEMENTS LABELS
- DEF LINE REFERENCES
  - 125 1 46 39
  - 155 2 61 50
  - 213 3 81 69
  - 219 4 17 75
  - 225 5 83 63
  - 220 6 76 71
  - 227 7 65 31 32

STATISTICS
- PROGRAM LENGTH 345B 229
SUBROUTINE FIND 73/74 OPT=1 FNN 4.8+50B 03/13/31 08.29.30 PAGE 1

1
SUBROUTINE FIND (B,GRD,N,IH,IX1,IX2)
C
C FINDS BOUNDS OF BURST POINT IN PK GRID.
C IX1,IX2 ARE ARRAY ELEMENTS WHICH BRACKET
SUBJECT COORDINATE.
C
DIMENSION GRD(B,41)
IF(B.LT.GRD(IH,1)) GO TO 1
IF(B.GT.GRD(IH,N)) GO TO 2
DO 3 I=2,N
IX2 = I
IF(B.LT.GRD(IH,1)) GO TO 4
3 CONTINUE
4 IX1 = IX2 - 1
RETURN
1 IX2 = '1
IX1 = 0
RETURN
2 IX2 = 0
IX1 = N
END

SYMBOLIC REFERENCE MAP (R=2)
ENTRY POINTS DEF LINE REFERENCES
3 FIND 1 15 18 21

VARIABLES SN TYPE RELOCATION
0 B REAL F.P. REF'S 8 9 12 DEFINED 1
0 GRD REAL ARRAY F.P. REF'S 7 8 9 12 DEFINED 1
37 I INTEGER F.P. REF'S 11 12 DEFINED 10
0 IH INTEGER F.P. REF'S 8 9 12 DEFINED 1
0 IX1 INTEGER F.P. DEFINED 1 10 17 20
0 IX2 INTEGER F.P. REF'S 14 DEFINED 1 11 16 19
0 N INTEGER F.P. REF'S 9 10 20 DEFINED 1

STATEMENT LABELS DEF LINE REFERENCES
30 1 16 6
33 2 10 9
0 3 13 10
25 4 14 12

LOOPS LABEL INDEX FROM-TO LENGTH PROPERTIES
16 3 1 10 13 7F INSTACK EXITS

STATISTICS
PROGRAM LENGTH 468 38
52000B CM USED
DISTRIBUTION LIST

Commander
U.S. Army Armament Research
and Development Command
ATTN: DRDAR-LCS, R. Corn
       DRDAR-LCU, A. Moss
       DRDAR-LCE-M, A. King
       DRDAR-LCU-CA, D. Costa
       DRDAR-LCS-A, J. Brooks (6)
       DRDAR-LCS-E, S. Einbinder
       DRDAR-TSS (5)
       DRDAR-GCL
Dover, NJ 07801

Project Manager
Cannon Artillery Weapons System
ATTN: DRCPM-CAWS-El, G. Waldron
Dover, NJ 07801

Director
U.S. Army Materiel Systems
   Analysis Activity
   ATTN: DRXSY-GS, J. Chernick
         DRXSY-MP
Aberdeen Proving Ground, MD 21005

Commandant
U.S. Army Field Artillery School
ATTN: ATSF-CD-AD, R. Wilde
Fort Sill, OK 73503

Director
U.S. Army TRASANA
ATTN: ATAA-TBA, J. Hodde
White Sands Missile Range, NM 88002

Commander
Naval Surface Weapons Center
ATTN: Code G29, R. Scheibe
Dahlgren, VA 22448

Commander
U.S. Army Electronics Research
   and Development Command
   ATTN: DELHD-R-RSD, R. Parkhurst
Adelphi, MD 20783

Commander
Naval Weapons Center
ATTN: Code 3524, J. Netzer
China Lake, CA 93555
Administrator
Defense Technical Information Center
ATTN: Accessions Division (12)
Cameron Station
Alexandria, VA 22314

Commander/Director
Chemical Systems Laboratory
U.S. Army Armament Research
and Development Command
ATTN: DRDAR-CLJ-L
       DRDAR-CLB-PA
APG, Edgewood Area, MD 21010

Director
Ballistics Research Laboratory
U.S. Army Armament Research
and Development Command
ATTN: DRDAR-TSB-S
Aberdeen Proving Ground, MD 21005

Chief
Benet Weapons Laboratory, LCWSL
U.S. Army Armament Research
and Development Command
ATTN: DRDAR-LCB-TL
Watervliet, NY 12189

Commander
U.S. Army Armament Materiel
Readiness Command
ATTN: DRSAR-LEP-L
Rock Island, IL 61299

Director
U.S. Army TRADOC Systems
Analysis Activity
ATTN: ATAA-SL
White Sands Missile Range, NM 88002