US ARMY
MATERIEL
COMMAND



776

AD A169610

TECHNICAL REPORT BRL-TR-2738

# A SHOTLINE METHOD FOR MODELING PROJECTILE GEOMETRY

Paul J. Tanenbaum

June 1986

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

US ARMY BALLISTIC RESEARCH LABORATORY
ABERDEEN PROVING GROUND, MARYLAND

# REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

| | | |
|---|---|---|
| 1. REPORT NUMBER<br>Technical Report BRL-TR-2738 | 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle)<br>A SHOTLINE METHOD FOR MODELING PROJECTILE GEOMETRY | | 5. TYPE OF REPORT & PERIOD COVERED<br>FINAL |
| | | 6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s)<br>Paul J. Tanenbaum | | 8. CONTRACT OR GRANT NUMBER(s) |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>US Army Ballistic Research Laboratory<br>ATTN: SLCBR-VL<br>Aberdeen Proving Ground, MD 21005-5066 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS<br>1L162618AH80 |
| 11. CONTROLLING OFFICE NAME AND ADDRESS<br>US Army Ballistic Research Laboratory<br>ATTN: SLCBR-DD-T<br>Aberdeen Proving Ground, MD 21005-5066 | | 12. REPORT DATE<br>June 1986 |
| | | 13. NUMBER OF PAGES<br>69 |
| 14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) | | 15. SECURITY CLASS. (of this report)<br>UNCLASSIFIED |
| | | 15a. DECLASSIFICATION DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

GIFT
Vulnerability Analysis
Shotlining
MISFIR
Target Description
COMGEOM
Warhead Lethality

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Most vulnerability/lethality programs represent the path of a penetrator through a target by a one-dimensional shotline. This approach is insufficient for projectiles with shoulder-fuzed contact or influence fuzes. It also fails to treat effectively discontinuities, obliquities, and small components found in typical targets. This report presents a modification of the shotline method in which the projectile is represented by a bundle of planetary rays disposed around the main ray. The program, MISFIR, computes effective

standoffs for each cell. MISFIR will be critical in determining the lethality of warheads with modern fuze design. An applications program, FUZES, which treats issues concerning a hypothetical projectile, is also discussed. Source listings and sample output are provided in appendixes. The sample run illustrates the importance of considering the 3-D geometry of the projectile/ target system.

# PREFACE

I should like to express my appreciation to all the people at BRL whose help greatly facilitated my work on this project. Notable among them are Gary Kuehl, for providing the BRANDX subroutine on which SHOTCYL was based; Howard Ege, author of SILPK, which is the ancestor of SILOET; and Claude Lapointe and Robert Wilson, for help in tracking down the more egregious bugs in my thinking and coding.

This page Left Intentionally Blank

## TABLE OF CONTENTS

This page Left Intentionally Blank

## LIST OF ILLUSTRATIONS

This page Left Intentionally Blank

# I. INTRODUCTION

Most current vulnerability/lethality programs represent a penetrator's path through a target by a shotline. This shotline is constructed, at least conceptually, by passing a ray through a geometric description of the target and recording each target component that the ray intersects and at what distance from its origin the ray enters and exits the component. The analyst will typically select azimuth and elevation angles from which to view the target, and then superpose over the view of the target (hereafter referred to as *V*) a rectilinear grid, as shown in figure 1. From a selected location with each cell in this grid a ray will be sent through the target to generate a shotline. The aggregate of these shotlines is then used in determining the entire target's vulnerability.

One drawback of the shotline approach is that, whereas many man-months might be spent on a detailed description of the geometry of the target, the projectile can only be represented by a straight-line trajectory, an abstraction with zero cross-sectional area. Of course, if the projectile's geometry and fuze design are accommodating, and in the absence of yaw and other complications, this limitation is not too serious for planar targets of infinite extent. But for some projectiles — either with piezoelectric contact fuzes on probes or with inductance- or capacitance-type influence fuzes — the single-ray approach is insufficient. Even such common target characteristics as discontinuities and high obliquities cause problems for zero-width modeling of projectiles.

This report presents a modification of the standard shotline method. A computer routine called SHOTCYL provides a means of modeling the three-dimensional geometry of projectile/target interaction. The projectile is represented by a main, or central, ray together with one or more rings of planetary rays disposed parallel to and at specified distances from the main ray.

This representation has been used in analyzing the behavior of several HEAT rounds. In such a warhead, the charge is mounted some distance behind the fuze. This built-in standoff is intended to allow the penetrating jet to form before it strikes the target. A jet's penetration into armor is sensitive to standoff, so any abnormal impact (such as that illustrated in figure 2) might significantly degrade a round's performance.

Given a bundle of rays for each grid cell, a program called MISFIR computes the effective standoff for each cell, using the built-in standoff and the geometrical details of the projectile's impact on the target. This provides for more realistic treatment of several projectile designs and of such target-surface properties as obliquity and edges.

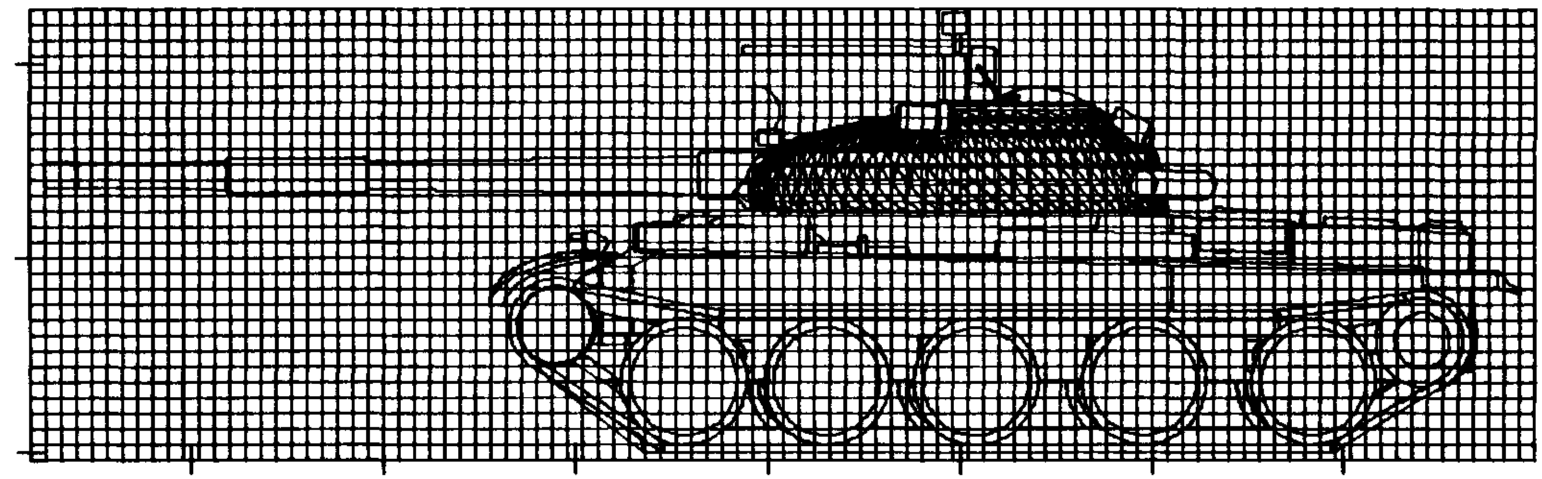


Figure 1.—*View of a Target with Grid Superposed.*

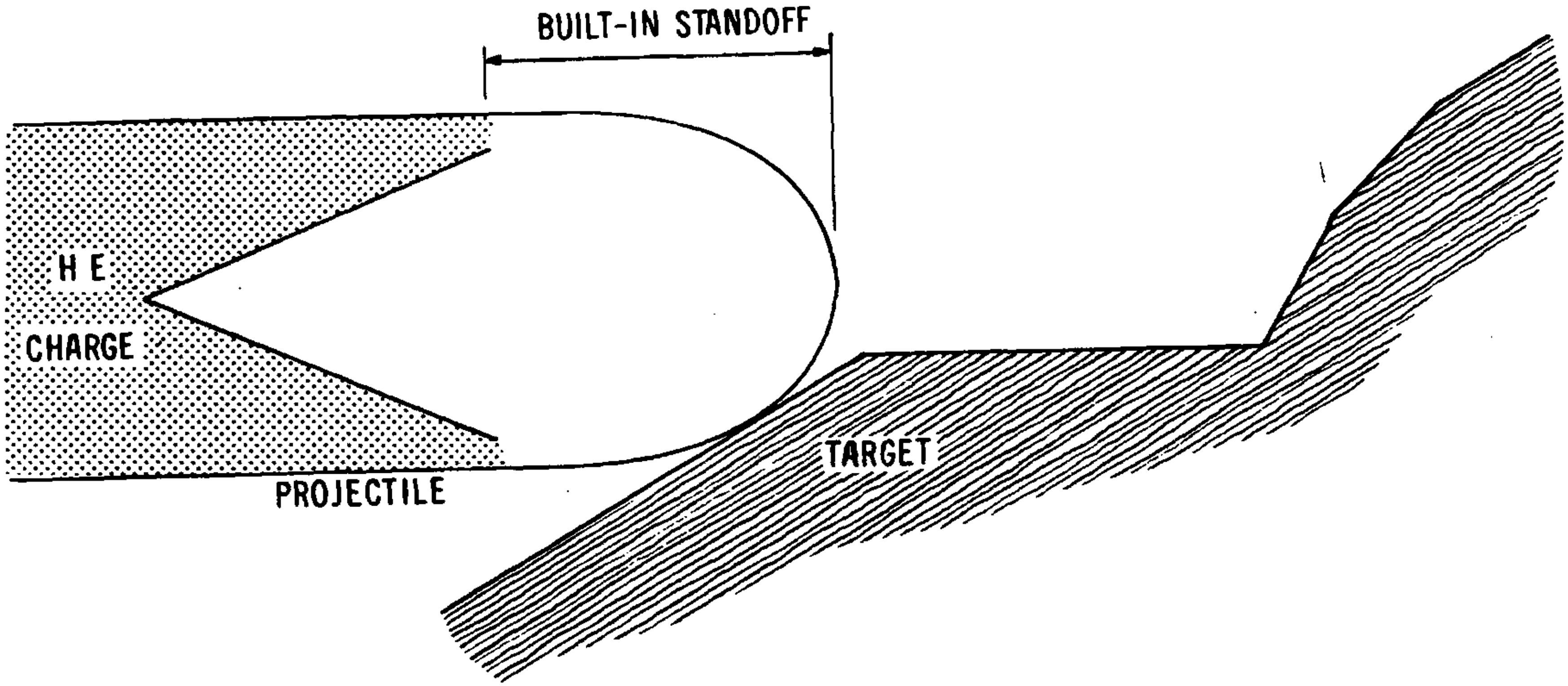



FIGURE 2.—*Edge Hit of a Shaped Charge Projectile on an Armored Target.*

Also discussed here is the employment of the new method in an analysis of the effectiveness of a hypothetical two-fuzed missile. The missile has a primary fuze that is mounted on a long probe and a secondary fuze further back on the shoulder of the missile, as shown in figure 3. The modified shotline method was used to answer two questions:

> 1) For a given projectile, target, view, and aimpoint, what is the probability that the secondary fuze will strike the target before the primary fuze?
>
> 2) Under the same conditions, what is the probability that the secondary fuze will strike the target after the primary fuze, but during the primary's built-in delay?

## II. PROJECTILE MODELING TECHNIQUE

The key input for the enhanced method is a description of the projectile. Depending on the nature of the projectile, this three-dimensional model will represent the projectile's fuze, its nose, its influence envelope, or whatever the relevant volume. The following paragraphs provide a conceptual explanation of the model.

Consider an arbitrary solid *S* traveling along a straight-line trajectory, *T*. The volume it sweeps out is a cylindroid. We add a frame of reference whose origin, *O*, is at the front tip of *S*. Now let us make the assumption that the solid is radially symmetric about *T*, and that its maximum circumference, *C*, occurs at some distance *D* along *T* (see figure 4). A snapshot (ignoring the early part of the trajectory) reveals that the shape of the volume swept out is that of a right circular cylinder capped with the patch, *P*, of *S*'s surface that is bounded by *C* and contains *O*.

Any collision between *S* and a stationary object will take place at a point on *P*. This patch is the crucial part of our model, and can be approximated by a set of circles that are centered on *T*, each circle being specified by its radius and its stepback — the distance from *O* to the circle's center. As the solid travels, the circles sweep out concentric cylinders. These cylinders, as they exist at any specified instant, can be represented by rings of rays originating on the circles and extending backwards.

In principle this is how a projectile is modeled. The program determines at what point each ray intersects the target and, considering the contours of the projectile's leading surface (as rendered by the model), decides where and how the

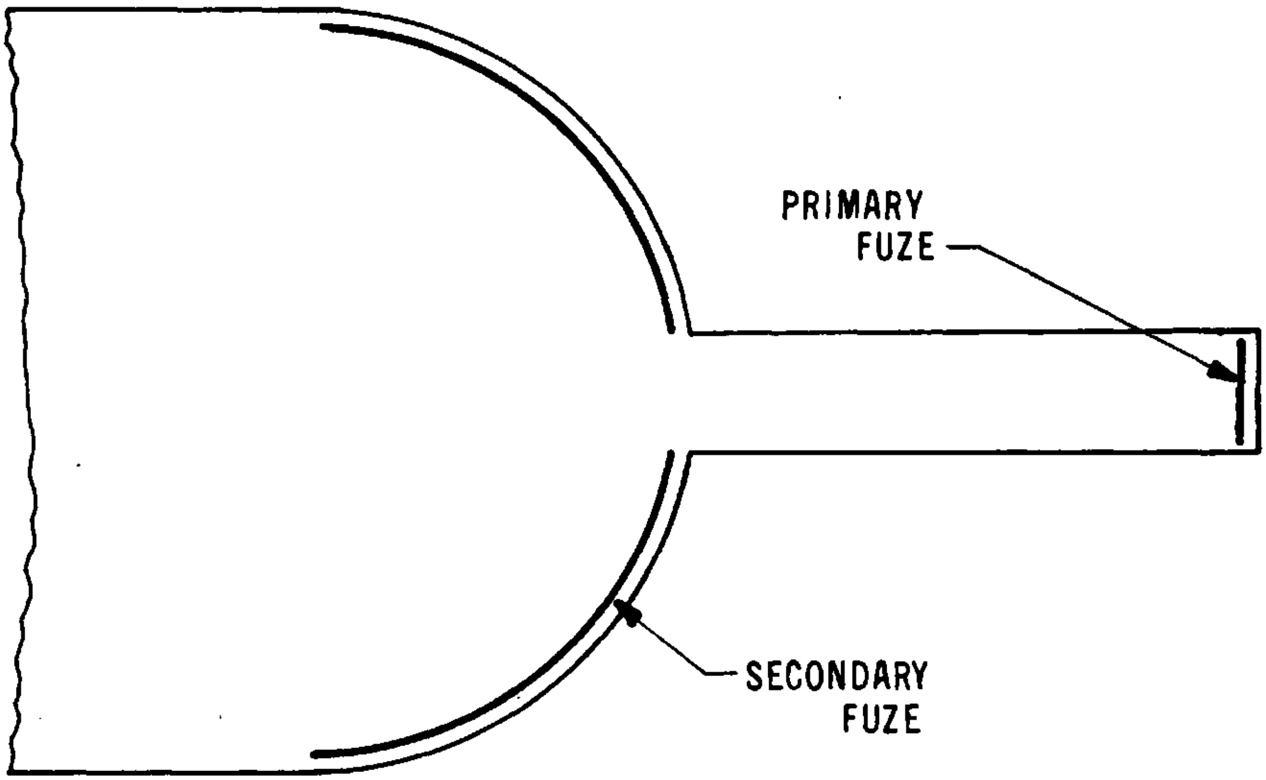



FIGURE 3.—*Hypothetical Two-Fuzed Missile.*

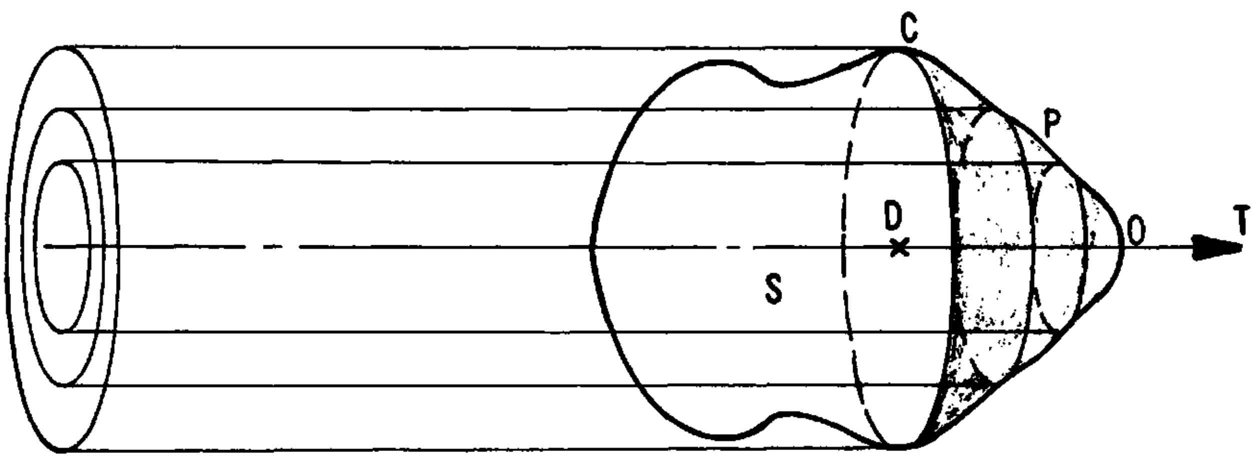



FIGURE 4.—*Flight of a Radially Symmetric Solid,* S, *Modeled as a Bundle of Rays.*

projectile first touches the target. This, together with the projectile's built-in standoff, provides enough information to compute the effective standoff for the hit being considered.

## III. SYNOPSIS OF ALGORITHM

The three-dimensional model of projectile/target interaction is centered upon the program MISFIR, written in CDC Fortran 5. MISFIR is built on the formalisms of the GIFT (*G*eometric *I*nformation *f*or *T*argets)* program, which processes targets described in terms of combinatorial geometry, or COMGEOM. The MISFIR package currently consists of a ray-tracing subroutine added to GIFT (viz. SHOTCYL); MISFIR itself, together with its subprograms; and an application program, called FUZES, which uses MISFIR's results to solve a typical problem in vulnerability analysis. A generalization of the familiar GIFT shotline provides the means by which MISFIR represents projectile geometry: The single ray is augmented by a bundle of rays, the combination being called a shotcylinder. The planetary rays are grouped around the central ray in orbits that, depending on their radii, can extend into adjacent grid cells.

The first step is to create shotcylinders for each cell in $V$, the current view of the target. This is done by the GIFT subroutine called SHOTCYL. Next the shotcylinders are used to compute the projectile's actual standoff in each cell. MISFIR then creates histograms of the standoff and a silhouette image of the target. In the application described here, FUZES is then run to analyze the effectiveness of the hypothetical warhead. Below are brief descriptions of each step in the process.

The user's input to SHOTCYL includes the angle of attack, expressed in terms of azimuth and elevation, and such ray-bundle building parameters as the number of orbits [layers] per bundle, the orbits' radii, and the number of planets [rays] in each orbit. The frame of reference for GIFT's calculations is defined by the grid plane, the plane that contains the origin of the target description's coordinate system and is normal to the line of sight, or attack angle. The grid plane is partitioned into rectilinear cells, and the target's projection onto it determines the cells for which bundles must be created.

---

* Kuehl, G.G., Bain, L.W., Jr., and Reisinger, M.J. *The GIFT Code User Manual.* BRL R1802, July 1975, and ARBRL-TR-02189, September 1979.

Also input to SHOTCYL is the manner of selecting the central ray's location within the grid cell. SHOTCYL can use an impact-point file to treat a target that has already been shotlined*. Alternatively, SHOTCYL can pass its central rays through either the center of, or a point chosen at random in, each grid cell. For each grid cell, SHOTCYL outputs the following information:

| variable names | contents |
|---|---|
| *HCEN, VCEN* | The grid-plane coordinates of the center of the cell |
| *IHIV* | A code for the location of the central ray within the cell |
| *H, V* | The grid-plane coordinates of the central ray |
| *RAYLEN* | The distance along the central ray from the grid plane back to the first item met. Thus, this number is negative for cells in which the central ray does not encounter the target until it has passed through the grid plane. |
| *ITMHIT* | The item number of the first item met by the central ray, or -1 if it never hits the target. |

And for each planetary ray:

| | |
|---|---|
| *PLNETH, PLNETV* | The grid-plane coordinates of the ray |
| *RAYLEN* | The distance along the ray from the grid plane back to the first item met. Thus, this number is negative for planetary rays which do not encounter the target until they have passed through the grid plane. |
| *ITMHIT* | The item number of the first item met by the ray, or -1 if it never hits the target. |

* High resolution in shotlining the typical target is not always computationally feasible. The relatively large grid cells resultant from this constraint render many applications programs very sensitive to ray location within a cell. In order to control for positional variation, Robert Wilson has developed a technique of recording cell impact points in a file parallel to the shotline file. When a target must be shotlined repeatedly, as in a parametric analysis, the ray coordinates for each cell can be reused, thus maintaining comparability.

SHOTCYL requires one subprogram that is not found in the GIFT or system libraries. It is a routine called SEEKVEW that scans through an impact-point file to find the data for a specified target aspect.

MISFIR is the centerpiece of the package. It produces its results — information about projectile impact for each cell on the target — in several forms. The subprograms called by MISFIR are CPA, PHIT, HISTOG, and SILOET.

CPA selects as the aimpoint the target's *c*enter of *p*resented *a*rea. This is equivalent to the centroid of $V$, and can be calculated as:

$$(\bar{X}, \bar{Y}) = \left( \frac{1}{N} \sum_{c \text{ in } V} X_c \, , \; \frac{1}{N} \sum_{c \text{ in } V} Y_c \right) \qquad (1)$$

where $N$ is the number of cells, $c$, in $V$. This aimpoint is then offset by $(X_{dc}, Y_{dc})$ to allow for dispersion corrections provided by the user.

Once the aimpoint has been computed, MISFIR determines which ray in each cell strikes the target first and calculates the size of the gap between projectile tip and target at the instant of impact (see figure 5). The basic formula for this gap is:

$$GAP = \max \left(0 \, , \; \max_{p \text{ in } B} \left(DPLNET_p - DMAIN - CSTPBK_p - CSTPBK_0\right)\right) \qquad (2)$$

where $p$ is a planetary ray in $B$, the current cell's bundle, $DPLNET_p$ is the distance along $p$ from the grid plane back to the first planet met, $DMAIN$ is the distance along the central ray from the grid plane back to the first item met, $CSTPBK_p$ is the stepback of $p$'s cylinder, and $CSTPBK_0$ is the length of a needle probe of negligible diameter.

For cells in which the primary fuze strikes the target before the secondary fuze, MISFIR also determines whether the secondary fuze will strike during the primary fuze's delay. Then MISFIR determines the warhead's actual standoff for that cell — the projectile's built-in standoff plus $GAP$ — and the probability of the cell's being hit ($P_H$). The cell's $P_H$ is computed by the function PHIT, which assumes that delivery errors are normally distributed.

HISTOG computes and displays two histograms. To do this, it partitions into bins the range of standoffs that were encountered in $V$. The first histogram gives the number of cells whose standoffs lie within each bin. The second displays similar data obtained by choosing an aimpoint on the target and using the weapon's delivery-accuracy characteristics to weight the standoff obtained in each cell by that cell's $P_H$. Thus, the histograms convey the relative frequencies of various values of

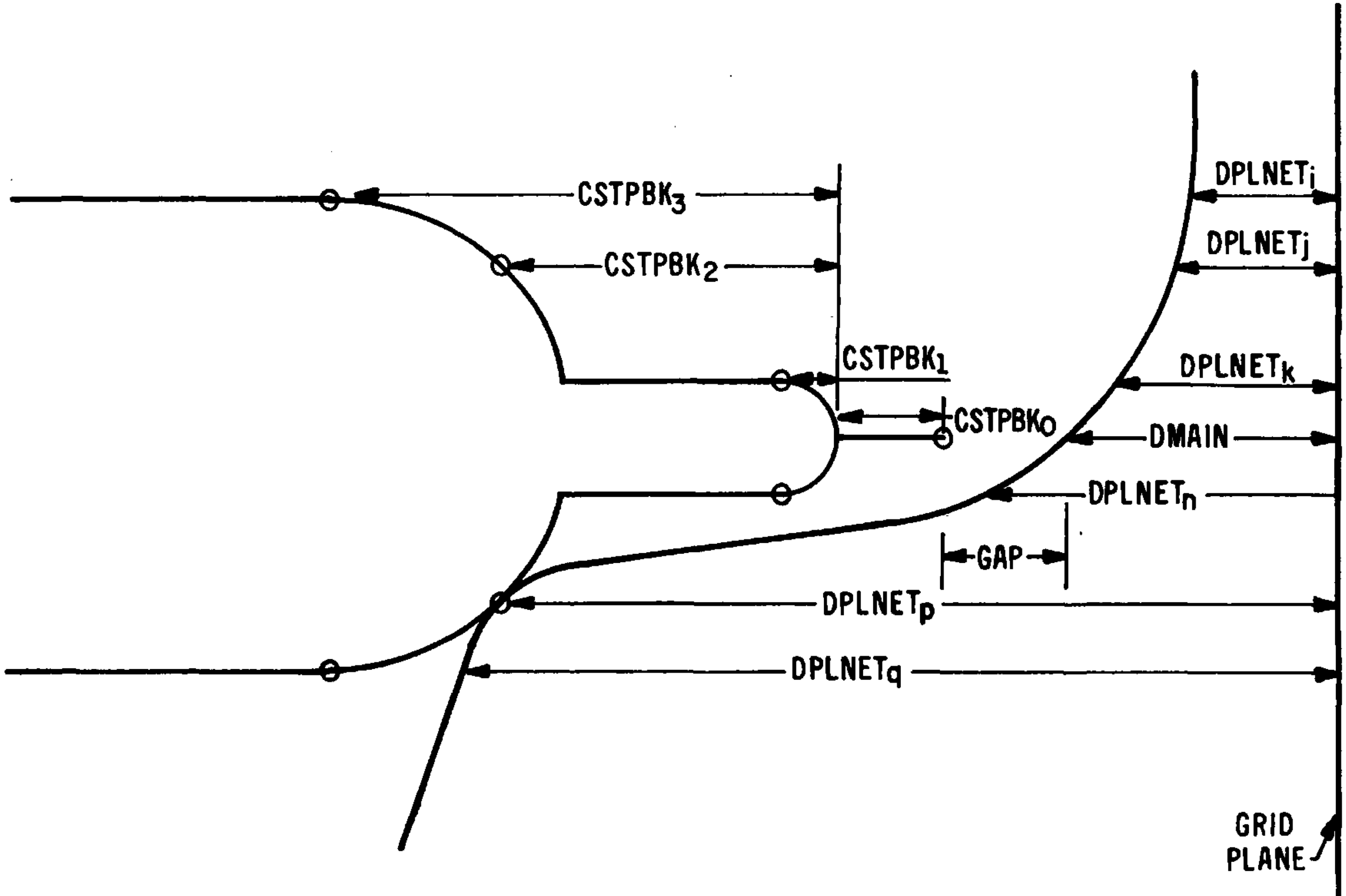



FIGURE 5.—*Geometry of MISFIR's Calculation of* GAP.

standoff, either assuming a random hit anywhere on the target or assuming a shot aimed at its centroid. Both histograms also include a bin for edge hits, cells where at least one planetary ray intersects the target, but the main ray misses completely. SILOET* displays an image of the non-empty portion of the grid plane, where each cell is represented by standoff to two decimal places, and in units of tenths of a charge diameter. MISFIR also prints all its results to a file for use by other programs. The information produced for each cell is:

| variable names | contents |
|---|---|
| *HCEN, VCEN* | The grid-plane coordinates of the cell. |
| *DIST* | The effective standoff for the cell. |
| *CYL1* | The number of the cylinder containing the ray that struck the target first. |
| *FZ2HIT* | A Boolean flag indicating whether both the primary fuze struck the target first and the secondary fuze struck during the primary fuze's delay. |
| *CELLPH* | The cell's $P_H$. |

FUZES provides information about the performance of the hypothetical missile by estimating the conditional probabilities of the two events

$E_1$: The secondary fuze strikes the target before the primary fuze does.

and

$E_2$: The secondary fuze strikes the target after the primary fuze does, but during the primary fuze's built-in delay.

both given a uniform-random hit on the target, and given that a shot was fired at the specified aimpoint. Computing the probabilities of $E_1$ required only one modification to MISFIR: the value of an existing variable (viz. *CYL*1) would have to be added to the output. $E_2$ induced a slightly larger task. Several lines of code had to be added to MISFIR to handle the computation and the printing of the value of *FZ2HIT*.

* Based on H. Ege's SILPK program

The probability of $E_1$ given a hit, $\mathbf{P}(E_1 \mid H)$, is simply the ratio of the number of cells in which the secondary fuze struck first (i.e. the number of cells for which *CYL*1 exceeded some user-specified threshold) to the the number of cells in $V$. The probability of $E_1$ given a shot, $\mathbf{P}(E_1 \mid S)$, is the sum of the $\mathbf{P}_H$'s of all the cells in which the secondary fuze struck first.

FUZES computes the probability of $E_2$ given a hit by using the formula

$$\mathbf{P}(E_2 \mid H) = \sum_{c \text{ in } V} FZ2HIT. \tag{3}$$

Similarly, $\mathbf{P}(E_2 \mid S)$ is the sum of the $\mathbf{P}_H$'s of all the cells for which $FZ2HIT = 1$.

The source code for SHOTCYL, MISFIR, FUZES, and their subprograms can be found in appendix A. Appendix B provides a sample run of the package. The histograms on pages 60 and 61 indicate that, whether the hits were distributed uniformly or normally, the preponderance of hits occurred at built-in standoff. They also show that, although there was about a 13% probability of achieving an edge hit given a random hit, when delivery accuracy was considered, the probability was reduced by an order of magnitude. The silhouette on page 62 illustrates these points for each cell. For the numbers in the sample run, the very common silhouette cell value of 25 represents 2.5 charge diameters, or 250 mm, which is the projectile's built-in standoff. FUZES' output, page 63, indicates that $\mathbf{P}(E_1 \mid H) \approx 22\%$, and $\mathbf{P}(E_1 \mid S) \approx 12\%$, while $\mathbf{P}(E_2 \mid H)$ is about 8%, and $\mathbf{P}(E_2 \mid S)$ is about 5%.

## IV. CONCLUSIONS

Initial runs of the SHOTCYL-MISFIR-FUZES package pitting the hypothetical missile against one important target suggest that the likelihood, and therefore the importance, of secondary-fuze detonation can be quite significant. More thorough testing — considering several projectiles and targets — could perhaps settle the issue more conclusively. In any event, the MISFIR-FUZES model should answer the question easily and cheaply. It ought therefore to be useful for designers of warheads and armored systems, and for those of us who assess the vulnerability of existing systems. BRL will be using MISFIR routinely in $\mathbf{P}_K$ analyses.

Additions and modifications to the program will no doubt be required in time. Some changes that are already under consideration are:

1) To allow the number of planetary rays per orbit to vary within the bundle.

2) To allow the option that planetary rays be positioned at random around the orbits. MISFIR currently spaces planetary rays evenly around the orbit.

3) To change the format of the histogram output for clarity and more precision.

This page Left Intentionally Blank

# APPENDIX A

## Source Listings

This page Left Intentionally Blank

```
      SUBROUTINE BRANDX
C
C     A.K.A.     SHOTCYL
C
C     COMPUTES FIRST INTERSECT FOR CYLINDER(S) AROUND A MAIN GRID RAY
C
C     DERIVED FROM GARY KUEHL'S 'GRID' SUBROUTINE,
C     SIGNIFICANTLY EMBELLISHED BY PAUL TANENBAUM
C
C     CONTROL CARD (3I5)
C        1-5  NMASPS          - NUMBER OF VIEWS FOR CYLINDER
C                                 (CURRENTLY MUST BE 1)
C        6-10 NOPRNT          - .NE. 0 ==> SUPPRESS PRINT OF UNIT 1 DATA
C       11-15 RAYLOC          - LOCATION OF RAY WITHIN GRID CELL
C                                 (.EQ. 0 ==> RANDOM POINT IN CELL)
C                                 (.EQ. 1 ==> READ FROM RAYFILE)
C                                 (.GE. 2 ==> CENTER OF CELL)
C
C     PROJECTILE CARD (A20, 2I5, 2F7.2, 2F8.3, I5)
C        1-20 RUNNAM          - CHAR STRING TO IDENTIFY RUN
C       21-25 NMCYLS          - NUMBER OF CYLINDERS
C                                 (1 <= NMCYLS <= 10, DEFAULT = 2)
C       26-30 NPLNTS          - NUMBER OF RAYS PER CYLINDER
C                                 (1 <= NPLNTS <=  50,  DEFAULT = 8)
C       31-37 CD              - CHARGE DIAMETER  [MM]
C       38-44 STDOFF          - BUILT-IN STANDOFF  [MM]
C       45-52 VEL             - VELOCITY  [M/S]
C       53-60 DELAY           - BUILT-IN DELAY OF PRIMARY FUZE [SEC]
C       61-65 FZ1MXC          - NUMBER OF LARGEST CYLINDER IN PRIMARY FUZE
C
C     PROJECTILE-SKIN CARDS (3F10.2)     [ONE CARD PER CYLINDER]
C        1-10 CRAD(N)         - RADIUS OF NTH CYLINDER  [MM]
C       11-20 CSTPBK(N)       - STEPBACK OF NTH CYLINDER  [MM]
C       21-30 CSTPBK(0)       - LENGTH OF ZERO-CROSS-SECTIONAL-AREA
C                               PROBE  [MM]
C                                 (LAST PROJECTILE-SKIN CARD ONLY)
C
C     DISPERSION CARD (4F8.3)
C        1-8  SIGX            - X-DISPERSION  [MM]
C        9-16 SIGY            - Y-DISPERSION  [MM]
C       17-24 XDC             - X DISPERSION CORRECTION  [MM]
C       25-32 YDC             - Y DISPERSION CORRECTION  [MM]
C
C     ASPECT CARDS (3F10.0, 3I5)     [ONE CARD PER VIEW]
C        1-10 AZIM            - AZIMUTH ANGLE  [DEGREES]
C       11-20 ELEV            - ELEVATION ANGLE  [DEGREES]
C       21-30 CELSIZ          - SIZE OF CELLS IN GRID PLANE  [MM]
C                               (DEFAULT = 4 IN)
C       31-35 IDLKP           - DELETE SELECTED REGIONS
C       36-40 ILIMIT          - LIMIT GRID PLANE
C       41-45 ISPOT           - NUMBER OF SELECTED SHOTLINES
C
C     LIMIT GRID PLANE CARD (4F10.0)     [IF ILIMIT.NE.0]
C        1-10  HMIN           - HORIZONTAL MINIMUM OF GRID PLANE
C       11-20  HMAX           - HORIZONTAL MAXIMUM OF GRID PLANE
C       21-30  VMIN           - VERTICAL MINIMUM OF GRID PLANE
C       31-40  VMAX           - VERTICAL MAXIMUM OF GRID PLANE
C
```

```
C      GRID PLANE COORDINATES CARDS (8F10.0)     (IFF ISPOT .GT. 0)
C         1-10  CH(1)          - HORIZONTAL COORDINATE OF POINT 1
C        11-20  CV(1)          - VERTICAL COORDINATE OF POINT 1
C         ...
C        71-80  CV(4)          - VERTICAL COORDINATE OF POINT 4
C

      CHARACTER *10 DAY
      CHARACTER *60 ITITLE
      CHARACTER *20 RUNNAM
      INTEGER ASPECT, CYL, F21MXC, I, IOLKP, IDUM, IHIV, IHORZ, ILIMIT,
     1        INPCEL, IRANH, IRANV, IREGON, ISEED, ISEED1, ISOLID,
     2        ISPOT, ISURF, ITMHIT, IVERT, J, L, LENRPP, LIRFO, NCELL,
     3        NHORZ, NMASPS, NMCYLS, NMHITS, NOPRNT, NPLNTS, NVERT,
     4        PLANET, RAYLOC
      LOGICAL MKRAYS, PRTALL
      REAL ABSCA, ABSCE, ABSSA, ABSSE, ANGINC, ASTER, AZIM, AZIMR,
     1     BACK, CAZIM, CD, CELEV, CELSIZ, CH, CRAD, CSTPBK, CV,
     2     DEG2RAD, DELAY, DEPTH, DFIRST, ELEV, ELEVR, ENGTH, EOV,
     3     ETIME, GDUM, H, HC, HCEN, HMAX, HMIN, HORZ, MMCVT, PLNETH,
     4     PLNETV, RAYLEN, RFA, RFCS, RFE, SANGLE, SAZIM, SELEV, SIGX,
     5     SIGY, SJIGLE, STOOFF, STIME, T, TCEN, TGTCVT, TLEN, TMAX,
     6     TMIN, V, VC, VCEN, VEL, VERT, VMAX, VMIN, VTIME, WB, WP, XB,
     7     XDC, YDC
      LEVEL 2,//
      COMMON /ENRPP / LENRPP
      COMMON /GEOM  / GDUM(4), LIRFO
      COMMON /ONEINT/ DFIRST, ISOLID, ISURF, IREGON
      COMMON /RAYPAR/ XB(3), WB(3)
      COMMON /SEED  / ISEED
      COMMON /TGTCVT/ TGTCVT
      COMMON /TITLE / ITITLE
      DIMENSION CH(4), CRAD(10), CSTPBK(0:10), CV(4),
     1          ITMHIT(0:10, 0:50), RAYLEN(0:10, 0:50), PLNETH(10, 50),
     2          PLNETV(10, 50), TMIN(3), TMAX(3), TCEN(3), TLEN(3),
     3          WP(3)
      COMMON ASTER(5000)
      DATA DEG2RAD /.017453292519943/
      DATA SJIGLE /.0001/

C ----    BEGIN EXECUTION    ----------------------------------------

      MMCVT = 25.4 / TGTCVT
C ----    TGTCVT = 1 IF TARGET DESCRIPTION IS IN INCHES,
C                = 25.4 IF TARGET DESCRIPTION IS IN MM.
C         THEREFORE, DIVIDING A LENGTH IN MM BY MMCVT WILL CONVERT
C         THE LENGTH TO THE UNITS OF THE TARGET DESCRIPTION.
C         CONVERSELY, MULTIPLYING A LENGTH IN THE DESCRIPTION'S
C ----    UNITS BY MMCVT WILL CONVERT THE LENGTH TO MM.
      CALL CLOCKS(STIME)

C ----    READ CONTROL CARD
      READ (*, 5010, ERR=160, END=170) NMASPS, NOPRNT, RAYLOC
C ----    NMASPS MUST BE .EQ. 1... IF NOT, ABORT
      IF (NMASPS .NE. 1) THEN
        WRITE (*, 6010)
        STOP
      ENDIF
```

```
C ----    DEFINE PRTALL, AND IF IT'S FALSE, WARN AS MUCH
      PRTALL = NOPRNT .EQ. 0
      IF(.NOT. PRTALL) THEN
        WRITE (*, 6020)
      ENDIF

C ----    READ PROJECTILE DATA
      READ (*, 5020, ERR=160, END=170) RUNNAM, NMCYLS, NPLNTS, CD,
     1                                 STDCFF, VEL, DELAY, FZ1MXC
C ----    ENSURE THAT FZ1MXC IS NO LARGER THAN NMCYLS
      IF (FZ1MXC .GT. NMCYLS) THEN
        WRITE (*, 6030) FZ1MXC, NMCYLS
        STOP
      ENDIF
C ----    ENSURE THAT 1 <= NMCYLS <= 10
      IF (NMCYLS .LE. 0) THEN
        NMCYLS = 2
      ELSE
        NMCYLS = MIN(NMCYLS, 10)
      ENDIF
C ----    ENSURE THAT 1 <= NPLNTS <= 50
      IF (NPLNTS .LE. 0) THEN
        NPLNTS = 8
      ELSE
        NPLNTS = MIN(NPLNTS, 50)
      ENDIF

C ----    READ CYLINDER RADII AND STEPBACKS.
C ----    IF RADII ARE NOT POSITIVE, IN NON-DECREASING ORDER, ABORT.
      DO 10 CYL = 1, NMCYLS
        IF (CYL .EQ. NMCYLS) THEN
          READ (*, 5030, ERR=160, END=170) CRAD(CYL), CSTPBK(CYL),
     1                                     CSTPBK(0)
        ELSE
          READ (*, 5030, ERR=160, END=170) CRAD(CYL), CSTPBK(CYL)
        ENDIF
        CRAD(CYL) = CRAD(CYL) / MMCVT
        CSTPBK(CYL) = CSTPBK(CTL) / MMCVT
   10 CONTINUE
      CSTPBK(0) = CSTPBK(0) / MMCVT
      DO 20 CYL = 1, NMCYLS - 1
        IF ((CRAD(CYL) .EQ. 0) .OR.
     1      (CRAD(CYL) .GT. CRAD(CYL + 1))) THEN
          WRITE (*, 6040)
          STOP
        ENDIF
   20 CONTINUE
C ----    DEFINE MKRAYS
      MKRAYS = (RAYLOC .NE. 1)

C ----    READ DISPERSION DATA
      READ (*, 5040, ERR=160, END=170) SIGX, SIGY, XDC, YDC
```

```
      CALL DATE(DAY)
      REWIND 1
      IF (PRTALL) THEN
        WRITE (*, 6050) NMASPS, DAY, ITITLE
        WRITE (*, 6060) PRTALL, RUNNAM, CD, STDOFF, VEL, DELAY, FZIMXC
        WRITE (*, 6070) SIGX, SIGY, XDC, YDC
      ENDIF
      WRITE (1, 6050) NMASPS, DAY, ITITLE
      WRITE (1, 6060) PRTALL, RUNNAM, CD, STDOFF, VEL, DELAY, FZIMXC
      WRITE (1, 6070) SIGX, SIGY, XDC, YDC

C ----     SEED FOR RANDOM NUMBER GENERATOR
      ISEED = 0

C ----     CYLINDER RAY TRACE
      WRITE (*, 6080)
      DO 30 CYL = 1, NMCYLS
        WRITE (*, 6090) CYL, CRAD(CYL) * MMCVT, CYL, CSTPBK(CYL) * MMCVT
   30 CONTINUE
      IF (CSTPBK(0) .GT. 0) THEN
        WRITE (*, 6100) CSTPBK(0)
      ENDIF
      WRITE (*, 6110) NPLNTS
      IF (PRTALL) THEN
        WRITE (*, 6120) NMCYLS, NPLNTS
        DO 40 CYL = 1, NMCYLS
          WRITE (*, 6130) CSTPBK(CYL)
   40   CONTINUE
        WRITE (*, 6130) CSTPBK(0)
      ENDIF
      WRITE (1, 6120) NMCYLS, NPLNTS
      DO 50 CYL = 1, NMCYLS
        WRITE (1, 6130) CSTPBK(CYL)
   50 CONTINUE
      WRITE (1, 6130) CSTPBK(0)

C ----     PROCESS EACH REQUESTED VIEW
      DO 150 ASPECT = 1, NMASPS
        CALL CLOCKS(VTIME)
        INPCEL = 0
        ISEED1 = ISEED
        READ (*, 5050, ERR=160, END=170) AZIM, ELEV, CELSIZ, IDLKP,
     1                                  ILIMIT, ISPOT
C ----     IF CELSIZ IS NOT GIVEN, OR IS ZERO, SET IT TO 4 INCHES
C          (CONVERTED TO THE UNITS OF THE TARGET DESCRIPTION).
C          OTHERWISE, SIMPLY CONVERT IT TO THE UNITS OF THE TARGET
C ----     DESCRIPTION.
        IF (CELSIZ .LE. 0.) THEN
          CELSIZ = 4. * TGTCVT
        ELSE
          CELSIZ = CELSIZ / MMCVT
        ENDIF
```

```
C ----    IF RAYS MUST BE MADE, THEN THEIR LOCATION WILL BE HARDWIRED TO
C ----    RANDOM POINT IN GRID CELL.
        IF (MKRAYS) THEN
          RAYLOC = 0
        ELSE
          CALL SEEKVEW(51, 'RAYFILE', AZIM)
          READ (51, 5060, ERR=180, END=190) RFA, RFE, RFCS
          IF ((ELEV .NE. RFE) .OR. (CELSIZ .NE. RFCS)) THEN
            WRITE (*, 5140) AZIM, ELEV, CELSIZ, RFA, RFE, RFCS
            STOP
          ENDIF
        ENDIF
        IF (IDLKP .NE. 0) THEN
          CALL DELETE
        ENDIF
        IF (ISPOT .GT. 0) THEN
          RAYLOC = -1
        ENDIF
C ----    RETRIEVE TARGET MIN AND MAX
        L = LENRPP
        DO 60 I = 1, 3
          TMIN(I) = ASTER(L)
          TMAX(I) = ASTER(L + 1)
          TLEN(I) = TMAX(I) - TMIN(I)
          TCEN(I) = .5 * (TMAX(I) + TMIN(I))
          L = L + 2
   60   CONTINUE
C ----    DIRECTION COSINES
        AZIMR = AZIM * DEG2RAD
        ELEVR = ELEV * DEG2RAD
        SAZIM = SIN(AZIMR)
        CAZIM = COS(AZIMR)
        SELEV = SIN(ELEVR)
        CELEV = COS(ELEVR)
        W3(1) = -CELEV * CAZIM
        W3(2) = -CELEV * SAZIM
        W3(3) = -SELEV
C ----    COMPUTE DIMENSIONS OF GRID PLANE
        ABSSA = ABS(SAZIM)
        ABSCA = ABS(CAZIM)
        ABSSE = ABS(SELEV)
        ABSCE = ABS(CELEV)
        ENGTH = TLEN(1) * ABSCA + TLEN(2) * ABSSA
        HORZ = ABS(TLEN(1) * ABSSA + TLEN(2) * ABSCA)
        VERT = ABS(ENGTH * ABSSE + TLEN(3) * ABSCE)
        DEPTH = ABS(ENGTH * ABSCE + TLEN(3) * ABSSE)
        BACK = AINT(DEPTH * .55 - DOT(W5, TCEN) + .5)
C ----    FIND COORDINATES OF CENTER OF GRID PLANE
        T = TCEN(1) * CAZIM + TCEN(2) * SAZIM
        HC = -TCEN(1) * SAZIM + TCEN(2) * CAZIM
        VC = -T * SELEV + TCEN(3) * CELEV
```

```
      IF (ILIMIT .EQ. 0) THEN
C ----    FIND RANGE OF GRID PLANE AND ROUND TO WHOLE CELSIZ
        HMAX = HC + .5 * HORZ
        HMIN = HC - .5 * HORZ
        VMAX = VC + .5 * VERT
        VMIN = VC - .5 * VERT
        HMAX = SIGN(AINT(ABS(HMAX) / CELSIZ + .5001) * CELSIZ, HMAX)
        HMIN = SIGN(AINT(ABS(HMIN) / CELSIZ + .5001) * CELSIZ, HMIN)
        VMAX = SIGN(AINT(ABS(VMAX) / CELSIZ + .5001) * CELSIZ, VMAX)
        VMIN = SIGN(AINT(ABS(VMIN) / CELSIZ + .5001) * CELSIZ, VMIN)
      ELSE
        WRITE (*, 6150)
        READ (*, 5070, ERR=150, END=170) HMIN, HMAX, VMIN, VMAX
      ENDIF
      NHORZ = (HMAX - HMIN) / CELSIZ + 1.0001
      NVERT = (VMAX - VMIN) / CELSIZ + 1.0001
      NCELL = NHORZ * NVERT

C ----    PRINT PRIMARY BLOCK OF HARDCOPY OUTPUT
      WRITE (*, 6160) AZIM, ELEV,
     1           TMIN(1) * MMCVT, TMIN(2) * MMCVT, TMIN(3) * MMCVT,
     2           TMAX(1) * MMCVT, TMAX(2) * MMCVT, TMAX(3) * MMCVT,
     3           TCEN(1) * MMCVT, TCEN(2) * MMCVT, TCEN(3) * MMCVT,
     4           TLEN(1) * MMCVT, TLEN(2) * MMCVT, TLEN(3) * MMCVT,
     5           BACK * MMCVT, CELSIZ * MMCVT, HORZ * MMCVT,
     6           VERT * MMCVT, HC * MMCVT, VC * MMCVT, HMIN * MMCVT,
     7           HMAX * MMCVT, VMIN * MMCVT, VMAX * MMCVT, NHORZ,
     8           NVERT, NCELL
      WRITE (*, 6170) ISEED1
      IF (RAYLOC .EQ. 0) THEN
        WRITE (*, 6180)
      ELSE IF (RAYLOC .EQ. 1) THEN
        WRITE (*, 6190) 'RAYFILE'
      ELSE IF (RAYLOC .GT. 1) THEN
        WRITE (*, 6200)
      ENDIF
      IF (PRTALL) THEN
        WRITE (*, 6210) AZIM, ELEV, CELSIZ * MMCVT
      ENDIF
      WRITE (1, 6210) AZIM, ELEV, CELSIZ * MMCVT
C ----    EITHER READ IN RAY PARAMETERS OR CALCULATE THEM
   70 IF (.NOT. MKRAYS) THEN
        READ (51, 5080, ERR=180, END=190) HCEN, VCEN, H, V, IHIV, EOV
        GOTO 100
      ENDIF
      IF (ISPOT .GT. 0) THEN
        J = MOD(INPCEL, 4) + 1
        IF (J .EQ. 1) THEN
          READ (*, 5090, ERR=160, END=170) (CH(I), CV(I), I = 1, 4)
        ENDIF
        H = CH(J)
        V = CV(J)
        HCEN = SIGN(AINT(ABS(H / CELSIZ) + .5) * CELSIZ, H)
        VCEN = SIGN(AINT(ABS(V / CELSIZ) + .5) * CELSIZ, V)
        WRITE (*, 6220) H * MMCVT, V * MMCVT
        IHIV = 0
      ENDIF
```

```
      VCEN = VMAX
C ----  PSEUDO-LOOP FOR VERTICAL SCAN - SCANS TOP TO BOTTOM
      IVERT = 1
   80 HCEN = HMAX
C ----  PSEUDO-LOOP FOR HORIZONTAL SCAN - SCANS RIGHT TO LEFT
      IHORZ = 1
   90 IF (RAYLOC .EQ. 0) THEN
C ----   CHOOSE RANDOM POINT IN CELL
        IRANV = 10. * RAN(-1)
        IRANH = 10. * RAN(-1)
        IHIV = 10 * IRANH + IRANV
        V = VCEN + CELSIZ * (.1 * FLOAT(IRANV) - .45)
        H = HCEN + CELSIZ * (.1 * FLOAT(IRANH) - .45)
      ELSE
C ----   CHOOSE CENTER OF CELL
        H = HCEN
        V = VCEN
        IHIV = 0
      ENDIF
C ----   JIGGLE RAY
  100 CALL TROPIC(WP)
      CALL CROSS(WP, WP, WB)
C ----   ROTATE H,V TO COORDINATE SYSTEM OF TARGET
      XB(1) = -V * CAZIM * SELEV - H * SAZIM + SJIGLE * WP(1) -
     1        BACK * WB(1)
      XB(2) = -V * SAZIM * SELEV + H * CAZIM + SJIGLE * WP(2) -
     1        BACK * WB(2)
      XB(3) = V * CELEV + SJIGLE * WP(3) - BACK * WB(3)
C ----           TRACK CENTER RAY
C ----   DISTANCE ALONG SHOTLINE TO FIRST CONTACT
C        (RAYLEN .GT. 0) ==> FIRST CONTACT IS BETWEEN START POINT
C                            AND GRID PLANE
C ----   (RAYLEN .LT. 0) ==> FIRST CONTACT IS BEYOND GRID PLANE
      CALL FIRST
      RAYLEN(0, 0) = 0.
      ITMHIT(0, 0) = -1.
      IF (IREGON .GT. 0) THEN
        RAYLEN(0, 0) = BACK - DFIRST
        CALL UN2(LIRFO + IREGON - 1, ITMHIT(0, 0), IDUM)
      ENDIF
C ----   TRACK PLANETARY RAYS
C        NMHITS IS THE NUMBER OF PLANETARY RAYS THAT HAVE INTERSECTED
C ----   THE TARGET
      NMHITS = 0
      ANGINC = 6.283185 / FLOAT(NPLNTS)
      DO 120 PLANET = 1, NPLNTS
        SANGLE = (PLANET - 1) * ANGINC
        DO 110 CYL = 1, NMCYLS
          PLNETH(CYL, PLANET) = H + CRAD(CYL) * SIN(SANGLE)
          PLNETV(CYL, PLANET) = V + CRAD(CYL) * COS(SANGLE)
          XB(1) = -PLNETV(CYL, PLANET) * CAZIM * SELEV -
     1            PLNETH(CYL, PLANET) * SAZIM + SJIGLE * WP(1) -
     2            BACK * WB(1)
          XB(2) = -PLNETV(CYL, PLANET) * SAZIM * SELEV +
     1            PLNETH(CYL, PLANET) * CAZIM + SJIGLE * WP(2) -
     2            BACK * WB(2)
          XB(3) = PLNETV(CYL, PLANET) * CELEV + SJIGLE * WP(3) -
     1            BACK * WB(3)
```

```
            CALL FIRST
            RAYLEN(CYL, PLANET) = 0.
            ITMHIT(CYL, PLANET) = -1.
            IF (IREGON .GT. 0) THEN
              NMHITS = NMHITS + 1
              RAYLEN(CYL, PLANET) = BACK - DFIRST
              CALL UN2(LIRFO + IREGON - 1, ITMHIT(CYL, PLANET), IDUM)
            ENDIF
  110     CONTINUE
  120   CONTINUE
        IF (NMHITS .NE. 0 .OR. ITMHIT(0, 0) .GE. 0) THEN
          IF (PRTALL) THEN
            WRITE (*, 6230) HCEN * MMCVT, VCEN * MMCVT, IHIV, H * MMCVT,
     1                      V * MMCVT, RAYLEN(0, 0) * MMCVT,
     2                      ITMHIT(0, 0)
          ENDIF
          WRITE (1, 6230) HCEN * MMCVT, VCEN * MMCVT, IHIV, H * MMCVT,
     1                    V * MMCVT, RAYLEN(0, 0) * MMCVT, ITMHIT(0, 0)
C ----    OUTPUT MAIN-RAY DATA
          DO 140 CYL = 1, NMCYLS
            DO 130 PLANET = 1, NPLNTS
              IF (PRTALL) THEN
                WRITE (*, 6240) PLNETH(CYL, PLANET) * MMCVT,
     1                          PLNETV(CYL, PLANET) * MMCVT,
     2                          RAYLEN(CYL, PLANET) * MMCVT,
     3                          ITMHIT(CYL, PLANET)
              ENDIF
              WRITE (1, 6240) PLNETH(CYL, PLANET) * MMCVT,
     1                        PLNETV(CYL, PLANET) * MMCVT,
     2                        RAYLEN(CYL, PLANET) * MMCVT,
     3                        ITMHIT(CYL, PLANET)
  130       CONTINUE
  140     CONTINUE
        ENDIF
C ----    END OF CELL

        INPCEL = INPCEL + 1
        IF (ISPOT .GT. INPCEL) THEN
          GOTO 70
        ENDIF
        IF (MKRAYS) THEN
          IF (ISPOT .LE. 0) THEN
C ----    NEXT HORIZONTAL
            HCEN = HCEN - CELSIZ
            IHORZ = IHORZ + 1
            IF (IHORZ .LE. NHORZ) THEN
              GOTO 90
            ENDIF
C ----    NEXT VERTICAL AT END OF HORIZONTAL SCAN
            VCEN = VCEN - CELSIZ
            IVERT = IVERT + 1
            IF (IVERT .LE. NVERT) THEN
              GOTO 80
            ENDIF
          ENDIF
```

```
          ELSE
            READ (51, 5090, ERR=180, END=190) HCEN, VCEN, H, V, IHIV, EDV
            IF (EDV .NE. 999.9) THEN
              GOTO 100
            ENDIF
          ENDIF
C ----      END OF VIEW
          IF (PRTALL .AND. NMASPS .GT. 1) THEN
            WRITE (*, 6250)
          ENDIF
          WRITE (*, 6260) ASPECT
          WRITE (*, 6270) ISEED1, ISEED
          ENDFILE 1
          IF (IDLKP .NE. 0) THEN
            CALL RECLM
          ENDIF
C ----      TIME FOR THIS VIEW
          CALL CLOCKS(ETIME)
          WRITE (*, 6280) ASPECT, ETIME - VTIME
  150 CONTINUE
C ----      END OF ALL VIEWS
      END FILE 1
      REWIND 1
C ----      TOTAL TIME FOR CYLINDER
      CALL CLOCKS(ETIME)
      WRITE (*, 6290) ETIME - STIME
      RETURN

C ----      HANDLE READ ERRORS
  160 WRITE (*, 6300) 'INPUT'
      STOP
  170 WRITE (*, 6310) 'INPUT'
      STOP
  180 WRITE (*, 6300) 'RAYFILE'
      STOP
  190 WRITE (*, 6310) 'RAYFILE'
      STOP

 5010 FORMAT (3I5)
 5020 FORMAT (A20, 2I5, 2F7.2, 2F9.3, I5)
 5030 FORMAT (3F10.2)
 5040 FORMAT (4F5.3)
 5050 FORMAT (3F10.0, 3I5)
 5060 FORMAT (3F5.0)
 5070 FORMAT (4F10.0)
 5080 FORMAT (/ 10X, 4F10.4, I4, 8X, F6.1)
 5090 FORMAT (9F10.0)
 6010 FORMAT (' NMASPS MUST EQUAL 1...'/
     1         ' SUBROUTINE ''SHOTCYL'' ABORTED.')
 6020 FORMAT ('OCPTION SET TO SUPPRESS PRINTER OUTPUT')
 6030 FORMAT (' FZ1MXC (', I3, ') > TOTAL NO. OF CYLS (', I3, ')'/
     1         ' SUBROUTINE ''SHOTCYL'' ABORTED.')
 6040 FORMAT (' ONE OR MORE CYLINDER(S) MUST BE SPECIFIED'/
     1         ' IN NON-DECREASING ORDER...'/
     2         ' SUBROUTINE ''SHOTCYL'' ABORTED.')
 6050 FORMAT (' ', I5, A10, A60)
 6060 FORMAT (' ', L2, A20, 2F7.2, 2F9.3, I5)
 6070 FORMAT (' ', 4F8.3)
```

```
6080 FORMAT ("0---- GEOMETRY OF PROJECTILE SKIN (MM) ----")
6090 FORMAT (" RADIUS OF CYLINDER    ", I2, "     ", F10.2/
    1        " STEPBACK OF CYLINDER ", I2, "     ", F10.2)
6100 FORMAT (" LENGTH OF NEEDLE-NOSE PROBE       ", F10.2)
6110 FORMAT ("0PLANETARY RAYS PER CYLINDER       ", I10)
6120 FORMAT (" ", 2I10)
6130 FORMAT (" ", F10.2)
6140 FORMAT ("0TARGET DESCRIPTION SAYS A=", F10.3/
    1        "                            E=", F10.3/
    2        "                       CELSIZ=", F10.3/
    3        " BUT RAYFILE SAYS           A=", F10.3/
    4        "                            E=", F10.3/
    5        "                       CELSIZ=", F10.3/
    6        " SUBROUTINE ""SHOTCYL"" ABORTED.")
6150 FORMAT ("0OPTION SET TO LIMIT GRID PLANE")
6160 FORMAT ("0AZIMUTH             ", F10.3, " DEGREES"/
    1        " ELEVATION           ", F10.3, " DEGREES"/
    2        "0---- TARGET --------          X", 9X, "Y", 9X, "Z"/
    3        " TARGET MINIMUM (MM)      ", 3F10.3/
    4        " TARGET MAXIMUM (MM)      ", 3F10.3/
    5        " TARGET CENTER (MM)       ", 3F10.3/
    6        " TARGET DIMENSIONS (MM)   ", 3F10.3/
    7        "0---- GRID PLANE ----"/
    8        " BACK OFF DISTANCE        ", F10.3, " MM"/
    Z        " CELL SIZE                ", F10.3, " MM"/
    1        " HORIZONTAL LENGTH        ", F10.3, " MM"/
    2        " VERTICAL LENGTH          ", F10.3, " MM"/
    3        " CENTER (MM)              ", 2F10.3/
    4        " HORIZONTAL RANGE (MM)    ", 2F10.3/
    5        " VERTICAL RANGE (MM)      ", 2F10.3/
    6        "0NUMBER HORZ CELLS        ",  I10/
    7        " NUMBER VERT CELLS        ",  I10/
    8        " NUMBER OF CELLS          ",  I10)
6170 FORMAT ("0FIRST SEED FOR RANDOM NUMBER GENERATOR", I12)
6180 FORMAT("0OPTION SET TO COMPUTE RANDOM POINT IN CELL")
6190 FORMAT("0OPTION SET TO READ POINTS FROM FILE """, A, """")
6200 FORMAT("0OPTION SET TO CHOOSE CENTER OF CELL")
6210 FORMAT ("0", 3F10.2)
6220 FORMAT ("0SPECIFIED  HORZ=", F10.2, " VERT=", F10.2)
6230 FORMAT ("0", 2F8.2, I3, 3F8.2, I6)
6240 FORMAT (" ", 3F8.2, I6)
6250 FORMAT (" 999.9",T1X,"END")
6260 FORMAT ("1END OF CASE", I5)
6270 FORMAT ("0FIRST SEED FOR RANDOM NUMBER GENERATOR", I12/
    1        " NEXT  SEED FOR RANDOM NUMBER GENERATOR", I12)
6280 FORMAT ("0TIME FOR CASE", I5, F9.3, " SECONDS"//)
6290 FORMAT ("0TOTAL TIME FOR CYLINDER", F9.3, " SECONDS")
6300 FORMAT ("0READ ERROR ON FILE ", A/
    1        " SUBROUTINE ""SHOTCYL"" ABORTED.")
6310 FORMAT ("0EOF ENCOUNTERED ON FILE ", A/
    1        " SUBROUTINE ""SHOTCYL"" ABORTED.")
     END
```

```
      SUBROUTINE SEEKVEW (WHENCE, FNAME, AZIMUTH)

      INTEGER          WHENCE
      CHARACTER *7     FNAME
      REAL             AZIMUTH, EOV, VAZIM
      LOGICAL          EXISTF

      INQUIRE (FILE=FNAME, EXIST=EXISTF)
      IF (.NOT. EXISTF) THEN
        WRITE (*, 610) FNAME
        STOP
      ENDIF
      VAZIM = -1
   10 CLOSE (WHENCE)
      OPEN (WHENCE, FILE=FNAME)
      READ (WHENCE, 510, END=30) VAZIM
      IF (VAZIM .EQ. AZIMUTH) THEN
        BACKSPACE WHENCE
        RETURN
      ENDIF
   20 READ (WHENCE, 520, END=40) EOV
      IF (EOV .EQ. 999.9) THEN
        READ (WHENCE, *, END=10)
        WRITE (*, 620) AZIMUTH, FNAME
        STOP
      ENDIF
      GOTO 20
   30 IF (VAZIM .GE. 0) THEN
        WRITE (*, 630) VAZIM
      ELSE
        WRITE (*, 640)
      ENDIF
      WRITE (*, 650) AZIMUTH, WHENCE
      STOP
   40 IF (VAZIM .GE. 0) THEN
        WRITE (*, 630) VAZIM
      ELSE
        WRITE (*, 640)
      ENDIF
      WRITE (*, 660) AZIMUTH, WHENCE
      STOP

  510 FORMAT (F5.0)
  520 FORMAT (/ 62X, F6.1)
  610 FORMAT ('0FILE ''', A, ''' NOT FOUND...'/
     1          ' SUBROUTINE ''SEEKVEW'' ABORTED')
  620 FORMAT ('0NECESSARY *EOF WAS NOT FOUND...'/
     1          ' DID NOT FIND ', F5.1, ' DEGREE VIEW ON FILE ', A/
     2          ' SUBROUTINE ''SEEKVEW'' ABORTED')
  630 FORMAT ('0LAST VIEW READ WAS ', F5.1, ' DEGREES')
  640 FORMAT ('0READ NO VIEWS')
  650 FORMAT (' DID NOT FIND ', F5.1, ' DEGREE VIEW ON UNIT', I5/
     1          ' SUBROUTINE ''SEEKVEW'' ABORTED')
  660 FORMAT (' NECESSARY 999.9 WAS NOT FOUND...'/
     1          ' DID NOT FIND ', F5.1, ' DEGREE VIEW ON FILE ', A/
     2          ' SUBROUTINE ''SEEKVEW'' ABORTED')
      END
```

```
      PROGRAM MISFIR
C
C     COMPUTES EFFECTIVE STANDOFF CELL-BY-CELL FOR A PARTICULAR
C     THREAT/TARGET COMBINATION.  USES OUTPUT FROM SHOTCYL TO
C     DETERMINE INTERSECTS FOR CYLINDER(S) REPRESENTING THREAT.
C
C     WRITTEN BY PAUL TANENBAUM, IN WHOM ALL BLAME LIES
C                ATTN: SLCBR-VL-S
C                APG, MD  21005-5066
C
      INTEGER CELL, CYL, CYL1, FNCHAR, FZ1MXC, FZ2HIT, IMAIN, IPLNET,
     1        NMASPS, NMCELLS, NMCYLS, NMHITS, NPLNTS, PLANET, READNO
      REAL AZIM, CD, CELLPH, CELSIZ, CSTPBK, DELAY, DIST, DLYDIS, DMAIN,
     1     DPLNET, ELEV, GAP, HCEN, PDIST, PGAP, SIGX, SIGY, STDOFF,
     2     VCEN, VEL, X, XDC, Y, YDC
      LOGICAL DEBUG, EXISTF, PRTALL
      CHARACTER *7 FNAME
      CHARACTER *10 DAY
      CHARACTER *20 RUNNAM
      CHARACTER *60 ITITLE
      CHARACTER *64 SHOTLN
      COMMON /DEBUG/ DEBUG
      COMMON /ASPECT/ AZIM, ELEV
      COMMON /CELSIZ/ CELSIZ
      COMMON /DISPCO/ XDC, YDC
      COMMON /DISPER/ SIGX, SIGY
      COMMON /NMS/ NMCELLS, NMCYLS, NPLNTS
      COMMON /ROUND/ CD, STDOFF
      DIMENSION CSTPBK(0:10)

C ----    BEGIN EXECUTION    ---------------------------------------

C ----    OPEN FILES

C ----    IF THE DATA FILE DOES NOT EXIST THEN ABORT
C          (ANY TRAILING BLANKS IN REPRESENTATION OF FILE NAME
C ----      MUST BE IGNORED)
      FNAME='TAPE1'
      DO 10 FNCHAR = 1, 6
        IF (FNAME(FNCHAR + 1:FNCHAR + 1) .EQ. ' ') THEN
          GOTO 20
        ENDIF
   10 CONTINUE
   20 INQUIRE (FILE=FNAME(:FNCHAR), EXIST=EXISTF)
      IF (.NOT. EXISTF) THEN
        WRITE (*, 6010) FNAME
        STOP
      ELSE
        OPEN (51, FILE=FNAME(:FNCHAR))
        REWIND 51
      ENDIF
      OPEN (61, FILE='XXXSTRT')
      OPEN (62, FILE='XXXWTD')
      OPEN (63, FILE='RESULT')
```

```
C ----    READ HEADER LINE FOR THIS RUN
      READNO = 1
      READ (51, 5010, ERR=110, END=120) NMASPS, DAY, ITITLE
      WRITE (63, 5020) NMASPS, DAY, ITITLE
      IF (PRTALL) THEN
        WRITE (*, 6020) NMASPS, DAY, ITITLE
      ENDIF

C ----    READ PROJECTILE DATA
      READNO = 2
      READ (51, 5020, ERR=90, END=100) PRTALL, RUNNAM, CD, STDOFF, VEL,
     1                                 DELAY, FZ1MXC
      DEBUG = .FALSE.
      IF ((VEL * DELAY * FZ1MXC .EQ. 0.0) .AND.
     1    (VEL + DELAY + FZ1MXC .NE. 0.0)) THEN
        WRITE (*, 6030)
        STOP
      ENDIF
      READNO = 3
      READ (51, 5030, ERR=90, END=100) SIGX, SIGY, XDC, YDC
      IF (DEBUG) THEN
        WRITE (*, 6040) RUNNAM
      ENDIF

C ----    READ PROJECTILE GEOMETRY INFORMATION
      READNO = 4
      READ (51, 5040, ERR=110, END=120) NMCYLS, NPLNTS
      READNO = 5
      DO 30 CYL = 1, NMCYLS
        READ (51, 5050, ERR=110, END=120) CSTPBK(CYL)
   30 CONTINUE
      READNO = 6
      READ (51, 5050, ERR=110, END=120) CSTPBK(0)

C ----    READ HEADER LINE FOR THIS VIEW
      READNO = 7
      READ (51, 5060, ERR=110, END=120) AZIM, ELEV, CELSIZ

      WRITE (63, 6050) AZIM, ELEV, CELSIZ, FZ1MXC
      IF (PRTALL) THEN
        WRITE (*, 6050) AZIM, ELEV, CELSIZ, FZ1MXC
      ENDIF

C ----    FIND THE AIMPOINT (ASSUMED TO BE CENTER OF PRESENTED AREA)
      CALL CPA (FNAME(:FNCHAR))

C ----    PRINT OUT RUN INFORMATION
      WRITE (*, 6060) ITITLE, DAY, AZIM, ELEV, CELSIZ, RUNNAM, CD,
     1                STDOFF, VEL, DELAY, SIGX, SIGY, XDC, YDC, NMCYLS,
     2                FZ1MXC, NPLNTS
      DO 40 CYL = 1, NMCYLS
        WRITE (*, 6070) CYL, CSTPBK(CYL)
   40 CONTINUE
      IF (CSTPBK(0) .GT. 0) THEN
        WRITE (*, 6080) CSTPBK(0)
      ENDIF
      WRITE (*, 5090)
```

```
C ----     DETERMINE HOW FAR PROJECTILE WILL MOVE DURING FUZE DELAY
C ----     (DISTANCE = RATE * TIME)
      DLYDIS = VEL * DELAY

C ----     READ IN MAIN SHOTLINE FOR NEXT CELL
      CELL = 0
   50 READNO = 8
        READ (51, 5070, ERR=110, END=90) SHOTLN
        READNO = 9
        READ (SHOTLN, 5090, ERR=130, END=140) HCEN, VCEN, X, Y, DMAIN,
     1        IMAIN
        IF (IMAIN .EQ. -1) THEN
          NMHITS = 0
        ELSE
          NMHITS = 1
        ENDIF
        GAP = 0.0
        CYL1 = 0
        FZ2HIT = 0

C ----     READ IN SHOTLINES FOR RAYS CONSTITUTING CYLINDER(S) AND
C          DETERMINE WHICH PLANET RAY, IF ANY, REPRESENTS THE
C ----     REGION OF THE PROJECTILE WHICH FIRST IMPACTS THE TARGET
        DO 70 CYL = 1, NMCYLS
          DO 60 PLANET = 1, NPLNTS
            READNO = 10
            READ (51, 5070, ERR=110, END=120) SHOTLN
            READNO = 11
            READ (SHOTLN, 5090, ERR=130, END=140) DPLNET, IPLNET
            IF (IPLNET .EQ. -1) THEN
              GOTO 60
            ENDIF
            NMHITS = NMHITS + 1
            PGAP = DPLNET - DMAIN - CSTPBK(CYL) - CSTPBK(0)
C ----     (PGAP .GT. 0) ==> THIS PLANET HITS THE TARGET "BEFORE" THE
C                            CENTER OF THE PROJECTILE DOES.
C          (FZ2HIT .EQ. 1) ==> THE PRIMARY FUZE HITS THE TARGET FIRST
C                              (I.E. 0 <= CYL1 <= FZ1MXC), BUT THE
C                              SECONDARY FUZE HITS, TOO, BEFORE THE FUZE
C ----                         DELAY IS UP.
            IF ((PGAP .GT. GAP) .OR. (NMHITS .EQ. 1)) THEN
              GAP = PGAP
              CYL1 = CYL
            ELSE IF ((CYL .GT. FZ1MXC) .AND. (CYL1 .LE. FZ1MXC)) THEN
              PDIST = DMAIN + GAP + CSTPBK(0) + CSTPBK(CYL) - DPLNET
              IF (PDIST .LT. DLYDIS) THEN
                FZ2HIT = 1
              ENDIF
            ENDIF
   60     CONTINUE
   70   CONTINUE
C ----     TURN OFF FZ2HIT FLAG IF SECONDARY FUZE HITS TARGET FIRST
        IF (CYL1 .GT. FZ1MXC) THEN
          FZ2HIT = 0
        ENDIF
```

```
C ----     DEFINE ACTUAL STANDOFF IN UNITS OF CHARGE DIAMETERS
          CELL = CELL + 1
          IF (OMAIN .EQ. 0) THEN
C ----       MARK THIS CELL, MEANING THE MAIN SHOTLINE MISSED THE TARGET
            DIST = 0
          ELSE
            DIST = (GAP + STDOFF) / CD
          ENDIF

C ----     DETERMINE THE PROBABILITY THAT THIS CELL WILL BE HIT
          CELLPH = PHIT(X, Y)

C ----     STORE THIS CELL'S LOCATION AND CONTENTS IN FILE 'XXXSTRT',
C           ITS LOCATION AND WEIGHTED CONTENTS IN FILE 'XXXWTD', AND
C ----      ITS CELL-LOCATION AND CONTENTS IN FILE 'RESULT'
          WRITE (61, 6100) HCEN, VCEN, DIST, 1.
          WRITE (62, 6100) HCEN, VCEN, DIST, CELLPH
          WRITE (63, 6110) HCEN, VCEN, DIST, CYL1, FZZHIT, CELLPH
          IF (PRTALL) THEN
C           WRITE (*, 6110) HCEN, VCEN, DIST, CYL1, FZZHIT, CELLPH
            WRITE (*, 6110) X, Y, DIST, CYL1, FZZHIT, CELLPH
          ENDIF
        GOTO 50

C ----     CLOSE FILES
   80 CLOSE (51)
      CLOSE (61)
      CLOSE (62)
      WRITE (63, 6120) 999.9
      CLOSE (63)

C ----     PRINT HISTOGRAMS OF FIRINGS VS. ACTUAL STANDOFF
      CALL HISTOG ('XXXSTRT', .FALSE., RUNNAM)
      CALL HISTOG ('XXXWTD', .TRUE., RUNNAM)

C ----     PRINT SILHOUETTE
      CALL SILOET ('XXXSTRT', .TRUE., RUNNAM)
      IF (DEBUG) THEN
        WRITE (*, 6130)
      ENDIF
      STOP

C ----     HANDLE READ ERRORS
   90 WRITE (*, 6140) READNO, 'ERROR IN FILE ', 'INPUT'
      STOP
  100 WRITE (*, 6140) READNO, 'EOF IN FILE ', 'INPUT'
      STOP
  110 WRITE (*, 6140) READNO, 'ERROR IN FILE ', FNAME
      STOP
  120 WRITE (*, 6140) READNO, 'EOF IN FILE ', FNAME
      STOP
  130 WRITE (*, 6140) READNO, 'ERROR IN ', 'SHOTLN'
      STOP
  140 WRITE (*, 6140) READNO, 'EOS IN ', 'SHOTLN'
      STOP
```

```
5010 FORMAT (1X, I5, A10, A60)
5020 FORMAT (1X, L2, A20, 2F7.2, 2F8.3, I5)
5030 FORMAT (1X, 4F8.3)
5040 FORMAT (1X, 2I10)
5050 FORMAT (1X, F10.2)
5060 FORMAT (1X, 3F10.2)
5070 FORMAT (A)
5080 FORMAT (1X, 2F8.2, 3X, 3F9.2, I6)
5090 FORMAT (17X, F8.2, I6)
6010 FORMAT ('1FILE ', A, ' NOT FOUND'/' PROGRAM ''MISFIR'' ABORTED')
6020 FORMAT (1X, I5, A10, A60)
6030 FORMAT ('1BAD PRIMARY FUZE DATA'/' PROGRAM ''MISFIR'' ABORTED')
6040 FORMAT ('1ENTER ''MISFIR''.  THE TARGET IS ', A)
6050 FORMAT (1X, 2F8.1, 24X, F8.1, I2)
6060 FORMAT ('1------ PROGRAM MISFIR ------', /
    1        '0TARGET IS ', A60, ' RUN ON ', A10, /
    2        ' AZIMUTH                    ', F6.2, ' DEGREES', /
    3        ' ELEVATION                  ', F6.2, ' DEGREES', /
    4        ' CELL SIZE                 ', F7.2, ' MM', /
    5        '0PROJECTILE IS ', A20, /
    6        ' CHARGE DIAMETER            ', F6.2, ' MM', /
    7        ' BUILT-IN STANDOFF          ', F6.2, ' MM', /
    8        ' IMPACT VELOCITY            ', F6.2, ' M/S', /
    9        ' FUZE DELAY TIME            ', F6.2, ' MS', /
    *        ' X DISPERSION            ', F8.3,
    1        ',      Y DISPERSION       ', F9.3, /
    2        ' X CORRECTION            ', F9.3,
    3        ',      Y CORRECTION       ', F8.3, /
    4        ' NUMBER OF CYLINDERS            ', I2, /
    5        ' LAST CYL IN PRIMARY FUZE       ', I2, /
    6        ' NUMBER OF RAYS PER CYL         ', I2, /)
6070 FORMAT (' STEPBACK OF CYLINDER ', I2, 2X, F6.2)
6080 FORMAT (' LENGTH OF 0-WIDTH PROBE  ', F6.2)
6090 FORMAT (' ---------------------------')
6100 FORMAT (3F8.2, F8.6)
6110 FORMAT (1X, F6.0, F7.1, F8.4, 2I5, F8.6)
6120 FORMAT (F6.1)
6130 FORMAT (' NORMAL EXIT OF ''MISFIR''')
6140 FORMAT ('1READ *', I2, 'ENCOUNTERED ', 2A/
    1        ' PROGAM ''MISFIR'' ABORTED')
     END
```

```
      SUBROUTINE CPA (FNAME)

      INTEGER NMCELLS, NMCYLS, NPLNTS, PLANET, READNO
      REAL DUMMY, X, XAIM, XBAR, XDC, Y, YAIM, YBAR, YDC
      LOGICAL DEBUG
      CHARACTER *7 FNAME
      COMMON /AIM/ XAIM, YAIM
      COMMON /DEBUG/ DEBUG
      COMMON /DISPCO/ XDC, YDC
      COMMON /NMS/ NMCELLS, NMCYLS, NPLNTS

C ----     BEGIN EXECUTION -------------------------------------------------
      IF (DEBUG) THEN
        WRITE (*, 610)
      ENDIF
      XBAR = 0
      YBAR = 0
      NMCELLS = 0

C ----     FOR EVERY BUNDLE, READ COORDINATES OF CENTER
   10 READNO = 1
      READ (51, 510, END=30) X, Y

C ----     ENSURE THAT REST OF DATA FOR CURRENT BUNDLE IS OK
      DO 20 PLANET = 1, NPLNTS * NMCYLS
        READNO = 2
        READ (51, 520, ERR=50, END=60) DUMMY
   20 CONTINUE

C ----     COMPUTE XBAR AND YBAR
        XBAR = XBAR + X
        YBAR = YBAR + Y
        NMCELLS = NMCELLS + 1
      GOTO 10
   30 IF (NMCELLS .EQ. 0) THEN
        WRITE (*, 620) 'ERROR:  NO BUNDLE DATA LOADED.'
        WRITE (*, 620) 'SUBROUTINE ''CPA'' ABORTED.'
        STOP
      ENDIF
      XBAR = XBAR / NMCELLS
      YBAR = YBAR / NMCELLS

C ----     CALCULATE THE COORDINATES OF THE AIMPOINT...
C ----     AIMPOINT IS CENTER OF PRESENTED AREA CORRECTED FOR DISPERSAL
      XAIM = XBAR + XDC
      YAIM = YBAR + YDC
      IF (DEBUG) THEN
        WRITE (*, 630) XBAR, YBAR, XAIM, YAIM
      ENDIF

C ----     REPOSITION FILE TO ORIGINAL POSITION
      REWIND 51
      DO 40 I = 1, NMCYLS + 6
        READ (51, *, ERR=50, END=60)
   40 CONTINUE

      RETURN
```

```
C ----     HANDLE FILE ERRORS
   50 WRITE (*, 620) '1READ #', READNO,
     1                ' ENCOUNTERED AN ERROR IN FILE ', FNAME,
     2                ' WHILE ATTEMPTING TO READ CELL NUMBER', NMCELLS
      GOTO 70
   60 WRITE (*, 620) '1READ #', READNO,
     1                ' ENCOUNTERED EOF IN FILE ', FNAME,
     2                ' WHILE ATTEMPTING TO READ CELL NUMBER', NMCELLS
   70 WRITE (*, 620) ' SUBROUTINE ''CPA'' ABORTED.'
      STOP

  510 FORMAT (1X, 19X, 2F8.2)
  520 FORMAT (1X, 16X, F9.2)
  610 FORMAT (' ENTER SUBROUTINE ''CPA''.')
  620 FORMAT (A, :I2, 2A, :A, I5)
  630 FORMAT (' XBAR = ', F10.4, '    YBAR = ', F10.4,
     1        ' XAIM = ', F10.4, '    YAIM = ', F10.4)
      END
```

```
      FUNCTION PHIT (X, Y)

      REAL CELSIZ, PHIT, PDFXY, QDFZ1, QDFZ2, QDFZ3, QDFZ4, SIGX, SIGY,
     1     X, XAIM, XGRID, Y, YAIM, YGRID
      LOGICAL DEBUG
      COMMON /AIM/ XAIM, YAIM
      COMMON /DEBUG/ DEBUG
      COMMON /DISPER/ SIGX, SIGY
      COMMON /CELSIZ/ CELSIZ
C------------------------------------------------------------------------
      IF (SIGX * SIGY .EQ. 0) THEN
        WRITE (*, *) 'ZERO DISPERSION NOT ALLOWED.'
        WRITE (*, *) 'FUNCTION ''PHIT'' ABORTED.'
        STOP
      ENDIF

C ----   CALCULATE THE PROBABLITY OF HITTING THE LOCATION (X, Y)

      XGRID = CELSIZ / 2
      YGRID = CELSIZ / 2
      QDFZ1 = DFN((X - XAIM + XGRID) / SIGX)
      QDFZ2 = DFN((X - XAIM - XGRID) / SIGX)
      QDFZ3 = DFN((Y - YAIM + YGRID) / SIGY)
      QDFZ4 = DFN((Y - YAIM - YGRID) / SIGY)
      PDFXY = (QDFZ1 - QDFZ2) * (QDFZ3 - QDFZ4)
      PHIT = PDFXY
      IF (DEBUG) THEN
        WRITE (*, 610) X, XAIM, XGRID, SIGX
        WRITE (*, 620) Y, YAIM, YGRID, SIGY
        WRITE (*, 630) QDFZ1, QDFZ2
        WRITE (*, 640) QDFZ3, QDFZ4
        WRITE (*, 650) PDFXY
      ENDIF
      RETURN

  610 FORMAT (' X=', F10.4, ' XAIM=', F10.4, ' XGRID=', F10.4,
     1        ' SIGX=', F10.4)
  620 FORMAT (' Y=', F10.4, ' YAIM=', F10.4, ' YGRID=', F10.4,
     1        ' SIGY=', F10.4)
  630 FORMAT (' QDFZ1=', F10.4, '  QDFZ2=', F10.4)
  640 FORMAT (' QDFZ3=', F10.4, '  QDFZ4=', F10.4)
  650 FORMAT (' PDFXY=', F10.4)
      END
```

```
      REAL FUNCTION DFN (X)

C     FROM HASTINGS APPROXIMATIONS FOR DIGITAL COMPUTERS
C     (BORROWED FROM WILSON'S FILE 'CETANK'
C      OBTAINED:  9 Jun 83)

      REAL ABSOFX, F, X
      LOGICAL DEBUG
      COMMON /DEBUG/ DEBUG
C     ----------------------------------------------------------------

      F = 0
      ABSOFX = ABS(X)
      IF (ABSOFX .LT. 5) THEN
        F = ((((((.5383E-5 * ABSOFX + .4889065E-4) * ABSOFX + .380036E-4)
     1        * ABSOFX + .0032776263) * ABSOFX + .0211410061) * ABSOFX
     2        + .0498673469) * ABSOFX + 1
        F = .5 / (F**16)
      ENDIF
      IF (X .GE. 0) THEN
        F = 1 - F
      ENDIF
      DFN = F
      IF (DEBUG) THEN
        WRITE (*, 610) X, F
      ENDIF
      RETURN

  610 FORMAT (' DFN(', F10.4, ') = ', F10.4)
      END
```

```
      SUBROUTINE HISTOG (FNAME, WTD, TARGNA)

      INTEGER BIN, CELL, COLO, COLUMN, ILABEL, LINE, NPLNTS, NMBINS,
     1        NMCELLS, NMCYLS
      REAL BINSIZ, CD, DFLTBS, FREQ, HMAX, HSCALF, MAXVAL, RLABEL,
     1     STDOFF, VALUE, VMAX, VSCALF, WT
      LOGICAL DEBUG, WTD
      CHARACTER *7 FNAME
      CHARACTER *20 TARGNA
      CHARACTER *120 IMAGE
      COMMON /DEBUG/ DEBUG
      COMMON /NMS/ NMCELLS, NMCYLS, NPLNTS
      COMMON /ROUND/ CD, STDOFF
      DIMENSION FREQ(-1:110), IMAGE(52), VALUE(10000), WT(10000)
      PARAMETER (COLO=11, DFLTBS=0.1)

C     ------------------------------------------------------------------
      IF (DEBUG) THEN
        WRITE (*, 6010)
      ENDIF
C ----    INITIALIZE DFLTBS, FREQ(), IMAGE(), MAXVAL, NMBINS, AND VMAX
      DO 10 BIN = -1, 100
        FREQ(BIN) = 0
   10 CONTINUE
      DO 20 LINE = 1, 52
        IMAGE (LINE) = ' '
   20 CONTINUE
      MAXVAL = 0
      NMBINS = 0
      VMAX = 0

C ----    READ VALUE() AND WT()...  DETERMINE THE LARGEST ELEMENT
C ----    IN VALUE()
      OPEN (5, FILE=FNAME)
      REWIND 5
      READ (5, 5010, ERR=150, END=150)
     1               (VALUE(CELL), WT(CELL), CELL=I, NMCELLS)
      CLOSE (5)
      DO 30 CELL = 1, NMCELLS
        MAXVAL = MAX(MAXVAL, VALUE(CELL))
   30 CONTINUE

C ----    FOR EACH CELL, ADD WT(CELL) TO THE BIN ASSOCIATED WITH
C ----    VALUE(CELL)
      IF (MAXVAL .LE. 100 * DFLTBS) THEN
C ----    THERE ARE NO MORE THAN 100 BINS; EACH IS OF SIZE DFLTBS
        BINSIZ = DFLTBS
      ELSE
C ----    THERE ARE 100 BINS; EACH IS OF SIZE MAXVAL / 100
        BINSIZ = MAXVAL / 100
      ENDIF

      DO 40 CELL = 1, NMCELLS
        IF (VALUE(CELL) .EQ. 0) THEN
          BIN = -1
        ELSE
          BIN = VALUE(CELL) / BINSIZ
        ENDIF
```

```
        FREQ(BIN) = FREQ(BIN) + WT(CELL)
        NMBINS = MAX (NMBINS, BIN)
        IF (DEBUG) THEN
          WRITE (*, 6020) CELL, VALUE(CELL), WT(CELL), BIN, NMBINS
        ENDIF
   40 CONTINUE
      HMAX = NMBINS * BINSIZ

C ----    SCALE THE DISTRIBUTION CURVE TO FIT ON THE PAGE
      DO 50 BIN = -1, NMBINS
        VMAX = MAX(VMAX, FREQ(BIN))
   50 CONTINUE
      IF (HMAX * VMAX .EQ. 0) THEN
        WRITE (*, 6030)
        STOP
      ENDIF
      HSCALF = 100 / HMAX
      VSCALF = 50 / VMAX

C ----    PLOT HISTOGRAM INTO IMAGE()
      DO 70 BIN = 0, NMBINS
        COLUMN = COLO + 101 - ((BIN + .5) * BINSIZ) * HSCALF
        DO 60 LINE = 50, 51.5 - VSCALF * FREQ(BIN), - 1
          IMAGE(LINE)(COLUMN:COLUMN) = '*'
   60   CONTINUE
   70 CONTINUE
      COLUMN = COLO + 105
      DO 80 LINE = 50, 51.5 - VSCALF * FREQ(-1), -1
        IMAGE(LINE)(COLUMN:COLUMN) = '*'
   80 CONTINUE

C ----    ADD AXES TO IMAGE()
      DO 90 LINE = 1, 51
        IF (MOD(LINE, 10) .EQ. 1) THEN
          IMAGE(LINE)(COLO - 1:COLO - 1) = '-'
        ELSE
          IMAGE(LINE)(COLO - 1:COLO - 1) = ':'
        ENDIF
   90 CONTINUE
      DO 100 COLUMN = COLO - 2, COLO + 100
        IF (MOD(COLUMN, 10) .EQ. 1) THEN
          IMAGE (51)(COLUMN:COLUMN) = 'I'
        ELSE
          IMAGE (51)(COLUMN:COLUMN) = '.'
        ENDIF
  100 CONTINUE
      COLUMN = COLO + 105
      IMAGE(51)(COLUMN:COLUMN) = 'E'

C ----    LABEL AXES IN IMAGE()
      IF (MAXVAL .GE. 50) THEN
        DO 110 LINE = 1, 41, 10
          ILABEL = (51 - LINE) / VSCALF
          WRITE (IMAGE(LINE)(COLO - 7:COLO - 3), 6040) ILABEL
  110   CONTINUE
```

```
      ELSE
        DO 120 LINE = 1, 41, 10
          RLABEL = (51 - LINE) / VSCALF
          WRITE (IMAGE(LINE)(COLO - 10:COLO - 3), 6050) RLABEL
  120   CONTINUE
      ENDIF
      IMAGE(51)(COLO-2:COLO-1) = '0:'
      IF (BINSIZ .EQ. DFLTBS) THEN
        WRITE (IMAGE(52)(COLO-2:), 6060)
      ELSE
        WRITE (IMAGE(52)(COLO-4:), 6070) (HMAX * COLUMN / 100.,
     1          COLUMN = 100, 20, -20), 0
      ENDIF

C ----    WHEN DEBUGGING, PRINT CONTENTS OF EACH BIN
      IF (DEBUG) THEN
        WRITE (*, 6080)
        DO 130 BIN = -1, NMBINS
          WRITE (*, 6090) BIN, FREQ(BIN)
  130   CONTINUE
      ENDIF

C ----    PRINT IMAGE()
      WRITE (*, 6100)
      IF (.NOT. WTD) THEN
        WRITE (*, 6110)
      ELSE
        WRITE (*, 6120)
      ENDIF
      WRITE (*, 6130)
      WRITE (*, 6140) TARGNA
      WRITE (*, 6150) HMAX / 100., HMAX * CD / 100.
      WRITE (*, 6160) VMAX / 50.
      DO 140 LINE = 1, 52
        WRITE (*, 6170) IMAGE(LINE)
  140 CONTINUE
      WRITE (*, 6080)
      RETURN

C ----    HANDLE READ ERRORS
  150 WRITE (*, 6180) 'READ SCREW UP.  SUBROUTINE ''HISTOG'' ABORTED.'
      STOP

 5010 FORMAT (16X, F8.2, F8.6)
```

```
6010 FORMAT (' ENTER SUBROUTINE ''HISTOG''.')
6020 FORMAT (' VALUE(', I4, ') = ', F10.3, ' WT = ', F10.3,
    1           ' BIN IS ', I3, ' OUT OF ', I3)
6030 FORMAT ('1ZERO DENOMINATOR '/'PROCEDURE ''HISTOG'' ABORTED')
6040 FORMAT (I5)
6050 FORMAT (1PE8.2)
6060 FORMAT ('10', 9X, '9', 9X, '8', 9X, '7', 9X, '6', 9X, '5', 9X,
    1           '4', 9X, '3', 9X, '2', 9X, '1', 9X, '0')
6070 FORMAT (5(1PE8.2, 12X), 4X, I1, 4X, 'H')
6080 FORMAT ('1')
6090 FORMAT (' ', 'FREQ(', I3, ') = ', F7.2)
6100 FORMAT ('1', 54X, 'HISTOGRAM')
6110 FORMAT (' ', 51X, 'NUMBER OF CELLS')
6120 FORMAT (' ', 42X, 'NUMBER OF CELLS (WEIGHTED FOR PH)')
6130 FORMAT (' ', 47X, 'VS ACTUAL STANDOFF (CD)')
6140 FORMAT ('0', 42X, 'TARGET IS ''', A, '''')
6150 FORMAT (' ', 30X, 'HORIZONTAL UNIT LENGTH IS ', 1PE9.2,
    1           ' CD  ( = ', 1PE9.2, ' MM)')
6160 FORMAT (' ', 42X, 'VERTICAL UNIT LENGTH IS ', 1PE9.2)
6170 FORMAT (11X, A)
6180 FORMAT (' ', A)
     END
```

```
      SUBROUTINE SILOET (FNAME, METRIC, TARGNA)
C         DERIVED FROM WILSON'S 'SILPK' SUBROUTINE,
C         OBTAINED:  8 JUNE 1983
C         DRASTICALLY REWORKED BY PAUL TANENBAUM

      INTEGER COL, IH, IV, IVC, LABEL, ROW, STARTC, STARTL, STOPC,
     1        STOPL, VALUE
      REAL AZIM, CD, CELSIZ, ELEV, PK, STDOFF, VAL, WT, X, XMAX, XMAX1,
     1     XMIN, XMIN1, Y, YMAX, YMAX1, YMIN, YMIN1
      LOGICAL DEBUG, LASPAG, METRIC
      CHARACTER *7 FNAME
      CHARACTER *2 BLNK, HLABEL, VLABEL, HZERO, IMAGE, VZERO
      CHARACTER *20 TARGNA
      DIMENSION IMAGE(250, 130), HLABEL(250), VLABEL(130), VAL(3)
      LEVEL 2, /BUFFER/
      COMMON /ASPECT/ AZIM, ELEV
      COMMON /BUFFER/ IMAGE
      COMMON /CELSIZ/ CELSIZ
      COMMON /DEBUG/ DEBUG
      COMMON /ROUND/ CD, STDOFF
      DATA VZERO /'--'/
      DATA HZERO /''''''/
      DATA BLNK/'  '/

C     ------------------------------------------------------------

      IF (DEBUG) THEN
        WRITE (*, 610)
      ENDIF

C ----    BLANK-FILL HLABEL(), VLABEL(), AND IMAGE()
      DO 10 COL = 1, 250
        HLABEL(COL) = BLNK
   10 CONTINUE
      DO 20 ROW = 1, 130
        VLABEL(ROW) = BLNK
   20 CONTINUE
      DO 40 COL = 1, 250
        DO 30 ROW = 1, 130
          IMAGE(COL, ROW) = BLNK
   30   CONTINUE
   40 CONTINUE

C ----    DEFINE DIMENSIONS OF OUTPUT
      XMAX = -125 * CELSIZ
      XMIN = 125 * CELSIZ
      YMAX = -56 * CELSIZ
      YMIN = 56 * CELSIZ
      XMAXI = XMIN
      XMINI = XMAX
      YMAXI = YMIN
      YMINI = YMAX
      LASPAG = .FALSE.
```

```
C ----    READ DATA FOR EACH CELL
      OPEN (5, FILE=FNAME)
      REWIND 5
   50 READ (5, 510, ERR=90, END=60) X, Y, PK, WT
      PK = PK * WT
C ----    REDEFINE DIMENSIONS OF OUTPUT, IF NECESSARY
        XMAX = MAX(XMAX, X)
        XMIN = MIN(XMIN, X)
        YMAX = MAX(YMAX, Y)
        YMIN = MIN(YMIN, Y)

C ----    WRITE INTO HLABEL() AND VLABEL()
        COL = ((X - XMINI) / CELSIZ) + 1
        LABEL = ABS(X) / CELSIZ
        CALL NUMRIT(HLABEL(COL), LABEL, METRIC)
        ROW = ((YMAXI - Y) / CELSIZ) + 1
        LABEL = ABS(Y) / CELSIZ
        CALL NUMRIT(VLABEL(ROW), LABEL, METRIC)

C ----    WRITE INTO IMAGE()
        VALUE = (PK * 10) + .5
        CALL NUMRIT(IMAGE(COL, ROW), VALUE, .TRUE.)
      GOTO 50

C ----    PRINT HEADERS
   60 CLOSE (5)
      WRITE (*, 620) TARGNA
      WRITE (*, 630) ELEV, AZIM, CELSIZ

C ----    FIND CENTER OF TARGET
      COL = ((0 - XMINI) / CELSIZ) + 1
      ROW = ((YMAXI - 0) / CELSIZ) + 1
      HLABEL(COL) = HZERO
      VLABEL(ROW) = VZERO

C ----    SET HORZ AND VERT SPACING ON PAGE
      IH = (XMAX - XMIN) / CELSIZ + 1
      IV = (YMAX - YMIN) / CELSIZ + 1
      IVC = IV
      IF (IH .LE. 63) THEN
        IH = (63 - IH) / 2 + 1
        LASPAG = .TRUE.
      ELSE
        IH = 1
      ENDIF
      IF (IVC .LE. 55) THEN
        IV = (56 - IV) / 2
      ELSE
        IV = 1
      ENDIF

C ----    DETERMINE THE WINDOW OF IMAGE() TO PRINT
      STARTC = (XMIN - XMINI) / CELSIZ + 1 - IH
      STOPC = STARTC + 63
      STARTL = (YMAXI - YMAX) / CELSIZ + 1 - IV
      IF (IVC .LT. 56) THEN
        STOPL = STARTL + 55
```

```
      ELSE
        STOPL = (YMAXI - YMIN) / CELSIZ + 1 - IV
      STARTL = MAX(STARTL, 1)
      STARTC = MAX(STARTC, 1)
      ENDIF

C ----    PRINT SILHOUETTE
      DO 70 ROW = STARTL, STOPL
        WRITE (*, 640) VLABEL(ROW), (IMAGE(COL, ROW),
     1                 COL = STARTC, STOPC)
   70 CONTINUE
      WRITE (*, 640) VLABEL(1), (HLABEL(COL), COL = STARTC, STOPC)
      WRITE (*, 650)

C ----    DID THE SILHOUETTE FIT ON ONE PAGE?
      IF (LASPAG) THEN
        STOP
      ENDIF

C ----    PRINT REMAINDER OF SILHOUETTE
      WRITE (*, 660)
      WRITE (*, 670)
      STARTC = STOPC + 1
      STOPC = STARTC + 62
      DO 80 ROW = STARTL, STOPL
        WRITE (*, 640) (IMAGE(COL, ROW), COL = STARTC, STOPC),
     1                 VLABEL(ROW)
   80 CONTINUE
      WRITE (*, 640) (HLABEL(COL), COL = STARTC, STOPC), VLABEL(1)
      RETURN

C ----    HANDLE READ ERRORS
   90 WRITE (*, 680) '1READ SCREWUP...  SUBROUTINE ''SILOET'' ABORTED.'
      STOP

  510 FORMAT (3F8.2, F8.6)
  610 FORMAT (' ENTER SUBROUTINE ''SILOET''.')
  620 FORMAT('1', 48X, 'CELL-BY-CELL STANDOFF IN TENTHS OF A CD', /,
     1       53X, 'TARGET IS ', A20)
  630 FORMAT(' ', 29X, F5.0, ' DEGREES ELEVATION  ', F5.0,
     1       ' DEGREES AZIMUTH     CELL SIZE = ', F7.0, ' MM')
  640 FORMAT(1X,65A2)
  650 FORMAT(130X,'.')
  660 FORMAT('1')
  670 FORMAT(1X,/)
  680 FORMAT (A)
      END
```

```
      SUBROUTINE NUMRIT (WHITHER, WHAT, HOW)

      INTEGER WHAT
      LOGICAL HOW
      CHARACTER *2 WHITHER

C     ------------------------------------------------------------------

      IF (HOW) THEN
        IF (WHAT .GE. 101) THEN
          WRITE (WHITHER, 610)
        ELSE
          WRITE (WHITHER, 620) WHAT
        ENDIF
      ELSE
        IF (WHAT .GE. 100) THEN
          WRITE (WHITHER, 620) 99
        ELSE
          WRITE (WHITHER, 620) MOD(WHAT, 25) * 4
        ENDIF
      ENDIF
      RETURN

  610 FORMAT ("##")
  620 FORMAT (I2.2)
      END
```

```
      PROGRAM FUZES
C
C     COMPUTES PROBABILITY THAT A 2-FUZED PROJECTILE'S SECONDART FUZE
C     WILL STRIKE TARGET:
C      A) BEFORE PRIMARY FUZE
C      B) DURING PRIMARY'S DELAY
C
C     WRITTEN BY PAUL TANENBAUM, IN WHOM ALL BLAME LIES
C                 ATTN: SLCBR-VL-G
C                 APG, MD  21005-5066
C

      INTEGER ASPECT, CYL1, DLY, DLYBAD, FUZE2, FZ1MXC, NMASPS,
     1        NMCELLS, READNO
      REAL AZIM, CELLPH, CELSIZ, DIST, DLYP, ELEV, FUZE2P, HCEN, VCEN
      CHARACTER *10 DAY
      CHARACTER *60 ITITLE

C ----    BEGIN EXECUTION    -----------------------------------------
      OPEN (51, FILE='FUZINP')

      READNO = 1
      READ (51, 510, ERR=30, END=40) NMASPS, DAY, ITITLE

C ----    PROCESS NEXT VIEW
      DO 20 ASPECT = 1, NMASPS

C ----    INITIALIZE THINGS FOR THIS VIEW
        NMCELLS = 0
        FUZE2 = 0
        FUZE2P = 0
        DLY = 0
        DLYP = 0
        READNO = 2
        READ (51, 520, ERR=30, END=40) AZIM, ELEV, CELSIZ, FZ1MXC

C ----    FOR EACH CELL... READ DATA AND COMPUTE PROBABILITY THAT
C         SECONDARY FUZE WILL STRIKE FIRST AND PROBABILITT THAT IT
C ----    WILL HIT DURING PRIMARY-FUZE DELAY.
        READNO = 3
   10   READ (51, 530, ERR=30, END=40) HCEN, VCEN, DIST, CYL1, DLYBAD,
     1                                 CELLPH
        IF (HCEN .NE. 999.9) THEN
          IF (CYL1 .GT. FZ1MXC) THEN
            FUZE2 = FUZE2 + 1
            FUZE2P = FUZE2P + CELLPH
          ENDIF
          IF (DLYBAD .EQ. 1) THEN
            DLY = DLY + 1
            DLYP = DLYP + CELLPH
          ENDIF
          NMCELLS = NMCELLS + 1
          GOTO 10
        ENDIF
```

```
C ----     OUTPUT RESULTS
        WRITE (*, 610) ITITLE, DAY, AZIM, ELEV, CELSIZ, NMCELLS,
     1                 FUZE2, REAL(FUZE2) / REAL(NMCELLS), FUZE2P,
     2                 DLY, REAL(DLY) / REAL(NMCELLS), DLYP
   20 CONTINUE
      STOP

C ----     HANDLE INPUT ERRORS
   30 IF (READNO .EQ. 1) THEN
        WRITE (*, 620) READNO
      ELSE IF (READNO .EQ. 2) THEN
        WRITE (*, 630) READNO, ASPECT
      ELSE
        WRITE (*, 640) READNO, ASPECT, NMCELLS + 1
      ENDIF
      STOP
   40 IF (READNO .EQ. 1) THEN
        WRITE (*, 650) READNO
      ELSE IF (READNO .EQ. 2) THEN
        WRITE (*, 660) READNO, ASPECT
      ELSE
        WRITE (*, 670) READNO, ASPECT, NMCELLS + 1
      ENDIF
      STOP

  510 FORMAT (1X, I5, A10, A60)
  520 FORMAT (1X, 2F8.1, 24X, F8.1, I2)
  530 FORMAT (1X, F6.0, F7.1, F9.4, 2I5, F8.6)
  610 FORMAT ('1---------------- PROGRAM FUZES ----------------', /
     1        '0TARGET IS ''', A60, '''  RUN ON ', A10, /
     2        ' AZIMUTH                          ', F6.2, ' DEGREES', /
     3        ' ELEVATION                        ', F6.2, ' DEGREES', /
     4        ' CELL SIZE                       ', F7.2, ' MM', /
     5        ' NONEMPTY CELLS THIS VIEW          ', I5, /
     6        '0CELLS HIT BY SECONDARY FUZE       ', I5, /
     7        '    PROBABILITY GIVEN A HIT          ', F4.3, /
     8        '    PROBABILITY GIVEN A SHOT         ', F4.3, /
     9        '0CELLS HIT DURING FUZE DELAY       ', I5, /
     *        '    PROBABILITY GIVEN A HIT          ', F4.3, /
     1        '    PROBABILITY GIVEN A SHOT         ', F4.3, /
     2        ' ------------------------------------------------')
  620 FORMAT ('1INPUT ERROR ON READ NUMBER ', I1, /
     1        'PROGRAM ''FUZES'' ABORTED')
  630 FORMAT ('1INPUT ERROR ON READ NUMBER ', I1, ' ASPECT=', I3, /
     1        'PROGRAM ''FUZES'' ABORTED')
  640 FORMAT ('1INPUT ERROR ON READ NUMBER ', I1, ' ASPECT=', I3,
     1        ' CELL=', I4, / 'PROGRAM ''FUZES'' ABORTED')
  650 FORMAT ('1PREMATURE EOF ON READ NUMBER ', I1, /
     1        'PROGRAM ''FUZES'' ABORTED')
  660 FORMAT ('1PREMATURE EOF ON READ NUMBER ', I1, ' ASPECT=', I3, /
     1        'PROGRAM ''FUZES'' ABORTED')
  670 FORMAT ('1PREMATURE EOF ON READ NUMBER ', I1, ' ASPECT=', I3,
     1        ' CELL=', I4, / 'PROGRAM ''FUZES'' ABORTED')
      END
```

## APPENDIX B

## Sample Output

This page Left Intentionally Blank

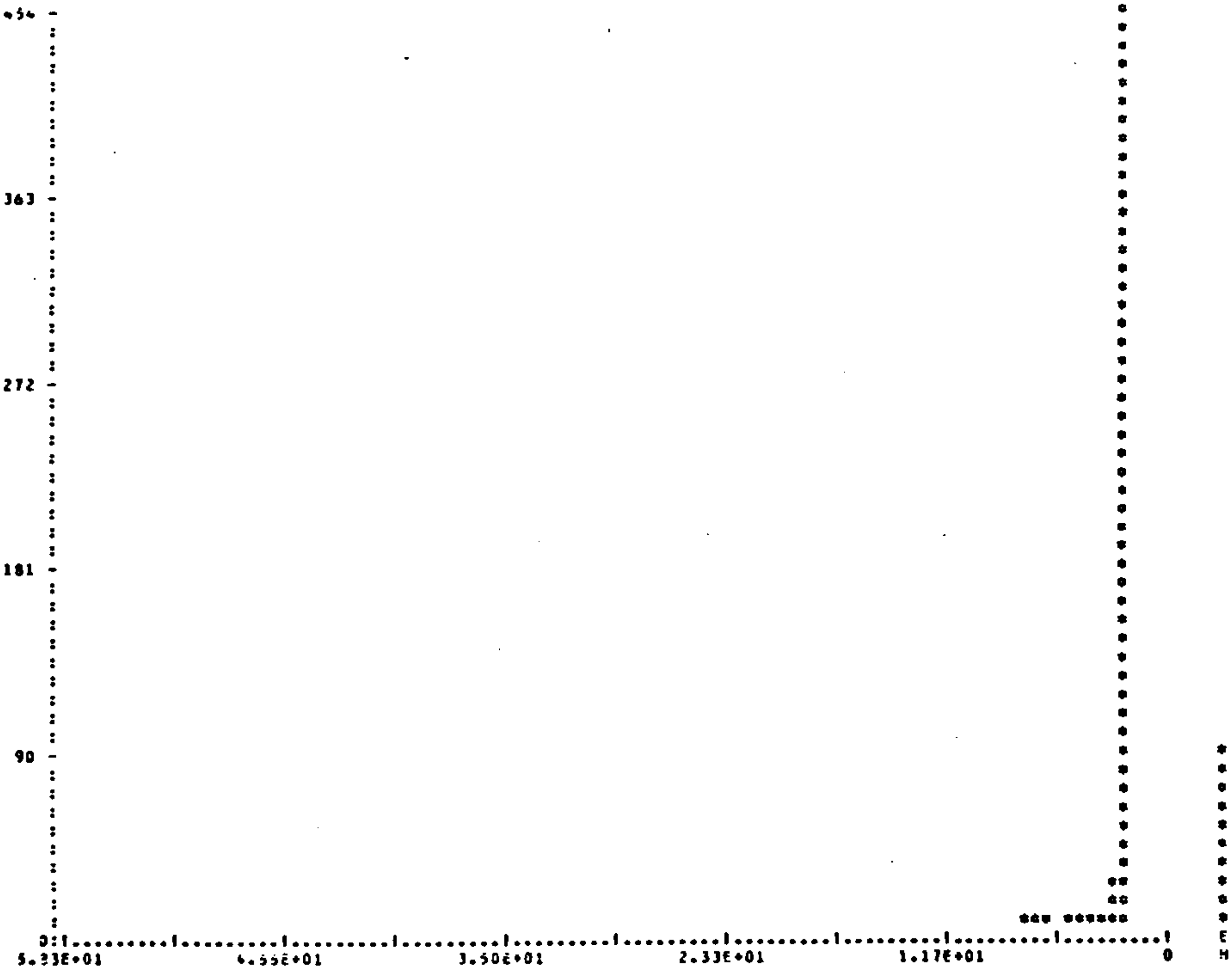
```
ENTER USER ROUTINE GRANDX

OPTION SET TO SUPPRESS PRINTER OUTPUT

---- GEOMETRY OF PROJECTILE SKIN (MM) ----
RADIUS OF CYLINDER     1         14.14
STEPBACK OF CYLINDER   1          5.86
RADIUS OF CYLINDER     2         20.00
STEPBACK OF CYLINDER   2         20.00
RADIUS OF CYLINDER     3         21.00
STEPBACK OF CYLINDER   3        153.00
RADIUS OF CYLINDER     4         57.45
STEPBACK OF CYLINDER   4        176.79
RADIUS OF CYLINDER     5         75.00
STEPBACK OF CYLINDER   5        225.00

PLANETARY RAYS PER CYLINDER                    8

AZIMUTH                    .000 DEGREES
ELEVATION                  .000 DEGREES

---- TARGET --------           X           Y           Z
TARGET MINIMUM (MM)      -3673.000  -1642.000        .000
TARGET MAXIMUM (MM)       6200.000   1642.000    2831.000
TARGET CENTER (MM)        1263.500       .000    1415.500
TARGET DIMENSIONS (MM)    9873.000   3284.000    2831.000

---- GRID PLANE ----
BACK OFF DISTANCE         6594.000 MM
CELL SIZE                  100.000 MM
HORIZONTAL LENGTH         3284.000 MM
VERTICAL LENGTH           2831.000 MM
CENTER (MM)                   .000   1415.500
HORIZONTAL RANGE (MM)    -1600.000   1600.000
VERTICAL RANGE (MM)           .000   2800.000

NUMBER HORZ CELLS               33
NUMBER VERT CELLS               29
NUMBER OF CELLS                957

FIRST SEED FOR RANDOM NUMBER GENERATOR            0

OPTION SET TO COMPUTE RANDOM POINT IN CELL
```

```
END OF CASE      1

FIRST SEED FOR RANDOM NUMBER GENERATOR                0
NEXT  SEED FOR RANDOM NUMBER GENERATOR         21837655

TIME FOR CASE     1    73.594 SECONDS


TOTAL TIME FOR CYLINDER    73.606 SECONDS

LEAVE USER ROUTINE BRANDX

END OF RUN
```

```
------ PROGRAM MISFIR ------

TARGET IS T-62A TANK DESCRIPTION (GIFT5)  RUN ON  09/23/85
AZIMUTH                          .00 DEGREES
ELEVATION                        .00 DEGREES
CELL SIZE                     100.00 MM

PROJECTILE IS DEMONSTRATION RUN
CHARGE DIAMETER               100.00 MM
BUILT-IN STANDOFF             250.00 MM
IMPACT VELOCITY               175.00 M/S
FUZE DELAY TIME                  .50 MS
X DISPERSION                400.000,      Y DISPERSION         400.000
X CORRECTION                   .000,      Y CORRECTION            .000
NUMBER OF CYLINDERS                5
LAST CYL IN PRIMARY FUZE           2
NUMBER OF RAYS PER CYL             8

STEPBACK OF CYLINDER  1    5.86
STEPBACK OF CYLINDER  2   20.00
STEPBACK OF CYLINDER  3  153.00
STEPBACK OF CYLINDER  4  176.79
STEPBACK OF CYLINDER  5  225.00
---------------------------
```

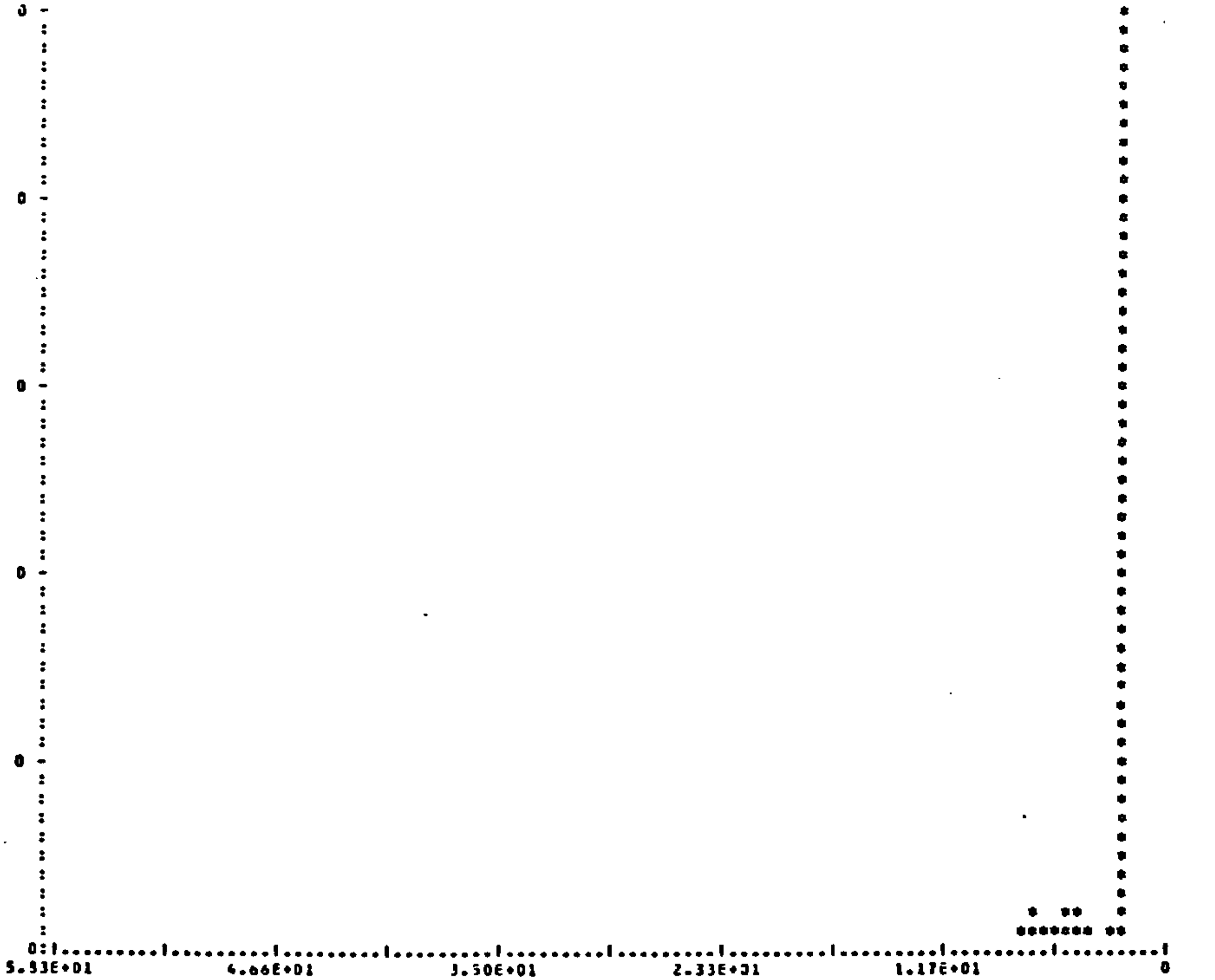
HISTOGRAM
NUMBER OF CELLS
VS ACTUAL STANDOFF (CD)

TARGET IS DEMONSTRATION RUN
HORIZONTAL UNIT LENGTH IS 5.83E-01 CD ( = 5.53E+01 MM)
VERTICAL UNIT LENGTH IS 9.08E+00



HISTOGRAM
NUMBER OF CELLS (WEIGHTED FOR PH)
VS ACTUAL STANDOFF (CD)

TARGET IS "DEMONSTRATION RUN "
HORIZONTAL UNIT LENGTH IS 5.53E-01 CD ( = 5.53E+01 MM)
VERTICAL UNIT LENGTH IS 1.56E-02



CELL-BY-CELL STANDOFF IN TENTHS OF A CD
TARGET IS DEMONSTRATION RUN
0. DEGREES ELEVATION     0. DEGREES AZIMUTH     CELL SIZE =    100. MM

```
28                 252500
27                 25250000
26                 0000****                   00000000
25                 0000**00                     252525
24                   00314400                 00252500  00
23               00230003**2525000000  000000004663717225OO
22               002627**252525257B**4842304733492525276900
21             00293131932525259732353534313537414430253529OO
20           0037292828282525522527296259273128282830293O7400
19         00322527272728262626272 7**25**262626262626232727 5800
18         002627262526262625252 5******25**25263126262627272729O0
17       002926272626262526252525 25****2525252525252626262627283000
16   00  0029****2676260925562525504257**4757697578702685262627 38  00
15   00000000*-**95**829573272753272727747375253425252527 99****25250000
14   2525262525252525562727258056255127272727272727272727272725**2525252525
13   7531322975756870272754257025252727272727272727272727273168323237276957
12   2629252627272956272727272727272727272727272727272727276727262727272829
11   2625262626252525272727272727272727272727272727272727782625262525 2626
10   25252525252525432727272727272727272727272727272727276925252525252500
09   4040252525252541272727252725272727272527272727252731424025254025 00
08   2525252525254526262626262626262626252626262626252626262725252525 2525
07   25252525252575322626262626262626262626262626262626**2525252525 00
06   262525262525269226262625262526262626262626262626265667262626262626
05   2626272726277725262625262630002626000026002626262540502727272727 27
04   0026262626262600**0000     00  0000      000000000000002727272727 2727
03   0026262626262600                                  00272727272727
02   0026262626262600                                  27272727272727
01   0026262626262500                                  00272727272700
--   25252600000000                                    00272700002700

     161514131211100908070605040302010""01020304050607080910111213141516
```

```
---------------- PROGRAM FUZES ----------------

TARGET IS 'T-62A TANK DESCRIPTION (GIFT5)  RUN ON  09/23/85
AZIMUTH                                   .00 DEGREES
ELEVATION                                 .00 DEGREES
CELL SIZE                              100.00 MM
NONEMPTY CELLS THIS VIEW                  677

CELLS HIT BY SECONDARY FUZE               146
  PROBABILITY GIVEN A HIT                .216
  PROBABILITY GIVEN A SHOT               .121

CELLS HIT DURING FUZE DELAY                55
  PROBABILITY GIVEN A HIT                .081
  PROBABILITY GIVEN A SHOT               .053
-----------------------------------------------
```

This page Left Intentionally Blank

DISTRIBUTION LIST

| No. of Copies | Organization |
| --- | --- |
| 12 | Administrator<br>Defense Technical Info Center<br>ATTN: DTIC-DDA<br>Cameron Station<br>Alexandria, VA 22314-6145 |
| 1 | HQDA<br>ATTN: DAMA-WSW<br>Washington, DC 20310 |
| 1 | HQDA<br>ATTN: DAMA-ART-M<br>Washington, DC 20310 |
| 1 | Commander<br>US Army Materiel Command<br>ATTN: AMCDRA-ST<br>5001 Eisenhower Avenue<br>Alexandria, VA 22333-0001 |
| 1 | Commander<br>US Army Materiel Command<br>ATTN: AMCLDC<br>5001 Eisenhower Avenue<br>Alexandria, VA 22333 |
| 1 | Commander<br>US Army Materiel Command<br>ATTN: AMCDE-DW<br>5001 Eisenhower Avenue<br>Alexandria, VA 22333 |

| No. of Copies | Organization |
| --- | --- |
| 10 | Central Intelligence Agency<br>Office of Central Reference<br>Dissemination Branch<br>Rm. GE-47, HQS<br>Washington, DC 20502 |
| 1 | Central Intelligence Agency<br>OSWR/OSD/GPWB<br>Washington, DC 20505 |
| 1 | Commander<br>Armament Research & Development Center<br>USAAMCCOM<br>ATTN: SMCAR-TDC<br>Dover, NJ 07801 |
| 2 | Commander<br>Armament Research & Development Center<br>USAAMCCOM<br>ATTN: SMCAR-TSS<br>Dover, NJ 07801 |
| 1 | Commander<br>Armament Research & Development Center<br>USAAMCCOM<br>ATTN: SMCAR-LC<br>Dover, NJ 07801 |

## DISTRIBUTION LIST

| No. of Copies | Organization |
|---|---|
| 1 | Commander<br>US Army Armament Munitions<br>& Chemical Command<br>ATTN: SMCAR-LEP-L<br>Rock Island, IL 61299 |
| 1 | Commander<br>US Army Armament Munitions<br>& Chemical Command<br>ATTN: SMCAR-ESP-L<br>Rock Island, IL 61299 |
| 1 | Director<br>Benet Weapons Laboratory<br>Armament Research &<br>Development Center<br>USAAMCCOM<br>ATTN: SMCAR-LCB-TL<br>Watervliet, NY 12189 |
| 1 | Commander<br>US Army Aviation Research<br>& Development Command<br>ATTN: AMSAV-E<br>4300 Goodfellow Blvd<br>St. Louis, MO 63120 |
| 1 | Director<br>US Army Air Mobility Research<br>& Development Laboratory<br>Ames Research Center<br>Moffett Field, CA 94035 |
| 1 | Commander<br>US Army Communications-<br>Electronics Command<br>ATTN: AMSEL-ED<br>Fort Monmouth, NJ 07703 |
| 1 | Commander<br>ERADCOM Technical Library<br>ATTN: DELSD-L (Reports Section)<br>Fort Monmouth, NJ 07703-5301 |
| 1 | Commander<br>US Army Missile Command<br>Research, Development, &<br>Engineering Center<br>ATTN: AMSMI-RD<br>Redstone Arsenal, AL 35898 |
| 1 | Director<br>US Army Missile & Space<br>Intelligence Center<br>ATTN: AIAMS-YDL<br>Redstone Arsenal, AL 35898-5500 |
| 1 | Commander<br>US Army Mobility Equipment<br>Research & Development<br>Command<br>ATTN: DRDME-WC<br>Fort Belvoir, VA 22060 |
| 1 | Commander<br>US Army Mobility Equipment Cmd<br>4300 Goodfellow Blvd<br>St. Louis, MO 63120 |

## DISTRIBUTION LIST

| No. of Copies | Organization |
|---|---|
| 1 | Commander<br>US Army Tank Automotive Command<br>ATTN: AMSTA-CG<br>Warren, MI 48090 |
| 1 | Commander<br>US Army Tank Automotive Command<br>ATTN: AMSTA-TSL<br>Warren, MI 48090 |
| 1 | US Army Armor & Engineer Board<br>ATTN: STEBB-AD-S<br>Fort Knox, KY 40121 |
| 1 | Project Manager<br>Tank Systems<br>ATTN: AMCPM-GCM-SW<br>Warren, MI 48090 |
| 1 | Program Manager<br>Tank Systems<br>ATTN: AMCPM-GCM-SA<br>Warren, MI 48090 |
| 1 | Project Manager<br>Fighting Vehicle Systems<br>ATTN: AMCPM-FVS<br>Warren, MI 48090 |
| 1 | Project Manager<br>Tank Main Armament Systems<br>ATTN: AMCPM-TMA-PA<br>Dover, NJ 07801 |

| No. of Copies | Organization |
|---|---|
| 1 | Commander<br>Combined Arms Center<br>ATTN: ATZL-CG<br>Fort Leavenworth, KS 66027 |
| 1 | Commander<br>US Army Combined Arms Combat Developments Activity<br>ATTN: ATCA-SA<br>Fort Leavenworth, KS 66027 |
| 1 | Commander<br>Concepts Analysis Agency<br>8120 Woodmont Avenue<br>Bethesda, MD 20014 |
| 1 | Commander<br>US Army Foreign Science & Technology Center<br>ATTN: AMXST-MC-3<br>220 Seventh Street, NE<br>Charlottesville, VA 22901 |
| 1 | Commander<br>US Army Training & Doctrine Cmd<br>ATTN: ATCD<br>Fort Monroe, VA 23351 |
| 1 | Commander<br>US Army Training & Doctrine Cmd<br>ATTN: ATCD-M<br>Fort Monroe, VA 23351 |

## DISTRIBUTION LIST

| No. of Copies | Organization |
|---|---|
| 1 | Director<br>US Army TRADOC Systems<br>Analysis Activity<br>ATTN: ATAA-SL, Tech Lib<br>White Sands Missile Range,<br>NM 88002 |
| 1 | Director<br>US Army TRADOC Systems<br>Analysis Activity<br>ATTN: ATAA-TEM<br>White Sands Missile Range,<br>NM 88002 |
| 1 | Director<br>US Army TRADOC Systems<br>Analysis Activity<br>ATTN: ATAA-TFB<br>White Sands Missile Range,<br>NM 88002 |
| 1 | Commander<br>US Army Field Artillery School<br>Fort Sill, OK 73503 |
| 1 | Commandant<br>US Army Infantry School<br>ATTN: Infantry Agency<br>Fort Benning, GA 31905 |
| 1 | Commandant<br>US Army Infantry School<br>ATTN: ATSH-CD-CSO-OR<br>Fort Benning, GA 31905 |

| No. of Copies | Organization |
|---|---|
| 1 | Commander<br>US Army Armor Center<br>ATTN: ATZK-CG<br>Fort Knox, KY 40121 |
| 1 | Commander<br>US Army Armor Center<br>ATTN: ATZK-CD<br>Fort Knox, KY 40121 |
| 1 | Commander<br>US Army Armor Center<br>ATTN: ATZK-M1<br>Fort Knox, KY 40121 |
| 1 | Commander<br>US Army Development &<br>Employment<br>Agency<br>ATTN: MODE-TED-SAB<br>Fort Lewis, WA 98433 |
| 1 | Air Force Armament Laboratory<br>ATTN: AFATL/DLODL<br>Eglin AFB, FL 32542-5000 |
| 1 | Commander<br>US Marine Corps<br>ATTN: APW<br>Washington, DC 20380 |

## DISTRIBUTION LIST

| No. of Copies | Organization |
|---|---|
| 1 | Commander<br>Marine Corps Development &<br>Education Command<br>ATTN: Plans & Studies Div<br>Quantico, VA 22134 |
| 1 | Commander<br>Naval Weapons Center<br>China Lake, CA 93555 |
| 1 | Commander<br>Marine Corps Development &<br>Education Command<br>ATTN: Firepower Division<br>Quantico, VA 22134 |
| 1 | AFELM, The Rand Corporation<br>ATTN: Library-D<br>1700 Main Street<br>Santa Monica, CA 90406 |

Aberdeen Proving Ground

Dir, USAMSAA
ATTN: AMXSY-D
AMXSY-MP (H. Cohen)
Cdr, USATECOM
ATTN: AMSTE-TO-F
Cdr, CRDC, AMCCOM, Bldg. E3516
ATTN: SMCCR-RSP-A
SMCCR-MU
SMCCR-SPS-IL

This page Left Intentionally Blank