Calhoun: The NPS Institutional Archive

DSpace Repository

Theses and Dissertations | 1. Thesis and Dissertation Collection, all items

1981-06

A comparison of computer warhead-target endgame simulations and recommendations pertaining to future endgame programs.

Fair, Christopher Kenneth

Monterey, California. Naval Postgraduate School

https://hdl.handle.net/10945/20457

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

Downloaded from NPS Archive: Calhoun

http://www.nps.edu/library

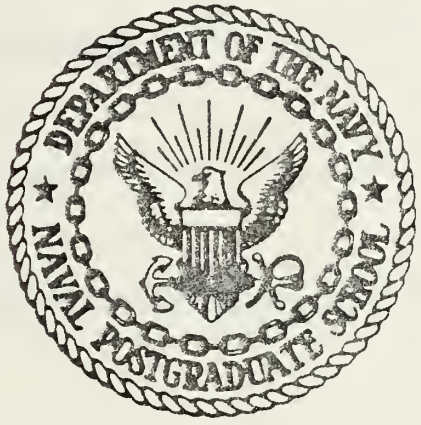
Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

A COMPARISON OF COMPUTER WARHEAD-TARGET ENDGAME SIMULATIONS AND RECOMMENDATIONS PERTAINING TO FUTURE ENDGAME PROGRAMS

Christopher Kenneth Fair

NAVAL POSTGRADUATE SCHOOL
Monterey, California



THESIS

A COMPARISON OF COMPUTER WARHEAD-TARGET ENDGAME SIMULATIONS AND RECOMMENDATIONS PERTAINING TO FUTURE ENDGAME PROGRAMS

by

Christopher Kenneth Fair

June 1981

Thesis Advisor: R. E. Ball

Approved for public release, distribution unlimited

T199897

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Comparison of Computer Warhead-Target Endgame Simulations and Recommendations Pertaining to Future Endgame Programs		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis June 1981 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Christopher Kenneth Fair		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE June 1981 13. NUMBER OF PAGES 127
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) 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)

SCAN, REFMOD, SESTEM II, SHAZAM, ATTACK, Endgame, computer, simulations, programming, missile, probability of kill(PK).

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

This study involves a description of several computer war-head Endgame simulations currently in use at various research facilities throughout the country. A comparison of the techniques and methods used in these programs is included. Recommendations and discussion

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 68 IS OBSOLETE
S/N 0102-014-6601
(Page 1)

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. concerning a desirable format for future computer Endgame simulations are included.

In addition, as a part of this effort, the Endgame program SCAN has been installed on the Naval Postgraduate School's IBM 3033 Computer system, and a Users Manual for the use of this program has been prepared.

Approved for public release, distribution unlimited

A Comparison of Computer Warhead-Target Endgame Simulations and Recommendations Pertaining to Future Endgame Programs

by

Christopher Kenneth Fair
Lieutenant, United States Navy
B.A., University of Washington, 1975

Submitted in partial fullfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ENGINEERING SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 1981

ABSTRACT

This study involves a description of several computer warhead-target Endgame simulations currently in use at various research facilities throughout the country. A comparison of the techniques and methods used in these programs is included. Recommendations and discussion concerning a desirable format for future computer Endgame simulations are included.

In addition, as a part of this effort, the Endgame program SCAN has been installed on the Naval Postgraduate School's IBM 3033 Computer System, and a Users Manual for the use of this program has been prepared.

TABLE OF CONTENTS

I. INTRODUCTION ---------- 12

II. EXISTING ENDGAME SIMULATION PROGRAMS ---------- 16

A. GENERAL ---------- 16

B. ATTACK ---------- 16

C. SCAN ---------- 26

D. SESTEM II ---------- 34

E. SHAZAM ---------- 41

F. REFMOD ---------- 44

III. COMPARISON OF FEATURES AND TECHNIQUES ---------- 60

A. GEOMETRIC MODELING; FINITE ELEMENT VERSUS COMBINATORIAL ---------- 60

B. VULNERABLE AREA VERSUS COMPONENT PROBABILITY OF KILL ---------- 63

C. WARHEADS AND FUZING ---------- 66

D. INPUT FORMAT ---------- 68

E. AVAILABILITY OF TARGET MODELS AND OTHER INPUT DATA ---------- 69

F. PROGRAM OUTPUT ---------- 69

G. GRAPHICS CAPABILITY ---------- 70

H. COMPUTATIONAL TIME AND COST REQUIREMENTS --- 70

I. DOCUMENTATION ---------- 72

IV. RECOMMENDATIONS FOR FUTURE ENDGAME PROGRAMS ---- 73

A. BASIC REQUIREMENTS ---------- 73

B. TARGET MODELING AND VULNERABILITY TYPE------- 74

C. WARHEAD AND FUZE MODELING ---------- 75

D. INPUT FORMAT ---------- 75

E. OUTPUT FORMAT ---------- 75

F. FLYOUT SIMULATION INTERFACE ---------- 76

G. GRAPHICS CAPABILITY ---------- 76

H. DOCUMENTATION ---------- 77

V. DIRECTION OF CURRENT RESEARCH ---------- 82

A. SCAN ---------- 82

B. SHAZAM ---------- 82

C. REFMOD ---------- 82
APPENDIX A ---------- 83
APPENDIX B ---------- 120
LIST OF REFERENCES ---------- 125
BIBLIOGRAPHY ---------- 126
INITIAL DISTRIBUTION LIST ---------- 127

LIST OF TABLES

Table		Page
II-1	Listing of Fuze Logics ----------	22
II-2	List of Material Types ----------	31
II-3	Vulnerability/Susceptability Options --------	32
AA-1	Targets Available for Use with SCAN ---------	85
AA-2	List of Material Types ----------	96
AA-3	Fragment Shapes for Use with SCAN Warhead Model ----------	98
AA-4	Missile Trajectory Options ----------	104

LIST OF FIGURES

Figure		Page
2-1	ATTACK Direct Hit Model	18
2-2	ATTACK Missile Representation	19
2-3	ATTACK Blast Model	20
2-4	ATTACK Structural Damage Model	23
2-5	ATTACK Single Fragment Vulnerable Components	25
2-6	ATTACK Encounter Geometry	27
2-7	Geometric Representation of the Exterior of the F-16 Aircraft	29
2-8	SESTEM II Terminal Geometry	36
2-9	SESTEM II Fuselage Model	39
2-10	SESTEM II Wing Model	40
2-11	SESTEM II Blast Contour	42
2-12	SESTEM II Warhead Model	43
2-13	REFMOD Direct Hit Modeling	47
2-14	REFMOD Blast Cylinder Model	48
2-15	REFMOD Structural Failure Model	49
2-16(a)	REFMOD Spherical Vulnerable Component	51
2-16(b)	REFMOD Linear Vulnerable Component	52
2-16(c)	REFMOD Cylindrical Vulnerable Component	53
2-16(d)	REFMOD Planar Vulnerable Component	54
2-17	REFMOD Mass/Velocity/Density Vulnerable Model	55

2-18 REFMOD Mass/Velocity/Density Function ------- 56

2-19 REFMOD Warhead Types ---------- 58

2-20 REFMOD Fuzing Model ---------- 59

3-1 Finite Element Model ---------- 61

3-2 COMGEOM Model ---------- 62

3-3 Aspect Angles for Vulnerable Area ---------- 64

3-4 Vulnerable Area Fuze Model ---------- 67

3-5 Computer Generated Graphics ---------- 71

4-1(a) Exterior View Graphics Example ---------- 78

4-1(b) Interior Components Graphics Example -------- 79

4-1(c) Exterior View Graphics Showing Hits --------- 80

A-1 SCAN1 FORTRAN Example Output ---------- 88

A-2(a) SCAN2 FORTRAN Example Output ---------- 89

A-2(b) SCAN2 FORTRAN Example Output ---------- 90

A-2(c) SCAN2 FORTRAN Example Output ---------- 91

A-3 Assumed Target CG ---------- 92

A-4 Warhead Static Polar Zones ---------- 95

A-5 SCAN Blast Model ---------- 101

A-6 Target Roll, Pitch and Yaw Angles ---------- 106

A-7 Missile Angle of Attack ---------- 107

A-8 Missile Pitch and Azimuth Angles ---------- 109

A-9 SCAN Coordinate System Relationships -------- 110

A-10 WARHEAD DATA and CASE DATA Examples --------- 115

A-11(a) WARHEAD DATA Quick Reference Guide ---------- 116

A-11(b) WARHEAD DATA Quick Reference Guide ---------- 117

A-12(a) CASE DATA Quick Reference Guide ---------- 118

A-12(b) CASE DATA Quick Reference Guide ---------- 119

ACKNOWLEDGEMENT

I wish to thank Russ Cramer, Pat Caulfield, Ken Grant, Bruce Nofrey and Gary Spradling for their assistance in obtaining reference material and information pertaining to computer Endgame simulations. I would also like to thank Professor R. E. Ball for the guidance and assistance he has provided during my studies at the Naval Postgraduate School.

I. INTRODUCTION

An aircraft versus missile engagement is traditionally described in two phases, the missile flyout and the Endgame. That portion of the engagement from missile launch to the terminal flight phase where the missile is in the vicinity of the target is called the missile flyout. The Endgame phase includes the missile fuzing sequence for target detection and subsequent warhead detonation, followed by an evaluation of the effectiveness of the damage mechanisms associated with a warhead on the target under the missile/aircraft encounter conditions at the time of warhead detonation.

The overall measure of effectiveness of a missile system against an aircraft can be stated in terms of the single shot probability of kill, PKSS. The PKSS value encompasses all missile system functions from launch through the complete target engagement. Significant factors considered in computing PKSS are launcher and missile reliability, guidance and control, fuzing, warhead detonation and evaluation of target damage and target kill. PKSS can be expressed in the general form as

$$PKSS = Pl * Pg/l * Pf/g * Pd/f * Ph/d * Pk/h$$

where Pl is the probability of launch, Pg/l is the probability (or occurance) ofsuccessful guidance given launch, Pf/g is the probability (or occurance) of fuzing given guidance, Ph/d is the probability (or number) of hits on the target given detonation, and Pk/h is the probability of target kill given the hits on the target.

Missile Endgame programs measure the probability of kill, PK/E, of a terminal encounter. This is done by

assuming that the probability of missile launch and sucessful guidance (Pg/l * Pl) is unity; therefore the missile is always in the vicinity of the target at the beginning of the encounter. The final result of the Endgame computations is an overall probability of kill given an encounter which can be expressed in the general form as

$$PK/E = Pf/g * Pd/f * Ph/d * Pk/h$$

The target detection, or fuzing, and the detonation are dependent upon the fuze operation and logic, the encounter geometry and the target signature. In many Endgame programs there are options to set Pf/g and Pd/f equal to unity in order to eliminate the fuze from consideration.

The value of Ph/d, or the number and location of hits on the target, is computed by the program and is a function of the encounter geometry conditions, the target size and the missile warhead parameters. The Pk/h is determined by the design of the target model with it's associated component Pk/h functions or by the component vulnerable area (Av) tables for a given target, depending upon the type of Endgame simulation used. These two values are utilized in computing Pk/d, the probability of kill given a detonation, which is of the general form

$$PK/D = Ph/d * Pk/h$$

The PK/E determined by the Endgame programs is not a single shot probability of kill PKSS. PKSS considers the entire engagement and hence involves many factors that are not accounted for in PK/E alone. The reliability of the missile and launcher, missile performance, the guidance and control, and many other factors are considered in computing

PKSS. It can therefore be seen that PK/E is only valid for the final terminal phase of an engagement and assumes all missile system functions prior to that point are perfect.

The need for valid Endgame simulation is obvious. The cost of conducting extensive tests on every missile/aircraft combination is not only prohibitive, but very time consuming and may be impossible due to unavailability of the systems. The modeling of encounters does require extensive physical test data, but does not require entire new systems to be tested; only those components for which no data is currently available need be tested. The time and money required for one physical test can be used to produce literally thousands of simulation runs. Simulation can also be used in planning an actual test firing to optimize the probability of collecting the data desired. The use of Endgame simulation will allow designers to evaluate changes made to missile or aircraft systems much earlier in the design process, allowing the manufacturer to produce a product needing fewer modifications to meet both performance and survivability/effectiveness specifications. A service interested in buying a new missile or aircraft will be able to conduct preliminary "fly-offs" between proposed designs prior to funding full scale prototypes in order to narrow the field of contenders to those meeting the desired specifications. The time required to go from preliminary design to an engineering model can be reduced substantially if non-viable options can be reduced or eliminated through simulated encounters. The use of Endgame simulations to develop improved tactics and engagement methods for both missiles and aircraft could result in substantially increased effectiveness of forces presently available. New or improved enemy systems can be modeled quickly and, as additional data becomes available,

updated to provide new tactics to counter the threat. Several of the aforementioned uses of Endgame simulation are currently functioning at various military and civilian research facilities, but much more needs to be done to realize fully the potential of this type of computer simulation.

Consequently, the main body of this study study encompasses (1) a description of several current Endgame programs, (2) a comparison of features and techniques used in computer simulation, (3) suggestions for developing improved Endgame programs and (4) a look at Endgame programs currently being developed or undergoing major modification.

The intent of this study is to provide potential users of Endgame simulation a guide to use in the selection of a specific program, to provide Endgame programmers with feedback concerning desirable features for future simulations, and to suggest additional research concerning Endgame simulation programs.

In addition, as a part of this effort, the Endgame program SCAN has been installed on the Naval Postgraduate School's IBM 3033 Computer System, and a Users Manual describing the preparation of input data and execution of the program on the IBM 3278 display terminal has been prepared. The Users Manual is given in Appendix A, and Appendix B contains the changes to SCAN required to make it operational on the IBM 3033 System and interface with the SCAN SPDRAW computer graphics program.

II. EXISTING ENDGAME SIMULATION PROGRAMS

A. GENERAL

The description of each program presented in this Chapter will include discussion about the type of modeling used, the input format, program options, how PK is assessed, the program originator and the original intended use of the simulation. This chapter is intended to present the various programs in a brief summary form and no attempt will be made here to evaluate the relative merits of the individual Endgame simulations.

B. ATTACK

The ATTACK program is the current version of the AIR-TO-AIR TERMINAL SIMULATION (NWC TN4565-1-70) [Ref. 1], a Naval Weapons Center, China Lake, revision of a methodology developed at the Naval Missile Center, Point Mugu. The documentation for ATTACK was published in June 1974. The objective of ATTACK, as stated in it's User Manual, "is to predict the ability of a missile to detect and destroy an airborne target." To this end, the program provides a PK assessment for four damage mechanisms. These are (1) direct hits, (2) blast, (3) multiple fragment (structural), and (4) single fragment (component) damage mechanisms.

This program requires four target geometric representations, one representation for each of the damage mechanisms. The fuzing portion of the program requires a fifth target representation which is dependent upon the fuze type selected.

The program is intended to provide results for the following purposes:

(1) Weapon system evaluation

(2) Warhead design

(3) Fuze optimization

(4) Survivability studies

(5) Trade-off studies

The methodology for damage assessment is composed of the following classes:

(1) Structural

  (a) direct hit model

  (b) blast model

  (c) multiple fragment model

(2) Component

  (a) single fragment model

The direct hit model consists of a target modeled using triangular plates (see Fig. 2-1) and a missile which is represented by a collection of points (see Fig. 2-2).

The missile trajectory is computed from a user specified encounter geometry. The program determines if one or more of the missile points will intersect the target and the time at which first contact will occur. If the first contact occurs before fuzing, a direct hit kill is assessed and kills by other possible damage mechanisms are not evaluated. If fuzing occurs prior to first contact then a "preempted" direct hit is recorded and noted in the output, but other damage mechanism kills are evaluated.

The blast model is composed of a group of cylinders and hemispherically shaped end caps surrounding the target body and extremities (see Fig. 2-3). The radius assigned to each of the blast cylinders is a function of both the strength of the included structure and the explosive charge of the warhead. The cylinder radii, which are determined external to the program in a seperate analysis, are scaled automatically to a user specified encounter altitude. If the warhead detonates within the volume of one of these blast cylinders, a blast kill occurs and no other damage mechanisms are

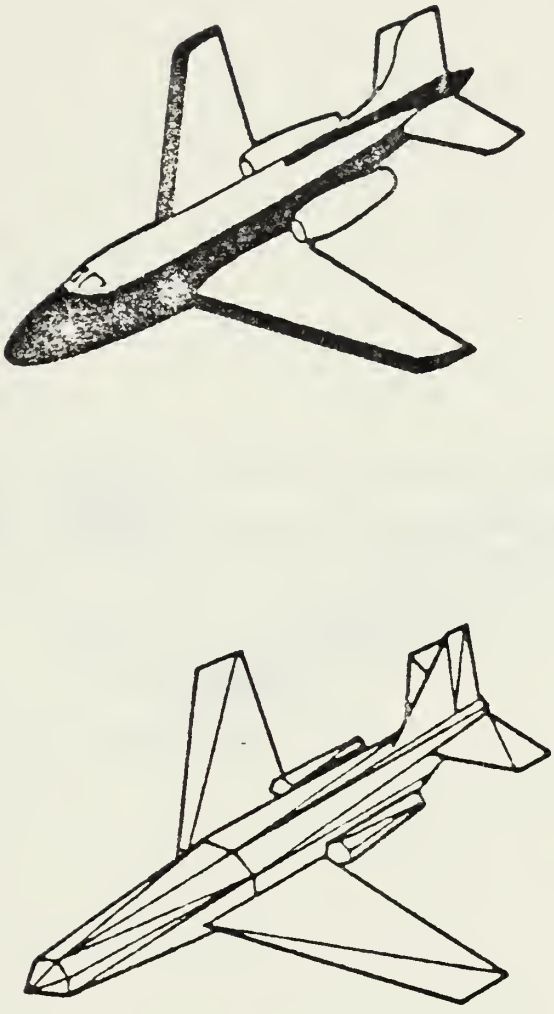


Fig. 2-1 ATTACK Direct Hit Model

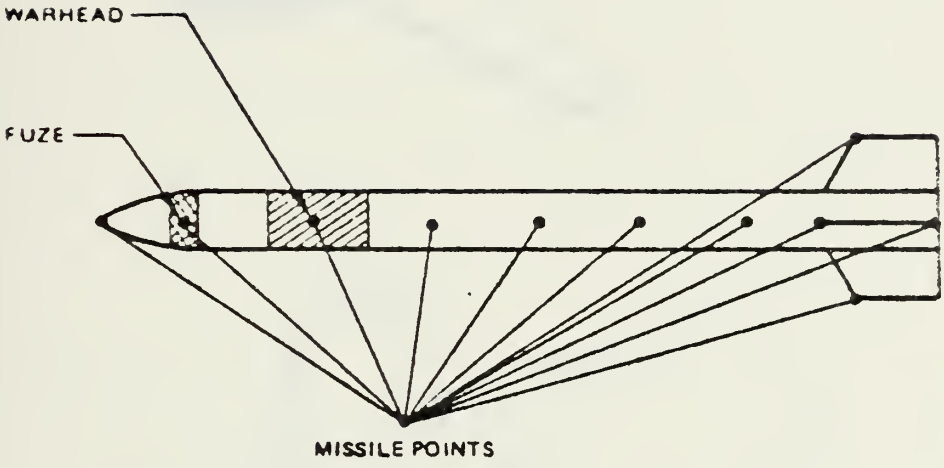


Fig. 2-2 ATTACK Missile Representation

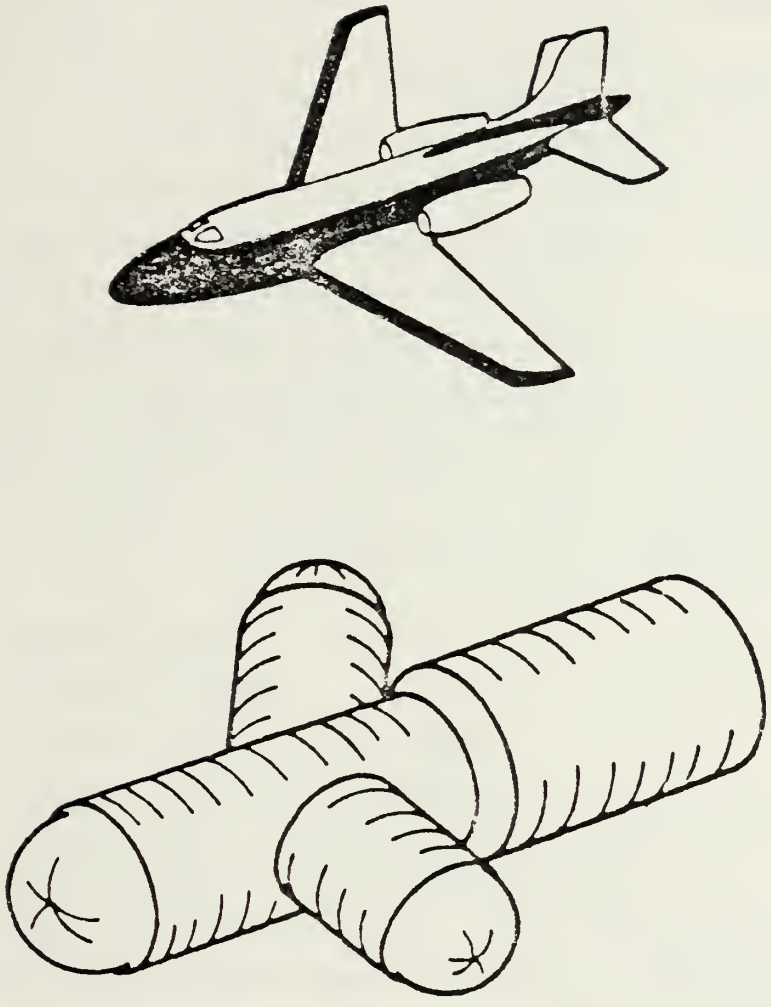


Fig. 2-3 ATTACK Blast Model

evaluated. If the warhead detonates outside the volume of the blast cylinder, no damage is credited to blast effects, and the program goes on to evaluate other damage modes.

The warhead model in ATTACK uses the concept of fragment spray zones and fragment weight classes. The warhead may have up to ten polar zones with up to five fragment mass classes in each. The user may also define up to eight radial zones which allows simulation of nonsymmetric fragment sprays. For detonation of the warhead, the user can choose from eleven fuze logics (the number of fuze logics may vary from one installation to another). A typical listing of fuze logics is given in Table II-1.

The multiple fragment model for structural damage uses a segmented cylindrical target representation (see Fig. 2-4). The program increments the centroid of the cylindrical segments by the target velocity vector from the time of warhead detonation. The fragment ballistics are computed as a function of:

(1) Fragment mass
(2) Fragment shape
(3) Fragment initial velocity
(4) Fragment drag coefficient
(5) Target range and aspect from warhead at detonation
(6) Fragment and target flight paths

The number of fragments with their associated energies which strike each cylindrical segment is determined by the location of the target segment within one or more of the warhead dynamic radial and polar zones. The energy density is computed and compared with a critical level of energy specified for that segment by the user. If the calculated value exceeds the specified energy density, a structural kill is assessed.

TABLE II-1

LISTING OF FUZE LOGICS

Logic 1	Semi-active doppler fuze
Logic 2	Semi-active doppler fuze with signal stretcher
Logic 3	Semi-active doppler fast track fuze
Logic 4	Semi-active doppler fuze for intercept arm; fixed angle fuze
Logic 5	Fixed angle active fuze
Logic 6	IR fuze operation in pursuit mode
Logic 7	Active fuze with fore and aft fixed angle fuze cones
Logic 8	Passive fixed angle fuze
Logic 9	Semi-active fuze with guard channel for intercept arm. Fixed angle for home on jam, fuze on jam
Logic 10	Semi-active doppler with guard channel arm
Logic 11	Instantaneous detection*

*Note: This option has been added to the NPS version

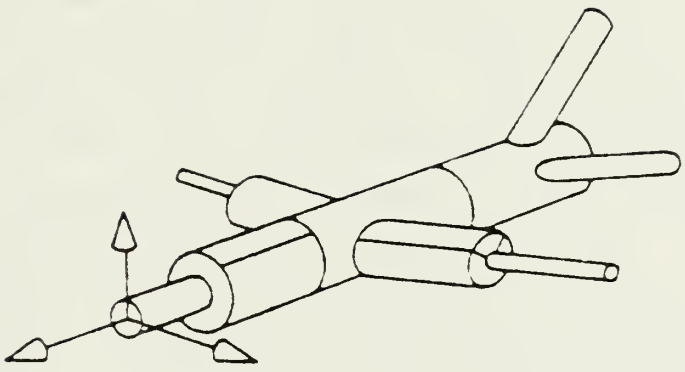


Fig. 2-4 ATTACK Structural Damage Model

The single fragment or component kill model consists of individual components, represented by spheres or points, located at appropriate positions with respect to the target coordinate system origin (see Fig. 2-5). The computational process for Pk used in this model is similar to that used in the multiple fragment case. The component (sphere) centroid location and radius are used to determine the fractional area (FRACT) of the component within a given polar and radial fragment spray zone.

$$\text{FRACT} = \frac{\text{area of the component hit by fragment spray (At)}}{\text{component presented area (Ap)}}$$

The vulnerability of the components to the impact of fragments is measured by the component vulnerable area, Av. Vulnerable area tables are input for each component in the model as a function of fragment impact aspect angle, fragment mass and fragment impact velocity. The distance of the component centroid from the warhead origin at detonation (DIST) is used to determine exact fragment impact velocity and the striking azimuth and elevation angles for a specific weight class. These are used in conjunction with the vulnerable area tables to compute the appropriate component vulnerable area (Av). A fragment beam area (FA) within the polar and radial zone boundaries is computed at the distance, DIST. The fragment spray density (RHO) is given by:

$$\text{RHO} = Q/\text{FA}$$

Where Q is the number of fragments in the weight class and polar/radial zone considered. The expected number of lethal hits (E) for the specified weight class is computed from:

$$E = \text{RHO} * \text{Av} * \text{FRACT}$$

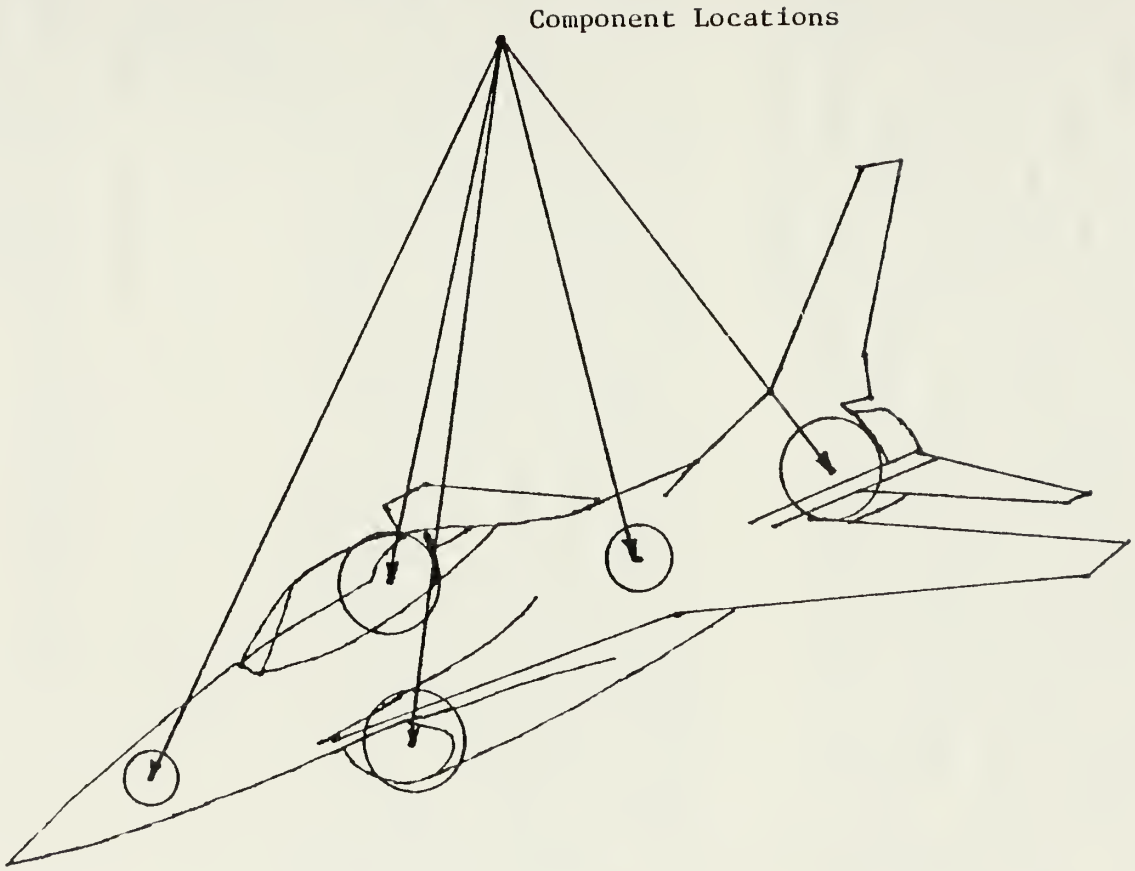


Fig. 2-5 ATTACK Single Fragment Vulnerable Components

The expected number of lethal hits is accumulated for each polar zone, radial zone and fragment weight class. The component Pk is computed by the following equation:

$$Pk(\text{component}) = 1.0 - \exp(-E)$$

The encounter geometry as shown in Figure 2-6 is specified by the user. The missile may be oriented with respect to either the target or to a relative velocity vector. The user may either specify a missile miss distance or closest point of approach (CPA) or utilize the program to generate random miss distances from a Gaussian distribution. The user may also specify a standard deviation for the miss distance distribution, and multiple trajectories may be simulated for a given scenario.

C. SCAN

SCAN is a digital computer program developed under the supervision of the Pacific Missile Test Center, Point Mugu [Ref. 2]. The documentation was completed in June 1976 under the auspices of the Joint Technical Coordinating Group on Aircraft Survivability (JCTG/AS). The objective of the SCAN Endgame simulation, as defined in the User Manual, is "to predict the probability that an aircraft will survive an attack by a missile armed with a warhead." A PK is computed for three cases:

(1) Direct hit
(2) Blast
(3) Fragment damage

This program can be used to provide data for:

(1) Aircraft design
(2) Aircraft survivabilty studies
(3) Justification of new survivability features

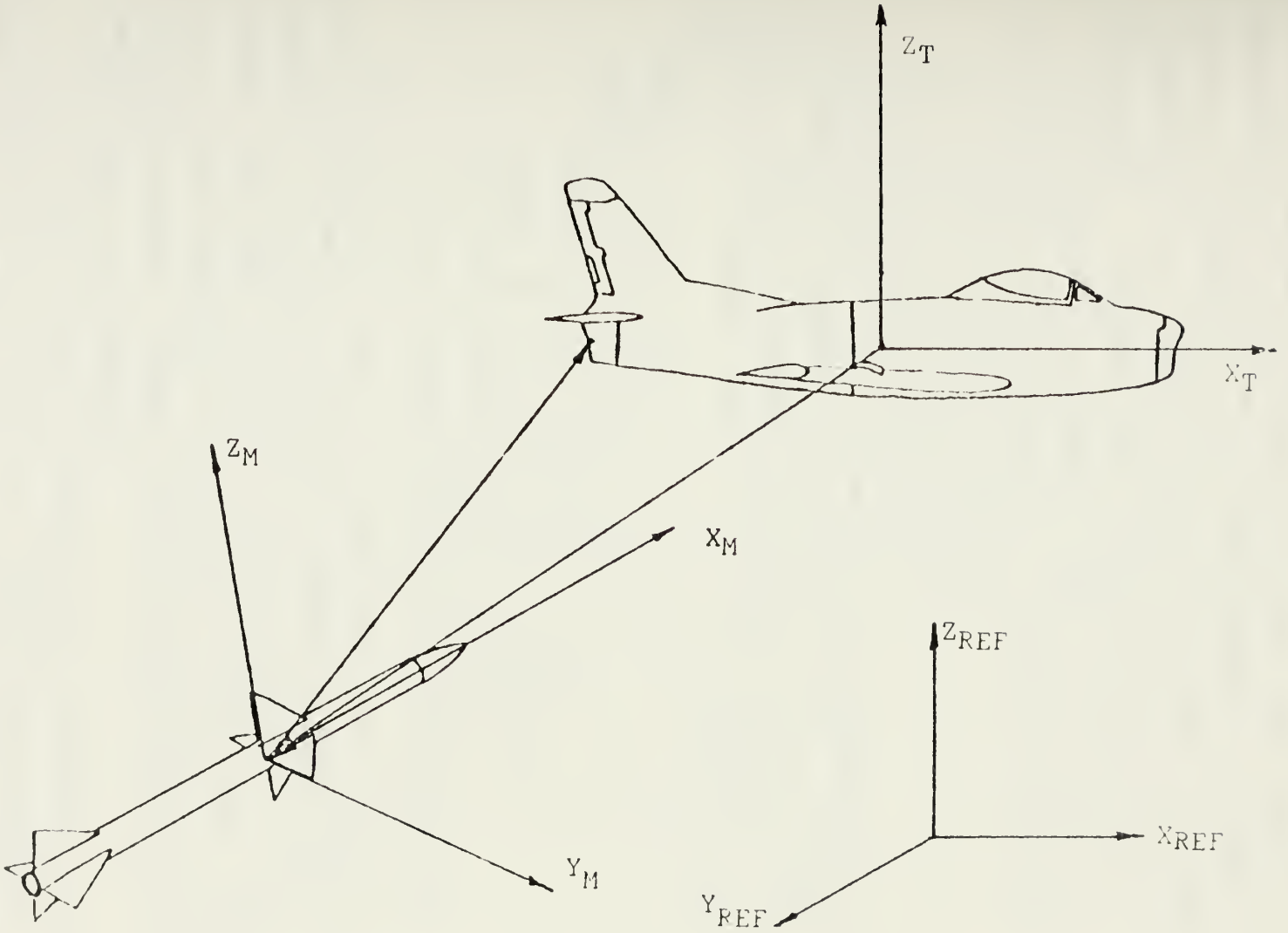


Fig. 2-6 ATTACK Encounter Geometry

One major feature of this program is the complex geometric model of the target. In SCAN, the target model is composed of a series of components, where each component is represented by one or more boxes, polygons, or quadric surfaces with bounding planes (e.g. finite length cylinders, ellipsoids, etc. linked together by logical .AND./.OR. statements). A sample model is shown in Figure 2-7.

Each component is assigned a Pk/h value based upon one of three types of vulnerability. The three types are:

(1) Single fragment vulnerable

(2) Energy density vulnerable

(3) Area removal vulnerable

For the first type, the measure of vulnerability is the probability of component kill given a hit by a fragment (Pk/h). This is expressed as a constant term plus a linear function of fragment mass and of impact velocity in the form:

Pk/h = PK(1) + PK(2) * M + PK(3) * V

where PK(1) is a constant term
PK(2) is the coefficient of mass
PK(3) is the coefficient of velocity
M is the fragment mass in grains
V is the fragment velocity in feet per second

For the second type of vulnerability, the energy density kill is expressed in terms of a required minimum area exposed to a threshold energy density level with a limiting fragment mass below which no computations are made. This type of kill probability is generally applicable to target structural members, whereas single fragment vulnerability is commonly used for components.

For the third type of vulnerability, the measure of an area removed kill is defined by a minimum area removed, below which no damage occurs, and an area which, if removed,

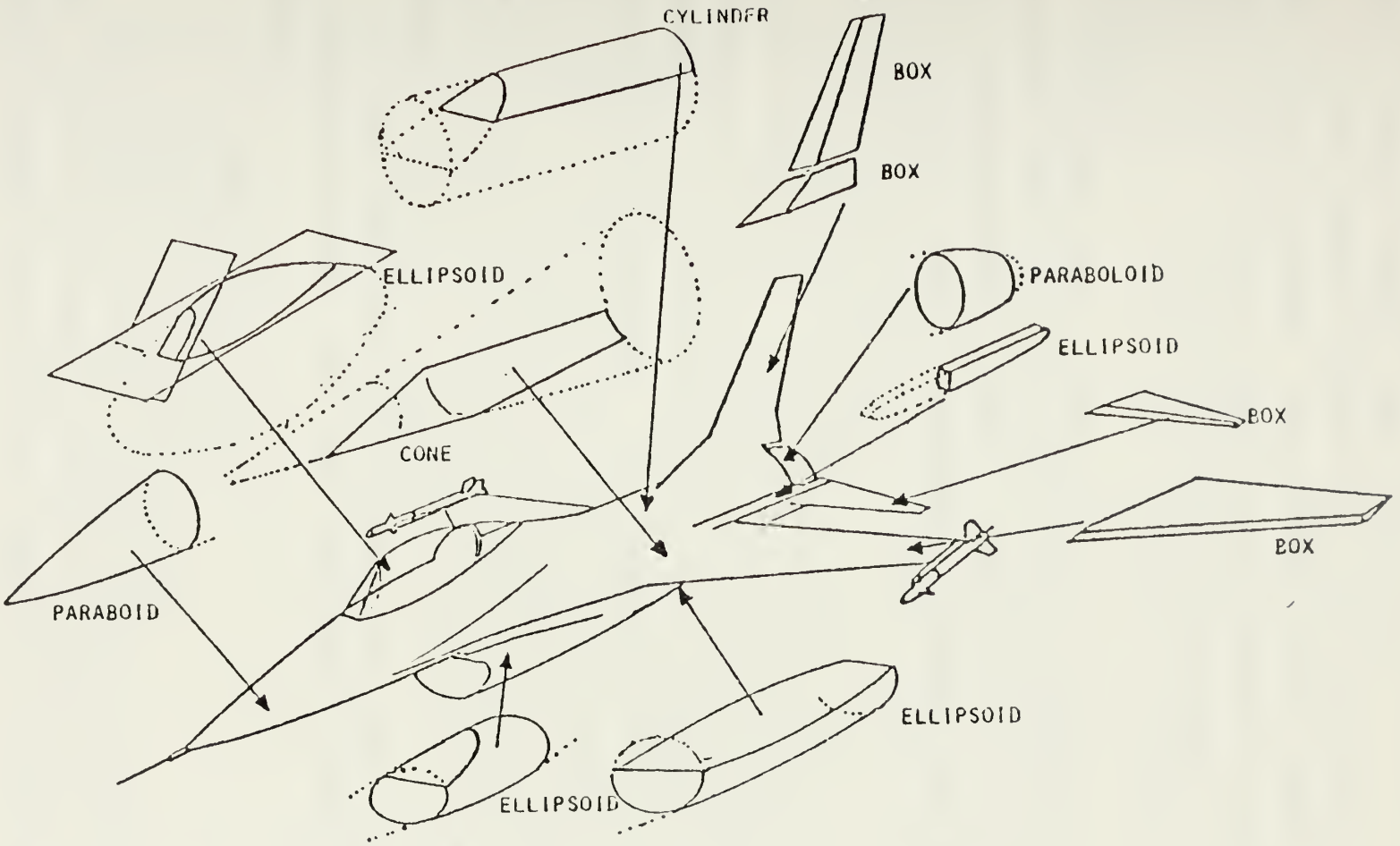


Fig. 2-7 Geometric Representation of the Exterior of the F-16 Aircraft

will cause complete failure. The kill probability is computed linearly between these two values.

For each target component the user must specify a material type and thickness. The material type is chosen from a list of ten options provided by the program and specified in Table II-2. A component surface is designated as solid or hollow and as either an internal or external aircraft component.

Each component's vulnerability and susceptability type is chosen from a list of eleven options shown in Table II-3. It is also possible to define a component to be non-vulnerable to specific damage mechanisms. Specific components may be designated as infrared (IR) sources allowing the simulation of IR fuzing mechanisms.

The killing of an individual component may or may not cause a target kill. Consequently, aircraft subsystems can be defined by linking components with logical .AND./.OR. statements, and aircraft systems can be composed of previously defined subsystems. The components are identified by the order in which they were input for the geometric representation. This feature of the program can be used to define multiply vulnerable or redundant components. Various levels of target kill can then be specified in terms of components, subsystems and systems.

The SCAN blast model and warhead model are both similar to the ATTACK model. The SCAN fuzing model has only three options:

(1) Instantaneous detection

(2) Infrared (IR) fuzing

(3) Single look-angle active fuze

The program has three possible scenario choices available. In one, the user may define a trajectory by fixing the initial missile range from the target and the orientation of

TABLE II-2

LIST OF MATERIAL TYPES

Material	Input Code
Magnesium	10
Aluminum 2024T	20
Titanium Alloy	30
Face Hardened Steel	40
Mild Steel	50
Hardened Steel	60
Lexan	70
Plexiglass	80
Doron	90
Bullet Resistant Glass	100

TABLE II-3

VULNERABILITY/SUSCEPTABILITY OPTIONS

Option number	Option Description
1	Energy density vulnerable
2	Single fragment vulnerable
3	Area removal vulnerable
4	Nonvulnerable to fragments, direct-hit vulnerable
5	Nonvulnerable to fragments and direct hit
6	IR source and nonvulnerable to fragments
7	Energy density vulnerable, invisible to EM fuze
8	Single frgament vulnerable, invisible to EM fuze
9	Area removal vulnerable, invisible to EM fuze
10	Nonvulnerable to fragments, invisible to EM fuze
11	Nonvulnerable, invisible to EM fuze

the missile to the target. The orientation is established by an elevation angle, azimuth angle, angle of attack, and sideslip angle for the missile and by roll, pitch, yaw, sideslip and attack angles for the target.

Another method requires the user to input a miss distance. This miss distance is used as an offset to the missile aimpoint. It will be the closest point of approach of the missile to the specified aimpoint without fuzing consideration. The numerical value selected for the miss distance will be dependent upon the missile guidance system being simulated. The missile and target are oriented in the same manner as for the fixed trajectory. The program determines the trajectory required to get the missile to the theoretical CPA with the specified orientation. This CPA is theoretical because it is possible that the warhead will detonate prior to this point, depending on the type of fuzing logic chosen.

The third option involves the input of a circular error probable (CEP) rather than a specified miss distance. The CEP is a statistical quantity which represents the radius of a sphere inside of which one half (50%) of the missile miss distances will occur. The trajectory used in the computation is obtained from a normally distributed sample. All other parameters are identical to those in the specified miss distance option.

Multiple missile trajectories are possible for each specified geometry. The user may also utilize the statistical capability of the program by providing standard deviation information for the missile elevation angle, azimuth angle and/or angle of attack.

The SCAN model utilizes the geometric model of the target and warhead detonation to determine the number of fragments which will impact the target. The program divides

the warhead polar and radial zones into a number of elements containing fragments of the same class which are all travelling in approximately the same direction. A representative ray is generated to characterize the fragment of each element, and the motion of this characteristic fragment is simulated along a trajectory. This procedure can be very time consuming when the number of fragments is large or when the target is complex. In order to reduce the computational requirements, the user must provide limiting spatial parameters. These parameters are dependent upon the physical dimensions of the target. Limits are established at values which slightly exceed the target dimensions. Outside of the limiting values no fragment computations take place.

SCAN has two graphic display programs available for use with the simulation. Several facilities, including the Naval Postgraduate School, have installed the capabilities. The SPLGEN program is a preprocessor which will check all of the geometric target model information for proper type and limits, then display the target model on a Tektronics 4081 graphics display terminal. The SPDRAW program uses fragment impact data generated by a SCAN simulation to display the hits on a displayed target model.

D. SESTEM II

SESTEM II is a digital computer program developed by the Aeronautical Systems Division - Deputy for Development Planning (ASD/XRHD), Wright - Patterson Air Force Base [Ref. 3]. The documentation was published in May 1977. The objective of the SESTEM II Endgame simulation, as defined in the model description, is "to evaluate the terminal effectiveness of missiles with nonnuclear warheads against U.S. and foreign aerial targets." The terminal effectiveness is measured in terms of the terminal encounter single shot probability of kill PK. The program is designed to evaluate missiles with

blast - fragment warheads. A PK is computed for three cases:

(1) Direct hit
(2) Fragment damage
(3) Blast

This program has been utilized for:

(1) Preliminary warhead design and fuze optimization
(2) Formulating tactics and countermeasures requirements
(3) Computerized air-to-air duel simulations
(4) Reconstructing and analyzing selected Southeast Asian combat incidents
(5) Evaluating existing and conceptual aircraft in support of an Offensive Air Support Mission Analysis

The program requires three general types of input data:

(1) Encounter data
    (a) terminal geometry (see Fig. 2-8)
    (b) missile aimpoint
    (c) target and missile encounter altitude
(2) Missile warhead and Fuze data
    (a) circular error probable (CEP), miss distance, or iso-Pk conditions
    (b) fuzing equations
    (c) fuze delay time
    (d) fragment sprayband and fragment density
    (e) fragment average mass and initial velocity
    (f) fragment cross-sectional area and coefficient of drag
(3) Target data
    (a) component size and location
    (b) individual component fragment vulnerable areas
    (c) external blast kill contours

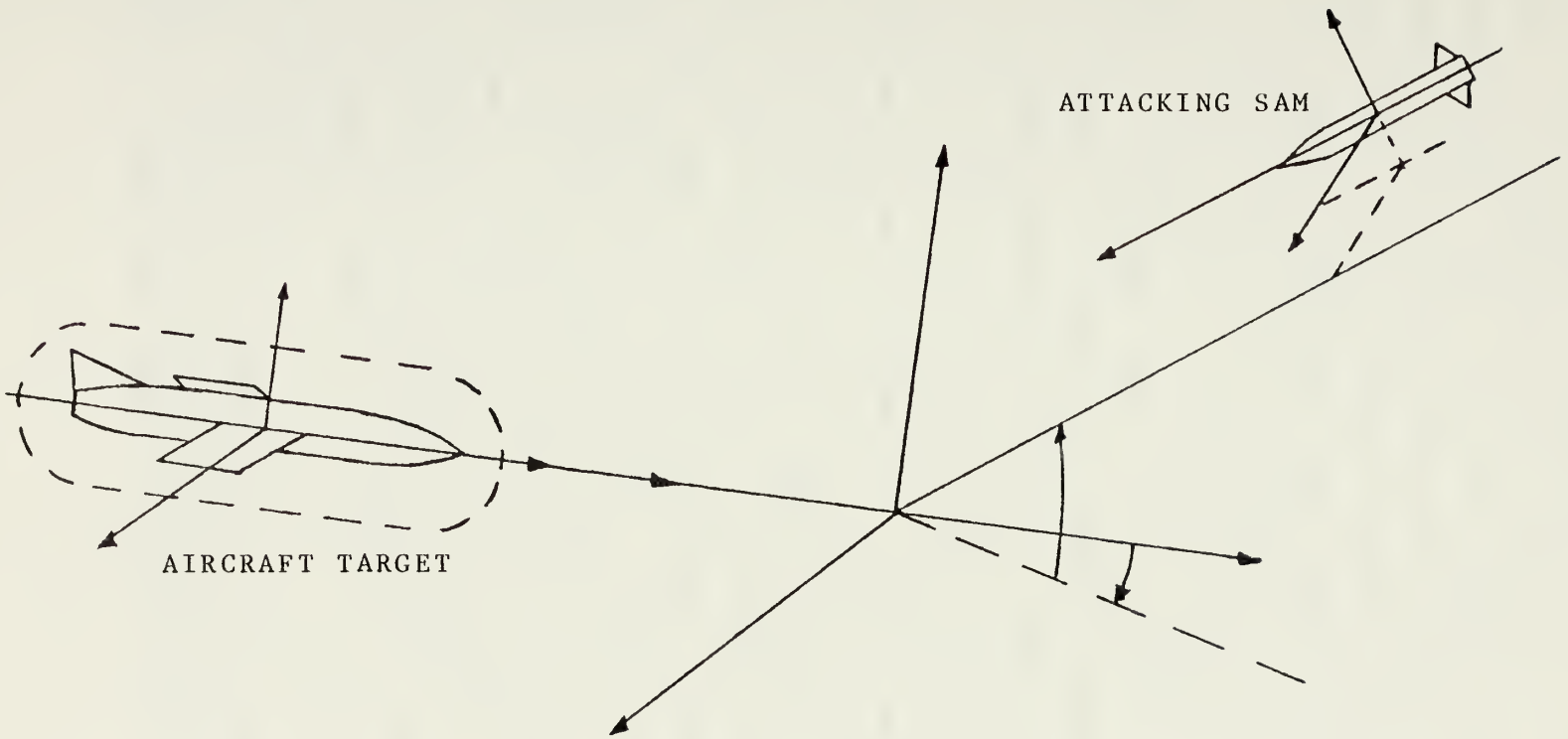


Fig. 2-8 SESTEM fI Terminal Geometry

The target is simulated in the computer as a collection of shapes, to be discussed later, representing fragment vulnerable, masking, or fuzing components. Each fragment vulnerable component has assigned it's own appropiate table of fragment vulnerable areas as a function of aspect angle and impact velocity just as in the ATTACK program. The program constructs an external blast kill contour for the target, missile and altitude being evaluated using the input data. The program represents each component by means of a grid of variable side length inscribed on the component surface. A "target point" is generated in the center of each grid square, represented by direction cosines and X, Y, Z coordinates. The target point is then used to represent that grid square in fragment interaction computations. The missile warhead parameters and fuzing equations are simulated in the program using the static input data. Dynamic resolution of the static warhead data is done by the program. Various types of fuzes, e.g. radar, contact, proximity, may be simulated. Both the target and the missile are assumed to be flying constant speed, straight-line trajectories during the terminal phase. The missile and target approach each other along their relative trajectories until the fuzing equations are satisfied. After the appropiate delay time, warhead detonation occurs and the dynamic interaction with the vulnerable components is computed. The probabilities of killing the target by blast, direct hit, and each component by fragments, are computed and combined to predict the probability of target kill. Fuzing effects may be examined by selecting as many as eleven fuzing points for each trajectory.

Trajectories may be generated by various methods with SESTEM II. Parallel trajectories may be generated randomly by assuming a bivariant normal or other type of expected

distributions. Sequential groups of up to thirty trajectories may be generated and averaged. Discrete trajectories may also be input and the PK computed for each case. Average PK values as a function of miss distance can be calculated by specifying various miss distances. The program locates trajectories on a circle of radius equal to the specified miss distance, calculates the individual trajectory PK values and the average PK for this miss distance.

The program can also be used to generate "iso-PK" contours. In this mode the program computes the blast kill boundary (PK = 1), the no fuzing boundary (PK = 0) and PK for varying distances between these two values.

The target is modeled using ellipses and parallelopipeds. A fuselage or fuselage-shaped component with a cross section parallel to the target Y-Z plane is represented by the upper and lower halves of two ellipses with common minor axes. The center of the ellipses may be displaced an arbitrary distance along a line parallel to the target Z-axis. Each component may be represented by using up to thirty cross sections (see Fig. 2-9). The equations defining the surface of the component and the direction cosines of any target point on the surface are generated by the program as required. A wing or wing-like component with a cross-section parallel to the X-Z plane of the target is represented by ellipses as shown in Figure 2-10. The vertical stabilizer or similarly configured components with cross-sections parallel to the X-Y plane are represented by ellipses also. The program will compute the direction cosines for a point on the surface of these structures as well as the equations for the surfaces generated. The program has the capability to represent components such as fuel tanks or electronics as rectangular parallelopipeds. The parallelopipeds are located by specifying the corner

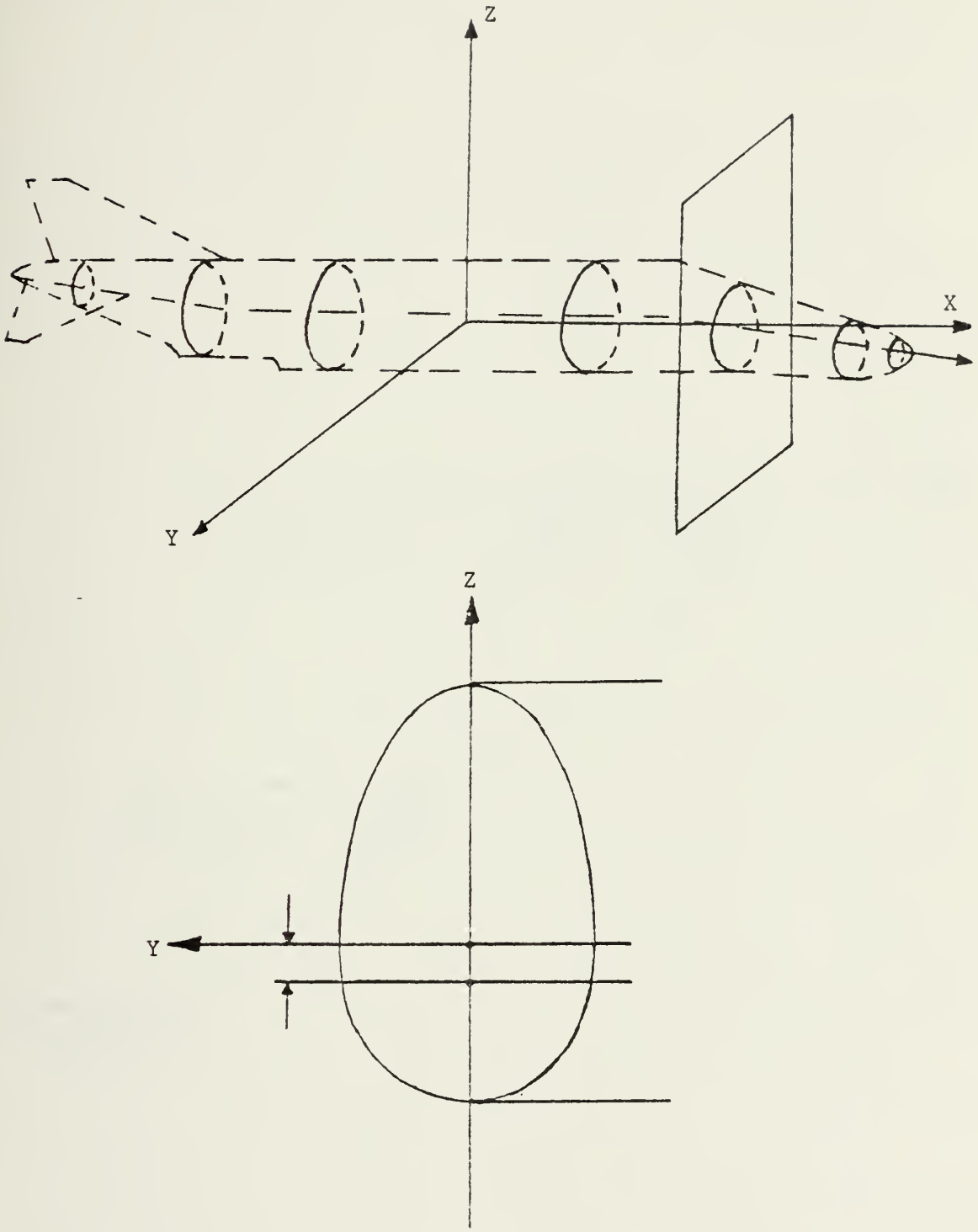


Fig. 2-9 SESTEM II Fuselage Model

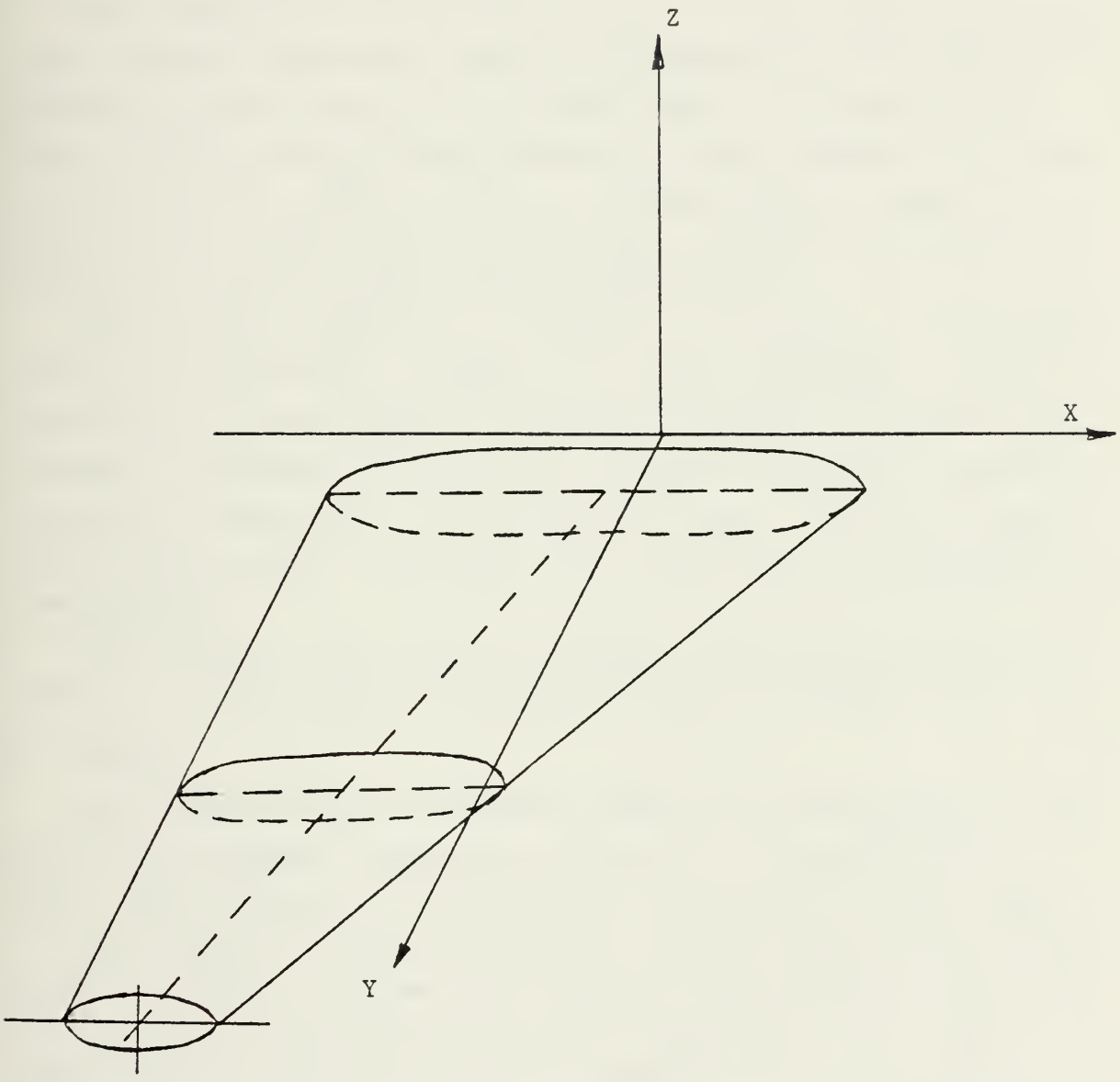


Fig. 2-10 SESTEM II Wing Model

with the lowest target coordinate values and X, Y, Z displacement values. The program has the capability to generate "mirror image" components by reversing the Y-values. This allows easy representation of axisymmetric components using only one side as input.

The expanding blast wave from the warhead detonation can cause target destruction from the overpressure and dynamic effects on the structure. A blast contour encloses that volume within which warhead detonation will result in a target kill. This blast contour is represented as shown in Figure 2-11. The input for the blast contour is calculated external to the program from available data.

Each target component, as input, is assigned a classification of vulnerable, masking, or fuzing component, or any combination of the three. For those components specified as fragment vulnerable for the kill category desired, tables of vulnerable areas must be provided as input. Only 36 components may be used to represent the target due to computing limitations.

The warhead has a single fragment sprayband bounded by forward and aft limiting angles as shown in Figure 2-12.

E. SHAZAM

The SHAZAM digital computer program was developed at the Air Force Armament Laboratory, Eglin Air Force Base, in 1976. The program is not formally documented at this time.

The SHAZAM simulation program is best described as "free form." The main program is a core of "bookkeeping" routines which compile information as it is generated. All other functions of the program are user designed. The program is structured to have the user design specific subroutines which fit the needs of the simulation being carried out. The Pk/h function is user defined and not contained within the body of the program.

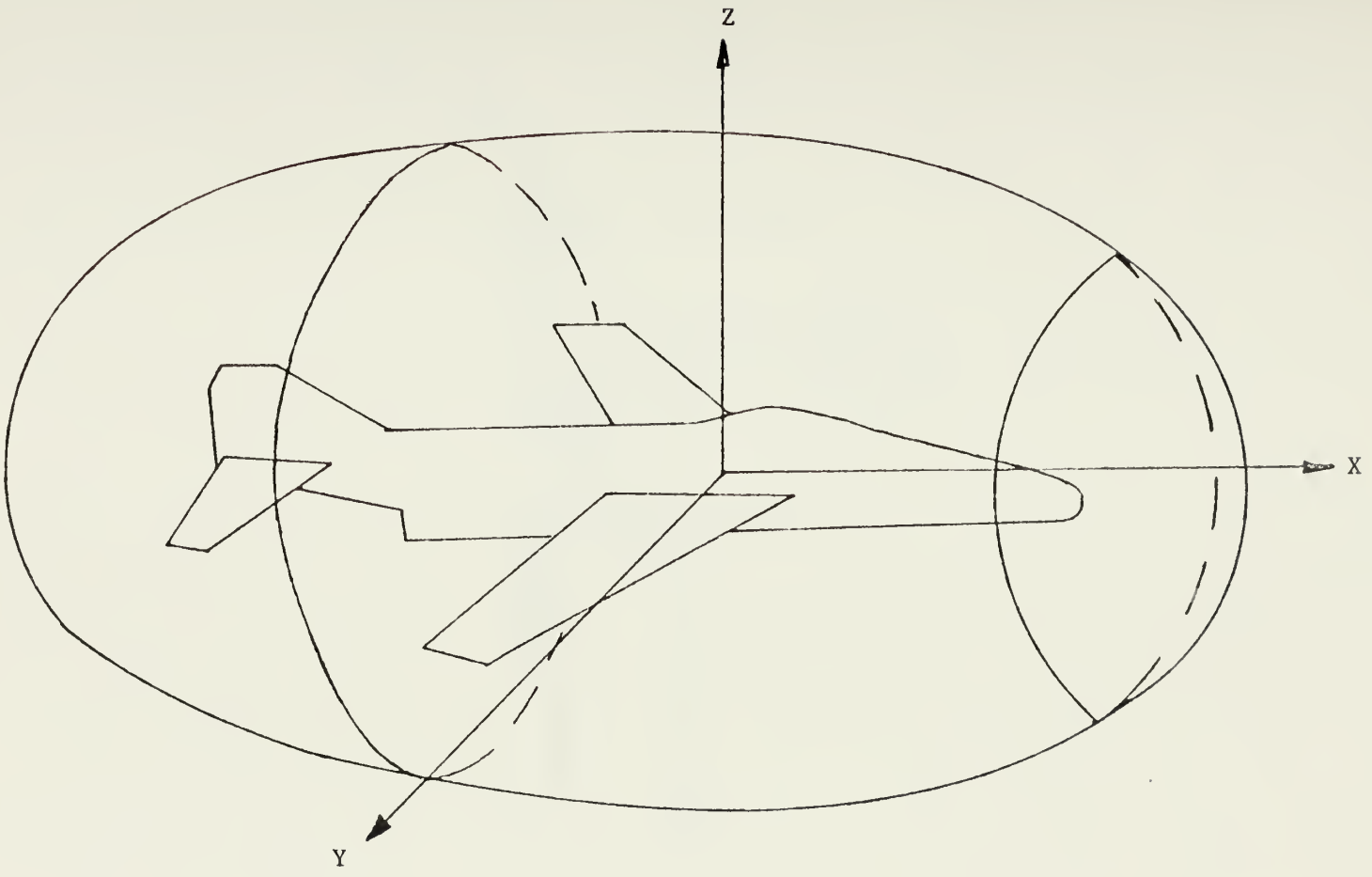


Fig. 2-11 SESTEM II Blast Contour

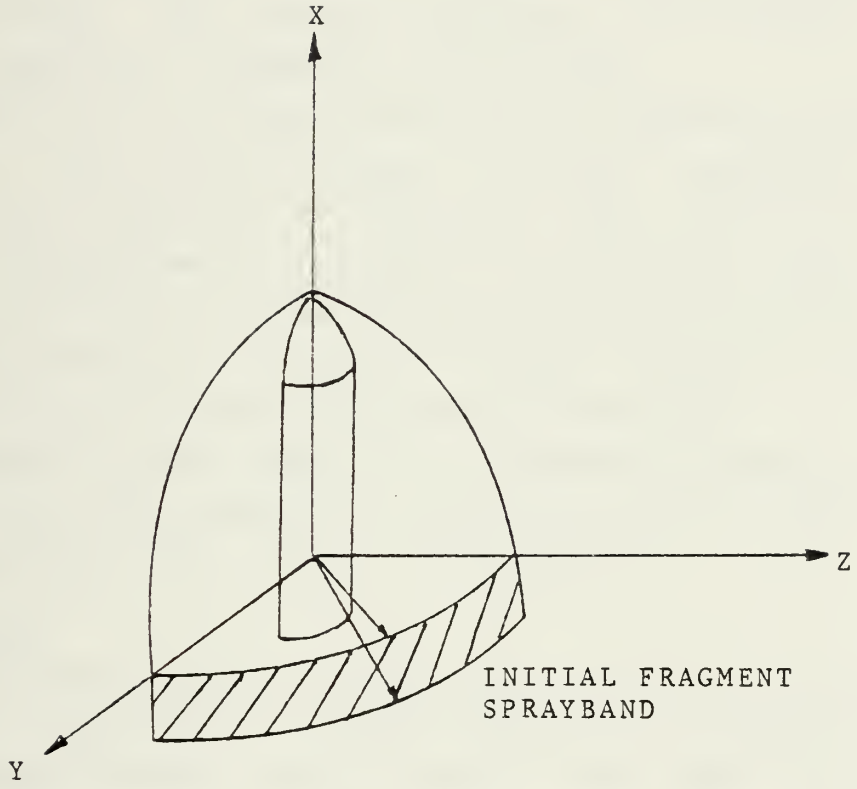


Fig. 2-12 SESTEM II Warhead Model

SHAZAM utilizes component Pk/h values to compute target vulnerability to fragments and uses a ray trace method to determine fragment paths from detonation.

The SHAZAM program uses a target model of a fixed format. The exterior surface of the target is a finite element model composed of polygons. This allows SHAZAM to utilize the target models generated by the FASTGEN or SHOTGEN computer programs. The interior components are modeled using lines, spheres, cylinders, and polygons.

Blast can be handled statically or dynamically by SHAZAM. Each external polygon can have a seperate blast kill radius specified to accurately model blast vulnerability. The program is capable of adjusting the shape of the blast contours with time for dynamic blast modeling.

The program can combine the blast, fragment, and direct hit Pk values to obtain an overall target Pk.

SHAZAM has the capability to generate graphics and utilize interactive graphics. This is an added option that is not currently part of the "core" program.

F. REFMOD

The REFMOD digital computer program is a reference model used for computing the effectiveness of externally detonating weapons against moving targets. The model was developed under the auspices of the Joint Technical Group for Munitions Effectiveness (JTCG/ME), Anti-Air Missile Evaluation group. The first version, REFMOD-1, had documentation completed in December 1979. The latest version, REFMOD-2, has preliminary documentation dated March 1981 [Ref. 4]. REFMOD-2 will be the version described in this section.

REFMOD has been assembled by incorporating methodologies from other existing Endgame simulations, such as AMEGS, ATTACK, SCAN, SHAZAM, and WHDEVAL. In order to combine features from these programs, it was necessary to modify them

to provide consistant nomenclature and coordinate systems for the resultant program.

REFMOD also includes some significant additional features which enable it to work with a wide variety of vulnerability models and allows it to evaluate warhead/target combinations that were previously too cumbersome to assess. The fuzing routine utilized will allow greater flexibility in fuze modeling.

The program is currently being rewritten in Standard FORTRAN using a structured programming format. This will result in a program usable at any facility with Standard FORTRAN capability and enough computing power.

The REFMOD program has a high degree of flexability. Three different modes of encounter input are available. The fuzing can either be performed by the program or determined externally, and any convential warhead type can be modeled. Three shape options are available for contact hit target modeling. There are three options for blast kill computations, and several types of fragment damage can be modeled.

The types of studies that REFMOD can be used for are:

(1) To evaluate the operational effectiveness of existing and proposed missile systems.

(2) To assist in fuze optimization and selection of warhead design.

(3) To assess aircraft survivability and countermeasure effectiveness.

The type of target model required as input data depends upon the damage mechanism specified in the vulnerability model. The direct hit model utilizes the physical shape of the target to determine if the missile contacts the target. The target model can be described by any combination of the three shapes listed:

(1) Truncated elliptical cone

(2) Polygon

(3) Ellipsoid

Figure 2-13 gives examples of direct hit modeling. A full range of target Pk values from zero to one can be specified for a contact or debris kill.

The evaluation of blast effects can use three different model types for evaluation. The three models are:

(1) The target blast model is represented by hemispherically capped cylinders. A kill is recorded according to whether the warhead detonation occurs inside at least one of the blast cylinders (see Fig. 2-14).

(2) The target is modeled using blast ellipsoids and a kill is recorded according to whether the warhead detonation occurs inside at least one of the blast ellipsoids.

(3) The target is modeled by blast ellipsoids and blast damage centers. Data on blast pressure and impulse are required to determine blast kills. This model provides for non-spherical blast representation.

The area removal model for fragment kills consists of a representation of individual structural members considered vulnerable to fragmentation effects. The probability of removal of a specified amount of material in a "contigious" path across a structural member is determined based upon fragment striking conditions, fragment spray angles, and other pertinent information. This value is used to determine the probability of structural member failure and is combined with the probabilities of failure of all other structural members to give a total kill probability. Figure 2-15 is an example of this model.

There are four other fragment vulnerable component types that can be used. Each component can use only one type for vulnerability modeling. The four models are:

(1) Vulnerable area

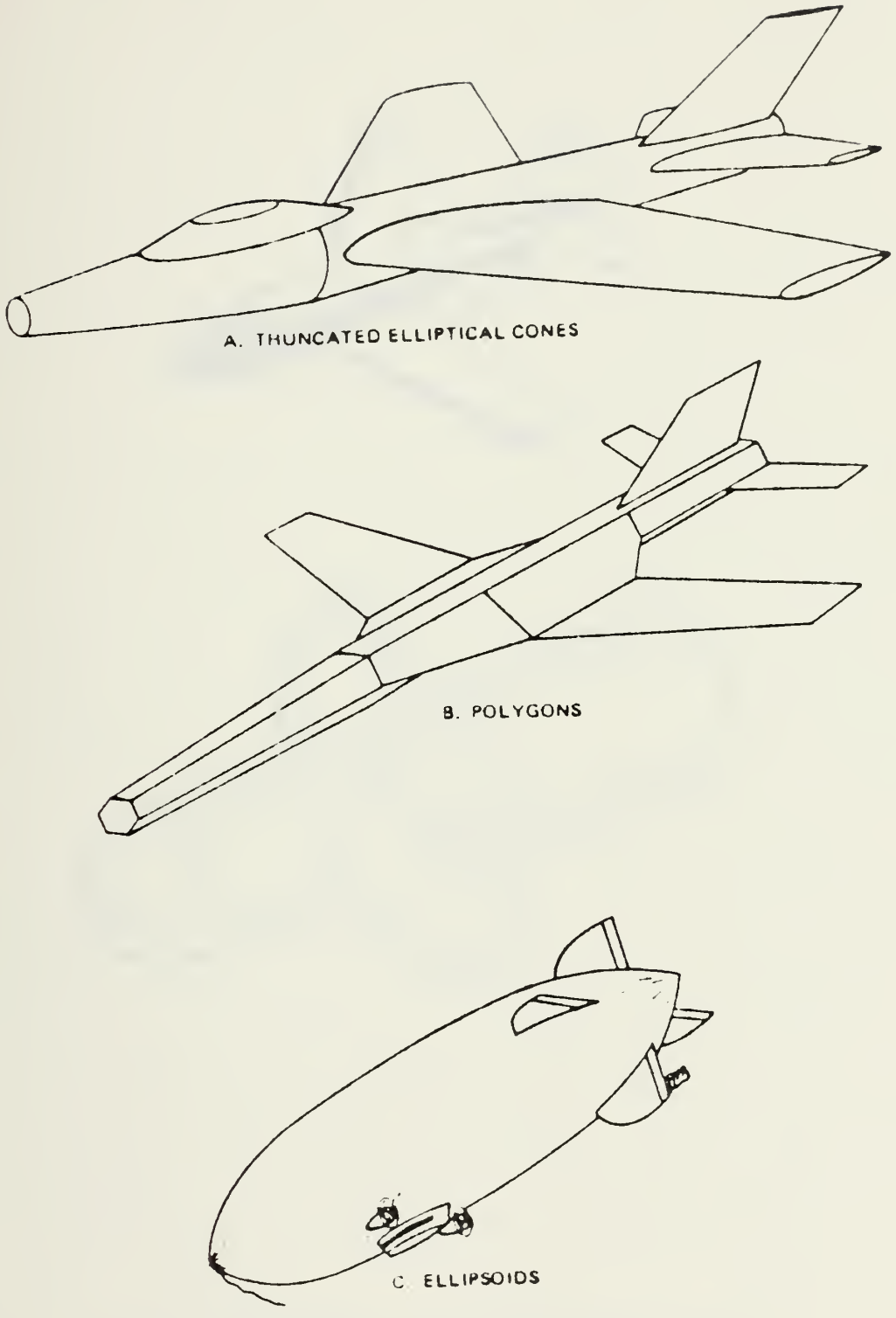


Fig. 2-13 REFMOD Direct Hit Modeling

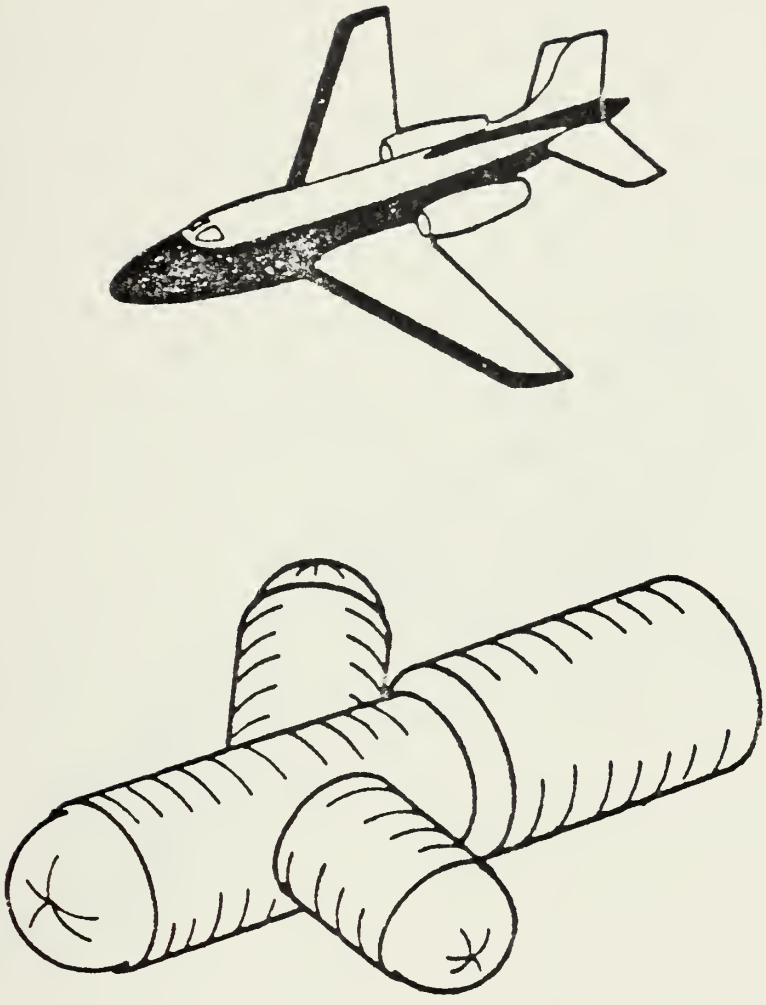


Fig. 2-14 REFMOD Blast Cylinder Model

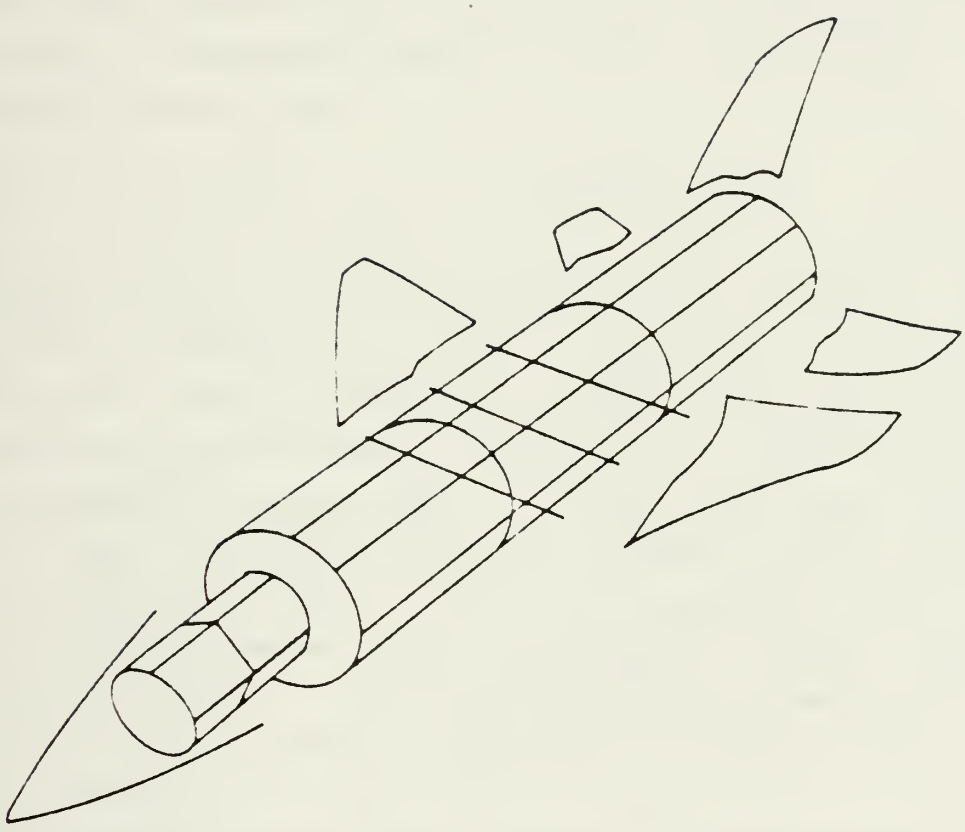


Fig. 2-15 REFMOD Structural Failure Model

(2) Function of mass/velocity/density

(3) Expected value

(4) Focused Fragment Controlled Motion (FFCM)

For the vulnerable area type, components can be described as being spherical, linear, cylindrical, or planar in shape (see Fig. 2-16). Each of these shapes is used with the traditional vulnerable area methodology, using the vulnerable area tables generated by the COVART computer program.

For the mass/velocity/density type the components are described as cylinders (see Fig 2-17). The component kill condition is given by:

F = A * (M ** B) * (V ** C) per unit area

Where A, B, and C are constants to be defined by the user, M is fragment mass, and V is fragment impact velocity. This vulnerability measure can vary as a function of the fragment impact angle, and is described by upper and lower threshold boundaries as illustrated in Figure 2-18. The conditional kill probability is zero when F is below the lower threshold value. When F is above the upper threshold level, the kill probability is set equal to one. The component PK is interpolated linearly between the two threshold values. Energy density (A = 0.5, B = 1, C = 2) and momentum density (A = 1, B = 1, C = 1) are specific examples of this type of vulnerability model.

The expected value model represents the components as cylinders or line segments that outline the components. The vulnerability varies as a function of impact angle. The vulnerability is described in terms of Pk/h for N fragments, where N varies from one to a maximum specified value. The number of hits is calculated using a Poission distribution. This model type can also be used to specify a lethal radius vulnerability criterion.

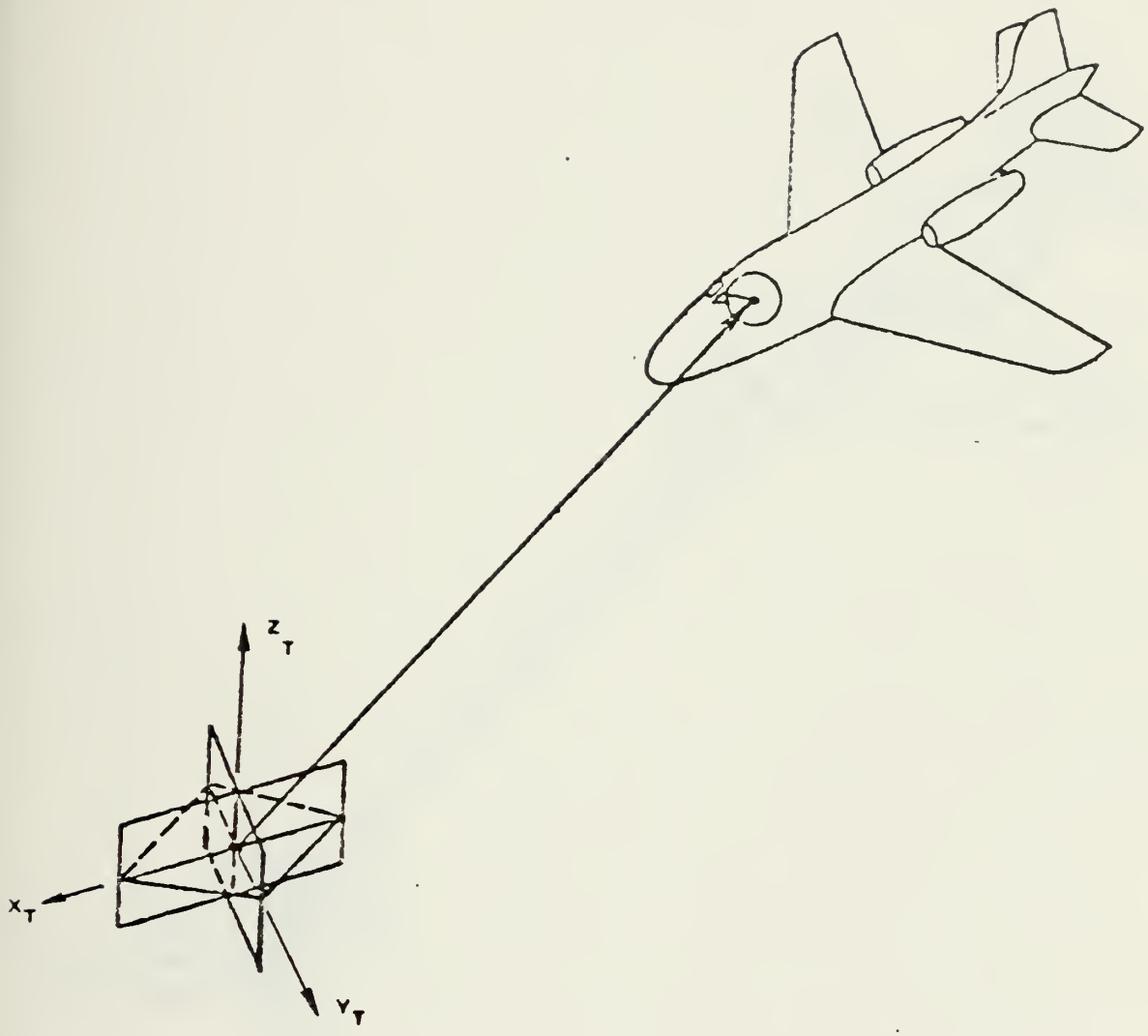


Fig. 2-16(a) REFMOD Spherical Vulnerable Component

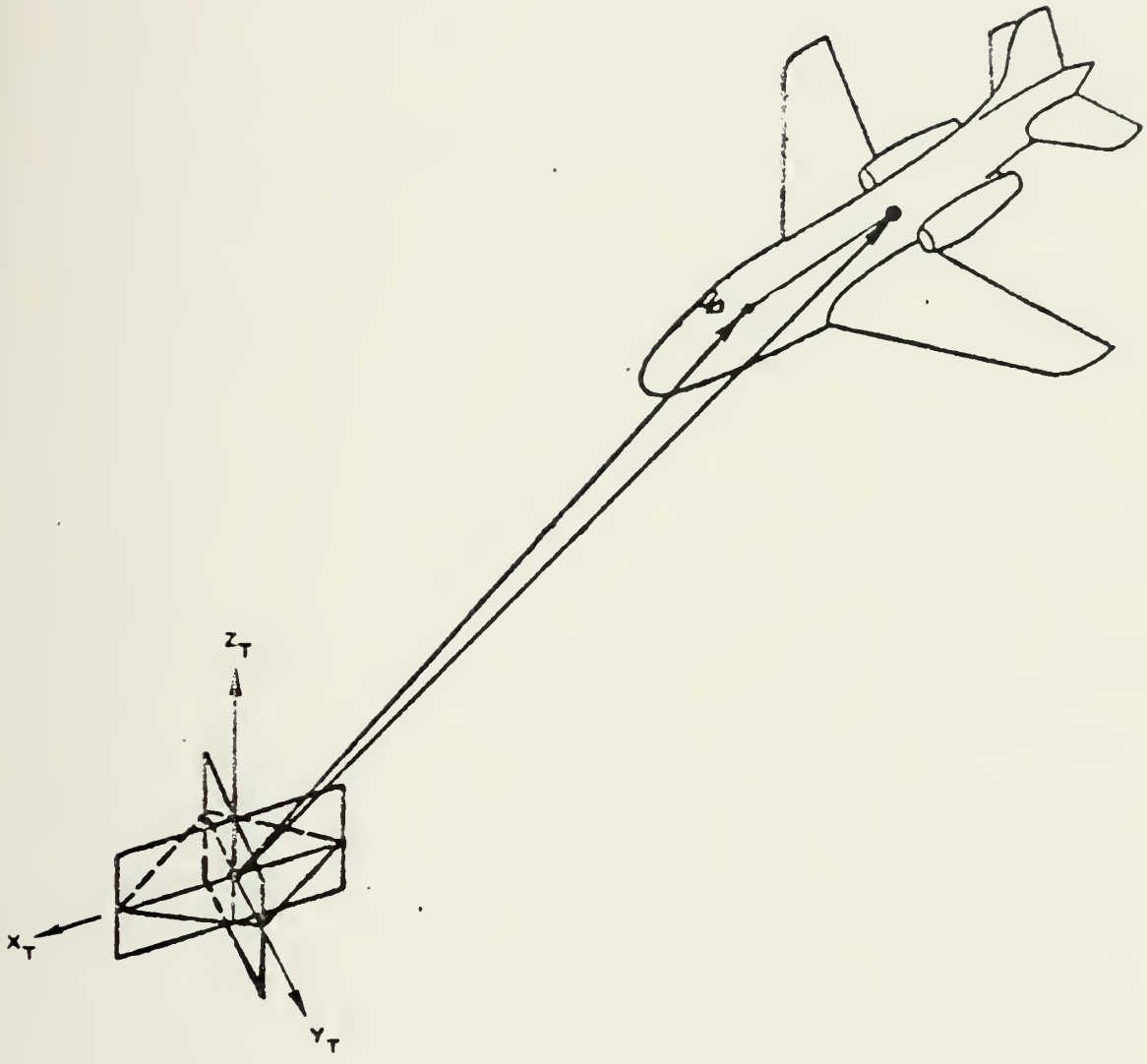


Fig. 2-16(b) REFMOD Linear Vulnerable Component

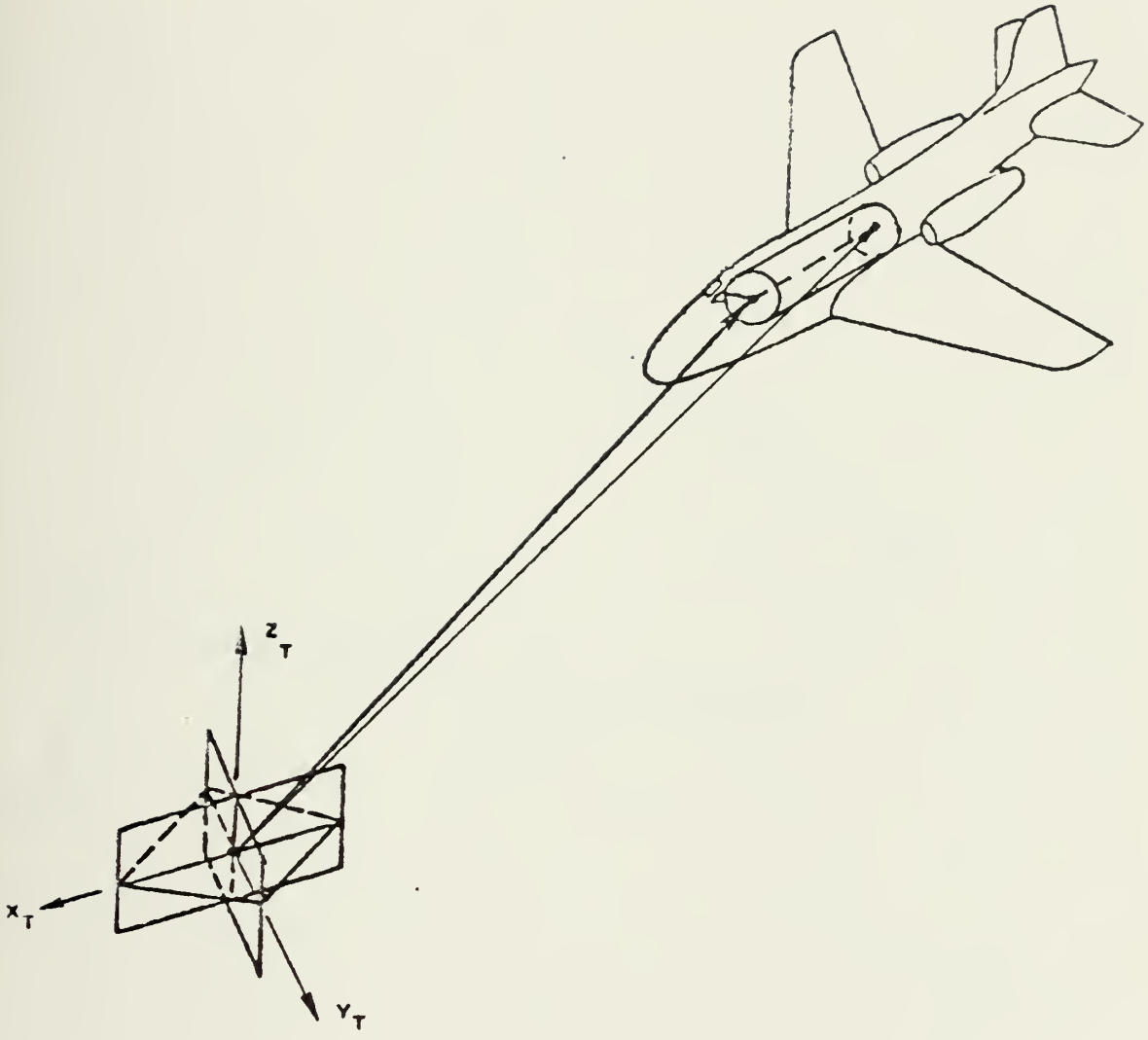


Fig. 2-16(c) REFMOD Cylindrical Vulnerable Component

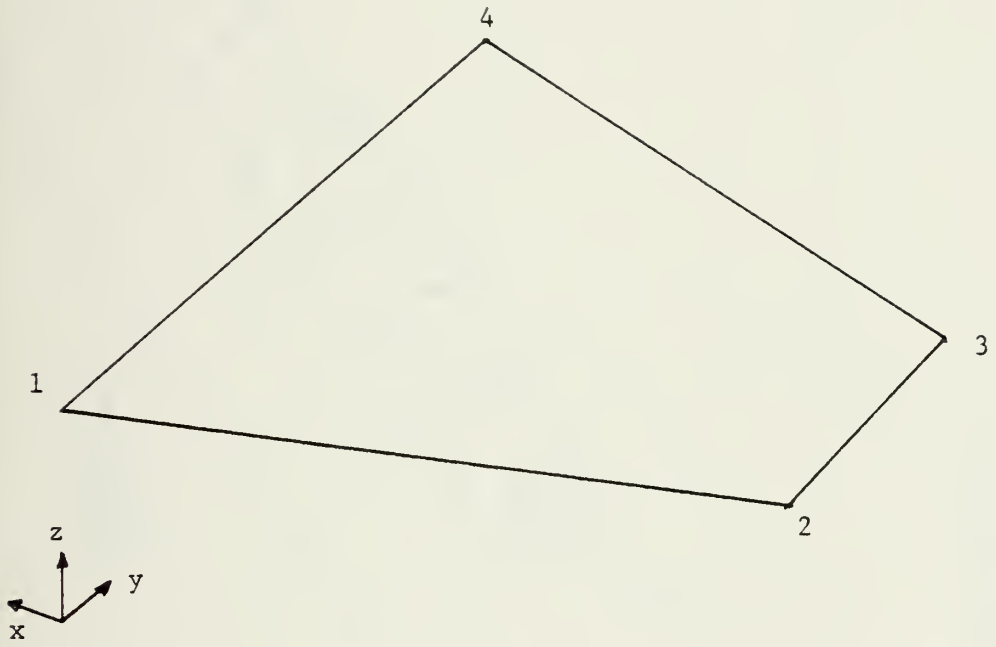


Fig. 2-16(d) REFMOD Planar Vulnerable Component

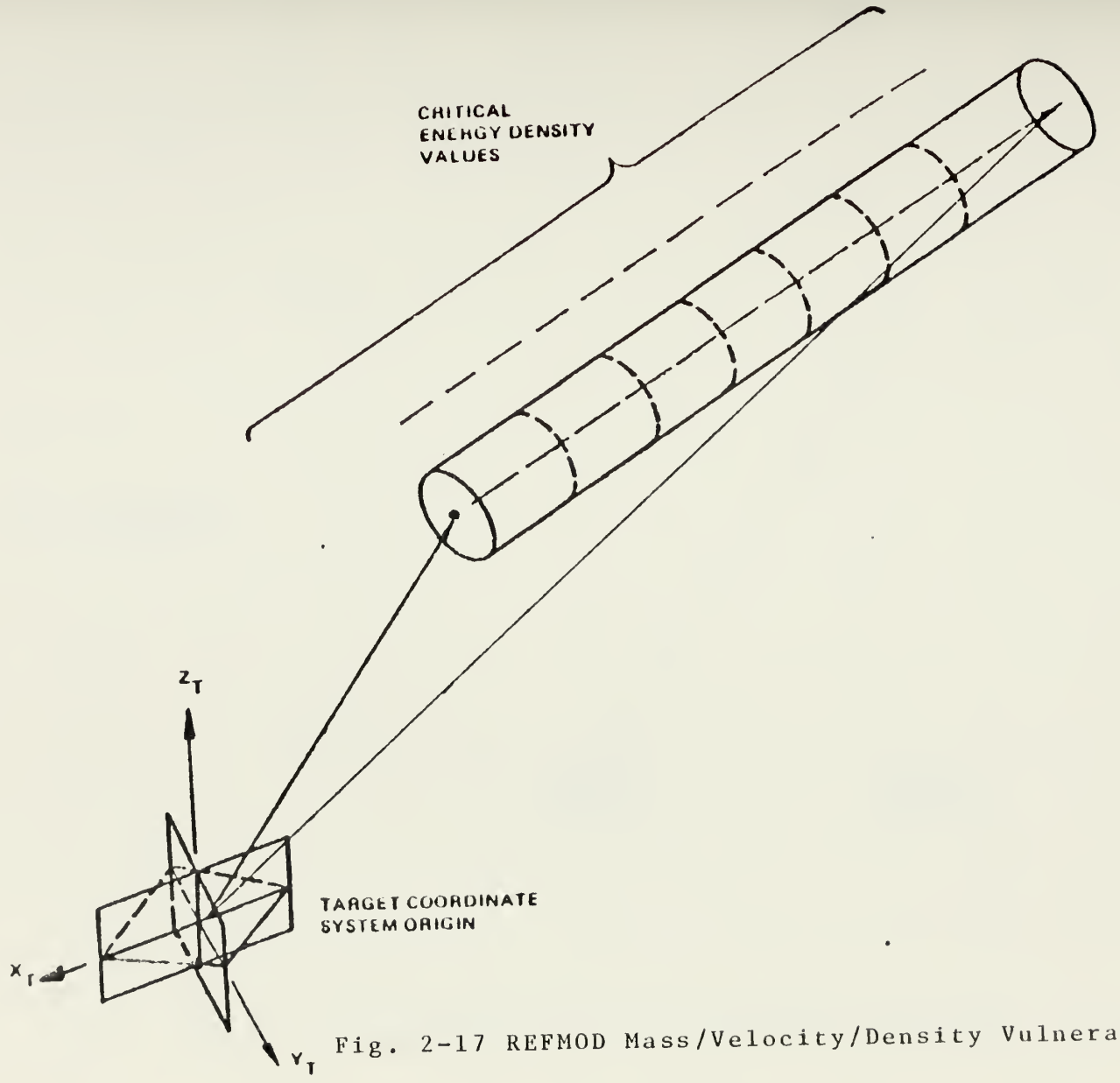


Fig. 2-17 REFMOD Mass/Velocity/Density Vulnerable Model

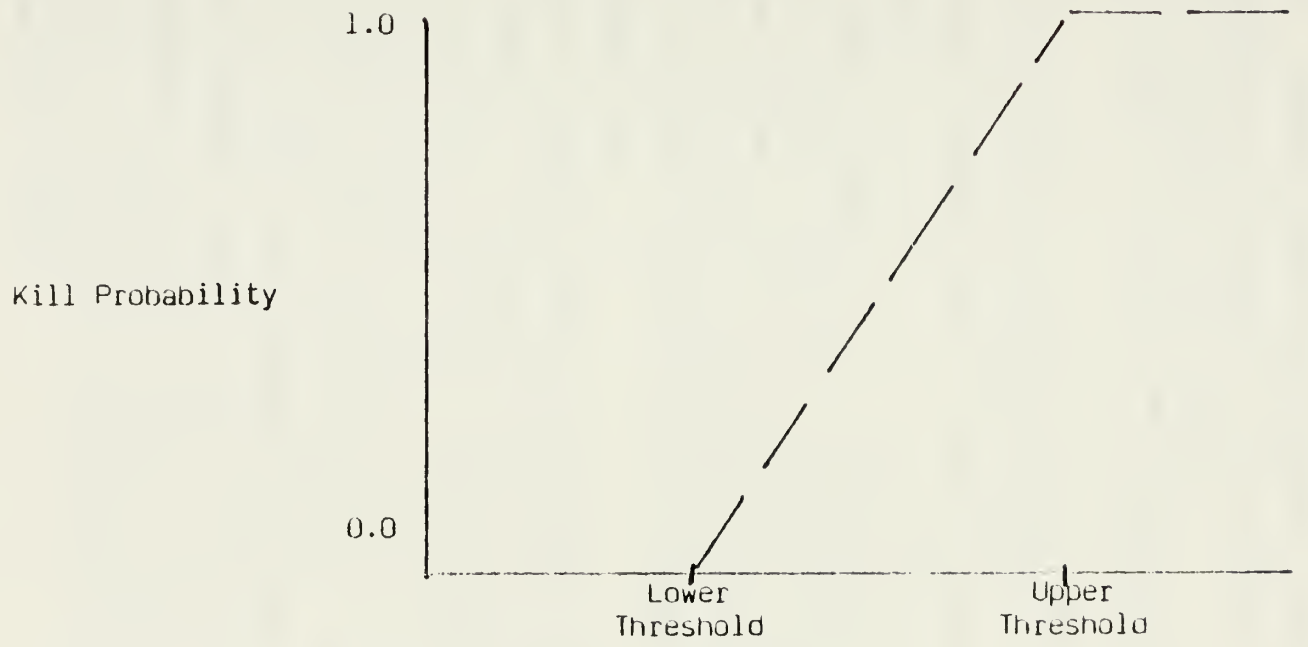


Fig. 2-18 REFMOD Mass/Velocity/Density Function

The Focused Fragment Controlled Motion (FFCM) model is the same as the expected value model except that the fragments are assumed to be uniformly spaced instead of uniformly distributed. The uniform spacing of the fragments is due to the use of large preformed fragments with predictable ejection angles. These preformed fragments maintain the same angular spacing with respect to the origin of the warhead detonation throughout their flight.

The REFMOD program is capable of handling the following warhead types:

(1) Continuous rods - the ends of the rods are welded together causing the rods to open in a hoop.

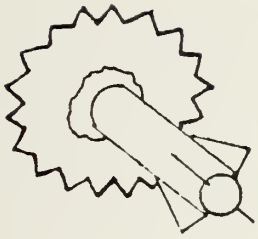
(2) Divergent fragmentation - most wide beam warheads are of this type.

(3) Convergent fragmentation - multiple point initiation causes the fragment patterns to cross over one another.

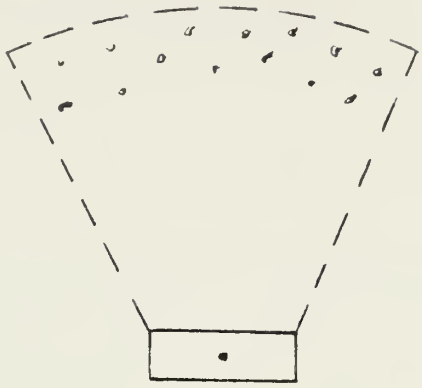
(4) Focused Fragment Controlled Motion - large preformed fragments with highly predictable ejection angles, resulting in a uniformly spaced pattern instead of a uniformly distributed one.

(5) Aimable - special cases of the above types in which the fragment density is non-uniform about the roll axis. Figure 2-19 illustrates the warhead types described above.

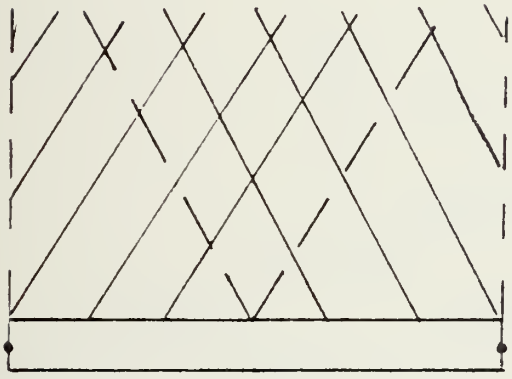
Several fuze routines have been revised and documented for use with REFMOD. These routines simulate the fuzes currently employed on existing missile systems. REFMOD is compatible with fuzing data obtained from one of these routines, from routines designed for conceptual fuzing, from fuzing data obtained from flight tests, or from any other source. The fuzing routine has a special target model made up of line segments representing surfaces that can be sensed by the fuze (see Fig. 2-20).



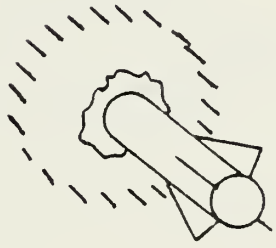
a. Continuous Rod



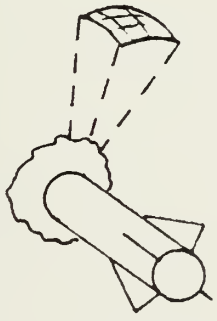
b. Divergent Fragments



c. Convergent Fragments



d. FFCM



e. Aimable

Fig. 2-19 REFMOD Warhead Types

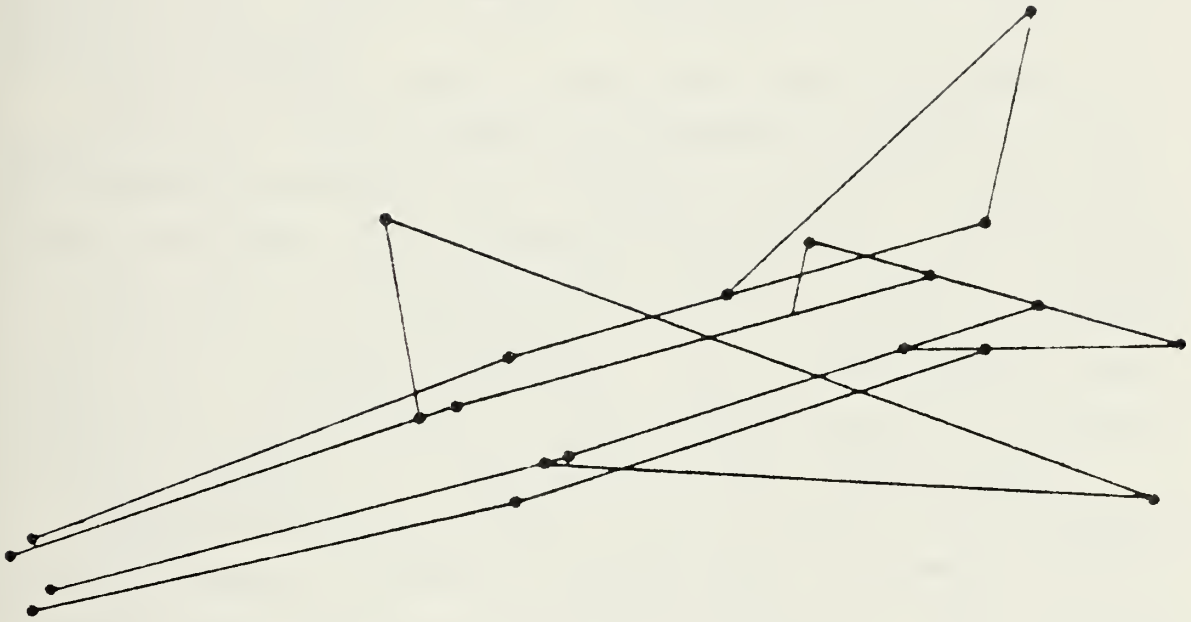


Fig. 2-20 REFMOD Fuzing Model

III. COMPARISON OF FEATURES AND TECHNIQUES

This chapter will compare the various techniques used in computer Endgame simulations.

A. GEOMETRIC MODELING; FINITE ELEMENT VERSUS COMBINATORIAL

The two types of target modeling commonly use in Endgame programs are finite element (polygons) or combinatorial geometry (COMGEOM).

Finite element modeling uses multi-sided polygons to construct the target and/or its components (see Fig. 3-1). The modeling accuracy can vary greatly, depending upon the types and number of polygons used or allowed. An advantage to this type of modeling is that there are computer programs available that are designed to generate this type of target model (e.g. FASTGEN or SHOTGEN). All of the programs discussed in this study use a finite element model for computing the results of at least one of their damage modes.

COMGEOM modeling involves the use of not only polygons, but also boxes, cylinders, hypreboloids, ellipsoids, and various conic sections with bounding planes. These complex shapes allow very exact modeling of a target and its components (see Fig. 3-2). However, the use of these complex geometric shapes adds a degree of difficulty to accurate target modeling. It appears that there are currently no computer programs designed to generate COMGEOM target models. One advantage to most programs that utilize COMGEOM models is that they can also accept finite element modeling as input. The SCAN program utilizes COMGEOM target models,

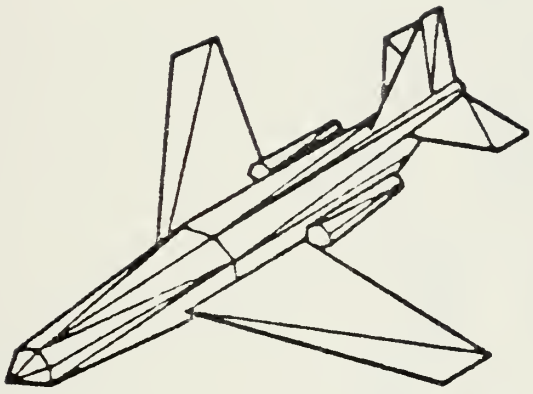


Fig. 3-1 Finite Element Model

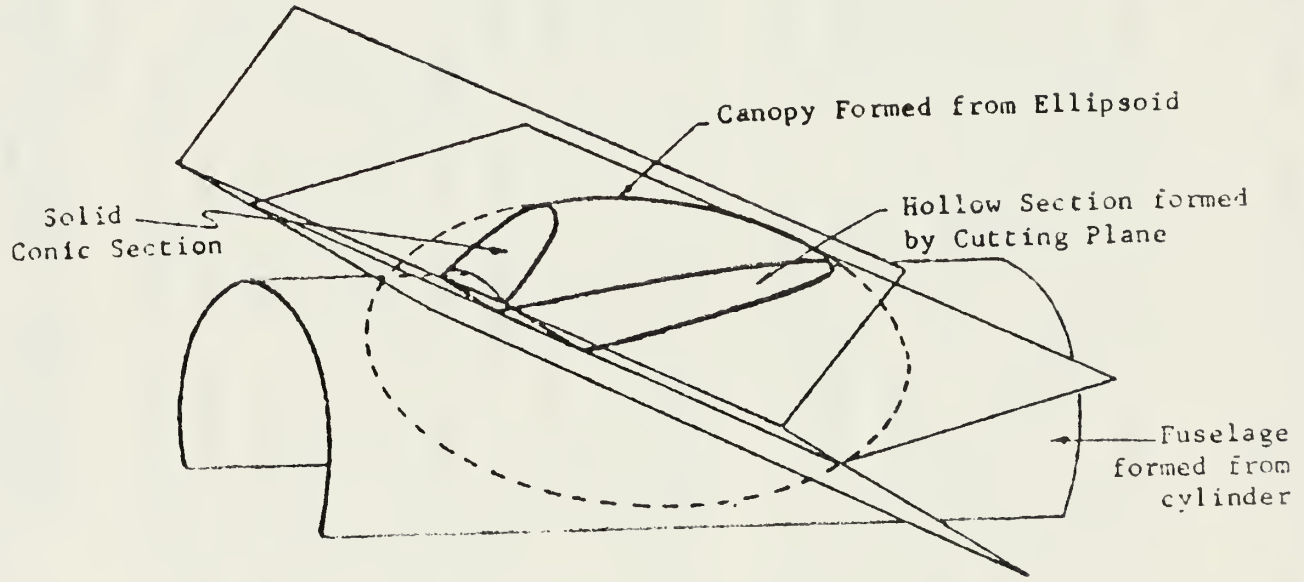


Fig. 3-2 COMGEOM Model

and both SESTEM II and SHAZAM utilize some of the features of COMGEOM target modeling.

B. VULNERABLE AREA VERSUS COMPONENT PROBABILITY OF KILL

The two techniques used in assessing fragment damage are vulnerable area and component probability of kill given a hit, Pk/h.

1. Vulnerable Area

The vulnerable area method utilizes vulnerable area tables for each fragment vulnerable component. Each component will have as many vulnerable area tables as there are fragment mass classes to be considered. The standard type of vulnerable area table has 26 entries per fragment mass class per component for eight velocities, based upon the 26 aspect angles shown in Figure 3-3 (for example, ATTACK requires 26 input data cards per component for one fragment mass class).

Components that are not fragment vulnerable, but act as masking or shielding for other components, must also have a full set of null vulnerable area tables in order to ascertain the amount of fragment slowdown or deflection.

The use of vulnerable area modeling of components simplifies the amount of computing required during program execution. The program computes the aspect angle of the fragments from the user input data, and then uses the input value of fragment mass to find the correct vulnerable area values from the tables input by the user. The programs usually use linear interpolation for aspect angles that fall between those tabulated. ATTACK, REFMOD and SESTEM II all use vulnerable area modeling.

2. Component Probability of Kill Given a Hit

The method of using component probability of kill given a hit, Pk/h, involves assigning each component modeled a set of Pk/h values. The values assigned to the component

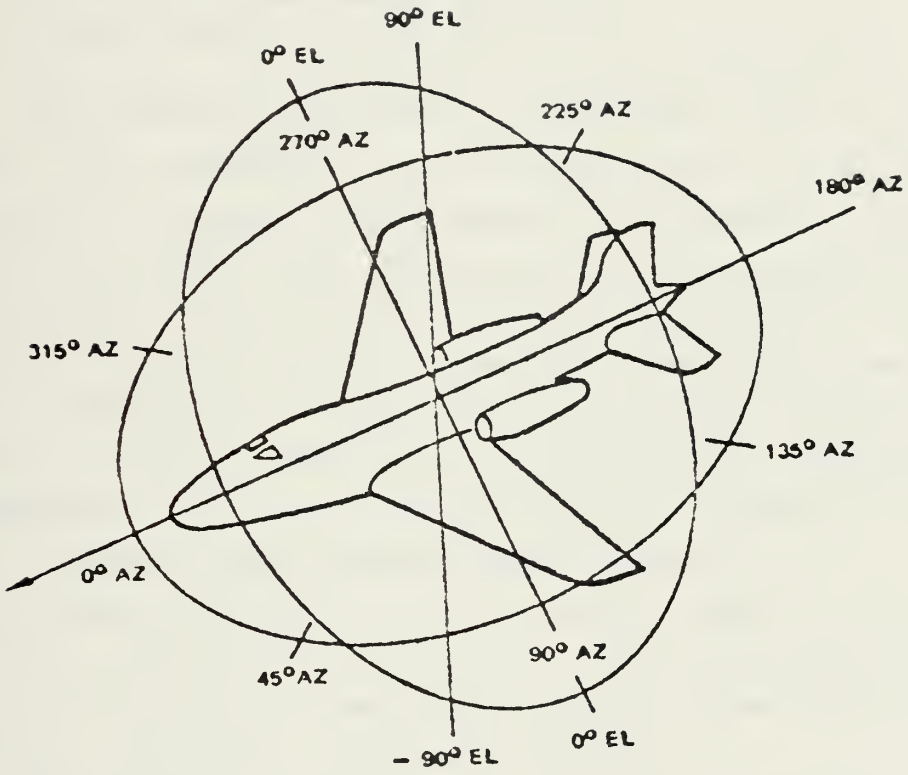


Fig. 3-3 Aspect Angles for Vulnerable Area

are usually in a functional relationship and are used to determine component kill levels. For example, SCAN uses a function of the form

$$Pk/h = PK(1) + PK(2) * M + PK(3) * V$$

where PK(1) is a constant term
PK(2) is the coefficient of mass
PK(3) is the coefficient of velocity
M is the fragment mass in grains
V is the fragment velocity in feet per second

The modeling required for component Pk/h is often complex. The components can be modeled with a high degree of accuracy to obtain valid results. The shapes used in modeling the target vary depending upon the program being utilized, but the SCAN program discussed earlier is a good example of how many possible geometric shapes can be used. The Pk/h assigned to a component has no aspect dependency that has to be tabulated as in vulnerable area modeling. The component Pk/h type of Endgame program does all the computing of aspect angle corrections internally.

A positive feature of the complex model is that it can serve a multitude of functions. The same geometric model can be used to assess other damage types such as direct hit, structural, and blast damage.

This method of modeling gives more flexability to the user. The Pk/h functional relationship for component kill assessment can be easily changed and the modification of individual components does not require an outside program or creation of a new input table of data.

Programs using component Pk/h modeling are capable of evaluating the effect of spallation. Since each component can have a seperate material type and thickness specified, it is possible to generate fragments of various

materials, sizes, and velocities from a single warhead fragment.

The amount of computing time required for this type of modeling is often greater than that required for a vulnerable area model; however, by limiting the spatial volume being considered by the program to the immediate vicinity of the target (thus not computing the paths of non-striking fragments), the computing time can be reduced substantially.

The difficulty level of constructing targets manually can vary greatly depending upon the target complexity and the geometric shapes used in modeling.

C. WARHEADS AND FUZING

All of the warhead models assume definable polar and radial zones of fragment ejection. The number of zones used and the number of fragment mass classes that may be contained within any one zone differs in the various Endgame programs. Only REFMOD permits the modeling of continuous rod and Focused Fragment Controlled Motion (FFCM) warheads. The SCAN warhead model is the only one that allows fragments to be composed of various material types and permits a choice of fragment shapes.

The programs considered here utilize two general methods to determine the fuze type. SCAN, ATTACK and SESTEM II all have option lists from which the user choses a specific fuze type. REFMOD and SHAZAM allow the user to model any fuzing desired in a subroutine and utilize that subroutine in the program to fuze the warhead.

Vulnerable area type Endgame programs often require a seperate model for the fuze initiation (see Fig 3-4). Component Pk/h type Endgames usually have the fuzing model incorporated into the geometric model.

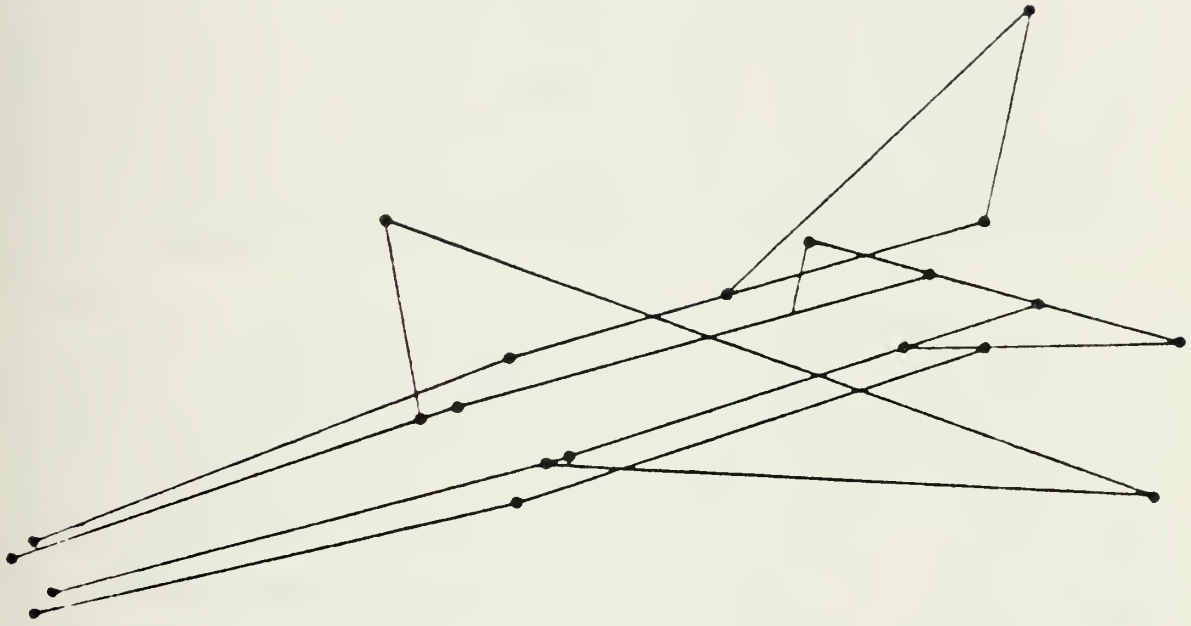


Fig. 3-4 Vulnerable Area Fuze Model

D. INPUT FORMAT

The ease of inputting information to the program can greatly affect the overall "usability" of an Endgame simulation. This section will discuss several methods of inputting the necessary information.

All of the programs use some type of formatted input. The input data is supposed to be in a predefined order, with specific values placed in designated columns on an IBM card (or a facsimile file on disk or tape storage).

One method of input is to use one data file for all of the information required. This requires the manipulation of a large block of data in order to make minor changes. It also requires the user to be familiar with the entire input file to make changes and assemble data for program execution. ATTACK, REFMOD and SESTEM II all utilize this type of input.

Another method is to subdivide the input data into smaller, functionally oriented input files. The SCAN program, as it is implemented at the Naval Postgraduate School, is an example of this technique. The function oriented files deal with a specific part of the input required such as warhead data, encounter geometry, and the target geometric model. This allows the user to modify a specific section of input without having to manipulate all of the input data. It also allows the compilation of a library of functional data files that may be combined to provide the desired scenario for a simulated encounter. This method is more user oriented, and while the programming required for this type of input may be more complex (not always the case), the ease of use justifies the time spent in additional programming.

One major feature that should be in any Endgame program is a preprocessing graphics capability for plotting the

targets. Since it is very easy to make mistakes when preparing the target modeling data, especially in the COMGEOM approach, a plot of the modelled target is essential for verification. Only SCAN currently has this capability. The SPLGEN program takes the same input for the geometric model as SCAN does. It uses the input data to check for correct modeling and then produces a plot or display of the model.

E. AVAILABILITY OF TARGET MODELS AND OTHER INPUT DATA

The two programs that have the most documented input data and target models available are SCAN and REFMOD. SCAN is a component Pk/h type program utilizing a COMGEOM target model and REFMOD is a vulnerable area type program using a finite element direct-hit model. Since both models have been extensively used, large quantities of both types of data are available. New and updated targets and data are continuously being produced for use with both SCAN and REFMOD. The ATTACK program can utilize the REFMOD data with only minor modifications. Information about the production of data and models for the other programs was not readily available.

F. PROGRAM OUTPUT

The output format must be user oriented if the program is to be useful. The computer must be utilized to do as much of the output organization and interpretation as possible. This does not mean it should supress any of the output data, but rather the computer should be used to process and collate the raw data in such a manner as to allow direct use of the output without having to do additional interpretation. All of the programs considered in this study incorporate this ideal to varying degrees. The ability of these programs to allow changes in output format varies from one simulation to the next.

The ability to combine damage types and create systems or subsystems from the target component output is an essential requirement. All of the programs have the capability to do this, but the SCAN program is the easiest to use. SCAN allows the linking of individual components into systems through the use of logical .AND/.OR statements.

G. GRAPHICS CAPABILITIES

The need to generate graphic output from Endgame simulations has been sorely underemphasized. A preprocessor graphics program can be invaluable to the target modeler in tracing errors in the input data file, and the added impact of a visual representation of the fragment hit locations using a post-processor is immense. The ability to view the damaged area of the target and to follow the fragments as they pass through the target can be a tremendous aid when interpreting the output data.

Only the SCAN and SHAZAM programs contain a graphics option. The version of SCAN used at the Naval Postgraduate School is linked to two graphics programs, a preprocessor and a postprocessor. The use of graphics at NPS allowed the correction of several complex target model inaccuracies that never would have been discovered without the added capabilities. Figure 3-5 is an example of computer generated graphics.

H. COMPUTATIONAL TIME AND COST REQUIREMENTS

It should be obvious that the faster the program executes, the better it seems from the standpoint of the user. The multitude of factors that control program execution time are, in most cases, dependent upon the computer facilities available and not upon the programs themselves. Since at a large modern computer facility, the execution times of even the larger simulations are under five seconds (single

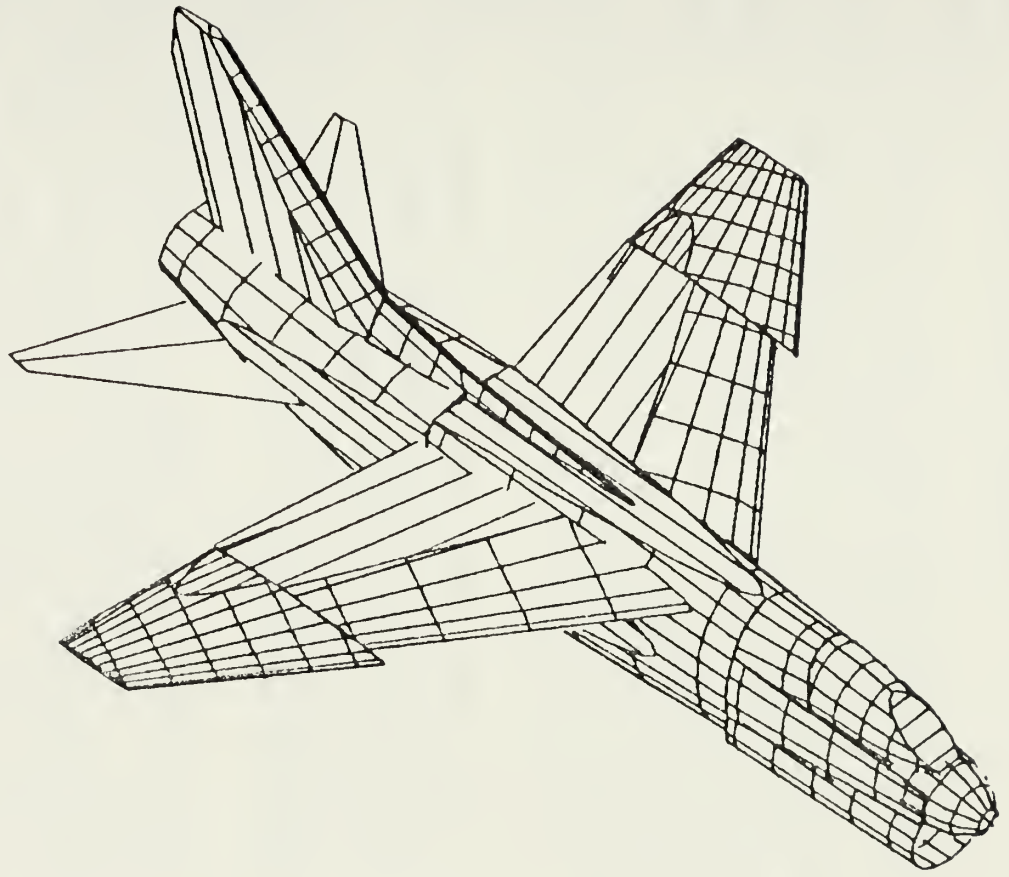


Fig. 3-5 Computer Generated Graphics

encounter case), the question of execution time is really more a question of the cost of computer time. The relatively high cost of computer time at many facilities could make a vulnerable area type of program more appealing due to a slightly faster execution time (this would be dependent upon the specific case). The use of any graphics programs can also be very time consuming and therefore costly. Since the cost-per-computation of computer time is dropping due to advances in technology, the cost differential between different types of programs should decrease.

I. DOCUMENTATION

With the exception of SCAN, the documentation for the current Endgame programs is wholly inadequate. Most of the programs had example problems to use for familiarization, but in several cases the example values given did not match the parameters of the User Manual. In all cases the documantation for the math models of the programs was very good. The key weaknesses were failure to explain how the actual program functioned and to thoroughly explain the input data parameters.

The SCAN documentation suffered from very few of these problems. It is user oriented and easily understood. In addition to an overall example problem, SCAN also has many small examples that illustrate individual parameters. There is a complete explanation of how the program functions and how each subroutine works.

The use of explanatory comments in the body of the computer program itself is another problem area. All of the programs had some explanation in the program text, but in all cases it was not sufficient to allow a new user to quickly locate and understand individual program functions.

IV. RECOMMENDATIONS FOR FUTURE ENDGAME PROGRAMS

This chapter will discuss those features that seem best suited to a standardized, user oriented Endgame simulation.

A. BASIC REQUIREMENTS

The programming should be done in a standard language that is compatible with all types of computers. The most logical choice would be the most current version of Standard FORTRAN. This will allow the program to be machine independent and avoids the present problem of program translation.

The program decided upon should be for use by all services and not oriented toward any special user group. This will eliminate the current tendency towards "private" Endgame programs that cannot readily be compared to one another. It would also simplify the problems of corporations that are working or bidding on service contracts. All companies would have the same Endgame simulation so that comparisons between bidders on the basis of simulation results would be valid, and a company or research facility could use the same simulation regardless of the particular service they were doing work for.

A single tri-service program would also generate a vast data base that could be referenced by any qualified user. This would avoid the current problem of constructing multiple, independent data bases that are not transferable due to program incompatability. This would require the creation of an organization to manage and maintain the program and it's data base.

The use of a single program also means that improvements would be disseminated quickly and that the program should

converge rapidly to a more valid simulation due to many facilities working with and trying to optimize the same program.

B. TARGET MODELING AND VULNERABILITY TYPE

The use of combinatorial geometry (COMGEOM) modeling is the most efficient choice. By using the COMGEOM model of the target for fragment, energy density, area removal, and direct hit vulnerability, only one additional model, the blast model, would be required. If the blast radius was modeled for each component as it is in the SHAZAM program, only one model would be required for all damage modes.

The selection of COMGEOM would still allow the use of finite element modeling and the associated computer generated targets, but it would also be able to use the more accurate complex geometric shapes. Since it is feasible that a program could be written to generate COMGEOM target models, it could eventually be possible to have both the accuracy of COMGEOM and the finite element ease of modeling. Until such a program is developed, a master library of models and components would eliminate most of the need for independent model development.

The use of component Pk/h for fragment vulnerability is well suited for use with COMGEOM target modeling since in many cases an entire component can be modeled with one shape. The use of component Pk/h also allows the user to modify the Pk/h of individual components, possibly to reflect new physical test data. This will also permit the accurate modeling of spallation effects by allowing the use of various material types in constructing the target model, and permit modification of the function used to determine component Pk/h. Additionally, the COMGEOM component Pk/h model also permits the use of many material types in simulating components, an important factor in view of the

rapid changes in aircraft structural materials presently occuring.

C. WARHEAD AND FUZE MODELING

The warhead model should be able to handle the numerous types of warheads in use or proposed for use. The most efficient way to handle the warhead model is not by using built-in warhead options, but instead by using specialized warhead subroutines that are input with the other data and called by the main program as is done in SHAZAM. This would allow very accurate warhead models to be created and permit new warhead designs to be added at a later date. It would streamline the main program by not having to offer an extensive selection of warhead options.

The fuzing models should also be subroutines input with the data for the same reasons that apply to the warhead model.

D. INPUT FORMAT

The input should be broken up into small functional subgroups, e.g. fuzing, warhead, target model etc. This type of input allows easy manipulation of data and simplfies tracing errors in input data.

E. OUTPUT FORMAT

The output format utilized by SCAN is a good choice to use. It mirrors the input values and encounter conditions, lists user defined systems and subsystems, tabulates PK information for each component and the user defined systems, and gives a summary table for each type of kill mechanism for the target as a whole.

F. FLYOUT SIMULATION INTERFACE

The ultimate missile versus aircraft program would be a missile flyout simulation that interfaced with an Endgame simulation. This interfacing of simulations would allow the computation of PKSS values for the missile systems simulated. SHAZAM has been used in this manner so that this is not an untried concept. Ideally, the flyout and Endgame programs should be designed as one large program that encompasses two "stand alone" programs. A truely sucessful integration of this type would require the generation of flyout programs that generate output specifically structured to provide the values required by the Endgame program.

G. GRAPHICS CAPABILITY

It is essential that new Endgame simulations provide for graphic output, preferably ineractive graphic routines. The two graphic routines designed for use with SCAN, SPLGEN and SPDRAW, are good examples of the graphics programs needed by Endgame simulations. One of these routines, SPLGEN, is a preprocessor that checks the target model for geometric accuracy then plots the model on a plotter or a cathode ray tube (CRT) display. The preprocessor is independent of the actual Endgame program, and is extremely useful to someone modeling targets for Endgame simulation use. The other routine, SPDRAW, is a postprocessor that utilizes the output from the Endgame program. The Endgame program generates a data file containing the location of all the fragment hits on the target model. This routine plots the aircraft then superimposes the fragment hits onto the aircraft plot. This program also displays output on either a plotter or a CRT display.

The visual image of the target damage is extremely useful in interpreting the damage caused by a missile warhead. The use of graphics will allow extensive investigation of

Endgame simulation validity by enabling comparisons to be made directly between photographic test data and simulations of the same encounter.

The COMGEOM component Pk/h modeling is ideal for accurate graphics output. An example of computer generated graphic output for a COMGEOM component Pk/h model is shown in Figure 4-1 (utilizing a SCAN target and the SPLGEN graphics program).

H. DOCUMENTATION

This is one of the weakest aspects of most Endgame programs. The three areas that require extensive documentation are (1) program use (a user's manual and explanation of program functioning with extensive example problems), (2) the math models for the program, and (3) the text of the program itself.

The SCAN User Manual and Analyst Manual are good examples of adequate documentation for program use and math models. The user is given step-by-step instructions complete with examples and diagrams for using the program. A sample problem is provided that illustrates the correct input format and gives the output values that should be obtained. The functioning and purpose of each routine or subroutine is explained in detail. Extensive flowchart diagrams trace out the program execution paths for easy reference if troubleshooting is required. The math models are well explained and the explanation relates the model to the functioning of the program for ease of understanding.

The present Endgame programs do not have enough documentation in the body of the program itself. The program should be well labeled and contain brief explanations that would enable an unfamiliar user to locate specific functional areas of the program.

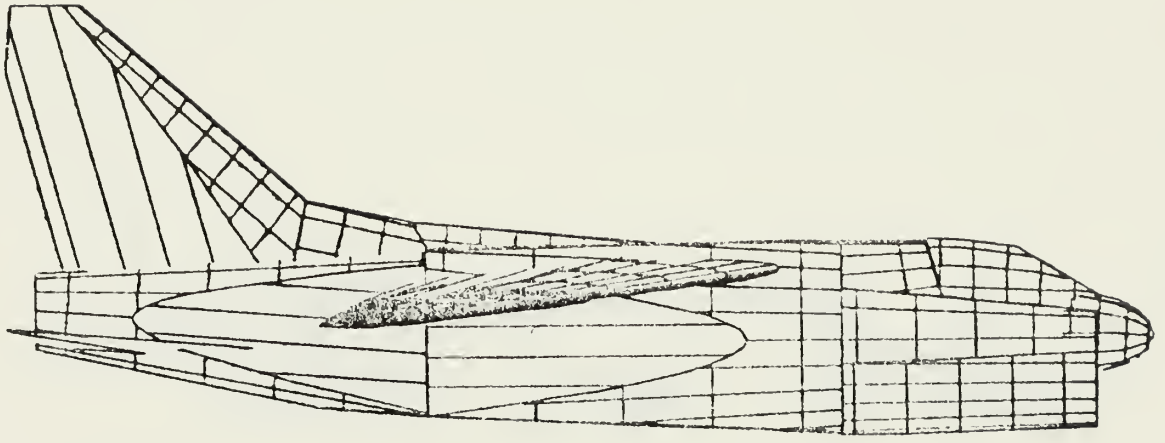


Fig. 4-1(a) Exterior View Graphics Example

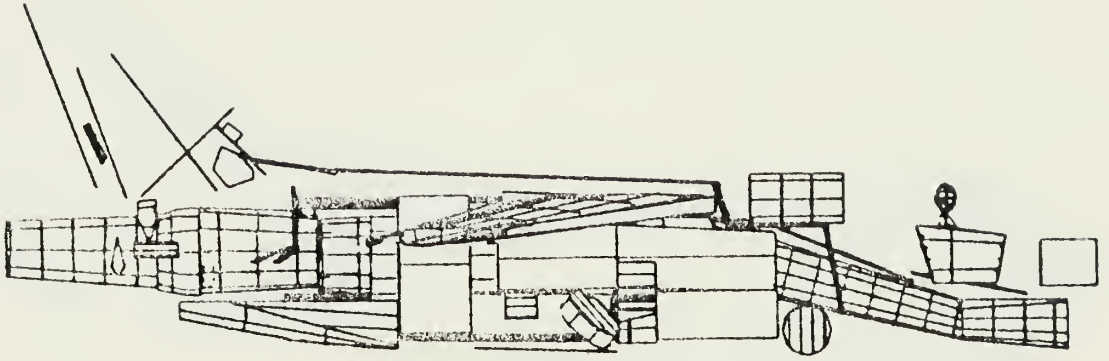


Fig. 4-1(b) Interior Components Graphics Example

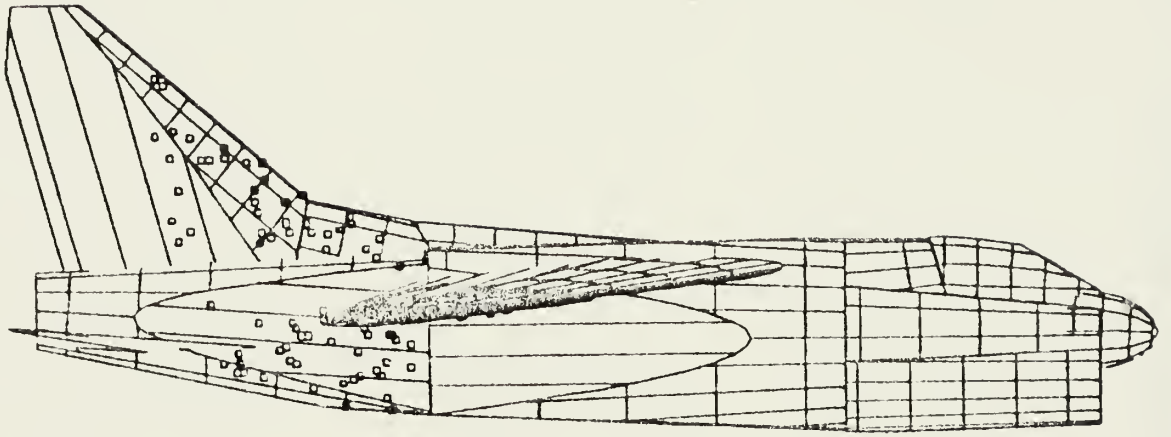


Fig. 4-1(c) Exterior View Graphics Showing Hits

It is important that any changes made to the program be thoroughly documented and overall program documentation updated on a regular basis. If the program is not kept well documented as changes occur, the whole purpose of documentation is defeated and an unfamiliar user will not be able to utilize the program without having to do extensive trial and error testing to find the undocumented changes.

V. DIRECTION OF CURRENT RESEARCH

This chapter will discuss those programs that are still in development or are undergoing major modification. This information was provided by the program developers.

A. SCAN

The SCAN program is currently in the process of being modified at the Pacific Missile Test Center, Point Mugu. The new version is referred to as "Son of SCAN" and has been substantially streamlined. Many of the modifications being incorporated are to enhance the graphics capability of the original program. It is not known at this time whether Son of SCAN will become generally available.

B. SHAZAM

The SHAZAM program, by intent, will always be in a state of modification. The documentation package for SHAZAM should become available within the next year. The documentation will explain the "bookkeeping" core of the program and discuss the requirements for subroutines to be added by the user.

C. REFMOD

The latest version of REFMOD, REFMOD II, is not fully operational at this time. Modifications are being carried out at the Naval Weapons Center, China Lake. The latest version has a more flexible fuze package, more warhead options, better documentation, and a much simpler input format. Preliminary documentation is currently available, and the new program should be available by August 1981.

APPENDIX A

A USER'S MANUAL FOR SCAN AT NPS

This User's Manual provides instruction for the execution of SCAN at NPS using the IBM 3278 display terminal.

A. STORAGE REQUIREMENTS

The compiled version of the SCAN program requires approximately one cylinder of storage on the IBM 3033 system (2240 records). The source code version (FORTRAN) requires approximately two cylinders of storage (4322 records). Execution of the program typically requires less than one cylinder of storage for the output. The exact amount of storage required during execution is dependent upon the number of cases, the number of fragments in the warhead, and the complexity of the target

B. USER INSTRUCTIONS FOR EXECUTION

This section will deal with the execution of SCAN using existing case, warhead, and target geometry files stored on a read-only disk file. The creation of new input files will be discussed in later sections.

1. Turn on the terminal.

2. When the large "NPS" logo appears on the screen, press the RESET key followed by the ENTER key.

3. When "CP READ" appears in the lower right of the screen, type in L nnnnP, where nnnn is your user identification number. Press ENTER.

4. You will now be asked for your password. Type in your password (it will not appear on the screen), then press ENTER.

5. Type "CP LINK TO xxxxP 191 AS 192 RR", where xxxx is the user number of the project file. Press ENTER.

6. You will now be asked for the project password. Type it in (it will not appear on the screen). Press ENTER.

7. Type "ACCESS 192 B" and press ENTER.

8. Type "PROFILE EXEC" and press ENTER. This step assumes that you do not currently have a PROFILE EXEC file on your private disk. If you have your own PROFILE EXEC, you must ensure that it contains the following command:

GLOBAL TXTLIB FORTMOD2 MOD2EEH

9. The SCAN program will require three data files in order to execute properly. These files are:

(a) GEOM DATA - a file containing the geometric target model, limiting parameters and kill expressions.

(b) WARHEAD DATA - a file containing the missile, warhead and fuzing parameters.

(c) CASE DATA - a file containing the encounter geometry information.

The WARHEAD DATA and CASE DATA files are normally preselected and ready for use. Changes to these files will be discussed in later sections. The GEOM DATA file must be chosen from a list of available targets (see Table AA-1). To transfer the target file chosen to the GEOM DATA file for input into the program, type in "COPY Fn Fm B1 GEOM DATA A1"and press ENTER. Fn refers to the file name of the target desired and Fm refers to the file mode of the target file as given in Table AA-1.

10. You are now ready to run SCAN. At NPS, the SCAN program is stored under the title "SCANMAIN". This version has been specially modified for the NPS computer system. You should insure that there is enough room on your disk for the output files (approximately one cylinder).

TABLE AA-1

TARGETS AVAILABLE FOR USE WITH SCAN

Target	Filename	filetype
A-7	A7DATA	
Harpoon	AGM86	DATA
Tomahawk	AGM109	DATA
Backfire	BACKF	DATA
Shoebox	BOX	DATA
Drone	BQM34	DATA
Drone	BQM107	DATA
Exocet	EXOCET	DATA
Foxbat	FOXBAT	DATA
Kingfish	KINGF	DATA
Kitchen	KITCH	DATA

12. Type in "RUN SCANMAIN" then press ENTER. You should get the following listing on your screen:

```
FILEDEF 01 DISK SCAN1 FORTRAN (RECFM FA BLOCK 131 PERM

FILEDEF 05 DISK CASE DATA

FILEDEF 06 DISK SCAN2 FORTRAN (RECFM FA BLOCK 131 PERM

FILEDEF 11 DISK GEOM DATA

FILEDEF 12 DISK WARHEAD DATA

FILEDEF 19 DISK IMPACT DATA (RECFM VBS LRECL 127 BLOCK 131

FILEDEF 30 DISK FILE DATA


LOAD SCANMAIN

START

EXECUTION BEGINS...
```

The program will take a few seconds to execute. The actual run time will depend upon computer workload, target and warhead complexity, and the number of cases per run.

12. Once the program has completed running, type "L" and press ENTER to see if the output files were generated on your disk. The output files that should be created are:

(a) SCAN1 FORTRAN A1 - this file is an echo print of target parameters.

(b) SCAN2 FORTRAN A1 - this file contains a summary of the encounter conditions, warhead/missile parameters, system definition statements and output summary.

(c) FILE DATA A1 - this file contains the system definition statements in assembly language. It is created and used by the SCAN program, and is not of any value to the user.

(d) IMPACT DATA A1 - this file contains the assembly language coding for the fragment impact points. It is used in the SCAN graphics program. For information concerning the use of graphics with the SCAN program, see the thesis of LCDR T. Hayes [Ref. 5].

13. To obtain printouts of the SCAN results, type "PRINT SCAN1 FORTRAN (CC" then press ENTER then type "PRINT SCAN2 FORTRAN (CC" and press ENTER. The printout of these two files will be output by the line printer in Ingersoll 140 and filed alphabetically in the output bins according to your last name. Figures A-1 and A-2 are examples of SCAN1 FORTRAN A1 and SCAN2 FORTRAN A1 printouts.

14. To view the output of the program at the terminal display screen, use the XEDIT mode of the NPS VM/CMS time sharing system to review the files created (a guide to the full capabilities of XEDIT is available in the consultants office at the computer center). It will be necessary to move the display right or left to view all of the information in the files due to the line length of the output (the method for doing this is explained in the XEDIT User's Guide).

C. INSTRUCTIONS FOR MODIFYING OR CREATING A WARHEAD

This section deals with the creation of new WARHEAD DATA files and the modification of existing WARHEAD files for use with the SCAN Endgame program. For modeling purposes, the CG of the target is assumed to be at the origin of the target coordinate system as shown in Figure A-3. In reality, this may not be the actual target CG. Check the target model to find the origin of the target coordinate system by utilizing the target plots available in the thesis of LCDR T. Hayes [Ref. 5] before you start in order to avoid inputting incorrect data values.

TARGET GEOMETRICAL DESCRIPTION

CONVERSION FACTOR NAME IS FEET

BOX

COMP 1 ALUM 2024 0.015 IN. THICKNESS BOX NUMBER 1 COMMON NUMBER 1 PKH 0.100E+01 0.0 0.0

SING FRAG VULNERABLE	AND	EXTERNAL SKIN COMPONENT				
1.500 -1.500	-1.500	1.500	1.500	-1.500	-1.500	1.500
16.500 13.500	13.500	13.500	16.500	16.500	13.500	13.500
-1.500 -1.500	-1.500	-1.500	1.500	1.500	1.500	1.500

COMP 2 ALUM 2024 0.015 IN. THICKNESS BOX NUMBER 2 COMMON NUMBER 2 PKH 0.100E+01 0.0 0.0

SING FRAG VULNERABLE	AND	EXTERNAL SKIN COMPONENT				
1.500 -1.500	-1.500	1.500	1.500	-1.500	-1.500	1.500
13.500 10.500	10.500	10.500	13.500	13.500	10.500	10.500
-1.500 -1.500	-1.500	-1.500	1.500	1.500	1.500	1.500

COMP 3 ALUM 2024 0.015 IN. THICKNESS BOX NUMBER 3 COMMON NUMBER 3 PKH 0.100E+01 0.0 0.0

SING FRAG VULNERABLE	AND	EXTERNAL SKIN COMPONENT				
1.500 -1.500	-1.500	1.500	1.500	-1.500	-1.500	1.500
10.500 7.500	7.500	7.500	10.500	10.500	7.500	7.500
-1.500 -1.500	-1.500	-1.500	1.500	1.500	1.500	1.500

COMP 4 ALUM 2024 0.015 IN. THICKNESS BOX NUMBER 4 COMMON NUMBER 4 PKH 0.100E+01 0.0 0.0

SING FRAG VULNERABLE	AND	EXTERNAL SKIN COMPONENT				
1.500 -1.500	-1.500	1.500	1.500	-1.500	-1.500	1.500
7.500 4.500	4.500	4.500	7.500	7.500	4.500	4.500
-1.500 -1.500	-1.500	-1.500	1.500	1.500	1.500	1.500

COMP 5 ALUM 2024 0.015 IN. THICKNESS BOX NUMBER 5 COMMON NUMBER 5 PKH 0.100E+01 0.0 0.0

SING FRAG VULNERABLE	AND	EXTERNAL SKIN COMPONENT				
1.500 -1.500	-1.500	1.500	1.500	-1.500	-1.500	1.500
4.500 1.500	1.500	1.500	4.500	4.500	1.500	1.500
-1.500 -1.500	-1.500	-1.500	1.500	1.500	1.500	1.500

COMP 6 ALUM 2024 0.015 IN. THICKNESS BOX NUMBER 6 COMMON NUMBER 6 PKH 0.100E+01 0.0 0.0

SING FRAG VULNERABLE	AND	EXTERNAL SKIN COMPONENT				
1.500 -1.500	-1.500	1.500	1.500	-1.500	-1.500	1.500
1.500 1.500	-1.500	-1.500	1.500	1.500	-1.500	-1.500
-1.500 -1.500	-1.500	-1.500	1.500	1.500	1.500	1.500

COMP 7 ALUM 2024 0.015 IN. THICKNESS BOX NUMBER 7 COMMON NUMBER 7 PKH 0.100E+01 0.0 0.0

SING FRAG VULNERABLE	AND	EXTERNAL SKIN COMPONENT				
1.500 -1.500	-1.500	1.500	1.500	-1.500	-1.500	1.500
-1.500 -1.500	-4.500	-4.500	-1.500	-1.500	-4.500	-4.500
-1.500 -1.500	-1.500	-1.500	1.500	1.500	1.500	1.500

COMP 8 ALUM 2024 0.015 IN. THICKNESS BOX NUMBER 8 COMMON NUMBER 8 PKH 0.100E+01 0.0 0.0

SING FRAG VULNERABLE	AND	EXTERNAL SKIN COMPONENT				
1.500 -1.500	-1.500	1.500	1.500	-1.500	-1.500	1.500
-4.500 -7.500	-7.500	-7.500	-4.500	-4.500	-7.500	-7.500
-1.500 -1.500	-1.500	-1.500	1.500	1.500	1.500	1.500

COMP 9 ALUM 2024 0.015 IN. THICKNESS BOX NUMBER 9 COMMON NUMBER 9 PKH 0.100E+01 0.0 0.0

SING FRAG VULNERABLE	AND	EXTERNAL SKIN COMPONENT				
1.500 -1.500	-1.500	1.500	1.500	-1.500	-1.500	1.500
-7.500 -10.500	-10.500	-10.500	-7.500	-7.500	-10.500	-10.500
-1.500 -1.500	-1.500	-1.500	1.500	1.500	1.500	1.500

COMP 10 ALUM 2024 0.015 IN. THICKNESS BOX NUMBER 10 COMMON NUMBER 10 PKH 0.100E+01 0.0 0.0

SING FRAG VULNERABLE	AND	EXTERNAL SKIN COMPONENT				
1.500 -1.500	-1.500	1.500	1.500	-1.500	-1.500	1.500
-10.500 -13.500	-13.500	-13.500	-10.500	-10.500	-13.500	-13.500
-1.500 -1.500	-1.500	-1.500	1.500	1.500	1.500	1.500

COMP 11 ALUM 2024 0.015 IN. THICKNESS BOX NUMBER 11 COMMON NUMBER 11 PKH 0.100E+01 0.0 0.0

SING FRAG VULNERABLE	AND	EXTERNAL SKIN COMPONENT				
1.500 -1.500	-1.500	1.500	1.500	-1.500	-1.500	1.500
-13.500 -16.500	-16.500	-16.500	-13.500	-13.500	-16.500	-16.500
-1.500 -1.500	-1.500	-1.500	1.500	1.500	1.500	1.500

END

Fig. A-1 SCAN1 FORTRAN Example output

KILL EXPRESSIONS

FWD = 1 OR 2 OR 3 OR 4 OR 5 OR 6
AFT = 7 OR 8 OR 9 OR 10 OR 11

CASE DATA

NUMBER OF MISSILES IN SALVO 1
VELOCITY OF THE TARGET 700.00 FEET/SEC
TARGET ROLL ANGLE 0.0 DEGREES
TARGET PITCH ANGLE 0.0 DEGREES
TARGET YAW ANGLE 0.0 DEGREES
VELOCITY OF THE MISSILE 2300.00 FEET/SEC
NOMINAL MISSILE ATTACK ANGLE 0.0 DEGREES
DEV. OF ATTACK ANGLE 0.0 DEGREES
NOMINAL MISSILE ELEVATION ANGLE 0.0 DEGREES
STD. DEV. OF ELEVATION ANGLE 0.0 DEGREES
NOMINAL MISSILE AZIMUTH ANGLE 0.0 DEGREES
STD. DEV. OF AZIMUTH ANGLE 0.0 DEGREES
MISSILE ALTITUDE 10000.00 FEET
NOMINAL AIM POINT (WRT TARG CG) 0.0 0.0 0.0 FEET
TGT VEL YAW ANG 0.0
TGT VEL PCH ANG 0.0
FIXED TRAJECTORY MISSILE INITIAL POSITION W.R.T. TARGET CG = 25.000 0.0 0.0 FEET
MSL VEL YAW ANG 0.0
MSL VEL PCH ANG 0.0
ROLMIN = -1.725 ROLMAX = -1.412

Fig. A-2(a) SCAN2 FORTRAN Example Output

```
                                    WARHEAD FRAGMENTATION DATA

   POLAR                  INITIAL FRAGMENT          FRAGMENT        NUMBER OF     DISTANCE FROM      FRAGMENT          FRAGMENT
 ZONES (DEG)              VELOCITIES(FPS)           MASS(GRAINS)     FRAGMENTS     WARHEAD CENTER     MATERIAL          SHAPE

 75.0     105.0          5130.0      5130.0          105.00          2000.0          0.0              HARD  STEL          IRRE

                                  FUZE DATA
        FUZE TYPE          )
        FUZE POSITION (RELATIVE TO WARHEAD)          3.25  FEET
        DELAY TIME        0.0      SEC.
        FUZING ANGLE          90.0   DEGREES
        STANDARD DEVIATION OF FUZING ANGLE          0.0     DEGREES
        RANGE CUTOFF        20.000  FEET
        RADIUS OF MISSILE          1.500  FEET
        NOSE OF MISSILE IS          6.500  FEET IN FRONT OF THE WARHEAD
        TAIL OF MISSILE IS          6.500  FEET IN BACK  OF THE WARHEAD

                                  BLAST MODEL
     RADIUS OF FUSELAGE CYLINDER        10.00
     CYLINDER LENGTH FORWARD OF CG        16.50
     CYLINDER LENGTH AFT OF CG        16.50
     RADIUS OF WING CYLINDER          0.0
     STARTING POINT OF CYLINDER (WING ROOT)        1.50        2.50 .      0.0
     END POINT OF CYLINDER (WING TIP)        1.510       2.500       0.0

                              FACTORS USED TO DECREASE RUN TIME
     MAXIMUM LENGTH OF TARGET (MEASURED FROM TARGET CG)    34.0  FEET
     MAXIMUM WIDTH OF TARGET(MEASURED FROM TARGET CG)     4.0  FEET
     TARGET HEIGHT (MEASURED FROM TARGET CG)     4.0  FEET
     MAXIMUM EXTENSION OF TAIL SURFACE FROM CG      2.00  FEET
     DISTANCE BETWEEN FRAGMENT RAYS     0.10  FEET
     DISTANCE BEYOND WHICH WARHEAD IS NO LONGER CONSIDERED EFFECTIVE     50.0  FEET

WING LEADING EDGE - FROM      1.500      2.500      0.0   TO      1.510      2.500      0.0  (FEET)
WING TRAILING EDGE - FROM      1.510      2.490      0.0   TO      1.500      2.490      0.0  (FEET)

HOR STABILIZER LEADING EDGE - FROM      1.500     -15.000      0.0   TO      1.510     -15.000      0.0    (FEET)
HOR STABILIZER TRAILING EDGE - FROM      1.510     -15.100      0.0   TO      1.500     -15.100      0.0    (FEET)

VERTICAL TAIL LEADING EDGE - FROM      0.0      -15.000      1.500 TO      0.0      -15.000      1.510  (FEET)
VERTICAL TAIL TRAILING EDGE - FROM      0.0      -15.100      1.510 TO      0.0      -15.100      1.500  (FEET)
```

Fig. A-2(b) SCAN2 FORTRAN Example Output

SUMMARY OF MISSILE TRAJECTORY NUMBER 1

MISSILE ELEVATION ANGLE WITH RESPECT TO THE AIRCRAFT 0.0 DEGREES
MISSILE AZIMUTH ANGLE WITH RESPECT TO THE AIRCRAFT 0.0 DEGREES
MISS DISTANCE AT DETONATION (MEASURED PERPENDICULAR TO AIRCRAFT CENTERLINE) 25.00 FEET
CLOSING VELOCITY VECTOR IN TARGET SYSTEM 0.0 1600.0 0.0
RANGE AT DETONATION IN TARGET SYSTEM 25.0 0.0 0.0

COMPON NUMBER	NAME OF COMPONENT	COMPONENT MATERIAL	COMPONENT VULNERABILITY	NUMBER OF FRAG. HITS	TOTAL AREA REMOVED FROM COMP. SURFACE	PROBABILITY OF COMPONENT KILL
1	COMP 1	ALUM 2024	SING FRAG	5.00	0.031	1.000
2	COMP 2	ALUM 2024	SING FRAG	14.00	0.085	1.000
3	COMP 3	ALUM 2024	SING FRAG	9.00	0.054	1.000
4	COMP 4	ALUM 2024	SING FRAG	9.00	0.054	1.000
5	COMP 5	ALUM 2024	SING FRAG	6.00	0.025	1.000
6	COMP 6	ALUM 2024	SING FRAG	1.00	0.004	0.999
7	COMP 7	ALUM 2024	SING FRAG	0.0	0.0	0.0
8	COMP 8	ALUM 2024	SING FRAG	0.0	0.0	0.0
9	COMP 9	ALUM 2024	SING FRAG	0.0	0.0	0.0
10	COMP 10	ALUM 2024	SING FRAG	0.0	0.0	0.0
11	COMP 11	ALUM 2024	SING FRAG	0.0	0.0	0.0

SURVIVAL PROBABILITIES, WITH SAMPLE SIZE = 1

SYSTEM CODES	PROBABILITY OF SURVIVAL	AVERAGE PROB. OF SURVIVAL	STANDARD DEVIATION	STANDARD ERROR OF THE MEAN	90 PERCENT CONFIDENCE INTERVAL	NO. OF TIMES SYSTEM SURVIVED
END	0.0	0.0				0
AFT	1.000	1.000				1
CLASS	1.000	1.000				1
DIGHT	1.000	1.000				1

Fig. A-2(c) SCAN2 FORTRAN Example Output

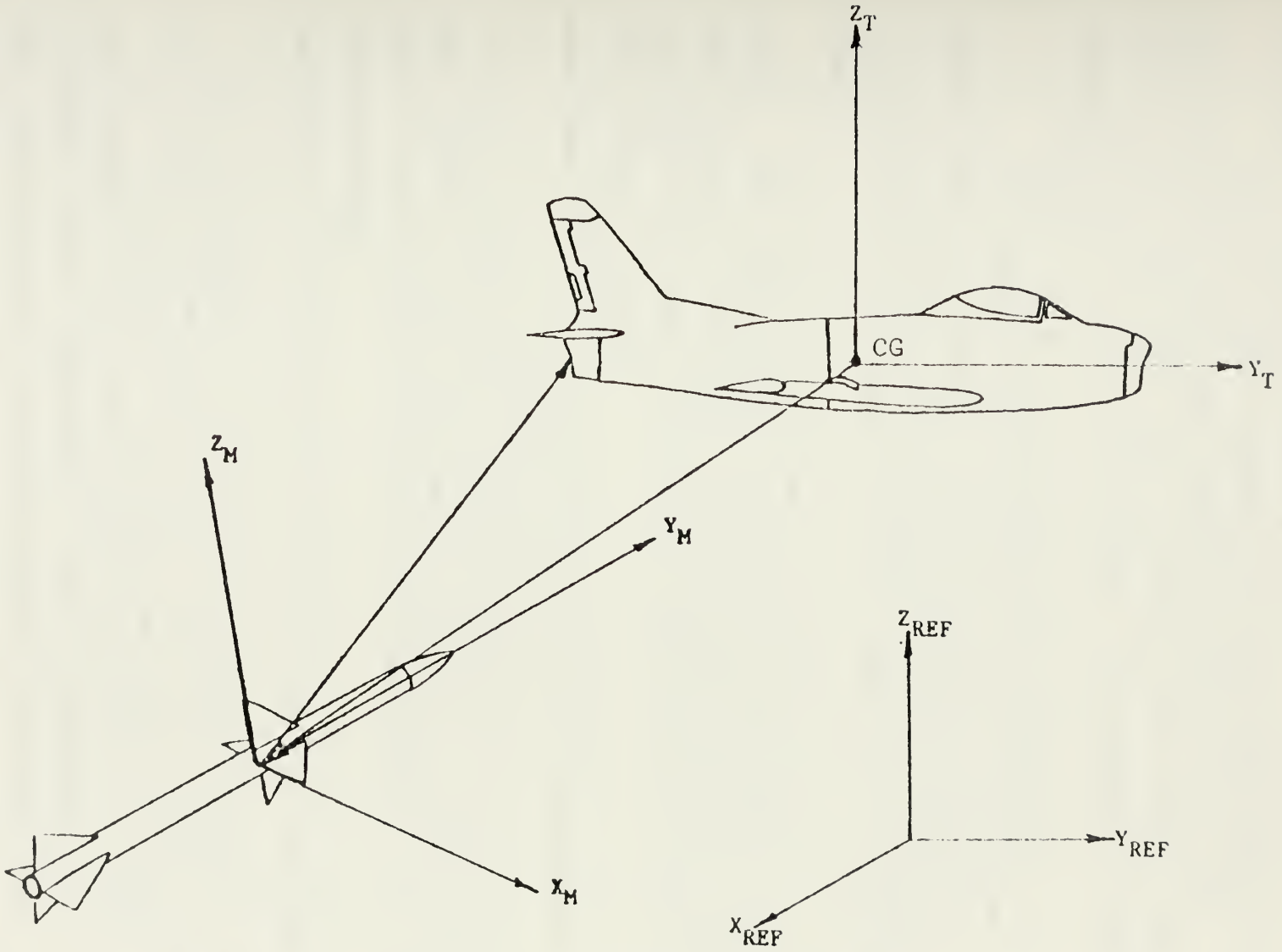


Fig. A-3 Assumed Target CG

1. Turn on the terminal and log onto the system as explained in Section B, steps 2 - 8.

2. Type in "XEDIT xxxxxxxx DATA"and press ENTER, where xxxxxxxx is the file name selected by you (up to eight spaces, the first space must be filled with a letter, numbers may be used in the other seven). An example is "XEDIT TESTHEAD DATA".

3. You are now in the XEDIT mode of operation.

4. For creation of a new file, type "I" and press ENTER. This will put you in the input mode required to create the new file (note: the input mode is not required to make changes to an existing file. This will be discussed later).

5. You will see an index line across the center of the screen. The index numbers correlate to the columns on an IBM data card. The input line is directly below the index line which will help you place the required data in the correct columns for input.

****

IMPORTANT NOTE!!!

All integer values must be RIGHT JUSTIFIED in the allotted input columns. Real values may be anywhere in the specified field. Letter characters must be LEFT JUSTIFIED in the specified columns.

****

6. In columns 1 - 10: enter the number of static polar zones in your warhead (up to 36). This is an integer value.

7. In columns 11 - 20: enter the number of fragment mass classes for each static polar zone (up to 3). This is an integer value. Press ENTER. You will note that the first line has moved up and the cursor is now ready for the next line.

8. In columns 1 - 10: enter the lower angle of the first static polar zone (see Fig. A-4) for definitions of lower and upper angles). A real value of F10.3 format, angle measured in degrees from the forward end of missile roll axis (0 - 180).

9. In columns 11 - 20: enter the upper angle of the first static polar zone. A real value, F10.3 format, measured in degrees from the forward end of missile roll axis (0 - 180).

10. In columns 21 - 30: enter the speed of the first mass class of fragments at the lower boundary of the first static polar zone. A real value, F10.3 format, in units of feet/sec.

11. In columns 31 - 40: enter the speed of the first mass class of fragments at the upper boundary of the first static polar zone. A real value, F10.3 format, in units of feet/sec.

12. In columns 41 - 50: enter the mass of the first mass class of fragments ejected in the first static polar zone. A real value, F10.3 format, in units of grains.

13. In columns 51 - 60: enter the total number of fragments of the first mass class contained in the first ststic polar zone. A real value, F10.3 format.

14. In columns 61 - 70: enter the initial position of the center of fragments of the first mass class in the first static polar zone with respect to the center of the warhead, as measured along the missile roll axis. A real value, F10.3 format, the units are feet.

15. In columns 71 -74: enter the material code indicating the type of material for the first mass class of fragment in the first static polar zone. Table AA-2 lists the material types available and the appropriate code for each type. An integer value, I4 format.

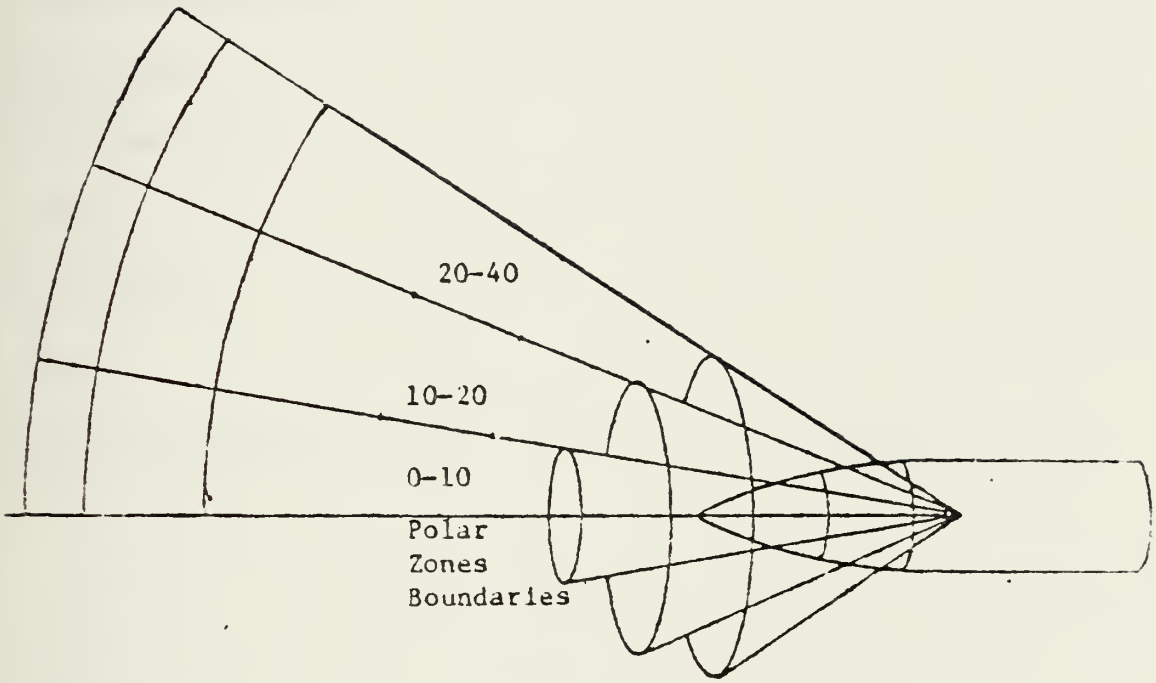


Fig. A-4 Warhead Static Polar Zones

TABLE AA-2

LIST OF MATERIAL TYPES

Material	Input Code
Magnesium	10
Aluminum 2024T	20
Titanium Alloy	30
Face Hardened Steel	40
Mild Steel	50
Hardened Steel	60
Lexan	70
Plexiglass	80
Doron	90
Bullet Resistant Glass	100

16. In columns 77 -80: enter the shape of the fragments in the first mass class in the first static polar zone. Table A-3 lists the shapes available for use. An alphanumeric character string, A4 format. Press ENTER.

17. Repeat steps 8 - 16 for each fragment mass class in the first static polar zone then repeat steps 8 - 16 for each additional static polar zone (up to 36), and repeat steps 8 - 16 again for each fragment mass class in each static polar zone. The number of times you will have to execute steps 8 - 16 is equal to the number of fragment mass classes multiplied by the number of static polar zones. EXAMPLE: for three fragment mass classes and two static polar zones, steps 8 - 16 will be repeated six times.

18. In columns 1 - 10: enter the type of fuze to be simulated. The options are:

(a) 0 for instantaneous detection and detonation.

(b) 1 for fuze on IR source only (requires IR source on target).

(c) 2 for fuze on any reflected target source.

An integer value, I10 format.

19. In columns 11 - 20: enter the position of the proximity fuze target detection device (TDD) with respect to the warhead center. This is a real value, F10.3 format, measured in feet along the missile roll axis.

20. In columns 21 - 30: enter the delay time between target detection and warhead detonation. A real value, F10.3 format, measured in seconds.

21. In columns 31 - 40: enter the mean value of the proximity fuze cone half-angle (look-angle) as measured from the forward missile roll axis. A real value, F 10.3 format, measured in degrees (0 - 180).

22. In columns 41 - 50: enter the standard deviation of the fuze cone half-angle, assuming a normal distribution of

TABLE AA-3

FRAGMENT SHAPES AVAILABLE FOR USE WITH SCAN WARHEAD MODEL

Fragment Shape	Program Code
Cube	CUBE
Spheroid	SPHE
Rectangular	RECT
Irregular	IRRE

angles. A real value, F10.3 format, measured in degrees (0 - 180).

23. In columns 51 - 60: enter the proximity fuze cutoff range. A real value, F10.3 format, measured in feet along the normal to the roll axis from the TDD position. If the fuze type selected was 0 then enter 0.0 for this value.

24. In columns 61 - 66: enter the radius of the missile cylindrical body. A real value, F6.2 format, measured in feet.

25. In columns 67 - 72: enter the position of the missile contact fuze (usually at the tip of the nose). A real value, F6.2 format, measured in feet from warhead center along the missile roll axis.

26. In columns 73 - 78: enter the distance the missile extends aft of the warhead center. A real value, F6.2 format, measured in feet along the missile roll axis. Press ENTER.

27. In columns 1 - 10: enter the fuselage blast radius. This is the maximum distance from the target centerline at which detonation of the warhead will cause catastrophic structural failure, assuming the target is at sea level. A real value, F10.2 format, measured in feet from the target centerline.

28. In columns 11 - 20: enter the distance from target CG to the front of the fuselage blast cylinder. A real value, F10.2 format, measured in feet.

29. In columns 21 - 30: enter the distance from target CG to the back of the fuselage blast cylinder. A real value, F10.2 format, measured in feet.

30. In columns 31 - 40: enter the wing blast radius. This is the maximum distance from the wing at which detonation of the warhead will cause catastrophic structural failure, assuming the target is at sea level. A real value, F10.2 format, measured in feet from the wing centerline.

31. In columns 41 - 70: enter the X, Y, Z components of the end point of the wing blast centerline closest to the target fuselage. The X, Y, Z values are real, 3F10.2 format, measured in feet from the target CG. Press ENTER.

32. In columns 1 - 30: enter the X, Y, Z components of the end point of the wing blast centerline furthest from the target fuselage. The X, Y, Z values are real, 3F10.2 format, measured in feet from the target CG. Press ENTER. An illustration of the blast model is shown in Figure A-5.

33. You have now completed your warhead design. Press ENTER. The cursor should now be back in the lower left corner of the screen and an END OF FILE statement should show up as the last entry in the file you just finished. Type "FILE" and press ENTER. Your screen should now revert to the format it had before you entered the XEDIT mode. Type "L" and ENTER. You should see a listing of all the files on your disk space. The warhead DATA file you created should now appear in that listing.

34. To use your warhead design in the SCANMAIN program, the warhead file must be named "WARHEAD". If the file you created is not named WARHEAD, you must rename it. Type in "COPY Fn DATA A1 WARHEAD = =" then press ENTER. The Fn refers to your original filename. This will not rename your original file, instead it creates a copy of it with the correct name. Thus, on your file listing you will have your original warhead file and a duplicate of it under the filename of WARHEAD.

D. MODIFYING A WARHEAD FILE

This section will deal with modifying an existing WARHEAD DATA file. Extensive reference will be made to Section C of this Manual.

1. Turn on the terminal and log onto the system.

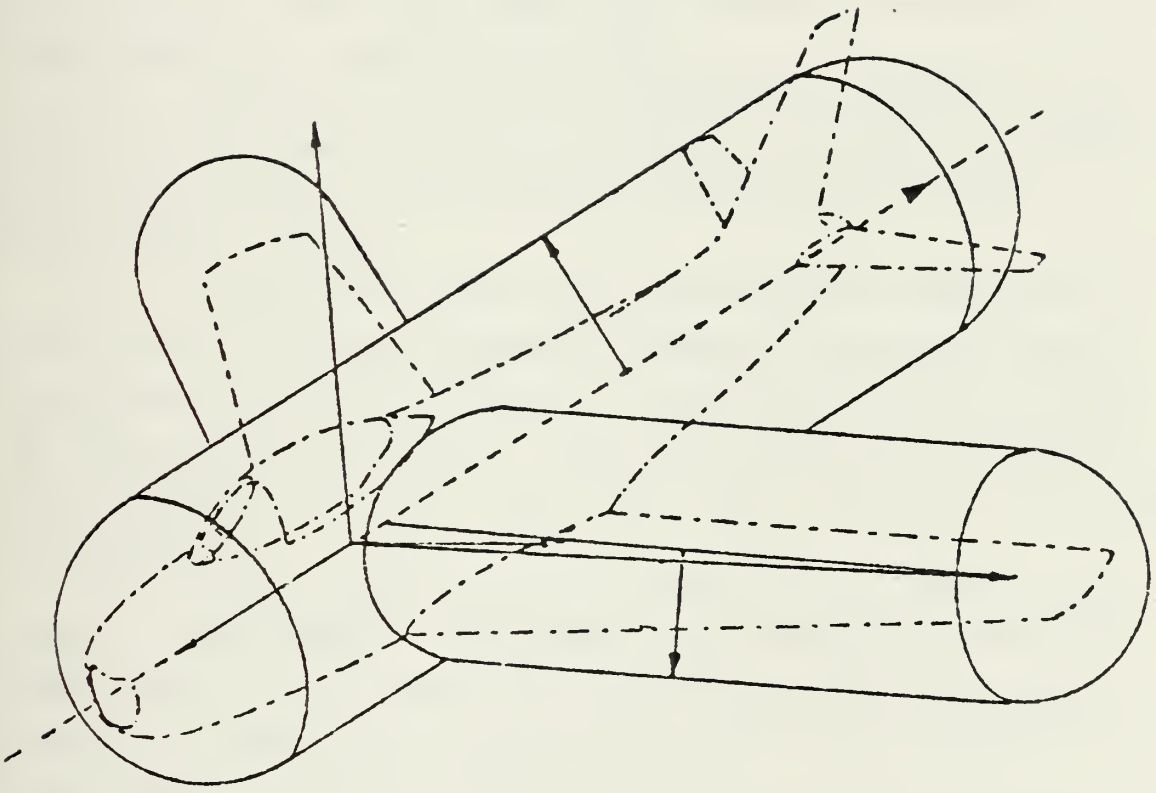


Fig. A-5 SCAN Blast Model

2. Type "XEDIT Fn Fm" and press ENTER. Fn refers to the filename of the file you wish to modify, and Fm refers to the filemode, usually a DATA file. An example is: XEDIT WARHEAD1 DATA.

3. You should now have the data in the file visible on the display screen. If the data covers more than one screen, use the following commands to survey the file:

(a) Press "ALT" and "PF8" keys at the same time to advance one screen forward in the file.

(b) Press "ALT" and "PF8" keys at the same time to revert back one screen in the file.

4. Use Section C of this Appendix to locate the position of the values you wish to change. Display the section of the data file to be changed on the screen by using instruction 3 above.

5. By using the four cursor positioning keys just to the right of the main keyboard, position the cursor under the values you wish to change. Simply type the new values in over the previous ones. Make sure that the new values are justified correctly and in the right format as discussed in Section C.

6. After completing your changes on the page on the screen, press ENTER. The cursor will return to the lower left corner of the screen and the changes will have been made. You must enter all changes on each full screen of data before "paging" through the file.

7. Once all changes have been entered and the cursor is back in the lower left corner of the screen, type "FILE" and press ENTER. This will permanently file your change and return you to the normal operating mode of the system. The corrected file is now ready for use.

E. CREATING A NEW CASE DATA FILE

This section will deal with the creation of new CASE DATA files for use with the SCAN Endgame simulation. The information in this section is also applicable to the modification of CASE DATA files currently available as discussed in the next section.

1. Turn on the terminal and log onto the system.

2. Type in "XEDIT xxxxxxxx DATA" and press ENTER. xxxxxxxx is the filename (up to eight spaces, the first space must be a letter, numbers may be used in the others). An example is: XEDIT TESTCASE DATA.

3. You are now in the XEDIT mode of operation. A guide to the full capabilities of XEDIT is available from the consultants office at the computer center.

4. Type "I" and press ENTER. This will put you in the input mode required to create a new file.

5. You will see an index line across the center of the screen. The index numbers correlate to the columns on an IBM data card. The input line is directly below the index line which will help you place the required data in the correct columns for input.

****

IMPORTANT NOTE!!!

All integer values must be RIGHT JUSTIFIED in the field specified. Real values may be anywhere in the specified field. Letter character strings must be RIGHT JUSTIFIED in the field specified.

****

6. In columns 1 - 10: enter the type of missile trajectory desired from Table AA-4. An integer value, I10 format. Press ENTER.

TABLE AA-4

MISSILE TRAJECTORY OPTIONS

Code	Meaning
0	Terminate execution of program
1	Indicates a fixed trajectory or detonation point specified by an initial position measured from the target CG
2	Indicates a trajectory with a fixed missile guidance error (CPA)
3	Indicates a trajectory in which the missile guidance error is computed from a normally distributed sample with a given CEP

7. In columns 1 - 10: enter the number of missile trajectories to be considered. An integer value, I10 format.

8. In columns 11 - 20: enter the target speed at time of intercept. A real value, F10.3 format, measured in feet/second. The value must be greater than zero.

The steps 9, 10, and 11 refer to Figure A-6.

9. In columns 21 - 30: enter the roll angle (PSI) of the target at intercept. A real value, F10.3 format, measured in degrees. The roll angle is measured with respect to the horizontal "flat earth" reference plane, right wing "down" is positive (0 - 360).

10. In columns 31 - 40: enter the pitch angle (THETA) of the target at intercept measured with respect to the flat earth plane. A real value, F10.3 format, measured in degrees. A positive angle indicates a climb, a negative angle, a dive (-90 to +90).

11. In columns 41 - 50: enter the yaw or heading angle (PHI) of the target at intercept. A real value, F10.3 format, measured in degrees from the Y-axis of the reference plane. Positively increasing yaw is measured in a counterclockwise rotation (0 - 360) as viewed looking down the Z-axis of the reference coordinate system toward the origin.

12. In columns 51 - 60: enter the missile speed at intercept. A real value, F10.3 format, measured in feet/second (>0).

13. In columns 61 - 70: enter a mean value for missile angle of attack. A real value, F10.3 format, measured in degrees. Figure A-7 depicts the angle desired.

14. In columns 71 - 80: enter the standard deviation of the missile angle of attack, assuming a normal distribution of angles. A real value, F10.3 format, measured in degrees. Press ENTER.

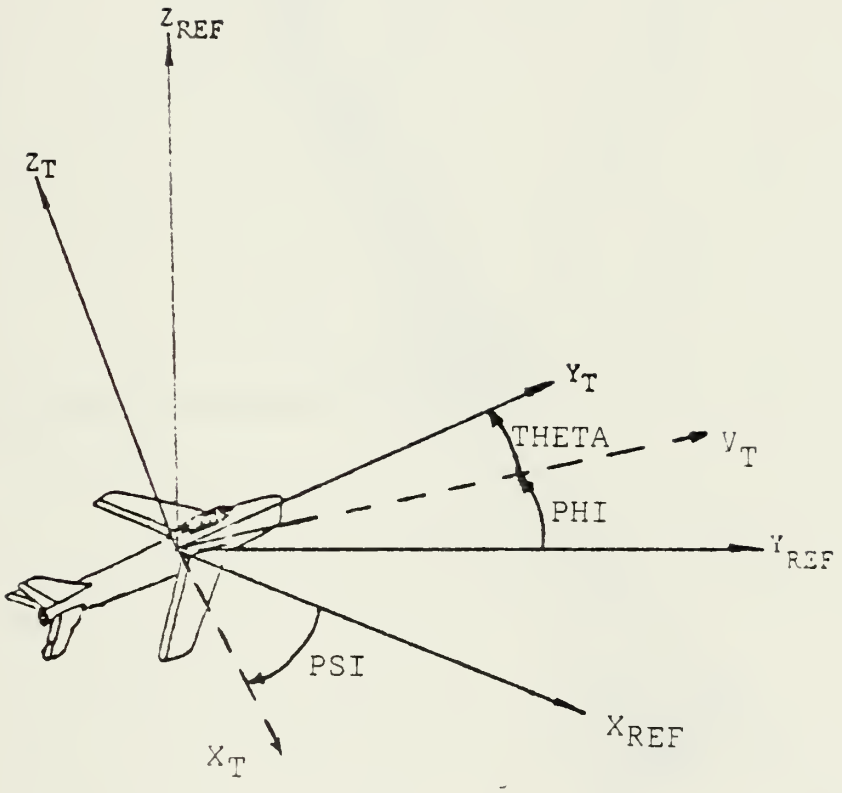


Fig. A-6 Target Roll, Pitch and Yaw Angles

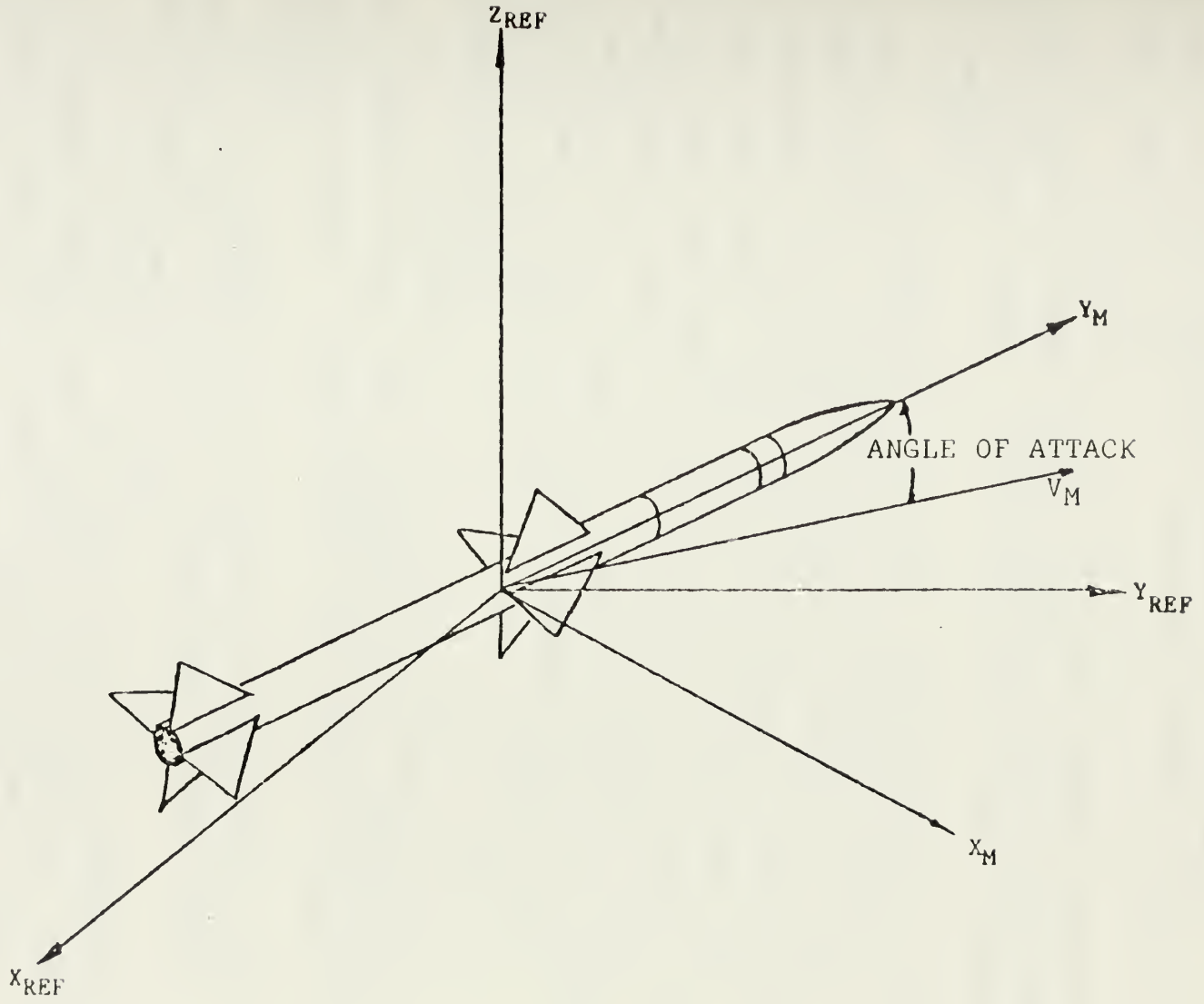


Fig. A-7 Missile Angle of Attack

15. In columns 1 - 10: enter the mean pitch angle of the missile. A real value, F10.3 format, measured in degrees (-90 to +90) with respect to a "flat earth". A positive angle indicates a climb and a negative angle, a dive (see Fig. A-8).

16. In columns 11 - 20: enter the standard deviation of the missile elevation angle, assuming a normal distribution of elevation angles. A real value, F10.3 format, measured in degrees.

17. In columns 21 - 30: enter the mean value of the azimuth angle of the missile in the terminal intercept. A real value, F10.3 format, measured in degrees (0 - 360) and referenced to the target coordinate system if target roll, pitch, and yaw are set at zero. If the target has roll, pitch, and yaw values the azimuth angle of the missile is referenced to the flat earth reference coordinate system (see Fig. A-9). If the roll, pitch, and yaw of the target are at zero, a value of zero for missile azimuth angle implys a tail-chase encounter, a value of 180 implys a head-on encounter.

18. In columns 31 - 40: enter the standard deviation of the missile azimuth angle assuming that the distribution of angles is normal. A real value, F10.3 format, measured in degrees.

19. In columns 41 - 50: enter the altitude above sea level at which the encounter takes place. A real value, F10.3 format, measured in feet.

20. In columns 51 - 80: enter the X, Y, Z components of the missile aimpoint with respect to the target CG. A set of real values, 3F10.3 format, measured in feet. This is the point on the target the missile is aiming for. Miss distances in the output are calculated with respect to the aimpoint selected. Press ENTER.

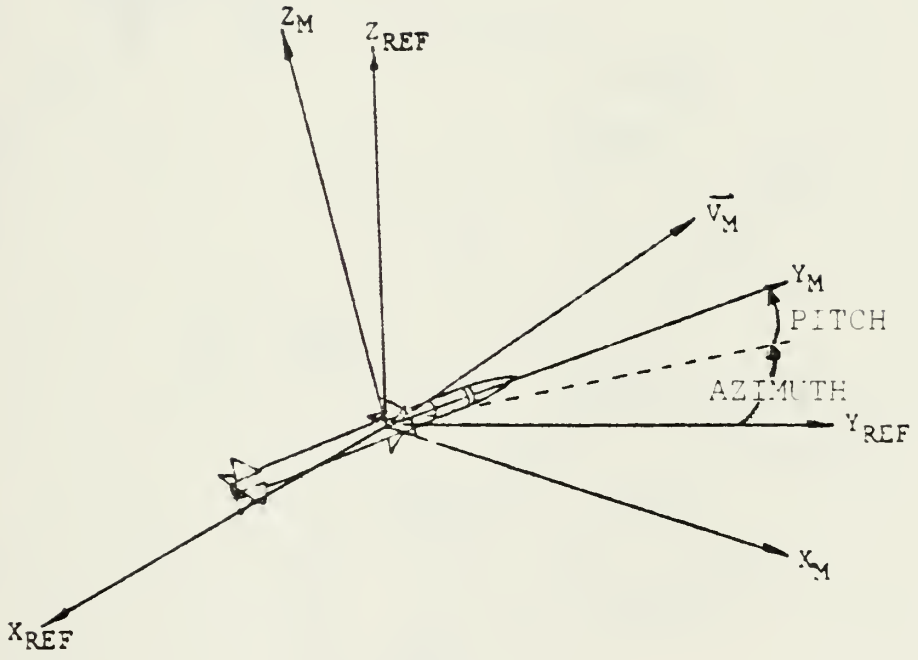


Fig. A-8 Missile Pitch and Azimuth Angles

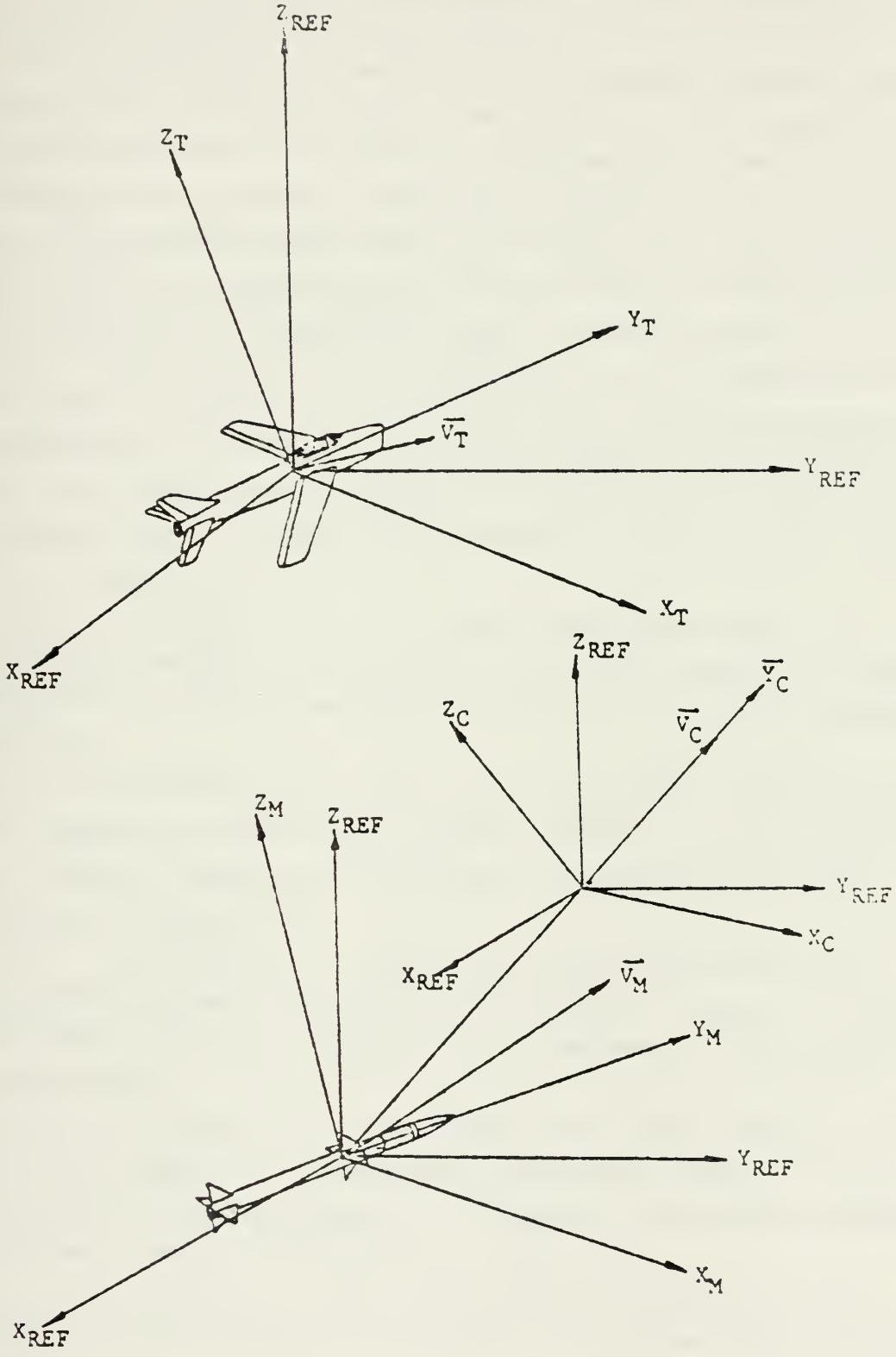


Fig. A-9 SCAN Coordinate System Relationships

For user specified trajectories or point detonation of the missile warhead, do steps 21-25.

For simulations to determine the average survival probability for a fixed guidance miss distance, do steps 26-28.

For situations in which the user desires the simulation to generate the initial engagement geometry from a distribution of encounter conditions and in which the miss distance for individual trajectories is drawn from a bi-variant normal distribution of specified CEP, do steps 29-31.

21. In columns 1 - 30: enter the X, Y, Z components of the initial position of the missile (for instantaneous detonation this is the point where the warhead will explode) measured in the target coordinate system (with respect to the target CG). A set of real values, 3F10.3 format, measured in feet.

22. In columns 31 - 40: enter the target angle of attack with respect to the target velocity vector. A real value, F10.3 format, measured in degrees, counterclockwise positive.

23. In columns 41 - 50: enter the target sideslip angle with respect to the target velocity vector. A real value, F10.3 format, measured in degrees, the counterclockwise direction is positive.

24. In columns 51 - 60: enter the missile angle of attack with respect to the missile velocity vector. A real value, F10.3 format, measured in degrees, counterclockwise positive.

25. In columns 61 - 70: enter the missile sideslip angle with respect to the missile velocity vector. A real value, F10.3 format, measured in degrees, positive counterclockwise. Press ENTER and go to step 32.

26. In columns 1 - 10: enter the closest point of approach of the missile trajectory to the specified aimpoint

on the target. A real value, F10.3 format, measured in feet from the missile aimpoint radially outward.

27. In columns 11 - 20: enter the target angle of attack with respect to the target velocity vector. A real value, F10.3 format, measured in degrees, counterclockwise positive.

28. In columns 21 - 30: enter the target sideslip angle with respect to the target velocity vector. A real value, F10.3 format, measured in degrees, positive counterclockwise. Press ENTER and go to step 32.

29. In columns 1 - 10: enter the circular error probable (CEP) of the missile. A real value, F10.3 format, measured in feet. The CEP will be centered about the missile aimpoint by the program.

30. In columns 11 - 20: enter the target angle of attack with respect to the target velocity vector. A real value, F10.3 format, measured in degrees, counterclockwise positive.

31. In columns 21 - 30: enter the target sideslip angle with respect to the target velocity vector. A real value, F10.3 format, measured in degrees, positive counterclockwise. Press ENTER.

32. You have now completed your CASE DATA file. Press ENTER. The cursor should now be in the lower left corner of the screen and an END OF FILE statement should show up as the last entry in the file you just finished. Type "FILE" and press ENTER. Your screen should now rever to the format it had prior to entering the XEDIT mode. Type "L" and press ENTER. You should now see a listing of all the files on your alloted disk space. The file you just created should now be included in that listing.

F. MODIFYING A CASE FILE

This section will deal with modifying an existing CASE DATA file. Extensive reference will be made to Section E of this Manual.

1. Turn on the terminal and log onto the system.

2. Type "XEDIT Fn Fm" and press ENTER. Fn refers to the filename of the file you wish to modify. Fm refers to the filemode of the file you wish to modify (usually DATA). An example is: XEDIT CASE1 DATA.

3. You should now have the data in the file visible on the display screen. If the data covers more than one full screen, use the following commands to survey the file:

(a) Press "ALT" and "PF8" keys at the same time to advance one screen forward in the data file.

(b) Press "ALT" and "PF8" keys at the same time to revert back one screen in the data file.

4. Use Section E of this Manual to locate the positions of the values you wish to change. Display the section of the data file to be changed using instruction 3 above.

5. By using the four cursor positioning keys just to the right of the main keyboard, position the cursor under the values you wish to modify. Simply type the new values in over the previous ones. Make sure that the new values are justified correctly and in the right format as shown in Section C.

6. After completing your changes on the page on the screen, press ENTER. The cursor will return to the lower left corner of the screen and the changes will have been made. You must enter all changes on each full screen of data before "paging" through the file.

7. Once all changes have been entered and the cursor is back in the lower left corner of the screen, type "FILE" and press ENTER. This will return you to the normal

operating mode of the system and store the corrected file back on your disk space. The corrected file is now ready for use.

G. OTHER MODIFICATIONS AND EXAMPLES

Modifications to the GEOM DATA files, which contain the target geometric model, limiting parameters and kill expressions, are also possible. Changes to these parameters are often very complex and are not advised for someone unfamilar with the mechanics of the SCAN program. Information concerning these changes can be found in the User Manual for the SCAN program (master copy, not modified for NPS).

Figure A-10 shows typical WARHEAD DATA and CASE DATA files.

Figure A-11 is a quick reference guide to the WARHEAD file input values.

Figure A-12 is a quick reference guide to the CASE file input values.

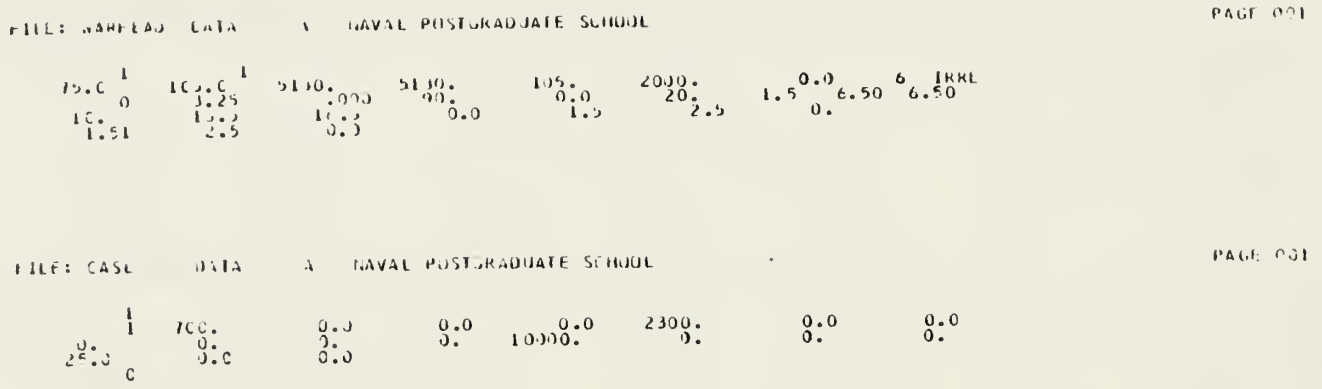


Fig. A-10 WARHEAD DATA and CASE DATA Examples

INDEX	0	10	20	30	40	50	60	70	75	80
LINE 1	1	1								
	no. of static polar zones col. 1-2	no. of mass classes per zone col. 11								
LINE 2	0.0	180.0	5000.0	5000.0	105.0	1000.0	0.5	6	IRRE	
	forward lower angle of polar zone 1 degrees	rear or upper angle of polar zone 1 degrees	lower fragment velocity in polar zone 1 ft/sec	upper fragment velocity in polar zone 1 ft/sec	fragment mass of polar zone 1 grains	no. of fragments in polar zone 1	fragment init. pos along missile X-axis feet	71-72 mat'l code from list	frag shape	
LINE 3	0	1.5	0.0	90.0	0.0	150.0	0.5	67 1.5	72 6.5	
	fuze type from list col. 1	fuze pos. along missile axis from WH center feet	fuze delay time seconds	fuze look angle from centerline degrees	std. dev. of fuze angle degrees	fuze cut-off range feet	misl body rad. ft.	pos. of misl nose fm WH ft.	dist. of missile aft end from WH feet	
LINE 4	15.0	25.0	20.0	10.0	5.0	0.0	0.0			
	fuselage blast radius from centerline feet	distance from CG to front of cylinder feet	distance from CG to back of cylinder feet	wing blast radius from centerline feet	X-coord. of inboard end of wing CL ft.	Y-coord of inboard end of wing CL feet	Z-coord. of inboard end of wing CL feet			

Fig. A-11(a) WARHEAD DATA Quick Reference Guide

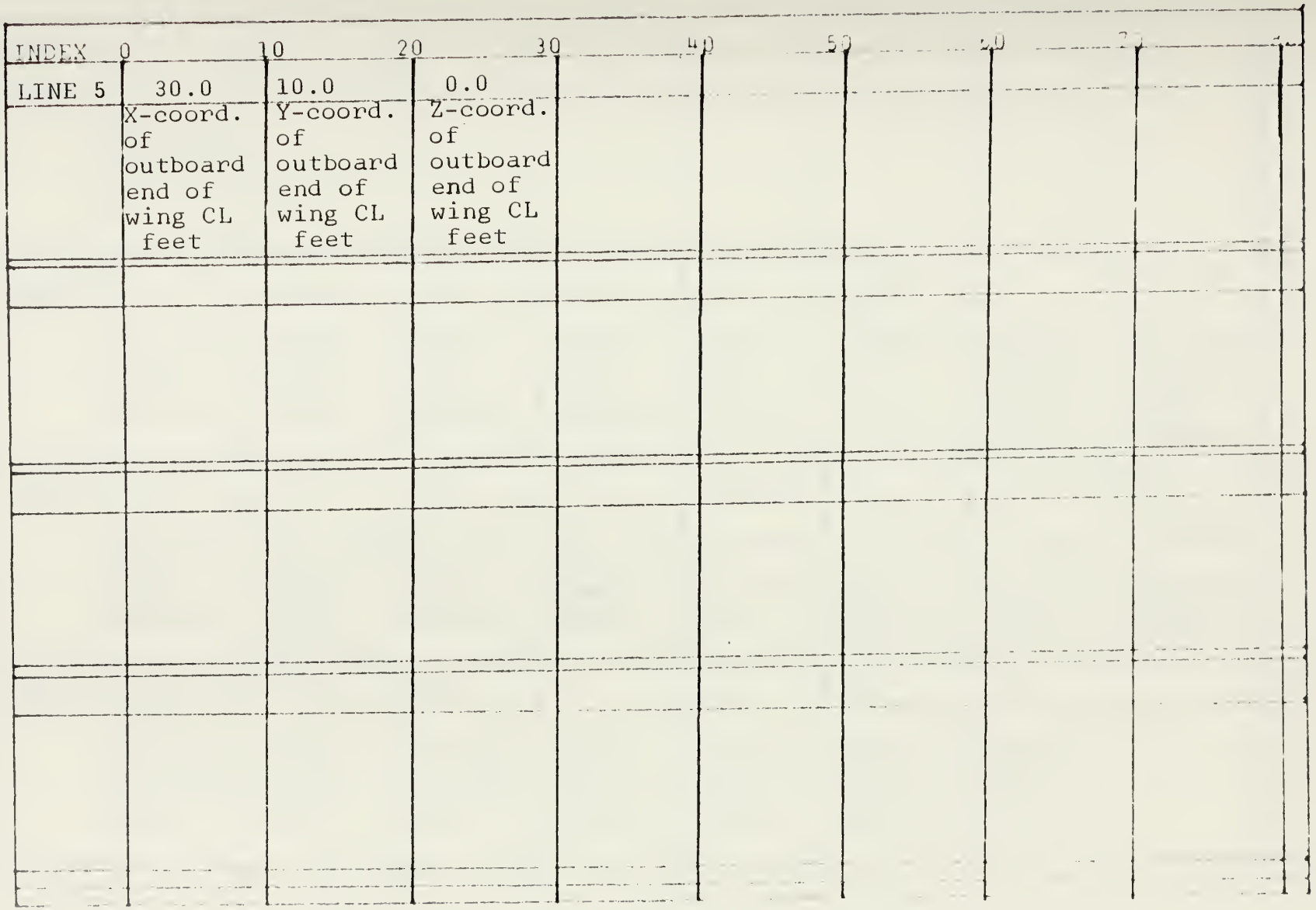


Fig. A-11(b) WARHEAD DATA Quick Reference Guide

INDEX	0	10	20	30	40	50	60	70	80
LINE 1	0								
	type of trajectory col. 1								
LINE 2	1	2000.0	0.0	0.0	0.0	3000.0	0.0	0.0	
	no. of runs col. 1	target speed ft/sec	target roll angle degrees	target pitch angle degrees	target yaw angle degrees	missile speed ft/sec	missile angle of attack degrees	std. dev. missile attack angle degrees	
LINE 3	0.0	0.0	0.0	0.0	10000.0	0.0	0.0	0.0	
	missile pitch angle degrees	std. dev. missile pitch angle degrees	missile azimuth angle degrees	std. dev missile azimuth degrees	altitude of scenario feet	X value of missile aimpoint feet	Y value of missile aimpoint feet	Z value of missile aimpoint feet	
LINE 4	100.0	100.0	50.0	0.0	0.0	0.0	0.0		
	initial missile X-coord. feet	initial missile Y-coord. feet	initial missile Z-coord. feet	target angle of attack degrees	target sideslip angle degrees	missile angle of attack degrees	missile sideslip angle degrees		

Fig. A-12(a) CASE DATA Quick Reference Guide

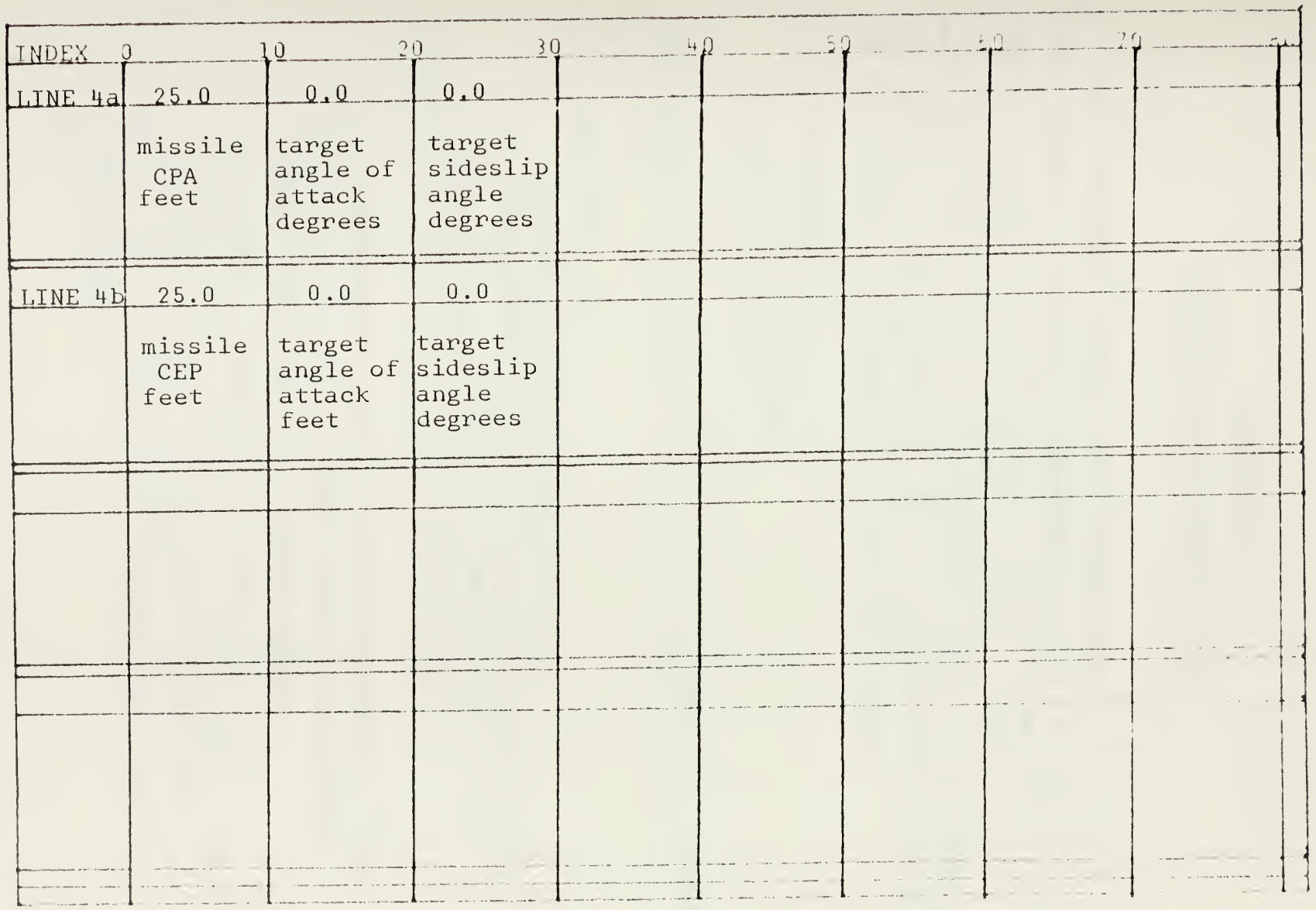


Fig. A-12(b) CASE DATA Quick Reference Guide

APPENDIX B

PROGRAM CHANGES MADE TO SCAN

This Appendix contains the changes made to the SCAN Endgame program at the Naval Postgraduate School in order to function on the IBM 3033 computer system and to output data for the SPDRAW computer graphics program. Several changes were also made in order to simplify the task of interpreting the output. The changes made will be listed and referenced by the routine or subroutine in which they occur.

A. MAIN PROGRAM

1. Added the following:

```
REWIND 19
```

This ensures that the data file used for graphics is stored correctly.

B. BLOCK DATA

1. Changed the DATA NAMARR array from:

```
DATA NAMARR/4HMAGN,4HSIUM,4HALUM,4H2024,4HTITN,4HALOY,
1  4HSTEE,4HFACE,4HSTEE,4HMILD,4HSTEE,4HHARD,4HLEXA,1H ,
2  4HPLX-,4HGLAS,4HDORO,1H ,4HBULL,4HRES /
```

to:

```
DATA NAMARR/4HMAGN,4HSIUM,4HALUM,4H2024,4HTITN,4HALOY,
1  4HFACE,4HSTEL,4HMILD,4HSTEL,4HHARD,4HSTEL,4HLEXN,1H ,
2  4HPLXI,4HGLAS,4HDORO,1H ,4HBULL,4HRESG/
```

2. Changed the DATA VULTYP array from:

```
DATA VULTYP/4HENER,4HDENS,4HS F ,4HVUL ,4HAREA,4HRMVL,
1  4HNON ,4HVUL ,4HNV  ,4HDHIT,4HNV  ,4HIR S,4HED  ,
1  4HRRTP,4HS F ,4HRRTP,4HAR  ,4HRRTP,4HNV  ,4HRRTP,
1  4HDHIT,4HRRTP/
```

to:

```
DATA VULTYP/4HENER,4HDENS,4HSING,4HFRAG,4HAREA,4HRMVL,
1  4H    ,4HNON ,4HNONV,4HDHIT,4HNONV,4HIR S,4HED R,
1  4HTRNS,4HSF R,4HTRNS,4HAR R,4HTRNS,4HNV R,4HTRNS,
1  4HNVDH,4HTRNS/
```

3. Changed the DATA SKNTYP array from:

```
DATA SKNTYP/4HEXTE,4HSKIN,4HEXTE,4HSLID,4HINTE,4HSKIN,
1  4HINTE,4HSLID/
```

to:

```
DATA SKNTYP/4HEXTE,4HSKIN,4HEXTE,4HSOLI,4HINTE,4HSKIN,
1  4HINTE,4HSOLI/
```

4. Changed DATA TYP(2) from:

```
DATA TYP(2)/4HHLPD/
```

to:

```
DATA TYP(2)/4HELIP/
```

5. Changed DATA TYP(4) from:

```
DATA TYP(4)/4HHCON/
```

to:

```
DATA TYP(4)/4HELCO/
```

C. SUBROUTINE READIN

1. Added the following:

```
COMMON/FCTR/FACTOR
```

This is a value used to correctly dimensionalize the output data for graphics.

D. SUBROUTINE UPDATE

1. Added the following:

```
COMMON/FCTR/FACTOR
```

This is a value recieved from Subroutine READIN to correctly dimensionalize the output data for graphics.

2. Added the following:

```
REAL RANC,TPC
```

These are output values.

```
    3.   Changed the following:
    DIMENSION RA(3),RANDHT(3),RANM(3),VBAR(3),VX(3)
    to:
    DIMENSION RA(3),RANDHT(3),RANM(3),VBAR(3),VX(3),
              RANC(3),TPC(3)
    4.   Added the following:
    INTEGER IHIT
```

An output value.

```
    5.   Added the following:
    REAL FRAGQ,TFRST,THIT
```

These are output values.

```
    6.   Added the following:
    TFRST = 0.
    THIT = 0.
    IHIT = 0
    FRAGQ = 0.
```

This initializes the new variables.

```
    7.   Added the following:
    RANC(1) = RAN(1) / FACTOR
    RANC(2) = RAN(2) / FACTOR
    RANC(3) = RAN(3) / FACTOR
    TPC(1) = TP(1) / FACTOR
    TPC(2) = TP(2) / FACTOR
    TPC(3) = TP(3) / FACTOR
    WRITE (19) FRAGM,TFRST,RANC,THIT,TPC,IHIT
```

This outputs the fragment mass and it's coordinates as it enters a component.

```
    8.   Added the following:
    RANC(1) = RAN(1) / FACTOR
    RANC(2) = RAN(2) / FACTOR
```

```
    RANC(3) = RAN(3) / FACTOR
    TPC(1) = TP(1) / FACTOR
    TPC(2) = TP(2) / FACTOR
    TPC(3) = TP(3) / FACTOR
    WRITE (19) FRAGM,TFRST,RANC,THIT,TPC,IHIT
```

This outputs the fragment mass and it's coordinates as it exits a component.

```
    9.   Added the following:
    FRAGQ = -999.
    RANC(1) = RAN(1) / FACTOR
    RANC(2) = RAN(2) / FACTOR
    RANC(3) = RAN(3) / FACTOR
    TPC(1) = TP(1) / FACTOR
    TPC(2) = TP(2) / FACTOR
    TPC(3) = TP(3) / FACTOR
    WRITE (19) FRAGQ,TFRST,RANC,THIT,TPC,IHIT
```

This outputs the end-of-file data for the graphics program.

```
E.  FUNCTION ASIN(X)
    1.   Changed FUNCTION ASIN(X) from:
    FUNCTION ASIN(X)
    ASIN = ARSIN(X)
    RETURN
    END
    to:
    FUNCTION AFNSN(X)
    AFNSN = ARSIN(X)
    RETURN
    END
```

This was required for compatibility with the IBM 3033 computer system. All calls for ASIN(X) in the entire program have been changed to AFNSN(X).

F. File definition statements for SCAN

1. The file definitions in effect when using SCAN are:

```
FILEDEF 01 DISK SCAN1 FORTRAN
FILEDEF 05 DISK CASE DATA
FILEDEF 06 DISK SCAN2 FORTRAN
FILEDEF 11 DISK GEOM DATA
FILEDEF 12 DISK WARHEAD DATA
FILEDEF 19 DISK IMPACT DATA
FILEDEF 30 DISK FILE DATA
```

These file definitions are contained in an executive program labeled SCANMAIN EXEC.

LIST OF REFERENCES

1. Air-to-Air Subgroup, Air Target End Game Methodology Panel Publication, ATTACK Computer Program, Volume I, User Manual, June 1974

2. Weapons Evaluation Department, Pacific Missile Test Center Publication, SCAN, A Computer Program for Survivability Analysis, Volume I, User Manual, July 1976

3. Aeronautical Systems Division, Deputy for Development Planning Publication, SESTEM II Missile Endgame Model Description, May 1977

4. Anti-Air Missile Evaluation Group, Joint Technical Coordinating Group for Munitions Effectiveness, REFMOD-2 Computer Program User Manual, Mar. 1981

5. Hayes, T. M., The Installation of Computer Graphics for Survivability Studies and a Discussion of PK/d Functions, Masters Thesis, Naval Postgraduate School, June 1981

BIBLIOGRAPHY

Aeronautical Systems Division, Deputy for Development Planning Publication, SESTEM II Missile Endgame Model description, May 1977

Air-to-Air Subgroup, Air Target End Game Methodology Panel Publication, ATTACK Computer Program, Volume I, User Manual, June 1974

Air-to-Air Subgroup, Air Target End Game Methodology Panel Publication, ATTACK Computer Program, Volume II, Analysis Manual, June 1974

Anti-Air Missile Evaluation Group, Joint Technical Coordinating Group for Munitions Effectiveness, REFMOD-1 Computer Program User Manual, Dec. 1979

Anti-Air Missile Evaluation Group, Joint Technical Coordinating Group for Munitions Effectiveness, REFMOD-2 Computer Program User Manual, Mar. 1981

Gottfried, B. S., Programming with FORTRAN IV, Quantum, 1972

Naval Weapons Center Technical Note 4565-1-70, Air-to-Air Terminal Encounter Simulation Program I, Volume III Fuze Addendum, by J. E. Musch and John E. Strong, May 1973

Weapons Evaluation Department, Pacific Missile Test Center Publication, SCAN, A Computer Program for Survivability Analysis, Volume I, User Manual, July 1976

Weapons Evaluation Department, Pacific Missile Test Center Publication, SCAN, A Computer Program for Survivability Analysis, Volume II, Analyst Manual, July 1976

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 67 Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
4.	Professor R. E. Ball, Code 67bp Department of Aeronautics Naval Postgraduate School Monterey, California 93940	4
5.	Pat Caulfield AFATL/DOYD Eglin Air Force Base, Florida 32542	1
6.	Commander, Naval Weapons Center Attention: Ken Grant, Code 3386 China Lake, California 93555	1
7.	Bruce Nofrey, Code 1245 Pacific Missile Test Center Point Mugu, California 93042	1
8.	LT Christopher Kenneth Fair, USN. 176 160th Place S.E. Bellevue, Washington 98008	1

Thesis
F
213
c.1

193769
Thesis
F213 Fair
c.1 A comparison of computer warhead-target endgame simulations and recommendations pertaining to future endgame programs.

20 APR 83

193769
Thesis
F213 Fair
c.1 A comparison of computer warhead-target endgame simulations and recommendations pertaining to future endgame programs.